

BAN CƠ YẾU CHÍNH PHỦ  
HỌC VIỆN KỸ THUẬT MẬT MÃ

## ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

ĐỀ TÀI  
NGHIÊN CỨU THIẾT KẾ VÀ CHẾ TẠO THIẾT BỊ THÔNG MINH  
HỖ TRỢ CHĂM SÓC SỨC KHỎE

Sinh viên thực hiện: Nguyễn Văn Phương

Khóa: 4

Chuyên ngành: Hệ thống nhúng và điều khiển tự động

Mã ngành: 7520207

Người hướng dẫn: ThS. Nguyễn Thị Thu Hiền

ĐH Công nghệ Giao thông Vận tải

HÀ NỘI - 2025

BAN CƠ YẾU CHÍNH PHỦ  
HỌC VIỆN KỸ THUẬT MẬT MÃ

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

ĐỀ TÀI  
NGHIÊN CỨU THIẾT KẾ VÀ CHẾ TẠO THIẾT BỊ THÔNG MINH  
HỖ TRỢ CHĂM SÓC SỨC KHỎE

Sinh viên thực hiện: Nguyễn Văn Phương

Khóa: 4

Chuyên ngành: Hệ thống nhúng và điều khiển tự động

Mã ngành: 7520207

Người hướng dẫn: ThS. Nguyễn Thị Thu Hiền

ĐH Công nghệ Giao thông Vận tải

HÀ NỘI - 2025

## LỜI CẢM ƠN

Sau quá trình học tập và rèn luyện tại Học viện Kỹ thuật mật mã, em lựa chọn thực hiện đề tài đồ án tốt nghiệp “**Thiết kế và chế tạo Thiết bị thông minh hỗ trợ chăm sóc sức khỏe**” dưới sự hướng dẫn của ThS. Nguyễn Thị Thu Hiền - Giảng viên ĐH Công nghệ Giao thông Vận tải. Em xin trân trọng cảm ơn cô vì sự hướng dẫn tận tình và hiệu quả trong quá trình thực hiện đồ án.

Em xin chân thành cảm ơn các giảng viên của Học viện Kỹ thuật mật mã đã nhiệt tình giảng dạy, truyền đạt kiến thức và chia sẻ kinh nghiệm, làm nền tảng cho việc thực hiện tốt đồ án tốt nghiệp.

Cuối cùng, em xin cảm ơn Ban Giám đốc, các phòng chức năng, hệ Quản lý sinh viên đã tạo mọi điều kiện thuận lợi và hỗ trợ em trong quá trình học tập tại Học viện; xin cảm ơn gia đình và người thân đã luôn động viên, khích lệ để em hoàn thành chương trình học tập một cách tốt nhất.

*Hà Nội, ngày      tháng      năm 2025*

**Sinh viên thực hiện**

*(Ký tên và ghi rõ họ tên)*

**Nguyễn Văn Phương**

**LỜI CAM ĐOAN**

Em xin cam đoan bản đồ án này do em tự nghiên cứu dưới sự hướng dẫn của cô giáo ThS. Nguyễn Thị Thu Hiền.

Để hoàn thành đồ án này, em chỉ sử dụng những tài liệu đã ghi trong mục tài liệu tham khảo, ngoài ra không sử dụng bất cứ tài liệu nào khác không được ghi trong mục này.

Nếu sai, em xin chịu mọi hình thức kỷ luật theo quy định của Học viện.

*Hà Nội, ngày      tháng      năm 2025*

**Sinh viên thực hiện**

*(Ký tên và ghi rõ họ tên)*

**Nguyễn Văn Phương**

## MỤC LỤC

	Trang
<b>LỜI CẢM ƠN .....</b>	<b>i</b>
<b>LỜI CAM ĐOAN .....</b>	<b>ii</b>
<b>MỤC LỤC .....</b>	<b>iii</b>
<b>CÁC KÝ HIỆU, CHỮ VIẾT TẮT .....</b>	<b>vi</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>ix</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>x</b>
<b>MỞ ĐẦU .....</b>	<b>1</b>

### CHƯƠNG 1.

#### TỔNG QUAN VỀ ĐỀ TÀI

1.1. Tính thực tiễn của đề tài .....	3
1.2. Tổng quan về chỉ số sức khỏe con người .....	5
1.2.1. Nhịp tim và chỉ số nhịp tim .....	5
1.2.2. Nồng độ oxy trong máu .....	9
1.2.3. Chỉ số chuyển động .....	10
1.2.4. Đo lường năng lượng tiêu hao .....	11
1.2.5. Thời gian ngủ .....	14
1.3. Thiết bị y tế theo dõi sức khỏe .....	16
1.3.1. Giới thiệu .....	16
1.3.2. Một số loại thiết bị theo dõi sức khỏe phổ biến .....	18
1.3.3. Ưu điểm và nhược điểm .....	19
1.3.4. Xu hướng phát triển .....	20
1.4. Sơ lược về Internet vạn vật - IoT .....	21
1.4.1. Giới thiệu .....	21
1.4.2. Cấu trúc của hệ thống IoT .....	23
1.4.3. Ứng dụng .....	24

### CHƯƠNG 2.

#### CÔNG NGHỆ THIẾT KẾ

2.1. Giải pháp thiết kế .....	28
2.1.1. Mục tiêu thiết kế.....	28
2.1.2. Sơ đồ khối .....	28
2.2. Công nghệ phần cứng.....	29
2.2.1. ESP32-S3-MINI-1.....	29
2.2.2. Cảm biến MAX30102 .....	33
2.2.3. Cảm biến LIS2DH12TR .....	34
2.2.4. IC sạc TP4056 .....	36
2.2.5. IC LDO LP2985-33DBVR .....	37
2.2.6. Màn hình OLED 0.96 Inch.....	38
2.2.6. Các linh kiện khác .....	40
2.3. Các chuẩn giao tiếp .....	41
2.3.1. Giao tiếp I2C .....	41
2.3.2. Giao thức không dây MQTT .....	43
2.3.3. Giao thức không dây BLE.....	45
2.4. Công nghệ phần mềm.....	47
2.4.1. Ngôn ngữ lập trình C.....	47
2.4.2. ESP-IDF Framework.....	48
2.4.3. Ngôn ngữ lập trình Python.....	50
2.4.4. Firebase .....	53
2.4.5. Ngôn ngữ lập trình Dart và Framework Flutter .....	57

### **CHƯƠNG 3.**

#### **THIẾT KẾ VÀ THI CÔNG**

3.1. Thiết kế phần cứng.....	66
3.1.1. Sơ đồ nguyên lý của hệ thống .....	66
3.1.2. Bản vẽ PCB của thiết bị .....	68
3.1.2. Thi công thiết bị .....	69
3.2. Thiết kế phần mềm.....	71
3.2.1. Lưu đồ thuật toán và nguyên lý hoạt động.....	71
3.3. Triển khai thử nghiệm và đánh giá hoạt động của sản phẩm.....	76

3.3.1. Thử nghiệm hoạt động của sản phẩm .....	76
3.3.2. Đánh giá hoạt động .....	82
<b>KẾT LUẬN .....</b>	<b>83</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>84</b>
<b>PHỤ LỤC .....</b>	<b>86</b>

## CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
AC	Alternating Current	Dòng điện xoay chiều
ACC	Acceleration (Vector Magnitude)	Gia tốc/Chỉ số chuyển động
ADC	Analog-to-Digital Converter	Bộ chuyển đổi tương tự-số
AI	Artificial Intelligence	Trí tuệ nhân tạo
BLE	Bluetooth Low Energy	Bluetooth năng lượng thấp
BMR	Basal Metabolic Rate	Tốc độ trao đổi chất cơ bản
CAN	Controller Area Network	Mạng truyền thông CAN
CLK	Clock	Xung nhịp
CPU	Central Processing Unit	Bộ xử lý trung tâm
DAC	Digital-to-Analog Converter	Bộ chuyển đổi số-tương tự
DC	Direct Current	Dòng điện một chiều
DMA	Direct Memory Access	Truy cập bộ nhớ trực tiếp
ECG	Electrocardiogram	Điện tâm đồ
EE	Energy Expenditure	Năng lượng tiêu hao
ESP32	Espressif Systems 32-bit SoC	Vi điều khiển ESP32
FPC	Flexible Printed Circuit	Mạch in linh hoạt
GPIO	General-Purpose I/O	Ngõ vào/ra đa mục đích
GPS	Global Positioning System	Hệ thống định vị toàn cầu
HR	Heart Rate	Nhịp tim
HTTP	HyperText Transfer Protocol	Giao thức truyền siêu văn bản



I2C	Inter-Integrated Circuit	Bus giao tiếp I <sup>2</sup> C
IC	Integrated Circuit	Vi mạch tích hợp
IDE	Integrated Development Environment	Môi trường phát triển tích hợp
IDF	IoT Development Framework (ESP-IDF)	Bộ khung phát triển ESP-IDF
IoT	Internet of Things	Internet vạn vật
IP	Internet Protocol	Giao thức Internet
IPS	In-Plane Switching (LCD)	Công nghệ màn hình IPS
LED	Light-Emitting Diode	Đi-ốt phát sáng
MET	Metabolic Equivalent of Task	Đơn vị cường độ chuyển hoá
MQTT	Message Queuing Telemetry Transport	Giao thức nhắn tin nhẹ MQTT
PCB	Printed Circuit Board	Bảng mạch in
PSRAM	Pseudo Static RAM	Bộ nhớ truy cập tĩnh giả
PWM	Pulse-Width Modulation	Điều chế độ rộng xung
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
REM	Rapid Eye Movement (sleep stage)	Pha ngủ chuyển động mắt nhanh
RMSE	Root Mean Square Error	Sai số căn trung bình bình phương
ROM	Read-Only Memory	Bộ nhớ chỉ đọc
RTC	Real-Time Clock	Đồng hồ thời gian thực
SCL	Serial Clock Line	Dây xung nhịp bus I <sup>2</sup> C
SDA	Serial Data Line	Dây dữ liệu bus I <sup>2</sup> C

SPI	Serial Peripheral Interface	Giao diện ngoại vi nối tiếp
SpO <sub>2</sub>	Peripheral Oxygen Saturation	Độ bão hoà oxy ngoại vi
SRAM	Static RAM	Bộ nhớ truy cập tĩnh
UART	Universal Asynchronous Receiver-Transmitter	Giao tiếp nối tiếp bất đồng bộ
USB	Universal Serial Bus	Bus nối tiếp vạn năng
VCC	Voltage Common Collector	Nguồn cấp VCC (dương)
VM	Vector Magnitude	Độ lớn vector gia tốc
Wi-Fi	Wireless Fidelity	Mạng không dây Wi-Fi

## DANH MỤC BẢNG BIỂU

	Trang
Bảng 1.1. Chỉ số nhịp tim theo độ tuổi của nam và nữ.....	6
Bảng 1.2. Nhịp tim của người trưởng thành .....	6
Bảng 1.3. Bảng phân loại mức độ vận động theo chỉ số PAL .....	11
Bảng 1.4. Bảng so sánh hiệu quả giữa các mô hình.....	14
Bảng 2.1. Bảng thông số kỹ thuật chính. ....	39
Bảng 2.2. Bảng chân tín hiệu. ....	40
Bảng 2.3. Bảng các linh kiện khác được sử dụng trong dự án. ....	40
Bảng 3.1. Bảng kết nối ESP32-S3-MIN-1 với các linh kiện .....	66

## DANH MỤC HÌNH ẢNH

	Trang
Hình 1.1. Điện tâm đồ .....	7
Hình 1.2. Đo nhịp tim bằng phương pháp hấp thụ quang học .....	8
Hình 1.3. Hoạt động thể chất .....	12
Hình 1.4. Thiết bị theo dõi sức khỏe thông minh .....	17
Hình 1.5. Đồng hồ thông minh.....	18
Hình 1.6. Máy đo huyết áp tự động và máy đo nồng độ oxy trong máu .....	19
Hình 1.7. Oura Ring .....	19
Hình 1.8. Đồng hồ thông minh tích hợp trí tuệ nhân tạo .....	21
Hình 1.9. Hệ thống IoT. ....	22
Hình 1.10. Cấu trúc hệ thống IoT. ....	23
Hình 1.11. IoT ứng dụng trong y tế .....	25
Hình 1.12. Thiết bị IoT trong nhà thông minh giúp cuộc sống dễ dàng hơn.....	27
Hình 2.1. Sơ đồ khối .....	28
Hình 2.2. Module ESP32-S3-MINI-1-N8.....	29
Hình 2.3. Cấu hình của module ESP32-S3-MINI-1-N8.....	31
Hình 2.4. Sơ đồ chức năng các chân của ESP32-S3-MINI-1-N8.....	32
Hình 2.5. Kích thước của ESP32-S3-MINI-1-N8.....	33
Hình 2.6. Cảm biến Max30102 .....	33
Hình 2.7. Cảm biến LIS2DH12TR .....	35
Hình 2.8. TP4056 – IC Sạc Li-Ion Linear 1 A (SOP-8) .....	36
Hình 2.9. LP2985-33DBVR – IC LDO 3.3 V 150 mA .....	37
Hình 2.10. Màn hình OLED 0.96 Inch .....	38
Hình 2.11. Chuẩn giao tiếp I2C .....	41
Hình 2.12. Mô hình MQTT.....	43
Hình 2.13. Mô hình BLE.....	45
Hình 2.14. Ngôn ngữ lập trình C.....	47
Hình 2.15. Logo ESP-IDF Framework .....	48

Hình 2.16. Giao diện cấu hình của ESP-IDF.....	50
Hình 2.17. Logo ngôn ngữ lập trình Python .....	51
Hình 2.18. Firebase .....	53
Hình 2.19. Firebase Realtime Database .....	56
Hình 2.20. Mô hình Firebase Realtime Database .....	57
Hình 2.21. Ngôn ngữ lập trình Dart .....	58
Hình 2.22. Khung Flutter .....	59
Hình 2.23. Kiến trúc của Flutter.....	62
Hình 3.1. Sơ đồ nguyên lý khối hiển thị màn hình OLED .....	66
Hình 3.2. Sơ đồ nguyên lý khối vi điều khiển và nguồn.....	67
Hình 3.3. Sơ đồ nguyên lý khối cảm biến gia tốc 3 trục.....	67
Hình 3.4. Sơ đồ nguyên lý khối cảm biến nhịp tim và nồng độ oxy trong máu .	68
Hình 3.5. Mặt sau bảng PCB.....	68
Hình 3.6. Mặt trước bảng PCB.....	69
Hình 3.7. Mạch điều khiển hoàn thiện - mặt sau .....	69
Hình 3.8. Mạch điều khiển hoàn thiện - mặt trước .....	70
Hình 3.9. Lắp đặt mạch vào trong hộp.....	70
Hình 3.10. Sản phẩm hoàn chỉnh .....	71
Hình 3.11. Lưu đồ thuật toán thiết bị.....	73
Hình 3.12. Lưu đồ thuật toán ứng dụng .....	74
Hình 3.13. Lưu đồ thuật toán Server.....	75
Hình 3.14. Màn hình hiển thị khi thiết bị hoạt động ổn định.....	76
Hình 3.15. Màn hình trên ứng dụng di động khi thiết bị hoạt động ổn định .....	76
Hình 3.16. So sánh đồng bộ dữ liệu giữa thiết bị và ứng dụng di động .....	77
Hình 3.17. Hiển thị thời gian ngủ theo các chỉ số.....	78
Hình 3.18. Năng lượng tiêu hao theo nhịp tim và hoạt động trong ngày .....	79
Hình 3.19. Lịch sử đo chỉ số nhịp tim.....	80
Hình 3.20. Lịch sử đo chỉ số nồng độ oxy trong máu.....	81

## MỞ ĐẦU

Trong những năm gần đây, với sự phát triển mạnh mẽ của cuộc cách mạng công nghiệp 4.0, khoa học kỹ thuật hiện đại đã có những bước tiến vượt bậc, tác động sâu rộng đến mọi mặt của đời sống xã hội, trong đó có lĩnh vực y tế và chăm sóc sức khỏe cộng đồng. Cùng với nhu cầu ngày càng cao của con người trong việc bảo vệ sức khỏe, các thiết bị công nghệ thông minh đang ngày càng được ứng dụng rộng rãi để hỗ trợ theo dõi và cải thiện tình trạng sức khỏe cá nhân một cách chủ động, tiện lợi và hiệu quả. Điều này là hết sức cần thiết, đặc biệt là trong bối cảnh cuộc sống ngày càng nhiều áp lực và hối hả khiến nhiều người không thể thu xếp được thời gian cho việc thăm khám sức khỏe định kỳ, môi trường sống ô nhiễm nghiêm trọng và ngày càng phát sinh nhiều dịch bệnh nguy hiểm, cùng với đó là tình trạng các hệ thống y tế luôn quá tải và đang phải đối mặt với những thách thức về yêu cầu ngày càng tăng của người dân đối với chất lượng phải được nâng cao cũng như chi phí của dịch vụ y tế hiện nay.

Xuất phát từ thực tế đó và mong muốn vận dụng những kiến thức đã được học trong quá trình đào tạo tại Học viện Kỹ thuật mật mã để thiết kế một sản phẩm có tính ứng dụng cao, đóng góp vào việc nâng cao nhận thức về chăm sóc sức khỏe chủ động, đề tài **“Nghiên cứu thiết kế và chế tạo thiết bị thông minh hỗ trợ chăm sóc sức khỏe”** được em lựa chọn làm đề án tốt nghiệp. Thiết bị được thiết kế và chế tạo nhằm hỗ trợ người dùng theo dõi các thông số sức khỏe cơ bản là đo nhịp tim và nồng độ ô xy trong máu. Bên cạnh đó, thiết bị được tích hợp chức năng theo dõi giấc ngủ và tính toán lượng calo tiêu thụ trong ngày, giúp người dùng biết được chính xác hơn về tổng thời gian ngủ, giúp cải thiện thói quen ngủ, thêm các hoạt động thể thao trong ngày, tập thể thao, đi bộ hoặc yoga,... tăng thêm các hoạt động giải trí, giảm stress,...nhằm cải thiện sức khỏe một cách toàn diện. Thiết bị cũng được tích hợp công nghệ định vị GPS, phục vụ cho mục đích hỗ trợ kịp thời cho người dùng trong các tình huống khẩn cấp.

Với việc sử dụng các cảm biến phù hợp, kết hợp công nghệ Internet vạn vật, thiết bị thông minh hỗ trợ chăm sóc sức khỏe được coi là một giải pháp hữu hiệu

trong việc chăm sóc sức khỏe chủ động và được kỳ vọng tiếp tục cải tiến để thực hiện được nhiều nhiệm vụ hơn nữa.

Nội dung của đồ án trình bày cơ sở lý thuyết, giải pháp và công nghệ thiết kế, cũng như quy trình thiết kế, chế tạo thiết bị, được cấu trúc thành 3 chương chính:

- ❖ Chương 1. Tổng quan về đề tài.
- ❖ Chương 2. Công nghệ thiết kế.
- ❖ Chương 3. Thiết kế và thi công.

Do vẫn còn hạn chế về kiến thức và kinh nghiệm thực tiễn nên chắc chắn đồ án không tránh khỏi thiếu sót. Em rất mong nhận được sự góp ý của thầy cô để nội dung đồ án hoàn thiện hơn.

## CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Tính thực tiễn của đề tài

Đối với mỗi người, sức khỏe là một tài sản vô giá, không có sức khỏe chúng ta không thể làm được bất kì thứ gì. Trong xã hội hiện đại, sức khỏe đóng vai trò quan trọng không thể phủ nhận. Nó là nền tảng đối với hạnh phúc cá nhân, sự phát triển của mỗi người, gia đình và xã hội. Cuộc sống được coi là trọn vẹn chỉ khi chúng ta có một tâm trí an lành bên trong một cơ thể khỏe mạnh. Thế nhưng, nhịp sống bận rộn, hối hả, quá nhiều lo toan và đầy áp lực của xã hội hiện đại đã khiến cho nhiều người vô tình có thái độ thờ ơ với sức khỏe của chính bản thân mình, việc khám sức khỏe định kỳ tại các cơ sở y tế bị xem như một việc làm xa xỉ, lãng phí, tốn thời gian, dẫn đến tình trạng bỏ qua cơ hội “vàng” để phát hiện sớm và điều trị thành công các bệnh lý nguy hiểm như bệnh lao, tim mạch, tiểu đường, các rối loạn chức năng hô hấp, cao huyết áp, ... Chính vì vậy, việc tự chăm sóc giữ gìn sức khỏe cần bắt đầu càng sớm càng tốt để phòng ngừa nguy cơ ốm bệnh, nâng cao chất lượng cuộc sống. Nếu như trước đây, người Việt quan tâm đến việc chữa bệnh hơn phòng bệnh, không có thói quen chủ động chăm sóc và theo dõi sức khỏe định kỳ thì trong xã hội hiện đại ngày nay, khi điều kiện kinh tế khá hơn, việc chăm sóc sức khỏe chủ động đã được chú trọng hơn rất nhiều, một bộ phận lớn người dân đã bắt đầu quen với xu hướng chăm sóc sức khỏe chủ động đã phổ biến trên thế giới từ rất lâu. Chăm sóc sức khỏe chủ động hướng đến việc tự chăm sóc bản thân, tăng cường sức khỏe để giảm nguy cơ mắc bệnh và theo dõi sức khỏe sát sao để sớm phát hiện các nguy cơ bệnh lý thay vì tập trung vào điều trị khi đã mắc bệnh. Điều này không chỉ giúp nâng cao sức khỏe, cải thiện chất lượng cuộc sống của mỗi người mà còn có tác dụng giảm gánh nặng y tế cho xã hội và không trở thành gánh nặng cho quỹ Bảo hiểm y tế Nhà nước.

Hiện nay, cùng với sự phát triển của xã hội, cuộc sống ngày càng được nâng cao thì việc áp dụng công nghệ khoa học kỹ thuật vào đời sống, công việc ngày càng trở nên cần thiết. Với sự phát triển của các ngành khoa học kỹ thuật, công nghệ kỹ thuật điện tử mà trong đó đặc biệt là kỹ thuật tự động đóng một vai trò



quan trọng trong mọi lĩnh vực khoa học kỹ thuật, quản lý, công nghiệp, nông nghiệp, đời sống, quản lý thông tin, ... ngày càng được ứng dụng rộng rãi, trong đó có cả lĩnh vực y tế và chăm sóc sức khỏe. Thực tế cho thấy, các thiết bị y tế ứng dụng công nghệ ngày càng đa dạng, phù hợp với nhiều môi trường khác nhau, từ quy mô lớn như bệnh viện, các cơ sở y tế cho đến các gia đình hoặc cá nhân. Hơn thế nữa, trong hoàn cảnh môi trường sống ngày càng ô nhiễm, tiềm ẩn nhiều dịch bệnh nguy hiểm, cộng thêm những áp lực từ cuộc sống thì việc trang bị các thiết bị y tế, thiết bị chăm sóc sức khỏe tại nhà là điều vô cùng cần thiết, giúp xử lý kịp thời các tình huống khẩn cấp. Việc chủ động theo dõi sức khỏe tại nhà cũng giúp phát hiện sớm các bất thường trong cơ thể và theo dõi tình trạng bệnh lý để có biện pháp điều trị kịp thời.

Ngoài ra, các thống kê gần đây cho thấy chất lượng giấc ngủ của người dân, nhất là người trẻ và người cao tuổi, đang có chiều hướng suy giảm nghiêm trọng, kéo theo nhiều hệ lụy tiêu cực về cả thể chất lẫn tinh thần. Vì vậy, theo dõi để có phương án cải thiện chất lượng giấc ngủ là cần thiết. Tuy nhiên, phần lớn các giải pháp theo dõi giấc ngủ hiện nay đều phụ thuộc vào các thiết bị chuyên dụng có chi phí cao hoặc yêu cầu hạ tầng y tế tương đối phức tạp, gây khó khăn cho việc triển khai trên diện rộng. Bên cạnh đó, các tình huống khẩn cấp xảy ra trong lúc ngủ như ngưng thở, loạn nhịp tim hay tai biến nếu không được phát hiện kịp thời có thể dẫn đến hậu quả nghiêm trọng. Tuy nhiên, việc theo dõi liên tục trong lúc nghỉ ngơi là điều mà các thiết bị thông thường không thể đáp ứng đầy đủ.

Với những lý do trên, đề án lựa chọn thiết kế và chế tạo một thiết bị có tính năng theo dõi các chỉ số sức khỏe nền tảng như nhịp tim, cử động cơ thể, mức độ vận động, theo dõi giấc ngủ với thao tác sử dụng đơn giản, kích thước gọn nhẹ. Ngoài ra, thiết bị được ứng dụng công nghệ IoT, kết nối wifi/4G, khả năng tích hợp truyền tin không dây có định vị cho phép thiết bị hoạt động như một công cụ hỗ trợ khẩn cấp, giúp kết nối với người thân hoặc trung tâm y tế khi phát hiện sự cố nghiêm trọng.

## 1.2. Tổng quan về chỉ số sức khỏe con người

Chỉ số sức khỏe là các thông số thể hiện tình trạng sức khỏe của một người ở thời điểm hiện tại. Dưới đây là một số chỉ số sức khỏe quan trọng nên quan tâm.

### 1.2.1. Nhịp tim và chỉ số nhịp tim

Nhịp tim là một trong các chỉ số quan trọng giúp bác sĩ chẩn đoán được các bất thường trong cơ thể của một người. Chính vì vậy mà nhịp tim tăng cao hay giảm xuống đột ngột có thể là dấu hiệu nhận biết các bệnh lý nguy hiểm mà người đang mắc phải.

Nhịp tim là tốc độ nhịp tim đo bằng số lần co thắt (nhịp đập) của tim trong mỗi phút (BPM - beat per minute). Trung bình một đời người, tim sẽ đập khoảng 3 tỉ lần. Mỗi chu kỳ tim thực hiện co bóp một lần trong tất cả các buồng tim, khoảng thời gian giữa 2 lần co bóp của 1 buồng tim được coi là 1 chu kỳ, thường nhịp co bóp của tâm thất trái được xác định để tính chu kỳ nghe nhịp tim. Trong quá trình đó, tim sẽ đưa máu có chứa oxy đi tới các cơ quan trong cơ thể và vận chuyển máu thiếu oxy trở lại phổi. Nhịp tim có thể thay đổi theo nhu cầu thể chất của cơ thể, bao gồm cả nhu cầu hấp thu oxy và bài tiết carbon dioxit. Các hoạt động có thể tạo ra thay đổi nhịp tim bao gồm tập thể dục, ngủ, lo lắng, căng thẳng, bệnh tật và khi uống thuốc. Nhịp tim của người bình thường cũng sẽ thay đổi theo tuổi tác. Nói chung, bất kỳ thay đổi lớn nào trong trạng thái thể chất hoặc tinh thần của một người thường dẫn đến thay đổi mạch. Ngoài ra, nhịp tim còn chịu tác động từ nhiều yếu tố khác như:

- ❖ Nhiệt độ không khí: Nhiệt độ tăng cao thường sẽ khiến tim bơm máu mạnh hơn và nhịp tim tăng 5-10 nhịp/phút.
- ❖ Tư thế khi đo: Yếu tố này chỉ ảnh hưởng đáng kể trong 15-20 giây đầu tiên.
- ❖ Cân nặng: Những người bị bệnh béo phì, thừa cân thường có nhịp tim cao hơn mức thông thường.
- ❖ Sử dụng thuốc: Các loại thuốc có chứa chất ức chế beta sẽ khiến nhịp tim chậm lại, trong khi thuốc chữa trị tuyến giáp lại khiến nhịp tim tăng lên.
- ❖ ....

Chỉ số nhịp tim có vai trò quan trọng trong việc nhận biết các dấu hiệu bất thường và chẩn đoán một số vấn đề sức khỏe. Chỉ số nhịp tim bình thường là khác nhau giữa các cá thể, phụ thuộc vào giới tính, độ tuổi, tình trạng sức khỏe (Bảng 1.1 và 1.2). Sự thay đổi của chỉ số nhịp tim có thể là dấu hiệu cho thấy một sự thay đổi của trạng thái tim, qua đó có thể phản ánh tình trạng sức khỏe cơ thể. Chỉ số nhịp tim nhằm đến khi cơ thể ở trạng thái hoạt động như làm việc, tập thể dục, vận động cơ thể,... Khi đó, tim cần hoạt động co bóp mạnh để cung cấp đủ lượng máu nuôi các cơ quan. Chỉ số nhịp tim thường được sử dụng để theo dõi trong quá trình luyện tập, bảo đảm các bài tập không quá sức. Vì vậy, theo dõi chỉ số nhịp tim của một người trong thời gian dài sẽ cho ta biết được phần nào tình trạng sức khỏe, thói quen hàng ngày, chất lượng cuộc sống của họ, qua đó có những biện pháp cải thiện kịp thời.

Bảng 1.1. Chỉ số nhịp tim theo độ tuổi của nam và nữ

Nam			Nữ		
Nhóm tuổi	Vùng nhịp tim trung bình (bpm)	Nhịp tim bình thường (bpm)	Nhóm tuổi	Vùng nhịp tim trung bình (bpm)	Nhịp tim bình thường (bpm)
Dưới 10	76 – 94	66 – 106	Dưới 10	70 - 92	66 – 105
11 – 19	69 – 87	58 – 99	11 – 19	66 – 87	57 – 98
20 – 39	66 – 82	57 – 95	20 – 39	61 – 78	52 – 92
40 – 59	64 – 79	56 – 92	40 – 59	61 – 77	52 – 89
60 – 79	64 – 78	56 – 92	60 – 79	60 – 75	50 – 91
Trên 80	64 – 77	56 – 93	Trên 80	61 – 78	51 – 94

\*Dữ liệu được tổng hợp dựa trên báo cáo của trung tâm Kiểm soát và ngăn ngừa dịch bệnh, Bộ thống kê y tế quốc gia

Theo Hội Tim mạch Hoa Kỳ (America Heart Association - AHA), với hầu hết người trưởng thành, nhịp tim khi nghỉ ngơi bình thường là 60 - 100 nhịp mỗi phút. Nhịp tim dưới 60 lần/phút hoặc trên 100 lần/phút được xem là bình thường trong một số trường hợp như khi ngủ (nhịp tim có thể giảm xuống dưới 60 lần/phút), khi chạy bộ quá sức, vận động mạnh hoặc căng thẳng (nhịp tim đập nhanh trên 100 lần/phút). Còn nhịp tim cao hoặc thấp bất thường được coi là đáng lo ngại khi xảy ra trong nhiều giờ hoặc nhiều ngày và đi kèm với các triệu chứng khác như nhịp tim quá nhanh, kéo dài liên tục trên 100 nhịp mỗi phút, kèm các triệu chứng như chóng mặt, khó thở, đau ngực hoặc quá chậm (dưới 40 lần/phút).

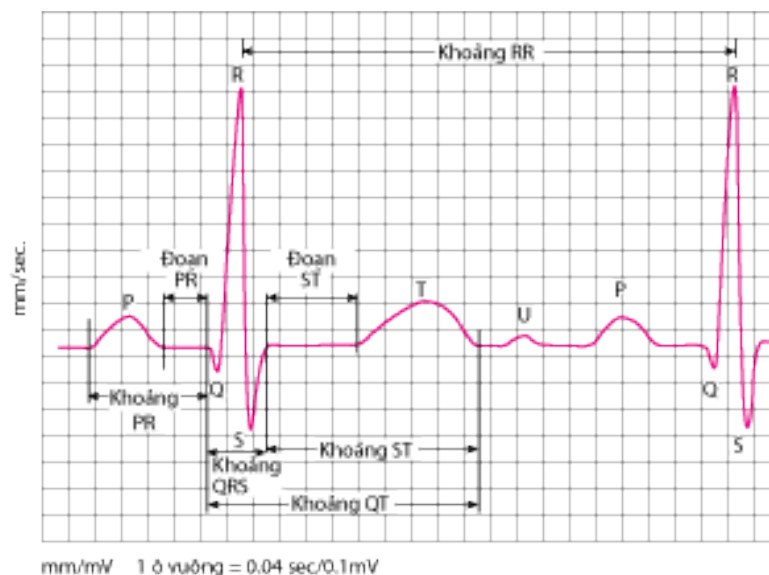
Bảng 1.2. Nhịp tim của người trưởng thành

Resting Heart Rate Chart						
Men (beats per minute)						
Age	18 - 25	26 - 35	36 - 45	46 - 55	56 - 65	65 +
Athlete	49 - 55	49 - 54	50 - 56	50 - 57	51 - 56	50 - 55
Excellent	56 - 61	55 - 61	57 - 62	58 - 63	57 - 61	56 - 61
Great	62 - 65	62 - 65	63 - 66	64 - 67	62 - 67	62 - 65
Good	66 - 69	66 - 70	67 - 70	68 - 71	68 - 71	66 - 69
Average	70 - 73	71 - 74	71 - 75	72 - 76	72 - 75	70 - 73
Below Average	74 - 81	75 - 81	76 - 82	77 - 83	76 - 81	74 - 79
Poor	82 +	82 +	83 +	84 +	82 +	80 +
Women (beats per minute)						
Age	18 - 25	26 - 35	36 - 45	46 - 55	56 - 65	65 +
Athlete	54 - 60	54 - 59	54 - 59	54 - 60	54 - 59	54 - 59
Excellent	61 - 65	60 - 64	60 - 64	61 - 65	60 - 64	60 - 64
Great	66 - 69	65 - 68	65 - 69	66 - 69	65 - 68	65 - 68
Good	70 - 73	69 - 72	70 - 73	70 - 73	69 - 73	69 - 72
Average	74 - 78	73 - 76	74 - 78	74 - 77	74 - 77	73 - 76
Below Average	79 - 84	77 - 82	79 - 84	78 - 83	78 - 83	77 - 84
Poor	85 +	83 +	85 +	84 +	84 +	85 +

agelessinvesting.com

Nhịp tim có thể được đo chính xác bằng nhiều kỹ thuật. Sau đây là một số phương pháp đo phổ biến:

❖ Đo bằng tín hiệu điện:



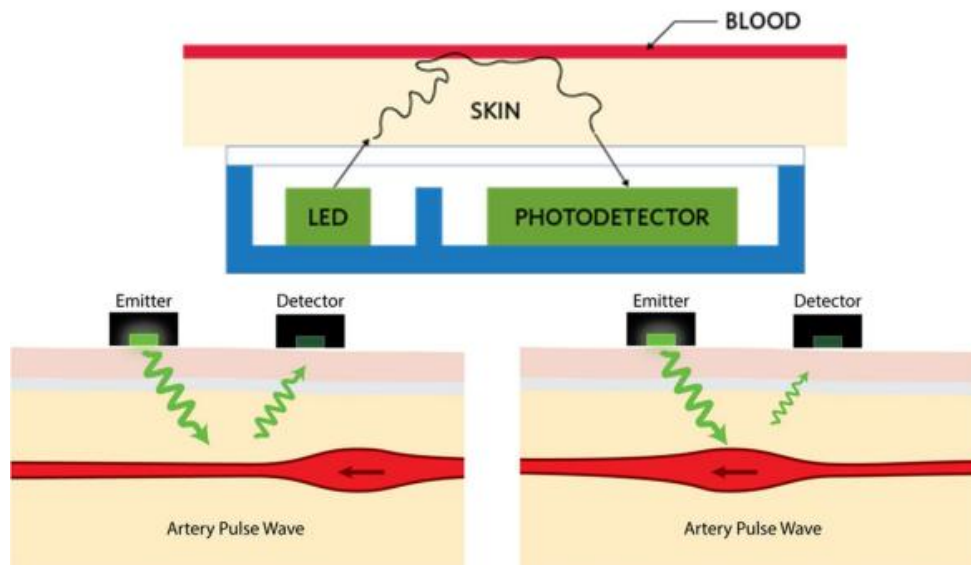
Hình 1.1. Điện tâm đồ

Điện tâm đồ (ECG) đo nhịp tim bằng cách sử dụng các miếng dán điện cực nhỏ dán lên da ngực, cánh tay và chân. Bởi vì các tín hiệu điện tâm đồ là tuần

hoàn, nhịp tim có thể được suy ra từ khoảng sóng R đến sóng R (RR) của các tín hiệu tuần hoàn này.

❖ Đo nhịp tim bằng phương pháp hấp thụ quang học:

Khi tim đập, máu sẽ được đẩy đi khắp cơ thể qua động mạch, tạo ra sự thay đổi về áp suất trên thành động mạch và lượng máu chảy qua động mạch. Vì vậy, ta có thể đo nhịp tim bằng cách đo những sự thay đổi đó. Khi hàm lượng máu trong thành động mạch thay đổi sẽ làm thay đổi mức độ hấp thụ ánh sáng của động mạch, do đó khi một tia sáng được truyền qua động mạch thì cường độ ánh sáng sau khi truyền qua sẽ biến thiên đồng bộ với nhịp tim. Khi tim giãn ra, lượng máu qua động mạch nhỏ nên hấp thụ ít ánh sáng, dẫn đến ánh sáng sau khi truyền qua động mạch sẽ có cường độ lớn; ngược lại khi tim co vào, lượng máu qua động mạch lớn hơn, ánh sáng sau khi truyền qua động mạch sẽ có cường độ nhỏ hơn.



Hình 1.2. Đo nhịp tim bằng phương pháp hấp thụ quang học

Về nguyên tắc có thể đặt nguồn sáng và photodiode ở bất cứ nơi nào trên cơ thể có chứa động mạch. Vị trí đặt cảm biến hợp lý nhất là các đầu ngón tay, tuy động mạch ở vị trí này không quá lớn nhưng bề dày cơ thể ánh sáng phải truyền qua lại tương đối ít nên chỉ cần dùng 1 LED làm nguồn phát. Mặt khác, ở vị trí này cho mức độ biến thiên cường độ ánh sáng nhận được là khá lớn so với toàn bộ ánh sáng nhận được, tỉ số giữa biên độ tín hiệu với nền một chiều là đủ lớn để phân xử lý tín hiệu hoạt động đưa ra kết quả chính xác nhất.

### 1.2.2. Nồng độ oxy trong máu.

Lượng oxy trong máu hay độ bão hòa oxy trong máu ( $SpO_2$ ) là một thông số cực kỳ quan trọng, là thước đo lượng oxy mà tế bào máu của một người đang vận chuyển. Duy trì sự cân bằng máu giàu oxy là điều cần thiết cho sức khỏe.

Một phân tử Hemoglobin (Hb) có thể kết hợp với 4 phân tử oxy, khi đã gắn đủ 4 phân tử oxy được gọi là bão hòa oxy. Độ bão hòa oxy trong máu còn được gọi là chỉ số  $SpO_2$ , biểu thị cho tỷ lệ Hemoglobin có oxy trên tổng lượng Hemoglobin trong máu. Nếu tất cả các phân tử Hemoglobin trong máu đều gắn với oxy thì độ bão hòa oxy là 100%. Chỉ số này được xem là một trong những dấu hiệu sinh tồn của cơ thể, bên cạnh các dấu hiệu như nhiệt độ, mạch, nhịp thở và huyết áp. Khi bị thiếu oxy máu, các cơ quan như tim, gan, não... sẽ chịu tác động tiêu cực rất nhanh. Vì vậy, cần theo dõi chỉ số nồng độ oxy trong máu thường xuyên để kịp thời can thiệp nếu xảy ra tình trạng nguy hiểm của cơ thể để đưa ra được những giải pháp kịp thời.

Hầu hết các phân tử Hb sẽ gắn với oxy khi chúng đi qua phổi. Thang đo chỉ số  $SpO_2$  tiêu chuẩn như sau:

- ❖ 97% - 99%: oxy trong máu tốt.
- ❖ 94% - 96%: oxy trong máu trung bình, cần cho thở thêm oxy.
- ❖ 90% - 93%: oxy trong máu thấp, nên có y tá hoặc bác sĩ theo dõi hoặc đến bệnh viện gần nhất.
- ❖ Dưới 92% không thở oxy hoặc dưới 95% có thở oxy: Suy hô hấp rất nặng.
- ❖ Dưới 90% là biểu hiện của một ca cấp cứu trên lâm sàng.

Một số nguyên nhân dẫn đến sự bất thường của nồng độ oxy trong máu:

- ❖ Thiếu máu: Hemoglobin trong máu giảm thấp hơn bình thường. Khi không có tình trạng thiếu oxy máu, máy đo oxy dựa vào mạch đập sẽ cho kết quả chỉ số  $SpO_2$  chính xác khi nồng độ hemoglobin giảm xuống 2 - 3g/dL.
- ❖ Hoạt động trong môi trường bí khí, thiếu oxy như nhà máy, lò đốt, mỏ quặng...
- ❖ Ngộ độc CO: CO thay thế oxy ở vị trí gắn vào sắt trên phân tử Hemoglobin gây ra ngộ độc CO, làm tăng COHb (CO gắn vào Hemoglobin) và giảm

HbO<sub>2</sub> (oxy gắn vào Hemoglobin). Hiện tượng này làm giảm độ bão hòa của oxy trong máu.

### 1.2.3. Chỉ số chuyển động

Chỉ số chuyển động của cơ thể là một đại lượng phản ánh mức độ vận động thể chất trong một khoảng thời gian xác định, được xem là yếu tố trung gian quan trọng trong đánh giá sức khỏe, kiểm soát cân nặng và ước lượng năng lượng tiêu hao. Việc đo lường chỉ số này ngày càng trở nên thiết yếu khi các thiết bị đeo thông minh được ứng dụng rộng rãi trong chăm sóc sức khỏe cá nhân, thể thao và y học lâm sàng.

Dữ liệu chuyển động chủ yếu được thu nhận qua cảm biến gia tốc kế (accelerometer), đo sự thay đổi vận tốc của cơ thể theo ba trục không gian x, y và z. Từ đó, một đại lượng tổng hợp được gọi là Vector Magnitude (VM) được tính theo công thức:

$$VM = \sqrt{(x^2 + y^2 + z^2)}$$

Chỉ số VM có khả năng phản ánh cường độ vận động tổng thể trong thời gian thực và thường được biểu diễn dưới dạng “đếm mỗi phút” để phục vụ phân tích hành vi vận động và tính toán các chỉ số của cơ thể như chỉ số năng lượng tiêu hao.

Để lượng hóa hoạt động thể chất một cách chuẩn hóa, các chỉ số phổ biến như MET (Metabolic Equivalent of Task) và PAL (Physical Activity Level) thường được sử dụng song song với VM. Chỉ số MET là đơn vị chuyển hóa năng lượng tương đối, định nghĩa là mức năng lượng tiêu hao khi nghỉ ngơi, xấp xỉ 1 kcal/kg/giờ. Mỗi hoạt động cụ thể sẽ tương ứng với một giá trị MET đặc trưng, ví dụ đi bộ nhẹ nhàng khoảng 3.0 METs, chạy bộ tốc độ trung bình 8.0–10.0 METs, còn chạy cường độ cao hoặc thi đấu chuyên nghiệp có thể vượt 15 METs.

Chỉ số PAL được định nghĩa là tỷ lệ giữa tổng năng lượng tiêu hao hàng ngày (TDEE) và năng lượng chuyển hóa cơ bản (BMR):

$$PAL = \frac{TDEE}{BMR}$$

PAL cung cấp cái nhìn tổng quát về mức độ vận động của một cá nhân trong cả ngày. Dưới đây là bảng phân loại mức độ vận động theo chỉ số PAL:

Bảng 1.3. Bảng phân loại mức độ vận động theo chỉ số PAL

Mức độ hoạt động	Ví dụ nghề nghiệp	Chỉ số PAL
Ít vận động	Nhân viên văn phòng	1.40 – 1.69
Trung bình	Lao động phổ thông, kỹ thuật	1.70 – 1.99
Vận động nặng	Vận động viên, công nhân xây dựng	$\geq 2.00$

Chỉ số chuyển động cũng có thể được biểu diễn qua các đại lượng dễ tiếp cận như số bước chân mỗi ngày hoặc số phút vận động ở cường độ trung bình – cao. Các thiết bị đeo thông minh hiện nay thường tích hợp cảm biến gia tốc và con quay hồi chuyển (gyroscope), kết hợp với các thuật toán học máy để phân biệt loại hình vận động (đi, chạy, đứng, leo cầu thang...) và ước lượng lượng calo tiêu hao tương ứng.

Việc theo dõi chỉ số chuyển động có ý nghĩa quan trọng không chỉ trong việc duy trì lối sống lành mạnh mà còn giúp phát hiện sớm các nguy cơ sức khỏe liên quan đến lối sống tĩnh tại, chẳng hạn như béo phì, rối loạn chuyển hóa, và bệnh tim mạch. Khi được kết hợp với dữ liệu nhịp tim (HR), chỉ số chuyển động còn giúp nâng cao độ chính xác trong mô hình ước lượng năng lượng tiêu hao, phục vụ cho các mục tiêu như giảm cân, huấn luyện thể thao, hoặc phục hồi chức năng sau điều trị.

Mặc dù có nhiều ứng dụng thực tiễn, chỉ số chuyển động vẫn có những hạn chế nhất định như phụ thuộc vào vị trí đeo thiết bị, loại hoạt động, hoặc đặc điểm hình thể cá nhân. Do đó, xu hướng hiện nay là phát triển các mô hình cảm biến tích hợp và hiệu chỉnh cá nhân hóa để cải thiện độ tin cậy, đồng thời hỗ trợ ra quyết định tự động trong hệ thống chăm sóc sức khỏe thông minh.

#### 1.2.4. Đo lường năng lượng tiêu hao

Năng lượng tiêu hao (Energy Expenditure - EE) là chỉ số thể hiện tổng lượng năng lượng mà cơ thể sử dụng để duy trì các chức năng sống, thực hiện hoạt động thể chất và xử lý chuyển hóa. Việc đo lường EE một cách chính xác không chỉ hỗ



trợ quá trình kiểm soát cân nặng và xây dựng chế độ dinh dưỡng - vận động hợp lý, mà còn đóng vai trò cốt lõi trong các hệ thống thiết bị chăm sóc sức khỏe thông minh. Trong các thiết bị đeo hiện đại, hai chỉ số sinh lý phổ biến nhất được sử dụng để ước tính năng lượng tiêu hao là nhịp tim (Heart Rate - HR) và mức độ vận động (Acceleration - ACC). Sự kết hợp giữa hai chỉ số này mang lại độ chính xác cao hơn đáng kể so với khi sử dụng riêng lẻ.



Hình 1.3. Hoạt động thể chất

#### 1.2.4.1. Công thức ước tính năng lượng tiêu hao dựa trên nhịp tim (HR)

Nhịp tim là một đại lượng có quan hệ tương đối tuyến tính với mức tiêu thụ oxy ( $VO_2$ ), từ đó suy ra mức năng lượng tiêu hao. Một công thức phổ biến do Keytel et al. (2005) phát triển được ứng dụng rộng rãi như sau:

❖ Đối với nam giới:

$$EE_{(male)} = -55.0969 + 0.6309.HR + 0.988.W + 0.2017.A$$

❖ Đối với nữ giới:

$$EE_{(female)} = -20.4022 + 0.4472.HR - 0.1263.W + 0.074.A$$

Trong đó:

- ❖ HR: nhịp tim (lần/phút)
- ❖ W: cân nặng (kg)
- ❖ A: tuổi (năm)

Các công thức này đặc biệt hiệu quả khi đo lường EE trong hoạt động có cường độ trung bình đến cao, nhưng dễ chịu sai số trong điều kiện nghỉ ngơi hoặc

vận động nhẹ, do nhịp tim bị ảnh hưởng bởi nhiều yếu tố ngoại sinh như stress, caffeine, nhiệt độ môi trường...

#### 1.2.4.2. Công thức ước tính EE dựa trên vận động (ACC)

Các thiết bị đeo có tích hợp cảm biến gia tốc kế (accelerometer) có thể đo lường chuyển động của người dùng thông qua đại lượng vector magnitude (VM), phản ánh tổng lực gia tốc theo ba trục. Dựa trên nghiên cứu của Freedson et al. (1998), năng lượng tiêu hao có thể được ước tính theo công thức:

$$EE_{(ACC)} = 0.00094.VM + 0.1346.W - 0.0487$$

Trong đó:

- ❖ VM: vector magnitude (đơn vị: số lần/phút)
- ❖ W: cân nặng (kg)

Công thức này đặc biệt phù hợp với các hoạt động có tính di chuyển như đi bộ, chạy bộ. Tuy nhiên, nó có độ chính xác thấp khi đo các hoạt động tĩnh hoặc cử động hạn chế như nâng tạ, chèo thuyền hay bơi.

#### 1.2.4.3. Mô hình kết hợp HR và ACC

Để khắc phục nhược điểm khi sử dụng đơn lẻ từng chỉ số, các mô hình hiện đại có xu hướng kết hợp HR và ACC vào cùng một biểu thức thống nhất. Một ví dụ điển hình là mô hình được đề xuất trong nghiên cứu trên PLOS ONE (2015). Theo đó, năng lượng tiêu hao được tính theo công thức:

$$EE = 0.047.VM + 0.028(HR_T - HR_N) + 0.019.W - 0.013.A$$

Trong đó:

- ❖ HR<sub>T</sub>: nhịp tim đo được
- ❖ HR<sub>N</sub>: nhịp tim khi nghỉ
- ❖ W: cân nặng (kg)
- ❖ A: tuổi (năm)

Kết quả thử nghiệm cho thấy mô hình kết hợp này giảm sai số gốc trung bình (RMSE) xuống khoảng 20 kJ/ngày/kg với độ tương quan  $r = 0.67$ , tốt hơn nhiều so với việc chỉ sử dụng HR hoặc ACC riêng biệt. Như vậy, việc tích hợp

đồng thời mang lại độ chính xác cao hơn trong dự đoán tiêu hao năng lượng so với việc dự đoán cho từng nguồn dữ liệu riêng lẻ.

#### 1.2.4.4. So sánh hiệu quả giữa các mô hình

Bảng 1.4 trình bày các thông số cụ thể theo từng phương pháp đo năng lượng tiêu hao. Kết quả cho thấy, mô hình kết hợp cho kết quả tối ưu nhất về độ chính xác cũng như độ tin cậy trong môi trường sống tự nhiên. Tuy đem lại độ chính xác tương đối cao, các công thức kể trên vẫn chịu ảnh hưởng bởi yếu tố cá nhân như giới tính, thể trạng, thể lực nền và cả điều kiện môi trường. Ngoài ra, một số công thức vẫn cần hiệu chuẩn cá nhân ban đầu để cải thiện độ chính xác khi ứng dụng trong các thiết bị đeo thực tế. Hiện nay, các hãng sản xuất thiết bị như Fitbit, Apple Watch, Garmin... đã tích hợp mô hình học máy vào hệ thống đo EE, tận dụng đồng thời nhiều cảm biến để hiệu chỉnh mô hình dự đoán theo thời gian thực, giúp cải thiện đáng kể độ chính xác so với công thức tuyến tính truyền thống.

Bảng 1.4. Bảng so sánh hiệu quả giữa các mô hình

Phương pháp	RMSE (kJ/ngày/kg)	Tương quan (r)
HR đơn lẻ	34	0.58
ACC đơn lẻ	24	0.52
Kết hợp HR + ACC	20	0.67

#### 1.2.5. Thời gian ngủ

Giấc ngủ là một trong những nhu cầu sinh lý cơ bản và thiết yếu của con người, có vai trò quan trọng trong việc duy trì sức khỏe thể chất, tinh thần và năng lực nhận thức. Trong nhiều nghiên cứu khoa học, thời gian ngủ được xem là chỉ số phản ánh trực tiếp mức độ phục hồi chức năng của cơ thể, ảnh hưởng đến cân bằng nội môi, hệ miễn dịch, hoạt động thần kinh và hiệu quả lao động. Việc duy trì thời lượng ngủ hợp lý không chỉ giúp cải thiện hiệu suất làm việc ban ngày mà còn góp phần làm giảm nguy cơ mắc các bệnh lý mãn tính như tim mạch, rối loạn chuyển hóa, và trầm cảm. Theo báo cáo của Hội đồng Giấc ngủ Quốc gia Hoa Kỳ (NSF), nhu cầu về thời lượng ngủ thay đổi theo từng nhóm tuổi. Trẻ sơ sinh cần từ 14–17 giờ/ngày, trong khi người trưởng thành có khuyến nghị ngủ từ 7–9 giờ

mỗi đêm. Đối với thanh thiếu niên, thời lượng ngủ lý tưởng là từ 8–10 giờ mỗi đêm, tuy nhiên thực tế cho thấy hơn 50% học sinh trung học không đạt ngưỡng này. Sự thiếu ngủ kéo dài có thể gây ra trạng thái “nợ ngủ” – tức là lượng thời gian ngủ bị thiếu hụt tích lũy qua nhiều ngày – làm suy giảm hiệu suất nhận thức, khả năng ghi nhớ và điều hòa cảm xúc.

Nhiều nghiên cứu quy mô lớn đã chỉ ra mối tương quan giữa thời gian ngủ và nguy cơ tử vong theo mô hình hình chữ U: cả việc ngủ quá ít ( $\leq 5$  giờ/đêm) và ngủ quá nhiều ( $\geq 9$  giờ/đêm) đều làm tăng nguy cơ tử vong sớm, lần lượt ở mức 12% và 30%. Ngoài ra, thiếu ngủ còn làm rối loạn chuyển hóa glucose, giảm độ nhạy insulin và tăng nguy cơ viêm mạn tính mức độ thấp, trong khi ngủ quá nhiều thường đi kèm với sự suy giảm hoạt động thể chất và tăng tỷ lệ trầm cảm.

Bên cạnh thời gian ngủ, chất lượng giấc ngủ rất quan trọng vì một số lý do. Cũng giống như thở, ăn và uống, ngủ là nhu cầu thiết yếu của con người. Nó ảnh hưởng đến khả năng của một người để cảm thấy được nghỉ ngơi và tràn đầy năng lượng cho ngày hôm sau và giảm buồn ngủ ban ngày. Giấc ngủ chất lượng hỗ trợ sức khỏe tinh thần và thể chất và góp phần vào chất lượng cuộc sống chung. Về mặt sinh học, thời gian ngủ cần thiết để phục hồi cơ thể không chỉ đơn thuần được tính bằng số giờ ngủ mà còn phụ thuộc vào hiệu suất giấc ngủ – được định nghĩa là tỷ lệ giữa thời gian ngủ thực tế và tổng thời gian nằm trên giường. Một hiệu suất ngủ  $\geq 85\%$  được xem là lý tưởng, trong khi hiệu suất  $< 75\%$  có liên quan đến nguy cơ tăng huyết áp, rối loạn hormone và giảm khả năng học tập. Khi thời gian ngủ thực tế không đạt yêu cầu trong thời gian dài, cơ thể sẽ kích hoạt các cơ chế bù trừ như tăng cường sóng delta trong giấc ngủ sâu (N3), thay đổi nhiệt độ nền để phục hồi. Tại Việt Nam, khảo sát ở 5 tỉnh thành cho thấy tỷ lệ người lao động trí óc ngủ dưới 6,5 giờ/đêm chiếm đến 68%, trong khi nhóm lao động chân tay là 42%. Điều này cho thấy tác động của môi trường công việc và áp lực tinh thần đến thời lượng ngủ là rất rõ rệt. Bên cạnh đó, các yếu tố như môi trường sống, ánh sáng nhân tạo, thói quen sử dụng thiết bị điện tử trước giờ ngủ và chế độ dinh dưỡng cũng ảnh hưởng đáng kể đến thời gian và chất lượng giấc ngủ. Do đó, việc nghiên cứu và ứng dụng các thiết bị thông minh nhằm theo dõi thời gian ngủ, phân

tích chất lượng giấc ngủ và đưa ra cảnh báo sớm là một hướng tiếp cận mang tính thực tiễn cao. Các thiết bị hiện đại có khả năng sử dụng cảm biến gia tốc, nhiệt độ da, độ dẫn điện để xác định các pha ngủ và đánh giá hiệu suất giấc ngủ một cách khách quan. Từ các dữ liệu thu thập được, thiết bị có thể hỗ trợ người dùng điều chỉnh hành vi, xác lập thời gian biểu phù hợp và cải thiện giấc ngủ về cả thời lượng lẫn chất lượng.

Như vậy, thời gian ngủ không chỉ là một chỉ số sức khỏe đơn thuần mà còn là một thước đo toàn diện phản ánh trạng thái sinh lý, hành vi và môi trường sống của con người. Việc tích hợp tính năng theo dõi thời gian ngủ vào các thiết bị chăm sóc sức khỏe thông minh sẽ góp phần nâng cao chất lượng cuộc sống, phòng ngừa bệnh tật và hỗ trợ hiệu quả cho các hệ thống y tế cộng đồng trong tương lai.

### **1.3. Thiết bị y tế theo dõi sức khỏe**

#### *1.3.1. Giới thiệu*

Thiết bị theo dõi sức khỏe là các công cụ công nghệ được thiết kế để đo, giám sát và phân tích các chỉ số sức khỏe của người sử dụng, từ đó giúp họ duy trì lối sống lành mạnh và cải thiện sức khỏe. Những thiết bị này thường có khả năng theo dõi các yếu tố như hoạt động thể chất, nhịp tim, giấc ngủ, mức độ căng thẳng, huyết áp, và nhiều thông số sức khỏe khác. Các thiết bị theo dõi sức khỏe có thể giúp người dùng phát hiện sớm các vấn đề sức khỏe, từ đó đưa ra biện pháp cải thiện phù hợp.

Thiết bị theo dõi sức khỏe chia làm 2 loại là thông minh và thông thường. Thiết bị theo dõi sức khỏe thông thường thường có cấu tạo đơn giản, bao gồm các thành phần cơ bản phục vụ cho việc đo lường các chỉ số sức khỏe cơ bản. Đầu tiên, thiết bị sử dụng các cảm biến để thu thập dữ liệu, ví dụ như cảm biến đo nhịp tim, huyết áp, hoặc nhiệt độ cơ thể. Màn hình hiển thị giúp người dùng theo dõi kết quả đo lường, có thể là màn hình analog hoặc kỹ thuật số tùy thuộc vào loại thiết bị. Các thiết bị này có thể hoạt động nhờ nguồn năng lượng từ pin hoặc điện trực tiếp, giúp đảm bảo tính di động và tiện lợi. Một số thiết bị còn có cơ chế điều khiển đơn giản như nút bấm hoặc cần gạt để người dùng thao tác. Mặc dù không có nhiều tính năng phức tạp như các thiết bị thông minh, các thiết bị theo dõi sức

khỏe thông thường vẫn có thể giúp người dùng theo dõi và quản lý các chỉ số sức khỏe cơ bản một cách hiệu quả và dễ dàng.



Hình 1.4. Thiết bị theo dõi sức khỏe thông minh

Thiết bị theo dõi sức khỏe thông minh có cấu tạo phức tạp, tích hợp nhiều công nghệ tiên tiến để theo dõi và phân tích các chỉ số sức khỏe của người dùng. Đầu tiên, thiết bị sử dụng các cảm biến như cảm biến nhịp tim, gia tốc kế, con quay hồi chuyển, cảm biến oxy trong máu, huyết áp, nhiệt độ cơ thể, và điện tâm đồ để thu thập dữ liệu về các chỉ số sinh lý. Dữ liệu từ các cảm biến này sẽ được bộ vi xử lý trong thiết bị xử lý và chuyển đổi thành các thông tin hữu ích cho người dùng. Thiết bị thường được trang bị màn hình hiển thị, có thể là OLED hoặc LCD, để người dùng theo dõi kết quả đo lường. Ngoài ra, thiết bị còn có bộ nhớ để lưu trữ dữ liệu và kết nối không dây như Bluetooth hoặc wifi để đồng bộ hóa với các ứng dụng trên điện thoại. Nguồn năng lượng của thiết bị thường là pin sạc lithium-ion, cung cấp năng lượng cho các tính năng của thiết bị. Vỏ ngoài của thiết bị được làm từ vật liệu bền chắc như nhựa hoặc kim loại, và dây đeo thường bằng silicone hoặc vải để tạo sự thoải mái khi sử dụng. Ngoài các tính năng cơ bản, nhiều thiết bị còn tích hợp GPS để theo dõi hoạt động ngoài trời, tính năng nhận dạng giấc ngủ để theo dõi chất lượng giấc ngủ, và cảnh báo sức khỏe để thông báo người dùng về các vấn đề sức khỏe tiềm ẩn. Tất cả các thành phần này phối hợp chặt chẽ, giúp thiết bị cung cấp thông tin sức khỏe chính xác và hỗ trợ người dùng cải thiện sức khỏe một cách hiệu quả.

### 1.3.2. Một số loại thiết bị theo dõi sức khỏe phổ biến

#### 1.3.2.1. Đồng hồ thông minh - Smartwatches



Hình 1.5. Đồng hồ thông minh.

Các đồng hồ thông minh như Apple Watch, Samsung Galaxy Watch hay Fitbit cung cấp nhiều tính năng theo dõi sức khỏe, bao gồm:

- ❖ Theo dõi nhịp tim: Đo lường nhịp tim liên tục và khi tập thể thao.
- ❖ Theo dõi giấc ngủ: Phân tích chất lượng giấc ngủ, giúp người dùng cải thiện thói quen ngủ.
- ❖ Theo dõi hoạt động thể chất: Đo bước đi, quãng đường di chuyển, lượng calo tiêu thụ.
- ❖ Tính năng ECG (Electrocardiogram): Đo điện tâm đồ để phát hiện các vấn đề về tim mạch.

#### 1.3.2.2. Vòng đeo tay thể thao - Fitness Bands

Các vòng đeo tay như Fitbit Charge, Xiaomi Mi Band thường có thiết kế nhẹ nhàng, đơn giản hơn đồng hồ thông minh và tập trung vào tính năng cơ bản:

- ❖ Đếm bước đi và đo quãng đường: Theo dõi mức độ vận động hàng ngày.
- ❖ Theo dõi giấc ngủ: Cung cấp báo cáo về các giai đoạn giấc ngủ.
- ❖ Đo nhịp tim: Giám sát nhịp tim trong suốt quá trình vận động.

#### 1.3.2.3. Máy đo huyết áp tự động

Thiết bị này cho phép người dùng đo huyết áp tại nhà một cách đơn giản. Các máy đo huyết áp tự động có thể kết nối với điện thoại thông minh để lưu trữ và phân tích dữ liệu theo thời gian.

#### 1.3.2.4. Máy đo nồng độ oxy trong máu - Pulse Oximeters

Thiết bị này đo mức độ oxy trong máu giúp phát hiện các vấn đề về hô hấp hoặc tim mạch. Thiết bị này rất quan trọng trong việc theo dõi sức khỏe của những người có bệnh lý liên quan đến tim và phổi.



Hình 1.6. Máy đo huyết áp tự động và máy đo nồng độ oxy trong máu

#### 1.3.2.5. Thiết bị theo dõi giấc ngủ



Hình 1.7. Oura Ring

Các thiết bị như Oura Ring hay tính năng giám sát giấc ngủ trên đồng hồ thông minh có thể phân tích các giai đoạn giấc ngủ. Thiết bị này cũng thu thập và phân tích dữ liệu sức khỏe chính xác và toàn diện, bao gồm nhịp tim, mức độ căng thẳng và nhiệt độ cơ thể, cung cấp những thông tin và gợi ý hữu ích để người dùng cải thiện sức khỏe.

#### 1.3.3. Ưu điểm và nhược điểm

Ưu điểm:

- ❖ Quản lý sức khỏe chủ động: Giúp người dùng theo dõi và điều chỉnh các yếu tố như hoạt động thể chất, giấc ngủ, và tình trạng tim mạch.



- ❖ Dự đoán và phát hiện bệnh sớm: Các chỉ số như nhịp tim, huyết áp hay mức độ oxy trong máu có thể giúp phát hiện sớm các dấu hiệu bất thường, từ đó người dùng có thể can thiệp kịp thời.
- ❖ Cải thiện thói quen sống: Các báo cáo chi tiết về mức độ vận động và giấc ngủ giúp người dùng điều chỉnh lối sống, tránh các thói quen có hại cho sức khỏe.
- ❖ Tăng cường động lực luyện tập: Các tính năng như mục tiêu hàng ngày hoặc thử thách giúp người dùng duy trì động lực trong việc tập luyện thể thao.  
Nhược điểm và thách thức:
- ❖ Độ chính xác: Mặc dù các thiết bị này ngày càng chính xác, nhưng không phải lúc nào kết quả đo lường cũng hoàn toàn giống với các thiết bị y tế chuyên dụng.
- ❖ Tiêu thụ năng lượng: Các thiết bị như đồng hồ thông minh thường cần phải sạc thường xuyên, đôi khi gây bất tiện.
- ❖ Tính khả dụng và đồng bộ hóa: Không phải tất cả các thiết bị đều tương thích với mọi hệ điều hành, và có thể gặp khó khăn trong việc đồng bộ hóa dữ liệu với các ứng dụng di động.

#### 1.3.4. Xu hướng phát triển

Thiết bị theo dõi sức khỏe thông minh đang ngày càng phát triển mạnh mẽ nhờ vào sự tiến bộ nhanh chóng của công nghệ và nhu cầu ngày càng cao trong việc chăm sóc sức khỏe cá nhân. Các thiết bị này không chỉ giúp người dùng theo dõi sức khỏe mà còn giúp cải thiện chất lượng cuộc sống và tăng cường hiệu quả chăm sóc sức khỏe chủ động. Dưới đây là một số xu hướng nổi bật trong lĩnh vực thiết bị theo dõi sức khỏe:

- ❖ Tích hợp công nghệ trí tuệ nhân tạo: Tập trung phát triển theo hướng:
  - Phân tích dữ liệu sức khỏe thông minh: Các thiết bị theo dõi sức khỏe đang ngày càng sử dụng trí tuệ nhân tạo (AI) để phân tích dữ liệu thu thập được từ người dùng và đưa ra các khuyến nghị về cách cải thiện sức khỏe. Ví dụ, AI có thể phân tích các mẫu giấc ngủ, nhận diện các xu hướng trong chỉ số

tim mạch, từ đó cung cấp lời khuyên cá nhân hóa về dinh dưỡng, luyện tập và các thói quen sống.

- Dự đoán bệnh lý: AI có khả năng phân tích dữ liệu sức khỏe từ nhiều người dùng để nhận diện sớm các dấu hiệu của các bệnh lý tiềm ẩn như tim mạch, tiêu đường, hoặc bệnh về hô hấp.



Hình 1.8. Đồng hồ thông minh tích hợp trí tuệ nhân tạo

- ❖ Tăng cường tính năng chăm sóc sức khỏe tâm lý: Những năm gần đây, người ta đã chú trọng hơn đến sức khỏe tâm lý, đặc biệt là trong bối cảnh dịch bệnh và áp lực công việc ngày càng tăng. Các thiết bị theo dõi sức khỏe đang dần tích hợp khả năng đo lường và theo dõi mức độ căng thẳng, lo âu và các chỉ số tâm lý khác.
- ❖ Ứng dụng công nghệ cảm biến sinh học (biometrics): Các thiết bị như đồng hồ thông minh không chỉ đo nhịp tim mà còn có thể nhận diện các dấu hiệu tâm lý thông qua các thay đổi sinh học (như huyết áp, nhịp thở) và cung cấp các bài tập thư giãn hoặc thiền định để giúp người dùng giảm stress.

## 1.4. Sơ lược về Internet vạn vật - IoT

### 1.4.1. Giới thiệu

Internet vạn vật (Internet of Things - IoT) là mạng lưới vạn vật được kết nối với nhau bằng Internet, được đưa ra bởi Kevin Ashton vào năm 1999 - người sáng

lập ra trung tâm Auto - ID tại đại học MIT. Thuật ngữ IoT được Liên minh Viễn thông Quốc tế (ITU) giới thiệu trong báo cáo Internet của ITU năm 2005. Đây là một hệ thống phức tạp, bởi nó bao gồm một lượng lớn các đường liên kết giữa máy móc, thiết bị và dịch vụ với nhau thông qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính.



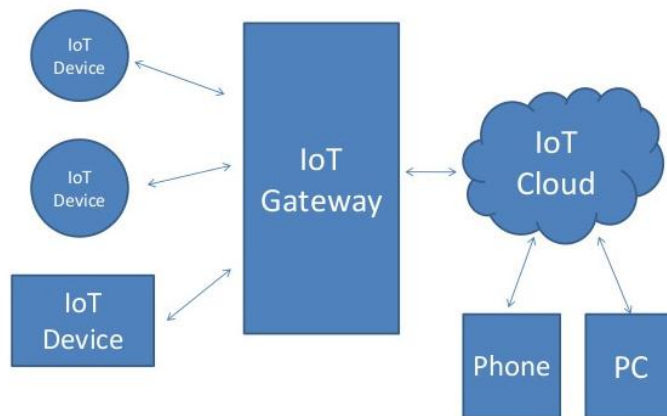
Hình 1.9. Hệ thống IoT.

Các thiết bị IoT bao gồm những đồ vật được gắn thêm cảm biến để thu thập dữ liệu về môi trường xung quanh, thu phát tín hiệu, nhận dữ liệu và ra lệnh điều khiển các thiết bị khác được kết nối qua mạng Internet, Wifi, Bluetooth, mạng viễn thông băng rộng (3G, 4G, 5G), hồng ngoại.... Các thiết bị này có thể thực hiện việc trao đổi dữ liệu với nhau được kết nối với ứng dụng để điều khiển nó trực tiếp hoặc gián tiếp. Ngoài ra chúng còn có thể thu thập dữ liệu từ các thiết bị khác và xử lý dữ liệu ngay tại thiết bị hoặc gửi dữ liệu tới các máy chủ tập trung hay các ứng dụng dựa trên đám mây để xử lý những dữ liệu này. Người dùng có thể kiểm soát mọi đồ vật của mình qua mạng chỉ bằng một vài thao tác đơn giản trên thiết bị thông minh như điện thoại thông minh, máy tính bảng, máy tính cá nhân hay thậm chí là một chiếc đồng hồ thông minh nhỏ bé trên tay mà không bị giới hạn không gian hay thời gian.

Hiện nay, IoT không còn là một dự đoán nữa mà là một cuộc cách mạng ứng dụng công nghệ mới đang diễn ra như vũ bão trên toàn thế giới. Với khoảng 14 tỷ thiết bị IoT trên thế giới ở thời điểm hiện tại, ước tính đạt 27 tỷ thiết bị vào năm

2025, thế giới sẽ chứng kiến các thiết bị IoT được triển khai rộng rãi với nhiều công nghệ và nhiều ứng dụng mới trên nhiều mặt của đời sống kinh tế - xã hội. IoT đã trở thành một trong những công nghệ quan trọng, giúp mọi người sống và làm việc thông minh hơn, chủ động hoàn toàn trong cuộc sống. Có thể thấy, tiềm năng ứng dụng của IoT trải rộng trên nhiều lĩnh vực và có xu hướng đột phá trong tương lai, những ứng dụng của IoT ngày càng phong phú và thiết thực đối với cuộc sống trong xã hội hiện đại, mang lại trải nghiệm phong phú và tiện nghi cho người sử dụng.

#### 1.4.2. Cấu trúc của hệ thống IoT



Hình 1.10. Cấu trúc hệ thống IoT.

Một hệ thống IoT về cơ bản cấu tạo gồm 4 phần: Thiết bị (Things), trạm kết nối (Gateways), hạ tầng mạng và điện toán đám mây (Network and Cloud) và các lớp tạo cung cấp dịch vụ (Services and Solution Layers).

- ❖ **Thiết bị (Things):** Đây là những thiết bị điện tử, máy móc hoặc cảm biến mà bạn muốn kết nối vào mạng IoT. Chúng có thể là điện thoại thông minh, máy tính, máy chủ, cảm biến nhiệt độ, đèn, thiết bị đo lường, xe cộ và nhiều loại thiết bị khác được kết nối trực tiếp thông qua băng tầng mạng không dây và truy cập vào Internet. Giải pháp IoT giúp các thiết bị thông minh được sàng lọc, kết nối và quản lý dữ liệu một cách cục bộ, còn các thiết bị chưa thông minh thì có thể kết nối được thông qua các trạm kết nối.
- ❖ **Trạm kết nối (Gateways):** Một trong những trở ngại chính trong khi triển khai IoT đó là gần 85% các thiết bị phần cứng đã không được thiết kế để có thể kết nối với Internet và không chia sẻ dữ liệu với điện toán đám mây. Các

trạm kết nối đóng vai trò là một trung gian trực tiếp, cho phép các vật dụng này kết nối với điện toán đám mây một cách được bảo mật và dễ dàng được quản lý.

- ❖ **Hạ tầng mạng và điện toán đám mây (Network and Cloud):** Internet là một hệ thống toàn cầu của nhiều mạng IP được kết nối với nhau và liên kết với hệ thống máy tính. Cơ sở hạ tầng mạng này bao gồm thiết bị định tuyến, trạm kết nối, thiết bị tổng hợp, thiết bị lặp và nhiều thiết bị khác có thể kiểm soát lưu lượng dữ liệu lưu thông và cũng được kết nối đến mạng lưới viễn thông và cáp được triển khai bởi các nhà cung cấp dịch vụ. Các công nghệ mạng được sử dụng trong IoT gồm wifi, Bluetooth, Zigbee, LTE, LoRaWAN và nhiều công nghệ khác. Các trung tâm dữ liệu và hạ tầng điện toán đám mây cung cấp lộ trình chia sẻ và lưu trữ dữ liệu IoT.
- ❖ **Các lớp tạo và cung cấp dịch vụ (Services and Solution Layers):** Lớp này gồm kiến trúc trung tâm dữ liệu cung cấp khả năng triển khai, quản lý và bảo vệ các ứng dụng, thông tin, cơ sở hạ tầng máy chủ một cách nhanh chóng, an toàn và tin cậy. Ngoài ra nó còn cung cấp các dịch vụ mở rộng cho cơ sở hạ tầng, đảm bảo sự liên kết các nguồn lực của trung tâm dữ liệu với các yêu cầu của ứng dụng một cách nhanh chóng và an toàn.

#### 1.4.3. Ứng dụng

Tính linh hoạt của IoT làm cho nó trở thành công cụ đắc lực đối với rất nhiều doanh nghiệp, tổ chức, cơ quan chính phủ, ... Dưới đây là một số ví dụ rõ nét nhất về ứng dụng của Internet vạn vật:

- ❖ **Ứng dụng IoT trong nông nghiệp thông minh**

Đối với việc trồng trọt trong nhà kính, IoT biến việc giám sát, quản lý các điều kiện khí hậu vi mô thành hiện thực, từ đó giúp tăng sản lượng. Đối với việc trồng cây ngoài trời, thiết bị sử dụng công nghệ IoT có thể cảm nhận được độ ẩm, chất dinh dưỡng của đất, kết hợp với dữ liệu thời tiết giúp kiểm soát tốt hơn hệ thống tưới tiêu và bón phân thông minh.

- ❖ **Ứng dụng IoT trong ngôi nhà thông minh**

Nhà thông minh đảm nhận những việc như kích hoạt và kiểm soát môi trường để ngôi nhà luôn ở trạng thái thoải mái nhất cho người ở. Bữa tối được nấu từ xa, thức ăn đã sẵn sàng trước khi chủ nhân quay trở về. An ninh cũng dễ dàng hơn nhờ khả năng điều khiển thiết bị theo dõi, cảnh báo và hệ thống điện từ xa, đồng thời kích hoạt khóa thông minh, cho phép những người thích hợp ra vào nhà ngay cả khi họ không có chìa khóa. Ngoài ra, bằng việc ứng dụng IoT, các thiết bị đeo bao gồm điện thoại, đồng hồ, kính thực tế ảo, máy theo dõi sức khỏe, ... đã giúp người dùng cải thiện khả năng và phương thức giải trí, kết nối mạng, theo dõi sức khỏe và luyện tập thể chất.

#### ❖ Ứng dụng IoT trong y tế



Hình 1.11. IoT ứng dụng trong y tế

Đầu tiên và quan trọng nhất, thiết bị đeo IoT cho phép bệnh viện, chuyên gia theo dõi sức khỏe của bệnh nhân tại nhà. Việc đưa cảm biến IoT vào các thiết bị quan trọng khác cũng giúp giảm tình trạng hỏng hóc và tăng độ tin cậy khi khám chữa bệnh. Điều này góp phần giảm thời gian điều trị nội trú trong bệnh viện nhờ cung cấp thông tin theo thời gian thực, chính xác đến từng phút, đảm bảo bệnh nhân được chăm sóc tốt khi điều trị ngoại trú. Trong bệnh viện, giường thông minh thông báo cho nhân viên y tế về tình trạng sẵn có nhằm cắt giảm thời gian chờ đợi. Quá trình chăm sóc người bệnh, đặc biệt là người cao tuổi nhờ vậy mà trở nên thuận tiện, thoải mái hơn. Ngoài việc theo dõi thời gian thực tại nhà, cảm



biến cũng có thể xác định bệnh nhân có gặp phải những vấn đề sức khỏe nghiêm trọng hơn như bị ngã hay bị đau tim hay không.

#### ❖ Ứng dụng IoT trong sản xuất

Công nghệ RFID và GPS có thể giúp doanh nghiệp theo dõi sản phẩm từ khi bắt đầu sản xuất trên dây chuyền ở nhà máy đến vị trí bày bán tại cửa hàng đích, toàn bộ chuỗi cung ứng từ đầu đến cuối. Những cảm biến này thu thập thông tin về thời gian di chuyển, tình trạng sản phẩm và điều kiện môi trường mà sản phẩm phải chịu. Cảm biến được gắn vào thiết bị trong nhà máy cũng giúp xác định "nút thắt cổ chai" (Bottleneck) trong dây chuyền, do đó giảm thiểu thời gian sản xuất, tránh lãng phí. Một số cảm biến khác cũng có thể theo dõi hiệu suất của máy, dự đoán khi nào thiết bị sẽ cần bảo trì, do đó ngăn ngừa hỏng hóc nặng nề, gây tốn kém nhiều chi phí.

#### ❖ Ứng dụng IoT trong bán lẻ

Công nghệ IoT có rất nhiều thứ để cung cấp cho thế giới bán lẻ. Số liệu bán hàng mua sắm trực tuyến hoặc tại cửa hàng được kiểm soát tự động. Phần lớn điều này phụ thuộc vào RFID, vốn đã được sử dụng rộng rãi trên toàn thế giới. Thêm vào đó, sự ra đời của mua sắm trực tuyến đã làm giảm nhu cầu đối với cơ sở bán lẻ truyền thống. Tuy nhiên, IoT có thể giúp phân tích lưu lượng truy cập của trung tâm mua sắm, các cửa hàng nằm trong trung tâm thương mại có thể thực hiện những điều chỉnh cần thiết, nhằm nâng cao trải nghiệm mua sắm của khách hàng, đồng thời giảm chi phí. Hơn nữa, IoT giúp nhà bán lẻ tiếp cận mục tiêu khách hàng dựa trên những lần mua hàng trước đây. Trang bị thông tin được cung cấp qua IoT, nhà bán lẻ có thể tạo chương trình khuyến mãi được cá nhân hóa cho khách hàng trung thành của họ, loại bỏ chương trình tiếp thị đại trà tốn kém mà không có nhiều cơ hội thành công. Phần lớn các chương trình khuyến mãi này có thể được thực hiện thông qua điện thoại thông minh của khách hàng.

#### ❖ Ứng dụng IoT trong giao thông vận tải

Đến thời điểm này, những tiến bộ đạt được với ô tô tự lái đã được nhiều người biết đến. Nhưng đó chỉ là một phần của tiềm năng to lớn trong lĩnh vực giao thông vận tải. GPS, một ví dụ khác của IoT, đang được sử dụng để giúp các công

ty vận tải vạch ra tuyến đường nhanh hơn, hiệu quả hơn cho xe tải vận chuyển hàng hóa, nhờ đó đẩy nhanh thời gian giao hàng. Ngoài ra, nhà quy hoạch thành phố cũng có thể sử dụng dữ liệu GPS đó nhằm xác định mô hình giao thông, nhu cầu chỗ đậu xe cũng như xây dựng và bảo trì đường bộ.



Hình 1.12. Thiết bị IoT trong nhà thông minh giúp cuộc sống dễ dàng hơn.

Tóm lại, IoT đã trở thành một trong những công nghệ quan trọng. IoT giúp mọi người sống và làm việc thông minh hơn, cũng như chủ động hoàn toàn trong cuộc sống. Do đó, IoT là một trong những công nghệ quan trọng nhất của cuộc sống hàng ngày với rất nhiều ứng dụng trong thế giới thực, trải dài trên nhiều lĩnh vực, hướng tới nhiều đối tượng từ cá nhân cho tới doanh nghiệp, nhà nước và đa quốc gia.



## CHƯƠNG 2. CÔNG NGHỆ THIẾT KẾ

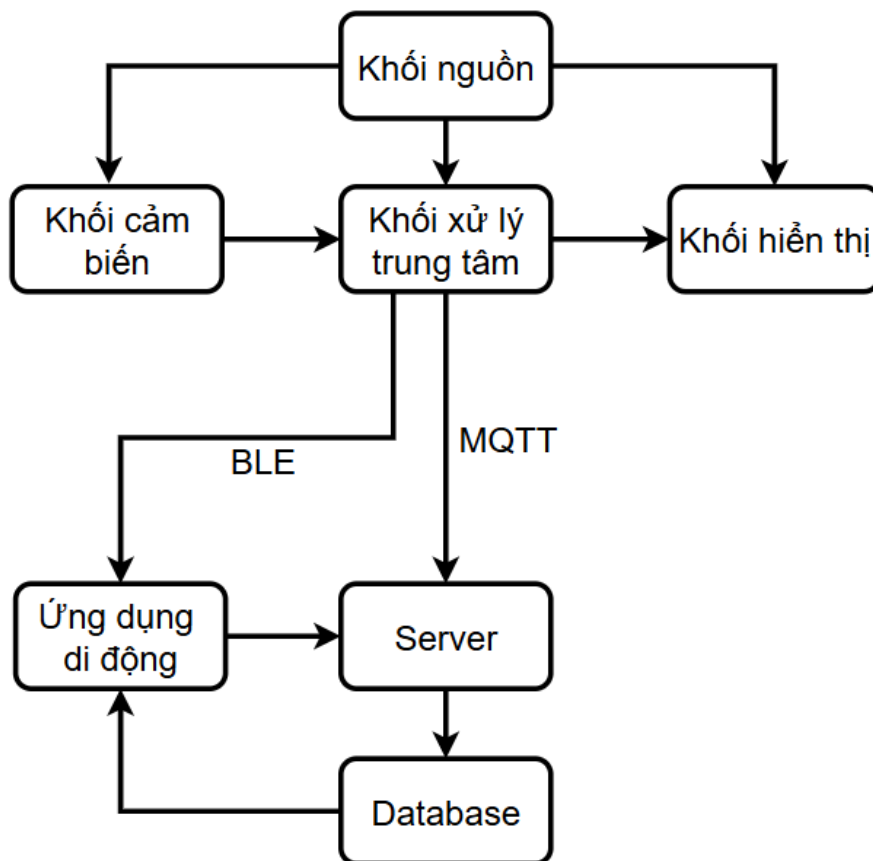
### 2.1. Giải pháp thiết kế

#### 2.1.1. Mục tiêu thiết kế

Thiết kế và chế tạo thiết bị thông minh hỗ trợ chăm sóc sức khỏe có các chức năng như sau:

- ❖ Thiết kế thiết bị thông minh hỗ trợ theo dõi các thông số sức khỏe cơ bản.
- ❖ Theo dõi giấc ngủ giúp cải thiện chất lượng giấc ngủ.
- ❖ Kết hợp truyền tin không dây để định vị.

#### 2.1.2. Sơ đồ khối



Hình 2.1. Sơ đồ khối

Sơ đồ khối của thiết bị được trình bày trên hình 2.1 với chức năng từng khối cụ thể như sau:

- ❖ Khối xử lý trung tâm: ESP32-S3-MINI-1-N8, có chức năng giao tiếp với tất cả các khối trong sơ đồ, truyền/nhận dữ liệu và điều khiển thiết bị hoạt động theo yêu cầu.

- ❖ Khối cảm biến: Đo các chỉ số sức khỏe và chuyển động cơ bản.
- ❖ Khối nguồn: Cung cấp điện ổn định cho toàn bộ hệ thống từ pin hoặc nguồn ngoài.
- ❖ Khối hiển thị: Hiển thị dữ liệu đo được lên màn hình.
- ❖ Server: Nhận MQTT, xử lý và chuyển tiếp vào Database.
- ❖ Database: Lưu trữ dài hạn dữ liệu sức khỏe của người dùng.
- ❖ Ứng dụng trên điện thoại: Giao diện để hiển thị các thông tin biểu đồ, lịch sử, cấu hình và hỗ trợ truyền tin.

## 2.2. Công nghệ phần cứng

### 2.2.1. ESP32-S3-MINI-1



Hình 2.2. Module ESP32-S3-MINI-1-N8

Module ESP32-S3-MINI-1-N8 là phiên bản thu gọn, hiệu năng cao của dòng ESP32-S3 do Espressif Systems sản xuất. Được đóng gói với kích thước chỉ  $18 \times 25,5$  mm, module này kết hợp vi xử lý lõi kép Tensilica® Xtensa® LX7 đạt tốc độ tối đa 240 MHz, Flash SPI 8 MB và SRAM 2 MB. Tích hợp anten PCB cùng khả năng kết nối Wi-Fi 802.11 b/g/n và Bluetooth Low Energy 5.0 (hỗ trợ advertising extension, long range), ESP32-S3-MINI-1-N8 lý tưởng cho các thiết bị IoT đeo tay, cảm biến di động, gateway BLE-to-MQTT và các ứng dụng nhúng tiêu thụ thấp.

Ưu điểm nổi bật:

- ❖ Đa nhiệm mạnh mẽ: Lỗi kép xử lý song song, chạy các tác vụ cảm biến, giao tiếp và xử lý dữ liệu mượt mà.
- ❖ Tiết kiệm năng lượng: Pin-friendly với chế độ deep-sleep chỉ  $\sim 5 \mu\text{A}$ , kéo dài thời gian hoạt động của thiết bị.
- ❖ Bộ nhớ dư dả: Cho phép cài đặt firmware phức hợp, OTA, và lưu trữ tạm thời dữ liệu lớn.
- ❖ Ngoại vi phong phú: Hỗ trợ đa dạng giao tiếp (I<sup>2</sup>C, SPI, UART, I<sup>2</sup>S, RMT, CAN, SDIO) cùng ADC, DAC, touch sensor.
- ❖ Thiết kế SMT dễ dàng: Pad 0,5 mm, tương thích dây chuyên SMT, giảm chi phí sản xuất.

Module ESP32-S3-MINI-1-N8 được thiết kế với hệ thống thông số kỹ thuật toàn diện, đáp ứng nhiều yêu cầu ứng dụng chuyên nghiệp:

- ❖ Vi xử lý & Bộ nhớ:
  - CPU: Dual-core Tensilica® Xtensa® LX7, tối đa 240 MHz cho mỗi lõi, hỗ trợ lệnh vector cho AI acceleration.
  - Flash: 8 MB SPI NOR flash, hỗ trợ secure boot và encrypted flash.
  - SRAM: 2 MB internal SRAM, đủ khả năng chạy các ứng dụng thời gian thực và giao thức mạng.
- ❖ Kết nối không dây:
  - Wi-Fi: 802.11 b/g/n trên băng tần 2,4 GHz, hỗ trợ HT40, WPA2.
  - Bluetooth: BLE 5.0, quảng bá extended advertising, long range PHY, đảm bảo phủ sóng rộng và tiết kiệm năng lượng.
- ❖ Giao tiếp ngoại vi:
  - GPIO: Tối đa 23 chân, hỗ trợ PWM, input capture và output compare.
  - ADC: 5 kênh 12-bit, tốc độ sampling đến 20 kSPS, phù hợp đo tín hiệu analog từ cảm biến.
  - DAC: 2 kênh 8-bit, tạo tín hiệu analog điều khiển ngoại vi.
  - I<sup>2</sup>C: 2 bus, tốc độ chuẩn up to 400 kHz.

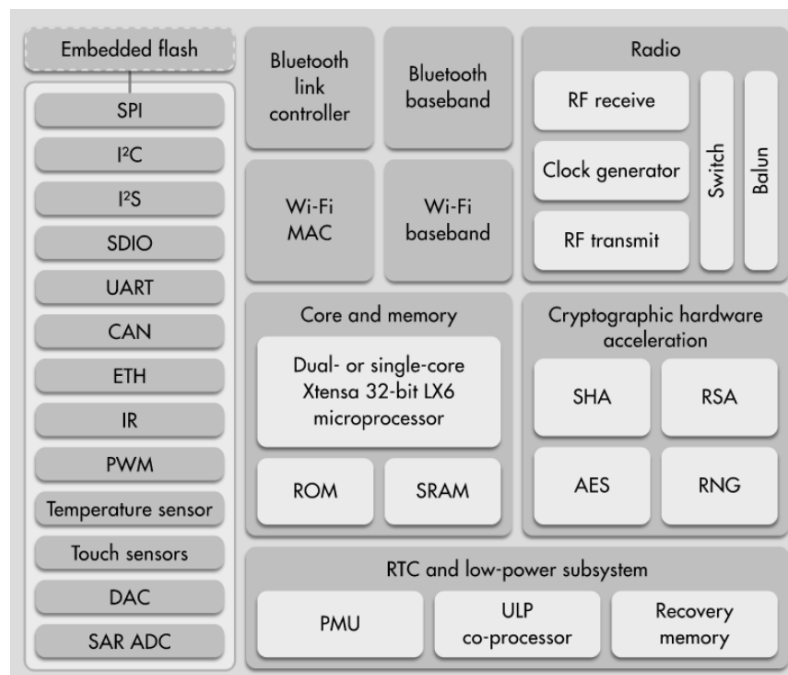
- SPI: 3 bus, hỗ trợ chế độ full-duplex, tốc độ up to 80 MHz.
- UART: 2 cổng, hỗ trợ RTS/CTS hardware flow control, baud rate lên đến 5 Mbps.
- I<sup>2</sup>S, RMT, CAN, SDIO: Hỗ trợ các ứng dụng âm thanh, remote control, công nghiệp và thẻ nhớ SD.

❖ Nguồn & Tiêu thụ:

- Điện áp hoạt động: 3,0 – 3,6 V.
- Deep-sleep: ~5  $\mu$ A.
- Active-mode: 80–240 mA tùy mức CPU và RF.

❖ Cơ khí và môi trường:

- Kích thước: 18 × 25,5 mm, pad pitch 0,5 mm.
- Nhiệt độ hoạt động: –40 °C đến 85 °C.
- Độ ẩm: 5 % – 95 % RH không ngưng tụ.

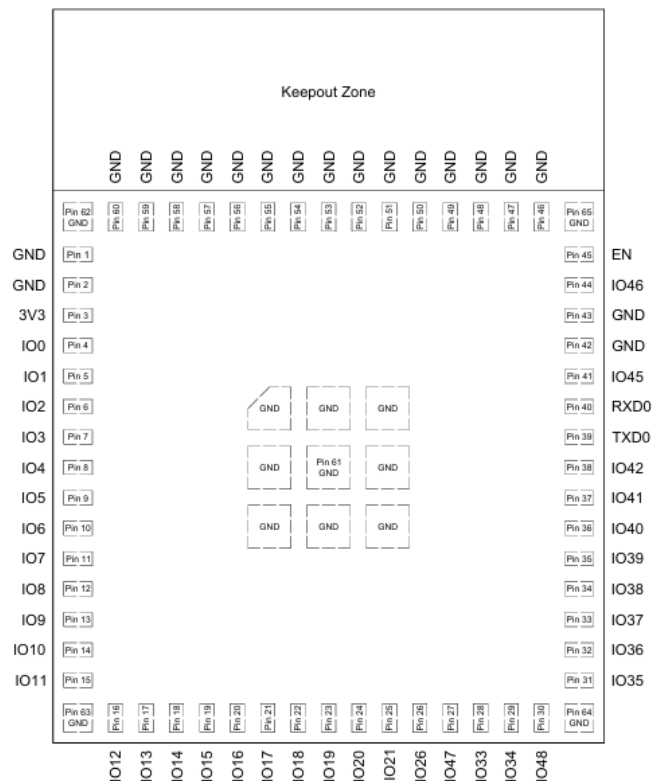


Hình 2.3. Cấu hình của module ESP32-S3-MINI-1-N8

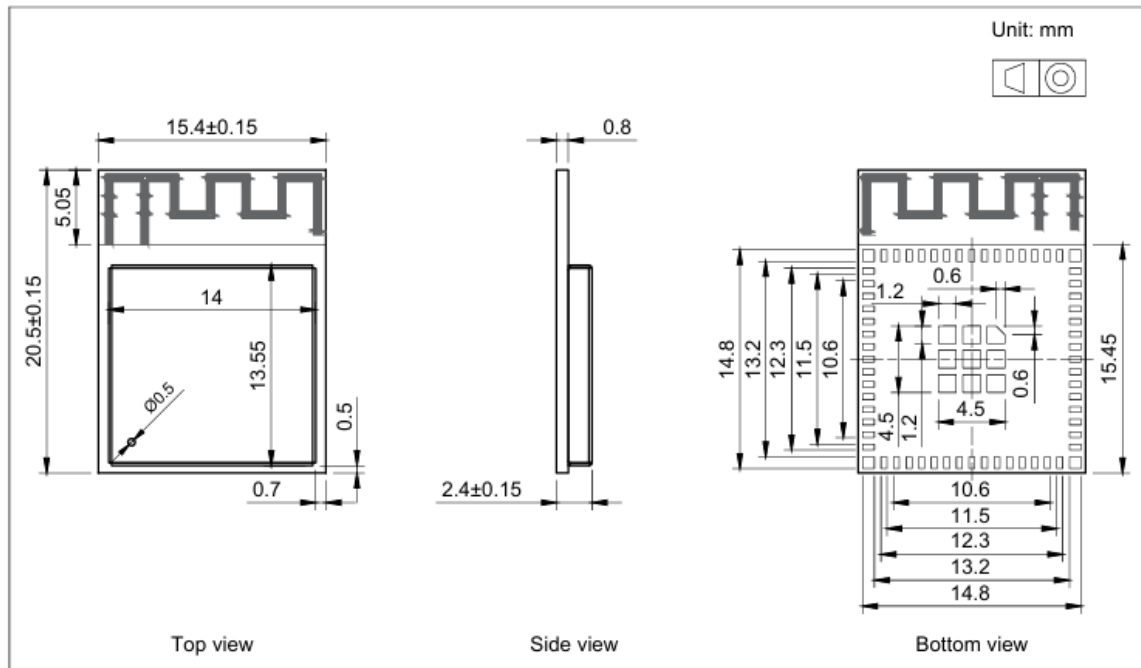
Sơ đồ chân và kích thước dùng trong thiết kế mạch in của vi điều khiển được trình bày trên hình 2.4 và hình 2.5. Chức năng của các chân:

- ❖ Nguồn và Flash SPI: Chân VCC cung cấp điện áp 3,3 V ổn định và GND đảm bảo kết nối đất chung; các chân GPIO6–GPIO11 chỉ phục vụ giao tiếp với Flash SPI 8 MB, không dùng cho I/O khác.

- ❖ GPIO đa năng: GPIO0–GPIO5, GPIO12–GPIO15, GPIO17–GPIO27 và GPIO32–GPIO39 hỗ trợ digital I/O, PWM, ADC 12-bit, DAC 8-bit, cảm biến điện dung touch sense và ngắt ngoài (external interrupt).
- ❖ GPIO chuyên biệt: GPIO34–GPIO39 chỉ nhận tín hiệu (input-only) và không tích hợp pull-up/pull-down; GPIO17 và GPIO18 là hai kênh DAC để xuất tín hiệu analog.
- ❖ Giao tiếp I<sup>2</sup>C và SPI: Bus I<sup>2</sup>C mặc định trên SDA=GPIO21, SCL=GPIO22; bus SPI VSPI với SCLK=GPIO18, MOSI=GPIO23, MISO=GPIO19, CS=GPIO5 cho tốc độ lên đến 80 MHz.
- ❖ UART và các giao thức khác: UART0 (TX0=GPIO1, RX0=GPIO3) và UART1 (TX1=GPIO10, RX1=GPIO9) có thể cấu hình lại qua GPIO matrix; module còn hỗ trợ I<sup>2</sup>S, RMT, CAN và SDIO cho âm thanh, điều khiển từ xa, ứng dụng công nghiệp và thẻ nhớ SD.
- ❖ Touch sensor và RTC\_GPIO: 10 kênh touch sensor trên các chân GPIO0, 2, 4, 12–15, 27, 32–33; bất kỳ chân RTC\_GPIO nào cũng có thể dùng để đánh thức module từ chế độ deep-sleep.



Hình 2.4. Sơ đồ chức năng các chân của ESP32-S3-MINI-1-N8



Hình 2.5. Kích thước của ESP32-S3-MINI-1-N8

### 2.2.2. Cảm biến MAX30102



Hình 2.6. Cảm biến Max30102

Cảm biến nhịp tim và oxy trong máu MAX30102 được sử dụng để đo nhịp tim và nồng độ Oxy trong máu, thích hợp cho nhiều ứng dụng liên quan đến y sinh. cảm biến sử dụng phương pháp đo quang phổ biến hiện nay với thiết kế và chất liệu mắt đo chuyên biệt từ chính hãng Maxim cho độ chính xác và độ bền cao, nó bao gồm đèn LED bên trong, bộ tách sóng quang, các bộ phận quang học và các thiết bị điện tử, có tiếng ồn thấp với khả năng loại bỏ ánh sáng xung quanh. Cảm biến sử dụng giao tiếp I2C với bộ thư viện sẵn có trên Arduino nên rất dễ sử dụng. Một số ưu điểm của cảm biến:

- ❖ Dòng tải cực thấp, 600μA (chế độ đo) và 0,7μA (chế độ chờ). Do đó, nó là một lựa chọn tuyệt vời để sử dụng trong các thiết bị đeo được như đồng hồ thông minh, v.v.

- ❖ Nó có khả năng tốc độ lấy mẫu cao cùng với khả năng xuất dữ liệu nhanh.
- ❖ Tích hợp tính năng khử ánh sáng xung quanh.
- ❖ Để giao tiếp với vi điều khiển, cảm biến sử dụng chân I2C SCL và SDA.
- ❖ Một tính năng khác của cảm biến này là nó sử dụng bộ đệm FIFO (First-In-First-Out) 32 mẫu để lưu trữ dữ liệu so với MAX30100 chỉ có 16 bộ đệm FIFO mẫu. Nói cách khác, nó tiếp tục làm giảm mức tiêu thụ điện năng vì nó đã giữ tối đa ba mươi hai giá trị nhịp tim và SpO<sub>2</sub>.
- ❖ MAX30102 cũng có thể được sử dụng với các ngắt, có thể được bật cho một số nguồn như sẵn sàng cấp nguồn, sẵn sàng dữ liệu mới, khử ánh sáng xung quanh, sẵn sàng nhiệt độ và FIFO gần đầy. Với việc tạo ra ngắt, vi điều khiển có thể thực hiện các sự kiện khác không xảy ra trong quá trình thực thi tuần tự chương trình trong khi cảm biến liên tục lấy các mẫu dữ liệu mới.

Thông số kỹ thuật

- ❖ Cảm biến: MAX30102
- ❖ Thông số đo lường: Nhịp tim (BPM), Độ bão hòa oxy trong máu (SpO<sub>2</sub>)
- ❖ Điện áp hoạt động: 1,8V đến 3,3V DC
- ❖ Giao diện truyền thông: I2C
- ❖ Tiêu thụ dòng điện: Công suất thấp (thường là 0,8mA khi hoạt động)
- ❖ Nhiệt độ hoạt động: -40°C đến +85°C
- ❖ Kích thước: 1.9 cm x 1.4 cm x 0.3 cm

### 2.2.3. Cảm biến LIS2DH12TR

Cảm biến gia tốc MEMS LIS2DH12TR của STMicroelectronics là module 3 trục, đo gia tốc tĩnh và động với độ nhạy cao và tiêu thụ điện năng thấp, phù hợp cho các thiết bị đeo tay, điện thoại thông minh và ứng dụng IoT. Được đóng gói trong QFN-16 kích thước 3 × 3 mm, LIS2DH12TR trang bị vi điều khiển nội bộ để xử lý ngưỡng ngắt, FIFO và giao diện hướng đến ứng dụng MCU.

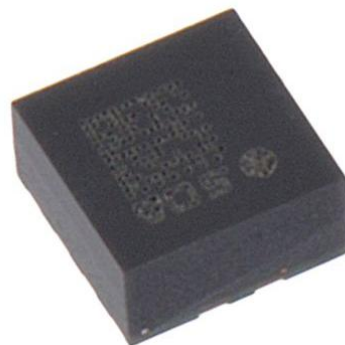
LIS2DH12TR cung cấp dải đo  $\pm 2/\pm 4/\pm 8/\pm 16$  g có thể cấu hình động, với độ phân giải 12-bit giúp phát hiện chuyển động tinh vi. Điện áp cung cấp từ 1,8 đến 3,6 V và độ tiêu thụ chỉ 2  $\mu$ A ở chế độ low-power, hỗ trợ đa cấp tốc độ

dữ liệu (ODR) từ 1 Hz đến 5 kHz. Tích hợp bộ lọc số và mạch FIFO 32 byte giảm tải cho MCU, cho phép ghi nhận 3 giá trị trục X, Y, Z liên tục và phát tín hiệu ngắt khi đầy FIFO hoặc khi vượt ngưỡng chuyển động đã định.

Module hỗ trợ giao tiếp I<sup>2</sup>C lên đến 400 kHz và SPI 4 dây max 10 MHz, với thanh ghi cấu hình linh hoạt để điều chỉnh full-scale, bandwidth, filter modes và threshold. Cảm biến còn cung cấp 2 chân ngắt (INT1, INT2) có thể lập trình cho các sự kiện như free-fall, tap/double-tap, activity/inactivity, zero motion, slope detection, thuận tiện trong phát hiện sự kiện thời gian thực. Ngoài ra, LIS2DH12TR tích hợp pinpoint self-test, cho phép đánh giá tình trạng cảm biến tại chỗ.

Thông số kỹ thuật:

- ❖ Điện áp hoạt động: 1.8–3.6 V
- ❖ Dải đo:  $\pm 2/\pm 4/\pm 8/\pm 16$  g
- ❖ Độ phân giải: 12-bit (LSB=1 mg ở  $\pm 2$  g)
- ❖ ODR: 1 Hz–5 kHz (tùy chế độ)
- ❖ Tiêu thụ: 2  $\mu$ A (low-power), 8  $\mu$ A–200  $\mu$ A (ODR phụ thuộc)
- ❖ FIFO: 32 byte, 3 trục 16-bit giá trị
- ❖ Ngắt: 2 chân (INT1, INT2) cho free-fall, tap, activity/inactivity, zero motion, slope detection
- ❖ Giao tiếp: I<sup>2</sup>C tốc độ 400 kHz, SPI 4 dây đến 10 MHz
- ❖ Nhiệt độ hoạt động:  $-40$  °C đến  $85$  °C
- ❖ Package: QFN-16,  $3 \times 3 \times 0.9$  mm, pad pitch 0.5 mm.

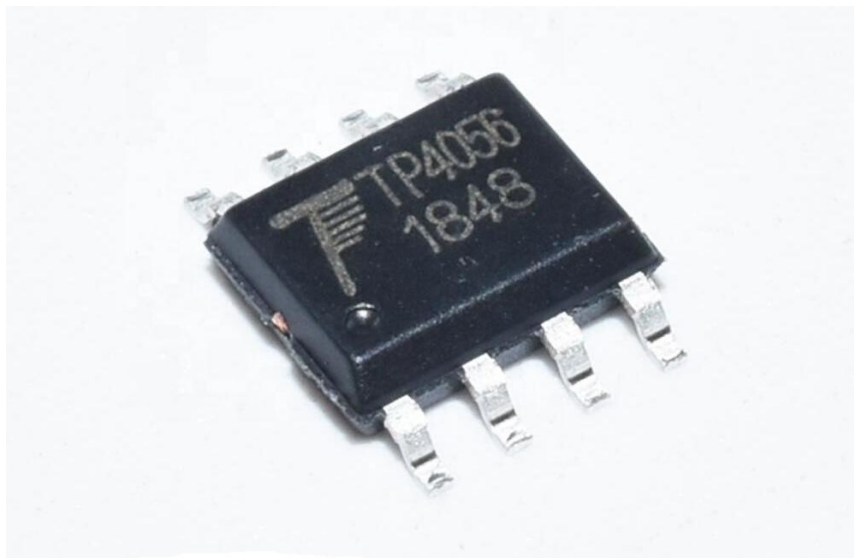


Hình 2.7. Cảm biến LIS2DH12TR



#### 2.2.4. IC sạc TP4056

TP4056 là một IC sạc tuyến tính (linear charger) chuyên dụng cho pin lithium-ion/lithium-polymer đơn cell, tích hợp đầy đủ mạch điều khiển sạc theo phương pháp Constant-Current/Constant-Voltage (CC/CV), bao gồm cả giai đoạn trickle charge cho pin đã xả sâu. IC này không yêu cầu MOSFET hay diode ngoài, cho phép thiết kế bo mạch đơn giản và chi phí thấp. TP4056 còn tích hợp chức năng bảo vệ quá dòng, bảo vệ quá áp, giám sát nhiệt độ qua chân TEMP, tự động tái sạc khi điện áp pin xuống dưới mức định sẵn và kết thúc sạc ở dòng C/10 để ngăn ngừa tình trạng overcharge.



Hình 2.8. TP4056 – IC Sạc Li-Ion Linear 1 A (SOP-8)

Thông số kỹ thuật:

- ❖ Điện áp đầu vào (IN): 4,0 – 8,0 V (thích hợp nguồn USB 5 V)
- ❖ Điện áp sạc cố định (BAT): 4,200 V  $\pm$  1,5 %
- ❖ Dòng sạc tối đa: Lập trình qua R\_PROG, tối đa 1 000 mA ( $R\_PROG = 1,2 \text{ k}\Omega \leftrightarrow 1 \text{ A}$ )
- ❖ Chế độ trickle-charge: Tự động sạc nhỏ ( $\sim 140 \text{ mA}$ ) khi BAT < 2,9 V
- ❖ Kết thúc sạc (termination): Khi  $I\_CHG < C/10$  ( $\approx 100 \text{ mA}$ ), dòng rò khi kết thúc  $\leq 55 \mu\text{A}$
- ❖ Soft-start nội: Giảm dòng đỉnh ban đầu, bảo vệ nguồn
- ❖ Chức năng bảo vệ: Bảo vệ quá dòng, quá áp; giám sát nhiệt độ qua chân TEMP

- ❖ Chỉ báo trạng thái: Hai chân STAT1/STAT2 điều khiển LED báo SẠC / HOÀN THÀNH
- ❖ Dòng tắt (shutdown):  $\leq 150 \mu\text{A}$
- ❖ Nhiệt độ hoạt động:  $-40^\circ\text{C}$  đến  $+85^\circ\text{C}$
- ❖ Đóng gói: SOP-8, hỗ trợ tản nhiệt qua pad GND lớn

#### 2.2.5. IC LDO LP2985-33DBVR

LP2985-33DBVR của Texas Instruments là bộ điều áp tuyến tính (LDO) cố định đầu ra 3,3 V, tối ưu cho cấp nguồn vi điều khiển và các mạch số tiêu thụ dòng không lớn ( $\leq 150 \text{ mA}$ ). IC này nổi bật với độ ồn rất thấp, PSRR cao, và khả năng chịu đựng dải điện áp đầu vào rộng (2,5–16 V), giúp bảo vệ mạch khỏi nhiễu nền do các nguồn chuyển mạch trước đó. LP2985-33DBVR cũng tích hợp chân Enable để dễ dàng điều khiển tắt/mở đầu ra, cùng các tính năng bảo vệ quá dòng và quá nhiệt đảm bảo an toàn cho ứng dụng.



Hình 2.9. LP2985-33DBVR – IC LDO 3.3 V 150 mA

#### Thông số kỹ thuật

- ❖ Điện áp đầu vào (VIN): 2,5 – 16 V
- ❖ Điện áp đầu ra cố định (VOUT):  $3,300 \text{ V} \pm 1 \%$  (A-grade)
- ❖ Dòng tải tối đa: 150 mA
- ❖ Dòng tiêu thụ không tải (IQ):  $\sim 95 \mu\text{A}$
- ❖ Dòng rò ở chế độ shutdown:  $\leq 0,01 \mu\text{A}$
- ❖ Ổn áp soft-start: Hạn chế dòng inrush khi khởi động
- ❖ Bảo vệ: Giới hạn dòng quá tải và ngắt khi quá nhiệt
- ❖ PSRR:  $> 70 \text{ dB}$  @ 1 kHz;  $> 40 \text{ dB}$  @ 1 MHz
- ❖ Độ ồn đầu ra:  $\sim 30 \mu\text{V RMS}$  (với tụ bypass 10 nF)

- ❖ Chân Enable (EN): Điều khiển bật/tắt LDO
- ❖ Nhiệt độ hoạt động:  $-40\text{ }^{\circ}\text{C}$  đến  $+125\text{ }^{\circ}\text{C}$
- ❖ Đóng gói: SOT-23-5 ( $2,9 \times 1,6\text{ mm}$ )

#### 2.2.6. Màn hình OLED 0.96 Inch

Module màn hình OLED 0.96 Inch 128x64 sử dụng giao tiếp I2C, tích hợp IC điều khiển SSD1306 của hãng Solomon Systech, được thiết kế đặc biệt cho các ứng dụng nhúng và thiết bị đeo thông minh nhờ kích thước nhỏ gọn, hiệu quả hiển thị cao và tiêu thụ điện năng thấp. Với độ phân giải 128x64 pixel, màn hình cung cấp khả năng hiển thị thông tin sắc nét và rõ ràng, đặc biệt phù hợp cho việc hiển thị các dữ liệu sinh học, trạng thái hệ thống và thông tin cảnh báo trên các thiết bị theo dõi sức khỏe cá nhân. Công nghệ OLED không cần đèn nền, giúp tăng cường độ tương phản và giảm điện năng tiêu thụ đáng kể, đặc biệt hiệu quả trong các ứng dụng chạy pin dài ngày.



Hình 2.10. Màn hình OLED 0.96 Inch

Màn hình OLED 128x64 SSD1306 được hỗ trợ rộng rãi bởi các thư viện phổ biến như Adafruit\_SSD1306 và u8g2, giúp việc lập trình trở nên dễ dàng và nhanh chóng, tương thích với nhiều nền tảng vi điều khiển như Arduino, ESP32, STM32 và Raspberry Pi. Module chỉ yêu cầu 2 chân dữ liệu (SDA, SCL) và nguồn cấp (VCC, GND), giúp giảm đáng kể độ phức tạp khi đấu nối, tiết kiệm tài nguyên GPIO và tối ưu hóa không gian thiết kế PCB. Ngoài ra, do không sử dụng đèn nền, màn hình có khả năng hiển thị thông tin rõ nét ngay cả dưới ánh sáng mạnh hay điều kiện ánh sáng yếu, phù hợp cho các thiết bị đeo sử dụng ngoài trời hoặc trong nhà.

Việc sử dụng công nghệ OLED giúp màn hình có góc nhìn rộng, đảm bảo thông tin hiển thị luôn dễ dàng quan sát từ nhiều hướng khác nhau. Màn hình OLED này cũng có tốc độ làm mới nhanh, không bị hiện tượng bóng mờ, rất phù hợp để hiển thị thông tin cập nhật liên tục như nhịp tim, oxy trong máu và các chỉ số sức khỏe khác. Thiết kế nhỏ gọn cùng khả năng tiêu thụ điện năng cực thấp giúp kéo dài thời gian sử dụng của các thiết bị chạy bằng pin, nâng cao trải nghiệm người dùng một cách đáng kể.

Kích thước nhỏ gọn 27 x 28 mm giúp module dễ dàng tích hợp vào các thiết kế mạch in nhỏ gọn mà không chiếm nhiều diện tích, rất phù hợp cho các ứng dụng đeo tay và thiết bị cá nhân. Việc tối ưu hóa điện năng tiêu thụ và khả năng hiển thị ổn định giúp màn hình OLED SSD1306 có thể hoạt động liên tục trong thời gian dài mà không cần thay thế hoặc sạc pin thường xuyên, giảm thiểu sự gián đoạn trong quá trình sử dụng.

Bảng 2.1. Bảng thông số kỹ thuật chính.

Thông số kỹ thuật	Giá trị
Loại màn hình	OLED
Kích thước màn hình	0.96 inch
Độ phân giải	128 x 64 pixel
IC điều khiển	SSD1306
Giao tiếp	I2C
Điện áp hoạt động	3.3V - 5V
Công suất tiêu thụ	khoảng 0.04 W
Góc nhìn	>160 độ
Màu hiển thị	Trắng
Địa chỉ mặc định I2C	0x3C hoặc 0x3D (tùy phiên bản)
Kích thước tổng thể module	27 x 28 mm
Kích thước hiển thị vùng ảnh	21.74 x 10.86 mm

Bảng 2.2. Bảng chân tín hiệu.

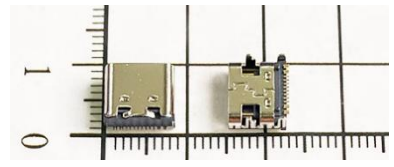
Chân tín hiệu	Chức năng mô tả	Chân tín hiệu
VCC	Chân cấp nguồn điện (3.3V - 5V)	VCC
GND	Chân nối đất	GND
SDA	Chân dữ liệu I2C	SDA
SCL	Chân xung nhịp I2C	SCL

### 2.2.6. Các linh kiện khác

Dưới đây là bảng liệt kê các linh kiện khác:

Bảng 2.3. Bảng các linh kiện khác được sử dụng trong dự án.

STT	Linh kiện	Mô tả	Hình ảnh
1	R	Điện trở kích chân dán	
2	C	Tụ điện chân dán	
3	BC817	Transistor NPN BC817	
4	BSS138	MOSFET N-kênh BSS138 cho chuyển mạch logic	
5	CH340C	IC chuyển USB ↔ UART, chip CH340C	
6	SW_Push	Nút nhấn (push-button)	

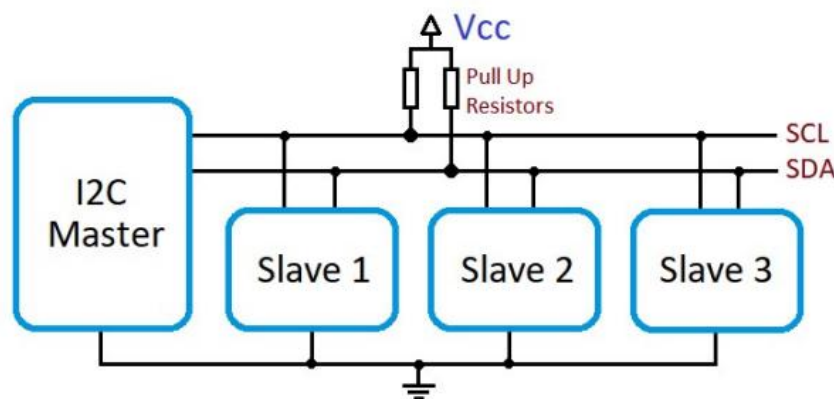
7	USB C Plug USB2.0	Cổng USB-C 2.0	
---	-------------------------	----------------	---

## 2.3. Các chuẩn giao tiếp

### 2.3.1. Giao tiếp I2C

Đầu năm 1980 Phillips đã phát triển một chuẩn giao tiếp nối tiếp 2 dây được gọi là I2C. I2C là tên viết tắt của cụm từ Inter-Integrated Circuit. Đây là đường Bus giao tiếp giữa các IC với nhau. I2C mặc dù được phát triển bởi Phillips, nhưng nó đã được rất nhiều nhà sản xuất IC trên thế giới sử dụng. I2C trở thành một chuẩn công nghiệp cho các giao tiếp điều khiển, có thể kể ra đây một vài tên tuổi ngoài Phillips như: Texas Instrument(TI), MaximDallas, analog Device, National Semiconductor, vv. Bus I2C được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM, vv chip nhớ như: RAM tĩnh (Static RAM), EEPROM, bộ chuyển đổi tương tự số (ADC), số tương tự (DAC), IC điều khiển LCD, LED, vv. I2C sử dụng hai đường truyền tín hiệu:

- Một đường xung clock (SCL) chỉ do Master phát đi (thông thường ở 100kHz và 400kHz. Mức cao nhất là 1MHz và 3.4MHz).
- Một đường dữ liệu (SDA) theo 2 hướng.



Hình 2.11. Chuẩn giao tiếp I2C

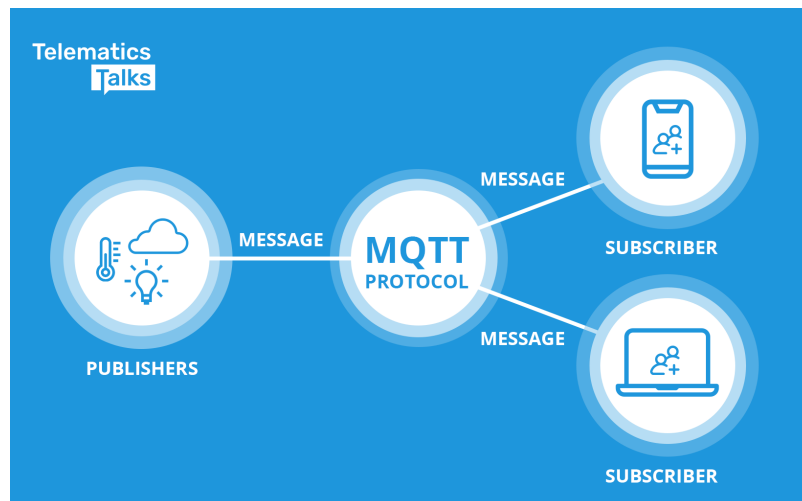
Có rất nhiều thiết bị có thể cũng được kết nối vào một bus I2C, tuy nhiên sẽ không xảy ra chuyện nhầm lẫn giữa các thiết bị, bởi mỗi thiết bị sẽ được nhận ra bởi một địa chỉ duy nhất với một quan hệ master/slave tồn tại trong suốt thời gian kết nối. Mỗi thiết bị có thể hoạt động như là thiết bị nhận hoặc truyền dữ liệu hay

có thể vừa truyền vừa nhận. Hoạt động truyền hay nhận còn tùy thuộc vào việc thiết bị đó là master hay slave. Một thiết bị hay một IC khi kết nối với bus I2C, ngoài một địa chỉ (duy nhất) để phân biệt, nó còn được cấu hình là thiết bị master hay slave. Đó là vì trên một bus I2C thì quyền điều khiển thuộc về thiết bị master hay slave. Thiết bị master nắm vai trò tạo xung clock đồng bộ cho toàn hệ thống, khi giữa hai thiết bị master – slave giao tiếp thì thiết bị master có nhiệm vụ tạo xung clock đồng bộ và quản lý địa chỉ của thiết bị slave trong suốt quá trình giao tiếp. Thiết bị master giữ vai trò chủ động, còn thiết bị slave giữ vai trò bị động trong việc giao tiếp. Về lý thuyết lẫn thực tế I2C sử dụng 7 bit để định địa chỉ, do đó trên một bus có thể có tới 27 địa chỉ tương ứng với 128 thiết bị có thể kết nối, nhưng chỉ có 112, 16 địa chỉ còn lại được sử dụng vào mục đích riêng. Bit còn lại quy định việc đọc hay ghi dữ liệu (1 là write, 0 là read). Có một lưu ý nhỏ về xung clock. Bản chất của I2C là dữ liệu trên đường SDA chỉ được ghi nhận ở sườn lên của chân CLK. Do vậy xung clock có thể không cần chính xác tốc độ là 1MHz hay 3.4MHz. Lợi dụng điểm này có thể sử dụng 2 chân GPIO để làm chân giao tiếp I2C mềm mà không nhất thiết cần một chân CLK tạo xung với tốc độ chính xác. Quá trình truyền dữ liệu giữa 2 thiết bị chế độ Master – Slave dùng bus I2C diễn ra như sau: Thiết bị master tạo xung START (tức là đường SDA thay đổi từ mức cao xuống mức thấp và đường SCL đang ở mức cao) để bắt đầu quá trình giao tiếp. Thiết bị master gửi địa chỉ của thiết bị slave muốn giao tiếp cùng với bit R/W và đợi xung ACK phản hồi. Địa chỉ của thiết bị slave được định nghĩa bởi 7 bit, ngoài ra có thể đánh địa chỉ dưới dạng 10 bit nếu sau lệnh START gửi chuỗi 11110 ra đường SDA. Bit R/W dùng để điều khiển hướng truyền- bit “0” là truyền từ master tới slave, ngược lại bit “1” là truyền từ slave tới master. Việc thiết lập bit R/W do thiết bị master quy định. Khi nhận được xung báo ACK từ thiết bị slave xác nhận đúng địa chỉ thì bắt đầu thực hiện truyền dữ liệu. Dữ liệu được gửi theo từng byte. Mỗi byte gồm 8 bit và sau mỗi byte đều bắt buộc có một xung ACK để đảm bảo quá trình truyền nhận diễn ra chính xác. Số lượng byte là không giới hạn. Xung ACK được định nghĩa là SDA kéo xuống mức thấp. Ngoài ra, khi không nhận được đúng địa chỉ hay khi muốn kết thúc quá trình giao tiếp thiết bị

nhận sẽ gửi một xung NACK tức là SDA ở mức cao. Kết thúc quá trình truyền, thiết bị master sẽ gửi xung STOP hoặc thiết bị slave gửi xung NACK để báo hiệu kết thúc. Xung STOP tương tự như xung START nhưng trạng thái của đường SDA thay đổi từ mức thấp lên mức cao. Ngoài ra, trong một quá trình giao tiếp có thể lặp lại xung START kèm theo một địa chỉ mới để bắt đầu một giao tiếp khác. Điểm mạnh của I2C chính là hiệu suất và sự đơn giản của nó: một khối điều khiển trung tâm có thể điều khiển cả một mạng thiết bị mà chỉ cần hai lối ra điều khiển.

### 2.3.2. Giao thức không dây MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông điệp dạng publish/subscribe nhẹ, tiết kiệm băng thông, được thiết kế dành riêng cho các ứng dụng IoT (Internet of Things). Giao thức này ra đời vào năm 1999, được phát triển bởi IBM với mục tiêu ban đầu là truyền tải dữ liệu cảm biến từ xa trong môi trường có băng thông hạn chế, độ trễ cao và không ổn định, điển hình là các ứng dụng giám sát từ xa thông qua vệ tinh. Với đặc điểm gọn nhẹ, đơn giản và hiệu quả cao, MQTT nhanh chóng được ứng dụng rộng rãi vào các hệ thống cảm biến không dây, các hệ thống giám sát sức khỏe, giám sát môi trường, hệ thống điều khiển tự động và đặc biệt là các thiết bị IoT hiện đại.



Hình 2.12. Mô hình MQTT

Giao thức MQTT hoạt động dựa trên mô hình truyền nhận thông điệp bất đồng bộ theo cấu trúc client-server. Trong mô hình này, một thành phần trung tâm được gọi là Broker (máy chủ môi giới) sẽ giữ vai trò quản lý luồng thông điệp



giữa các client, đảm bảo việc truyền tải thông tin diễn ra an toàn, nhanh chóng và tin cậy. Các client trong hệ thống MQTT có thể hoạt động ở hai vai trò chính: Publisher (bên gửi thông điệp) và Subscriber (bên nhận thông điệp). Các publisher sẽ gửi thông điệp lên broker thông qua các chủ đề cụ thể (topics), broker tiếp nhận và chuyển tiếp các thông điệp này đến các subscriber đã đăng ký nhận thông tin từ các topic tương ứng. Cách thức này giúp tách biệt hoàn toàn giữa người gửi và người nhận, giảm sự phụ thuộc trực tiếp giữa các thiết bị, từ đó đơn giản hóa việc phát triển và mở rộng hệ thống.

MQTT được thiết kế đặc biệt để tối ưu hóa băng thông truyền dẫn và giảm thiểu điện năng tiêu thụ cho các thiết bị sử dụng pin hoặc hoạt động trong môi trường tài nguyên hạn chế. Thông điệp MQTT có cấu trúc đơn giản, gọn nhẹ, bao gồm các trường cơ bản như topic, payload (dữ liệu), QoS (Quality of Service – mức độ đảm bảo chất lượng truyền tải) và Retain (lưu giữ trạng thái cuối cùng). MQTT cung cấp ba mức QoS cơ bản nhằm cân bằng giữa hiệu suất truyền tải và độ tin cậy của thông điệp:

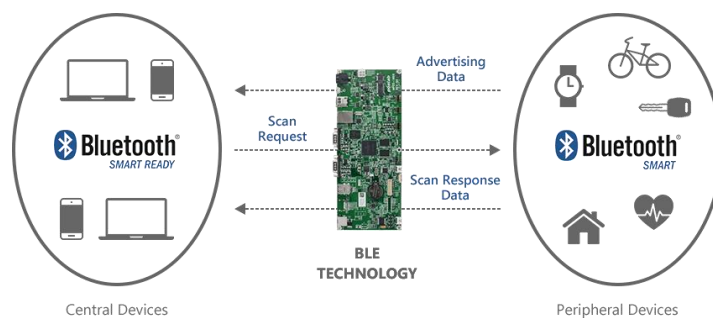
- ❖ QoS 0 (At most once): gửi thông điệp tối đa một lần, không đảm bảo chắc chắn thông điệp tới được nơi nhận, nhưng đạt tốc độ cao nhất.
- ❖ QoS 1 (At least once): đảm bảo thông điệp chắc chắn được gửi đến bên nhận ít nhất một lần, đôi khi có thể gây trùng lặp.
- ❖ QoS 2 (Exactly once): đảm bảo chắc chắn thông điệp chỉ được gửi đi chính xác một lần duy nhất, đảm bảo độ tin cậy tuyệt đối nhưng đòi hỏi nhiều tài nguyên và thời gian xử lý hơn.

Giao thức MQTT hoạt động tốt trên nền TCP/IP, hỗ trợ bảo mật qua việc sử dụng các chuẩn SSL/TLS, xác thực người dùng bằng username/password và có thể mã hóa dữ liệu truyền tải nhằm đảm bảo an toàn thông tin. Nhờ cấu trúc nhẹ, việc triển khai giao thức này trên các hệ thống nhúng với vi điều khiển công suất thấp như ESP32, ESP8266, STM32 và Arduino trở nên rất thuận tiện và hiệu quả. Bên cạnh đó, việc MQTT hỗ trợ cơ chế kết nối liên tục hoặc kết nối ngắt quãng định kỳ (keep-alive), giúp tối ưu đáng kể thời lượng pin và tài nguyên mạng trong các ứng dụng IoT di động hoặc từ xa.

Với các đặc điểm vượt trội kể trên, MQTT đã trở thành lựa chọn phổ biến cho các ứng dụng IoT trong nhiều lĩnh vực, từ nông nghiệp thông minh, tự động hóa công nghiệp, quản lý và giám sát năng lượng, cho tới các ứng dụng thiết bị đeo tay thông minh và y tế cá nhân hiện đại. Giao thức này giúp giảm thiểu phức tạp trong việc truyền tải dữ liệu cảm biến, đồng thời tối ưu hóa độ tin cậy, bảo mật và khả năng mở rộng, đáp ứng tốt các yêu cầu khắt khe của hệ thống IoT trong thực tế.

### 2.3.3. Giao thức không dây BLE

Bluetooth Low Energy (BLE), còn được gọi là Bluetooth Smart, là một công nghệ giao tiếp không dây tiết kiệm năng lượng, được phát triển nhằm phục vụ cho các ứng dụng đòi hỏi thời lượng pin dài và tốc độ truyền dữ liệu thấp hoặc trung bình. Chuẩn giao tiếp BLE được giới thiệu lần đầu tiên bởi tổ chức Bluetooth SIG (Bluetooth Special Interest Group) vào năm 2010 như một phần của chuẩn Bluetooth 4.0. Với đặc tính nổi bật là khả năng tiết kiệm năng lượng vượt trội và tương thích rộng rãi với hầu hết các thiết bị di động hiện đại như smartphone, máy tính bảng và máy tính cá nhân, BLE đã nhanh chóng trở thành một trong những chuẩn giao tiếp phổ biến nhất trong các ứng dụng Internet of Things (IoT), thiết bị đeo tay thông minh, các hệ thống theo dõi sức khỏe cá nhân và thiết bị nhà thông minh.



Hình 2.13. Mô hình BLE

Cấu trúc hoạt động của BLE dựa trên mô hình giao tiếp không đối xứng giữa hai loại thiết bị chính: Central và Peripheral. Trong đó, thiết bị Central thường là các thiết bị chủ động như điện thoại thông minh hoặc máy tính, chịu trách nhiệm quét và thiết lập kết nối tới các thiết bị Peripheral. Ngược lại, thiết bị Peripheral là các cảm biến, thiết bị đeo tay hay các thiết bị đầu cuối, thường hoạt động ở chế

độ tiêu thụ năng lượng thấp, phát các gói tin quảng bá (advertising packets) để các thiết bị Central có thể phát hiện và kết nối. Khi kết nối được thiết lập, BLE cho phép hai thiết bị trao đổi dữ liệu thông qua một cấu trúc dữ liệu đặc biệt gồm các dịch vụ (Services) và các đặc tính (Characteristics), mỗi dịch vụ và đặc tính được nhận dạng thông qua các định danh UUID (Universally Unique Identifier).

BLE có một cơ chế đặc biệt giúp tiết kiệm điện năng đáng kể so với các chuẩn không dây khác. Điều này được thực hiện thông qua việc quản lý chặt chẽ chu kỳ kết nối, giảm thời gian phát sóng radio và tối ưu các gói dữ liệu truyền tải có kích thước nhỏ gọn (thường không vượt quá vài chục byte mỗi lần). Chính vì vậy, BLE có thể hoạt động hiệu quả ngay cả khi sử dụng các nguồn năng lượng rất nhỏ như pin đồng xu (coin-cell battery). Một ưu điểm nổi bật nữa là khả năng hỗ trợ nhiều chế độ phát quảng bá như Advertising mode, Scanning mode, và Connected mode. Chế độ quảng bá cho phép các thiết bị liên tục phát tín hiệu để dễ dàng được nhận biết, trong khi Connected mode cho phép giao tiếp hai chiều ổn định, tin cậy và bảo mật giữa thiết bị Peripheral và thiết bị Central.

BLE cung cấp ba mức bảo mật khác nhau dựa trên việc sử dụng mã hóa AES-CCM 128-bit, xác thực, ghép đôi (pairing), và bonding nhằm đảm bảo an toàn dữ liệu truyền tải giữa các thiết bị. Các thiết bị BLE có thể hoạt động trên băng tần 2,4 GHz ISM (Industrial, Scientific, Medical) với tổng cộng 40 kênh, trong đó 3 kênh dành riêng cho quảng bá và 37 kênh còn lại sử dụng cho các phiên kết nối thực sự. Việc sử dụng thuật toán nhảy tần thích nghi (Adaptive Frequency Hopping - AFH) giúp giảm thiểu tối đa các vấn đề nhiễu sóng và xung đột với các thiết bị khác trên cùng băng tần, tăng cường sự ổn định và tin cậy của kết nối.

Với cấu trúc nhẹ và khả năng tiêu thụ năng lượng thấp vượt trội, giao thức BLE rất phù hợp cho các hệ thống cảm biến thông minh, thiết bị y tế cá nhân như đo nhịp tim, SpO<sub>2</sub>, theo dõi chuyển động và giấc ngủ; ngoài ra còn ứng dụng rộng rãi trong các thiết bị nhà thông minh, beacon định vị trong nhà, và nhiều hệ thống IoT công nghiệp khác. Hiện nay, BLE đã được tích hợp rộng rãi vào các nền tảng vi điều khiển phổ biến như ESP32, STM32, Nordic nRF, cũng như có sẵn các thư viện hỗ trợ cho các nền tảng phần mềm như Android, iOS, và Windows, giúp cho

việc triển khai và phát triển ứng dụng BLE trở nên dễ dàng và thuận tiện hơn bao giờ hết.

## 2.4. Công nghệ phần mềm

### 2.4.1. Ngôn ngữ lập trình C



Hình 2.14. Ngôn ngữ lập trình C.

Ngôn ngữ lập trình C là một ngôn ngữ lập trình cấp trung, được phát triển vào năm 1972 bởi Dennis Ritchie tại Bell Labs với mục đích ban đầu là viết hệ điều hành UNIX. Ngôn ngữ C kết hợp những đặc điểm của cả ngôn ngữ cấp thấp và cấp cao, giúp lập trình viên vừa có thể kiểm soát phần cứng hiệu quả, vừa dễ dàng phát triển các ứng dụng phức tạp.

C là một ngôn ngữ có hiệu suất cao nhờ khả năng tối ưu hóa tài nguyên hệ thống và quản lý bộ nhớ trực tiếp. Điều này khiến nó trở nên lý tưởng cho các ứng dụng yêu cầu tốc độ và hiệu quả, chẳng hạn như hệ điều hành, trình điều khiển thiết bị, phần mềm nhúng và các ứng dụng thời gian thực. Khả năng điều khiển bộ nhớ bằng con trỏ và cấp phát động trong C cho phép lập trình viên tự quản lý việc sử dụng bộ nhớ, điều mà không nhiều ngôn ngữ lập trình khác cung cấp.

Một trong những điểm mạnh của C là tính di động cao, nghĩa là mã nguồn viết bằng C có thể biên dịch và chạy trên nhiều nền tảng khác nhau mà không cần chỉnh sửa nhiều. Điều này làm cho C trở thành ngôn ngữ phổ biến trong phát triển phần mềm hệ thống, các ứng dụng nhúng, và thậm chí là phát triển các ngôn ngữ lập trình khác.

Ngôn ngữ C có ảnh hưởng lớn đến nhiều ngôn ngữ lập trình hiện đại như C++, Java, C#, và Python. Các hệ điều hành lớn như UNIX, Linux, và Windows

đều được phát triển phần lớn bằng C, và nó vẫn là lựa chọn hàng đầu trong lập trình hệ thống nhưng cũng như các ứng dụng yêu cầu tốc độ và khả năng tối ưu hóa cao.

Với cú pháp rõ ràng, mạnh mẽ và khả năng kiểm soát chi tiết hệ thống, ngôn ngữ lập trình C đã trở thành nền tảng của nhiều hệ thống công nghệ quan trọng và vẫn được sử dụng rộng rãi cho đến ngày nay.

#### 2.4.2. ESP-IDF Framework



Hình 2.15. Logo ESP-IDF Framework

ESP-IDF (Espressif IoT Development Framework) là một bộ khung phát triển phần mềm chính thức được phát triển bởi Espressif Systems dành riêng cho dòng chip ESP32 và ESP32-S series, hỗ trợ đầy đủ các chức năng cần thiết cho các ứng dụng IoT hiện đại. ESP-IDF cung cấp một môi trường lập trình đầy đủ, tích hợp sẵn hệ điều hành thời gian thực FreeRTOS, thư viện giao tiếp ngoại vi, quản lý bộ nhớ linh hoạt, hỗ trợ mạnh mẽ các kết nối không dây như Wi-Fi, Bluetooth Classic và Bluetooth Low Energy (BLE), cùng khả năng bảo mật nâng cao. Tất cả các tính năng này giúp nhà phát triển có thể nhanh chóng xây dựng, thử nghiệm và triển khai các ứng dụng nhưng một cách chuyên nghiệp và đáng tin cậy.

ESP-IDF được xây dựng dựa trên ngôn ngữ lập trình C và C++, đảm bảo hiệu suất hoạt động cao, thời gian đáp ứng nhanh và khả năng tùy biến mạnh mẽ. Framework này cung cấp một kiến trúc module rất rõ ràng và trực quan, giúp người dùng dễ dàng quản lý các chức năng riêng biệt như giao tiếp ngoại vi

UART, I<sup>2</sup>C, SPI; giao thức mạng TCP/IP, MQTT, HTTP; hệ thống tập tin ảo SPIFFS, FATFS và đặc biệt là khả năng cập nhật phần mềm OTA (Over-The-Air) một cách dễ dàng. Bên cạnh đó, ESP-IDF còn tích hợp các công cụ mạnh mẽ hỗ trợ debug và profiling hiệu suất như ESP32 OpenOCD và ESP SystemView, giúp quá trình phát triển và kiểm thử ứng dụng trở nên thuận lợi và chính xác hơn.

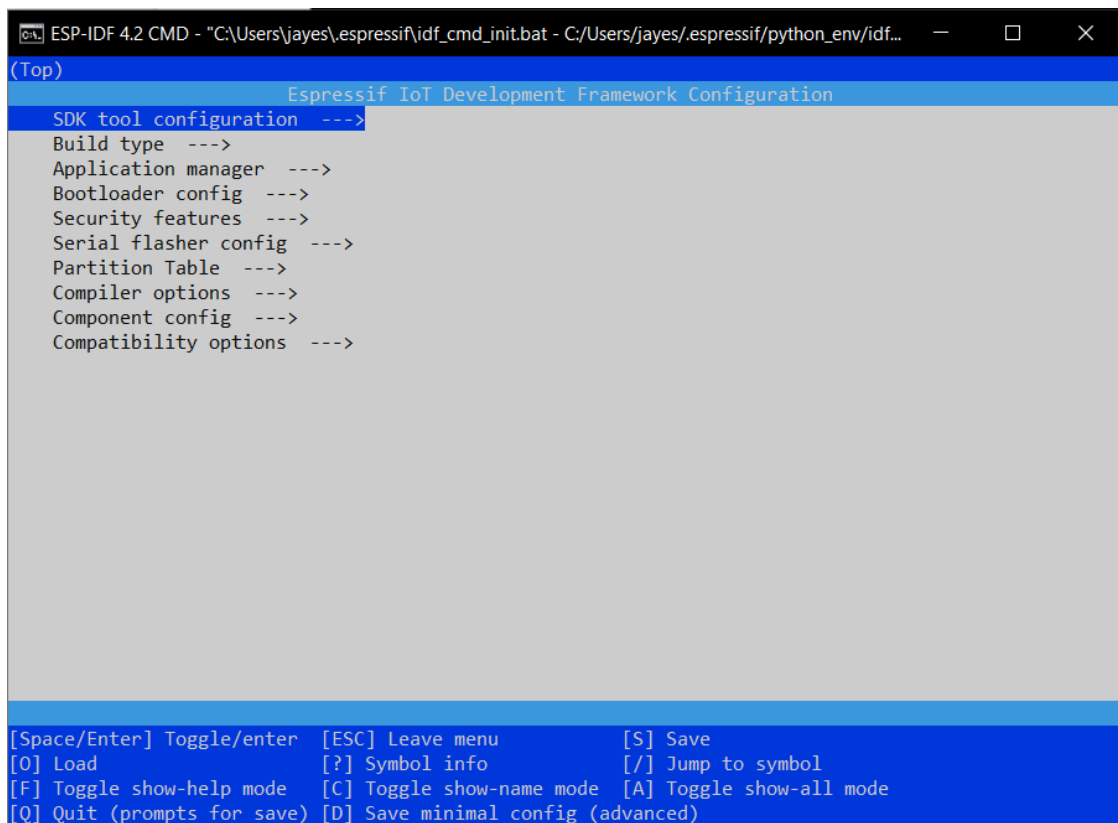
Một trong những ưu điểm nổi bật nhất của ESP-IDF là khả năng quản lý và phân chia tài nguyên hệ thống một cách linh hoạt và hiệu quả nhờ vào FreeRTOS được tích hợp sẵn. FreeRTOS cho phép tạo nhiều tác vụ song song (multi-tasking), đồng bộ hóa các tác vụ bằng semaphore, mutex và event groups, quản lý bộ nhớ heap một cách linh hoạt, giúp tối ưu hóa hiệu năng và độ ổn định trong các ứng dụng đòi hỏi tính thời gian thực nghiêm ngặt. Bên cạnh đó, ESP-IDF hỗ trợ khả năng quản lý năng lượng thông minh, cho phép chip ESP32 chuyển đổi linh hoạt giữa nhiều chế độ vận hành như active mode, modem-sleep mode, light-sleep mode và deep-sleep mode, giúp tối ưu đáng kể điện năng tiêu thụ trong các ứng dụng chạy pin dài ngày.

Khả năng bảo mật cũng là một ưu điểm quan trọng của ESP-IDF. Framework này tích hợp sẵn các công nghệ bảo mật hàng đầu như mã hóa TLS/SSL để bảo vệ dữ liệu khi truyền tải qua mạng, mã hóa flash AES-256 để bảo vệ firmware và dữ liệu lưu trữ cục bộ, hỗ trợ các cơ chế xác thực dựa trên chứng chỉ số X.509, JWT (JSON Web Token), và Secure Boot để đảm bảo tính toàn vẹn và xác thực của firmware trong quá trình khởi động. Những tính năng bảo mật nâng cao này giúp ESP-IDF trở thành lựa chọn tối ưu cho các ứng dụng IoT yêu cầu bảo mật dữ liệu cao như hệ thống giám sát sức khỏe, nhà thông minh, hay các ứng dụng công nghiệp nhạy cảm.

Ngoài ra, Espressif Systems còn cung cấp một cộng đồng phát triển đông đảo và hỗ trợ liên tục cập nhật, sửa lỗi và bổ sung các tính năng mới cho ESP-IDF, giúp người dùng luôn được tiếp cận những công nghệ mới nhất và có sự hỗ trợ kịp thời từ nhà sản xuất. ESP-IDF cũng tích hợp sâu với các công cụ phổ biến như Eclipse, VS Code, PlatformIO và Espressif IDE, tạo điều kiện thuận lợi để lập

trình viên dễ dàng phát triển, quản lý và triển khai dự án trong nhiều môi trường khác nhau.

Nhìn chung, với những ưu điểm nổi bật như cấu trúc module rõ ràng, hỗ trợ FreeRTOS mạnh mẽ, khả năng bảo mật cao, quản lý năng lượng hiệu quả và cộng đồng hỗ trợ đông đảo, ESP-IDF thực sự là lựa chọn lý tưởng và chuyên nghiệp cho các nhà phát triển muốn xây dựng các hệ thống IoT hiện đại, ổn định và bảo mật dựa trên nền tảng vi điều khiển ESP32.

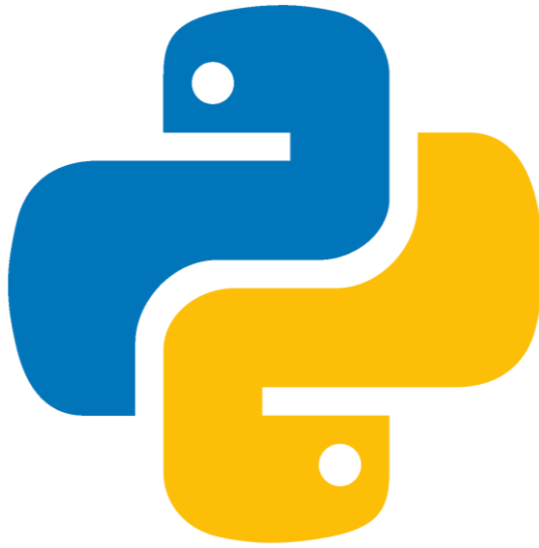


Hình 2.16. Giao diện cấu hình của ESP-IDF.

### 2.4.3. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao, đa mục đích, được Guido van Rossum giới thiệu lần đầu vào năm 1991 với tiêu chí cốt lõi là đơn giản hóa cú pháp và tăng cường tính dễ đọc cho mã nguồn. Với đặc trưng là ngôn ngữ lập trình thông dịch, Python giúp người lập trình nhanh chóng kiểm thử các ý tưởng và phát triển các ứng dụng phần mềm một cách hiệu quả và trực quan. Hiện nay, Python trở thành một trong những ngôn ngữ phổ biến nhất trên thế giới, được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ các hệ thống nhúng IoT, trí tuệ

nhân tạo, phân tích dữ liệu, tự động hóa quy trình đến các ứng dụng web và desktop.



Hình 2.17. Logo ngôn ngữ lập trình Python

Một trong những ưu điểm lớn nhất của Python là cú pháp ngắn gọn, dễ hiểu, gần với ngôn ngữ tự nhiên, nhờ vậy Python giảm thiểu đáng kể thời gian cần thiết để học tập và làm quen, ngay cả đối với những lập trình viên mới bắt đầu. Việc quản lý bộ nhớ tự động cùng khả năng xử lý linh hoạt các kiểu dữ liệu động cho phép người lập trình giảm đáng kể thời gian viết mã, hạn chế lỗi xảy ra trong quá trình phát triển. Bên cạnh đó, Python sở hữu một hệ sinh thái thư viện cực kỳ đa dạng và mạnh mẽ, như NumPy, Pandas, Matplotlib trong xử lý và phân tích dữ liệu, TensorFlow, PyTorch trong lĩnh vực trí tuệ nhân tạo, Django và Flask trong phát triển web, giúp người dùng dễ dàng mở rộng và áp dụng vào đa dạng các dự án và lĩnh vực chuyên biệt. Với dữ kiện trên, Python là lựa chọn lý tưởng cho mọi dự án từ cơ bản đến chuyên sâu.

Ngoài ra, Python đặc biệt phù hợp với các hệ thống IoT và ứng dụng nhúng nhờ khả năng dễ dàng kết nối và giao tiếp với phần cứng thông qua các thư viện có sẵn. Ví dụ như thư viện PySerial cho phép Python tương tác với các thiết bị ngoại vi thông qua giao tiếp UART, hoặc thư viện Paho MQTT cung cấp khả năng tích hợp nhanh chóng với các hệ thống sử dụng giao thức MQTT. Với các bo mạch nhúng thông dụng như Raspberry Pi hay ESP32, Python giúp đơn giản hóa



việc lập trình, giảm thời gian phát triển, đồng thời tối ưu hóa hiệu quả sử dụng tài nguyên của hệ thống nhúng.

Một ưu điểm quan trọng nữa của Python chính là khả năng tương thích đa nền tảng. Các ứng dụng viết bằng Python có thể hoạt động một cách ổn định và nhất quán trên nhiều nền tảng hệ điều hành khác nhau như Windows, Linux, macOS và thậm chí trên các nền tảng nhúng. Python còn được hỗ trợ mạnh mẽ bởi cộng đồng phát triển vô cùng đông đảo và năng động trên toàn thế giới, điều này giúp các nhà phát triển nhanh chóng tiếp cận được những giải pháp mới nhất, nhận hỗ trợ nhanh chóng khi gặp vấn đề và cập nhật liên tục các phiên bản nâng cấp cũng như bản vá bảo mật.

Như vậy, với cú pháp đơn giản, dễ hiểu, hệ thống thư viện đồ sộ, khả năng tương thích đa nền tảng và sự hỗ trợ rộng rãi từ cộng đồng, Python là lựa chọn tối ưu không chỉ trong việc phát triển nhanh các ứng dụng phần mềm mà còn đặc biệt hữu ích trong việc xây dựng và triển khai các hệ thống IoT và ứng dụng nhúng hiện đại, giúp tiết kiệm đáng kể thời gian phát triển, tối ưu hóa hiệu suất hoạt động và nâng cao chất lượng sản phẩm cuối cùng.

Dưới đây là các ưu điểm nổi bật của ngôn ngữ lập trình Python:

- ❖ Cú pháp đơn giản, dễ hiểu: Python có cú pháp rõ ràng, ngắn gọn, gần với ngôn ngữ tự nhiên, giúp người mới học dễ dàng tiếp cận và người phát triển dễ dàng bảo trì, nâng cấp mã nguồn.
- ❖ Ngôn ngữ thông dịch: Python cho phép thực thi từng dòng mã trực tiếp, giúp lập trình viên nhanh chóng kiểm thử, sửa lỗi và triển khai các tính năng mới một cách hiệu quả và nhanh chóng.
- ❖ Hệ sinh thái thư viện mạnh mẽ và đa dạng: Python sở hữu rất nhiều thư viện mã nguồn mở chất lượng cao, hỗ trợ đa dạng các lĩnh vực như trí tuệ nhân tạo (TensorFlow, PyTorch), xử lý dữ liệu (Pandas, NumPy), trực quan hóa dữ liệu (Matplotlib, Seaborn), lập trình web (Django, Flask), và lập trình hệ thống nhúng (PySerial, MicroPython).
- ❖ Khả năng tương thích đa nền tảng: Các chương trình viết bằng Python có thể dễ dàng chạy trên nhiều hệ điều hành khác nhau như Windows, Linux,

macOS, Raspberry Pi OS, giúp tăng cường khả năng di động và dễ dàng triển khai trên nhiều nền tảng khác nhau.

- ❖ Quản lý bộ nhớ tự động: Python hỗ trợ cơ chế quản lý bộ nhớ tự động (garbage collection), giúp lập trình viên tránh các lỗi liên quan đến việc cấp phát và giải phóng bộ nhớ, từ đó giảm thiểu thời gian debug và nâng cao chất lượng phần mềm.
- ❖ Khả năng mở rộng và tích hợp linh hoạt: Python cho phép dễ dàng tích hợp với các ngôn ngữ lập trình khác như C, C++, Java thông qua các thư viện mở rộng như ctypes, Cython, JNI.
- ❖ Cộng đồng phát triển lớn và tích cực: Python được hỗ trợ bởi một cộng đồng phát triển đông đảo trên toàn thế giới.
- ❖ Ứng dụng rộng rãi trong lĩnh vực IoT và hệ thống nhúng: Python có khả năng giao tiếp hiệu quả với phần cứng nhúng thông qua các thư viện chuyên dụng như PySerial, GPIO, và MicroPython.
- ❖ Hiệu quả cho các ứng dụng trí tuệ nhân tạo và khoa học dữ liệu: Python là ngôn ngữ được sử dụng nhiều nhất trong lĩnh vực AI và khoa học dữ liệu.

Với những ưu điểm kể trên, Python xứng đáng là một lựa chọn hàng đầu cho việc phát triển các ứng dụng IoT, nhúng, trí tuệ nhân tạo và phần mềm ứng dụng tổng quát.

#### 2.4.4. *Firebase*



Hình 2.18. Firebase

Firebase là nền tảng dịch vụ điện toán đám mây toàn diện, cung cấp giải pháp Backend-as-a-Service (BaaS), được phát triển bởi Firebase Inc. vào năm 2011, sau đó được Google mua lại vào năm 2014. Firebase ra đời nhằm hỗ trợ việc xây

dựng nhanh chóng, triển khai dễ dàng và quản lý hiệu quả các ứng dụng web, di động, cũng như các thiết bị IoT. Thay vì phải tự quản lý toàn bộ cơ sở hạ tầng backend phức tạp, Firebase cho phép nhà phát triển tập trung vào phần front-end và trải nghiệm người dùng, giảm đáng kể chi phí phát triển, thời gian triển khai và các khó khăn trong vận hành.

Firebase cung cấp một hệ thống dịch vụ đồng bộ, tích hợp sâu rộng với các nền tảng phổ biến như Android, iOS, Web và thậm chí các ứng dụng nhúng. Nó tận dụng các API và SDK đơn giản nhưng mạnh mẽ, giúp người phát triển dễ dàng tích hợp các chức năng nâng cao như xác thực người dùng, phân tích dữ liệu, gửi thông báo push, và đồng bộ hóa dữ liệu thời gian thực mà không mất nhiều công sức lập trình. Điều này đặc biệt hữu ích cho các nhóm phát triển nhỏ hoặc các dự án startup với nguồn lực giới hạn.

Ngoài ra, Firebase sở hữu một cộng đồng hỗ trợ đông đảo và tích cực, với nhiều tài liệu hướng dẫn chi tiết, dễ hiểu cùng các diễn đàn thảo luận sôi nổi. Google cũng liên tục cập nhật và bổ sung thêm các tính năng mới vào Firebase nhằm đáp ứng tốt hơn nhu cầu ngày càng cao của các nhà phát triển và các ứng dụng đa nền tảng hiện đại. Vì thế, Firebase nhanh chóng trở thành một lựa chọn ưu tiên cho cả các ứng dụng nhỏ, đơn giản lẫn các hệ thống lớn, phức tạp cần quản lý hiệu quả trên quy mô toàn cầu.

Firebase nổi bật với hệ thống tính năng phong phú, toàn diện, và có khả năng đáp ứng hầu hết các nhu cầu phát triển ứng dụng hiện đại. Một số tính năng quan trọng và đặc biệt hữu ích bao gồm:

- ❖ Realtime Database: Lưu trữ và đồng bộ dữ liệu tức thời giữa các client kết nối thông qua WebSocket.
- ❖ Cloud Firestore: Cơ sở dữ liệu dạng NoSQL dựa trên tài liệu (document-based), hỗ trợ khả năng truy vấn phức tạp hơn và khả năng mở rộng cao hơn Realtime Database.
- ❖ Firebase Authentication: Cung cấp giải pháp xác thực linh hoạt, đa dạng phương thức như email/password, số điện thoại, Google, Facebook, Twitter, GitHub, giúp người dùng dễ dàng đăng nhập ứng dụng.

- ❖ Cloud Storage: Cho phép lưu trữ dữ liệu file lớn một cách an toàn, đáng tin cậy, hỗ trợ upload, download nhanh chóng và phân phối nội dung đa khu vực.
- ❖ Firebase Hosting: Cung cấp hosting nhanh, bảo mật, hỗ trợ CDN tích hợp sẵn để lưu trữ và phân phối nội dung web tĩnh và động với tốc độ truy cập cao.
- ❖ Cloud Functions: Viết mã backend dưới dạng hàm đơn giản chạy trực tiếp trên hạ tầng Google Cloud, được kích hoạt dựa vào sự kiện (event-driven) để tự động thực hiện các tác vụ backend mà không cần cấu hình máy chủ riêng.
- ❖ Firebase Analytics (Google Analytics for Firebase): Công cụ phân tích người dùng mạnh mẽ, cung cấp các báo cáo chuyên sâu về hành vi và tương tác người dùng với ứng dụng, giúp tối ưu hóa trải nghiệm và hiệu quả hoạt động.
- ❖ Firebase Cloud Messaging (FCM): Gửi thông báo push hiệu quả, hỗ trợ đầy đủ trên Android, iOS, Web, giúp giao tiếp với người dùng nhanh chóng và hiệu quả hơn.
- ❖ Firebase Remote Config: Cho phép điều chỉnh tham số ứng dụng từ xa, giúp thử nghiệm A/B, cá nhân hóa trải nghiệm người dùng, mà không cần cập nhật ứng dụng mới.
- ❖ Crashlytics và Performance Monitoring: Giúp giám sát lỗi ứng dụng và hiệu suất thực tế, phát hiện và khắc phục vấn đề nhanh chóng, duy trì chất lượng trải nghiệm người dùng ở mức cao nhất.

Realtime Database là dịch vụ nổi bật nhất của Firebase, đóng vai trò cơ sở dữ liệu NoSQL thời gian thực, sử dụng JSON làm định dạng lưu trữ chính, giúp đồng bộ nhanh chóng và hiệu quả giữa các client. Khác với các cơ sở dữ liệu truyền thống thường yêu cầu client chủ động truy vấn theo chu kỳ để cập nhật dữ liệu, Firebase Realtime Database cho phép tự động đồng bộ dữ liệu ngay lập tức giữa tất cả các client theo cơ chế event-driven, giảm thiểu độ trễ và nâng cao khả năng phản hồi của ứng dụng.



Hình 2.19. Firebase Realtime Database

Dữ liệu được lưu trữ theo cấu trúc cây JSON, cung cấp một cách lưu trữ dữ liệu linh hoạt, rõ ràng, dễ dàng quản lý, đặc biệt phù hợp với các dữ liệu có cấu trúc phân cấp. Firebase cũng cung cấp hệ thống Rule-based Security linh hoạt, cho phép định nghĩa quyền truy cập một cách chi tiết, đảm bảo dữ liệu luôn an toàn và chỉ được truy cập bởi những client được cho phép.

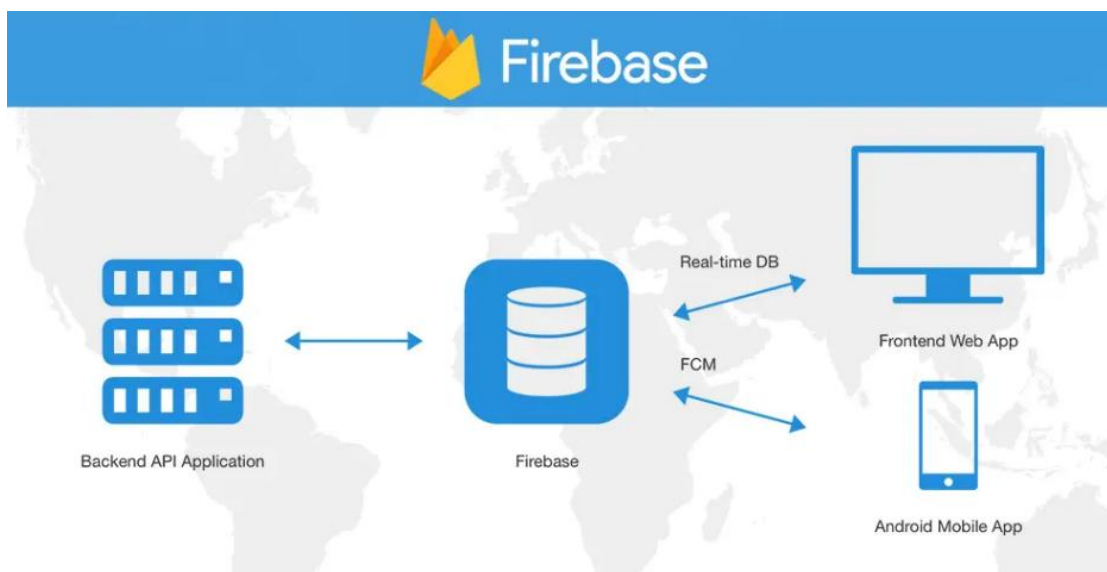
Realtime Database đặc biệt thích hợp với các ứng dụng cần đồng bộ dữ liệu liên tục, cập nhật tức thời như các ứng dụng chat, mạng xã hội, game thời gian thực, và hệ thống giám sát IoT.

Firebase Realtime Database hoạt động trên mô hình client-server với cơ chế đồng bộ dữ liệu theo thời gian thực dựa trên các kết nối WebSocket. Khi một client kết nối vào Firebase, nó mở ra một đường truyền liên tục và trực tiếp đến server Firebase, giúp quá trình đồng bộ dữ liệu diễn ra ngay lập tức khi có sự thay đổi:

- ❖ Client thực hiện ghi (write) hoặc cập nhật (update) dữ liệu vào một node cụ thể trong cơ sở dữ liệu.
- ❖ Firebase server nhận thông tin cập nhật này và ngay lập tức ghi lại trạng thái mới của dữ liệu vào bộ nhớ của mình.
- ❖ Sau đó, Firebase server lập tức thông báo qua kết nối WebSocket đến tất cả các client khác đang “lắng nghe” (subscribe) vào node dữ liệu tương ứng.
- ❖ Client khi nhận được thông báo sẽ tự động tải xuống các thay đổi này, cập nhật trạng thái ứng dụng mà không cần gửi yêu cầu truy vấn dữ liệu mới.
- ❖ Toàn bộ quá trình này diễn ra hoàn toàn tự động, tức thời, liên tục và không cần client phải liên tục gửi truy vấn, giảm thiểu đáng kể băng thông mạng và độ trễ.

Ngoài ra, Realtime Database có khả năng lưu trữ dữ liệu cục bộ trên client, giúp ứng dụng vẫn tiếp tục hoạt động ngay cả khi offline. Mọi thay đổi khi offline sẽ được ghi lại vào cache nội bộ và tự động đồng bộ lên server Firebase ngay khi kết nối internet được khôi phục.

Cơ chế đồng bộ thời gian thực này là nền tảng cho rất nhiều ứng dụng hiện đại, từ ứng dụng cộng tác làm việc nhóm, ứng dụng giám sát hệ thống IoT theo thời gian thực, ứng dụng chat tức thì, cho tới các hệ thống theo dõi và cập nhật trạng thái liên tục như các nền tảng thương mại điện tử, quản lý chuỗi cung ứng hay ứng dụng y tế cá nhân.



Hình 2.20. Mô hình Firebase Realtime Database

#### 2.4.5. Ngôn ngữ lập trình Dart và Framework Flutter

Ngôn ngữ lập trình Dart là một ngôn ngữ lập trình hướng đối tượng, đa nền tảng, được thiết kế và phát triển bởi Google, chính thức công bố lần đầu tiên vào năm 2011. Ban đầu Dart được tạo ra nhằm giải quyết các hạn chế hiệu năng của JavaScript, đặc biệt là trong các ứng dụng web lớn. Tuy nhiên, sau khi được lựa chọn làm ngôn ngữ nền tảng cho framework Flutter, Dart ngày càng trở nên phổ biến và được sử dụng rộng rãi hơn trong việc phát triển ứng dụng di động và đa nền tảng.

Dart sở hữu cú pháp rõ ràng, dễ đọc, dễ hiểu và gần với các ngôn ngữ lập trình phổ biến như Java, JavaScript và C#. Điều này giúp lập trình viên nhanh chóng tiếp cận và giảm đáng kể thời gian học tập. Dart hỗ trợ biên dịch hai chế

độ: biên dịch tức thời (Just-in-Time - JIT) giúp nhà phát triển nhanh chóng thử nghiệm ứng dụng ngay trong quá trình lập trình, và biên dịch trước (Ahead-of-Time - AOT) để tối ưu hóa hiệu suất thực thi trên thiết bị di động và các nền tảng khác. Dart còn cung cấp tính năng quản lý bộ nhớ tự động thông qua cơ chế Garbage Collection, giúp lập trình viên giảm thiểu các vấn đề về quản lý bộ nhớ và hạn chế lỗi runtime.



Hình 2.21. Ngôn ngữ lập trình Dart

Flutter là một framework phát triển ứng dụng đa nền tảng, được xây dựng trên nền ngôn ngữ Dart, do Google phát hành vào năm 2017. Flutter hỗ trợ lập trình viên tạo ra các ứng dụng chất lượng cao trên nhiều nền tảng khác nhau như Android, iOS, Web, Windows, macOS và Linux chỉ từ một bộ mã nguồn duy nhất. Điều này không chỉ tiết kiệm thời gian, chi phí phát triển, mà còn tối ưu hóa hiệu quả quá trình bảo trì và cập nhật ứng dụng.

Flutter sử dụng bộ engine đồ họa hiệu năng cao gọi là Skia để vẽ và hiển thị giao diện người dùng, hoàn toàn độc lập với các thành phần giao diện gốc của hệ điều hành. Chính vì vậy, Flutter mang lại hiệu suất đồ họa mượt mà, đồng nhất trên mọi thiết bị và nền tảng, loại bỏ được các rào cản về hiệu năng khi so sánh với các framework đa nền tảng khác. Ngoài ra, Flutter cung cấp một bộ thư viện widget đồ họa phong phú, hỗ trợ nhiều loại giao diện từ cơ bản đến phức tạp, dễ dàng tùy chỉnh, tối ưu và đáp ứng mọi nhu cầu thiết kế ứng dụng đa dạng.

Ngôn ngữ Dart có nhiều đặc điểm nổi bật giúp lập trình viên nhanh chóng xây dựng các ứng dụng chất lượng cao và dễ bảo trì, cụ thể như sau:

- ❖ Cú pháp trực quan và dễ tiếp cận: Dart cung cấp cú pháp rõ ràng, cấu trúc logic chặt chẽ, giúp lập trình viên dễ dàng hiểu và quản lý mã nguồn ngay cả với các ứng dụng có độ phức tạp cao.

- ❖ Hỗ trợ lập trình hướng đối tượng đầy đủ: Dart tuân thủ đầy đủ các nguyên lý lập trình hướng đối tượng như đóng gói (encapsulation), kế thừa (inheritance), đa hình (polymorphism), và trừu tượng (abstraction), giúp người lập trình dễ dàng quản lý và mở rộng mã nguồn.
- ❖ Cơ chế biên dịch linh hoạt (JIT và AOT): Dart có khả năng biên dịch JIT trong quá trình phát triển, cho phép lập trình viên nhanh chóng kiểm tra và chỉnh sửa mã nguồn. Ngược lại, AOT cung cấp hiệu suất thực thi tối ưu, giảm thời gian khởi động ứng dụng và tăng tốc độ xử lý.
- ❖ Xử lý bất đồng bộ hiệu quả: Dart hỗ trợ mạnh mẽ việc xử lý các tác vụ bất đồng bộ thông qua cú pháp `async`, `await`, và kiểu dữ liệu `Future` và `Stream`, giúp ứng dụng phản hồi nhanh hơn và quản lý tài nguyên hiệu quả hơn khi xử lý dữ liệu mạng, truy cập cơ sở dữ liệu và các tác vụ đòi hỏi thời gian chờ khác.
- ❖ An toàn về kiểu dữ liệu (Type Safety): Dart hỗ trợ kiểm tra kiểu dữ liệu tĩnh (static typing) và cả kiểu động (dynamic typing), giúp giảm thiểu lỗi runtime và tăng độ ổn định cho ứng dụng.
- ❖ Quản lý bộ nhớ tự động (Garbage Collection): Dart tích hợp trình quản lý bộ nhớ tự động, giúp giảm thiểu gánh nặng cho lập trình viên, hạn chế các vấn đề rò rỉ bộ nhớ và lỗi liên quan đến việc quản lý thủ công bộ nhớ.



Hình 2.22. Khung Flutter

Flutter là một framework phát triển ứng dụng đa nền tảng tiên tiến và mạnh mẽ, do Google phát hành lần đầu vào năm 2017. Flutter được thiết kế để đáp ứng nhu cầu xây dựng nhanh chóng các ứng dụng chất lượng cao trên nhiều nền tảng



như Android, iOS, Web, Windows, Linux và macOS chỉ với một cơ sở mã nguồn duy nhất. Nhờ đó, Flutter nhanh chóng trở thành một trong những công cụ phổ biến nhất trong lĩnh vực phát triển ứng dụng hiện đại, với mức độ phổ biến và tốc độ tăng trưởng ngày càng mạnh mẽ.

Flutter sử dụng ngôn ngữ lập trình Dart làm ngôn ngữ lập trình chính, giúp tận dụng tối đa khả năng biên dịch linh hoạt (AOT và JIT) để tối ưu hiệu suất ứng dụng và trải nghiệm phát triển nhanh chóng. Điểm đặc biệt của Flutter là cơ chế hoạt động dựa trên một engine đồ họa độc lập (Skia Graphics Engine), cho phép các ứng dụng được tạo ra luôn giữ được hiệu suất cao, mượt mà, và nhất quán trên tất cả các thiết bị và nền tảng hỗ trợ.

Dưới đây là những ưu điểm chính khiến Flutter trở nên đặc biệt hấp dẫn đối với cộng đồng phát triển:

- ❖ Phát triển đa nền tảng hiệu quả: Flutter cho phép các nhà phát triển viết một bộ mã nguồn duy nhất và triển khai đồng thời trên nhiều nền tảng khác nhau bao gồm Android, iOS, Web, Desktop (Windows, macOS, Linux), và các thiết bị nhúng. Điều này không chỉ giúp rút ngắn thời gian phát triển và thử nghiệm, mà còn giúp tối ưu hóa chi phí bảo trì và cập nhật ứng dụng. Flutter còn giúp đảm bảo tính nhất quán cao về giao diện và trải nghiệm người dùng giữa các nền tảng khác nhau.
- ❖ Hiệu năng vượt trội nhờ kiến trúc độc đáo: Khác với các framework đa nền tảng truyền thống dựa trên WebView hoặc JavaScript Bridge (ví dụ như React Native), Flutter sử dụng bộ engine đồ họa Skia được viết bằng C++ để vẽ trực tiếp giao diện lên màn hình thiết bị. Cách tiếp cận này giúp loại bỏ hoàn toàn các lớp trung gian và tối ưu hóa đáng kể hiệu suất đồ họa, giảm độ trễ, đảm bảo trải nghiệm người dùng mượt mà gần như ứng dụng native. Các animation, chuyển cảnh và tương tác phức tạp đều có thể được xử lý hiệu quả và nhanh chóng nhờ cơ chế này. Tạo ra những trải nghiệm thị giác rất mượt mà.
- ❖ Hot Reload giúp phát triển nhanh chóng và trực quan: Một trong những tính năng hấp dẫn nhất của Flutter là khả năng "Hot Reload", cho phép lập trình

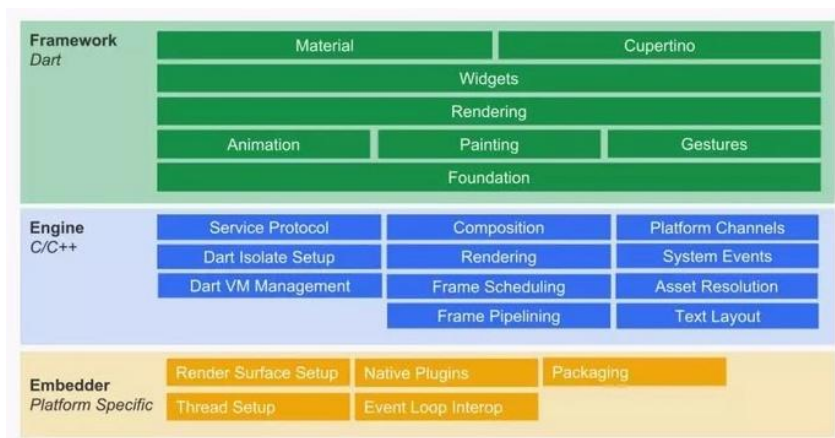
viên xem ngay lập tức kết quả thay đổi khi chỉnh sửa mã nguồn mà không cần khởi động lại ứng dụng. Hot Reload giúp rút ngắn đáng kể chu kỳ phát triển, tạo điều kiện thuận lợi cho các nhà thiết kế và lập trình viên thử nghiệm nhiều ý tưởng thiết kế, sửa lỗi nhanh chóng, và cải tiến ứng dụng liên tục trong thời gian ngắn nhất. Tính năng này đặc biệt hữu ích trong các dự án lớn, có yêu cầu cập nhật thường xuyên và cải tiến tối ưu giao diện liên tục.

- ❖ Thư viện widget đồ họa đa dạng, linh hoạt và mạnh mẽ: Flutter cung cấp một bộ thư viện widget rất lớn và linh hoạt, được xây dựng dựa trên nguyên tắc "Everything is a widget" (mọi thứ đều là widget). Các widget của Flutter bao gồm từ các thành phần giao diện cơ bản như nút nhấn, văn bản, hình ảnh, đến các thành phần giao diện phức tạp như bảng điều khiển, thanh trượt, biểu đồ, hiệu ứng animation tùy chỉnh. Các widget này có khả năng tùy biến cao, dễ dàng sử dụng và có thể kết hợp để tạo ra các giao diện người dùng sáng tạo, đẹp mắt, trực quan, đáp ứng nhu cầu của cả người dùng lẫn nhà phát triển.
- ❖ Cộng đồng phát triển đông đảo và hỗ trợ mạnh mẽ từ Google: Flutter nhận được sự hậu thuẫn mạnh mẽ từ Google và sở hữu một cộng đồng phát triển rất đông đảo trên toàn thế giới. Google liên tục cập nhật, cải tiến các tính năng, sửa lỗi và tối ưu hiệu suất cho Flutter thông qua các bản cập nhật thường xuyên. Flutter còn có một hệ sinh thái rộng lớn với nhiều thư viện mã nguồn mở chất lượng cao, giúp dễ dàng tích hợp các dịch vụ bên ngoài như Firebase, RESTful API, SQLite, cũng như các công cụ phân tích và quản lý trạng thái phổ biến.
- ❖ Khả năng mở rộng và tích hợp linh hoạt: Flutter cho phép dễ dàng tích hợp các tính năng nền tảng đặc trưng thông qua các plugin native (Platform Channels). Điều này giúp các ứng dụng Flutter tận dụng tối đa sức mạnh từ các API nền tảng như camera, GPS, cảm biến, lưu trữ, notification, và các tính năng đặc biệt khác mà không gặp bất kỳ trở ngại đáng kể nào về hiệu suất hoặc trải nghiệm người dùng.
- ❖ Quản lý trạng thái ứng dụng linh hoạt và hiệu quả: Flutter cung cấp các phương pháp quản lý trạng thái ứng dụng hiện đại và hiệu quả, hỗ trợ nhiều

mô hình quản lý trạng thái phổ biến như Provider, BLoC, Redux, MobX, GetX. Các mô hình này giúp quản lý trạng thái một cách rõ ràng, tách biệt logic nghiệp vụ khỏi giao diện người dùng, từ đó giúp ứng dụng dễ dàng mở rộng, bảo trì, và giảm thiểu các lỗi phát sinh trong quá trình phát triển ứng dụng.

- ❖ Công cụ kiểm thử (Testing) toàn diện và mạnh mẽ: Flutter cung cấp đầy đủ các công cụ và thư viện hỗ trợ kiểm thử tự động toàn diện từ unit test, widget test đến integration test.

Như vậy, với những ưu điểm nổi bật kể trên, Flutter trở thành một trong những lựa chọn hàng đầu trong phát triển ứng dụng đa nền tảng hiện đại. Framework này không chỉ giúp các nhà phát triển xây dựng nhanh chóng các ứng dụng đẹp, hiệu suất cao, mà còn tối ưu hóa mạnh mẽ hiệu quả kinh tế và thời gian phát triển, giúp các sản phẩm công nghệ có thể nhanh chóng tiếp cận người dùng một cách hiệu quả nhất.



Hình 2.23. Kiến trúc của Flutter

Framework Layer (Lớp ứng dụng - Dart): Đây là lớp trên cùng của Flutter, nơi mà lập trình viên trực tiếp tương tác để xây dựng giao diện và logic ứng dụng. Flutter Framework được viết hoàn toàn bằng ngôn ngữ Dart, cung cấp các công cụ mạnh mẽ và linh hoạt để thiết kế giao diện người dùng (UI). Framework này bao gồm các thư viện widget phong phú và đa dạng, giúp việc xây dựng các giao diện người dùng trực quan, đẹp mắt và dễ dàng tùy biến theo yêu cầu.

Các widget trong Flutter được tổ chức thành các cây widget (widget tree), mỗi widget đóng vai trò như một thành phần cấu tạo nên giao diện ứng dụng. Các

widget này có thể là `StatelessWidget` (không thay đổi trạng thái) hoặc `StatefulWidget` (thay đổi trạng thái), cho phép quản lý hiệu quả trạng thái ứng dụng khi tương tác với người dùng.

Flutter Framework cũng bao gồm các hệ thống quản lý trạng thái mạnh mẽ như `Provider`, `BLoC`, `Redux`, `GetX`, giúp ứng dụng phản hồi tốt với sự thay đổi dữ liệu. Ngoài ra, Framework Layer cũng quản lý các vấn đề bất đồng bộ, xử lý sự kiện và định tuyến (routing) giữa các màn hình trong ứng dụng thông qua `Navigator`.

Engine Layer (Lớp xử lý đồ họa và lỗi - C++): Bên dưới Framework Layer là Flutter Engine, được viết chủ yếu bằng C++ nhằm đảm bảo hiệu suất xử lý đồ họa tối ưu. Flutter Engine sử dụng thư viện đồ họa Skia – một thư viện đồ họa mã nguồn mở mạnh mẽ, hiệu suất cao được Google sử dụng trong nhiều sản phẩm nổi bật như Chrome và Android – để vẽ và hiển thị giao diện ứng dụng lên màn hình thiết bị.

Engine chịu trách nhiệm biên dịch mã Dart từ Dart VM thành mã máy native hiệu suất cao (biên dịch Ahead-of-Time - AOT). Flutter Engine còn quản lý các tác vụ quan trọng khác như xử lý animation, hiệu ứng chuyển cảnh (transition), quản lý bộ nhớ hiệu quả, xử lý text rendering và layout phức tạp.

Ngoài ra, Engine Layer còn quản lý sự kiện đầu vào từ người dùng như thao tác cảm ứng, chuột, bàn phím, và chuyển những sự kiện này đến Framework Layer để xử lý logic tương tác. Điều này đảm bảo khả năng phản hồi nhanh và mượt mà trong quá trình tương tác người dùng với ứng dụng.

Embedder Layer (Lớp tương tác nền tảng): Embedder là lớp thấp nhất của kiến trúc Flutter, chịu trách nhiệm tương tác trực tiếp với hệ điều hành của từng nền tảng cụ thể (Android, iOS, Web, Windows, Linux, macOS). Embedder đảm nhiệm việc khởi tạo ứng dụng, quản lý vòng đời ứng dụng (lifecycle events) như khởi động, tạm dừng, tiếp tục, kết thúc ứng dụng, và cung cấp các dịch vụ liên quan đến nền tảng như camera, bộ nhớ, truy cập tập tin, âm thanh, cảm biến.

Flutter cung cấp sẵn một bộ embedder riêng biệt cho từng nền tảng hỗ trợ, cho phép ứng dụng Flutter tận dụng tối đa hiệu suất và khả năng tương thích với

nền tảng gốc. Embedder layer sử dụng các API native để tương tác trực tiếp với hệ điều hành, giúp ứng dụng Flutter có trải nghiệm tương đương ứng dụng native về hiệu suất và độ ổn định.

Ngôn ngữ lập trình Dart kết hợp với Framework Flutter ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực thực tế nhờ vào những ưu điểm vượt trội về hiệu suất, khả năng phát triển nhanh chóng và khả năng tương thích đa nền tảng mạnh mẽ. Với những ưu điểm và khả năng như vậy thì sau đây là những ứng dụng chính mà Dart và Flutter được ứng dụng:

**Ứng dụng di động (Mobile Applications):** Flutter hiện được ứng dụng rộng rãi nhất trong việc phát triển ứng dụng di động đa nền tảng. Các doanh nghiệp lớn như Google, Alibaba, BMW, eBay, Grab, Reflectly đều lựa chọn Flutter để xây dựng các ứng dụng di động nhờ vào hiệu suất cao, thời gian phát triển nhanh, và khả năng duy trì tính nhất quán giao diện và trải nghiệm người dùng trên cả hai nền tảng Android và iOS. Flutter giúp các nhà phát triển tiết kiệm đáng kể thời gian và chi phí do chỉ cần duy trì một bộ mã nguồn duy nhất mà vẫn đảm bảo ứng dụng có hiệu suất gần như tương đương native.

**Ứng dụng web (Web Applications):** Flutter Web cho phép tạo các ứng dụng web có giao diện phong phú, độ tương tác cao với trải nghiệm đồng nhất với ứng dụng di động. Thay vì sử dụng các framework web truyền thống như React hay Angular, Flutter Web cho phép tái sử dụng mã nguồn Dart từ ứng dụng di động để xây dựng nhanh chóng các ứng dụng web, giúp giảm đáng kể thời gian phát triển. Ứng dụng Flutter Web thích hợp với các dự án yêu cầu giao diện đẹp, hiệu ứng animation mượt mà như các trang landing page, dashboard quản lý, hay các hệ thống nội bộ doanh nghiệp.

**Ứng dụng desktop (Desktop Applications):** Flutter hiện hỗ trợ phát triển ứng dụng desktop native trên các hệ điều hành như Windows, macOS, Linux. Điều này mở rộng đáng kể khả năng ứng dụng của Flutter, cho phép xây dựng các ứng dụng quản lý, các ứng dụng văn phòng, công cụ đa năng trên nhiều nền tảng khác nhau, đồng thời tái sử dụng hầu hết mã nguồn từ ứng dụng di động, giúp giảm đáng kể thời gian và chi phí bảo trì.

Ứng dụng IoT và hệ thống nhúng: Flutter và Dart cũng được ứng dụng vào các thiết bị IoT, hệ thống nhúng và bảng điều khiển thông minh. Nhờ giao diện linh hoạt, đẹp mắt, và khả năng tối ưu hóa tài nguyên tốt, Flutter cho phép tạo giao diện người dùng trực quan, dễ thao tác cho các thiết bị y tế thông minh, nhà thông minh, thiết bị đo đạc và giám sát môi trường. Một số ứng dụng cụ thể bao gồm các hệ thống điều khiển trung tâm thông minh, màn hình tương tác trên xe hơi, và các thiết bị giám sát sức khỏe cá nhân.

Lĩnh vực khác: Flutter và Dart còn được sử dụng rộng rãi trong các ứng dụng trò chơi đơn giản, ứng dụng giáo dục tương tác, và các ứng dụng AR/VR nhờ khả năng xử lý đồ họa hiệu suất cao và khả năng tích hợp dễ dàng với các thư viện đồ họa mở rộng.

Với khả năng tương thích đa nền tảng vượt trội, hệ sinh thái rộng lớn và hiệu suất xử lý mạnh mẽ, Dart và Flutter đã và đang trở thành lựa chọn hàng đầu cho các doanh nghiệp và nhà phát triển ứng dụng trên toàn thế giới.

## CHƯƠNG 3. THIẾT KẾ VÀ THI CÔNG

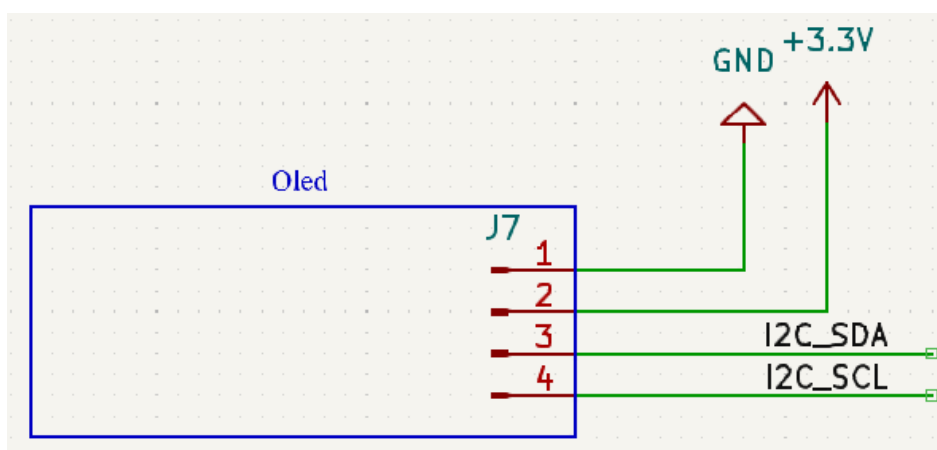
### 3.1. Thiết kế phần cứng

#### 3.1.1. Sơ đồ nguyên lý của hệ thống

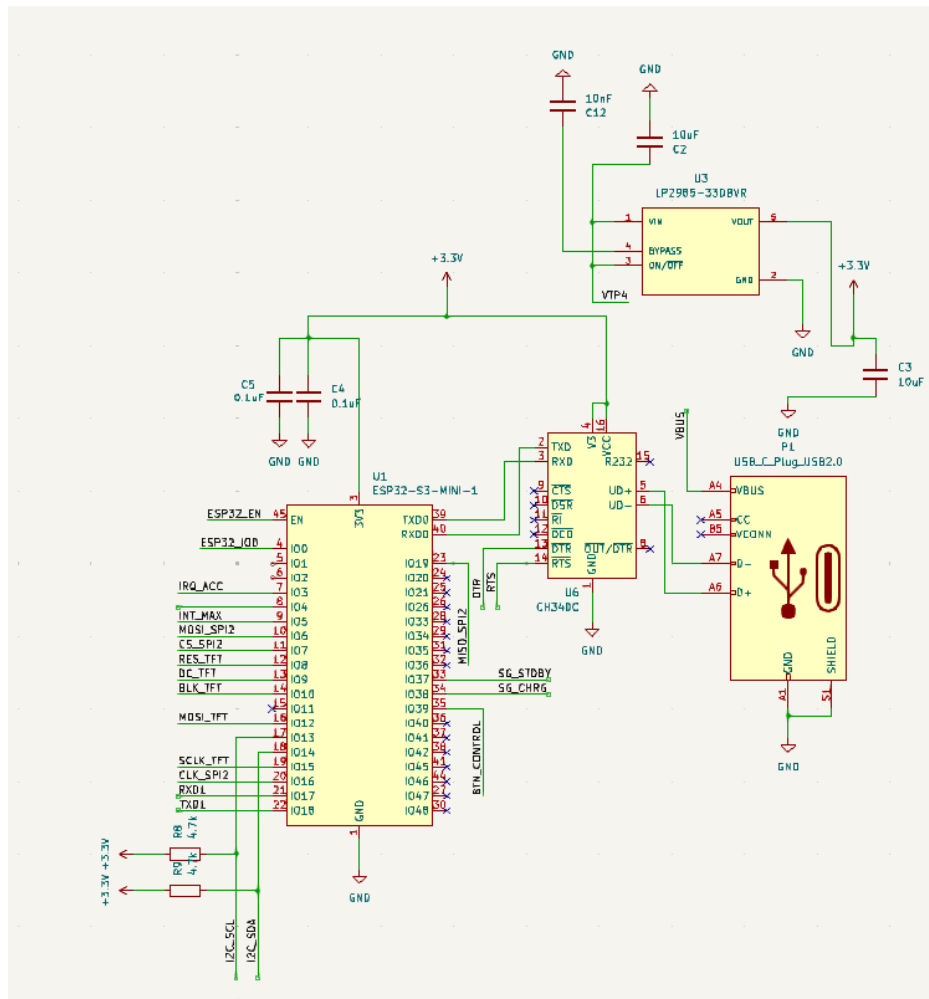
Sơ đồ nguyên lý của các khối trong hệ thống được trình bày trên các hình từ hình 3.1 đến 3.4. Các chân của module ESP32-S3-MINI-1 được đấu nối với các chân của các linh kiện và thiết bị khác trong hệ thống theo bảng 3.1.

Bảng 3.1. Bảng kết nối ESP32-S3-MIN-1 với các linh kiện

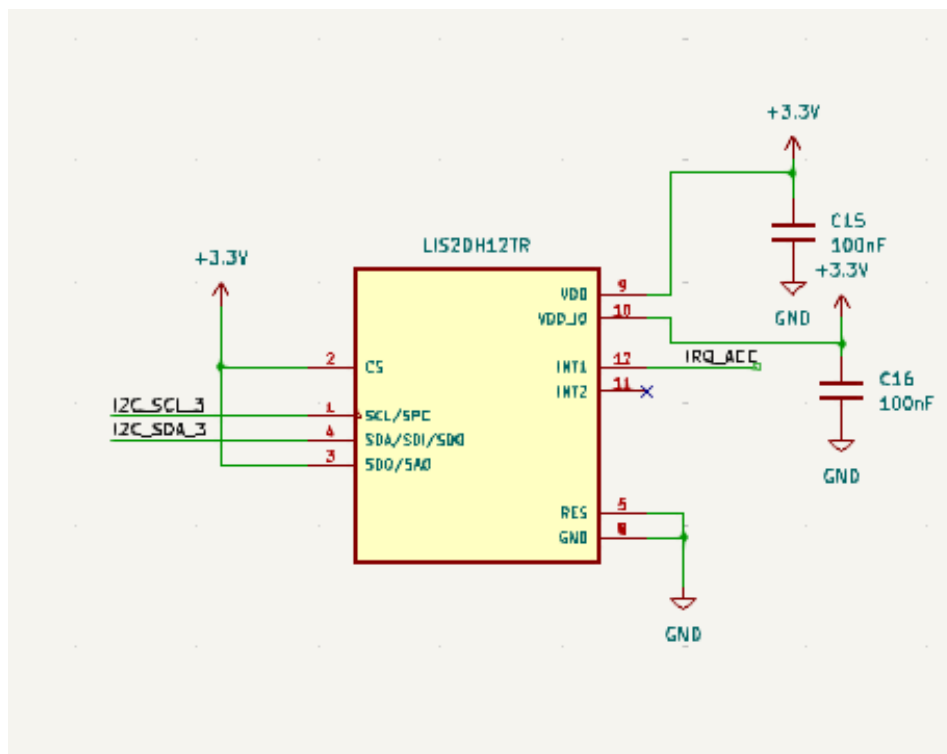
Chân	Kết nối	Linh kiện và thiết bị
GND	GND	Nguồn -
3V3	+3.3 V (nguồn cấp)	Nguồn +
IO3	IRQ_ACC (ngắt ACC)	LIS2DH12TR
IO5	INT_MAX	MAX30102
IO13	I2C_SCL	LIS2DH12TR MAX30102 Oled
IO14	I2C_SDA	LIS2DH12TR MAX30102 Oled



Hình 3.1. Sơ đồ nguyên lý khối hiển thị màn hình OLED

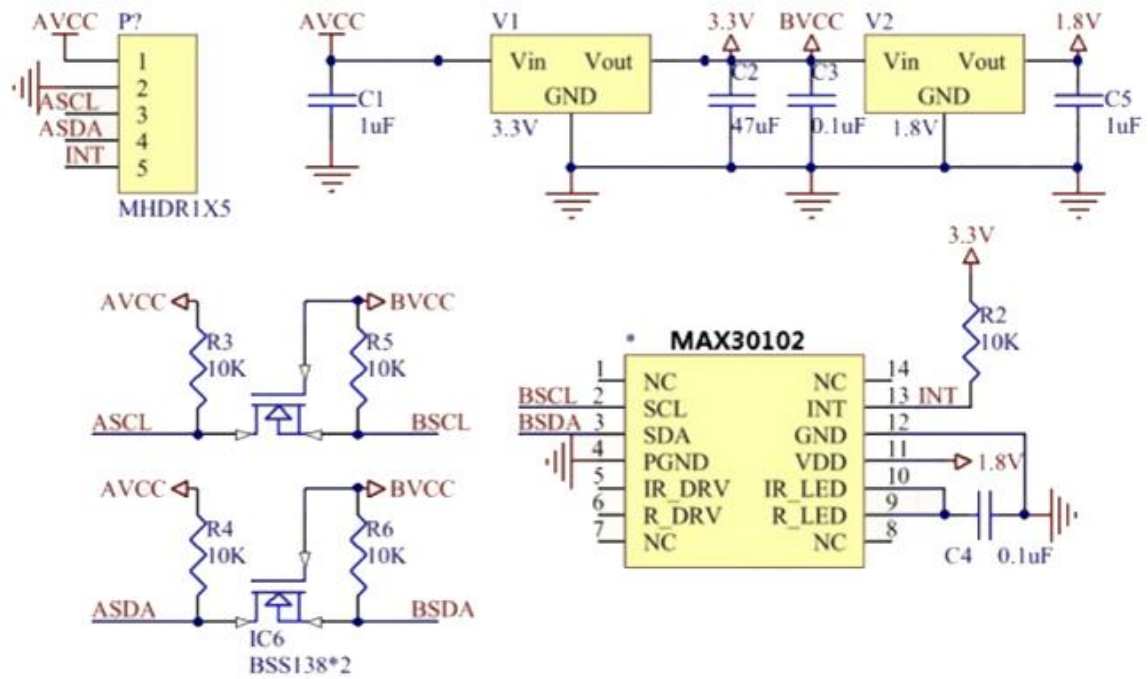


Hình 3.2. Sơ đồ nguyên lý khối vi điều khiển và nguồn



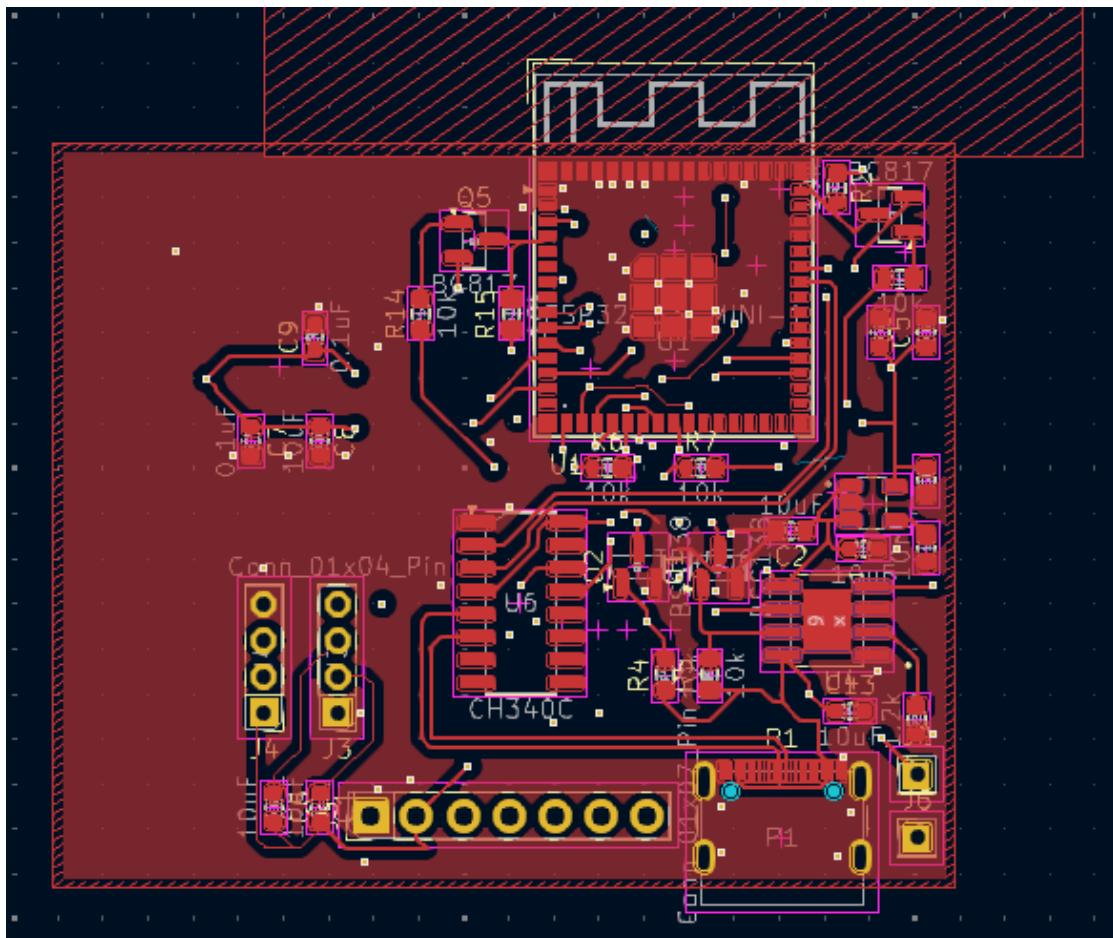
Hình 3.3. Sơ đồ nguyên lý khối cảm biến gia tốc 3 trục



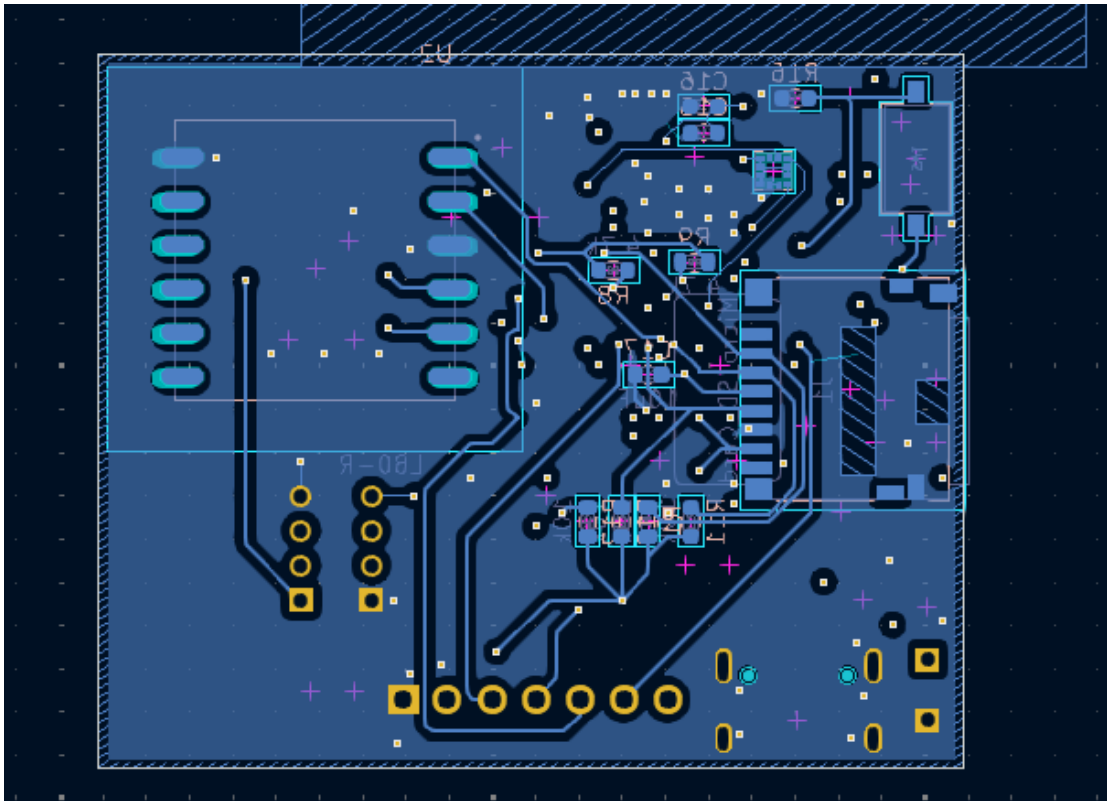


Hình 3.4. Sơ đồ nguyên lý khối cảm biến nhịp tim và nồng độ oxy trong máu

### 3.1.2. Bản vẽ PCB của thiết bị

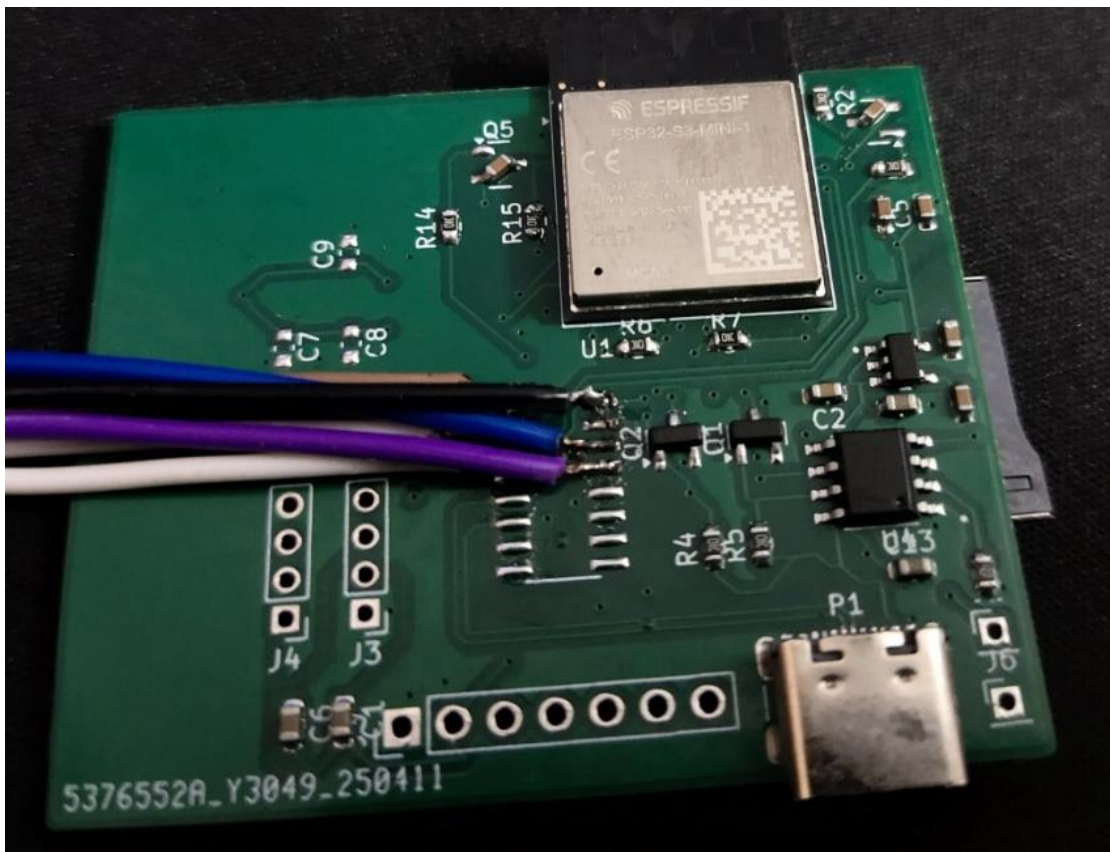


Hình 3.5. Mặt sau bảng PCB

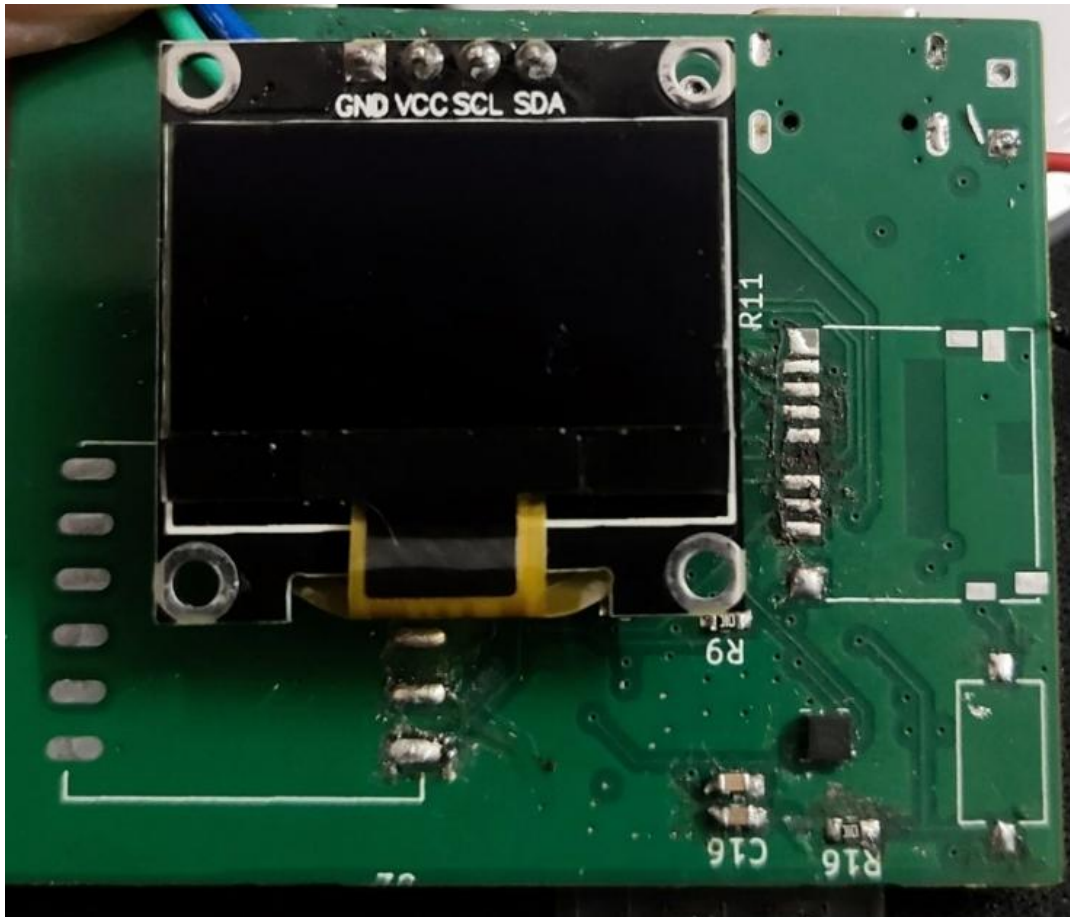


Hình 3.6. Mặt trước bảng PCB

### 3.1.2. Thi công thiết bị



Hình 3.7. Mạch điều khiển hoàn thiện - mặt sau

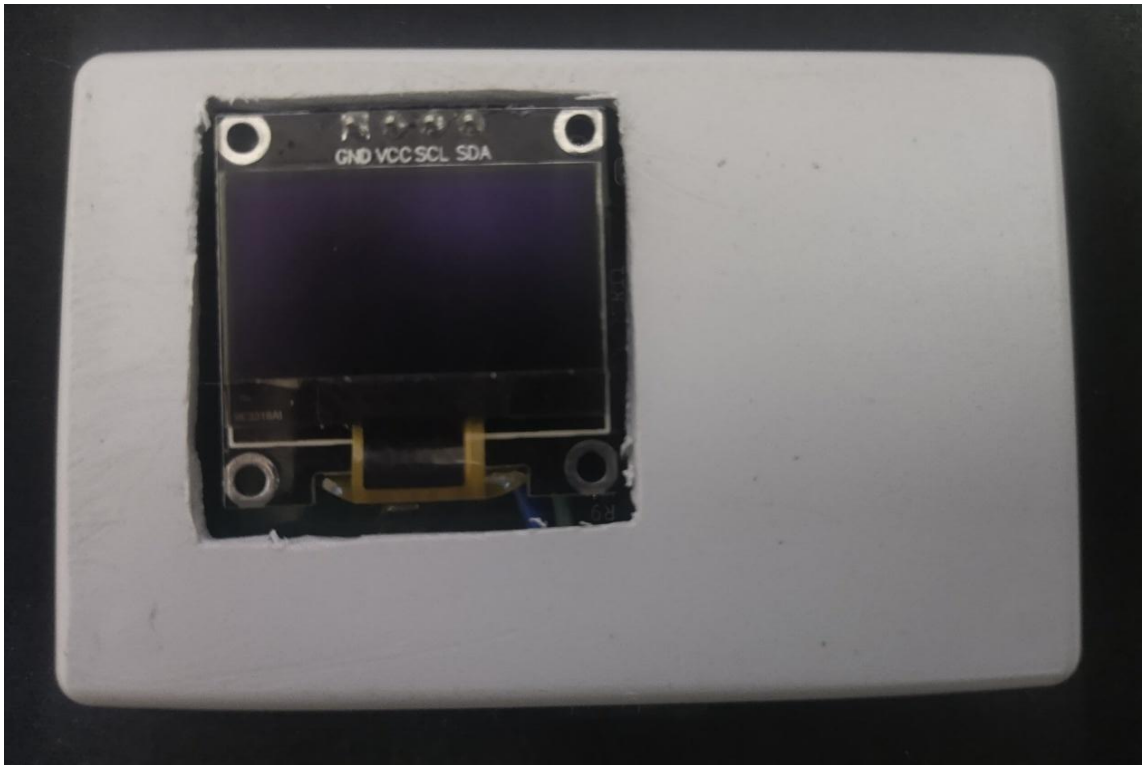


Hình 3.8. Mạch điều khiển hoàn thiện - mặt trước



Hình 3.9. Lắp đặt mạch vào trong hộp





Hình 3.10. Sản phẩm hoàn chỉnh

### 3.2. Thiết kế phần mềm

#### 3.2.1. Lưu đồ thuật toán và nguyên lý hoạt động

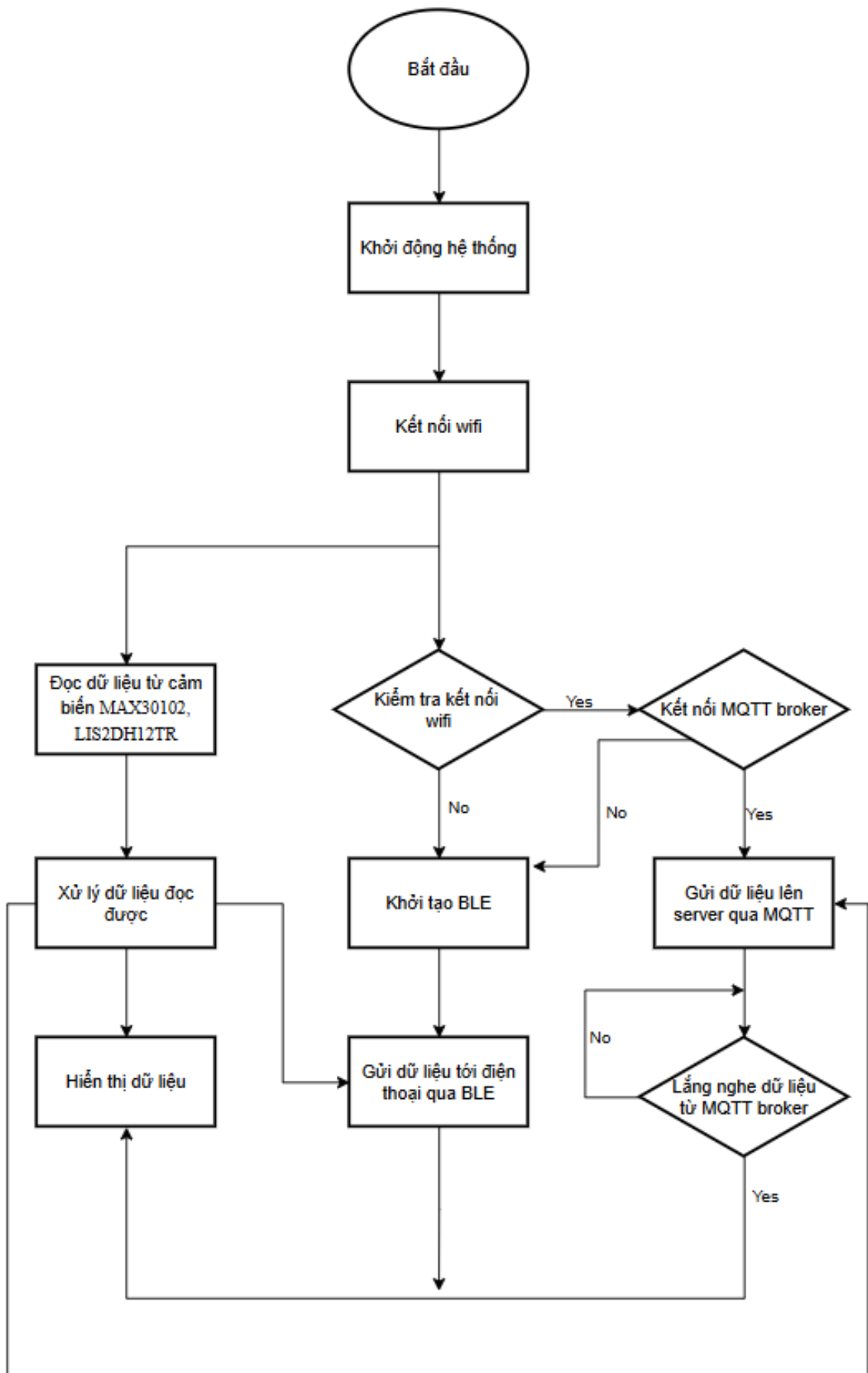
Nguyên lý hoạt động:

Sau khi cấp nguồn, thiết bị sẽ tiến hành quá trình khởi tạo kết nối mạng không dây wifi và BLE, kết nối với các cảm biến và MQTT broker. Quá trình đo thông số nhịp tim, nồng độ oxy trong máu và gia tốc 3 trục sẽ diễn ra liên tục và tổng hợp lại kết quả mỗi phút. Từ những chỉ số đo được sẽ được tính toán trực tiếp tại thiết bị để ra được chỉ số năng lượng cơ thể đã tiêu hao và lưu chúng lại vào bộ nhớ NVS flash của ESP32-S3-MINI-1. Các chỉ số sẽ được gửi lên Server qua MQTT broker khi có kết nối wifi và BLE tới điện thoại di động khi không có kết nối wifi. Các chỉ số sẽ được cập nhật và hiển thị trên màn hình Oled của thiết bị theo thời gian thực với chu kỳ mỗi 1 phút 1 lần trong.

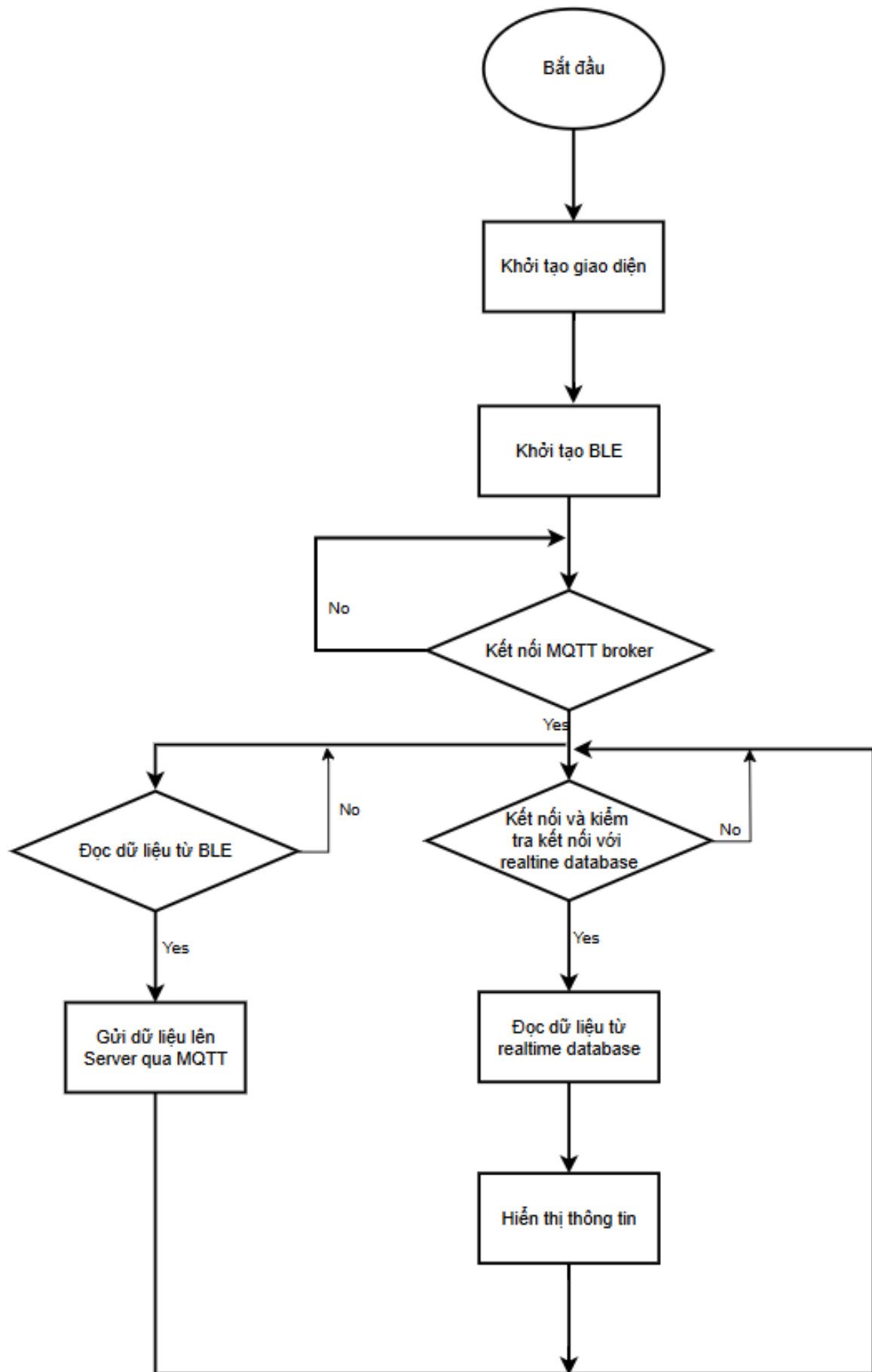
Trên ứng dụng di động, sau khi đăng nhập thành công vào tài khoản rồi sẽ lấy dữ liệu từ Firebase Realtime Database về để hiển thị lên giao diện ứng dụng. Song song với đó ứng dụng sẽ quét, lọc và đọc liên tục dữ liệu quảng bá từ BLE để lấy được dữ liệu từ thiết bị khi mất kết nối. Các dữ liệu lấy được từ thiết bị sẽ được gửi lên Server qua MQTT broker.

Tại Server khi bắt đầu sẽ khởi động các đối tượng thiết bị đã ảo hóa và kết nối với MQTT broker để cập nhật dữ liệu được gửi lên từ thiết bị hoặc ứng dụng di động. Các dữ liệu nhận được này sẽ được lựa chọn để tính toán ra được thời gian ngủ mỗi ngày của người dùng. Cuối cùng là dữ liệu sẽ được lưu trữ trên Firebase Realtime Database để dễ dàng truy xuất và xử lý.

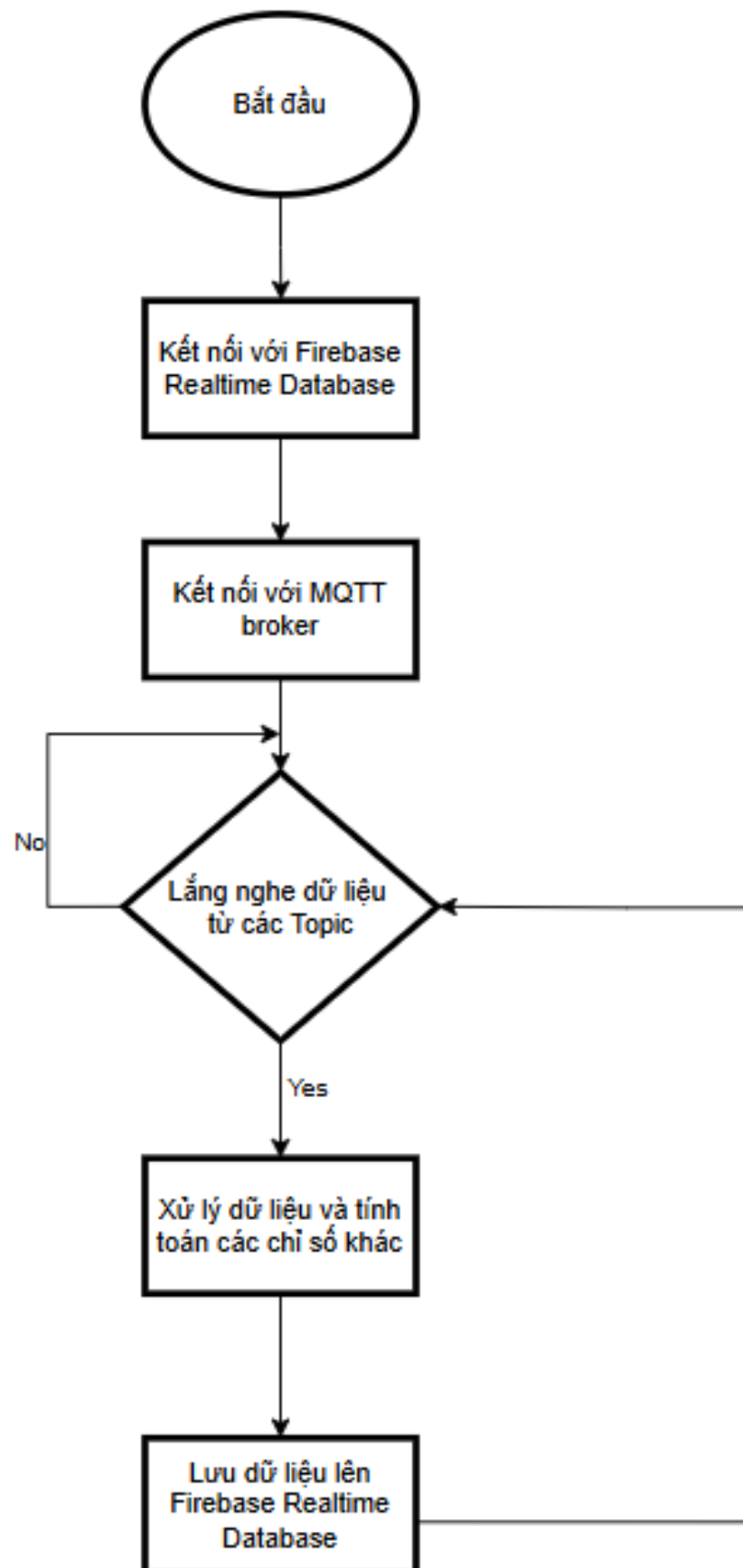
Lưu đồ thuật toán được trình bày trên các hình từ 3.11 đến 3.13.



Hình 3.11. Lưu đồ thuật toán thiết bị.



Hình 3.12. Lưu đồ thuật toán ứng dụng



Hình 3.13. Lưu đồ thuật toán Server

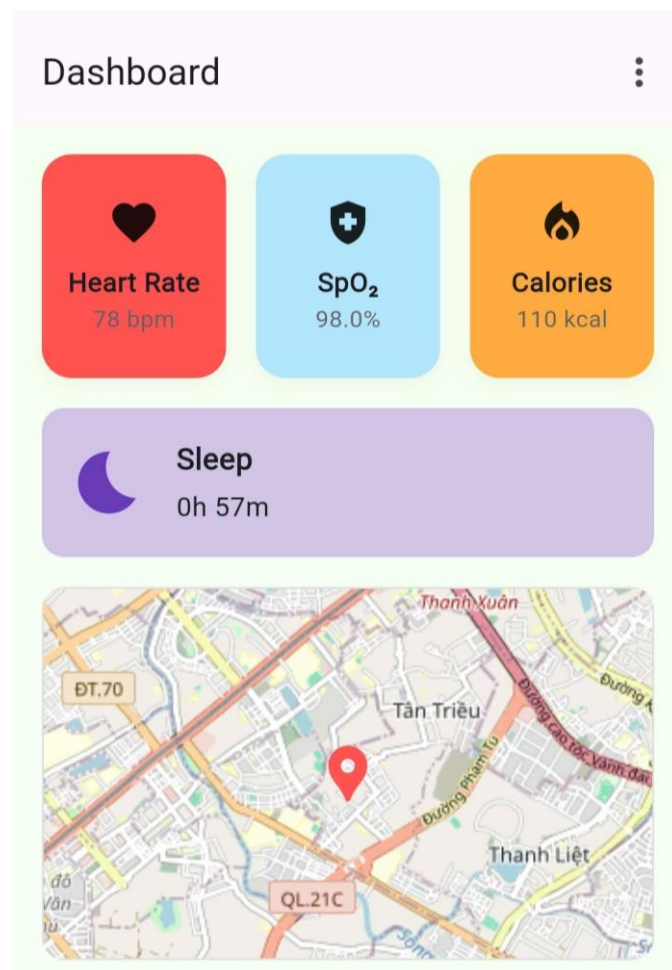


### 3.3. Triển khai thử nghiệm và đánh giá hoạt động của sản phẩm

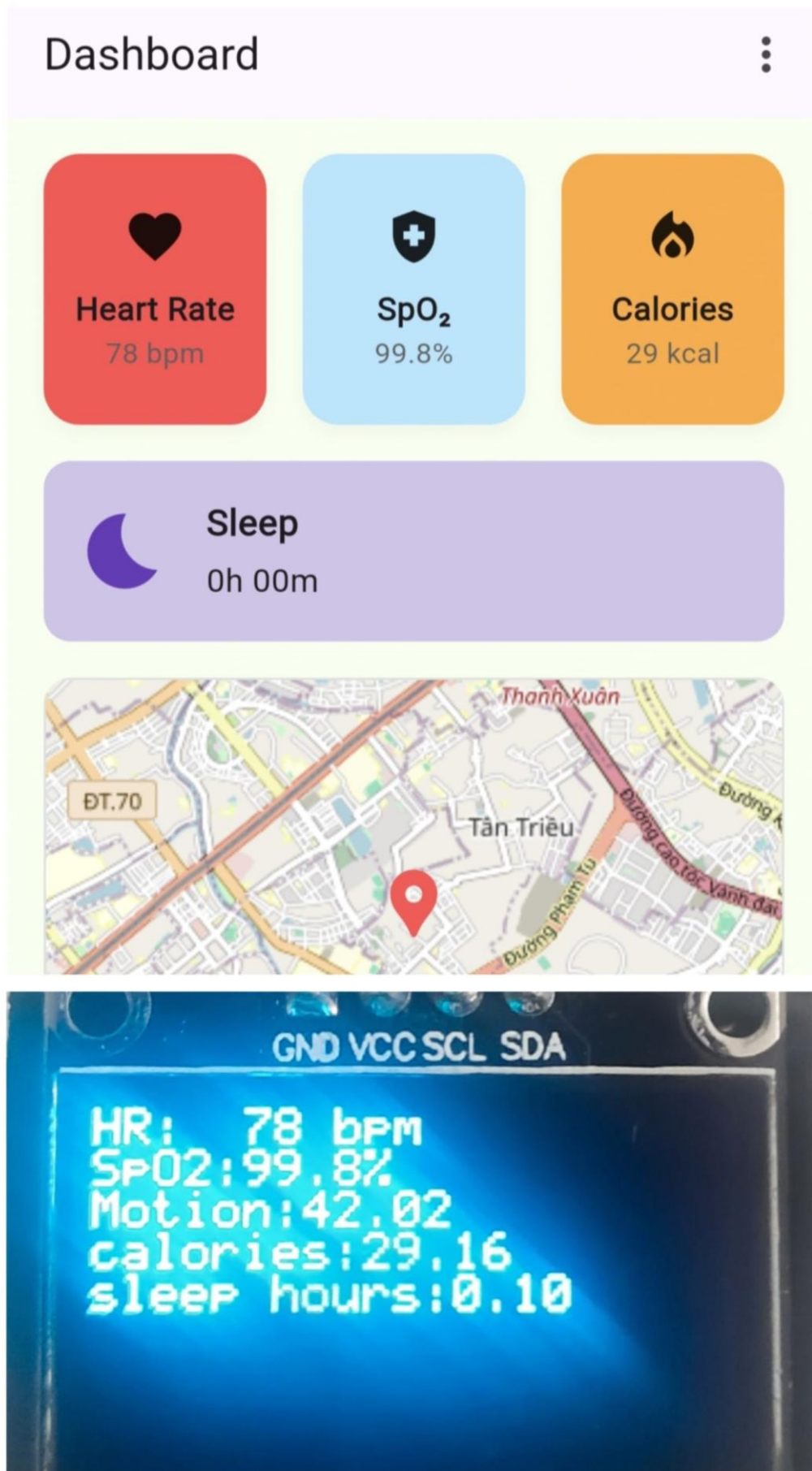
#### 3.3.1. Thử nghiệm hoạt động của sản phẩm



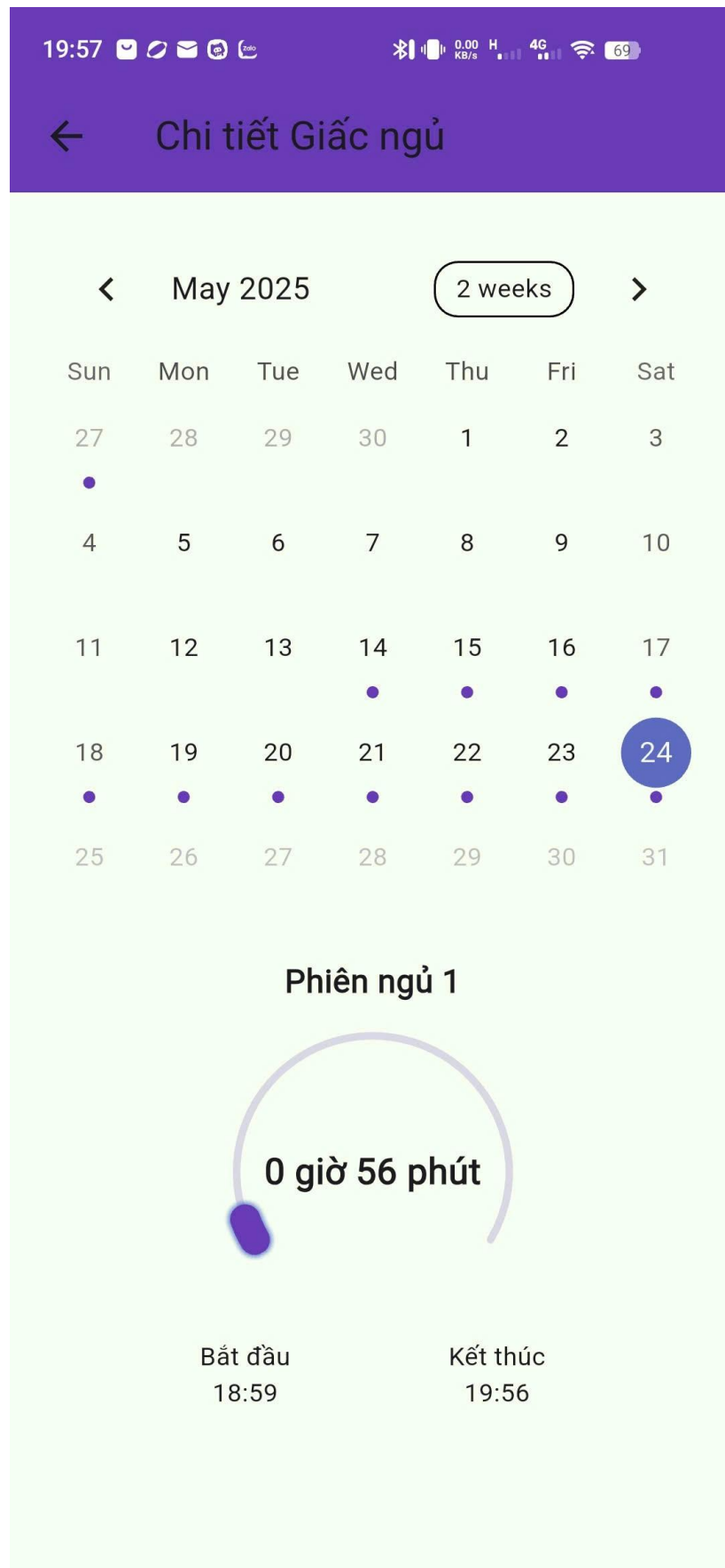
Hình 3.14. Màn hình hiển thị khi thiết bị hoạt động ổn định



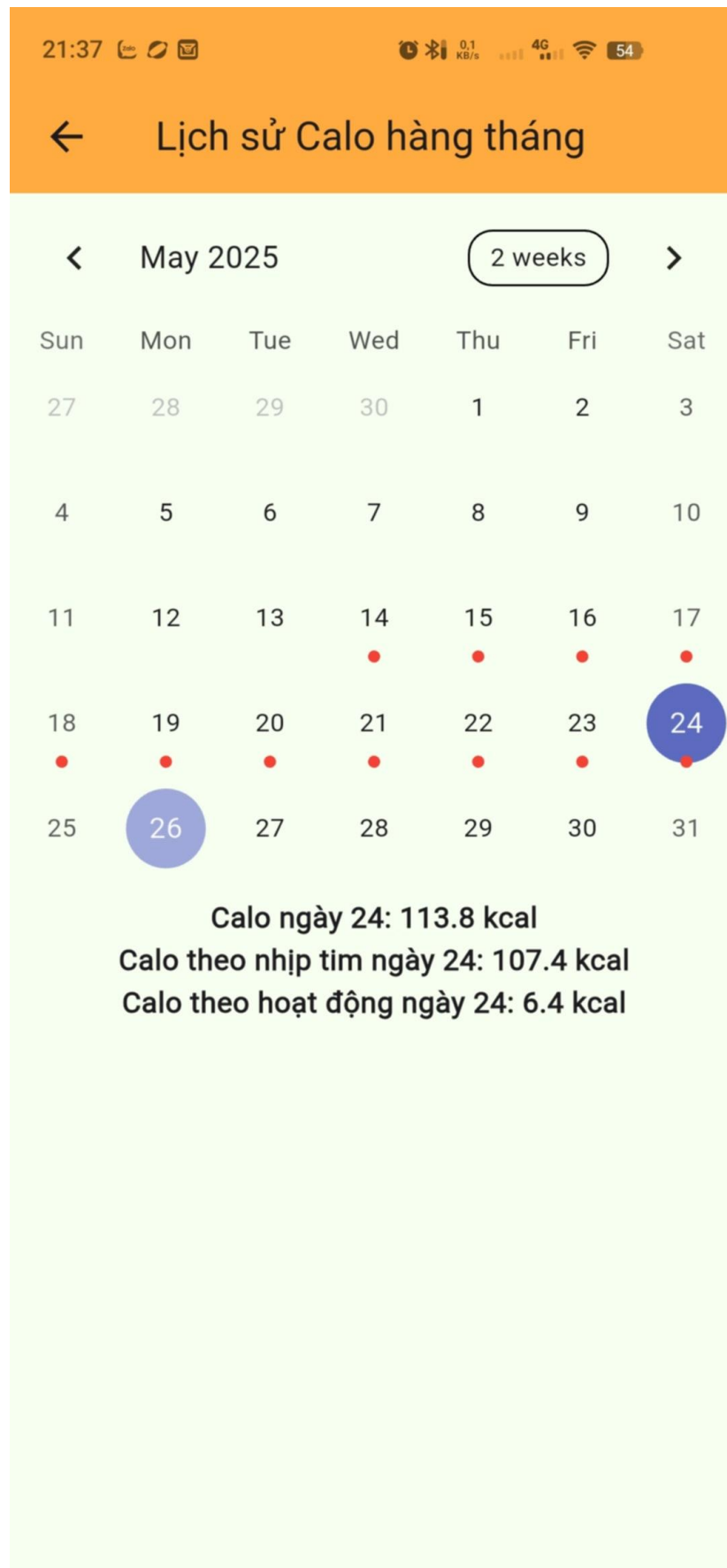
Hình 3.15. Màn hình trên ứng dụng di động khi thiết bị hoạt động ổn định



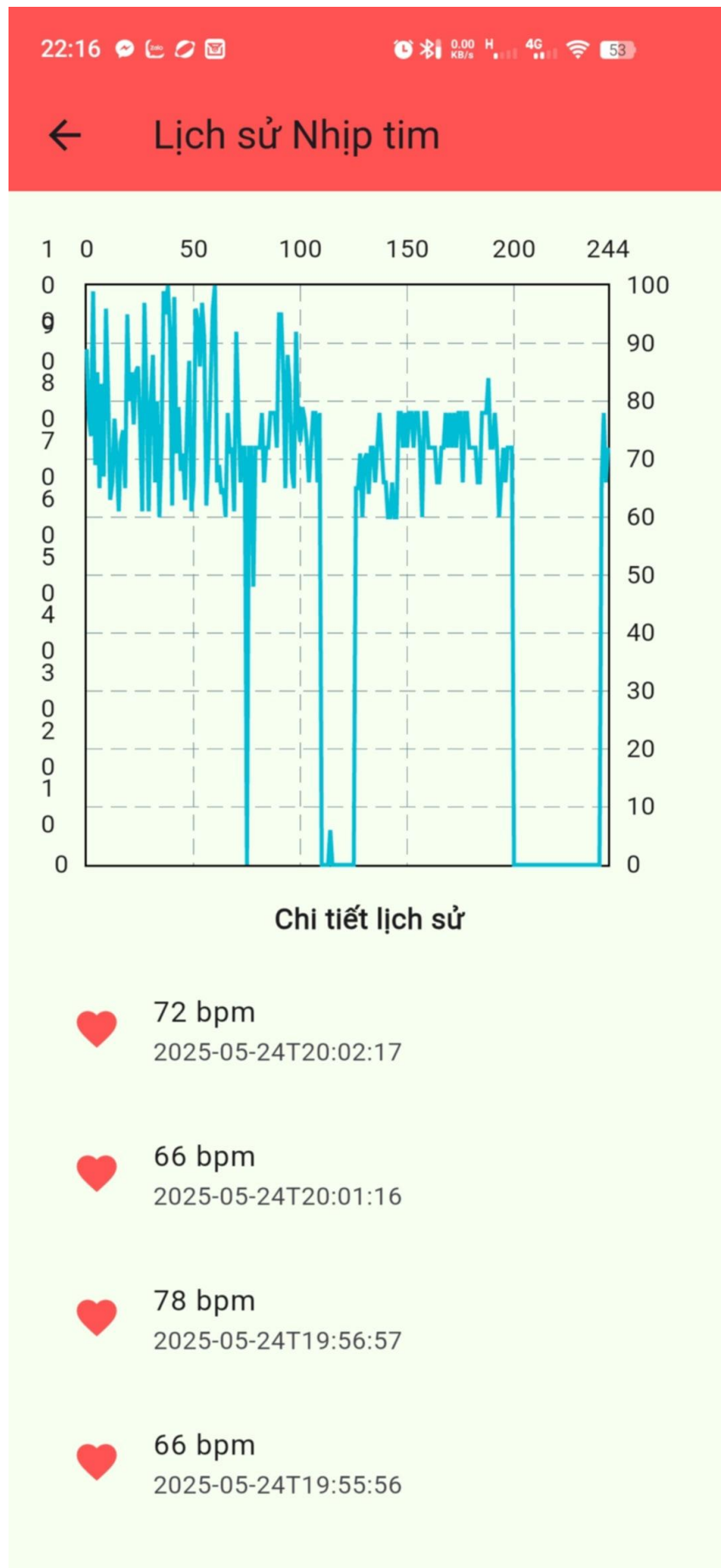
Hình 3.16. So sánh đồng bộ dữ liệu giữa thiết bị và ứng dụng di động



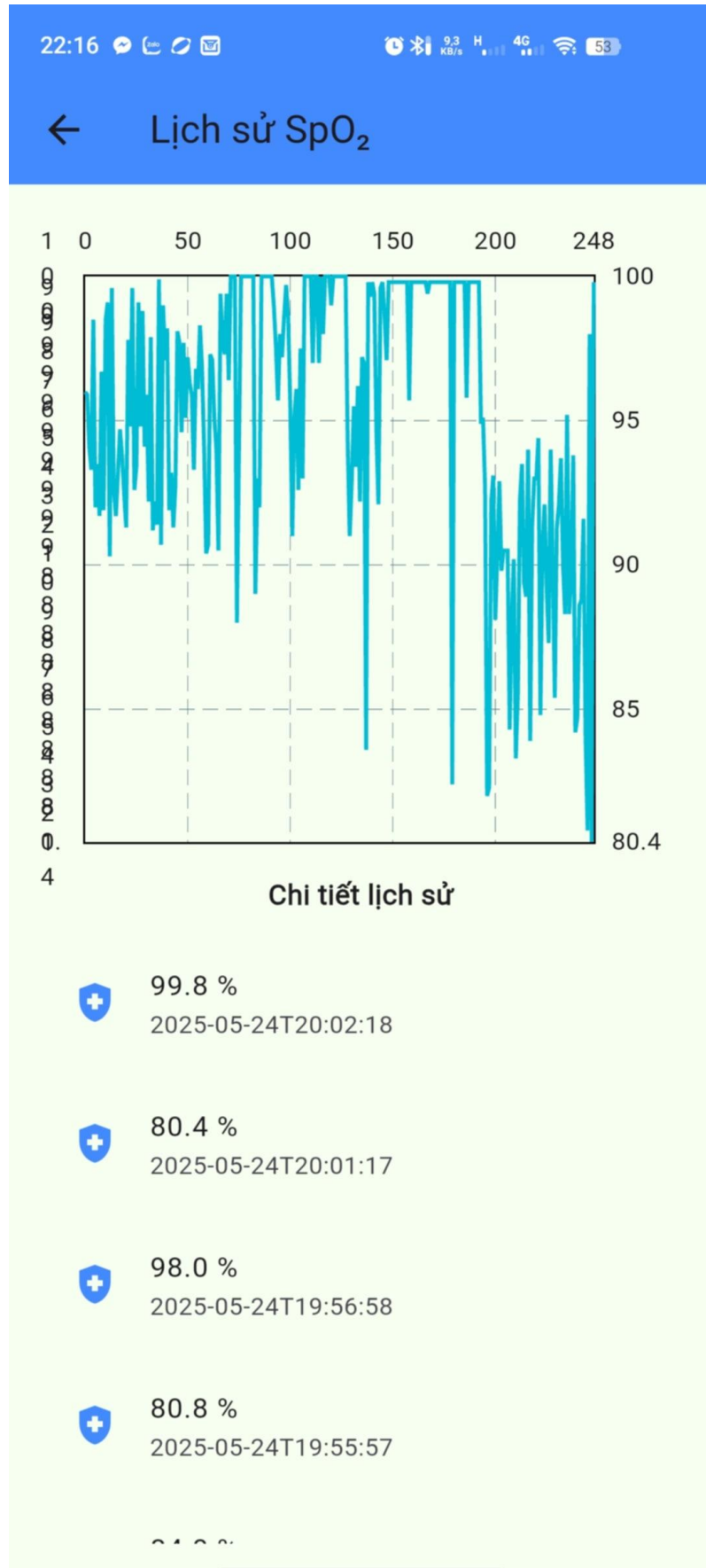
Hình 3.17. Hiện thị thời gian ngủ theo các chỉ số



Hình 3.18. Năng lượng tiêu hao theo nhịp tim và hoạt động trong ngày



Hình 3.19. Lịch sử đo chỉ số nhịp tim



Hình 3.20. Lịch sử đo chỉ số nồng độ oxy trong máu

### 3.3.2. *Đánh giá hoạt động*

- ❖ Thiết bị giám sát thông số sức khỏe hoạt động và thực hiện được các chức năng như thiết kế ban đầu đề ra.
- ❖ Các thông số sức khỏe đo bằng thiết bị do sinh viên thiết kế và chế tạo hiển thị rõ ràng trên ứng dụng.
- ❖ Dữ liệu được cập nhật và hiển thị trên ứng dụng theo thời gian thực với độ trễ thấp, có thể chấp nhận được.

## KẾT LUẬN

### Những nhiệm vụ đã thực hiện được:

- ❖ Tìm hiểu tổng quan về thông số sức khỏe và Internet vạn vật.
- ❖ Tìm hiểu các đặc tính kỹ thuật của linh kiện để phục vụ cho việc thiết kế mạch và lắp ráp sản phẩm.
- ❖ Tìm hiểu ngôn ngữ lập trình C với esp-idf framework, ngôn ngữ lập trình Python và cơ sở dữ liệu Firebase Realtime Database. Tìm hiểu về Dart và khung Flutter để phục vụ quá trình thiết kế ứng dụng Android.
- ❖ Thiết kế, lắp ráp, thử nghiệm và đánh giá hoạt động của thiết bị. Kết quả cho thấy thiết bị hoạt động và thực hiện được các chức năng như thiết kế ban đầu đề ra, các thông số đo được khá chính xác, cập nhật và hiển thị lên ứng dụng kịp thời.
- ❖ Tích lũy kinh nghiệm chế tạo phần cứng và thiết kế phần mềm, làm cơ sở định hướng nghề nghiệp tương lai.

### Hạn chế của sản phẩm:

- ❖ Tính thẩm mỹ của sản phẩm chưa cao, chức năng chưa đa dạng.
- ❖ Các linh kiện được sử dụng chỉ phù hợp với dự án của học sinh, sinh viên, chưa đảm bảo chất lượng nên đôi khi mô hình hoạt động không ổn định.
- ❖ Hoạt động của thiết bị phụ thuộc vào độ ổn định của wifi/BLE.
- ❖ Chưa có những phương án bảo mật cho hệ thống.
- ❖ Không có phương án hoạt động khi không thể gửi dữ liệu lên điện thoại khi mất wifi và mất ble.
- ❖ Chưa có chức năng thông báo khi có trường hợp khẩn cấp.

### Hướng phát triển đề tài:

- ❖ Phát triển thêm chức năng lưu trữ và gửi thông tin khi mất hoàn toàn kết nối với wifi và BLE.
- ❖ Tích hợp chức năng cảnh báo, khuyến nghị phương án cải thiện chất lượng giấc ngủ, nhắc nhở vận động, gửi thông báo trong trường hợp khẩn cấp.
- ❖ Tối ưu thiết kế để nâng cao tính thẩm mỹ và tiện lợi khi sử dụng.



# TÀI LIỆU THAM KHẢO

## Tiếng Việt

1. Nguyễn Văn Dũng, Phát triển ứng dụng IoT với ESP32 và ESP-IDF, Nhà xuất bản Khoa học và Kỹ thuật, 2022.
2. Trần Ngọc Hòa, Lương Minh Tuấn, Lập trình ESP32-S3 với ESP-IDF: Thực hành IoT, Nhà xuất bản Bách khoa Hà Nội, 2023.
3. Phạm Quốc Thanh, Thiết kế mạch in PCB và nguyên lý thiết kế, Nhà xuất bản Giao thông Vận tải, 2021.
4. Lê Minh Chính, Thiết kế mạch in và chế tạo bo mạch PCB, Nhà xuất bản Đại học Quốc gia Hà Nội, 2020.
5. Nguyễn Phương Nam, Ứng dụng KiCad trong thiết kế mạch in, Nhà xuất bản Khoa học và Kỹ thuật, 2021.

## Tiếng Anh

6. Neil Kolban, Kolban's Book on ESP32 (The definitive guide to programming on the ESP32), Leanpub, 2020. Espressif Systems
7. Dogan Ibrahim, The Official ESP32 Book, Wiley / Elektor, 2019. Amazon
8. Dr. Peter Dalmaris, KiCad Like a Pro: Fundamentals and Projects (3rd ed.), CircuitWorks Press, 2022. KiCad
9. Simon Monk, Programming with MicroPython: Embedded Programming with Microcontrollers and ESP32, Maker Media, 2017.
10. Christopher Hallinan, PCB Design for Real-World Applications, Springer, 2021.

## Danh mục các Website tham khảo:

11. "[ESP32-S3-MINI-1 Datasheet](#)," Espressif Systems, truy cập 01/05/2025.
12. "[MAX30102 Pulse Oximeter and Heart-Rate Sensor IC](#)," SparkFun Electronics (datasheet), truy cập 02/05/2025.
13. "[LIS2DH12 3-Axis Accelerometer](#)," STMicroelectronics (datasheet), truy cập 03/05/2025.

14. “[u-blox L80-R GNSS Module Product Description](#),” u-blox, truy cập 04/05/2025.
15. “[MQTT Version 3.1.1 Specification](#),” OASIS Standard, truy cập 05/05/2025.
16. “[Firebase Realtime Database](#),” Firebase Documentation, Google, truy cập 06/05/2025.
17. “[Tiêu chuẩn nhịp tim và nồng độ SpO<sub>2</sub> của người bình thường](#),” Viện Sức khỏe & Đời sống, truy cập 07/05/2025.

## **PHỤ LỤC**

Link dự án: [https://github.com/nguyen-van-phuong-19/Home\\_health\\_care](https://github.com/nguyen-van-phuong-19/Home_health_care)