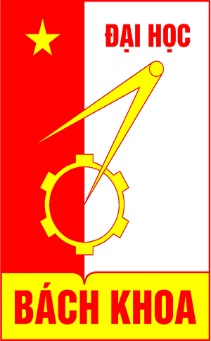
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG ĐIỆN – ĐIỆN TỬ**



**ĐỒ ÁN 2**

**XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN CHIẾU SÁNG CHO NHÀ THÔNG MINH BẰNG CÔNG NGHỆ MẠNG BLE MESH**

**NGUYỄN VĂN ĐỘ**

do.nv212759@sis.hust.edu.vn

**Ngành Kỹ thuật Điều khiển và Tự động hóa**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Trần Thị Anh Xuân  Chữ ký của GVHD |
| **Khoa:** | Tự động hóa |
| **Trường:** | Điện – Điện tử |
| **HÀ NỘI, 4/2024** | |

|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **ĐẠI HỌC BÁCH KHOA HÀ NỘI** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc** |

**NHIỆM VỤ**

**ĐỒ ÁN 2**

Họ và tên sinh viên: **Nguyễn Văn Độ**

Khóa: **66**

Trường: **Điện- Điện tử**

Ngành:

1. *Tên đề tài:*
2. *Nội dung đề tài:*

Công việc

1. *Thời gian giao đề tài: Ngày 08 Tháng 03 Năm 2024*
2. *Thời gian hoàn thành:*

*Ngày ...... tháng ...... năm ......*

**CÁN BỘ HƯỚNG DẪN**

(Ký, ghi rõ họ tên)

**LỜI CẢM ƠN**

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng năm 2024  Sinh viên thực hiện  **Nguyễn Văn Độ** |

**TÓM TẮT ĐỒ ÁN**

**MỤC LỤC**

[**DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT** viii](#_Toc163599797)

[**CHƯƠNG 1: GIỚI THIỆU CHUNG** 1](#_Toc163599798)

[**1.** **Tổng quan hệ thống nhà thông minh** 1](#_Toc163599799)

[**1.1.** **Khái niệm về IoT** 1](#_Toc163599800)

[**1.2.** **Khái niệm Smart Home** 1](#_Toc163599801)

[**1.3.** **Các thành phần trong một hệ thống Smart Home** 2](#_Toc163599802)

[**1.4.** **Lợi ích của hệ thống Smart Home** 2](#_Toc163599803)

[**2.** **Khảo sát một số thiết bị trong Smart Home** 3](#_Toc163599804)

[**2.1.** **Điều khiển chiếu sáng trong hệ thống Smart Home** 3](#_Toc163599805)

[**2.2.** **Khảo sát chi tiết một số thiết bị đèn thông minh thực tế trên thị trường** 3](#_Toc163599806)

[**2.2.1.** **Bóng đèn thông minh đổi màu RGB của Rạng Đông** 3](#_Toc163599807)

[**2.2.2.** **Bóng đèn thông minh WiFi Yeelight LED Buld 1S của SmartHome Kit** 4](#_Toc163599808)

[**2.3.** **Khảo sát chi tiết một số thiết bị công tắc thông minh thực tế trên thị trường** 5](#_Toc163599809)

[**2.3.1.** **Công tắc thông minh Rạng Đông** 5](#_Toc163599810)

[**2.3.2.** **Công tắc thông minh Lumi** 6](#_Toc163599811)

[**2.3.3.** **Công tắc thông minh Tuya** 8](#_Toc163599812)

[**2.4.** **Lựa chọn công nghệ mạng không dây** 9](#_Toc163599813)

[**2.4.1.** **Khảo sát về công nghệ mạng không dây phổ biến** 9](#_Toc163599814)

[**3.** **Lựa chọn công nghệ mạng và lên phương án xây dựng hệ thống** 10](#_Toc163599815)

[**3.1.** **Lựa chọn công nghệ mạng không dây cho hệ thống điều khiển chiếu sáng** 10](#_Toc163599816)

[**3.2.** **Phương án xây dựng hệ thống điều khiển chiếu sáng thông minh cho Smart Home** 11](#_Toc163599817)

[**3.2.1.** **Thiết bị** 11](#_Toc163599818)

[**3.2.2.** **Xây dựng WEB SERVER** 13](#_Toc163599819)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 14](#_Toc163599820)

[**1.** **Công nghệ Bluetooth Low Energy MESH** 14](#_Toc163599821)

[**1.1.** **Lịch sử phát triển công nghệ BLE** 14](#_Toc163599822)

[**1.1.1.** **Định nghĩa cơ bản các layer** 15](#_Toc163599823)

[**1.1.2.** **Atribute Protocol – ATT** 16](#_Toc163599824)

[**1.1.3.** **GATT - Generic Attribute Profile** 17](#_Toc163599825)

[**1.2.** **Công nghệ BLE MESH** 18](#_Toc163599826)

[**1.2.1.** **Giới thiệu về BLE MESH** 18](#_Toc163599827)

[**1.2.2.** **Các thuật ngữ trong BLE MESH** 20](#_Toc163599828)

[**1.2.3.** **Model** 21](#_Toc163599829)

[**1.2.4.** **Giới thiệu về Publish, Subscribe và security trong BLE MESH** 23](#_Toc163599830)

[**1.2.5.** **Kiến trúc mạng BLE MESH** 23](#_Toc163599831)

[**1.2.5.1.** **Bearer Layer** 24](#_Toc163599832)

[**1.2.5.2.** **Network Layer** 25](#_Toc163599833)

[**1.2.5.2.1.** **Addresses** 25](#_Toc163599834)

[**1.2.5.2.2.** **Network PDU (Protocol Data Unit)** 28](#_Toc163599835)

[**1.2.5.3.** **Bảo mật trong mạng BLE MESH** 33](#_Toc163599836)

[**1.2.5.4.** **Quá trình một thiết bị tham gia mạng mesh** 35](#_Toc163599837)

[**1.2.5.4.1.** **Một số định nghĩa cơ bản về provisioning process** 35](#_Toc163599838)

[**1.2.5.4.2.** **Generic Provisioning PDU** 37](#_Toc163599839)

[**1.2.5.4.3.** **Chi tiết về quá trình gia nhập mạng** 39](#_Toc163599840)

[**1.2.5.5.** **Quá trình chuyển một bản tin** 41](#_Toc163599841)

[**2.** **WiFi provisioning thông qua SoftAP** 42](#_Toc163599842)

[**2.1.** **Chế độ softAP trên vi điều khiển** 42](#_Toc163599843)

[**2.2.** **Kết nối WiFi router thông qua chế độ SoftAP trên vi điều khiển** 43](#_Toc163599844)

[**3.** **Giao thức MQTT** 45](#_Toc163599845)

[**3.1.** **Khái niệm giao thức MQTT** 45](#_Toc163599846)

[**3.2.** **Kiến trúc Pub/Sub** 45](#_Toc163599847)

[**3.3.** **Quality of service** 46](#_Toc163599848)

[**CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG** 47](#_Toc163599849)

[**1.** **Tổng quan hệ thống** 47](#_Toc163599850)

[**1.1.** **Mục tiêu thiết kế** 47](#_Toc163599851)

[**1.2.** **Sơ đồ khối tổng quan** 47](#_Toc163599852)

[**2.** **Thiết kế phần cứng** 48](#_Toc163599853)

[**2.1.** **Mục tiêu thiết kế** 48](#_Toc163599854)

[**2.2.** **Thiết kế công tắc âm tường** 48](#_Toc163599855)

[**2.2.1.** **Lựa chọn linh kiện** 48](#_Toc163599856)

[**2.2.2.** **Sơ đồ khối thiết bị** 50](#_Toc163599857)

[**2.2.3.** **Thiết kế mạch nguyên lý** 51](#_Toc163599858)

[**3.** **Thiết kế phần mềm nhúng** 52](#_Toc163599859)

[**3.1.** **Mục tiêu thiết kế** 52](#_Toc163599860)

[**3.1.1.** **Thiết bị gateway** 52](#_Toc163599861)

[**3.1.2.** **Thiết bị switch** 53](#_Toc163599862)

[**3.2.** **Triển khai thiết kế phần mềm nhúng** 53](#_Toc163599863)

[**3.2.1.** **Thiết bị gateway** 53](#_Toc163599864)

[**3.2.1.1.** **Kết nối WIFI cho gateway** 53](#_Toc163599865)

[**3.2.1.2.** **Thực hiện provisioning BLE MESH cho thiết bị** 56](#_Toc163599866)

[**3.2.1.3.** **Thực hiện reset cấu hình lại thiết bị khi cần thiết** 56](#_Toc163599867)

[**3.2.1.4.** **Phương thức để gateway gửi bản tin tới thiết bị đích và điều khiển nhóm** 57](#_Toc163599868)

[**3.2.1.5.** **Bóc tách bản tin gửi từ cloud để điều khiển thiết bị đích và điều khiển nhóm** 57](#_Toc163599869)

[**3.2.1.6.** **Lưu đồ thuật toán của gateway** 58](#_Toc163599870)

[**3.2.2.** **Thiết bị Switch** 58](#_Toc163599871)

[**3.2.2.1.** **Phương thức để thiết bị nhận bản tin từ gateway và phản hồi bản tin** 59](#_Toc163599872)

[**3.2.2.2.** **Điều khiển thiết bị switch bằng smartphone thông qua BLE** 59](#_Toc163599873)

[**3.2.2.3.** **Điều khiển đèn bằng cách bấm nút thủ công** 59](#_Toc163599874)

[**3.2.2.4.** **Lưu đồ thuật toán của thiết bị switch** 59](#_Toc163599875)

[**4.** **Thiết kế cloud server** 59](#_Toc163599876)

[**5.** **Thiết kế giao diện người dùng** 59](#_Toc163599877)

# **DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| Bluetooth Low Energy | BLE |

ADV Advertising

PDU Protocol Data unit

**DANH MỤC HÌNH VẼ**

[Hình 1. 1 Hệ thống Smart Home 1](#_Toc161316404)

[Hình 1. 2 : Hệ thống Smart Home 2](#_Toc161316405)

[Hình 1. 3 Bóng Led pulb đổi màu RGB của Rạng Đông 4](#_Toc161316406)

[Hình 1. 4 Bóng đèn thông minh YeeLight 5](#_Toc161316407)

[Hình 1. 5 Công tắc thông minh BLE 3 nút bấm dáng vuông 6](#_Toc161316408)

[Hình 1. 6 Công tắc Luto Kính phẳng Viền nhôm 7](#_Toc161316409)

[Hình 1. 7 Công tắc Luto Kính phẳng viền nhôm 7](#_Toc161316410)

[Hình 1. 8 Công tắc Wifi viền kim loại điều khiển qua app Tuya TUYA-VKL-EU 8](#_Toc161316411)

[Hình 1. 9 Mô hình tổng quan thiết bị đóng vai trò gateway 11](#_Toc161316412)

[Hình 1. 10 Mô hình BLE MESH 12](#_Toc161316413)

[Hình 1. 11 Bóng đèn đổi màu theo ngữ cảnh 13](#_Toc161316414)

[Hình 2. 1 Các layer của Bluetooth Low Energy 15](#_Toc162773884)

[Hình 2. 2 Cấu trúc dữ liệu ATT 16](#_Toc162773885)

[Hình 2. 3 Tổ chức dữ liệu ATT trong server 17](#_Toc162773886)

[Hình 2. 4 Phương pháp truy cập dữ liệu theo mô hình client – server 17](#_Toc162773887)

[Hình 2. 5 GATT - Phân cấp cơ sở dữ liệu ATT 18](#_Toc162773888)

[Hình 2. 6 Ảnh thể hiện mối liên hệ giữa GATT và ATT 18](#_Toc162773889)

[Hình 2. 7 Công nghệ mạng BLE MESH 19](#_Toc162773890)

[Hình 2. 8 Sự liên kết trong mạng MESH 21](#_Toc162773891)

[Hình 2. 9 Sự giao tiếp giữa Client Model và Server Model 22](#_Toc162773892)

[Hình 2. 10 Kiến trúc mạng BLE MESH 24](#_Toc162773893)

[Hình 2. 11 Mesh message với AD type 24](#_Toc162773894)

[Hình 2. 12 Khung địa chỉ kiểu Unassigned 26](#_Toc162773895)

[Hình 2. 13 Khung địa chỉ kiểu Unicast 26](#_Toc162773896)

[Hình 2. 14 Khung địa chỉ ảo 27](#_Toc162773897)

[Hình 2. 15 Khung của group address 27](#_Toc162773898)

[Hình 2. 16 Khung Network PDU 29](#_Toc162773899)

[Hình 2. 17 Lưu đồ thuật toán nhận network PDU 30](#_Toc162773900)

[Hình 2. 18 Khung bản tin truy cập không bị chia 31](#_Toc162773901)

[Hình 2. 19 Định nghĩa các thành phần trong khung 31](#_Toc162773902)

[Hình 2. 20 Khung bản tin điều khiển không chia 31](#_Toc162773903)

[Hình 2. 21 Định nghĩa trường trong khung bị chia 32](#_Toc162773904)

[Hình 2. 22 Khung bị chia 32](#_Toc162773905)

[Hình 2. 23 Bản tên bị phân chia dần từ tầng trên xuống tầng dưới 32](#_Toc162773906)

[Hình 2. 24 Khung của upper Transport Layer 33](#_Toc162773907)

[Hình 2. 25 Provisioning protocol stack 36](#_Toc162773908)

[Hình 2. 26 Phân lớp của Provisioning Protocol 37](#_Toc162773909)

[Hình 2. 27 Khung của Provisioning PDU 37](#_Toc162773910)

[Hình 2. 28 Định dạng kiểu cho Provisioning control 37](#_Toc162773911)

[Hình 2. 29 Transaction Start PDU 37](#_Toc162773912)

[Hình 2. 30 Transaction ACK format 38](#_Toc162773913)

[Hình 2. 31 Transaction continue PDU 38](#_Toc162773914)

[Hình 2. 32 Provisioning Bearer Frame 38](#_Toc162773915)

[Hình 2. 33 Frame của Provisioning PDU 38](#_Toc162773916)

[Hình 2. 34 Dang sách các dạng của Provisioning PDU 39](#_Toc162773917)

[Hình 2. 35 Mời gia nhập và kiểm tra khả năng kết nối 39](#_Toc162773918)

[Hình 2. 36 Cấp phép không có khóa chung OOB 40](#_Toc162773919)

[Hình 2. 37 Cấp phép có khóa chung OOB 40](#_Toc162773920)

[Hình 2. 38 Phôi phối dữ liệu cấp phép 41](#_Toc162773921)

[Hình 2. 39 Quá trình chuyển một bản tin 42](#_Toc162773922)

**DANH MỤC BẢNG BIỂU**

[Bảng 1 Khảo sát một số công nghệ mạng không dây phổ biến 9](#_Toc161316561)

[Bảng 2. 1 Chức năng của các layer trong BLE 16](#_Toc162773939)

[Bảng 2. 2 Thuật ngữ trong BLE MESH 20](#_Toc162773940)

[Bảng 2. 3 Phân bổ 16 bit địa chỉ 26](#_Toc162773941)

[Bảng 2. 4 Định nghĩa các trường trong Network PDU 28](#_Toc162773942)

[Bảng 2. 5 Các dạng của PDU trong Transport Lower Layer 31](#_Toc162773943)

[Bảng 2. 6 Khung truyền của PB ADV 36](#_Toc162773944)

[Bảng 2. 7 Chi tiết bên trong của PB ADV PDU 36](#_Toc162773945)

[Bảng 2. 8 Dạng dữ liệu cấp phép của khóa phiên 41](#_Toc162773946)

# **CHƯƠNG 1: GIỚI THIỆU CHUNG**

## **Tổng quan hệ thống nhà thông minh**

### **Khái niệm về IoT**

IoT (Internet of Things) đề cập đến một hệ thống các thiết bị vật lý, xe cộ, thiết bị gia dụng và các đối tượng khác được tích hợp phần mềm và kết nối mạng. Các thiết bị này có khả năng thu thập và trao đổi dữ liệu qua internet và không cần sự tương tác trực tiếp của con người.

Mục đích của IoT là tạo ra một mạng lưới kết nối vạn vật thông minh, cho phép chúng tương tác và chia sẻ thông tin với nhau một cách tự động trong thời gian thực. Mở ra các ứng dụng IoT rất đa dạng như Smart Home, Công nghiệp thông minh, Giao thông thông minh, Y tế, Nông nghiệp thông minh….

### **Khái niệm Smart Home**

Smart Home, hay Nhà thông minh, là kiểu nhà hoặc tòa nhà được lắp đặt các thiết bị điện, điện tử được điều khiển hoặc tự động hoá hoặc bán tự động với mục đích hay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển, mang tới sự tiện lợi và tiết kiệm thời gian.

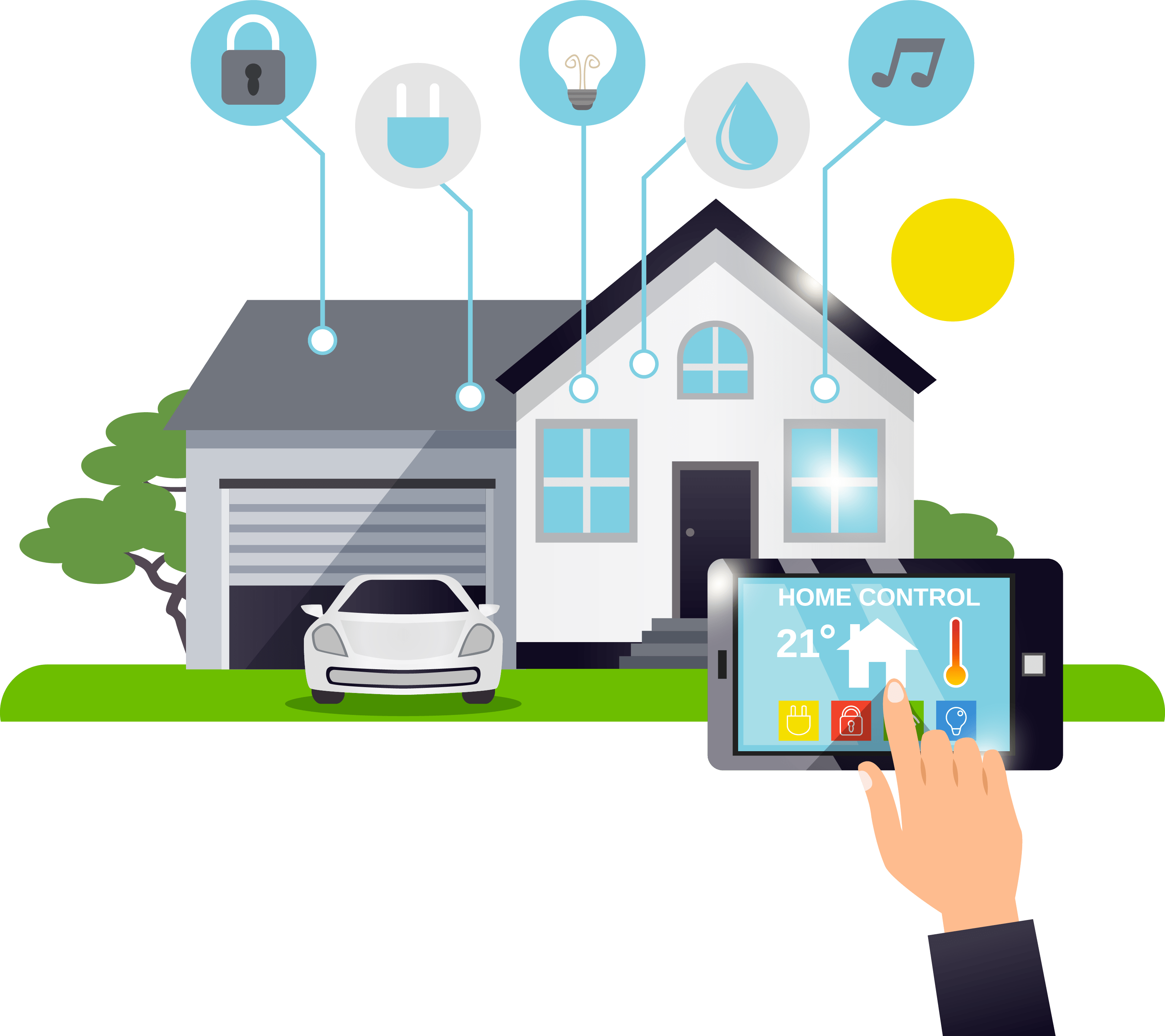
Hệ thống Smart Home giao tiếp với người dùng thông qua giao diện Web, ứng dụng điện thoại hoặc thông qua các bảng điều khiển đặt trong nhà, … Nhờ tích hợp các công nghệ như hồng ngoại, IoT, đám mây, … Smart Home có thể giúp người dùng thực hiện việc kiểm soát ngôi nhà từ xa, tự động quản lý công việc nhà, …

A room with wifi icons

Description automatically generated

Hình 1. 1 Hệ thống Smart Home

### **Các thành phần trong một hệ thống Smart Home**



Hình 1. 2 : Hệ thống Smart Home

Một hệ thống Smart Home tiêu chuẩn bao gồm:

* *Phần cứng:* Là các thành phần vật lý trong hệ thống bao gồm các thiết bị như camera thông minh, ổ cắm thông minh, bóng đèn thông minh, cảm biến thông minh, Router mạng, …
* *Phần mềm nhúng:* Là phần mềm được lập trình và cài đặt lên các thiết bị nhúng, cho phép chúng điều khiển các hoạt động và thực hiện việc giao tiếp với các thiết bị khác hoặc giao tiếp với các phần khác trong hệ thống.
* *Máy chủ đám mây (Cloud Server):* Là một máy chủ có hiệu suất cao luôn bật, được quản lý và bảo vệ nghiêm ngặt. Máy chủ đám mây kết nối với phần cứng thông qua phần mềm nhúng và kết nối với người dùng thông qua giao diện người dùng. Máy chủ đám mây được cấu hình để xử lý và lưu trữ tất cả các thông tin trong hệ thống.
* *Giao diện người dùng:* Là màn hình điều khiển, các website hoặc ứng dụng di động có giao diện, cho phép người dùng nhận và gửi thông tin điều khiển đến các thiết bị thông qua máy chủ.

### **Lợi ích của hệ thống Smart Home**

Hệ thống Smart Home đem lại những sự tiện lợi và an toàn cho người dùng về nhiều mặt. Dưới đây là một số sự thuận tiện mà hệ thống Smart Home có thể đem lại:

* Điều khiển, kiểm soát thiết bị trong nhà từ xa thông qua điện thoại thông minh
* Các thiết bị trong nhà có thể tự động hóa
* Điều khiển thiết bị trong nhà thông qua giọng nói
* Đảm bảo an ninh cao, cho phép nhận diện chủ nhân của thiết bị, tránh rủi ro về an ninh.
* Tiết kiệm năng lượng

## **Khảo sát một số thiết bị trong Smart Home**

### **Điều khiển chiếu sáng trong hệ thống Smart Home**

Điều khiển chiếu sáng trong hệ thống smart home là quá trình sử dụng công nghệ để điều chỉnh và kiểm soát các thiết bị chiếu sáng trong ngôi nhà thông minh. Điều này thường được thực hiện thông qua việc sử dụng các thiết bị điều khiển như điện thoại thông minh, máy tính bảng, hoặc giọng nói để tương tác với các hệ thống điều khiển như các ứng dụng hoặc loa thông minh.

Trong hệ thống smart home, người dùng có thể điều chỉnh cường độ sáng, màu sắc, và thậm chí là mẫu đèn sáng thông qua các thiết bị điều khiển từ xa, thậm chí là từ xa qua internet. Điều này mang lại cho người dùng sự linh hoạt và tiện ích trong việc tạo ra môi trường chiếu sáng phù hợp với nhu cầu và sở thích của họ, cũng như trong việc tối ưu hóa tiêu thụ năng lượng.

Hệ thống điều khiển chiếu sáng trong smart home cũng có thể được tích hợp với các công nghệ khác như cảm biến chuyển động hoặc hẹn giờ tự động. Điều này có nghĩa là đèn sẽ tự động bật hoặc tắt khi phát hiện có người đi vào hoặc rời khỏi không gian, hoặc theo các lịch trình đã được đặt trước. Bằng cách kết hợp các tính năng này, người dùng có thể tối ưu hóa không chỉ sự thoải mái và tiện ích mà còn hiệu quả sử dụng năng lượng

Bóng đèn và công tắc thông minh là hai thành phần chính trong hệ thống điều khiển chiếu sáng của nhà thông minh và có mối liên quan chặt chẽ với nhau.

Bóng đèn thông minh và công tắc thông minh là hai phần cơ bản nhưng quan trọng trong hệ thống điều khiển chiếu sáng của nhà thông minh, và sự kết hợp giữa chúng giúp tạo ra một môi trường chiếu sáng linh hoạt, tiện ích và tiết kiệm năng lượng.

### **Khảo sát chi tiết một số loại đèn trong nhà trên thị trường**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên đèn | Vị trí thường đặt | Công suất tiêu thụ | Điện áp | Hệ số công suất | Dòng đỉnh |
| Bóng đèn Led Pulb dạng trụ | Phòng ăn | 60 W | 220VAC | 0.5 | 0.77A |
| Đèn huỳnh quang xoắn ốc | Đèn phòng khách | 135W | 220VAC | 0.6 | 0.87A |
| Đèn ngủ Tesla  SP115 | Phòng ngủ | 10 W | 220V AC |  | 0.065 A |

### **Khảo sát chi tiết một số thiết bị công tắc thông minh thực tế trên thị trường**

#### **Công tắc thông minh Rạng Đông**

Công tắc thông minh là sản phẩm trong hệ thống nhà thông minh RalliSmart của Rạng Đông. Các sản phẩm công tắc thông minh của Rạng Đông bao gồm nhiều loại với các ứng dụng khác nhau: công tắc bình nóng lạnh, công tắc rèm, công tắc đèn, công tắc cảm ứng âm tường, công tắc cảm ứng BLE, công tắc cảm ứng wifi…

A white light switch with green lights

Description automatically generated

Hình 1. 5 Công tắc thông minh BLE 3 nút bấm dáng vuông

**Thông số kĩ thuật:**

* Điện áp nguồn danh định: 220 V/50 Hz
* Công suất tối đa trên nút: 3000W
* Tiêu chuẩn chống bụi, nước: IP44
* Số lần bật tắt trên mỗi nút: 100.000 lần
* Kích thước (Dài x Rộng x Cao):
  + Với công tắc dáng vuông: (86x86x43) mm
  + Với công tắc dáng chữ nhật: (120x72x34) mm
* Chuẩn điều khiển: Wifi hoặc BLE

**Tính năng của sản phẩm:**

* Điều khiển từ xa
* Điều khiển hẹn giờ
* Thiết lập kịch bản
* Độ bền cao
* Dễ dàng cài đặt thông qua app điện thoại của Rạng Đông.

#### **Công tắc thông minh Lumi**

Công tắc thông minh Lumi là công tắc cảm ứng điều khiển từ xa được sử dụng trong hệ sinh thái nhà thông minh Lumi. Công tắc thông minh điều khiển từ xa qua điện thoại cho phép người dùng điều khiển từ xa bằng Smartphone, Tablet qua Internet hoặc Wifi. Thiết bị công tắc điện thông minh với nhiều phiên bản thiết kế (kính phẳng, kính lõm), màu sắc (đen, trắng, viền vàng, viền champagne, viền nhôm), loại hình vuông và chữ nhật. Cách đấu công tắc thông minh được hướng dẫn chi tiết đối với từng mẫu công tắc.



Hình 1. 6 Công tắc Luto Kính phẳng Viền nhôm

A black square with blue lights

Description automatically generated

Hình 1. 7 Công tắc Luto Kính phẳng viền nhôm

**Thông số kĩ thuật:**

* Điện áp hoạt động: 100 – 240VAC ~ 50/60Hz
* Nhiệt độ hoạt động: 0℃ – 50℃
* Truyền thông: ZigBee/BLE Mesh
* Công suất phát Zigbee: 10 dbm
* Công suất tiêu thụ không tải: < 0.5W
* Công suất tải:
  + Tải trở thuần: 3000W/kênh
  + Tải LED: 300W/kênh
  + Tổng công suất tải: <4000W
* Kích thước (D x R x C):
  + Hình chữ nhật: 123 x 78 x 32.5 mm
  + Hình vuông: 95 x 95 x 32.5 mm
* Nguồn: Nguồn cách ly
* Mặt kính tản sáng: Sáng đều
* Phiên bản: Toàn bộ các nút đều là CSC cho bản 1,2,3,4 nút
* Tương thích đế âm đơn simon, đế âm đôi: Có
* Khả năng chống ẩm: Cao

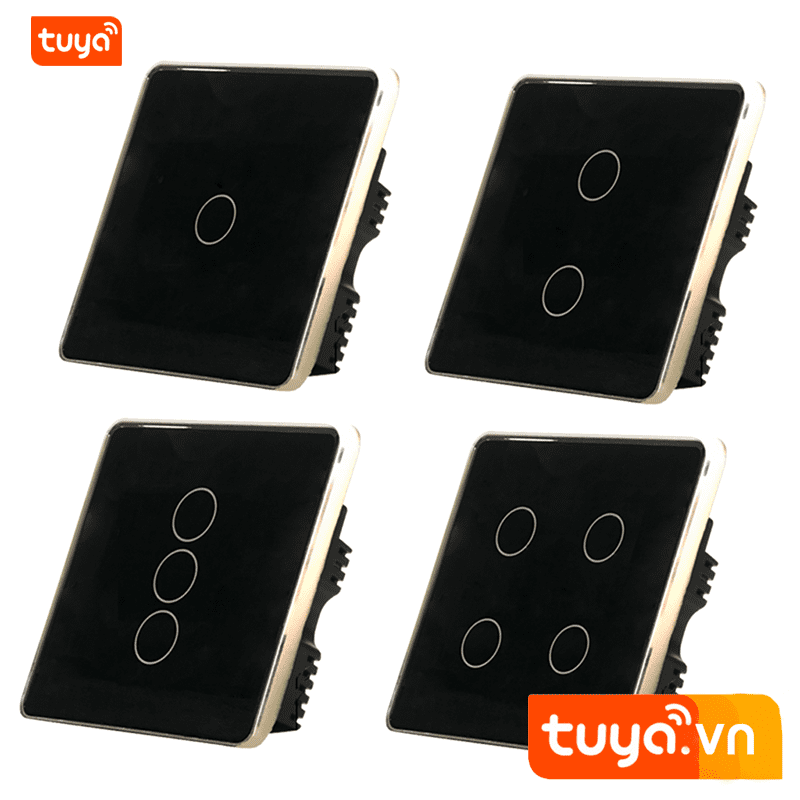
**Tính năng sản phẩm:**

* Điều khiển từ xa thông qua internet/ mạng LAN trong nhà (Qua app Lumi Life+)
* Điều khiển theo nhóm Rule, Cảnh, Lịch (Tùy biến theo nhu cầu người sử dụng).
* Có thể điều khiển thiết bị bằng giọng nói thông qua việc tích hợp Google Assistant, Alexa, Apple Siri
* An toàn, chống cháy nổ

#### **Công tắc thông minh Tuya**

Công tắc cảm ứng wifi có các nút chạm điều khiển cảm ứng được hiển thị bằng dải đèn LED để dễ dàng nhìn thấy trong tối. Có chế độ tự hẹn giờ tắt mở, tự ngắt có thể điều khiển qua mạng, qua internet bằng điện thoại, ipad. Lắp đặt dễ dàng sử dụng đế âm vuông tiêu chuẩn thông dụng tại Việt Nam, cần có dây nguội trong quá trình đấu nối dây.

Ứng dụng: Thay thế các công tắc cơ truyền thống tại các gia đình hiện tại mà không cần đi lại dây điện. Các thiết bị này phù hợp với các tải tiêu thụ như bóng đèn, quạt, tivi...



Hình 1. 8 Công tắc Wifi viền kim loại điều khiển qua app Tuya TUYA-VKL-EU

**Thông số kĩ thuật:**

* Số nút bấm: 1,2,3,4 nút.
* Điện áp: 220V ~ 50Hz.
* Công suất chịu tải: 6A <1000W (tải trở, đèn sợi đốt), < 200W (tải dung, đèn led).
* Wifi: 2.4Ghz IEEE 802.11b/g/n.
* Hỗ trợ Amazon Alexa, Google Home.
* App: Tuya và Smartlife hỗ trợ hệ điều hành: IOS và Android.
* Chất liệu: nhựa, kính cường lực.
* Màu sắc: đen.
* Kích thước: 86x86mm.

**Tính năng sản phẩm:**

* Điều khiển từ xa thông qua internet/ mạng LAN trong nhà (Qua app Tuya)
* Có thể điều khiển thiết bị bằng giọng nói thông qua việc tích hợp Google Assistant, Alexa, Apple Siri

### **Lựa chọn công nghệ mạng không dây**

#### **Khảo sát về công nghệ mạng không dây phổ biến**

Hiện nay Zigbee, Bluetooth Mesh, Wifi và BLE đang là các công nghệ không dây phổ biến. Đây cũng là các chuẩn kết nối đang được các đơn vị nhà thông minh sử dụng tích hợp trong các thiết bị.  Mỗi công nghệ có những ưu và nhược điểm riêng, được chọn lựa dựa trên các yêu cầu về hiệu suất, khoảng cách truyền, tiêu thụ năng lượng, và khả năng tương thích.

Dưới đây là các thông số so sánh giữa các giao thức truyền thông không dây được ứng dụng trong thiết kế Smart Home được mô tả trong Bảng :

Bảng 1 Khảo sát một số công nghệ mạng không dây phổ biến

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Công nghệ** | **Wifi**  **(802.11n)** | **BLE** | **Zigbee** | **Bluetooth Mesh** |
| Băng tần hoạt động | 2.4 GHz  5 GHz | 2.4 GHz | 2.4 GHz | 2.4-2.48 GHz |
| Chuẩn | IEEE 802.11b | IEEE 802.15.4 | IEEE 802.15.4 | IEEE 802.15.4 |
| Công suất tiêu thụ | Cao | Thấp | Thấp | Thấp |
| Khoảng cách thu phát | ~70 m | 10-20 m | 75-100 m | 30-100 m |
| Tốc độ truyền dữ liệu | 54-600 Mbps | < 2Mbps | 250 Kbps | 250 Kbps |
| Tính bảo mật | Thấp | Cao | Cao | Cao |
| Số node có trong mạng | 32/access point | 7-20 | ~64k | 32767 |

## **Lựa chọn công nghệ mạng và lên phương án xây dựng hệ thống**

### **Lựa chọn công nghệ mạng không dây cho hệ thống điều khiển chiếu sáng**

a, Đối với mạng WIFI

Ưu điểm: Tính tương thích rất cao vì hầu hết các gia đình ở Việt Nam đều sử dụng công nghệ này, có thể truyền tải dữ liệu lớn

Nhược điểm: Rất tốn năng lượng và do là mô hình mạng STAR nên sẽ rất tốn kém khi muốn mở rộng mạng. Đối với điều khiển chiếu sáng trong Smart home, các công tắc hay bóng đèn đều là thiết bị tĩnh (không chuyển động), không yêu cầu về tốc độ phục hồi cao nên việc trao đổi dữ liệu giữa các linh kiện sẽ không sử dụng công nghệ mạng WIFI vì điều này sẽ dẫn đến tổn thất nhiều công suất tiêu thụ. Còn hai công nghệ mạng BLE MESH và Zigbee khá phổ biến cho hệ thống điều khiển chiếu sáng.

b, Đối với công nghệ mạng Zigbee

Ưu điểm: Tiết kiệm năng lượng, Tương thích được các thiết bị từ nhà sản xuất theo chuẩn, khoảng cách truyền nhận đủ lớn, dễ dàng mở rộng phạm vi

Nhược điểm: Đối với công nghệ mạng Zigbee, thông thường, các thiết bị điện thoại thông minh, máy tính sẽ thường không hỗ trợ Zigbee, nên trong mạng Zigbee luôn luôn phải có gateway để người dùng có thể kiểm soát trạng thái các thiết bị. Mô hình mạng Zigbee là cây, trong zigbee sẽ có coordinator đóng vai trò kiểm soát sự tham gia của các node, do vậy, coordinator là node duy nhất có thể trao đổi dữ liệu với các node khác, điều này khá bất lợi trong trường hợp coordinator gặp trục trặc. Khi đó dường như người dùng sẽ không thể điều khiển được các node nữa do toàn bộ mạng Zigbee sẽ bị sập.

c, Đối với công nghệ mạng BLE MESH

Đối với công nghệ mạng BLE MESH, hầu hết tất cả các thiết bị thông minh đều hỗ trợ bluetooth và ngoài ra người dùng dễ dàng để kiểm soát các node thông qua bluetooth. Ngoài ra, BLE MESH là mô hình mạng mesh, điều đó có nghĩa mạng vẫn sẽ hoạt động khi một node bị hỏng. Do đó, việc lựa chọn BLE MESH cho điều khiển chiếu sáng hợp lý trong trường hợp này.

Một số ưu điểm mà BLE MESH mang lại:

* Mở rộng Phạm vi: Mạng Mesh cho phép tín hiệu được truyền qua nhiều thiết bị (node), mở rộng phạm vi của mạng lưới
* Tự phục hồi: Mạng có khả năng tự phục hồi, tự động định tuyến lại dữ liệu qua các node khác nếu một node gặp sự cố, giảm thiểu nguy cơ mất mát dữ liệu.
* Hiệu quả Năng lượng: Dù phức tạp hơn BLE truyền thống nhưng Bluetooth Mesh vẫn duy trì được tính năng tiêu thụ năng lượng thấp, nhờ vào các cơ chế quản lý năng lượng thông minh.
* Xét về độ tin cậy thì đây là mạng có độ tin cậy cao, nó không bị gián đoạn kết nối nếu có một thiết bị nào đó không hoạt động.
* Về khả năng bảo mật, Bluetooth Mesh có độ an toàn và bảo mật cao.

Một số nhược điểm:

* Chưa thể sử dụng để truyền các gói dữ liệu lớn, liên tục giống như việc kết nối để phát âm thanh trực tiếp đên loa nghe nhạc hoặc tivi
* Chỉ hỗ trợ các thiết bị trang bị coongn ghệ 4.o trở lên

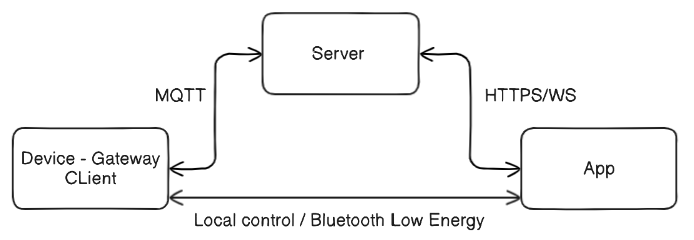
Để có thể điều khiển từ xa, với các thiết bị trong mạng BLE MESH, sẽ có một node làm gateway nhằm phục vụ việc cập nhật dữ liệu lên web/server và điều khiển từ xa.

### **Phương án xây dựng hệ thống điều khiển chiếu sáng thông minh cho Smart Home**

#### **Thiết bị**

Đối với BLE MESH, mô hình server/client được sử dụng. Trong đó thiết bị đóng vai trò client là thiết bị gửi lệnh điều khiển, thiết bị đóng vai trò server là thiết bị nhận lệnh điều khiển.

Thiết bị gửi lệnh điều khiển sẽ đóng vai trò gateway. Thiết bị được kết nối WIFI để giao tiếp với app và server, đồng thời nằm trong mạng BLE MESH để thực hiện việc điều khiển các thiết bị khác.



Hình 1. 9 Mô hình tổng quan thiết bị đóng vai trò gateway

Thiết bị gateway này sẽ giao tiếp với server và app, nhận lệnh từ web server và điều khiển các thiết bị khác (server) trong nhà.

A diagram of a device

Description automatically generated

Hình 1. 10 Mô hình BLE MESH

Device sẽ là bóng đèn hoặc công tắc miễn là ở gần WiFi router, ở trên device sẽ có nút gạt chuyển đổi linh hoạt client và server.

Đối với device là công tắc, sẽ được thiết kế âm tường, kết nối thẳng với nguồn điện 220V 50HZ, sẽ được phân ra công tắc đóng vai trò client hoặc server dựa trên thanh gạt chọn chế độ được thiết kế ở công tắc, có nút thực hiện RESET nút bấm trở về trạng thái khi chưa thực hiện kết nối WiFi , BLE MESH giúp việc thay đổi linh hoạt.

Đối với công tắc đóng vai trò server sẽ nhận lệnh điều khiển từ công tắc đóng vai trò client. Công tắc đóng vai trò server sẽ kết nối với loại bóng đèn bình thường trong nhà, điều đó có nghĩa khi muốn bật tắt một bóng đèn thì có thể bật tắt như công tắc bình thường hoặc bật tắt từ xa thông qua BLE MESH. Mọi hành động, các chức năng phát triển cho công tắc server như hẹn giờ tắt bóng đèn,… sẽ được điều khiển hoàn toàn bằng client.

Đối với công tắc đóng vai trò client cũng sẽ được kết nối âm tường, kết nối chung với dây đi cùng bóng đèn thông thường để có thể điều khiển như bình thường. Công tắc này được kết nối với wifi để giao tiếp với web server để cập nhật trạng thái các bóng đèn lên server hay nhận lệnh để thực hiện các chức năng mà trên web/server hiển thị.



Hình 1. 11 Bóng đèn đổi màu theo ngữ cảnh

Đối với device là bóng đèn (thông minh), bóng đèn được thiết kế chỉ cần lắp thẳng vào đui bóng đèn, có chế độ thay đổi màu sắc bóng đèn từ web,app , điều khiển bật tắt từ xa,hẹn giờ hoặc bóng đèn có thể thiết lập chế độ thay đổi màu sắc dựa trên ngữ cảnh xung quanh, ví dụ nếu nhiệt độ trên 25 độ C, bóng đèn chuyển thành màu mát, nếu nhiệt độ nhỏ hơn 15 độ C, bóng đèn chuyển màu ấm hơn giúp người dùng có cảm giác dễ chịu.

#### **Xây dựng WEB SERVER**

Mục đích của web server được tạo ra trong hệ thống này để người dùng có thể sử dụng các thiết bị thông minh như điện thoại, laptop để quản lý các thiết bị trong nhà một cách dễ dàng, thiết lập các chức năng tiện ích cho tiết bị, theo dõi thiết bị, điều khiển từ xa.

Một số tính năng có trên web/server như:

* Người dùng có thể đăng ký, đăng nhập, đổi mật khẩu, quên mật khẩu
* Điều khiển thiết bị, lập lịch hoạt động cho thiết bị, xem trạng thái thiết bị
* Đăng ký thiết bị mới

Tài liệu tham khảo:

<https://rangdong.com.vn/bong-led-bulb-rgb-dieu-khien-bang-bluetooth-pr1772.html>

<https://smarthomekit.vn/review-den-yeelight-1s-gen-3/>

<https://rangdong.com.vn/cong-tac-2-chieu-khong-day-3-nut-pr2715.html>

<https://fptsmarthome.vn/kien-thuc/bluetooth-mesh-la-gi>

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **Công nghệ Bluetooth Low Energy MESH**

### **Lịch sử phát triển công nghệ BLE**

Công nghệ BLE (Bluetooth Low Energy) là một giao thức truyền thông không dây được thiết kế để tiết kiệm năng lượng, dành cho việc kết nối các thiết bị di động như điện thoại thông minh, đồng hồ thông minh, hoặc cảm biến với các thiết bị khác mà không cần một lượng lớn năng lượng. Dưới đây là một cái nhìn tổng quan về lịch sử phát triển công nghệ BLE:

Nền tảng Ban đầu (2006-2010): Công nghệ BLE được phát triển ban đầu bởi công ty Nokia với tên gọi Wibree. Mục tiêu của Wibree là cung cấp một giao thức không dây tiết kiệm năng lượng cho các thiết bị di động. Trong thời gian này, Nokia đã hợp nhất Wibree vào Bluetooth SIG (Bluetooth Special Interest Group) và đổi tên thành Bluetooth Low Energy.

Phiên bản Bluetooth 4.0 (2010): Công nghệ BLE chính thức được giới thiệu trong phiên bản Bluetooth 4.0. Việc này mở ra cánh cửa cho việc tích hợp BLE vào nhiều loại thiết bị di động, từ điện thoại thông minh đến các thiết bị IoT (Internet of Things).

Sự Phát triển và Mở Rộng (2010-2016): BLE đã trở thành một phần quan trọng của nhiều ứng dụng di động và IoT. Công nghệ này được tích hợp rộng rãi vào các thiết bị như đồng hồ thông minh, cảm biến sức khỏe, thiết bị định vị, và nhiều thiết bị khác.

Bluetooth 5.0 (2016): Phiên bản Bluetooth 5.0 đã được công bố vào năm 2016, đồng thời cung cấp nâng cấp đáng kể cho BLE. Bluetooth 5.0 tăng gấp đôi tốc độ truyền dữ liệu, tăng khoảng cách hoạt động và cải thiện hiệu suất năng lượng.

Sự Lan Rộng Trong Các Ứng Dụng (2016-nay): BLE đã trở thành một phần không thể thiếu trong việc kết nối và tương tác giữa các thiết bị di động, đặc biệt là trong lĩnh vực sức khỏe và thể thao, IoT, công nghiệp thông minh và gia đình thông minh.

Tiêu Chuẩn Liên Kết (2020-nay): Công nghệ BLE tiếp tục được cải tiến và phát triển thông qua việc ra mắt các tiêu chuẩn mới, cũng như việc tích hợp với các công nghệ khác như Bluetooth Mesh để mở rộng ứng dụng của nó trong các môi trường mạng lưới lớn.

BLE được phát triển theo chuẩn 802.15.4, là chuẩn mạng truyền thông nhiều lợi thế , giải quyết nhiều vấn đề lớn của mạng cảm biến không dây bao gồm:

* Vùng phủ sóng lớn
* Khả năng quản lý và kiểm soát một số lượng thiết bị lớn
* Cho phép mở rộng mạng
* Tương thích với hầu hết các thiết bị thông minh ngày nay
* Tính bảo mật cao
* Tiết kiệm năng lượng so với Bluetooth cổ điển

**Các layer của Bluetooth Low energy**

#### **Định nghĩa cơ bản các layer**

A screenshot of a computer

Description automatically generated

Hình 2. 1 Các layer của Bluetooth Low Energy

**Controller:** Là phần cứng trực tiếp **thực hiện** các hoạt động **vật lý của kết nối Bluetooth**, bao gồm việc **tạo** và duy trì kết nối, **xử lý sóng radio và quản lý gói tin truyền thông**.

**Host**: Là **lớp phần mềm** chịu trách nhiệm về **quản lý và điều khiển các hoạt động cụ thể của thiết bị Bluetooth**. Host thường chạy trên một vi điều khiển hoặc một thiết bị điện tử khác, thực hiện các tác vụ như quản lý giao thức, xử lý dữ liệu và tương tác với người dùng.

**Host Controller Interface:** Là giao diện để host và controller giao tiếp. HCI định nghĩa lệnh , làm mọi nhiệm vụ liên quan khởi động, kết nối. HCI hiểu đơn giản là một quy chuẩn để cho thấy rằng host và controller muốn giao tiếp được với nhau thì phải theo đúng chuẩn của HCI.

Trong BLE có các lớp gồm PHY layer, Link Layer, IAL layer, L2CAP layer, ATT, SMP layer, GATT layer và GAP layer.

Bảng dưới đây giải thích cơ bản về các lớp:

Bảng 2. 1 Chức năng của các layer trong BLE

|  |  |
| --- | --- |
| Tên các layer | Chức năng chính |
| GAP – Generic Access Profile | Lớp GAP chịu trách nhiệm quản lý quá trình kết nối, công việc, tính năng, bảo mật và quảng bá của các thiết bị BLE để đảm bảo hiệu suất và tính ổn định của mạng BLE |
| GATT – Generic Attribute Profile | Lớp GATT là một phần quan trọng của cơ sở hạ tầng BLE, quản lý việc truy cập và tương tác với dữ liệu giữa các thiết bị BLE |
| ATT – Atribute Protocol | Quản lý thuộc tính của dữ liệu, thực hiện giao tiếp giữa hai thiết bị theo mô hình server và client |
| SMP – Securiry manager protocol | Quản lý bảo mật trong quá trình kết nối và truyền thông |
| L2CAP | Quản lý các kênh giao tiếp, phân mảnh dữ liệu thành các packet phù hợp để truyền xuống lớp dưới thông qua HCI |
| Isochronous Adaptation layer | Quản lý truyền dữ liệu trong thời gian thực |
| Link Layer | Xác định cách thức truyền dữ liệu thông qua các gói tin, có cơ chế kiểm soát lỗi để toàn vẹn dữ liệu |
| Physical layer | Thực hiện truyền dữ liệu |

#### **Atribute Protocol – ATT**

ATT là lớp được coi là một phương pháp tiêu chuẩn để lưu trữ và tổ chức dữ liệu, có cơ chế để truy cập dữ liệu. Dưới đây là hĩnh ảnh mô tả cấu trúc dữ liệu của ATT



Hình 2. 2 Cấu trúc dữ liệu ATT

Ở trên đã mô tả thứ tự các thành phần trong cấu trúc dữ liệu bao gồm:

* **Handle** : 16 bit identifier là cái để client có thể tham chiếu thuộc tính trên server . Nó làm cho thuộc tính có thể định địa chủ và không thay đổi trong mỗi kết nối .
* **Type ( UUID):** Unique ( 2 byte – 16 byte) UUID identifier để định nghĩa kiểu và loại dữ liệu. Có hai (types) kiểu là Service UUID và Characteristic UUDI.
* **Value**: Dữ liệu
* **Permission Field**: Cấp độ bảo mật.

Bluetooth Mesh sẽ được phân ra theo mô hình client, server, trong đó client đóng vai trò như central. Ví dụ như điện thoại thông minh, còn server là các thiết bị như vi điều khiển,… Trong server, dữ liệu sẽ được tổ chức trong bảng như sau:

A blue and black text on a blue background

Description automatically generated

Hình 2. 3 Tổ chức dữ liệu ATT trong server

Để có thể đọc hoặc viết lên các dữ liệu trong cơ sở dữ liệu, server sẽ cũng cấp cho client một số phương thức bao gồm: Có thể đọc, có thể viết, vừa đọc vừa viết hoặc không được phép truy cập.

A diagram of a process

Description automatically generated

Hình 2. 4 Phương pháp truy cập dữ liệu theo mô hình client – server

#### **GATT - Generic Attribute Profile**

GATT Là lớp phân cấp cơ sở dữ liệu:

* Nút gốc được đặt tên là profile. (cấp 0)
* Con của Profile được đặt tên service. (cấp độ 1)
* Con của service được đặt tên characteristic. (cấp độ 2)
* Các characteristic có thể được xác định bởi một giá trị duy nhất (cấp độ 3)

A diagram of a service

Description automatically generated

Hình 2. 5 GATT - Phân cấp cơ sở dữ liệu ATT

A screenshot of a computer

Description automatically generated

Hình 2. 6 Ảnh thể hiện mối liên hệ giữa GATT và ATT

<https://novelbits.io/bluetooth-le-att-gatt-explained-connection-oriented-communication/?fbclid=IwAR3JyAe_OHPq4fypRQ0fZTwFW3E_jrF4BfqUPJzpvu0xIo5WHmJITNkxyvs>

### **Công nghệ BLE MESH**

#### **Giới thiệu về BLE MESH**

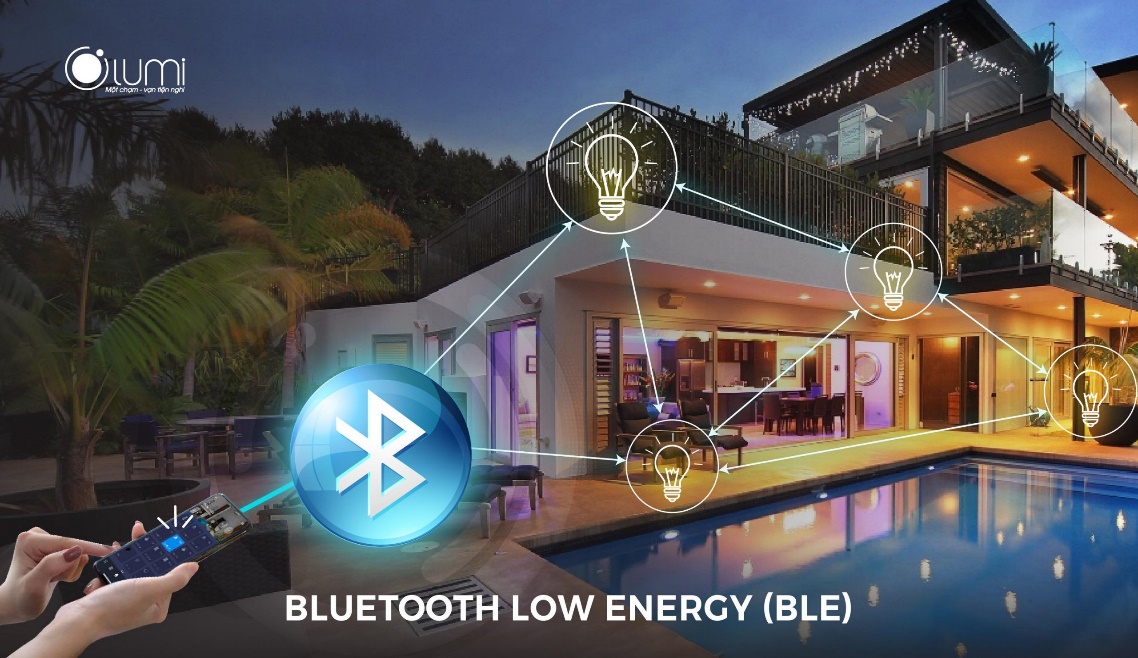
Công nghệ BLE Mesh(BLE) cho phép tạo mạng lưới kết nối bluetooth giữa các thiết bị, tạo thành một hệ thống mạng quy mô lớn. Dữ liệu được truyền từ thiết bị này sang thiết bị khác mà không cần thông qua trung gian, giúp tối ưu hoá quá trình giao tiếp.

Bluetooth Low Energy (BLE) được xem là phiên bản Bluetooth mang tính bước ngoặt trong lịch sử phát triển của Bluetooth. Được thiết kế để vận hành các thiết bị sử dụng mức năng lượng thấp, Bluetooth giới thiệu tính năng “Tiết kiệm năng lượng” và những chức năng liên quan.

Những đặc điểm này có nghĩa là BLE (mạng Bluetooth tiết kiệm năng lượng) có thể tích hợp vào nhiều thiết bị IoT cho nhiều mục đích sử dụng (ví dụ như cảm biến chuyển động chạy bằng pin).

Để kích hoạt hoạt động trong dải tần 2.4GHz, nó sử dụng phương pháp trải phổ nhảy tần truyền dữ liệu trên 40 kênh.

Bên cạnh đó, sóng radio của BLE cung cấp cho các nhà phát triển rất nhiều sự linh hoạt, bao gồm nhiều tùy chọn PHY hỗ trợ tốc độ dữ liệu từ 125 Kb/s đến 2 Mb/s, nhiều mức công suất, từ 1mW đến 100 mW, cũng như nhiều tùy chọn bảo mật khác.



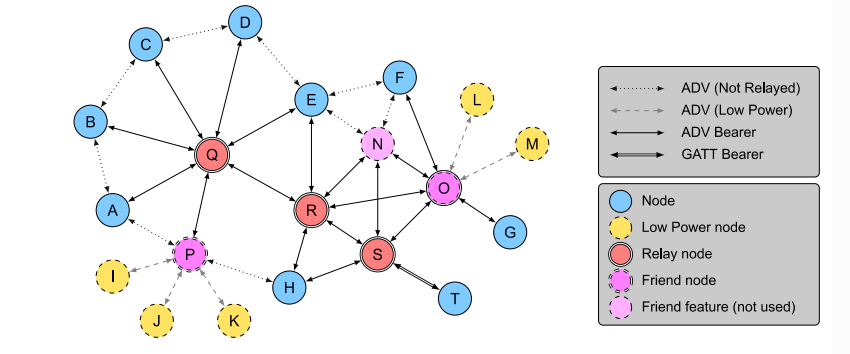
Hình 2. 7 Công nghệ mạng BLE MESH

#### **Các thuật ngữ trong BLE MESH**

Bảng 2. 2 Thuật ngữ trong BLE MESH

|  |  |
| --- | --- |
| Thuật ngữ | Định nghĩa |
| Address | Nhận dạng một hay nhiều phẩn từ trong một hoặc nhiều node |
| Configuration client | Node đóng vai trò client model |
| Destination | Địa chỉ đích nơi mà message được gửi đến |
| Element | Một phần tử được định nghĩa bởi một address, một thiết bị bắt buộc phải có một phần tử |
| Message | Các octet (các byte) được gửi từ source đến destination |
| Neighbors | Single hop ( node ở trong phạm vi sóng ) |
| Network | Một nhóm node liên kết với nhau trong mạng |
| Node | Là thiết bị đã được provisioned |
| Provision | Quá trình xác thực và cung cấp thông tin cơ bản (bao gồm địa chỉ unicast và một khóa mạng - network key) cho một thiết bị. Một thiết bị phải được cung cấp thông tin để trở thành một nút trong mạng. Sau khi đã được cung cấp thông tin, một nút có thể truyền hoặc nhận các tin nhắn trong một mạng lưới mesh. |
| Provisioner | Là node có khả năng thêm node khác vào trong mạng |
| Relay | Một node có khả năng chuyển tiếp message |
| Souces | Là address nơi mà tin nhắn được gửi đi |
| State | Giá trị thể hiện trạng thái của một element |
| Subnet | Một nhóm node có thể giao tiếp với nhau |
| Relay feature | Một node có đặc điểm này có khả năng chuyển message thông qua advertising bearer để mở rộng mạng |
| Proxy feature | Có khả năng truyền nhận message thông qua GATT và advertising bearer |
| Low power feature | Khả năng hoạt động trong mạng lưới ở mức giảm đáng kể chu kỳ nhiệm vụ chỉ kết hợp với nút hỗ trợ friend feature |
| Friend feature | Có khả năng giúp node hỗ trợ các node đang chạy low power feature, nó có khả năng lưu giữ message cho các low power node. |

**Cấu trúc liên kết của các loại node trong mạng BLE MESH:**

****

Hình 2. 8 Sự liên kết trong mạng MESH

Có ba nút Relay: Q, R, và S. Ba nút hỗ trợ tính năng Friend là N, O, và P, tuy nhiên N không có bất kỳ mối quan hệ Friend nào; do đó chỉ có O và P là các nút Friend. Có năm nút Low Power: I, J, K, L, và M. Các nút I, J, và K có P làm bạn của họ, trong khi L và M có O làm bạn của họ. Nút T chỉ được kết nối vào mạng lưới mesh bằng cách sử dụng bearer GATT; do đó S phải relay tất cả các tin nhắn tới và từ T.

Ví dụ, nếu một tin nhắn được gửi từ T đến L, thì T sẽ gửi tin nhắn đến nút S bằng cách sử dụng bearer GATT. Nút S sẽ chuyển tiếp lại tin nhắn này bằng cách sử dụng bearer quảng cáo. Các nút H, R, N, và O nằm trong phạm vi sóng radio của nút S; do đó họ sẽ nhận được tin nhắn này. Nút O, là bạn của nút L, sẽ lưu trữ tin nhắn này, và nếu tin nhắn là một tin nhắn được phân đoạn, nút O sẽ phản hồi bằng một sự xác nhận ở tầng vận chuyển thấp hơn. Một thời gian sau, L sẽ hỏi nút O để kiểm tra tin nhắn mới, để O sẽ chuyển tiếp tin nhắn được gửi ban đầu bởi T đến L.

#### **Model**

Trong mạng lưới BLE Mesh, mô hình client và server được sử dụng để mô tả cách mà các thiết bị trong mạng tương tác với nhau. Dưới đây là chi tiết về mô hình client và server:

**Mô hình Client:**

Trong mô hình client, một thiết bị được gọi là client khi nó gửi yêu cầu hoặc lệnh đến một thiết bị khác để yêu cầu hoặc thực hiện một hành động cụ thể.

Client thường gửi các yêu cầu hoặc lệnh đến các thiết bị khác, được gọi là server, để thực hiện các tác vụ như kiểm soát thiết bị, truy vấn trạng thái, hoặc thu thập dữ liệu.

Ví dụ về mô hình client có thể là một điều khiển từ xa điều khiển ánh sáng trong phòng bằng cách gửi yêu cầu bật hoặc tắt đến một đèn LED trong mạng.

**Mô hình Server:**

Trong mô hình server, một thiết bị được gọi là server khi nó lắng nghe và xử lý các yêu cầu hoặc lệnh được gửi từ các thiết bị khác, tức là từ các client.

Server thực hiện các hành động cụ thể dựa trên các yêu cầu hoặc lệnh nhận được, chẳng hạn như bật hoặc tắt đèn, cập nhật trạng thái, hoặc gửi thông tin cảm biến.

Một thiết bị có thể hoạt động như server cho một hoặc nhiều dịch vụ, và có thể phản hồi với các phản hồi hoặc dữ liệu cần thiết.

A diagram of a state

Description automatically generated

Hình 2. 9 Sự giao tiếp giữa Client Model và Server Model

Hình ảnh này mô tả cấu trúc mô hình phần tử cho một thiết bị thực hiện mô hình server (Thiết bị C) với một trạng thái và các tin nhắn hỗ trợ R, S, T, X, Y, Z; và hai thiết bị thực hiện mô hình client, với Thiết bị A hỗ trợ các tin nhắn X, Y, và Z và Thiết bị B hỗ trợ các tin nhắn R, S, T, và Z.

Các Model có thể định nghĩa các chức năng của một thiết bị như một nút mạng, chẳng hạn như quản lý network key, application key , gán địa chỉ và relay các tin nhắn. Các Model cũng định nghĩa các hành vi của một thiết bị xây dựng xung quanh một nút mạng, chẳng hạn như kiểm soát điện nănn, kiểm soát ánh sáng và thu thập dữ liệu cảm biến, điều khiển chiếu sáng. Có thể có các nút thực hiện chỉ các chức năng liên quan đến mạng, như các nút Relay hoặc Proxy, trong khi phần lớn các nút có thể tương tác với thế giới vật lý bằng cách kiểm soát điện năng, kiểm soát phát sáng hoặc cảm biến dữ liệu môi trường.

Một message có thể được sử dụng bởi nhiều Model khác nhau. Hành vi của tin nhắn là như nhau trong mỗi model, giúp tạo ra một sự hiểu biết chung giữa các Model client, server và điều khiển vì hành vi là nhất quán bất kể Model nào gửi và xử lý tin nhắn.

#### **Giới thiệu về Publish, Subscribe và security trong BLE MESH**

Trong BLE Mesh, cơ chế Publish-Subscribe được sử dụng để giao tiếp giữa các thiết bị, cùng với các cơ chế bảo mật như NetKey và AppKey để bảo vệ dữ liệu.

**Publish-Subscribe trong BLE Mesh**

Trong mô hình Publish-Subscribe, các thiết bị được phân thành hai loại chính: Publisher (người xuất bản) và Subscriber (người đăng ký). Publisher là những thiết bị gửi dữ liệu, còn Subscriber là những thiết bị nhận dữ liệu. Một thiết bị có thể hoạt động như cả Publisher và Subscriber đồng thời.

Khi một thiết bị Publisher có dữ liệu mới muốn chia sẻ, nó sẽ gửi thông điệp (message) theo một hoặc nhiều địa chỉ mục tiêu (có thể là một hoặc nhiều Subscriber). Các thiết bị Subscriber sẽ nhận thông điệp này và xử lý theo nó. Điều này cho phép việc truyền thông dữ liệu từ một nguồn đến nhiều đích trong mạng BLE Mesh.

**Security trong BLE Mesh**

**NetKey:**

NetKey (Network Key) là một khóa dùng để mã hóa và giải mã thông điệp trao đổi giữa các thiết bị trong mạng. Nó đảm bảo tính bảo mật của dữ liệu được truyền trong mạng Mesh. NetKey được chia sẻ giữa tất cả các thiết bị trong cùng một mạng Mesh, đảm bảo rằng tất cả các thiết bị đều có thể mã hóa và giải mã dữ liệu mà họ trao đổi với nhau.

**AppKey:**

AppKey (Application Key) là một khóa được sử dụng để mã hóa và giải mã dữ liệu ở mức ứng dụng. Mỗi ứng dụng trong mạng Mesh có thể sử dụng một hoặc nhiều AppKey để đảm bảo tính bảo mật cho dữ liệu của riêng nó. Khác với NetKey, AppKey không được chia sẻ rộng rãi và thường chỉ được biết đến bởi các thiết bị hoặc ứng dụng cụ thể trong mạng.

#### **Kiến trúc mạng BLE MESH**

Bluetooth MESH 8 tầng .Các tầng theo thứ tự từ thấp nhất là Bluetooth Core Specification, Bearer, Network, Lower Transport, Upper Transport, Access, Foundation Model, Model.

A screenshot of a computer

Description automatically generated

Hình 2. 10 Kiến trúc mạng BLE MESH

##### **Bearer Layer**

Đây là tầng định nghĩa cách mà cách bản tin được truyền giữa các node, chúng được phân ra thành GATT bearer và advertising bearer. Cần có bearer để các node có thể biết đến nhau, nắm bắt được message chuyển tới từ các node khác nhau.

**Advertising bearer:**

Khi mà sử dụng adv bearer, các packet của mạng mesh này sẽ gửi các các bản tin quảng cáo được định nghĩa theo dạng AD. Trong quá trình quảng cáo (advertising) của BLE, các thiết bị có thể gửi dữ liệu quảng cáo thông qua gói tin lưới (mesh packet) bằng cách đặt nó trong dữ liệu quảng cáo của một PDU quảng cáo. Điều này cho phép các thiết bị BLE tham gia vào mạng lưới mạnh mẽ và linh hoạt hơn, với khả năng gửi và nhận dữ liệu không chỉ thông qua kết nối điểm-điểm mà còn thông qua một mạng lưới. Khi sử dụng bearer này phải sử dụng chu kỳ quét khá là lớn để không bị bỏ lỡ bản tin. Độ dài của message beaer này là 8 bit như trong hình 2.3.



Hình 2. 11 Mesh message với AD type

**GATT Layer:**

Trong mạng BLE Mesh, GATT (Generic Attribute Profile) bearer là một trong các bearer (phương tiện truyền thông) được sử dụng để truyền thông tin giữa các thiết bị trong mạng lưới. GATT bearer cho phép truyền dữ liệu thông qua các dịch vụ và thuộc tính được xác định trong GATT. GATT bearer được sử dụng khi các thiết bị cần giao tiếp thông qua các dịch vụ và thuộc tính GATT. Điều này có thể xảy ra khi các thiết bị BLE Mesh cần truyền dữ liệu như cập nhật trạng thái, lấy thông tin từ các thiết bị khác hoặc thực hiện các tác vụ khác dựa trên giao thức GATT.

GATT bearer thường được dùng cho các thiết bị không hỗ trợ adv bearer trong mạng mesh. GATT bearer sử dụng các characteristic để đọc và viết các thông báo từ bản tin sử dụng giao thức ATT. Nó được phân loại thành GATT server và ATT client. Thường thì GATT bearer chỉ support cho dịch vụ mesh Proxy.

Trong mạng BLE Mesh, một thiết bị Proxy là một thiết bị có khả năng giao tiếp với thiết bị ngoài mạng lưới Mesh thông qua GATT (Generic Attribute Profile). Nói cách khác, thiết bị Proxy đóng vai trò là cầu nối giữa mạng lưới Mesh và các thiết bị ngoài Mesh, cho phép truy cập vào mạng lưới Mesh thông qua giao thức GATT. Hiểu đơn giản nhờ vào GATT bearer các thiết bị bên ngoài như smartphone, … có thể dễ dàng gửi dữ liệu cho mạng BLE MESH thông qua giao thức ATT đã được đề cập ở phần lý thuyết bên trên.

Việc sử dụng Proxy trong mạng BLE Mesh có thể phù hợp với GATT bearer vì các thiết bị ngoài mạng lưới Mesh thường không thể truy cập trực tiếp vào mạng lưới Mesh bằng các phương thức truyền thông như Advertising bearer. Thay vào đó, chúng cần phải giao tiếp thông qua GATT để truy cập vào dữ liệu và chức năng trong mạng lưới Mesh.

##### **Network Layer**

Tầng mạng trong mạng lưới Bluetooth Mesh có nhiệm vụ quan trọng là xác định cách dữ liệu được đóng gói và chuyển tiếp qua mạng. Ở tầng này, dữ liệu từ các tầng dưới được "gói" vào các gói tin gọi là Network PDUs, để có thể được gửi qua mạng và đến các thiết bị khác.

Khi một thiết bị nhận được một tin nhắn, tầng mạng sẽ giải mã và kiểm tra tính hợp lệ của nó, đảm bảo rằng tin nhắn đến từ nguồn tin cậy và không bị thay đổi. Sau đó, tin nhắn sẽ được chuyển tiếp lên các tầng khác để xử lý hoặc đến các thiết bị khác trong mạng.

Khi một thiết bị muốn gửi một tin nhắn đi, tầng mạng sẽ mã hóa và xác thực tin nhắn để đảm bảo tính bảo mật. Sau đó, tin nhắn sẽ được chuyển tiếp đến các thiết bị khác trong mạng.

###### **Addresses**

Được định nghĩa 4 kiểu address bao gồm : Unassigned address, Unicast address, Virtual Address và Group Address. (16 bit)

Bảng 2. 3 Phân bổ 16 bit địa chỉ

|  |  |
| --- | --- |
| Giá trị | Kiểu địa chỉ |
| 0b0000000000000000 | Unassigned Address |
| b0xxxxxxxxxxxxxxx (trừ cái trên) | Unicast address |
| 0b10xxxxxxxxxxxxxx | Virtual Address |
| 0b11xxxxxxxxxxxxxx | Group Address |

**Unassigned Address:**

Một địa chỉ chưa được gán là một địa chỉ mà phần tử của một nút chưa được cấu hình hoặc không có địa chỉ nào đã được phân bổ. Địa chỉ chưa được gán sẽ có giá trị là 0x0000 như hình 2.10.

A close-up of a number

Description automatically generated

Hình 2. 12 Khung địa chỉ kiểu Unassigned

**Unicast Address:**

Một địa đơn nhất (unicast address) là một địa chỉ duy nhất được cấp cho mỗi phần tử. Một địa chỉ đơn phát có bit 15 được đặt thành 0. Địa chỉ đơn nhất không được phép có giá trị là 0x0000, và do đó có thể có bất kỳ giá trị nào từ 0x0001 đến 0x7FFF. Một địa chỉ đơn phát được cấp cho mỗi phần tử của một nút trong suốt thời gian tồn tại của nút trên mạng bởi một Provisioner trong quá trình provisioning.

A white rectangular sign with black text

Description automatically generated

Hình 2. 13 Khung địa chỉ kiểu Unicast

**Virtual Address:**

Trong mạng lưới BLE Mesh, địa chỉ ảo (Virtual Address) là một khái niệm được sử dụng để đại diện cho các nhóm thiết bị hoặc các tập hợp các thiết bị trong mạng, thay vì chỉ một thiết bị cụ thể như trong trường hợp của địa chỉ unicast. Địa chỉ ảo không tương ứng với một thiết bị cụ thể mà thường được sử dụng để định rõ các nhóm thiết bị hoặc các thiết bị tham gia vào các tác vụ cụ thể trong mạng.

Việc sử dụng địa chỉ ảo cho phép các thiết bị trong mạng thực hiện tương tác với nhau dựa trên các nhóm thiết bị hoặc các tập hợp cụ thể thay vì chỉ giao tiếp với các thiết bị cụ thể duy nhất. Điều này hữu ích trong việc quản lý và điều khiển mạng lưới BLE Mesh, cho phép các ứng dụng và kịch bản phức tạp hơn như điều khiển đèn trong một phòng hoặc quản lý nhóm thiết bị trong một khu vực cụ thể.

Mỗi địa chỉ ảo logic đại diện cho một nhãn UUID, là một giá trị 128-bit không cần phải được quản lý tập trung. Một hoặc nhiều phần tử có thể được cấu hình để xuất bản hoặc đăng ký một nhãn UUID. Địa chỉ ảo là một giá trị 16 bit có bit thứ 15 được thiết lập thành 1, bit thứ 14 được thiết lập thành 0, và các bit từ 13 đến 0 được thiết lập thành giá trị của một hàm băm (hash).

Các nhãn UUID có thể được tạo ngẫu nhiên theo định nghĩa. Một Configuration Client có thể gán và theo dõi các địa chỉ ảo, tuy nhiên hai thiết bị cũng có thể tạo ra một địa chỉ ảo bằng cách sử dụng một cơ chế ngoại vi (OOB). Khác với địa chỉ nhóm (group address), những địa chỉ này có thể được các thiết bị liên quan đồng ý và không cần phải đăng ký trong cơ sở dữ liệu cấu hình tập trung, vì khả năng trùng lặp của chúng thấp. Dải địa chỉ ảo nằm từ 0x8000 cho tới 0xBFFF.

A white rectangular sign with black text

Description automatically generated

Hình 2. 14 Khung địa chỉ ảo

**Group Address:**

Một địa chỉ nhóm là một địa chỉ được lập trình vào không hoặc nhiều phần tử. Một địa chỉ nhóm có bit thứ 15 được thiết lập thành 1 và bit thứ 14 được thiết lập thành 1, như được hiển thị trong Các địa chỉ nhóm trong khoảng từ 0xFF00 đến 0xFFFF được dành cho các địa chỉ nhóm cố định và các địa chỉ trong khoảng từ 0xC000 đến 0xFEFF thông thường được sử dụng cho các mục đích khác.

A white rectangular sign with black text

Description automatically generated

Hình 2. 15 Khung của group address

* Dưới đây là bảng Group Address cố định:

|  |  |
| --- | --- |
| Giá trị | Địa chỉ group address cố định |
| 0xFFFC | Tất cả proxy |
| 0xFFFD | Tất cả Friend |
| 0xFFFE | Tất cả Relay |
| 0xFFFF | Toàn bộ các node trong mạng |

###### **Network PDU (Protocol Data Unit)**

Trong BLE Mesh, dữ liệu được chuyển đi thông qua các gói dữ liệu được gọi là "PDU" (Protocol Data Unit). Các PDU này được sử dụng để truyền các lệnh, tin nhắn và dữ liệu khác giữa các thiết bị trong mạng Mesh.

Bảng dưới đây minh họa các trường của PDU Network:

Bảng 2. 4 Định nghĩa các trường trong Network PDU

|  |  |  |
| --- | --- | --- |
| Tên trường | Số bit | Chức năng |
| IVI | 1 | Là bit nhỏ nhất trong IV index, là thứ dùng để xác thực và mã hóa Network PDU. |
| NID | 7 | Là giá trị bắt nguồn từ network key dùng để xác thực các khóa bảo mật, dùng để mã hóa CPU. |
| CTL | 1 | Xác định loại bản tin   * CTL = 0, Access PDU là bản tin data * CTL = 1, Access PDU là bản tin command |
| TTL | 7 | (Time to Live)  Sau mỗi lần message được chuyển, giá trị TTL giảm đi 1.  Các giá trị của nó đại diện cho số lần message có thể được chuyển đi trong mạng MESH. |
| SEQ | 24 | Là một giá trị độc nhất cho mỗi Network PDU |
| SRC | 16 | ( source address) là giá trị nguồn của element |
| DST | 16 | (Destination address) là giá trị đích mà một PDU muốn truyền tới (unicast address, virtual address ,…) |
| Transport PDU | 8 - 128 | Số đơn vị Dât vận chuyển |
| NetMic | 32 hoặc 64 bit | Kiểm tra tính toàn vẹn của bản tin |

Hình ảnh dưới đây mô tả khung của một Network PDU:

A close-up of a document

Description automatically generated

Hình 2. 16 Khung Network PDU

**Nhận Network PDU:**

Các message được chuyển từ bearer Layer tới Network Layer thông qua network interface. Trước khi tới lớp Network Layer, message sẽ có một sự lọc đầu vào theo một số luật của nó. Nếu qua được bước lọc này, message sẽ chính thức được chuyển tới network layer.

Khi nhận Network PDU, node sẽ kiểm tra trường NID có đúng với NID mà node đã biết trước đó không. Nếu không đúng thì message tự động bị phớt lờ. Nếu trường này đúng, node sẽ xác thực MIC xem bản tin gửi đến có vấn đề gì không. Nếu bản tin gửi đến bị sai, thiếu octet, bản tin bị phớt lờ. Sau khi thỏa mãn NID và MIC, node kiểm tra địa chỉ nguồn và đích, nếu thỏa mãn, node kiểm tra **network message cache**.

Trong BLE Mesh, Network Message Cache là một cơ chế được sử dụng để giảm thiểu việc thực hiện các kiểm tra bảo mật không cần thiết và việc truyền dẫn thông điệp mạng qua nhiều nút mạng một cách quá mức. Điều này giúp tăng hiệu suất và giảm độ trễ trong mạng lưới.

Cụ thể, khi một thiết bị trong mạng nhận được một thông điệp mạng (Network PDU), nó sẽ kiểm tra xem thông điệp này đã từng được nhận trước đó không, bằng cách so sánh với dữ liệu trong Network Message Cache. Nếu thông điệp đã tồn tại trong Network Message Cache, điều này ngụ ý rằng thông điệp đã được xử lý hoặc đã đi qua nút này trước đó, do đó nút sẽ loại bỏ ngay lập tức thông điệp đó mà không cần phải xử lý nó nữa.

Tuy nhiên, nếu thông điệp không tồn tại trong Network Message Cache, điều này ngụ ý rằng đó là một thông điệp mới và chưa được xử lý bởi nút đó. Do đó, nút sẽ tiến hành xử lý thông điệp này, bao gồm các bước kiểm tra tính hợp lệ của thông điệp (ví dụ: kiểm tra bảo mật của thông điệp), và sau đó lưu trữ thông điệp này vào Network Message Cache để sử dụng cho các kiểm tra và xử lý tiếp theo**.**

Sau khi qua bước kiểm tra network message cache thì node kiểm tra giá trị TTL, nếu giá trị TTL lớn hơn 2 và giá trị DST được kiểm tra không phải của chính phần tử của nó, giá trị TTL trừ 1 và bản tin lại được chuyển đi. Sẽ có một độ trễ xảy ra giữa việc truyền và nhận PDU để tránh xung đột xảy ra.

Nếu TTL = 1 thì nó sẽ nhận PDU này và chuyển khung của nó lên Lower Transport Layer để xử lý. Hình ảnh bên dưới thể hiện lưu đồ thuật toán của việc nhận khung Network PDU từ một node.

A diagram of a computer program

Description automatically generated

Hình 2. 17 Lưu đồ thuật toán nhận network PDU

**Gửi Network PDU:**

Sử dụng các vùng trong khung gồm IVI, NID, CTL, TTL, SRC, DST, TransportPDU, NETMIC. Ở đây, sẽ cần thêm trường SEQ để đại diện cho số thứ tự của gói mạng. Mỗi khi có một khung Network PDU mới, giá trị này được tăng lên 1.

* + - * 1. **Lower Transport Layer**

PDU của layer này được chia thành hai dạng định nghĩa bởi hai trường CTL và SEG. MSB của layer này là trường SEG, là trường định nghĩa rằng các PDU của Lower Transport layer là dạng nào. SEG và CTL sẽ xác định định dạng khung của layer, CTL thì được quyết định trên Upper Transport Layer. Ở đây, nhà sản xuất phân ra hai dạng này vì liên quan đến độ dài dữ liệu, nếu dữ liệu bị vượt quá khung dữ liệu của mỗi gói truyền đi, thì bản tin sẽ bị chia để phù hợp khi chuyển sang Network Transport Layer.

Bảng 2. 5 Các dạng của PDU trong Transport Lower Layer

|  |  |  |
| --- | --- | --- |
| CTL | SEG | Lower Transport PDU |
| 0 | 0 | Bản tin truy cập không bị chia |
| 0 | 1 | Bản tin truy cập bị chia |
| 1 | 0 | Bản tin điều khiển không bị chia |
| 1 | 1 | Bản tin điều khiển bị chia |

**Bản tin truy cập không bị chia:**

A close-up of a label

Description automatically generated

Hình 2. 18 Khung bản tin truy cập không bị chia

A screenshot of a computer screen

Description automatically generated

Hình 2. 19 Định nghĩa các thành phần trong khung

**Bản tin điều khiển không bị chia:**

A black text on a white background

Description automatically generated

Hình 2. 20 Khung bản tin điều khiển không chia

A screenshot of a computer

Description automatically generated

Hình 2. 21 Định nghĩa trường trong khung bị chia

Bản tin bị chia ( Segmented Control Message) được gùng khi mà các PDU truyền từ tầng trên xuống không vừa khung của Network PDU. Dưới đây là hình ảnh của khung.

PDU của tầng trên bị chia thành các segment.

A close-up of a graph

Description automatically generated

Hình 2. 22 Khung bị chia

Hình ảnh dưới đây thể hiện rõ ràng về việc PDU tầng trên bị phân chia và đưa xuống Network Layer.

A computer screen shot of a diagram

Description automatically generated

Hình 2. 23 Bản tên bị phân chia dần từ tầng trên xuống tầng dưới

Giải thích hình ảnh trên: Một Upper Transport Access PDU được gửi đi, trong đó có một octet opcode (mã hoạt động), 3 octet cho trường NetKeyIndexAndAppKeyIndex, và 16 octet cho AppKey. Điều này có nghĩa là khi được mã hóa và xác thực bằng một khóa ứng dụng, Upper Transport PDU có kích thước là 24 octet. Dữ liệu này được phân mảnh bởi tầng lower transport layer thành hai đoạn, Đoạn 0 và Đoạn 1. Mỗi đoạn có một tiêu đề để xác định số đoạn, và sau đó được chuyển cho tầng network layer, nơi mà Network PDU hoàn chỉnh được tính toán. Tầng network layer sau đó mã hóa Network PDU sử dụng số thứ tự cho Network PDU.

* + - * 1. **Upper Transport Layer**

Upper Transport Layer sẽ là chứa các payload từ access layer. Khung của nó sẽ được thể hiện như sau**: A white background with black text

Description automatically generated**

Hình 2. 24 Khung của upper Transport Layer

Dữ liệu truy cập được cung cấp bởi tầng truy cập (access layer). Nếu TransMIC là 32 bit, dữ liệu truy cập có thể từ một octet đến 380 octet. Nếu TransMIC là 64 bit, dữ liệu truy cập có thể từ một octet đến 376 octet. Đây là tầng sẽ mã hóa , giải mã và xác thực dữ liệu của (tới) Access Layer.

* + - * 1. **Access Layer**

PDU: Chứa 32 segment có 12 octet, tạo thành tối đa 384 octet. Số octet của TransMIC dài 4 octet có Access Payload dài 380 octet. Số octet sử dụng sẽ quy định octet cho ứng dụng. Lower Transport sẽ chia PDU khi đưa vào Network.

Tầng Access Layer có vùng đầu tiên là opcode cho ứng dụng, net index, app index, và các octet của ứng dụng.

* + - * 1. **Foundation model**

Tầng Foundation Model trong mạng lưới Bluetooth Low Energy (BLE) Mesh là một phần của các mô hình tiêu chuẩn được xác định bởi Bluetooth Special Interest Group (SIG). Tầng này cung cấp các khả năng cơ bản và chung cho các thiết bị trong mạng lưới, bao gồm các chức năng như quản lý địa chỉ, quản lý bảo mật, quản lý thời gian sống của gói tin, và các khả năng khác liên quan đến quản lý và điều khiển mạng lưới.

Tầng Foundation Model cung cấp một cơ sở chung cho việc phát triển các ứng dụng BLE Mesh và đảm bảo tính tương thích giữa các thiết bị từ các nhà sản xuất khác nhau. Đây là tầng ứng dụng, người dùng sẽ cấu hình các ứng dụng mong muốn của người dùng ở tầng này. Một số cái có thể kể đến như Generic Configuration Model, Generic Onoff Model….

##### **Bảo mật trong mạng BLE MESH**

Trong mạng lưới Bluetooth Mesh, việc đảm bảo bảo mật là một yêu cầu bắt buộc và được thực hiện thông qua một hệ thống bảo vệ hai lớp. Mục tiêu chính là bảo vệ các nút trong mạng khỏi các cuộc tấn công từ bên ngoài. Mỗi nút trong mạng được bảo vệ độc lập với các nút khác, cũng như các ứng dụng và thiết bị khác trong cùng mạng. Các yếu tố quan trọng trong bảo mật mạng Mesh bao gồm:

* Tất cả các tin nhắn được mã hóa và xác thực, đảm bảo tính toàn vẹn và bảo mật của dữ liệu truyền đi.
* Bảo mật mạng, bảo mật ứng dụng và bảo mật thiết bị được xử lý độc lập, giúp tăng cường tính linh hoạt và an toàn của mạng.
* Các khóa bảo mật có thể được thay đổi trong quá trình hoạt động của mạng thông qua quá trình Key Refresh, tăng cường tính động và khả năng chống lại các cuộc tấn công.
* Cung cấp cơ chế chống lại tấn công phát lại, ngăn chặn việc sử dụng lại các thông điệp đã được gửi trước đó.
* Quá trình cấp phép cho một thiết bị được xem như một quá trình bảo mật, đảm bảo rằng chỉ các thiết bị được ủy quyền mới có thể tham gia vào mạng.
* Các nút trong mạng có khả năng gỡ bỏ mình từ mạng một cách an toàn, ngăn chặn việc tấn công từ các thiết bị không được ủy quyền.

**Các loại khóa bảo mật:**

Trong mạng lưới Bluetooth Mesh, để đảm bảo tính bảo mật của dữ liệu, hệ thống sử dụng ba loại khóa bảo mật như sau:

* Khóa Mạng (NetKey): Đây là loại khóa được cấp cho mỗi nút trong mạng khi thiết bị gia nhập mạng Mesh. Quá trình cấp phép này được thực hiện bởi thiết bị Provisioner. Khóa Mạng được sử dụng để mã hoá và giải mã dữ liệu ở tầng Network của mạng lưới. Nó là cơ sở cho việc mã hoá và giải mã của bản tin truyền đi và nhận về trong mạng Mesh.
* Khóa Ứng Dụng (AppKey): Mỗi thiết bị trong mạng sử dụng một hoặc nhiều Khóa Ứng Dụng để mã hoá và giải mã dữ liệu ở tầng Access. Các AppKey này được cấp bởi thiết bị cấp phép và được sử dụng cho các mục đích cụ thể của ứng dụng hoặc tính năng mà thiết bị đó hỗ trợ.
* Khóa Thiết Bị (DevKey):Đây là một loại khóa đặc biệt độc lập cho mỗi nút trong mạng. DevKey chỉ thuộc sở hữu của thiết bị cấp phép và chỉ nút đó có quyền sử dụng. DevKey được sử dụng trong quá trình cấp phép để bảo vệ kết nối giữa thiết bị cấp phép và thiết bị muốn gia nhập mạng.

**Tấn công mạng “trash Can” và cơ chế để bảo vệ:**

Trong giao thức Bluetooth Mesh, một nút có thể sở hữu nhiều khóa bảo mật. Tuy nhiên, khi cần gỡ bỏ một nút khỏi mạng, các khóa bảo mật mà nó sở hữu không tự động bị xóa. Điều này có thể tạo ra một rủi ro bảo mật nếu thông tin từ thiết bị bị gỡ bỏ có thể được sử dụng để tấn công mạng.

Để giải quyết vấn đề này, một giải pháp được đề xuất là thiết bị cấp phép sẽ thêm địa chỉ của nút cần gỡ bỏ vào một danh sách đen (blacklist). Sau đó, một quá trình Key Refresh sẽ được khởi tạo. Sau khi quá trình Key Refresh hoàn tất, tất cả các nút trong mạng, trừ những nút nằm trong danh sách đen, sẽ nhận các khóa mạng và khóa ứng dụng mới. Khi một nút bị gỡ bỏ khỏi mạng, nó sẽ vẫn giữ các khóa mạng và khóa ứng dụng cũ, nhưng do đã bị thêm vào danh sách đen, nên không thể sử dụng để truy cập mạng, từ đó giảm thiểu mối đe dọa bảo mật.

**Relay attack và cơ chế để bảo vệ:**

Tấn công Relay là một kỹ thuật tấn công mà một thiết bị nghe trộm sẽ chặn và ghi lại một hoặc nhiều bản tin, sau đó phát lại chúng với mục đích đánh lừa thiết bị đích, thực hiện hành động nào đó mà không được ủy quyền. Ví dụ điển hình là khi tín hiệu khóa xe bị chặn và ghi lại, sau đó được phát lại để đánh cắp.

Bluetooth Mesh cung cấp một cơ chế chống lại tấn công Relay. Cơ chế này dựa trên việc sử dụng hai trường trong gói dữ liệu (PDU), được gọi là Sequence Number (SEQ) và IV Index. Mỗi khi một bản tin được gửi đi, các thiết bị trong mạng sẽ tăng giá trị của SEQ lên 1. Khi một thiết bị nhận được một bản tin từ một thiết bị khác có giá trị SEQ nhỏ hơn so với giá trị SEQ của bản tin hợp lệ gần nhất từ thiết bị đó, thiết bị sẽ cảnh báo hoặc bỏ qua bản tin đó, nhằm ngăn chặn tấn công Relay.

IV Index là một trường dữ liệu độc lập, được sử dụng cùng với SEQ. Bản tin sẽ không được chấp nhận nếu giá trị IV Index trong bản tin nhỏ hơn so với giá trị IV Index của bản tin hợp lệ gần nhất. Các biện pháp này giúp bảo vệ mạng Bluetooth Mesh khỏi các cuộc tấn công Relay và tăng cường tính bảo mật của hệ thống.

##### **Quá trình một thiết bị tham gia mạng mesh**

###### **Một số định nghĩa cơ bản về provisioning process**

Trước hết, quá trình một thiết bị tham gia mạng (provisioning) bao gồm các bước như sau:

Provisioning là quá trình đưa thiết bị mới vào mạng, được quản lý bởi một Provisioner. Provisioner cung cấp cho thiết bị chưa được provision các dữ liệu cấp phát cần thiết như một khóa mạng, chỉ số IV hiện tại và địa chỉ unicast cho từng phần tử, giúp thiết bị trở thành một nút trong mạng. Provisioner thường là một điện thoại thông minh hoặc một thiết bị di động khác.

Để provisioning cho một thiết bị, bearer provisioning phải được thiết lập giữa Provisioner và thiết bị. Thiết bị có thể được xác định đến provisioner bằng UUID Thiết bị và thông tin bổ sung khác nếu có.

Sau khi bearer provisioning được thiết lập, Provisioner tạo ra một bí mật chia sẻ với thiết bị bằng cách sử dụng giao thức Elliptic Curve Diffie-Hellman (ECDH). Sau đó, thiết bị được xác thực bằng thông tin (OOB), có thể bao gồm khóa công khai của thiết bị, một bí mật dài, hoặc yêu cầu đầu vào/đầu ra cụ thể. Thông tin OOB này đảm bảo xác thực cho thiết bị.

Sau khi được xác thực, dữ liệu cấp phát được truyền đến thiết bị một cách an toàn, được mã hóa bằng một khóa được tạo ra từ bí mật chia sẻ. Khóa của thiết bị được tạo ra từ cả ECDHSecret và ProvisioningSalt. Đối với provisioning này sẽ có một giao thức riêng.

A screenshot of a computer screen

Description automatically generated

Hình 2. 25 Provisioning protocol stack

PB ADV là provisioning beaer được dùng sử dụng Generic Provisioning PDU thông qua kênh quảng cáo. Bảng bên dưới định nghĩa khung truyền trong một lần ADV.

Bảng 2. 6 Khung truyền của PB ADV

|  |  |  |
| --- | --- | --- |
| Trường | Size | Mô tả |
| Độ dài | 1 | Độ dài của khung |
| Kiểu AD | 1 | PB ADV |
| Nội dung | Var | PB ADV PDU |

Bảng 2. 7 Chi tiết bên trong của PB ADV PDU

|  |  |  |
| --- | --- | --- |
| Trường | Size | Mô tả |
| Link ID | 4 | Nhận dạng kết nối giữa hai thiết bị |
| Chỉ số giao dịch | 1 | Chỉ số để cho biết số lần PDU được gửi một thiết bị, mỗi lần gửi +1 |
| Generic Provisioning PDU | 1-24 | Số Generic Provisioning PDU được truyền đi |

Các trường cơ bản được phân theo các layer của mạng Mesh từ tầng Network đến tầng Access (bao gồm các khóa).

**A diagram of a diagram

Description automatically generated**

Hình 2. 26 Phân lớp của Provisioning Protocol

###### **Generic Provisioning PDU**

**A graph of a graph

Description automatically generated with medium confidence**

Hình 2. 27 Khung của Provisioning PDU

**Generic Provisioning Control:**

Bit thứ 6 và bit thứ 7 của trường này định dạng kiểu khung của trường.

A screenshot of a computer

Description automatically generated

Hình 2. 28 Định dạng kiểu cho Provisioning control

**Trường hợp 1: Transaction Start ( 0b00)**

A white rectangular object with black text

Description automatically generated

Hình 2. 29 Transaction Start PDU

Ở đây, Total Length là chỉ ra số lượng octet của provisioning PDU, Còn FSC là trường để kiểm tra khung. FSC là là kiểm tra khung theo trình tự.

**A screenshot of a computer

Description automatically generatedTrường hợp 2: Transaction ACK:**

Hình 2. 30 Transaction ACK format

**A white rectangular object with black text

Description automatically generated**

Đây là trường để xác thực cho các Provisioning PDU.

**Trường hợp 3: Transaction continue:**

A close-up of a white sign

Description automatically generated

Hình 2. 31 Transaction continue PDU

Ở đây, Segment index là segment number của transaction. Đây là transaction được gửi để thêm các segment của provisioning PDU sau khi trải qua các transaction đầu tiên.

**Trường hợp 4: Provisioning Bearer Control**:

A close-up of a coupon

Description automatically generated

Hình 2. 32 Provisioning Bearer Frame

Ở đây, bearer opcode là opcode của provisioning bearer control PDU. Parameter là các biến định nghĩa bởi bearer opcode.

**Provisioning PDU – Bên trong Generic Provisioning Payload:**

A close-up of a ruler

Description automatically generated

Hình 2. 33 Frame của Provisioning PDU

Octet đầu tiên chính là định nghĩa kiểu cho provisioning PDU.

A screenshot of a computer

Description automatically generated

Hình 2. 34 Dang sách các dạng của Provisioning PDU

###### **Chi tiết về quá trình gia nhập mạng**

**Bước 1: Quảng bá và mời gia nhập**

**Quảng bá:** Nếu một thiết bị hỗ trợ PB-ADV nhưng chưa được "provisioned" trong mạng, nó sẽ phát sóng "Unprovisioned Device Beacon", giúp các thiết bị "Provisioner" có thể nhận được dữ liệu cần thiết để tiến hành bước tiếp theo.

**Mời gia nhập**: Trong quá trình này, Provisioner sẽ gửi một lời mời đến thiết bị để được cấp phép, được gọi là Provisioning Invite PDU. Thiết bị cần được cấp phép sẽ phản hồi bằng cách cung cấp các thông tin đầy đủ và chính xác về nó.

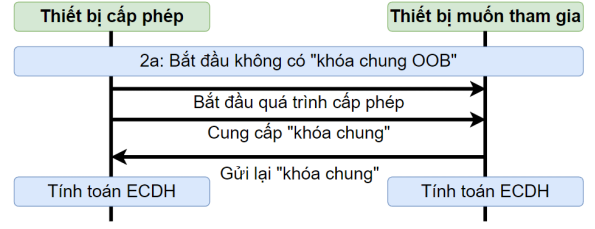
A screenshot of a chat

Description automatically generated

Hình 2. 35 Mời gia nhập và kiểm tra khả năng kết nối

**Bước 2: Trao đổi khóa chung**

Có hai trường hợp phụ thuộc vào tính khả dụng của khóa chung. Trong trường hợp thiết bị cấp phép xác định có thể cấp phép cho thiết bị khác, nó sẽ gửi một bản tin "Provisioning Start PDU". Khi thiết bị mới nhận được bản tin này, nó sẽ thiết lập "Attention Timer" về 0x00 để chỉ ra sự chú ý tới quá trình cấp phép. Thiết bị cấp phép sau đó sẽ chọn một thuật toán từ bản tin "Provisioning Capability PDU" được gửi bởi thiết bị mới, và sẽ lựa chọn thuật toán tốt nhất. Trong trường hợp khóa chung không sử dụng được với công nghệ OOB (Out of Band), cả thiết bị cấp phép và thiết bị mới sẽ tạo một mã mới cho mỗi quá trình trao đổi.



Hình 2. 36 Cấp phép không có khóa chung OOB

Nếu khóa chung có sẵn thông qua một cơ chế OOB (Out-of-Band), thì một cặp khóa mới sẽ được tạo ra bởi thiết bị cấp phép. Khóa chung của cặp khóa được tạo ra sẽ được truyền từ thiết bị cấp phép đến thiết bị được cấp phép, và một khóa chung tĩnh sẽ được đọc từ thiết bị thông qua công nghệ OOB thích hợp.

A screenshot of a chat

Description automatically generated

Hình 2. 37 Cấp phép có khóa chung OOB

**Bước 3: Xác thực**

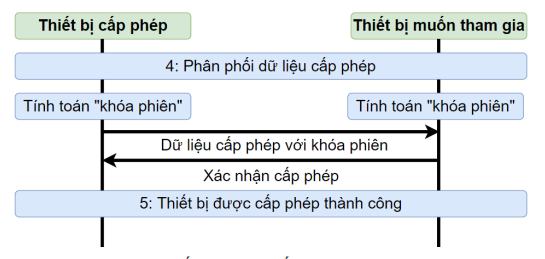
Trong quá trình xác thực, thiết bị cần cấp phép sẽ tạo ra một số ngẫu nhiên, có thể là một hoặc nhiều chữ số, và hiển thị cho người dùng. Người dùng sau đó sẽ nhập số ngẫu nhiên này vào thiết bị cấp phép, để xác định sự khớp nhau. Số ngẫu nhiên này được sử dụng để mã hóa thông tin bảo mật trong quá trình trao đổi giữa hai thiết bị.

**Bước 4: Phân phối dữ liệu cấp phép**

Sau khi quá trình xác thực thành công, một khóa phiên sẽ được tạo ra từ số ngẫu nhiên và các khóa trước đó trên cả hai thiết bị. Khóa phiên này được sử dụng để mã hóa dữ liệu cấp phép, bao gồm địa chỉ được cấp, khóa mạng và khóa ứng dụng, từ thiết bị cấp phép đến thiết bị cần được cấp phép. Khi quá trình phân phối dữ liệu hoàn thành, quá trình cấp phép được coi là thành công.

Bảng 2. 8 Dạng dữ liệu cấp phép của khóa phiên

|  |  |  |
| --- | --- | --- |
| Trường | Số Octet | Ghi chú |
| Network Key | 16 | Netkey |
| Key Index | 2 | Chỉ số của netkey |
| Flags | 1 | Flag bitmask |
| IV index | 4 | Giá trị của IV index |
| Unicast Address | 2 | Địa chỉ unicast của phần từ |



Hình 2. 38 Phôi phối dữ liệu cấp phép

##### **Quá trình chuyển một bản tin**

Tại nguồn (Source), tầng ứng dụng gọi hàm của mô hình (model) mà muốn truyền, kết hợp với opcode và gửi tới tầng Access. Tại tầng Access, các trường TTL, địa chỉ nguồn (SRC) và địa chỉ đích (DST) được thêm vào bản tin, sau đó chuyển tới tầng Transport. Tại tầng Upper Transport, bản tin được mã hóa bằng khóa ứng dụng hoặc khóa thiết bị, và các trường SEQ (độc nhất cho mỗi PDU), AKF, AID và TransMIC được thêm vào để kiểm tra tính toàn vẹn. Tầng Lower Transport quy định việc chia nhỏ bản tin theo trường SEG nếu cần, sau đó chuyển tới tầng Network. Tại tầng Network, các trường IVI, CTL và NetMIC được sử dụng để kiểm tra tính đầy đủ và kết hợp với khóa mạng, trước khi chuyển tới tầng BLE Controller để phát bản tin qua kênh quảng bá và định thời gian phát.

Tại Relay, bản tin được nhận từ tầng Bearer, sau đó được chuyển tới tầng Network. Ở tầng Network, khóa mạng được sử dụng để giải mã và kiểm tra địa chỉ đích của bản tin. Nếu địa chỉ đích không đúng, TTL giảm đi 1 và bản tin được mã hóa lại với khóa mạng cũ trước khi chuyển trở lại tầng BLE Controller

Tại Đích (Destination), khóa mạng được giải mã để kiểm tra địa chỉ đích, trường IVI và TTL. Nếu địa chỉ đích đúng nhưng giá trị TTL không thỏa mãn, bản tin sẽ bị hủy bỏ. Nếu TTL đúng, bản tin được chuyển lên tầng Transport. Tại tầng Lower Transport, các trường được kiểm tra và các phần nhỏ được tổng hợp lại. Nếu địa chỉ đích là Unicast, một ACK được gửi từ nút nhận đến nút gửi. Nếu là địa chỉ Group hoặc Virtual, không có ACK được gửi. Tầng Upper Transport giải mã khóa ứng dụng và kiểm tra các trường, trước khi chuyển bản tin tới tầng Access để kiểm tra opcode, khóa ứng dụng và địa chỉ nhận, và cuối cùng gửi bản tin tới tầng ứng dụng để xác định mô hình và thực hiện các lệnh tương ứng.

A diagram of a network connection

Description automatically generated

Hình 2. 39 Quá trình chuyển một bản tin

## **WiFi provisioning thông qua SoftAP**

### **Chế độ softAP trên vi điều khiển**

SoftAP là bản viết tắt của "Software Enabled Access Point" tức là "Truy cập được Kích hoạt bằng Phần mềm". Đây là một tính năng phổ biến được tìm thấy trong các mạng thiết bị, đặc biệt là các bộ định tuyến và trạm truy cập. SoftAP cho phép thiết bị hoạt động như một điểm truy cập cho các thiết bị có khả năng kết nối Wi-Fi, tương tự như một phần cứng truy cập thông thường.

Với SoftAP, một thiết bị có thể tạo mạng Wi-Fi riêng cho mình, cho phép các thiết bị khác kết nối với nó bằng cách không dây. Tính năng này thường được sử dụng trong các vấn đề khi cần mở rộng hoặc tạo một mạng Wi-Fi mới mà chưa có sẵn cố định truy cập.

Chức năng SoftAP thường được sử dụng trong các ứng dụng khác nhau, bao gồm:

* Điểm phát Wi-Fi công cộng: Khách sạn, quán cà phê và các địa điểm công cộng khác có thể sử dụng SoftAP để cung cấp quyền truy cập Wi-Fi tạm thời cho khách hàng.
* Chia sẻ internet: Nó cho phép thiết bị chia sẻ kết nối internet của mình với các thiết bị khác trong phạm vi gần đó.
* Trên mạng không dây: Một số máy sử dụng SoftAP để tạo mạng tạm thời nhằm kết nối các thiết bị trong mạng không dây.
* Thiết bị Internet of Things (IoT): Các thiết bị IoT có thể sử dụng SoftAP để tạo điều kiện cho việc cài đặt và cấu hình cấm đầu bằng cách cho phép người dùng kết nối trực tiếp với thiết bị trước khi nó được tích hợp vào một mạng hiện có.
* SoftAP có thể cấu hình và quản lý thiết bị cài đặt giao diện thông tin, thường có thể truy cập vào web trình duyệt thông tin hoặc một ứng dụng dành riêng. Người dùng có thể thiết lập các thông số như mạng tên (SSID), cài đặt bảo mật và cấu hình DHCP để tùy chỉnh SoftAP theo nhu cầu cụ thể.

### **Kết nối WiFi router thông qua chế độ SoftAP trên vi điều khiển**

Bước 1: Đầu tiên, người dùng sẽ dùng thiết bị thông minh như điện thoại để kết nối vào Accesspoint mà vi điều khiển phát.

Bước 2: Sau khi kết nối wifi thành công, đến phiên thiết lập, người dùng sẽ được yêu cầu nhập mật khẩu và gửi lại cho thiết bị kiểm tra trước khi thiết lập các thông số WiFi router mà người dùng mong muốn cho thiết bị.

Bước 3: Sau khi thiết bị chấp nhận cho người dùng cấu hình, điện thoại thông minh của người dùng sẽ scan toàn bộ wifi router xung quanh và lựa chọn một router mà người dùng mong muốn thiết bị kết nối vào

Bước 4: Người dùng chọn xong router , nhập SSID và password của router cho thiết bị, sau đó, SSID và password sẽ được chuyển tới thiết bị, thiết bị sẽ thực hiện check và phản hồi lại trạng thái kết nối của thiết bị cho người dùng

Bước 5: Khi thiết bị chính thức kết nối với router mà người dùng mong muốn, softAP sẽ bị tắt và thiết bị chuyển sang chế độ station, hoạt động dựa trên IP router cấp phát.

Giao tiếp dựa trên SoftAP là một phương tiện tương tác cao. Tuy nhiên, cũng có một số yếu tố cần xem xét:

* Thiết bị sử dụng cùng một sóng vô tuyến để tổ chức SoftAP và kết nối với AP được cấu hình. Bởi vì chúng có thể nằm trên các kênh khác nhau, điều này có thể gây ra việc cập nhật kết nối trạng thái không được nhận đủ độ tin cậy của điện thoại.
* Điện thoại người dùng phải ngắt kết nối từ AP hiện tại của nó để kết nối với SoftAP. Mạng ban đầu sẽ được khôi phục chỉ khi quá trình cung cấp thông tin hoàn tất và SoftAP không được sử dụng.

Hình ảnh bên dưới là quá trình chi tiết:

A screenshot of a computer

Description automatically generated

Hình 2. 40 Quá trình người dùng kết nối thiết bị với router thông qua softAP

## **Giao thức MQTT**

### **Khái niệm giao thức MQTT**

MQTT là một giao thức truyền tải tin nhắn theo mô hình client-server với hệ thống Publish/Subscribe. Nó rất nhẹ, mã nguồn mở, đơn giản và được thiết kế để dễ triển khai. Những đặc điểm này khiến nó trở nên lý tưởng cho nhiều tình huống, bao gồm cả môi trường có hạn như giao tiếp trong ngữ cảnh Machine to Machine (M2M) và Internet of Things (IoT) nơi yêu cầu một dải mã nguồn nhỏ và/hoặc băng thông mạng hạn chế. MQTT sử dụng định dạng tin nhắn nhị phân để giao tiếp giữa các máy khách và các trình điều khiển. Điều này khác biệt so với các giao thức khác sử dụng định dạng văn bản, như HTTP hoặc SMTP.

### **Kiến trúc Pub/Sub**

Kiến trúc Pub/Sub mang đến một phương án khác biệt so với các mô hình truyền thống dựa trên client-server (yêu cầu-phản hồi). Trong mô hình yêu cầu-phản hồi, máy khách phải trực tiếp giao tiếp với điểm cuối máy chủ, tạo ra một điểm trở ngại làm giảm hiệu suất. Tuy nhiên, mô hình pub/sub giải quyết vấn đề này bằng cách tách rời người gửi tin nhắn và những người nhận tin. Người gửi và người nhận không cần phải biết về sự tồn tại của nhau. Thay vào đó, một thành phần thứ ba, gọi là trung gian, đảm nhận vai trò kết nối giữa chúng. Sự tách rời này tạo ra một quy trình giao tiếp nhanh chóng và hiệu quả hơn.

A diagram of a bee

Description automatically generated

Hình 2. 41 Ví dụ về kiến trúc Publish Subscribe

MQTT client và MQTT broker:

Trong MQTT, một MQTT client thường đề cập đến các nhà xuất bản và người đăng ký. Một nhà xuất bản là một client gửi tin nhắn, trong khi một người đăng ký là một client nhận tin nhắn. Tuy nhiên, một MQTT client cũng có thể là một nhà xuất bản và một người đăng ký.

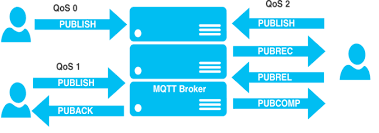
Một MQTT client có thể là bất kỳ thiết bị nào, từ một vi điều khiển nhỏ cho đến một máy chủ lớn, chạy một thư viện MQTT và kết nối với một trung tâm MQTT qua mạng.

Một MQTT client có thể là một máy tính thông thường chạy một ứng dụng MQTT đồ họa được sử dụng cho mục đích kiểm tra. Bất kỳ thiết bị nào sử dụng giao thức mạng TCP/IP và có phần mềm thực hiện chức năng của một MQTT client có thể được gọi là một MQTT client. MQTT được thiết kế để hoạt động trên nền giao thức TCP/IP, vì vậy bất kỳ thiết bị nào nói chuyện TCP/IP và thực hiện giao thức MQTT có thể là một MQTT client. Việc triển khai của client cho giao thức MQTT là đơn giản và được tối ưu hóa, làm cho nó lý tưởng cho các thiết bị nhỏ.

Một MQTT broker là trung tâm chính trong hệ thống giao tiếp theo mô hình phát hành/đăng ký mà nhận các tin nhắn từ các nhà xuất bản và phân phối chúng đến các người đăng ký. Nó đóng một vai trò quan trọng trong việc quản lý luồng truyền thông giữa các client MQTT và đảm bảo việc giao nhận tin nhắn được tin cậy.

### **Quality of service**

Chất lượng dịch vụ (QoS) rất quan trọng trong MQTT do vai trò của nó trong việc cung cấp cho khách hàng khả năng chọn mức dịch vụ phù hợp với cả độ tin cậy của mạng và yêu cầu của ứng dụng. Khả năng vốn có của MQTT trong việc xử lý việc truyền lại tin nhắn và đảm bảo gửi, ngay cả trong điều kiện mạng không đáng tin cậy, khiến QoS trở nên cần thiết để tạo điều kiện liên lạc liền mạch trong những môi trường đầy thách thức như vậy. Bằng cách cung cấp các mức QoS khác nhau, MQTT trao quyền cho khách hàng tối ưu hóa việc sử dụng mạng của họ và đạt được sự cân bằng mong muốn giữa độ tin cậy và hiệu quả.



Hình 2. 42 Các mức độ của QoS trong MQTT

Hình ảnh trên cho thấy bao gồm 3 mức độ QoS:

* Qos0: Ở mức thấp nhất, QoS 0 trong MQTT cung cấp cơ chế gửi nỗ lực tốt nhất trong đó người gửi không mong đợi sự xác nhận hoặc đảm bảo gửi tin nhắn.
* Qos1: Trong QoS 1 của MQTT, trọng tâm là đảm bảo gửi tin nhắn ít nhất một lần đến người nhận. Khi một tin nhắn được xuất bản với QoS 1, người gửi sẽ giữ một bản sao của tin nhắn cho đến khi nhận được gói PUBACK từ người nhận, xác nhận việc nhận thành công.
* Qos2: Ở QoS 2 cung cấp mức dịch vụ cao nhất trong MQTT, đảm bảo rằng mỗi tin nhắn được gửi chính xác một lần đến người nhận dự định.

# **CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG**

## **Tổng quan hệ thống**

### **Mục tiêu thiết kế**

Tạo ra giải pháp điều khiển chiếu sáng thông minh và ổn định cho smart home. Dưới đây là các mục tiêu thiết kế chính của hệ thống:

* Hệ thống được thiết kế sao cho các thiết bị trong nhà có thể được tích hợp để dễ dàng sử dụng và mở rộng, cho phép người dùng có thể quản lý, thêm và xóa thiết bị một cách linh hoạt trên smartphone.
* Các thiết bị có khả năng tự phục hồi khi có sự cố xảy ra khi mất điện đột ngột. Ngay sau khi có điện, các thiết bị ngay lập tức được liên kết với nhau thông qua BLE MESH.
* Hệ thống được thiết kế để cho phép người dùng linh hoạt thiết lập các bối cảnh trong nhà như hẹn giờ, điều khiển các đèn theo khu vực,….
* Đảm bảo tính bảo mật cao cho người dùng trong việc quản lý các thiết bị trong nhà. Thiết bị trong nhà được liên kết với nhau thông qua khóa mạng và khóa ứng dụng nhất định tránh xung đột từ thiết bị bên ngoài. Người dùng cần phải có tài khoản riêng để quản lý và tương tác với thiết bị trong nhà của mình.
* Hệ thống phải hoạt động ổn định, trễ lan truyền không quá 3 giây trong phạm vi 30 mét, trong trường hợp không thể điều khiển thông qua internet do router trong nhà gặp sự cố, yêu cầu các thiết bị phải điều khiển được bình thường thông qua Bluetooth.

### **Sơ đồ khối tổng quan**



Hình 3. 1 Sơ đồ khối tổng quan của hệ thống

Từ hình 3.1, hệ thống gồm các thiết bị sau:

* Gateway: Kết nối trực tiếp vào nguồn điện, được đóng vai trò gateway để trở thành thiết bị trung gian vận chuyển tin nhắn từ bên ngoài vùng phủ của mạng BLE MESH vào mạng và ngược lại.
* Switch: Là nút bấm mà người dùng có thể trực tiếp tương tác để điều khiển đèn.
* Light: Được điều khiển bằng cách bấm nút hoặc điều khiển từ xa bởi người dùng.

Giải thích chi tiết sơ đồ khối tổng quan:

* Người dùng sử dụng điện thoại thông minh tương tác với web/server để có thể điều khiển từ xa thông qua internet hoặc họ có thể điều khiển và đọc trạng thái thiết bị thông qua Bluetooth khi ở trong phạm vi BLE MESH. Trong mạng BLE MESH có hai vai trò thiết bị chính. Một là client device, là thiết bị gửi lệnh điều khiển cho thiết bị khác. Hai là server device, là thiết bị nhận lệnh điều khiển từ client.
* Ở trong sơ đồ này, client device sẽ đóng vai trò là gateway, đóng vai trò trung gian, thay đổi linh hoạt giữa giao thức BLE và WiFi để có thể trung chuyển bản tin lên cloud hoặc nhận bản tin của cloud để điều khiển thiết bị theo nhu cầu của người dùng.
* Các thiết bị trong BLE MESH sẽ liên kết với nhau thông qua BLE MESH, chúng sẽ biết đến nhau thông qua netkey và có thể thực sự giao tiếp với nhau khi có chung app key.

## **Thiết kế phần cứng**

### **Mục tiêu thiết kế**

Công tắc âm tường được sử dụng để bật tắt, điều khiển bóng đèn. Mục tiêu thiết kế cần đáp ứng:

* Điện áp đầu vào: 220VAC
* Điện áp đầu ra cấp nguồn cho vi điều khiển hoạt động
* Điện áp đầu ra 220VAC được điều khiển bởi tín hiệu từ vi điều khiển
* Có nút nhấn điều khiển thủ công
* Có đèn báo trạng thái
* Kích thước mạch vừa với đế âm đơn chữ nhật SINO (W62mm x L100mm x Hx38)

### **Thiết kế công tắc âm tường**

#### **Lựa chọn linh kiện**

**Lựa chọn Relay**



Hình 3. 2 Relay

Sử dụng relay Songle 220V 5A có:

* Dòng Relay 5 chân
* 3 tiếp điểm: COM – NO – NC
* Điện áp định mức: 5V | 12V | 24V DC
* Dòng tải: 5A/220V AC, 10A/125V AC, 20A/14V DC
* Dòng tiêu thụ:
  + relay 5V: ~ 115mA | 0.54W
  + relay 12V: ~ 45mA | 0.54W
  + relay 24V: ~ 20mA | 0.5W
* Kích thước: vui lòng xem ảnh kích thước thực tế
* Trọng lượng: ~ 5g

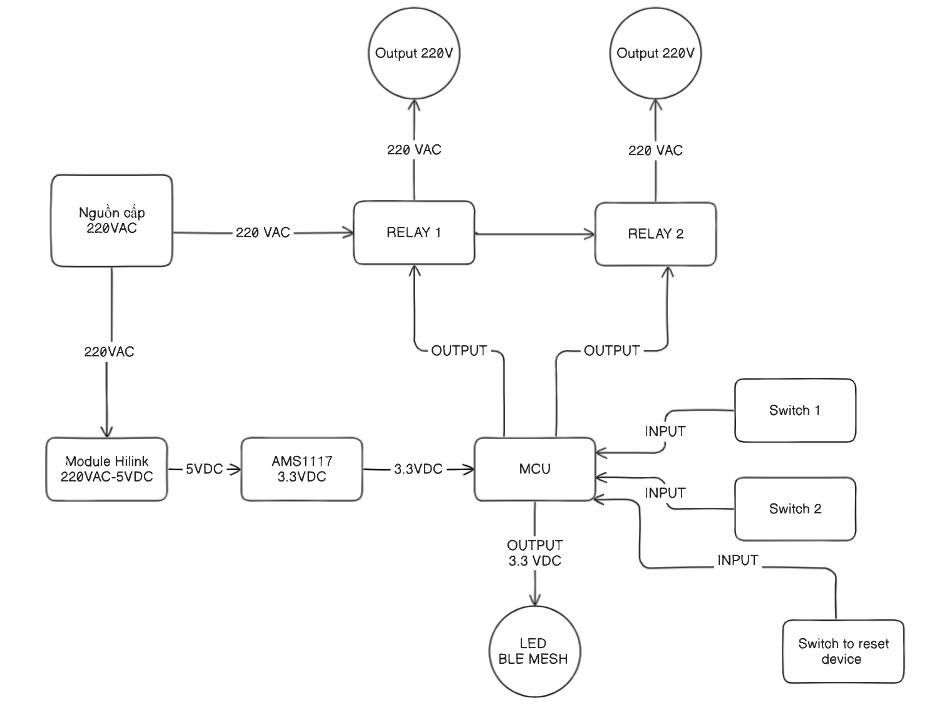
**Lựa chọn khối điều khiển trung tâm**

Khối điều khiển trung tâm cần có yêu cầu về WIFI và có khả năng join mạng BLE MESH. Đối với các kit hiện tại, kit ESP32 hỗ trợ cả hai giao thức này nên lựa chọn ESP32 sẽ là lựa chọn vật lý.



Hình 3. 3 Chip ESP32

#### **Sơ đồ khối thiết bị**



Hình 3. 4 Sơ đồ khối thiết bị switch

Các khối trong sơ đồ khối thiết bị bao gồm:

* Nguồn cấp: Nhận nguồn điện 220VAC từ nguồn điện lưới
* Khối hạ áp:
  + Module Hlink 220VAC-5VDC: Chuyển đổi nguồn 220VAC sang nguồn 5VDC 0.6A
  + Hạ áp từ 5VDC sang 3.3VDC (sử dụng IC AMS1117-3.3V)
* Khối điều khiển trung tâm (Module ESP-WROOM-32):
  + Nhận tín hiệu từ Switch
  + Xuất tín hiệu đầu ra điều khiển LED
  + Xuất tín hiệu đầu ra điều khiển Relay
* RELAY:
  + Điều khiển ON/OFF nguồn điện đầu ra 220VAC bằng tín hiệu điều khiển

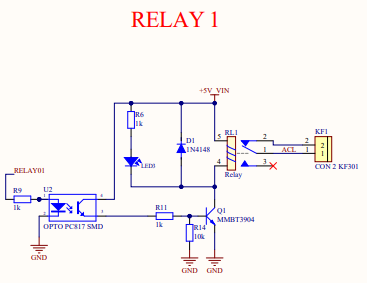
#### **Thiết kế mạch nguyên lý**

A diagram of a power source

Description automatically generated

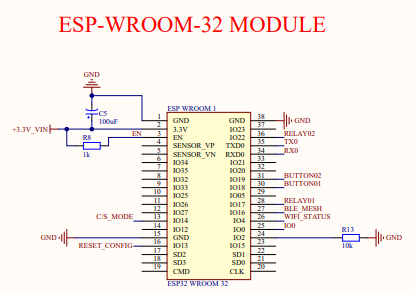
Hình 3. 5 Khối nguồn

* Đầu vào: 220VAC, bao gồm 3 dây là dây nóng( ACN), dây lạnh (ACL) và dây đất (Ground\_earth) . Chân GND Earth nối qua trở siêu dẫn tới chân GND. Mục đích là để hấp thụ dòng dò trên mạch.
* Cầu chì F1 sẽ đứt khi mạch có sự cố chập cháy, khi đó mạch được ngắt điện. Tiếp tới là MOV1 là tụ chống sét 10S471K để bảo vệ mạch chống lại sự biến thiên đột ngột về điện áp.
* Tiếp tới là module Hilink để biến đổi điện áp 220 VAC sang 5VDC.
* Đầu ra 5VDC đưa vào mạch hạ áp 3.3VDC sử dụng IC LDO AMS1117 – 3.3V . Tụ C1 để lọc nguồn đầu vào **(Tụ Bypass),** C2 và C3 **(Tụ de-couple link ? )** sử dụng lọc nguồn đầu ra.

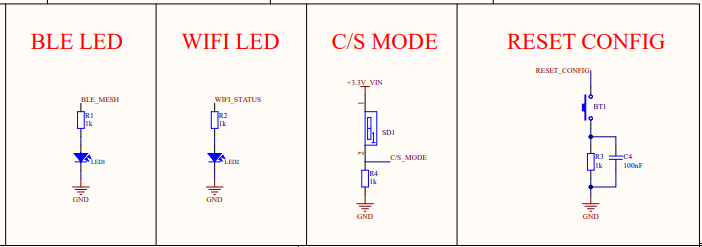


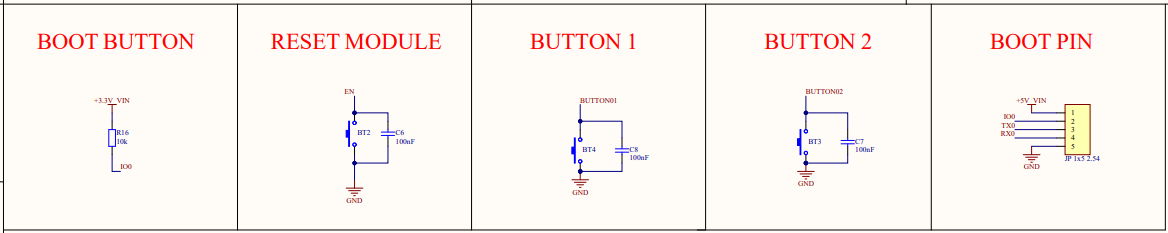
Hình 3. 6 Khối Relay

* Khi có tín hiệu mức cao tại RELAY01, thì opto cho phép dòng điện đi qua tại chân 4 và 3, khi đó transistor Q1 mở cho phép dòng điện đi qua. Khi đó RELAY có mức điện áp 5V và Relay mở.



Hình 3. 7 Khối ESP32 và các chân kết nối





Hình 3. 8 Các khối khác

## **Thiết kế phần mềm nhúng**

### **Mục tiêu thiết kế**

#### **Thiết bị gateway**

Chạy multiprotocol, bao gồm WIFI và BLE MESH

* Nhận bản tin từ internet và gửi cho các thiết bị trong mạng và ngược lại
* Thiết bị phải có sự ổn định, có khả năng tự động kết nối lại wifi và rejoin vào mạng BLE MESH khi bị mất nguồn
* Có hai led riêng biệt gồm led báo hiệu đã kết nối wifi và led báo hiệu đã tham gia vào mạng BLE MESH cho người dùng
* Thiết lập network key và Application key cho thiết bị để chống lại các cuộc tấn công mạng từ bên ngoài

#### **Thiết bị switch**

* Hệ thống được thiết kế để có khả năng mở rộng phạm vi mạng BLE MESH
* Mạng phải có tính bảo mật cao để tránh các thiết bị bên ngoài xâm nhập, việc này được thực hiện thông qua network key và application key, bảo mật bởi 32 ký tự hexa, mỗi bản tin sẽ luôn có chứa các chỉ số bảo mật trong đó để tránh các relay attack ( nghe bản tin và phát lại bản tin đó)
* Độ trễ bản tin nhỏ hơn 3 giây đối với khoảng cách 30 mét
* Bản tin từ gateway ( client device ) phải được chuyển tới mọi thiết bị đích cần nhận trong mạng
* Các thiết bị phải có khả năng rejoin mạng BLE MESH sau khi gặp sự cố đột ngột như mất điện
* Các thiết bị trong hệ thống đều phải có nút reset trong trường hợp người dùng muốn cấu hình lại thiết bị. Thiết bị sẽ đi vào trạng thái reset và thiết lập lại từ đầu khi người dùng giữ nút reset trong 5 giây
* Khi thiết bị join vào một mạng BLE MESH cụ thể, cần có led báo hiệu cho người dùng
* Mỗi thiết bị “ switch” sẽ bao gồm hai switch để điều khiển bật tắt hai đèn riêng biệt
* Độ trễ khi bật tắt thủ công phải là nhỏ nhất, được thực hiện thông qua ngắt

### **Triển khai thiết kế phần mềm nhúng**

#### **Thiết bị gateway**

##### **Kết nối WIFI cho gateway**

Việc kết nối WIFI cho thiết bị gateway ( client device ) là điều bắt buộc đối với thiết bị gateway. Trong hệ thống điều khiển chiếu sáng, khi MCU được kết nối WIFI, từ đó gateway có khả năng nhận lệnh của người dùng thông qua Internet. Để thực hiện kết nối WIFI cho thiết bị gateway, người dùng cần cung cấp SSID và Password của router trong nhà để thiết bị xác lập kết nối.

Đối với ESP32, nhà sản xuất họ đã cung cấp ứng dụng softAP Prov. Khi truy cập ứng dụng này, người dùng sẽ bắt wifi Accesspoint do thiết bị khởi tạo và kết nối gateway với router trong nhà. Người dùng nhận dạng WIFI Accesspoint do MCU ESP32 khởi tạo thông qua SSID : PROV\_ ở đầu và nhập mật khẩu kết nối mà thiết bị cung cấp.

**Dưới đây là lưu đồ thuật toán cho việc kết nối WIFI cho thiết bị:**

A diagram of a flowchart

Description automatically generated

Hình 3. 9 Lưu đồ thuật toán kết nối WIFI cho gateway

**Giải thích lưu đồ thuật toán:**

Sau khi cấp nguồn cho thiết bị gateway, bản thân gateway sẽ phát ra wifi AP, lúc này người dùng cần tìm wifi có SSID có dạng PROV ở đầu và thực hiện kết nối ( không yêu cầu nhập mật khẩu). Khi kết nối thành công, người dùng được cấp phát địa chỉ IP và khi này người dùng cần nhập bảo mật là một khóa phiên trước khi nhập SSID và password của router cho gateway, nếu nhập sai sẽ yêu cầu nhập lại. Sau khi người dùng nhập đúng, người dùng sẽ lựa chọn router có RSSI khỏe nhất, theo mong muốn của người dùng và nhập SSID và password của router. Trong trường hợp người dùng nhập sai ở bước này, gateway sẽ không kết nối wifi và liên tục nhấp nháy đèn báo wifi cho biết đã nhập sai, người dùng thực hiện reset thiết bị và làm lại từ đầu. Trong trường hợp nhập đúng, trên app hiển thị kết nối thành công và thiết bị gateway sẽ chuyển sang chế độ station. Sau khi được kết nối WIFI thành công, gateway sẽ tự động kết nối lên cloud và sẵn sàng nhận lệnh từ cloud server.

Những hình ảnh dưới đây là các bước thực tế người dùng thực hiện.

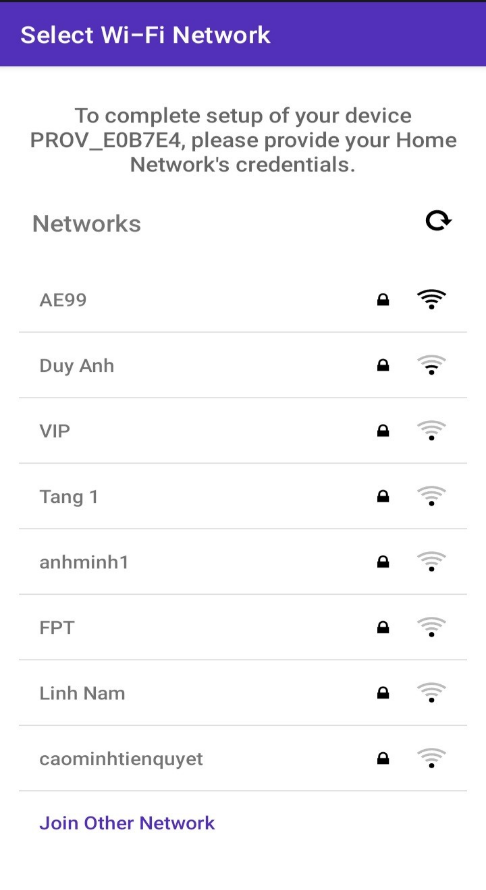
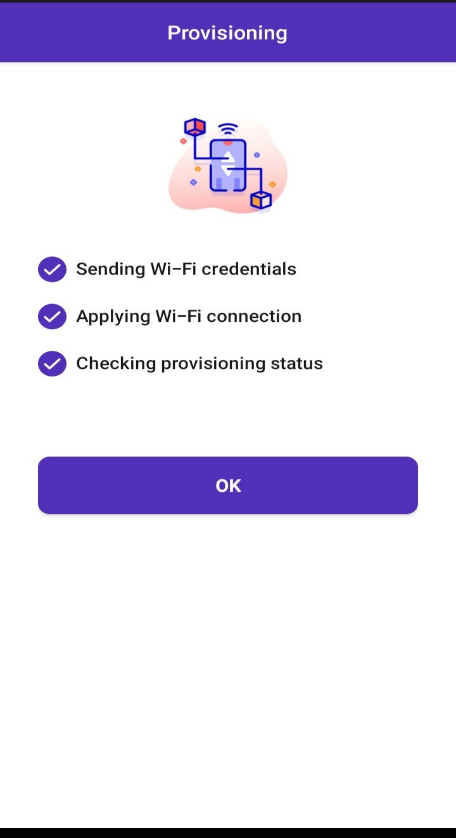
A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Tìm wifi có đầu SSID dạng PROV\_

Thực hiện connect với wifi ap

A screenshot of a phone

Description automatically generated

Phản hồi trạng thái kết nối

Lựa chọn wifi mà gateway sẽ kết nối

Nhập khóa phiên

##### **Thực hiện provisioning BLE MESH cho thiết bị**

Việc provisioning cho thiết bị là việc rất quan trọng. Người dùng sẽ thiết lập hai lớp khóa bảo mật bao gồm network key và application key theo sự hướng dẫn của nhà sản xuất. BLE MESH trên thiết bị gateway sẽ được thiết kế chạy song song liên tục với WIFI provisioning kể từ thời điểm được cấp nguồn. BLE Mesh sẽ sử dụng eventloop để chạy, nó sẽ khởi tạo các hàm callback liên quan tới provisioning, callback khi người dùng thiết lập cho thiết bị và hàm callback liên quan tới việc lệnh được gửi đi.

Dưới đây là lưu đồ thuật toán:

A diagram of a flowchart

Description automatically generated

Hình 3. 10 Lưu đồ thuật toán khởi tạo mạng BLE MESH

Khi cấp nguồn, thiết bị sẽ khởi tạo các lớp của BLE MESH và đăng ký các hàm ngắt sự kiện. sự kiện xảy ra khi có các opcode được gửi tới. Trước tiên người dùng sẽ dùng smartphone để scan thiết bị gateway, khi phát hiện ra sẽ thực hiện kết nối, khi này thiết bị gateway gửi tới cho người dùng tên của nó, lúc này người dùng sẽ lựa chọn network key và đặt địa chỉ theo yêu cầu nhà sản xuất và thực hiện provisioning. Sau khi thiết bị cho phép và connect với điện thoại người dùng, người dùng sẽ thêm các application Key cụ thể cho thiết bị.

Dưới đây là các bước người dùng thiết lập trên smartphone:

A screenshot of a phone

Description automatically generatedA black background with a black square

Description automatically generated with medium confidence

Lựa chọn application key và address

Người dùng Scan gateaway

A screenshot of a computer

Description automatically generated

A screen shot of a phone

Description automatically generated

Gateway hiển thị trên list các thiết bị đang kết nối

Chọn một trong các application key

A screenshot of a phone

Description automatically generatedA screenshot of a black screen

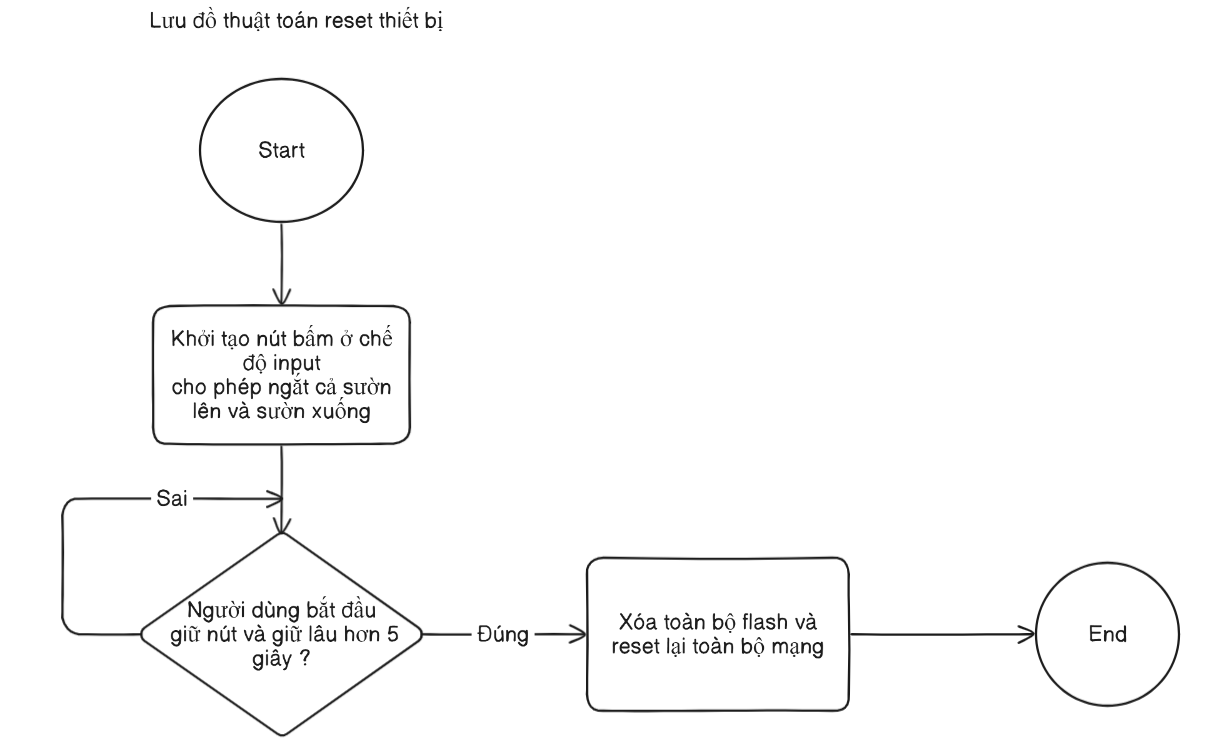
Description automatically generated

Chính thức nhúng appkey cho thiết bị

Lựa chọn Generic

##### **Thực hiện reset cấu hình lại thiết bị khi cần thiết**

Trong một số trường hợp, ví dụ như người dùng nhập sai cấu hình WiFi ban đầu hoặc mong muốn thay đổi cấu hình như đặt lại địa chỉ cho mạng BLE MESH người dùng sẽ sử dụng nút ấn để reset toàn mạng. Khi người dùng giữ nút ấn này 5 giây trở lên và nhả tay ra thì cả thiết bị sẽ được reset. Dưới đây là lưu đồ thuật toán cho nút bấm reset:



Hình 3. 11 Lưu đồ thuật toán để reset toàn mạng

Khi bắt đầu cấp nguồn, nút bấm reset sẽ được khởi tạo và nó sẽ hoạt động dựa trên phương thức giao tiếp ngắt. Ở đây nút bấm được thiết kế dựa trên ngắt hai sườn xung cả sườn lên và sườn xuống. Việc quyết định thời gian ngắt dựa trên số tick đếm được giữa hai lần thực hiện ngắt. Khi người dùng bắt đầu ấn, khởi tạo một biến lưu trữ giá trị tick vào thời điểm ấy, đối với esp32, một tick là 10ms. Sau một khoảng thời gian, khi người dùng nhả phím, MCU phát hiện xung sườn lên và sẽ có một biến nữa lưu số tick của hệ thông tại thời điểm nhả phím. Chỉ cần lấy số tick thời điểm sau trừ thời điểm trước và nhân với thời gian của một tick là có số thời gian nhấn là bao nhiêu ms. Từ đó ra lệnh điều khiển có reset toàn hệ thống hay không.

##### **Phương thức để gateway gửi bản tin tới thiết bị đích và điều khiển nhóm**

Ở đây để gateway ( client device ) để có thể gửi sang các server device sẽ cần xác định những trường sau:

* Opcode: Là mã lệnh mà client gửi sang server, thường là phân biệt đây là lệnh yêu cầu điều khiển hoặc lệnh yêu cầu lấy dữ liệu.
* Model: Đại diện cho ứng dụng
* Net\_index: Là giá trị netkey người dùng cấu hình cho thiết bị lúc ban đầu
* Key\_index: Là giá trị appkey người dùng cấu hình
* TTL: Số lần sống của một message
* Message timeout: Là thời gian cho phép tối đa của một bản tin gửi đi và nhận phản hồi
* Message role: Vai trò của node
* Value: Giá trị gửi
* Transaction ID: Số lần giao dịch

Đây là tất cả trường mà client device sẽ gửi sang server device, các trường này sẽ được chuyển từ application layer xuống dần các lớp dưới như đã đề cập ở phần lý thuyết và gửi sang cho server device.

##### **Bóc tách bản tin gửi từ cloud để điều khiển thiết bị đích và điều khiển nhóm**

Khi có kết nối WIFI, gateway sẽ kết nối với broker và đăng ký bản tin điều khiển. Khi này, người dùng gửi lệnh điều khiển từ cloud thì gateway sẽ nhận hai trường bao gồm address và value của thiết bị muốn điều khiển . Khi này gateway sẽ nhận hai trường này ở dạng string và thực hiện tách hai để lấy giá trị của trường . Ví dụ client gửi

{

“address” : “ 0x0002” ,

“value” : “0”

}

Khi này, gateway thực hiện xử lý và tách được hai trường và lấy giá trị 16 bit là 0x0002 và trường value , giá trị là 0. Khi này hai giá trị này sẽ được gán vào phần address và value của bản tin gửi cho server device và thực hiện gửi đi.

##### **Lưu đồ thuật toán của gateway**

###### **Hàm xử lý việc gửi lệnh khi gateway nhận lệnh từ cloud:**

Đây là hàm thực hiện việc nhận lệnh từ cloud và sau đó phân tách trường address và giá trị điều khiển để thực hiện việc gửi bản tin cho server device. Bên dưới là lưu đồ thuật toán hàm xử lý:

A diagram of a server

Description automatically generated

Hình 3. 12 Lưu đồ thuật toán của gateway gửi lệnh cho server device

#### **Thiết bị Switch**

Đây là server device và quá trình provisioning và reset toàn mạng y hệt như thiết bị gateway nên sẽ không được đề cập lại ở phần này.

##### **Phương thức để thiết bị nhận bản tin từ gateway và phản hồi bản tin**

Để thiết bị switch có thể nhận bản tin của gateway, thiết bị sẽ thực hiện kiểm tra netkey, appkey, kiểm tra giá trị địa chỉ đích thuộc về thiết bị không và các thông số khác. Sau khi kiểm tra thành công, thiết bị kiểm tra opcode nhận được, Nếu nhận được mã lệnh tương ứng, thiết bị switch sẽ lưu lại payload ở phía sau opcode và thực hiện những công việc cần thiết . Trong ứng dụng này, giá trị nhận được sẽ có dạng 0 và 1, để điều khiển đèn tương ứng. Việc thiết bị switch phản hồi lại là rất quan trọng vì giá trị này sẽ quyết định trạng thái thiết bị trên cloud là đúng đắn. Không phải chỉ đơn thuần là phản hồi dựa trên giá trị điều khiển. Do đó, trước khi giá trị phản hồi được gửi tới gateway, thiết bị gateway sẽ đọc mức điện áp phản hồi của relay thuộc mức cao hay thấp và phản hồi giá trị đó cho server và từ đó cloud có thể thực hiện update trạng thái của thiết bị switch đúng đắn.

##### **Điều khiển thiết bị switch bằng smartphone thông qua BLE**

Ngoài việc các thiết bị trong nhà có thể được điều khiển từ xa thông qua internet thì trong trường hợp người dùng ở trong nhà thì họ hoàn toàn có thể điều khiển thiết bị thông qua BLE trên smartphone của họ. Khi này smart home của họ có thể coi là một Node được join vào mạng và cũng thực hiện gửi lệnh điều khiển hoặc lệnh yêu cầu đọc giá trị. Sau khi gửi, giá trị phản hồi sẽ phản hồi lại cho người dùng và hiển thị trên smartphone.

##### **Điều khiển đèn bằng cách bấm nút thủ công**

Ngoài việc điều khiển đèn không dây, người dùng hoàn toàn có thể điều khiển đèn thông qua nút bấm thông thường. Điều này là cần thiết để phòng trường hợp các thiết bị có vấn đề về kết nối. Để hạn chế tối đa về độ trễ, việc giao tiếp nút nấm sẽ sử dụng phương thức ngắt.

Trên thiết bị được thiết kế với hai nút cơ và điều khiển hai đèn riêng biệt. Dưới đây là lưu đồ thuật toán :

A diagram of a flowchart

Description automatically generated

Hình 3. 13 Lưu đồ thuật toán khi điều khiển bằng nút ấn bình thường

Khi MCU được cấp nguồn, các input và output được khởi tạo. Cứ khi nào nút được bấm, hàm ngắt được gọi ra và thực hiện function ngắt tương ứng.

##### **Lưu đồ thuật toán của thiết bị switch**

Dưới đây là thuật toán của thiết bị switch:

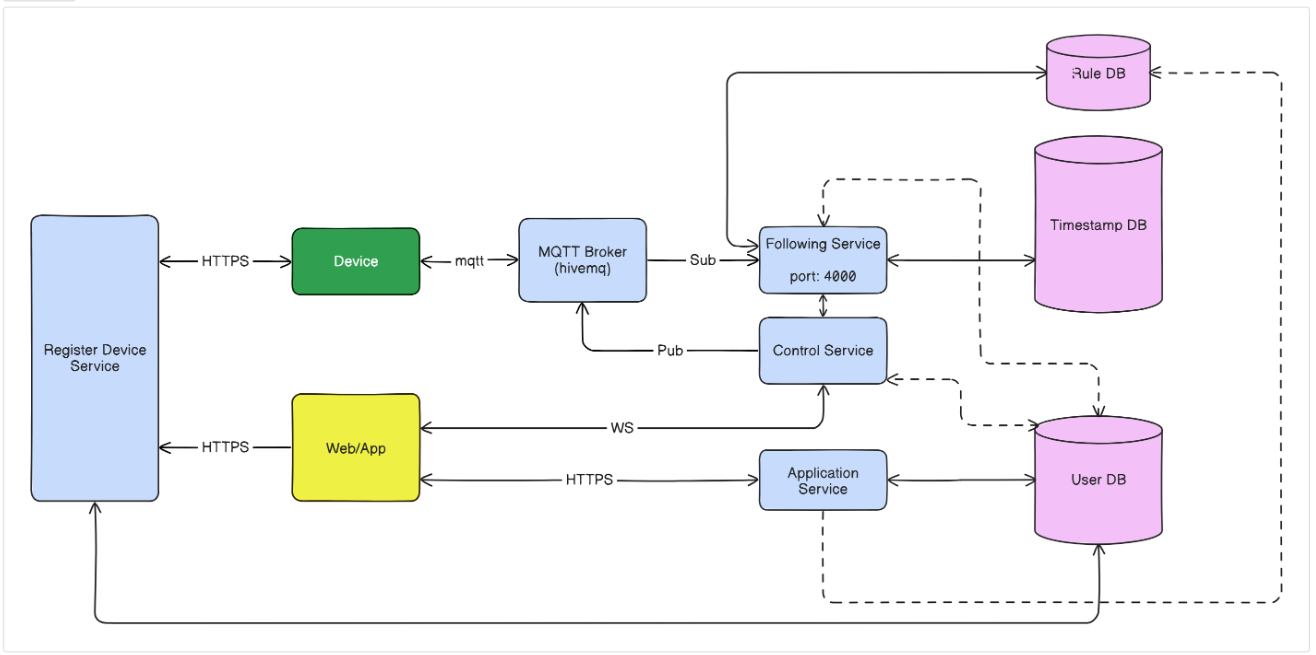
A diagram of a flowchart

Description automatically generated

Hình 3. 14 Lưu đồ thuật toán của thiết bị switch

## **Thiết kế cloud server**

### **Tổng quan Cloud Server**



Hình 3. 15 Sơ đồ khối của cloud

Đây là sơ đồ khối IoT cloud được sử dụng để tương tác người dùng và thiết bị của họ thông qua API. Cloud được xác định có 3 service chính:

Một là Application service, là dịch vụ để người dùng có thể đăng ký, đăng nhập tài khoản của họ và quản lý các thiết bị trong gia đình họ. Người dùng sẽ sử dụng web/app để gửi các yêu cầu tới dịch vụ, mọi thông tin người dùng cũng như thiết bị của họ sẽ luôn được lưu giữ trên cơ sở dữ liệu. Web/App sẽ giao tiếp với Application service thông qua giao thức HTTPS.

Hai là Control service, được sử dụng để người dùng tương tác, điều khiển thiết bị trong nhà thông qua web app. Ví dụ khi muốn thay đổi trạng thái của bóng đèn, người dùng sẽ gửi request tới control service, dịch vụ này sẽ xử lý và gửi một lệnh publish tới thiết bị thông qua broker, Gateway khi nhận được bản tin điều khiển từ cloud sẽ phân tích bản tin và gửi lệnh điều khiển tới thiết bị đích.

Ba là Following service, là dịch vụ được sử dụng để cập nhật trạng thái thiết bị bị thay đổi. Ví dụ, khi trạng thái của một bóng đèn bị thay đổi, ngay lập tức server device sẽ gửi bản tin cho gateway. Gateway sẽ có nhiệm vụ xử lý bản tin đó và publish lên cloud, khi này người dùng sẽ thấy trạng thái thiết bị được cập nhật.

Đối với database, hệ thống sử dụng mongoDB, MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

### **Application Service**

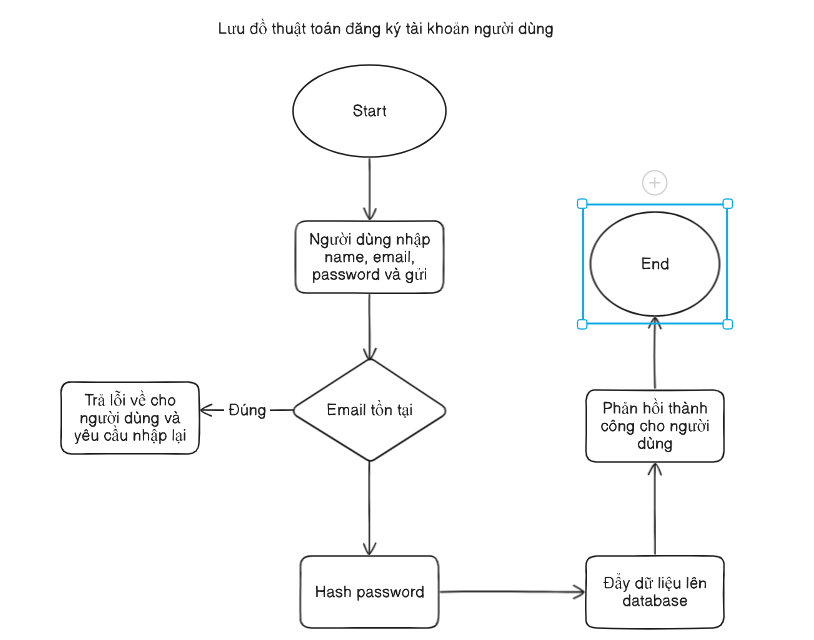
Như đề cập ở 4.1, application service là dịch vụ để người dùng có thể đăng ký tài khoản người dùng và quản lý các thiết bị trong gia đình họ. Dưới đây là các thành phần có trong service này:

* Đăng ký tài khoản người dùng
* Đăng nhập tài khoản người dùng
* Bảo vệ các url kiểm soát thiết bị
* Đăng ký thiết bị
* Xem toàn bộ thiết bị
* Xem chi tiết một thiết bị
* Xóa thiết bị

#### **Đăng ký tài khoản người dùng**

Trên cơ sở dữ liệu cần định nghĩa model cho người dùng để khi người dùng nhập thông tin, các thông tin đó sẽ được lưu trên database. Model của người dùng sẽ bao gồm những trường name, email, password, các device và network\_key, trong đó khi thực hiện đăng ký người dùng, người dùng sẽ được yêu cầu nhập name, email và password. Người dùng cần nhập url … /v1/api/access/user/signup để đi tới trang đăng ký, sau khi người dùng nhập đủ 3 trường name, email, password thì khi đó một req sẽ được gửi tới cloud để xử lý và phản hồi lại cho người dùng tên đã đăng ký để họ biết đã đăng ký thành công.

Dưới đây là lưu đồ thuật toán:



Hình 3. 16 Lưu đồ thuật toán người dùng đăng ký tài khoản

Trong lưu đồ thuật toán này, có một phần là hash password, mục đích của việc là là để bảo vệ mật khẩu mật khẩu cho người dùng, từ một dãy ký tự mà người dùng nhập, nó sẽ trở thành một dãy số random được biểu diễn dạng hexa và sau đó mới update lên database.

#### **Người dùng thực hiện đăng nhập**

Người dùng cần nhập url … /v1/api/access/user/login để đi tới trang đăng nhập, sau khi người dùng nhập đủ 2 trường email, password thì khi đó một req sẽ được gửi tới cloud để xử lý và phản hồi lại cho người dùng về trạng thái đăng nhập. Đây là khâu để người dùng được cấp quyền tiến vào các url kiểm soát thiết bị.

Dưới đây là lưu đồ thuật toán của cloud để xử lý khi người dùng thực hiện đăng nhập:

A diagram of a flowchart

Description automatically generated

Hình 3. 17 Lưu đồ thuật toán người dùng đăng nhập trên DB

Trong lưu đồ thuật toán này có một mục là model token, đây là model mà token sẽ được update lên database, token này được tham chiếu tới user tương ứng.

#### **Bảo vệ Url Kiểm soát thiết bị**

Việc bảo vệ các url được coi là vấn đề tối quan trọng đối với bất kỳ một hệ thống IoT nào vì các thiết bị trong nhà chỉ được kiểm soát bởi người dùng. Mọi người ngoài đều không được cấp phép để tham gia vào các url thiết bị người dùng. Để thực hiện việc bảo mật này, ở phần login ở trên là phần cấp token cho người dùng. Người dùng ơhair dùng token của mình thì mới truy cập được vào quản lý thiết bị.

Bên dưới là lưu đồ thuật toán bảo vệ url kiểm soát thiết bị:

A diagram of a diagram

Description automatically generated

Hình 3. 18 Lưu đồ thuật toán kiểm tra quyền truy cập url

Ở đây, có một phần là giải mã token. Khi truy cập vào bất kỳ url nào, người dùng sẽ cần phải có Client ID và token. Bản chất token khi giải mã ra nó chính là ClientID của người dùng. Vậy Sẽ cần kiểm tra phần giải mã token ấy với clientID xem token có hợp lệ không. Ngoài ra còn một phần là kiểm tra token đăng nhập vào có phải là token mới nhất trên database thuộc về user không. Nếu vượt qua hai lớp kiểm tra đó người dùng sẽ được đi vào url thuộc về mình.

#### **Đăng ký thiết bị**

Sau khi thực hiện đăng nhập, người dùng sẽ cần nhập url …/v1/api/device/claim để thực hiện đăng ký thiết bị. Một model của device sẽ bao gồm name, deviceID, type, value, tham chiếu tới người dùng tương ứng. Khi người dùng đăng ký, người dùng sẽ cần nhập trường deviceID và đi cùng với đó là userID đã được đặt sẵn khi mà người dùng thực hiện đăng nhập. Device ID là do người làm ra hệ thống quy định.

Dưới đây là lưu đồ thuật toán để cloud có thể xử lý việc người dùng đăng ký thiết bị:

A diagram of a diagram

Description automatically generated

Hình 3. 19 Lưu đồ thuật toán cloud xử lý khi người dùng đăng ký thiết bị

#### **Xem toàn bộ thiết bị**

Đây là chức năng để người dùng có thể xem toàn bộ danh sách thiết bị hiện có của mình. Sau khi thực hiện đăng nhập, người dùng sẽ tới url …/v1/api/device/alldevice.

Trong model của người dùng sẽ định nghĩa trường device, là một mảng lưu trữ toàn bộ deviceID của các thiết bị. Nếu người dùng yêu cầu xem tất cả thiết bị, cloud sẽ cần phản hồi về cho người dùng mảng lưu trữ này. Dưới đây là lưu đồ thuật toán:

A diagram of a flowchart

Description automatically generated

Hình 3. 20 Lưu đồ thuật toán cloud xử lý khi người dùng yêu cầu thông tin tất cả thiết bị

#### **Xem chi tiết một thiết bị**

Đây là chức năng mà người dùng sẽ có thể xem chi tiết toàn bộ trạng thái của thiết bị, như tên, ID, trạng thái thiết bị, network key, lịch sử điều khiển, loại thiết bị. Khinày người dùng sẽ cần sử dụng một url có dạng … …/v1/api/device/onedevice/:id, ở đây ID là deviceID của thiết bị. Ví dụ người muốn đọc thông tin của thiết bị có ID là 0x0002 thì sẽ là … …/v1/api/device/onedevice/0x0002. Dưới đây là lưu đồ thuật toán :

A diagram of a flowchart

Description automatically generated

Hình 3. 21 Lưu đồ thuật toán cloud xử lý khi người dùng yêu cầu thông tin chi tiết của thiết bị

#### **Xóa một thiết bị**

Đây là chức năng người dùng xóa thiết bị ra khỏi danh sách thiết bị. Khi này, người dùng cần đi tới url …/v1/api/device/deletedevice/0x0002

## **Thiết kế giao diện người dùng**