ĐẠI HỌC HUẾ

TRƯỜNG ĐẠI HỌC KHOA HỌC

KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI TIỂU LUẬN

HỆ THỐNG ĐO NHỊP TIM VÀ GỬI DỮ LIỆU QUA ESP32

Sinh viên thực hiện: NGÔ NGUYỄN ĐỨC QUÝ

Khóa: K45 - Hệ chính quy

Huế, tháng 4 – năm 2025

ĐẠI HỌC HUẾ

TRƯỜNG ĐẠI HỌC KHOA HỌC

KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI TIỂU LUẬN

HỆ THỐNG ĐO NHỊP TIM VÀ GỬI DỮ LIỆU QUA ESP32

Sinh viên thực hiện: NGÔ NGUYỄN ĐỨC QUÝ

Khóa: K45 - Hệ chính quy

Giảng viên hướng dẫn: ThS. Võ Việt Dũng

Huế, tháng 4 – năm 2025

LỜI CÁM ƠN

Bài tiểu luận về đề tài: Hệ thông đo nhịp tim và gửi dữ liệu qua ESP32 thuộc bộ môn Phát triển ứng dụng IOT là kết quả của quá trình học tập, tiếp thu kiến thức tại trường, lớp và cả những tìm tòi, nghiên cứu riêng của bản thân em và sự hướng dẫn của thầy Võ Việt Dũng. Do vậy, qua đây em xin phép được gửi lời cảm ơn chân thành nhất tới thầy. Mặc dù đã dành nhiều thời gian và nỗ lực để hoàn thành bài tiểu luận này, nhưng do sự hạn chế về mặt kiến thức nên bài làm khó tránh khỏi những thiếu sót. Em kính mong nhận được những lời góp ý của quý thầy, cô để bài làm ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

Huế, tháng 4 năm 2025

Sinh viên thực hiện

Ngô Nguyễn Đức Quý

DANH MỤC CÁC TỪ VIẾT TẮT

ABBREVIATIONS	MEANING
IoT	Internet of Things
ESP32	Espressif Systems 32

MỤC LỤC

PHẦN MỞ ĐẦU

Trong thời đại công nghệ Internet of Things (IoT) phát triển mạnh mẽ, các hệ thống giám sát sức khỏe thông minh ngày càng thu hút sự quan tâm nhằm hỗ trợ con người theo dõi tình trạng thể chất một cách nhanh chóng, chính xác và tiện lợi. Một trong những chỉ số quan trọng để đánh giá sức khỏe tim mạch là nhịp tim và nồng độ oxy trong máu (SpO2). Việc theo dõi liên tục hai thông số này giúp phát hiện sớm các bất thường như nhịp tim nhanh (Tachycardia), nhịp tim chậm (Bradycardia) hay thiếu oxy máu, từ đó có biện pháp xử lý kịp thời, đặc biệt đối với người cao tuổi, bệnh nhân tim mạch hoặc vận động viên.

Trước nhu cầu đó, đề tài "Hệ thống đo nhịp tim và SpO2 bằng ESP32" được thực hiện nhằm tận dụng vi điều khiển ESP32 DevKit-C V4 – một nền tảng mạnh mẽ với khả năng kết nối Wi-Fi và Bluetooth – để thu thập, hiển thị và truyền dữ liệu nhịp tim, SpO2 theo thời gian thực. Hệ thống sử dụng màn hình OLED SSD1306 để hiển thị thông tin và gửi cảnh báo qua Telegram khi phát hiện các dấu hiệu bất thường. Đặc biệt, dữ liệu nhịp tim và SpO2 còn được đồng bộ lên nền tảng Blynk, cho phép người dùng theo dõi tình trạng sức khỏe từ xa thông qua ứng dụng trên điện thoại.

Hệ thống được mô phỏng trên nền tảng Wokwi, giúp kiểm tra và đánh giá hoạt động trước khi triển khai thực tế. Bài tiểu luận sẽ trình bày chi tiết các khía cạnh của hệ thống, từ khái niệm, thông số kỹ thuật, nguyên lý hoạt động, thiết kế phần cứng, lập trình phần mềm đến ưu điểm, hạn chế và ứng dụng thực tiễn. Qua đó, bài viết cung cấp cái nhìn tổng quan về ứng dụng IoT trong lĩnh vực chăm sóc sức khỏe, góp phần nâng cao chất lượng cuộc sống trong thời đại số hóa.

PHẦN NỘI DUNG

I. Mục tiêu của đề tài

1. Mục tiêu:

Bài tiểu luận này cung cấp cho bạn một cái nhìn tổng quan về công nghệ Internet of Things (IoT), ứng dụng, các thành phần quan trọng của một hệ thống IoT. Và quan trọng là chúng ta làm được hệ thống cảm biến nhịp tim và gửi dữ liệu qua ESP32.

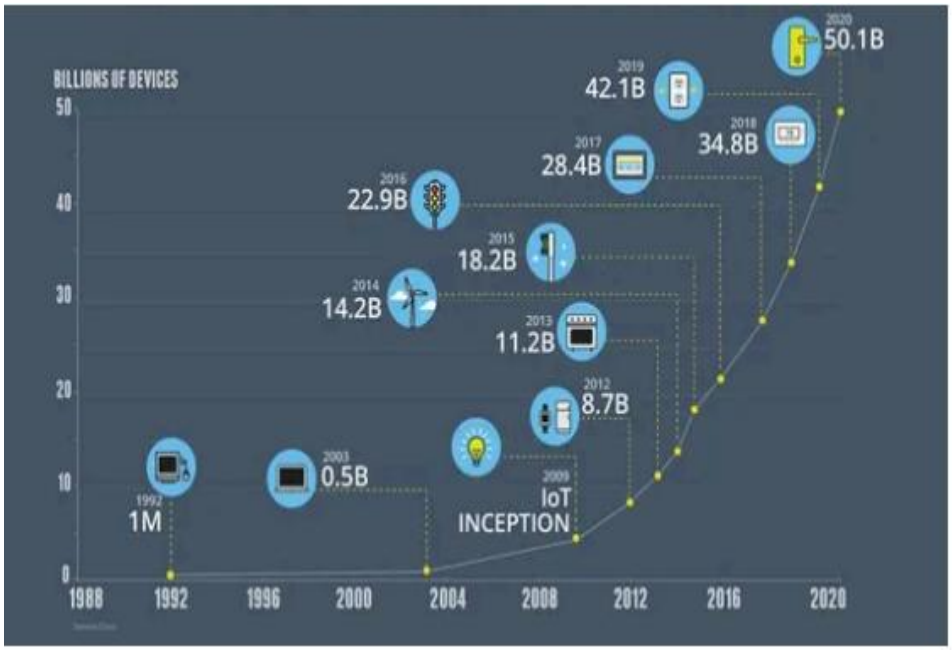
2. Khái niệm:

IoT (Internet of Things), hay Công nghệ Vạn vật kết nối Internet, là một khái niệm chỉ sự kết nối của các thiết bị vật lý và ảo với Internet thông qua các cảm biến và thiết bị giám sát, cho phép chúng thu thập và chia sẻ dữ liệu. Bằng cách kết hợp các công nghệ kết nối không dây, mạng cảm biến, dữ liệu lớn và khai thác thông minh, IoT đã mở ra kỷ nguyên công nghệ mới vượt trội.

Hình 1. *Tương tác các ứng dụng*

Cảm biến thu thập thông tin từ thế giới xung quanh, chẳng hạn như đo nhiệt độ, ánh sáng, tốc độ hoặc vị trí... Thiết bị truyền động chuyển đổi tín hiệu điện thành các tương tác trong thế giới thực như kích hoạt công tắc, bật đèn, tạo âm thanh hoặc gửi tín hiệu điều khiển đến phần cứng khác, chẳng hạn như để bật ổ cắm điện. IoT là một lĩnh vực công nghệ đang phát triển nhanh chóng. Nhờ việc tối ưu kích thước và giá thành của các cảm biến cũng như thiết bị truyền động, mà các ứng dụng IOT có thể dễ dàng được hiện thực và tích hợp. Ước tính đến cuối năm 2020, 50 tỷ thiết bị IoT đã được triển khai và kết nối vào mạng Internet. Nhìn về tương lai, ước tính đến năm 2025, các thiết bị IoT sẽ thu thập được gần 80 zettabyte dữ liệu hoặc 80 nghìn tỷ gigabyte. Đó là rất nhiều dữ liệu, là nguồn thông tin quan trọng cho các công nghệ phía sau IOT, chẳng hạn như phân tích dữ liệu, khoa học dữ liệu hay các bộ não nhân tạo.

3. Lịch sử và sự phát triển:



Hình 2. *Sự phát triển của các thiết bị*

- **Năm 1982 - 1990: Giai đoạn sơ khai**

Trong những năm từ 1982 đến 1990, IoT lúc này chưa có gì là rõ ràng khi đây vẫn chỉ là ý tưởng đưa các cảm biến, trí thông minh vào các vật, mang đến một mạng lưới thiết bị thông minh. Tuy nhiên tiến độ thực hiện các ý tưởng này được diễn ra khá chậm, vì bấy giờ công nghệ chưa đạt đủ trình độ để thực hiện hóa.

- **Năm 1991 - 1994: Những khái niệm phổ quát đầu tiên**

Năm 1991, Mark Weiser đã đưa ra khái niệm về điện toán phổ quát, cho thấy tầm nhìn của một môi trường máy tính toàn cầu với các thiết bị nhúng thông minh. Năm 1994, Reza Raji mô tả khái niệm IoT như "chuyển các gói dữ liệu nhỏ sang tập hợp các nút mạng lớn, để tích hợp và tự động hóa mọi thứ từ các thiết bị gia dụng với cả một nhà máy sản xuất".

- **Năm 1999: Khái niệm IoT ra đời và bắt đầu phổ biến**

Khái niệm IoT được đưa ra lần đầu tiên vào năm 1999 bởi Kevin Ashton, một nhà nghiên cứu tại MIT. Tuy nhiên, công nghệ cần thiết để thực hiện ý tưởng này chưa sẵn sàng cho đến khi Internet và công nghệ không dây trở nên phổ biến hơn.

Trong những năm tiếp theo, nhiều công ty đã đưa ra các giải pháp IoT trong đó có Microsoft và Novell là nổi bật nhất. Năm 1999, Bill Joy đã đề xuất phương thức truyền tải thiết bị-tới-thiết bị (D2D) trong bộ khung "Six Webs" của ông, được ông trình bày tại Diễn đàn Kinh tế Thế giới ở Davos.

Trong cùng năm 1999, Trung tâm Auto-ID tại Viện Công nghệ Massachusetts đã đưa ra khái niệm Internet Vạn Vật và công nghệ Nhận dạng qua tần số vô tuyến (RFID) được xem là một điều kiện tiên quyết cho IoT. Từ đó, IoT đã trở nên phổ biến và phát triển nhanh chóng, với sự hội tụ của nhiều công nghệ như truyền tải vô tuyến, phân tích dữ liệu thời gian thực, học máy, cảm biến và hệ thống nhúng.

- **Năm 2000 - 2013: Tích hợp các công nghệ mới cho IoT**

Các công nghệ IoT đã được phát triển rộng rãi, bao gồm các chuẩn giao tiếp IoT như: MQTT, CoAP, Zigbee, Z-Wave, BLE, WiFi, và 6LoWPAN. Ngoài ra, việc tăng cường khả năng kết nối với IPv6 đã cho phép IoT trở nên phổ biến hơn và dễ dàng kết nối với các thiết bị khác nhau trên toàn cầu.

- **Năm 2013-2016: Phát triển công nghệ IoT**

Trong giai đoạn này, IoT đã được tích hợp với các công nghệ mới như điện toán đám mây và Big Data để thu thập và phân tích dữ liệu từ các thiết bị IoT. Internet of Everything (IoE)* và 5G cũng đã được giới thiệu vào năm 2016, đưa IoT đến một tầm cao mới về tốc độ truyền thông và khả năng kết nối.

**Chú thích: Internet of Everything (IoE) là một khái niệm mở rộng hơn, đưa ra một khái niệm về mạng lưới các thực thể được kết nối với nhau, bao gồm cả con người, các thiết bị, quy trình và dịch vụ, nhằm tạo ra các giá trị kinh tế và xã hội mới. IoE tập trung vào việc kết nối không chỉ các thiết bị IoT mà cả những người dùng, hệ thống và quy trình.*

- **Năm 2016 - nay: Phát triển cùng cách mạng công nghiệp 4.0**

IoT đã trở thành một phần quan trọng trong cuộc cách mạng công nghiệp 4.0, với sự hội tụ của nhiều công nghệ như truyền tải vô tuyến, phân tích dữ liệu thời gian thực, học máy, cảm biến và hệ thống nhúng. IoT cũng đã được áp dụng rộng rãi trong nhiều lĩnh vực từ nhà thông minh đến tự động hóa công trình.

4. Ứng dụng:

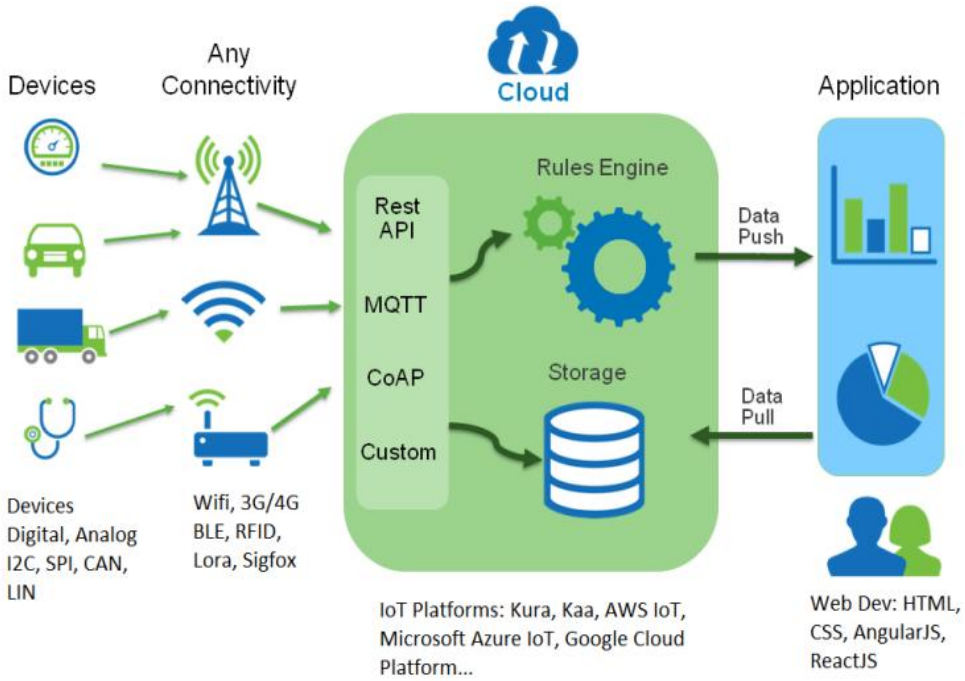
Công nghệ IoT có thể được ứng dụng trong rất nhiều lĩnh vực và có thể chia thành 1 số nhóm lớn như:

- Giải pháp tiêu dùng (smart home, wearable)
- Giải pháp thương mại, doanh nghiệp (smart tracking, building management system…)
- Công nghiệp (smart factory, smart agriculture, smart healthcare…)
- Cơ sở hạ tầng (smart city, smart grid…)

Hình 3. *IoT ứng dụng vào những lĩnh vực*

5. Các thành phần trong hệ thống IoT

*Dưới đây là kiến trúc một hệ thống IoT tiêu biểu:



Hình 4. *Kiến trúc hệ thống IoT*

- Thiết bị (Things):

● Là các thiết bị được tích hợp cảm biến, bộ vi điều khiển và có khả năng kết nối ra Internet hoặc các gateway trong hệ thống.

● Có nhiệm vụ thu thập dữ liệu về môi trường xung quanh như nhiệt độ, độ ẩm, áp suất, chuyển động, v.v.

● Giao tiếp và chia sẻ dữ liệu với các thiết bị khác qua mạng Internet.

● Ví dụ: cảm biến nhiệt độ, camera thông minh, đồng hồ thông minh, v.v.

IoT Gateway:

● Hỗ trợ và là trung gian kết nối giữa các thiết bị IoT và mạng Internet.

● Hỗ trợ nhiều giao thức kết nối khác nhau như Wi-Fi, Bluetooth, Zigbee, v.v.

● Giúp quản lý và bảo mật dữ liệu truyền tải giữa các thiết bị.

● Giúp xử lý các dữ liệu thu thập được ở cục bộ trước khi đưa lên Internet để tăng hiệu suất và giảm chi phí xử lý (Edge computing).

- IoT Platform:

● Nhận và lưu trữ các dữ liệu thu thập được từ các thiết bị IoT.

● Phân tích dữ liệu thu thập được từ các thiết bị IoT.

● Hỗ trợ ra quyết định và thực hiện các hành động tự động theo kịch bản được cấu hình sẵn (Rules Engine).

● Thường được triển khai trên các dịch vụ đám mây

- Giao diện người dùng (User Interface):

● Cung cấp giao diện cho người dùng tương tác với hệ thống IoT.

● Hiển thị dữ liệu được thu thập và phân tích từ các thiết bị IoT.

● Cho phép người dùng điều khiển và quản lý các thiết bị IoT.

● Bao gồm ứng dụng di động, trang web, bảng điều khiển, v.v.

● Ví dụ: ứng dụng Smart Home, bảng điều khiển điều khiển tòa nhà, v.v. Ngoài ra, hệ thống IoT có thể bao gồm các thành phần khác như:

● Hệ thống lưu trữ: lưu trữ dữ liệu thu thập được từ các thiết bị IoT.

● Hệ thống bảo mật: bảo vệ hệ thống IoT khỏi các mối đe dọa an ninh mạng.

● Hệ thống quản lý: quản lý và điều phối các hoạt động của hệ thống IoT. Tóm lại, một hệ thống IoT bao gồm nhiều thành phần khác nhau hoạt động cùng nhau để thu thập, truyền tải, phân tích và xử lý dữ liệu từ các thiết bị IoT. Nhờ đó, hệ thống IoT giúp con người đưa ra quyết định sáng suốt hơn, tự động hóa các quy trình và cải thiện chất lượng cuộc sống.

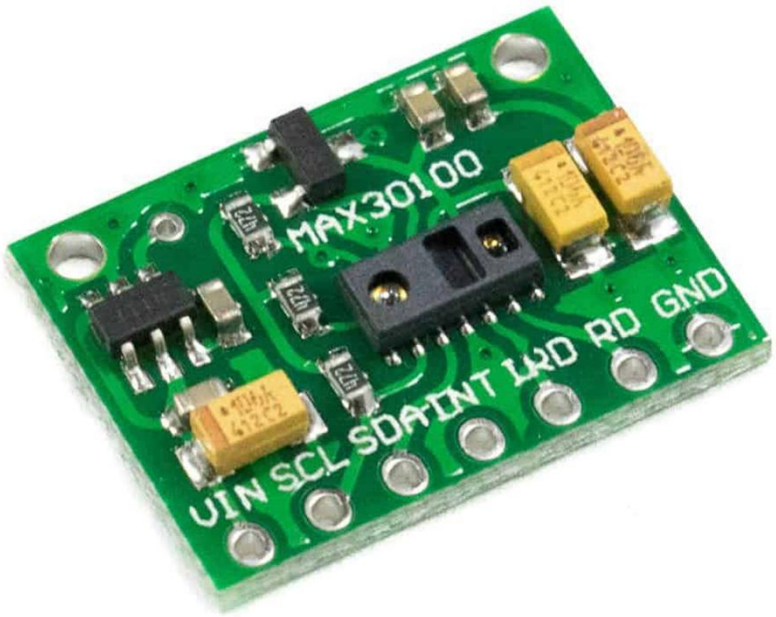
II. Cấu trúc hệ thống cảm biến nhịp tim

1. ESP32 DevKit-C V4

***Hình 5**. ESP32 DevKit-C V4*

- **Khái niệm:** ESP32 DevKit-C V4 là phiên bản phát triển của vi điều khiển ESP32 do Espressif Systems thiết kế, tích hợp Wi-Fi và Bluetooth, phù hợp cho các ứng dụng IoT.
- **Thông số kỹ thuật:**
    - **CPU:** Xtensa dual-core 32-bit LX6, 160-240 MHz.
    - **RAM:** 520 KB SRAM.
    - **Kết nối:** Wi-Fi 802.11 b/g/n, Bluetooth v4.2.
    - **GPIO:** 36 chân (ADC, DAC, PWM, I2C, SPI).
    - **Nguồn:** 5V qua USB hoặc 3.3V qua pin, dòng tiêu thụ 80 mA.

2. Cảm biến MAX30100



Hình 6. MAX301000

- **Khái niệm:** MAX30100 là cảm biến đa năng được sử dụng cho nhiều ứng dụng. Là cảm biến theo dõi nhịp tim và cũng là máy đo oxy. Cảm biến có hai diode phát sáng, một cảm biến quang (photodetector) và các linh kiện xử lý tín hiệu để phát hiện nhịp tim và đo xung oxy.
- **Thông số kỹ thuật:**
  - Điện áp hoạt động từ 1,8V đến 3,3V
  - Dòng điện đầu vào 20mA
  - Tích hợp loại bỏ nhiễu từ ánh sáng xung quanh
  - Tốc độ lấy mẫu tín hiệu cao
  - Xuất đầu ra dữ liệu nhanh

3. Màn hình LCD 16x2 (I2C)

Hình 7. *Màn hình LCD 16*x2 (I2C)

- **Khái niệm:** LCD 16x2 là màn hình tinh thể lỏng hiển thị 16 ký tự trên 2 dòng, tích hợp module I2C để giao tiếp với vi điều khiển qua giao thức I2C .
- **Thông số kỹ thuật:**
    - **Kích thước:** 16 ký tự x 2 dòng.
    - **Điện áp:** 5V.
    - **Giao thức:** I2C (SDA, SCL).
    - **Địa chỉ I2C:** 0x27 (có thể thay đổi).
    - **Dòng tiêu thụ:** 20-40 mA.

4. Buzzer (Còi cảnh báo)

Hình 8. *Buzzer (Còi cảnh báo)*

- **Khái niệm:** Buzzer là thiết bị âm thanh nhỏ gọn, hoạt động dựa trên dao động điện từ hoặc áp điện để tạo âm thanh cảnh báo.

- **Thông số kỹ thuật:**
    - **Điện áp:** 3.3-5V.
    - **Tần số:** 2-4 kHz.
    - **Độ ồn:** 85-100 dB.
    - **Dòng tiêu thụ:** 15-30 mA.
    - **Kích thước:** Đường kính 12 mm.

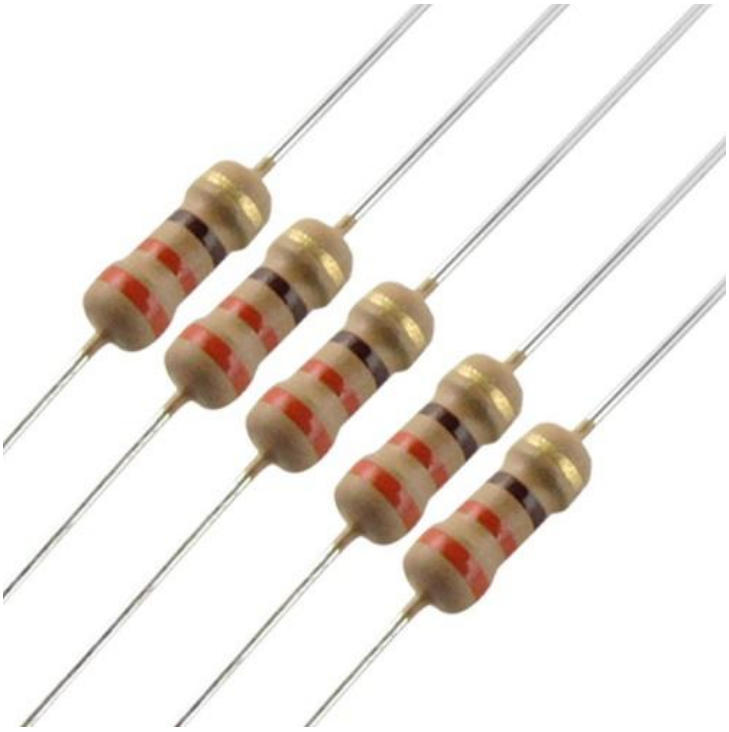
5. LED đỏ (Đèn báo động)

Hình 9. *LED đỏ (Đèn báo động)*

- **Khái niệm:** LED đỏ là diode phát quang, sử dụng để báo hiệu bằng ánh sáng khi có sự kiện xảy ra.
- **Thông số kỹ thuật:**
    - **Điện áp:** 1.8-2.2V.

- **Dòng điện:** 10-20 mA.
- **Màu sắc:** Đỏ.
- **Độ sáng:** 100-200 mcd.

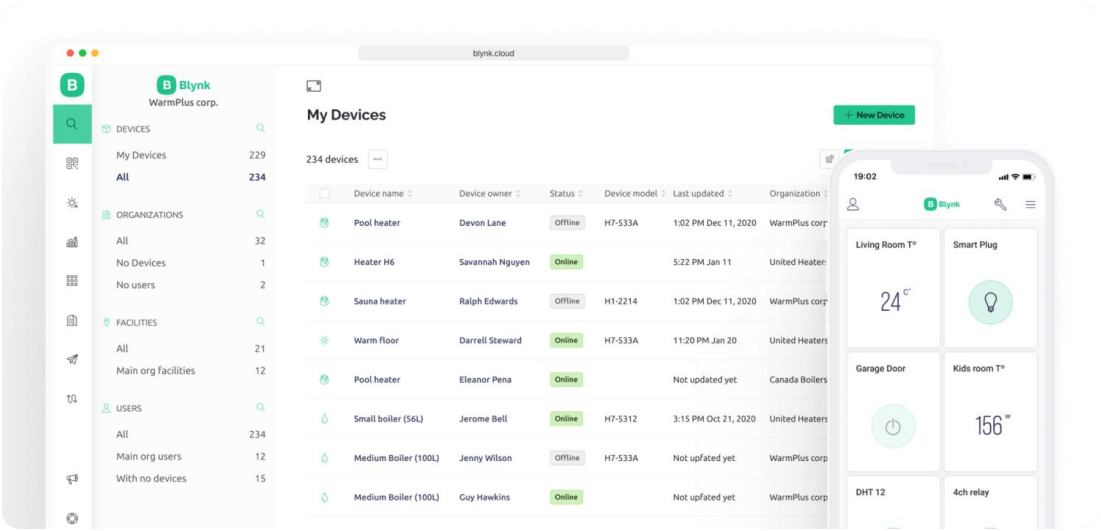
6. Điện trở 220Ω



Hình 10. *Điện trở 220* **Ω**

- **Khái niệm:** Điện trở 220Ω là linh kiện thụ động, dùng để hạn dòng cho LED, đảm bảo hoạt động an toàn.
- **Thông số kỹ thuật:**
  - **Giá trị:** 220 Ohm ± 5%.
  - **Công suất:** 0.25W.
  - **Kích thước:** Chuẩn 1/4W.

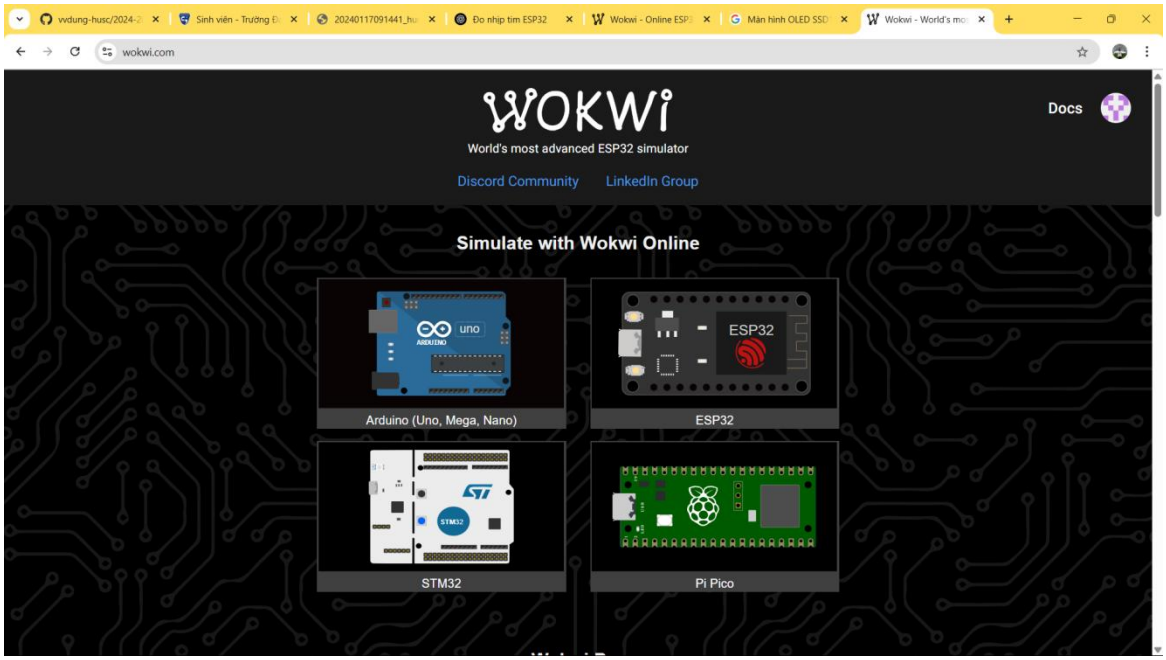
7. Web/Ứng dụng Blynk



Hình 11. *Web/Ứng dụng Blynk*

- **Giới thiệu:** Blynk được phát triển bởi Blynk Inc., ra mắt năm 2014, là nền tảng IoT phổ biến giúp thiết kế giao diện điều khiển và giám sát thiết bị qua điện thoại [6].
- **Khái niệm:** Blynk là nền tảng IoT cho phép giám sát và điều khiển thiết bị từ xa qua ứng dụng di động.
- **Đặc tả dịch vụ:**
    - **Hỗ trợ:** Wi-Fi, Bluetooth, Ethernet.
    - **Giao thức:** TCP/IP, HTTP, WebSocket.
    - **Nền tảng:** iOS, Android, Web.
    - **Dung lượng:** 100 thiết bị miễn phí.
    - **Thời gian phản hồi:** 1-3 giây.

8. Wokwi



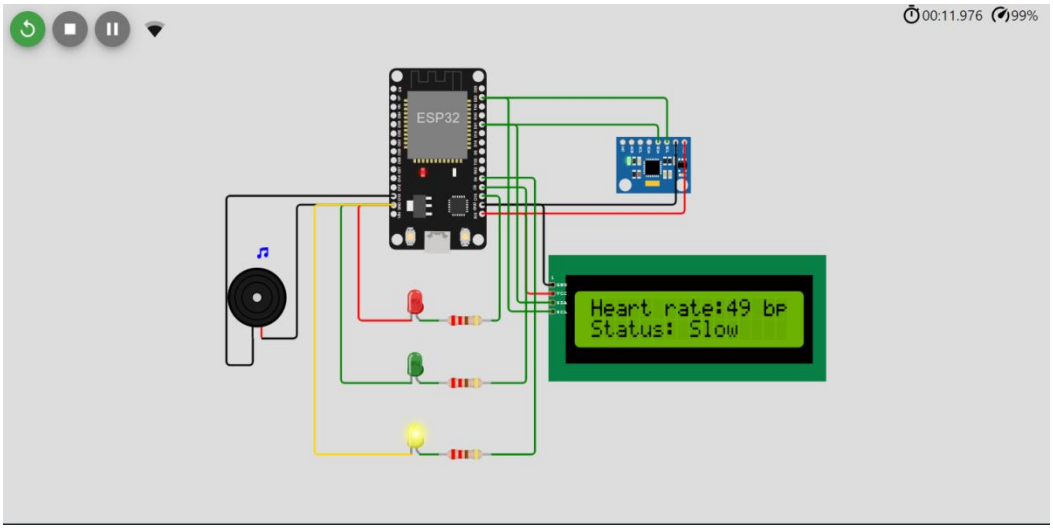
Hình 12. *Wokwi*

- **Giới thiệu:** Wokwi là công cụ mô phỏng phần cứng trực tuyến, ra mắt năm 2020 bởi nhóm CodeCraft, hỗ trợ thử nghiệm các dự án IoT.
- **Khái niệm:** Wokwi là nền tảng mô phỏng trực tuyến cho phép thiết kế, lập trình và kiểm tra mạch điện tử với vi điều khiển như ESP32.
- **Đặc tả dịch vụ:**
    - **Hỗ trợ:** ESP32, Arduino, Raspberry Pi Pico**.**
    - **Ngôn ngữ:** C/C++, MicroPython.
    - **Giao diện:** Trình chỉnh sửa mã và sơ đồ mạch.
    - **Định dạng:** Hỗ trợ JSON.
    - **Truy cập:** Web-based.

III. Nguyên lý hoạt động của hệ thống cảm biến nhịp tim

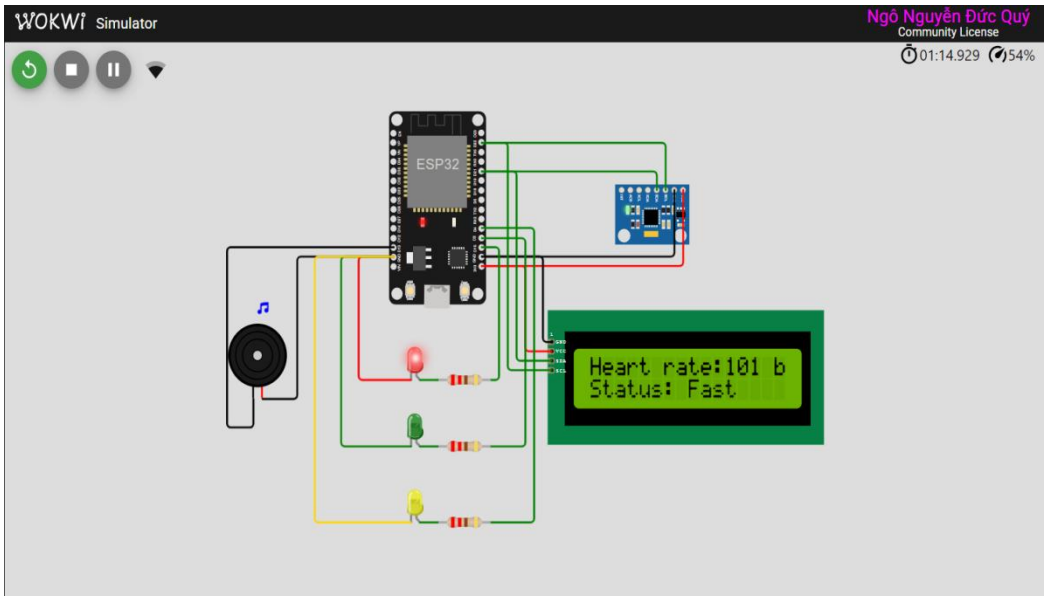
Hệ thống đo nhịp tim và gửi dữ liệu qua ESP32 DevKit-C V4 hoạt động dựa trên sự phối hợp giữa các linh kiện phần cứng và phần mềm, được chia thành các giai đoạn cụ thể như sau:

1. **Thu thập dữ liệu từ cảm biến:**
   - Cảm biến MPU6050(mô phỏng cảm biến MAX30100) liên tục đo nhịp tim bằng cách sử dụng các LED hồng ngoại và LED đỏ để phát hiện sự thay đổi cơn hồng ngoại phản chiếu từ máu.
   - Dữ liệu thu thập được chuyển thành tín hiệu số và gửi về ESP32 qua giao tiếp I2C.
2. **Xử lý dữ liệu:**
   - ESP32 nhận tín hiệu từ MPU6050, tiến hành xử lý tín hiệu và tính toán nhịp tim (bpm - beats per minute).
   - Hệ thống so sánh nhịp tim với ngưỡng an toàn:
     - Nhỏ hơn **60 bpm**: Nhịp tim chậm.
     - Lớn hơn **100 bpm**: Nhịp tim nhanh.
     - Trong khoảng **60 - 100 bpm**: Nhịp tim bình thường.
3. **Kích hoạt cảnh báo khi nhịp tim bất thường:**
   - **Nhịp tim chậm (< 60 bpm):**
     - **LED vàng** sáng (GPIO 14) để cảnh báo.
     - **Buzzer** phát âm thanh cảnh báo.
     - **Màn hình LCD** hiển thị "Status: Slow".
     - **Gửi thông báo qua Blynk**: " Heart Slow ".



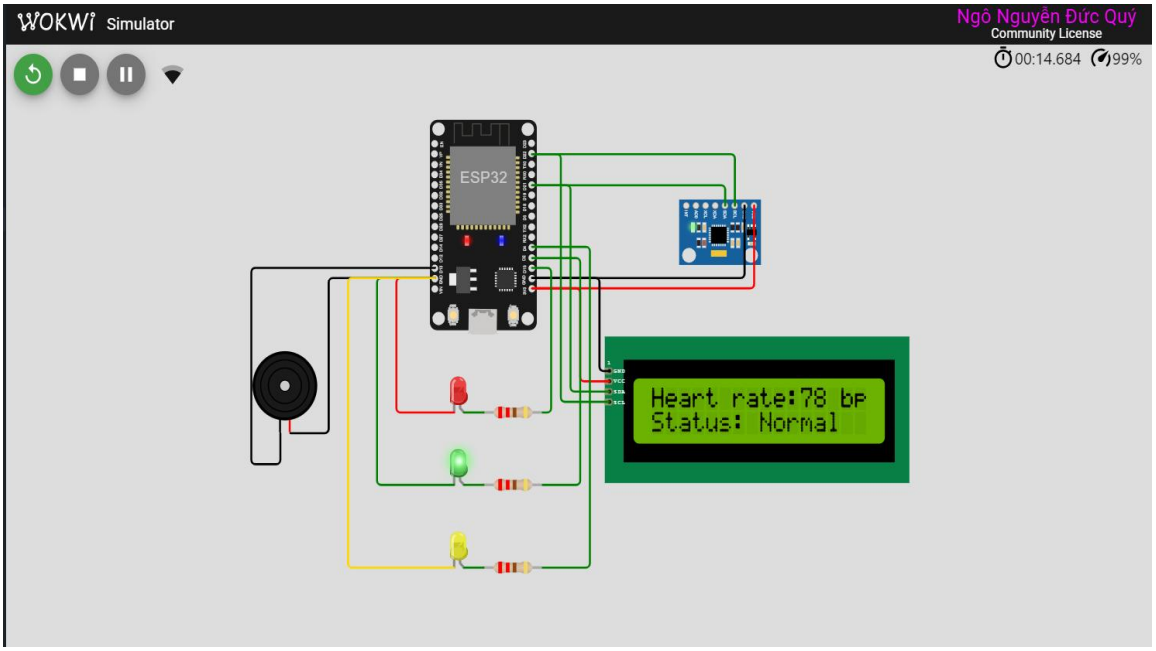
Hình 13. *Trạng thái cảnh báo khi nhịp tim chậm*

- **Nhịp tim nhanh (> 100 bpm):**
  - **LED đỏ** sáng (GPIO 27) để cảnh báo.
  - **Buzzer** phát âm thanh cảnh báo.
  - **Màn hình LCD** hiển thị "Status: Fast".
  - **Gửi thông báo qua Blynk**: "Heart Fast



Hình 14. *Trạng thái cảnh báo khi nhịp tim nhanh*

4. **Trạng thái nhịp tim bình thường:**
    - Khi nhịp tim nằm trong khoảng an toàn (60 - 100 bpm):
        - **LED xanh** sáng (GPIO 12), báo hiệu tình trạng ổn định.
        - **Buzzer** tắt.
        - **Màn hình LCD** hiển thị "Status: Normal".
        - **Gửi thông báo qua Blynk**: "Heart Normal



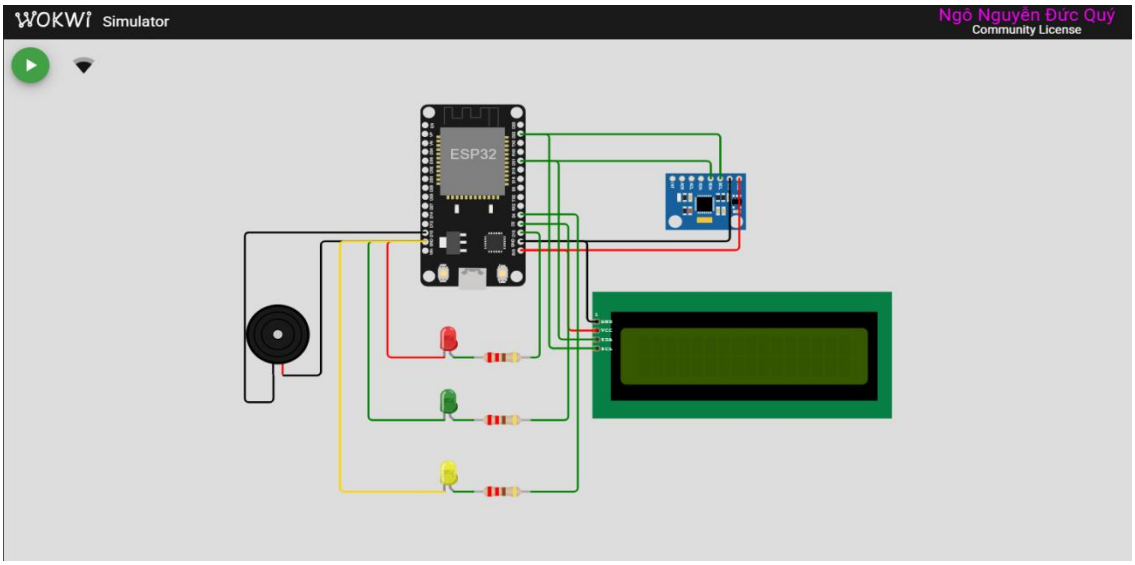
Hình 15. *Trạng thái khi nhịp tim bình thường*

5. **Chu kỳ hoạt động:**

- Hệ thống hoạt động theo chu kỳ lặp lại, ESP32 liên tục đọc dữ liệu từ MPU6050 mỗi 10 giây (có thể điều chỉnh trong phần mềm), đảm bảo giám sát nhịp tim liên tục và cảnh báo kịp thời.

  Nguyên lý hoạt động này tận dụng khả năng xử lý nhanh chóng của ESP32 và sự kết hợp giữa nhiều thiết bị ngoài, đem lại hiệu quả cảnh báo cao cả tại chỗ và từ xa.

IV. Thiết kế sơ đồ khối



Hình 16. *Sơ đồ hệ thống cảm biến nhịp tim*

Thành phần cần có:

1. ESP32 Dev Module
2. Cảm biến nhịp tim MPU6050
3. Màn hình LCD 16x2 (I2C - PCF8574)
4. Buzzer (còi cảnh báo)
5. LED (đèn báo động)
6. Điện trở 220Ω (cho LED)
7. Dây nối (jumper)

Kết nối linh kiện:

- Cảm biến MAX30100:

    - VCC nối với 3.3V hoặc 5V
    - GND nối với GND
    - SDA nối với GPIO21
    - SCL nối với GPIO22
- LCD 20x4 (I2C):
    - VCC nối với 5V
    - GND nối với GND
    - SDA nối với GPIO21
    - SCL nối với GPIO22
- Buzzer:
    - Cực dương (+) nối với GPIO13
    - Cực âm (-) nối với GND
- LED cảnh báo:
    - LED đỏ (cảnh báo nhịp tim nhanh):
        - Cực dương (+) nối với GPIO27 (qua điện trở 220Ω)
        - Cực âm (-) nối với GND
    - LED vàng (cảnh báo nhịp tim chậm):
        - Cực dương (+) nối với GPIO14 (qua điện trở 220Ω)
        - Cực âm (-) nối với GND
    - LED xanh (trạng thái bình thường):
        - Cực dương (+) nối với GPIO12 (qua điện trở 220Ω)
        - Cực âm (-) nối với GN

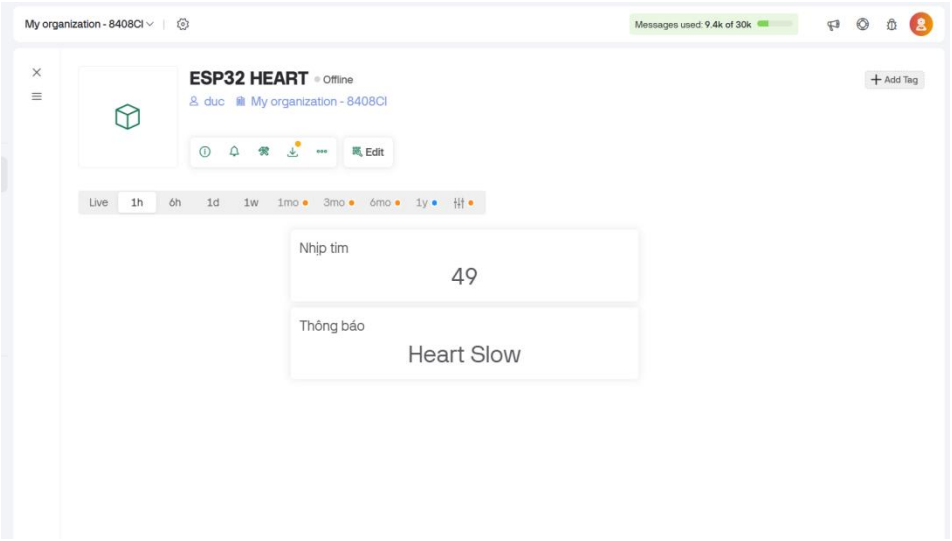
V. Lập trình

1. Phần mềm phát triển trên Arduino IDE:

- **Thư viện sử dụng:** BlynkSimpleEsp32.h, LiquidCrystal_I2C.h, Wire.h, MPU6050.h.
- **Đọc dữ liệu:** Sử dụng mpu.getAcceleration() để phân tích dao động nhịp tim.

- **Điều khiển:** digitalWrite(13, HIGH/LOW) cho buzzer, digitalWrite(27, HIGH/LOW) cho LED, lcd.print() hiển thị trên LCD.
- **Thông báo:** Blynk.notify() gửi cảnh báo qua Wi-Fi khi nhịp tim bất thường.
- **Cấu hình:**
    - Kết nối Wi-Fi với ssid: "Wokwi-GUEST", pass: "".
    - Blynk sử dụng **Auth Token** để giao tiếp với ứng dụng.
    - Giao tiếp I2C với địa chỉ LCD là 0x27.

Sử dụng gửi thông tin bằng Blynk



Hình 17. *Gửi dữ liệu và thông báo lên Blynk*

2. Phần lập trình C++ trên Visual Code

Mã lập trình C++:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Arduino.h>
#include <WiFi.h>
#include <UniversalTelegramBot.h>
#include <WiFiClientSecure.h>
#define REPORTING_PERIOD_MS 3000 // 10 giây
#define I2C_ADDR 0x27
#define LCD_COLUMNS 20
#define LCD_LINES 4
#define BLYNK_TEMPLATE_ID "TMPL6STChy6Cb"
#define BLYNK_TEMPLATE_NAME "ESP32 HEART"
#define BLYNK_AUTH_TOKEN "q5P7bB9w_tMdV4e8PjR9wxqqMbJ3VqPA"
```

```
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define BOT_TOKEN "7226596485:AAF4YaLRF30HTPW58ZL9p3TCJipO8lIrptQ"
#define CHAT_ID "-4657079728"

LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLUMNS, LCD_LINES);

const char *ssid = "Wokwi-GUEST";const char *password = "";

// Telegram bot setup
WiFiClientSecure client;
UniversalTelegramBot bot(BOT_TOKEN, client);

// Time at which the last beat occurred
uint32_t tsLastReport = 0;

// Ngưỡng nhịp tim
#define MAX_HEART_RATE 100
#define MIN_HEART_RATE 60

// Chân cho các thiết bị
#define LED_RED_PIN 15
#define LED_YELLOW_PIN 4
#define LED_GREEN_PIN 2
#define BUZZER_PIN 13


void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);

  Serial.begin(9600);
  Serial.print("Initializing..");
  lcd.print("Initializing..");

  // Cấu hình chân cho LED và Buzzer
  pinMode(LED_RED_PIN, OUTPUT);
  pinMode(LED_YELLOW_PIN, OUTPUT);
  pinMode(LED_GREEN_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  // Kết nối WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("WiFi connected");

  // Kết nối Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, password);

  // Đảm bảo rằng mọi thứ đã sẵn sàng
  delay(2000); // Thời gian khởi tạo
```

```
  lcd.println("SUCCESS");
  digitalWrite(LED_RED_PIN, HIGH);
  digitalWrite(LED_YELLOW_PIN, HIGH); // Đèn vàng
  digitalWrite(LED_GREEN_PIN, HIGH);
}

void loop()
{
  // Giả lập nhịp tim ngẫu nhiên trong khoảng 40-120 bpm
  int heartRate = random(40, 121);

  // Hiển thị nhịp tim lên LCD
  if (millis() - tsLastReport > REPORTING_PERIOD_MS)
  {
    lcd.clear();
    lcd.print("Heart rate:");
    lcd.print(heartRate);
    lcd.print(" bpm");

    // Kiểm tra nhịp tim và hiển thị trạng thái
    lcd.setCursor(0, 1); // Di chuyển con trỏ xuống dòng thứ 2
    String statusMessage;
    if (heartRate < MIN_HEART_RATE)
    {
      statusMessage = "Heart Slow";
      lcd.print("Status: Slow");

      // Điều khiển LED và Buzzer cho nhịp tim chậm
      digitalWrite(LED_RED_PIN, LOW);
      digitalWrite(LED_YELLOW_PIN, HIGH); // Đèn vàng
      digitalWrite(LED_GREEN_PIN, LOW);
      tone(BUZZER_PIN, 1000); // Phát âm thanh
    }
    else if (heartRate > MAX_HEART_RATE)
    {
      statusMessage = "Heart Fast";
      lcd.print("Status: Fast");

      // Điều khiển LED và Buzzer cho nhịp tim nhanh
      digitalWrite(LED_RED_PIN, HIGH); // Đèn đỏ
      digitalWrite(LED_YELLOW_PIN, LOW);
      digitalWrite(LED_GREEN_PIN, LOW);
      tone(BUZZER_PIN, 2000); // Phát âm thanh
    }
    else
    {
      statusMessage = "Heart Normal";
      lcd.print("Status: Normal");

      // Điều khiển LED cho nhịp tim bình thường
      digitalWrite(LED_RED_PIN, LOW);
      digitalWrite(LED_YELLOW_PIN, LOW);
      digitalWrite(LED_GREEN_PIN, HIGH); // Đèn xanh
      noTone(BUZZER_PIN);                // Tắt âm thanh
    }

    // Kiểm tra giá trị của heartRate và tình trạng LED
```

```
    Serial.print("Heart rate: ");
    Serial.println(heartRate);

    // Gửi nhịp tim và trạng thái lên Blynk (V1: Nhịp tim, V2: Trạng thái)
    Blynk.virtualWrite(V0, heartRate);     // Gửi nhịp tim lên Blynk (V1
là một ví trí ảo)
    Blynk.virtualWrite(V1, statusMessage); // Gửi trạng thái lên Blynk (V2
là một ví trí ảo)

    // Gửi dữ liệu về Telegram
    String msg = "Nhịp tim: " + String(heartRate) + " bpm\nTrạng thái: " +
statusMessage;
    bot.sendMessage(CHAT_ID, msg);

    tsLastReport = millis();
  }

  // Blynk.run() cần được gọi trong vòng lặp để duy trì kết nối Blynk
  Blynk.run();
}
```

VI. Ưu điểm và hạn chế

- Ưu điểm:
  - Đa dạng phương thức cảnh báo:
    - Hệ thống tích hợp nhiều hình thức cảnh báo như âm thanh (buzzer), ánh sáng (LED), hiển thị trực quan (LCD), và thông báo từ xa (Blynk). Điều này giúp người dùng nhận thông tin kịp thời qua nhiều kênh, cả tại chỗ và từ xa.
  - Chi phí thấp và dễ triển khai:
    - Các linh kiện như ESP32, MPU6050, LCD I2C, buzzer và LED có giá thành hợp lý, dễ mua trên thị trường.
    - Hệ thống dễ lập trình và triển khai, phù hợp với sinh viên và người mới học IoT.
  - Tính linh hoạt và khả năng mở rộng:
    - Nhờ tích hợp IoT qua Blynk, hệ thống có thể giám sát từ xa qua điện thoại.

    - ESP32 hỗ trợ nhiều chân GPIO, có thể mở rộng thêm cảm biến hoặc relay.
    - Công cụ Wokwi hỗ trợ mô phỏng trước khi triển khai thực tế, giảm rủi ro sai sót.

- Hạn chế:
  - Độ chính xác của MPU6050:
    - MPU6050 đo dao động của cơ thể để suy ra nhịp tim, nên có thể bị ảnh hưởng bởi chuyển động mạnh hoặc rung lắc ngoài ý muốn.
    - Để cải thiện độ chính xác, có thể cần hiệu chỉnh cảm biến hoặc kết hợp thuật toán lọc nhiễu.
  - Phụ thuộc vào kết nối Wi-Fi:
    - Tính năng thông báo qua Blynk yêu cầu kết nối Wi-Fi ổn định.
    - Nếu mạng bị gián đoạn, hệ thống mất khả năng gửi cảnh báo từ xa, chỉ còn hoạt động tại chỗ.
  - Phạm vi phát hiện hạn chế:
    - MPU6050 chỉ đo được nhịp tim dựa trên dao động cơ thể, nên có thể không chính xác trong mọi trường hợp, đặc biệt khi người dùng ít di chuyển.

VII. Ứng dụng thực tiễn

- **Chăm sóc sức khỏe tại nhà:**

  Hệ thống có thể được lắp đặt tại nhà để theo dõi nhịp tim của người cao tuổi hoặc bệnh nhân có tiền sử tim mạch. Khi phát hiện nhịp tim bất thường, hệ thống sẽ cảnh báo bằng âm thanh, hiển thị thông tin trên màn hình LCD và gửi thông báo qua Blynk, giúp người thân hoặc bác sĩ kịp thời can thiệp.
- **Phòng tập thể dục & thể thao:**

  Trong các trung tâm thể dục hoặc phòng gym, hệ thống có thể hỗ trợ người tập theo dõi nhịp tim trong quá trình luyện tập. Nếu nhịp tim vượt quá ngưỡng an toàn, hệ thống sẽ cảnh báo, giúp người dùng điều chỉnh cường độ tập luyện phù hợp.

- **Ứng dụng trong bệnh viện & phòng khám:**
  Các cơ sở y tế có thể tích hợp hệ thống vào giường bệnh hoặc ghế khám để theo dõi tình trạng bệnh nhân theo thời gian thực. Dữ liệu được hiển thị trực tiếp trên màn hình và gửi đến bác sĩ qua ứng dụng, hỗ trợ công tác chẩn đoán và điều trị.
- **Nghiên cứu khoa học & IoT:**
  Hệ thống có thể được sử dụng trong các dự án nghiên cứu về sức khỏe, theo dõi phản ứng sinh lý của con người trong các môi trường khác nhau. Đồng thời, đây cũng là một ứng dụng điển hình trong lĩnh vực IoT, giúp sinh viên và kỹ sư thực hành với cảm biến, vi điều khiển và nền tảng đám mây.

VIII. Phương hướng phát triển

1. Sử dụng MAX30100 thay thế cho MPU6050

- Hiệu chuẩn cảm biến:
  - Việc sử dụng MAX30100 thay cho MPU6050 giúp hệ thống đo nhịp tim chính xác hơn, chuyên dụng hơnCần bộ lọc tín hiệu số (DSP) để loại bỏ nhiễu từ chuyển động hoặc rung tay.
  - Áp dụng thuật toán lọc Butterworth hoặc Moving Average để làm mượt tín hiệu đo được.
- Cải tiến thuật toán đo nhịp tim:
  - Sử dụng thư viện "PulseSensor" hoặc "MAX30100lib" để tối ưu hóa việc đọc dữ liệu.
  - Kết hợp phương pháp FFT (Fast Fourier Transform) để phân tích tín hiệu và xác định nhịp tim chính xác hơn.

2. Tối ưu hóa khả năng kết nối và truyền dữ liệu

- Tăng tốc độ xử lý trên ESP32:

    - Cải thiện hiệu suất bằng cách sử dụng RTOS (Real-Time Operating System) trên ESP32 để xử lý đa nhiệm tốt hơn.
    - Giảm độ trễ giữa các lần đo nhịp tim để có phản hồi nhanh hơn.
- Cải tiến giao tiếp I2C:
    - MAX30100 sử dụng I2C để giao tiếp với ESP32, có thể tối ưu bằng cách giảm tần suất polling hoặc sử dụng ngắt (interrupt) để giảm tải xử lý.
    - Kiểm tra kết nối và xử lý lỗi khi cảm biến không phản hồi.

3. Ứng dụng AI và Machine Learning

- Phân tích dữ liệu nhịp tim bằng AI:
    - Áp dụng mô hình Machine Learning (ML) để phát hiện bất thường từ dữ liệu nhịp tim.
    - Sử dụng K-Means Clustering hoặc Random Forest để dự đoán các nguy cơ về tim mạch.
- Dự đoán bệnh lý tim mạch:
    - Kết hợp dữ liệu nhịp tim với các yếu tố như tuổi, huyết áp, mức độ vận động để cảnh báo nguy cơ.
    - Lưu trữ dữ liệu dài hạn trên Google Firebase hoặc ThingSpeak để phân tích xu hướng nhịp tim.

4. Cải thiện giao diện người dùng trên Blynk

- Thêm biểu đồ nhịp tim thời gian thực:
    - Hiển thị nhịp tim dưới dạng biểu đồ đường (line chart) để dễ dàng theo dõi biến động.
    - Lưu lại dữ liệu nhịp tim hàng ngày để người dùng có thể xem lại.
- Cảnh báo thông minh:
    - Khi nhịp tim vượt quá ngưỡng, hệ thống sẽ gửi cảnh báo rung (vibration alert) trên điện thoại.

- Hỗ trợ tự động gọi điện hoặc nhắn tin khẩn cấp khi phát hiện nhịp tim bất thường kéo dài.

PHẦN KẾT LUẬN

Hệ thống đo nhịp tim với ESP32 DevKit-C V4 và MPU6050 là một giải pháp IoT hiệu quả, giúp theo dõi nhịp tim theo thời gian thực, cung cấp cảnh báo đa dạng thông qua buzzer, LED, LCD và nền tảng Blynk. Nhờ thiết kế phần cứng đơn giản, chi phí thấp và khả năng mô phỏng qua Wokwi, hệ thống dễ dàng triển khai và thử nghiệm.

Hệ thống có thể ứng dụng trong nhiều lĩnh vực như giám sát sức khỏe cá nhân, hỗ trợ y tế từ xa và tích hợp vào các hệ thống nhà thông minh. Tuy nhiên, để nâng cao hiệu quả, cần khắc phục các hạn chế về độ chính xác của MPU6050, sự phụ thuộc vào Wi-Fi và tính ổn định của cảm biến. Trong tương lai, việc kết hợp trí tuệ nhân tạo (AI) để phân tích dữ liệu nhịp tim và dự đoán bất thường sẽ là một hướng phát triển tiềm năng, giúp hệ thống hoạt động thông minh và chính xác hơn.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

1. Nguyễn Minh Tú (2021). *Cảm biến MAX30100 và ứng dụng trong đo nhịp tim*, Tạp chí Công nghệ Mới, tập 10, số 01, tr. 45-49.
2. Trần Anh Tuấn (2020). *Ứng dụng cảm biến MAX30100 trong hệ thống theo dõi sức khỏe*, Nxb Khoa học và Kỹ thuật, Hà Nội.

Tiếng Anh:

3. Maxim Integrated (2014). "MAX30100: Integrated Pulse Oximeter and Heart-Rate Sensor," *Maxim Integrated Technical Document*, DS000229, Maxim Integrated, https://www.maximintegrated.com.

4. Jonsdottir, S. and Araki, Y. (2016). "Application of MAX30100 for Pulse Oximetry Monitoring in Wearable Health Devices," *Journal of Medical Devices*, vol. 10, pp. 14-20.

5. Brown, S. (2018). "Using MAX30100 for Heart Rate Monitoring in IoT Devices," *Internet of Things Journal*, vol. 5, no. 2, pp. 22-29.

6. Gomez, R., & Johnson, M. (2017). "MAX30100: A Pulse Oximeter and Heart Rate Sensor for Portable Health Monitoring," *IEEE Sensors Journal*, vol. 17, pp. 1245-1253.

7. Maxim Integrated (2015). "MAX30100: Design and Application Guide for Pulse Oximetry and Heart Rate," *Maxim Integrated Application Note*, AN1434.

PHỤ LỤC