**TÓM TẮT**

Tên đề tài: Thiết kế và thi công hệ thống giám sát và điều khiển không dây các thiết bị trong nhà sử dụng board Raspberry Pi 3

Sinh viên thực hiện: Trần Thanh An

Số thẻ SV: 106120005 Lớp: 12DT1

Sinh viên thực hiện: Lê Xuân Bắc

Số thẻ SV: 106120010 Lớp: 12DT1

Sinh viên thực hiện: Phạm Phú Quỳnh

Số thẻ SV: 106120190 Lớp: 12DT4

Đây là quyển báo cáo đồ án tốt nghiệp của chúng tôi với đề tài: “Thiết kế và thi công hệ thống giám sát và điều khiển không dây các thiết bị trong nhà sử dụng board Raspberry Pi 3”

Được chia làm 3 chương lớn:

Chương 1: Đặt vấn đề

Chương này sẽ trình bày về các vấn đề tên đề tài, phạm vi sử dụng và phát triển của đề tài, số lượng sinh viên thực hiện, và công nghệ dự kiến sẽ sử dụng trong đề tài.

Chương 2: Cơ sở lý thuyết

Trong chương này chúng tôi sẽ trình bày về xu thế công nghệ Internet of Thing tại Việt Nam và trên Thế giới. Các công cụ, giao thức, ngôn ngữ cũng như các thiết bị mà chúng tôi sử dụng trong đồ án.

Chương 3:Kết quả và hướng phát triển đề tài

Trong chương này chúng tôi sẽ trình bày về kết quả trong suốt quá trình thực hiện đề tài, các lỗi gặp phải và cách xử lý trong quá trình thi công và hoàn thiện đồ án.

Và phần cuối cùng của quyển đồ án là hướng phát triển của đề tài trong tương lai

# LỜI CAM ĐOAN

Kính gửi: Hội đồng bảo vệ đồ án tốt nghiệp khoa Điện tử - Viễn thông, Trường Đại học Bách Khoa-Đại học Đà Nẵng

Chúng tôi là Trần Thanh An lớp 12DT1, Lê Xuân Bắc lớp 12DT1 Phạm Phú Quỳnh lớp 12DT4 khoa Điện tử-Viễn thông, trường Đại học Bách Khoa –Đại học Đà Nẵng.

Chúng tôi xin cam đoan nội dung của Đồ án này không phải là bản sao chép của bất cứ Đồ án hoặc Công trình đã có từ trước. Nếu vi phạm tôi xin chịu mọi hình thức kỷ luật của Khoa.

# Đà Nẵng, tháng 05 năm 2017

# Sinh viên thực hiện

# 

# Trần Thanh An

# Lê Xuân Bắc

# Phạm Phú Quỳnh

**MỤC LỤC**

**DANH SÁCH CÁC BẢNG, HÌNH VẼ**

Bảng 3.1 Bảng mô tả các messages được sử dụng trong hệ thống

Hình 1.1 Mô hình tổng quan về hệ thống IoT

Hình 2.1 Sơ đồ chân STM32F103C8T6

Hình 2.2 Sơ đồ chân nRF24L01

Hình 2.3 Sơ đồ chân DHT11

Hình 2.4 Module cảm biến chuyển động PIR

Hình 2.5 Module cảm biến rung HDX-2

Hình 2.6 Module cảm biến cửa từ

Hình 2.7 Module relay 5V

Hình 2.8 Board mạch Raspberry Pi 3

Hình 2.9 Màn hình cảm ứng

Hình 2.10 Cấu trúc cơ sở của Linux

Hình 2.11 Cấu trúc mạng Star

Hình 2.12 Cấu trúc mạng Tree

Hình 2.13 Mô hình MQTT

Hình 2.14 Mối liên hệ giữa publisher và subscriber.

Hình 2.15 Sơ đồ publish/subscribe topic

Hình 3.1 Mô hình khối điều khiển trung tâm

Hình 3.2 Sơ đồ khối điều khiển thiết bị đầu cuối

Hình 3.3 Sơ đồ khối hệ thống

Hình 3.4 Sơ đồ khối điều khiển trung tâm

Hình 3.5 Sơ đồ khối điều khiển thiết bị đầu cuối

Hình 3.6 Usecase diagram của hệ thống

Hình 3.7 Sequence diagram đăng ký thiết bị lên Gateway

Hình 3.8 Sequence diagram khiển thiết bị đầu cuối

Hình 3.9 Sequence diagram khởi tạo hệ thống

Hình 3.10 Cấu trúc mạng RF24Network

Hình 3.11 Sequence diagram điều khiển thiết bị output

Hình 3.12 Sequence diagram điều khiển thiết bị input

Hình 3.13 Giao diện đăng nhập của hệ thống

Hình 3.14 Giao diện đăng ký và điều khiển thiết bị đầu cuối

**DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

KÝ HIỆU:

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

CHỮ VIẾT TẮT:

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

*Ghi chú:*

* Ký hiệu: mỗi mục ký hiệu gồm ký hiệu và phần tên gọi, diễn giải ký hiệu.
* Cụm từ viết viết tắt là các chữ cái và các ký hiệu thay chữ được viết liền nhau, để thay cho một cụm từ có nghĩa, thường được lặp nhiều lần trong đồ án.

**MỞ ĐẦU**

Ngày nay trên thế giới nói chung và ở Việt Nam nói riêng, với sự bùng nổ của công nghệ thông tin và điện tử, cùng với đó là chúng ta đang bước vào cuộc cách mạng công nghiệp 4.0, các thiết bị thông minh ngày càng ảnh hưởng đến đời sống sinh hoạt hằng ngày của con người. Trên các phương tiện thông tin đại chúng, chúng ta từng nghe nhiều về thuật ngữ nhà thông minh, Internet of things (IoT). Là một sinh viên khoa Điện tử- Viễn thông của trường Đại học Bách Khoa - Đại Học Đà Nẵng, với những kiến thức đã học tập và tìm tòi, cùng với việc muốn thiết kế ra hệ thống điều khiển từ xa các thiết bị điện trong nhà thông qua điện thoại Android và hiển thị lên Gateway trong ngôi nhà. Khi dự án hoàn thành chúng ta có thể điều khiển các thiết bị điện trong nhà thông qua mạng Internet bao gồm 3G, Wifi. Chúng ta có thể kiểm soát và điều chỉnh trạng thái hoạt động của các thiết bị thông qua Gateway hoặc Smartphone. Toàn bộ thông tin về người sử dụng như tài khoản quản lý, thông tin gateway được lưu trữ và xử lý trên server chung. Dù chúng ta ở bất cứ đâu, chỉ cần có mạng internet ta đều có thể điều khiển và kiểm soát được các thiết bị trong nhà.

Xuất phát từ ý tưởng trên, tôi đã chọn đề tài “THIẾT KẾ VÀ THI CÔNG HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN KHÔNG DÂY CÁC THIẾT BỊ TRONG NHÀ SỬ DỤNG BOARD RASBERRY PI 3”. Nội dung đồ án gồm có 3 chương:

Chương 1: Đặt vấn đề

Chương 2: Cơ sở lý thuyết

Chương 3: Kết quả và hướng phát triển đề tài

Phương pháp nghiên cứu thực hiện đề tài là phân tích và thiết kế các chức năng của hệ thống, đưa ra các trường hợp xấu nhất có thể xảy ra đối với hệ thống , sau đó sẽ tìm hướng giải quyết tốt nhất. Kiểm tra hoạt động của hệ thống trong các trường hợp.

## *Số lượng sinh viên thực hiện*

Nhóm thực hiện đề tài đồ án chúng tôi gồm 5 thành viên được chia làm 3 nhóm:

Phần Device, có nhiệm vụ giao tiếp và điều khiển các thiết bị trong nhà bằng vi điều khiển STM32F103C8T6 do bạn Nguyễn Khương Nghĩa thực hiện

Phần Gateway, có nhiệm vụ điều khiển, xử lý các thông tin của thiết bị, giao tiếp với server và smartphone, sử dụng board Raspberry pi 3 do 3 bạn Phạm Phú Quỳnh, Trần Thanh An và Lê Xuân Bắc thực hiện

Phần Server có nhiệm vụ quản lý các thông tin người dùng, tạo kết nối cho gateway và smartphone, do bạn Lê Thanh Sang thực hiện.

**Phân chia nhiệm vụ:**

Trần Thanh An: Thiết kế sequence các tiến trình hoạt động của hệ thốngLiên kết lớp giao diện và lớp điều khiển phần cứng

Lê Xuân Bắc:Thiết kế lớp giao diện người dùng+ Kiểm tra hoạt động của hệ thống

Phạm Phú Quỳnh: Lập trình các tiến trình hệ thống dựa theo sequence+ Kiểm tra hoạt động của hệ thống

Tuy đã cố gắng rất nhiều nhưng không thể tránh khỏi những thiếu sót. Rất mong sự góp ý của thầy cô và các bạn để đồ án này được hoàn thiện hơn.

Tôi xin chân thành cảm ơn quý thầy cô trong Khoa Điện tử-Viễn Thông, đặc biệt là ThS. Hồ Viết Việt đã tận tình hướng dẫn để tôi có thể hoàn thành tốt đồ án này.

Đà Nẵng, tháng 06, 2017

Nhận xét của người hướng dẫn Sinh viên thực hiện

TS. Phan Trần Đăng Khoa Lê Xuân Bắc

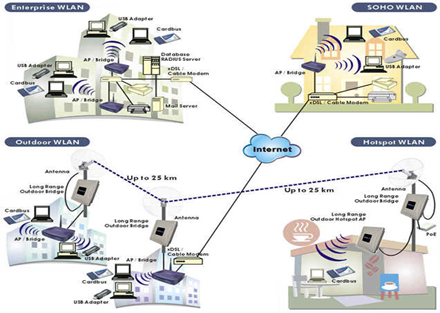
# CHƯƠNG 1: TÌM HIỂU VỀ IOT VÀ TỔNG QUAN ĐỀ TÀI

## Giới thiệu chương

Chương này sẽ giới thiệu về xu thế Internet of Things trình bày về các vấn đề tên đề tài, phạm vi sử dụng và phát triển của đề tài, số lượng sinh viên thực hiện, và công nghệ dự kiến sẽ sử dụng trong đề tài.

## Giới thiệu về Internet of Things

## *Định nghĩa*



Hình 1.1 Mô hình tổng quan về hệ thống IoT

Internet of Things (IoT) là khi tất cả mọi thứ đều được kết nối với nhau qua mạng Internet, người dùng (chủ) có thể kiểm soát mọi đồ vật của mình qua mạng mà chỉ bằng một thiết bị thông minh, chẳng hạn như smartphone, tablet, PC

Từ vài năm nay, "Internet của vạn vật" ( Internet of Things - IoT ) là một trong những yếu tố quan trọng của mọi dự báo về công nghệ tương lai. Thực ra, IoT đã hình thành trong hiện tại và đang là động lực của mọi thành tựu công nghệ.

Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng (identifiable) cũng như chỉ sự tồn tại của chúng trong một kiến trúc mang tính kết nối. Nếu mọi đối tượng, kể cả con người, được "đánh dấu" để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lí được nó thông qua máy tính. Việc đánh dấu (tagging) có thể được thực hiện thông qua nhiều công nghệ, chẳng hạn như RFID, NFC, mã vạch, mã QR, watermark kĩ thuật số... Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại, RF...[2]

## *Kiến trúc và tính chất của The Internet of Things*

Kiến trúc dựa trên sự kiện:

Các thực thể, máy móc trong IoT sẽ phản hồi dựa theo các sự kiện diễn ra trong lúc chúng hoạt động theo thời gian thực. Một số nhà nghiên cứu từng nói rằng một mạng lưới các sensor chính là một thành phần đơn giản của IoT.

Là một hệ thống phức tạp:

Trong một thế giới mở, IoT sẽ mang tính chất phức tạp bởi nó bao gồm một lượng lớn các đường liên kết giữa những thiết bị, máy móc, dịch vụ với nhau, ngoài ra còn bởi khả năng thêm vào các nhân tốc mới.[2]

## *Ứng dụng*

IoT có ứng dụng vô cùng rộng rãi, có thể kể ra như sau:

* Quản lí chất thải
* Quản lí và lập kế hoạch quản lí đô thị
* Quản lí môi trường
* Phản hồi trong các tinh huống khẩn cấp
* Mua sắm thông minh
* Quản lí các thiết bị cá nhân
* Đồng hồ đo thông minh
* Tự động hóa ngôi nhà

### Lý do chọn đề tài

Hiện nay, cuộc cách mạng công nghiệp 4.0 và Internet of Things đang phát triển mạnh. Các công ty lớn như Facebook, BKAV,... đang tập trung mạnh vào mảng IOT, Big Data, AI,...

Đi cùng với xu hướng công nghệ này thì tập đoàn FPT nói chung và Fsoft Đà Nẵng nói riêng đang tập trung nghiên cứu và đẩy mạnh phát triển về Internet Of Things.

Bản thân nhóm chúng tôi là những người đầu tiên đang nghiên cứu và phát triển hệ thống Internet Of Things tại Fsoft Đà Nẵng với đề tài “THIẾT KẾ VÀ THI CÔNG HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN KHÔNG DÂY CÁC THIẾT BỊ TRONG NHÀ SỬ DỤNG RASPBERRY PI 3”

Lịch sử phát triển

Tại Việt Nam, công ty BKAV đã có một sản phẩm về điều khiển thiết bị trong nhà từ xa. Nhưng hệ thống đó có một số nhược điểm:

Khi ta muốn thêm thiết bị mới vào hệ thống thì ta phải sử dụng đúng thiết bị của BKAV cung cấp thì thiết bị đó mới có thể kết nối vào toàn bộ hệ thống, thêm vào đó toàn bộ hệ thống Bkav SmartHome sử dụng công nghệ không dây ZigBee, những module hỗ trợ công nghệ này có giá thành không rẻ.

Do đó những thiết bị do BKAV cung cấp có giá thành khá cao so với giá thành của các thiết bị cùng loại trên thị trường, và không hỗ trợ mở rộng kết nối với các thiết bị khác.

### Phạm vi phát triển

Đầu tiên hệ thống này sẽ được ứng dụng vào các hệ thống căn hộ tại hệ thống “Khu đô thị công nghệ cao FPT”.

Và trong tương lai chúng tôi hy vọng hệ thống này sẽ được ứng dụng rộng rãi tại Việt Nam và rộng ra thế giới.

## *Công nghệ dự kiến*

Trong đề tài này nhóm đã sử dụng:

* Phần cứng:
* Board mạch Rasperberry Pi 3
* Board STM32F103C8T6
* Module nRF24L01+
* Relay 5V
* Màn hình cảm ứng
* Phần mềm và môi trường phát triển:
* Hệ điều hành Raspbian Jessie
* Framework Qt 5.3 trên Linux
* IDE QtCreator
* ARM tool chain cho Linux
* Eclipse cho Linux
* Android Studio
* My SQL
* Ngôn ngữ lập trình C, C++11, Java

## Kết luận

Chương này chúng tôi đã trình bày tổng quan về sự cần thiết của đề tài, phân chia được nhiệm vụ công việc của từng thành viên và quan trọng nhất là những công nghệ mà chúng tôi trực tiếp sử dụng trong đề tài này.

Chương tiếp theo chúng tôi sẽ trình bày sơ lược về cơ sở lý thuyết của các công nghệ và chính là nền tảng để chúng tôi thực hiện đề tài này.

## CHƯƠNG 2: MÔI TRƯỜNG, CÔNG CỤ VÀ THIẾT BỊ TRONG HỆ THỐNG

## Giới thiệu chương

Giới thiệu về Framework Qt 5.8, giao thức MQTT, kiểu dữ liệu JSON, cách thức lập trình đa luồng trên linux, MySQL, các ngôn ngữ lập trình,cũng như các phần cứng mà chúng tôi sử dụng trong đồ án.

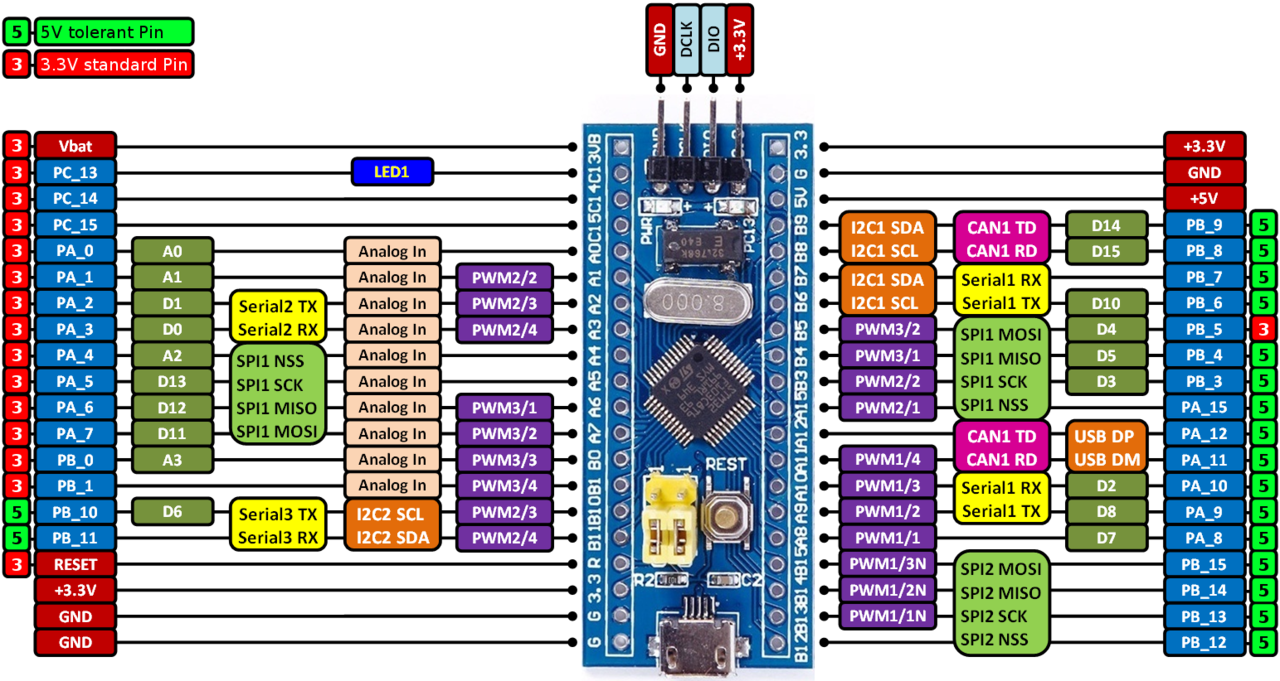
## Thiết bị sử dụng

## *Device*

## *Vi điều khiển STM32F103C8T6*

## *Giới thiệu*

Vi điều khiển STM32F103C8T6 là họ vi điều khiển 32bit của hãng Texas Instrument



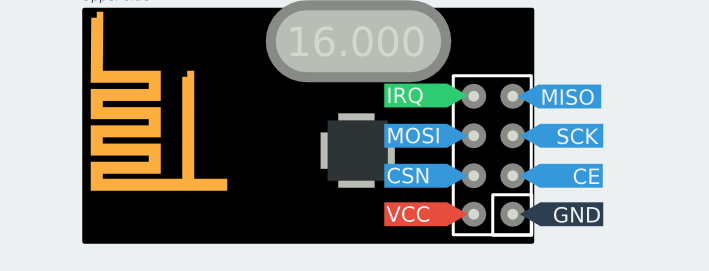
Hình 2.1 Sơ đồ chân STM32F103C8T6

## *Thông số kỹ thuật STM32F103C8T6*

Phụ lục 1

## *Module nRF24L01*

## *Giới thiệu*

****

Hình 2.2 Sơ đồ chân nRF24L01

Module nRF24L01 hoạt động ở tần số song ngắn 2.4G nên module này có khả năng truyền dữ liệu tốc độ cao và truyền nhận dữ liệu trong điều kiện môi trường có vật cản.

Module nRF24L01 có 126 kênh truyền. Vì vậy ta có thể truyền nhận dữ liệu trên nhiều kênh khác nhau trong cùng một thời điểm.

Module này có khả năng thay đổi công suất phát bằng chương trình, điều này giúp nó có thể hoạt động trong chế độ tiết kiệm năng lượng.

Chú ý: Điện áp thường cung cấp là 3.3V. Nhưng các chân IO tương thích với chuẩn 5V. Điều này giúp nó giao tiếp rộng dãi với các dòng vi điều khiển.

## *Thông số kỹ thuật nRF24L01*

Phụ lục 1

## *Module cảm biến nhiệt độ, độ ẩm DHT11*

## *Giới thiệu*

****

Hình 2.3 Sơ đồ chân DHT11

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.

Chuẩn này cung cấp cả dữ liệu tốc độ thấp, truyền tín hiệu, và nguồn nuôi qua cùng một chân tín hiệu đơn. 1-Wire cũng tương tự như [I²C](https://vi.wikipedia.org/wiki/I%C2%B2C), nhưng với tốc độ truyền dữ liệu thấp và khoảng cách xa hơn. Nó thường được sử dụng để giao tiếp với các [thiết bị](https://vi.wikipedia.org/wiki/Linh_ki%E1%BB%87n_%C4%91i%E1%BB%87n_t%E1%BB%AD) nhỏ giá rẻ như [nhiệt kế](https://vi.wikipedia.org/wiki/Nhi%E1%BB%87t_k%E1%BA%BF) kĩ thuật số và [công cụ đo thời tiết](https://vi.wikipedia.org/w/index.php?title=Weather_instruments&action=edit&redlink=1).

## *Thông số kỹ thuật Module cảm biến nhiệt độ, độ ẩm DHT11*

Phụ lục 1

## *Module cảm biến chuyển động PIR*

## *Giới thiệu*



Hình 2.4 Module cảm biến chuyển động PIR

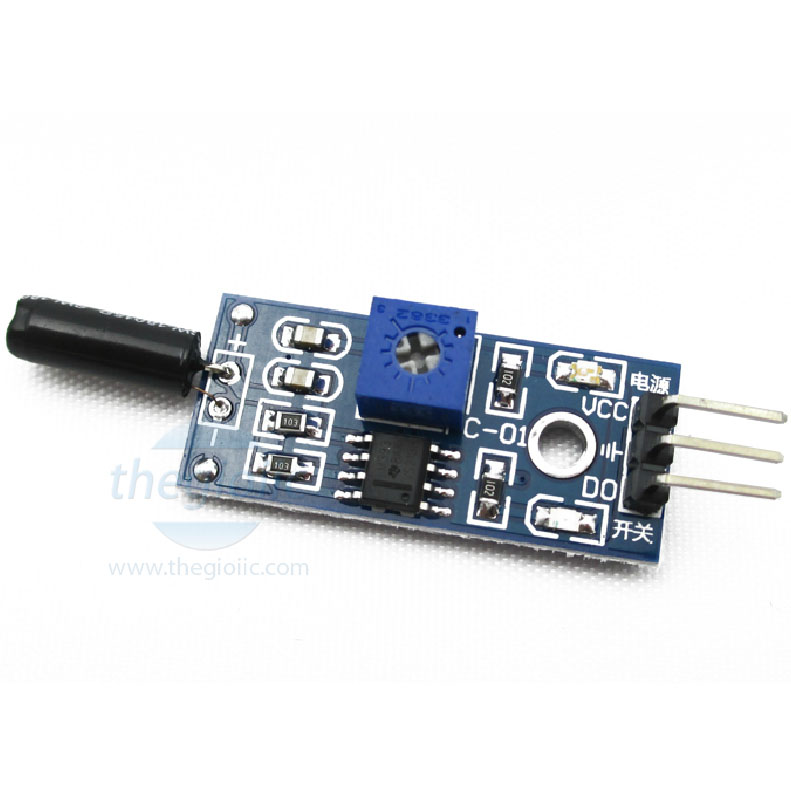
Cảm biến PIR (Passive Infra­Red ) phát hiện chuyển động bằng cách đo những thay đổi trong bức xạ hồng ngoại phát ra bởi các đối tượng . Khi phát hiện chuyển động cảm biến PIR sẽ xuất ra 1 xung ở mức cao, xung này được đọc bởi một vi điều khiển để thực hiện chức năng mong muốn.

## *Thông số kỹ thuật Module cảm biến chuyển động PIR*

## *Phụ lục 1*

## *Module cảm biến rung HDX-2*

## *Giới thiệu*



Hình 2.5 Module cảm biến rung HDX-2

Được sử dụng trong một loạt các hiệu ứng rung động kích hoạt báo động trộm cắp, xe thông minh, báo động động đất, báo động xe gắn máy.

## *Đặc tính kỹ thuật Module cảm biến rung HDX-2*

Phụ lục 1

## *Module cảm biến cửa từ*

## *Giới thiệu*

****

Hình 2.6 Module cảm biến cửa từ

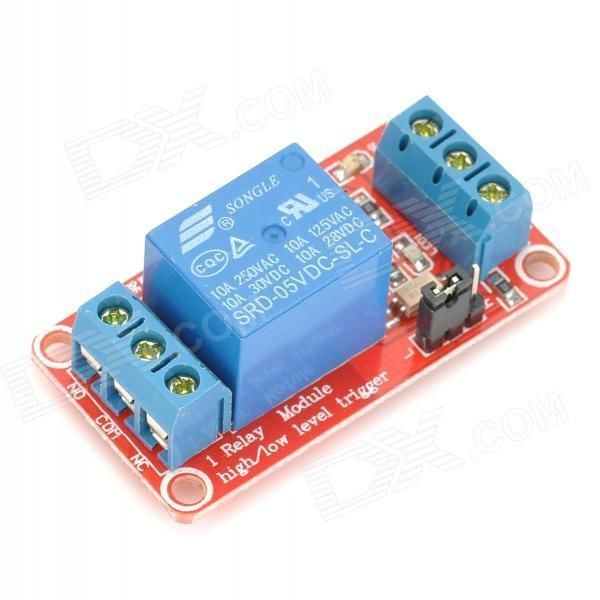
Cảm biến cửa từ sẽ nhanh chóng phát hiện bất kỳ cửa sổ mở hoặc cửa ra vào trong nhà thông minh.

## *Thông số kỹ thuật Module cảm biến cửa từ*

Phụ lục 1

## *Module Relay*

## *Giới thiệu*

****

Hình 2.7 Module relay 5V

Module 1 Relay với opto cách ly nhỏ gọn, có opto và transistor cách ly giúp cho việc sử dụng trở nên an toàn với board mạch chính, mạch được sử dụng để đóng ngắt nguồn điện công suất cao AC hoặc DC, có thể chọn đóng khi kích mức cao hoặc mức thấp bằng Jumper.

## *Thông số kỹ thuật Module Relay*

Phụ lục 1

## Gateway

## Phần cứng

Đối với phần Gateway, nhóm tôi sử dụng bo mạch Raspberry Pi 3 kết nối với module RF nRF24L01+ để điều khiển thiết bị đầu cuối qua màn hình Touch.

## Raspberry Pi 3



Hình 2.8 Board mạch Raspberry Pi 3

## Giới thiệu

Đặc tính của Raspberry Pi xây dựng xoay quanh bộ xử lí SoC Broadcom BCM2835 ( là chip xử lí mobile mạnh mẽ có kích thước nhỏ hay được dùng trong điện thoại di động ) bao gồm CPU , GPU , bộ xử lí âm thanh /video , và các tính năng khác … tất cả được tích hợp bên trong chip có điện năng thấp này.

Raspberry Pi không thay thế hoàn toàn hệ thống để bàn hoặc máy xách tay . Ta không thể chạy Windows trên đó vì BCM2835 dựa trên cấu trúc ARM nên không hỗ trợ mã x86/x64 , nhưng vẫn có thể chạy bằng Linux với các tiện ích như lướt web , môi trường Desktop và các nhiệm vụ khác . Tuy nhiên Raspberry Pi là một thiết bị đa năng đáng ngạc nhiên với nhiều phần cứng có giá thành rẻ nhưng rất hoàn hảo cho những hệ thống điện tử , những dự án DIY , thiết lập hệ thống tính toán rẻ tiền cho những bài học trải nghiệm lập trình …

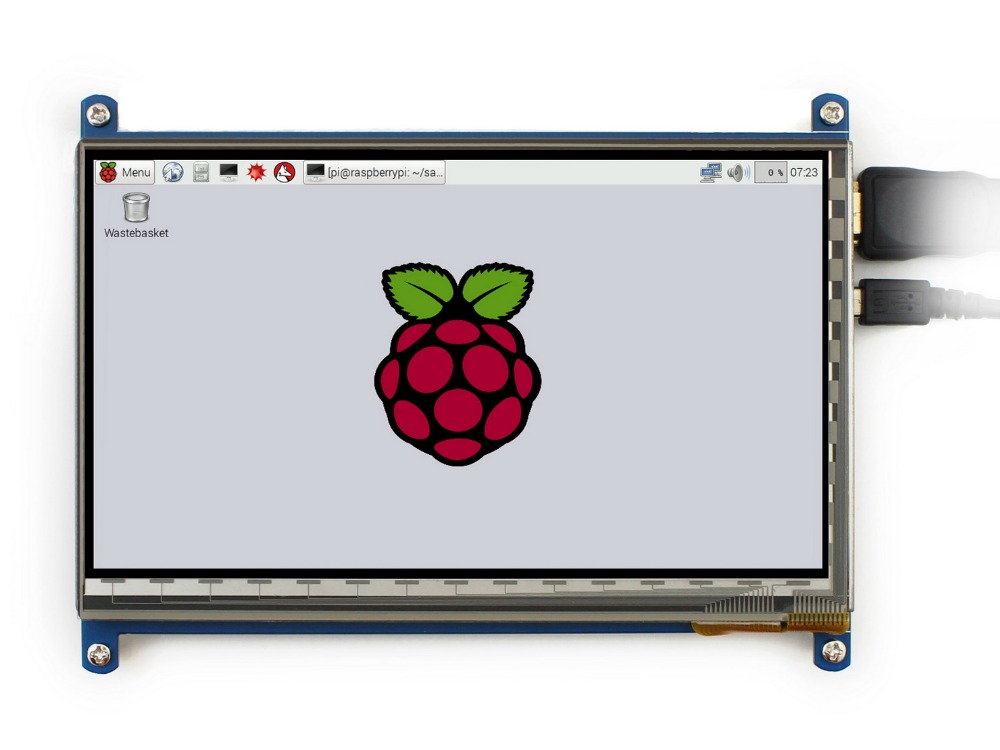
## Thông số kĩ thuật Raspberry Pi 3

Phụ lục 1

## Màn hình cảm ứng

## Giới thiệu

Màn hình 800 x 480 với hỗ trợ cảm ứng đa điểm, kết nối thông qua cáp HDMI

****

Hình 2.9 Màn hình cảm ứng

## Thông số kĩ thuật Màn hình cảm ứng

Phụ lục 1

## Môi trường và công cụ

## Hệ điều hành Raspbian Jessie

Đây là hệ điều hành cơ bản, phổ biến nhất và do chính Raspberry Pi Foundation cung cấp. Raspbian được hướng đến người dùng có mục đích:

Sử dụng Raspberry Pi như máy tính văn phòng để lướt web, soạn văn bản, check mail và thi thoảng nghe nhạc/xtôi phim.

Nghiên cứu phát triển các thiết bị điều khiển tự động.

Sử dụng như một máy chủ cung cấp các dịch vụ như web, file server, printer server, ..

Raspbian hoạt động rất ổn định, tốc độ nhanh và nó dựa trên nền Debian (Gần giống ubuntu) với giao diện LXDE (thay vì GNOME).

Nhược điểm của nó là giao diện đơn giản, cổ điển. Nếu ta không quá quan tâm tới giao diện mà hướng đến hiệu năng thì Raspbian rất phù hợp.

## Framework Qt 5.3

Trong hệ thống thì chúng tôi sử dụng Qt 5.3 với ngôn ngữ C/C++ để tạo giao diện ngươi dùng điều khiển thiết bị tại Gateway, giúp điều khiển một cách trực quan và dễ dàng.

## Giới thiệu

Qt là một Application Framework. Mục tiêu của các nhà phát triển nên Qt là tạo ra một framework có khả năng thiết kế những phần mềm có thể chạy trên nhiều nền tảng phần mềm lẫn phần cứng khác nhau mà không phải thay đổi nhiều về code. Qt không chỉ là thứ giúp ta viết giao diện cho phần mềm của mình, nó có đầy đủ các khía cạnh để tạo nên một phần mềm hoàn chỉnh ở nhiều góc độ, cho dù phần mềm đó có giao diện hay không. Ta có thể dùng Qt viết ra những phần mềm chạy bằng dòng lệnh, hoặc là các ứng dụng console chạy trên server, thậm chí là các web framework.

## Một số nền tảng mà Qt hỗ trợ

* Windows
* Linux
* OS X
* Android
* iOS
* WinRT (Windows 8/8.1 và Windows Phone 8/8.1)
* Blackberry 10
* Sắp tới sẽ hỗ trợ thêm Tizen, hệ điều hành dựa trên Linux cho nhiều loại thiết bị.
* Các hệ điều hành nhúng như: Android/Linux/Windows Tôibedded

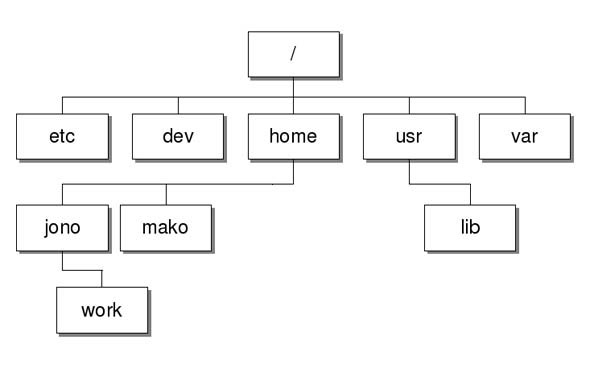
## Một số thành phần của Qt

* Qt Core: Chứa các thành phần cơ sở, bao gồm các containers (tương tự thư viện chuẩn của các ngôn ngữ lập trình: QVector, QList, QMap,…), các thư viện nhập xuất đa nền tảng, lập trình đa luồng (threading) và xử lý song song (concurrency),….
* Qt GUI: Thành phần chính để lập trình giao diện.
* Qt Widgets: Các widget, button, hộp thoại và những thứ tương tự trong giao diện đều gọi là các widget.

|  |  |
| --- | --- |
|  |  |

## Linux

 Linux là một nền tảng cho phép các ứng dụng cũng như người điều hành máy tính truy cập vào các thiết bị trên máy tính để thực hiện những chức năng mong muốn. Ví dụ như hệ điều hành (OS) sẽ chuyển tiếp những hướng dẫn từ một ứng dụng tới bộ xử lí của máy tính. Bộ xử lí này sẽ thực hiện các nhiệm vụ đã được hướng dẫn sau đó gửi lại kết quả cho ứng dụng thông qua hệ điều hành.



Hình 2.10 Cấu trúc cơ sở của Linux

Cấu trúc cơ sở của Linux khác hoàn toàn so với Windows. Nó được phát triển trên một mã gốc riêng với các nhà phát triển riêng rẽ. Ta sẽ không tìm thấy thư mục My Documents trên Ubuntu hay Program Files trên Fedora. Cũng không có các ổ đĩa C: hay D: xuất hiện. Thay vào đó, có một cây dữ liệu và các ổ đĩa được bung vào cây đó. Tương tự, thư mục home và desktop đều là một phần trong cây dữ liệu. Về mặt kỹ thuật, bạn sẽ cần tìm hiểu một hệ thống và kiến trúc file mới hoàn toàn. Thực tế thì việc này không quá khó nhưng sự khác biệt vẫn là rõ rệt.

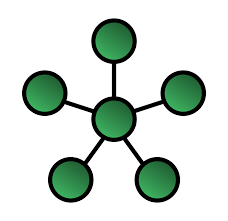
Lệnh đầu cuối

Linux là hệ điều hành dành sử dụng giao diện dòng lệnh (terminal) để ta có thể sử dụng các lệnh thực thi nên nó rất mạnh mẽ và hiệu quả. Nói cách khác, nó giống như Command Prompt của Windows.

## Cấu trúc liên kết mạng

Các cấu trúc liên kết mạng mô tả các phương pháp được sử dụng làm hệ thống dây điện vật lý của mạng.Trong đề tài thì chúng tôi sử dụng cấu trúc mạng Star kết hợp với cấu trúc mạng Tree để điều khiển các node.

## Cấu trúc mạng STAR



Hình 2.11 Cấu trúc mạng Star

Mạng dạng hình sao bao gồm một trung tâm và các nút thông tin.Các nút thông tin là các trạm đầu cuối, các máy tính và các thiết bị khác của mạng. Trung tâm của mạng điều phối mọi hoạt động trong mạng với các chức năng cơ bản là:

* Xác định cặp địa chỉ gửi và nhận được phép chiếm tuyến thông tin và liên lạc với nhau.
* Cho phép theo dõi và xử lý sai trong quá trình trao đổi thông tin.
* Thông báo các trạng thái của mạng.

Ưu điểm:

* Hoạt động theo nguyên lý nối song song nên nếu có một thiết bị nào đó ở một nút thông tin bị hỏng thì mạng vẫn hoạt động bình thường.
* Cấu trúc mạng đơn giản và các thuật toán điều khiển ổn định.
* Cho tốc độ nhanh vì tận dụng được tối đa tốc độ truyền của đường truyền vật lý.
* Việc thiết lập mạng đơn giản, dễ dàng cấu hình lại mạng (thêm, bớt các trạm) và có thể kiểm soát và khắc phục sự cố nhanh.

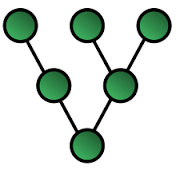
Nhượcđiểm:

* Mạng Star cung cấp tài nguyên và chế độ quản lý tập trung. Tuy nhiên, do mỗi máy tính nối vào một trung tâm điểm, nên cấu hình này cần rất nhiều cáp nếu cài đặt mạng ở quy mô lớn.
* Nếu trung tâm bị hỏng thì toàn bộ mạng cũng bị đứt.

## Cấu trúc mạng Tree

Cấu trúc mạng Tree cho phép mở rộng một mạng lưới hiện có. Cấu hình mạng dạng này có bộ phận tách tín hiệu (spitter) giữ vai trò thiết bị trung tâm.

Ưu điểm của cấu hình này là mạng có thể gồm nhiều nhóm làm việc ở cách xa nhau. Cấu hình dạng này đưa lại sự uyển chuyển trong việc bố trí đường dây tương thích dễ dàng



Hình 2.12 Cấu trúc mạng Tree

## Server

## *Giao thức MQTT*

## *Định nghĩa:*

MQTT viết tắt của Message Queuing Teltôietry Transport là một giao thức gởi dạng publish/subscribe sử dụng cho các thiết bị Internet of Things với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định.

## *Đặc điểm:*

* Giao thức vận chuyển các message đơn giản.
* Các message trong MQTT không đồng bộ
* Gửi nhận các message theo publish/Subscribe.
* Có 3 cấp độ QoS (Qualities of service)
* Giao thức MQTT chạy trên nền tảng TCP/IP , có thể kết hợp với 6LoWPAN.

## *Mô hình MQTT*

Trong MQTT có các phần tử cốt lõi đó là : client, servers(=broker) , session , subscriptions và các topic.

MQTT Server   
(= broker)

ứng dụng (Cảm biến nhiệt độ)

Topic A

MQTT  
 Client  
 (=publisher,

subscriber)

MQTT Session

Topic B

Client Subcriptions

TCP/IP

TCP/IP

TCP Connection

TCP/IP Network

Hình 2.13 Mô hình MQTT

## *MQTT Client ( = publisher,subscriber):*

MQTT Client là cốt lỗi cho giao thức publish và subscibe.

Clients sẽ subscribe vào topic đã publish và nhận các message từ publish.

Publisher và subscriber đóng vai trò đặc biệt trong MQTT client.

Client gửi một message gọi là publisher

Một hoặc nhiều Client nhận message gọi là Subscriber.

Client

Publisher

Subscriber

Hình 2.14 Mối liên hệ giữa publisher và subscriber.

## *MQTT Server*

Trong giao thức MQTT, MQTT server chạy topic từ client publish lên, nhận các Subscription từ clients trên topic, nhận message từ client và phản hồi thông tin cho client dựa vào client subsriptions trước đó.

Một số MQTT server thường dùng hiện nay cho các dự án IoT đó là : mosquito broker……

## *Topic*

Topic thực chất là message queue, hay còn được gọi là “Chuỗi cấu trúc phân cấp” .

MQTT dùng “topic” để xác định được message được gửi đi từ Client nào.

Topic hỗ trợ tham gia vào quá trình publishe và subcribe từ client.

Topic có các cách đặt tên như sau :

+ Kí hiệu “+” đại diện cho bất kỳ phân cấp nào ngang hàng với nó.

Ví dụ : a/+/b/c , khi client subscribe vào topic này , thì client có thể nhận các bản tin như : a/x/b/c hoặc a/y/b/c.

Các subscribe hợp lệ như: a/b/c/d;+/b/c/d;a/+/+/d

+ Kí hiệu “#” đại diện cho các phân cấp dưới , nghĩa là nó phải đứng cuối cùng trong địa chỉ đăng ký.

Ví dụ: a/b/c/d;a/#; +/b/c/#...

message

Publish

message

Subscribe

Topic

Hình 2.15 Sơ đồ publish/subscribe topic

## *Session*

Trong MQTT, session dùng để xác định một tập tin của client gửi đến server.

Tất cả các kết nối giữa client và server đều diễn ra thông qua các session.

## *Subscription*

Khi subscribe đến một topic , thì client có thể thực thi message với cùng topic subscribe.

Subscription có thể độc lập với cờ session trong việc kết nối các message trong giao thức MQTT.

Trong một subscription bao gồm topic và mức QoS.

## *Message*

Trong MQTT, message là một đơn vị của việc thực thi dữ liệu của client giữa các topic hoặc giữa client với Server.

Trong message có các loại message như : CONNECT,CONNACK,PUBLISH,SUBSCRIBE,SUBACK…….

Ngoài các loại message thì message còn có các cờ như : DUP, QoS,Retain.

DUP: Cờ này được bật khi client hoặc server đang cố chuyển lại một message.

QoS: Cờ này nói đến mức độ tin cậy của message PUBLISH.

+ QoS = 0 : gửi rồi quên ngay <=1

+ Qos = 1: Xác nhận bằng ACK >=1.

+QoS = 2:Nhận đảm bảo = 1.

RETAIN: Cờ này được sử dụng trong message PUBLISH. Client gửi một message PUBLISH đến Server , nếu cờ retain được set thì Server phải hiểu rằng cần giữ lại message khi chuyển nó đến subscriber.

Để xóa được cờ RETAIN topic này ,thì ta cần set cờ RETAIN ở mức FALSE.

Ngoài các trường nói trên , trong MQTT còn có Client ID, username va password, keepAlive, payload …v.v

## *Mosquitto*

Mosquitto là một open source có thể chạy trên các hệ điều hành khác nhau như Window, linux, Mac,.

Mosquitto phổ biến trong MQTT server (MQTT broker), nó hỗ trợ việc kết nối đơn giản, dễ cài đặt và cấu hình.

MQTT Server có rất nhiều chương trình như : Hivtôiq, mqtt-Dashboard, Cloud MQTT, nhưng nhóm đã sử dụng MQTT Mosquitto vì nó dễ sử dụng và có tính ổn định cao nên dễ dàng áp dụng cho các dự án phát triển về IoT.

## *MySQL*

## *Định nghĩa:*

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ nhanh và dễ sử dụng. Nó được lưu trữ dữ liệu của các ứng dụng website khi người dùng thao tác trên website.

## *Đặc điểm:*

Trong hệ quản trị cơ sở dữ liệu bao gồm :

Database : là một cơ sở dữ liệu (CSDL), trong CSDL có các Table.

Table : là bảng dữ liệu, trong table gồm có các trường (filed).

Filed : là trường của dữ liệu.

Nhìn chung ,cấu trúc của MySQL là: Database => Table => Filed.

## *Các câu lệnh truy vấn trong MySQL [6]*

## *Lệnh Insert*

* Lệnh Insert được dùng để chèn một bản ghi vào table.

## *Lệnh Update*

* Lệnh Update được sử dụng để cập nhật lại giá trị cho một hoặc nhiều bản ghi đang có trong bảng .

## *Lệnh Delete*

* Lệnh Delete dùng để xóa một hay nhiều bản ghi.

## *Lệnh Select*

* Dùng để truy vấn các bản ghi trong cơ sở dữ liệu. Đây là câu lệnh được sử dụng nhiều nhất, để giải quyết yêu cầu bài toán.

***Trong đề tài đồ án của nhóm, thì phần Server đang sử dụng Local Host trên máy tính, và lưu dữ liệu bằng MySQL.***

## *Ngôn ngữ lập trình JAVA*

## *Định nghĩa:*

JaVa là ngôn ngữ lập trình bậc cao , có tính chất hướng đối tượng , dựa trên các lớp, thường được sử dụng cho các hệ thống có tính độc lập cao, chạy được trên nhiều nền tảng khác nhau như Window, MAC OS.

## *Đặc điểm :*

Ngôn ngữ lập trình JaVa có các đặc điểm sau :

* + **Tựa trên C++, hướng đối tượng hoàn toàn**.

+ JaVa đã kế thừa lại cú pháp của C và C++.

+ Trong JaVa thao tác với con trỏ bị lược bỏ nhằm bảo đảm tính an toàn và dễ sử dụng .

+ Các thao tác Overload , goto hay struct bị loại bỏ khỏi JaVa.

* + **Độc lập phần cứng với hệ điều hành**.

+ Chương trình ngôn ngữ Java được chạy trên nhiều môi trường khác nhau, còn được gọi là **cross-platform.**

* + **Đa luồng**

**+** Java hỗ trợ lập trình đa tiến trình (Mutilthread) để thực hiện các công việc đồng thời.

* + **Tính an toàn và bảo mật.**

+ Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu.

+ Java cung cấp môi trường quản lý chương trình với nhiều mức khác nhau.

## *Ưu điểm*

* + Có nhiều ngôn ngữ để lập trình Server cho hệ thống như PHP, Nodejs.. Nhưng nhóm đã sử dụng ngôn ngữ lập trình Java cho sự án vì những ưu điểm sau :

+ Dùng mã nguồn mở, rõ ràng.

+ Hoạt động trên Linux,.

+ Có nhiều công cụ để built chương trình như : Eclipse, netbean….

* + Nhóm chúng tôi đã sử dụng bộ thư viện ***Eclipse Paho Java Client*** cho dự án của mình.
  + ***Eclipse Paho Java Client[8]*** cung cấp bộ APIs :

+  MqttAsyncClient cung cấp đầy đủ API , ở đây các hoạt động được hoàn thành thông qua hàm Callback.

## *Eclipse*

## *Eclipse là gì?*

* Eclipse là môi trường phát triển tích hợp cho Java , được phát triển bởi IBM .Ngoài ra , Eclipse còn hỗ trợ cho nhiều ngôn ngữ lập trình khác như : C/C++/PHP /C# ..

## *Ưu, nhược điểm của Eclipse*

* + Uư điểm :

+ Hỗ trợ các công cụ, ngôn ngữ lập trình như Java, C,C#.

+ Có thể chạy trên nhiều hệ điều hành như Linux, Window.

+ Hỗ trợ phát triển các GUI.

+ Có nhiều phiên bản tùy chọn.

* + Nhược điểm :

+ Cài đặt phức tạp.

+ Có thể tốn bộ nhớ.

Với những ưu điểm như vậy, nhóm của chúng tôi đã sử dụng bản ***Eclipse Neon*** chạy trên hệ điều hành Linux để xây dựng Server cho dự án.

## *Android*

## *Định nghĩa :*

* Android là một hệ điều hành mã nguồn mở , hệ điều hành dựa trên Linux cho các thiết bị mobile .

## *Đặc điểm*

* Giao diện (UI) đẹp.
* Có thể lưu trữ dữ liệu trên MySQL.
* Hỗ trợ Media.
* Có tính đa nhiệm
* Widget tùy chỉnh.
* Đa ngôn ngữ.

## *Ứng dụng Android*

* Các ứng dụng Andriod thường được phát triển trong ngôn ngữ Java bởi Android Software Development Kit.
* Android có thể được đóng gói và dễ dàng thông qua các cửa hàng như ***Google Play*** …
* Nhóm chúng tôi đã sử dụng chương trình Android dùng để quản lý các ứng dụng cũng như việc đăng ký hoặc điều khiển hệ thống.
* Nhóm đã sử dụng phần mềm Android Studio để viết ứng dụng cho ứng dụng Android.

## Kết luận

Trong chương này, ta thấy được sự phát triển nhanh chóng củng như tầm ảnh hưởng to lớn của IOT đối với cuộc sống hiện đại. Ngoài ra cho thấy nhu cầu sử dụng các thiết bị được điều khiển từ xa là rất cần thiết cho con người, các loại truyền nhận dữ liệu hoặc điều khiển thiết bị sử dụng sóng vô tuyến( RF) ngày càng phổ biến. Nhờ đó khi điều khiển thiết bị chúng ta dễ dàng và nhanh chóng biết được trạng thái của thiết bị để kịp thời phản ứng với những tình huống xấu xảy ra.

Phần cơ sở lý thuyết, các thiết bị và các công nghệ được sử dụng trong đề tài cho thấy với sự phát triển của các module hiện nay, các bảng mạch trở nên đơn giản, gọn nhẹ hơn rất nhiều, nếu biết ứng dụng tốt thì sẽ tiết kiệm rất nhiều không gian lắp đặt thiết bị.

Và trong chương sau chúng tôi sẽ trình bày về kết quả thực hiện và hướng phát triển đề tài.

# CHƯƠNG 3: THIẾT KẾ, THI CÔNG VÀ KẾT QUẢ

## Giới thiệu chương

Sau 5 tháng tìm hiểu, nghiên cứu thì nhóm tôi đã hoàn thành đề tài: “Thiết kế và thi công hệ thống giám sát và điều khiển không dây các thiết bị trong nhà sử dụng Raspberry Pi 3” . Trong chương này chúng tôi sẽ trình bày về kết quả trong suốt quá trình thực hiện đề tài, các lỗi gặp phải và cách xử lý trong quá trình thi công và hoàn thiện đồ án.

## Thiết kế

## *Mô hình tổng quan của hệ thống*

Chúng tôi xây dựng hệ thống với dựa theo sơ đồ điều khiển như sau:

Khối điều khiển trung tâm:

Raspberry Pi 3

Localhost

MQTT





MQTT

MQTT

SPI





App

Hình 3.1 Mô hình khối điều khiển trung tâm

Khối điều khiển thiết bị đầu cuối:

nRF24L01

STM32F103C8T6

GPIO

GPIO

SPI

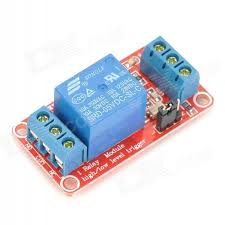
SPI



Relay

Thiết bị đầu cuối







Hình 3.2 Sơ đồ khối điều khiển thiết bị đầu cuối

## Sơ đồ khối hệ thống

## Sơ đồ tổng quát

Wifi

Web Server

RF

Điều khiển trung tâm

Thiết bị

đầu cuối

Wifi

RF

Hình 3.3 Sơ đồ khối hệ thống

## Sơ đồ bộ điều khiển trung tâm

Nguồn

HDMI

SPI

Màn hình điều khiển

Board Raspberry Pi 3

Module RF(receive & transmitter)

SPI

HDMI

Hình 3.4 Sơ đồ khối điều khiển trung tâm

## Sơ đồ bộ điều khiển thiết bị đầu cuối

SPI

STM32F103C8T6

Module RF (receive & transmitter)

SPI

GPIO

GPIO

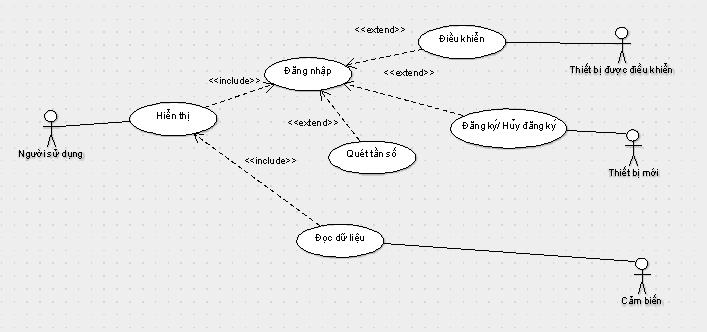
Nguồn

Thiết bị đầu cuối

Hình 3.5 Sơ đồ khối điều khiển thiết bị đầu cuối

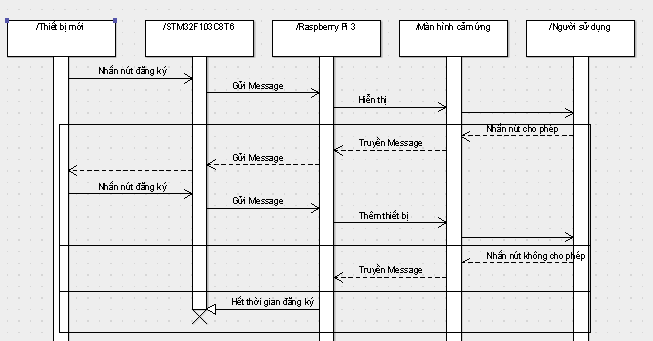
## Một số Diagram

## Usecase Diagram của hệ thống



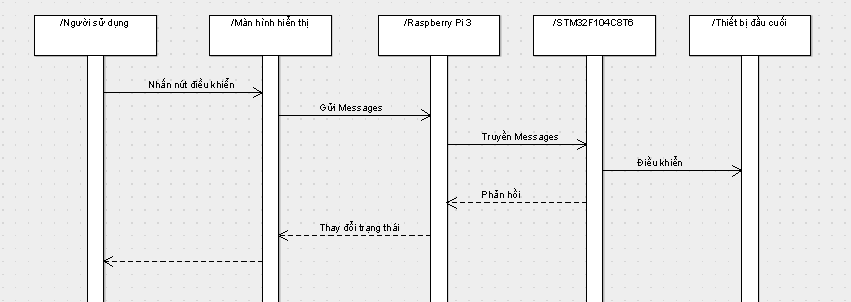
Hình 3.6 Usecase diagram của hệ thống

## Sequence Diagram: Đăng ký thiết bị lên Gateway



Hình 3.7 Sequence diagram đăng ký thiết bị lên Gateway

## Sequence Diagram: Điều khiển thiết bị từ Gateway



Hình 3.8 Sequence diagram khiển thiết bị đầu cuối

Hệ thống gồm nhiều thread được quản lý bởi bộ điều khiển trung tâm sử dụng Qt Creator trên nền Raspberry Pi 3.

## Sequence diagram mô tả các tiến trình hoạt động của hệ thống.

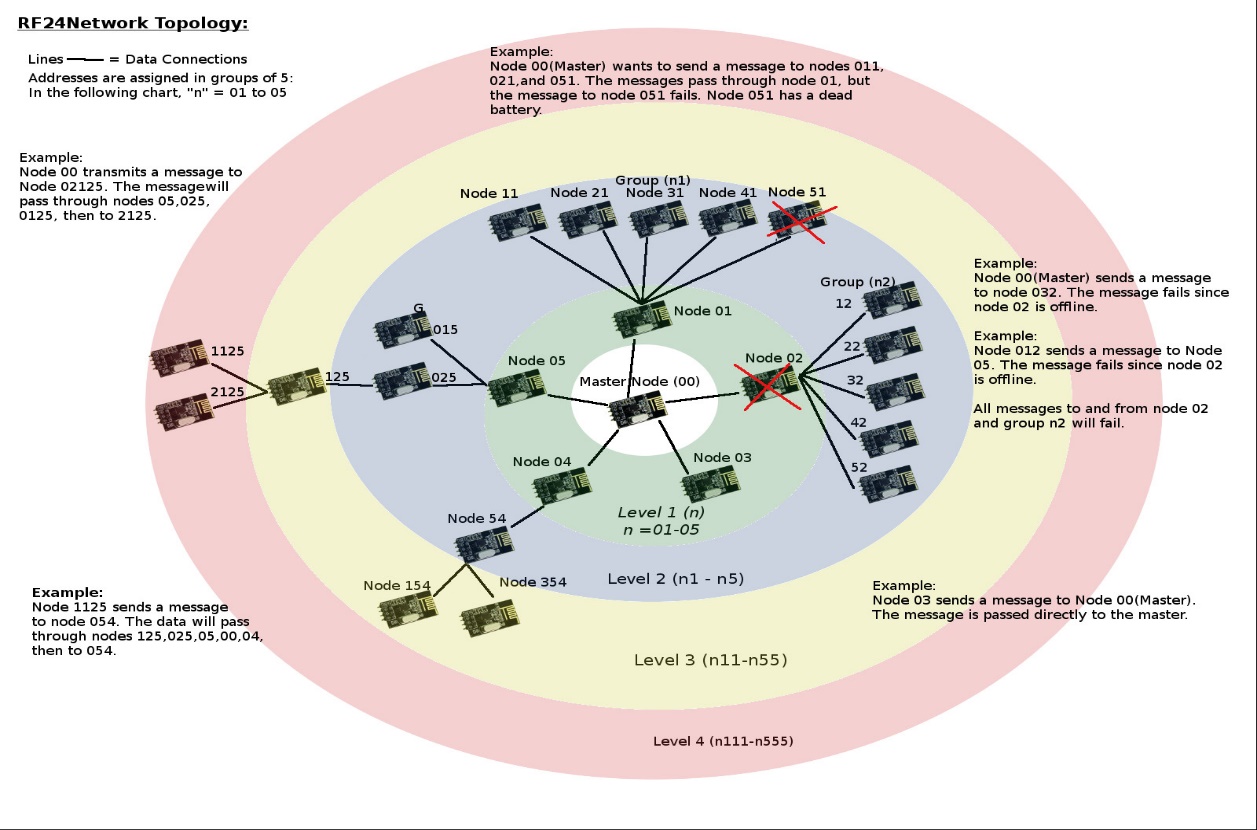
Ban đầu khi khởi động thì hệ thống sẽ tìm kênh hoạt động cho mình bằng việc quét để lựa chọn kênh chưa được sử dụng.

Chương trình điều khiển sử dụng hàm bool testRPD(void) trong thư viện RF24.



Hình 3.9 Sequence diagram khởi tạo hệ thống

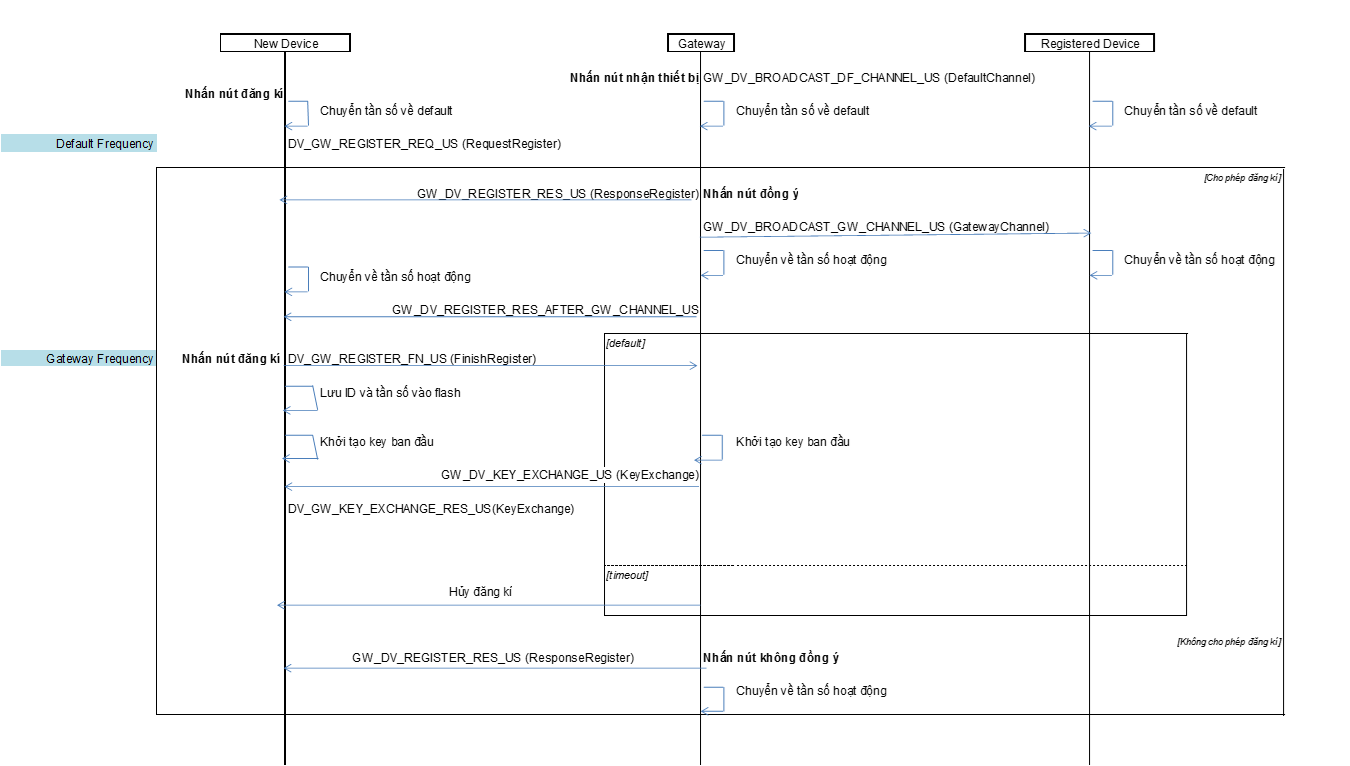
Để tham gia vào mạng thì divice phải có địa chỉ hợp lệ trong mạng đó. Địa chỉ này được cấp bởi các divice đã tham gia vào mạng hoặc gateway như hình vẻ sau:



Hình 3.10 Cấu trúc mạng RF24Network

DV\_ADDRESS\_QUE\_US : Thiết bị mới yêu cầu cấp địa chỉ từ thiết bị đã đăng ký hoặc gateway.

ADDRESS\_RES\_US: Thiết bị đã đăng ký hoặc gateway phản hồi yêu cầu cấp địa chỉ đến thiết bị cần đăng ký.



Hình 3.10 Sequence diagram đăng ký thiết bị

Để đăng ký thiết bị mới thì yêu cầu hệ thống sẽ phải chuyển về tần số mặc định.

Ở bước đầu đăng ký, thì thiết bị gửi về gói dữ liệu chứa thông tin kiểu struct sau:

typedef struct RequestRegister {

header\_msg\_t HeaderMsg;

SubDevice subdevice[MAX\_SUB\_DEVICES];

uint8\_t FirmwareVer;

uint32\_t CodeProduct;

}RequestRegister;

Với kiểu dữ liệu header\_msg:

typedef struct header\_msg\_t{

uint16\_t SourceID;

uint16\_t DestID;

uint16\_t SourceAddress;

uint16\_t DestAddress;

uint64\_t MACGateway;

uint16\_t seq\_num;

}header\_msg\_t;

typedef struct SubDevice {

uint8\_t SubID;

uint8\_t DeviceType;

uint8\_t Direction;

uint8\_t ActiveStatus;

}SubDevice;

Sau khi qua 3 bước xác nhận đăng ký thì thiết bị mới sẽ được cấp ID, địa chỉ trong mạng RF và kênh hoạt động của hệ thống.



Hình 3.11 Sequence diagram điều khiển thiết bị output



Hình 3.12 Sequence diagram điều khiển thiết bị input

Bảng mô tả các messages được sử dụng trong hệ thống:

Bảng 3.1 Bảng mô tả các messages được sử dụng trong hệ thống



## Thi công

## Bộ điều khiển trung tâm (Raspberry Pi 3)

Các chức năng dự kiến sẽ hoàn thành cho bộ điều khiển trung tâm

* Chọn kênh hoạt động cho hệ thống:   
  Hệ thống sử dụng mạng RF thông qua module nRF24L01+, có thể hoạt động ở 126 kênh. Ta đặt vấn đề sẽ có nhiều nhà trong một khu vực sử dụng các hệ thống này, thì mỗi hệ thống phải sử dụng kênh riêng biệt để tránh xung đột.
* Xử lý yêu cầu đăng kí của các thiết bị mới.  
  Hệ thống có khả năng mở rộng tương tác với các thiết bị của nhiều nhà sản xuất khác nhau nếu thiết bị đó chạy chương trình tương thích với hệ thống. Các thiết bị trong mạng được bộ điều khiển trung tâm quản lý qua ID duy nhất của thiết bị.
* Điều khiển các thiết bị loại input.  
  Bộ điều khiển trung tâm điều khiển các thiết bị input như cài đặt thời gian cập nhật trang thái của thiết bị, cài đặt ngưỡng cảnh báo, kiểm tra trạng thái thiết bị.
* Điều khiển các thiết bị loại output

Bộ điều khiển trung tâm điều khiển các thiết bị output như bật/tắt thiết bị, điều chỉnh giá trị trạng thái thiết bị, hẹn giờ điều khiển thiết bị.

* Giao tiếp với server và smartphone  
  Bộ điều khiển trung tâm được quản lý qua tài khoản lưu trữ trên server, smartphone có thể giao tiếp với bộ điều khiển trung tâm để gián tiếp điều khiển thiết bị trong hệ thống.
* Mã hóa dữ liệu trong hệ thống  
  Các dữ liệu giao tiếp giữa bộ xử lý trung tâm và các thiết bị sẽ được mã hóa AES128.
* Sao lưu dữ liệu hệ thống để phòng trường hợp mất điện  
  Toàn bộ thông tin về các thiết bị như ID, khóa bảo mật, kênh hoạt động sẽ được lưu trữ vào tập tin trên bộ nhớ để có thể sao lưu khi mất điện, hệ thống crash.

**Các tác vụ chính trong chương trình điều khiển**

Dựa theo mô hình sequence, bộ điều khiển trung tâm sẽ sử dụng phương pháp lập trình đa luồng trên linux, cụ thể chương trình chính sẽ chạy đa luồng các tác vụ sau:

void\* gw\_task\_init\_gateway\_entry(void \*): tác vụ khởi tạo hệ thống, gồm việc chọn tần số hoạt động cho hệ thống.  
void\* gw\_task\_input\_device\_entry(void \*): tác vụ điều khiển các thiết bị input như PIR, DHT11, cảm biến cửa từ v.v  
void\* gw\_task\_output\_device\_entry(void \*): tác vụ điều khiển các thiết bị output như bóng đèn, quạt...

void\* gw\_task\_query\_address\_entry(void\*): tác vụ cấp phát địa chỉ cho các node con cần đăng kí vào mạng.

void\* gw\_task\_register\_device\_entry(void\*): tác vụ xử lý yêu cầu đăng kí từ các thiết bị mới.

void\* gw\_task\_mqtt\_entry(void\*) : tác vụ xử lý giao tiếp mqtt với smartphone và server.

void\* gw\_task\_if\_rf24\_entry(void\*): tác vụ xử lý thông tin qua mạng rf.

void\* gw\_task\_if\_entry(void\* ): tác vụ xử lý các tin nhắn giao tiếp giữa các thread.

void\* timer\_entry(void\*) : tác vụ xử lý thời gian cho hệ thống.

Khởi tạo các pthread trong chương trình.

Để khởi tạo và sử dụng pthread, ta sử dụng các hàm pthread\_create(), pthread\_join() trong thư viện pthread. Cụ thể trong source code như sau.

for (uint32\_t index = 0; index < ak\_thread\_table\_len; index++) {

/ init mailbox /

q\_msg\_init(task\_list[index].mailbox);

/ create task /

pthread\_create(&(task\_list[index].pthread), NULL, task\_list[index].task, NULL);

APP\_DBG("ID:%08x\tCREATE: %s\n",(unsigned int)task\_list[index].pthread, task\_list[index].info);

}

for (uint32\_t index = 0; index < ak\_thread\_table\_len; index++) {

pthread\_join(task\_list[index].pthread, NULL);

}

Mã hóa dữ liệu theo thuật toán AES128

typedef struct {

uint8\_t task\_id;

uint8\_t sig;

uint8\_t len;

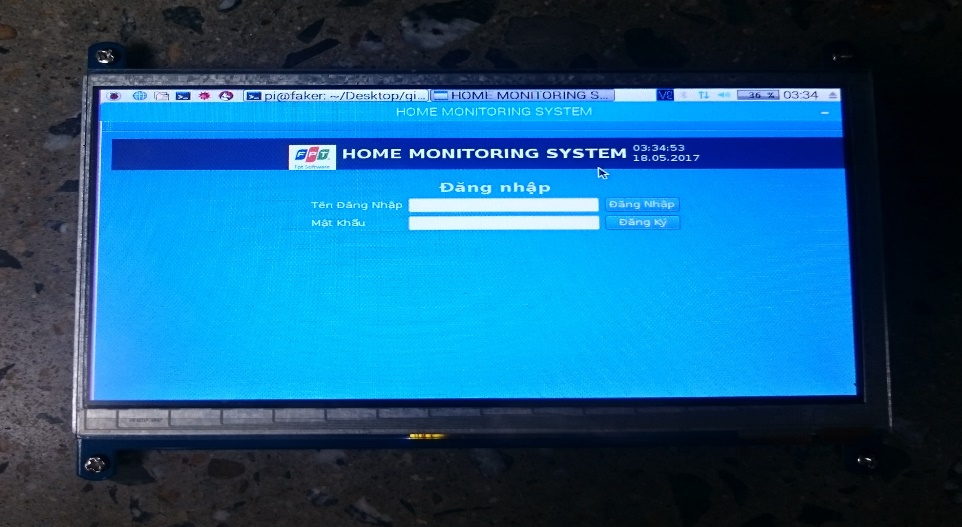
uint8\_t data[AK\_COMMON\_MSG\_DATA\_SIZE];

uint16\_t address;

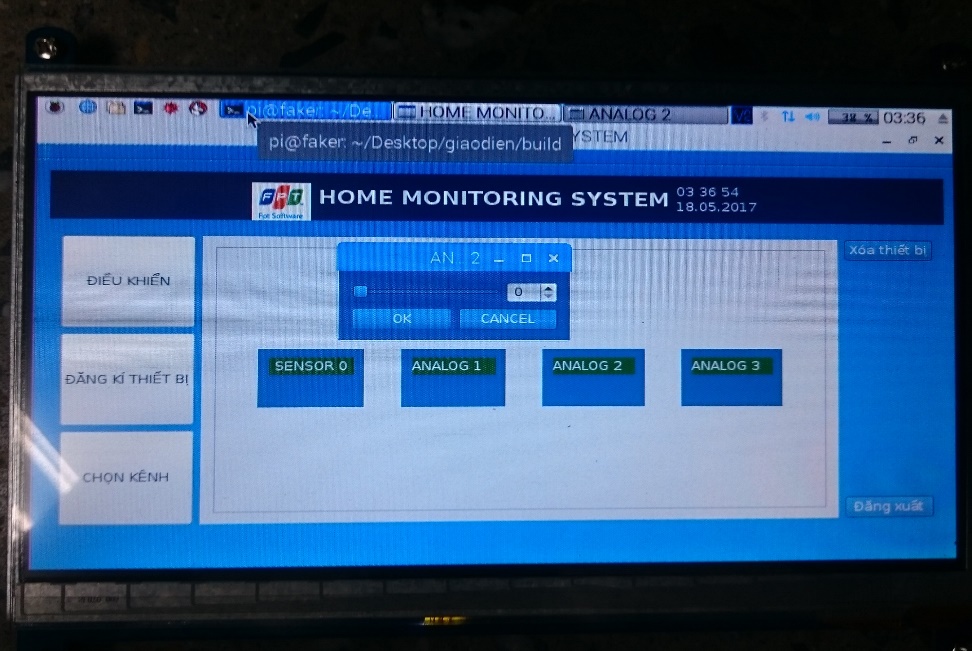
} ak\_msg\_common\_if\_t;

Mảng data [AK\_COMMON\_MSG\_DATA\_SIZE] chứa dữ liệu cần gửi giữa các thiết bị. Phần dữ liệu struct header\_msg\_t không được mã hóa, toàn bộ dữ liệu trong mãng sẽ được mã hóa để đảm bảo bảo mật cho hệ thống.

**Thiết kế giao diện**

Sử dụng framework Qt 5.3, chúng tôi thiết kế giao diện dựa theo chức năng của hệ thống như sau.

Hình 3.13 Giao diện đăng nhập của hệ thống



Liên kết giữa giao diện và chương trình điều khiển

Ban đầu, chúng tôi sử dụng QThread để chạy chương trình điều khiển song song với giao diện. Tuy đã chạy được nhưng chương trình thường bị treo. Sau quá trình tìm hiểu, debug thì chúng tôi xác định được lỗi nằm ở phần cấp phát bộ nhớ, nên phương pháp này không tối ưu.

Vì vậy chúng tôi sử dụng socket để giao tiếp giữa giao diện và chương trình điều khiển.

Hình 3.14 Giao diện đăng ký và điều khiển thiết bị đầu cuối

## Một số lỗi gặp phải

1. Chương trình chạy chiếm nhiều CPU làm raspberry rất nóng  
   Phân tích nguyên nhân: pthread chạy while(1) liên tục, tức là CPU chạy liên tục, kiểm tra có pthread nào không vào chế độ sleep khi rảnh không.  
   Kiểm tra thấy pthread chạy “task\_init\_gateway” sót lệnh usleep (100) ở cuối vòng lặp  
   Sau khi thêm lênh usleep(100) vào , CPU giảm từ >50% xuống <30%. Raspberry không còn nóng
2. Dữ liệu gửi từ device đến gateway không chính xác, cụ thể device gửi gói dữ liệu với ID bằng 1 nhưng gateway nhận dữ liệu ID khác 1  
   Phân tích nguyên nhân: lúc này gói dữ liệu chưa được mã hóa, nên khả năng bị lỗi có thể ở kích thước gói dữ liệu quy định giữa device và gateway khác nhau.  
   Kiểm tra thấy khai báo kiểu dữ liệu biến ID trong kiểu struct của device là uint16\_t, nhưng của Gateway là uint8\_t. Sau khi sửa tại Gateway kiểu dữ liệu của biến ID thành uint16\_t, dữ liệu truyền nhận giữa device và gateway hoàn toàn chính xác.
3. Mỗi node có 5 địa chỉ cố định để cấp cho node con cần đăng kí. Tức là tối đa 5 thiết bị mới có thể là node con của thiết bị đã đăng kí. Nhưng trong quá trình đăng kí, nếu thiết bị mới không hoàn tất đăng kí thì địa chỉ đã cấp trước đó cho thiết bị mới này không thể cấp lần sau cho thiết bị khác.  
   Phân tích nguyên nhân: có thể thiếu sót trong thiết kế mô hình sequence của quá trình đăng kí. Ban đầu node cha cấp địa chỉ cho node con, và gắn cờ báo địa chỉ này đã được sử dụng, nhưng khi node con không hoàn tất đăng kí, thì không hủy cờ báo này nên node cha sẽ không dùng địa chỉ này cho node con khác cần đăng kí. Sau khi sửa lại thì node cha có thể cấp
4. Giao diện sử dụng ở một vài trường hợp bị crash.  
   Phân tích nguyên nhân: chương trình bị crash thường liên quan đến vấn đề cấp phát bộ nhớ động, truy xuất dữ liệu từ con trỏ NULL.  
   Kiểm tra thấy ở phần xóa thiết bị, trong danh sách lưu các con trỏ trỏ đến thông tin thiết bị, theo thuật toán thì sẽ xóa con trỏ cuối cùng trong danh sách và giải phóng vùng nhớ nó trỏ đến, nhưng phần code chỉ xóa vùng nhớ của con trỏ đó trỏ đến chứ không xóa con trỏ khỏi danh sánh, nên ở lần duyệt danh sách tiếp theo, việc truy xuất đến con trỏ này sẽ làm crash chương trình.
5. Giao diện và chương trình điều khiển vài trường hợp không thể giao tiếp qua chuẩn I/O stream.  
   Phân tích nguyên nhân: chương trình điều khiển sử dụng nhiều pthread có điều khiển I/O stream. Vì không sử dụng mutex nên I/O stream có thể sẽ không hoạt động ở một pthread nào đó nếu nó đang được sử dụng bởi một pthread khác.  
   Sau khi thiết kế lại, chỉ dùng một pthead sử dụng I/O stream thì vấn đề được khắc phục.

## Kết quả

Sau quá trình hơn 5 tháng nghiên cứu, tìm hiểu và thi công thì nhóm đã hoàn thành Đề tài: Thiết kế và thi công hệ thống giám sát và điều khiển không dây các thiết bị trong nhà sử dụng Raspberry Pi 3.

Thực hiện được các chức năng :

Xây dựng xong mô hình hệ thống gồm 3 node: Bộ điều khiển trung tâm, bộ điều khiển thiết bị ở lớp mạng thứ nhất, bộ điều khiển thiết bị ở lớp mạng thứ hai.

Quét tần số kiểm tra xtôi tần số nào đang không có thiết bị sử dụng sẽ đưa hệ thống về tần số đó

Đưa hệ thống về kênh mặc định để thiết bị có thể đăng ký mới vào hệ thống.

Điều khiển ,quản lý thiết bị từ xa thông qua Smarthome hoặc Gateway

Cho phép hoặc không cho phép khi một hoặc nhiều thiết bị muốn thêm vào hệ thống

Lưu dữ liệu người dùng vào Database

Cho phép đăng ký thêm nhà mới, lấy thông tin, update thông tin người dùng

Hệ thống có thể lưu lại trạng thái, thông tin của thiết bị khi bị mất điện

Bảo mật dữ liệu trao đổi trong hệ thống bằng mã hóa AES128

Có thể kiểm soát và điều chỉnh nhiệt độ phòng, điều khiển và hiển thị trạng thái hoạt động của các thiết bị trên Gateway hoặc Smartphone,…Chúng ta cũng có thể đăng ký người dùng , thêm nhà , .. thông qua Smartphone để Update cơ sở dữ liệu .Dù chúng ta ở bất cứ đâu, chỉ cần có mạng internet đều có thể điều khiển và kiểm soát được các thiết bị trong nhà.

# KẾT LUẬN

Sau khi hoàn thành đồ án, chúng tôi đã biết cách tìm hiểu và áp dụng được quá trình thiết kế và thi công một dự án theo quy chuẩn của doanh nghiệp. Ngoài ra chúng tôi đã làm quen được với môi trường làm việc thực tế, cải thiện được các kỹ năng cần thiết để phục vụ công việc sau này như kỹ năng nghiên cứu, tìm hiểu công nghệ, kỹ năng làm việc nhóm, kỹ năng thuyết trình, .... Và điều quan trọng nhất là đồ án đã hoàn thành theo đúng kế hoạch ban đầu đặt ra .

Tuy nhiên trong quá trình thực hiện đồ án này còn rất nhiều thiếu sót về mặt kiến thức nên sản phẩm chưa được thực sự hoàn thiện. Trong tương lai sản phẩm có thể phát triển thêm để hệ thống hoàn thiện hơn và biến thành sản phẩm thương mại, được sử dụng rộng rãi.

Hướng phát triển đề tài:

Đây là một đề tài hấp dẫn nên có rất nhiều hướng phát triển có thể khai thác được. Chúng ta có thể:

Phát triển để hệ thống có thể hoạt động ổn định và hiệu quả hơn, phù hợp với điều kiện thực tế.

Phát triển thành hệ thống thông minh có khả năng hoạt động theo kịch bản đã được định sẵn ví dụ như tự động bật đèn khi trời tối, khi xtôi phim thì hệ thống đèn tự động giảm độ sáng,…Và có thể điều khiển, qua giọng nói, cử chỉ.

Phát triển giao thức mạng không dây ổn định hơn về độ trễ tính nhanh chóng và chuẩn xác.

Hệ thống có thể hỗ trợ thêm nhiều ngôn ngữ như tiếng Anh, Nhật, Pháp, … để phát triển hệ thống ra Đông Nam Á và thế giới.

# TÀI LIỆU THAM KHẢO

[1] : <https://iotvietnam.com/>

[2]:<http://www.athlsolutions.com/web/ho-tro/kien-thuc-co-ban/internet-of-things-la-gi-tim-hieu-ve-internet-of-things>

[3]: <http://tmrh20.github.io/RF24/>

[4]: <http://www.raspbian.org/>

[5]: <http://doc.qt.io/>

[6]: [https://raspberrypi.vn](https://raspberrypi.vn/tin-tuc/tren-tay-man-hinh-cam-ung-7-cho-raspberry-pi-1485.pi)

[7]: <https://www.raspberrypi.org/>

**KẾT LUẬN**

**Ghi chú về phần Kết luận**

* Phần Kết luận cần phải nêu được những kết luận chung, khẳng định những kết quả đạt được, những đóng góp, đề xuất và kiến nghị (nếu có);
* Trong phần này, có thể định dạng các điểm/ mục kết luận theo dạng Outline hoặc Numbering hoặc Bullets.

**TÀI LIỆU THAM KHẢO**

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

**Ghi chú:**

Sinh viên xtôi “Quy định về liêm chính học thuật” ban hành kèm theo Quyết định số 29/QĐ-ĐHBK ngày 09/01/2017 và “Hướng dẫn trích dẫn và lập Danh mục tài liệu tham khảo” được ban hành theo văn bản số 30/HD-ĐHBK ngày 09/01/2017 để thực hiện trích dẫn và lập Danh mục tài liệu tham khảo.

**PHỤ LỤC 1**

## *Thông số kỹ thuật STM32F103C8T6*

* Lõi : ARM 32 bit Cortex-M3
* Tần số hoạt động lên tới 72 Mhz
* Bộ nhớ : 64-128 Kb Flash, 20 Kb SRAM
* ADC : 2×12 bit, tần số lấy mẫu 1Mhz
* DMA : Điều khiển 7 kênh DMA
* Timer : 7 bộ, 16 bit
* Giao diện kết nối : 2xI2C, 3xUSART, 2xSPI, CAN, USB 2.0 full-speed.
* Kiểu chân : VFQFPN36, UFQFPN48, BGA100, LQFP48, LQFP64, LQFP100
* 37 chân GPIO, hỗ trợ 5V lẫn 3.3V
* Điện áp cấp 5VDC qua cổng Micro USB được chuyển đồi thành 3.3V qua IC nguồn và cấp cho vi điều khiển chính

## *Thông số kỹ thuật nRF24L01*

Phụ lục 1

* Hoạt động ở dãi tần 2.4G
* Có 126 kênh truyền và nhận dữ liệu.
* Truyền tốc độ cao 1Mbps hoặc 2Mbps.
* Công suất phát: có thể cài đặt được 4 công suất phát: 0, -6,-12,-18 Dbm.
* Công suất thu: có bộ lọc nhiễu tại đầu thu.
* Khuyếch đại bị ảnh hưởng bởi nhiễu thấp LNA.
* Nguồn cấp: 1.9-3.6V.
* Các chân IO chạy được cả 3.3 lẫn 5V.
* Giao tiếp 4 pin SPI.
* Tốc độ tối đa 8Mbps.
* 3-32bytes trên một khung truyền nhận.

## *Thông số kỹ thuật Module cảm biến nhiệt độ, độ ẩm DHT11*

Phụ lục 1

* Đo độ ẩm: 20%-95%.
* Nhiệt độ: 0-50ºC.
* Sai số độ ẩm ±5%.
* Sai số nhiệt độ: ±2ºC.

## *Thông số kỹ thuật Module cảm biến chuyển động PIR*

## *Phụ lục 1*

* Phạm vi phát hiện : góc 360 độ hình nón, độ xa tối đa 6m.
* Nhiệt độ hoạt động : 32­-122 ° F ( 0-50 ° C).
* Điện áp hoạt động : DC 3.8V- ­ 5V.
* Mức tiêu thụ dòng: ≤ 50 uA.
* Thời gian báo: 30 giây có thể tùy chỉnh bằng biến trở.
* Độ nhạy có thể điều chỉnh bằng biến trở.

## *Đặc tính kỹ thuật Module cảm biến rung HDX-2*

Phụ lục 1

* Độ bền, độ chính xác cao.
* Led xanh báo phát hiện rung động, tín hiệu đầu ra DO ở mức thấp.
* Cổng ra DO có thể điều khiển một Rơ le 5V, hoặc được đưa trực tiếp và chân ngắt của MCU
* Dùng LM393  để so sánh điện áp
* Điện áp làm việc: 3.3 - 5VDC. Có đèn led đỏ báo nguồn
* Có thể điều chỉnh độ nhạy của module cảm biến.

## *Thông số kỹ thuật Module cảm biến cửa từ*

Phụ lục 1

* Kim loại lá chắn chống cháy ABS
* Dòng hiện tại: 100mA
* Điện áp định mức: 200 VDC
* Khoảng cách hoạt động: 15- 25 m.
* Công suất: 3W

## *Thông số kỹ thuật Module Relay*

Phụ lục 1

* Sử dụng điện áp nuôi DC 5V.
* Relay tiêu thụ dòng khoảng 80mA.
* Điện thế đóng ngắt tối đa: AC250V ~ 10A hoặc DC30V ~ 10A.
* Có đèn báo đóng ngắt trên mỗi Relay. Có thể chọn mức kích 0 hoặc 1.

## Thông số kĩ thuật Raspberry Pi 3

Phụ lục 1

* Broadcom BCM2837 chipset running at 1.2 GHz
* 64-bit quad-core ARM Cortex-A53
* 802.11 b/g/n Wireless LAN
* Bluetooth 4.1 (Classic & Low Energy)
* Dual core Videocore IV® Multimedia co-processor
* 1 GB LPDDR2 mtôiory
* Supports all the latest ARM GNU/Linux distributions and Windows 10
* MicroUSB connector for 2.5 A power supply
* 1 x 10/100 Ethernet port
* 1 x HDMI video/audio connector
* 1 x RCA video/audio connector
* 4 x USB 2.0 ports
* 40 GPIO pins
* Chip antenna
* DSI display connector
* MicroSD card slot
* Dimensions: 85 x 56 x 17 mm

## Thông số kĩ thuật Màn hình cảm ứng

Phụ lục 1

* Kích cỡ màn hình: 7 “
* Nguồn điện sử dụng: DC 5V – 2 (Tránh trường hợp bị cấp nguồn ngược trở lại mạch Raspberry Pi sẽ dẫn đến hỏng mạch)
* Công suất: 6-7W.
* Tín hiệu đầu vào: HDMI 1.2
* Định dạng video: PAL / NTSC
* Độ phân giải: 1024×600 Pixel
* Sử dụng cảm ứng điện dung.
* Ngôn ngữ hỗ trợ: Trung Quốc, English, Japanese, Korean, Spanish, French, German, Italian, Portuguese.
* Điều khiển: Multi-function OSD, có hỗ trợ điều khiển từ xa.
* Hỗ trợ đảo ngược màn hình (flip up and down)
* Tỷ lệ màn hình: 4:3 hoặc 16:9
* Kích thước:  165.0 (W) × 100.0 (H) × 5 (D) mm
* Góc nhìn: L70 o R70 o U50 o D60
* Loại bảng điều khiển: TFT
* Màu hiển thị: 262K
* Tỷ lệ tương phản 500: 1
* Độ sáng: 220cd / m2
* Thời gian phản hồi: 20ms
* Nhiệt độ hoạt động:  -20 0 ~ 80 0

**PHỤ LỤC 2**

{Font: Time New Roman; thường; cỡ chữ: 12; dãn dòng: 1,3; căn lề: justified}