TRƯỜNG ĐẠI HỌC KHOA HỌC

KHOA CÔNG NGHỆ THÔNG TIN

HỆ THỐNG ĐO NHỊP TIM VÀ GỬI DỮ LIỆU QUA ESP32

TÊN LỚP HỌC PHẦN : Phát triển ứng dụng IoT

MÃ HỌC PHẦN: 2024-2025.2.TIN4024.005

GIẢNG VIÊN HƯỚNG DẪN: Võ Việt Dũng

HUẾ, THÁNG 4 NĂM 2025

Mục Lục

I.Mở đầu

II.Nội Dung

1. Giới thiệu về ESP32

-ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems,kế thừa từ Soc ESP8266.Nó tích hợp cả wifi và bluetooth,phù hợp cho dự án về IoT(Internet of Things)

2. Cảm biến MAX30100

Cảm biến MAX30100 là cảm biến đo nhịp tim và nồng độ oxy trong máu bằng phương pháp quang học. Cảm biến này sử dụng đèn LED hồng ngoại và đỏ để phát hiện sự thay đổi trong lưu lượng máu dưới da.

2.1. Thông số kỹ thuật của MAX30100

- Điện áp hoạt động: 1.8V - 3.3V
- Giao tiếp: I2C
- Tích hợp LED hồng ngoại và LED đỏ
- Tốc độ lấy mẫu: 50Hz
- Đo nhịp tim và SpO2
- Tiêu thụ điện năng thấp

2.2. Nguyên lý hoạt động

MAX30100 hoạt động dựa trên nguyên lý quang học, sử dụng hai đèn LED (đỏ và hồng ngoại) để chiếu sáng vào da. Cảm biến quang học bên trong sẽ đo lượng ánh sáng bị hấp thụ và phản xạ bởi máu, từ đó tính toán nhịp tim và nồng độ oxy trong máu. Khi máu chảy qua mạch máu, mức độ hấp thụ ánh sáng sẽ thay đổi, và cảm biến sẽ ghi lại sự thay đổi này để xác định nhịp tim.

2.3. Cơ chế hoạt động của MAX30100

1. **Phát tín hiệu ánh sáng**: MAX30100 sử dụng hai loại LED - LED đỏ và LED hồng ngoại để phát ra ánh sáng vào da.
2. **Phản xạ ánh sáng**: Khi máu chảy qua các mạch máu, một phần ánh sáng bị hấp thụ, trong khi phần còn lại phản xạ lại.
3. **Cảm biến quang học**: Bộ thu quang trong MAX30100 nhận ánh sáng phản xạ và chuyển đổi nó thành tín hiệu điện.
4. **Xử lý tín hiệu**: Vi điều khiển trong cảm biến phân tích tín hiệu để xác định biên độ dao động của ánh sáng phản xạ, từ đó tính toán nhịp tim và nồng độ oxy trong máu (SpO2).
5. **Gửi dữ liệu**: Dữ liệu đo được sẽ được truyền qua giao tiếp I2C đến vi điều khiển ESP32 để hiển thị hoặc gửi lên hệ thống lưu trữ.

2.4. Ứng dụng của cảm biến MAX30100

- Thiết bị đeo thông minh (smartwatch, fitness tracker)
- Thiết bị y tế đo nhịp tim và nồng độ oxy trong máu
- Hệ thống theo dõi bệnh nhân từ xa
- Ứng dụng thể thao giúp kiểm soát cường độ tập luyện

III. Thiết kế hệ thống

1.Phần cứng

Hệ thống bao gồm các thành phần chính sau:

- **Vi điều khiển ESP32**: Xử lý dữ liệu và truyền thông.
- **Cảm biến MAX30100**: Đo nhịp tim và nồng độ oxy trong máu.
- **Màn hình LCD 1602**: Hiển thị dữ liệu nhịp tim.
- **Nguồn điện**: Cấp nguồn cho hệ thống.
- **Dây nối và linh kiện phụ trợ**: Kết nối các thành phần.

2.Phần mềm

-PlaformIO

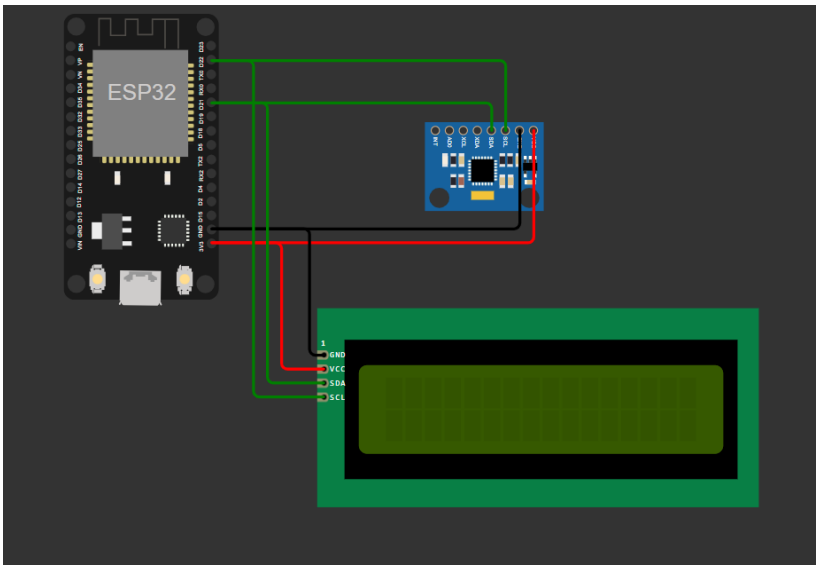
-Blynk

3.Cài đặt và lập trình

3.1Kết nối MAX30100 với ESP32 qua giao tiếp 12C:

Thiết bị MAX30100	**Chân ESP32**
MAX30100 VCC	3V3
MAX30100 GND	GND
MAX30100 SCL	GPIO 22
MAX30100 SDA	GPIO 21

3.2 Sơ đồ Diagram




```
{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp1", "top": -33.7, "left": -120.2, "attrs": {} },
    { "type": "wokwi-max30100", "id": "imu1", "top": 42.22, "left": 127.12, "attrs": {} },
    {
      "type": "wokwi-lcd1602",
      "id": "lcd1",
      "top": 169.6,
      "left": 53.6,
      "attrs": { "pins": "i2c" }
    }
  ],
```

```
  "connections": [
    [ "imu1:VCC", "esp1:3V3", "red", [ "v0" ] ],
    [ "imu1:GND", "esp1:GND.1", "black", [ "v0" ] ],
    [ "imu1:SDA", "esp1:D21", "green", [ "v0" ] ],
    [ "imu1:SCL", "esp1:D22", "green", [ "v0" ] ],
    [ "lcd1:GND", "esp1:GND.1", "black", [ "h-9.6", "v-86.4", "h-67.2" ] ],
    [ "lcd1:VCC", "esp1:3V3", "red", [ "h-28.8", "v-38.3" ] ],
    [ "lcd1:SDA", "esp1:D21", "green", [ "h-38.4", "v-163" ] ],
    [ "lcd1:SCL", "esp1:D22", "green", [ "h-48", "v0.3" ] ]
  ],
  "dependencies": {}
}
```

3.3Lập trình:

```
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"   // Thư viện cho cảm biến MAX30100
#include <LiquidCrystal_I2C.h>        // Thư viện LCD I2C
#include <Adafruit_GFX.h>             // Thư viện đồ họa cho OLED
#include <Adafruit_SSD1306.h>         // Thư viện OLED SSD1306

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// Thông tin WiFi và Blynk
#define BLYNK_TEMPLATE_ID "TMPL6xhCp3qYq"
#define BLYNK_TEMPLATE_NAME "thuanthi"
#define BLYNK_AUTH_TOKEN "aIC_5exweCvUlu33BAQrl90gS3BP_0ch"
// Đối tượng cảm biến MAX30100
PulseOximeter pox;

// Đối tượng LCD với địa chỉ I2C (ví dụ: 0x27), 16x2
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Cấu hình cho màn hình OLED
```

```
#define SCREEN_WIDTH 128 // Chiều rộng màn hình OLED
#define SCREEN_HEIGHT 64 // Chiều cao màn hình OLED
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Chân LED (kết nối với ESP32: D5)
#define LED_PIN 5

// Các biến dùng để điều khiển LED không chặn (non-blocking)
bool ledState = false;          // trạng thái LED hiện tại
unsigned long ledOnTime = 0;    // thời điểm LED được bật
const unsigned long ledInterval = 100; // thời gian bật LED (ms)

// Biến cờ báo hiệu nhịp tim được phát hiện
volatile bool beatDetected = false;

// Hàm callback khi phát hiện nhịp tim
void onBeatDetected() {
  beatDetected = true;
}

void setup() {
  Serial.begin(115200);

  // Cấu hình chân LED
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);

  // Khởi tạo LCD
  lcd.init();
  lcd.backlight();

  // Khởi tạo OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("Khởi tạo OLED thất bại");
    while (1); // Dừng lại nếu không khởi tạo được
  }
  display.clearDisplay();
```

```
  display.display();

  // Khởi tạo cảm biến MAX30100
  if (!pox.begin()) {
    Serial.println("Khởi tạo MAX30100 thất bại!");
    while (1); // Dừng lại nếu không khởi tạo được
  }

  // Gán callback khi phát hiện nhịp tim
  pox.setOnBeatDetectedCallback(onBeatDetected);

  // Khởi tạo kết nối Blynk
  Blynk.begin(auth, ssid, pass);

  Serial.println("Khởi tạo hoàn tất!");
}

void loop() {
  // Chạy Blynk để xử lý các sự kiện và gửi dữ liệu
  Blynk.run();

  // Cập nhật dữ liệu từ cảm biến MAX30100
  pox.update();

  static unsigned long lastDisplayUpdate = 0;
  unsigned long currentMillis = millis();

  // Cập nhật hiển thị và gửi dữ liệu Blynk mỗi giây
  if (currentMillis - lastDisplayUpdate > 1000) {
    lastDisplayUpdate = currentMillis;

    // Lấy giá trị BPM và SpO2
    float bpm = pox.getHeartRate();
    float spo2 = pox.getSpO2();

    // In ra Serial để theo dõi
    Serial.print("BPM: ");
```

```
  Serial.print(bpm);
  Serial.print(" - SpO2: ");
  Serial.println(spo2);

  // Cập nhật dữ liệu lên LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("BPM: ");
  lcd.print(bpm);
  lcd.setCursor(0, 1);
  lcd.print("SpO2: ");
  lcd.print(spo2);

  // Cập nhật dữ liệu lên OLED
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.print("BPM: ");
  display.println(bpm);
  display.print("SpO2: ");
  display.println(spo2);
  display.display();

  // Gửi dữ liệu qua Blynk
  Blynk.virtualWrite(V1, bpm);
  Blynk.virtualWrite(V2, spo2);
}

// Nếu phát hiện nhịp tim, bật LED mà không dùng delay()
if (beatDetected) {
  digitalWrite(LED_PIN, HIGH);
  ledState = true;
  ledOnTime = currentMillis;
  beatDetected = false;  // Reset cờ nhịp tim
}
```

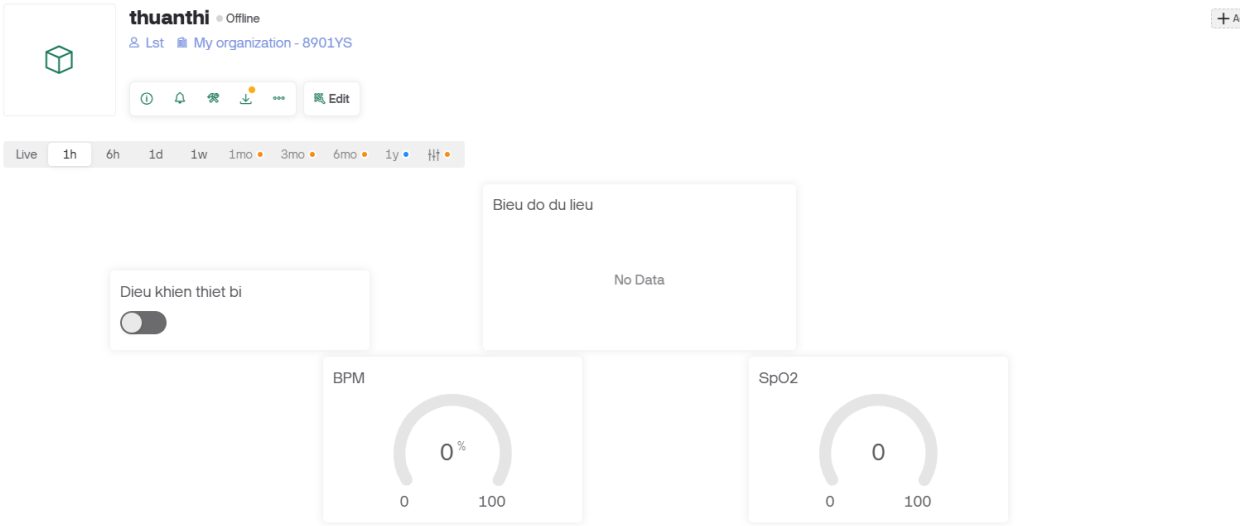
```
 // Kiểm tra nếu LED đang bật và đã bật đủ thời gian thì tắt
 if (ledState && (currentMillis - ledOnTime >= ledInterval)) {
   digitalWrite(LED_PIN, LOW);
   ledState = false;
 }
}
```

3.4 Kết nối blynk:

#define BLYNK_TEMPLATE_ID "TMPL6xhCp3qYq"

#define BLYNK_TEMPLATE_NAME "thuanthi"

#define BLYNK_AUTH_TOKEN "aIC_5exweCvUlu33BAQrl90gS3BP_0ch"



IV. Kết luận

Hệ thống sử dụng ESP32 và cảm biến MAX30100 để đo nhịp tim với độ chính xác cao, giúp theo dõi sức khỏe cá nhân một cách hiệu quả.

- Dữ liệu nhịp tim sau khi thu thập được xử lý và gửi đến ứng dụng di động hoặc lưu trữ trên đám mây để người dùng dễ dàng theo dõi.

- ESP32 đóng vai trò trung gian quan trọng, giúp kết nối cảm biến với các nền tảng lưu trữ và hiển thị dữ liệu thông qua WiFi hoặc Bluetooth.

- Ứng dụng của hệ thống rất đa dạng, bao gồm giám sát sức khỏe trong y tế, hỗ trợ vận động viên trong thể thao và tích hợp vào các thiết bị đeo thông minh.

- Người dùng có thể theo dõi nhịp tim theo thời gian thực, nhận cảnh báo khi phát hiện dấu hiệu bất thường để kịp thời xử lý.

- Việc lưu trữ dữ liệu trên đám mây giúp phân tích sức khỏe lâu dài, hỗ trợ bác sĩ và chuyên gia y tế trong việc chẩn đoán bệnh.

- Một số yếu tố như nhiễu tín hiệu, tác động của ánh sáng môi trường và chuyển động của cơ thể có thể ảnh hưởng đến độ chính xác của phép đo.

- Để cải thiện hiệu suất, có thể áp dụng các thuật toán lọc nhiễu và hiệu chỉnh tín hiệu nhằm tăng độ chính xác và độ tin cậy của hệ thống.

- Việc tích hợp trí tuệ nhân tạo (AI) có thể giúp phân tích dữ liệu sâu hơn, phát hiện các bất thường về nhịp tim một cách chính xác hơn.

- Trong tương lai, hệ thống có thể được nâng cấp với các cảm biến tiên tiến hơn, cải thiện tính năng kết nối và mở rộng phạm vi ứng dụng trong chăm sóc sức khỏe.

TÀI LIỆU THAM KHẢO

- Datasheet cảm biến MAX30100
(https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf)

- Trang web chính thức của Espressif

(https://www.espressif.com/)

- Tài liệu hướng dẫn sử dụng ESP32

(https://docs.espressif.com/projects/esp-idf/en/latest/)

- Arduino Forum - Thảo luận và hỗ trợ kỹ thuật về ESP32
(https://forum.arduino.cc/c/using-arduino/hardware/esp32/93)

- GitHub - Các dự án mẫu liên quan đến ESP32 và MAX30100
(https://github.com/topics/esp32-max30100)

- Stack Overflow - Hỏi đáp kỹ thuật liên quan đến ESP32 và MAX30100
(https://stackoverflow.com/questions/tagged/esp32)

- Tài liệu hướng dẫn lập trình ESP32 với Arduino:
https://randomnerdtutorials.com/esp32-tutorials/)