

HỌC VIỆN CÔNG NGHỆ Bưu Chính Viễn Thông
KHOA CÔNG NGHỆ THÔNG TIN 1



BÁO CÁO CUỐI KÌ

Lớp : E22HTTT

Nhóm : 1

Chủ đề : Đèn chiếu sáng thông minh

Thành viên : **Đinh Trí Đạt - B22DCVT124**
Nguyễn Hoàng - B22DCVT209
Nguyễn Bá Dương - B22DCDT070
Trịnh Quang Bách - B22DCVT045

Contents

I. Tổng quan Đề tài.....	3
1. Giới thiệu Đề tài.....	3
2. Lý do chọn Đề tài.....	3
3. Mục tiêu Thành công.....	3
4. Phạm vi Triển khai.....	3
II. Xác định Yêu cầu Chức năng (Functional Requirements).....	4
III. Xác định Yêu cầu Phi Chức năng (Non-Functional Requirements).....	6
IV. Phân tích Ràng buộc (Constraints Analysis).....	7
V. Mô hình hóa Yêu cầu.....	7
VI. Công nghệ sử dụng.....	9
1. Giao thức Truyền tin: MQTT.....	9
2. Máy chủ MQTT Broker: HiveMQ Public Broker.....	10
3. Firebase Realtime Database.....	10
4. Nền tảng Ứng dụng: Flutter.....	10
5. OTA (Over-the-Air).....	11
6. WiFiManager.....	11
VII. Thiết bị sử dụng.....	11
1. Vi điều khiển (Chip): ESP32.....	11
2. Thiết bị Chấp hành: Module Relay 5V.....	12
3. Thiết bị mạng.....	13
4. Led giao thông.....	13
VIII. Phần mềm.....	14
1. Lập trình.....	14
2. Server + Backend MQTT.....	14
3. Firebase.....	14
4. App / Dashboard điều khiển.....	14
• Flutter.....	14
• React Native.....	14
• Web Dashboard (React/HTML/JS).....	14
IX. Kết luận.....	15

I. Tổng quan Đề tài

1. Giới thiệu Đề tài

- Tên Đề tài: Xây dựng Hệ thống Điều khiển và Quản lý Đèn Chiếu sáng Thông minh sử dụng IoT với ESP32, MQTT và Firebase Realtime Database, hỗ trợ phân quyền người dùng.
- Mục tiêu: Phát triển một giải pháp điều khiển chiếu sáng linh hoạt, cho phép người dùng giám sát trạng thái thiết bị và điều khiển từ xa, đồng thời tích hợp các tính năng bảo trì từ xa.
- Mức độ Dự án: Trung bình (Intermediate - Mức độ 2). Dự án này yêu cầu tích hợp nhiều giao thức (Wi-Fi, MQTT, BLE) và triển khai các tính năng nâng cao như OTA Firmware Update .

2. Lý do chọn Đề tài

- Đề tài đáp ứng nhu cầu thực tiễn về tăng tiện nghi trong môi trường sống hiện đại. Việc sử dụng giao thức nhẹ MQTT giúp tối ưu hiệu suất và băng thông, trong khi tính năng OTA đảm bảo khả năng mở rộng và bảo trì hệ thống dễ dàng sau này.

3. Mục tiêu Thành công

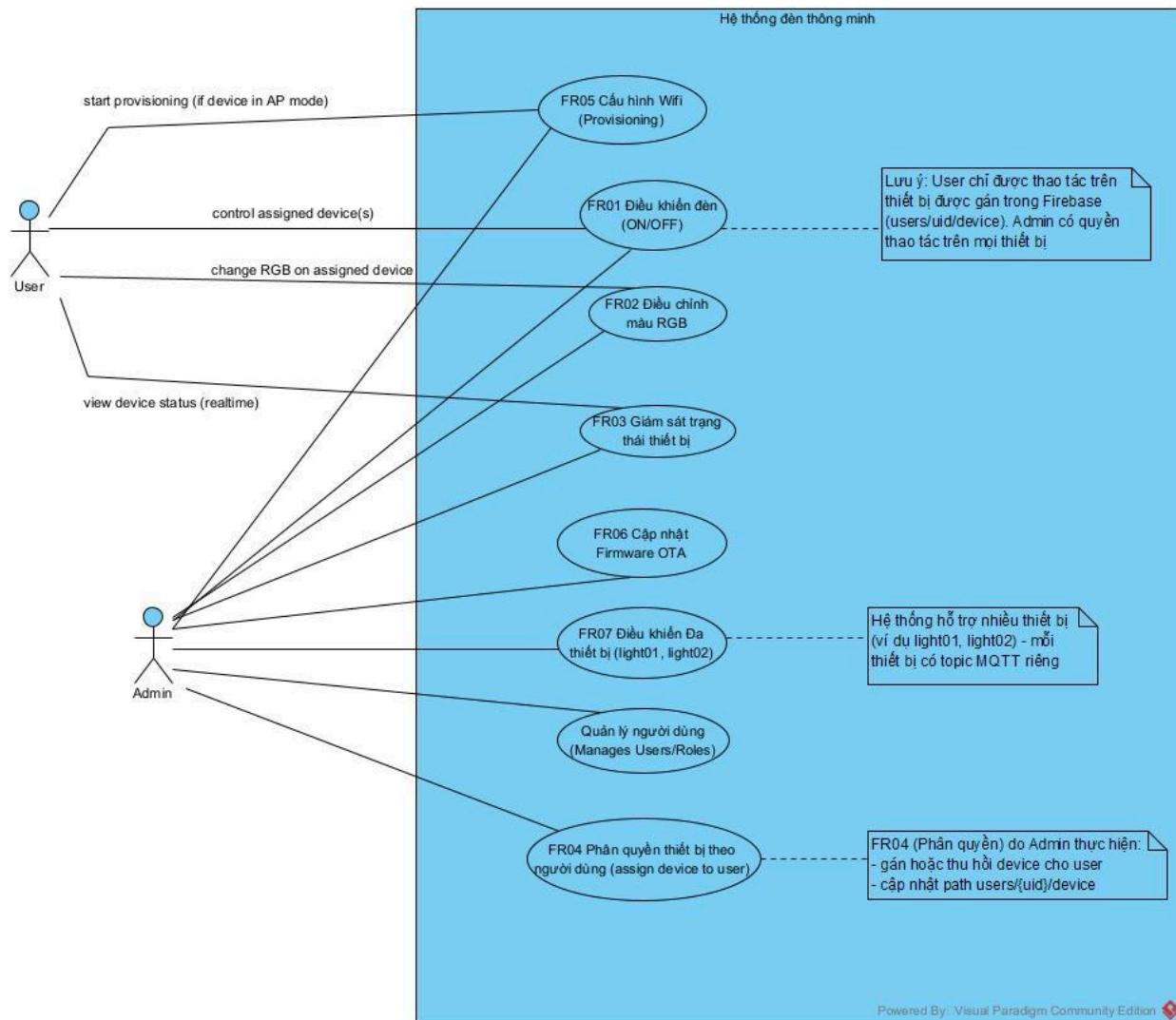
- Tốc độ phản hồi: Lệnh điều khiển truyền qua MQTT đến ESP32 và phản hồi trạng thái trên Firebase
- Độ tin cậy: Thiết bị tự động kết nối lại MQTT khi mất mạng và tiếp tục làm việc bình thường.
- Quản lý thiết bị theo user: Mỗi người dùng chỉ được phép điều khiển đúng một thiết bị đã được phân quyền trong Firebase.
- Cập nhật OTA thành công: Ứng dụng OTA thực hiện cập nhật firmware ESP32 mà không cần thao tác vật lý.

4. Phạm vi Triển khai

- Quy mô: Hệ thống điều khiển hai đèn: LED RGB và LED 3 chạy trên ESP32, mỗi đèn hoạt động như một thiết bị độc lập.
- Phân quyền: Firebase quản lý tài khoản → mỗi user tương ứng một thiết bị (light01 hoặc light02).
- Môi trường triển khai: Wi-Fi nội bộ (2.4GHz) và MQTT Cloud (HiveMQ).
- Ứng dụng điều khiển: Flutter App chạy trên Android/iOS.
- Phương thức giao tiếp: Giao thức MQTT (Pub/Sub) kết hợp với Firestore Database.

II. Xác định Yêu cầu Chức năng (Functional Requirements)

ID	Chức năng	Mô tả Chi tiết
FR01	Điều khiển Đèn (ON/OFF)	Người dùng gửi lệnh bật/tắt đến thiết bị tương ứng (light01 hoặc light02) thông qua ứng dụng Flutter. Lệnh được Publish qua MQTT và ESP32 xử lý ngay lập tức.
FR02	Điều chỉnh màu RGB	Ứng dụng Flutter cho phép người dùng chọn màu RGB, gửi lệnh MQTT với cấu trúc JSON {r, g, b} đến ESP32; ESP32 điều chỉnh LED RGB tương ứng.
FR03	Giám sát Trạng thái Thiết bị	ESP32 cập nhật trạng thái đèn (ON/OFF, giá trị RGB) lên Firebase Realtime Database; ứng dụng Flutter lắng nghe dữ liệu realtime và hiển thị cho người dùng.
FR04	Phân quyền Thiết bị theo Người dùng	Mỗi người dùng chỉ được thao tác trên thiết bị được gán cho họ trong Firebase (users/uid/device). App chỉ hiển thị đúng một đèn tương ứng user.
FR05	Cấu hình WiFi (Provisioning)	Thiết bị ESP32 sử dụng WiFiManager để tự tạo AP cấu hình khi không có mạng hoặc khi người dùng reset.
FR06	Cập nhật Firmware OTA	Cho phép cập nhật firmware mới qua WiFi mà không cần kết nối cáp USB (Over-the-Air Update).
FR07	Điều khiển Đa Thiết bị	Hệ thống hỗ trợ đồng thời 2 thiết bị độc lập (light01 và light02), mỗi thiết bị có một topic MQTT riêng.



III. Xác định Yêu cầu Phi Chức năng (Non-Functional Requirements)

Chất lượng	Yêu cầu	Công nghệ/Giải pháp đáp ứng
Hiệu năng	Độ trễ điều khiển < 2 giây giữa thời điểm gửi lệnh và thay đổi trạng thái trên ứng dụng.	MQTT (Pub/Sub): Đảm bảo giao tiếp nhẹ và nhanh chóng. Firebase Realtime Database: Dùng để đồng bộ trạng thái thời gian thực về ứng dụng (FR03).
Độ tin cậy	ESP32 phải tự động reconnect khi mất mạng và tiếp tục thực thi nhiệm vụ sau khi kết nối lại.	HiveMQ Cloud: Broker MQTT ổn định, đáng tin cậy. WiFiManager: Tự động cấu hình và kết nối lại WiFi (FR05).
Tính mở rộng	Hệ thống dễ dàng thêm thiết bị mới chỉ bằng cách thêm node Firebase và topic MQTT mới.	Cấu trúc dữ liệu linh hoạt (NoSQL): Dễ dàng thêm node light03, light04 mà không cần thay đổi cấu trúc cơ sở dữ liệu. MQTT Topic Tree: Dễ dàng thêm topic mới theo cấu trúc cây (ví dụ: ptit/smartlight/light03/control).
Chi phí thấp	Sử dụng ESP32 và LED RGB chi phí rẻ, dễ mua.	ESP32 DevKit, LED RGB: Lựa chọn phần cứng tiêu chuẩn, chi phí thấp.

Tính bảo mật	User chỉ điều khiển được thiết bị được gán, tránh điều khiển chéo.	Firebase Auth: Xác thực người dùng (User ID - uid). Firebase Rules: Thiết lập quy tắc chỉ cho phép uid đọc/ghi dữ liệu của thiết bị đã được gán (users/uid/device).
--------------	--	---

IV. Phân tích Ràng buộc (Constraints Analysis)

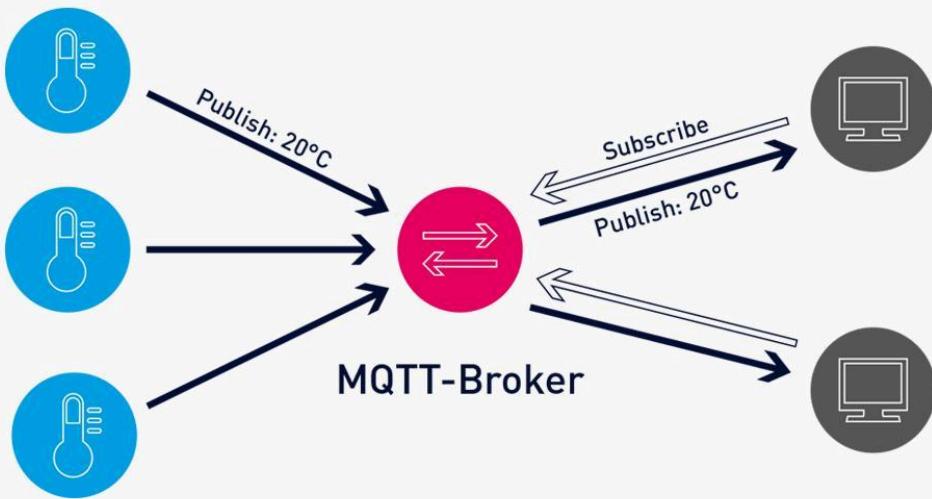
Ràng buộc	Chi tiết	Tác động
Phần cứng	Sử dụng ESP32 để điều khiển LED RGB.	Yêu cầu thiết kế code tối ưu, phân luồng điều khiển rõ ràng
Mạng WiFi	Hệ thống phụ thuộc WiFi và broker MQTT bên ngoài (HiveMQ Cloud).	Cần đảm bảo mạng ổn định để giảm độ trễ (Hiệu năng < 2s). Cần triển khai tính năng Tự động kết nối lại (Auto-reconnect) trong code Firmware (Độ tin cậy).
Phân quyền User	Mỗi user chỉ có thể điều khiển 1 thiết bị duy nhất (light01 hoặc light02).	Cấu trúc Database và ứng dụng phải tách biệt dữ liệu theo user (Tính bảo mật). Cần định nghĩa rõ Firebase Rules để ngăn chặn truy cập chéo (FR04).

V. Mô hình hóa Yêu cầu

Lớp	Thành phần	Chức năng
Perception – Tầng Cảm nhận	ESP32, LED RGB	Tầng gần gũi với phần cứng vật lý. Nhận lệnh MQTT (từ Tầng Mạng), điều khiển LED RGB (thực hiện chức năng vật lý), và gửi/cập nhật trạng thái lên Firebase (qua Tầng Mạng).
Network – Tầng Mạng	WiFi 2.4GHz, HiveMQ MQTT Broker	Đảm nhận nhiệm vụ giao tiếp và truyền tải dữ liệu. Đảm bảo luồng dữ liệu hai chiều: App -> HiveMQ -> ESP32.
Application – Tầng Ứng dụng	App Flutter + Firebase	Tầng tương tác trực tiếp với người dùng. Xử lý logic nghiệp vụ (Đăng nhập, Phân quyền), giao diện điều khiển đèn, và hiển thị trạng thái realtime (lắng nghe dữ liệu Firebase).

VI. Công nghệ sử dụng

1. Giao thức Truyền tin: MQTT



Tất tần tật về giao thức MQTT

- Lý do sử dụng: Là giao thức nhắn tin nhẹ nhất hiện nay, hoạt động theo mô hình Publish/Subscribe.
- Giá trị mang lại:
 - o Đảm bảo lệnh điều khiển và trạng thái được truyền tải gần như tức thời (độ trễ dưới 3 giây).
 - o Tiết kiệm băng thông và tài nguyên bộ nhớ cho ESP32.

2. Máy chủ MQTT Broker: HiveMQ Public Broker



- Lý do sử dụng: Sử dụng dịch vụ Public Broker (Máy chủ công cộng miễn phí).
- Giá trị mang lại:
 - o Giúp nhóm loại bỏ hoàn toàn khâu thiết lập, cài đặt và duy trì máy chủ (server) riêng.
 - o Cho phép các thành viên bắt đầu lập trình và kiểm thử ngay lập tức mà không bị phụ thuộc vào môi trường Backend phức tạp.

3. Firebase Realtime Database

- Lưu trữ trạng thái thiết bị.
- Hỗ trợ realtime cho app Flutter.
- Dùng Firebase Rules để phân quyền thiết bị theo user.

4. Nền tảng Ứng dụng: Flutter



- Lý do sử dụng: Cung cấp trải nghiệm người dùng mượt mà và khả năng tận dụng phần cứng tốt.

- Giao diện trực quan, đa nền tảng.
- Thư viện MQTT client mạnh mẽ.
- Tích hợp Firebase nhanh chóng.

5. OTA (Over-the-Air)

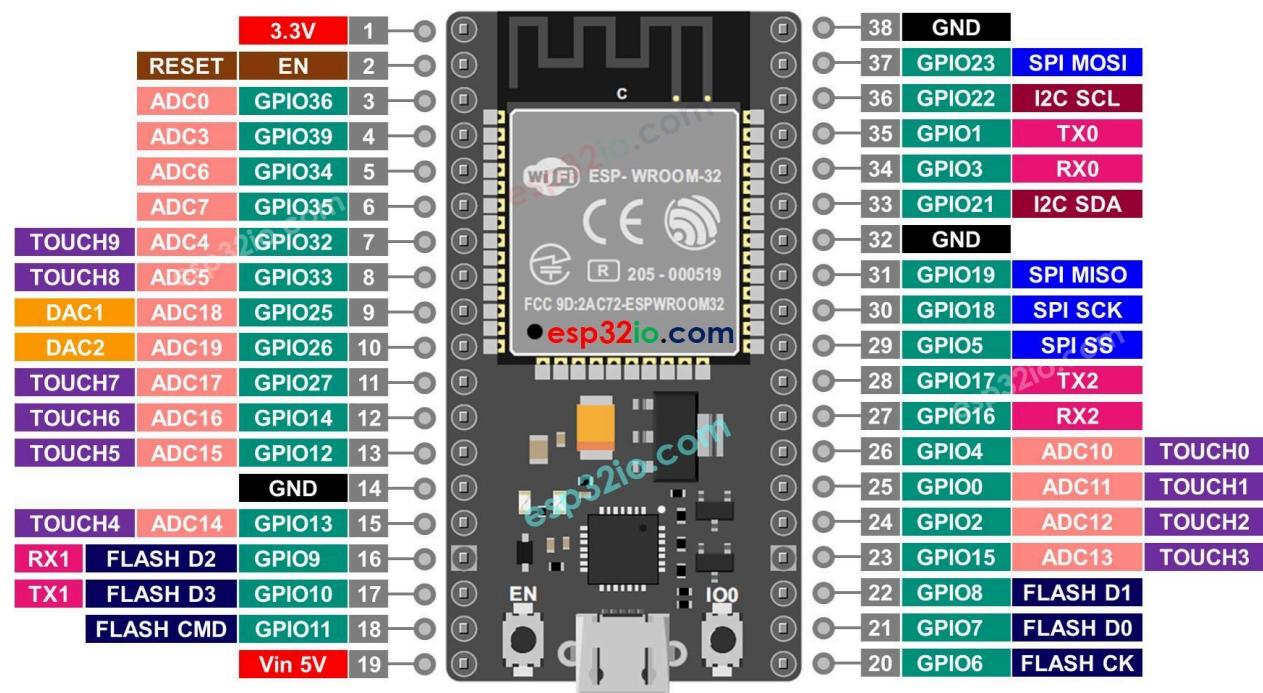
- Cập nhật firmware không cần cáp.
- Dùng HTTP hoặc MQTT-based OTA.

6. WiFiManager

- Cho phép thiết bị tự tạo AP để cấu hình WiFi khi cần.

VII. Thiết bị sử dụng

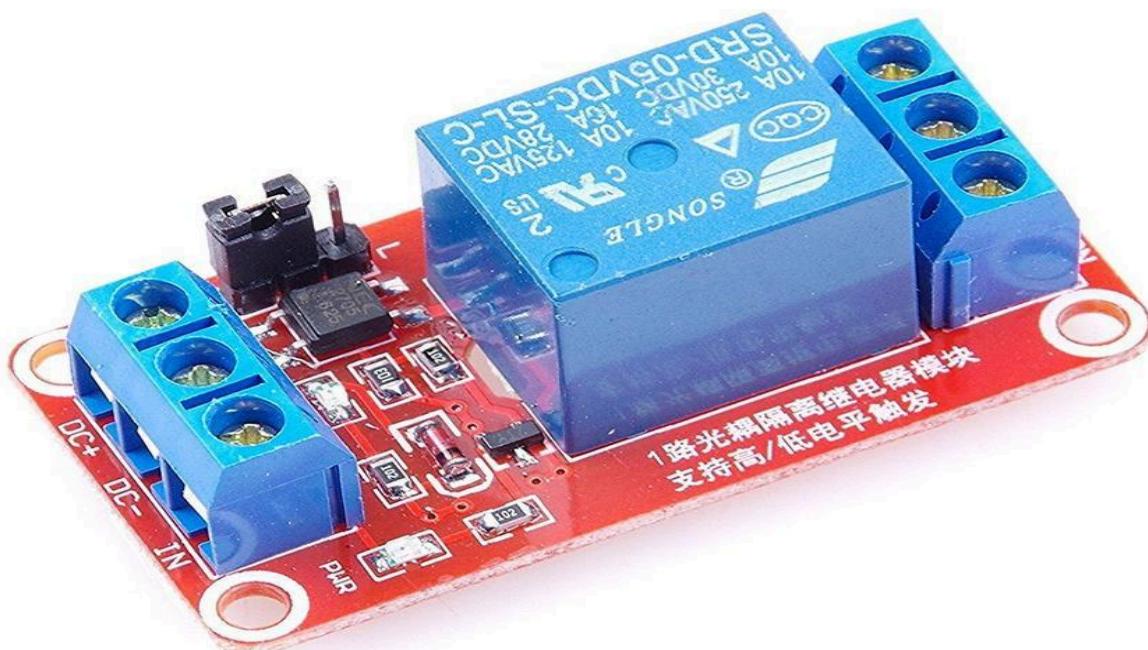
1. Vi điều khiển (Chip): ESP32



- Lý do sử dụng:

- o Chi phí thấp và thông dụng trên thị trường.
- o Tích hợp sẵn Wi-Fi để kết nối MQTT và Bluetooth (BLE) để sử dụng cho Provisioning (tính năng NICE-TO-HAVE).
- o Hỗ trợ lập trình dễ dàng qua môi trường Arduino, phù hợp với người mới bắt đầu.

2. Thiết bị Chấp hành: Module Relay 5V



- Lý do sử dụng:

- An toàn và Đơn giản: Relay 5V hoạt động với nguồn 5V, có thể dùng chung nguồn cấp từ cổng USB máy tính hoặc củ sạc điện thoại với ESP32.
- Chức năng: Thực hiện việc BẬT/TẮT nguồn điện cho đèn mô phỏng theo lệnh từ ESP32.

3. Thiết bị mạng

- Router WiFi (bắt buộc)



- Nếu triển khai thực tế → có thể cần WiFi riêng cho IoT để ổn định hơn

4. Led giao thông



5. Led RGB 3 màu



VIII. Phần mềm

1. Lập trình

- Arduino IDE
- Các thư viện:
 - PubSubClient (MQTT)
 - Firebase ESP Clients
 - WiFi.h

2. Server + Backend MQTT

- HiveMQ Cloud

3. Firebase

- Fire Database
- Firebase Authentication (để phân quyền user)

4. App / Dashboard điều khiển

- Flutter
- React Native
- Web Dashboard (React/HTML/JS)
- Ứng dụng cần tích hợp:
 - Điều khiển đèn (On/Off, chỉnh màu RGB)
 - Theo dõi trạng thái thời gian thực
 - Phân quyền người dùng:
 - Admin
 - User thường

IX. Kết luận

Trong dự án “Hệ thống Điều khiển và Quản lý Đèn Chiếu sáng Thông minh sử dụng IoT với ESP32, MQTT và Firebase Realtime Database”, nhóm đã xây dựng thành công một mô hình giúp điều khiển và giám sát đèn chiếu sáng theo thời gian thực thông qua Internet. Hệ thống sử dụng ESP32 làm bộ điều khiển trung tâm, giao tiếp với MQTT Broker để truyền – nhận dữ liệu nhanh và ổn định, đồng thời tích hợp Firebase Realtime Database nhằm lưu trữ trạng thái thiết bị, lịch sử hoạt động và phân quyền người dùng.

Ứng dụng điều khiển (mobile/web) cho phép người dùng bật/tắt đèn, thay đổi màu sắc đối với LED RGB, xem trạng thái thiết bị và theo dõi dữ liệu ngay lập tức. Việc phân quyền dựa trên Firebase Authentication giúp bảo đảm an toàn, phân chia rõ ràng vai trò Admin – User, hạn chế truy cập trái phép.

Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, tốc độ phản hồi nhanh, độ trễ thấp nhờ cơ chế publish/subscribe của MQTT. Mô hình có khả năng mở rộng dễ dàng, phù hợp để triển khai trong nhà thông minh, tòa nhà, văn phòng hoặc khu chiếu sáng công cộng.

Nhìn chung, đề tài đã đạt được mục tiêu đề ra, đáp ứng yêu cầu kỹ thuật, mang tính ứng dụng cao và thể hiện tiềm năng phát triển thành một giải pháp chiếu sáng thông minh hoàn chỉnh.

