|  |
| --- |
| ĐẠI HỌC BÁCH KHOA HÀ NỘI  **TRƯỜNG ĐIỆN-ĐIỆN TỬ**  logo_128  ĐỒ ÁN  **TỐT NGHIỆP ĐẠI HỌC**  **Đề tài:**  **ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TỬ THÔNG QUA WEBSERVER**  Sinh viên thực hiện: TRẦN MINH CHIẾN  Lớp ĐT09 – K63  Giảng viên hướng dẫn: TS. NGUYỄN QUANG MINH  Hà Nội, 3-2023 |
| ĐẠI HỌC BÁCH KHOA HÀ NỘI  **TRƯỜNG ĐIỆN-ĐIỆN TỬ**  logo_128  ĐỒ ÁN  **TỐT NGHIỆP ĐẠI HỌC**  **Đề tài:**  **ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TỬ THÔNG QUA WEBSERVER**  Sinh viên thực hiện: TRẦN MINH CHIẾN  Lớp ĐT09 – K63  Giảng viên hướng dẫn: TS. NGUYỄN QUANG MINH  Cán bộ phản biện:  Hà Nội, 3-2023 |

**ĐÁNH GIÁ QUYỂN ĐỒ ÁN TỐT NGHIỆP**

(Dùng cho giảng viên hướng dẫn)

Tên giảng viên đánh giá: NGUYỄN QUANG MINH

Họ và tên Sinh viên: TRẦN MINH CHIẾN MSSV: 20182387

Tên đồ án:

**Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:**

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết quyển đồ án (10)** | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | |
| 10a | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | |  | | | | |

***Nhận xét khác*** *(về thái độ và tinh thần làm việc của sinh viên)*

|  |  |
| --- | --- |
|  | Ngày: … / … / 20…  **Người nhận xét**  (Ký và ghi rõ họ tên) |

**ĐÁNH GIÁ QUYỂN ĐỒ ÁN TỐT NGHIỆP**

(Dùng cho cán bộ phản biện)

Giảng viên đánh giá:

Họ và tên sinh viên: TRẦN MINH CHIẾN MSSV: 20182387

Tên đồ án:

**Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:**

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết quyển đồ án (10)** | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | |
| 10a | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | |  | | | | |

***Nhận xét khác của cán bộ phản biện***

|  |  |
| --- | --- |
|  | Ngày: … / … / 20…  **Người nhận xét**  (Ký và ghi rõ họ tên) |

# LỜI NÓI ĐẦU

Nhờ sự phát triển mạnh mẽ của công nghệ đã giúp cho cuộc sống của con người ngày càng trở nên dễ dàng và thuận tiện. Một trong số các công nghệ thể hiện điều đó phải kể đến công nghệ IoT (Internet of Things). Điển hình nhất của công nghệ IoT này đó chính là việc ứng dụng trong những ngôi nhà thông minh gắn liền nhất với cuộc sống thực tế.

Những năm gần đây smart home không còn là điều xa lạ gì với người dân Việt Nam và có thể nói rằng đó là thành công lớn của IoT gây ra được một làn sóng công nghệ hoàn toàn mới và được tìm kiếm nhiều nhất trên mạng Internet. Giờ đây, việc bạn ra khỏi nhà mà quên không tắt các thiết bị điện như điều hòa, tivi, đèn điện,...Hay việc vào mùa đông sau khi tan làm về phải đợi 15-20 phút sau khi bật bình nóng lạnh mới có nước nóng để tắm đã không còn là vấn đề. Với hệ thống smart home, bạn có thể bật tắt các thiết bị điện trong nhà dù đang ở 1 nơi rất xa ngôi nhà của bạn, chỉ qua 1 lần chạm trên chiếc điện thoại của bạn hay 1 cú click chuột trên chiếc laptop. Điều này không chỉ giúp chọn bạn tiết kiệm thời gian, chi phí mà ở đó còn là sự yên tâm về mức độ an toàn cũng như sự hiện đại, sang trọng của cuộc sống mà do chính bản thân bạn tạo ra.

Em xin gửi lời cảm ơn chân thành và sâu sắc đến toàn bộ thầy cô của ngôi trường Đại học Bách Khoa Hà Nội, đặc biệt là những thầy cô trường Điện – Điện tử đã hết lòng dạy dỗ, truyền đạt kinh nghiệm quý báu cho em trong suốt 4 năm vừa qua. Em xin gửi lời biết ơn sâu sắc đến TS. Nguyễn Quang Minh là thầy giáo hướng dẫn trực tiếp đồ án tốt nghiệp cho em. Thầy là một người tận tâm, tận tình với sinh viên, thầy không chỉ định hướng cho em đồ án tốt nghiệp, thầy còn định hướng cho công việc và các vấn đề khác trong cuộc sống của em. Những chia sẻ của thầy là những kinh nghiệm quý báu là hành trang mà chúng em sẽ mang theo trên chặng đường hướng về tương lai phía trước.

# LỜI CAM ĐOAN

Họ và tên sinh viên: Trần Minh Chiến

Mã số sinh viên: 20182387

Điện thoại liên lạc: 0981.506.236

Email: chien.tm182387@sis.hust.edu.vn

Lớp: ĐT09-K63

Hệ đào tạo: Cử nhân kỹ thuật

Tôi – *Trần Minh Chiến* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Quang Minh*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 03 tháng 03 năm 2023*  Tác giả ĐATN |

**MỤC LỤC**

[LỜI NÓI ĐẦU iii](#_Toc128847820)

[LỜI CAM ĐOAN iv](#_Toc128847821)

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT i](#_Toc128847822)

[DANH MỤC HÌNH VẼ ii](#_Toc128847823)

[DANH MỤC BẢNG BIỂU iv](#_Toc128847824)

[TÓM TẮT ĐỒ ÁN v](#_Toc128847825)

[ABSTRACT vii](#_Toc128847826)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 1](#_Toc128847827)

[1.1 Đặt vấn đề 1](#_Toc128847828)

[1.1.1 Tiềm năng 2](#_Toc128847829)

[1.1.2 Thách thức: 2](#_Toc128847830)

[1.2 Mục tiêu và phạm vi đề tài 3](#_Toc128847831)

[1.2.1 Mục tiêu 3](#_Toc128847832)

[1.2.2 Phạm vi đề tài 3](#_Toc128847833)

[1.3 Định hướng giải pháp 4](#_Toc128847834)

[1.4 Bố cục đồ án 4](#_Toc128847835)

[CHƯƠNG 2. GIỚI THIỆU TỔNG QUAN VỀ CÁC LINH KIỆN SỬ DỤNG TRONG DỰ ÁN 6](#_Toc128847836)

[2.1 Tổng quan về ESP8266 và board ESP8266 NodeMCU 6](#_Toc128847837)

[2.1.1 ESP8266 6](#_Toc128847838)

[2.1.2 Board phát triển ESP8266 NodeMCU 8](#_Toc128847839)

[2.2 Giớithiệu các module sử dụng trong dự án 9](#_Toc128847840)

[2.2.1 Module cảm biến nhiệt độ, độ ẩm (DHT11) 9](#_Toc128847841)

[2.2.2 Module phát sóng hồng ngoại (KY-005 Infrared Transmitter) 10](#_Toc128847842)

[CHƯƠNG 3. THIẾT KẾ HỆ THỐNG 13](#_Toc128847843)

[3.1 Tổng quan hệ thống 13](#_Toc128847844)

[3.2 Chức năng và sơ đồ khối 14](#_Toc128847845)

[3.2.1 Chức năng của toàn hệ thống 14](#_Toc128847846)

[3.2.2 Sơ đồ khối 14](#_Toc128847847)

[3.2.3 Mạch nguyên lý 15](#_Toc128847848)

[3.3 Mô phỏng, mạch in và sản phẩm thực tế 17](#_Toc128847849)

[3.3.1 Mô phỏng 17](#_Toc128847850)

[3.3.2 Mạch in 17](#_Toc128847851)

[3.3.3 Sản phẩm thực tế 19](#_Toc128847852)

[3.4 Lập trình cho ESP 19](#_Toc128847853)

[3.4.1 Giới thiệu tổng quan về Arduino IDE 20](#_Toc128847854)

[3.4.2 Mã nguồn cho ESP 20](#_Toc128847855)

[3.5 Thiết kế phần giao diện điều khiển (web-client) 24](#_Toc128847856)

[3.5.1 Tổng quan về giao diện điều khiển 25](#_Toc128847857)

[3.5.2 Phần lập trình xử lý các sự kiện ‘Click’ và ghép nối với API (phần code JavaScript) 25](#_Toc128847858)

[3.6 Thiết kế phần WebServer (phần back-end) 26](#_Toc128847859)

[3.7 Thiết kế CSDL 29](#_Toc128847860)

[CHƯƠNG 4. CÁC CÔNG NGHỆ SỬ DỤNG 33](#_Toc128847861)

[4.1 Công nghệ sử dụng cho phần cứng 33](#_Toc128847862)

[4.1.2 Các thư viện hỗ trợ lập trình cho ESP và các module sử dụng trong dự án 33](#_Toc128847863)

[4.2 Công nghệ sử dụng cho phần mềm 35](#_Toc128847864)

[4.2.1 Giới thiệu về mô hình MVC 36](#_Toc128847865)

[4.2.2 Framework “Microsoft.AspNetCore.App” và “Microsoft.NetCore.App” 37](#_Toc128847866)

[4.2.3 SQL Server 37](#_Toc128847867)

[4.2.4 Sử dụng “API SpeechRecognition” để nhận diện giọng nói tiếng Việt 38](#_Toc128847868)

[CHƯƠNG 5.KIỂM THỬ VÀ ĐÁNH GIÁ 39](#_Toc128847869)

[5.1 Kiểm thử phần giao diện điều khiển 39](#_Toc128847870)

[5.2 Kiểm thử phần mạch 39](#_Toc128847871)

[5.3 Đánh giá 43](#_Toc128847872)

[CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐỐNG GÓP NỔI BẬT 44](#_Toc128847873)

[6.1 Xây dựng một WebServer riêng để vận hành chức năng 44](#_Toc128847874)

[6.2 Xây dựng dự án theo mô hình MVC (model-view-controller) 44](#_Toc128847875)

[6.3 Xấy dựng chức năng điều khiển thiết bị bằng sóng hồng ngoại 44](#_Toc128847876)

[6.3.1 Giới thiệu cơ bản về sóng hồng ngoại 45](#_Toc128847877)

[6.3.2 Những lợi ích khi kết hợp sóng hồng ngoại với hệ thông smart home 46](#_Toc128847878)

[6.4 Tích hợp cảm biến nhiệt độ - độ ẩm cho sản phẩm 46](#_Toc128847879)

[6.5 Tích hợp cảm biến tia lửa cho sản phẩm 46](#_Toc128847880)

[CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 47](#_Toc128847881)

[7.1 Kết luận 47](#_Toc128847882)

[7.2 Hướng phát triển 47](#_Toc128847883)

[TÀI LIỆU THAM KHẢO 49](#_Toc128847884)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Kí hiệu** | **Thuật ngữ đầy đủ** |
| **1** | **CSDL** | **Cơ sở dữ liệu** |
| **2** | **VĐK** | **Vi điều khiển** |
| **3** | **ESP** | **Board ESP8266 NodeMCU** |
| **4** | **IoT** | **Internet of Things** |
| **5** | **API** | **Application Programming Interface** |
| **6** | **HTML** | **Hypertext Markup Language** |
| **7** | **CSS** | **Cascading Style Sheets** |
| **8** | **GND** | **Ground** |
| **9** | **Linq** | **Language Integrated Query** |

# DANH MỤC HÌNH VẼ

[Hình 2.1 Sơ đồ chân ESP8266EX 7](#_Toc128837944)

[Hình 2.2 ESP8266 NodeMCU và các chất kết nối 9](file:///D:\DATN-20221-nho-thay-check.docx#_Toc128837945)

[Hình 2.3 Cảm biến nhiệt độ độ ẩm DHT11 10](#_Toc128837946)

[Hình 2.4 Module phát hồng ngoại KY-005 11](#_Toc128837947)

[Hình 3.1 Sơ đồ tổng quan hệ thống 13](#_Toc128837948)

[Hình 3.2 Sơ đồ khối hệ thống 14](#_Toc128837949)

[Hình 3.3 Sơ đồ nguyên lý khối nguồn của kit ESP8266 NodeMCU 15](#_Toc128837950)

[Hình 3.4 Sơ đồ nguyên lý của chip ESP-12E và chip nạp CP 2102 15](#_Toc128837951)

[Hình 3.5 Sơ đồ nguyên lý khối cảm biến 16](#_Toc128837952)

[Hình 3.6 Sơ đồ nguyên lý khối ngoại vi 16](#_Toc128837953)

[Hình 3.7 Mô phỏng mạch trên phần mềm Fritzing 0.9.3 17](#_Toc128837954)

[Hình 3.8 Mạch in lớp Bottom 17](#_Toc128837955)

[Hình 3.9 Mô hình mạch 3D 18](#_Toc128837956)

[Hình 3.10 Sản phẩm thực tế 19](#_Toc128837957)

[Hình 3.11 Giao diện Arduino IDE 20](#_Toc128837958)

[Hình 3.12 Khởi tạo kết nối wifi cho ESP 21](#_Toc128837959)

[Hình 3.13 Định nghĩa các chân kết nối của ESP 21](#_Toc128837960)

[Hình 3.14 Khởi tạo các kết nối trong hàm void setup 22](#_Toc128837961)

[Hình 3.15 Gọi các hàm chức năng chạy trong hàm void loop 22](#_Toc128837962)

[Hình 3.16 Logic thực hiện bật tắt LED1 trong mã nguồn cho ESP 23](#_Toc128837963)

[Hình 3.17 Giao diện điều khiển trên máy tính 24](#_Toc128837964)

[Hình 3.18 Giao diện điều khiển trên điện thoại 24](#_Toc128837965)

[Hình 3.19 Hàm xử lý sự kiện bật tắt đèn 1 khi click trên giao diện 26](#_Toc128837966)

[Hình 3.20 Hàm gửi giá trị “KeyRemote” lên server 26](#_Toc128837967)

[Hình 3.21 Giao diện và bố cục dự án trên phần mềm Visual Studio 2022 27](#_Toc128837968)

[Hình 3.22 Thư mục controller 27](#_Toc128837969)

[Hình 3.23 Thư mục Service 28](#_Toc128837970)

[Hình 3.24 Thư mục Model 28](#_Toc128837971)

[Hình 3.25 Thư mục Configuration 29](#_Toc128837972)

[Hình 3.26 Bảng thực thể liên kết 30](#_Toc128837973)

[Hình 4.1 Các thư viện sử dụng trong lập trình ESP 33](#_Toc128837974)

[Hình 4.2 Mô hình MVC 36](#_Toc128837975)

[Hình 4.3 Mã nguồn khởi tạo “API SpeechRecognition” nhận dạng giọng nói tiếng việt 38](#_Toc128837976)

# DANH MỤC BẢNG BIỂU

[Bảng 3‑1 Cấu trúc bảng Remotes 30](#_Toc128837977)

[Bảng 3‑2 Cấu trúc bảng FireAlarms 30](#_Toc128837978)

[Bảng 3‑3 Cấu trúc bảng TempHums 31](#_Toc128837979)

[Bảng 3‑4 Cấu trúc bảng StatusBulbs 31](#_Toc128837980)

[Bảng 3‑5 Cấu trúc bảng StatusSpeaks 31](#_Toc128837981)

# TÓM TẮT ĐỒ ÁN

Với đề tài đồ án *“Điều khiển thiết bị trong nhà thông qua webserver*”*,* về phần cứngem đã sử dụng board ESP8266 NodeMCU, module cảm biến nhiệt độ - độ ẩm DHT11, module cảm biến tia lửa, module phát hồng ngoại.

Về lập trình cho ESP8266 NodeMCU sử dụng ngôn ngữ lập trình C++, được viết, biên dịch và nạp thông qua công cụ Arduino IDE – một công cụ lập trình cho VĐK phổ biến nhờ có cộng đồng hỗ trợ lớn.

Phần giao diện điều khiển (phần front-end) được viết bằng ngôn ngữ HTML, CSS và xử lý logic hiển thị, ghép nối API bằng ngôn ngữ lập trình Javascript. Phần logic xử lý các lệnh điều khiển, xử lý các trạng thái của các thiết bị và là ghép nối giao tiếp với CSDL (phần back-end) được viết bằng ngôn ngữ lập trình C#. Cơ sở dữ liệu sử dụng là SQL Server.

Trong đồ án này, front end được viết trên Visual Studio Code và được chạy thông qua extention “live server” trên chính Visual Studio Code. Back end được viết trên Visual Studio 2022, sử dụng .NET6 và một số framework phát triển bởi Microsoft hỗ trợ làm việc với CSDL một cách dễ dàng thông qua ngôn ngữ truy vấn “Linq” và cho phép ghép nối với phần front end hay các clients (ESP) thông qua 1 đường dẫn swagger.

Về nguyên lý hay logic hoạt động, dự án hoạt động theo mô hình “Clients – Server”. Ở đây, ESP và phần giao diện điều khiển đóng vai trò như các clients, ESP lấy giá trị của cảm biến nhiệt độ - độ ẩm, cảm biến lửa và gửi lên server thông qua 1 API gửi dữ liệu. Server nhận và lưu lại giá trị nhận được đó vào database. Giá trị này được thay đổi liên tục sau mỗi khoảng thời gian nếu như giá trị mới đọc được khác với giá trị cũ. Khoảng thời gian đó do mình setup trong khi lập trình đọc và gửi dữ liệu của VĐK. Front end thông qua 1 API khác sẽ lấy các giá trị lưu trong database và hiển thị lên giao diện. Đồng thời nếu giá trị của cảm biến cháy là “true”(cảm biến phát hiện có tia lửa), giao diện sẽ hiện lên 1 cảnh báo cháy và tự động tắt các thiết bị điện tử công suất lớn (trong dự án là điều hòa nhiệt độ) để giảm thiểu độ nguy hiểm khi xảy ra cháy nổ. Về phần điều khiển, trên giao diện điều khiển khi ta click 1 nút trên giao diện điều khiển sẽ gọi đến 1 API thực hiện việc thay đổi lệnh điều khiển lưu vào database. ESP liên tục lấy các lệnh điều khiển đó từ database và thực hiện điều khiển bật tắt hay điều khiển 1 chức năng nào đó của thiết bị mỗi khi có lệnh mới.

Các thiết bị được sử dụng để điều khiển đề mô cho bài gồm có:

* 1 đèn led 2 chân, chân âm (-) nối với chân GND, chân dương (+) được nối trực tiếp với chân D4 của VĐK có thể điều khiển bật tắt led.
* 1 đèn led RGB (led hiển thị 3 màu: red-green-blue) 4 chân, chân âm (-) nối với chân GND, 3 chân còn lại được nối với 3 chân điều khiển D5, D6, D7 của VĐK có thể điều khiển bật tắt và thay đổi màu sắc của led.
* 1 điều hòa thương hiệu Panasonic có thể điều khiển các chức năng bằng remote hồng ngoại (ở bài này ta sử dụng module phát hồng ngoại để điều khiển các chức năng của điều hòa thay cho remote)
* 1 chiếc loa, tương tự như điều hòa, có thể điều khiển thông qua module phát hồng ngoại thay cho remote.

ABSTRACT

For the project "Controlling devices inside the house through a webserver," I used the ESP8266 NodeMCU board, the DHT11 temperature and humidity sensor module, the flame sensor module, and the infrared transmitter module for hardware. The ESP8266 NodeMCU was programmed using the C++ programming language, written, compiled, and uploaded through the popular programming tool for microcontrollers, the Arduino IDE, which has a large supporting community.

The front-end control interface was written in HTML, CSS, and JavaScript for handling logic, displaying data, and connecting APIs. The back-end logic for processing control commands, handling device states, and connecting with a database was written in C# programming language. The database used was SQL Server.

In this project, the front end is written in Visual Studio Code and run through the "live server" extension within Visual Studio Code. The back end is written in Visual Studio 2022, using .NET6 and several Microsoft-developed frameworks that support easy database querying through the "Linq" query language and allow for integration with the front end or clients (ESP) through a swagger path.

In terms of principle or logic of operation, the project operates according to the "Clients-Server" model. Here, the ESP and control interface act as clients, with the ESP retrieving temperature-humidity sensor values, flame sensor values, and sending them to the server through an API to send data. The server receives and stores the received values in the database. These values are constantly updated after each period of time if the newly read value is different from the old value. This period of time is set up during the programming of the VDK for reading and sending data. The front end, through another API, retrieves the stored values in the database and displays them on the interface. At the same time, if the value of the flame sensor is "true" (detecting flames), the interface will display a fire alert and automatically turn off high-power electronic devices (in this project, air conditioners) to reduce the danger in the event of a fire.

Regarding control, when a button on the control interface is clicked, it calls an API to perform the control command, which is saved to the database. The ESP continuously retrieves these control commands from the database and performs the turn on/off or control of a device function whenever there is a new command.

The devices used to control in the simulation include:

* 1 two-pin LED light, the negative (-) pin is connected to the GND pin, and the positive (+) pin is directly connected to the D4 pin of the VDK to turn the LED light on/off.
* 1 RGB LED light (displaying 3 colors: red-green-blue) with 4 pins, the negative (-) pin is connected to the GND pin, and the other 3 pins are connected to the D5, D6, D7 control pins of the VDK to turn the LED light on/off and change its color.
* 1 Panasonic air conditioner that can be controlled by infrared remote (in this project, an infrared transmitter module is used to control the air conditioner functions instead of the remote).
* 1 speaker, similar to the air conditioner, can be controlled through an infrared transmitter module instead of a remote.

# TỔNG QUAN ĐỀ TÀI

Trong chương này sẽ giới thiệu tổng quan về công nghệ IoT (Internet of Things) nói chung và công nghệ “Smart Home” nói riêng. Chỉ ra những tiềm năng và thách thức của “Smart Home” ở hiện tại cũng như trong tương lai. Bên cạnh đó, chương này cũng sẽ trình bày mục tiêu chính của dự án và giới hạn phạm vi của dự án, đồng thời sẽ đưa ra các định hướng giải pháp để hoàn thành những mục tiêu đề ra. Phần cuối chương sẽ tóm tắt bố cục tổng thể của những chương tiếp theo.

## Đặt vấn đề

IoT là viết tắt của "Internet of Things", hay còn gọi là "Internet của Vạn Vật" trong tiếng Việt. Đây là một mạng lưới kết nối các thiết bị, đối tượng, và các phương tiện khác nhau thông qua internet, cho phép chúng tương tác với nhau và với con người.

Các thiết bị IoT có thể là các cảm biến, máy móc, thiết bị điện tử, xe hơi, thiết bị y tế, và nhiều hơn nữa. Chúng được kết nối với mạng internet thông qua các giao thức truyền thông như Wi-Fi, Bluetooth, Zigbee, và các giao thức khác.

Các ứng dụng của IoT rất đa dạng, bao gồm giám sát, kiểm soát và điều khiển các thiết bị và hệ thống, quản lý tài sản, giám sát môi trường, cải thiện đời sống con người, và nhiều hơn nữa. IoT đang trở thành một lĩnh vực phát triển mạnh mẽ và đầy triển vọng trong thời đại kỹ thuật số hiện nay.

Công nghệ smart home là một phần của công nghệ IoT, đem lại nhiều tiềm năng và thách thức.



Hình 1.1 Công nghệ Smart home kết nối và điều khiển các thiết bị thông qua internet

### 1.1.1 Tiềm năng

* Tiềm năng tiết kiệm năng lượng và tiện lợi: Smart home có thể giúp người dùng tiết kiệm năng lượng và cải thiện tiện ích, đồng thời giúp kiểm soát được hầu hết các thiết bị trong nhà chỉ bằng một nút bấm hoặc giọng nói.
* Tương lai phát triển: Smart home đang dần trở nên phổ biến và được ứng dụng rộng rãi. Vì vậy, tiềm năng phát triển của smart home là rất lớn, và công nghệ này có thể giúp cải thiện cuộc sống của người dùng theo nhiều cách khác nhau.

### 1.1.2 Thách thức:

* Phụ thuộc vào mạng internet: Smart home hoạt động thông qua kết nối internet, vì vậy khi mạng bị gián đoạn hoặc có vấn đề kết nối, các thiết bị smart home sẽ không hoạt động đúng cách.
* Tính tương thích: Smart home được thiết kế với nhiều thiết bị khác nhau, từ các cảm biến, đèn led, điều hòa nhiệt độ đến các hệ thống an ninh. Để các thiết bị này hoạt động với nhau, chúng cần phải tương thích với nhau. Điều này đòi hỏi sự phối hợp giữa các nhà sản xuất và chuẩn hóa các giao thức kết nối.
* Giá thành: Giá thành của các thiết bị smart home thường cao hơn các thiết bị thông thường, do đó chi phí triển khai và cài đặt các hệ thống smart home cần được xem xét.
* Và 1 thách thức lớn nhất đó là vấn đề bảo mật: Như đã đề cập trong công nghệ IoT, bảo mật là một thách thức quan trọng của công nghệ smart home. Vì các thiết bị trong smart home liên kết với nhau thông qua mạng internet, chúng cần được bảo vệ an toàn để tránh bị tấn công từ bên ngoài hoặc nội bộ.

## Mục tiêu và phạm vi đề tài

### 1.2.1 Mục tiêu

Dự án có mục tiêu chính là xây dựng 1 hệ thống Smart Home sử dụng trong các hộ gia đình nhằm tăng tính tiện lợi, tính thuận tiện và giải quyết một số vấn đề thường gặp khi sử dụng các thiết bị điện như:

* Khi ra khỏi nhà thường quên tắt các thiết bị điện như đèn điện, điều hòa,v.v. gây lãng phí điện năng.
* Remote thông thường sau 1 thời gian sử dụng hay bị hỏng và không thể điều khiển được nữa.
* Nhà có con nhỏ, các bé thường sử dụng điều hòa nhiệt độ với mức nhiệt thấp trong thời gian dài gây ảnh hưởng sức khỏe.
* Khi xảy ra sự cố như cháy nổ thường để lại hậu quả rất nghiêm trọng do phát hiện ra đám cháy muộn hoặc đám cháy lan tới các thiết bị điện có công suất lớn đang hoạt động.
* Một số hệ thống smart home hiện nay khi lắp đặt mất rất nhiều chi phi do các thiết bị điện cũ không tích hợp các chức năng với công nghệ mới, muốn sử dụng cần lắp thêm các sản phẩm hỗ trợ hoặc phải thay mới.

### 1.2.2 Phạm vi đề tài

Đề tài tập trung xây dựng và phát triển sản phẩm dựa theo các tiêu chí:

* Tính tiện nghi
* Dễ dàng lắp đặt và sử dụng
* Giá thành tốt
* Tăng độ an toàn
* Tính tương thích tốt
* Dễ dàng nâng cấp trong tương lai

## Định hướng giải pháp

Các giải pháp được em đưa ra và sử dụng nhằm khắc phục các vấn đề gặp phải như nói ở trên:

* Sử dụng sóng hồng ngoại để điều khiển vì hầu như toàn bộ các thiết bị điện trong nhà hiện tại đều sử dụng sóng hồng ngoại để điều khiển.
* Tích hợp thêm cảm biến báo cháy để có thể cảnh báo, báo động cho chủ nhà khi có đám cháy xảy ra, đồng thời tự động tắt các thiết bị tiêu thụ điện lớn (là 1 phần nguyên nhân làm tăng mức độ nghiêm trọng của đám cháy) để làm giảm phần nào hậu quả khi có cháy nổ xảy ra.
* Tích hợp thêm cảm biến nhiệt độ - độ ẩm nhằm giám sát được mức nhiệt trong phòng từ đó có thể điều chỉnh mức nhiệt cho phù hợp, tránh tình trạng sử dụng mức nhiệt thấp trong thời gian dài gây ảnh hưởng đến sức khỏe.
* Lựa chọn một mô hình xây dựng phù hợp vừa dễ dàng bảo trì, vừa dễ dàng nâng cấp trong tương lai.

## Bố cục đồ án

Nội dung chính của quyển đồ án tốt nghiệp gồm các chương:

Chương 2: Tổng quan về các linh kiện sử dụng trong dự án

* Tập trung giới thiệu về ESP8266 và board phát triển sử dụng trong dự án là board ESP8266 NodeMCU, đồng thời giới thiệu các module : cảm biến nhiệt độ độ ẩm dht11, cảm biến tia lửa (flame sensor), module phát hồng ngoại (Infrared LED module).

Chương 3: Thiết kế hộ thống

* Phần này sẽ trình bày tổng thể mô hình của dự án, trình bày mô phỏng sản phẩm, mạch nguyên lý, mạch in và sản phẩm thực tế. Đồng thời sẽ trình bày những phần quan trọng nhất, là cốt lõi trong mã nguồn của cả phần cứng và phần mềm.

Chương 4: Các giải pháp và đóng góp nổi bật

* Phần này giới thiệu các công nghệ sử dụng trong dự án gồm các công nghệ cho phần cứng, phần mềm, cho phần giao diện điều khiển (front-end) lẫn phần logic xử lý điều khiển, tạo liên kết với database (back-end) và database sử dụng. Giới thiệu các thư viện, các framework.

Chương 5: Kiểm thử và đánh giá

* Phần nãy sẽ ghi lại quá trình thử nghiệm các chức năng của sản phẩm tại các thời điểm ngẫu nhiên và đưa ra đánh giá dựa trên quá trình thử nghiệm đó.

Chương 6: Các giải pháp và đóng góp tiêu biểu

* Phần này sẽ tổng hợp và trình bày những đóng góp tiêu biểu về mô hình dự án, về cách thức vận hành và các ý tưởng mới được áp dụng trong dự án.

Chương 7: Kết luận và hướng phát triển

* Trình bày về các kết quả em đã đạt được và chưa được trong ĐATN lần này, cùng với đó là những định hướng trong tương lai dựa trên những tính năng mà dự án đã xây dựng và sẽ xây dựng.

# GIỚI THIỆU TỔNG QUAN VỀ CÁC LINH KIỆN SỬ DỤNG TRONG DỰ ÁN

Chương 2 sẽ giới thiệu toàn bộ các linh kiện sử dụng trong dự án bao gồm chip ESP8266 nói chung và board mạch phát triển của nó ESP8266 NodeMCU nói riêng. Giới thiệu các module và cảm biến được tích hợp trong dự án bao gồm module cảm biến nhiệt độ - độ ẩm dht11, module phát hiện tia lửa và module phát hồng ngoại. Giới thiệu về định nghĩa, thông số kĩ thuật, sơ đồ chân và hình vẽ cụ thể của các linh kiện.

## Tổng quan về ESP8266 và board ESP8266 NodeMCU

### 2.1.1 ESP8266

ESP8266 là một module Wi-Fi thông minh được phát triển bởi công ty Espressif Systems. Nó cung cấp cho các thiết bị IoT (Internet of Things) khả năng kết nối không dây, cho phép chúng ta kiểm soát và giám sát các thiết bị thông qua Internet.

Hiện nay tất cả các dòng chip ESP8266 trên thị trường đều mang nhãn ESP8266EX, là phiên bản nâng cấp của ESP8266.

Sơ đồ chân:

Chart

Description automatically generated with low confidence

Hình . Sơ đồ chân ESP8266EX

Thông số kĩ thuật

* 32-bit RISC CPU : Tensilica Xtensa LX106 chạy ở xung nhịp 80 MHz
* Hổ trợ Flash ngoài từ 512KiB đến 4MiB
* 64KBytes RAM thực thi lệnh
* 96KBytes RAM dữ liệu
* 64KBytes boot ROM
* Chuẩn wifi EEE 802.11 b/g/n, Wi-Fi 2.4 GHz
* Tích hợp TR switch, balun, LNA, khuếch đại công suất và matching network
* Hổ trợ WEP, WPA/WPA2, Open network
* Tích hợp giao thức TCP/IP
* Hổ trợ nhiều loại anten
* 16 chân GPIO
* Hổ trợ SDIO 2.0, UART, SPI, I²C, PWM,I²S với DMA
* 1 ADC 10-bit
* Dải nhiệt độ hoạt động rộng : -40 ~ 125 ℃

### 2.1.2 Board phát triển ESP8266 NodeMCU

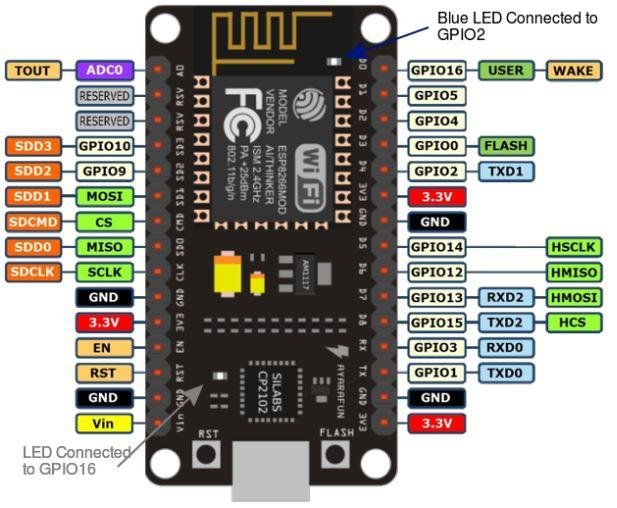
Board phát triển ESP8266 NodeMCU là một bộ phát triển phần cứng dựa trên module ESP8266, được thiết kế để giúp các nhà phát triển nhanh chóng phát triển các ứng dụng IoT thông qua mạng Wi-Fi. Board này bao gồm một board mạch NodeMCU và một bộ cấu hình cho phép các nhà phát triển kết nối và lập trình board một cách dễ dàng.

Board mạch NodeMCU được thiết kế với nhiều tính năng hữu ích, bao gồm:

* Bộ xử lý đơn lõi ESP8266EX với tốc độ xung nhịp 80MHz hoặc 160MHz.
* Giao diện Wi-Fi 802.11 b/g/n, hỗ trợ chế độ chạy phát sóng (AP) và chế độ kết nối (Station).
* Kết nối UART và USB để lập trình và giao tiếp với máy tính.
* Đầu nối GPIO với khả năng đọc/ghi tín hiệu số và tín hiệu analog.

Board phát triển ESP8266 NodeMCU cũng được hỗ trợ bởi một cộng đồng lớn các nhà phát triển, cung cấp các tài liệu, ví dụ và hướng dẫn để giúp người dùng bắt đầu với

việc phát triển ứng dụng IoT. Các chức năng của board này có thể được lập trình bằng các ngôn ngữ lập trình như C/C++ hoặc Lua, và có thể sử dụng các công cụ phát triển phổ biến như Arduino IDE để lập trình và nạp chương trình cho board mạch.



Hình . ESP8266 NodeMCU và các chất kết nối

## Giớithiệu các module sử dụng trong dự án

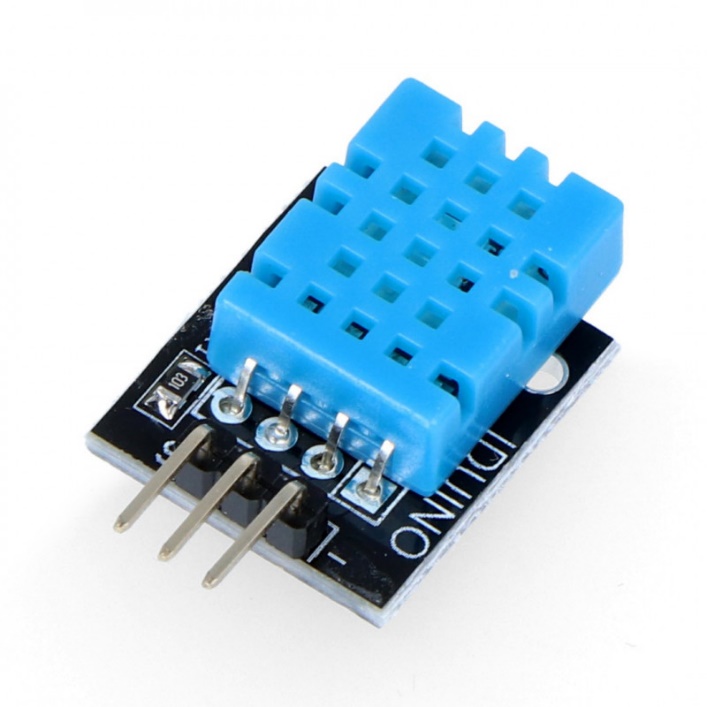
### 2.2.1 Module cảm biến nhiệt độ, độ ẩm (DHT11)

Module cảm biến nhiệt độ, độ ẩm DHT11 là một cảm biến kỹ thuật số được sử dụng để đo nhiệt độ và độ ẩm trong các ứng dụng IoT. Nó có thể được sử dụng để giám sát môi trường xung quanh và cung cấp dữ liệu cho các thiết bị nhà thông minh hoặc các hệ thống giám sát.

Module cảm biến DHT11 sử dụng công nghệ cảm biến độ ẩm và nhiệt độ để đo giá trị của chúng và truyền dữ liệu đến bộ điều khiển thông qua giao thức kỹ thuật số. Nó có thể được sử dụng với nhiều loại vi điều khiển khác nhau, và có thể lập trình bằng các ngôn ngữ lập trình như C/C++ hoặc Python để đọc dữ liệu từ cảm biến và điều khiển các thiết bị khác.

Thông số kỹ thuật :

* Điện áp hoạt động: 3.3V đến 5V
* Độ ẩm đo được: từ 20% đến 80% với độ chính xác ±5%
* Nhiệt độ đo được: từ 0 đến 50 độ C với độ chính xác ±2 độ C
* Tần số đọc dữ liệu: tối đa 1 lần mỗi giây
* Giao tiếp: Digital (sử dụng giao thức 1-wire)
* Kích thước: 12mm x 15.5mm x 5.5mm



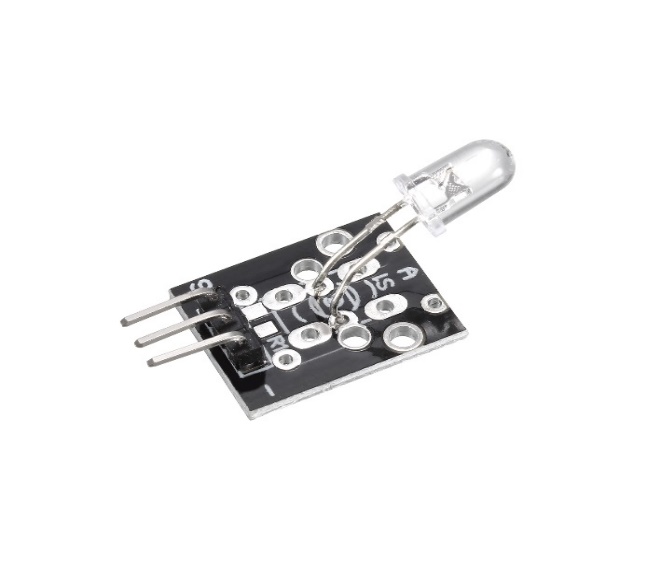
Hình . Cảm biến nhiệt độ độ ẩm DHT11

### 2.2.2 Module phát sóng hồng ngoại (KY-005 Infrared Transmitter)

Module KY-005 Infrared Transmitter là một module phát hồng ngoại được sử dụng để gửi tín hiệu hồng ngoại điều khiển các thiết bị khác hoặc truyền dữ liệu từ một thiết bị này sang thiết bị khác.

Module KY-005 bao gồm một LED phát hồng ngoại (IR LED) và một resistor để giới hạn dòng điện. Khi một tín hiệu được đưa vào module, LED phát hồng ngoại sẽ phát ra tín hiệu hồng ngoại, tần số và mã hóa tùy thuộc vào ứng dụng cụ thể.

Module KY-005 được sử dụng rộng rãi trong các ứng dụng điều khiển từ xa, chẳng hạn như điều khiển các thiết bị như TV, điều hòa không khí, đầu đĩa DVD và các thiết bị khác từ xa. Nó cũng có thể được sử dụng trong các ứng dụng truyền dữ liệu từ một thiết bị này sang thiết bị khác thông qua kết nối hồng ngoại.



Hình . Module phát hồng ngoại KY-005

Thông số kỹ thuật

* Điện áp hoạt động: 3-5V
* Dòng tiêu thụ: 20mA
* Tần số phát: 38kHz
* Khoảng cách phát: khoảng 10m
* Kích thước: 19mm x 15mm x 5mm2.2.3 Module cảm biến tia lửa (Flame sensor)

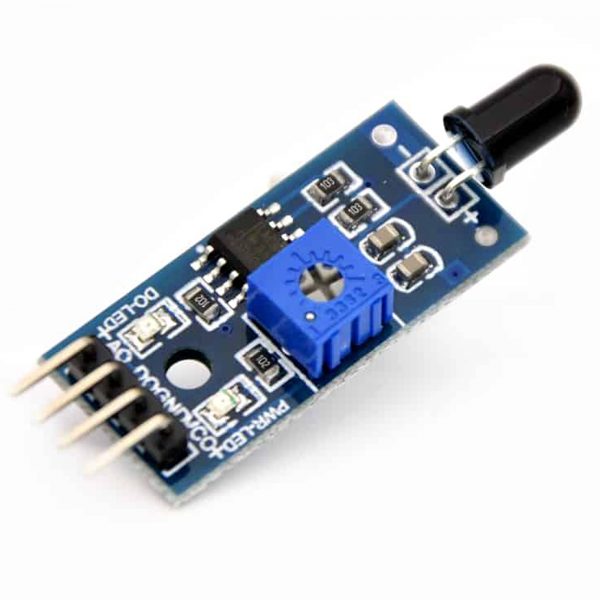
Cảm biến phát hiện tia lửa (flame sensor) là một loại cảm biến được sử dụng để phát hiện sự hiện diện của lửa hoặc ngọn lửa. Cảm biến này có thể phát hiện các ngọn lửa với các bước sóng trong dải từ khoảng 760nm đến 1100nm.

Cảm biến phát hiện lửa thường được sử dụng trong các ứng dụng an ninh, bảo vệ cháy, các hệ thống tự động điều khiển nhà thông minh và các ứng dụng điện tử khác. Khi cảm biến phát hiện lửa phát hiện được sự hiện diện của lửa, nó sẽ cho ra một tín hiệu ngõ ra để điều khiển các hệ thống khác, chẳng hạn như ngắt điện hoặc kích hoạt hệ thống báo động.

Cảm biến phát hiện lửa có thể được sử dụng độc lập hoặc kết hợp với các module khác để tạo ra các ứng dụng phát hiện lửa thông minh hoặc hệ thống bảo vệ cháy.

Thông số kỹ thuật:

* Nguồn cấp: 3.3V - 5VDC
* Dòng tiêu thụ: 15mA
* Tín hiệu ra: Digital 3.3 - 5VDC tùy nguồn cấp.
* Khoảng cách : 80 cm
* Góc quét : 60 độ
* Kích thước : 3.2 x 1.4 cm



**Hình 2.5 Cảm biến phát hiện tia lửa**

# THIẾT KẾ HỆ THỐNG

Chương 3 sẽ trình bày toàn bộ hệ thống bao gồm toàn bộ phần cứng và phần mềm. Trình bày lần lượt về tổng quan hệ thống, sơ đồ khối và chức năng của từng khối trong hệ thống, hình ảnh mạch in, mạch mô phỏng và mạch thực tế của sản phẩm. Đồng thời sẽ trình bày những phần quan trọng trong mã nguồn lập trình cho ESP, lập trình phần giao diện, cho logic xử lý cũng như CSDL sử dụng trong dự án này.

## 3.1 Tổng quan hệ thống

Diagram

Description automatically generated

Hình . Sơ đồ tổng quan hệ thống

Cách thức hoạt động của hệ thống:

* Phần điều khiển thiết bị:
* Truy cập giao diện điều khiển thông qua trình duyệt trên điện thoại hoặc máy tính và thực hiện nhấn nút điều khiển trên giao diện hoặc điều khiển bằng giọng nói sẽ gửi một lệnh điều khiển đến webserver và được lưu lại trong database
* ESP thông qua webserver lấy lệnh điều khiển trong database và thực hiện điều khiển đèn hoặc gửi sóng hồng ngoại để điều khiển các thiết bị có sử dụng công nghệ remote hồng ngoại.
* Phần hiển thị nhiệt độ, độ ẩm, cảm biến cháy:
* ESP lấy dữ liệu từ các cảm nhiệt độ - độ ẩm và cảm biến cháy rồi gửi lên server, server nhận và lưu giá trị đó vào trong database
* Giao diện lấy giá trị đó trong database và hiển thị

## 3.2 Chức năng và sơ đồ khối

### 3.2.1 Chức năng của toàn hệ thống

* Điều khiển thiết bị điện thông qua giao diện điều khiển (web client) tích hợp điều khiển bằng giọng nói,
* Hiển thị trạng thái thiết bị (đèn, điều hòa, loa) và các thông số môi trường như nhiệt độ, độ ẩm, cảnh báo cháy (nếu có cháy nổ xảy ra) lên giao diện web client.
* Tự động tắt các thiết bị có công suất lớn khi có cháy nổ xảy ra.

### 3.2.2 Sơ đồ khối

Diagram

Description automatically generated

Hình . Sơ đồ khối hệ thống

### 3.2.3 Mạch nguyên lý

Khối nguồn:

* Các module, cảm biến và LED trong dự án em sử dụng nguồn trực tiếp từ ESP

Diagram, schematic

Description automatically generated

Hình . Sơ đồ nguyên lý khối nguồn của kit ESP8266 NodeMCU

Khối xử lý trung tâm và truyền thông

* Phần xử lý trung tâm và truyền thông chính là chip ESP-12E – là một module Wi-Fi được tích hợp trên board phát triển ESP8266 NodeMCU gồm bộ xử lý ESP8266EX, bộ nhớ flash, bộ nhớ RAM, anten Wi-Fi tích hợp và các chân ngoại vi để kết nối với các thiết bị khác

Diagram, schematic

Description automatically generated

Hình . Sơ đồ nguyên lý của chip ESP-12E và chip nạp CP 2102

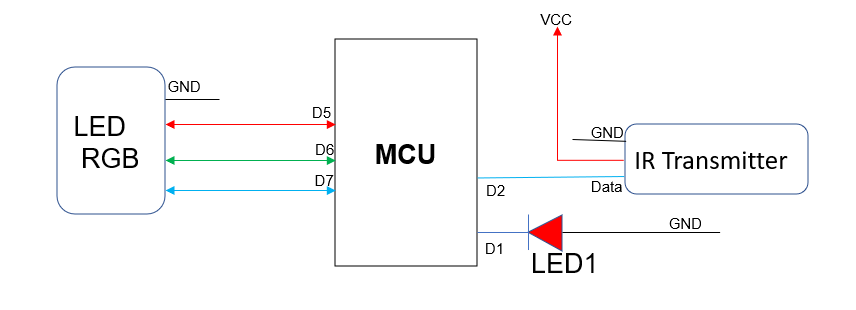
Khối cảm biến

Diagram

Description automatically generated

Hình . Sơ đồ nguyên lý khối cảm biến

Khối ngoại vi



Hình . Sơ đồ nguyên lý khối ngoại vi

## 3.3 Mô phỏng, mạch in và sản phẩm thực tế

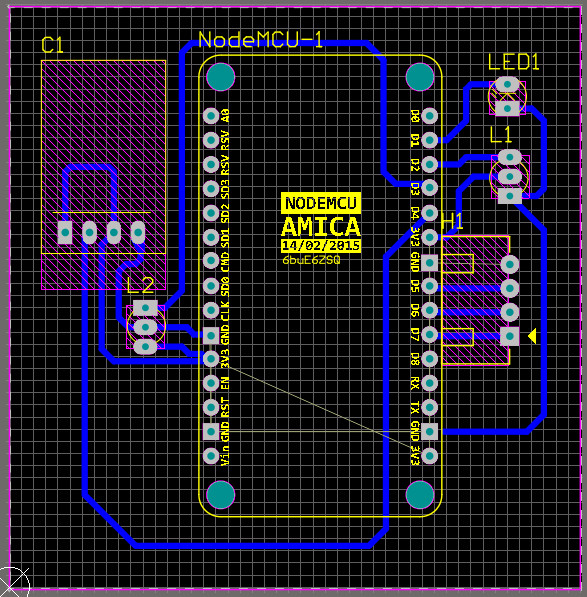
### 3.3.1 Mô phỏng

Chart

Description automatically generated

Hình . Mô phỏng mạch trên phần mềm Fritzing 0.9.3

### 3.3.2 Mạch in



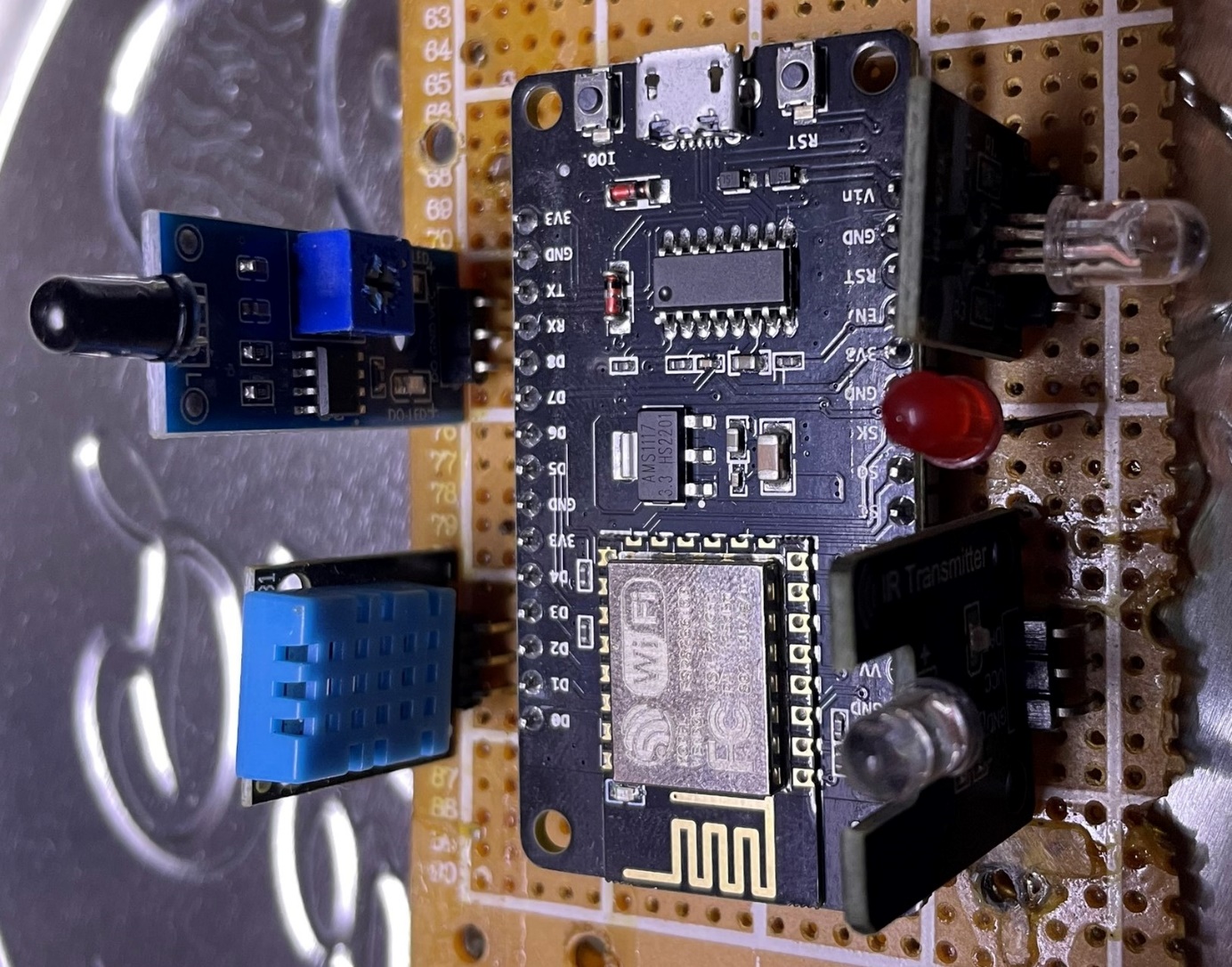
Hình . Mạch in lớp Bottom

A picture containing text, electronics, screenshot

Description automatically generated

Hình . Mô hình mạch 3D

### 3.3.3 Sản phẩm thực tế



Hình . Sản phẩm thực tế

## 3.4 Lập trình cho ESP

Về lập trình cho ESP8266, em dử dụng công cụ Arduino IDE kèm theo một số thư viện hỗ trợ được xây dựng sẵn.

### 3.4.1 Giới thiệu tổng quan về Arduino IDE

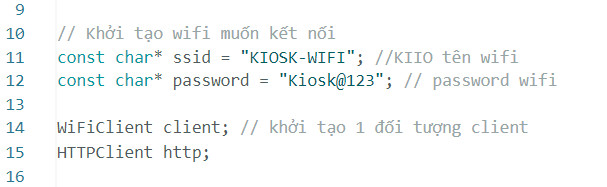


Hình . Giao diện Arduino IDE

Arduino IDE là một phần mềm được sử dụng để lập trình cho các board vi điều khiển của Arduino. IDE là viết tắt của "Integrated Development Environment" (Môi trường Phát triển tích hợp), và nó bao gồm một trình soạn thảo mã nguồn, trình biên dịch và trình nạp chương trình vào board Arduino.

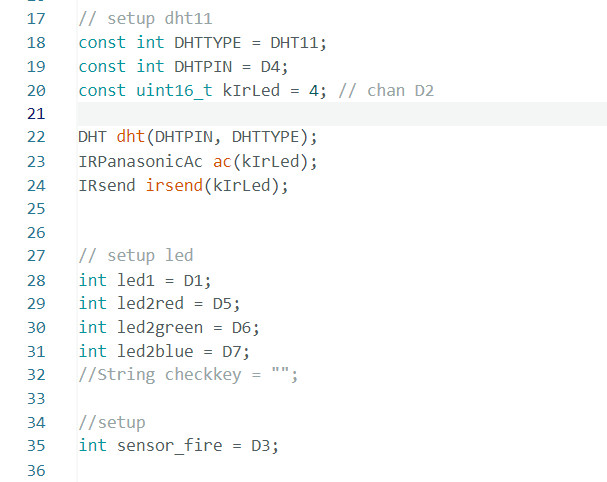
Arduino IDE cung cấp cho người dùng một giao diện đơn giản và thân thiện để viết và tải chương trình vào board Arduino một cách dễ dàng. Nó cũng đi kèm với thư viện chứa các hàm tiêu chuẩn để giúp lập trình viên tạo ra các chương trình điều khiển đơn giản cho các thiết bị như đèn LED, mô-tơ, cảm biến và các thiết bị khác.

### 3.4.2 Mã nguồn cho ESP

* Trong dự án này, ESP đóng vai trò như một “Client” có khả năng truy cập vào WebServer để nhận và gửi dữ liệu.
* Đầu tiên ta cần khởi tạo một kết nối Wifi và một đối tượng là client cho ESP: 

Hình 3.12 Khởi tạo kết nối wifi cho ESP

* Tiếp theo ta định nghĩa các chân GPIO sẽ dùng cho các cảm biến và các LED:



Hình . Định nghĩa các chân kết nối của ESP

* Khởi tạo các kết nối, các kiểu đầu ra/vào của các chân GPIO đã định nghĩa:

Text

Description automatically generated

Hình . Khởi tạo các kết nối trong hàm void setup

Viết các hàm cho từng chức năng và gọi hàm đó chạy ở trong hàm void loop():

Graphical user interface, text, application

Description automatically generated

Hình . Gọi các hàm chức năng chạy trong hàm void loop

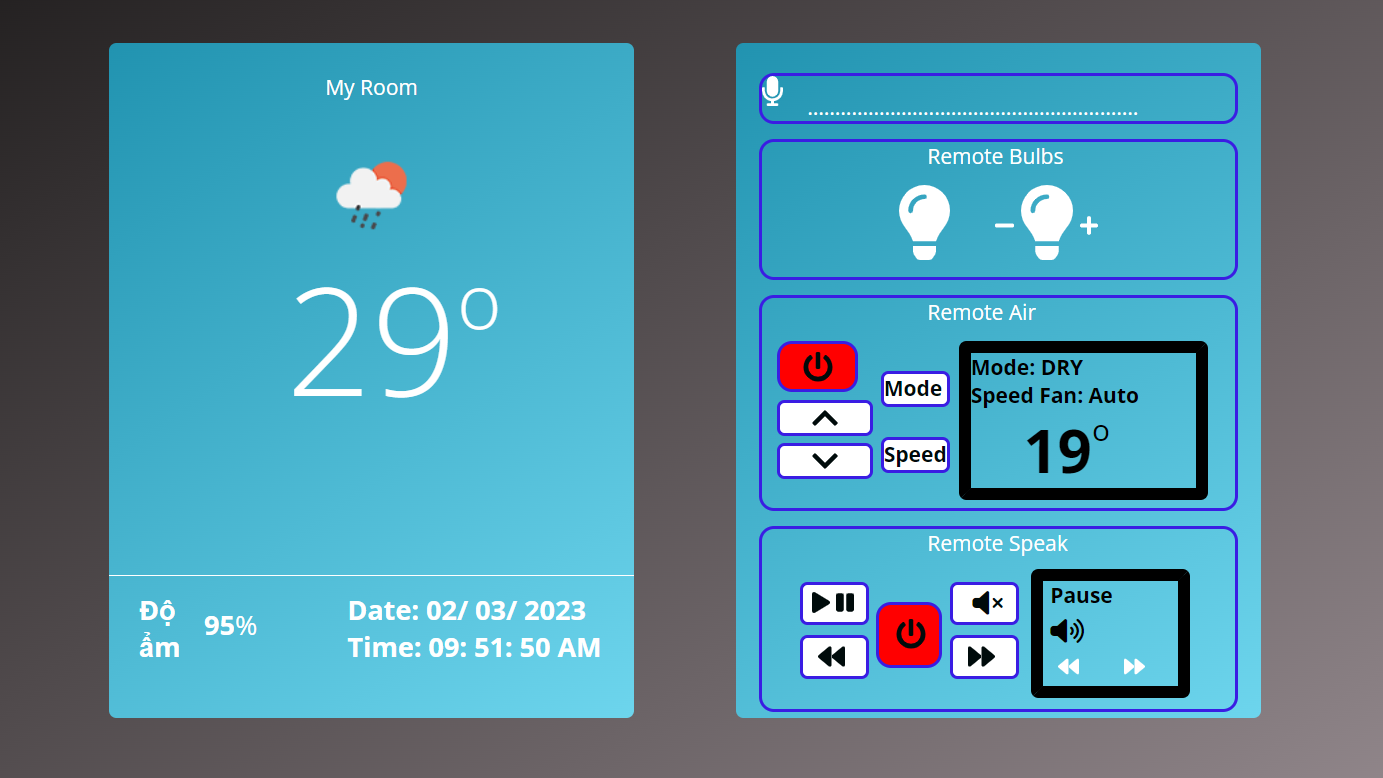
* Hàm firealarm() và temphum(): là hai hàm sẽ thực hiện đọc giá trị của cảm biến lửa, cảm biến nhiệt độ - độ ẩm DHT11 và gửi các giá trị đó lên Server.
* Hàm getAPI(): là hàm có nhận vào một tham số là một đường dẫn (url), sau khi thực hiện, hàm sẽ trả về giá trị của “keyRemote” lưu trong database, mỗi “keyRemote” sẽ ứng với một chức năng điều khiển khác nhau.
* Hàm remote(): là hàm thực hiện các chức năng điều khiển mỗi khi hàm getAPI() lấy được một “keyRemote” mới. Ví dụ về giá trị của “keyRemote” và thực thi lệnh điều khiển LED1 trong hàm remote():

Text

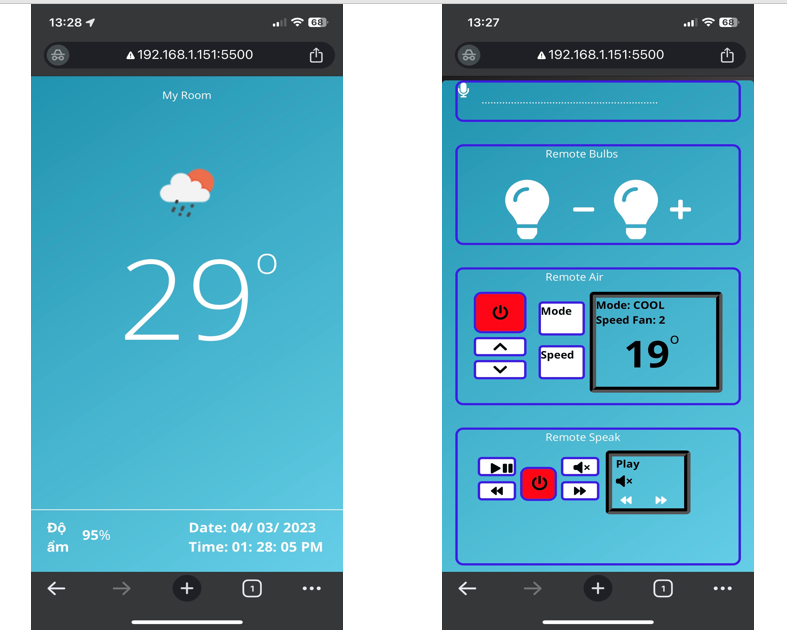
Description automatically generated

Hình . Logic thực hiện bật tắt LED1 trong mã nguồn cho ESP

## 3.5 Thiết kế phần giao diện điều khiển (web-client)



Hình . Giao diện điều khiển trên máy tính



Hình . Giao diện điều khiển trên điện thoại

### 3.5.1 Tổng quan về giao diện điều khiển

* Giao diện điều khiển được viết bằng ngôn ngữ HTML, CSS và xử lý các sự kiện click các nút điều khiển, ghép nối API bằng ngôn ngữ lập trình JavaScript
* Về tổng quan, giao diện điều khiển chia làm 2 khối chính : khối hiển thị và khối điều khiển thiết bị
* Khối hiển thị có các chức năng: hiển thị nhiệt độ, độ ẩm và hiển thị đồng hồ thời gian thực
* Khối điều khiển thiết bị chia làm 4 phần nhỏ:
* Phần điều khiển bằng giọng nói (Voice-remote) : hiển thị micro cho phép click để bắt đầu sự kiện thu và xử lý giọng nói tiếng Việt đồng thời hiển thị giọng nói thu được dưới dạng ‘text’.
* Phần điều khiển đèn: trong dự án em đề mô điều khiển 2 đèn LED, đèn LED1 có chức năng bật tắt khi click trên giao diện, đèn LED2 vừa có chức năng bật tắt như LED1, vừa có khả năng thay đổi màu sắc (đỏ - xanh lục – xanh lam).
* Phần điều khiển điều hòa: em tích hợp một số chức năng cơ bản, thông dụng nhất của một remote điều hòa gồm: tắt/bật điều hòa, tăng/giảm nhiệt độ, thay đổi chế độ và điều chỉnh tốc độ của điều hòa. Đồng thời bên cạnh các nút điều khiển sẽ có một màn hình hiển thị các trạng thái của điều hòa.
* Phần điều khiển loa: gồm một số nút điều khiển cơ bản: tắt/bật loa, play/pause nhạc, tắt/bật âm thanh và next/back nhạc. Bên cạnh các nút điều khiển cũng có một màn hình hiển thị trạng thái của loa.

### 3.5.2 Phần lập trình xử lý các sự kiện ‘Click’ và ghép nối với API (phần code JavaScript)

Về logic xử lý các sự kiện click nút điều khiển, em viết thành các hàm nhỏ, mỗi hàm sẽ thực hiện gửi một mã điều khiển (keyRemote) thông qua một API đến Server.

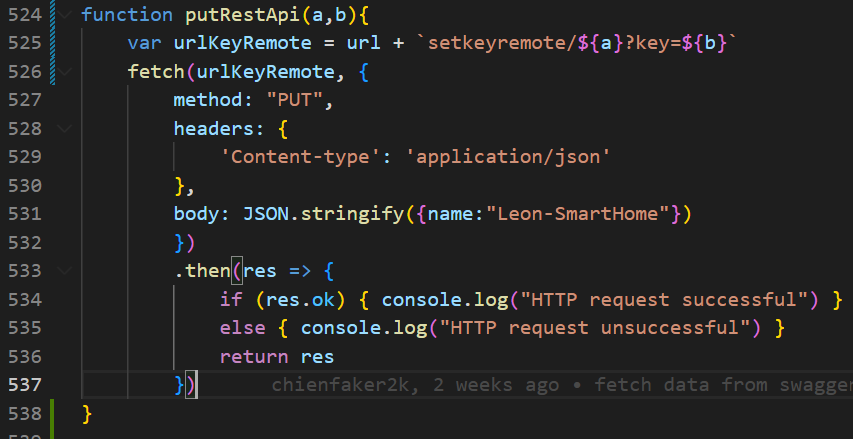
* Ví dụ với hàm bật/tắt 2 LED:

Text

Description automatically generated

Hình . Hàm xử lý sự kiện bật tắt đèn 1 khi click trên giao diện

* Khi click các sự kiện sẽ gọi đến 1 hàm gửi giá trị “keyRemote” lên server:

****

Hình . Hàm gửi giá trị “KeyRemote” lên server

## 3.6 Thiết kế phần WebServer (phần back-end)

Phần WebServer được viết bằng ngôn ngữ lập trình C#, sử dụng 2 framework của microsoft là ‘Microsoft.AspNetCore.App” và ‘Microsoft.NetCore.App” (sẽ được giới thiệu ở chương sau). Trong dự án này, em viết, biên dịch và chạy trên Visual Studio 2022.

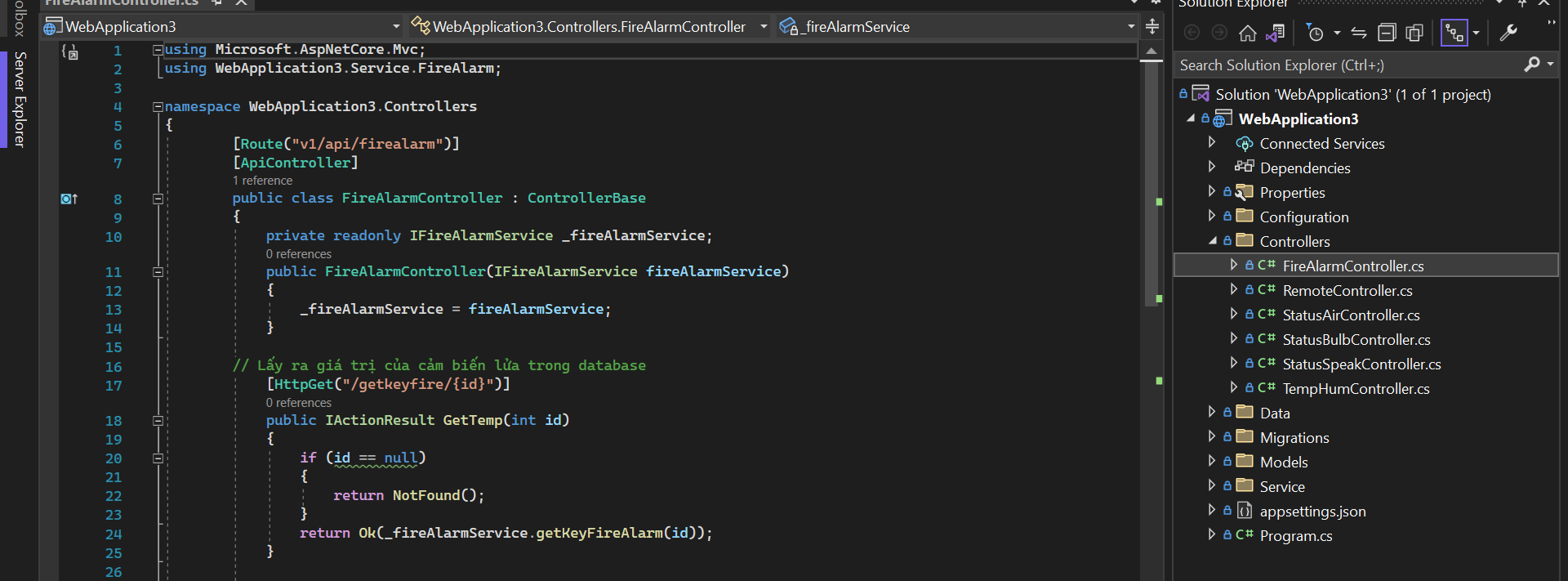
A screenshot of a computer

Description automatically generated with medium confidence

Hình . Giao diện và bố cục dự án trên phần mềm Visual Studio 2022

Bố cục của phần back end gồm các thư mục chính:

* Controllers: chứa các tệp điều khiển của dự án. Trong dự án này, em chia làm 6 tệp nhỏ, mỗi tệp là một file “.cs” (file C#) sẽ thực hiện gọi đến các service tương ứng trong file “Service” để thực thi chức năng.



Hình . Thư mục controller

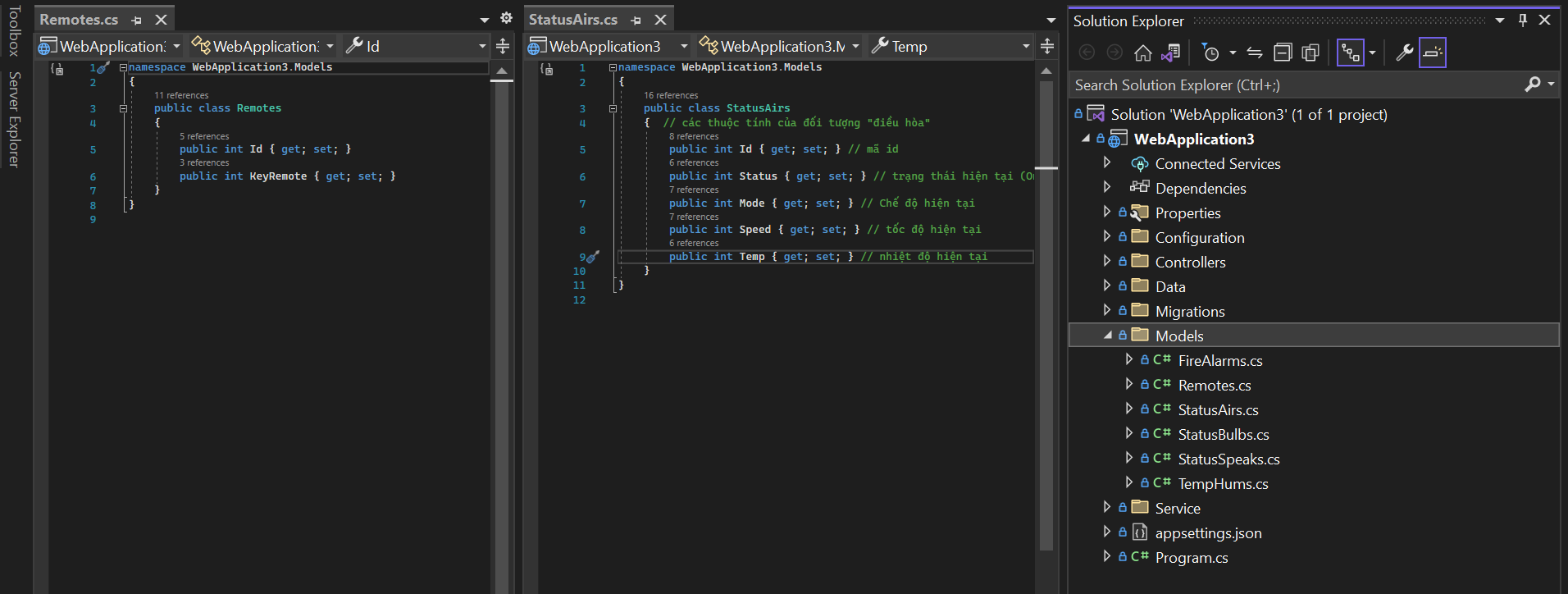
* Service: việc xử lý các chức năng, giao tiếp với CSDL được thực thi tại đây. Ứng với mỗi tệp “Controller” sẽ có một tệp “Service”, việc tách biệt 2 lớp “Controller” và “Service” giúp cho dự án dễ dàng mở rộng và phát triển mỗi khi muốn thêm chức năng mới. Ngoài ra em viết thêm các “IService” ứng với các “Service” để quản lý các hàm xử lý quan trọng.

Graphical user interface, text

Description automatically generated

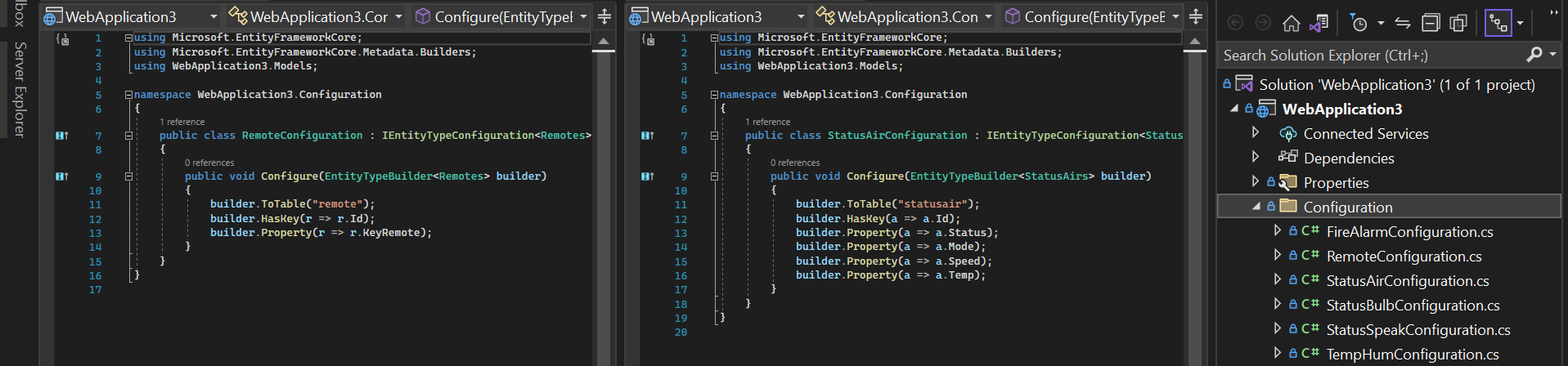
Hình . Thư mục Service

* Models: dùng để định nghĩa các lớp đối tượng và đại diện cho dữ liệu của chúng trong ứng dụng. Trong dự án này, đối tượng là các thiết bị được điều khiển, các cảm biến và một model đại diện cho các chức năng điều khiển.



Hình . Thư mục Model

* Configuration: là một phần không thể thiếu trong dự án ASP.NET Core để cấu hình Entity Framework Core (EF Core). Cụ thể là cấu hình các thuộc tính cho từng đối tượng. Trong phần này, em sử dụng phương thức "ToTable" để đặt tên cho bảng CSDL tương ứng với từng đối tượng. Phương thức "HasKey" để xác định khóa chính của từng bảng và phương thức "Property" được sử dụng để cấu hình các thuộc tính của đối tượng.

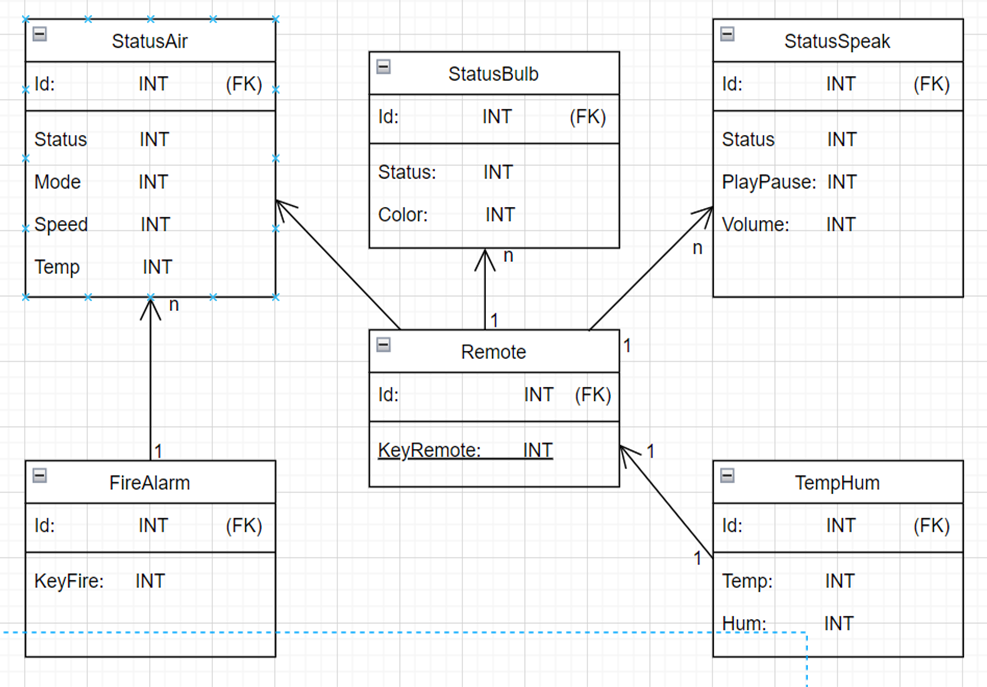


Hình . Thư mục Configuration

* Data: tại đây, em tạo ra một lớp kế thừa từ lớp “DbContext” để tương tác với cơ sở dữ liệu. Lớp này cung cấp các thuộc tính DbSet để truy vấn và thao tác dữ liệu của các bảng và truyền đối số là một đối tượng DbContextOptions để cấu hình kết nối đến cơ sở dữ liệu. Tóm lại, thư mục “Data” em viết các mã nguồn nhằm định nghĩa cấu trúc của CSDL và tương tác với CSDL thông qua các thư viễn hỗ trợ từ Entity Framework Core.
* Migrations: Lớp migrations trong dự án ASP.NET Core được sử dụng để quản lý phiên bản của cơ sở dữ liệu. Nó cho phép tạo hoặc cập nhật các bảng trong cơ sở dữ liệu một cách tự động, từ các thay đổi trong các lớp đại diện cho các bảng (được xác định bởi lớp DbContext) trong ứng dụng.
* “Program.cs”: tại đây sẽ xây dựng các dịch vụ và cấu hình cơ sở dữ liệu cho ứng dụng ASP.NET Core. Sau đó nó cấu hình pipeline xử lý các yêu cầu HTTP, kích hoạt Swagger và đăng ký các điểm cuối. Cuối cùng, nó khởi chạy ứng dụng.

## 3.7 Thiết kế CSDL

Sau khi phân tích các thông tin cần thiết cho đồ án, các thông tin về các cảm biến, các thiết bị điện, em đã tạo 1 database lưu trữ các thông tin cần thiết để có thể thực hiện các chức năng đã đặt ra ban đầu.



Hình . Bảng thực thể liên kết

Bảng ‑ Cấu trúc bảng Remotes

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Tùy chọn |
| 1 | Id | Int | PK, not null |
| 2 | KeyRemote | Int | Not null |

Bảng ‑ Cấu trúc bảng FireAlarms

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Tùy chọn |
| 1 | Id | Int | PK, not null |
| 2 | KeyFire | Int | Not null |

Bảng ‑ Cấu trúc bảng TempHums

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ kiệu | Tùy chọn |
| 1 | Id | Int | PK, not null |
| 2 | Temp | Int | Not null |
| 3 | Hum | Int | Not null |

Bảng 3‑4 Cấu trúc bảng StatusBulbs

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Tùy chọn |
| 1 | Id | Int | PK, not null |
| 2 | Status | Int | Not null |
| 3 | Color | Int | Not null |

Bảng ‑ Cấu trúc bảng StatusSpeaks

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Tùy chọn |
| 1 | Id | Int | PK, not null |
| 2 | Status | Int | Not null |
| 3 | PlayPause | Int | Not null |
| 4 | Volume | Int | Not null |

Bảng ‑6 Cấu trúc bảng StatusSpeaks

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Tùy chọn |
| 1 | Id | Int | PK, not null |
| 2 | Status | Int | Not null |
| 3 | Mode | Int | Not null |
| 4 | Speed | Int | Not null |
| 5 | Temp | Int | Not null |

# CÁC CÔNG NGHỆ SỬ DỤNG

## 4.1 Công nghệ sử dụng cho phần cứng

### 4.1.2 Các thư viện hỗ trợ lập trình cho ESP và các module sử dụng trong dự án

Text

Description automatically generated

Hình . Các thư viện sử dụng trong lập trình ESP

* Thư viện “ESP8266WiFI.h”[7]: Thư viện "ESP8266WiFi.h" là một thư viện được sử dụng trong việc lập trình điều khiển ESP8266 để kết nối với mạng Wi-Fi. Thư viện này cung cấp các chức năng để cấu hình và kết nối với các mạng Wi-Fi, thường được sử dụng với các ứng dụng Arduino IDE và các nền tảng phát triển vi điều khiển khác để thêm tính năng kết nối Wi-Fi vào các dự án. Nó cho phép các nhà phát triển tạo ra các dự án có thể kết nối với Internet và tương tác với các dịch vụ trực tuyến, chẳng hạn như các máy chủ web, các nền tảng đám mây và các dịch vụ IoT (Internet of Things). Thư viện này là thiết yếu đối với nhiều dự án dựa trên Wi-Fi sử dụng module ESP8266.
* Thư viện “ESP8266HTTPClient.h”[8]: Thư viện "ESP8266HTTPClient.h" là một thư viện được sử dụng trong việc lập trình điều khiển ESP8266 để gửi và nhận các yêu cầu HTTP. Nó cho phép ESP8266 thực hiện các yêu cầu HTTP GET, POST, PUT, DELETE và các yêu cầu khác đến các máy chủ web, các API trực tuyến và các dịch vụ khác. Đồng thời nó cung cấp các chức năng để thiết lập các thông số của yêu cầu HTTP, bao gồm các thông số như URL, tham số yêu cầu, tiêu đề yêu cầu và nội dung yêu cầu. Nó cũng cho phép xử lý các phản hồi HTTP và truy cập các thông tin trong các phản hồi, bao gồm các mã trạng thái, tiêu đề phản hồi và nội dung phản hồi.
* Thư viện “ArduinoJson.h”[9]: Thư viện ArduinoJson.h là một thư viện mã nguồn mở cho phép thao tác dữ liệu JSON trên nền tảng ESP hoặc các dòng vi điều khiển khác. Thư viện này cung cấp các hàm và lớp đối tượng để phân tích, tạo và xử lý các đối tượng JSON. JSON là định dạng dữ liệu phổ biến trong các ứng dụng web và IoT, vì vậy ArduinoJson.h là một thư viện hữu ích để xử lý dữ liệu này trong các ứng dụng nhúng.

Thư viện ArduinoJson.h cho phép người dùng dễ dàng truy cập và sử dụng các giá trị JSON, như đối tượng (object), mảng (array) và các giá trị cơ bản như số nguyên, số thực và chuỗi. Nó cũng hỗ trợ các phép toán như thêm, xóa và sửa đổi các giá trị JSON.

* Thư viện “SoftwareSerial.h”[10]: Thư viện SoftwareSerial.h là một thư viện của Arduino IDE cho phép truyền và nhận dữ liệu qua các chân GPIO không hỗ trợ giao tiếp chuẩn như UART hay SPI. Thư viện này cung cấp một lớp đối tượng để tạo và điều khiển một kết nối serial bằng phần cứng bằng cách sử dụng phần mềm trên các chân GPIO. Điều này cho phép các module không hỗ trợ giao tiếp chuẩn như GPS, Bluetooth hay RFID được kết nối với ESP thông qua các chân GPIO.Nó hỗ trợ đa kết nối (multi-instance), có nghĩa là bạn có thể tạo nhiều kết nối serial trên một board ESP.
* Thư viện “DHT.h”[11]: Thư viện DHT.h là một thư viện của Arduino IDE cho phép đọc các giá trị đo từ các cảm biến nhiệt độ và độ ẩm của dòng cảm biến DHT (DHT11, DHT22 và các phiên bản khác). Thư viện DHT.h cung cấp các hàm đơn giản để đọc giá trị nhiệt độ và độ ẩm từ cảm biến DHT. Nó cũng hỗ trợ đọc các giá trị số và chuỗi, cho phép người dùng tùy chỉnh các đơn vị đo và giá trị chính xác của cảm biến. Thư viện này cũng có thể được sử dụng để đọc giá trị từ nhiều cảm biến DHT cùng một lúc.
* Thư viện “IRemoteESP8266.h”[12]: Thư viện "IRremoteESP8266.h" là một thư viện mã nguồn mở của Arduino IDE, được sử dụng để điều khiển các thiết bị từ xa bằng các tín hiệu hồng ngoại (IR). Thư viện này hỗ trợ các dòng vi điều khiển ESP8266 và ESP32 và cung cấp các hàm để nhận và gửi các tín hiệu IR.

Thư viện "IRremoteESP8266.h" có thể được sử dụng để tạo một bộ điều khiển từ xa cho các thiết bị điện tử như đèn, máy lạnh, tivi, đầu đĩa DVD, và nhiều thiết bị khác. Nó cũng có thể được sử dụng để mô phỏng tín hiệu IR để kiểm tra và thử nghiệm các thiết bị đóng vai trò là thiết bị nhận.

Thư viện này có tính năng phong phú, hỗ trợ nhiều loại mã hồng ngoại khác nhau như NEC, Sony, RC5, RC6, và nhiều loại khác. Nó cũng cung cấp các tính năng linh hoạt để đáp ứng nhu cầu sử dụng của người dùng, bao gồm việc tùy chỉnh độ dài và thời gian nghỉ giữa các tín hiệu.

* Thư viện “IRsend.h”[13]: Thư viện "IRsend.h" là một thư viện mã nguồn mở của Arduino IDE, được sử dụng để gửi các tín hiệu hồng ngoại (IR) để điều khiển các thiết bị từ xa. Thư viện này cung cấp các hàm để tạo ra các chuỗi tín hiệu IR có thể gửi đến các thiết bị điện tử như TV, máy lạnh, đầu đĩa DVD, hệ thống giải trí và nhiều thiết bị khác.

Thư viện "IRsend.h" được sử dụng kết hợp với một cảm biến hồng ngoại để đọc và mã hóa tín hiệu hồng ngoại, sau đó gửi chuỗi tín hiệu đã được mã hóa đến thiết bị mục tiêu. Thư viện này cung cấp một loạt các hàm để tạo ra các chuỗi tín hiệu hồng ngoại từ các mã khác nhau được sử dụng bởi các hãng sản xuất thiết bị khác nhau.

Thư viện "IRsend.h" cũng cung cấp các chức năng bổ sung để tùy chỉnh độ dài và thời gian nghỉ giữa các tín hiệu, cho phép người dùng tùy chỉnh các chuỗi tín hiệu hồng ngoại để phù hợp với các thiết bị khác nhau. Nó cũng có thể được sử dụng để gửi các tín hiệu hồng ngoại tùy chỉnh.

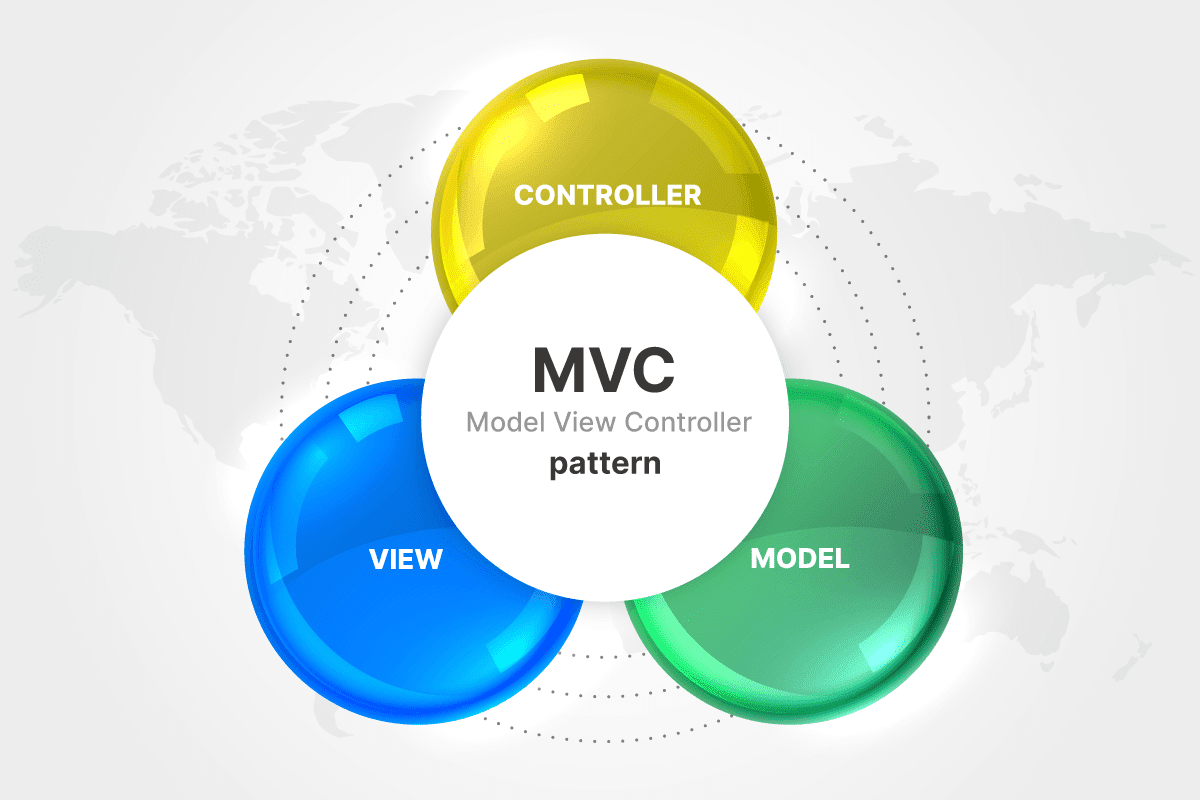
* Thư viện “ir\_Panasonic.h”[14]: Thư viện "ir\_Panasonic.h" là một thư viện mã nguồn mở của Arduino IDE. Thư viện này cung cấp các hàm để giải mã các chuỗi tín hiệu hồng ngoại từ các điều khiển từ xa Panasonic, bao gồm các tín hiệu điều khiển từ xa của TV, đầu đĩa DVD, hệ thống âm thanh, v.v. Nó cung cấp một loạt các hàm để giải mã các chuỗi tín hiệu hồng ngoại và trích xuất thông tin từ các tín hiệu đó, cho phép người dùng tùy chỉnh các chức năng của thiết bị mục tiêu.

## 4.2 Công nghệ sử dụng cho phần mềm

Phần mềm gồm phần giao diện điều khiển(Front-end), phần code xử lý logic điều khiển, liên kết với cơ sở dữ liệu(Back-end) và phần cơ sở dữ liệu (Database).

Như có nói ở phần thiết kế WebServer, trong dự án em có sử dụng 2 framework của microsoft là “Microsoft.AspNetCore.App” và “Microsoft.NetCore.App”, hỗ trợ xây dựng dự án theo mô hình MVC (Model-View-Controller). Phần CSDL em sử dụng SQL Server và phần điều khiển bằng giọng nói em sử dụng “Speech Web API” của google để nhận diện giọng nói tiếng Việt.

### 4.2.1 Giới thiệu về mô hình MVC



Hình . Mô hình MVC

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm được sử dụng để thiết kế và phát triển các ứng dụng web và phần mềm máy tính.

Mô hình MVC bao gồm ba thành phần chính:

* Model: Đại diện cho dữ liệu và logic xử lý dữ liệu. Nó đóng vai trò là lớp mô hình để lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu.
* View: Đại diện cho giao diện người dùng, nó hiển thị dữ liệu cho người dùng. Nó là thành phần đầu tiên mà người dùng nhìn thấy khi sử dụng ứng dụng.
* Controller: Là thành phần trung gian giữa Model và View, quản lý các sự kiện từ View và thay đổi Model tương ứng. Nó điều khiển và quản lý tất cả các hoạt động của ứng dụng, như điều hướng các truy vấn đến Model để lấy dữ liệu hoặc cập nhật View để hiển thị dữ liệu mới.

Mô hình MVC giúp tách biệt logic xử lý dữ liệu (Model) và giao diện người dùng (View) thành các thành phần riêng biệt, từ đó giúp dễ dàng bảo trì và phát triển ứng dụng hơn.

### 4.2.2 Framework “Microsoft.AspNetCore.App” và “Microsoft.NetCore.App”

"Microsoft.AspNetCore.App" và "Microsoft.NetCore.App" đều là các framework của Microsoft dùng để xây dựng ứng dụng web và ứng dụng dòng lệnh trên nền tảng .NET Core.

"Microsoft.AspNetCore.App" là một framework cho phép phát triển ứng dụng web ASP.NET Core, bao gồm các tính năng như middleware, routing, dependency injection, và các công cụ xử lý HTTP request/response. Đây là framework được sử dụng phổ biến nhất cho các ứng dụng web ASP.NET Core.

Trong khi đó, "Microsoft.NetCore.App" là một framework cung cấp các thành phần cơ bản của .NET Core, bao gồm các lớp cơ bản và các tính năng hệ thống như thread, file IO, network IO, và bộ nhớ đệm. Framework này được sử dụng chủ yếu cho các ứng dụng dòng lệnh và các ứng dụng đơn giản không có yêu cầu phức tạp về web.

Các framework này cũng được cung cấp bởi Microsoft qua NuGet, cho phép bạn cập nhật phiên bản một cách dễ dàng.

### 4.2.3 SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) do Microsoft phát triển. Nó được sử dụng để lưu trữ, quản lý và truy xuất dữ liệu từ các ứng dụng phía máy chủ hoặc các trang web. SQL Server có khả năng hỗ trợ các loại dữ liệu khác nhau bao gồm dữ liệu văn bản, hình ảnh, âm thanh và video.

SQL Server được phát hành cho nhiều phiên bản khác nhau, từ phiên bản miễn phí như SQL Server Express đến phiên bản cao cấp như SQL Server Enterprise. Các phiên bản khác nhau của SQL Server có các tính năng và quyền lợi khác nhau, cho phép người dùng chọn phiên bản phù hợp với nhu cầu của họ.

SQL Server hỗ trợ ngôn ngữ truy vấn SQL (Structured Query Language), được sử dụng để truy xuất và quản lý cơ sở dữ liệu. Nó cũng có thể tích hợp với các công nghệ khác

của Microsoft như .NET Framework, Visual Studio và Azure để tạo ra các ứng dụng web và dịch vụ trực tuyến phức tạp.

### 4.2.4 Sử dụng “API SpeechRecognition” để nhận diện giọng nói tiếng Việt

API SpeechRecognition là một API của trình duyệt web cho phép ứng dụng web nhận dạng giọng nói của người dùng và chuyển đổi nó thành văn bản. Nó cho phép người dùng tương tác với trang web thông qua giọng nói mà không cần sử dụng bàn phím hoặc chuột.

SpeechRecognition API cung cấp các sự kiện và phương thức cho phép ứng dụng web ghi âm âm thanh từ micro và phân tích giọng nói thành văn bản. Nó hỗ trợ nhiều ngôn ngữ và các tùy chọn khác nhau cho phép người dùng tùy chỉnh cách ghi âm và phân tích giọng nói.

API SpeechRecognition được hỗ trợ trên hầu hết các trình duyệt web hiện đại, bao gồm Google Chrome, Firefox và Safari.

Graphical user interface, text

Description automatically generated

Hình . Mã nguồn khởi tạo “API SpeechRecognition” nhận dạng giọng nói tiếng việt

Biến **“***SpeechRecognition***”** được khởi tạo bằng cách kiểm tra xem trình duyệt có hỗ trợ API **“***SpeechRecognition***”** của HTML5 hay không. Sau đó, một đối tượng **“***recognition***”** của lớp **“***SpeechRecognition***”** được tạo ra để sử dụng cho các chức năng nhận diện giọng nói. Đối tượng **“***synth***”** của lớp **“***speechSynthesis***”** được tạo để tổng hợp giọng nói trên trình duyệt web. Biến “*recognition.lang*” được thiết lập để chỉ định ngôn ngữ mà nhận diện giọng nói sẽ hoạt động, ở đây là tiếng Việt. Biến **“***recognition.continuous***”** được thiết lập để đảm bảo rằng nhận diện giọng nói chỉ diễn ra một lần và sau đó dừng.

# CHƯƠNG 5. KIỂM THỬ VÀ ĐÁNH GIÁ

## 5.1 Kiểm thử phần giao diện điều khiển

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thiết bị | Thông tin thiết bị | Giao diện | Chức năng |
| 1 | ThinkPad T14 | 14’’ Window 10 | ĐẠT | 6/6 |
| 2 | HP Victus 16 | 16’’ Window 11 | ĐẠT | 6/6 |
| 3 | Iphone 12 | 6.1’’ IOS 16 | ĐẠT | 5/6 |
| 4 | Xiaomi redmi note 9 | 6.53’’ Android 10 | ĐẠT | 5/6 |

## 5.2 Kiểm thử phần mạch

Cho sản phẩm cấp nguồn liên tục trong 1 ngày, kiểm tra kết quả gửi nhận dữ liệu và điều khiển các thiết bị tại 1 thời điểm ngẫu nhiên vào mỗi ngày.

Thời gian kiểm nghiệm từ 9h00 ngày 27/ 02/ 2023 đến 9h00 ngày 28/02/2023 ta thu được bảng kết quả:

* Lần 1: 9h05 ngày 27/ 02/ 2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chức năng | Đèn 1 | Đèn 2 | Loa | Điều hòa | Hiển thị nhiệt độ-độ ẩm | Cảnh báo cháy |
| Bật/ tắt | Bật | Bật | Bật | Bật | 23 – 80 | Tắt |
| Các chức năng khác | None | Màu sắc: Đỏ | Pause  Unmute | Nhiệt độ 25  Mode Cool  Speed: Auto | None | None |
| Độ trễ | <2s | <2s | <2s | <2s | <2s | <2s |

* Lần 2: 9h10 ngày 27/ 02/ 2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chức năng | Đèn 1 | Đèn 2 | Loa | Điều hòa | Hiển thị nhiệt độ-độ ẩm | Cảnh báo cháy |
| Bật/ Tắt | Tắt | Bật | Bật | Bật | 23 – 80 | Bật |
| Các chức năng khác | None | Màu sắc: Xanh lam | Play  Unmute | Nhiệt độ 27  Mode Dry  Speed: 1 | None | None |
| Độ trễ | <2s | <2s | <2s | <2s | <2s | <2s |

* Lần 3: 11h30 ngày 27/ 02/ 2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chức năng | Đèn 1 | Đèn 2 | Loa | Điều hòa | Hiển thị nhiệt độ-độ ẩm | Cảnh báo cháy |
| Bật/ Tắt | Bật | Bật | Bật | Bật | 24 – 80 | Tắt |
| Các chức năng khác | None | Màu sắc: Xanh lục | Play  Mute | Nhiệt độ 24  Mode Auto  Speed: 2 | None | None |
| Độ trễ | <2s | <2s | <2s | <2s | <2s | <2s |

* Lần 4: 11h50 ngày 27/ 02/ 2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chức năng | Đèn 1 | Đèn 2 | Loa | Điều hòa | Hiển thị nhiệt độ-độ ẩm | Cảnh báo cháy |
| Bật/ Tắt | Tắt | Tắt | Bật | Bật | 26 – 75 | Tắt |
| Các chức năng khác | None |  | Play  Unmute | Nhiệt độ 24  Mode Heat  Speed: 3 | None | None |
| Độ trễ | <2s | <2s | <2s | <2s | <2s | <2s |

* Lần 5: 13h20 ngày 27/ 02/ 2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chức năng | Đèn 1 | Đèn 2 | Loa | Điều hòa | Hiển thị nhiệt độ-độ ẩm | Cảnh báo cháy |
| Bật/ Tắt | Tắt | Bật | Tắt | Tắt | 26 – 75 | Tắt |
| Các chức năng khác | None | Màu sắc: Đỏ |  |  | None | None |
| Độ trễ | <2s | <2s | <2s | <2s | <2s | <2s |

* Lần 6: 06h30 ngày 28/ 02/ 2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chức năng | Đèn 1 | Đèn 2 | Loa | Điều hòa | Hiển thị nhiệt độ-độ ẩm | Cảnh báo cháy |
| Bật/ Tắt | Bật | Bật | Bật | Bật | 22 – 80 | Bật |
| Các chức năng khác | None | Màu sắc: Xanh lục | Play  Unmute | Nhiệt độ: 26  Mode: Cool  Speed: 4 | None | None |
| Độ trễ | <2s | <2s | <3s | <3s | <2s | <2s |

* Lần 7: 09h00 ngày 28/ 02/ 2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Chức năng | Đèn 1 | Đèn 2 | Loa | Điều hòa | Hiển thị nhiệt độ-độ ẩm | Cảnh báo cháy |
| Bật/ Tắt | Tắt | Tắt | Bật | Tắt | 24 – 80 | Tắt |
| Các chức năng khác | None |  | Pause  Unmute |  | None | None |
| Độ trễ | <2s | <2s | <3s | <2s | <2s | <2s |

## 5.3 Đánh giá

* Sản phẩm chạy khá ổn định và chính xác
* Độ trễ khá thấp
* Phần điều khiển bằng giọng nói trên laptop chạy tương đối ổn định, trên điện thoại chức năng này chưa hoạt động (có thể do trình duyệt trên điện thoại chưa thực sự tương thích).
* Giá thành hoàn thiện sản phẩm rẻ (khoảng 200.000đ).

# CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐỐNG GÓP NỔI BẬT

## 6.1 Xây dựng một WebServer riêng để vận hành chức năng

Đa số các sản phẩm hiện nay đều sử dụng 1 số Server hay các dịch vụ đám mây hỗ trợ sẵn như “Firebase”, “Blynk”, v.v. để xây dựng. Việc sử dụng các công cụ đó sẽ khó để phát triển các chức năng một cách linh hoạt vì phải phụ thuộc vào nhà phát triển các công cụ hỗ trợ sẵn đó. Hơn nữa khi xảy ra sự cố việc bảo trì và khắc phục sự cố sẽ gặp rất nhiều khó khăn. Việc xây dựng 1 WebServer riêng sẽ khắc phục được các vấn đề đó, đồng thời việc phân chia các chức năng và thiết bị điều khiển riêng biệt sẽ dễ dàng quản lý và phát triển thêm nhiều chức năng như: quản lý thời gian sử dụng của từng thiết bị để tính ra điện năng tiêu thụ hàng tháng hoặc nắm được tuổi thọ của các thiết bị hay sẽ lưu lại được giá trị nhiệt độ của từng thời điểm trong ngày để có thể truy xuất nhiệt độ trung bình khi có nhu cầu, v.v.

## 6.2 Xây dựng dự án theo mô hình MVC (model-view-controller)

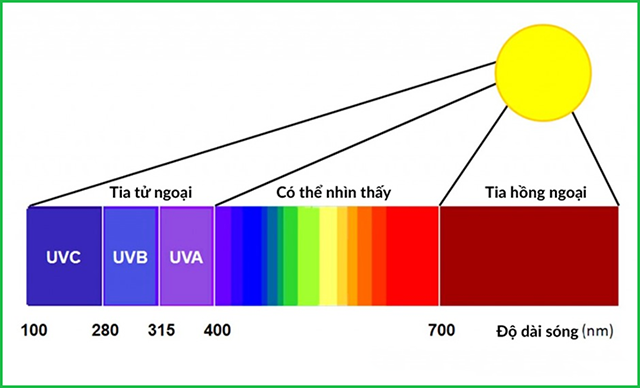
Như đã giới thiệu ở chương các công nghệ sử dụng, mô hình MVC là thực sự hữu hiệu, là hướng phát triển chủ yếu của các dự án về web hiện tại. Việc áp dụng vào dự án smart home cũng sẽ mang lại nhiều tiềm năng cho hướng phát triển dự án lâu dài. Một số ưu điểm nổi bật khi mang mô hình MVC vào dự án smart home:

* Dễ dàng mở rộng và phát triển :
* Khi có một thiết bị điện mới được thêm vào hoặc thay thế thiết bị cũ sẽ dễ dàng được cập nhập vào hệ thống.
* Việc nâng cấp về giao diện điều khiển hay nâng cấp, thêm mới các chức năng điều khiển cũng rất dễ dàng.
* Việc tách biệt từng phần cũng tăng tính bảo mật cho dự án.

## 6.3 Xấy dựng chức năng điều khiển thiết bị bằng sóng hồng ngoại

Sóng hồng ngoại đã được áp dùng cho việc điều khiển các thiết bị điện tử từ khá lâu và ngày nay nó vẫn được sử dụng trong hầu hết các thiết bị điện, từ đó ta có thể thấy tầm quan trọng của nó trong việc điều khiển các thiết bị điện.

### 6.3.1 Giới thiệu cơ bản về sóng hồng ngoại



**Hình 6.1 Độ dài sóng của các tia sáng trong thực tế**

Sóng hồng ngoại (IR) là một dạng sóng điện từ có bước sóng từ khoảng 700 nm đến 1 mm và được sử dụng rộng rãi trong việc điều khiển các thiết bị điện tử như TV, đầu đĩa DVD, máy chiếu, điều hòa không khí và các thiết bị gia đình khác.

* Trong thiết bị điện tử, bộ điều khiển sử dụng tín hiệu sóng hồng ngoại để truyền lệnh điều khiển đến thiết bị. Tín hiệu sóng hồng ngoại được tạo ra bởi một bộ phát sóng hồng ngoại, thường được tích hợp trên bề mặt của bộ điều khiển. Khi người dùng ấn nút trên bộ điều khiển, bộ phát sóng hồng ngoại sẽ phát ra tín hiệu sóng hồng ngoại có mã hóa chứa lệnh điều khiển tương ứng. Thiết bị nhận sẽ bắt sóng hồng ngoại và xử lý tín hiệu để thực hiện lệnh điều khiển.
* Sóng hồng ngoại là một phương tiện truyền thông rất đáng tin cậy và phổ biến trong việc điều khiển các thiết bị điện tử. Tuy nhiên, sóng hồng ngoại có một hạn chế là chỉ hoạt động trong phạm vi tầm nhìn trực tiếp giữa bộ phát và thiết bị nhận. Do đó, nếu có vật cản ở giữa thì sóng hồng ngoại không thể truyền được và việc điều khiển thiết bị sẽ không thành công.

### 6.3.2 Những lợi ích khi kết hợp sóng hồng ngoại với hệ thông smart home

* Như đã đề cập ở trên, các thiết bị điện ngày nay đa số vẫn sử dụng remote hồng ngoại để điều khiển từ xa nên việc tích hợp remote hồng ngoại cho hệ thống smart home sẽ giúp cho người dùng dễ dàng tiếp cận hơn với công nghệ IoT này bởi lẽ nếu muốn áp dụng nó, người dùng vẫn có thể sử dụng lại các thiết bị điện sẵn có, không phải thay mới hay lắp đặt thêm một thiết bị hỗ trợ thứ 3 nào, từ đó sẽ giảm chi phí lắp đặt và giúp người dùng dễ dàng sử dụng công nghệ mới này vì nó vẫn kế thừa công nghệ cũ mà người dùng đã quen.
* So với các loại sóng khác thì sóng hồng ngoại tiết kiệm năng lượng hơn, ít xảy ra sự cố hơn và an toàn hơn.

## 6.4 Tích hợp cảm biến nhiệt độ - độ ẩm cho sản phẩm

Nhận thấy một vấn đề thực tế trong cuộc sống khi sử dụng các thiết bị liên quan đến nhiệt độ, độ ẩm như điều hòa. Thông thường, người lớn tuổi và trẻ nhỏ thường không để ý đến nhiệt độ của điều hòa sau một thời gian sử dụng. Ban đầu nhiệt độ của phòng cao, họ bật và để nhiệt độ của điều hòa xuống mức thấp. Sau 1 khoảng thời gian nhiệt độ của phòng đã hạ nhưng điều hòa vẫn liên tục chạy ở mức nhiệt thấp sẽ gây ảnh hưởng đến sức khuẻ của người sử dụng.

Việc sử dụng cảm biến nhiệt độ độ ẩm cho sản phẩm giúp ta đo được nhiệt độ và độ ẩm của môi trường, từ đó có thể điều khiển các chức năng của điều hòa cho phù hợp.

## 6.5 Tích hợp cảm biến tia lửa cho sản phẩm

Trong thời gian gần đây, không ít các vụ cháy nổ xảy ra ở các hộ gia đình gây ra những hậu quả nghiêm trọng cả về tính mạng, sức khỏe con người lẫn của của cải vật chất. Một trong những nguyên nhân làm tăng mức độ nguy hiểm của đám cháy là việc đám cháy cháy lan đến các thiết bị điện công suất lớn đang hoạt động. Hiểu được điều đó, em đã tích hợp cảm biến lửa cho sản phẩm để khi có cháy nổ xảy ra sẽ tự động bật và phát cảnh báo ra loa, đồng thời sẽ tự động tắt các thiết bị điện công suất lớn, góp phần làm giảm thiệt hại nếu có cháy nổ xảy ra.

# CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 7.1 Kết luận

Sau một thời gian tìm hiểu và phân tích nghiệp vụ, yêu cầu của đề tài, về tổng quan em đã hoàn thành 90% số yêu cầu ban đầu. Một số đóng góp chính là (i) Xây dựng một webserver riêng (ii) Cho phép điều khiển thiết bị thông qua sóng hồng ngoại (iii) Sử dụng mô hình MVC cho khả năng nâng cấp dễ dàng (iv) Tích hợp các cảm biến nhiệt độ - độ ẩm và cảm biến tia lửa giải quyết được những vấn đề thực tế (v) Lưu trạng thái của thiết bị vào database giúp hiển thị được trạng thái của thiết bị ở mọi client.

Trong quá trình làm đồ án, em đã học thêm được nhiều điều. Đầu tiên là học được cách xây dựng một mô hình dự án smart home thực tế.

Thứ hai, biết thêm nhiều kỹ năng về lập trình cho VĐK đặc biệt là các VĐK họ ESP có khả năng truy cập internet.

Thứ ba, từ việc tìm hiểu các kiến thức để tạo nên 1 giao diện điều khiển, tạo ra 1 webserver riêng cho dự án giúp em học hỏi được rất nhiều kiến thức mới về công nghệ, giúp em có được nhiều hướng phát triển bản thân hơn trong tương lai.

Hơn nữa, ĐATN không những giúp em học thêm các kiến thức chuyên ngành, các yêu cầu khi phân tích nghiệp vụ, lập kế hoạch sản phẩm. Hơn nữa, giúp em nhìn nhận được khả năng của bản thân, rút ra kinh nghiệm để tránh mắc phải những sai lầm sau này. Đây là những bài học vô cùng quý báu đối với em.

Lời cuối cùng, em xin cảm ơn thầy Nguyễn Quang Minh đã giúp đỡ em rất nhiều để có thể hoàn thành ĐATN này.

## 7.2 Hướng phát triển

Ý tưởng ban đầu cho việc xây dựng dự án này của em ngắm đến đối tượng khách hàng là các hộ gia đình. Mỗi một sản phẩm như trong dự án sẽ được lắp đặt ở một phòng của ngôi nhà và các phòng luôn đảm bảo có sự liên kết với nhau.

Trong thời gian tới, em sẽ tiếp tục nghiên cứu và phát triển dự án này để tối ưu và hoàn thiện hơn nữa.

Một số dự định trong tương lai:

* Xây dựng chức năng phân quyền, chức năng đăng nhập, tạo tài khoản điều khiển cho hệ thống.
* Nâng cấp giao diện điều khiển cho chuyên nghiệp và hiện đại hơn
* Tối ưu các sourse code nhằm tăng hiệu năng cho sản phẩm.
* Viết thêm nhiều chức năng mới như lưu trữ thông tin về thời gian sử dụng của từng thiết bị, về nhiệt độ tại các thời điểm đo được để tính toán ra các thông tin cần thiết cho người dùng như về tổng thời gian hoạt động của thiết bị để có thể tính được điện năng tiêu thụ, tính được mức nhiệt trung bình trong ngày,v.v.

# TÀI LIỆU THAM KHẢO

1. Jon Duckett. Wiley, "*Web Design with HTML, CSS, JavaScript and jQuery Set*", 2014.
2. Marco Schwartz, "*Mã hóa và giải mã với ESP8266 Wi-Fi Module: Một giải pháp IoT đơn giản*", Packt Publishing, 2018.
3. Lee Naylor. Apress, "*MVC 5 with Entity Framework 6 in Visual Studio 2015*", 2016.
4. <https://github.com/holetexvn/virtual-assistant>, truy nhập cuối cùng ngày 3/3/2023.
5. [Internet Of Things (IoT) : cho người mới bắt đầu (iotmakervn.github.io)](https://iotmakervn.github.io/iot-starter-book/#_esp8266), truy nhập cuối cùng ngày 3/3/2023.
6. [TA/Test1234.ino at a76aadef3a50d57e002ea2d49071f8a40eb82cff · fariefdswr/TA (github.com)](https://github.com/fariefdswr/TA/blob/a76aadef3a50d57e002ea2d49071f8a40eb82cff/Test1234/Test1234.ino), truy nhập cuối cùng ngày 3/3/2023.
7. [Arduino/ESP8266WiFi.h at master · esp8266/Arduino (github.com)](https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/src/ESP8266WiFi.h), truy nhập cuối cùng ngày 3/3/2023.
8. [Arduino/ESP8266HTTPClient.h at master · esp8266/Arduino (github.com)](https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266HTTPClient/src/ESP8266HTTPClient.h), truy nhập cuối cùng ngày 3/3/2023.
9. [arduino-libraries/Arduino\_JSON: Official JSON Library for Arduino (github.com)](https://github.com/arduino-libraries/Arduino_JSON), truy nhập cuối cùng ngày 3/3/2023.
10. [SoftwareSerial/SoftwareSerial.h at master · PaulStoffregen/SoftwareSerial (github.com)](https://github.com/PaulStoffregen/SoftwareSerial/blob/master/SoftwareSerial.h), truy nhập cuối cùng ngày 3/3/2023.
11. [DHT-sensor-library/DHT.h at master · adafruit/DHT-sensor-library (github.com)](https://github.com/adafruit/DHT-sensor-library/blob/master/DHT.h), truy nhập cuối cùng ngày 3/3/2023.
12. [IRremote: IRsend Class Reference (arduino-irremote.github.io)](https://arduino-irremote.github.io/Arduino-IRremote/classIRsend.html), truy nhập cuối cùng ngày 3/3/2023.
13. [IRremoteESP8266/IRsend.h at master · crankyoldgit/IRremoteESP8266 (github.com)](https://github.com/crankyoldgit/IRremoteESP8266/blob/master/src/IRsend.h), truy nhập cuối cùng ngày 3/3/2023.
14. [IRremoteESP8266/ir\_Panasonic.h at master · crankyoldgit/IRremoteESP8266 (github.com)](https://github.com/crankyoldgit/IRremoteESP8266/blob/master/src/ir_Panasonic.h), truy nhập cuối cùng ngày 3/3/2023.