**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

=====o0o=====

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: MẠNG CẢM BIẾN**

**Đề tài: Hệ thống theo dõi sức khỏe và phát hiện ngã cho người cao tuổi**

**Giảng viên hướng dẫn: Cô Trần Thị Thanh Thủy**

**Nhóm Bài tập lớn:** Nhóm 7

**Họ và tên sinh viên:**

|  |  |
| --- | --- |
| Phạm Trường Giang  Nguyễn Hữu Hoàng Hát | Mã sinh viên: B21DCDT082  Mã sinh viên: B21DCDT086 |

**Lớp:** D21DTVM1 **Khóa:** D21

**Ngành đào tạo**: Kỹ thuật điện - điện tử **Hệ đào tạo**: Đại học chính quy

**Hà Nội, 2025**

# BẢNG PHÂN CHIA CÔNG VIỆC

|  |  |
| --- | --- |
| **Công Việc** | **Thành viên phụ trách** |
| Lập trình ESP32 | Phạm Trường Giang, Nguyễn Hữu Hoàng Hát |
| Thiết kế mô hình hệ thống | Phạm Trường Giang |
| Thiết kế PCB | Phạm Trường Giang |
| Thu thập dữ liệu | Phạm Trường Giang, Nguyễn Hữu Hoàng Hát |
| Huấn luyện AI trên Edge Impulse | Phạm Trường Giang |
| Thiết kế giao diện trên ThingsBoard | Phạm Trường Giang |
| Kiểm thử hệ thống | Phạm Trường Giang, Nguyễn Hữu Hoàng Hát |
| Viết báo cáo | Phạm Trường Giang, Nguyễn Hữu Hoàng Hát |
| Thiết kế Slide, PowerPoint | Phạm Trường Giang |

# LỜI CAM ĐOAN

Chúng em xin cam đoan đề tài "Hệ thống theo dõi sức khỏe và phát hiện ngã cho người cao tuổi" là kết quả của quá trình tìm hiểu và nghiên cứu nghiêm túc do nhóm thực hiện. Toàn bộ nội dung trong đề tài được xây dựng dựa trên các tài liệu tham khảo đáng tin cậy, giáo trình đã học, cùng với sự tổng hợp từ những kiến thức tích lũy trong suốt quá trình học tập và nghiên cứu. Các tài liệu tham khảo sử dụng trong quá trình thực hiện đề tài đều được trích dẫn đầy đủ và chính xác theo quy định. Chúng em hoàn toàn chịu trách nhiệm về tính trung thực, tính chính xác và nội dung của đề tài này.

Hà Nội, ngày 20 tháng 6 năm 2025

Sinh viên thực hiện

**Phạm Trường Giang**

**Nguyễn Hữu Hoàng Hát**

# LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời tri ân sâu sắc đến quý thầy cô khoa Điện tử, những người đã tận tâm truyền đạt kiến thức và hướng dẫn chúng em trong suốt quá trình học tập. Nhờ sự chỉ dạy tận tình của thầy cô, chúng em đã có nền tảng vững chắc để hoàn thành môn Mạng Cảm Biến. Đặc biệt, chúng em xin bày tỏ lòng biết ơn chân thành đến cô Trần Thị Thanh Thủy, người đã trực tiếp giảng dạy, hướng dẫn và tạo điều kiện thuận lợi để chúng em hoàn thành đề tài một cách tốt nhất.

Chúng em cũng xin cảm ơn các anh chị, bạn bè cùng khoa đã luôn hỗ trợ, chia sẻ kinh nghiệm, giúp nhóm vượt qua nhiều khó khăn trong quá trình thực hiện đề tài.

Dù đã cố gắng nhưng chắc chắn đề tài vẫn còn thiếu sót, nhóm em mong nhận được sự góp ý từ thầy cô để có thể hoàn thiện hơn trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn!

Sinh viên thực hiện

**Phạm Trường Giang**

**Nguyễn Hữu Hoàng Hát**

MỤC LỤC

[BẢNG PHÂN CHIA CÔNG VIỆC i](#_Toc201855980)

[LỜI CAM ĐOAN ii](#_Toc201855981)

[LỜI CẢM ƠN iii](#_Toc201855982)

[DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT vi](#_Toc201855983)

[DANH MỤC HÌNH VẼ vii](#_Toc201855984)

[DANH MỤC BẢNG BIỂU ix](#_Toc201855985)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI VÀ AN TOÀN LAO ĐỘNG 1](#_Toc201855986)

[1.1. Vấn đề về già hóa dân số tại Việt Nam và trên thế giới 1](#_Toc201855987)

[1.2. Giới thiệu về Internet of Things trong chăm sóc sức khỏe 3](#_Toc201855988)

[1.3. Mục tiêu đề tài 5](#_Toc201855989)

[1.4. Phạm vi thực hiện 5](#_Toc201855990)

[1.5. Ý nghĩa khoa học và thực tiễn 6](#_Toc201855991)

[1.6. Tổng quan các hệ thống tương tự 6](#_Toc201855992)

[1.7. Luật An toàn, vệ sinh lao động số 84/2015/QH13 7](#_Toc201855993)

[1.8. Phương pháp thực hiện 8](#_Toc201855994)

[1.9. Cấu trúc báo cáo 10](#_Toc201855995)

[1.10. Kết luận chương 1 10](#_Toc201855996)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ XÂY DỰNG NÚT CẢM BIẾN 11](#_Toc201855997)

[2.1. Cơ sở lý thuyết về phần cứng 11](#_Toc201855998)

[2.1.1. Vi xử lý ESP32 11](#_Toc201855999)

[2.1.2. Cảm biến nhiệt độ DS18B20 14](#_Toc201856000)

[2.1.3. Cảm biến chất lượng không khí MQ-135 15](#_Toc201856001)

[2.1.4. Cảm biến nhịp tim & SPO2 trong MAX30100 16](#_Toc201856002)

[2.1.5. Cảm biến gia tốc MPU6050 17](#_Toc201856003)

[2.1.6. Module khuếch đại âm thanh MAX98357 19](#_Toc201856004)

[2.2. Cơ sở lý thuyết phần mềm và giao thức 20](#_Toc201856005)

[2.2.1. Các giao thức kết nối module ngoại vi 20](#_Toc201856006)

[2.2.3. PlatformIO - Công cụ phát triển phần mềm 23](#_Toc201856007)

[2.2.4. ThingsBoard - Nền tảng quản lý thiết bị IoT 24](#_Toc201856008)

[2.2.5. Edge Impulse - Nền tảng phát triển mô hình AI dành cho thiết bị nhúng 25](#_Toc201856009)

[2.3. Xây dựng nút cảm biến 25](#_Toc201856010)

[2.3.1. Nút cảm biến 1 - phát hiện té ngã và phát âm thanh cảnh báo 25](#_Toc201856011)

[2.3.2. Nút cảm biến 2 - theo dõi sức khỏe 26](#_Toc201856012)

[2.4. Kết luận chương 2 27](#_Toc201856013)

[CHƯƠNG 3: MÔ HÌNH, KỊCH BẢN VÀ ĐÁNH GIÁ HỆ THỐNG 28](#_Toc201856014)

[3.1. Mô hình hệ thống 28](#_Toc201856015)

[3.1.1. Mô hình nút cảm biến 1 28](#_Toc201856016)

[3.1.2. Mô hình nút cảm biến 2 36](#_Toc201856017)

[3.1.3. Thiết kế PCB trên Easy EDA 39](#_Toc201856018)

[3.1.4. Mô hình tổng quát của hệ thống 42](#_Toc201856019)

[3.2. Kịch bản thử nghiệm hệ thống 43](#_Toc201856020)

[3.2.1. Hiệu suất sử dụng mô hình 43](#_Toc201856021)

[3.2.2. Các kịch bản kiểm thử 43](#_Toc201856022)

[3.2.3. Kết luận kiểm thử hệ thống 47](#_Toc201856023)

[3.3. Kết luận và hướng phát triển trong tương lai 47](#_Toc201856024)

[TÀI LIỆU THAM KHẢO 49](#_Toc201856025)

# DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Viết đầy đủ** | **Nghĩa** |
| IoT | Internet of Things | Internet vạn vật – hệ thống kết nối các thiết bị qua Internet |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| SoC | System on Chip | Hệ thống tích hợp trên một vi mạch |
| WiFi | Wireless Fidelity | Mạng không dây nội bộ |
| BLE | Bluetooth Low Energy | Chuẩn Bluetooth năng lượng thấp |
| DAC | Digital to Analog Converter | Bộ chuyển đổi tín hiệu số sang tương tự |
| ADC | Analog to Digital Converter | Bộ chuyển đổi tín hiệu tương tự sang số |
| RAM | Random Access Memory | Bộ nhớ truy cập ngẫu nhiên |
| ROM | Read-Only Memory | Bộ nhớ chỉ đọc |
| EEPROM | Electrically Erasable Programmable ROM | Bộ nhớ có thể xóa/lập trình bằng điện |
| OTA | Over The Air | Cập nhật chương trình qua mạng không dây |
| SPIFFS | SPI Flash File System | Hệ thống file lưu trên bộ nhớ flash SPI |
| NVS | Non-Volatile Storage | Bộ nhớ không mất dữ liệu khi mất nguồn |
| GPIO | General Purpose Input Output | Chân vào/ra tổng quát |
| I2C | Inter-Integrated Circuit | Giao thức truyền dữ liệu 2 dây giữa các IC |
| I2S | Integrated Interchip Sound | Giao tiếp âm thanh kỹ thuật số giữa các chip |

# DANH MỤC HÌNH VẼ

[Hình 1.1. Già hóa dân số ở Việt Nam. 1](#_Toc201856026)

[Hình 1.2. Thống kê về tình trạng té ngã ở người cao tuổi tại Mỹ. 2](#_Toc201856027)

[Hình 1.3. IoT hỗ trợ sức khỏe con người. 3](#_Toc201856028)

[Hình 1.4. Sơ đồ các nút cảm biến và giao tiếp giữa nút và cloud. 5](#_Toc201856029)

[Hình 1.5. Một đồng hồ giám sát sức khỏe. 7](#_Toc201856030)

[Hình 1.6. Các thiết bị ngoại vi sử dụng trong đề tài. 9](#_Toc201856031)

[Hình 2.1. ESP32 Devkit V1 và cấu trúc chân. 11](#_Toc201856032)

[Hình 2.2. Hình module DS18B20 thực tế. 14](#_Toc201856033)

[Hình 2.3. Hình module MQ-135 thực tế. 15](#_Toc201856034)

[Hình 2.4. Hình module MAX30100 thực tế. 17](#_Toc201856035)

[Hình 2.5. Hình module MPU6050 thực tế. 18](#_Toc201856036)

[Hình 2.6. Hình module MAX98357 thực tế. 19](#_Toc201856037)

[Hình 2.7. Giao diện sử dụng trong PlatformIO. 23](#_Toc201856038)

[Hình 2.8. Giao diện sử dụng của ThingsBoard. 24](#_Toc201856039)

[Hình 2.9. Sơ đồ kết nối trong nút 1. 26](#_Toc201856040)

[Hình 2.10. Sơ đồ kết nối trong nút 2. 26](#_Toc201856041)

[Hình 3.1. Sơ đồ kết nối chân của hệ thống. 28](#_Toc201856042)

[Hình 3.2. Sơ đồ nguyên lý hệ thống. 28](#_Toc201856043)

[Hình 3.3. Thời gian thu thập dữ liệu, các nhãn và tỷ lệ giữa huấn luyện - kiểm thử. 29](#_Toc201856044)

[Hình 3.4. Phần cấu hình trong Create impulse. 30](#_Toc201856045)

[Hình 3.5. Kết quả sau khi Generate features trong phần Spectral features. 31](#_Toc201856046)

[Hình 3.6. Kết quả Sau khi Training với tỷ lệ chính xác của các Nhãn cao. 31](#_Toc201856047)

[Hình 3.7. Chọn thiết bị và định dạng cho mô hình AI. 32](#_Toc201856048)

[Hình 3.8. DashBoard của nút cảm biến 1. 34](#_Toc201856049)

[Hình 3.9. Rule Chain gửi cảnh báo phát hiện ngã. 35](#_Toc201856050)

[Hình 3.10. Kết nối Rule chain vừa tạo với Save Timeseries. 35](#_Toc201856051)

[Hình 3.11. Mô hình kết nối chân trong nút cảm biến 2. 36](#_Toc201856052)

[Hình 3.12. DashBoard của cả hệ thống (gồm cả nút 1 và 2). 37](#_Toc201856053)

[Hình 3.13. Rule Chain cảnh báo chất lượng không khí kém. 37](#_Toc201856054)

[Hình 3.14. Rule Chain cảnh báo nhiệt độ cao. 38](#_Toc201856055)

[Hình 3.15. Rule Chain cảnh báo nhiệt độ thấp. 38](#_Toc201856056)

[Hình 3.16. Kết nối tất Rule chain với Save Timeseries. 39](#_Toc201856057)

[Hình 3.17. Giao diện sử dụng của Easy EDA. 40](#_Toc201856058)

[Hình 3.18. Schematic của mô hình hệ thống. 40](#_Toc201856059)

[Hình 3.19. Kết quả sau khi sắp xếp linh kiện, đi dây và phủ đồng PCB. 41](#_Toc201856060)

[Hình 3.20. Ảnh thực tế của mô hình hệ thống. 42](#_Toc201856061)

# DANH MỤC BẢNG BIỂU

[Bảng 2.1. Các chân ảnh hưởng đến khởi động ESP32. 12](#_Toc201856062)

[Bảng 2.2. Các chân hỗ trợ DAC. 12](#_Toc201856063)

[Bảng 2.3. I2C - Giao tiếp nối tiếp đồng bộ. 12](#_Toc201856064)

[Bảng 2.4. Bảng chức năng chân trên MPU6050. 19](#_Toc201856065)

[Bảng 2.5. Bảng chức năng chân trên MAX98357. 20](#_Toc201856066)

[Bảng 2.6. Tính năng chính của PlatformIO. 23](#_Toc201856067)

[Bảng 3.1. Kiểm tra độ chính xác của mô hình AI. 43](#_Toc201856068)

[Bảng 3.2. Kiểm thử độ chính xác của cảm biến nhiệt độ. 44](#_Toc201856069)

[Bảng 3.3. Kiểm thử độ chính xác của cảm biến chất lượng không khí. 45](#_Toc201856070)

[Bảng 3.4. Kiểm thử độ chính xác của cảm biến đo nhịp tim và SPO2. 45](#_Toc201856071)

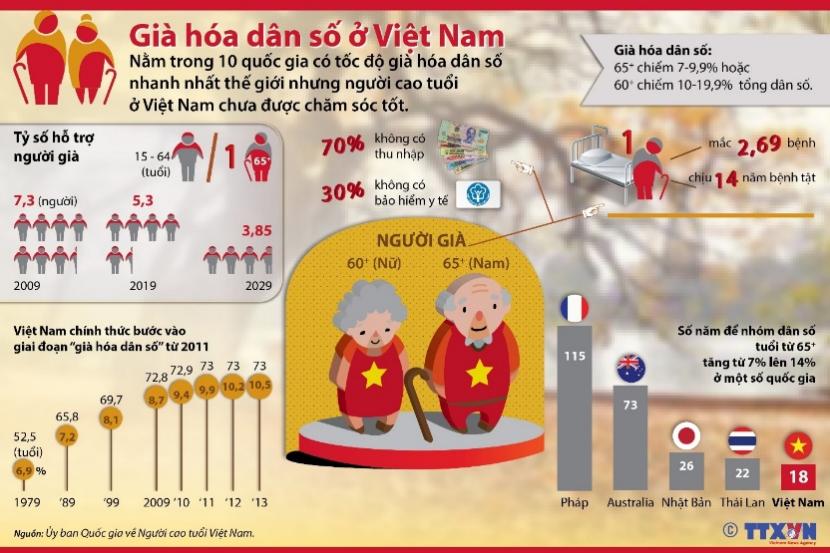
[Bảng 3.5. Kiểm thử độ chính xác của module khuếch đại âm thanh và mô hình hệ thống khi phát hiện tình huống ngã. 46](#_Toc201856072)

[Bảng 3.6. Kiểm thử độ chính xác khi nhận các cảnh báo trên ThingsBoard. 46](#_Toc201856073)

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI VÀ AN TOÀN LAO ĐỘNG

## 1.1. Vấn đề về già hóa dân số tại Việt Nam và trên thế giới

Trong thế kỷ 21, sự già hóa dân số đã trở thành một xu thế tất yếu trên toàn cầu. Theo dự báo của Tổ chức Y tế Thế giới (WHO), đến năm 2050, số người từ 60 tuổi trở lên sẽ chiếm khoảng 22% dân số thế giới – tương đương khoảng 2 tỷ người. Điều này đặt ra yêu cầu cấp thiết về việc phát triển các giải pháp công nghệ hỗ trợ chăm sóc sức khỏe, giúp người cao tuổi có thể sống độc lập, an toàn và có chất lượng cuộc sống tốt hơn.



Hình 1.1. Già hóa dân số ở Việt Nam.

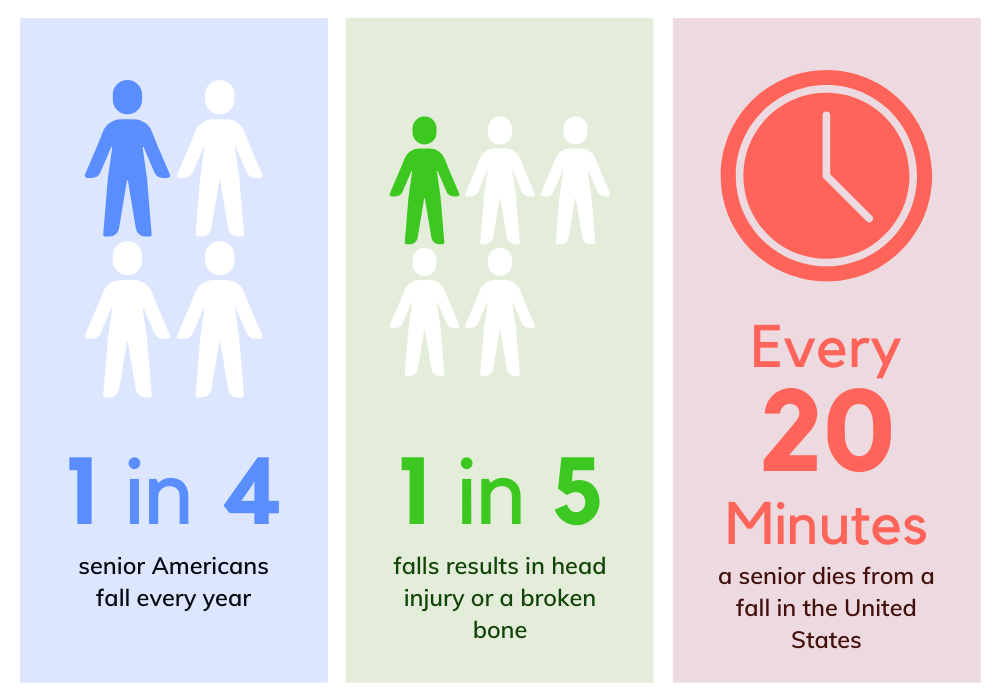
Tại Việt Nam, tiến trình già hóa dân số đang diễn ra với tốc độ nhanh chóng. Theo số liệu từ Tổng cục Thống kê:

* Tỷ lệ người từ 60 tuổi trở lên đã tăng từ 7,3% (2009) lên 13% (2019).
* Dự báo sẽ đạt 18,4% vào năm 2029 và khoảng 25% vào năm 2050 – tương đương 30 triệu người.

Hệ quả của quá trình này rất rõ ràng:

* 70% người cao tuổi không có nguồn thu nhập ổn định.
* 30% không có bảo hiểm y tế, gây khó khăn trong tiếp cận dịch vụ y tế.
* Trung bình, mỗi người mắc 2,69 bệnh mãn tính và chịu 14 ngày bệnh tật/năm.
* Tỷ lệ hỗ trợ giữa người trẻ và người già giảm nhanh, từ 7,3 (2009) còn 3,85 (2029).

Một trong những vấn đề sức khỏe nghiêm trọng ở người cao tuổi là nguy cơ bị té ngã và các biến chứng liên quan. Té ngã không chỉ gây ra chấn thương nghiêm trọng mà còn ảnh hưởng đến tâm lý, làm suy giảm chất lượng cuộc sống và khả năng tự chăm sóc bản thân. Ngoài ra, các chỉ số sức khỏe như nhịp tim, nhiệt độ cơ thể, chất lượng không khí xung quanh cũng cần được theo dõi liên tục để kịp thời phát hiện các dấu hiệu bất thường.



Hình 1.2. Thống kê về tình trạng té ngã ở người cao tuổi tại Mỹ.

Hình 1.2 nêu thực trạng té ngã là một trong những nguyên nhân hàng đầu dẫn đến thương tích nặng hoặc tử vong ở người già tại Mỹ:

* 1 trong 4 người cao tuổi té ít nhất 1 lần/năm.
* 20% các trường hợp gây chấn thương nghiêm trọng (gãy xương, chấn thương sọ não).
* Cứ mỗi 20 phút lại có một người tử vong vì té ngã.

Nhiều người cao tuổi sống một mình không được giám sát, dẫn đến chậm trễ trong xử lý y tế sau khi té ngã. Do đó, cần có một hệ thống giám sát thông minh, có khả năng phát hiện té ngã và theo dõi sức khỏe kịp thời.

Trong bối cảnh công nghệ ngày càng phát triển, đặc biệt là sự bùng nổ của Internet of Things (IoT), các hệ thống cảm biến thông minh, vi điều khiển tích hợp mạng không dây đã mở ra khả năng xây dựng các giải pháp y tế cá nhân hóa, hiệu quả và chi phí thấp. Đây chính là nền tảng để nghiên cứu và phát triển hệ thống theo dõi sức khỏe và phát hiện ngã dành cho người cao tuổi.

## 1.2. Giới thiệu về Internet of Things trong chăm sóc sức khỏe

Internet of Things – Internet vạn vật – là một xu hướng công nghệ hiện đại cho phép các thiết bị vật lý (things) có khả năng kết nối Internet, thu thập, chia sẻ và phân tích dữ liệu theo thời gian thực. Trong lĩnh vực chăm sóc sức khỏe, IoT đã mở ra hướng tiếp cận hoàn toàn mới, cho phép xây dựng các hệ thống giám sát y tế từ xa, phản ứng nhanh, tiết kiệm chi phí và cá nhân hóa cho từng bệnh nhân.



Hình 1.3. IoT hỗ trợ sức khỏe con người.

Các thiết bị IoT trong y tế có thể bao gồm: cảm biến sinh học (nhịp tim, SpO₂, nhiệt độ), cảm biến môi trường (chất lượng không khí, độ ẩm), thiết bị đeo thông minh, vi điều khiển và nền tảng điện toán đám mây để xử lý, lưu trữ và hiển thị dữ liệu.

* Hệ thống IoT trong lĩnh vực chăm sóc sức khỏe thường bao gồm 4 lớp chính:
  + Lớp cảm nhận (Perception Layer): Bao gồm các cảm biến gắn trực tiếp lên cơ thể hoặc môi trường sống để đo các chỉ số như nhịp tim, SpO₂, nhiệt độ, chuyển động, âm thanh hoặc khí độc.
  + Lớp mạng (Network Layer): Truyền dữ liệu từ cảm biến đến trung tâm xử lý qua WiFi, Bluetooth, Zigbee hoặc các giao thức truyền thông như MQTT, HTTP.
  + Lớp xử lý (Processing Layer): Dữ liệu được xử lý sơ cấp trên vi điều khiển (như ESP32), hoặc nâng cao hơn tại máy chủ trung tâm, có thể tích hợp trí tuệ nhân tạo (AI) để phát hiện bất thường.
  + Lớp ứng dụng (Application Layer): Hiển thị dữ liệu cho người dùng cuối qua nền tảng đám mây như ThingsBoard, web dashboard hoặc ứng dụng di động.
* Ứng dụng cụ thể cho người cao tuổi:
  + IoT mang lại nhiều ứng dụng thiết thực trong giám sát và chăm sóc sức khỏe người cao tuổi:
  + Theo dõi liên tục thông số sức khỏe: Người cao tuổi thường có các bệnh nền mãn tính. Hệ thống IoT giúp theo dõi nhịp tim, oxy trong máu, thân nhiệt, giúp phát hiện sớm dấu hiệu bất thường mà không cần đến bệnh viện.
  + Phát hiện té ngã và sự cố đột ngột: Với cảm biến gia tốc (MPU6050) và âm thanh (MAX98357), hệ thống có thể nhận biết các chuyển động bất thường và phát hiện va chạm, từ đó gửi cảnh báo tức thì cho người thân hoặc cơ sở y tế.
  + Giám sát chất lượng môi trường sống: Cảm biến khí gas, bụi hoặc CO₂ (ví dụ: MQ135) giúp phát hiện ô nhiễm không khí trong nhà – yếu tố ảnh hưởng lớn đến người cao tuổi mắc bệnh hô hấp.
  + Hỗ trợ sống độc lập: Hệ thống cho phép người cao tuổi sống một mình nhưng vẫn được theo dõi liên tục, giúp giảm gánh nặng chăm sóc cho gia đình mà vẫn đảm bảo an toàn.
  + Cảnh báo khẩn cấp tự động: Khi các thông số vượt ngưỡng nguy hiểm, hệ thống có thể tự động gửi tin nhắn, cảnh báo trên dashboard hoặc gọi điện khẩn cấp nếu cần.

IoT trong chăm sóc sức khỏe không chỉ dừng lại ở mức thu thập và hiển thị dữ liệu. Khi kết hợp với trí tuệ nhân tạo (AI), học máy (machine learning) hoặc phân tích dữ liệu lớn (big data analytics), hệ thống có thể học thói quen sinh hoạt, phát hiện xu hướng bất thường, hoặc dự đoán nguy cơ bệnh tật. Điều này có tiềm năng tạo ra các hệ thống chăm sóc thông minh, cá nhân hóa và chủ động – một hướng đi quan trọng trong y tế tương lai.

Với những đặc điểm trên, IoT không chỉ là một xu hướng công nghệ mà còn là giải pháp thực tiễn, phù hợp để áp dụng trong các mô hình chăm sóc sức khỏe người cao tuổi, đặc biệt ở các quốc gia đang phát triển như Việt Nam – nơi đang đối mặt với tốc độ già hóa dân số nhanh và nguồn lực y tế còn hạn chế.

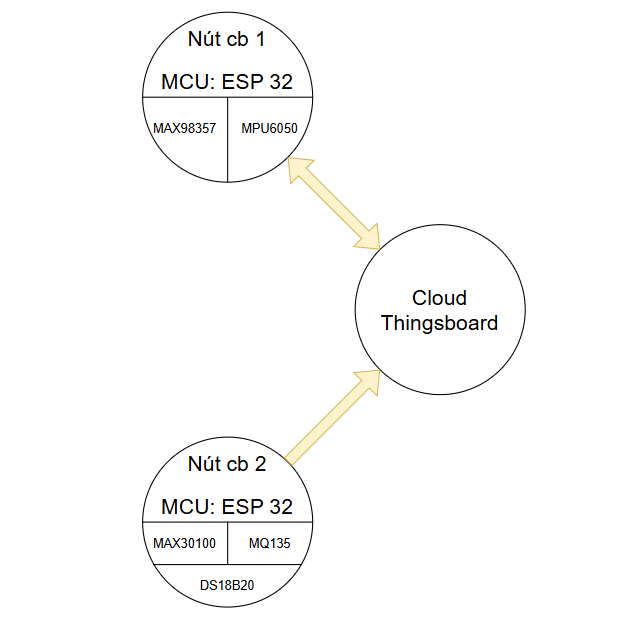
## 1.3. Mục tiêu đề tài

Đề tài hướng đến mục tiêu chính là thiết kế và xây dựng một hệ thống IoT tích hợp nhiều loại cảm biến nhằm:

* + Thiết kế và triển khai hệ thống giám sát sức khỏe tích hợp cảm biến đo nhịp tim, nhiệt độ cơ thể, chất lượng không khí và chuyển động.
  + Xây dựng hệ thống phát hiện ngã dựa trên dữ liệu từ cảm biến chuyển động (MPU6050) kết hợp với module khuếch đại âm thanh (MAX98357).
  + Xây dựng hai thiết bị cảm biến hoạt động độc lập, thu thập dữ liệu và truyền dữ liệu không dây về máy chủ xử lý qua giao thức MQTT.
  + Thiết kế giao diện hiển thị thông tin sức khỏe thời gian thực thông qua nền tảng ThingsBoard.
  + Đưa ra kịch bản kiểm thử mô hình trong môi trường giả lập, đánh giá hiệu quả hoạt động hệ thống.

## 1.4. Phạm vi thực hiện

* Thiết kế hai nút cảm biến với chức năng riêng biệt:
  + Nút cảm biến 1: Bao gồm ESP32, MPU6050 và MAX98357. Dùng để phát hiện chuyển động bất thường như té ngã và phân tích âm thanh đi kèm (tiếng va chạm).
  + Nút cảm biến 2: Bao gồm ESP32, MAX30100, DS18B20 và MQ135. Dùng để đo nhịp tim, SpO₂, nhiệt độ cơ thể và chất lượng không khí.



Hình 1.4. Sơ đồ các nút cảm biến và giao tiếp giữa nút và cloud.

* Xây dựng chương trình điều khiển trên ESP32 để thu thập và xử lý sơ cấp dữ liệu cảm biến.
* Giao tiếp dữ liệu qua WiFi bằng giao thức MQTT đến nền tảng ThingsBoard.
* Thiết kế giao diện giám sát trực quan, dễ sử dụng, hiển thị thông tin theo thời gian thực.
* Hệ thống không áp dụng cho mục đích chẩn đoán y tế chính thức, chỉ dùng để hỗ trợ theo dõi và cảnh báo.
* Không triển khai thực nghiệm trên người dùng thật, thay vào đó là kiểm thử trên mô hình mô phỏng (test bench) và kiểm tra tính năng đơn lẻ từng phần.

## 1.5. Ý nghĩa khoa học và thực tiễn

Đề tài không chỉ có giá trị trong lĩnh vực kỹ thuật mà còn đóng vai trò hỗ trợ trong chăm sóc sức khỏe cộng đồng. Hệ thống giúp:

* Về mặt khoa học:
  + Áp dụng các lý thuyết về xử lý tín hiệu, vi điều khiển, truyền thông không dây vào thực tiễn.
  + Thử nghiệm thuật toán phát hiện ngã với dữ liệu cảm biến thực tế.
  + Tích hợp đa loại cảm biến trên một nền tảng thống nhất.
* Về mặt thực tiễn:
  + Cung cấp giải pháp chi phí thấp cho chăm sóc sức khỏe từ xa.
  + Phù hợp cho người cao tuổi sống độc lập, đặc biệt ở vùng nông thôn, nơi thiếu cơ sở y tế.
  + Giảm tải cho các cơ sở y tế, hỗ trợ bác sĩ và người thân trong việc theo dõi tình trạng bệnh nhân.
  + Hệ thống có thể được phát triển thêm để tích hợp AI, học máy, hoặc mở rộng mạng lưới cảm biến để áp dụng trong viện dưỡng lão, bệnh viện hoặc nhà riêng.

## 1.6. Tổng quan các hệ thống tương tự

Hiện nay có nhiều thiết bị đeo tay thông minh như Apple Watch, Mi Band, Fitbit,... có thể đo nhịp tim, đếm bước chân và một số còn có chức năng phát hiện ngã. Tuy nhiên, các thiết bị này thường có chi phí cao và yêu cầu người dùng phải sạc thường xuyên. Ngoài ra, khả năng tích hợp nhiều loại cảm biến cùng lúc như đo khí độc, đo nhiệt độ môi trường chưa thật sự phổ biến.



Hình 1.5. Một đồng hồ giám sát sức khỏe.

Một số hệ thống giám sát dùng camera (như hệ thống của Intel RealSense hoặc các giải pháp AI sử dụng camera hồng ngoại) có thể phát hiện ngã với độ chính xác cao, tuy nhiên lại đối mặt với vấn đề riêng tư, yêu cầu hạ tầng mạng ổn định và chi phí cao.

Đề tài của nhóm đi theo hướng sử dụng vi điều khiển ESP32 kết hợp với các cảm biến phổ thông để giảm thiểu chi phí, đơn giản hóa vận hành, đồng thời vẫn đảm bảo các chức năng cơ bản cần thiết trong theo dõi sức khỏe và phát hiện ngã.

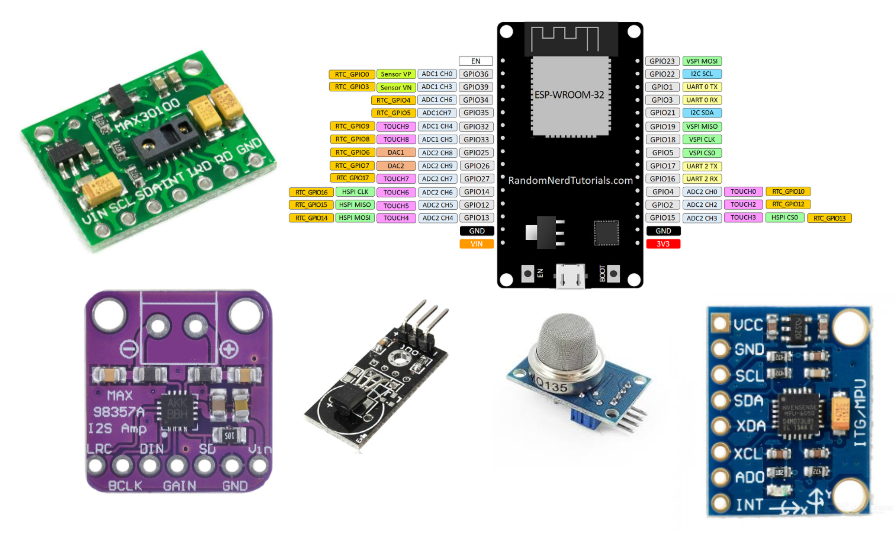
## 1.7. Luật An toàn, vệ sinh lao động số 84/2015/QH13

* Phạm vi và đối tượng áp dụng:
  + Quy định các tiêu chuẩn đảm bảo an toàn, vệ sinh lao động, chế độ với lao động bị tai nạn/bệnh nghề nghiệp, và trách nhiệm các bên gồm người sử dụng lao động, người lao động và cơ quan quản lý nhà nước.
  + Luật áp dụng cho cả người lao động làm theo hợp đồng và người làm việc tự do, bao gồm cả lao động giúp việc, học nghề, lao động thuê ngoài.
* Nguyên tắc đảm bảo ATVSLĐ (Điều 5–6)
  + Mọi người lao động phải được làm việc trong môi trường an toàn, vệ sinh.
  + Ưu tiên các biện pháp phòng ngừa, như loại trừ hoặc kiểm soát mối nguy.
  + Tư vấn, tham vấn có sự tham gia của người lao động, công đoàn, tổ chức ATVSLĐ.
* Quyền và nghĩa vụ
  + Người lao động:
    - Được huấn luyện, trang bị kiến thức ATVSLĐ.
    - Yêu cầu bảo hộ, an toàn, vệ sinh tại nơi làm việc.
    - Có quyền khiếu nại, tố cáo, khởi kiện khi các quyền vi phạm.
  + Người sử dụng lao động:
    - Xây dựng và thựс hiện chương trình, quy trình ATVSLĐ.
    - Cứu hỏa, khám sức khỏe định kỳ, đóng bảo hiểm tai nạn, bệnh nghề nghiệp.
    - Không được buộc người lao động làm việc khi điều kiện nguy hiểm.
    - Thanh tra, kiểm kê, điều tra tai nạn, bệnh nghề nghiệp, phối hợp với công đoàn.
  + Công đoàn và tổ chức xã hội:
    - Tham gia giám sát, tư vấn, tuyên truyền ATVSLĐ.
    - Hỗ trợ, đại diện người lao động khi xảy ra tai nạn, vi phạm.
* Quản lý tai nạn lao động, bệnh nghề nghiệp
  + Trợ cấp, bồi thường cho người lao động bị thương tật hoặc thân nhân nếu tử vong.
  + Khám sức khỏe, giám định y khoa sau tai nạn để xác định mức suy giảm khả năng lao động.
* Quản lý yếu tố nguy hiểm tại nơi làm việc (Thông qua Nghị định 39/2016)
  + Phải định kỳ kiểm kê, đánh giá, kiểm soát yếu tố nguy hiểm (> tiếng ồn, khí, bụi...).
  + Lưu hồ sơ và công khai kết quả cho người lao động.
* Áp dụng luật an toàn lao động trong đề tài:
  + Đảm bảo hệ thống cảm biến được thiết kế đúng tiêu chuẩn điện và an toàn sinh học, gồm cách ly nguồn, vỏ bọc phù hợp, và an toàn kết nối mạng.
  + Người thực hiện dự án cần được huấn luyện cách lắp ráp, vận hành, an toàn điện và bảo hộ khi thử nghiệm.
  + Giai đoạn kiểm nghiệm mô hình nên ghi nhận, báo cáo bất kỳ sự cố nào xảy ra (chập, chấn động, ồn vượt giới hạn) để xử lý kịp thời.
  + Thực hiện kê khai, phản ánh các yếu tố nguy cơ tai nạn (ví dụ rung, nguồn điện không ổn định), đồng thời thực hiện biện pháp khắc phục theo đúng yêu cầu pháp luật.

## 1.8. Phương pháp thực hiện

Đề tài được thực hiện theo các bước chính sau:

* Khảo sát tài liệu: Tìm hiểu các công trình nghiên cứu trước, datasheet của từng cảm biến, các tiêu chuẩn kỹ thuật.
* Thiết kế phần cứng:
  + Linh kiện trong đề tài: ESP32, MAX98357, MPU6050, MAX30100, DS18B20, MQ-135, 2N2222, trở 510R và 10K.



Hình 1.6. Các thiết bị ngoại vi sử dụng trong đề tài.

* + Vẽ và thiết kế mạch PCB: Sử dụng công cụ Easy EDA để vẽ schematic và thiết kế mạch in.
  + Thực hiện lắp ráp: Hàn, khò các linh kiện, thiết bị ngoại vi để kết nối các chân lên mạch in PCB.
* Lập trình và kiểm thử phần mềm:
  + Huấn luyện mô hình AI: Sử dụng cảm biến gia tốc (MP6050) để phát hiện ngã trên nền tảng Edge Impulse.
  + Đọc dữ liệu nhiệt độ (DS18B20), chất lượng không khí (MQ - 135) và nhịp tim & SPO2 (MAX30100).
  + Kiểm thử phần mềm: Kiểm tra độ chính xác của các dữ liệu cảm biến đọc được, so sánh với các thiết bị chuyên dụng có độ chính xác cao để đánh giá.
* Giao tiếp dữ liệu: Cấu hình mạng WiFi, giao thức MQTT, tích hợp với nền tảng ThingsBoard.
* Hiển thị dữ liệu: Thiết kế dashboard theo dõi tình trạng sức khỏe người dùng.
* Thử nghiệm và đánh giá: Kiểm tra hệ thống qua các tình huống mô phỏng, ghi nhận kết quả.

## 1.9. Cấu trúc báo cáo

* Chương 1: Tổng quan, mục tiêu nghiên cứu, phạm vi thực hiện, an toàn và phương pháp.
* Chương 2: Cơ sở lý thuyết về phần cứng, phần mềm, thiết kế và triển khai các nút cảm biến.
* Chương 3: Mô hình hệ thống hoàn chỉnh, kịch bản sử dụng, kết quả thử nghiệm và đánh giá.

## 1.10. Kết luận chương 1

Chương 1 đã trình bày tổng quan về bối cảnh nghiên cứu, vai trò thiết yếu của việc giám sát sức khỏe người cao tuổi, đặc biệt là chức năng phát hiện ngã. Chương cũng đã trình bày rõ mục tiêu, phạm vi nghiên cứu, các yêu cầu an toàn cần thiết, cùng phương pháp thực hiện và ý nghĩa của đề tài. Những nội dung này tạo nền tảng vững chắc để đi sâu vào phân tích chi tiết phần cứng, phần mềm và mô hình hệ thống trong các chương tiếp theo.

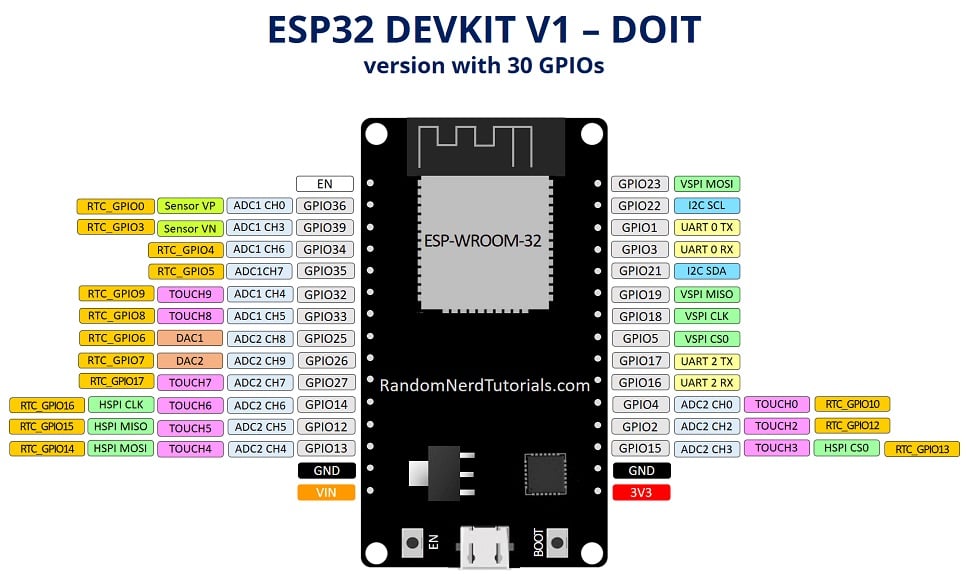
# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ XÂY DỰNG NÚT CẢM BIẾN

## 2.1. Cơ sở lý thuyết về phần cứng

### 2.1.1. Vi xử lý ESP32

2.1.1.1. Tổng quan về ESP32

ESP32 là vi điều khiển trên một vi mạch tích hợp (SoC) của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp. Điểm tốt về ESP32, giống như ESP8266 là các thành phần RF tích hợp của nó như bộ khuếch đại công suất, bộ khuếch đại có độ nhiễu thấp, công tắc ăng-ten, bộ lọc và Balun RF. Điều này làm cho việc thiết kế phần cứng xung quanh ESP32 rất dễ dàng vì ta cần rất ít thành phần bên ngoài. Một điều quan trọng khác cần biết về ESP32 là nó được sản xuất bằng công nghệ 40 nm công suất cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng hoạt động bằng pin như thiết bị đeo, thiết bị âm thanh, đồng hồ thông minh, …, sử dụng ESP32 sẽ rất dễ dàng.



Hình 2.1. ESP32 Devkit V1 và cấu trúc chân.

2.1.1.2. Chức năng chân trên ESP32

Một số chân chức năng của ESP:

Bảng 2.1. Các chân ảnh hưởng đến khởi động ESP32.

|  |  |  |
| --- | --- | --- |
| **Ký hiệu chân** | **GPIO** | **Lưu ý khi khởi động** |
| D2 | GPIO2 | Lỗi nếu kéo lên cao |
| D12 | GPIO12 | Lỗi nếu kéo lên cao |
| D34 | GPIO34 | Lỗi nếu kéo xuống thấp |
| D35 | GPIO35 | Lỗi nếu kéo xuống thấp |

Bảng 2.2. Các chân hỗ trợ DAC.

|  |  |
| --- | --- |
| **Kênh DAC** | **GPIO** |
| DAC1 | GPIO25 |
| DAC2 | GPIO26 |

Bảng 2.3. I2C - Giao tiếp nối tiếp đồng bộ.

|  |  |
| --- | --- |
| **Tín hiệu** | **GPIO mặc định trên ESP32** |
| SDA | GPIO21 |
| SCL | GPIO22 |

Có thể tùy chọn lại SDA/SCL trên bất kỳ GPIO nào.

2.1.1.3. Bộ nhớ ESP32

Một số loại bộ nhớ trong ESP32:

* RAM (SRAM) - Bộ nhớ truy cập ngẫu nhiên:
  + Tổng dung lượng SRAM: ~520 KB.
  + Một phần bộ nhớ RAM có thể bị chiếm bởi hệ điều hành (FreeRTOS hoặc firmware Arduino).
* ROM - Bộ nhớ chỉ đọc:
  + Dung lượng ROM: ~448 KB.
  + Chức năng chính:
    - Chứa bootloader gốc.
    - Các chức năng hệ thống cấp thấp.
    - Xử lý cập nhật OTA, khởi động an toàn (Secure Boot), mã hóa flash.
* Flash - Bộ nhớ lưu trữ chương trình:
  + Dung lượng bộ nhớ Flash thường được phân vùng:
    - bootloader (~40 KB).
    - app (firmware, 1.25 MB).
    - spiffs hoặc littlefs (hệ thống file).
    - nvs (non-volatile storage – lưu cấu hình).
    - otadata (phục vụ cập nhật OTA nếu có).
  + Chức năng:
    - Lưu chương trình firmware người dung.
    - Lưu file hệ thống (LittleFS, SPIFFS).
    - Lưu dữ liệu cấu hình (EEPROM, NVS).

2.1.1.4. Giao thức truyền thông không dây (Wifi, Bluetooth)

ESP32 là một vi điều khiển mạnh mẽ tích hợp hai giao thức truyền thông không dây quan trọng: Wi-Fi và Bluetooth, giúp thiết bị giao tiếp hiệu quả trong các ứng dụng IoT như hệ thống theo dõi sức khỏe và phát hiện ngã. Dưới đây là phân tích chi tiết:

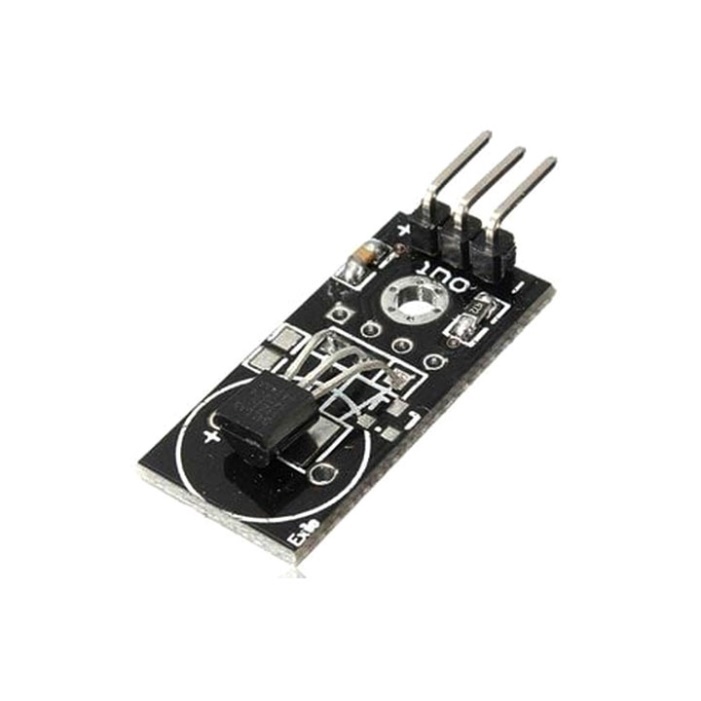
* Giao thức Wi-Fi trên ESP32:
  + Chuẩn hỗ trợ: IEEE 802.11 b/g/n (2.4 GHz).
  + Tốc độ truyền tối đa: ~150 Mbps.
  + Hỗ trợ các chế độ:
    - Station (STA): ESP32 kết nối vào mạng Wi-Fi hiện có.
    - Access Point (AP): ESP32 tạo điểm phát Wi-Fi riêng.
    - STA + AP: Kết hợp cả hai chế độ
  + Ứng dụng thường dùng:
    - Gửi dữ liệu sensor đến ThingsBoard, Firebase, hoặc MQTT broker.
    - Thiết lập webserver mini trên ESP32 để điều khiển và hiển thị dữ liệu.
    - Cập nhật phần mềm từ xa (OTA).
* Giao thức Bluetooth trên ESP32:
  + Loại Bluetooth hỗ trợ:
    - Bluetooth Classic (BR/EDR).
    - Bluetooth Low Energy (BLE).
  + Tính năng BLE nổi bật của ESP32:
    - GATT Server/Client: Cho phép ESP32 đóng vai trò như máy chủ gửi dữ liệu cảm biến đến smartphone.
    - Advertising (quảng bá): Dùng để gửi thông tin cơ bản như tên, trạng thái kết nối.
    - Notification/Indication: Tự động gửi dữ liệu thay đổi cho client (điện thoại, máy tính bảng).
    - BLE Beacon: Phát sóng như iBeacon hoặc Eddystone (Google).
  + Ứng dụng BLE điển hình:
    - Gửi dữ liệu nhịp tim, SpO₂ từ ESP32 đến điện thoại.
    - Dùng smartphone để cấu hình hoặc theo dõi thiết bị không cần Wi-Fi.
    - Nhận cảnh báo khi phát hiện té ngã.

### 2.1.2. Cảm biến nhiệt độ DS18B20

2.1.2.1. Tổng quan về DS18B20

DS18b20 là cảm biến (loại digital) đo nhiệt độ mới của hãng MAXIM với độ phân giải cao (12bit). IC sử dụng giao tiếp 1 - wire rất gọn gàng, dễ lập trình. IC còn có chức năng cảnh báo nhiệt độ khi vượt ngưỡng và đặc biệt hơn là có thể cấp nguồn từ chân data (parasite power).

Cảm biến có thể hoạt động ở 125 độ C. Đây cảm biến kỹ thuật số, nên không bị suy hao tín hiệu đường dây dài. Cảm biến giúp kiểm soát nhiệt môi trường, đo nhiệt độ bên trong các tòa nhà, thiết bị, máy móc và trong hệ thống giám sát. Hình sau là hình ảnh thực tế của module cảm biến nhiệt độ môi trường DS18B20:



Hình 2.2. Hình module DS18B20 thực tế.

2.1.2.2. Thông số kỹ thuật DS18B20

* Nguồn: 3 – 5.5V
* Dải đo nhiệt độ: -55 – 125 độ C ( -67 – 257 độ F)
* Sai số: 0.5 độ C khi đo ở dải -10 – 85 độ C
* Độ phân giải: người dùng có thể chọn từ 9 – 12 bits
* Chuẩn giao tiếp: 1-Wire (1 dây).
* Có cảnh báo nhiệt khi vượt ngưỡng cho phép và cấp nguồn từ chân data.
* Thời gian chuyển đổi nhiệt độ tối đa : 750ms ( khi chọn độ phân giải 12bit ).
* Mỗi IC có một mã riêng (lưu trên EEPROM của IC) nên có thể giao tiếp nhiều DS18B20 trên cùng 1 dây.

### 2.1.3. Cảm biến chất lượng không khí MQ-135

2.1.3.1. Tổng quan về MQ-135

Cảm biến chất lượng không khí MQ-135 sử dụng để kiểm tra chất lượng không khí trong môi trường. Cảm biến có độ nhạy cao khả năng phản hồi nhanh, độ nhạy có thể điều chỉnh được bằng biến trở. Cảm biến chất lượng không khí thường được dùng trong các thiết bị kiểm tra chất lượng không khí bên trong cao ốc, văn phòng, thích hợp để phát hiện NH3, NOx, Ancol, Benzen, khói, CO2,… Cảm biến giúp đọc giá trị tạp chất trong không khí vượt ngưỡng cho phép hay không. Hình sau là hình ảnh thực tế của module cảm biến chất lượng không khí MQ-135:



Hình 2.3. Hình module MQ-135 thực tế.

Cảm biến MQ-135 đo khí CO chuyển thành điện áp đưa ra chân A0. Biến trở trên module có chức năng điều chỉnh điện áp tham chiếu (ngưỡng), khi cảm biến MQ-135 phát hiện khí CO đến ngưỡng thì chân D0 sẽ đảo trạng thái.

2.1.3.2. Thông số kỹ thuật của cảm biến chất lượng không khí MQ-135

* Điện áp nguồn: ≤24VDC.
* Điện áp của heater: 5V±0.1 AC/DC.
* Điện trở tải: thay đổi được (2kΩ-47kΩ).
* Điện trở của heater: 33Ω±5%.
* Công suất tiêu thụ của heater: ít hơn 800mW.
* Khoảng phát hiện: 10 – 300 ppm NH3, 10 – 1000 ppm Benzene, 10 – 300 Alcol.
* Phát hiện nhanh, độ nhạy cao.

2.1.3.3. Các loại khí phát hiện của cảm biến chất lượng không khí MQ-135

* NH3
* NOx
* Ancol
* Benzen
* khói
* CO2
* Khí gas
* Khói

### 2.1.4. Cảm biến nhịp tim & SPO2 trong MAX30100

2.1.4.1. Tổng quan về MAX30100

Cảm biến nhịp tim và oxy trong máu MAX30100 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, cảm biến sử dụng giao tiếp I2C với bộ thư viện sẵn có trên Arduino rất dễ sử dụng. Cảm biến nhịp tim MAX30100 được tích hợp 16-bit sigma delta ADC và bộ xử lý tín hiệu tương tự với độ nhiễu thấp giúp cảm biến hoạt động chính xác và ổn định cao. Đồng thời được thiết kế nhỏ gọn có thể sử dụng làm thiết bị đeo tay và dễ dàng giao tiếp với các MCU như ESP32, Arduino, STM32… Hình sau là hình ảnh thực tế của module cảm biến MAX30100:



Hình 2.4. Hình module MAX30100 thực tế.

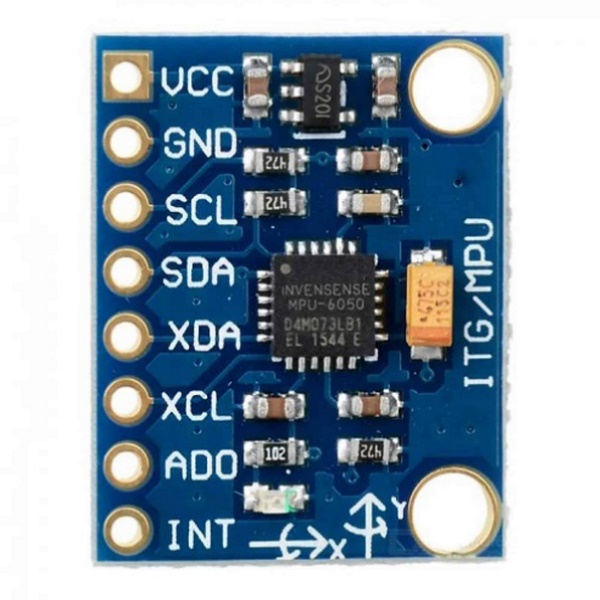
2.1.4.2. Thông số kỹ thuật module cảm biến MAX30100

* Đèn LED tích hợp.
* Nhỏ gọn 5,6mm x 2,8mm x 1,2mm được cải tiến về mặt quang học.
* Hoạt động năng lượng thấp, tăng tuổi thọ pin cho các thiết bị đeo được.
* Tốc độ mẫu có thể lập trình và dòng điện LED cho tiết kiệm điện.
* Dòng tắt máy cực thấp (0,7μA, typ).
* Chức năng nâng cao cải thiện hiệu suất đo lường.
* SNR cao cung cấp khả năng phục hồi chuyển động mạnh mẽ.
* Tích hợp hủy ánh sáng xung quanh.
* Khả năng tỷ lệ mẫu cao.
* Khả năng xuất dữ liệu nhanh.
* Điện áp: 3.3VDC.
* Giao tiếp: I2C, mức tín hiệu TTL.

### 2.1.5. Cảm biến gia tốc MPU6050

2.1.5.1. Tổng quan về MPU6050

Cảm biến IMU MPU6050 được sử dụng để đo 6 thông số: 3 trục Góc quay (Gyro), 3 trục gia tốc hướng (Accelerometer), là loại cảm biến gia tốc phổ biến nhất trên thị trường hiện nay, ví dụ và code dành cho nó rất nhiều và hầu như có trên mọi loại vi điều khiển, nếu ta muốn mua cảm biến gia tốc để làm các mô hình như con lắc động, xe tự cân bằng, máy bay,… thì MPU6050 sẽ là sự lựa chọn tối ưu. MPU6050 dùng để đo gia tốc góc, vận tốc góc, góc quay với độ chính xác lên tới 0.01 độ, đo gia tốc, vận tốc chuyển động tịnh tiến Module cảm biến MPU6050 có thể kết nối với vi điều khiển qua 1 trong 2 giao thức là SPI hoặc I2C. Bên trong cảm biến tích hợp 6 trục cảm biến bao gồm con quay quy hồi 3 trục(trục X, Y và Z) và cảm biến gia tốc 3 trục. MPU6050 còn có 1KB bộ nhớ để lưu trữ lệnh từ vi điều khiển và dữ liệu sau khi nó tính toán xong các giá trị đo được. Hình sau là hình ảnh thực tế của module cảm biến MPU6050:



Hình 2.5. Hình module MPU6050 thực tế.

2.1.5.2. Thông số kỹ thuật module cảm biến MPU6050

* Điện áp sử dụng: 3~5VDC
* Điện áp giao tiếp: 3~5VDC
* Chuẩn giao tiếp: I2C
* Giá trị Gyroscopes trong khoảng: +/- 250 500 1000 2000 degree/sec
* Giá trị Acceleration trong khoảng: +/- 2g, +/- 4g, +/- 8g, +/- 16g
* Board mạch mạ vàng, linh kiện hàn tự động bằng máy chất lượng tốt nhất.
* Chip : MPU-6050
* Giao tiếp: I2C, SPI
* Hỗ trợ AD 16 Bit

2.1.5.3. Chức năng chân trên MPU6050

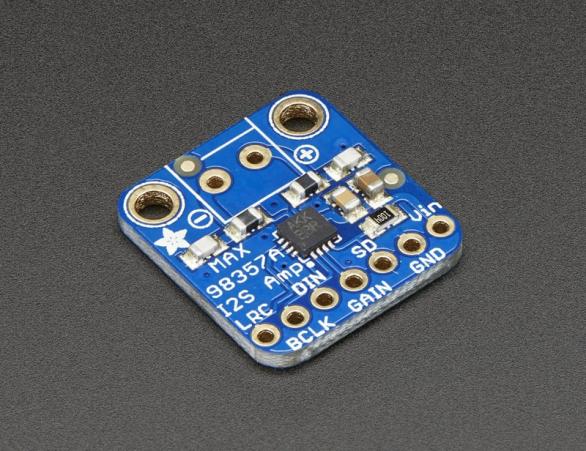
Bảng 2.4. Bảng chức năng chân trên MPU6050.

|  |  |
| --- | --- |
| **Ký hiệu chân** | **Chức năng** |
| VCC | 5V/3V3 |
| GND | 0V |
| SCL | Chân SCL trong giao tiếp I2C |
| SDA | Chân SDA trong giao tiếp I2C |
| XDA | Chân dữ liệu (kết nối với cảm biến khác) |
| XCL | Chân xung (kết nối với cảm biến khác) |
| AD0 | Bit0 của địa chỉ I2C |
| INT | Chân ngắt |

### 2.1.6. Module khuếch đại âm thanh MAX98357

2.1.6.1. Tổng quan về MAX98357

Module sử dụng chip MAX98357A, là một bộ khuếch đại Class‑D đơn kênh tích hợp DAC, nhận tín hiệu âm thanh số chuẩn I2S, giải mã thành tín hiệu analog và khuếch đại trực tiếp đến loa. Thiết kế hiệu suất cao, hoạt động từ 2.7 V đến 5.5 V, cho công suất tối đa 3.2 W (với loa 4 Ω, THD 10%, điện áp 5 V). Class‑D cho khả năng tiết kiệm năng lượng, rất phù hợp với dự án di động, kèm bảo vệ nhiệt và quá dòng. Hình sau là hình ảnh thực tế của module MAX98357:



Hình 2.6. Hình module MAX98357 thực tế.

2.1.6.2. Thông số kỹ thuật module khuếch đại âm thanh MAX98357

* Công suất đầu ra: 3.2 W vào 4 Ω (10% THD, 5 V); 1.8 W vào 8 Ω (10% THD, 5 V).
* PSRR: 77 dB (điển hình @1 kHz).
* Tần số mẫu I2S: 8 kHz – 96 kHz.
* Điện áp hoạt động: 2.7 V – 5.5 V.
* Chức năng giảm tiếng “click-pop”: Có chế độ giảm pop khi bật/tắt nguồn.
* Mức khuếch đại: Lựa chọn qua chân GAIN: 3 dB, 6 dB, 9 dB, 12 dB, 15 dB.
* Bảo vệ: Chống quá nhiệt, bảo vệ quá dòng.

2.1.6.3. Chức năng chân trên MAX98357

Bảng 2.5. Bảng chức năng chân trên MAX98357.

|  |  |
| --- | --- |
| **Ký hiệu chân** | **Chức năng** |
| VIN | 5V/3V3 |
| GND | 0V |
| DIN | Dữ liệu I2S Data In |
| BCLK | Xung clock bit I2S |
| LRC | Xung chọn kênh trái/phải |
| GAIN | Điều chỉnh mức khuếch đại |
| SD | Dùng để bật/tắt chip và chọn kênh mono |

## 2.2. Cơ sở lý thuyết phần mềm và giao thức

### 2.2.1. Các giao thức kết nối module ngoại vi

2.2.1.1. Giao thức I2C (Inter-Integrated Circuit)

* Tổng quan:
  + I2C là giao thức truyền thông đồng bộ, nối tiếp, hai dây, được thiết kế để kết nối nhiều thiết bị trên cùng một bus.
  + Do NXP phát triển, hiện là chuẩn phổ biến trong vi điều khiển.
* Đặc điểm:
  + Dây tín hiệu:
    - SDA (Serial Data) – Truyền dữ liệu
    - SCL (Serial Clock) – Xung đồng hồ
  + Kiểu truyền: Đồng bộ, half-duplex.
  + Sơ đồ liên kết: 1 master → nhiều slave (dựa trên địa chỉ 7-bit hoặc 10-bit).
  + Tốc độ chuẩn:
    - Standard Mode: 100 kHz
    - Fast Mode: 400 kHz
    - Fast Mode Plus: 1 MHz
* Ưu điểm:
  + Kết nối được nhiều thiết bị chỉ với 2 dây.
  + Phổ biến, thư viện hỗ trợ sẵn.
  + Không yêu cầu nhiều chân GPIO.
* Nhược điểm:
  + Tốc độ trung bình – không phù hợp truyền dữ liệu lớn.
  + Dễ nhiễu nếu dây dài, do có điện trở pull-up trên bus.

2.2.1.2. Giao thức I2S

* Tổng quan:
  + I2S là giao thức truyền âm thanh kỹ thuật số, được phát triển bởi Philips.
  + Dùng để giao tiếp giữa vi điều khiển và thiết bị âm thanh số như DAC, microphone kỹ thuật số, module phát loa.
* Đặc điểm kỹ thuật:
  + Dây tín hiệu:
    - WS (Word Select) / LRCLK: xác định kênh trái/phải.
    - SCK (Bit Clock): đồng hồ cho từng bit.
    - SD (Serial Data): dữ liệu PCM.
  + Định dạng dữ liệu:
    - Âm thanh PCM 16-bit / 24-bit / 32-bit.
    - Dạng mono hoặc stereo.
* Ưu điểm:
  + Tốc độ cao, truyền âm thanh thời gian thực.
  + Chuẩn công nghiệp cho thiết bị âm thanh.
  + Được hỗ trợ phần cứng riêng trên ESP32 (khác UART/SPI).
* Nhược điểm:
  + Chỉ dùng được cho tín hiệu âm thanh, không truyền dữ liệu đa dạng như I2C/SPI.
  + Cần cấu hình cẩn thận (tốc độ, độ sâu bit, chế độ...).

2.2.1.3. Giao thức 1 - Wire

* Tổng quan:
  + Giao thức 1-Wire do Dallas Semiconductor (nay là Maxim Integrated) phát triển.
  + Chỉ dùng 1 dây dữ liệu + GND, cực kỳ tiết kiệm chân kết nối.
  + Hoạt động bằng việc gửi xung thời gian đặc biệt để phân biệt bit '1' và '0'.
* Đặc điểm kỹ thuật:
  + Dây tín hiệu: 1 data line (bi-directional).
  + Kiểu truyền: Đồng bộ, half-duplex.
  + Tốc độ: khoảng 15.6 kbps (chuẩn), hoặc 125 kbps (overdrive).
  + Tất cả thiết bị có địa chỉ 64-bit duy nhất.
* Ưu điểm:
  + Rất đơn giản, chỉ cần 1 dây nối nhiều cảm biến.
  + Thiết bị có thể cấp nguồn từ đường data (parasite power).
* Nhược điểm:
  + Tốc độ chậm → không phù hợp truyền dữ liệu lớn.
  + Dễ nhiễu nếu dây dài hoặc nhiều thiết bị trên bus.

2.2.2. Giao thức MQTT (Message Queuing Telemetry Transport)

* Tổng quan:
  + MQTT là giao thức truyền thông nhẹ, dựa trên TCP/IP, được thiết kế cho các ứng dụng IoT có tài nguyên hạn chế.
  + Theo mô hình pub/sub (publish/subscribe) – nơi thiết bị gửi dữ liệu không cần biết cụ thể ai nhận.
* Cấu trúc hoạt động:
  + Client (ESP32): Gửi dữ liệu (Publish) hoặc đăng ký nhận (Subscribe).
  + Broker (ThingsBoard hoặc Mosquitto): Trung gian phân phối dữ liệu đến các client đăng ký.
  + Topic: Chủ đề nội dung dữ liệu (ví dụ: /device/heart\_rate).
* Ưu điểm của MQTT:
  + Siêu nhẹ: Gói tin nhỏ, tiết kiệm băng thông → phù hợp với vi điều khiển.
  + Realtime: Gửi/nhận gần như tức thì.
  + QoS (Quality of Service):
    - QoS 0: gửi 1 lần, không đảm bảo.
    - QoS 1: đảm bảo nhận ít nhất 1 lần.
    - QoS 2: nhận đúng 1 lần duy nhất.
  + Hỗ trợ mã hóa TLS và xác thực người dùng.
* Bảo mật MQTT:
  + ThingsBoard hỗ trợ mã hóa dữ liệu bằng TLS (MQTTs).
  + Có thể cấu hình Access Token, hoặc chứng thực bằng username/password để bảo vệ dữ liệu nhạy cảm.

### 2.2.3. PlatformIO - Công cụ phát triển phần mềm

2.2.3.1. Tổng quan về PlatformIO

PlatformIO là một hệ sinh thái phát triển phần mềm nhúng đa nền tảng, mã nguồn mở, hoạt động như một phần mở rộng của Visual Studio Code (VS Code). Nó hỗ trợ hơn 1000 bo mạch vi điều khiển khác nhau, bao gồm ESP32, Arduino, STM32, Raspberry Pi Pico...



Hình 2.7. Giao diện sử dụng trong PlatformIO.

2.2.3.2. Những tính năng chính

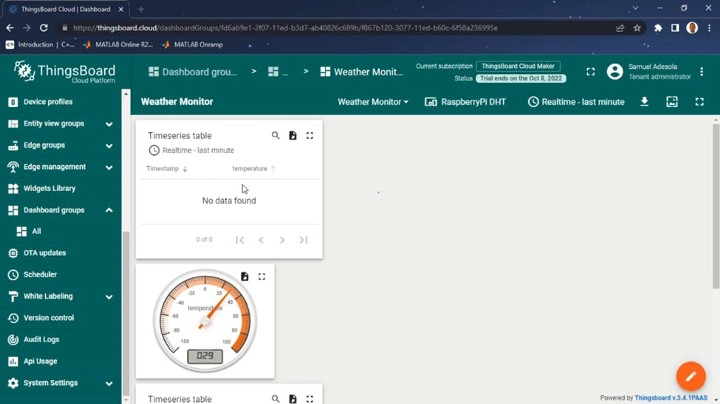
Bảng 2.6. Tính năng chính của PlatformIO.

|  |  |
| --- | --- |
| Tính năng | Mô tả |
| Hỗ trợ nhiều bo mạch | Tự động nhận diện board |
| Quản lý thư viện tự động | Tải và cập nhật thư viện từ PIO Registry |
| Tích hợp Git & Version Control | Hỗ trợ làm việc nhóm, quản lý phiên bản |
| Tích hợp Serial Monitor, Debug, Upload | Nạp code và xem log đơn giản ngay trong VSCode |
| Hỗ trợ FreeRTOS, ESP-IDF | Tích hợp trực tiếp các SDK của Espressif |

### 2.2.4. ThingsBoard - Nền tảng quản lý thiết bị IoT

2.2.4.1. Tổng quan về ThingsBoard

ThingsBoard là một nền tảng IoT mã nguồn mở hỗ trợ việc thu thập, xử lý, lưu trữ và hiển thị dữ liệu từ thiết bị IoT. Nó có thể hoạt động trên nền tảng đám mây (cloud) hoặc máy chủ riêng (on-premise).



Hình 2.8. Giao diện sử dụng của ThingsBoard.

Các thành phần chính:

* Device: Gửi dữ liệu qua MQTT hoặc HTTP.
* Gateway (Tùy chọn): Trung gian giữa nhiều thiết bị và ThingsBoard.
* Broker MQTT: Trung chuyển dữ liệu từ thiết bị đến hệ thống.
* ThingsBoard Server: Xử lý, lưu trữ và điều hướng dữ liệu.
* Dashboard: Hiển thị dữ liệu cảm biến dạng biểu đồ, cảnh báo.

2.2.4.2. Những tính năng chính

* Dashboard trực quan: Vẽ biểu đồ, đồng hồ, trạng thái thiết bị, cảnh báo.
* Rule Engine: Tự động xử lý dữ liệu, tạo cảnh báo (email, SMS, Telegram,...).
* Device Management: Thêm, xóa, nhóm thiết bị, xem trạng thái trực tuyến.
* Bảo mật & phân quyền: Quản lý người dùng, giới hạn truy cập theo vai trò.
* Hỗ trợ MQTT, CoAP, HTTP, LwM2M: Linh hoạt nhiều giao thức.

### 2.2.5. Edge Impulse - Nền tảng phát triển mô hình AI dành cho thiết bị nhúng

2.2.5.1. Tổng quan về Edge Impulse

Edge Impulse là một nền tảng mã nguồn mở (miễn phí cho nghiên cứu) cho phép người dùng xây dựng, huấn luyện và triển khai mô hình học máy (ML) trực tiếp lên các thiết bị nhúng có tài nguyên hạn chế như ESP32, STM32, Arduino Nano 33 BLE, Raspberry Pi,...

Nền tảng này đặc biệt phù hợp với các ứng dụng IoT và thiết bị đeo như:

* Nhận dạng cử chỉ
* Phát hiện ngã
* Nhận dạng âm thanh (ho, tiếng va chạm)
* Phân loại hoạt động con người (Human Activity Recognition)

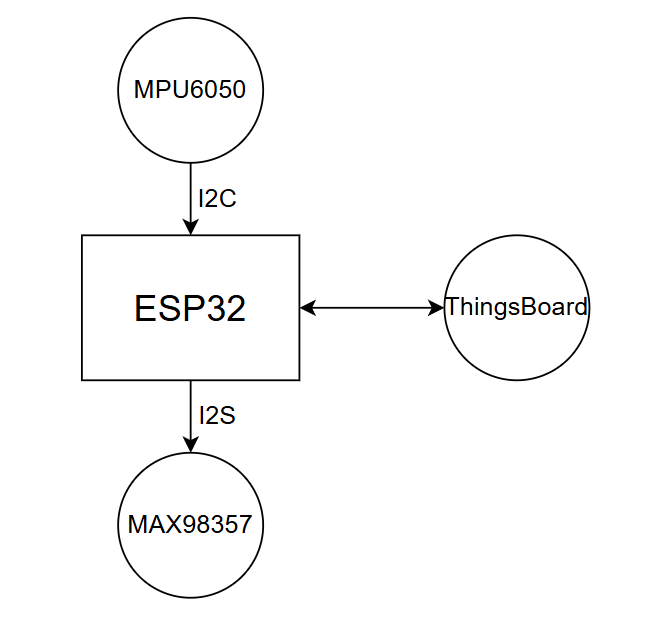
2.2.5.2. Các bước phát triển mô hình với Edge Impulse

* Thu thập dữ liệu: Ghi dữ liệu từ cảm biến (IMU, microphone,...) thông qua điện thoại, ESP32 hoặc thiết bị hỗ trợ.
* Tiền xử lý (Signal Processing): Lọc nhiễu, biến đổi FFT, MFCC (âm thanh), tách cửa sổ thời gian (sliding window).
* Huấn luyện mô hình ML/DL: Dùng các thuật toán như KNN, Random Forest, SVM, hoặc mạng nơ-ron (Neural Network).
* Đánh giá & kiểm thử: Kiểm tra độ chính xác của mô hình với tập dữ liệu kiểm thử.
* Triển khai (Deployment): Biên dịch mô hình thành file nhị phân, thư viện .h nhúng vào phần mềm vi điều khiển (C++, Arduino, STM32Cube, Zephyr OS, WebAssembly...).

## 2.3. Xây dựng nút cảm biến

### 2.3.1. Nút cảm biến 1 - phát hiện té ngã và phát âm thanh cảnh báo

Nút 1 gồm MCU là ESP32, MPU6050 và MAX98357. ESP32 sẽ giao tiếp với MPU6050 bằng giao thức I2C, giao tiếp với MAX98357 bằng giao thức I2S. Hình sau là mô tả về sơ đồ của nút cảm biến:

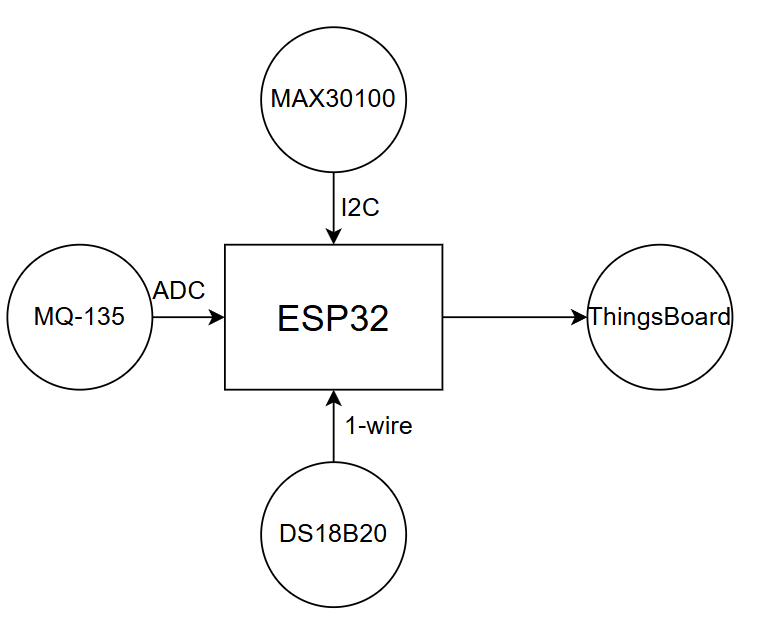


Hình 2.9. Sơ đồ kết nối trong nút 1.

Trong nút 1, chúng ta sẽ thu thập dữ liệu trực tiếp ở trên cảm biến MPU6050 để phân loại chuyển động và phân biệt giữa ngã với các chuyển động xuống tương tự như ngồi, xuống bậc thang… Sau khi huấn luyện trên Edge Impulse và nhúng mô hình vào trong ESP32, ta lập trình trên PlatformIO để khi phát hiện ngã sẽ phát âm thanh qua MAX98357, đồng thời gửi tín hiệu cảnh báo lên ThingsBoard để cảnh báo người dùng từ xa.

### 2.3.2. Nút cảm biến 2 - theo dõi sức khỏe

Nút 2 gồm MCU ESP32, MAX30100, DS18B20 và MQ135. ESP32 sẽ giao tiếp với MAX30100 bằng giao thức I2C, giao tiếp với DS18B20 bằng giao thức 1 - wire, đọc cảm biến MQ - 135 qua chân ADC trên ESP32. Hình sau là mô tả về sơ đồ của nút cảm biến:



Hình 2.10. Sơ đồ kết nối trong nút 2.

Trong nút 2, chúng ta sẽ đọc dữ liệu từ các cảm biến MAX30100, DS18B20 và MQ-135 rồi gửi lên ThingsBoard. Trên ThingsBoard Cloud, ta sẽ thiết kế các ngưỡng để gửi cảnh báo về máy khi các giá trị từ cảm biến vượt quá ngưỡng cho phép.

## 2.4. Kết luận chương 2

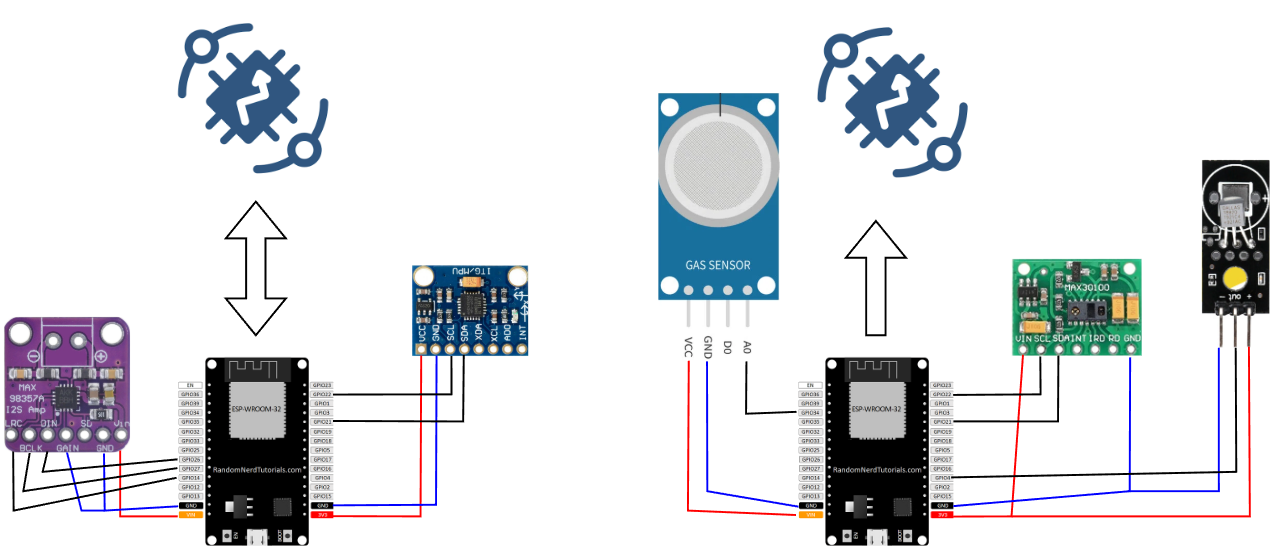
Chương 2 đã trình bày các cơ sở lý thuyết và công nghệ nền tảng phục vụ cho việc thiết kế và triển khai hệ thống giám sát sức khỏe và phát hiện ngã dựa trên nền tảng IoT. Về phần cứng, chương đã phân tích chi tiết vai trò và nguyên lý hoạt động của các linh kiện chính bao gồm: vi điều khiển ESP32, cảm biến nhiệt độ DS18B20, cảm biến chất lượng không khí MQ-135, cảm biến nhịp tim và SpO₂ MAX30100, cảm biến gia tốc MPU6050 và module âm thanh MAX98357. Đây là những thành phần then chốt giúp thu thập và xử lý các tín hiệu liên quan đến sức khỏe và hoạt động của người dùng.

Về phần mềm, chương cũng đã giới thiệu các giao thức kết nối phổ biến như I2C, UART và OneWire, cùng với các công cụ phát triển phần mềm như PlatformIO, nền tảng ThingsBoard để hiển thị dữ liệu và Edge Impulse để xây dựng mô hình trí tuệ nhân tạo (AI) phát hiện ngã. Những nền tảng này cho phép tích hợp toàn diện giữa phần cứng, phần mềm và dữ liệu theo thời gian thực, tạo ra một hệ thống theo dõi thông minh, linh hoạt và có khả năng mở rộng.

Cuối cùng, chương đã trình bày chi tiết quá trình xây dựng hai nút cảm biến: một nút chuyên dùng để phát hiện té ngã và phát âm thanh cảnh báo, một nút dùng để theo dõi các chỉ số sức khỏe quan trọng. Đây là bước nền quan trọng trước khi triển khai mô hình hệ thống hoàn chỉnh trong chương tiếp theo.

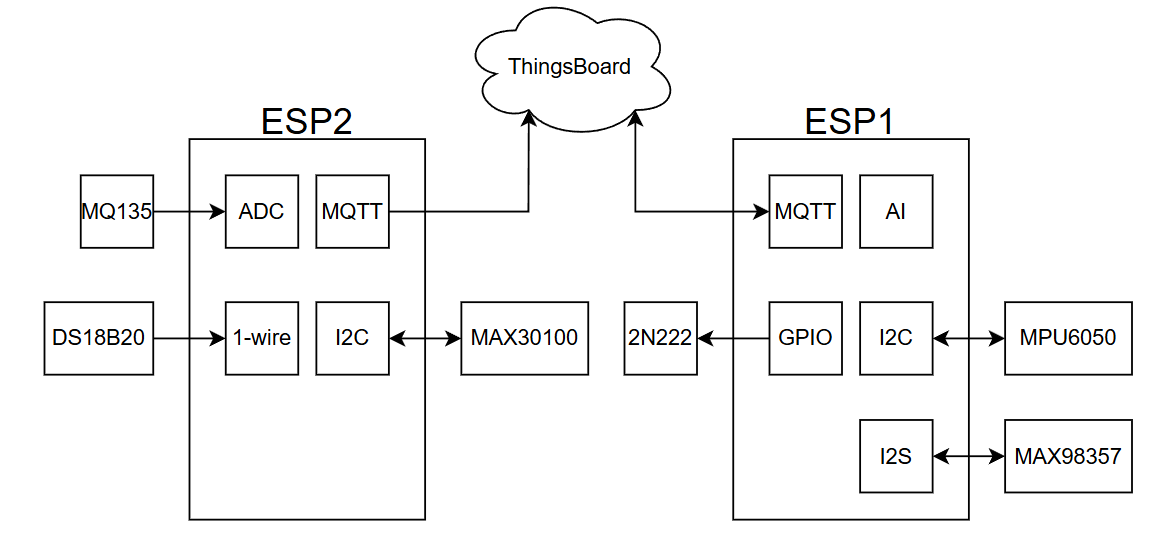
# CHƯƠNG 3: MÔ HÌNH, KỊCH BẢN VÀ ĐÁNH GIÁ HỆ THỐNG

## 3.1. Mô hình hệ thống



Hình 3.1. Sơ đồ kết nối chân của hệ thống.

Hình trên là mô hình hệ thống chi tiết và các chân kết nối của mô hình, hình dưới là sơ đồ nguyên lý của mô hình hệ thống.

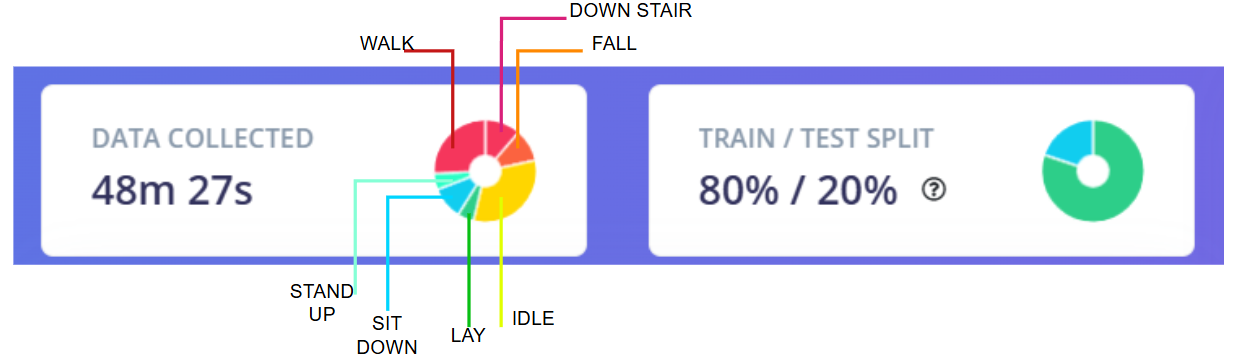


Hình 3.2. Sơ đồ nguyên lý hệ thống.

### 3.1.1. Mô hình nút cảm biến 1

3.1.1.1. Mô hình AI phát hiện ngã

Để đảm bảo kết quả nhận diện chính xác và trực quan nhất, nhóm đã tiến hành huấn luyện trực tiếp mô hình trí tuệ nhân tạo bằng dữ liệu thu thập từ cảm biến MPU6050 kết hợp với vi điều khiển ESP32 thông qua nền tảng Edge Impulse.



Hình 3.3. Thời gian thu thập dữ liệu, các nhãn và tỷ lệ giữa huấn luyện - kiểm thử.

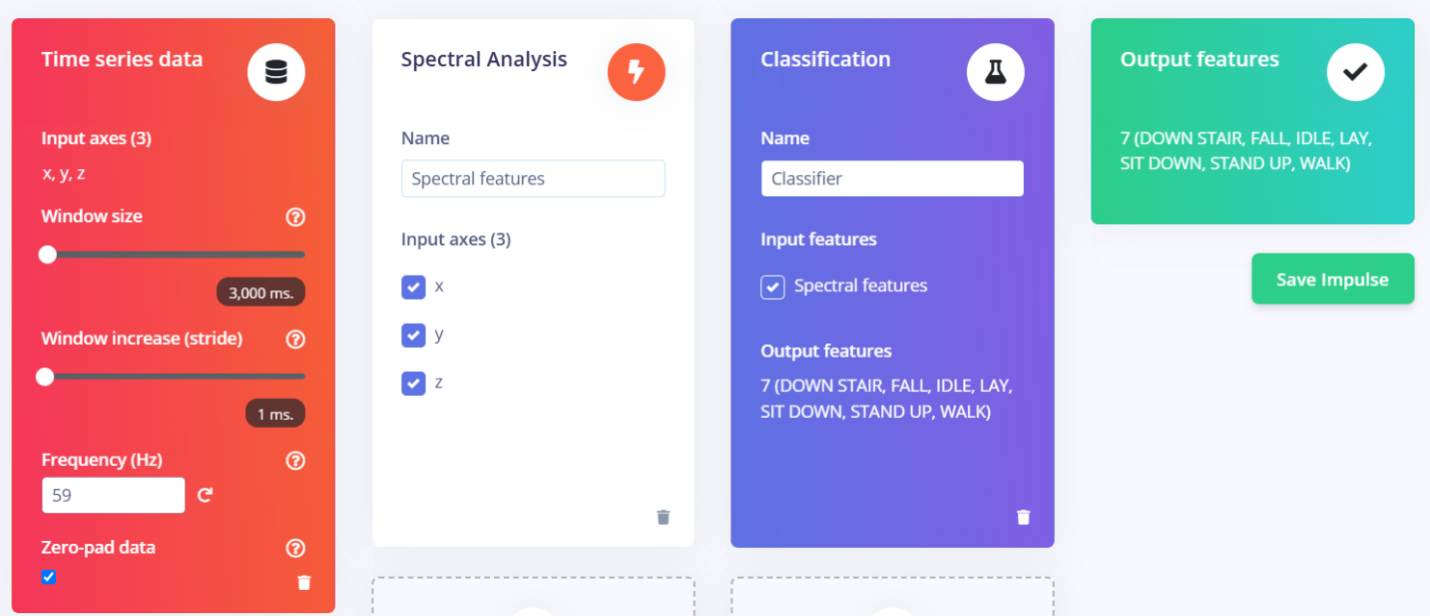
Quan sát hình minh họa, có thể thấy mô hình đã được huấn luyện với tổng cộng 7 nhãn hành vi, thời lượng dữ liệu là 48 phút 27 giây. Tập dữ liệu được chia theo tỷ lệ 80% cho huấn luyện và 20% cho kiểm thử, đây là tỷ lệ phổ biến và được khuyến nghị trong huấn luyện mô hình học máy trên Edge Impulse.

Mặc dù mục tiêu chính của mô hình là phát hiện té ngã, việc phân biệt chính xác giữa các hành vi hàng ngày của con người là điều cần thiết để tránh cảnh báo sai. Do đó, quá trình huấn luyện bao gồm các nhãn sau:

* IDLE: Đứng yên
* WALK: Đi bộ
* DOWN STAIR: Xuống cầu thang
* SIT DOWN: Ngồi xuống
* STAND UP: Đứng lên
* LAY: Nằm xuống
* FALL: Ngã

Việc xây dựng một tập nhãn đa dạng, đặc biệt là các hành vi có đặc điểm chuyển động tương đồng với té ngã, giúp mô hình nâng cao độ chính xác trong phân loại và giảm thiểu nhầm lẫn giữa các hành động gần giống.

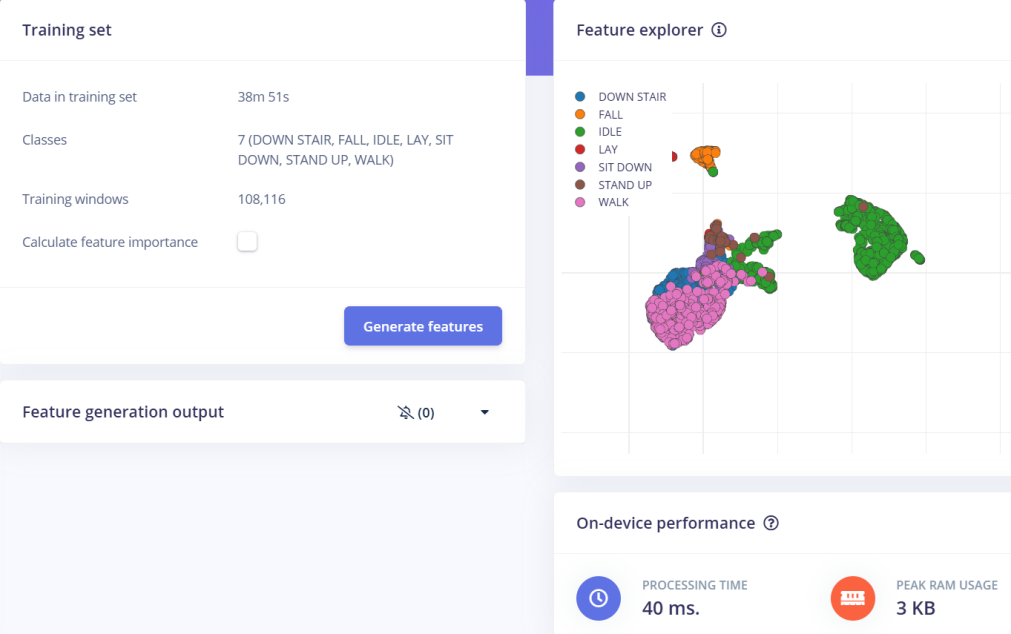
Sau khi thu thập dữ liệu cho huấn luyện và kiểm thử, ta tới phần Create impulse. Đây là bước đầu tiên để ta bắt đầu định nghĩa cấu trúc impulse của mình. Ta sẽ chọn loại dữ liệu đầu vào, cách tiền xử lý dữ liệu đó, và mô hình học máy sẽ được sử dụng.



Hình 3.4. Phần cấu hình trong Create impulse.

* Dữ liệu chuỗi thời gian (Time series data): Cấu hình cách dữ liệu cảm biến thô được cắt thành các "cửa sổ" với kích thước và bước nhảy nhất định, sẵn sàng cho việc xử lý tiếp theo. Ở đây lần lượt có các tham số:
  + Input axes (3): Số lượng trục dữ liệu đầu vào. Ở đây x, y, z là 3 trục gia tốc của gia tốc kế MPU6050.
  + Window size: Kích thước cửa sổ thời gian, mỗi mẫu huấn luyện sẽ được trích xuất từ 3 giây dữ liệu liên tiếp.
  + Window increase (stride): Bước nhảy của cửa sổ, sau mỗi mẫu được trích xuất trong 3 giây, cửa sổ sẽ dịch sang phải 1ms để lấy mẫu kế tiếp.
  + Frequency (Hz): Tần số lấy mẫu, cảm biến đang được cấu hình để lấy 59 mẫu mỗi giây.
* Phân tích phổ (Spectral Analysis): Chuyển đổi dữ liệu chuỗi thời gian đã được cắt cửa sổ từ miền thời gian sang miền tần số, trích xuất các "tính năng phổ" quan trọng. Đây là bước tiền xử lý để làm nổi bật các đặc điểm cần thiết cho mô hình.
* Phân loại (Classification): Sử dụng các tính năng phổ đã được trích xuất làm đầu vào để huấn luyện một mô hình học máy (bộ phân loại). Mô hình này sẽ học cách nhận diện và gán dữ liệu vào một trong các "lớp" đầu ra đã định nghĩa.
* Cuối cùng, sau khi cấu hình các bước này, ta sẽ "Lưu Impulse" để tiếp tục quá trình huấn luyện và triển khai mô hình.

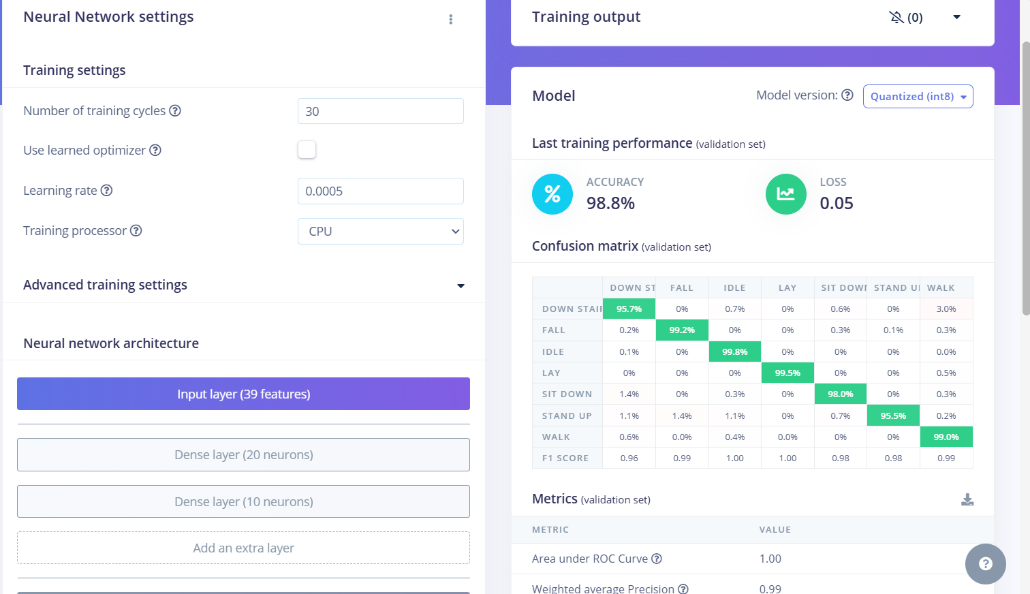
Phần tiếp theo sẽ là Spectral features. Phần này giúp trích xuất các đặc trưng tần số từ dữ liệu thô, biến chúng thành định dạng mà mô hình học máy có thể hiểu được.



Hình 3.5. Kết quả sau khi Generate features trong phần Spectral features.

"Generate features" (Tạo các đặc trưng) biến dữ liệu thô đã được dán nhãn thành các "đặc trưng" mà mô hình học máy có thể hiểu được, và "Feature explorer" cung cấp cái nhìn trực quan về chất lượng của các đặc trưng đó trước khi ta tiến hành huấn luyện mô hình. Ngoài ra, Edge Impulse sẽ cho chúng ta biết Processing time là 40 ms (thời gian cần thiết để thiết bị xử lý xong một cửa sổ dữ liệu đầu vào và trích xuất đặc trưng từ dữ liệu đó) và Peak RAM usage là 3 KB (dung lượng bộ nhớ RAM tối đa mà mô hình sử dụng trong quá trình xử lý đặc trưng trên thiết bị).

Phần tiếp là Classifier. Đây là phần ta sẽ chọn và cấu hình thuật toán học máy để phân loại hoặc nhận diện các mẫu trong dữ liệu đã được tiền xử lý.



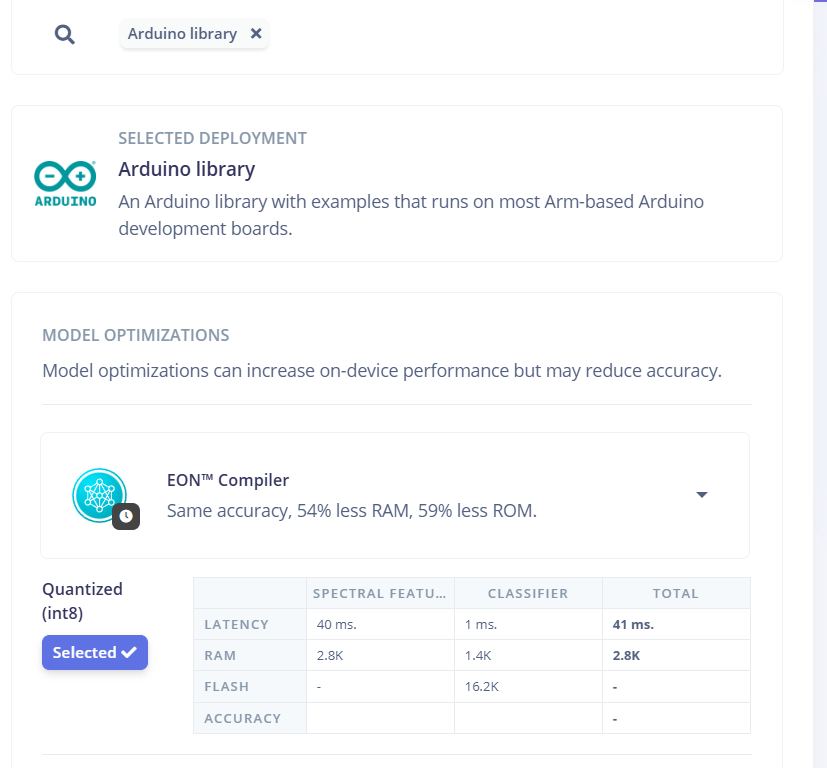
Hình 3.6. Kết quả Sau khi Training với tỷ lệ chính xác của các Nhãn cao.

Hình ảnh này cho thấy phần "Neural Network settings" (Cài đặt Mạng Nơ-ron) và "Training output" (Kết quả huấn luyện), là bước tiếp theo sau khi bạn đã tạo các đặc trưng. Đây chính là nơi bạn cấu hình và huấn luyện mô hình học máy (Classifier) và xem kết quả của quá trình huấn luyện.

Theo kết quả trong hình, tỷ lệ chính xác của các Nhãn thấp nhất là 95.5% và cao nhất là 99.8%, độ chính xác trung bình 98.8%. Đây là một kết quả thể hiện rằng mô hình có khả năng nhận diện chính xác từng loại hoạt động một cách vượt trội.

Điểm F1, một chỉ số cân bằng giữa Precision (độ chính xác) và Recall (độ thu hồi). Điểm F1 càng gần 1.00 càng tốt. Các điểm F1 cao cho tất cả các lớp (0.96 đến 1.00) càng khẳng định hiệu suất xuất sắc của mô hình.

Phần cuối cùng, Deployment (Triển khai). Đây là bước cuối cùng trong quy trình phát triển. Nếu đã hài lòng với hiệu suất của mô hình, phần này cho phép ta xuất mô hình đã huấn luyện thành một định dạng có thể chạy trên ESP32.



Hình 3.7. Chọn thiết bị và định dạng cho mô hình AI.

Quantized (int8) là kỹ thuật lượng tử hóa mô hình, trong đó toàn bộ trọng số và dữ liệu từ định dạng số thực (float32) được chuyển sang định dạng số nguyên 8-bit (int8). Việc lượng tử hóa giúp đạt được các lợi ích sau:

* Giảm kích thước mô hình.
* Tiết kiệm RAM và Flash.
* Tăng tốc độ suy luận.
* Giảm điện năng tiêu thụ trên thiết bị nhúng.

Dưới đây là các thông số kỹ thuật sau khi mô hình được huấn luyện và lượng tử hóa:

* Latency (Thời gian xử lý 1 mẫu dữ liệu):
  + Spectral features: 40 ms – thời gian trích đặc trưng.
  + Classifier: 1 ms – thời gian mô hình phân loại.
  + Total: 41 ms – tổng thời gian xử lý.
* RAM usage:
  + Spectral features: 2.8 KB – RAM dùng khi trích đặc trưng.
  + Classifier: 1.4 KB – RAM dùng cho mô hình.
  + Total: 2.8 KB – RAM tối đa hệ thống cần.
* Flash usage:
  + Classifier: 16.2 KB – dung lượng bộ nhớ flash dùng để lưu mô hình.

Với những thông số trên, có thể thấy mô hình AI này có kích thước gọn nhẹ, tốc độ xử lý nhanh và tiêu tốn rất ít tài nguyên. Do đó, nó hoàn toàn phù hợp để triển khai trực tiếp trên các nền tảng phần cứng hạn chế như ESP32 trong ứng dụng phát hiện té ngã theo thời gian thực.

3.1.1.2. Phát âm thanh khi phát hiện ngã

Cấu hình I2S giao tiếp với MAX98357:

* GPIO 27: BCLK (Bit clock)
* GPIO 14: LRCLK (Word Select)
* GPIO 26: DATA (DOUT)

Ngoài ra, tham số data\_in\_num = -1 được sử dụng để vô hiệu hóa chức năng thu âm, do MAX98357 là module chỉ phát âm thanh, không có khả năng ghi âm. (Đối với các module micro như INMP441, điều này sẽ ngược lại: không có data\_out, nhưng vẫn cần cấu hình chân data\_in để thu dữ liệu.)

Trong hệ thống, mô hình AI sẽ được sử dụng để phát hiện té ngã. Khi nhãn FALL được nhận diện với xác suất trên 80%, hệ thống sẽ tự động kích hoạt cảnh báo bằng cách phát tệp âm thanh alarm.wav từ bộ nhớ SPIFFS trong 30 giây.

3.1.1.3. Điều khiển từ xa GPIO với RPC

ESP32 sẽ đăng ký hàm callback để xử lý các yêu cầu từ xa (RPC) bằng cách thiết lập một hàm được gọi tự động mỗi khi có tin nhắn MQTT đến. Đây là cơ chế cho phép thiết bị phản hồi các lệnh điều khiển được gửi từ nền tảng ThingsBoard.

Cụ thể, ESP32 sẽ đăng ký lắng nghe trên một chủ đề MQTT dành riêng cho các lệnh RPC. ThingsBoard sẽ gửi lệnh đến chủ đề này dưới dạng một chuỗi JSON, ví dụ:

{

"method": "control\_gpio",

"params": true

}

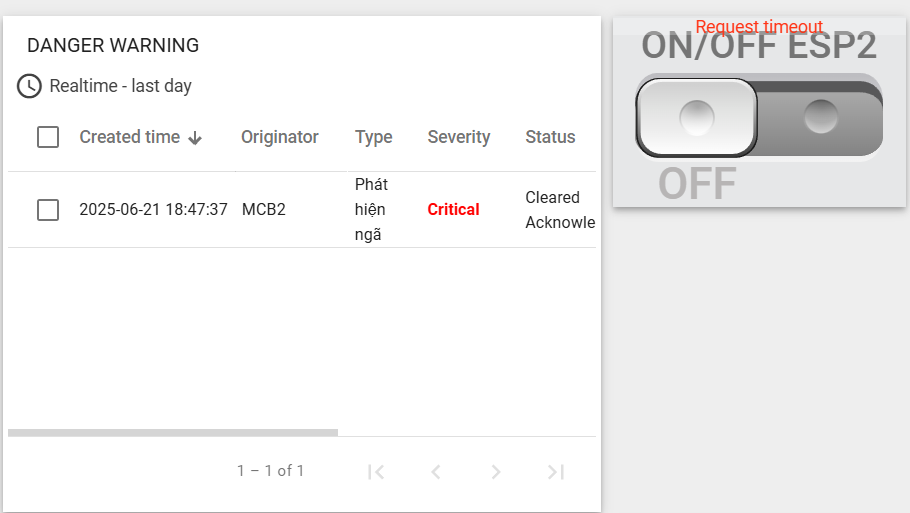
Khi nhận được lệnh, hàm rpcCallback() sẽ được gọi. Hàm này có nhiệm vụ giải mã nội dung JSON và kiểm tra tên phương thức ("method"). Nếu phương thức là "control\_gpio", thì:

* Nếu "params" là true, ESP32 sẽ kéo chân GPIO xuống mức thấp (LOW).
* Nếu "params" là false, GPIO sẽ được kéo lên mức cao (HIGH).

3.1.1.4. Thiết kế DashBoard và Rule Chain của nút 1 trên nền tảng ThingsBoard

Trong DashBoard, ta sẽ tạo ra 2 Widget bao gồm:

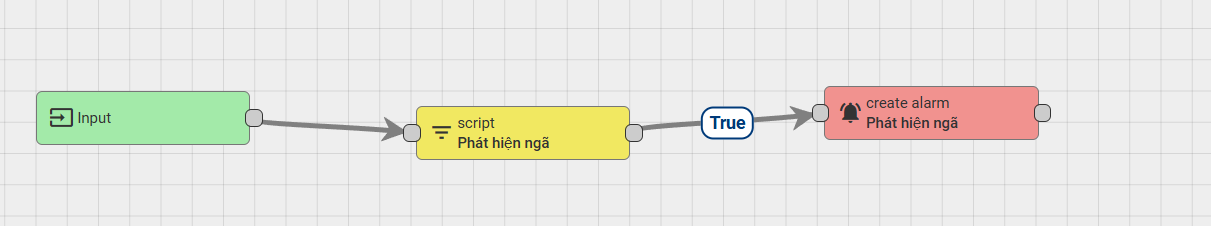
* Alarms table: Bảng nhận các cảnh báo phát hiện ngã.
* Switch Control: Công tắc bật tắt GPIO.



Hình 3.8. DashBoard của nút cảm biến 1.

Phần Update value settings (Gửi lệnh điều khiển) là phần quan trọng để gửi lệnh từ widget tới ESP32 khi bạn nhấn nút ON/OFF. RPC set value method là tên phương thức được gửi xuống khi bật/tắt nút. Tên này sẽ là control\_gpio, trùng với hàm xử lý RPC trong ESP32.

Tiếp theo là cấu hình Rule chain để gửi cảnh báo tới Alarms table:



Hình 3.9. Rule Chain gửi cảnh báo phát hiện ngã.

* Input: Dữ liệu đầu vào mà ESP32 gửi lên ThingsBoard
* Script: Phát hiện ngã (ESP32 sẽ liên tục gửi lên ThingsBoard 1 giá trị 0/1 để xác định có người ngã hay không theo). Script sẽ kiểm tra biến đầu vào alarm, và chỉ khi nó >0 (tức alarm = 1) thì mới tới Create alarm.
* Create alarm: Gửi cảnh báo, chọn mức độ nghiêm trọng là Critical (Nguy cấp) để phản ánh mức độ nghiêm trọng trong cảnh báo.

Cuối cùng, sau khi đã hoàn thành, ta chọn Lưu và qua Root Rule Chain. Root Rule Chain là chuỗi xử lý mặc định đầu tiên mà mọi dữ liệu gửi từ thiết bị IoT sẽ đi qua khi được gửi lên ThingsBoard. Hình sau là cách thiết kế để gửi cảnh báo ngã tới Alarms table:



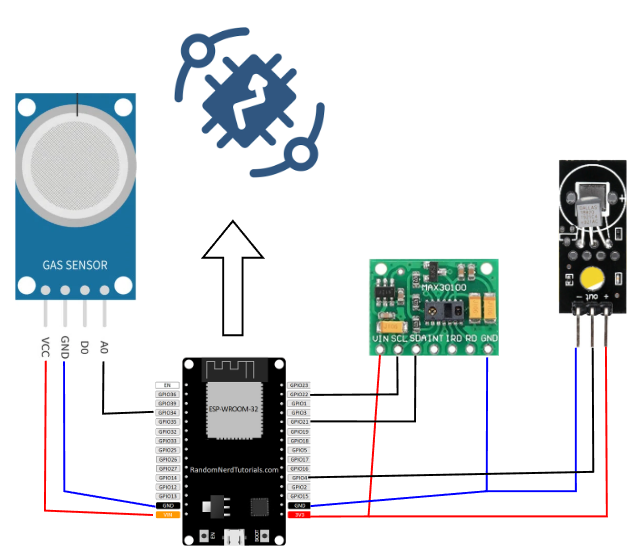
Hình 3.10. Kết nối Rule chain vừa tạo với Save Timeseries.

* Nút "Save Timeseries" có nhiệm vụ lưu trữ dữ liệu chuỗi thời gian (telemetry data) từ các thiết bị vào cơ sở dữ liệu của ThingsBoard.
* Nút "rule chain" là bộ gọi Rule Chain con (Sub Rule Chain). Thay vì thực hiện một hành động cụ thể, nó chuyển tiếp thông điệp đến một Rule Chain "Phát hiện ngã". Rule Chain "Phát hiện ngã" mà ta cấu hình trước đó sẽ chứa logic chi tiết để xử lý dữ liệu và xác định xem có phải là một sự kiện ngã hay không rồi gửi cảnh báo tới Alarms table.

### 3.1.2. Mô hình nút cảm biến 2

3.1.2.1. Đọc dữ liệu từ các cảm biến rồi gửi lên ThingsBoard

Mô hình sẽ lần lượt đọc nhịp tim và SpO2 từ cảm biến MAX30100, đo chất lượng không khí bằng MQ135 và đo nhiệt độ với cảm biến DS18B20.



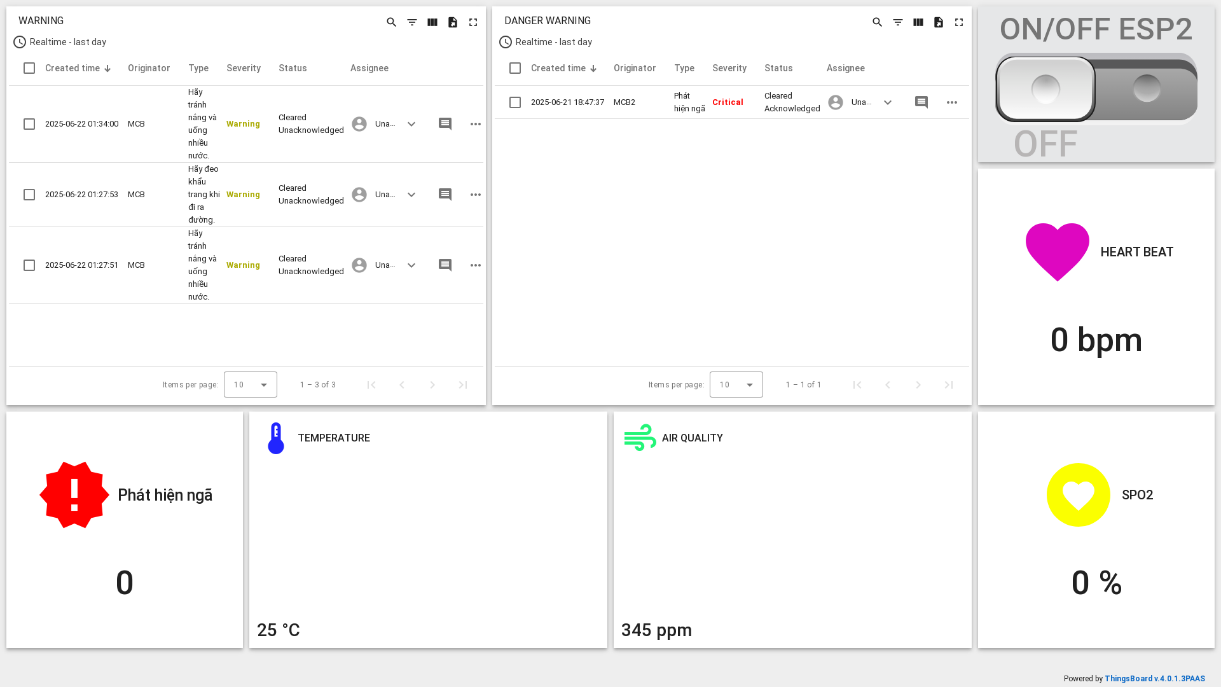
Hình 3.11. Mô hình kết nối chân trong nút cảm biến 2.

Cảm biến MAX30100 (đo nhịp tim và SpO₂) không có bộ nhớ đệm dữ liệu lớn, nên nếu bạn không gọi hàm pox.update() thường xuyên (khoảng mỗi 10–20ms), nó sẽ bị trễ, mất nhịp, hoặc trả về giá trị sai. Vậy nên việc sử dụng 2 task gồm taskMAX30100 và taskSensorRead chạy song song giúp cho việc sử dụng trơn tru nhất có thể.

Sau khi đọc xong dữ liệu từ các cảm biến nêu trên, ESP32 sẽ gửi các dữ liệu này tới máy chủ ThingsBoard bằng Access Token đã đăng ký.

3.1.2.2. Thiết kế DashBoard và Rule Chain của nút 2 trên nền tảng ThingsBoard

Trong DashBoard, ta sẽ gọi thêm các Widget như hình sau:

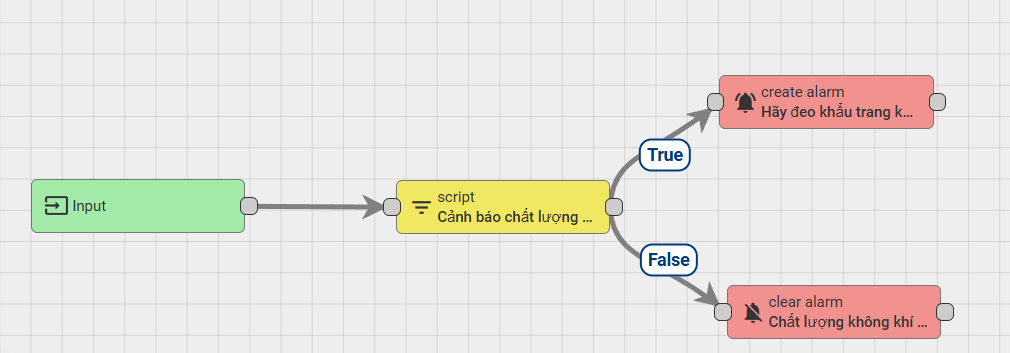


Hình 3.12. DashBoard của cả hệ thống (gồm cả nút 1 và 2).

* Alarm table (WARNING): Bảng nhận các cảnh báo lưu ý liên quan tới sức khỏe.
* Biểu đồ Simple Value and chart card (TEMPERATURE và AIR QUALITY): Biểu đồ thể hiện các giá trị của Nhiệt độ và Chất lượng không khí.
* Value card (HEART BEAT và SPO2): Thể hiện giá trị các thông số của nhịp tim và SPO2 trong máu.

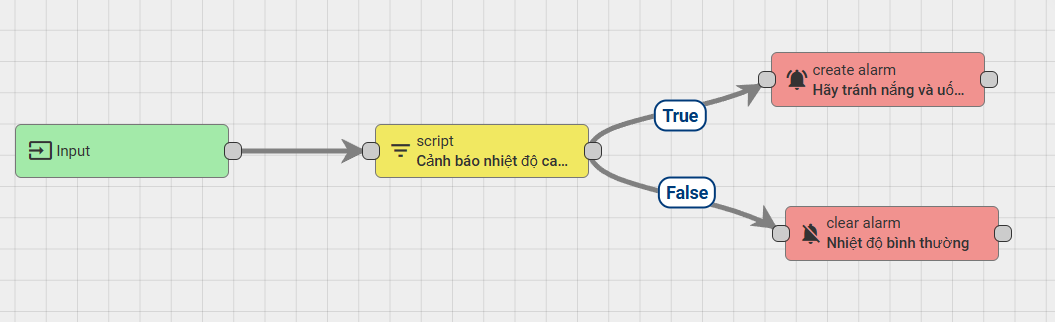
Tại mỗi widget, ta thiết lập cần lựa chọn đúng Data key (khóa dữ liệu) tương ứng với tên trường dữ liệu mà thiết bị gửi lên. Điều này đảm bảo thông tin được hiển thị chính xác và đúng vị trí trên giao diện giám sát.

Tiếp theo là cấu hình Rule Chains để gửi các cảnh báo liên quan tới sức khỏe:



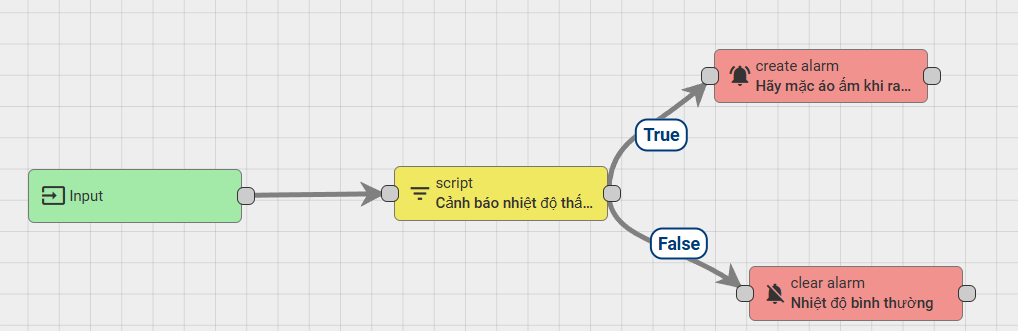
Hình 3.13. Rule Chain cảnh báo chất lượng không khí kém.

Khi key data air\_quality từ ESP32 gửi tới quá 500, Rule Chain này sẽ có chức năng gửi cảnh báo rằng: Chất lượng không khí đang ở mức xấu, hãy đeo khẩu trang khi ra đường.



Hình 3.14. Rule Chain cảnh báo nhiệt độ cao.

Khi key temperature từ ESP32 gửi tới quá 33 (tức nhiệt độ trên 33 độ C), Rule Chain này sẽ có chức năng gửi cảnh báo rằng: Nhiệt độ đang ở mức cao, hãy tránh nắng và uống nhiều nước.

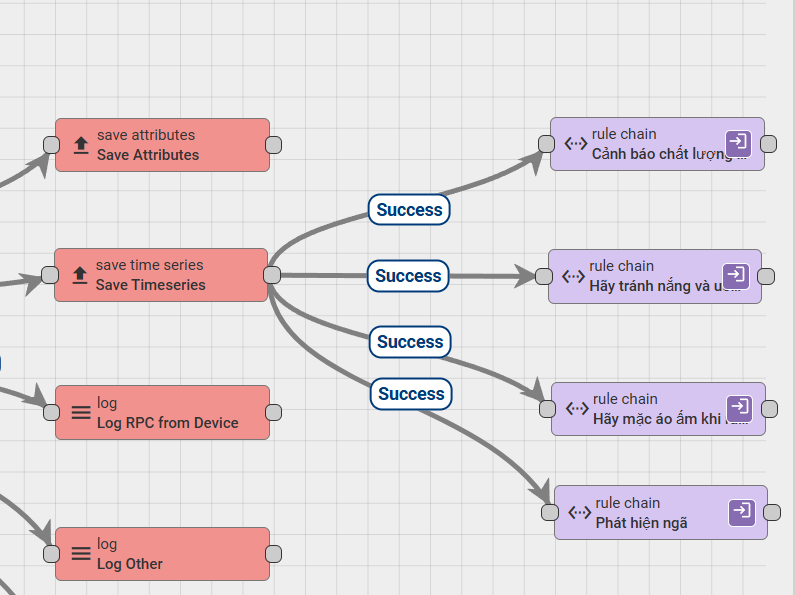


Hình 3.15. Rule Chain cảnh báo nhiệt độ thấp.

Khi key temperature từ ESP32 gửi dưới ngưỡng 15 (tức nhiệt độ dưới 15 độ C), Rule Chain này sẽ có chức năng gửi cảnh báo rằng: Nhiệt độ đang ở mức thấp, hãy mặc áo ấm khi ra ngoài trời.

Cả 3 cảnh báo trên sẽ được gửi với mức độ nghiêm trong Warning, thấp hơn 1 mức so với Critical và tự động xóa cảnh báo khi nhiệt độ bình thường.

Sau khi đã hoàn thành, ta chọn Lưu và qua Root Rule Chain



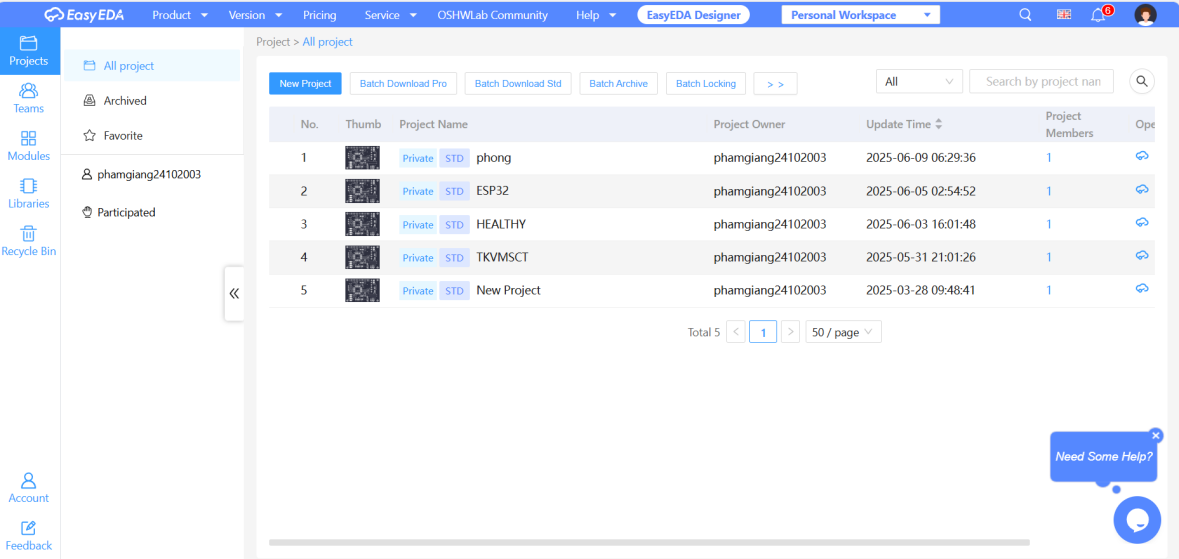
Hình 3.16. Kết nối tất Rule chain với Save Timeseries.

Tương tự như rule chain Phát hiện ngã, Save Timeseries sẽ chuyển tiếp thông điệp đến các Rule Chain mà ta cấu hình trước đó, xử lý dữ liệu và gửi cảnh báo tới Alarms table nếu các ket data vượt quá ngưỡng.

### 3.1.3. Thiết kế PCB trên Easy EDA

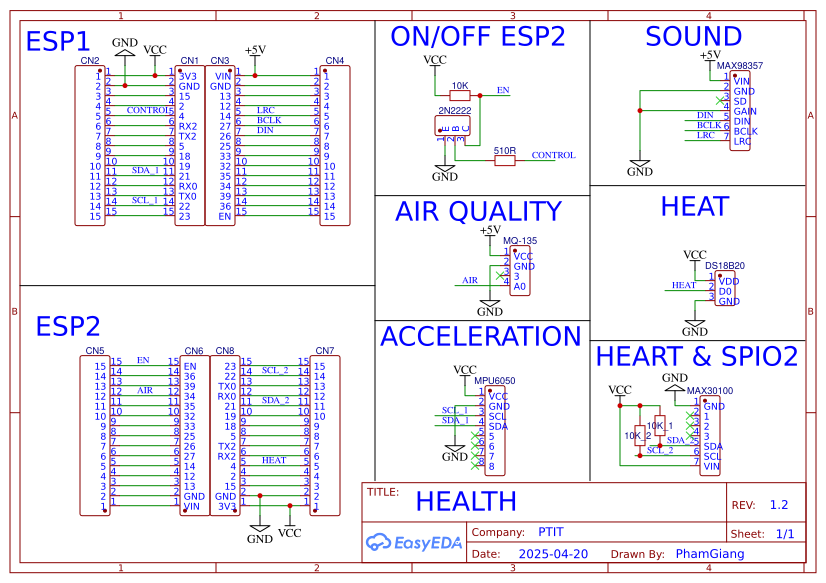
EasyEDA là một công cụ thiết kế mạch điện tử miễn phí và chạy trực tuyến (cloud-based), cho phép người dùng:

* Vẽ sơ đồ nguyên lý (schematic)
  + Vẽ mạch điện đơn giản đến phức tạp bằng cách kéo thả linh kiện.
  + Thư viện linh kiện phong phú, hỗ trợ cả custom symbol.
* Thiết kế mạch in PCB (PCB Layout)
  + Tự động chuyển từ sơ đồ mạch sang bố trí PCB.
  + Hỗ trợ nhiều lớp (2 lớp, 4 lớp,...).
  + Có chức năng auto-routing và kiểm tra lỗi mạch (DRC).
* Liên kết với JLCPCB: Sau khi thiết kế xong, bạn có thể đặt hàng in PCB trực tiếp từ JLCPCB với vài cú nhấp chuột.
* Chạy trực tiếp trên trình duyệt: Không cần cài đặt phần mềm, chỉ cần đăng ký tài khoản.



Hình 3.17. Giao diện sử dụng của Easy EDA.

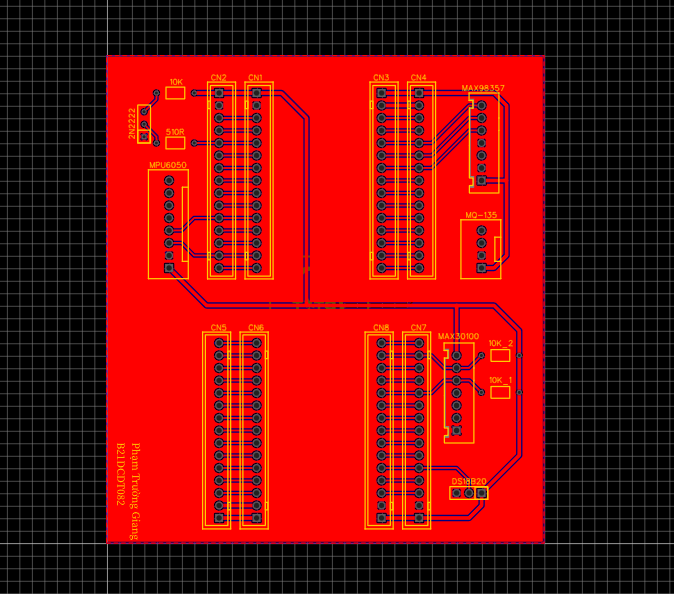
Hình sau là sơ đồ nguyên lý (Schematic) của mô hình:



Hình 3.18. Schematic của mô hình hệ thống.

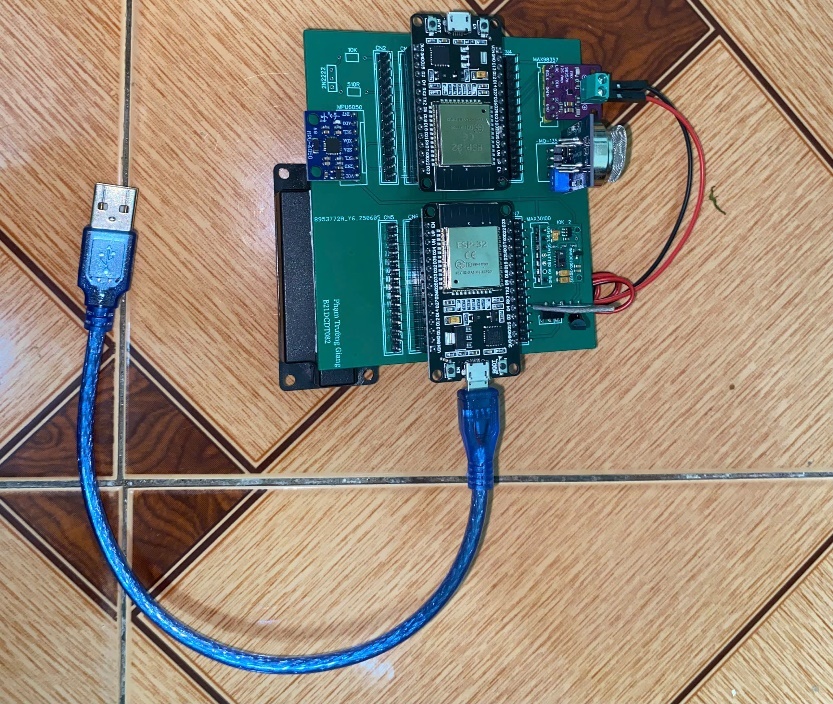
* ESP1: Gồm 2 jump đực - cái để cắm ESP32 số 1, 2 jump đực - đực để về sau có thể mở rộng kết nối.
* ESP2: Gồm 2 jump đực - cái để cắm ESP32 số 2, 2 jump đực - đực để về sau có thể mở rộng kết nối.
* ON/OFF ESP2: Sử dụng 2N2222 kết nối với ESP1 để bật tắt ESP2.
* AIR QUALITY: Sử dụng MQ135 để đọc chất lượng không khí, kết nối với chân GPIO 34 của ESP2.
* ACCELERATION: Sử dụng MPU6050 kết nối với ESP1 để huấn luyện và triển khai mô hình AI.
* SOUND: Sử dụng MAX98357 kết nối với ESP1 để phát âm thanh cảnh báo khi phát hiện ngã.
* HEAT: Sử dụng DS18B20 kết nối ESP2 để đọc nhiệt độ môi trường.
* HEART & SPO2: Sử dụng MAX30100 kết nối ESP2 để đọc nhịp tim và SPO2.

Sau khi chuyển đổi sang thiết kế PCB, sắp xếp linh kiện và kết nối các dây, ta được kết quả như hình:



Hình 3.19. Kết quả sau khi sắp xếp linh kiện, đi dây và phủ đồng PCB.

### 3.1.4. Mô hình tổng quát của hệ thống



Hình 3.20. Ảnh thực tế của mô hình hệ thống.

Sau khi hoàn thiện hai nút cảm biến riêng biệt, hệ thống được tích hợp thành một mô hình tổng thể, kết hợp giữa chức năng giám sát sức khỏe và phát hiện té ngã, tạo thành một giải pháp toàn diện nhằm bảo vệ và hỗ trợ người cao tuổi trong sinh hoạt hằng ngày. Hệ thống bao gồm:

* Nút cảm biến 1 được tích hợp mô hình trí tuệ nhân tạo huấn luyện trên Edge Impulse để phát hiện các hành vi chuyển động của người dùng, đặc biệt là ngã (FALL). Khi phát hiện té ngã với xác suất cao, nút cảm biến này sẽ:
  + Phát âm thanh cảnh báo thông qua module MAX98357.
  + Gửi cảnh báo tức thời lên nền tảng ThingsBoard để hiển thị trên Dashboard.
  + Cho phép điều khiển từ xa chân GPIO thông qua RPC.
* Nút cảm biến 2 có nhiệm vụ theo dõi các chỉ số sức khỏe và môi trường bao gồm:
  + Nhịp tim (Heart rate) và SpO₂ thông qua cảm biến MAX30100.
  + Chất lượng không khí (Air Quality) bằng cảm biến MQ135.
  + Nhiệt độ cơ thể hoặc môi trường với cảm biến DS18B20.

Cả hai nút cảm biến đều truyền dữ liệu về ThingsBoard thông qua giao thức MQTT, nơi chúng được hiển thị trực quan trên giao diện Dashboard và giám sát bằng hệ thống Rule Chain tự động, các chỉ số vượt ngưỡng sẽ kích hoạt cảnh báo WARNING hoặc CRITICAL. Dữ liệu được lưu trữ và hiển thị theo thời gian thực dưới dạng biểu đồ, bảng cảnh báo, hoặc thẻ giá trị.

Mô hình tổng thể này có thể được triển khai thực tế tại nhà người cao tuổi hoặc các cơ sở chăm sóc sức khỏe, với khả năng mở rộng thêm nhiều nút cảm biến tùy theo nhu cầu.

## 3.2. Kịch bản thử nghiệm hệ thống

### 3.2.1. Hiệu suất sử dụng mô hình

Về mức sử dụng tài nguyên phần cứng, các thiết bị đều đang hoạt động trong giới hạn an toàn và còn dư để mở rộng chức năng:

* Đối với ESP1, mức sử dụng bộ nhớ như sau:
  + RAM: Sử dụng 50,164 byte trên tổng 327,680 byte (~15.3%).
  + Flash: Sử dụng 916,213 byte trên tổng 1,310,720 byte (~69.9%).
* Đối với ESP2, mức sử dụng bộ nhớ như sau:
  + RAM: Sử dụng 45,572 byte trên tổng 327,680 byte (~13.9%).
  + Flash: Sử dụng 787,577 byte trên tổng 1,310,720 byte (~60.1%).

Các chỉ số trên cho thấy hệ thống đang sử dụng tài nguyên một cách hiệu quả, với dung lượng RAM và Flash còn trống tương đối lớn. Điều này tạo điều kiện thuận lợi để tiếp tục mở rộng thêm các chức năng mới trong tương lai mà không lo ngại vượt quá giới hạn phần cứng.

### 3.2.2. Các kịch bản kiểm thử

3.2.2.1. Kiểm tra độ chính xác của mô hình AI

Ta kiểm tra độ chính xác của từng nhãn trong mô hình:

Bảng 3.1. Kiểm tra độ chính xác của mô hình AI.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Hành động | Độ chính xác | Đánh giá |
| 1 | Đứng im (IDLE) | 95% | Độ chính xác cao |
| 2 | Ngồi xuống/Đứng lên (SU/SD) | 85% | Độ chính xác cao |
| 3 | Nằm xuống (LAY) | 90% | Độ chính xác cao |
| 4 | Đi lại (WALK) | 90% | Độ chính xác cao |
| 5 | Ngã (FALL) | 90% | Độ chính xác cao |

Kết luận: Mô hình AI nhận diện hành vi cho kết quả khả quan với độ chính xác cao ở các hành động thường ngày như đứng im, đi lại (≥90%) và đặc biệt là ngã (FALL) với độ chính xác 90%. Một số hành động tương tự như ngồi xuống hoặc đứng lên có độ chính xác thấp hơn (85%) do dễ bị nhầm lẫn. Nhìn chung, hệ thống hoạt động hiệu quả, đáng tin cậy và hoàn toàn có thể cải thiện thêm qua việc mở rộng dữ liệu huấn luyện.

3.2.2.2. Kiểm thử độ chính xác của cảm biến nhiệt độ

Ta kiểm tra độ chính xác của nhiệt độ giữa DS18B20 và cảm biến chất lượng cao khác, độ chính xác của cảm biến đọc được và giá trị hiển thị trên ThingsBoard:

Bảng 3.2. Kiểm thử độ chính xác của cảm biến nhiệt độ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Nhiệt độ DS18B20 đo được và hiển thị trên Serial Monitor (℃) | Cảm biến nhiệt độ chất lượng cao (℃) | Sai số (%) | Nhiệt độ hiển thị trên Thingsboard (℃) | Đánh giá |
| 1 | 30.8 | 31.1 | 0.97 | 31 | Đạt |
| 2 | 30.5 | 30.9 | 1.29 | 31 | Đạt |
| 3 | 30.2 | 30.4 | 0.66 | 30 | Đạt |
| 4 | 29.7 | 29.8 | 0.34 | 30 | Đạt |
| 5 | 29.8 | 30.2 | 1.32 | 30 | Đạt |
| 6 | 30.1 | 30.3 | 0.66 | 30 | Đạt |
| 7 | 30.5 | 30.7 | 0.65 | 31 | Đạt |
| 8 | 30.7 | 30.8 | 0.32 | 31 | Đạt |
| 9 | 30.2 | 30.4 | 0.66 | 30 | Đạt |
| 10 | 30.3 | 30.6 | 0.98 | 30 | Đạt |
| 11 | 31.3 | 31.6 | 0.95 | 32 | Đạt |
| 12 | 31.1 | 31.5 | 1.27 | 32 | Đạt |
| 13 | 30.9 | 31.2 | 0.96 | 31 | Đạt |
| 14 | 30.7 | 31.0 | 0.97 | 31 | Đạt |

Kết luận: Qua quá trình kiểm tra, cảm biến nhiệt độ DS18B20 cho kết quả đo khá chính xác khi so sánh với cảm biến nhiệt độ chất lượng cao, với sai số dao động từ 0.32% đến 1.32% và trung bình khoảng 0.79%. Đồng thời, giá trị nhiệt độ hiển thị trên nền tảng ThingsBoard bám sát kết quả đo thực tế do chỉ lấy phần nguyên làm tròn. Tất cả các lần đo đều nằm trong giới hạn chấp nhận được, cho thấy độ tin cậy cao của cảm biến DS18B20 cũng như tính chính xác trong quá trình thu thập và truyền dữ liệu đến hệ thống giám sát.

3.2.2.3. Kiểm thử độ chính xác của cảm biến chất lượng không khí

Ta kiểm tra độ chính xác của chất lượng không khí giữa những nơi môi trường nhiều khói bui, môi trường ít khói bụi, môi trường có khí ga và khí CO2:

Bảng 3.3. Kiểm thử độ chính xác của cảm biến chất lượng không khí.

|  |  |  |
| --- | --- | --- |
| Môi trường | Chất lượng không khí đo được (ppm) | Cảnh báo trên ThingsBoard |
| Ít bụi (Sử dụng máy lọc không khí) | 30 | không |
| Nhiều bụi (Ngoài đường chính) | 307 | Có |
| Có khí ga (Bật lửa) | >1000 | Có |
| Có khí CO2 (Đốt giấy) | 856 | Có |

Kết luận: Cảm biến MQ135 phản ánh chính xác chất lượng không khí ở các môi trường khác nhau. Trong môi trường sạch (30 ppm), hệ thống không cảnh báo; ngược lại, ở nơi ô nhiễm như ngoài đường (300 ppm), có khí gas (>1000 ppm) hoặc CO₂ (800 ppm), cảnh báo được kích hoạt trên ThingsBoard. Điều này chứng tỏ cảm biến hoạt động ổn định, nhạy và hệ thống cảnh báo hiệu quả.

3.2.2.4. Kiểm thử độ chính xác của cảm biến đo nhịp tim và SPO2

Ta kiểm tra độ chính xác của cảm biến giữa MAX30100 với đồng hồ thông minh cao cấp:

Bảng 3.4. Kiểm thử độ chính xác của cảm biến đo nhịp tim và SPO2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Nhịp tim đo từ MAX30100 | Nhịp tim đo từ đồng hồ thông minh | Sai số (%) | SPO2 đo từ MAX30100 (%) | SPO2 đo từ đồng hồ thông minh (%) | Sai số (%) |
| 1 | 66 | 70 | 5.71 | 94 | 94 | 0 |
| 2 | 68 | 75 | 9.33 | 94 | 95 | 1.05 |
| 3 | 78 | 80 | 2.5 | 95 | 95 | 0 |
| 4 | 80 | 84 | 4.76 | 97 | 94 | 3.19 |
| 5 | 78 | 75 | 4 | 94 | 95 | 1.05 |
| 6 | 84 | 82 | 2.44 | 95 | 96 | 1.04 |
| 7 | 69 | 71 | 2.82 | 94 | 94 | 0 |
| 8 | 72 | 75 | 4 | 95 | 95 | 0 |
| 9 | 75 | 72 | 4.17 | 96 | 94 | 2.13 |
| 10 | 78 | 84 | 7.14 | 95 | 96 | 1.04 |

Kết luận: Qua quá trình so sánh dữ liệu đo được từ cảm biến MAX30100 với đồng hồ thông minh cao cấp, kết quả cho thấy cảm biến hoạt động với độ chính xác tương đối tốt. Cụ thể, sai số trung bình của nhịp tim là 4.69%, nằm trong ngưỡng cho phép đối với các ứng dụng theo dõi sức khỏe không yêu cầu độ chính xác y tế cao. Trong khi đó, sai số trung bình của chỉ số SpO2 chỉ là 0.85%, thể hiện độ tin cậy cao của cảm biến trong việc đo nồng độ oxy trong máu.

3.2.2.5. Kiểm thử độ chính xác của module khuếch đại âm thanh và mô hình hệ thống khi phát hiện tình huống ngã

Ta kiểm tra độ chính xác của module khuếch đại âm thanh và mô hình hệ thống khi phát hiện tình huống ngã:

Bảng 3.5. Kiểm thử độ chính xác của module khuếch đại âm thanh và mô hình hệ thống khi phát hiện tình huống ngã.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lần thử (ngã) | Kiểu ngã | Cảnh báo trên Thingsboard | Module khuếch đại âm thanh | Đánh giá |
| 1 | Ngã về phía trước | Có | Phát âm thanh | Đạt |
| 2 | Ngã ngửa | Có | Phát âm thanh | Đạt |
| 3 | Ngã về phía trước | Có | Phát âm thanh | Đạt |
| 4 | Ngã ngửa | Có | Phát âm thanh | Đạt |

Kết luận: Hệ thống phát hiện ngã hoạt động chính xác trong cả 4 lần thử với các kiểu ngã khác nhau. Cảnh báo được gửi lên ThingsBoard và âm thanh được phát ra đầy đủ. Điều này cho thấy hệ thống ổn định, phản ứng nhanh và đáng tin cậy.

3.2.2.6. Kiểm thử độ chính xác khi nhận các cảnh báo trên ThingsBoard

Ta kiểm tra độ chức năng gửi cảnh báo khi nhiệt độ mức cao, thấp, chất lượng không khí kém hoặc phát hiện ngã:

Bảng 3.6. Kiểm thử độ chính xác khi nhận các cảnh báo trên ThingsBoard.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tình huống | Cảnh báo trên ThingsBoard | Đánh giá |
| 1 | Nhiệt độ 35 độ C | Nhiệt độ cao | Đạt |
| 2 | Nhiệt độ 31 độ C | Không cảnh báo | Đạt |
| 3 | Nhiệt độ 27 độ C | Không cảnh báo | Đạt |
| 4 | Nhiệt độ 13 độ C | Nhiệt độ thấp | Đạt |
| 5 | Ngã | Phát hiện ngã | Đạt |
| 6 | Chất lượng không khí đo được là 88 ppm | Không cảnh báo | Đạt |
| 7 | Chất lượng không khí đo được là 307 ppm | Cảnh báo không khí ô nhiễm | Đạt |

Kết luận: Hệ thống gửi cảnh báo hoạt động chính xác trong các tình huống được kiểm tra, bao gồm nhiệt độ cao, nhiệt độ thấp, chất lượng không khí kém và phát hiện ngã. Các cảnh báo được hiển thị đúng trên ThingsBoard, cho thấy hệ thống đáp ứng tốt, ổn định và phù hợp cho giám sát sức khỏe và môi trường.

### 3.2.3. Kết luận kiểm thử hệ thống

Qua các kịch bản kiểm thử, hệ thống giám sát sức khỏe và phát hiện té ngã tích hợp trí tuệ nhân tạo hoạt động ổn định và hiệu quả trên nền tảng phần cứng ESP32. Mô hình AI đạt độ chính xác trung bình từ 85–95% với các hành vi thường gặp, trong đó hành động té ngã – yếu tố quan trọng nhất – được nhận diện chính xác đến 90%, đảm bảo độ tin cậy cao trong thực tế.

Các cảm biến sử dụng trong hệ thống cũng cho kết quả khả quan:

* Cảm biến nhiệt độ DS18B20 cho sai số trung bình chỉ 0.79%,
* Cảm biến chất lượng không khí MQ135 phản ánh đúng sự thay đổi nồng độ bụi và khí độc,
* Cảm biến MAX30100 đo nhịp tim và SpO₂ với sai số lần lượt là 4.69% và 0.85%, hoàn toàn chấp nhận được trong ứng dụng theo dõi sức khỏe thông thường.

Hệ thống cảnh báo qua ThingsBoard hoạt động nhanh chóng, hiển thị chính xác theo ngưỡng cấu hình. Đặc biệt, chức năng phát âm thanh khi phát hiện ngã đã được kiểm thử thành công, cho thấy khả năng phản ứng tức thì và phù hợp với các tình huống khẩn cấp.

Tổng thể, hệ thống đạt được mục tiêu thiết kế, hoạt động ổn định, độ chính xác cao, phù hợp ứng dụng thực tế trong giám sát sức khỏe người cao tuổi. Trong tương lai, việc mở rộng tập dữ liệu huấn luyện và tối ưu thêm mô hình sẽ giúp nâng cao hiệu suất và độ tin cậy hơn nữa.

## 3.3. Kết luận và hướng phát triển trong tương lai

Hệ thống giám sát sức khỏe và phát hiện té ngã sử dụng ESP32, các cảm biến sinh học – môi trường, cùng với mô hình trí tuệ nhân tạo (AI) đã được triển khai và kiểm thử thành công. Qua các kịch bản kiểm thử, hệ thống chứng minh được tính ổn định, độ chính xác cao và khả năng phản ứng nhanh trong các tình huống khẩn cấp, đặc biệt là té ngã – một vấn đề nghiêm trọng với người cao tuổi.

Các thành phần chính như mô hình AI, cảm biến DS18B20, MQ135, MAX30100 và hệ thống cảnh báo ThingsBoard đều cho thấy hiệu suất tốt, đáp ứng đúng yêu cầu thiết kế. Hệ thống không chỉ giám sát các chỉ số sức khỏe như nhịp tim, SpO₂, nhiệt độ, mà còn theo dõi chất lượng không khí và phát hiện ngã kịp thời, cảnh báo cả bằng tín hiệu âm thanh và trực tuyến thông qua ThingsBoard.

Chúng gem có một số hướng phát triển trong tương lai như sau:

* Tăng độ chính xác của mô hình AI:
  + Thu thập thêm dữ liệu huấn luyện, đặc biệt với các hành động dễ nhầm lẫn như "Ngồi xuống/Đứng lên".
  + Cải thiện thuật toán học máy (ví dụ sử dụng mô hình nhẹ của deep learning hoặc mạng tích chập CNN).
  + Tối ưu hóa mô hình để chạy nhanh và nhẹ hơn trên vi điều khiển.
* Mở rộng chức năng giám sát:
  + Bổ sung đo huyết áp, nhịp thở hoặc ECG nếu phần cứng hỗ trợ.
  + Phát triển thuật toán phân tích xu hướng sức khỏe theo thời gian dài.
* Cải thiện giao diện và quản lý thiết bị từ xa:
  + Thiết kế giao diện ThingsBoard trực quan hơn, hỗ trợ biểu đồ theo thời gian thực.
  + Cho phép cấu hình ngưỡng cảnh báo động qua giao diện quản trị.
* Tích hợp các phương thức thông báo linh hoạt:
  + Gửi tin nhắn SMS hoặc cảnh báo qua ứng dụng điện thoại.
  + Hỗ trợ kết nối với các dịch vụ chăm sóc y tế từ xa hoặc trung tâm hỗ trợ người già.
* Tối ưu hóa năng lượng và tính di động: Tích hợp pin sạc và hệ thống quản lý năng lượng thông minh để thiết bị có thể đeo được.
* Tích hợp giao tiếp mạng qua 4G hoặc LoRa để sử dụng ở nơi không có Wi-Fi.
* Đánh giá thực tế: Triển khai thí điểm hệ thống trong môi trường thực tế như viện dưỡng lão hoặc hộ gia đình có người cao tuổi để ghi nhận phản hồi và điều chỉnh.

Tổng kết lại, hệ thống hiện tại đã đạt các tiêu chí kỹ thuật và chức năng cơ bản. Trong tương lai, việc mở rộng tính năng, nâng cao độ chính xác và ứng dụng thực tế sẽ giúp hệ thống không chỉ là một công cụ theo dõi sức khỏe, mà còn trở thành một trợ lý chăm sóc sức khỏe thông minh và đáng tin cậy cho người cao tuổi.

# TÀI LIỆU THAM KHẢO

espressif. (2025). *ESP32 Technical Reference Manual.* Shanghai: espressif.

Maxim. (2019). *MAX98357 I2S.* Maxim Integrated Products, Inc.

SEMICONDUCTOR, D. (2007). *DS18B20 Programmable Resolution1-Wire®Digital Thermometer.* DALLAS.

Winsen. (2015). *Air Quality Gas Sensor.* Zhengzhou: Zhengzhou Winsen Electronics Technology Co., Ltd.