



Do An Thiet Ke Mach Phan Van Hieu 61DDT2

Đồ án Thiết kế điện (Đại học Nha Trang)



Scan to open on Studocu

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NHA
TRANG KHOA ĐIỆN – ĐIỆN TỬ**



ĐỒ ÁN THIẾT KẾ MẠCH ĐIỆN TỬ
Đề tài: THIẾT KẾ CHẾ TẠO MẠCH GIÁM SÁT VÀ ĐIỀU KHIỂN
CHO KHO LẠNH TỪ XA ỨNG DỤNG BLYNK

Giảng viên hướng dẫn: TS. Nhữ Khải Hoàn

Sinh viên thực hiện: Phan Văn Hiếu

Mã số sinh viên: 61136482

Khánh Hòa – 01/2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NHA TRANG
KHOA ĐIỆN – ĐIỆN TỬ



ĐỒ ÁN THIẾT KẾ MẠCH ĐIỆN TỬ
Đề tài: THIẾT KẾ CHẾ TẠO MẠCH GIÁM SÁT VÀ ĐIỀU KHIỂN
CHO KHO LẠNH TỪ XA ỨNG DỤNG BLYNK

Giảng viên hướng dẫn: TS. Nhữ Khải Hoàn

Sinh viên thực hiện: Phan Văn Hiếu

Mã số sinh viên: 61136482

Khánh Hòa – 01/2023

LỜI CAM ĐOAN

Tên đề tài: Thiết kế, chế tạo mạch giám sát và điều khiển cho kho lạnh từ xa ứng dụng Blynk

Giảng viên hướng dẫn: TS. Nhữ Khải Hoàn

Sinh viên thực hiện: Phan Văn Hiếu

Mã số sinh viên: 61136482

Lớp: 61.DDT-2

Ngành: Công Nghệ Kỹ Thuật Điện – Điện Tử

Số điện thoại liên lạc: 0363670162

Email: hieu.pv.61ddt@ntu.edu.vn

Lời cam đoan:

- Em xin cam đoan đề này là công trình do chính em nghiên cứu và thực hiện dưới sự hướng dẫn và cố vấn của thầy Nhữ Khải Hoàn.
- Em không sao chép từ bất kỳ một bài viết nào đã được công bố mà không trích dẫn nguồn gốc.
- Nếu có bất kỳ một vi phạm nào, em xin chịu hoàn toàn trách nhiệm.

Khánh Hòa, ngày 3 tháng 01 năm 2023

Sinh viên thực hiện

Phan Văn Hiếu

LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc đến TS. Nhữ Khải Hoàn đã trực tiếp hướng dẫn, góp ý, chia sẻ nhiều kinh nghiệm quý báu, tận tình giúp đỡ và tạo điều kiện để em hoàn thành đề tài.

Em xin chân thành cảm ơn các thầy cô trong Khoa Điện-Điện tử đã tạo điều kiện tốt nhất để em hoàn thành đề tài.

Em cũng xin gửi lời cảm ơn đến các bạn lớp 6.DDT-2 đã chia sẻ trao đổi kiến thức cũng như những kinh nghiệm quý báu trong thời gian thực hiện đề tài.

Xin chân thành cảm ơn!

Khánh Hòa, ngày 3 tháng 01 năm 2023

Sinh viên thực hiện

Phan Văn Hiếu

TÓM TẮT ĐỒ ÁN

Mở đầu: Giới thiệu sơ lược về đồ án, tính cấp thiết đề tài, lý do chọn đề tài, mục tiêu, đối tượng, phạm vi, phương pháp nghiên cứu và ý nghĩa thực tiễn của đề tài.

Chương 1: Tổng quan

Nội dung chương này bao gồm: việc nêu tổng quan về kho lạnh và các hình thức giám sát điều khiển kho lạnh hiện có.

Chương 2: Cơ sở lý thuyết

Nội dung chương này bao gồm: đã nêu khái niệm về Internet và công nghệ WiFi đây là cơ sở quan trọng cho việc phát triển các hệ thống Iot.

Chương 3: Giới thiệu các linh kiện và module

Nội dung chương này bao gồm: giới thiệu một số linh kiện điện tử và các module quan trọng sử dụng trong chế tạo mạch.

Chương 4: Thiết kế và thi công mạch điện tử

Nội dung chương này bao gồm: các bước thiết kế mạch, sơ đồ khối và các thành phần chính và chức năng

Chương 5: Xây dựng giải thuật điều khiển, giao diện giám sát và lập trình

Nội dung chương này bao gồm: việc xây dựng giải thuật cho các phần của hệ thống và cách thức tạo giao diện giám sát điều khiển.

Kết luận và hướng phát triển: Nêu ra kết luận và các giải pháp để phát triển đồ án sau này.

DANH MỤC HÌNH ẢNH

Hình 1.1 Kho lạnh công nghiệp.....	4
Hình 1.2 Kho lạnh sinh hoạt.....	5
Hình 1.3 Các tấm Panel và cách kết nối để lắp ráp kho lạnh	7
Hình 1.4 Cấu tạo cụm máy nén	8
Hình 1.5 Dàn lạnh trong kho lạnh công nghiệp	8
Hình 1.6 Cảm biến nhiệt độ thường dùng trong kho lạnh.....	11
Hình 1.7 Sơ đồ mạch điều khiển nhiệt độ kho dùng vi điều khiển	12
Hình 1.8 Một bộ điều khiển nhiệt độ.....	13
Hình 1.9 Một bộ giám sát kho lạnh theo thời gian thực.....	14
Hình 2.1 Một hệ thống giám sát nhiệt độ, độ ẩm qua mạng Internet.....	18
Hình 3.1 Sơ đồ ra chân và ngoại vi trên ESP32 DevKit	23
Hình 3.2 Giao diện làm việc của VS Code.....	24
Hình 3.3 IC ACS712 loại 30A	26
Hình 3.4 Sơ đồ chân cảm biến dòng điện ACS712	26
Hình 3.5 Sơ đồ chân IC hạ áp LM2596.....	27
Hình 3.6 Module hạ áp sử dụng LM2596	27
Hình 3.7 Module PZEM-004T và sơ đồ đấu nối.....	29
Hình 3.8 Màn hình OLED 0.96 inch	30
Hình 3.9 Cảm biến nhiệt độ sử dụng điện trở NTC	31
Hình 3.10 Sơ đồ đấu nối cảm biến nhiệt độ NTC	31
Hình 3.11 Cảm biến DS18B20 và sơ đồ đấu nối	32
Hình 3.12 Cấu tạo Opto quang.....	33
Hình 3.13 Opto PC817	33
Hình 3.14 1 Sơ đồ chân Relay.....	34
Hình 4.1 Sơ đồ khối của mạch	36
Hình 4.2 Sơ đồ nguyên lý của mạch.....	37

Hình 4.3 Giao diện khởi động phần mềm Altium Designer.....	40
Hình 4.4 Mạch in thiết kế trên phần mềm Altium Designer	41
Hình 4.5 Hình 3D mô phỏng trên Altium Designer	41
Hình 4.6 Mạch thực tế sau khi hoàn thiện.....	42
Hình 5.1 Sơ đồ khối tổng quát của hệ thống	44
Hình 5.2 Logo biểu tượng của Blynk	45
Hình 5.3 Lưu đồ của hệ thống	46
Hình 5.4 Lưu đồ chế độ Auto	47
Hình 5.5 Lưu đồ khối sự cố và khối cảnh báo, bảo vệ thiết bị.....	48
Hình 5.6 Lưu đồ nút nhấn (phần 1)	49
Hình 5.7 Lưu đồ nút nhấn (phần 2)	50
Hình 5.8 Các datastreams giao tiếp dữ liệu	51
Hình 5.9 Thiết kế giao diện bằng các kéo thả các khối phù hợp.....	52
Hình 5.10 Giao diện Web hoàn thiện (phần 1).....	52
Hình 5.11 Giao diện Web hoàn thiện (phần 2).....	52
Hình 5.12 Giao diện Tab 1 trên app điện thoại	54
Hình 5.13 Giao diện Tab 2 trên app điện thoại	55
Hình 5.14 Kết nối mạch chạy thử các tình huống	57
Hình 5.15 Giao diện Web khi giám sát kho lạnh	58
Hình 5.16 Giao diện trên App điện thoại khi giám sát kho lạnh.....	59
Hình 5.17 Thông báo cảnh báo gửi về điện thoại.....	59

DANH MỤC BẢNG

Bảng 4. 1 Danh sách linh kiện sử dụng trong từng khối	38
---	----

MỤC LỤC

LỜI CAM ĐOAN.....	i
LỜI CẢM ƠN.....	ii
TÓM TẮT ĐỒ ÁN.....	iii
DANH MỤC HÌNH ẢNH	iv
DANH MỤC BẢNG	v
MỤC LỤC LỜI CAM ĐOAN i LỜI CẢM ƠN ii TÓM TẮT ĐỒ ÁN iii.....	vi
MỞ ĐẦU.....	1
CHƯƠNG 1: TỔNG QUAN.....	3
1.1. Tổng quan về kho lạnh.....	3
1.1.1. Giới thiệu chung về kho lạnh.....	3
1.1.1.1. Phân loại kho lạnh.....	3
1.1.2. Cấu tạo và nguyên lý làm việc cơ bản của kho lạnh.....	6
1.1.3. Yêu cầu chung của kho lạnh.....	8
1.1.4. Mục đích của việc giám sát nhiệt độ, độ ẩm trong kho lạnh	10
1.1.5. Mục đích của việc giám sát và điều khiển kho lạnh từ xa	10
1.2. Phương pháp giám sát nhiệt độ và độ ẩm của kho lạnh đơn giản	10
1.3. Các phương pháp giám sát nhiệt độ và độ ẩm của kho lạnh hiện đại.....	11
1.3.1. Điều khiển hoạt động của bộ phận làm lạnh bằng mạch điện tử	11
1.3.2. Kho lạnh khả năng tự giám sát và thay đổi thông số vận hành	12
1.3.3. Giám sát từ xa qua thông qua máy tính, mạng không dây	13
1.3.4. Có khả năng ứng dụng hệ thống giám sát thời gian thực	14
1.4. Kết luận chương 1	15
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	16
2.1. Internet	16

2.1.1.	Khái niệm về mạng Internet.....	16
2.1.2.	Lợi ích của internet.....	17
2.1.3.	Ứng của Internet vào giám sát và điều khiển kho lạnh.....	17
2.2.	Công nghệ WiFi.....	18
2.2.1.	Giới thiệu	18
2.2.2.	Cách thức và nguyên lý hoạt động của WiFi.....	18
2.2.3.	Ứng dụng của WiFi.....	19
2.2.4.	Chuẩn WiFi mới nhất hiện nay	19
CHƯƠNG 3: GIỚI THIỆU CÁC LINH KIỆN VÀ MODULE		21
3.1.	Module ESP32-WROOM.....	21
3.1.1.	Giới thiệu ESP32	21
3.1.2.	Cấu hình của ESP32	21
3.1.3.	Sơ đồ ra chân trên Kit ESP32 DevKit	23
3.1.4.	Môi trường lập trình.....	23
3.2.	Module đo dòng điện ACS712	25
3.2.1.	Thông số kỹ thuật chip ACS712.....	25
3.2.2.	Sơ đồ chân Module ACS712	26
3.3.	Module hạ áp LM2596	26
3.3.1.	Giới thiệu IC LM2596	27
3.3.2.	Thông số kỹ thuật	27
3.3.3.	Sơ đồ chân.....	27
3.4.	Module đo điện AC PZEM-004T	28
3.4.1.	Giới thiệu	28
3.4.2.	Thông số kỹ thuật	28
3.4.3.	Cách đấu nối	29

3.5.	Màn hình OLED	29
3.5.1.	Giới thiệu	29
3.5.2.	Thông số kỹ thuật	29
3.6.	Một số linh kiện khác.....	30
3.6.1.	Cảm biến nhiệt độ NTC	30
3.6.2.	Cảm biến nhiệt độ DS18B20	31
3.6.3.	Opto quang PC817.....	32
3.6.4.	Relay	33
3.7.	Kết luận chương 3.....	34
CHƯƠNG 4: THIẾT KẾ VÀ THI CÔNG MẠCH ĐIỆN TỬ		35
4.1.	Yêu cầu thiết kế và sơ đồ khối của mạch điện tử	35
4.1.1.	Yêu cầu thiết phần cứng mạch điện tử.....	35
4.1.2.	Phương án thiết kế mạch.....	35
4.1.3.	Sơ đồ khối của mạch.....	36
4.2.	Các linh kiện sử dụng trong mạch	38
4.3.	Thi công mạch điện tử	39
4.3.1.	Giới thiệu phần mềm thiết kế mạch in Altium Designer	39
4.3.2.	Thiết kế mạch in	40
4.4.	Thi công mạch thực	42
4.5.	Kết luận chương 4.....	42
CHƯƠNG 5: XÂY DỰNG GIẢI THUẬT ĐIỀU KHIỂN , GIAO DIỆN GIÁM SÁT VÀ LẬP TRÌNH		43
5.1.	Yêu cầu xây dựng hệ thống	43
5.2.	Sơ đồ khối của hệ thống	44
5.3.	Giới thiệu về Blynk.....	44

5.4.	Lưu đồ thuật toán	45
5.4.1.	Lưu đồ thuật toán của hệ thống	45
5.4.2.	Lưu đồ thuật toán chế độ điều khiển Auto.....	47
5.4.3.	Lưu đồ thuật toán khởi sự cố và khởi cảnh báo, bảo vệ thiết bị	48
5.4.4.	Lưu đồ thuật toán nút nhấn	49
5.5.	Xây dựng giao diện giám sát kho lạnh trên Web Blynk và App Blynk	50
5.5.1.	Thiết kế giao diện giám sát trên Web Blynk	50
5.5.2.	Thiết kế giao diện App điện thoại.....	54
5.6.	Vận hành thử theo các bài toán giả định của kho lạnh	55
5.6.1.	Hoạt động của mạch	55
5.6.2.	Đánh giá kết quả	59
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		61
TÀI LIỆU THAM KHẢO		62
PHỤ LỤC		63

MỞ ĐẦU

Trong suốt quá trình lịch sử phát triển của con người, những cuộc cách mạng khoa học kỹ thuật đóng một vai trò rất quan trọng, nó ảnh hưởng trực tiếp đến cuộc sống của con người và luôn luôn phát triển theo hướng ngày càng hiện đại hơn. Năm 2013, một từ khóa mới là "Công nghiệp 4.0" bắt đầu nổi lên xuất phát từ một báo cáo của chính phủ Đức đề cập đến cụm từ này nhằm nói tới chiến lược công nghệ cao, điện toán hóa ngành sản xuất mà không cần sự tham gia của con người.

Sau sự kiện trên chúng ta đã chứng kiến những sự thay đổi cực kỳ nhanh chóng của thế giới trong mọi lĩnh vực từ sản xuất đến đời sống. Mọi thứ quen thuộc dần trở nên thông minh hơn và có thể giao tiếp với con người và quản lý, giám sát điều khiển từ xa từ các dây chuyền sản xuất trong nhà máy, ô tô, đến từng thiết bị trong chính ngôi nhà của chúng ta. Và để đạt được những thành tựu trên thì những năm gần đây các công nghệ về trí tuệ nhân tạo AI, Big data, Internet kết nối vạn vật Iot đang được đẩy mạnh nghiên cứu phát triển ở nhiều nước trên thế giới.

Để không bị tụt hậu so với thế giới Việt Nam ta những năm gần đây cũng đang đẩy mạnh phát triển các lĩnh vực mũi nhọn của cuộc cách mạng 4.0 trong đó có lĩnh vực Internet kết nối vạn vật Iot. Từ việc mở thêm các ngành đào tạo có liên quan ở các trường đại học đến việc đưa ra nhiều chính sách nhằm khuyến khích đẩy mạnh phong trào nghiên cứu học tập trong sinh viên.

Nhận thấy tính thiết thực của việc nghiên cứu ứng dụng công nghệ Iot vào đời sống sản xuất và tình hình thực tế là các kho lạnh công nghiệp nói riêng và các hệ thống cần quản lý nhiệt độ nói chung việc giám sát và điều khiển từ xa là rất quan trọng nhưng ít khi được chú ý đến.

Từ những yêu cầu trên, với những kiến thức đã được học tại trường trong thời gian qua về chuyên ngành Điện – Điện Tử em đã chọn đề tài –**Thiết kế chế tạo mạch giám sát và điều khiển từ xa cho kho lạnh ứng dụng Blynk** " để làm đồ án Chế tạo mạch điện tử này .

Mục tiêu của đề tài:

- Hiểu được nguyên lí hoạt động của hệ thống điều khiển từ xa qua mạng internet.
- Nắm được phương pháp giám sát và điều khiển kho lạnh.
- Chế tạo thành công bộ điều khiển.
- củng cố lại những kiến thức đã được học tập tại trường và trao đổi, bổ sung thêm kiến thức mới.

Đối tượng nghiên cứu:

- Nghiên cứu về phương thức dữ liệu qua mạng Internet.
- Nghiên cứu về hoạt động của kho lạnh và cách điều khiển
- Cách thức lập trình cũng như nguyên lí làm việc của các linh kiện.

Phạm vi nghiên cứu:

- Nghiên cứu lý thuyết.
- Làm mạch thực.

Phương pháp nghiên cứu:

- Tìm kiếm và nghiên cứu lý thuyết liên quan.
- Vận dụng tham khảo kế thừa các đồ án liên quan.

Ý nghĩa thực tiễn của đề tài:

- Có khả năng ứng dụng cho các kho lạnh nhỏ.
- Là tiền đề để phát triển được mạch điều khiển cho các kho lạnh công nghiệp
- Tạo tiền đề cho việc nghiên cứu phát triển các sản phẩm liên quan trong tương lai.

CHƯƠNG 1: TỔNG QUAN

1.1. Tổng quan về kho lạnh

1.1.1. Giới thiệu chung về kho lạnh

Kho lạnh là một nhà kho được thiết kế cách nhiệt với môi trường bên ngoài có chức năng làm lạnh, có thể điều chỉnh nhiệt độ phù hợp với đặc tính vật lý, hóa học của lô hàng, tránh được mọi tác động tiêu cực của môi trường như nhiệt độ, độ ẩm, ánh sáng, ... gây ảnh hưởng xấu đến chất lượng hàng hóa. Hiểu đơn giản thì kho lạnh cũng giống như một chiếc tủ lạnh, nhưng có quy mô to và rộng lớn hơn, được thiết kế và lắp đặt với dàn lạnh công nghiệp với nhiệt độ thích hợp để bảo quản các loại hàng hóa như: Thực phẩm, nông sản, các sản phẩm của công nghiệp hóa chất, công nghiệp thực phẩm, công nghiệp nhẹ, y tế, ... [1]

Ngày nay việc bảo quản lạnh không chỉ giới hạn ở việc bảo quản các loại sản phẩm, hàng hóa hữu cơ như nông sản, hải sản, ... mà các loại hàng hóa nhạy cảm khác như hóa chất, thuốc men, Vắc-xin, ... cũng phải đòi hỏi bảo quản trong kho lạnh.

1.1.1. Phân loại kho lạnh

Có nhiều cách phân loại kho lạnh khác nhau.

a. Phân loại kho lạnh theo công dụng

- Kho lạnh sơ bộ: Dùng làm lạnh sơ bộ hay bảo quản tạm thời thực phẩm tại các nhà máy chế biến trước khi chuyển sang một khâu chế biến khác.
- Kho chế biến: Được dùng trong các nhà máy chế biến và bảo quản thực phẩm (nhà máy đồ hộp, nhà máy sữa, nhà máy chế biến thủy sản, ...). Các kho lạnh loại này thường có dung tích lớn, cần phải trang bị hệ thống có công suất lạnh lớn. Phụ tải của kho lạnh thay đổi do phải xuất nhập thường xuyên.
- Kho phân phối, trung chuyển: Dùng điều hòa cung cấp thực phẩm cho các khu dân cư, thành phố và dự trữ lâu dài. Kho phân phối thường có dung tích lớn, trữ nhiều mặt hàng và có ý nghĩa rất lớn đối với đời sống sinh hoạt.

- Kho thương nghiệp: Kho lạnh bảo quản các mặt hàng thực phẩm của hệ thống thương nghiệp. Kho dùng bảo quản tạm thời các mặt hàng đang được doanh nghiệp bán trên thị trường.
- Kho vận tải (trên tàu thủy, tàu hỏa, ô tô): Đặc điểm của kho là dung tích lớn, hàng bảo quản mang tính tạm thời để vận chuyển.
- Kho sinh hoạt: Đây là loại kho rất nhỏ dùng trong hộ gia đình, khách sạn, nhà hàng dùng bảo quản một lượng hàng nhỏ.



Hình 1.1 Kho lạnh công nghiệp



Hình 1.2 Kho lạnh sinh hoạt

b. Phân loại kho lạnh theo nhiệt độ

- Kho bảo quản lạnh: Nhiệt độ bảo quản nằm trong khoảng -2°C đến 5°C . Các mặt hàng chủ yếu là rau củ quả và các mặt hàng nông sản.
- Kho bảo quản đông: Kho được dùng để bảo quản các mặt hàng đã qua cấp đông. Thường là các mặt hàng có nguồn gốc động vật. Nhiệt độ phù hợp tối thiểu đạt -18°C để các vi sinh vật không thể phát triển làm hư hỏng thực phẩm.
- Kho đa năng: Nhiệt độ bảo quản là -12°C , buồng bảo quản đa năng thường được thiết kế ở -12°C nhưng khi cần bảo quản lạnh có thể đưa lên nhiệt độ bảo quản 0°C hoặc khi cần bảo quản đông có thể đưa xuống nhiệt độ bảo quản -18°C tùy theo yêu cầu công nghệ.

c. Phân loại kho lạnh theo dung tích chứa.

Kích thước kho lạnh bảo quản phụ thuộc chủ yếu vào dung tích chứa hàng của nó. Do đặc điểm về khả năng chất tải mỗi loại thực phẩm khác nhau nên thường quy dung tích ra tấn thịt (MT - Meat Tons). Ví dụ kho 10MT là kho có khả năng chứa 10 tấn.

d. Phân loại theo phương pháp cách nhiệt

- Kho xây: Là kho mà kết cấu là kiến trúc xây dựng và bên trong người ta bọc cách nhiệt. Kho xây chiếm diện tích lớn, giá thành cao, không đẹp, khó tháo dỡ, di chuyển. Vì vậy ở Việt Nam ít được sử dụng.
- Kho panel: Được lắp ghép từ các tấm panel tiền chế polyuretan và được lắp ghép với nhau bằng các móc khóa cam locking và mộng âm dương. Kho panel có hình thức đẹp, gọn và giá thành tương đối rẻ, rất tiện lợi khi lắp đặt, tháo dỡ nên được dùng hầu hết ở các kho lạnh.

1.1.2. Cấu tạo và nguyên lý làm việc cơ bản của kho lạnh

Các kho lạnh ngày nay thường được cấu tạo gồm các phần như sau:

❖ Vỏ kho

Vỏ kho là được xem như lớp cách nhiệt giữa kho với môi trường bên ngoài. Vỏ kho phải được làm từ những vật liệu gọn nhẹ cách nhiệt tốt.

Vỏ kho thường được tạo nên từ hai loại chất liệu chính là panel là EPS và PU. Chất liệu Panel PU thường được dùng cho kho đông còn panel EPS lại phù hợp với các loại kho mát.

Cả hai loại vật liệu này đều mang đến nhiều ưu điểm như:

- Độ bền cao, cách nhiệt tốt do phần lõi có khả năng ngăn hơi lạnh thoát ra bên ngoài và cách nhiệt cực tốt.
- Cách nhiệt, cách âm và chống nóng lạnh.
- Dễ dàng vệ sinh và tính thẩm mỹ cao.
- Tiết kiệm được thời gian thi công do có khả năng di dời hoặc lắp đặt linh hoạt, nhanh chóng.

Vỏ kho cũng được sử dụng phương pháp lắp ghép như sau:

Ghép bằng mộng âm dương hoặc ghép bằng khoá camlocking. Tuy nhiên, phương pháp lắp ghép bằng khoá camlocking được ưa chuộng hơn do sự nhanh chóng và tiện lợi hơn.



Hình 1.3 Các tấm Panel và cách kết nối để lắp ráp kho lạnh

❖ Cửa kho

Cửa kho thường sử dụng sản phẩm đến từ thương hiệu Gatter – tiêu chuẩn Đức với hai loại cửa trượt và cửa mở phù hợp với mọi yêu cầu của khách hàng.

❖ Hệ thống cụm máy nén

Máy nén là bộ phận đóng vai trò quan trọng trong cấu tạo làm lạnh của kho lạnh. Với chức năng làm nén môi chất lạnh đến mức cao để đủ chất làm lạnh và có thể ngưng tụ.

• Cấu tạo cụm máy nén:

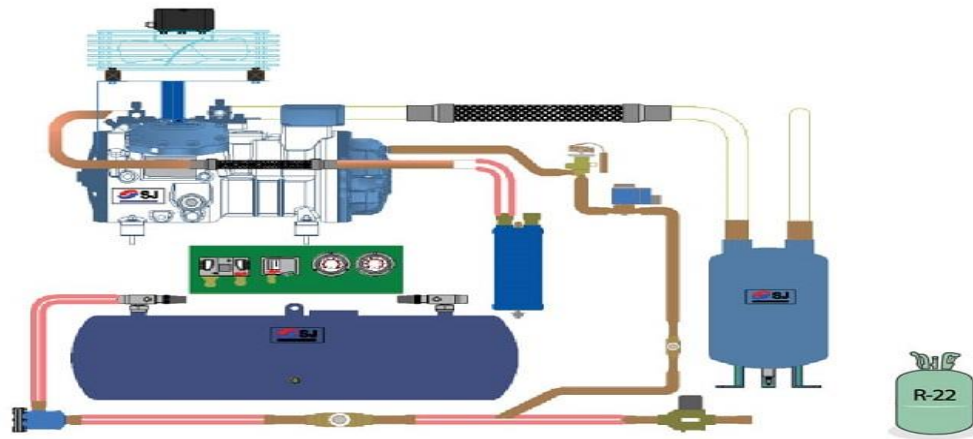
Cụm máy nén hiện nay thường có cấu tạo gồm có giải nhiệt gió, giải nhiệt nước và cụm máy nén dàn ngưng. Trong đó:

• Giải nhiệt gió:

- Loại dàn kín hoặc loại dàn hở, tất cả các cụm đều được thiết kế phù hợp với nhiệt độ môi trường cũng như mùa hè tại Việt Nam.
- Công nghệ tích nhiệt có trong hệ thống giúp tiết kiệm nhiên liệu, điện khi vận hành.

• Giải nhiệt nước:

Gồm hai loại giải nhiệt nước mặn và giải nhiệt nước ngọt.



Hình 1.4 Cấu tạo cụm máy nén

❖ Hệ thống dàn lạnh

Dàn lạnh kho lạnh (dàn bay hơi) có nhiệm vụ hoá hơi gas bão hoà ẩm sau tiết lưu đồng thời làm lạnh môi trường cần làm lạnh. [2]

Kiểu dàn lạnh: có 2 loại kết cấu dàn lạnh là tiết lưu kiểu khô và kiểu ngập lỏng.



Hình 1.5 Dàn lạnh trong kho lạnh công nghiệp

1.1.3. Yêu cầu chung của kho lạnh

Mặc dù có nhiều cách phân loại kho lạnh khác nhau, nhưng kho lạnh có những yêu cầu chung giống nhau. Một số yêu cầu chính của kho lạnh như sau:

❖ Yêu cầu về kiến trúc:

- Kho lạnh được xây dựng ở nơi cao ráo không bị ngập hoặc đọng nước, thuận tiện về giao thông, xa các nguồn gây ô nhiễm có đủ nguồn cung cấp điện ổn định.
- Có mặt bằng rộng cả trong lẫn ngoài, được bố trí thuận tiện cho việc tiếp nhận,

bốc dỡ, vận chuyển sản phẩm, tránh được khả năng gây nhiễm chéo cho sản phẩm

- Có tường bao hoặc vách ngăn cách với bên ngoài.
- Thiết kế kho lạnh phải có kết cấu vững chắc, có mái che không dột, cách nhiệt tốt.
- Trần và tường kho lạnh, phòng đệm và phòng thay bao bì, đồng gói lại (nếu có) được làm bằng vật liệu bền, không dột, không gỉ, không ngấm nước, cách nhiệt tốt, màu sáng, dễ vệ sinh.
- Cửa kho lạnh được làm bằng vật liệu bền, không dột, không gỉ, không ngấm nước, cách nhiệt tốt. Khi đóng cửa phải bảo đảm kín.
- Kho lạnh được thiết kế sao cho khi xả băng, nước từ dàn lạnh, trên trần kho, nền kho được chảy hết ra ngoài.

❖ **Yêu cầu thiết bị, phương tiện bảo quản, vận chuyển:**

- Thiết bị làm lạnh phải có công suất đủ để đảm bảo sản phẩm ở nhiệt độ cần thiết và ổn định. Môi chất sử dụng là loại môi chất được phép sử dụng, không ảnh hưởng đến môi trường.
- Các thiết bị áp lực chứa môi chất lạnh, ống dẫn, thiết bị trao đổi nhiệt phải đảm bảo an toàn, không bị rò rỉ và phải kiểm tra định kỳ.
- Bộ báo nhiệt độ và điều khiển kho lạnh được đặt ở nơi dễ nhìn, dễ đọc, dễ thao tác.

1.1.4. Mục đích của việc giám sát nhiệt độ, độ ẩm trong kho lạnh

Bất kì kho lạnh nào cũng cần có những yêu cầu về nhiệt độ. Có những kho còn yêu cầu cả ngưỡng nhiệt trên và ngưỡng nhiệt dưới.

Giám sát nhiệt độ rất quan trọng cho phòng lạnh. Điều này đặc biệt quan trọng đối với việc lưu trữ các loại thuốc, vắc-xin, máu, phòng thí nghiệm, cũng như các loại thực phẩm như trái cây, rau củ, bơ, sữa, vv...

Việc giám sát nhiệt độ kho lạnh giúp cho người vận hành nắm được nhiệt độ hiện thời của kho lạnh, từ đó có những nhưng án điều khiển cho phù hợp.

Ngoài nhiệt độ, độ ẩm cũng một thông số cần phải chú ý khi vận hành kho lạnh. Các điều kiện tối ưu khác nhau tùy thuộc vào từng hàng hóa như các loại trái cây, rau củ, hoa tươi hay các sản phẩm thực phẩm khác đang được lưu trữ nhưng thông thường, cần có độ ẩm khoảng 95%RH ở nhiệt độ chỉ cao hơn nhiệt độ đóng băng, nhưng ngay cả khi sản phẩm được lưu trữ trong điều kiện ẩm hơn thì cũng có lợi với độ ẩm khoảng 75%RH. Độ ẩm yêu cầu được giải phóng trực tiếp vào không khí, đồng đều trong kho.

Mức nhiệt độ và độ ẩm phù hợp đảm bảo chất lượng sản phẩm được duy trì.

1.1.5. Mục đích của việc giám sát và điều khiển kho lạnh từ xa

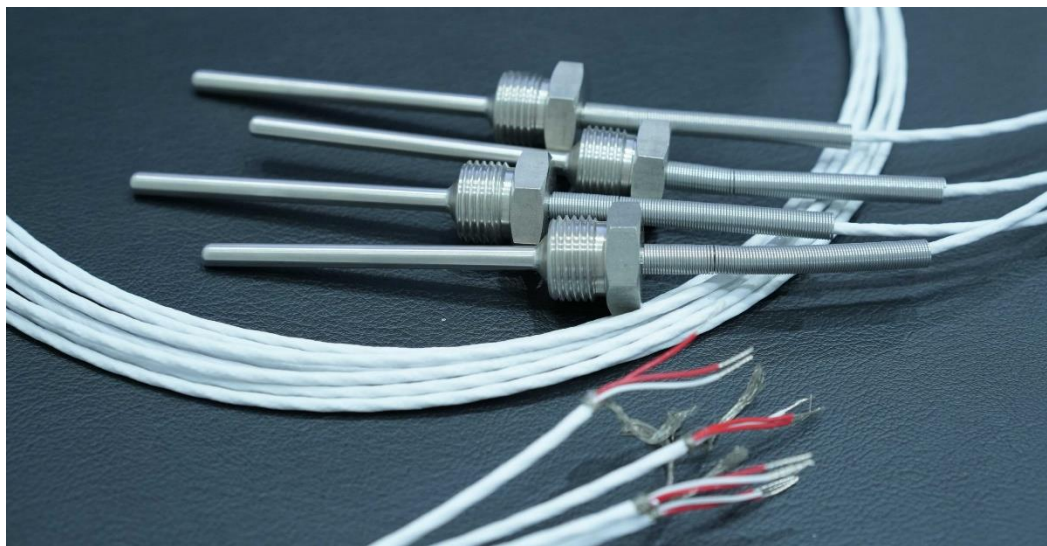
Ngày nay với việc công nghệ phát triển nhanh chóng, Internet và điện thoại di động được phổ cập thì việc áp dụng các phương pháp giám sát và điều khiển kho lạnh từ xa sẽ giúp kho lạnh vận hành an toàn hơn đảm bảo các yêu cầu kỹ thuật cần có, nhanh chóng phát hiện và chuẩn đón các sự cố không mong muốn và kịp thời đưa ra các lệnh điều khiển phù hợp với tình hình của kho lạnh. [1]

Áp dụng việc giám sát và điều khiển vận hành kho lạnh từ xa giúp giảm được công sức vận hành nhưng lại tăng tính an toàn của kho lạnh do có thể tự động vận hành theo các thông số cài đặt và nhiều tính năng bảo vệ khác.

1.2. Phương pháp giám sát nhiệt độ và độ ẩm của kho lạnh đơn giản

Về cơ bản, muốn giám sát nhiệt độ và độ ẩm thì cần có cảm biến nhiệt độ và độ ẩm. Sự phát triển của các phương pháp giám sát ở đây gắn liền với việc phát triển công nghệ chế tạo cảm biến cũng như sự phát triển của các bộ xử lý tín hiệu cảm biến.

Những kho lạnh đơn giản chỉ sử dụng cảm biến nhiệt độ và độ ẩm dạng điện trở hoặc tính chất giãn nở của thủy ngân sau đó hiển thị lên thang chia vạch hoặc đèn báo. Một số loại hiện đại hơn có tích hợp vi xử lý thì phần hiển thị có tích hợp thêm phần hiển thị là LCD hoặc LED bảy thanh. Tuy nhiên điểm chung là đơn giản chỉ có tính chất giám sát cục bộ chứ không có khả năng giám sát và điều khiển từ xa. [1]



Hình 1.6 Cảm biến nhiệt độ thường dùng trong kho lạnh

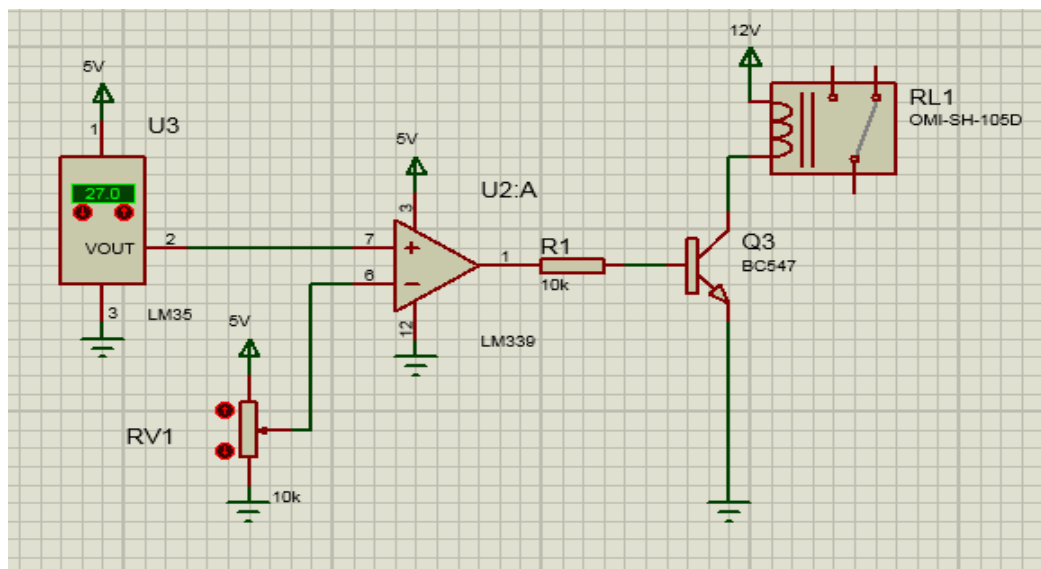
1.3. Các phương pháp giám sát nhiệt độ và độ ẩm của kho lạnh hiện đại

Như đã trình bày ở phần trước các hệ thống giám sát kho lạnh đơn giản chỉ có thể đo và hiển thị thông số tại kho lạnh đồng thời cảnh báo bằng đèn hoặc còi (nếu có). Điều này rất bất tiện cho việc muốn giám sát và điều khiển cùng lúc nhiều điểm đo, nhiều kho lạnh cùng lúc. Để khắc phục những điểm đó ngày nay những kho lạnh được giám sát theo nhiều hình thức, phương tiện tiên tiến hơn. [1]

1.3.1. Điều khiển hoạt động của bộ phận làm lạnh bằng mạch điện tử

Đây là phương pháp cải tiến đầu tiên trong hệ thống giám sát kho lạnh. Trước đây các hệ thống giám sát kho lạnh chỉ dừng lại ở việc “giám sát”, còn phần điều khiển được tách rời hoặc là chỉ có thể điều khiển thủ công. Ví dụ khi người vận hành quan sát nhiệt độ hiển thị của kho lạnh vượt hoặc dưới ngưỡng thì sẽ bật hoặc tắt bộ phận làm lạnh của kho lạnh.

Với sự ra đời của những cảm biến nhiệt độ, của linh kiện bán dẫn đã kết nối phần cảm biến và cơ cấu chấp hành lại với nhau. Sự thay đổi giá trị của cảm biến nhiệt độ sẽ tác động trực tiếp đến phần công suất và điều khiển hoạt động của phần làm lạnh. Khi vi điều khiển chưa được phổ biến ứng dụng trong điều khiển kho lạnh thì những hệ thống như vậy rất phổ biến. Dưới đây là sơ đồ khối hệ thống như vậy.



Hình 1.7 Sơ đồ mạch điều khiển nhiệt độ kho dùng vi điều khiển

1.3.2. Kho lạnh khả năng tự giám sát và thay đổi thông số vận hành

Sau quá trình vận hành, nghiên cứu và thử nghiệm với những ứng dụng khác nhau, người ta chọn ra được những thông số, giới hạn nhiệt độ độ ẩm cho kho lạnh. Ví dụ như việc lưu trữ thuốc, nhiệt độ được lưu trữ trong phạm vi 2°C đến 8°C , trong lưu trữ vắc-xin độ ẩm thích hợp là 65%. Và để giữ được các thông số thích hợp như vậy phương án tối ưu nhất chính là dùng vi điều khiển, khi đó nhiệt độ đo được và nhiệt độ cài đặt có thể xử lý để hiển thị. Ngoài ra có thể cài đặt nhiều chế độ hoạt động khác nhau và có thể dễ dàng thay đổi thông qua nút nhấn và chương trình điều khiển. [1]

Trên thực tế đây là những hệ thống giám sát và khống chế nhiệt độ, độ ẩm đang được sử dụng phổ biến nhất, do tính thuận tiện và giá thành hợp lý, những hệ thống này phù hợp với những mô hình kho lạnh đơn lạnh đơn lẻ không cần yêu cầu độ chính xác và thời gian đáp ứng nhanh.



Hình 1.8 Một bộ điều khiển nhiệt độ

1.3.3. Giám sát từ xa qua thông qua máy tính, mạng không dây

Ở phần trên vai trò của vi điều khiển đã được thể hiện rất rõ, tuy nhiên những thông số của kho lạnh chỉ được điều chỉnh và khống chế cục bộ tại chỗ. Với những hệ thống cần giám sát từ xa thì cần có thêm sự hỗ trợ của máy tính, và mạng Internet. Sau đây là một số giao thức truyền thông hay được áp dụng để thu thập số liệu, giám sát và điều khiển kho lạnh.

❖ Giao thức ZigBee

ZigBee cũng giống như Bluetooth là một loại truyền thông trong khoảng cách ngắn hiện được sử dụng với số lượng lớn trong công nghiệp. ZigBee là công nghệ mới với tầm hoạt động trực tiếp lên đến 1Km trong không khí và ít có chướng ngại vật. ZigBee hoạt động ở tần số cho phép ở hầu hết các quốc gia. Ngoài ra các Module thu phát ZigBee có thể hoạt động với yêu cầu năng lượng rất nhỏ

❖ Lora

LoRa là từ viết tắt của từ Long Range Radio nó được nghiên cứu và phát triển bởi Cycleo và sau này được mua lại bởi công ty SemTech năm 2012. Với công nghệ này, chúng ta có thể truyền dữ liệu với khoảng cách lên đến hàng Km mà không cần các mạch khuếch đại công suất, từ đó giúp tiết kiệm năng lượng khi truyền/nhận dữ liệu. Do đó LoRa có thể áp dụng rộng rãi trong các ứng dụng thu thập dữ liệu từ xa. Các dữ liệu từ các Sensor node sẽ được truyền về trung tâm dữ liệu để xử lý.

LoRa sử dụng kỹ thuật điều chế gọi là Chirp Spread Spectrum. Có thể hiểu nôm na là dữ liệu sẽ được băm bằng các xung cao tần để tạo ra tín hiệu có dải tần số cao

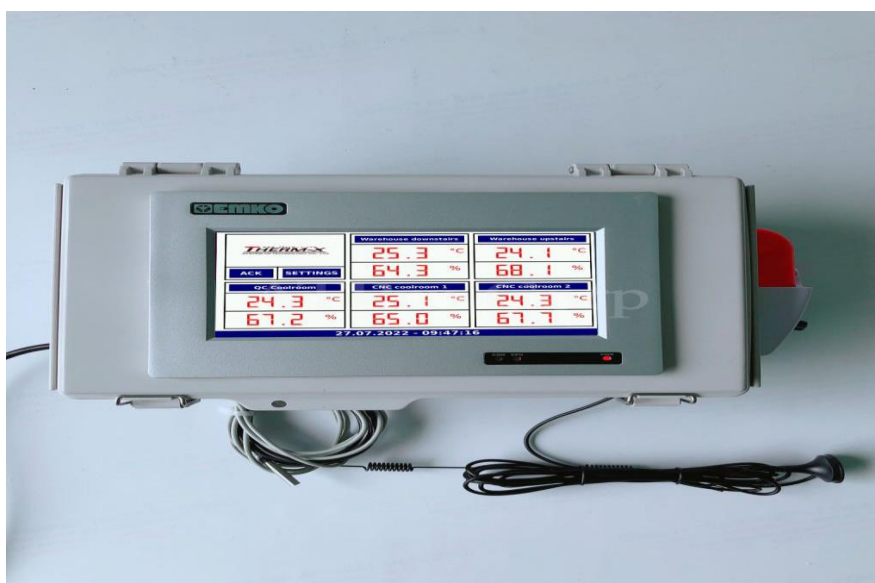
hơn (việc này gọi là Chipped); sau đó các tín hiệu cao tần này tiếp tục được mã hóa theo các chuỗi Chirp signal (là các tín hiệu hình Sin có tần số thay đổi theo thời gian; có hai loại Chirp signal là up-chirp có tần số tăng theo thời gian và down-chirp có tần số giảm theo thời gian; và việc mã hóa bit 1 sẽ sử dụng up-chirp và mã hóa bit 0 sẽ sử dụng down-chirp trước khi được truyền đi.

Ngoài những chuẩn kể trên thì còn có các chuẩn truyền nhận không dây khác cũng được sử dụng như Z-Wave, 6LoWPAN, ...

1.3.4. Có khả năng ứng dụng hệ thống giám sát thời gian thực

Với những hệ thống kho lạnh yêu cầu chặt chẽ, đáp ứng nhanh thì sự lựa chọn tối ưu là giám sát sử dụng các phương thức truyền nhận theo thời gian thực. Một số hệ thống kho lạnh cần giám sát theo thời gian thực:

- Giám sát nhiệt độ, độ ẩm cho tủ lạnh y tế bảo quản thuốc, vắc-xin.
- Giám sát nhiệt độ, độ ẩm cho tủ đông công nghiệp.
- Giám sát nhiệt độ, độ ẩm cho xe cộ vận chuyển hàng hóa nhạy cảm cần bảo quản lạnh.
- Giám sát nhiệt độ, độ ẩm cho ngân hàng máu.



Hình 1.9 Một bộ giám sát kho lạnh theo thời gian thực

1.4. Kết luận chương 1

Trong chương này đề án đã đưa ra một số khái niệm chung về kho lạnh. Qua đó giúp người đọc hiểu được cấu trúc, nguyên lý hoạt động của kho lạnh cũng như phân loại và ứng dụng của chúng trong thực tế đời sống.

Đồng thời cũng nên ra một số phương pháp giám sát và điều khiển kho lạnh. Trong đó có những phương pháp đơn giản và các phương pháp hiện đại hơn xuất hiện gần đây. Từ những kiến thức trên thì các phần sau sẽ đi từng phần nhằm xây dựng một bộ giám sát và điều khiển kho lạnh theo từ xa theo thời gian thực.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Internet

2.1.1. Khái niệm về mạng Internet

Internet là một hệ thống thông tin toàn cầu có thể được truy nhập công cộng gồm các mạng máy tính được liên kết với nhau. Hệ thống này truyền thông tin theo kiểu nối chuyển gói dữ liệu (packet switching) dựa trên một giao thức liên mạng đã được chuẩn hóa (giao thức IP). [3]

Hệ thống này bao gồm hàng ngàn máy tính nhỏ hơn của các doanh nghiệp, của các viện nghiên cứu và các trường đại học, người dùng cá nhân và các chính phủ trên toàn cầu.

Các giao thức truyền thông trong mạng internet

Có nhiều giao thức được sử dụng để giao tiếp hoặc truyền đạt thông tin trên Internet, dưới đây là một số các giao thức tiêu biểu:

- *Control Protocol*): thiết lập kết nối giữa các máy tính để truyền dữ liệu. Nó chia nhỏ dữ liệu ra thành những gói (*packet*) và đảm bảo việc truyền dữ liệu thành công.
- *IP (Internet Protocol)*: định tuyến (*route*) các gói dữ liệu khi chúng được truyền qua Internet, đảm bảo dữ liệu sẽ đến đúng nơi cần nhận.
- *HTTP (HyperText Transfer Protocol)*: cho phép trao đổi thông tin (chủ yếu ở dạng siêu văn bản) qua Internet.
- *FTP (File Transfer Protocol)*: cho phép trao đổi tập tin qua Internet.
- *SMTP (Simple Mail Transfer Protocol)*: cho phép gửi các thông điệp thư điện tử (*e-mail*) qua Internet.
- *POP3 (Post Office Protocol, phiên bản 3)*: cho phép nhận các thông điệp thư điện tử qua Internet.

- MIME (*Multipurpose Internet Mail Extension*): một mở rộng của giao thức SMTP, cho phép gửi kèm các tập tin nhị phân, phim, nhạc,... theo thư điện tử.
- WAP (*Wireless Application Protocol*): cho phép trao đổi thông tin giữa các thiết bị không dây, như điện thoại di động.

2.1.2. Lợi ích của internet

Mạng Internet mang lại rất nhiều tiện ích hữu dụng cho người sử dụng, một trong các tiện ích phổ thông của Internet là hệ thống thư điện tử (email), trò chuyện trực tuyến (chat), công cụ tìm kiếm (search engine), các dịch vụ thương mại và chuyển ngân và các dịch vụ về y tế giáo dục như là chữa bệnh từ xa hoặc tổ chức các lớp học ảo. Chúng cung cấp một khối lượng thông tin và dịch vụ khổng lồ trên Internet.

Nguồn thông tin khổng lồ kèm theo các dịch vụ tương ứng chính là hệ thống các trang Web liên kết với nhau và các tài liệu khác trong WWW (World Wide Web). Trái với một số cách sử dụng thường ngày, Internet và WWW không đồng nghĩa. Internet là một tập hợp các mạng máy tính kết nối với nhau bằng dây đồng, cáp quang, v.v.; còn WWW, hay Web, là một tập hợp các tài liệu liên kết với nhau bằng các siêu liên kết (*hyperlink*) và các địa chỉ URL và nó có thể được truy nhập bằng cách sử dụng Internet.

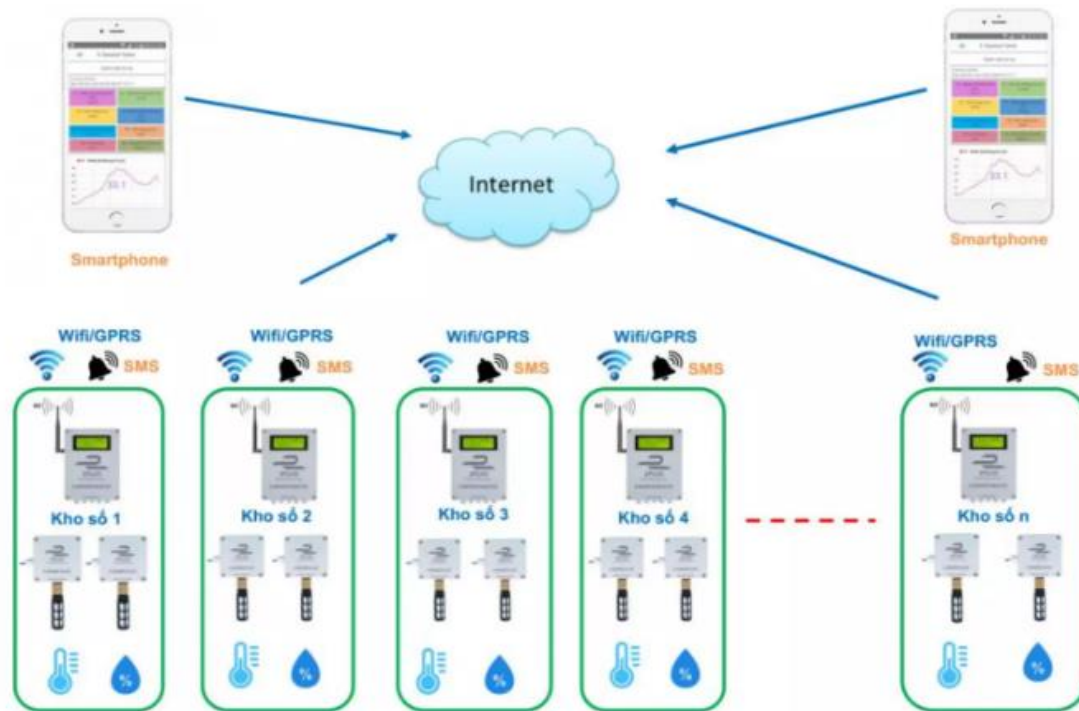
2.1.3. Ứng của Internet vào giám sát và điều khiển kho lạnh

Không quá khi nói Internet là một trong những nhân tố quan trọng nhất trong việc thay đổi thế giới trong những năm gần đây. Trong đề án này ta sẽ tập trung vào phân tích ứng dụng của nó trong công nghiệp nói chung và trong việc giám sát và điều khiển kho lạnh từ xa nói riêng.

Hiện nay trên thị trường đã có một số loại thiết bị giám sát nhiệt độ, độ ẩm của kho lạnh rồi gửi thông tin lên mạng Internet. Ví dụ như hệ thống phía dưới, dữ liệu được truyền không dây qua WiFi để gửi từ đầu cảm biến đến thiết bị thu thập. Sau đó từ các thiết bị thu thập lại gửi thông tin đó lên mạng và người truy cập có thể truy cập mạng Internet bằng điện thoại để theo dõi, giám sát hệ thống. Ngoài ra, các thông số

hoặc tin cảnh báo có thể được gửi thẳng đến điện thoại thông qua tin nhắn SMS.

2.2. Công nghệ WiFi



Hình 2.1 Một hệ thống giám sát nhiệt độ, độ ẩm qua mạng Internet

2.2.1. Giới thiệu

WiFi là một mạng thay thế cho mạng có dây thông thường, được dùng để kết nối các thiết bị ở chế độ không dây bằng việc sử dụng công nghệ sóng vô tuyến. Dữ liệu được truyền qua sóng vô tuyến cho phép các thiết bị truyền nhận dữ liệu ở tốc độ cao trong phạm vi của mạng WiFi. Kết nối các máy tính với nhau, với Internet với mạng có dây. [4]

WiFi (Wireless Fidelity) là thuật ngữ dùng chung chỉ tiêu chuẩn IEEE802.11 cho mạng cục bộ không dây (Wireless Local Networks) hoặc WLANs.

Việc sử dụng rộng rãi và tính sẵn có của nó ở nhà và nơi công cộng như công viên, trường học, ... đã khiến WiFi trở thành công nghệ truyền nhận dữ liệu phổ biến nhất hiện nay.

2.2.2. Cách thức và nguyên lý hoạt động của WiFi

Để hiểu một cách đơn giản thì WiFi chỉ là sóng vô tuyến phát từ bộ định tuyến WiFi, một thiết bị phát hiện và giải mã các sóng và sau đó gửi dữ liệu về Router. Cách thức hoạt động của WiFi giống như một đài AM/FM nhưng nó là kênh truyền thông hai chiều. WiFi sẽ hoạt động trong phạm vi rộng hơn Bluetooth. Cho nên WiFi sẽ phù hợp với các thiết bị di động như điện thoại, máy tính... [4]

2.2.3. Ứng dụng của WiFi

Với thời đại 4.0 như hiện nay, công nghệ được nâng cấp, đổi mới qua từng ngày thì WiFi vẫn được ứng dụng quan trọng và sử dụng phổ biến như một yếu tố thiết yếu không thể tách rời. WiFi được sử dụng trong các hệ thống mạng máy tính như kết nối các máy tính bàn, máy tính xách tay, máy tính bảng, điện thoại thông minh, máy in,... Điều đặc biệt ở đây là không cần đến cáp mạng cũng như kết nối Internet for device. Các địa điểm công cộng như quán ăn, quán cà phê, khu vui chơi hay đặc biệt là các quán điện tử thì WiFi là thứ vô cùng quan trọng. Có những tệp khách hàng không chỉ đến sử dụng dịch vụ của quán mà còn phải sử dụng Internet nữa, nên WiFi bây giờ là cực kỳ quan trọng không chỉ trong những công việc đặc thù mà còn trong cuộc sống hàng ngày nữa.

2.2.4. Chuẩn WiFi mới nhất hiện nay

Hiện nay, chuẩn WiFi mới nhất là chuẩn 802.11 ax, hay còn được hiểu là công nghệ WiFi thế hệ thứ 6. Chuẩn mới này được áp dụng chính thức kể từ năm 2019, sang đến năm 2020 thì mới thực sự phổ biến.

Trên thực tế, tốc độ truyền sóng WiFi của chuẩn 802.11ax có thể sẽ đạt từ 10 Gbps đến 14 Gbps hơn hẳn nhiều lần so với WiFi 5 chỉ đạt 3.5Gbps. Phiên bản mới nhất của WiFi thế hệ thứ 6 là IEEE 802.11ax-2021 được công bố phát hành ngày 9/2/2021. [4]

Tuy nhiên ưu điểm vượt trội nhưng do mới ra mắt nên nhiều thiết bị cũ trong các hệ thống không yêu cầu tốc độ cao vẫn dùng các chuẩn WiFi cũ.

2.3. Kết luận chương 2

Chương này của đồ án đã giới thiệu về Internet và công nghệ WiFi đây là hai yếu tố quan trọng để xây dựng nên một hệ thống ứng dụng Iot nói chung và trong đồ án này là ứng dụng để xây dựng nên mạch ứng dụng để giám sát và điều khiển từ xa.

CHƯƠNG 3: GIỚI THIỆU CÁC LINH KIỆN VÀ MODULE

3.1. Module ESP32-WROOM

3.1.1. Giới thiệu ESP32

ESP32 là một series các vi điều khiển trên một vi mạch giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth (tạm dịch: 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. 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. [5]

3.1.2. Cấu hình của ESP32

❖ CPU

- CPU: Xtensa Dual-Core LX6 microprocessor.
- Chạy hệ 32 bit.
- Tốc độ xử lý 160MHZ up to 240 MHz.
- Tốc độ xung nhịp đọc flash chip 40mhz đến 80mhz (tùy chỉnh khi lập trình).
- RAM: 520 KByte SRAM.
- 520 KB SRAM liền chip (trong đó 8 KB RAM RTC tốc độ cao – 8 KB RAM RTC tốc độ thấp (dùng ở chế độ DeepSleep).

❖ Hỗ trợ hai giao tiếp không dây

- Wi-Fi: 802.11 b/g/n/e/i
- Bluetooth: v4.2 BR/EDR and BLE

❖ Hỗ trợ tất cả các loại ngoại vi giao tiếp

- 8-bit DACs (digital to analog) 2 cổng
- Analog(ADC) 12-bit 16 cổng.
- I²C – 2 cổng
- UART – 3 cổng
- SPI – 3 cổng (1 cổng cho chip FLASH)
- I²S – 2 cổng
- SD card /SDIO/MMC host
- Slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN bus 2.0
- IR (TX/RX)
- Băm xung PWM (tất cả các chân)

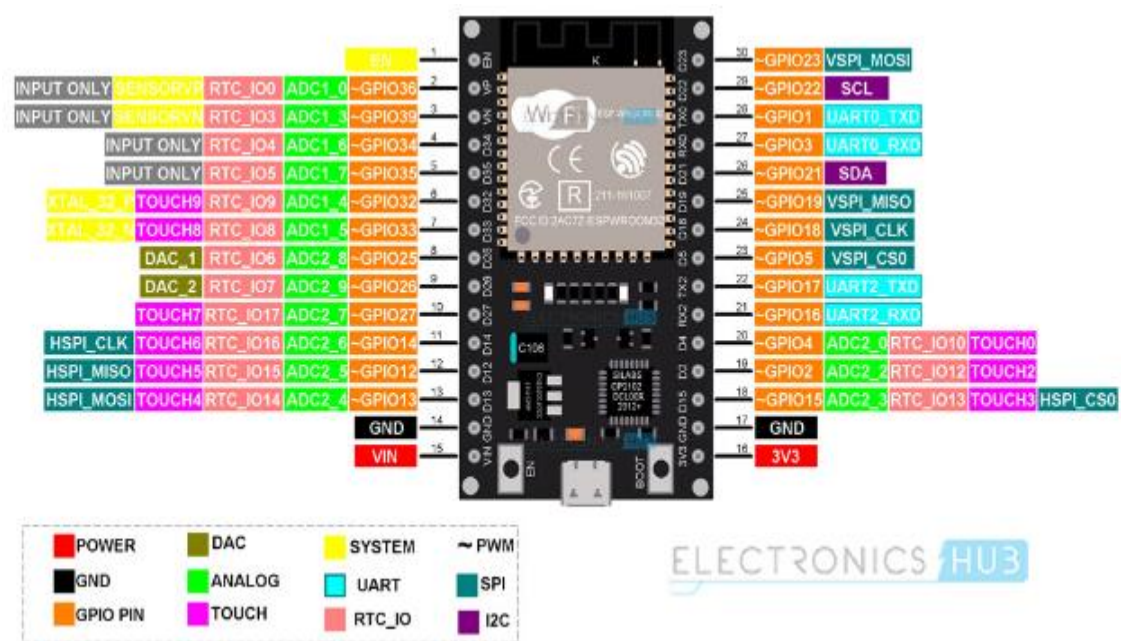
❖ Cảm biến tích hợp trên chip ESP32

- 1 cảm biến Hall (cảm biến từ trường)
- 1 cảm biến đo nhiệt độ
- Cảm biến chạm (điện dung) với 10 đầu vào khác nhau.

❖ Thông số hoạt động

- Nhiệt độ hoạt động -40°C đến 85°C
- Điện áp hoạt động: 2.2 đến 3.6 VDC

3.1.3. Sơ đồ ra chân trên Kit ESP32 DevKit



Hình 3.1 Sơ đồ ra chân và ngoại vi trên ESP32 DevKit

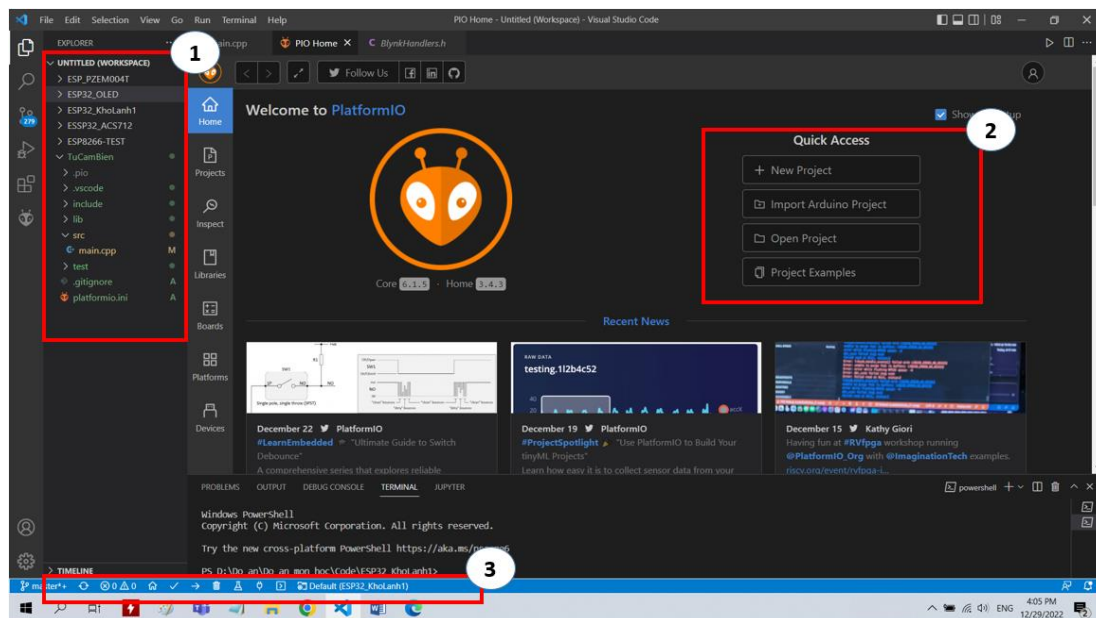
3.1.4. Môi trường lập trình

Hiện nay hai IDE được dùng nhiều nhất để lập trình cho ESP32 là VS Code và Arduino.

Với Arduino thì quá là nổi tiếng rồi, thế nhưng có những điểm hạn chế như:

- Không có khả năng tự hoàn thiện code, gợi ý (dẫn đến sai chính tả)
- Không thể view source các thư viện include vào. Dẫn tới việc bạn không biết dùng thư viện đó như thế nào, và phải dùng 1 trình soạn thảo khác đi mở ra
- Không thể tìm tới hàm gốc. Khi lập trình bạn quên mất việc truyền tham số nào vào thì với VS Code chỉ cần thao tác đơn giản là Ctr + Click, nó sẽ mở file chứa hàm gốc đó lên, thế nhưng Arduino thì không có điều này ảnh hưởng rất nhiều tới quá trình code.
- Không thể tự nhận cổng COM khi nạp code.
- Không thể chuyển đổi dễ dàng Platform của các dòng chip với nhau

VS code ra đời sau đã khắc phục được các điểm yếu của Arduino IDE.



Hình 3.2 Giao diện làm việc của VS Code

Về cơ bản thì các thao tác thông thường thì VS Code cũng tương tự như Arduino IDE. Để làm việc thì ta cần chú ý ba vùng như hình 3.2.

❖ Vùng 1:

- Workspace là khu vực chứa các thư mục chứa các dự án mà bạn đang làm việc. Người dùng có thể dễ dàng thao tác qua lại giữa các dự án. Không giống như Arduino IDE để xem code của dự án khác người dùng phải mở Tab mới.
- Trong thư mục của dự án để làm việc ta cần quan tâm đến các thư mục con như: “lib”- Nơi chứa các thư viện được thêm thủ công sử dụng trong dự án; “include”- Nơi chứa các thư viện được thêm trực tiếp bằng VS code sử dụng trong dự án; “SRC”- Chứa file main.cpp là Source Code của dự án.

❖ Vùng 2: Là khu vực để tạo dự án mới hoặc mở dự án cũ.

❖ Vùng 3: Status bar là thanh chứa các biểu tượng thao tác nhanh như: Build code, nạp code, mở giao diện Monitor.

3.2. Module đo dòng điện ACS712

Cảm biến dòng điện ACS712 (Hall Effect Current Sensor) dựa trên hiệu ứng Hall để đo dòng điện AC/DC, cảm biến có kích thước nhỏ gọn, dễ kết nối, giá trị trả ra là điện áp Analog tuyến tính theo cường độ dòng điện cần đo nên rất dễ kết nối và lập trình với Vi điều khiển, thích hợp với các ứng dụng cần đo dòng AC/DC với độ chính xác cao.

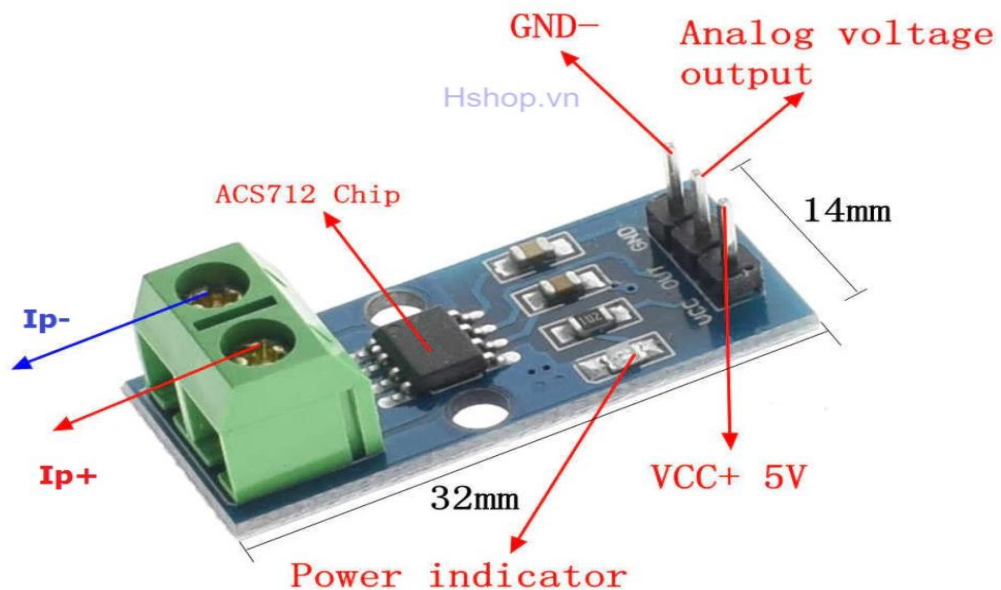
3.2.1. Thông số kỹ thuật chip ACS712

- Cảm biến dòng: AC / DC.
- Dòng điện: 5A/20A/30A.
- Đầu ra: điện áp.
- Độ nhạy: 100mV / A.
- Tần suất: DC ~ 80kHz.
- Tính tuyến tính: $\pm 1,5\%$.
- Độ chính xác: $\pm 1,5\%$.
- Điện áp: 5V.
- Thời gian đáp ứng: 5 μ s.
- Dòng điện chờ: 13mA.
- Nhiệt độ hoạt động: -40°C ~ 85°C.
- Độ nhạy:
 - ACS 712-05B (5Ampe): 180 – 190 mV/A
 - ACS 712-20A (20Ampe): 96 – 104 mV/A
 - ACS 712-30A (30Ampe): 64 – 68 mV/A



Hình 3.3 IC ACS712 loại 30A

3.2.2. Sơ đồ chân Module ACS712



Hình 3.4 Sơ đồ chân cảm biến dòng điện ACS712

3.3. Module hạ áp LM2596

Mạch hạ áp DC LM2596S 3A là mạch hạ áp nhỏ gọn được sử dụng rất nhiều do giá thành rẻ có khả năng giảm áp từ 30V xuống 1.5V mà vẫn đạt hiệu suất cao (92%) . Thích hợp cho các ứng dụng chia nguồn, hạ áp, cấp cho các thiết bị như camera, motor, robot,... Module sử dụng IC hạ áp LM2596.

3.3.1. Giới thiệu IC LM2596

LM2596 là một IC xung hạ áp được sử dụng phổ biến. Phiên bản điều chỉnh có thể nhận điện áp đầu vào từ 4,5V đến 40V và chuyển đổi nó thành nguồn điện áp thay đổi với dòng điện liên tục lên tới 3A. Khả năng dòng điện cao của nó thường được sử dụng trong các module nguồn để cấp nguồn hoặc điều khiển tải nặng.

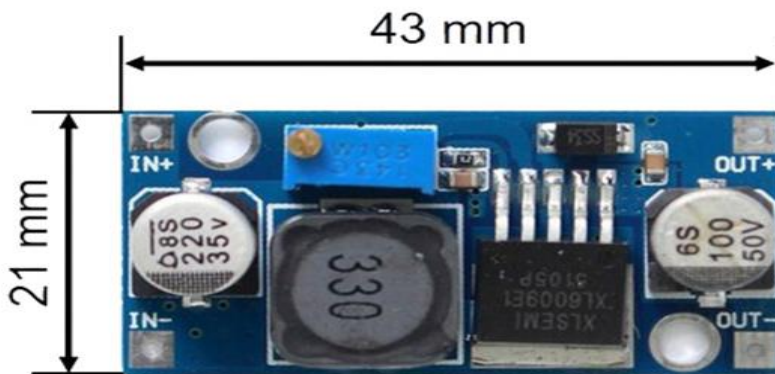
3.3.2. Thông số kỹ thuật

- Điện áp đầu vào: 4,5V đến 40V
- Điện áp đầu ra tối thiểu: 3,16V
- Dòng điện đầu ra liên tục: 3A
- Dòng đầu ra đỉnh: 6.9A
- Tần số chuyển mạch: 150KHz

3.3.3. Sơ đồ chân



Hình 3.5 Sơ đồ chân IC hạ áp LM2596



Hình 3.6 Module hạ áp sử dụng LM2596

3.4. Module đo điện AC PZEM-004T

3.4.1. Giới thiệu

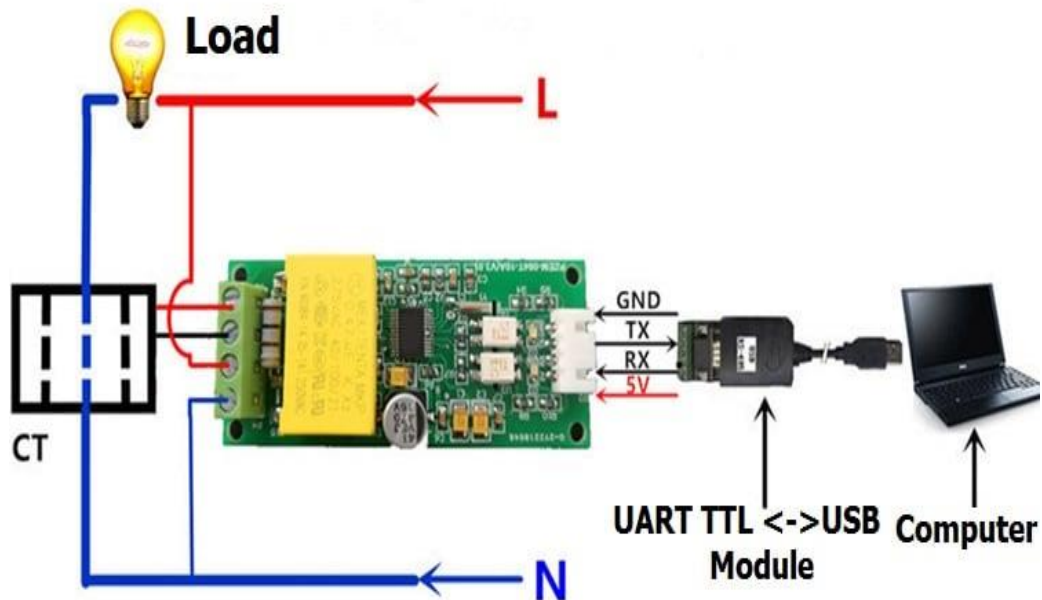
Module đo điện AC PZEM-004T được sử dụng để đo và theo dõi gần như các thông số về điện năng tiêu thụ của mạch điện như: Điện áp hoạt động, dòng tiêu thụ, công suất và năng lượng tiêu thụ, giao tiếp UART dễ dàng kết nối với vi điều khiển hoặc máy tính

3.4.2. Thông số kỹ thuật

- **Điện áp**
 - Dải đo: 80 ~ 260VAC
 - Tần số: 45 – 65Hz
 - Độ phân giải: 0.1V
 - Độ chính xác của phép đo: 0.5%
- **Dòng điện**
 - Dải đo: 0 ~ 100A
 - Phép đo bắt đầu: 0.02A
 - Độ phân giải: 0.001A
 - Độ chính xác của phép đo: 0.5%
- **Điện năng**
 - Dải đo: 0 ~ 23kW
 - Phép đo bắt đầu: 0.4W
 - Độ phân giải: 0.1W
 - Độ chính xác của phép đo: 0.5%

3.4.3. Cách đấu nối

3.5. Màn hình OLED



Hình 3.7 Module PZEM-004T và sơ đồ đấu nối

3.5.1. Giới thiệu

Màn hình Oled 0.96 inch giao tiếp I2C 2 màu cho khả năng hiển thị đẹp, sang trọng, rõ nét vào ban ngày và khả năng tiết kiệm năng lượng tối đa với mức chi phí phù hợp, màn hình Oled 0.96 inch sử dụng giao tiếp I2C cho chất lượng đường truyền ổn định và rất dễ giao tiếp chỉ với 2 chân GPIO.

3.5.2. Thông số kỹ thuật

- Điện áp sử dụng: 2.2~5.5VDC.
- Công suất tiêu thụ: 0.04w
- Góc hiển thị: lớn hơn 160 độ
- Số điểm hiển thị: 128×64 điểm.
- Độ rộng màn hình: 0.96inch
- Màu hiển thị: Vàng – Xanh
- Giao tiếp: I2C



Hình 3.8 Màn hình OLED 0.96 inch

3.6. Một số linh kiện khác

3.6.1. Cảm biến nhiệt độ NTC

NTC là nhiệt điện trở được ứng dụng nhiều trong việc đo nhiệt độ trong công nghiệp do giá thành và độ ổn định cao nên được rất tin dùng.

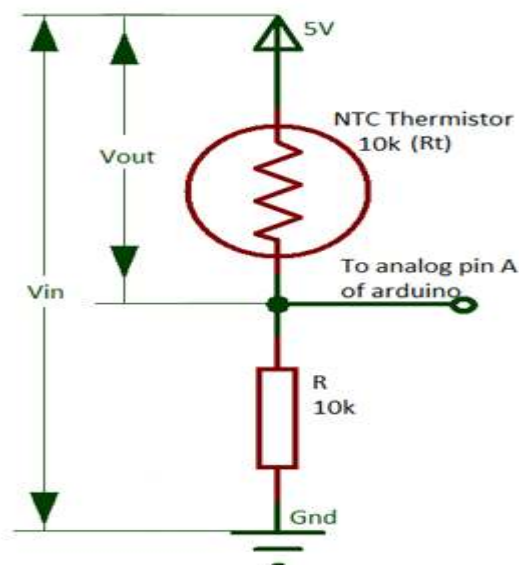
NTC là điện trở nhiệt cũng giống như cảm biến đo nhiệt độ nhưng nó chỉ hoạt động hiệu quả trong một khoảng nhiệt độ nhất định. Điện trở nhiệt NTC sẽ giảm khi nhiệt độ tăng, do đó, nó có thể được dùng để thay đổi trở kháng dưới tác dụng của nhiệt. Đây cũng chính là điều khác biệt rõ nét nhất giữa điện trở nhiệt NTC với những loại điện trở thông thường khác.

Nhiệt điện trở NTC là điện trở có hệ số nhiệt độ âm và phạm vi nhiệt độ hoạt động của NTC dao động trong khoảng từ -55°C đến 200°C .



Hình 3.9 Cảm biến nhiệt độ sử dụng điện trở NTC

Sơ đồ đấu nối



Hình 3.10 Sơ đồ đấu nối cảm biến nhiệt độ NTC

3.6.2. Cảm biến nhiệt độ DS18B20

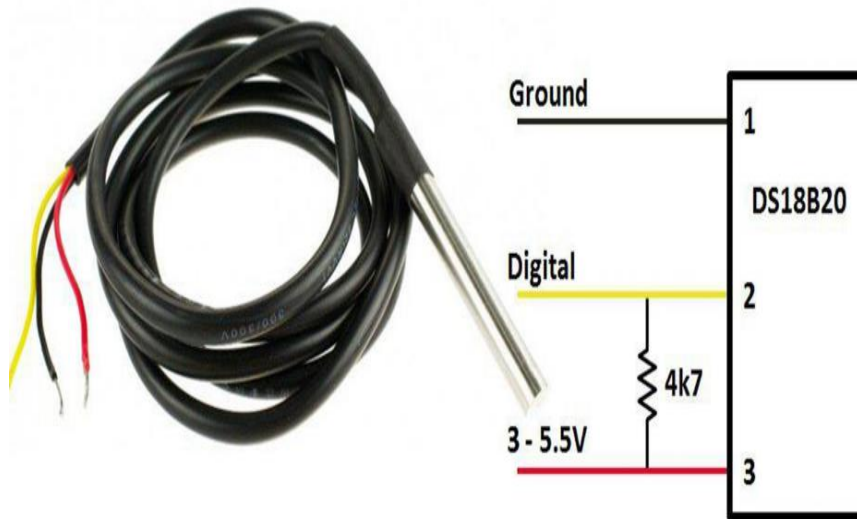
Cảm biến nhiệt độ DS18B20 là cảm biến (loại digital) đo nhiệt độ mới của hãng MAXIM với độ phân giải cao (12bit). IC sử dụng giao tiếp 1 dây rất gọn gàng, dễ lập trình. IC còn có chức năng cảnh báo nhiệt độ khi vượt ngưỡng và đặc biệt hơn là có thể cấp nguồn từ chân data (parasite power).

Cảm biến nhiệt độ này có thể hoạt động ở 125°C nhưng cáp bọc PVC nên giữ nó dưới 100°C. Đây là cảm biến kỹ thuật số, nên không bị suy hao tín hiệu đường dây dài.

Thông số kỹ thuật

- Nguồn: 3 – 5.5V
- Dải đo nhiệt độ: -55 đến 125 °C (-67 đến 257 °F)
- Sai số: +- 0.5 °C khi đo ở dải -10 – 85 °C
- Độ phân giải: người dùng có thể chọn từ 9 – 12 bits
- Chuẩn giao tiếp: 1-Wire (1 dây).
- Thời gian chuyển đổi nhiệt độ tối đa: 750ms (khi chọn độ phân giải 12bit).

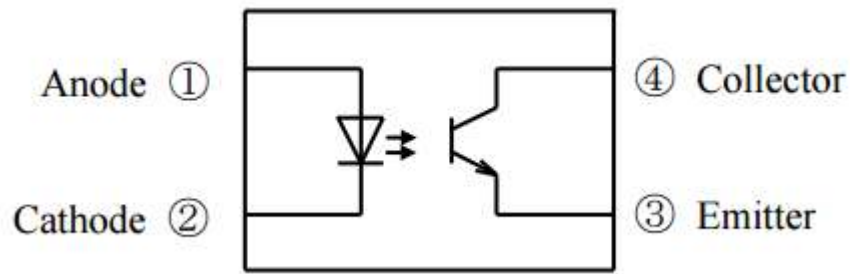
Sơ đồ đấu nối



Hình 3.11 Cảm biến DS18B20 và sơ đồ đấu nối

3.6.3. Opto quang PC817

Opto (phần tử cách ly quang) là một phần tử bán dẫn thực hiện truyền tín hiệu giữa hai phần mạch bị cách ly với nhau về điện bằng cách sử dụng ánh sáng.



Hình 3.12 Cấu tạo Opto quang

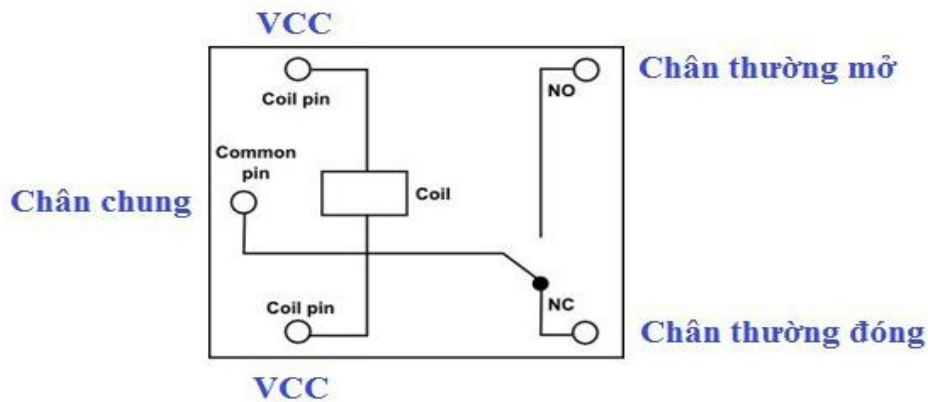
Opto được dùng trong các thiết bị số cần đến cách ly điện với nhau, nhằm tránh sốc điện hoặc gây nhiễu lẫn nhau. Các thiết bị đo đặc thí nghiệm thường được thiết kế số hóa số liệu, và truyền số liệu sang máy tính nhúng thông qua opto. Khi đó các phần mạch được cung cấp nguồn từ các module nguồn khác nhau.



Hình 3.23 Opto PC817

3.6.4. Relay

Relay (Rơ le) là một công tắc chạy bằng điện. Nhiều Relay sử dụng một nam châm điện để vận hành cơ khí công tắc, nhưng nguyên lý vận hành khác cũng được sử dụng, chẳng hạn như rơ le trạng thái rắn.



Hình 3.14 Sơ đồ chân Relay

Relay được sử dụng khi cần kiểm soát một mạch điện bằng một tín hiệu công suất thấp (với đầy đủ cách điện giữa kiểm soát và mạch điều khiển), hoặc trong trường hợp một số mạch phải được kiểm soát bởi một tín hiệu. Các relay đầu tiên được sử dụng trong các mạch điện báo đường dài với vai trò bộ khuếch đại: chúng lặp đi lặp lại các tín hiệu đến từ một mạch và truyền lại nó trên mạch khác.

3.7. Kết luận chương 3

Chương này đã giới thiệu và nêu ra đặc điểm thông số kỹ thuật của một số module và linh kiện quan trọng được dùng trong mạch điện tử của đồ án. Giúp người đọc phần nào hiểu được chức năng và ứng dụng của các module và linh kiện này trong đồ án. Chương tiếp theo ta sẽ đi vào thiết kế và thi công mạch điện tử để hiểu hơn chức năng của chúng.

CHƯƠNG 4: THIẾT KẾ VÀ THI CÔNG MẠCH ĐIỆN TỬ

4.1. Yêu cầu thiết kế và sơ đồ khối của mạch điện tử

4.1.1. Yêu cầu thiết phần cứng mạch điện tử

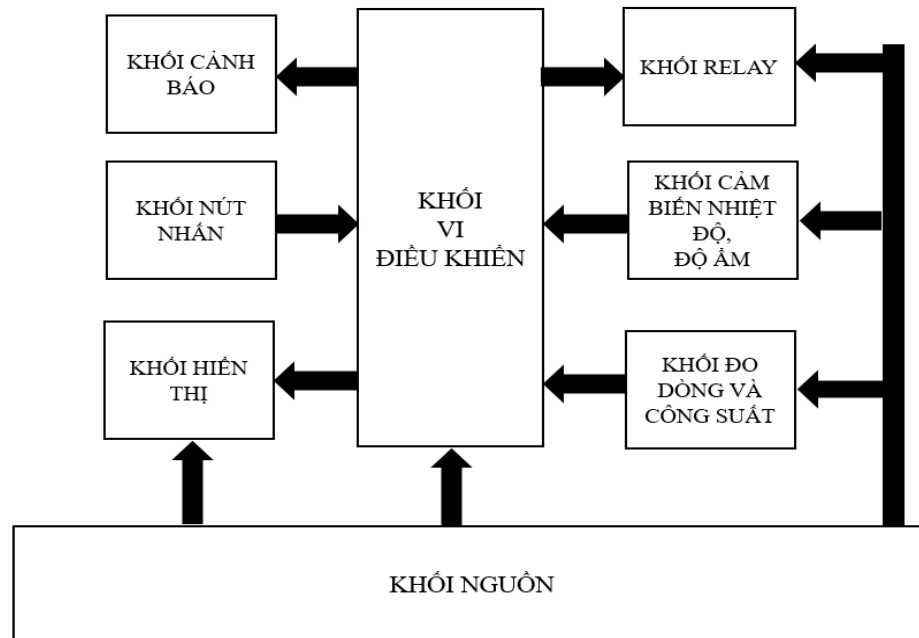
Thiết kế mạch điện tử giám sát và điều khiển đáp ứng được các yêu cầu kỹ thuật của kho lạnh:

- ❖ Có màn hình giám sát tại chỗ các thông số của kho lạnh.
- ❖ Điều khiển được các thành phần của kho lạnh như: máy nén, quạt dàn lạnh, quạt dàn nóng, xả đá.
- ❖ Có nút nhấn điều khiển tại chỗ các thiết bị và cấu hình các thông số vận hành cho kho lạnh một cách dễ dàng.
- ❖ Mạch có thể kết nối Internet để có thể giám sát và điều khiển từ xa
- ❖ Có phản hồi trạng thái của các thiết bị.
- ❖ Có các cảm biến đo đạt các thông số quan trọng của kho.
- ❖ Có thể phát tín hiệu cảnh báo khi có sự cố về thiết bị.
- ❖ Có thể tự ngắt để bảo vệ thiết bị quan trọng khi có sự cố.
- ❖ Chi phí hợp lý, gọn nhẹ dễ sử dụng và lắp đặt.

4.1.2. Phương án thiết kế mạch

- ❖ Sử dụng vi điều khiển là Module ESP 32 để có thể kết nối WiFi và xử lý các yêu cầu.
- ❖ Sử dụng hai cảm biến nhiệt độ và một cảm biến độ ẩm để đo nhiệt độ, độ ẩm trong kho và nhiệt độ dàn lạnh.
- ❖ Sử dụng hai module đo dòng điện ACS712 để giám sát dòng làm việc của máy nén và quạt dàn nóng đó là hai thiết bị rất quan trọng để gặp sự cố của kho lạnh.
- ❖ Sử dụng module đo điện PZEM-004T để theo dõi chất lượng nguồn điện.
- ❖ Sử dụng còi chip 5V làm thiết bị phát tín hiệu cảnh báo
- ❖ Sử dụng Relay 5V để tác động Contactor điều khiển các thiết bị của kho lạnh.

4.1.3. Sơ đồ khối của mạch



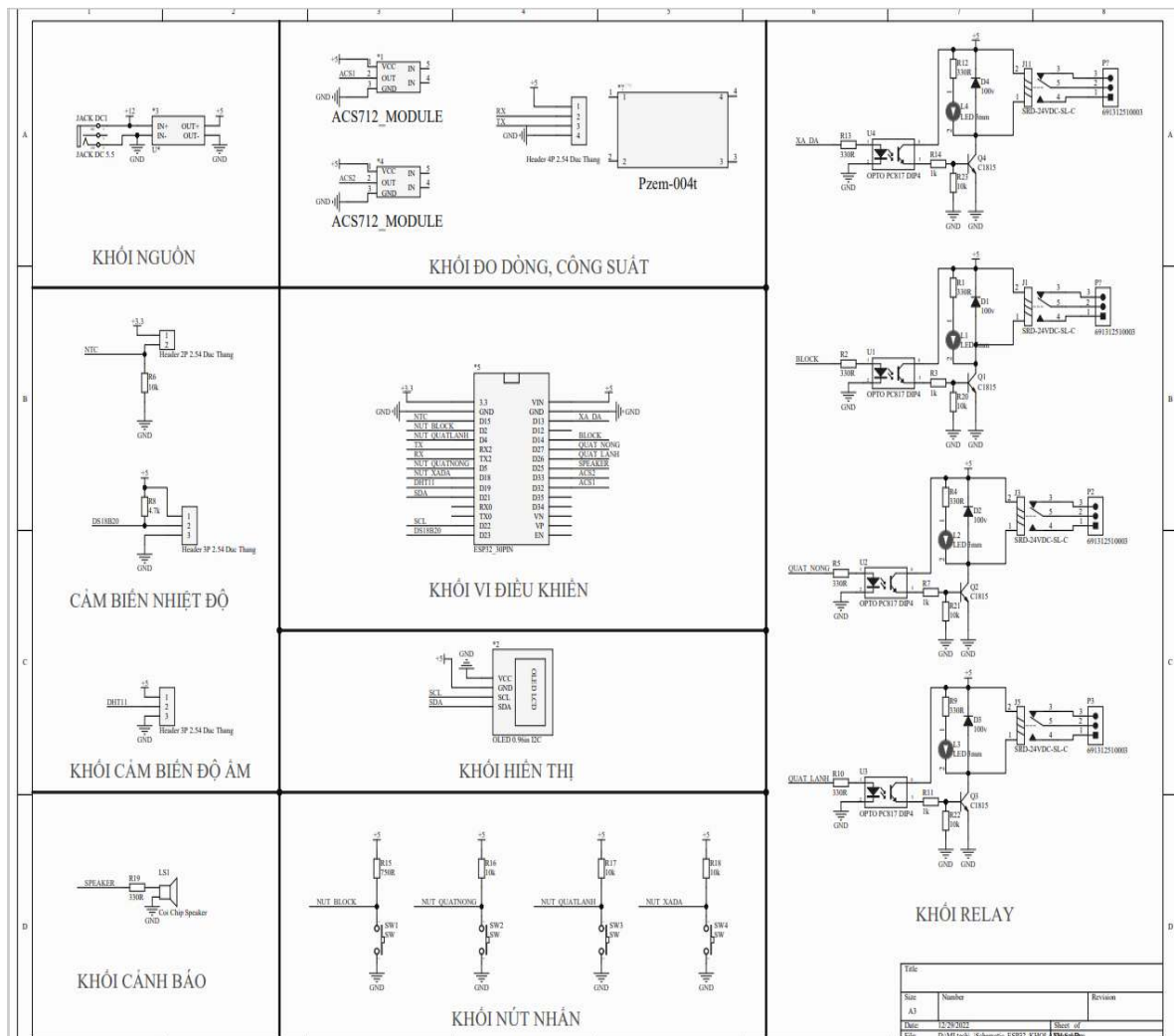
Hình 4.1 Sơ đồ khối của mạch

- Khối nguồn hạ áp điện áp 12V từ adapter cấp cho toàn mạch.
- Khối cảm biến nhiệt độ và độ ẩm sử dụng cảm biến nhiệt điện trở NTC, cảm biến nhiệt DS18B20 và cảm biến độ ẩm DHT11 có chức năng thu thập dữ liệu nhiệt độ và độ ẩm gửi về cho vi điều khiển xử lý.
- Khối cảnh báo sử dụng còi chip có chức năng phát tín hiệu cảnh báo khi có sự cố với các thiết bị hoặc nhiệt độ kho lạnh nằm ngoài ngưỡng an toàn.
- Khối nút nhấn có chức năng điều khiển các bị quan trọng của kho lạnh như: Quạt dàn nóng, quạt dàn lạnh, máy nén, điện trở xả đá và cài đặt một số thông số của kho.
- Khối Relay nối với các thiết bị động lực khác để bật, tắt các thiết bị của kho đáp ứng theo các yêu cầu vận hành của kho lạnh.
- Khối hiển thị dùng màn hình OLED hiển thị các thông số về nhiệt độ, độ ẩm, công suất và chất lượng lưới điện cung cấp cho kho lạnh.
- Khối đo dòng và công suất sử dụng module đo dòng điện ACS712 và

module đo điện PZEM-004T. Khối có chức năng đo dòng làm việc của máy nén, quạt dàn nóng, dòng điện toàn hệ thống, công suất và chất lượng nguồn điện cấp cho kho gửi về cho vi điều khiển xử lý.

- Khối vi điều khiển sử dụng kit ESP32 DevKit có chức năng xử lý các tín hiệu của cảm biến, nút nhấn, giao tiếp với Web và app trên điện thoại qua WiFi để giám sát và hiển thị, điều khiển kho lạnh.

4.1.4. Sơ đồ nguyên lý của mạch



Hình 4.2 Sơ đồ nguyên lý của mạch

4.2. Các linh kiện sử dụng trong mạch

Các linh kiện sử dụng trong từng khối của mạch được liệt kê trong bảng sau:

Bảng 4. 1 Danh sách linh kiện sử dụng trong từng khối

STT	Tên linh kiện	Số lượng	Giá trị
Khối vi điều khiển			
1	Kit ESP32 DevKit	1	
Khối nguồn			
2	Jack DC	1	
3	Module hạ áp LM2596	1	5V-3A
Khối cảm biến			
4	Module đo dòng ACS712	2	30A
5	Module đo điện PZEM-004T	1	
6	Cảm biến nhiệt điện trở NTC	1	10 k Ω
7	Cảm biến nhiệt độ DS18B20	1	
8	Cảm biến độ ẩm DHT11	1	
9	Header 4 pin	1	
10	Header 3 pin	2	
11	Header 2 pin	1	
12	Điện trở	1	10 k Ω
13	Điện trở	1	4.7 k Ω
Khối Relay			
14	Relay 5 pin	4	5V
15	Opto PC817	4	
16	Diode 1N4007	4	
17	Led đỏ	4	
18	Điện trở	4	10 k Ω
19	Điện trở	12	750 Ω
20	Transistor C1815	4	

Khối hiển thị			
21	Màn hình OLED 0.96"	1	
Khối cảnh báo			
22	Còi chip	1	
23	Điện trở	1	750 Ω
Khối nút nhấn			
24	Nút nhấn 2 chân	4	
25	Điện trở	4	10 k Ω

4.3. Thi công mạch điện tử

4.3.1. Giới thiệu phần mềm thiết kế mạch in Altium Designer

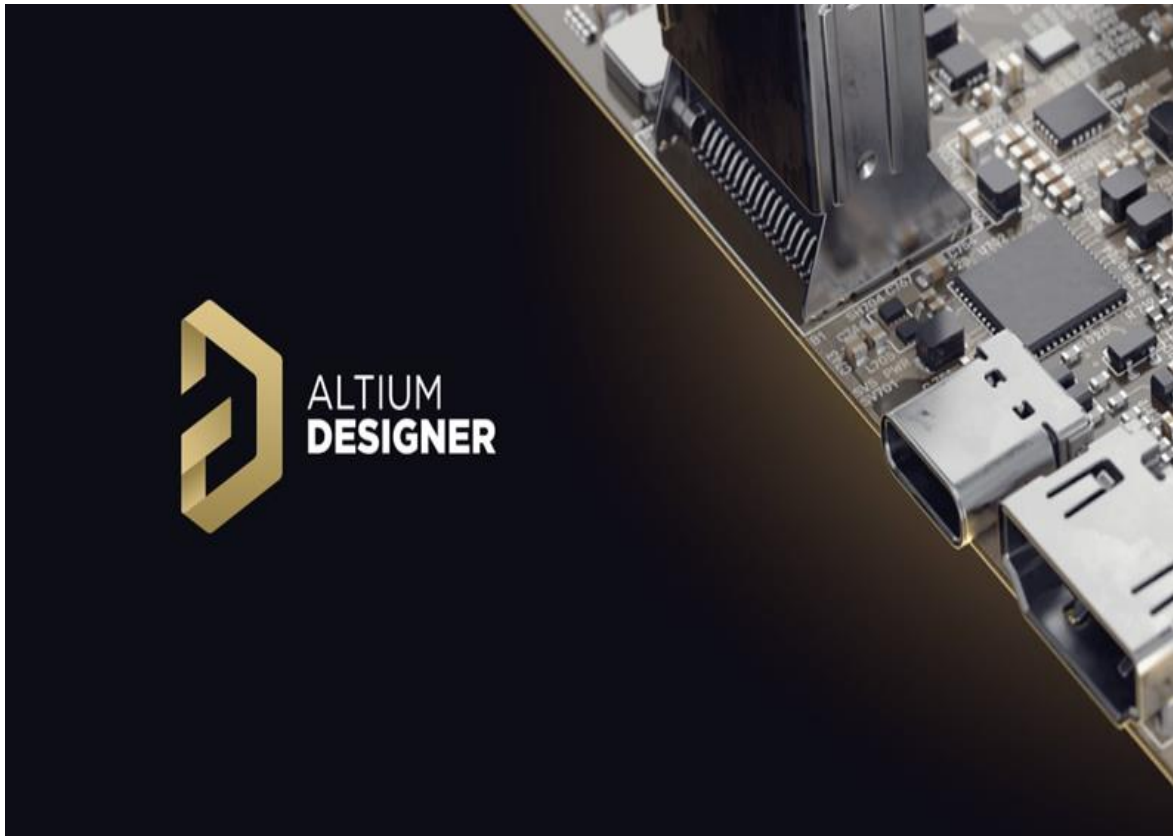
Altium Designer là một phần mềm thiết kế mạch điện tử tích hợp được phát triển bởi công ty Altium Limited có trụ sở tại Úc. Hiện nay, Altium là một trong những phần mềm vẽ mạch điện tử và thiết kế PCB được ưa chuộng ở Việt Nam.

Các tính năng của Altium Designer:

- Giao diện thiết kế, quản lý và chỉnh sửa thân thiện, dễ dàng biên dịch, quản lý file, quản lý phiên bản cho các tài liệu thiết kế.
- Hỗ trợ mạnh mẽ cho việc thiết kế tự động, đi dây tự động theo thuật toán tối ưu, phân tích lắp ráp linh kiện. Hỗ trợ việc tìm các giải pháp thiết kế hoặc chỉnh sửa mạch, linh kiện, netlist có sẵn từ trước theo các tham số mới.
- Mở, xem và in các file thiết kế mạch dễ dàng với đầy đủ các thông tin linh kiện, netlist, dữ liệu bản vẽ, kích thước, số lượng...
- Hệ thống các thư viện linh kiện phong phú, chi tiết và hoàn chỉnh bao gồm tất cả các linh kiện nhúng, số, tương tự...
- Đặt và sửa đổi tượng trên các lớp cơ khí, định nghĩa các luật thiết kế, tùy chỉnh các lớp mạch in, chuyển từ schematic sang PCB, đặt vị trí linh kiện.
- Mô phỏng mạch PCB 3D, đem lại hình ảnh mạch điện trung thực trong không gian 3 chiều, hỗ trợ MCAD-ECAD, liên kết trực tiếp với mô hình STEP, kiểm tra khoảng cách cách điện, cấu hình cho cả 2D và 3D

- Hỗ trợ thiết kế PCB sang FPGA và ngược lại.

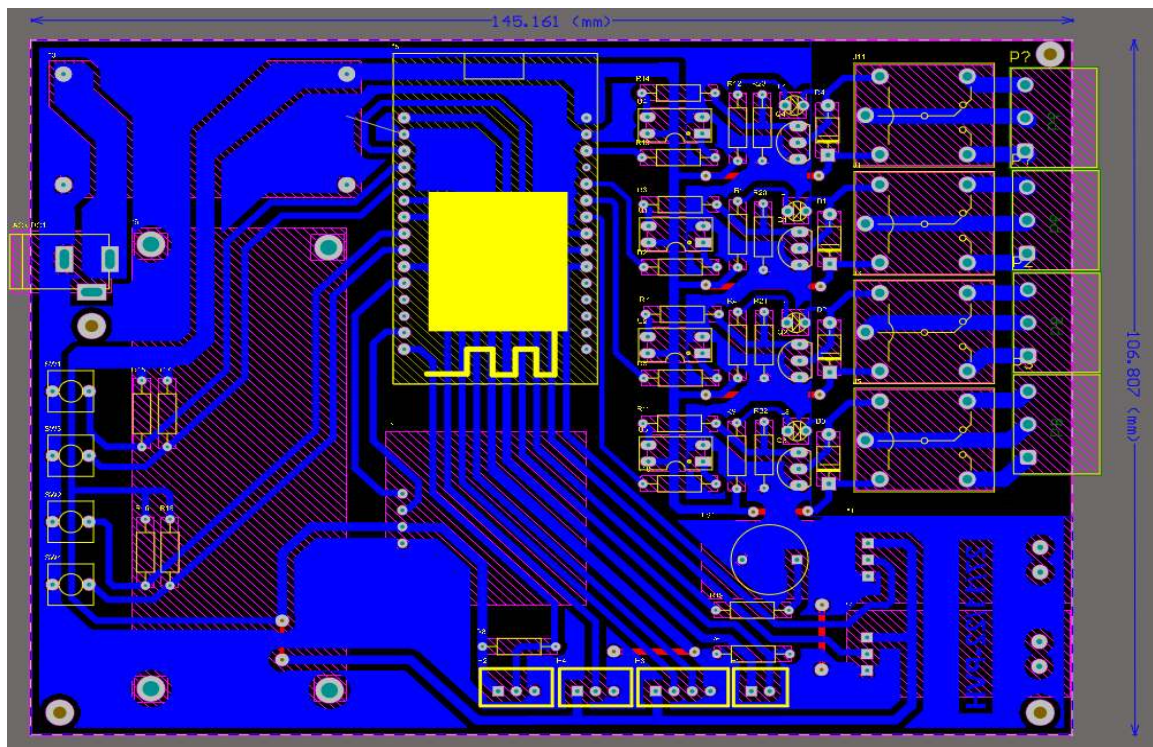
Từ đó, chúng ta thấy Altium designer có nhiều điểm mạnh so với các phần mềm khác như đặt luật thiết kế, quản lý đề tài mô phỏng dễ dàng, giao diện thân thiện,



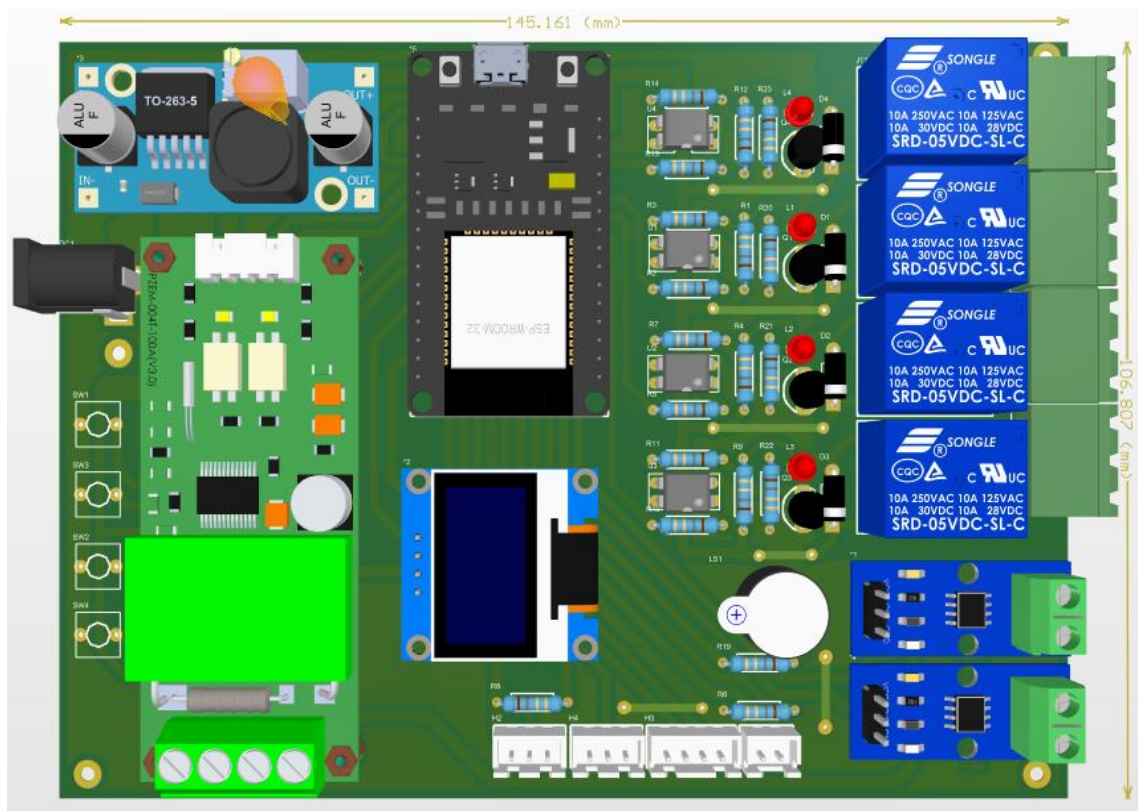
Hình 4.1 Giao diện khởi động phần mềm Altium Designer

4.3.2. Thiết kế mạch in

Sử dụng phần mềm thiết kế mạch in Altium Designer thiết kế theo sơ đồ nguyên lý đã thiết kế ở phần trước ta có được bản vẽ PCB như hình dưới.

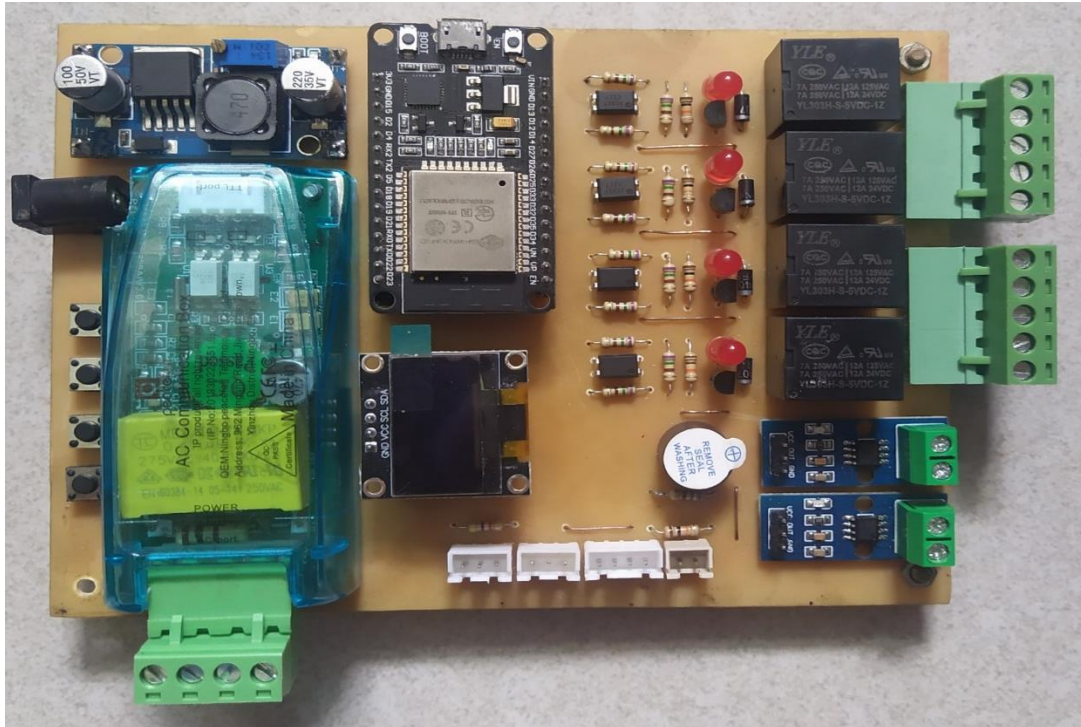


Hình 4.1 Mạch in thiết kế trên phần mềm Altium Designer



Hình 4.2 Hình 3D mô phỏng trên Altium Designer

4.4. Thi công mạch thực



Hình 4.1 Mạch thực tế sau khi hoàn thiện

4.5. Kết luận chương 4

Trong chương này, đồ án đã chỉ ra các yêu cầu trong việc thiết kế mạch giám sát và điều khiển cho kho lạnh. Liệt kê ra các thành phần của mạch và phương án hoàn thiện mạch thực. Kết quả ta đã có phần cứng mạch điện tử phù hợp và sang chương tiếp theo ta sẽ xây dựng một hệ thống giám sát và điều khiển cho kho lạnh từ xa thông qua Internet hoàn chỉnh.

CHƯƠNG 5: XÂY DỰNG GIẢI THUẬT ĐIỀU KHIỂN , GIAO DIỆN GIÁM SÁT VÀ LẬP TRÌNH

5.1. Yêu cầu xây dựng hệ thống

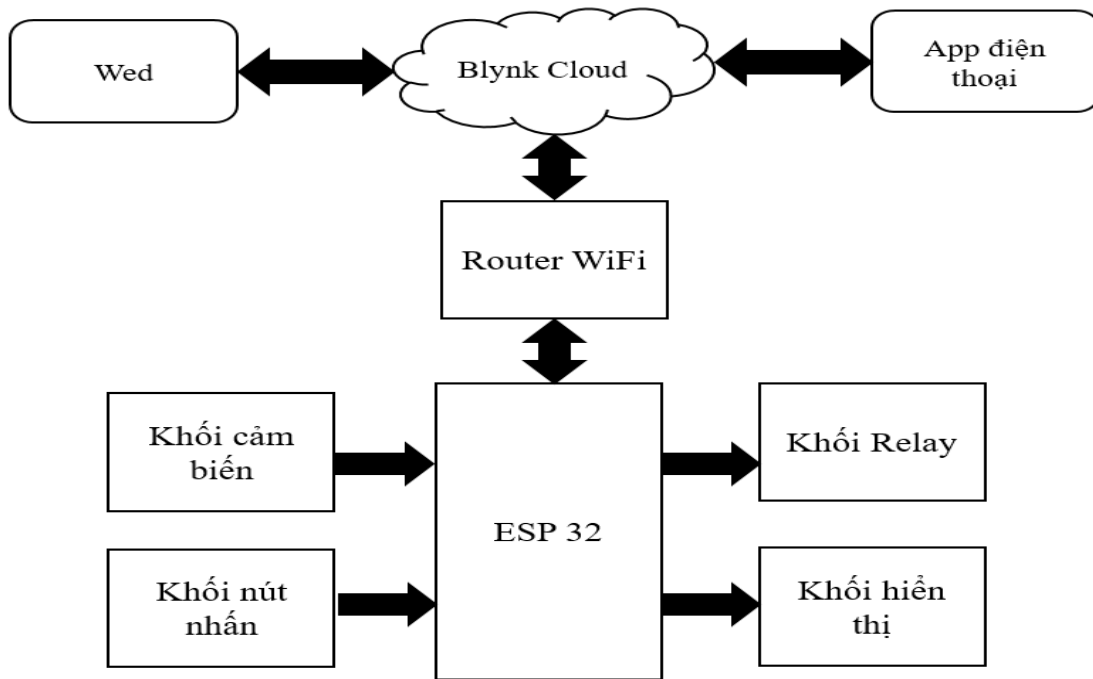
❖ Đối tượng giám sát

- Nhiệt độ, độ ẩm trong kho lạnh.
- Các giá trị chất lượng của nguồn điện như điện áp, tần số.
- Giám sát dòng làm việc của các thiết bị quan trọng và công suất làm việc của hệ thống.

❖ Mô tả bài toán:

- Bài toán giám sát và điều khiển nhiệt độ: Trong kho sẽ có hai cảm biến nhiệt độ và một cảm biến độ ẩm. Một cảm biến nhiệt độ và cảm biến độ ẩm sẽ giám sát nhiệt độ, độ ẩm trong kho; cảm biến nhiệt độ còn lại sẽ đo nhiệt độ ở dàn lạnh. Các cảm biến này sẽ liên tục gửi giá trị về ESP 32 xử lý bật, tắt các thiết bị làm lạnh (máy nén, quạt dàn lạnh) và xả đá cho phù hợp. Đồng thời các giá trị cảm biến cũng sẽ được hiển thị trên màn hình OLED và gửi lên giao diện giám sát trên web và app điện thoại cho người dùng theo dõi.
- Bài toán cảnh báo và bảo vệ thiết bị: Các cảm biến dòng điện và module đo điện sẽ liên tục giám sát trạng thái làm việc của thiết bị và cả hệ thống nếu vượt ngưỡng cho phép ESP 32 sẽ đưa ra tín hiệu cảnh báo phát ra còi chip và app điện thoại và tùy vào loại sự cố mà ESP 32 sẽ tắt thiết bị sự cố để bảo vệ hệ thống mà vẫn đảm bảo nhiệt độ trong kho lạnh.

5.2. Sơ đồ khối của hệ thống



Hình 5.1 Sơ đồ khối tổng quát của hệ thống

ESP 32 sẽ đóng vai trò là vi điều khiển trung tâm nhận tín hiệu từ khối cảm biến và nút nhấn xử lý để đưa ra tín hiệu điều khiển phù hợp. Đồng thời ESP 32 kết nối với môi trường Internet thông qua Router WiFi cũng gửi và nhận dữ liệu với Web và app điện thoại thông qua Blynk Cloud.

5.3. Giới thiệu về Blynk

Ở phần trên việc giao tiếp với giao diện Web và app điện thoại thông qua Blynk Cloud vậy Blynk là gì?

Khái niệm

Blynk là một nền tảng với các ứng dụng iOS và Android để điều khiển Arduino, Raspberry Pi và các ứng dụng tương tự qua Internet. Nó là một bảng điều khiển kỹ thuật số nhờ đó bạn có thể xây dựng giao diện đồ họa cho dự án của mình bằng cách kéo và thả các widget.

Việc thiết lập mọi thứ rất đơn giản và bạn sẽ bắt đầu sau chưa đầy 5 phút.

Blynk không bị ràng buộc với một số board hoặc shield cụ thể. Thay vào đó, nó hỗ trợ phần cứng mà bạn lựa chọn. Cho dù Arduino hoặc Raspberry Pi của bạn được

liên kết với Internet qua Wi-Fi, Ethernet hoặc chip ESP8266, Blynk sẽ giúp bạn online và sẵn sàng cho IoT.

Cách hoạt động của Blynk

Blynk được thiết kế cho IoT. Nó có thể điều khiển phần cứng từ xa, nó có thể hiển thị dữ liệu cảm biến, nó có thể lưu trữ dữ liệu, trực quan hóa và làm nhiều thứ hay ho khác.

Có ba thành phần chính trong nền tảng:

Ứng dụng Blynk - cho phép bạn tạo giao diện cho các dự án của mình bằng cách sử dụng các widget khác nhau.

Blynk Server - chịu trách nhiệm về tất cả các giao tiếp giữa điện thoại thông minh và phần cứng. Bạn có thể sử dụng Blynk Cloud hoặc chạy cục bộ máy chủ Blynk riêng của mình. Nó là mã nguồn mở, có thể dễ dàng xử lý hàng nghìn thiết bị và thậm chí có thể được khởi chạy trên Raspberry Pi.

Thư viện Blynk - dành cho tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và lệnh đi.

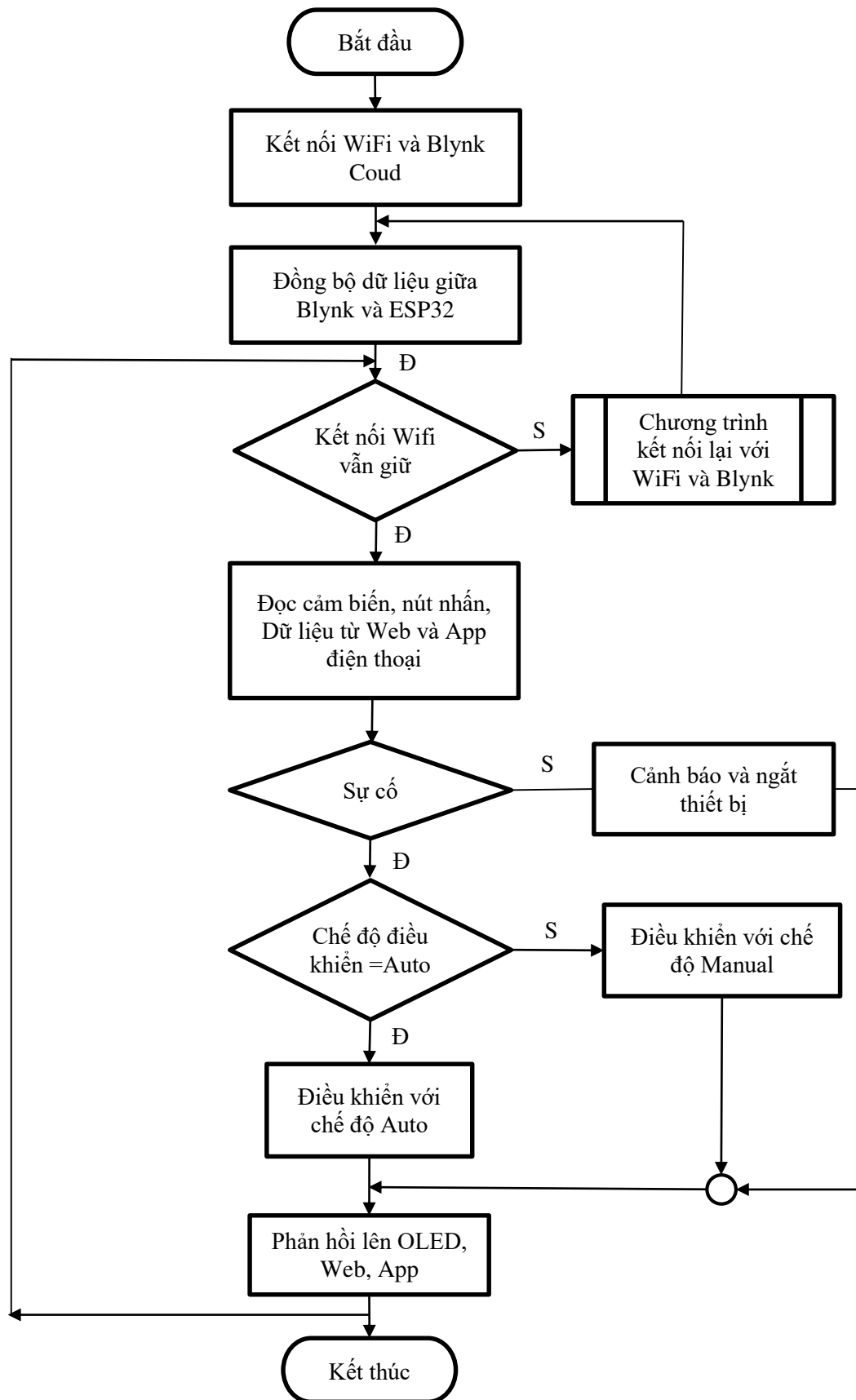
Mỗi khi bạn nhấn một nút trong ứng dụng Blynk, thông điệp sẽ truyền đến không gian của đám mây Blynk, và tìm đường đến phần cứng của bạn.



Hình 5.2 Logo biểu tượng của Blynk

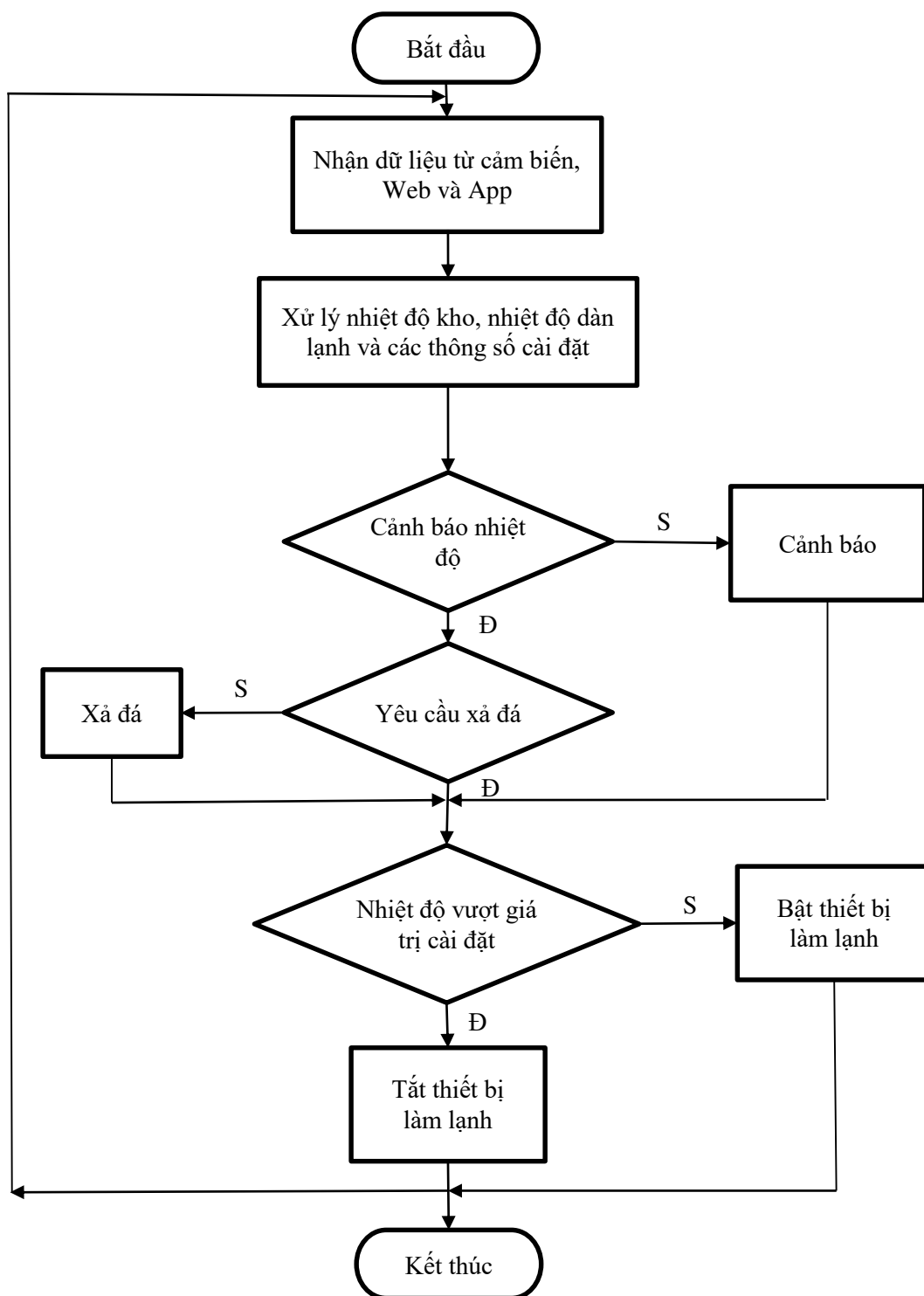
5.4. Lưu đồ thuật toán

5.4.1. Lưu đồ thuật toán của hệ thống



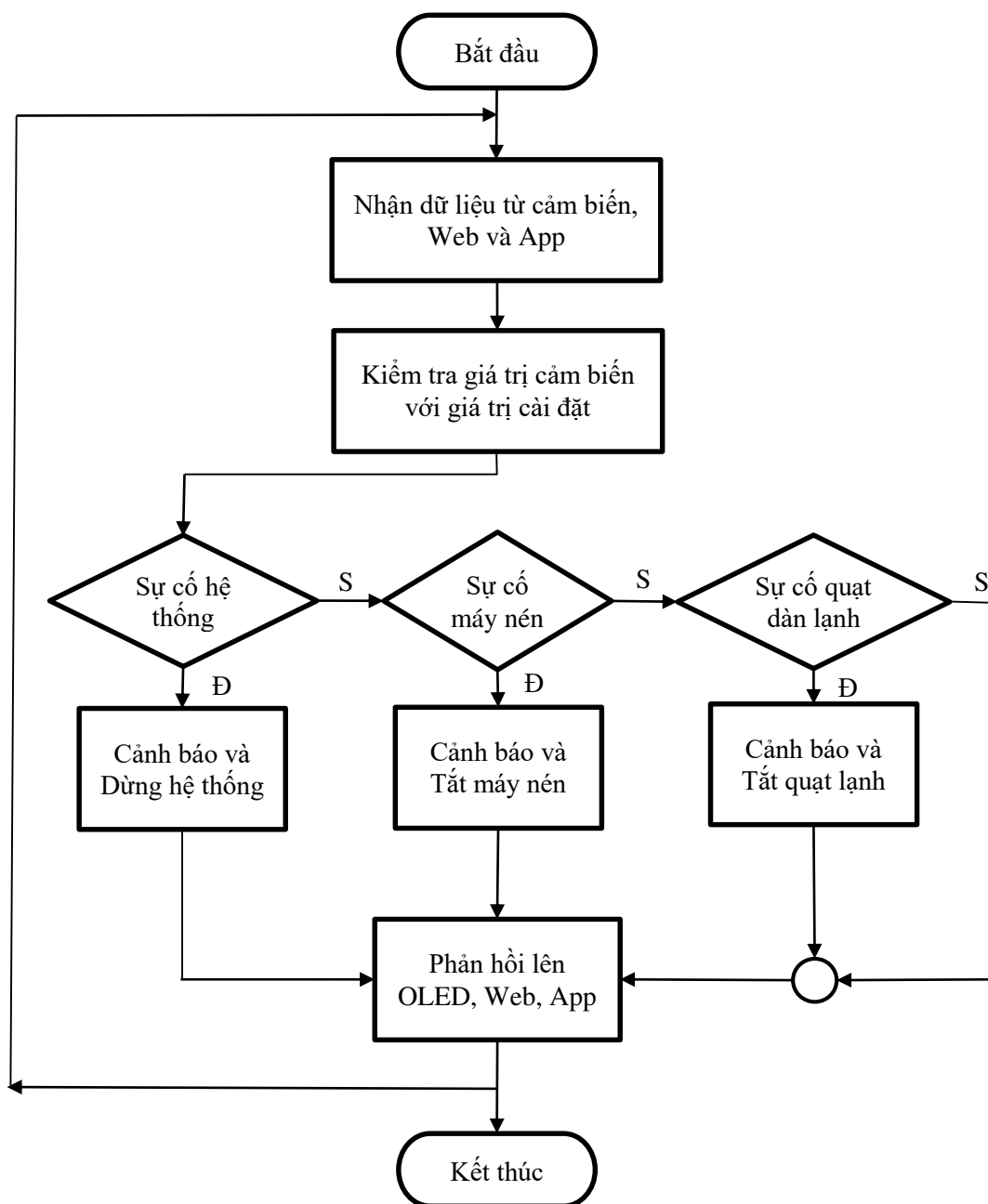
Hình 5.3 Lưu đồ của hệ thống

5.4.2. Lưu đồ thuật toán chế độ điều khiển Auto



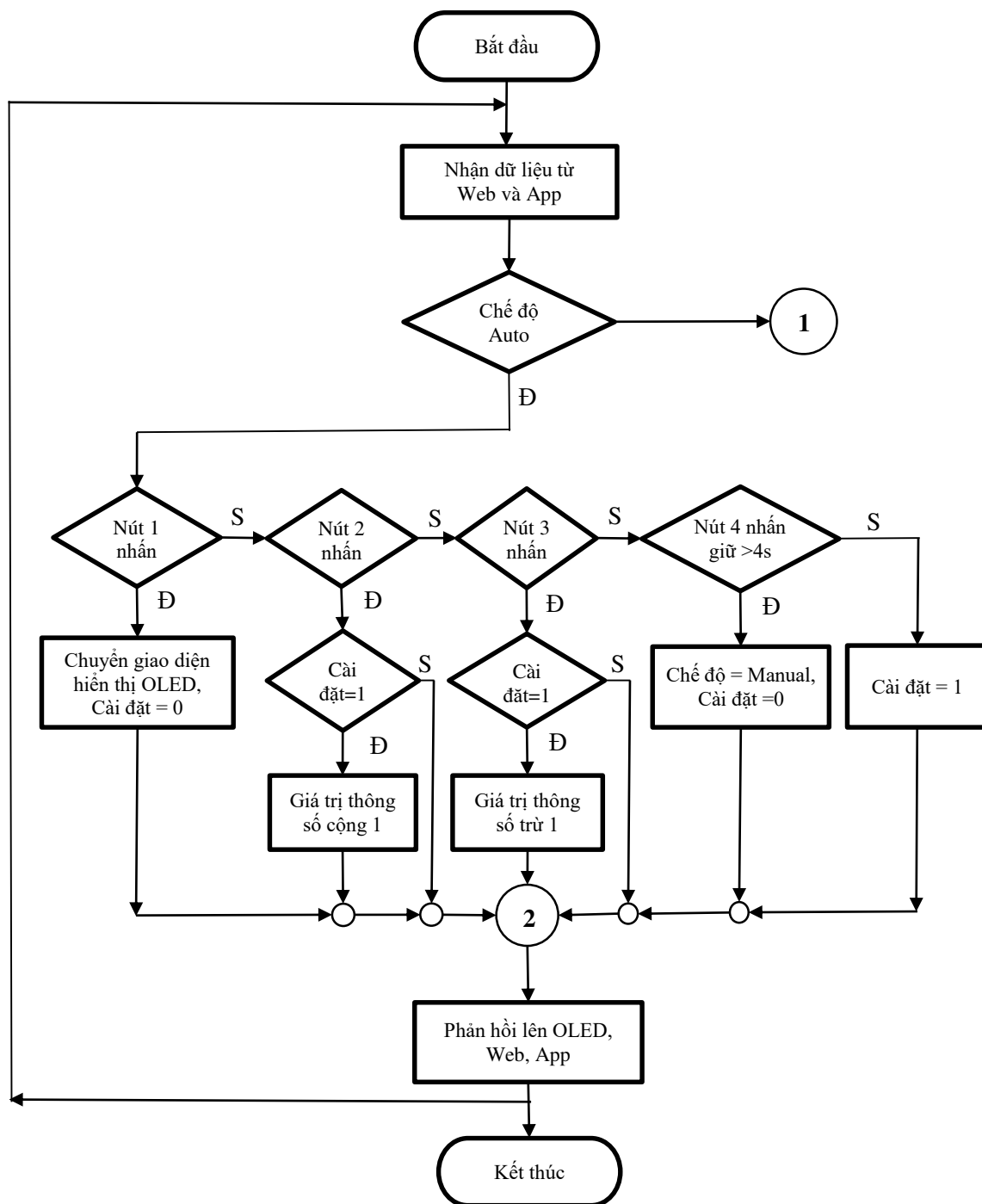
Hình 5.4 Lưu đồ chế độ Auto

5.4.3. Lưu đồ thuật toán khối sự cố và khối cảnh báo, bảo vệ thiết bị

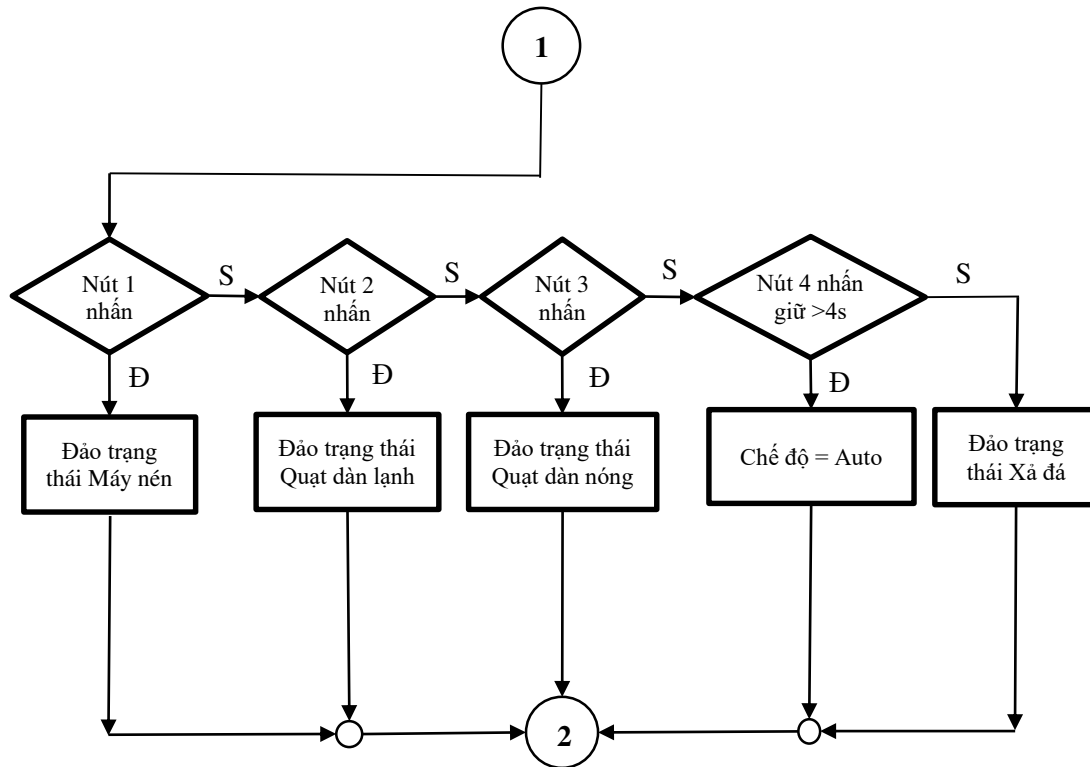


Hình 5.5 Lưu đồ khối sự cố và khối cảnh báo, bảo vệ thiết bị

5.4.4. Lưu đồ thuật toán nút nhấn



Hình 5.6 Lưu đồ nút nhấn (phần 1)



Hình 5.7 Lưu đồ nút nhấn (phần 2)

5.5. Xây dựng giao diện giám sát kho lạnh trên Web Blynk và App Blynk

5.5.1. Thiết kế giao diện giám sát trên Web Blynk

Để thiết kế giao diện giám sát sử dụng trên Web trước khi thiết kế giao diện ta vào mục Datastreams để tạo các datastreams để giao tiếp với dữ liệu với Blynk Cloud và ESP32.

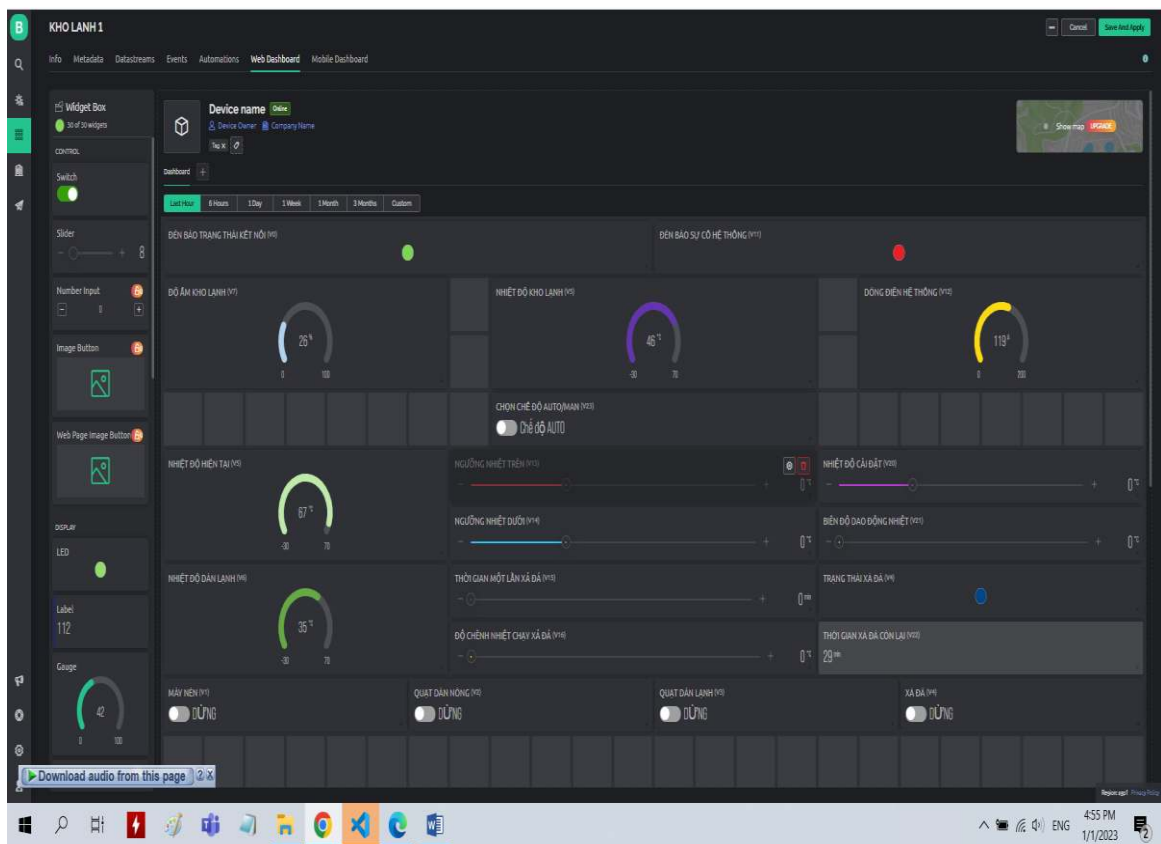
The screenshot shows the Blynk Web Dashboard interface. At the top, there's a header with the project name 'KHO LANH 1' and buttons for 'Duplicate' and 'Edit'. Below the header is a navigation bar with tabs: 'Info', 'Metadata', 'Datastreams' (selected), 'Events', 'Automations', 'Web Dashboard', and 'Mobile Dashboard'. A search bar labeled 'Search datastream' is present. The main area displays a table of datastreams.

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Decimals	Default Value
5	COLD STORAGE TEMPERATURE	COLD STORAGE TEMPERATURE	Blue	V5	Double	°C	false	-30	70	###	
1	COMPRESSOR	COMPRESSOR	Orange	V1	Integer		false	0	1	-	0
8	COMPRESSOR CURRENT	COMPRESSOR CURRENT	Yellow	V8	Double	A	false	0	100	###	
2	CONDENSER FAN	CONDENSER FAN	Light Blue	V2	Integer		false	0	1	-	0
9	CONDENSER FAN CURRENT	CONDENSER FAN CURRENT	Light Green	V9	Double	A	false	0	100	###	
11	CONNECT	CONNECT	Red	V0	Integer		false	0	1	-	0
4	DEFROST	DEFROST	Dark Blue	V4	Integer		false	0	1	-	0
6	DEFROST TEMPERATURE	DEFROST TEMPERATURE	Light Blue	V6	Double	°C	false	-30	70	###	
17	DO CHENH NHiet	DO CHENH NHiet	Blue	V16	Double	°C	false	0	50	###	
22	DO DAO DONG NHiet	DO DAO DONG NHiet	White	V21	Double	°C	false	0	50	###	
13	DONG HE THONG	DONG HE THONG	Light Green	V12	Double	A	false	0	200	###	

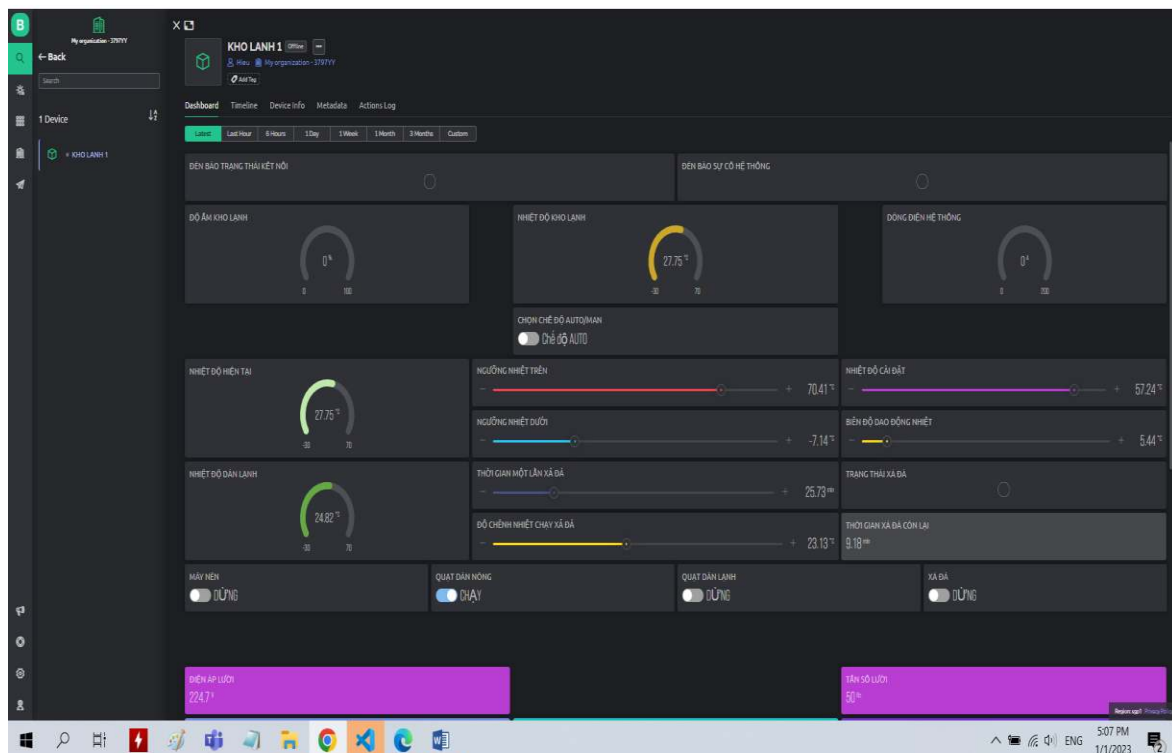
At the bottom right of the table, there are links for 'Register', 'Sign In', and 'Privacy Policy'. The Windows taskbar is visible at the very bottom of the image.

Hình 5.8 Các datastreams giao tiếp dữ liệu

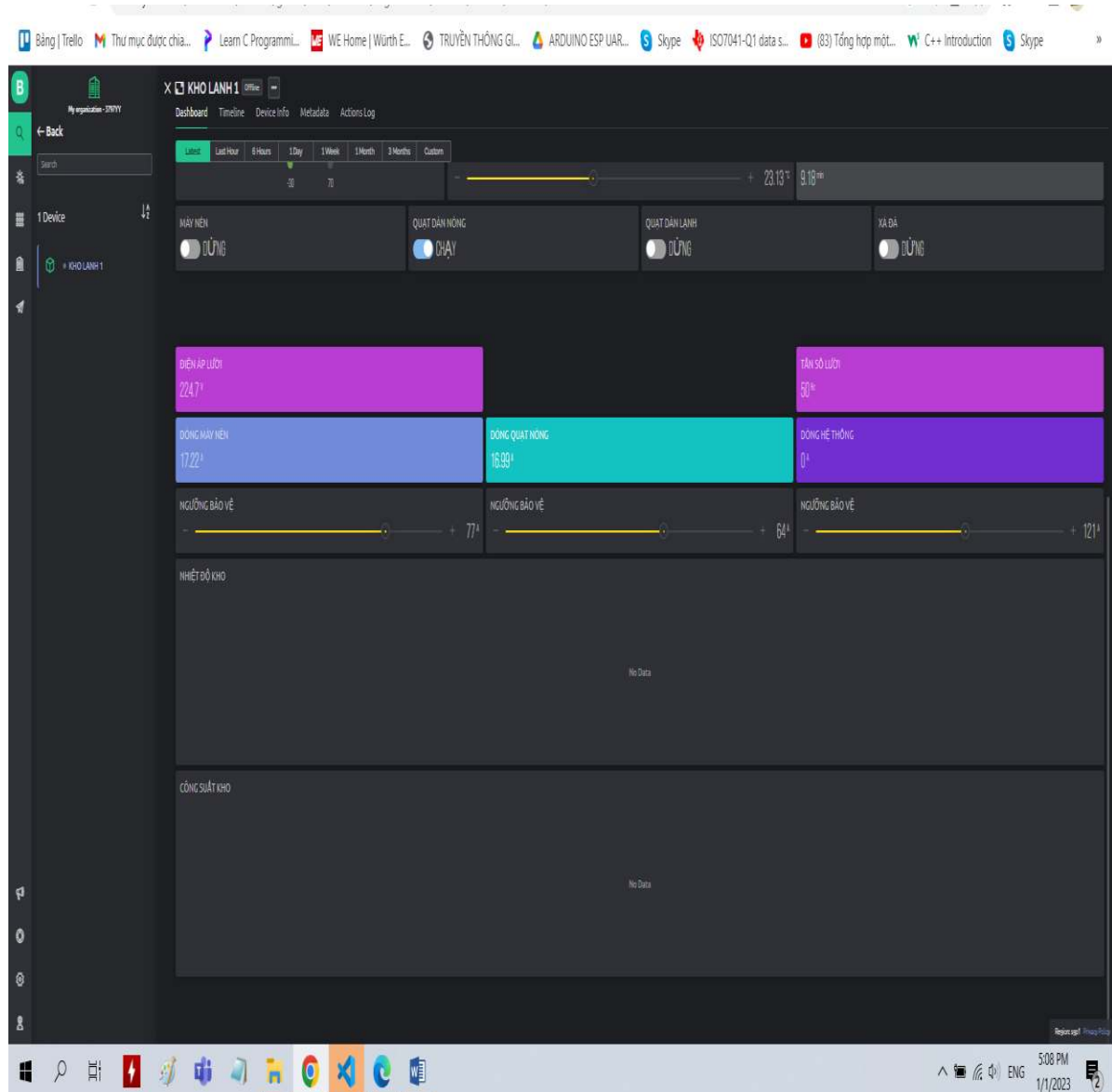
Vào mục Web Dashboard để thiết kế giao diện Web. Blynk thiết kế bằng phương pháp kéo thả các khối có sẵn và chọn các Datastreams dữ liệu phù hợp với mục đích sử dụng.



Hình 5.9 Thiết kế giao diện bằng các kéo thả các khối phù hợp

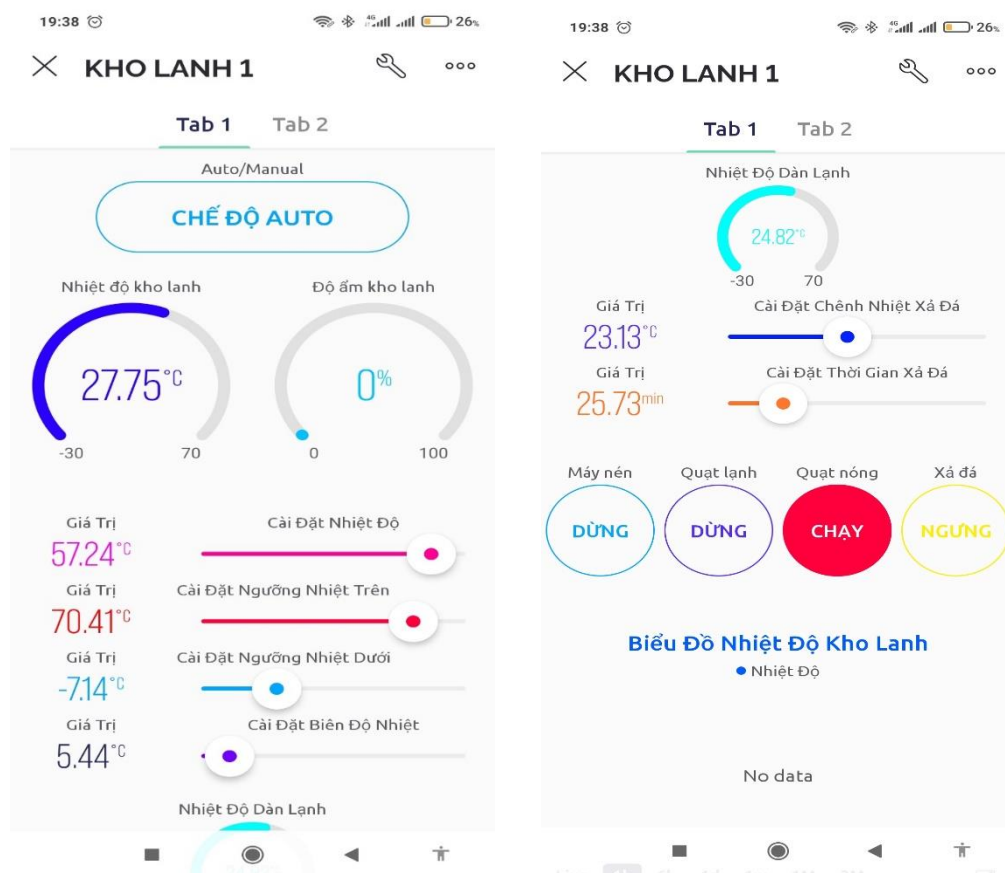


Hình 5.10 Giao diện Web hoàn thiện (phần 1)



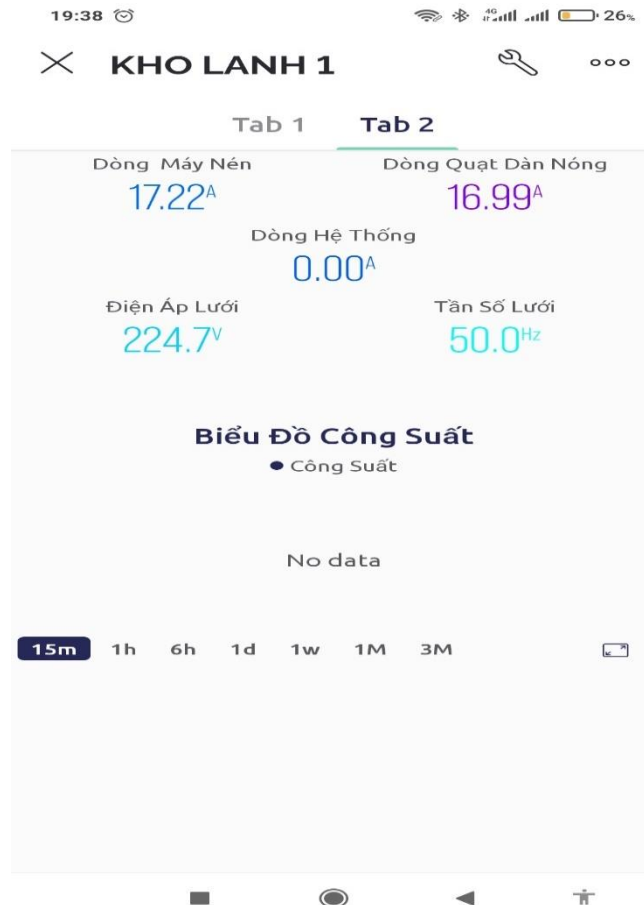
Giao diện các cá biểu đồ Gauge hiển thị giá trị của nhiệt độ kho, độ ẩm , và dòng điện vận hành của kho; có các thanh dạng slider để chỉnh các thồn số cài đặt; có bốn switch để điều khiển thiết bị và hiển thị trạng thái của thiết bị; có các ô để hiển thị giá trị các thông số như điện áp, tần số lưới điện, dòng làm việc của một số thiết bị quan trọng; đồng thời web cũng có hai biểu đồ biểu diễn giá của nhiệt độ và công suất giúp người vận hành dễ giám sát và theo dõi quá trình vận hành của kho lạnh.

5.5.2. Thiết kế giao diện App điện thoại



Hình 5.1 Giao diện Tab 1 trên app điện thoại

Tab 1 là màn hình giám sát các thông số nhiệt độ, độ ẩm của kho lạnh, các nút điều khiển các thiết bị cũng như cài đặt các thông số để vận hành kho lạnh.



Hình 5.2 Giao diện Tab 2 trên app điện thoại

Giao diện Tab 2 theo dõi các thông số vận hành của hệ thống như điện áp, tần số lưới, dòng điện, công suất. Người dùng chuyển Tab bằng cách nhấn vào Tab 1 hoặc Tab 2 phía trên màn hình.

5.6. Vận hành thử theo các bài toán giả định của kho lạnh

5.6.1. Hoạt động của mạch

Các thông số cần cài đặt để vận hành kho lạnh:

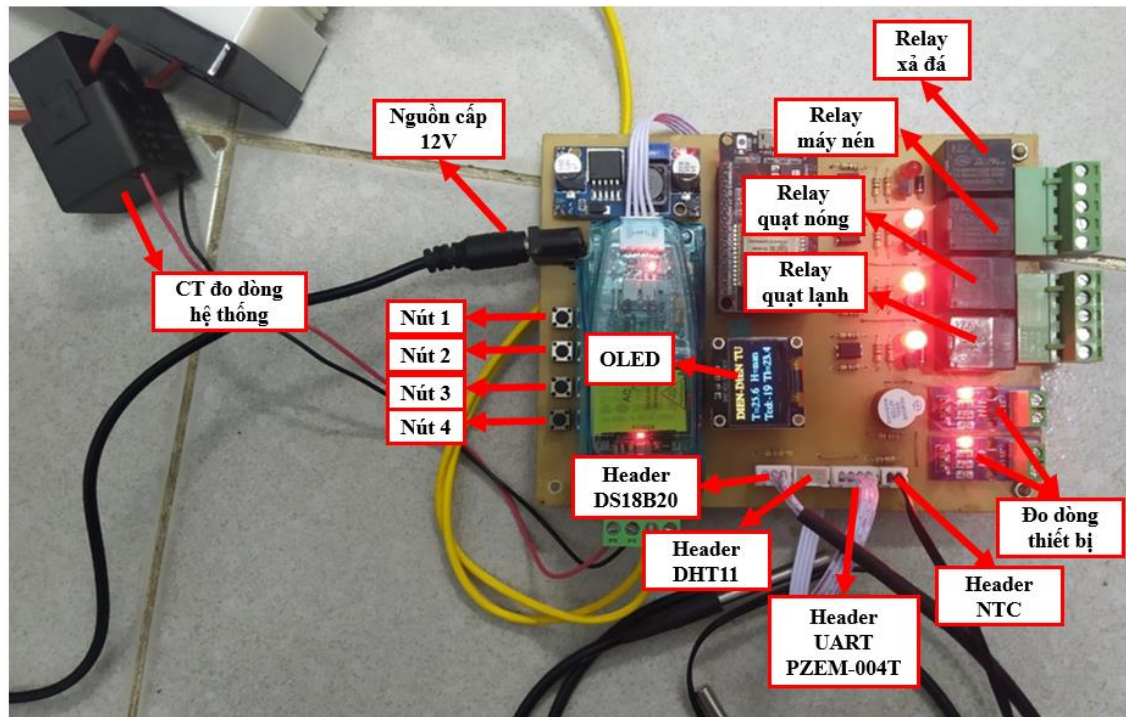
- Nhiệt độ cài đặt: Nhiệt độ kho lạnh mong muốn.
- Biên độ dao động nhiệt: Độ chênh lệch nhiệt độ so với nhiệt độ cài đặt để tác động các thiết bị làm lạnh.
- Ngưỡng nhiệt trên/dưới: Nhiệt độ mà khi vượt quá hoặc dưới sẽ có tín hiệu cảnh báo.

- Thời gian xả đá: Thời gian một lần xả đá.
- Độ chênh nhiệt xả đá: Nếu nhiệt độ trong kho lạnh lớn hơn nhiệt độ trong dàn lạnh bằng độ chênh nhiệt xả đá thì chế độ xả đá hoạt động.
- Ngưỡng bảo vệ máy nén: Giá trị mà nếu dòng làm việc máy nén vượt quá thì xem như là đang gặp sự cố.
- Ngưỡng bảo vệ quạt dàn nóng: Giá trị mà nếu dòng làm việc quạt dàn nóng vượt quá thì xem như là đang gặp sự cố.
- Ngưỡng bảo vệ hệ thống: Giá trị mà nếu dòng làm việc của cả hệ thống vượt quá thì xem như là đang gặp sự cố.

Các tình huống vận hành để kiểm tra hoạt động của mạch:

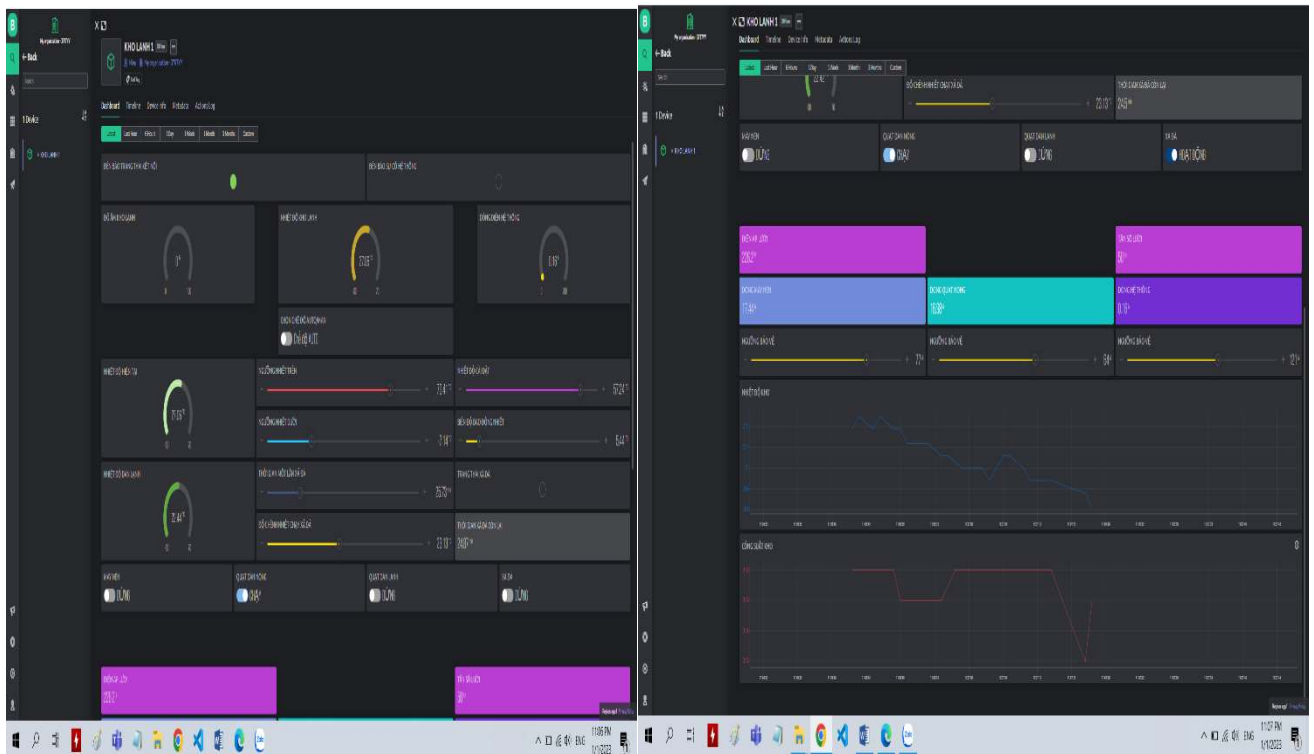
- ❖ Tình huống 1: Vận hành bình thường ở chế độ Auto
 - Khi nhiệt độ trong kho lạnh vượt quá “nhiệt độ cài đặt” + “biên độ dao động nhiệt” thì sau 1 phút (thời gian xác nhận sự kiện) kể từ khi nhận tín hiệu nếu nhiệt độ vẫn vượt quá thì quạt dàn lạnh và máy nén sẽ bật ngược lại nếu sau 1 phút tín hiệu là sai thì kho vẫn hoạt động như trạng thái cũ.
 - Khi nhiệt độ trong kho lạnh hạ dưới “nhiệt độ cài đặt” – “biên độ dao động nhiệt” thì sau 1 phút (thời gian xác nhận sự kiện) kể từ khi nhận tín hiệu nếu nhiệt độ vẫn dưới ngưỡng thì quạt dàn lạnh và máy nén sẽ tắt ngược lại nếu sau 1 phút tín hiệu là sai thì kho vẫn hoạt động như trạng thái cũ.
 - Khi nhiệt độ trong kho lạnh – nhiệt độ trong dàn lạnh lớn hơn “độ chênh nhiệt xả đá” sau 3 phút (thời gian xác nhận sự kiện) nếu sự kiện là đúng thì chế độ xả đá được kích hoạt máy nén, quạt dàn lạnh sẽ được tắt ngược lại nếu sau 3 phút sự kiện là sai thì kho vẫn hoạt động ở trạng thái cũ.
- ❖ Tình huống 2: Vận hành bình thường ở chế độ Manual
 - Khi có tín hiệu chuyển chế độ sang Manual từ nút nhấn/Web/App điện thoại thì người vận hành sẽ sử dụng nút nhấn để vận hành kho lạnh. Nút 1 điều khiển máy nén, nút 2 điều khiển quạt dàn lạnh, nút 3 điều khiển quạt dàn nóng, nút 4 điều khiển chế độ xả đá.

- ❖ Tình huống 3: Kho lạnh gặp lỗi hoặc sự cố với thiết bị
 - Khi nhiệt độ kho lạnh vượt quá hoặc dưới ngưỡng cảnh báo thì sẽ có tín hiệu cảnh báo bằng còi chip đồng thời màn hình OLED sẽ hiển thị lỗi về nhiệt độ và sẽ có thông báo cảnh báo gửi về điện thoại.
 - Khi có sự cố với thiết bị (quá dòng hoặc đứt mạch) thì còi chip phát tín hiệu cảnh báo, màn hình OLED hiển thị sự cố về thiết bị và thông báo cảnh báo sự cố được gửi về điện thoại.
 - Khi mất kết nối WiFi thì tín hiệu cảnh báo được phát và hệ thống sẽ tự kết nối lại WiFi. Mất hết kết nối WiFi thì mạch vẫn hoạt động bình thường cả ở chế độ Auto và Manual.

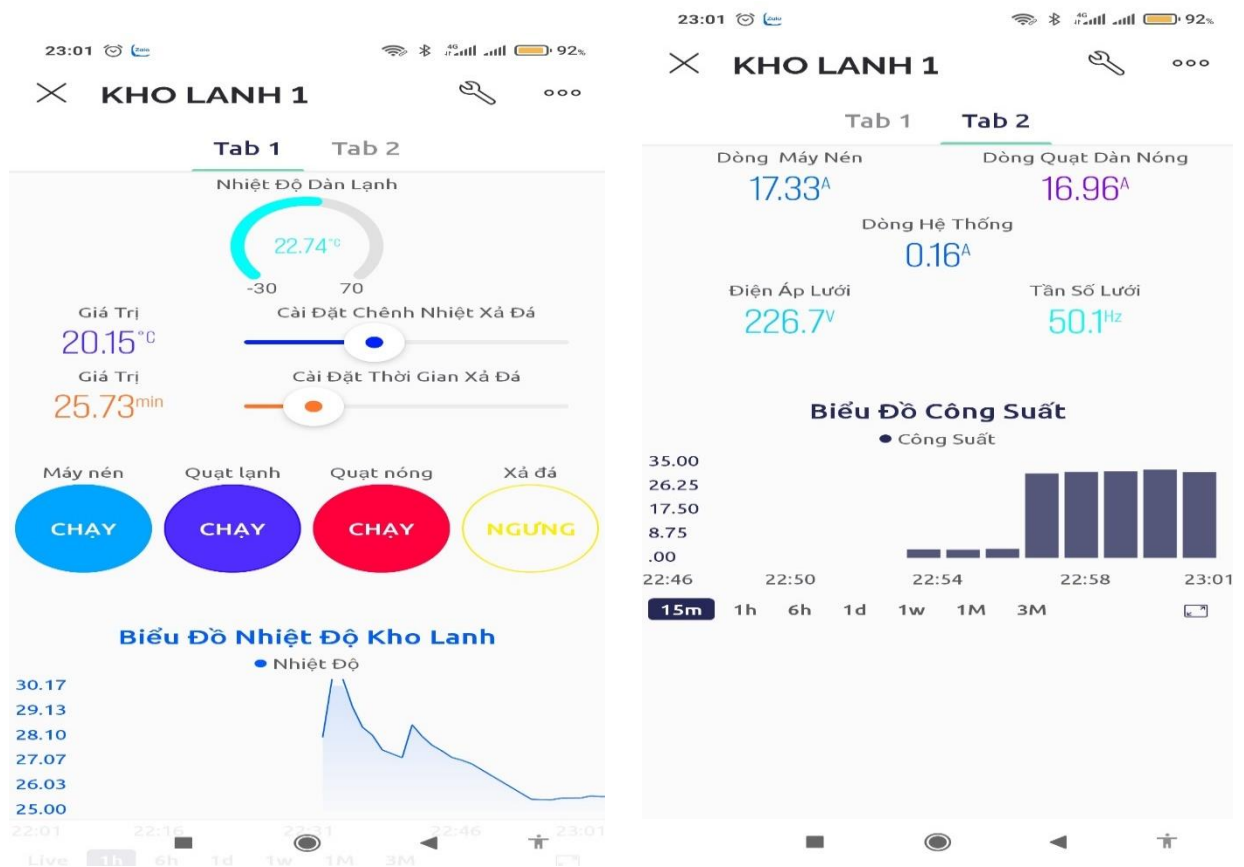


Hình 5.1 Kết nối mạch chạy thử các tình huống

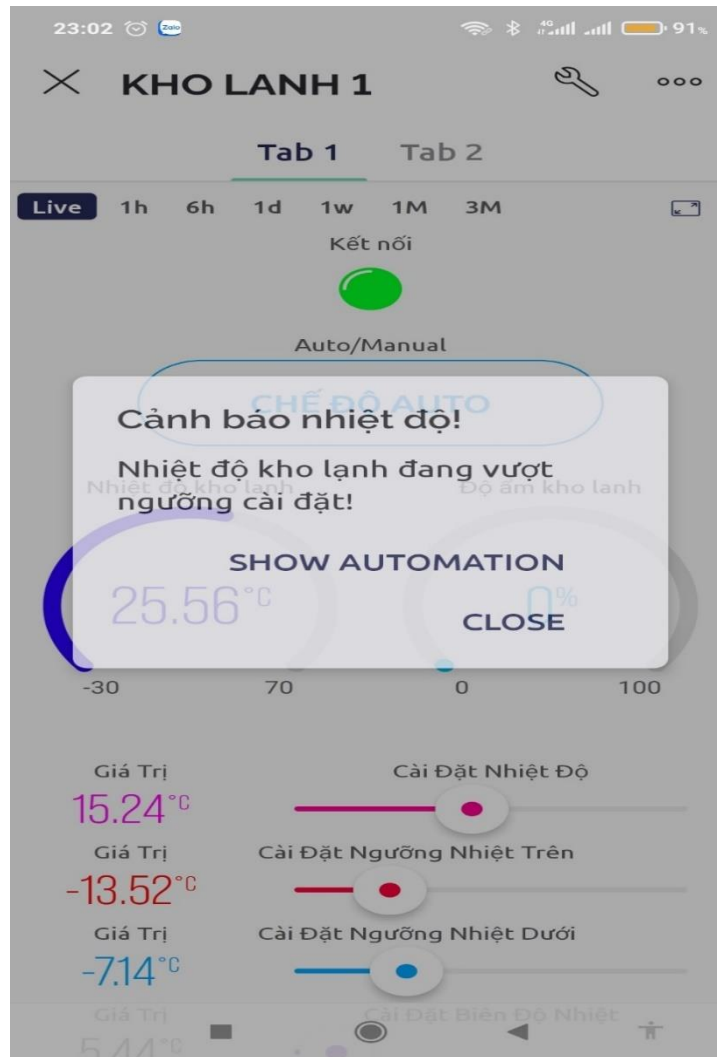
Màn hình OLED có thể hiển thị tất cả các thông số giám sát của kho lạnh và được chuyển giao diện hiển thị thông qua nút nhấn 1 ở chế độ Auto. Ngoài ra người vận hành cũng có thể cài đặt thông số và chuyển chế độ hoạt động của kho thông qua bốn nút nhấn.



Hình 5.2 Giao diện Web khi giám sát kho lạnh



Hình 5.1 Giao diện trên App điện thoại khi giám sát kho lạnh



Hình 5.2 Thông báo cảnh báo gửi về điện thoại

5.6.2. Đánh giá kết quả

Với các tình huống đặt ra mạch hoạt động đúng theo yêu cầu, tuy nhiên hai module đo dòng điện ACS712 đo kết quả không chính xác, kết quả dao động nhiều. Khi điều khiển qua Internet thỉnh thoảng nếu chất lượng WiFi không ổn định vẫn có thời gian trễ khoảng 1s nhưng nhìn chung với điều thực tế là điều khiển kho lạnh thì khoản thời gian trên là chấp nhận được.

Chưa vận hành được trong môi trường kho lạnh thực tế nên việc đánh giá hiện tại chỉ mang tính tạm thời để phát triển mạch sau này.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận:

Sau khi thực hiện xong đồ án Thiết kế mạch điện tử em đã thu được những kết quả như sau:

- ❖ Ôn tập được một khối lượng kiến thức về các linh kiện điện tử cơ bản, lập trình vi điều khiển...
- ❖ Nắm được các phương pháp thi công, thiết kế mạch điện tử, vẽ mạch in...
- ❖ Nâng cao được khả năng tìm kiếm tài liệu, soạn thảo và trình bày văn bản...
- ❖ Tìm hiểu được phương thức truyền nhận tín hiệu qua mạng internet.
- ❖ Hiểu biết được về nguyên lý và cách thức vận hành của kho lạnh
- ❖ Hoàn thành mạch giám sát và điều khiển kho lạnh từ xa với các tính năng như:
 - Giám sát và điều khiển qua mạng internet
 - Có phản hồi tín hiệu điều khiển
 - Đáp ứng được các yêu cầu của bài toán vận hành kho lạnh

Tuy nhiên trong quá trình thực hiện do thời gian thực hiện đề tài và lượng kiến thức của bản thân còn hạn chế nên không thể tránh khỏi thiếu sót mong quý thầy cô thông cảm.

Hướng phát triển cho đề tài:

Với những kết quả thu được sau khi hoàn thành đồ án và rút kinh nghiệm với những hạn chế sai sót em xin đề ra những hướng phát triển sau:

- ❖ .Có thêm tính năng giám sát hình ảnh
- ❖ Sử dụng các cảm biến tốt hơn theo chuẩn công nghiệp để mạch hoạt động ổn định, chính xác hơn
- ❖ Xây dựng các điểm thu thập dữ liệu với khoảng cách xa qua Lora, Zigbee, ...
- ❖ Kiểm tra hoạt động mạch trong một trường thực tế

TÀI LIỆU THAM KHẢO

- [1] N. H. Toàn, Luận văn thạc sỹ "Nghiên cứu giám sát nhiệt độ, độ ẩm của kho lạnh qua Internet", TP.Thái Nguyên: Đại học Thái Nguyên, 2020.
- [2] "Thietbipauviet," [Online]. Available: <https://thietbipauviet.com/nguyen-ly-hoat-dong-kho-lanh/>.
- [3] "wikipedia," [Online]. Available: <https://vi.wikipedia.org/wiki/Internet>.
- [4] "Thegioididong," [Online]. Available: <https://www.thegioididong.com/hoi-dap/toc-do-mang-wifi-nhu-the-nao-thi-moi-goi-la-nhanh-567510>.
- [5] "wikipedia," [Online]. Available: <https://vi.wikipedia.org/wiki/ESP32>.

PHỤ LỤC

Code của hệ thống:

```
#include <Arduino.h>
#define BLYNK_TEMPLATE_ID "TMPLiJ8pzhF6"
#define BLYNK_DEVICE_NAME "KHO LANH 1"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <PZEM004Tv30.h>
#include <DHT.h>
#include "ACS712.h"
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Fonts/FreeSerif9pt7b.h>

#define NutBL 2
#define NutQL 4
#define NutQN 5
#define NutXD 18
#define pinNTC 34
#define pinDHT 19
#define pinACS1 32
#define pinACS2 33
#define QuatNong 26
#define QuatLanh 27
#define XaDa 13
#define Block 14
#define Speaker 25
#define Delay_NhietDo 10000
#define Delay_XaDa 30000

#define ONE_WIRE_BUS 23
#define PZEM_RX_PIN 16
#define PZEM_TX_PIN 17
#define PZEM_SERIAL Serial2

char auth[] = "JDYJxdF_jUVa7_6rrDj9uxpQuxin1lmi";
char ssid[] = "Redmi Note 7";
char pass[] = "4815162342";

const int DHTPIN = pinDHT;
const int DHTTYPE = DHT11;
```

```

PZEM004Tv30 pzem(PZEM_SERIAL, PZEM_RX_PIN, PZEM_TX_PIN);
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
ACS712 sensor1(ACS712_30A, pinACS1);
ACS712 sensor2(ACS712_30A, pinACS2);
DHT dht(DHTPIN, DHTTYPE);
WidgetLED ledconnect(V0);

Adafruit_SSD1306 display(128, 64, &Wire, -1);
uint32_t tsLastReport = 0;
typedef enum key1
{
    HienThi1, // hien thi nhiet do, do ẩm , nhiệt độ cài đặt
    HienThi2, // hiển thị dòng điện, điện áp, tần số
    HienThi3, // hiển thị thời gian xả đá và nhiệt độ cài đặt
    HienThi4, // hien thi cai dat
    HienThi5, // hiển thị trạng thái sự cố
    HienThi6, // Hien thi thong so nguon dien
} KEY1;
KEY1 _key;

bool CanhBaoTong = 0;
bool CanhBaoQN = 0;
bool CanhBaoBL = 0;
bool CanhBaoQuaNhiet = 0;
bool CanhBaoHaNhiet = 0;
bool CanhBaoWifi = 0;

bool Status_XaDa = 0;
bool Status_QL = 0;
bool Status_QN = 0;
bool Status_BL = 0;
bool cheDo = 0; // 0=auto 1=man

int Set_NguongBaoVeBL;
int Set_NguongBaoVeQN;
float Set_NhietDoMax;
float Set_NhietDoMin;
int Set_NhietDoXaDa;
float Set_ThoiGianXaDa;
int Set_NguongBaoVeHT;
float Set_NhietDo;
float Set_DaoDongNhiet;

unsigned long time1;
unsigned long time2;

```

```

unsigned long time_delay;
unsigned long time_delay1;
unsigned long time_delay2;

float DongMayNen;
float DongQuatNong;
float DongHeThong;
float Power;
float dienAp;
float HZ;
float PF; // Cos phi
float NhietDoKho;
float NhietDoDanLanh;
float DoAm;
float ThoiGianConLai;
bool i = 0;          // khóa xả đá
bool i1 = 0;         // khóa điều khiển nhiệt độ
bool i2 = 0;         // khóa thời gian xả đá
bool CaiDatThongSo; // 0=setNhietDo ; 1=SetThoiGianXaDa
int dem;             // dem thi tư giao dien hien thi oled

//===== DOC CAM BIEN /=====
float readNTC(float ADC);
float readDS18();
float readACS1();
float readACS2();
float readDHT();
float readVoltage();
float readCurrent();
float readPower();
float readEnergy();
float readHZ();
float readPF();
float thoiGianXaDaConLai(float times);
void suCo();
void kiemTraNutNhan();
void docCamBien();
void guiDuLieu();
void setupOled();
void hienThiOled();
void hienThi1();
void hienThi2();
void hienThi3();
void hienThi4();
void hienThi5();
void hienThi6();
void reConnectWF();

```

```

void baoVe();
void batSpeaker();
void canhBaoNhietDo();

//=====DIEU KHIEN =====

void DKOutput();
void DKXaDa();
void DKNhietDo();
//=====
void setup()
{
    pinMode(NutBL, INPUT);
    pinMode(NutQL, INPUT);
    pinMode(NutQN, INPUT);
    pinMode(NutXD, INPUT);
    pinMode(pinNTC, INPUT);
    pinMode(pinDHT, INPUT);
    pinMode(pinACS1, INPUT);
    pinMode(pinACS2, INPUT);
    pinMode(QuatLanh, OUTPUT);
    pinMode(QuatNong, OUTPUT);
    pinMode(XaDa, OUTPUT);
    pinMode(Block, OUTPUT);
    pinMode(Speaker, OUTPUT);
    digitalWrite(QuatLanh, 0);
    digitalWrite(QuatNong, 0);
    digitalWrite(XaDa, 0);
    digitalWrite(Speaker, 0);
    digitalWrite(Block, 0);
    sensors.begin();
    Serial.begin(9600);
    setupOled();
    Serial.print("Connecting to wifi");
    Serial.println(ssid);
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    Blynk.begin(auth, ssid, pass);
}

```

```

    DKOutput();
    delay(5000);
    time1 = millis();
    time2 = millis();
}

void loop()
{
    Blynk.run();
    if (millis() - time1 > 1000)
    {
        if (ledconnect.getValue())
        {
            ledconnect.off();
        }
        else
        {
            ledconnect.on();
        }

        if (cheDo == 0)
        {
            if (Status_XaDa == 1)
            {
                Blynk.virtualWrite(V22, thoiGianXaDaConLai(millis() - time_delay2));
            }
        }
        time1 = millis();
    }

    if (millis() - time2 > 1000)
    {
        guiDuLieu();
        hienThiOled();
    }
    hienThiOled();
   kiemTraNutNhan();
    suCo();
    canhBaoNhietDo();
    batSpeaker();
   kiemTraNutNhan();
    baoVe();
    DKNhietDo();
    DKXaDa();
   kiemTraNutNhan();
    reConnectWF();
}

```

```

float readNTC(float ADC)
{
    float temp = ADC * (3.3 /
4094.0); // = V2
    temp = (((3.3 * 10000.0) / temp) -
10000.0); // = R_ntc
    temp = ((1.0 / (25.5 + 273.15)) + ((1.0 / 3950.0) * log(temp / 10000.0)));
// 1/T=1/T0 + (1/B)*ln(R_ntc/R2) (1/K)
    temp = ((1.0 / temp)) -
273.15; // đổi độ K sang C
    return temp;
}

float readDS18()
{
    sensors.requestTemperatures();
    // Serial.println(sensors.getTempCByIndex(0));
    float A;
    A = sensors.getTempCByIndex(0);
    return A;
}

float readVoltage()
{
    float voltage = pzem.voltage();
    return voltage;
}

float readCurrent()
{
    float current = pzem.current();
    return current;
}

float readPower()
{
    float power = pzem.power();
    return power;
}

float readEnergy()
{
    float energy = pzem.energy();
    return energy;
}

float readHZ()
{
    float frequency = pzem.frequency();
    return frequency;
}

```



```

float readPF()
{
    float pf = pzem.pf();
    return pf;
}

float readDHT()
{
    float h = dht.readHumidity();
    return h;
}

float readACS1()
{
    float I = sensor1.getCurrentAC();
    return I;
}
float readACS2()
{
    float I = sensor2.getCurrentAC();
    return I;
}

void kiemTraNutNhan()
{
    static unsigned long time_delay3;
    static bool i3 = 0; // khóa tine_delay3 chon che do man
    if (cheDo == 1)
    {
        if (digitalRead(NutBL) == 0)
        {
            delay(5);
            if (digitalRead(NutBL) == 0)
            {
                while (digitalRead(NutBL) == 0)
                ;
                Status_BL = !Status_BL;
                digitalWrite(Block, Status_BL);
                Blynk.virtualWrite(V1, Status_BL);
                // Serial.println("Nut BL= " + String(Status_BL));
            }
        }
        if (digitalRead(NutQN) == 0)
        {

```

```

delay(5);
if (digitalRead(NutQN) == 0)
{
    while (digitalRead(NutQN) == 0)
        ;
    Status_QN = !Status_QN;
    digitalWrite(QuatNong, Status_QN);
    Blynk.virtualWrite(V2, Status_QN);
    // Serial.println("Nut QN= " + String(Status_QN));
}
}
if (digitalRead(NutQL) == 0)
{
    delay(5);
    if (digitalRead(NutQL) == 0)
    {
        while (digitalRead(NutQL) == 0)
            ;
        Status_QL = !Status_QL;
        digitalWrite(QuatLanh, Status_QL);
        Blynk.virtualWrite(V3, Status_QL);
        // Serial.println("Nut QL= " + String(Status_QL));
    }
}
if (digitalRead(NutXD) == 0)
{
    delay(5);
    if (digitalRead(NutXD) == 0)
    {
        delay(100);
        while (digitalRead(NutXD) == 0)
        {
            if (i3 == 0)
            {
                time_delay3 = millis();
                i3 = 1;
            }
            if (millis() - time_delay3 > 4000)
            {
                i3 = 0;
                cheDo = 0;
                Blynk.virtualWrite(V23, cheDo);
                digitalWrite(Speaker, 1);
                delay(600);
                digitalWrite(Speaker, 0);
                goto jum1;
            }
        }
    }
}

```

```

    };
    i3 = 0;
    Status_XaDa = !Status_XaDa;
    digitalWrite(XaDa, Status_XaDa);
    Blynk.virtualWrite(V4, Status_XaDa);
jum1:
    delay(1);
}
}
}
else if (cheDo == 0)
{
    if (digitalRead(NutBL) == 0)
    {
        delay(5);
        if (digitalRead(NutBL) == 0)
        {
            while (digitalRead(NutBL) == 0)
                ;

            dem++;
            if (dem > 4)
            {
                dem = 0;
            }
            // Serial.println("Hien thi = " + String(dem));
            if (dem == 0)
            {
                _key = HienThi1;
            }
            else if (dem == 1)
            {
                _key = HienThi2;
            }
            else if (dem == 2)
            {
                _key = HienThi3;
            }
            else if (dem == 3)
            {
                _key = HienThi6;
            }
            hienThiOled();
        }
    }
}

if (digitalRead(NutQL) == 0)
{

```

```

if (digitalRead(NutQL) == 0)
{
    while (digitalRead(NutQL) == 0)
        ;
    if (CaiDatThongSo == 0)
    {
        Set_NhietDo = Set_NhietDo + 1.0;
        Blynk.virtualWrite(V20, Set_NhietDo);
        // Serial.println("set nhiet do + = ");
    }
    else if (CaiDatThongSo == 1)
    {
        // Serial.println("set thoi gian = " + String(Set_ThoiGianXaDa));
        Set_ThoiGianXaDa = Set_ThoiGianXaDa + 60000;
        Set_ThoiGianXaDa = Set_ThoiGianXaDa / 1000.0 / 60.0;
        Blynk.virtualWrite(V15, Set_ThoiGianXaDa);
        // Serial.println("set thoi gian + = " + String(Set_ThoiGianXaDa));
        Set_ThoiGianXaDa = Set_ThoiGianXaDa * 1000.0 * 60.0;
    }
    hienThiOled();
}
}

if (digitalRead(NutQN) == 0)
{
    delay(5);
    if (digitalRead(NutQN) == 0)
    {
        while (digitalRead(NutQN) == 0)
            ;
        if (CaiDatThongSo == 0)
        {
            Set_NhietDo = Set_NhietDo - 1.0;
            Blynk.virtualWrite(V20, Set_NhietDo);
            // Serial.println("set nhiet do - = ");
        }
        else if (CaiDatThongSo == 1)
        {
            // Serial.println("set thoi gian = " + String(Set_ThoiGianXaDa));
            Set_ThoiGianXaDa = Set_ThoiGianXaDa - 60000;
            Set_ThoiGianXaDa = Set_ThoiGianXaDa / 1000.0 / 60.0;
            Blynk.virtualWrite(V15, Set_ThoiGianXaDa);
            // Serial.println("set thoi gian - = " + String(Set_ThoiGianXaDa));
            Set_ThoiGianXaDa = Set_ThoiGianXaDa * 1000.0 * 60.0;
        }
        hienThiOled();
    }
}

```

```

    }

    if (digitalRead(NutXD) == 0)
    {
        delay(5);
        if (digitalRead(NutXD) == 0)
        {
            while (digitalRead(NutXD) == 0)
            {
                if (i3 == 0)
                {
                    time_delay3 = millis();
                    i3 = 1;
                }
                if (millis() - time_delay3 > 4000)
                {
                    i3 = 0;
                    cheDo = 1;
                    Blynk.virtualWrite(V23, cheDo);
                    // Serial.println("vao che do man");
                    digitalWrite(Speaker, 1);
                    delay(600);
                    digitalWrite(Speaker, 0);
                    goto jum;
                }
            }
            i3 = 0;
            CaiDatThongSo = !CaiDatThongSo; // 1 vào cài đặt thời gian xa đá
            _key = HienThi4;
            dem = 4;
            hienThiOled();

            digitalWrite(Speaker, 1);
            delay(600);
            digitalWrite(Speaker, 0);
            Serial.println("cai dat thong so=" + String(CaiDatThongSo));
            jum:
            delay(1);
        }
    }
}

void DKOutput()
{
    batSpeaker();
    baoVe();
    suCo();
}

```

```

    Status_XaDa = 0;
    Status_QN = 1;
    Status_QL = 1;
    Status_BL = 1;
    digitalWrite(XaDa, Status_XaDa);
    Blynk.virtualWrite(V4, Status_XaDa);
    digitalWrite(QuatNong, Status_QN);
    Blynk.virtualWrite(V2, Status_QN);
    delay(1500);
    digitalWrite(QuatLanh, Status_QL);
    Blynk.virtualWrite(V3, Status_QL);
    delay(1500);
    digitalWrite(Block, Status_BL);
    Blynk.virtualWrite(V1, Status_BL);
    Serial.println("vao DK");
    delay(1500);
}
void DKXaDa()
{
    if (cheDo == 0)
    {
        if ((NhietDoKho - NhietDoDanLanh) > Set_NhietDoXaDa)
        {
            if (i == 0)
            {
                time_delay1 = millis();
                i = 1;
            }
            if (millis() - time_delay1 > Delay_XaDa)
            {
                if ((NhietDoKho - NhietDoDanLanh) > Set_NhietDoXaDa)
                {
                    i = 0;
                    Status_XaDa = 1;
                    Status_QN = 1;
                    Status_QL = 0;
                    Status_BL = 0;
                    digitalWrite(Block, Status_BL);
                    Blynk.virtualWrite(V1, Status_BL);
                    delay(1500);
                    digitalWrite(QuatLanh, Status_QL);
                    Blynk.virtualWrite(V3, Status_QL);
                    delay(1500);
                    digitalWrite(XaDa, Status_XaDa);
                    Blynk.virtualWrite(V4, Status_XaDa);
                    digitalWrite(QuatNong, Status_QN);
                    Blynk.virtualWrite(V2, Status_QN);

```

```

        if (i2 == 0)
        {
            i2 = 1;
            time_delay2 = millis();
            Blynk.virtualWrite(V22, thoiGianXaDaConLai(millis() -
time_delay2));
            if (cheDo == 0)
            {
                if (Status_XaDa == 1)
                {
                    Blynk.virtualWrite(V22, thoiGianXaDaConLai(millis() -
time_delay2));
                }
            }
        }
        else
        {
            i = 0;
        }
    }
}
else if (Status_XaDa == 1)
{
    if (millis() - time_delay2 > Set_ThoiGianXaDa)
    {
        Blynk.virtualWrite(V22, thoiGianXaDaConLai(millis() - time_delay2));
        i2 = 0;
        Status_XaDa = 0;
        Status_QL = 1;
        Status_BL = 1;
        Status_QN = 1;
        digitalWrite(XaDa, Status_XaDa);
        Blynk.virtualWrite(V4, Status_XaDa);
        delay(1500);
        digitalWrite(QuatLanh, Status_QL);
        Blynk.virtualWrite(V3, Status_QL);
        digitalWrite(QuatNong, Status_QN);
        Blynk.virtualWrite(V2, Status_QN);
        delay(1500);
        digitalWrite(Block, Status_BL);
        Blynk.virtualWrite(V1, Status_BL);
    }
}
}
}
void suCo()

```

```

{
    if (readACS1() > Set_NguongBaoVeBL)
    {
        CanhBaoBL = 1;
        dem = 4;
        _key = HienThi5;
    }
    else
    {
        CanhBaoBL = 0;
    }

    if (readACS2() > Set_NguongBaoVeQN)
    {
        CanhBaoQN = 1;
        dem = 4;
        _key = HienThi5;
    }
    else
    {
        CanhBaoQN = 0;
    }

    if (pzem.current() > Set_NguongBaoVeHT)
    {
        CanhBaoTong = 1;
        dem = 4;
        _key = HienThi5;
    }
    else
    {
        CanhBaoTong = 0;
    }
    if (Status_BL == 1)
    {
        if (DongMayNen <= 0)
        {
            CanhBaoBL = 1;
            dem = 4;
            _key = HienThi5;
        }
    }
    if (Status_QN == 1)
    {
        if (DongQuatNong <= 0)
        {
            CanhBaoQN = 1;

```



```

        dem = 4;
        _key = HienThi5;
    }
}
hienThiOled();
if (CanhBaoTong == 1 || CanhBaoBL == 1 || CanhBaoQN == 1)
{
    Blynk.virtualWrite(V11, 1);
}
else
{
    Blynk.virtualWrite(V11, 0);
}
}
void baoVe()
{
    if (CanhBaoTong == 1)
    {
        Status_BL = 0;
        Status_QN = 0;
        Status_QL = 0;
        Status_XaDa = 0;
        digitalWrite(Block, Status_BL);
        Blynk.virtualWrite(V1, Status_BL);
        digitalWrite(QuatNong, Status_QN);
        Blynk.virtualWrite(V2, Status_QN);
        digitalWrite(QuatLanh, Status_QL);
        Blynk.virtualWrite(V3, Status_QL);
        digitalWrite(XaDa, Status_XaDa);
        Blynk.virtualWrite(V4, Status_XaDa);
    }
    else if (CanhBaoBL == 1)
    {
        Status_BL = 0;
        digitalWrite(Block, Status_BL);
        Blynk.virtualWrite(V1, Status_BL);
    }
    else if (CanhBaoQN == 1)
    {
        Status_QN = 0;
        digitalWrite(QuatNong, Status_QN);
        Blynk.virtualWrite(V2, Status_QN);
    }
}

void docCamBien()
{

```

```

    kiemTraNutNhan();
    DongHeThong = readCurrent();
    DongMayNen = readACS1();
    DongQuatNong = readACS2();
    NhietDoKho = readDS18();
    Serial.println(NhietDoKho);
    NhietDoDanLanh = readNTC(analogRead(pinNTC));
    Power = readPower();
    DoAm = readDHT();
    dienAp = readVoltage();
    HZ = readHZ();
    PF=readPF();
    kiemTraNutNhan();
}
void guiDuLieu()
{
    docCamBien();
    Blynk.virtualWrite(V5, NhietDoKho);
    Blynk.virtualWrite(V7, DoAm);
    Blynk.virtualWrite(V12, DongHeThong);
    kiemTraNutNhan();

    delay(5);
    Blynk.virtualWrite(V6, NhietDoDanLanh);
    Blynk.virtualWrite(V8, DongMayNen);
    Blynk.virtualWrite(V9, DongQuatNong);
    kiemTraNutNhan();
    delay(5);
    Blynk.virtualWrite(V10, Power);
    Blynk.virtualWrite(V24, HZ);
    Blynk.virtualWrite(V25, dienAp);
    kiemTraNutNhan();
}
void canhBaoNhietDo()
{
    if (NhietDoKho < Set_NhietDoMin)
    {
        CanhBaoHaNhiet = 1;
        dem = 4;
        _key = HienThi5;
    }
    else if (NhietDoKho > Set_NhietDoMax)
    {
        CanhBaoQuaNhiet = 1;

        dem = 4;
        _key = HienThi5;
    }
}

```

```

    }
    else if ((NhietDoKho >= Set_NhietDoMin) && (NhietDoKho <= Set_NhietDoMax))
    {
        CanhBaoQuaNhiet = 0;
        CanhBaoHaNhiet = 0;
    }
    Blynk.virtualWrite(V26, CanhBaoQuaNhiet);
    Blynk.virtualWrite(V27, CanhBaoHaNhiet);
}
void DKNhietDo()
{
    if (cheDo == 0)
    {
        if (Status_XaDa == 0)
        {
            if (Status_BL == 0)
            {
                if (NhietDoKho > (Set_NhietDo + Set_DaoDongNhiet))
                {
                    if (i1 == 0)
                    {
                        time_delay = millis();
                        i1 = 1;
                    }

                    if (millis() - time_delay > Delay_NhietDo)
                    {
                        if (NhietDoKho > (Set_NhietDo + Set_DaoDongNhiet))
                        {
                            Status_BL = 1;
                            Status_QL = 1;
                            Status_QN = 1;
                            Serial.println("vao 2");
                            digitalWrite(QuatNong, Status_QN);
                            Blynk.virtualWrite(V2, Status_QN);
                            delay(500);
                            digitalWrite(QuatLanh, Status_QL);
                            Blynk.virtualWrite(V3, Status_QL);
                            delay(1000);
                            digitalWrite(Block, Status_BL);
                            Blynk.virtualWrite(V1, Status_BL);
                            i1 = 0;
                        }
                    }
                    else
                    {
                        i1 = 0;
                    }
                }
            }
        }
    }
}

```

```

    }
  }
}
else if (Status_BL == 1)
{
    if (NhietDoKho < (Set_NhietDo - Set_DaoDongNhiet))
    {
        if (i1 == 0)
        {
            time_delay = millis();
            i1 = 1;
        }

        if (millis() - time_delay > Delay_NhietDo)
        {
            if (NhietDoKho < (Set_NhietDo - Set_DaoDongNhiet))
            {
                Status_BL = 0;
                Status_QL = 0;
                Status_QN = 1;
                digitalWrite(QuatNong, Status_QN);
                Blynk.virtualWrite(V2, Status_QN);
                delay(500);
                digitalWrite(QuatLanh, Status_QL);
                Blynk.virtualWrite(V3, Status_QL);
                delay(1000);
                digitalWrite(Block, Status_BL);
                Blynk.virtualWrite(V1, Status_BL);
                i1 = 0;
            }
            else
            {
                i1 = 0;
            }
        }
    }
}
}
}

void batSpeaker()
{
    if (CanhBaoBL == 1 || CanhBaoQN == 1 || CanhBaoTong == 1 || CanhBaoQuaNhiet
    == 1 || CanhBaoHaNhiet == 1 || CanhBaoWifi == 1)
    {

```

```

        digitalWrite(Speaker, 1);
    }
    else
    {
        digitalWrite(Speaker, 0);
    }
}

float thoiGianXaDaConLai(float times)
{
    float ThoiGianConLai;
    ThoiGianConLai = (Set_ThoiGianXaDa - times) / 1000.0 / 60.0;
    return ThoiGianConLai;
}

void setupOled()
{
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
    {
        Serial.println("SSD1306 allocation failed");
        for (;;)
            ;
    }
    display.setFont(&FreeSerif9pt7b);
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(3, 15);
    display.println("DIEN-DIEN TU");
    display.setCursor(25, 38);
    display.println("61.DDT-2");
    display.setCursor(7, 58);
    display.println("Phan Van Hieu");
    display.display();
    display.setTextSize(1);
}

void hienThiOled()
{
    switch (_key)
    {
        case HienThi1:
            hienThi1();
            break;
        case HienThi2:
            hienThi2();
            break;
        case HienThi3:

```

```

        hienThi3();
        break;
    case HienThi4:
        hienThi4();
        break;
    case HienThi5:
        hienThi5();
        break;
        case HienThi6:
            hienThi6();
            break;
    }
}
void hienThi1()
{
    display.clearDisplay();
    display.setCursor(3, 15);
    display.println("DIEN-DIEN TU");
    display.setCursor(0, 38);
    display.print("T=");
    display.print(NhietDoKho, 1);
    display.setCursor(64, 38);
    display.print("H=");
    display.print(DoAm, 1);
    display.setCursor(0, 59);
    display.print("Tcd:");
    display.print(Set_NhietDo, 0);
    display.setCursor(64, 59);
    display.print("Tl=");
    display.print(NhietDoDanLanh, 1);
    display.display();
}
void hienThi2()
{
    display.clearDisplay();
    display.setCursor(3, 15);
    display.println("DIEN-DIEN TU");
    display.setCursor(0, 38);
    display.print("I1=");
    display.print(DongHeThong, 2);
    display.setCursor(64, 38);
    display.print("P=");
    display.print(Power, 2);
    display.setCursor(0, 59);
    display.print("I2=");
    display.print(DongQuatNong, 1);
    display.setCursor(64, 59);

```

```

        display.print("I3=");
        display.print(DongMayNen,1);
        display.display();
    }
    void hienThi3()
    {
        display.clearDisplay();
        display.setCursor(3, 15);
        display.println("DIEN-DIEN TU");
        display.setCursor(0, 38);
        display.print("V=");
        display.print(dienAp,0);
        display.setCursor(64, 38);
        display.print("H=");
        display.print(HZ,1);
        display.setCursor(0, 59);
        display.print("Cos=");
        display.print(PF,2);
        display.display();
    }
    void hienThi4()
    {
        display.clearDisplay();
        display.setCursor(3, 15);
        display.println("DIEN-DIEN TU");
        if (CaiDatThongSo == 0)
        {
            display.setCursor(5, 38);
            display.print("SET NHIET DO");
            display.setCursor(33, 59);
            display.print("Tcd=");
            display.print(Set_NhietDo, 1);
        }
        else if (CaiDatThongSo == 1)
        {
            display.setCursor(12, 38);
            display.print("TIME XA DA");
            display.setCursor(33, 59);
            display.print("Txd=");
            display.print(Set_ThoiGianXaDa / (60.0 * 1000.0), 0);
        }

        display.display();
    }
    void hienThi5()
    {
        display.clearDisplay();

```

```

display.setCursor(3, 15);
display.println("DIEN-DIEN TU");
if (CanhBaoBL == 1)
{
    display.setCursor(20, 38);
    display.print("CANH BAO");
    display.setCursor(8, 59);
    display.print("LOI MAY NEN!");
}
else if (CanhBaoQN == 1)
{
    display.setCursor(20, 38);
    display.print("CANH BAO");
    display.setCursor(0, 59);
    display.print("LOI DAN LANH!");
}
else if (CanhBaoQuaNhiet == 1)
{
    display.setCursor(20, 38);
    display.print("NHIET DO");
    display.setCursor(5, 59);
    display.print("QUA NGUONG!");
}
else if (CanhBaoHaNhiet == 1)
{
    display.setCursor(20, 38);
    display.print("NHIET DO");
    display.setCursor(3, 59);
    display.print("DUOI NGUONG!");
}
else if (CanhBaoWifi == 1)
{
    display.setCursor(35, 38);
    display.println("Loi WiFI");
    display.setCursor(0, 58);
    display.println("Dang ket noi....");
}

display.display();
}
void hienThi6()
{
    display.clearDisplay();
    display.setCursor(3, 15);
    display.println("DIEN-DIEN TU");
    display.setCursor(34, 38);
    display.print("Tcd=");
}

```



```

display.print(Set_NhietDo, 0);
display.setCursor(34, 59);
display.print("Txd=");
display.print(Set_ThoiGianXaDa / (60.0 * 1000.0), 0);
display.display();
}
void reConnectWF()
{
    if (WiFi.status() == WL_DISCONNECTED)
    {
        dem = 4;
        _key = HienThi5;
        CanhBaoWifi = 1;
        WiFi.begin(ssid, pass);
        while (WiFi.status() != WL_CONNECTED)
        {
            hienThiOled();
            kiemTraNutNhan();
            docCamBien();
            suCo();
            batSpeaker();
            canhBaoNhietDo();
        }

        Serial.println(WiFi.localIP());
        Blynk.begin(auth, ssid, pass);
        delay(100);
        if (WiFi.status() == WL_CONNECTED)
        {
            display.clearDisplay();
            display.setCursor(3, 15);
            display.println("DIEN-DIEN TU");
            display.setCursor(40, 38);
            display.println("WiFi");
            display.setCursor(20, 58);
            display.println("Da ket noi");
            display.display();
            CanhBaoWifi = 0;
            delay(3000);
        }
    }
}
BLYNK_CONNECTED()
{
    Blynk.syncVirtual(V1);
    Blynk.syncVirtual(V2);
}

```

```

    Blynk.syncVirtual(V3);
    Blynk.syncVirtual(V4);
    Blynk.syncVirtual(V13);
    Blynk.syncVirtual(V14);
    Blynk.syncVirtual(V15);
    Blynk.syncVirtual(V16);
    Blynk.syncVirtual(V17);
    Blynk.syncVirtual(V18);
    Blynk.syncVirtual(V19);
    Blynk.syncVirtual(V20);
    Blynk.syncVirtual(V21);
    Blynk.syncVirtual(V23);
}
BLYNK_WRITE(V1)
{
    if (cheDo == 1)
    {
        Status_BL = param.asInt();
        digitalWrite(Block, Status_BL);
        Serial.println("vao V1");
        // kiemTraNutNhan();
    }
    else
    {
        Status_BL = param.asInt();
        Status_BL = !Status_BL;
        Serial.println("vao elsse");
        Blynk.virtualWrite(V1, Status_BL);
    }
}
BLYNK_WRITE(V2)
{
    if (cheDo == 1)
    {
        Status_QN = param.asInt();
        digitalWrite(QuatNong, Status_QN);
    }
    else
    {
        Status_QN = param.asInt();
        Status_QN = !Status_QN;
        Blynk.virtualWrite(V2, Status_QN);
    }
}
BLYNK_WRITE(V3)
{
    if (cheDo == 1)

```

```

    {
        Status_QL = param.asInt();
        digitalWrite(QuatLanh, Status_QL);
    }
    else
    {
        Status_QL = param.asInt();
        Status_QL = !Status_QL;
        Blynk.virtualWrite(V3, Status_QL);
    }
}
BLYNK_WRITE(V4)
{
    if (cheDo == 1)
    {
        Status_XaDa = param.asInt();
        digitalWrite(XaDa, Status_XaDa);
    }
    else
    {
        Status_XaDa = param.asInt();
        Status_XaDa = !Status_XaDa;
        Blynk.virtualWrite(V4, Status_XaDa);
    }
}
BLYNK_WRITE(V13)
{
    Set_NhietDoMax = param.asFloat();
}
BLYNK_WRITE(V14)
{
    Set_NhietDoMin = param.asFloat();
}
BLYNK_WRITE(V15)
{
    Set_ThoiGianXaDa = param.asFloat();
    Set_ThoiGianXaDa = (Set_ThoiGianXaDa * 60.0 * 1000.0);
}
BLYNK_WRITE(V16)
{
    Set_NhietDoXaDa = param.asFloat();
}
BLYNK_WRITE(V17)
{
    Set_NguongBaoVeQN = param.asInt();
}
BLYNK_WRITE(V18)

```

```

{
    Set_NguongBaoVeBL = param.asInt();
}
BLYNK_WRITE(V19)
{
    Set_NguongBaoVeHT = param.asInt();
}
BLYNK_WRITE(V20)
{
    Set_NhietDo = param.asFloat();
}
BLYNK_WRITE(V21)
{
    Set_DaoDongNhiet = param.asFloat();
}
BLYNK_WRITE(V23)
{
    cheDo = param.asInt();
}

```