HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI TẬP LỚN

CÁC GIẢI PHÁP VÀ HỆ THỐNG IOT TIÊN TIẾN

ĐỀ TÀI: IOT TRONG Y TẾ VÀ SẢN PHẨM GIÁM SÁT NHỊP TIM VÀ NỒNG ĐỘ O2 TRONG MÁU

Giảng viên: Lê Hải Châu

Nhóm: 03

Nhóm báo cáo: 11

Họ và tên:	Trần Thanh Hoàng:	B19DCVT160
	Ngô Quang Thái:	B19DCVT368
	Lê Đức Trung:	B19DCVT416
	Hoàng Văn Cơ:	B19DCVT032
	Nguyễn Thành Nam:	B19DCVT269

Hà Nội - 4/2023

Mục Lục

I CƠ SỞ LÝ THUYẾT

1.1 Tổng quan về IoT trong y tế

Internet of Things (IoT) là một công nghệ mới đang được áp dụng rộng rãi trong nhiều lĩnh vực, bao gồm cả y tế. IoT trong y tế có thể được sử dụng để giải quyết nhiều vấn đề liên quan đến sức khỏe của con người, bao gồm giám sát sức khỏe cá nhân, đẩy nhanh quá trình chẩn đoán và điều trị, giảm thiểu sai sót trong quá trình chăm sóc bệnh nhân, và cải thiện chất lượng cuộc sống.

Dưới đây là một số ứng dụng IoT trong y tế:

1.1.1 Theo dõi sức khỏe cá nhân:

Các thiết bị IoT như đồng hồ thông minh, vòng đeo tay thông minh, cảm biến sức khỏe, có thể được sử dụng để theo dõi các chỉ số sức khỏe của cá nhân, bao gồm nhịp tim, đường huyết, áp suất máu và nhiệt độ cơ thể.

1.1.2 Theo dõi bệnh nhân từ xa:

Các thiết bị IoT có thể được sử dụng để theo dõi các thông tin sức khỏe của bệnh nhân từ xa, đồng thời cung cấp các lời khuyên về sức khỏe và giúp đưa ra quyết định chẩn đoán và điều trị.

1.1.3 Giám sát quá trình chẩn đoán và điều trị:

Các thiết bị IoT có thể được sử dụng để giám sát quá trình chẩn đoán và điều trị bệnh, từ việc theo dõi dữ liệu y tế cho đến việc cung cấp các lời khuyên về sức khỏe và giúp giảm thiểu sai sót trong quá trình chăm sóc bệnh nhân.

1.1.4 Quản lý thuốc:

Các thiết bị IoT có thể được sử dụng để theo dõi việc sử dụng thuốc của bệnh nhân, giúp đưa ra lời khuyên về liều lượng và giúp người dùng nhớ lịch uống thuốc.

1.1.5 Quản lý thiết bị y tế:

Các thiết bị IoT có thể được sử dụng để quản lý các thiết bị y tế, giúp đảm bảo chúng được sử dụng đúng cách và đủ sức mạnh để phục vụ cho các quy trình chăm sóc bệnh nhân.

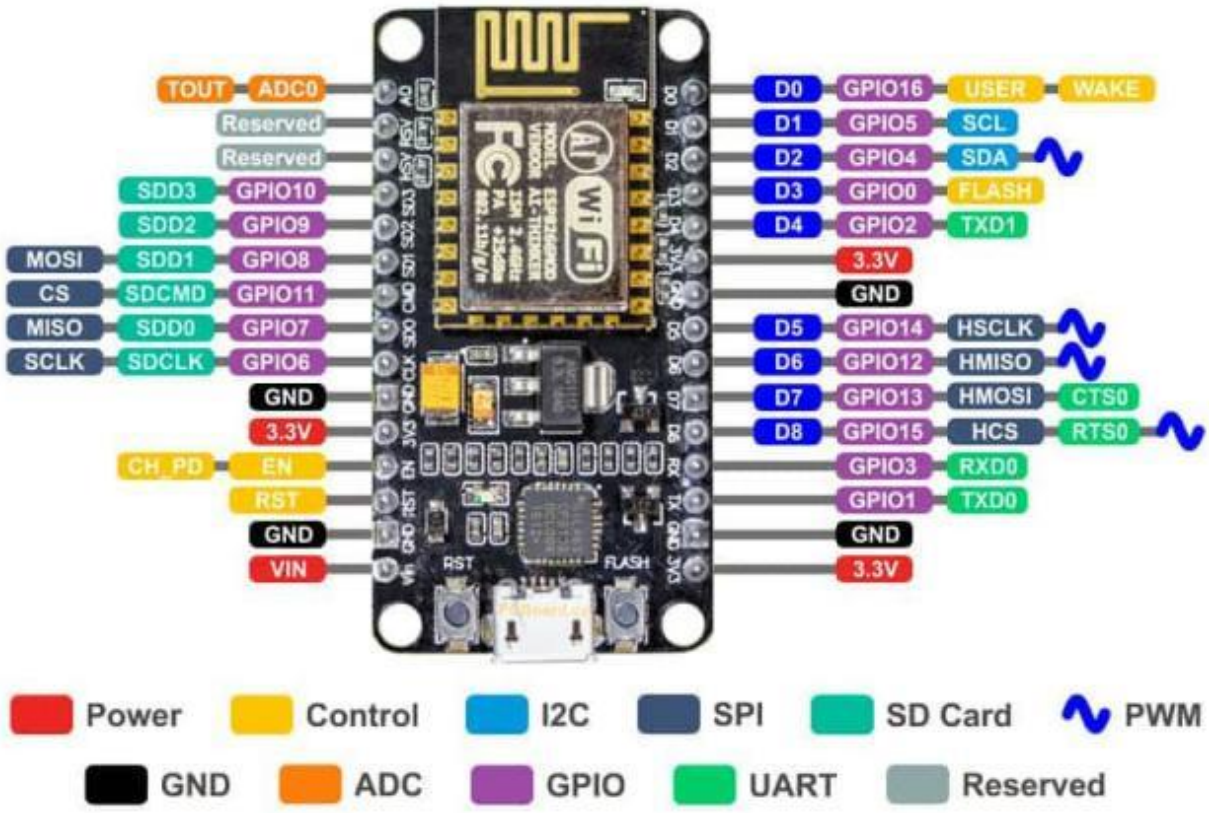
1.2 NodeMCU ESP8266

1.2.1 Giới thiệu về NodeMCU ESP8266

NodeMCU là một phần mềm nguồn mở dựa trên Lua và bảng phát triển được nhắm mục tiêu đặc biệt cho các Ứng dụng dựa trên IoT. Nó bao gồm phần sụn chạy trên ESP8266 Wi-Fi SoC của Espressif Systems và phần cứng dựa trên mô-đun ESP-12.

1.2.2 Cấu tạo của NODEMCU ESP8266

Các module ESP8266 được sử dụng rộng rãi nhất là ESP-01, ESP8266 NodeMCU (ESP8266-12E) và Wemos D1 Mini. Hình bên dưới cho thấy sơ đồ chân của Kit NodeMCU ESP8266-12E. Nếu bạn đang sử dụng những module khác, các bạn có thể dễ dàng lên google để tìm sơ đồ chân.



Thông số kỹ thuật

- WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
- Điện áp hoạt động: 5VDC thông qua cổng micro 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, Lua

1.2.3 Tính năng của NODEMCU ESP8266

ESP8266 có thể được dùng làm module Wifi bên ngoài, sử dụng firmware tập lệnh AT tiêu chuẩn bằng cách kết nối nó với bất kỳ bộ vi điều khiển nào sử dụng UART nối tiếp hoặc trực tiếp làm bộ vi điều khiển hỗ trợ Wifi, bằng cách lập trình một chương trình cơ sở mới sử dụng SDK được cung cấp.

Các chân GPIO cho phép IO Analog và Digital, cộng với PWM, SPI, I2C, v.v.

ESP8266 có nhiều ứng dụng khi nói đến IoT. Đây chỉ là một số chức năng mà chip này được sử dụng

Kết nối mạng: Ăng-ten Wi-Fi của module cho phép các thiết bị nhúng kết nối với bộ định tuyến và truyền dữ liệu

Xử lý dữ liệu: Bao gồm xử lý đầu vào cơ bản từ cảm biến analog và kỹ thuật số để tính toán phức tạp hơn nhiều với RTOS hoặc SDK không phải hệ điều hành

Kết nối P2P: Tạo giao tiếp trực tiếp giữa các ESP và các thiết bị khác bằng kết nối IoT P2P

Máy chủ Web: Truy cập các trang được viết bằng HTML hoặc ngôn ngữ phát triển.

1.3 Giới thiệu về Max30100, OLED 0.96

1.3.1 Giới thiệu về Max30100

MAX30100 là cảm biến đa năng được sử dụng cho nhiều ứng dụng. Là cảm biến theo dõi nhịp tim và cũng là máy đo oxy. Cảm biến có hai diode phát sáng, một cảm biến quang (photodetector) và các linh kiện xử lý tín hiệu để phát hiện nhịp tim và đo xung oxy.

1.3.2 Nguyên lý hoạt động máy đo oxy MAX30100

Cảm biến có một cặp diode phát quang phát ra ánh sáng đỏ đơn sắc có bước sóng 660nm và ánh sáng hồng ngoại có bước sóng 940nm. Các bước sóng này đặc biệt được chọn vì ở bước sóng này, giữa hemoglobin được oxy hóa (Oxygenated Hb) và khử oxy (Deoxygenated Hb) có các đặc tính hấp thụ bước sóng rất khác nhau.

Như biểu đồ dưới đây, có thể thấy sự khác biệt giữa HbO2 (Oxygenated Hb) và Hb (Deoxygenated Hb) khi hấp thụ các bước sóng này.

1.3.3 Giới thiệu về OLED 0.96

Màn hình Oled 0.96 inch giao tiếp I2C cho khả năng hiển thị đẹp, sang 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.

1.3.4 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: 0.96 inch
- Màu hiển thị: Trắng / Xanh Dương.
- Giao tiếp: I2C
- Driver: SSD1306

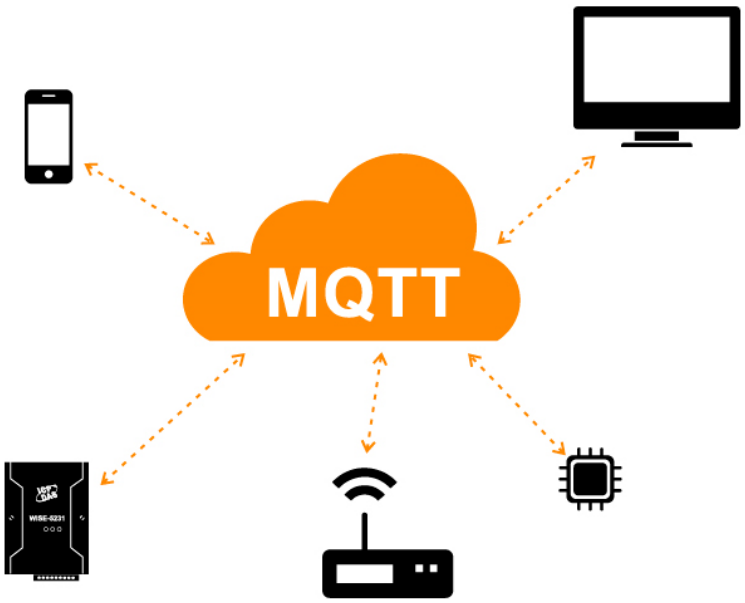
1.4. Giới thiệu về MQTT

1.4.1 Tổng quan về MQTT

MQTT là một giao thức truyền thông điệp theo mô hình Cung cấp/Thuê bao (Publish/Subscribe). Giao thức này được sử dụng cho các thiết bị IoT với băng thông thấp, khả năng sử dụng được trong mạng lưới không ổn định và độ tin cậy cao. MQTT dựa trên một máy chủ môi giới (Broker) khá ít xử lý và được thiết kế có tính mở, đơn giản, dễ cài đặt và không đặc trưng cho ứng dụng cụ thể nào.

MQTT là lựa chọn phù hợp trong các môi trường như:

- Chi phí mạng viễn thông đắt đỏ/ băng thông thấp, thiếu tin cậy.
- Thiết bị nhúng bị giới hạn về bộ nhớ và tốc độ tài nguyên.
- Các ứng dụng M2M (Machine to Machine) do giao thức MQTT sử dụng băng thông thấp trong môi trường có độ trễ cao.
- Facebook Messenger để truyền thông điệp.



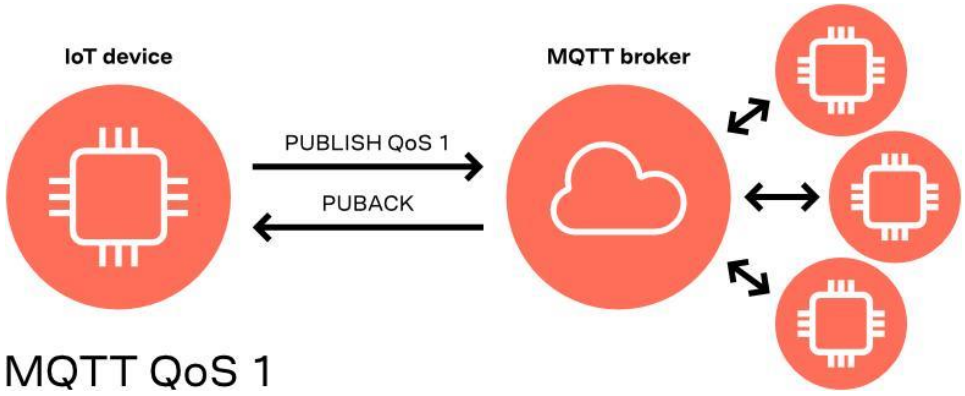
Hình 1.4.1: Giao thức truyền thông điệp theo mô hình publisher/subcriber

MQTT có một số tính năng, đặc điểm nổi bật như:

- Cung cấp khả năng truyền tin phân tán một chiều, tách biệt với phần ứng dụng nhờ dạng truyền thông điệp theo mô hình Pub/Sub.
- Truyền thông điệp ngay lập tức mà không quan tâm tới nội dung được truyền đi.
- Sử dụng giao thức nền là TCP/IP.
- Tồn tại ba mức độ tin cậy cho việc truyền dữ liệu lần lượt là:
  - **QoS 0:** Ở mức độ này, Broker/Client sẽ gửi dữ liệu đúng một lần, quá trình gửi được xác nhận duy nhất bởi giao thức TCP/IP.

- **QoS 1:** Broker/Client sẽ gửi dữ liệu với ít nhất một lần xác nhận từ phía bên kia (tức là có thể nhiều hơn một lần xác nhận rằng đã nhận được dữ liệu).
- **QoS 2:** Broker/Client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được duy nhất một lần và quá trình này phải trải qua đủ 4 bước bắt tay.

– Để giảm tải cho đường truyền, phần bao bọc dữ liệu truyền nhỏ và được giảm tải đến mức tối đa.



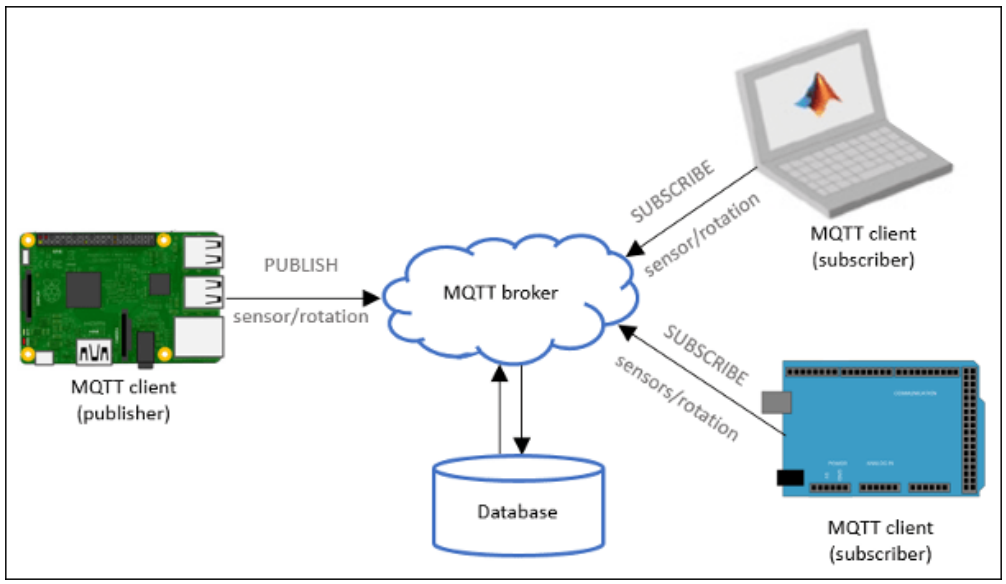
Hình 1.4.2: MQTT truyền thông điệp mà không quan tâm tới nội dung được truyền đi

- Về đặc điểm riêng:

- **MQTT Subject-based:** MQTT sử dụng cơ chế lọc thông điệp dựa vào tiêu đề.
- **MQTT Quality of Services-QoS:** Đây được gọi là tầng chất lượng dịch vụ, hỗ trợ nhận biết các message có được truyền thành công hay không một cách dễ dàng.

Về cơ chế hoạt động, MQTT hoạt động theo cơ chế Client/ Server nơi mà mỗi cảm biến là một Client và được kết nối tới máy chủ. Nó có thể như một máy chủ môi giới và thông qua giao thức **TCP** (Transmission Control Protocol). Broker chịu trách nhiệm điều phối toàn bộ thông điệp giữa phía gửi đến đúng phía nhận.

MQTT là một giao thức định hướng bản tin mà trong đó, mỗi bản tin mà một đoạn rời rạc của tín hiệu và broker không thể nhìn thấy. Các bản tin được publish một địa chỉ tương ứng và có thể hiểu như một kênh (topic). Client đăng ký vào một hoặc nhiều kênh để nhận và gửi dữ liệu. Mỗi Client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu vào kênh đã đăng ký (bản tin được gửi tới một kênh nào đó gọi là publish).



Hình 1.4.3: MQTT hoạt động theo cơ chế Client/ Server

1.4.2. Ưu, nhược điểm nổi bật của MQTT

Ưu điểm:

Kiến trúc gọn nhẹ của giao thức MQTT giúp việc truyền dữ liệu trơn tru với băng thông thấp và giảm tải cho CPU và RAM. MQTT có nhiều lợi thế hơn so với các giao thức khác như:

- Triển khai giao thức một cách nhanh chóng và dễ dàng, ngay cả trên các thiết bị có CPU năng lượng thấp và RAM thấp
- MQTT là một giao thức gọn nhẹ, mức tiêu thụ điện năng ít hơn
- Phân phối dữ liệu hiệu quả
- MQTT sử dụng gói có kích thước thấp và do đó có thể được sử dụng cho các ứng dụng băng thông thấp.
- MQTT có thể mở rộng do mô hình Publish/Subscribe

Nhược điểm

Không thể phủ nhận rằng giao thức MQTT có rất nhiều ưu điểm hữu ích, đặc biệt là khi kết hợp với các giao thức HTTP. Tuy nhiên, MQTT không phải là giải pháp hoàn hảo bởi có một vài nhược điểm:

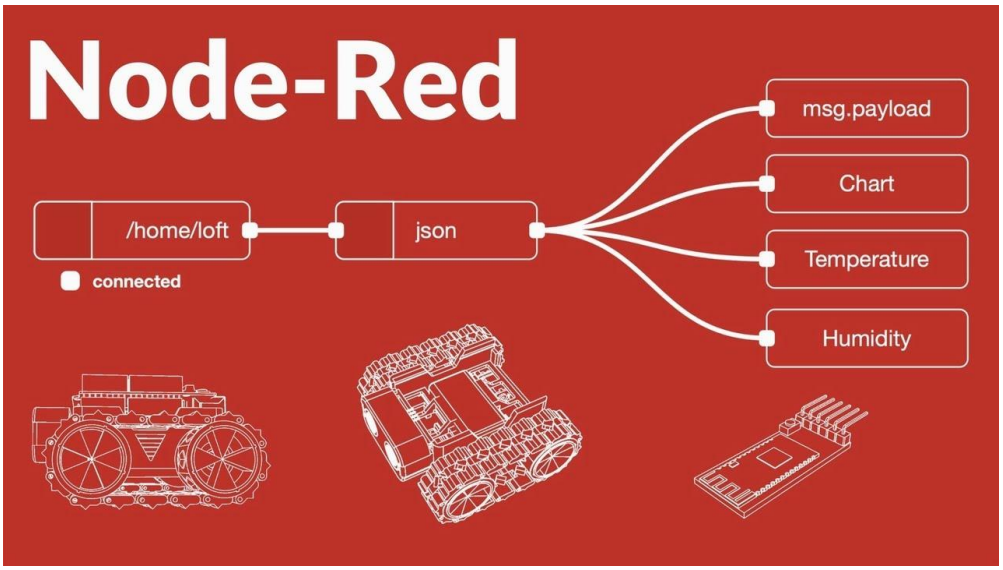
- MQTT sử dụng giao thức TCP đòi hỏi nhiều sức mạnh xử lý hơn và nhiều bộ nhớ hơn
- So với CoAP, MQTT có chu kỳ truyền chậm hơn
- Không hỗ trợ các tính năng nâng cao như flow control
- Trong giao thức MQTT, máy khách phải hỗ trợ TCP/IP
- MQTT không thể hỗ trợ phát trực tuyến video
- Không thân thiện với nhà phát triển
- Thiếu mã hoá bảo mật. Thay vào đó, MQTT sử dụng TLS/SSL
- Việc tạo ra một mạng MQTT có thể mở rộng toàn cầu là rất khó khăn.

1.5 Giới thiệu về Node-red

1.5.1 Tổng quan về node-red

Node-RED là một công cụ phát triển ứng dụng IoT được xây dựng trên nền tảng Node.js và được phát triển bởi IBM. Công cụ này cho phép người dùng kết nối các thiết bị và dịch vụ khác nhau để tạo ra các luồng dữ liệu tự động. Node-RED được cung cấp hoàn toàn miễn phí và là một dự án mã nguồn mở, cung cấp cho người dùng khả năng mở rộng tùy ý.

Node-RED được phát triển bởi IBM tại trung tâm nghiên cứu của họ tại Cambridge, Anh. Node-RED ban đầu được giới thiệu vào năm 2013 và đã nhanh chóng trở thành một công cụ phát triển IoT phổ biến. Node-RED được cung cấp theo giấy phép Apache 2.0, cho phép người dùng sử dụng, sao chép, sửa đổi và phân phối mã nguồn mở.



Hình 1.5.1: Tổng quan về node-red

Node-RED được sử dụng rộng rãi bởi các nhà phát triển IoT và các doanh nghiệp trên toàn thế giới. Nó cũng được sử dụng trong các dự án mã nguồn mở và cộng đồng Node-RED ngày càng lớn mạnh.

Node-RED cũng đã được tích hợp với các nền tảng IoT lớn khác như Amazon Web Services, Microsoft Azure và Google Cloud Platform. Điều này cho phép người dùng sử dụng Node-RED để tạo các ứng dụng IoT và tích hợp chúng với các nền tảng cloud khác nhau.

1.5.2 Các tính năng của node-red

- *Giao diện đồ họa dễ sử dụng*:

  Node-RED cung cấp một giao diện đồ họa dễ sử dụng để tạo và quản lý các luồng dữ liệu. Với giao diện trực quan, người dùng có thể kéo và thả các nút và kết nối chúng với nhau để tạo thành các luồng dữ liệu.

- *Thư viện các nút tiêu chuẩn:*

  Node-RED cung cấp thư viện các nút tiêu chuẩn để kết nối với các thiết bị và dịch vụ khác nhau, bao gồm các dịch vụ như Twitter, Facebook và SQL databases. Với các nút tiêu chuẩn này, người dùng có thể tạo các luồng dữ liệu đầy đủ chỉ bằng cách kéo và thả chúng vào các luồng.

- *Tích hợp với các nút và thư viện từ các nhà phát triển bên ngoài:*

  Node-RED cũng hỗ trợ tích hợp với các nút và thư viện từ các nhà phát triển bên ngoài. Điều này cho phép các nhà phát triển tùy chỉnh Node-RED để phù hợp với nhu cầu cụ thể của họ và tạo ra các tính năng mới cho Node-RED.

- *Công cụ dự đoán và phân tích dữ liệu*

  Node-RED cung cấp công cụ dự đoán và phân tích dữ liệu để giúp người dùng phân tích dữ liệu thu thập được từ các thiết bị IoT và đưa ra các quyết định dựa trên dữ liệu này. Công cụ này cho phép người dùng tạo các đường cong dự đoán và biểu đồ để hiển thị các dữ liệu thu thập được.

- *Tích hợp với các nền tảng IoT*

  Node-RED hỗ trợ tích hợp với các nền tảng IoT như MQTT và Modbus để kết nối với các thiết bị IoT khác. Điều này cho phép người dùng kết nối với các thiết bị IoT khác nhau và lấy dữ liệu từ chúng để thực hiện các tác vụ tự động.

- *Các tính năng bảo mật*

  Node-RED cung cấp các tính năng bảo mật như chứng thực người dùng và mã hóa dữ liệu để giữ cho dữ liệu của người dùng an toàn và bảo mật.

1.5.3 Các ứng dụng của node-red

Node-RED có thể được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm IoT, tự động hóa, giám sát và quản lý dữ liệu. Nó có thể được sử dụng để kết nối các thiết bị khác nhau với nhau và thực hiện các tác vụ tự động hoặc cung cấp dữ liệu cho các hệ thống quản lý dữ liệu khác.

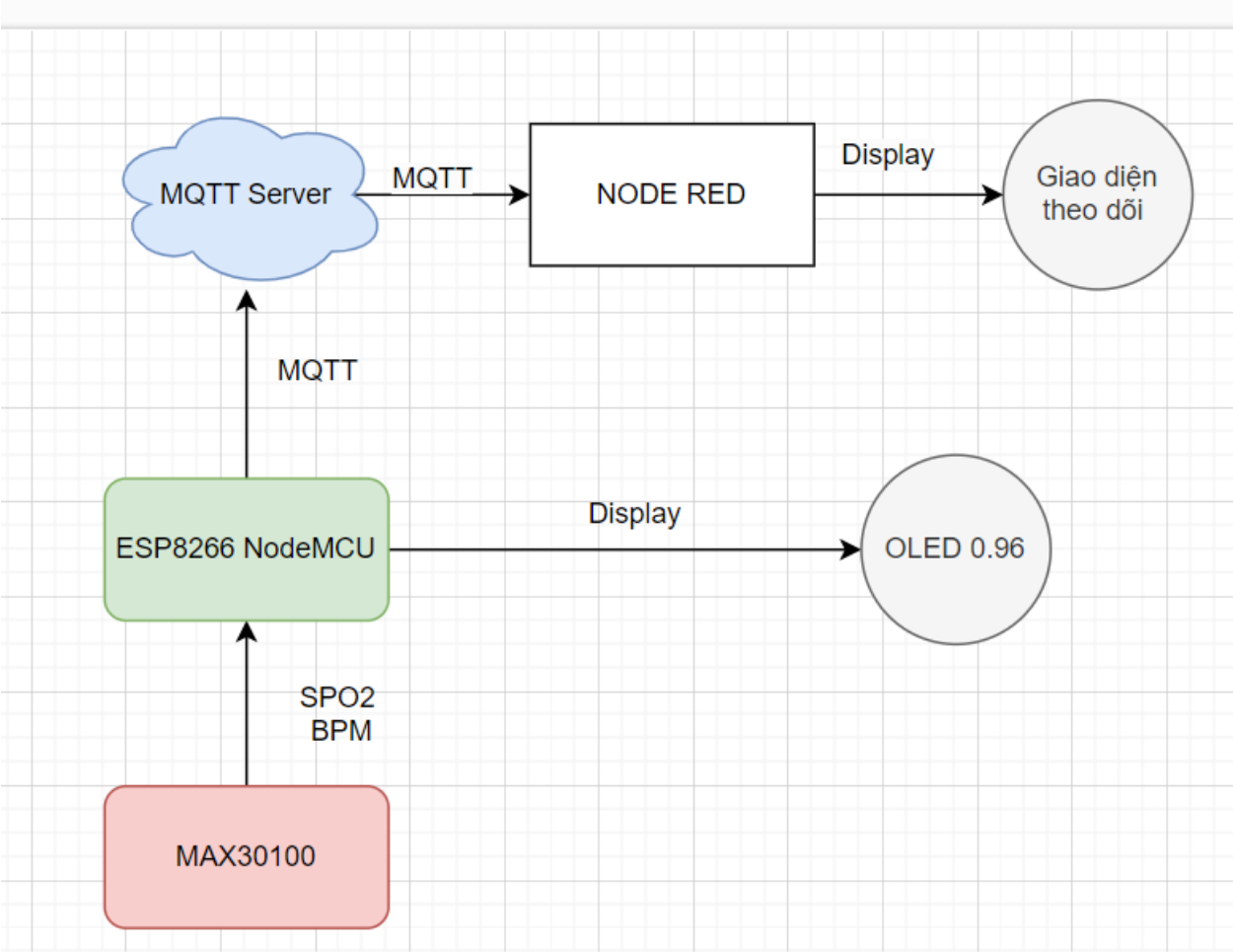
Trong lĩnh vực IoT, Node-RED có thể được sử dụng để tạo các ứng dụng như hệ thống giám sát môi trường, hệ thống quản lý năng lượng và hệ thống quản lý thiết bị. Nó cũng có thể được sử dụng để kiểm soát các thiết bị thông minh trong một nhà thông minh hoặc để giám sát các máy móc trong một nhà máy.

Node-RED cũng có thể được sử dụng để phát triển các ứng dụng web, các ứng dụng di động và các ứng dụng máy tính để bàn. Nó cũng có thể được sử dụng để phát triển các ứng dụng cho các thiết bị nhúng và các hệ thống nhúng.

II THIẾT KẾ MẠCH

2.1 Thiết kế hệ thống

2.1.1 Sơ đồ khối



2.1.2 Chức năng của từng khối

+Cảm biến nhịp tim và oxy trong máu MAX30100

+ Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340: Thu thập dữ liệu, hiển thị lên OLED, có khả năng kết nối internet, gửi các dữ liệu lên Server bằng MQTT

+ Màn hình Oled 0.96 inch giao tiếp I2C

+ Node red: Công cụ thiết kế giao diện, thu thập dữ liệu MQTT server để hiển thị lên giao diện

2.2 Nguyên lý hoạt động

+ Cảm biến nhịp tim MAX30100 thực hiện đo 2 thông số SPO2 và BPM

+ Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 thu thập dữ liệu mà cảm biến đo được và thực hiện điều khiển OLED hiển thị 2 thông số SPO2 và BPM

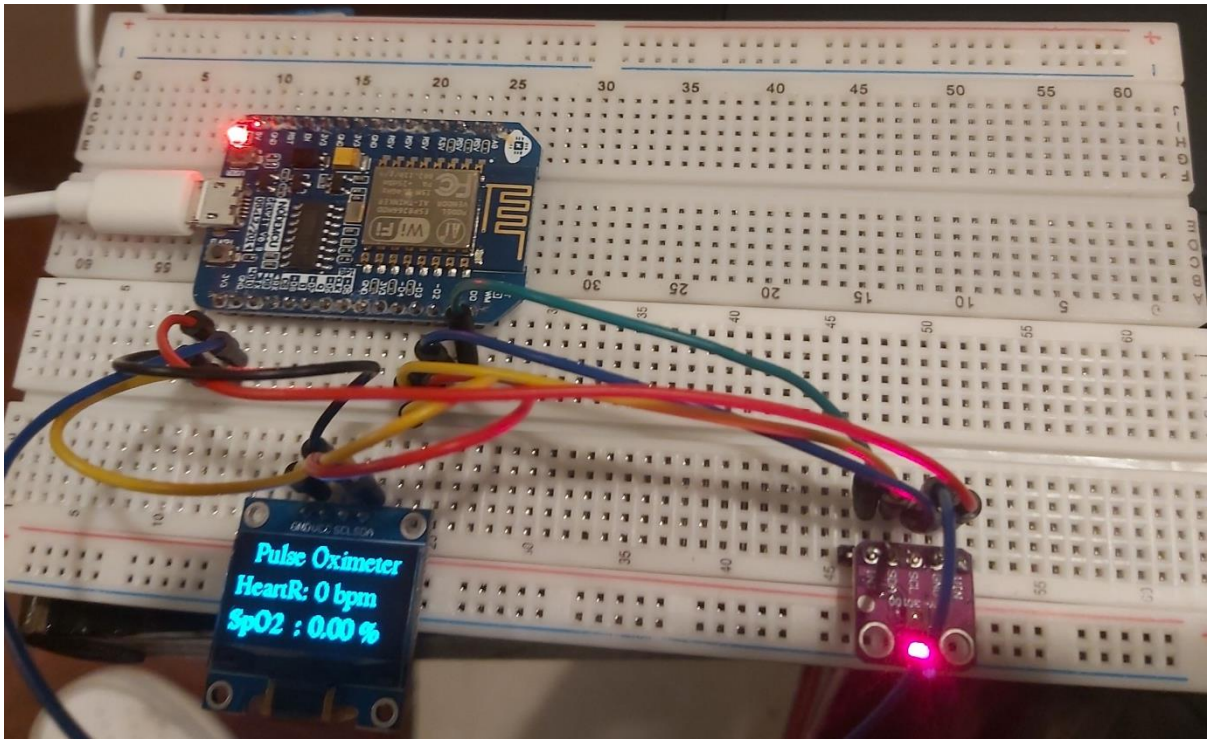
+ Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 kết nối với internet thực hiện gửi 2 Topic bao gồm"SPO2" và "BPM" gửi lên MQTT server

+ Node red là công cụ thiết kế giao diện có thể thực hiện kết nối từ xa lấy dữ liệu của MQTT server để lấy dữ liệu, thực hiện hiển thị, điều khiển thiết bị từ xa.

III THI CÔNG HỆ THỐNG

3.1 Thi công hệ thống

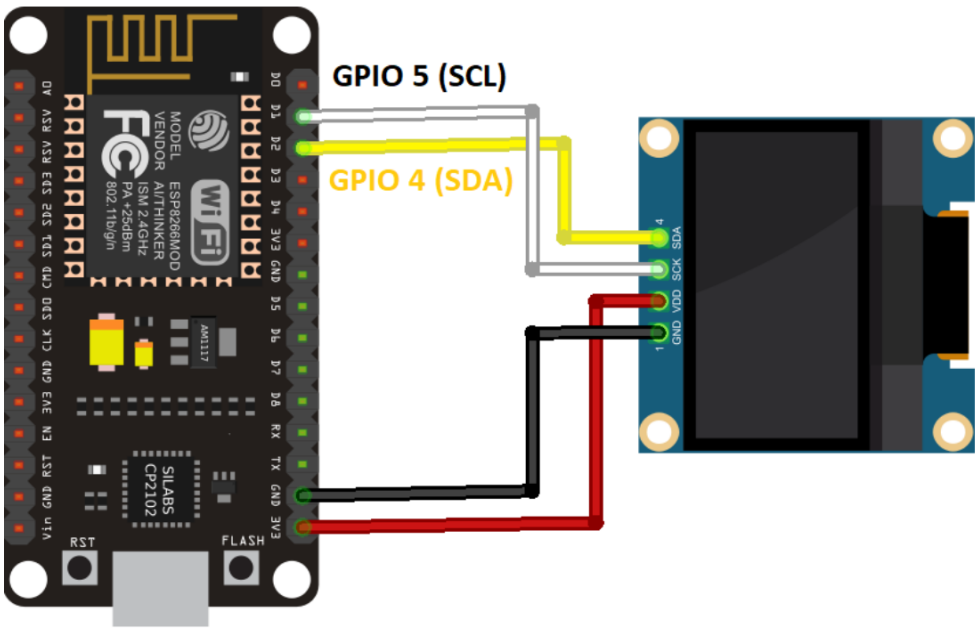
3.1.1 Thực hiện lắp ráp, ghép nối các mạch và Module



Hình 3.1a. Hình ảnh thiết bị

3.1.2 Sơ đồ kết nối từng thiết bị.

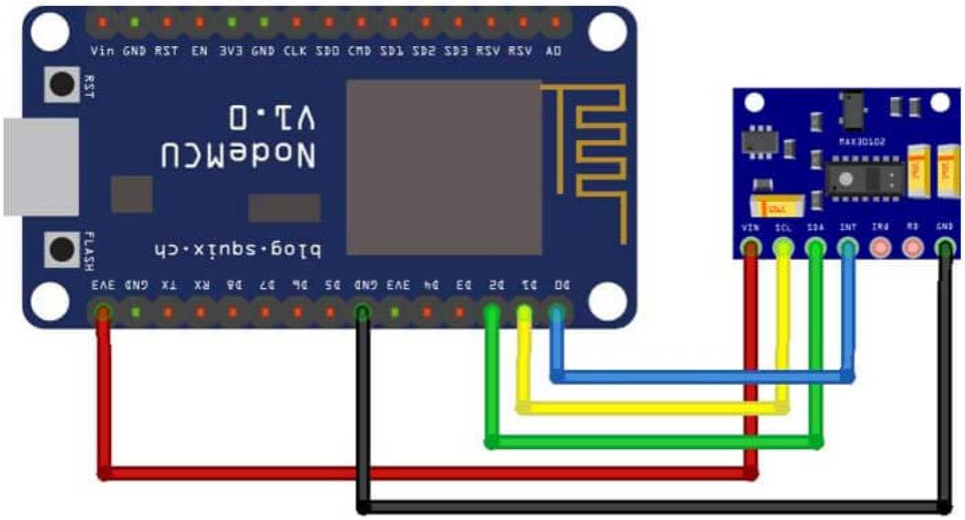
3.1.2a. Sơ đồ kết nối ESP8266 với OLED 0.96



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

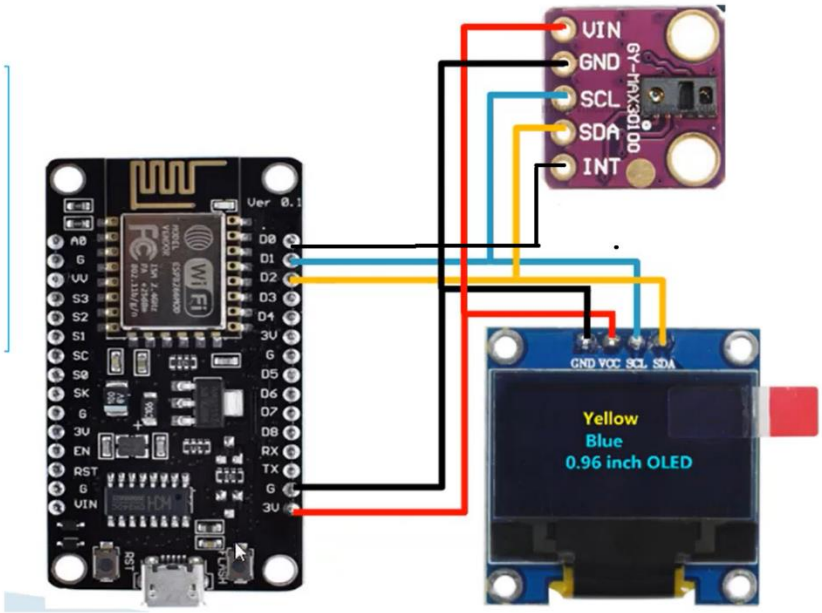
Hình 3.1.2a. Sơ đồ kết nối ESP8266 với OLED

3.1.2b. Sơ đồ kết nối ESP8266 với MAX30100



Hình 3.1.2b. Sơ đồ kết nối ESP8266 với MAX30100

3.1.2c. Sơ đồ tổng quát



Hình 3.1.2c. Sơ đồ tổng quát

3.2 Lập trình hệ thống

3.2.1 Phần mềm lập trình cho vi điều khiển(Arduino IDE)

Arduino IDE (Integrated Development Environment) là một phần mềm được sử dụng để lập trình và nạp chương trình cho các bo mạch điều khiển nhúng (microcontroller) của Arduino. Arduino IDE cung cấp cho người dùng một giao diện đơn giản và dễ sử dụng để viết mã và tải nó lên bo mạch Arduino để thực thi các chức năng cụ thể.

Arduino IDE hỗ trợ đầy đủ việc kết nối và sử dụng module MAX30100 và ESP8266, cùng với giao thức MQTT để truyền tải dữ liệu từ thiết bị đến máy chủ MQTT

3.2.2 Chương trình điều khiển

Bước 1: Các thư viện được sử dụng

```
#include <ESP8266WebServer.h>
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Fonts/FreeSerif9pt7b.h>
Adafruit_SSD1306 display(128, 64, &Wire, -1);

#include <PubSubClient.h>
#include <WiFiClient.h>
```

Bước 2: Khai báo các dữ liệu cần thiết kết nối với MQTT Server

```
// Update these with values suitable for your network.
const char* mqtt_server = "mqtt-dashboard.com";  // MQTT server address
const char* mqtt_user = "hoang2k1"; // MQTT username
const char* mqtt_password = "hoang2k1"; // MQTT password
const char* mqtt_client_name = "esp8266-client"; // MQTT client name
const char* mqtt_topic_spo2 = "SPO2"; // MQTT topic for SpO2
const char* mqtt_topic_bpm = "BPM"; // MQTT topic for BPM
const int mqtt_port = 1883;
WiFiClient espClient;
PubSubClient client(espClient);
```

Bước 3: Khai báo các dữ liệu kết nối Wifi

```
#define REPORTING_PERIOD_MS     1000
float BPM, SpO2;
/*Put your SSID & Password*/
const char* ssid = "TH TRUEMILK";  // Enter SSID here
const char* password = "hoang001";  //Enter hoangPassword here
PulseOximeter pox;
uint32_t tsLastReport = 0;
```

Bước 4: Xây dựng các hàm callback và reconnect để thực hiện kết nối với MQTT Server và gửi các topic

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect(mqtt_client_name, mqtt_user, mqtt_password)) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("esp8266/status", "connected");
      // ... and resubscribe
      client.subscribe("esp8266/receive");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

```
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}
```

Bước 5: Thiết lập setup và loop

Thiết lập setup

```
void setup() {
  Serial.begin(115200);
  pinMode(16, OUTPUT);
  delay(100);
  Serial.println("Connecting to ");
  Serial.println(ssid);
  //connect to your local wi-fi network
  WiFi.begin(ssid, password);

  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: ");  Serial.println(WiFi.localIP());
  Serial.print("Initializing pulse oximeter..");

    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
      Serial.println("SSD1306 allocation failed");
      for(;;);
    }
    client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
  reconnect();
```

```
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("SSD1306 allocation failed");
    for(;;);
  }
  client.setServer(mqtt_server, mqtt_port);
client.setCallback(callback);
reconnect();
  display.setFont(&FreeSerif9pt7b);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(20,15);
  display.println("Welcom to");
  display.setCursor(0,40);
  display.println("TEAM 12");
  display.display();
  display.setTextSize(1);
  delay(2000);

if (!pox.begin()) {
  Serial.println("FAILED");
  for (;;);
} else {
  Serial.println("SUCCESS");
```

Thiết lập loop

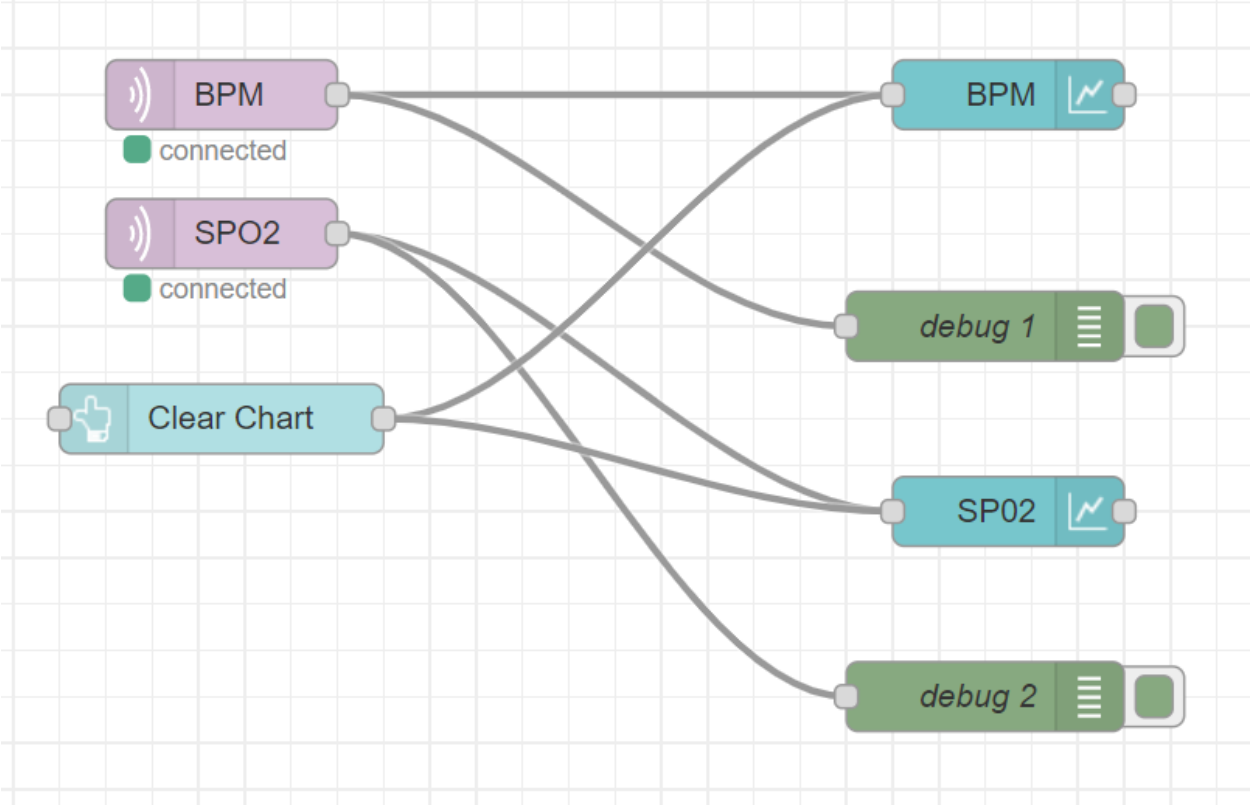
```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  pox.update();
  if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
    BPM = pox.getHeartRate();
    SpO2 = pox.getSpO2();
    Serial.print("BPM: ");
    Serial.println(BPM);
    Serial.print("SpO2: ");
    Serial.print(SpO2);
    Serial.println("%");
    Serial.println("*********************************");

    display.clearDisplay();
    display.setCursor(10,12);
    display.print("Pulse Oximeter");
    display.setCursor(0,35);
    display.print("HeartR:");
    display.setCursor(62,35);
    display.print(BPM,0);
    display.println(" bpm");
    display.setCursor(0,59);
    display.print("SpO2  : ");
    display.setCursor(62,59);
    display.print(SpO2);
    display.println(" %");
    display.display();
    client.publish(mqtt_topic_spo2, String(SpO2).c_str());
    client.publish(mqtt_topic_bpm, String(BPM).c_str());
    tsLastReport = millis();

  }
}
```

3.2.3 Phần mềm thiết kế giao diện theo dõi bện nhân từ xa (Node red)

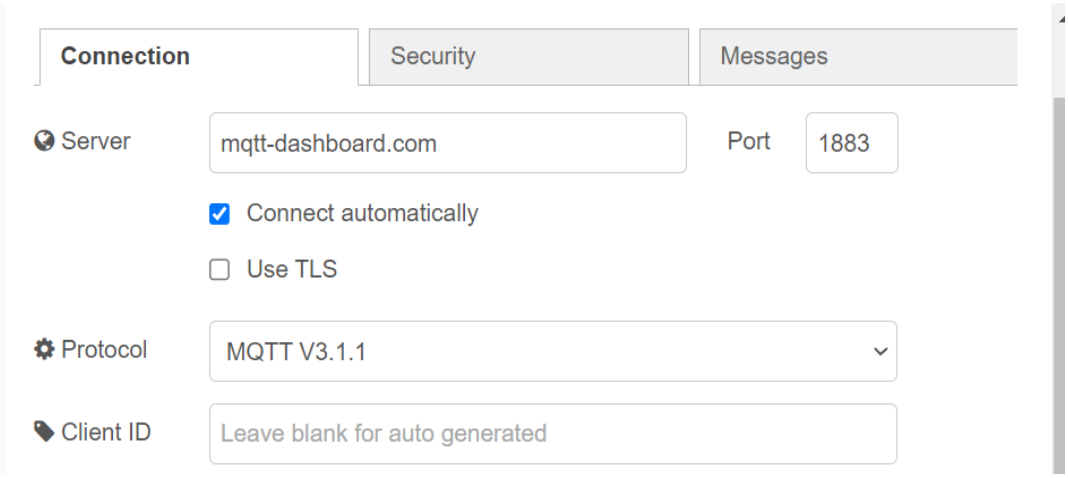
3.2.3a. Thực hiện thiết kế theo các khối chức năng sau.

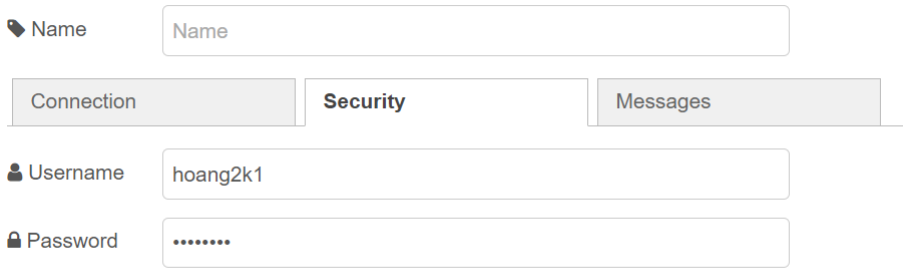


Hình 3.2.3a. Thực hiện thiết kế theo các khối chức năng

3.2.3b. Setup từng khối trên Node red

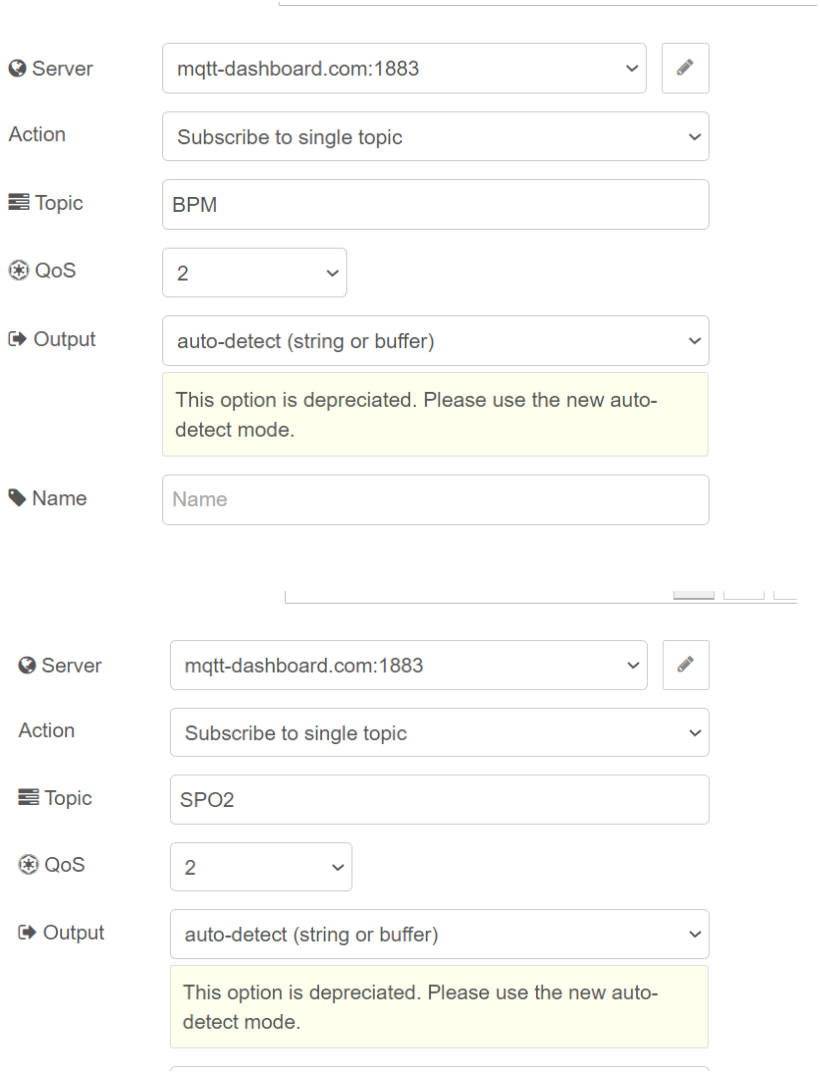
Bước 1: Kết nối Node Red với server MQTT





Hình 3.2.3b Kết nối Node Red với server MQTT

Bước 2: Thiết lập Hai khối MQTT IN: SPO2 và BPM

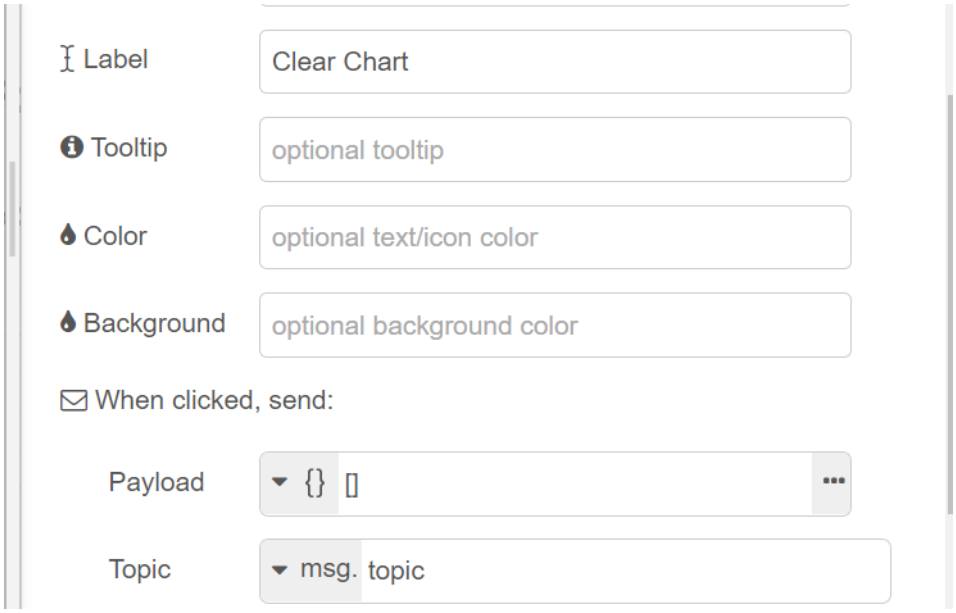


Hình 3.2.3c. Thiết lập Hai khối MQTT IN: SPO2 và BPM sử dụng QoS = 2

Bước 3 : Thiết kế bảng hiển thị thời gian thực BPM và SPO2

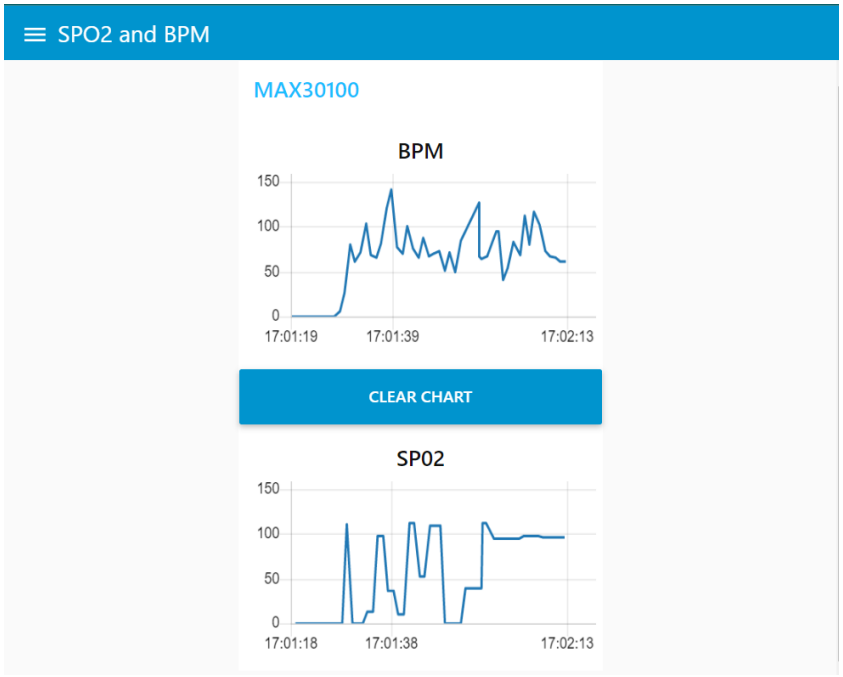
Hình 3.2.3d. Thiết kế bảng hiển thị thời gian thực BPM và SPO2

Bước 5: Xây dựng button Clear Chart có chức năng xóa bảng hiển thị và đo lại



Hình 3.2.3e. Xây dựng button Clear Chart có chức năng xóa bảng hiển thị và đo lại

Bước 6: Thực hiện Deploy và có giao diện như sau



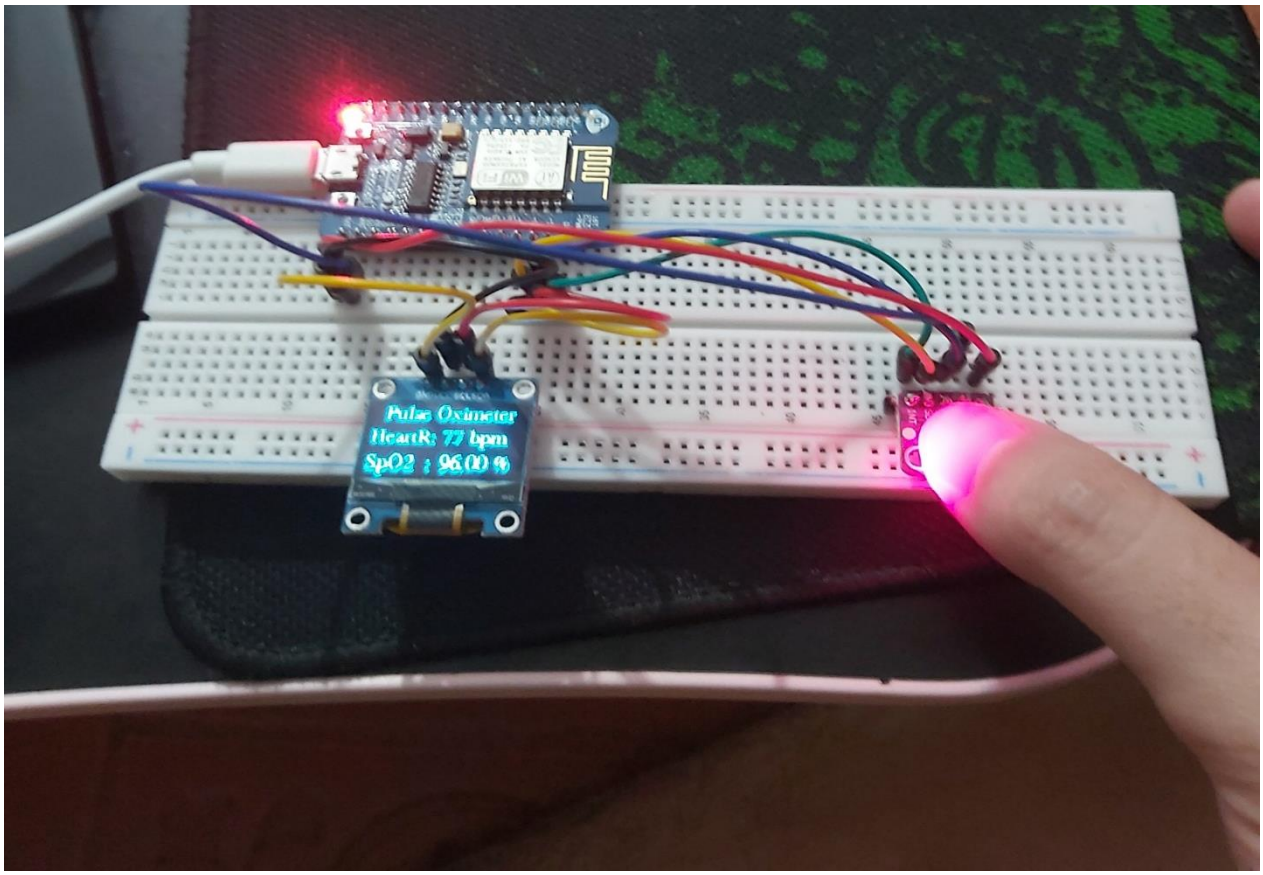
Hình 3.2.3f. Giao diện theo dõi nhịp tim và nồng độ O2 từ xa

IV KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ

4.1 Kết quả

4.1.1 Mô hình hệ thống

Kết quả khi đo trực tiếp trên thiết bị , hiển thị hai thông số HeartR(BPM) và nồng độ O2 (Spo2)

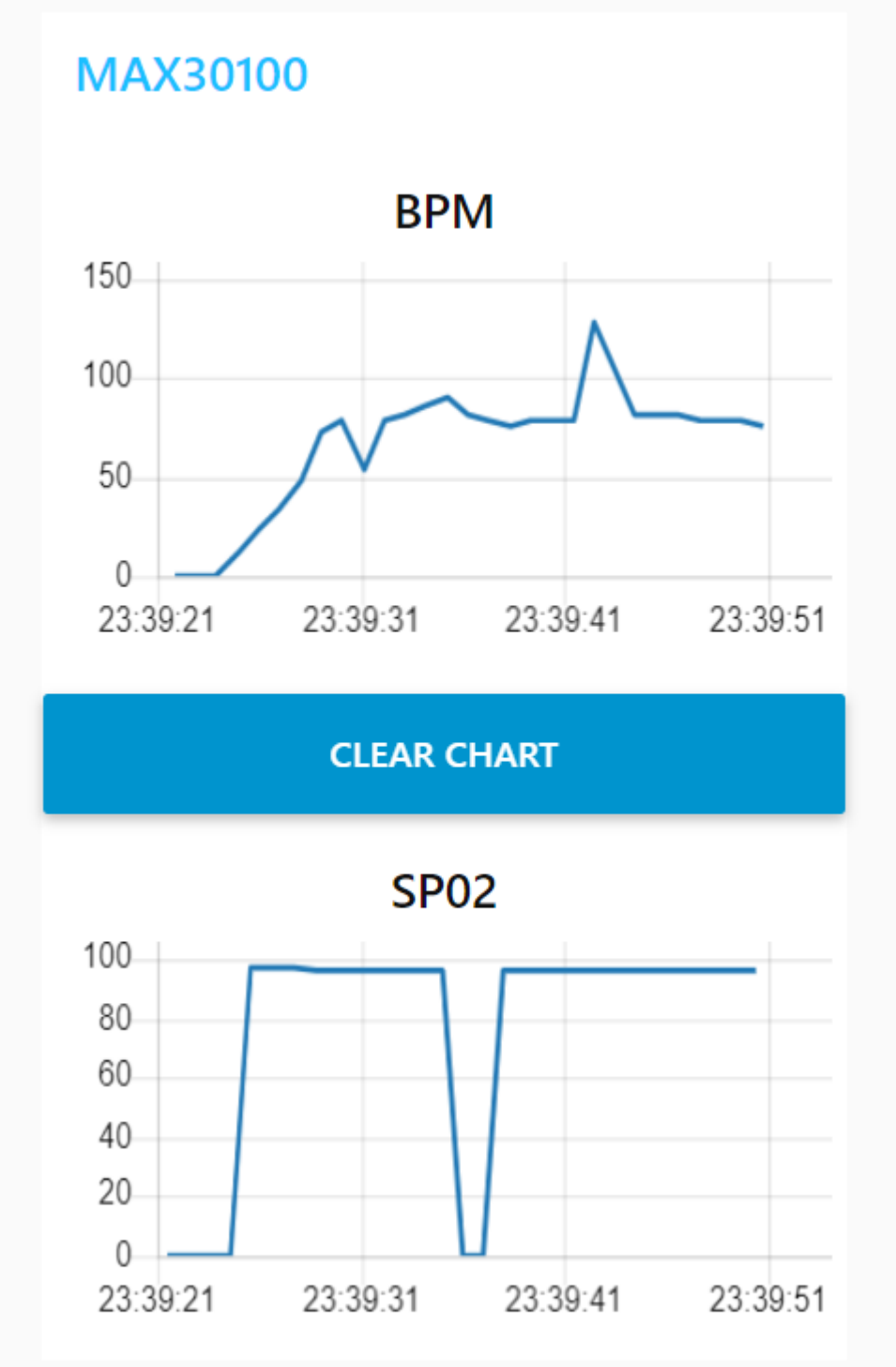


Hình 4.1.1a Kết quả thiết bị

Hình 4.1.1b Kết quả các thông số đo được hiện lên Serial Monitor của Arduino IDE

Hình 4.1.1b Các topic SPO2 và BPM được publish lên server MQTT được quan sát bởi WireShark

4.1.2 Điều khiển và giám sát thiết bị



Hình 4.1.2 Giám sát thiết bị đo thông qua giao diện Node-red đã thiết kế

4.2 Nhận xét và đánh giá

4.2.1 Nhận xét

- Chỉ số đo BPM:

+ Nhịp tim ổn định là khi nhịp tim hoạt động ở một tần số ổn định, không có những dao động lớn hay không đều trong nhịp tim. Tần số nhịp tim ổn định thường dao động trong khoảng từ 60 đến 100 nhịp/phút ở người trưởng thành trong tình trạng nghỉ ngơi.

Các yếu tố có thể ảnh hưởng đến tính ổn định của nhịp tim bao gồm sức khỏe tổng thể, cường độ và thời lượng hoạt động thể chất, mức độ căng thẳng, cảm xúc, chế độ ăn uống, thói quen sinh hoạt, và các yếu tố bên ngoài như thời tiết, môi trường.

Nhịp tim ổn định là một dấu hiệu cho thấy hệ thống tim mạch hoạt động tốt và không gặp vấn đề về sức khỏe. Tuy nhiên, nếu bạn có bất kỳ triệu chứng nào như đau thắt ngực, khó thở, hoa mắt, chóng mặt, hoặc nhịp tim không đều, bạn nên đi khám bác sĩ để được chẩn đoán và điều trị kịp thời.

+ Trong khi tập luyện hoặc hoạt động cường độ cao, nhịp tim của bạn có thể dao động trong khoảng từ 50% đến 85% của tốc độ nhịp tim tối đa được tính bằng cách trừ tuổi của bạn từ số 220. Ví dụ, nếu bạn là là sinh viên 22 tuổi, tốc độ nhịp tim tối đa của bạn là khoảng 198 nhịp/phút (220 - 22). Do đó, khi hoạt động cường độ và căng thẳng, nhịp tim của bạn có thể dao động từ 99 nhịp/phút (50% của 190) đến 166 nhịp/phút (85% của 198).

+ Khi đi ngủ, tốc độ nhịp tim sẽ giảm xuống trong quá trình nghỉ ngơi. Trong khi ngủ, tốc độ nhịp tim bình thường ở người trưởng thành có thể dao động từ 40 đến 60 nhịp/phút

- Chỉ số SPO2:

Thường thì chỉ số SpO2 bình thường nằm trong khoảng từ 95% đến 100%

4.2.2 Đánh giá

4.2.2a Chỉ số BPM

- Chỉ số BPM khá chính xác khi đo khá chính xác tại các thời điểm:

+ Nhịp tim đập mạnh khi hoạt động mạnh và căng thẳng.

+ Nhịp tim bình thường là từ 60-100 bpm.

+ Khi đi ngủ là 40-60 bpm

Tuy nhiên khi sử dụng đồng hồ thông minh để kiểm tra thì có những sai số nhất định.

Nguyên nhân là bởi có thể do chất lượng của cảm biến nhịp tim MAX30100 không đảm bảo.

4.2.2b. Chỉ số SPO2

- Chúng ta có thể thấy trên đồ thị ở hình 4.1.2 thì chất lượng SPO2 khá ổn định ở mức trên 95%. Tuy nhiên có những đoạn bị tụt xuống, điều này có thể có 1 số nguyên nhân:

+ Bộ phận cham vào cảm biến (tay, chân,…) không thực sự tiếp xúc với cảm biến.

+ Cảm biến MAX30100 chưa thực sự đáp ứng chất lượng do giá thành rẻ và dễ sử dụng.

V KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT LUẬN

Qua đề tài "giải pháp theo dõi nhịp tim và nòng độ O2 trong máu" nhóm thực hiện đã nghiên cứu được những vấn đề mà mục tiêu đã đưa ra:

- Cách sử dụng và xây dựng dự án trên board ESP8266 NodeMCU .
- Cách sử dụng cảm biến Max30100 để đo nhịp tim và nồng độ oxy trong máu và sử dụng màn hình oled để hiện thị thông tin.
- Cấu tạo và nguyên lý hoạt động của nhịp tim.
- Thiết kế giao diện theo dõi bện nhân từ xa (Node red).
- Thiết kế là lắp đặt hệ thống.
- Hiển thị thông tin lên web server.

Tuy nhiên, do thời gian và trình độ còn hạn hẹp nên không tránh khỏi những sai sót, hạn chế trong hệ thống như:

- Chưa đảm bảo được sự tiện lợi vì các thiết bị phần cứng còn khá cồng kềnh. Thiết bị vẫn chưa đạt được sự ổn định và chính xác cao.

- Hệ thống chưa có sự cảnh báo đến người sử dụng khi nhịp tim quá nhanh hoặc chậm, nồng độ oxy trong máu thấp hoặc cao.

5.2 HƯỚNG PHÁT TRIỂN

Mặc dù còn nhiều hạn chế nhưng đề tài có thể ứng dụng vào việc đo hoặc theo dõi nhịp tim cho người sử dụng.

Hệ thống về phần cứng nếu được hỗ trợ các IC và module mạnh mẽ hơn thì thiết bị có thể nhỏ gọn và tích hợp nhiều chức năng hơn nữa.

Có thể cải tiến thiết bị trở thành vòng đeo tay, tích hợp sử dụng thêm nhiều cảm biến khác để đo huyết áp, nhiệt độ cơ thể,… để có thêm thông tin sức khỏe toàn diện hơn.

Về phần mềm cần thêm cơ sở dữ liệu được hỗ trợ để Lưu trữ dữ liệu đo được lên đám mây (cloud) để theo dõi sức khỏe lâu dài và nhận biết những thay đổi. kết hợp với trí thông minh nhân tạo (AI) để đưa ra chẩn đoán, cảnh báo bệnh một cách kịp thời và chính xác.

Thêm chức năng gửi cảnh báo qua SMS, email hoặc notification trên ứng dụng di động nếu phát hiện giá trị bất thường của nhịp tim hoặc SpO2.

Kết nối với hệ thống chăm sóc sức khỏe để cảnh báo cho bác sĩ hoặc gia đình nếu cần trợ giúp khẩn cấp.