

ĐẠI HỌC QUỐC GIA TP. HCM  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA ĐIỆN TỬ - VIỄN THÔNG

HỌ VÀ TÊN SINH VIÊN

Nguyễn Vũ Nhật Thành

Võ Thành Tiến

TÊN ĐỒ ÁN MÔN HỌC

***ĐO NHỊP TIM VÀ NỒNG ĐỘ OXY TRONG MÁU***

NGƯỜI HƯỚNG DẪN:

1. Lê Trung Khanh

# Lời mở đầu

Cùng với sự phát triển của cuộc sống, mô hình bệnh tật trên thế giới cũng như ở Việt Nam đang dần thay đổi. Tỷ lệ đang mắc và tử vong do các bệnh truyền nhiễm đang thay đổi, khi đó tỷ lệ mắc và tử vong do bệnh mãn tính không lây đang dần nổi lên. Cụ thể như là: Đái tháo đường, tăng huyết áp, xơ vữa động mạch.... Một trong những biến chứng đối với cơ quan chính của các loại bệnh tật trên đứng đầu là tim mạch. Vì vậy việc phát hiện bệnh giai đoạn sớm nhận biết các dấu hiệu nguy hiểm đe dọa tính mạng như: Ngưng tim, rối loạn nhịp tim hay nồng độ oxy trong máu là hết sức quan trọng. Bên cạnh đó, việc theo dõi đáp ứng điều trị, theo dõi khi bác sĩ ra y lệnh đối với một số loại thuốc ảnh hưởng đến tim mạch là rất cần thiết. Từ đó giảm được tỉ lệ tử vong nâng cao chất lượng cuộc sống cho bệnh nhân. Với việc theo dõi nhịp tim, chúng ta có thể theo dõi được những bệnh nêu ở trên. Cùng với đó chúng ta có thể kiểm tra thân nhiệt hằng ngày cho các thành viên trong gia đình cũng góp phần kiểm tra thân nhiệt phát hiện sớm các triệu chứng của bệnh cảm hay sốt đặc biệt là cho trẻ em. Chính vì vậy nhóm chúng em thực hiện đề tài “**THIẾT KẾ THIẾT BỊ ĐO NHỊP TIM VÀ NỒNG ĐỘ OXY TRONG MÁU**”.

Với thiết bị này thì ta có thể đo nhiệt độ, nhịp tim và nồng độ oxy trong máu, theo dõi nhịp chúng mỗi ngày. Như vậy chúng ta có thể tự nhận biết được các dấu hiệu như: Nhiệt độ, nhịp tim hay nồng độ SpO2 đang tăng lên hay giảm xuống, nhịp tim có đều hay không. Đáng nói hơn là với việc đo nhịp tim rất thuận tiện cho bác sĩ theo dõi việc đáp ứng một số loại thuốc. Đồng thời với sự phát triển của công nghệ nên việc tiếp cận với các linh kiện thiết bị cảm biến hay board mạch xử lý của dễ dàng hơn. Không gì hơn với sự trợ giúp ý tưởng từ thầy cô trong khoa Điện tử Viễn Thông thì nhóm dự định thiết kế một thiết bị đo nhiệt độ nhịp tim người sử dụng cảm biến quang học MAX30102 trên board mạch xử lý STM32 giao tiếp với máy tính. Với thiết bị này nó như một lời cảnh báo đến với bệnh nhân thay cho bác sĩ, nó sẽ thông báo cho chúng ta khi nhịp tim chúng ta có dấu hiệu bất thường.

## **I. Mục tiêu:**

Thiết kế được thiết bị đo nhiệt độ, nhịp tim và nồng độ oxy trong máu. Thiết bị này giúp người dùng đo nhiệt độ, nhịp tim và nồng độ oxy trong máu khi cần thiết, cảnh báo cho người dùng cũng như người thân biết được các thông số sức khỏe để kịp thời có những giải pháp hiệu quả và tránh được các sự cố không mong muốn xảy ra. Các thông số sức khỏe sẽ được hiển thị trên giao diện điện thoại đồng thời lưu trữ dưới dạng tệp tài liệu để dễ dàng chia sẻ thông tin khi cần sự tư vấn của bác sĩ.

## **II. Các linh kiện chính:**

### **1.VI ĐIỀU KHIỂN STM32F103C8T6**

Tính năng sản phẩm:

- STM32F103C8T6 là một bộ vi điều khiển 32 bit dựa trên hạt nhân Cortex-M3 , thuộc dòng STM 32 trong bộ vi điều khiển
- Vi điều khiển có dòng hiệu suất mật độ trung bình kết hợp lõi RISC hiệu suất cao với tần số hoạt động 72MHz
- Bo mạch dựa trên mạch MCU cơ bản nhất, mạch tinh thể 8M và 32768, mạch cấp nguồn USB.
- Bảng lõi được chia thành hai hàng dẫn đến tất cả các cổng I / O.
- Tất cả các thiết bị đều cung cấp hai ADC 12 bit, ba bộ định thời 16 bit đa năng cộng với một bộ hẹn giờ PWM, cũng như các giao diện giao tiếp tiêu chuẩn và nâng cao: lên đến hai I2C và SPI, ba USART, một USB và CAN.
- Với việc sử dụng giao diện Micro USB, bạn có thể thực hiện giao tiếp USB và cấp nguồn, giao diện USB, tương thích với giao diện sạc điện thoại di động Andrews thông thường.

Ứng dụng:

- Dùng cho driver để điều khiển ứng dụng, điều khiển ứng dụng thông thường, thiết bị cầm tay và thuốc, máy tính và thiết bị ngoại vi
- Ứng dụng trong công nghiệp, thiết bị lập trình PLC, biến tần, máy in, máy quét, hệ thống cảnh báo, thiết bị liên lạc nội bộ...

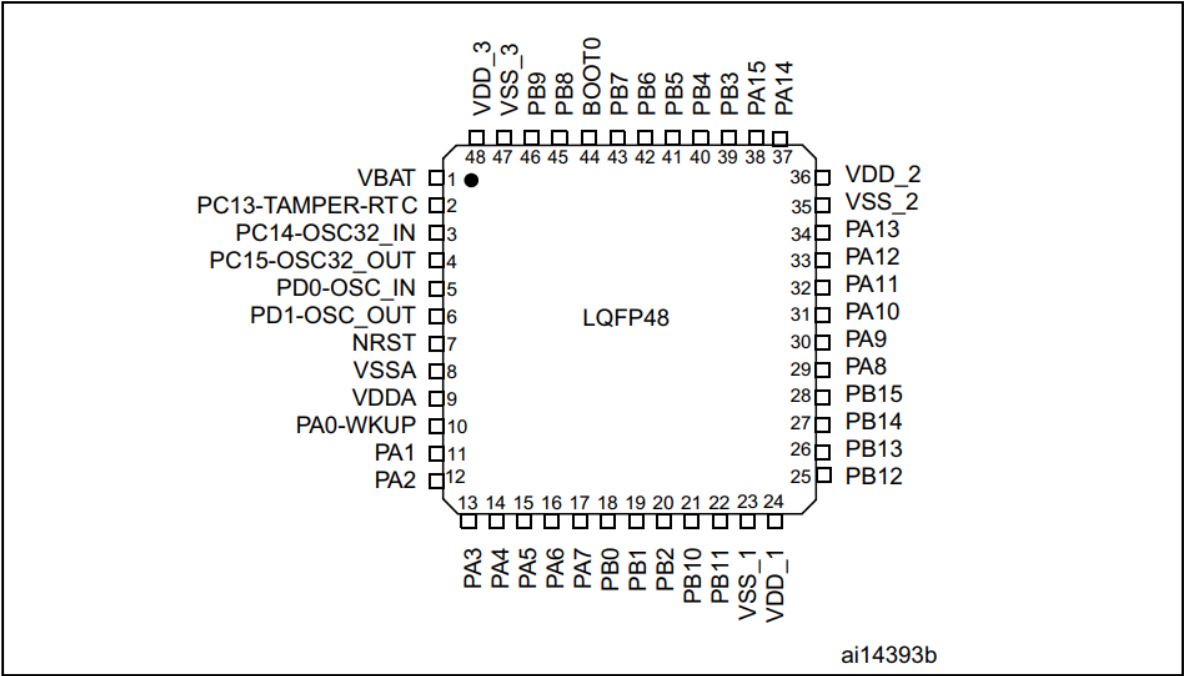
### Bảng thông số kỹ thuật:

Manufacturer	STMicroelectronics
Product Category	ARM Microcontrollers - MCU
Series	STM32F103C8
Mounting Style	SMD/SMT
Package / Case	LQFP-48
Core	ARM Cortex M3
Program Memory Size	64 kB
Data Bus Width	32bit
ADC Resolution	12bit
Maximum Clock Frequency	72Mhz
Number of I/Os	37
Data RAM Size	20Kb
Allowable operating temperature range	- 40 C -> + 85 C
Supply Voltage - Min	2V
Supply Voltage - Max	3.6V
Data RAM Type	SRAM
Interface Type	CAN, I2C, SPI, USART, USB

Moisture Sensitive	Yes
Processor Series	Processor Series
Product Type	ARM Microcontrollers - MCU
Program Memory Type	Flash
Subcategory	Microcontrollers - MCU
Tradename	STM32
Unit Weight	181.700 mg

Sơ đồ chân:

Figure 8. STM32F103xx performance line LQFP48 pinout



Hình 1: Các chân pinout của STM32F103XX

Bảng chân :

Pins	Pin name	Type	Level	Main function	Alternate funtions	
LQFP48					Default	Remap
1	V <sub>BAT</sub>	S	-	V <sub>BAT</sub>	-	-
2	PC13-TAMPER-RTC <sup>(5)</sup>	I/O	-	PC13 <sup>(6)</sup>	TAMPER-RTC	-
3	PC14-OSC32_IN <sup>(5)</sup>	I/O	-	PC14 <sup>(6)</sup>	OSC32_IN	-
4	PC15-OSC32_OUT <sup>(5)</sup>	I/O	-	PC15 <sup>(6)</sup>	OSC32_OUT	-
5	OSC_IN	I	-	OSC_IN	-	PD0 <sup>(7)</sup>
6	OSC_OUT	O	-	OSC_OUT		PD1 <sup>(7)</sup>
7	NRST	I/O	-	NRST	-	-
8	V <sub>SSA</sub>	S	-	V <sub>SSA</sub>	-	-
9	V <sub>DDA</sub>	S	-	V <sub>DDA</sub>	-	-
10	PA0-WKUP	I/O	-	PA0	WKUP/ USART2_CTS <sup>(9)</sup> / ADC12_IN0/ TIM2_CH1_ ETR <sup>(9)</sup>	-
11	PA1	I/O	-	PA1	USART2_RTS <sup>(9)</sup> / ADC12_IN1/ TIM2_CH2 <sup>(9)</sup>	-
12	PA2	I/O	-	PA2	USART2_TX <sup>(9)</sup> / ADC12_IN2/ TIM2_CH3 <sup>(9)</sup>	-
13	PA3	I/O	-	PA3	USART2_RX <sup>(9)</sup> / ADC12_IN3/ TIM2_CH4 <sup>(9)</sup>	-
14	PA4	I/O	-	PA4	SPI1_NSS <sup>(9)</sup> / USART2_CK <sup>(9)</sup> / ADC12_IN4	-
15	PA5	I/O	-	PA5	SPI1_SCK <sup>(9)</sup> / ADC12_IN5	-
16	PA6	I/O	-	PA6	SPI1_MISO <sup>(9)</sup> / ADC12_IN6/ TIM3_CH1 <sup>(9)</sup>	TIM1_BKIN
17	PA7	I/O	-	PA7	SPI1_MOSI <sup>(9)</sup> / ADC12_IN7/ TIM3_CH2 <sup>(9)</sup>	TIM1_CH1N

18	PB0	I/O	-	PB0	ADC12_IN8/ TIM3_CH3 <sup>(9)</sup>	TIM1_CH2N
19	PB1	I/O	-	PB1	ADC12_IN9/ TIM3_CH4 <sup>(9)</sup>	TIM1_CH3N
20	PB2	I/O	FT	PB2/BOOT1	-	-
21	PB10	I/O	FT	PB10	I2C2_SCL/ USART3_TX <sup>(9)</sup>	TIM2_CH3
22	PB11	I/O	FT	PB11	I2C2_SDA/ USART3_RX <sup>(9)</sup>	TIM2_CH4
23	VSS_1	S	-	VSS_1	-	-
24	VDD_1	S	-	VDD_1	-	-
25	PB12	I/O	FT	PB12	SPI2_NSS/ I2C2_SMBAL/ USART3_CK <sup>(9)</sup> / TIM1_BKIN <sup>(9)</sup>	-
26	PB13	I/O	FT	PB13	SPI2_SCK/ USART3_CTS <sup>(9)</sup> / TIM1_CH1N <sup>(9)</sup>	-
27	PB14	I/O	FT	PB14	SPI2_MISO/ USART3_RTS <sup>(9)</sup> / TIM1_CH2N <sup>(9)</sup>	-
28	PB15	I/O	FT	PB15	SPI2_MOSI/ TIM1_CH3N <sup>(9)</sup>	-
29	PA8	I/O	FT	PA8	USART1_CK/ TIM1_CH1 <sup>(9)</sup> / MCO	-
30	PA9	I/O	FT	PA9	USART1_TX <sup>(9)</sup> / TIM1_CH2 <sup>(9)</sup>	-
31	PA10	I/O	FT	PA10	USART1_RX <sup>(9)</sup> / TIM1_CH3 <sup>(9)</sup>	-
32	PA11	I/O	FT	PA11	USART1_CTS/ CANRX <sup>(9)</sup> / USBDM/ TIM1_CH4 <sup>(9)</sup>	-
33	PA12	I/O	FT	PA12	USART1_RTS/ CANTX <sup>(9)</sup> / USBDP TIM1_ETR <sup>(9)</sup>	-
34	PA13	I/O	FT	JTMS/SWDIO	-	PA13
35	VSS_2	S	-	VSS_2	-	-
36	VDD_2	S	-	VDD_2	-	-
37	PA14	I/O	FT	JTCK/SWCLK	-	PA14
38	PA15	I/O	FT	JTDI	-	TIM2_CH1_ ETR/ PA15 /SPI1_NSS
39	PB3	I/O	FT	JTDO	-	TIM2_CH2 / PB3 TRACESWO SPI1_SCK

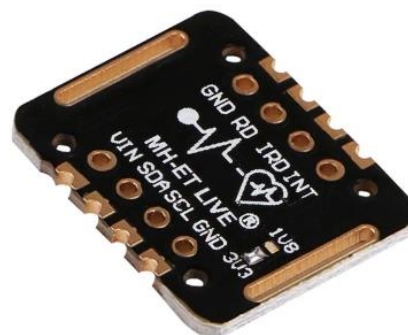
40	PB4	I/O	FT	JNTRST	-	TIM3_CH1/ PB4/ SPI1_MISO
41	PB5	I/O		PB5	I2C1_SMBAL	TIM3_CH2/ SPI1_MOSI
42	PB6	I/O	FT	PB6	I2C1_SCL <sup>(9)</sup> / TIM4_CH1 <sup>(9)</sup>	USART1_TX
43	PB7	I/O	FT	PB7	I2C1_SDA <sup>(9)</sup> / TIM4_CH2 <sup>(9)</sup>	USART1_RX
44	BOOT0	I		BOOT0	-	-
45	PB8	I/O	FT	PB8	TIM4_CH3 <sup>(9)</sup>	I2C1_SCL/ CANRX
46	PB9	I/O	FT	PB9	TIM4_CH4 <sup>(9)</sup>	I2C1_SDA/ CANTX
47	VSS_3	S	-	VSS_3	-	-
48	VDD_3	S	-	VDD_3	-	-

Link tham khảo thêm : <https://www.mouser.vn/datasheet/2/389/cd00161566-1796535.pdf>

## 2. Cảm biến: MAX30102

Thông số kỹ thuật:

- IC chính: MAX30102.
- Đo được nhịp tim và nồng độ Oxy trong máu.
- Điện áp sử dụng: 3.3~5VDC.
- Nhỏ gọn, siêu tiết kiệm năng lượng, thích hợp cho các thiết bị đo nhỏ gọn, Wearable Devices.
- Giao tiếp: I2C, mức tín hiệu TTL.
- Kích thước: 20.6 x 15.5mm



Hình 2: Cảm biến MAX30102

## 3. Màn hình lcd tft 1.8 inch st7789



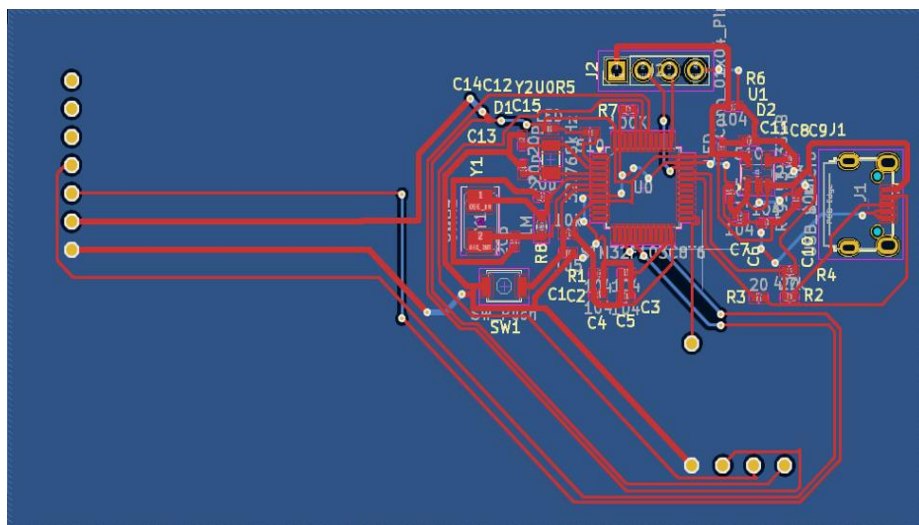
Thông số kỹ thuật:

- Điện áp sử dụng: 3V3 đến 5V (DC)
- Công suất tiêu thụ: 0.04W
- Góc hiển thị: Lớn hơn 160 độ
- Độ phân giải: 128X64 pixel (Điểm ảnh)
- Độ rộng màn hình: 0.96inch
- Giao tiếp: I2C
- Màu: Trắng và Đen
- Driver: SSD1306

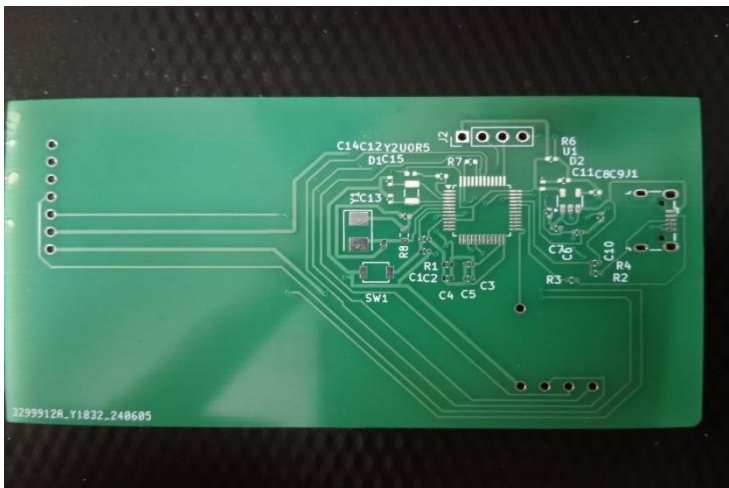


Hình 3: Màn hình Oled

### III. Thiết kế PCB:

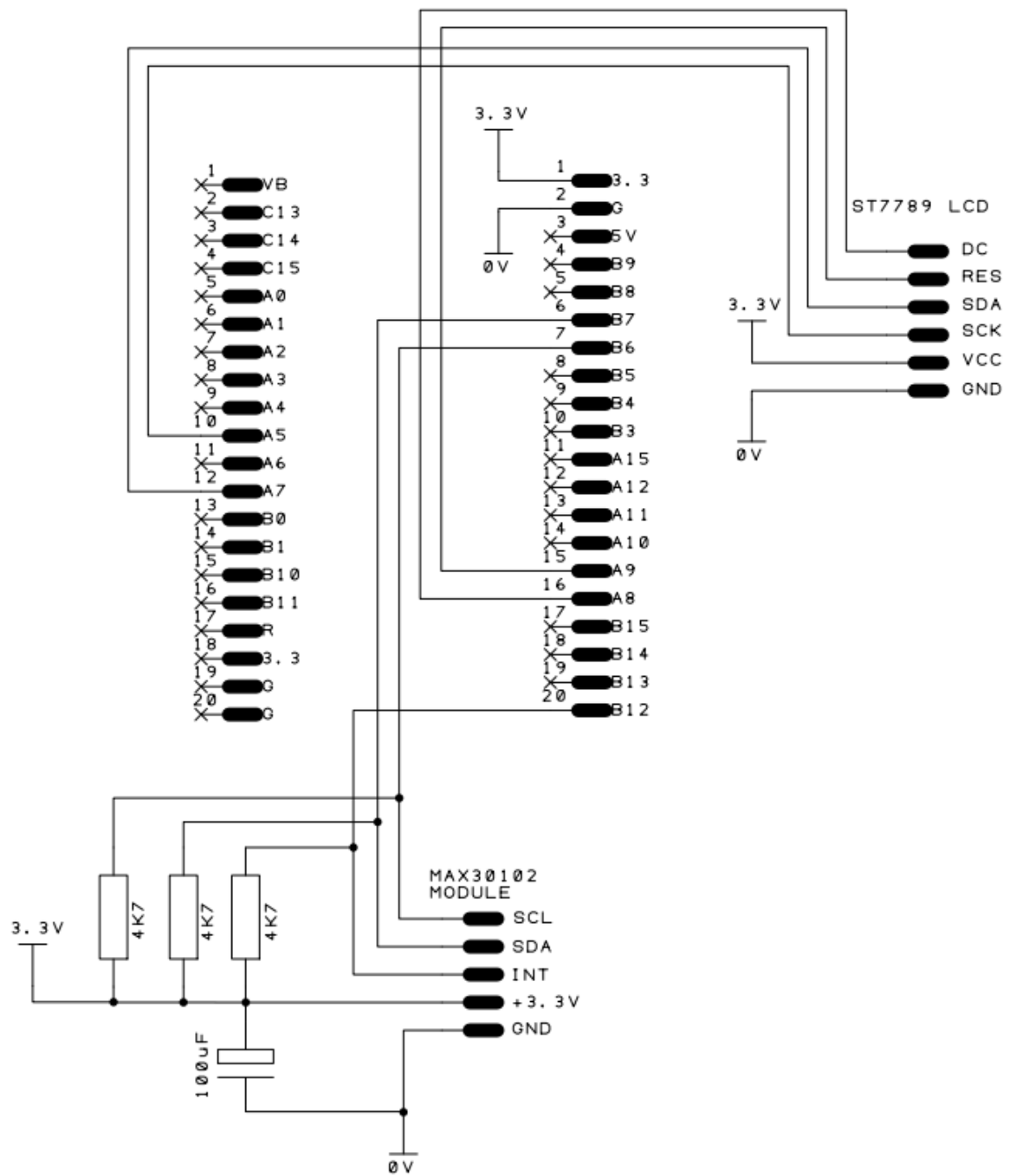


Hình 4: Mạch PCB mô phỏng trên Kicad

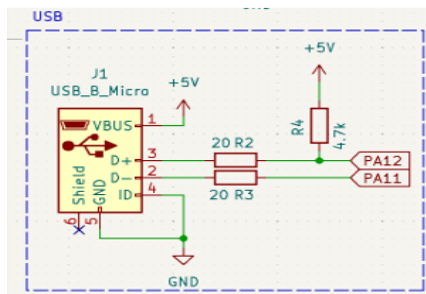


Hình 5: Mạch PCB thực tế

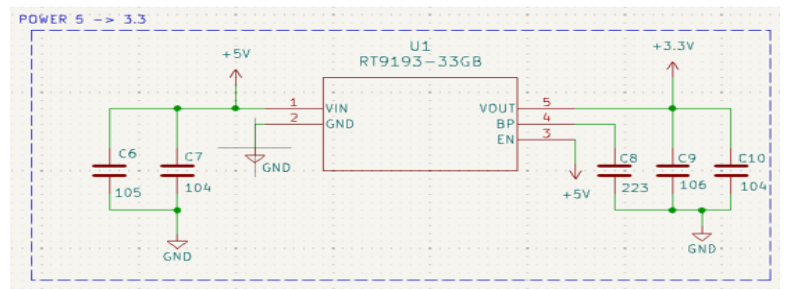
➤ Sơ đồ mạch :



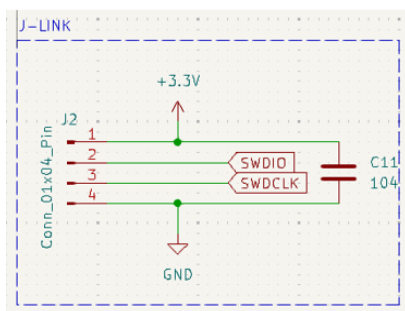
Hình 6: Mạch nối chân



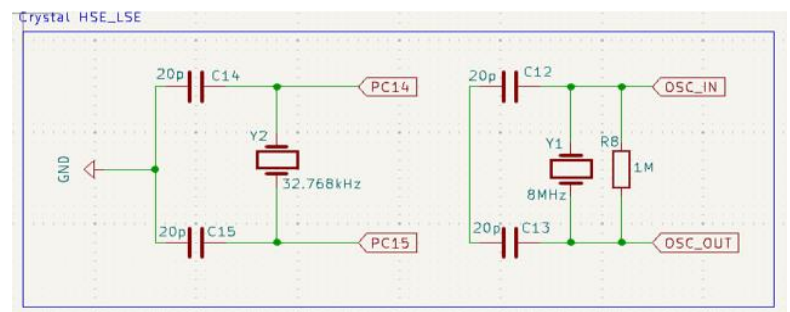
Hình 7: USB



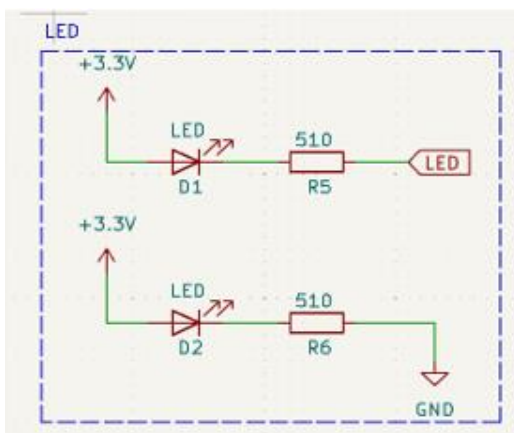
Hình 8: POWER



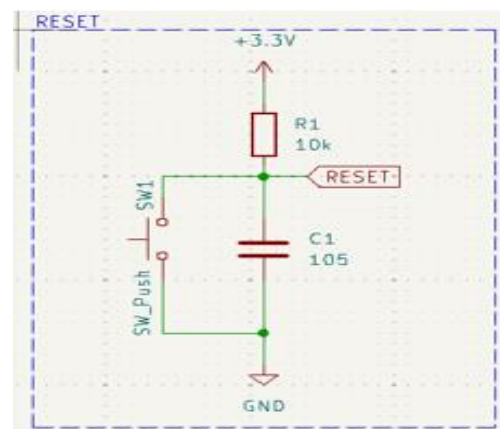
Hình 9: J-Link



Hình 10: Crystal HSE\_LSE



Hình 11: LED



Hình 12: RESET

➤ **Code :**

```

/* USER CODE BEGIN Header */
/**
 * @file      : main.c
 * @brief     : Main program body
 *
 * @attention
 * Bản quyền (c) 2023 STMicroelectronics.
 * Bảo lưu mọi quyền.
 * Phần mềm này được cấp phép theo các điều khoản có thể được tìm thấy trong tệp LICENSE
 * trong thư mục gốc của thành phần phần mềm này.
 * Nếu không có tệp LICENSE đi kèm với phần mềm này, nó được cung cấp AS-IS.
 *
 */
/* USER CODE END Header */

#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include <stdbool.h>
#include <stdint.h>
#include <string.h>
#include "MAX30102.h"
#include "printf.h"
#include "algorithm_by_RF.h"
#include "graphics.h"
/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

```

```

#define MAX_HEART_BEAT_TRACE    140.0f
#define MY_SNPRINTF              snprintf // Thư viện newlib snprintf hỗ trợ số thực quá
nặng và không an toàn với đa luồng, vì vậy sử dụng snprintf từ tệp printf.c cục bộ
// Gây ra cảnh báo biên dịch rằng hỗ trợ số thực bị tắt, sử dụng
macro này để ngăn chặn nó
#define FAKE_MAX30102_CLONE      // Cảm biến MAX30102 nhái của China
/* USER CODE END PM */

/* Private variables -----*/
I2C_HandleTypeDef hi2c1;

SPI_HandleTypeDef hspi1;
DMA_HandleTypeDef hdma_spi1_tx;

/* USER CODE BEGIN PV */
static uint32_t aun_ir_buffer[BUFFER_SIZE]; // dữ liệu cảm biến đèn LED hồng ngoại (infrared
lamp)
static uint32_t aun_red_buffer[BUFFER_SIZE]; // dữ liệu cảm biến đèn LED ?
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_I2C1_Init(void);
static void MX_SPI1_Init(void);
/* USER CODE BEGIN PFP */
static void Max30102Setup(void);
static void Max30102Loop(void);
/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */

void Max30102Setup()
{
    uint8_t uch_dummy;    //uint8_t: Đây là kiểu dữ liệu 8-bit không dấu (unsigned), có giá trị
từ 0 đến 255.

    maxim_max30102_reset(); // reset cam bien
    maxim_max30102_read_reg(REG_INTR_STATUS_1, &uch_dummy); // được xóa trạng thái
ngắt
    maxim_max30102_init(); // khởi tạo cảm biến
}

// Lấy mẫu từ MAX30102. Tần số nhịp tim và SpO2 được tính mỗi giây ST
void Max30102Loop()
{
    char buf[20];          // Mảng đệm để chứa chuỗi kết quả.
    float n_spo2;          // Biến để lưu trữ giá trị SpO2.

```

```

float ratio; // Biến để lưu trữ tỷ lệ.
float correl; // Biến để lưu trữ hệ số tương quan.
int8_t ch_spo2_valid; // chỉ báo để hiển thị tính hợp lệ của phép tính SPO2
int32_t n_heart_rate; // giá trị nhịp tim
int8_t ch_hr_valid; // chỉ báo để hiển thị tính hợp lệ của phép tính nhịp tim
uint8_t i; // Biến đếm cho vòng lặp.
static uint32_t un_min = 0UL; // Biến tĩnh lưu trữ giá trị tối thiểu của dữ liệu (giá trị số
// nguyên không dấu 262143)
static uint32_t un_max = 0x3FFFFUL; // Biến tĩnh lưu trữ giá trị tối đa của dữ liệu
static uint32_t un_prev_data = 0UL; // các biến để tính toán độ sáng của đèn LED trên bo
// mạch phản ánh nhịp tim
static float f_heartbeatTrace = 0UL; // Biến tĩnh lưu trữ dấu vết nhịp tim.
float f_temp; // Biến tạm thời để tính toán

static int16_t x = 1; // Biến tĩnh lưu trữ vị trí đầu tiên của đường trạng thái nhịp tim.
static int16_t lastY = 35; // Biến tĩnh lưu trữ vị trí đầu tiên của trạng thái nhịp tim cuối cùng.

// độ dài bộ đệm của BUFFER_SIZE lưu trữ các mẫu trong giây ST chạy ở sps FS - tốc độ lấy
mẫu (sampling rate)
for (i = 0U; i < BUFFER_SIZE; i++)
{
    while (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12) == GPIO_PIN_SET); // cho đến khi
    chân ngắt được kích hoạt

#ifdef FAKE_MAX30102_CLONE
    maxim_max30102_read_fifo((aun_ir_buffer + i), (aun_red_buffer + i)); // từ FIFO của
    MAX30102
#else
    maxim_max30102_read_fifo((aun_red_buffer + i), (aun_ir_buffer + i)); // từ FIFO của
    MAX30102
#endif
    // tính toán độ sáng của đèn LED
    if (aun_red_buffer[i] > un_prev_data)
    {
        f_temp = aun_red_buffer[i] - un_prev_data;
        f_temp /= (un_max - un_min); // chuẩn hóa hoặc tỷ lệ dữ liệu.
        f_temp *= MAX_HEART_BEAT_TRACE; // điều chỉnh hoặc thay đổi biên độ của
        dữ liệu.
        f_heartbeatTrace -= f_temp;
        if (f_heartbeatTrace < 5.0f)
        {
            f_heartbeatTrace = 5.0f;
        }
    }
    else
    {
        f_temp = un_prev_data - aun_red_buffer[i];
        f_temp /= (un_max - un_min);
        f_temp *= MAX_HEART_BEAT_TRACE;
        f_heartbeatTrace += f_temp;
    }
}

```

```

    if (f_heartbeatTrace > MAX_HEART_BEAT_TRACE + 25.0f)
    {
        f_heartbeatTrace = MAX_HEART_BEAT_TRACE + 25.0f;
    }
}

GraphicsLine(x, 210 - lastY, x + 3, 210 - ((int16_t)f_heartbeatTrace), RED);
lastY = (int16_t)f_heartbeatTrace;
x += 3;
if (x == 238)
{
    GraphicsFilledRectangle(1, 1, 238, 208, BLACK);
    x = 1;
}
un_prev_data = aun_red_buffer[i];
}

un_min = 0UL;
un_max = 0x3FFFFUL;
for (i = 0U; i < BUFFER_SIZE; i++)
{
    if (un_min > aun_red_buffer[i])
    {
        un_min = aun_red_buffer[i]; // cập nhật tín hiệu tối thiểu cho chu kỳ tiếp theo sử dụng các
giá trị từ chu kỳ này
    }
    if (un_max < aun_red_buffer[i])
    {
        un_max = aun_red_buffer[i]; // cập nhật tín hiệu tối đa cho chu kỳ tiếp theo sử dụng các
giá trị từ chu kỳ này
    }
}

// tính toán nhịp tim và SpO2 sau các mẫu BUFFER_SIZE (các mẫu giây ST) bằng phương
pháp của Robert
rf_heart_rate_and_oxygen_saturation(aun_ir_buffer, BUFFER_SIZE, aun_red_buffer,
&n_spo2, &ch_spo2_valid, &n_heart_rate, &ch_hr_valid, &ratio, &correl);

// hiển thị kết quả
GraphicsFilledRectangle(10, 212, 229, 28, BLACK);
if (ch_hr_valid && ch_spo2_valid)
{
    MY_SNPRINTF(buf, (size_t)20, "SpO2 %3.1f%%", n_spo2);
    GraphicsLargeString(10, 212, buf, GREEN);
    snprintf(buf, (size_t)20, "Nhịp tim %d b/m", n_heart_rate);
    GraphicsLargeString(10, 227, buf, GREEN);
}
else
{
    GraphicsLargeString(10, 220, "Không hợp lệ", RED);
}

```

```

    }
}
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_I2C1_Init();
    MX_SPI1_Init();
    /* USER CODE BEGIN 2 */
    Max30102Setup();
    SPI_1LINE_TX(&hspi1);
    GraphicsInit();
    GraphicsClear(BLACK);
    GraphicsRectangle(0, 0, 240, 210, BLUE);
    GraphicsLargeString(10, 220, "Sampling", BLUE);
    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */

```



```

    /* USER CODE BEGIN 3 */
    Max30102Loop();
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Initializes the RCC Oscillators according to the specified parameters
     * in the RCC_OscInitTypeDef structure.
     */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    /** Initializes the CPU, AHB and APB buses clocks
     */
    RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK
[RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
    {
        Error_Handler();
    }
}

/**
 * @brief I2C1 Initialization Function
 * @param None
 * @retval None

```

```

*/
static void MX_I2C1_Init(void)
{

/* USER CODE BEGIN I2C1_Init 0 */

/* USER CODE END I2C1_Init 0 */

/* USER CODE BEGIN I2C1_Init 1 */

/* USER CODE END I2C1_Init 1 */
hi2c1.Instance = I2C1;
hi2c1.Init.ClockSpeed = 400000;
hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
hi2c1.Init.OwnAddress1 = 0;
hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
hi2c1.Init.OwnAddress2 = 0;
hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
if (HAL_I2C_Init(&hi2c1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN I2C1_Init 2 */

/* USER CODE END I2C1_Init 2 */

}

/**
 * @brief SPI1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_SPI1_Init(void)
{

/* USER CODE BEGIN SPI1_Init 0 */

/* USER CODE END SPI1_Init 0 */

/* USER CODE BEGIN SPI1_Init 1 */

/* USER CODE END SPI1_Init 1 */
/* SPI1 parameter configuration*/
hspi1.Instance = SPI1;
hspi1.Init.Mode = SPI_MODE_MASTER;
hspi1.Init.Direction = SPI_DIRECTION_1LINE;
hspi1.Init.DataSize = SPI_DATASIZE_8BIT;

```

```

hspi1.Init.CLKPolarity = SPI_POLARITY_HIGH;
hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
hspi1.Init.NSS = SPI_NSS_SOFT;
hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_4;
hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
hspi1.Init.CRCPolynomial = 10;
if (HAL_SPI_Init(&hspi1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN SPI1_Init 2 */

/* USER CODE END SPI1_Init 2 */

}

/**
 * Enable DMA controller clock
 */
static void MX_DMA_Init(void)
{

    /* DMA controller clock enable */
    __HAL_RCC_DMA1_CLK_ENABLE();

    /* DMA interrupt init */
    /* DMA1_Channel3_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Channel3_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel3_IRQn);

}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    /* USER CODE BEGIN MX_GPIO_Init_1 */
    /* USER CODE END MX_GPIO_Init_1 */

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

```

```

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8|GPIO_PIN_9, GPIO_PIN_RESET);

/*Configure GPIO pin : PB12 */
GPIO_InitStruct.Pin = GPIO_PIN_12;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pins : PA8 PA9 */
GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/* USER CODE BEGIN MX_GPIO_Init_2 */
/* USER CODE END MX_GPIO_Init_2 */
}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* Thêm mã xử lý lỗi của bạn vào đây */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 *        where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{

```

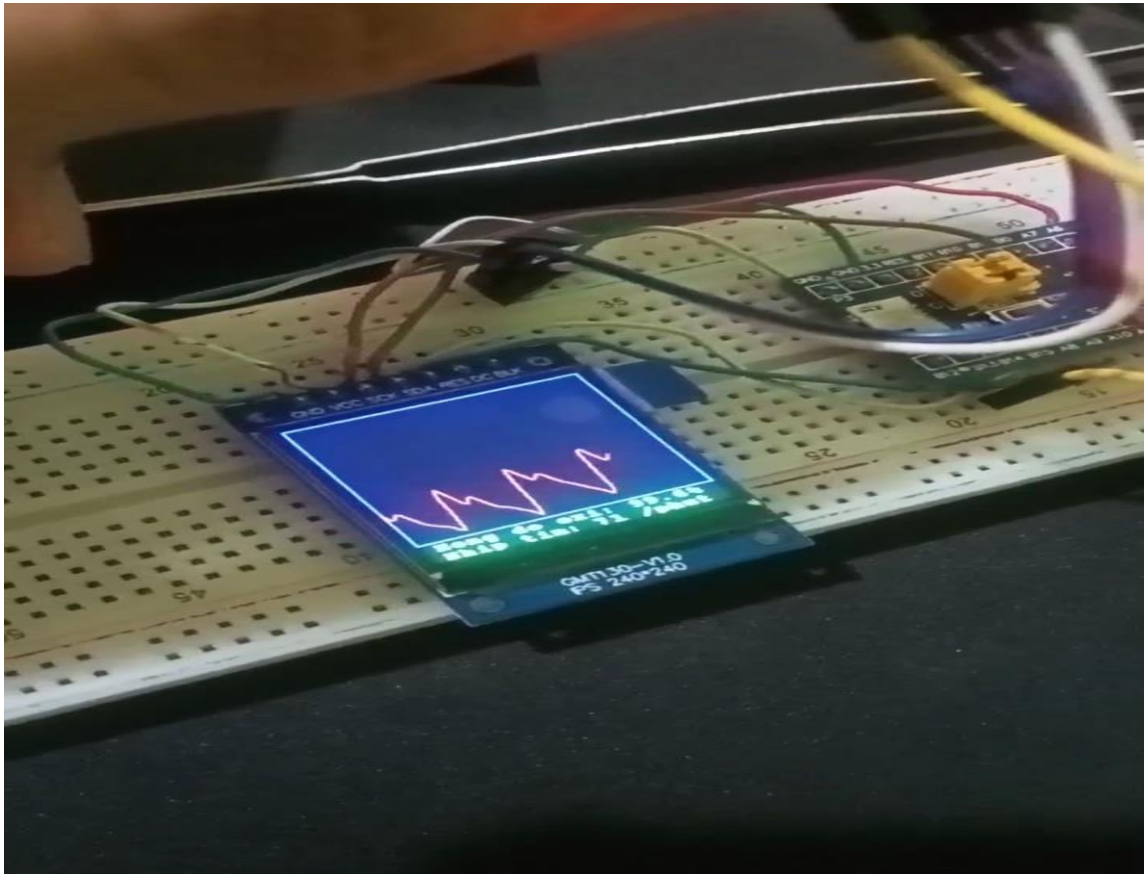
```

/* USER CODE BEGIN 6 */
/* Thêm mã xử lý lỗi của bạn vào đây */
/* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```

#### IV. Sản Phẩm:

- Demo sản phẩm



- Thực hành đo:

STT	Họ và tên	Đo 1	Đo 2	Đo 3	Đánh giá
1	Nguyễn Vũ Nhật Thành	65	67	74	Ổn định
2	Võ Thành Tiến	66	68	120	Chưa ổn định

#### V.Đánh giá:

- **Đánh giá sản phẩm:** Sản phẩm đồ án thiết bị đo nhịp tim và nồng độ oxy trong máu đã hoàn thành với những thành tựu đáng kể. Thiết bị này có thể đo lường nhịp tim của người dùng một cách chính xác và đồng thời đo lường nồng độ oxy trong máu, giúp xác định mức oxy hóa của máu.

*Một số điểm mạnh của sản phẩm:*

1. Chính xác và đáng tin cậy: Thiết bị được thiết kế để đo lường chính xác và đáng tin cậy, giúp người dùng theo dõi sức khỏe.
2. Ứng dụng trong nhiều lĩnh vực: Sản phẩm có thể được áp dụng trong nhiều lĩnh vực y học, thể thao và quản lý sức khỏe cá nhân.

*Một số điểm cần cải thiện:*

1. Hiệu suất pin và thời gian sử dụng: Cần tối ưu hóa hiệu suất pin để có thể sử dụng lâu hơn.
2. Thiết kế còn thô sơ cần được cải tiến nhỏ gọn tiện lợi hơn
3. Tương thích với các thiết bị khác: Phát triển thiết bị tương thích với nhiều thiết bị và trên nhiều nền tảng hơn,...

Đồ án thiết bị đo nhịp tim và nồng độ oxy trong máu là một sản phẩm hứa hẹn và có đóng góp tích cực cho việc chăm sóc sức khỏe của con người