**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN**

**BỘ MÔN THIẾT BỊ ĐIỆN – ĐIỆN TỬ**

****

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**Đề tài: Thiết kế thiết bị đóng cắt 4 kênh sử dụng WiFi**

**Sinh viên: Nguyễn Tuấn Anh**

**Email: anh.nt173616@sis.hust.edu.vn**

**MSSV: 20173616**

**Học phần: Đồ Án Tốt Nghiệp Cử Nhân**

**Mã học phần: EE**

**Mã lớp:**

**Giảng viên hướng dẫn: TS. Nguyễn Văn Ánh**

**HÀ NỘI, 01/2021**

---------------------------------------------------------------------------------------------------

Mục lục

[Phiếu giao nhiệm vụ đồ án tốt nghiệp 3](#_Toc60748738)

[Lời cảm ơn 3](#_Toc60748739)

[Tóm tắt nội dung báo cáo 4](#_Toc60748740)

[Chương 1. Giới thiệu đề tài 4](#_Toc60748741)

[1.1. Đặt vấn đề 4](#_Toc60748742)

[1.2. Giới thiệu chung về thiết bị điện thông minh 5](#_Toc60748743)

[1.3. Một số thiết bị điện thông minh trong thực tế 6](#_Toc60748744)

[1.4. Mô tả tổng quan hệ thống 7](#_Toc60748745)

[1.5. Mục tiêu và nhiệm vụ của đề tài 7](#_Toc60748746)

[1.6. Giải pháp 8](#_Toc60748747)

[1.6.1. Giải pháp phần cứng 8](#_Toc60748748)

[1.6.2. Giải pháp phần mềm 8](#_Toc60748749)

[1.6.3. Công cụ lập trình và môi trường phát triển 8](#_Toc60748750)

[Chương 2. Thiết kế phần cứng 9](#_Toc60748751)

[2.1. Giới thiệu các linh kiện sử dụng 9](#_Toc60748752)

[2.1.1. Vi điều khiển ESP8266 9](#_Toc60748753)

[2.1.2. Triac 10](#_Toc60748754)

[2.1.3. Triac driver 11](#_Toc60748755)

[2.1.4. Module chuyền đổi nguồn AC/DC 11](#_Toc60748756)

[2.1.5. Các linh kiện thụ động khác 12](#_Toc60748757)

[2.2. Sơ đồ phần cứng 14](#_Toc60748758)

[2.3. Thiết kế mạch 15](#_Toc60748759)

[2.3.1. Thiết kế khối đóng cắt 15](#_Toc60748760)

[2.3.2. Thiết kế khối đọc tín hiệu từ người dùng 15](#_Toc60748761)

[2.3.3. Thiết kế khối điều khiển 15](#_Toc60748762)

[2.3.4. Sơ đồ nguyên lý 15](#_Toc60748763)

[2.3.5. Mạch in 16](#_Toc60748764)

[Chương 3. Phát triển firmware 19](#_Toc60748765)

[3.1. Giới thiệu một số giao thức truyền thông sử dụng trong thiết bị điện thông minh 19](#_Toc60748766)

[3.2. Giới thiệu giao thức truyền thông MQTT 19](#_Toc60748767)

[3.2.1. Mô hình của giao thức MQTT 19](#_Toc60748768)

[3.2.2. Tại sao lại lựa chọn giao thức MQTT? 21](#_Toc60748769)

[3.3. Lưu đồ thuật toán cho Vi điều khiển 21](#_Toc60748770)

[3.4. Chương trình Vi điều khiển 21](#_Toc60748771)

[Chương 4. Phát triển Software 21](#_Toc60748772)

[4.1. Lưu đồ thuật toán cho Software 21](#_Toc60748773)

[4.2. Chương trình Software 21](#_Toc60748774)

[Chương 4. Mô hình và kết quả đạt được 22](#_Toc60748775)

[Kết luận 22](#_Toc60748776)

[Tài liệu tham khảo 22](#_Toc60748777)

**Danh mục bảng biểu**

**Danh mục hình vẽ**

**Danh mục các từ viết tắt**

# Phiếu giao nhiệm vụ đồ án tốt nghiệp

# Lời cảm ơn

Để hoàn thành bản báo cáo Thực tập kĩ thuật, em xin tỏ lòng biết ơn sâu sắc đến thầy TS. Nguyễn Văn Ánh đã tận tình hướng dẫn trong suốt quá trình tìm hiều, nghiên cứu và làm Đồ án tốt nghiệp cử nhân.

Em xin chân thành cảm ơn các thầy cô trong Viện Điện, Trường Đại học Bách Khoa Hà Nội đã tận tình truyền đạt những kiến thức để em có thể tự tin hoàn thành bản Đồ án này.

Cuối cùng, em xin kính chúc quý thầy, cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

Em xin chân thành cảm ơn!

# Tóm tắt nội dung báo cáo

Chương 1. Tổng quan về thiết bị điện thông minh

Chương 2. Cấu tạo phần cứng

Chương 3. Thiết kế firmware

Chương 4. Thiết kế software

Chương 4. Mô hình và kết quả thu được

# Chương 1. Giới thiệu đề tài

## Đặt vấn đề

Đi cùng với cuộc cách mạng công nghệ 4.0 là sự bùng nổ về số lượng của các thiết bị IoT (Internet vạn vật) cung cấp khả năng kết nối với Internet cho các thiết bị điện, các phương tiện,… Năm 2013, tổ chức Global Standards Intiative on Internet of Things định nghĩa IoT là “hạ tầng cơ sở toàn cầu phục vụ cho xã hội thông tin, hỗ trợ các dịch vụ (điện toán) chuyên sâu thông qua các vật thể (cả thực lẫn ảo) được kết nối với nhau nhờ công nghệ thông tin và truyền thông hiện hữu thích hợp,” và với mục đích ấy một “vật” là “một thứ trong thế giới thực (vật thực) hoặc thế giới thông tin “vật ảo”, mà vật đó có thể được nhận dạng và được tích hợp vào một mạng lưới truyền thông”. Hệ thống IoT cho phép các thiết bị độc lập trước đây được kết nối, điều khiển từ xa thông qua hạ tầng mạng Internet hiện hữu, tạo cơ hội nâng cao hiệu năng, độ tin cậy và lợi ích kinh tế được tăng cường bên cạnh việc giảm sự can dự của con người. Khi IoT được tích hợp thêm các thành phần cảm biến và các cơ cấu chấp hành, công nghệ dần này trở thành một hệ thống hoàn thiện như lưới điện thông minh, nhà máy điện ảo, nhà thông minh, vận tải thông minh và thành phố thông minh. Để đưa các hệ thống này thành hiện thực, bên cạnh cơ sở hạ tầng về truyền thông thì không thể thiếu được mạng lưới các thiết bị điện thông minh, các thiết bị nhúng với khả năng tính toán và kết nối với các thiết bị khác thông qua các giao thức truyền thông nói chung và Internet nói riêng.

Nhận thấy sự phát triển của các thiết bị điện thông minh ở Việt Nam còn nhiều hạn chế, các sản phẩm hiện này thường được nghiên cứu và phát triển bởi các công ty ngoài nước, tôi quyết định thiết kế Thiết bị đóng cắt 4 kênh sử dụng WiFi

## Giới thiệu chung về thiết bị điện thông minh

Vị trí của thiết bị điện thông minh ngày cảng trở nên quan trọng trong đời sống thường ngày cũng như trong các lĩnh vực sản xuất nông nghiệp, công nghiệp và dịch vụ. Các thiết bị này đem lại sự tiện lợi khi sử dụng, góp phần giúp con người có được sự tiện thoải mái, tiện nghi, tiết kiệm điện và công sức trong việc điều khiển các thiết bị điện trong nhà.

Các thiết bị điện thông minh trong nhà là các thiết bị điện mà bạn có thể điều khiển chúng thông qua những thao tác mạng internet hoặc kết nối không dây của các thiết bị bạn đang sử dụng như Laptop, Smartphone. Thậm chí các loại thiết bị điện thông minh có khả năng cảm ứng nhiệt, ánh sáng giúp cho các thao tác đóng cắt được tự động, không cần đến sự can thiệp của người dùng.

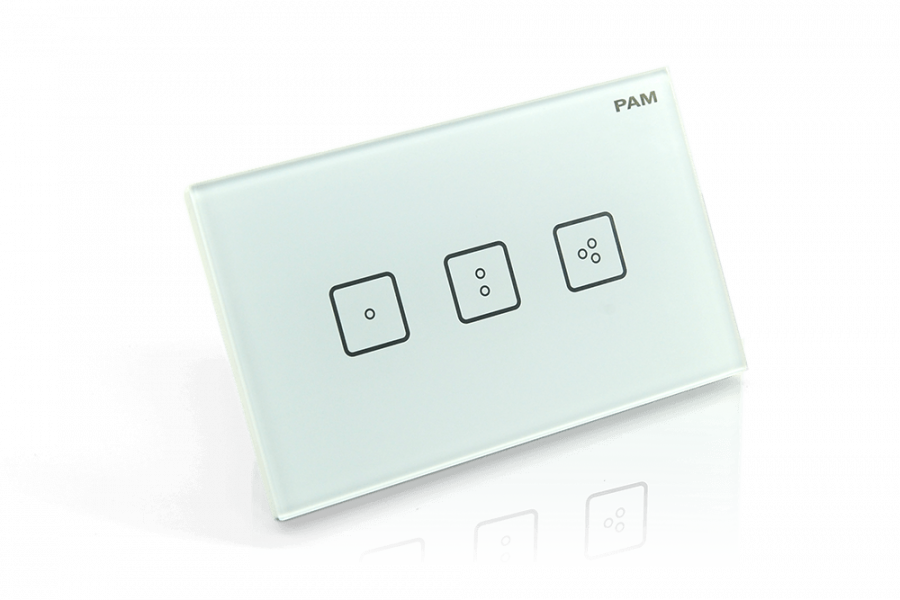
Thiết bị điện thông minh là hệ thống cho phép kết nối các thiết bị sử dụng điện trong nhà với nhau. Thiết bị mang đến nhiều cách để điều khiển, có khả năng tự động xử lý và thông báo cho người sử dụng. Ngoài ra, có khả năng tương tác được với các thông số môi trường, giúp người sử dụng có thể giám sát và điều khiển các thiết bị từ xa, đem lại sự an toàn, tiện nghi, linh hoạt, tiết kiệm.

Xu thế thiết bị điện thông minh là phát triển công nghệ vi mạch ngày càng nhỏ, công nghệ nano, tốc độ xử lý ngày càng nhanh, với chip đa nhân, khả năng kết hợp trao đổi dữ liệu với các thiết bị với nhau, các chuẩn giao tiếp dùng dây dẫn và đặc biệt là giao tiếp không dây như hồng ngoại, RF, WiFi, zigbee, …

## Một số thiết bị điện thông minh trong thực tế

Các thiết bị điện thông minh phát triển với tốc độ rất nhanh trong thời gian gần đây, mang lại nhiều tiện ích và cuộc sông tiện nghi cho người sử dụng. Một số thiết bị thông minh tiêu biểu có thể kể đến như:

* Công tắc thông minh cho phép điều khiển tại chô thông qua mặt cảm ứng và điều khiển từ xa thông qua ứng dụng trên thiết bị di động

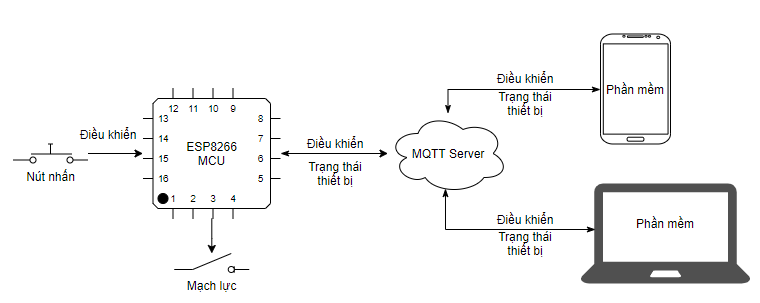


* Ổ cắm WiFi cho phép đóng cắt cung cấp điện ở ổ cắm thông qua ứng dụng điều khiển trên điện thoại thông minh và laptop kết nối Internet



## Mô tả tổng quan hệ thống

Hệ thống điều khiển đóng cắt 4 kênh qua WiFi gồm 3 thành phần chính: Thiết bị đóng cắt và điều khiển tại hiện trường, hệ thống server truyền dẫn dữ liệu, phần mềm điều khiển và giám sát từ xa.



Phần thiết bị đóng cắt và điều khiển tại trường có chức năng đóng cắt trực tiếp các thiết bị điện thông qua Triac và các nút nhấn. Việc trao đổi dữ liệu giữa thiết bị đóng cắt, phần mềm điều khiển và server được thực hiện thông qua giao thức IoT thông dụng hiện nay là MQTT. Phía server MQTT có nhiệm vụ nhận và truyển các bản tin giữa thiết bị điều khiển và thiết bị trường.

## Mục tiêu và nhiệm vụ của đề tài

Đồ án tập trung vào phát triển thiết bị đóng cắt 4 kênh sử dụng WiFi, bao gồm:

* Phát triển phần cứng thiết bị:
  + Thiết kế mạch lực
  + Thiết kế mạch nút nhấn giao tiếp người – máy
  + Thiết kế khối điều khiển
* Phát triển Firmware cho thiết bị:
  + Phát triển chức năng điều khiển mạch lực
  + Phát triển chức năng giao tiếp người – máy
  + Phát triển trao đổi dữ liệu thiết bị và Server qua giao thức MQTT
* Phát triển Software cho thiết bị trên điện thoại và máy tính:
  + Phát triển ứng dụng cho phép hiển thị trạng thái và điều khiển thiết bị
  + Phát triển ứng dụng trao đổi dữ liệu thiết bị và Server qua giao thức MQTT

## Giải pháp

### Giải pháp phần cứng

Thiết bị được xây dựng xung quanh kit NodeMCU - vi điều khiển ESP8266 ở mạch điều khiển và phần tử bán dẫn Triac ở mạch lực.

### Giải pháp phần mềm

Hệ thống được xây dựng bao gồm phần điều khiển đóng cắt điện áp xoay chiều tại chỗ thông qua nút nhấn và điều khiển qua ứng dụng trên máy tính và điện thoại thông minh. Tôi quyết định sử dụng giao thức MQTT để truyền thông, phát triển firmware dựa trên thư viện ESP8266 cho Arduino và phát triển ứng dụng dựa trên thư viện Kivy cho ngôn ngữ Python.

Giao thức MQTT là giao thức phổ biến trong thời đại IoT ngày nay. Đây là một giao thức gọn nhẹ, được thiết kế để kết nối các thiết bị mà có mạng băng thông thấp rất phù hợp với hệ thống điều khiển đóng cắt này.

Kivy là một thư viện mã nguồn mở được viết bằng Python để phát triển ứng dụng, đặc biệt là giao diện người dùng. Ứng dụng viết bằng Kivy có thể chạy trên cả Windows, Linux, OS X, iOS và Android.

### Công cụ lập trình và môi trường phát triển

1. Công cụ để lập trình ESP8266

Trong đồ án sử dụng Arduino IDE làm nền tảng lập trình bởi vì các lý do sau:

* Miễn phí và cộng đồng phát triển mạnh
* Hỗ trợ nhiều thư viện

1. Công cụ để lập trình ứng dụng điều khiển

Trong đồ án sử dụng Python IDE làm nền tảng cho lập trình ứng dụng vì các lý do sau:

* Python có cú pháp đơn giản
* Python miễn phí và cộng đồng phát triển mạnh
* Hệ thống thư viện hỗ trợ lớn
* Hỗ trợ trên nhiều hệ điều hành

1. Công cụ thiết kế phần cứng

Phần cứng của đồ án được thiết kế bằng phần mềm Altium Designer 17. Đây là phần mềm chuyên dụng được dùng trong lĩnh vực thiết kế mạch điện tử.

# Chương 2. Thiết kế phần cứng

Chương này sẽ mô tả chi tiết các thành phần phần cứng của thiết bị và kèm với file nguyên lý. Khối điều khiển của thiết bị được xây dựng dựa trên module ESP8266 nodeMCU, bao gồm các chức năng: đọc tín hiệu nút bấm, điều khiển đóng cắt triac thông qua các triac driver.

## Giới thiệu các linh kiện sử dụng

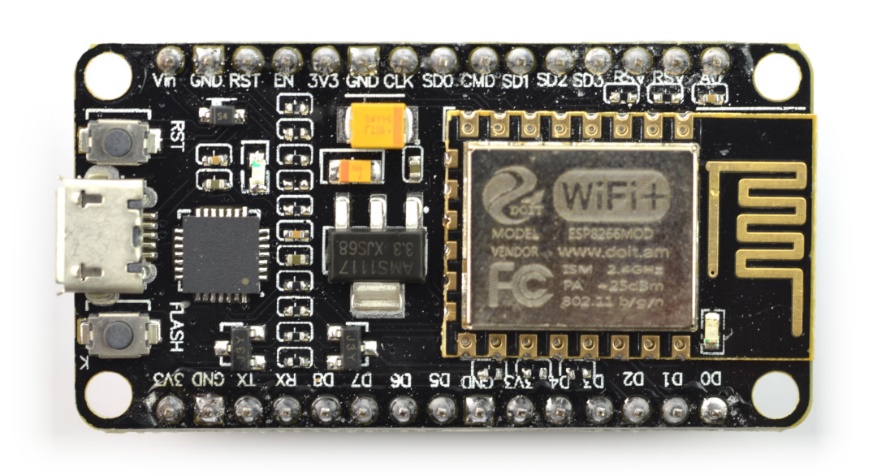
### Vi điều khiển ESP8266

ESP8266 là dòng chip tích hợp Wi-Fi 2.4Ghz có thể lập trình được, rẻ tiền được sản xuất bởi một công ty bán dẫn Trung Quốc: Espressif Systems.

Được phát hành đầu tiên vào tháng 8 năm 2014, đóng gói đưa ra thị trường dạng Mô dun ESP-01, được sản xuất bởi bên thứ 3: AI-Thinker. Có khả năng kết nối Internet qua mạng Wi-Fi một cách nhanh chóng và sử dụng rất ít linh kiện đi kèm. Với giá cả có thể nói là rất rẻ so với tính năng và khả năng ESP8266 có thể làm được.

ESP8266 có một cộng đồng các nhà phát triển trên thế giới rất lớn, cung cấp nhiều Module lập trình mã mở giúp nhiều người có thể tiếp cận và xây dựng ứng dụng rất nhanh.

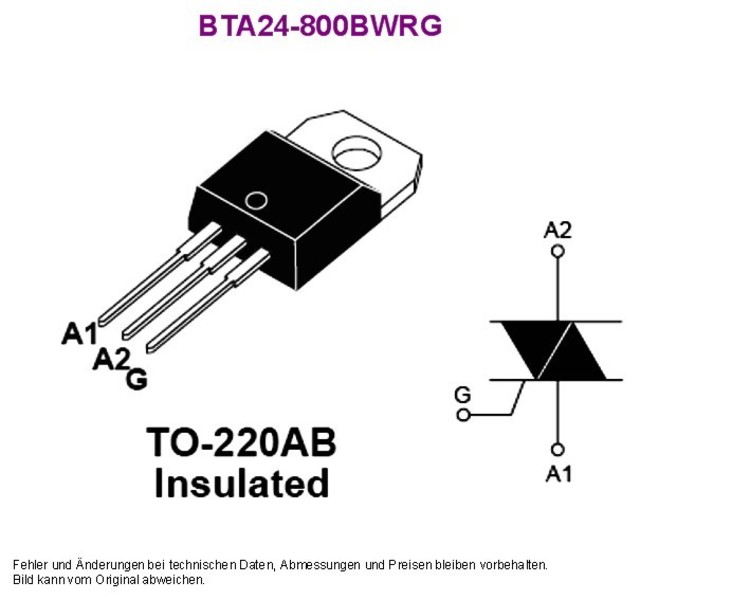
Hiện nay tất cả các dòng chip ESP8266 trên thị trường đều mang nhãn ESP8266EX, là phiên bản nâng cấp của ESP8266. Trong đồ án này sử dụng kit phát triển cho chip ESP8266 là NodeMCU



Thông số phần cứng:

* 32-bit RISC CPU: Tensilica Xtensa LX106 running at 80 MHz
* Hổ trợ Flash ngoài từ 512KiB đến 4MiB
* 64KBytes RAM thực thi lệnh
* 96KBytes RAM dữ liệu
* 64KBytes boot ROM
* Chuẩn wifi EEE 802.11 b/g/n, Wi-Fi 2.4 GHz Tích hợp TR switch, balun, LNA, khuếch đại công suất và matching network Hổ trợ WEP, WPA/WPA2, Open network
* Tích hợp giao thức TCP/IP
* Hổ trợ nhiều loại anten
* 16 chân GPIO
* Hổ trợ SDIO 2.0, UART, SPI, I²C, PWM, I²S với DMA
* 1 ADC 10-bit
* Dải nhiệt độ hoạt động rộng: -40oC ~ 125oC

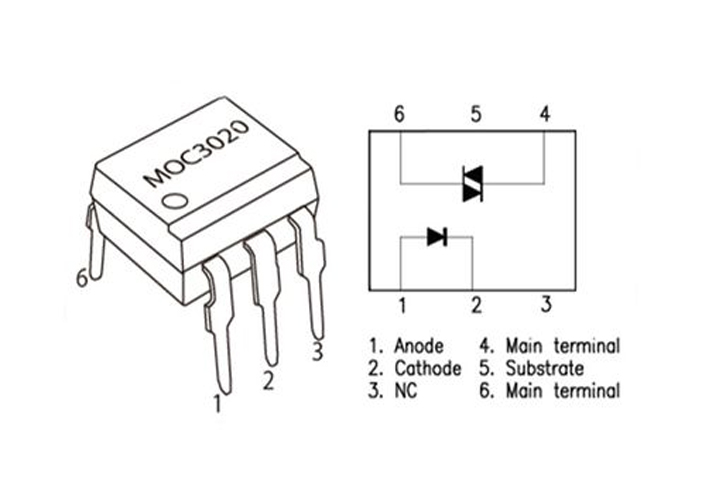
### Triac



Triac là phần tử bán dẫn có cấu trúc bán dẫn gồm năm lớp, tạo nên cấu trúc p-n-p-n như ở Tiristo theo cả hai chiều giữa cực A1 và A2. Về nguyên tắc, triac hoàn toàn có thể coi tương đương với 2 tiristo đấu song song ngược. Triac có thể điều khiển mở dẫn dòng bằng cả xung dòng dương (dòng đi vào cực điều khiển) hoặc dung dòng âm (dòng đi ra khỏi cực điều khiển). Tuy nhiên xung dòng điều khiển xung dòng điều khiển âm có độ nhạy kém hơn, nghĩa là dòng chỉ có thể chạy qua triac khi điện áp giữa A1 và A2 phải lớn hơn một giá trị nhất định, lớn hơn khi dùng dòng điều khiển dương. Vì vậy trong thực tế để đảm bảo tính đối xứng của dòng điện qua triac thì sử dụng xung điều khiển âm là tốt hơn cả. Triac đặc biệt hữu ích trong các ứng dụng cần điều chỉnh điện áp xoay chiều hoặc các công tắc tơ tĩnh ở dải công suất vừa và nhỏ.

### Triac driver

Triac driver là IC cách ly quang được thiết kế đặc biệt để cung cấp dòng điều khiển cho triac đồng thời cách ly mạch điều khiển với mạch lực. Triac driver thường gồm đầu vào là 1 diode phát quang (LED) và đầu ra gắn với 1 chip điều khiển Triac nhạy sáng. Khi có dòng điện bên phía điều khiển LED sẽ phát sáng và chip điều khiển Triac sẽ cho phép dòng điện chạy qua driver. Dòng điện này sẽ chảy vào cực gate của Triac lực và từ đó cho phép đóng cắt tải.



### Module chuyền đổi nguồn AC/DC

Module nguồn AC-DC Hi-Link HLK-PM01 5VDC 3W có thiết kế nhỏ gọn với vỏ bọc nhựa an toàn, chuyên nghiệp, được sử dụng để chuyển nguồn xoay chiều AC sang 5VDC công suất tối đa 3W cấp cho thiết bị, module được sản xuất bởi hãng Hi-Link chuyên về các module nguồn được sử dụng trong công nghiệp với độ bền, chống nhiễu tốt và độ an toàn cao.



Thông số kĩ thuật:

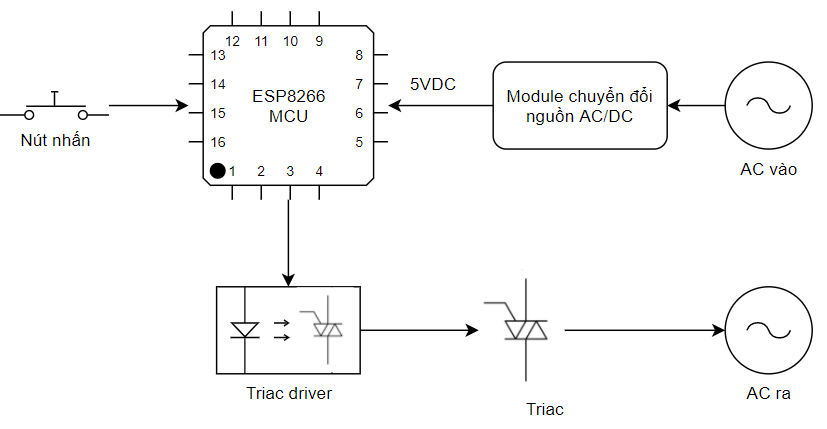
* Điện áp ngõ vào: 100~240 VAC/50~60 Hz
* Điện áp ngõ ra: 5 VDC
* Công suất trung bình: 3W
* Răng cưa và nhiễu nhỏ
* Bảo vệ quá tải và ngắn mạch
* Hiệu suất 72%

### Các linh kiện thụ động khác

* Điện trở: Điện trở là một linh kiện điện tử thụ động gồm 2 tiếp điểm kết nối, thường được dùng để hạn chế cường độ dòng điện chảy trong mạch, điều chỉnh mức độ tín hiệu, dùng để chia điện áp, kích hoạt các linh kiện điện tử chủ động như transistor, tiếp điểm cuối trong đường truyền điện và có trong rất nhiều ứng dụng khác. Điện trở công suất có thể tiêu tán một lượng lớn điện năng chuyển sang nhiệt năng có trong các bộ điều khiển động cơ, trong các hệ thống phân phối điện. Các điện trở thường có trở kháng cố định, ít bị thay đổi bởi nhiệt độ và điện áp hoạt động. Biến trở là loại điện trở có thể thay đổi được trở kháng như các núm vặn điều chỉnh âm lượng. Các loại cảm biến có điện trở biến thiên như: cảm biến nhiệt độ, ánh sáng, độ ẩm, lực tác động và các phản ứng hóa học. Điện trở là loại linh kiện phổ biến trong mạng lưới điện, các mạch điện tử, Điện trở thực tế có thể được cấu tạo từ nhiều thành phần riêng rẽ và có nhiều hình dạng khác nhau, ngoài ra điện trở còn có thể tích hợp trong các vi mạch IC. Điện trở được phân loại dựa trên khả năng chống chịu, trở kháng, ... tất cả đều được các nhà sản xuất ký hiệu trên nó.
* Tụ điện: Tụ điện là một loại linh kiện điện tử thụ động tạo bởi hai bề mặt dẫn điện được ngăn cách bởi điện môi. Khi có chênh lệch điện thế tại hai bề mặt, tại các bề mặt sẽ xuất hiện điện tích cùng điện lượng nhưng trái dấu. Sự tích tụ của điện tích trên hai bề mặt tạo ra khả năng tích trữ năng lượng điện trường của tụ điện. Khi chênh lệch điện thế trên hai bề mặt là điện thế xoay chiều, sự tích luỹ điện tích bị chậm pha so với điện áp, tạo nên trở kháng của tụ điện trong mạch điện xoay chiều. Về mặt lưu trữ năng lượng, tụ điện có phần giống với ắc qui. Mặc dù cách hoạt động của chúng thì hoàn toàn khác nhau, nhưng chúng đều cùng lưu trữ năng lượng điện. Ắc qui có 2 cực, bên trong xảy ra phản ứng hóa học để tạo ra electron ở cực này và chuyển electron sang cực còn lại. Tụ điện thì đơn giản hơn, nó không thể tạo ra electron - nó chỉ lưu trữ chúng. Tụ điện có khả năng nạp và xả rất nhanh. Đây là một ưu thế của nó so với ắc qui.
* Đèn LED: LED (viết tắt của Light Emitting Diode, có nghĩa là diode phát sáng hoặc diode phát quang) là các diode có khả năng phát ra ánh sáng hay tia hồng ngoại, tử ngoại. Cũng giống như diode, LED được cấu tạo từ một khối bán dẫn loại p ghép với một khối bán dẫn loại n.
* Nút bấm: Nút ấn là một loại công tắc đơn giản điều khiển hoạt động của máy hoặc một số loại quá trình. Hầu hết, các nút nhấn là nhựa hoặc kim loại. Hình dạng của nút ấn có thể phù hợp với ngón tay hoặc bàn tay để sử dụng dễ dàng. Tất cả phụ thuộc vào thiết kế cá nhân. Nút ấn có 2 loại chính là nút nhấn thường mở hoặc nút nhấn thường đóng. Nút nhấn có ba phần: Bộ truyền động, các tiếp điểm cố định và các rãnh. Bộ truyền động sẽ đi qua toàn bộ công tắc và vào một xy lanh mỏng ở phía dưới. Bên trong là một tiếp điểm động và lò xo. Khi nhấn nút, nó chạm vào các tiếp điểm tĩnh làm thay đổi trạng thái của tiếp điểm. Trong một số trường hợp, người dùng cần giữ nút hoặc nhấn liên tục để thiết bị hoạt động. Với các nút nhấn khác, chốt sẽ giữ nút bật cho đến khi người dùng nhấn nút lần nữa. Công tắc nút nhấn sử dụng nhiều trong các ứng dụng khác nhau như máy tính, điện thoại nút nhấn và nhiều thiết bị gia dụng. Bạn có thể nhìn thấy chúng trong nhà, văn phòng và trong các ứng dụng công nghiệp ngày nay. Chúng có thể bật, tắt máy hoặc làm cho thiết bị thực hiện các hoạt động cụ thể, như trường hợp với máy tính. Trong một số trường hợp, các nút nhấn có thể kết nối thông qua liên kết cơ học, điều khiển một nút nhấn khác hoạt động. Đa số, các nút sẽ có màu sắc cụ thể để biểu thị mục đích của chúng. Ví dụ như nút nhất màu xanh thường được sử dụng để bật thiết bị hay nút nhấn màu đỏ để tắt thiết bị. Điều này tránh gây nên một sô nhầm lẫn. Nút dừng khẩn cấp thường là các nút ấn lớn, thường có màu đỏ và có đầu lớn hơn để sử dụng dễ dàng hơn.

## Sơ đồ tổng quan phần cứng

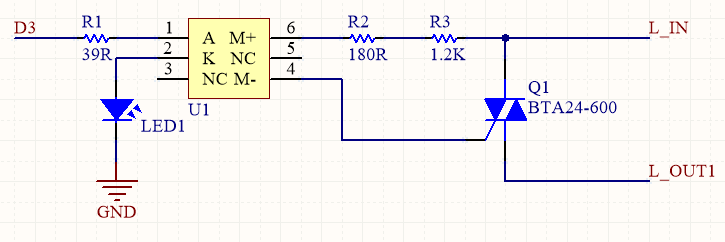
Phần cứng của thiết bị được cấp nguồn bởi Module chuyển đổi nguồn từ 220VAC sang 5VDC. Vi điều khiển ESP8266 hoạt động với nguồn cấp 3.3VDC nhưng trên kit phát triển nodeMCU đã có sẵn IC hạ áp AMS1117 từ 5V xuống 3.3VDC nên chỉ cần một module nguồn duy nhất cung cấp 5VDC cho toàn bộ hệ thống.



## Thiết kế mạch

### Thiết kế khối đóng cắt

Khối đóng cắt 1 kênh gồm 1 triac driver MOC3020 (U1) và 1 triac BTA24

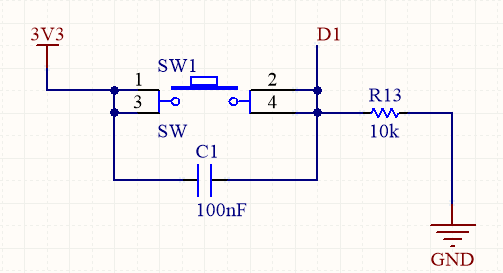


### Thiết kế khối đọc tín hiệu nút bấm

Khối đọc nút nhấn gồm:

* Nút nhấn nhả
* Tụ điện 100nF chống dội nút nhấn
* Điện trở 10K Ohm kéo xuống đất

Nút nhấn được thiết kế hoạt động ở chế độ tích cực mức cao. Khi không nhấn vi điều khiển sẽ đọc chân nút nhấn có mức logic thấp. Khi nút nhấn được nhấn vi điều khiển sẽ đọc chân nút nhấn có mức logic cao và xử lý tương ứng.



### Thiết kế khối điều khiển

### Sơ đồ nguyên lý phần cứng

### Mạch in

# Chương 3. Phát triển firmware

## Giới thiệu một số giao thức truyền thông sử dụng trong thiết bị điện thông minh

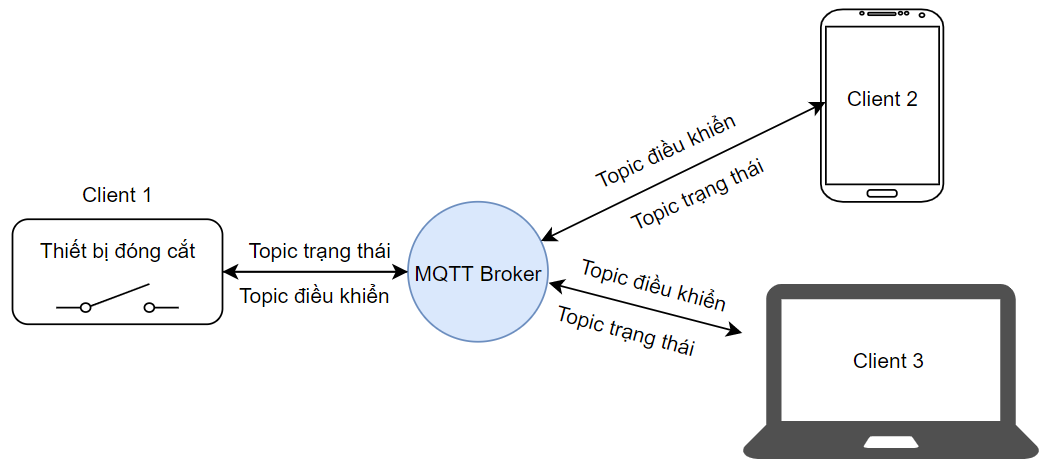
Trong IoT việc giao tiếp giữa các thiết bị với nhau là một trong các yếu tố cơ bản, từ đó dẫn đến yêu cầu phải có một giao thức chung để các thiết bị với phần cứng khác nhau có thể giao tiếp với nhau. Các giao thức truyền dữ liệu phổ biến trong lĩnh vực thiết bị điện thông minh hiện nay là CoAP, MQTT, AMQP, DSS, …

* Giao thức CoAP: Giao thức được thiết kế cho các thiết bị nhúng, về cơ bản giao thức CoAP tương tự như HTTP nhưng các gói tin nhỏ hơn, chủ yếu là giao thức 1-1 để truyền trạng thái thông tin giữa client và server
* Giao thức AMQP: Giao thức AMQP được thiết kế với các gói tin được định hướng, xếp hàng và định tuyến có độ tin cậy và bảo mật cao
* Giao thức DDS: Giao thức truyền tin được thiết kế cho giao tiếp máy-máy, hiệu năng cao, tương thích tốt giữa các phần cứng và phần mềm khác nhau, thời gian thực, có khả năng mở rộng tốt
* Giao thức MQTT: MQTT là giao thức truyền thông theo mô hình Publish/Subscribe.

## Giới thiệu giao thức truyền thông MQTT

MQTT là giao thức truyền thông theo mô hình Publish/Subscribe, sử dụng băng thông thấp, có độ tin cậy cao, có khả năng hoạt động trong môi trường đường truyền không ổn định.

### Mô hình của giao thức MQTT



Mô hình gồm 2 phần chính là là Broker (với vai trò máy chủ) và Client (máy khách)

* Broker đóng vai trò máy chủ đảm nhận vai trò nhận và chuyển các bản tin. Broker có thể cài đặt trên các máy tính nhúng, hoặc sử dụng Broker Cloud
* Client có thể là các thiết bị chấp hành, các cảm biến, cloud nhận dữ liệu, các thiết bị điện thông minh, …
* Client của MQTT gửi và nhận bản tin dựa trên mô hình Topic. Để nhận bản tin của 1 Topic, Client sẽ Subscribe (đăng kí) Topic đó và khi có bản tin được Publish (xuất bản) đến Topic đó Broker sẽ tự động gửi bản tin đến Client.
* Broker nhận tin nhắn từ các Client gửi đến các Topic, chuyển tiếp bản tin hoặc lưu lại. Ngoài ra Broker cũng đảm nhận thêm một số tính năng khác như bảo mật, …
* QoS: Mức độ tin cậy khi gửi bản tin. Có 3 mức QoS: 0,1 và 2

+ QoS 0: Broker/Client gửi bản tin đúng 1 lần

+ QoS 1: Broker/Client gửi bản tin cho đến khi có xác nhận từ bên nhận

+ QoS 2: Broker/Client gửi bản tin và đảm bảo có xác nhận từ cả 2 phía

* Retain: Bản tin có flag Retain sẽ được Broker lưu lại tại Topic và chuyển đến các Client mới Subscribe Topic đó.
* Birth/Death/LWT:

+ Birth là bản tin được gửi tới các Client khi có thiết bị mới kết nối

+ Death là bản tin được gửi khi có thiết bị mất kết nối

+ LWT là bản tin được cài đặt bới Client, sẽ được gửi tới Topic chỉ định khi Client đó mất kết nối

* Mức độ bảo mật: 1 lớp xác thực bằng ID và mật khẩu khi các Client kết nối với Broker. Mức độ bảo mật có thể tăng thêm bằng các giải pháp bảo mật ở tầng mạng

### Tại sao lại lựa chọn giao thức MQTT?

* Ưu điểm:

+ Yêu cầu băng thông thấp

+ Thích hợp với các thiết bị nhúng

+ Chi phí triển khai thấp

+ Dễ nâng cấp hệ thống

+ Các client dễ kết nối lại khi bị mất kết nối

+ Có các tính năng giúp duy trì kết nối và phát hiện khi các node bị mất kết nối

* Nhược điểm:

+ Sử dụng Broker nên khi Broker gặp sụ cố sẽ cả hệ thống sẽ mất kết nối. Để đảm bảo kết nối thông suốt cần có Broker backup

+ Số lượng Client bị giới hạn bởi cấu hình của Broker

+ Cần sử dụng các biện pháp bên ngoài để tăng tính bảo mật cho hệ thống

Có thể thấy đây là một giao thức gọn nhẹ, được thiết kế để kết nối các thiết bị mà có mạng băng thông thấp rất phù hợp với các thiết bị điện thông minh.

### Thông tin server và quy định bản tin MQTT

* Thông tin server MQTT

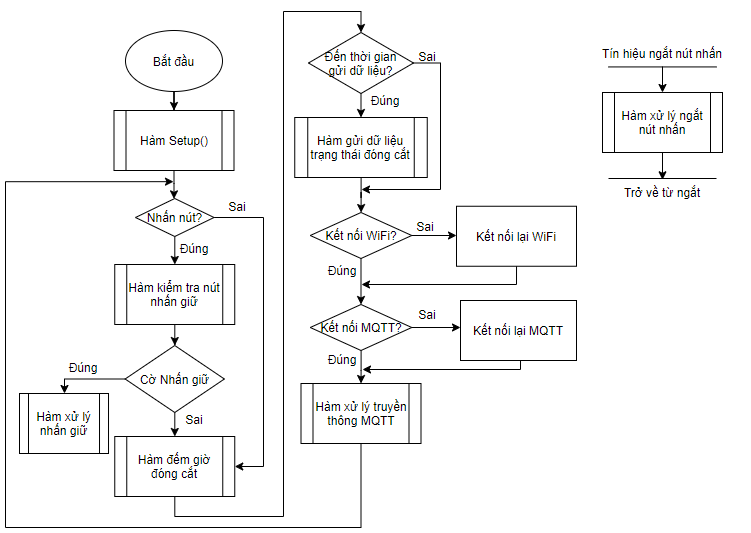
|  |  |
| --- | --- |
| Broker | broker.hivemq.com |
| Port | 1883 |

* Quy định bản tin MQTT

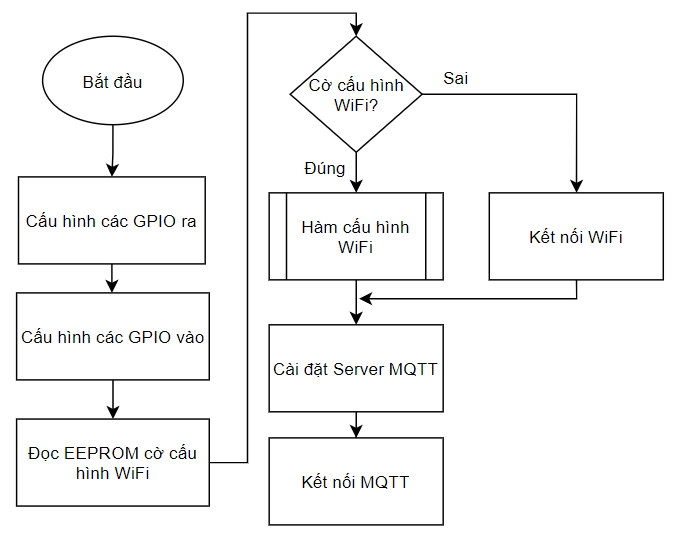
|  |  |  |
| --- | --- | --- |
| Nội dung | Topic | Payload |
| Trạng thái thiết bị đóng cắt | status/datnta | /1/x/2/x/3/x/4/x  - x là trạng thái của từng kênh. x = 1: kênh đang dẫn. x = 0: kênh mở ra  VD: /1/0/2/1/3/1/4/0: kênh 1 và 4 đang mở, kênh 2 và 3 đang dẫn. |
| Lệnh điều khiển thiết bị đóng cắt | cmd/datnta | /kênh đóng cắt/điều khiển/thời gian  - Kênh đóng cắt: 1, 2, 3, 4  - Điều khiển: 0 hoặc 1 tương ứng là mở hoặc dẫn  - Thời gian: thời gian dẫn nếu lệnh điều khiển là 1, thời gian trễ mở kênh nếu lệnh điều khiển là 0. Đơn vị tính bằng giây.  VD:  /1/1/60: điều khiển kênh 1 dẫn trong 60 giây rồi mở ra.  /1/1/00: điều khiển kênh 1 dẫn không giới hạn thời gian. |

## Lưu đồ thuật toán và chương trình cho Vi điều khiển

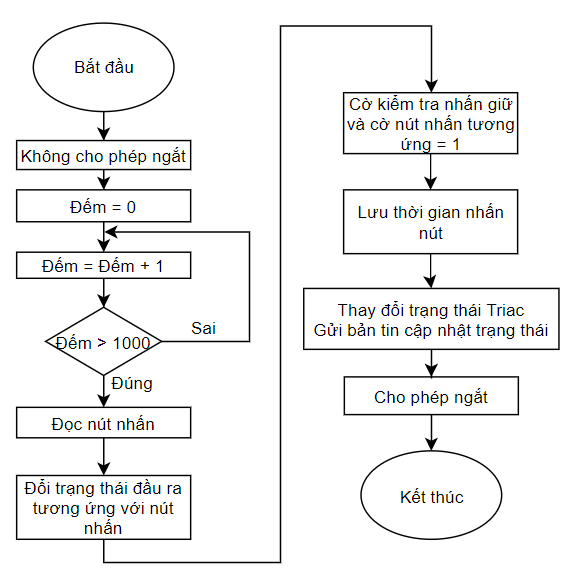
* Lưu đồ thuật toán



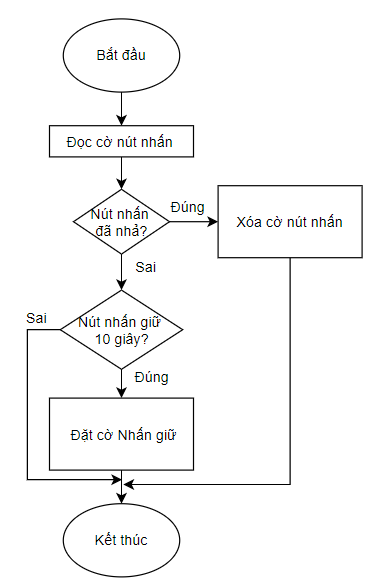
* Hàm setup



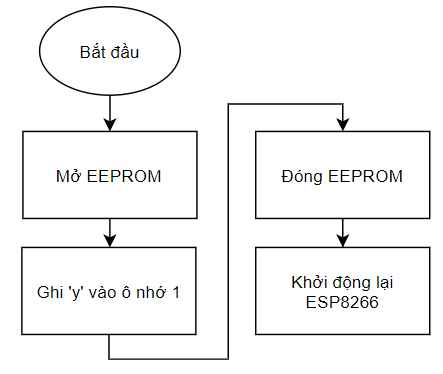
* Hàm xử lý ngắt nút nhấn



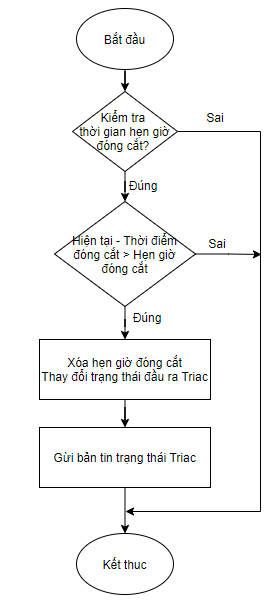
* Hàm kiểm tra nút nhấn giữ



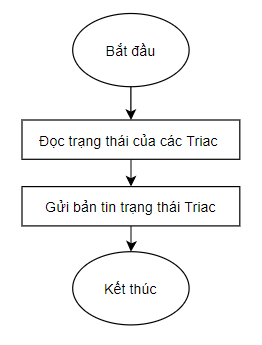
* Hàm xử lý nhấn giữ



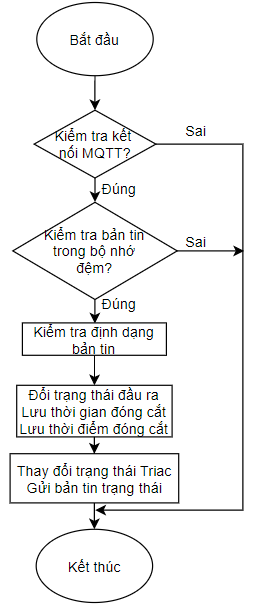
* Hàm hẹn giờ đóng cắt



* Hàm gửi dữ liệu trạng thái đóng cắt



* Hàm xử lý truyền thông MQTT



* Hàm cấu hình WiFi

# Chương 4. Phát triển ứng dụng điều khiển

## Giới thiệu ngôn ngữ Python và module Kivy

## Lưu đồ thuật toán và chương trình cho ứng dụng điều khiển

### Chương trình logic chính

### Chương trình giao diện

Chương trình giao diện được chia làm 2 phần gồm có phần giao diện hiển thị dạng lưới ở trạng thái bình thường và giao diện cửa sổ thông báo khi người dùng sử dụng các nút nhấn:

* Giao diện dạng lưới:

Mô hình giao diện dạng lưới:

|  |  |
| --- | --- |
| Thông tin | |
| Nút nhấn 1 | Nút nhấn 2 |
| Nút nhấn 3 | Nút nhấn 4 |
| Nút cấu hình WiFi | |

Chương trình xây dựng giao diện dạng lưới:

|  |
| --- |
| <MyGrid>:  button1: button1  button2: button2  button3: button3  button4: button4  configWifi: configWifi  infoLabel1: infoLabel1  infoLabel2: infoLabel2  infoLabel3: infoLabel3  GridLayout:  cols: 1  size: root.width, root.height  Label:  id: infoLabel1  size\_hint: 1, 0.05  text: "Sinh vien thuc hien: NGUYEN TUAN ANH\_MSSV: 20173616"  Label:  id: infoLabel2  size\_hint: 1, 0.05  text: "Giang vien huong dan: TS. NGUYEN VAN ANH"  Label:  id: infoLabel3  size\_hint: 1, 0.05  text: "De tai: Thiet ke thiet bi dong cat 4 kenh qua WiFi"  GridLayout:  cols: 2  Button:  id: button1  text: "Cong tac 1"  font\_size: 50  on\_release: root.showPopup(1)  Button:  id: button2  text: "Cong tac 2"  font\_size: 50  on\_release: root.showPopup(2)  Button:  id: button3  text: "Cong tac 3"  font\_size: 50  on\_release: root.showPopup(3)  Button:  id: button4  text: "Cong tac 4"  font\_size: 50  on\_release: root.showPopup(4)  Button:  size\_hint: 1, 0.2  id: configWifi  text: "Cau hinh WiFi"  font\_size: 50  on\_release: root.buttonConfigWifi() |

* Giao diện cửa sổ thông báo

Mô hình giao diện cửa sổ:

|  |
| --- |
| Thông tin |
| Nút nhấn 1 |
| Nút nhấn 2 |
| Nút nhấn 3 |

Chương trình giao diện cho cửa sổ thông báo:

|  |
| --- |
| <PopupTurnOn>:  Button:  text: "Bat"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0.75}  on\_release: root.switch(1, 00)  Button:  text: "Bat 30 giay"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0.5}  on\_release: root.switch(1, 30)  Button:  text: "Bat 1 phut"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0.25}  on\_release: root.switch(1, 60)  Button:  text: "Bo qua"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0}  on\_release: root.close()  <PopupTurnOff>:  Button:  text: "Tat"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0.75}  on\_release: root.switch(0, 00)  Button:  text: "Tat sau 30 giay"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0.5}  on\_release: root.switch(0, 30)  Button:  text: "Tat sau 1 phut"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0.25}  on\_release: root.switch(0, 60)  Button:  text: "Bo qua"  size\_hint: 1, 0.25  pos\_hint: {"x":0, "y":0}  on\_release: root.close() |

# Chương 5. Mô hình và kết quả đạt được

# Kết luận

Trong quá trình thực hiện do kiến thức còn ít ỏi nên có thể gặp nhiều sai sót, em mong được thầy góp ý để bài làm của em trở nên hoàn thiện hơn.

Em xin chân thành cảm ơn.

# Tài liệu tham khảo