

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN VIỄN THÔNG**

-----o0o-----



**ĐỒ ÁN MÔN HỌC 2**

**THIẾT KẾ HỆ THỐNG ĐO NHỊP TIM**

**GVHD: PGS. TS. Hà Hoàng Kha**

**SVTH: Lê Tuấn Thành**

**MSSV: 2012046**

**TP. HỒ CHÍ MINH, THÁNG 9, NĂM 2024**

## LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Bách Khoa Thành phố Hồ Chí Minh đã đưa môn học Đồ án môn học 2 vào chương trình giảng dạy. Chân thành cảm ơn các thầy cô Khoa Điện – Điện tử đã hỗ trợ thiết bị thí nghiệm để em có thể thực hiện đồ án này. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn – PGS. TS. Hà Hoàng Kha đã tận tình hướng dẫn, giải đáp thắc mắc cũng như truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua.

Với sự tìm hiểu, sáng tạo cũng như sự giúp đỡ của thầy em đã hoàn thành được các mục tiêu đề ra của đề tài. Tuy nhiên trong quá trình nghiên cứu, do kiến thức còn hạn chế nên vẫn còn nhiều thiếu sót trong việc tìm hiểu, trình bày và đánh giá kết quả của đề tài. Rất mong nhận được sự chú ý, đóng góp của thầy/ cô giảng viên bộ môn để em có thể rút kinh nghiệm và hoàn thiện hơn.

*Tp. Hồ Chí Minh, ngày 19 tháng 09 năm 2024.*

**Giảng viên hướng dẫn**

**Sinh viên**

## TÓM TẮT ĐỒ ÁN

Đồ án này tập trung nghiên cứu và phát triển hệ thống để theo dõi nhịp tim và độ bão hoà oxi trong máu. Mục tiêu chính là cung cấp một thiết bị hiệu quả giúp hỗ trợ người dùng có thể dễ dàng đo đạc các thông số sức khoẻ và từ đó có thể chuẩn đoán được một số vấn đề có thể gặp phải. Trong quá trình này sẽ trình bày về quá trình tìm hiểu, nghiên cứu cách thức hoạt động của một hệ thống đo nhịp tim ở người. Tiếp đó tập trung tìm hiểu về nguyên lý của IC MAX30102, tính toán các thông số và ứng dụng của IC vào trong thiết kế. Các bước thiết kế mạch nguyên lý, mô phỏng, layout PCB cũng được trình bày chi tiết trong Đồ án này. Cuối cùng, là bước đo đạc, đánh giá các thông số đặc trưng của hệ thống và các hướng phát triển của hệ thống này. Kết quả nghiên cứu của đề tài giúp ứng dụng vào lĩnh vực y tế nhằm đo đạc nhịp tim và độ bão hoà oxi trong máu từ đó có thể dễ dàng chuẩn đoán các vấn đề liên quan về sức khoẻ.

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>i</b>
<b>TÓM TẮT ĐỒ ÁN.....</b>	<b>ii</b>
<b>DANH SÁCH HÌNH ẢNH.....</b>	<b>v</b>
<b>DANH SÁCH BẢNG.....</b>	<b>vii</b>
<b>DANH SÁCH CHỮ VIẾT TẮT .....</b>	<b>viii</b>
<b>CHƯƠNG 1: PHẦN MỞ ĐẦU.....</b>	<b>1</b>
<b>1. Tổng quan đề tài .....</b>	<b>1</b>
<b>2. Nhiệm vụ đề tài .....</b>	<b>1</b>
<b>CHƯƠNG 2: PHẦN NỘI DUNG.....</b>	<b>2</b>
<b>1. LÝ THUYẾT .....</b>	<b>2</b>
1.1. Nguyên lý hoạt động của các máy đo nhịp tim và độ bão hoà oxi trong máu .....	2
1.2. Hệ thống đo nhịp tim .....	4
<b>2. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG.....</b>	<b>5</b>
2.1. Yêu cầu thiết kế.....	5
2.2. Phân tích thiết kế.....	5
2.3. Sơ đồ khối tổng quát .....	6
2.4. Sơ đồ khối chi tiết .....	7
2.5. Lựa chọn các linh kiện và thiết kế mạch nguyên lý.....	8
2.6. Thiết kế và bố trí linh kiện – Layout PCB .....	19
<b>3. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM.....</b>	<b>21</b>
3.1. Yêu cầu cho hệ thống.....	21
3.2. Giới thiệu các phần mềm và nền tảng .....	22
3.3. Cấu hình trên STM32CubeIDE cho STM32F103C8T6 .....	26
3.4. MAX30102 .....	29

<b>4. Kết quả thực hiện.....</b>	<b>34</b>
4.1. Kết quả phần cứng .....	34
4.2. Kết quả phần mềm .....	35
<b>CHƯƠNG 3: PHẦN KẾT LUẬN .....</b>	<b>38</b>
<b>1. Kết luận.....</b>	<b>38</b>
<b>2. Hướng phát triển .....</b>	<b>38</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>39</b>
<b>PHỤ LỤC .....</b>	<b>40</b>

## DANH SÁCH HÌNH ẢNH

Hình 1-1: Nguyên lý đo nhịp tim bằng quang học.....	2
Hình 1-2: Độ hấp thụ ánh sáng của phân tử $HbO_2$ và $Hb$ .....	3
Hình 1-3: Mối quan hệ giữa tỉ số $R/IR$ và độ bão hoà của oxy trong máu .....	3
Hình 1-4: Sơ đồ khối hệ thống đo nhịp tim .....	4
Hình 2-1: Sơ đồ khối tổng quát của hệ thống .....	6
Hình 2-2: Sơ đồ khối chi tiết của hệ thống .....	7
Hình 2-3: Sơ đồ nguyên lý mạch chuyển đổi nguồn cấp cho hệ thống.....	9
Hình 2-4: IC Ổn Áp RT9193 – 33GB 3.3V .....	9
Hình 2-5: Thiết kế đề xuất của IC ổn áp RT9193 – 33GB 3.3V.....	10
Hình 2-6: Sơ đồ nguyên lý mạch nguồn 3.3V .....	10
Hình 2-7: IC ổn áp AMS1117 – 1.8V .....	11
Hình 2-8: Thiết kế đề xuất của IC ổn áp AMS1117 – 1.8V.....	11
Hình 2-9: Sơ đồ nguyên lý mạch nguồn 1.8V .....	12
Hình 2-10: Sơ đồ chân của STM32F103C8T6 .....	13
Hình 2-11: Đề xuất thiết kế cấp nguồn cho vi điều khiển STM32F103C8T6 .....	13
Hình 2-12: Mạch nguyên lý của STM32F103C8T6 .....	14
Hình 2-13: Thiết kế header để cấp nguồn và đường dây tín hiệu ra ESP8266 .....	14
Hình 2-14: Module Node MCU – ESP8266 .....	15
Hình 2-15: Các khối chức năng của IC MAX30102.....	16
Hình 2-16: Cấu hình chân IC MAX30102.....	16
Hình 2-17: Thiết kế đề xuất cho IC MAX30102 .....	18
Hình 2-18: Mạch nguyên lý cho khối cảm biến IC MAX30102.....	18
Hình 2-19: Thiết kế nguyên lý cho OLED SSD1306 .....	19
Hình 2-20: Thiết kế Layout (lớp trên).....	19

Hình 2-21: Thiết kế Layout (lớp dưới) .....	20
Hình 2-22: Khối MCU, các tụ nguồn và thạch anh (mặt trên).....	20
Hình 2-23: Đường dây vi sai (differential pair) cho hai đường dây tín hiệu .....	21
Hình 3-1: Giao diện STM32CubeIDE .....	22
Hình 3-2: Giao diện ArduinoIDE.....	23
Hình 3-3: Giao diện Thingsboard .....	24
Hình 3-4: Cấu hình chân hệ thống trên STM32CubeIDE (Pinout view).....	26
Hình 3-5: Cấu hình các chức năng sử dụng (system view).....	26
Hình 3-6: Thông tin thanh ghi trạng thái .....	29
Hình 3-7: Thông tin thanh ghi ngắt.....	29
Hình 3-8: Thông tin thanh ghi cho phép ngắt .....	30
Hình 3-9: Thông tin thanh ghi FIFO .....	31
Hình 3-10: Sơ đồ giải thuật tổng quát.....	32
Hình 3-11: Lưu đồ giải thuật STM32F103C8T6 .....	32
Hình 3-12: Lưu đồ giải thuật ESP8266.....	33
Hình 4-1: Mạch PCB trước khi hàn linh kiện .....	34
Hình 4-2: Mạch sau khi hoàn thiện .....	35
Hình 4-3: Giá trị tín hiệu số thu được của R và IR khi chưa đặt tay lên cảm biến .....	35
Hình 4-4: Giá trị tín hiệu số thu được khi đặt tay lên cảm biến.....	36
Hình 4-5: Giá trị nồng độ SpO <sub>2</sub> thu được .....	36
Hình 4-6: Mạch hiển thị giá trị SpO <sub>2</sub> .....	37
Hình 4-7: Giá trị SpO <sub>2</sub> hiển thị trên Thingsboard .....	37

## DANH SÁCH BẢNG

Bảng 2-1: Thông số IC ổn áp RT9193 – 33GB .....	10
Bảng 2-2: Thông số IC ổn áp AMS1117 – 1.8V.....	11
Bảng 2-3: Thông số STM32F103C8T6 .....	12
Bảng 2-4: Thông số kỹ thuật ESP8266.....	15
Bảng 2-5: Bảng mô tả chức năng chân IC MAX30102 .....	17
Bảng 2-6: Thông số OLED SSD1306.....	19
Bảng 3-1: Bảng danh sách các chân cấu hình GPIO và cấu hình chế độ.....	27
Bảng 3-2: Bảng danh sách cấu hình Timer 2 tạo PWM.....	27
Bảng 3-3: Cấu hình thông số cho Timer 2 tạo xung PWM.....	27
Bảng 3-4: Bảng danh sách cấu hình giao thức I2C .....	27
Bảng 3-5: Cấu hình thông số cho giao thức I2C .....	28
Bảng 3-6: Bảng danh sách cấu hình giao thức UART .....	28
Bảng 3-7: Cấu hình thông số cho giao thức I2C .....	28



**DANH SÁCH CHỮ VIẾT TẮT**

<b>Tên viết tắt và kí hiệu</b>	<b>Giải thích</b>
Hb	The deoxygenated hemoglobin
HbO <sub>2</sub>	The oxygenated hemoglobin
HR	Thông số xung nhịp tim
I2C	Inter-Integrated Circuit
IC	Integrated Circuit – Mạch tích hợp
IR	Tia hồng ngoại
R	Ánh sáng đỏ
PWM	Pulse Width Modulation – Điều chế độ rộng xung
SpO <sub>2</sub>	Oxygen saturation – Độ bão hoà Oxy tại ngón tay
UART	Universal Asynchronous Receiver-Transmitter.

## CHƯƠNG 1: PHẦN MỞ ĐẦU

### 1. Tổng quan đề tài

Sức khoẻ là một yếu tố đóng vai trò quan trọng. Nó là cơ sở để con người có thể làm mọi việc hằng ngày trong cuộc sống như học tập, đi làm, vui chơi, ... Ngày nay, khi xã hội ngày càng phát triển con người lại càng chú trọng hơn vào sức khoẻ cá nhân. Chính vì vậy việc thiết kế và ứng dụng các thiết bị có khả năng hỗ trợ đo đạc, tính toán các thông số sức khoẻ giúp ngành y tế nhanh có thể chuẩn đoán chính xác và nhanh chóng là rất cần thiết.

Đề tài này tập trung vào việc thiết kế hệ thống đo nhịp tim. Thiết bị dựa trên nguyên lý sự thay đổi mật độ ánh sáng đi qua trên bề mặt da để tính toán nhịp tim bằng photodetector trên IC MAX30102. Từ kết quả trả về, người dùng có thể ứng dụng để phân tích và chuẩn đoán một số vấn đề tim mạch liên quan như nhịp tim không đều, suy tim, nhồi máu cơ tim hoặc một số vấn đề liên quan đến độ bão hoà của oxi trong máu.

### 2. Nhiệm vụ đề tài

Nội dung 1: Tìm hiểu nguyên lý, ứng dụng của hệ thống

Nội dung 2: Tìm hiểu về dòng vi điều khiển STM32F103C8T6 và các thiết bị ngoại vi cần thiết

Nội dung 3: Thiết kế mạch nguyên lý, mạch testboard, kiểm tra mạch

Nội dung 4: Thực hiện thiết kế mạch in PCB trên phần mềm Kicad

Nội dung 5: Thiết kế lưu đồ giải thuật và lập trình trên STM32CubeIDE

Nội dung 6: Đo đạc, so sánh với các thiết bị khác, đánh giá sản phẩm và kết luận

## CHƯƠNG 2: PHẦN NỘI DUNG

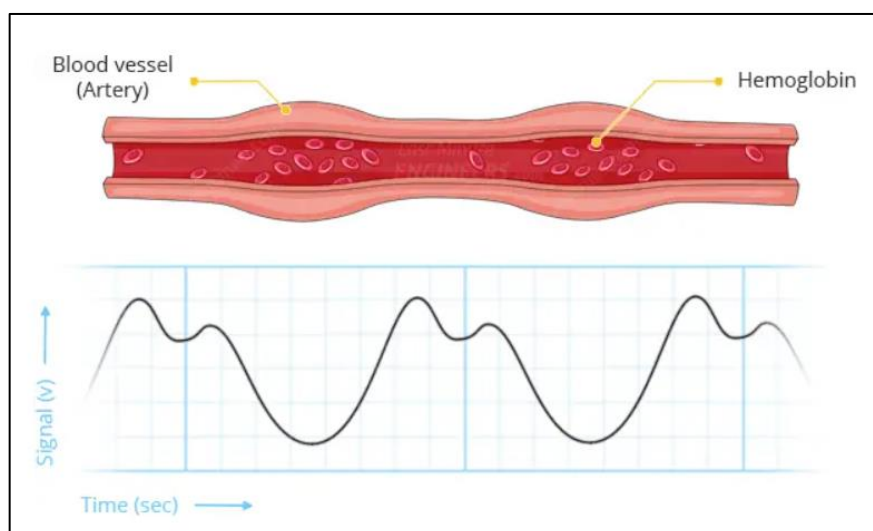
### 1. LÝ THUYẾT

#### 1.1. Nguyên lý hoạt động của các máy đo nhịp tim và độ bão hoà oxy trong máu

Hiện nay có hai loại đo nhịp tim là loại đeo ngay ngực và theo dõi nhịp tim quang học

Thứ nhất, sử dụng điện tâm đồ để ghi lại hoạt động điện tim của người dùng. “Phần điện cực trong thiết bị cần có hơi nước hoặc mồ hôi để bắt các tín hiệu điện tim. Do đó, khi bạn đang vận động và đổ mồ hôi, các điện cực sẽ bắt những tín hiệu điện tim trong cơ thể, sau đó gửi dữ liệu vào bộ phận phát.”<sup>1</sup>. Các thiết bị này thường được thiết kế để đeo trên ngực.

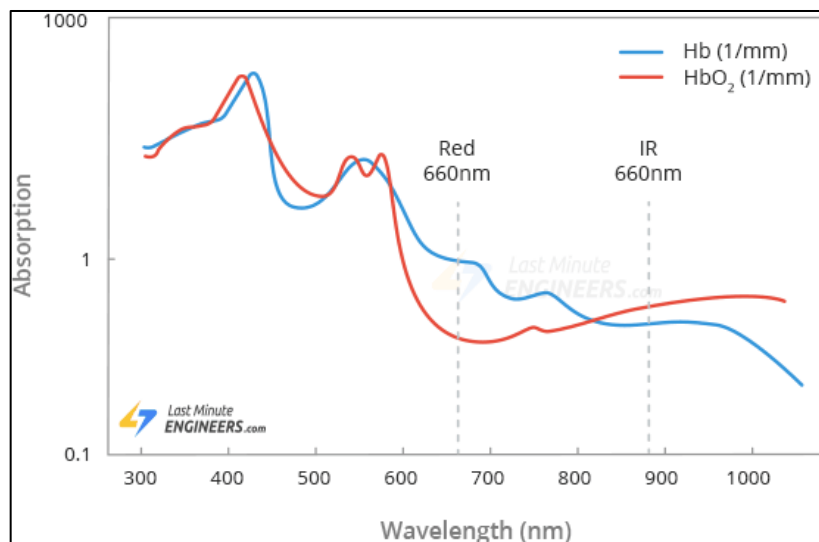
Thứ hai, sử dụng các thiết bị đo nhịp tim quang học. Nó dựa trên cơ chế sự thay đổi mật độ ánh sáng đi qua các vùng da như da tai, ngón tay (nơi mà da mỏng, không quá dày). Các thiết bị thu thập dữ liệu bằng cách đo mật độ ánh sáng thông qua photo – detector. The oxygenated hemoglobin (HbO<sub>2</sub>) trong động mạch có đặc tính hấp thụ ánh sáng IR (tia hồng ngoại). Khi màu càng đỏ đồng nghĩa với việc nồng độ hemoglobin càng cao, lúc này ánh sáng IR càng được hấp thụ nhiều. Khi máu được bơm qua ngón tay sau mỗi chu kỳ nhịp tim, lượng ánh sáng IR phản xạ lại sẽ thay đổi. Chính vì vậy nó sẽ tạo ra dạng sóng thay đổi tới máy dò quang. Khi liên tục chiếu sáng và đo các chỉ số của máy dò quang sẽ đo được thông số xung nhịp tim (HR) .



**Hình 1-1: Nguyên lý đo nhịp tim bằng quang học**

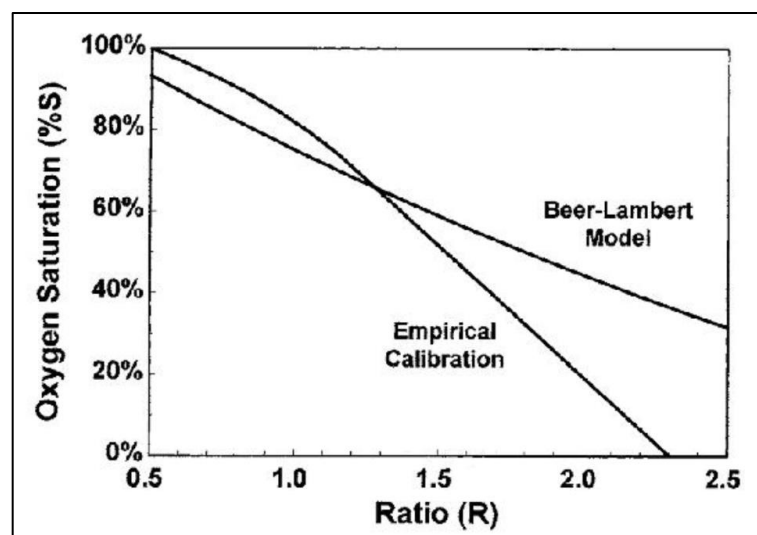
<sup>1</sup> D. Kim Thoa, (06/ 04/ 2017), “Máy đo nhịp tim hoạt động ra sao?”, Truy cập từ: <https://tuoitre.vn/tu-van-tieu-dung/may-do-nhip-tim-hoat-dong-ra-sao-1292606.htm>.

Trong khi đó để đo độ bão hoà oxy trong máu dựa vào sự hấp thụ các bước sóng khác nhau của các phân tử  $\text{HbO}_2$  và Hb (The deoxygenated hemoglobin). Mối quan hệ của việc hấp thụ của từng loại phân tử theo mức quang phổ được thể hiện qua **hình 1 – 2**. Có thể thấy rằng với Hb hấp thụ ánh sáng đỏ nhiều hơn (với bước sóng 660nm). Trong khi đó, đối với  $\text{HbO}_2$  lại hấp thụ nhiều ánh sáng IR hơn (với bước sóng 880nm).



**Hình 1-2: Độ hấp thụ ánh sáng của phân tử  $\text{HbO}_2$  và Hb**

Bằng cách đo tỉ lệ ánh sáng IR và R từ photo – detector có thể tính toán và đo được thông số  $\text{SpO}_2$  trong máu. Mối liên hệ giữa tỉ lệ ánh sáng R/ IR với  $\text{SpO}_2$  được biểu diễn thông qua đồ thị ở **hình 1 – 3**. Nó được xây dựng dựa trên định luật Beer – Lambert<sup>2</sup> và thực nghiệm.



**Hình 1-3: Mối quan hệ giữa tỉ số R/ IR và độ bão hoà của oxy trong máu**

<sup>2</sup> Định luật Beer – Lambert: Được phát triển dựa trên hiện tượng hấp thụ bức xạ điện từ của một dung dịch

Bên cạnh đó, giá trị SpO<sub>2</sub> còn có thể tính bằng công thức như sau:

$$\text{Ratio} = \frac{\frac{\text{Red}_{AC}}{\text{Red}_{DC}}}{\frac{\text{IR}_{AC}}{\text{IR}_{DC}}}$$

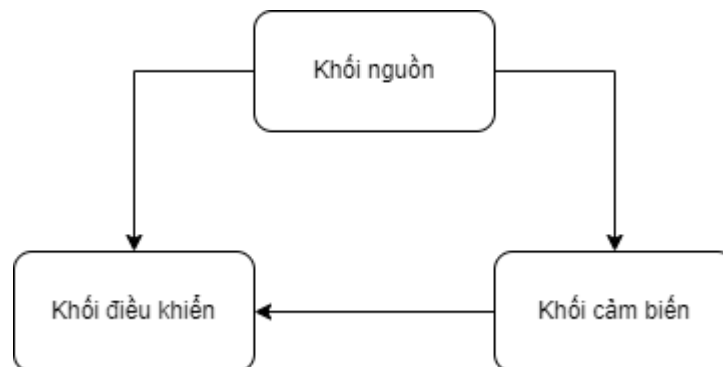
$$\text{SpO}_2 = A - B \cdot \text{Ratio}$$

**Trong đó:** A, B là các hằng số thực nghiệm thường có giá trị A = 110, B = 25

## 1.2. Hệ thống đo nhịp tim

Trong đề tài này, nghiên cứu chủ yếu về cách thức đo nhịp tim bằng quang học. Hệ thống thiết kế được chia làm các khối chính bao gồm:

- Khối điều khiển
- Khối nguồn DC
- Khối cảm biến



**Hình 1-4: Sơ đồ khối hệ thống đo nhịp tim**

### 1.2.1. Khối chuyển đổi nguồn DC – DC

Khối nguồn DC bao gồm 4 mức điện áp:  $V_{in}$  (điện áp của pin Lithium), 5V, 3.3V và 1.8V.

Khối nguồn được thiết kế để cung cấp nhiều mức điện áp khác nhau nhằm cấp nguồn cho các thiết bị hoạt động ở các điện mức khác nhau. Khối sử dụng các IC ổn áp để thay đổi các mức điện áp trong mạch.

### 1.2.2. Khối điều khiển

Khối điều khiển là tiếp nhận dữ liệu cũng như tín hiệu của các khối khác để xử lý từ đó điều khiển hoạt động của toàn hệ thống. Trong đề tài này, khối thực hiện nhận dữ liệu từ khối cảm biến để xử lý trả về thông số đo nhịp tim.

### 1.2.3. Khối cảm biến

Khối cảm biến có chức năng thu thập các tín hiệu cụ thể là mật độ ánh sáng để gửi dữ liệu về cho khối điều khiển xử lý các dữ liệu này.

## 2. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

### 2.1. Yêu cầu thiết kế

Hệ thống sau khi được thiết kế cần thỏa mãn các yêu cầu sau:

- Thiết kế phần cứng hoàn chỉnh của hệ thống sử dụng STM32
- Xây dựng mạch nguyên lý ( Schematic), thiết kế mạch nguyên lý.
- Thiết kế mạch in PCB, thực hiện thi công mạch in.
- Đảm bảo mạch in hoạt động đúng với thiết kế.

### 2.2. Phân tích thiết kế

Hệ thống được thiết kế, phát triển theo hai hướng như sau:

- **Sử dụng các module được thiết kế sẵn:** Sử dụng các module được thiết kế sẵn, dùng các loại dây nối để kết nối các thiết bị ngoại vi, MCU với nhau trên mạch breadboard
- **Thiết kế mạch in PCB:** Thiết kế lại toàn bộ hệ thống trên phần mềm chuyên dụng để thiết kế PCB. Các thiết bị được hàn trực tiếp trên mạch và được kết nối bằng các đường dẫn được thiết kế trên mạch.

#### 2.2.1. Ưu điểm của các phương pháp

**Sử dụng module:** Các module được thiết kế sẵn rất đa dạng, phong phú để lựa chọn. Các thiết kế đã được kiểm tra chính xác trước khi sản xuất nên hạn chế tình trạng mạch thiết kế bị lỗi. Được thiết kế để thuận tiện kết nối tới các thiết bị khác.

**Thiết kế mạch PCB:** Tối ưu được kích thước mạch cũng như hiệu năng do chỉ thiết kế những chức năng cần thiết trên mạch. Đường dây trên mạch ổn định không bị xê dịch nên hạn

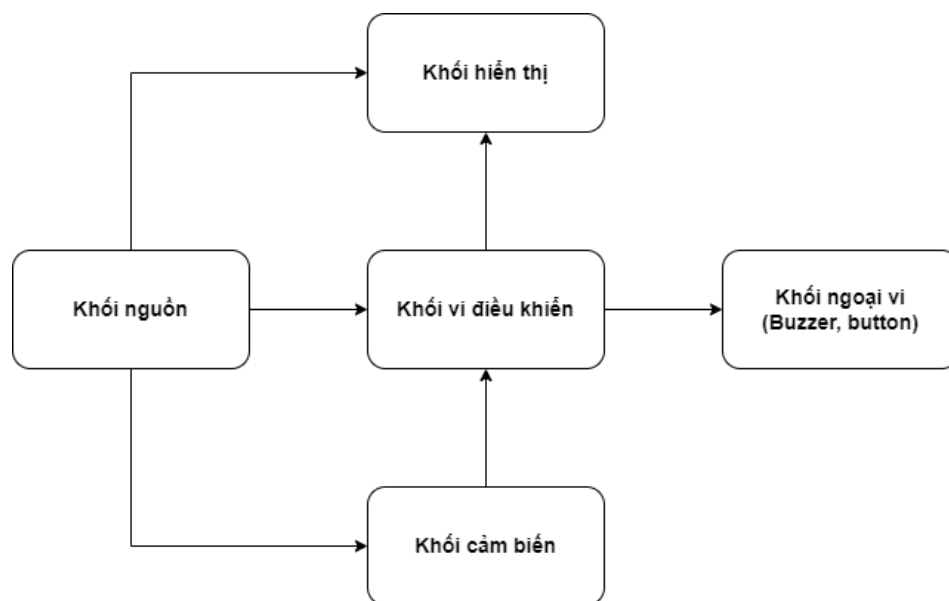
chế được vấn đề tín hiệu bị gián đoạn. Bên cạnh đó, việc thiết kế mạch PCB có thể linh hoạt trong việc lựa chọn các linh kiện cũng như thông số tùy vào mục đích của người thiết kế.

### 2.2.2. Nhược điểm của các phương pháp

**Sử dụng module:** Việc kết nối giữa các module bằng dây nối có thể gây tín hiệu bị gián đoạn do quá trình sử dụng đường dây có thể bị lỏng. Bên cạnh đó, việc kiểm tra và sửa lỗi trên các module gặp nhiều hạn chế do không nắm rõ toàn bộ mạch nguyên lý của mạch. Ngoài ra, các chức năng và linh kiện trên module được thiết kế cố định nên khó khăn trong việc phát triển vào ứng dụng khác nhau.

**Thiết kế mạch PCB:** Việc thực hiện mạch in yêu cầu người thiết kế phải nắm rõ cách sử dụng các phần mềm để thiết kế (Altium, Kicad, Orcad, ...). Tiếp đó, cần phải có kiến thức tốt về mạch điện để thiết kế mạch nguyên lý. Cuối cùng, việc thiết kế mạch PCB đòi hỏi cần phải tỉ mỉ, cẩn thận vì khi thiết kế sai sẽ khó để sửa lại thành phẩm mà cần phải thiết kế lại từ đầu.

### 2.3. Sơ đồ khối tổng quát



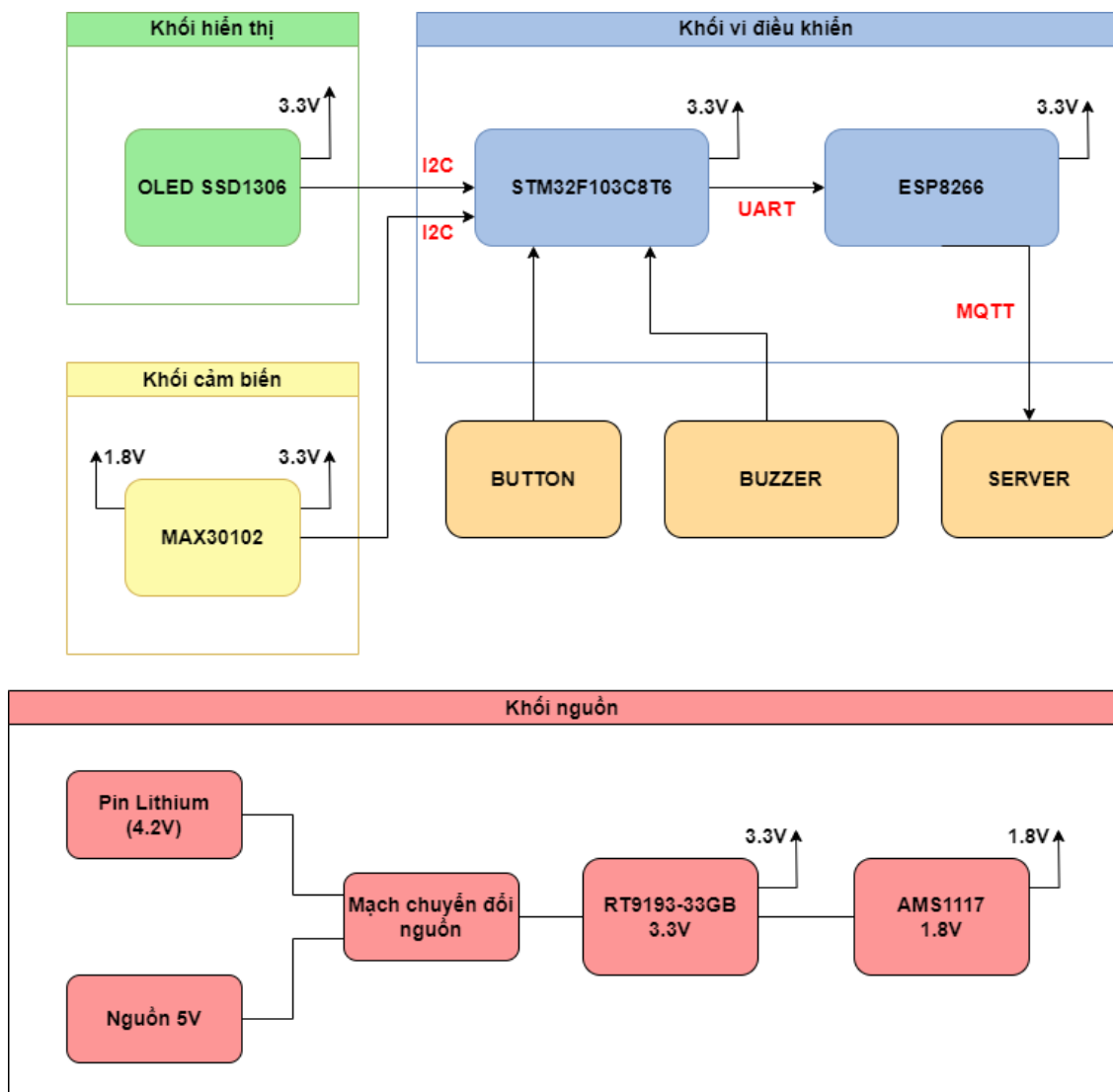
**Hình 2-1: Sơ đồ khối tổng quát của hệ thống**

Chi tiết các khối như sau:

- Khối nguồn: Cung cấp nguồn điện ổn định cho khối cảm biến, khối hiển thị và các ngoại vi.

- Khối vi điều khiển: Tiếp nhận, xử lý các dữ liệu từ khối cảm biến. Điều khiển các khối hiển thị và các ngoại vi.
- Khối hiển thị: Hiển thị thông tin hoạt động của hệ thống, được điều khiển bởi khối vi xử lý.
- Khối cảm biến: tiếp nhận tín hiệu nhịp tim từ đầu ngón tay người.
- Khối ngoại vi: Các nút nhấn chuyển tiếp giữa các chế độ khác nhau, buzzer giúp thông báo hệ thống đang hoạt động.

## 2.4. Sơ đồ khối chi tiết



Hình 2-2: Sơ đồ khối chi tiết của hệ thống



### **2.4.1. Khối nguồn**

Bao gồm một mạch chuyển đổi giữa nguồn từ pin Lithium (khoảng 4.2VDC) với nguồn ngoài 5VDC. Bên cạnh đó sử dụng hai IC nguồn là RT9193 – 33GB và AMS1117 – 1.8 để hạ mức điện áp xuống các mức, lần lượt là 3.3VDC và 1.8VDC.

### **2.4.2. Khối vi điều khiển**

Bao gồm hai dòng vi điều khiển là STM32F103C8T6 và Node MCU ESP8266. Mỗi dòng có chức năng như sau:

Với STM32F103C8T6 được thiết kế với các thành phần khác như thạch anh 8MHz, cổng nạp chương trình SWD, cổng USB. Khối này có chức năng giao tiếp và xử lý tín hiệu từ các thiết bị ngoại vi khác từ đó điều khiển và hiển thị dữ liệu sau khi được xử lý lên khối hiển thị. Bên cạnh đó, IC STM32F103C8T6 còn chịu trách nhiệm truyền dữ liệu cho ESP8266 thông qua giao thức UART.

Trong khi đó với Node MCU ESP8266 đảm nhận việc nhận dữ liệu từ STM32F103C8T6 và truyền nó lên server để người dùng dễ dàng kiểm tra. Chúng truyền nhận thông qua giao thức MQTT.

### **2.4.3. Khối cảm biến**

Bao gồm một cảm biến quang học MAX30102. Chịu trách nhiệm đo các thông số như nhịp tim, độ bão hoà oxy trong máu. MAX30102 giao tiếp với khối vi điều khiển thông qua giao thức I2C.

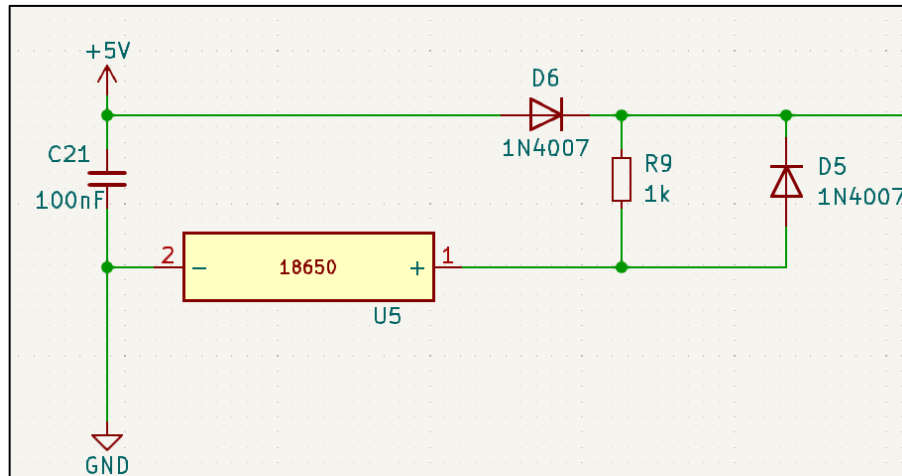
### **2.4.4. Khối hiển thị**

Khối hiển thị bao gồm một con màn hình OLED, nó giao tiếp với khối điều khiển thông qua giao thức I2C. Có chức năng hiển thị dữ liệu cần thiết cho người dùng dễ dàng quan sát kết quả.

## **2.5. Lựa chọn các linh kiện và thiết kế mạch nguyên lý**

### **2.5.1. Khối nguồn**

Để tối ưu cũng như tiện lợi trong quá trình cấp nguồn cho thiết bị. Cần thiết kế một mạch để chuyển đổi qua lại giữa hai nguồn ngõ vào là điện áp từ pin Lithium (khoảng 4.2VDC) và nguồn 5VDC.

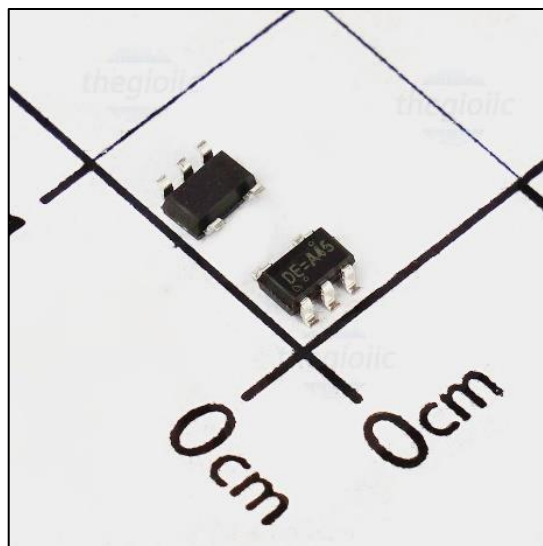


**Hình 2-3: Sơ đồ nguyên lý mạch chuyển đổi nguồn cấp cho hệ thống**

Từ điện áp  $V_{in}$ , điện áp tiếp tục được hạ xuống các mức điện áp 3.3V và 1.8V nhằm sử dụng cho các thiết bị với các mức điện áp hoạt động khác nhau.

#### **Khối nguồn 3.3VDC:**

Khối nguồn này cấp điện áp cho hầu hết các thiết bị trong hệ thống bao gồm: STM32F103C8T6, cảm biến MAX30102, OLED. Chính vì sự quan trọng của nó, cần thiết kế mạch sao cho nguồn điện được ổn định nhất có thể, hạn chế tình trạng rã cưa, nhiễu ở ngõ ra và nguồn dòng ngõ ra đủ cung cấp cho hệ thống. Do đó chọn IC ổn áp RT9193 – 33GB với độ ổn định điện áp ngõ ra tốt và nhiễu thấp.

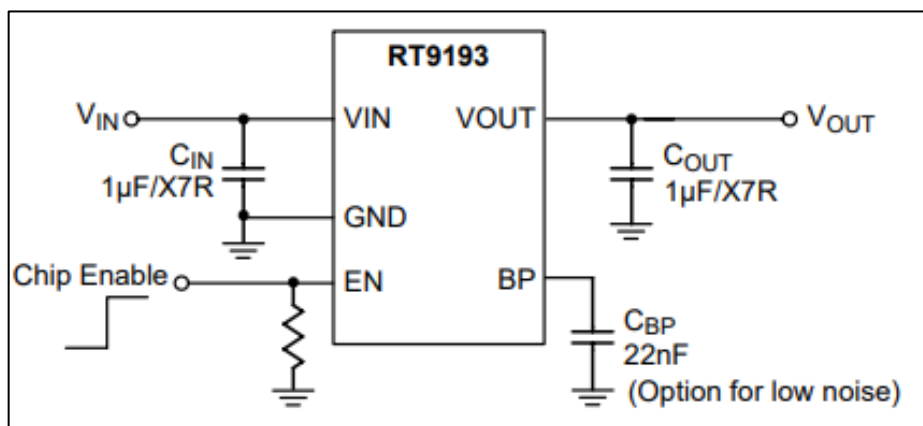


**Hình 2-4: IC Ổn Áp RT9193 – 33GB 3.3V**

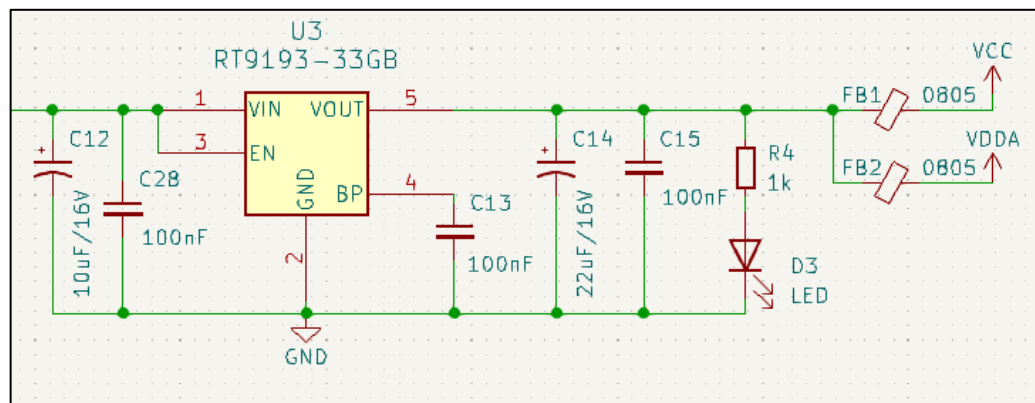
**Bảng 2-1: Thông số IC ổn áp RT9193 – 33GB**

Thuộc tính	Giá trị
Điện áp ngõ vào ( max )	5.5V
Điện áp ngõ ra	3.3V
Dòng điện ngõ ra	300mA

Dựa vào đề xuất thiết kế với IC của hãng, thiết kế lại mạch nguyên lý.



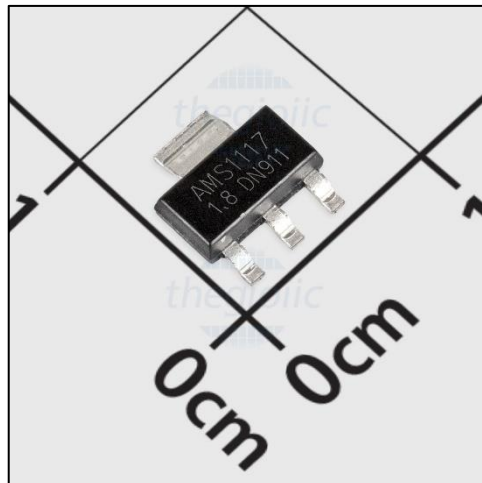
**Hình 2-5: Thiết kế đề xuất của IC ổn áp RT9193 – 33GB 3.3V**



**Hình 2-6: Sơ đồ nguyên lý mạch nguồn 3.3V**

### Khối nguồn 1.8V:

Khối này chủ yếu cung cấp điện áp cho IC MAX30102. Vì là IC chính trong việc cung cấp dữ liệu cho MCU nên cần cung cấp điện áp ổn định. Trong đồ án này sử dụng con IC AMS1117 – 1.8V.

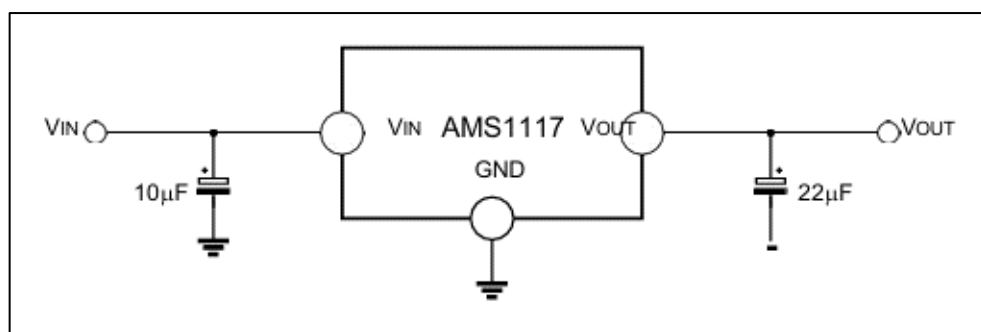


Hình 2-7: IC ổn áp AMS1117 – 1.8V

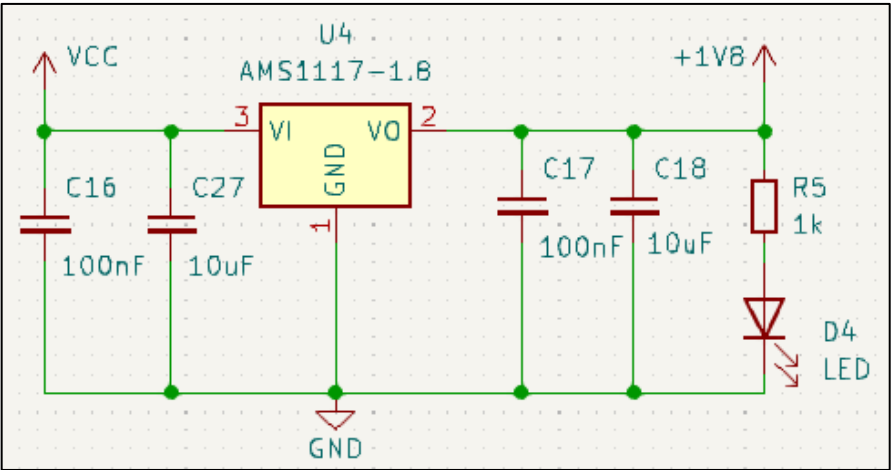
Bảng 2-2: Thông số IC ổn áp AMS1117 – 1.8V

Thuộc tính	Giá trị
Điện áp ngõ vào ( max )	15V
Điện áp ngõ ra	1.8V
Dòng điện ngõ ra	1A

Dựa vào đề xuất thiết kế với IC của hãng, thiết kế lại mạch nguyên lý.



Hình 2-8: Thiết kế đề xuất của IC ổn áp AMS1117 – 1.8V



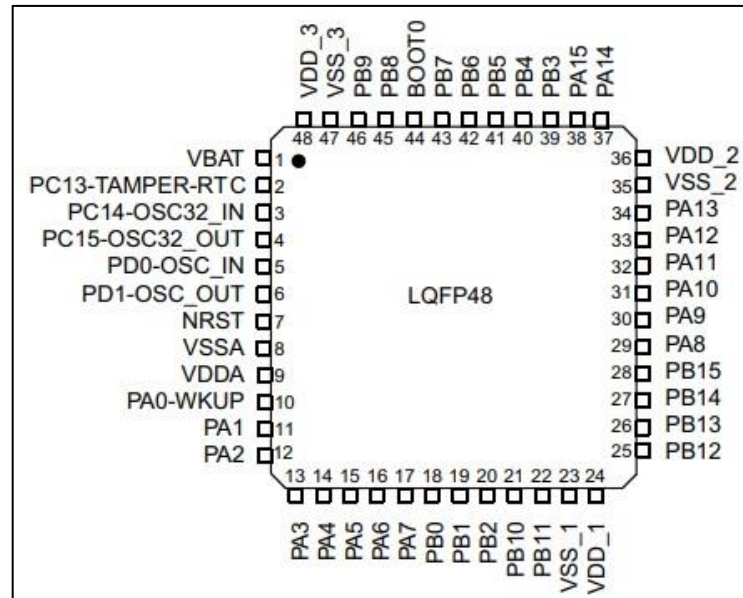
Hình 2-9: Sơ đồ nguyên lý mạch nguồn 1.8V

2.5.2. Khối vi điều khiển

2.5.2.1. STM32F103C8T6

Bảng 2-3: Thông số STM32F103C8T6

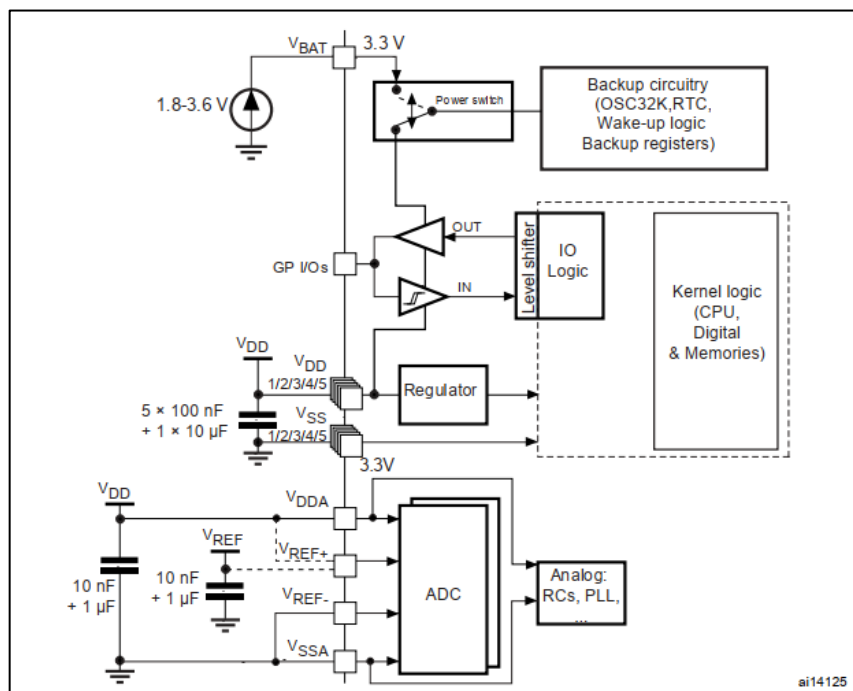
Thuộc tính	Giá trị
Họ IC	ARM Cortex M3
Tần số hoạt động	72MHz
Giao tiếp	CANbus, I2C, IrDA, LINbus, SPI, UART/USART, USB
Điện áp cấp	2 V ~ 3.6 V
Bộ nhớ	64K Byte Flash, 20K Byte SRAM



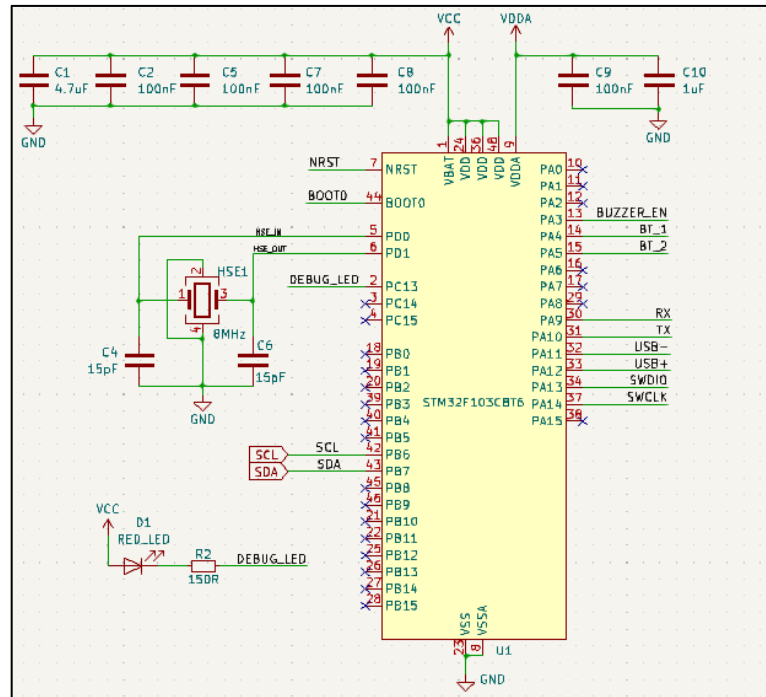
### Hình 2-10: Sơ đồ chân của STM32F103C8T6

STM32F103C8T6 cần thiết kế với các yêu cầu sau:

- Một bộ giao động thạch anh ngoại với tần số 8MHz
- Các tụ decoupling giữa các chân VDD và VSS được đề xuất như **Hình 2 – 11**.



### Hình 2-11: Đề xuất thiết kế cấp nguồn cho vi điều khiển STM32F103C8T6

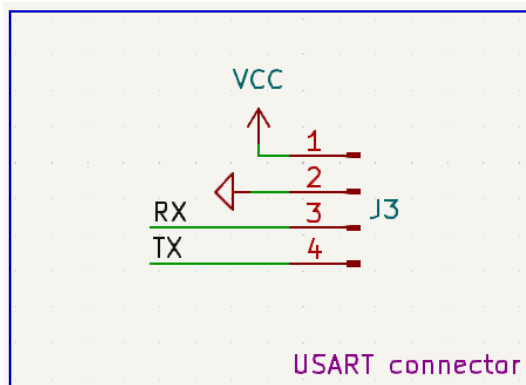


**Hình 2-12: Mạch nguyên lý của STM32F103C8T6**

#### 2.5.2.2. ESP8266

ESP8266 là một vi xử lý tích hợp mạnh mẽ đặc biệt được thiết kế cho nhu cầu Internet of thing (IOT). Nó cung cấp một giải pháp kết nối mạng Wi-fi đầy đủ và khép kín, cho phép nó có thể lưu trữ các ứng dụng để giảm tải các chức năng kết nối mạng Wi-fi từ bộ xử lý ứng dụng, cho phép ESP8266 được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng khác thông qua GPIOs với một chi phí và một PCB nhỏ. ESP8266 là một chip được sử dụng khá nhiều trong các sản phẩm IOT.

Trong đồ án này sử dụng trực tiếp module ESP có sẵn. Thiết kế header để cấp nguồn và đường dây tín hiệu cho ESP8266.



**Hình 2-13: Thiết kế header để cấp nguồn và đường dây tín hiệu ra ESP8266**



**Hình 2-14: Module Node MCU – ESP8266**

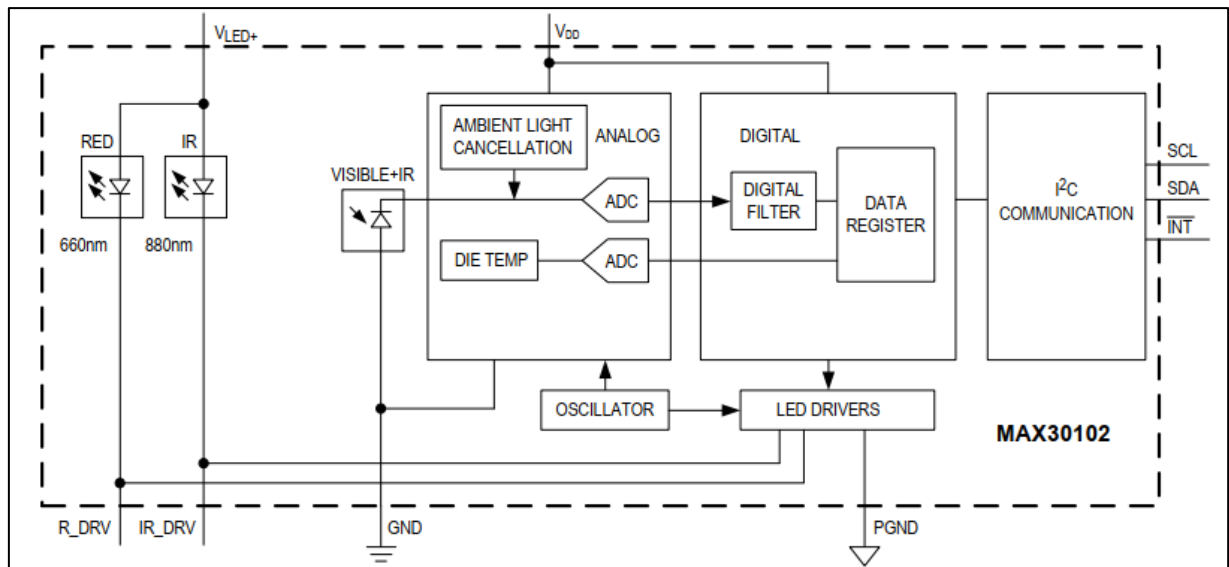
**Bảng 2-4: Thông số kỹ thuật ESP8266**

Thông số kỹ thuật	
Bộ vi điều khiển	CPU RISC 32-bit Tensilica Xtensa LX106
Điện áp hoạt động	3.3V
Chân I / O kỹ thuật số (DIO)	16
Chân đầu vào tương tự (ADC)	1
Giao tiếp	1 UARTs, 1 SPI, 1 I2Cs
Bộ nhớ Flash	4 MB
SRAM	64KB
Tốc độ đồng hồ	80 MHz

### **2.5.3. Khối cảm biến**

Khối cảm biến bao gồm IC MAX30102. Đây là IC có chức năng đo độ bão hoà oxy trong máu kèm tích hợp đo nhịp tim. Nó tích hợp hai đèn LED bên trong lần lượt là đỏ và tia hồng ngoại. Ngoài ra còn có bộ tách sóng quang, là một yếu tố quang học và thiết bị điện tử có độ nhiễu thấp với khả năng loại bỏ ánh sáng xung quanh.





Hình 2-15: Các khối chức năng của IC MAX30102

Các chân của IC có các chức năng được mô tả như sau:

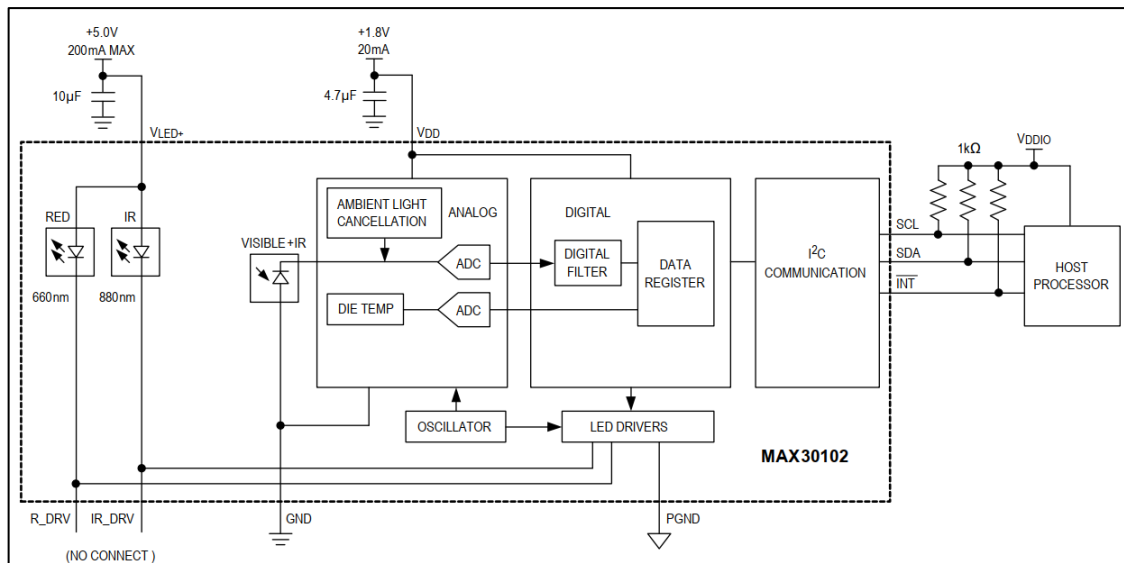


Hình 2-16: Cấu hình chân IC MAX30102

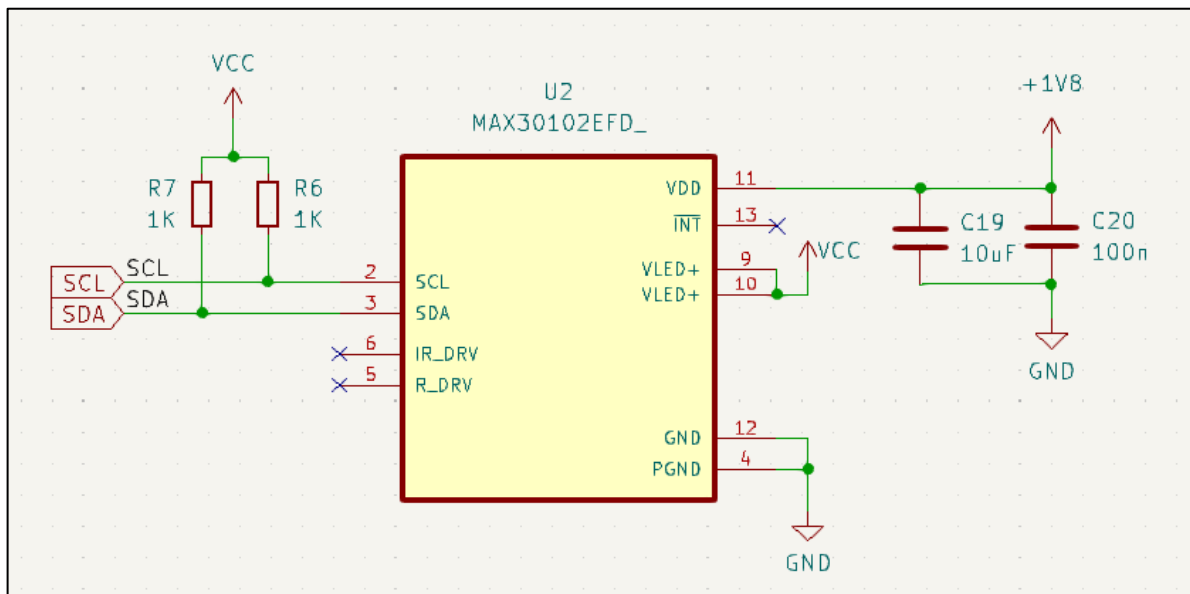
**Bảng 2-5: Bảng mô tả chức năng chân IC MAX30102**

Chân	Tên	Chức năng
1, 7, 8, 14	N.C.	Không kết nối. Kết nối với Pad trên PCB để ổn định linh kiện trên mạch
2	SCL	Ngõ vào xung clock I <sup>2</sup> C
3	SDA	Ngõ dữ liệu, hai chiều (Open – Drain)
4	PGND	Điện áp đất cho khối điều khiển đèn LED
5	R_DRV	Điều khiển đèn LED màu đỏ
6	IR_DRV	Điều khiển đèn LED tia hồng ngoại
9	V <sub>LED+</sub>	Cấp nguồn cho LED (kết nối anode). Sử dụng tụ bypass tới PGND cho hiệu suất tốt nhất
10	V <sub>LED+</sub>	
11	V <sub>DD</sub>	Điện áp ngõ vào. Sử dụng tụ bypass tới GND để cho hiệu năng tốt nhất
12	GND	Nguồn đất
13	$\overline{\text{INT}}$	Ngắt tích cực mức thấp (Open – Drain). Kết nối tới chân điện áp ngoài với điện trở kéo lên

Thiết kế khối cảm ứng dựa vào thiết kế khuyến nghị từ nhà sản xuất như sau:



Hình 2-17: Thiết kế đề xuất cho IC MAX30102



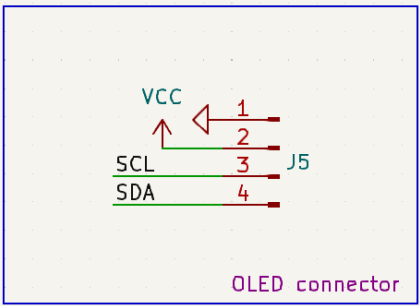
Hình 2-18: Mạch nguyên lý cho khối cảm biến IC MAX30102

#### 2.5.4. Khối hiển thị

Khối hiển thị sử dụng module OLED SSD1306 0.91inch với các thông số và thiết kế nguyên lý như sau:

Bảng 2-6: Thông số OLED SSD1306

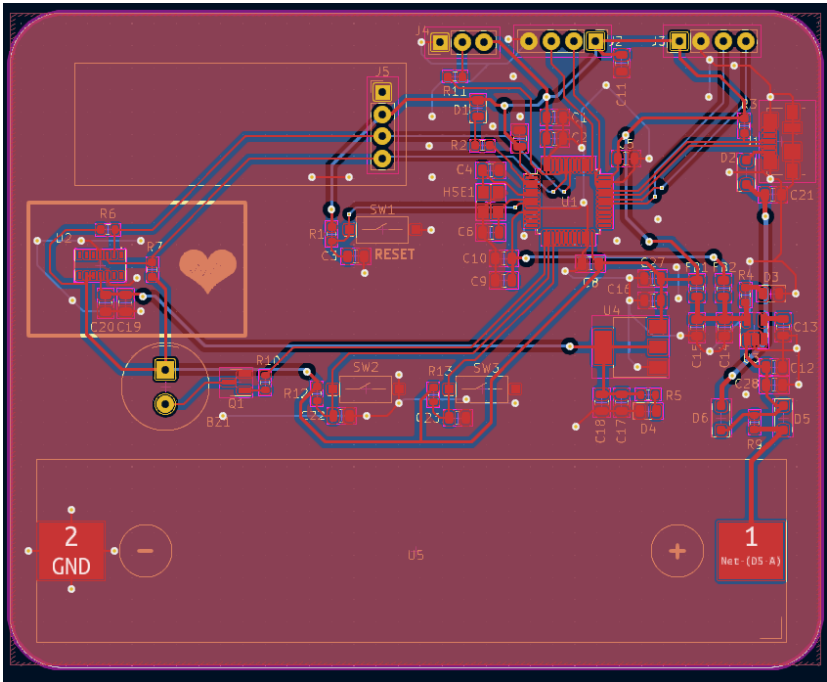
Thông số kỹ thuật	
IC Driver	SSD1306
Độ phân giải	128 x 32 pixels
Loại giao tiếp	I2C
Điện áp hoạt động	3.3 / 5VDC



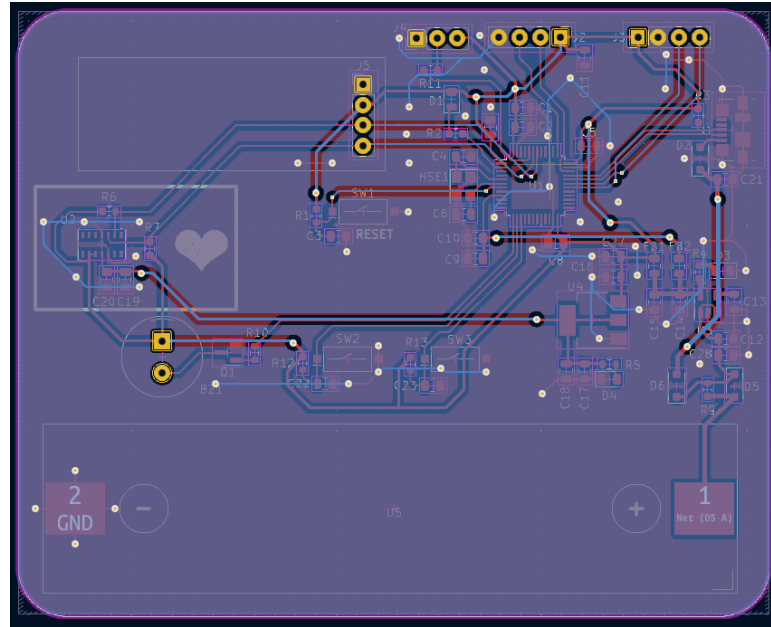
Hình 2-19: Thiết kế nguyên lý cho OLED SSD1306

2.6. Thiết kế và bố trí linh kiện – Layout PCB

2.6.1. Tổng quan thiết kế bố trí linh kiện – Layout:



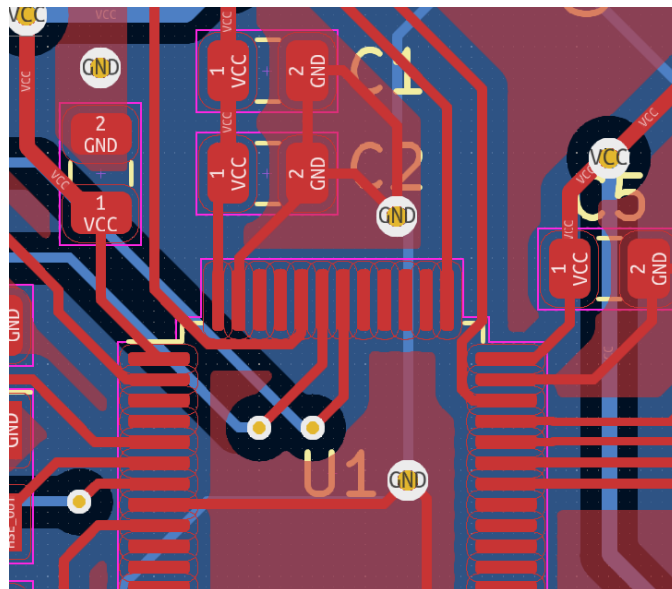
Hình 2-20: Thiết kế Layout (lớp trên)



Hình 2-21: Thiết kế Layout (lớp dưới)

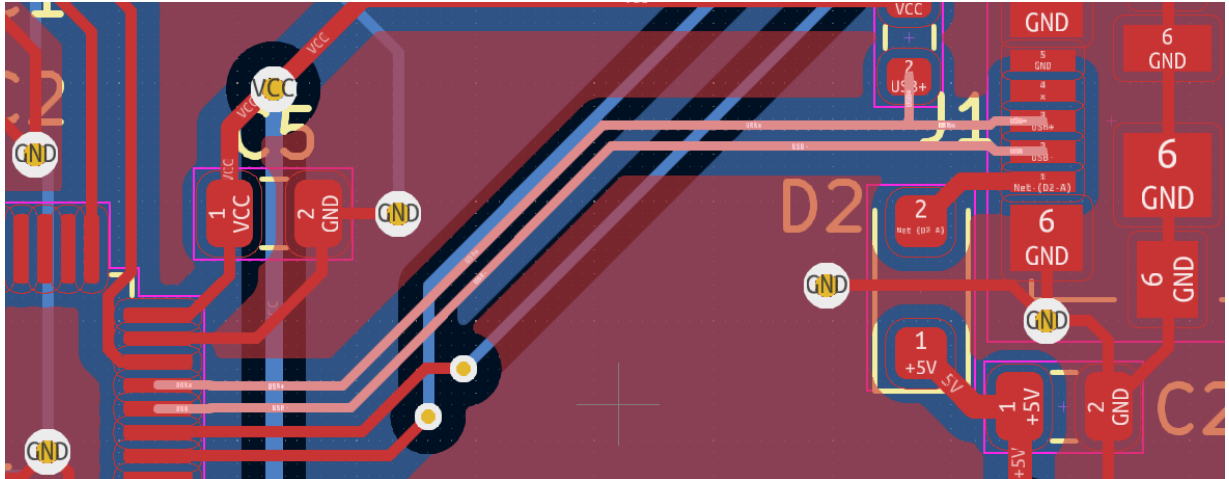
### 2.6.2. Những lưu ý khi thiết kế

- Các tụ decoupling cần được đặt gần các chân nguồn của MCU nhất có thể.
- Vị trí của thạch anh cần phải tối ưu sao cho đường dây là ngắn nhất.



Hình 2-22: Khối MCU, các tụ nguồn và thạch anh (mặt trên)

Đối với thiết kế đường dây dữ liệu cho Micro USB, mong muốn truyền được với tốc độ full speed ở tần số  $f = 48\text{MHz}$ . Chính vì vậy cần thiết kế đường dây vi sai (differential pair) cho hai đường dây tín hiệu DP và DM để có thể đảm bảo tính tốc độ và tính chính xác, chống nhiễu xuyên kênh.



Hình 2-23: Đường dây vi sai (differential pair) cho hai đường dây tín hiệu

### 3. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

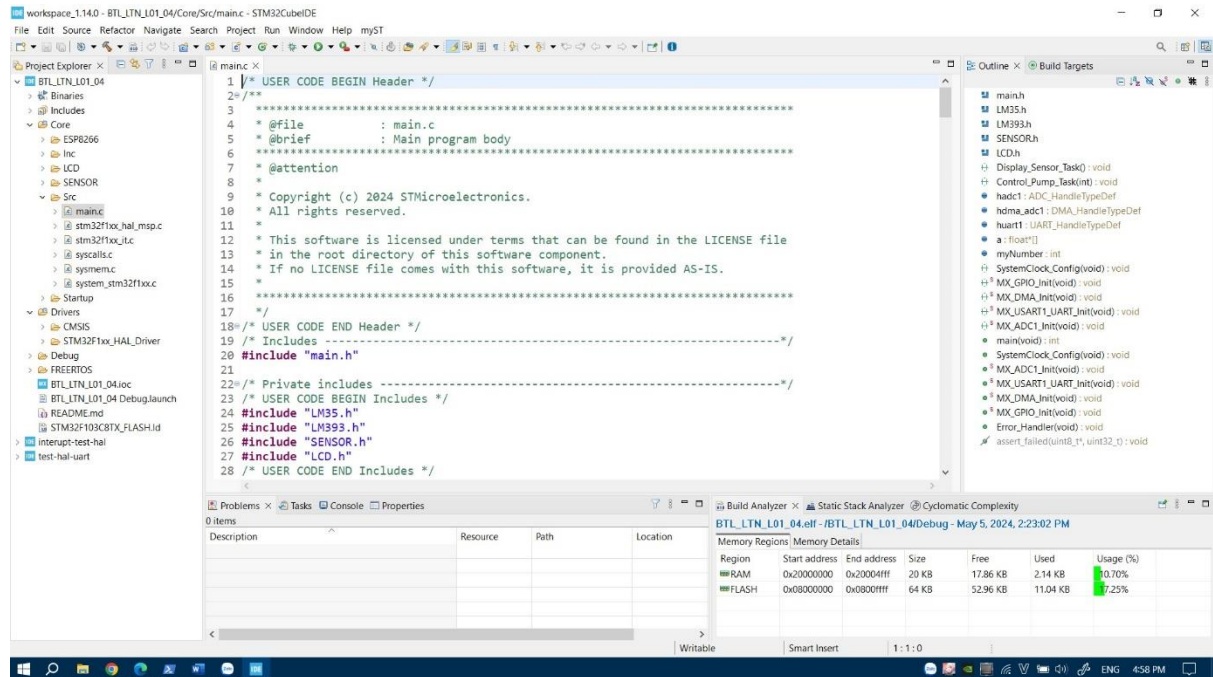
#### 3.1. Yêu cầu cho hệ thống

Để thiết kế phần mềm (firmware) cho hệ thống trên cần có những yêu cầu cụ thể như sau:

- Thực hiện phân lập trình trên phần mềm STM32CubeIDE bằng ngôn ngữ lập trình C.
- Xây dựng lưu đồ giải thuật tổng quát, lưu đồ chi tiết cho từng khối.
- Từ lưu đồ giải thuật, thực thi lập trình từng khối và tổng hợp thành chương trình hoàn thiện.
- Kiểm chứng, so sánh các giá trị đo được từ thiết kế so với các số liệu thực tế hoặc thiết bị khác. Yêu cầu sai số không quá 5%.

## 3.2. Giới thiệu các phần mềm và nền tảng

### 3.2.1. STM32CubeIDE



Hình 3-1: Giao diện STM32CubeIDE

STM32CubeIDE là một công cụ phát triển phần mềm đa hệ điều hành, tất cả trong một, là một phần của hệ sinh thái phần mềm STM32Cube.

Đây là nền tảng phát triển C/C++ tiên tiến với các tính năng như cấu hình ngoại vi, tạo mã, biên dịch mã và gỡ lỗi cho dòng vi điều khiển và vi xử lý STM32. Nó dựa trên Eclipse®/CDT™ framework, GCC toolchain để phát triển và GDB để gỡ lỗi.

STM32CubeIDE tích hợp các chức năng tạo dự án và cấu hình STM32 từ STM32CubeMX. Sau khi chọn lựa giữa STM32 MCU trống, hoặc được cấu hình sẵn, dự án sẽ được tạo và các đoạn mã khởi tạo được tự động sinh ra. Bất cứ lúc nào trong quá trình phát triển, người dùng có thể quay lại quá trình khởi tạo và cấu hình của các thiết bị ngoại vi hoặc phần mềm trung gian và tạo lại các mã khởi tạo mà không ảnh hưởng đến mã của người dùng.

STM32CubeIDE bao gồm các bộ phân tích thông số bản dựng và vùng nhớ stack nhằm cung cấp cho người dùng thông tin cần thiết về trạng thái của dự án và các yêu cầu bộ nhớ.

Nó cũng bao gồm các tính năng gỡ lỗi tiêu chuẩn và nâng cao như chế độ xem các thanh ghi lỗi CPU, bộ nhớ và thanh ghi ngoại vi, cũng như theo dõi biến số trực tiếp, giao diện Serial Wire Viewer hoặc bộ phân tích lỗi.

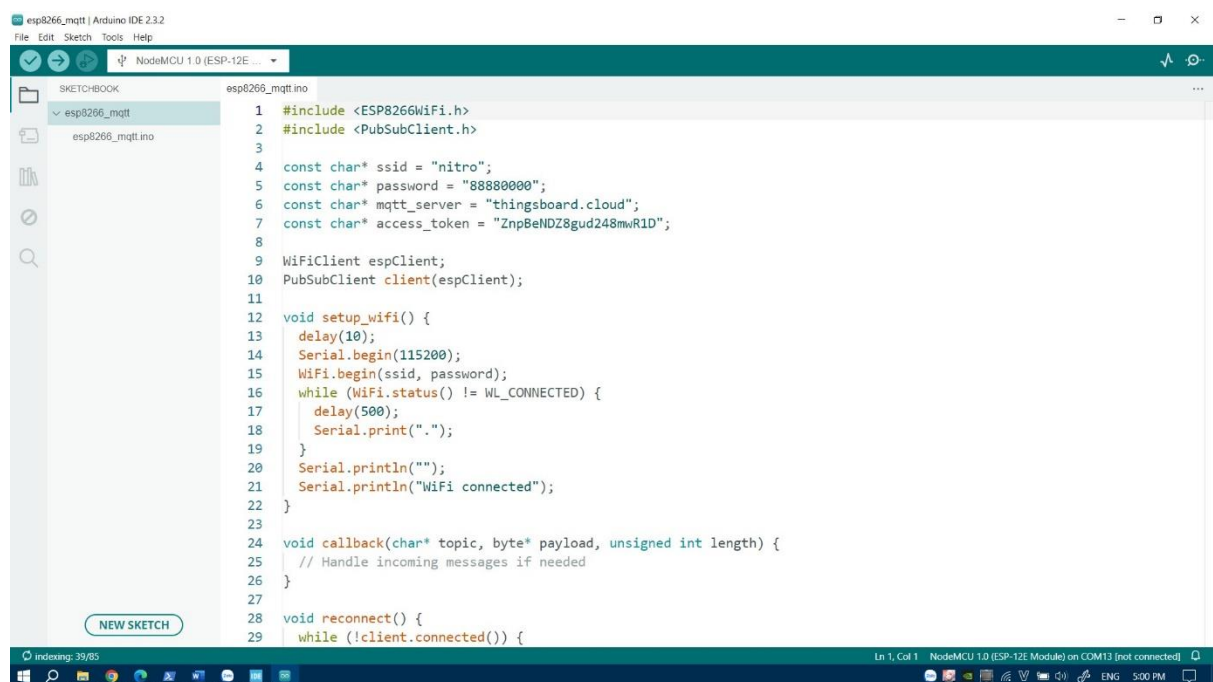
### 3.2.2. *Arduino IDE*

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã cho các module Arduino hoặc các module khác có hỗ trợ chạy code Arduino.

Đây là phần mềm này miễn phí tải về và không yêu cầu bản quyền. Do đó người dùng có quyền sửa đổi, cải tiến, phát triển, nâng cấp theo một số nguyên tắc chung được nhà phát hành cho phép mà không cần xin giấy phép.

Arduino IDE hỗ trợ ngôn ngữ lập trình C/C++ vốn rất phổ biến trong lập trình nhúng.

Arduino IDE hoạt động trên 3 hệ điều hành phổ biến là Windows, Mac OS và Linux.

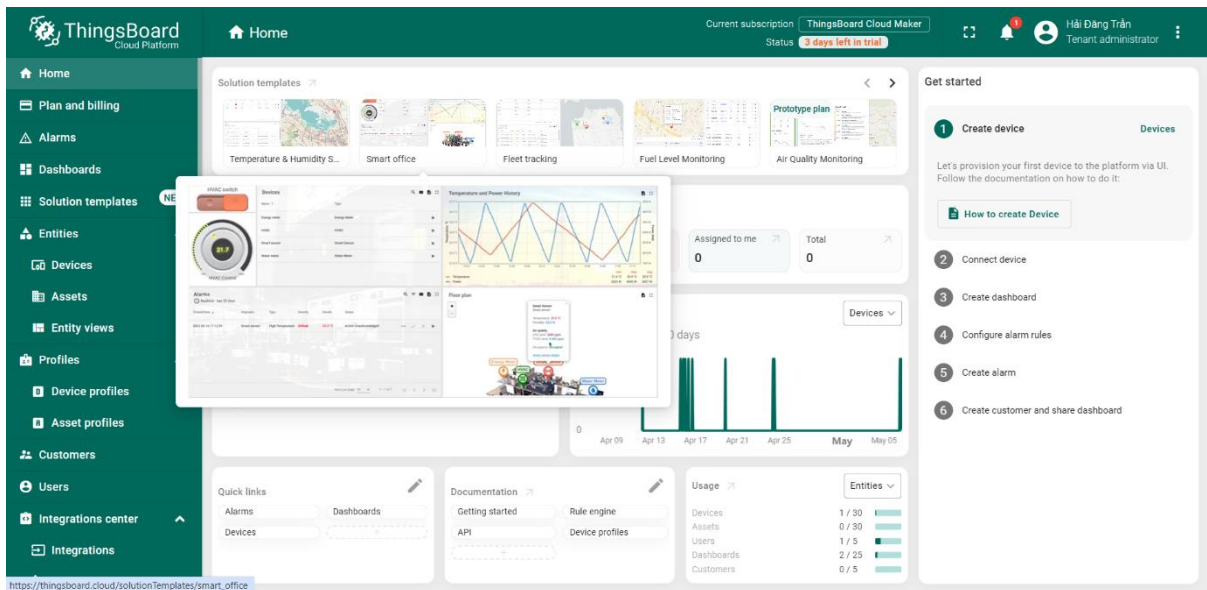


Hình 3-2: Giao diện ArduinoIDE

### 3.2.3. *Thingsboard*

Thingsboard là một nền tảng mã nguồn Internet of Things (IoT) mở, cho phép các tổ chức tạo ra, quản lý và theo dõi các ứng dụng IoT một cách hiệu quả. Với Thingsboard, người dùng có thể kết nối và quản lý hàng ngàn thiết bị IoT khác nhau từ xa, thu thập dữ liệu, hiển thị thông tin và thậm chí tương tác với các thiết bị đó.





**Hình 3-3: Giao diện Thingsboard**

### **Các tính năng nổi bật của Thingsboard**

**Hỗ trợ đa ngôn ngữ:** Thingsboard là nền tảng IoT mạnh mẽ và linh hoạt với khả năng hỗ trợ đa ngôn ngữ. Điều này giúp người dùng trên khắp thế giới dễ dàng sử dụng, đồng thời tương tác với hệ thống một cách thuận tiện. Nhờ khả năng hỗ trợ nhiều ngôn ngữ khác nhau, Thingsboard giúp tạo ra trải nghiệm người dùng tốt và chuyên nghiệp.

**Giao diện người dùng thân thiện:** Một trong những ưu điểm nổi bật của Thingsboard chính là giao diện người dùng thân thiện. Với thiết kế đơn giản, dễ sử dụng và trực quan, người dùng có thể dễ dàng thao tác cũng như quản lý hệ thống IoT một cách hiệu quả. Giao diện người dùng của Thingsboard được tối ưu hóa để cung cấp trải nghiệm tốt nhất cho người dùng, từ những người mới bắt đầu đến những chuyên gia.

**Hỗ trợ đa nền tảng:** Thingsboard cung cấp khả năng hỗ trợ đa nền tảng, giúp các doanh nghiệp dễ dàng tích hợp với các thiết bị và ứng dụng khác nhau. Với ưu điểm là độ tương thích rộng lớn, Thingsboard cho phép người dùng kết nối và quản lý dữ liệu từ nhiều nguồn khác nhau một cách dễ dàng và hiệu quả.

**Hiệu suất cao:** Nhờ thiết kế tối ưu và khả năng mở rộng linh hoạt, Thingsboard cung cấp hiệu suất cao trong việc xử lý dữ liệu IoT. Hệ thống của Thingsboard được tối ưu hóa để xử lý lượng dữ liệu lớn mà vẫn đảm bảo độ tin cậy và tốc độ phản hồi nhanh chóng. Điều này giúp doanh nghiệp tiết kiệm thời gian và tăng cường hiệu quả trong việc quản lý dữ liệu IoT.

**Bảo mật:** Thingsboard cung cấp hệ thống bảo mật mạnh mẽ, cho phép người dùng kiểm soát quyền truy cập và quản lý dữ liệu một cách an toàn. Hệ thống bảo mật của Thingsboard bao gồm mã hóa dữ liệu, xác thực người dùng và giám sát liên tục để đảm bảo tính toàn vẹn của hệ thống.

**Tổng quan và trực quan hóa dữ liệu:** Thingsboard cung cấp một giao diện trực quan và dễ sử dụng cho người dùng để theo dõi cũng như quản lý dữ liệu hiệu quả. Người dùng có thể tạo các bảng điều khiển tùy chỉnh để theo dõi các thông số quan trọng và hiển thị dữ liệu theo cách mong muốn.

**Hỗ trợ Rest API và RPC:** Thingsboard cung cấp hỗ trợ đầy đủ cho Rest API và RPC, giúp người dùng dễ dàng tương tác với hệ thống và thiết bị IoT. Việc này giúp tối ưu hóa quá trình phát triển ứng dụng và tích hợp thiết bị vào hệ thống một cách linh hoạt, hiệu quả.

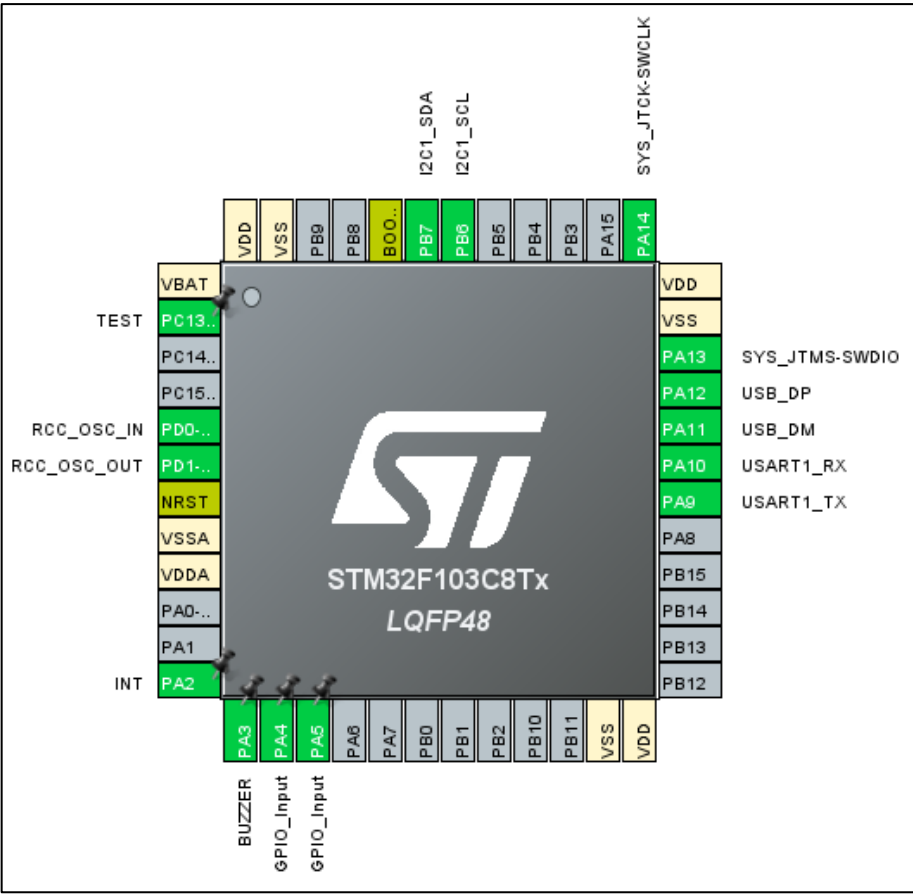
**Đẩy dữ liệu thiết bị:** Với Thingsboard, người dùng có thể đẩy dữ liệu từ các thiết bị IoT lên hệ thống một cách tự động và đồng nhất. Điều này giúp giảm thời gian cần thiết cho việc thu thập và xử lý dữ liệu, đồng thời tăng cường tính chính xác và đáng tin cậy của thông tin được hiển thị trên giao diện.

**Tích hợp với các hàng đợi tin khác nhau:** Thingsboard cho phép tích hợp với các hàng đợi tin nhắn khác nhau như Kafka, MQTT, RabbitMQ, giúp tối ưu hóa việc truyền tải dữ liệu giữa các thành phần trong hệ thống IoT. Điều này giúp cải thiện hiệu suất và độ tin cậy của hệ thống, đồng thời tạo ra một môi trường linh hoạt cho việc phát triển cũng như mở rộng ứng dụng IoT.

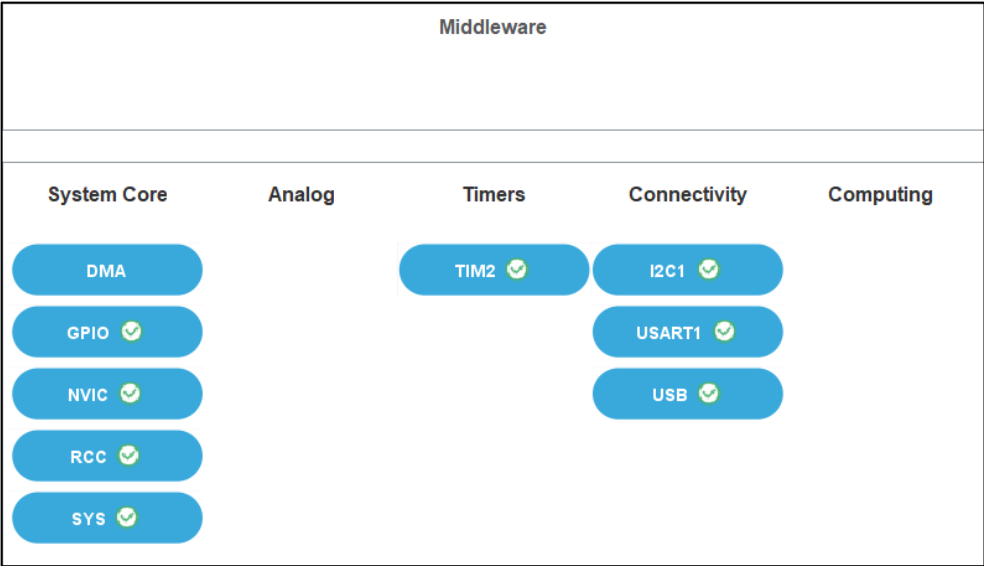
**Hỗ trợ cơ sở dữ liệu kết hợp:** Thingsboard cung cấp khả năng tương thích với nhiều loại cơ sở dữ liệu khác nhau, từ SQL đến NoSQL, giúp người dùng linh hoạt trong việc lưu trữ và truy xuất dữ liệu theo nhu cầu cụ thể của họ.

**Hỗ trợ nhiều chuẩn giao tiếp:** Ngoài việc hỗ trợ các tiêu chuẩn ngành như MQTT, CoAP và HTTP, Thingsboard còn có khả năng tùy chỉnh cụm theo yêu cầu cụ thể. Nhờ đó mà Thingsboard đáp ứng được đa dạng nhu cầu của người dùng từ các lĩnh vực khác nhau.

3.3. Cấu hình trên STM32CubeIDE cho STM32F103C8T6



Hình 3-4: Cấu hình chân hệ thống trên STM32CubeIDE (Pinout view)



Hình 3-5: Cấu hình các chức năng sử dụng (system view)

### 3.3.1. Cấu hình GPIO

**Bảng 3-1: Bảng danh sách các chân cấu hình GPIO và cấu hình chế độ.**

Tên chân	Chế độ GPIO	GPIO Pull – up/ Pull – down	Nhãn (User label)
PA4	Ngắt ngoài chế độ phát hiện xung cạnh xuống	Pull – up	CHANGE_DISPLAY
PA9	Ngõ ra kiểu Push – Pull	Không	BUCK_EN
PB0	Ngõ ra kiểu Push – Pull	Không	LOAD_CTR
PC1	Ngõ ra kiểu Push – Pull	Không	TEST

### 3.3.2. Cấu hình Timer2 tạo xung PWM

**Bảng 3-2: Bảng danh sách cấu hình Timer 2 tạo PWM**

Tên chân	Tín hiệu trên chân	Chế độ GPIO	Nhãn (User label)
PA3	TIM2_CH4	Alternate Function Push Pull	BUZZER

**Bảng 3-3: Cấu hình thông số cho Timer 2 tạo xung PWM**

<b>Counter Settings</b>	<ul style="list-style-type: none"> <li>– <i>Prescaler:</i> 1</li> <li>– <i>Counter mode:</i> Up</li> <li>– <i>Internal Clock Division:</i> No Division</li> </ul>
<b>PWM Generation Channel 4</b>	<ul style="list-style-type: none"> <li>– <i>Mode:</i> PWM mode 1</li> <li>– <i>Pulse:</i> 0</li> <li>– <i>Output compare preload:</i> Enable</li> </ul>

### 3.3.3. Cấu hình giao thức I2C

**Bảng 3-4: Bảng danh sách cấu hình giao thức I2C**

Tên chân	Tín hiệu trên chân	Chế độ GPIO	Nhãn (User label)
----------	--------------------	-------------	-------------------

PB6	I2C1_SCL	Alternate Function Open Drain	I2C1_SCL
PB7	I2C_SDA	Alternate Function Open Drain	I2C_SDA

**Bảng 3-5: Cấu hình thông số cho giao thức I2C**

<b>Master Features</b>	<ul style="list-style-type: none"> <li>– <b>I2C Speed Mode:</b> Fast mode</li> <li>– <b>I2C Clock Speed:</b> 400000Hz</li> </ul>
------------------------	--

### 3.3.4. Cấu hình giao thức UART

**Bảng 3-6: Bảng danh sách cấu hình giao thức UART**

Tên chân	Tín hiệu trên chân	Chế độ GPIO	Nhãn (User label)
PA10	USART1_RX	Input Mode	USART1_RX
PA9	USART1_TX	Alternate Function Push Pull	USART1_TX

**Bảng 3-7: Cấu hình thông số cho giao thức I2C**

<b>Parameters</b>	<ul style="list-style-type: none"> <li>– <b>Baud Rate:</b> 115200 Bits/ s</li> <li>– <b>Word Length:</b> 8 bits</li> <li>– <b>Parity:</b> None</li> <li>– <b>Stop bits:</b> 1</li> <li>– <b>Data Direction:</b> Receive and transmit</li> <li>– <b>Over Sampling:</b> 16 Samples</li> </ul>
-------------------	---

### 3.4. MAX30102

#### 3.4.1. Thanh ghi trạng thái (Register Status)

STATUS											
Interrupt Status 1	A_FULL	PPG_RDY	ALC_OVF	PROX_INT				PWR_RDY	0x00	0x00	R
Interrupt Status 2							DIE_TEMP_RDY		0x01	0x00	R
Interrupt Enable 1	A_FULL_EN	PPG_RDY_EN	ALC_OVF_EN	PROX_INT_EN					0x02	0x00	R/W
Interrupt Enable 2							DIE_TEMP_RDY_EN		0x03	0x00	R/W
FIFO											
FIFO Write Pointer				FIFO_WR_PTR[4:0]					0x04	0x00	R/W
Overflow Counter				OVF_COUNTER[4:0]					0x05	0x00	R/W
FIFO Read Pointer				FIFO_RD_PTR[4:0]					0x06	0x00	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W

**Hình 3-6: Thông tin thanh ghi trạng thái**

##### 3.4.1.1. Thanh ghi trạng thái ngắt (0x00 – 0x01)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
Interrupt Status 1	A_FULL	PPG_RDY	ALC_OVF	PROX_INT				PWR_RDY	0x00	0x00	R
Interrupt Status 2							DIE_TEMP_RDY		0x01	0x00	R

**Hình 3-7: Thông tin thanh ghi ngắt**

#### **BIT 7 – A\_FULL:**

Trong chế độ SpO2 và HR, ngắt này được kích hoạt khi con trỏ ghi của FIFO còn lại một số không gian trống nhất định. Số lượng kích hoạt có thể được đặt bằng thanh ghi FIFO\_A\_FULL [3:0]. Ngắt được xóa bằng cách đọc thanh ghi Trạng thái Ngắt 1 (0x00).

#### **BIT 6 – PPG\_RDY:**

Trong chế độ SpO2 và HR, ngắt này được kích hoạt khi có một mẫu mới trong FIFO dữ liệu. Ngắt được xóa bằng cách đọc thanh ghi Trạng thái Ngắt 1 (0x00), hoặc bằng cách đọc thanh ghi FIFO\_DATA

#### **BIT 5 – ALC\_RDY (Ambient Light Cancellation Overflow):**

Ngắt này xảy ra khi chức năng khử ánh sáng xung quanh đạt đến giới hạn cực đại. Vì vậy nó đang ảnh hưởng đến ngõ ra của ADC. Ngắt này sẽ được xóa bởi việc đọc thanh ghi trạng thái ngắt 1(0x00)

#### **BIT 4 – PROX\_INT (Proximity Threshold Triggered):**

Ngắt gần ngưỡng kích hoạt xảy ra khi đạt được ngưỡng gần kích hoạt and chế độ SpO2/HR bắt đầu. Điều này cho phép bộ xử lý chủ biết để bắt đầu chạy thuật toán SpO2/HR và thu thập dữ liệu. Ngắt này được xóa bằng cách đọc thanh ghi Trạng thái Ngắt 1 (0x00).

#### **BIT 1 – DIE\_TEMP\_RDY (Internal Temperature Ready Flag):**

Khi quá trình chuyển đổi nhiệt độ bên trong hoàn tất, ngắt này được kích hoạt để bộ xử lý có thể đọc các thanh ghi dữ liệu nhiệt độ. Ngắt này được xóa bằng cách đọc thanh ghi Trạng thái Ngắt 2 (0x01) hoặc thanh ghi TFRAC (0x20).

#### **BIT 0 – PWR\_RDY (Power Ready Flag):**

Khi khởi động hoặc sau một điều kiện suy giảm điện áp, khi điện áp cung cấp VDD chuyển từ dưới điện áp khóa dưới áp (UVLO) lên trên điện áp UVLO, một ngắt sẵn sàng nguồn điện được kích hoạt để báo hiệu rằng mô-đun đã được cấp điện và sẵn sàng thu thập dữ liệu

#### *3.4.1.2. Thanh ghi cho phép ngắt*

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
Interrupt Enable 1	A_FULL_EN	PPG_RDY_EN	ALC_OVF_EN	PROX_INT_EN					0x02	0x00	R/W
Interrupt Enable 2							DIE_TEMP_RDY_EN		0x03	0x00	R/W

**Hình 3-8: Thông tin thanh ghi cho phép ngắt**

Mỗi nguồn ngắt phần cứng ngoại trừ ngắt sẵn sàng nguồn có thể bị vô hiệu hoá trong thanh ghi phần mềm. Ngắt nguồn thì không thể bị vô hiệu hoá bởi vì trạng thái của module sẽ khởi động lại khi xảy ra tình trạng thiếu điện (điện áp cung cấp thấp), và điều kiện mặc định là tất cả các ngắt đều bị vô hiệu hóa. Ngoài ra, điều quan trọng là hệ thống phải biết rằng tình trạng thiếu điện đã xảy ra và dữ liệu trong module đã được đặt lại do kết quả của việc này.

Các bit không sử dụng thì luôn luôn được cho về 0 để hoạt động bình thường.

## 3.4.1.3. FIFO

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
FIFO Write Pointer				FIFO_WR_PTR[4:0]					0x04	0x00	R/W
Over Flow Counter				OVF_COUNTER[4:0]					0x05	0x00	R/W
FIFO Read Pointer				FIFO_RD_PTR[4:0]					0x06	0x00	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W

Hình 3-9: Thông tin thanh ghi FIFO

**FIFO Write Pointer**

FIFO Write Pointer chỉ đến vị trí nơi mà MAX30102 viết cho mẫu kế tiếp. Con trỏ này tiến tới mỗi mẫu được đẩy vào FIFO. Nó cũng có thể được thay đổi thông qua giao diện I2C khi MODE[2:0] là 010, 011, hoặc 111.

**FIFO Overflow Counter**

Khi FIFO đầy, mẫu sẽ không được đẩy tới FIFO, mẫu lúc này sẽ bị mất. OVF\_COUNTER sẽ đếm số lượng mẫu bị mất. Nó bão hoà ở giá trị 0xF. Khi một mẫu hoàn chỉnh được “popped” (tức là, loại bỏ dữ liệu FIFO cũ và dịch các mẫu xuống) từ FIFO (khi con trỏ đọc tiến tới), OVF\_COUNTER sẽ được khởi tạo lại giá trị 0.

**FIFO Read Pointer**

Con trỏ đọc FIFO chỉ đến vị trí mà bộ xử lý sẽ lấy mẫu tiếp theo từ FIFO thông qua giao diện I2C. Con trỏ này tiến tới mỗi khi một mẫu được bật ra từ FIFO. Bộ xử lý cũng có thể ghi vào con trỏ này sau khi đọc các mẫu để cho phép đọc lại các mẫu từ FIFO nếu có lỗi trong quá trình truyền dữ liệu.

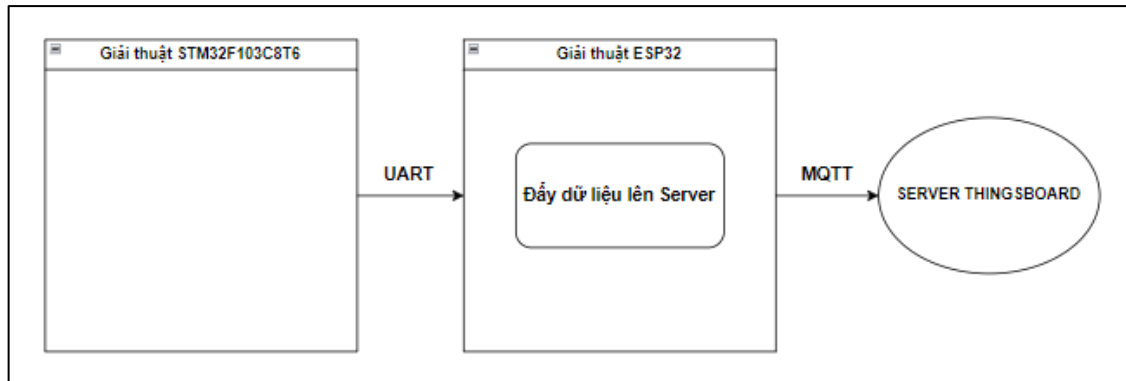
**FIFO Data Register**

FIFO có thể chứa tới 32 mẫu dữ liệu. Kích thước mẫu phụ thuộc vào số lượng kênh LED (còn gọi là kênh) được cấu hình hoạt động. Vì mỗi tín hiệu kênh được lưu trữ dưới dạng tín hiệu dữ liệu 3 byte, nên chiều rộng của FIFO có thể là 3 byte hoặc 6 byte.

Thanh ghi FIFO\_DATA trong bảng thanh ghi I2C trỏ đến mẫu tiếp theo sẽ được đọc từ FIFO. FIFO\_RD\_PTR trỏ đến mẫu này. Việc đọc thanh ghi FIFO\_DATA không tự động tăng địa chỉ thanh ghi I2C. Lưu đồ giải thuật



### 3.4.2. Lưu đồ giải thuật chính

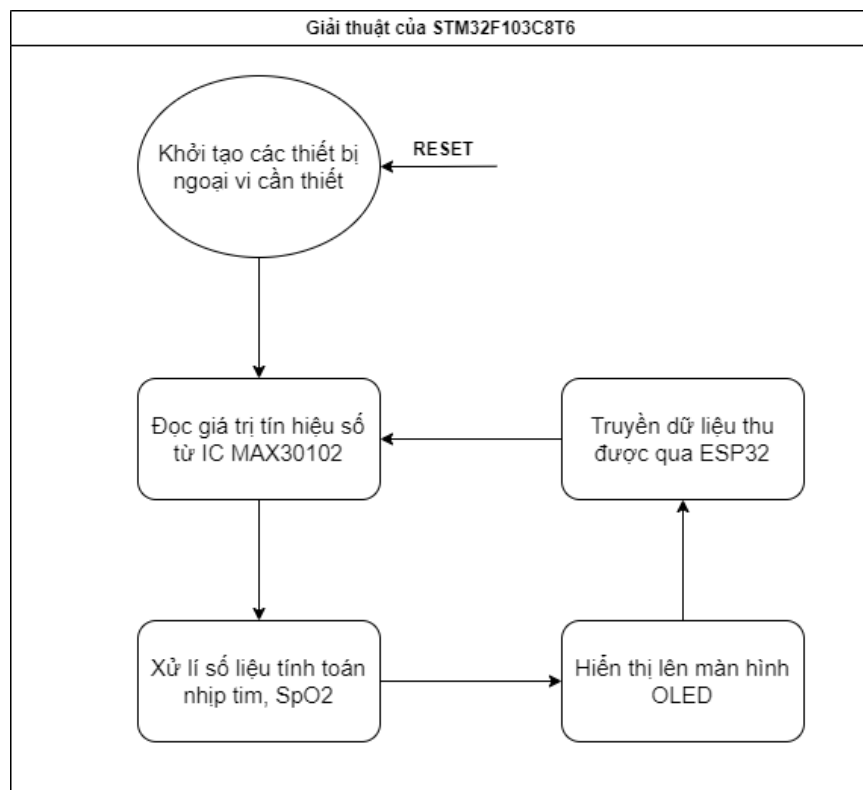


**Hình 3-10: Sơ đồ giải thuật tổng quát**

Hệ thống gồm hai khối giải thuật xử lý chính bao gồm:

- **Khối xử lý của STM32F103C8T6:** Thực hiện xử lý giao tiếp với các thiết bị ngoại vi khác, xử lý số liệu nhận được và hiển thị thông tin của hệ thống lên màn hình OLED
- **Khối xử lý của ESP8266:** Thực hiện nhận dữ liệu từ khối điều khiển STM32F103C8T6 thông qua giao thức UART và truyền lên Server thông qua giao thức MQTT.

### 3.4.3. Lưu đồ giải thuật cho STM32F103C8T6



**Hình 3-11: Lưu đồ giải thuật STM32F103C8T6**

**Bước 1:** Khởi tạo các thiết bị ngoại vi bao gồm: OLED, MAX30102

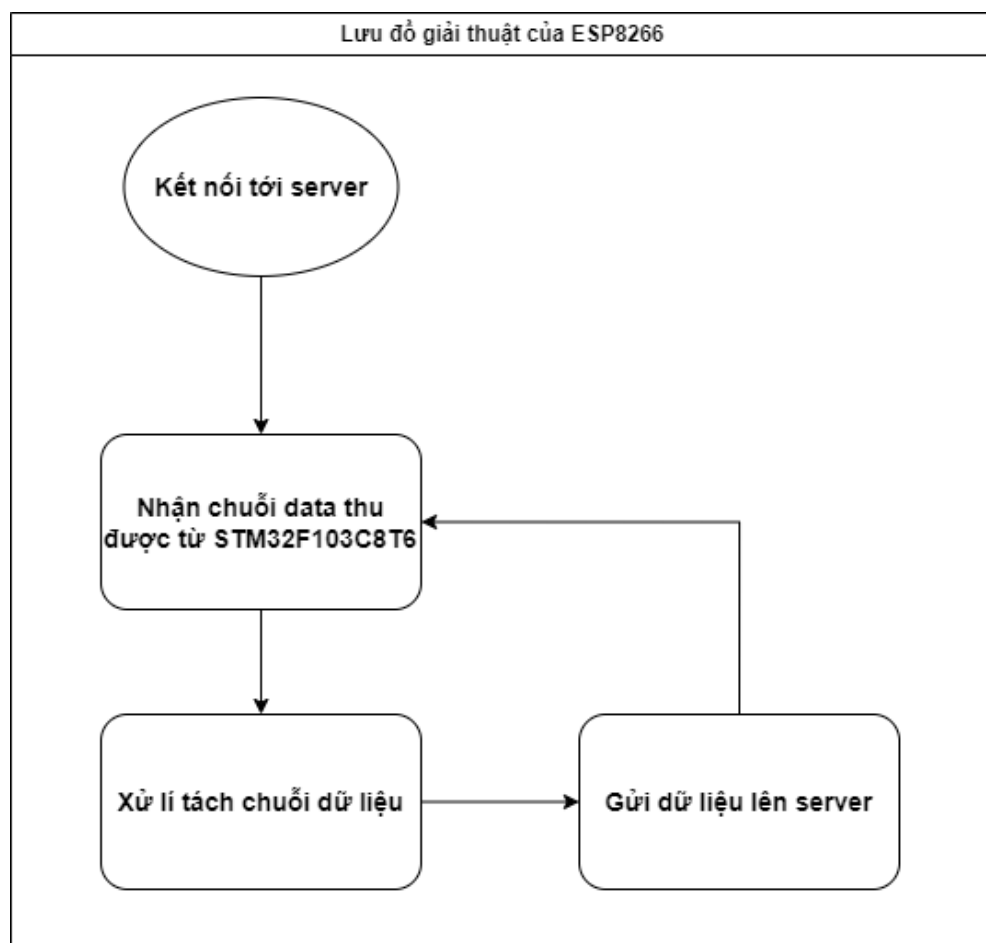
**Bước 2:** Đọc giá trị tín hiệu số của cường độ ánh sáng đỏ (RED) và ánh sáng hồng ngoại (IR) thì được từ IC MAX30102 thông qua giao thức truyền thông I2C.

**Bước 3:** Xử lý dữ liệu nhận được thành giá trị nhịp tim và,  $SpO_2$ .

**Bước 4:** Đưa dữ liệu hiển thị lên màn hình OLED.

**Bước 5:** Truyền dữ liệu qua ESP8266 thông qua giao thức MQTT rồi quay lại bước 2.

#### 3.4.4. Lưu đồ giải thuật cho ESP8266



**Hình 3-12:** Lưu đồ giải thuật ESP8266

**Bước 1:** Kết nối tới wifi và Thingsboard.

**Bước 2:** Nhận chuỗi dữ liệu từ STM32F103C8T6 thông qua giao thức UART.

**Bước 3:** Xử lý tách chuỗi dữ liệu để thu được giá trị nhịp tim và  $SpO_2$ .

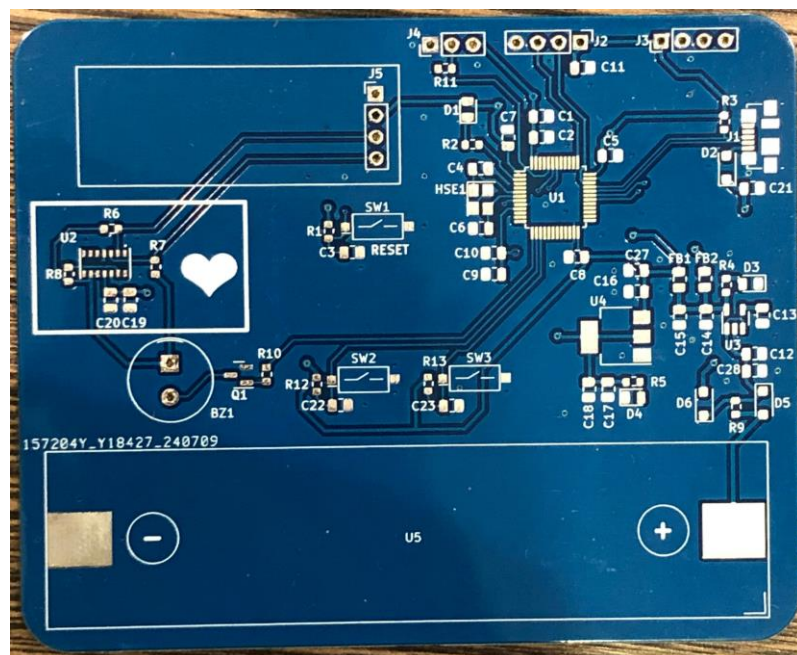
**Bước 4:** Gửi dữ liệu lên server thông qua giao thức MQTT và quay lại bước 2.

## 4. Kết quả thực hiện

Sau khi thực hiện thi công mạch và hàn các linh kiện vào mạch, ta có kết quả mạch thực tế như sau:

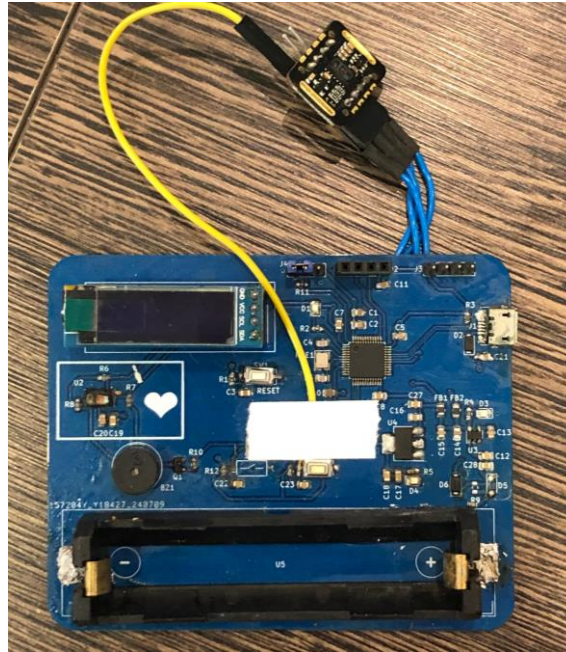
### 4.1. Kết quả phần cứng

Mạch được hàn đầy đủ, các chân linh kiện không bị chồng chéo lên nhau. Mạch hoạt động ổn định, đúng như mục đích thiết kế ban đầu. Được kiểm chứng bằng cách đo điện áp ở chân các linh kiện bằng VOM và thử một số đoạn code lập trình đơn giản để kiểm chứng mạch nạp hoạt động đúng.



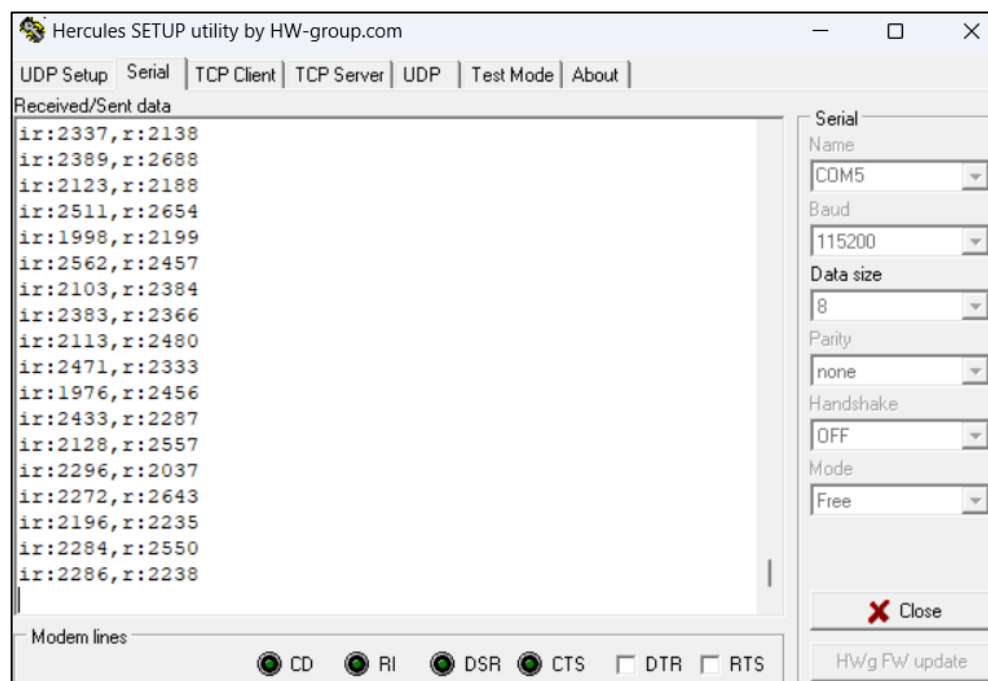
**Hình 4-1: Mạch PCB trước khi hàn linh kiện**

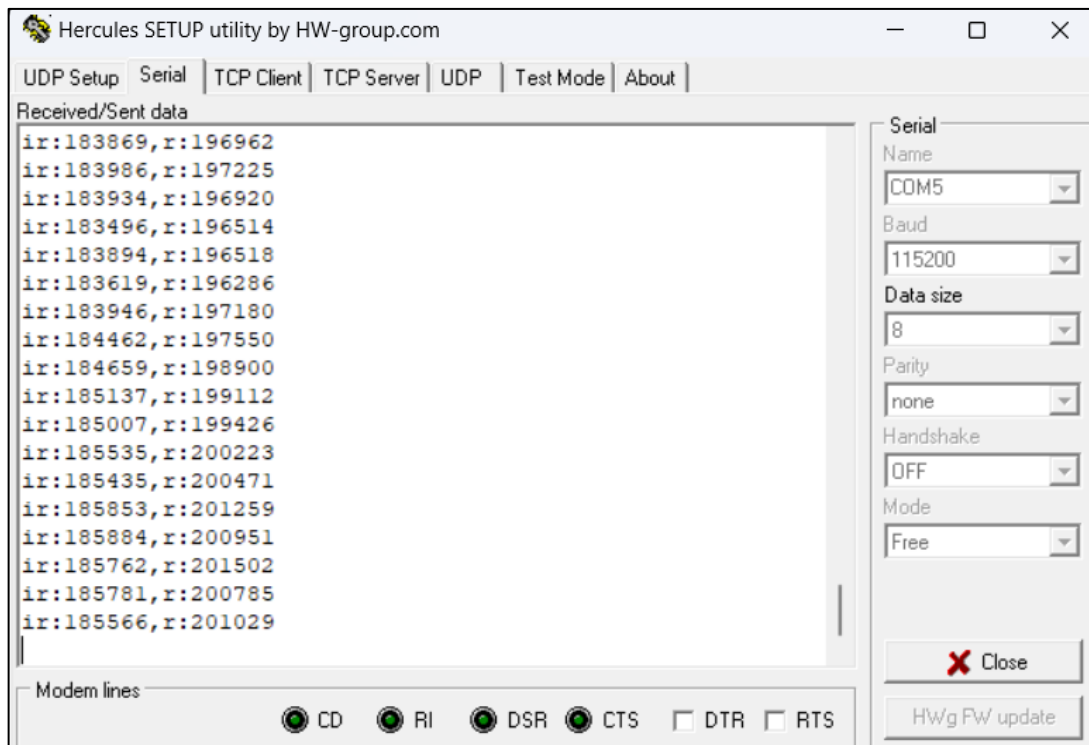
Tuy nhiên, mạch còn vài lỗi và mối hàn chưa được đẹp khi hàn linh kiện như cổng USB, IC MAX30102 do còn hạn chế về dụng cụ và kinh nghiệm khi hàn mạch. Vì vậy, khắc phục bằng cách sử dụng module IC MAX30102.

**Hình 4-2: Mạch sau khi hoàn thiện**

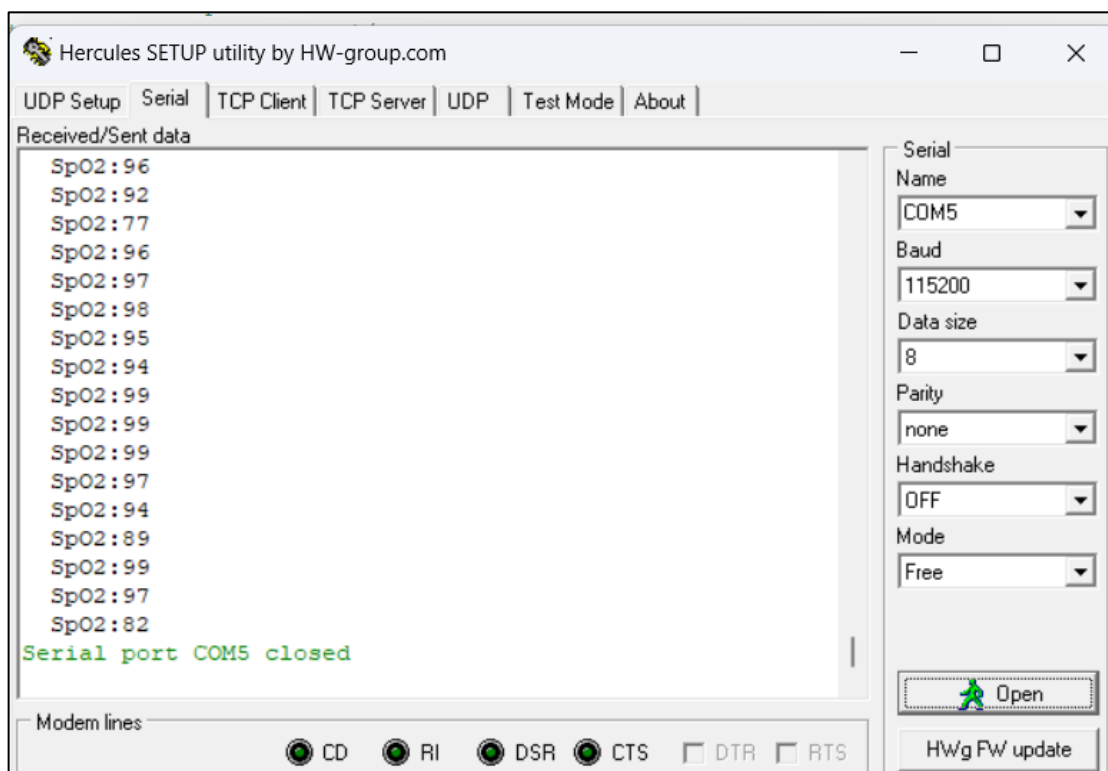
#### 4.2. Kết quả phần mềm

Mạch chạy được đúng chức năng ghi, đọc dữ liệu từ IC MAX30102. Các giá trị đọc về được kiểm chứng bằng cách debug truyền dữ liệu qua giao thức UART tới máy tính cá nhân. Giá trị đọc từ IC được thể hiện như sau:

**Hình 4-3: Giá trị tín hiệu số thu được của R và IR khi chưa đặt tay lên cảm biến**

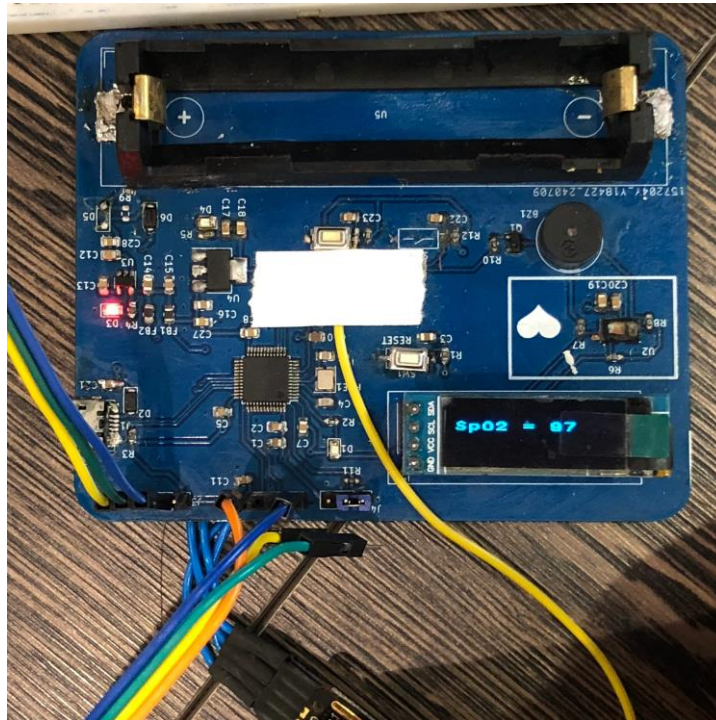


Hình 4-4: Giá trị tín hiệu số thu được khi đặt tay lên cảm biến



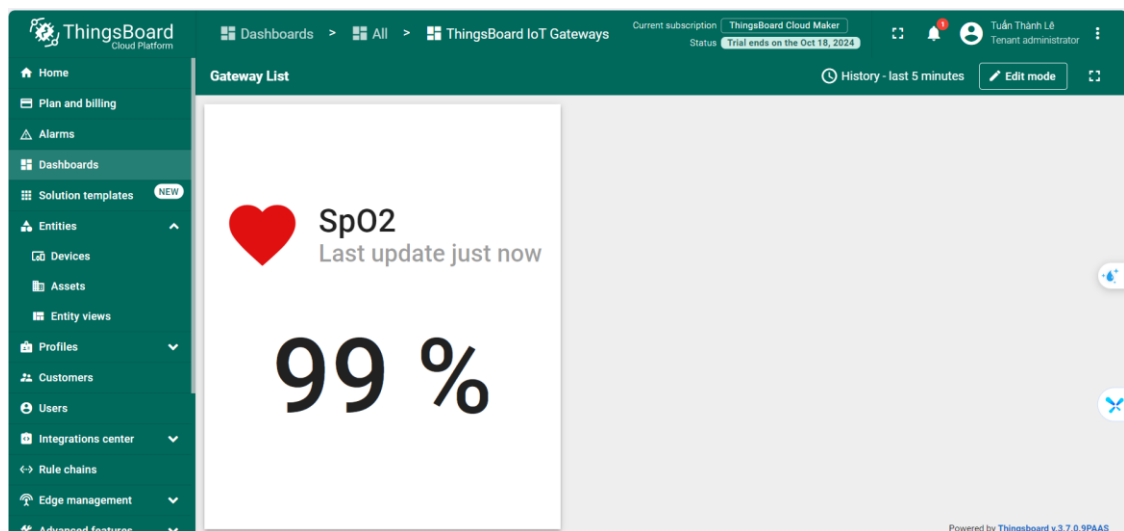
Hình 4-5: Giá trị nồng độ SpO<sub>2</sub> thu được

Giá trị nồng độ  $\text{SpO}_2$  hiển thị được lên trên màn hình OLED



**Hình 4-6: Mạch hiển thị giá trị  $\text{SpO}_2$**

Mạch truyền được qua ESP8266 và tiếp đó lên Server.



**Hình 4-7: Giá trị  $\text{SpO}_2$  hiển thị trên Thingsboard**

Bên cạnh đó, chưa thực hiện được việc xử lý tín hiệu nhận được từ IC MAX30102 để tính ra nhịp tim. Do hạn chế về khả năng xử lý, tính toán số liệu

## CHƯƠNG 3: PHẦN KẾT LUẬN

### 1. Kết luận

Đã thực hiện hoàn thiện thiết kế và thi công mạch PCB. Mạch hoạt động đúng với các đa số lý thuyết thiết kế, điện áp trên các linh kiện ổn định. Về phần mềm, thực hiện được đa phần các chức năng của hệ thống. Đầu tiên, chức năng giao tiếp I2C (cảm biến, OLED) và giao tiếp UART (ESP8266, debug mạch USB – UART) hoạt động tốt. Trong quá trình thực hiện đề tài, đã rút được nhiều kinh nghiệm trong quy trình thiết kế lên một hệ thống nhưng từ việc lên ý tưởng, thi công mạch và lập trình. Bên cạnh đó, cải thiện được khả năng giải quyết những vấn đề gặp phải khi mạch gặp lỗi.

Nhìn chung, hoàn thành được đa phần các nhiệm vụ đặt ra từ đầu. Tuy nhiên vẫn còn nhiều thiếu sót. Đầu tiên là việc thiếu kinh nghiệm trong việc thi công mạch nên mỗi hàn chưa được đẹp và một số linh kiện bị hư hại (IC MAX30102, cổng USB). Từ đó phải sử dụng module thay thế, do chưa cân đối được thời gian nên không kịp để thiết kế và in lại mạch mới. Tiếp đó, mạch vẫn chưa chạy được một số chức năng do hạn chế trong việc xử lý thuật toán tín hiệu.

### 2. Hướng phát triển

Mạch có thể thay thế IC khác để đo điện tim thay vì quang học. Kết quả thu được sẽ chính xác hơn. Bên cạnh đó khi sử dụng tín hiệu điện tim có thể dự đoán thêm một số bệnh ở tim.

Ngoài ra, có thể thêm kết hợp thiết kế web hoặc app mobile để người dùng tiện cho việc theo dõi. Tiếp đó có thể tích hợp thêm các mô hình máy học để dự đoán một số căn bệnh từ điện tim.



## TÀI LIỆU THAM KHẢO

- [1]. D. Kim Thoa, (06/ 04/ 2017), “*Máy đo nhịp tim hoạt động ra sao?*”, Truy cập từ: <https://tuoitre.vn/tu-van-tieu-dung/may-do-nhip-tim-hoat-dong-ra-sao-1292606.htm>.
- [2]. Nguyễn Văn Ngọc, (02/ 06/ 2013), “*Nghiên cứu thiết kế mô hình máy đo nhịp tim và nồng độ oxy trong máu*”, Truy cập từ: <https://1tailieu.com/document/nguyen-cuu-thiet-ke-mo-hinh-may-do-nhip-tim-va-nong-do-oxy-trong-mau-1397933.html>
- [3]. Muhibul Haque Bhuyan and Mashuk Sheikh, (04/ 11/ 2021), “*Design, Implementing, and Testing of a Microcontroller and IoT – based Pulse Oximeter Device*”, Truy cập từ: <https://www.iosrjournals.org/iosr-jeee/Papers/Vol16-Issue5/Ser-2/E1605023848.pdf>
- [4]. How To Electronics, (02/ 04/ 2024), “*MAX30102 & Arduino: Heart Rate + Blood Oxygen Monitoring*”, Truy cập từ: <https://how2electronics.com/max30102-arduino-heart-rate-blood-oxygen-monitoring/>
- [5]. MAXIM [Maxim Integrated Products], “*MAX30102 datasheet*”, Truy cập từ: <https://www.alldatasheet.com/datasheet-pdf/pdf/859400/MAXIM/MAX30102.html>
- [6]. Anh Như (23/04/2024), “*Khám phá Thingsboard là gì? Các tính năng nổi bật của Thingsboard mà người dùng cần biết*”, Truy cập từ: <https://fptshop.com.vn/tin-tuc/danh-gia/kham-pha-thingsboard-la-gi-181213>



## PHỤ LỤC

Chi tiết đồ án được trình bày trong repository GitHub có đường dẫn:

“<https://github.com/tuanthanhhh/project233>” hoặc QR code

