

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA



BÁO CÁO ĐỒ ÁN

ĐỒ ÁN THIẾT KẾ LUẬN LÝ

CHỦ ĐỀ : ỨNG DỤNG IOT ĐỂ ĐO VÀ HIỂN THỊ
NHỊP TIM VÀ NỒNG ĐỘ OXY TRONG MÁU

Giảng viên hướng dẫn: Trần Thanh Bình.
Thành viên trong nhóm: Vũ Huỳnh Tấn Phát - 2114391.
Trần Tiến Đạt - 2113162.
Trần Đình Phong - 2114404.

TP. Hồ Chí Minh, 4/2023

Contents

1	Danh sách thành viên & Phân công công việc	2
2	Tóm tắt	2
3	Giới thiệu	2
4	Thiết bị	3
4.1	Cảm biến MAX30100, GY-MAX30100 và MAX30102	4
4.2	Module Esp8266	7
4.3	Kết nối thiết bị	9
5	Database và ngôn ngữ sử dụng	9
5.1	Firebase	9
5.2	HTML, CSS, JavaScript	10
5.2.1	HTML	10
5.2.2	CSS	10
5.2.3	JavaScript	11
5.3	C++ và Arduino IDE	11
6	Mã nguồn của ESP8266	12
6.1	Tiền xử lý dữ liệu	12
6.2	Xử lý dữ liệu	13
6.2.1	Đo nồng độ oxy trong máu với MAX30100 và GY-MAX30100	13
6.2.2	Đo nhịp tim với MAX30102	16
7	Web hiển thị	19
7.1	Các chức năng chính	19
7.1.1	Hiển thị kết quả đo	19
7.1.2	Hiển thị lịch sử đo	19
7.1.3	Hiển thị đồ thị cho nhịp tim	20
7.1.4	Đọc và ghi kết quả đo từ database	21
7.1.5	Xóa lịch sử đo	23
8	Ứng dụng hiển thị kết quả đo	23
8.1	Giới thiệu	23
8.1.1	Giới thiệu về Kodular	23
8.1.2	Mục đích	24
8.1.3	Phạm vi sử dụng	24
8.2	Các chức năng chính	24
8.2.1	Hiển thị kết quả đo:	24
8.2.2	Hiển thị lịch sử đo:	27
9	Kết quả thu được	29
10	Thảo luận về đề tài	29

1 Danh sách thành viên & Phân công công việc

STT	Họ và tên	MSSV	Phân công công việc	Đóng góp
1	Trần Tiến Đạt	2113162		33%
2	Vũ Huỳnh Tấn Phát	2114391		33%
3	Trần Đình Phong	2114404		33%

2 Tóm tắt

Đề tài của chúng tôi giới thiệu một công cụ được thiết kế để đo nhịp tim và mức độ oxy trong máu. Công cụ này chủ yếu được sử dụng để đánh giá các tình trạng sức khỏe cơ bản. Các triệu chứng phát sinh từ nhịp tim bất thường bao gồm các điều kiện không bình thường như tăng nhịp tim (tachycardia) và giảm nhịp tim (bradycardia). Các triệu chứng gây ra bởi mức độ oxy trong máu bao gồm các tình trạng như viêm đường hô hấp cấp (ARI) và hen suyễn, một tình trạng thở không đều. Các triệu chứng của nhịp tim và mức độ oxy không bình thường có thể chỉ ra sự xuất hiện của bệnh tật. Vì vậy, chúng tôi đã phát triển một hệ thống theo dõi nhịp tim và mức độ oxy trên nền web để nghiên cứu này.

Công cụ này sử dụng công nghệ IoT (Internet of Things) để sản xuất. Nó được lập trình với các cảm biến như cảm biến cảm biến MAX30100 và vi điều khiển ESP8266 làm bộ xử lý dữ liệu, sau đó dữ liệu được gửi qua kết nối Wi-Fi để đọc từ cảm biến một cách không dây. Dữ liệu được truyền không dây tới trang web và ứng dụng của chúng tôi theo thời gian thực. Trang web và ứng dụng sẽ hiển thị dữ liệu cảm biến được gửi từ vi điều khiển và sau đó được lưu trữ trong cơ sở dữ liệu của chúng tôi. Hệ thống thu thập hoặc theo dõi dữ liệu này giúp cho chúng tôi, đặc biệt là nhân viên y tế, kiểm tra sức khỏe và phân tích kết quả từ dữ liệu được lưu trữ trong cơ sở dữ liệu máy chủ của chúng tôi một cách dễ dàng hơn.

3 Giới thiệu

Cùng với sự phát triển của khoa học kỹ thuật, việc chăm sóc sức khỏe của con người trở nên phát triển hơn bao giờ hết. Theo thống kê của Bộ Y tế, mỗi năm có khoảng 200.000 người Việt tử vong vì bệnh tim mạch, chiếm 33% ca tử vong, cao hơn cả số ca tử vong vì ung thư hàng năm (khoảng 115.000 người). Vì vậy, vấn đề theo dõi sức khỏe con người nói chung và sức khỏe tim mạch nói riêng là cần được chú trọng. Vì vậy, việc sử dụng công nghệ không dây được tăng cường nhằm đáp ứng nhu cầu điều khiển, giám sát từ xa. Theo dõi bệnh nhân từ xa (Remote patient monitoring) là một công nghệ giúp chúng ta theo dõi bệnh nhân ngay cả khi bệnh nhân không có mặt tại bệnh viện. Nó có thể giúp dễ dàng tiếp cận các dịch vụ y tế trong khi giảm thiểu chi phí và thời gian hơn so với việc giám sát truyền thống.

Nhịp tim và nhiệt độ cơ thể gần như là những thông số đầu tiên được các bác sỹ đo sau khi bệnh nhân tới. Nhịp tim đề cập đến số lần tim co dẫn trong một đơn vị thời gian (thường là mỗi phút). Nhịp tim thay đổi ở các nhóm tuổi khác nhau. Đối với người trưởng thành từ 18 tuổi trở lên, nhịp tim khi ngủ rơi vào khoảng 72 nhịp mỗi phút (bpm). Hoạt động của tim có thể gọi là hiệu quả nếu có nhịp tim thấp hơn khi người bệnh nghỉ ngơi. Trẻ sơ sinh có nhịp tim cao hơn nhiều so với người lớn, vào khoảng 120 bpm và 90 bpm khi lớn hơn.

Giống như nhịp tim, nhiệt độ cơ thể bình thường cũng thay đổi ở mỗi người và thay đổi trong ngày. Nhiệt độ cơ thể thấp nhất vào buổi sáng sớm và cao nhất vào đầu buổi tối. Nhiệt độ cơ thể bình thường là khoảng 37° C hoặc 98,6° F. Tuy nhiên, nhiệt độ có thể thấp tới 36,1° C (97° F) vào sáng sớm và cao tới 37,2° C (99° F) và vẫn được xem xét. Bình thường. Do đó, phạm vi bình thường của nhiệt độ cơ thể là 97 đến 100 độ F hoặc 36,1 đến 37,8 độ C. Nhiệt độ có thể được đo bằng cách sử dụng các loại cảm biến khác nhau. Các cảm biến này có nhiều dạng khác nhau như cặp nhiệt điện, nhiệt điện trở, đầu dò nhiệt độ điện trở (RTD) và cảm biến mạch tích hợp (IC).

Đề tài này trình bày một hệ thống với các thiết bị được sử dụng trong các dự án Internet of Things (IoT). IoT đã và đang trở thành một xu hướng toàn cầu khi nó đang phát triển các giải pháp tự động hóa trong gia đình, xử lý, robotics...

Các dự án được phát triển và coi là Internet of Things, sử dụng các công nghệ trong đó một số thiết bị có thể được kết nối và điều khiển từ xa, do đó cho phép kiểm soát và định hướng tốt hơn cho các loại nhu cầu khác nhau của các dự án cụ thể, chẳng hạn như điều khiển TV từ xa, chuẩn bị cà phê, sưởi ấm bồn tắm, kéo rèm, kết nối và điều khiển các chương trình tivi, radio một cách thông minh, ghi nhớ những khoảnh khắc nhất định của mỗi cá nhân.

Có thể tìm thấy nhiều loại giải pháp và thiết bị có sẵn trên thị trường, đáp ứng nhu cầu khác nhau của mỗi cá nhân, vấn đề lớn nhất là chi phí nhập khẩu cao, cũng như việc chuẩn bị nhân sự kỹ thuật và trình độ để phát triển các giải pháp liên quan đến Internet of Things, vì đây là một công nghệ mới nổi, cần có kiến thức về các ngành nghiên cứu khác nhau như: điện tử; hệ thống; tin học; cơ sở dữ liệu, bảo mật, thiết kế, kiến trúc, mạng WiFi và liên lạc vô tuyến.

Những tiến bộ trong công nghệ y tế và nhất là sự phát triển của IoT đang đóng vai trò quan trọng trong việc cải thiện việc chăm sóc sức khỏe, đặc biệt là trong việc theo dõi nhịp tim. Với số lượng người tử vong vì bệnh tim mạch ngày càng tăng, việc sử dụng các thiết bị IoT để giám sát từ xa trở nên cấp thiết hơn bao giờ hết. Hệ thống theo dõi từ xa cho phép chúng ta không chỉ quan sát nhịp tim mà còn nhiều thông số sức khỏe khác một cách liên tục và không cần bệnh nhân phải có mặt tại bệnh viện. Điều này không chỉ giúp tiết kiệm thời gian và chi phí mà còn tạo điều kiện thuận lợi cho việc tiếp cận dịch vụ y tế.

Những thông số như nhịp tim và nhiệt độ cơ thể có thể cung cấp thông tin quan trọng về sức khỏe của bệnh nhân. Các thiết bị IoT cung cấp khả năng theo dõi liên tục và chính xác, giúp phát hiện sớm các biểu hiện không bình thường và can thiệp kịp thời.

Mặc dù có nhiều giải pháp và thiết bị IoT trên thị trường, tuy nhiên, vấn đề về chi phí và chuyên môn kỹ thuật vẫn còn là thách thức lớn. Sự kết hợp của nhiều lĩnh vực như điện tử, tin học, bảo mật và mạng lưới vô tuyến là cần thiết để phát triển các giải pháp IoT trong y tế. Điều này đặt ra yêu cầu cao về kiến thức và kỹ năng chuyên ngành, đồng thời đòi hỏi sự đầu tư lớn về nguồn lực và đào tạo để phát triển và triển khai các giải pháp IoT hiệu quả trong lĩnh vực y tế.

4 Thiết bị

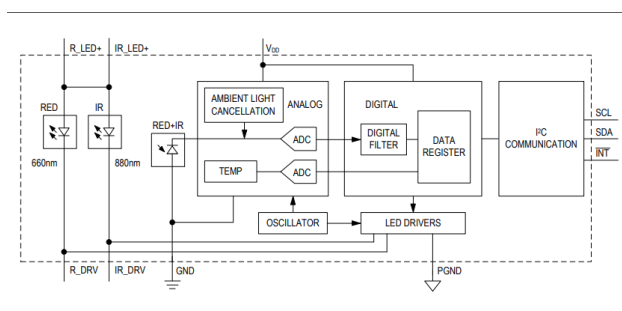
Trong đề tài lần này, chúng tôi sử dụng 2 thiết bị phần cứng. Đầu tiên là cảm biến MAX3010 để đọc dữ liệu đầu vào từ người dùng. Thứ hai là NodeMCU12e, hay còn gọi là ESP8266, là nơi để xử lý và truyền dữ liệu lên server.

4.1 Cảm biến MAX30100, GY-MAX30100 và MAX30102



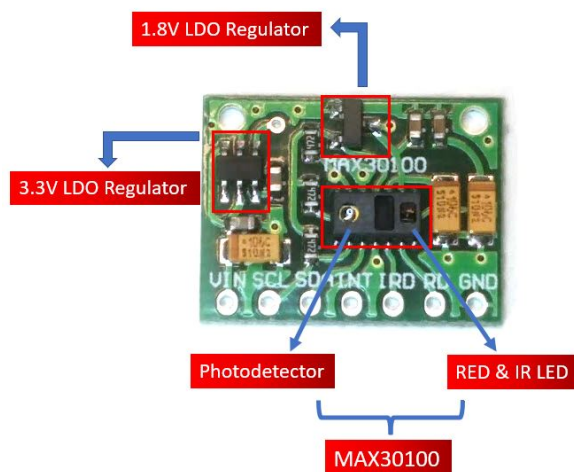
Hình 1: Cảm biến MAX30100

Cảm biến nhịp tim và đo nồng độ oxy MAX30100 là một cảm biến sinh học plug-and-play tiết kiệm năng lượng sử dụng giao thức I2C. Nó có thể được sử dụng bởi nhiều đối tượng, ví dụ như sinh viên, người đam mê, kỹ sư, nhà sản xuất, và các nhà phát triển game và di động muốn tích hợp dữ liệu nhịp tim trực tiếp vào dự án của họ.



Hình 2: Schematic của MAX30100

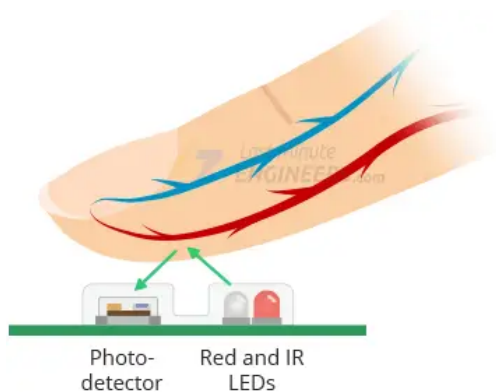
Module này tích hợp MAX30100 – một vi mạch cảm biến nhịp tim và đo nồng độ oxy tiên tiến từ Analog Devices. Nó kết hợp hai đèn LED, một bộ phát hiện ánh sáng, quang học tối ưu hóa, và xử lý tín hiệu analog ít nhiễu để phát hiện tín hiệu đo nhịp tim (SpO2) và nhịp tim (HR).



Hình 3: Tổng quan

MAX30100 có hai đèn LED - một đèn LED màu đỏ và một đèn LED hồng ngoại (IR). Ở phía bên kia là một bộ phát hiện ánh sáng rất nhạy cảm. Ý tưởng là bạn chiếu sáng một đèn LED vào một thời điểm, phát hiện lượng ánh sáng phản xạ lại tại bộ phát hiện, và dựa vào tín hiệu đó bạn có thể đo được mức độ oxy trong máu và nhịp tim.

Bộ cảm biến nhịp tim và đo nồng độ oxy trong máu như MAX30100, hoặc bất kỳ cảm biến nhịp tim quang học nào cũng gồm một cặp đèn LED có cường độ cao (một LED màu ĐỎ và một LED màu hồng ngoại - IR) với hai bước sóng khác nhau và một cảm biến quang. Cụ thể, bước sóng của các đèn LED này lần lượt là 660nm và 880nm

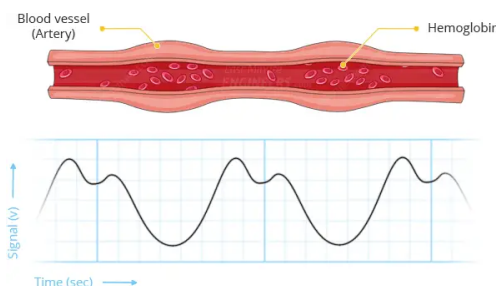


Hình 4: Cấu tạo bộ phát hiện quang đồ (Detection-Photoplethysmogram)

MAX30100 hoạt động bằng cách chiếu cả hai loại ánh sáng lên đầu ngón tay hoặc trên thùy tai (hoặc bất cứ nơi nào da không quá dày, để ánh sáng có thể dễ dàng xuyên qua mô) và đo lượng ánh sáng phản chiếu bằng cảm biến quang. Phương pháp này của việc phát hiện nhịp tim thông qua ánh sáng được gọi là Photoplethysmogram.

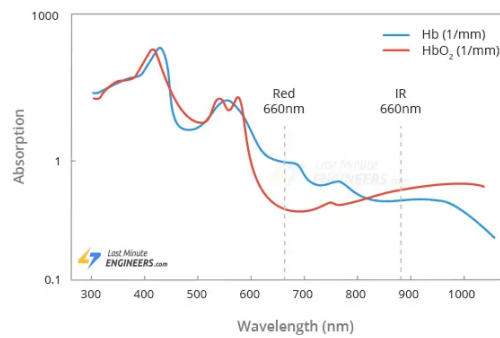
Hoạt động của MAX3010 có thể chia thành hai phần chính: Đo nhịp tim và Oxy huyết quản (đo lượng oxy trong máu).

Hemoglobin oxy hóa (HbO_2) trong máu bên trong động mạch có đặc tính hấp thụ ánh sáng hồng ngoại. Máu càng đỏ (hemoglobin cao) thì càng hấp thụ nhiều ánh sáng hồng ngoại hơn. Khi máu được bơm qua ngón tay cùng với mỗi nhịp tim, lượng ánh sáng phản chiếu thay đổi, tạo ra một biểu đồ sóng biến đổi tại đầu ra của cảm biến quang. Khi tiếp tục chiếu sáng và đọc cảm biến quang, bạn nhanh chóng bắt đầu có thể đọc được nhịp tim (HR).



Hình 5: Cơ chế hoạt động

Tín hiệu sóng hấp thụ ánh sáng đỏ và hồng ngoại được sử dụng trong cảm biến nhịp tim dựa trên nguyên tắc rằng lượng ánh sáng đỏ và hồng ngoại bị hấp thụ sẽ thay đổi tùy thuộc vào lượng oxy trong máu. Đồ thị sau đây là phác đồ hấp thụ của hemoglobin oxy hóa (HbO_2) và hemoglobin không oxy hóa (Hb).



Hình 6: Đồ thị hemoglobin

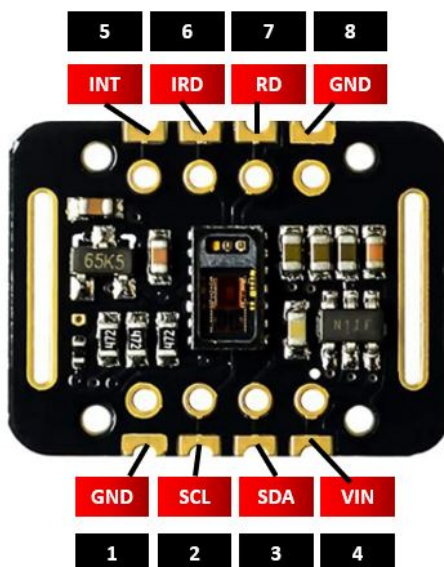
Từ đồ thị, có thể thấy máu không oxy hóa hấp thụ nhiều ánh sáng đỏ hơn (660nm), trong khi máu oxy hóa hấp thụ nhiều ánh sáng hồng ngoại hơn (880nm). Bằng cách đo tỷ lệ ánh sáng hồng ngoại và đỏ nhận được bởi cảm biến quang, ta có thể tính toán được mức độ oxy (SpO_2) trong máu.

Ở đây, chúng tôi sử dụng thêm 1 cảm biến GY-MAX30100 nhằm phục vụ cho mục đích kiểm chứng cũng như so sánh kết quả. Cấu tạo và cơ chế hoạt động của GY-MAX30100 gần như tương tự với MAX30100.



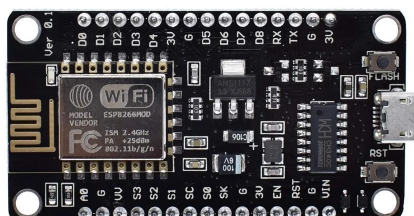
Hình 7: Cảm biến GY-MAX30100

Cảm biến MAX30102 là bản phát triển hơn từ MAX30100. Các chân pin out và cách giao tiếp của nó gần như tương đồng với max30100.



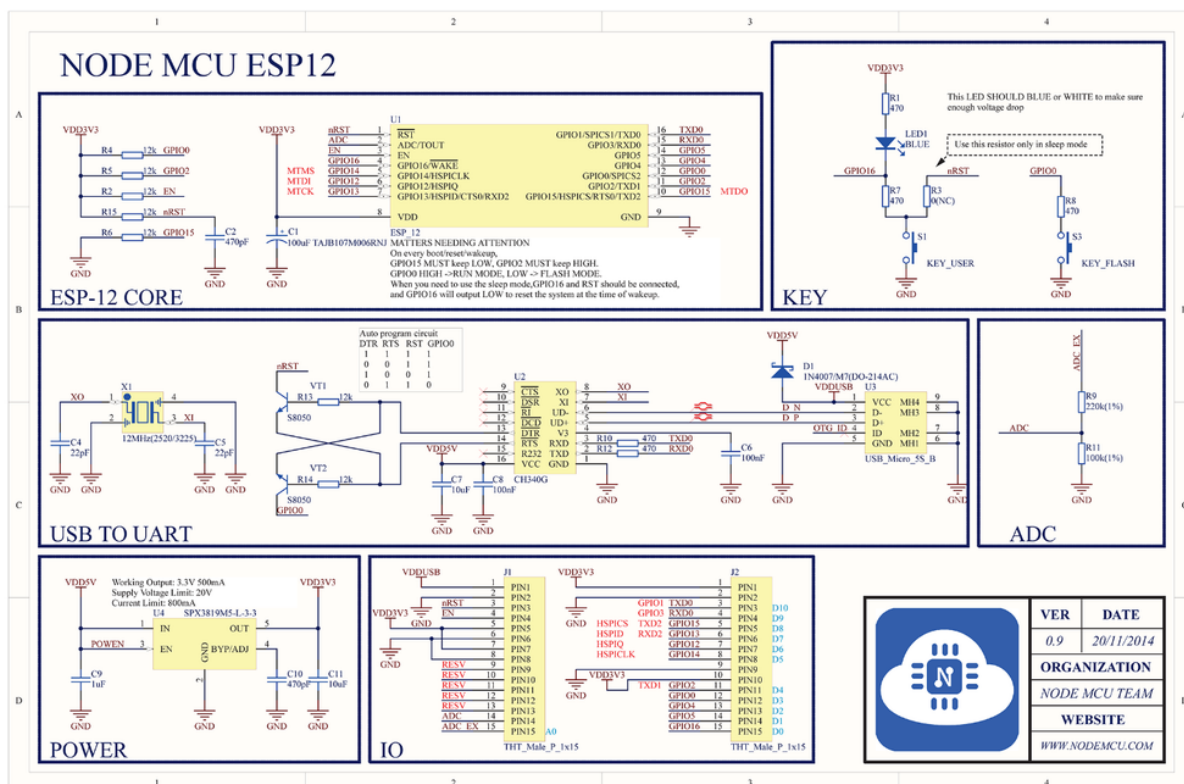
Hình 8: Cảm biến MAX30102

4.2 Module Esp8266

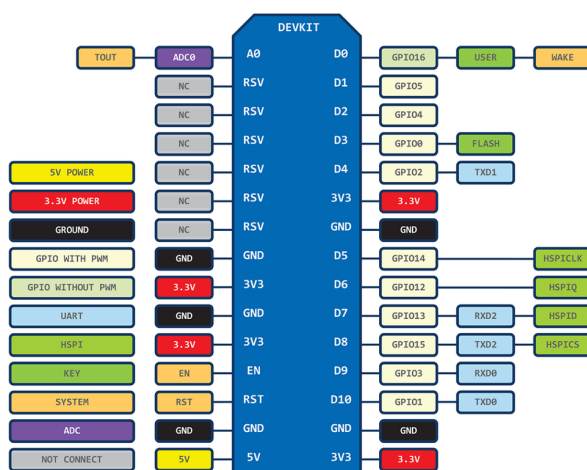


Hình 9: Esp8266 NodeMCU

NodeMCU (Node Micro Controller Unit) là một môi trường phát triển phần cứng và phần mềm mã nguồn mở được xây dựng xung quanh một vi điều khiển hệ thống giá rẻ gọi là ESP8266. ESP8266, được thiết kế và sản xuất bởi Espressif Systems, chứa các thành phần cần thiết của một máy tính: CPU, RAM, mạng (WiFi), và thậm chí là một hệ điều hành hiện đại và bộ công cụ phát triển phần mềm (SDK). Điều này làm cho nó trở thành lựa chọn cho các dự án Internet of Things (IoT).



Hình 10: Esp8266 NodeMCU schematic



Hình 11: Esp8266 NodeMCU pin out

ESP8266 có thể được sử dụng như một mô-đun Wifi bên ngoài bằng cách sử dụng Firmware chuẩn với bộ lệnh AT, bằng cách kết nối nó với bất kỳ vi điều khiển nào sử dụng giao tiếp serial UART, hoặc có thể trực tiếp hoạt động như một vi điều khiển có khả năng kết nối Wifi, bằng cách lập trình một Firmware mới sử dụng SDK được cung cấp.

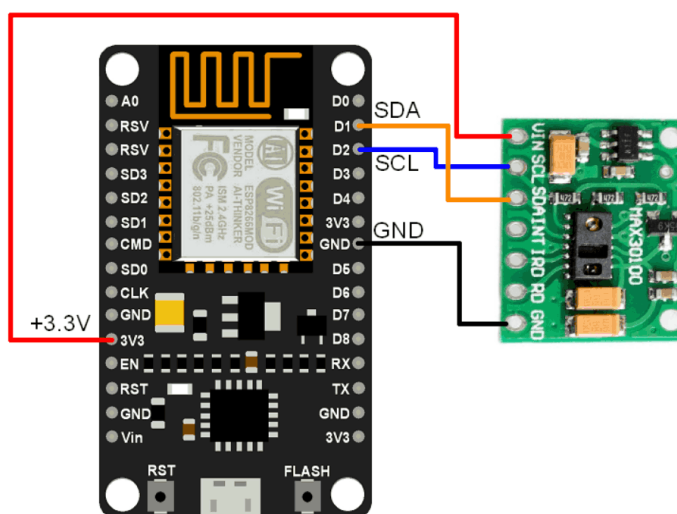
Các chân GPIO của ESP8266 cho phép IO Analog và Digital, cũng như PWM, SPI, I2C, v.v. Điều này cho phép nó tham gia vào nhiều hoạt động ngoại vi như giao tiếp số, giao tiếp tương tự, và các hoạt động điều khiển tín hiệu đầu vào/ra.

4.3 Kết nối thiết bị

Có thể kết nối giữa cảm biến MAX30100 và ESP8266 như sau:

MAX30100 Module	ESP8266
VCC	3.3V
SCL	D1
SDA	D2
GND	GND

Để kết nối các chân I2C mặc định của ESP8266 với chân SCL và SDA của mô-đun cảm biến, chúng ta có thể thực hiện như sau:



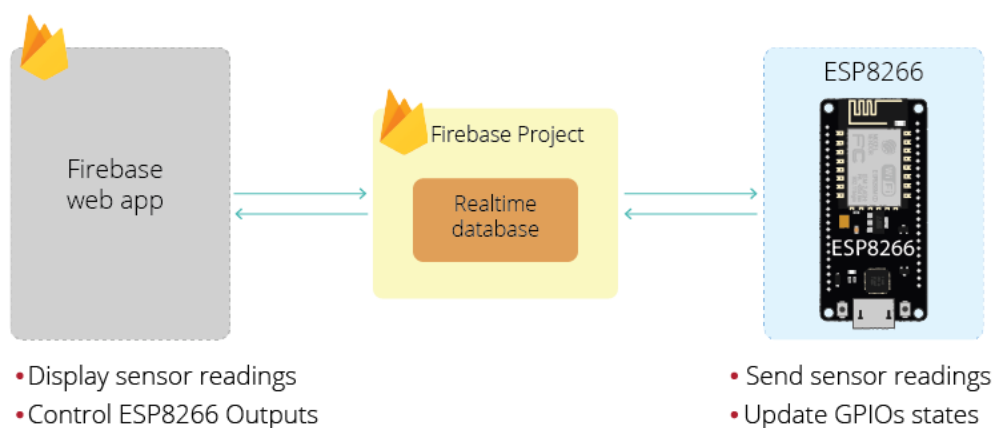
Hình 12: Kết nối ESP8266 và MAX30100

5 Database và ngôn ngữ sử dụng

5.1 Firebase

Firebase là nền tảng phát triển ứng dụng di động của Google giúp xây dựng, cải thiện và phát triển ứng dụng của bạn. Nó bao gồm nhiều dịch vụ quản lý dữ liệu từ bất kỳ ứng dụng nào trên Android, iOS hoặc web.

Chúng ta sẽ tạo một dự án Firebase với một cơ sở dữ liệu thời gian thực, và chúng ta sẽ sử dụng ESP8266 để lưu trữ và đọc dữ liệu từ cơ sở dữ liệu đó. ESP8266 có thể tương tác với cơ sở dữ liệu từ bất kỳ nơi nào trên thế giới miễn là nó được kết nối với internet.



Hình 13: Kết nối giữa firebase và Esp8266

5.2 HTML, CSS, JavaScript

5.2.1 HTML

HTML (HyperText Markup Language) là loại ngôn ngữ đánh dấu siêu văn bản được sử dụng trong việc phát triển các trang web.

HTML không phải là ngôn ngữ lập trình. Nó thực chất chỉ là một cách xác định cấu trúc, ý nghĩa và mục đích của các phân tử trên trang web, như các phần tử văn bản, hình ảnh, liên kết,...bằng cách sử dụng các thẻ. Ví dụ: nếu bạn muốn tạo một đoạn văn bản, chỉ cần sử dụng thẻ <p>, nếu bạn muốn tạo tiêu đề cho trang web sử dụng ngay thẻ <h1>.

Nhiệm vụ chính của HTML xây dựng cấu trúc siêu văn bản trên một trang web và khai báo các tập tin kỹ thuật số như hình ảnh, video, âm thanh. Ngoài ra, HTML còn có các công dụng quan trọng khác, bao gồm:

- Phân chia và định dạng nội dung: chia đoạn, định dạng và nhấn mạnh văn bản, đặt tiêu đề, ngắt dòng, tạo danh sách, và tạo ký tự đặc biệt.
- Tạo liên kết và chèn hình ảnh: bao gồm liên kết nội bộ và liên kết ngoài web, ảnh minh họa,... thu hút người dùng
- Tạo các kiểu mẫu: gồm cách hiển thị các đoạn văn bản, hình ảnh, bảng,... giúp cho trang web đồng nhất và dễ dàng quản lý hơn.
- Xác định các thuộc tính của trang web: bao gồm màu sắc, kích thước, vị trí và các thông tin khác.

5.2.2 CSS

CSS (Cascading Style Sheets) là một ngôn ngữ thiết kế được sử dụng trong thiết kế trang web để mô tả cách trình bày, bố cục và phong cách của nội dung trên trang web. CSS giúp tạo ra các trang web chuyên nghiệp và dễ đọc hơn. Lưu ý rằng, CSS không phải ngôn ngữ lập trình.

Nếu ví HTML là bức tường thì CSS giống như lớp sơn trang trí. HTML cung cấp các công cụ cơ bản để cấu trúc nội dung trên trang web còn CSS giúp định dạng (style) nội dung này để nó được hiển thị cho người dùng theo đúng ý định thiết kế. Ba phong cách chính của CSS gồm:

- **Internal CSS:** Là phong cách được đặt trực tiếp trong thẻ <head> của trang web, cho phép chỉnh sửa các phong cách của trang web nhanh chóng và dễ dàng. Tuy nhiên, nó chỉ ảnh hưởng đến trang web đó, không áp dụng cho các trang khác.

- **Inline CSS:** Bạn có thể thêm phong cách trực tiếp vào mỗi phần tử HTML trên trang web bằng thuộc tính “style”, cho phép dễ dàng chỉnh sửa phong cách của phần tử mà không cần truy cập trực tiếp vào file CSS.
- **External CSS:** Cho phép bạn tạo file CSS riêng biệt và áp dụng nó cho nhiều trang web khác nhau. Bạn có thể chỉnh sửa phong cách trong file CSS để thay đổi toàn bộ trang web, qua đó bạn giúp cải thiện tốc độ tải trang web, giảm thiểu thời gian và công sức để thay đổi phong cách của trang web.

Vai trò của CSS trong lập trình web có thể kể đến như:

- Xác định phong cách và định nghĩa nội dung dưới dạng văn bản HTML trên trang web.
- Tạo ra trang web tương thích với nhiều thiết bị khác nhau.
- Tạo ra bố cục, hiệu ứng và sự độc đáo để thu hút, tăng tính tương tác và trải nghiệm của người dùng.
- Điều khiển định dạng và tái sử dụng các kiểu dáng và phong cách giúp tiết kiệm thời gian và công sức.
- Giúp giảm thiểu lượng mã nguồn và tăng tốc độ tải trang.

5.2.3 JavaScript

JavaScript là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay. Nó được tạo ra bởi Brendan Eich (1995) và trở thành một phần quan trọng của các trang web hiện đại. JavaScript giúp biến những trang web tĩnh trở nên động, tạo ra sự tương tác và cải thiện hiệu suất của máy chủ. Nó làm cho các tính năng như Slideshow, Pop-up quảng cáo, cùng tính năng Autocomplete của Google trở nên dễ dàng và thuận tiện hơn.

Đồng thời, sự kết hợp giữa JavaScript, HTML và CSS giúp tăng cường trải nghiệm người dùng và cải thiện giao diện của trang web.

Vai trò của JavaScript gồm:

- Tạo bố cục chuyển động với thư viện ảnh và hiệu ứng động.
- Xây dựng các trò chơi, hoạt hình 2D hoặc 3D, và các ứng dụng cơ sở dữ liệu phức tạp để tăng tính tương tác trên trang web.
- Tăng cường hành vi người dùng bằng các tính năng như xác nhận hộp thoại, các lời kêu gọi hành động, và các tính năng tương tác khác.
- Kiểm soát mặc định của trình duyệt bằng cách thay đổi các hành vi mặc định của trình duyệt, chẳng hạn như xử lý các biểu mẫu đăng ký và đăng nhập của người dùng.
- Hỗ trợ việc xử lý dữ liệu phía máy khách (client-side), giảm tải cho máy chủ và tăng tốc độ xử lý trang.

5.3 C++ và Arduino IDE

Để giao tiếp với ESP8266, chúng ta cần có 1 IDE và 1 ngôn ngữ lập trình. Ở đây chúng tôi lựa chọn Arduino IDE. Đây là một IDE thông dụng, miễn phí, mã nguồn mở dùng để giao tiếp với các thiết bị arduino. Arduino IDE cung cấp cho người dùng một giao diện trực quan để sử dụng, tải lên mã nguồn cho arduino và cả ESP8266.

Ngôn ngữ C++ được dùng để giao tiếp với ESP bởi đây là một ngôn ngữ cấp thấp, cho phép kiểm soát chặt chẽ ESP8266. Điều này là cần thiết với các thiết bị hiệu suất cao như ESP8266.

6 Mã nguồn của ESP8266

6.1 Tiền xử lý dữ liệu

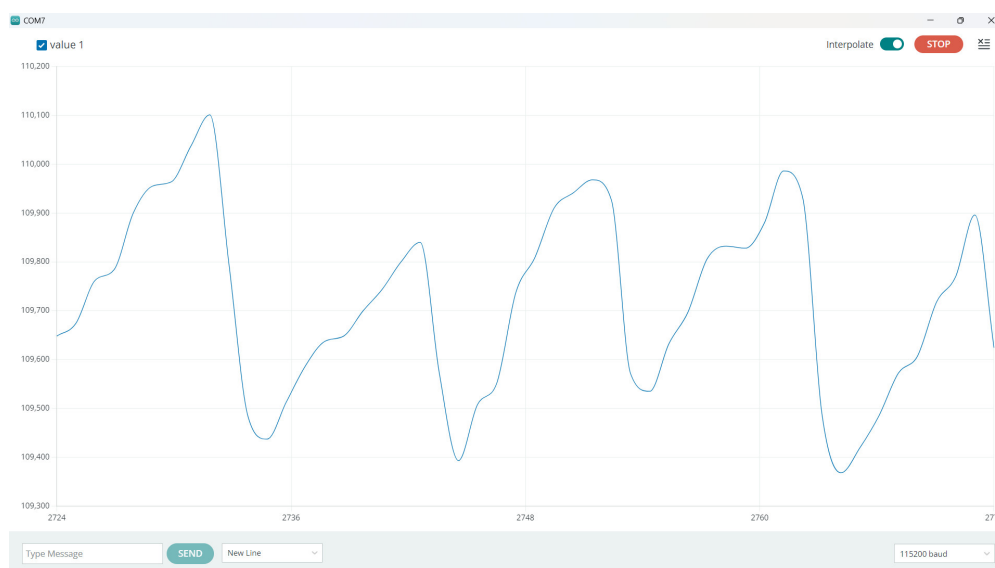
Việc tín hiệu bị nhiễu là một trong những điều cần phải kiểm tra trước khi tiến hành tính toán với các dạng tín hiệu analog. Đối với leds IR, việc nhiễu có thể xảy ra khi đèn led sáng liên tục. Có một cách giải quyết khi bị nhiễu là ta cho đèn led nhấp nháy với một tần số lớn hơn một lần tìm đập.

Đầu tiên, chúng ta cần kiểm tra tín hiệu đầu vào có bị nhiễu hay không, cụ thể ở đây là tín hiệu IR. Chúng ta có chương trình sau để kiểm tra tín hiệu:

```
1 #include <Wire.h>
2 #include "MAX30105.h"
3
4 MAX30105 particleSensor;
5
6 #define debug Serial //Uncomment this line if you're using an Uno or ESP
7 //#define debug SerialUSB //Uncomment this line if you're using a SAMD21
8
9 void setup()
10 {
11     debug.begin(9600);
12     debug.println("MAX30105 Basic Readings Example");
13
14     // Initialize sensor
15     if (particleSensor.begin() == false)
16     {
17         while (1);
18     }
19
20     particleSensor.setup(); //Configure sensor. Use 6.4mA for LED drive
21 }
22
23 void loop()
24 {
25     debug.println(particleSensor.getIR());
26 }
```

Listing 1: Kiểm tra tín hiệu IR

Sau đó, chúng ta sử dụng Serial plotter của Arduino IDE để kiểm tra kết quả tín hiệu:



Hình 14: IR plotter

Tín hiệu chúng ta nhận được khá ổn định, một số điểm nhiễu không ảnh hưởng tới việc xử lý của thuật toán tính toán nhịp tim. Đây được coi là dữ liệu khả dụng và không cần phải tiến hành khử nhiễu.

6.2 Xử lý dữ liệu

Ở đây, chúng tôi gặp một vài khó khăn về giải thuật tính toán Nồng độ oxy trong máu SpO_2 . Vì vậy ở đề tài lần này, chúng tôi xin được thực hiện trên 2 cảm biến, một cảm biến để đọc nhịp tim và một cảm biến để đọc nồng độ oxy trong máu.

6.2.1 Đo nồng độ oxy trong máu với MAX30100 và GY-MAX30100

Ở đây, chúng tôi dùng thư viện Arduino-max30100. Đây là một thư viện cho phép giao tiếp với max30100, lấy dữ liệu và tính toán nhịp tim cũng như nồng độ oxy trong máu dựa vào dữ liệu thu thập được.

Tiếp theo là thư viện Firebase-ESP8266. Thư viện này cho phép kết nối giữa ESP và Firebase, điều này cho phép nhận và gửi dữ liệu giữa 2 đầu.

Một số thư viện được sử dụng như thư viện Arduino, Wire hay ESP8266WiFi đều là những thư viện cơ bản được tích hợp sẵn ở trong arduino IDE, nhằm giúp chúng ta dễ dàng lập trình ESP8266.

```
1 #include <Arduino.h>
2 #include <ESP8266WiFi.h>
3 #include <FirebaseESP8266.h>
4 #include <Wire.h>
5 #include "MAX30100_PulseOximeter.h"
6 #include "addons/TokenHelper.h" //Provide the token generation process info.
7 #include "addons/RTDBHelper.h" //Provide the RTDB payload printing info and other helper functions.
```

Listing 2: Library

Sau đó chúng ta tiến hành định nghĩa những hằng số như tên và mật khẩu WiFi, API key và URL của Firebase Realtime Database, Authentication cũng như khởi tạo các biến của Firebase và MAX30100.

```
1 #define REPORTING_PERIOD_MS    1000
2 // Insert your network credentials
3 #define WIFI_SSID "FPT phong 12"
4 #define WIFI_PASSWORD "matkhau1a"
5 // Insert Firebase project API Key
```

```
6 #define API_KEY "AIzaSyBZUvbeKXJlo_5RJqKJEV4b0GqgnE3FCY"
7 // Insert RTDB URLe fine the RTDB URL */
8 #define DATABASE_URL "https://doantkl1-73b19-default-rtdb.firebaseio.com/" // Insert RTDB
    URLe fine the RTDB URL */
9 // Define user authentication
10 #define USER_EMAIL "ttdat170703@gmail.com" // Define user authentication
11 #define USER_PASSWORD "admin@"
12
13 // PulseOximeter is the higher level interface to the sensor
14 // it offers:
15 // * beat detection reporting
16 // * heart rate calculation
17 // * SpO2 (oxidation level) calculation
18 PulseOximeter pox;
19
20 //Define Firebase Data object
21 FirebaseData fbdo;
22 FirebaseAuth auth;
23 FirebaseConfig config;
24
25 unsigned long sendDataPrevMillis = 0;
26 bool signupOK = false; //since we are doing an anonymous sign in
27
28 String uid;
29 String databasePath;
30
31 uint32_t tsLastReport = 0;
32
33 float beat_send = 0;
34 uint8_t spo2_send = 0;
35
36 // Callback (registered below) fired when a pulse is detected
37 void onBeatDetected(){
38     Serial.println("Beat!");
39 }
```

Listing 3: Define anything

Tiếp theo, trong hàm setup chúng ta sẽ tiến hành connect với WiFi, tiếp đó là tiến hành cấu hình và kết nối với Firebase cũng như cảm biến MAX30100

```
1 void setup()
2 {
3     Serial.begin(115200);
4     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
5     Serial.print("Connecting to Wi-Fi");
6     while (WiFi.status() != WL_CONNECTED) {
7         Serial.print(".");
8         delay(300);
9     }
10    Serial.println();
11    Serial.print("Connected with IP: ");
12    Serial.println(WiFi.localIP());
13    Serial.println();
14
15    Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
16    /* Assign the api key (required) */
17    config.api_key = API_KEY;
18
19    /* Assign the user sign in credentials */
20    auth.user.email = USER_EMAIL;
```



```
21 auth.user.password = USER_PASSWORD;
22
23 /* Assign the RTDB URL (required) */
24 config.database_url = DATABASE_URL;
25
26 /* Assign the callback function for the long running token generation task */
27 config.token_status_callback = tokenStatusCallback; // see addons/TokenHelper.h
28
29 // Comment or pass false value when WiFi reconnection will control by your code or third
    party library e.g. WiFiManager
30 Firebase.reconnectNetwork(true);
31
32 Firebase.begin( & config, & auth);
33
34 Firebase.setDoubleDigits(5);
35 Serial.print("Initializing pulse oximeter..");
36 // Initialize the PulseOximeter instance
37 // Failures are generally due to an improper I2C wiring, missing power supply
38 // or wrong target chip
39 if (!pox.begin()) {
40     Serial.println("FAILED");
41     for(;;);
42 } else {
43     Serial.println("SUCCESS");
44 }
45 // The default current for the IR LED is 50mA and it could be changed
46 // by uncommenting the following line. Check MAX30100_Registers.h for all the
47 // available options.
48 // pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
49 // Register a callback for the beat detection
50 pox.setOnBeatDetectedCallback(onBeatDetected);
51 }
```

Listing 4: setup Function

Cuối cùng, trong hàm loop chúng ta tiến hành các việc theo thứ tự bao gồm:

- lấy dữ liệu
- tính toán nhịp tim và nồng độ oxy trong máu
- lấy userId hiện tại
- Dừng cảm biến để gửi dữ liệu
- Gửi Nhịp tim và nồng độ oxy trong máu lên firebase
- Tiếp tục đọc dữ liệu
- Lưu thời gian hiện tại, sau một khoảng thời gian tính từ thời điểm hiện tại thì lại tiếp tục đọc và tính toán dữ liệu.

```
1 void loop()
2 {
3     // Make sure to call update as fast as possible
4     pox.update();
5     if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
6         uid = auth.token.uid.c_str();
7         beat_send = pox.getHeartRate();
8         Serial.print("Heart rate:");
9         Serial.print(pox.getHeartRate());
```

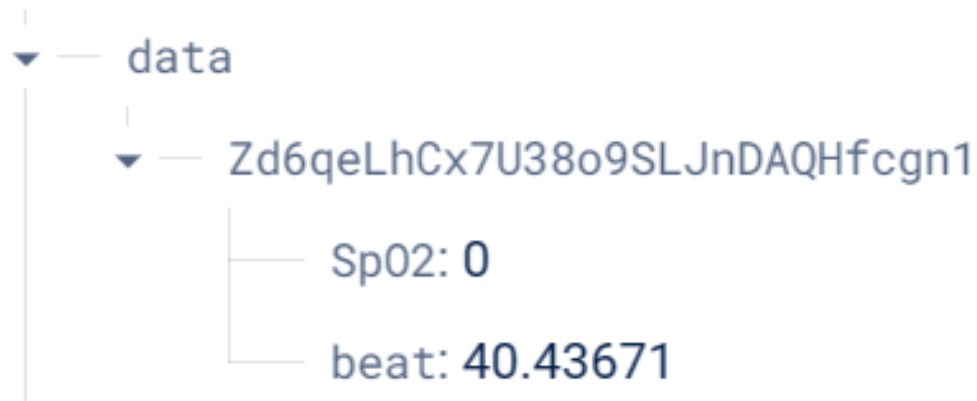
```

10     spo2_send = pox.getSpO2();
11     Serial.print("bpm / SpO2:");
12     Serial.print(pox.getSpO2());
13     Serial.println("%");
14     pox.shutdown();
15     databasePath = "/data/" + uid + "/beat";
16     Firebase.setFloat(fbdo, databasePath, beat_send);
17     databasePath = "/data/" + uid + "/SpO2";
18     Firebase.setFloat(fbdo, databasePath, spo2_send);
19     pox.resume();
20
21     tsLastReport = millis();
22 }
23 }

```

Listing 5: setup Function

Kết quả là ta có được dữ liệu lưu trên Realtime Database của Firebase như sau:



Hình 15: Data trên Firebase

Từ đây, chúng ta có thể lấy dữ liệu để hiển thị lên Web và App cho người dùng thấy kết quả.

6.2.2 Đo nhịp tim với MAX30102

Chúng tôi sử dụng thư viện MAX30105 và heartrate của SparkFun. Thư viện này cho phép chúng ta cấu hình cảm biến MAX30102, sử dụng để đọc tín hiệu IR và tính toán dựa theo đồ thị IR có được. Khoảng cách giữa 2 đỉnh của đồ thị là một lần tim đập, lấy thời gian một phút chia cho khoảng thời gian đó sẽ ra được nhịp tim trung bình. Sau khi có được giá trị trung bình thì tiến hành gửi kết quả lên FireBase để hiển thị lên Web và App. Dưới đây là source code của chương trình đo nhịp tim.

```

1
2 #include <Arduino.h>
3
4 #include <ESP8266WiFi.h>
5
6 #include <FirebaseESP8266.h>
7
8 #include <Wire.h>
9
10 #include "MAX30105.h"
11
12 #include "heartRate.h"
13
14 //Provide the token generation process info.
15 #include "addons/TokenHelper.h"

```

```
16 //Provide the RTDB payload printing info and other helper functions.
17 #include "addons/RTDBHelper.h"
18
19 // Insert your network credentials
20 #define WIFI_SSID "FPT phong 12"
21 #define WIFI_PASSWORD "matkhaula1"
22
23 // Insert Firebase project API Key
24 #define API_KEY "AIzaSyBZUvbeKXJlo_5RJqKJEV4b0GqgnE3FCY"
25
26 // Insert RTDB URLdefine the RTDB URL */
27 #define DATABASE_URL "https://doantkl1-73b19-default-rtdb.firebaseio.com/"
28
29 // Define user authentication
30 #define USER_EMAIL "ttdat170703@gmail.com"
31 #define USER_PASSWORD "admin@"
32
33 //Define Firebase Data object
34 FirebaseData fbdo;
35 FirebaseAuth auth;
36 FirebaseConfig config;
37
38 unsigned long sendDataPrevMillis = 0;
39 bool signupOK = false; //since we are doing an anonymous sign in
40
41 const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
42 byte rates[RATE_SIZE]; //Array of heart rates
43 byte rateSpot = 0;
44 long lastBeat = 0; //Time at which the last beat occurred
45 MAX30105 particleSensor;
46
47 float beatsPerMinute;
48 int beatAvg;
49 String uid;
50 String databasePath;
51
52 void setup() {
53   Serial.begin(115200);
54   // Initialize sensor
55   if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
56   {
57     Serial.println("MAX30105 was not found. Please check wiring/power. ");
58     while (1);
59   }
60   Serial.println("Place your index finger on the sensor with steady pressure.");
61
62   particleSensor.setup(); //Configure sensor with default settings
63   particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is
        running
64   particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
65
66   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
67   Serial.print("Connecting to Wi-Fi");
68   while (WiFi.status() != WL_CONNECTED) {
69     Serial.print(".");
70     delay(300);
71   }
72   Serial.println();
73   Serial.print("Connected with IP: ");
74   Serial.println(WiFi.localIP());
```

```
75 Serial.println();
76 Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
77
78 /* Assign the api key (required) */
79 config.api_key = API_KEY;
80
81 /* Assign the user sign in credentials */
82 auth.user.email = USER_EMAIL;
83 auth.user.password = USER_PASSWORD;
84
85 /* Assign the RTDB URL (required) */
86 config.database_url = DATABASE_URL;
87
88 /* Assign the callback function for the long running token generation task */
89 config.token_status_callback = tokenStatusCallback; // see addons/TokenHelper.h
90
91 // Comment or pass false value when WiFi reconnection will control by your code or third
  party library e.g. WiFiManager
92 Firebase.reconnectNetwork(true);
93
94 Firebase.begin( & config, & auth);
95
96 Firebase.setDoubleDigits(5);
97
98 }
99
100 void loop() {
101   long irValue = particleSensor.getIR();
102   beatsPerMinute = 0;
103   if (checkForBeat(irValue) == true) {
104     //We sensed a beat!
105     long delta = millis() - lastBeat;
106     lastBeat = millis();
107
108     beatsPerMinute = 60 / (delta / 1000.0);
109
110     if (beatsPerMinute < 255 && beatsPerMinute > 20) {
111       rates[rateSpot++] = (byte) beatsPerMinute; //Store this reading in the array
112       rateSpot %= RATE_SIZE; //Wrap variable
113
114       //Take average of readings
115       beatAvg = 0;
116       for (byte x = 0; x < RATE_SIZE; x++)
117         beatAvg += rates[x];
118       beatAvg/=RATE_SIZE;
119     }
120     if (Firebase.ready() && (millis() - sendDataPrevMillis > 100 || sendDataPrevMillis == 0)
121     ) {
122       sendDataPrevMillis = millis();
123
124       Serial.println(beatsPerMinute);
125
126       uid = auth.token.uid.c_str();
127       databasePath = "/data/" + uid + "/beat";
128       delay(0);
129       if (Firebase.setFloat(fbdo, databasePath, beatAvg)) {
130         // This command will be executed even if you dont serial print but we will make sure
131         // its working
132         Serial.print("beat : ");
133         Serial.println(beatAvg);
134       }
135     }
136   }
137 }
```

```
132     } else {  
133         Serial.println("Failed to Read from the Sensor");  
134         Serial.println("REASON: " + fbdo.errorReason());  
135     }  
136 }  
137 }  
138 //  
139 }
```

Listing 6: Chương trình đo nhịp tim bằng MAX30102

7 Web hiển thị

7.1 Các chức năng chính

7.1.1 Hiển thị kết quả đo

Đầu tiên là lấy giá trị nhịp tim và nồng độ oxy trong máu từ database xuống. Sau đó 2 giá trị sẽ được hiển thị trên trang web, mỗi khi giá trị trên database thay đổi thì giá trị trên web cũng thay đổi theo.

Đồng thời ta tiến hành lấy trung bình của 2 giá trị trên, do biên độ dao động của 2 giá trị này khi đo có thể quá cao hoặc quá thấp nên ta cần 1 giá trị đại diện tương đối ứng với từng loại giá trị. Nhóm em chọn giá trị trung bình. Mỗi lần nhịp tim hoặc nồng độ oxy có sự thay đổi trên database, ta tiến hành cập nhật lại, đồng thời hiển thị giá trị trung bình lên web.

Sau đây là minh họa cho việc hiển thị kết quả đo:

Beat:	62	BPM	SpO2:	94	%
Average Beat:	63.50	BPM	Average SpO2:	95.25	%

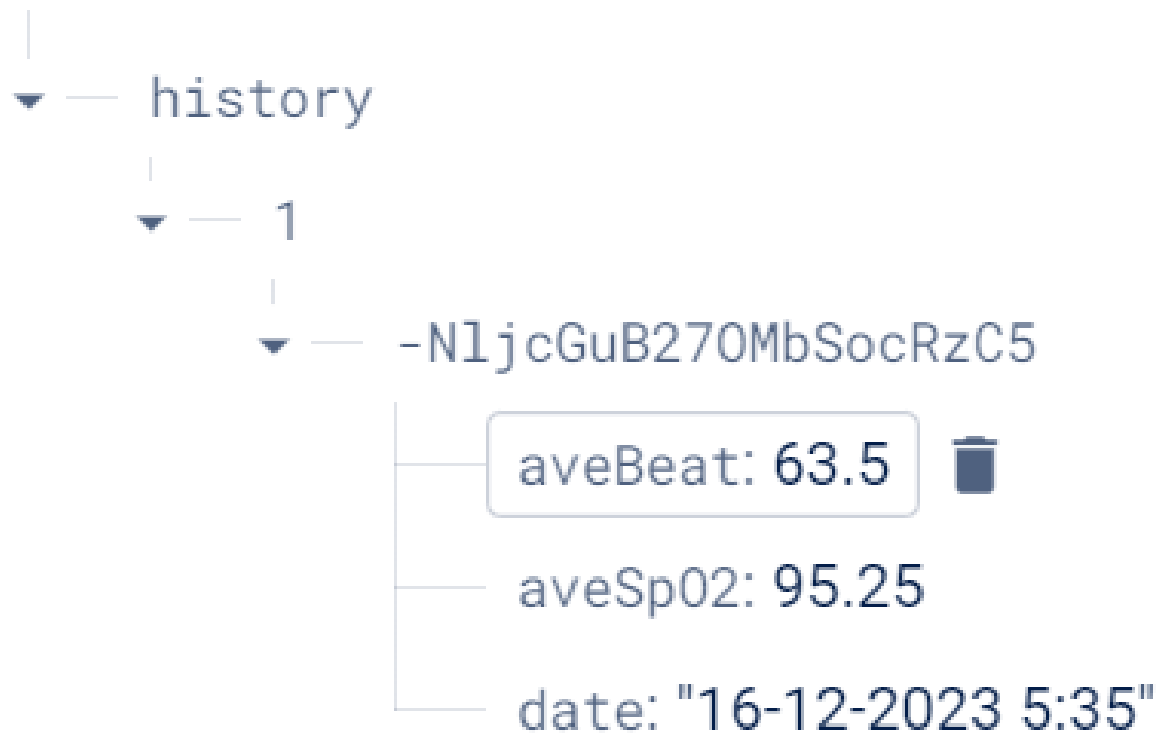
Hình 16: Kết quả đo hiện trên web

Kết quả đo cũng được thể hiện rõ ràng, Beat và SpO2 là 2 giá trị ta đọc từ database, còn Average Beat và Average SpO2 là giá trị trung bình được tính từ các dữ liệu của Beat và SpO2.

7.1.2 Hiển thị lịch sử đo

Nhóm em chỉ lưu 2 lần đo gần nhất. Sau mỗi lần đo và bấm nút Save, dữ liệu sau khi được gửi lên database.

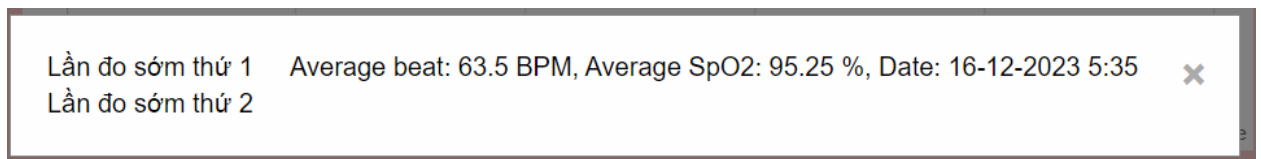
Minh họa lịch sử đo lưu trên database:



Hình 17: Lịch sử đo được lưu trên database

Ta sẽ lưu 3 giá trị: Nhịp tim trung bình, nồng độ oxy trung bình và thời gian đo.

Muốn xem lịch sử đo trên web thì ta bấm nút History và nó sẽ hiển lên 1 cửa sổ pop-up chứa thông tin về 2 lần đo gần nhất.



Hình 18: Lịch sử đo hiện trên web

7.1.3 Hiển thị đồ thị cho nhịp tim

Đồ thị sẽ giúp ta theo dõi sự thay đổi của giá trị và quan sát toàn cục dữ liệu.



Hình 19: Đồ thị

7.1.4 Đọc và ghi kết quả đo từ database

Em dùng nút Save để điều khiển việc đọc hay ghi dữ liệu. Lần đầu vào web nút Save đang ở chế độ ghi, nhưng do lúc này chưa có dữ liệu(số dữ liệu bằng 0) nên việc dữ liệu là không được phép.

Bấm vào 1 lần, ta sẽ thấy nút Save có màu đỏ thẫm, đại diện cho việc đọc dữ liệu.



Hình 20: Đọc dữ liệu từ database

Bấm nút Save 1 lần nữa, lúc này nút có màu xanh, do đã có dữ liệu nên có thể tiến hành đưa dữ liệu lên database.



Hình 21: Gửi dữ liệu lên database

7.1.5 Xóa lịch sử đo

Khi bấm nút Clear, lịch sử đo sẽ được xóa khỏi database và 2 giá trị trung bình trên web cũng vậy.

Beat:	62	BPM	SpO2:	94	%
Average Beat:		BPM	Average SpO2:		%

Hình 22: Sau khi bấm nút Clear

8 Ứng dụng hiển thị kết quả đo

8.1 Giới thiệu

8.1.1 Giới thiệu về Kodular

Kodular là một nền tảng phát triển ứng dụng độc đáo và mạnh mẽ, cho phép người dùng tạo ra các ứng dụng di động mà không cần phải có kiến thức sâu về lập trình. Được thiết kế để thân thiện với người mới bắt đầu, Kodular cung cấp một giao diện kéo và thả dễ sử dụng, nơi người dùng có thể xây dựng ứng dụng của mình bằng cách kết hợp các thành phần khác nhau một cách trực quan. Nền tảng này hỗ trợ nhiều tính

năng và chức năng khác nhau, từ giao diện người dùng đơn giản đến tích hợp cơ sở dữ liệu phức tạp và các dịch vụ nền tảng đám mây. Điểm mạnh của Kodular là khả năng tùy chỉnh cao và khả năng truy cập vào nhiều API và dịch vụ bên thứ ba, cho phép người dùng tạo ra các ứng dụng di động chất lượng cao và chức năng đa dạng mà không cần phải viết code truyền thống. Kodular trở thành công cụ lý tưởng cho những người muốn bước chân vào lĩnh vực phát triển ứng dụng di động, từ học sinh, sinh viên cho đến các chuyên gia không chuyên về lập trình.

8.1.2 Mục đích

Trong thế giới hiện đại, khi mà vấn đề sức khỏe được nâng lên hàng đầu, việc theo dõi nhịp tim đã trở thành một phần quan trọng trong việc duy trì lối sống lành mạnh. Với lượng lớn nhu cầu sử dụng, nhóm em đã triển khai một ứng dụng mang tên **Pulse Check**, được phát triển trên nền tảng **Kodular**, một công cụ mạnh mẽ cho phép tạo ra ứng dụng di động mà không yêu cầu kiến thức sâu về lập trình. Mục đích chính là cung cấp một công cụ đơn giản nhưng hiệu quả để người dùng có thể theo dõi nhịp tim của mình một cách liên tục và chính xác. Ứng dụng này giúp người dùng nhận biết sớm các vấn đề sức khỏe tiềm ẩn và hỗ trợ trong việc quản lý các tình trạng sức khỏe liên quan đến tim mạch.

8.1.3 Phạm vi sử dụng

Pulse Check được thiết kế để phục vụ nhu cầu của một loạt đối tượng người dùng, bao gồm cả những người không có nhiều kiến thức về công nghệ. Điều này được thực hiện thông qua việc sử dụng Kodular, cho phép tạo ra một giao diện người dùng thân thiện và dễ hiểu. Ứng dụng này không chỉ hữu ích cho những người yêu thích thể thao muốn theo dõi nhịp tim trong quá trình luyện tập, mà còn hỗ trợ những người có các vấn đề sức khỏe cần theo dõi nhịp tim định kỳ và chặt chẽ. Với khả năng tích hợp với các thiết bị đo nhịp tim thông minh, **Pulse Check** mang lại một giải pháp đo nhịp tim toàn diện và tiện lợi cho mọi lứa tuổi.

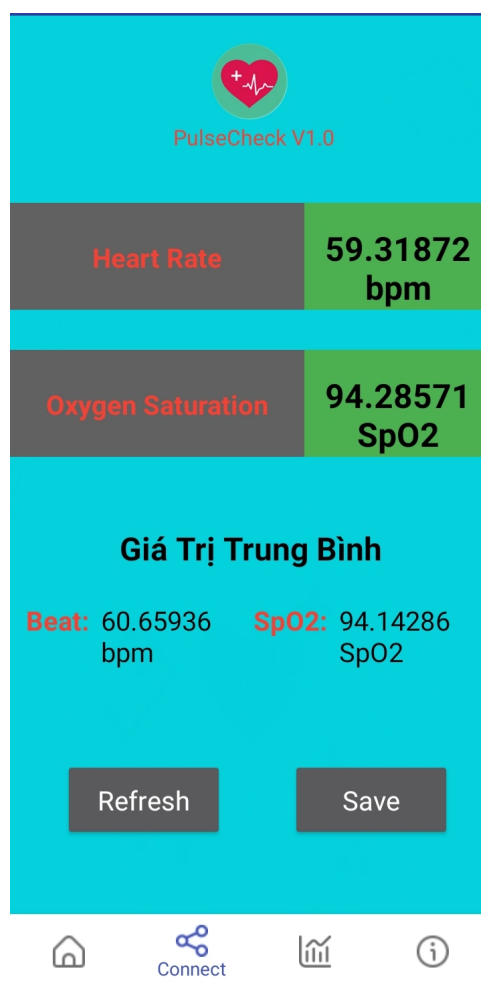
8.2 Các chức năng chính

8.2.1 Hiển thị kết quả đo:

Mục đích giao diện này để hiển thị chỉ số lấy được từ thiết bị đo. Các thành phần chính bao gồm:

- **Heart Rate:** Hiển thị chỉ số nhịp tim hiện tại của người dùng (beats per minute - bpm).
- **Oxygen Saturation:** Hiển thị chỉ số bão hòa oxy trong máu (SpO2).
- **Beat:** Hiển thị nhịp tim trung bình của người dùng theo thời gian.
- **SpO2:** Hiển thị lượng oxy trung bình trong máu của người dùng theo thời gian.
- **Refresh:** Nút bấm có chức năng làm mới dữ liệu khi bắt đầu đo hoặc sau khi đo xong.
- **Save:** Nút bấm có chức năng lưu trữ dữ liệu vào lịch sử đo.

Kết quả ta có một giao diện hoàn chỉnh như sau:

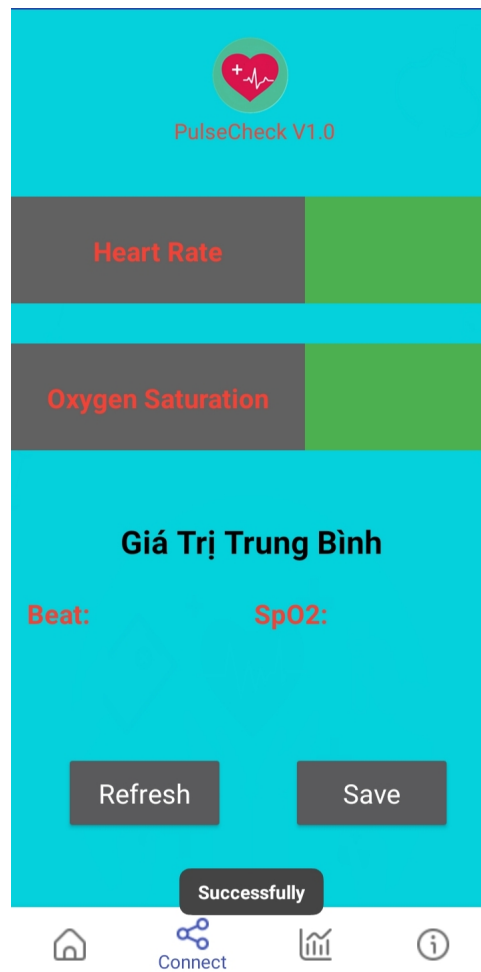


Hình 23: Hiển thị kết quả đo

Các thao tác chính:

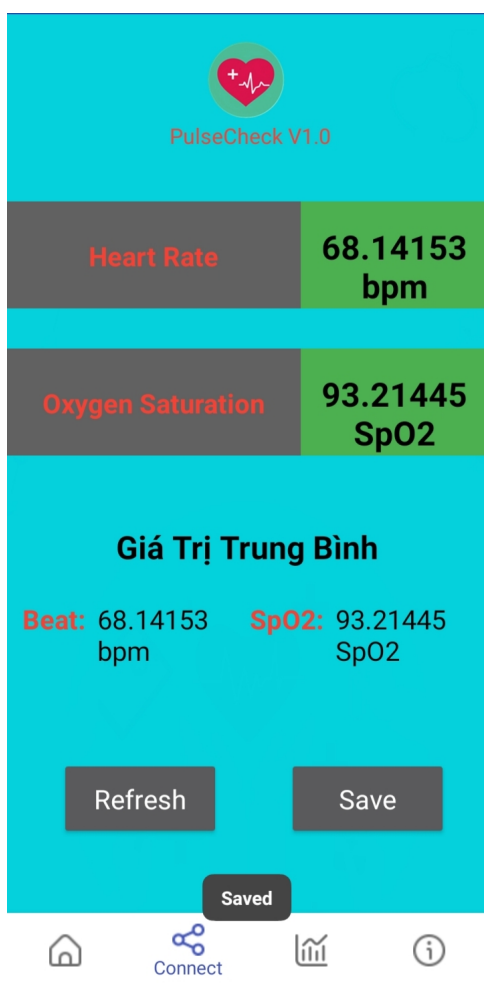
- **Refresh (làm mới):** Khi người dùng ấn nút, dữ liệu trên màn hình được làm mới, một thông báo xuất hiện khi làm mới thành công.

Kết quả:

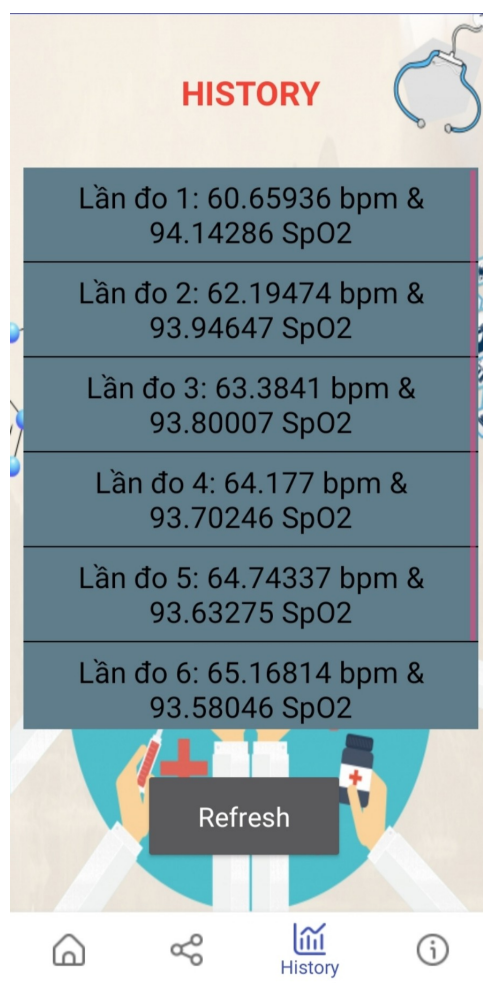


Hình 24: Làm mới thành công

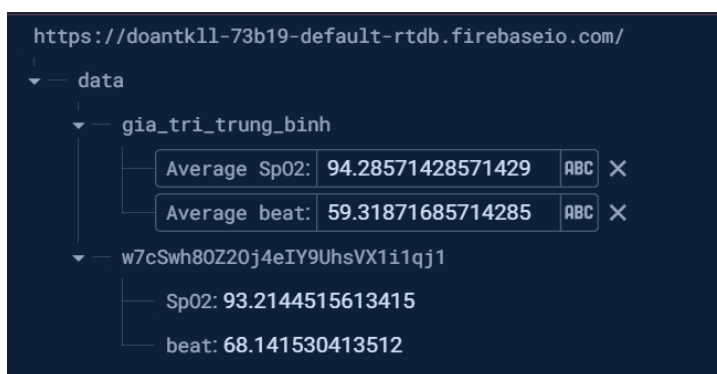
- **Save (lưu trữ):** Khi người dùng ấn nút, dữ liệu trên màn hình được lưu vào phần lịch sử đo của ứng dụng và firebase, một thông báo xuất hiện khi làm mới thành công. Kết quả:



Hình 25: Lưu thành công



Hình 26: Dữ liệu lưu vào lịch sử đo



Hình 27: Dữ liệu lưu vào firebase

8.2.2 Hiện thị lịch sử đo:

Giao diện này dùng để hiển thị lịch sử đo của người dùng, có thành phần chính là một danh sách các mục, mỗi mục ghi lại một lần đo đặc với nhịp tim và chỉ số SpO2 tương ứng khi dữ liệu của chúng thay đổi. Giống như màn hình trước, ở giao diện này cũng có một nút "Refresh" dưới cùng, dùng để làm mới bảng dữ liệu. Kết quả ta có một giao diện hoàn chỉnh như sau:



Hình 28: Lịch sử đo

Giao diện này chỉ có một thao tác chính là làm mới bảng lịch sử đo. Khi ấn nút "Refresh", dữ liệu sẽ được xóa đi và lần đo sẽ được bắt đầu lại từ 1, cuối cùng một dòng thông báo được hiển thị khi bạn xóa thành công. Kết quả:



Hình 29: Xóa dữ liệu thành công

9 Kết quả thu được

Sau đây là kết quả thu được khi nhóm sử dụng 2 thiết bị cảm biến để đo nhịp tim của các thành viên trong nhóm:

Tên	MAX30100	GY-MAX30100	MAX30102	Manual
Phát	33 bpm / 95% SpO_2	69 bpm/94 SpO_2	78 bpm	83 bpm
Đạt	33 bpm / 93% SpO_2	35 bpm/95 SpO_2	77 bpm	80 bpm
Phong	32 bpm / 92 % SpO_2	32 bpm/95 SpO_2	77 bpm	79bpm

Có thể thấy có sự chênh lệch về kết quả giữa 2 cảm biến và giữa cảm biến với việc đo thủ công. Đối với cảm biến, ta nhận thấy MAX30100 có sự sai số rất lớn so với kết quả thủ công. Về việc này thì có thể là một lỗi phần cứng, hoặc một lỗi không tương thích với mã nguồn được sử dụng hoặc một lỗi do giải thuật tính toán SpO_2 . Nhóm hiện tại chưa khắc phục được vấn đề với cảm biến MAX30100. Đối với giá trị SpO_2 thì có vẻ như cả 2 cảm biến đều đọc được với giá trị trong khoảng chấp nhận được.

Đối với cảm biến MAX30102, giá trị nhịp tim trả về gần như chính xác so với giá trị đo thủ công. Hơn nữa, giá trị này cực kỳ ổn định. Tuy nhiên khi sử dụng giải thuật tính toán nồng độ oxy trong máu thì kết quả trả về lại cực kỳ thiếu chính xác và tính ổn định cũng rất thấp. Đây cũng là một điểm trừ của nhóm khi quá phụ thuộc vào thư viện.

Nhìn chung, việc thao tác với Esp8266 và Firebase là không có vấn đề. Tuy nhiên nhóm lại có vấn đề đối với cảm biến, cụ thể là về giải thuật tính toán.

10 Thảo luận về đề tài

Nếu xét một cách đơn lẻ từng cảm biến thì có vẻ như các giá trị chúng ta đo được đều có thể sử dụng. Tuy nhiên cách làm này quá nặng tính chất từ chối giải quyết vấn đề. Đây là một nhược điểm to lớn của nhóm. Nhóm hi vọng lần tiếp theo sẽ khắc phục được tình trạng này.

Dữ liệu mà nhóm thu thập được chỉ mang tính chất tham khảo, không thể đại diện một cách chính xác cho sức khỏe của người sử dụng. Đề tài này mang tính chất học hỏi là chủ yếu, vì vậy nếu muốn chăm sóc tốt sức khỏe của bản thân thì hãy đến cơ sở y tế để được theo dõi bởi bác sỹ cũng như những thiết bị chuyên khoa hơn.

Qua đề tài lần này, nhóm chúng em đã có thêm kiến thức về thiết kế hệ thống IOT, cũng như database và web server. Cùng với những thứ đã đạt được, nhóm em vẫn có nhiều sự hạn chế, cụ thể là việc phụ thuộc thư viện, tiền xử lý dữ liệu cũng như giao tiếp với cảm biến. Đây là những khuyết điểm mà chúng em rút ra từ đề tài lần này và cố gắng cải thiện cho các môn học tiếp theo.

Dự án này đã mở ra một cánh cửa mới cho chúng tôi để hiểu rõ hơn về ứng dụng thực tế của kiến thức học tập. Việc áp dụng kiến thức từ lý thuyết vào thực tế không chỉ giúp chúng tôi nắm vững hơn mà còn cho phép chúng tôi nhìn nhận rõ hơn về những thách thức và cơ hội thực tế trong lĩnh vực này.

Chúng tôi cũng nhận thức rõ ràng rằng, trong quá trình học tập và nghiên cứu, việc đối mặt với những hạn chế và thất bại là điều không thể tránh khỏi. Tuy nhiên, điều quan trọng là chúng tôi đã học được từ những sai lầm và sẵn sàng điều chỉnh, cải thiện để trở nên tốt hơn trong tương lai.

Chúng tôi quyết tâm không ngừng học hỏi và phát triển khả năng của mình. Bằng cách tiếp tục nỗ lực và tập trung vào việc khắc phục những hạn chế, chúng tôi hy vọng sẽ có thể đạt được những kết quả tốt hơn trong các dự án và học phần tiếp theo.

References

References

- [1] Antonio Carlos Bento. An experimental survey with nodemcu12e+ shield with tft nextion and max30102 sensor. In *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 0082–0086. IEEE, 2020.
 - [2] Hiền Minh. Các nguyên nhân làm gia tăng số ca mắc và tử vong do bệnh tim mạch. *Báo Điện Tử Chính phủ*, 2023.
 - [3] Vikramsingh R Parihar, Akesh Y Tonge, Pooja D Ganorkar, et al. Heartbeat and temperature monitoring system for remote patients using arduino. *International Journal of Advanced Engineering Research and Science*, 4(5):55–58, 2017.
 - [4] Yuli Yessy Richa Rachmawati, Yulia Putri Ayu Sanjaya, and Stevany Edilia. Web-based temperature, oxygen saturation, and heart rate monitoring system. *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, 4(1):38–45, 2022.
- [4] [2] [3] [1]