

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*\*\*

Icon

Description automatically generated

BÁO CÁO ĐỒ ÁN MÔN HỌC

**PHÁT TRIỂN HỆ THỐNG IOT**

**Đề tài:**

**Thiết kế hệ thống cảm biến nhịp tim, định vị GPS**

Nhóm sinh viên: **Nhóm 1**

Danh sách sinh viên: **Vũ Hoàng Giang – 0188166**

**Tạ Huy Hoàng - 0191366**

**Nguyễn Quang An – 0178166**

**Trần Văn Phát – 0199366**

**Nguyễn Đình Tùng – 0208266**

Lớp học phần: **66MHT**

Giảng viên hướng dẫn: **Thầy Lê Đức Quang**

Hà Nội, 5/2025

**MỤC LỤC**

[LỜI NÓI ĐẦU 3](#_Toc199451746)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc199451747)

[**1.1.** Tổng quan về IoT 4](#_Toc199451748)

[**1.1.1.** Lịch sử hình thành 5](#_Toc199451749)

[**1.1.2.** Lợi ích 5](#_Toc199451750)

[1.2 Mục tiêu và phạm vi nghiên cứu của đề tài 6](#_Toc199451751)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 7](#_Toc199451752)

[**2.1 Module NodeMCU ESP8266** 7](#_Toc199451753)

[**2.1.1.** Thông số kỹ thuật 7](#_Toc199451754)

[**2.1.2.** Vi điều khiển 8](#_Toc199451755)

[**2.1.3.** Các chân của ESP8266 8](#_Toc199451756)

[**2.1.4.** Giao thức UART 8](#_Toc199451757)

[**2.2 Tìm hiểu về Thingspeak** 9](#_Toc199451758)

[**2.2.1. Giới thiệu phần mềm Thingspeak** 9](#_Toc199451759)

[**2.2.2 Các trường dữ liệu gửi lên Thingspeak** 9](#_Toc199451760)

[**2.3** **Cảm biến và các module chức năng** 10](#_Toc199451761)

[2.3.1. Mạch định vị GPS ATGM336H 10](#_Toc199451762)

[2.3.2 Module MAX30102 12](#_Toc199451763)

[2.3.2 SSD1306 13](#_Toc199451764)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 14](#_Toc199451765)

[**3.1** **Nguyên lý hoạt động** 14](#_Toc199451766)

[**3.2 Sơ đồ đấu nối** 14](#_Toc199451767)

[**3.3** **Code sản phẩm** 15](#_Toc199451768)

[**3.4** **Thông tin hình ảnh sản phẩm** 21](#_Toc199451769)

[LỜI KẾT 23](#_Toc199451770)

# LỜI NÓI ĐẦU

Trong bối cảnh xã hội ngày càng phát triển, nhu cầu chăm sóc sức khỏe của con người ngày càng được quan tâm đúng mức. Tuy nhiên, ở nước ta, việc áp dụng các thiết bị công nghệ hiện đại để theo dõi sức khỏe cá nhân trong thời gian thực vẫn còn nhiều hạn chế. Nhiều người vẫn chưa có thói quen kiểm tra tình trạng sức khỏe thường xuyên, đặc biệt là các chỉ số sinh tồn quan trọng như nhịp tim. Điều này có thể dẫn đến tình trạng phát hiện bệnh muộn hoặc không kịp thời xử lý các dấu hiệu bất thường của cơ thể.

Trong lĩnh vực y tế, nhịp tim là một trong những chỉ số sinh lý quan trọng nhất, phản ánh trực tiếp tình trạng hoạt động của hệ tim mạch. Việc theo dõi nhịp tim liên tục không chỉ cần thiết đối với người cao tuổi, người có tiền sử bệnh lý tim mạch mà còn rất hữu ích cho những người thường xuyên luyện tập thể thao hoặc làm việc trong môi trường áp lực cao. Trong xu hướng hiện đại hóa và cá nhân hóa việc chăm sóc sức khỏe, các thiết bị đeo thông minh tích hợp cảm biến đo nhịp tim kết nối IoT đang ngày càng được quan tâm và phát triển.

Hệ thống đồng hồ cảm biến nhịp tim kết hợp theo dõi từ xa thông qua điện thoại là một giải pháp hữu ích, giúp người dùng chủ động nắm bắt được tình trạng sức khỏe của bản thân mọi lúc, mọi nơi. Đây là ứng dụng tiêu biểu của công nghệ IoT vào đời sống, góp phần nâng cao chất lượng chăm sóc sức khỏe cá nhân, giảm tải cho các cơ sở y tế, đồng thời hỗ trợ cảnh báo sớm các nguy cơ tiềm ẩn.

Nắm bắt được nhu cầu thiết thực này và mong muốn được áp dụng kiến thức đã học vào thực tế, nhóm chúng em quyết định thực hiện đề tài: **“Xây dựng hệ thống đồng hồ cảm biến nhịp tim và theo dõi từ xa qua điện thoại”** làm đồ án môn *Phát triển hệ thống IoT*.

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

### Tổng quan về IoT

Hands holding a tablet with icons around it

Description automatically generated

*Hình* *1.1.* *Internet* *vạn* *vật*

Internet vạn vật (IoT) là khái niệm kết nối các thiết bị với nhau và với Internet. IoT là một mạng lưới khổng lồ gồm các vật (things) và con người được kết nối - tất cả đều thu thập và chia sẻ dữ liệu với nhau. Việc kết nối có thể thực hiện qua Wifi, Bluetooth…

### Lịch sửA person with a beard Description automatically generated hình thành

*Hình* *1.2.* *Kevin* *Ashton* *(1968)*

Năm [1999](https://vi.wikipedia.org/wiki/1999), Kevin Ashton đã đưa ra cụm từ Internet of Things nhằm để chỉ các đối tượng có thể được nhận biết cũng như sự tồn tại của chúng.

Đến năm 2016, Internet Vạn Vật khẳng định được bước tiến của mình nhờ sự hội tụ của nhiều công nghệ, bao gồm truyền tải vô tuyến hiện diện dầy đặc, [phân tích](https://vi.wikipedia.org/wiki/Ph%C3%A2n_t%C3%ADch_d%E1%BB%AF_li%E1%BB%87u) [dữ liệu](https://vi.wikipedia.org/wiki/Ph%C3%A2n_t%C3%ADch_d%E1%BB%AF_li%E1%BB%87u) thời gian thực, [máy](https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_m%C3%A1y) học, cảm biến hàng hóa, và [hệ thống nhúng](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_nh%C3%BAng). Điều này có nghĩa là tất cả các dạng thức của hệ thống nhúng cổ điển, như [mạng cảm biến không](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%A1ng_c%E1%BA%A3m_bi%E1%BA%BFn_kh%C3%B4ng_d%C3%A2y&action=edit&redlink=1) [dây](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%A1ng_c%E1%BA%A3m_bi%E1%BA%BFn_kh%C3%B4ng_d%C3%A2y&action=edit&redlink=1), [hệ thống điều khiển](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_%C4%91i%E1%BB%81u_khi%E1%BB%83n), [tự động hóa](https://vi.wikipedia.org/wiki/T%E1%BB%B1_%C4%91%E1%BB%99ng_h%C3%B3a) (bao gồm [nhà thông minh](https://vi.wikipedia.org/wiki/Nh%C3%A0_th%C3%B4ng_minh) và [tự động hóa công](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%B1_%C4%91%E1%BB%99ng_h%C3%B3a_c%C3%B4ng_tr%C3%ACnh&action=edit&redlink=1) [trình](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%B1_%C4%91%E1%BB%99ng_h%C3%B3a_c%C3%B4ng_tr%C3%ACnh&action=edit&redlink=1)), vân vân đều đóng góp vào việc vận hành Internet Vạn Vật (IoT).

### Lợi ích

IoT được coi là chìa khóa thành công của con người trong tương lai gần, nó tác động tích cực đến đời sống, công việc thông qua nhiều ứng dụng:

* Tự động hóa hệ thống nhà thông minh
* Quản lý các thiết bị cá nhân bằng kết nối mạng
* Mua sắm thông minh qua các phần mềm máy tính, điện thoại
* Quản lý môi trường, chất thải trong các nhà máy, xí nghiệp
* Quản lý, lập kết hoạch công việc cho các doanh nghiệp, công ty
* Theo dõi sức khỏe từ xa
* …

Hầu hết các ngành nghề hiện nay đều phát triển hơn dựa trên sự kết nối linh hoạt của mạng lưới IoT. Bao gồm từ giáo dục, nông nghiệp, công nghiệp, y tế,…

Điển hình như các nhà máy sản xuất bắt đầu áp dụng cảm biến cho các thành phần làm ra sản phẩm. Từ đó theo dõi hoạt động của chúng và nâng cao chất lượng. Hay các doanh nghiệp sử dụng công nghệ IoT để quản lý nhân sự, dữ liệu công ty cải thiện hiệu suất làm việc.

1.2 Mục tiêu và phạm vi nghiên cứu của đề tài

* Thiết kế và xây dựng một **vòng tay theo dõi sức khỏe** có khả năng đo **nhịp tim** và **vị trí GPS** của người đeo theo thời gian thực.
* Ứng dụng công nghệ **IoT** để truyền dữ liệu **nhịp tim** và **vị trí địa lý** từ thiết bị đến điện thoại thông minh hoặc máy chủ ThingSpeak thông qua kết nối không dây (WiFi hoặc module SIM).
* Góp phần nâng cao ý thức theo dõi sức khỏe cá nhân và tạo tiền đề cho việc ứng dụng các thiết bị IoT trong lĩnh vực chăm sóc sức khỏe cộng đồng, đặc biệt hỗ trợ **giám sát từ xa** cho người cao tuổi, trẻ nhỏ hoặc bệnh nhân.
* Hệ thống thực hiện các chức năng cơ bản: **Đo nhịp tim**, **Xác định vị trí GPS**, **Hiển thị thông số nhịp tim.**
* Không đi sâu vào phân tích chuyên sâu về y học hoặc xử lý tín hiệu sinh học phức tạp như ECG/PPG nâng cao.
* Thử nghiệm hệ thống trên quy mô nhỏ, chủ yếu trong điều kiện **mô phỏng và môi trường học tập**, đủ để kiểm chứng tính khả thi về mặt công nghệ.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## **2.1 Module NodeMCU ESP8266**



*Hình* *2.1.* *NodeMCU ESP8266*

### Thông số kỹ thuật

|  |  |
| --- | --- |
| Vi điều khiển | CPU RISC 32-bit Tensilica Xtensa LX106 |
| Điện áp hoạt động | 3.0 – 3.6V (thường dùng 3.3V) |
| Tần số hoạt động | 80 MHz (có thể ép xung lên 160 MHz) |
| Dòng tiêu thụ | Khoảng 70 mA khi hoạt động |
| Điện áp vào giới hạn | 5V DC qua cổng USB (trên bo NodeMCU có bộ chuyển xuống 3.3V) |
| Số chân Digital I/O | 17 chân GPIO (tùy vào module, NodeMCU có khoảng 11 dùng được) |
| Số chân Analog | 1 kênh ADC 10-bit (0–1V đầu vào, hoặc đến 3.3V có chia áp) |
| Dòng tối đa trên mỗi chân I/O | Khoảng 12 mA (an toàn), tối đa lên đến 20 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3.3V) | ~600 mA |
| Bộ nhớ flash | 4 MB |
| SRAM | ~50 KB |
| EEPROM | Không có thật, nhưng có thể giả lập qua flash |
| Khối lượng | 7.5 gram |

### Vi điều khiển

Bảng phát triển NodeMCU ESP8266 đi kèm với mô-đun ESP-12E chứa chip ESP8266 có bộ vi xử lý Tensilica Xtensa 32-bit LX106 RISC. Bộ vi xử lý này hỗ trợ RTOS và hoạt động ở tần số xung nhịp có thể điều chỉnh từ 80MHz đến 160 MHz.

NodeMCU có 128 KB RAM và 4MB bộ nhớ Flash để lưu trữ dữ liệu và chương trình. Sức mạnh xử lý cao của nó với Wi-Fi / Bluetooth và các tính năng Điều hành Ngủ sâu tích hợp khiến nó trở nên lý tưởng cho các dự án IoT.

NodeMCU có thể được cấp nguồn bằng giắc cắm Micro USB và chân VIN (Chân nguồn cung cấp bên ngoài). Nó hỗ trợ giao diện UART, SPI và I2C.

### Các chân của ESP8266

Các chân năng lượng: GND (Ground), 3.3V, Vin (Voltage Input), EN (Enable), RST (Reset).

ESP8266 sử dụng điện áp hoạt động 3.3V, không hỗ trợ 5V trực tiếp vào các chân GPIO. Vin có thể nhận điện áp từ 4.5V đến 10V (qua cổng USB hoặc nguồn ngoài).

Các cổng ra/vào: NodeMCU có tổng cộng 11 chân digital (D0 – D10) có thể đọc hoặc xuất tín hiệu, trong đó 8 chân hỗ trợ PWM.

### Giao thức UART

UART (Universal Asynchronous Receiver-Transmitter – Bộ truyền nhận dữ liệu không đồng bộ) là một giao thức truyền thông phần cứng dùng giao tiếp nối tiếp không đồng bộ và có thể cấu hình được tốc độ.

Giao thức UART là một giao thức đơn giản và phổ biến, bao gồm hai đường truyền dữ liệu độc lập là TX (truyền) và RX (nhận). Dữ liệu được truyền và nhận qua các đường truyền này dưới dạng các khung dữ liệu (data frame) có cấu trúc chuẩn, với một bit bắt đầu (start bit), một số bit dữ liệu (data bits), một bit kiểm tra chẵn lẻ (parity bit) và một hoặc nhiều bit dừng (stop bit).

Thông thường, tốc độ truyền của UART được đặt ở một số chuẩn, chẳng hạn như 9600, 19200, 38400, 57600, 115200 baud và các tốc độ khác. Tốc độ truyền này định nghĩa số lượng bit được truyền qua mỗi giây. Các tốc độ truyền khác nhau thường được sử dụng tùy thuộc vào ứng dụng và hệ thống sử dụng.

## **2.2 Tìm hiểu về Thingspeak**

### **2.2.1. Giới thiệu phần mềm Thingspeak**

ThingSpeak là một nền tảng IoT miễn phí được phát triển bởi MathWorks. Nó cho phép người dùng thu thập, lưu trữ, phân tích và hiển thị dữ liệu từ xa qua Internet.

ThingSpeak được sử dụng rộng rãi trong các dự án IoT, đặc biệt với Arduino, ESP8266, ESP32 và các thiết bị nhúng khác. Ngoài việc hiển thị dữ liệu theo thời gian thực qua biểu đồ, ThingSpeak còn hỗ trợ xử lý dữ liệu bằng MATLAB, tự động gửi cảnh báo hoặc điều khiển thiết bị từ xa.

### **2.2.2 Các trường dữ liệu gửi lên Thingspeak**

Thực hiện gửi các thông tin thu được từ các cảm biến đến Thingspeak để biểu diễn, phân tích, đánh giá. Các thông tin được gửi gồm: Kinh độ, Vĩ độ được thu thập qua cảm biến GPS, Nhịp tim, Nồng độ O2 được thu thập từ cảm biến Nhịp Tim, O2.A screenshot of a computer

AI-generated content may be incorrect.

*Các trường được gửi lên Thingspeak: Kinh độ, Vĩ độ, Nhịp tim, Nồng độ O2*

2.2.3 Hiển thị GPS bằng MATLAB Visualizations

Thực hiện hiển thị tọa độ GPS trên Map thông qua MATLAB Visualization

**A screenshot of a map

AI-generated content may be incorrect.**

Ảnh hiển thị GPS

## **Cảm biến và các module chức năng**

### 2.3.1. Mạch định vị GPS ATGM336H



*Hình 2.3.1. GPS ATGM336H*

###### **Mô Tả Sản Phẩm**

Mạch định vị GPS BDS ATGM336H có thiết kế nhỏ gọn sử dụng IC chính SoC GNSS AT6558 thế hệ thứ 4 với khả năng tiết kiệm năng lượng vượt trội, mạch bắt tín hiệu định vị và thời gian nên các hệ thống GPS/US, Beidou/CN, GLONASS/RU, Galileo/EU, QZSS/JP, SBAS/enhanced system qua 32 kênh tracking chanel.

Mạch định vị GPS BDS ATGM336H có chuẩn đầu ra tín hiệu của mạch tương thích với các module của Ublox (NEO-6M / NEO-7/ NEO-M8N) nên có thể thay thế dễ dàng, sử dụng chung code mẫu Arduino và phần mềm U-Center trên máy tính, phù hợp với các ứng dụng định vị vị trí và lấy thời gian qua GPS.

###### **Thông số kỹ thuật**

* Điện áp cấp: 3.3~ 5VDC
* IC chính: SoC GNSS AT6558
* Giao tiếp UART/TTL.
* Baudrate: 9600 (Default), 1200, 2400, 4800, 19200, 38400, 57600, 115200, 230400, 460800, 921600.
* Tracking channels: 32
* GNSS engine for GPS/US, Beidou/CN, GLONASS/RU, Galileo/EU, QZSS/JP, SBAS/enhanced system.
* Support A-GNSS
* Cold start capture sensitivity: -148dBm
* Tracking sensitivity: -162dBm
* Positioning accuracy: 2.5 meters (CEP50, open area)
* First positioning time: 32 seconds
* Low power consumption: continuous operation <25mA (@ 3.3V)
* Kích thước: 15.7 x 13.1 mm

### 2.3.2 Module MAX30102



Hình 2.3.2.1 Max30102

***Mô Tả Sản Phẩm***

Cảm biến nhịp tim và oxy trong máu Max30102 là một mô-đun đo nhịp tim và oxy trong máy tích hợp. Nó bao gồm đèn LED bên trong, bộ tách sóng quang, các bộ phận quang học và các thiết bị điện tử có tiếng ồn thấp với khả năng loại bỏ ánh sáng xung quanh.

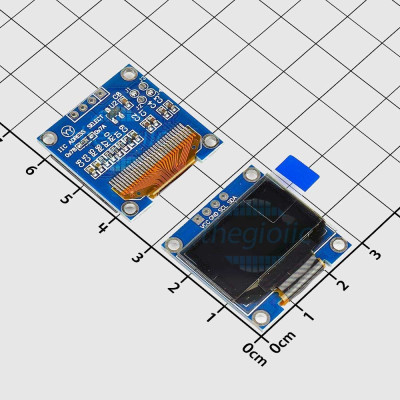
Cảm biến nhịp tim và oxy trong máu Max30102 hoạt động trên một nguồn cung cấp điện 1.8V và một nguồn cấp điện 5.0V riêng biệt cho các đèn LED bên trong.

Giao tiếp thông qua giao diện tương thích I2C tiêu chuẩn và có thể được tắt thông qua phần mềm với chế độ chờ bằng không. Nó cho phép các thanh ray nguồn vẫn được cấp nguồn ở mọi thời điểm.

###### **Thông số kỹ thuật**

* IC chính: MAX30102.
* Đo được nhịp tim và nồng độ Oxy trong máu.
* Điện áp sử dụng: 2.5~5.5VDC.
* 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

### 2.3.2 SSD1306



Hình 2.3.2.1 LCD OLED SSD1306

***Mô Tả Sản Phẩm***

Màn hình OLED SSD1306 với kích thước 0.96 inch, cho khả năng hiển thị hình ảnh tốt với khung hình 128×64 pixel. Ngoài ra, màn hình còn tương thích với hầu hết các vi điều khiển hiện nay thông qua giao tiếp I2C. Màn hình sử dụng driver SSD1306 cùng thiết kế nhỏ gọn sẽ giúp bạn phát triển các sản phẩm DIY hoặc các ứng dụng khác một cách nhanh chóng..

***Thông số kỹ thuật***

Driver: SSD1306

Tương thích với Arduino, 51 Series, MSP430 Series, STM32/2, CSRIC, …

Tiêu thụ điện năng thấp: 0.08W (fullscreen)

Có thể điều chỉnh độ sáng và độ tương phản

Chuẩn giao tiếp: I2C (thông qua 2 chân SCL, SDA)

Điện áp hoạt động: 3V-5V DC

Nhiệt đô hoạt động: -30℃-70℃

Kích thước màn hình: 0.96 inch (128×64 pixel)

Kích thước module: 26.70\* 19.26\* 1.85mm (1.030.760.07 inch).

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## **Nguyên lý hoạt động**

Thiết bị cung cấp các chức năng: Định vị theo thời gian thực qua module GPS ATGM336H; Đo nhịp tim, nồng độ CO2 qua module MAX30102; Hiển thị trên module SSD1306 sau đó các thông tin được tải lên Thingspeak.

## **3.2 Sơ đồ đấu nối**

| **Module** | **ESP8266 Pin** | **Module Pin** | **Chức Năng** |
| --- | --- | --- | --- |
| **GPS ATGM336H** | D6 (RX) | TX | Giao tiếp UART (TX của GPS nối với RX của ESP8266) |
| **GPS ATGM336H** | D7 (TX) | RX | Giao tiếp UART (RX của GPS nối với TX của ESP8266) |
| **GPS ATGM336H** | 3.3V | VCC | Cung cấp nguồn 3.3V cho GPS |
| **GPS ATGM336H** | GND | GND | Nối đất giữa ESP8266 và GPS |
| **MAX30102** | D1 (SCL) | SCL | Giao tiếp I2C (SCL của MAX30102 nối với D1 của ESP8266) |
| **MAX30102** | D2 (SDA) | SDA | Giao tiếp I2C (SDA của MAX30102 nối với D2 của ESP8266) |
| **MAX30102** | 3.3V | VCC | Cung cấp nguồn 3.3V cho MAX30102 |
| **MAX30102** | GND | GND | Nối đất giữa ESP8266 và MAX30102 |
| **SSD1306** | Kết nối như Max30102 | Kết nối như Max30102 | Hiển thị Bpm và spO2 để người đeo có thể theo dõi trực tiếp |

## **Code sản phẩm**

//Danh sách thư viện

#include <Wire.h>

#undef I2C\_BUFFER\_LENGTH // tránh bị redefine

#include <MAX30105.h>

#include "heartRate.h"

#include <ESP8266WiFi.h>

#include "ThingSpeak.h"

#include <TinyGPSPlus.h>

#include <SoftwareSerial.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

// ===== SSD1306 OLED =====

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define OLED\_RESET -1

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

// ===== WiFi & ThingSpeak =====

const char\* ssid = "Anh An 0103";

const char\* password = "0913524247";

const unsigned long channelID = 2840798;

const char\* writeAPIKey = "WATQA4GV3ZR4KZIU";

const uint32\_t UPLOAD\_INTERVAL = 15000;

// ===== GPS =====

static const uint32\_t GPSBaud = 9600;

SoftwareSerial gpsSerial(D6 /\*RX\*/, D7 /\*TX\*/);

TinyGPSPlus gps;

// ===== Biến toàn cục =====

WiFiClient client;

MAX30105 particleSensor;

const uint8\_t IBI\_SAMPLES = 8;

uint32\_t ibiTimes[IBI\_SAMPLES];

uint8\_t ibiIndex = 0;

bool ibiFilled = false;

uint32\_t lastBeatTime = 0;

float beatsPerMinute = 0.0;

float spO2Value = 0.0;

float latitude = NAN;

float longitude = NAN;

uint32\_t lastGpsMsg = 0;

const uint32\_t GPS\_MSG\_INTERVAL = 5000;

void setup() {

Serial.begin(115200);

delay(10);

Serial.println(F("ESP8266 + MAX30102 + GPS + OLED + ThingSpeak"));

// 1) Kết nối WiFi

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

Serial.print(F("Connecting to WiFi"));

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(F("."));

}

Serial.println(F("\nWiFi connected"));

// 2) ThingSpeak

ThingSpeak.begin(client);

// 3) MAX30102

if (!particleSensor.begin(Wire, I2C\_SPEED\_FAST)) {

Serial.println(F("ERROR: MAX30102 not found!"));

while (1) delay(10);

}

particleSensor.setup();

Serial.println(F("MAX30102 initialized"));

// 4) GPS

gpsSerial.begin(GPSBaud);

Serial.println(F("GPS initialized"));

// 5) SSD1306 OLED

if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

Serial.println(F("ERROR: SSD1306 not found!"));

while (1);

}

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(0, 0);

display.println(F("OLED initialized"));

display.display();

delay(1000);

}

void loop() {

// —— GPS ——

while (gpsSerial.available()) gps.encode(gpsSerial.read());

if (gps.location.isValid()) {

latitude = gps.location.lat();

longitude = gps.location.lng();

} else if (millis() - lastGpsMsg > GPS\_MSG\_INTERVAL) {

lastGpsMsg = millis();

Serial.println(F("Waiting for GPS lock..."));

}

// —— MAX30102: BPM & SpO2 ——

long irValue = particleSensor.getIR();

if (checkForBeat(irValue)) {

uint32\_t now = millis();

uint32\_t delta = now - lastBeatTime;

lastBeatTime = now;

if (delta > 250 && delta < 2000) {

ibiTimes[ibiIndex] = delta;

ibiIndex = (ibiIndex + 1) % IBI\_SAMPLES;

if (ibiIndex == 0) ibiFilled = true;

uint32\_t sum = 0;

uint8\_t count = ibiFilled ? IBI\_SAMPLES : ibiIndex;

for (uint8\_t i = 0; i < count; i++) sum += ibiTimes[i];

float avgIBI = float(sum) / float(count);

beatsPerMinute = 60000.0 / avgIBI;

int16\_t redLED = particleSensor.getRed();

spO2Value = 110.0 - 25.0 \* (float(redLED) / float(irValue));

spO2Value = constrain(spO2Value, 70.0, 100.0);

// —— Serial Debug ——

Serial.print(F("BPM=")); Serial.print(beatsPerMinute, 1);

Serial.print(F(" SpO2=")); Serial.print(spO2Value, 1);

if (!isnan(latitude) && !isnan(longitude)) {

Serial.print(F(" Lat=")); Serial.print(latitude, 6);

Serial.print(F(" Lon=")); Serial.println(longitude, 6);

} else {

Serial.println();

}

// —— OLED Display ——

display.clearDisplay(); display.setTextSize(2);

display.setCursor(0, 0); display.print("BPM: "); display.println(beatsPerMinute, 1); display.setTextSize(1);

display.print("SpO2: "); display.print(spO2Value, 1); display.println("%");

if (!isnan(latitude) && !isnan(longitude)) {

display.setTextSize(1); display.print("Lat: ");

display.println(latitude, 4); display.print("Lon: "); display.println(longitude, 4);

} else {

display.println("Waiting GPS..."); }

display.display(); } }

// —— ThingSpeak ——

static uint32\_t lastUpload = 0;

if (millis() - lastUpload >= UPLOAD\_INTERVAL) {

lastUpload = millis();

ThingSpeak.setField(1, beatsPerMinute);

ThingSpeak.setField(2, spO2Value);

if (!isnan(latitude)) ThingSpeak.setField(3, latitude);

if (!isnan(longitude)) ThingSpeak.setField(4, longitude);

int httpCode = ThingSpeak.writeFields(channelID, writeAPIKey);

if (httpCode == 200) {

Serial.println(F("✔ ThingSpeak upload successful")); } else {

Serial.print(F("✖ Upload failed, HTTP code="));

Serial.println(httpCode);

}

}

delay(10);

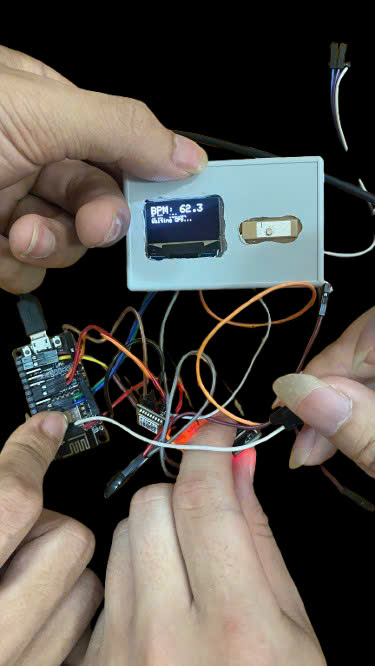
}

## **Thông tin hình ảnh sản phẩm**

A close-up of a switch with wires

AI-generated content may be incorrect.

Ảnh Sơ đồ ghép nối



Thông tin hiển thị trên màn Led

A hand holding a device

AI-generated content may be incorrect.A device on a wrist

AI-generated content may be incorrect.

Ảnh sản phẩm hoàn thiện

# LỜI KẾT

Việc theo dõi vị trí và sức khỏe người dùng, đặc biệt với các hoạt động ngoài trời hoặc người lớn tuổi là việc rất cấp thiết hiện nay. Khi xảy ra vấn đề về sức khỏe, sự cứu chữa kịp thời đóng vai trò quan trọng trong việc cứu sống bệnh nhân. Thiết bị của chúng em sẽ giúp giải quyết vấn đề bằng việc lưu lại dữ liệu về nhịp tim, nồng độ O2 để phân tích, dự đoán về bệnh cho người dùng, thông báo đến bác sĩ vị trí của bệnh nhân khi phát hiện bất thường.

Bọn em đã sử dụng Bot Telegram để thông báo về vị trí nhưng do sự việc Khong mong muốn nên đã không thể sử dụng được. Ngoài ra, khi phát triển về module SIM, do lượng tiêu thị điện của module SIM khiến việc tìm giải pháp cấp nguồn di động rất khó khăn. Việc giới hạn về sản phẩm là vòng đeo tay nên việc hoàn thiện còn chưa đạt đến kết quả mà bọn em mong muốn.

Từ đó, thiết bị còn nhiều hạn chế như không tự cung cấp được Internet, thiếu giải pháp phân tích dữ liệu …. Những hạn chế đó sẽ được bọn em khắc phục trong tương lai để sản phẩm ngày một hoàn thiện hơn.

|  |  |  |
| --- | --- | --- |
| Công việc | Mô tả | Người làm |
| Node MCU phần Máu và GPS : Code | Kết nối Node với Module Máu và GPS, cập nhập hiển thị trên Thingspeak | Phát, Tùng |
| Hoàn thiện sản phẩm | Kết nối các module và hoàn thiện sản phẩm: Vòng đeo tay, dây đeo Yêu cầu nhỏ gọn nhất có thể | An, Giang |
| Báo cáo | Báo cáo theo mẫu | Hoàng |
| Trưởng nhóm |  | Giang |