



TRƯỜNG ĐẠI HỌC BÁCH KHOA ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
LẬP TRÌNH HỆ THỐNG VÀ VI ĐIỀU KHIỂN

TÊN ĐỀ TÀI:

THIẾT BỊ CHỐNG TRỘM CHO PHÒNG TRỢ

GIẢNG VIÊN HƯỚNG DẪN: Bùi Thị Thanh Thanh

SINH VIÊN THỰC HIỆN:

Sinh viên 1: Võ Đức Phong 16NH11C (16T1)

Sinh viên 2: Nguyễn Thái Học 16NH11C (16T1)

Sinh viên 3: Nguyễn Văn Hà 16NH11B (15T3)

Đà Nẵng, 20-04-2019

BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Mô tả công việc			Thành viên thực hiện
1	Đưa ra ý tưởng, phân tích và thiết kế hệ thống			Cả nhóm
	Tìm hiểu chức năng, cách sử dụng và mua thiết bị			
	Nghiên cứu sơ đồ mạch để lắp đặt các linh kiện			
2	NODEMCU ESP8266	Quản lý việc kết nối đến mạng wifi gia đình	Đăng nhập vào wifi của Esp8266 phát ra	Võ Đức Phong
			Tìm các trạm phát sóng wifi gần kề	
			Kết nối esp8266 vào mạng wifi mới	
	Kết nối Esp8266 với Server	Kết nối Esp8266 (sau khi đã kết nối thành công với wifi) với Server	Nguyễn Thái Học	
		Nhận dữ liệu từ Server		
		Gửi dữ liệu lên Server		
3	Loa và cảm biến khoảng cách HC-SR04	Kết nối cảm biến khoảng cách HC-SR04 với NODEMCU ESP8266	Võ Đức Phong	
		Kết nối loa với NODEMCU ESP8266		
5	Module cảm biến rung, LED	Kết nối module cảm biến rung với NODEMCU ESP8266	Nguyễn Văn Hà	
		Kết nối LED với NODEMCU ESP8266		
6	Android App	Bật tắt thông báo	Nguyễn Thái Học	
		Bật tắt hoạt động		
		Nhận thông báo khi cửa bị mở		
7	Mô hình	Lắp đặt mô hình	Nguyễn Văn Hà	
8	Kiểm tra và sửa lỗi			Cả nhóm

LỜI MỞ ĐẦU

Ngày nay đại đa số các lĩnh vực sản xuất, điều khiển, giám sát, đo lường,... đều được trang bị hệ thống tự động hóa. Một số vi mạch được sử dụng đó là kỹ thuật vi điều khiển. Nhờ tính năng ưu việt của bộ vi điều khiển như: khả năng lập trình phù hợp với thiết kế nhỏ và lớn cũng như giao tiếp với các thiết bị ngoại vi và máy tính đã đem lại sự hoàn hảo, độ chính xác và tính mềm dẻo cao thông qua giao tiếp giữa người và máy.

Như bạn cũng thấy trên các trang an ninh trật tự thì tình hình tội phạm đang ngày càng gia tăng, đặc biệt là tại thành phố lớn như Đà Nẵng. Số lượng các vụ trộm cắp tài sản gia đình tăng cao với mức độ tinh vi và nguy hiểm. Do đó việc mỗi gia đình nên lắp đặt cho mình một hệ thống chống trộm hay các thiết bị chống trộm gia đình phù hợp là điều hết sức cần thiết. Nắm bắt được nhu cầu bảo vệ tài sản cũng như sự an toàn cho những người thân yêu, nhóm em đã thực hiện đồ án với đề tài: “Thiết bị chống trộm phòng trọ”.

Thực hiện đồ án là cơ hội để chúng em áp dụng, tổng hợp các kiến thức đã học trên lớp, đồng thời đúc kết được những bài học thực tế phục vụ cho việc học tập và làm việc sau này. Mặc dù đã rất cố gắng thực hiện đề tài nhưng vì năng lực cũng như thời gian còn hạn chế nên khó tránh khỏi những thiếu sót, rất mong thầy cô thông cảm. Những góp ý của thầy cô là bài học, là hành trang để em vững bước vào cuộc sống sau này. Qua đây, em xin trân trọng cảm ơn cô Bùi Thị Thanh Thanh, người đã nhiệt tình hướng dẫn, chỉ bảo chúng em trong quá trình thực hiện, hoàn thành đồ án.

Nhóm sinh viên thực hiện

Võ Đức Phong

Nguyễn Thái Học

Nguyễn Văn Hà

MỤC LỤC

BẢNG PHÂN CÔNG CÔNG VIỆC	2
LỜI MỞ ĐẦU	3
MỤC LỤC	4
TÓM TẮT ĐỒ ÁN	5
DANH MỤC HÌNH VẼ	6
I. Giới thiệu đề tài	7
1. Vấn đề cần giải quyết.....	7
2. Kết quả sản phẩm hiện có	7
3. Giải pháp thực hiện	7
4. Linh kiện cần chuẩn bị	8
5. Một số hình ảnh của đề tài	8
II. Phân tích đề tài	10
1. Bộ kết nối wifi.....	10
a) Quản lý việc kết nối đến wifi gia đình:	10
b) Kết nối đến server:.....	14
2. Bộ điều khiển thiết bị	17
a) Cảm biến Rung	17
b) Cảm biến khoảng cách.....	18
c) LED, Loa	21
III. Giải pháp triển khai	25
IV. Kết quả đạt được	28
Đánh giá và kết luận	28
Tài liệu tham khảo	29

TÓM TẮT ĐỒ ÁN

- ✓ Vấn đề cần giải quyết: Đồ án của nhóm em là tạo ra được thiết bị có thể phát hiện trộm đột nhập vào nhà thông qua cảm biến rung và cảm biến khoảng cách được lắp đặt ngay tại cửa ra vào, thông báo và gửi sms tới người dùng thông qua app Android, có thể bật tắt thiết bị và thông báo khi không cần thiết (khi ở nhà), đồng thời lưu lại lịch sử mở cửa.
- ✓ Phương pháp giải quyết: Thiết kế mạch sử dụng NODEMCU ESP8266 sử dụng cảm biến rung và khoảng cách để kiểm tra tình trạng đóng mở cửa, nếu cửa mở thì phát âm thanh cảnh báo đồng thời gửi thông báo tới người dùng thông qua Server và Firebase.
- ✓ Kết quả đạt được: Đáp ứng được mục tiêu ban đầu của đồ án, tạo ra một thiết bị có khả năng cảnh báo khi có trộm đột nhập bất ngờ.

DANH MỤC HÌNH VẼ

Hình 1.5.1 – Mô hình hệ thống chống trộm.

Hình 1.5.2 – Sơ đồ nối mạch của hệ thống chống trộm.

Hình 1.5.3 – Hình ảnh app kết nối cói hệ thống.

Hình 2.1.1 – Hiện thị Access Point để người dùng có thể kết nối vào.

Hình 2.1.2 – Trang chủ quản lý kết nối wifi.

Hình 2.1.3 – Trang thiết lập SSID và PASSWORD của Wifi.

Hình 2.1.4 – Màn hình trang sau khi kết nối với mạng Wifi.

Hình 2.1.5 – Danh sách wifi của thiết bị người dùng.

Hình 2.2.1 – Module cảm biến rung.

Hình 2.2.2 – Cảm biến khoảng cách HC-SR04.

Hình 2.2.3 – Đèn LED.

Hình 2.2.4 – Hai bản âm (bản lớn) và dương (bản nhỏ) trong đèn LED.

Hình 2.2.5 – Loa buzzer 5V.

Hình 3.1 – Mô hình tổng quát.

Hình 3.2 – Màn hình chính.

Hình 3.3 – Màn hình thông báo.

I. Giới thiệu đề tài

1. Vấn đề cần giải quyết

- Phát hiện được tình trạng cửa mở hay đóng
- Cách gửi thông báo tới người dùng và phát cảnh báo tại bộ thiết bị
- Bật tắt thiết bị khi không cần thiết
- Tổ chức dữ liệu trên Sever

2. Kết quả sản phẩm hiện có

- Phát hiện được tình trạng cửa mở hay đóng
- Gửi được thông báo và sms đến người dùng nhưng chỉ trên thiết bị Android
- Phát thông báo khi cửa bị rung và bị mở thông qua loa ở thiết bị
- Bật tắt được thiết bị khi không cần thiết nhưng chỉ có thể bật trên app Android và trực tiếp trên Sever
- Tổ chức được dữ liệu hợp lí và hiệu quả

3. Giải pháp thực hiện

- Lắp đặt thiết bị sao cho khi đóng cửa khoảng cách từ cảm biến khoảng cách đến vật cản (tự lắp đặt) cố định khi cửa mở thì khoảng cách phải thay đổi, từ đó phát hiện tình trạng cửa đóng hay mở
- Kết nối wifi thông qua mạch NODEMCU từ đó kết nối đến Sever thông qua thư viện ESP8266HttpClient.h, khi nhận được sự kiện từ cảm biến thì thay đổi giá trị biến trên Sever, Sever sẽ gửi dữ liệu xuống app Android thông tin bao gồm: cảnh báo, ngày giờ gửi thông báo. Để cảnh báo cho người dùng khi không có kết nối internet nhóm em có làm thêm chức năng nhắn tin về số điện thoại cho người dùng
- Đọc cảm biến, nếu cửa bị rung hoặc mở thì loa phát âm thanh cảnh báo cho những người xung quanh
- Tạo một button trên app Android khi bật hoặc tắt thì giá trị của biến trên Firebase thay đổi theo, thiết bị sẽ lấy giá trị của biến này đầu tiên nếu đang bật thì hoạt động ngược lại thì tắt

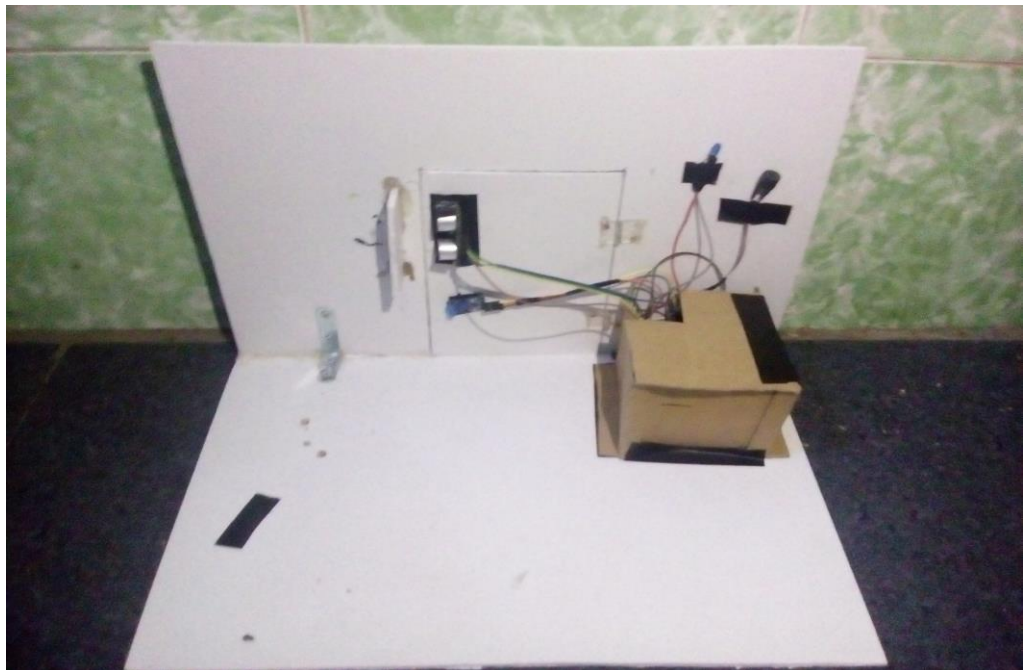
- Dữ liệu gồm biến on_off là biến bật hoặc tắt thiết bị, on_of_send là biến bật tắt gửi dữ liệu trên Sever xuống app, và một node historis lưu danh sách thông tin những lần phòng bị đột nhập

4. Linh kiện cần chuẩn bị

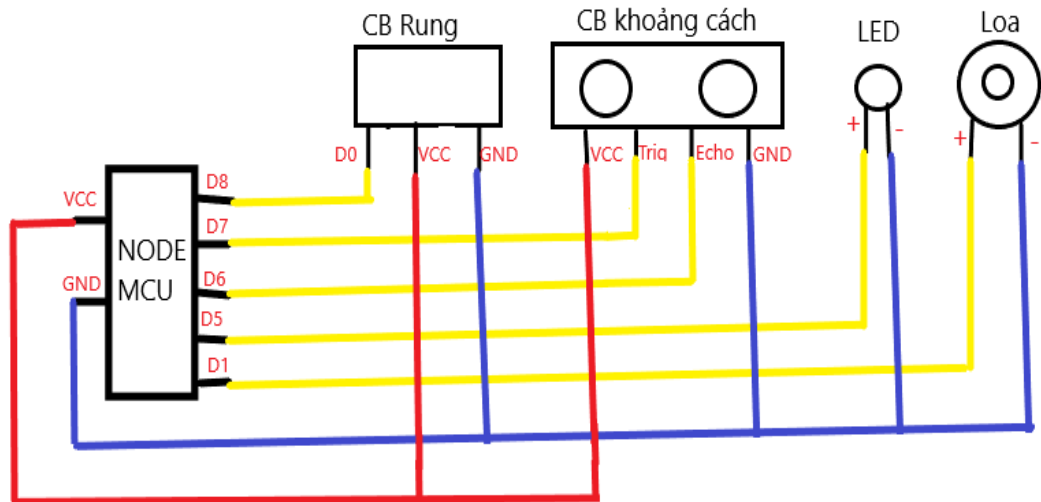
Bảng 1. Linh kiện cần sử dụng

STT	Tên linh kiện	Giá tiền(đ)	Tổng tiền(đ)
1	Mạch NODEMCU ESP8266	90.000	191.000
2	Module cảm biến rung	15.000	
3	Cảm biến khoảng cách	30.000	
4	LED	1.000	
5	Loa	5.000	
6	Dây dẫn, keo, bìa cát tông, ốc vít	50.000	

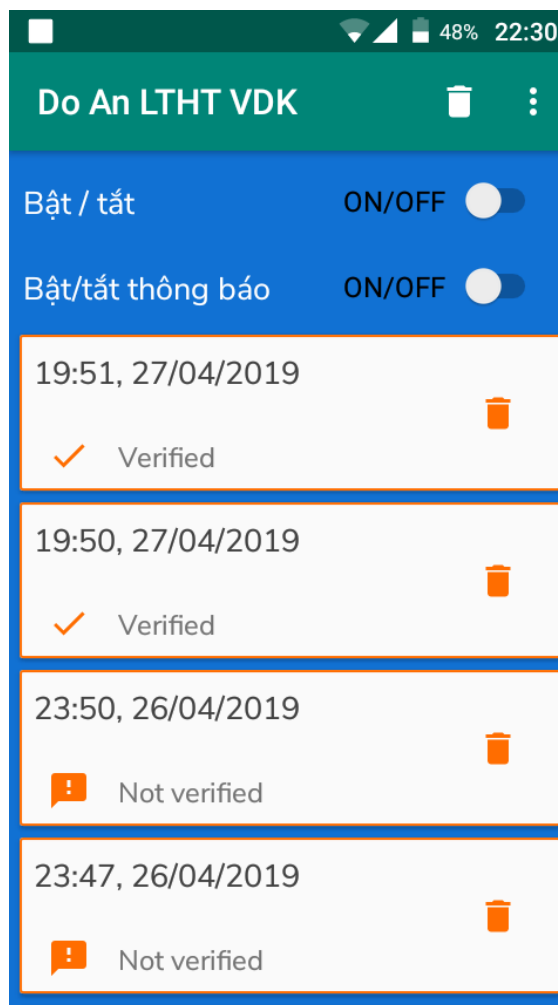
5. Một số hình ảnh của đề tài



Hình 1.5.1 Mô hình hệ thống chống trộm



Hình 1.5.2 Sơ đồ nối mạch của hệ thống



Hình 1.5.3 Hình ảnh app kết nối với hệ thống

II. Phân tích đề tài

1. Bộ kết nối wifi

Để có thể vận hành được hệ thống thì NODEMCU phải giải quyết được một số vấn đề sau:

- Phải Kết nối được đến mạng wifi của gia đình và cho phép kết nối mạng wifi khác khi người dùng đổi mật khẩu/SSID của wifi đã kết nối.
- Phải kết nối được với firebase để có thể nhận được yêu cầu từ người dùng.

Để giải quyết các vấn đề đã đặt ra ở trên thì nhóm em đã làm như sau.

a) Quản lý việc kết nối đến wifi gia đình:

- **Ý tưởng thực hiện**

Đầu tiên nếu người dùng muốn kết nối mạng wifi với mạch NODEMCU thì người dùng phải kết nối đến wifi do mạch NODEMCU phát ra sau đó, truy cập vào địa chỉ ip của NODEMCU và kết nối đến wifi mà mình muốn kết nối, điều này nhằm giúp quản lý tốt hơn phần kết nối wifi cho mạch NODEMCU nếu mạng wifi gia đình có sự thay đổi.

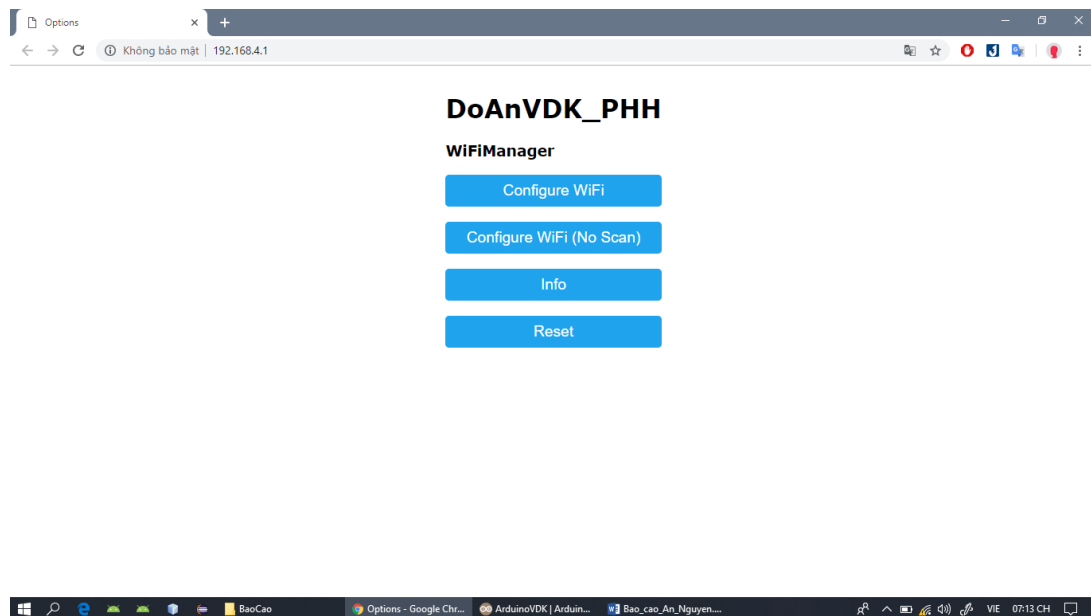
- **Cách thực hiện**

Để thực hiện ý tưởng trên nhóm em đã làm như sau: Thiết lập cho mạch NODEMCU ở chế độ Access Point (Điểm truy cập), lúc này người dùng sẽ thấy một điểm truy cập wifi có tên là DoAnVDK_PHH như hình dưới đây.



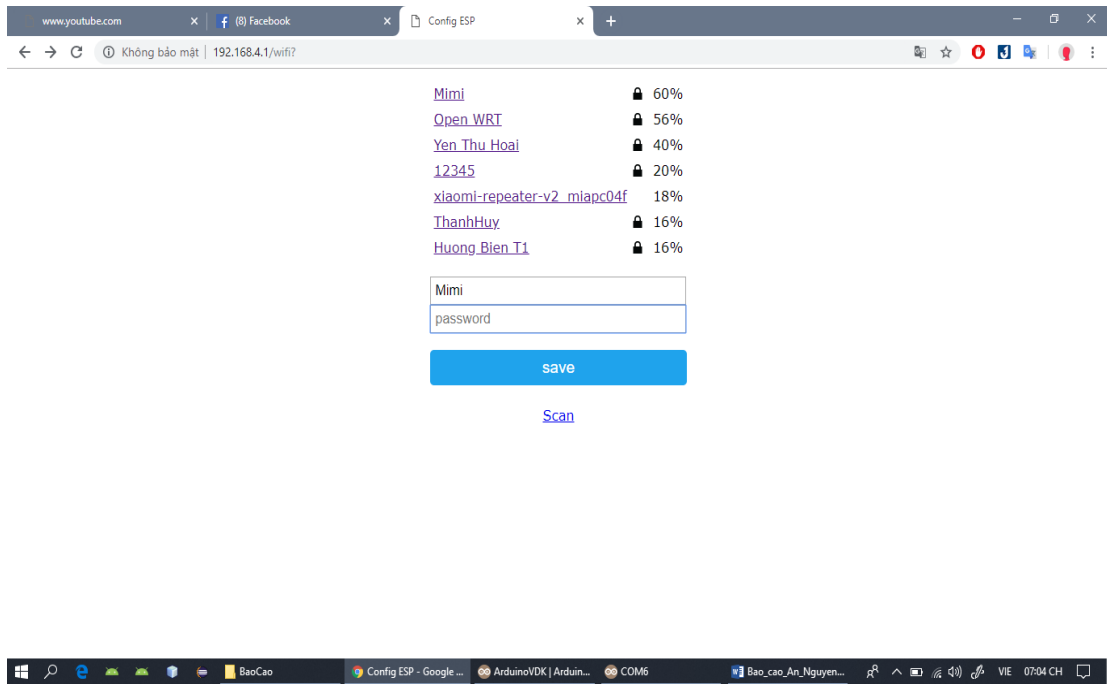
Hình 2.1.1 Hiển thị Access Point để người dùng có thể kết nối vào

Lúc này người dùng sẽ kết nối với mạng wifi với mật khẩu mà nhóm em cung cấp cụ thể ở đây là “12345678”. Sau khi đăng nhập xong thì trình duyệt của người dùng sẽ tự động vào trang có địa chỉ IP “192.168.4.1” và sẽ thấy giao diện như thế này.



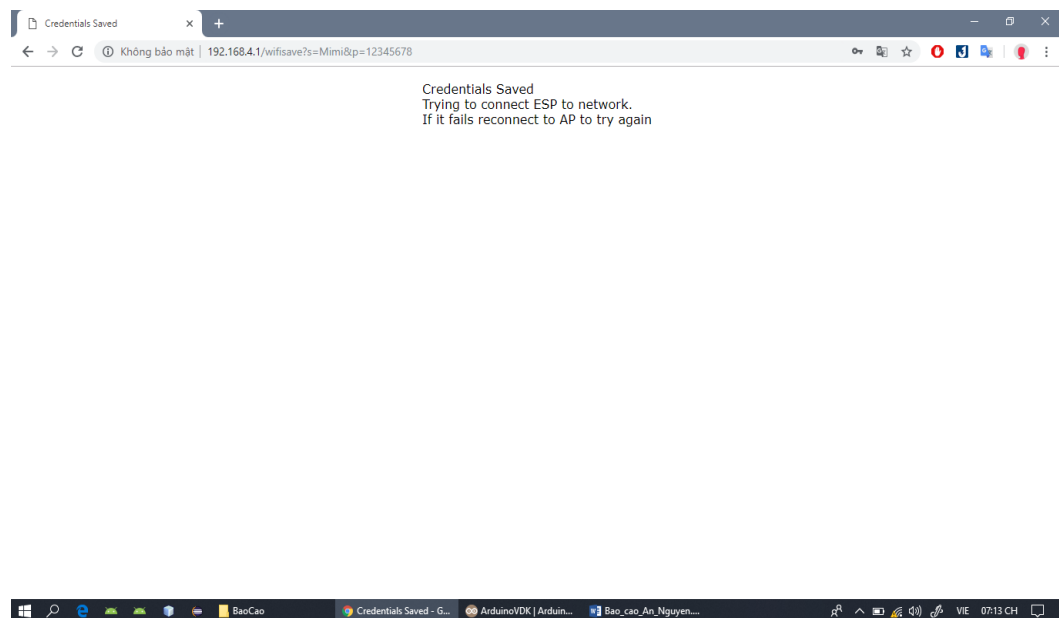
Hình 2.1.2: Trang chủ quản lý kết nối Wifi

Tiếp theo, người dùng lựa chọn Configure WiFi và chọn mạng wifi muốn kết nối, sau đó nhập vào mật khẩu của mạng Wifi đó vào ô password.



Hình 2.1.3: Trang thiết lập SSID và PASSWORD của Wifi

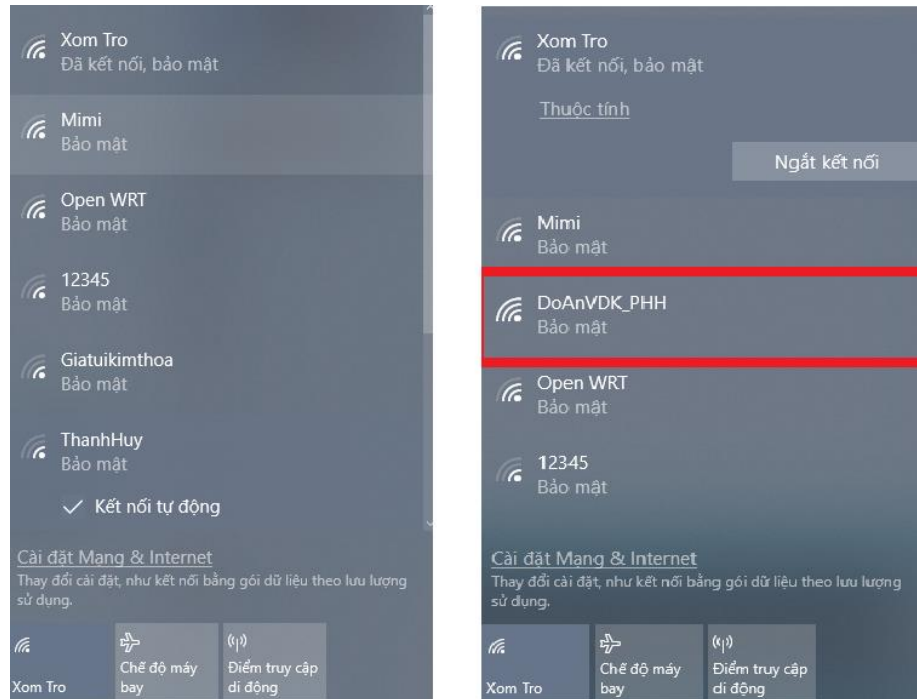
Sau khi nhập vào SSID và mật khẩu thì trang web sẽ chuyển đến màn hình đã kết nối như bên dưới.



Hình 2.1.4: Màn hình trang sau khi kết nối mạng với Wifi

Nếu SSID và PASSWORD chính xác thì khi người dùng vào phần quản lý wifi của máy sẽ không còn thấy Access Point là teamVDK nữa. Còn không thì người

dùng sẽ thấy Access Point là DoAnVDK_PHH lúc này người dùng kết nối lại với DoAnVDK_PHH và thực hiện lại như trên. Hình minh họa bên dưới hình bên trái là kết nối thất bại, bên phải là thành công.



Hình 2.1.5: Hình ảnh danh sách Wifi của người dùng

Tiếp theo là chi tiết mã nguồn của nhóm em.

- Phần khai báo:

```
#include <ESP8266WiFi.h>
#include "WiFiManager.h"
#include "pitches.h"
#include <ESP8266HTTPClient.h>
```

- Phần cấu hình SSID và PASSWORD:

```
#define SSID "DoAnVDK_PHH"
#define PASSWORD "12345678"
```

- Trong phần này em khai báo các thư viện cần thiết trong đó có thư viện WiFiManager (đây là thư viện rất tốt và dễ sử dụng khi dùng để kết nối wifi

cho NODECMU ESP8266) và phần SSID và Password của Access Point mà NODECMU ESP8266 phát ra.

- Để kiểm tra liệu rằng thiết bị đã đăng nhập vào mạng gia đình hay là chưa , em dùng đoạn mã dưới đây.

```
if (!wifiManager.autoConnect(SSID, PASSWORD)) {
    Serial.println("[ERROR] failed to connect and hit
    timeout");
    ESP.reset();
    delay(1000);
}
```

- Trong trường hợp thiết bị vẫn chưa đăng nhập vào mạng wifi nào thì sẽ chuyển sang Mode Access Point với tên thiết bị là “DoAnVDK_PHH” và có mật khẩu là “12345678”.
- Trong phần mã nguồn chúng em có dung hàm configModeCallback để in thông tin mạng Wifi ra Serial để kiểm tra cách làm như sau:

Hàm configModeCallback:

```
void configModeCallback (WiFiManager *myWiFiManager) {
    Serial.println("Entered config mode");
    Serial.println(WiFi.softAPIP());
    Serial.println(myWiFiManager->getConfigPortalSSID());
    Serial.println("-----");
}
}
```

Trong hàm setup() thêm dòng lệnh sau:

```
wifiManager.setAPCallback(configModeCallback);
```

b) Kết nối đến server:

Thiết bị sẽ thực hiện request lên server, server sẽ đọc và ghi vào Firebase database. Phía android app sẽ kết nối trực tiếp đến Firebase database. Còn phía NODECMU chúng em phải sử dụng 1 server trung gian bởi vì khi sử dụng thư viện firebase-arduino kết nối trực tiếp đến firebase vẫn còn gặp một số vấn đề và lỗi.

- **Ý tưởng thực hiện**

Nhóm em sẽ lập trình làm sao để NODEMCU nhận dữ liệu từ server cũng như gửi dữ liệu lên server. Sau đó thực hiện các yêu cầu có trong server. Ví dụ: như muốn thực hiện đóng hay bật thiết bị thì phải request lên server để lấy response chứa giá trị của biến “on_off”, nếu giá trị này là “1” thì bật thiết bị, nếu là “0” thì tắt thiết bị.

- **Cách thực hiện:**

Để làm được chức năng này thì nhóm em đã làm như sau:

- Phần khai báo header bọn em khai báo thư viện ESP8266HTTPClient cần sử dụng, sau đó là URL của server.

```
#include <ESP8266HTTPClient.h>
const String& BASE_URL =
    "http://node-
    auth081098.herokuapp.com/do_an_lttht_vdk";
```

- Để lấy được dữ liệu của biến “on_off” từ server thì nhóm em sẽ thực hiện GET request lên server thông qua hàm getHTTP như sau:

```
int getHTTP() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.setTimeout(10000);
        http.begin(BASE_URL + "/on_off");
        int httpCode = http.GET();
        if (httpCode > 0) {
            String payload = http.getString();
            Serial.print("[GET_on_off_SUCCESS] = ");
            Serial.println(payload);
            return payload.toInt();
        } else {
            Serial.print("[GET_on_off_ERROR] httpCode = ");
```

```
        Serial.println(httpCode);
        return -1;
    }
    http.end();
} else {
    Serial.print("[GET_on_off_ERROR]    no    connected
wifi");
    return -1;
}
}
```

- Tương tự như phần lấy dữ liệu, phần đưa dữ liệu lên server để thông báo tới Android App thì nhóm em thực hiện POST request lên server thông qua hàm postHTTP như sau:

```
void postHTTP() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.setTimeout(10000);
        http.begin(BASE_URL + "/notification");
        int httpCode = http.POST("");
        if (httpCode > 0) {
            String payload = http.getString();
            Serial.print("[POST_NOTIFICATION_SUCCESS]
result = ");
            Serial.println(payload);
        } else {
            Serial.print("[POST_NOTIFICATION_ERROR]
httpCode = ");
            Serial.println(httpCode);
        }
        http.end();
    }
}
```



```
    } else {  
        Serial.println("[POST_NOTIFICATION_ERROR]      no  
connected wifi");  
    }  
}
```

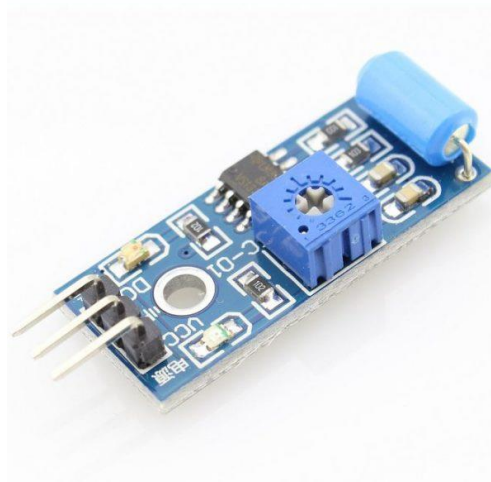
2. Bộ điều khiển thiết bị

Sau khi nhận được dữ liệu từ Sever chúng ta sẽ điều khiển các thiết bị như LED, Loa, Cảm biến rung, Cảm biến khoảng cách để sản phẩm có thể hoàn thành chức năng chống trộm.

a) Cảm biến Rung

Nguyên lí hoạt động:

Khi module cảm biến rung được kích hoạt, khi đó sự thay đổi điện áp tại đầu vào IC LM393. IC này nhận biết được sự thay đổi nói sẽ đưa ra một tín hiệu thấp để báo hiệu có sự rung động



Hình 2.2.1 Module cảm biến rung

Thông số kĩ thuật:

- Đầu ra số: D0
- Led D0-LED báo phát hiện rung động, tín hiệu đầu ra D0 ở mức thấp.
- Dùng Ic LM393 để so sánh điện áp
- Điện áp làm việc: 3.3-5VDC. Có đèn PWR-LED báo nguồn
- Kích thước: 3.2cm x 1.4cm

- Cảm biến rung có 3 chân:

Bảng 2. Chức năng của các chân trên cảm biến rung

Vcc	Nguồn 5V
D0	Ngõ ra của tín hiệu cảm biến
GND	Nối âm

Kết nối:

- Phần header, khai báo các chân kết nối: mắc chân D0 vào D8. Vcc vào cổng 5V, chân GND vào chân GND:

```
#define CAM_BIEN_RUNG D8
```

- Trong hàm loop ta đọc giá trị cảm biến bằng lệnh sau sau:

```
int rung = analogRead(CAM_BIEN_RUNG);
```

Giá trị trả về là 0 hoặc 1023 vì nhóm em dùng module chỉ trả về hai giá trị, 0 là cảm biến chưa bị rung 1023 là cảm biến bị rung có thể điều khiển độ nhạy của cảm biến bằng button trên module.

b) Cảm biến khoảng cách

Nguyên lí hoạt động:

Cảm biến khoảng cách siêu âm HC-SR04 được sử dụng rất phổ biến để xác định khoảng cách vì rẻ và chính xác. Cảm biến sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 -> 300 cm, với độ chính xác gần như chỉ phụ thuộc vào cách lập trình. Cảm biến HC-SR04 có 4 chân là: Vcc, Trig, Echo, GND.



Hình 2.2.2. Cảm biến khoảng cách siêu âm HC-SR04

Để đo khoảng cách, ta sẽ phát 1 xung rất ngắn (5 microseconds - μs) từ chân **Trig**. Sau đó, cảm biến sẽ tạo ra 1 xung HIGH ở chân **Echo** cho đến khi nhận lại được sóng phản xạ ở pin này. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và quay trở lại.

Tốc độ của âm thanh trong không khí là 340 m/s (hằng số vật lý), tương đương với 29,412 $\mu s/cm$ ($10^6 / (340 \cdot 100)$). Khi đã tính được thời gian, ta sẽ chia cho 29,412 để nhận được khoảng cách.

Thông số kỹ thuật:

- Nguồn làm việc: 5V (một số mạch điện tử có thể cấp nguồn 3.3V vẫn hoạt động bình thường nhưng cảm biến siêu âm cần hoạt động ở mức 5V)
- Dòng tiêu thụ : nhỏ hơn 2mA
- Tín hiệu đầu ra: xung HIGH (5V) và LOW (0V)
- Khoảng cách đo: 2cm - 300cm (3 mét)
- Độ chính xác: 0.5cm
- Cảm biến khoảng cách siêu âm có 4 chân:

Bảng 3. Chức năng của các chân trên cảm biến khoảng cách

Vcc	Nguồn 5V
-----	----------

Trig	Một chân Digital output (ngõ phát)
Echo	Một chân Digital input (ngõ thu)
GND	Nối âm

Kết nối:

- Phần header, khai báo các chân kết nối: mắc chân Trig của cảm biến vào chân D7 của NODECMU, chân Echo vào chân D6. Vcc vào cổng 5V, chân GND vào chân GND.

```
#define TRIG D7
```

```
#define ECHO D6
```

- Trong hàm setup, thiết đặt chế độ của chân TRIG là OUTPUT, chân ECHO là INPUT:

```
void setup() {
    // other code
    pinMode(ECHO, INPUT);
    pinMode(TRIG, OUTPUT);
    // other code
}
```

- Hàm dokhoangcach trả về giá trị khoảng cách từ cảm biến tới vật cảm (đơn vị cm):

```
int dokhoangcach() {
    unsigned long duration; // biến đo thời gian
    int distance;           // biến lưu khoảng cách

    /* Phát xung từ chân trig */
    digitalWrite(TRIG, 0); // tắt chân trig
    delayMicroseconds(2);
```

```
digitalWrite(TRIG, 1); // phát xung từ chân trig
delayMicroseconds(5);      // xung có độ dài 5
microSeconds
digitalWrite(TRIG, 0); // tắt chân trig

/* Tính toán thời gian */
// Đo độ rộng xung HIGH ở chân echo.
duration = pulseIn(ECHO, HIGH);
// Tính khoảng cách đến vật.
distance = int(duration / 2 / 29.412);
return distance;
}
```

c) LED, Loa

➤ Đèn LED:

Đèn LED (Light-Emitting Diode) hay còn gọi đi-ốt phát sáng là một linh kiện được sử dụng rất phổ biến bởi vì giá thành rẻ và cách sử dụng đơn giản. Không như điện trở, đèn LED là một linh kiện điện tử có phân cực - trong đó anode là cực dương (+) và cathode (-) là cực âm. Và người ta quy định rằng chân dài hơn của đèn LED là cực dương (+) và chân ngắn hơn là cực âm (-). Trong trường hợp 2 chân bằng nhau, ta có thể xem ở bên trong đèn. Đầu nhỏ (bên trái của hình trên) chính là cực dương và đầu lớn hơn là cực âm của đèn. Vì vậy, khi nối đèn LED vào mạch, ta cần phải nối chính các cực tương ứng.



Hình 2.2.3 Đèn LED



Hình 2.2.4. Hai bán âm (bản lớn) và dương (bản nhỏ) trong đèn LED

Nguyên lý hoạt động:

Khi đèn led hoạt động dựa vào sử dụng chip led phát sáng nhờ công nghệ bán dẫn. Trong khối bán dẫn có hai cực loại P và loại N. Trong đó, khối bán dẫn P được thiết kế có chứa nhiều loại lỗ trống tự do được mang điện tích dương. Khi tiến hành ghép vào khối bán dẫn N (chứa điện tử dạng tự do), các lỗ trống bên khối bán dẫn P chuyển sang khối bán dẫn N. Đồng thời, bên khối P sẽ nhận thêm các điện tích âm được chuyển từ khối N sang. Như vậy, khối bán dẫn P đã tích điện âm và ngược lại khối N tích điện dương. Ở nơi tiếp giáp giữa hai khối bán dẫn này có những điện tử bị các lỗ trống hút vào. Đặc biệt, khi chúng càng tiến lại gần nhau sẽ tạo nên những nguyên tử trung hòa. Cả quá trình này sẽ giải phóng các năng lượng dạng ánh sáng (hay còn được gọi là bức xạ ánh sáng). Đây chính là nguyên lý phát sáng của đèn led. Việc sử dụng hai khối bán dẫn tương tác với nhau theo nguyên tắc vật lý đã giúp đèn led không cần sử dụng lượng lớn điện năng để buộc các điện tử chuyển thành bức xạ ánh sáng. Do vậy, đèn led được coi là thiết bị chiếu sáng tiết kiệm điện năng tối ưu thay thế đèn sợi đốt hoặc đèn huỳnh quang buộc phải sử dụng nhiều điện năng để chuyển hóa.

Thông số kỹ thuật:

- Điện áp: 2.0 đến 2.2V.
- Kích thước bóng: đường kính 5mm.
- Số chân: 2 chân.
- Chiều dài chân: 28 mm - 29 mm.

Kết nối:

- Phần header, khai báo các chân kết nối: mắc chân (+) của LED vào chân D5 của NODECMU:

```
#define LED D5
```

- Trong hàm setup, thiết đặt chế độ của chân LED là OUTPUT:

```
void setup() {  
    // other code  
    pinMode(LED, OUTPUT);  
    // other code  
}
```

- Trong hàm loop, kiểm tra điều kiện bởi các giá trị cảm biến, từ đó bật tắt đèn LED:

Bật đèn: `digitalWrite(LED, HIGH);`

Tắt đèn: `digitalWrite(LED, LOW);`

➤ **Loa buzzer:**

Buzzer còn gọi là loa mini hay còi báo. Nó dùng để phát ra các âm thanh. Nó gồm 2 chân: Chân GND nối GND arduino và chân tín hiệu nối pin arduino. Ta sẽ lập trình để nó phát ra cao độ hay cường độ hay âm sắc gì đó thay đổi từ đó tạo ra những bản nhạc thú vị mang phong cách arduino.



Hình 2.2.5. Loa buzzer 5V

Thông số kỹ thuật:

- Nguồn: 3.5V – 5.5V.

- Dòng điện tiêu thụ: 25mA.
- Tần số cộng hưởng: $2300\text{Hz} \pm 500\text{Hz}$.
- Biên độ âm thanh: 80 dB.
- Nhiệt độ hoạt động: $-20\text{ }^{\circ}\text{C}$ đến $70\text{ }^{\circ}\text{C}$.
- Kích thước : Đường kính 12mm, cao 9,7mm.

Kết nối:

- Phần header, khai báo các chân kết nối: mắc chân tín hiệu của buzzer vào chân D1 của NODECMU:

```
#define LOA D1
```

- Trong hàm loop, kiểm tra điều kiện bởi các giá trị cảm biến, từ đó phát âm thanh ra buzzer hay không:

Tắt âm thanh: `noTone(LOA);`

Phát âm thanh: sử dụng hàm play như sau:

```
// notes in the melody:
```

```
int melody[] = {
```

```
    NOTE_C4,  NOTE_G3,  NOTE_G3,  NOTE_A3,  NOTE_G3,  0,
    NOTE_B3, NOTE_C4
```

```
};
```

```
// note durations: 4 = quarter note, 8 = eighth note,
etc.:
```

```
int noteDurations[] = {
```

```
    4, 8, 8, 4, 4, 4, 4, 4
```

```
};
```

```
void play() {
```

```
    Serial.println("[PLAY]");
```

```
    // iterate over the notes of the melody:
```

```
    for (int thisNote = 0; thisNote < 8; thisNote++) {
```

```
        // to calculate the note duration, take one second
        divided by the note type.
```

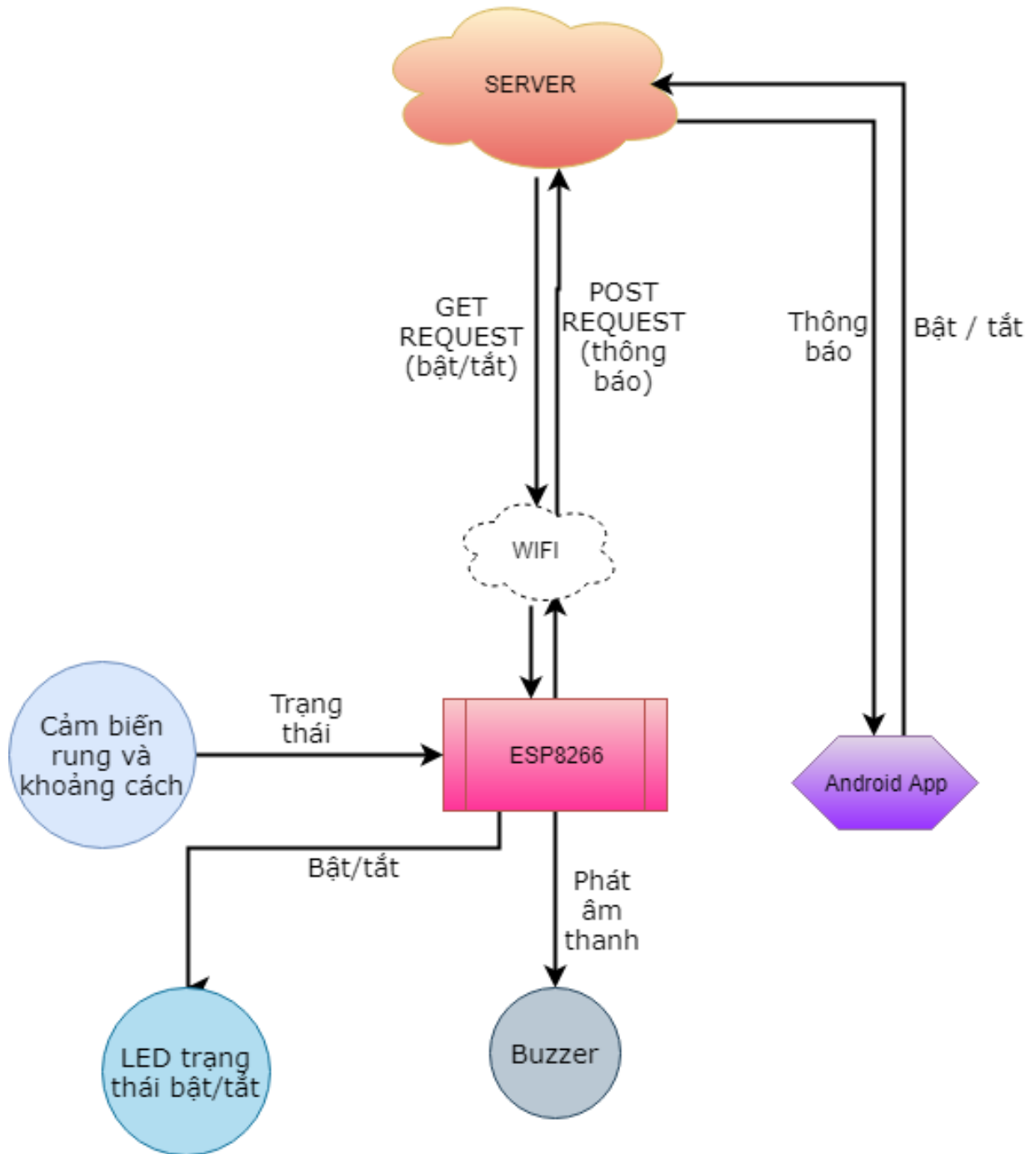


```
// e.g. quarter note = 1000 / 4, eighth note =  
1000/8, etc.  
  
int noteDuration = 1000 / noteDurations[thisNote];  
tone(LOA, melody[thisNote], noteDuration);  
  
// to distinguish the notes, set a minimum time  
between them.  
  
// the note's duration + 30% seems to work well:  
int pauseBetweenNotes = noteDuration * 1.30;  
delay(pauseBetweenNotes);  
// stop the tone playing:  
noTone(LOA);  
}  
Serial.println("[END_PLAY]");  
}
```

III. Giải pháp triển khai

Nguyên lý hoạt động:

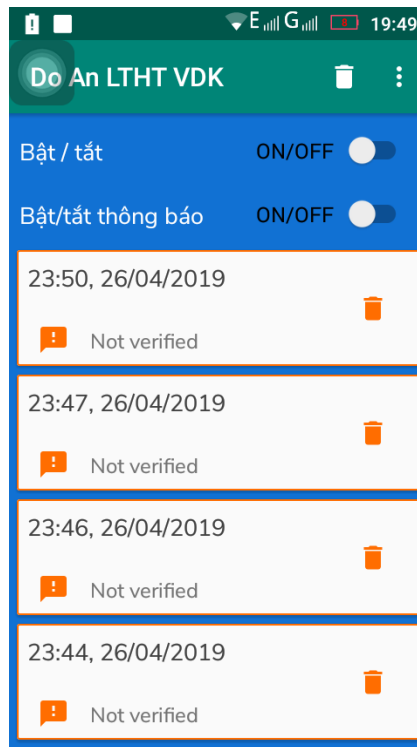
- B1: kết nối esp8266 đến mạng Wifi
- B2: thực hiện GET request từ server để xem trạng thái bật/tắt của thiết bị
- B3: nếu tắt thì tắt đèn và quay lại B2, nếu bật thì tiếp tục thực hiện B4
- B4: đọc giá trị cảm biến rung và khoảng cách
- B5: kiểm tra nếu cửa bị rung thì phát âm thanh ra loa, nếu cửa bị mở thì thực hiện POST request lên server để gửi thông báo về Android App, và quay lại B2



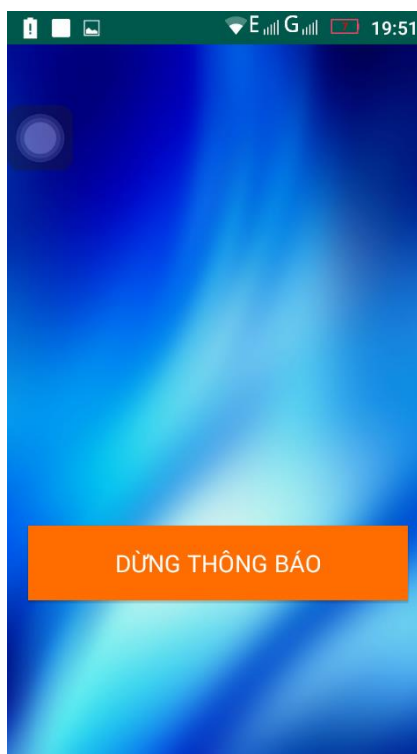
Hình 3.1. Mô hình tổng quát

Ở phần trên em đã trình bày chi tiết về cách mà chúng em thiết lập, kết nối và lập trình cho mạch hoạt động. Tiếp theo, em sẽ trình bày về việc triển khai dự án, ở phần này em sẽ giới thiệu về App Android. App Android này được viết bằng ngôn ngữ Java và sử dụng server là Firebase bởi vì tính realtime và độ bảo mật của nó.

App có mục đích bật tắt thiết bị, bật tắt thông báo, nhận thông báo khi cửa bị mở, xem lại lịch sử mở cửa, xóa lịch sử mở cửa.



Hình 3.2. Màn hình chính



Hình 3.3. Màn hình thông báo

IV. Kết quả đạt được

Sinh viên trình bày các kết quả đạt được, các chức năng đã triển khai/thử nghiệm, kết quả thực hiện các chức năng, đã thử chức năng bao nhiêu lần, vận hành, kết quả, độ ổn định, ...

Qua thời gian nghiên cứu tìm hiểu chúng em đã làm được các kết quả dưới đây:

1. Xây dựng được Android App để cho phép người dùng có nhận thông báo và xem lại lịch sử cửa bị mở, tắt bật thiết bị, bật tắt thông báo.
2. Thông báo bằng loa buzzer khi cửa bị rung, và vừa thông báo bằng loa buzzer và gửi thông báo và sms tới Android App khi cửa bị mở.
3. Lập trình để cho NODEMCU có thể kết nối với mạng wifi gia đình và có thể kết nối vào mạng wifi mới.
4. Vẫn gặp 1 số vấn đề về cảm biến rung không được nhạy, nhận sms bị delay một thời gian khá nhỏ, khi không có kết nối Internet hoặc kết nối Internet quá chậm thì không nhận được thông báo.

Đánh giá và kết luận

Qua thời gian nghiên cứu tìm hiểu và thực hiện, nhóm chúng em đã đáp ứng được mục tiêu ban đầu của đồ án, tạo ra một thiết bị có khả năng cảnh báo khi có trộm đột nhập bất ngờ, dựa vào đó cũng đáp ứng được một phần yêu cầu đặt ra ban đầu cho thiết bị. Quá trình làm đồ án là quãng thời gian bọn em học được rất nhiều kiến thức bổ ích và thực tế.

Vì chưa thử nghiệm dự án ra ngoài thực tế trong thời gian dài, nên để triển khai thì trong thời gian kế đến chúng em cần thử nghiệm và tối ưu hệ thống liên tục, sẽ cố gắng hoàn thiện để thiết bị có thể hoạt động ổn định nhất có thể.

Nhóm đồ án xin chân thành cảm ơn!

Tài liệu tham khảo

1. Các trang web về arduino:
 - <http://www.arduino.cc>
 - <http://www.arduino.vn>
2. Trang bài viết về esp8266 NodeMCU, HTTP request:
 - <http://arduino.vn/bai-viet/1496-esp8266-ket-noi-internet-phan-1-cai-dat-esp8266-lam-mot-socket-client-ket-noi-toi>
 - <https://techtutorialsx.com/2016/07/17/esp8266-http-get-requests/>
 - <https://www.arduino.cc/en/Tutorial/HttpClient>