

Mục lục

| | |
|--|----|
| Chương 1. GIỚI THIỆU ĐỀ TÀI | 1 |
| 1.1. Giới thiệu..... | 1 |
| 1.1.1. Xu hướng Internet of Things..... | 2 |
| 1.1.2. Các tính năng nhà thông minh Smarthome | 3 |
| 1.2. Tình hình nghiên cứu trong nước và thế giới | 6 |
| 1.2.1. Thế giới..... | 6 |
| 1.2.2 Trong nước..... | 10 |
| Chương 2. CƠ SỞ LÝ THUYẾT | 13 |
| 2.1. Chọn chuẩn truyền phù hợp..... | 13 |
| 2.1.1. Giới thiệu | 13 |
| 2.1.2. Đặc tính kỹ thuật | 13 |
| 2.2. Vi điều khiển | 17 |
| 2.2.1. Lý thuyết vi điều khiển | 17 |
| 2.2.2. Cấu trúc cơ bản | 18 |
| 2.2.3. Một số dòng vi điều khiển | 18 |
| 2.3. Đo lường điện áp..... | 19 |
| 2.4. Đo lường dòng điện | 20 |
| 2.4.1. Điện trở shunt..... | 20 |
| 2.4.2. Cảm biến Hall | 20 |
| 2.4.3. Biến dòng..... | 21 |
| 2.5. Đo lường công suất..... | 21 |
| Chương 3. THIẾT KẾ PHẦN CỨNG | 22 |

| | |
|--|----|
| 3.1. Sơ đồ kết nối | 22 |
| 3.2. Thiết kế phần cứng cho mạch Master | 23 |
| 3.2.1. Truyền thông giao tiếp UART | 24 |
| 3.2.2. Giao tiếp I2C | 27 |
| 3.2.3. Kết nối LCD | 27 |
| 3.2.4. Thiết kế nguồn..... | 29 |
| 3.3. Thiết kế phần cứng cho mạch Slave:..... | 30 |
| 3.3.1. Thiết kế khối nguồn cho các mạch Slave:..... | 32 |
| 3.3.2. Thiết kế khối giao tiếp RS485 cho các mạch Slave: | 32 |
| 3.3.3. Thiết kế khối vi điều khiển cho các mạch Slave: | 33 |
| 3.3.4. Thiết kế khối button cho các mạch Slave Button:..... | 34 |
| 3.3.5. Thiết kế khối Relay cho các mạch Slave Device: | 34 |
| 3.3.6. Thiết kế khối Sensor cho mạch Slave Sensor: | 35 |
| 3.3.7. Thiết kế khối Speaker cho mạch Slave Sensor:..... | 36 |
| 3.3.8. Thiết kế khối đo các thông số điện năng..... | 37 |
| 3.3.9. Sơ đồ thiết kế các mạch hoàn chỉnh..... | 41 |
| 3.4. Mạch giám sát hệ thống qua camera: | 47 |
| Chương 4. THIẾT KẾ HỆ THỐNG | 50 |
| 4.1. Các tính năng kỹ thuật..... | 50 |
| 4.1.1. Khả năng đóng ngắt tiện dụng..... | 50 |
| 4.1.2. Sơ đồ khối chức năng tổng quát | 51 |
| 4.1.3. Giao thức truyền thông..... | 52 |
| 4.2. Thiết kế phần mềm cho Master và Slave..... | 59 |

| | | |
|-----------|---|----|
| 4.2.1. | Thiết kế phần mềm cho Slave Device..... | 59 |
| 4.2.2. | Thiết kế phần mềm cho Slave Button | 59 |
| 4.2.3. | Thiết kế phần mềm cho Slave Sensor | 60 |
| 4.2.4. | Thiết kế phần mềm cho Slave Compteur..... | 60 |
| 4.2.5. | Thiết kế phần mềm cho Master | 61 |
| 4.3. | Thiết kế Webserver điều khiển hệ thống nhà thông minh Smarthome | 62 |
| 4.3.1. | Thiết kế Back-end | 62 |
| 4.3.2. | Thiết kế Front-end..... | 64 |
| 4.3.3. | Webserver sau khi thiết kế..... | 65 |
| 4.4. | Thiết kế App Android điều khiển nhà thông minh Smarthome | 67 |
| 4.4.1. | React Native..... | 67 |
| 4.4.2. | Redux..... | 68 |
| 4.4.3. | AI | 69 |
| 4.4.4. | Một số hình ảnh của App Android | 70 |
| 4.5. | Hiệu chỉnh phép đo các thông số dòng điện | 71 |
| 4.5.1. | Hiệu chỉnh điện áp..... | 71 |
| Chương 5. | KẾT QUẢ THỰC HIỆN..... | 73 |
| 5.1. | Tổng thể mô hình..... | 73 |
| 5.2. | Khối Master | 73 |
| 5.3. | Khối Slave..... | 74 |
| 5.3.1. | Khối Slave Button: | 74 |
| 5.3.2. | Khối Slave Device:..... | 75 |
| 5.3.3. | Khối Slave Sensor: | 76 |

Mục lục

| | |
|---|----|
| 5.4. Bảng đo các thông số điện năng..... | 77 |
| 5.4.1. Đo lường điện áp | 78 |
| 5.4.2. Đo lường dòng điện..... | 79 |
| 5.4.3. Đo lường công suất..... | 79 |
| Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN..... | 81 |
| 6.1. Kết quả đạt được..... | 81 |
| 6.2. Hạn chế..... | 81 |
| 6.3. Hướng phát triển | 81 |

DANH MỤC HÌNH VẼ

| | |
|---|----|
| Hình 1.1: Giới thiệu mô hình nhà thông minh smarthome | 11 |
| Hình 1.2: Panel điều khiển nhà thông minh smarthome | 13 |
| Hình 1.3: Điều khiển Smarthome qua công tắc hoặc điện thoại | 14 |
| Hình 1.4: Bật tắt đèn theo ngữ cảnh hoặc cảm biến ánh sáng | 15 |
| Hình 1.5: Giám sát thông qua camera | 16 |
| Hình 1.6: Google Home | 17 |
| Hình 1.7: Amazon Echo | 18 |
| Hình 1.8: Amazon Look | 18 |
| Hình 1.9: HomeKit và Apple TV | 19 |
| Hình 1.10: Hệ sinh thái IoT | 20 |
| Hình 2.1: Dạng tín hiệu trên 2 đường RS485 | 22 |
| Hình 2.2: Phân cực fail-safe trên đường truyền đa trạm | 23 |
| Hình 2.3: Dạng sóng ngõ ra trên dây A | 23 |
| Hình 2.4: Cáp xoắn đôi 24AWG có bọc chống nhiễu | 24 |
| Hình 2.5: Nguyên lý mạch chia áp | 27 |
| Hình 2.6: Điện trở shunt | 28 |
| Hình 2.7: Nguyên lý cảm biến Hall | 28 |
| Hình 2.8: Nguyên lý hoạt động của biến dòng | 29 |
| Hình 3.1: Sơ đồ kết nối hệ thống | 30 |
| Hình 3.2: Sơ đồ kết nối mạch Master | 31 |
| Hình 3.3: Sơ đồ mạch RS485 | 32 |
| Hình 3.4: Module sim800A | 32 |
| Hình 3.5: ESP8266 | 33 |
| Hình 3.6: Sơ đồ khối Wifi | 33 |
| Hình 3.7: Sơ đồ khối thời gian thực | 34 |
| Hình 3.8: LCD 20x4 | 34 |
| Hình 3.9: Sơ đồ khối nguồn 12V/5V | 36 |

Mục lục

| | |
|---|----|
| Hình 3.10: Mạch chuyển 5V về 3.3V | 37 |
| Hình 3.11: Vi điều khiển PIC16F628A | 37 |
| Hình 3.12: Vi điều khiển PIC16F887 | 38 |
| Hình 3.13: Sơ đồ mạch nguồn | 39 |
| Hình 3.14: Sơ đồ mạch RS485 | 39 |
| Hình 3.15: Sơ đồ mạch vi điều khiển..... | 40 |
| Hình 3.16: Sơ đồ mạch đo MCU | 40 |
| Hình 3.17: Sơ đồ mạch Button | 41 |
| Hình 3.18: Sơ đồ mạch Relay | 41 |
| Hình 3.19: Cảm biến nhiệt độ DHT11 | 42 |
| Hình 3.20: Cảm biến gas MQ135 | 42 |
| Hình 3.21: Sơ đồ mạch đo Sensor | 43 |
| Hình 3.22: Sơ đồ mạch đo MCU | 43 |
| Hình 3.23: EMIC ADE7753 | 44 |
| Hình 3.24: Khối chuyển đổi điện áp..... | 47 |
| Hình 3.25: Khối chuyển dòng điện | 47 |
| Hình 3.26: Sơ đồ nguyên lý mạch Master | 48 |
| Hình 3.27: Sơ đồ nguyên lý mạch Master | 49 |
| Hình 3.28: Sơ đồ nguyên lý mạch Slave Button..... | 50 |
| Hình 3.29: Sơ đồ nguyên lý mạch Slave Device | 51 |
| Hình 3.30: Sơ đồ nguyên lý mạch Slave Sensor..... | 52 |
| Hình 3.31: Sơ đồ nguyên lý mạch Slave Compteur | 53 |
| Hình 3.32: Sơ đồ nguyên lý mạch Slave Compteur | 54 |
| Hình 3.33: Máy tính nhúng Raspberry Pi 3 | 55 |
| Hình 3.35: Kết nối Rasp Pi 3 và Camera | 56 |
| Hình 4.1: Tổng quan chức năng mô hình smarthome | 57 |
| Hình 4.2: Sơ đồ khối chức năng smathome..... | 58 |
| Hình 4.3: Ví dụ frame truyền trong hệ thống | 65 |
| Hình 4.4: Sơ đồ khối Slave Device | 66 |

Mục lục

| | |
|--|----|
| Hình 4.5: Sơ đồ khối Slave Button..... | 66 |
| Hình 4.6: Sơ đồ khối Slave Sensor | 67 |
| Hình 4.7: Sơ đồ khối Slave Compteur | 68 |
| Hình 4.8: Sơ đồ khối Master | 69 |
| Hình 4.9: Nodejs | 70 |
| Hình 4.10: Trang Login vào hệ thống..... | 71 |
| Hình 4.11: Webserver điều khiển nhà thông minh SmartHome | 72 |
| Hình 4.12: Trang điều khiển thiết bị | 72 |
| Hình 4.13: Trang ngữ cảnh của hệ thống | 72 |
| Hình 4.14: Trang hiển thị chart hệ thống..... | 73 |
| Hình 4.15: React-native..... | 74 |
| Hình 4.16: Redux..... | 75 |
| Hình 4.17: Sơ đồ hoạt động của AI | 76 |
| Hình 4.18: Một số hình ảnh của App Android | 77 |
| Hình 4.19: Bố trí thí nghiệm để đo đặc..... | 79 |
| Hình 5.1: Sơ đồ tổng thể mô hình | 80 |
| Hình 5.2: Layout khối master | 80 |
| Hình 5.3: Mặt sau khối master | 81 |
| Hình 5.4: Layout khối Slave Button | 81 |
| Hình 5.5: Mạch khối Slave Button | 82 |
| Hình 5.6: Layout khối Slave Device..... | 82 |
| Hình 5.7: Mạch khối Slave Device..... | 83 |
| Hình 5.8: Layout khối Slave Sensor | 83 |
| Hình 5.9: Mạch khối Slave Sensor | 84 |
| Hình 5.10: Thí nghiệm đo đặc các thông số | 84 |

DANH MỤC BẢNG

| | |
|---|----|
| Bảng 2.1 Các đặc điểm kỹ thuật của các module kết nối không dây | 25 |
| Bảng 3.1 Một vài lệnh AT cơ bản..... | 32 |
| Bảng 3.2 Chức năng của các chân LCD | 35 |
| Bảng 3.3 Thông số kỹ thuật của vi điều khiển PIC16F628A..... | 38 |
| Bảng 3.4 Thông số kỹ thuật của vi điều khiển PIC16F887 | 38 |
| Bảng 3.5 Sơ đồ và chức năng các chân của EMIC ADE7753 | 45 |
| Bảng 5.1 Thông số kỹ thuật thiết bị..... | 85 |
| Bảng 5.2 Kết quả đo lường điện áp..... | 85 |
| Bảng 5.3 Kết quả đo lường dòng điện..... | 86 |
| Bảng 5.4 Kết quả đo lường công suất khi $\cos\theta \approx 1$ | 86 |
| Bảng 5.5 Kết quả đo lường công suất khi $\cos\theta \neq 1$ | 87 |

DANH MỤC TỪ VIẾT TẮT

ARM (Advanced RISC Machine) : Cấu trúc vi xử lý tân tiến được phát triển bởi RISC

CPU (Central Processing Unit): Bộ xử lý trung tâm.

CSMA/CD (Carrier Sense Multiple Access with Collision Detect): đa truy cập nhận biết sóng mang phát hiện xung đột.

Iot (Internet of Things): Mọi vật kết nối internet

AI (Artificial Intelligence): Trí tuệ nhân tạo

TÓM TẮT LUẬN VĂN

Ngày nay, với sự phát triển rất nhanh về công nghệ cùng với cuộc cách mạng công nghiệp 4.0, công nghệ thông minh đang ngày càng làm chủ thị trường. Nhà thông minh dường như không còn quá xa lạ với mọi người. Kéo theo đó là ngày càng nhiều những tiện ích mà nhà thông minh mang lại cho chủ sở hữu. Nhằm bắt được điều này, tôi đã theo đuổi đề tài về nhà thông minh trong luận văn. Luận văn về thiết kế và thi công mô hình nhà thông minh với một số tính năng đã hiện hữu trên các sản phẩm nhà thông minh của các công ty trên thị trường. Mặt khác, sự gia tăng về chi phí điện cũng như mối quan tâm về ấm lên toàn cầu đã đặt ra nhu cầu phát triển các sản phẩm, ứng dụng với mục tiêu cắt giảm điện năng tiêu thụ. Vì lý do này luận văn cũng tập trung vào việc giám sát, kiểm tra phân tích công suất của thiết bị trong nhà, giúp người dùng sử dụng chúng một cách hiệu quả hơn. Cùng với đó, dữ liệu về năng lượng tiêu thụ được gửi lên internet cũng trở rất có giá trị trong việc phân tích nhu cầu, thói quen của người dùng, giúp các công ty, tổ chức xác định được nhu cầu năng lượng hay thậm chí là tình hình phát triển kinh tế xã hội của một địa bàn.

Chương 1: Giới thiệu đề tài: Khái quát định nghĩa, chức năng của nhà thông minh cũng như tình hình nghiên cứu trong và ngoài nước, đề xuất thiết kế mô hình hệ thống smarthome.

Chương 2: Cơ sở lý thuyết về chuẩn truyền, vi điều khiển cũng như những phương pháp đo lường điện áp, dòng điện, công suất.

Chương 3: Thiết kế phần cứng: Thiết kế và thi công phần cứng cho hệ thống nhà thông minh smarthome.

Chương 4: Thiết kế hệ thống: Kết quả thực hiện và hình ảnh thực tế của mô hình.

Chương 5: Kết quả thực hiện, hình ảnh thực tế của mô hình cũng như bảng số liệu đo đạc các thông số điện năng.

Chương 6: Kết luận và hướng phát triển của đề tài.

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu

Nhà thông minh hay Smarthome được hiểu là một ngôi nhà/căn hộ được trang bị hệ thống tự động tiên tiến gồm các thiết bị điện tử gia dụng được kết nối với nhau thành mạng thiết bị và hoạt động theo kịch bản tùy biến, nhằm tạo ra cuộc sống tiện nghi, an toàn, tiết kiệm năng lượng. Thời đại công nghệ đã mang cho chúng ta rất nhiều tiện ích thậm chí là viễn tưởng cách đây vài thập niên trước. Các kiểu mẫu nhà thông minh hiện đại được trang bị đầy đủ những công nghệ tiên tiến. Các thiết bị gia dụng như tủ lạnh, tivi, máy tính hay camera an ninh... có khả năng tự động hóa và giao tiếp với nhau theo một lịch trình/ kịch bản định sẵn. Đặc biệt chúng có thể được điều khiển ở bất cứ đâu, từ trong chính ngôi nhà thông minh đó đến bất kỳ nơi nào trên thế giới thông qua các thiết bị di động dựa trên mạng internet.

Ngoài ra, hệ thống các cảm biến trong ngôi nhà thông minh sẽ liên tục cập nhật các thông số về độ ẩm, nhiệt độ... của từng khu vực trong ngôi nhà. Máy chủ sẽ



Hình 1.1: Giới thiệu mô hình nhà thông minh smarthome

phân tích các thông số này và ra lệnh điều khiển các thiết bị như điều hòa, máy hút ẩm nhằm tạo ra và duy trì trạng thái môi trường tốt nhất.

Hiện nay, các hệ thống Smarthome ở Việt Nam cũng như trên thế giới hầu như tập trung vào điều khiển mà ít tập trung vào phân tích năng lượng tiêu thụ. Với xu hướng phân tích hoạt động của từng thiết bị độc lập vì những lợi ích của nó, khả năng tích hợp thiết bị giám sát hệ thống điện vào các hệ thống nhà thông minh hiện nay là rất khả thi.

1.1.1. Xu hướng Internet of Things

Với sự phát triển nhanh chóng của Internet, điện thoại thông minh, đặc biệt là các thiết bị cảm biến và các giải pháp kết nối thế giới thực vào mạng không gian ảo. Mạng lưới vạn vật kết nối Internet (Internet of things - IoT) đang trở thành xu hướng mới của thế giới. Các cảm biến nhỏ hơn, rẻ hơn và thông minh hơn đang được lắp đặt trong nhà, quần áo, phụ kiện, các thành phố, mạng lưới giao thông và năng lượng cũng như các quy trình sản xuất.

IoT là một viễn cảnh tương lai của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, qua mạng Internet mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT là cấu thành công nghệ không dây, công nghệ vi mạch và Internet. Nói đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.

Việc kết nối giữa các thiết bị có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại. Các thiết bị có thể là điện thoại thông minh, công tơ điện thông minh, máy pha cafe, máy giặt, tai nghe, bóng đèn, và nhiều thiết bị khác. Cisco, nhà cung cấp giải pháp và thiết bị mạng hàng đầu hiện nay dự báo: Đến năm 2020, sẽ có khoảng 50 tỷ đồ vật kết nối vào Internet, thậm chí con số này còn gia tăng nhiều hơn nữa. IoT sẽ là mạng khổng lồ kết nối tất cả mọi thứ, bao gồm cả con người và sẽ tồn tại các mối quan hệ

giữa người và người, người và thiết bị, thiết bị và thiết bị. Một mạng lưới IoT có thể chứa đến 50 đến 100 nghìn tỉ đối tượng được kết nối và mạng lưới này có thể theo dõi sự di chuyển của từng đối tượng. Một con người sống trong thành thị có thể bị bao bọc xung quanh bởi 1000 đến 5000 đối tượng có khả năng theo dõi.

Hòa theo xu hướng IoT, đồng thời nhận thấy những rào cản trong việc kiểm soát và quản lý năng lượng hiện tại. Rất nhiều giải pháp và ứng dụng liên quan đến nhà thông minh, quản lý năng lượng và điều khiển thiết bị đang được đầu tư, phát triển theo hướng này trên toàn thế giới cũng như tại Việt Nam. Sự ra đời của các giải pháp này đã giúp cho việc kiểm soát và quản lý năng lượng đạt được hiệu quả và tối ưu hơn.

1.1.2. Các tính năng nhà thông minh Smarthome

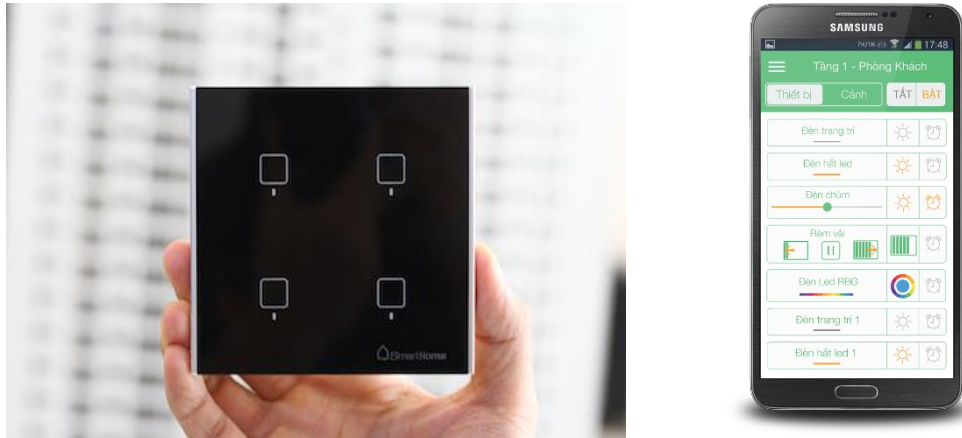
1.1.2.1. Điều khiển thiết bị

Tính năng điều khiển thông minh chính được thực hiện qua các panel điều khiển với công nghệ tiên tiến, được đặt ở các vị trí phù hợp với nội thất và thuận tiện cho việc sử dụng. Từ các panel này có thể thiết lập và tùy chỉnh toàn bộ các hệ thống chiếu sáng, rèm cửa, an ninh, điều hòa.



Hình 1.2: Panel điều khiển nhà thông minh smarthome

Bên cạnh đó việc điều khiển các thiết bị cũng được thực hiện qua thiết bị điều khiển như công tắc trong nhà. Ngoài ra, chúng ta hoàn toàn có thể giám sát và điều khiển thiết bị trong nhà thông qua Smart phone hoặc internet.



Hình 1.3: Điều khiển Smarthome qua công tắc hoặc điện thoại

1.1.2.2 Chiếu sáng trong nhà

Các mẫu nhà thông minh hiện nay đều sử dụng các thiết bị tự động bật tắt đèn khi nhận thấy sự hiện diện của con người, các tính năng này được mở rộng theo các tính năng đem lại tiện ích cho chủ nhà.

- Tính năng theo lịch trình: Các đèn ở các vị trí nhất định được cài đặt để bật hay tắt theo một khung giờ nhất định được quy định trước bởi chủ nhà. Ví dụ, các đèn ngoài sân và xung quanh nhà sẽ bật sáng khi trời sầm tối (khoảng 6h chiều), tắt khi cả nhà đi ngủ (10h), ...
- Tính năng cảm biến chuyển động: Cảm biến chuyển động sẽ nhận diện khi có người. Khi có người bước vào, các nhóm đèn sẽ tự động tăng dần độ sáng đến 30%, tạo ánh sáng dịu mắt và đảm bảo đủ sáng cho việc đi lại trong nhà.
- Tính năng điều khiển theo hoạt cảnh: Với tính năng này chủ nhà có thể thiết lập các chế độ định sẵn như một kịch bản ứng với mỗi hoàn

cảnh cụ thể, giúp tiết kiệm thời gian chỉnh định, mang lại hiệu quả sử dụng cao nhất và tiết kiệm nhất cho ngôi nhà.



Hình 1.4: Bật tắt đèn theo ngữ cảnh hoặc cảm biến ánh sáng

1.1.2.3 An ninh

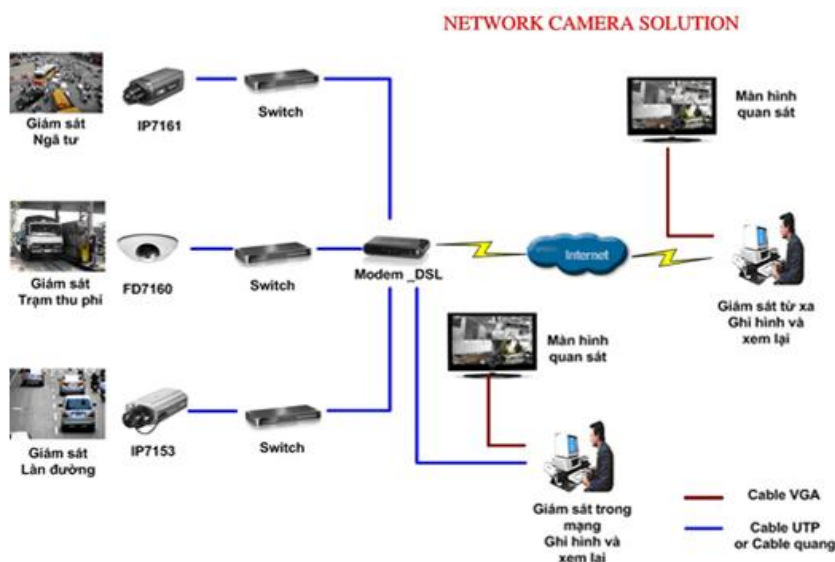
Việc sử dụng camera hiện đại tích hợp trong hệ thống liên kết mạng giúp chúng ta có thể quan sát mọi hoạt động trong và ngoài ngôi nhà cũng như có thể cảnh báo cho chúng ta biết khi có kẻ gian qua điện thoại di động và phát tín hiệu để cảnh báo, báo động. Ngoài ra, sử dụng các khóa điện tử và cảm biến gắn trong ngôi nhà là lớp bảo vệ vững chắc cho ngôi nhà để ngăn người lạ đột nhập.

Hệ thống an ninh gồm tổ hợp các cảm biến như cảm biến báo nhiệt, báo khói, báo trộm, cảm biến chuyển động, các thiết bị báo hiệu như loa, còi hú, đèn chớp...

Hình minh họa cho hệ thống giám sát hiện đại, có khả năng ghi lại mọi hoạt động xảy ra đồng thời có thể quan sát từ xa qua Laptop. Các thiết bị quan sát còn giúp người chủ có thể tiếp khách qua camera ngoài cửa khi không có mặt tại nhà.

Cảnh báo khí gas, báo cháy: Một bộ điều khiển trung tâm có thể quản lý nhiều thiết bị cảm biến khói và nhiệt sẽ kích hoạt tín hiệu báo bằng còi kêu tại chỗ và xử lý theo một số tính năng được thiết lập sẵn như bật sáng toàn

bộ đèn trong, ngoài nhà, gửi cảnh báo đến người chủ qua điện thoại... nếu không có phản ứng hệ thống sẽ tự động ngắt điện toàn nhà tránh việc chập cháy lan truyền, phun nước dập lửa.



Hình 1.5: Giám sát thông qua camera

1.1.2.4 Tiết kiệm điện và các nguồn năng lượng một cách tối đa

Smarthome sẽ cài đặt nhiều cảm biến quanh nhà để tự động tắt mở các thiết bị ở thời điểm thích hợp nhằm tiết kiệm điện và năng lượng một cách tối đa. Các cảm biến sẽ tự nhận biết nếu không có người trong nhà và ngay lập tức đưa các máy móc như đèn, tivi về trạng thái ngắt điện.

1.2. Tình hình nghiên cứu trong nước và thế giới

1.2.1. Thế giới

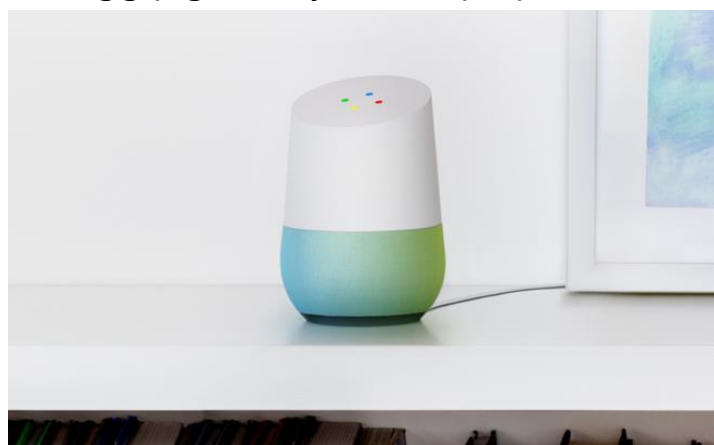
Các ông lớn công nghệ đang chạy đua để chiếm lĩnh thị trường Smart Home đầy tiềm năng. Với cuộc cách mạng công nghiệp lần thứ 4 và xu thế công nghệ Internet of Things (IoT) đang bùng nổ thì các thiết bị trong nhà ngày càng trở nên thông minh.

Thị trường thiết bị thông minh trong gia đình (Smart Home) là thị trường đầy hứa hẹn. Tuy vậy nhưng đây là thị trường chưa định hình sẵn chơi từ công nghệ, đến sản phẩm, và thói quen của người tiêu dùng. Hiện nay chưa có một chuẩn chung cho việc kết nối các thiết bị thông minh trong gia đình (Smart Home networking); Wi-Fi, Bluetooth, Z-Wave và Zigbee là các chuẩn kết nối thông dụng mà các thiết bị thông minh hiện tại đang sử dụng; mỗi công nghệ có ưu và nhược điểm riêng.

Tất cả các hãng công nghệ lớn như Google, Amazon, Apple và Samsung đều đang tìm cách chiếm lĩnh thị trường Smart Home này. Đối với những hãng công nghệ lớn thì việc chiếm lĩnh cũng như làm chủ hệ sinh thái của một xu hướng mới là vô cùng quan trọng.

1.2.1.1. Google

Google Home có thiết kế rất cơ bản và đơn giản. Google Home giống như một cô giúp việc thông thái. Chúng ta hỏi cô ta về các thông tin, nhờ nhắc lịch hẹn, nhờ gọi dậy buổi sáng, nhờ bật/tắt đèn, nhờ mở nhạc... Với tính năng trợ lý, Home có thể tra cứu thông tin trên internet và trả lời nhanh và chính xác cho chúng ta. Ngoài ra nó có thể kết nối với các sản phẩm của bên thứ 3. Ví dụ nếu chúng ta muốn dùng bóng đèn thông minh của Phillips thì có thể kết nối với nó qua Home. Khi đó chúng ta sẽ ra lệnh cho Home bật/tắt đèn bằng giọng nói. Tuy nhiên hiện tại vẫn còn khá ít các dịch vụ có



Hình 1.6: Google Home

thể sử dụng với Home. Ngoài ra nó Google Home còn đóng vai trò là một chiếc loa nghe nhạc để giải trí như những loại loa Bluetooth khác.

1.2.1.2 Amazon



Hình 1.7: Amazon Echo

Amazon khiến mọi người ngạc nhiên khi đột ngột trình làng một sản phẩm loa thông minh điều khiển bằng giọng nói Echo. Thiết bị phần cứng kết nối Internet này hỗ trợ cho hệ thống trợ lý ảo Alexa của Amazon, hoạt động tương tự Siri của Apple hay Cortana của Microsoft. Người dùng sẽ giao tiếp với chiếc loa qua giọng nói. Ngoài việc chơi nhạc, Amazon Echo có thể trả lời người dùng các câu hỏi về thời tiết, đọc truyện, mua hàng (tích hợp với hệ thống thương mại của Amazon).



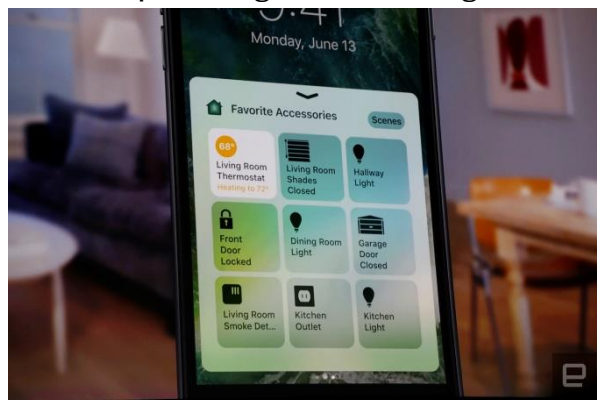
Hình 1.8: Amazon Look

Gần đây amazon vừa giới thiệu Echo Look với camera có thể nhận diện khoảng cách đến chủ thể, khả năng nhận lệnh giọng nói của chủ nhân cũng như dùng trí tuệ nhân tạo để làm mờ hậu cảnh phía sau. Ngoài ra Echo Look còn đưa ra lời khuyên bộ nào hợp với chủ nhân hơn. Bên cạnh đó nó có thể chụp ảnh cũng như quay video lại và lưu vào iphone.

1.2.1.3 Apple

Apple với nền tảng di động iOS, Apple đưa ra chuẩn HomeKit để kết nối các thiết bị thông minh trong nhà lại với hệ điều hành iOS và Iphone. Người dùng có thể dùng trợ lý ảo Siri để điều khiển các thiết bị trong nhà. Apple TV sẽ hoạt động như một bộ điều khiển trung tâm để kết nối với các thiết bị khác trong gia đình.

Chiến lược chung của các hãng lớn là không những tạo ra doanh thu lớn

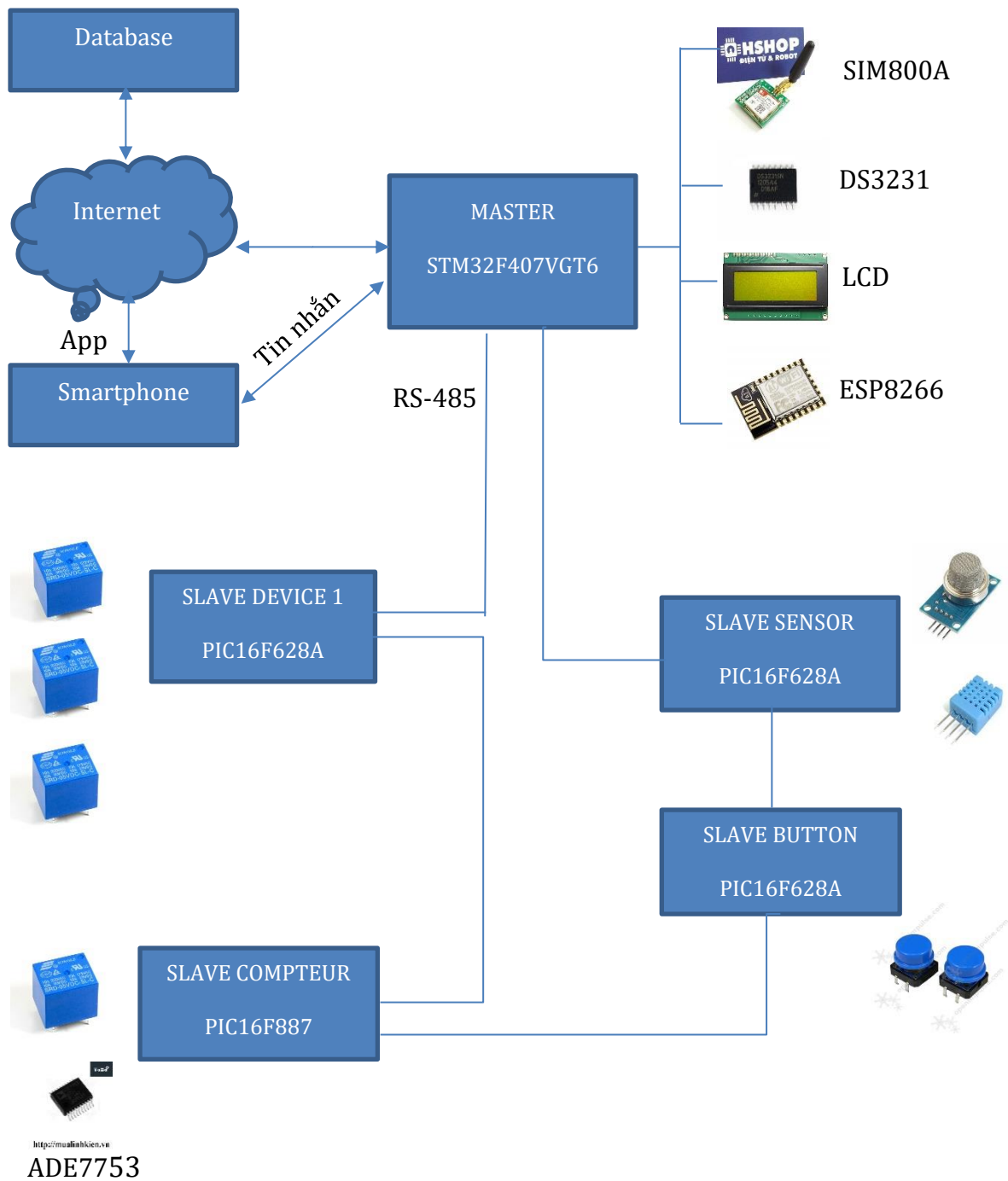


Hình 1.9: HomeKit và Apple TV

trước mắt mà về lâu dài cần phải kiểm soát được hệ sinh thái của thị trường nhà thông minh. Với nền tảng công nghệ hiện nay thì 2 yếu tố quan trọng cho hệ sinh thái nhà thông minh là smartphone và smart home hub. Apple và Google có lợi thế nhất định vì sở hữu 2 hệ điều hành di động lớn là Android và iOS. Họ có thể áp dụng những hệ điều hành này giúp họ triển khai những công nghệ mới.



1.3. Mô hình đề xuất



Hình 1.11: Mô hình đề xuất

Bảng 1.1 Bảng thông số lý thuyết thiết kế hệ thống

| Chức năng | Thông số | Ghi chú |
|---|---------------|--|
| Số thiết bị tối đa trong hệ thống | 99 thiết bị | Có thể mở rộng thêm bằng cách mở rộng frame truyền |
| Chiều dài dây từ Master đến Slave cuối cùng | Lên tới 1200m | |
| Chuẩn truyền | RS-485 | |
| Kết nối wifi với hệ thống | Wifi 2.4GHz | |

Giải thích:

Master kết nối với các Slave qua mạng truyền thông RS-485. Bên cạnh đó trên Master có gắn các module DS3231 dùng đồng bộ thời gian thực trong hệ thống, module LCD để hiển thị cho người xem các thông số của hệ thống, module ESP8266 kết nối hệ thống với internet bằng wifi giúp người dùng điều khiển thiết bị qua Webserver, module Sim800A giúp người dùng điều khiển thiết bị qua điện thoại khi ở những vùng không có internet. Các Slave kết nối vào mạng RS485 gồm Slave Device 1 để bật tắt Relay on/off thiết bị thông thường, Slave Device 2 ngoài chức năng bật tắt Relay còn có chức năng thu thập các thông số điện năng của thiết bị và gửi lên Database. Slave Button có chức năng điều khiển các Slave Device từ xa. Slave Sensor ngoài chức năng thu thập dữ liệu về nhiệt độ, độ ẩm của hệ thống còn có chức năng cảnh báo người dùng khi phát hiện khí gas dễ gây cháy nổ.

Webserver kết nối với Master thông qua wifi giúp điều khiển và giám sát hệ thống qua mạng internet. App Smarthome trên điện thoại thông minh giúp kết nối với webserver đồng thời còn có thêm AI giúp người dùng điều khiển các thiết bị qua giọng nói.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Chọn chuẩn truyền phù hợp

2.1.1. Giới thiệu

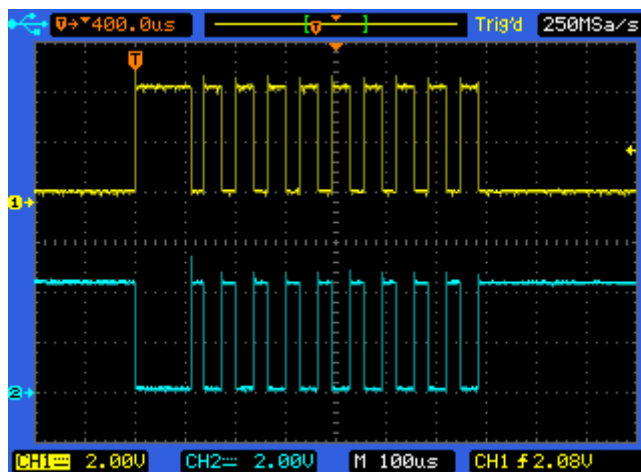
Hiện nay có rất nhiều chuẩn truyền thông ra đời để đáp ứng cho những mục đích khác nhau. Những chuẩn truyền thông thường thấy bao gồm không dây và có dây I2C, ISP, RS232, RS485, CAN, Bluetooth, Wifi, Zigbee...

Năm 1983, Hiệp hội công nghiệp điện tử (EIA) đã phê duyệt một tiêu chuẩn truyền cân bằng mới gọi là RS-485 đã được chấp nhận rộng rãi trong công nghiệp, y tế, dân dụng. Chuẩn RS-485 là một phát triển của RS-232 trong việc truyền dữ liệu nối tiếp. Liên kết RS485 được hình thành cho việc thu nhận dữ liệu ở khoảng cách xa và điều khiển cho những ứng dụng. Những đặc điểm nổi trội của RS485 là tốc độ baud có thể lên tới 115200 cho một khoảng cách là 4000 feet (1200m). Nguyên nhân mà RS485 có thể tăng tốc độ và khoảng cách truyền thông là do RS485 sử dụng phương pháp truyền 2 dây vi sai cân bằng (vì 2 dây có đặc tính giống nhau, tín hiệu truyền đi là hiệu số điện áp giữa 2 dây do đó loại trừ được nhiễu chung). Mặt khác do chuẩn truyền thông RS232 không cho phép có hơn 2 thiết bị truyền nhận tin trên đường dây.

2.1.2. Đặc tính kỹ thuật

2.1.2.1. Thông số cơ bản

Như đã nêu trên, RS485 là chuẩn truyền sử dụng đường truyền vi sai, mức tín hiệu ở các ngõ ra được xác định dựa trên sai biệt điện áp giữa 2 dây tín hiệu, nếu $V_{AB} > 200\text{mV}$ sẽ cho ra mức logic 1, $V_{AB} < 200\text{mV}$ sẽ cho ra mức logic 0, khi độ chênh điện áp ở dây A và B nằm giữa mức này được xem là vùng bất định. Điện thế của mỗi dây tín hiệu so với mass bên phía thu phải nằm trong giới hạn -7V đến 12V. Nhờ đặc tính này cùng với việc sử dụng cáp tín hiệu loại dây xoắn giúp loại bỏ được nhiễu chung nên chuẩn RS485 có khả năng kháng nhiễu mạnh.



Hình 2.1: Dạng tín hiệu trên 2 đường truyền RS 485

Hình trên cho thấy sự đối nhau ở 2 đường truyền, khi đường này mức 1 thì đường kia mức 0 và ngược lại, điều này nhằm đảm bảo cho sự chênh lệch điện áp giữa 2 dây để xác định chính xác mức logic ở ngõ ra.

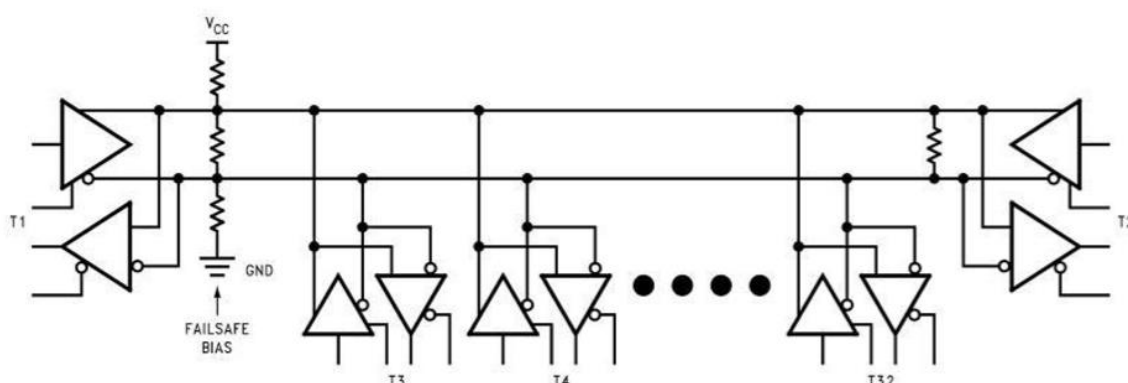
Điểm mạnh khác của RS485 là tốc độ truyền khá cao, hiện nay có thể đạt đến hơn 10Mbit/s và khoảng cách truyền có thể lên tới 1200m (4000feet). Đương nhiên 2 giới hạn này không thể đạt được cùng lúc. Theo đồ thị trên, ta thấy được mối tương quan giữa 2 đại lượng này, tốc độ 10Mbit/s chỉ có thể dùng với cự ly không quá 3m (10feet). Ngược lại, với khoảng cách 1200m tốc độ tối đa có thể lên tới là khoảng 100kbit/s.

Về cơ bản RS485 chỉ có thể truyền bán song công do sử dụng cùng lúc cả 2 đường tín hiệu.

2.1.2.2. Phân cực đường truyền

Trong mạng đa trạm ngang quyền, khi không có trạm nào phát, đường truyền phải được nằm trong một trạng thái idle xác định và các ngõ vào mỗi trạm đều ở trạng thái tổng trở cao. Đối với RS485, khi ở trạng thái idle, ngõ ra phải được đặt ở mức cao để chờ Startbit (mức thấp) báo hiệu có dữ liệu được phát. Điều này được đảm bảo bằng việc phân cực fail-safe trên đường truyền. Mục đích việc phân cực này nhằm giữ cho điện áp trên dây

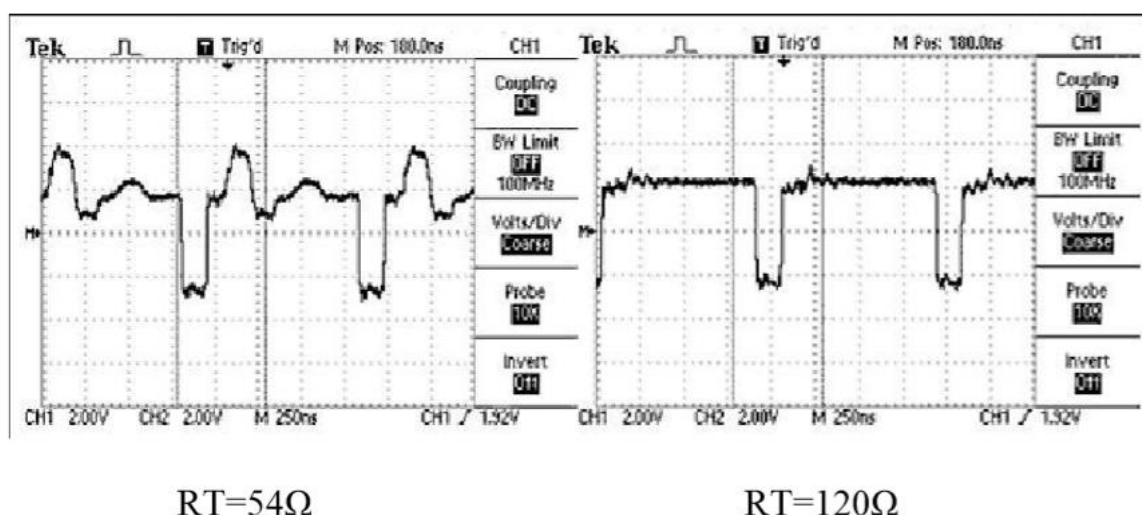
A luôn lớn hơn dây ít nhất 200mV khi không có trạm nào phát, do đó giữ được mức tín hiệu ở ngõ ra ở mức cao.



Hình 2.2: Phân cực fail-safe trên đường truyền đa trạm

2.1.2.3. Điện trở đầu cuối

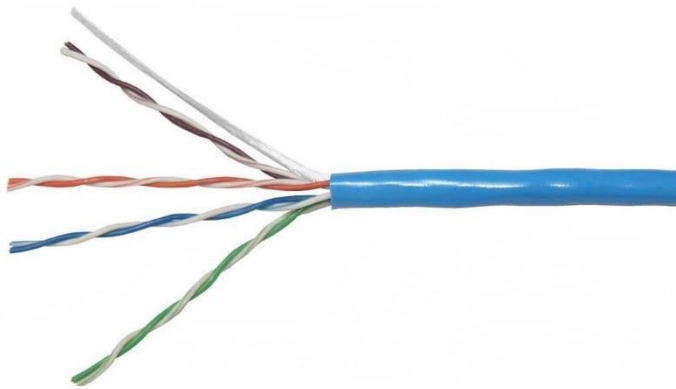
Điện trở đầu cuối là điện trở nối giữa 2 dây tín hiệu được đặt tại đầu ngoài cùng của đường truyền phía thu, có tác dụng phối hợp với trở kháng đặc tính Z_0 của cáp tín hiệu (do nhà sản xuất cung cấp) nhằm loại bỏ sóng phản xạ trên đường truyền dài. Do chuẩn RS485 sử dụng chung cặp dây tín hiệu cho việc thu và phát nên cần đặt điện trở đầu cuối ở cả 2 phía của đường truyền. Giá trị của điện trở đầu cuối lý tưởng bằng giá trị trở kháng đặc tính cáp tín hiệu, thông thường vào khoảng 100Ω - 120Ω .



Hình 2.3: Dạng sóng ngõ ra trên dây A tương ứng với 2 giá trị điện trở đầu cuối

2.1.2.4. Dây dẫn tín hiệu

Việc lựa chọn dây dẫn cho đường truyền cũng là một vấn đề khá quan trọng do yêu cầu về phối hợp trở kháng và sử dụng trong môi trường nhiễu điện từ. Theo khuyến cáo, nên chọn loại dây xoắn 24AWG có trở kháng khoảng 120Ω . Nếu có thể nên chọn loại có bọc kim cho khả năng chống nhiễu tốt hơn tuy nhiên giá thành cao hơn đáng kể.



Hình 2.4: Cáp xoắn đôi 24AWG có bọc chống nhiễu

2.1.2.5. Các chuẩn truyền khác

Ngoài chuẩn RS485 ra, còn có các chuẩn truyền không dây như Bluetooth, Wifi, Zigbee...

Bluetooth là một tiêu chuẩn truyền thông không dây sử dụng sóng radio truyền trong dải tần số từ 2400-2480 MHz. Bluetooth được sử dụng rất rộng rãi trong các ứng dụng như điện thoại di động, tai nghe, máy in và nhiều thiết bị khác. Bluetooth có hạn chế là khoảng cách kết nối ngắn.

Một công nghệ truyền thông không dây khác là Wi-Fi. Wi-Fi hoạt động tương tự như Bluetooth ở cùng tần số 2.4 GHz. Wi-Fi là công nghệ phổ biến cho các thiết bị kết nối với internet. Phạm vi kết nối có thể lên đến 1000 m.

Trong các ứng dụng nhà thông minh, ngày nay nổi lên công nghệ truyền không dây Zigbee. Zigbee là tập hợp các giao thức giao tiếp của mạng không dây với khoảng cách tương đối, tiết kiệm năng lượng và có tốc độ truyền

dữ liệu đáp ứng được các ứng dụng điều khiển và giám sát trong nhà thông minh. Điểm mạnh nhất của Zigbee là hỗ trợ kết nối theo kiểu mạng lưới. 3 dải tần số của công nghệ Zigbee là 868 MHz, 915 MHz và 2.4 GHz. Tuy nhiên có nhiều điểm mạnh nhưng Zigbee là một công nghệ mới, giá thành còn rất đắt (khoảng 350.000 – 500.000/module).

Bảng 2.1 Bảng so sánh các đặc điểm kỹ thuật của các module kết nối không dây

| Thông số | Module | | |
|-------------------------|-----------|----------|----------|
| So sánh giữa các module | Bluetooth | Wifi | Zigbee |
| Phạm vi | 1-100m | 35-2000m | 0-75m |
| Chi phí | Cao | Thấp | Rất cao |
| Công suất | Cao | Cao | Thấp |
| Tốc độ | 25 Mb/s | 54 Mb/s | 250 kb/s |

Ở đây, ngoài chuẩn RS485, tôi dùng thêm module wifi để giao tiếp giữa các thiết bị và Web server trên máy tính.

2.2. Vi điều khiển

2.2.1. Lý thuyết vi điều khiển

Vi điều khiển (μ C hay UC) là một siêu máy tính với kích thước rất nhỏ. Vi điều khiển là một hệ thống độc lập với thiết bị ngoại vi, bộ nhớ và bộ vi xử lý sử dụng như một hệ thống nhúng. Ngày nay hầu hết vi điều khiển được lập trình để nhúng vào các sản phẩm tiêu dùng hoặc thiết bị máy móc, điện thoại, thiết bị ngoại vi, xe ô tô và chế tạo thiết bị cho các hệ thống máy tính. có rất nhiều loại vi điều khiển trên thị trường như: 4bit, 8bit, 64bit và 128bit. Vi điều khiển sử dụng trong các thiết bị được người dùng kiểm soát bằng các tập lệnh.

2.2.2. Cấu trúc cơ bản

CPU - là bộ não trung tâm của vi điều khiển. CPU là thiết bị quản lý tất cả các hoạt động của hệ thống và thực hiện tất cả các thao tác trên dữ liệu như: nạp, giải mã và thực thi lệnh. CPU kết nối tất cả các thành phần của vi điều khiển thành một hệ thống duy nhất.

Memory (bộ nhớ): trong vi điều khiển, bộ nhớ hoạt động giống như bộ vi xử lý. Bộ nhớ lưu trữ tất cả các chương trình và dữ liệu. Bộ nhớ của vi điều khiển là bộ nhớ ROM(EPROM, EEPROM) hoặc bộ nhớ RAM với dung lượng nhất định. Ngày nay còn có bộ nhớ flash lưu trữ mã nguồn chương trình.

Cổng Input/output (vào/ra) - cổng I/O sử dụng để giao tiếp hoặc điều khiển các thiết bị khác nhau như máy in, LCD, LED, ...

Serial Ports - Những cổng này cung cấp giao tiếp nối tiếp giữa vi điều khiển và thiết bị ngoại vi khác nhau.

Timers - Vi điều khiển được xây dựng với một hoặc nhiều Timer hoặc bộ định thời. Các Timer và bộ định thời kiểm soát tất cả bộ đếm và thời gian hoạt động bên trong vi điều khiển. Timer được sử dụng đếm xung bên ngoài. Các hoạt động chính được thực hiện bởi timers “tạo xung, đo tần số, điều chế, tạo dao động,...

ADC - ADC là bộ chuyển đổi tín hiệu tương tự sang tín hiệu số.

DAC (digital to analog converter) : Có chức năng ngược lại với ADC. DAC thường được sử dụng để giám sát các thiết bị tương tự.

Interrupt Control (điều khiển ngắt)- là một sự kiện khẩn cấp bên trong hoặc bên ngoài bộ vi điều khiển xảy ra, buộc vi điều khiển tạm dừng thực hiện chương trình hiện tại, phục vụ ngay lập tức nhiệm vụ mà ngắt yêu cầu – nhiệm vụ này gọi là trình phục vụ ngắt (ISR).

2.2.3. Một số dòng vi điều khiển

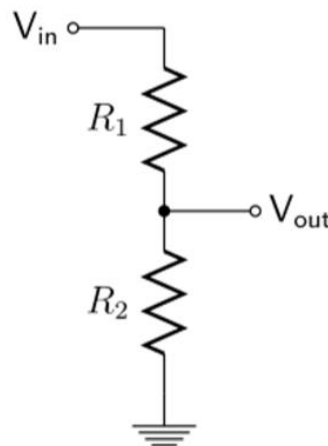
- Intel 8051
- STMicroelectronics STM8S (8-bit), ST10 (16-bit) và STM32 (32-bit)
- Atmel AVR (8-bit), AVR32 (32-bit), và AT91SAM (32-bit)

- Freescale ColdFire (32-bit) và S08 (8-bit)
- PIC (8-bit PIC16, PIC18, 16-bit dsPIC33 / PIC24)
- Renesas Electronics: RL78 16-bit MCU; RX 32-bit MCU; SuperH; V850 32-bit MCU; H8; R8C 16-bit MCU
- PSoC (Programmable System-on-Chip)
- Texas Instruments Microcontrollers MSP430 (16-bit), C2000 (32-bit), và Stellaris (32-bit)

2.3. Đo lường điện áp

Để đo điện áp, có 2 phương pháp cần được xem xét:

- Một là sử dụng mạch chia áp. Một chuỗi các điện trở được mắc nối tiếp với nguồn điện. Điện áp ra trên điện trở ra tại điểm chia áp tỉ lệ với giá trị trở ra trên tổng trở.
- Hai là đo dòng điện xuyên qua một điện trở với giá trị biết trước.



Hình 2.5: Nguyên lý mạch chia áp

Qua 2 cách tôi chọn cách một để đo áp. Ưu điểm của nó là giá thành rẻ thiết kế đơn giản. Tuy vậy không đảm bảo được cách ly cho mạch đo lường. Để đảm bảo cách ly tôi sử dụng thêm một máy biến áp phía trước mạch chia áp. Tuy nhiên máy biến áp có thể gây ra sự biến dạng và dịch pha điện áp, có thể làm giảm sự chính xác của phép đo.

2.4. Đo lường dòng điện

2.4.1. Điện trở shunt

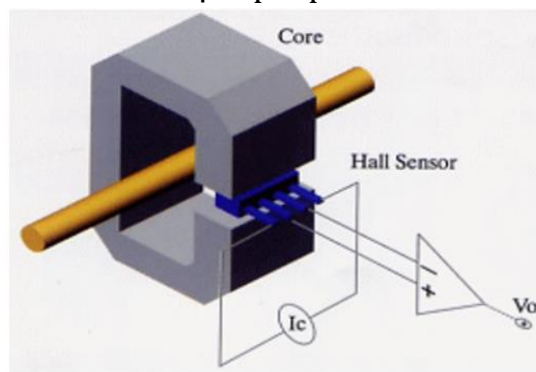
Cách đơn giản nhất để đo dòng là đo điện áp qua một điện trở Shunt. Điện trở Shunt là một điện trở có giá trị rất nhỏ (khoảng một vài milliohm) và cho dòng điện lớn đi qua. Do vậy điện áp rơi trên điện trở rất nhỏ và tổn thất điện năng rất ít. Nhược điểm của phương pháp này là mạch đo không được cách ly với mạch động lực.



Hình 2.6: Điện trở shunt

2.4.2. Cảm biến Hall

Một phương pháp khác là sử dụng cảm biến Hall. Cảm biến Hall được xây dựng từ lý thuyết Hall: khi áp dụng một từ trường vuông góc lên một bản làm bằng kim loại, chất bán dẫn hay chất dẫn điện nói chung (thanh Hall) đang có dòng điện chạy qua, ta nhận được hiệu điện thế (điện thế Hall) sinh ra tại mặt đối diện của thanh Hall. Ưu điểm là cách ly so với mạch động lực. Tuy nhiên, từ trường xung quanh dễ làm sai lệch phép đo.

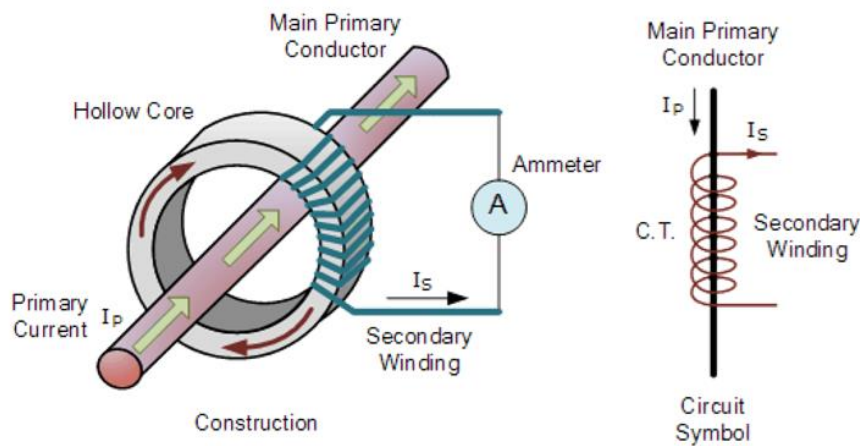


Hình 2.7: Nguyên lý cảm biến Hall

2.4.3. Biến dòng

Giống như cảm biến Hall, biến dòng sử dụng từ trường tạo ra bởi dòng điện trong dây dẫn. Dòng điện trong dây dẫn gây ra từ thông chạy trong lõi, từ đó gây ra dòng điện trong cuộn thứ cấp.

Ưu điểm của biến dòng là cách ly mạch đo với mạch động lực. Nhược điểm là có thể gây ra hiện tượng dịch pha dòng điện từ 0.1° đến 0.3° , dẫn đến sai số trong các phép đo công suất. Tuy nhiên sai số này không đáng kể.



Hình 2.8: Nguyên lý hoạt động của biến dòng

Từ các tìm hiểu trên, tôi quyết định sử dụng biến dòng vì có thể cách ly mạch đo, dễ tìm mua và thiết kế không quá phức tạp. Cảm biến Hall không đáng tin cậy vì dễ bị ảnh hưởng bởi từ tính xung quanh và không an toàn vì không thể cách ly mạch đo.

2.5. Đo lường công suất

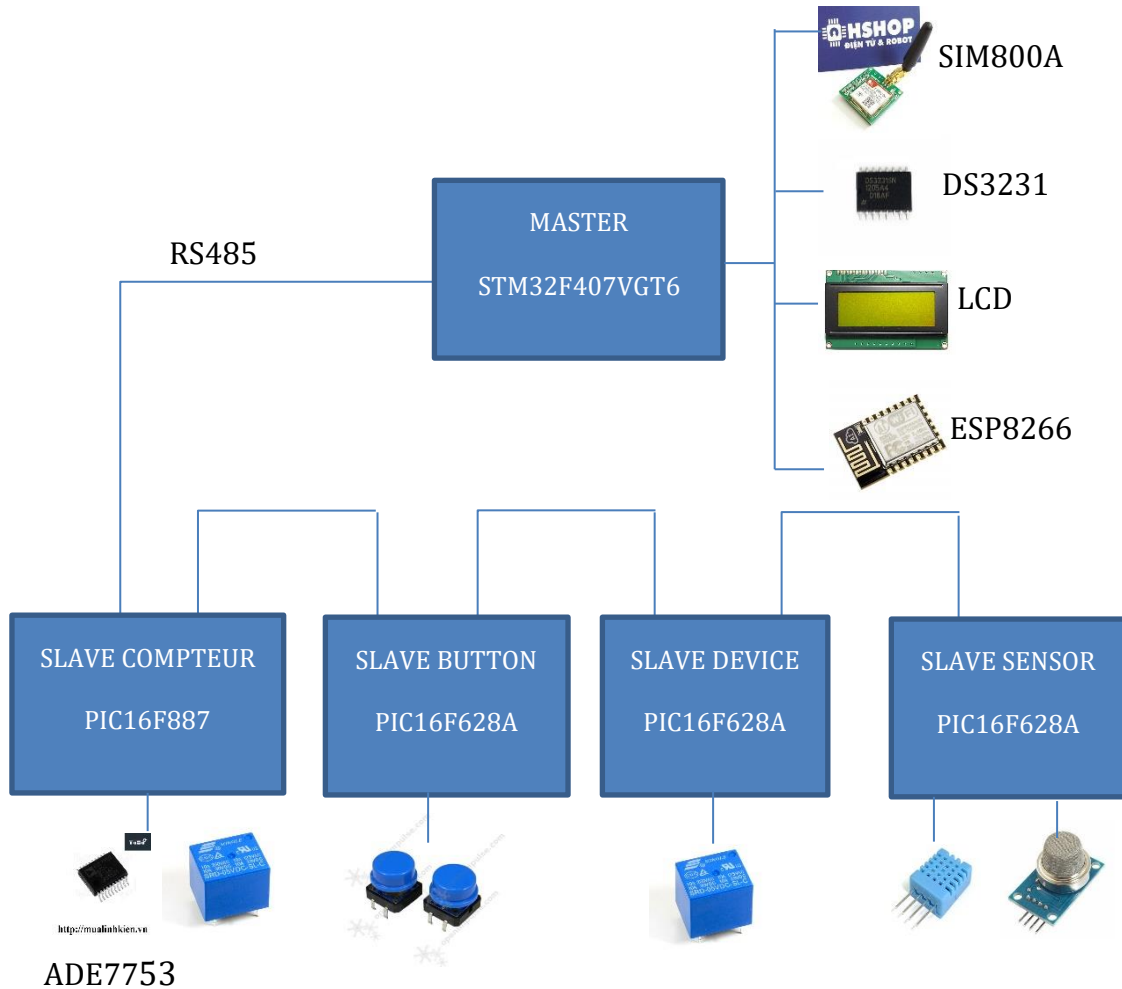
Công suất thực được đo lường bởi công được thực hiện trong một đơn vị thời gian

$$P_{rms} = V_{rms} \times I_{rms} \times \cos\theta \quad (2.1)$$

Trong đó V_{rms} và I_{rms} là điện áp và dòng hiệu dụng và θ là độ lệch pha giữa điện áp và dòng điện.

Chương 3. THIẾT KẾ PHẦN CỨNG

3.1. Sơ đồ kết nối



Hình 3.1: Sơ đồ kết nối phần cứng hệ thống

Bảng 3.1 Bảng thông số lý thuyết thiết kế hệ thống

| Chức năng | Thông số | Ghi chú |
|---|---------------|--|
| Số thiết bị tối đa trong hệ thống | 99 thiết bị | Có thể mở rộng thêm bằng cách mở rộng frame truyền |
| Chiều dài dây từ Master đến Slave cuối cùng | Lên tới 1200m | |

Chương 3: Thiết kế phần cứng

| | | |
|---------------------------|-------------|--|
| Chuẩn truyền | RS-485 | |
| Kết nối wifi với hệ thống | Wifi 2.4GHz | |

Giải thích:

Master kết nối với các Slave qua mạng truyền thông RS-485. Bên cạnh đó trên Master có gắn các module DS3231 dùng đồng bộ thời gian thực trong hệ thống, module LCD để hiển thị cho người xem các thông số của hệ thống, module ESP8266 kết nối hệ thống với internet bằng wifi giúp người dùng điều khiển thiết bị qua Webserver, module Sim800A giúp người dùng điều khiển thiết bị qua điện thoại khi ở những vùng không có internet. Các Slave kết nối vào mạng RS485 gồm Slave Device 1 để bật tắt Relay on/off thiết bị thông thường, Slave Device 2 ngoài chức năng bật tắt Relay còn có chức năng thu thập các thông số điện năng của thiết bị và gửi lên Database. Slave Button có chức năng điều khiển các Slave Device từ xa. Slave Sensor ngoài chức năng thu thập dữ liệu về nhiệt độ, độ ẩm của hệ thống còn có chức năng cảnh báo người dùng khi phát hiện khí gas dễ gây cháy nổ.

Webserver kết nối với Master thông qua wifi giúp điều khiển và giám sát hệ thống qua mạng internet. App Smarthome trên điện thoại thông minh giúp kết nối với webserver đồng thời còn có thêm AI giúp người dùng điều khiển các thiết bị qua giọng nói.

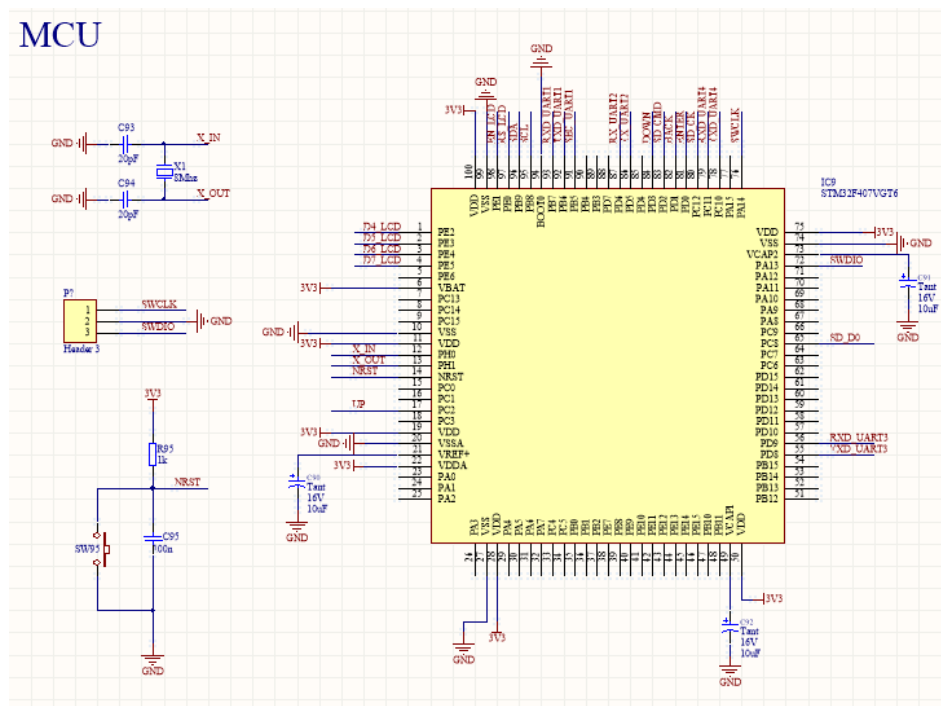
Thiết kế phần cứng cho mạch Master

Yêu cầu đối với vi điều khiển:

- Hỗ trợ giao tiếp UART để giao tiếp module wifi, module SIM800A, RS485
- Hỗ trợ giao tiếp I2C để giao tiếp với IC thời gian thực DS3231
- Đủ tốc độ và bộ nhớ cho chương trình

Dựa trên những yêu cầu, tôi chọn MCU STM32F407VGT6 vì những lý do sau:

- Lập trình bằng ngôn ngữ C có nguồn tài liệu tham khảo tốt
- Lõi ARM Cortex 32bit M4 tốc độ lên tới 168 MHz
- Hỗ trợ lên tới 140 chân I/O
- Hỗ trợ lên tới 4 USART và 2 UART
- Bộ nhớ Flash 1Mbyte
- Giá thành rẻ



Hình 3.2: Sơ đồ khối MCU mạch Master

Ở đây để giảm giá thành, thay vì gắn kit trực tiếp lên board như những luận văn trước, tôi chọn cách thiết kế mạch để hàn thẳng trực tiếp con chip lên board.

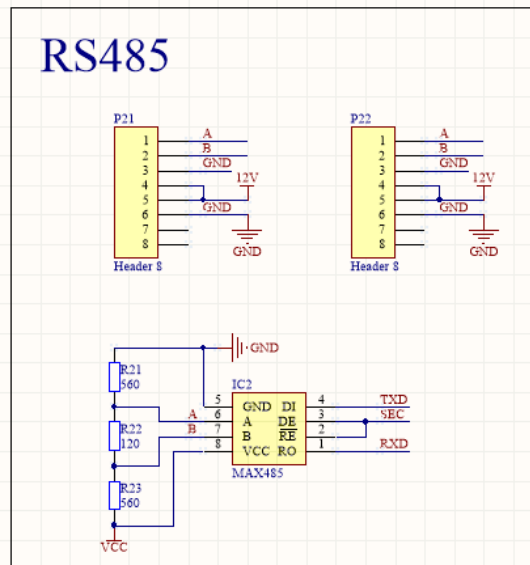
3.1.1. Truyền thông giao tiếp UART

UART (Universal Asynchronous Receiver/Transmitter) là kiểu truyền thông tin nối tiếp không đồng bộ. UART thường được dùng trong máy tính công nghiệp, truyền thông, vi điều khiển, hay một số các thiết bị truyền tin khác.

Trong ứng dụng, Master giao tiếp với RS485 qua USART1 chân PB6-PB7, với module Wifi qua USART2 chân PD5-PD6, với module SIM800A USART3 qua chân PD8-PD9.

Module RS485:

Dùng IC MAX485 đưa ngõ ra qua các jack RJ11



Hình 3.3: Sơ đồ mạch RS485

Module SIM800A:



Hình 3.4: Module sim800A

Lựa chọn module Sim 800A của hãng SimCom. Đây là một module GSM khá phổ biến tại thị trường Việt Nam, có giá thành rẻ nhưng lại chứa nhiều hạn chế như tiêu thụ công suất lớn, truyền tải còn chậm.

Module Sim800A kết với vi điều khiển bằng giao thức truyền thông nối tiếp.

Chương 3: Thiết kế phần cứng

Bảng 3.2 Một vài lệnh AT cơ bản sử dụng trong gửi nhận tin nhắn, cuộc gọi:

| Lệnh | Mô tả |
|------------------------|------------------------------------|
| AT+CNMI = 2,1,0,0,0 | Báo hiệu khi có tin nhắn tới |
| ATD0939863350 | Gọi tới một số điện thoại |
| AT + CMGS = 0939863350 | Gửi tin nhắn tới một số điện thoại |

Module Wifi:

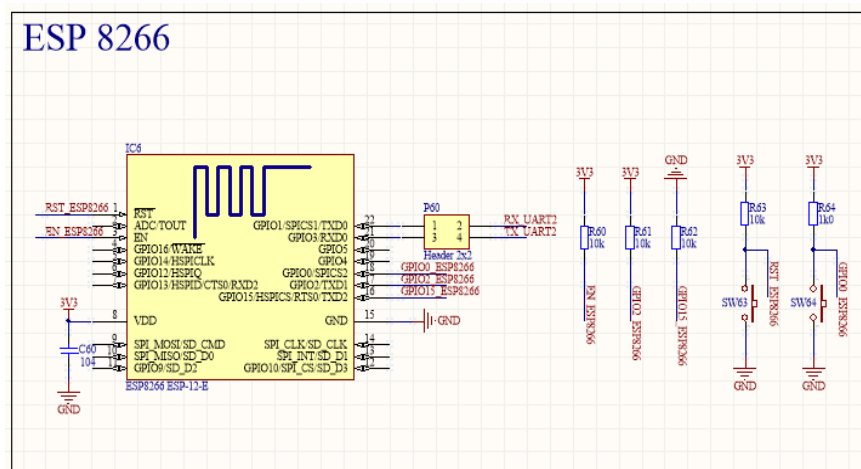
Module Wifi ESP8266 được dùng để giao tiếp giữa hệ thống với Webserver.

Một số thông số cơ bản:

- Điện áp sử dụng: 3V – 3.6V
- Dòng tiêu thụ 70mA
- Wifi 2.4GHz, hỗ trợ WPA/WPA2
- Tích hợp sẵn TCP/IP protocol



Hình 3.5: ESP8266



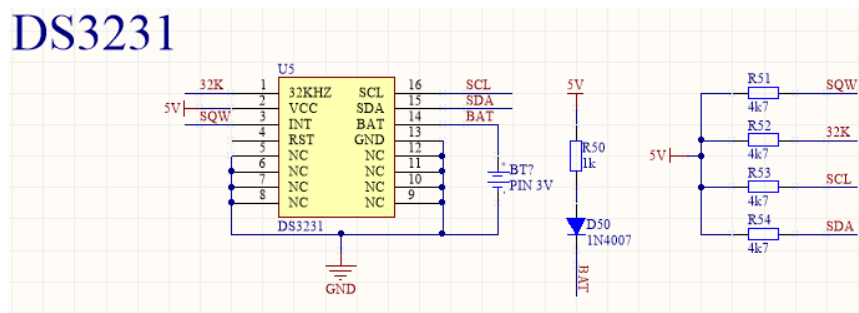
Hình 3.6: Sơ đồ khối Wifi

3.1.2. Giao tiếp I2C

Chuẩn giao tiếp nối tiếp 2 dây I2C (Inter-Integrated Circuit) được phát triển bởi Phillips năm 1980. I2C sử dụng 2 đường truyền tín hiệu:

- Một đường xung clock SCL do master phát đi
- Một đường dữ liệu SDA theo 2 hướng

Rất nhiều thiết bị có thể kết nối với bus I2C, tuy nhiên sẽ không xảy ra chuyện nhầm lẫn do mỗi thiết bị sẽ chỉ nhận ra bởi một địa chỉ duy nhất. Ở đây để giao tiếp với IC thời gian thực DS3231 tôi sử dụng chuẩn giao tiếp I2C: SCL nối vào PB8 và SDA nối vào PB9.



Hình 3.7: Sơ đồ khối thời gian thực

3.1.3. Kết nối LCD

LCD dùng để hiện các thông số ngày và giờ, nhiệt độ, độ ẩm của hệ thống. Tôi sử dụng LCD 20x4 để hiển thị và sau này có thể phát triển lên thành bảng điều khiển.



Hình 3.8: LCD 20x4

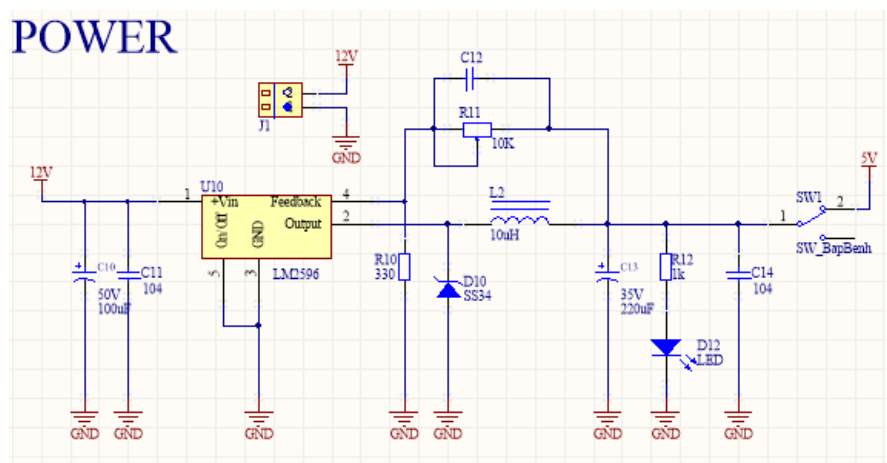
Bảng 3.3 Chức năng của các chân LCD

| Chân | Ký hiệu | Mô tả |
|------|---------|--|
| 1 | VSS | Chân nối đất cho LCD, khi thiết kế mạch nối chân này với GND của mạch điều khiển |
| 2 | VDD | Chân cấp nguồn cho LCD, khi thiết kế mạch nối chân này với VCC=5V của mạch điều khiển |
| 3 | VEE | Điều chỉnh độ tương phản của LCD |
| 4 | RS | Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD |
| 5 | R/W | 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 |
| 6 | E | Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp |
| 7-14 | DB0-DB7 | Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này: |

| | | |
|----|--|---|
| | | + Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7 |
| 15 | | Nguồn dương cho đèn nền |
| 16 | | GND cho đèn nền |

3.1.4. Thiết kế nguồn

Do khi hoạt động GSM và Wifi tiêu thụ dòng lớn nên tôi dùng mạch nguồn tổ ong 220VAC/12DC cho đi qua mạch nguồn xung 12V/5V để nuôi mạch master. Trong luận văn tôi sử dụng IC LM2596 để thiết kế mạch nguồn. Sơ đồ nguyên lý như sau:



Hình 3.9: Sơ đồ khối nguồn 12V/5V

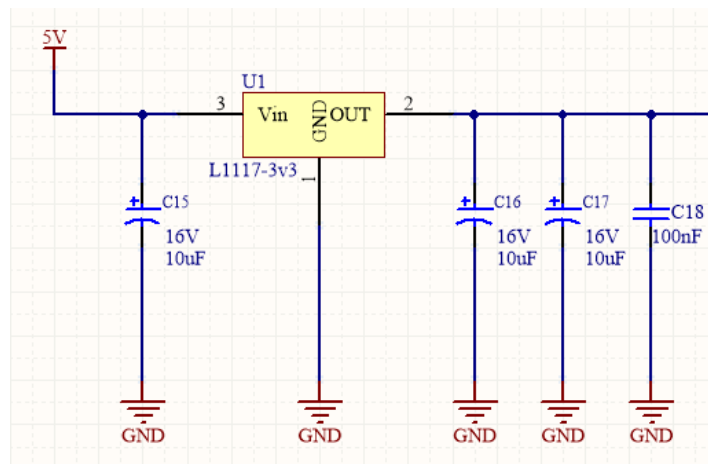
LM1117:

Để cấp nguồn 3.3V cho vi điều khiển cần phải có mạch có chức năng hạ áp từ 5V về 3.3V. IC AMS1117 là IC có khả năng làm việc đó cùng với khả năng cấp

dòng lên đến 1A. Dòng IC này được tối ưu hóa cho các thiết bị có điện áp thấp, các mạch cần nguồn có dòng ổn định và kích thước nhỏ.

Các tính năng chính:

- Tối ưu hóa cho điện áp thấp
- Nhiệt tỏa trên chip hạn chế
- Tiêu chuẩn SOT-223



Hình 3.10: Mạch chuyển 5V về 3.3V

3.2. Thiết kế phần cứng cho mạch Slave:

Yêu cầu đặt ra với các mạch Slave:

- Kích thước nhỏ gọn, giá rẻ
- Đủ khả năng xử lý
- Hỗ trợ giao tiếp UART để giao tiếp với mạch Master
- Dễ dàng lắp đặt, thay thế, sửa chữa



Hình 3.11: Vi điều khiển PIC16F628A

Chương 3: Thiết kế phần cứng

Từ những yêu cầu đặt ra, các vi điều khiển dòng PIC được lựa chọn cho Slave Button và Slave Device vì có kích thước nhỏ gọn cũng như giá cả hợp lý.

Bảng 3.4 Thông số kỹ thuật của vi điều khiển PIC16F628A

| | |
|-------------------|-----------|
| Điện áp hoạt động | 2 – 5.5V |
| Số chân | 18 |
| RAM | 224 bytes |
| EEPROM | 128 bytes |
| Ngoại vi ngoài | 1 - UART |



Hình 3.12: Vi điều khiển PIC16F887

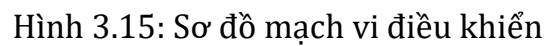
Với Slave Compteur dùng để đo các thông số điện năng, vi điều khiển PIC 16F887 được chọn vì giao tiếp SPI cũng như giá cả phải chăng

Bảng 3.5 Thông số kỹ thuật của vi điều khiển PIC16F887

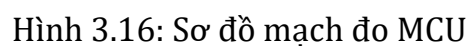
| | |
|-------------------|---------------------------|
| Điện áp hoạt động | 2 – 5.5V |
| Số chân | 40 |
| RAM | 368 bytes |
| EEPROM | 256 bytes |
| Ngoại vi ngoài | 1 – UART, 1- SPI, 1 – I2C |

1. *Journal of the American Medical Association*, 1997; 277: 1001-1005.

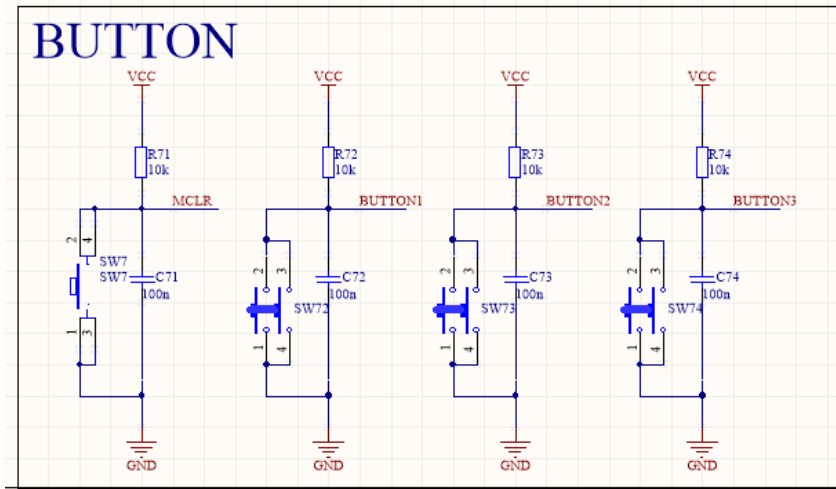
Sử dụng vi điều khiển PIC16F628A kết nối với thạch anh 20MHz



Sử dụng vi điều khiển PIC16F887 kết nối thạch anh 20MHz



3.2.4. Thiết kế khối button cho các mạch Slave Button:

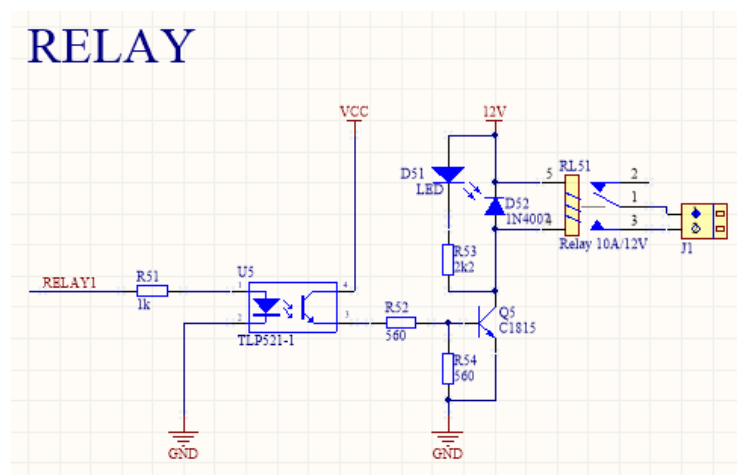


Hình 3.17: Sơ đồ mạch Button

3.2.5. Thiết kế khối Relay cho các mạch Slave Device:

Relay:

Relay là một công tắc chuyển đổi hoạt động bằng điện. Nó là một công tắc vì relay có 2 trạng thái ON và OFF. Relay ở trạng thái ON hay OFF phụ thuộc vào có dòng điện chạy qua cuộn coil trên relay hay không. Relay 12V 10A để tránh đóng cắt sai relay khi truyền nguồn qua những đoạn dây dài.

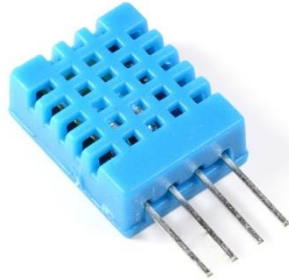


Hình 3.18: Sơ đồ mạch Relay

3.2.6. Thiết kế khối Sensor cho mạch Slave Sensor:

2 khối sensor để kết nối với cảm biến nhiệt độ DHT11 và cảm biến GAS MQ135

Cảm biến nhiệt độ:



Hình 3.19: Cảm biến nhiệt độ DHT11

Cảm biến nhiệt độ và độ ẩm DHT11 là cảm biến cơ bản và giá rẻ so với các cảm biến khác, rất thích hợp cho những ứng dụng thu thập dữ liệu cơ bản. Cảm biến DHT11 có 2 phần, 1 cảm biến độ ẩm điện dung và một điện trở nhiệt. Dữ liệu ngõ ra của cảm biến DHT là dạng số, có thể dùng bất cứ vi điều khiển nào để lấy dữ liệu ra. Dữ liệu độ ẩm mà cảm biến đo được mức từ 20% ~ 90%. Nhiệt độ đo từ 0 ~ 50 Độ C, thời gian trả dữ liệu < 50ms.

Thông số kỹ thuật:

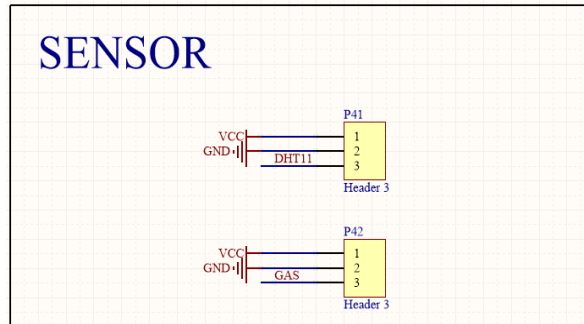
- Đo độ ẩm: 20%-95%
- Nhiệt độ: 0 – 50oC
- Sai số độ ẩm $\pm 5\%$
- Sai số nhiệt độ: $\pm 2^{\circ}\text{C}$

Cảm biến khí gas MQ135:



Hình 3.20: Cảm biến gas MQ135

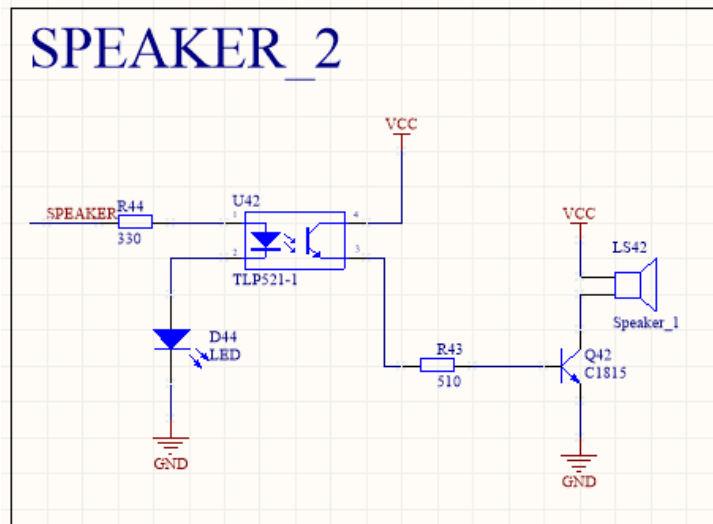
Cảm biến khí gas MQ135 là cảm biến dùng để phát hiện khí gas, rất thích hợp để trong nhà bếp gần các nguồn gas để cảnh báo người dùng khi gas bị rò rỉ.



Hình 3.21: Sơ đồ mạch đo Sensor

3.2.7. Thiết kế khối Speaker cho mạch Slave Sensor:

Khối Speaker dùng để cảnh báo cho người sử dụng biết khi nào có sự cố xảy ra.



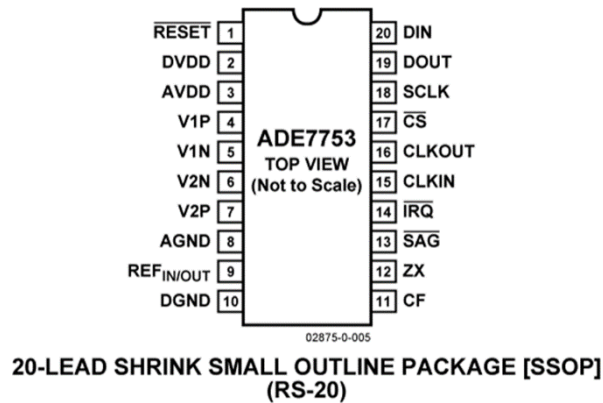
Hình 3.22: Sơ đồ mạch đo MCU

3.2.8. Thiết kế khối đo các thông số điện năng

EMIC ADE7753:

ADE7753 là 1 IC do hãng analog device sản xuất và nó có những tính năng cơ bản sau:

- Độ chính xác cao
- Tích hợp bộ tích phân
- Tính giá trị hiệu dụng của dòng điện và điện áp với sai số nhỏ hơn 0,1%
- Hiệu chuẩn số cho nguồn, pha và bù đầu vào
- Lập trình tần số xng ngõ ra
- Yêu cầu ngắt ở chân IRQ và thanh ghi trạng thái



Hình 3.23: EMIC ADE7753

- Điện áp chuẩn 2.4V, cho đưa từ ngoài
- Nguồn nuôi 5V, công suất thấp 25mW

Bảng 3.6 Sơ đồ và chức năng các chân của EMIC ADE7753

| Chân số | Ký hiệu | Mô tả |
|---------|---------|---|
| 1 | RESET | Thiết lập lại ADE7753, tín hiệu ở mức thấp sẽ giữ cho ADC và những mạch kỹ thuật số ở chế độ khởi động lại. |
| 2 | DVDD | Nguồn mạch số, chân này sẽ cung cấp nguồn cho các mạch số trong ADE7753. Điện áp cần được duy trì ở mức $5V \pm 5\%$. Chân AVDD được tách riêng với DGND |

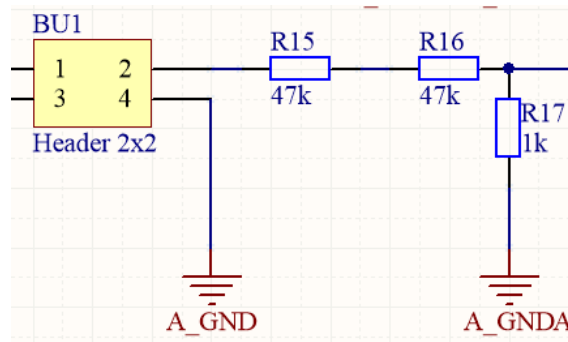
Chương 3: Thiết kế phần cứng

| | | |
|-----|---------|--|
| | | qua hai tụ điện mắc song song là tụ $10\mu\text{F}$ và tụ gốm $100\mu\text{F}$. |
| 3 | AVDD | Nguồn mạch tương tự, chân này cung cấp điện áp tương tự cho các mạch tương tự trong ADE7753. Điện áp cần được duy trì ở mức $5\text{V}\pm 5\%$. Chân AVDD được tách riêng với DGND qua hai tụ điện mắc song song là tụ $10\mu\text{F}$ và tụ gốm $100\mu\text{F}$. |
| 4,5 | V1P,V1N | Tín hiệu tương tự sẽ vào bởi kênh 1. Kênh được thiết kế để sử dụng với một bộ chuyển đổi dòng di/dt (như cuộn Rogowski), cảm biến dòng (ví dụ Shunt) hoặc biến áp dòng CT. |
| 6,7 | V2P,V2N | Tín hiệu tương tự sẽ vào kênh 2. Kênh 2 được thiết kế để sử dụng dung cho bộ chuyển đổi điện áp. Tín hiệu đầu vào lớn nhất là $\pm 0.5\text{V}$, có thể chịu được quá điện áp là $\pm 6\text{V}$. |
| 8 | AGND | Điện áp chuẩn tương tự, chân này cung cấp cho sự so sánh của tất cả các mạch tương tự bên trong ADE7753 như ADCs. Chống xung răng cưa của bộ lọc, bộ biến dòng, và điện áp. |
| 9 | REF | Truy cập và kiểm tra điện áp trên Chip, các giá trị điện áp trên Chip là $2,4\text{V}\pm 8\%$ và giá trị nhiệt độ là 30 ppm/oC . Một số nguồn bên ngoài cũng so sánh ở chân này. Trong cả hai trường hợp, chân này cần được tách riêng để nối với tụ $1\mu\text{F}$. |
| 10 | DGND | Điện áp chuẩn số, chân này cung cấp sự so sánh cho các mạch số bên trong ADE7753 như bộ nhân, bộ lọc, và bộ chuyển đổi tần số. Bởi vì dòng điện số trong ADE7753 là rất nhỏ. Tuy nhiên các điện dung cao trên chân |

Chương 3: Thiết kế phần cứng

| | | |
|----|--------|--|
| | | DOUT có thể dẫn đến tín hiệu số bị nhiễu, làm ảnh hưởng đến sự so sánh. |
| 11 | CF | Tần số đầu ra mức logic. Chân CF cung cấp thông tin về công suất hiệu dụng. Các tần số ra sẽ được điều chỉnh bởi CSDEN và CFNUM. |
| 12 | ZX | Điện áp dạng sóng (kênh 2) |
| 13 | SAG | Cổng này mở ra khi mức công suất xuống thấp. |
| 14 | IRQ | Ngắt ngõ ra. Khi công suất xuống thấp cổng này sẽ mở ra. |
| 15 | CLKIN | Xung đồng hồ ngõ vào cho ADEs và quá trình điều chế tín hiệu số xung đồng hồ là 3.579545 MHz. Dùng 2 tụ điện cho mạch là 22pF và 33pF. |
| 16 | CLKOUT | Cung cấp xung đồng hồ cho ADE7753. Có thể dung một tải CMOS. |
| 17 | CS | Chọn Chip. Dùng 4 dây nối tiếp SPI. ADE7753 sẽ hoạt động với đầu vào là mức thấp. |
| 18 | SCLK | Xung đầu vào đồng bộ. Tất cả dữ liệu truyền nối tiếp được quản lý bởi xung này. |
| 19 | DOUT | Dữ liệu đầu ra, dữ liệu được chuyển khi sườn lên của SCLK tác động. Trong trường hợp trở kháng cao, sẽ xuất ra mức logic thông thường. |
| 20 | DIN | Dữ liệu đầu vào, dữ liệu được chuyển khi sườn xuống của SCLK tác động. |

Thiết kế khối chuyển điện áp:



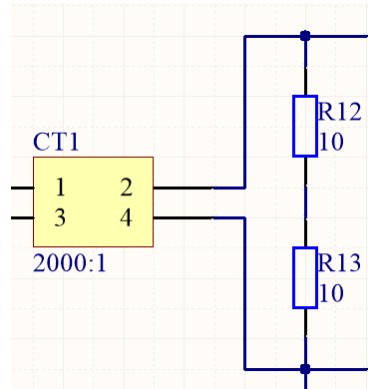
Hình 3.24: Khối chuyển đổi điện áp

Dùng biến áp tỷ số 220/12. Chọn $V_{max} = 250\sqrt{2}$ V, $V_{R17max} = 250\text{mV}$, $R17 = 1\text{k}\Omega$.

$$V_{R17} = \frac{V_{max} \cdot 12}{220} * \frac{R17}{R17 + R15 + R16} = 19.28 * \frac{1}{1 + 2 \cdot R15} \Rightarrow R15 = R16 > 38.07\text{k}\Omega$$

Chọn $R15 = R16 = 47\text{ k}\Omega \Rightarrow$ điện áp hiệu dụng tối đa đo được là 307V.

Thiết kế khối chuyển dòng điện:



Hình 3.25: Khối chuyển dòng điện

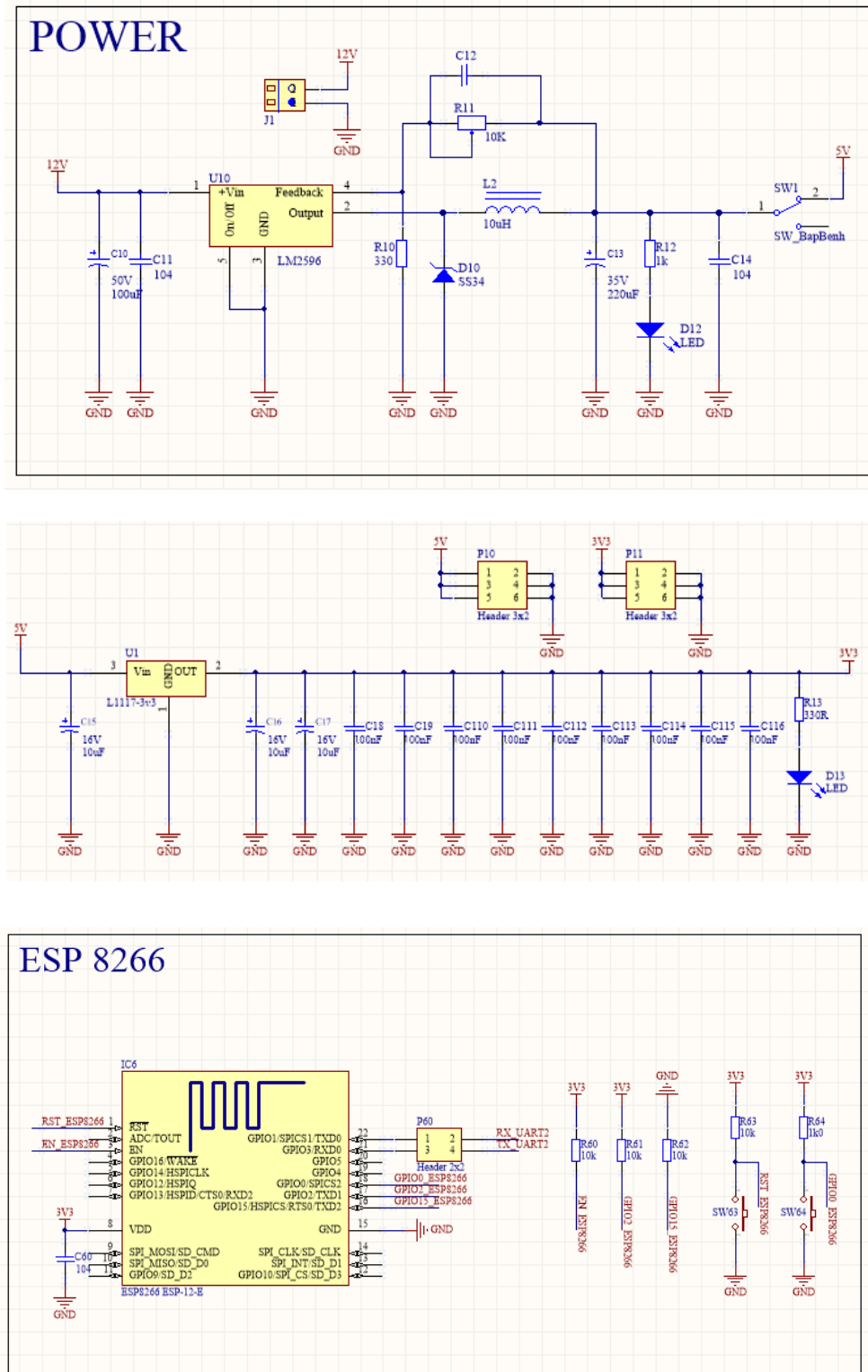
Chọn biến dòng 2000:1 Dòng điện cực đại cần đo 10A

$$V_{in} = \frac{10}{2000} * (R12 + R13)$$

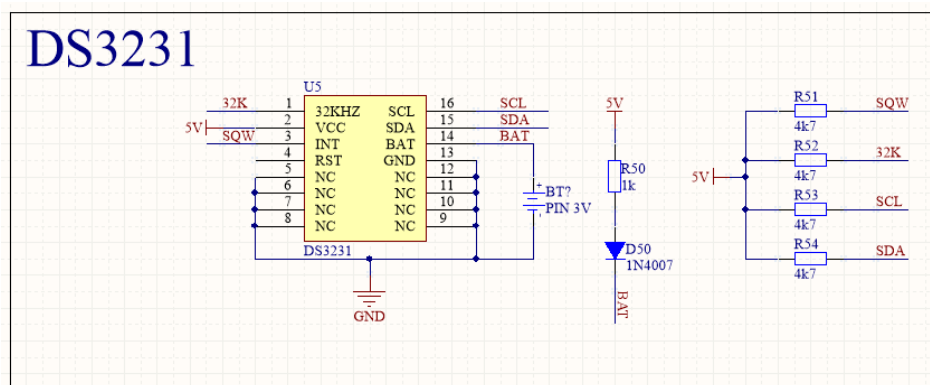
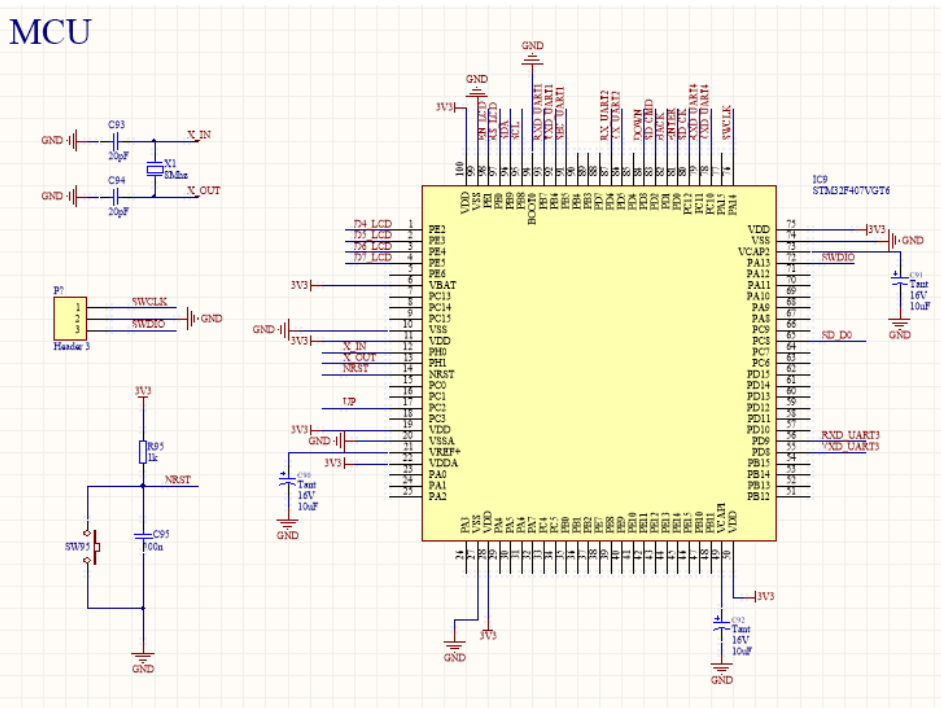
Mà $V_{in} < 0.25/\sqrt{2} \Rightarrow R12 = R13 < 17.7\text{ }\Omega$. Chọn $R12 = R13 = 10\text{ }\Omega \Rightarrow$ Dòng điện tối đa đo được là 17.7A.

3.2.9. Sơ đồ thiết kế các mạch hoàn chỉnh

Mạch Master:

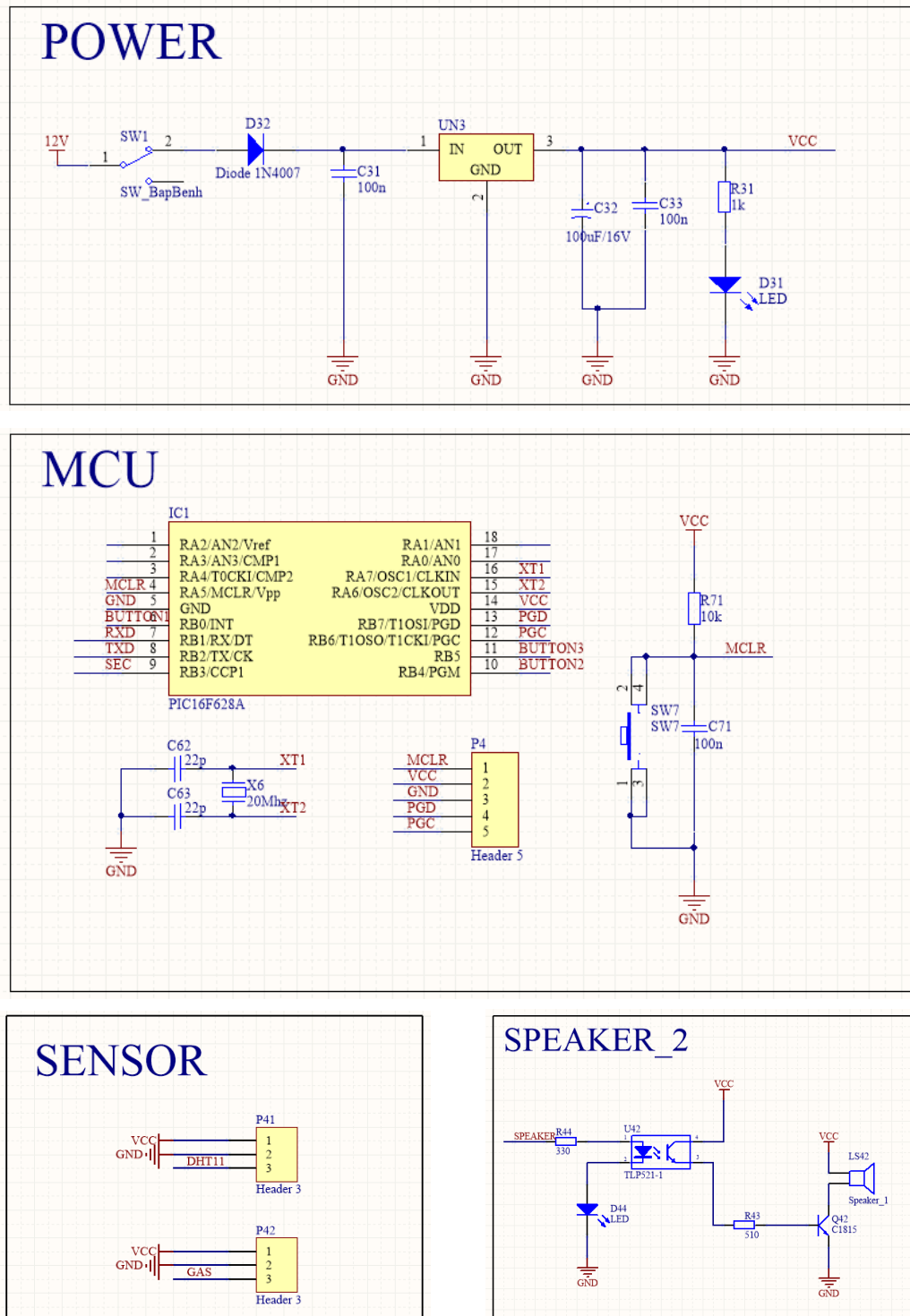


Hình 3.26: Sơ đồ nguyên lý mạch Master

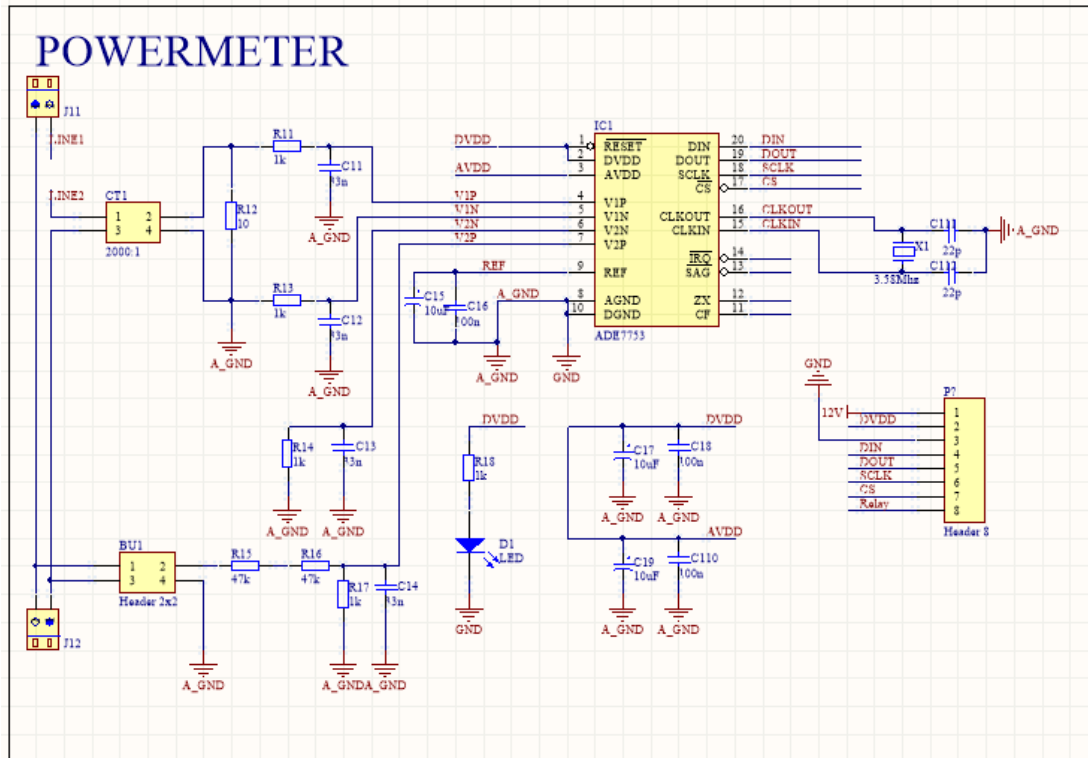


42

Mạch Slave Sensor:



Hình 3.30: Sơ đồ nguyên lý mạch Slave Sensor



Hình 3.32: Sơ đồ nguyên lý mạch Slave Compteur

3.3. Mạch giám sát hệ thống qua camera:

Ngày trước, những khi nhắc đến camera giám sát thì người dùng thường nghĩ tới việc ngồi tại chỗ, nơi lắp đặt màn hình theo dõi hình ảnh camera quan sát thì mới có thể xem những gì đang diễn ra ở nơi lắp đặt camera. Chứ một khi đã bước ra khỏi phạm vi đó thì không thể nào xem được. Tuy nhiên với sự phát triển không ngừng của hệ thống mạng internet thì hầu như rào cản đó đã bị phá bỏ, người dùng không còn bị gò bó nơi sử dụng như trước kia nữa. Hình ảnh camera giám sát dần dần được cải tiến sang bước mới là xem camera qua mạng internet. Cho dù bạn đi đến đâu, miễn ở đó có internet, có thiết bị truy cập như máy tính, laptop thì có thể xem được camera giám sát tại nơi đó. Mạch giám sát camera sử dụng máy tính nhưng Raspberry Pi 3 và camera Logitech C270 dùng để phát trực tiếp camera lên webserver. Người dùng có thể dùng laptop để truy cập vào web xem camera.



Hình 3.33: Máy tính nhúng Raspberry Pi 3

Máy tính Raspberry Pi 3 là board mạch Mini Computer được sử dụng nhiều nhất hiện nay, ngoài việc sử dụng như một máy tính bình thường chạy hệ điều hành Linux hoặc windows 10 IoT, máy còn có khả năng xuất tín hiệu ra 40 chân GPIO

Thông số kỹ thuật:

- CPU phiên bản BCM2837 tốc độ 1.2Ghz 4 nhân với kiến trúc ARM Cortex-A53 64bit
- Tích hợp chuẩn Wifi 802.11n và Bluetooth 4.1
- Thẻ microSD
- Giao tiếp camera
- Port HDMI giao tiếp màn hình
- 4 cổng USB 2.0 giao tiếp ngoại vi

Logitech C270 là webcam cho phép người dùng quay phim với độ phân giải HD720. Với công nghệ Fluid Crystal có thể tạo những hình ảnh và âm thanh rõ nét trung thực. Ngoài ra chế độ chỉnh sáng tự động giúp hình ảnh trở nên sáng hơn kể cả trong điều kiện ánh sáng tối.

Chương 3: Thiết kế phần cứng

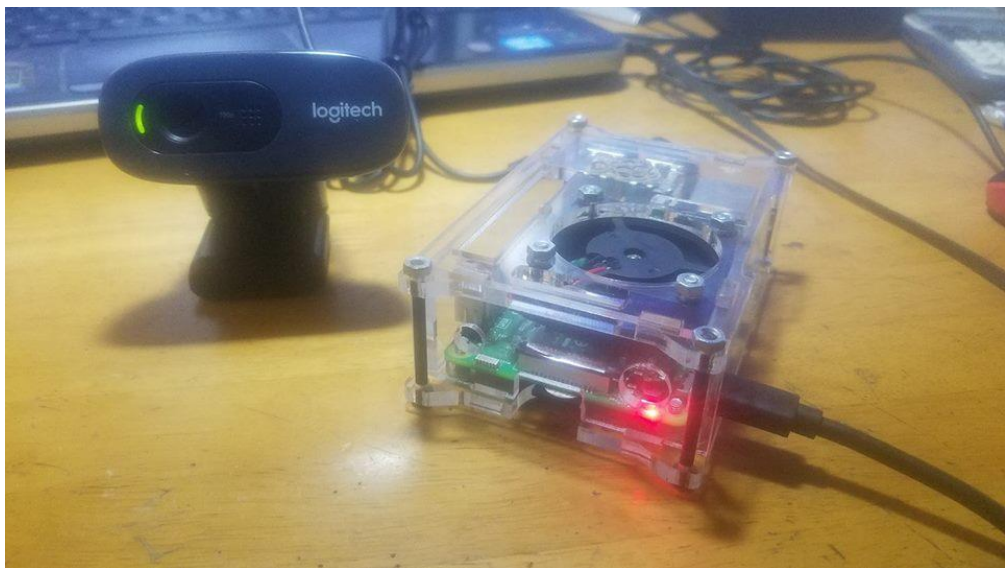
Sau khi hoàn thành nhiệm vụ luận văn, còn ít thời gian, tôi nghiên cứu thêm về linux và máy tính nhúng sau đó dùng webcam đưa hình ảnh trực tiếp lên webserver để giám sát từ xa.

Thông số kỹ thuật:

- Camera 1.3MP chuẩn HD 720p
- Tương thích với nhiều hệ điều hành khác nhau



Hình 3.34: Camera Logitech C270



Hình 3.35: Kết nối Rasp Pi 3 và Camera

Chương 4. THIẾT KẾ HỆ THỐNG

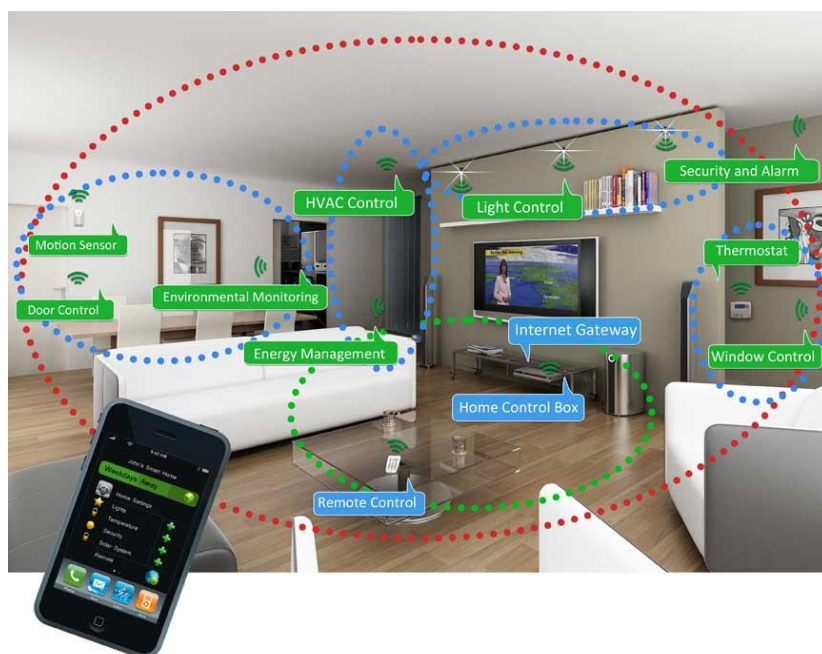
4.1. Các tính năng kỹ thuật

4.1.1. Khả năng đóng ngắt tiện dụng

Mạch relay để có thể đóng/ ngắt thiết bị điện AC. Với các chế độ điều khiển tiện dụng như công tắc 2 trạng thái, điều khiển trực tiếp tại nhà hoặc khi đang ở bên ngoài thông qua chuẩn kết nối wifi sử dụng smartphone, laptop. Điều này sẽ vô cùng tiện lợi trong trường hợp chúng ta muốn máy lạnh bật sẵn để nhiệt độ phòng đủ lạnh trước khi về nhà. Bên cạnh đó việc kết hợp thiết bị đóng - ngắt với các hoạt cảnh được định sẵn: Như chế độ tiếp khách, chế độ không cần nhiều ánh sáng...

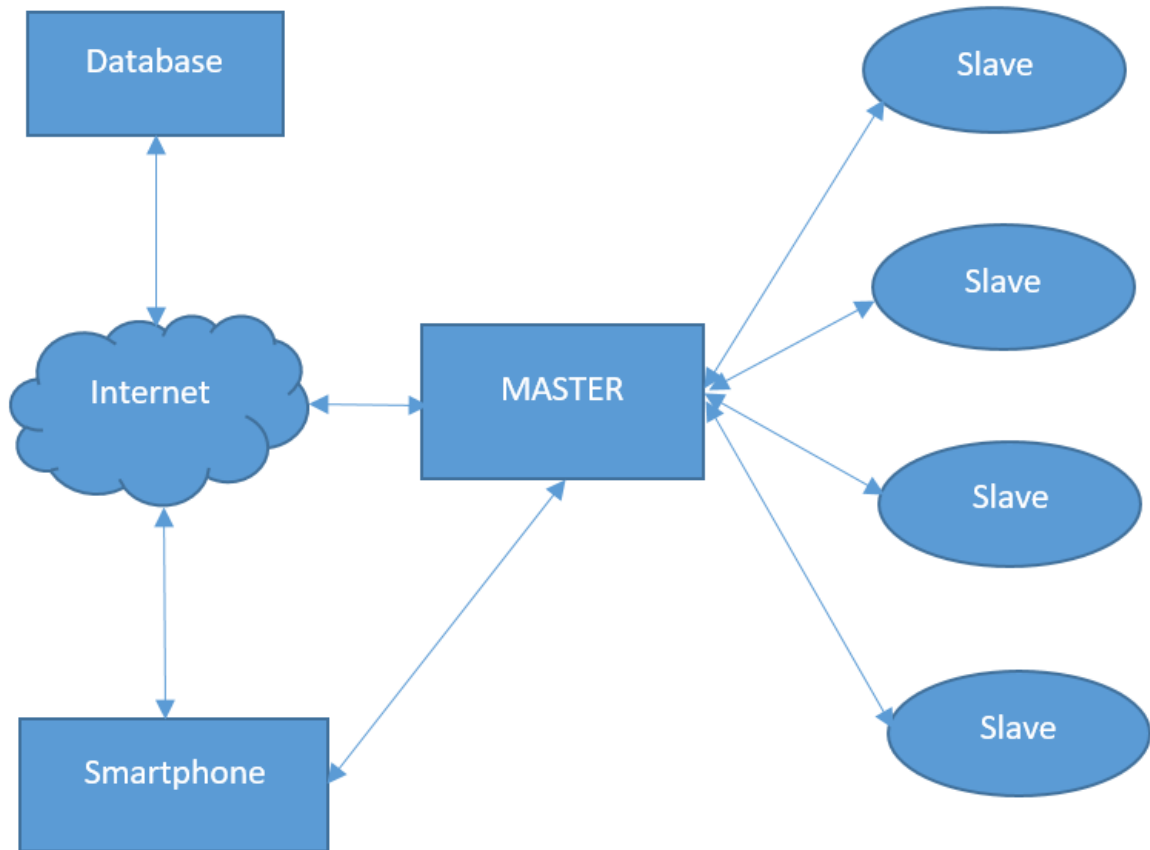
Tính năng hẹn giờ bật tắt thiết bị. Ngôi nhà có thể được hẹn giờ để bật tắt thiết bị theo nhu cầu của chủ nhân. Trang bị cảm biến nhiệt độ và độ ẩm giúp chúng ta biết được trạng thái hiện tại của ngôi nhà.

Bên cạnh đó hệ thống còn được trang bị cảm biến khí gas giúp chúng ta có thể tránh được những nguy hiểm từ việc rò rỉ gas.



Hình 4.1: Tổng quan chức năng mô hình smarthome

4.1.2. Sơ đồ khối chức năng tổng quát



Hình 4.2: Sơ đồ khối chức năng smathome

Mô tả sơ đồ khối:

Khối Master:

Gồm 1 IC STM32F407VGT6 được hàn thẳng lên board và không dùng kit nhằm tiết kiệm chi phí cũng như làm board mạch gọn lại rất nhiều. Ngoài ra còn IC thời gian thực DS3231, chip wifi ESP8266, khối giao tiếp RS485, module GSM và LCD. Master có nhiệm vụ quản lý các Slave, nhận dữ liệu từ các Slave sau đó gửi thông tin truyền tương ứng với các nhiệm vụ được giao. Master kết nối với internet qua wifi để nhận thông tin điều khiển từ internet cũng như cập nhật thông tin hệ thống lên webserver. Ngoài ra người dùng có thể điều khiển hệ thống qua GSM phòng khi đang ở xa không có internet.

Khối Slave Button:

Slave Button dùng vi điều khiển PIC 16F628A của microchip có nhiệm vụ điều khiển các Slave bằng các gửi frame tín hiệu lên đường truyền tới Master để thực hiện các chức năng nhất định.

Khối Slave Device:

Khối Slave Device sử dụng vi điều khiển PIC 16F628A của microchip có nhiệm vụ nhận tín hiệu từ master để bật tắt các relay nhằm điều khiển thiết bị AC.

Khối Slave cảm biến:

Khối Slave cảm biến sử dụng vi điều khiển PIC 16F628A của microchip có nhiệm vụ gửi các tín hiệu nhiệt độ, độ ẩm, cảm biến khí gas về master xử lý.

Khối Slave Compteur:

Khối Slave Compteur có các tính năng của khối Slave Device nhưng có thêm tính năng giám sát và đo lường V, I, P nhằm giám sát công suất và điện năng tiêu thụ của thiết bị từ đó truyền dữ liệu lên webserver.

4.1.3. Giao thức truyền thông

Trong mô hình Smarthome được thiết kế trong luận văn. Việc truyền dẫn dữ liệu giữa các thiết bị bằng cách thông qua các cổng UART của vi điều khiển, từ đó tín hiệu được đưa đến IC RS485 và đưa lên đường truyền. Như vậy, chuẩn giao tiếp RS485 sử dụng duy nhất 1 cáp mạng sẽ kết nối đến tất cả thiết bị trong mô hình Smarthome. Trong đó cáp mạng sẽ gồm 4 dây bao gồm:

- Dây 12V: có chức năng đưa nguồn điện đến các mạch điều khiển
- Dây A: Dây truyền tín hiệu
- Dây B: Dây truyền tín hiệu
- Dây GND: Kết nối đất chung giữa các mạch điều khiển.

Tuy nhiên việc truyền dẫn sử dụng chuẩn giao tiếp RS485 chỉ mang tính truyền dẫn vật lý vì vậy đòi hỏi một giao thức truyền để có thể quản lý dữ liệu một cách khoa học từ đó phát triển mô hình thêm nhiều thiết bị hơn. Ngoài ra, để IC RS485 hoạt động ta cần điều khiển trực tiếp chân cho phép hoạt động

của IC RS485 bằng vi điều khiển. Chính vì vậy việc thiết kế và xây dựng một giao thức truyền là vô cùng cần thiết trong việc thiết kế mô hình Smarthome. Trong quá trình tìm hiểu và nghiên cứu để thiết kế một giao thức truyền phù hợp với mô hình luận văn. Tôi có tìm hiểu và đưa ra hai mô hình truyền dẫn dựa trên các mô hình trên lý thuyết như sau:

Thiết kế giao thức theo hình thức hỏi vòng tuần tự:

- Master:

Master có nhiệm vụ hỏi vòng tuần tự các Slave được kết nối trong hệ thống. Sau khi gửi một thông tin hỏi để yêu cầu làm việc đến một Slave, Master sẽ Delay một thời gian nhỏ để chờ thông tin từ Slave đang được hỏi. Nếu như Slave đang được hỏi không có yêu cầu làm việc gì thì Master sẽ bỏ qua và tiến hành hỏi một Slave khác. Nếu một Slave Button đang được Master hỏi và có tín hiệu từ bên ngoài. Slave này sẽ được quyền ngắt vòng hỏi của Master bằng cách gửi một thông tin đến cho Master để yêu cầu quyền làm việc, sau đó thông tin nhận được sẽ được ưu tiên truyền dữ liệu trực tiếp đến Slave Device để thực hiện công việc được yêu cầu. Kết thúc quá trình kết nối giữa Master-Slave Input- Slave Output, Master sẽ tiếp tục hỏi tuần tự các Slave sau đó lặp lại chu kỳ hỏi lặp. Thời gian hỏi lặp của Master phải rất nhỏ (tính bằng ms) để có thể hỏi tất cả các Slave đang hiện có của hệ thống. Ngoài ra, thời gian hỏi và thời gian nhận tín hiệu từ Slave phải trùng nhau để tín hiệu có thể được gửi đi.

- Slave Button:

Có nhiệm vụ chờ được Master hỏi và gửi thông tin lên trên đường truyền. Slave Button luôn ở trong trạng thái nhận dữ liệu (tức chân Enable của RS485 luôn ở trạng thái LOW). Khi đồng thời được hỏi và có tín hiệu từ bên ngoài Slave Input sẽ bật trạng thái Enable của RS485 lên HIGH và gửi thông tin theo thông tin được yêu cầu từ bên ngoài.

- Slave Device:

Chân Enable của RS485 của Slave Device luôn ở trạng thái LOW (tức trạng thái chờ làm việc). Khi được Master hỏi hoặc có dữ liệu gửi từ một Slave Button. Slave Device có nhiệm vụ phân tích và xử lý thông tin được truyền đến. Nếu thông tin truyền đúng với cài đặt thì sẽ làm việc theo chức năng định sẵn.

- Phân tích:

Như vậy quá trình hỏi lặp sẽ được diễn ra liên tục. Việc này sẽ hạn chế hiện tượng trùng khi có 2 tín hiệu được gửi cùng một lúc, khi Slave nào được hỏi Slave đó sẽ trả lời. Tuy nhiên sẽ rất khó quản lý dữ liệu vì những nguyên nhân sau đây:

- Về vấn đề thời gian: Sau khi Master gửi thì sẽ phải một khoảng thời gian delay để chờ nhận thông tin truyền từ Slave và chuyển trạng thái RS485 về trạng thái nhận (LOW). Bên cạnh đó là một khoảng thời gian Delay để có thể đặt trạng thái của RS485 lên trạng thái (HIGH) trạng thái gửi. Như vậy khi càng có nhiều Slave thì thời gian Delay sẽ càng lớn lên gây thông tin truyền sẽ trở nên chậm.

- Về vấn đề quản lý: Việc truyền thông tin liên tục như vậy sẽ dẫn đến trên đường truyền sẽ bận liên tục, điều này sẽ dẫn đến phí phạm tài nguyên chưa kể là trường hợp có 2 thông tin truyền cùng lúc sẽ dẫn đến sai lệch thông tin truyền, ngoài ra còn các trường hợp nhiễu, tín hiệu truyền liên tục nên các Slave sẽ phải nhận và xử lý thông tin một cách liên tục dẫn đến tình trạng xử lý sai. Mất frame truyền. Vì nhiều bất cập trong việc sử dụng hỏi vòng lặp nên trong luận văn đã chuyển sang xây dựng một giao thức truyền thông khác dựa trên lý thuyết về giao thức truyền thông CSMA/CD.

Thiết kế giao thức theo hình thức hỏi vòng tuần tự:

Dựa trên lý thuyết về giao thức truyền thông CSMA/CD, nghĩa là đa truy cập nhận biết sóng mang phát hiện xung đột. Tư tưởng của nó là: khi một máy trạm cần truyền dữ liệu trước hết phải “nghe” xem đường truyền bận hay rỗi. Nếu đường truyền rỗi thì truyền dữ liệu theo khuôn dạng chuẩn. Ngược lại,

nếu đường truyền đang bận (đã có máy trạm khác truyền dữ liệu rồi) thì máy trạm đợi một khoảng thời gian ngẫu nhiên rồi bắt đầu nghe lại hoặc tiếp tục nghe cho đến khi đường truyền rỗi thì truyền đi với xác suất $p(0 < p < 1)$. Để có thể phát hiện xung đột, người ta bổ sung thêm quy tắc “nghe trong khi nói” tức là trong khi một trạm đang truyền nó vẫn “nghe” đường truyền. Nếu phát hiện xung đột thì nó dừng ngay việc truyền và phát đi sóng mang báo hiệu xung đột cho máy trạm khác. Việc phát hiện xung đột đã khiến cho việc truyền dẫn thông tin trên cáp được đảm bảo hơn. Việc xây dựng giao thức truyền thông được dựa trên lý thuyết về giao tiếp Master - Slave kết hợp với lý thuyết về lắng nghe đường truyền rỗi cụ thể như sau.

- Master:

Nhiệm vụ quan trọng nhất trong giao thức truyền thông. Khi hệ thống được thiết lập Master sẽ được khởi động đầu tiên. Master sẽ ở trạng thái chờ thông tin truyền, khi có một Slave hòa vào mạng giao tiếp thì Master sẽ nhận một Frame đăng kí đến từ Slave sau đó Master sẽ phản hồi lại để cho phép Slave được hoạt động, chỉ khi Slave nhận được Frame cho phép chuyển sang trạng thái hoạt động từ Master thì Slave mới có quyền được hoạt động. Sau khi gửi dữ liệu để các Slave được hoạt động Master sẽ chuyển về trạng thái chờ. Ngoài việc ghi nhận, quản lý các Slave đăng kí trạng thái hoạt động Master còn hoạt động giống như một trung gian để truyền thông tin giữa các Slave. Khi các frame truyền từ Slave Input yêu cầu các công việc, chỉ duy nhất Master có thể xử lý thông tin được truyền đi đó và từ đó gửi thông tin truyền đến các Slave. Chứ không có kết nối trực tiếp giữa Slave Input và Slave Output trên đường truyền. Ngoài nhiệm vụ trên, Master sẽ có nhiệm vụ khi nhận trạng thái hoạt động của Slave và lưu vào các biến quản lý, bên cạnh đó là các trạng thái hoạt động của Slave Button và Slave Device đều được lưu lại và xử lý để quản lý đường truyền một cách hiệu quả.

- Slave Button:

Slave Button sẽ có nhiệm vụ gửi Frame truyền đăng kí trạng thái hoạt động đến Master, sau đó nếu được Master cho phép hoạt động thì Slave Button sẽ chuyển về trạng thái hoạt động. Khi ở trạng thái hoạt động, Slave Button sẽ ở trạng thái chờ (Chân Enable của RS485 ở trạng thái LOW). Khi có yêu cầu làm việc từ bên ngoài Slave Button sẽ đọc và xử lý thông tin từ đó gửi Frame truyền đến Master sau đó trở về trạng thái chờ hoạt động. Tuy nhiên trong quá trình gửi dữ liệu Slave Input sẽ lắng nghe xem đường truyền bận hay rỗi. Nếu đường truyền rỗi thì sẽ truyền dữ liệu như thông thường. Ngược lại, nếu đường truyền đang bận (đã có Slave khác truyền dữ liệu lên trên đường truyền) thì Slave này sẽ đợi một khoảng thời gian rồi bắt đầu truyền dữ liệu lên trên đường truyền sau đó trở về trạng thái truyền và tiếp tục lắng nghe đường truyền.

- Slave Device:

Khi được khởi động và hòa vào mạng truyền thông của mô hình. Slave Device sẽ có nhiệm vụ gửi Frame truyền đăng kí trạng thái hoạt động đến Master, sau đó nếu được Master cho phép hoạt động thì Slave Device sẽ chuyển về trạng thái hoạt động. Khi ở trạng thái hoạt động, Slave Device sẽ ở trạng thái chờ (Chân Enable của RS485 ở trạng thái LOW). Khi nhận được Frame truyền yêu cầu làm việc đến từ Master, Slave Device sẽ chuyển sang trạng thái hoạt động và xử lý yêu cầu đến từ Master sau đó trở về trạng thái chờ hoạt động.

- Phân tích:

Như vậy khi bắt đầu khởi động Master sẽ cập nhật tất cả trạng thái của các Slave. Sau đó cả Master và Slave sẽ đi vào trạng thái “chờ hoạt động”. Phương thức truyền thông sẽ giải quyết các vấn đề vẫn còn tồn tại ở cơ chế hỏi lặp vòng cụ thể như sau:

- Về vấn đề thời gian: Phương thức truyền thông không truyền frame lên đường truyền liên tục nên thời gian Delay liên tục dường như không có, thay vào đó khi một frame được gửi lên đường truyền thì khoảng Delay sẽ rất nhỏ

(đủ để bật trạng thái IC RS485 lên HIGH) sau đó trở về trạng thái chờ. Điều này khiến ta có thể mở rộng thêm nhiều Slave mà không cần quan tâm nhiều về vấn đề Delay.

- Về quản lý: Phương thức truyền thông không truyền frame lên đường truyền liên tục nên đường truyền sẽ trống. Điều này sẽ tránh các trường hợp đường truyền chồng chéo dẫn đến sai frame truyền. Bên cạnh đó với cơ chế lắng nghe đường truyền sẽ giúp giải quyết vấn đề chồng chéo đường truyền khi 2 Slave cùng truyền một lúc. Tuy nhiên tỉ lệ 2 Slave cùng frame rất hiếm xảy vì thời gian gửi và xử lý được tính bằng ms nên tỉ lệ rất rất nhỏ. Trong phương thức truyền như trên thì việc luân chuyển giữa các trạng thái là rất quan trọng. Vì ở mỗi trạng thái các thiết bị sẽ làm những công việc khác nhau. Sau đây là sơ đồ trạng thái đối với các thiết bị sử dụng trong mô hình.

Thiết kế và qui định Frame truyền:

Một Frame truyền gồm 11 byte được cấu trúc theo dạng :

$SX_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}E$

Trong đó byte đầu 'S' có tác dụng để nhận biết bắt đầu Frame truyền.

Byte X_1 dùng để nhận biết Slave truyền cho master hay master truyền cho slave.

$X_1 = 1$: Master truyền cho Slave

$X_1 = 0$: Slave truyền cho Master

Byte X_2 là byte function:

$X_2 = 0$: điều khiển đóng tắt relay

$X_2 = 3$: Đọc dữ liệu từ sensor về

Byte X_3, X_4, X_5 là byte địa chỉ của thiết bị:

D01: Slave Device số 1

B01: Slave Button số 1

C01: Slave cảm biến số 1

Các byte còn lại tùy theo function cụ thể:

- Nếu $X_2 = 0$:

Slave Button gửi cho Master: S00 B01 D01 0 E để tắt Slave Device D01 hoặc S00 B01 D01 1 E để bật Slave Device D01

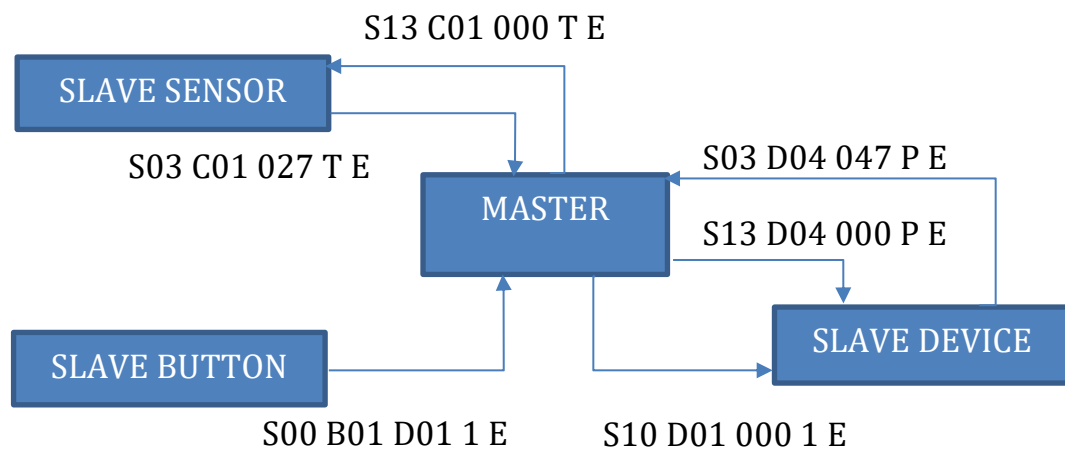
Slave Master sau khi nhận được Frame từ Slave Button sẽ gửi cho Slave Device Frame: S10 D01 000 0 E để tắt hoặc S10 D01 000 1 E để bật Slave Device D01.

- Nếu $X_2 = 3$:

Slave Master gửi cho Slave Sensor: S13 C01 000 T E hoặc S13 C01 000 H E để yêu cầu slave sensor gửi nhiệt độ, độ ẩm về cho mster xử lý hoặc S13 D04 000 P E để yêu cầu Slave Device D04 gửi giá trị công suất về cho Master xử lý.

Slave Sensor muốn gửi dữ liệu cho Slave Master có thể gửi S03 C01 027 TE để cho Slave Master biết là 27 độ C.

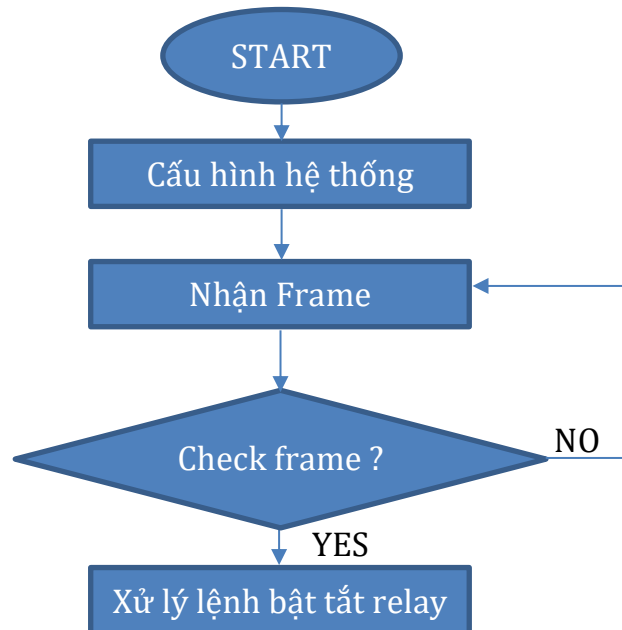
Frame truyền và trạng thái sẽ được thể hiện rõ trong phần code chạy chương trình.



Hình 4.3: Ví dụ frame truyền trong hệ thống

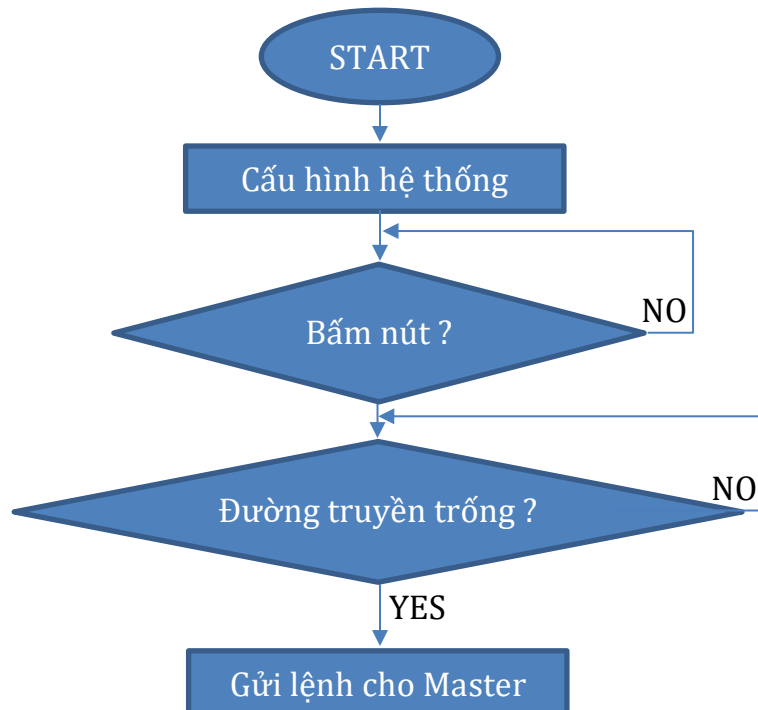
4.2. Thiết kế phần mềm cho Master và Slave

4.2.1. Thiết kế phần mềm cho Slave Device



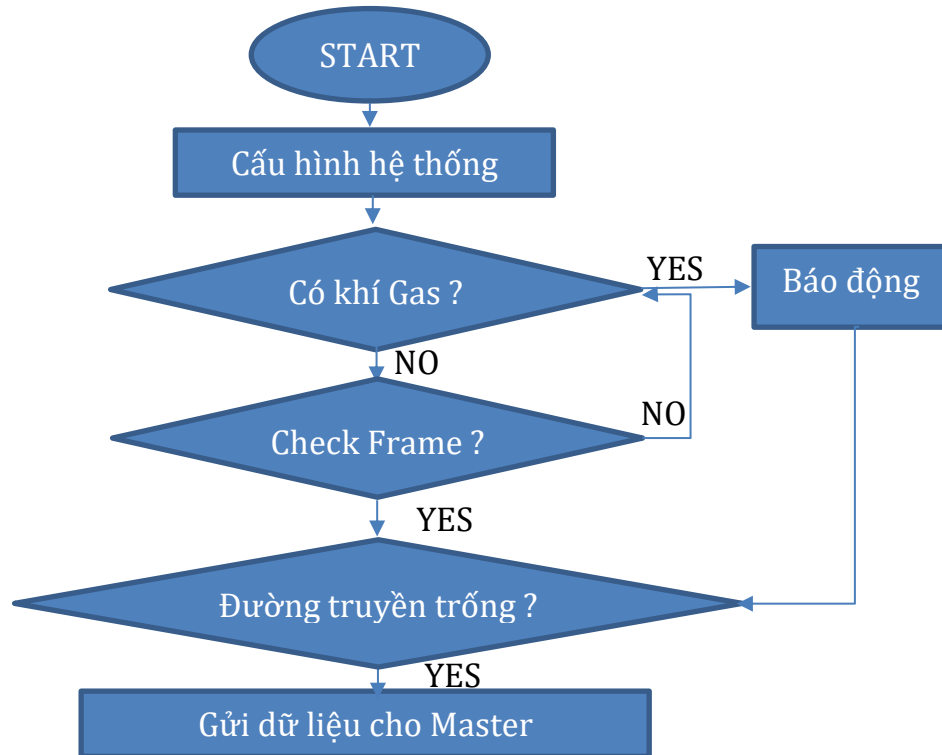
Hình 4.4: Sơ đồ khối Slave Device

4.2.2. Thiết kế phần mềm cho Slave Button



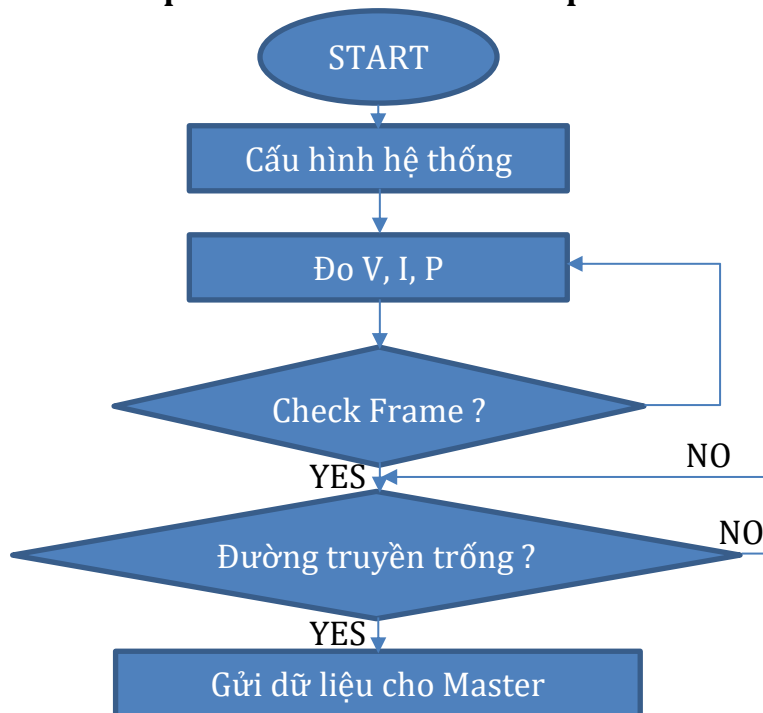
Hình 4.5: Sơ đồ khối Slave Button

4.2.3. Thiết kế phần mềm cho Slave Sensor



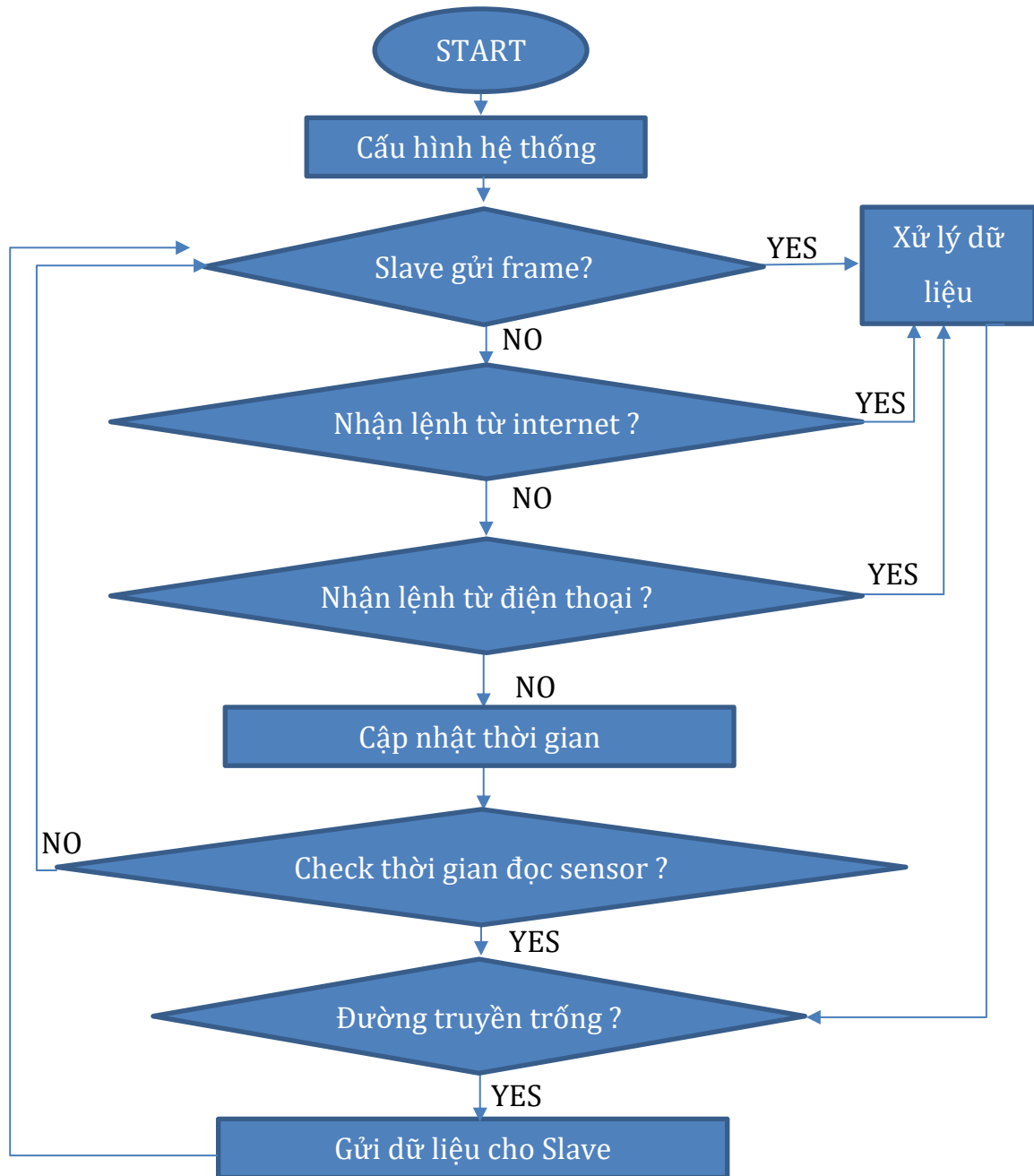
Hình 4.6: Sơ đồ khối Slave Sensor

4.2.4. Thiết kế phần mềm cho Slave Compteur



Hình 4.7: Sơ đồ khối Slave Compteur

4.2.5. Thiết kế phần mềm cho Master



Hình 4.8: Sơ đồ khối Master

4.3. Thiết kế Webserver điều khiển hệ thống nhà thông minh Smarthome

4.3.1. Thiết kế Back-end

Có 2 nền tảng phổ biến cho lập trình webserver: Dùng môi trường server điển hình LAMP (linux-Apache-SQL-PHP) hoặc nền tảng Nodejs chạy trên môi trường V8 JavaScript runtime. Khi dùng LAMP, chúng ta có một webserver là Apache nằm dưới cùng PHP chạy trên nó. Mỗi một kết nối tới server sẽ sinh ra một thread mới và điều này khiến webserver trở nên chậm chạp. Nodejs giải quyết được bài toán này. Một số module của Nodejs giúp ta có thể quản lý kiến trúc lõi của máy chủ, kết quả web sẽ có độ thực thi cao.

Nodejs là một nền tảng serverside được phát triển bởi Ryan Dahl năm 2009 với mục đích tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán. Mặt khác Nodejs là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía server và các ứng dụng liên quan đến mạng. Một số công ty sử dụng Nodejs cho server-side của mình như eBay, GE, Microsoft, Paypal, Uber...



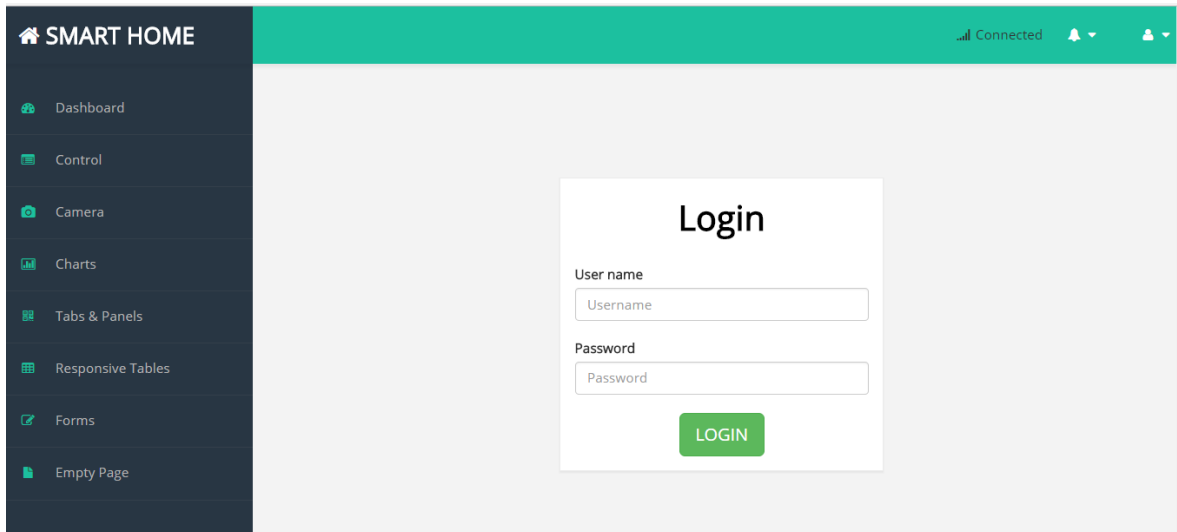
Hình 4.9: Nodejs

Giao tiếp giữa Server và hệ thống:

Trong lập trình webserver có 2 phương thức POST và GET. Cả hai đều là phương thức của giao thức HTTP. Khi hệ thống truyền thông tin lên server, dữ liệu sẽ được mã hóa theo dạng url encoding. Phương thức GET sẽ để lộ thông

tin trên đường truyền url nhưng tốc độ xử lý rất nhanh. Mặt khác, phương thức POST bảo mật hơn vì mọi dữ liệu đều được gửi ngầm, không đưa lên URL bằng việc sử dụng secure HTTP. Ở đây tôi dùng kết hợp nhuần nhuyễn cả 2 phương thức POST và GET để gửi thông tin truyền lên và truyền xuống hệ thống.

Login vào hệ thống dùng phương thức POST. Dữ liệu sẽ được bảo mật:



Hình 4.10: Trang Login vào hệ thống

Mongodb:

Mọi số liệu gửi lên từ hệ thống sẽ được lưu trữ vào database nhằm sau này có thể đem ra xử lý, lập đồ thị... Ở đây tôi dùng mongodb để làm database. Dữ liệu công suất của thiết bị sẽ được gửi lên internet và lưu vào mongodb dưới dạng json:

```
{“deviceId”: “D04”, “date”: 6, “month”: 12, “year”: 2017, “time”: 9:45, “P”: 60}
```

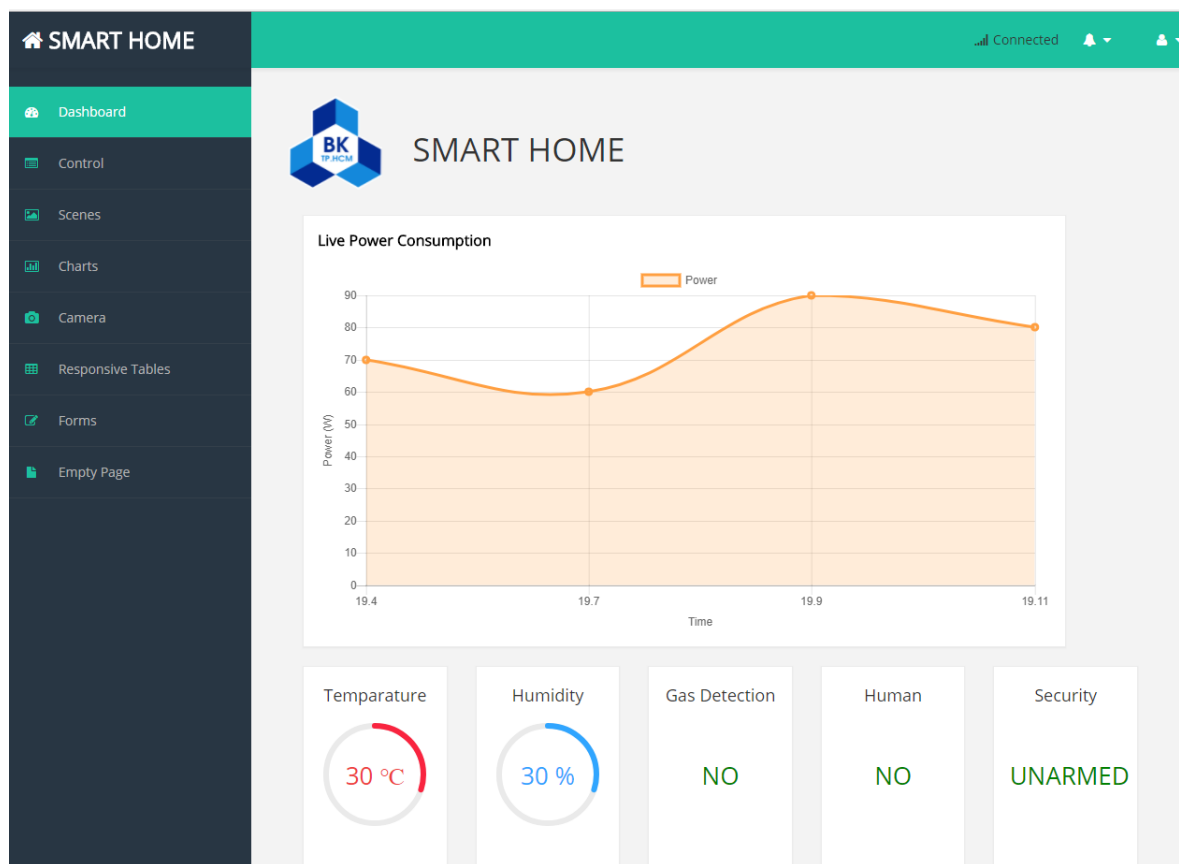
Sau đó những dữ liệu này sẽ được đem ra phân tích và xử lý.

4.3.2. Thiết kế Front-end

HTML và CSS:

HTML là ngôn ngữ được sử dụng để nói cho trình duyệt web biết đâu là nơi bắt đầu một trang web. Vì vậy chúng ta có thể xác định tiêu đề, đoạn văn bản, liên kết, hình ảnh...

CSS là ngôn ngữ cung cấp hình thức và định dạng cho các trang web. Nói cách khác CSS sẽ làm cho trang web trông đẹp hơn.



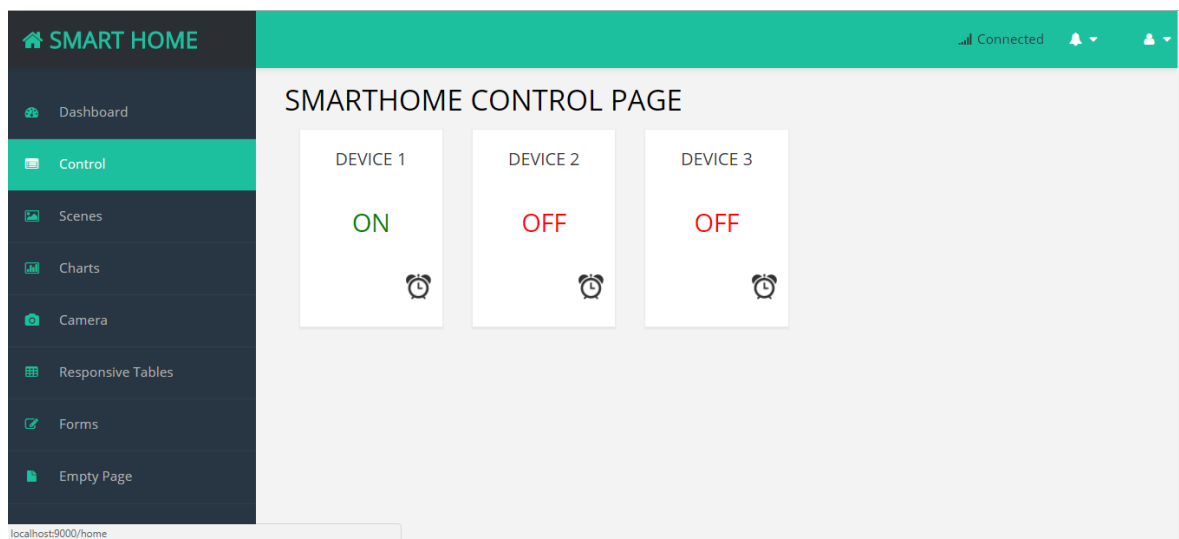
Hình 4.11: Webserver điều khiển nhà thông minh SmartHome

AJAX:

Ajax cho phép các trang web được cập nhật một cách không đồng bộ bằng cách trao đổi các dữ liệu nhỏ với server. Tức Ajax giúp cập nhật các phần nhỏ trong trang mà không cần phải tải lại trang.

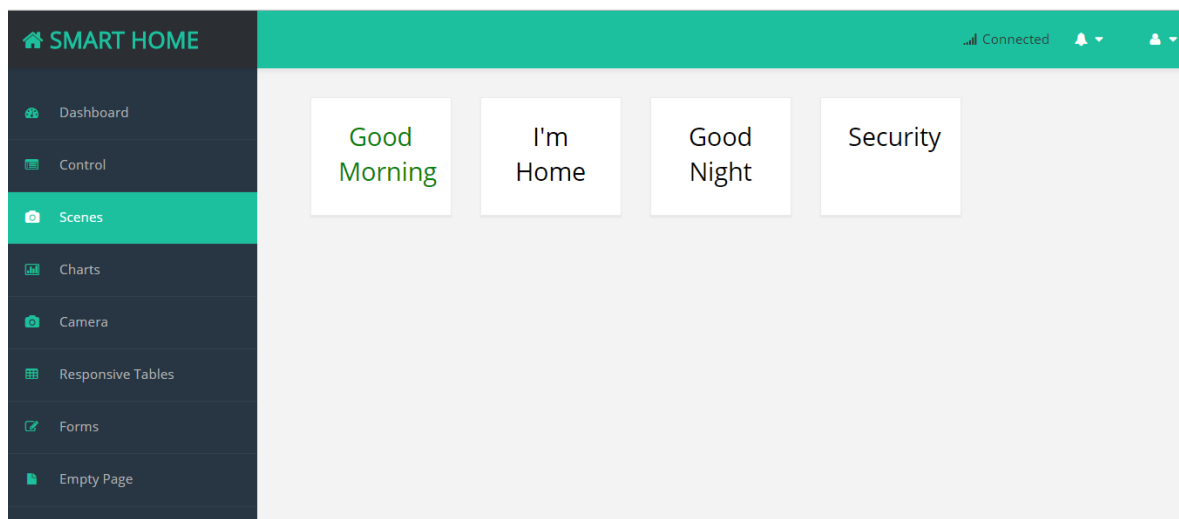
4.3.3. Webserver sau khi thiết kế

Trang điều khiển thiết bị:



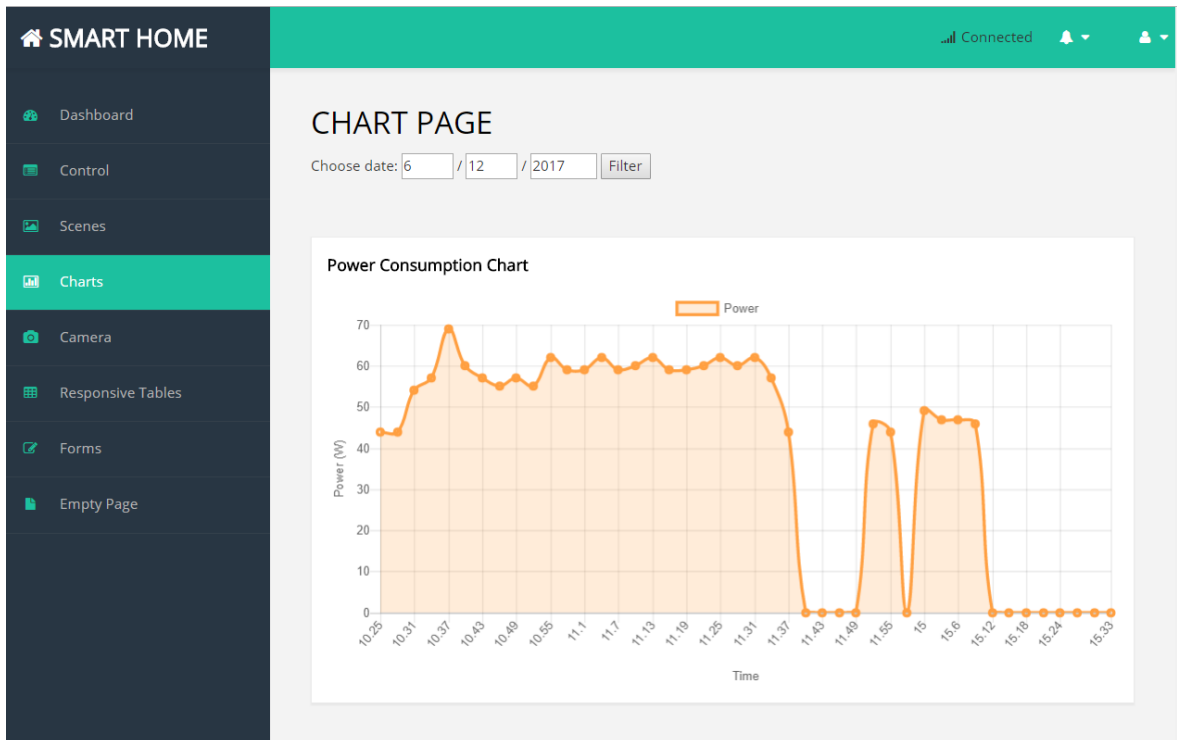
Hình 4.12: Trang điều khiển thiết bị

Trang ngữ cảnh:

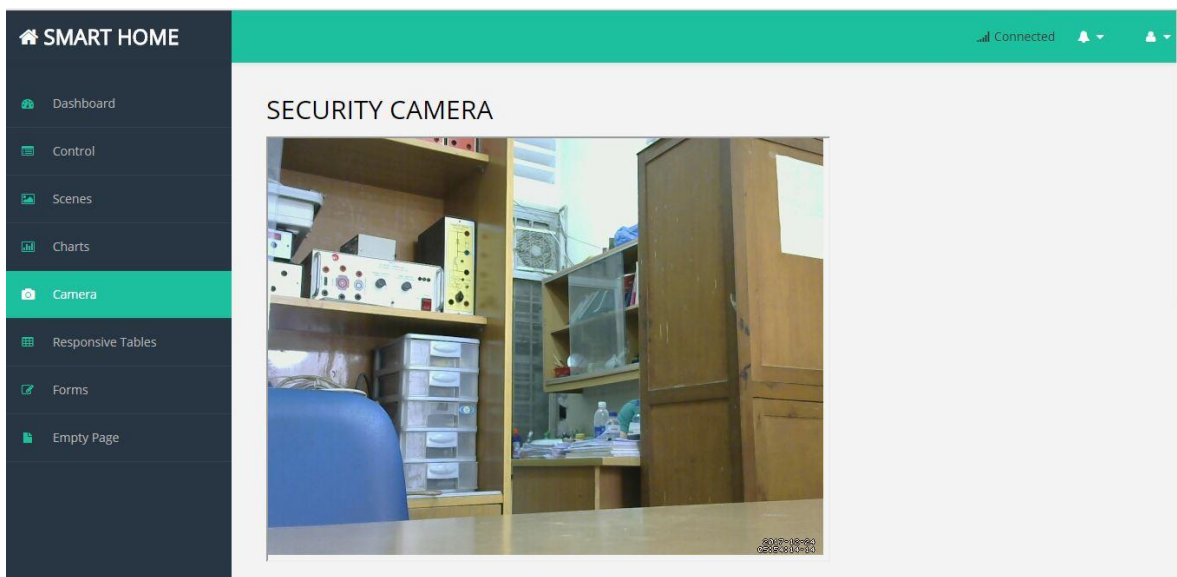


Hình 4.13: Trang ngữ cảnh của hệ thống

Trang hiển thị Chart:



Hình 4.14: Trang hiển thị chart hệ thống

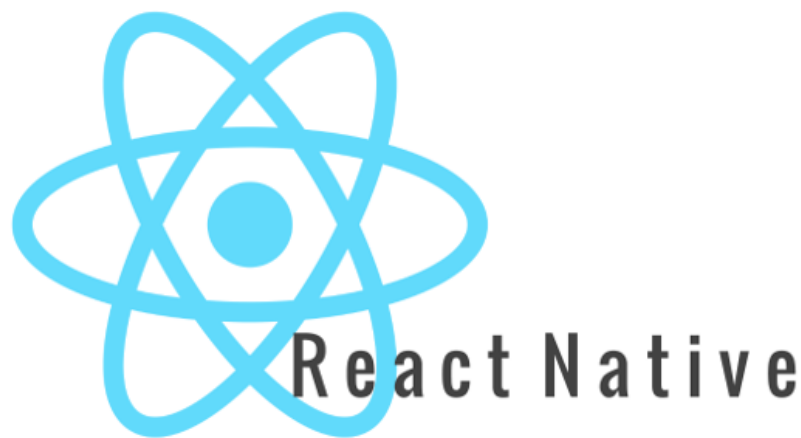


Hình 4.14: Trang hiển thị camera an ninh

4.4. Thiết kế App Android điều khiển nhà thông minh Smarthome

4.4.1. React Native

React Native là công nghệ được tạo bởi Facebook, cho phép làm ứng dụng app cả trên Android và iOS. Một lý do lập trình mobile app rất khó và tốn thời gian là vì thực tế để viết một app chạy cả trên 2 hệ điều hành Android và iOS chúng ta cần học Java và Android SDK. Và nếu muốn phát triển cả iOS thì cần học cả Swift, Objective C. Tuy nhiên việc này tốn khá nhiều thời gian. Mặt khác khả



Hình 4.15: React Native

năng sử dụng lại đóng vai trò quan trọng trong lập trình phần mềm và React Native đáp ứng tốt điều đó. Ở đây tôi dùng React Native để code và build trên Android Studio để tạo ra 1 file apk có thể cài đặt ở nhiều thiết bị. Ngoài ra, React Native cũng là một nguồn mở nên có một cộng đồng hỗ trợ rất lớn. Một số ứng dụng đã được viết bằng react-native như facebook, skype, Instagram, Walmart, yeti smarthome

4.4.2. Redux

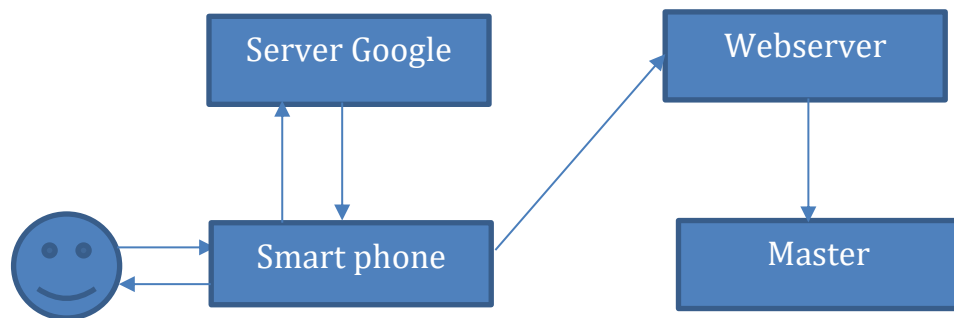
Redux là một thư viện quan trọng dùng để quản lý các state của ứng dụng. Được dựa trên nền tảng kiến trúc Flux do Facebook giới thiệu, do vậy Redux và React Native là bộ đôi hoàn hảo. Việc kiểm soát các state của ứng dụng giúp việc kiểm soát lỗi dễ dàng hơn đồng thời đồng bộ tất cả các state của dữ liệu của app lại với nhau. Redux giúp cho tất cả các dữ liệu được lưu vào một nguồn duy nhất và chúng chỉ được phép đọc. Việc thay đổi dữ liệu chỉ xảy ra khi có sự kiện chứ không được thay đổi một cách trực tiếp. Ở trong App Android này, redux sẽ đóng vai trò là một kho chứa tất cả dữ liệu gồm địa chỉ IP, port của server, trạng thái của các thiết bị trong mạng. Như thế lúc đồng bộ với server tôi chỉ cần đồng bộ redux với server.



Hình 4.16: Redux

4.4.3. AI

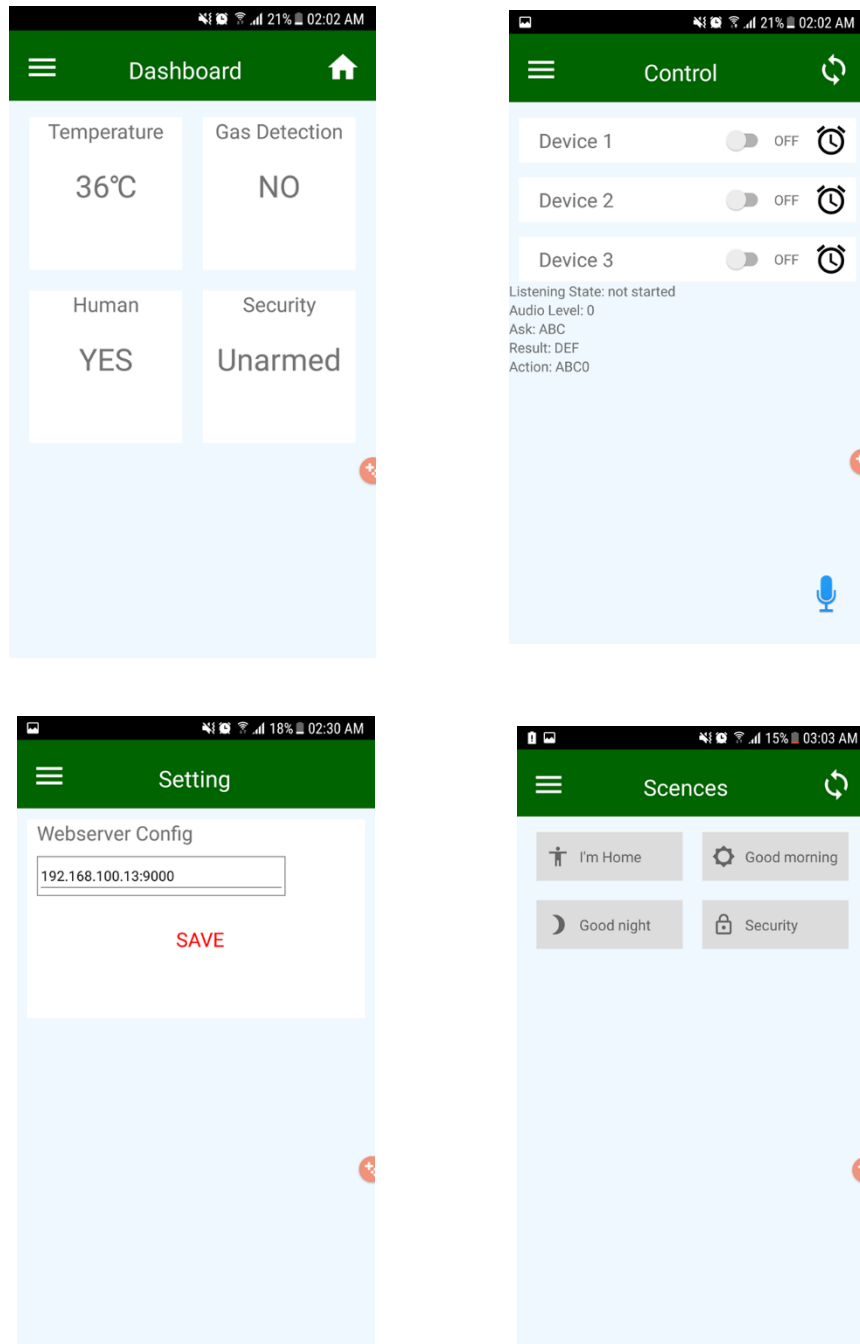
Những thiết bị tích hợp AI như Alexa trong Amazon Echo hay Siri của Apple giúp con người có thể tương tác và điều khiển thiết bị của mình thông qua giọng nói. Trong luận văn tôi suy nghĩ tại sao không tự tạo riêng cho mình một con AI để nó có thể giúp điều khiển các thiết bị thông qua giọng nói trên smartphone. Tôi đặt tên cho nó là Geni. Ở đây tôi dùng mã nguồn mở Api.ai của google để tạo ra Geni. Từ việc thu âm giọng nói trên Smartphone tôi truy cập và gửi dữ liệu lên server của google để so sánh và đưa ra action từ đó đồng bộ với webserver để điều khiển thiết bị.



Hình 4.17: Sơ đồ hoạt động của AI

Ví dụ: Khi tôi nói: Turn on device 1. Geni sẽ ghi âm và gửi lên database của google để so sánh. Sau đó nó trả về câu trả lời: Ok boss, I turned it on for you đồng thời gửi về action: turnOnDevice1. Dựa vào đó với React Native tôi sẽ phân tích và gửi nó lên Webserver để bật thiết bị 1. Ngoài ra khi hỏi về thời tiết: Weather in Ho Chi Minh city. Geni sẽ trả về: The weather looks great kèm theo nhiệt độ cũng như tình trạng thời tiết ở thành phố Hồ Chí Minh.

4.4.4. Một số hình ảnh của App Android



Hình 4.18: Một số hình ảnh của App Android

4.5. Hiệu chỉnh phép đo các thông số dòng điện

Trong việc chế tạo hay sử dụng một thiết bị đo, hiệu chỉnh là giai đoạn không thể thiếu để làm nên một thiết bị đo chính xác. Để đạt được độ chính xác, ADE7753 ngoài việc tích hợp bộ chuyển đổi analog sang số (ADC) chính xác cao, sử dụng điện áp tham chiếu Uref được tự động bù bằng phần mềm nhằm đưa ra được kết quả đảm bảo độ chính xác theo yêu cầu.

Giá trị đọc về từ những thanh ghi của EMIC ADE7753 là những giá trị thập phân. Mỗi đơn vị của số thập phân này gọi là một LSB. Mục tiêu của việc hiệu chỉnh là chuyển những giá trị trong thanh ghi của EMIC thành những giá trị thực có nghĩa và bù những sai số do nhiễu gây nên.

Có 2 phương pháp để hiệu chỉnh thông số thiết bị: thứ nhất là sử dụng một đồng hồ chuẩn, thứ hai là sử dụng một nguồn chuẩn. Do điều kiện phòng thí nghiệm ICA chỉ có máy biến áp (0V - 250V) và đồng hồ AMPROBE 34XR-A và đồng hồ MFM383A, nhóm đề tài chọn phương pháp hiệu chỉnh thông số đo lường bằng đồng hồ chuẩn.

Trong phạm vi đề tài, Slave cần đo 4 thông số là dòng điện, điện áp và công suất. Vì vậy, tôi thực hiện một số thí nghiệm để hiệu chỉnh các thông số trên.

4.5.1. Hiệu chỉnh điện áp

B1: Điều chỉnh giá trị V từ 0 – 250V, đọc giá trị thanh ghi VRMS và giá trị thực đo được từ đồng hồ

Có 2 cách hiệu chỉnh:

Cách 1:

Thông thường người ta hiệu chỉnh bằng cách tính sai số tuyệt đối

$$V_{RMSOS} = \left| \frac{V_1 \times VRMS_2 - V_2 \times VRMS_1}{V_2 - V_1} \right| \quad (4.1)$$

Chương 4: Thiết kế hệ thống

Sau đó tính hệ số điện áp K_v :

$$K_v = \frac{220}{V_{RMS} - V_{RMSOS}} \quad (4.2)$$

Công thức tính toán giá trị điện áp thực:

$$V = (V_{RMS} - V_{RMSOS}) * K_v \quad (4.3)$$

Cách 2: Dùng phương pháp xấp xỉ hàm

Dựa trên bảng số liệu dùng Matlab tính xấp xỉ hàm ra ta sẽ được 1 hàm số dạng $V = f(V_{RMS})$ với V là giá trị thực và V_{RMS} là giá trị đọc được từ thanh ghi. Làm tương tự với I và P .



Hình 4.19: Bố trí thí nghiệm để đo đạc

Chương 5. KẾT QUẢ THỰC HIỆN

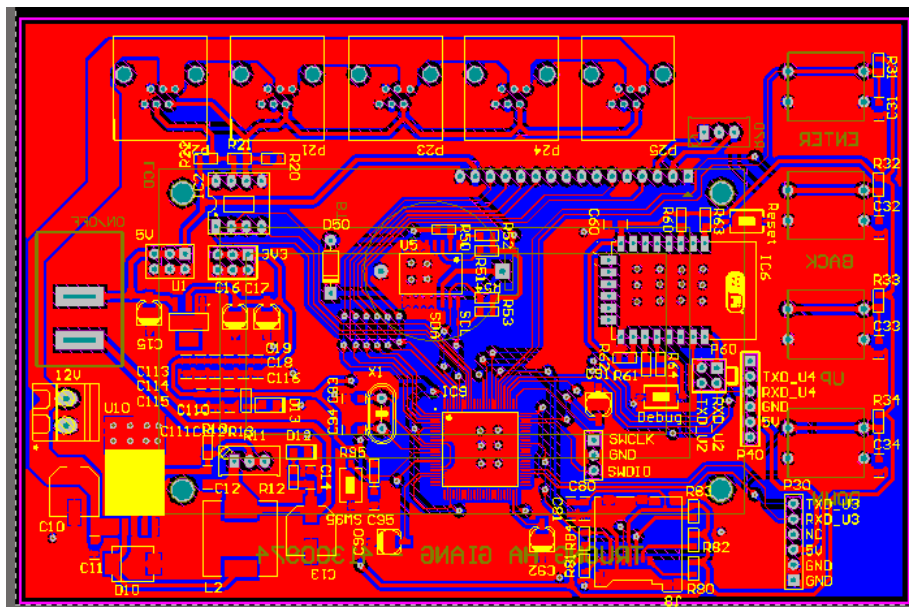
5.1. Tổng thể mô hình



Hình 5.1: Sơ đồ tổng thể mô hình

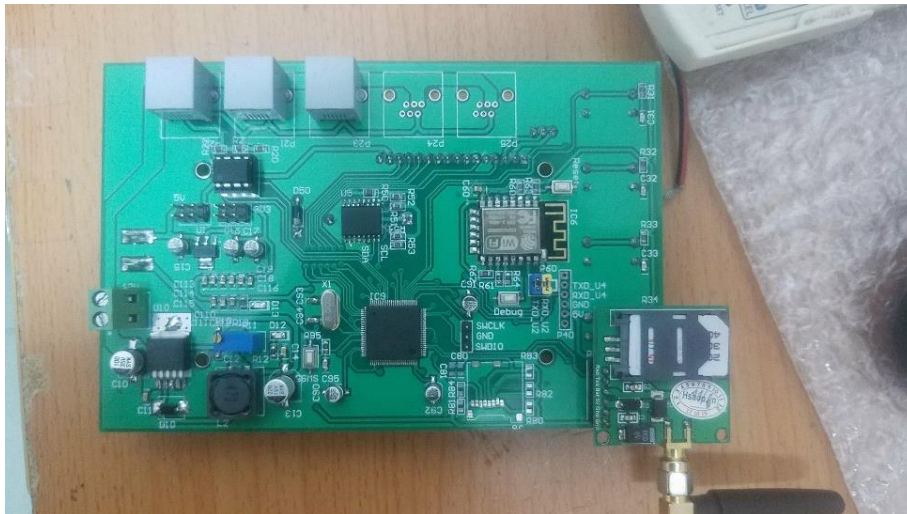
5.2. Khối Master

Layout:



Hình 5.2: Layout khối master

Mạch hoàn chỉnh:

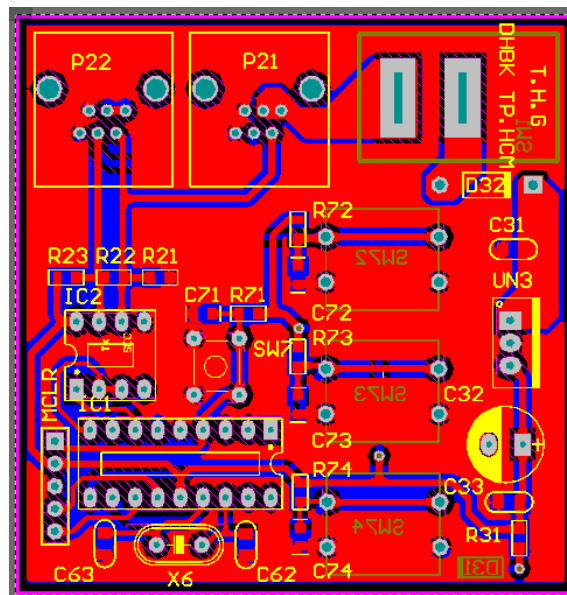


Hình 5.3: Mặt sau khối master

5.3. Khối Slave

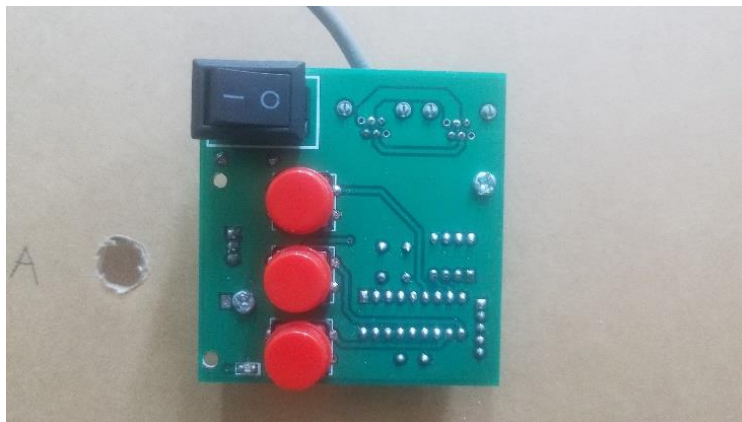
5.3.1. Khối Slave Button:

Layout:



Hình 5.4: Layout khối Slave Button

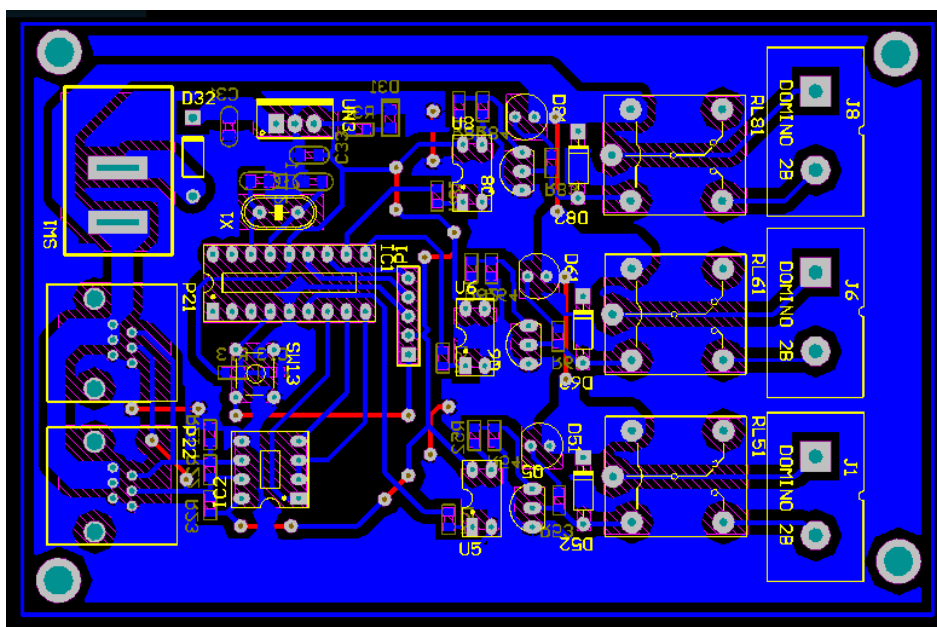
Mạch hoàn chỉnh:



Hình 5.5: Mạch khối Slave Button

5.3.2. Khối Slave Device:

Layout:



Hình 5.6: Layout khối Slave Device

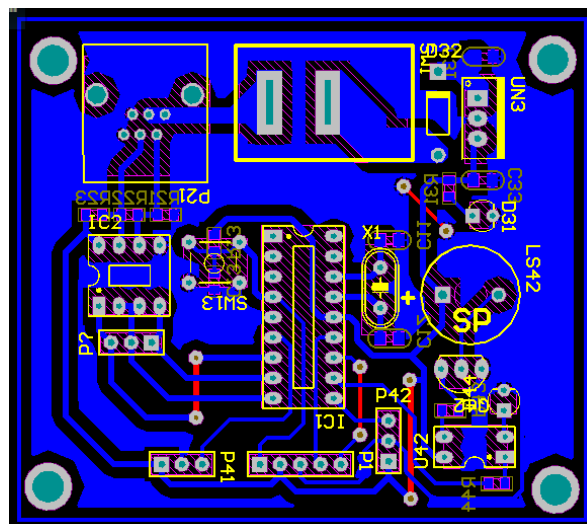
Mạch hoàn chỉnh:



Hình 5.7: Mạch khối Slave Device

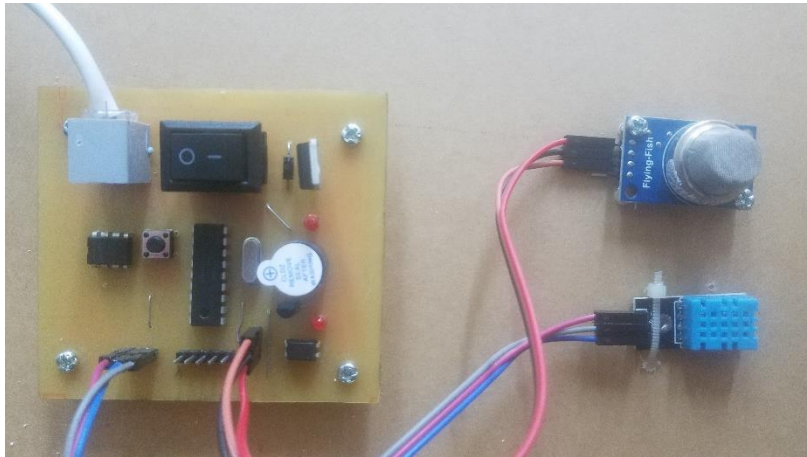
5.3.3. Khối Slave Sensor:

Layout:



Hình 5.8: Layout khối Slave Sensor

Mạch hoàn chỉnh:



Hình 5.9: Mạch khối Slave Sensor

5.4. Bảng đo các thông số điện năng



Hình 5.10: Thí nghiệm đo đặc các thông số

Chương 5: Kết quả thực hiện

Bảng 5.1: Thông số kỹ thuật thiết bị

| | |
|--------------------|------------|
| Kiểu pha | 1 pha |
| Điện áp định mức | 220V |
| Dải điện áp | 100 – 250V |
| Dòng điện định mức | 5A |
| Dòng cực đại | 10A |

5.4.1. Đo lường điện áp

Bảng 5.2 Kết quả đo lường điện áp

| Giá trị điện áp đọc về từ ADE7753 (V) | Giá trị điện áp đọc về từ đồng hồ Amprobe (V) | Sai số (%) |
|--|--|------------|
| 39.65 | 40.1 | 1 |
| 58.83 | 60.2 | 2 |
| 78.98 | 80 | 1 |
| 100.49 | 100.4 | 0.8 |
| 120.47 | 120.2 | 0.2 |
| 140.79 | 140.4 | 0.2 |
| 160.6 | 160.4 | 0.1 |
| 180.12 | 180 | 0.6 |
| 200.08 | 200.4 | 0.1 |
| 220 | 220 | 0 |
| 237 | 240 | 1 |

Chương 5: Kết quả thực hiện

Nhận xét: Sai số đo từ 0.2% đến 2% => thỏa yêu cầu

5.4.2. Đo lường dòng điện

Bảng 5.3: Kết quả đo lường dòng điện

| Giá trị dòng điện đọc về từ ADE7753 (A) | Giá trị dòng điện đọc về từ MFM383A (A) | Sai số (%) |
|--|--|------------|
| 0.28 | 0.28 | 0 |
| 0.86 | 0.86 | 0 |
| 1.5 | 1.5 | 0 |
| 2.12 | 2.12 | 0 |
| 3.31 | 3.31 | 0 |
| 3.91 | 3.91 | 0 |
| 4.54 | 4.54 | 0 |

Nhận xét: Sai số dưới 1% => thỏa yêu cầu

5.4.3. Đo lường công suất

Bảng 5.4: Kết quả đo lường công suất khi $\cos\theta \approx 1$

| Điện áp (V) | Công suất (W) | | | Dòng điện (A) | |
|----------------|-------------------|-------------------|---------------|-------------------|-------------------|
| | Đọc từ ADE7753 | Đọc từ MFM383A | Sai số (%) | Đọc từ ADE7753 | Đọc từ MFM383A |
| 110 | 92 | 93 | 1.1 | 0.86 | 0.87 |
| | 163 | 165 | 1.2 | 1.51 | 1.51 |
| | 230.1 | 232 | 0.8 | 2.13 | 2.14 |
| | 300.07 | 300 | 0 | 2.76 | 2.77 |

Chương 5: Kết quả thực hiện

| | | | | | |
|--|-----|-------|-----|------|------|
| | 417 | 415 | 0.5 | 3.94 | 3.93 |
| | 479 | 478.1 | 0.2 | 4.53 | 4.53 |

Bảng 5.5: Kết quả đo lường công suất khi $\cos\theta \neq 1$

| Điện áp (V) | Dòng điện (A) | P _{đo} từ ADE7753 (W) | P _{thực đo} từ MFM383A (W) | $\cos\theta$ | Sai số (%) |
|-------------|---------------|--------------------------------|-------------------------------------|--------------|------------|
| 58.1 | 1.336 | 54.34 | 56 | 0.7 | 3 |
| 58.1 | 1.261 | 54.34 | 55 | 0.75 | 1.2 |
| 58.3 | 1.135 | 52.69 | 53 | 0.8 | 0.6 |
| 58.3 | 0.992 | 47.77 | 49 | 0.85 | 2.5 |
| 58.6 | 0.811 | 42.84 | 43 | 0.9 | 0.37 |

Nhận xét: Sai số trong khoảng 3% => thỏa yêu cầu

Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết quả đạt được

- Mô hình đã chạy thành công như thiết kế trên lý thuyết và có khả năng mở rộng thêm số thiết bị trong hệ thống
- Thiết kế và chạy tốt giao thức truyền thông
- Thiết kế và thi công mạch bằng Altium Designer
- Xây dựng được webserver để điều khiển và giám sát thiết bị cũng như các thông số điện năng của thiết bị
- Xây dựng được database để quản lý số liệu công suất gửi về từ thiết bị
- Xây dựng được App Android trên điện thoại để điều khiển và giám sát thiết bị
- Tích hợp được AI vào App Android để điều khiển được thiết bị bằng giọng nói
- Sử dụng camera để theo dõi tình trạng căn nhà trên webserver

6.2. Hạn chế

Bên cạnh những kết quả đạt được còn một vài điểm hạn chế ảnh hưởng đến chất lượng luận văn:

- Mô hình còn nhỏ chưa thành một hệ thống lớn do giới hạn về cơ sở vật chất
- AI khi tích hợp vào vẫn còn đang ở dạng sơ khai, chưa khai thác nhiều tính năng của AI

6.3. Hướng phát triển

- Truyền Master – Slave không dây qua RF thay vì RS485
- Cải thiện AI cho những chức năng phức tạp hơn
- Tích hợp thêm một số tính năng vào camera như phát hiện và nhận dạng người
- Xây dựng thêm nhiều tính năng vào mô hình smarthome
- Từ những thông số điện năng đó được đưa ra lời khuyên giúp tiết kiệm điện năng dựa trên hành vi của con người
- Tối ưu hóa thiết kế, cải thiện tính thẩm mỹ, độ bền và tin cậy của thiết bị

TÀI LIỆU THAM KHẢO

- [1] Trương Trọng Tiến (2010). Intelligent Home. Luận văn tốt nghiệp ngành Kỹ Thuật Điện, ĐHBK TP.HCM.
- [2] Trần Xuân Tuấn Anh (2016). Thiết kế mô hình I-Home. Luận văn tốt nghiệp ngành Kỹ Thuật Điện, ĐHBK TP.HCM.
- [3] Võ Đình Thịnh (2016). Thiết kế thi công thiết bị giám sát hệ thống điện của tòa nhà từ xa. Luận văn tốt nghiệp ngành Kỹ Thuật Điện, ĐHBK TP.HCM.
- [3] Hein Marais. RS-485/RS-422 Circuit Implementation Guide
- [4] Datasheet ADE 7753 từ Analog Device : <http://www.analog.com/>
- [5] Datasheet Sim900A từ Simcom : <http://www.simcom.eu/>
- [6] Tập lệnh AT Sim 900A từ Simcom : <http://www.simcom.eu/>
- [7] Datasheet C1815 từ Toshiba : <http://www.alldatasheet.com/>

WEBSITE THAM KHẢO

- [8] Tham khảo lập trình webserver:
- <https://www.w3schools.com/>
- <https://nodejs.org/en/>
- <https://www.mongodb.com/>
- [9] Tham khảo lập trình Android:
- <https://facebook.github.io/react-native/>
- <https://stackoverflow.com/>