**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN ĐIỆN TỬ**

---------------o0o---------------

****

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**NHÀ THÔNG MINH**

**VỚI MỘT SỐ TÍNH NĂNG CƠ BẢN**

**GVHD : ThS. PHAN VÕ KIM ANH**

**SVTH: NGUYỄN HOÀNG LONG**

**MSSV: 1412085**

**TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2018**

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC BÁCH KHOA Độc lập – Tự do – Hạnh phúc.

-----✩----- -----✩-----

Số: \_\_\_\_\_\_ /BKĐT

Khoa: **Điện – Điện tử**

Bộ Môn: **Điện Tử**

**NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

1. HỌ VÀ TÊN : NGUYỄN HOÀNG LONG MSSV: 1412085

1. NGÀNH: **ĐIỆN TỬ - VIỄN THÔNG** LỚP : DD14DV2
2. Đề tài: NHÀ THÔNG MINH VỚI MỘT SỐ TÍNH NĂNG CƠ BẢN
3. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):

Tìm hiểu về vi điều khiển họ ARM cortex-M4 của hãng ST (STMicroelectronics) là STM32F4 Discovery. Tìm hiểu các loại cảm biến ( nhiệt độ, khói-khí gas, chuyển động, ánh sang, mưa,...). Tìm hiểu module RFID RC522 sử dụng IC MFRC522 của Phillip và các động cơ RC Servo 9G. Tìm hiểu module wifi ESP8266, cụ thể ở đây là kit RF thu phát wifi ESP8266 NodeMCU Lua V3 CH340 ( phiên bản NodeMCU sử dụng IC nạp giá rẻ CH340 từ Lolin với bộ xử lý trung tâm là module wifi SoC ESP8266 )… Tìm hiểu môi trường lập trình Arduino IDE, cũng như ngôn ngữ lập trình Arduino để lập trình cho NodeMCU… Tìm hiểu về cách truyền nhận dữ liệu lên mạng, cụ thể ở đây là Cloud Firebase Google . Tìm hiểu cách viết App Android bằng Mit App Inventor. Sau khi tìm hiểu kỹ các phần trên, tiến hành kết nối chúng lại với nhau và lập trình để tạo ra các chức năng như: điều khiển thiết bị (đèn,quạt,cửa,…), giám sát qua các cảm biến và cảnh báo,… Từ mô hình với một số tính năng cơ bản trên, sẽ tối ưu và nâng cao hơn để tạo thành một mô hình nhà thông minh ứng dụng được thực tiễn, cần thiết và tiện lợi cho người dung.

1. Ngày giao nhiệm vụ luận văn: 19/09/2018.
2. Ngày hoàn thành nhiệm vụ: 26/12/2018.
3. Họ và tên người hướng dẫn: Phần hướng dẫn

ThS. PHAN VÕ KIM ANH .....................................

................................................................. .....................................

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

*Tp.HCM, ngày…... tháng….. năm 2018*

**CHỦ NHIỆM BỘ MÔN NGƯỜI HƯỚNG DẪN CHÍNH**

**PHẦN DÀNH CHO KHOA, BỘ MÔN:**

Người duyệt (chấm sơ bộ):.......................

Đơn vị:......................................................

Ngày bảo vệ : ...........................................

Điểm tổng kết: .........................................

Nơi lưu trữ luận văn: ...............................

***LỜI CẢM ƠN***

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Sau hơn 4 năm học tập, rèn luyện, tu dưỡng tại trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh, được sự giảng dạy tận tâm của các giảng viên các khoa ngành, sự hỗ trợ đắc lực từ gia đình cũng như sự giúp đỡ nhiệt tình từ bạn bè,… để ngày hôm nay bước vào chặng cuối của đời sinh viên - hoàn thành luận văn tốt nghiệp để ra trường; mang những gì đã tích lũy được ở trường, góp phần xây dựng xã hội và phát triển đất nước.

Với những tích lũy có được ngày hôm nay, ngoài việc cố gắng và nổ lực từ bản thân, thì có rất nhiều người đã giúp đỡ và đưa ra lời khuyên để em hoàn thiện bản thân. Với lòng biết ơn sâu sắc nhất, em xin gửi lời cảm ơn chân thành nhất đến:

Em xin cảm ơn bố và mẹ của em, họ đã hy sinh rất nhiều và tạo mọi điều kiện để em được theo học tại trường Đại học Bách Khoa. Trong những lúc khó khăn của việc học lẫn cuộc sống, họ luôn là điểm tựa và luôn cho em những khích lệ để vượt qua những khó khăn đấy. Một lần nữa, cảm ơn bố và mẹ rất nhiều.

Em xin cảm ơn Ban Giám Hiệu trường đại học Bách Khoa, ban chủ nhiệm khoa Điện-Điện tử, quý Thầy Cô của các khoa ngành đã tạo điều kiện cho em và các sinh viên khác được học tập và rèn luyện trong một môi trường tốt nhất để phát triển toàn diện bản thân. Đặc biệt em xin gửi lời cảm ơn đến cô Phan Võ Kim Anh; cô đã chỉ bảo, hết lòng giúp đỡ và thúc dục em hoàn thành nhiệm vụ trong suốt hành trình dài từ Đồ Án, Thực Tập Tốt Nghiệp đến Luận Văn Tốt Nghiệp; chân thành cảm ơn cô đã tạo mọi điều kiện thuận lợi nhất để em hoàn thành đề tài này.

Và cuối cùng, qua đây em xin gửi lời cảm ơn đến các bạn học và bạn cùng phòng của em là Hoàng Đình Tiến, Đào Duy Luân, Phạm Ngọc Dương, Tân Ngọc Ty. Họ là những người đã giúp đỡ em rất nhiều trong những lúc gặp khó khăn của cuộc sống cũng như trong việc học; gợi ý và chỉ ra những hướng giải quyết phù hợp cho em trong thời gian làm luận văn để em có thể tự mình nghiên cứu và hoàn thành đề tài này tốt nhất trong khả năng có thể của mình.

Sau khoảng thời gian tìm hiểu, thiết kế và thi công luận văn tốt nghiệp với đề tài: “Nhà Thông Minh Với Một Số Tính Năng Cơ Bản”; cuối cùng em cũng đã hoàn thành công việc. Tuy nhiên, do trình độ chuyên môn còn hạn chế nên luận văn của em không thể tránh khỏi thiếu sót. Rất mong nhận được những góp ý, sửa chữa từ Ban hội đồng bảo vệ để luận văn của em được hoàn chỉnh hơn; đó cũng là những ý kiến quý báu để em có thể thay đổi trước khi bước ra cuộc sống thực tiễn. Một lần nữa cảm ơn mọi người rất nhiều.

*Tp. Hồ Chí Minh, ngày 26 tháng 12 năm 2018 .*

**Sinh viên**

**NGUYỄN HOÀNG LONG**

**TÓM TẮT LUẬN VĂN**

Ngày nay cùng với sự phát triển mạnh mẽ của các ngành khoa học kỹ thuật, công nghệ kỹ thuật điện tử có rất nhiều sản phẩm đã ra đời để phục vụ cuộc sống con người. Trong đó, không thể không nhắc đến các sản phẩm thuộc lĩnh vực IoT (Internet of Things – Internet vạn vật); IoT có thể hiểu là tất cả các thiết bị có thể kết nối với nhau trong mạng lưới internet. Mà trong lĩnh vực IoT này, lại càng không thể không nhắc đến Nhà Thông Minh (Smart Home); Nhà Thông Minh là một ngôi nhà mang đầy đủ các tiện ích cho con người, nó được sinh ra để phục vụ nhu cầu thiết thực cho chúng ta.

Như chúng ta đã biết, nhu cầu về cuộc sống ngày càng được nâng cao, nên nhu cầu về sự tiện nghi trong ngôi nhà cũng không nằm ngoài xu thế đó. Ngoài ra, các vụ việc đột nhập nhà ở, cháy nhà không những gây thiệt hại lớn về tài sản của các gia đình mà còn tạo sự lo lắng cho nhiều người, ảnh hưởng đến tính mạng cũng như an toàn trật tự của toàn xã hội. Mà phần lớn các vụ trộm cắp tài sản hay cháy nhà của các gia đình chủ yếu xảy ra khi tất cả mọi người trong gia đình đều đã đi làm hoặc đang ngủ say. Các thiết bị trong nhà ở được điều khiển một cách tự động cũng như giám sát từ xa sẽ đem lại một cuộc sống thoải mái cho mọi thành viên trong gia đình.

Có rất nhiều công ty đã đưa ra sử dụng Ngôi Nhà Thông Minh của họ như BKAV, ACIS, VITY,... Tuy vậy, em vẫn mạnh dạn chọn đề tài này vì công nghệ còn phát triển, Nhà Thông Minh sẽ vẫn cần được nâng cấp lên công nghệ mới.

Với đề tài này, em có thể bật tắt đèn, quạt, cửa…từ xa bằng chiếc điện thoại của mình, có hệ thống theo dõi nhà như chế độ báo trộm, theo dõi nhiệt độ, khói-khí gas…để cấp báo kịp thời khi có sự cố. Tất cả ứng dụng trên đều được điều khiển qua app android và môi trường kết nối là wifi. Việc điều khiển và theo dõi có thể thực hiện được ở bất kì đâu có internet. Ngoài ra, em cũng có thể điều khiển offline tất cả các ứng dụng trên; đồng thời, có thể đóng mở cửa qua thẻ RFID.

**MỤC LỤC**

[1. GIỚI THIỆU 1](#_Toc533546158)

[1.1 Tổng quan 1](#_Toc533546159)

[1.2 Tình hình nghiên cứu trong và ngoài nước 2](#_Toc533546160)

[1.3 Nhiệm vụ luận văn 4](#_Toc533546161)

[2. LÝ THUYẾT 5](#_Toc533546162)

[2.1 Tìm hiểu về vi điều khiển 5](#_Toc533546163)

[2.1.1 Tổng quan về các vi điều khiển có trong luận văn 5](#_Toc533546164)

[2.1.2 Tìm hiểu về NodeMCU 9](#_Toc533546165)

[2.1.3 Tìm hiểu về kit STM32F4 Discovery 12](#_Toc533546166)

[2.2 Tìm hiểu về MIT App Inventor 2 14](#_Toc533546167)

[2.2.1 Tìm hiểu tổng quan về MIT App Inventor 2 14](#_Toc533546168)

[2.2.2 Thiết kế giao diện 18](#_Toc533546169)

[2.2.3 Lập trình chức năng cho app 24](#_Toc533546170)

[2.3 Tìm hiểu về Firebase Google 27](#_Toc533546171)

[2.3.1 Tìm hiểu tổng quan Firebase 27](#_Toc533546172)

[2.3.2 Tìm hiểu về Realtime Database 30](#_Toc533546173)

[2.4 Tìm hiểu về các linh kiện 34](#_Toc533546174)

[2.4.1 Các loại cảm biến 34](#_Toc533546175)

[2.4.2 Động cơ RC servo 9G 39](#_Toc533546176)

[2.4.3 Các thiết bị khác 40](#_Toc533546177)

[3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 48](#_Toc533546178)

[3.1 Yêu cầu thiết kế 48](#_Toc533546179)

[3.2 Phân tích 49](#_Toc533546180)

[3.3 Sơ đồ khối tổng quát 50](#_Toc533546181)

[3.4 Sơ đồ khối chi tiết 51](#_Toc533546182)

[4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM 55](#_Toc533546183)

[4.1 Yêu cầu thiết kế 55](#_Toc533546184)

[4.2 Phân tích 55](#_Toc533546185)

[4.3 Sơ đồ khối tổng quát 56](#_Toc533546186)

[4.4 Sơ đồ khối chi tiết 56](#_Toc533546187)

[5. KẾT QUẢ THỰC HIỆN 60](#_Toc533546188)

[5.1 Kết quả phần mềm 60](#_Toc533546189)

[5.2 Kết quả phần cứng 61](#_Toc533546190)

[6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 61](#_Toc533546191)

[6.1 Kết luận 61](#_Toc533546192)

[6.2 Hướng phát triển 62](#_Toc533546193)

[7. TÀI LIỆU THAM KHẢO 62](#_Toc533546194)

[8. PHỤ LỤC 63](#_Toc533546195)

DANH SÁCH HÌNH MINH HỌA

[Hình 2-1 Kiến trúc của STM32 nhánh Performance và Access 6](#_Toc533599631)

[Hình 2-2 Kit NodeMCU 11](#_Toc533599632)

[Hình 2-3 Sơ đồ chân của NodeMCU 12](#_Toc533599633)

[Hình 2-4 Kit STM32F407VG Discovery 14](#_Toc533599634)

[Hình 2-5 Giao diện quản lý project của MIT App Inventor 2 15](#_Toc533599635)

[Hình 2-6 Giao diện chính của MIT App Inventor 2 16](#_Toc533599636)

[Hình 2-7 Giao diện lập trình theo kiểu kéo và thả của MIT App Inventor 2 16](#_Toc533599637)

[Hình 2-8 Giao diện thiết kế của MIT App Inventor 2 19](#_Toc533599638)

[Hình 2-9 Screen 1 của App 20](#_Toc533599639)

[Hình 2-10 Main Screen 20](#_Toc533599640)

[Hình 2-11 Control-device Screen 21](#_Toc533599641)

[Hình 2-12 Security Screen 21](#_Toc533599642)

[Hình 2-13 Survey Screen 22](#_Toc533599643)

[Hình 2-14 Control-light Screen 22](#_Toc533599644)

[Hình 2-15 Control-fan Screen 23](#_Toc533599645)

[Hình 2-16 Control-door Screen 23](#_Toc533599646)

[Hình 2-17 Add Firebase vào MIT App Inventor 2 24](#_Toc533599647)

[Hình 2-18 Mục Properties khi thêm Firebase 24](#_Toc533599648)

[Hình 2-19 Mã của FirebaseToken 25](#_Toc533599649)

[Hình 2-20 Thuật toán lấy dữ liệu từ Firebase để App đồng nhất với chế độ offline 26](#_Toc533599650)

[Hình 2-21 Thuật toán gửi lệnh điều khiển từ App 26](#_Toc533599651)

[Hình 2-22 Tạo một project trên Firebase 31](#_Toc533599652)

[Hình 2-23 Add một project 32](#_Toc533599653)

[Hình 2-24 Mục Database của project 32](#_Toc533599654)

[Hình 2-25 Tab rules trong mục Database 33](#_Toc533599655)

[Hình 2-26 Mã cho phép truy cập vào Firebase 33](#_Toc533599656)

[Hình 2-27 Theo dõi dữ liệu trên Firebase 34](#_Toc533599657)

[Hình 2-28 Cảm biến nhiệt độ LM35 34](#_Toc533599658)

[Hình 2-29 Cảm biến khói-khí gas MQ-2 36](#_Toc533599659)

[Hình 2-30 Cảm biến chuyển động 37](#_Toc533599660)

[Hình 2-31 Cảm biến mưa 38](#_Toc533599661)

[Hình 2-32 Động cơ RC servo 9G 39](#_Toc533599662)

[Hình 2-33 Mạch còi Buzzer 40](#_Toc533599663)

[Hình 2-34 Cảm ứng 1 chạm điện dung 41](#_Toc533599664)

[Hình 2-35 RFID 42](#_Toc533599665)

[Hình 2-36 LCD1602 43](#_Toc533599666)

[Hình 2-37 LCD2004 44](#_Toc533599667)

[Hình 2-38 Mạch chuyển giao tiếp LCD sang I2C 45](#_Toc533599668)

[Hình 2-39 Module 1 relay (5VDC) 46](#_Toc533599669)

[Hình 2-40 Module 8 relay (5VDC) 47](#_Toc533599670)

[Hình 2-41 Mach giảm áp DC XL4005 (5A) 48](#_Toc533599671)

[Hình 3-1 Schematic để thực hiện mạch in 49](#_Toc533599672)

[Hình 3-2 Mạch in 50](#_Toc533599673)

[Hình 3-3 Sơ đồ khối tổng quát phần cứng 50](#_Toc533599674)

[Hình 3-4 Mach in sau khi hàn và gắn linh kiện 53](#_Toc533599675)

[Hình 3-5 Tổng thể mô hình thực tế mặt cạnh 54](#_Toc533599676)

[Hình 3-6 Tổng thể mô hình thực tế mặt bằng 54](#_Toc533599677)

[Hình 4-1 Sơ đồ khối tổng quát phần mềm 56](#_Toc533599678)

[Hình 4-2 Lưu đồ giải thuật giao tiếp giữa STM32F4 Discovery và NodeMCU 56](#_Toc533599679)

[Hình 4-3 Lưu đồ giải thuật báo động khí gas 57](#_Toc533599680)

[Hình 4-4 Lưu đồ giải thuật chống trộm 57](#_Toc533599681)

[Hình 4-5 Lưu đồ giải thuật đóng mở cửa bằng RFID 58](#_Toc533599682)

[Hình 4-6 Lưu đồ giải thuật điều khiển thiết bị 59](#_Toc533599683)

[Hình 4-7 Lưu đồ giải thuật phần cảm biến 60](#_Toc533599684)

# GIỚI THIỆU

## Tổng quan

Xã hội phát triển kéo theo những cải tiến không ngừng về công nghệ và sự ra đời của nhiều giải pháp hiện đại giúp nâng cao chất lượng sống, trong đó có nhà thông minh – Smarthome. Nhà thông minh là xu hướng phổ biến trên thế giới khi hỗ trợ tiết kiệm thời gian và công sức nhờ áp dụng công nghệ hiện đại. Tại Việt Nam, ứng dụng này cũng ngày càng được lòng người dùng và trở thành lựa chọn ưu tiên.

Vài năm trở lại đây, công nghệ có những bước ấn tượng, trong đó, trí tuệ nhân tạo AI (Artificial Intelligence) được tích hợp ở hầu hết thiết bị điện tử. Từ Smartphone trang bị khả năng nhận diện khuôn mặt, tự động chọn chế độ chụp phù hợp hay tivi, máy giặt, máy hút bụi,… đều được tự động hóa. Nhờ vậy, người dùng có thể thuận tiện hơn trong quá trình sử dụng, tiết kiệm thời gian chỉnh sửa hoặc tìm kiếm thông tin. Sự phát triển của từng thiết bị điện tử dẫn đến một xu hướng mới – Nhà Thông Minh. Smarthome là nơi thiết bị sử dụng điện năng gồm hệ thống đèn, hệ thống an ninh và các thiết bị điện tử gia dụng, đều được kết nối qua mạng internet và có khả năng “giao tiếp” với nhau. Nhờ đó, chủ nhân ngôi nhà có thể dễ dàng điều khiển mọi thiết bị qua bảng điều khiển hoặc di động cầm tay từ bất cứ đâu. Cuộc sống hiện đại, cuốn con người theo vòng quay của xã hội và mang đến nhiều áp lực. Điều này khiến quỹ thời gian dành cho bản thân và chăm sóc gia đình, quan tâm đến ngôi nhà bị thu hẹp. Nếu không có biện pháp kịp thời có thể khiến chúng ta thêm căng thẳng, mệt mỏi. Từ nhu cầu thực tế, smarthome ra đời như giải pháp đáp ứng mong muốn của người dùng. Sống trong ngôi nhà thông minh, chúng ta sẽ được nhắc nhở khi quên tắt điện, được cảnh báo nếu mở đèn quá sáng hay để nhiệt độ quá thấp. Cùng với đó, chúng ta được ”tư vấn” loại ánh sáng nào phù hợp để nghỉ ngơi, thư giãn, ăn uống… Chúng ta cũng không còn lo lắng không biết ngôi nhà sẽ thế nào khi đi vắng, bởi mọi nơi trong nhà đều được kiểm soát.

Để hiểu rõ hơn và có thể thực hành ứng dụng IoT, cũng như AI vào cuộc sống, chúng ta cần phải nghiên cứu nhiều hơn về nó. Chính vì thế, em đã chọn đề tài Nhà Thông Minh – Smarthome để làm luận văn của mình. Một mặt, em có thể tìm hiểu, đi sâu nghiên cứu các vi điều khiển, cũng như các linh kiện và các kit ứng dụng có trong đề tài. Qua đó giúp em cũng cố thêm nền tảng điện-điện tử của mình. Mặt khác, giúp em trau dồi kỹ năng ứng dụng thực tiễn. Để những kiến thức lý thuyết có thể áp dụng vào cuộc sống.

Vậy để bắt đầu nghiên cứu cho đề tài này, ta cần phải chọn cho mình một board mạch thật toàn diện, có vai trò là trung tâm điều khiển cho toàn hệ thống Nhà Thông Minh. Board mạch này phải có tốc độ xử lý nhanh, ổn định, có thể kết nối được internet, giá cả phải chăng, đồng thời có cộng đồng hỗ trợ mạnh để dần dần phát triển lên cho đề tài. Để đáp ứng hoàn hảo cho các yêu cầu trên, em quyết định chọn STM32F4 Discovery của hãng ST (STMicroelectronics) kết hợp với module wifi ESP8266 – kit RF thu phát wifi ESP8266 NodeMCU Lua V3 CH340 ( phiên bản NodeMCU sử dụng IC nạp giá rẻ CH340 từ Lolin với bộ xử lý trung tâm là module wifi SoC ESP8266 ), cả 2 sẽ giao tiếp qua UART.

## Tình hình nghiên cứu trong và ngoài nước

**Tình hình nghiên cứu ngoài nước:**

Trên thế giới đang diễn ra cuộc đua về sản phẩm thông minh. Trong xu hướng Internet kết nối vạn vật ( IoT - Internet of things ), nhiều công ty cố gắng làm nhiều sản phẩm để giữ họ ở sâu trong cuộc chơi. Năm 2016, Samsung tuyên bố rằng 90% sản phẩm của công ty sẽ được kết nối Internet vào năm 2017. Nhờ kết nối Internet, người dùng có thể ra lệnh cho chiếc loa tìm kiếm và phát ra bản nhạc muốn nghe, smarthome có thể kết nối tất cả các vật dụng trong gia đình. Tập hợp các sản phẩm thông minh sẽ tạo ra ngôi nhà thông minh.

Tất cả các hãng công nghệ lớn như Google, Amazon, Apple và Samsung đều đang tìm cách tiến sâu hơn vào thị trường smarthome. Theo thống kê của Statista, năm 2020, giá trị thị trường của smarthome dự báo đạt 43 tỉ USD, gấp 3 lần so với năm 2014.

Nhắc đến những thương hiệu nhà thông minh đến từ nước ngoài, không thể không kể đến: hãng Schneider của Pháp, Smartg4 của Mỹ, Gamma của Gamma JSC, Arteor của hãng Legrand (Pháp), My Home của hãng Bticino (Ý), WattStopper (Mỹ), Mhouse, Home access, Came với đại lý chính thức là NTMC, Hager (Pháp), Crestron (Mỹ),… Các sản phẩm thiết bị điện thông minh đến từ nước ngoài đều mang trong mình những ưu điểm vượt trội về thiết kế, tính năng với những giải pháp tiên tiến và hiện đại.

**Tình hình nghiên cứu trong nước:**

Tại Việt Nam, cùng với xu hướng tăng trưởng mạnh của thị trường, nhà thông minh đang dần trở thành một lĩnh vực hấp dẫn mà không “ đại gia” công nghệ nào muốn bỏ qua, trong đó có các doanh nghiệp công nghệ Việt. Thị trường nhà thông minh ở Việt Nam lại đang cạnh tranh khốc liệt với sự góp mặt của hàng loạt các thương hiệu ngoài nước lẫn trong nước.

Bkav là công ty tiên phong giới thiệu công nghệ smarthome vào năm 2013, kèm theo tuyên bố đã mất 10 năm để theo đuổi và hoàn thiện công nghệ này, gồm cả phần cứng thiết bị và phần mềm điều khiển thông minh. Tất cả các hệ thống thiết bị của ngôi nhà như chiếu sáng, cấp nước, điều hòa, âm thanh, truyền hình,… đều được gắn các bộ điều khiển điện tử để có thể kết nối với Internet và điện thoại di động, cho phép người dùng điều khiển vật dụng từ xa hoặc lập trình cho thiết bị ở nhà hoạt động theo lịch. Từng giới thiệu tại CES 2015 ( Mỹ) và nhiều triển lãm công nghệ trên thế giới, hệ thống Smarthome của Bkav khiến người tiêu dùng trong nước chú ý hơn tới giải pháp thông minh cho ngôi nhà.

Từ dấu ấn của Bkav, thị trường các thiết bị thông minh tại Việt Nam ngày càng phát triển. Dù chưa tiến tới khái niệm smarthome, nhiều doanh nghiệp trong nước cũng muốn mở rộng thị trường thiết bị tự động ( home automation ). Chẳng hạn, cuối năm 2016, một nhóm các sinh viên ở Đại học Bách Khoa TP.HCM quyết tâm phát triển ý tưởng từ luận văn tốt nghiệp: “ Điều khiển hệ thống điện trong nhà bằng điện thoại Android. Kết quả là tháng 4 năm 2016, công ty Cổ phần Vsmarttek ra đời ở khu Công nghệ cao Tp.HCM. Vsmarttek đang chập chững đi lại con đường mà những người khởi nghiệp cùng lĩnh vực ở Đại học Bách Khoa Hà Nội đã từng trải qua với các sản phẩm thiết bị tự động thương hiệu Lumi. Hay mới đây Onsky, một công ty tư nhân có trụ sở tại Thung lũng Silicon (Mỹ), cũng tung ra loạt sản phẩm thông minh trong nhà, phần cứng và phần mềm do các kỹ sư Việt Nam thiết kế.

Ngoài ra, Acis cũng là một thương hiệu nhà thông minh Việt Nam được biết đến như một đơn vị startup công nghệ thành công. Ra đời với sự nghiên cứu, tìm tòi của các cựu sinh viên Đại học Bách Khoa Tp.HCM. Công nghệ nhà thông minh – smarthome vốn là lĩnh vực tưởng như Việt Nam khó bắt kịp về mặt công nghệ đối với Mỹ, Nhật Bản hay Israel. Tuy nhiên, Asic lại là một trong những thương hiệu hiếm hoi chứng tỏ tầm vóc của mình. Đặc biệt, “cuộc cách mạng” của Asic là sự song hành giữa chất lượng và giá cả. Với chỉ khoảng 20-30 triệu đồng cho một dự án nhà riêng và tầm 100 triệu cho biệt thự, Asic rõ ràng có lợi thế giá cả cạnh tranh hơn nhiều so với những công trình tiêu tốn ít nhất 150 triệu tới ngưỡng 500-700 triệu của các công nghệ nước ngoài. Tất nhiên, việc bảo hành 5 năm là cách Asic chứng tỏ sự đảm bảo về sự bền bỉ.

## Nhiệm vụ luận văn

+ Trang bị kiến thức về vi điều khiển họ ARM cortex-M4, cụ thể là STM32F4 Discovery.

+ Tìm hiểu, nghiên cứu các loại cảm biến thông dụng hiện nay: cảm biến nhiệt độ LM35, cảm biến khói-khí gas MQ-2, cảm biến chuyển động, cảm biến mưa, cảm biến ánh sáng, cảm ứng một chạm điện dung,…

+ Tìm hiểu về module RFID RC522 sử dụng IC MFRC522 và động cơ RC servo 9G để đóng mở cửa.

+ Tìm hiểu về các thiết bị như bóng đèn, quạt, còi,…

+ Tìm hiểu board NodeMCU và lập trình Arduino trên phần mềm lập trình Arduino IDE.

+ Tìm hiểu cách đưa dữ liệu lên và lấy dữ liệu xuống từ NodeMCU với Firebase Google.

+ Tìm hiểu và viết ứng dụng Android bằng MIT App Inventor 2.

+ Tìm hiểu về allegro orcad 16.6 để thực hiện vẽ schematic và layout mạch.

+ Kết nối và lập trình để điều khiển thiết bị thông qua app Adroid.

+ Dùng cảm biến chuyển động để chống trộm hoặc bật tắt đèn.

+ Dùng cảm ứng một chạm điện dung để điều khiển thiết bị.

+ Đọc giá trị cảm biến và đưa lên Firebase Google, đồng thời hiển thị giá trị cảm biến đó trên app Android dùng để giám sát nhà của người dùng.

+ Kết hợp tất cả các chức năng có thể có ở trên để tạo thành một mô hình nhà thông minh với một số tính năng cơ bản.

# LÝ THUYẾT

## Tìm hiểu về vi điều khiển

### Tổng quan về các vi điều khiển có trong luận văn

Vi điều khiển là một phần không thể thiếu trong các dự án điện tử. Cho đến thời điểm hiện nay, trên thế giới cũng như ở Việt Nam, đã có rất nhiều vi điều khiển. Chúng có các ưu điểm và nhược điểm, tùy thuộc vào loại dự án mà chúng ta sử dụng.

Có thể nói việc sử dụng các loại vi điều khiển và vi xử lý trong các thiết bị điện tử tự động ở Việt Nam rất đa dạng, phong phú tùy vào yêu cầu kỹ thuật và giá thành sản phẩm.

Chúng ta có thể liệt kê các vi điều khiển và vi xử lý có trên thị trường hiện nay:

* Freescale 68HC11 (8-bit)
* Intel 8051
* STMicroelectronics STM8S (8-bit), ST10 (16-bit) và STM32 (32-bit)
* Atmel AVR (8-bit), AVR32 (32-bit) và AT91SAM (32-bit)
* Freescale ColdFire (32-bit) và S08 (8-bit)
* Hitachi H8 (8-bit), Hitachi SuperH (32-bit)
* MIPS (32-bit PIC32)
* PIC (8-bit PIC16, PIC18, 16-bit dsPIC33/PIC24)
* PowerPC ISE
* PSoC (Programmable System-on-Chip)
* Texas Instruments Microcontrollers MSP430 (16-bit), C2000 (32-bit) và Tiva C (32-bit)
* Toshiba TLCS-870 (8-bit/16-bit)
* Zilog Ez8 (16-bit), Ez80 (8-bit)
* Philips Semiconductors LPC2000, LPC900, LPC700

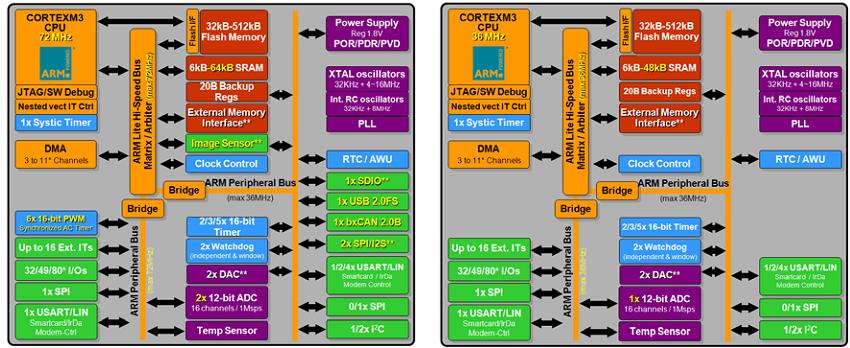
Qua tham khảo tài liệu, internet, cũng như bạn bè có kinh nghiệm trong việc phát triển các ứng dụng IoT, em quyết định chọn vi điều khiển STM32 ( kit STM32F4 Discovery ) và ESP8266 ( kit NodeMCU Lua V3 CH340 ). Đây là các loại kit thông dụng trong lĩnh vực IoT hiện nay. Nó mạnh mẽ, cộng đồng lớn mạnh, dễ dàng nghiên cứu và phát triển.

* **STM32**

STM32 là vi điều khiển do hãng STMicroelectronics sản xuất dựa trên nền tảng lõi vi xử lý ARM Cortex-M. Là một dòng sản phẩm vi điều khiển 32 bit kết hợp các ưu điểm về hiệu suất cao, khả năng xử lý thời gian thực, xử lý tín hiệu số, tiêu thụ ít năng lượng, hoạt động điện áp thấp, trong khi duy trì khả năng tích hợp đầy đủ và dễ dàng phát triển ứng dụng.

Vi điều khiển STM32 dựa trên một lõi tiêu chuẩn công nghiệp, có thể sử dụng nhiều công cụ và phần mềm để phát triển ứng dụng. Điều này làm cho dòng STM32 là sự lựa chọn lý tưởng đối với các dự án nhỏ hoặc cho thiết kế nền tảng.

ST đã đưa ra thị trường 4 dòng vi điều khiển dựa trên ARM7 và ARM9, nhưng STM32 là một bước tiến quan trọng trên đường cong chi phí và hiệu suất, giá chỉ gần 1 Euro với số lượng lớn, STM32 là sự thách thức thật sự với các vi điều khiển 8 và 16 bit truyền thống. STM32 đầu tiên gồm 14 biến thể khác nhau, được phân thành hai dòng: dòng Performance có tần số hoạt động của CPU lên tới 72 Mhz và dòng Access có tần số hoạt động lên tới 36 Mhz. Các biến thể STM32 trong hai nhóm này tương thích hoàn toàn về cách bố trí chân (pin) và phần mềm, đồng thời kích thước bộ nhớ FLASH ROM có thể lên tới 512K và 64K SRAM. (Hình 2-1)



Hình 2-1 Kiến trúc của STM32 nhánh Performance và Access

* **Sự tinh vi**

Thoạt nhìn thì các ngoại vi của STM32 cũng giống như những vi điều khiển khác, như hai bộ chuyển đổi ADC, timer, I2C, SPI, CAN, USB và RTC. Tuy nhiên mỗi ngoại vi trên đều có rất nhiều đặc điểm thú vị. Ví dụ như bộ ADC 12 bit có tích hợp một cảm biến nhiệt độ để tự động hiệu chỉnh nhiệt độ thay đổi và hỗ trợ nhiều chế độ chuyển đổi. Mỗi bộ định thời có 4 khối capture compare (dùng để bắt sự kiện với tính năng input capture và tạo dạng sóng ở ngõ ra với output compare), mỗi khối định thời có thể liên kết với các khối định thời khác để tạo ra một mảng các định thời chính xác hơn. Một bộ định thời chuyên hỗ trợ điều khiển động cơ với 6 đầu ra tín hiệu điều biến độ rộng xung PWM với dead time (khoảng thời gian được chèn vào giữa hai đầu tín hiệu xuất PWM bù nhau trong điều khiển mạch cầu H) lập trình được và một đường break input (khi phát hiện điều kiện dừng khẩn cấp) sẽ buộc tín hiệu PWM sang một trạng thái an toàn đã được cài sẵn. Ngoại vi nối tiếp SPI có một khối kiểm tổng (CRC) bằng phần cứng cho 8 và 16 word hỗ trợ tích cực cho giao tiếp thẻ nhớ SD hoặc MMC.

STM32 có hỗ trợ thêm tối đa 12 kênh DMA (Direct Memory Access). Mỗi kênh có thể được dùng để truyền dữ liệu đến các thanh ghi ngoại vi hoặc từ các thanh ghi ngoại vi với kích thước từ (word) dữ liệu truyền đi có thể là 8 hoặc 16 hoặc 32 bit. Mỗi ngoại vi có thể có một bộ điều khiển DMA đi kèm dùng để gửi hoặc truy vấn dữ liệu như yêu cầu. Một bộ phân xử bus nội và ma trận bus tối thiểu hóa sự tranh chấp bus giữa truy cập dữ liệu thông qua CPU và các kênh DMA. Điều đó cho phép các đơn vị DMA hoạt động linh hoạt, dễ dùng và tự động điều khiển các luồng dữ liệu bên trong vi điều khiển.

* **Khả năng an toàn**

Ngày nay các ứng dụng hiện đại thường phải hoạt động trong môi trường khắt khe, đòi hỏi tính an toàn cao, cũng như đòi hỏi sức mạnh xử lý và càng nhiều thiết bị ngoại vi. Để đáp ứng các yêu cầu khắt khe đó, STM32 cung cấp một số tính năng phần cứng hỗ trợ các ứng dụng một cách tốt nhất. Chúng bao gồm một bộ phát hiện điện áp thấp, một hệ thống bảo vệ xung Clock và hai bộ Watchdogs. Bộ đầu tiên là một Watchdog cửa sổ. Watchdog này phải được làm tươi trong một khung thời gian xác định. Nếu nhấn nó quá sớm, hoặc quá muộn, thì Watchdog sẽ kích hoạt. Bộ thứ hai là một Watchdog độc lập, có bộ dao động bên ngoài tách biệt với xung nhịp hệ thống. Hệ thống bảo vệ xung nhịp có thể phát hiện lỗi của bộ dao động chính bên ngoài (thường là thạch anh) và tự động chuyển sang dùng bộ dao động nội RC 8 Mhz.

* **Tính bảo mật**

Một trong những yêu cầu khắt khe khác của thiết kế hiện đại là nhu cầu bảo mật mã chương trình để ngăn chặn sao chép trái phép phần mềm. Bộ nhớ Flash của STM32 có thể được khóa để chống truy cập đọc Flash thông qua cổng gỡ lỗi (Debug). Khi tính năng bảo vệ đọc được kích hoạt, bộ nhớ Flash cũng được bảo vệ chống ghi để ngăn chặn mã không tin cậy được chèn vào bảng vector ngắt. STM32 cũng có một đồng hồ thời gian thực và một khu vực nhỏ dữ liệu trên SRAM được nuôi nhờ nguồn pin. Khu vực này có một đầu vào chống giả mạo, có thể kích hoạt một sự kiện ngắt khi có sự thay đổi trạng thái ở đầu vào này. Ngoài ra, một sự kiện chống giả mạo sẽ tự động xóa dữ liệu được lưu trữ trên SRAM được nuôi bằng nguồn pin.

* **Phát triển phần mềm**

Các công cụ phát triển cho ARM hiện có đã được hỗ trợ tập lệnh Thumb-2 và dòng Cortex. Ngoài ra ST cũng cung cấp một thư viện điều khiển thiết bị ngoại vi, một bộ thư viện phát triển USB như là một thư viện ANSI C và mã nguồn tương thích với các thư viện trước đó được công bố cho vi điều khiển STR7 và STR9. Có rất nhiều hệ điều hành thời gian thực RTOS (Real Time Operating System) mã nguồn mở, thương mại và middleware (TCP/IP, hệ thống tập tin,…) hỗ trợ cho họ Cortex. Dòng Cortex-M3 cũng đi kèm với một hệ thống gỡ lỗi hoàn toàn mới gọi là CoreSight. Truy cập vào hệ thống CoreSight thông qua cổng truy cập gỡ lỗi (Debug Access Port), cổng này hỗ trợ kết nối chuẩn JTAG hoặc giao diện 2 dây (serial wire – 2 Pin), cũng như cung cấp trình điều khiển chạy gỡ lỗi, hệ thống CoreSight trên STM32 cung cấp hệ thống điểm truy cập và một công cụ theo dõi. Công cụ này có thể gửi thông tin về ứng dụng được lựa chọn đến công cụ gỡ lỗi. Điều này có thể cung cấp thêm các thông tin gỡ lỗi và cũng có thể được sử dụng trong quá trình thử nghiệm phần mềm.

STM32 có sẵn một bộ thư viện ngoại vi chuẩn và mẫu, ví dụ hỗ trợ lập trình mà không cần kiến thức chuyên sâu hay hiểu rõ datasheet của chip, giúp nhanh chóng tập trung vào việc viết chương trình, tiết kiệm thời gian phát triển sản phẩm.

* **ESP8266**

ESP8266 là dạng vi điều khiển tích hợp Wifi (Wifi SoC) được phát triển bởi Espressif Systems, một nhà sản xuất Trung Quốc có trụ sở tại Thượng Hải. Với vi điều khiển và wifi tích hợp, ESP8266 cho phép lập trình viên có thể thực hiện vô số các tác vụ TCP/IP đơn giản để thực hiện vô số các ứng dụng khác nhau, đặc biệt là các ứng dụng IoT. Tuy nhiên, vào thời điểm ra mắt năm 2014, hầu như chỉ có tài liệu bằng tiếng Trung Quốc nên ESP8266 chưa được phổ biến như hiện nay.

Module ESP8266 có giá thành rẻ, phải nói là rẻ nhất trong tất cả các loại wifi SoC từ trước tới nay (trước ESP8266 có series CC3xxx từ TI rất mắc nên không phổ biến), chỉ khoảng 2 USD cho phiên bản đầu tiên, điều này đã thu hút các IoT-er khám phá cũng như dịch các tài liệu của ESP8266 sang tiếng Anh và phát triển vô số các ứng dụng kèm theo. ESP8266 là dòng chip Low-power và là một wifi SoC nên cần rất ít linh kiện ngoài. Hoạt động trong dãi nhiệt -40­­­­­℃ tới +125℃.

Sau nhiều năm phát triển, hiện nay đã có hơn 14 phiên bản ESP ra đời, trong đó phổ biến nhất là ESP-12.

Trong các dòng ESP8266 thì ESP8266EX là một trong những dòng chip tích hợp wifi trong lĩnh vực công nghiệp. Với kích thước chỉ 5x5 (mm), ESP8266EX cần rất ít linh kiện hỗ trợ ngoài. ESP8266EX được tích hợp 32 bit Tensilica MCU, các ngoại vi cơ bản, antenna switches, RF balun, khuếch đại công suất, khuếch đại nhận nhiễu thấp, bộ lọc và các modules quản lý nguồn. ESP8266EX tích hợp vi xử lý Tensilica L106 32 bit (MCU) là dòng chip low-power, 16 bit RSIC, tốc độ clock cao nhất là 160 Mhz. Nếu hệ thống hoạt động với Real Time Operation System (RTOS) và wifi stack thì ta có khoảng 80% khả năng xử lý cho ứng dụng người dùng.

Với những ứng dụng cho mobile, thiết bị điện tử cầm tay và IoT, ESP8266EX hoạt động với mức tiêu thụ năng lượng rất thấp nhưng công nghệ độc quyền. Tính năng tiết kiệm năng lượng với 3 chế độ hoạt động – active mode, sleep mode và deep sleep mode, vì vậy cho phép hiện thực những thiết bị với thời lượng pin rất lớn.

* + 1. Tìm hiểu về NodeMCU
* **Khả năng hoạt động như một modem wifi**

+ Có thể quét và kết nối đến một mạng wifi bất kỳ (Wifi Client) để thực hiện các tác vụ như lưu trữ, truy cập dữ liệu từ server.

+ Tạo điểm truy cập wifi (Wifi Access Point) cho phép các thiết bị khác kết nối, giao tiếp và điều khiển.

+ Là một server để sử lý dữ liệu từ các thiết bị sử dụng internet khác,…

* **Nguồn vào và nguồn ra**

+ ESP8266 NodeMCU nhận nguồn từ cổng micro USB tích hợp sẵn trên mạch, giúp việc nạp code trở nên dễ dàng hơn. Bên cạnh đó, việc cấp nguồn cho module cũng linh động hơn vì chúng ta có thể sử dụng sạc dự phòng hoặc nguồn tổ ong thay cho nguồn từ USB trên máy tính (nguồn cấp tối đa là 5V).

+ Khi sử dụng các chân cấp nguồn, hãy luôn kiểm tra để chắc chắn không cắm nhầm chân dương (trên board là 3V, Vin, VV) và chân âm (trên board là G). Tuy nhiên, 3 chân 3V đều được bảo vệ, khi cắm ngược cực, module sẽ chỉ nóng lên và dừng hoạt động. Chân Vin hay VV nếu cắm ngược cực sẽ cháy.

* **Truyền và nhận tín hiệu**

+ ESP8266 NodeMCU có tổng cộng 13 chân GPIO (General Purpose Input/Output) – chân có thể truyền/nhận tín hiệu (trên board là D0 đến D8, RX & TX, S2 & S3).

* **Thông số kỹ thuật của kit RF thu phát wifi ESP8266 NodeMCU Lua V3 CH340**

Kit RF thu phát wifi ESP8266 NodeMCU Lua V3 CH340 là phiên bản NodeMCU sử dụng IC nạp giá rẻ CH340 từ Lolin với bộ xử lý trung tâm là module Wifi SoC ESP8266, kit có thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản. Kit này được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua song wifi, đặc biệt là các ứng dụng liên quan đến IoT:

+ IC chính: ESP8266 Wifi SoC.

+ Phiên bản firmware: NodeMCU Lua.

+ Chip nạp và giao tiếp UART: CH340.

+ GPIO tương thích hoàn toàn với firmware NodeMCU.

+ Cấp nguồn: 5VDC MicroUSB hoặc Vin (có thể là VV nữa).

+ GPIO giao tiếp mức 3.3VDC.

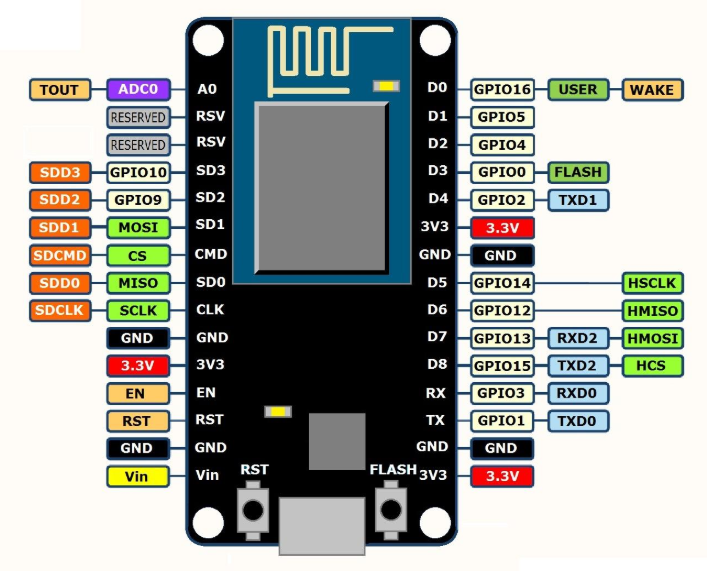
+ Tích hợp Led báo trạng thái, nút Reset, Flash.

+ Tương thích hoàn toàn với trình biên dịch Arduino.

+ Kích thước: 59x32mm.



Hình 2-2 Kit NodeMCU



Hình 2-3 Sơ đồ chân của NodeMCU

* + 1. Tìm hiểu về kit STM32F4 Discovery

Bộ kít STM32F4 Discovery với vi điều khiển hiệu suất cao STM32F407VGT6, cho phép người dùng dễ dàng phát triển các ứng dụng xử lý tín hiệu số. Nó bao gồm một công cụ ST-LINK tích hợp sẵn trên board mạch giúp nạp chương trình, gỡ lỗi nhanh chóng. Có các tính năng chính sau:

+ Vi điều khiển 32-bit ARM Cortex-M4 STM32F407VGT6 với lõi FPU hỗ trợ xử lý, tính toán dấu chấm động, 1 MB bộ nhớ Flash, 192 KB RAM.

+ On-board ST-LINK/V2 trên STM32F4 Discovery giúp nạp chương trình, gỡ lỗi.

+ Nguồn điện cung cấp cho board mạch: thông qua cổng USB hoặc từ một nguồn cung cấp điện áp 5V bên ngoài.

+ Từ board mạch, có thể cấp nguồn 3V và 5V cho các ứng dụng.

+ Cảm biến chuyển động LIS302DL, ST MEMS 3 trục gia tốc.

+ Cảm biến âm thanh MP45DT02 ST-MEMS, mic cảm biến âm thanh vô hướng kỹ thuật số.

+ Bộ chuyển đổi DAC âm thanh CS43L22.

+ Tám đèn Led:

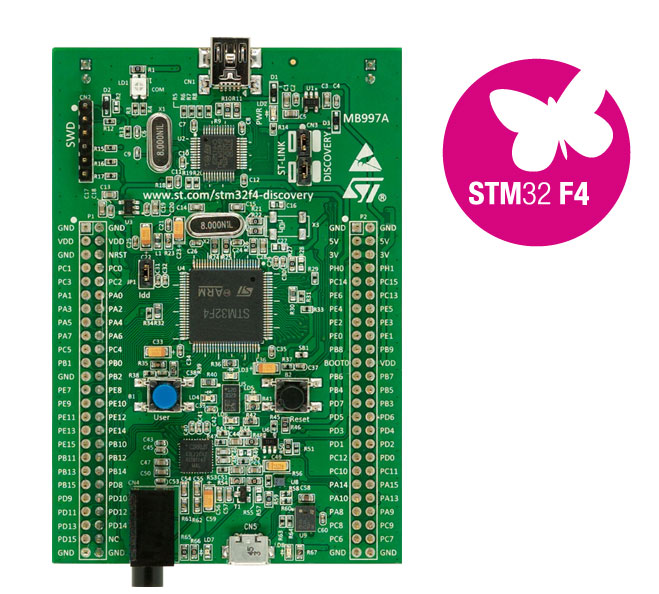
* LD1 (đỏ/xanh lá) để giao tiếp USB.
* LD2 (màu đỏ) báo hiệu nguồn 3V3 on.
* Bốn đèn Led màu: LD3 (màu cam), LD4 (màu xanh lá cây), LD5 (màu đỏ) và LD6 (màu xanh dương).
* Hai USB OTG LED LD7 (màu xanh lá cây) VBUS và LD8 (màu đỏ).

+ Hai nút bấm (nút bấm User màu xanh, nút bấm Reset màu đen).

+ OTG FS USB với cổng nối micro-AB.

+ Header mở rộng cho tất cả LQFP100 I/O.

+ Phần mềm miễn phí bao gồm một loạt các ví dụ, sử dụng thư viện chuẩn cũng như thư viện HAL của ST.



Hình 2-4 Kit STM32F407VG Discovery

## Tìm hiểu về MIT App Inventor 2

### Tìm hiểu tổng quan về MIT App Inventor 2

MIT App Inventor dành cho Android là một ứng dụng web nguồn mở ban đầu được cung cấp bởi Google và hiện tại được duy trì bởi Viện Công Nghệ Massachusetts (MIT).

Nền tảng cho phép nhà lập trình tạo ra các ứng dụng phần mềm cho hệ điều hành Android (OS). Bằng cách sử dụng giao diện đồ họa, nền tảng cho phép người dùng kéo và thả các khối mã (blocks) để tạo ra các ứng dụng có thể chạy trên thiết bị Android. Đến thời điểm 07/2017, phiên bản IOS của nền tảng này đã bắt đầu được đưa vào thử nghiệm bởi Thunkable, là một trong các nhà cung cấp ứng dụng web cho ngôn ngữ này.

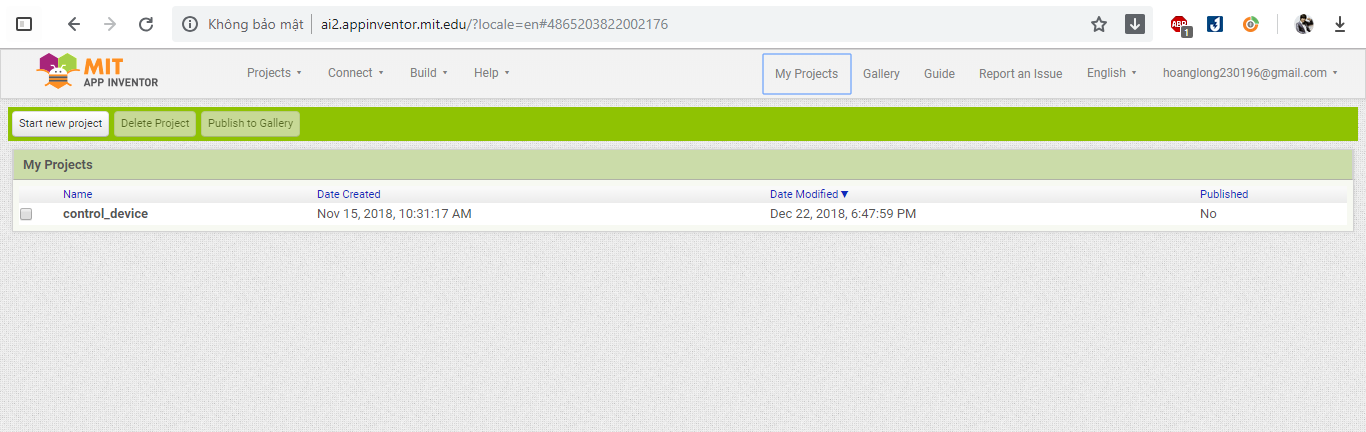
Vào ngày 6 tháng 12 năm 2013, MIT đã phát hành App Inventor 2, đổi tên từ tên gốc “App Inventor Classic”. Tính đến tháng 5 năm 2014, nền tảng đã có 87 nghìn người dùng hoạt động hàng tuần và 1.9 triệu đăng kí từ 195 quốc gia với tổng số 4.7 triệu ứng dụng được xây dựng. Vào tháng 12 năm 2015, có 140 nghìn người sử dụng hoạt động hàng tuần và 4 triệu đăng kí tại 195 quốc gia với tổng số 12 triệu ứng dụng được xây dựng. Hiện nay, mỗi tháng MIT App Inventor có hơn 400.000 người dùng đến từ 195 quốc gia và đã tạo ra gần 22 triệu ứng dụng, MIT App Inventor đang thay đổi cách thế giới tạo ra các ứng dụng và cách mà học sinh nhỏ tuổi bắt đầu về máy tính.

Mục tiêu cốt lõi của MIT App Inventor là giúp đỡ những người chưa có kiến thức về ngôn ngữ lập trình từ trước có thể tạo ra những ứng dụng có ích trên hệ điều hành Android. Phiên bản mới nhất là MIT App Inventor 2.

Ngày nay, MIT đã hoàn thiện App Inventor và nó được chia sẻ ngay trên tài khoản Google. Các lập trình viên mới bắt đầu hoặc bất kỳ ai muốn tạo ra ứng dụng Android, chỉ cần vào địa chỉ web của MIT, nhập thông tin tài khoản Google, và từ những mảnh ghép nhỏ, xây dựng những ý tưởng của mình.

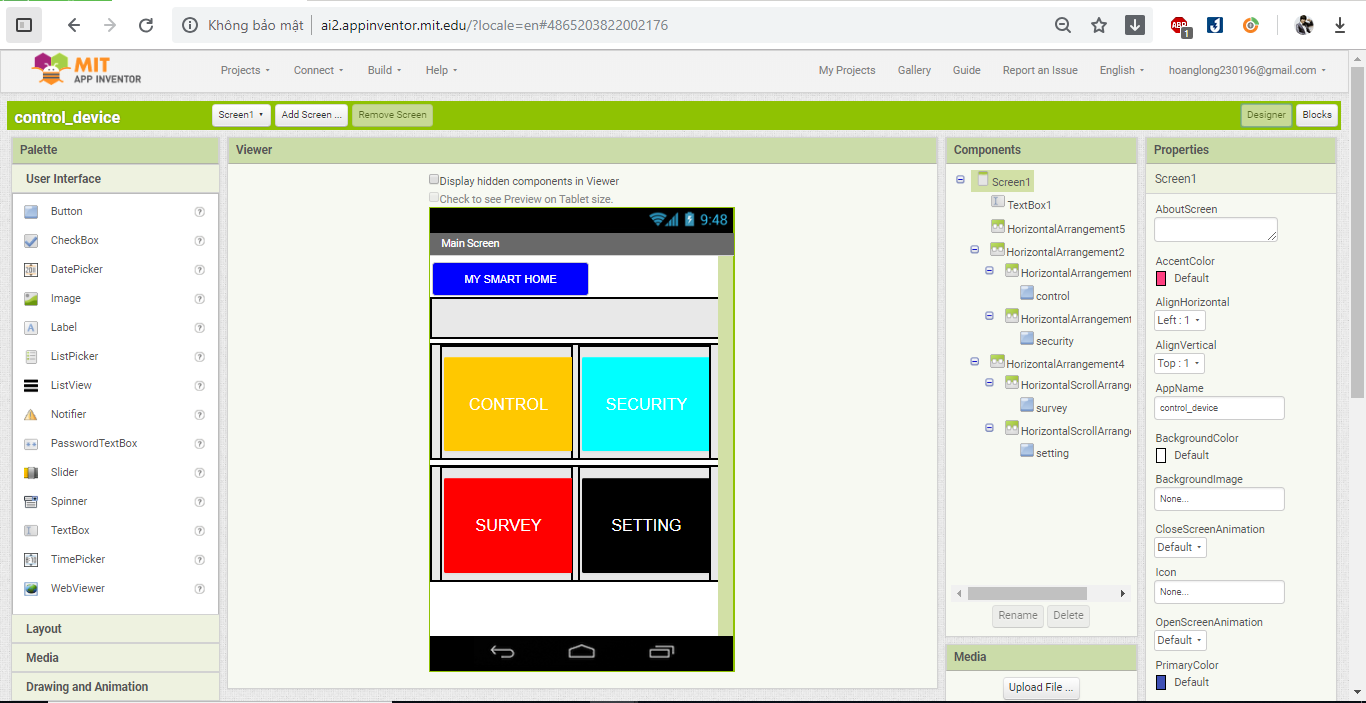
Để sử dụng được App Inventor, chúng ta truy cập vào địa chỉ <http://ai2.appinventor.mit.edu> . Sau đó tiến hành đăng nhập bằng tài khoản Google của chúng ta để mở trang quản lí các project.

+ Giao diện quản lý project



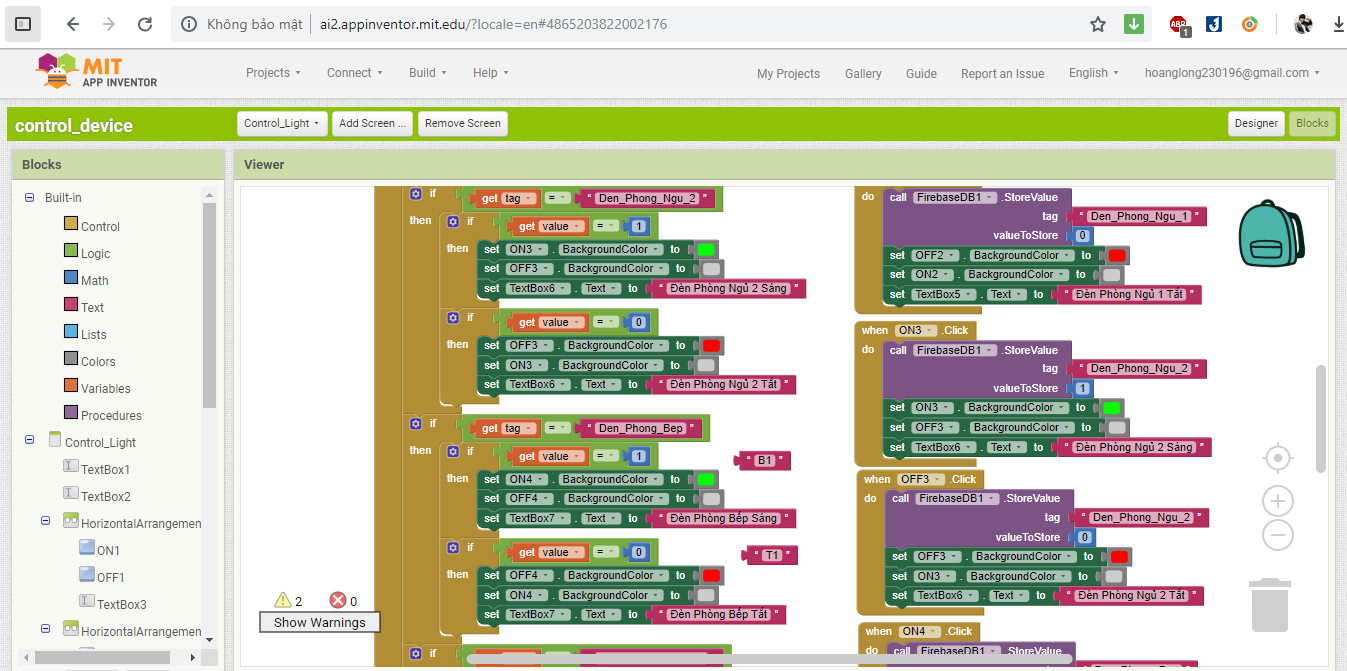
Hình 2-5 Giao diện quản lý project của MIT App Inventor 2

+ Giao diện thiết kế (Design)



Hình 2-6 Giao diện chính của MIT App Inventor 2

+ Giao diện lập trình (Blocks)



Hình 2-7 Giao diện lập trình theo kiểu kéo và thả của MIT App Inventor 2

Với mục tiêu trên, MIT App Inventor đã xây dựng một kho dữ liệu đồ sộ và chi tiết về cách sử dụng App Inventor. Vì vây, nói chung tất cả rất đơn giản chỉ là các thao tác kéo và thả, bao gồm cả phần thiết kế giao diện và viết code.

**Những tính năng có trên MIT App Inventor:**

+ Cho phép xây dựng nhanh chóng những thành phần cơ bản (components) của một ứng dụng Android: Nút bấm, nút lựa chọn, chọn ngày giờ, ảnh, văn bản, thông báo, kéo trượt, trình duyệt web.

+ Sử dụng nhiều tính năng trên điện thoại: Chụp ảnh, quay phim, chọn ảnh, bật video hoặc audio, thu âm, nhận diện giọng nói, chuyển lời thoại thành văn bản, dịch.

+ Hỗ trợ xây dựng game bằng các components: Ball, Canvas, ImageSprite.

+ Cảm biến: đo gia tốc (AccelerometerSensor), đọc mã vạch, tính giờ, con quay hồi chuyển (gyroscopeSensor), xác định địa điểm (locationSensor), NFC, đo tốc độ (pedometer), đo khoảng cách xa gần với vật thể (proximitySensor).

+ Kết nối: Danh bạ, email, gọi điện, chia sẻ thông qua các ứng dụng mạng xã hội khác trên thiết bị, nhắn tin, sử dụng twitter qua API, bật ứng dụng khác, Bluetooth, bật trình duyệt.

+ Lưu trữ: đọc hoặc lưu tệp txt, csv, sử dụng FusiontablesControl, tạo cơ sở dữ liệu đơn giản trên điện thoại hoặc trên đám mây thông qua server tự tạo hoặc Firebase

+ Điều khiển robot thông qua LegoMindstorms.

+ Và rất nhiều mở rộng do các nhà lập trình hoạt động riêng liên tục thêm vào như: mua bán trong ứng dụng, Float button, báo thức, cảm biến ánh sáng, kết nối dữ liệu SQLite,…

**Những nhược điểm chính của App Inventor:**

+ Lập trình viên chưa thể sử dụng mọi tính năng của Android, việc này phụ thuộc vào khi nào mở rộng, mới có tính năng chúng ta cần có được tạo ra. Khuyết điểm này chỉ có thể khắc phục bằng cách tự xây dựng mở rộng cho App Inventor.

+ Vì là website với mục đích giáo dục, MIT App Inventor không hỗ trợ quảng cáo. Chính vì nhược điểm này Thunkable và AppyBuilder được sinh ra.

+ Giao diện chưa chuyên nghiệp.

+ Chuyển mã từ ngôn ngữ Drag and Drop sang Java chưa thực sự dễ dàng.

+ Do ứng dụng được phát triển trên server của MIT, giới hạn dung lượng của mỗi project chỉ là 5mb.

Mặc dù có những nhược điểm như trên, MIT App Inventor vẫn là một nền tảng mạnh mẽ giúp những ai mới bắt đầu lập trình trên Android có thể tạo ra được những ứng dụng hoàn thiện hoặc giúp nhà phát triển chuyên nghiệp nhanh chóng phác thảo lên ý tưởng của mình.

* + 1. Thiết kế giao diện

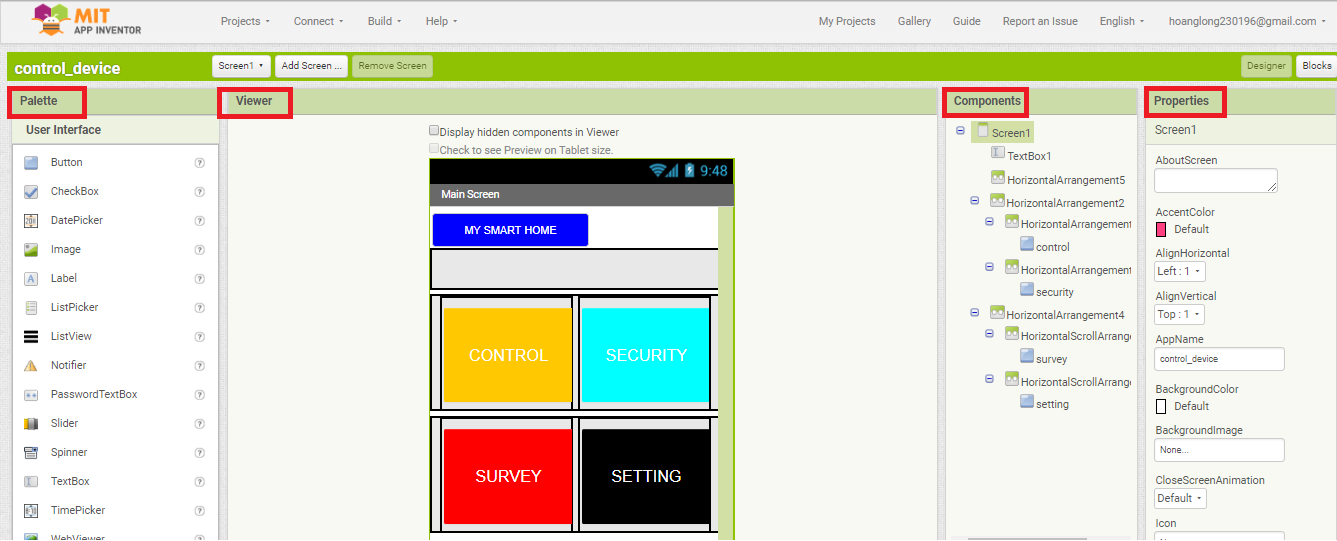
Trên giao diện chính của App Inventor 2, nhìn hàng menu có:

* **Projects**: chứa tất cả các dự án của chúng ta.
* **Connect**: mục này giúp kết nối với các thiết bị khác để xem demo app, các thiết bị này có thể là: giả lập có sẵn trong App Inventor 2, giả lập bên thứ ba (như Genymotion), hoặc là một chiếc smartphone.
* **Build**: sau khi xây dựng app xong, ta xuất ra file .apk để cài lên thiết bị Android.
* **Help**: hỗ trợ người dùng với rất nhiều mục.
* Tiếp theo là các menu hỗ trợ khác: **My Projects**, **Gallery**, **Guide**, **Report an Issue**, **Ngôn ngữ**,…

Cửa sổ thiết kế gồm 4 khung chức năng chính (màu đỏ) (Hình 2-8):

* **Palette**: Chứa các thành phần có thể đặt lên trên Screen như: Button, Label, Image, Listview, Video player,… Đến các thành phần chức năng không nhìn thấy trên Screen như: BLE extension, Notifier, các sensors,…
* **Viewer**: Hiển thị giao diện Screen. Kéo thả các thành phần từ khung Palette sang đây để thiết kế giao diện cho phần mềm của chúng ta.
* **Components**: Sơ đồ cây thể hiện cấu trúc các thành phần đã được bố trí trên Screen.
* **Properties**: Hiển thị thuộc tính của component tương ứng được chọn.

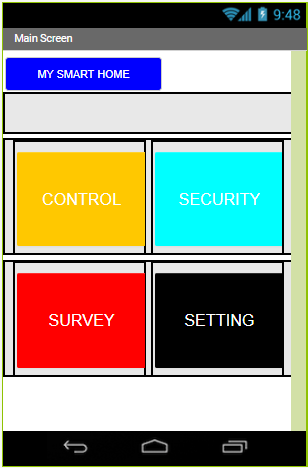
Ngoài ra, còn khung **Media** chứa các file media chúng ta tải lên để sử dụng trong chương trình như: ảnh icon, ảnh nền,…



Hình 2-8 Giao diện thiết kế của MIT App Inventor 2

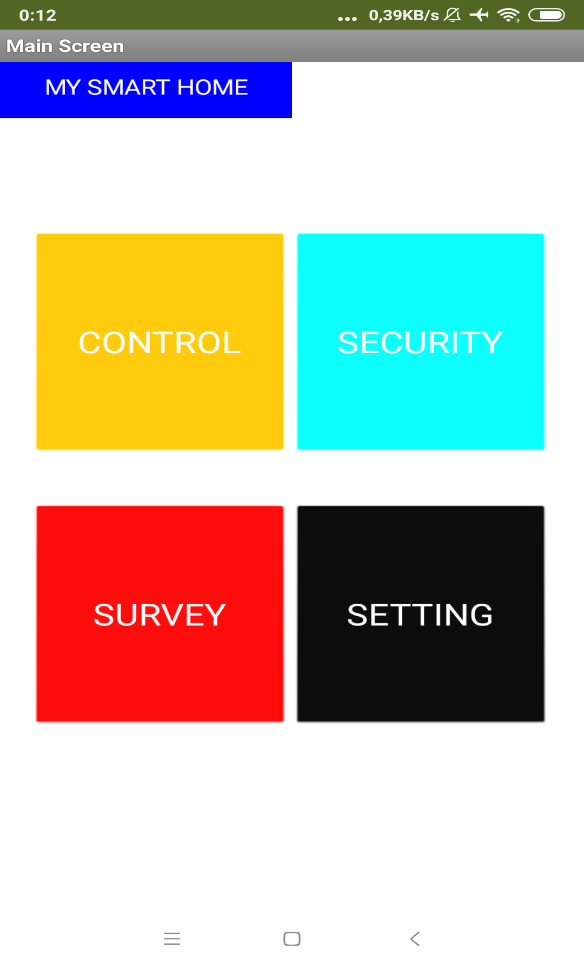
Phần giao diện của chúng ta sẽ gồm nhiều Screen: Screen 1, Screen 2, Screen 3,… tùy thuộc vào mình muốn tạo bao nhiêu Screen. Trong đó, Screen 1 luôn là màn hình chính, màn hình đầu tiên xuất hiện khi chúng ta mở app lên. Do đó, lúc mới vào app thì chúng ta sẽ gặp Screen 1, tức là giao diện chính của app (Hình 2.9)

Ở giao diện chính này, chúng ta bố trí các chức năng chính của App như: CONTROL, SECURITY, SURVEY, SETTING,…

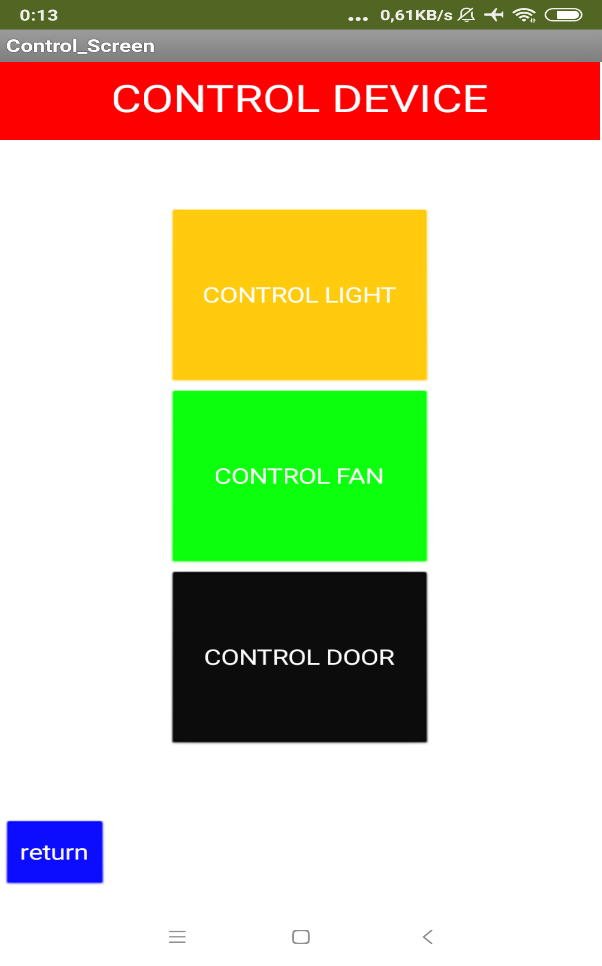


Hình 2-9 Screen 1 của App

Dưới đây là các giao diện Screen sau khi build và chạy thực tế trên App



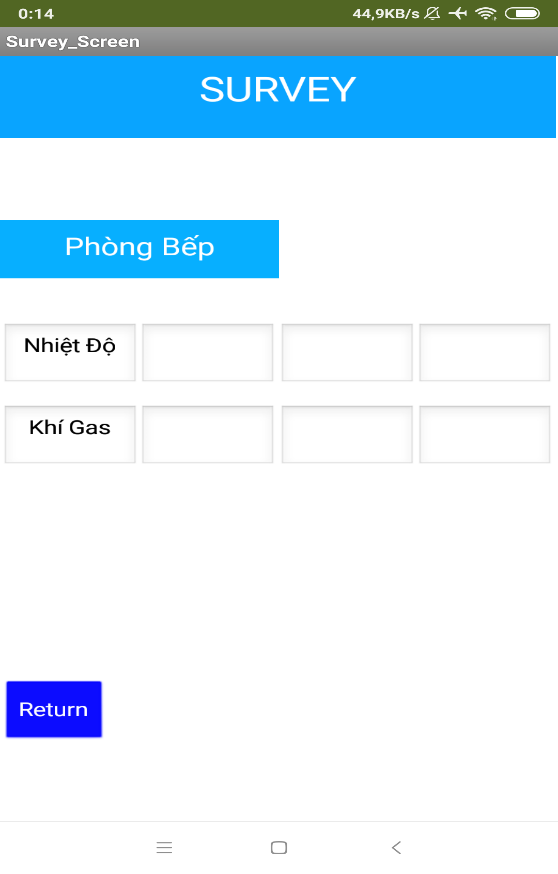
Hình 2-10 Main Screen



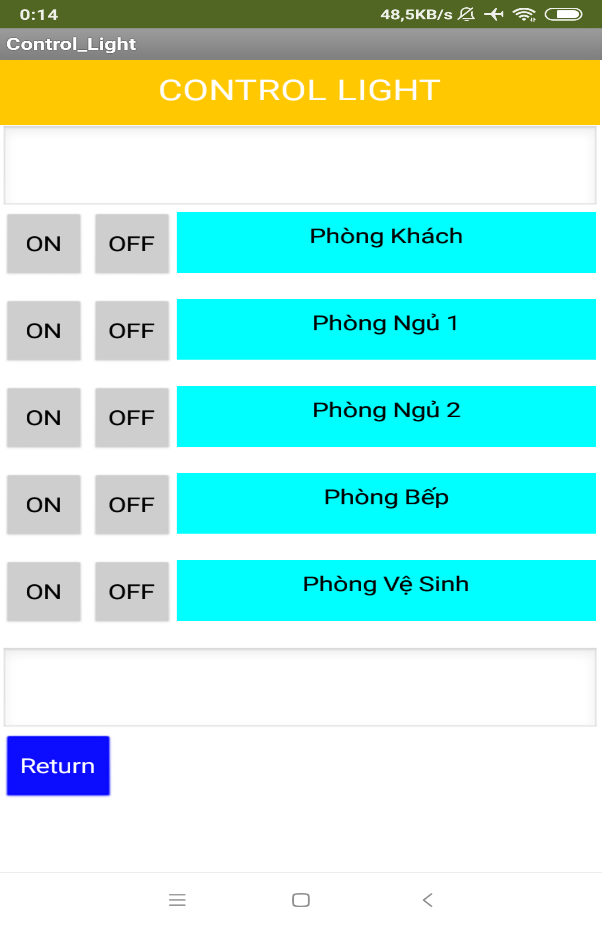
Hình 2-11 Control-device Screen



Hình 2-12 Security Screen



Hình 2-13 Survey Screen



Hình 2-14 Control-light Screen



Hình 2-15 Control-fan Screen

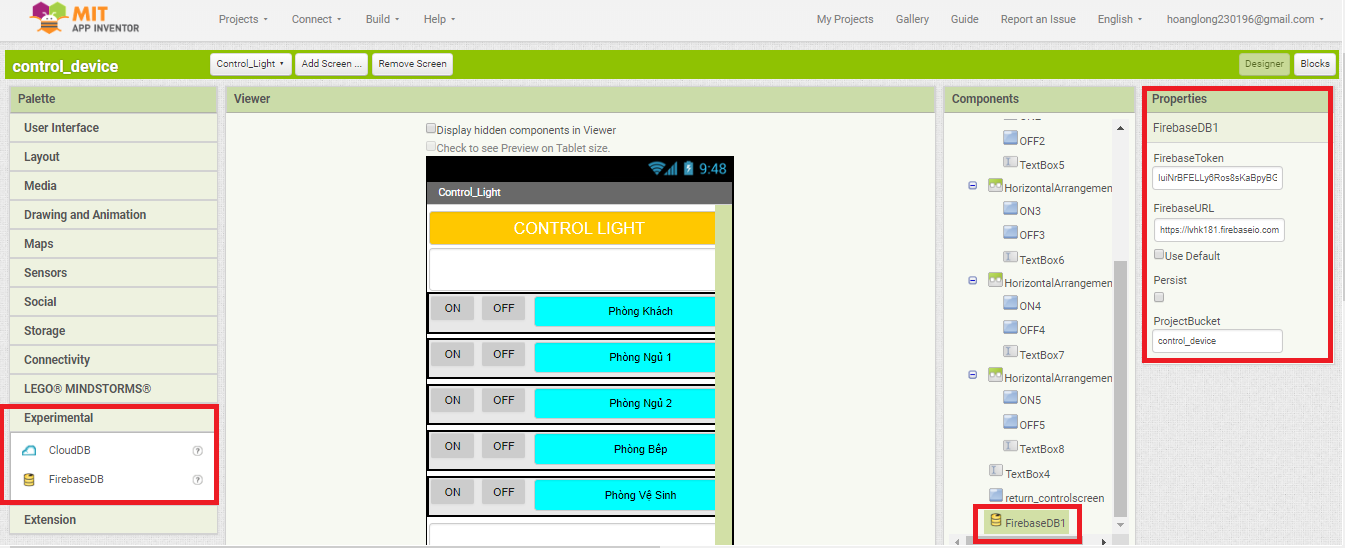


Hình 2-16 Control-door Screen

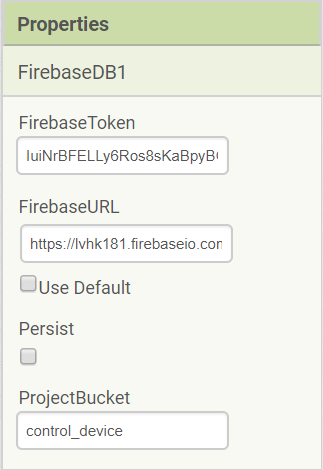
### Lập trình chức năng cho app

Sau khi thiết kế xong giao diện, để các thành phần của giao diện có thể hoạt động, thì chúng ta phải lập trình cho chúng.

Trước khi lập trình, ta phải thêm FirebaseDxx (sẽ đề cập ở phần sau) vào bên giao diện thiết kế (**designer**) ở những phần cần điều khiển hoặc giám sát.(Hình 2-17, Hình 2-18, Hình 2-19)

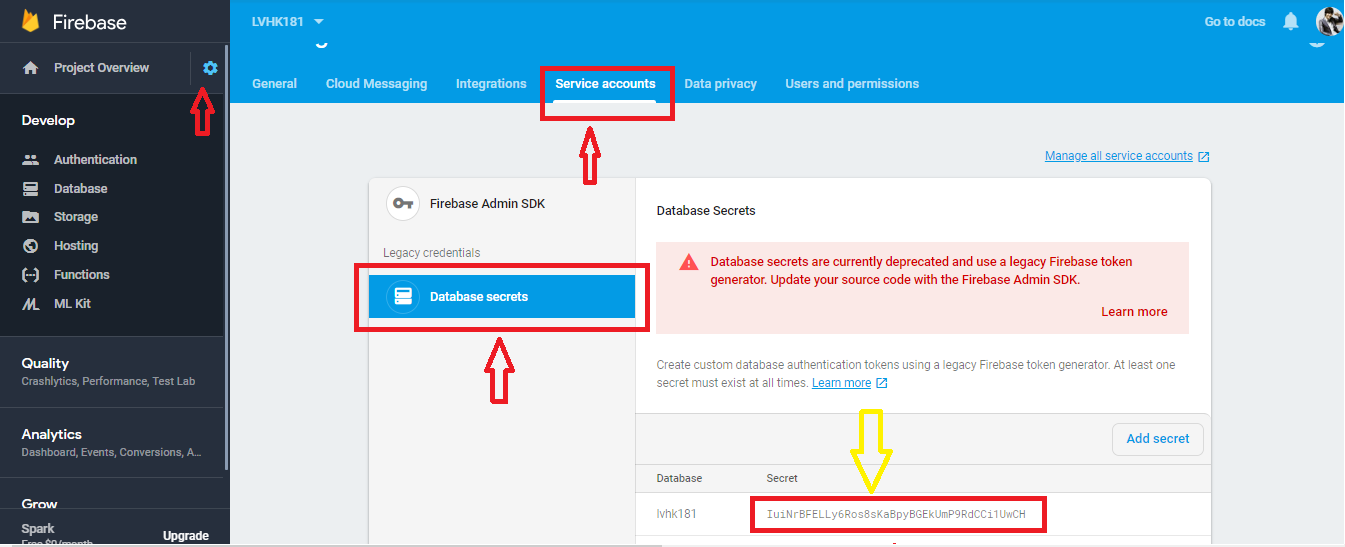


Hình 2-17 Add Firebase vào MIT App Inventor 2



Hình 2-18 Mục Properties khi thêm Firebase

Chú ý, tại mục **Properties** ta thêm vào ô “FirebaseURL” là đường link khi đã tạo Firebase ( <https://lvhk181.firebaseio.com/> sẽ hướng dẫn ở phần sau ). Tại ô “FirebaseToken” thì sẽ điền tại mũi tên màu vàng (Hình 2-19). Còn ô “ProjectBucket” tùy ý thích của mình.



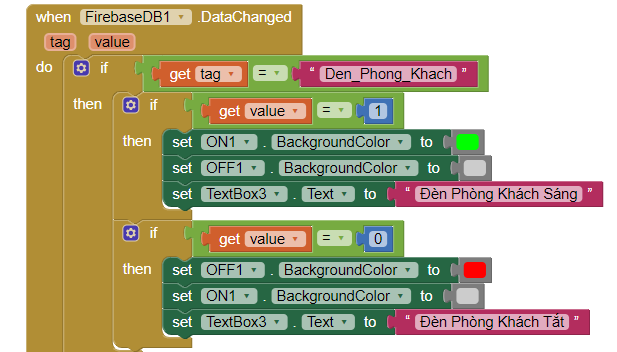
Hình 2-19 Mã của FirebaseToken

Chuyển sang mục **Blocks**, ta tiến hành kéo thả các thuật toán để lập trình chức năng cho App

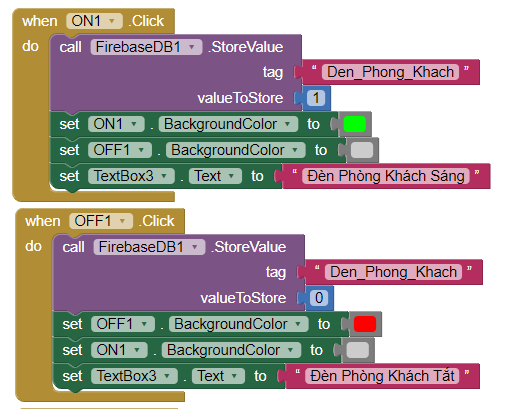
Dưới đây là các thuật toán lấy dữ liệu từ Firebase Google về App và thuật toán gửi dữ liệu từ App lên Firebase Google khi App hoạt động:

+ Giải thích thuật toán (Hình 2-20) như sau:

* Trên Firebase ta có các giá trị khi điều khiển offline sẽ được gán vào các biến, giả sử ở đây là biến “Den\_Phong\_Khach”. Vậy muốn App đồng nhất với chế độ offline, thì ta dùng lệnh **get tag = “Den\_Phong\_Khach”**, nghĩa là lấy giá trị từ biến “Den\_Phong\_Khach”
* Sau đó so sánh với giá trị ta mong muốn (cài đặt trước), ở đây là giá trị “1” hoặc giá trị “0”; nếu là “1” thì nút “ON1” sẽ bật (màu xanh lá cây) còn nút “OFF1” sẽ tắt (màu xám), đồng thời gửi về trạng thái của đèn lúc đó; ngược lại nếu là “0” thì nút “OFF1” sẽ bật (màu đỏ) còn nút “ON1” sẽ tắt (màu xám), đồng thời cũng gửi về trạng thái của đèn lúc đấy.



Hình 2-20 Thuật toán lấy dữ liệu từ Firebase để App đồng nhất với chế độ offline



Hình 2-21 Thuật toán gửi lệnh điều khiển từ App

+ Giải thích thuật toán (Hình 2-21) như sau:

* Khi nút “ON1” trên App được nhấn thì giá trị của biến “Den\_Phong\_Khach” trên Firebase sẽ được set lên “1” thông qua khối lệnh **call FirebaseDB1. StoreValue**. Đồng thời cũng hiển thị lên App bằng cách cho nút “ON1” sẽ bật (màu xanh lá cây), nút “OFF1” sẽ tắt (màu xám) và gửi về trạng thái của đèn lúc này.
* Khi nút “OFF1” trên App được nhấn thì giá trị của biến “Den\_Phong\_Khach” trên Firebase sẽ được set xuống “0” thông qua khối lệnh **call FirebaseDB1. StoreValue**. Đồng thời cũng hiển thị lên App bằng cách cho nút “ON1” sẽ tắt (màu xám), nút “OFF1” sẽ bật (màu đỏ) và gửi về trạng thái của đèn lúc này.

## Tìm hiểu về Firebase Google

### Tìm hiểu tổng quan Firebase

Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu. Nếu cần xây dựng một ứng dụng cho mobile hoặc các thiết bị di động khác, mà chúng ta đang gặp khó khăn vì không biết chọn dịch vụ VPS nào, loại database gì,… thì Firebase sẽ là dịch vụ dành cho chúng ta.

Firebase có thể rất mạnh mẽ đối với ứng dụng backend, nó bao gồm việc lưu trữ dữ liệu, xác thực người dùng, static hosting,… Nên lập trình viên chỉ cần chú tâm đến việc nâng cao trải nghiệm người dùng.

Trang chủ của Firebase Google: <https://firebase.google.com>

Điểm qua một số thông tin cơ bản về dịch vụ này:

* Thành lập: tháng 9 năm 2011
* Tổ chức mẹ: Google
* Trụ sở: San Francisco, California, Hoa Kỳ
* Nhà sáng lập: Andrew Lee, James Tamplin
* Công ty con: Fabric
* **Các tính năng chính, thông dụng của Firebase:**

Sau nhiều năm xây dựng và phát triển, cho đến thời điểm hiện nay, Firebase đã ra mắt rất nhiều tính năng mới. Tuy nhiên, chúng ta chỉ nêu ra các tính năng thông dụng, cũng như là các tính năng mà trong phạm vi luận văn này dùng tới, đó là:

+ Lưu trữ dữ liệu thời gian thực (Realtime Database).

+ Xác thực người dùng.

+ Firebase hosting.

Cụ thể, chúng ta sẽ tìm hiểu chi tiết hơn về mỗi tính năng:

* **Realtime Database**
* Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng hồ database tới tất cả các client theo thời gian thực. Cụ thể hơn là chúng ta có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.
* Tự động tính toán quy mô ứng dụng của chúng ta, giúp chúng ta dễ dàng hơn rất nhiều mỗi khi cần nâng cấp hay mở rộng dịch vụ. Ngoài ra Firebase sử dụng NoSQL, giúp cho database của chúng ta sẽ không bị bó buộc trong các bảng và các trường mà chúng ta có thể tùy ý xây dựng database theo cấu trúc của riêng mình.
* Cho phép chúng ta phân quyền một cách đơn giản bằng cú pháp tương tự như javascript. Khi ứng dụng của chúng ta muốn phát triển, chúng ta không cần lo lắng về việc nâng cấp máy chủ…Firebase sẽ xử lý việc tự động cho chúng ta. Các máy chủ của Firebase quản lý hàng triệu kết nối đồng thời và hàng tỉ lượt truy vấn mỗi tháng.
* **Các tính năng bảo mật**
* Firebase hoạt động dựa trên nền tảng cloud và thực hiện kết nối thông qua giao thức bảo mật SSL, chính vì vậy chúng ta sẽ bớt lo lắng rất nhiều về việc bảo mật của dữ liệu cũng như đường truyền giữa client và server. Không chỉ có vậy, việc cho phép phân quyền người dùng database bằng cú pháp javascipt cũng nâng cao hơn nhiều độ bảo mật cho ứng dụng của chúng ta, bởi chỉ những user mà chúng ta cho phép mới có thể có quyền chỉnh sửa cơ sở dữ liệu.
* **Làm viêc offline**
* Ứng dụng Firebase của chúng ta sẽ duy trì tương tác bất chấp một số các vấn đề về internet xảy ra. Trước khi bất kỳ dữ liệu được ghi đến server thì tất cả dữ liệu lập tức sẽ được viết vào một cơ sử dữ liệu Firebase ở local. Ngay khi có thể kết nối lại, client đó sẽ nhận bất kỳ thay đổi mà nó thiếu và đồng bộ hoá nó với trạng thái hiện tại server.
* **Xác thực người dùng**
* Với Firebase, chúng ta có thể dễ dàng xác thực người dùng từ ứng dụng của chúng ta trên Android, iOS và JavaScript SDKs chỉ với một vài đoạn mã. Firebase đã xây dựng chức năng cho việc xác thực người dùng với Email, Facebook, Twitter, GitHub, Google, và xác thực nặc danh. Các ứng dụng sử dụng chức năng xác thực của FireBase có thể giải quyết được vấn đề khi người dùng đăng nhập, nó sẽ tiết kiện thời gian và rất nhiều các vấn đề phức tạp về phần backend. Hơn nữa chúng ta có thể tích hợp xác thực người dùng với các chức năng backend đã có sẵn dùng custom auth tokens.
* **Firebase Hosting**
* Chúng ta có thể triển khai một ứng dụng nền web chỉ với vài giây với hệ thống Firebase, và các dữ liệu sẽ được lưu trữ đám mây đồng thời được bảo mật thông qua giao thức truy cập SSL.
* Các ứng dụng sẽ được cấp 1 tên miền dạng \*.firebaseio.com hoặc chúng ta có thể trả tiền để sử dụng tên miền của riêng mình.
* **Triến khai siêu tốc**
* Với Firebase, chúng ta có thể giảm bớt rất nhiều thời gian cho việc viết các dòng code để quản lý và đồng bộ cơ sở dữ liệu, mọi việc sẽ diễn ra hoàn toàn tự động với các API của Firebase. Không chỉ có vậy, Firebase còn hỗ trợ đa nền tảng nên chúng ta sẽ càng đỡ mất thời gian rất nhiều khi ứng dụng chúng ta muốn xây dựng là ứng dụng đa nền tảng. Không chỉ nhanh chóng trong việc xây dựng database, Google Firebase còn giúp ta đơn giản hóa quá trình đăng kí và đăng nhập vào ứng dụng bằng các sử dụng hệ thống xác thực do chính Firebase cung cấp.
* **Sự ổn định**
* Firebase hoạt động dựa trên nền tảng Cloud đến từ Google. Vì vậy, hầu như chúng ta không bao giờ phải lo lắng về việc sập server, tấn công mạng như DDOS, tốc độ kết nối lúc nhanh lúc chậm nữa.
* **Giá thành**
* Google Firebase có rất nhiều gói dịch vụ với các mức dung lượng lưu trữ cũng như băng thông khác nhau với mức giá dao động từ Free đến $1500 đủ để đáp ứng được nhu cầu của tất cả các đối tượng. Chính vì vậy, chúng ta có thể lựa chọn gói dịch vụ phù hợp nhất với nhu cầu của mình. Điều này giúp chúng ta tối ưu hóa được vốn đầu tư và vận hành của mình tùy theo số lượng người sử dụng. Ngoài ra, chúng ta còn không mất chi phí để bảo trì, nâng cấp, khắc phục các sự cố…

Với nền tảng web và ngôn ngữ chính là javascript; để đi sâu hơn, Firebase cung cấp và hỗ trợ cho ta:

+ Về thư viện firebase hỗ trợ chính thức:

* GeoFire (Geo Location).
* GularFire (AngularJS).
* BerFire (Ember).
* ReactFire (ReactJS).
* Ionic.

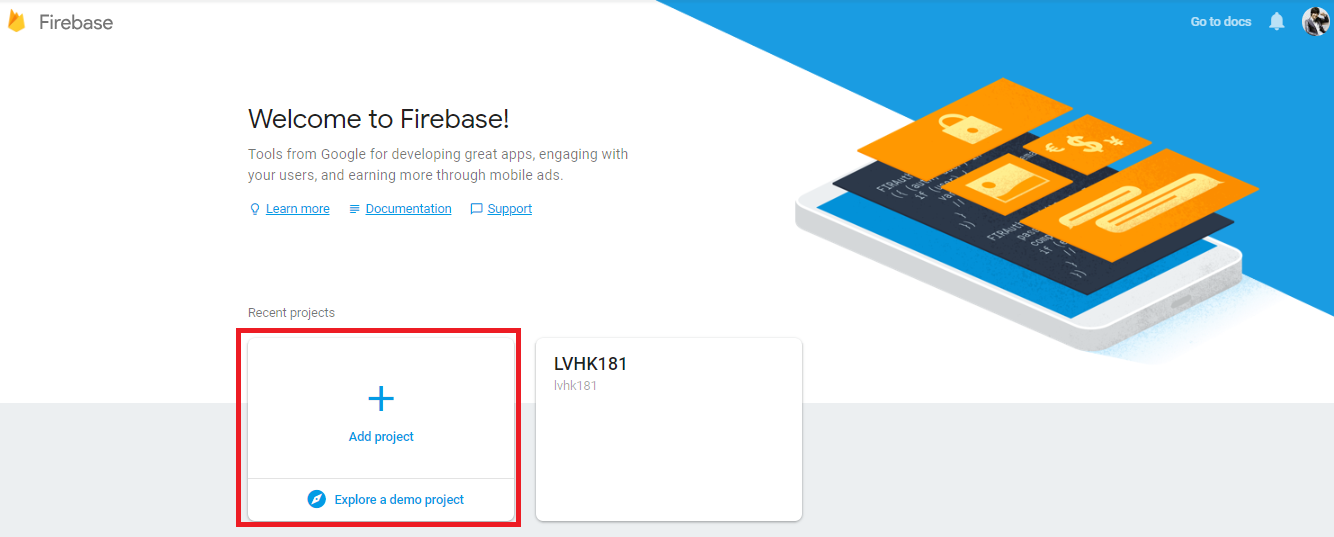
+ Hoặc thông qua bên thứ 3:

* FirebaseIndex.
* Sublime Text 2 Snippets.
* Polymer Firebase-Element (Polymer).
* VueFire (Vue.Js).
* Ember-Model-Firebase-Adapter (Ember).
  + 1. Tìm hiểu về Realtime Database

Trong các tính năng vừa kể ở trên, thì Realtime Database (Dữ liệu thời gian thực) là một tính năng rất nổi trội. Và trong luận văn này, chúng ta sẽ sử dụng tính năng này để thực hiện đề tài.

Trước tiên, chúng ta cần có một tài khoản để sử dụng Firebase. Việc đăng kí tài khoản này là miễn phí. Để đăng kí tài khoản Firebase, chúng ta cần có một tài khoản gmail. Việc đăng kí và cài đặt cho Firebase rất đơn giản:

Ta truy cập địa chỉ sau: <https://firebase.google.com> . Sau đó đăng nhập bằng tài khoản gmail, chúng ta sẽ được chuyển tới trang sau: <https://console.firebase.google.com> . Sau đó, ta tạo một project mới để thực hiện:



Hình 2-22 Tạo một project trên Firebase

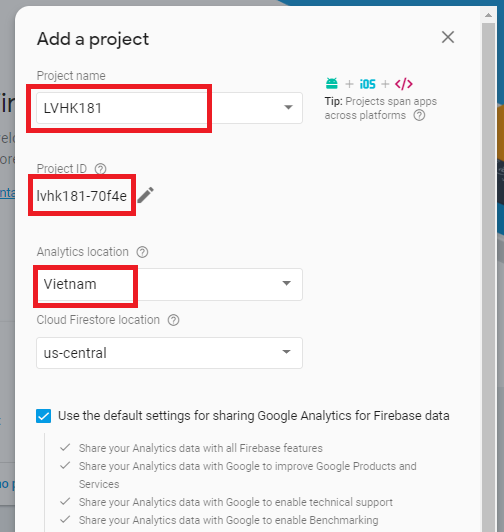
Tiếp tục với các bước sau:

+ Đặt tên cho project.

+ Tạo một ID cho project (mỗi project có một ID khác nhau).

+ Chọn quốc gia.

Tiếp theo ta tick vào ô “ I accept…”, sau đó nhấn nút “Create project”



Hình 2-23 Add một project

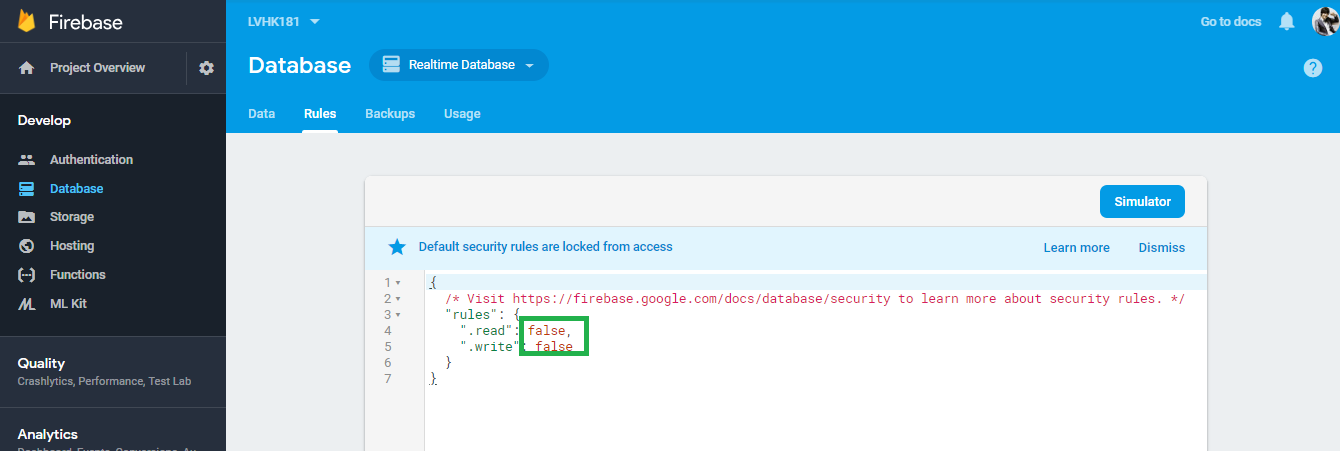
Sau khi tạo project thành công, ta vào mục Database để tạo mới Realtime Database:



Hình 2-24 Mục Database của project

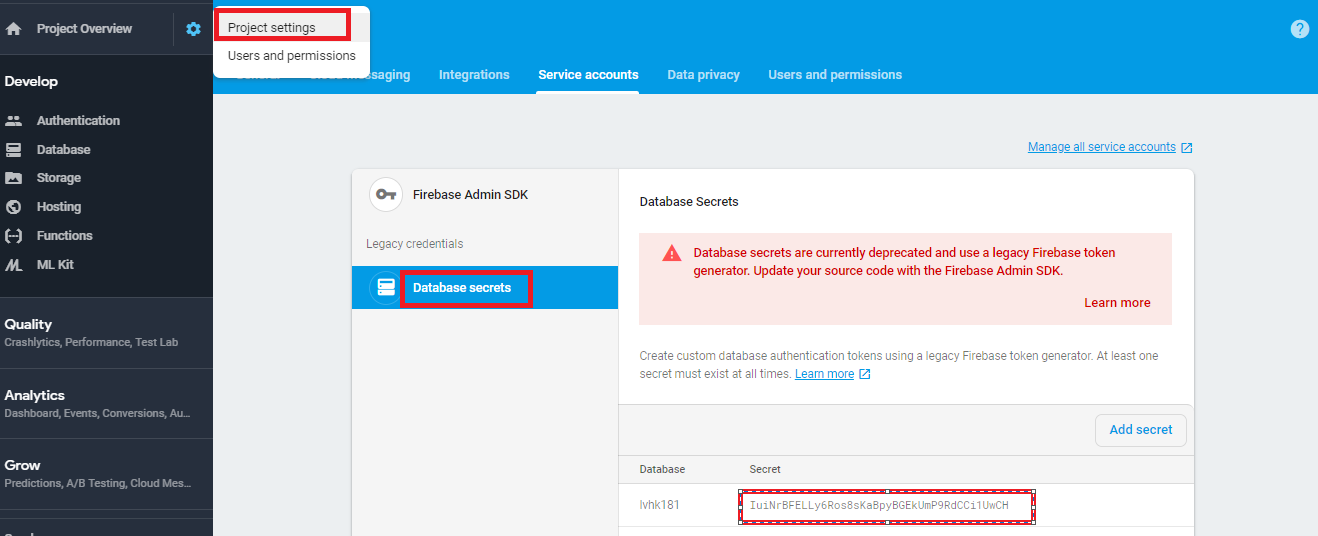
Khi có được một Database, ta sẽ cài đặt rules tùy theo mong muốn của bản thân

+ Nếu chỉnh rules: “false” thành “true” sau đó publish thì Database sẽ cho phép bất kỳ ai cũng có thể ghi vào, thuận tiện trong quá trình học và debug, vì thế không có tính bảo mật.



Hình 2-25 Tab rules trong mục Database

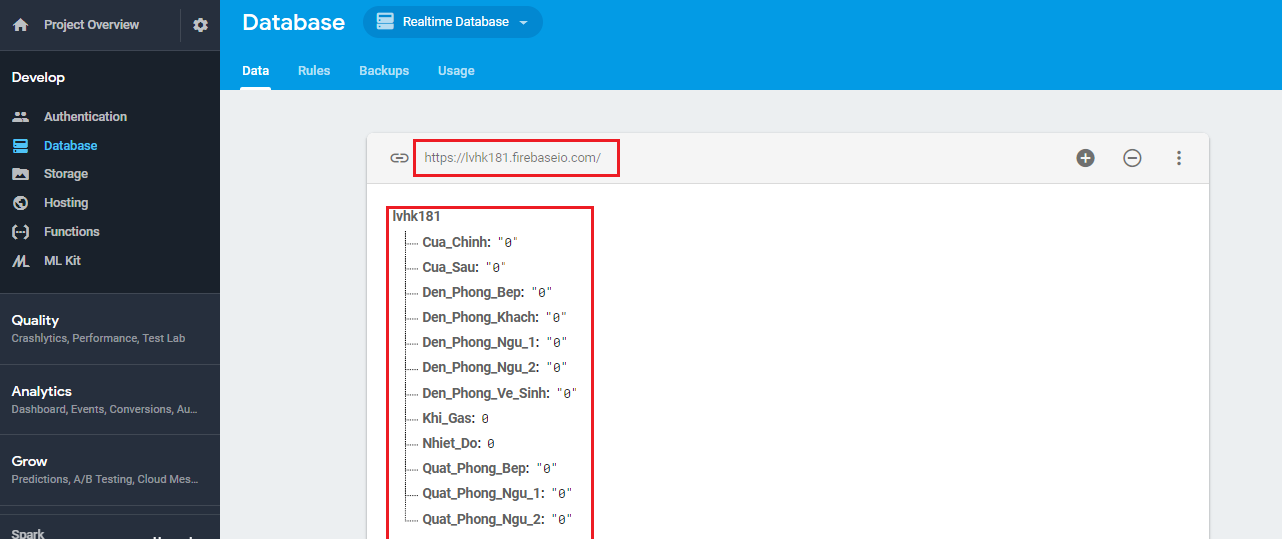
+ Nếu để rules như cũ, tức là “false” thì sẽ bảo mật hơn với một mã nằm trong mục Database Secrects



Hình 2-26 Mã cho phép truy cập vào Firebase

Sau khi tạo project thành công, chúng ta sẽ có một đường link dùng để kết nối với các ứng dụng, cụ thể ở đây là: <https://lvhk181.firebaseio.com/>

Hình dưới đây là dữ liệu đầy đủ khi ta đẩy dữ liệu lên Firebase:

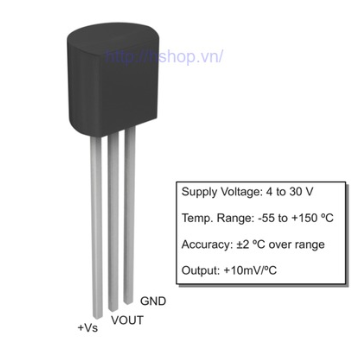


Hình 2-27 Theo dõi dữ liệu trên Firebase

## Tìm hiểu về các linh kiện

* + 1. Các loại cảm biến

**Cảm biến nhiệt độ LM35:**



Hình 2-28 Cảm biến nhiệt độ LM35

**+ Mô tả:**

Cảm biến LM35 là bộ cảm biến nhiệt mạch tích hợp chính xác cao, mà điện áp đầu ra của nó tỷ lệ tuyến tính với nhiệt độ theo thang độ Celsius. Chúng cũng không yêu cầu cân chỉnh ngoài vì vốn chúng đã được cân chỉnh. Cảm biến LM35 hoạt động bằng cách cho ra một giá trị hiệu điện thế nhất định tại chân Vout (chân giữa) ứng với mỗi mức nhiệt độ.

+ **Thông số kỹ thuật:**

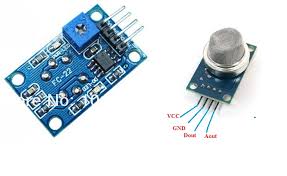
* Điện áp đầu vào từ 4V đến 30V.
* Điện áp ra: -1V đến 6V.
* Công suất tiêu thụ là 60uA.
* Độ phân giải điện áp đầu ra là 10mV/℃.
* Độ chính xác cao ở 25℃ là 0.5℃.
* Trở kháng đầu ra thấp  0.1 cho 1mA tải.
* Độ chính xác thực tế: 1/4℃ ở nhiệt độ phòng và 3/4℃ ngoài khoảng -55℃ tới 150℃.

**+ Ví dụ:**

Dải nhiệt độ đo được của LM35 là từ -55℃ đến 150℃ với các mức điện áp ra khác nhau. Xét một số mức điện áp sau:

* Nhiệt độ -55℃ điện áp đầu ra -550 mV .
* Nhiệt độ 25℃ điện áp đầu ra 250 mV
* Nhiệt độ 150℃ điện áp đầu ra 1500 mV

**Cảm biến khói-khí gas (MQ-2):**



Hình 2-29 Cảm biến khói-khí gas MQ-2

**+ Mô tả:**

MQ-2 sử dụng phần tử SnO2 có độ dẫn điện thấp hơn trong không khí sạch. Khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của SnO2 sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện. MQ-2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan (CH4) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau.

**+ Thông số kỹ thuật:**

* Nguồn hoạt động: 5V
* Loại dữ liệu: cảm biến xuất ra cả hai dạng tín hiệu Analog và Digital, tín hiệu Digital có thể điều chỉnh mức báo bằng biến trở.
* Phạm vi phát hiện rộng.
* Tốc độ phản hồi nhanh và độ nhạy cao.
* Mạch đơn giản.
* Ổn định khi sử dụng trong thời gian dài.

**Cảm biến thân nhiêt chuyển động PIR HC-SR501:**



Hình 2-30 Cảm biến chuyển động

**+ Mô tả:**

Cảm biến thân nhiệt chuyển động PIR (Passive infrared sensor) HC-SR501 được sử dụng để phát hiện chuyển động của các vật thể phát ra bức xạ hồng ngoại (con người, con vật, các vật phát nhiệt,...), cảm biến có thể chỉnh được độ nhạy để giới hạn khoảng cách bắt xa gần cũng như cường độ bức xạ của vật thể mong muốn, ngoài ra cảm biến còn có thể điều chỉnh thời gian kích trễ (giữ tín hiệu bao lâu sau khi kích hoạt) qua biến trở tích hợp sẵn.

Cảm biến thân nhiệt chuyển động PIR HC-SR501 có cảm biến, thấu kính và board mạch chất lượng tốt cho độ nhạy và độ bền cao nhất.

**+ Thông số kỹ thuật:**

* 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.
* Kích thước: 1,27 x 0,96 x 1.0 ( 32,2 x 24,3 x 25,4 mm).

**Cảm biến mưa:**



Hình 2-31 Cảm biến mưa

**+ Mô tả:**

* Raindrops và bộ điều khiển riêng biệt, dễ dàng nối dây.
* Tấm cảm biến mưa lớn, thuận lợi để phát hiện mưa.
* Board có lỗ định vị dễ dàng lắp đặt.
* LM393 khoảng so sánh điện áp rộng.

**+ Thông số kỹ thuật:**

* Kích thước tấm cảm biến mưa: 54 x 40mm
* Kích thước board PCB: 30 x 16mm
* Điện áp: 5V
* Đầu ra: đầu ra kỹ thuật số (0 và 1) và đầu ra tương tự điện áp A0
* Lỗ cố định bu lông dễ dàng để cài đặt
* Có đèn báo hiệu nguồn và đầu ra
* Đầu ra TTL, tín hiệu đầu ra TTL có giá trị thấp. Có thể điều khiển trực tiếp relay, buzzer, a small fan...
* Độ nhạy có thể được điều chỉnh thông qua chiết áp
* LED sáng lên khi không có mưa đầu ra cao, có mưa, đầu ra thấp LED tắt.

**+ Chế độ kết nối:**

* VCC: Nguồn
* GND: Đất
* D0: Đầu ra tín hiệu TTL chuyển đổi
* A0: Đầu ra tín hiệu Analog
  + 1. Động cơ RC servo 9G



Hình 2-32 Động cơ RC servo 9G

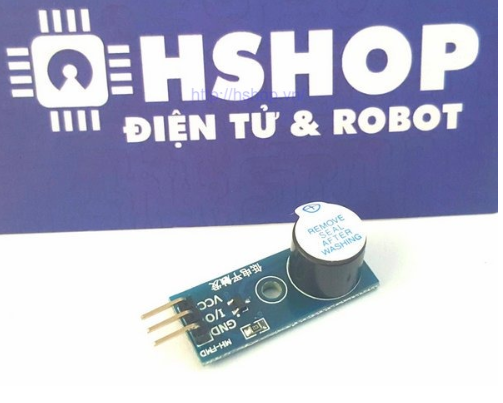
**+ Mô tả:**

Động cơ RC Servo 9G có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng. Động cơ RC Servo 9G có tốc độ phản ứng nhanh, các bánh răng được làm bằng nhựa nên cần lưu ý khi nâng tải nặng vì có thể làm hư bánh răng, động cơ RC Servo 9G có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

**+ Thông số kỹ thuật:**

* Điện áp hoạt động: 4.8-5VDC
* Tốc độ: 0.12 sec/ 60 degrees (4.8VDC)
* Lực kéo: 1.6KG.CM
* Kích thước: 21x12x22mm
* Trọng lượng: 9g.
  + 1. Các thiết bị khác

**Mạch còi Buzzer:**



Hình 2-33 Mạch còi Buzzer

**+ Mô tả:**

Mạch còi buzzer được sử dụng để phát ra âm thanh khi kích tín hiệu, ứng dụng trong các hệ thống báo hiệu, báo trộm…

**+ Thông số kỹ thuật:**

* Điện áp sử dụng: 3.3~5VDC
* Tín hiệu kích: TTL mức thấp Low 0VDC.
* Kích thước: 32 x 13 mm

**Cảm ứng 1 chạm điện dung TTP223B Mini:**



Hình 2-34 Cảm ứng 1 chạm điện dung

**+ Mô tả:**

Cảm ứng 1 chạm điện dung TTP223B mini có kích thước nhỏ gọn và giá thành rẻ, thường được sử dụng trong các ứng dụng cảm ứng điện dung: bàn phím, công tắc cảm ứng, báo động…

**+ Thông số kỹ thuật:**

* IC chính: TTP223B
* Điện áp làm việc: 3 ~ 5VDC
* Dòng điện tiêu thụ: 0.025mA
* Cảm ứng xuyên qua các phi kim như kính, nhựa, acrylic, ...

**+ Cách sử dụng:**

* Cấp điện 3~5VDC vào chân VCC và GND.
* Nếu có thao tác chạm cảm ứng chân OUT sẽ xuất tín hiệu và đèn Led sẽ sáng báo hiệu.

**Mạch RFID RC522 NFC 13.56Mhz:**



Hình 2-35 RFID

**+ Mô tả:**

Module RFID RC522 sử dụng IC MFRC522 của Phillip dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13.56mhz, với mức giá rẻ thiết kế nhỏ gọn, module này là sự lựa chọn hàng đầu cho các ứng dụng về ghi đọc thẻ RFID.

**+ Thông số kỹ thuật:**

* Nguồn: 3.3VDC, 13-26mA
* Dòng ở chế độ chờ: 1013mA
* Dòng ở chế độ nghỉ: <80uA
* Tần số sóng mang: 13.56Mhz
* Khoảng cách hoạt động: 0～60mm（mifare1 card）
* Giao tiếp: SPI
* Tốc độ truyền dữ liệu: tối đa 10Mbit/s
* Các loại card RFID hỗ trợ: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire.
* Kích thước: 40mm × 60mm

**LCD Text – LCD1602:**



Hình 2-36 LCD1602

**+ Mô tả:**

Màn hình text LCD1602 xanh dương sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ sử dụng thích hợp cho những người mới học và làm dự án.

**+ Thông số kỹ thuật:**

* Điện áp hoạt động là 5 V.
* Kích thước: 80 x 36 x 12.5 mm
* Chữ trắng, nền xanh dương
* Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
* Tên các chân được ghi ở mặt sau của màn hình LCD hổ trợ việc kết nối, đi dây điện.
* Có đèn led nền, có thể dùng biến trở hoặc PWM điều chình độ sáng để sử dụng ít điện năng hơn.
* Có thể được điều khiển với 6 dây tín hiệu.
* Có bộ ký tự được xây dựng hổ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet .

**LCD Text - LCD2004:**



Hình 2-37 LCD2004

**+ Mô tả:**

Màn hình text LCD2004 xanh dương sử dụng driver HD44780, có khả năng hiển thị 4 dòng với mỗi dòng 20 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ sử dụng thích hợp cho những người mới học và làm dự án.

**+ Thông số kỹ thuật:**

* Điện áp hoạt động là 5 V.
* Kích thước: 98 x 60 x 13.5 mm
* Chữ trắng, nền xanh dương.
* Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
* Tên các chân được ghi ở mặt sau của màn hình LCD hổ trợ việc kết nối, đi dây điện.
* Có đèn led nền, có thể dùng biến trở hoặc PWM điều chình độ sáng để sử dụng ít điện năng hơn.
* Có thể được điều khiển với 6 dây tín hiệu.
* Có bộ ký tự được xây dựng hổ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet để biết thêm chi tiết.

**Mạch chuyển giao tiếp LCD sang I2C:**

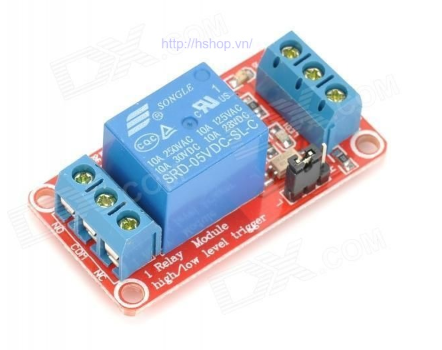


Hình 2-38 Mạch chuyển giao tiếp LCD sang I2C

**+ Mô tả:**

Để sử dụng các loại LCD có driver là HD44780 (LCD 1602, LCD 2004, ... ), cần có ít nhất 6 chân của MCU kết nối với các chân RS, EN, D7, D6, D5 và D4 để có thể giao tiếp với LCD. Nhưng với module chuyển giao tiếp LCD sang I2C, các bạn chỉ cần 2 chân (SDA và SCL)  của MCU kết nối với 2 chân (SDA và SCL) của module là đã có thể hiển thị thông tin  lên LCD. Ngoài ra có thể điều chỉnh được độ tương phản bởi biến trở gắn trên module.

**Module 1 relay với Opto cách ly kích H/L (5VDC):**



Hình 2-39 Module 1 relay (5VDC)

**+ Mô tả:**

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.

Tiếp điểm đóng ngắt gồm 3 tiếp điểm NC (thường đóng), NO (thường mở) và COM (chân chung) được cách ly hoàn toàn với board mạch chính, ở trạng thái bình thường chưa kích NC sẽ nối với COM, khi có trạng thái kích COM sẽ chuyển sang nối với NO và mất kết nối với NC.

**+ Thông số kỹ thuật:**

* Sử dụng điện áp nuôi DC 5V.
* Relay mỗi 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 tín hiệu kích 0 hoặc 1 qua jumper.
* Kích thước: 1.97 in x 1.02 in x 0.75 in (5.0 cm x 2.6 cm x 1.9 cm)
* Weight: 0.60 oz (17 g)

**Module 8 relay với Opto cách ly (5VDC):**



Hình 2-40 Module 8 relay (5VDC)

**+ Mô tả:**

Module 8 relay thích hợp cho các ứng dụng đóng ngắt điện thế cao AC hoặc DC, các thiết bị tiêu thụ dòng lớn, module thiết kế nhỏ gọn, có opto và transistor cách ly, kích đóng bằng mức thấp (0V) phù hợp với mọi loại MCU và thiết kế có thể sử dụng nguồn ngoài giúp cho việc sử dụng trở nên thật linh động và dễ dàng.

**+ Thông số kỹ thuật:**

* Sử dụng điện áp nuôi 5VDC.
* 8 Relay đóng ngắt ở điện thế kích bằng 0V nên có thể sử dụng cho cả tín hiệu 5V hay 3v3 (cần cấp nguồn ngoài), mỗi 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.

**Mạch giảm áp DC XL4005 (5A):**



Hình 2-41 Mach giảm áp DC XL4005 (5A)

**+ Mô tả:**

Mạch chuyển đổi điện áp DC-DC đầu ra 5A với kích thước nhỏ gọn, thay thế cho XL4005 và đảm bảo được hiệu suất tốt, dùng để thiết kế các mạch nguồn cho các thiết bị nhỏ như mô hình, thiết bị di động, camera…

**+ Thông số kỹ thuật:**

* Điện áp đầu vào: từ 5V đến 32V.
* Điện áp đầu ra: điều chỉnh từ 1.25V đến 30V.
* Dòng đỉnh tối đa là 5A ( dòng trung bình 3.5A).
* Độ gợn sóng: 30mV
* Tần số 300KHz
* Nhiệt độ làm việc: -40° C đến 85° C
* Kích thước: dài 43-rộng 21-cao 14 mm.

# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

## Yêu cầu thiết kế

* Nhiệm vụ luận văn yêu cầu thiết kế một hệ thống vừa có thể điều khiển thiết bị bằng tay, vừa bằng App; cảm biến, cảnh báo và hiển thị vừa lên App, vừa ra LCD; có thể tự động điều khiển theo hoàn cảnh.
* Phần cứng gồm các phần quan trọng như: vi điều khiển ( kit STM32F4 Discovery và NodeMCU), thiết bị để điều khiển (đèn,quạt,động cơ,…) và các cảm biến. Chúng ta sẽ kết hợp các phần này lại với nhau để tạo nên một khối thống nhất. Phần đi dây từ vi điều khiển ra các linh kiện, ta thực hiện mạch in qua công cụ allegro orcad 16.6.

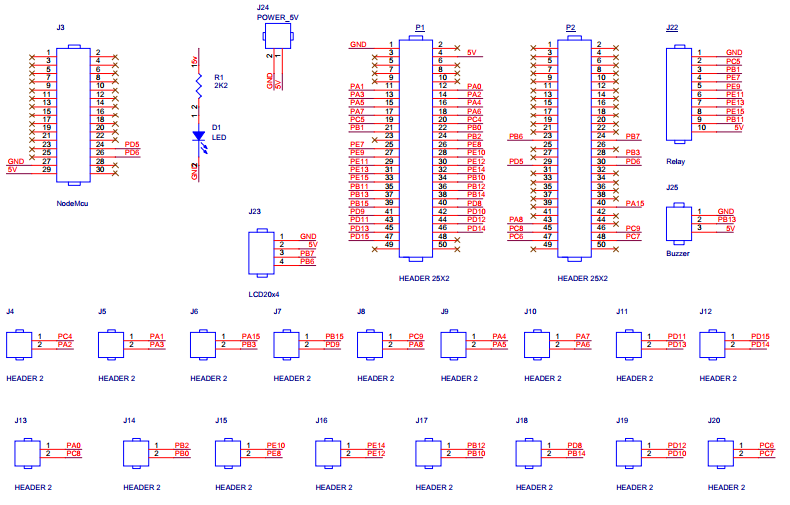
## Phân tích

* Chúng ta sẽ có 3 phương án để kết nối các phần cứng lại với nhau: làm mạch in, cắm trên BreadBoard hoặc hàn trên Test Board hàn.
* Sauk hi suy nghĩ về các ưu nhược điểm và đối chiếu với tính chất của đề tài này, em đã chọn phương án là làm mạch in. Vì với phương án này có rất nhiều ưu điểm như sau:

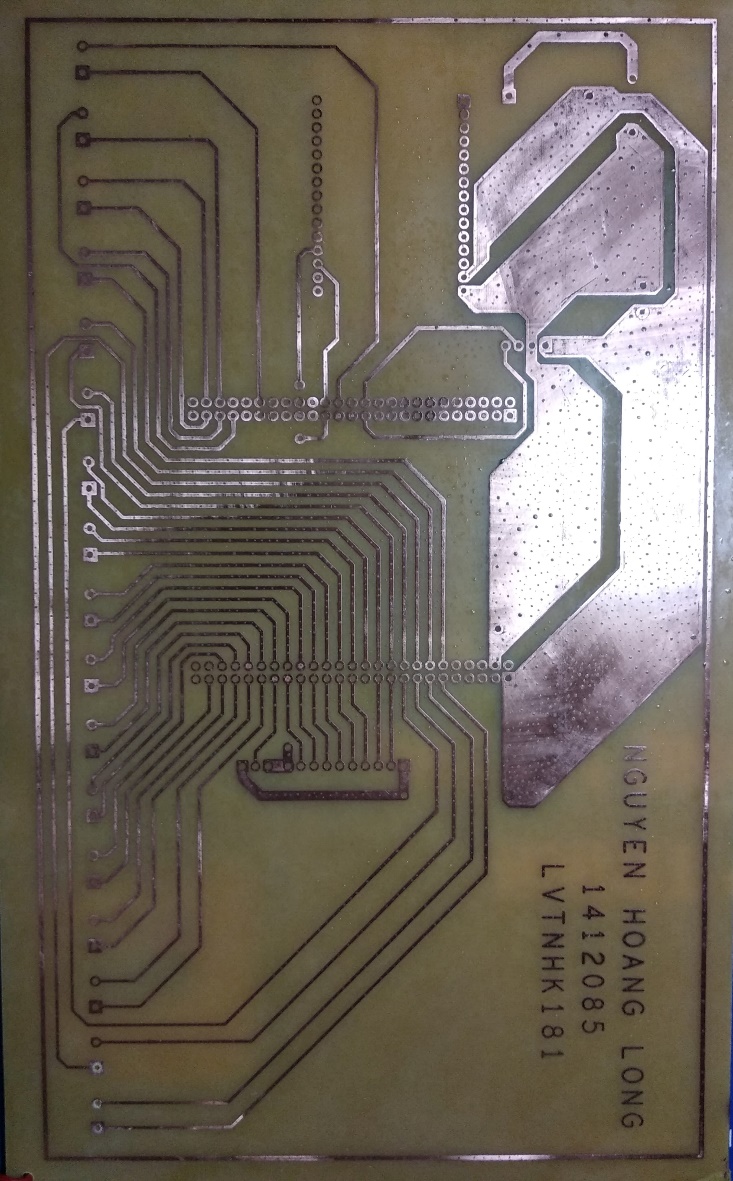
+ Làm cho mạch của mình gọn hơn.

+ Tránh được các tình trạng đứt dây nối mà mình không hay biết.

+ Về thẩm mỹ cũng đẹp hơn.

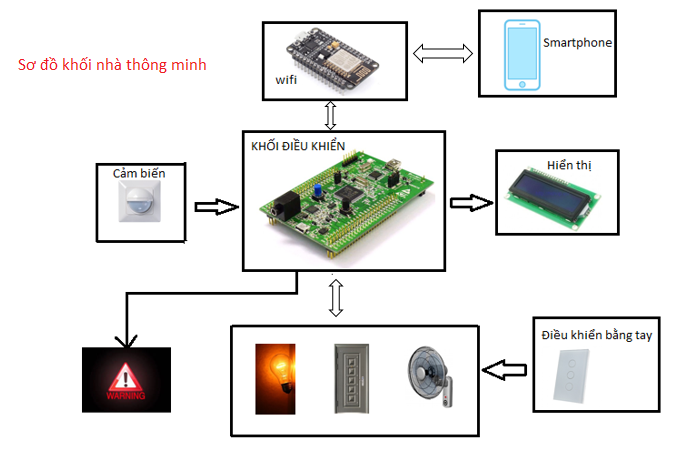


Hình 3-1 Schematic để thực hiện mạch in



Hình 3-2 Mạch in

## Sơ đồ khối tổng quát



Hình 3-3 Sơ đồ khối tổng quát phần cứng

\* Giải thích về sơ đồ khối tổng quát phần cứng:

Phần cứng của đề tài này gồm 7 khối chính:

+ Khối điều khiển: gồm kit STM32F4 Discovery giao tiếp với NodeMCU bằng UART.

+ Khối Smartphone: sử dụng App được tạo từ MIT App Inventor 2, điều khiển cũng như theo dõi thiết bị và cảm biến.

+ Khối thiết bị: gồm đèn, quạt, động cơ,…

+ Khối cảm biến: gồm các loại cảm biến thông dụng hiện nay (nhiệt độ, khói -khí gas, chuyển động,…)

+ Khối hiển thị: gồm 2 LCD (LCD1602 và LCD2004)

+ Khối cảnh báo: gồm còi buzzer và led đỏ.

+ Khối điều khiển bằng tay: gồm nút nhất và cảm ứng 1 chạm điện dung.

## Sơ đồ khối chi tiết

Ta đi vào cụ thể từng khối sau đây:

**Khối điều khiển:**

Khối điều khiển gồm 2 kit điều khiển là STM32F4 Discovery và NodeMCU.

Chức năng của từng kit như sau:

+ STM32F4 Discovery: xử lý và điều khiển tất cả các thiết bị trong nhà như đèn, quạt, cửa,… Đồng thời, xử lý các dữ liệu cảm biến để đưa lên khối Smartphone, cũng như nhận lệnh điều khiển từ Smartphone thì sẽ thực hiện ngay lập tức; cùng lúc đó nếu có sự thay đổi cảm biến so với mức quy định thì điều khiển khối cảnh báo.

+ NodeMCU: có chức năng gửi tất cả các thay đổi lên Firebase khi STM32F4 Discovery đang làm việc trong mạng lưới có kết nối Wifi, cũng như khi có thay đổi từ khối Smartphone xuống Firebase thì NodeMCU cũng gửi những thay đổi đó cho STM32F4 Discovery thực hiện thông qua giao tiếp UART .

**Khối Smartphone:**

Đây là khối truyền lệnh từ người dùng để điều khiển. Người dùng truyền lệnh điều khiển thông qua App được cài đặt trên Smartphone chạy hệ điều hành Android.

Ngoài việc gửi lệnh điều khiển từ người dùng. Khối này còn có chức năng hiển thị một số thông số cho người dùng. Ví dụ thông số nhiệt độ, tình trạng khói – khí gas trong nhà, hay có phát hiện lửa hay không.

**Khối thiết bị:**

Khối này gồm quạt, đèn, động cơ cửa.

Phạm vi mô hình này gồm 5 phòng: phòng khách, 2 phòng ngủ, phòng bếp, phòng vệ sinh; được trang bị đèn và quạt. Ngoài ra còn có động cơ để đóng mở cửa và thẻ RFID góp phần trong việc đóng mở cửa.

**Khối cảm biến:**

Khối cảm biến có nhiệm vụ tiếp nhận dữ liệu thay đổi từ môi trường. Sau đó, dữ liệu này được đưa tới khối điều khiển để xử lý. Sau khi xử lý dữ liệu thì sẽ hiển thị qua khối hiển thị LCD và App để người dùng biết. Đồng thời, nếu quá ngưỡng người dùng quy định thì khối điều khiển sẽ điều khiển khối cảnh báo.

**Khối hiển thị:**

Khối hiển thị gồm 2 LCD: 1 LCD 16x2 và 1 LCD 20x4

LCD 16x2 được đặt tại cửa chính để hiển thị cho người dùng biết đã quẹt đúng thẻ chưa hay đã nhập đúng password chưa.

LCD 20x4 được đặt tại trung tâm nhà để tiện theo dõi tình trạng ngôi nhà với thông số mà khối cảm biến gửi đến.

**Khối cảnh báo:**

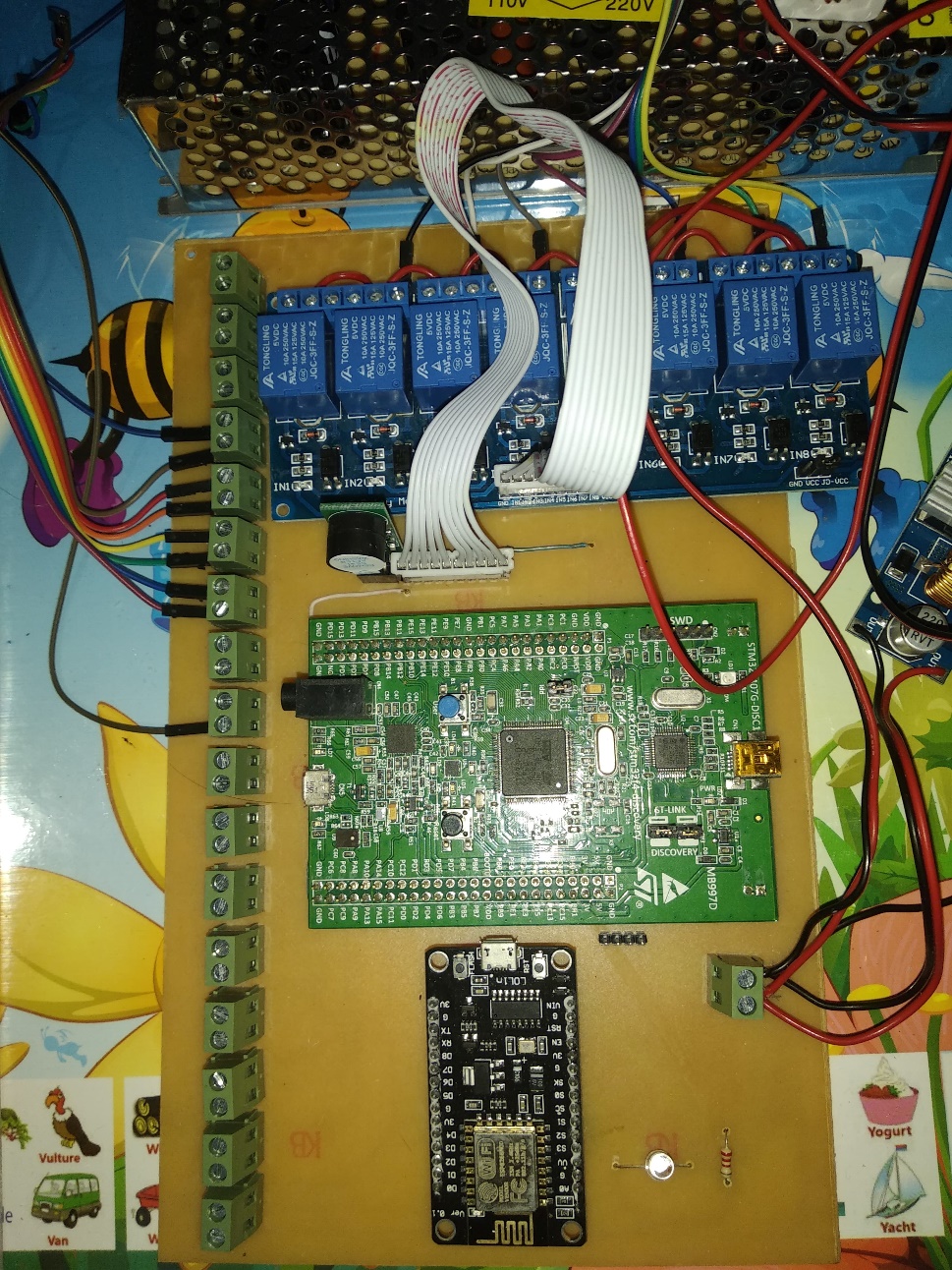
Khối cảnh báo gồm còi buzzer và led đỏ để thông báo đến người dùng những nguy hiểm cũng như giúp người dùng nhanh chóng có biện pháp khắc phục sự cố.

**Khối điều khiển bằng tay:**

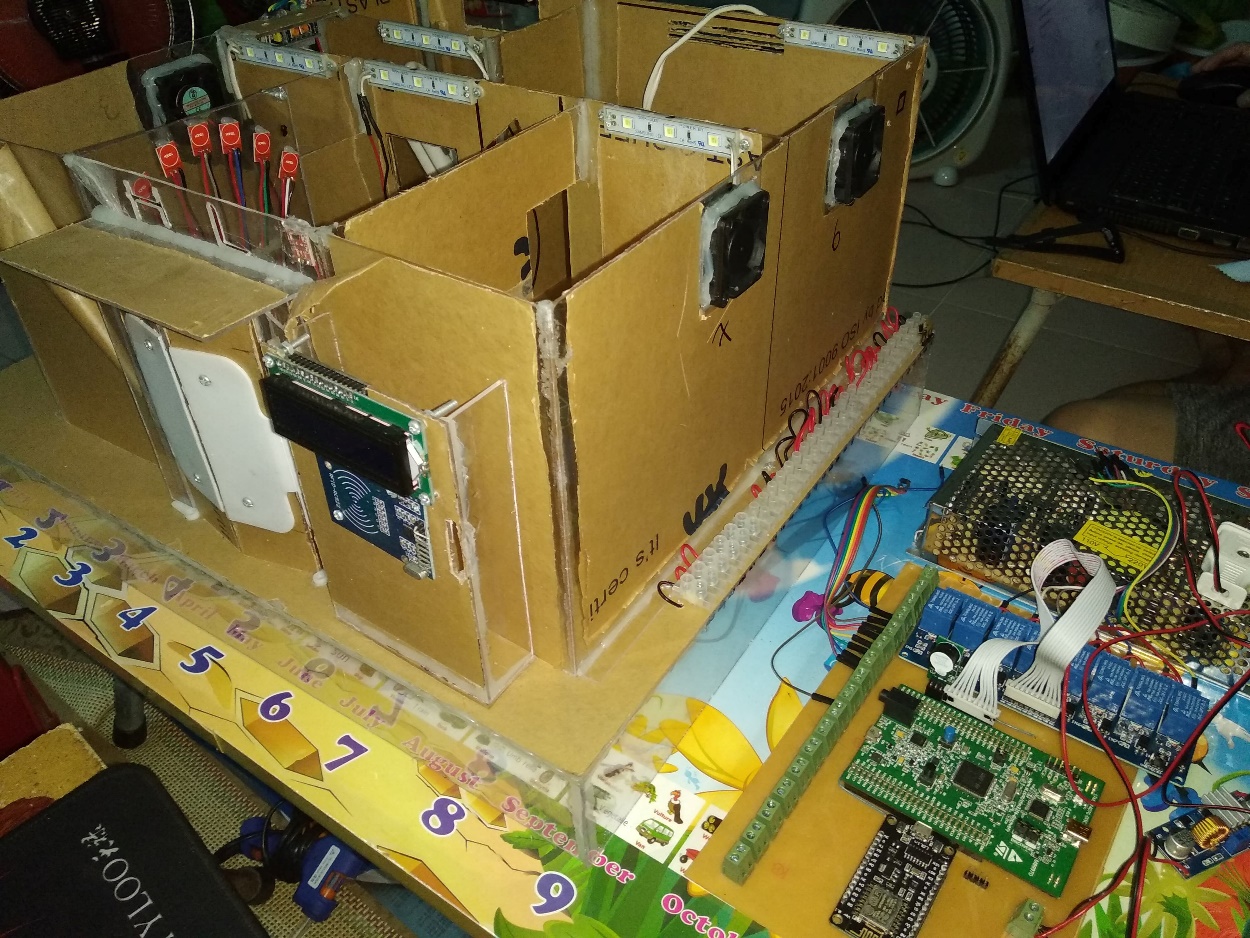
Khối này được trang bị công tắc cảm ứng 1 chạm điện dung TTP223B Mini.

Khối này được thực hiện để giúp người dùng có thể điều khiển thiết bị khi người dùng không dùng khối Smartphone.

Sau đây là tổng quan về mô hình thực tế:



Hình 3-4 Mach in sau khi hàn và gắn linh kiện



Hình 3-5 Tổng thể mô hình thực tế mặt cạnh



Hình 3-6 Tổng thể mô hình thực tế mặt bằng

# THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

## Yêu cầu thiết kế

Trong mục phần mềm này, chúng ta phân ra hai phần:

+ App

+ Code cho vi điều khiển

Chúng ta lần lượt đi vào từng phần:

**Phần App:**

Cài lên Smartphone, vào các màn hình con, ta sẽ có vùng điều khiển từng thiết bị. Ở vùng này, ngoài việc điều khiển thiết bị, ta còn hiển thị trạng thái của từng thiết bị (đang bật hay tắt).

App được thiết kế và lập trình bằng MIT App Inventor 2.

**Phần Code cho vi điều khiển:**

Giao tiếp được với NodeMCU qua UART, từ đó gửi trạng thái hoạt động của các thiết bị và cảm biến lên Firebase, cũng như nhận lại dữ liệu khi Firebase thay đổi rồi vi điều khiển thực hiện theo đúng sự thay đổi đó.

## Phân tích

**Phần mềm để điều khiển:**

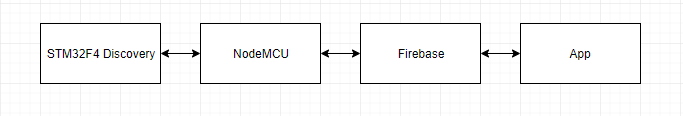
Chúng ta sẽ có ba lựa chọn: App, web và software trên máy tính.

Qua các phân tích ưu điểm và nhược điểm trên thì em chọn dùng App ( MIT App Inventor) vì nó có nhiều ưu điểm phù hợp với khả năng của em hơn so với 2 lựa chọn còn lại như: tiện lợi cho người dùng không có kiến thức sâu về lập trình, tốc độ thực thi nhanh, phần mềm free, có thể sử dụng ở bất cứ đâu khi có kết nối internet, dễ dàng thiết kế và lập trình.

**Phần mềm để lưu trữ dữ liệu:**

Chúng ta có rất nhiều sự lựa chọn. Nhưng qua quá trình tìm hiểu và phân tích ưu nhược điểm thì em quyết định chọn Firebase Google vì nó có nhiều ưu điểm phù hợp với khả năng của em như: tiện lợi cho người dùng, phần mềm free, lưu trữ lại trạng thái cũ khi ngắt kết nối wifi…

## Sơ đồ khối tổng quát



Hình 4-1 Sơ đồ khối tổng quát phần mềm

Trong sơ đồ này, chúng ta có 2 phần quan trọng là App và Firebase. Chúng sẽ giao tiếp với STM32F4 Discovery thông qua mạng wifi ( tức là qua NodeMCU)

## Sơ đồ khối chi tiết

Trong sơ đồ khối tổng quát ở trên, chúng ta sẽ giao tiếp giữa phần mềm và phần cứng thông qua NodeMCU, Firebase và App.



Hình 4-2 Lưu đồ giải thuật giao tiếp giữa STM32F4 Discovery và NodeMCU



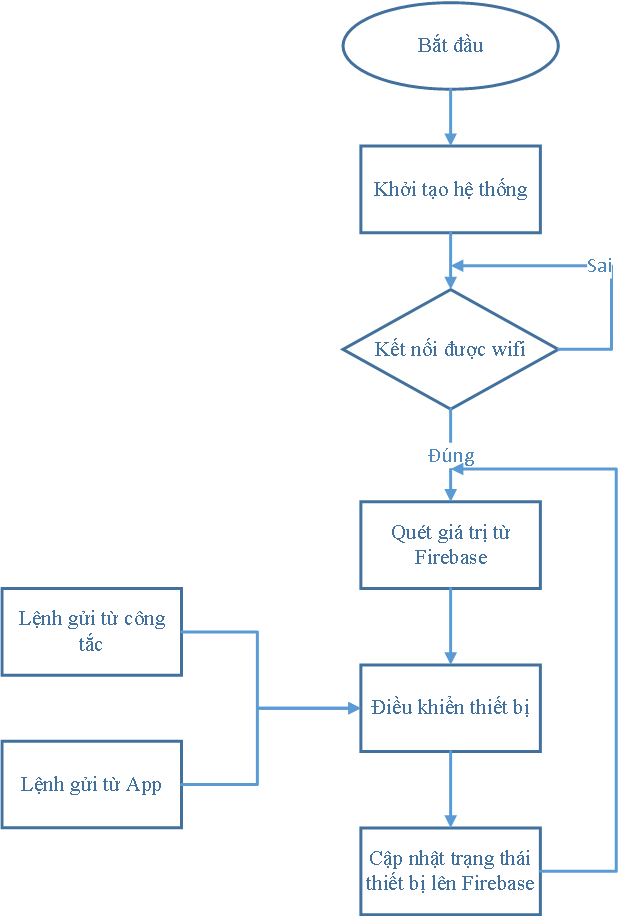
Hình 4-3 Lưu đồ giải thuật báo động khí gas



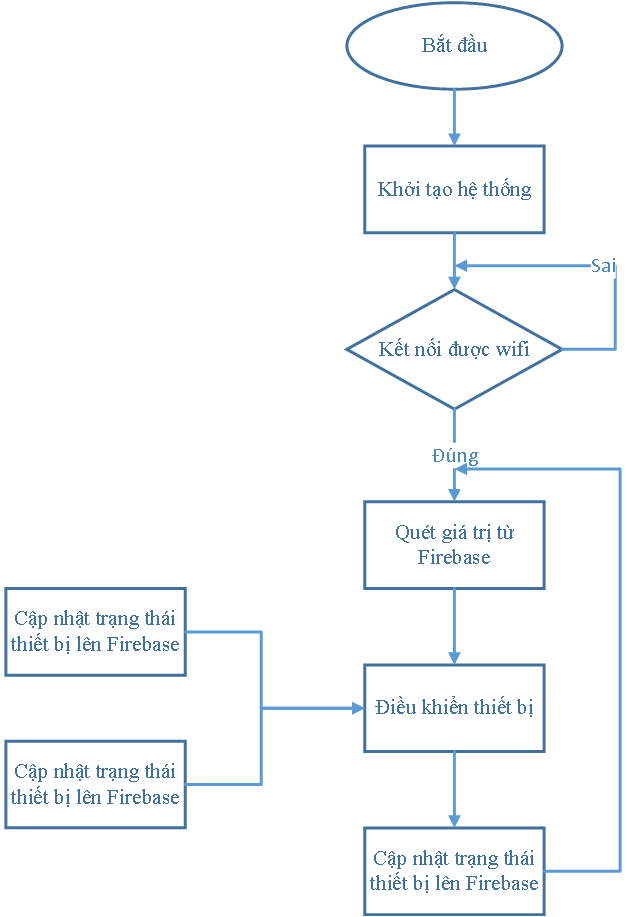
Hình 4-4 Lưu đồ giải thuật chống trộm



Hình 4-5 Lưu đồ giải thuật đóng mở cửa bằng RFID



Hình 4-6 Lưu đồ giải thuật điều khiển thiết bị



Hình 4-7 Lưu đồ giải thuật phần cảm biến

# KẾT QUẢ THỰC HIỆN

## Kết quả phần mềm

App trên smartphone chạy ổn định trên hệ điều hành Android. Dung lượng file .apk khá nhé, có thể tương thích với nhiều loại thiết bị Android.

Các thao tác trên app mượt mà, ổn định. Ngoài ra, cách bố cục các chức năng trên app dễ nhìn, dễ hiểu.

Để app có thể kết nối với hệ thống phần cứng, chúng ta phải kết nối thiết bị android và phần cứng vào chung một mạng wifi. Mạng wifi này có thể là từ Modem wifi phát ra hoặc từ Hotspot trên di động phát ra. Trong luận văn này, em dùng Hotspot điện thoại di động để phát sóng wifi.

Ngoài phần app, ta còn có phần Firebase Google. Phần Firebase cũng chạy ổn định, thời gian thực thi rất nhanh, dường như là đáp ứng thời gian thực.

Firebase là phần trung gian giữa hệ thống phần cứng và phần app, nó thực hiện chức năng của nó một cách rất hiệu quả. Tuy nhiên, trong giới hạn luận văn này, người dùng không cần tương tác với Firebase

## Kết quả phần cứng

Hệ thống phần cứng chúng ta gồm 2 phần chính là phần điều khiển và phần cảm biến, thì cả 2 phần này đều chạy thành công và ổn định

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Mô hình sau khi thiết kế đã đạt được những kết quả sau:

* Thiết kế hoàn chỉnh mô hình nhà thông minh. Gồm đầy đủ các chức năng chính:

+ Điều khiển thiết bị (đèn, quạt, cửa,…) qua app android được cài đặt lên smartphone có thể điều khiển từ xa ở bất kỳ đâu có internet bằng smartphone. Ngoài ra còn có chế độ điều khiển theo ngữ cảnh và dựa vào hẹn giờ của người dùng.

+ Đọc giá trị các cảm biến và hiển thị lên app. Và có thể cảnh báo cho người dùng khi có sự cố xảy ra (như báo cháy, báo khói – khí gas).

* Hệ thống hoạt động khá ổn định, rất ít xảy ra lỗi khi hoạt động. Sử dụng internet là wifi rất phổ biến hiện nay nên đề tài này rất thiết thực và gần gũi với người dùng.
* Việc nghiên cứu đề tài “Nhà Thông Minh” đã mang lại cho em rất nhiều kiến thức và kinh nghiệm. Đề tài giúp cho em nắm rõ được họ ESP8266 nói chung và NodeMCU nói riêng. Nắm rõ cách lập trình Arduino cho NodeMCU trên môi trường lập trình Arduino IDE. Hiểu sâu hơn về các cảm biến. Ngoài ra, làm chủ được cách thức đưa dữ liệu lên internet cũng như lấy dữ liệu trở lại cho vi điều khiển với Cloud Firebase Google. Đồng thời, nắm được cơ bản việc xây dựng một app cơ bản cho điện thoại để ứng dụng vào IoT. Và cuối cùng là kết nối các kiến thức trên với nhau để tạo thành một sản phẩm hữu ích cho đời sống, có mục đích và ý nghĩa thiết thực cho con người.
* Mặc dù hệ thống đã đạt được những ưu điểm nhất định, tuy nhiên do hạn chế về thời gian, tài chính và sự hiểu biết nên sản phẩm vẫn có những mặt hạn chế sau:

+ Tốc độ của hệ thống chưa được tối ưu một cách hoàn hảo.

+ Vẫn chưa làm được các chức năng cao cấp hơn của một ngôi nhà thông minh hiện nay.

+ Ứng dụng android vẫn còn hạn chế, chưa thực sự chuyên nghiệp và thẩm mỹ như các sản phẩm trên thị trường hiện nay.

## Hướng phát triển

Hiện tại đề tài đang dừng ở mức mô hình, nhưng hoàn toàn có thể ứng dụng vào thực tiễn, với chi phí phải chăng, dễ thi công.

App Android có thể được nâng cấp hơn nữa bằng việc lập trình android với các môi trường chuyên nghiệp hơn như Android Studio, Eclipse,…

Có thể phát triển thêm nhiều chức năng cho nhà thông minh, đồng thời tối ưu cả phần cứng và phần mềm hơn nữa để sản phẩm thêm hoàn hảo.

# TÀI LIỆU THAM KHẢO

[1] Giang Di Linh.”Nhà thông minh – xu hướng mới thời công nghệ.”[Online]. <https://news.zing.vn/nha-thong-minh-xu-huong-moi-thoi-cong-nghe-post881944.html> ,05/10/2018.

[2] Tạp chí Nhịp Cầu Đầu Tư.”Nhà thông minh nở rộ: Cơ hội tỷ đô.”[Online]. <https://www.bkav.com.vn/bao-chi-quoc-te-viet-ve-bkav/-/chi_tiet/435260/nha-thong-minh-no-ro-co-hoi-ty-do> ,02/03/2017.

[3] Bảo sgreentech.”Thị trường công nghệ nhà thông minh tại Việt Nam hiện nay.”[Online]. <https://tinhte.vn/threads/thi-truong-cong-nghe-nha-thong-minh-tai-viet-nam-hien-nay.2676688/> ,8/2/2017

[4] Đỗ Minh Phương.”Xử lý ảnh video thời gian thực trên kit STM32.”thạc sỹ,trường đại học Công Nghệ,Hà Nội,2016.

[5] HOCARM.”ESP8266 cho người không biết gì.”[Online]. <https://hocarm.org/esp8266-cho-nguoi-khong-biet-gi/> ,15/10/2016.

[6] vietdung126.”Giới thiệu về ngôn ngữ Drag and Drop của MIT App Inventor.”[Online]. <https://tinhte.vn/threads/gioi-thieu-ve-ngon-ngu-drag-and-drop-cua-mit-app-inventor.2714263/> ,30/7/2017.

# PHỤ LỤC