

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC CNKT ĐIỆN TỬ - VIỄN THÔNG

**THIẾT KẾ MÔ HÌNH CÔNG TÁC CẢM ỨNG THÔNG MINH SỬ
DỤNG MODULE ESP32**

CBHD: ThS. Phạm Thị Thanh Huyền

Sinh viên: Nguyễn Đức Mạnh

Mã số sinh viên:

Hà Nội – 2022

MỤC LỤC

MỤC LỤC.....	1
LỜI CẢM ƠN	4
LỜI MỞ ĐẦU	5
CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG CÔNG TÁC CẢM ỨNG THÔNG MINH.	9
1.1 Tình hình nghiên cứu trong nước và quốc tế.	9
1.1.1 Tình hình nghiên cứu ngoài nước.	9
1.1.2 Tình hình nghiên cứu trong nước.	11
1.2 Thiết kế sơ đồ khối của công tác cảm ứng thông minh.	15
1.2.1 Yêu cầu thiết kế công tác cảm ứng thông minh.....	15
1.2.2 Sơ đồ khối hệ thống.	15
1.3 Tổng kết chương 1.	16
CHƯƠNG 2: THIẾT KẾ MÔ HÌNH.....	17
2.1 Sơ đồ nguyên lý của hệ thống công tác cảm ứng thông minh.	17
2.1.1 Khối nguồn.....	17
2.1.2 Khối xử lý trung tâm.	18
2.1.3 Khối Relay.	22
2.1.4 Khối nút nhấn.....	24
2.1.5 Sơ đồ nguyên lý toàn mạch.....	25
2.2 Lưu đồ thuật toán của hệ thống.....	26
2.3 Thiết kế mạch in.....	27
2.4 Mô hình thực tế.	29
2.5 Kết luận chương 2.	29
CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM.....	30
3.1 Phân tích, giải thích kết quả thực nghiệm.....	30

3.2 Tính năng và hiệu quả sử dụng của sản phẩm **Error! Bookmark not defined.**

3.2.1 Tính năng và hiệu quả sử dụng của sản phẩm	30
3.2.2 Tính ứng dụng, mức độ an toàn và tác động của sản phẩm thiết kế tới môi trường, kinh tế và xã hội.....	30
3.3 Hướng dẫn sử dụng sản phẩm thiết kế.....	31
3.4 Kết luận chương 3	33
KẾT LUẬN	35
TÀI LIỆU THAO KHẢO	37
PHỤ LỤC.....	38

DANH MỤC HÌNH ẢNH. BẢNG BIỂU

Hình 1.1. Công tắc cảm ứng Xiaomi	9
Hình 1.2. Công tắc cảm ứng thương hiệu Schneider.....	11
Hình 1.3. Công tắc cảm ứng của FPT Smart Home	12
Hình 1.4. Công tắc thông minh Vconnex	13
Hình 1.5. Sơ đồ khối của hệ thống.	15
Hình 2.1. Mạch nguyên lý khối nguồn.	17
Hình 2.2. Module Hilink.....	18
Hình 2.3 Module ESP32	18
Hình 2.4. Sơ đồ nguyên lý khối xử lý trung tâm.	21
Hình 2.5. Khối Relay.	22
Hình 2.6. Relay 5V 10A	22
Hình 2.7. Mạch nguyên lý khối nút nhấn	24
Hình 2.8. IC cảm biến chạm	24
Hình 2.9. Sơ đồ nguyên lý của hệ thống.....	25
Hình 2.10 Lưu đồ thuật toán.	26
Hình 2.11. Sơ đồ mạch in PCB 2D.....	27
Hình 2.12. Sơ đồ mạch in PCB 3D.....	28
Hình 2.13 Mô hình thực tế.....	29
Hình 3.1. Truy cập internet.....	31
Hình 3.2. Điều khiển hệ thống bằng nút nhấn	32
Hình 3.3. File Index.html.....	32
Hình 3.4. Giao diện điều khiển hệ thống	33
 Bảng 2.1. Thông số kỹ thuật relay 5V [5]	23
Bảng 2.2. Thông số kỹ thuật IC cảm biến chạm.....	24

LỜI CẢM ƠN

Trước tiên em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất của em tới ThS. Phạm Thị Thanh Huyền, người đã hướng dẫn tận tình và hiệu quả, thường xuyên động viên chúng em trong quá trình hoàn thiện đề tài. Người đã dành cho em sự ưu ái nhất trong thời gian học tập, nghiên cứu cũng như quá trình hoàn thành đồ án tốt nghiệp.

Em xin cảm ơn các Thầy giáo, Cô giáo trong khoa Điện Tử trường Đại học Công Nghiệp Hà Nội cùng tất cả thành viên lớp Điện tử 04 – K13 đã tạo điều kiện và đóng góp ý kiến để em hoàn thành tốt đồ án tốt nghiệp.

Mặc dù em đã cố gắng để hoàn thành thực tập nhưng do kiến thức cũng như khả năng còn hạn hẹp nên quá trình thực hiện đề tài còn có sai sót. Rất mong nhận được sự góp ý và chỉ bảo của quý thầy cô.

Em xin chân thành cảm ơn!

Hà Nội, Ngày 26 tháng 03 năm 2022
Sinh viên thực hiện

Nguyễn Đức Mạnh

LỜI MỞ ĐẦU

Lý do chọn đề tài

Internet vạn vật, hay còn được gọi là IoT trong những năm gần đây đã phát triển và được ứng dụng rất nhiều trong cuộc sống hàng ngày của chúng ta. Công nghệ này mang tới cho con người sự tiện lợi, tiện nghi khi sử dụng các sản phẩm về điện tử. Trong công nghiệp, IoT mang lại sự dễ dàng trong điều khiển và giám sát các thiết bị, máy móc và quy trình tự động.

Thuật ngữ IoT đề cập đến những thứ mà chúng ta sử dụng hàng ngày cũng kết nối với internet, cho phép chúng ta kiểm soát hoặc nhận dữ liệu về từ điện thoại thông minh hoặc máy tính của mình.

Với sự phát triển của xã hội, khoa học kỹ thuật nói chung, việc tích hợp IoT vào vi điều khiển ngày càng được ứng dụng ở hầu hết các lĩnh vực. Cùng với đó khái niệm nhà thông minh (Smart home) không còn xa lạ với tất cả mọi người. Trong ngôi nhà thông minh, ngoài hệ thống đèn thông minh ra thì công tắc điện thông minh có thể nói là một phần quan trọng của nhà thông minh và là thiết bị không thể thiếu.

Xã hội ngày càng phát triển, đời sống con người được nâng cao, bên cạnh nhu cầu vui chơi giải trí thì sự tiện nghi cho ngôi nhà luôn là yếu tố hàng đầu được mọi người quan tâm và đầu tư. Hiểu rõ được điều đó, hiện nay trên thị trường có những bộ sản phẩm thông minh cho ngôi nhà như chuông cửa có màn hình, khóa vân tay... Đặc biệt, công tắc cảm ứng chạm tay là một trong những thiết bị tiên tiến, tiện lợi được nhiều khách hàng lựa chọn.

Công tắc cảm ứng chạm tay là sản phẩm thông minh được thiết kế dựa trên sự tinh tế, khoa học, đầy sáng tạo nhưng vô cùng tiện lợi và an toàn. Với mặt cảm ứng được làm bằng kính cường lực, không bị trầy xước hay phai màu theo thời gian. Vỏ công tắc được làm bằng chất liệu nhựa, chịu được nhiệt độ cao và chống nước, vì vậy khi tay ướt, bạn vẫn bật công tắc bình thường mà không lo bị điện giật.

Sử dụng công nghệ cảm ứng điện dung không những thực hiện thao tác cảm ứng một cách nhanh chóng, dễ dàng mà còn cho phép người dùng điều chỉnh độ sáng trên máy tính bảng, điện thoại kết hợp với vòng tròn tỏa sáng của đèn led, làm cho ngôi nhà trở nên sang trọng, hiện đại, ấm áp hơn.

Ngoài ra, thiết bị được thiết kế tương thích với đế âm tường sẵn có trên thị trường nên việc lắp đặt, thay thế không ảnh hưởng đến kiến trúc thượng tầng của ngôi nhà. Luôn bảo vệ tình trạng quá nhiệt và quá tải, mang đến sự an toàn cho ngôi nhà. Đồng thời, với thiết kế đặc biệt, công tắc cảm ứng luôn giữ cho ngôi nhà của bạn một cách yên tĩnh, không phát ra tiếng bật – tắt cũng như với điều khiển trực tiếp từ một chuyên mạch.

Không những thế, công tắc cảm ứng còn phát huy tính tiện lợi của mình, mang đến sự hài lòng nhất cho gia chủ, đó là không cần cấp nguồn, lắp đặt một cách nhanh chóng, chỉ cần thay thế cho công tắc cũ mà không cần phải mắc mới hệ thống điện.

Chỉ cần chạm nhẹ tay lên bề mặt công tắc điện là ngay lập tức cảm ứng nhận được tín hiệu chạm để mở hoặc tắt đèn. Hiện nay trên thị trường có khá nhiều loại công tắc thông minh. Mỗi loại sẽ có những tính năng và cách hoạt động đặc biệt khác nhau.

Trước thực tiễn ấy, em đã quyết định chọn đề tài **“Thiết kế mô hình công tắc cảm ứng thông minh sử dụng module ESP32”** nhằm tìm hiểu về vấn đề ứng dụng vi điều khiển ESP32 vào trong IoT và những ứng dụng của IoT vào trong cuộc sống hàng ngày.

Đối tượng nghiên cứu

Đối tượng nghiên cứu trong đề án này bao gồm: Thiết kế mạch vi điều khiển ESP32, tìm hiểu các chức năng, ứng dụng của vi điều khiển ESP32 vào trong IoT để điều khiển và giám sát.

Tìm hiểu cơ sở lý thuyết về mô hình công tắc cảm ứng thông minh nói chung và các loại công tắc cảm ứng tích hợp nhiều tiện ích nói riêng; tìm hiểu về nguyên lý của mô hình cảm ứng thông minh; cách đọc giá trị tín hiệu nhận

được từ cảm biến về vi điều khiển ESP32 và hiển thị qua LCD thông qua giao thức truyền thông I2C.

Mục đích nghiên cứu

Mục đích em đưa ra đề tài này nhằm ứng dụng IoT vào cá thiết bị điện tử giúp chúng trở nên thông minh hơn, giúp con người có thể điều khiển và giám sát các thiết bị bất cứ nơi đâu chỉ cần điện thoại hoặc máy tính có kết nối internet

Cũng cố kiến thức đã học, thu thập các kiến thức thực tiễn trong quá trình làm. Đồng thời đưa ra hướng phát triển sản phẩm ra thực tiễn sản xuất.

Phạm vi nghiên cứu

Sử dụng kiến thức đã học, nghiên cứu thiết kế hệ thống đơn giản nhằm đáp ứng đủ yêu cầu của hệ thống, thiết kế mô hình mini mô phỏng quá trình hoạt động của hệ thống. Trong đề tài này, em sẽ thực hiện thiết kế một hệ thống IoT đơn giản bao gồm 2 phần chính:

- Phần cứng có chứa vi điều khiển ESP32 để đưa ra các tín hiệu điều khiển một thiết bị.
- Một trang web để điều khiển và hiển thị trạng thái bật/tắt của bóng đèn bất cứ khi nào chỉ cần người dùng có điện thoại hoặc máy tính có kết nối với internet.

Ý nghĩa khoa học và thực tiễn của đề tài

Hiện nay hầu hết các sản phẩm điện tử đều có tích hợp việc điều khiển từ xa điển hình là dùng hồng ngoại. Tuy nhiên, các sản phẩm điện tử hiện này đều hướng tới việc tích hợp IoT với giao diện điều khiển thân thiện, dễ sử dụng.

- Thiết bị thông minh
- Hệ thống an ninh thông minh
- Trung tâm nhà thông minh
- Trợ lý thông minh

Chính vì lý do trên, em đưa ra tập trung vào lĩnh vực IOT với mục đích tìm hiểu và nghiên cứu về các ứng dụng IOT, giúp phát triển một số ứng dụng IOT trong cuộc sống hàng ngày và trong công nghiệp.

Với đề tài “**Thiết kế mô hình công tắc cảm biến thông minh sử dụng module ESP32**”, báo cáo em gồm các phần sau:

Chương 1: Tổng quan đề tài và cơ sở lý thuyết

Chương 2: Thiết kế mô hình công tắc cảm ứng thông minh

Chương 3: Kết luận và bài học kinh nghiệm

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG CÔNG TÁC CẢM ỨNG THÔNG MINH.

1.1 Tình hình nghiên cứu trong nước và quốc tế.

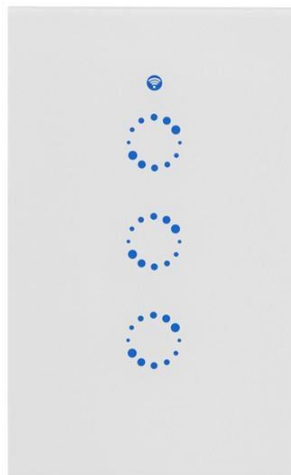
1.1.1 Tình hình nghiên cứu ngoài nước.

Hiện nay IOT là vấn đề được quan tâm rất nhiều trên thế giới, vì vậy các công ty lớn như Google, Amazon ... luôn mong muốn phát triển các hệ thống của họ nhằm cung cấp cho người dùng các thiết bị thông minh, tiện lợi và dễ sử dụng.

Vì vậy các nước lớn trên thế giới cũng quan tâm tới phát triển công tác cảm ứng thông minh phục vụ cho hệ thống ngôi nhà thông minh hiện đại và mang tính cách mạng công nghệ cao.

Thương hiệu Xiaomi.

Xuất hiện trên thị trường với nhiều sản phẩm đạt chất lượng cao và hoàn thiện trong từng chi tiết nên thương hiệu Xiaomi đến từ Trung Quốc được người dùng tin tưởng lựa chọn từ rất lâu. Điểm mạnh của thương hiệu này là sản xuất các công tắc thông minh có mức giá khá bình dân, đảm bảo an toàn bởi được làm từ những chất liệu cao cấp.



Hình 1.1. Công tắc cảm ứng Xiaomi

Bên cạnh đó thì với nhiều mẫu mã đẹp, thiết kế đa dạng để đáp ứng nhu cầu của người dùng cũng là điểm cộng. Và sự tiện lợi của Xiaomi thể hiện rõ qua các công tắc đơn giản, công suất cao và kiểu dáng bắt mắt. Nổi bật là mẫu công tắc Xiaomi Square Wall Switch có kết nối sóng Zigbee, công tắc ổn định cao hơn so với qua wifi nên trở thành sự lựa chọn của người dùng trong thời đại ngày nay

Thương hiệu Schneider

Thương hiệu Schneider đến từ Pháp nên chất lượng vượt trội hơn hẳn các thương hiệu khác. Schneider đã sản xuất ra rất nhiều thiết bị thông minh, đa dạng các dòng sản phẩm. Hoạt động hiệu quả với bộ điều khiển trung tâm, bảng điều khiển chi tiết, bộ phát sóng...

Công tắc thông minh được thiết kế sang trọng, nút bấm dạng tròn hoặc chữ nhật. Đế được làm từ kính dày. Hai loại phổ biến hiện nay là công tắc 2 nút hoặc nhiều nút.

Với những đường nét thiết kế tinh tế, sản phẩm phù hợp với các công trình cao cấp như chung cư cao cấp, biệt thự, nhà máy, xí nghiệp,...ưu tiên sử dụng sản phẩm thiết bị điện Schneider vì an toàn, độ bền cao và khả năng hoạt động ổn định, bền. Điểm cộng cho thương hiệu Schneider là lắp đặt trong môi trường khắc nghiệt thì sản phẩm vẫn có vẫn chịu được độ ẩm, độ ăn mòn cao.



Hình 1.2. Công tắc cảm ứng thương hiệu Schneider

Và điểm làm giúp thương hiệu Schneider đến với người dùng nhanh hơn đó là chất lượng tốt, giá không quá cao nhưng thiết kế hoàn thiện để mang đến cho người dùng sự trải nghiệm thú vị. Đồng thời, nếu so với các thương hiệu khác luôn khiến nhiều người phải lo lắng thì với những chế độ bảo hành dài hạn, cam kết hàng chính hãng Schneider đã hoàn toàn cho người dùng cảm giác thoải mái và yên tâm nhất trong suốt quá trình sử dụng.

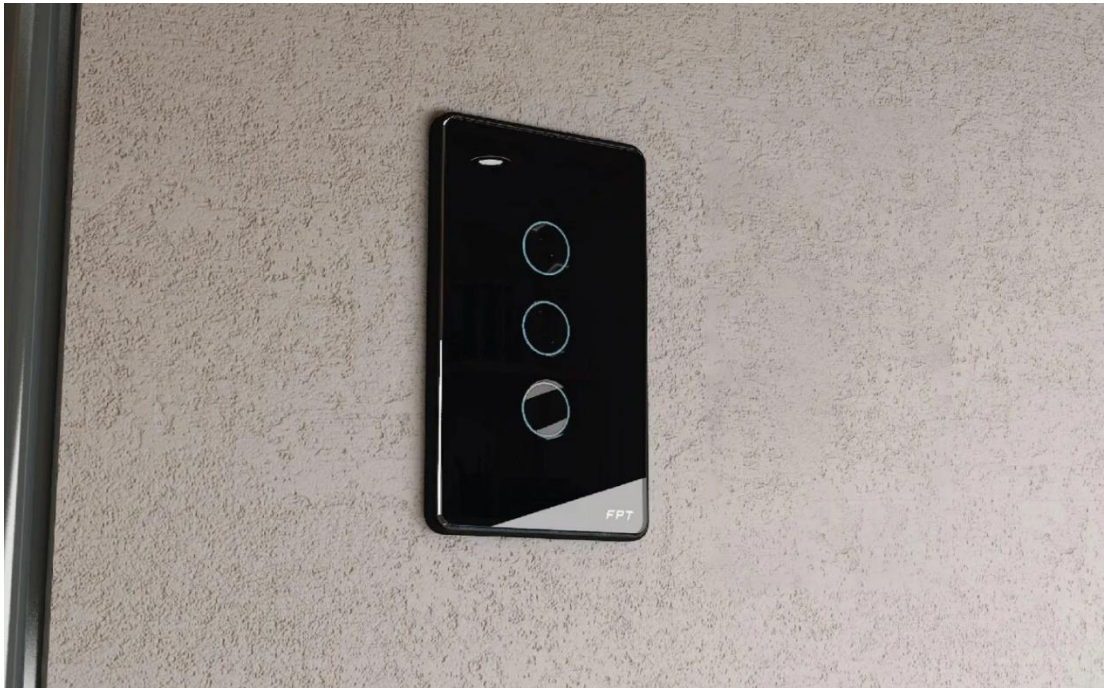
1.1.2 Tình hình nghiên cứu trong nước.

Không chỉ nước ngoài phát triển công tắc công cảm ứng thông minh, mà Việt Nam cũng có những công nghệ công tắc được sử dụng nhiều trong những ngôi nhà hiện đại.

Công tắc thông minh của FPT Smart Home.

Với sự lột xác về thiết kế so với phiên bản cũ cùng nhiều công nghệ mới được tích hợp, bộ sưu tập Athena do FPT Smart Home phát triển không chỉ mang đến nhiều tiện ích cho ngôi nhà thông minh, mà còn là điểm nhấn sang trọng, tinh tế cho mọi không gian.

Ở phiên bản mới, công tắc cảm ứng thông minh trong bộ sưu tập Athena được thiết kế viền kim loại, bo cạnh sắc nét cùng với 2 màu đen và trắng giúp tạo điểm nhấn cho mọi không gian. Bề mặt của công tắc Athena còn được trang bị kính cường lực, giúp chống trầy xước, chống bám vân tay. Đây là một trong những điểm thiết kế khiến bộ công tắc Athena trở nên nổi bật hơn hẳn so với phiên bản tiền nhiệm và cả những sản phẩm khác trên thị trường.



Hình 1.3. Công tắc cảm ứng của FPT Smart Home

Ra mắt thị trường từ năm 2018, Giải pháp nhà thông minh FPT Smart Home của [FPT Telecom](https://www.fpttelecom.vn) đã gây ấn tượng mạnh mẽ tới người dùng Việt bằng sự tích hợp nhiều tính năng thông qua các thiết bị: FPT Play Box S, cảm biến cửa FPT iHome, FPT Camera, đèn thông minh, khóa cửa thông minh, bộ điều khiển rèm...

Công tắc cảm ứng công suất cao tương thích với những thiết bị có công suất lớn như: Bình nước nóng, bếp từ, máy bơm, lò nướng,... đáp ứng đa dạng nhu cầu sử dụng của thị trường.

Ổ cắm âm tường mặt kính cường lực thiết kế đẹp, hỗ trợ đa chuẩn phích cắm. Công tắc cảm ứng rèm giúp điều khiển đóng mở rèm cửa từ xa, hẹn giờ... bằng smartphone hoặc trực tiếp bằng giọng nói. Ngoài ra, sản phẩm còn điều khiển các loại động cơ khác như: cửa cổng, cửa cuốn, cửa tự động.... thông dụng trên thị trường.

FPT Smart Home cũng đặc biệt chú trọng vào việc sử dụng các chất liệu chống cháy nổ cho bộ sưu tập, không chỉ đảm bảo được tính an toàn trong quá trình sử dụng mà hạn chế được các tình trạng rò điện, giật điện gây nguy hiểm.

FPT Smart Home là giải pháp “made in Việt Nam” hiếm hoi do FPT Telecom phát triển làm chủ nhiều mảng công nghệ từ: thiết bị, hạ tầng Internet, hệ thống Cloud đặt tại Data Center đạt chuẩn thiết kế Tier 3 của Uptime Institute... Tất cả đều được phát triển, tích hợp và triển khai bởi một nhà cung cấp duy nhất sẽ là bảo chứng chất lượng và giúp mang đến nhiều lợi ích cho người dùng khi được trải nghiệm hệ sinh thái toàn diện từ giải pháp tới bảo hành.

Công tắc thông minh Vconnex.



Hình 1.4. Công tắc thông minh Vconnex

Sử dụng công tắc cảm ứng Vconnex, các thiết bị điện trong nhà như đèn điện, bình nước nóng, rèm cửa,... trở nên thông minh, tiện lợi hơn. Người dùng có thể tạo các ngữ cảnh tự động theo cuộc sống riêng của gia đình, phù hợp với cuộc sống hiện đại.

Mang hàm lượng cao với các công nghệ IoT tiêu biểu như điện toán đám mây, điện toán biên, tự động cập nhật tính năng mới OTA, big data,... công tắc cảm ứng Vconnex có tính kế thừa cao. Hoạt động trên nền tảng IoT riêng biệt, công tắc cảm ứng Vconnex đảm bảo bảo mật dữ liệu và tối ưu tốc độ xử lý do toàn bộ thông tin được lưu trữ và xử lý tại Việt Nam.

Mặc dù mới gia nhập thị trường smarthome không lâu nhưng công tắc cảm ứng và thiết bị nhà thông minh Vconnex “được lòng” người tiêu dùng Việt. Mới đây nhất, trong khuôn khổ giải thưởng Tech Awards cho báo Vnexpress tổ chức, Vconnex gây bất ngờ khi là thương hiệu năm đầu tham gia nhưng nhận được số lượt bình chọn ấn tượng tại hạng mục Nền tảng nhà thông minh Việt xuất sắc.

Trong năm 2021, Vconnex liên tục gây tiếng vang với người tiêu dùng và giới chuyên môn trong ngành khi cùng lúc nhận Giải thưởng Thành phố Thông minh Việt Nam 2021 và Giải thưởng Make in Việt Nam 2021. Những giải thưởng công nghệ danh giá là khẳng định thuyết phục nhất về chất lượng thiết bị, giải pháp và nền tảng cho Vconnex nghiên cứu và phát triển.

1.2 Thiết kế sơ đồ khối của công tắc cảm ứng thông minh.

1.2.1 Yêu cầu thiết kế công tắc cảm ứng thông minh.

Bài toán:

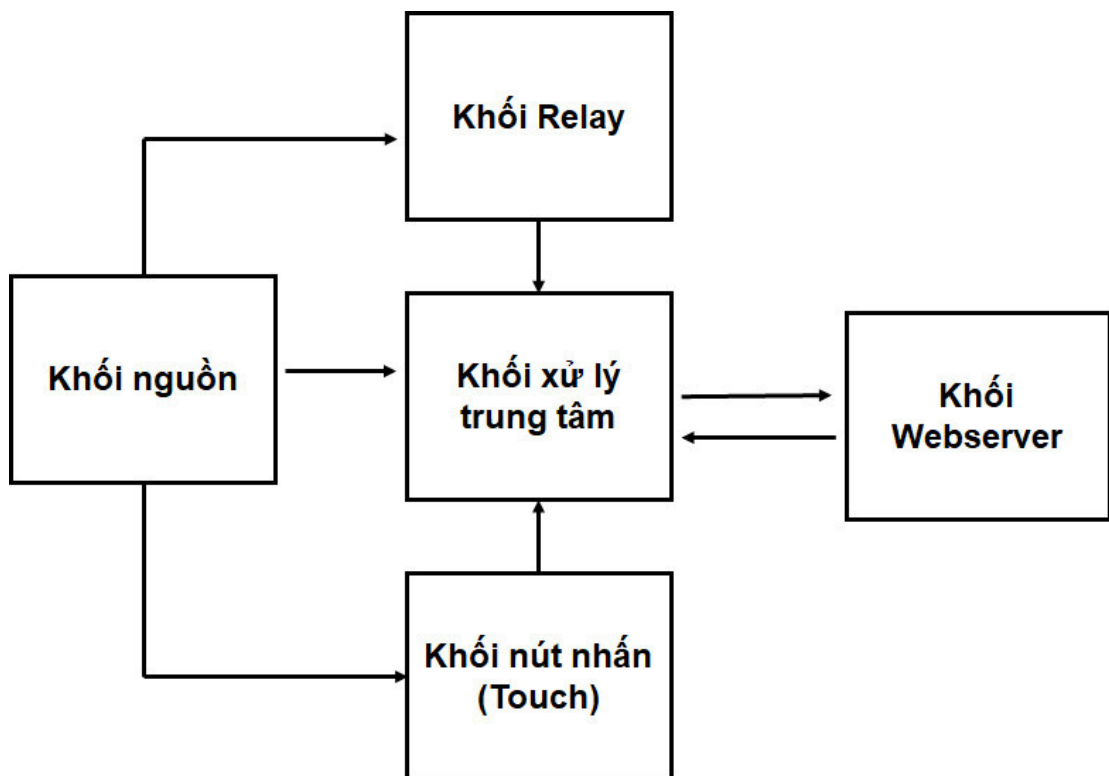
Thiết kế hệ thống công tắc cảm ứng thông minh, nhằm điều khiển thiết bị là đèn 5w. Hệ thống có 4 công tắc là cảm biến chạm tắc chạm, bật tắt các đèn thông qua việc chạm tay vào cảm biến.

Yêu cầu đặt ra:

Hệ thống có đầy đủ chức năng của một hệ thống công tắc cảm ứng thông minh đã có trên thị trường. Hệ thống có mức độ an toàn cao với người sử dụng. Mô hình lắp ráp đơn giản, nhỏ gọn, người dùng có thể sử dụng nhanh chóng.

Thiết kế một giao diện điều khiển hệ thống từ xa. Giao diện đơn giản nhưng đầy đủ thông tin giúp người nhìn dễ dàng hình dung hoặc cách sử dụng nhanh chóng.

1.2.2 Sơ đồ khối hệ thống.



Hình 1.5. Sơ đồ khối của hệ thống.

Khối nguồn: Cung cấp nguồn nuôi cho cả hệ thống

Khối xử lý trung tâm: Nhận dữ liệu từ cảm biến, xử lý dữ liệu và xuất tín hiệu điều khiển.

Khối relay: Nhận tín hiệu điều khiển từ khối xử lý và thực hiện bật tắt các thiết bị chấp hành: đèn

Khối chấp hành: Nhận tín hiệu điều khiển từ khối xử lý và thực hiện bật tắt các thiết bị chấp hành: đèn.

1.3 Tổng kết chương 1.

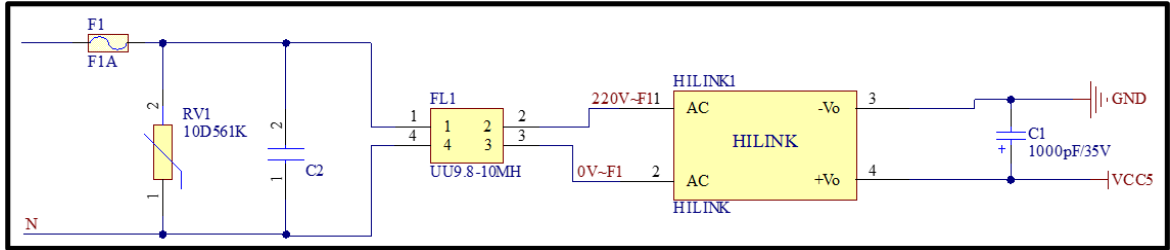
Tìm hiểu lịch sử phát triển của hệ thống công tắc cảm ứng thông minh ở trong nước và nước ngoài. Nắm rõ lợi ích và tiện ích của hệ thống đối với xã hội, đời sống của con người. Từ đó em đã lên ý tưởng thực hiện đề tài, nhằm xây dựng mô hình công tắc cảm ứng thông minh, phù hợp với sự phát triển của công nghệ 4.0. Qua tìm hiểu, em đã đặt ra yêu cầu đối với công tắc thông minh mà em sẽ làm trong đề tài này.

Từ yêu cầu đặt ra, thiết kế sơ đồ khối của hệ thống. Sơ đồ khối mô tả đầy đủ chức năng và hoạt động của hệ thống. Đưa ra chức năng cho từng khối trong sơ đồ. Từ đó sẽ đi đến bước thiết kế mạch nguyên lý cho hệ thống ở chương tiếp theo.

CHƯƠNG 2: THIẾT KẾ MÔ HÌNH

2.1 Sơ đồ nguyên lý của hệ thống công tắc cảm ứng thông minh.

2.1.1 Khối nguồn.



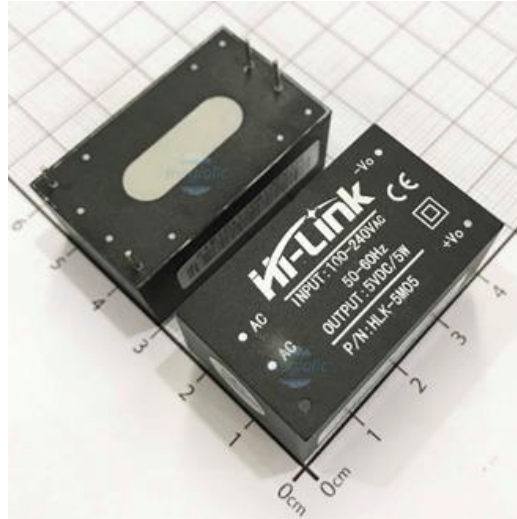
Hình 2.1. Mạch nguyên lý khối nguồn.

Vì sử dụng nguồn 220V, nên mạch nguồn sử dụng một cầu chì (F1) tự động ngắt mạch điện khi cường độ dòng điện tăng quá mức, đặc biệt là khi đoản mạch.

Sử dụng tụ chống sét (RV1) là một điện trở đặc biệt được sử dụng để bảo vệ mạch điện chống lại sự đột biến điện áp cao trong khoảng thời gian ngắn. Những xung áp cao và những xung gai sẽ tấn công đường dây điện và sẽ phá hủy nguồn cung cấp điện của các thiết bị. Khi đó, một tụ chống sét được lắp vào mạch sẽ có thể ngăn những xung áp cao và những xung gai này, tránh việc chúng phá hỏng thiết bị [1].

Bên cạnh đó sử dụng cuộn lọc nguồn (FL1) trên mạch cung cấp điện sẽ giảm lượng gợn sóng đến điểm mà điện áp một chiều đầu ra gần như một đường thẳng. Điều quan trọng trong một số mạch mà điện áp một chiều được chuyển đổi trở lại thành điện áp xoay chiều là tất cả các dấu vết của tần số ban đầu của điện áp đầu vào đều bị loại bỏ [2].

Module nguồn biến đổi từ 220V về 5V được sử dụng trong mạch là HiLink. Module cung cấp điện áp 1 chiều cho toàn bộ mạch điều khiển như KIT ESP32, Relay 5V. Module nguồn HiLink trong thực tế [3].



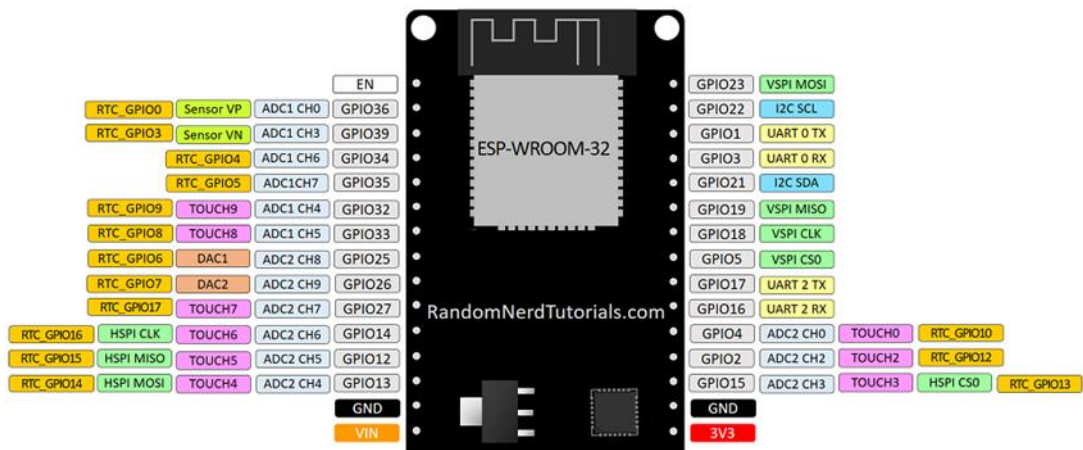
Hình 2.2. Module Hilink

Một số thông số của module như sau:

Đầu vào	Đầu ra	
Dải điện áp đầu vào 90V – 245 V	Điện áp đầu ra khi có tải	5.0 ±0.2V
	Dòng điện định mức	1A

2.1.2 Khối xử lý trung tâm.

ESP32 là một series các vi điều khiển trên một vi mạch giá rẻ, năng lượng thấp có hỗ trợ WiFi và Bluetooth chế độ kép. Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng.



Hình 2.3 Module ESP32

ESP32 được chế tạo và phát triển bởi Espressif Systems, một công ty Trung Quốc có trụ sở tại Thượng Hải, và được sản xuất bởi TSMC bằng cách sử dụng công nghệ 40 nm. ESP32 là sản phẩm kế thừa từ vi điều khiển ESP8266.

Các tính năng của ESP32 bao gồm [4]:

Bộ xử lý:

CPU: Bộ vi xử lý Xtensa lõi kép (hoặc lõi đơn) 32-bit LX6, hoạt động ở tần số 240 MHz (160 MHz cho ESP32-S0WD và ESP32-4WDH) và hoạt động ở tối đa 600 MIPS (200 MIPS với ESP32-S0WD/ESP32-U4WDH).

Bộ đồng xử lý (co-processor) công suất cực thấp (Ultra low power, viết tắt: ULP).

Hệ thống xung nhịp: CPU Clock, RTC Clock và Audio PLL Clock.

Bộ nhớ nội:

448 KB bộ nhớ ROM cho việc booting và các tính năng lõi.

520 KB bộ nhớ SRAM trên chip cho dữ liệu và tập lệnh.

Kết nối không dây:

Wi-Fi: 802.11 b/g/n.

Bluetooth: v4.2 BR/EDR và BLE (chia sẻ sóng vô tuyến với Wi-Fi).

34 GPIO và các ngoại vi:

- ADC SAR 12 bit, 18 kênh, DAC 2×8 -bit, 10 cảm biến cảm ứng (touch sensor) (GPIO cảm ứng điện dung).
- 3 SPI (SPI, HSPI và VSPI) hoạt động ở cả 2 chế độ master/slave.[8]
Module ESP32 hỗ trợ 4 ngoại vi SPI với SPI0 và SPI1 kết nối đến bộ nhớ flash của ESP32 còn SPI2 và SPI3 tương ứng với HSPI và VSPI.[9].
- 2 I2C, hoạt động được ở cả chế độ master và slave, với chế độ Standard mode (100 Kbit/s) và Fast mode (400 Kbit/s). Hỗ trợ 2 chế độ định địa chỉ là 7-bit và 10-bit. Các GPIO đều có thể được dùng để triển khai I2C.
- 3 UART (UART0, UART1, UART2) với tốc độ lên đến 5 Mbps [10].
- SD/SDIO/CE-ATA/MMC/eMMC host controller.

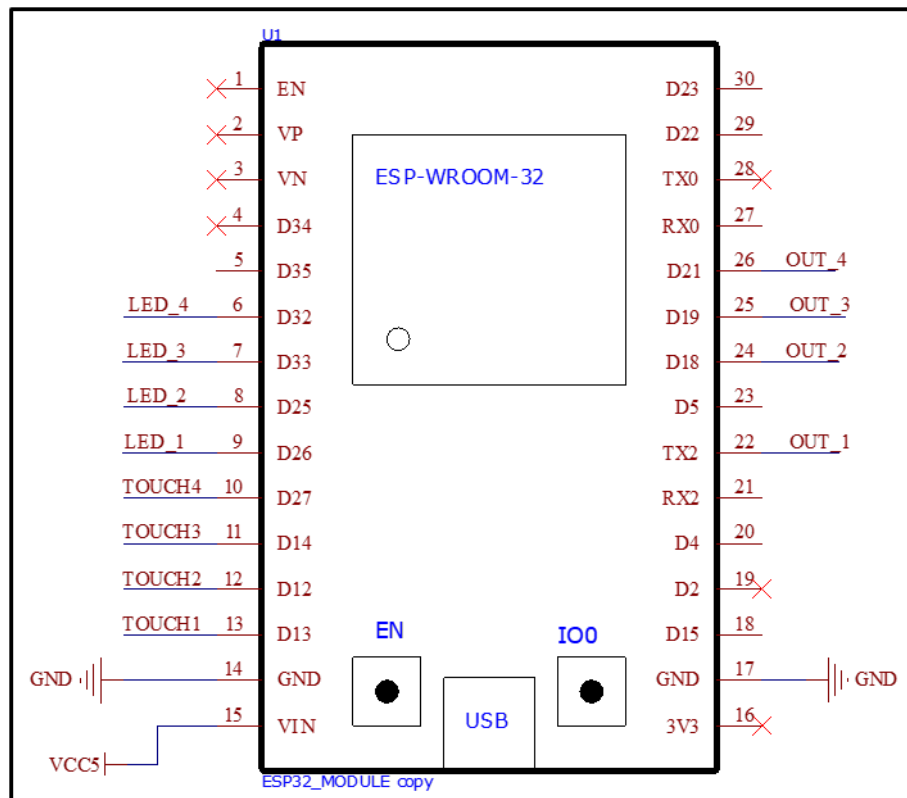
- SDIO/SPI slave controller.
- Bộ điều khiển hồng ngoại từ xa (TX/RX, lên đến 8 kênh).
- PWM cho điều khiển động cơ.LED PWM (lên đến 16 kênh).
- Cảm biến hiệu ứng Hall.
- Bộ tiền khuếch đại analog công suất cực thấp (Ultra low power analog pre-amplifier).

Bảo mật:

- Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WFA, WPA/WPA2 và WAPI.
- Secure boot (tạm dịch: khởi động an toàn).
- Mã hóa flash 1024-bit OTP, lên đến 768-bit cho khách hàng
- Tăng tốc mã hóa phần cứng: AES, SHA-2, RSA, elliptic curve cryptography (ECC, tạm dịch: mật mã đường cong ellip), trình tạo số ngẫu nhiên (random number generator, viết tắt: RNG).

Quản lý năng lượng:

- Bộ ổn áp nội với điện áp rơi thấp (internal low-dropout regulator).
- Miền nguồn riêng (individual power domain) cho RTC.
- Dòng 5 μ A cho chế độ deep sleep.
- Trở lại hoạt động từ ngắt GPIO, timer, đo ADC, ngắt với cảm ứng điện dung.



Hình 2.4. Sơ đồ nguyên lý khối xử lý trung tâm.

Sử dụng tính năng Label của Altium để kết nối với từng khối.

Chân D32 (6) kết nối với đèn số 4 (LED 4)

Chân D33 (7) kết nối với đèn số 3 (LED 3).

Chân D25 (8) kết nối với đèn số 2 (LED 2).

Chân D26 (9) kết nối với đèn số 1 (LED 1).

Chân D27 (10) kết nối với công tắc cảm ứng 4 (TOUCH 4).

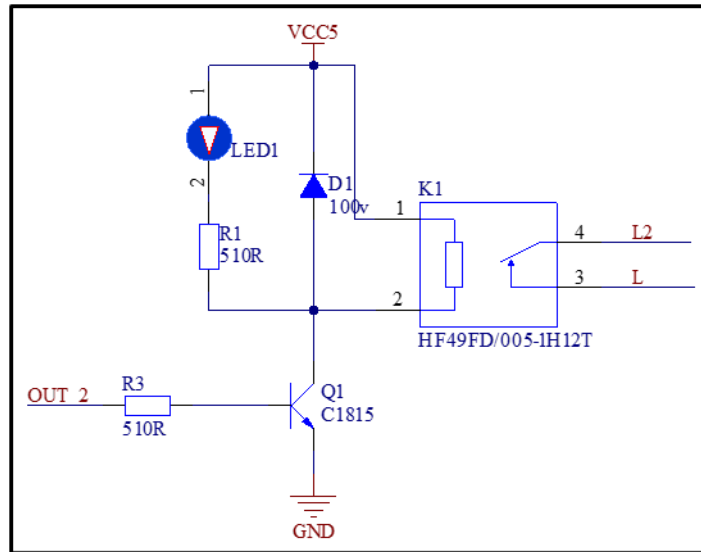
Chân D14 (11) kết nối với công tắc cảm ứng 3 (TOUCH 3).

Chân D12 (12) kết nối với công tắc cảm ứng 2 (TOUCH 2).

Chân D13 (13) kết nối với công tắc cảm ứng 1 (TOUCH 1).

Chân D21(26) , D19(25), D18(24) và TX2(22) kết nối với khối Relay.

2.1.3 Khối Relay.



Hình 2.5. Khối Relay.

Khối Relay được điều khiển bởi các chân GPIO của vi điều khiển ESP32. Mạch sử dụng 7 relay để điều khiển các phụ tải

Ngoài ra, mạch sử dụng thêm các diode (1N4148) mắc song song và mắc ngược với cuộn hút của Relay. Mục đích của việc làm này nhằm ngăn chặn sự tăng đột biến điện áp lớn phát sinh khi nguồn điện bị ngắt (gọi là điện áp ngược) bảo vệ cho Relay và mạch điều khiển.

Do tín hiệu lấy ra từ vi điều khiển và lấy ra không đủ, mạch sử dụng thêm các mạch khuếch đại với transistor C1815 mắc phân cực cố định nhằm tăng dòng điều khiển cho cuộn hút của Relay



Hình 2.6. Relay 5V 10A

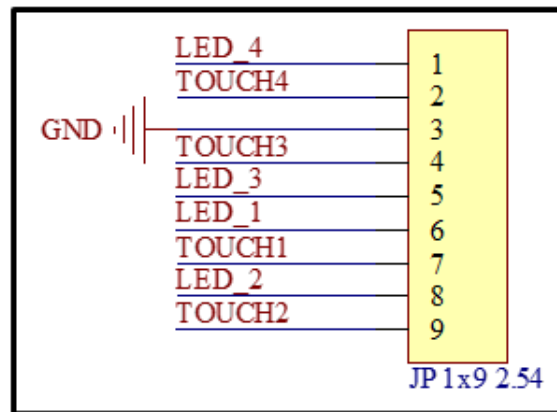
Ứng dụng:

- Nhìn chung, công dụng của relay là “dùng một năng lượng nhỏ để đóng cắt nguồn năng lượng lớn hơn”.
- Relay được dùng khá thông dụng trong các ứng dụng điều khiển động cơ và chiếu sáng.
- Khi cần đóng cắt nguồn năng lượng lớn, relay thường được ghép nối tiếp. Nghĩa là một relay nhỏ điều khiển một relay lớn hơn, và relay lớn sẽ điều khiển nguồn công suất.

Thông số kỹ thuật:*Bảng 2.1. Thông số kỹ thuật relay 5V [5]*

Dòng AC max	10 A
Dòng AC min	6A
Diameter, PCB hole	1.3 mm
Length / Height, external	22 mm
Material, contact	Silver alloy
Nhiệt độ hoạt động	- 45 °C to 75 °C
Thời gian tác động	10 ms
Thời gian nhả hãm	5 ms
Điện áp điều khiển cuộn dây	5 V

2.1.4 Khối nút nhấn.



Hình 2.7. Mạch nguyên lý khối nút nhấn

Khối nút nhấn sử dụng header 9 chân kết nối với mạch cắm kết nối với các nút nhấn. Các chân của header nối với khối xử lý trung tâm. Khối nút nhấn sử dụng cảm biến chạm để thu thập dữ liệu gửi về cho vi điều khiển.



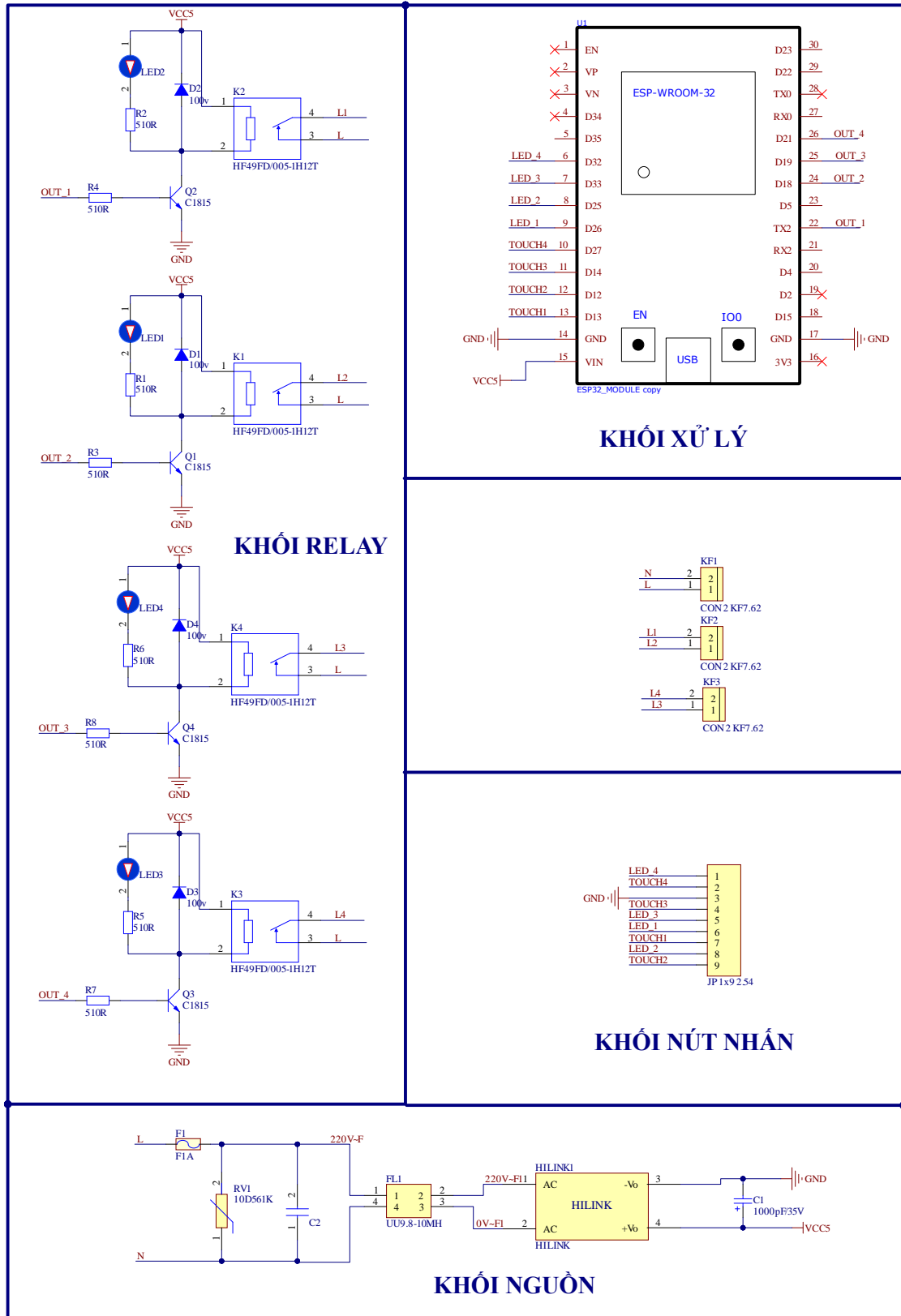
Hình 2.8. IC cảm biến chạm

Thông số kỹ thuật [6]:

Bảng 2.2. Thông số kỹ thuật IC cảm biến chạm

IC cảm biến	TP223
Dòng tiêu thụ	0.025mA
Điện áp sử dụng	2.5 - 5.5VDC
Kích thước	15 x 11 mm
Độ dày	2mm - 6mm

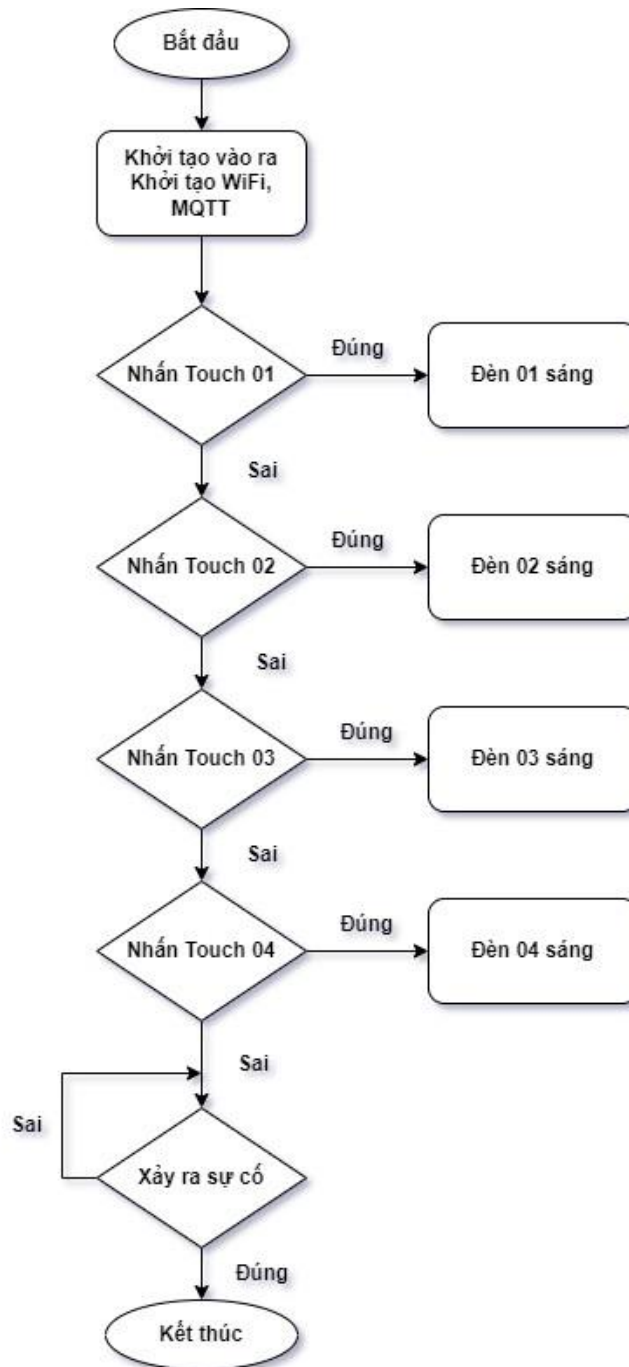
2.1.5 Sơ đồ nguyên lý toàn mạch.



Hình 2.9. Sơ đồ nguyên lý của hệ thống.

Mạch nguyên lý của hệ thống được chia làm 5 khối riêng biệt. Các khối được kết nối với khối MCU thông qua Label – tính năng của phần mềm Altium. Giúp cho mạch nguyên lý nhìn một cách rành mạch, dễ hiểu. Không gây rối cho người nhìn, dễ tìm ra lỗi và khắc phục được một cách nhanh chóng.

2.2 Lưu đồ thuật toán của hệ thống.



Hình 2.10 Lưu đồ thuật toán.

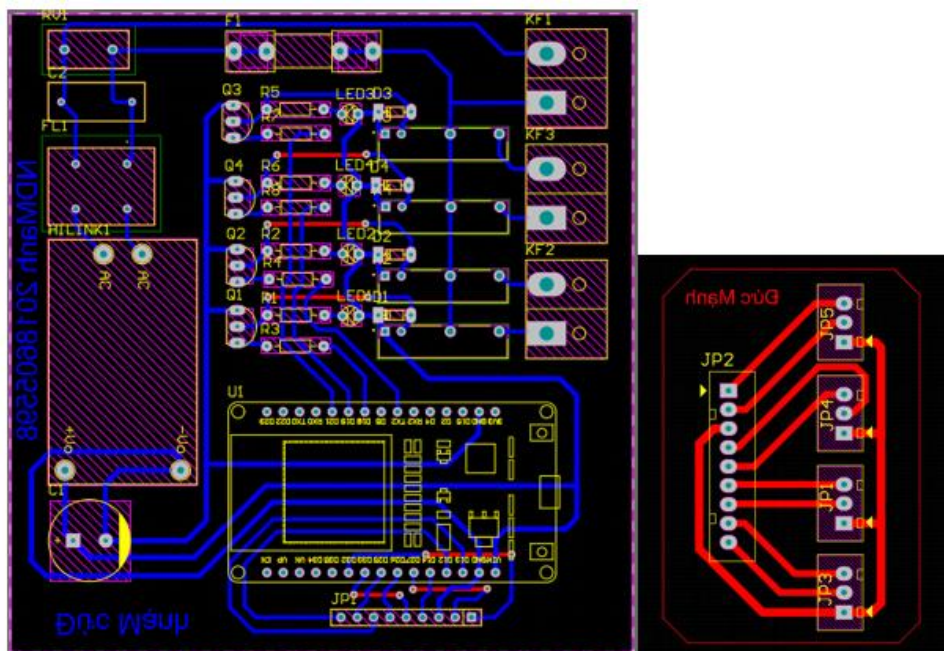
2.3 Thiết kế mạch in.

Sau khi hoàn thiện mạch nguyên lý, chuyển qua thiết kế mạch in PCB. Mạch in PCB giúp các linh kiện kết nối với nhau thông qua việc đi dây của người dùng. Phần mềm Altium trợ giúp người dùng có thể vẽ mạch in một cách nhanh chóng thông qua việc đặt luật đi dây.

Đặt luật cho mạch in:

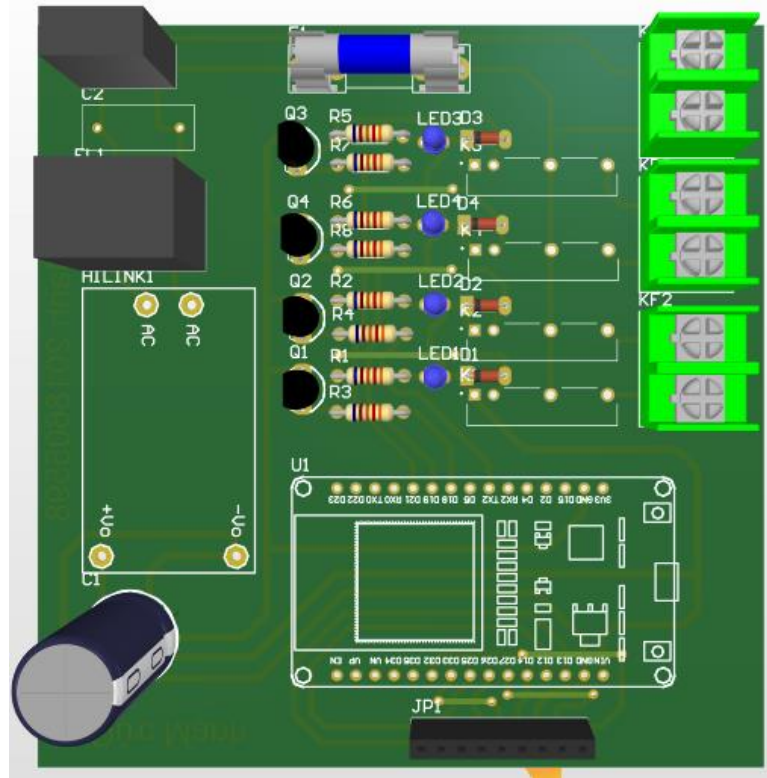
- Khoảng cách giữa các thành phần (clearance): 1mm
- Chiều rộng dây tín hiệu: 0.8mm
- Chiều rộng dây GND: 1mm
- Chiều rộng dây VCC: 1mm

Sơ đồ mạch in sau khi đi dây:



Hình 2.11. Sơ đồ mạch in PCB 2D.

Trong quá trình vẽ mạch PCB do việc sắp xếp linh kiện chưa được tối ưu dẫn đến còn nhiều đường dây vẫn bị cắt ngang không thể kết nối các chân. Do vậy, em sử dụng phương pháp đi dây 2 lớp. Lớp dây xanh là Bottom Layer, dây đỏ là lớp Top Layer.



Hình 2.12. Sơ đồ mạch in PCB 3D

Mạch in PCB 3D cũng là một tính năng của Altium, giúp cho người dùng có thể nhìn được mạch thực tế trước khi chuyển tới bước tạo mạch và hàn linh kiện. Thuận tiện cho việc sắp xếp tổng thể, xác định khoảng cách các linh kiện một cách chân thực. Khi hàn linh kiện sẽ không mắc phải các lỗi thường gặp phải.

2.4 Mô hình thực tế.



Hình 2.13 Mô hình thực tế.

Như hình 2.13, mô hình gồm 2 bộ phận chính:

Một là, hộp điều khiển nằm phía bên phải mô hình, hộp gồm mạch điều khiển ở trong và 4 cảm biến chạm phía trên mặt hộp.

Hai là, 4 bóng đèn được sử dụng nhằm mô phỏng sự hoạt động của từng cảm biến chạm.

2.5 Kết luận chương 2.

Sau khi đặt ra yêu cầu cho hệ thống, thông qua tìm hiểu và tham khảo một số tài liệu liên quan, em đã thiết kế sơ đồ khối cho toàn bộ hệ thống. Sơ đồ khối đầy đủ chức năng, phù hợp với mục đích ban đầu đặt ra. Từ việc thiết kế sơ đồ khối, tiếp theo là lựa chọn linh kiện cho từng khối sao cho phù hợp nhất, đảm bảo được đầy đủ chức năng cho từng khối. Hệ thống hoạt động nhanh chính xác.

CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

3.1 Phân tích, giải thích kết quả thực nghiệm.

Mô hình chạy ổn định với các nút nhấn cảm ứng điện dung không chạm. Kết quả thực nghiệm cho thấy mô hình có thể đáp ứng nhanh với tần số nhấn nhả thử nghiệm cao, mạch vẫn đáp ứng yêu cầu được đưa ra.

Mạch có sử dụng hệ điều hành cho phép người sử dụng có thể thao tác nhiều sự kiện nhấn nút cùng một lúc, vì vậy sản phẩm mang lại một trải nghiệm rất tốt cho người sử dụng

3.1.1 Tính năng và hiệu quả sử dụng của sản phẩm

Sản phẩm có các tính năng cơ bản của một thiết bị điều khiển không dây, bằng cảm biến chạm, ta có thể thực hiện điều khiển bằng cách cảm ứng vào các nút nhấn. Ngoài ra, chúng ta có thể dùng chính chiếc điện thoại thông minh của mình để điều khiển các thiết bị.

Hiệu quả sử dụng: dù chính là mô hình nhỏ, nhưng nếu được áp dụng vào thực tế thì hiệu quả của nó rất cao. Người dùng có thể sử dụng nó ở bất cứ đâu trong nhà, thông qua wifi và sử dụng điện thoại thông minh, giảm chi phí sửa chữa khi các nút nhấn vật lý bị hỏng.

3.1.2 Tính ứng dụng, mức độ an toàn và tác động của sản phẩm thiết kế tới môi trường, kinh tế và xã hội.

Hệ thống công tác cảm ứng được em thiết kế với mức độ an toàn cao, mạch điều khiển được để trong hộp điện chuyên dụng nhằm bảo vệ mạch đồng thời tránh tác động từ bên ngoài ảnh hưởng tới hoạt động của mạch. Dây nối giữa hệ thống với đèn được nối ngầm qua lớp bìa cứng, bảo vệ người dùng trong các trường hợp mạch điện bị hở hoặc chập mạch gây cháy nổ.

Hệ thống sử dụng những vật liệu đơn giản, thân thiện tới môi trường, dễ dàng phân hủy. Cùng với đó, việc thiết kế mô hình cũng rất đơn giản, từ những vật liệu dễ dàng mua được ở cửa hàng, giá thành rẻ, phù hợp với sinh viên muốn nghiên cứu phát triển thêm, hệ thống không quá cầu kì về ngoại hình, em thiết

kế đơn giản nhỏ gọn, có đầy đủ chức năng được đặt ra trước đó. Đồng thời hệ thống cũng có nhiều không gian để phát triển thêm.

Vì vậy, hệ thống công tắc cảm ứng thông minh có mức độ an toàn cao, thân thiện với môi trường, dễ dàng lắp ráp, phù hợp với sinh viên hoặc muốn tìm hiểu, phát triển hệ thống.

3.2 Hướng dẫn sử dụng sản phẩm thiết kế.

Bước 1: Dùng thiết bị phát Wifi chia sẻ mạng với thông tin như sau (hình :

Tên wifi: “ducmanh” ;Mật khẩu: “ducmanhdt4”.

Bước này là kết nối wifi với module ESP32 trên mô hình. Nhằm mở giao diện điều khiển trên Wed.

Bước 2: Cấp nguồn 12V qua Jack DC 5.5 cho mô hình.

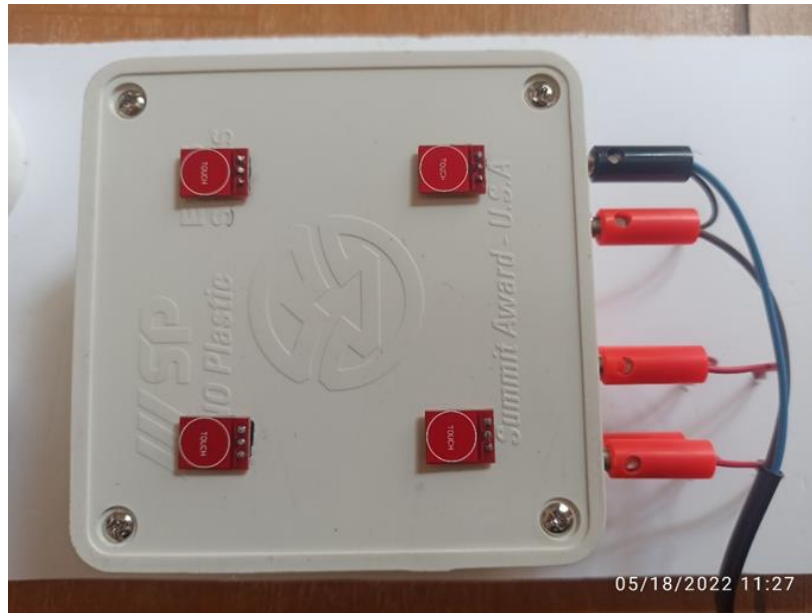
Sử dụng 2 nguồn 12V. Một nguồn là cấp cho toàn mô hình. Một nguồn cấp cho khối chấp hành.



Hình 3.1. Truy cập internet

Bước 3: Bật tắt các thiết bị bằng nút nhấn có trên bộ điều khiển.

Bộ điều khiển có 4 nút (màu đỏ) cho phép nhấn lần đầu để bật đèn tương ứng, nhấn thêm một lần nữa để tắt đèn tương ứng



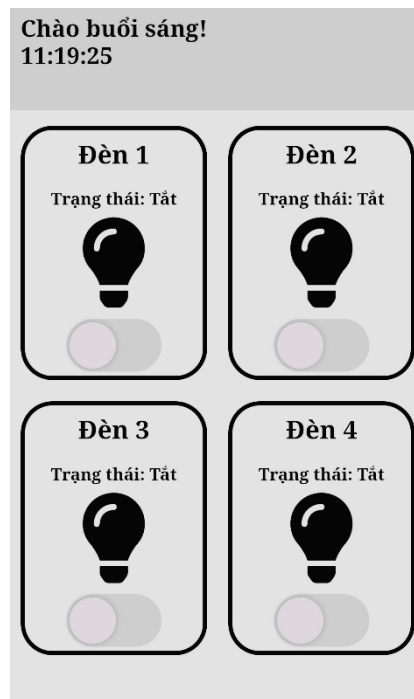
Hình 3.2. Điều khiển hệ thống bằng nút nhấn

Bước 4: Mở file index.html chứa giao diện của website điều khiển và giám sát hệ thống.



Hình 3.3. File Index.html

Bước 5: Điều khiển hệ thống bằng cách nhấn các nút nhấn có trên giao diện, đồng thời quan sát được trạng thái của từng thiết bị trên giao diện



Hình 3.4. Giao diện điều khiển hệ thống

3.3 Kết luận chương 3

Hoàn thành việc thiết kế mạch nguyên lý cho toàn hệ thống. Xuất mạch in PCB và thực hiện bước tiếp theo hàn mạch hoàn thành hộp điều khiển hệ thống. Thiết kế phần mềm điều khiển bằng cách vẽ lưu đồ thuật toán của các chế độ hoạt động.

Về phần thiết kế mô hình

Việc thiết kế mạch thực tế phải tính tới các phần cách ly để không bị nhiễu tới các khối khác, đặc biệt là khối vi điều khiển. Hệ thống được lắp đặt trong các hộ gia đình, đảm bảo an toàn cho người sử dụng, tránh được những nguy hiểm từ nguồn điện cao hay các vấn đề như hở điện hay dễ gây cháy nổ.

Về phần lập trình

Việc lập trình vi điều khiển yêu cầu người lập trình phải nắm rõ các kiến thức về ngôn ngữ lập trình C, kiến thức về các ngoại vi cơ bản của vi điều khiển, xử lý dữ liệu để đưa ra tín hiệu điều khiển... Việc lập trình với Sever yêu cầu

người lập trình phải nắm rõ các kiến thức về Python, về công cụ mình đang sử dụng. Ngoài ra việc sử dụng thành thạo các công cụ sửa chữa, các công cụ kiểm soát và lập trình rất quan trọng để sửa chữa trong quá trình xảy ra lỗi. Việc lập trình và đưa trang web lên webserver yêu cầu người lập trình phải nắm vững các kiến thức về web cơ bản, xử lý sự kiện trên webserver, thao tác với giao diện điều khiển ...

Về phần thiết kế giao diện Web

Tạo ra một giao diện Web đơn giản không quá phức tạp, đầy đủ các tính năng giúp người sử dụng có thể hiểu và nắm bắt cách sử dụng một cách nhanh chóng.

KẾT LUẬN

Kết quả đạt được

Qua quá trình thực hiện làm đồ án, em đã trình bày các cơ sở lý thuyết liên quan và chạy thành công hệ thống **“Thiết kế mô hình công tắc cảm ứng thông minh sử dụng module ESP32”**. Từ việc tìm hiểu công nghệ phát triển của hệ thống cho tới các bước thiết kế hoàn thành mô hình.

Cũng cố kiến thức, áp dụng vào việc thiết kế hệ thống. Sử dụng thành thạo các phần mềm hỗ trợ trong việc hoàn thành đề tài như: Altium... Nghiên cứu và tìm hiểu về các hệ thống điều khiển và giám sát trên thực tế, ưu điểm và nhược điểm của từng hệ thống. Nghiên cứu và tìm hiểu về vi điều khiển ESP32, các ngoại vi và lập trình ESP32. Tìm hiểu về chuẩn truyền thông không dây WiFi, ứng dụng nó vào trong việc điều khiển, truyền nhận tín hiệu và giám sát. Nắm được các hệ thống điều khiển và giám sát từ xa hoạt động. Xây dựng được giao diện điều khiển và giám sát. Thiết kế và vận hành thành công mạch giám sát và điều khiển thiết bị qua WiFi sử dụng vi điều khiển ESP32.

Ưu điểm của hệ thống

- Thiết kế đơn giản, đầy đủ chức năng của hệ thống công tắc thông minh đã có trên thị trường.
- Hệ thống thiết kế mặt kính cường lực bao phủ hoàn toàn cách điện giúp hạn chế tối đa trường hợp chập cháy, rò rỉ điện, đảm bảo an toàn tuyệt đối cho người sử dụng.
- Kiểm soát thông minh và giám sát hệ thống từ trung tâm
- Hệ thống tiết kiệm điện năng, tiêu thụ điện ít.
- Dễ dàng phát hiện thiết bị cháy hỏng, sửa chữa nhanh chóng và an toàn.
- Phần mềm có thể tương thích với PC, Laptop và các sản phẩm di động khác.

Nhược điểm của hệ thống.

- Chưa thể hiện được lợi ích của hệ thống Iot trong công nghiệp.
- Điều khiển ít thiết bị, mới sử dụng đèn thay thế, chưa thể hiện tính công nghiệp trong mô hình.
- Giao diện điều khiển còn đơn giản, chưa thể giám sát được nhiều trạng thái của thiết bị cũng như hệ thống.
- Chưa có hệ thống phân tích bảo trì dự đoán tình trạng của thiết bị công nghiệp

Hướng phát triển đề tài.

- Tạo giao diện phong phú và đa dạng hơn
- Giám sát và điều khiển các thiết bị không chỉ ON/OFF mà cả các tín hiệu nhiều dạng khác nhau
- Phát triển đa nền tảng (Android, IOS...)
- Điều khiển và giám sát thêm các thiết bị khác, các loại thiết bị có công suất lớn
- Phát triển đa nền tảng: đề tài không chỉ giới hạn ở Web mà còn có thể phát triển để phù hợp với điện thoại Android và IOS
- Phát triển các chức năng cấu hình ngay trên điện thoại
- Phát triển nhiều giao thức hơn nữa thay vì chỉ sử dụng MQTT để trong tất cả các trường hợp đều có thể bật tắt các thiết bị
- Phát triển không chỉ điều khiển các thiết bị bật hoặc tắt mà còn có thể điều khiển nhiều tín hiệu khác nữa

TÀI LIỆU THAO KHẢO

[1] Bkaii. 2021. “Tụ chống set”. Truy cập ngày 12/4. <https://bkaii.com.vn/tin-tuc/482-tu-chong-set-la-gi-tim-hieu-ve-tu-chong-set>. (Bkaii 2021).

[2] Mobitool. 2019. “Cuộn lọc nguồn”. Truy cập ngày 13/4. <https://mobitool.net/tac-dung-cua-cuon-cam-trong-mach-loc-nguon.html>. (Mobitool 2019).

[3] Linh kien thu duc. 2011. “Module nguồn Hilink”. Truy cập ngày 15/4. <https://linhkientheduc.com/module-nguon-ac-dc-hi-link-hlk-pm01-5vdc-3w>. (linhkientheduc 2011).

[4] Wikipedia. 2021. “ESP32”. Truy cập ngày 12/4. <https://vi.wikipedia.org/wiki/ESP32>. (Wikipedia 2021).

[5] Linh kiện 888. 2021. “Relay 5v 10A”. Truy cập ngày 30/4. <https://linhkien888.vn/relay-5v-10a-5-chan-srd-05vdc-sl-c>. (Linh kiện 888 2021).

[6] Robocon. 2021. “Cảm biến chạm”. Truy cập ngày 30/4. <http://robocon.vn/detail/mdl322-module-cam-bien-1-cham-ttp223-mini-do-pg2.html>. (Robocon 2021).

PHỤ LỤC

```

#include <PubSubClient.h>
#include <WiFi.h>
#include <ArduinoJson.h>
#define ssid "ducmanh"
#define password "ducmanhdt4"
#define mqtt_server "broker.hivemq.com"
#define mqtt_user ""
#define mqtt_pwd ""
const uint16_t mqtt_port = 1883;
String mqtt_topic_control_TB = "nguyenducmanh/controlTB";
String mqtt_topic_infor_TB = "nguyenducmanh/inforTB";//publish
String mqtt_topic_infor_TB_Now = "nguyenducmanh/inforTBNow";
WiFiClient espClient;
PubSubClient client(espClient);
void callback(char* topic, byte* payload, unsigned int length);
void setup_wifi();
void SendDataMQTT();
void reconnect();
void taskInit();
void sendMqttTask();
void sendMqtt(void *parameter);
void sendDataMQTT();
void otaUpdate();
String Data = "";
uint16_t dataInt = 0;
float dataFloat = 0;
int touch1pin = 25;

```

```
int touch2pin = 26;
int touch3pin = 32;
int touch4pin = 33;
int relay1pin = 17;
int relay2pin = 18;
int relay3pin = 19;
int relay4pin = 21;
int button1PressCount = 0;
int button2PressCount = 0;
int button3PressCount = 0;
int button4PressCount = 0;
void pinInit();
void handleButton1Task();
void handleButton2Task();
void handleButton3Task();
void handleButton4Task();
void handleButton1(void *parameter);
void handleButton2(void *parameter);
void handleButton3(void *parameter);
void handleButton4(void *parameter);
void sendData(void *parameter);
void handleAutoTask();
void handleAuto(void *parameter);
void setup() {
    Serial.begin(115200);
    WiFi.disconnect(true);
    delay(1000);
    Serial.println();
    Serial.println();
}
```



```

Serial.println("Wait for WiFi... ");
pinInit();
// setup_wifi() ;
client.setServer(mqtt_server, mqtt_port);
client.setCallback(callback);
taskInit();
otaUpdate();
}

void loop()
{
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
}

void callback(char* topic, byte* payload, unsigned int length)
{
    String Data = "";
    for (int i = 0; i < length; i++)
    {
        Data += (char)payload[i]; // abcde
    }
    dataInt = Data.toInt();
    if(strcmp(topic, mqtt_topic_control_TB.c_str()) == 0)
    {
        String Data = "";
        for (int i = 0; i < length; i++)
        {

```

```

    Data += (char)payload[i]; // abcde
}
dataInt = Data.toInt();
Serial.println(dataInt);
///button 1
if (dataInt == 10)
{
    button1PressCount = 0;
    digitalWrite(relay1pin, LOW);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "10");
}
else if (dataInt == 11)
{
    button1PressCount = 1;
    digitalWrite(relay1pin, HIGH);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "11");
}
    ///button 2
else if (dataInt == 20)
{
    button2PressCount = 0;
    digitalWrite(relay2pin, LOW);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "20");
}
else if (dataInt == 21)
{
    button2PressCount = 1;
    digitalWrite(relay2pin, HIGH);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "21");
}

```

```

    }

    ///button 3
else if (dataInt == 30)
{
    button3PressCount = 0;
    digitalWrite(relay3pin, LOW);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "30");
}
else if (dataInt == 31)
{
    button3PressCount = 1;
    digitalWrite(relay3pin, HIGH);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "31");
}

    ///button 4
else if (dataInt == 40)
{
    digitalWrite(relay4pin, LOW);
    button4PressCount = 0;
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "40");
}
else if (dataInt == 41)
{
    digitalWrite(relay4pin, HIGH);
    button4PressCount = 1;
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "41");
}
}
}
}

```

```

void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void reconnect()
{
    while (!client.connected())
    {
        String clientId = String(random(0xffff), HEX);
        if (client.connect(clientId.c_str(), mqtt_user, mqtt_pwd))
        {
            Serial.println("Connected MQTT");
            client.subscribe(mqtt_topic_control_TB.c_str());
        }
        else
        {
            Serial.println("Not Connected MQTT");
            delay(3000);
        }
    }
}

```

```

    }
}

void sendDataMQTT()
{
    DynamicJsonDocument doc(1024);
    char inforInverterBuff[256];
    doc["relay1"] = button1PressCount;
    doc["relay2"] = button2PressCount;
    doc["relay3"] = button3PressCount;
    doc["relay4"] = button4PressCount;
    serializeJson(doc, inforInverterBuff);
    client.publish(mqtt_topic_infor_TB.c_str(), inforInverterBuff);
}

//rtos task
void sendMqttTask()
{
    xTaskCreatePinnedToCore( // Use xTaskCreate() in vanilla FreeRTOS
        sendMqtt, // Function to be called
        "sendMqtt", // Name of task
        2048, // Stack size (bytes in ESP32, words in FreeRTOS)
        NULL, // Parameter to pass to function
        1, // Task priority (0 to configMAX_PRIORITIES - 1)
        NULL, // Task handle
        1);
}

void sendMqtt(void *parameter) {

```

```

while (1) {
  if (WiFi.status() != WL_CONNECTED)
  {
    WiFi.reconnect();
    vTaskDelay(2000 / portTICK_PERIOD_MS);
  }
  sendDataMQTT();
  vTaskDelay(2000 / portTICK_PERIOD_MS);
}

void handleButton1Task()
{
  xTaskCreatePinnedToCore( // Use xTaskCreate() in vanilla FreeRTOS
    handleButton1, // Function to be called
    "handleButton1", // Name of task
    2048, // Stack size (bytes in ESP32, words in FreeRTOS)
    NULL, // Parameter to pass to function
    1, // Task priority (0 to configMAX_PRIORITIES - 1)
    NULL, // Task handle
    1);
}

void handleButton2Task()
{
  xTaskCreatePinnedToCore( // Use xTaskCreate() in vanilla FreeRTOS
    handleButton2, // Function to be called
    "handleButton2", // Name of task
    2048, // Stack size (bytes in ESP32, words in FreeRTOS)
    NULL, // Parameter to pass to function

```

```

    1,          // Task priority (0 to configMAX_PRIORITIES - 1)
    NULL,       // Task handle
    1);
}

void handleButton3Task()
{
    xTaskCreatePinnedToCore( // Use xTaskCreate() in vanilla FreeRTOS
        handleButton3, // Function to be called
        "handleButton3", // Name of task
        2048,          // Stack size (bytes in ESP32, words in FreeRTOS)
        NULL,          // Parameter to pass to function
        1,             // Task priority (0 to configMAX_PRIORITIES - 1)
        NULL,          // Task handle
        1);
}

void handleButton4Task()
{
    xTaskCreatePinnedToCore( // Use xTaskCreate() in vanilla FreeRTOS
        handleButton4, // Function to be called
        "handleButton4", // Name of task
        2048,          // Stack size (bytes in ESP32, words in FreeRTOS)
        NULL,          // Parameter to pass to function
        1,             // Task priority (0 to configMAX_PRIORITIES - 1)
        NULL,          // Task handle
        1);
}

void handleButton1(void *parameter) {
    while (1) {
        while (digitalRead(touch1pin) == HIGH);
    }
}

```

```

while (digitalRead(touch1pin) == LOW);
button1PressCount ++;
if (button1PressCount == 2)
{
    button1PressCount = 0;
}
if (button1PressCount == 0)
{
    digitalWrite(relay1pin, LOW);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "10");
}
else if (button1PressCount == 1)
{
    digitalWrite(relay1pin, HIGH);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "11");

}
}
}

void handleButton2(void *parameter) {
    while (1) {
        while (digitalRead(touch2pin) == HIGH);
        while (digitalRead(touch2pin) == LOW);
        button2PressCount ++;
        if (button2PressCount == 2)
        {
            button2PressCount = 0;
        }
        if (button2PressCount == 0)

```



```

{
    digitalWrite(relay2pin, LOW);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "20");
}
else if (button2PressCount == 1)
{
    digitalWrite(relay2pin, HIGH);
    client.publish(mqtt_topic_infor_TB_Now.c_str(), "21");
}
}
}

void handleButton3(void *parameter) {
    while (1) {
        while (digitalRead(touch3pin) == HIGH);
        while (digitalRead(touch3pin) == LOW);
        button3PressCount ++;
        if (button3PressCount == 2)
        {
            button3PressCount = 0;
        }
        if (button3PressCount == 0)
        {
            digitalWrite(relay3pin, LOW);
            client.publish(mqtt_topic_infor_TB_Now.c_str(), "30");
        }
        else if (button3PressCount == 1)
        {
            digitalWrite(relay3pin, HIGH);
            client.publish(mqtt_topic_infor_TB_Now.c_str(), "31");
        }
    }
}

```

```

    }
}
}

void handleButton4(void *parameter) {
    while (1) {
        while (digitalRead(touch4pin) == HIGH);
        while (digitalRead(touch4pin) == LOW);
        button4PressCount ++;
        if (button4PressCount == 2)
        {
            button4PressCount = 0;
        }
        if (button4PressCount == 0)
        {
            digitalWrite(relay4pin, LOW);
            client.publish(mqtt_topic_infor_TB_Now.c_str(), "40");
        }
        else if (button4PressCount == 1)
        {
            digitalWrite(relay4pin, HIGH);
            client.publish(mqtt_topic_infor_TB_Now.c_str(), "41");
        }
    }
}

void taskInit()
{
    sendMqttTask();
    handleButton1Task();
    handleButton2Task();

```

```
    handleButton3Task();
    handleButton4Task();
}

void otaUpdate()
{
    AsyncElegantOTA.begin(&server);
    server.begin();
}

void pinInit()
{
    pinMode(touch1pin, INPUT_PULLUP);
    pinMode(touch2pin, INPUT_PULLUP);
    pinMode(touch3pin, INPUT_PULLUP);
    pinMode(touch4pin, INPUT_PULLUP);
    pinMode(relay1pin, OUTPUT);
    pinMode(relay2pin, OUTPUT);
    pinMode(relay3pin, OUTPUT);
    pinMode(relay4pin, OUTPUT);
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(14, OUTPUT);
    pinMode(27, OUTPUT);
    digitalWrite(13, HIGH);
    digitalWrite(12, HIGH);
    digitalWrite(14, HIGH);
    digitalWrite(27, HIGH);
    ///
}
```