

Trường Đại Học Bách Khoa Tp. Hồ Chí Minh  
Khoa Khoa Học & Kỹ Thuật Máy Tính



MÔN HỆ THỐNG NHÚNG

---

PROJECT

# NHẬN DIỆN GIỌNG NÓI CHO SMARTHOME VÀ ĐỒNG BỘ DỮ LIỆU QUA MQTT CLOUD

---

CBGD:

TS. Phạm Hoàng Anh

**Thành viên thực hiện**

Đặng An Thịnh - 1413765

Hồ Bảo Quốc - 1413171

Trần Minh Thông - 1414272



## Mục lục

<b>1</b>	<b>Giới thiệu đề tài</b>	<b>4</b>
1.1	Giới thiệu chung . . . . .	4
1.2	Chi tiết đề tài . . . . .	4
1.3	Nền tảng . . . . .	4
1.4	Tổng quan tài liệu . . . . .	5
<b>2</b>	<b>Phân tích và thiết kế</b>	<b>6</b>
2.1	Trung tâm xử lý . . . . .	6
2.2	Module cảm biến . . . . .	7
2.3	Module tiếp nhận âm thanh . . . . .	10
2.4	Module điều khiển thiết bị . . . . .	11
2.5	Nền tảng đám mây, MQTT cloud . . . . .	12
2.6	Smartphone . . . . .	13
<b>3</b>	<b>Hiện thực</b>	<b>14</b>
3.1	Cấu trúc cơ bản chương trình . . . . .	14
3.1.1	Raspberry . . . . .	14
3.1.2	Arduino . . . . .	19
3.2	Kết quả . . . . .	20
<b>4</b>	<b>Đánh giá</b>	<b>22</b>
4.1	Phân tích ưu nhược điểm của thiết kế . . . . .	22
4.1.1	Ưu điểm . . . . .	22
4.1.2	Nhược điểm . . . . .	22
4.2	Vấn đề cần biết . . . . .	22
4.3	Đánh giá chi phí . . . . .	22
<b>5</b>	<b>Tương lai</b>	<b>23</b>
	<b>Tài liệu</b>	<b>24</b>



## Danh sách hình vẽ

1	Sơ đồ khối tổng quát . . . . .	6
2	Sơ đồ khối tổng quát . . . . .	7
3	Arduino uno R3 . . . . .	8
4	Module dht11 . . . . .	9
5	Module bh1750 . . . . .	9
6	Tai nghe SteelSeries Siberia V2 Frost Blue . . . . .	10
7	Module relay 5V . . . . .	11
8	Kết nối relay để điều khiển . . . . .	11
9	schematic module relay 5v . . . . .	12
10	Ảnh minh họa Mqtt Cloud . . . . .	12
11	Ứng dụng Mqtt dashboard . . . . .	13
12	Class diagram cho chương trình trên Raspberry . . . . .	14
13	Class MainPage . . . . .	14
14	Class mqttCloud . . . . .	15
15	Chuyển trạng thái cho đèn . . . . .	16
16	Class speechRecognizer . . . . .	17
17	Class sensorListener . . . . .	18
18	Cách kết nối dây giữa raspberry và Arduino . . . . .	18
19	Class buttonListener . . . . .	19



## Danh sách bảng

1	Cách nối dây raspberry với relay 5v . . . . .	16
2	Cách nối relay 5v với đồ dùng điện . . . . .	16
3	Cách nối dây BH1750 với arduino . . . . .	19
4	Cách nối dây raspberry với button . . . . .	19
5	Cách nối dây raspberry với arduino . . . . .	20
6	Cách nối dây BH1750 với arduino . . . . .	20
7	Bảng chi phí giá thiết bị . . . . .	23

## 1 Giới thiệu đề tài

Hiện nay, công nghệ càng ngày càng tập trung nâng cao cuộc sống con người. Công nghệ đang phát triển lên theo từng ngày. Và bằng việc áp dụng những công nghệ đó vào hệ thống nhà thông minh đã đem lại những trải nghiệm vô cùng tuyệt vời. Nay nhóm muốn nâng cao chất lượng đó hơn bằng việc điều khiển bằng giọng nói.

### 1.1 Giới thiệu chung

SmartHome hoặc hệ thống nhà thông minh là một ngôi nhà hay căn hộ được trang bị hệ thống tự động tiên tiến dành cho điều khiển đèn chiếu sáng, nhiệt độ, truyền thông đa phương tiện, an ninh, rèm cửa, cửa và nhiều tính năng khác nhằm mục đích làm cho cuộc sống ngày càng tiện nghi, an toàn và góp phần sử dụng hợp lý các nguồn tài nguyên.

Chắc hẳn cũng đã đôi lần bạn cảm thấy phiền toái vì cái remote không nghe lời mình khi pin bị yếu, những chiếc nút “đỉnh công” sau thời gian dài sử dụng, hay cái điều khiển đã biến mất trong một xó xỉnh nào rồi. đem lại những trải nghiệm người dùng tốt hơn, nhóm quyết định tích hợp khả năng điều khiển thiết bị bằng giọng nói thông qua microphone.

Với việc điều khiển chỉ bằng một lời nói, cuộc sống sẽ trở nên dễ dàng hơn với tất cả mọi người.

Cùng với đó, nhóm cũng tích hợp khả năng điều khiển từ xa và hiển thị trạng thái thông qua thiết bị gần gũi nhất đó chính là chiếc smartphone nhờ sử dụng MqttCloud.

### 1.2 Chi tiết đề tài

Phát triển ứng dụng SmartHome bằng Raspberry với các chức năng sau

- Điều khiển thiết bị trong nhà(đèn, máy lạnh, ...)thông qua giọng nói.
- Đồng bộ dữ liệu trạng thái hoạt động và giá trị cảm biến(nhiệt độ, độ ẩm, ánh sáng) qua cloud.
- Có cơ chế điều khiển thiết bị bật,tắt thủ công khi gặp sự cố.
- Điều khiển bằng thiết bị di động (Smartphone,...).
- Cơ chế điều khiển tự động thiết bị trong nhà nhờ đọc giá trị từ các cảm biến.

### 1.3 Nền tảng

Dưới đây là một số ràng buộc về phần mềm:

- Raspberry phải được cài Window 10 Iot Core.
- Lập trình trên môi trường Visual Studio 2017 với ngôn ngữ C#.
- Lập trình bằng Arduino IDE với ngôn ngữ C++.
- Một số thư viện bắt buộc sẽ được nói cụ thể sau đó.

## 1.4 Tổng quan tài liệu

Dưới đây là phần giới thiệu sơ về các mục của bài báo cáo:

**Chương 1 Giới thiệu đề tài** sẽ giới thiệu tổng quát về bài báo cáo và một số ràng buộc.

**Chương 2 Phân tích và thiết kế** mô tả một cách chi tiết cách sử dụng và thông tin các module được sử dụng.

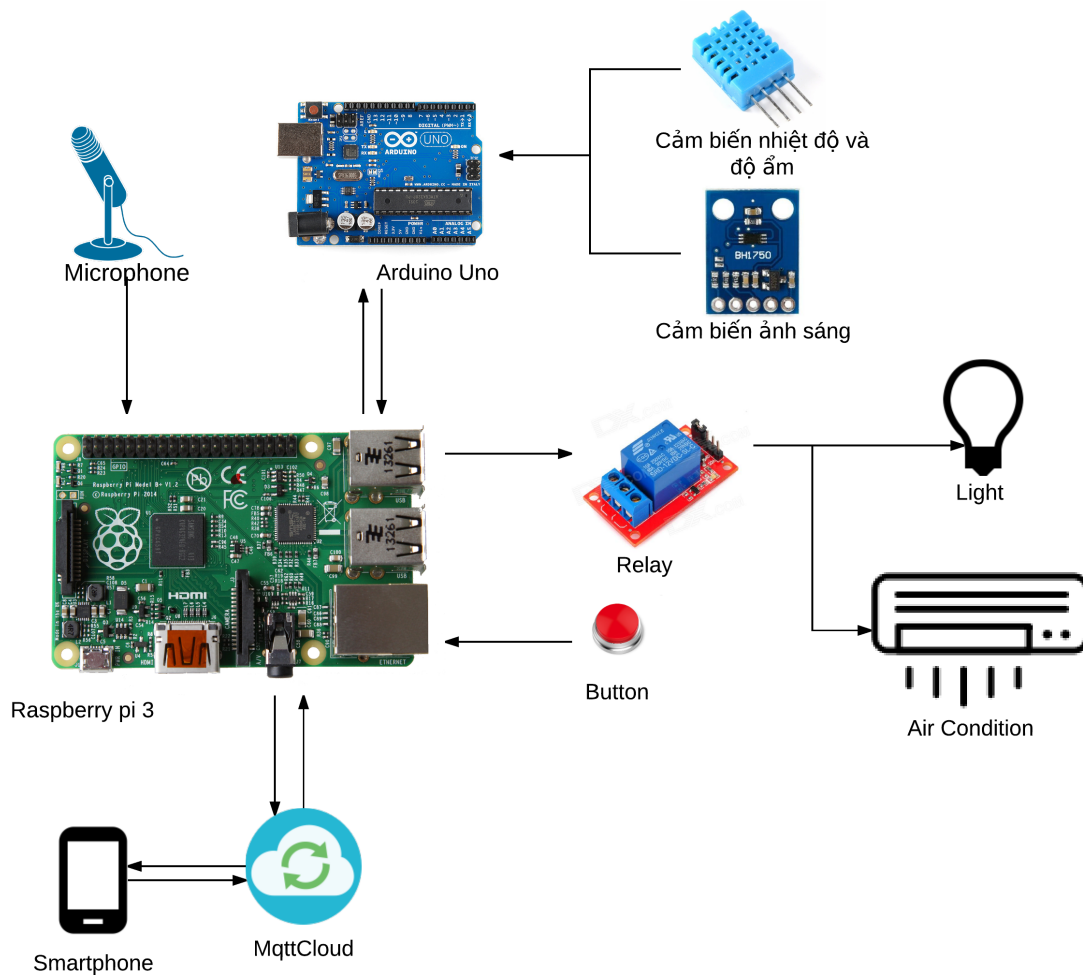
**Chương 3 Hiện thực** giúp hiểu cách tổng quát cách mà nhóm hiện thực project.

**Chương 4 Đánh giá** là giai đoạn tổng kết và đánh giá lại những gì nhóm đã thực hiện.

**Chương 5 Tương lai** sẽ gợi ý những hướng đi tiếp theo nếu project được phát triển tiếp.

## 2 Phân tích và thiết kế

Bằng kiến thức đã có và sự tìm hiểu, nhóm quyết định kết hợp Raspberry pi và Arduino uno để hiện thực project SmartHome. Cụ thể như sau:

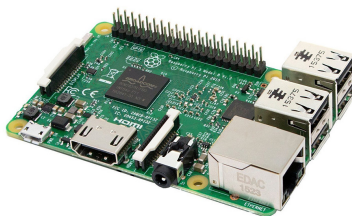


Hình 1: Sơ đồ khối tổng quát

### 2.1 Trung tâm xử lý

Đối với trung tâm xử lý chính, nhóm cần một board mạch khả năng xử lý mạnh mẽ cùng việc định thời bằng OS để đem lại khả năng đáp ứng tốt, điều quan trọng là phải có

khả năng kết nối internet. Vì vậy, nhóm quyết định chọn Raspberry Pi 3 để đáp ứng được những yêu cầu trên.



Hình 2: Sơ đồ khối tổng quát

### 1. Thông số kỹ thuật

- Broadcom BCM2837 chipset running at 1.2 GHz
- 64-bit quad-core ARM Cortex-A53
- 802.11 b/g/n Wireless LAN
- Bluetooth 4.1 (Classic & Low Energy)
- 1 GB LPDDR2 memory
- Supports all the latest ARM GNU/Linux distributions and Windows 10 IoT
- MicroUSB connector for 2.5 A power supply
- 1 x 10/100 Ethernet port
- 1 x HDMI video/audio connector
- 4 x USB 2.0 ports
- 40 GPIO pins
- MicroSD card slot

### 2. Thông số cụ thể trong project

- Raspberry chạy trên nền tảng Windows 10 IoT Core.
- Lập trình bằng ngôn ngữ C# trên Visual Studio 2017.

## 2.2 Module cảm biến

Gồm 3 phần chính:

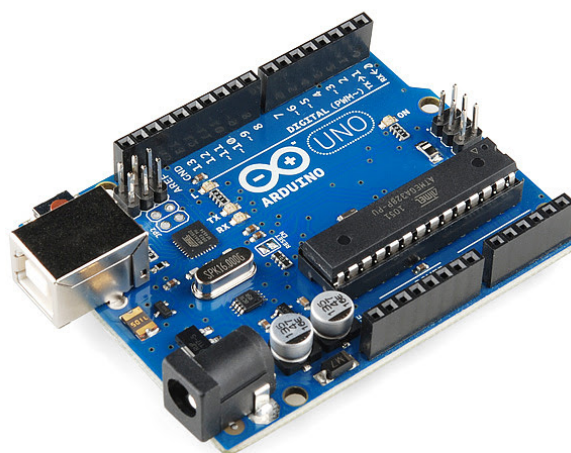
- Trung tâm thu thập dữ liệu và truyền dữ liệu cho Raspberry. Cụ thể là: Arduino Uno R3



- Bộ phận cảm biến nhiệt độ và độ ẩm. Cụ thể là module cảm biến DHT11.
- Bộ phận cảm biến ánh sáng. Cụ thể là module cảm biến BH1750.

Arduino sẽ thu thập dữ liệu từ hai module cảm biến sau đó tổng hợp và tạo frame dữ liệu chuẩn để gửi cho Raspberry thông qua giao thức I2C. Thông số kỹ thuật cụ thể như sau:

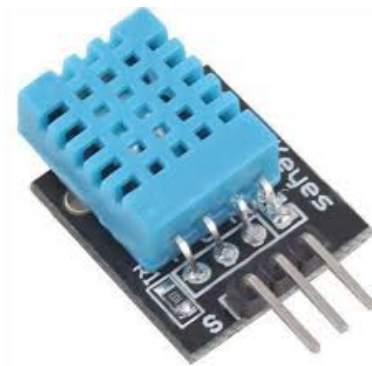
#### 1. Arduino uno R3



Hình 3: Arduino uno R3

- Vi điều khiển ATmega328P
- Điện áp hoạt động 5V
- Digital I/O pin 14 (trong đó 6 pin có khả năng băm xung)
- Flash Memory 32 KB (ATmega328P)
- 0.5 KB được sử dụng bởi bootloader
- SRAM 2 KB (ATmega328P)
- EEPROM 1 KB (ATmega328P)
- Tốc độ 16 MHz

#### 2. Module cảm biến nhiệt độ và độ ẩm DHT11



Hình 4: Module dht11

Thông số kỹ thuật:

- Đo độ ẩm: 20%-95%
- Nhiệt độ: 0-50°C
- Sai số độ ẩm: 5%
- Sai số nhiệt độ: 2°C

### 3. Module cảm biến ánh sáng BH1750



Hình 5: Module bh1750

Thông số kỹ thuật:

- Bộ chuyển đổi cường độ ánh sáng sang tín hiệu kỹ thuật số
- Ngưỡng đo rộng và độ phân giải cao. ( 1 – 65535 lx )
- Dòng tiêu thụ thấp, tiết kiệm điện

- Chức năng chống nhiễu ánh sáng tần số 50Hz / 60Hz
- Giao tiếp I2C(Hỗ trợ chế độ f / s)
- Không cần linh kiện phụ
- Có thể chọn 2 kiểu slave-address của IC.
- Đo cường độ ánh sáng từ 0.11 lx – 100000 lx

### 2.3 Module tiếp nhận âm thanh



Hình 6: Tai nghe SteelSeries Siberia V2 Frost Blue

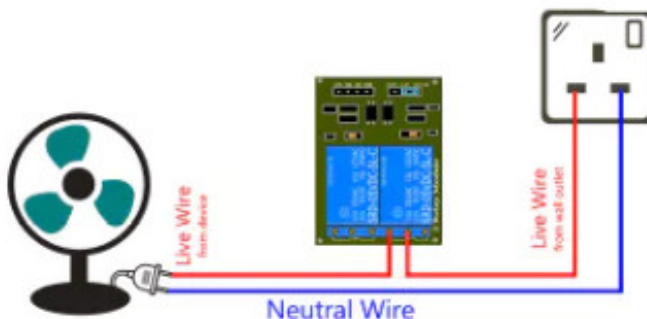
Nhóm sử dụng tai nghe SteelSeries Siberia V2 Frost Blue để thu nhận âm thanh thông qua micro và gửi về cho Raspberry xử lý. Tai nghe này sử dụng cổng USB để giao tiếp với Raspberry.

## 2.4 Module điều khiển thiết bị

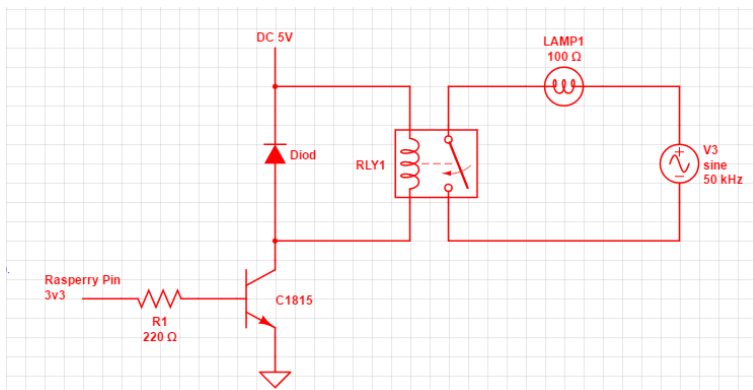


Hình 7: Module relay 5V

Để điều khiển được thiết bị hoạt động với nguồn 220V, chúng ta cần sử dụng relay. Nhưng Relay mức điều khiển thấp nhất là 5V mà Raspberry chỉ có chân GPIO điều khiển ở mức 3V3 nên việc sử dụng chân 3V3 điều khiển có thể mang đến sự bất ổn định. Giải pháp, sử dụng module tăng dòng từ 3V3 lên 5V để việc điều khiển ổn định hơn. Cách kết nối đơn giản cho nguồn 220V:

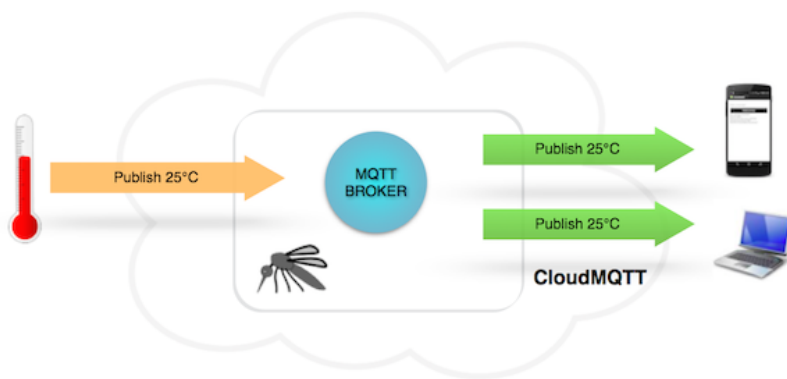


Hình 8: Kết nối relay để điều khiển



Hình 9: schematic module relay 5v

## 2.5 Nền tảng đám mây, MQTT cloud



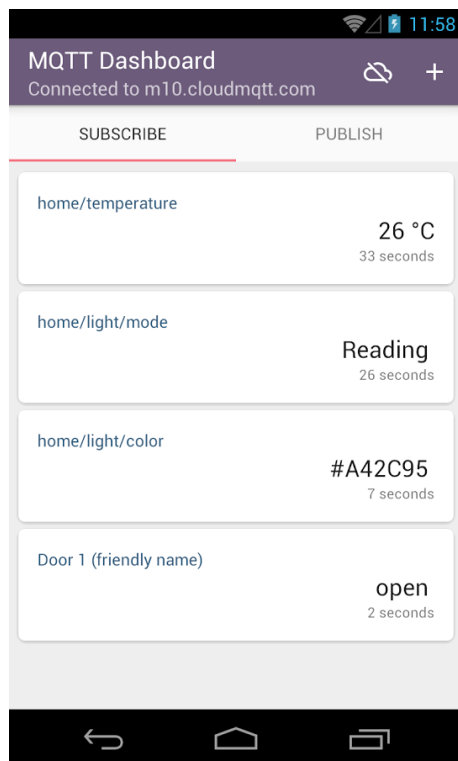
Hình 10: Ảnh minh họa Mqtt Cloud

Mqtt cloud là giao thức "machine-to-machine" trong tương lai. Giải pháp tuyệt vời cho Internet of thing. Các tin nhắn giao tiếp giữa chúng được đưa vào hàng đợi, người gửi và người nhận không cần phải tương tác cùng lúc. Các tin nhắn sẽ nằm trong hàng đợi cho đến khi người nhận lấy chúng hoặc các tin nhắn "time-out".

Nhóm sẽ sử dụng mqtt cloud này trong mục đích lấy thông tin từ các cảm biến và cả việc điều khiển thiết bị.

Vì thư viện M2mqtt cho C# không có cơ chế reconnect khi gặp sự cố mạng nên nhóm có làm thêm khả năng reconnect khi gặp sự cố mạng.

## 2.6 Smartphone



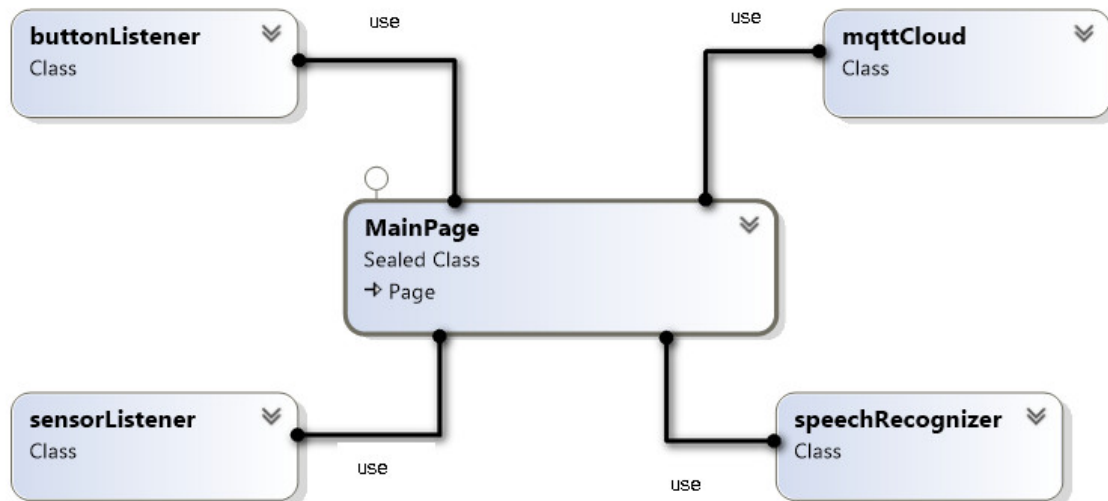
Hình 11: Ứng dụng Mqtt dashboard

- Để đơn giản hóa project, nhóm chọn chương trình điều khiển và hiển thị status bằng ứng dụng MQTT dashboard trên kho ứng dụng CH Play. Ứng dụng có khả năng kết nối với MQTT cloud một cách tự động. Ngoài ra, ứng dụng còn có giao diện hỗ trợ cho việc điều khiển thiết bị và hiển thị trạng thái thiết bị.

### 3 Hiện thực

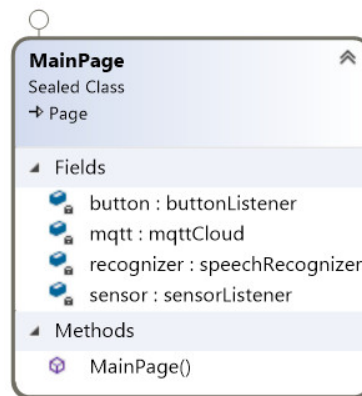
#### 3.1 Cấu trúc cơ bản chương trình

##### 3.1.1 Raspberry



Hình 12: Class diagram cho chương trình trên Raspberry

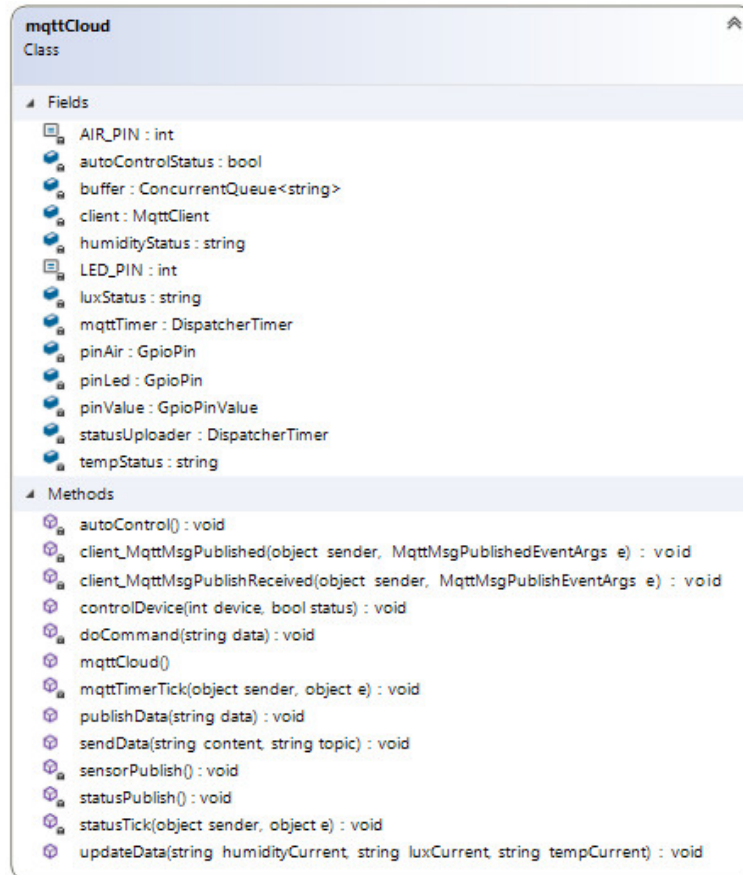
##### 1. Class MainPage



Hình 13: Class MainPage

- Chương trình chính, thực hiện khởi tạo và gọi các Class con.

## 2. Class mqttCloud



Hình 14: Class mqttCloud

- Class này đảm nhận nhiệm vụ gửi và nhận dữ liệu từ mqttCloud nhờ thư viện M2mqtt và điều khiển thiết bị.
- Cách nối dây giữa relay 5V và raspberry:



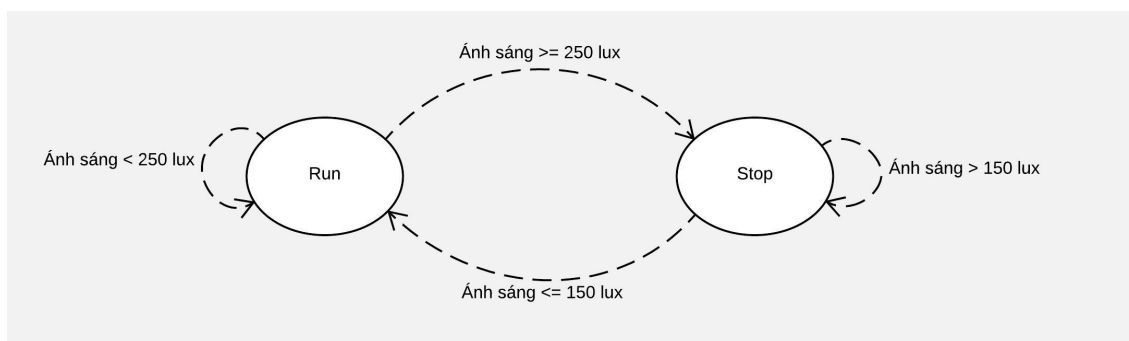
Chân raspberry	Chân relay 5v
5V	DC+
GND	DC-
pin 19	IN

Bảng 1: Cách nối dây raspberry với relay 5v

Tên	Chân relay 5v
nguồn 220V( chân nóng)	NC
nguồn 220V( chân lạnh)	COM

Bảng 2: Cách nối relay 5v với đồ dùng điện

- Việc gửi dữ liệu, nhóm đã tạo riêng một "buffer" để kiểm tra dữ liệu trước khi gửi và đảm bảo việc gửi dữ liệu tin cậy hơn. Các dữ liệu trước khi được gửi lên mqttCloud sẽ được đưa vào buffer bằng hàm *sendData*.
- Các dữ liệu từ sensor được chứa tạm trong các biến: *luxStatus*( chứa giá trị ánh sáng), *tempStatus*( chứa giá trị nhiệt độ), *humidityStatus*( chứa giá trị độ ẩm).
- Nhóm sử dụng hai biến để tạo interrupt là *mqttTimer* và *statusUploader*. Đi kèm là các hàm xử lý interrupt là *statusStick*(đưa dữ liệu vào buffer để gửi lên Mqtt Cloud theo thời gian định trước) và *mqttTimerStick*(hàm kiểm tra dữ liệu trong buffer và gửi lên Mqtt Cloud đến khi buffer rỗng).

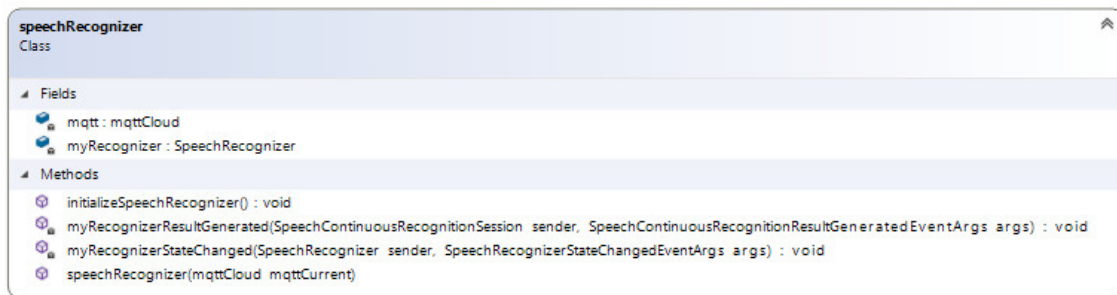


Hình 15: Chuyển trạng thái cho đèn

- Ngoài ra còn có hàm *autoControl* phục vụ cho chế độ tự động điều khiển đèn và máy lạnh theo điều kiện hợp lý. Hình 13 là sơ đồ chuyển trạng thái cho đèn,

với máy điều hòa cũng tương tự nhưng với tham số là nhiệt độ và 2 mức giới hạn trên dưới là  $28^{\circ}C$  và  $24^{\circ}C$ .

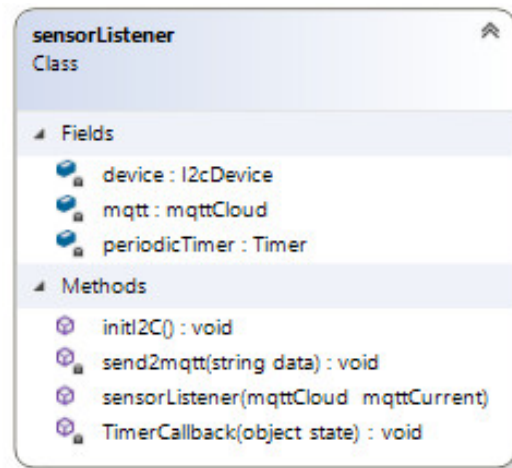
### 3. Class speechRecognizer



Hình 16: Class speechRecognizer

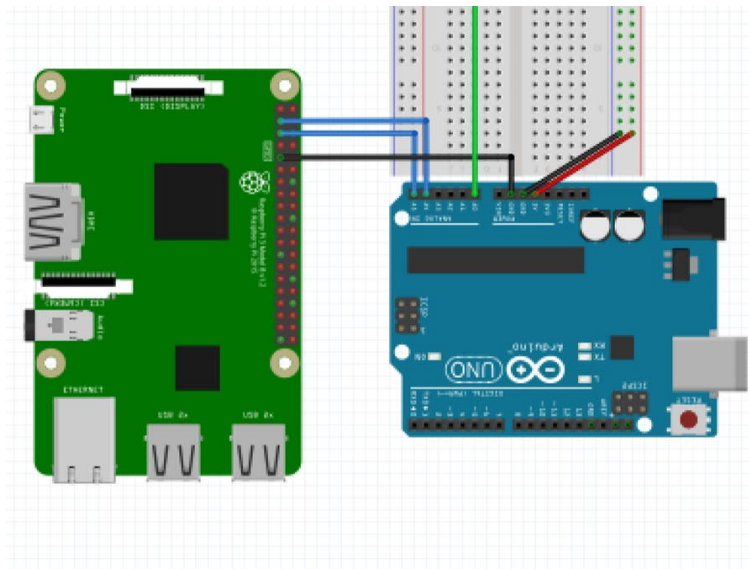
- Class dùng để xử lý tín hiệu âm thanh và chuyển nó về dạng chữ nhờ thư viện *SpeechRecognition* và các định nghĩa Grammar tự xây dựng.
- Khi bộ recognizer của thư viện *SpeechRecognition* nhận dạng được âm thanh và chuyển nó về dạng text, tiếp theo, text đó sẽ được so sánh với các định nghĩa trong grammar để trả kết quả về cho người lập trình.
- Sau khi khởi tạo bằng lời gọi *initializeSpeechRecognizer* từ mainPage, chương trình sẽ thực hiện subscribe hai hàm *myRecognizerStateChanged* ( hiển thị trạng thái thu nhận âm thanh) và *myRecognizerResultGenerated* ( xử lý âm thanh nhận được sau khi so sánh với trong Grammar).

### 4. Class sensorListener



Hình 17: Class sensorListener

- Class này dùng để giao tiếp với board Arduino, lấy dữ liệu thông qua các gói tin được định nghĩa trước từ Arduino gửi đến bằng giao tiếp I2C thông qua thư viện *Devices.I2c*.



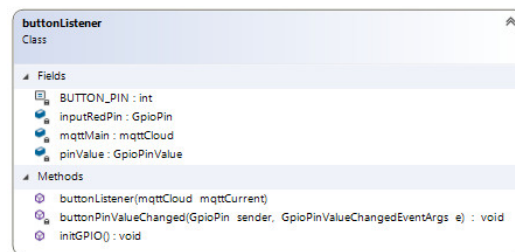
Hình 18: Cách kết nối dây giữa raspberry và Arduino

Chân raspberry	Chân Arduino
GND(pin 9)	GND
SDA(pin 3)	A4
SCL(pin 5)	A5

Bảng 3: Cách nối dây BH1750 với arduino

- Class có một timer *periodic* để "pool" dữ liệu từ Arduino theo thời gian định trước bằng hàm *TimerCallback* và cập nhật giá trị các biến ở class *mqttCloud* thông qua hàm *send2mqtt*.

## 5. Class buttonListener



Hình 19: Class buttonListener

- Class này lắng nghe chân GPIO đã kết nối với button, dùng thực hiện lệnh điều khiển bằng tay với người dùng. Class sử dụng thư viện chính là *Devices.Gpio* để lắng nghe chân GPIO.
- Với hai hàm đơn giản là hàm tạo *initGPIO* và hàm thực hiện lệnh điều khiển khi button được nhấn là *buttonPinValueChanged*.

Chân raspberry	Chân button
Pin 19	Chân tín hiệu

Bảng 4: Cách nối dây raspberry với button

### 3.1.2 Arduino

Việc giao tiếp giữa arduino và raspberry ta sẽ sử dụng giao tiếp I2C giữa 2 thiết bị này dựa vào thư viện "Wire.h" có sẵn của arduino. Cùng với đó, việc lấy giá trị nhiệt độ và độ ẩm từ dht11 nhờ thư viện "dht.h", và lấy giá trị ánh sáng nhờ thư viện "BH1750FVI.h".

Các thư viện đều đính kèm trong bộ code đi kèm tài liệu.

Ngoài ra, để tiện cho việc lấy giá trị từ sensor đơn giản hơn, nhóm đã tạo class sensor-Controller với nhiệm vụ đơn giản là một interface, hàm chính chỉ việc gọi các hàm có sẵn để lấy giá trị cần thiết.

Việc giao tiếp I2C giữa 2 board rất đơn giản. Đầu tiên, raspberry gửi một lệnh pull xuống cho arduino. Arduino sẽ có 3 bên để lưu giá trị ánh sáng, nhiệt độ và độ ẩm. Arduino sẽ tạo frame dữ liệu dựa vào giá trị 3 biến đó và gửi lại cho raspberry.

Cách nối dây cho việc giao tiếp arduino với DHT11:

Chân DHT11	Chân Arduino
GND	GND
Vcc	5V
Signal	D2

Bảng 5: Cách nối dây raspberry với arduino

Cách nối dây cho việc giao tiếp arduino với BH1750:

Chân BH1750	Chân Arduino
GND	GND
Vcc	3.3V
SDA	A4
SCL	A5
ADDR	A3

Bảng 6: Cách nối dây BH1750 với arduino

### 3.2 Kết quả

Project của nhóm đã đảm bảo đầy đủ các chức năng mà nhóm đã đưa ra từ trước và còn có thêm nhiều tính năng mở rộng.

Ngoài ra, nhóm còn thực hiện một vài bài test đặc biệt để kiểm tra khả năng chịu lỗi và xử lý lỗi của trung tâm điều khiển.

Một số hình ảnh sản phẩm: [www.google.com.vn](http://www.google.com.vn)



Link video:

## 4 Đánh giá

### 4.1 Phân tích ưu nhược điểm của thiết kế

#### 4.1.1 Ưu điểm

- Dễ dàng hiện thực các chức năng nhờ vào các thư viện có sẵn.
- Tận dụng tốt các kỹ thuật có sẵn để áp dụng vào project
- Tối ưu hóa chi phí trong việc mua các linh kiện nhưng đem lại hiệu quả cao và kết quả chính xác.
- Project được thiết kế với khả năng chịu lỗi nhất định.

#### 4.1.2 Nhược điểm

- Giao thức giao tiếp giữa các thiết bị còn chưa rõ ràng.
- Còn phụ thuộc nhiều vào module có sẵn.
- Nhận dạng giọng nói vẫn chưa chuẩn xác.

### 4.2 Vấn đề cần biết

- Giao tiếp I2C: việc giao tiếp giữa raspberry và arduino có sự bất ổn định vì chân giao tiếp của raspberry là 3v3 và arduino là 5v, cần module tăng dòng để đảm bảo việc giao tiếp ổn định hơn.
- Thư viện M2mqtt: vì thư viện này không hỗ trợ khả năng tự động reconnect hay thời gian time out của message nên khi muốn phát triển những tính năng cao hơn phải tự nâng cấp bằng tay.
- Microphone: nên chọn micro có chức năng chống ồn vì bộ thư viện nhận diện không có khả năng lọc tạp âm nên việc nhận âm thanh thường xuyên không chuẩn.

### 4.3 Đánh giá chi phí

Để hiện thực project này, nhóm đã sử dụng các board và module trong bảng dưới đây:

Tên thiết bị	Số lượng(cái)	Giá(VND)
Raspberry	1	1,450,000
Arduino	1	145,000
Module DHT11	1	30,000
Module BH1750	1	50,000
Module Relay 5v	1	30,000
Ổ cắm đầu cái	1	5,000
Tổng		1,710,000

Bảng 7: Bảng chi phí giá thiết bị

## 5 Tương lai

Có vô số tính năng có thể phát triển tiếp tục dựa trên porject này để nâng cao trải nghiệm cho người sử dụng. Những điều nhóm đề xuất dưới đây là những tính năng mà nhóm nghĩ là có thể phát triển được và cần thiết:

- Giao diện người dùng cho Raspberry và hiển thị trên màn hình LCD.
- Phát triển một ứng dụng riêng để điều khiển thiết bị trên smartphone với các tính năng báo mật.
- Nâng cao khả năng bảo mật để điều khiển các thiết bị cần độ an toàn cao như cửa chính, gara.
- Streaming video để theo dõi hình ảnh trong nhà.





## Tài liệu

1. [www.hackster.io](http://www.hackster.io)
2. [www.arduino.vn](http://www.arduino.vn)
3. [www.instructables.com](http://www.instructables.com)
4. [stackoverflow.com](http://stackoverflow.com)
5. [www.cloudmqtt.com](http://www.cloudmqtt.com)