**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**──────── \* ───────**



**BÁO CÁO**

**Đồ án thiết kế KTMT**

***Đề tài:***

**Tìm hiểu IOT, thiết kế hệ thống cảnh báo đột nhập qua điện thoại**

**Giảng viên hướng dẫn: TS. Đặng Tuấn Linh**

**Sinh viên thực hiện: Lê Minh Trung - 20173422**

**Lớp: KTMT06 - K62**

**Hà Nội, 6/2021**

**Mục Lục**

[**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI** 1](#_Toc75728946)

[**1.1 Đặt vấn đề** 1](#_Toc75728947)

[**1.2 Mục tiêu và phạm vi đề tài** 1](#_Toc75728948)

[**1.3 Định hướng giải pháp** 2](#_Toc75728949)

[**CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU** 3](#_Toc75728950)

[**2.1 Khảo sát hiện trạng** 3](#_Toc75728951)

[**2.2 Tổng quan chức năng** 3](#_Toc75728952)

[**2.3 Cách thức hoạt động** 4](#_Toc75728953)

[**2.4 Sơ đồ tổng quan hệ thống** 6](#_Toc75728954)

[**CHƯƠNG 3. CÔNG NGHỆ** 8](#_Toc75728955)

[**3.1 Phần cứng** 8](#_Toc75728956)

[**3.1.1 ESP8266 V3 CH340 NODEMCU LUA WIFI** 8](#_Toc75728957)

[**3.1.2 Modul Sim 900A** 9](#_Toc75728958)

[**3.1.3 Cảm biến ánh sáng** 10](#_Toc75728959)

[**3.1.4 Đầu phát laser** 11](#_Toc75728960)

[**3.1.5 Nguồn Power Adaptor AC-DC 5V 2A** 12](#_Toc75728961)

[**3.2 Hướng dẫn sử dụng Module Sim 900A** 13](#_Toc75728962)

[**3.2.1 Giao tiếp Module Sim 900A với Esp8266 để thực hiện chức năng call/sms** 13](#_Toc75728963)

[**3.2.2 Các lênh AT cơ bản** 13](#_Toc75728964)

[**3.2.3 Các lệnh và hàm sử dụng trong Project** 14](#_Toc75728965)

[**3.3 Hướng dẫn sử dụng ESP8266** 17](#_Toc75728966)

[**3.3.1 Sử dụng ESP8266 là 1 điểm Access Point** 17](#_Toc75728967)

[**3.3.2 Sử dụng ESP8266 là Webserver** 19](#_Toc75728968)

[**3.4 ESP8266 làm WebSocket Server** 20](#_Toc75728969)

[**3.4.1 WebSocket là gì?** 20](#_Toc75728970)

[**3.4.2 ESP8266 làm WebSocket Server** 21](#_Toc75728971)

[**3.4.3 Mô hình của WebSocket** 22](#_Toc75728972)

[**3.5 Bộ nhớ EEPROM trong ESP8266** 23](#_Toc75728973)

[**3.5.1 Sử dụng bộ nhớ EEPROM trong ESP8266** 23](#_Toc75728974)

[**3.5.2 Lưu trạng thái vào EEPROM** 23](#_Toc75728975)

[**CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ HỆ THỐNG** 26](#_Toc75728976)

[**4.1 Kết quả** 26](#_Toc75728977)

[**4.2 Đánh giá hệ thống** 28](#_Toc75728978)

[**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 29](#_Toc75728979)

[**5.1 Kết luận** 29](#_Toc75728980)

[**5.2 Hướng phát triển** 29](#_Toc75728981)

[**TÀI LIỆU THAM KHẢO** 29](#_Toc75728982)

**Danh mục hình ảnh**

[**Hình 1: Sơ đồ hoạt động của từng phần: 4**](#_Toc75728389)

[**Hình 2: Thiết lập Webserver 4**](#_Toc75728390)

[**Hình 3: Gửi tin nhắn modul sim 5**](#_Toc75728391)

[**Hình 4: Chương trình cảnh báo 5**](#_Toc75728392)

[**Hình 5: Sơ đồ tổng quan 6**](#_Toc75728393)

[**Hình 6: ESP8266 V3 CH340 NODEMCU LUA WIFI 8**](#_Toc75728394)

[**Hình 7: Sơ dồ mạch ESP8266 V3 CH340 NODEMCU LUA WIFI 8**](#_Toc75728395)

[**Hình 8: Modul Sim 900A 9**](#_Toc75728396)

[**Hình 9: Cảm biến ánh sáng 11**](#_Toc75728397)

[**Hình 10: Đầu phát laser 11**](#_Toc75728398)

[**Hình 11: Nguồn Power Adaptor AC-DC 5V 2A 12**](#_Toc75728399)

[**Hình 12: WebSocket 20**](#_Toc75728400)

[**Hình 13: Mô hình của WebSocket 22**](#_Toc75728401)

[**Hình 14: Sử dụng bộ nhớ EEPROM trong ESP8266 23**](#_Toc75728402)

[**Hình 15: Lưu trạng thái vào EEPROM 23**](#_Toc75728403)

[**Hình 16: Lắp đặt mạch hệ thống 26**](#_Toc75728404)

[**Hình 17: Giao diện màn hình chính Webserver 26**](#_Toc75728405)

[**Hình 18: Giao diện thiết lập Wifi 27**](#_Toc75728406)

[**Hình 19: Giao diện thêm số điện thoại gửi tin nhắn 27**](#_Toc75728407)

# **CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**

## **1.1 Đặt vấn đề**

Như chúng ta đã biết, thời gian gần đây tệ nạn trộm cướp, những vụ đột nhập vào nhà gây ra những vụ thảm sát thương tâm, những vụ trộm cướp ở các cửa hàng điện thoại, vàng bạc hay ngân hàng gia tăng báo động, với những thủ đoạn, mánh khóe tinh vi hơn. Vì vậy việc nâng cao hệ thống an ninh là việc làm cấp thiết. Trên thế giới, hệ thống an ninh luôn được chú trọng, là thứ không thể thiếu trong các chung cư, hộ gia đình với các hệ thống an ninh cực kì hiện đại khi họ áp dụng nhiều loại cảm biến có thể sử dụng sóng cao tần, sóng siêu âm hay hồng ngoại,… để có thể phát hiện vật nhanh và chính xác nhất. Vậy còn ở Việt Nam thì sao? Ngày nay thì người dân cũng ý thức được tầm quan trọng của hệ thống an ninh, cũng đã bắt đầu sử dụng hệ thống an ninh cho gia đình mình, tuy nhiên nó chỉ dừng lại ở mức đơn giản như sử dụng camera để quan sát. Một phần vì hệ thống an ninh ở nước ta chưa phát triển mạnh, một phần vì giá cả quá cao.

Từ những nhu cầu cần thiết, thực tế như vậy em đã đi đến quyết định chọn đề tài “ THIẾT KẾ HỆ THỐNG CẢNH BÁO ĐỘT NHẬP QUA ĐIỆN THOẠI” để nghiên cứu làm đồ án thiết kế KTMT cũng như nhằm tổng hợp và củng cố kiến thức đã học cũng như áp dụng công nghệ tự động vào đời sống thực tiễn đó là ngôi nhà thông minh.

Với đề tài này, em sẽ thiết kế mô hình cảnh báo, sử dụng cảm biến ánh sáng hoặc cảm biến chuyển động. Các cảm biến sẽ gửi tín hiệu về vi điều khiển và gọi điện thoại đến người dùng thông qua Modul Sim 900A khi phát hiện có đột nhập.

## **1.2 Mục tiêu và phạm vi đề tài**

Nắm được những kiến thức về lập trình cho một số loại vi điều khiển cho việc áp dụng vào mô hình hệ thống cảnh báo đột nhập. Mô hình cần hoàn thiện hơn về mặt thiết kế cũng như hiệu năng để có thể phù hợp với đại đa số người dùng. Hiện nay mô hình này đang hướng tới những ngôi nhà ở thành thị và dần phổ cập đến cả nhà ở nông thôn.

Các chức năng của hệ thống cảnh báo đột nhập qua điện thoại như là có thể gọi điện và nhắn tin cho người dùng khi phát hiện có người lạ đột nhập thông qua các cảm biến và Modul Sim 900A.

## **1.3 Định hướng giải pháp**

Thiết kế, lập trình một WebServer cho Modul Wifi Node MCU ESP8266 có một số chức năng đơn giản như thiết lập wifi, số điện thoại gọi đến, nội dung tin nhắn, sử dụng ngôn ngữ JavaScript đơn giản và Frontend sử dụng HTML, CSS đơn giản.

Lắp ráp mô hình, mạch điện bằng các linh kiện điện tử dễ tìm kiếm và phổ biến.

Giúp phần cải thiện khả năng giao tiếp với phần cứng cũng như có thêm kiến thức về lập trình cho một số loại vi điều khiển. Khả năng tìm kiếm và tổng hợp thông tin về các giao thức được sử dụng trong IoT, và áp dụng nó vào mô hình trên.

# **CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU**

## **2.1 Khảo sát hiện trạng**

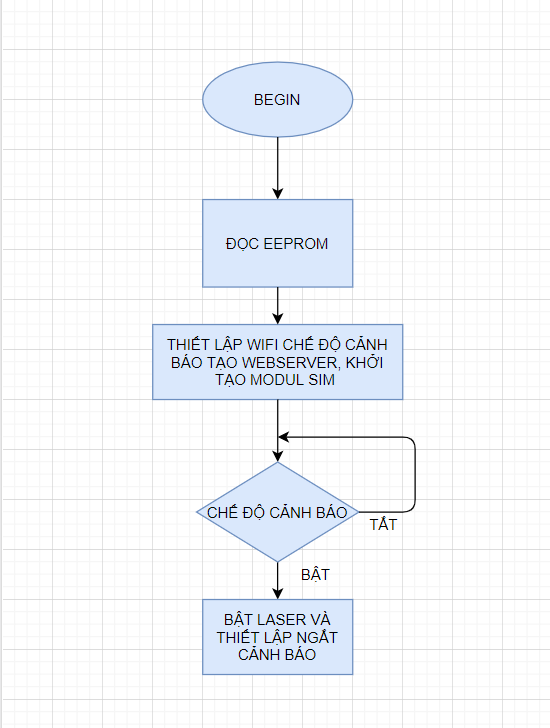
Tiềm năng của hệ thống cảnh báo đột nhập qua điện thoại trong nhà thông minh vô cùng lớn, với sự phát triển công nghiệp hóa, hiện đại hóa như ngày nay thì lượng khách hàng có nhu cầu lắp đặt và sử dụng nhà thông minh ngày càng cao. Theo Statista, doanh thu Smarthome tại Việt Nam cán mốc 105.5 triệu USD cho đến tháng 9/2019. Dự đoán có thể đạt mức 163 triệu USD năm 2020. Đồng thời, Việt Nam sẽ tăng hạng lên thứ 34 trong bảng xếp hạng Smarthome trên toàn cầu. Mức tăng trưởng tốt hơn rất nhiều nước trong khu vực, trong đó có Thái Lan. Từ những con số trên, các chuyên gia nhận định, Việt Nam là một thị trường tiềm năng phát triển rất lớn.

## **2.2 Tổng quan chức năng**

* Chức năng bật hoặc tắc chế độ cảnh báo.
* Thiết lập wifi
* Thiết lập được tên và mật khẩu của wifi do ESP8266 ở chế độ access point phát ra
* Wifi được thiết lập ở chế ẩn hoặc hiển thị. Khi wifi ở chế độ ẩn sẽ có độ bảo mật cao hơn vì khi đó muốn kết nối được đến webserver thì sẽ phải kết nối vào mạng do ESP8266 phát ra trước nên sẽ không dễ dàng vào được webserver để thay đổi cấu hình khi mà không biết tên và mật khẩu của wifi.
* Thiết lập cho Module SIM 900A
* Thiết lập số điện thoại và nội dung tin nhắn khi có đột nhập thì module SIM 900A sẽ gọi về hoặc gửi tin nhắn về cho số điện thoại mà ta đã cấu hình.
* Có chức năng testcall, khi thực hiện nó sẽ gọi về cho số điện thoại mà ta đã cấu hình (không quan trọng có đột nhập).
* Sử dụng điên thoại đã được cấu hình
* Gọi điện thoại hoặc gửi tin nhắn “OFF CB” đến Module SIM 900A để tắt cảnh báo.
* Gửi Tin nhắn “ON CB0” và “ON CB1” để bật chế độ cảnh báo Call Module hoặc Send SMS

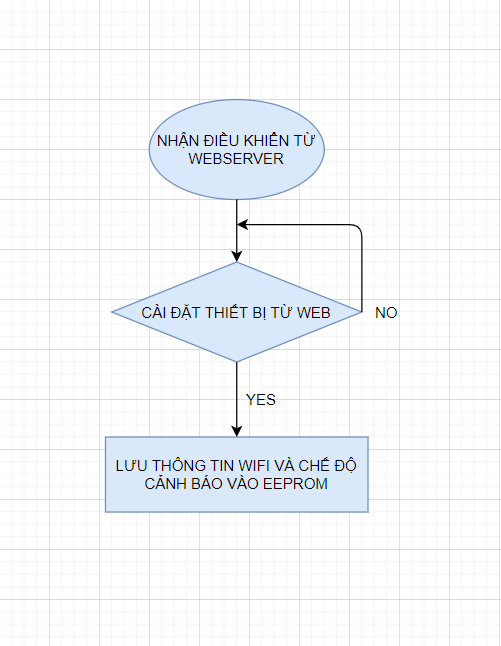
## **2.3 Cách thức hoạt động**

* Sơ đồ hoạt động của từng phần:



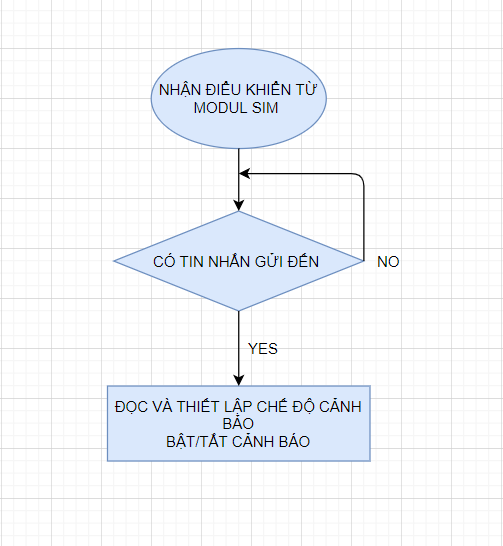
Hình 1: Sơ đồ hoạt động của từng phần:

* Webserver thiết lập wifi, số điện thoại và nội dung tin nhắn, chế độ hoạt động modul sim, chế độ bật tắt cảnh báo,…



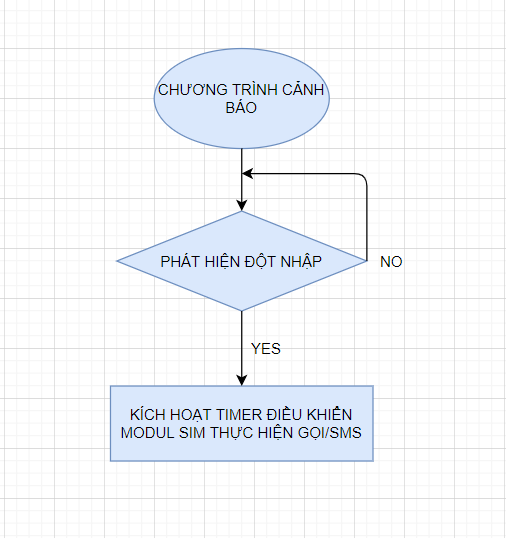
Hình 2: Thiết lập Webserver

* Gửi tin nhắn cho Modul Sim 900A để tắt hoặc bật cảnh báo:



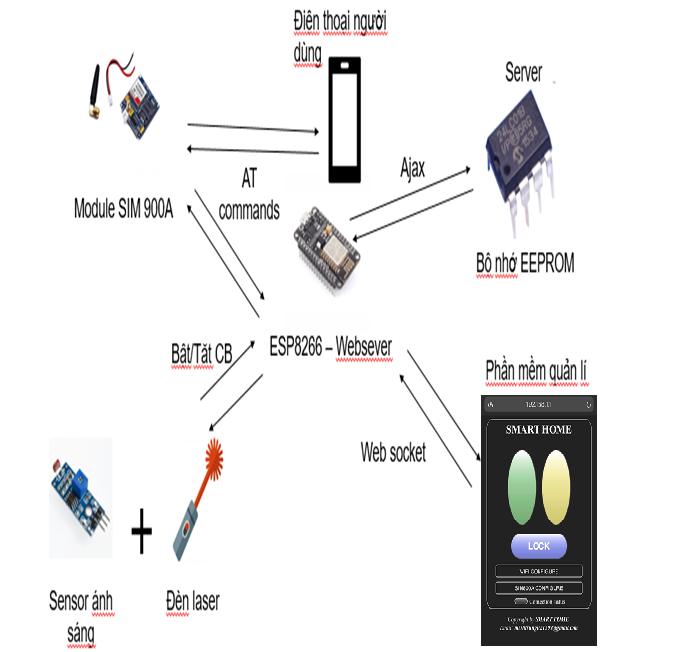
Hình 3: Gửi tin nhắn modul sim

* Chương trình cảnh báo khi có đột nhâp:



Hình 4: Chương trình cảnh báo

## **2.4 Sơ đồ tổng quan hệ thống**



Hình 5: Sơ đồ tổng quan

* Đầu tiên vi xử lý ESP8266 phát ra một mạng có tên và mật khẩu wifi do ta cấu hình từ trước.
* Sau đó ta dùng mội thiết bị khác kết nối vào mạng và vào webserver với địa chỉ ip là 192.168.1.1 đề cấu hình lại wifi nếu muốn sau đó cấu hình tiếp số điện thoại, nội dung tin nhắn để khi phát hiện đột nhập thì Module Sim 900A sẽ gọi báo động về cho số điện thoại cài đặt.
* Khi đã cấu hình xong thì mọi dữ liệu sẽ được lưu trữ vào bộ nhớ EEPROM, nếu có đột nhập thì sẽ gọi điện cho chúng ta mà chúng ta muốn tắt chế độ cảnh báo mà không muốn vào webserver nữa thì ta có thể gọi lại cho số của Module Sim 900A hoặc gửi tin nhắn với cú pháp “OFF CB”.

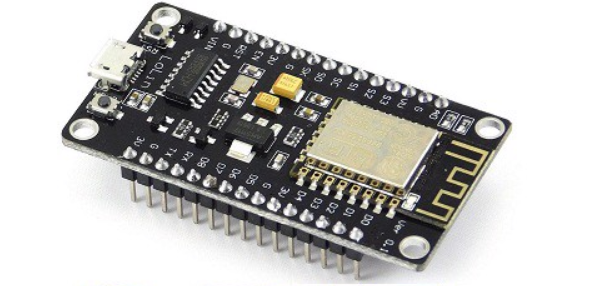
# **CHƯƠNG 3. CÔNG NGHỆ**

## **3.1 Phần cứng**

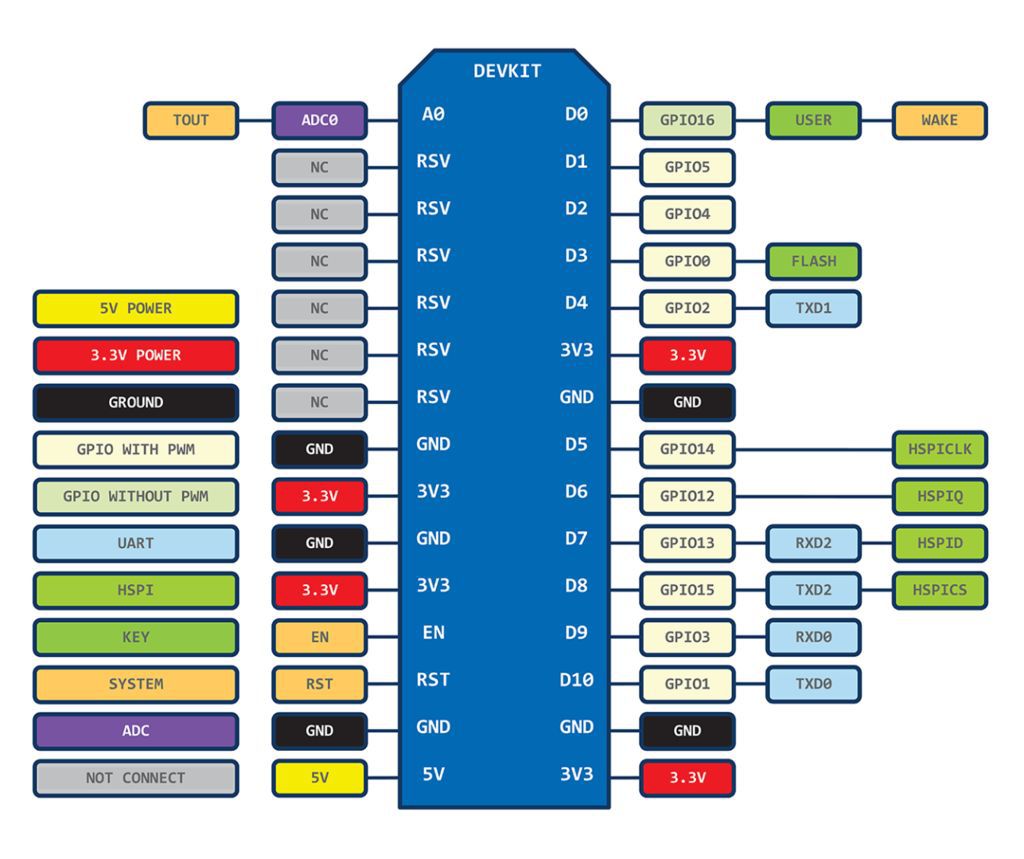
### **3.1.1 ESP8266 V3 CH340 NODEMCU LUA WIFI**

Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 là phiên bản NodeMCU sử dụng IC nạp 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 RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT*.*



Hình 6: ESP8266 V3 CH340 NODEMCU LUA WIFI



Hình 7: Sơ dồ mạch ESP8266 V3 CH340 NODEMCU LUA WIFI

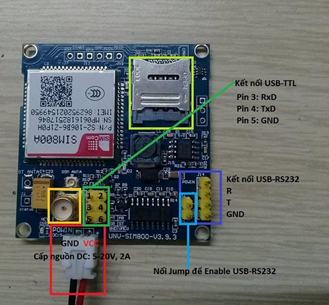
**Thông số kỹ thuật:**

* IC chính: ESP8266 Wifi SoC.
* Phiên bản firmware: NodeMCU Lua
* Chip nạp và giao tiếp UART: CH340
* GPIO tương thích hoàn toàn với firmware Node MCU.
* Cấp nguồn: 5VDC MicroUSB hoặc Vin.
* GIPO 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: 59 x 32mm

### **3.1.2 Modul Sim 900A**

[Module SIM 900A GSM GPRS Mini](https://www.thegioiic.com/products/module-sim800a-gsm-gprs-mini) là phiên bản Module cải thiện về độ bền, độ ổn định và vẫn có các tính năng cơ bản của một chiếc điện thoại như gọi điện thoại, nhắn tin, truy cập GPRS…

Modul tích hợp nguồn xung và ic đệm được thiết kế nhỏ gọn nhưng vẫn giữ được các yếu tố cần thiết của thiết kế như: mạch chuyển mức tín hiệu logic sử dụng Mosfet, IC giao tiếp RS232 MAX232, mạch nguồn xung dòng cao, khe sim chuẩn và các đèn led báo hiệu



Hình 8: Modul Sim 900A

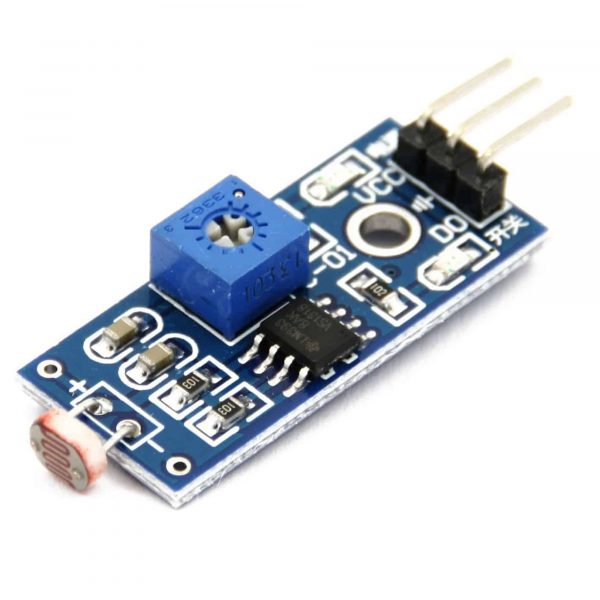
**Thông số kỹ thuật:**

* Sử dụng module GSM GPRS Sim 900A.
* Nguồn cấp đầu vào: 5 – 18VDC, lớn hơn 1A.
* Mức tín hiệu giao tiếp: TTL (3.3-5VDC) hoặc RS232.
* Tích hợp chuyển mức tín hiệu TTL Mosfet tốc độ cao.
* Tích hợp IC chuyển mức tín hiệu RS232 MAX232.
* Tích hợp nguồn xung với dòng cao cung cấp cho Sim900A.
* Sử dụng khe Micro Sim.
* Thiết kế mạch nhỏ gọn, bền bỉ, chống nhiễu
* Kết nối theo kiểu USB-TTL:
* Pin3 là chân RxD kết nối với chân TxD của MCU/USB-TTL
* Pin4 là chân TxD kết nối với chân RxD của MCU/USB-TTL
* GND kết nối với chân GND của MCU/USB-TTL
* Pin1 ban đầu JUMP với Pin3 khi mua về
* Pin2 ban đầu JUMP với Pin4 khi mua về
* Pin6 là chân cấp nguồn cho MCU
* Kết nối theo kiểu USB-RS232
* JUMP 2 pin như mô tả trên hình để Enable.
* Kết nối với USB-RS232 với module sim qua 3 chân R-T-GND như trên hình.

### **3.1.3 Cảm biến ánh sáng**

Cảm biến ánh sáng quang trở phát hiện cường độ ánh sáng, sử dụng bộ cảm biến photoresistor loại nhạy cảm, cho tín hiệu ổn định, rõ ràng và chính xác hơn so với quang trở.

Ngõ ra D0 trên cảm biến được dùng để xác định cường độ sáng của môi trường, khi ở ngoài sáng, ngõ ra D0 là giá trị 0, khi ở trong tối, ngõ ra D0 là 1. Trên cảm biến có 1 biến trở để điều chỉnh cường độ sáng phát hiện, khi văn cùng chiều kim đồng hồ thì sẽ làm giảm cường độ sáng nhận biết của cảm biến, tức là môi trường phải ít sáng hơn nữa thì cảm biến mới đọc gía trị digital là 1.



Hình 9: Cảm biến ánh sáng

**Thông số kỹ thuật:**

* Điện áp làm việc: 3.3 ~ 5VDC
* Output: Digital
* Có thể điều chỉnh cường độ ánh sáng phát hiện bằng biến trở gắn trên cảm biến
* Kích thước: 3.2cm x 1.4cm

### **3.1.4 Đầu phát laser**

Đầu phát laze là bộ sản phẩm có rất nhiều ứng dụng khác nhau: Đồ chơi Laser, ứng dụng chống trộm, đèn tín hiệu,...



Hình 10: Đầu phát laser

**Thông số kỹ thuật**

* Sử dụng điện áp: 3VDC hoặc 5VDC
* Truyền tải điện: 50mW
* Kích thước: Φ6mm
* Độ bền: > 1000h
* Kieur dọi: Điểm
* Bước sóng: 650nm
* Công suất tiêu thụ: 5mW
* Dòng điện: <20mA
* Nhiệt độ làm việc: -36 ℃ ~ 65 ℃
* Độ dọi xa: Khoảng cách 15m ( Điểm dọi Φ10mm)

### **3.1.5 Nguồn Power Adaptor AC-DC 5V 2A**

Nguồn Power Adaptor AC-DC 5V 2A được sử dụng để cấp nguồn cho các thiết bị sử dụng điện áp 5VDC, nguồn có thiết kế nhỏ gọn, linh kiện gia công chất lượng tốt, dây điện có lõi đồng dày, độ bền cao, dòng đầu theo thông số nhà sản xuất lên đến 2A



Hình 11: Nguồn Power Adaptor AC-DC 5V 2A

**Thông số kỹ thuật:**

* Điện áp ngõ vào:100~240VAC, 50/60Hz.
* Điện áp ngõ ra: 5VDC
* Dòng điện ngõ ra tối đa: 2A (nếu sử dụng liên tục nên cung cấp ở mức 80% công suất).
* Kiểu nguồn: nguồn xung.
* Tích hợp led hiển thị trạng thái nguồn.
* Kiểu giắc ngõ ra: Chuẩn Jack DC tròn 5.5\*2.1~2.5mm và 4.0\*1.7mm
* Chiều dài dây dẫn: 1m

## **3.2 Hướng dẫn sử dụng Module Sim 900A**

### **3.2.1 Giao tiếp Module Sim 900A với Esp8266 để thực hiện chức năng call/sms**

* Đấu dây ST của Module Sim 900A với dây D1 của Esp8266
* Đấu dây SR của Module Sim 900A với dây D2 của Esp8266
* Đấu dây G của Module Sim 900A với dây GND của Esp8266
* Cấp nguồn 5V cho Modul Sim 900A

### **3.2.2 Các lênh AT cơ bản**

* Các lệnh chung
* **Lệnh:**      **AT**<CR><LF>

**Mô tả :** Kiểm tra đáp ứng của Module Sim 900A, nếu trả về OK thì Module hoạt động

* **Lệnh:**      **ATE[x]**<CR><LF>

**Mô tả :** Chế độ echo là chế độ phản hồi dữ liệu truyền đến của module Sim 900A,

            x = 1 bật chế độ echo , x = 0 tắt chế độ echo (bạn nên tắt chế độ này khi giao tiếp với vi điều khiển)

* **Lệnh:**      **AT+IPR=[baud rate]**<CR><LF>

**Mô tả :** cài đặt tốc độ giao tiếp dữ liệu với Module Sim 900A, chỉ cài được các tốc độ sau

**baud rate :**0  (auto), 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

* **Lệnh:**      **AT&W**<CR><LF>

**Mô tả :** lưu lại các lệnh đã cài đặt

* Các lệnh điều khiển cuộc gọi
* **Lệnh:**      **AT+CLIP=1**<CR><LF>

**Mô tả :** Hiển thị thông tin cuộc gọi đến

* **Lệnh:**      **ATD[Số\_điện\_thoại];**<CR><LF>

**Mô tả :** Lệnh thực hiện cuộc gọi

* **Lệnh:**      **ATH**<CR><LF>

**Mô tả :** Lệnh thực hiện kết thúc cuộc gọi, hoặc cúp máy khi có cuộc gọi đến

* **Lệnh:**      **ATA**<CR><LF>

**Mô tả :** Lệnh thực hiện chấp nhận khi có cuộc gọi đến

* Các lệnh điều khiển tin nhắn
* **Lệnh:**      **AT+CMGF=1**<CR><LF>

**Mô tả :** Lệnh đưa SMS về chế độ Text , phải có lệnh này mới gửi nhận tin nhắn dạng Text

* **Lệnh:      AT+CMGS=”Số\_điện \_thoại”<**CR**><**LF**>**

      Đợi đến khi có ký tự ‘**>**’ được gửi về thì đánh nối dung tin nhắn. Gửi mã Ctrl+Z  hay  **0x1A** để kết thúc nội dung và gửi tin nhắn

**Mô tả :** Lệnh gửi tin nhắn

* **Lệnh:**      **AT+CMGR=x**<CR><LF>

          x là địa chỉ tin nhắn cần đọc

**Mô tả :** Đọc một nhắn vừa gửi đến, lệnh được trả về nội dung tin nhắn, thông tin người gửi, thời gian gửi

* **Lệnh:**      **AT+CMGDA="DEL ALL"**<CR><LF>

**Mô tả :** Xóa toàn bộ tin nhắn trong các hộp thư

* **Lệnh:**      **AT+CNMI**=2,2<CR><LF>

**Mô tả :**Hiển thị nội dung tin nhắn ngay khi có tin nhắn đến

### **3.2.3 Các lệnh và hàm sử dụng trong Project**

* Hàm cấu hình cho Module Sim 900A sau đó lưu những thông tin đã có vào bộ nhớ EEPROM

function configuresim900a(){

document.getElementById("homecontrol").style.display = "none";

document.getElementById("wifisetupap").style.display = "none";

document.getElementById("sim900asetup").style.display = "block";

document.getElementById("sodt\_send").value = sodtsend;

document.getElementById("sms\_send").value = smssend;

var cdhd\_sim900a = document.getElementsByName("cdhdsim900a");

for (var i = 0, length = cdhd\_sim900a.length; i < length; i++) {

if (cdhd\_sim900a[i].value == cdhdsim) {

document.getElementsByName("cdhdsim900a")[i].checked = "true";

break;

}

}

}

function writeEEPROMsim900a(){

if(Empty(document.getElementById("sodt\_send"), "Please enter mobile number!")&&Empty(document.getElementById("sms\_send"), "Please enter sms content")){

var sodtgui = document.getElementById("sodt\_send").value;

var smsgui = document.getElementById("sms\_send").value;

var cdhd\_sim900a = document.getElementsByName("cdhdsim900a");

for (var i = 0, length = cdhd\_sim900a.length; i < length; i++) {

if (cdhd\_sim900a[i].checked) {

cdhd\_sim900a=cdhd\_sim900a[i].value;

break;

}

}

xhttp.open("GET","/writeEEPROMsim900a?sodtsend="+sodtgui+"&smssend="+smsgui+"&cdhdsim900a="+cdhd\_sim900a,true);

xhttp.onreadystatechange = process;//nhận reponse

xhttp.send();

}

}

* Hàm dùng để setup cho Module Sim 900A

void setupSIM900A(){

SIM900.println("ATE0"); //Tắt chế độ echo khi gửi lệnh đi

delay(1000);

SIM900.println("AT+IPR=9600"); //Cài tốc độ baud 9600

delay(1000);

SIM900.println("AT+CMGF=1"); //Hiển thị tin nhắn ở chế độ txt

delay(1000);

SIM900.println("AT+CLIP=1"); //Hiển thị số điện thoại gọi đến

delay(1000);

SIM900.println("AT+CNMI=2,2"); //Hiển thị trực tiếp nội dung tin nhắn gửi đến

delay(1000);

}

* Hàm dùng để Module gọi về số điện thoại

void SIM900Call(String sdt){

SIM900.println("ATD" + sdt + ";");

delay(1000);

Serial.print("Đang thực hiện cuộc gọi đến số: ");

Serial.println(sdt);

}

* Hàm dùng để Module Sim 900A thực hiện tin nhắn đến số điện thoại

void SIM900SMS(String phone, String content){

SIM900.println("AT+CMGS=\"" + phone + "\""); // Lenh gui tin nhan

delay(3000); // Cho ky tu '>' phan hoi ve

SIM900.print(content); // Gui noi dung

SIM900.print((char)26); // Gui Ctrl+Z hay 26 de ket thuc noi dung tin nhan va gui tin di

delay(5000); // delay 5s

Serial.print("Đã gủi tin nhắn đến số: ");

Serial.println(phone);

Serial.print("Nội dung tin nhắn: ");

Serial.println(content);

}

## **3.3 Hướng dẫn sử dụng ESP8266**

Trong project này thì ESP8266 vừa là 1 điểm Access Point vừa là 1 Webserver có địa chỉ ip là 192.168.1.1.

### **3.3.1 Sử dụng ESP8266 là 1 điểm Access Point**

* Về Access Point

Chế độ soft access point (soft-AP) được dùng để hỗ trợ việc kết nối vào mạng Wifi. Khi đó, ta thiết lập một mạng WiFi mới với SSID và password do ta thiết lập. Sau khi thiết lập Wifi ở chế độ soft Access point, thì các thiết bị có thể kết nối tới mạng WiFi đó. Khi đó, ta có thể quản lí các kết nối với các trạm kết nối (station).

Chế độ soft-AP có thể được dùng để thiết lập mesh network (một mạng lưới). Mesh network là những kết nối mạng theo lưới với nhiều điểm truy nhập (access points) khác nhau trong những khu vực địa lý rộng lớn.

Một hệ thống mesh network bao gồm các client, router và gateway. Do đó, để module ESP8266 tham gia vào một mesh network lớn thì hệ thống cần có các router. Mesh network truyền thông tin giữa các nút. Do ESP8266 có thể làm việc ở 2 chế độ station và soft-AP nên nó có thể là một nút trong mesh network.

* softAP

Cách thiết lập đơn giản nhất chỉ yêu cầu một tham số và được sử dụng để thiết lập một mạng Wi-Fi mở.

WiFi.softAP (ssid)

Để thiết lập mạng được bảo vệ bằng mật khẩu, hoặc để cấu hình các thông số mạng bổ sung, sử dụng quá tải sau đây:

WiFi.softAP(ssid, password, channel, hidden)

Tham số đầu tiên của hàm này là bắt buộc, còn lại ba tùy chọn.

* + - ssid: chuỗi ký tự chứa SSID mạng (tối đa 63 ký tự)
    - password: chuỗi ký tự tùy chọn với mật khẩu. Đối với mạng WPA2-PSK, nó phải có ít nhất 8 ký tự. Nếu không có mật khẩu, thì đây sẽ là mạng WiFi mở.
    - channel: Tham số tùy chọn để thiết lập kênh Wi-Fi, từ 1 đến 13. Kênh mặc định = 1.
    - hidden: Tham số tùy chọn, thiết lập là true để ẩn SSID
    - Trả về true hoặc false phụ thuộc vào kết quả của việc cài đặt soft-AP.
* softAPConfig

softAPConfig(local\_ip, gateway, subnet)

Tất cả các thông số đều có kiểu IPAddress và được định nghĩa như sau:

* + local\_ip: Địa chỉ IP của điểm truy cập mềm
  + gateway: địa chỉ IP gateway
  + subnet: subnet mask

Trả về true hoặc false phụ thuộc vào kết quả của việc thay đổi cấu hình.

* softAPgetStationNum

Lấy số lượng các station kết nối đến softAP

WiFi.softAPgetStationNum()

Serial.printf("Stations connected to soft-AP = %d\n", WiFi.softAPgetStationNum());

* softAPdisconnect

Ngắt kết nối các trạm từ mạng được thiết lập bởi softAP.

WiFi.softAPdisconnect(wifioff)

Chức năng sẽ thiết lập cấu hình SSID và password của soft-AP giá trị là null. Tham số wifioff là tùy chọn. Nếu thiết lập là true nó sẽ tắt chế độ soft-AP.

Trả về true nếu hoạt động đã thành công, false nếu không.

* softAPIP

Trả lại địa chỉ IP của mạng softAP.

WiFi.softAPIP()

Trả về giá trị có kiểu là IPAddress.

* softAPmacAddress

Trả lại địa chỉ MAC của softAP. Chức năng này có hai phiên bản khác nhau về kiểu trả về. Trả về một con trỏ hoặc một String.

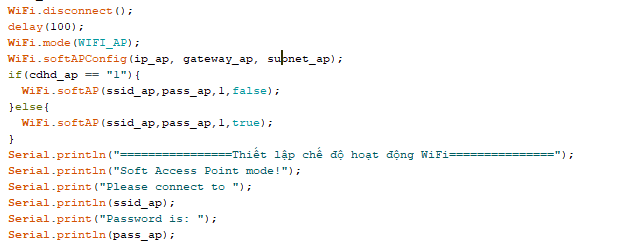
Với kiểu trả về là Con trỏ

WiFi.softAPmacAddress(mac)

Tham số mac là một con trỏ trỏ đến vị trí bộ nhớ (một mảng uint8\_t có 6 phẩn tử) để lưu địa chỉ mac. Cùng một giá trị con trỏ được trả về bởi chính hàm đó.

* Mã nguồn của Project để làm Access Point là :



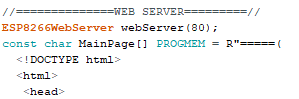


### **3.3.2 Sử dụng ESP8266 là Webserver**

* Thiết kế Webserver với mã nguồn sau:

<https://github.com/minhtrungle/Project-KTMT>

* Các câu lệnh quan trọng để khởi tạo Webserver là:







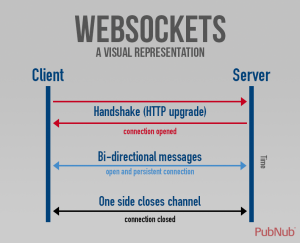
## **3.4 ESP8266 làm WebSocket Server**

### **3.4.1 WebSocket là gì?**

Websocket là giao thức chuẩn cho trao đổi dữ liệu hai chiều giữa client và server. Giao thức WebSocket không chạy trên HTTP, thay vào đó nó thực hiện trên giao thức TCP.

Ưu điểm của Websocket so với HTTP đó là nhẹ hơn rất nhiều, đỗ trễ thấp. Chính vì thế Websocket phù hợp cho những ứng dụng yêu cầu real time (thời gian thực) và ít tốn tài nguyên hệ thống.

Đường dẫn của WebSocket bắt đầu bằng **ws://** hoặc **wss://** (cho kết nối được mã hóa).



Hình 12: WebSocket

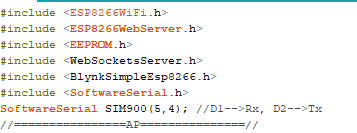
Máy khách (Client) thiết lập kết nối WebSocket thông qua một quá trình được gọi là handshake (bắt tay). Quá trình này bắt đầu bằng việc client gửi một yêu cầu nâng cấp HTTP đến máy chủ, thông báo cho máy chủ rằng máy khách muốn thiết lập kết nối WebSocket.

Sau khi handshake thành công, kết nối HTTP ban đầu được thay thế bằng kết nối WebSocket. Tại thời điểm này, một trong hai bên có thể bắt đầu gửi dữ liệu.

Việc kết thúc kết nối WebSocket có thể thực hiện từ 1 trong 2 phía, thông thường ở phía máy khách.

### **3.4.2 ESP8266 làm WebSocket Server**

* Khai báo các thư viện cần thiết:



* Để sử dụng thư viện **WebSocketsServer** ta cần khởi tạo một đối tượng tên là là **webSocket** chạy trên port 81.

WebSocketsServer webSocket = WebSocketsServer(81);

* Khai báo hàm dùng để khởi tạo và xử lý các sự kiện của WebSocket

**void** startWebSocket() {

webSocket.begin();

webSocket.onEvent(webSocketEvent);

}

Server có thể gởi message đến một client cụ thể:

webSocket.sendTXT(client\_id,message);

hoặc toàn bộ client đang kết nối:

webSocket.broadcastTXT("Hi All!");

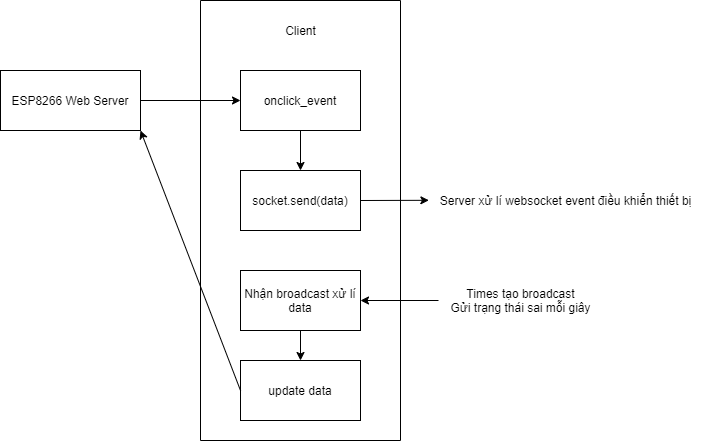
* Cuối cùng chúng ta cần gọi phương thức **webSocket.loop()** trong vòng lặp chính để kiểm tra khi có sự kiện WebSocket xảy ra.

**void** loop() {

webSocket.loop();

}

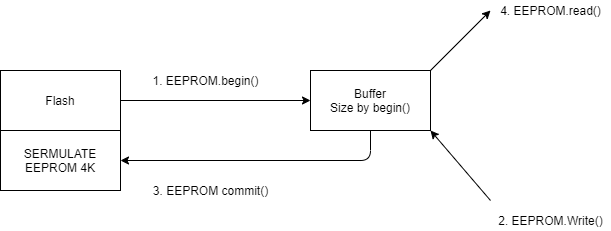
### **3.4.3 Mô hình của WebSocket**



Hình 13: Mô hình của WebSocket

## **3.5 Bộ nhớ EEPROM trong ESP8266**

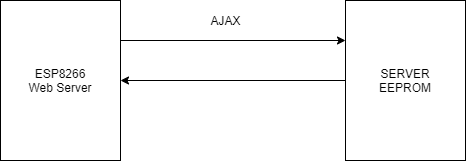
### **3.5.1 Sử dụng bộ nhớ EEPROM trong ESP8266**



Hình 14: Sử dụng bộ nhớ EEPROM trong ESP8266

Đầu tiên ta sẽ dùng câu lệnh EEPROM.begin(size); để khởi tạo vùng nhớ trong EEPROM tiếp đó ta sẽ dùng câu lệnh EEPROM.write(); để ghi dữ liệu vào và dùng câu lệnh EEPROM.commit(); để hoàn tất việc ghi dữ liệu cuối cùng để đọc bộ nhớ EEPROM ra thì ta chỉ cần câu lệnh EEPROM.read()

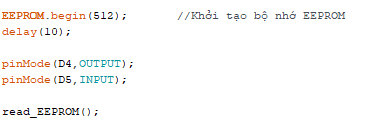
### **3.5.2 Lưu trạng thái vào EEPROM**

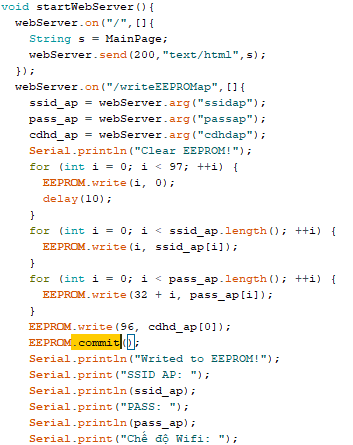


Hình 15: Lưu trạng thái vào EEPROM

* SERVER

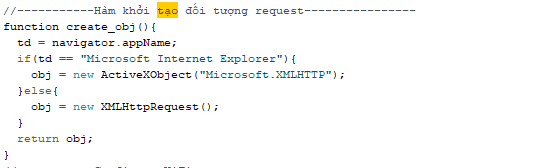


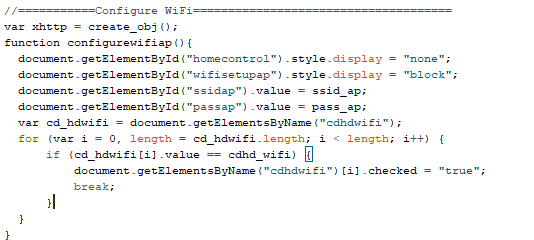


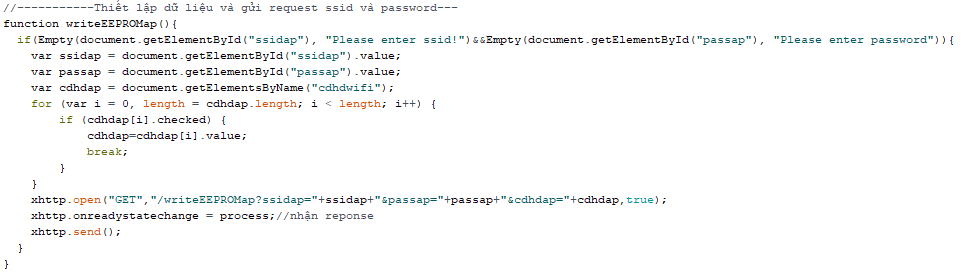


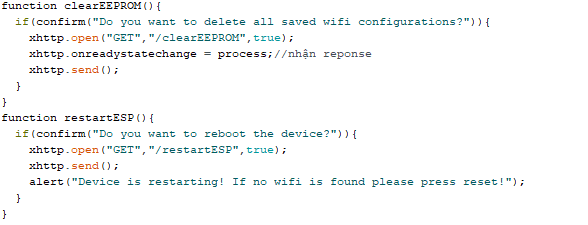
* AJAX

Dưới đấy là một số hàm sử dụng ajax trong project





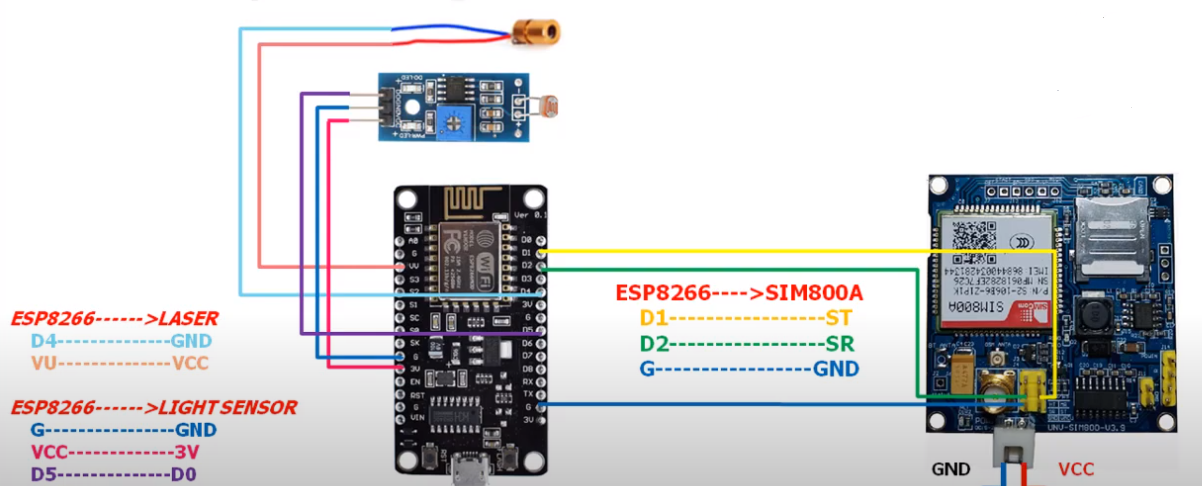




# **CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ HỆ THỐNG**

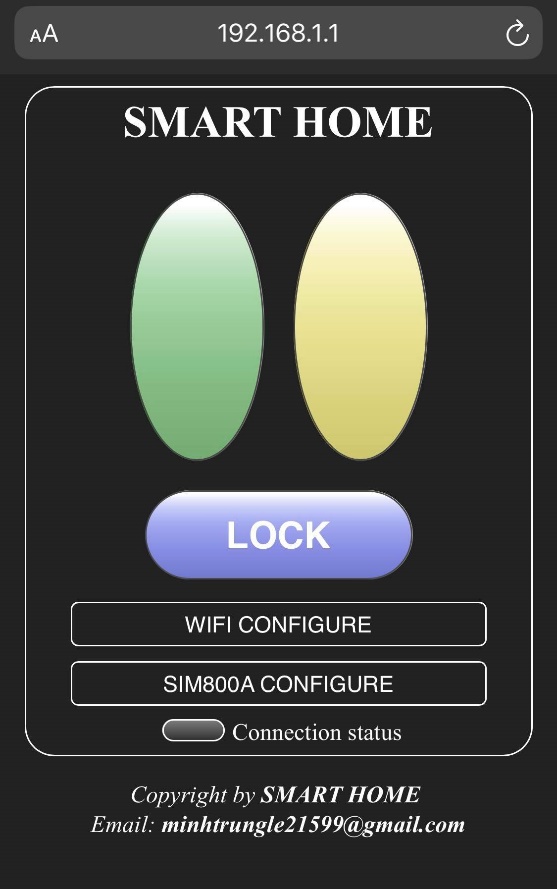
## **4.1 Kết quả**

* Mạch của hệ thống được lắp như sau:

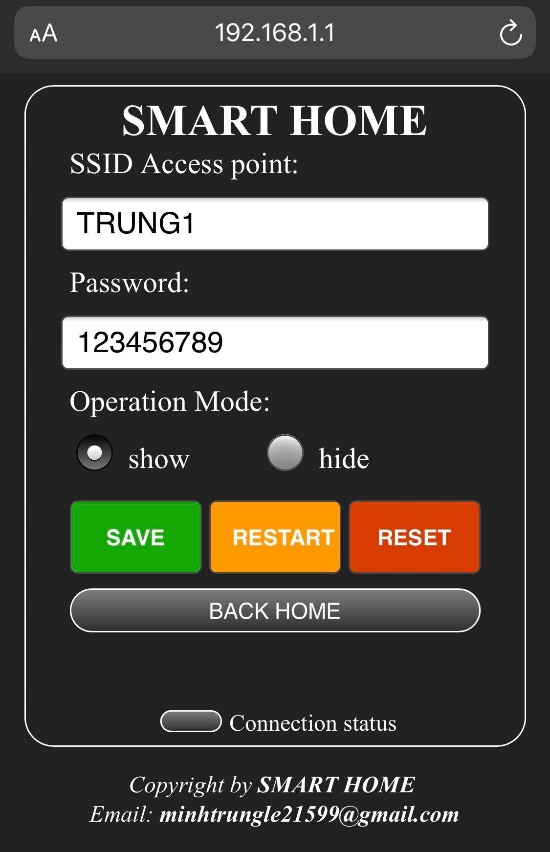


Hình 16: Lắp đặt mạch hệ thống

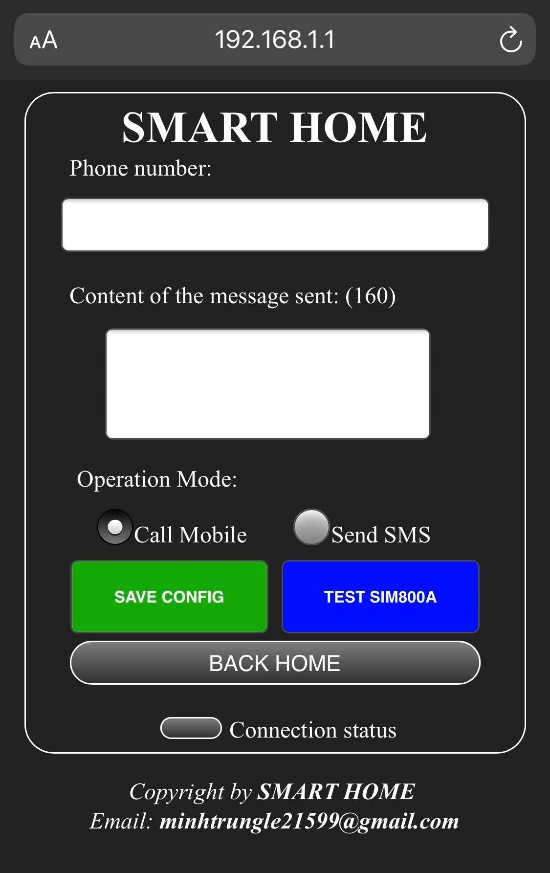
* Giao diện của Webserver:



Hình 17: Giao diện màn hình chính Webserver



Hình 18: Giao diện thiết lập Wifi



Hình 19: Giao diện thêm số điện thoại gửi tin nhắn

## **4.2 Đánh giá hệ thống**

* Ưu điểm :
* Đã đáp ứng nhu cầu của người dùng là cảnh bá đột nhập
* Hệ thống đã có bảo mật bằng wifi để chứng minh rằng không phải ai cũng vào được webserver.
* Hệ thống đã thực hiện bật tắt chế độ cảnh báo ở nhiều dạng khác nhau không cần phải trực tiếp vào webserver
* Do còn kiến thức còn hạn chế, nên đề tài không tránh khỏi những thiếu sót và những vấn đề chưa khắc phục được. Có thể nêu ra một số điểm như sau:
* Tính năng còn rất cơ bản, cần bổ sung thêm nhiều tính năng mới để ngôi nhà được thông minh hơn.
* Webserver vẫn cần kết nối wifi để vào được nếu ở xa thì không thể kết nối được.
* Bộ điều khiển nhận tín hiệu từ cảm biến bằng dây nên để sử dụng được ngoài thực tế cần phải tính toán lại điện trở dây và áp dụng các biện pháp chống nhiễu cho cảm biến.

# **CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **5.1 Kết luận**

Qua quá trình tìm hiểu và thực hiện thiết kế hệ thống cảnh báo đột nhập qua điện thoại, so với các sản phẩm đã đi vào thực tiễn của các công ty khác thì còn nhiều mặt hạn chế về thiết bị cũng như việc tối ưu. Tuy nhiên cũng đã có những kết quả, hệ thống đã có những chức năng cơ bản, Trong quá trình thực hiện đã học hỏi thêm được nhiều kiến thức về lĩnh vực IoT, được tiếp xúc nhiều với phần cứng và thực hiện kết nối phần cứng với phần mềm, góp phần củng cố thêm kiến thức về chuyên ngành, hoàn thiện hơn những kiến thức lý thuyết trong những môn học trên lớp.

## **5.2 Hướng phát triển**

Mô hình về phần cứng có thể thêm hệ thống camera nhận diện, thêm cảm biến nhiệt hoặc cảm biến sóng âm,…

# **TÀI LIỆU THAM KHẢO**

* Tài liệu về ESP8266

[esp8266-technical\_reference\_en.pdf (espressif.com)](https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf)

<https://getblocky.com/courses/p2-ket-noi-internet/lessons/che-do-wifi-access-point/>

* Tài liệu về Module Sim900A

<https://mlab.vn/index.php?_route_=huong-dan-su-dung-module-sim900.html>

* Tài liệu về Websocket

<https://getblocky.com/courses/p2-ket-noi-internet/lessons/websocket-la-gi/>