

MỤC LỤC

MỞ ĐẦU	3
1. Đặt vấn đề	3
2. Tổng quan tình hình nghiên cứu thuộc lĩnh vực đề tài	3
3. Tính cấp thiết của vấn đề nghiên cứu	4
4. Mục tiêu	4
5. Phương pháp nghiên cứu	5
6. Đối tượng và phạm vi nghiên cứu	5
6.1 Hệ thống nhúng Arduino	6
6.2 Ngôn ngữ lập trình server php và framework CodeIgniter v3.2	6
6.3 Framework Ionic Cordova giúp xây dựng ứng dụng di động	6
I. NỘI DUNG NGHIÊN CỨU	8
CHƯƠNG 1: THIẾT KẾ PHẦN CỨNG.....	8
1.1 Bo mạch Arduino mega 2560.....	8
1.2 Các Module cảm biến:.....	14
1.3 Các module khác	23
1.4 Thiết kế hệ thống phần cứng:	28
CHƯƠNG 2: XÂY DỰNG NGUYÊN LÝ HOẠT ĐỘNG	32
2.1 Tầng mạng.....	32
2.2. Tầng xử lý	39
CHƯƠNG 3: LẬP TRÌNH HỆ THỐNG.....	43
3.1 Lập trình thiết bị	43
3.1.3: Xử lý thời gian thực trong Arduino.....	49
3.1.5 Xử lý hệ thống.....	49
3.2 Lập trình server	53
II: KẾT QUẢ NGHIÊN CỨU	64
II.1. Các kết quả nghiên cứu đạt được	64
II.1.1. Về việc Nghiên cứu xây dựng hệ thống nhà thông minh.	64
II.1.2. Về việc Nghiên cứu xây dựng ứng dụng di động:	64
II.1.3. Về kết quả của sản phẩm tại thời điểm báo cáo:.....	65
II.2. Đánh giá về các kết quả của nhóm đề tài	65
II.2.1. Tính nhanh và hiệu quả.....	65
II.2.2. Ý nghĩa của các kết quả:	65
KẾT LUẬN VÀ KIẾN NGHỊ	67

1. Kết luận về các kết quả đã đạt được	67
2. Kiến nghị về các lĩnh vực ứng dụng.....	67
3. Những định hướng nghiên cứu trong tương lai	67
LỜI CẢM ƠN	69
TÀI LIỆU THAM KHẢO	70
PHỤ LỤC 1:	71

MỞ ĐẦU

1. Đặt vấn đề

Khái niệm smarthome đã được nhắc đến rất nhiều trong những năm gần đây, bên cạnh sự phát triển của công nghệ, các thiết bị tự động hóa ngày càng nâng cao chất lượng cuộc sống và giúp tiết kiệm năng lượng cũng như sức lao động của con người. Nhà thông minh là xu thế mới trong kiến trúc nhà ở hiện đại, nhằm đem lại cuộc sống đích thực cho con người sống trong đó.

Hãy tưởng tượng một ngôi nhà trong đó có thể kết nối và tự động hóa các thiết bị điện tử, điện gia dụng, từ đèn chiếu sáng, mô tơ rèm cửa, máy lạnh, cho tới hệ thống giải trí nghe nhìn, an ninh, an toàn và nhiều tiện ích khác. Bạn sẽ dễ dàng kiểm soát hoạt động của ngôi nhà ngay cả khi vắng mặt từ bất cứ đâu với máy tính hay smartphone, giảm thiểu những rủi ro ngoài ý muốn. Những kịch bản tự động hóa hoạt động của các thiết bị theo ngữ cảnh đem lại tiện nghi, thoải mái và an toàn cho cuộc sống của bạn.

Xuất phát từ những nhu cầu đó nhóm sinh viên quyết định thực hiện đề tài nghiên cứu **“Xây dựng hệ thống nhà thông minh smarthome”**.

2. Tổng quan tình hình nghiên cứu thuộc lĩnh vực đề tài

• Thế giới

Nhà thông minh trên thế giới hiện nay đã được nghiên cứu và phát triển khá rộng rãi, thị trường của các smarthome tập trung chủ yếu vào các nước phát triển. Các tập đoàn công nghệ lớn trên thế giới đã cho ra mắt những hệ thống với rất nhiều những tính năng nổi bật. Nhưng bên cạnh đó, chi phí để xây dựng một hệ thống thông minh là khá lớn và không phải đối tượng nào cũng có thể tiếp cận được.

Những sản phẩm nhà thông minh tạo được uy tín trên thị trường như đầu điều chỉnh nhiệt Nest, ổ cắm điện thông minh Belkin WeMo, đèn Hue của Philips và rất nhiều sản phẩm khác tiện cho người dùng đầu tư từng phần, lắp đặt theo từng mô đun.

• Việt Nam

Trước những thay đổi công nghệ trên thế giới, các công ty công nghệ Việt Nam cũng bắt đầu và nghiên cứu và phát triển các sản phẩm nhà thông minh. Các sản phẩm chủ yếu tập trung vào các thiết bị đơn lẻ như ổ cắm thông minh, công tắc

thông minh, đèn thông minh... Phải kể đến các công ty như BKAV, LUMI... với các sản phẩm đã được phân phối trên thị trường. Nhưng bên cạnh đó vẫn còn các yếu tố khiến cho các thiết bị thông minh chưa phổ biến tại Việt Nam, đó là chi phí cho các thiết bị còn đắt đỏ, chưa có khả năng tích hợp thành hệ thống mà là các thiết bị điều khiển đơn lẻ. Hơn nữa các thiết bị phần lớn là do Việt Nam nhập khẩu, lắp ráp trong nước và chỉ phụ trách về mảng lập trình điều khiển.

3. Tính cấp thiết của vấn đề nghiên cứu

Chúng ta vẫn luôn quan tâm đến các hệ thống tự động hóa giúp ích cho con người vào đời sống, chỉ khi có tự động hóa con người mới thực sự phát triển. Việc điều khiển bằng tay mang lại cảm giác bất tiện và thiếu an toàn khi thực hiện, hơn nữa còn gây lãng phí năng lượng do không thể kiểm soát. Trong thời đại tiên tiến, mọi sự vật đều hiểu được suy nghĩ của con người và tuân theo mọi yêu cầu từ phía người sử dụng.

Để giao tiếp được với các thiết bị con người phải thông qua các công cụ, và hiện nay công cụ phổ biến và tiện dụng là smartphone, với một chiếc điện thoại thông minh luôn bên mình con người có thể dễ dàng theo dõi cũng như điều khiển ngôi nhà của mình ở bất kì đâu. Hãy thử hình dung bạn không ở nhà, nhưng vẫn có thể kiểm soát được toàn bộ ngôi nhà của mình, theo dõi tình hình an ninh trong nhà và bật tắt những thiết bị không dùng đến chỉ với một chiếc điện thoại trên tay có kết nối Internet.

4. Mục tiêu

Đề tài nghiên cứu “Xây dựng hệ thống nhà thông minh” nhằm xây dựng một hệ thống giá rẻ cho mọi nhà, với tính mở ai cũng có thể tiếp cận được với chi phí tối thiểu mà vẫn có đầy đủ các tính năng cơ bản và nâng cao của một ngôi nhà smarthome thực thụ. Đề tài này sẽ đưa ra các giải pháp thiết kế từ tổng thể đến chi tiết cho một ngôi nhà với chi phí dưới 2 triệu đồng.

Các chức năng chính thường sử dụng trong nhà thông minh:

1. Điều khiển chiếu sáng (Tắt/bật thiết bị, độ sáng, tối, hẹn giờ...).
2. Điều khiển màn hình, rèm, cửa cổng.
3. Hệ thống An ninh, báo động, báo cháy.
4. Điều khiển Điều hòa, máy lạnh.
5. Chuông báo động.
6. Các tiện ích và ứng dụng khác.

Các thao tác được điều khiển hoàn toàn qua Internet. Toàn bộ phần cứng và phần mềm do nhóm sinh viên nghiên cứu tự thiết kế và lắp đặt.

5. Phương pháp nghiên cứu

- **Khảo sát nhu cầu thực tế:**

Khảo sát xem nhu cầu hiện tại và trong tương lai của vấn đề nghiên cứu như thế nào. Sinh viên thực hiện nhận thấy sự phát triển mạnh mẽ của các thiết bị tự động hóa, luôn gắn liền với cuộc sống hiện tại, và có hướng phát triển khả quan trong tương lai. Trong đó, ứng dụng về điều khiển và theo dõi nhà thông minh trên các thiết bị di động lại đang ngày càng phổ dụng,.

- **Lựa chọn phương pháp:**

Có nhiều hướng đi cho một hệ thống nhà thông minh, mỗi hệ thống phần cứng và phần mềm có những ưu/nhược điểm khác nhau. Qua quá trình nghiên cứu, nhóm sinh viên thực hiện đề tài đã lựa chọn cho mình phương pháp tiếp cận từ các phần cứng mở (Electrics Opensoure Platform) và các ngôn ngữ lập trình có tính mở cao.

- **Lựa chọn công nghệ:**

- Phần cứng: Sử dụng bo mạch Arduino Mega 2560 với ngôn ngữ lập trình nhúng giống C.
- Phần mềm: Để tiết kiệm chi phí và hiệu năng cho hệ thống, nhóm sinh viên nghiên cứu đã quyết định lựa chọn ngôn ngữ PHP với cơ sở dữ liệu MySQL cho phía server, và bộ framework ionic Cordova để xây dựng ứng dụng di động bên phía người dùng (client).

Ngoài ra nhóm sinh viên còn sử dụng Android Studio để build Ứng dụng cho di động.

Xuất phát điểm từ những tìm hiểu và nghiên cứu của bản thân, cùng sự trợ giúp từ nhiều phía, nhóm sinh viên thực hiện bắt tay vào xây dựng ứng dụng điều khiển và theo dõi nhà thông minh cùng với hệ thống phần cứng tương ứng.

6. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

Dựa trên các yêu cầu của hệ thống, nhóm sinh viên đã thực hiện việc nghiên cứu các đối tượng sau:

6.1 Hệ thống nhúng Arduino

Arduino dùng làm trung tâm xử lý của nhiều thiết bị từ đơn giản đến phức tạp: Máy in 3D, Robot (một trong những robot sửa chữa cáp quang dưới đáy biển cũng được sử dụng), Các thiết bị bay không người lái, Game tương tác, điều khiển ánh sáng, kích hoạt chụp ảnh tốc độ cao,...

Arduino sử dụng vi điều khiển AVR: đây là bộ xử lý trung tâm của toàn bộ mạch, mỗi mẫu Arduino khác nhau thì con chip khác nhau, ở đây Mega sử dụng chip điều khiển ATmega256

6.2 Ngôn ngữ lập trình server php và framework CodeIgniter v3.2

CodeIgniter là một nền tảng ứng dụng web (web application framework) nguồn mở được dùng để xây dựng các ứng dụng web động tương tác với PHP. Nó cho phép các nhà phát triển xây dựng một ứng dụng web nhanh hơn - so với việc viết mã hỗn tạp - bằng cách cung cấp 1 bộ thư viện đầy đủ cho các tác vụ thông thường, cũng như cung cấp một mô hình tương tác đơn giản và dễ hiểu cho việc kết nối tới những bộ thư viện đó.

CodeIgniter cũng mang một số các khái niệm đặc thù và các tính năng cơ bản của các mô hình MVC khác như Ruby on Rails:

- Hỗ trợ kết nối và tương tác đa nền tảng cơ sở dữ liệu
- Tương tác với cơ sở dữ liệu thông qua active records
- Session Management (quản lý Session)
- Định dạng và chuẩn hóa form và dữ liệu đầu vào
- Hỗ trợ Caching toàn trang để tăng tốc độ thực thi và giảm tải tối thiểu cho máy chủ.
- Scaffolding (Bị phản đối từ cộng đồng nên đã được lược bỏ trong phiên bản 2.0 trở về sau)
- Hỗ trợ Template Engine hoặc sử dụng chính PHP tags để điều hướng trong Views
- Hỗ trợ Hooks, các lớp ngoại (Class Extensions), và các Plugins

6.3 Framework Ionic Cordova giúp xây dựng ứng dụng di động

Ionic là một framework dùng để phát triển ứng dụng hybrid dựa trên HTML5. Một ứng dụng hybrid là một ứng dụng di động được xây dựng bằng các công nghệ phát triển web như HTML5, CSS, JavaScript và sau đó được đóng gói trong một môi trường hiển

thị nội dung có thể hoạt động trên hầu hết các thiết bị di động giống như trình duyệt web (Native container). Các ứng dụng này sử dụng các thành phần hiển thị nội dung website (các trình duyệt ẩn như UIWebView trên iOS, hay Webview trên Android và các thiết bị khác) để hiển thị các đoạn mã HTML.

Ứng dụng hybrid đem lại nhiều lợi thế như khả năng hiển thị nội dung trên tất cả các thiết bị di động, cũng như tận dụng tối đa các tính năng khác của thiết bị di động như GPS, camera, danh sách liên lạc,... Bên cạnh đó, thời gian và chi phí dùng để tạo nên một ứng dụng hybrid cũng thấp hơn so với các ứng dụng di động thông thường.

Sử dụng Ionic mang lại nhiều lợi ích, trong đó có các lợi thế từ ứng dụng hybrid như:

- Người phát triển có thể sử dụng các kỹ năng từ lập trình web, giúp giảm thiểu thời gian cũng như chi phí đào tạo nói riêng và thời gian cũng như chi phí dành cho dự án nói chung.
 - Chỉ một bộ mã lệnh có thể sử dụng trên nhiều hệ điều hành khác nhau.
 - Khả năng truy cập đến các tính năng của thiết bị và hệ điều hành.
 - Khả năng sử dụng trong chế độ offline.
 - Việc thiết kế giao diện cho các thiết bị có kích cỡ khác nhau trở nên dễ dàng với thiết kế tùy ứng (responsive design) của web.
 - Việc sử dụng AngularJS làm core cũng mang lại lợi thế lớn so với các framework cho ứng dụng hybrid khác.
 - Ionic cung cấp đầy đủ các thành phần trong giao diện người dùng như Pull-to-Restore, Infinite-loader, tabs, ..
 - Một lợi thế khác khi sử dụng Ionic là nó có một cộng đồng người dùng lớn mạnh, giúp cho việc tìm hiểu và hỗ trợ dễ dàng hơn.
- Tuy vậy Ionic cũng có các mặt bất lợi của nó:
- Vẫn còn trong giai đoạn beta, vẫn còn có thể xuất hiện bugs.
 - Các hiệu ứng và khả năng chuyển đổi giữa các trang có hiệu suất chưa cao

I. NỘI DUNG NGHIÊN CỨU

CHƯƠNG 1: THIẾT KẾ PHẦN CỨNG

1.1 Bo mạch Arduino mega 2560

- **Giới thiệu:** Bo mạch Arduino mega 2560 là một nền tảng phần cứng mở (Opensource Platform) sử dụng vi điều khiển AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau. Arduino được sử dụng phổ biến trên thế giới để xây dựng các hệ điều khiển và tự động hóa như robot, máy bay không người lái, máy in 3D... và nhiều lĩnh vực phổ biến khác.

Được giới thiệu vào năm 2005, Những nhà thiết kế của Arduino cố gắng mang đến một phương thức dễ dàng, không tốn kém cho những người yêu thích, sinh viên và giới chuyên nghiệp để tạo ra những thiết bị có khả năng tương tác với môi trường thông qua các cảm biến và các cơ cấu chấp hành. Những ví dụ phổ biến cho những người yêu thích mới bắt đầu bao gồm các robot đơn giản, điều khiển nhiệt độ và phát hiện chuyển động. Đi cùng với nó là một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Arduino bằng ngôn ngữ C hoặc C++.

- **Lịch sử:**

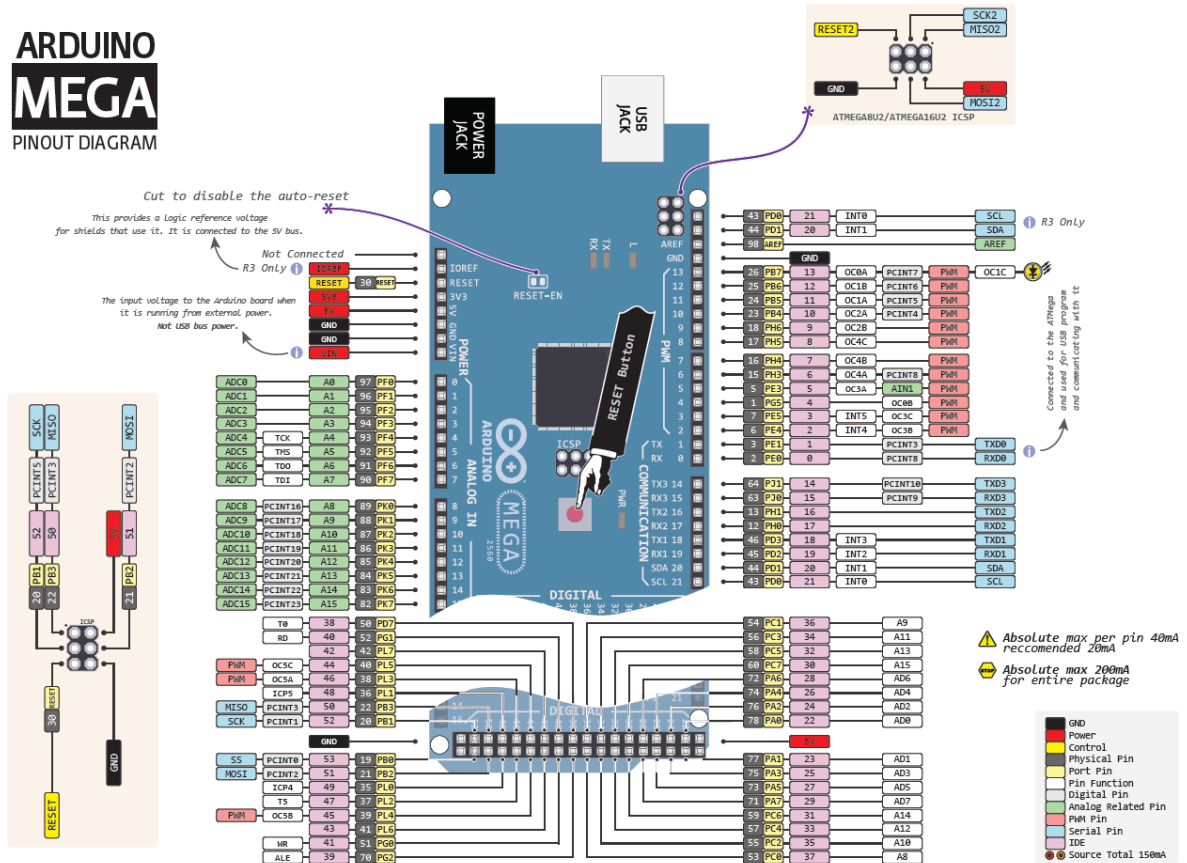
Arduino được khởi động vào năm 2005 như là một dự án dành cho sinh viên tại Interaction Design Institute Ivrea (Viện thiết kế tương tác Ivrea) tại Ivrea, Italy. Vào thời điểm đó các sinh viên sử dụng một "BASIC Stamp" (con tem Cơ Bản) có giá khoảng \$100, xem như giá dành cho sinh viên. Massimo Banzi, một trong những người sáng lập, giảng dạy tại Ivrea. Cái tên "Arduino" đến từ một quán bar tại Ivrea, nơi một vài nhà sáng lập của dự án này thường xuyên gặp mặt. Bản thân quán bar này có được lấy tên là Arduino, Bá tước của Ivrea, và là vua của Italy từ năm 1002 đến 1014.¹

Lý thuyết phần cứng được đóng góp bởi một sinh viên người Colombia tên là Hernando Barragan. Sau khi nền tảng Wiring hoàn thành, các nhà nghiên cứu đã làm việc với nhau để giúp nó nhẹ hơn, rẻ hơn, và khả dụng đối với cộng đồng mã nguồn mở. Trường này cuối cùng bị đóng cửa, vì vậy các nhà nghiên cứu, một trong số đó là David Cuarlierles, đã phổ biến ý tưởng này.

Giá hiện tại của board mạch này dao động xung quanh \$30 và được làm giả đến mức chỉ còn \$9. Một mạch bắt chước đơn giản Arduino Mini Pro có lẽ được xuất phát từ Trung Quốc có giá rẻ hơn \$4, đã trả phí bưu điện.

- **Cấu trúc và nguyên lý hoạt động:**

Sơ đồ chân:



Hình 1.1.1: Sơ đồ chân Arduino Mega 2560

Một mạch Arduino bao gồm một vi điều khiển AVR với nhiều linh kiện bổ sung giúp dễ dàng lập trình và có thể mở rộng với các mạch khác. Một khía cạnh quan trọng của Arduino là các kết nối tiêu chuẩn của nó, cho phép người dùng kết nối với CPU của board với các module thêm vào có thể dễ dàng chuyển đổi, được gọi là *shield*. Vài shield truyền thông với board Arduino trực tiếp thông qua các chân khách nhau, nhưng nhiều shield được định địa chỉ thông qua serial bus I²C-nhiều shield có thể được xếp chồng và sử dụng dưới dạng song song. Arduino chính thức thường sử dụng các dòng chip megaAVR, đặc biệt là ATmega8, ATmega168, ATmega328, ATmega1280, và ATmega2560. Một vài các

bộ vi xử lý khác cũng được sử dụng bởi các mạch Aquino tương thích. Hầu hết các mạch gồm một bộ điều chỉnh tuyến tính 5V và một thạch anh dao động 16 MHz (hoặc bộ cộng hưởng ceramic trong một vài biến thể), mặc dù một vài thiết kế như LilyPad chạy tại 8 MHz và bỏ qua bộ điều chỉnh điện áp onboard do hạn chế về kích cỡ thiết bị. Một vi điều khiển Arduino cũng có thể được lập trình sẵn với một boot loader cho phép đơn giản là upload chương trình vào bộ nhớ flash on-chip, so với các thiết bị khác thường phải cần một bộ nạp bên ngoài. Điều này giúp cho việc sử dụng Arduino được trực tiếp hơn bằng cách cho phép sử dụng 1 máy tính gốc như là một bộ nạp chương trình.

Theo nguyên tắc, khi sử dụng ngăn xếp phần mềm Arduino, tất cả các board được lập trình thông qua một kết nối RS-232, nhưng cách thức thực hiện lại tùy thuộc vào đòi hỏi phần cứng. Các board Serial Arduino có chứa một mạch chuyển đổi giữa RS232 sang TTL. Các board Arduino hiện tại được lập trình thông qua cổng USB, thực hiện thông qua chip chuyển đổi USB-to-serial như là FTDI FT232. Vài biến thể, như Arduino Mini và Boarduino không chính thức, sử dụng một board adapter hoặc cáp nối USB-to-serial có thể tháo rời được, Bluetooth hoặc các phương thức khác. (Khi sử dụng một công cụ lập trình vi điều khiển truyền thống thay vì ArduinoIDE, công cụ lập trình AVR ISP tiêu chuẩn sẽ được sử dụng.)

Board Arduino sẽ đưa ra hầu hết các chân I/O của vi điều khiển để sử dụng cho những mạch ngoài. Diecimila, Duemilanove, và bây giờ là Uno đưa ra 14 chân I/O kỹ thuật số, 6 trong số đó có thể tạo xung PWM (điều chế độ rộng xung) và 6 chân input analog, có thể được sử dụng như là 6 chân I/O số. Những chân này được thiết kế nằm phía trên mặt board, thông qua các header cái 0.10-inch (2.5 mm). Nhiều shield ứng dụng plug-in cũng được thương mại hóa. Các board Arduino Nano, và Arduino-compatible Bare Bones Board và Boarduino có thể cung cấp các chân header đực ở mặt trên của board dùng để cắm vào các breadboard.

Có nhiều biến thể như Arduino-compatible và Arduino-derived. Một vài trong số đó có chức năng tương đương với Arduino và có thể sử dụng để thay thế qua lại. Nhiều mở rộng cho Arduino được thực hiện bằng cách thêm vào các driver đầu ra, thường sử dụng trong các trường học để đơn giản hóa các cấu trúc của các 'con rệp' và các robot nhỏ. Những board khác thường tương đương về điện nhưng có thay đổi về hình dạng-đôi khi còn duy trì độ tương thích với các shield, đôi khi không. Vài biến thể sử dụng bộ vi xử lý hoàn toàn khác biệt, với các mức độ tương thích khác nhau.

Mạch Arduino bao gồm:

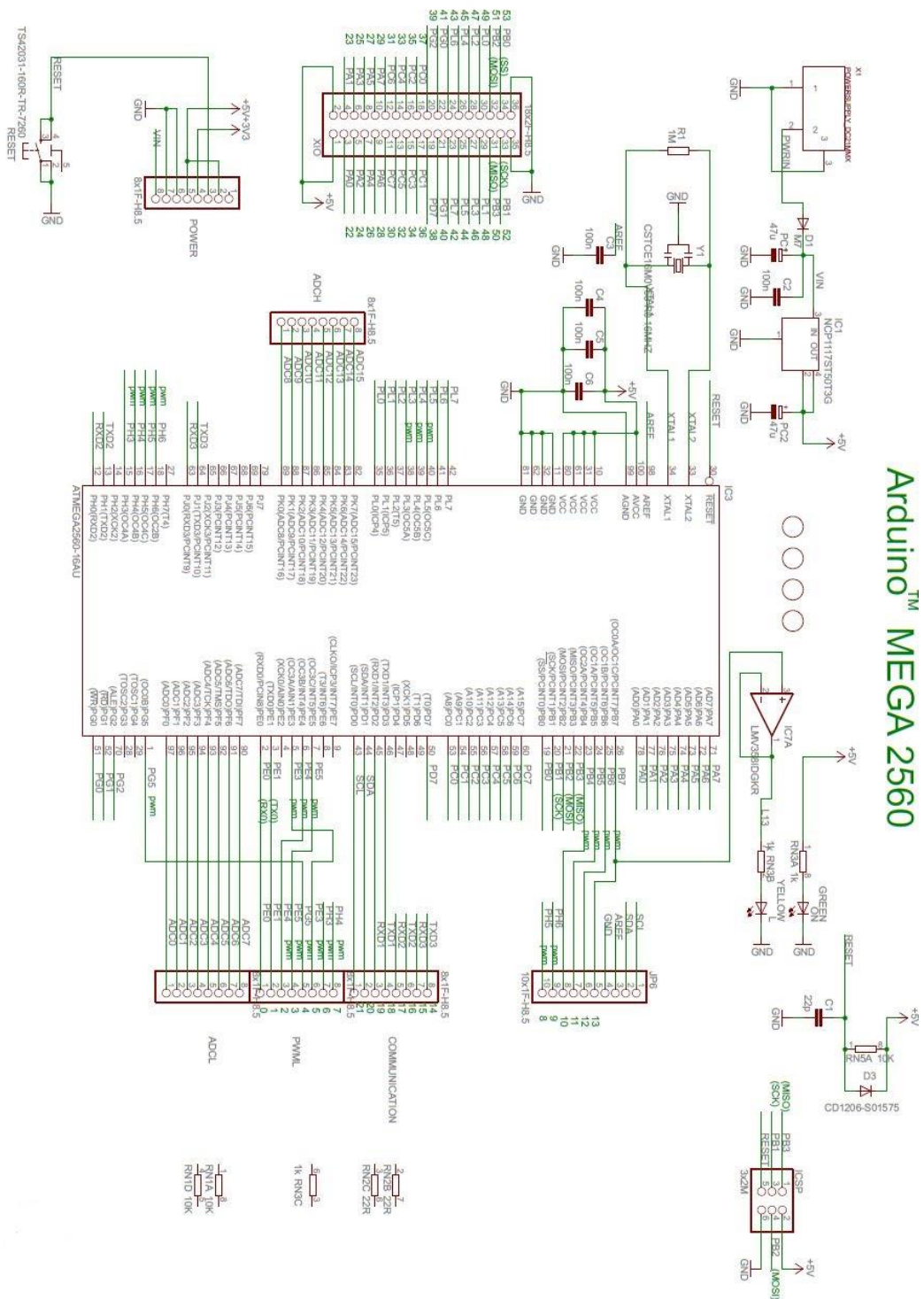
- 54 chân digital (15 có thể được sử dụng như các chân PWM)
- 16 đầu vào analog,
- 4 UARTs (cổng nối tiếp phân cứng),
- 1 thạch anh 16 MHz,
- 1 cổng kết nối USB,
- 1 jack cắm điện,
- 1 đầu ICSP,
- 1 nút reset.

Nó **chứa tất cả mọi thứ** cần thiết để hỗ trợ các vi điều khiển.

Arduino Mega2560 khác với tất cả các vi xử lý trước giờ vì không sử dụng FTDI chip điều khiển chuyển tín hiệu từ USB để xử lý. Thay vào đó, nó sử dụng ATmega16U2 lập trình như là một công cụ chuyển đổi tín hiệu từ USB. Ngoài ra, Arduino Mega2560 **cơ bản vẫn giống Arduino Uno R3**, chỉ khác số lượng chân và nhiều tính năng mạnh mẽ hơn, nên các bạn vẫn có thể lập trình cho con vi điều khiển này bằng chương trình lập trình cho Arduino Uno R3.

Hình 1.1.2: Hình ảnh bo mạch Arduino 2560

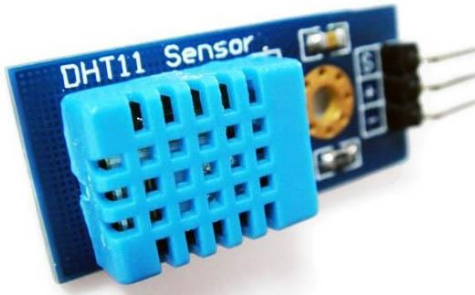
Datasheet của Arduino:



Hình 1.1.4: Datasheet Arduino Mega 2560

1.2 Các Module cảm biến:

1.2.1. Module cảm biến nhiệt độ:



Hình 1.2.1: Module cảm biến Nhiệt độ - độ ẩm DHT11

DHT11

- Giới thiệu:**

DHT11 là cảm biến nhiệt độ và độ ẩm. Nó ra đời sau và được sử dụng thay thế cho dòng SHT1x ở những nơi không cần độ chính xác cao về nhiệt độ và độ ẩm.

DHT11 có cấu tạo 4 chân như hình. Nó sử dụng giao tiếp số theo chuẩn 1 dây.

- Thông số kỹ thuật:

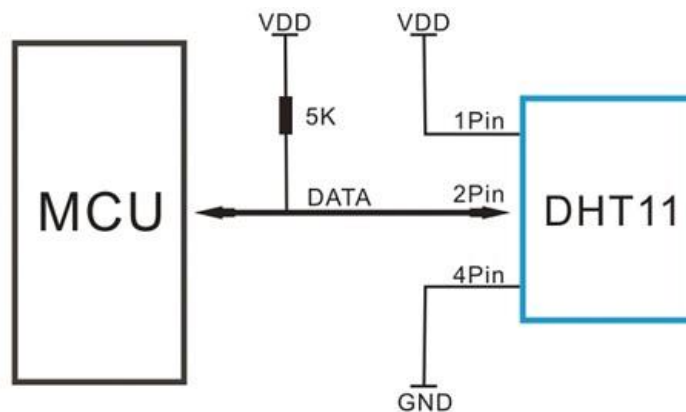
- o Do độ ẩm: 20%-95%

- o Nhiệt độ: 0-50°C

- o Sai số độ ẩm $\pm 5\%$

- o Sai số nhiệt độ: $\pm 2^\circ\text{C}$

Sơ đồ kết nối vi xử lý:



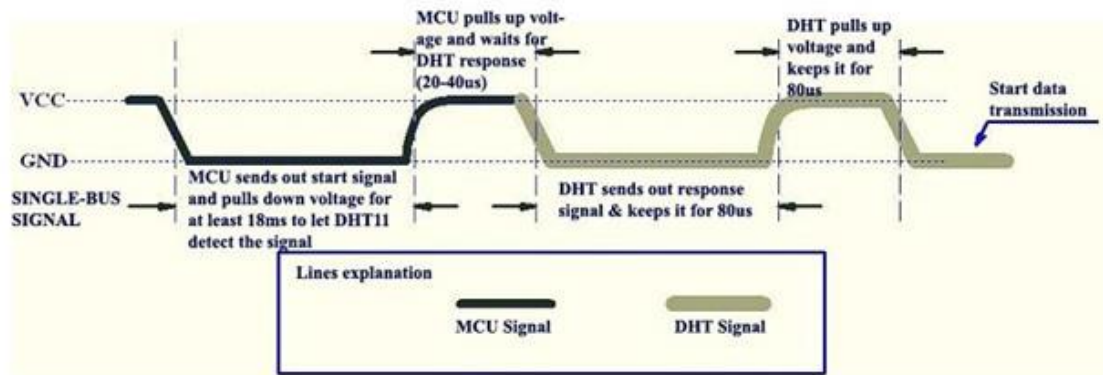
Hình 1.2.2: Sơ đồ kết nối cảm biến DHT11

- **Nguyên lý hoạt động:**

Để có thể giao tiếp với DHT11 theo chuẩn 1 chân vi xử lý thực hiện theo 2 bước:

- o Gửi tín hiệu muốn đo (Start) tới DHT11, sau đó DHT11 xác nhận lại.
- o Khi đã giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.

- Bước 1: gửi tín hiệu Start



- o MCU thiết lập chân DATA là Output, kéo chân DATA xuống 0 trong khoảng thời gian >18ms. Trong Code mình để 25ms. Khi đó DHT11 sẽ hiểu MCU muốn đo giá trị nhiệt độ và độ ẩm.

- o MCU đưa chân DATA lên 1, sau đó thiết lập lại là chân đầu vào.

- o Sau khoảng 20-40µs, DHT11 sẽ kéo chân DATA xuống thấp. Nếu >40µs mà chân DATA không được kéo xuống thấp nghĩa là không giao tiếp được với DHT11.

- o Chân DATA sẽ ở mức thấp 80µs sau đó nó được DHT11 kéo lên cao trong 80µs. Bằng việc giám sát chân DATA, MCU có thể biết được có giao tiếp được với DHT11 không. Nếu tín hiệu đo được DHT11 lên cao, khi đó hoàn thiện quá trình giao tiếp của MCU với DHT.

- Bước 2: đọc giá trị trên DHT11

- o DHT11 sẽ trả giá trị nhiệt độ và độ ẩm về dưới dạng 5 byte. Trong đó:

- § Byte 1: giá trị phần nguyên của độ ẩm (RH%)

- § Byte 2: giá trị phần thập phân của độ ẩm (RH%)

- § Byte 3: giá trị phần nguyên của nhiệt độ (TC)

- § Byte 4 : giá trị phần thập phân của nhiệt độ (TC)

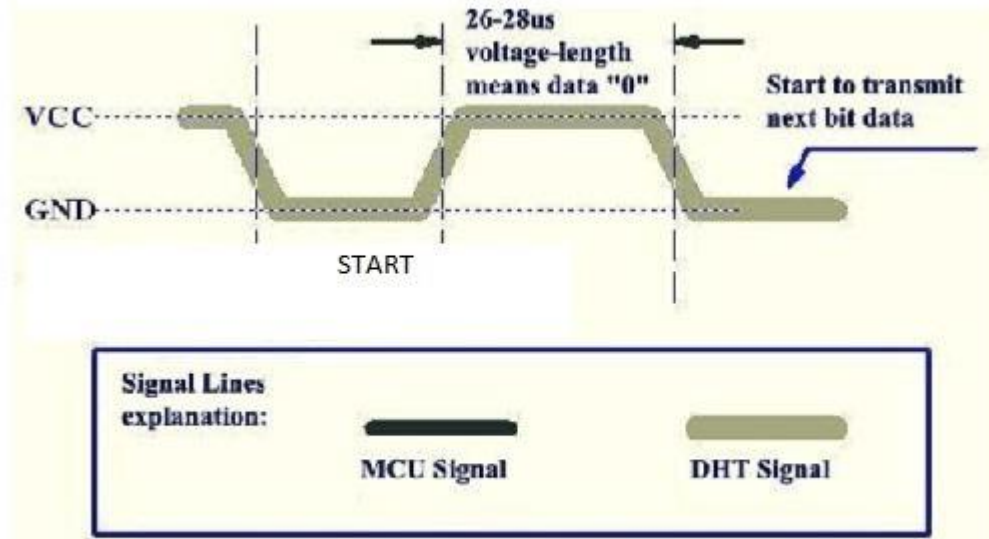
- § Byte 5 : kiểm tra tổng.

• Nếu Byte 5 = (8 bit) (Byte1 + Byte2 + Byte3 + Byte4) thì giá trị độ ẩm và nhiệt độ là chính xác, nếu sai thì kết quả đo không có nghĩa.

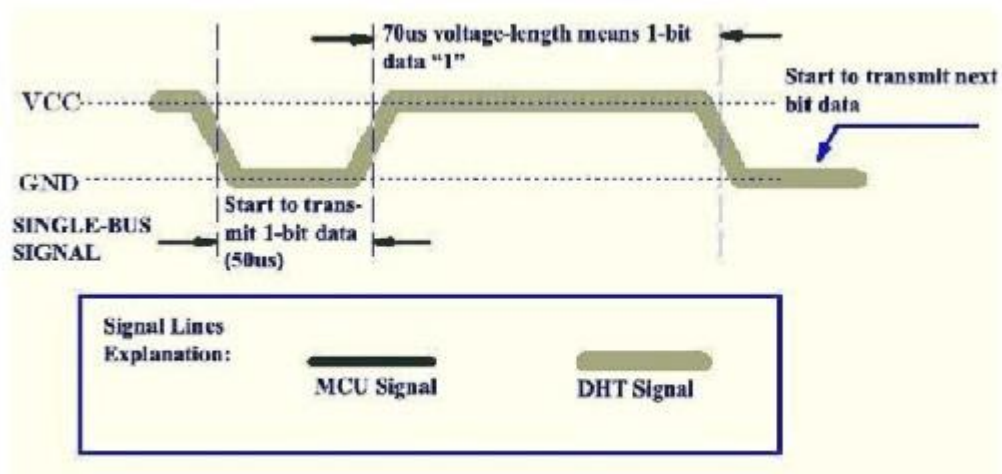
- o Đọc dữ liệu:

Sau khi giao tiếp được với DHT11, DHT11 sẽ gửi liên tiếp 40 bit 0 hoặc 1 về MCU, tương ứng chia thành 5 byte kết quả của Nhiệt độ và độ ẩm.

§ Bit 0:

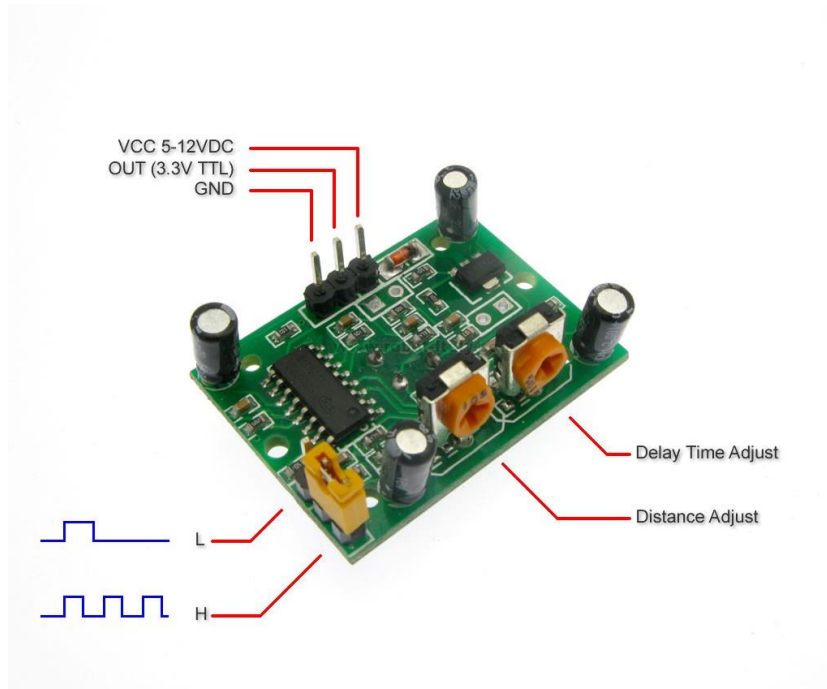


§ Bit 1:



Sau khi tín hiệu được đưa về 0, ta đợi chân DATA của MCU được DHT11 kéo lên 1. Nếu chân DATA là 1 trong khoảng 26-28 us thì là 0, còn nếu tồn tại 70us là 1. Do đó trong lập trình ta bắt sườn lên của chân DATA, sau đó delay 50us. Nếu giá trị đo được là 0 thì ta đọc được bit 0, nếu giá trị đo được là 1 thì giá trị đo được là 1.

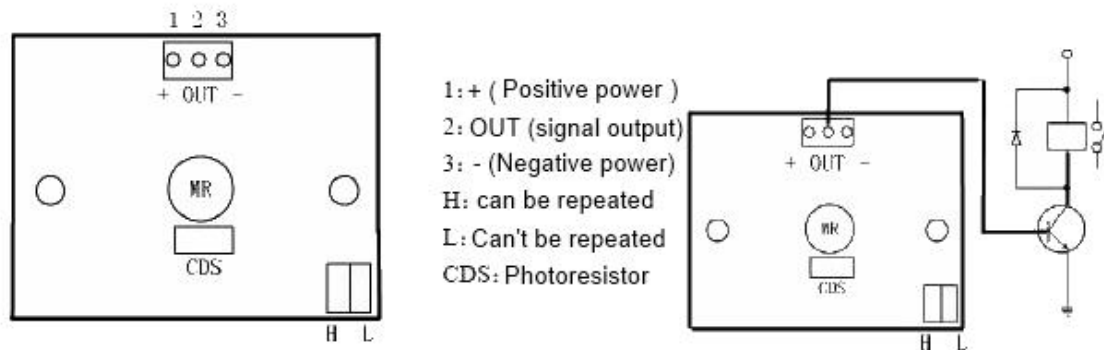
1.2.2 Module cảm biến chuyển động



Hình 1.2.3: Module cảm biến HC-SR501

Thông số kỹ thuật:

- Điện áp hoạt động: 5VCD
- Dòng nghỉ: < 50uA
- Nhiệt độ hoạt động: -15 -> 70 độ C
- Góc quét: 120o
- Tầm xa: 7m
- Kích thước module: 32mm*24mm
- Đường kính gương Fresnel: 32mm
- Thời gian trễ: điều chỉnh bằng biến trở, từ 5s -> 200s
- Điện áp ra: 0 hoặc 3.3v



Hình 1.2.4: Sơ đồ nối dây PIR

- Chế độ kích hoạt:

- + Kích hoạt L (không lặp lại) Điện áp ra tự động chuyển về 0 khi hết thời gian trễ.
- + Kích hoạt H (có lặp lại) (mặc định) Điện áp ra tự động giữ nguyên 3.3V cho đến khi không còn chuyển động.
- **Tính năng hoạt động:** khi có người đi vào vùng hoạt động của cảm biến, điện áp chân ra tự động đặt lên 3.3V, sau khi người đi ra khỏi vùng hoạt động, sau khi hết khoảng thời gian trễ (có thể đặt bằng biến trở) điện áp ra trở về 0.

+Quang trở (CDS) (tính năng mở rộng, có thể không lắp đặt): khi được lắp đặt, cảm biến sẽ chỉ hoạt động vào ban đêm, không hoạt động vào ban ngày.

* Ứng Dụng Và Cách Sử Dụng :

- Tự động phát hiện khi có người vào trong phạm vi hoạt động của cảm biến chuyển động.
- Đầu ra 1.5-3.3V thì module cảm biến chuyển động phát hiện vật thể.
- Kiểm soát ánh sáng (tùy chọn): Bạn có thể xem vị trí lắp quang trở, khi có quang trở, sẽ thiết lập Module hoạt động ban ngày hoặc ban đêm.
- Làm mạch cảm biến chống trộm, phát hiện vật thể, làm vòi xả nước tự động...

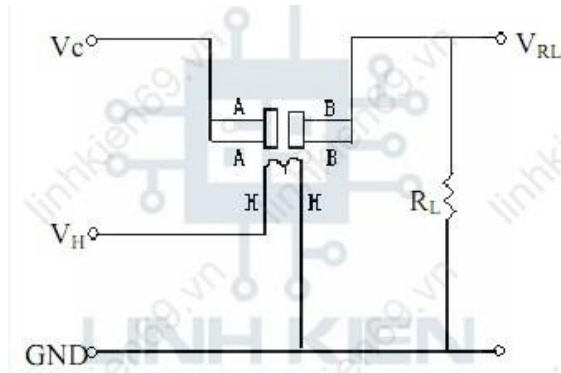
* **Cài đặt:** Khi khởi tạo, Module cần thời gian khởi tạo khoảng 1 phút. Trong thời gian này, module tạo ra điện áp cao từ 1-3 lần sau đó vào chế độ chờ. Điện áp ra 1.5-3.3V, nếu sử dụng I/O 4.5-5.5V bạn nên lắp thêm transistor.

- Nên cố gắng tránh ánh sáng trực tiếp và nguồn nhiễu gần với bề mặt lăng kính của các mô-đun, để tránh đưa ra tín hiệu nhiễu, tránh sử dụng môi trường nhiễu gió

I.2.3 Module cảm biến khí cháy MQ2

- MQ2 là cảm biến khí, dùng để phát hiện các khí có thể gây cháy. Nó được cấu tạo từ chất bán dẫn SnO_2 . Chất này có độ nhạy cảm thấp với không khí sạch. Nhưng khi trong môi trường có chất gây cháy, độ dẫn của nó thay đổi ngay. Chính nhờ đặc điểm này người ta thêm vào mạch đơn giản để biến đổi từ độ nhạy này sang điện áp.
- Khi môi trường sạch điện áp đầu ra của cảm biến thấp, giá trị điện áp đầu ra càng tăng khi nồng độ khí gây cháy xung quanh MQ2 càng cao.
- MQ2 hoạt động rất tốt trong môi trường khí hóa lỏng LPG, H_2 , và các chất khí gây cháy khác. Nó được sử dụng rộng rãi trong công nghiệp và dân dụng do mạch đơn giản và chi phí thấp.

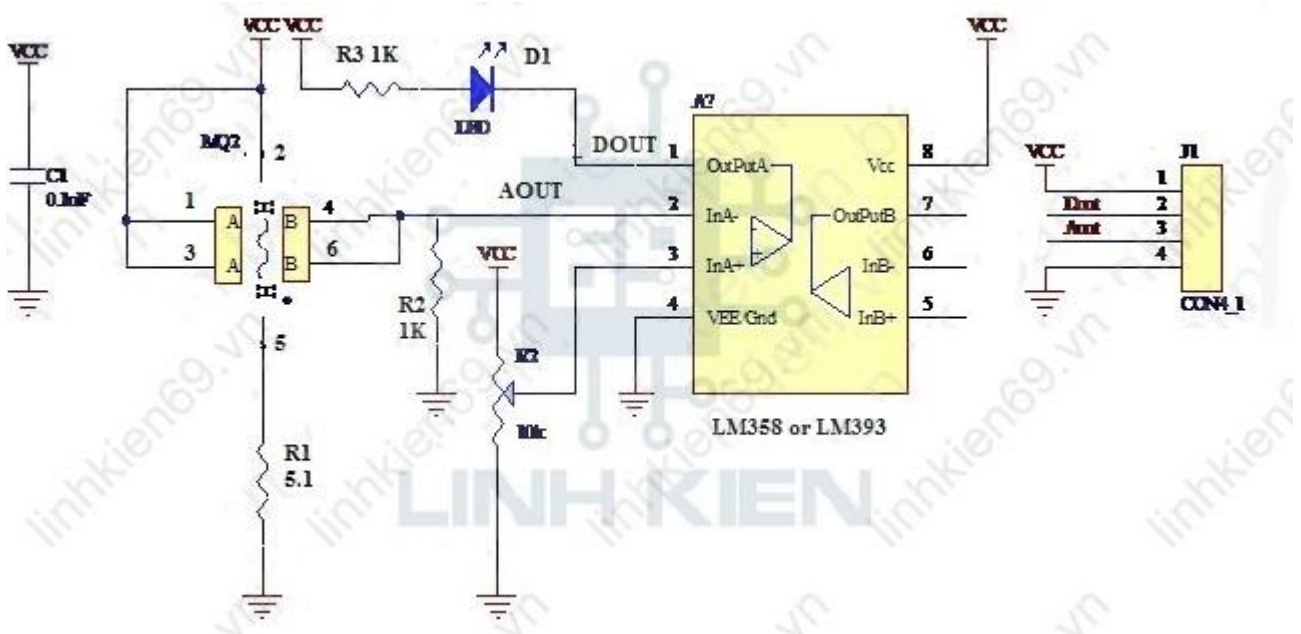
Sơ đồ nguyên lý:



Sơ đồ chân MQ2



- Trong đó:
 - o Chân 1,3 là A
 - o Chân 2,5 là B
 - o Chân 4,6 là C



Hình 1.2.5 sơ đồ nguyên lý cảm biến MQ2

- Trong mạch có 2 chân đầu ra là Aout và Dout. Trong đó:

Aout: điện áp ra tương tự. Nó chạy từ 0.3 đến 4.5V, phụ thuộc vào nồng độ khí xung quanh MQ2.

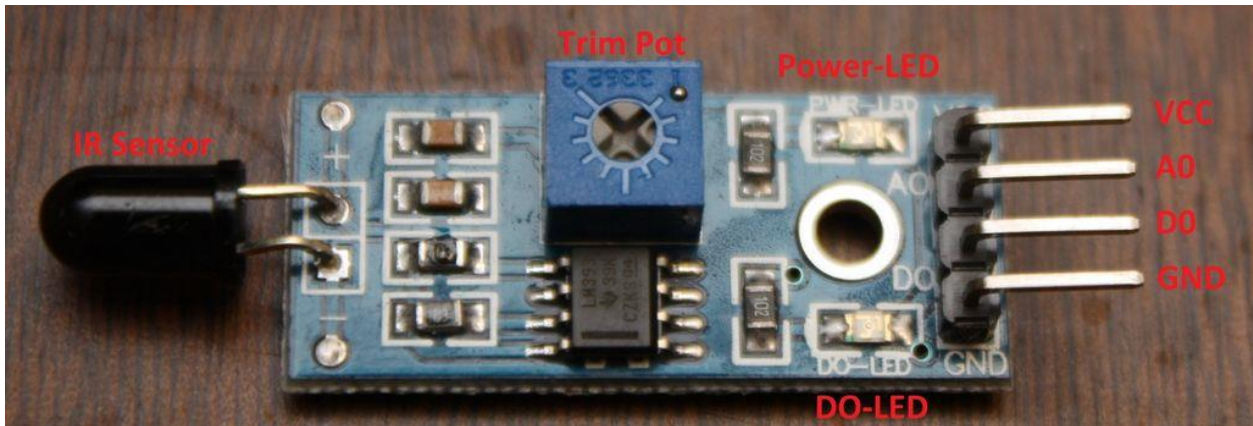
o Dout: điện áp ra số, giá trị 0,1 phụ thuộc vào điện áp tham chiếu và nồng độ khí mà MQ2 đo được.

Việc có chân ra số Dout rất tiện cho ta mắc các ứng dụng đơn giản, không cần đến vi điều khiển. Khi đó ta chỉ cần chỉnh giá trị biến trở tới giá trị nồng độ ta muốn cảnh báo. Khi nồng độ MQ2 đo được thấp hơn mức cho phép thì Dout = 1. Đèn Led tắt. Khi nồng độ khí đo được lớn hơn nồng độ cho phép, Dout = 0, đèn led sáng. Ta có thể ghép nối vào mạch Relay để điều khiển bật tắt đèn, còi, hoặc thiết bị cảnh báo khác.

- Một điều khó khăn khi làm việc với MQ2 là chúng ta khó có thể quy từ điện áp Aout về giá trị nồng độ ppm. Rồi từ đó hiển thị và cảnh báo theo ppm. Do giá trị điện áp trả về từng loại khí khác nhau, lại bị ảnh hưởng nhiệt độ, độ ẩm.

1.2.3. Module cảm biến phát hiện ngọn lửa

- Nguyên lý hoạt động: Về cơ bản Cảm biến có nguyên lý hoạt động tương tự như cảm biến chuyển động HC-SR501 nên chúng ta sẽ không đề cập ở đây.



Hình 1.2.6: Hình ảnh cảm biến ngọn lửa

Các tính năng chính:

- Điện áp hoạt động 5 V
- Sử dụng IC so sánh áp LM393, ngõ ra TTL tích cực mức cao.
- Phát hiện ánh sáng có bước sóng từ 760nm đến 1100nm (ánh sáng hồng ngoại , lửa).
- Góc phát hiện < 60°
- Độ nhạy có thể tùy chỉnh .
- Kích thước 48 x 14 mm

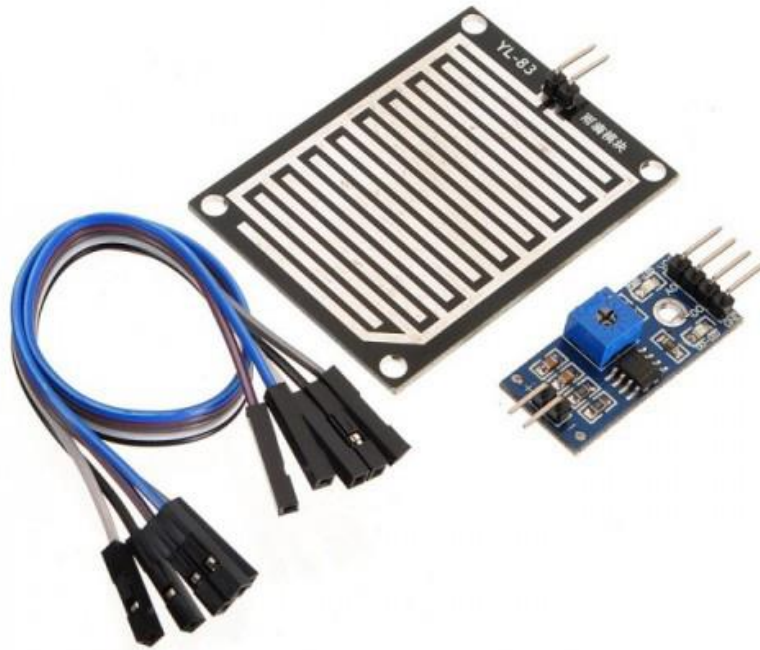
Ứng Dụng:

- Hệ thống báo cháy, tự động điều khiển chống cháy...

Mô tả:

- Module cảm biến phát hiện ngọn lửa 1 kênh được thiết kế đơn giản tối đa mà vẫn mang lại hiệu quả cao.
- Module có thể phát hiện ánh sáng trong vùng hồng ngoại với độ nhạy có thể tùy chỉnh linh hoạt.
- Có khả năng chống cháy thấp, do đó nên được bảo vệ khoảng cách với lửa.

1.2.4 Module cảm Biến Mưa



Hình 1.2.7: Module cảm biến mưa

*** Thông số kỹ thuật :**

- Điện áp: 5V
- Led báo nguồn (Màu xanh)
- Led cảnh báo mưa (Màu đỏ)
- Hoạt động dựa trên nguyên lý: Nước rơi vào board sẽ tạo ra môi trường dẫn điện.
- Có 2 dạng tín hiệu: Analog(AO) và Digital (DO)
- Dạng tín hiệu : TTL, đầu ra 100mA (Có thể sử dụng trực tiếp Relay, Còi công suất nhỏ...)
- Điều chỉnh độ nhạy bằng biến trở.
- Sử dụng LM358 để chuyển AO --> DO

*** Chuẩn đầu ra:** Kết nối với nguồn 5V

- DO: Đầu ra ở mức cao (1), khi có nước đèn đỏ sáng, đồng thời đầu ra về mức thấp (0).

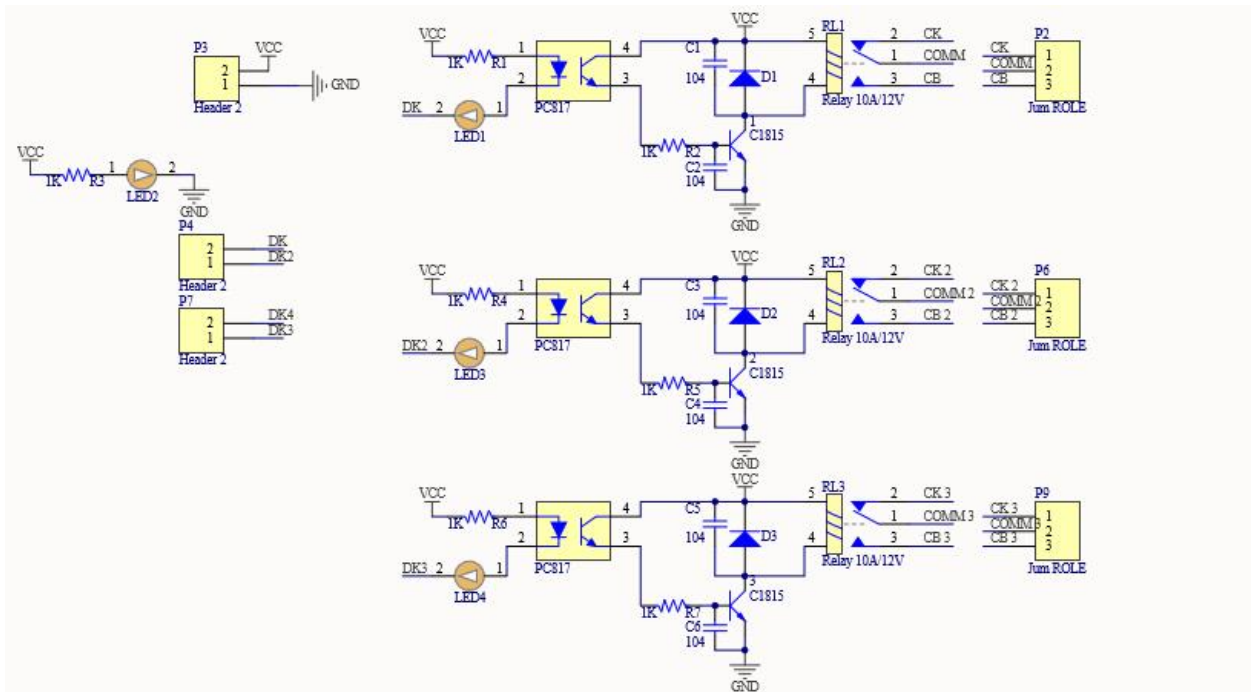
Có thể sử dụng để điều khiển relay, Còi..

Hoặc đưa vào chân I/O của Vi điều khiển.

- AO: Dùng để xác định độ lớn của giọt nước, bằng cách đưa vào ADC của Vi điều khiển.
- Điều chỉnh độ nhạy bằng biến trở.

1.3 Các module khác

1.3.1: Module Relay

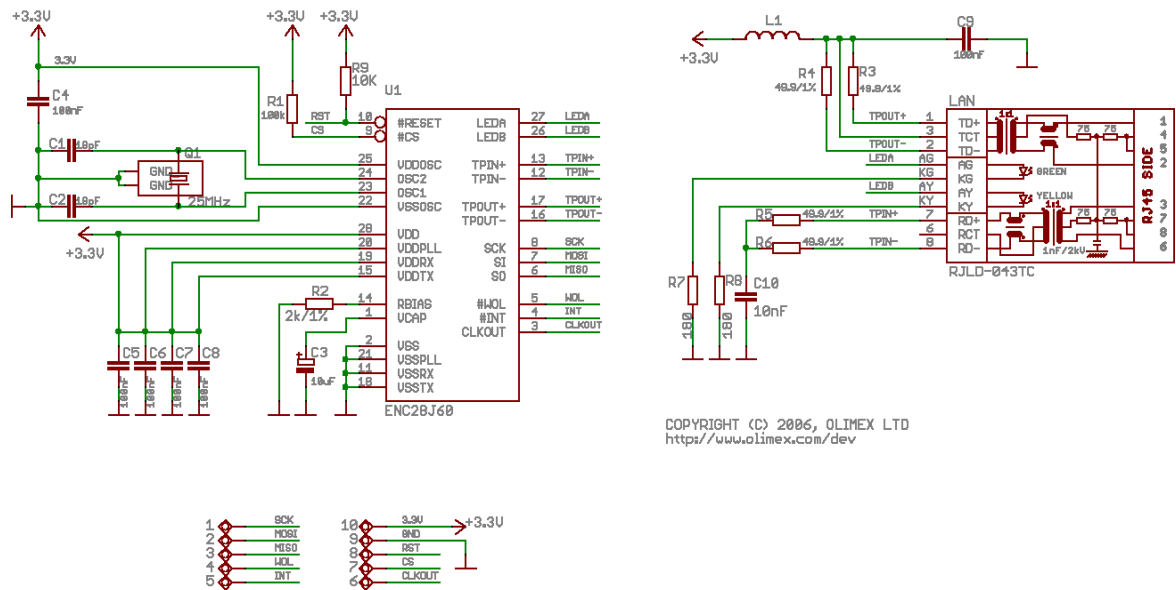


Hình 1.3.1: Sơ đồ nguyên lý mạch Relay

- **Trạng thái:** Module Relay hoạt động ở 2 trạng thái 0 (bật) và 1 (tắt) với điện áp đầu vào 5v lấy trực tiếp từ vi điều khiển.
- **Chức năng:** Module Relay hoạt động như một công tắc đóng mở mạch điện, sử dụng dòng điều khiển cách ly hoàn toàn với nguồn điện thiết bị.
- **Nguyên lý:** Sử dụng PC817 để kích hoạt dòng điện vào relay khi trạng thái đầu vào ở mức thấp.
- **Dòng điện bật tắt định mức:** 220v 10A là khá lớn đối với các thiết bị điện trong gia đình

1.3.2 Module Ethernet Shield ENC 28J60

a. Sơ đồ nguyên lý



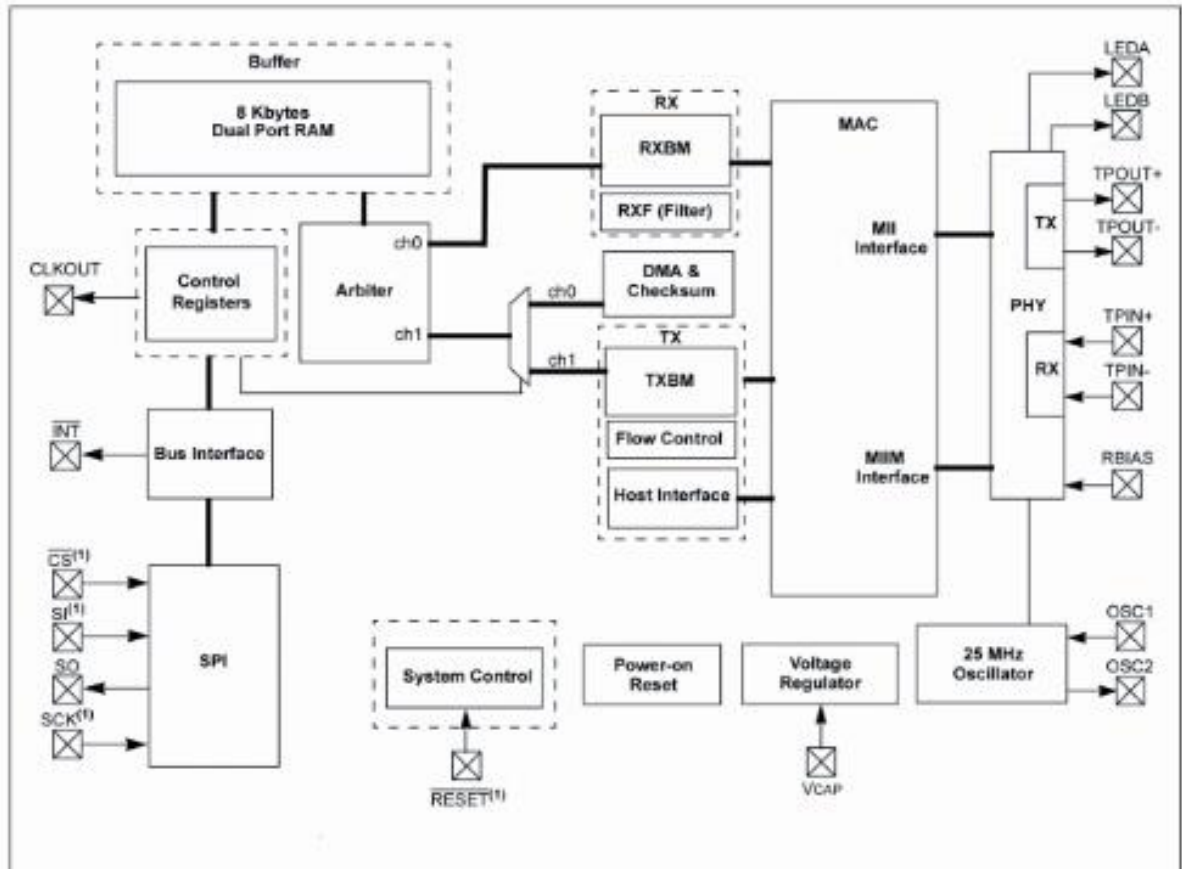
Hình 1.3.2 Sơ đồ nguyên lý Module Ethernet ENC 28J60

Vi Điều Khiển Hỗ Trợ kết nối Ethernet -ENC28J60 :

Vi điều khiển ENC28J60 là vi điều khiển hỗ trợ kết nối Ethernet loại phổ biến nhất hiện nay, được thiết kế chế tạo bởi Microchip.

- Phần cứng của ENC28J60 có tính hợp 2 lớp dưới nhất trong mô hình mạng nói trên (Data link Layer và Physical Player)
- Đi kèm với phần cứng này là gói thư viện hỗ trợ của Microchip-TCP/IP.

b. Các khối phần cứng cơ bản trong ENC28J60



1. MAC modul , phục vụ cho lớp liên kết dữ liệu
2. PHY modul , phục vụ cho việc encode (mã hóa) và decode(giải mã) dữ liệu trong lớp vật lí.
3. SPI interface , đây là modul rất quan trọng. Modul này phục vụ cho việc giao tiếp giữa VĐK master (gọi là HOST) với ENC28J60.
4. Một tập thanh ghi điều khiển (Control Register) , phục vụ điều khiển các khối MAC , PHY
5. Một tập thanh ghi dữ liệu (RAM buffer for transmitted and received data)

Nguyên lý hoạt động của mạch như sau :

ENC28J60

FIGURE 1-2: TYPICAL ENC28J60-BASED INTERFACE

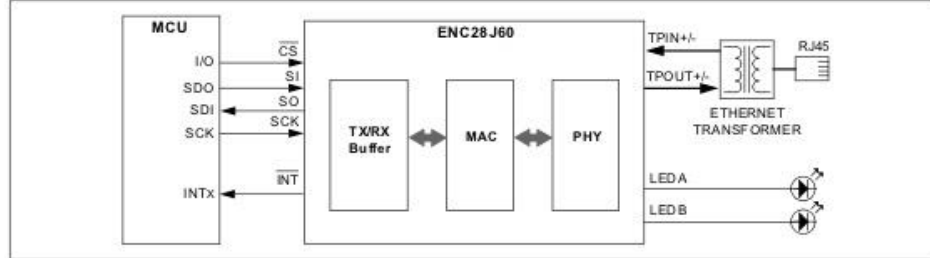


TABLE 1-1: PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	SPDIP, SOIC, SSOP	QFN			
VCAP	1	25	P	—	2.5V output from internal regulator. A low Equivalent Series Resistance (ESR) capacitor, with a typical value of 10 mF and a minimum value of 1 mF to ground, must be placed on this pin.
VSS	2	26	P	—	Ground reference.
CLKOUT	3	27	O	—	Programmable clock output pin. ⁽¹⁾
INT	4	28	O	—	INT interrupt output pin. ⁽²⁾
NC	5	1	O	—	Reserved function; always leave unconnected.
SO	6	2	O	—	Data out pin for SPI interface. ⁽²⁾
SI	7	3	I	ST	Data in pin for SPI interface. ⁽³⁾
SCK	8	4	I	ST	Clock in pin for SPI interface. ⁽³⁾
CS	9	5	I	ST	Chip select input pin for SPI interface. ^(3,4)
RESET	10	6	I	ST	Active-low device Reset input. ^(3,4)
VSSRX	11	7	P	—	Ground reference for PHY RX.
TPIN-	12	8	I	ANA	Differential signal input.
TPIN+	13	9	I	ANA	Differential signal input.
RBIAS	14	10	I	ANA	Bias current pin for PHY. Must be tied to ground via a resistor (refer to Section 2.4 "Magnetics, Termination and Other External Components" for details).
VDDTX	15	11	P	—	Positive supply for PHY TX.
TPOUT-	16	12	O	—	Differential signal output.
TPOUT+	17	13	O	—	Differential signal output.
VSSTX	18	14	P	—	Ground reference for PHY TX.
VDDRX	19	15	P	—	Positive 3.3V supply for PHY RX.
VDDPLL	20	16	P	—	Positive 3.3V supply for PHY PLL.
VSSPLL	21	17	P	—	Ground reference for PHY PLL.
VSSOSC	22	18	P	—	Ground reference for oscillator.
OSC1	23	19	I	ANA	Oscillator input.
OSC2	24	20	O	—	Oscillator output.
VDDOSC	25	21	P	—	Positive 3.3V supply for oscillator.
LEDB	26	22	O	—	LEDB driver pin. ⁽⁵⁾
LEDA	27	23	O	—	LEDA driver pin. ⁽⁵⁾
VDD	28	24	P	—	Positive 3.3V supply.

Legend: I = Input, O = Output, P = Power, ANA = Analog Signal Input, ST = Schmitt Trigger

- Note** 1: Pins have a maximum current capacity of 8 mA.
 2: Pins have a maximum current capacity of 4 mA.
 3: Pins are 5V tolerant.
 4: Pins have an internal weak pull-up to VDD.
 5: Pins have a maximum current capacity of 12 mA.

- Vi điều khiển ENC28J60 được điều khiển hoàn toàn thông qua giao tiếp SPI với Arduino.
- Arduino đóng vai trò Master trong giao tiếp SPI với ENC28J60 Giao tiếp SPI Cổng RJ45

6 . Tương tự như kết nối mạng trên PC , Arduino đóng vai trò PC , còn ENC28J60 đóng vai trò như card mạng. Tín hiệu yêu cầu từ mạng truyền qua cổng RJ45 vào ENC28J60 . ENC28J60 được thiết kế để giải mã tín hiệu và chuyển tín hiệu đó thành dữ liệu và lưu vào bộ đệm thu. Thông qua giao tiếp SPI , Arduino liên tục kiểm tra bộ đệm của ENC28J60. Nếu phát hiện có dữ liệu , nó sẽ đọc dữ liệu về và xử lý.

- Phát dữ liệu : Thông qua giao tiếp SPI , Arduino gửi dữ liệu vào bộ đệm phát của ENC28J60. ENC28J60 sẽ mã hóa dữ liệu và truyền ra đường RJ45 đến địa chỉ mong muốn (được ghi trong khung dữ liệu)

Tập lệnh giao tiếp : ENC28J60 được điều khiển hoàn toàn bằng một vi điều khiển khác đóng vai trò là Host. Host dùng tập lệnh này để điều khiển việc truyền và nhận dữ liệu từ ENC28J60

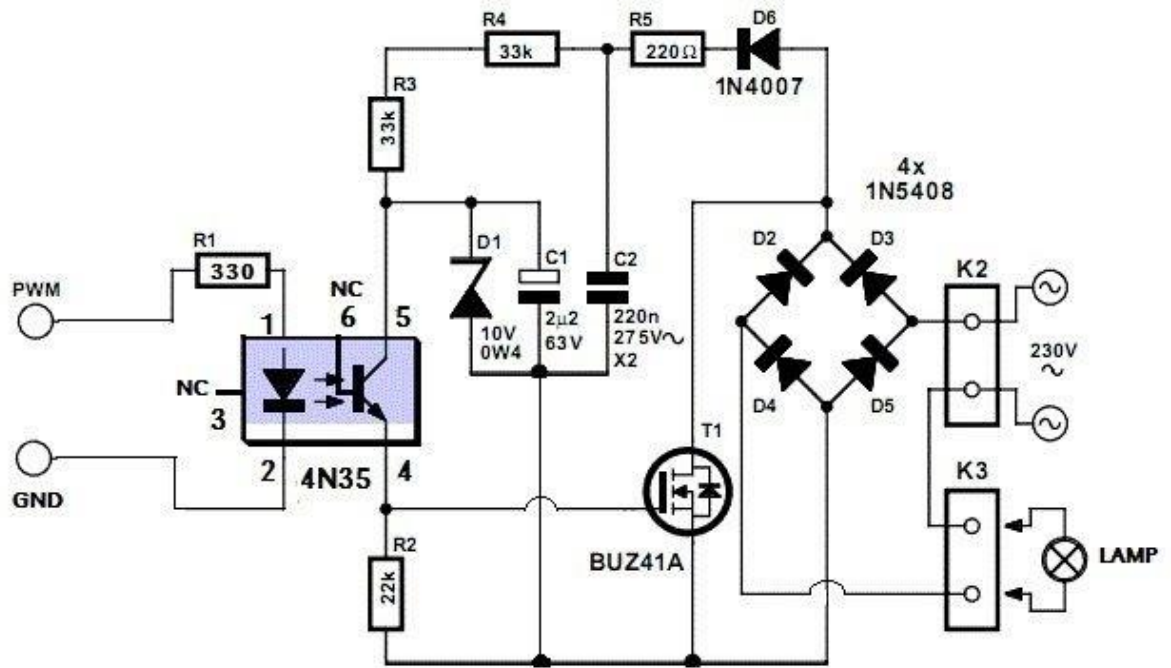
- Tập lệnh chỉ gồm 7 lệnh , được truyền từ Host đến ENC28J60 thông qua đường giao tiếp SPI.

Sơ đồ nối chân:

ENC28J60 module	Arduino Uno/Due	Arduino Mega
CS	D10	D53
SI	D11	D51
SO	D12	D50
SCK	D13	D52
RESET	RESET	RESET
INT	D2	D2
VCC	3V3	3V3
GND	GND	GND

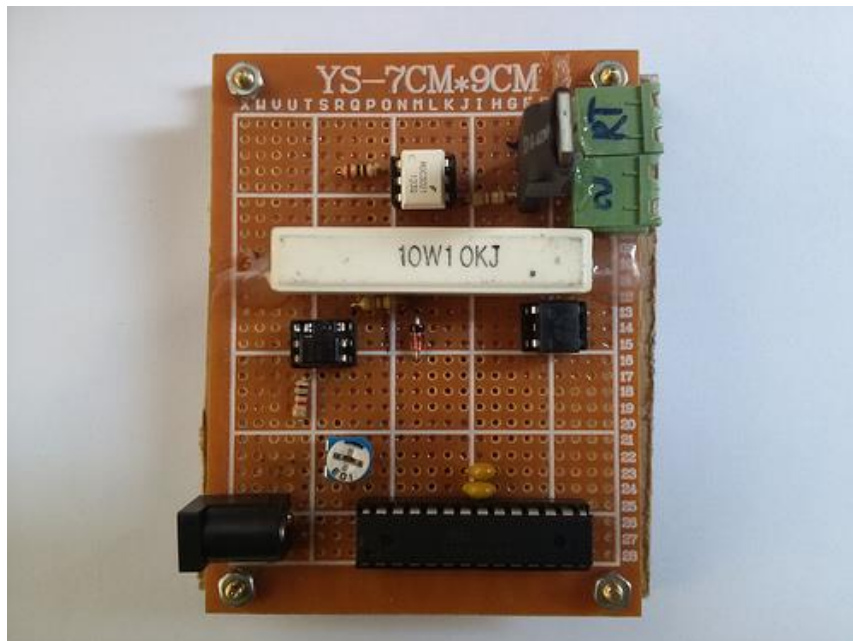
1.3.3. Module điều khiển mức công suất thiết bị

Đây là module do nhóm sinh viên tự nghiên cứu thiết kế và lắp ráp, không bán sẵn trên thị trường



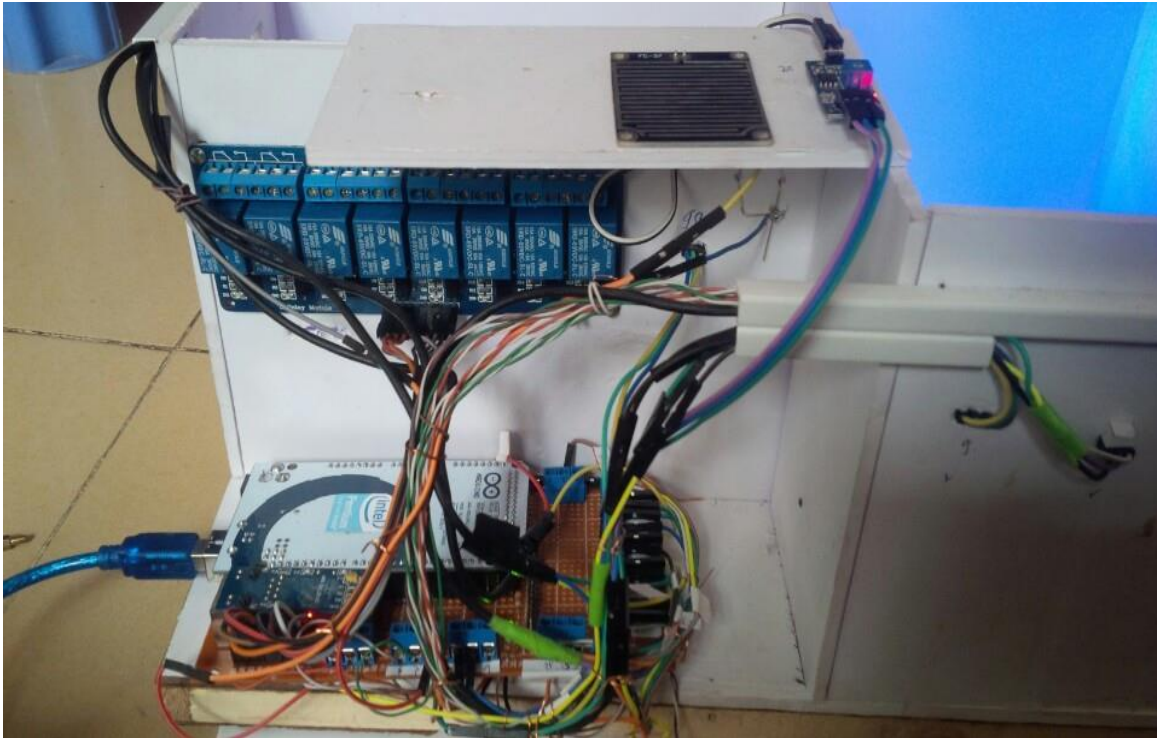
Author: Nguyễn Đình Quý

Sơ đồ nguyên lý mạch điều khiển độ sáng đèn 220V dùng điều chế xung PWM



1.4 Thiết kế hệ thống phần cứng:

Phần cứng được nhóm sinh viên lắp ráp toàn bộ trên 1 bo mạch với 15 chân đầu ra điều khiển thiết bị, 6 chân dữ liệu xuất xung PWM và 6 chân analog cho các tín hiệu dữ liệu dạng tương tự.



Hình ảnh lắp đặt mô hình nhà thông minh

Hệ thống được tích hợp trên 1 bo mạch duy nhất với kích thước chưa thu gọn là 15x10cm (có thể thu gọn hơn nữa) đảm bảo tính nhỏ gọn có thể lắp đặt trực tiếp trên điều kiện thực tế.

Thông số kỹ thuật:

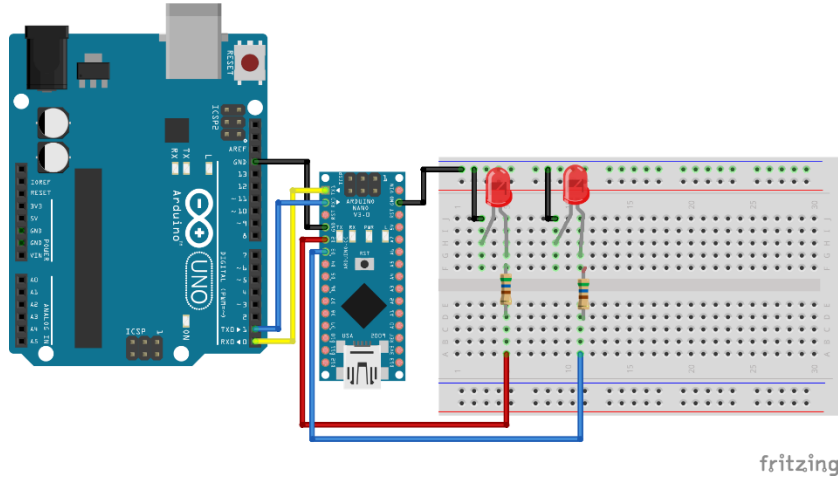
Tên thông số	Giá trị
Điện áp hoạt động	5v qua USB hoặc 9v DC qua Adapter
Số chân Digital Output	7 Outputs
Số chân Logic Output	8 Outputs
Số chân analog	6 Analogs
Số chân xuất xung PWM	6 Outputs
LAN input	1 port
Led báo trạng thái	2

Cấu hình thiết bị:

Tên thông số	Giá trị
SRAM	8KB
Flash Memory	256KB trong đó 8 KB sử dụng bởi bộ nạp khởi động (Bootloader)
Clock Speed	16Mhz
Serial Port	3 port
MCU	ATMega 2560
Ethernet	Module 28J60 100Mbps

Điểm quan trọng trong thiết kế phần cứng do nhóm sinh viên tự nghiên cứu đó là khả năng lắp ghép với các bo mạch khác để mở rộng cấu hình cũng như thông số của hệ thống. Giao tiếp giữa 2 mạch thông qua giao tiếp Serial với khả năng ghép nối không giới hạn.

Một bo mạch được kết nối tới các bo mạch khác trong cùng hệ thống sẽ giúp tiết kiệm tối đa chi phí vì khả năng có thể nâng cấp tùy theo yêu cầu người dùng mà không làm ảnh hưởng đến hệ thống đã có.



Hình 1.4.1: Ví dụ nguyên lý về việc ghép nối 2 thiết bị

- Nhiệm vụ các thành phần:**

Các cảm biến dùng trong nhà thông minh được kết nối tới bo mạch chính (mainboard), mainboard có nhiệm vụ nhận dữ liệu, phân tích dữ liệu và trả về khi có yêu cầu.

Mainboard là trung tâm điều khiển của hệ thống nhà thông minh, có nhiệm vụ dựa vào các cảm biến sẽ đưa ra các đáp ứng theo ngữ cảnh hoặc thực hiện lệnh từ người dùng gửi đến, trực tiếp điều khiển thiết bị.

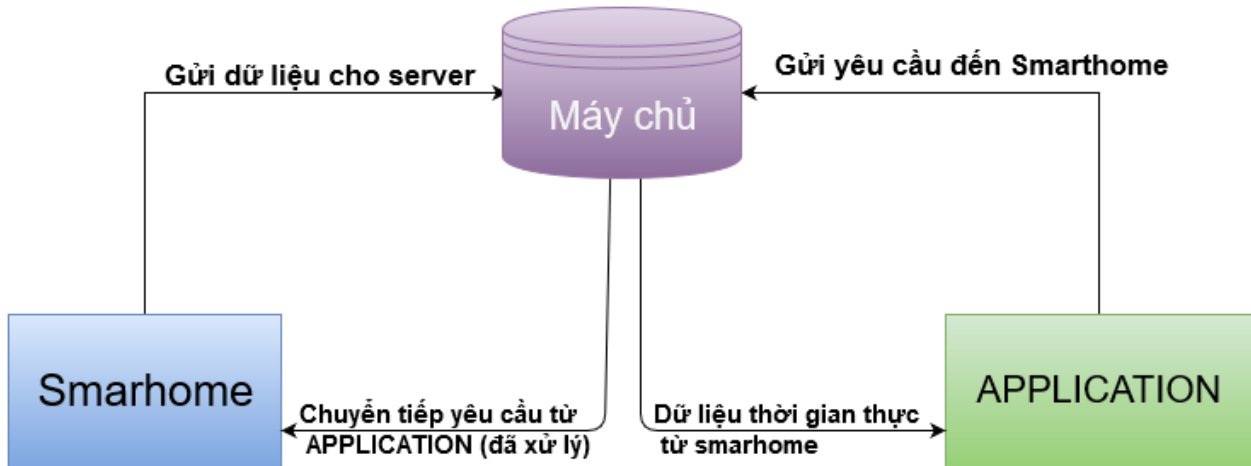
- Ưu điểm:

Hiện nay các thiết bị thông minh trên thị trường Việt Nam mới chỉ là các thiết bị đơn lẻ với những bộ vi xử lý đơn giản, thực hiện một công việc cố định được thiết lập sẵn và lập trình trong bộ nhớ ROM, những hệ thống được tích hợp có giá thành cao và chưa phổ biến. Với tất cả phần cứng do nhóm sinh viên tự thiết kế có chi phí chỉ dưới 2 triệu đồng, rẻ hơn rất rất nhiều lần so với những thiết bị nhập khẩu hoặc lắp ráp trong nước. Nếu xét về giá cũng được coi là một điểm mạnh, nếu xét về chức năng thì phần cứng tự lắp này có chức năng không kém những thiết bị đang được bán trên thị trường.

Tính tương thích và ổn định cao:

- Phần cứng tương thích với tất cả các loại cảm biến có xuất chuẩn dữ liệu
- Độ trễ cho các phản hồi từ thiết bị và cảm biến thấp, tốc độ gửi và nhận dữ liệu từ server nhanh đảm bảo tính ổn định trong suốt quá trình sử dụng.
-

CHƯƠNG 2: XÂY DỰNG NGUYÊN LÝ HOẠT ĐỘNG



Sơ đồ nguyên lý hoạt động cho hệ thống nhà thông minh

2.1 Tầng mạng

2.1.1 Tổng quan

Máy chủ server sẽ làm trung gian trong việc giao tiếp giữa người dùng (Ứng dụng) với thiết bị bằng các giao thức web thông thường, việc đồng bộ dữ liệu từ thiết bị qua máy chủ sẽ giúp việc che giấu thông tin được tốt.

Với mỗi 1 khoảng thời gian đặt trước hệ thống smarthome sẽ gửi dữ liệu về cho server để thông báo các trạng thái hiện tại của ngôi nhà (bao gồm thông tin về cảm biến, các trạng thái cảnh báo phát sinh), bên phía server sẽ lưu lại các thông tin này. Nếu xuất hiện một ngắt (ngắt do cảnh báo an ninh hoặc an toàn) hệ thống sẽ gửi thông điệp cảnh báo về cho server. Bên phía Application sẽ lấy các dữ liệu từ server để hiển thị ra ứng dụng thông báo cho người dùng biết.

Khi người dùng có yêu cầu đối với hệ thống (yêu cầu bật tắt thiết bị chẳng hạn), ứng dụng sẽ gửi thông điệp đến cho server, ngay lập tức, server sẽ chuyển tiếp yêu cầu ngay tới hệ thống smarthome. Tuy nhiên, việc chuyển tiếp dữ liệu đã được qua xử lý 1 bước trước khi chuyển tới smarthome.

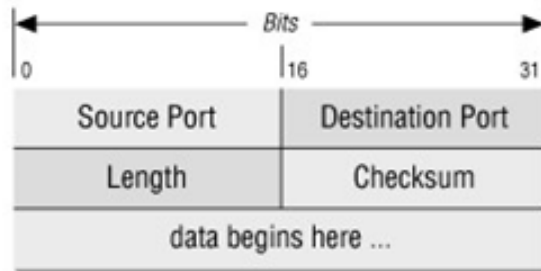
Tất cả các yêu cầu gửi đi và chuyển tiếp yêu cầu đều trong khoảng thời gian rất ngắn yêu cầu độ trễ càng thấp càng tốt. Với các giao thức hiện tại mà nhóm sinh viên đã xây dựng thì tốc độ đều ở mức chấp nhận được (cỡ vài trăm mili giây).

2.1.2 Giao thức kết nối Internet

Nhóm nghiên cứu đã sử dụng giao thức UDP (User Datagram Protocol) và TCP (Transmission Control Protocol) trong việc truyền và nhận dữ liệu cho hệ thống của mình.

Giao thức UDP (User Datagram Protocol)

UDP là giao thức không liên kết, cung cấp dịch vụ giao vận (Transport Layer) không tin cậy được, sử dụng thay thế cho TCP trong tầng giao vận. Khác với TCP, UDP không có chức năng thiết lập và giải phóng liên kết, không có cơ chế báo nhận (ACK), không sắp xếp tuần tự các đơn vị dữ liệu (datagram) đến và có thể dẫn đến tình trạng mất hoặc trùng dữ liệu mà không hề có thông báo lỗi cho người gửi. Khuôn dạng của UDP datagram được mô tả như sau :



- Số hiệu cổng nguồn (Source Port - 16 bit): số hiệu cổng nơi đã gửi datagram
- Số hiệu cổng đích (Destination Port - 16 bit): số hiệu cổng nơi datagram được chuyển tới
- Độ dài UDP (Length - 16 bit): độ dài tổng cộng kể cả phần header của gói UDP datagram.
- UDP Checksum (16 bit): dùng để kiểm soát lỗi, nếu phát hiện lỗi thì UDP datagram sẽ bị loại bỏ mà không có một thông báo nào trả lại cho trạm gửi. UDP có chế độ gán và quản lý các số hiệu cổng (port number) để định danh duy nhất cho các ứng dụng chạy trên một trạm của mạng. Do có ít chức năng phức tạp nên UDP có xu thế hoạt động nhanh hơn so với TCP. Nó thường dùng cho các ứng dụng không đòi hỏi độ tin cậy cao trong giao vận.

UDP và IPv4 header

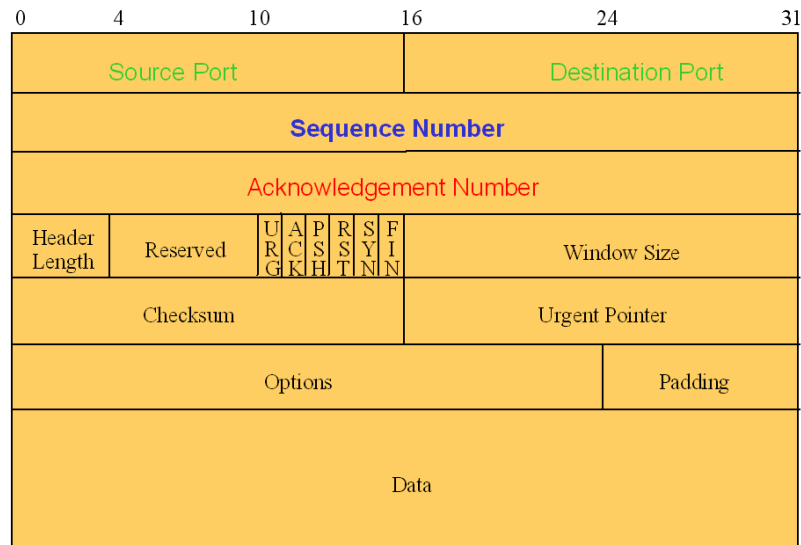
Version	IHL	Type of Service	Total Length		IP Header
Identification			Flag	Fragment Offset	
Time-to-live		Protocol	Header Checksum		
Source Address					
Destination Address					
Options				Padding	
Source Port			Destination Port		UDP
Length			Checksum		
Data begins here ...					

Giao thức TCP (Transmission Control Protocol)

TCP và UDP là 2 giao thức ở tầng giao vận và cùng sử dụng giao thức IP trong tầng mạng. Nhưng không giống như UDP, TCP cung cấp dịch vụ liên kết tin cậy và có liên kết. Có liên kết ở đây có nghĩa là 2 ứng dụng sử dụng TCP phải thiết lập liên kết với nhau trước khi trao đổi dữ liệu. Sự tin cậy trong dịch vụ được cung cấp bởi TCP được thể hiện như sau:

- Dữ liệu từ tầng ứng dụng gửi đến được được TCP chia thành các segment có kích thước phù hợp nhất để truyền đi .
 - Khi TCP gửi 1 segment, nó duy trì một thời lượng để chờ phúc đáp từ trạm nhận. Nếu trong khoảng thời gian đó phúc đáp không tới được trạm gửi thì segment đó được truyền lại.
 - Khi TCP trên trạm nhận nhận dữ liệu từ trạm gửi nó sẽ gửi tới trạm gửi 1 phúc đáp tuy nhiên phúc đáp không được gửi lại ngay lập tức mà thường trễ một khoảng thời gian .
 - TCP duy trì giá trị tổng kiểm tra (checksum) trong phần Header của dữ liệu để nhận ra bất kỳ sự thay đổi nào trong quá trình truyền dẫn. Nếu 1 segment bị lỗi thì TCP ở phía trạm nhận sẽ loại bỏ và không phúc đáp lại để trạm gửi truyền lại segment bị lỗi đó.
- Giống như IP datagram, TCP segment có thể tới đích một cách không tuần tự. Do vậy TCP ở trạm nhận sẽ sắp xếp lại dữ liệu và sau đó gửi lên tầng ứng dụng đảm bảo tính đúng đắn của dữ liệu. Khi IP datagram bị trùng lặp TCP tại trạm nhận sẽ loại bỏ dữ liệu trùng lặp đó .

TCP cũng cung cấp khả năng điều khiển luồng. Mỗi đầu của liên kết TCP có vùng đệm (buffer) giới hạn do đó TCP tại trạm nhận chỉ cho phép trạm gửi truyền một lượng dữ liệu nhất định (nhỏ hơn không gian buffer còn lại). Điều này tránh xảy ra trường hợp trạm có tốc độ cao chiếm toàn bộ vùng đệm của trạm có tốc độ chậm hơn.



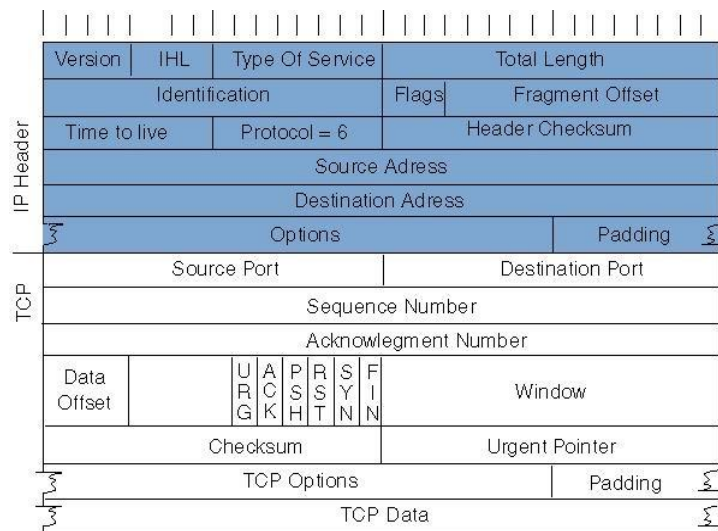
Các tham số trong khuôn dạng trên có ý nghĩa như sau:

- Source Port (16 bits) là số hiệu cổng của trạm nguồn .
- Destination Port (16 bits) là số hiệu cổng trạm đích .
- Sequence Number (32 bits) là số hiệu byte đầu tiên của segment trừ khi bit SYN được thiết lập. Nếu bit SYN được thiết lập thì sequence number là số hiệu tuần tự khởi đầu ISN (Initial Sequence Number) và byte dữ liệu đầu tiên là $ISN + 1$. Thông qua trường này TCP thực hiện việc quản lý từng byte truyền đi trên một kết nối TCP.
- Acknowledgment Number (32 bits). Số hiệu của segment tiếp theo mà trạm nguồn đang chờ để nhận và ngầm định báo nhận tốt các segment mà trạm đích đã gửi cho trạm nguồn .
- Header Length (4 bits). Số lượng từ (32 bits) trong TCP header, chỉ ra vị trí bắt đầu của vùng dữ liệu vì trường Option có độ dài thay đổi. Header length có giá trị từ 20 đến 60 byte .
- Reserved (6 bits). Dành để dùng trong tương lai .
- Control bits : các bit điều khiển
 - URG : xác định vùng con trỏ khẩn có hiệu lực.
 - ACK : vùng báo nhận ACK Number có hiệu lực.
 - PSH : chức năng PUSH.
 - RST : khởi động lại liên kết.
 - SYN : đồng bộ hoá các số hiệu tuần tự (Sequence number).
 - FIN : không còn dữ liệu từ trạm nguồn.
- Window size (16 bits) : cấp phát thẻ để kiểm soát luồng dữ liệu (cơ chế cửa sổ trượt). Đây chính là số lượng các byte dữ liệu bắt đầu từ byte được chỉ ra trong vùng ACK

number mà trạm nguồn sẵn sàng nhận.

- Checksum (16 bits). Mã kiểm soát lỗi cho toàn bộ segment cả phần header và dữ liệu.
- Urgent Pointer (16 bits). Con trỏ trỏ tới số hiệu tuần tự của byte cuối cùng trong dòng dữ liệu khẩn cho phép bên nhận biết được độ dài của dữ liệu khẩn. Vùng này chỉ có hiệu lực khi bit URG được thiết lập.
- Option (độ dài thay đổi). Khai báo các tùy chọn của TCP trong đó thông thường là kích thước cực đại của 1 segment: MSS (Maximum Segment Size).
- TCP data (độ dài thay đổi). Chứa dữ liệu của tầng ứng dụng có độ dài ngầm định là 536 byte. Giá trị này có thể điều chỉnh được bằng cách khai báo trong vùng Option.

TCP và IPv4 header



2.1.3 Cơ chế xác định địa chỉ thiết bị

• Mục đích của cơ chế xác định địa chỉ:

Đối với Smarthome nói riêng và các thiết bị IoT(Internet of Things) nói chung khi kết nối vào Internet muốn điều khiển hay giao tiếp được phải thông qua địa chỉ IP của nó.

Đối với **IPv4**, chúng ta có **IP public** và **IP private**, Smarthome trong mạng LAN được đặt **IP private** và không được sử dụng để kết nối ra ngoài Internet vì có vô số **IP private** giống nhau, Smarthome chỉ ra ngoài Internet được khi nó có **địa chỉ public** vì **địa chỉ public** là duy nhất, do vậy cần phải có một kỹ thuật để chuyển đổi các IP private trong mạng LAN thành IP public để ra ngoài Internet, NAT(Network Address Translation) sẽ giải quyết vấn đề này.

• **NAT(Network Address Translation):**

Ban đầu, NAT được đưa ra nhằm giải quyết vấn đề thiếu hụt địa chỉ của IPv4, nhưng sau này được phát triển nhằm giải quyết

- NAT giúp chia sẻ kết nối Internet (hay 1 mạng khác) với nhiều máy trong LAN chỉ với 1 IP duy nhất, hay 1 dãy IP cụ thể.
- NAT che giấu IP bên trong LAN.
- NAT giúp quản trị mạng lọc các gói tin được gửi đến hay gửi từ một địa chỉ IP và cho phép hay cấm truy cập đến một port cụ thể.

- **NAT Overloading (hay PAT)**

Dùng để ánh xạ nhiều địa chỉ IP riêng sang một địa chỉ công cộng vì mỗi địa chỉ riêng được phân biệt bằng số port. Có tới 65. 356 địa chỉ nội bộ có thể chuyển đổi sang 1 địa chỉ công cộng. Nhưng thực tế thì khoảng 4000 port.

PAT hoạt động bằng cách đánh dấu một số dòng lưu lượng TCP hoặc UDP từ nhiều máy cục bộ bên trong xuất hiện như cùng từ một hoặc một vài địa chỉ Inside Global. Với PAT, thay vì chỉ dịch địa chỉ IP, NAT cũng dịch các cổng khi cần thiết.

Và bởi vì các trường của cổng có chiều dài 16 bit, mỗi địa chỉ Inside Global có thể hỗ trợ lên đến 65000 kết nối TCP và UDP đồng thời. Ví dụ, trong một hệ thống mạng có 1000 máy, một địa chỉ IP thực được dùng như là địa chỉ Inside Global duy nhất có thể quản lý trung bình sáu dòng dữ liệu đến và đi từ các máy trên Internet.

Ví dụ :

PAT map nhiều địa chỉ Private đến một địa chỉ Public, việc phân biệt các địa chỉ Private này được dựa theo port, ví dụ IP address 10. 12. 11. 3 sẽ được map đến ip address 192. 168. 1. 6:port_number.

Rút ra: Modem sẽ sử dụng NAT để trở Smarthome ra ngoài Internet thông qua số hiệu cổng (port number). Tuy nhiên đối với hầu hết các mạng Internet thuê bao hộ gia đình thì địa chỉ IP được cấp phát là địa chỉ IP động, nghĩa là địa chỉ này sẽ bị thay đổi theo mỗi khoảng thời gian sử dụng Internet. Để giải quyết vấn đề này chúng ta sẽ sử dụng cơ chế cập nhật địa chỉ thường xuyên mỗi khi địa chỉ IP của hệ thống thay đổi.

Đây là công nghệ được các dịch vụ DDNS sử dụng, DDNS lúc này chính là địa chỉ máy chủ mà Smarthome sẽ trở đến, tại sao phải sử dụng vì để giải quyết vấn đề IP động, đa số chúng ta hiện nay đều đăng ký dịch Internet IP động . Ip động là mỗi khi ta tắt mở Modem, Router hoặc do vấn đề về điện của nhà bạn hay cúp điện thì khi ta mở Modem, Router lại thì bên ISP (nhà cung cấp dịch Internet) sẽ cấp cho chúng ta một địa chỉ IP

mới, vì vậy trước đó nếu ta được cấp địa chỉ ip là A thì sau khi mở lại modem sẽ được cấp địa chỉ IP là B. Vấn đề trên sẽ được giải quyết hoàn toàn bằng dịch vụ DDNS được tích hợp sẵn trên **Smarthome**, DDNS sẽ tự động cập nhật địa chỉ IP của thiết bị sau khi đã bị đổi để gửi lên Server chạy dịch vụ DDNS tự thiết lập, từ đó bạn có thể truy cập vào **Smarthome** bằng địa chỉ của máy chủ xử lý.

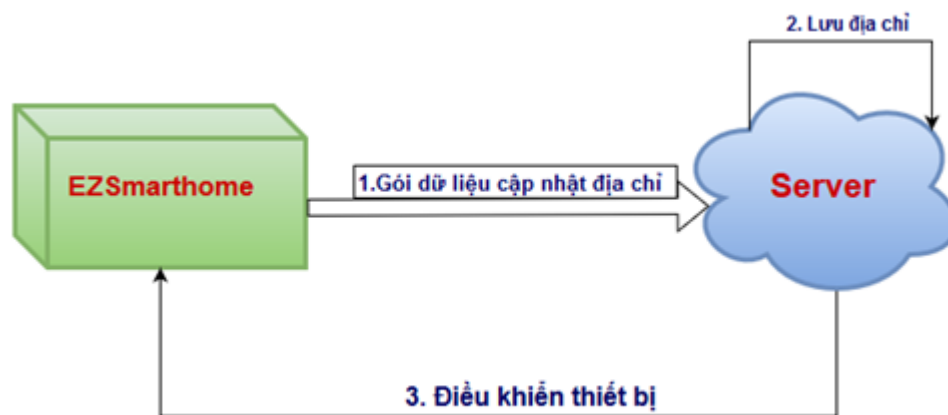
Nhóm sinh viên nghiên cứu không sử dụng dịch vụ DDNS trung gian mà tự xây dựng dựa trên nguyên lý làm việc tương tự. Từ đó giúp giảm thiểu tối đa chi phí xây dựng hệ thống.

- **Mô tả dịch vụ DDNS tự xây dựng :**

Mỗi một thiết bị khi được kết nối Internet đều được đặt 1 mã thiết bị duy nhất, dựa vào mã thiết bị hệ thống sẽ xác định được địa chỉ ứng với mỗi mã thiết bị đó. Hệ thống sử dụng giao thức TCP/IP cho kết nối của mình vì vậy nên mỗi gói tin được truyền tới server sẽ được định nghĩa địa chỉ nguồn và địa chỉ đích. Bên đích ta sẽ sử dụng hàm có sẵn trong php để bắt được địa chỉ nguồn. Khi có địa chỉ, server sẽ gửi lệnh điều khiển thông qua địa chỉ xác định được này.

Địa chỉ cùng với mã thiết bị sẽ được tự động lưu vào cơ sở dữ liệu, từ đó sẽ xác định được user nào có quyền điều khiển thiết bị nào.

- **Sơ đồ cơ chế:**



Hình 1.3.1 Sơ đồ cơ chế xác định địa chỉ thiết bị

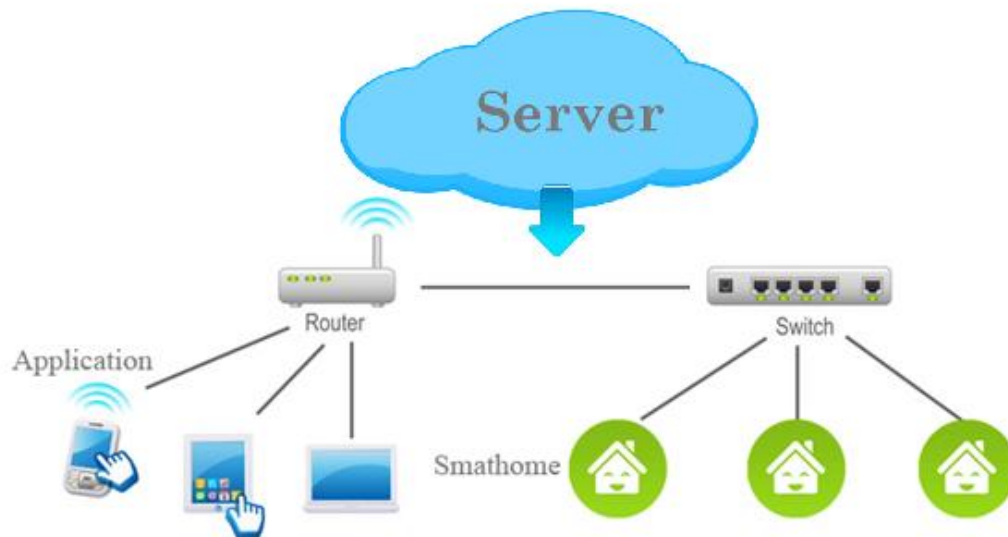
2.2. Tầng xử lý

2.2.1 Xử lý bên phía server

Server sẽ cho phép nhiều nhà thông minh cùng kết nối và điều khiển, hệ thống server sẽ là đầu não của tất cả các nhà thông minh.

- **Chức năng**

- (1) Server sẽ có nhiệm vụ nhận những thông tin mà smarthome gửi lên, sau đó lưu lại những thông tin nhận được. Những thông tin sẽ bao gồm dữ liệu cảm biến, trạng thái các thiết bị trong nhà. Từ đó server sẽ phân tích dữ liệu và trả về những thông báo cho người dùng biết thông qua ứng dụng bên phía người dùng (như app di động).
- (2) Server sẽ nhận yêu cầu điều khiển từ người dùng, phân tích yêu cầu, mã hóa và chuyển tiếp lại cho smarthome thực hiện.
- (3) Một server sẽ đảm nhận xử lý tất cả những hệ thống smarthome được kết nối vào, cơ chế xác định địa chỉ sẽ giúp cho server phân biệt được các smarthome trên toàn bộ mạng.



Hình II.2.1: Mô hình kết nối hệ thống

- **Phía ứng dụng người dùng:**

Nhóm sinh viên đã xây dựng các Web API để người dùng có thể tương tác với server. Khi người dùng đăng nhập vào hệ thống sẽ được 1 Token key (thẻ bài), mỗi lần request tới server sẽ phải kèm theo Token Key thì hệ thống mới chấp nhận yêu cầu, xử lý và trả về dữ liệu.

Token Key là một cơ chế tương tác của các WebAPI. Dựa vào token ta sẽ xác định được **mã thiết bị, địa chỉ thiết bị** mà người dùng muốn điều khiển

Ví dụ 1 request muốn bật đèn sẽ có địa chỉ dạng như sau:

http://118.70.177.146/smarthome/rooms/room/switch_device/8/225/?token=VFZSuk1VOVvaekJQUkVreVRuYzlQUT09

Trong đó:

Màu xanh: Tham số gửi đi

Màu cam: Mã token xác minh

Màu đen: địa chỉ server

- **Cơ chế an toàn:**

Khi nhận được các tham số điều khiển và địa chỉ đích sẽ gửi đến, phía server ngay lập tức sẽ gửi dữ liệu đến smarthome với đúng các tham số đó. Nhưng lưu ý lệnh được gửi đến smarthome sẽ kèm thêm các cơ chế xác minh người kết nối.

Việc sử dụng cơ chế xác minh có liên quan tới vấn đề về bảo mật bởi vì mọi yêu cầu từ ứng dụng người dùng đến smarthome sẽ được server che giấu, và những xác minh chỉ có server biết và ngầm hiểu với smarthome. Điều này tránh những truy cập không được phép từ bên ngoài vào smarthome mà không thông qua Server.

- **Dữ liệu trả về**

Khi người dùng yêu cầu dữ liệu, webservice sẽ trả về dữ liệu dạng json.

Ví dụ:

```
{ "rooms": [
  { "RoomName": "Batch room", "icon": "ion-ios-albums", "roomstate": false, "value": 0 },
  { "RoomName": "Demo Room", "icon": "ion-filing", "roomstate": false, "value": 0 },
  { "RoomName": "Garage", "icon": "ion-android-car", "roomstate": false, "value": 0 },
  { "RoomName": "Kids Bedroom", "icon": "ion-ios-paw", "roomstate": false, "value": 0 },
  { "RoomName": "Kitchen", "icon": "", "roomstate": false, "value": 0 },
  { "RoomName": "Loughe", "icon": "", "roomstate": false, "value": 0 },
  { "RoomName": "Master Bedroom", "icon": "", "roomstate": false, "value": 0 },
  { "RoomName": "Sauna", "icon": "ion-happy", "roomstate": false, "value": 0 },
  { "RoomName": "Study", "icon": "ion-ios-book", "roomstate": false, "value": 0 },
  { "RoomName": "Whole House", "icon": "ion-ios-grid-view", "roomstate": false, "value": 0 },
  ],
  "status": false }
```


2.2.2 Xử lý bên phía thiết bị phần cứng

- **Gửi dữ liệu**

Smarthome sẽ liên tục gửi các dữ liệu về server theo khoảng thời gian định trước , dữ liệu sẽ bao gồm trạng thái các cảm biến, trạng thái của các thiết bị trong nhà và tình trạng an ninh.

- **Trả về dữ liệu:**

Smarthome sẽ trả về dữ liệu dạng XML phục vụ các tiến trình thời gian thực khi tương tác. Có 2 cách để lấy dữ liệu từ smarthome đó là lấy trực tiếp và lấy thông qua cơ sở dữ liệu.

- (1) Nếu lấy thông qua cơ sở dữ liệu mọi trạng thái sẽ bị trễ 1 khoảng thời gian bằng thời gian cách nhau giữa mỗi lần smarthome gửi dữ liệu lên.
- (2) Nếu lấy trực tiếp mỗi lần yêu cầu thì dữ liệu gần như không bị trễ nhưng sẽ không đảm bảo được băng thông do quá trình gửi nhận thường xuyên dẫn đến smarthome không kịp thời gian xử lý.

Ưu điểm của cách 1 đó là chúng ta chỉ cần lấy thông qua database, thời gian xử lý sẽ nhanh hơn nhiều và các trạng thái sẽ được lưu trữ lại phục vụ việc thống kê và theo dõi.

Ưu điểm cách 2 đó chính là đảm bảo được tính đồng bộ trạng thái một cách nhanh chóng giữa ứng dụng và smarthome. Nhược điểm của cách này cũng chính là ưu điểm của cách kia trong việc lấy dữ liệu từ smarthome.

- **Lệnh điều khiển phần cứng**

Phần cứng sẽ được điều khiển thông qua tập lệnh được xây dựng sẵn, tập lệnh hoàn toàn được định nghĩa trong quá trình lập trình phần cứng. Nội dung tập lệnh do nhóm sinh viên nghiên cứu tự đặt và có cấu trúc bao gồm : “Tham số + Giá trị”.

Với mỗi lệnh mà server gửi cho smarthome, smarthome sẽ dựa vào vào cấu trúc lệnh để có thể hiểu được hệ thống muốn gì và thực hiện theo yêu cầu đó.

(Cấu trúc lệnh sẽ được mô tả trong chương III: Lập trình hệ thống)

2.2.3 Xử lý phía ứng dụng người dùng

Người dùng sẽ tương tác với smarthome thông qua ứng dụng trên điện thoại di động, server sẽ nhận lệnh từ người dùng, trả về dữ liệu thông báo trạng thái smarthome.

Người dùng sẽ tương tác thông qua các button, các dialog box và một số tiện ích trên ứng dụng.

Mobile app sử dụng công nghệ ajax nhận dữ liệu json mà server trả về. Mỗi request sẽ bao gồm 1 mã token đã đề cập trong phần trước để xác minh yêu cầu.

Ứng dụng di động có thể được build ra App cho Android, iOS, Microsoft Phone... và ứng dụng web..(Công nghệ sử dụng sẽ được đề cập trong chương sau)

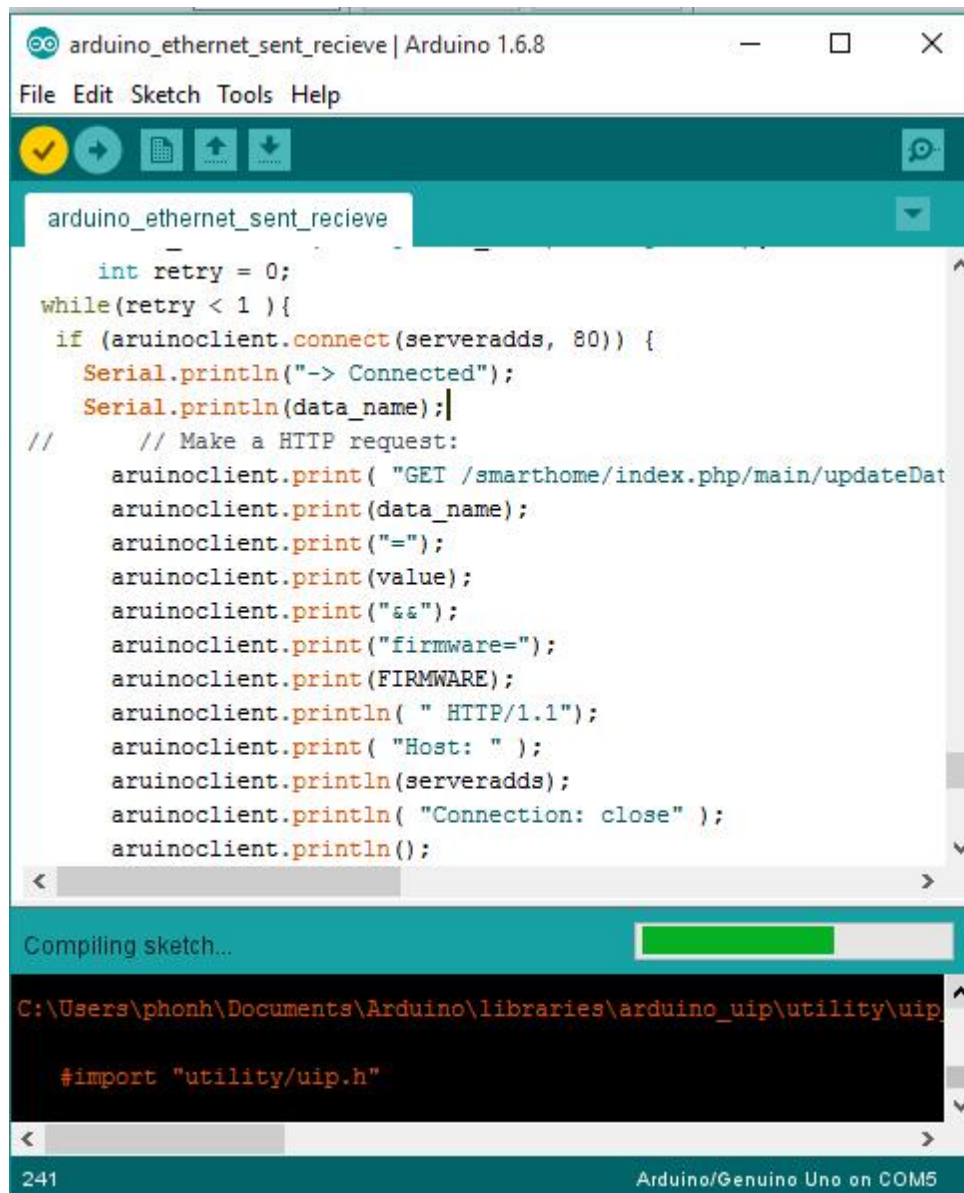
CHƯƠNG 3: LẬP TRÌNH HỆ THỐNG

3.1 Lập trình thiết bị

3.1.1 Công cụ lập trình.







Để lập trình cho smarthome phải sử dụng phần mềm có tên là Arduino IDE, đây là một IDE để lập trình cho các bo mạch Arduino.

Giao diện phần mềm:



Vùng lệnh

Bao gồm các nút lệnh menu (File, Edit, Sketch, Tools, Help). Phía dưới là các icon cho phép sử dụng nhanh các chức năng thường dùng của IDE được miêu tả như sau:

Icon	Chức năng
	Biên dịch chương trình đang soạn thảo để kiểm tra các lỗi lập trình.
	Biên dịch và upload chương trình đang soạn thảo.
	Mở một trang soạn thảo mới.
	Mở các chương trình đã lưu.
	Lưu chương trình đang soạn.
	Mở cửa sổ Serial Monitor để gửi và nhận dữ liệu giữa máy tính và board Arduino.

Vùng viết chương trình

Chúng ta sẽ viết các đoạn mã của mình tại đây. Tên chương trình được hiển thị ngay dưới đây các Icon, ở đây nó tên là “**Arduino_ethernet_sent_recieve**”. Để ý rằng phía sau tên chương trình có một dấu “\$”. Điều đó có nghĩa là đoạn chương trình chưa được lưu lại.

Vùng thông báo (debug)



Những thông báo từ IDE sẽ được hiển thị tại đây. Để ý rằng góc dưới cùng bên phải hiển thị loại board Arduino và cổng COM được sử dụng. Luôn chú ý tới mục này bởi nếu chọn sai loại board hoặc cổng COM, bạn sẽ không thể upload được code của mình.

Cấu trúc chương trình trong Arduino

Cấu trúc chương trình được chia làm 3 phần chính, đó là:

- Header: Bao gồm các khai báo thư viện và các hằng số cho chương trình chính
- Hàm setup(): Bao gồm các câu lệnh khởi tạo giá trị ban đầu cho arduino

- Hàm `loop()`: Giống như 1 hàm callback, hàm sẽ được lặp đi lặp lại mỗi khi thực hiện xong. Đây là hàm chính để giao tiếp với vi điều khiển, hàm này sẽ xử lý các ngắt, gọi các chương trình để thực hiện tuần tự, đến khi kết thúc chương trình sẽ tự động gọi lại.

Chương trình sau đây dùng để tạo một đèn nhấp nháy liên tục:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

3.1.2 Lập trình đọc dữ liệu từ cảm biến.

Để đọc được dữ liệu từ cảm biến chúng ta phải xây dựng các thư viện để có thể sử dụng hàm gọi ra trong chương trình chính. Sau đây là một mẫu code demo xây dựng thư viện đọc dữ liệu từ cảm biến nhiệt độ độ ẩm DHT11.

Code DTH11.cpp

```
#include "dht.h"
#define TIMEOUT 10000
// return values:
// 0 : OK
// -1 : Mã kiểm tra lỗi
// -2 : Thời gian hết hạn
int dht::read11(uint8_t pin)
{
  // Đọc giá trị
  int rv = read(pin);
  if (rv != 0) return rv;    // Chuyển đổi và lưu trữ
  humidity = bits[0]; // bit[1] == 0;
  temperature = bits[2]; // bits[3] == 0;
  // Kiểm tra mã kiểm tra lỗi
```

```

    uint8_t sum = bits[0] + bits[2]; // bits[1] && bits[3] both 0
    if (bits[4] != sum) return -1;
    return 0;
}
// Trả về: // 0 : OK // -1 : Lỗi // -2 : hết giờ
int dht::read22(uint8_t pin)
{
    // Đọc giá trị
    int rv = read(pin);
    if (rv != 0) return rv; // CONVERT AND STORE
    humidity = word(bits[0], bits[1]) * 0.1;
    int sign = 1;
    if (bits[2] & 0x80) // Nhiệt độ dương
    {
        bits[2] = bits[2] & 0x7F;
        sign = -1;
    }
    temperature = sign * word(bits[2], bits[3]) * 0.1; // Kiểm tra mã kiểm tra lỗi
    uint8_t sum = bits[0] + bits[1] + bits[2] + bits[3];
    if (bits[4] != sum) return -1;
    return 0;
}
// Trả về
// 0 : OK
// -2 : hết giờ
int dht::read(uint8_t pin)
{
    uint8_t cnt = 7;
    uint8_t idx = 0;
    for (int i=0; i< 5; i++) bits[i] = 0;
    pinMode(pin, OUTPUT);
    digitalWrite(pin, LOW);
    delay(20);
    digitalWrite(pin, HIGH);
    delayMicroseconds(40);
    pinMode(pin, INPUT);
    unsigned int loopCnt = TIMEOUT;
    while(digitalRead(pin) == LOW)

```

```

        if (loopCnt-- == 0) return -2;
    loopCnt = TIMEOUT;
    while(digitalRead(pin) == HIGH)
        if (loopCnt-- == 0) return -2;
    // Đọc đầu ra - 40 BITS => 5 BYTES
    for (int i=0; i<40; i++)
    {
        loopCnt = TIMEOUT;
        while(digitalRead(pin) == LOW)
            if (loopCnt-- == 0) return -2;

        unsigned long t = micros();

        loopCnt = TIMEOUT;
        while(digitalRead(pin) == HIGH)
            if (loopCnt-- == 0) return -2;
        if ((micros() - t) > 40) bits[idx] |= (1 << cnt);
        if (cnt == 0) // next byte?
        {
            cnt = 7;
            idx++;
        }
        else cnt--;
    }

    return 0;
}

```

File Header

```

#ifndef dht_h
#define dht_h

#if ARDUINO >= 100
    #include "Arduino.h"
#else

```

```

#include "WProgram.h"
#endif

#define DHT_LIB_VERSION "0.1.00"

class dht
{
public:
    int read11(uint8_t pin);
    int read22(uint8_t pin);
    double humidity;
    double temperature;

private:
    uint8_t bits[5]; // buffer to receive data
    int read(uint8_t pin);
};
#endif

```

Để đọc được **dữ liệu từ cảm biến** trong chương trình chính ta thực hiện:

```

//Khởi tạo các tham số cho cảm biến.
#include "DHT.h"
const int DHTPIN = 43;    //Đọc dữ liệu từ DHT11 ở chân 2 trên mạch Arduino
const int DHTTYPE = DHT11; //Khai báo loại cảm biến, có 2 loại là DHT11 và DHT22
DHT dht(DHTPIN, DHTTYPE);
//Hàm setup (chạy khi kích hoạt thiết bị)
float h = 0; //Đọc độ ẩm
float t = 0; //Đọc nhiệt độ
//Hàm gọi để đọc giá trị từ cảm biến
void updateTemp(){
    h = dht.readHumidity();
    t = dht.readTemperature();
}

```


3.1.3: Xử lý thời gian thực trong Arduino.

Sử dụng hàm `millis()` để lấy được thời gian hiện tại theo mili giây. Giá trị của hàm trả về sẽ là giá trị nguyên tăng dần theo thời gian. Dựa vào giá trị của hàm này hoàn toàn ta có thể xác định được thời gian hiện tại so với một mốc thời gian quy định sẵn.

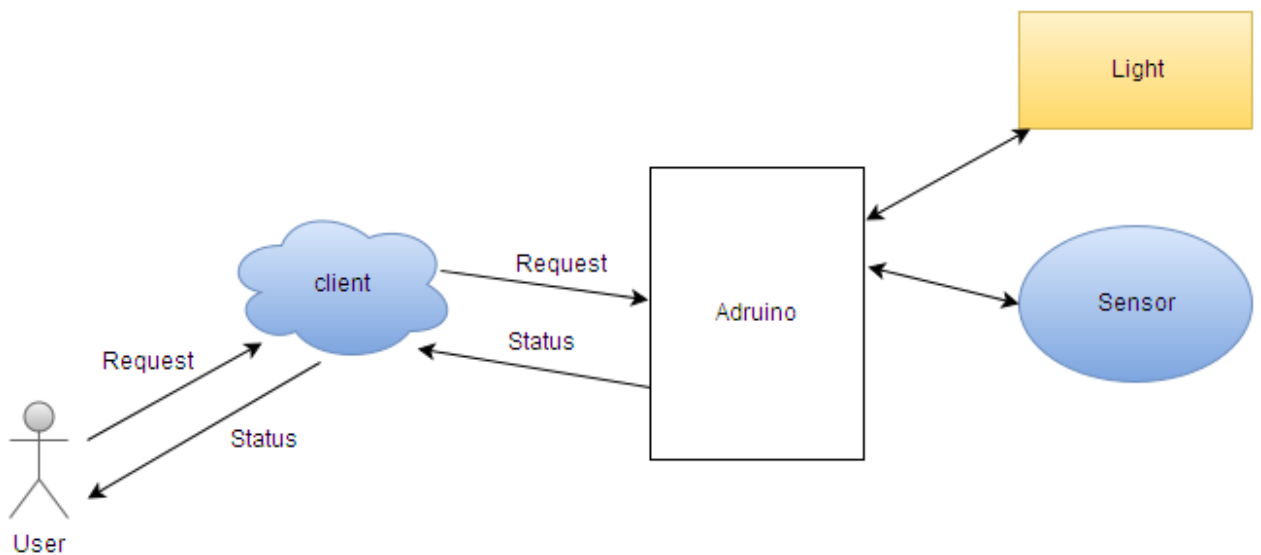
3.1.4: Các chế độ làm việc của smarthome

Smarthome sẽ làm việc theo chế độ tùy theo ngữ cảnh của người dùng, hiện tại hệ thống cung cấp 4 chế độ làm việc và có thể kích hoạt thông qua Mobile App từ người dùng:

- Mode = 0: Kích hoạt chế độ chống trộm, khi phát hiện người đột nhập hệ thống sẽ hiểu đây là trộm và đưa ra thông báo cho người dùng biết đồng thời kích hoạt chuông báo động.
- Mode = 1: Kích hoạt chế độ tự động: Đèn sẽ tự động sáng khi trời tối, tự động sáng khi phát hiện người vào phòng, tự động tắt khi người bước ra.
- Mode = 2: Kích hoạt chế độ bình thường: Mọi thiết bị sẽ được điều khiển thông qua app di động từ người dùng.
- Mode = 3: Chế độ ra ngoài, khi cả nhà đi ra ngoài tất cả các thiết bị sẽ được tắt để tiết kiệm điện, đồng thời kích hoạt chế độ chống trộm.

3.1.5 Xử lý hệ thống

Tổng quan : Arduino sẽ nhận **request** của người dùng dưới dạng chuỗi sau đó phân tích chuỗi để đưa ra xử lý bên trong hệ thống. Hệ thống xử lý gửi lại cập nhật trên **client arduino**:



3.1.5.1 Tạo client và xử lý Request

Khi đối tượng Client được tạo, Client sẽ lắng nghe ở cổng được khai báo (mặc định là cổng 80). Khi có request client sẽ đọc từng byte kí tự và ghép thành một chuỗi.

Để tách được dữ liệu trong chuỗi ta dùng biểu thức chính quy và thư viện Regrex.h trong C++ để phân tích các thành phần điều khiển trong request gửi lên.

```
String ParseRequestStr(String reqStr, String Name, boolean type)
{
    //type =1: get String
    //Type = 0: get Number

    MatchState ms;

    unsigned int index = 0;

    char bufferReqStr[reqStr.length() + 1];    //Buf for Reg Ex
    char bufferStr[reqStr.length()+1];        // Buf for char array
    reqStr.toCharArray(bufferStr, reqStr.length() + 1);    // String to Char Array
    //to char array

    String myWord = Name;

    char pattent[myWord.length()+1];    //as 1 char space for null is also required
    ms.Target(bufferStr);

    strcpy(pattent, myWord.c_str());

    (type == true) ? strcat(pattent, "=(%a+)") : strcat(pattent, "=(%d+)");

    // Serial.println (pattent);

    //end to char array

    while (true)
    {
        //(%a+)(%d)=(%a+)

        //r0=on&r1=on&postTemp=123
```

```

char result = ms.Match ( pattent , index);

if (result == REGEXP_MATCHED)

{

// Serial.print ("Matched on: ");

// Serial.println (ms.GetMatch(bufferReqStr));

// Serial.println ("Captures:");

for (int j = 0; j < ms.level; j++)

return (ms.GetCapture(bufferReqStr, j));

index = ms.MatchStart + ms.MatchLength;

}else

break;

}

return "0";

}

```

Hàm ParseRequestStr() trả về chuỗi được phân tích từ yêu cầu của người dùng gửi lên , chuỗi ký tự nhận được từ yêu cầu người dùng có dạng biểu thức chính quy , dựa vào đó ta xây dựng hàm ParseRequestStr() để tách ra chuỗi cần thiết để xử lý .

Đầu vào của hàm ParseRequestStr() gồm các tham số reqStr là chuỗi nhận được từ yêu cầu của người dùng , Name là tham số truyền vào để so sánh nhận dạng yêu cầu , type là tham số kiểu bool xác định kiểu chuỗi mà ta lấy về là gì : type =1 chuỗi lấy về ở dạng ký tự ; type =0 chuỗi lấy về ở dạng số mang giá trị

3.1.5.2 Cập nhật nhiệt độ và độ ẩm

Khi nhận được request của người dùng yêu cầu cập nhật lại nhiệt độ và độ ẩm , chương trình adruino chạy vào hàm updateTemp() để lấy dữ liệu từ cảm biến nhiệt độ và độ ẩm

```
void updateTemp(){
```

```

h = dht.readHumidity();

t = dht.readTemperature();

sento_webserver("temp", (String)t);

delay(500);

sento_webserver("humidity", (String)h);

Serial.print("Nhiet do: ");

Serial.println(t);          //Xuất nhiệt độ

Serial.print("Do am: ");

Serial.println(h);          //Xuất độ ẩm

Serial.println();

}

```

3.1.5.3 Cập nhật trạng thái đèn

Khi nhận yêu cầu của người dùng về thay đổi công suất của đèn hệ thống sẽ đáp ứng lại gần như đồng thời nhờ việc sử dụng thời gian thực để xử lý .

```
led.turnOn(light[i], capacity[i]) // capacity được lấy ra từ chuỗi yêu cầu của người dùng
```

3.1.5.4 Cập nhật lại client

Sau khi xử lý xong client sẽ được cập nhật lại để phản hồi lại người dùng.

```

readString="";

client.println("<?xml version='1.0'?>");

client.print("<dataSoucre>");

//return state of device

client.print("<currentmode>"); client.print(mode); client.print("</currentmode>"); client.println("");

for(int i = 0; i< 5; i++){

    client.print("<light" + (String)light[i] + ">"); client.print(capacity[i]); client.print("</light" +
    (String)light[i] + ">"); client.println("");

}

client.print("<rainsensor>"); client.print(israining); client.print("</rainsensor>"); client.println("");

```

```

client.print("<temp>"); client.print(t); client.print("</temp>"); client.println("");

client.print("<humidity>"); client.print(h); client.print("</humidity>"); client.println("");

//motion_state

client.print("<motion_state1>"); client.print(motion_state[0]); client.print("</motion_state1>");
client.println("");

client.print("<motion_state2>"); client.print(motion_state[1]); client.print("</motion_state2>");
client.println("");

client.print("</dataSoucre>");

client.stop();

//end return device}

```

3.2 Lập trình server

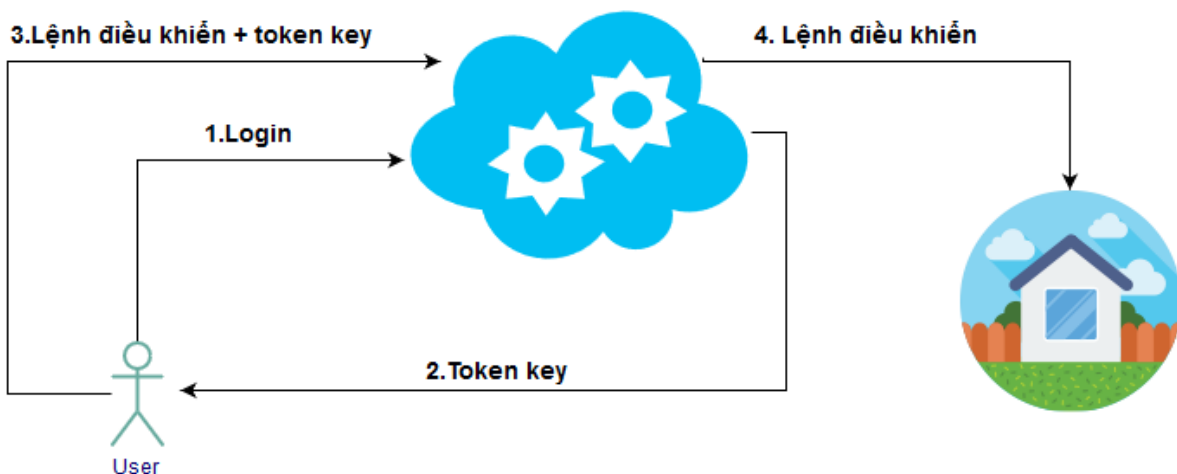
3.2.1 Thiết lập webserver

Nhóm sinh viên nghiên cứu đã xây dựng 1 webserver kết nối Internet chạy hệ điều hành Ubuntu server, sử dụng Apache, MySQL.

3.2.2 Xây dựng JSON và WebAPI.

3.2.2.1 Khởi tạo và nhận mã Token

Khi người dùng đăng nhập bằng tài khoản và mật khẩu webserver sẽ tạo ra một mã Token để tương tác với hệ thống



Hình III.2.1. 4 bước người dùng điều khiển hệ thống

- **Bước 1: Login**

Người dùng sẽ đăng nhập vào hệ thống bằng tài khoản và mật khẩu, hệ thống sẽ xác minh tài khoản + Mật khẩu, nếu tài khoản + mật khẩu đúng sẽ đưa đến bước 2. Nếu sai không làm gì cả.

- **Bước 2: Trả về mã Token**

Sau khi đăng nhập hệ thống thành công, server sẽ tạo ra một mã token duy nhất, mã này sẽ được lưu trong cơ sở dữ liệu và có giá trị trong 1 giờ nếu không có hành động nào đối với hệ thống.

Như đã đề cập ở trên, mã token sẽ thay cho việc đăng nhập, về sau mỗi yêu cầu gửi đến sẽ kèm theo mã token để xác minh đăng nhập.

- **Bước 3: Gửi lệnh điều khiển smarthome**

Người dùng có thể có bất cứ yêu cầu nào đối với smarthome thông qua webserver, mỗi yêu cầu sẽ phải kèm theo 1 mã token còn hiệu lực. Nếu mã này đúng sẽ webserver sẽ chuyển sang bước 5.

- **Bước 4: Gửi lệnh điều khiển cho smarthome.**

Webserver sẽ chuyển tiếp lệnh của người dùng đến smarthome thực thi.

Trên đây là 4 bước để điều khiển một hệ thống, ưu điểm của việc xây dựng này là chúng ta có thể tạo ra ứng dụng người dùng tách biệt hẳn với phần xử lý của server và tạo ra những rào cản bảo mật giúp hệ thống được an toàn. Hầu hết các web API đều có sử dụng các xác minh bằng mã token giúp cho các ứng dụng web có thể tương tác được dễ dàng.

Bảng dữ liệu lưu trữ token key

TokenID	Username	TokenKey	ExpiryTime
148	admin	VFZSUK1VNXFUVEpQVkdjMVQxRTIQUT09	1456974699
149	admin	VFZSUK1VNXFVVE5OZWtsNFRYYzIQUT09	1457078013
150	admin	VFZSUK1VNXFVVE5QVkdNMVRsRTIQUT09	1457084595
151	admin	VFZSUK1VNXFZekJOUkdzMVRrRTIQUT09	1457345794
152	admin	VFZSUK1VNXFaelZPVkVsNVRVRTIQUT09	1457500020

Mỗi token key được tạo bằng thuật toán mã hóa thời gian hiện tại với 1 số tham số khác, vì vậy chỉ server được cấp quyền mới tạo ra được một token key hợp lệ.

3.2.2 Xây dựng web API

Mọi tương tác của người dùng đối với hệ thống đều được thông qua web API, một web API sẽ bao gồm: “**địa chỉ + lệnh + tham số + mã xác minh**”

3.2.2.2 Header cho API

```
header('Access-Control-Allow-Origin: *');
header("Access-Control-Allow-Headers: X-API-KEY, Origin, X-Requested-With, Content-Type, Accept, Access-Control-Request-Method");
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");    $this->load->library('core/user');
if(!$this->user->isLogin()){
    die('You have not logged in!');
}
```

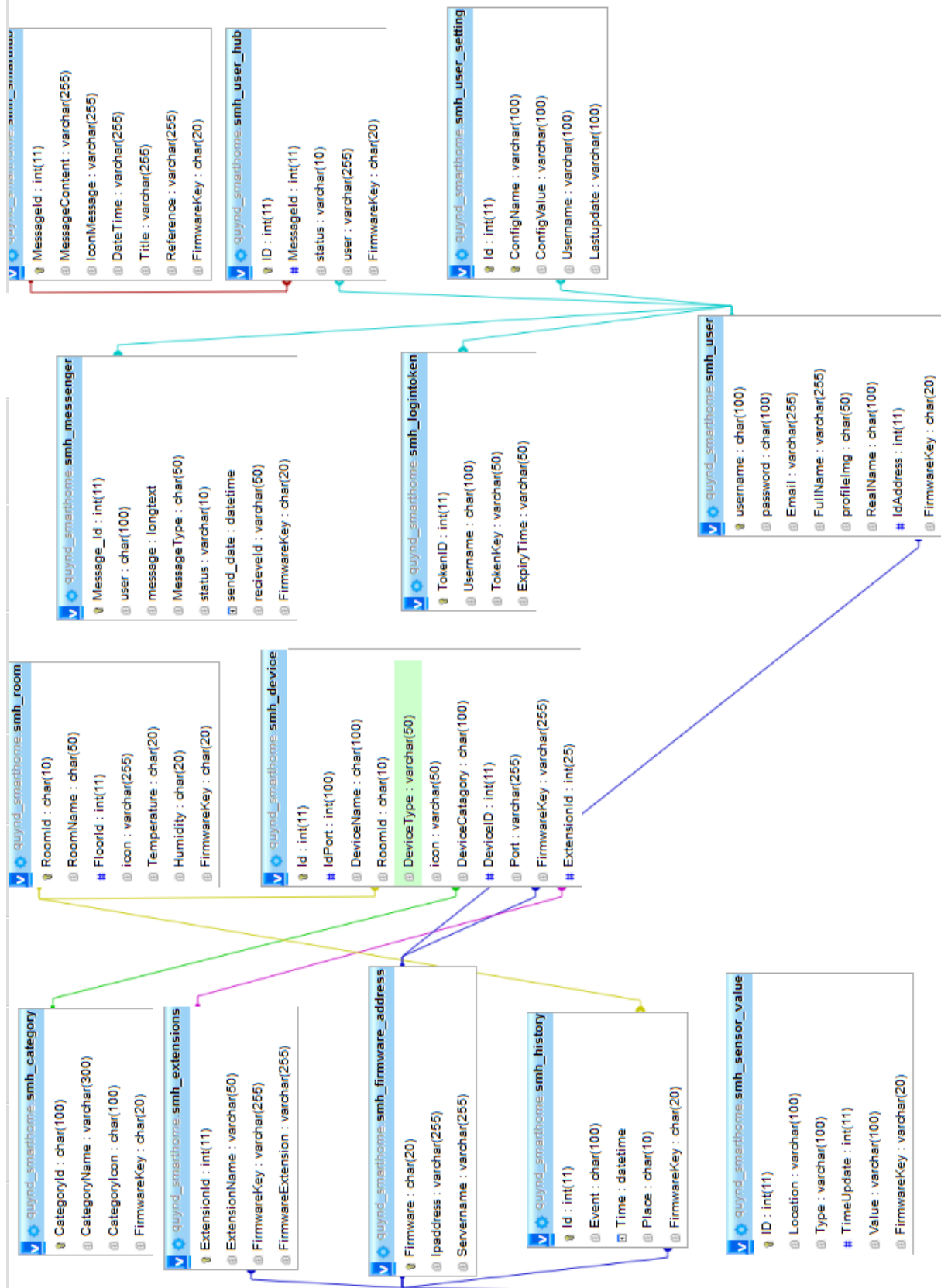
Phương thức isLogin của đối tượng user có chức năng xác minh xem mỗi request gửi đến là của đối tượng đã đăng nhập hay chưa, nếu chưa sẽ dừng ngay việc xử lý.

Header này được đặt ngay trong hàm tạo của đối tượng cha mà tất cả các API sẽ kế thừa lại. Vì vậy nên nó sẽ có hiệu lực đối với tất cả các API muốn cơ chế xác minh đăng nhập cho phương thức của mình.

Hàm gửi lệnh:

```
class Maincontrol{
    public function sendCommand($cmd){
        $CI =& get_instance();
        $CI->load->model('user/getUserData');
        $IdAddress = $CI->getUserData->getAddress(); //Lấy địa chỉ thiết bị điều khiển
        $firmware = $CI->getUserData->getFirmware(); //Lấy mã thiết bị
        //Địa chỉ nhận
        $ch = curl_init("http://".$IdAddress.":8083/?".$cmd."&firmware=".$firmware);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_BINARYTRANSFER, true);
        $content = curl_exec($ch);
        curl_close($ch);
        return true;
    }
}
```

3.2.3: Xây dựng Cơ sở dữ liệu



III.3 Lập trình ứng dụng di động

3.3.1 Nền tảng sử dụng

Ứng dụng di động sử dụng HTML5 và AngularJS trên nền tảng Cordova (giống như Phonegap) để tạo ra một Hybrid App (ứng dụng lai). Nền tảng cordova đi liền với framework ionic giúp xây dựng ứng dụng một cách nhanh chóng.

Bản chất của Hybrid là giao diện được viết bằng ngôn ngữ HTML, CSS, Javascript nhưng nó cho phép gọi được các API của hệ thống để thao tác với hệ điều hành (đọc ghi file, bluetooth, camera, sensor,...). Vì vậy hybrid vừa tận dụng được sức mạnh của Mobile webapp vừa khả năng thao tác hệ thống của Native app. Trong tương lai công nghệ Hybrid hứa hẹn sẽ là sự thay thế hoàn hảo có các ứng dụng mobile hiện nay. Đại diện cho công nghệ Hybrid này có thể kể đến PhoneGap, Ionic, Trigger.IO,... Hybrid có thể chạy trên iOS, Android, windows phone và một số nền tảng phổ biến khác. Lý do sử dụng công nghệ này giúp giảm thiểu thời gian và chi phí xây dựng cũng như bảo trì hệ thống.

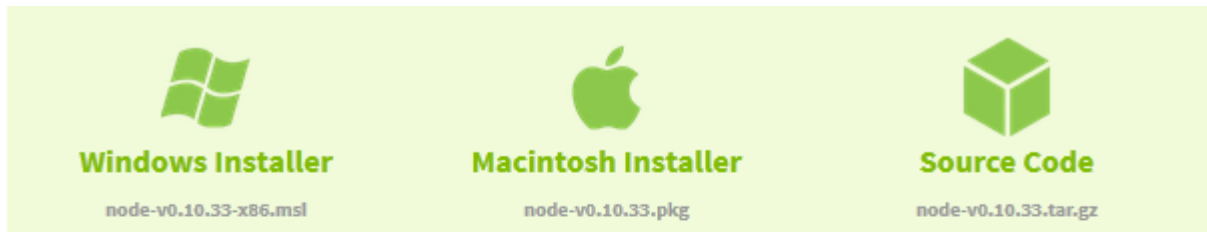
Nếu so về hiệu suất ứng dụng có thể đáp ứng được yêu cầu giống như Native App (Ứng dụng gốc), ngoài ra còn có thể chạy trên nhiều môi trường khác nhau rất tiện lợi.

3.3.2 Xây dựng ứng dụng

3.3.2.1: Thiết lập và cài đặt môi trường phát triển Ionic Framework

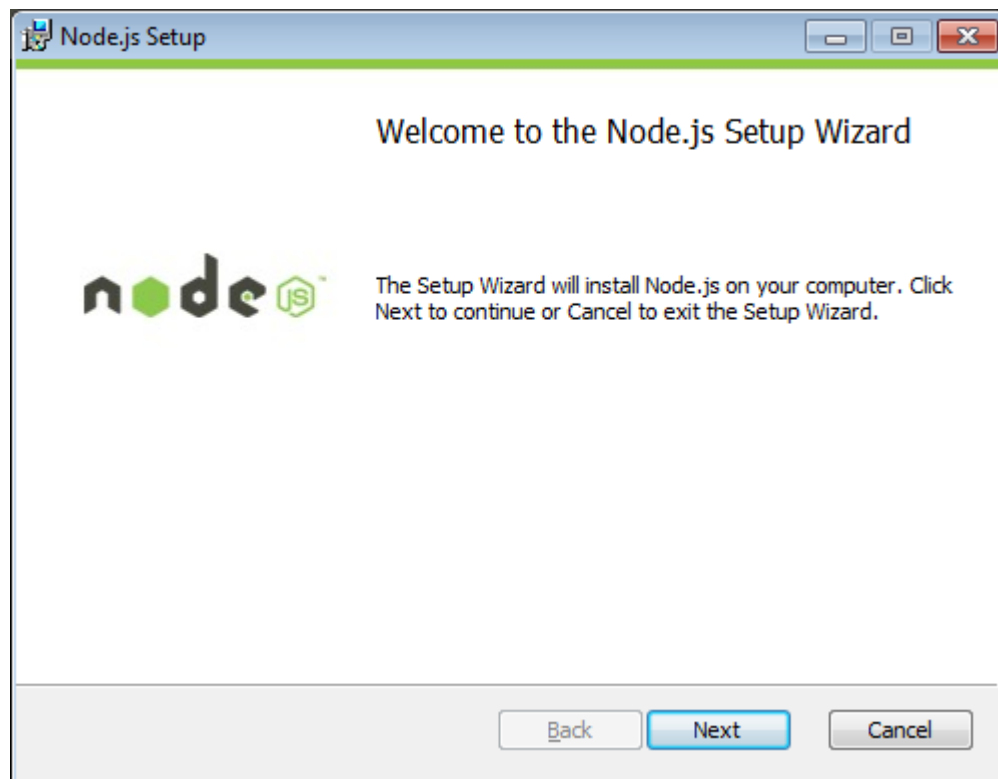
1. Cài đặt Nodejs

– Trong bài viết này mình sẽ hướng dẫn các bạn cài đặt Nodejs trên window, trước tiên truy cập website nodejs.org để download phần mềm về máy. Sau khi tải về chúng ta tiến hành cài đặt bình thường như những phần mềm khác

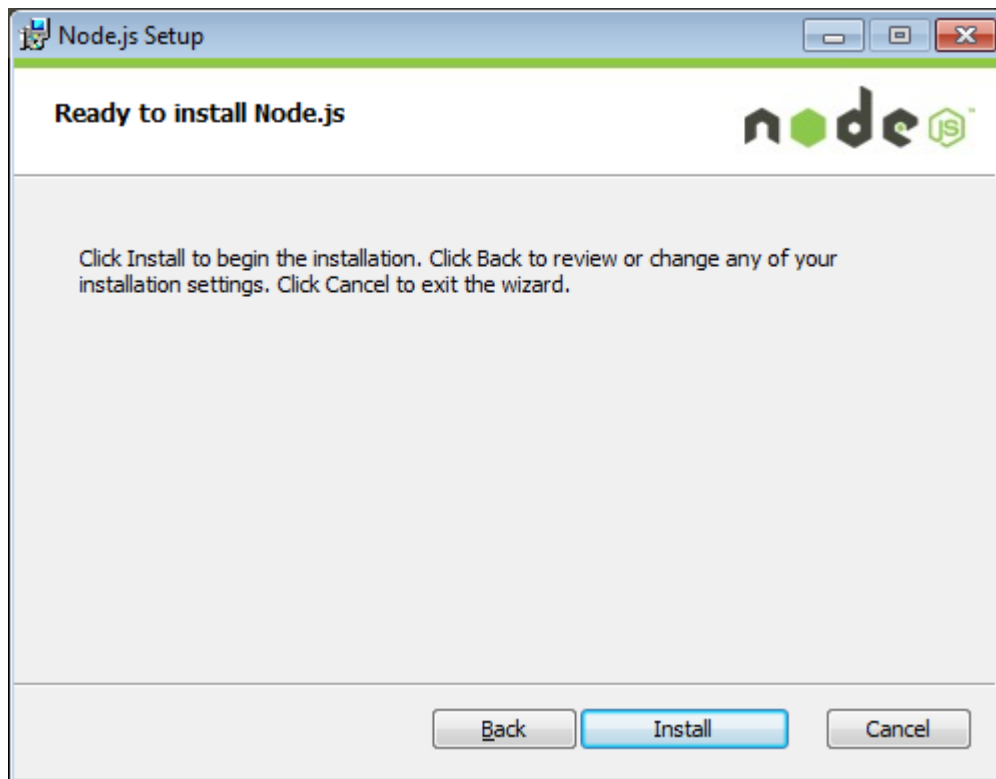


Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.exe)	32-bit	64-bit
Mac OS X Installer (.pkg)	Universal	
Mac OS X Binaries (.tar.gz)	32-bit	64-bit
Linux Binaries (.tar.gz)	32-bit	64-bit
SunOS Binaries (.tar.gz)	32-bit	64-bit
Source Code	node-v0.10.33.tar.gz	

Download Nodejs

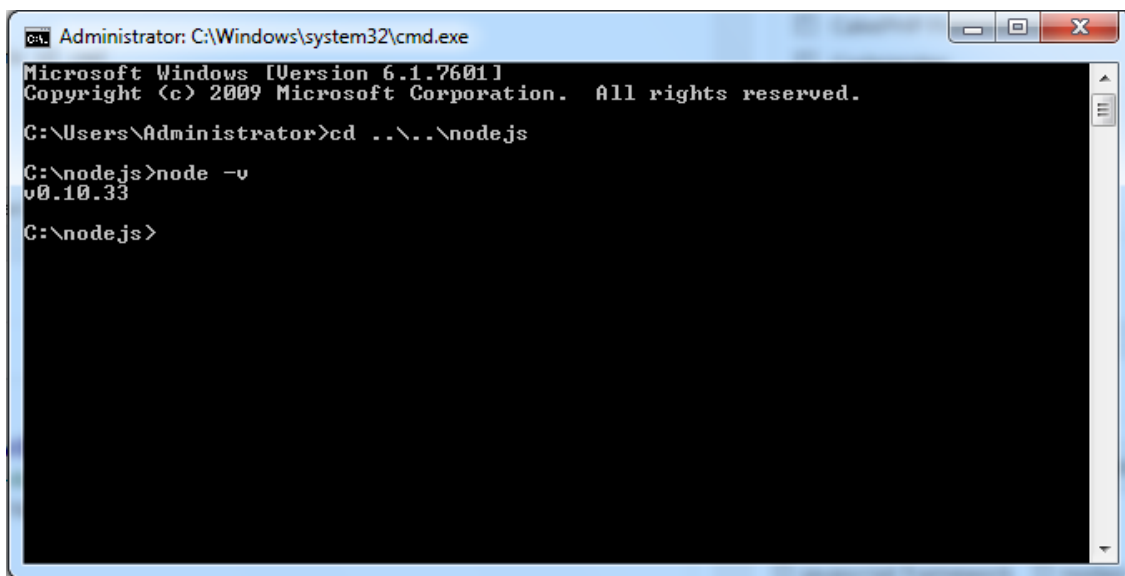


Nhấn Next để bắt đầu cài đặt



Nhấn Install để thực hiện cài đặt

– Để kiểm tra đã cài thành công hay chưa vào cửa sổ cmd, di chuyển tới thư mục cài đặt, mình cài đặt trong C:\nodejs, và chạy dòng lệnh sau để kiểm tra phiên bản cài đặt:
node -v



Xem phiên bản cài đặt

2. Cài đặt Ionic Cordova

Bước 1: Cài đặt Cordova

```
$ sudo npm install -g cordova
```

Bước 2: Cài đặt Ionic

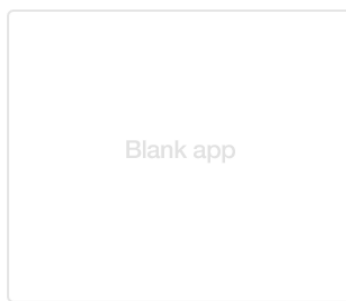
```
$ sudo npm install -g ionic
```

Bước 3: Tạo 1 project

```
$ ionic start Todo blank
```

Trong đó:

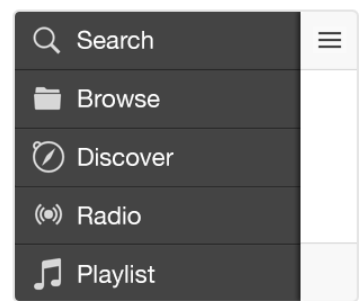
- Todo là tên của app
- blank là tùy chọn kiểu app mà Ionic hỗ trợ
 - tabs
 - sidemenu
 - blank



```
$ ionic start myApp blank
```



```
$ ionic start myApp tabs
```



```
$ ionic start myApp sidemenu
```

Bước 4: Cấu hình cho nền tảng

```
$ cd Todo
```

Dùng cho iOS

```
$ ionic platform add ios ( dùng cho iOS)
```

Dùng cho Android

```
$ ionic platform add android ( dùng cho Android)
```

Bước 5: Khởi động project

Dùng cho iOS

```
$ ionic build ios
$ ionic emulate ios
```

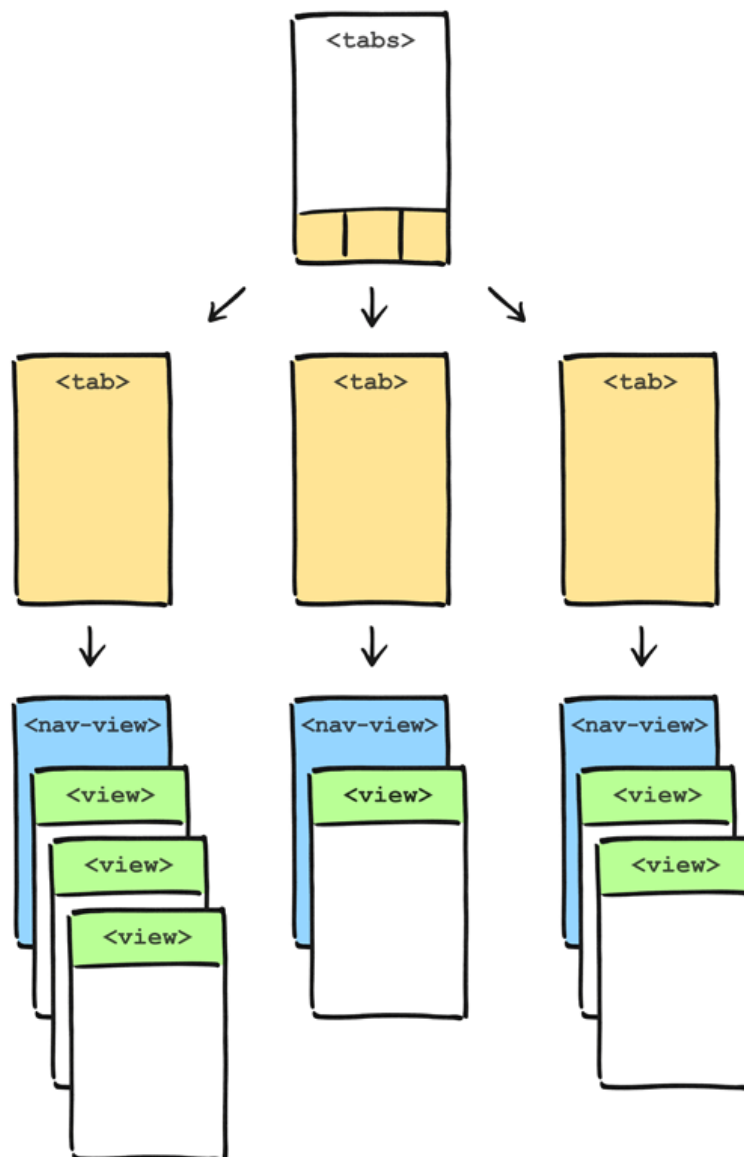
Dùng cho Android

```
$ ionic run android
```

Tiếp sau đó để xuất ra file có thể cài đặt được ta sử dụng phần mềm phát triển ví dụ như Android studio cho Android.

3.3.2.2 Tạo một Tabs và View trong Ionic

Cấu trúc trong Ionic



Hình III.3.2.2 Cấu trúc trong Ionic Framework

1. Cấu trúc file định nghĩa Tabs

```

1. <ion-tabs class="tabs-icon-top tabs-dark" >
    <!-- Login Tab -->
2. <ion-tab title="Đăng nhập" icon-off="ion-ios-pulse" icon-on="ion-
   ios-pulse-strong" href="#/tab/login" hidden=true>
3. <ion-nav-view name="tab-login"></ion-nav-view>
4. </ion-tab>
5. <!-- Dashboard Tab -->
6. <ion-tab title="Nhà tôi" icon-off="ion-ios-home-outline" icon-
   on="ion-ios-home" href="#/tab/dash">
7. <ion-nav-view name="tab-dash"></ion-nav-view>
8. </ion-tab>
9. <!-- Sensors Tab -->
10. <ion-tab title="Cảm biến" icon-off="ion-speedometer" icon-on="ion-
    speedometer" href="#/tab/sensors">
11. <ion-nav-view name="tab-sensor"></ion-nav-view>
12. </ion-tab>
13. </ion-tabs>

```

2. Cấu trúc file tab và view:

```

1. <ion-view title="Đăng nhập thiết bị" cache-view="false">
2. <ion-content hide-tabs has-header="true" class="padding
   background-login" ng-controller="LoginCtrl">
3. </ion-content>
4. </ion-view>

```

3. Khai báo đối tượng Tabs và view

```

.config(function($stateProvider,$httpProvider, $urlRouterProvider) {
    $stateProvider

```

```

        .state('tab', {
            url: '/tab',
            abstract: true,
            templateUrl: 'templates/tabs.html'
        })
        .state('tab.dash', {
            url: '/dash',
            views: {
                'tab-dash': {
                    templateUrl: 'templates/tab-dash.html',
                    controller: 'DashCtrl'
                }
            }
        });
    }
})

```

4. Cấu trúc file Controller để xử lý (Sử dụng AngularJS)

```

angular.module('starter.controllers', [])

.controller('DashCtrl', function($scope, $rootScope){

    //Các hàm xử lý view

})

```

Xử lý trong Ionic Framework sử dụng AngularJS, tài liệu có tại trang chủ của Ionic.

3.3.3 Một số chức năng đã xây dựng

1. Chức năng đăng nhập
2. Điều khiển thiết bị theo từng phòng trong nhà
3. Theo dõi tình trạng ngôi nhà thông qua cảm biến
4. Theo dõi thời tiết
5. Gửi tin nhắn giữa các thành viên trong gia đình
6. Cấu hình cài đặt cá nhân hóa người dùng
7. Thêm phần mở rộng thiết bị
8. Cấu hình hệ thống

9. Quản lý tài khoản

10. ...đang phát triển thêm một số chức năng:

- Kết nối thiết bị qua wifi
- Sử dụng QR-Code để ghép nối thiết bị
- Tích hợp camera an ninh

II: KẾT QUẢ NGHIÊN CỨU

II.1. Các kết quả nghiên cứu đạt được

II.1.1. Về việc Nghiên cứu xây dựng hệ thống nhà thông minh.

(1). Xây dựng thành công hệ thống nhà thông minh điều khiển và kết nối qua Internet

(2). Biết lập trình nhúng và ứng dụng trong tự động hóa.

(3). Xây dựng được các thư viện giao tiếp phần cứng với Arduino.

(4). Hiểu được cách giao tiếp giữa các cảm biến điện tử với môi trường.

(5) Hiểu được nguyên lý làm việc và nguyên lý điều khiển của các hệ vi xử lý.

(6). Có thể nghiên cứu và triển khai các giải pháp về tự động hóa trong đời sống cũng như công nghiệp.

(7). Tự có thể nghiên cứu và chế tạo các thiết bị điện tử, mạch điện có thể lập trình được.

(8). Tự xây dựng được các tập lệnh điều khiển cho phần cứng.

(9). Biết cách giao tiếp với các thiết bị phần cứng qua Internet.

(10). Biết cách tự cấu hình được một hệ thống mạng và server Internet.

(11). Hiểu được quy trình tiếp cận và xây dựng một hệ thống ứng dụng thực tế.

(12) Hiểu hơn về lập trình C++, cách tối ưu hóa thuật toán và tiết kiệm ô nhớ trong điều kiện hạn hẹp về tài nguyên hệ thống.

II.1.2. Về việc Nghiên cứu xây dựng ứng dụng di động:

(1). Có thể tự xây dựng được một ứng dụng Hybrid chạy trên nhiều nền tảng di động

(2). Hiểu và lập trình được với AngularJS, HTML5, CSS, PHP...

- (3). Biết cách sử dụng Ionic Framework cho các giải pháp về ứng dụng di động.
- (4). Biết giao tiếp ứng dụng thông qua các API kết nối đến server
- (5) Tự xây dựng được các API và cấu hình máy chủ web server trên hệ điều hành Ubuntu.
- (6). Sử dụng được PHP CodeIgniter Framework trong việc xử lý các yêu cầu trên Internet.
- (7). Biết giao tiếp ứng dụng với các thiết bị phần cứng thông qua Internet.

II.1.3. Về kết quả của sản phẩm tại thời điểm báo cáo:

Sản phẩm của nhóm sinh viên thực hiện đề tài cho đến thời điểm hiện tại là đã xây dựng được một ứng dụng điều khiển các thiết bị điện, điện tử trong gia đình, đọc được các chỉ số cảm biến môi trường, tự động hóa các tác vụ thay thế con người. Sản phẩm được lắp đặt trên mô hình thực tế và ứng dụng được cài đặt trên điện thoại di động bước đầu cho kết quả khả quan, đảm bảo được các yêu cầu thiết kế và bảo mật cơ bản. Ứng dụng có thời gian trễ ngắn, trong khoảng có thể chấp nhận được.

II.2. Đánh giá về các kết quả của nhóm đề tài

II.2.1. Tính nhanh và hiệu quả

Các tác vụ làm việc nhanh và cho kết quả thực hiện đúng, đảm bảo yêu cầu của người dùng về một hệ thống điều khiển:

- Xử lý được theo ngữ cảnh người dùng
- Xử lý được theo ngữ cảnh môi trường
- Tự động hóa hoàn toàn đối với ngôi nhà
- Tiết kiệm điện năng cho thiết bị điện

II.2.2. Ý nghĩa của các kết quả:

Các kết quả nghiên cứu có tính ứng dụng thực tế cao trong nhà thông minh nói riêng và trong lĩnh vực điều khiển và tự động hóa nói chung. Từ các kết quả nghiên cứu hoàn toàn có thể ứng dụng triển khai đối với các hệ thống khác mà không gặp phải bất kỳ khó khăn gì. Một ưu điểm so với những giải pháp hiện tại nằm ở chỗ giá thành rẻ hơn rất nhiều mà vẫn đảm bảo được tính chính xác và tính ổn định.

Nếu được áp dụng giải pháp này cho ngôi nhà sẽ giúp tiết kiệm đáng kể điện năng sử dụng vì ta có thể hoàn toàn có thể theo dõi và sử dụng một cách hợp lý các thiết bị điện trong gia đình.

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận về các kết quả đã đạt được

Sau thời gian nghiên cứu phát triển nhà thông minh nhóm sinh viên đã đạt được những kết quả bước đầu: Tạo ra ứng dụng di động cùng với hệ thống phần cứng kết nối và điều khiển qua Internet.

Nói chung, hệ thống làm việc chính xác và xử lý nhanh, các lỗi phát sinh đang được phát hiện và sửa chữa, sản phẩm sẽ ngày một hoàn thiện hơn. Nhưng cũng cần thời gian để tăng cường các tính năng bảo mật và tương tác người dùng được thuận tiện.

Bên cạnh việc xây dựng nhà thông minh nhóm sinh viên đang nghiên cứu thêm về lĩnh vực tự động hóa sử dụng những công nghệ đã tiếp thu được trong quá trình nghiên cứu đề tài này.

2. Kiến nghị về các lĩnh vực ứng dụng

Hiện tại sản phẩm của nhóm sinh viên đang hướng tới đối tượng là các hộ gia đình có thu nhập thấp và vừa, các thiết bị và công nghệ được giảm chi phí tới mức thấp nhất để mọi đối tượng đều có thể tiếp cận được.

- Trong Xây Dựng và Kiến Trúc: Sản phẩm giúp tạo ra cho căn nhà một không gian thân thiện và hiện đại cho những người sống bên trong. Đây là một xu hướng đã và sẽ phát triển trong thời gian tới.
- Trong những lĩnh vực khác: Có thể sử dụng để theo dõi nhà kính trồng cây trong nông nghiệp hoặc theo dõi các nhà xưởng trong công nghiệp.

3. Những định hướng nghiên cứu trong tương lai

Trong tương lai, nhóm sinh viên thực hiện đề tài muốn phát triển sản phẩm của mình trong nhiều lĩnh vực nếu có điều kiện. Nhưng trước mắt, để phát triển sản phẩm của mình, nhóm sinh viên đã có một vài định hướng ngắn hạn và dài hạn sau:

- Ngắn hạn
 - Nâng cao tính bảo mật và tối ưu hóa cho hệ thống.
 - Tích hợp một số công nghệ như nhận diện tiếng nói, nhận diện hình ảnh trong điều khiển.
 - Xây dựng thêm ngữ cảnh cho thiết bị, ngữ cảnh người dùng.

- Đưa hệ thống mở hơn với người dùng, có thể tùy biến hệ thống theo ý muốn của mình.
- Đưa ứng dụng chạy ngầm như một service trên các hệ điều hành điện thoại để có thể cập nhật tình trạng ngôi nhà bất kì lúc nào.

LỜI CẢM ƠN

Chúng em xin được gửi lời cảm ơn chân thành đến:

- Thầy Nguyễn Hải Dương bộ môn Công nghệ phần mềm cùng các Thầy Cô giáo trong khoa Công Nghệ Thông Tin đã cung cấp tài liệu và hướng dẫn chúng em trong suốt quá trình thực hiện đề tài này.
- Cùng sự trợ giúp của các bạn sinh viên đã cùng nghiên cứu trong thời gian đầu thực hiện.

Xin cảm ơn thầy cô và các bạn rất nhiều!

TÀI LIỆU THAM KHẢO

[1] Arduino

<https://vi.wikipedia.org/wiki/Arduino>

[2] Giới thiệu Arduino Mega2560

<http://arduino.vn/bai-viet/542-gioi-thieu-arduino-mega2560>

[3] DHT11 - Cảm Biến Độ Ẩm

<http://mcu.banlinhkien.vn/threads/dht11-cam-bien-do-am.122/>

[4] Giới thiệu Ionic Framework

<http://labs.septeni-technology.jp/none/gioi-thieu-ionic-framework/>

[5] Giao thức UDP (User Datagram Protocol) và TCP (Transmission Control Protocol)

<http://engisv.com/ctt.php?idTin=49>

[6] Arduino Datasheet

– Arduino Company

[7] Safe and simple AC PWM Dimmer for arduino / Raspberry pi

<http://www.instructables.com/id/safe-and-simple-AC-PWM-Dimmer-for-arduino-Raspberr/>

[8] Cách tạo 1 Hybrid App sử dụng HTML, CSS, JS

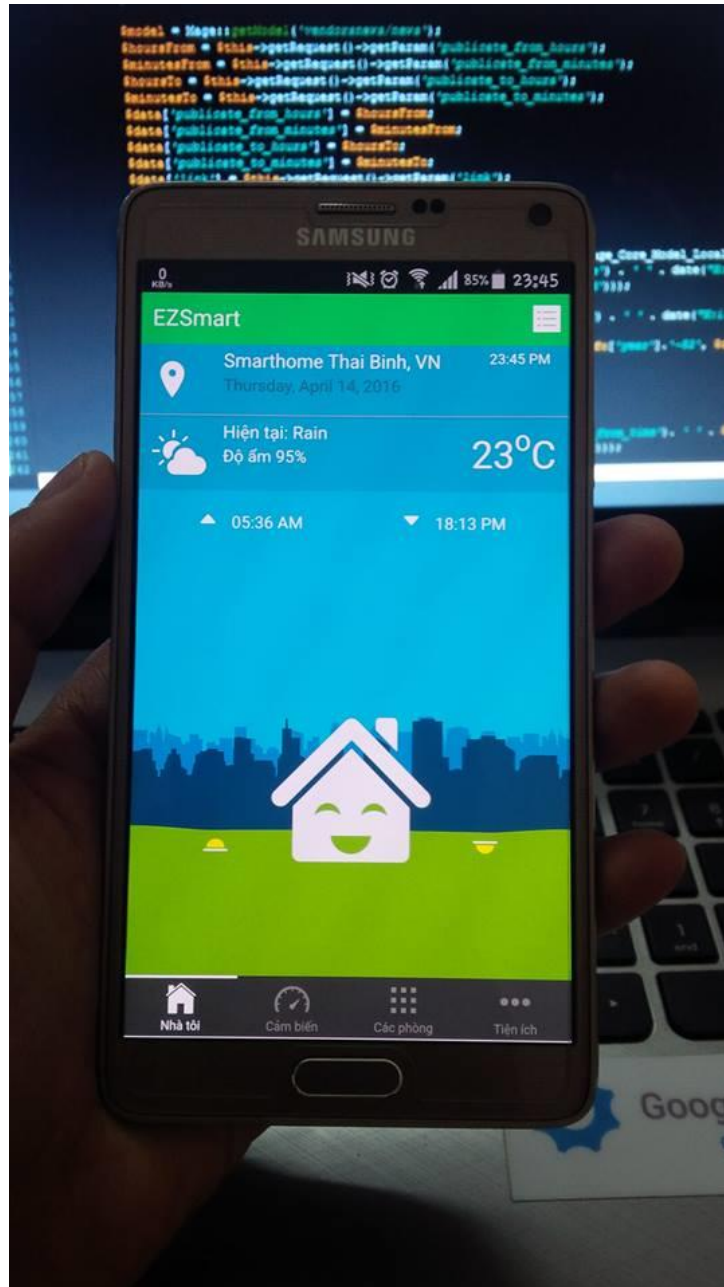
<http://kipalog.com/posts/Cach-tao-1-Hybrid-App-su-dung-HTML--CSS--JS>

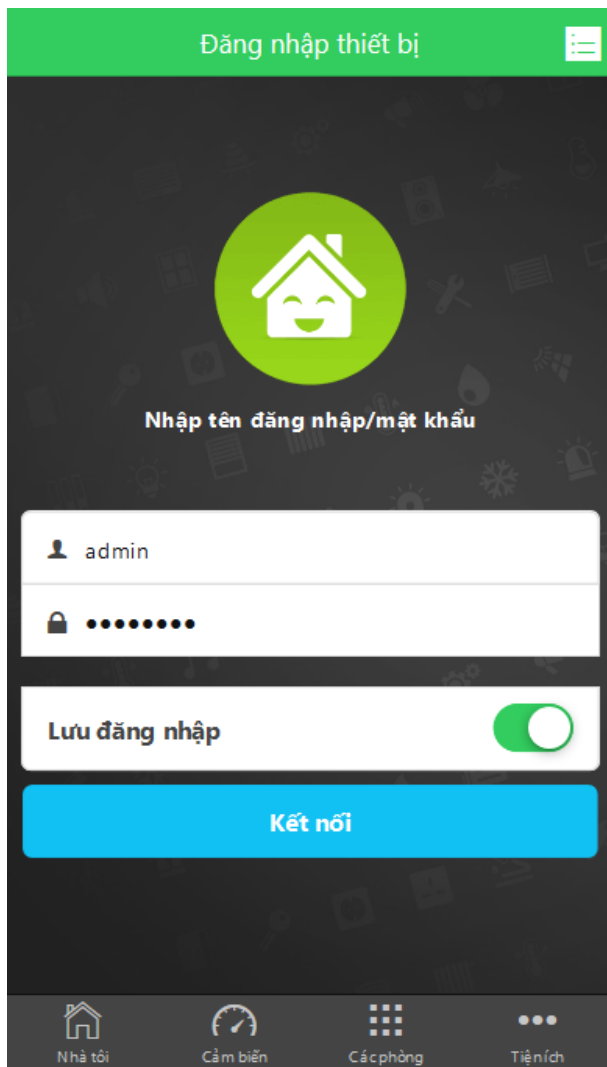
[9] Giáo trình Vi xử lý

<http://www.ziddu.com/download/11167114/olutionmanual-MicroprocessorsandInterfacing-DVHall.pdf.html>

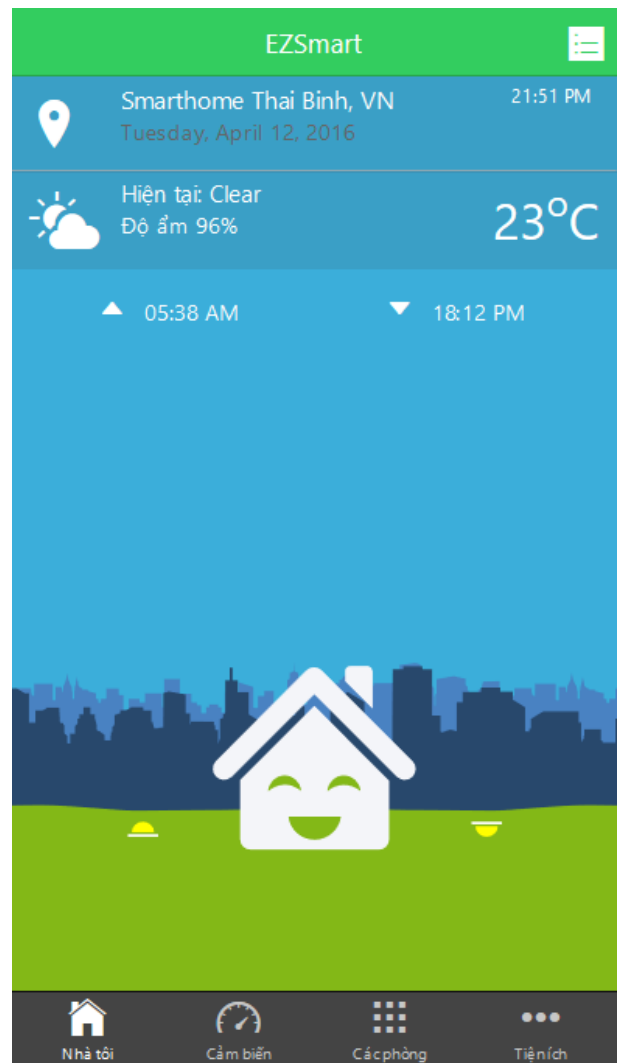
PHỤ LỤC 1:

Giao diện Ứng dụng di động EZSmart giao tiếp với nhà thông minh

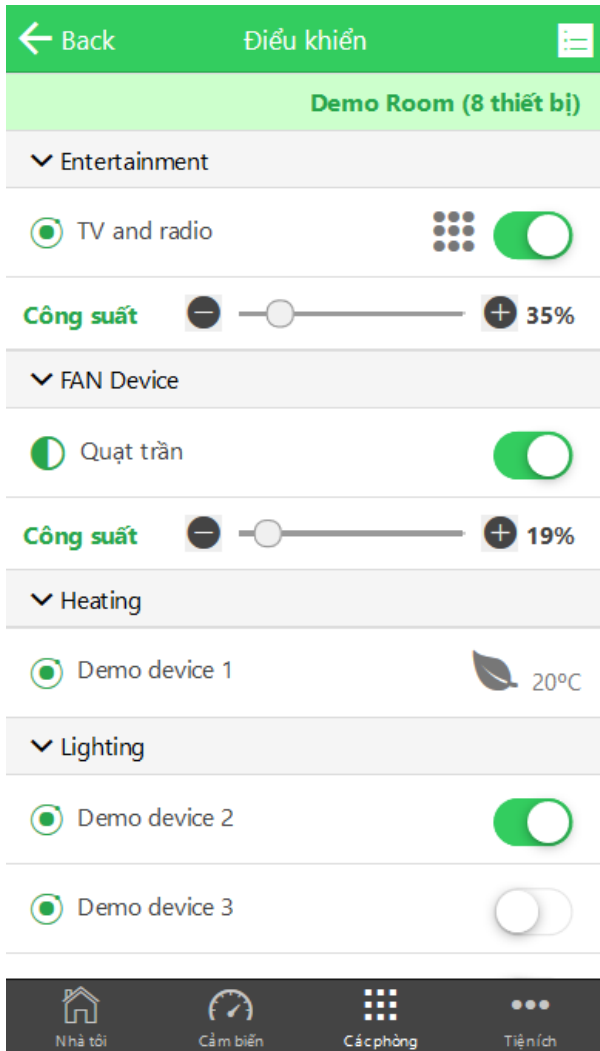




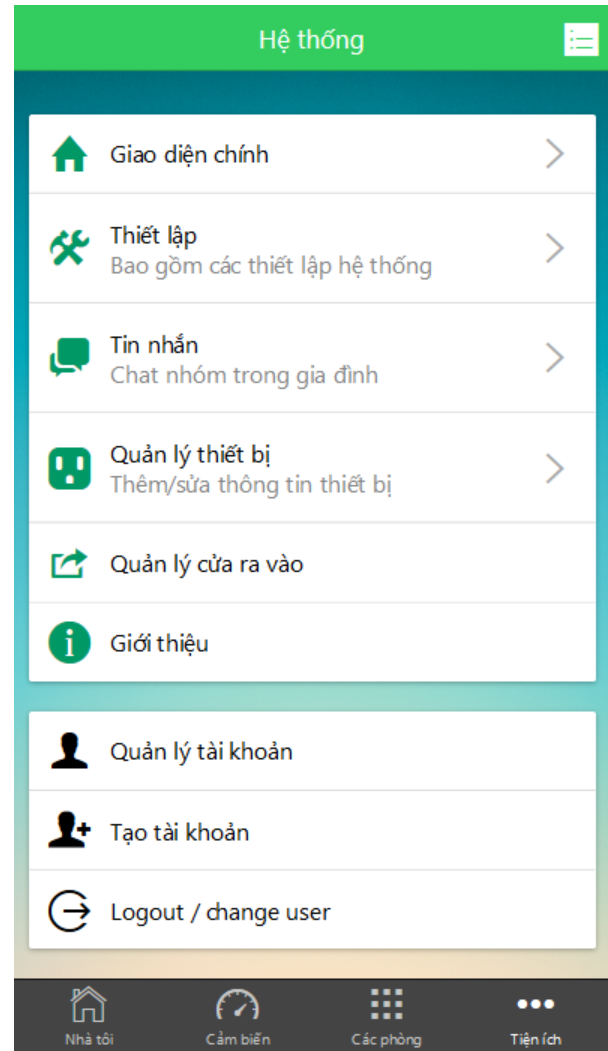
Giao diện đăng nhập



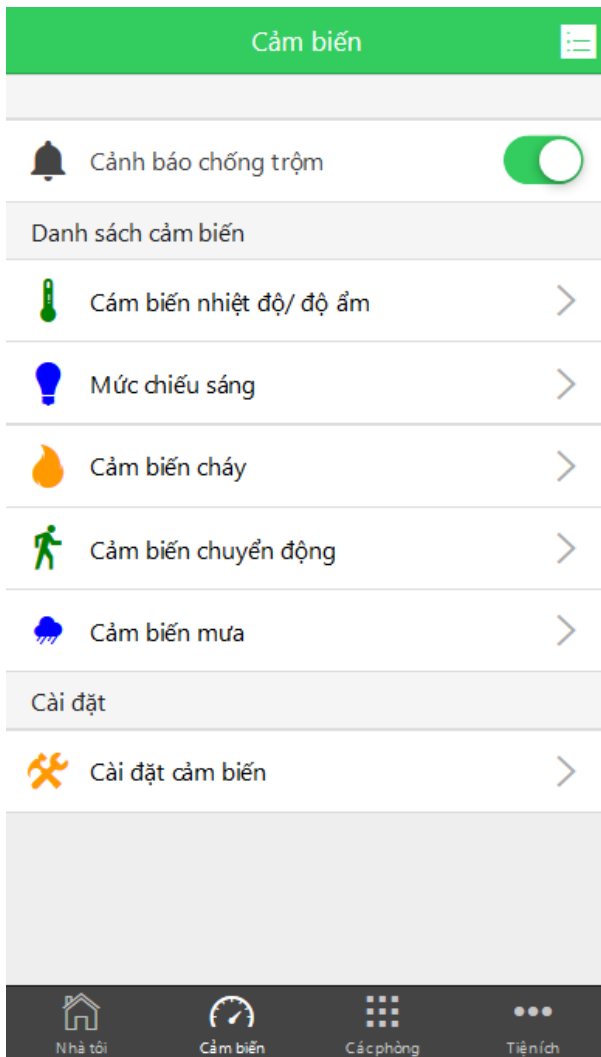
Giao diện màn hình chính



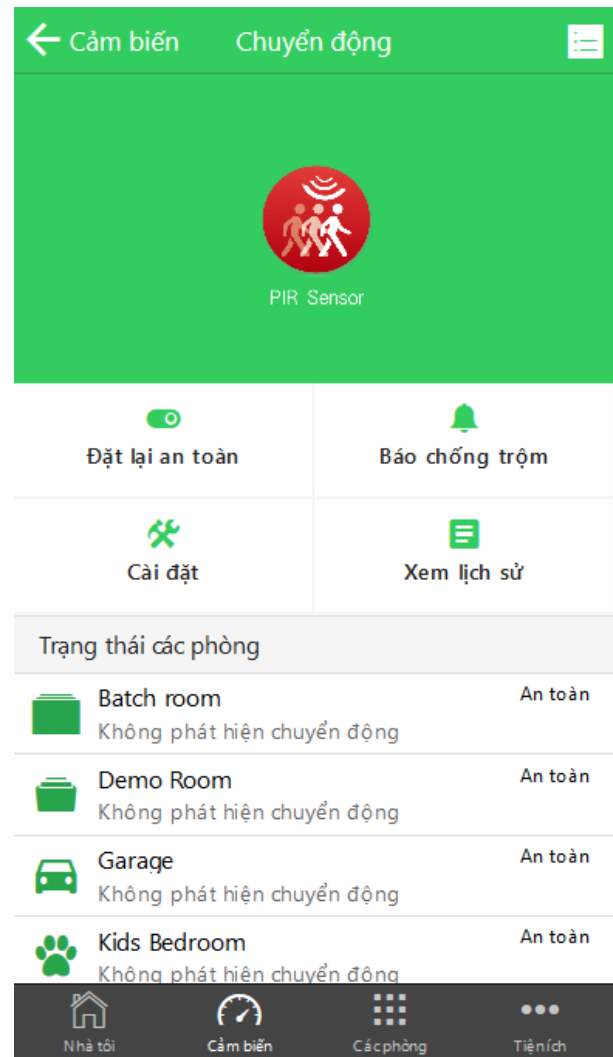
Giao diện điều khiển



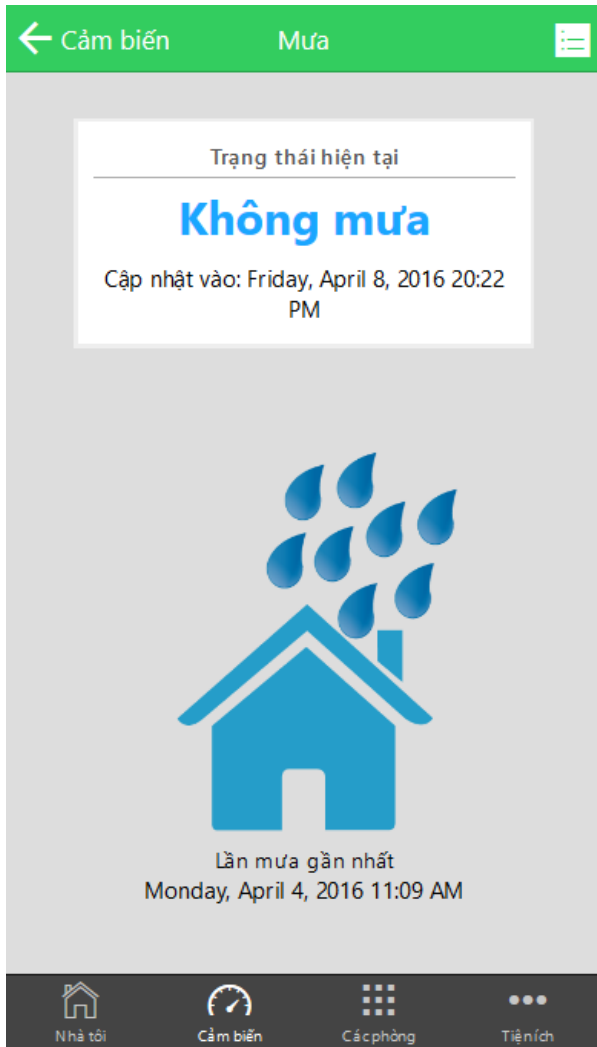
Các tùy chọn hệ thống



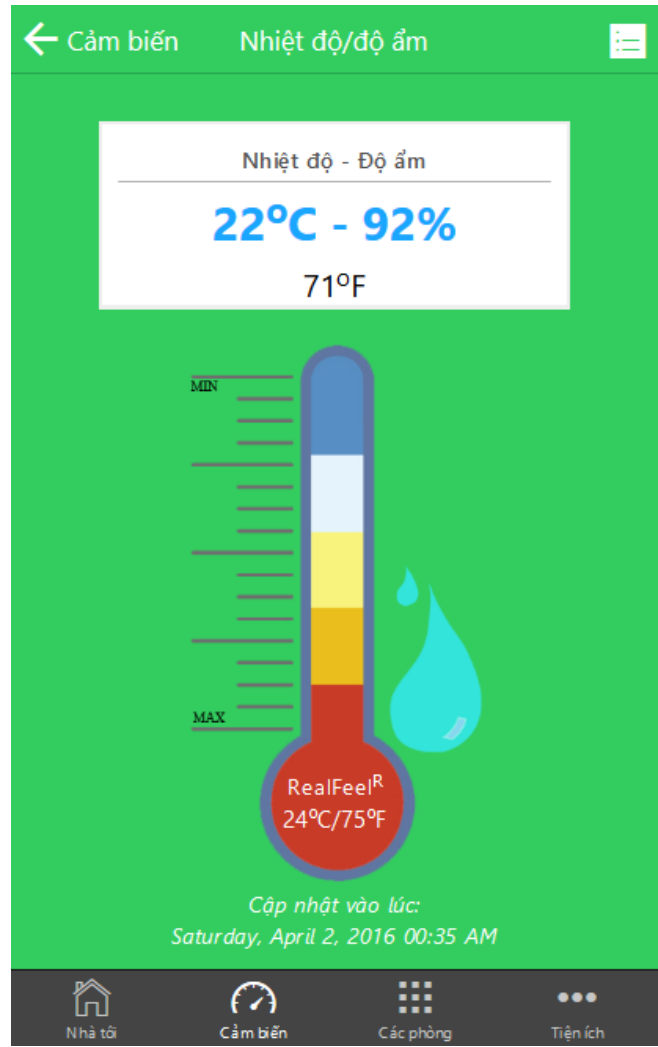
Các cảm biến



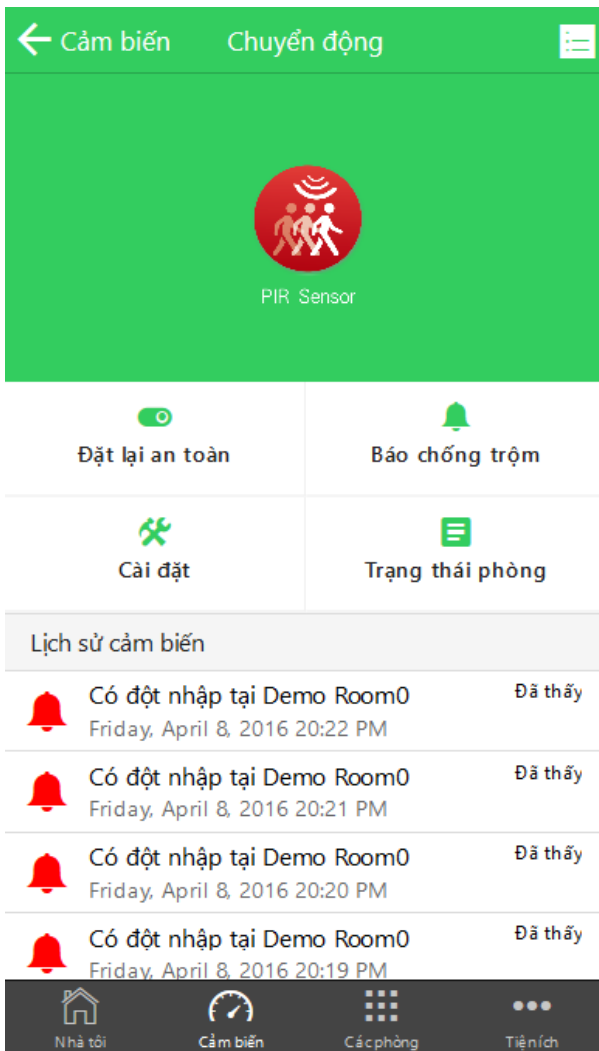
Cảm biến chuyển động



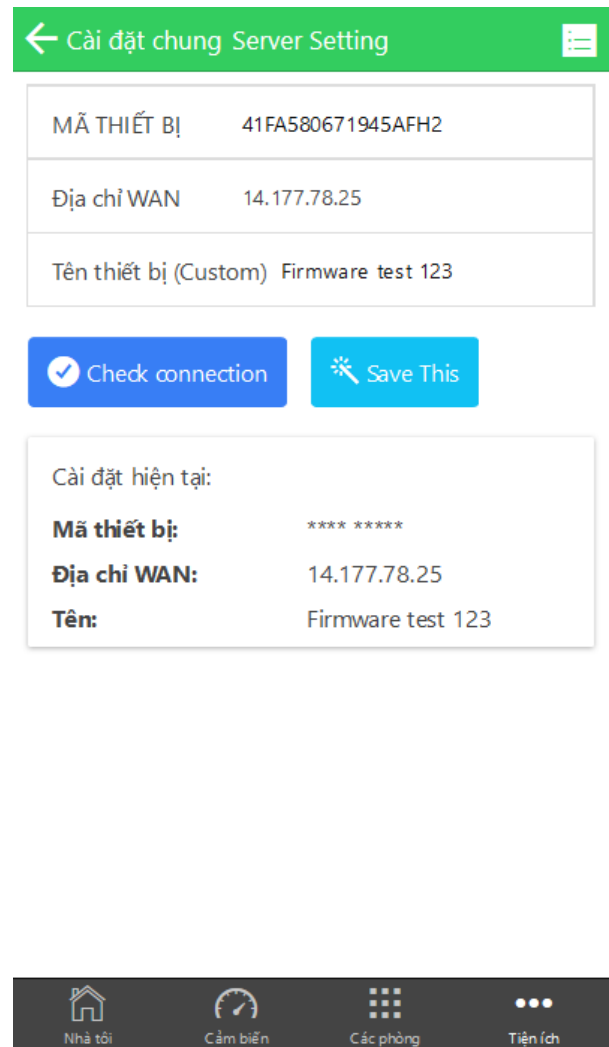
Cảm biến mưa



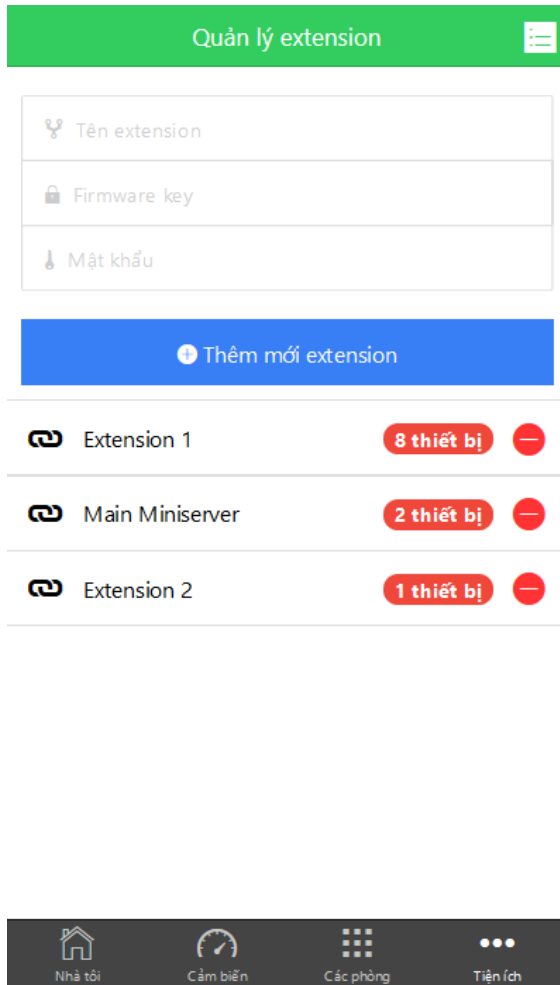
Cảm biến nhiệt độ/độ ẩm



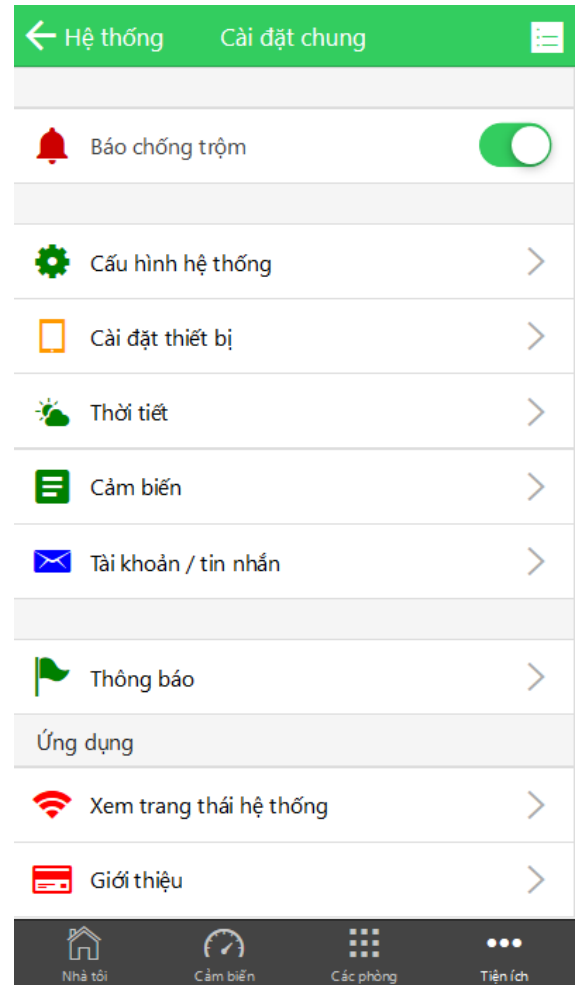
Lịch sử hệ thống



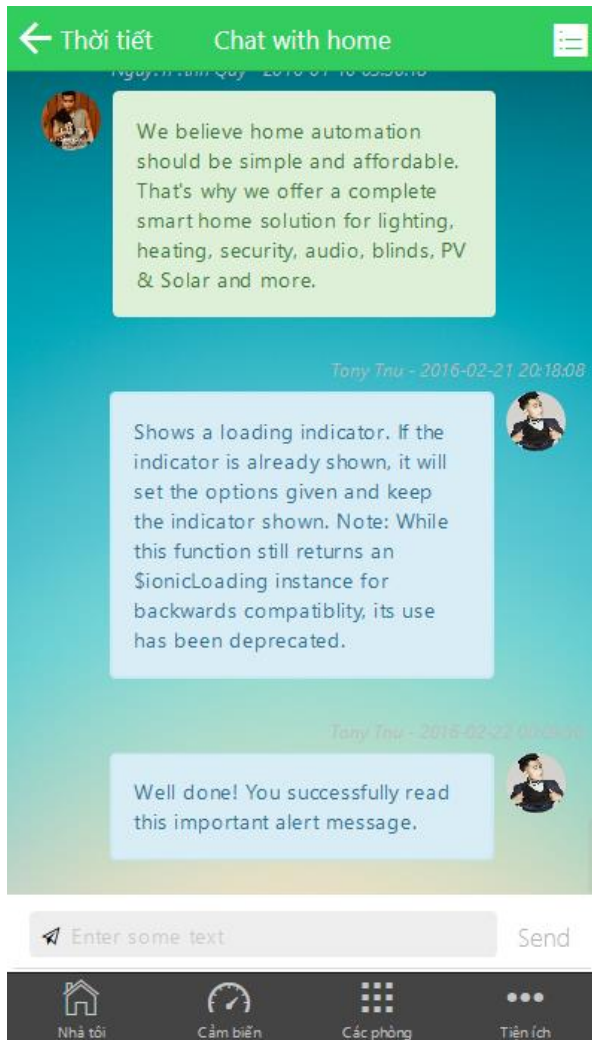
Các cấu hình hệ thống



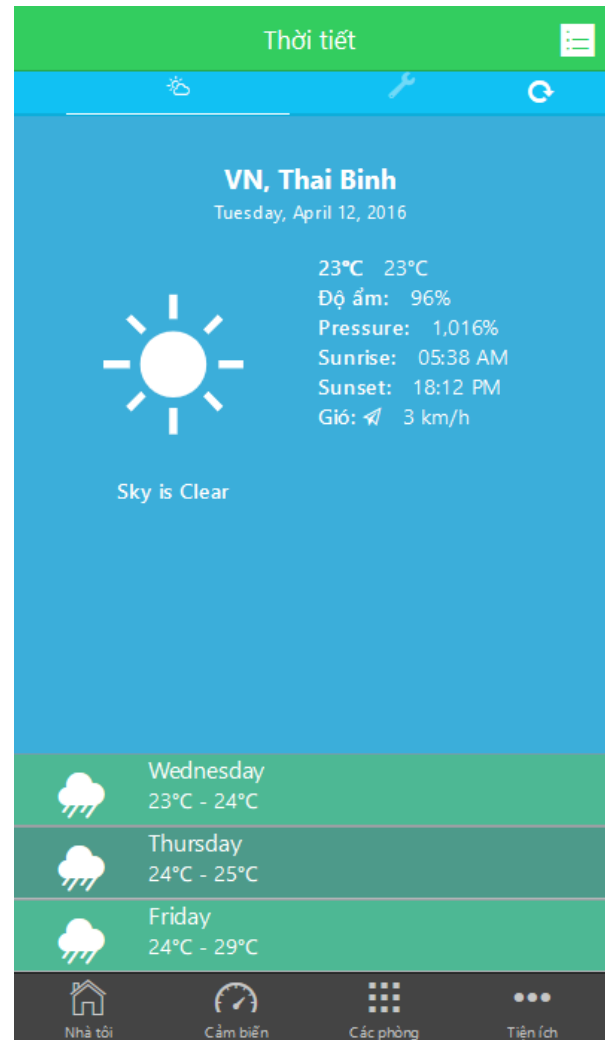
Giao diện điều khiển



Các thiết lập người dùng



Ứng dụng nhắn tin trong gia đình



Xem tình hình thời tiết

- HẾT -