MỤC LỤC

DANH MỤC CÁC BẢNG....3
DANH MỤC CÁC HÌNH ẢNH VÀ ĐỒ THỊ....4
I. PHẦN MỞ ĐẦU....6
1. Tính cấp thiết thực hiện nhiệm vụ....6
2. Tổng quan tình hình nghiên cứu....8
3. Mục tiêu nghiên cứu....10
4. Đối tượng và phạm vi nghiên cứu....10
II. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU....11
1. Nội dung nghiên cứu....11
1.1. Nghiên cứu tổng quan, cơ sở lý thuyết....11
1.2. Tính toán, thiết kế và xây dựng mô hình thực nghiệm....11
2. Vật liệu và phương pháp nghiên cứu....11
2.1. Vật liệu sử dụng trong nghiên cứu....11
2.2. Phương pháp nghiên cứu....12
III. KẾT QUẢ NGHIÊN CỨU VÀ THẢO LUẬN....13
1. KẾT QUẢ NGHIÊN CỨU....13
1.1. NGUYÊN LÝ HOẠT ĐỘNG CỦA MẠNG ĐIỆN THOẠI....13
1.1.1. Tổng đài....13
1.1.1.1. Khái niệm về tổng đài:....13
1.1.1.2. Chức năng của tổng đài:....13
1.1.1.3. Phân loại tổng đài:....13
1.1.1.4. Các loại âm hiệu:....13
1.1.1.5. Phương thức chuyển mạch của tổng đài điện tử:....15
1.1.2. Máy điện thoại....16
1.1.2.1. Các thông số cơ bản của máy điện thoại:....16
1.1.2.2. Các hoạt động trên mạng của máy điện thoại:....16
1.1.3. Phương thức hoạt động giữa tổng đài và máy điện thoại....18
1.2. HỆ THỐNG BÁO CHÁY TỰ ĐỘNG VÀ GIẢI PHÁP TÍCH HỢP....20
1.2.1. Nhiệm vụ của hệ thống báo cháy tự động....20
1.2.2. Nguyên lý làm việc hệ thống báo cháy tự động....21
1.2.3. Ứng dụng mạng viễn thông trong hệ thống báo cháy tự động....21
1.2.4. Ứng dụng vi mạch tích hợp trong hệ thống báo cháy tự động....22
1.2.5. Giải pháp tích hợp hệ thống báo cháy tự động....27

1.3. THIẾT KẾ VÀ LẬP TRÌNH HỆ THỐNG.....30
1.3.1. Tính toán hệ thống.....30
1.3.1.1. Khối cảm biến đầu vào.....30
1.3.1.2. Khối kết nối thuê bao để gửi tin nhắn, gọi điện.....34
1.3.1.3. Khối giải mã, hiển thị và giao tiếp phím nhấn.....36
1.3.1.4. Khối điều khiển động lực đóng ngắt tiếp điểm khô.....38
1.3.1.5. Khối điều khiển ngắt điện thiết bị cần được bảo vệ.....39
1.3.1.6. Khối cảnh báo tại chỗ bằng tín hiệu đèn và âm thanh.....40
1.3.1.7. Khối kết nối với hệ thống báo cháy có sẵn.....41
1.3.1.8. Khối giao tiếp phần mềm giám sát.....41
1.3.1.9. Khối backend xử lý dữ liệu trên cloud.....44
1.3.1.10. Khối frontend phần mềm giám sát chạy trên điện thoại thông minh….44
1.3.1.11. Khối xử lý trung tâm.....45
1.3.2. LẬP TRÌNH HỆ THỐNG.....49
1.3.2.1. Lập trình bộ xử lý trung tâm.....49
1.3.2.2. Xây dựng hệ thống Backend.....99
1.3.2.2.1. Cơ sở dữ liệu của Backend.....99
1.3.2.2.2. Giao thức tương tác với Backend.....101
1.3.2.2.3. Các thuật toán xử lý và chương trình điều khiển của Backend.....102
1.3.2.3. Xây dựng hệ thống fontend.....106
1.3.2.3.1. Các chức năng chính.....106
1.3.2.3.2. Giao diện.....120
1.3.3. GHÉP NỐI, THỬ NGHIỆM VÀ HOÀN THIỆN HỆ THỐNG.....136
1.3.3.1. Ghép nối hệ thống.....136
1.3.3.2. Vận hành thử nghiệm.....137
1.3.3.3. Sản phẩm hoàn chỉnh sau hiệu chỉnh.....139
2. THẢO LUẬN.....140
2.1. Thảo luận 1.....140
2.2. Thảo luận 2.....141
IV. KẾT LUẬN VÀ KIẾN NGHỊ.....144
1. Kết luận.....144
2. Đề xuất, kiến nghị.....145
DANH MỤC TÀI LIỆU THAM KHẢO.....146

DANH MỤC CÁC BẢNG

Bảng 1: Bảng phân loại tần số tín hiệu Tone.....17
Bảng 2: So sánh một số vi điều khiển họ AVR.....24
Bảng 3: Bảng xác thực người dùng trên backend.....100
Bảng 4: Bảng dữ liệu trạng thái thiết bị trên backend.....100
Bảng 5: Bảng thông tin điều khiển thiết bị trên backend.....101
Bảng 6: Module chương trình xử lý cập nhật trạng thái thiết bị.....102
Bảng 7: Module chương trình kiểm tra lệnh điều khiển.....104
Bảng 8: Module chương trình đọc trạng thái thiết bị.....105
Bảng 9: Module chương trình gửi lệnh điều khiển.....105
Bảng 10: Module chương trình đọc trang thái thiết bị từ backend.....107
Bảng 11: Chương trình hiển thị trạng thái thiết bị trên giao diện frontend.....110
Bảng 12: Thiết kế giao diện hiển thị của frontend.....120

I. DANH MỤC CÁC HÌNH ẢNH VÀ ĐỒ THỊ

Hình 1: Tín hiệu Dial tone.....14
Hình 2: Tín hiệu Busy Tone.....14
Hình 3: Tín hiệu Ring Tone.....14
Hình 4: Tín hiệu Ring back Tone.....15
Hình 5: Quay số kiểu dạng Pulse.....17
Hình 6: Sơ đồ chân vi điều khiển Atmega2560.....23
Hình 7: Sơ đồ khối vi điều khiển ATmega2560.....25
Hình 8: Hình ảnh thực của ESP8266.....26
Hình 9: Sơ đồ chân của ESP8266.....27
Hình 10: Sơ đồ khối của ESP8266.....27
Hình 11: Sơ đồ khối hệ thống báo cháy tự động.....28
Hình 12: Đầu báo nhiệt AHR-871.....31
Hình 13: Đầu báo khói AH-0311-2.....32
Hình 14: Sơ đồ mạch xử lý khối cảm biến đầu vào.....34
Hình 15: Module GSM SIM800L.....34
Hình 16: Mạch giao tiếp module GSM SIM800L.....35
Hình 17: LCD 20x4 tích hợp đèn nền.....36
Hình 18: Mạch chuyển đổi I2C cho LCD 20x4.....37
Hình 19: Sơ đồ mạch ghép nối LCD 20x4 với bộ xử lý trung thâm.....38
Hình 20: Sơ đồ mạch ghép nối bàn phím với bộ xử lý trung thâm.....38
Hình 21: Mạch xử lý đầu vào tiếp điểm khô.....39
Hình 22: Khối ngắt điện nguồn cần bảo vệ.....40
Hình 23: Mạch điều khiển rơ le trung gian tới khởi động từ.....40
Hình 24: Sơ đồ đấu nối các hệ thống báo cháy hiện có.....41
Hình 25: Module ESP8266 trang bị kết nối mạng Wifi.....42
Hình 26: Module kết nối mạng Ethernet.....43
Hình 27: Sơ đồ mạch module kết nối mạng Ethernet.....44
Hình 28: Sơ đồ ghép nối fontend với backend và hệ thống báo cháy tự động.....45
Hình 29: Chip ATmega 2560.....46
Hình 30: Sơ đồ khối chức năng chính ATmega 2560.....47
Hình 31: Sơ đồ mạch DS1307.....47
Hình 32: Thiết kế board mạch bộ xử lý trung tâm.....48
Hình 33: Board mạch bộ xử lý trung tâm.....48

Hình 34: Lưu đồ thuật toán Khối xử lý cảm biến.....49
Hình 35: Lưu đồ thuật toán Khối xử lý đầu vào mở rộng.....49
Hình 36: Lưu đồ thuật toán Khối xử lý đầu bàn phím.....50
Hình 37: Lưu đồ thuật toán khối xử lý chính.....50
Hình 38: Lưu đồ thuật toán đọc trạng thái thiết bị từ backend.....107
Hình 39: Lưu đồ thuật toán hiển thị trạng thái thiết bị lên giao diện.....110
Hình 40: Giao diện hiển thị trên Mobile App (1).....134
Hình 41: Giao diện hiển thị trên Mobile App (2).....134
Hình 42: Giao diện hiển thị trên Mobile App (3).....135
Hình 43: Sơ đồ ghép nối các thành phần của hệ thống Báo cháy tự động.....136
Hình 44: Tủ điều khiển báo cháy tự động.....139

I. PHẦN MỞ ĐẦU

1. Tính cấp thiết thực hiện nhiệm vụ

Theo thống kê, 6 tháng đầu năm 2017, cả nước xảy ra 2.364 vụ cháy (tăng 858 vụ, 56,9%), làm 51 người chết, 95 người bị thương, thiệt hại về tài sản khoảng 1.173 tỷ đồng và 806 ha rừng. Cháy tại nhà dân, khu công nghiệp vẫn diễn biến phức tạp. Nguyên nhân cháy chủ yếu là do sự cố hệ thống điện và thiết bị điện.

Tại Hưng Yên:

- Đêm ngày 19/3/2014 đã xảy ra vụ cháy tại chợ Phố Hiến, Thành phố Hưng Yên. Hàng trăm kios cùng hàng hóa ở hai tầng chợ đã bị thiêu rụi.

- Ngày 29/10/2014, tại một kho chứa hàng mút xốp Công ty TTHH Ngọc Long ở thôn Sài Phi, xã Minh Đức, huyện Mỹ Hào, tỉnh Hưng Yên đã xảy ra một vụ cháy lớn với diện tích kho, xưởng: 3600m2 .

- Ngày 13/3/2016, một đám cháy bùng phát từ khu vực nhà kho sản xuất dao cạo của công ty TNHH Dorco Living Vina (Khu công nghiệp Phố Nối A, Văn Lâm, Hưng Yên), sau đó nhanh chóng lan rộng ra khu nhà xưởng rộng hàng trăm mét vuông.

- Ngày 21/3/2017, một nhà kho chứa, thu mua phế liệu ở xã Tân Quang, huyện Văn Lâm, tỉnh Hưng Yên bất ngờ bốc cháy. Chỉ trong chốc lát, ngọn lửa nhanh chóng lan rộng khắp khu nhà kho.

- Sáng ngày 18/10/2017, một cửa hàng điện thoại ở gần Cầu Hầu (xã Trung Hòa, huyện Yên Mỹ, Hưng Yên) bất ngờ bốc cháy dữ dội.

- Gần đây, ngày 25/7/2018 đã xảy ra vụ cháy lớn Công ty CP nhựa Hưng Yên và Chợ Gạo trên đường Nguyễn Văn Linh, TP Hưng Yên, tỉnh Hưng Yên. Đám cháy thiêu rụi hàng trăm mét vuông nhà xưởng cùng nhiều ki ốt bán hàng của các tiểu thương ở Chợ Gạo. Sau một đêm, nhiều tiểu thương lâm vào cảnh trắng tay bởi hậu quả của vụ hỏa hoạn.

Thực tế, nhiều tổ chức, cơ quan, xí nghiệp, người dân đã trang bị hệ thống báo cháy với các tính năng phát hiện các dấu hiệu cháy để cảnh báo (tại chỗ) hoặc có thể điều khiển thiết bị chữa cháy tại chỗ như xả khí, phun nước… Các hệ thống này có hạn chế là chưa có tính năng cảnh báo qua điện thoại hay phương tiện truyền thông khoảng cách xa khác. Do vậy, khi các đám cháy được phát hiện và cơ quan phòng cháy chữa cháy đến hiện trường thì khả năng đảm bảo an toàn tài sản gần như không có, đã gây thiệt hại lớn về con người và tài sản của các tổ chức, cá nhân và doanh nghiệp.

Hiện nay, Trung tâm tích hợp dữ liệu tỉnh (do Trung tâm Công nghệ thông tin và Truyền thông quản lý, vận hành) được đầu tư cơ sở hạ tầng, kỹ thuật với nhiều trang thiết bị có giá trị lớn, triển khai các hệ thống thông tin, lưu trữ dữ liệu tập trung của các cơ quan, đơn vị trên địa bàn tỉnh như:

- Hệ thống thư điện tử công vụ tỉnh Hưng Yên: đã cấp hơn 6000 hộp thư để phục vụ cho cán bộ, công chức, viên chức của các sở, ban, ngành, UBND các huyện/thành phố và các xã trên địa bàn tỉnh.

- Hệ thống quản lý cán bộ, công chức viên chức tỉnh Hưng Yên: với hơn 10.000 hồ sơ của cán bộ, công chức, viên chức tỉnh Hưng Yên được nhập vào hệ thống.

- Hệ thống 1 cửa điện tử: phục vụ việc cung cấp, tiếp nhận và giải quyết các thủ tục hành của tổ chức, người dân và doanh nghiệp của các sở, ban, ngành, UBND các huyện/thành phố trên địa bàn tỉnh.

- Hệ thống phần mềm quản lý văn bản và điều hành tỉnh Hưng Yên: phục vụ việc lưu trữ, trao đổi và xử lý văn bản đến, đi của các sở, ban, ngành, UBND các huyện/thành phố trên địa bàn tỉnh.

- Hệ thống Hội nghị truyền hình trực tuyến tỉnh Hưng Yên: phục vụ việc họp trực tuyến của các sở, ban, ngành, UBND các huyện/thành phố.

Trung tâm Công nghệ thông tin và truyền thông đã được trang bị hệ thống phát hiện và chữa cháy tại chỗ bằng phương pháp xả khí và cảnh báo trực tiếp tại nơi xảy ra sự cố. Hệ thống này, trong điều kiện hoạt động bình thường có thể phát hiện được cháy thông qua các dấu hiệu nhận biết cháy như: khói, nhiệt và thực hiện báo cháy tại chỗ qua hệ thống đèn, còi lắp tại hiện trường. Tuy nhiên, hệ thống này có một số hạn chế sau:

- Khi hệ thống báo cháy hoạt động thì sự cố cháy có thể đã bắt đầu do việc báo cháy là dựa trên các dấu hiệu cháy như khói, nhiệt.

- Hệ thống này không có khả năng cảnh bảo trước qua các dấu hiệu hoạt động bất thường của hệ thống có thể là nguyên nhân gây cháy như thiết bị điện.

Năm 2018, Trung tâm tích hợp dữ liệu tỉnh Hưng Yên tiếp tục được đầu tư theo Quyết định số 2391/QĐ-UBND ngày 31/10/2016, ước tính tổng giá trị tài sản của các hệ thống khoảng hơn 60 tỷ đồng. Tuy nhiên, việc quản lý về an toàn, phòng chống cháy nổ và điều khiển thiết bị tại Trung tâm tích hợp dữ liệu tỉnh còn nhiều hạn chế, bởi các yếu tố như: các hệ thống thông tin hoạt động liên tục 24/24h nên rất dễ xảy ra tình trạng quá tải đường điện, các thiết bị được sử dụng lâu ngày mà không được bảo dưỡng, thay thế thường xuyên… Nếu xảy ra sự cố cháy nổ sẽ gây ra tổn thất vô cùng lớn. Tổn thất đầu tiên, dễ dàng nhận thấy là

trang thiết bị đắt tiền được trang bị tại Trung tâm. Bên cạnh đó, có những tổn thất có thể là khó nhận thấy hơn nhưng hậu quả lại có thể rất lớn đó là làm mất dữ liệu hệ thống, dữ liệu khách hàng, làm gián đoạn các tiến trình hoạt động trên hệ thống.

Để tăng cường khả năng phòng và chữa cháy, cần thiết phải có hệ thống phòng, báo và chữa cháy toàn diện hơn với khả năng giám sát, phát hiện sớm nguy cơ cháy, đặc biệt là nguy cơ cháy do quá tải điện tại Trung tâm. Ngoài chức năng phát hiện các dấu hiệu cháy và điều khiển chữa cháy thì cần thiết có phương tiện, ứng dụng giúp người có trách nhiệm có thể giám sát liên tục, giám sát từ xa để sớm tiếp cận thông tin sự cố cháy giúp phòng ngừa, phát hiện và xử lý sự cố kịp thời, tăng mức độ an toàn phòng và chữa cháy tại Trung tâm.

Xuất phát từ lý do nêu trên, nhóm nghiên cứu đề xuất và thực hiện đề tài "Nghiên cứu, ứng dụng tiến bộ khoa học công nghệ xây dựng hệ thống báo cháy tự động và điều khiển thiết bị từ xa qua mạng điện thoại di động tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên" nhằm khắc phục các hạn chế của hệ thống cảnh báo và chữa chạy hiện có, nâng cao khả năng phát hiện, chữa cháy hiệu quả, đảm bảo an toàn tài sản, thông tin tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên.

2. Tổng quan tình hình nghiên cứu

2.1. Tình hình nghiên cứu trong nước

Ở phạm vi trong nước, vấn đề báo cháy tự động và điều khiển thiết bị từ xa luôn là tâm điểm nghiên cứu của các nhà khoa học. Trong những năm qua, việc nghiên cứu, ứng dụng và phát triển các thành tựu khoa học để áp dụng vào thực tiễn ở Việt Nam đã và đang phát triển rất mạnh, đặc biệt trong lĩnh vực tự động hóa và điều khiển từ xa:

- Một số đề tài nghiên cứu báo động khi có sự cố cháy nhưng các đề tài này phần lớn được tập trung nghiên cứu về mặt lý thuyết còn ứng dụng vẫn có các hạn chế như chỉ cảnh báo tại nơi xảy ra sự cố, không tự động thông báo đến bộ phận an ninh cách xa nơi xảy ra sự cố hoặc cơ quan phòng cháy chữa cháy bằng phương pháp tự động gọi điện thoại hoặc đã có nhưng hoạt động độc lập theo các hệ thống riêng lẻ.

- Một số hệ thống báo chạy có sẵn mặc dù đã tích hợp tính năng thông báo qua điện thoại song hoạt động độc lập. Khó khắc hoặc không có khả năng tích hợp với hệ thống báo cháy có sẵn. Trong thực tế, nhiều tổ chức, cơ quan, xí nghiệp, người dân đã trang bị hệ thống báo cháy trước kia với các tính năng phát hiện các dấu hiệu cháy để cảnh báo (tại chỗ) hoặc có thể điều khiển thiết bị chữa cháy tại chỗ như xả khí, phun nước… Các hệ thống này có hạn chế là chưa có tính năng cảnh báo qua điện thoại hay phương tiện truyền thông khoảng cách xa

khác. Nếu trang bị hệ thống mới có tính năng cảnh báo qua điện thoại sẵn có và thay thế hoàn toàn hệ thống thông thường hiện có sẽ dẫn tới lãng phí vì phải bỏ hệ thống cũ. Việc tích hợp hệ thống mới sẵn có với tính năng cảnh báo qua điện thoại là khó khăn hoặc không thực hiện được do các hệ thống này thường chưa có hỗ trợ.

- Phương pháp cảnh báo qua điện thoại di động đã được sử dụng khá phổ biến trong nhiều lĩnh vực khác nhau không riêng trong lĩnh vực phòng và chữa cháy. Đến nay đã có những đề tài nghiên cứu sử dụng mạng điện thoại để điều khiển thiết bị như: hệ thống tưới nước tự động, hệ thống giám sát và điều khiển camera tự động… Điều đó cho thấy xu hướng sử dụng điện thoại di động để truyền tải thông tin cảnh báo cũng như sử dụng như một thiết bị điều khiển từ xa đã trở lên phổ biến và có nhiều ưu điểm, thường được lựa chọn.

- Gần đây đã có một số hệ thống báo cháy hỗ trợ tính năng cảnh báo qua điện thoại thậm chí là quan sát trạng thái trên máy tính kết nối mạng cục bộ với bộ điều khiển trung tâm song tính năng mới tập trung ở các số liệu đầu vào từ các cảm biến khói, nhiệt độ. Do vậy không thực hiện cảnh báo sớm được. Khi có dấu hiệu khói hay nhiệt tăng cao thì thường đã bắt đầu cháy. Chưa hỗ trợ giám sát từ xa qua Internet với điện thoại thông minh là phương tiện mỗi người thường mang theo giúp gia tăng khả năng giám sát và tiếp nhận thông tin cảnh báo từ hệ thống.

2.2. Tình hình nghiên cứu ngoài nước

Trên thế giới hiện nay, việc báo cháy tự động và điều khiển thiết bị từ xa qua mạng điện thoại di động không còn mới mẻ nữa. Đã có không ít những công trình nghiên cứu khoa học thành công khi dùng hệ thống viễn thông sẵn có (mạng cố định và di động) để ứng dụng trong việc báo động, cảnh báo sự cố và điều khiển thiết bị từ xa như:

- Tại Nga: có những nhà máy điện, những kho lưu trữ tài liệu đã ứng dụng hệ thống báo động và điều khiển từ xa thông qua đường điện thoại để đóng ngắt những nơi cao áp, tự động quay số báo động đến bộ phận an ninh hoặc cơ quan phòng cháy chữa cháy, tự động xả bình chữa cháy khi có sự cố… và báo động thông qua mạng Internet để điều khiển nhà máy điện nguyên tử.

- Tại Mỹ: áp dụng các thành tựu khoa học để phát triển các sản phẩm trong việc đảm bảo an toàn, an ninh trong các công trình dân dụng, tòa nhà, chung cư lớn như: sử dụng hệ thống khóa cửa, két sắt tự động được lắp đặt bí mật thông qua tổng đài nội bộ.

3. Mục tiêu nghiên cứu

Nghiên cứu, ứng dụng các tiến bộ, thành tựu của khoa học công nghệ trong việc phòng chống cháy nổ, điều khiển thiết bị điện từ xa nhằm bảo đảm an toàn tài sản, các hệ thống thông tin tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên, đáp ứng yêu cầu theo Chỉ thị số 11/CT-UBND ngày 10/10/2017 của UBND tỉnh Hưng Yên về việc tăng cường công tác phòng cháy chữa cháy trên địa bàn tỉnh Hưng Yên, với mục tiêu cụ thể như sau:

- Nghiên cứu về cơ sở khoa học (lý luận và thực tiễn) hoạt động của hệ thống báo cháy tự động và điều khiển thiết bị từ xa qua mạng điện thoại di động.

- Ứng dụng kết quả nghiên cứu để xây dựng hệ thống báo cháy tự động tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên.

4. Đối tượng và phạm vi nghiên cứu

- **Về lý thuyết:**

- Nghiên cứu lý thuyết về hệ thống viễn thông, nguyên lý hoạt động của mạng điện thoại.

- Nghiên cứu, khảo sát về nguyên lý hoạt động của một số loại vi mạch tích hợp IC (Integrated Circuit) và vi điều khiển được sử dụng trong thiết kế và mô hình thực nghiệm.

- **Về mô hình thực nghiệm:**

- Nghiên cứu các giải pháp triển khai giám sát hệ thống qua cloud và phần mềm chạy trên điện thoại thông minh.

- Nghiên cứu giải pháp tích hợp với hệ thống báo cháy hiện có tại Trung tâm tích hợp dữ liệu tỉnh.

- Nghiên cứu mô hình hệ thống báo cháy tự động của các tỉnh có Trung tâm tích hợp dữ liệu (tại Bắc Ninh, Quảng Ninh) để bổ sung, có phương án tối ưu áp dụng vào quá trình xây dựng, hoàn thiện hệ thống báo cháy tự động tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên.

II. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

1. Nội dung nghiên cứu

1.1. Nghiên cứu tổng quan, cơ sở lý thuyết

- Nghiên cứu lý thuyết về hệ thống viễn thông, nguyên lý hoạt động của mạng điện thoại.

- Nghiên cứu, khảo sát về nguyên lý hoạt động của một số loại vi mạch tích hợp IC (Integrated Circuit) và vi điều khiển được sử dụng trong thiết kế và mô hình thực nghiệm.

- Nghiên cứu giải pháp tích hợp với hệ thống báo cháy hiện có tại Trung tâm tích hợp dữ liệu tỉnh.

- Nghiên cứu các giải pháp triển khai giám sát hệ thống qua cloud và phần mềm chạy trên điện thoại thông minh.

- Báo cáo đánh giá nghiên cứu và cơ sở khoa học thực hiện đề tài.

1.2. Tính toán, thiết kế và xây dựng mô hình thực nghiệm

Trên cơ sở nghiên cứu về mặt lý thuyết, thu thập tài liệu và xử lý dữ liệu để tiến hành thực hiện việc tính toán và thiết kế các khối điều khiển, bao gồm:

- Khối cảm biến đầu vào
- Khối kết nối thuê bao để gửi tin nhắn, gọi điện
- Khối giải mã, hiển thị và giao tiếp phím nhấn
- Khối điều khiển động lực đóng ngắt tiếp điểm khô
- Khối điều khiển ngắt điện thiết bị cần được bảo vệ
- Khối cảnh báo tại chỗ bằng tín hiệu đèn và âm thanh
- Khối kết nối với hệ thống báo cháy có sẵn.
- Khối giao tiếp phần mềm giám sát chạy trên điện thoại thông minh và đẩy dữ liệu trạng thái hệ thống lên cloud
- Khối backend xử lý dữ liệu trên cloud
- Khối frontend phần mềm giám sát chạy trên điện thoại thông minh
- Khối xử lý trung tâm.

2. Vật liệu và phương pháp nghiên cứu

2.1. Vật liệu sử dụng trong nghiên cứu

Sử dụng các vật liệu có sẵn trên thị trường phù hợp với tính năng và yêu cầu thiết kế, như:

- Cảm biến dòng SCT013-000V 100A/1V, HST21-200A
- Khởi động từ LS 3P contactor 150A
- Module Ghi Đọc Âm Thanh WT588D Arduino
- KIT Arduino MEGA2560 ATMEGA
- Đầu báo cháy khói
- Đầu báo cháy nhiệt
- Chuông báo cháy
- Linh kiện số, linh kiện vào ra, tụ điện, điện trở, dây điện…
- Acquy 12V5A
- …….

2.2. Phương pháp nghiên cứu

Đề tài được thực hiện dựa trên hai phương pháp nghiên cứu chủ yếu sau:

- Phương pháp nghiên cứu lý thuyết: tìm hiểu, nghiên cứu các tài liệu, công trình nghiên cứu trong và ngoài nước về hệ thống viễn thông, kỹ thuật số, kỹ thuật điện tử, các IC chuyên dùng và các tài liệu về vi điều khiển, vi xử lý.

- Phương pháp thực nghiệm: trên cơ sở các nghiên cứu về lý thuyết liên quan đến việc thực hiện mục tiêu của đề tài để lựa chọn phương án thực hiện, thiết kế, xây dựng mô hình thực nghiệm, lập trình phần mềm cho vi điều khiển để kết nối với phần cứng và hệ thống thông tin để tiến hành thực nghiệm.

III. KẾT QUẢ NGHIÊN CỨU VÀ THẢO LUẬN

1. KẾT QUẢ NGHIÊN CỨU

1.1. NGUYÊN LÝ HOẠT ĐỘNG CỦA MẠNG ĐIỆN THOẠI

1.1.1. Tổng đài

1.1.1.1. Khái niệm về tổng đài:

Tổng đài là một hệ thống chuyển mạch có hệ thống kết nối các cuộc liên lạc giữa các thuê bao với nhau, với số lượng thuê bao lớn hay nhỏ tuỳ thuộc vào từng loại tổng đài, từng khu vực.

1.1.1.2. Chức năng của tổng đài:

Tổng đài điện thoại có chức năng :

- ¬ Nhận biết được khi thuê bao nào có nhu cầu xuất phát cuộc gọi.
- ¬ Thông báo cho thuê bao biết mình sẵn sàng tiếp nhận các yêu cầu của thuê bao.

Xử lí thông tin từ thuê bao chủ gọi để điều khiển kết nối theo yêu cầu.

- ¬ Báo cho thuê bao bị gọi biết có người cần muốn liên lạc.
- ¬ Giám sát thời gian và tình trạng thuê bao để ghi cước và giải tỏa.
- ¬ Giao tiếp được với những tổng đài khác để phối hợp điều khiển.

1.1.1.3. Phân loại tổng đài:

a. Tổng đài công nhân:

Việc kết nối thông thoại, chuyển mạch dựa vào con người.

b. Tổng đài cơ điện:

Bộ phận thao tác chuyển mạch là hệ thống cơ khí, được điều khiển bằng hệ thống mạch từ. Gồm hai hệ thống chuyển mạch cơ khí cơ bản: chuyển mạch từng nấc và chuyển mạch ngang dọc.

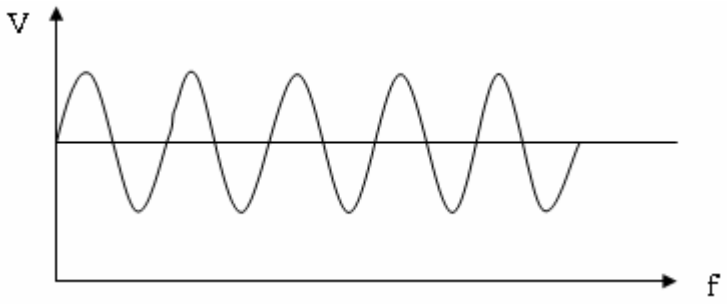
c. Tổng đài điện tử:

Quá trình điều khiển kết nối hoàn toàn tự động, vì vậy người sử dụng cũng không thể cung cấp cho tổng đài những yêu cầu của mình bằng lời nói được. Ngược lại, tổng đài trả lời cho người sử dụng cũng không thể bằng lời nói. Do đó, cần qui định một số thiết bị cũng như các tín hiệu để người sử dụng và tổng đài có thể làm việc được với nhau.

1.1.1.4. Các loại âm hiệu:

a. Tín hiệu mời quay số (Dial tone):

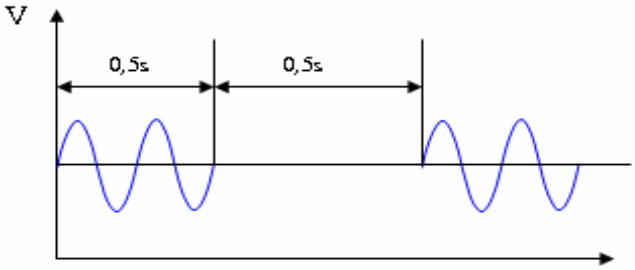
Khi thuê bao nhấc tổ hợp để xuất phát cuộc gọi sẽ nghe âm hiệu mời quay số do tổng đài cấp cho thuê bao gọi, là tín hiệu hình sin có tần số 425 ± 25 Hz liên tục.



Hình 1: Tín hiệu Dial tone

b. Tín hiệu báo bận (Busy tone):

Tín hiệu này báo cho người sử dụng biết thuê bao bị gọi đang trong tình trạng bận hoặc trong trường hợp thuê bao nhấc máy quá lâu mà không quay số thì tổng đài gởi âm hiệu báo bận này. Tín hiệu báo bận là tín hiệu hình sin có tần số 425 ± 25 Hz, ngắt quãng 0.5 giây có và 0.5 giây không.



Hình 2: Tín hiệu Busy Tone

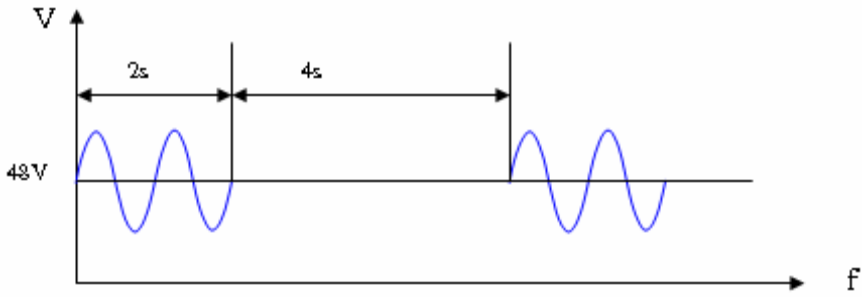
c. Tín hiệu chuông (Ring tone):

Tín hiệu chuông do tổng đài cung cấp cho thuê bao bị gọi, là tín hiệu hình sin có tần số 25 Hz và điện áp 90V hiệu dụng, ngắt quãng tuỳ thuộc vào tổng đài, thường 2 giây có và 4 giây không.

Hình 3: Tín hiệu Ring Tone

d. Tín hiệu hồi chuông (Ring back tone):

Tín hiệu hồi chuông do tổng đài cấp cho thuê bao gọi, là tín hiệu hình sin có tần số 425 ± 25 Hz là hai tín hiệu ngắt quãng 2s có 4s không tương ứng với nhịp chuông.



Hình 4: Tín hiệu Ring back Tone

1.1.1.5. Phương thức chuyển mạch của tổng đài điện tử:

Tổng đài điện tử có những phương thức chuyển mạch sau :

- Tổng đài điện tử dùng phương thức chuyển mạch không gian (SDM: Space Devision Multiplexer)

- Tổng đài điện tử dùng phương thức chuyển mạch thời gian (TDM : Timing Devision Multiplexer): có hai loại.

Phương thức ghép kênh theo tần số (FDM: Frequence Devision Multiplexer).

Phương thức ghép kênh tương tự theo thời gian (Analog TDM) gồm có :

+ Ghép kênh bằng phương thức truyền đạt cộng hưởng.

+ Ghép kênh PAM (PAM : Pulse Amplitude Modulation).

Trong kỹ thuật ghép kênh PCM người ta lại chia 2 loại: điều chế Delta và điều chế PCM.

Ngoài ra, đối với tổng đài có dung lượng lớn và rất lớn (dung lượng lên đến cỡ vài chục ngàn số) người ta phối hợp cả hai phương thức chuyển mạch SDM và TDM thành T – S – T, T – S, S – T – S ….

Ưu điểm của phương thức kết hợp này là tận dụng tối đa số link trống và giảm bớt số link trông không cần thiết, làm cho kết cấu của toàn tổng đài trở nên đơn giản hơn. bởi vì, phương thức ghép kênh TDM luôn luôn tạo ra khả năng toàn thông, mà thông thường đối với tổng đài có dung lượng lớn, việc dư link là không cần thiết. Người ta đã tính ra thông thường chỉ có tối đa 10% các thuê bao có yêu cầu cùng 1 lúc, nên số link trống chỉ cần đạt 10% tổng số thuê bao là đủ.

1.1.2. Máy điện thoại

1.1.2.1. Các thông số cơ bản của máy điện thoại:

Tổng đài được nối với các thuê bao qua 2 đước truyền TIP và RING. Thông qua 2 đường dây này thông tin từ tổng đài qua các thuê bao được cấp bằng nguồn dòng từ 25 mA đến 40 mA (trung bình chọn 35 mA) đến cho máy điện thoại.

- Tổng trở DC khi gác máy lớn hơn từ 20 K
- Tổng trở AC khi gác máy từ 4K đến 10K
- Tổng trở DC khi nhấc máy nhỏ hơn 1K (từ 0,2K ÷ 0,6K).

Các thông số và giới hạn máy điện thoại :

Thông số	Các giá trị mẫu	Giá trị sử dụng
Dòng làm việc	20 – 80 mA	20 – 120 mA
Nguồn tổng đài	-48 -> -60V	-40 -> -150V
Điện trở vòng	0 – 1300 Ohm	0 – 1600 Ohm
Suy hao	8dB	17dB
Méo dạng	50dB	50dB
Dòng chuông	90Vmrs/20Hz	75 – 90Vmrs/16 – 25Hz
Thanh áp ống nói	70 – 90 dB	<15dB

1.1.2.2. Các hoạt động trên mạng của máy điện thoại:

Tổng đài nhận biết trạng thái nhấc máy của thuê bao hay gác máy bằng cách sử dụng nguồn một chiều 48VDC.

Khi gác máy tổng trở DC bằng 20K rất lớn xem như hở mạch.

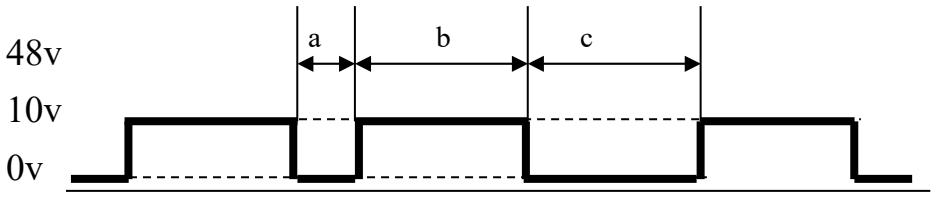
Khi nhấc máy tổng trở DC giảm xuống nhỏ hơn 1K và hai tổng đài nhận biết trạng thái này thông qua dòng DC xuất hiện trên đường dây. Sau đó, tổng đài cấp tín hiệu mời gọi lên đường dây đến thuê bao.

¬ Quay số :

Người gọi thông báo số mình muốn gọi cho tổng đài biết bằng cách gởi số máy điện thoại của mình muốn gọi đến cho tổng đài. Có hai cách gởi số đến tổng đài :

Phương thức quay số tone DTMF và PULSE: Khi có một phím được ấn thì trên đường dây sẽ xuất hiện 2 tấn số khác nhau thuộc nhóm fthấp và fcao. Phương pháp tần ghép này chống nhiễu tốt hơn, ngoài ra dùng dạng tone DTMF sẽ tăng được tốc độ quay nhanh gấp 10 lần so với việc thực hiện quay số PULSE. Mặt khác phương pháp sẽ sử dụng được một số dịch vụ cộng thêm tổng đài.

Phương pháp quay số pulse: tín hiệu quay số là chuỗi xung vuông, tần số chuỗi dự án = 10Hz,số điện thoại bằng số xung ra, riêng số 0 sẽ là 10 xung, biên độ ở mức cao là 48v, ở mức thấp là 10v, dạng sóng được cho ở hình dưới:



Hình 5: Quay số kiểu dạng Pulse

a: chu kỳ làm việc (thời gian 48v)
b: thời gian ở 10v, ta có a/b = 66/33 = 2
c: khoảng thời gian giữa 2 lần quay số trong một cuộc gọi
Số xung trên một giây 10 – 20 pulse/s

¬ Quay số bằng Tone (Tone – Dialing): Máy điện thoại phát ra cùng lúc hai tín hiệu với tần số dao động khác nhau tương ứng với số muốn quay (DTMF : Dual Tone Multi Frequence) theo bảng sau

Bảng 1: Bảng phân loại tần số tín hiệu Tone

F_{LOW}	F_{HIGH}	Digit
697	1209	1
697	1336	2
697	1477	3
770	1209	4
770	1336	5
770	1477	6

852	1209	7
852	1336	8
852	1477	9
941	1336	0
941	1209	*
941	1477	#

¬ Kết nối thuê bao :

Tổng đài nhận được các số liệu sẽ xem xét:

Nếu các đường dây nối thông thoại đều bị bận thì tổng đài sẽ cấp tín hiệu báo bận.

Nếu đường dây nối thông thoại không bận thì tổng đài sẽ cấp cho người bị gọi tín hiệu chuông và người gọi tín hiệu hồi chuông. Khi người được gọi nhấc máy, tổng đài nhận biết trạng thái này, thì tổng đài ngưng cấp tín hiệu chuông để không làm hư mạch thoại và thực hiện việc thông thoại. Tín hiệu trên đường dây đến máy điện thoại tương ứng với tín hiệu thoại cộng với giá trị khoảng 300 mV đỉnh – đỉnh. Tín hiệu ra khỏi máy điện thoại chịu sự suy hao trên đường dây với mất mát công suất trong khoảng 10 dB ÷ 25 dB. Giả sử suy hao là 20 dB, suy ra tín hiệu ra khỏi máy điện thoại có giá trị khoảng 3V đỉnh – đỉnh.

¬ Ngưng thoại:

Khi một trong 2 thuê bao gác máy, thì tổng đài nhận biết trạng thái này, cắt thông thoại cho cả 2 máy đồng thời cấp tín hiệu báo bận cho máy còn lại

¬ Tín hiệu thoại:

Tín hiệu thoại trên đường dây là tín hiệu điện mang các thông tin có nguồn gốc từ âm thanh trong quá trình trao đổi giữa 2 thuê bao. Trong đó, âm thanh được tạo ra bởi các dao động cơ học, nó truyền trong môi trường dẫn âm.

Khi truyền đi trong mạng điện thoại là tín hiệu thường bị méo dạng do những lý do: nhiễu, suy hao tín hiệu trên đường dây do bức xạ sóng trên đường dây với các tần số khác nhau. Để đảm bảo tín hiệu điện thoại nghe rõ và trung thực, ngày nay trên mạng điện thoại người ta sử dụng tín hiệu thoại có tần số từ 300 Hz ÷ 3400 Hz.

1.1.3. Phương thức hoạt động giữa tổng đài và máy điện thoại

Tổng đài nhận dạng thuê bao gọi nhấc máy thông qua sự thay đổi tổng trở mạch vòng của đường dây thuê bao. Bình thường khi thuê bao ở vị trí gác máy điện trở mạch vòng là rất lớn. Khi thuê bao nhấc máy, điện trở mạch vòng thuê bao giảm xuống còn khoảng từ 150 đến 1500. Tổng đài có thể nhận biết sự thay

đổi tổng trở mạch vòng này (tức là thay đổi trạng thái của thuê bao) thông qua các bộ cảm biến trạng thái. Tổng đài cấp âm hiệu mời quay số (Dial Tone) cho thuê bao. Dial Tone là tín hiệu mời quay số hình sin có tần số 425 ± 25 Hz. Khi thuê bao nhận biết được tín hiệu Dial Tone, người gọi sẽ hiểu là được phép quay số. Người gọi bắt đầu tiến hành gửi các xung quay số thông qua việc quay số hoặc nhấn phím chọn số. Tổng đài nhận biết được các số được quay nhờ vào các chuỗi xung quay số phát ra từ thuê bao gọi. Thực chất các xung quay số là các trạng thái nhấc máy hoặc gác máy của thuê bao. Nếu các đường kết nối thông thoại bị bận hoặc thuê bao được gọi bị bận thì tổng đài sẽ phát tín hiệu báo bận cho thuê bao. Am hiệu này có tần số f = 425 ± 25 Hz ngắt nhịp 0,5s có 0,5 s không. Tổng đài nhận biết các số thuê bao gọi đến và nhận xét:

Nếu số đầu nằm trong tập thuê bao thì tổng đài sẽ phục vụ như cuộc gọi nội đài.

Nếu số đầu là số qui ước gọi ra thì tổng đài phục vụ như một cuộc gọi liên đài qua trung kế và gửi toàn bộ phần định vị số quay sang tổng đài đối phương để giải mã.

Nếu số đầu là mã gọi các chức năng đặc biệt, tổng đài sẽ thực hiện các chức năng đó thuê yêu cầu của thuê bao. Thông thường, đối với loại tổng đài nội bộ có dung lượng nhỏ từ vài chục đến vài trăm số, có thêm nhiều chức năng đặc biệt làm cho chương trình phục vụ thuê bao thêm phong phú, tiện lợi, đa dạng, hiệu quả cho người sử dụng làm tăng khả năng khai thác và hiệu suất sử dụng tổng đài.

Nếu thuê bao được gọi rảnh, tổng đài sẽ cấp tín hiệu chuông cho thuê bao với điện áp 90Vrms (AC), f = 25 Hz, chu kì 2s có 4s không. Đồng thời, cấp âm hiệu hồi chuông (Ring Back Tone) cho thuê bao gọi, âm hiệu này là tín hiệu sin f = 425 ÷ 25 Hz cùng chu kì nhịp với tín hiệu chuông gởi cho thuê bao được gọi.

Khi thuê bao được gọi nhấc máy, tổng đài nhận biết trạng thái máy này tiến hành cắt dòng chuông cho thuê bao bị gọi kịp thời tránh hư hỏng đáng tiếc cho thuê bao. Đồng thời, tiến hành cắt âm hiệu Ring Back Tone cho thuê bao gọi và tiến hành kết nối thông thoại cho 2 thuê bao.

Tổng đài giải tỏa một số thiết bị không cần thiết để tiếp tục phục vụ cho các cuộc đàm thoại khác.

Khi hai thuê bao đang đàm thoại mà 1 thuê bao gác máy, tổng đài nhận biết trạng thái gác máy này, cắt thông thoại cho cả hai bên, cấp tín hiệu bận (Busy Tone) cho thuê bao còn lại, giải tỏa link để phục vụ cho các đàm thoại khác. Khi thuê bao còn lại gác máy, tổng đài xác nhận trạng thái gác máy, cắt âm hiệu báo bận, kết thúc chương trình phục vụ thuê bao.

Tất cả hoạt động nói trên của tổng đài điện tử đều được thực hiện một cách hoàn toàn tự động. Nhờ vào các mạch điều khiển bằng điện tử, điện thoại viên có thể theo dõi trực tiếp toàn bộ hoạt động của tổng đài ở mọi thời điểm nhờ vào các bộ hiển thị, cảnh báo.

Điện thoại viên có thể trực tiếp điều khiển các hoạt động của tổng đài qua các thao tác trên bàn phím, hệ thống công tắc….các hoạt động đó có thể bao gồm: nghe xen vào các cuộc đàm thoại, cắt cưỡng bức các cuộc đàm thoại có ý đồ xấu, tổ chức điện thoại hội nghị…. Tổng đài điện tử cũng có thể được liên kết với máy điện toán để điều khiển hoạt động hệ thống.

1.2. HỆ THỐNG BÁO CHÁY TỰ ĐỘNG VÀ GIẢI PHÁP TÍCH HỢP

1.2.1. Nhiệm vụ của hệ thống báo cháy tự động

Theo tiêu chuẩn Việt Nam TCVN 5738-2001, hệ thống báo cháy tự động là hệ thống thiết bị tự động phát hiện và thông báo địa điểm cháy. Hệ thống báo cháy tự động bao gồm: Trung tâm báo cháy, các đầu báo cháy, chuông, đèn, nút nhấn và các thiết bị ngoại vi khác.

Nhiệm vụ của hệ thống báo cháy tự động là phát hiện ra cháy một cách nhanh chóng, chính xác và kịp thời trong vùng bảo vệ của hệ thống. Tự động phát ra các tín hiệu báo động, gửi tín hiệu điều khiển tới các thiết bị ngoại vi của hệ thống báo cháy tự động nhằm thực hiện một nhiệm vụ cụ thể nào đó. Đặc biệt, với hệ thống báo cháy tự động sử dụng đầu báo cháy khói thì nó còn có nhiệm vụ quan trọng hơn là cảnh báo, nghĩa là phát hiện và thông báo sự sắp cháy, sự cháy âm ỉ chưa có ngọn lửa.

Hệ thống báo cháy tự động có thể được chia thành nhiều loại khác nhau dựa trên công nghệ đầu dò (cảm biến) phát hiện cháy và phạm vi bảo vệ hay đặc điểm kỹ thuật của hệ thống.

Theo nguyên lý làm việc của đầu báo cháy, hệ thống báo cháy có thể chia làm một số loại như sau:

- Hệ thống báo cháy tự động sử dụng đầu báo cháy khói: là hệ thống báo cháy tự động làm việc dựa vào nguyên lý làm việc của đầu báo cháy dựa trên dấu hiệu khói. Hệ thống này chủ yếu phát hiện ra sự gia tăng nồng độ khói ở trong khu vực được bảo vệ để đưa ra cảnh báo.

- Hệ thống báo cháy tự động sử dụng đầu báo cháy nhiệt: là hệ thống báo cháy tự động làm việc dựa theo nguyên lý làm việc của đầu báo cháy nhiệt. Hệ thống này chủ yếu phát hiện ra sự thay đổi nhiệt độ ở trong khu vực được bảo vệ để đưa ra cảnh báo.

- Hệ thống báo cháy tự động sử dụng đầu báo cháy ánh sáng: làm việc dựa

vào nguyên lý làm việc của đầu báo cháy lửa. Hệ thống này chủ yếu phát hiện ra sự gia tăng cường ánh sáng của ngọn lửa trong khu vực được bảo vệ để phát tín hiệu cảnh báo.

- Hệ thống báo cháy tự động sử dụng đầu báo cháy hỗn hợp: là hệ thống làm việc dựa trên nguyên lý làm việc của đầu báo cháy hỗn hợp như: đầu báo cháy nhiệt và khói; đầu báo cháy nhiệt và lửa… Hệ thống này chủ yếu phát hiện ra sự thay đổi các yếu tố môi trường mà đầu dò được trang bị có khả năng phát hiện trong khu vực được bảo vệ để đưa ra tín hiệu cảnh báo.

Dựa trên đặc điểm kỹ thuật, phạm vi và độ chính xác vùng báo cháy, hệ thống báo cháy tự động có thể được chia làm các loại:

- Hệ thống báo cháy tự động theo vùng: là hệ thống báo cháy tự động có chức năng báo cháy ở một khu vực, một địa điểm. Hệ thống này có thể được trang bị nhiều đầu dò phát hiện dấu hiệu cháy. Chỉ cần một trong số các đầu báo cháy phát hiện dấu hiệu cháy thì hệ thống thực hiện báo cháy cho cả vùng được bảo vệ mà không chỉ ra chính xác cháy ở vị trí nào, do đầu dò nào phát hiện. Phạm vi bảo vệ có thể vài chục đến vài nghìn m^2.

- Hệ thống báo cháy tự động theo địa chỉ: là hệ thống báo cháy tự động có khả năng báo cháy chính xác đến từng vị trí từng đầu báo riêng biệt (từng địa chỉ cụ thể). Diện tích bảo vệ của một địa chỉ báo cháy chỉ giới hạn trong khoảng vài chục mét vuông (tuỳ thuộc vào từng loại đầu báo cháy).

- Hệ thống báo cháy tự động thông minh: với sự phát triển khoa học công nghệ, hệ thống báo cháy tự động theo địa chỉ được phát triển thành hệ thống báo cháy thông minh với nhiều tính năng cao cấp như ngoài xác định rõ địa chỉ cháy, hệ thống này còn có thể đo được một số thông số về môi trường của khu vực lắp đặt đầu báo cháy như nhiệt độ, nồng độ khói, các chỉ số một số loại khí,…

1.2.2. Nguyên lý làm việc hệ thống báo cháy tự động

Một hệ thống báo cháy tự động thông thường duy trì ở 3 trạng thái đó là: trạng thái thường trực, trạng thái sự cố, trạng thái báo cháy.

Ở trạng thái thường trực (đây cũng chính là trạng thái bình thường của hệ thống), bộ xử lý trung tâm có tín hiệu đến các thiết bị trong hệ thống để kiểm tra trạng thái làm việc của các bộ phần này. Nếu tín hiệu phản hồi về trong phạm vi an toàn thì hệ thống duy trì ở trạng thái này. Khi xuất hiện bộ phận nào đó không có phản hồi hay tín hiệu phản hồi ngoài phạm vi an toàn thì hệ thống chuyển sang trạng thái sự cố.

Trạng thái sự cố xuất hiện khi có ít nhất một bộ phận nào đó trong hệ thống không hoạt động bình thường, không có tín hiệu phản hồi hoặc báo lỗi của bộ

phận nào đó trong hệ thống. Các thông tin sự cố sẽ được hiển thị, cảnh báo cho người vận hành biết để khắc phục thông qua màn hinh LCD hay các đèn báo tùy thuộc cấu tạo của hệ thống.

Trạng thái báo cháy xuất hiện khi tín hiệu phát hiện ở đầu dò nào đó vượt ngưỡng giới hạn đã thiết lập. Ở trạng thái này, bộ xử lý trong tâm sẽ kích hoạt các thiết bị cảnh báo như còi, đèn hay gửi tín hiệu điều khiển tới các thiết bị ngoại vi như mở van, bơm nước, xả khí chữa cháy…

1.2.3. Ứng dụng mạng viễn thông trong hệ thống báo cháy tự động

Khi phát hiện cháy, hệ thống báo cháy tự động kích hoạt các thiết bị báo cháy tại chỗ như hệ thống loa (còi hú), hệ thống đèn nhằm gây chú ý và báo động. Thông qua các tín hiệu này mà đám cháy được phát hiện bởi những người ở gần từ đó thực hiện các biện pháp xử lý khi có cháy.

Việc báo cháy tại chỗ trong nhiều trường hợp không phát huy tác dụng khi không có người ở trong phạm vi có thể tiếp nhận được các tín hiệu báo cháy (không nghe thấy còi hú, không quan sát được đèn báo cháy). Vì vậy, một số hệ thống báo cháy tự động tích hợp khả năng báo cháy qua mạng viễn thông để có thể gửi tín hiệu báo cháy nhanh và trực tiếp đến người giám sát. Ở hệ thống báo cháy tự động có tính năng này, khi phát hiện cháy, ngoài thực hiện cảnh báo tại chỗ, hệ thống sẽ tiến hành quay số, thực hiện cuộc gọi tới số điện thoại đã được thiết lập từ trước để cảnh báo cháy. Liên lạc điện thoại có thể được thiện thông qua mạng điện thoại cố định hoặc di động. Hiện nay, liên lạc điện thoại di động đã trở lên phổ biến, dễ dàng thiết lập nên các hệ thống báo cháy tự động thường tích hợp khả năng liên lạc qua mạng di động. Ngoài gọi điện, với mạng di động, hệ thống báo cháy có thể gửi tin nhắn đến các số điện thoại đặt trước.

Ứng dụng mạng viễn thông trong báo cháy tự động có nhiều ưu điểm tích cực. Một số hệ thống báo cháy cũ chưa có khả năng khai thác mạng viễn thông cho báo cháy thì cần thiết được nâng cấp, mở rộng khả năng liên lạc, báo cháy thông qua mạng viễn thông để gia tăng khả năng bảo vệ của hệ thống.

1.2.4. Ứng dụng vi mạch tích hợp trong hệ thống báo cháy tự động

Với hệ thống báo cháy tự động hiện đại, với khả năng báo cháy chính xác (theo địa chỉ) cũng như tích hợp nhiều tính năng có thể được xây dựng dựa trên các vi mạch tích hợp.

Việc sử dụng các vi mạch tích hợp giúp giảm kích thước hệ thống đồng thời gia tăng khả năng xử lý như:

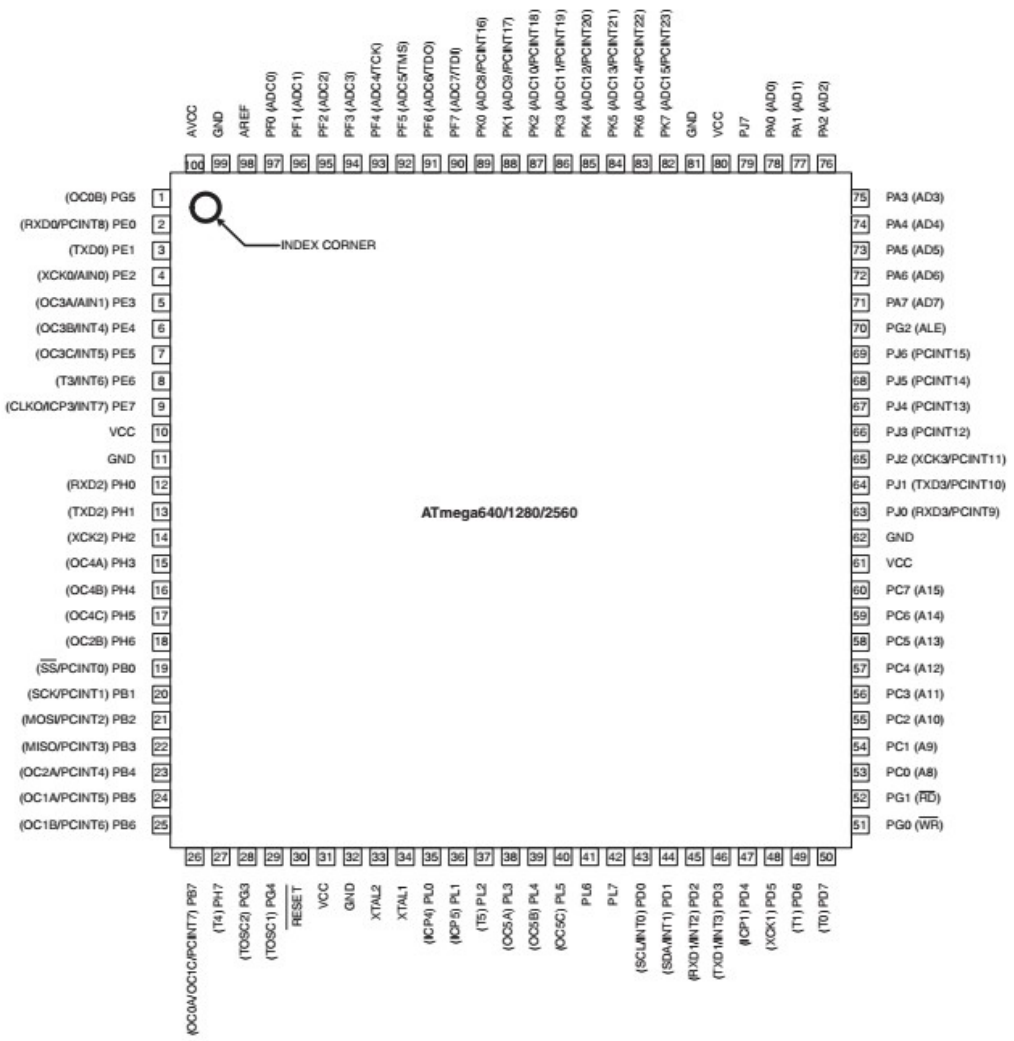
- Tiếp nhận và xử lý được tín hiệu từ nhiều loại đầu dò khác nhau
- Thực hiện được các thuật toán phân tích hiệu quả tín hiệu cảnh báo từ đầu

dò để đưa ra quyết định chính xác, hiệu quả

- Có khả năng tích hợp với các hệ thống khác giúp tăng khả năng bảo vệ và xử lý sự cố
- Kết nối, giao tiếp các thiết bị khác qua Internet, cập nhật dữ liệu môi trường lên cloud qua đó giúp giám sát hệ thống từ xa qua thiết bị di động, điện thoại thông minh đồng thời có thể điều khiển hoạt động của hệ thống mà không phải tới gần

Ngoài các vi mạch tích hợp chuyên dụng mà các hãng thiết kế, chế tạo hệ thống báo cháy xây dựng riêng cho sản phẩm của mình, một số loại vi xử lý, vi điều khiển đa chức năng có thể được ứng dụng vào thiết kế, chế tạo hệ thống báo cháy tự động thông minh như các dòng vi điều khiển AVR, một số vi mạch được phát triển cho các ứng dụng điều khiển tự động, IoT như ESP8266…

Vi điều khiển AVR là một trong các họ vi điều khiển đang được ứng dụng khá rộng rãi cho các ứng dụng giám sát, điều khiển. Đây là dòng vi điều khiển mạnh mẽ với nhiều tính năng hỗ trợ trong khi chi phí đầu tư nằm ở mức không cao, khả thi để triển khai cho các hệ thống điều khiển tự động. Vi điều khiển AVR có nhiều loại với cấu hình khác nhau và thích hợp cho các ứng dụng có mức độ phức tạp khác nhau. Trong số đó có thể kể đến chip vi điều khiển ATmega 2560. Đây là vi điều khiển 8 bit có dung lượng bố nhớ lớn và tốc độ làm việc cao.



Hình 6: Sơ đồ chân vi điều khiển Atmega2560

Atmega 2560 là loại vi điều khiển được thiết kế theo kiến trúc RISC với tập lệnh khá gọn bao gồm 135 lệnh. Hầu hết các lệnh được thực hiện trong một chu kỳ máy với 32 thanh ghi công dụng chung 8 bit, tần số làm việc lên đến 16MHz.

Các thông số kỹ thuật của ATMega2560:

- Bộ nhớ chương trình và bộ nhớ dữ liệu:
- 256KB bộ nhớ chương trình (Flash) cho phép ghi lại tới 10000 lần.
- 4KB bộ nhớ chương trình EEPROM
- 8KB bộ nhớ RAM tĩnh (SRAM), cho phép mở rộng bộ nhớ ngoài lên đến 64KB.
- Có chức năng khóa bảo vệ chương trình

Ngoại vi:

- Hai bộ Timer/Counters 8 bit
- Bốn bộ Timer/Counters 16 bit

- Bốn kênh PWM 8 bit
- Lên đến 12 kênh PWM với độ phân giải khả trình từ 2 tới 16 bit
- 16 kênh ADC 10 bit
- 4 cổng truyền thông USART
- Giao tiếp SPI chủ/tớ
- Giao tiếp hướng byte 2 dây
- Bộ Watchdog Timer khả trình với bộ dao động riêng
- Chân xử lý ngắt
- Có đến 86 chân vào ra công dụng chung

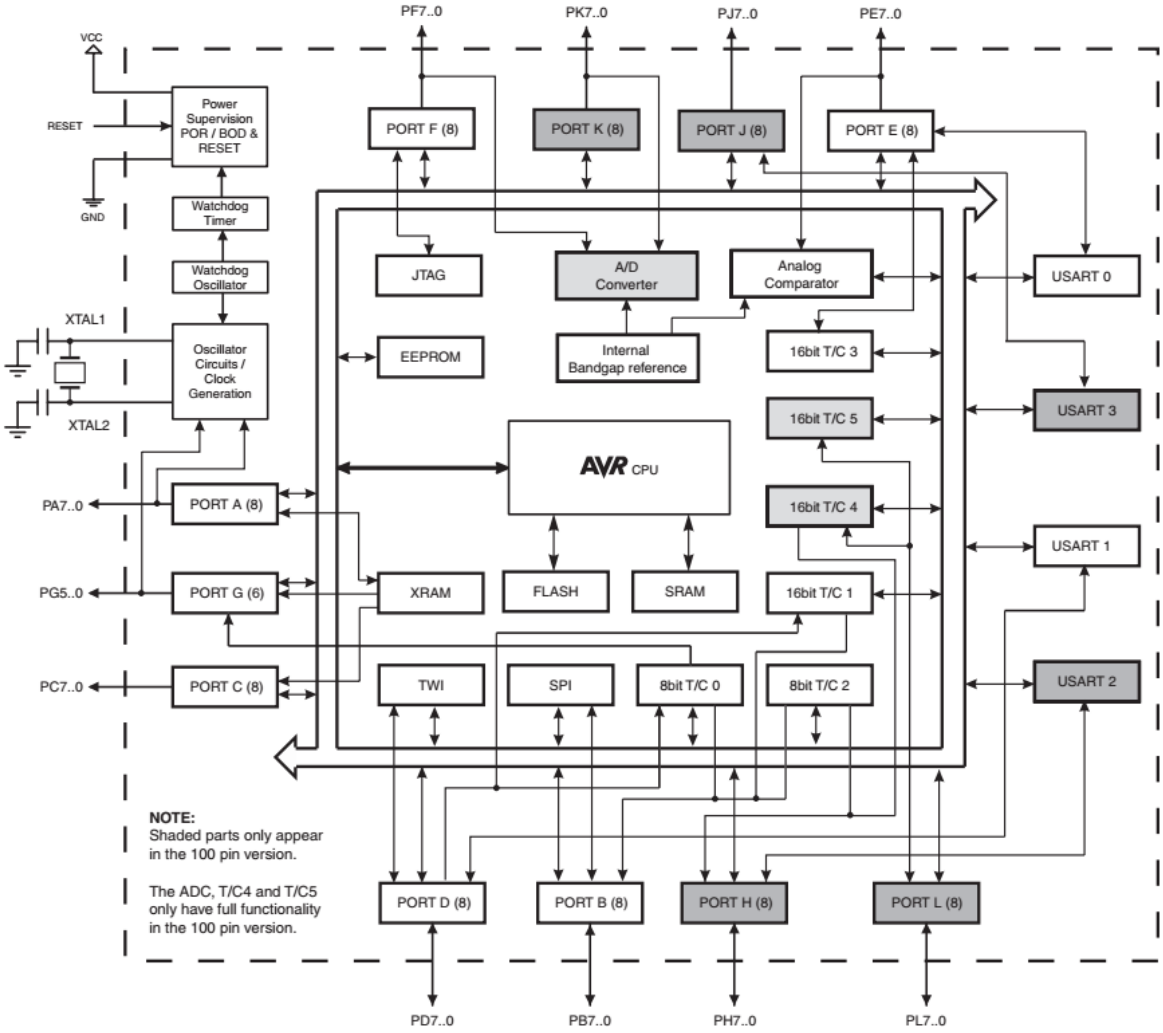
Ngoài Atmega2560, có thể kể đến một số mã VĐK cùng họ khác như Atmega640, Atmega1280, Atmega1281, Atmega2561 với các đặc điểm được tóm lược trên bảng sau:

Bảng 2: So sánh một số vi điều khiển họ AVR

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

Hình ảnh thực của Atmega2560:

Sơ đồ khối vi điều khiển này được cho ở hình sau:



Hình 7: Sơ đồ khối vi điều khiển ATmega2560

Để phát triển các ứng dụng giám sát, điều khiển có khả năng kết nối mạng, đặc biệt là mạng không dây thì ESP8266 là một lựa chọn thích hợp với nhiều ưu điểm nổi bật.

ESP8266 được phát triển bởi Espressif đặc biệt thích hợp cho các ứng dụng IoT (Internet of Thing). ESP8266 sử dụng bộ vi xử lý 32 bit L106 của Tensilica với kiến trúc RISC 16 bit, công suất tiêu thụ thấp. Tốc độ làm việc của vi xử lý lên đến 160MHz hỗ trợ hệ điều hành thời gian thực RTOS (Real-Time Operating System), kết nối Wifi.

Thông số của ESP8266:

3. Phần cứng:

3.1. CPU: Tensilica L106 32-bit processor

3.2. Giao tiếp ngoại vi: UART/SDIO/SPI/I2C/I2S/IR Remote Control, GPIO/ADC/PWM/LED Light & Button

3.3. Điện áp làm việc: 2,5V đến 3,6V

3.4. Dòng làm việc: 80mA

4. Wifi:

4.1. Hỗ trợ giao thức: 802.11 b/g/n

4.2. Tần số: 2.4G ~ 2.5G (2400M ~ 2483.5M)

4.3. Antena: on board hoặc loại lắp ngoài

5. Phần mềm:

5.1. Chế độ Wifi: Station/SoftAP/SoftAP+Station

5.2. Bảo mật: hỗ trợ WPA/WPA2

5.3. Mã hóa: WEP/TKIP/AES

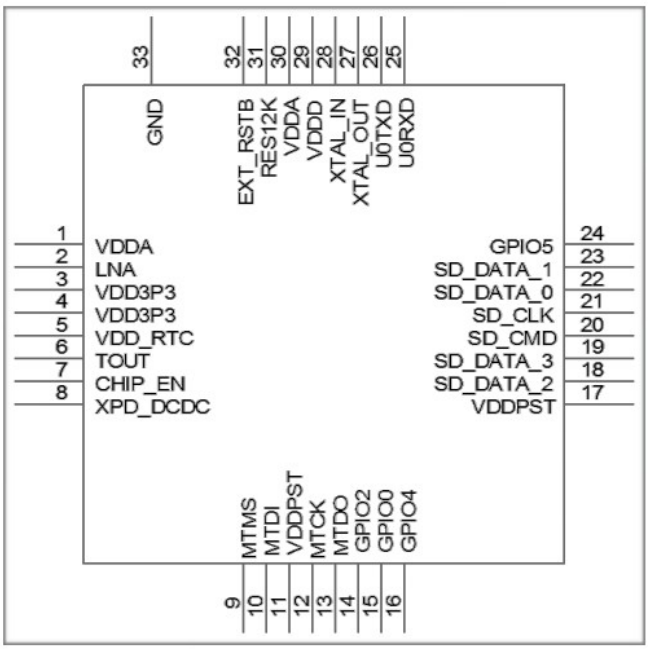
5.4. Cập nhật firmware: UART Download / OTA (via network)

5.5. Giao thức mạng: IPv4, TCP/UDP/HTTP/FTP

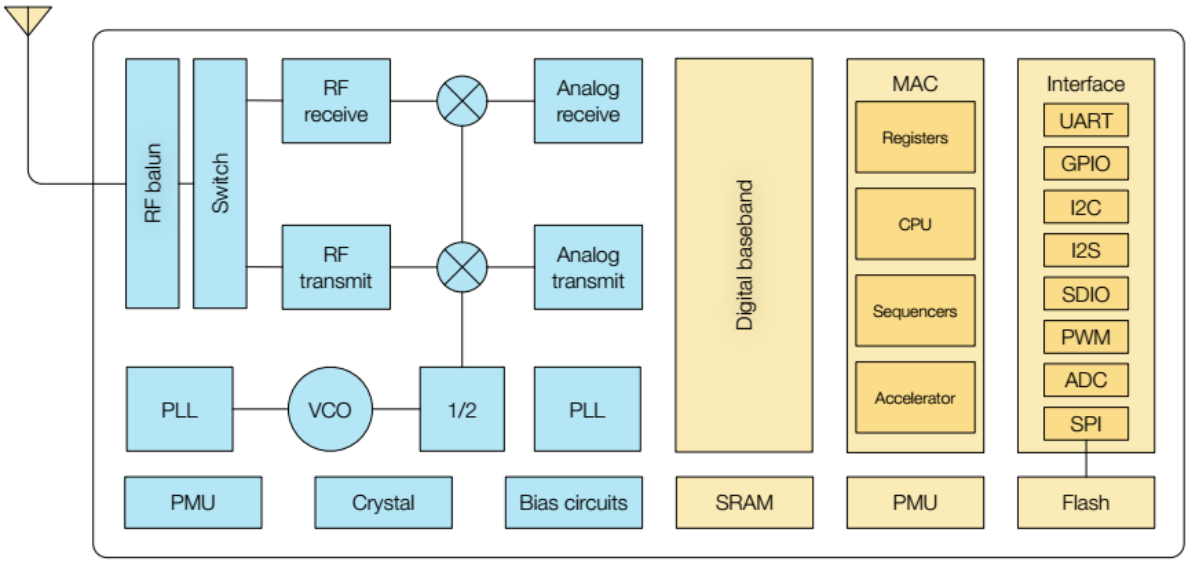
Hình ảnh thực của ESP8266 cho ở hình sau:

Hình 8: Hình ảnh thực của ESP8266

Sơ đồ chân và sơ đồ khối chức năng chính của ESP8266 cho ở các hình sau:



Hình 9: Sơ đồ chân của ESP8266



Hình 10: Sơ đồ khối của ESP8266

1.2.5. Giải pháp tích hợp hệ thống báo cháy tự động

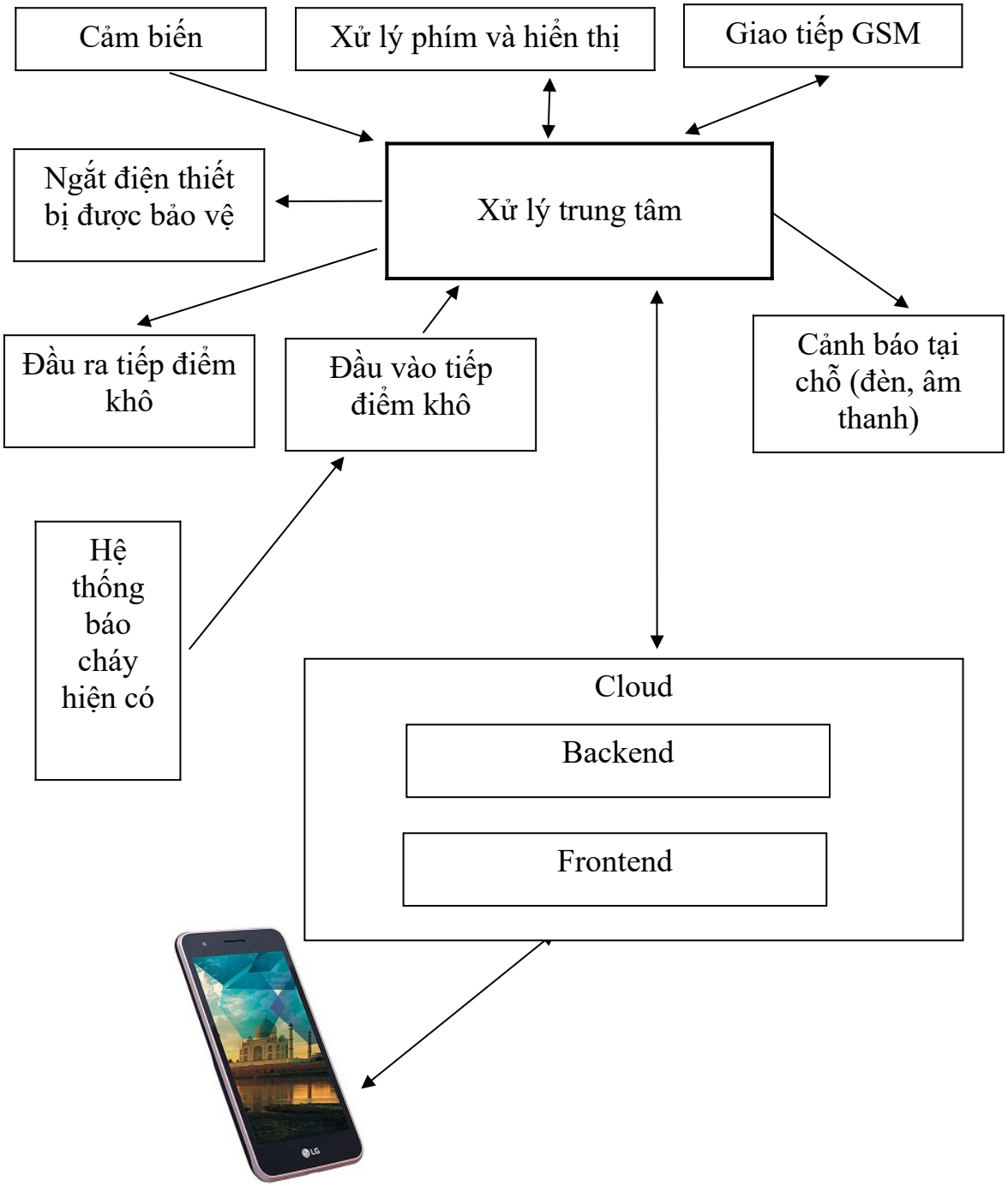
Các hệ thống báo cháy tự động ngoài các thiết bị ngoại vi được kết nối với các cổng giao tiếp chuẩn và dành riêng từ bộ điều khiển trung tâm còn sẵn có cổng mở rộng kết nối thông qua tiếp điểm khô (Dry Contact).

Khi có báo cháy, bộ điều khiển trung tâm hệ thống báo cháy sẽ xuất tín hiệu qua ghép nối tiếp điểm khô. Các thiết bị xử lý khác có thể ghép nối với bộ điều khiển trung tâm của hệ thống báo cháy tự động của giao tiếp này.

Giải pháp để tích hợp hệ thống báo cháy tự động mới với hệ thống báo

cháy hiện có nhằm phát huy vai trò hệ thống hiện có là sử dụng tiếp điểm khô.

Sơ đồ khối hệ thống như sau:



Hình 11: Sơ đồ khối hệ thống báo cháy tự động

Như vậy, cơ sở khoa học của việc "Nghiên cứu, ứng dụng tiến bộ khoa học công nghệ xây dựng hệ thống báo cháy tự động và điều khiển thiết bị từ xa qua mạng điện thoại di động tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên" chính là sự kết hợp giữa các ngành Điện – Điện tử và Viễn thông, sự phối hợp ứng dụng vi điều khiển hiện đại và hệ thống thông tin liên lạc đã hình thành một hướng nghiên cứu và phát triển không nhỏ trong khoa học kỹ thuật. Điều khiển thiết bị điện từ xa thông qua mạng điện thoại khắc phục được nhiều giới hạn trong hệ thống điều khiển từ xa và báo động thông thường. Hệ thống này không phụ thuộc vào khoảng cách, môi trường, đối tượng điều khiển và đối tượng báo động. Điểm đặc trưng nổi bậc của hệ thống là tính lưu động của tác nhân điều khiển (người điều khiển) và đối tượng được điều khiển là cố định.

1.3. THIẾT KẾ VÀ LẬP TRÌNH HỆ THỐNG

1.3.1. Tính toán hệ thống

1.3.1.1. Khối cảm biến đầu vào

Cảm biến đầu vào dùng trong hệ thống báo cháy tự động thông thường bao gồm cảm biến cháy theo nhiệt độ, cảm biến cháy theo khói.

- Cảm biến cháy theo nhiệt độ hoạt động dựa trên việc phát hiện sự gia tăng nhiệt độ và gửi tín hiệu báo cháy khi nhiệt đạt vượt ngưỡng giới hạn. Tín hiệu báo cháy được chuyển tới bộ xử lý trung tâm để thực hiện các khâu còn lại của quá trình xử lý báo cháy tự động.

- Cảm biến cháy theo khói hoạt động dựa trên việc phát hiện có khói. Có nhiều loại với cấu tạo khác nhau song được dùng phổ biến hiện nay là cảm biến khói hoạt động theo nguyên tắc quang học. Buồng phát hiện khói có đèn LED phát ánh sáng và mắt thu tín hiệu ánh sáng. Trong điều kiện bình thường (không có khói) ánh sáng được phát ra từ đèn LED sẽ truyền thẳng ra ngoài mà không tới được mắt thu (vì được đặt lệch nhau). Khi có khói vào buồng phát hiện khói, ánh sáng sẽ bị khuếch tán qua khói làm lệch hướng đi và chuyển tới mắt thu. Dựa trên tín hiệu anh sáng thu được cảm biến sẽ phát tín hiệu báo có khói trong phảm vi giám sát. Tin hiếu đó được chuyển tới bộ xử lý trong tâm để thực hiện các thủ tục xử lý tiếp theo.

Để đảm bảo tính tin cậy của các cảm biến, thông thường liên lạc giữa cảm biến với bộ xử lý trung tâm luôn được duy trì trong cả trường hợp hoạt động bình thường và khi có báo cháy.

Công nghệ thường được dùng để truyền tải thông tin và liên lạc giữa cảm biến với bộ xử lý trung tâm là sử dụng nguồn dòng. Theo phương pháp xử lý, truyền thông tin này, tín hiệu được chuyển đi dưới dạng biến thiên dòng điện thay vì hiệu điện thế như ở hệ thống dùng nguồn áp. Ưu điểm lớn của phương pháp truyền thông tin này là hạn chế được tác động của nhiễu lên tín hiệu trên đường truyển. Hơn nữa dễ dàng duy trình sự liên lạc thường xuyên giữa hai bên truyền nhận. Phương pháp thường được dùng để xác nhận liên lạc liên tục là ở điều kiện bình thường, hệ thống sẽ duy trì một dòng điện nhỏ. Nếu liên lạc giữa hai bên không bị gián đoạn, cable không bị đứt thì luôn có dòng duy trì này. Khi cần truyền dữ liệu, bên truyền sẽ làm tăng dòng điện trên đường truyền theo giao thức đã định. Từ biến thiên dòng điện trên đường truyền, bên nhận giải mã được thông tin truyền qua đó là gì.

Một số loại cảm biến báo cháy phát dựa trên nhiệt và khói được dùng trong nhiều hệ thống báo chạy tự động hiện nay có thể kể đến như:

Cảm biến báo cháy nhiệt AHR-871:

Hình 12: Đầu báo nhiệt AHR-871

Đặc điểm:

- Thiết bị có thể hoạt động không giới hạn số lần và không bị ảnh hưởng bởi nhiệt độ phòng. Các bộ phận được mạ vàng để tránh quá trình oxy hóa.
- Bộ phận cân bằng không khí làm bằng hợp kim để tránh độ ẩm và ổn định các chức năng thông gió. Hợp kim có thể tự tản nhiệt để tránh hiện tượng báo động giả.
- Thiết bị được cấu tạo hoàn toàn kín và không bị ảnh hưởng bởi độ ẩm, bụi bẩn hoặc côn trùng.
- Việc thiết kế, chế tạo được kiểm soát chất lượng nghiêm ngặt và thử nghiệm nhiều lần, lặp đi lặp lại tự động và được hiệu chỉnh tự động theo nhiệt độ và độ ẩm trong điều kiện ổn định. Vì vậy, luôn đảm bảo chất lượng và độ tin cậy trong quá trình sản xuất.

Thông số kỹ thuật:

- Model: AHR-871
- Xuất xứ: Horing Lih - Đài Loan
- Loại: 2 đây
- Contact báo động: không
- Điện áp: 12-30VDC
- Dòng báo động (24VDC 470Ω): 40mA
- Nhiệt độ báo động: Theo EN54, CNS
- Nhiệt độ môi trường xung quanh: 0-55^{o}C
- Vật liệu: Nhựa chống cháy
- Kích thước 102mm (đường kính) x 45mm (cao)

- Trọng lượng: 130g
- Màu: Trắng
- Diện tích bảo vệ:
- Chiều cao công trình dưới 4m: Công trình bằng vật liệu chống cháy 90m², vật liệu thường 50m$^{2.}$
- Chiều cao công trình từ 4-2m: Công trình bằng vật liệu chống cháy 45m², vật liệu thường 30m$^{2.}$

Cảm biến báo cháy khói AH-0311-2:

Hình 13: Đầu báo khói AH-0311-2

Đầu báo khói AH-0311-2 là đầu dò báo khói 2, báo động tại chỗ và truyền tín hiệu về thiết bị báo cháy trung tâm, để kích hoạt toàn bộ hệ thống còi báo động khi phát hiện có khói hay các dấu hiệu có cháy xảy ra để đưa ra cảnh báo một cách kịp thời.

Đặc điểm:

- Hộp cảm ứng được thiết kế đặc biệt để tránh bụi bặm xâm nhập, gia tăng độ tin cậy cho nhiệm vụ báo cháy.
- Sử dụng kỹ thuật tiên tiến, không bị nhiễu tín hiệu do đó tránh được hiện tượng báo động giả.
- Sử dụng 2 đèn LED, có thể quét được góc 360^{o}.
- Thiết bị có đầu ra rơ le, sử dụng nguồn riêng.
- Ngoài tính năng sử dụng để báo cháy, thiết bị còn có thể dùng để giám sát mất nguồn hoặc hở mạch, qua đầu ra rơ le.

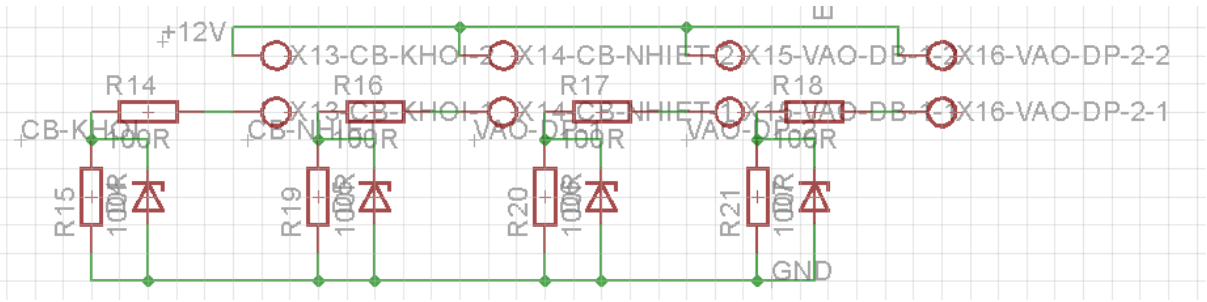
- Khi đầu báo khói kích hoạt, tín hiệu báo cháy sẽ truyền tới bộ xử lý trung tâm. Khi đầu báo khói được làm sạch, nó sẽ tự động reset về trạng thái giám sát bình thường.
- Thiết bị được tích hợp còi bên trong, nó sẽ kêu cảnh báo trước khi đưa ra báo động chính thức.
- Có dòng tiêu thụ thấp.
- Có tiếp điểm đôi trên đế đầu báo, giúp ổn định kết nối giữa đầu báo và đế đầu báo.
- Đế đầu báo được trang bị một lò xo có nhiệm vụ giúp không bị ngắn mạch khi tháo thiết bị ra để bảo trì hay sửa chữa.

Thông số kỹ thuật:

- Model: AH-0311-2
- Điện áp sử dụng: 12-30VDC
- Dòng dự phòng: 35μA
- Dòng cho phép: 120mA
- Dòng báo động (24VDC 470Ω): 30 mA
- Nhiệt độ môi trường: Từ -10oC đến 55oC.
- Vật liệu: Nhựa chống cháy
- Kích thước: 102mm x 47mm
- Trọng lượng: 155g
- Màu: Trắng ngà
- Diện tích bảo vệ:
- Chiều cao công trình dưới 4m: 150m²
- Chiều cao công trình từ 4-2m: 75m²

Hệ thống báo chạy tự động được thiết kế khối đầu vào cảm biến phù hợp để kết nối được hai loại cảm biến trên (và có khả năng tương thích với nhiều loại cảm biến khác).

Các cảm biến trên đều hoạt động dựa trên nguyên tắc truyền tín hiệu nguồn dòng có độ tin cậy cao. Ở chế độ giám sát, các cảm biến tiêu thụ dòng đều rất nhỏ (vài chục μA). Ở chế độ báo động, dòng điều khiển lên vài chụ mA.



Hình 14: Sơ đồ mạch xử lý khối cảm biến đầu vào

Mạch xử lý khối cảm biến đầu vào được thiết kế làm việc với điện áp nguồn 12V cho dòng điều khiển lên đến 40mA với điện áp đưa vào bộ xử lý được khống chế không quá giới hạn chuẩn đầu vào 5V.

Mạch bao gồm 4 kênh giao tiếp nguồn dòng. Kết quả thử nghiệm làm việc hiệu quả với các cảm biến khói và nhiệt kiểu nguồn dòng có điện áp làm việc từ 12 đến 24VDC.

1.3.1.2. Khối kết nối thuê bao để gửi tin nhắn, gọi điện

Một chức năng quan trọng của hệ thống báo cháy được thiết kế là hỗ trợ cảnh báo qua tin nhắn (SMS) và thực hiện cuộc gọi tới số điện thoại đã được cài đặt sẵn.

Để thực hiện chức năng cảnh báo qua tin nhắn và gọi điện, hệ thống sử dụng module giao tiếp GSM SIM800L mini. Đây là module hỗ trợ giao tiếp mạng điện thoại di động có kích thước nhỏ gọn, thích hợp trong các ứng dụng điều khiển, giám sát từ xa qua mạng GSM.

Hình 15: Module GSM SIM800L

Thông số kỹ thuật:

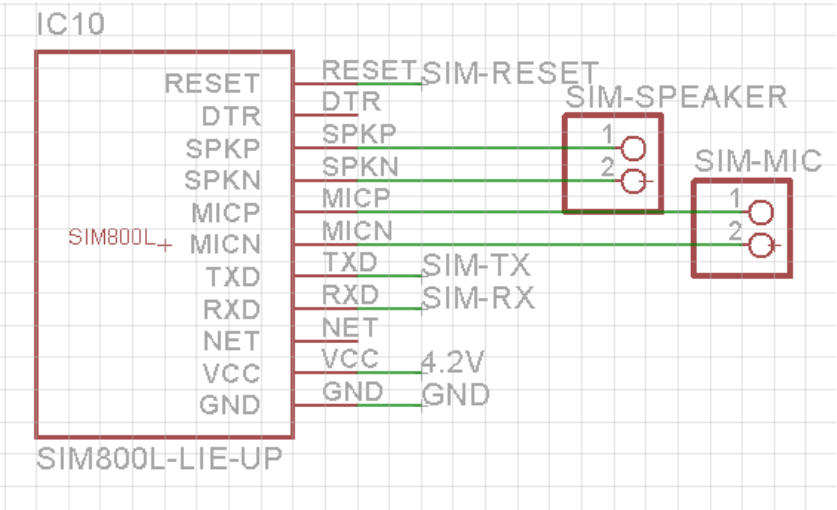
- Nguồn cấp: 4.2VDC-1A
- Khe cắm SIM : MICROSIM
- Dòng khi ở chế độ chờ: 10 mA
- Dòng khi hoạt động: 100 mA đến 1A.
- Hỗ trợ 4 băng tần phổ biến.
- Kích thước: 25mm x 22cm.

Chức năng các chân:

- VCC: Nguồn vào 4.2V.
- TXD: Chân truyền Uart TX.
- RXD: Chân nhận Uart RX.
- DTR : Chân UART DTR
- SPKP, SPKN: ngõ ra âm thanh, nối với loa để phát âm thanh.
- MICP, MICN: ngõ vao âm thanh, phải gắn thêm Micro để thu âm thanh.
- Reset: Chân khởi động lại Sim800L.
- RING : báo có cuộc gọi đến.
- GND: Chân Mass, cấp 0V.

Module SIM800L giao tiếp với các bộ xử lý thông qua chuẩn truyền thông UART sử dụng các lệnh AT để điền khiển.

Mạch xử lý khối giao tiếp thuê bao di động nhắn tin gọi điện sử dụng module SIM800L như sau:



Hình 16: Mạch giao tiếp module GSM SIM800L

1.3.1.3. Khối giải mã, hiển thị và giao tiếp phím nhấn

Hệ thống được trang bị màn hình hiển thị LCD 20x4 (20 ký tự x 4 dòng) có kích thước phù hợp để hiện thị thông tin một cách đầy đủ, rõ ràng thay vì chỉ là các đèn báo (LED) thường thấy ở một số hệ thống báo chạy tự động đơn giản khác.

Người dùng giao tiếp với hệ thống qua bộ 4 phím nhấn tích hợp các tính năng linh hoạt. Thông qua phím bấm, người dùng có thể điều khiển lựa chọn chức năng điều khiển trên menu hiển thị trên màn hình LCD cũng như nhập liệu cả dưới dạng số và chữ.

Đặc điểm kỹ thuật của LCD 20x4 được sử dụng trong hệ thống:

Hình 17: LCD 20x4 tích hợp đèn nền

Thông số kỹ thuật:

- Điện áp hoạt động: 5V
- Hiển thị tối đa 20 tự trên 4 dòng.
- Chữ đen nền xanh lá.
- Chức năng các chân trên LCD:
- Chân 1: (VSS) Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển.
- Chân 2: (VDD) Là chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC = 5V của mạch điều khiển.
- Chân 3: (V0) là chân điều chỉnh độ tương phản của LCD.
- Chân 4: (RS) Là chân chọn thanh ghi (Register select).

- Chân 5: (R/W) là chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
- Chân 6: (E) Là chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E.
- Chân 7 - 14: (DB0 - DB7) - Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU.
- Chân 15: Nguồn dương cho đèn nền.
- Chân 16: GND cho đèn nền.

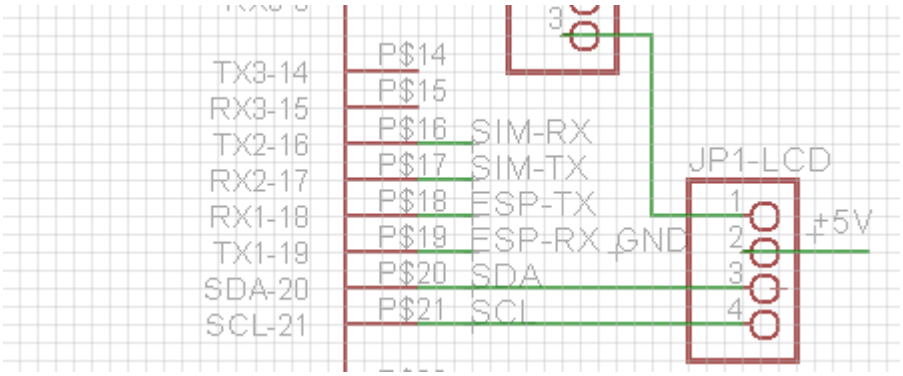
LCD được thiết kế gắn trên cánh tủ điều khiển có khoảng cách xa tới mạch điều khiển trung tâm. Vì vậy, mạch chuyển đổi giao tiếp song song sang nối tiếp sử dụng chuẩn I2C đã dùng thay cho phương pháp truyền song song.

Với giao tiếp chuẩn của LCD 20x4, cần tới sợi cable truyền có 16 sợi. Như vậy kích thước cable lớn, ảnh hưởng tới thẩm mĩ và công tác đi dây, đấu nối. Mạch chuyển đổi I2C rút gọn sợi cable chỉ còn 4 dây với kích thước nhỏ gọn, làm việc hiệu quả.

Mạch chuyển đổi giao tiếp sử dụng chuẩn truyền thông I2C để làm việc với LCD 20x4:

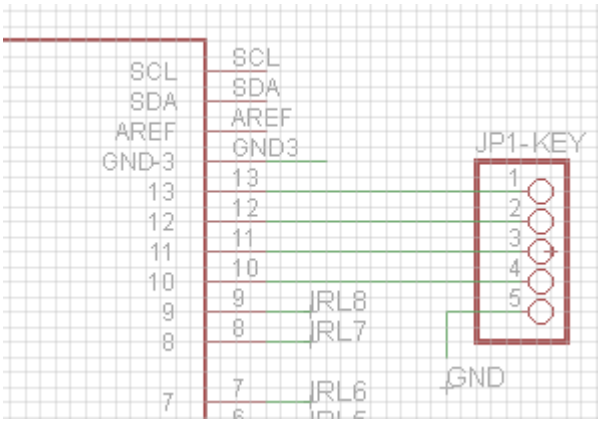
Hình 18: Mạch chuyển đổi I2C cho LCD 20x4

Sơ đồ mạch ghép nối LCD với bộ xử lý trung tâm:



Hình 19: Sơ đồ mạch ghép nối LCD 20x4 với bộ xử lý trung thâm

Bàn phím bao gồm 4 phím được kết nối trực tiếp với 4 chân vào ra đa chức năng của bộ xử lý trung tâm:



Hình 20: Sơ đồ mạch ghép nối bàn phím với bộ xử lý trung thâm

1.3.1.4. Khối điều khiển động lực đóng ngắt tiếp điểm khô

Tiếp điểm khô thường được dùng để giao tiếp giữa các hệ thống có thông tin truyền tải đơn giản dưới dạng ON/OFF. Đây là phương pháp ghép nối đơn giản nhưng hiệu quả.

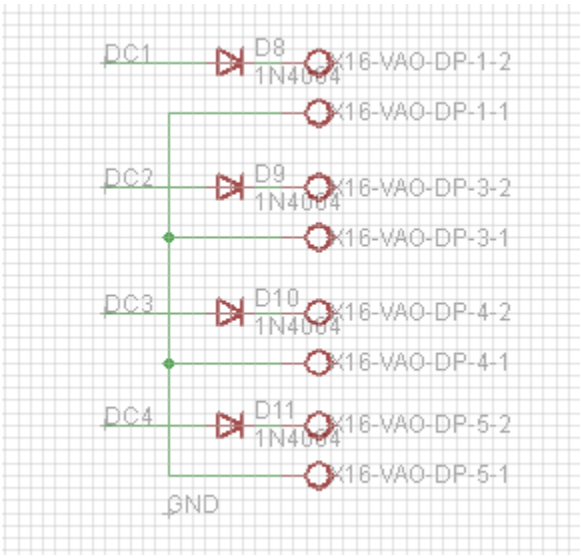
Với hệ thống báo cháy, thông tin sự kiện cháy có thể được xuất ra thông qua các cặp tiếp điểm rơ le dạng thường đóng hoặc mở. Tiếp điểm dạng này có thể dùng để bật đèn hay chuông báo cháy trực tiếp. Ngoài ra, thông qua việc đóng hoặc mở tiếp điểm, thiết bị khác có thể nhận được thông tin sự kiện từ thiết bị xử lý hiện hành.

Hệ thống báo chạy tự động mới được thiết kế để có thể nhận được các thông báo sự kiện từ các hệ thống khác gửi sang qua các tiếp điểm khô như vậy.

Mạch xử lý là đầu vào số được treo sẵn điện trở trong có mức logic cao. Khi tiếp điểm khô kích hoạt đóng sẽ làm thay đổi mức logic từ cao xuống thấp qua đo báo cho bộ xử lý trung tâm xử lý.

Với sự trang bị đầu vào tiếp điểm khô, hệ thống báo cháy tự động mới có thể kết nối với các hệ thống báo chạy hiện có khác để gia tăng hiệu quả báo cháy và tận dụng được tài nguyên hiện có mà không phải thay thế, lãng phí đầu tư từ đầu mọi thứ.

Mạch xử lý tín hiệu đầu vào tiếp điểm khô như sau:



Hình 21: Mạch xử lý đầu vào tiếp điểm khô

Hệ thống được trang bị sẵn 4 đầu vào tiếp điểm kho cho phép tiếp nhận 4 kênh thông tin độc lập. Mạch trang bị diode chống xông điện áp ngoài vào làm hỏng cổng xử lý. Các đường DC1 đến DC4 đều được treo dương nguồn bằng điện trở trong bộ xử lý.

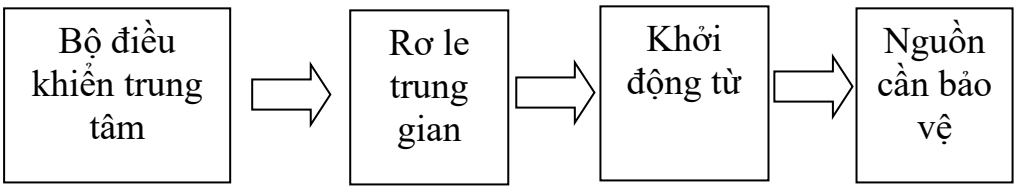
1.3.1.5. Khối điều khiển ngắt điện thiết bị cần được bảo vệ

Một trong các chức năng ưu việt của hệ thống báo chạy tự động mới này là khả năng điều khiển ngắt điện tự động khi có sự kiện cháy hoặc được điều khiển từ xa qua phần mềm cài đặt trên thiết bị thông minh như smart phone.

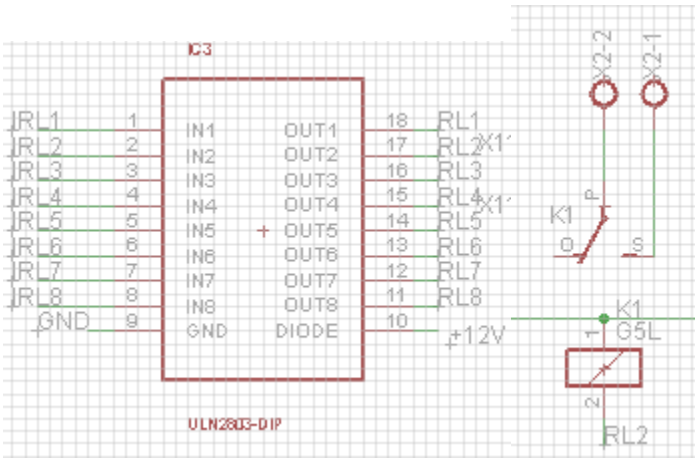
Việc điều khiển đóng ngắt điện đòi hỏi thiết bị đóng ngắt phải có công suất đóng ngắt lớn. Dòng đóng ngắt có thể lên đến hàng chục A. Để đáp ứng được yêu cầu này, tín hiệu điều khiển đóng ngắt ở mức logic (TTL) sẽ được đưa đến điều khiển một rơ le trung gian. Rơ le trung gian chỉ cần nguồn điều khiển có công suất nhỏ. Tiếp điểm của rơ le cho phép đóng dòng điện lên đến vài A thâm chí là cả chục A. Tiếp theo, sử dụng rơ le trung gian điều khiển đóng ngắt

nguồn điện cấp cho cuộn hút của khởi động từ. Khởi động từ là thiết bị đóng ngắt chuyên dụng được thiết kế với công suất đóng ngắt cao. Dòng đóng ngắt lên đến hàng chụ A và cao hơn. Khởi động từ thường được ứng dụng trong điều khiển nguồn cấp cho các động cơ ba pha công suất lớn (hàng KW) cũng như các nguồn chính cấp cho cả tòa nhà, văn phòng.

Sơ đồi khối điều khiển ngắt điện nguồn cần bảo vệ:



Hình 22: Khối ngắt điện nguồn cần bảo vệ



Hình 23: Mạch điều khiển rơ le trung gian tới khởi động từ

1.3.1.6. Khối cảnh báo tại chỗ bằng tín hiệu đèn và âm thanh

Hệ thống báo cháy tự động được trang bị sẵn đầu ra điều khiển trực tiếp các thiết bị báo cháy tại chỗ như đèn báo động, còi hú (chuông).

Để đảm bảo tính tương thích, dải điện áp làm việc rộng, phù hợp cho các loại đèn báo, còi hú hay chuông báo động khác nhau, khối điều khiển cảnh báo thiết kế đóng ngắt bằng rơ le.

Tùy thiết bị đầu cuối sử dụng nguồn gì mà người dùng có thể lựa chọn cách đấu nối cho phù hợp.

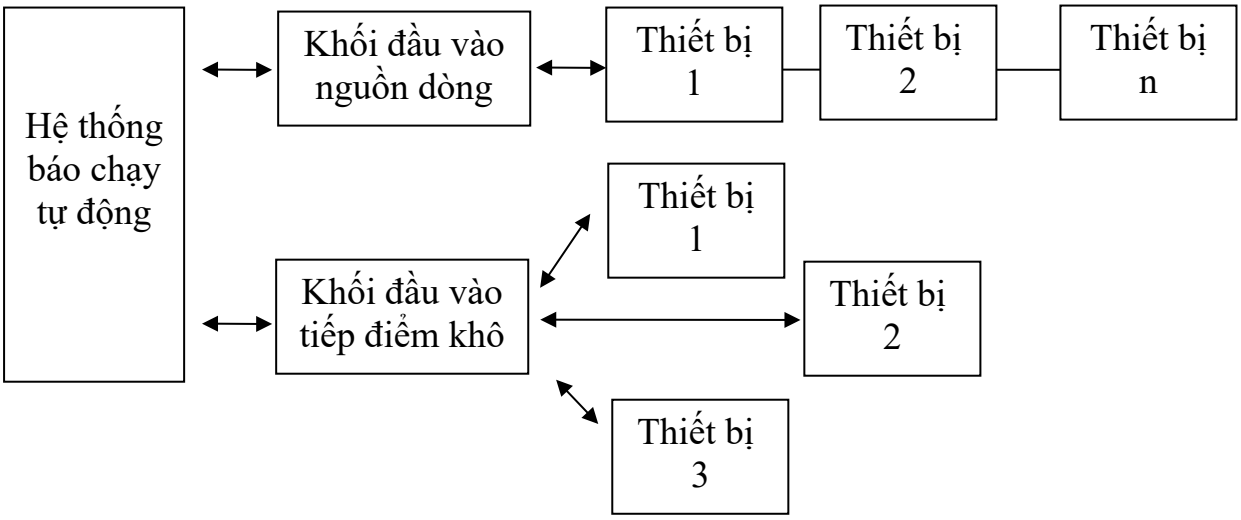
Hệ thống được trang bị sẵn 4 đầu ra là tiếp điểm rơ le với dòng đóng ngắt tối đa đến 10A.

1.3.1.7. Khối kết nối với hệ thống báo cháy có sẵn

Hệ thống báo cháy tự động đang xây dựng trang bị sẵn hai loại đầu vào sẵn sàng cho kết nối, mở rộng, kế thừa từ các hệ thống khác.

Các đầu vào này được thiết kế theo hai chuẩn đó là nguồn dòng và nguồn áp. Với hệ thống khác sử dụng phương pháp kết nối nguồn dòng, người dùng có thể đấu nối với hệ thống hiện tại qua các kênh vào nguồn dòng (có sẵn 4 kênh). Tương tự vật, nếu thiết bị khác sử dụng nguồn áp hay tiếp điểm khô thì có thể kết nối tới hệ thống hiện tại qua đầu vào tiếp điểm khô.

Sơ đồ đấu nối thể hiện như sau:



Hình 24: Sơ đồ đấu nối các hệ thống báo cháy hiện có

1.3.1.8. Khối giao tiếp phần mềm giám sát

Điểm vượt trội hệ thống báo cháy mới này là hỗ trợ đẩy dữ liệu lên cloud. Qua đó kết nối phần mềm giám sát chạy trên các thiết bị thông minh để cung cấp một cách chủ động, thường xuyên các số liệu hoạt động của hệ thống.

Kết nối cloud không chỉ dùng để giám sát hệ thống, nắm bắt diễn tiến cũng như tiếp nhận các sự kiện. Thông qua kết nối này, người dùng còn có thể điều khiển từ xa hệ thống báo cháy, trả lời các sự kiện, thậm chí là ngắt điện nguồn thiết bị cần bảo vệ.

Nền tảng triển khai giao tiếp cloud là phần cứng kết nối mạng. Hệ thống được thiết kế trang bị sẵn module giao tiếp không dây qua mạng Wifi chuẩn 802.11n.

Hệ thống sử dụng module kết nối Wifi ESP8266 đang được ưa chuộng trong nghiên cứu, phát triển các sản phẩm IoT cho phép thiết lập kết nối mạng hiệu quả với kết cấu nhỏ gọn.

ESP8266 trang bị bộ vi xử lý 32 bit mạnh mẽ cho phép triển khai được nhiều chương trình xử lý tích hợp sẵn trong thiết bị với hiệu suất cao.

Hình 25: Module ESP8266 trang bị kết nối mạng Wifi

Đặc điểm:

- Module ESP8266 sử dụng IC Wifi SoC ESP8266 của hãng Espressif, được sử dụng để kết nối với vi điều khiển thực hiện chức năng truyền nhận dữ liệu qua Wifi.
- Module có thiết kế nhỏ gọn, sử dụng giao tiếp UART được sử dụng rất nhiều trong các ứng dụng IOT.
- Thông số kỹ thuật:
- Chipset: Wifi SoC ESP8266
- Điện áp sử dụng: 3.3VDC
- Điện áp giao tiếp: 3.3VDC
- Dòng tiêu thụ: Max 320mA
- Hỗ trợ chuẩn 802.11 b/g/n.
- Wi-Fi 2.4 GHz, hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK
- Hỗ trợ cả 2 giao tiếp TCP và UDP.
- Chuẩn giao tiếp UART với Firmware hỗ trợ bộ tập lệnh AT Command, tốc độ Baudrate mặc định 9600 hoặc 115200.
- Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.
- Kích thước: 24.8 x 14.3mm.

Chức năng chân:

- VCC: 3.3V , dòng có thể lên 300mA.
- GND: 0V
- Tx: Chân Tx của giao thức UART, kết nối đến chân Rx của vi điều khiển.
- Rx: Chân Rx của giao thức UART, kết nối đến chân Tx của vi điều khiển.
- RST: chân reset, kéo xuống mass để reset.
- CH_PD: chân này nếu được kéo lên mức cao module sẽ bắt đầu thu phát wifi, kéo xuống mức thấp module dừng phát wifi. Vì ESP8266 khởi động hút dòng lớn nên chúng ta giữ chân này ở mức 0V khi khởi động hệ thống của mình, sau 2s hãy kéo chân CH_PD lên 3.3V, để đảm bảo module hoạt động ổn định.
- GPIO0: kéo xuống thấp cho chế độ upgrade firmware.
- GPIO2: không sử dụng.

Ngoài kết nối mạng theo chuẩn không dây, hệ thống báo cháy tự động còn sẵn sàng cho việc mở rộng kết nối với hỗ trợ giao tiếp mạng có dây Ethernet. Tính năng này có thể được mở rộng bằng cách lắp thêm module giao tiếp Ethernet sử dụng chuẩn truyền thông SPI giao tiếp với bộ xử lý trung tâm (module ENC28J60).

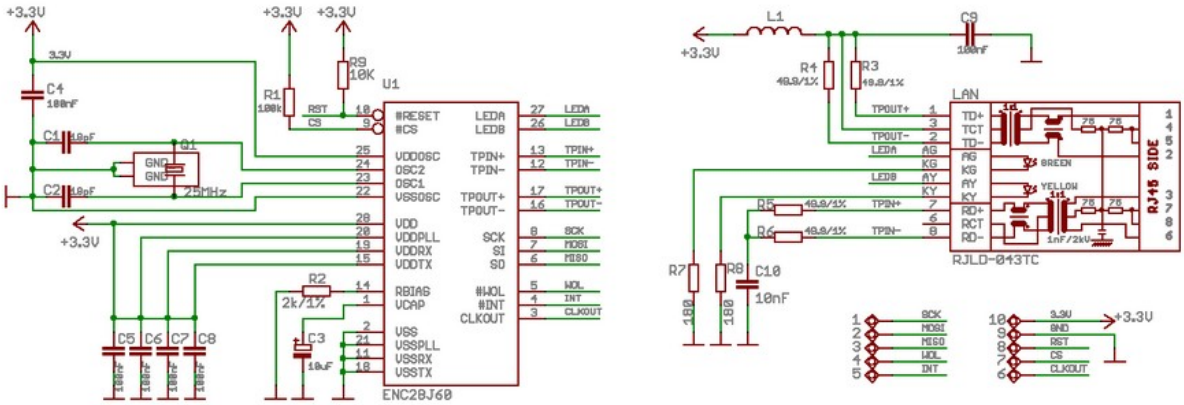
Main board chính của bộ xử lý trung tâm đã thiết kế sẵn khe cắn cho việc mở rộng này.

Đặc điểm của module kết nối mạng chuẩn Ethernet sử dụng ENC28J60:

Hình 26: Module kết nối mạng Ethernet

Sử dụng chip ENC28J60, điện áp làm việc 3.3, ghép nối truyền thông chuẩn SPI.

Sơ đồ mạch:



Hình 27: Sơ đồ mạch module kết nối mạng Ethernet

1.3.1.9. Khối backend xử lý dữ liệu trên cloud

Khối backend xử lý dữ liệu trên cloud có nhiệm vụ tiếp nhận dữ liệu trạng thái vận hành của hệ thống báo cháy tự động gửi lên từ bộ xử lý trung tâm qua giao tiếp mạng không dây (Wifi) hoặc mạng có dây (Ethernet).

Dự liệu được tiền xử lý, lưu trữ lại trên máy chủ động thời gửi cho các ứng dụng di động để giám sát quá trình vận hành của hệ thống.

Khối backend không có thể hiện giao diện trực tiếp với người dùng mà đều thông qua khối frontend, các ứng dụng di động.

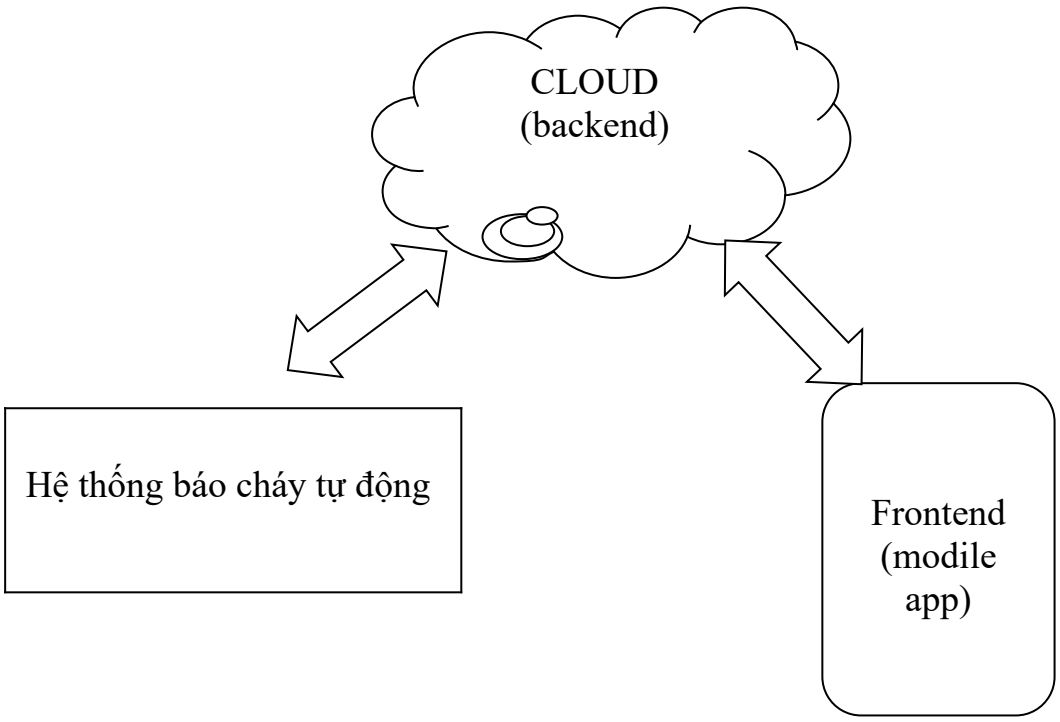
1.3.1.10. Khối frontend phần mềm giám sát chạy trên điện thoại thông minh

Khối này có chức năng tạo ra giao diện người dùng, thể hiện bằng phần mềm chạy trên máy tính hoặc điện thoại di động.

Chức năng chính:

- Hiển thị trạng thái vận hành hệ thống báo cháy, trạng thái các cảm biến kết nối đến hệ thống đang bình thường hay có sự cố kết nối.
- Thể hiện các số đo công suất tiêu thụ điện của phần nguồn điện chính được giám sát. Thông qua các số liệu đo có thể kịp thời đưa ra các cảnh báo nguy cơ xảy ra cháy do sự cố điện (quá tải).
- Cung cấp công cụ cho phép người dùng điều chỉnh chế độ vận hành của hệ thống, điều khiển việc đóng, ngắt nguồn điện được bảo vệ.

Mô hình giao tiếp khối frontend tới hệ thống:



Hình 28: Sơ đồ ghép nối fontend với backend và hệ thống báo cháy tự động

1.3.1.11. Khối xử lý trung tâm

Khối xử lý trung tâm là phần cốt lõi của hệ thống báo cháy tự động. Vai trò thông thường của hệ thống xử lý trung tâm:

- Tiếp nhận tín hiệu báo cháy từ hệ thống cảm biến
- Xử lý tránh báo giả
- Thực hiện cảnh báo tại chỗ (thông quan đèn, còi, chuông) khi xác định được chính xác sự kiện cháy.
- Điều khiển thiết bị đầu cuối như xả khí chữa cháy, bơm nước dập lửa…

Với hệ thống báo cháy mới, ngoài thực hiện các cảnh báo tạo chỗ, điều khiển thiết bị đầu cuối, bộ xử lý trung tâm còn đảm nhiệm chức năng quan trọng khác là giao tiếp với phần mềm giám sát điều khiển qua cloud. Tiếp nhận điều khiển tử xa để đóng/ngắt nguồn điện cần bảo vệ, thay đổi chế độ vận hành. Tiếp nhận tín hiệu báo cháy từ hệ thống báo chạy hiện có khác nhằm tối đa hóa hiệu quả phòng vào chữa cháy, tận dụng được các hệ thống hiện có mà không phải thay thế hoàn toàn.

Toàn bộ hệ thống xử lý trung tâm được tích hợp từ các khối chức năng trên một board mạch duy nhất có kích thước nhỏ gọn, đảm bảo đầy đủ các chức năng theo thiết kế.

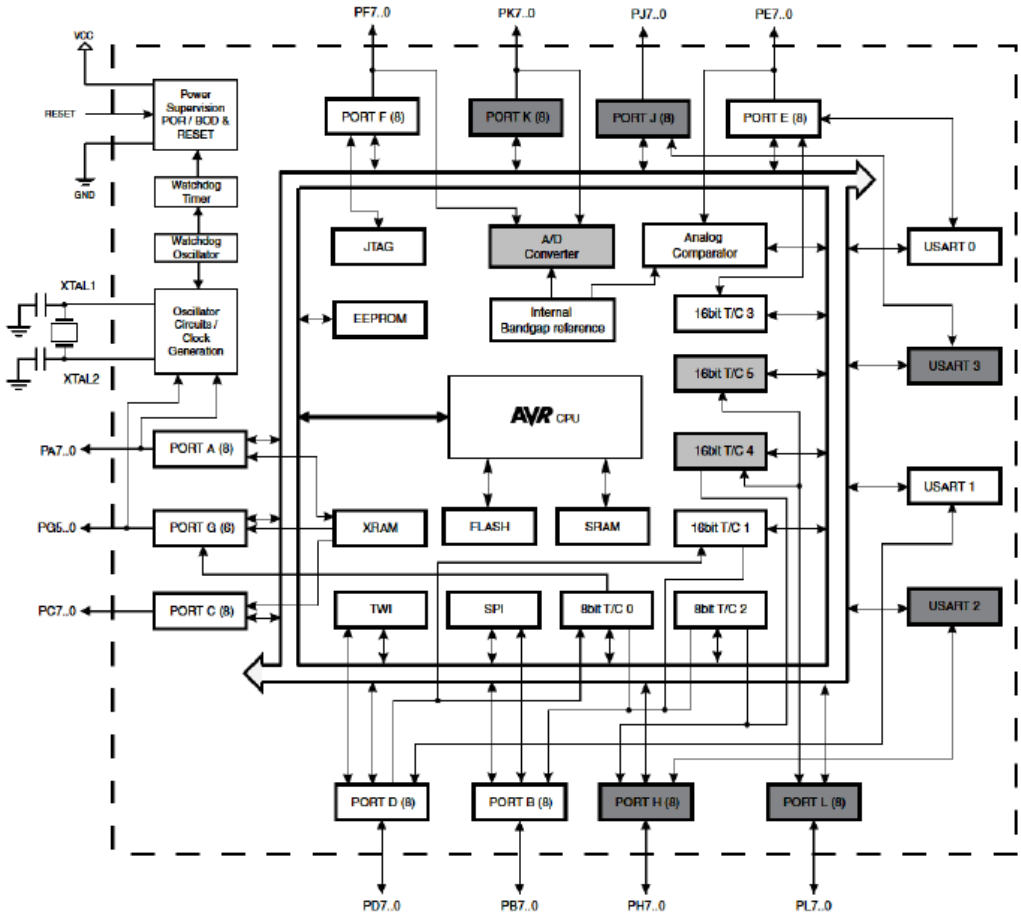
a) Phần cứng bộ xử lý trung tâm (board mạch chủ)

Khối xử lý trung tâm sử dụng vi điều khiển dòng AVR với chip xử lý chính là Atmega 2560.

Hình 29: Chip ATmega 2560

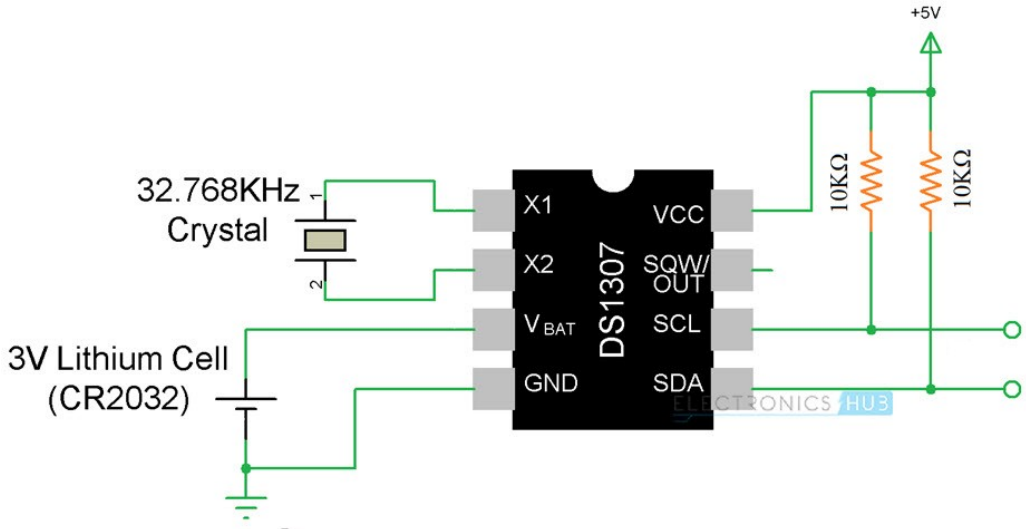
Thông số kỹ thuật:

- Vi điều khiển: ATmega2560
- IC nạp và giao tiếp UART: ATmega16U2.
- Điện áp hoạt động: 5V
- Điện áp đầu vào (Đề nghị): 7 – 12V
- Điện áp đầu vào (Giới hạn): 6 – 20V
- Số lượng chân I/O: 54 chân, trong đó có 15 chân PWM.
- Số lượng chân analog: 6 chân
- Dòng điện DC trên mỗi I/O: 20mA
- Bộ nhớ flash: 256KB
- SRAM: 8KB
- EEPROM: 4KB
- Xung clock: 16MHz
- Số lượng UART: 4 cổng
- Số lượng SPI: 1
- Số lượng I2C: 1
- Kích thước: 101,52mm x 53,3mm
- Trọng lượng: 37g

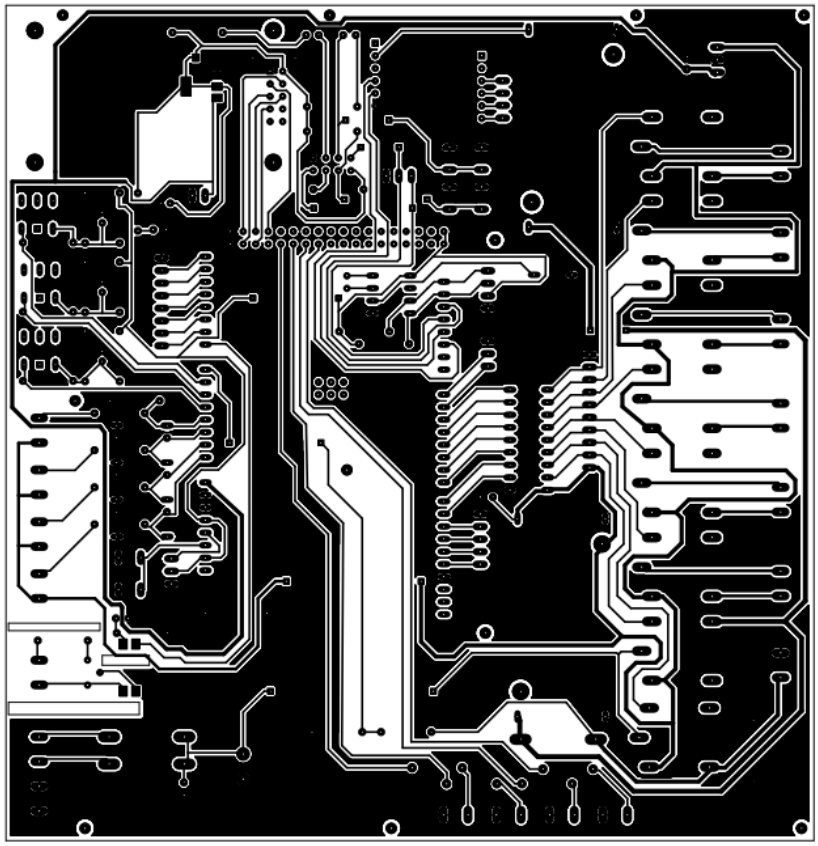


Hình 30: Sơ đồ khối chức năng chính ATmega 2560

Chip xử lý đồng hồ thời gian thực là DS1307 được trang bị pin cấp nguồn riêng loại 3V mã hiệu CR1220.

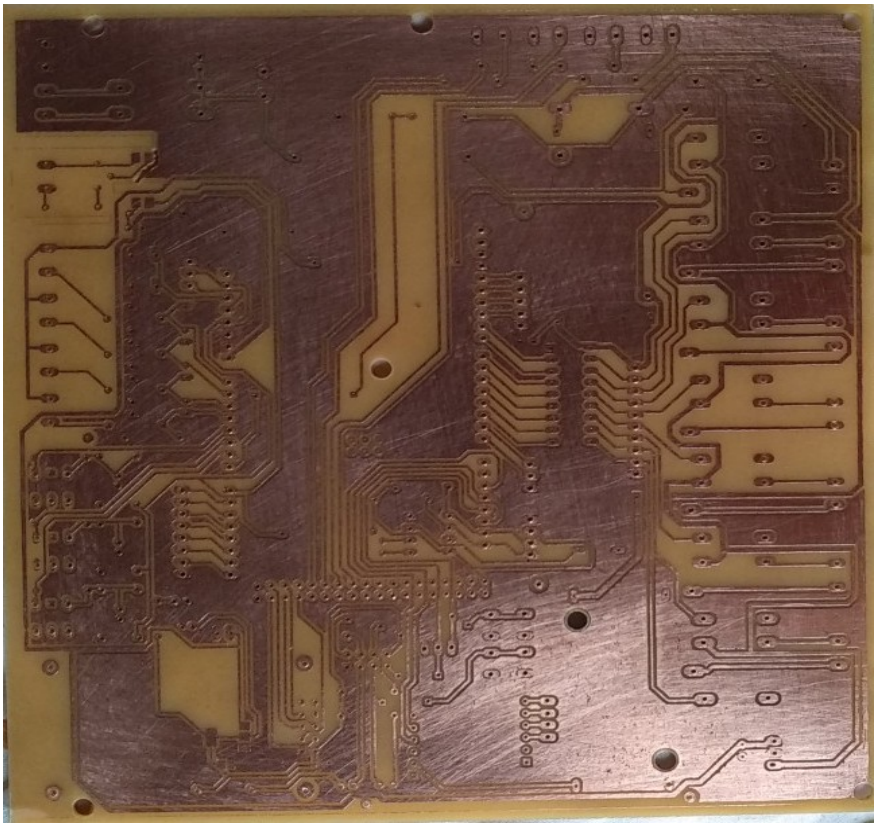


Hình 31: Sơ đồ mạch DS1307



Hình 32: Thiết kế board mạch bộ xử lý trung tâm

Ảnh chụp board mạch xử lý trung tâm chế tạo xong (chưa lắp ráp linh kiện).



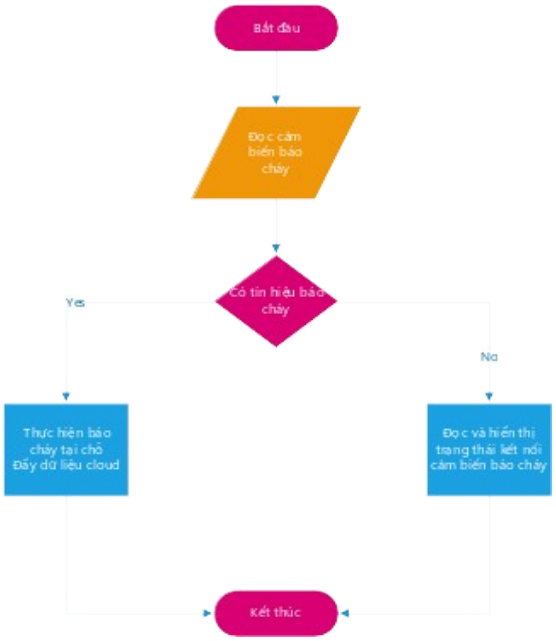
Hình 33: Board mạch bộ xử lý trung tâm

1.3.2. LẬP TRÌNH HỆ THỐNG

1.3.2.1. Lập trình bộ xử lý trung tâm

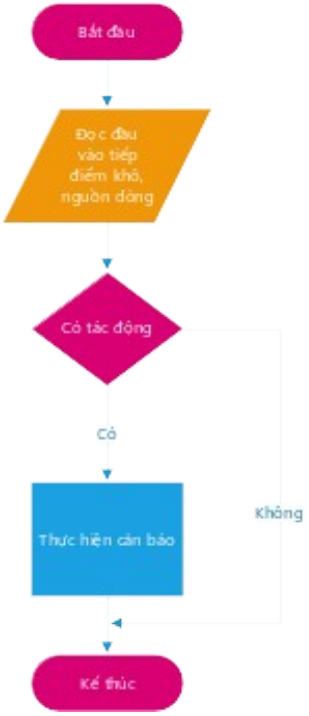
Lưu đồ thuật toán khối xử lý cảm biến báo cháy:

Khối này có niệm vụ đọc dữ liệu từ các cảm biến theo thiết lập kênh giao tiếp.



Hình 34: Lưu đồ thuật toán Khối xử lý cảm biến

Lưu đồ thuật toán khối xử lý tín hiệu báo cháy từ hệ thống khác thông qua tiếp điểm khô, kênh nguồn dòng:



Hình 35: Lưu đồ thuật toán Khối xử lý đầu vào mở rộng

Khối tương tác người dùng: khối này có nhiệm vụ đọc và giải mã bàn phím, kích hoạt đèn nền LCD. Hiển menu cho phép người dùng chọn lệnh trên menu để đưa ra quyết định xử lý.

Hình 36: Lưu đồ thuật toán Khối xử lý đầu bàn phím

Hình 37: Lưu đồ thuật toán khối xử lý chính

Mã chương trình khối xử lý trung tâm:

```
        //------------------------------------------------------------
        // Setup
        //------------------------------------------------------------
        void setup() {
          SetPinConfig();
          SerialForWifi.begin(9600);
          SerialForWifi.println("Start");
          InitDefaultValue();
          if (ReadConfig() == 0) WriteConfig();
          Wire.begin();

          lcd.SetAddr(LCDAddr);
          lcd.begin();
          lcd.backlight();
          lcd.createChar(DegreeChar, degree);
          lcd.createChar(EnterChar, Enter);
          lcd.createChar(SelectChar, SelectSign);
          lcd.createChar(LogoChar, LogoSign);
          lcd.createChar(ClockChar, ClockSign);
          lcd.createChar(CalendarChar, CalendarSign);

          lcd.clear();
          lcd.setCursor(0, 0);
          ShowTitle();
          lcd.setCursor(0, 1);
          lcd.print("GSM init...");
          SerialForSIM800L.begin(9600);
          InitSIM();
          lcd.print("DONE");
          delay(1000);
          ShowScreen();
        }
        //
=============================================================
===
        // Khoi tao cac gia tri ban dau
        //
=============================================================
===
        void SetPinConfig()
        {
```

```
        int i;
        for (i = 0; i < OutputCount; i++)
        {
          pinMode(OutputPin[i], OUTPUT);
          digitalWrite(OutputPin[i], LOW);
        }
        for (i = 0; i < InputCount; i++)
        {
          pinMode(InputPin[i], INPUT_PULLUP);
        }
        pinMode(FlashLightPort, OUTPUT);

      }
      //
=================================================================
===
      // Khoi tao cac gia tri ban dau
      //
=================================================================
===
      void InitDefaultValue()
      {
        //Cau hinh co ban thiet bi phan cung
        Password = F("0");
        //Cau hinh du lieu ung dung
        GSMSupport = 1;//Canh bao qua tin nhan
        TempCalibration = 0;
        HumiCalibration = 0;
        UpdateTime = default_UpdateTime;
        DS1307Addr = default_DS1307Address;
        LCDAddr = 0x3F;
        ForwardGSMData = 0;
      }

      //
=================================================================
===
      //Restore all seting by Default Factory
      //
=================================================================
===
      void RestoreFactory()
      {
```

```
          InitDefaultValue();
          WriteConfig();
        }
        //
=================================================================
===
        int ReadConfig()
        {
          if ((EEPROM.read(0) == RomID1) && (EEPROM.read(1) == RomID2))
          {
            RomConfigued = 1; //ROM saved config data
            addr = 2;
            //Doc so dien thoai
            Password = ReadROMItem("");
            GSMSupport = ReadROMItem("").toInt();
            TempCalibration = ReadROMItem("").toFloat();
            HumiCalibration = ReadROMItem("").toFloat();
            UpdateTime = ReadROMItem("").toInt();
            DS1307Addr = ReadROMItem("").toInt();
            LCDAddr = ReadROMItem("").toInt();
            ForwardGSMData = ReadROMItem("").toInt();

            EEPROM.end();
            return 1;
          }
          else
          {
            EEPROM.end();
            return 0;
          }
        }

        //
=================================================================
===
        int WriteConfig()
        {
          addr = 0;
          //Write ROM indicator
          EEPROM.write(addr++, RomID1);
          EEPROM.write(addr++, RomID2);
          //Write PhoneNumberCount
          WriteROMItem(Password, "");
```

```
          WriteROMItem(String(GSMSupport), "");
          WriteROMItem(String(TempCalibration), "");
          WriteROMItem(String(HumiCalibration), "");
          WriteROMItem(String(UpdateTime), "");
          WriteROMItem(String(DS1307Addr), "");
          WriteROMItem(String(LCDAddr), "");
          WriteROMItem(String(ForwardGSMData), "");

          EEPROM.end();
        }

        //
================================================================
===
        String ReadROMItem(String What)
        {
          String s = "";
          int i;
          int n = EEPROM.read(addr++);
          for (i = 0; i < n; i++)
            s += (char)EEPROM.read(addr++);
        #ifdef DEBUGING
          Serial.print(What);
          Serial.println(s);
        #endif
          return s;
        }
        //
================================================================
===
        void WriteROMItem(String s, String What)
        {
          int i;
          byte d;
          d = s.length();
          EEPROM.write(addr++, d);
          for (i = 0; i < d; i++)
            EEPROM.write(addr++, s[i]);
        #ifdef DEBUGING
          Serial.print(What);
          Serial.println(s);
        #endif
        }
```

```
//-------------------------------------------------------------
// Show Title
//-------------------------------------------------------------
void ShowTitle()
{
  lcd.setCursor(0, 0);
  lcd.write(LogoChar);
  //lcd.print("12345678901234567890");
  lcd.print(F(" FIRE ALARM SYSTEM "));
}
//-------------------------------------------------------------
// Read Temp and Humi
//-------------------------------------------------------------
unsigned long lastReadTemp = 0;
void ReadTempAndHumi()
{
  if (millis() < lastReadTemp) return;
  //  Temperature = sht1x.readTemperatureC() + TempCalibration;
  //  Humidity = sht1x.readHumidity() + HumiCalibration;
  lastReadTemp = millis() + 5000;
}
//-------------------------------------------------------------
// Hien thi nhiet do, do am
//-------------------------------------------------------------
float lastTemp = 0, lastHumi = 0;
void ShowTempAndHumi(bool force)
{
  if (force == false)
  {
    if ((lastTemp == Temperature) && (lastHumi == Humidity)) return;
  }
  lcd.setCursor(0, 1);
  lcd.print("T=");
  lcd.print(Temperature, 1);
  lcd.write(1);
  lcd.print("C  ");

  lcd.setCursor(10, 1);
  lcd.print("H=");
  lcd.print(round(Humidity), DEC);
  lcd.print("% ");
```

```
  lastTemp = Temperature;
  lastHumi = Humidity;
 }
 //------------------------------------------------------------
 // Hien thi nhiet do, do am
 //------------------------------------------------------------
 void ShowTempAndHumi()
 {
  ShowTempAndHumi(false);
 }
 /*****************************************************************************
   Hien thi thoi gian
 *****************************************************************************/
 void DisplayTime()
 {
  lcd.setCursor(0, 1);
  lcd.write(ClockChar);
  lcd.print(Format2D(hour, minute, second, ":"));
  lcd.print(" ");
  lcd.write(CalendarChar);
  lcd.print(Format2D(day, month, year, "-"));

 }
 //------------------------------------------------------------
 // Hien thi lai man hinh chinh
 //------------------------------------------------------------
 void ShowScreen()
 {
  ShowTitle();
  ShowTempAndHumi(true);
  DisplayTime();
 }
 /*****************************************************************************
   Đợi phím nhất, trả về trạng thái của phím 0=nhấn, 1=không nhấn
 *****************************************************************************/
 int KeyMod = 0, KeyUp = 0, KeyDown = 0, KeyOK = 0;

 int ReadKey(int KeyCode)
 {
  unsigned long WaitTime = millis() + 50;
  while (digitalRead(KeyCode) == HIGH)
  {
```

```
            if (millis() > WaitTime) return 0; //Khong nhan phim
          }
          if (digitalRead(KeyCode) == HIGH) return 0;
          //Phim da nhan (KeyCode=LOW)
          WaitTime = millis() + 300;
          while (millis() < WaitTime)
          {
            if  (digitalRead(KeyCode) == HIGH)//Nha phim
                if (millis() > (WaitTime - 200)) return 1;//Chap nhan khi da nhan it nhat 100mS, nguoc lai rung phim, chua chap nhan
          }
          return 1;//Nha phim hoac da nhan duoc it nhat 300mS
        }

        /*****************************************************************************
          Đọc, kiểm tra trạng thái của 4 phím nhấn
         *****************************************************************************/
        int CheckKey()
        {
          KeyMod = ReadKey(MenuButtonPin);
          KeyDown = ReadKey(DownButtonPin);
          KeyUp = ReadKey(UpButtonPin);
          KeyOK = ReadKey(OKButtonPin);
        }

        /*****************************************************************************
          Trả về true nếu có ít nhất một phím được nhấn
         *****************************************************************************/
        bool KeyPress()
        {
          if (KeyMod > 0) return true;
          if (KeyUp > 0) return true;
          if (KeyDown > 0) return true;
          if (KeyOK > 0) return true;
          return false;
        }
        /*****************************************************************************
          Doi phim nhan trong khoang thoi gian dat truoc
          Tra ve:
          - true: neu co phim nhan
          - false: neu qua gio khong co phim nao nhan
         *****************************************************************************/
```

```
bool WaitKey(long timeout)
{
  CheckKey();
  long t = millis();
  while ((KeyPress() == false) && ((millis() - t) < timeout)) CheckKey();
  if (!KeyPress()) return false;//Qua gio, ket thuc false
  return true;//Co phim nhan, ket thuc true
}

/*************************************************************************
   Nhap muc chon tren danh sach hien thi 1 dong, nhap nhay
   Cach goi:
   i=ReadListItem(7, 2, {"Enabled", "Disabled"}, 2, i, timeout);
 *************************************************************************/
int ReadListItem(int x, int y, const String Items[], int Count,  int item , long timeout)
{
  int n = 0;
  bool on = true;
  int tout = 0;
  String blank = "";
  int saveitem = item;
  for (int i = 0; i < Count; i++)
    if (String(Items[i]).length() > n) n = String(Items[i]).length();
  for (int i = 0; i < n; i++)
    blank += " ";
  lcd.setCursor(x, y);
  lcd.print(blank);//Xoa dau Enter vi tri cu

  while (true)
  {
    if (on)
    {
      lcd.setCursor(x, y);
      lcd.print(String(Items[item]));//Dau Enter
    }
    else
    {
      lcd.setCursor(x, y);
      lcd.print(blank);//Xoa dau Enter vi tri cu

    }
    if (!WaitKey(300))
```

```
    {
      tout += 300;
      if (tout > timeout) return -1;
    }
    else
      tout = 0;
    on = !on;
    if (KeyUp == 1)
      if (item > 0)
        item--;
      else
        item = Count - 1;
    if (KeyDown == 1)
    {
      if (item < (Count - 1))
        item++;
      else
        item = 0;
    }
    if (KeyOK == 1)
    {
      lcd.setCursor(x, y);
      lcd.print(String(Items[item]));//Dau Enter
      return item;
    }
    if (KeyMod == 1)
    {
      lcd.setCursor(x, y);
      lcd.print(String(Items[item]));//Dau Enter
      return saveitem;
    }
  }
}

/*********************************************************************
   Thiet lap che do co canh bao bang cach nhan tin hay khong
 *********************************************************************/
void SetGSMSupport(long timeout)//Set SMS alarm or not
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("SET GSM SUPPORT"));
```

```
          lcd.setCursor(0, 1);
          lcd.print(F("Value:"));
          int old = GSMSupport;
          GSMSupport = ReadListItem(7, 1, EnabledDisabledList, 2, GSMSupport, timeout);
          WriteConfig();
          if (GSMSupport != old)
            if (GSMSupport == 1)  InitSIM();
        }

        /*************************************************************************
          Nhap thoi nhiet do gioi han
         *************************************************************************/
        void SetTempCalibration(long timeout)
        {
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print(F("SET TEMP CALIB "));
          lcd.setCursor(0, 1);
          lcd.print(F("Value:"));
          lcd.setCursor(14, 1);
          lcd.write(1);
          lcd.print("C");
               TempCalibration    =    ReadStringNumber(7,    1,    5,    String(TempCalibration),
timeout).toFloat();
          WriteConfig();
        }



        /*************************************************************************
          Nhap thoi nhiet do gioi han
         *************************************************************************/
        void SetHumiCalibration(long timeout)
        {
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print(F("SET HUMI CALIB "));
          lcd.setCursor(0, 1);
          lcd.print(F("Value:"));
          lcd.setCursor(15, 1);
          lcd.print("%");
               HumiCalibration    =    ReadStringNumber(7,    1,    5,    String(HumiCalibration),
timeout).toFloat();
```

```
  WriteConfig();
}


/*************************************************************************
  Thiet lap thoi gian cap nhat du lieu len Gateway
 *************************************************************************/
void SetGatewayUpdateTime(long timeout)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("SET GW UPDATE T"));
  lcd.setCursor(0, 1);
  lcd.print(F("Value:"));
  lcd.setCursor(14, 1);
  lcd.print("mS");
  UpdateTime = ReadStringNumber(7, 1, 5, String(UpdateTime), timeout).toInt();
  WriteConfig();
}

/*************************************************************************
  Cap nhat thoi gian vao chip
 *************************************************************************/
void DS1307SetTime(byte hr, byte minu, byte sec, byte d, byte mth, byte yr)
{
  byte wd = 1;
  Wire.beginTransmission(DS1307Addr);
  Wire.write(byte(0x00)); // đặt lại pointer
  Wire.write(dec2bcd(sec));
  Wire.write(dec2bcd(minu));
  Wire.write(dec2bcd(hr));
  Wire.write(dec2bcd(wd)); // day of week: Sunday = 1, Saturday = 7
  Wire.write(dec2bcd(d));
  Wire.write(dec2bcd(mth));
  Wire.write(dec2bcd(yr));
  Wire.endTransmission();
}


/*************************************************************************
  Cập nhật thơi gian
 *************************************************************************/
```

```
unsigned long lastUpdateTimer = -5000;

void UpdateTimer()
{
  long delta = millis() - lastUpdateTimer;
  if (delta < 950) return;
  lastUpdateTimer = millis();
  Wire.beginTransmission(DS1307Addr);
  Wire.write((byte)0x00);
  Wire.endTransmission();
  Wire.requestFrom(DS1307Addr, NumberOfFields);
  second = bcd2dec(Wire.read() & 0x7f);
  minute = bcd2dec(Wire.read() );
  hour   = bcd2dec(Wire.read() & 0x3f); // chế độ 24h.
  wday   = bcd2dec(Wire.read() );
  day    = bcd2dec(Wire.read() );
  month  = bcd2dec(Wire.read() );
  year   = bcd2dec(Wire.read() );
  DisplayTime();
#ifdef USEDS1307Advanced
  SystemTime.Update(second, minute, hour, wday, day, month, year);
#endif
}
/*********************************************************************
   Dinh dang du lieu ngay thang, gio
 *********************************************************************/
String Format2D(int a, int b, int c, String sp)
{
  String s = "", s1 = "";
  s1 = String(a);
  if (s1.length() < 2) s1 = "0" + s1;
  s = s1;
  s1 = String(b);
  if (s1.length() < 2) s1 = "0" + s1;
  s = s + sp + s1;
  s1 = String(c);
  if (s1.length() < 2) s1 = "0" + s1;
  s = s + sp + s1;
  return s;
}
```

```
#ifdef USEDS1307
#ifdef USEDS1307Advanced
/*************************************************************************
  Nhap, dat lai ngay, gio he thong
 *************************************************************************/
void SetTimer(long timeout)//Set the system timer
{
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print(F("SET SYSTEM TIMER"));
 lcd.setCursor(0, 1);
 lcd.print(String(F("Time: ")) + SystemTime.GetTimeStringToPrint());//Time: 12:31:22
 SystemTime.Hour = ReadIntNumber(6, 1, SystemTime.Hour, 0, 23, timeout, 2);
 SystemTime.Min = ReadIntNumber(9, 1, SystemTime.Min, 0, 59, timeout, 2);
 SystemTime.Sec = ReadIntNumber(12, 1, SystemTime.Sec, 0, 59, timeout, 2);
 lcd.setCursor(0, 1);
 lcd.print(String(F("Day:  ")) + SystemTime.GetDayStringToPrint()); //Day:  03/09/17
 SystemTime.Day = ReadIntNumber(6, 1, SystemTime.Day, 1, 31, timeout, 2);
 SystemTime.Month = ReadIntNumber(9, 1, SystemTime.Month, 1, 12, timeout, 2);
 SystemTime.Year = ReadIntNumber(12, 1, SystemTime.Year, 1, 99, timeout, 2);

 if (SystemTime.Year > 99) SystemTime.Year = 18;
 if (SystemTime.Month > 12) SystemTime.Month = 1;
 if (SystemTime.Day > 31) SystemTime.Day = 1;

 if (SystemTime.Hour > 23) SystemTime.Hour = 0;
 if (SystemTime.Min > 59) SystemTime.Min = 0;
 if (SystemTime.Sec > 59) SystemTime.Sec = 0;

  DS1307SetTime(SystemTime.Hour, SystemTime.Min, SystemTime.Sec, SystemTime.Day,
SystemTime.Month, SystemTime.Year);

}

#else


/*************************************************************************
  Nhap, dat lai ngay, gio he thong
 *************************************************************************/
void SetTimer(long timeout)//Set the system timer
{
```

```
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print(F("SET SYSTEM TIMER"));
          lcd.setCursor(0, 1);

          lcd.print(String(F("Time: ")) + Format2D(hour, minute, second, ":")); //Time: 12:31:22
          hour = ReadIntNumber(6, 1, hour, 0, 23, timeout, 2);
          minute = ReadIntNumber(9, 1, minute, 0, 59, timeout, 2);
          second = ReadIntNumber(12, 1, second, 0, 59, timeout, 2);
          lcd.setCursor(0, 1);
          lcd.print(String(F("Day:  ")) + Format2D(day, month, year, "-")); //Day:  03/09/17
          day = ReadIntNumber(6, 1, day, 1, 31, timeout, 2);
          month = ReadIntNumber(9, 1, month, 1, 12, timeout, 2);
          year = ReadIntNumber(12, 1, year, 1, 99, timeout, 2);

          if (year > 99) year = 18;
          if (month > 12) month = 1;
          if (day > 31) day = 1;

          if (hour > 23) hour = 0;
          if (minute > 59) minute = 0;
          if (second > 59) second = 0;

          DS1307SetTime(hour, minute, second, day, month, year);

        }
        #endif
        #endif

        /*************************************************************************
          Hien thi thong tin he thong
         *************************************************************************/
        void About()
        {
          SimBuffer = F("Fire Alarm System. Temperature and Humidity sensor support. Send SMS
notify. Verion ");
          SimBuffer += String(Version) + ".";
          DisplaySimBuffer(5000);
          SimBuffer = "";
          ShowScreen();
        }
```

```
/*****************************************************************************
  Kiem tra tien trong tai khoa SIM
 *****************************************************************************/
int SimReadMessageStr(long timeout)
{
 unsigned long t = millis() + timeout;
 byte d = 0;
 while (millis() < t) {
  while ((millis() < t) && (SerialForSIM800L.available()))
  {
   char c = (char) SerialForSIM800L.read();
   if (c == '"')
   {
    d++;
    if (d == 1)
     SimBuffer = "";
    else if (d > 1) {
     t = millis() - 10;
     break;
    }
   }
   else if (SimBuffer.length() < 255) SimBuffer += c;
  }
 }
 return SimBuffer.length();
}
/*****************************************************************************
  Hien thi du lieu doc tu SIM duoi dang cuon toan man hinh
 *****************************************************************************/
void DisplaySimBuffer(int ScreenDelay)
{
 int i1 = 0, i2 = 16;
 lcd.clear();
 int x = 0, y = 0;
 while (i1 < SimBuffer.length())
 {
  i2 = i1;
  x = 0; y = 0;
  while ((i2 < SimBuffer.length()) && (y < Max_LCD_Rows))
  {
   lcd.setCursor(x, y);
   if (i2 < SimBuffer.length())
```

```
            lcd.print(SimBuffer[i2]);
          else
            lcd.print(" ");
          i2++;
          x++;
          if (x >= Max_LCD_Cols)
          {
            x = 0;
            y++;
            if (y >= Max_LCD_Rows) break;
          }
        }
        CheckKey();
        if (KeyPress()) break;
        if (i1 == 0)
          delay(1000);
        else
          delay(200);
        i1++;
        if ((i1 + (Max_LCD_Rows * Max_LCD_Cols)) > SimBuffer.length())
        {
          delay(2000);
          break;
        }
      }
    }
    /**************************************************************************
      Kiem tra tien trong tai khoa SIM
     **************************************************************************/
    void CheckMoney(long timeout)
    {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(F("CHECK MONEY"));
      lcd.setCursor(0, 1);
      if (GSMSupport == 1)
        lcd.print(F("Please wait ..."));
      else
      {
        lcd.print(F("GSM disabled!"));
        delay(1000);
        return;
```

```
  }
  delay(1000);
  SimBuffer = "";
  SendCheckMoney();
  LastCommingPhoneNumber = -1;
  SimReadMessageStr(15000);
  lcd.clear();
  if (SimBuffer.length() > 0)
  {
    DisplaySimBuffer(5000);
    SimBuffer = "";
  }
  else
  {
    lcd.setCursor(0, 0);
    lcd.print(F("CHECK MONEY"));
    lcd.setCursor(0, 1);
    lcd.print(F("Please try late"));
    delay(3000);
  }
  ShowTitle();
}
/*********************************************************************************
   Nap tien vao tai khoan SIM
 *********************************************************************************/
void ChargeMoney(long timeout)//Set Call alarm or not
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("CHARGE MONEY"));
  lcd.setCursor(0, 1);
  if (GSMSupport == 1)
    lcd.print(F("Enter Code..."));
  else
  {
    lcd.print(F("GSM disabled!"));
    delay(1000);
    return;
  }
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
```

```
  lcd.print(":");
  String s = ReadStringNumber(1, 0, 19, "", timeout);
  if (s.length() == 0)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(F("Code blank..."));
    delay(1000);
    return;
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("CHARGING MONEY"));
  lcd.setCursor(0, 1);
  lcd.print(F("Please wait ..."));
  SendAddMoney(s);
  SimReadMessageStr(15000);
  lcd.clear();
  if (SimBuffer.length() > 0)
  {
    DisplaySimBuffer(5000);
    SimBuffer = "";
  }
  ShowTitle();
}

/*****************************************************************************
  Read Number in GSM Data
 *****************************************************************************/
String GetNumberInGSMData()
{
  if (SimBuffer.length() > 10) //avoid empty sms
  {
    uint8_t i1 = SimBuffer.indexOf("+CMT:");
    if (i1 >= 0)
    {
      i1 = i1 + 7;
      uint8_t i2 = SimBuffer.indexOf("\",\"", i1 + 4);
      if (i2 > i1)
        return "0" + SimBuffer.substring(i1 + 3, i2);//Bo qua +84 thay bang 0 vao dau
    }
  }
```

```
  return "";
}

/*****************************************************************************
  Kiem tra so dien thoai trong TempStr co nam trong danh ba khoong
 *****************************************************************************/
bool NumberInContact()
{
  return true;
}

/*****************************************************************************
  Read Code in GSM Data
 *****************************************************************************/
String GetCodeInGSMData()
{
  if (SimBuffer.length() > 10) //avoid empty sms
  {
    uint8_t i1 = SimBuffer.indexOf("CODE=");
    if (i1 >= 0)
    {
      i1 = i1 + 5;
      uint8_t i2 = SimBuffer.indexOf(".", i1 + 4);
      if (i2 > i1)
        return SimBuffer.substring(i1, i2);
    }
  }
  return "";
}

/*****************************************************************************
  Change Password
 *****************************************************************************/
void ChangePassword(long timeout)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("CHANGE PASSWORD"));
  lcd.setCursor(0, 1);
  lcd.print(F("Pass:"));
  Password = ReadStringNumber(6, 1, 10, Password, timeout);
  WriteConfig();
```

```
}
/*****************************************************************************
  Hien thi man hinh chinh
 *****************************************************************************/
int GetMainMenuItem(const String Items[], int Count, long timeout, int item = 0)
{
 int start = 0, lastitem = item + 1;
 bool redraw = true;
 if ((start + 1) < item) start = item - 1;

 while (true)
 {
  if (redraw)
  {
   int y = 0;
   lcd.clear();
   delay(10);
   for (int i = start; i < Count; i++)
   {
    lcd.setCursor(1, y);
     //Bo sung phan ps de dam bao in du 20 ky tu tren dong cho khong bi nhieu (2018-
01-30)
    String ps = String(Items[i]);
    while (ps.length() < Max_LCD_Cols) ps = ps + " ";
    ps = ps.substring(0, Max_LCD_Cols);
    lcd.print(ps);
    y++;
    if (y >= Max_LCD_Rows) break;
   }
   redraw = false;
  }
  if (lastitem != item)
  {
   lcd.setCursor(0, lastitem - start);
   lcd.print(" ");//Xoa dau Enter vi tri cu
   lastitem = item;
   lcd.setCursor(0, item - start);
   lcd.write(3);//Dau Enter
  }
  if (!WaitKey(timeout)) return -1;
  if (KeyUp == 1)
   if (item > 0)
```

```
      {
        item--;
        if (item < start) {
          start = item;
          redraw = true;
        }
      }
    if (KeyDown == 1)
    {
      if (item < (Count - 1))
      {
        item++;
        if ((start + 1) < item)
        {
          start = item - 1;
          redraw = true;
        }
      }
    }
    if (KeyOK == 1)
      return item;
    if (KeyMod == 1)
      return (Count - 1);
  }
}
/**************************************************************************
   Bật/Tắt đèn nền
 **************************************************************************/
bool backlightIsOn = true;
void TurnBacklightON()
{
  lcd.backlight();
  backlightIsOn = true;
}

void TurnBacklightOFF()
{
  lcd.noBacklight();
  backlightIsOn = false;
}
/**************************************************************************
   Đọc dãy số
```

```
 ***************************************************************************/
String ReadStringNumber(int x, int y, int Count, String initValue, long timeout)
{
  if (Count > 20) Count = 20;
  char ReadBuffer[Count];
  for (int i = 0; i < Count; i++)
    if (i < initValue.length())
      ReadBuffer[i] = initValue.charAt(i);
    else
      ReadBuffer[i] = 32;
  int vt = 0;
  long tout = 0;
  lcd.blink();
  lcd.cursor();
  lcd.setCursor(x, y);
  for (int i = 0; i < Count; i++)
    lcd.print(ReadBuffer[i]);
  char old = ReadBuffer[vt] + 1;
  int oldvt = vt + 1;
  while (true)
  {
    if ((old != ReadBuffer[vt]) || (vt != oldvt))
    {
      lcd.setCursor(x + vt, y);
      lcd.print(ReadBuffer[vt]);
      lcd.setCursor(x + vt, y);
      oldvt = vt;
      old = ReadBuffer[vt];
    }
    if (!WaitKey(300))
    {
      tout += 300;
      if (tout > timeout)
      {
        lcd.noBlink();
        lcd.noCursor();
        return initValue;
      }
    }
    else
      tout = 0;
```

```
if (KeyUp == 1)
  if (ReadBuffer[vt] == '.')
  {
    if (vt == 0)
      ReadBuffer[vt] = '-';
    else
      ReadBuffer[vt] = ' ';
  }
  else if (ReadBuffer[vt] == '-')
    ReadBuffer[vt] = 32;
  else if (ReadBuffer[vt] == ' ')
    ReadBuffer[vt] = '0';
  else if (ReadBuffer[vt] < '9')
    ReadBuffer[vt]++;
  else
    ReadBuffer[vt] = '.';

//'0123456789.- '

if (KeyDown == 1)
  if (ReadBuffer[vt] == ' ')
  {
    if (vt == 0)
      ReadBuffer[vt] = '-';
    else
      ReadBuffer[vt] = '.';
  }
  else if (ReadBuffer[vt] == '-')
    ReadBuffer[vt] = '.';
  else if (ReadBuffer[vt] == '.')
    ReadBuffer[vt] = '9';
  else if (ReadBuffer[vt] > '0')
    ReadBuffer[vt]--;
  else
    ReadBuffer[vt] = ' ';

//Huy bo viec nhapj
if (KeyMod == 1)
{
  lcd.noBlink();
  lcd.noCursor();
```

```
            return initValue;
          }
          if (KeyOK == 1)
          {
            //Hoan tat neu khong nhap tiep hoac nhap du do dai
            if ((ReadBuffer[vt] == 32) || (vt == (Count - 1)))
              break;
            //Chuyen nhap ky tu so tiep theo
            if (vt < (Count - 1)) vt++;
          }
        }
        lcd.noBlink();
        lcd.noCursor();
        initValue = "";
        for (int i = 0; i < Count; i++)
        {
          if (ReadBuffer[i] == 32) break;
          initValue += ReadBuffer[i];
        }
        return initValue;
      }

      /************************************************************************
        Nhap vao mot gia tri so
        - Tai toa do: x,y
        - Gia tri khoi dau la initValue
        - Gia tri nho nhat duoc nhap la: minValue
        - Gia tri lon nhat duoc nhap la: maxValue
        - Thoi gian toi da cho nhap: timeout
       ************************************************************************/
      int ReadIntNumber(int x, int y, int initValue, int minValue, int maxValue, long timeout, int
Count)
      {
        String Blank = "";
        if (Count == 0)
          Count = 20;
        else
          Count = x + Count;
        for (int i = x; i < Count; i++) Blank += " ";
        int Value = initValue;
        lcd.noCursor();
        lcd.setCursor(x, y);
```

```
        lcd.print(Blank);
        bool cl = false;
        long tout = 0;
        while (true)
        {
         if (cl)
         {
          lcd.setCursor(x, y);
          lcd.print(Blank);
         }
         else
         {
          lcd.setCursor(x, y);
          lcd.print(String(Value));
         }
         cl = !cl;
         if (WaitKey(300))
         {
          tout = 0;
          if (KeyUp == 1)
          {
           if (Value < maxValue) Value++;
          }

          if (KeyDown == 1)
          {
           if (Value > minValue) Value--;
          }
          if (KeyMod == 1)
          {
           lcd.setCursor(x, y);
           lcd.print(String(Value));
           return initValue;
          }
          if (KeyOK == 1)
          {
           lcd.setCursor(x, y);
           lcd.print(String(Value));
           return Value;
          }
         }
         else
```

```
    {
      tout += 300;
      if (tout > timeout)
      {
        lcd.setCursor(x, y);
        lcd.print(String(Value));
        return initValue;
      }
    }
  }
}
/*****************************************************************************
   Đọc noi dung tu SIM gui den
 *****************************************************************************/
int ReadSIM(bool passEnter)
{
  unsigned long wait = millis() + 3000;
  while ((millis() < wait) && (SerialForSIM800L.available()))
  {
    char c = (char) SerialForSIM800L.read();
    if (passEnter)
    {
      if ((c == 10) || (c == 13)) SimBuffer = "";
      else if (SimBuffer.length() < 255) SimBuffer += c;
    }
    else if (SimBuffer.length() < 255) SimBuffer += c;
  }
  return SimBuffer.length();
}
/*****************************************************************************
   Đọc noi dung tu SIM gui den
 *****************************************************************************/
int ReadESP(bool passEnter)
{
  unsigned long wait = millis() + 50;
  while (millis() < wait) if (SerialForWifi.available()) break;
  wait = millis() + 3000;
  while ((millis() < wait) && (SerialForWifi.available()))
  {
    char c = (char) SerialForWifi.read();
    if (passEnter)
    {
```

```
      if ((c == 10) || (c == 13)) SimBuffer = "";
      else if (SimBuffer.length() < 255) SimBuffer += c;
     }
     else if (SimBuffer.length() < 255) SimBuffer += c;
   }
   return SimBuffer.length();
}
/**************************************************************************
   Đọc noi dung tu SIM gui den
 **************************************************************************/
int ReadSIM()
{
  return ReadSIM(false);
}
/**************************************************************************
   Đọc noi dung tu SIM gui den
 **************************************************************************/
int ReadESP()
{
  return ReadESP(false);
}
/**************************************************************************
   Gui lenh AT
 **************************************************************************/
void ATCommand(String ATCmd)
{
  SerialForSIM800L.print(ATCmd);
  SerialForSIM800L.print (F("\r\n"));
  delay(10);
  ReadSIM();
  lastSimCmdTime = millis();
}

/**************************************************************************
   Khoi tao SIM
 **************************************************************************/
void InitSIM()
{
  ATCommand(F("ATE0"));//Tat che do phan hoi lenh AT
  ATCommand(F("AT+CUSD=1"));//Bat che do hien thi thong tin cua nha mang
  ATCommand(F("AT+DDET=1"));//Bat che do DTMF
  ATCommand(F("AT+CNMI=2,2"));//Hien thi noi dung tin nhan truc tiep
```

```
  ATCommand(F("AT+CMGF=1"));//Chon che do Text
  ATCommand(F("AT+CSCS=\"GSM\""));//Chon che do Text
  ATCommand(F("AT+CLIP=1"));//Hien thi SDT goi den
  lastSimCmdTime = millis();
}
/**************************************************************************
   Thuc hien cuoc goi
 **************************************************************************/
void CallNumber(String Number)
{
  if (GSMSupport == 0) return;
  SerialForSIM800L.print (F("ATD"));
  SerialForSIM800L.print (Number);
  SerialForSIM800L.print (F(";\r\n"));
  delay(1000);
  lastSimCmdTime = millis();
}
/**************************************************************************
   Gui tin nhan
 **************************************************************************/
void SendSMS(String Number, String SMS)
{
  if (GSMSupport == 0) return;
  SerialForSIM800L.print(F("AT+CMGF=1\r\n"));    //Sets the GSM Module in Text Mode
  delay(1000);
  SerialForSIM800L.print(SMS);
  delay(100);
  SerialForSIM800L.print((char)26);// ASCII code of CTRL+Z
  delay(1000);
  lastSimCmdTime = millis();
}
/**************************************************************************
   Gui tin nhan tu bo dem SimBuffer va so dien thoai tu TempStr
   Tu dong tach thanh cac tin ngan 150 ky tu
 **************************************************************************/
void SendSMSinSimBuffer(int Start, int Stop)
{
  if (GSMSupport == 0) return;
  int i = Start, n = 0;
  SerialForSIM800L.print(F("AT+CMGF=1\r\n"));    //Sets the GSM Module in Text Mode
  delay(1000);
  if (Stop == 0) Stop = SimBuffer.length();
```

```
          while (i < Stop)
          {
            SerialForSIM800L.print("AT+CMGS=\"" + TempStr + "\"\r\n"); //Mobile phone number
to send message
            delay(1000);
            while ((i < Stop) && (n < 150))
            {
              SerialForSIM800L.print(SimBuffer[i]);
              i++;
              n++;
            }
            delay(100);
            SerialForSIM800L.print((char)26);// ASCII code of CTRL+Z
            delay(1000);
            n = 0;
            if (i < Stop) delay(5000);
          }
          lastSimCmdTime = millis();
        }
        /********************************************************************************
          Nap tien tai khoan
         ********************************************************************************/
        void SendAddMoney(String _code) {
          if (GSMSupport == 0) return;
          SerialForSIM800L.print(F("ATD*100*"));
          SerialForSIM800L.print(_code);
          SerialForSIM800L.print(F("#\r\n"));
          lastSimCmdTime = millis();
        }
        /********************************************************************************
          Kiem tra tien tai khoan
         ********************************************************************************/
        void SendCheckMoney() {
          if (GSMSupport == 0) return;
          SerialForSIM800L.print(F("ATD*101#\r\n"));
          lastSimCmdTime = millis();
        }
        /********************************************************************************
          Kiem tra tien tai khoan
         ********************************************************************************/
        void DeleteAllSMS()
        {
```

```
  if (GSMSupport == 0) return;
  SerialForSIM800L.print(F("AT+CMGDA=\"del all\"\n\r"));
  delay(1000);
  lastSimCmdTime = millis();
}
/*****************************************************************************
  Tuong tac nguoi dung qua LCD+Keys
 *****************************************************************************/
bool Login()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("SYSTEM LOGIN"));
  lcd.setCursor(0, 1);
  lcd.print(F("Pass:"));
  String s = ReadStringNumber(5, 1, 10, "", 30000);//Password
  bool t = (s.length() == Password.length());
  if (t)
    for (int i = 0; i < Password.length(); i++)
    {
      if (s[i] != Password[i])
      {
        t = false;
        break;
      }
    }
  lcd.setCursor(0, 1);
  if (t)
  {
    lcd.print(F("OK, Login System"));
    delay(500);
    return true;
  }
  else if (s.length() > 0)
  {
    lcd.print(F("Invalid Pass..."));
    delay(2000);
  }
  return false;
}
/*****************************************************************************
  Dieu khien LED bao trang thai
```

```
 ***************************************************************************/
long LastFlashCheck = 0;
byte FlashLightState = 0;
void MonitorSignal()
{
  if ((millis() - LastFlashCheck) > 500)
  {
    digitalWrite(FlashLightPort, FlashLightState);
    FlashLightState = FlashLightState > 0 ? 0 : 1; //Nhay LED san co tren board de quan sat trang thai dang hoat dong cua board
    LastFlashCheck = millis();
  }
}

/***************************************************************************
  Gui xac nhan OK toi ESP
 ***************************************************************************/
void ResponseOKtoESP(String s)
{
  SerialForWifi.print(F("#PNH#OK"));
  SerialForWifi.print(s);
  if (s.length() > 0)
  {
    if (s[s.length() - 1] != '#')
      SerialForWifi.print("#");
  }
  else SerialForWifi.print("#");
  SerialForWifi.println(F("END"));
}
/***************************************************************************
  Gui bao loi toi ESP
 ***************************************************************************/
void ResponseERRORtoESP(String s)
{
  SerialForWifi.print(F("#PNH#OK"));
  SerialForWifi.print(s);
  if (s.length() > 0)
  {
    if (s[s.length() - 1] != '#')
      SerialForWifi.print("#");
  }
  else SerialForWifi.print("#");
```

```
  SerialForWifi.println(F("END"));
}
/*****************************************************************************
   Gui du lieu ti bo dem SIM toi ESP
 *****************************************************************************/
void SendSimBufferToESP(int From, int Length)
{
  int i1;
  for (i1 = From; i1 < Length; i1++)
    if (SimBuffer[i1] == '#')
      SerialForWifi.print(".");
    else
      SerialForWifi.print(SimBuffer[i1]);
}
/*****************************************************************************
   Xu ly du lieu dieu khien gui tu ESP toi
 *****************************************************************************/
void ProcessESPData()
{
  if (SimBuffer.indexOf(String(F("#PNH#"))) < 0) return;
  if (SimBuffer.indexOf(String(F("#END"))) < 0) return;
  int i1, i2;
  unsigned long Wait;
  String s;

#ifdef DEBUG
  lcd.setCursor(0, 0);
  lcd.print(F("Valid data"));
  delay(3000);
  TurnBacklightON();
#endif
  //Nhan tin ngan
  s = String(F("#SMS#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
    if (GSMSupport > 0)
    { //Nhan tin
      i2 = SimBuffer.indexOf("#", i1 + s.length());
      if (i2 > i1)
      {
        TempStr = SimBuffer.substring(i1 + s.length(), i2);
        i1 = i2 + 1;
```

```
                i2 = SimBuffer.indexOf(String(F("#END")), i1);
                if (i2 > i1) SendSMSinSimBuffer(i1, i2);
                SerialForWifi.println(F("SMSDONE,"));
                SerialForWifi.println(i2 - i1);

          #ifdef DEBUG
                lcd.setCursor(0, 0);
                lcd.print(F("Send SMS"));
                lcd.setCursor(0, 1);
                lcd.print(TempStr);
                delay(3000);
          #endif
               }
              }
              else SerialForWifi.println(F("NOGSM"));

            //Nhan tin noi dung dai
            s = String(F("#SMSL#"));
            i1 = SimBuffer.indexOf(s);
            if ( i1 > 0)
              if (GSMSupport > 0)
              { //Nhan tin
                i2 = SimBuffer.indexOf("#", i1 + s.length());
                if (i2 > i1)
                {
                  TempStr = SimBuffer.substring(i1 + s.length(), i2);
                     SerialForSIM800L.print(F("AT+CMGF=1\r\n"));     //Sets the GSM Module in Text
Mode
                  SerialForSIM800L.print(F("AT+CMGS=\""));
                  SerialForSIM800L.print(TempStr);
                  SerialForSIM800L.print(F("\"\r\n")); //Mobile phone number to send message
                   //     SerialForSIM800L.print("AT+CMGS=\"" + TempStr + "\"\r\n"); //Mobile phone
number to send message
                  delay(500);
                  TempStr = "";
                  int n = 0;
          #ifdef DEBUG
                  SerialForWifi.println(TempStr);
          #endif

                  SerialForWifi.println(F("OK"));
                  Wait = millis() + 2000;
```

```
            while (millis() < Wait)
            {
              if (SerialForWifi.available()) break;
            }
            Wait = millis() + 5000;
            while  (millis() < Wait)
            {
              if (SerialForWifi.available())
              {
                char c = (char) SerialForWifi.read();
                if (c != '#') TempStr = TempStr + String(c);
                if ((TempStr.length() > 50) || (c == '#'))
                {
                  n += TempStr.length();
                  SerialForSIM800L.print(TempStr);
                  TempStr = "";
                }
                if (c == '#') break;
              }
            }
            if (TempStr.length() > 0)
            {
              n += TempStr.length();
              SerialForSIM800L.print(TempStr);
              TempStr = "";
            }
            delay(100);
            SerialForSIM800L.print((char)26);// ASCII code of CTRL+Z
            delay(1000);
            SerialForWifi.println(F("LSMSDONE,"));
            SerialForWifi.println(n);
        #ifdef DEBUG
            lcd.setCursor(0, 0);
            lcd.print(F("Send Long SMS"));
            lcd.setCursor(0, 1);
            lcd.print(n);
            delay(3000);
        #endif
          }
        }
        else SerialForWifi.println(F("NOGSM"));
```

```
  //Goi dien
  s = String(F("#CALL#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
    if (GSMSupport > 0)
    { //Goi dien
      i2 = SimBuffer.indexOf("#", i1 + s.length());
      if (i2 > i1)
      {
        TempStr = SimBuffer.substring(i1 + s.length(), i2);
        CallNumber(TempStr);
        SerialForWifi.println(F("CALLDONE"));
#ifdef DEBUG
        lcd.setCursor(0, 0);
        lcd.print(F("Call:"));
        lcd.setCursor(0, 1);
        lcd.print(TempStr);
        delay(3000);
#endif
      }
    }
    else SerialForWifi.print(F("NOGSM"));


  //Dat thoi gian update trang thai len ESP
  s = String(F("#UPDATETIME#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
  { //Dat thoi gian cap nhat du lieu len ESP
    i2 = SimBuffer.indexOf("#", i1 + s.length());
    if (i2 > i1)
    {
      UpdateTime = SimBuffer.substring(i1 + s.length(), i2).toInt();
      if (UpdateTime < 0) UpdateTime = 0;
      WriteConfig();
      ResponseOKtoESP(s);
#ifdef DEBUG
      lcd.setCursor(0, 0);
      lcd.print(F("Forward GSM:"));
      lcd.setCursor(0, 1);
      lcd.print(ForwardGSMData);
      delay(3000);
```

```
#endif
  }
 }

 //Dat dia chi LCD
 s = String(F("#LCDADDR#"));
 i1 = SimBuffer.indexOf(s);
 if ( i1 > 0)
 { //Dat dia chi LCD
  i2 = SimBuffer.indexOf("#", i1 + s.length());
  if (i2 > i1)
  {
   LCDAddr = SimBuffer.substring(i1 + s.length(), i2).toInt();
   WriteConfig();
   ResponseOKtoESP(s + String(LCDAddr));

#ifdef DEBUG
   lcd.setCursor(0, 0);
   lcd.print(F("LCDAddr"));
   lcd.setCursor(0, 1);
   lcd.print(LCDAddr);
   delay(3000);
#endif
  }
 }

 //Dat dia chi LCD
 s = String(F("#DS1307ADDR#"));
 i1 = SimBuffer.indexOf(s);
 if ( i1 > 0)
 { //Dat dia chi DS1307
  i2 = SimBuffer.indexOf("#", i1 + s.length());
  if (i2 > i1)
  {
   DS1307Addr = SimBuffer.substring(i1 + s.length(), i2).toInt();
   WriteConfig();
   ResponseOKtoESP(s + String(DS1307Addr));

#ifdef DEBUG
   lcd.setCursor(0, 0);
   lcd.print(F("DS1307Addr"));
   lcd.setCursor(0, 1);
```

```
      lcd.print(DS1307Addr);
      delay(3000);
#endif
    }
  }


  //Dat che do chuyen tiep du lieu GSM
  s = String(F("#FGSMDATA#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
  { //Dat dia chi DS1307
    i2 = SimBuffer.indexOf("#", i1 + s.length());
    if (i2 > i1)
    {
      ForwardGSMData = SimBuffer.substring(i1 + s.length(), i2).toInt();
      WriteConfig();
      ResponseOKtoESP(s + String(ForwardGSMData));

#ifdef DEBUG
      lcd.setCursor(0, 0);
      lcd.print(F("DS1307Addr"));
      lcd.setCursor(0, 1);
      lcd.print(DS1307Addr);
      delay(3000);
#endif
    }
  }

  //Dat GSM Support
  s = String(F("#GSMSUPPORT#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
  { //Bat/tat che do ho tro GSM
    i2 = SimBuffer.indexOf("#", i1 + s.length());
    if (i2 > i1)
    {
      GSMSupport = SimBuffer.substring(i1 + s.length(), i2).toInt();
      WriteConfig();
      ResponseOKtoESP(s + String(GSMSupport));

#ifdef DEBUG
```

```
      lcd.setCursor(0, 0);
      lcd.print(F("GSMSupport"));
      lcd.setCursor(0, 1);
      lcd.print(GSMSupport);
      delay(3000);
#endif
    }
  }

  //Dat TempCalibration
  s = String(F("#TEMPCALI#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
  { //Dat hieu chinh nhiet do
    i2 = SimBuffer.indexOf("#", i1 + s.length());
    if (i2 > i1)
    {
      TempCalibration = SimBuffer.substring(i1 + s.length(), i2).toFloat();
      WriteConfig();
      ResponseOKtoESP(s + String(TempCalibration));

#ifdef DEBUG
      lcd.setCursor(0, 0);
      lcd.print(F("TempCalibration"));
      lcd.setCursor(0, 1);
      lcd.print(TempCalibration);
      delay(3000);
#endif
    }
  }

  //Dat HumiCalibration
  s = String(F("#HUMICALI#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
  { //Dat hieu chinh nhiet do
    i2 = SimBuffer.indexOf("#", i1 + s.length());
    if (i2 > i1)
    {
      HumiCalibration = SimBuffer.substring(i1 + s.length(), i2).toFloat();
      WriteConfig();
      ResponseOKtoESP(s + String(HumiCalibration));
```

```
#ifdef DEBUG
      lcd.setCursor(0, 0);
      lcd.print(F("HumiCalibration"));
      lcd.setCursor(0, 1);
      lcd.print(HumiCalibration);
      delay(3000);
#endif
    }
  }

  //Dat Restart Device
  s = String(F("#CHECKMONEY#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
    if (GSMSupport > 0)
    { //Goi dien
      SimBuffer = "";
      SendCheckMoney();
      SimReadMessageStr(15000);
      if (SimBuffer.length() > 0)
      {
        SerialForWifi.print(F("#PNH#OK"));
        SerialForWifi.print(s);
        SendSimBufferToESP(0, SimBuffer.length());
        SerialForWifi.println(F("#END"));
        SimBuffer = "";
      }
      else
      {
        ResponseERRORtoESP(s);
      }
#ifdef DEBUG
      lcd.setCursor(0, 0);
      lcd.print(F("Check Money:"));
      delay(3000);
#endif
    }

  //Dat Nap tien
  s = String(F("#CHARGEMONEY#"));
  i1 = SimBuffer.indexOf(s);
```

```
          if ( i1 > 0)
            if (GSMSupport > 0)
            { //Nap tien
              i2 = SimBuffer.indexOf("#", i1 + s.length());
              if (i2 > i1)
              {
                SendAddMoney(SimBuffer.substring(i1 + s.length(), i2));
                SimReadMessageStr(15000);
                WriteConfig();
                if (SimBuffer.length() > 0)
                {
                  SerialForWifi.print(F("#PNH#OK"));
                  SerialForWifi.print(s);
                  SendSimBufferToESP(0, SimBuffer.length());
                  SerialForWifi.println(F("#END"));
                  SimBuffer = "";
                }
                else
                {
                  ResponseERRORtoESP(s);
                }


        #ifdef DEBUG
                lcd.setCursor(0, 0);
                lcd.print(F("Charge Money:"));
                delay(3000);
        #endif
              }
            }

           //Dat Dat thoi gian
           s = String(F("#SETDATETIME#"));
           i1 = SimBuffer.indexOf(s);
           if ( i1 > 0)
             if (GSMSupport > 0)
             { //Nap tien
               i2 = SimBuffer.indexOf("#", i1 + s.length());
               if (i2 > i1)
               {
                 TempStr = SimBuffer.substring(i1 + s.length(), i2);
                 if (TempStr.length() == 12)
```

```
        { //194315021018
          DS1307SetTime(
            TempStr.substring(0, 2).toInt(),
            TempStr.substring(2, 4).toInt(),
            TempStr.substring(4, 6).toInt(),
            TempStr.substring(6, 8).toInt(),
            TempStr.substring(8, 10).toInt(),
            TempStr.substring(10, 12).toInt());
          ResponseOKtoESP(s);
        } else ResponseERRORtoESP(s);


#ifdef DEBUG
        lcd.setCursor(0, 0);
        lcd.print(F("Set Date time:"));
        delay(3000);
#endif
      }
    }



  //Khoi phuc mac dinh
  s = String(F("#RESTOREDEFAULT#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
  { //Khoi phuc mac dinh
    ResponseOKtoESP(s);
    delay(1000);
    RestoreFactory();
  }

  //Dat Restart Device
  s = String(F("#RESTART#"));
  i1 = SimBuffer.indexOf(s);
  if ( i1 > 0)
  { //Khoi dong lai

    ResponseOKtoESP(s);
    delay(1000);
    RestartDevice();
  }
```

```
  SimBuffer = "";
 }

 /*************************************************************************
   Kiem tra trang thai module SIM
 *************************************************************************/
 String SIMState = "NEW";
 unsigned long LastCheckSIM = 0;

 void CheckSIM()
 {
  long delta = millis() -  LastCheckSIM;
  if ((delta > 60000) || (delta < 0))
  {
   delta = millis() - lastSimCmdTime;
   if ((delta > 0) && (delta < 10000)) return; //Khong gui lenh toi SIM neu moi gui khoang
10 giay tranh anh huong lenh quan trong truoc

   ATCommand(F("AT"));
   if (SimBuffer.indexOf(F("OK")) >= 0)
    SIMState = "OK";
   else
    SIMState = "ERR";
   LastCheckSIM = millis();
   SimBuffer = "";
  }
 }
 /*************************************************************************
   Gui du lieu trang thai toi ESP
 *************************************************************************/
 unsigned long lastUpdateStateToGateway = 0;
 void UpdateStateToGateway()
 {
 #ifdef USEDS1307
 #ifdef USEDS1307Advanced
  SerialForWifi.print(String(F("#PNH#ST#Sim=")) + SIMState +
                      String(F("#GP=")) + String(GSMSupport));
  SerialForWifi.print(String(F("#Temp=")) + String(Temperature) +
                      String(F("#Humi=")) + String(Humidity));
  SerialForWifi.println(String(F("#Time=")) + SystemTime.NowEn() +
                        String(F("#END")));
 #else
```

```
          SerialForWifi.print(String(F("#PNH#ST#Sim=")) + SIMState +
                              String(F("#GP=")) + String(GSMSupport));
          SerialForWifi.print(String(F("#Temp=")) + String(Temperature) +
                              String(F("#Humi=")) + String(Humidity));
           SerialForWifi.println(String(F("#Time=")) + Format2D(day, month, year, "-") + " " +
Format2D(hour, minute, second, ":") +
                               String(F("#END")));

        #endif
        #else
         /*
          SerialForWifi.println(String(F("#PNH#ST#Sim=")) + SIMState +
                              String(F("#GP=")) + String(GSPSupport) +
                              String(F("#Temp=")) + String(Temperature) +
                              String(F("#Humi=")) + String(Humidity) +
                              String(F("#END")));
         */
          SerialForWifi.print(String(F("#PNH#ST#Sim=")) + SIMState +
                              String(F("#GP=")) + String(GSMSupport));
          SerialForWifi.print(String(F("#Temp=")) + String(Temperature) +
                              String(F("#Humi=")) + String(Humidity));
           SerialForWifi.println(String(F("#Time=")) + Format2D(day, month, year, "-") + " " +
Format2D(hour, minute, second, ":") +
                               String(F("#END")));
        #endif
        }

        //------------------------------------------------------------
        // Menu process
        //------------------------------------------------------------
        void UserInteractive()
        {
         long TimeOut = 60000;
         int mainmenu = 0, menu = 0, submenu = 0;
         while (true)
         {
          mainmenu = GetMainMenuItem(mnMain, mnMainCount, TimeOut, mainmenu);
          if (mainmenu == -1) return;
          menu = 0;
          switch (mainmenu)
          {
           case 0://SMS Alarm
```

```
                SetGSMSupport(TimeOut);//Set SMS alarm or not
                break;

              case 1://Check Money
                CheckMoney(TimeOut);//Set Call alarm or not
                break;

              case 2://Charge Money
                ChargeMoney(TimeOut);//Set Call alarm or not
                break;

              case 3://System
                while (true)
                {
                  submenu = GetMainMenuItem(mnSystemSetting, mnSystemSettingCount, TimeOut,
submenu);
                  if (submenu == (mnSystemSettingCount - 1)) break;
                  switch (submenu)
                  {
                    case 0://Temp Calibration
                      SetTempCalibration(TimeOut);
                      break;

                    case 1://Humi Calibration
                      SetHumiCalibration(TimeOut);
                      break;

                    case 2://Change Password
                      ChangePassword(TimeOut);
                      break;

          #ifdef USEDS1307

                    case 3://SetTimer
                      SetTimer(TimeOut);
                      break;

                    case 4://About
                      SetGatewayUpdateTime(TimeOut);
                      break;

                    case 5://About
```

```
            About();
            break;
#else

          case 3://About
            SetGatewayUpdateTime(TimeOut);
            break;

          case 4://About
            About();
            break;

#endif

          default:
            break;
         }
        }
        break;

      default://Exit
        return;

    }
    if (menu == -1) break;
  }
}

void LCDPrintStr(String s, int Length)
{
  int i;
  for (i = 0; i < Length; i++)
    if (i < s.length())
      lcd.print(s[i]);
    else
      lcd.print(" ");
}
/*
   #define CamBienKhoiPin A0
  #define CamBienNhietPin A1
  #define CamDongDuPhong1Pin A2
  #define CamDongDuPhong2Pin A3
```

```
  #define CamBienDong1Pin A4
  #define CamBienDong2Pin A5
  #define CamBienDong3Pin A6

  #define BatterVoltagePin A7
  #define MainVoltagePin A8
*/
void TestReadADC()
{
  lcd.setCursor(0, 2);
  LCDPrintStr(String(analogRead(CamBienDong1Pin)), 6);
  LCDPrintStr(String(analogRead(CamBienDong2Pin)), 6);
  LCDPrintStr(String(analogRead(CamBienDong3Pin)), 6);
  lcd.setCursor(0, 3);
  LCDPrintStr(String(analogRead(CamBienNhietPin)), 6);
  LCDPrintStr(String(analogRead(CamBienKhoiPin)), 6);
  LCDPrintStr(String(analogRead(MainVoltagePin)), 6);
}

////////////////////////////////////////////////////////////////////////////

void loop() {
  long timeout = millis() - lastKeyPress;
  if (timeout > 30000) TurnBacklightOFF();
  ReadTempAndHumi();
  ShowTempAndHumi();
  UpdateTimer();
  TestReadADC();
  //Xu ly du lieu tu SIM gui den
  int len = ReadESP();
  if (len > 10) ProcessESPData();
  len = ReadSIM();
  //if (len>0) Serial.println(SimBuffer);
  if (len > 20)
  {
    //Serial.println(SimBuffer);
    if (SimBuffer.indexOf(F("CHECKMONEY")) >= 0)
    {
      TempStr = GetNumberInGSMData();
      if (NumberInContact())
      {
```

```
            SimBuffer = "";
            SendCheckMoney();
            SimReadMessageStr(15000);
            if (SimBuffer.length() > 0)
            {
              SendSMSinSimBuffer();
            }
          }
          TempStr = "";
          DeleteAllSMS();
        }
        else if (SimBuffer.indexOf(F("CODE=")) >= 0)
        {
          TempStr = GetNumberInGSMData();
          TempStr2 = GetCodeInGSMData();
          if (TempStr2.length() > 0)
          {
            SimBuffer = "";
            SendAddMoney(TempStr2);
            TempStr2 = "";
            SimBuffer = "";
            SimReadMessageStr(15000);
            if (SimBuffer.length() > 0)
            {
              SendSMSinSimBuffer();
            }
          }
          TempStr = "";
          DeleteAllSMS();
        }
        else if (SimBuffer.indexOf(F("READDATA")) >= 0)
        { //Doc du lieu ve nhiet do, do am
          TempStr = GetNumberInGSMData();
          if (NumberInContact())
          {
             SimBuffer = String(F("TEMP=")) + String(Temperature) + String(F("'C. HUMI=")) +
String(Humidity) + String(F("%."));
            SendSMSinSimBuffer();
          }
          TempStr = "";
          DeleteAllSMS();
        }
```

```
  if ((ForwardGSMData > 0) && (SimBuffer.length() > 0))
  {
    SerialForWifi.print(F("#PNH#GSMDATA#"));
    SendSimBufferToESP(0, SimBuffer.length());
    SerialForWifi.println(F("#END"));
  }
  //SimBuffer = "";
}


if (UpdateTime > 0)
{
  timeout = millis() - lastUpdateStateToGateway;
  if ((timeout > UpdateTime) || (timeout < 0))
  {
    UpdateStateToGateway();
    lastUpdateStateToGateway = millis();
  }
}
//Xu ly phim nhan
CheckKey();
if (KeyPress())
{
  if (!backlightIsOn)
  {
    TurnBacklightON();
    delay(300);
    lastKeyPress = millis();
  }
  else
  {
    TurnBacklightON();
    long t = millis();
    while (KeyPress() && ((millis() - t) < 5000)) CheckKey();
    if (KeyPress())
      if ((KeyUp == 1) && (KeyDown == 1) && (KeyOK == 1))
      {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(F("Restore factory"));
        lcd.setCursor(0, 1);
        lcd.print(F("data in 10 second"));
```

```
            lcd.setCursor(0, 2);
            for (int i = 0; i < 10; i++)
            {
              lcd.print(".");
              delay(1000);
            }
            RestoreFactory();
            setup();
            return;
          }
        if (Login())
        {
          UserInteractive();
        }
        ShowScreen();
        lastKeyPress = millis();
      }
    }
    MonitorSignal();
    CheckSIM();
    if (SimBuffer.indexOf(F("CARRIER")) >= 0)
    {
      SimBuffer = "";
    }
  }
```

1.3.2.2. Xây dựng hệ thống Backend

Hệ thống backend được xây dựng với mô hình hoạt động là một webserver. Nhiệm vụ của hệ thống backend bao gồm:

- Tiếp nhận các dữ liệu trạng thái, sự kiện do thiết bị gửi lên thông qua kết nối Internet.
- Giao tiếp với phần mềm chạy trên thiết bị di động (frontend). Gửi cập nhật trạng thái của thiết bị báo cháy gửi lên cho phần mềm trên thiết bị di động hiển thị, đồng thời nhận các lệnh điều khiển, thiết lập từ phần mềm chạy trên thiết bị di động, lưu trữ lại trên cơ sở dữ liệu đồng thời gửi xuống cho thiết bị báo cháy thực hiện. Các lệnh tiêu biểu như xác nhận báo chạy để ngừng phát tín hiệu báo cháy, điều khiển bật, tắt lộ điện được bảo vệ.
- Tổ chức lưu trữ dữ liệu quá trình diễn biến của hệ thống. Các dữ liệu này do thiết bị báo chạy gửi lên và được cập nhật thường xuyên như:

trạng thái các cảm biến, dòng điện, điện áp của hệ thống điện đang được giám sát và điều khiển…

1.3.2.2.1. Cơ sở dữ liệu của Backend

Tổ chức cơ sở dữ liệu lưu trữ thông tin trạng thái thiết bị báo cháy gửi lên, các lệnh giao tiếp điều khiển thiết bị báo cháy thông qua phần mềm chạy trên thiết bị di động:

a) Bảng lưu trữ thông tin xác thực người dùng

- Mỗi người dùng được cung cấp một tài khoản lưu trữ dữ liệu trên cloud và được xác thực mỗi khi đăng nhập, truy cập dữ liệu qua tài khoản này.
- Người dùng cần cung cấp thông tin "username" và "password" mỗi khi truy cập dữ liệu trên cloud (đăng nhập/xác thực người dùng).

Bảng 3: Bảng xác thực người dùng trên backend

	#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm
☐	1	**id** 🔑	int(11)			Không	*Không*		AUTO_INCREMENT
☐	2	**username**	text	utf8_unicode_ci		Không	*Không*		
☐	3	**password**	text	utf8_unicode_ci		Không	*Không*		
☐	4	**fullname**	text	utf8_unicode_ci		Không	*Không*		
☐	5	**Address**	text	utf8_unicode_ci		Không	*Không*		
☐	6	**Email**	text	utf8_unicode_ci		Không	*Không*		
☐	7	**Mobile**	text	utf8_unicode_ci		Không	*Không*		
☐	8	**Fax**	text	utf8_unicode_ci		Không	*Không*		
☐	9	**TableID**	text	utf8_unicode_ci		Không	*Không*		
☐	10	**Notes**	text	utf8_unicode_ci		Không	*Không*		

b) Bảng lưu trữ trạng thái thiết bị

- Bảng này sử dụng để lưu trữ dữ liệu trạng thái thiết bị trên cloud.
- Mỗi khi thiết bị gửi thông tin trạng thái lên cloud, backend phân tích và tạo bản ghi mới hoặc cập nhật vào bản ghi trạng thái đã có của thiết bị. Mỗi thiết bị sẽ chỉ lưu một bản ghi mới nhất có thay đổi so với trước đó ở bảng này. Trạng thái cũ được cập nhật trên một bảng log có cấu trúc tương tự phục vụ cho phân tích, khai thác sau ở phiên bản nâng cấp.

Bảng 4: Bảng dữ liệu trạng thái thiết bị trên backend

	#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm
☐	1	id	bigint(20)			Không	*Không*		AUTO_INCREMENT
☐	2	deviceID	text	utf8_general_ci		Không	*Không*	Device ID	
☐	3	deviceSE	text	utf8_general_ci		Không	*Không*	Device security code	
☐	4	data	text	utf8_general_ci		Không	*Không*	Data, status of device	
☐	5	updateTime	timestamp		on update CURRENT_TIMESTAMP	Không	CURRENT_TIMESTAMP		ON UPDATE CURRENT_TIMESTAMP
☐	6	changeInterval	int(11)			Không	*Không*		
☐	7	clientTime	varchar(12)	utf8_general_ci		Không	*Không*	Time from client	
☐	8	description	text	utf8_general_ci		Không	*Không*		

c) Bảng thông tin điều khiển

- Bảng này được thiết kế để lưu trữ các lệnh điều khiển gửi tới thiết bị.
- Ban đầu, lệnh điều khiển được lưu trữ trên bảng. Định kỳ thiết bị kiểm tra và lấy lệnh điều khiển để thực hiện. Sau khi thực hiện xong sẽ cập nhật xác nhận đã thực hiện lệnh lên bảng (trường visited).

Bảng 5: Bảng thông tin điều khiển thiết bị trên backend

	#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm
☐	1	id	bigint(20)			Không	*Không*		AUTO_INCREMENT
☐	2	time	timestamp		on update CURRENT_TIMESTAMP	Không	CURRENT_TIMESTAMP		ON UPDATE CURRENT_TIMESTAMP
☐	3	data	text	utf8_general_ci		Không	*Không*		
☐	4	visited	int(11)			Không	0		
☐	5	deviceID	text	utf8_general_ci		Không	*Không*	Device ID	
☐	6	deviceSE	text	utf8_general_ci		Không	*Không*	Device security	

1.3.2.2.2. Giao thức tương tác với Backend

a) Cập nhật trạng thái thiết bị

Cú pháp:

API-URL?LoginPara&CMD&DeviceID&DeviceSE&Data

Giải thích cú pháp:

LoginPara: tham số đăng nhập hệ thống

CMD: lệnh điều khiển, giá trị là “UPST”

DeviceID: mã thiết bị báo cháy

DeviceSE: mã bảo mật của thiết bị báo cháy (được thay đổi theo thời gian, chủ động bởi thiết bị).

Data: dữ liệu trạng thái thiết bị, ngăn cách các trường bằng dầu chấm phẩy (;).

Kết quả trả về "OK" nếu thực hiện thành công.

b) Kiểm tra lệnh điều khiển

Cú pháp:

API-URL?LoginPara&CMD&DeviceID

Giải thích cú pháp:

LoginPara: tham số đăng nhập hệ thống

CMD: lệnh điều khiển, giá trị là "RCTA"

DeviceID: mã thiết bị báo cháy

Kết quả trả về là dữ liệu của lệnh điều khiển mới nhất chưa được xử lý kèm ID của lệnh dưới dạng xâu văn bản.

c) Đọc trạng thái thiết bị

Cú pháp:

API-URL?LoginPara&CMD&DeviceID

Giải thích cú pháp:

LoginPara: tham số đăng nhập hệ thống

CMD: lệnh điều khiển, giá trị là "RSTA"

DeviceID: mã thiết bị báo cháy

Kết quả trả về là trạng thái mới nhất của thiết bị được cập nhật lên cloud.

d) Gửi lệnh điều khiển

Cú pháp:

API-URL?LoginPara&CMD&DeviceID&DeviceSE&Data

Giải thích cú pháp:

LoginPara: tham số đăng nhập hệ thống

CMD: lệnh điều khiển, giá trị là "UCCT"

DeviceID: mã thiết bị báo cháy

DeviceSE: mã bảo mật của thiết bị báo cháy (mới nhất/hiện thời).

Data: chứa dữ liệu lệnh điều khiển, ngăn cách các trường bằng dầu chấm phẩy (;) nếu có nhiều trường thông tin.

Kết quả trả về “OK” nếu thực hiện thành công.

1.3.2.2.3. Các thuật toán xử lý và chương trình điều khiển của Backend

a) Cập nhật trạng thái thiết bị

Thiết bị định kỳ cập nhật trang thái lên cloud thông qua API với lệnh UPST. Module xử lý sau khi nhận được lệnh sẽ tiến hành phân tích dữ liệu trạng thái được gửi lên và cập nhật lên cơ sở dữ liệu.

Trích đoạn code xử lý của module:

Bảng 6: Module chương trình xử lý cập nhật trạng thái thiết bị

```
        if (($cmd=="UPST")&&($deid!="")&&($dese!="")&&($LogOK>0)&&($data != ""))
        {
                $sql = "SELECT deviceID,deviceSE,data,changeInterval FROM " . $tableHardDevices . " WHERE (deviceID='" . $deid. "')";
                $result = $conn->query($sql);
                $changeInterval=0;

                if ($result->num_rows > 0)
                {//Cap nhat cache neu da co
                        $oldData="";
                        $oldDese="";
                        if($row = $result->fetch_assoc())
                        {
                                $oldData=$row["data"];
                                $oldDese=$row["deviceSE"];
                                $changeInterval=$row["changeInterval"];
                        }
                        $dataChanged=0;
                        if (strcmp($oldData,$data)!=0) $dataChanged++;
                        if ($dataChanged==0)
                                if (strcmp($oldDese,$dese)!=0) $dataChanged++;

                        if ($dataChanged)
                        {
                                $changeInterval++;
                                //Cập nhật thông tin data và SE thay đổi nếu có vào bảng vật lý
                                $conn->query("UPDATE " . $tableHardDevices . " SET changeInterval=". $changeInterval .", data='".$data."', deviceSE='".$dese."', clientTime='".$deti."' WHERE (deviceID='". $deid. "')");
                                //Bổ sung bản ghi log vao log trạng thái thiết bị
```

```
                                $conn->query("INSERT INTO " . $tableHardDevicesLog . "
(data,deviceID,deviceSE,clientTime) VALUES ('" . $data . "','".$deid."','".$dese."','".$deti."')");
                                //echo "Cap nhat them";
                        }
                        else
                                $conn->query("UPDATE " . $tableHardDevices . " SET
changeInterval=". $changeInterval .", deviceSE='".$dese."', clientTime='".$deti."' WHERE
(deviceID='". $deid. "')");

                }
                else
                {//Tao ban ghi moi neu lan dau, chua co du lieu
                        $conn->query("INSERT INTO " . $tableHardDevices . "
(data,deviceID,deviceSE,changeInterval,clientTime) VALUES ('" . $data . "','".$deid."','".$dese."',".
$changeInterval .",'".$deti."')");
                        //echo "Tao moi";
                }

           echo "OK";
                exit();
        }
```

b) Kiểm tra lệnh điều khiển

Module này có chức năng nhận yêu cầu từ phía client kiểm tra có lệnh điều khiển nào được đẩy lên cloud hay không. Nếu có sẽ gửi lệnh kèm ID nhận dạng về cho client thực hiện.

Client nhận lệnh, phân tích, xử lý sau đó gửi xác nhận đã thực hiện lệnh theo ID mà backend đã gửi về.

Bảng 7: Module chương trình kiểm tra lệnh điều khiển

```
        if (($cmd == "RCTA") &&($deid!="")&&($LogOK>0))
        {
                //Lay ve trang thai dieu khien
                $sql = "SELECT  * FROM " . $tableDown . " WHERE (deviceID='"  . $deid .
"')AND(visited =0)  ORDER BY time DESC";
                $result = $conn->query($sql);
                if ($result->num_rows > 0)
                {
                        while($row = $result->fetch_assoc())
                        {
```

```
                                echo "ID=" . $row["id"]. ";TIME=" . $row["time"] . ";DID=" .
$row["deviceID"] . ";DSE=" . $row["deviceSE"] . ";DATA=" . $row["data"];
                                exit();
                        }
                }
        }
```

c) Đọc trạng thái thiết bị

Module này cung cấp API cho client, App lấy thông tin trạng thái thiết bị. Thông tin được gửi về bao gồm: các số liệu của thiết bị gửi lên, thời gian cập nhật, và chỉ số cho biết có sự thay đổi dữ liệu hay không.

Các tham số thể hiện như sau:

- DATA: số liệu của thiết bị, chứa đựng các thông tin chi tiết về trạng thái thiết như số đo công suất, tình trạng các cảm biến…
- ChangedInterval: chỉ số ghi nhận có sự thay đổi dữ liệu. Nếu dữ liệu do thiết bị gửi lên khác biệt so với dữ liệu có trước đó thì chỉ số này tăng giá trị. Dựa vào giá trị này mà client khi nhận được dữ liệu gửi về sẽ biết có sự thay đổi dữ liệu hay không để phân tích, xử lý.

Bảng 8: Module chương trình đọc trạng thái thiết bị

```
        if (($cmd=="RSTA")&&($deid!="")&&($LogOK>0))
        {
                if ($uid=="") $uid="0";
                $sql = "SELECT  * FROM " . $tableHardDevices . " WHERE (deviceID='" . $deid .
"')";
                $result = $conn->query($sql);
                if ($result->num_rows > 0)
                {
                        while($row = $result->fetch_assoc())
                        {
                                echo "ID=" . $row["id"]. ";updateTime=" . $row["updateTime"] .
";DID=" . $row["deviceID"] . ";DSE=" . $row["deviceSE"] . ";DATA=" . $row["data"] .
";ChangedInterval=" . $row["changeInterval"];
                                exit();
                        }
                }
        }
```

d) Gửi lệnh điều khiển

Client hay App muôn gửi lệnh điều khiển thiết bị sẽ gỏi API được cung cấp bởi module này. Dữ liệu lệnh sẽ được lưu trữ lại trên tableDown và sẽ gửi lại cho thiết bị, client khi khi được yêu cầu.

Bảng 9: Module chương trình gửi lệnh điều khiển

```
        if (($cmd == "UCCT") &&($deid!="")&&($dese!="")&&($LogOK>0)&&($data!=""))
        {
              $conn->query("INSERT INTO " . $tableDown . " (data,deviceID,deviceSE,visited)
VALUES ('" . $data . "','".$deid."','".$dese."',0)");
           echo "OK";
        }
```

1.3.2.3. Xây dựng hệ thống fontend

Hệ thống frontend là phần mềm chạy trên thiết bị di động với chức năng chính là đọc thông tin trạng thái hệ thống báo cháy đang được đồng bộ trên cloud qua giao thức làm việc với backend sau đó hiển thiện cho người dùng tiếp nhận. Cùng với đó, hệ thống tiếp nhận các lệnh điều khiển của người dùng thông qua giao diện trên thiết bị di động để gửi đến thiết bị báo cháy thông qua giao thức làm việc được hỗ trợ từ backend.

Hệ thống frontend được thiết kế và lập trình chạy trên hệ điều hành Android.

1.3.2.3.1. Các chức năng chính

a) Kết nối Internet

Để giao tiếp được với thiết bị, phần mềm cần được cài đặt trên phần cứng có hỗ trợ kết nối Internet (qua Wifi hoặc mạng di động 3G/4G).

b) Đọc trạng thái thiết bị từ bachkend

Việc đọc trạng thái thiết bị được thực hiện định kỳ với tần suất ở đơn vị giây (có thể điều chỉnh) để cập nhật trạng thái mới nhất của thiết bị được cập nhật trên cloud sử dụng để hiển thị trên giao diện người dùng.

Thuật toán:

Hình 38: Lưu đồ thuật toán đọc trạng thái thiết bị từ backend

Trích đoạn code xử lý:

Bảng 10: Module chương trình đọc trang thái thiết bị từ backend

```
private String ReadHTTPData(String... params) {


  String link = params[0]; // Lay link

  Log.e(" LINK  ", link);

  HttpURLConnection urlConnection = null;

  BufferedReader reader = null;
```

```
// Khai báo chuỗi JSON nhận được đưa về dạng String
String forecastJsonStr = null;

try {

    URL url = new URL(link);

    // Tạo yêu cầu gửi lên link
    urlConnection = (HttpURLConnection) url.openConnection();
    urlConnection.setRequestMethod("GET");
    urlConnection.connect(); // mở kết nối

    // Đọc từng dòng inputStream vào thành một String
    InputStream inputStream = urlConnection.getInputStream();
    StringBuffer buffer = new StringBuffer();
    if (inputStream == null) {
        // Không có yêu cầu gì thì trả về rỗng
        return "";
    }
    reader = new BufferedReader(new InputStreamReader(inputStream));

    String line;
    while ((line = reader.readLine()) != null) {
        // Kết quả trả về có dưới dạng JSON ở dưới dạng từng dòng với mỗi kết quả,
        //vì vậy sẽ nối tất cả kết quả trả về thành một chuỗi rồi in ra màn hình
        buffer.append(line + "\n");
    }

    if (buffer.length() == 0) {
        // trường hợp kết quả trả về bị rỗng
        return "";
    }
```

```
        forecastJsonStr = buffer.toString();
        InternetOK=true;
        return forecastJsonStr;
    } catch (IOException e) {
        InternetOK=false;
        Log.e("PlaceholderFragment", "Error ", e);
        // nếu trong quá trình gửi nhận bị lỗi
        return "";
    } finally {
        if (urlConnection != null) {
            urlConnection.disconnect();
        }
        if (reader != null) {
            try {
                reader.close();
            } catch (final IOException e) {
                Log.e("PlaceholderFragment", "Error closing stream", e);
            }
        }
    }
}
```

c) Hiển thị trạng thái thiết bị trên giao diện

Thuật toán:

- Nếu dữ liệu trạng thái rỗng hiển thị thông báo nhắc kiểm tra kết nối Internet và kết thúc.
- Phân tích các trường thông tin theo ký tự phân tách (;).
- Mỗi trường tách lấy tên và dữ liệu
- Dựa trên tên trường xử lý dữ liệu tương ứng và hiển thị trên giao diện

Hình 39: Lưu đồ thuật toán hiển thị trạng thái thiết bị lên giao diện

Trích đoạn code xử lý:

Bảng 11: Chương trình hiển thị trạng thái thiết bị trên giao diện frontend

```
private void UpdateScreen(String s)

{

   try {

      if (s.length()<1) {

         if (lastOK==true)

            Toast.makeText(FireAlarm.this, "Không tìm thấy máy chủ. Hãy kiểm tra kết nối
INTERNET.", Toast.LENGTH_LONG).show();//PNH

         lastOK=false;
```

```
            return;
        }
        lastOK=true;


        String[] items = s.split(";");
        int GP = 0;
        String Sim = "";
        String RLM = "000000";
        String RLS = "000000";
        String DCM = "0000";
        String DCS = "1111";


        for (String item : items) {
            String[] itemsData = item.split("=");
            String Data = itemsData[1];
            switch (itemsData[0].toUpperCase()) {
                //Cong suat
                case "P1":
                    P1 = Data;
                    break;
                case "U1":
                    U1 = Data;
                    break;
                case "I1":
                    I1 = Data;
                    break;
                case "P2":
                    P2 = Data;
                    break;
                case "U2":
                    U2 = Data;
                    break;
```

```
            case "I2":
               I2 = Data;
               break;
            case "P3":
               P3 = Data;
               break;
            case "U3":
               U3 = Data;
               break;
            case "I3":
               I3 = Data;
               break;

            //GP: GSM
            case "GP":
               if (Data.startsWith("1"))
                  GP = 1;
               else
                  GP = 0;
               break;

            case "SIM":
               if (Data.startsWith("OK"))
                  Sim = " (Đã kết nối)";
               else if (Data.startsWith("NEW"))
                  Sim = " (Mới)";
               else
                  Sim = " (Chưa rõ trạng thái)";
               break;

            case "AS":
               long l = Long.parseLong(Data);
```

```
if (Data.startsWith("0")) {
    tvTrangThai.setText("An toàn");
    tvTrangThai.setTextColor(Color.rgb(0,150,0));//GREEN
    imgAlarm.setImageResource(R.drawable.checkedok);

}
else {
    imgAlarm.setImageResource(R.drawable.alarm);
    if (BigInteger.valueOf(l).testBit(0)) {
        tvTrangThai.setText("Báo cháy (NHIỆT)!");
        tvTrangThai.setTextColor(Color.RED);
    }
    else if (BigInteger.valueOf(l).testBit(1)) {
        tvTrangThai.setText("Báo cháy (KHÓI)!");
        tvTrangThai.setTextColor(Color.RED);
    }
    else if (BigInteger.valueOf(l).testBit(2)) {
        tvTrangThai.setText("Báo cháy (TỪ HỆ THỐNG KHÁC)!");
        tvTrangThai.setTextColor(Color.RED);
    }
    else if (BigInteger.valueOf(l).testBit(3)) {
        tvTrangThai.setText("Báo cháy (THỬ NGHIỆM)!");
        tvTrangThai.setTextColor(Color.RED);
    }
    else if (BigInteger.valueOf(l).testBit(4)) {
        tvTrangThai.setText("Cảm biến mất kết nối (CẢM BIẾN NHIỆT)!");
        tvTrangThai.setTextColor(Color.rgb(255,192,0));
    }
    else if (BigInteger.valueOf(l).testBit(5)) {
        tvTrangThai.setText("Cảm biến mất kết nối (CẢM BIẾN KHÓI)!");
        tvTrangThai.setTextColor(Color.rgb(255,192,0));
```

```
            }
            else if (BigInteger.valueOf(l).testBit(6)) {
                tvTrangThai.setText("Cảnh báo quá tải ĐIỆN (>=70%)!");
                tvTrangThai.setTextColor(Color.rgb(255,192,0));
            }
            else if (BigInteger.valueOf(l).testBit(7)) {
                tvTrangThai.setText("Cảnh báo quá tải ĐIỆN (>=100%)!");
                tvTrangThai.setTextColor(Color.RED);
            }
            else {
                tvTrangThai.setText("Đang cập nhật");
            }

        }
        break;

    case "TIME":
        tvUpdateTime.setText("Lần cập nhật gần nhất: " + Data);
        break;

    case "SS": //Smock State
        if (Data.startsWith("0")) {
            tvTTCamBienKhoi.setText("Mất kết nối");
            tvTTCamBienKhoi.setTextColor(Color.rgb(255,192,0));
        }
        else if (Data.startsWith("1")) {
            tvTTCamBienKhoi.setText("Hoạt động");
            tvTTCamBienKhoi.setTextColor(Color.rgb(0,150,0));//GREEN
        }
        else if (Data.startsWith("2")) {
            tvTTCamBienKhoi.setText("Báo động, có khói");
            tvTTCamBienKhoi.setTextColor(Color.RED);
```

```
            }
            else
              tvTTCamBienKhoi.setText("");

            break;

        case "TS": //Temperature State
            if (Data.startsWith("0")) {
                tvTTCamBienNhiet.setText("Mất kết nối");
                tvTTCamBienNhiet.setTextColor(Color.rgb(255,192,0));
            }
            else if (Data.startsWith("1")) {
                tvTTCamBienNhiet.setText("Hoạt động");
                tvTTCamBienNhiet.setTextColor(Color.rgb(0,150,0));//GREEN
            }
            else if (Data.startsWith("2")) {
                tvTTCamBienNhiet.setText("Báo động, nhiệt cao");
                tvTTCamBienNhiet.setTextColor(Color.RED);
            }
            else
              tvTTCamBienNhiet.setText("");

            break;

        case "RLM": //Relay Mode (000000)
            RLM = Data;
            break;

        case "RLS": //Relay State (000000)
            RLS = Data;
            break;
```

```
            case "DCM": //Dry Contact Mode (0000)
                DCM = Data;
                break;

            case "DCS": //Dry Contact State (0000)
                DCS = Data;
                break;

            case "PS": //Nguồn cấp cho bộ điều khiển
                if (Data.startsWith("1")) {
                    tvNguonCap.setText("Điện Acquy");
                }
                else
                {
                    tvNguonCap.setText("Điện lưới");
                }
                break;

            case "ACH": //Tình trạng sạc điện Acquy
                if (Data.startsWith("1")) {
                    tvSacAcquy.setText("Đang sạc");
                }
                else
                {
                    tvSacAcquy.setText("Không sạc");
                }
                break;

            case "AV": //Nguồn điện Acquy
                tvDienApAcquy.setText(Data+"V");
                break;
        }
```

```
    }


    if (GP == 0) {
        tvSim.setText("Không hỗ trợ");
    } else {
        tvSim.setText("Hỗ trợ" + Sim);
    }


    if (Double.parseDouble(U1)<50)
    {
        I1="0";
    }
    if (Double.parseDouble(U2)<50)
    {
        I2="0";
    }
    if (Double.parseDouble(U3)<50)
    {
        I3="0";
    }


    tvGTPha1.setText(U1 + "V /" + I1 + "A (" + P1 + "KWh)");
    tvGTPha2.setText(U2 + "V /" + I2 + "A (" + P2 + "KWh)");
    tvGTPha3.setText(U3 + "V /" + I3 + "A (" + P3 + "KWh)");


    for (int i = 0; i < RLM.length(); i++) {
        String m = RLM.substring(i, i + 1);
        String s1 = "0";
        String s2 = "đang cập nhật";
        if (i < RLS.length()) s1 = RLS.substring(i, i + 1);
        if (m.startsWith("0"))//Không dung
            s2 = "Không dùng";
```

```
        else if (m.startsWith("1")) //Điều khiển đèn
            s2 = "Điều khiển đèn";
        else if (m.startsWith("2")) //Điều khiển chuông
            s2 = "Điều khiển chuông";
        else if (m.startsWith("3")) //Điều khiển bật thiết bị
            s2 = "Bật thiết bị";
        else if (m.startsWith("4")) //Điều khiển tắt thiết bị
            s2 = "Tắt thiết bị";
        else
            s2 = "Không dùng";
        if (!m.startsWith("0")) {
            if (s1.startsWith("1"))
                s2 += " (BẬT)";
            else
                s2 += " (TẮT)";
        }

        switch (i) {
            case 0:
                tvGTKenh1.setText(s2);
                break;
            case 1:
                tvGTKenh2.setText(s2);
                break;
            case 2:
                tvGTKenh3.setText(s2);
                break;
            case 3:
                tvGTKenh4.setText(s2);
                break;
            case 4:
                tvGTKenh5.setText(s2);
```

```
            break;
        case 5:
            tvGTKenh6.setText(s2);
            break;
    }

}


for (int i = 0; i < DCM.length(); i++) {
    String m = DCM.substring(i, i + 1);
    String s1 = "0";
    String s2 = "đang cập nhật";
    if (i < DCS.length()) s1 = DCS.substring(i, i + 1);
    if (m.startsWith("0"))//Không dung
        s2 = "Không dùng";
    else if (m.startsWith("1")) //Điều khiển đèn
        s2 = "Xác nhận báo cháy";
    else if (m.startsWith("2")) //Điều khiển chuông
        s2 = "Đầu vào báo cháy";
    else
        s2 = "Không dùng";
    if (!m.startsWith("0")) {
        if (s1.startsWith("0"))
            s2 += " (KÍCH HOẠT)";
        else
            s2 += " (TẮT)";
    }

    switch (i) {
        case 0:
            tvGTTiepDiem1.setText(s2);
```

```
                break;
            case 1:
                tvGTTiepDiem2.setText(s2);
                break;
            case 2:
                tvGTTiepDiem3.setText(s2);
                break;
            case 3:
                tvGTTiepDiem4.setText(s2);
                break;
        }

    }
}catch (RuntimeException  e )
{
    Log.e("  Update State Error ", e.getMessage());
}
finally {
    Log.i("  Update State ", " Finally");
}

}
```

1.3.2.3.2. Giao diện

a) Mô tả thiết kế bằng XML

Bảng 12: Thiết kế giao diện hiển thị của frontend

```
<android.support.v4.widget.NestedScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".FireAlarm"
    tools:showIn="@layout/activity_fire_alarm">


    <LinearLayout
        android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:orientation="vertical">


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:background="#feeac6"
        android:gravity="center_vertical"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:text="Trạng thái"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="80dp"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:id="@+id/imgAlarm"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@drawable/checkedok" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/tvTrangThai"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="An toàn"
                android:textColor="#008000" />

            <TextView
                android:id="@+id/tvUpdateTime"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:text="Lần cập nhật gần nhất: 2019-01-28"
                android:textColor="#A0A0A0"
                android:textSize="8dp"
                android:textStyle="italic" />
        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:background="#feeac6"
        android:gravity="center_vertical"
        android:orientation="vertical">
```

```
        <TextView
            android:id="@+id/tvCamBien"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:text="Các cảm biến"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@drawable/cambienkhoi" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/tvCamBienKhoi"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

                android:text="Cảm biến báo khói" />

            <TextView
                android:id="@+id/tvTTCamBienKhoi"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:text="Hoạt động"
                android:textColor="#008000"
                android:textStyle="bold|italic" />
        </LinearLayout>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@drawable/cambiennhiet" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="vertical">
```

```
            <TextView
                android:id="@+id/tvCamBienNhiet"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

                android:text="Cảm biến báo nhiệt" />

            <TextView
                android:id="@+id/tvTTCamBienNhiet"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:text="Mất kết nối"
                android:textColor="#FFC000"
                android:textStyle="bold|italic" />
        </LinearLayout>

    </LinearLayout>

    <!-- Thông số nguồn điện -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:background="#feeac6"
        android:gravity="center_vertical"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:text="Thông số nguồn điện"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="vertical">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Pha 1: " />
                <TextView
                    android:id="@+id/tvGTPha1"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:textStyle="bold"
```

```
                android:textColor="#000080"
                android:text="220V / 10A" />
        </LinearLayout>

        <TextView
            android:id="@+id/tvTTPha1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Bình thường"
            android:textColor="#008000"
            android:textSize="10dp"
            android:textStyle="bold|italic" />
    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/line_height"
    android:layout_marginLeft="5dp"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        app:srcCompat="@android:drawable/presence_online" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Pha 2: " />
            <TextView
                android:id="@+id/tvGTPha2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textStyle="bold"
                android:textColor="#000080"
                android:text="220.5V / 10.5A" />
        </LinearLayout>

        <TextView
            android:id="@+id/tvTTPha2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Bình thường"
            android:textColor="#008000"
            android:textSize="10dp"
            android:textStyle="bold|italic" />
    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/line_height"
    android:layout_marginLeft="5dp"
```

```
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="vertical">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Pha 3: " />
                <TextView
                    android:id="@+id/tvGTPha3"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:textStyle="bold"
                    android:textColor="#000080"
                    android:text="220V / 11A" />
            </LinearLayout>

            <TextView
                android:id="@+id/tvTTPha3"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Bình thường"
                android:textColor="#008000"
                android:textSize="10dp"
                android:textStyle="bold|italic" />
        </LinearLayout>

    </LinearLayout>

    <!-- Các thiết bị điều khiển -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:background="#feeac6"
        android:gravity="center_vertical"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:text="Các thiết bị điều khiển"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">
```

```
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Kênh 1: " />

            <TextView
                android:id="@+id/tvGTKenh1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="ON"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Kênh 2: " />

            <TextView
                android:id="@+id/tvGTKenh2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="OFF"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
```

```
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Kênh 3: " />

            <TextView
                android:id="@+id/tvGTKenh3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Không dùng"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Kênh 4: " />

            <TextView
                android:id="@+id/tvGTKenh4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="OFF (loa)"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
```

```
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Kênh 5: " />

            <TextView
                android:id="@+id/tvGTKenh5"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="OFF (đèn báo)"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Kênh 6: " />

            <TextView
                android:id="@+id/tvGTKenh6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="OFF (phun nước)"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>
```

```
<!-- Nguồn đầu vào tiếp điểm khô -->
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="@dimen/line_normal_height"
   android:background="#feeac6"
   android:gravity="center_vertical"
   android:orientation="vertical">

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="5dp"
      android:text="Các đầu vào ghép nối thiết bị"
      android:textSize="18sp"
      android:textStyle="bold" />

</LinearLayout>

<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="@dimen/line_normal_height"
   android:layout_marginLeft="5dp"
   android:gravity="center_vertical"
   android:orientation="horizontal">

   <ImageView
      android:layout_width="wrap_content"
      android:layout_height="match_parent"
      app:srcCompat="@android:drawable/presence_online" />

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginLeft="10dp"
      android:orientation="horizontal">

      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Đầu vào 1: " />

      <TextView
         android:id="@+id/tvGTTiepDiem1"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="OFF (Hệ báo cháy khác)"
         android:textColor="#000080"
         android:textStyle="bold" />

   </LinearLayout>
</LinearLayout>

<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="@dimen/line_normal_height"
   android:layout_marginLeft="5dp"
   android:gravity="center_vertical"
   android:orientation="horizontal">

   <ImageView
      android:layout_width="wrap_content"
      android:layout_height="match_parent"
      app:srcCompat="@android:drawable/presence_online" />

   <LinearLayout
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Đầu vào 2: " />

            <TextView
                android:id="@+id/tvGTTiepDiem2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="OFF (không dùng)"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Đầu vào 3: " />

            <TextView
                android:id="@+id/tvGTTiepDiem3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="OFF (không dùng)"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />
```

```
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Đầu vào 4: " />

            <TextView
                android:id="@+id/tvGTTiepDiem4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="OFF (không dùng)"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>
    </LinearLayout>

    <!-- Thông tin SIM -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:background="#feeac6"
        android:gravity="center_vertical"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:text="Tình trạng SIM"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@drawable/simcard" />

        <TextView
            android:id="@+id/tvSim"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:text="Kết nối bình thường"
            android:textColor="#000080"
            android:textStyle="bold" />

    </LinearLayout>


    <!-- Nguồn nuôi thiết bị -->
```

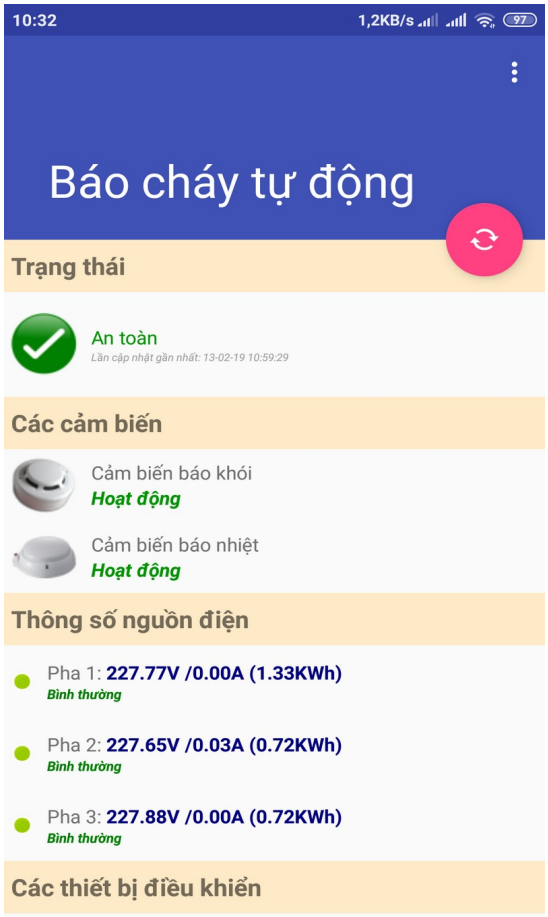
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/line_normal_height"
    android:background="#feeac6"
    android:gravity="center_vertical"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:text="Nguồn cung cấp và Acquy"
        android:textSize="18sp"
        android:textStyle="bold" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/line_normal_height"
    android:layout_marginLeft="5dp"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        app:srcCompat="@android:drawable/presence_online" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nguồn cung cấp: " />

        <TextView
            android:id="@+id/tvNguonCap"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Điện lưới"
            android:textColor="#000080"
            android:textStyle="bold" />

    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/line_normal_height"
    android:layout_marginLeft="5dp"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        app:srcCompat="@android:drawable/presence_online" />

    <LinearLayout
        android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Điện áp Acquy: " />

            <TextView
                android:id="@+id/tvDienApAcquy"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="12V"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/line_normal_height"
        android:layout_marginLeft="5dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:srcCompat="@android:drawable/presence_online" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Tình trạng sạc Acquy: " />

            <TextView
                android:id="@+id/tvSacAcquy"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Không sạc"
                android:textColor="#000080"
                android:textStyle="bold" />

        </LinearLayout>

    </LinearLayout>


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:gravity="center"
        android:orientation="vertical">

    </LinearLayout>
```

```
    </LinearLayout>
</android.support.v4.widget.NestedScrollView>
```

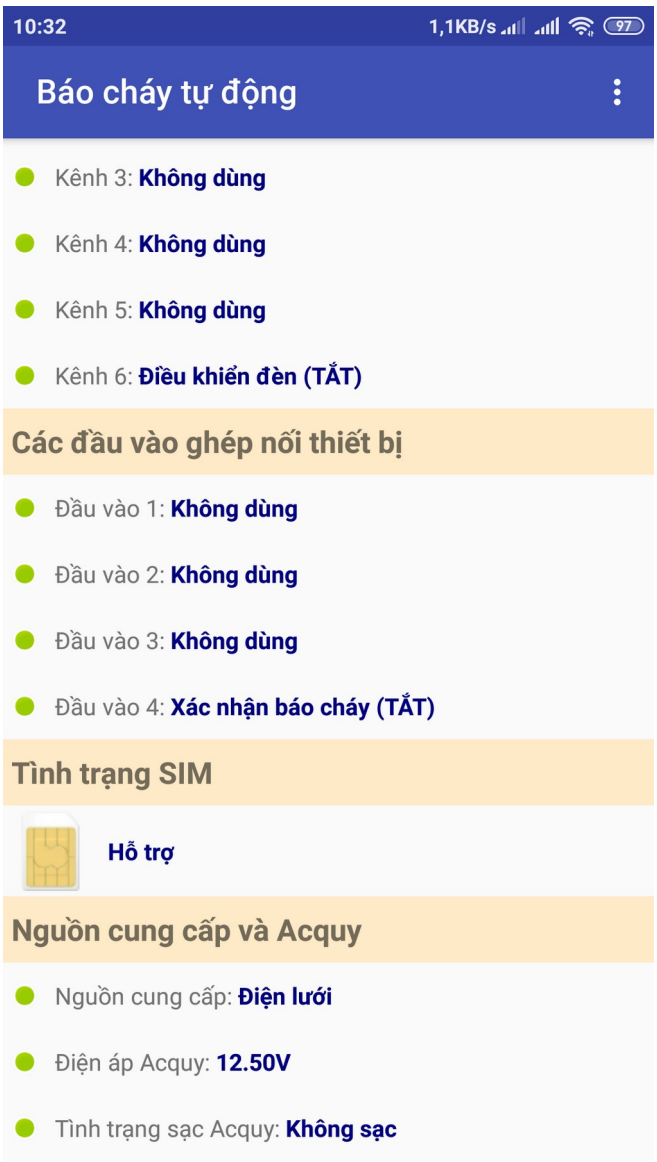
b) Hình ảnh hiển thị

Hình ảnh giao diện thực tế:



Hình 40: Giao diện hiển thị trên Mobile App (1)

Hình 41: Giao diện hiển thị trên Mobile App (2)

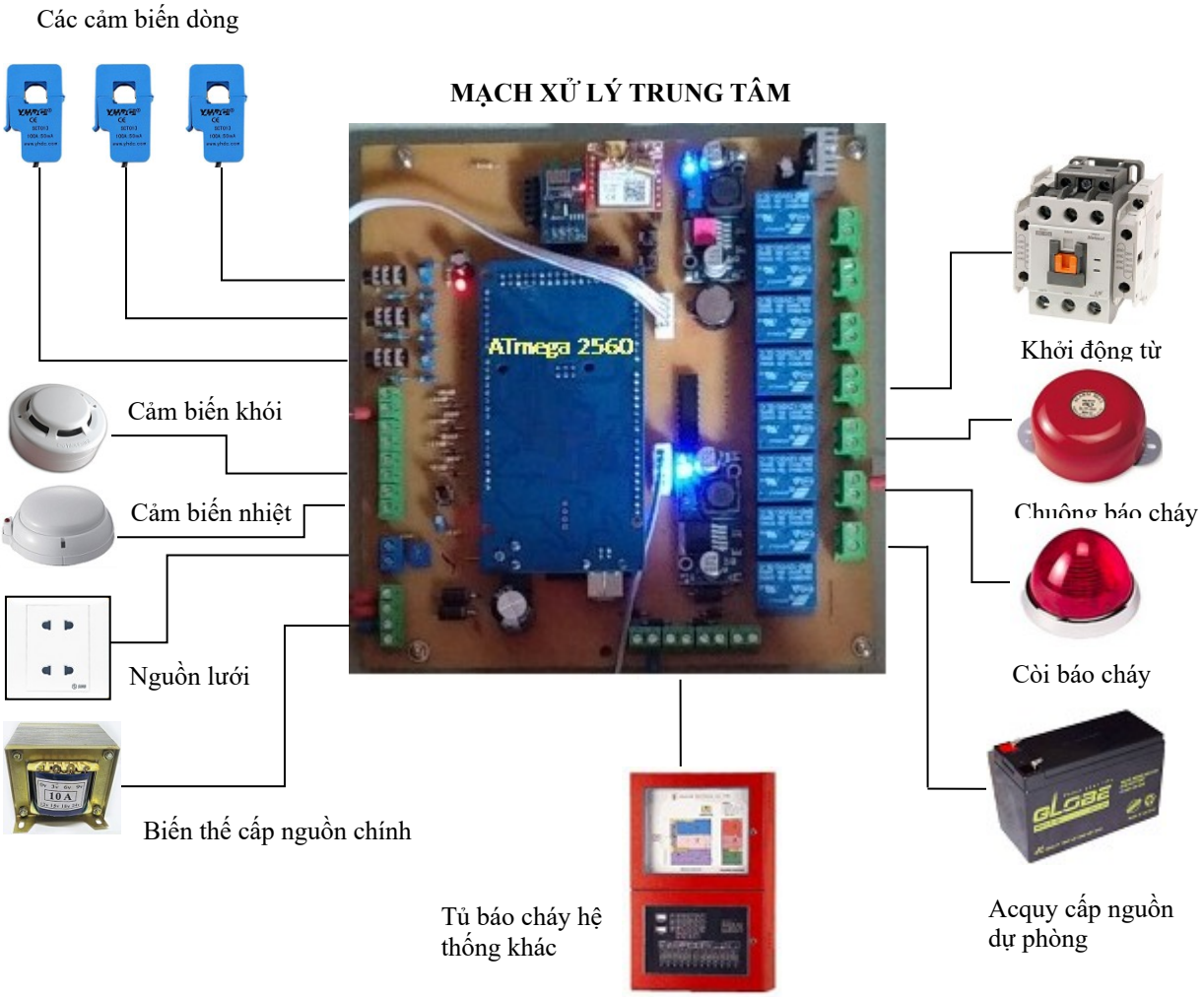


Hình 42: Giao diện hiển thị trên Mobile App (3)

1.3.3. GHÉP NỐI, THỬ NGHIỆM VÀ HOÀN THIỆN HỆ THỐNG

1.3.3.1. Ghép nối hệ thống

Sơ đồ ghép nối các thành phần:



Hình 43: Sơ đồ ghép nối các thành phần của hệ thống Báo cháy tự động

Board mạch xử lý trung tâm, các cảm biến, hệ thống LCD và các nút điều khiển, đèn báo đã được lắp ráp kết nối với nhau trên tủ báo cháy tự động.

Cảm biến khói được lắp trên kênh cảm biến thứ nhất để để phát hiện khói và gửi tín hiệu về cho bộ xử lý trung tâm. Kênh giao tiếp cảm biến khói tương thích để làm việc với các loại cảm biến khói chuyên dụng với tín hiệu báo khói tương ứng với dòng điện điều khiển từ 4mA đến 30mA, điện áp làm việc 12-24V. Cảm biến khói được sử dụng trong hệ thống là cảm biến AH-0311-2 phát hiện khói bằng công nghệ quang với dòng làm việc là 30mA, điện áp 24V.

Cảm biến nhiệt có chức năng phát hiện cháy dưới dạng tăng nhiệt được lắp trên kênh cảm biến đầu vào số 2. Đây là loại cảm biến hoạt động theo nguồn dòng tương tự cảm biến khói với điện áp làm việc là 24V, dòng từ 4mA-30mA. Ở chế độ bình thường, cảm biến khói và cảm biến nhiệt chỉ tiêu thụ một công suất rất nhỏ. Dòng duy trì 40uA.

Để đảm bảo kết nối được liên tục và các cảm biến đang ở trạng thái vận hành, khi đấu nối cần lưu ý lắp điện trở 2.2K ở cảm biến cuối cùng trong chuỗi cảm biến sử dụng trên mỗi kênh. Nếu thiếu điện trở này hệ thống sẽ báo lỗi cảm biến. Nếu cảm biến được tháo ra khỏi đế, hệ thống cũng thực hiện cảnh báo lỗi cảm biến.

Chuông và còi báo cháy cần được cấp nguồn 12-24VDC đóng ngắt bởi rơ le được lựa chọn trong cấu hình thiết bị báo cháy. Vai trò rơ le như công tắc cấp điện cho các thiết bị này vận hành.

Khởi động từ 3 pha được thiết kế để tự động ngắt nguồn điện của các thiết bị điện cần được bảo vệ, ngắt điện khi có sự cố cháy xảy ra. Cuộn hút khởi động từ được cấp nguồn thông qua tiếp điểm đóng ngắt của một rơ le đầu ra của hệ thống báo cháy.

1.3.3.2. Vận hành thử nghiệm

a) Kiểm tra hoạt động của cảm biến báo khói

- Cảm biến khói cần lắp đúng như sơ đồ đấu nối.
- Cấp điện cho tủ báo cháy vận hành.

- Tạo nguồn khói tác động đến cảm biến. Khi lượng khói đủ lớn tác động, cảm biến kích hoạt báo có khói. Khi đó đèn báo trên cảm biến sáng đỏ. Hệ thống báo cháy nháy đèn báo trên mặt tủ điều khiển đồng thời kích hoạt các chế độ cảnh báo qua tin nhắn, gọi điện (nếu đã được thiết lập). Màn LCD trên mặt tủ hiển thị thông tin trạng thái báo cháy với nguồn được phát hiện từ cảm biến khói.
- Để ngưng tác động báo cháy do khói từ cảm biến khói cần ngắt cảm biến (tháo ra khỏi đế) sau đó lắp trở lại. Nhấn nút xác nhận trên mặt tủ (nút mầu đỏ)

b) Kiểm tra hoạt động của cảm biến nhiệt

- Cảm biến nhiệt cần lắp đúng như sơ đồ đấu nối.
- Cấp điện cho tủ báo cháy vận hành.
- Tạo nguồn nhiệt tác động đến cảm biến (có thể dùng máy sấy tóc thổi nhiệt lên cảm biến). Khi lượng nhiệt đủ lớn tác động, cảm biến kích hoạt báo có sự gia tăng nhiệt (ở mức báo cháy bởi nhiệt). Khi đó đèn báo trên cảm biến sáng đỏ. Hệ thống báo cháy nháy đèn báo trên mặt tủ điều khiển đồng thời kích hoạt các chế độ cảnh báo qua tin nhắn, gọi điện (nếu đã được thiết lập). Màn LCD trên mặt tủ hiển thị thông tin trạng thái báo cháy với nguồn được phát hiện từ cảm biến nhiệt.
- Để ngưng tác động báo cháy do nhiệt cần giảm nhiệt trên cảm biến (để tự nhiên sau thời gian nhiệt độ từ giảm xuống đến ngưỡng an toàn và hết sự kiện báo cháy). Nhấn nút xác nhận trên mặt tủ (nút mầu đỏ).

c) Kiểm tra kết nối của cảm biến nhiệt và cảm biến khói

- Mục đích là kiểm tra hệ thống có phát hiện được cảm biến bị vô hiệu hóa (đứt dây, không được lắp đặt) hay không. Nếu hệ thống không phát hiện được tình trạng kết nối của cảm biến thì không đảm bảo độ tin cậy hoạt động.
- Để thử nghiệm hãy lần lượt tháo từng cảm biến khỏi đế lắp và quan sát phản ứng của hệ thống. Trong điều kiện hoạt động bình thường, hệ thống sẽ phát hiện cảm biến bị mất kế nối và kích hoạt đèn nháy báo hiệu trên tủ đồng thời hiển thị thông tin mất kế nối tới cảm biến để người giám sát có thể phát hiện và kiểm tra cảm biến liên quan.

d) Kiểm tra kết nối với hệ thống báo cháy khác

- Hệ thống báo cháy khác có thể được nối với hệ thống báo cháy tự động thông qua tiếp điểm khô.

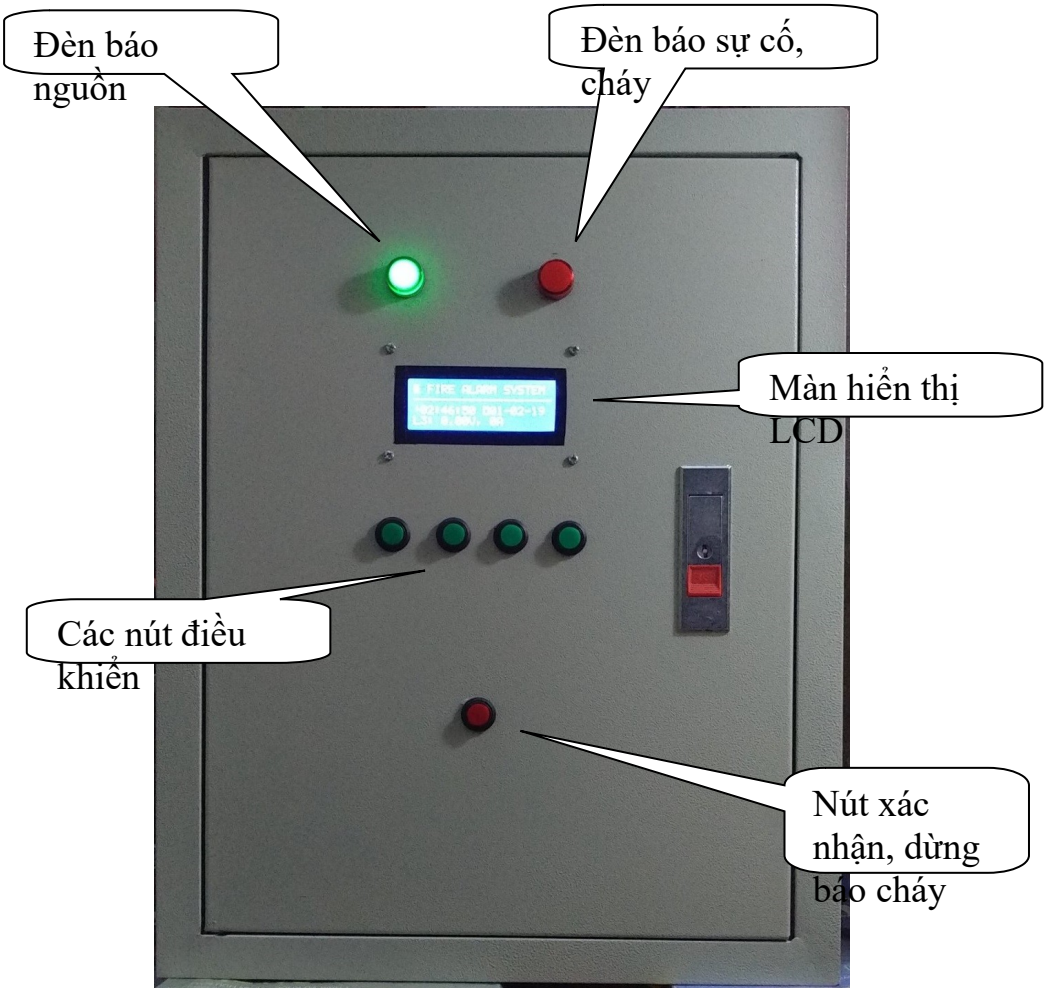
- Để thử nghiệm cần thiết lập cho một tiếp điểm khô nào đó nhận thông tin báo cháy từ hệ thống báo cháy khác. Tiếp theo thực hiện nối thông 2 đầu tiếp điểm kết nối. Hệ thống sẽ phát hiện tín hiệu báo cháy gửi đến và kích hoạt chế độ báo cháy (nháy đèn, điều khiển thiết bị, gửi tin nhắn, gọi điện…)

1.3.3.3. Sản phẩm hoàn chỉnh sau hiệu chỉnh

Sản phẩm hoàn chỉnh là một tủ Báo cháy tự động tích hợp mạch điều khiển trung tâm, hệ thống kết nối mạng không dây Wifi, mạng di động GSM, hệ thống hiển thị tại chỗ (LCD), đèn báo, các nút nhấn để thiết lập trạng thái cùng biến thế nguồn và acquy dự phòng.

Cảm biến dòng, còi và đèn báo cháy được lắp đặt bên ngoài tủ và kết nối với tủ qua cable nối.

Hình ảnh tủ Báo chạy tự động hoàn chỉnh:



Hình 44: Tủ điều khiển báo cháy tự động

- Đèn báo nguồn: mầu xanh, luôn sáng khi hệ thống hoạt động, được cấp nguồn từ điện lưới hoặc bình Acquy.

- Đèn báo sự cố: nháy sáng khi có sự cố (báo cháy, báo mất kết nối cảm biến).

- Màn hình LCD: hiển thị thạng thái hoạt động của tủ (báo cháy hay không, trạng thái từng cảm biến, các số đo nguồn điện được giám sát, tình trạng nguồn điện dự phòng acquy (điện áp, có đang sạc điện hay không), nguồn đang sử dụng cho hệ thống (điện lưới, điện từ bình Acquy)).

- Các nút điều khiển: dùng để thiết lập thông số vận hành cho tủ báo cháy tự động. Người dùng kết hợp sử dụng LCD để thực hiện các thiết lập cho hệ thống.

- Nút xác nhận, dừng báo cháy: khi có tín hiệu báo cháy hoặc sự cố kết nối cảm biến, đèn báo nhấp nháy… người vận hành có thể nhấn nút này để xác nhận đã biết thông tin và dự quá trình báo sự cố, báo cháy lại.

Với việc thiết kế hệ thống báo cháy bao gồm các khối giao tiếp và điều khiển trên cho phép hệ thống thực hiện các cảnh báo tạo chỗ, điều khiển thiết bị đầu cuối; bộ xử lý trung tâm đóng vai trò quan trọng trong giao tiếp với phần mềm giám sát điều khiển qua cloud. Tiếp nhận điều khiển tử xa để đóng/ngắt nguồn điện cần bảo vệ, thay đổi chế độ vận hành. Tiếp nhận tín hiệu báo cháy từ hệ thống báo chạy hiện có khác nhằm tối đa hóa hiệu quả phòng và chữa cháy, tận dụng được các hệ thống hiện có mà không phải thay thế hoàn toàn.

2. THẢO LUẬN

2.1. Thảo luận 1

Đề tài được triển khai dựa trên việc nghiên cứu tỉ mỉ, kiến trúc cũng như cách thức vận hành hệ thống phát hiện và báo cháy tiêu biểu hiện có trên thị trường. Trên cơ sở phân tích các ưu, nhược điểm của các hệ thống báo cháy hiện có, nhóm tác giả đưa ra đề xuất ý tưởng mới nhằm xây dựng một hệ thống phát hiện và báo cháy hiệu quả hơn, khắc phục được một số hạn chế và bổ sung các tính năng giúp hệ thống báo cháy mới hoạt động hiệu quả hơn. Tiêu biểu có thể thấy ở hệ thống báo cháy hiện có tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên:

- Hệ thống báo cháy hiện có chưa tích hợp đầy đủ tính năng báo cháy từ xa. Hệ thống có hỗ trợ thì mới dừng lại ở mức cảnh báo qua tin nhắn, gọi điện. Việc giám sát thường xuyên vẫn cần tiến hành trực tiếp trên hệ thống báo cháy.

- Một số sản phẩm có tính năng giám sát thông số vận hành thông qua phần mềm trên máy tính có kết nối mạng với hệ thống báo cháy.

- Các hệ thống chưa tích hợp các tính năng phát hiện sớm nguy cơ cháy bắt nguồn từ quá tải điện. Chưa tích hợp tính năng điều khiển cho phép ngắt nguồn điện cho các thiết bị cần bảo vệ.

Hệ thống báo cháy tự động theo nghiên cứu, đề xuất của đề tài không những thực hiện được đầy đủ tính năng phát hiện cháy thông qua các dấu hiệu như tăng nhiệt (sử dụng đầu dò/cảm biến nhiệt), phát hiện khói (thông qua đầu dò/cảm biến khói). Hệ thống còn tích hợp tính năng giám sát công suất tiêu thụ nguồn điện chính tại nơi cần bảo vệ, phòng cháy xảy ra do nguyên nhân quá tải điện. Đây là một trong những nguyên nhân thường trực gây sự cố cháy. Hệ thống gửi tới người dùng cảnh báo khi công suất tiêu thụ vượt ngưỡng giới hạn đã đặt trược để giúp người quản lý sớm phát hiện nguy cơ cháy do quá tải dẫn tới cháy, chập điện. Tính năng giám sát từ xa qua phần mềm trên thiết bị di động cũng là một ưu điểm lớn của sản phẩm. Như vậy người giám sát có thể không cần ngồi trước một máy tính kết nối cố định tới hệ thống báo cháy mà thông qua kết nối Internet là có thể giám sát cho dù ở cách xa hệ thống. Các tính năng cảnh báo tại chỗ, cảnh báo qua tin nhắn, gọi điện cũng được tích hợp đầy đủ giúp việc cảnh báo được thực hiện nhanh và hiệu quả.

Qua kết quả thể hiện ở giai đoạn hiện tại cho thấy sản phẩm có khá nhiều các ưu điểm. Nhóm tác giả nên thúc đẩy hoàn thiện sản phẩm để sớm đưa vào ứng dụng trong thực tiễn.

Bên cạnh các ưu điểm nêu trên, nhóm tác giả nên xem xét bổ sung, cải tiến một số tính năng:

- Hỗ trợ người dùng có thể kiểm tra số dư tài khoản của thẻ SIM trực tiếp trên giao diện bộ điều khiển và nạp tiền qua việc nhập mã thẻ cào mà không cần phải tháo SIM ra khỏi hệ thống. Như vậy sẽ giúp việc duy trì hệ thống được thuận lợi hơn. Đảm bảo thẻ SIM luôn ở trạng thái sẵn sàng, có số dư tài khoản đủ để thực hiện được các cảnh báo qua tin nhắn hay gọi điện. Việc thao tác của nhân viên giám sát, bảo trì cũng thuận lợi hơn.

- Hệ thống điều khiển hỗ trợ nhiều đầu ra (sử dụng rơ le) để điều khiển thiết bị. Việc lựa chọn kênh điều khiển nên để người dùng thiết lập một cách linh hoạt trên giao diện tương tác ngay tại tủ điều khiển.

2.2. Thảo luận 2

Đề tài định hướng nghiên cứu giải quyết một vấn đề thực tiễn và có tính ứng dụng cao. Phương pháp nghiên cứu được nhóm tác giả lựa chọn là rất phù hợp để triển khai các nội dung nghiên cứu và hướng tới sản phẩm mục tiêu của đề tài.

Nhóm tác giả đã chỉ ra được các mặt hạn chế, tồn tại của một số hệ thống báo cháy được sử dụng phổ biến trong nước đồng thời đề xuất được giải pháp khắc phục nhằm thiết kế ra hệ thống báo cháy mới hoàn thiện hơn.

Để tài đã được triển khai dựa trên việc nghiên cứu kiến trúc đặc trưng của một hệ thống báo cháy tự động, các tiêu chuẩn liên quan theo hệ thống tiêu chuẩn của Việt Nam trước khi tiến hành các thiết kế, đề xuất cải tiến để cho ra đời một hệ thống báo cháy hoạt động hiệu quả, có độ tin cậy cao hơn.

Có thể chỉ ra một số ưu điểm của hệ thống báo cháy tự động, sản phẩm của đề tài nghiên cứu như sau:

- Tương thích, sử dụng được các loại cảm biến phát hiện dấu hiệu cháy thông qua hiện tượng tăng nhiệt, có khói được dùng phổ biến trên thị trường, đa phần các hệ thống báo cháy khác đang sử dụng. Như vậy việc lắp lẫn, triển khai hệ thống được thuận lợi mà không phải tìm cảm biến đặc biệt dành riêng cho một hệ thống cho hiệu quả triển khai được thuận lợi hơn.

- Thực hiện cảnh báo tại chỗ qua hệ thống đèn, còi

- Hỗ trợ tính năng cảnh báo qua tín nhắn, gọi điện giúp việc cảnh báo được thực hiện kịp thời hiệu quả.

- Một ưu điểm lớn của của hệ thống là giám sát công suất tiêu thụ của nguồn điện thuộc đối tượng cần giám sát, là nguyên nhân tiềm ẩn gây cháy khi có hiện tượng quá tải điện xảy ra. Khi có biến động bất thường, hệ thống kịp thời gửi cảnh báo tới người có trách nhiệm thông qua tin nhắn hay trực tiếp trên phần mềm giám sát.

- Có khả năng kết nối với hệ thống báo cháy khác giúp nâng cao hiệu quả hoạt động. Khi trang bị hệ thống báo cháy tự động này không cần thiết phải thay thế hệ thống báo cháy cũ nếu vẫn vận hành tốt mà cho phép ghép nối, tích hợp với hệ thống mới tạo nên hệ thống báo cháy lớn, tận dụng được đầu tư trước đó, giảm phát sinh chi phí.

Ưu điểm khác cũng rất đáng kể của sản phẩm này là khả năng điều khiển lịch hoạt, hỗ trợ nhiều đầu ra điều khiển qua rơ le. Như vậy có thể sử dụng các đầu ra này điều khiển các thiết bị khác hoạt động khi có sự cố cháy như kích hoạt hệ thống báo cháy, chữa cháy tại chỗ, ngắt nguồn điện của thiết bị cần bảo vệ.

Ưu điểm của hệ thống báo cháy tự động mới là rất rõ ràng và có tính ứng dụng, rất khả thi để triển khai trong thực tế.

Để giúp cho việc vận hành hệ thống được thuân lợi và tăng tính tin cậy hơn nữa, nhóm tác giả nên cân nhắc một số góp ý sau:

- Hệ thống báo cháy đã được thiết kế nguồn dự phòng là một bình acquy đồng thời tích hợp tính năng tự động sạc điện cho bình acquy. Như vậy sẽ đảm bảo hệ thống vẫn vận hành được ngay cả khi mất nguồn cung cấp chính từ điện

lưới. Để tăng độ bền cho bình acquy, nhóm tác giả nên tích hợp tính năng bảo dưỡng bình bằng cách định kỳ vận hành từ nguồn điện của bình sau đó nạp lại điện cho bình.

- Các thông số tình trạng điện của bình, nguồn đang sử dụng nên thể hiện luôn phiên ngay trên giao diện màn LCD có trên bộ điều khiển để người vận hành thuận tiện trong việc giám sát hệ thống.

IV. KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Đề tài đã hoàn thành đáp ứng đầy đủ mục tiêu cũng như các nội dung nghiên cứu đã đặt. Sản phẩm của đề tài là Hệ thống báo cháy tự động bao gồm phần cứng là tủ báo chạy tự động và phần mềm xử lý dữ liệu trên cloud và ứng dụng cài đặt trên thiết bị di động.

Phần cứng được thiết kế đầy đủ các tính năng hoạt động đạt các yêu cầu đặt ra cho hệ thống:

- Giao tiếp với hệ thống cảm biến báo cháy dựa trên phát hiện khói, phát hiện gia tăng nhiệt.

- Thực hiện cảnh báo tại chỗ thông qua hệ thống chuông và đèn báo hiệu.

- Cảnh báo cháy và sự cố kết nối cảm biến qua tin nhắn SMS và cuộc gọi không số giúp người giám sát tiếp nhận nhanh chóng, kịp thời thông báo và ra quyết định xử lý.

- Tiếp nhận được thông tin báo cháy từ thiết bị báo cháy khác qua tiếp điểm khô giúp tận dụng được các hệ thống có sẵn, gia tăng khả năng phát hiện và báo cháy, không bắt buộc thay thế giúp tiết kiệm chi phí.

- Giám sát điện áp, công suất tiêu thụ của nguồn điện cung cấp cho thiết bị, phụ tải chính khu vực cần bản vệ. Từ đó phát hiện biến động bất thường về công suất tiêu thụ để cảnh báo quá tải, nguy cơ cháy liên quan đến quá tải điện.

- Có khả năng ngắt nguồn điện cho phụ tải khu vực cần bảo vệ.

- Kết nối Internet qua Wifi cập nhật trạng thái thiết bị lên cloud.

- Có hệ thống nguồn dự phòng đảm bảo hệ thống vẫn duy trì hoạt động trong một thời gian dài khi mất nguồn điện lưới cung cấp chính.

- Thông tin vận hành hệ thống được hiển thị đầy đủ trên tủ điều khiển qua màn hình LCD.

- Người dùng có thể thiết lập cấu hình hệ thống một cách thuận lợi thông qua các nút điều khiển trên mặt tủ và màn hình LCD tích hợp sẵn.

Phần mềm được thiết kế bao gồm 2 phân hệ. Phân hệ thứ nhất là hệ thống backend có chức năng tiếp nhận và lưu trữ thông tin trạng thái thiết bị báo cháy lên cloud; cung cấp dữ liệu cho frontend. Phân hệ thứ hai là ứng dụng chạy trên thiết bị di động (frontend) có chức năng hiển thị thông tin trạng thái vận hành của Hệ thống báo cháy tự động; các thông tin giám sát điện năng, cảnh báo quá tải; tình trạng cảm biến và các thông tin khác của hệ thống.

2. Đề xuất, kiến nghị

- Đề xuất áp dụng kết quả nghiên cứu vào sản xuất và đời sống:

Có thể ứng dụng kết quả nghiên cứu của đề tài trong việc sản xuất, kinh doanh, phát triển các sản phẩm về điều khiển tự động, thiết bị thông minh… do điều kiện thi công rất khả thi, giá thành thấp nhưng mang tính tiện ích rất cao.

Hiện nay, nhu cầu của thị trường về thiết bị báo cháy tự động và điều khiển thiết bị từ xa qua mạng điện thoại di động là rất lớn nhằm đem lại các tiện ích và tiết kiệm chi phí, hạn chế các thiệt hại về tài sản, con người... trong quá trình sử dụng. Những nơi đã trang bị hệ thống báo cháy theo kiến trúc cũ, chưa có tính năng báo cháy và điều khiển qua điện thoại di động, giám sát trực tuyến qua phần mềm chạy trên điện thoại thông minh hoàn toàn có thể được nâng cấp khi áp dụng hệ thống được xây dựng ở đề tài này.

- Kiến nghị:

Hệ thống đã phát huy ưu điểm công nghệ 4.0 với khả năng kết nối, giám sát từ xa, hoạt động hiệu quả. Dữ liệu trạng thái vận hành hệ thống đã được cập nhật liên tục theo thời gian thực lên cloud. Ở phiên bản hiện hành, phần mềm mới khai thác trực tiếp thông tin trạng thái hệ thống và hiển thị. Chưa khai thác lịch sử vận hành.

Hướng phát triển trong tương lai là ứng dụng trí tuệ nhân tạo (AI: Artificial Intelligence) để phân tích dữ liệu lịch sử để đánh giá xu hướng vận hành và đưa ra dự báo các nguy cơ sớm liên quan đến các sự kiện về khai thác nguồn điện, các sự kiện kết nối thiết bị…

Kiến nghị triển khai áp dụng sản phẩm tại Trung tâm tích hợp dữ liệu tỉnh Hưng Yên; kết nối với hệ thống báo cháy hiện có để nâng cao khả năng phát hiện và cảnh báo cháy, bảo đảm an toàn tài sản. Có thể mở rộng ứng dụng ở các địa điểm khác để nâng cao năng lực phòng và chữa cháy.

CHỦ NHIỆM ĐỀ TÀI	TỔ CHỨC THỰC HIỆN TRUNG TÂM CNTT&TT GIÁM ĐỐC
ThS. Bùi Xuân Hạnh	Bùi Xuân Hạnh

DANH MỤC TÀI LIỆU THAM KHẢO

1. TS. Đinh Anh Tuấn, Khoa Điện – Điện tử, Trường Đại học Hàng Hải Việt Nam: Nghiên cứu, thiết kế hệ thống báo cháy phân tán (Research, design distributed fire alarm system).
2. Bùi Thị Linh, Trường Đại học Công nghệ: Quản lý, giám sát và điều khiển thiết bị bằng phần mềm trên Android thông qua mạng điện thoại di động.
3. TS. Lưu Kim Thành, KS. Nguyễn Kiêm Thanh, Khoa Điện – Điện tử TB, Trường Đại học Hàng Hải: Nghiên cứu chế tạo hệ thống báo cháy tự động trên tàu thủy dựa trên cơ sở công nghệ mạng AS-I (Building the fire detecting and general alarm system in the ship by using the AS-I Netword).
4. Giáo trình: Tổng quan về viễn thông, Học viện công nghệ Bưu chính viễn thông.
5. http://www.alldatasheet.com/view.jsp: ATMEGA2560
6. https://nettigo.eu/products/sim800l-gsm-grps-module
7. Chỉ thị số 11/CT-UBND ngày 10/10/2017 của UBND tỉnh Hưng Yên về việc tăng cường công tác phòng cháy chữa cháy trên địa bàn tỉnh Hưng Yên.
8. Tiêu chuẩn Việt Nam TCVN 5738:2001 về hệ thống báo cháy tự động - yêu cầu kỹ thuật.