

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM**

**KHOA ĐIỆN – ĐIỆN TỬ**

---☒☒☒---



**ĐỒ ÁN 1**

**ĐỀ TÀI: THIẾT KẾ ĐỒNG HỒ THÔNG MINH  
ĐO NHỊP TIM VÀ OXY TRONG MÁU**

**GVHD :**

**TS. Đỗ Duy Tân**

**SVTH :**

**Trần Đăng Quang**

**17141220**

**Lê Thành Nhân**

**17141211**

**Thành phố Hồ Chí Minh, ngày 20 tháng 7 năm 2020**

*Thành phố Hồ Chí Minh, ngày 20 tháng 7 năm 2020*

**LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN MÔN HỌC**

Họ tên sinh viên: Trần Đăng Quang

MSSV: 17141220

Lê Thành Nhân

: 17141211

Lớp: 1714VTB

ĐAMH: Đồ án 1

Tên đề tài: **Thiết kế đồng hồ thông minh đo nhịp tim và oxy trong máu**

<b><i>Tuần /ngày</i></b>	<b><i>Nội dung</i></b>	<b><i>Xác nhận GVHD</i></b>
Tuần 1(30/3-5/4)	Gặp GVHD để nghe phổ biến yêu cầu làm đồ án và chọn đề tài đồ án.	
Tuần 2 (6/4-12/4)	Viết lịch trình thực hiện đồ án, đề cương chi tiết đồ án và sơ đồ khối.	
Tuần 3 (13/4-19/4)	- Tìm hiểu module Wifi. -Tìm hiểu về công nghệ truyền không dây của mode MCU ESP8266.	
Tuần 4 (20/4-26/4)	-Tìm hiểu về màn hình OLED, module thời gian thực -Hiển thị thông số, sơ đồ huấn thị trên màn hình OLED	
Tuần 5 (27/4-3/5)	- Tìm hiểu và nghiên cứu hiển thị hình ảnh qua android. - Tìm hiểu và nghiên cứu hiển thị hình ảnh trên web thông qua mạng wifi	

Tuần 6 (4/5-10/5)	-Viết app trên android giao tiếp với hệ thống - Tìm hiểu cách lập trình web và giao tiếp với hệ thống web	
Tuần 7 (11/5-17/5)	- Thiết kế giao diện web. - Kết hợp phương thức đo hiển thị trên cả 2 hướng web và amdroid.	
Tuần 8 (18/5-24/5)	Mô phỏng mạch, kiểm tra và cân chỉnh mạch. Vẽ PCB.	
Tuần 9 - 10 (25/5-7/6)	-Tiến hành thi công mạch -Kiểm tra mạch thi công	
Tuần 11 (8/6-14/6)	Chạy sản phẩm thử nghiệm và thu thập số liệu.	
Tuần 12-13 (15/6-28/6)	Viết báo cáo nội dung đã làm.	
Tuần 14 (29/6-5/6)	Hoàn thiện báo cáo gửi cho GVHD để xem xét và góp ý trước khi in ra.	
Tuần 15 (6/6-12/6)	Làm slide báo cáo và nộp báo cáo và báo cáo đề tài	

# MỤC LỤC

<b>CHƯƠNG 1. TỔNG QUAN</b> .....	1
<b>1.1. Lý do chọn đề tài</b> .....	1
<b>1.2. Mục tiêu đề tài</b> .....	1
<b>1.3. Giới hạn đề tài</b> .....	1
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT</b> .....	2
<b>2.1. Giới thiệu màn hình OLED</b> .....	2
<b>2.1.1. Thông tin kỹ thuật</b> .....	2
<b>2.1.2. Nguyên lý hoạt động của màn hình OLED:</b> .....	2
<b>2.2 Giới thiệu về cảm biến nhịp tim MAX30102</b> .....	3
<b>2.2.1 Thông số kỹ thuật:</b> .....	4
<b>2.2.2. Xác định chỉ số SPO2</b> .....	4
<b>2.2.3. Nguyên lý vật lý được sử dụng để đo SPO2.</b> .....	5
<b>2.3 Tổng quan về ESP8266</b> .....	6
<b>2.3.1. Thông số kỹ thuật:</b> .....	7
<b>2.3.2. Cài đặt NodeMCU ESP8266 WIFI:</b> .....	7
<b>2.3.3. Sơ lược về chuẩn giao tiếp I2C:</b> .....	9
<b>2.4. Giới thiệu về lập trình Web:</b> .....	10
<b>2.4.1. Ngôn ngữ HTML</b> .....	10
<b>2.4.2. Ngôn ngữ CSS</b> .....	10
<b>2.5. Lấy dữ liệu thời gian từ internet</b> .....	10
<b>2.5.1 Giới thiệu giao thức NTP</b> .....	10
<i>a. Giới thiệu về NTP:</i> .....	10
<i>b. NTP server:</i> .....	11
<i>c. Phương thức hoạt động:</i> .....	12
<b>2.5.2 Giới thiệu giao thức UDP</b> .....	12
<i>a. Khái niệm:</i> .....	12

<i>b. Cách hoạt động:</i> .....	13
<b>CHƯƠNG 3. XÂY DỰNG VÀ THIẾT KẾ HỆ THỐNG</b> .....	14
<b>3.1. Sơ đồ khối và chức năng của từng khối</b> .....	14
3.1.1. Sơ đồ khối .....	14
3.1.2. Chức năng .....	14
3.1.3. Hoạt động của hệ thống .....	14
<b>3.2. Thiết kế chi tiết</b> .....	14
3.2.1. Khối nguồn .....	14
3.2.2. Khối xử lý trung tâm .....	16
3.2.3. Khối cảm biến .....	17
<i>a. Sơ đồ chức năng cảm biến MAX30102:</i> .....	17
<i>b. Phương pháp theo dõi độ bão hòa oxy trong máu và nhịp tim:</i> .....	18
3.2.4. Khối hiển thị trên OLED .....	19
<i>a. Giao tiếp MCU I2C:</i> .....	20
<i>b. Sơ đồ kết nối:</i> .....	21
3.2.5. Khối hiển thị Web .....	21
<i>a. Tổng quan về Webserver:</i> .....	21
<i>b. Nguyên lý hoạt động webserver:</i> .....	23
<b>CHƯƠNG 4. THI CÔNG HỆ THỐNG</b> .....	24
4.1. Sơ đồ nguyên lý .....	24
4.2. Giải thích sơ đồ nguyên lý .....	24
4.3. Lưu đồ giải thuật .....	25
4.3.1. Chương trình chính .....	25
4.3.2. Giải thuật đo nhịp tim và spo2: .....	26
4.3.3. Giải thuật dò WiFi: .....	27
4.3.4. Giải thuật thời gian thực: .....	28
<b>CHƯƠNG 5. KẾT QUẢ THỰC HIỆN VÀ KẾT LUẬN</b> .....	29

<b>5.1. Kết quả thực hiện.....</b>	<b>29</b>
<b>5.1.1. Hình ảnh thực tế của sản phẩm: .....</b>	<b>29</b>
<b>5.1.2. Kết quả thực nghiệm: .....</b>	<b>30</b>
<b>5.1.3. Kết luận: .....</b>	<b>32</b>
<b>5.2. Hướng phát triển .....</b>	<b>32</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>..</b>

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Lý do chọn đề tài

Hiện tại vấn đề về Đại dịch Covid – 19 ( đại dịch truyền nhiễm được gây ra bởi virus SARS-CoV-2 ) là vấn đề cấp thiết đang được xã hội quan tâm. Vậy nên để mọi người có thể thuận tiện trong việc theo dõi và kiểm tra sức khỏe, và để ngăn ngừa những sự cố đáng tiếc có thể xảy ra. Nhóm em đã nghĩ đến một thiết bị y tế nhỏ gọn có thể đo được nhịp tim và nồng độ oxy trong máu với hy vọng có thể giúp ích được phần nào cho những người dân hiện tại, đồng thời vừa không tốn quá nhiều thời gian cho việc theo dõi cũng như chi phí khám chữa bệnh.

Chính vì lý do trên mà chúng em đã chọn đề tài “ **Thiết kế đồng hồ thông minh đo nhịp tim và oxy trong máu** ”.

Ý tưởng của đề tài được dựa trên đồ án tốt nghiệp “Thiết kế vòng tay đo nhịp tim sử dụng công nghệ iots” của anh Nguyễn Thanh Hoàn và Nguyễn Hoàng Nam năm 2019.

Sau quá trình thực hiện thì mạch có nhiều cải tiến sau:

- Có khả năng tìm và lựa chọn wifi.
- Thêm tính năng đo nồng độ oxy trong máu (SPO2) và theo dõi thời gian thực.

## 1.2. Mục tiêu đề tài

- Thiết kế và thi công mô hình đồng hồ thông minh đo nhịp tim và nồng độ Oxy trong máu, đảm bảo được tính chính xác, các thông số được cập nhật liên tục. Đồng thời gửi các thông số đo đạc được lên trang Web để nâng cao khả năng giám sát.
- Ứng dụng những kiến thức đã được học vào thực tế, giúp mọi người có thể thường xuyên theo dõi tình trạng sức khỏe thông qua những công nghệ mới hiện đại ngày nay.
- Tăng thêm khả năng làm việc nhóm, khả năng tư duy, tìm tòi, học hỏi.

## 1.3. Giới hạn đề tài

Phạm vi nghiên cứu của đề tài:

- Mô hình chỉ hoạt động trong mạng local
- Tạo web server giao tiếp phần cứng thông qua module ESP8266
- Xây dựng trang web hiển thị các số liệu liên quan về thời gian, nhịp tim, nồng độ oxy trong máu.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Giới thiệu màn hình OLED



Hình 2.1. Màn Hình Oled 1.3 Inch Giao Tiếp I2C.[9]

Màn hình Oled 1.3 inch giao tiếp I2C cho khả năng hiển thị đẹp, sáng trọng, rõ nét vào ban ngày và khả năng tiết kiệm năng lượng tối đa với mức chi phí phù hợp, màn hình sử dụng giao tiếp I2C cho chất lượng đường truyền ổn định và rất dễ giao tiếp chỉ với 2 chân GPIO. [2]

#### 2.1.1. Thông tin kỹ thuật

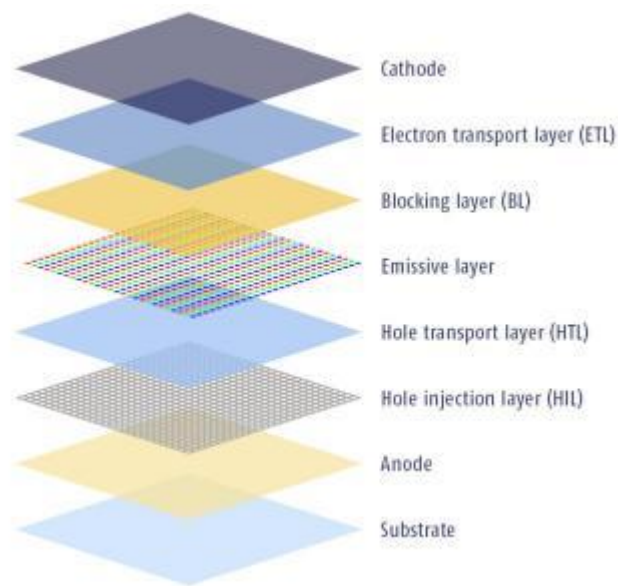
- Điện áp sử dụng: 2.2~5.5VDC
- Công suất tiêu thụ: 0.04w
- Góc hiển thị: lớn hơn 160 độ
- Số điểm hiển thị: 128x64 điểm.
- Độ rộng màn hình: 1.3 inch.
- Màu hiển thị: Trắng / Xanh Dương.
- Giao tiếp: I2C
- Driver: SH1106.[9]

#### 2.1.2. Nguyên lý hoạt động của màn hình OLED:

OLED (Điốt phát sáng hữu cơ) là một công nghệ phát sáng phẳng, được tạo ra bằng cách đặt một loạt màng mỏng hữu cơ giữa hai dây dẫn. Khi có dòng điện đi vào sẽ phát ra ánh sáng. OLED là màn hình phát xạ không cần đèn nền và do đó mỏng hơn và hiệu quả hơn màn hình LCD (vốn cần đèn nền màu trắng).

Thành phần chính trong màn hình OLED là bộ phát OLED - một vật liệu hữu cơ (dựa trên carbon) phát ra ánh sáng khi có điện. Cấu trúc cơ bản của OLED là một lớp phát xạ được kẹp giữa cực âm (bơm electron) và cực dương (loại bỏ electron).





Hình 2.2 Cấu trúc màn hình oled.[10]

Một tấm nền OLED được làm từ chất nền (substrate), bảng nối đa năng (thiết bị điện tử - trình điều khiển), mặt trước (vật liệu hữu cơ và điện cực như đã giải thích ở trên) và một lớp đóng gói. OLED rất nhạy cảm với oxy và độ ẩm và vì vậy lớp đóng gói là rất quan trọng.[10]

#### **Ưu điểm và nhược điểm của OLED**

- Màn hình OLED có nhiều ưu điểm so với màn hình LED và LCD:
- Chúng nhẹ hơn và mỏng hơn màn hình LCD.
- Chúng linh hoạt, cho phép sản xuất màn hình video cong.
- OLED sáng hơn đèn LED truyền thống.
- Chúng có thể được làm mới nhanh hơn nhiều so với LCD, làm cho chúng phù hợp với video tốc độ cao và tốc độ khung hình cao.
- OLED có thể được nhìn thấy trong nhiều môi trường ánh sáng, cả trong nhà và ngoài trời.
- Cũng có một vài nhược điểm đối với màn hình OLED:
- Màn hình OLED sớm có vấn đề về tuổi thọ. Điều này đã được cải thiện, tuy nhiên OLED màu xanh vẫn có tuổi thọ ngắn hơn so với đèn LED tương đương.
- OLED có thể bị hỏng do nước. Điều này loại trừ chúng từ nhiều ứng dụng ngoài trời.
- Giá thành cao.

## **2.2 Giới thiệu về cảm biến nhịp tim MAX30102**

Cảm biến nhịp tim và oxy trong máu MAX30102 được sử dụng để đo nhịp tim và nồng độ Oxy trong máu, thích hợp cho nhiều ứng dụng liên quan đến y sinh, cảm biến sử dụng phương pháp đo quang phổ biến hiện nay với thiết kế và chất liệu mắt đo chuyên biệt từ chính

hãng Maxim cho độ chính xác và độ bền cao, cảm biến sử dụng giao tiếp I2C với bộ thư viện sẵn có trên Arduino rất dễ sử dụng.



Hình 2.2. Cảm Biến Nhịp Tim Và Oxy Trong Máu MAX30102.[9]

### 2.2.1 Thông số kỹ thuật:

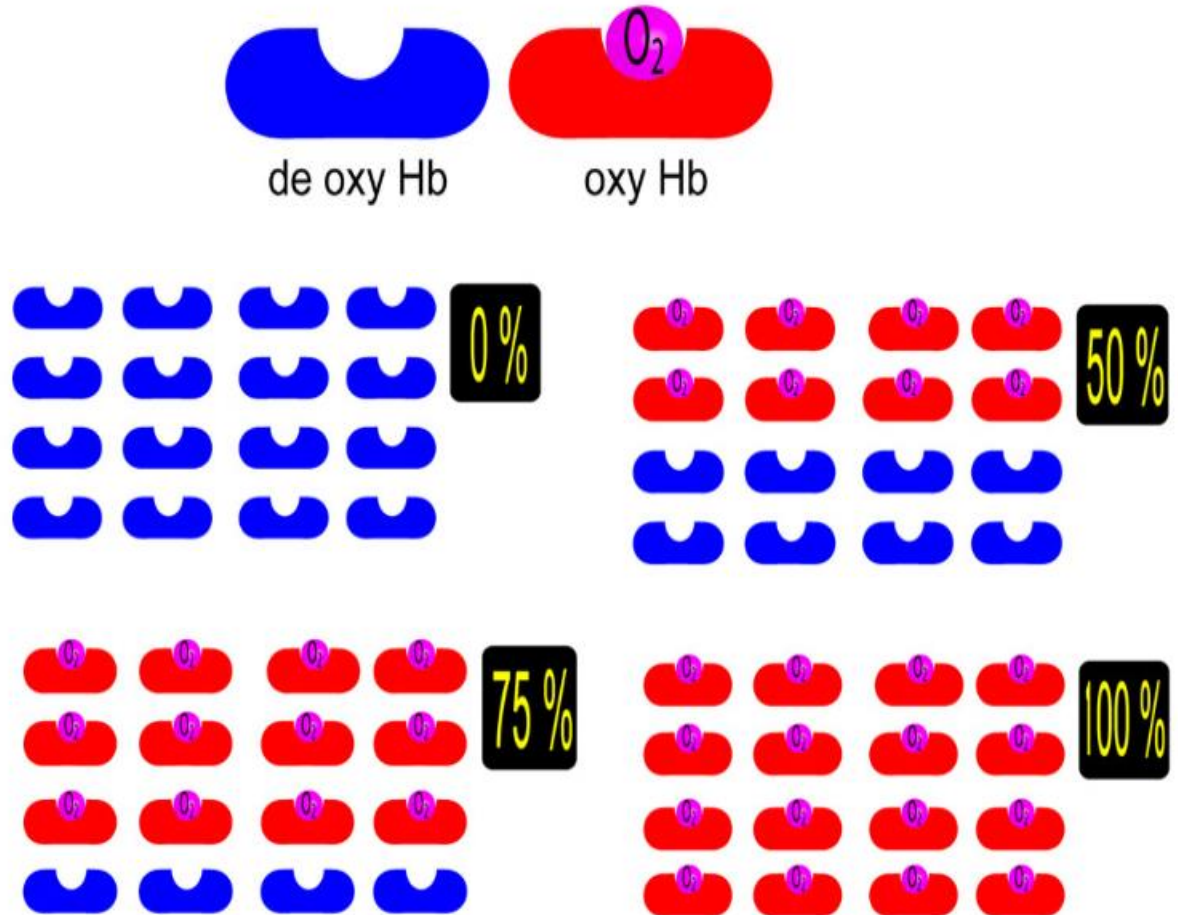
- Cảm biến nhịp tim và oxy trong máu MAX30102
- IC chính: MAX30102
- Đo được nhịp tim và nồng độ Oxy trong máu.
- Điện áp sử dụng: 3.3~5VDC.
- Nhỏ gọn, siêu tiết kiệm năng lượng, thích hợp cho các thiết bị đo nhỏ gọn, Wearable Devices.
- Giao tiếp: I2C, mức tín hiệu TTL.
- Kích thước: 1.9 cm x 1.4 cm x 0.3 cm.[9]

### 2.2.2. Xác định chỉ số SPO2

**Hemoglobin là gì:** (viết tắt là Hb) là một protein phức hợp được tìm thấy trong các tế bào hồng cầu có chứa một phân tử sắt. Chức năng chính của hemoglobin là vận chuyển oxy từ phổi tới các mô của cơ thể, và trao đổi oxy cho carbon dioxid, và sau đó vận chuyển carbon dioxid trở lại phổi, nơi nó trao đổi oxy. Phân tử sắt trong hemoglobin giúp duy trì hình dạng bình thường của các tế bào hồng cầu.

**Độ bão hòa oxy là gì:** là tỉ lệ của Hb có oxy trên tổng số Hb

- Hemoglobin mà không có oxy gọi là Deoxy Hb
- Hemoglobin có oxy gọi là oxy Hb

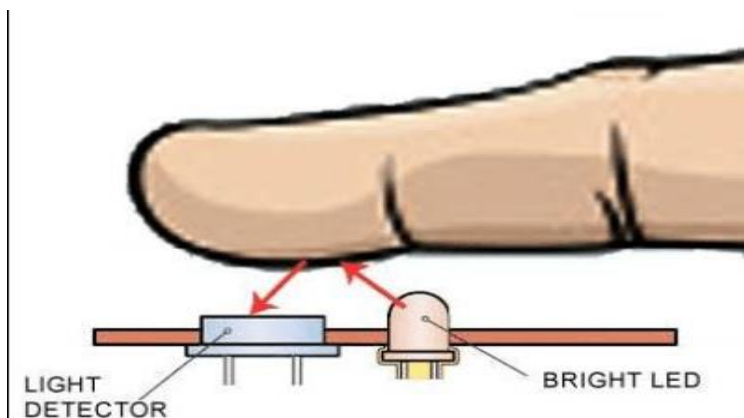


Hình 2.3. Tỷ lệ độ bão hòa oxy.[8]

### 2.2.3. Nguyên lý vật lý được sử dụng để đo SPO<sub>2</sub>.

Cảm biến sử dụng nguyên tắc đo quang phổ để đo độ bão hòa, ánh sáng LED sẽ được phát ra và thu lại bởi cảm biến đối diện, ngón tay sẽ được đặt vào giữa nguồn sáng và cảm biến. Trong phương pháp phản chiếu sẽ có một số phản xạ ánh sáng cố định trở lại cảm biến do ngón tay. Với mỗi nhịp tim sẽ có sự tăng thể tích máu trong ngón tay, điều này sẽ dẫn đến phản xạ ánh sáng trở lại cảm biến nhiều hơn.[8]

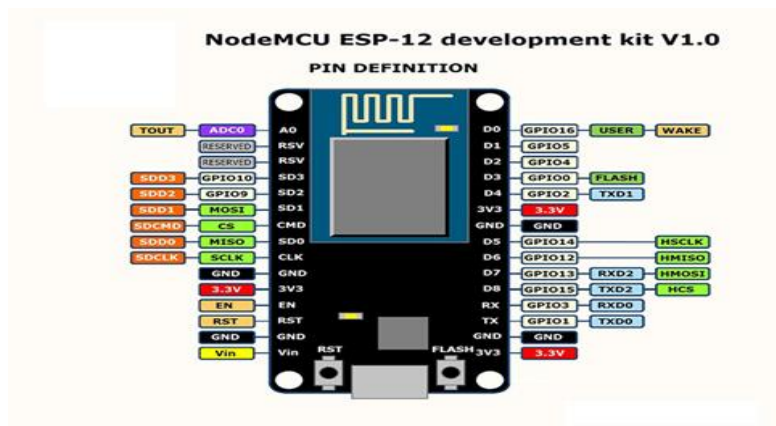
Do đó, nếu chúng ta thấy dạng sóng của tín hiệu ánh sáng nhận được, nó sẽ bao gồm các đỉnh ở mỗi nhịp tim.[8]



Hình 2.4. LED và LDR dùng trong cảm biến.[8]

## 2.3 Tổng quan về ESP8266

NodeMcu8266 là một bảng phát triển được tích hợp trong mô-đun wifi ESP8266. NodeMcu8266 là một thiết bị phần cứng nguồn mở.



Hình 2.3. Sơ đồ pinout NodeMCU8266.[9]

Giới thiệu về NodeMCU8266 và các bước để nhập thư viện bảng ESP: NodeMcu8266

Thiết bị phần cứng nguồn mở cho IOT là NodeMcu8266. Nó có thể được lập trình bằng Arduino IDE và cấu trúc mã hóa vẫn tương tự như của Arduino. Nhưng Arduino IDE không đi kèm với các bảng được hỗ trợ ESP được cài đặt sẵn trên nó, vì vậy người ta cần thêm nhập các tệp đính kèm bảng vào IDE trước khi lập trình. Nó rất đơn giản để làm. Các bo mạch dòng ESP ban đầu không được phát triển để tương thích với Arduino IDE, do đó, trong Arduino IDE, mỗi chân trên NodeMCU tương ứng với các chân GPIO khác nhau cho IDE. Đây là ảnh xạ chân của các chân GPIO NodeMCU. Trong Mã hóa, bạn có thể viết trực tiếp các chân kỹ thuật số của NodeMCU dưới dạng D1, D2, D3 hoặc pin GPIO tương ứng. Đặc trưng:

- Mã nguồn mở
- Tương tác
- Lập trình

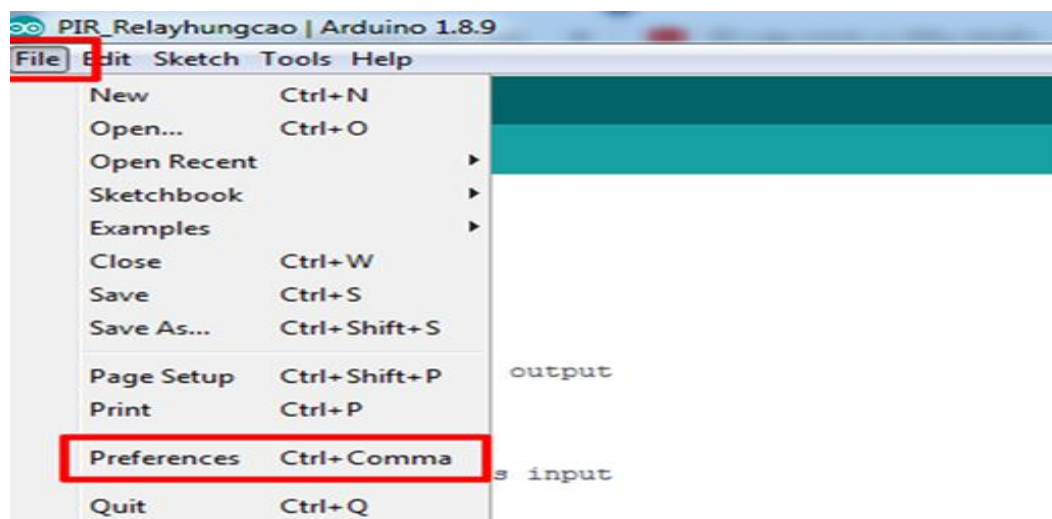
- Giá thấp
- Đơn giản
- Thông minh
- Kết nối Wi-Fi

### 2.3.1. Thông số kĩ thuật:

- WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
- Điện áp hoạt động: 3.3V
- Điện áp vào: 5V thông qua cổng USB
- Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
- Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
- Bộ nhớ Flash: 4MB
- Giao tiếp: Cable Micro USB ( tương đương cáp sạc điện thoại )
- Hỗ trợ bảo mật: WPA/WPA2
- Tích hợp giao thức TCP/IP
- Lập trình trên các ngôn ngữ: C/C++, Micropython,... [7]

### 2.3.2. Cài đặt NodeMCU ESP8266 WIFI:

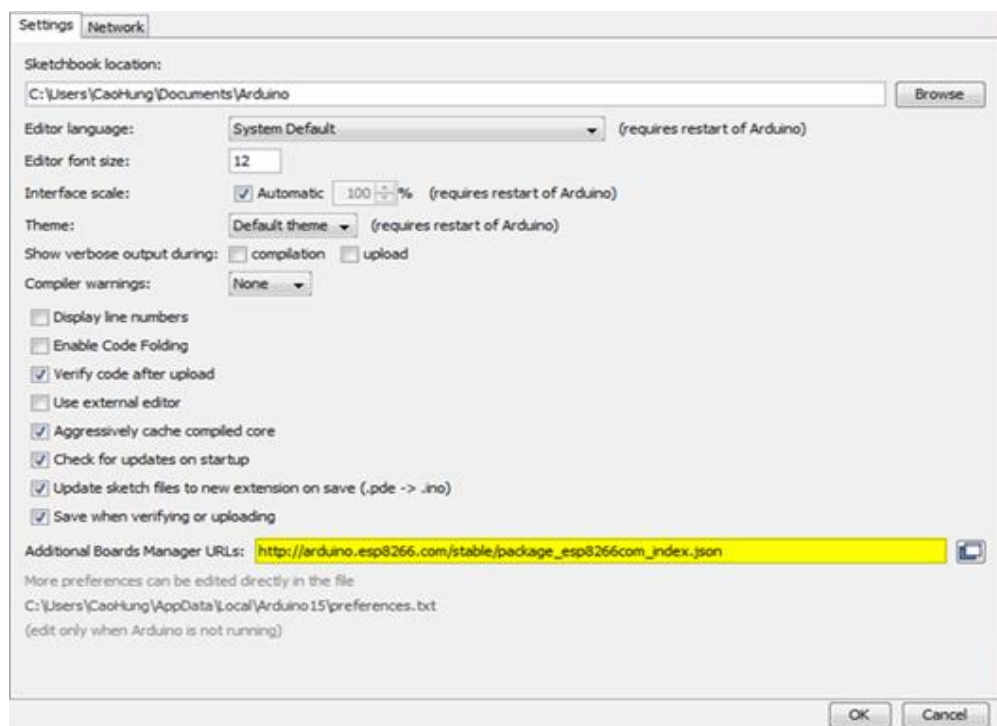
Khởi động Arduino IDE, click vào File trên thanh công cụ chọn Preferences. Chèn một đường Link để Arduino IDE có thể nhận Board. [9]



Hình 2.4. Giao diện trên Arduino IDE

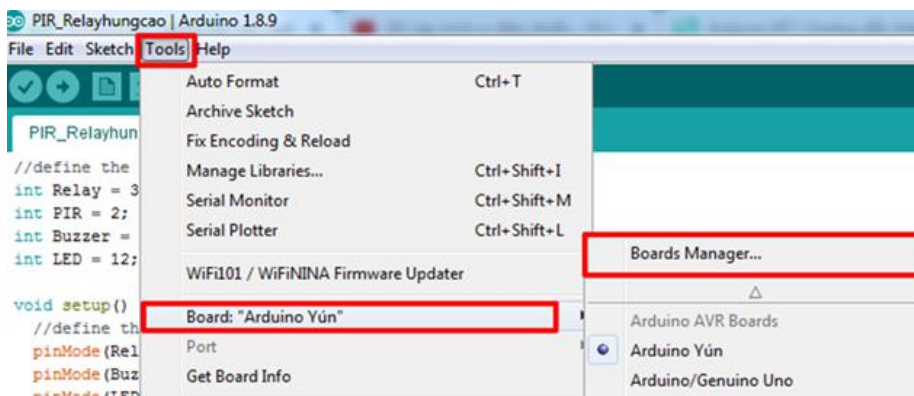
Copy Link tại đây:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



Hình 2.5. Giao diện trên Arduino IDE

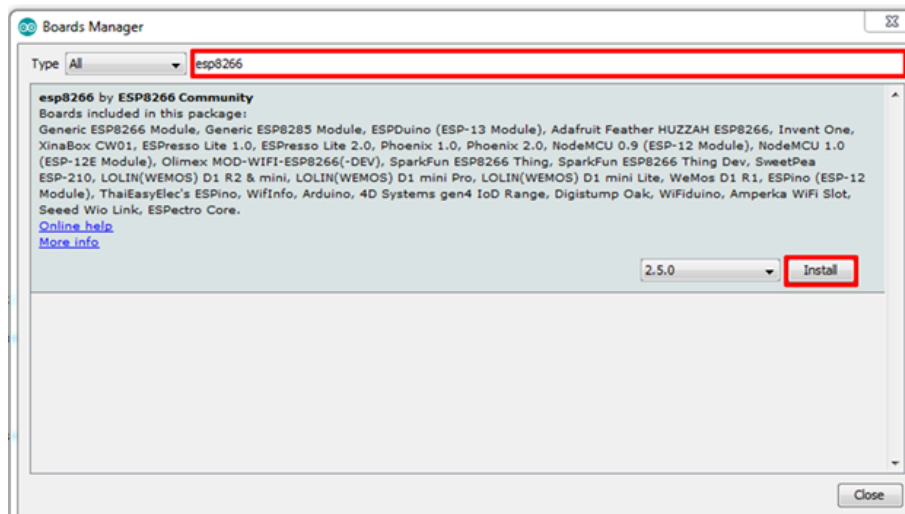
Tiếp theo, Tools > Board > Boards Manager



Hình 2.6. Giao diện trên Arduino IDE

Search Esp8266 để tải danh mục của các Board về. Nhấn Install để tiến hành cài đặt.

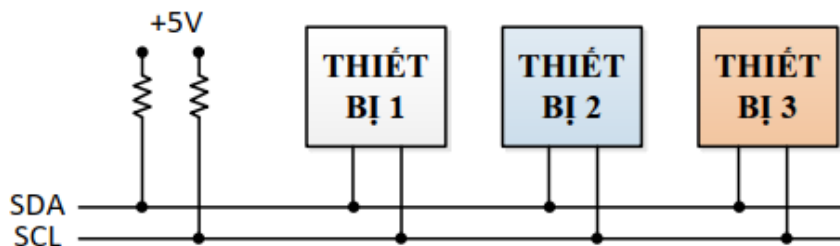




Hình 2.7. Giao diện trên Arduino IDE

### 2.3.3. Sơ lược về chuẩn giao tiếp I2C:

I2C, viết tắt của từ Inter-Integrated Circuit, là một chuẩn truyền thông do hãng điện tử Philips Semiconductor sáng lập, cho phép giao tiếp một thiết bị chủ với nhiều thiết bị tớ với nhau. [2]



Hình 2.8. Hệ thống các thiết bị giao tiếp theo chuẩn I2C.[2]

Chuẩn giao tiếp I2C có 2 đường tín hiệu tên là SDA (serial data) có chức năng truyền tải dữ liệu và tín hiệu SCL (serial clock) truyền tải xung clock để dịch chuyển dữ liệu.

Trong hệ thống truyền dữ liệu I2C, thiết bị nào cung cấp xung clock thì được gọi là chủ (master), thiết bị nhận xung clock được gọi là tớ (slave).

Thiết bị chủ chỉ có 1, thiết bị tớ thì có nhiều, mỗi thiết bị tớ sẽ có 1 địa chỉ độc lập, chuẩn truyền ban đầu dùng địa chỉ 7 bit nên có thể 1 chuẩn giao tiếp với 128 thiết bị tớ. Các thiết bị sau này tăng thêm số bit địa chỉ nên có thể giao tiếp nhiều hơn.

Địa chỉ của thiết bị tớ thường do nhà chế tạo thiết bị thiết lập sẵn.

Trình tự truyền bit trên đường truyền: Thiết bị chủ tạo một điều kiện start. Điều kiện này thông báo cho tất cả các thiết bị tớ lắng nghe dữ liệu trên đường truyền. Sau đó, thiết bị chủ sẽ gửi đi một địa chỉ của thiết bị tớ mà thiết bị chủ muốn giao tiếp và chờ đọc/ghi dữ liệu. Thiết bị tớ mang địa chỉ đó trên bus I2C sẽ phản hồi lại bằng một xung ACK. Khi đó, việc giao tiếp

giữa thiết bị chủ - tớ bắt đầu. Bộ truyền gửi 8 bit dữ liệu đến bộ nhận, bộ nhận trả lời với 1 bit ACK. Để kết thúc, thiết bị chủ tạo ra một điều kiện STOP.[2]

## **2.4. Giới thiệu về lập trình Web**

### **2.4.1. Ngôn ngữ HTML**

**HTML** (tiếng anh, viết tắt cho *HyperText Markup Language*, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web.

### **2.4.2. Ngôn ngữ CSS**

**CSS** là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.[11]

## **2.5. Lấy dữ liệu thời gian từ internet**

### **2.5.1 Giới thiệu giao thức NTP**

#### **a. Giới thiệu về NTP:**

Network Time Protocol (NTP) là một thuật toán phần mềm giữ cho các máy tính và các thiết bị công nghệ khác nhau có thể đồng bộ hóa thời gian với nhau. NTP đã đạt được thành công trong việc giữ các thiết bị đồng bộ hóa hiệu quả trong chỉ trong vài milliseconds (1/1000s), nhưng để có thể làm được điều này nó cần phải có một hệ thống thời gian đáng tin cậy để sử dụng làm điểm thời gian chính cho việc đồng bộ.

NTP hoạt động bằng cách sử dụng một nguồn thời gian chính duy nhất (NTP Server), nó sử dụng để đồng bộ tất cả các thiết bị trên mạng.[11]

NTP là một trong những giao thức Internet lâu đời nhất vẫn còn được sử dụng (từ trước năm 1985). NTP được thiết kế đầu tiên bởi Dave Mills tại trường đại học Delaware, hiện ông vẫn còn quản lý nó cùng với một nhóm người tình nguyện.[11]

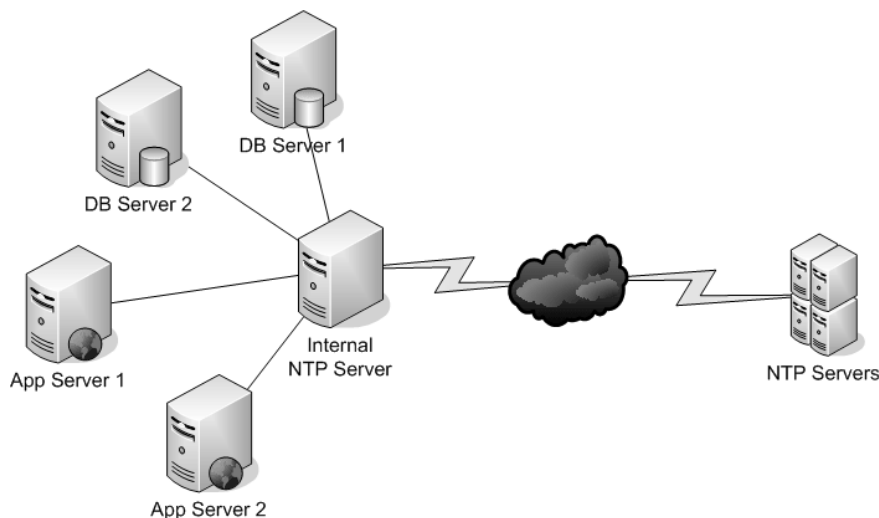
NTP sử dụng thuật toán Marzullo, và nó cũng hỗ trợ các tính năng như giây nhuận. NTPv4 thông thường có thể đảm bảo độ chính xác trong khoảng 10 mili giây (1/100s) trên mạng Internet công cộng, và có thể đạt đến độ chính xác 200 micro giây (1/5000s) hay hơn nữa trong điều kiện lý tưởng của môi trường mạng cục bộ.

Chi tiết hoạt động của NTP được quy định trong các RFC 778, RFC 891, RFC 956, RFC 958 (thay thế bởi 1305), và RFC 1305. Chuẩn đang được triển khai là phiên bản 4 (NTPv4) .[10]



***b. NTP server:***

Máy chủ NTP hay máy chủ thời gian là các thuật ngữ cùng mô tả một khái niệm: một thiết bị được sử dụng để nhận biết yêu cầu đồng bộ thời gian và phân phối tín hiệu thông tin thời gian. Thật ra, một máy chủ NTP Server cũng chỉ sử dụng Network Time Protocol (NTP), trong vô vàn các giao thức thời gian khác nhau tồn tại thì NTP được sử dụng phổ biến tới hơn 90%.

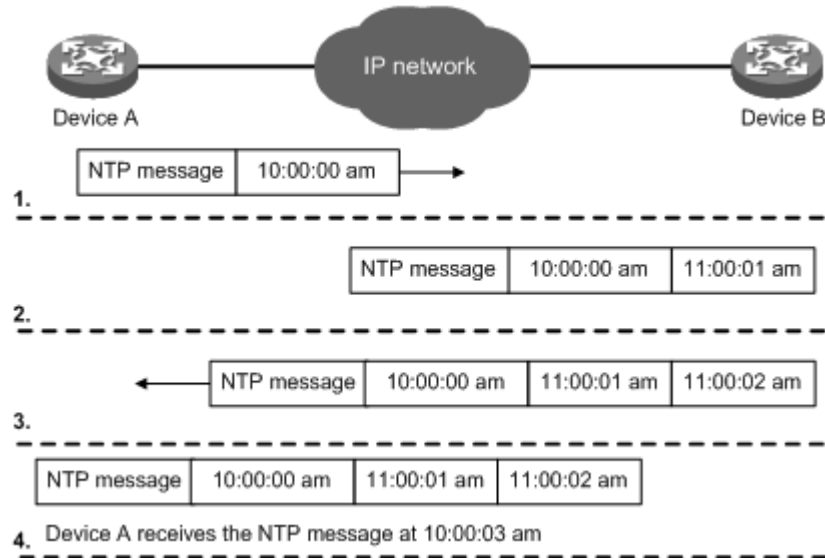


Hình 2.8. Hình ảnh minh họa về NTP.[10]

Các tín hiệu thời gian được sử dụng bởi hầu hết các máy chủ NTP là nguồn thời gian UTC. UTC (Coordinated Universal Time) là thời gian toàn cầu dựa trên thời gian đồng hồ nguyên tử. Bằng cách sử dụng UTC, máy chủ NTP có thể tác động, đồng bộ hóa mạng cùng thời gian với hàng triệu mạng máy tính khắp nơi trên thế giới. Nếu không có UTC, nhiều giao dịch trực tuyến sẽ không thể nào thực hiện được.

Tín hiệu thời gian có thể được nhận bởi NTP server hoặc (time server) qua một số cách như mạng Internet toàn cầu, thời gian của các quốc gia, tần số truyền (sóng dài) hoặc mạng GPS (hệ thống định vị toàn cầu). Khi nhận được tín hiệu, time server kiểm tra tính xác thực của tín hiệu này (trừ các nguồn internet không thể xác thực), đánh giá tính chính xác của nó và phân phối nó trong mạng.[12]

**c. Phương thức hoạt động:**



Hình 2.9. Minh họa phương thức hoạt động của NTP.[10]

NTP client gửi một gói tin trong đó chứa tem thời gian được chuyển tiếp đến máy chủ ở dạng số ngày càng tăng từ một điểm thời gian cố định trước đó (unixtime): ngày 1 tháng 1 năm 1990.

Máy chủ NTP sử dụng tem thời gian để tính toán, nếu thời gian trên mạng không khớp sẽ được cộng trừ thêm số giây để khớp với đồng hồ/máy chủ NTP cấp cao hơn đang tham chiếu. Sau đó gửi trả lại NTP client một gói tin khác, có thể thời gian là thời điểm nó gửi gói tin đó đi.

NTP client nhận được gói tin đó, tính toán độ trễ, dựa và tem thời gian mà nó nhận được cùng với độ trễ đường truyền, NTP client sẽ set lại thời gian của nó.

NTP chính xác trong khoảng 1/100s trên môi trường internet và thực hiện tốt hơn trên mạng LAN và WAN với độ chính xác là 1/5000s.

## 2.5.2 Giới thiệu giao thức UDP

**a. Khái niệm:**

UDP là viết tắt của cụm từ User Datagram Protocol. UDP là một phần của bộ giao thức Internet được sử dụng bởi các chương trình chạy trên các máy tính khác nhau trên mạng. Không giống như TCP/IP, UDP được sử dụng để gửi các gói tin ngắn gọi là datagram, cho phép truyền nhanh hơn. Tuy nhiên, UDP không cung cấp kiểm tra lỗi nên không đảm bảo toàn vẹn dữ liệu.

***b. Cách hoạt động:***

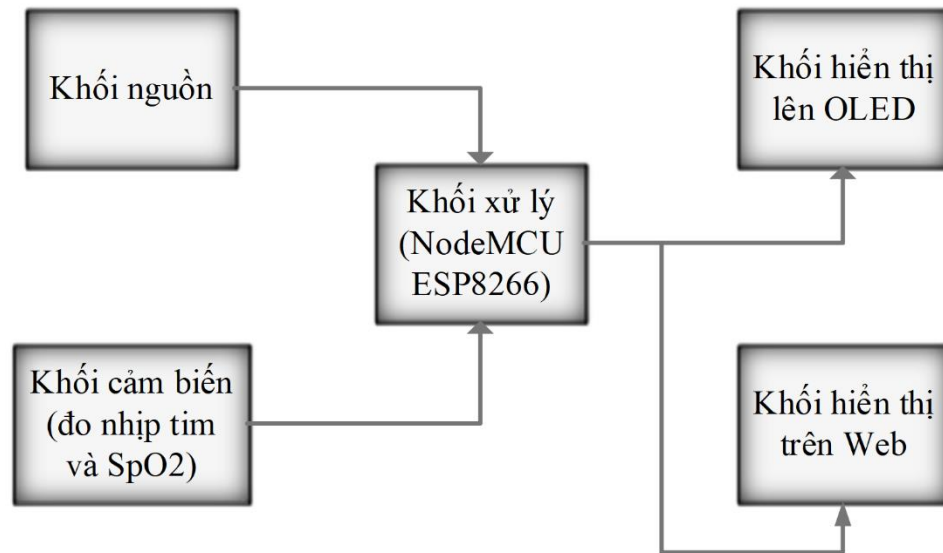
Giao thức UDP hoạt động tương tự như TCP nhưng nó không cung cấp kiểm tra lỗi khi truyền gói tin.

Khi một ứng dụng sử dụng UDP, các gói tin chỉ được gửi đến người nhận. Người gửi không đợi để đảm bảo người nhận nhận được gói tin hay không, mà nó tiếp tục gửi các gói tiếp theo. Nếu người nhận bỏ lỡ một vài gói tin UDP, gói tin đó bị mất vì người gửi sẽ không gửi lại chúng. Điều này có nghĩa là các thiết bị có thể giao tiếp nhanh hơn.[12]

## CHƯƠNG 3. XÂY DỰNG VÀ THIẾT KẾ HỆ THỐNG

### 3.1. Sơ đồ khối và chức năng của từng khối

#### 3.1.1. Sơ đồ khối.



Hình 3.1. Sơ đồ khối của hệ thống

#### 3.1.2. Chức năng

- **Khối nguồn:** có chức năng cung cấp nguồn cho toàn bộ mạch hoạt động.
- **Khối cảm biến max30102:** có chức năng đo nhịp tim và nồng độ SpO2 của con người và gửi giá trị đo được cho NodeMCU ESP8266 khi có tín hiệu yêu cầu.
- **Khối xử lý trung tâm:** có chức năng điều khiển cảm biến max30102, đồng thời điều khiển hiển thị dữ liệu trên khối hiển thị.
- **Khối hiển thị:** có chức năng hiển thị giá trị nhịp tim và nồng độ SpO2 trong máu đo được.

#### 3.1.3. Hoạt động của hệ thống

Tín hiệu nhịp tim sẽ được thu thập và xử lý để tính toán giá trị heart rate (HR) nồng độ oxy trong máu (SpO2) thông qua hai khối là khối cảm biến nhịp tim và khối xử lý trung tâm. Giá trị HR, SpO2 sau khi được tính toán sẽ được hiển thị ra màn hình cùng với đồng hồ thời gian thực được cập nhật thông qua sever online và giá trị HR, SpO2 đồng thời được gửi lên hệ thống website để người dùng có thể theo dõi từ xa.

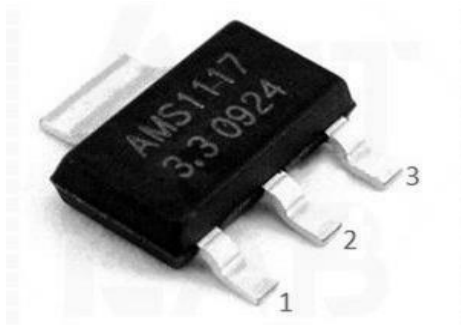
### 3.2. Thiết kế chi tiết

#### 3.2.1. Khối nguồn.

Trước khi làm mạch nguồn ta phải tính toán dòng hoạt động của tất cả các linh kiện trong mạch để biết dòng tổng mà từ đó thiết kế mạch nguồn.

Bảng 3.1: Thông số sử dụng nguồn của các thiết bị trong mạch

STT	Tên linh kiện	Dòng làm việc	Áp làm việc	Tổng dòng
1	Esp8266	300mA	3.3v	312mA
2	Màn hình oled	12mA		
3	Max30102	12mA	3.3V	12mA



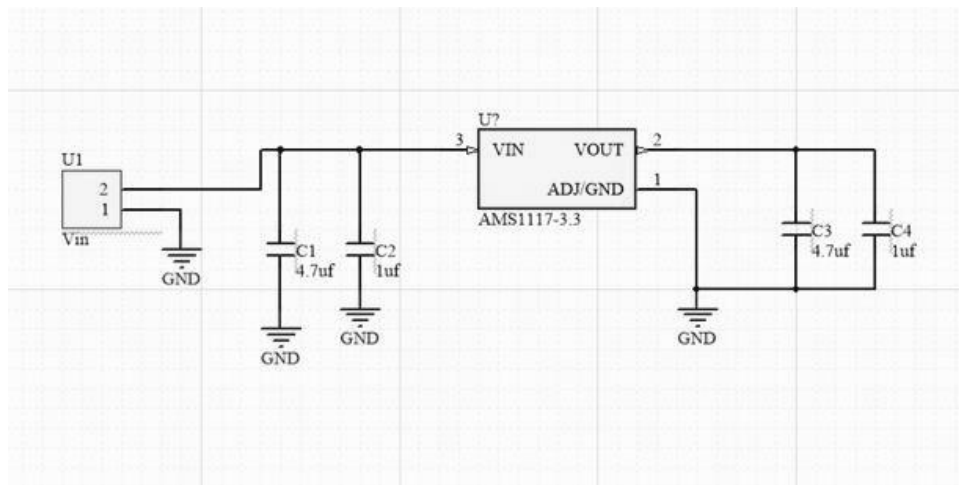
Hình 3.2. IC nguồn ASM117. [1]

1- Ground

2- VOUT

3- VIN

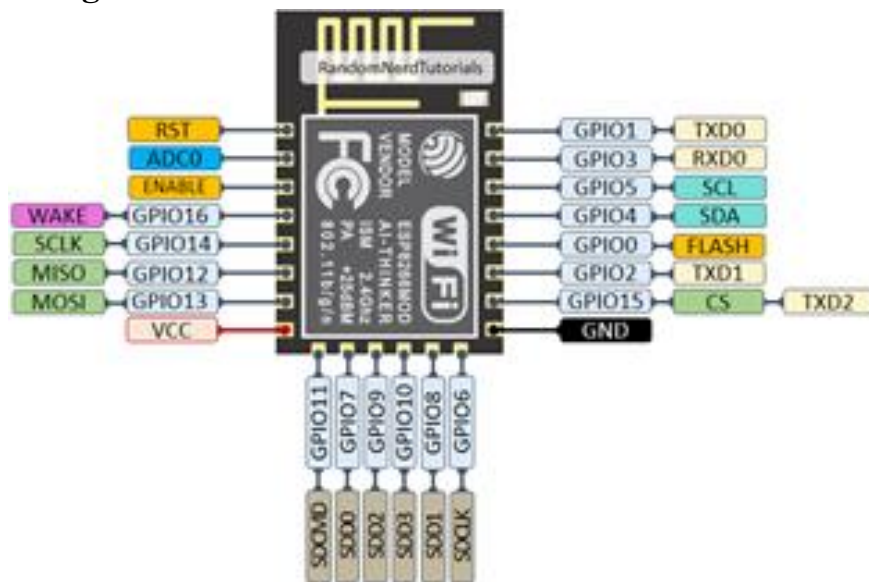
- Điện áp ngõ ra: 3.3V
- Dòng ra tối đa: 1A
- Áp ngõ vào:  $1.5 < V_{in} - V_{out} < 12V$  (4.8-8.7VDC)
- Nhiệt độ hoạt động: âm 40 đến 125 độ C. [1]



Hình 3.3. Mạch nguồn ổn áp

- Sử dụng ic ổn áp AMS1117-3.3V có định điện áp đầu ra 3.3v
- Tụ C1 C2 C3 C4 các tụ lọc nguồn
- Cung cấp điện áp một chiều ở đầu ra không đổi trong hai trường hợp điện áp đầu vào thay đổi hoặc dòng tiêu thụ của tải thay đổi, tuy nhiên sự thay đổi này phải có giới hạn.
- Cho điện áp một chiều đầu ra có chất lượng cao, giảm thiểu được hiện tượng gợn xoay chiều.

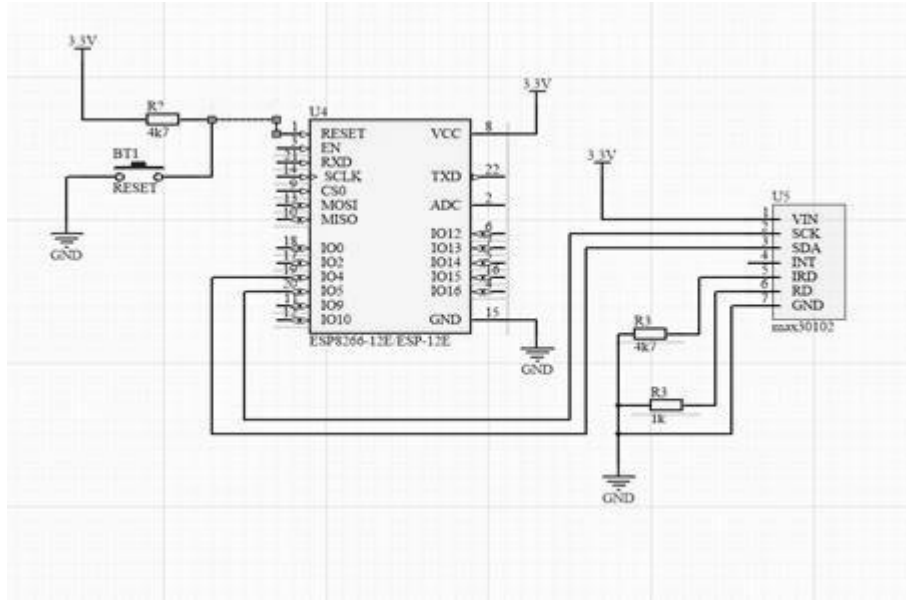
### 3.2.2. Khối xử lý trung tâm



Hình 3.4. esp8266.[9]

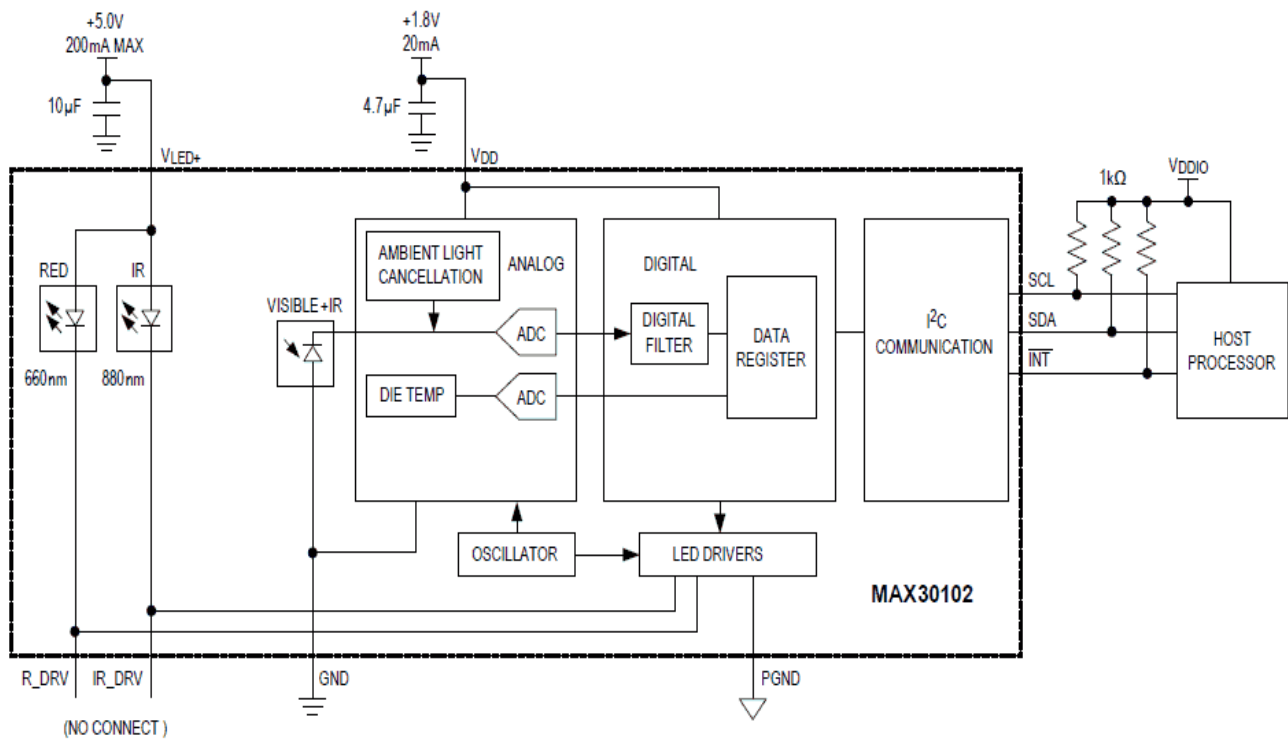
Được kết nối với màn hình oled và cảm biến max30102 thông qua chuẩn giao tiếp I2c ở chân SCL và SDA, vi xử lý esp8266 được lập trình đọc dữ liệu cảm biến max30102 gửi về màn hình oled và webserver.

### 3.2.3 Khôi cảm biến

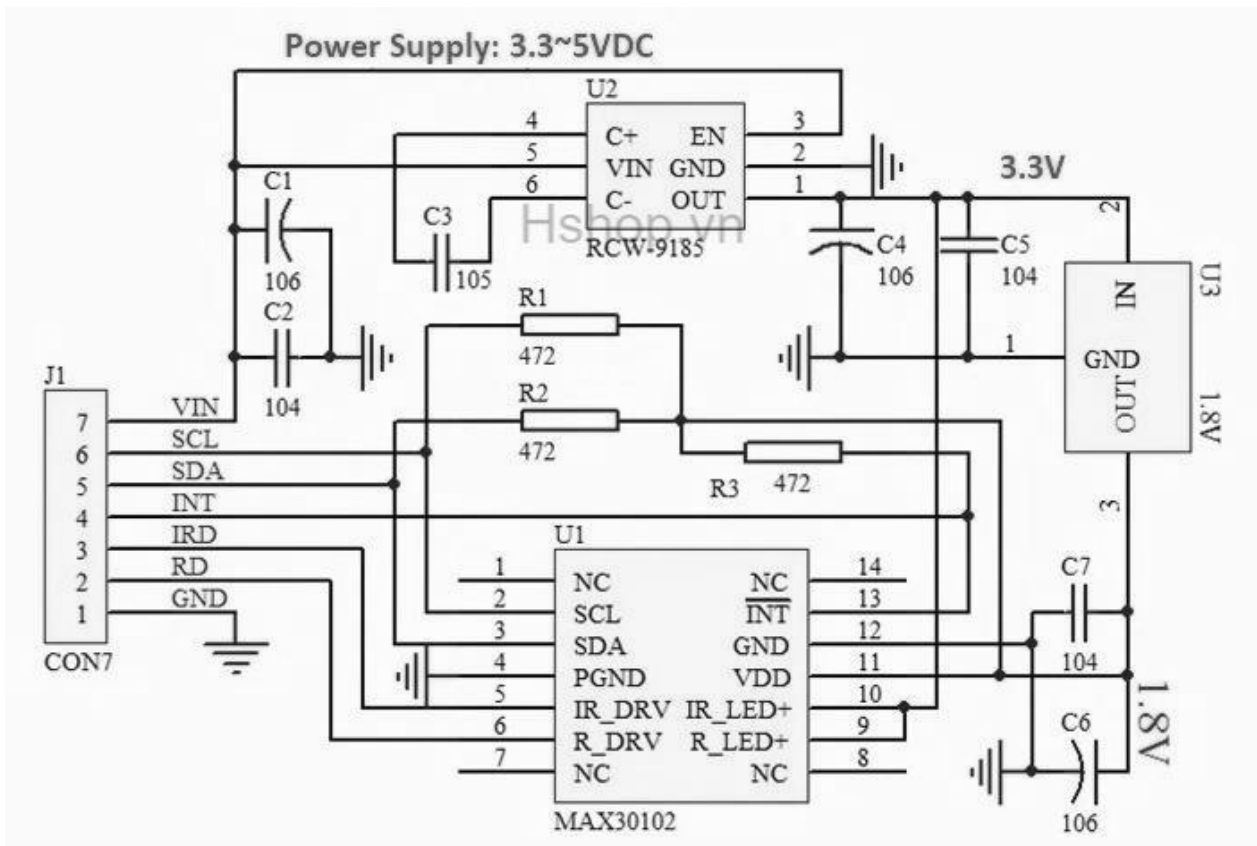


Hình 3.5. Sơ đồ kết nối max30102 với esp8266

a. Sơ đồ chức năng cảm biến MAX30102:



Hình 3.6. Sơ đồ chức năng.[4]



Hình 3.7. Sơ đồ nguyên lý cảm biến max30102.[9]

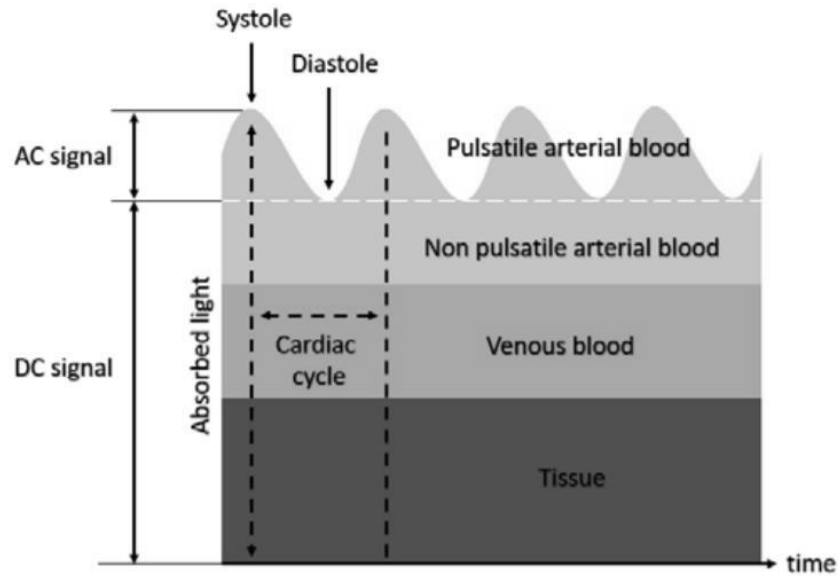
MAX30102 tích hợp trình điều khiển LED đỏ và IR để điều chỉnh xung LED cho phép đo SpO2 và HR. Dòng LED có thể được lập trình từ 0 đến 50mA với điện áp cung cấp phù hợp. Độ rộng xung LED có thể được lập trình từ 69us đến 411us để cho phép thuật toán tối ưu hóa độ chính xác của SpO2 và HR và mức tiêu thụ điện dựa trên các trường hợp sử dụng. [4]

b. Phương pháp theo dõi độ bão hòa oxy trong máu và nhịp tim:

### Heart Rate

Trong một phép đo đo nhịp tim và nồng độ Oxy trong máu được thiết lập, đèn LED chiếu sáng mô da và tín hiệu phản xạ được phát hiện bởi photodiode. Tín hiệu phản xạ này chứa ánh sáng được điều chế quang học bởi những thay đổi thể tích của động mạch và mao mạch. Tín hiệu quang học (PPG) này cực kỳ quan trọng trong việc xác định nhịp tim và mức SpO2. Tín hiệu PPG có thành phần DC và thành phần AC kết hợp với nó, như có thể thấy bên dưới trong Hình 3.9. [4]



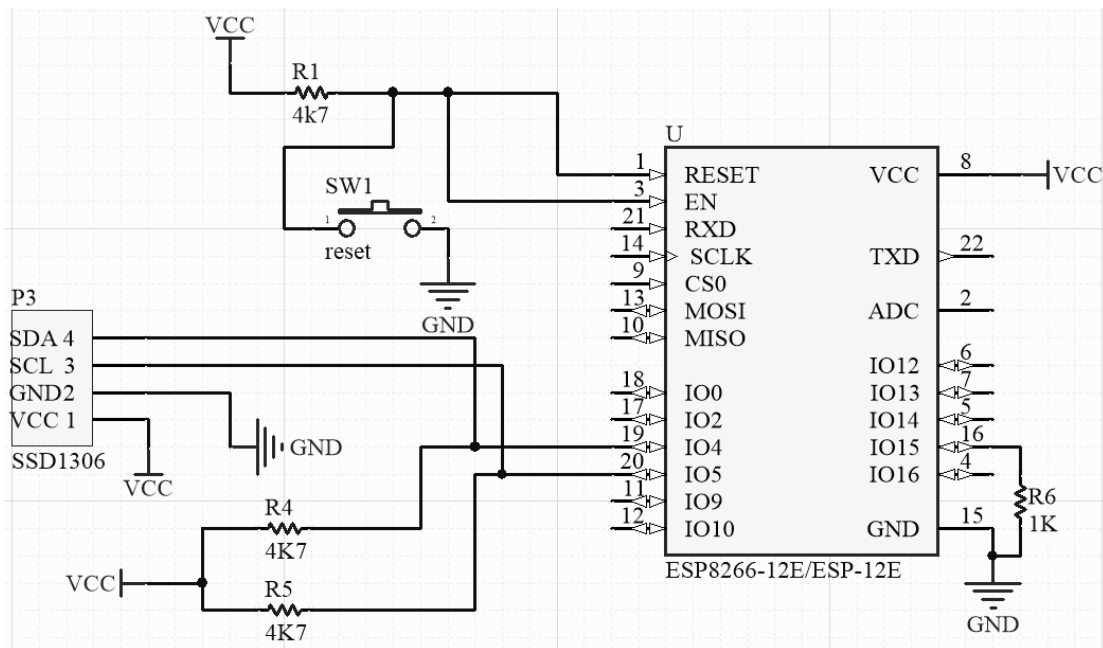


Hình 3.9. Thành phần DC và AC của tín hiệu PPG.[4]

## SpO2

Các phép đo MAX3010x SpO2 sử dụng hai đèn LED bước sóng khác nhau để xác định tỷ lệ hemoglobin có oxy hóa với hemoglobin không có oxy. Đèn LED màu đỏ và hồng ngoại được sử dụng để xác định tín hiệu PPG riêng biệt. Vì các thành phần DC và các thành phần AC của hai LED có biên độ khác nhau, chúng phải được chuẩn hóa để so sánh hữu ích. [4]

### 3.2.4. Khởi hiển thị trên OLED



Hình 3.10. Khởi hiển thị kết nối khởi xử lý.

### a. Giao tiếp MCU I2C:

Giao diện truyền dữ liệu I2C bao gồm bit địa chỉ slave SA0, tín hiệu dữ liệu bus I2C SDA (D2 cho đầu ra và D1 cho đầu vào) và tín hiệu xung clock bus I2C-SCL (D0). Cả dữ liệu và tín hiệu xung clock phải được kết nối với điện trở kéo lên được sử dụng để khởi tạo thiết bị.[6]

Slave address bit (SA0)  $b_7b_6b_5b_4b_3b_2b_1b_0$

0 1 1 1 0 SA0 R/W#.

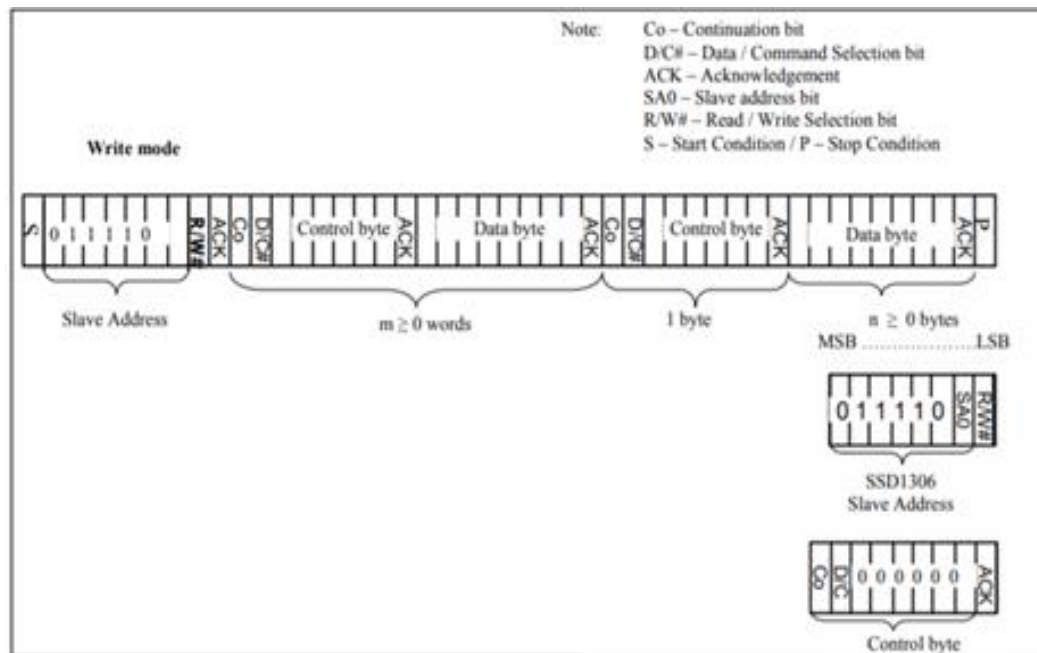
R / W # bit được sử dụng để xác định chế độ hoạt động của giao diện I2C-bus.

R / W # = 1 ở chế độ đọc. R / W # = 0 ở chế độ ghi.

I2C - bus data signal (SDA) hoạt động như một kênh giao tiếp giữa máy phát và máy thu. Dữ liệu và xác nhận được gửi qua SDA

I2C - bus clock signal (SCL) Việc truyền thông tin trong bus I2C theo tín hiệu xung clock SCL. Mỗi lần truyền bit dữ liệu được thực hiện trong một khoảng thời gian xung clock SCL.

**Ghi dữ liệu bus - I2C:** Giao diện bus I2C cho phép truy cập để ghi dữ liệu và lệnh vào thiết bị



Hình 3.11. Định dạng dữ liệu bus I2C.[6]

*b. Sơ đồ kết nối:*

Bảng 3.2. Cách kết nối của màn hình OLED sử dụng giao thức truyền thông I2C

PIN	ESP8266
Vin	3.3V
GND	GND
SCL	GPIO 5 (D1)
SDA	GPIO 4 (D2)

### 3.2.5. Khởi hiển thị Web

*a. Tổng quan về Webserver:*

Một trang web là một tập hợp các trang web đó là tập tin kỹ thuật số thường được viết bằng HyperText Markup Language (HTML). Đối với một trang web có sẵn cho tất cả mọi người trên thế giới ở mọi thời điểm, cần phải được lưu trữ hoặc "hosted" trên một máy tính được kết nối với Internet 24/7/365.

Máy chủ Web (Web Server) là máy tính mà trên đó cài đặt phần mềm phục vụ web, đôi khi người ta cũng gọi chính phần mềm đó là web server. Tất cả các web server đều hiểu và chạy được các file \*.htm và \*.html. Tuy nhiên mỗi web server lại phục vụ một số kiểu file chuyên biệt chẳng hạn như IIS của Microsoft dành cho \*.asp, \*.aspx; Apache dành cho \*.php; Sun Java system web server của SUN dành cho \*.jsp.

Máy Web Server là máy chủ chứa những website đã được thiết kế cùng với những thông tin liên quan khác. (Các mã Script, các chương trình, và các file Multimedia).

Web Server có khả năng gửi đến máy khách những trang Web thông qua môi trường Internet (hoặc Intranet) qua giao thức HTTP – giao thức được thiết kế để gửi các file đến trình duyệt Web (Web Browser), và các giao thức khác.

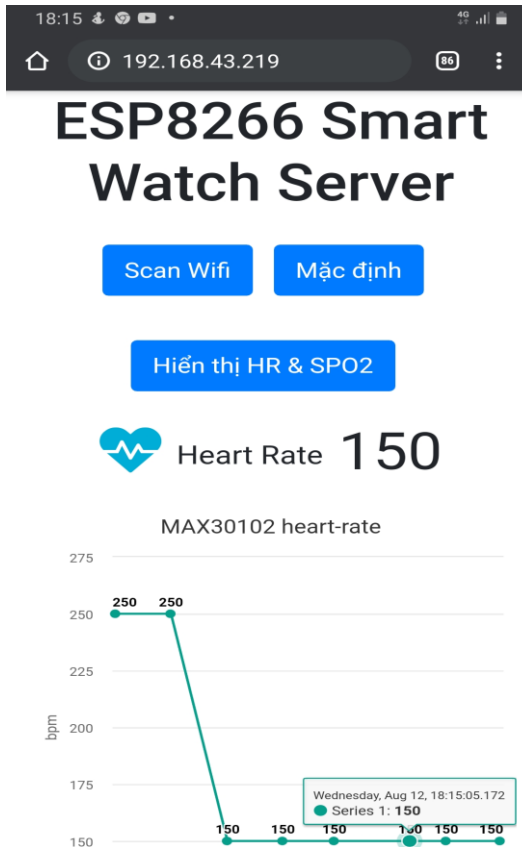
Tất cả các Web Server đều có một địa chỉ IP (IP Address) hoặc cũng có thể có một Domain Name. Giả sử khi bạn đánh vào thanh Address trên trình duyệt của bạn một dòng <http://www.abc.com> sau đó gõ phím Enter bạn sẽ gửi một yêu cầu đến một

Server có Domain Name là [www.abc.com](http://www.abc.com). Server này sẽ tìm trang Web có tên là [index.htm](#) rồi gửi nó đến trình duyệt của bạn.

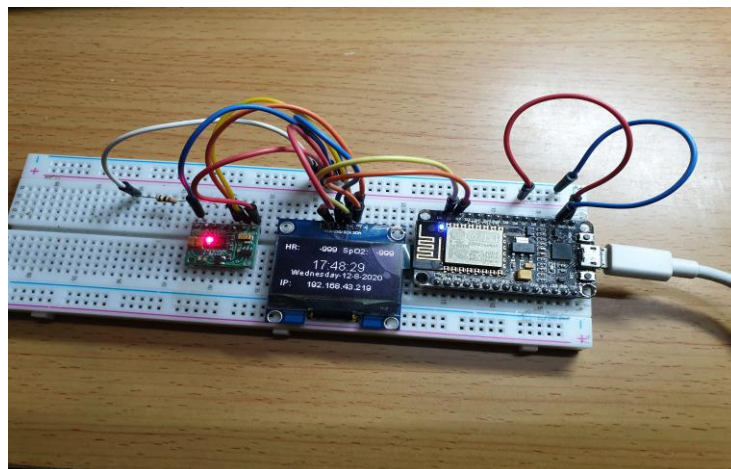
Web server chia ra làm 2 phần chính: Phần cứng (Hardware) và Phần mềm (Software)

Bất kỳ một máy tính nào cũng có thể trở thành một Web Server bởi việc cài đặt lên nó một chương trình phần mềm Server Software và sau đó kết nối vào Internet.

Khi máy tính của bạn kết nối đến một Web Server và gửi đến yêu cầu truy cập các thông tin từ một trang Web nào đó, Web Server Software sẽ nhận yêu cầu và gửi lại cho bạn những thông tin mà bạn mong muốn.



Hình 3.13. Hình ảnh hiển thị trên Web server



Hình 3.14. Hình ảnh hiển thị trên màn hình Oled

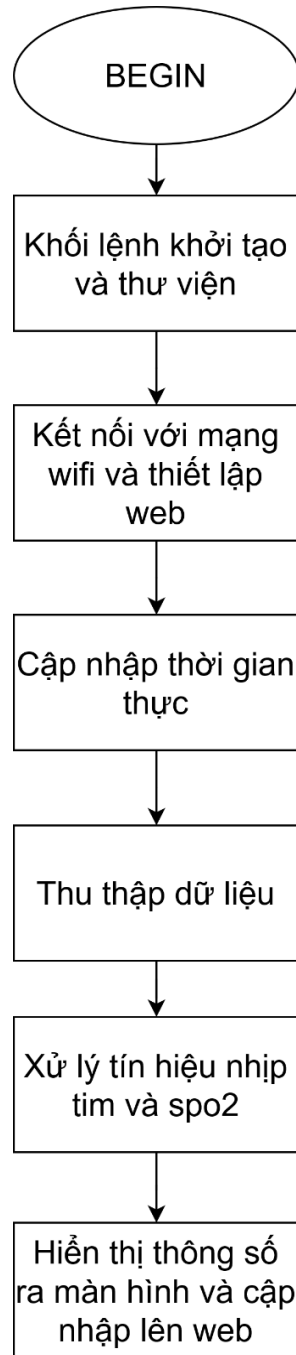
*b. Nguyên lý hoạt động webservice:*

Cảm biến đo nhịp tim và oxy trong máu MAX30102 và đồng hồ NTP sẽ được gửi thông qua giao thức HTTP, dùng javascript để bắt các sự kiện và hiển thị các giá trị trên web bằng html



## 4.3. Lưu đồ giải thuật

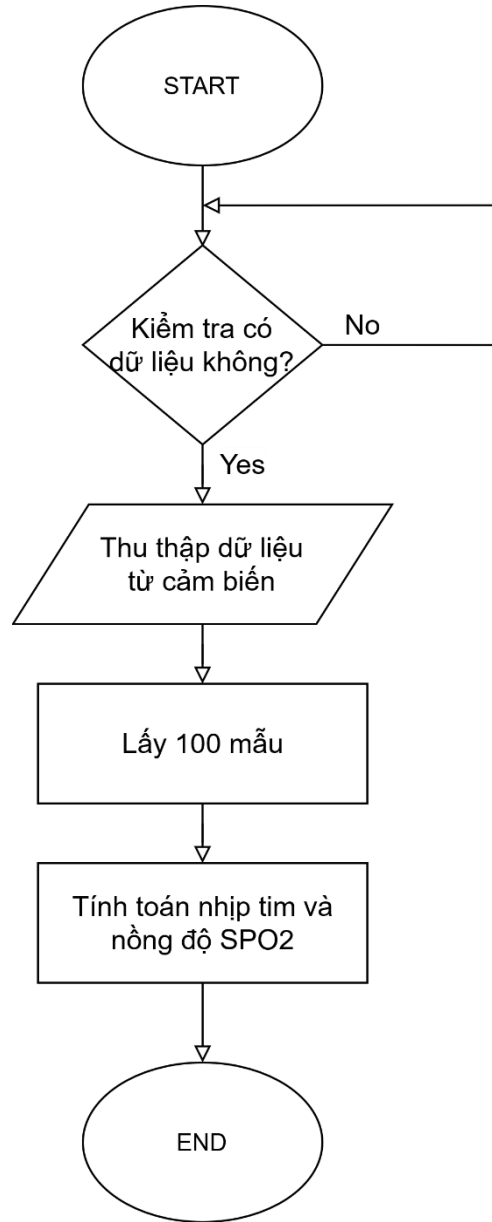
### 4.3.1 Chương trình chính



Hình 4.2: Lưu đồ thuật toán chương trình

Ban đầu khởi tạo các biến sử dụng trong chương trình, sau đó sẽ hiển thị giao diện cũng như giá trị của dữ liệu. Tiếp theo khối xử lý trung tâm sẽ kết nối với mạng wifi và thiết lập trang web đồng thời cập nhật thời gian thực. Sau đó khối xử lý trung tâm sẽ xử lý tín hiệu nhịp tim và huyết áp để có thể hiển thị thông số lên màn hình và cập nhật lên trang web.

### 4.3.2 Giải thuật đo nhịp tim và spo2:

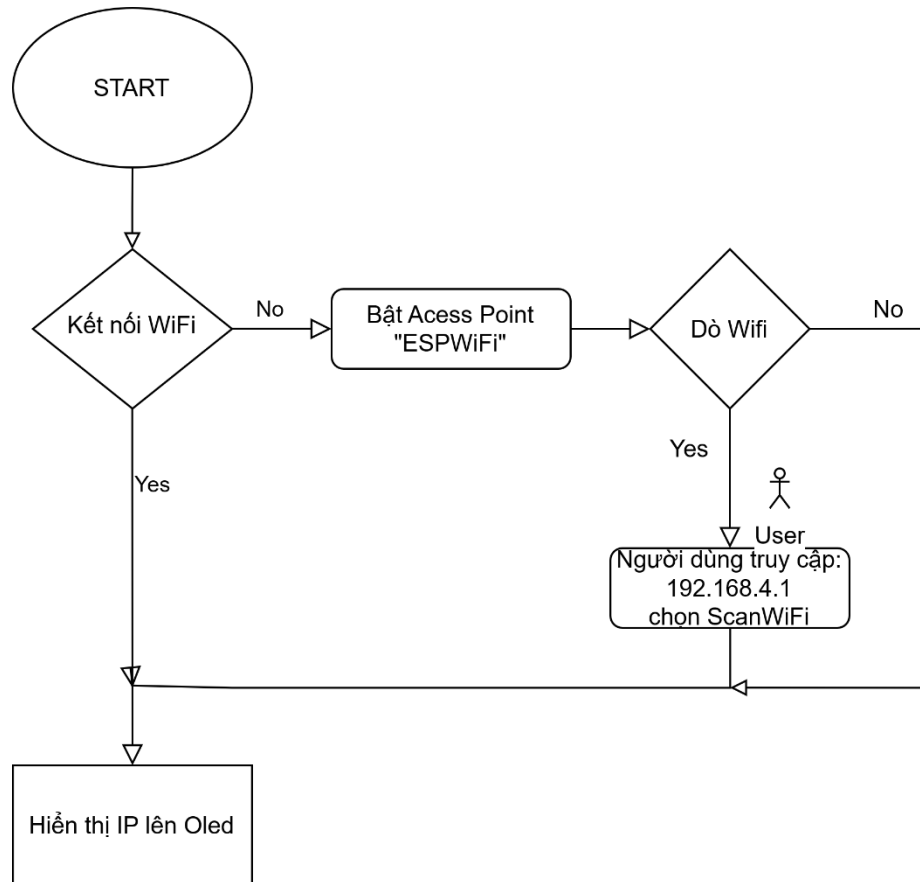


Hình 4.3: Lưu đồ giải thuật đo nhịp tim

Đầu tiên đọc giá trị analog từ đèn LED của cảm biến, lưu 100 giá trị đọc được nhằm giảm nhiễu và tăng độ chính xác. Dùng các giá trị đã lưu tính toán nhịp tim và nồng độ oxy trong máu dựa vào cơ sở lý thuyết đã nêu.



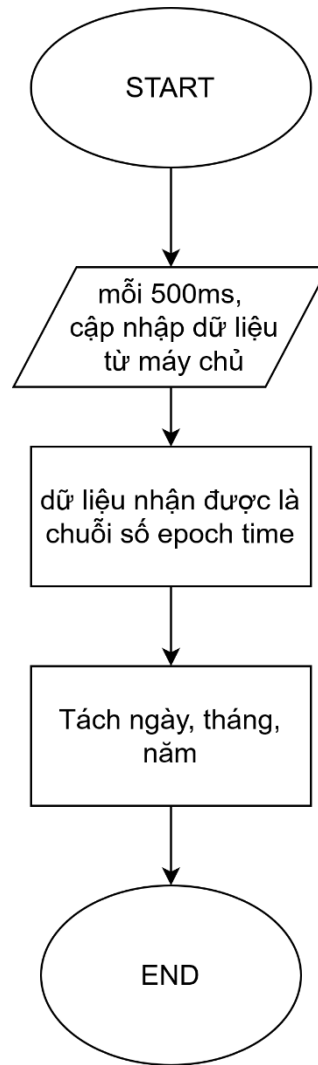
### 4.3.3 Giải thuật dò WiFi:



Hình 4.4: lưu đồ giải thuật dò wifi

ESP8266 dò wifi và lưu các tên wifi tìm được vào bộ nhớ Flash. Người dùng chọn wifi và nhập mật khẩu trên giao diện web, mật khẩu được gửi về thông qua HTTP request và được đóng gói theo dạng JSON. ESP sẽ tách ra thành tên và mật khẩu và kết nối với wifi đã nhập.

#### 4.3.4 Giải thuật thời gian thực:



Hình 4.5: Lưu đồ giải thuật thời gian thực

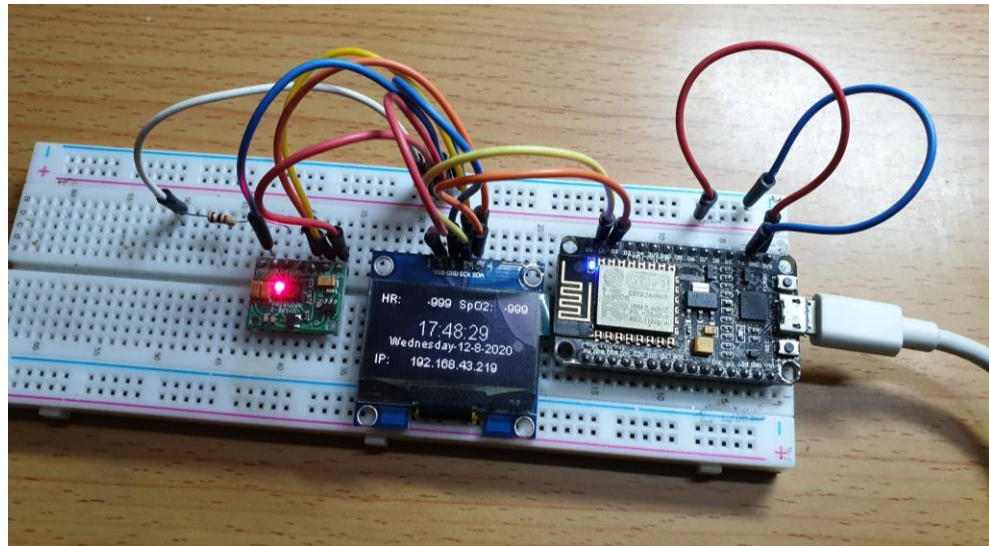
Cứ sau mỗi 500ms, ESP sẽ cập nhập dữ liệu về thời gian thông qua giao thức NTP. Thời gian lúc này là 1 biến số với điểm bắt đầu là 7:00 1/1/1970, tách biến số đó ra và so sánh với mốc bắt đầu (1970) ta sẽ tìm được thời gian hiện tại.

Link source code: <https://github.com/CodeMarker01/DoAn1-dowifi>

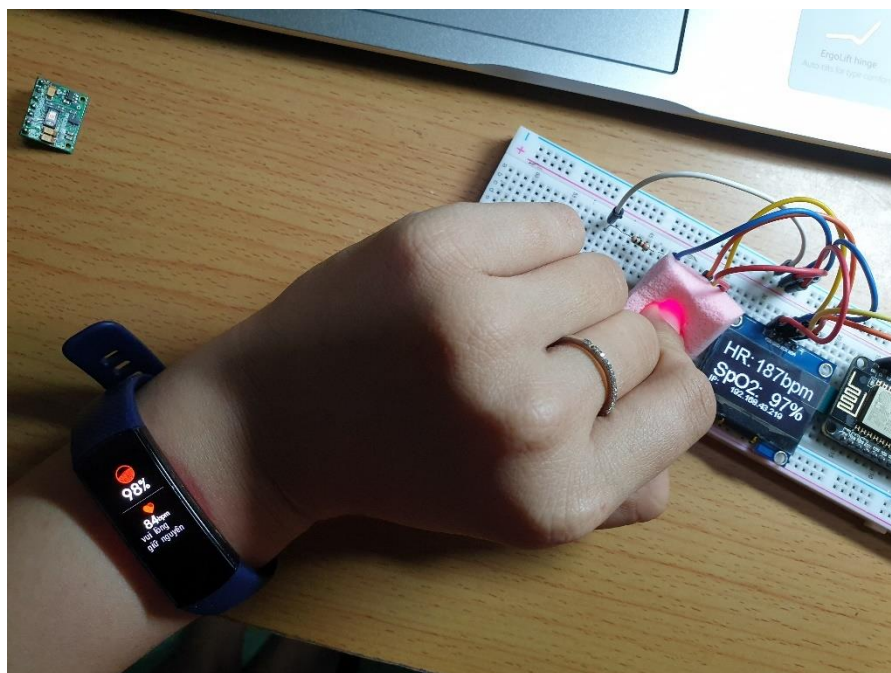
## CHƯƠNG 5. KẾT QUẢ THỰC HIỆN VÀ KẾT LUẬN

### 5.1. Kết quả thực hiện

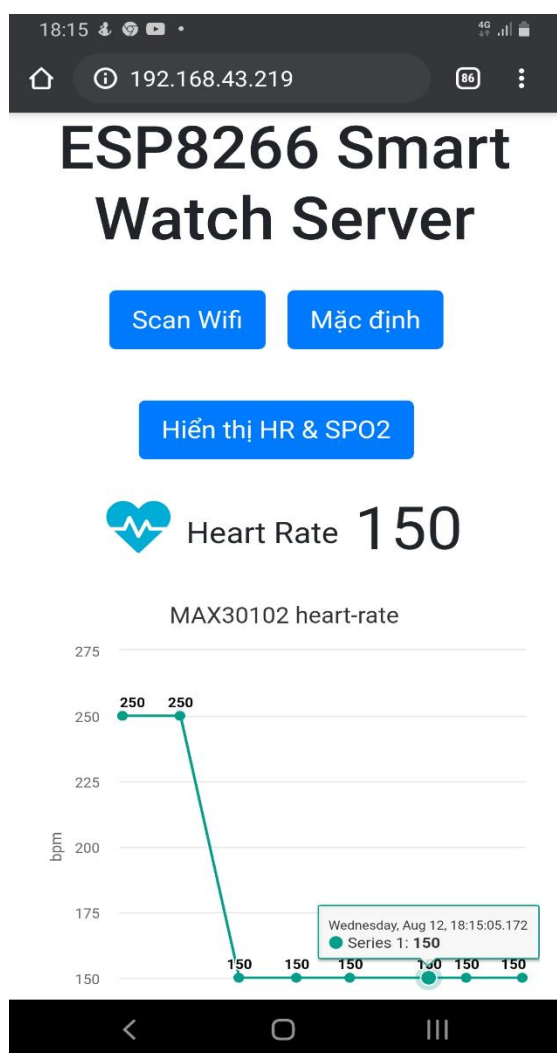
#### 5.1.1. Hình ảnh thực tế của sản phẩm:



Hình 5.1: Hình ảnh thực tế sản phẩm



Hình 5.2 : Hình ảnh thực tế sản phẩm



Hình 5.3 : Hình ảnh giao diện web

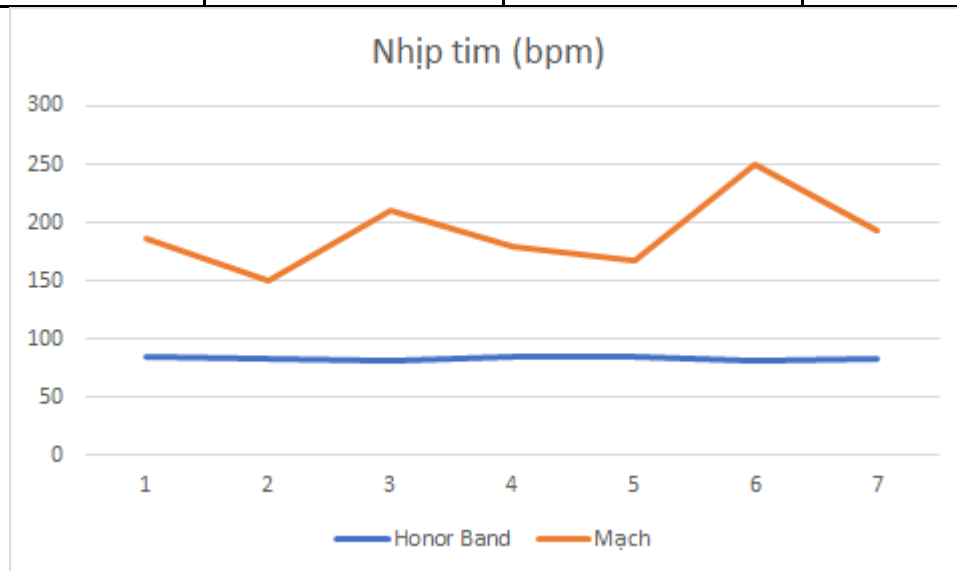
### 5.1.2. Kết quả thực nghiệm:

So sánh kết quả đo thực tế của mạch thí công với thiết bị đo đáng tin cậy hơn. Thiết bị được sử dụng để so sánh kết quả trong đề tài này là vòng đeo tay thông minh Honor Band.

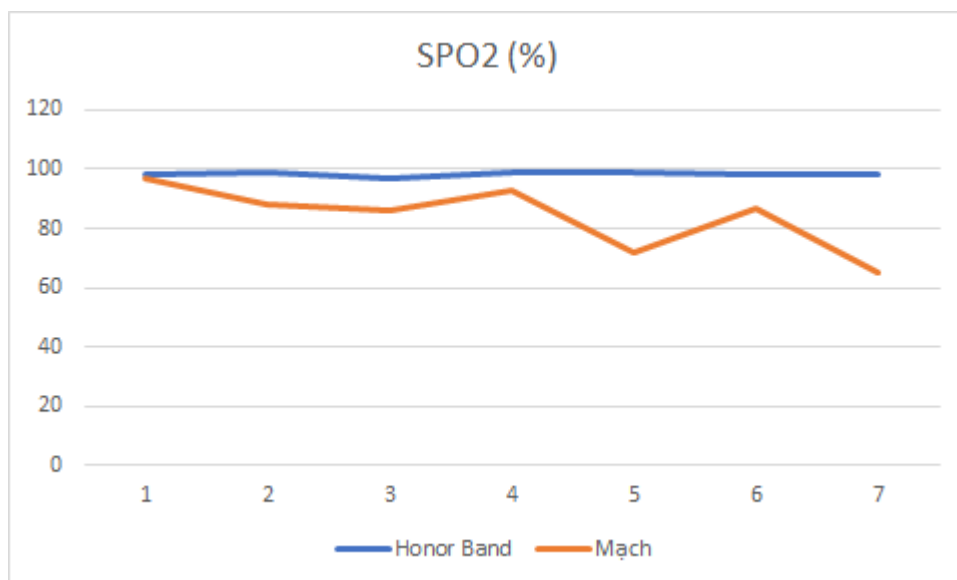
Bảng 5.1: Kết quả thử nghiệm thiết bị

Nhịp tim (bpm)		Nồng độ oxy trong máu (%)	
Honor Band	Mạch	Honor Band	Mạch
84	187	98	97
83	150	99	88

81	210	97	86
85	179	99	93
84	168	99	72
81	250	98	87
83	193	98	65



Hình 5.4: Biểu đồ đường so sánh nhịp tim giữa mạch thi công và Honor Band



Hình 5.5: Biểu đồ đường so sánh SPO2 giữa mạch thi công và Honor Band

### 5.1.3. Kết luận:

Sau khi thực hiện đề tài thì cá nhân đã hoàn thành các nội dung sau:

*Về kiến thức:*

- Có thêm kiến thức, hiểu biết về các vi điều khiển Nodemcu esp8266
- Biết được cách giao tiếp giữa Nodemcu esp8266 với các thiết bị ngoại vi.
- Mở rộng kiến thức về lập trình trên phần mềm Arduino IDE.
- Tìm hiểu về cảm biến max30102, Webserver, oled ssd1306.

*Về sản phẩm:*

- Sản phẩm hoạt động đo nhịp tim và Spo2 chưa ổn định, đạt yêu cầu về hiển thị thông tin bao gồm ký tự, chữ, số cần thông báo trên oled và webserver.
- Tuy nhiên do đây là lần đầu tiếp xúc với vi điều khiển Module Wifi , nguồn tài liệu tham khảo chủ yếu từ nước ngoài, cũng như sự hạn chế về kiến thức của bản thân nên trong quá trình thực hiện vẫn còn một số hạn chế như sau:
  - Có một số ý trong chương trình chưa hoàn toàn hiểu sâu, hiểu rõ.
  - Chưa khai thác được hết tính năng của Arduino.
  - Người dùng chỉ có thể thay đổi thông tin mới bằng cách nạp lại dữ liệu mới cho Arduino chứ chưa có giao diện nhập dữ liệu.
  - Tính thẩm mỹ và ứng dụng chưa cao .

### 5.2. Hướng phát triển

- Hoàn thiện thêm về khả năng di động:
  - Mạch nguồn dùng pin
- Hoàn thiện thêm về tính năng:
  - Cải thiện độ chính xác của sản phẩm
  - Điều khiển từ xa các thiết bị thông minh
  - Phát hiện té ngã + gọi khẩn cấp
  - Định vị
- Đưa thiết bị sang mạng toàn cầu
  - Khả năng truy cập vào trang web trên toàn thế giới
- Tối ưu trang web
  - Xây dựng cơ sở dữ liệu

- Giao diện thân thiện dễ dùng
- Thiết bị theo dõi sức khỏe từ xa:
  - Bác sĩ có thể theo dõi sức khỏe bệnh nhân dễ dàng, thường xuyên và liên tục
  - Người dùng không cần phải đi ra bệnh viện hằng ngày
  - Người thân có thể theo dõi sức khỏe người dùng từ xa.

# TÀI LIỆU THAM KHẢO

- [1] datasheet ASM1117, “AMS1117 - Advanced Monolithic Systems”
- [2] Nguyễn Đình Phú, Vi xử lý - PIC, Đại Học Sư Phạm Kỹ Thuật TP HCM, 2017.
- [3] Nguyễn Thanh Hoàn, Nguyễn Hoàng Nam “Thiết kế vòng tay đo nhịp tim sử dụng công nghệ iots”, Đồ án tốt nghiệp, Trường ĐHSPKT, TP.HCM, 2019.
- [4] datasheet MAX30102, “High Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health”.
- [5] Recommended Configurations and Operating Profiles for MAX30101/MAX30102 EV Kits, Maxim integrated.
- [6] datasheetsSSD1306, “128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller”
- [7] datasheet ESP8266-12E,” ESP-12E WiFi Module Version1.0”.
- [8] <https://benhvienducgiang.com/thong-tin-chuyen-mon/ky-thuat-do-va-theo-doi-spo2>
- [9] <https://hshop.vn>
- [10] <https://www.oled-info.com/oled-technology>
- [11] <https://cuongquach.com/ntp-sever-la-gi-tong-quan-ve-dich-vu-network-time-protocol.html>
- [12] <https://viettuts.vn/lap-trinh-mang-voi-java/giao-thuc-udp>
- [13] <https://www.w3schools.com>
- [14] <https://developer.mozilla.org>