

IoT Course

Capstone Project Final Report

Thiết kế hệ thống quản lý và điều khiển thiết bị sử dụng trợ lý giọng nói dùng ESP32-S3 và Home Assistant

Nhóm 9

Trần Nguyễn Khánh Hoàng	SIC251472
Võ Anh Trường	SIC251481
Văng Công Quỳnh	SIC251480
Phan Minh Tân	SIC251465
Hồ Minh Tuấn	SIC251422

GVHD: Trương Quang Phúc

TP. Hồ Chí Minh, ngày 4 tháng 8 năm 2025

©2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

LỜI CẢM ƠN

Trước tiên, nhóm thực hiện xin chân thành cảm ơn quý thầy cô bộ môn đã tận tình giảng dạy và cung cấp những kiến thức chuyên môn quý báu trong suốt quá trình học tập, là nền tảng vững chắc để nhóm hoàn thành đề tài này.

Nhóm xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn, Thầy Trương Quang Phúc, người đã dành nhiều thời gian hỗ trợ, định hướng, và góp ý tận tình trong suốt quá trình nghiên cứu và triển khai đề tài “Thiết kế hệ thống quản lý và điều khiển thiết bị sử dụng trợ lý giọng nói dùng ESP32-S3 và Home Assistant”. Những ý kiến chuyên môn và định hướng quý báu của Thầy đã giúp nhóm hoàn thiện tốt hơn sản phẩm của mình.

Nhóm thực hiện cũng xin cảm ơn các bạn bè đã hỗ trợ nhóm trong các khâu thử nghiệm, kiểm tra và góp ý về tính năng hệ thống.

Mặc dù nhóm đã nỗ lực hết sức để hoàn thành tốt đề tài, tuy nhiên không tránh khỏi những thiếu sót. Kính mong nhận được sự góp ý từ quý Thầy Cô để đề tài được hoàn thiện hơn.

TP. Hồ Chí Minh, ngày 4 tháng 8 năm 2025

Nhóm sinh viên thực hiện

NHÓM 9

TÓM TẮT

Báo cáo này trình bày quá trình thiết kế và triển khai một hệ thống điều khiển thiết bị thông minh, trong đó các thiết bị ngoại vi được quản lý tập trung bởi nền tảng Home Assistant OS (HAOS) chạy trên Raspberry Pi 5. Hệ thống hướng đến việc xây dựng một mô hình nhà thông minh hiện đại, nơi người dùng có thể điều khiển và giám sát các thiết bị thông qua nhiều hình thức: ứng dụng di động, giao diện web, và đặc biệt là điều khiển bằng giọng nói thông qua trợ lý ảo cục bộ.

Các thiết bị như đèn chiếu sáng, cảm biến hiện diện, và bộ thu phát tín hiệu hồng ngoại được điều khiển bởi các vi điều khiển chẳng hạn như ESP32 (bao gồm ESP32-S3 chạy voice assist), hoạt động như các node IoT độc lập và giao tiếp với Home Assistant qua Wi-Fi. Các node sử dụng firmware xây dựng bởi add-on ESPHome trên HAOS đơn giản hóa quá trình xây dựng chức năng mà không cần phải viết quá nhiều code, cho phép tích hợp nhanh chóng, linh hoạt.

Bên cạnh tính linh hoạt và khả năng mở rộng cao với các thiết bị IoT từ bên thứ ba như cảm biến môi trường hoặc camera an ninh, hệ thống đặc biệt chú trọng đến bảo mật và quyền riêng tư. Dữ liệu và quá trình xử lý giọng nói được thực hiện cục bộ (on-device/local), không phụ thuộc vào các dịch vụ đám mây bên ngoài, giúp loại bỏ rủi ro rò rỉ thông tin cá nhân và đảm bảo tính riêng tư trong quá trình vận hành. Ngoài ra, Home Assistant OS cho phép người dùng tự chủ hoàn toàn hệ thống, đồng thời hỗ trợ các giao thức mã hóa và xác thực kết nối giữa các node để đảm bảo an toàn thông tin trong mạng nội bộ.

Hệ thống sau khi triển khai cho thấy hiệu năng ổn định, độ trễ tương đối thấp, khả năng phản hồi tốt và dễ dàng mở rộng, là một giải pháp khả thi cho mô hình nhà thông minh an toàn, bảo mật và thân thiện với người dùng.

MỤC LỤC

LỜI CẢM ƠN	1
TÓM TẮT.....	2
MỤC LỤC	3
DANH MỤC HÌNH ẢNH	6
DANH MỤC BẢNG	7
CÁC TỪ VIẾT TẮT	8
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI	10
1.1. Đặt vấn đề.....	10
1.1.1. Lý do chọn đề tài.....	10
1.1.2. Tổng quan các nghiên cứu	11
1.2. Mục tiêu nghiên cứu.....	12
1.3. Phương pháp nghiên cứu.....	13
1.4. Nội dung thực hiện	13
1.5. Giới hạn đề tài	13
1.6. Bố cục báo cáo	14
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	16
2.1. Internet of Things (IoT).....	16
2.2. Giao tiếp thức truyền thông.....	16
2.2.1. Wi-Fi	16
2.2.2. I2S	17
2.2.2.1. Giới thiệu	17
2.2.2.2. Nguyên lý hoạt động	17
2.2.2.3. Ứng dụng trong đề tài.....	18
2.3. Home Assistant (HA)	18
2.2.3.1. YAML	19
2.2.3.2. Voice Assistant.....	20
2.2.3.2.1. Mô hình LLM.....	20
2.2.3.2.1.1. Giới thiệu	20

2.2.3.2.1.2. Nguyên lý hoạt động.....	21
2.2.3.2.2. Pipeline.....	22
2.2.3.3. ESP Home.....	22
2.2.3.3.1. Giới thiệu.....	22
2.2.3.3.2. Đặc điểm nổi bật	23
2.2.3.3.3. Ứng dụng trong đề tài	23
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG	24
3.1. Yêu cầu hệ thống.....	24
3.2. Đặc tả hệ thống.....	24
3.2.1. Chức năng	24
3.2.2. Đặc tính kỹ thuật.....	24
3.3. Thiết kế phần cứng	25
3.3.1. Sơ đồ tổng quát hệ thống	25
3.3.2. Trung tâm điều khiển/Máy chủ (Home Assistant Core/Sever)	26
3.3.2.1. Nhiệm vụ	26
3.3.2.2. Phần cứng sử dụng	26
3.3.3. Node nhận diện giọng nói.....	28
3.3.3.1. Nhiệm vụ	28
3.3.3.2. Phần cứng sử dụng	28
3.3.3.2.1. ESP32-S3 N16R8.....	28
3.3.3.2.2. INMP441	29
3.3.3.2.3. MAX98357A.....	30
3.3.3.2.4. Loa 3W 40hm.....	32
3.3.3.3. Sơ đồ nguyên lý Node nhận diện giọng nói	33
3.3.4 Node điều khiển thiết bị/ngoại vi.....	33
3.3.4.1 Nhiệm vụ	33
3.3.4.2. Phần cứng sử dụng	34
3.3.4.2.1. ESP32 Dev kit3	34
3.3.4.2.2. DHT11.....	35

3.3.4.2.3. IR TX/RX.....	36
3.3.4.2.4. HW-416A PIR Sensor.....	38
3.3.4.3. Sơ đồ nguyên lý Node điều khiển ngoại vi phòng ngủ	40
3.3.4.4. Sơ đồ nguyên lý Node điều khiển ngoại vi phòng khách.....	40
3.4. Thiết kế phần mềm.....	41
3.4.1. Lưu đồ giải thuật Node nhận diện giọng nói	41
3.4.2. Lưu đồ giải thuật Node điều khiển ngoại vi phòng ngủ (Bed Room)	42
CHƯƠNG 4: KẾT QUẢ THỰC HIỆN	46
4.1. Mô hình thực tế	46
4.2. Kết quả hoạt động.....	46
4.2.1. Node nhận diện giọng nói	46
4.2.2. Node điều khiển thiết bị/ngoại vi.....	48
4.3. Nhận xét và đánh giá	49
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	52
5.1. Kết luận	52
5.2. Hướng phát triển.....	53
TÀI LIỆU THAM KHẢO.....	55

DANH MỤC HÌNH ẢNH

Hình 1: Cấu trúc khung truyền của I2S	18
Hình 2: Giao diện dashboard của Home Assistant với sever local	19
Hình 3: Minh họa mã YAML	20
Hình 4: Pipeline voice assist trong Home Assistant [12]	22
Hình 5: Giao diện add-on ESPHome trong Home Assistant.....	23
Hình 6: Sơ đồ tổng quát hệ thống.....	25
Hình 7: Raspberry Pi 5	27
Hình 8: ESP32S3-N16R8	28
Hình 9: INMP441	30
Hình 10: Amp MAX98357.....	31
Hình 11: Loa 3W 4Ohm	32
Hình 12: Sơ đồ nguyên lý node nhận diện giọng nói	33
Hình 13: ESP32 Dev Kit.....	34
Hình 14: DHT11	35
Hình 15: IR RX VS1838.....	36
Hình 16: IR TX.....	37
Hình 17: HW-416A	39
Hình 18: Sơ đồ nguyên lý node điều khiển ngoại vi phòng ngủ	40
Hình 19: Sơ đồ nguyên lý node điều khiển ngoại vi phòng khách.....	40
Hình 20: Lưu đồ giải thuật Node nhận diện giọng nói.....	42
Hình 21: Lưu đồ giải thuật Node điều khiển ngoại vi phòng ngủ (Bed Room)	43
Hình 22: Lưu đồ giải thuật Node điều khiển ngoại vi phòng khách (Living Room) .	45
Hình 23: Mô hình thực tế của hệ thống	46
Hình 24: Trạng thái hỗ trợ ngôn ngữ tiếng Việt từ Open Home Foundation	50

DANH MỤC BẢNG

Bảng 1: Thông số kỹ thuật Raspberry Pi 5	27
Bảng 2: Thông số kỹ thuật vi điều khiển ESP32-S3 N16R8.....	29
Bảng 3: Thông số kỹ thuật micro INMP441	30
Bảng 4: Thông số kỹ thuật module MAX98357A	32
Bảng 5: Thông số kỹ thuật loa 3W 4Ω	33
Bảng 6: Thông số kỹ thuật bo mạch ESP32 DevKit	35
Bảng 7: Thông số kỹ thuật cảm biến DHT11	36
Bảng 8: Thông số kỹ thuật cảm biến VS1838B	37
Bảng 9: Thông số kỹ thuật đèn LED IR phát	38
Bảng 10: Thông số kỹ thuật cảm biến PIR HW-416A	39
Bảng 11: Kết quả kiểm thử chức năng node nhận diện giọng nói.....	48
Bảng 12: Kết quả kiểm thử chức năng node điều khiển thiết bị/ngoại vi	49
Bảng 13: Thống kê đánh giá tổng quát hệ thống.....	51

CÁC TỪ VIẾT TẮT

Từ viết tắt	Chú thích
ADC	Analog to Digital Converter
API	Application Programming Interface
AP	Access Point
BLE	Bluetooth Low Energy
DAC	Digital to Analog Converter
DSP	Digital Signal Processing
GPIO	General Purpose Input/Output
HA	Home Assistant
IEEE	Institute of Electrical and Electronics Engineers
I2C	Inter-Integrated Circuit
IR	Infrared
JSON	JavaScript Object Notation
LLM	Large Language Model
MQTT	Message Queuing Telemetry Transport
NLP	Natural Language Processing
OTA	Over-The-Air (update)
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
STT	Speech to Text
TSS	Text to Speech
Tx/Rx	Transmit / Receive

UART	Universal Asynchronous Receiver-Transmitter
------	---

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. Đặt vấn đề

1.1.1. Lý do chọn đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu tự động hóa và quản lý các thiết bị điện trong gia đình ngày càng trở nên phổ biến. Việc điều khiển thiết bị bằng các phương pháp truyền thống như công tắc cơ, điều khiển từ xa hay ứng dụng di động tuy vẫn hiệu quả, nhưng còn tồn tại nhiều hạn chế về tính linh hoạt, trải nghiệm người dùng và khả năng mở rộng. Trong khi đó, sự phát triển của công nghệ Internet of Things (IoT) và trí tuệ nhân tạo (AI), đặc biệt là công nghệ nhận dạng giọng nói, đã mở ra cơ hội mới trong việc xây dựng các hệ thống nhà thông minh dễ tiếp cận hơn, hiệu quả hơn và thân thiện với người dùng hơn.

Dựa trên nền tảng đó, đề tài này tập trung vào việc thiết kế và triển khai một hệ thống quản lý và điều khiển thiết bị thông minh sử dụng trợ lý giọng nói, kết hợp giữa phần cứng ESP32-S3 – một vi điều khiển mạnh mẽ tích hợp Wi-Fi và Bluetooth – cùng với nền tảng Home Assistant – phần mềm mã nguồn mở phổ biến cho hệ thống tự động hóa nhà ở. Hệ thống cho phép người dùng tương tác trực tiếp với các thiết bị điện thông qua giọng nói, đồng thời hỗ trợ quản lý trên nhiều nền tảng (máy tính, điện thoại, trình duyệt web).

Bên cạnh việc hướng đến tính tiện nghi và trải nghiệm tự nhiên trong điều khiển thiết bị, hệ thống cũng được xây dựng theo định hướng bảo mật và riêng tư dữ liệu, với khả năng mở rộng và tích hợp thêm các thiết bị IoT từ bên thứ ba như cảm biến, camera, công tắc thông minh,... Đây không chỉ là một giải pháp công nghệ hữu ích trong đời sống hàng ngày, mà còn là nền tảng nghiên cứu, học tập và phát triển các ứng dụng nhúng kết hợp AI – một xu hướng đầy tiềm năng trong tương lai. Vì vậy nhóm thực hiện quyết định chọn đề tài "**Thiết kế hệ thống quản lý và điều khiển thiết bị sử dụng trợ lý giọng nói dùng ESP32-S3 và Home Assistant**".

1.1.2. Tổng quan các nghiên cứu

Thông qua quá trình tìm hiểu và nghiên cứu, người thực hiện đã tiến hành tham khảo các đề tài, các sản phẩm và tính năng liên quan đã có từ trước:

- Nền tảng Home Assistant [1], một hệ thống mã nguồn mở phổ biến dùng để tự động hóa và điều khiển thiết bị thông minh trong gia đình. Hệ sinh thái của Home Assistant hỗ trợ nhiều giao thức như MQTT, ESPHome, Zigbee, Bluetooth, và tích hợp dễ dàng với nhiều thiết bị IoT, giúp người thực hiện thiết kế hệ thống có tính mở rộng và dễ bảo trì.
- Dự án "ESP-S3-Box Voice Assistant" của Espressif [2]: sử dụng ESP32-S3 cùng thư viện ESP-SR để nhận diện wake word và điều khiển cục bộ mà không cần kết nối Internet, là một gợi ý quan trọng cho phần xử lý giọng nói nội bộ trong dự án.
- Home Assistant Voice Preview Edition (Voice PE) [3]: là thiết bị chính thức do Home Assistant giới thiệu, chạy trên ESP32-S3, hỗ trợ wake word nội bộ, xử lý giọng nói offline hoặc kết hợp với cloud, thiết kế mở và tập trung vào bảo mật và quyền riêng tư.
- ESPHome Voice Assistant documentation [4]: Mô tả cách cấu hình các node ESP có microphone streaming âm thanh tới Home Assistant để xử lý giọng nói, với hỗ trợ wake word, chế độ push-to-talk / continuous, và các hooks automation rất mạnh
- Tài liệu "Talking with Home Assistant – get your system up & running" [5]: Hướng dẫn cấu hình Assist để sử dụng voice assistant trong Home Assistant, mô tả cách tích hợp phần cứng và cấu hình sử dụng các dịch vụ nội bộ như Whisper, Piper hoặc Cloud.
- Tài liệu kỹ thuật của các module phần cứng như INMP441 (micro I2S) [6], MAX98357A [7] (bộ khuếch đại âm thanh I2S), LED WS2812, cảm biến PIR HC-SR501 [8], và module IR phát/thu [9] được sử dụng để hoàn thiện phần cứng cho hệ thống, giúp tương tác đa chiều với người dùng.

Từ các mục tham khảo trên, người thực hiện đã tổng hợp được ý tưởng và các thành phần chức năng trong hệ thống. Hệ thống là sự kết hợp của các công nghệ hiện có đồng thời tích hợp các phương pháp điều khiển hiện đại dựa trên giọng nói, truyền thông không dây và giao diện trực quan dựa trên mô hình thực thi chức năng điều khiển giọng nói theo cấu trúc phân tán hiện đại: wake word được xử lý cục bộ ngay trên ESP32-S3, dữ liệu âm thanh được truyền vào Home Assistant để chạy STT (Whisper) và TTS (Piper), đồng thời xử lý intent thông qua Assist nội bộ hoặc kết hợp LLM (nếu cần).

1.2. Mục tiêu nghiên cứu

- Thiết kế hệ thống điều khiển và giám sát thiết bị thông minh hoạt động đa nền tảng: đảm bảo tương thích với web, ứng dụng di động, máy tính bảng và các thiết bị điều khiển từ xa khác.
- Tích hợp điều khiển giọng nói (voice assistant): cho phép người dùng thực hiện thao tác điều khiển, giám sát trạng thái thiết bị và hệ thống thông qua giao tiếp ngôn ngữ tự nhiên.
- Tích hợp chức năng nhận diện wake word cục bộ trên ESP32-S3 đảm bảo phản hồi nhanh và không cần kết nối Internet khi kích hoạt.
- Sử dụng các mô hình như Whisper và Piper để thực hiện nhận dạng giọng nói và tạo phản hồi thoại
- Kết nối và điều khiển các thiết bị đầu ra như đèn, quạt hoặc thiết bị IR thông qua các node ESPHome trong hệ thống Home Assistant dựa vào intent giọng nói.
- Hỗ trợ tùy biến và mở rộng thiết bị IoT: dễ dàng tích hợp thêm các phần tử phần cứng từ bên thứ ba như cảm biến môi trường, camera giám sát, hoặc các module điều khiển khác.
- Phát triển khả năng phản hồi theo ngữ cảnh và tự động hóa: áp dụng các kịch bản ngữ cảnh (context-based automation) nhằm tạo ra trải nghiệm điều khiển thông minh và phù hợp với nhu cầu thực tế của người dùng.

- Xây dựng hệ thống dễ cấu hình, cài đặt và bảo trì: sử dụng các nền tảng mã nguồn mở, thư viện phổ biến và giao diện cấu hình thân thiện nhằm tạo điều kiện thuận lợi cho triển khai và mở rộng hệ thống.
- Đảm bảo tính bảo mật và riêng tư: toàn bộ hệ thống hoạt động cục bộ (local), không phụ thuộc vào nền tảng đám mây, từ đó tránh rò rỉ thông tin cá nhân và tăng cường độ tin cậy.

1.3. Phương pháp nghiên cứu

- Nghiên cứu lý thuyết: Tìm hiểu qua sách, tài liệu, bài báo, diễn đàn và nguồn chính thống như Home Assistant, ESPHome để xây dựng cơ sở lý thuyết.
- Thu thập số liệu: Tổng hợp thông tin kỹ thuật từ tài liệu và thử nghiệm thực tế.
- Quan sát và thực nghiệm: Theo dõi hoạt động của hệ thống khi triển khai thực tế để đánh giá độ ổn định, độ trễ và hiệu quả điều khiển bằng giọng nói.
- Phân tích định lượng: So sánh kết quả thực nghiệm với lý thuyết, đánh giá hiệu quả vận hành.
- Phương pháp hệ thống: Thiết kế hệ thống mở rộng, dễ cấu hình và bảo trì.

1.4. Nội dung thực hiện

Các công việc được thực hiện trong đề tài gồm:

- Tìm hiểu các khía cạnh liên quan đến đề tài.
- Thu thập thông tin, tổng hợp các yêu cầu về đề tài.
- Đặc tả hệ thống.
- Thiết kế, xây dựng hệ thống (phần cứng - phần mềm).
- Kiểm thử, đánh giá kết quả.
- Hoàn thiện, viết báo cáo.
- Quay video demo sản phẩm.

1.5. Giới hạn đề tài

- Các node điều khiển các ngoại vi đơn giản như LED để mô phỏng trạng thái điều khiển bật/tắt hoặc cường độ tín hiệu xung PWM của các thiết bị khác
- Tín hiệu hồng ngoại từ điều khiển của thiết bị được nhận diện từ log của node có module nhận hồng ngoại và lưu thủ công vào phần mềm
- Sử dụng tiếng Anh trong điều khiển giọng nói
- Chưa tích hợp các thiết bị IoT khác như camera, hay các thiết bị từ bên thứ 3
- Các câu lệnh điều khiển và kiểm tra trạng thái ở mức cơ bản như bật/tắt, kiểm tra nhiệt độ/độ ẩm theo khu vực và theo vị trí cụ thể.

1.6. Bố cục báo cáo

- Chương I: **Tổng quan đề tài:** Đặt vấn đề, khái quát lý do chọn đề tài, tổng quan các nghiên cứu đã có liên quan đến đề tài, đưa ra mục tiêu và phương pháp nghiên cứu cùng với giới hạn của đề tài.
- Chương II: **Cơ sở lý thuyết:** Giới thiệu về các lý thuyết liên quan từ các ý tưởng và vấn đề đặt ra tại Chương 1 được thể hiện tại chương này giúp củng cố về lý thuyết để thuận tiện cho các công việc về sau.
- Chương III: **Thiết kế hệ thống:** Được trình bày thành hai phần thiết kế phần cứng và thiết kế phần mềm sau khi đã khái quát ý tưởng dựa trên hai chương trước. Thiết kế phần cứng thể hiện các chức năng từ đó suy ra đặc tính cùng đặc của hệ thống dựa trên các phần cứng được lựa chọn. Thiết kế phần mềm mô tả chi tiết lưu đồ giải thuật của các chức năng của hệ thống trong đề tài.
- Chương IV: **Kết quả thực hiện:** Lắp đặt các chi tiết phần cứng được lựa chọn tại Chương III và tiến hành kết hợp phần cứng với phần mềm. Quá trình tinh chỉnh, kiểm thử và đánh giá được diễn ra ngay sau đó.
- Chương V: **Kết luận và hướng phát triển:** Đưa ra kết luận từ các kết quả và đánh giá về hệ thống, nêu các nhược điểm và thiếu sót từ đó đề xuất cách khắc phục cùng hướng mở rộng đề tài trong tương lai.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Internet of Things (IoT)

IoT được xây dựng trên một nền tảng gồm bốn thành phần chính:

Thiết bị IoT (Things) là những vật thể vật lý như cảm biến, máy móc hoặc thiết bị gia dụng có khả năng thu thập dữ liệu từ môi trường xung quanh. Chúng được trang bị bộ xử lý, cảm biến, và các mô-đun truyền thông để có thể hoạt động độc lập và kết nối với mạng.

Kết nối (Connectivity) là phương thức mà các thiết bị IoT sử dụng để truyền dữ liệu. Các công nghệ kết nối phổ biến bao gồm Wi-Fi, Bluetooth, Zigbee, sóng di động (4G, 5G), và mạng LPWAN (Low-Power Wide-Area Network) như LoRaWAN. Việc lựa chọn công nghệ kết nối phụ thuộc vào khoảng cách, tốc độ truyền dữ liệu và mức tiêu thụ năng lượng cần thiết.

Nền tảng IoT (IoT Platform) là phần mềm trung gian thu thập, xử lý và quản lý dữ liệu từ các thiết bị. Nền tảng này giúp kết nối các thiết bị, lưu trữ dữ liệu trên đám mây và cung cấp các công cụ để phân tích dữ liệu, tạo ra các ứng dụng và giao diện người dùng. Nền tảng IoT đóng vai trò như một bộ não, nơi dữ liệu thô được biến thành thông tin có giá trị.

Ứng dụng (Applications) là các chương trình phần mềm được xây dựng trên nền tảng IoT để thực hiện các chức năng cụ thể. Các ứng dụng này cho phép người dùng tương tác với thiết bị, giám sát dữ liệu và đưa ra các quyết định.

2.2. Giao tiếp thức truyền thông

2.2.1. Wi-Fi

Cơ sở lý thuyết của Wifi xoay quanh một bộ tiêu chuẩn kỹ thuật được định nghĩa bởi Viện Kỹ sư Điện và Điện tử (IEEE) [10], đặc biệt là loạt tiêu chuẩn IEEE 802.11. Mỗi tiêu chuẩn trong loạt này xác định các phương pháp truyền dữ liệu khác nhau, tốc độ và băng tần:

IEEE 802.11b: Tiêu chuẩn ban đầu, hoạt động ở băng tần 2.4 GHz với tốc độ tối đa 11 Mbps.

IEEE 802.11a: Hoạt động ở băng tần 5 GHz với tốc độ tối đa 54 Mbps.

IEEE 802.11g: Cải tiến của 802.11b, hoạt động ở băng tần 2.4 GHz với tốc độ 54 Mbps.

IEEE 802.11n (Wi-Fi 4): Cải tiến đáng kể với công nghệ MIMO (Multiple-Input Multiple-Output), cho phép sử dụng nhiều anten để truyền và nhận dữ liệu đồng thời, tăng tốc độ và phạm vi. Hoạt động ở cả 2.4 GHz và 5 GHz.

IEEE 802.11ac (Wi-Fi 5): Chỉ hoạt động ở băng tần 5 GHz, hỗ trợ băng thông rộng hơn và MIMO đa người dùng (MU-MIMO), cho phép truyền dữ liệu đến nhiều thiết bị cùng lúc.

IEEE 802.11ax (Wi-Fi 6): Tiêu chuẩn mới nhất, cải thiện hiệu suất, tốc độ và hiệu quả năng lượng. Hỗ trợ băng tần 2.4 GHz, 5 GHz và 6 GHz (Wi-Fi 6E), sử dụng công nghệ OFDMA (Orthogonal Frequency-Division Multiple Access) để tối ưu hóa việc sử dụng kênh truyền.

2.2.2. I2S

2.2.2.1. Giới thiệu

I2S (Inter-IC Sound) là chuẩn giao tiếp số chuyên biệt dùng để truyền dữ liệu âm thanh giữa các thiết bị như bộ giải mã (DAC), bộ mã hóa (ADC), ampli số và vi điều khiển. I2S hỗ trợ truyền tải dữ liệu âm thanh dạng PCM (Pulse Code Modulation) với chất lượng cao, độ trễ thấp, phù hợp cho các ứng dụng audio [11].

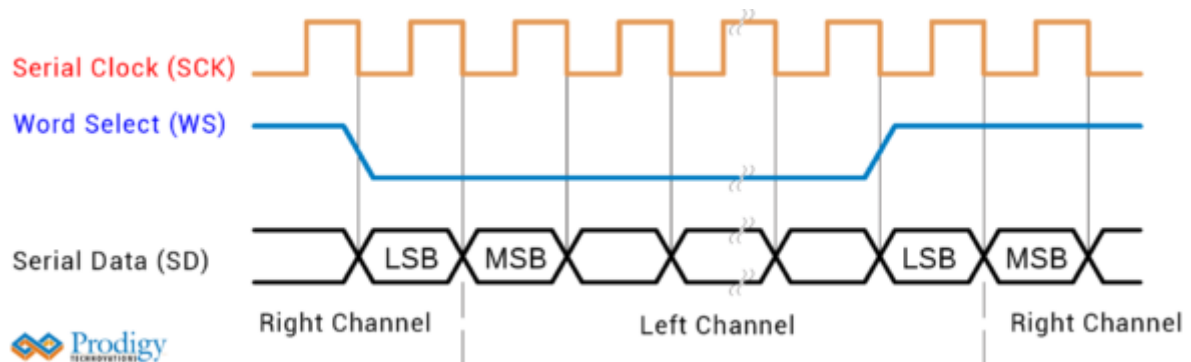
2.2.2.2. Nguyên lý hoạt động

Giao tiếp I2S gồm ba dòng tín hiệu chính:

- Bit Clock (BCLK): Tín hiệu xung clock đồng bộ bit dữ liệu.
- Word Select (WS) hay LRCLK: Xác định khung dữ liệu trái hoặc phải trong âm thanh stereo.

- Data Line (SD): Dòng dữ liệu số âm thanh truyền đi theo từng bit.

Dữ liệu được truyền tuần tự theo khung định sẵn, đồng bộ bởi các tín hiệu BCLK và WS. Thiết bị nhận sẽ giải mã dữ liệu thành tín hiệu âm thanh analog qua DAC hoặc xử lý tiếp [11].



Hình 1: Cấu trúc khung truyền của I2S

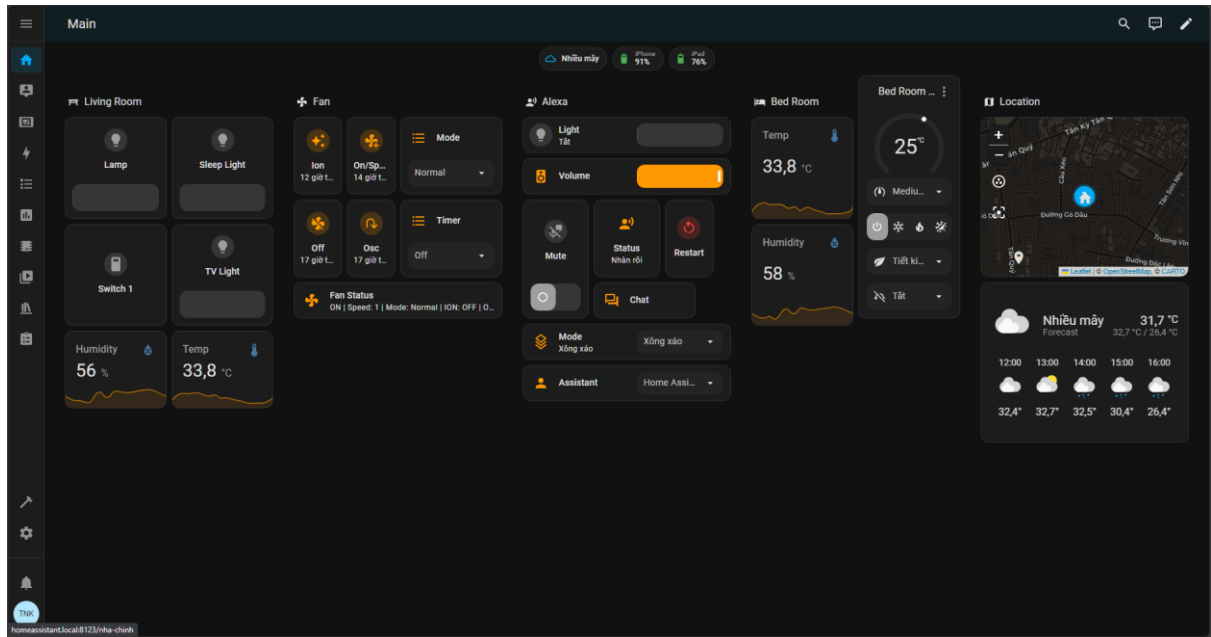
2.2.2.3. Ứng dụng trong đề tài

Trong hệ thống trợ lý giọng nói với ESP32-S3 và Home Assistant, I2S được sử dụng để:

- Thu âm thanh từ microphone I2S (ví dụ INMP441) với chất lượng cao, độ trễ thấp.
- Xuất âm thanh qua ampli số như MAX98357A, giúp phát giọng nói phản hồi rõ nét.
- I2S giúp đảm bảo tín hiệu âm thanh được truyền và xử lý hiệu quả giữa các module phần cứng trong dự án.

2.3. Home Assistant (HA)

Home Assistant (HA) là một nền tảng mã nguồn mở dùng để tự động hóa và điều khiển các thiết bị trong hệ sinh thái nhà thông minh. Được phát triển bằng ngôn ngữ Python, Home Assistant hỗ trợ hàng nghìn thiết bị và dịch vụ, cho phép người dùng thiết lập và điều khiển hệ thống IoT qua giao diện web, ứng dụng di động hoặc bằng giọng nói.



Hình 2: Giao diện dashboard của Home Assistant với sever local

Một số đặc điểm nổi bật của Home Assistant:

- Tự lưu trữ (self-hosted): Chạy cục bộ trên các thiết bị như Raspberry Pi, máy chủ mini, giúp đảm bảo quyền riêng tư và độ bảo mật cao.
- Tùy biến mạnh: Hỗ trợ YAML để cấu hình, tạo tự động hóa phức tạp và kịch bản ngữ cảnh.
- Hỗ trợ đa nền tảng: Truy cập qua trình duyệt web, app điện thoại, hoặc điều khiển bằng các trợ lý ảo như Google Assistant, Alexa.
- Khả năng tích hợp: Kết nối với nhiều thiết bị từ các hãng khác nhau như Philips Hue, Sonoff, Xiaomi, ESP32,... thông qua giao thức như MQTT, Zigbee, Z-Wave.
- Cộng đồng phát triển lớn: Hệ sinh thái plugin và hỗ trợ từ cộng đồng mạnh mẽ, giúp dễ dàng triển khai và mở rộng hệ thống.

Trong dự án này, Home Assistant đóng vai trò là trung tâm điều phối, quản lý trạng thái và xử lý yêu cầu điều khiển các thiết bị thông minh, đặc biệt là qua giao tiếp giọng nói.

2.2.3.1. YAML

YAML (YAML Ain't Markup Language) là một ngôn ngữ đánh dấu dữ liệu dạng văn bản, đơn giản và dễ đọc, thường được sử dụng để cấu hình hệ thống.

Trong Home Assistant và ESPHome, YAML được dùng để mô tả cấu hình thiết bị, các tự động hóa (automation), cảm biến, điều khiển và logic hoạt động. Các đặc điểm chính:

Cú pháp đơn giản, dễ đọc hơn so với JSON hoặc XML.

Dựa vào khoảng trắng (indentation) để thể hiện cấu trúc phân cấp.

Hỗ trợ ánh xạ (mapping), danh sách (list), và chuỗi (string).

Dễ dàng chỉnh sửa thủ công, giúp việc cấu hình hệ thống linh hoạt và nhanh chóng.

```
sensor:
  - platform: dht
    pin: GPIO23
    temperature:
      name: "Main Room Temperature"
    humidity:
      name: "Main Room Humidity"
    model: DHT11
    update_interval: 10s
```

Hình 3: Minh họa mã YAML

2.2.3.2. Voice Assistant

2.2.3.2.1. Mô hình LLM

2.2.3.2.1.1. Giới thiệu

LLM (Large Language Model) là các mô hình học sâu (deep learning) có hàng tỉ tham số, được huấn luyện trên lượng lớn dữ liệu văn bản để xử lý và sinh ngôn ngữ tự nhiên. Các mô hình như GPT (OpenAI), BERT (Google), LLaMA (Meta), hay Whisper (OpenAI, dùng cho nhận diện giọng nói) là ví dụ điển hình.

LLM có khả năng:

- Hiểu ngữ cảnh và sinh câu trả lời logic theo văn bản đầu vào
- Chuyển đổi giọng nói thành văn bản (speech-to-text) và ngược lại
- Phân tích ý định người dùng, phù hợp với các hệ thống trợ lý ảo

Trong đề tài này, mô hình LLM được sử dụng gián tiếp qua các dịch vụ tích hợp sẵn trong Home Assistant, ví dụ như:

- Whisper để chuyển giọng nói thành văn bản trên thiết bị ESP32-S3
- Piper để chuyển văn bản thành giọng nói phản hồi
- Kết hợp với các nền tảng cloud hoặc local như OpenAI API hoặc HASS Agent Voice, giúp thiết bị điều khiển thiết bị gia dụng thông minh qua ngôn ngữ tự nhiên.

2.2.3.2.1.2. Nguyên lý hoạt động

Trong hệ thống Home Assistant, các mô hình ngôn ngữ lớn (LLM) không hoạt động độc lập mà được tích hợp vào chuỗi xử lý giọng nói như sau:

- Nhận diện từ khóa đánh thức (Wake Word Detection)
- Chuyển giọng nói thành văn bản (Speech-to-Text): Sau khi đánh thức, âm thanh được thu qua microphone (ví dụ: INMP441), sau đó truyền về Home Assistant hoặc xử lý cục bộ bằng mô hình Whisper. Whisper là một LLM được huấn luyện để chuyển âm thanh giọng nói sang văn bản chính xác, hỗ trợ nhiều ngôn ngữ.
- Xử lý ngôn ngữ tự nhiên (Natural Language Processing): Văn bản được sinh ra sẽ được phân tích ý định (intent recognition). Mục tiêu là hiểu được yêu cầu người dùng.
- Thực thi hành động điều khiển: Home Assistant sẽ ánh xạ lệnh đó tới các automation, script, hoặc thực hiện hành động điều khiển các thiết bị (bật đèn, tắt quạt, v.v.) thông qua MQTT, GPIO, IR...
- Phản hồi người dùng (Text-to-Speech): Cuối cùng, nội dung phản hồi được tạo và chuyển thành giọng nói thông qua mô hình TTS như Piper, sau đó truyền về thiết bị để phát qua loa.

2.2.3.2.2. Pipeline

Khái niệm và Kiến trúc Pipeline:

Assist Pipeline là chuỗi xử lý giọng nói tự động trong Home Assistant, chuyển đổi từ âm thanh đến hành động điều khiển, gồm 4 bước chính:

- Wake Word Detection: Kích hoạt trợ lý bằng từ khóa (ví dụ: “Hey Home Assistant”).
- Speech-to-Text (STT): Chuyển giọng nói thành văn bản.
- Intent Recognition: Phân tích ý định người dùng để xác định hành động.
- Text-to-Speech (TTS): Chuyển văn bản phản hồi thành giọng nói.

Pipeline hoạt động qua WebSocket API với dữ liệu JSON, cho phép tùy chỉnh tham số âm thanh và dễ dàng tích hợp trong hệ thống.



Hình 4: Pipeline voice assist trong Home Assistant [12]

2.2.3.3. ESP Home

2.2.3.3.1. Giới thiệu

ESPHome là một framework mã nguồn mở, giúp người dùng dễ dàng lập trình và cấu hình các thiết bị dựa trên vi điều khiển ESP8266 và ESP32 thông qua file cấu

hình YAML. ESPHome tích hợp sâu với Home Assistant, cho phép điều khiển, giám sát các thiết bị IoT một cách dễ dàng và linh hoạt.

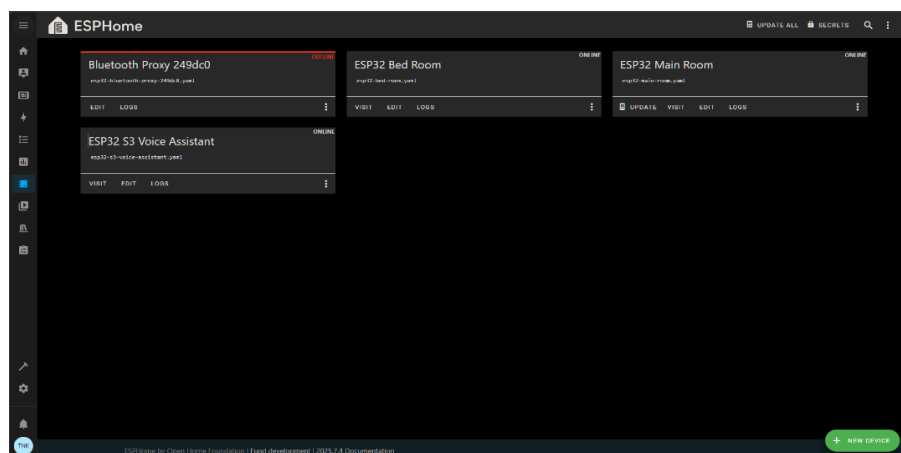
2.2.3.3.2. Đặc điểm nổi bật

- Cấu hình đơn giản: Dùng file YAML để mô tả phần cứng, cảm biến, actuators và hành vi thiết bị.
- Tích hợp đa dạng: Hỗ trợ nhiều cảm biến, thiết bị ngoại vi, giao thức truyền thông (GPIO, I2C, SPI, UART, IR, BLE...).
- Tự động tạo firmware: Từ cấu hình YAML, ESPHome biên dịch và nạp firmware trực tiếp vào thiết bị qua OTA hoặc cáp.
- Tương thích với Home Assistant: Thiết bị ESPHome tự động kết nối và hiển thị trong Home Assistant, dễ dàng tạo automation, điều khiển.
- Hỗ trợ mở rộng: Cho phép viết custom code C++ khi cần chức năng đặc biệt.

2.2.3.3.3. Ứng dụng trong đề tài

Trong dự án trợ lý giọng nói và điều khiển thiết bị thông minh:

- ESPHome được sử dụng để lập trình ESP32-S3, cấu hình các cảm biến âm thanh (micro I2S INMP441), ampli MAX98357A, LED trạng thái và nút nhấn.
- ESPHome hỗ trợ các tính năng như wake word detection, speech-to-text và điều khiển thiết bị qua MQTT hoặc trực tiếp.
- ESPHome giúp tối ưu quá trình phát triển, dễ cập nhật, và tích hợp liền mạch với Home Assistant.



Hình 5: Giao diện add-on ESPHome trong Home Assistant

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1. Yêu cầu hệ thống

Hệ thống được thiết kế để điều khiển thiết bị thông minh bằng giọng nói, kết hợp giữa phần cứng và phần mềm. Về phần cứng, hệ thống sử dụng ESP32-S3 làm trung tâm, tích hợp kết nối Wi-Fi để giao tiếp với microphone I2S (INMP441) và các ngoại vi điều khiển thiết bị. Về phần mềm, nền tảng Home Assistant đóng vai trò là máy chủ, xử lý lệnh giọng nói và điều phối các lệnh điều khiển. Giao thức I2S, Wi-Fi đảm bảo hệ thống hoạt động hiệu quả, độ trễ thấp và có khả năng mở rộng.

3.2. Đặc tả hệ thống

3.2.1. Chức năng

Đánh thức bằng từ khóa 'alexa'.

Điều khiển bằng giọng nói: Hệ thống nhận lệnh từ giọng nói của người dùng thông qua ESP32-S3, sau đó chuyển đến Home Assistant để phân tích và thực thi.

Phản hồi bằng âm thanh: Hệ thống phản hồi bằng giọng nói để xác nhận lệnh đã được thực hiện.

Điều khiển thủ công: Người dùng có thể điều khiển thiết bị từ xa qua giao diện của Home Assistant.

Theo dõi nhiệt độ, độ ẩm, trạng thái thời tiết trong khu vực thiết đặt trước.

Điều khiển thiết bị không thông minh/thông minh thông qua tín hiệu hồng ngoại.

Quản lý thiết bị: Cho phép thêm, xóa và giám sát trạng thái của các thiết bị một cách dễ dàng.

Cấu hình thiết bị thông qua ESP Home và Home Assistant.

Thiết lập các kịch bản, tự động hóa cho hệ thống.

3.2.2. Đặc tính kỹ thuật

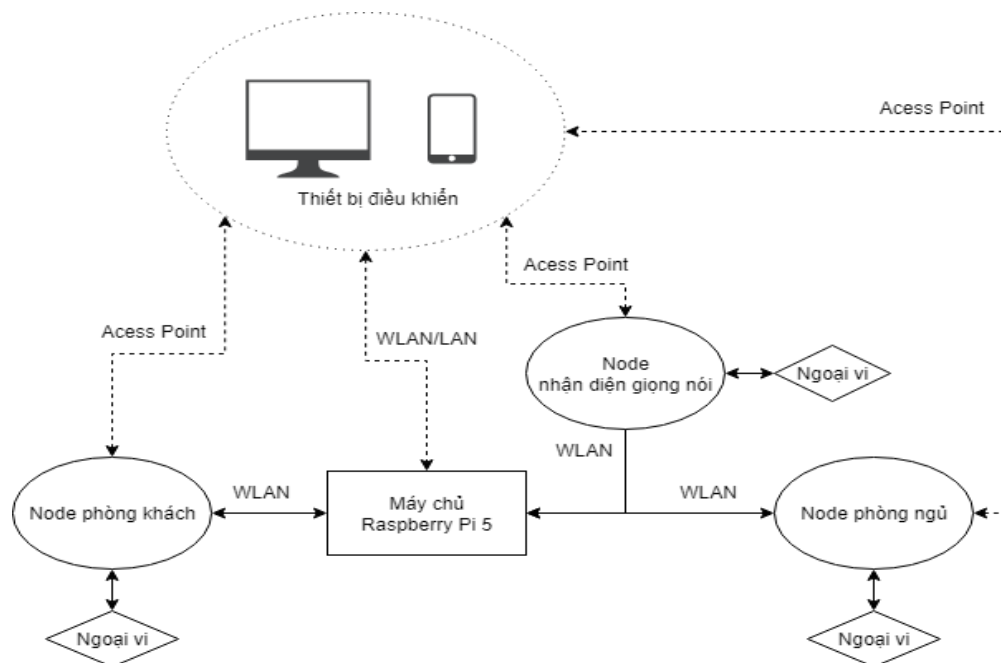
- Thời gian nhận diện đánh thức: <500ms

- Thời gian phản hồi câu lệnh: <10s
- Bảo mật wifi sử dụng WPA2
- Quản lý quyền truy cập thông qua Home Assistant
- Thời gian đồng bộ thông tin giữa các thiết bị: <2s
- Thời gian phản hồi điều khiển từ hệ thống: <1s
- Sai số về cảm biến nhiệt độ, độ ẩm: $\pm 5\%$
- Nguồn điện sử dụng : 5V 1A đối với các node điều khiển thiết bị và 5V 2A đối với node nhận diện giọng nói (có thể cao hơn nếu dùng loa công suất lớn và chip xử lý âm thanh riêng).

3.3. Thiết kế phần cứng

3.3.1. Sơ đồ tổng quát hệ thống

Các yêu cầu và đặc tả hệ thống được tổng hợp lại và sơ đồ khối hóa để thể hiện các liên kết giữa các khối chức năng và bước đầu hình thành thiết kế. Hình bên dưới là sơ đồ khối của toàn hệ thống.



Hình 6: Sơ đồ tổng quát hệ thống

3.3.2. Trung tâm điều khiển/Máy chủ (Home Assistant Core/Sever)

3.3.2.1. Nhiệm vụ

Home Assistant Core/Server là bộ não trung tâm của toàn bộ hệ thống. Nhiệm vụ chính của nó là tổng hợp, xử lý và điều phối mọi hoạt động.

Xử lý giọng nói: Tiếp nhận dữ liệu âm thanh từ ESP32-S3, sử dụng công cụ NLP để chuyển đổi giọng nói thành văn bản và phân tích ý định của người dùng.

Điều phối lệnh: Dựa trên ý định đã phân tích, tạo ra các lệnh điều khiển và gửi đến các Node điều khiển thiết bị ngoại vi.

Quản lý và tự động hóa: Chịu trách nhiệm quản lý trạng thái của tất cả thiết bị, cung cấp giao diện điều khiển thủ công, và thiết lập các tự động hóa phức tạp.

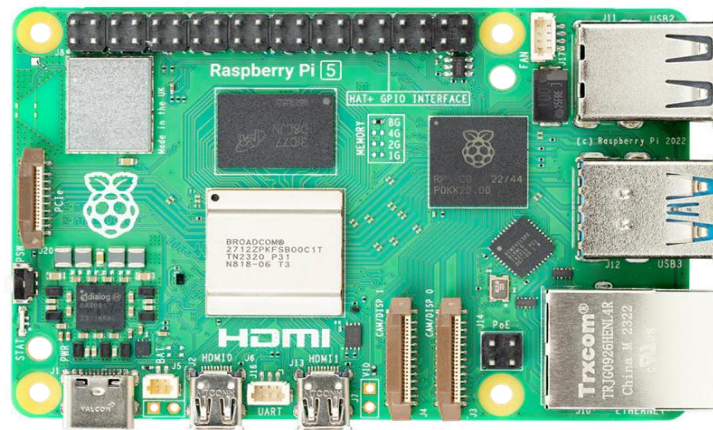
Tích hợp: Đóng vai trò là điểm tích hợp với các dịch vụ và thiết bị thông minh khác, đảm bảo hệ thống hoạt động mạch lạc và có khả năng mở rộng.

3.3.2.2. Phần cứng sử dụng

Raspberry Pi 5

Hoạt động như bộ não trung tâm của hệ thống. Nó tiếp nhận dữ liệu giọng nói từ các node, sử dụng các công cụ xử lý ngôn ngữ tự nhiên (NLP) để chuyển đổi thành văn bản, phân tích ý định của người dùng và tạo ra các lệnh điều khiển. Ngoài ra, nó còn quản lý trạng thái thiết bị, cung cấp giao diện người dùng và thiết lập các quy tắc tự động hóa.

Phần cứng sử dụng (Raspberry Pi 5): Raspberry Pi 5 được chọn làm phần cứng cho máy chủ nhờ hiệu năng mạnh mẽ. Nó là nền tảng đáng tin cậy để chạy Home Assistant Core, xử lý dữ liệu giọng nói và điều phối các tín hiệu điều khiển, biến nó thành một trung tâm điều khiển hiệu quả và có khả năng mở rộng.



Hình 7: Raspberry Pi 5

Bảng thông số kỹ thuật Raspberry Pi 5:

Thông số	Giá trị
Bộ xử lý	Broadcom BCM2712, Quad-core ARM Cortex-A76 2.0 GHz
Bộ nhớ RAM	4GB hoặc 8GB LPDDR4x
Đồ họa GPU	VideoCore VII GPU, hỗ trợ OpenGL ES 3.1
Lưu trữ	Hỗ trợ thẻ microSD, USB 3.0 SSD
Cổng kết nối	2 x USB 3.0, 2 x USB 2.0, Gigabit Ethernet (2.5Gbps), HDMI 2.1
Hỗ trợ kết nối không dây	Wi-Fi 6, Bluetooth 5.0
Hệ điều hành	Raspberry Pi OS hoặc các bản Linux tương thích
Kích thước	85mm x 56mm

Bảng 1: Thông số kỹ thuật Raspberry Pi 5

3.3.3. Node nhận diện giọng nói

3.3.3.1. Nhiệm vụ

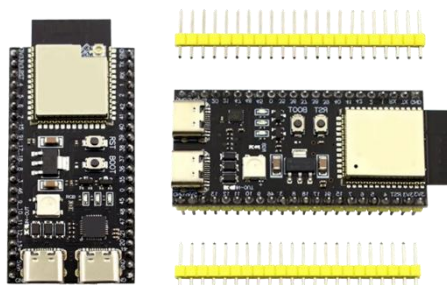
Đây là điểm tương tác đầu tiên với người dùng, được hiện thực hóa bằng ESP32-S3 và microphone I2S. Nhiệm vụ chính của nó là thu nhận tín hiệu âm thanh analog, chuyển đổi thành dữ liệu số chất lượng cao, thực hiện các bước tiền xử lý như lọc nhiễu, sau đó truyền dữ liệu đã xử lý qua Wi-Fi đến Home Assistant. Node này đảm bảo chất lượng tín hiệu giọng nói, là yếu tố tiên quyết cho việc nhận diện chính xác ở bước sau.

3.3.3.2. Phân cứng sử dụng

3.3.3.2.1. ESP32-S3 N16R8

ESP32-S3 N16R8 là phiên bản nâng cấp của dòng vi điều khiển ESP32, được thiết kế đặc biệt để hỗ trợ các ứng dụng trí tuệ nhân tạo (AI) và xử lý tín hiệu số (DSP). ESP32-S3 tích hợp lõi kép Xtensa LX7 với tốc độ lên đến 240 MHz, đi kèm khả năng Wi-Fi và Bluetooth 5.0, đáp ứng các yêu cầu về kết nối không dây và xử lý đa nhiệm trong các hệ thống IoT và trợ lý giọng nói.

Trong đề tài này, ESP32-S3 N16R8 là bộ xử lý trung tâm, đảm nhiệm việc thu nhận dữ liệu âm thanh từ micro INMP441 qua giao tiếp I2S, xử lý nhận diện từ khóa (wake word), chuyển đổi giọng nói sang văn bản, điều khiển các thiết bị ngoại vi và giao tiếp với Home Assistant qua Wi-Fi. Hình ảnh thực tế bo mạch được thể hiện tại hình dưới.



Hình 8: ESP32S3-N16R8

Bảng thông số kỹ thuật vi điều khiển ESP32-S3 N16R8:

Thông số	Giá trị
Lõi xử lý	2 x Xtensa® 32-bit LX7
Tốc độ xung nhịp	Tối đa 240 MHz
Bộ nhớ SRAM	512 KB
Bộ nhớ Flash	16 MB (tuỳ loại)
Kết nối không dây	Wi-Fi 802.11 b/g/n, Bluetooth 5.0
Hỗ trợ AI & DSP	Tích hợp Vector Extensions cho AI/DSP
Giao tiếp	GPIO, I2C, SPI, UART, I2S, ADC, DAC
Điện áp hoạt động	3.0V – 3.6V
Kích thước chip	Khoảng 7 mm x 7 mm

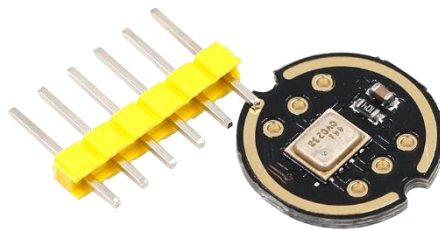
Bảng 2: Thông số kỹ thuật vi điều khiển ESP32-S3 N16R8

3.3.3.2.2. INMP441

INMP441 là một microphone kỹ thuật số dạng MEMS (Micro-Electro-Mechanical System) có giao tiếp I²S, được thiết kế để thu âm thanh chất lượng cao với tiếng ồn thấp và độ trễ nhỏ. Micro này tích hợp sẵn bộ chuyển đổi analog sang số (ADC), cho phép truyền dữ liệu âm thanh kỹ thuật số trực tiếp qua giao tiếp I²S, giảm thiểu nhiễu và đơn giản hoá mạch thu âm.

Trong đề tài này, INMP441 đóng vai trò thu nhận giọng nói người dùng, sau đó gửi tín hiệu âm thanh đến bộ vi điều khiển ESP32-S3 để xử lý wake word và thực hiện các lệnh điều khiển thiết bị.

Hình ảnh thực tế được thể hiện tại hình dưới.



Hình 9: INMP441

Bảng thông số kỹ thuật micro INMP441:

Thông số	Giá trị
Loại micro	MEMS kỹ thuật số
Giao tiếp	I ² S
Điện áp hoạt động	1.8V – 3.3V
Dải tần đáp ứng	60Hz – 15kHz
Độ nhạy	–26 dBFS (typical)
SNR (Tỷ số tín hiệu/nhiều)	61 dBA (typical)
Kích thước micro	~3.5mm × 4.0mm
Hướng thu	Omnidirectional (đa hướng)
Dòng tiêu thụ	< 1.4 mA (typical)
Giao tiếp data	WS (Word Select), SCK, SD

Bảng 3: Thông số kỹ thuật micro INMP441

3.3.3.2.3. MAX98357A

MAX98357A là mạch khuếch đại âm thanh Class-D tích hợp bộ giải mã tín hiệu âm thanh số I2S, có khả năng phát tín hiệu âm thanh ra loa với chất lượng cao và

hiệu suất năng lượng tốt. Module này rất phổ biến trong các ứng dụng trợ lý ảo, phát nhạc, hệ thống phản hồi âm thanh,...

Trong đề tài, MAX98357A được sử dụng để nhận dữ liệu âm thanh từ vi điều khiển ESP32-S3 thông qua giao tiếp I2S và khuếch đại tín hiệu đó để phát ra loa. Đây là thành phần quan trọng giúp hệ thống có khả năng phản hồi người dùng bằng giọng nói. Hình ảnh thực tế được thể hiện tại hình dưới.



Hình 10: Amp MAX98357

Bảng thông số kỹ thuật module MAX98357A:

Thông số	Giá trị
Điện áp hoạt động	2.5V – 5.5V
Giao tiếp dữ liệu	I2S
Công suất đầu ra	3.2W @ 4Ω, VDD = 5V
Tỉ lệ tín hiệu / nhiễu (SNR)	91dB
Kênh âm thanh	Mono (có thể chọn L/R)
Trở kháng ngõ ra loa	4Ω – 8Ω
Hiệu suất khuếch đại	Lên đến 92%
Tích hợp bộ lọc chống nhiễu	Có (pop/click suppression)

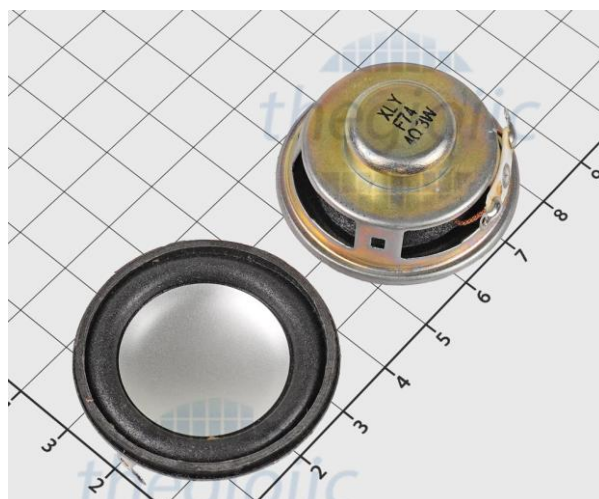
Kích thước module	18mm x 15mm x 3mm (tùy phiên bản)
-------------------	-----------------------------------

Bảng 4: Thông số kỹ thuật module MAX98357A

3.3.3.2.4. Loa 3W 4Ohm

Loa 3W 4Ω là loại loa nhỏ gọn, thường được sử dụng trong các thiết bị nhúng và ứng dụng âm thanh nhúng như máy trợ lý ảo, loa thông minh, hệ thống cảnh báo,... Với công suất 3W và trở kháng 4Ω, loa phù hợp với các mạch khuếch đại âm thanh số công suất thấp như MAX98357A.

Trong đề tài này, loa 3W 4Ω được kết nối trực tiếp với bộ khuếch đại MAX98357A để phát ra âm thanh phản hồi từ hệ thống trợ lý giọng nói. Đây là phần tử đầu ra giúp người dùng nhận được kết quả dưới dạng âm thanh. Hình ảnh thực tế được thể hiện tại dưới.



Hình 11: Loa 3W 4Ohm

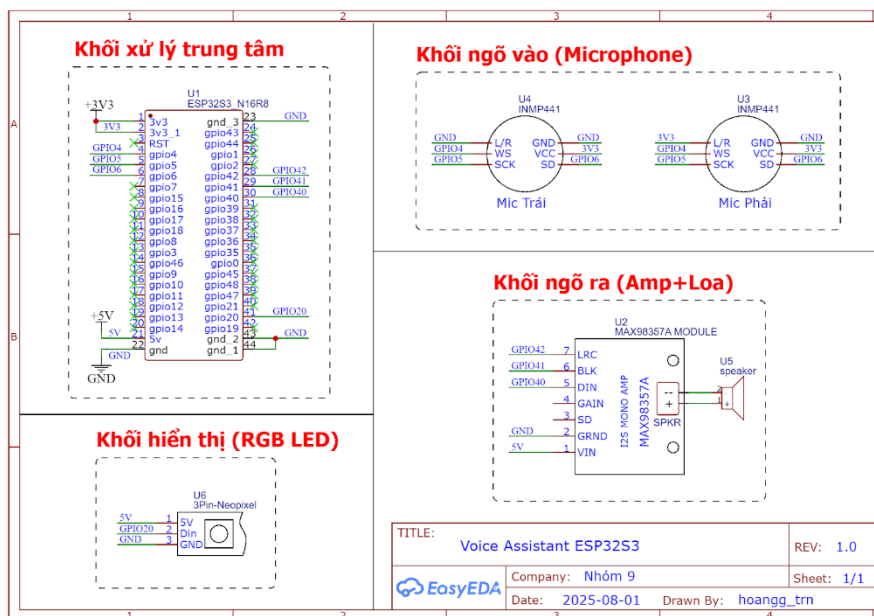
Bảng thông số kỹ thuật loa 3W 4Ω:

Thông số	Giá trị
Công suất định mức	3W
Trở kháng	4Ω

Tần số đáp ứng	150Hz – 20kHz (tùy loại)
Đường kính loa	40mm (hoặc 27mm, tùy phiên bản)
Loại loa	Full-range
Độ nhạy	~88dB (tùy phiên bản)
Trọng lượng	Khoảng 20 – 50g (tùy loại)

Bảng 5: Thông số kỹ thuật loa 3W 4Ω

3.3.3.3. Sơ đồ nguyên lý Node nhận điện giọng nói



Hình 12: Sơ đồ nguyên lý node nhận điện giọng nói

3.3.4 Node điều khiển thiết bị/ngoại vi

3.3.4.1 Nhiệm vụ

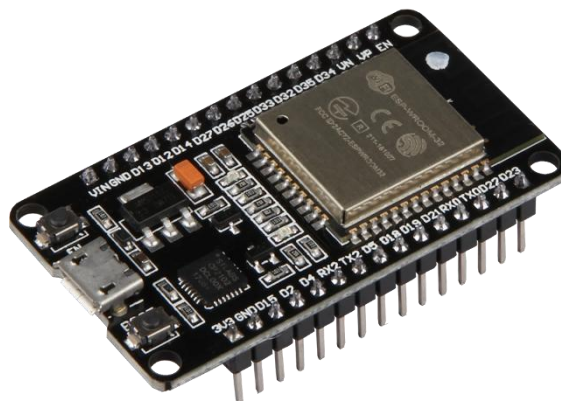
Node điều khiển thiết bị ngoại vi đóng vai trò là cầu nối vật lý, biến các lệnh điều khiển thành hành động thực tế. Nhiệm vụ chính của node này là tiếp nhận và thực thi các lệnh từ Home Assistant. Cụ thể, nó giải mã tín hiệu và kích hoạt các module rơ-le để bật/tắt hoặc điều chỉnh các thiết bị điện như đèn và quạt. Node này đảm bảo rằng mọi lệnh đều được thực thi một cách chính xác và đáng tin cậy.

3.3.4.2. Phần cứng sử dụng

3.3.4.2.1. ESP32 Dev kit3

ESP32 DevKit là bo mạch phát triển dựa trên vi điều khiển ESP32 của Espressif, tích hợp Wi-Fi và Bluetooth, hỗ trợ đa dạng giao tiếp như GPIO, I2C, SPI, UART, PWM,... Bo mạch có khả năng xử lý mạnh mẽ với hai nhân xử lý (dual-core), phù hợp cho các ứng dụng IoT, điều khiển thiết bị, thu phát tín hiệu không dây,...

Trong đề tài này, ESP32 DevKit đóng vai trò trung tâm xử lý của hệ thống. Nó thu thập dữ liệu từ các cảm biến (IR, PIR, DHT11,...), xử lý giọng nói, điều khiển thiết bị, giao tiếp với Home Assistant qua Wi-Fi, và thực hiện các phản hồi thông minh. Hình ảnh thực tế của bo mạch thể hiện tại hình dưới.



Hình 13: ESP32 Dev Kit

Bảng thông số kỹ thuật bo mạch ESP32 DevKit:

Thông số	Giá trị
Vi điều khiển	ESP32-D0WDQ6 (tùy phiên bản)
Số nhân xử lý	2 nhân Xtensa® 32-bit LX6
Tốc độ xung nhịp	Lên đến 240MHz

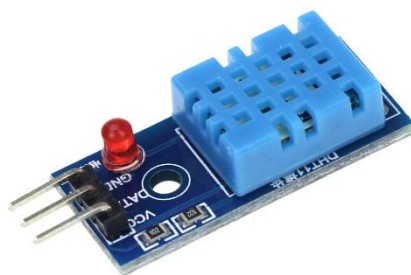
Bộ nhớ SRAM	520KB
Bộ nhớ flash	4MB (hoặc tùy loại 8MB, 16MB...)
Kết nối không dây	Wi-Fi 802.11 b/g/n, Bluetooth 4.2 BLE
Giao tiếp I/O	GPIO, I2C, SPI, UART, PWM, ADC, DAC
Điện áp hoạt động	3.0V – 3.3V (qua USB: 5V)
Cổng giao tiếp lập trình	USB (chuyển UART)
Kích thước	Khoảng 52mm x 27mm x 13mm

Bảng 6: Thông số kỹ thuật bo mạch ESP32 DevKit

3.3.4.2.2. DHT11

DHT11 là cảm biến kỹ thuật số tích hợp dùng để đo nhiệt độ và độ ẩm không khí. Cảm biến sử dụng đầu ra dạng tín hiệu số, giao tiếp đơn giản qua một chân tín hiệu, phù hợp cho các ứng dụng giám sát môi trường, nhà thông minh, hệ thống tự động hóa,...

Trong đề tài này, DHT11 được sử dụng để theo dõi nhiệt độ và độ ẩm trong khu vực lắp đặt, từ đó hỗ trợ các chức năng điều khiển thiết bị phù hợp với điều kiện môi trường (ví dụ: bật quạt khi nóng, cảnh báo khi ẩm thấp,...). Hình ảnh thực tế cảm biến được thể hiện tại hình dưới.



Hình 14: DHT11

Bảng thông số kỹ thuật cảm biến DHT11:

Thông số	Giá trị
Điện áp hoạt động	3.3V – 5.5V
Dòng tiêu thụ	0.3mA (trung bình), 2.5mA (max)
Dải đo nhiệt độ	0°C đến 50°C
Sai số nhiệt độ	±2°C
Dải đo độ ẩm	20% – 90% RH
Sai số độ ẩm	±5% RH
Tần suất đo	1Hz (1 lần/giây)
Giao tiếp	Digital (1 dây dữ liệu)
Kích thước	15.5mm x 12mm x 5.5mm

Bảng 7: Thông số kỹ thuật cảm biến DHT11

3.3.4.2.3. IR TX/RX

VS1838B là module thu tín hiệu hồng ngoại hoạt động ở tần số 38kHz, thường dùng để nhận tín hiệu từ remote IR. Cảm biến có khả năng lọc nhiễu tốt, phạm vi thu rộng, phù hợp trong các ứng dụng điều khiển thiết bị từ xa bằng sóng hồng ngoại.

Trong đề tài này, cảm biến VS1838B được sử dụng để thu tín hiệu từ remote



Hình 15: IR RX VS1838

điều khiển, cho phép hệ thống học và phản hồi lại mã lệnh IR tương ứng. Hình ảnh thực tế được thể hiện tại hình dưới.

Bảng thông số kỹ thuật cảm biến VS1838B:

Thông số	Giá trị
Điện áp hoạt động	2.7V – 5.5V
Dòng tiêu thụ	0.4 – 1.5mA
Tần số hoạt động	38kHz
Góc thu tín hiệu	$\pm 45^\circ$
Khoảng cách thu	Lên đến 18 mét (tùy môi trường)
Giao tiếp	Digital (1 chân OUT)
Tín hiệu ngõ ra	Mức thấp (LOW) khi nhận tín hiệu
Kích thước	5 x 7 x 5mm

Bảng 8: Thông số kỹ thuật cảm biến VS1838B

IR LED là loại diode phát ánh sáng hồng ngoại không nhìn thấy được bằng mắt thường, sử dụng phổ biến trong các ứng dụng truyền dữ liệu điều khiển từ xa. Đèn hồng ngoại có thể được điều khiển bật/tắt theo mã cụ thể để gửi lệnh tới các thiết bị như TV, quạt, điều hòa,...

Trong đề tài, IR LED TX được sử dụng kết hợp với ESP32 để phát lại các mã lệnh đã học, phục vụ chức năng điều khiển từ xa. Hình ảnh thực tế thể hiện tại hình dưới.



Hình 16: IR TX

Bảng thông số kỹ thuật đèn LED IR phát:

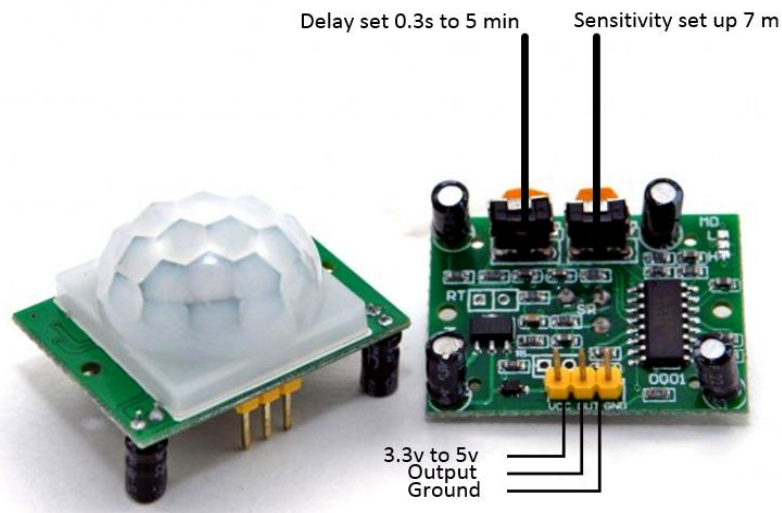
Thông số	Giá trị
Điện áp hoạt động	1.2V – 1.5V
Dòng phát tối đa	50 – 100mA
Bước sóng phát	940nm
Góc phát	20° – 60° (tùy loại)
Tần số điều chế phù hợp	38kHz – 56kHz
Khoảng cách phát	5 – 10 mét (tùy dòng và tản nhiệt)
Loại LED	IR LED 5mm hoặc 3mm
Kích thước	Ø5mm x 8.7mm

Bảng 9: Thông số kỹ thuật đèn LED IR phát

3.3.4.2.4. HW-416A PIR Sensor

Cảm biến HW-416A sử dụng công nghệ hồng ngoại thụ động (PIR – Passive Infrared) để phát hiện chuyển động của người hoặc vật thể phát ra bức xạ nhiệt trong phạm vi quan sát. Cảm biến thường được sử dụng trong các hệ thống báo động, điều khiển đèn tự động, tiết kiệm năng lượng,...

Trong đề tài này, cảm biến HW-416A đóng vai trò phát hiện sự hiện diện của người trong phòng, từ đó tự động bật hoặc tắt các thiết bị như đèn hoặc kích hoạt hệ thống trợ lý giọng nói. Hình ảnh thực tế của module được thể hiện tại hình dưới.



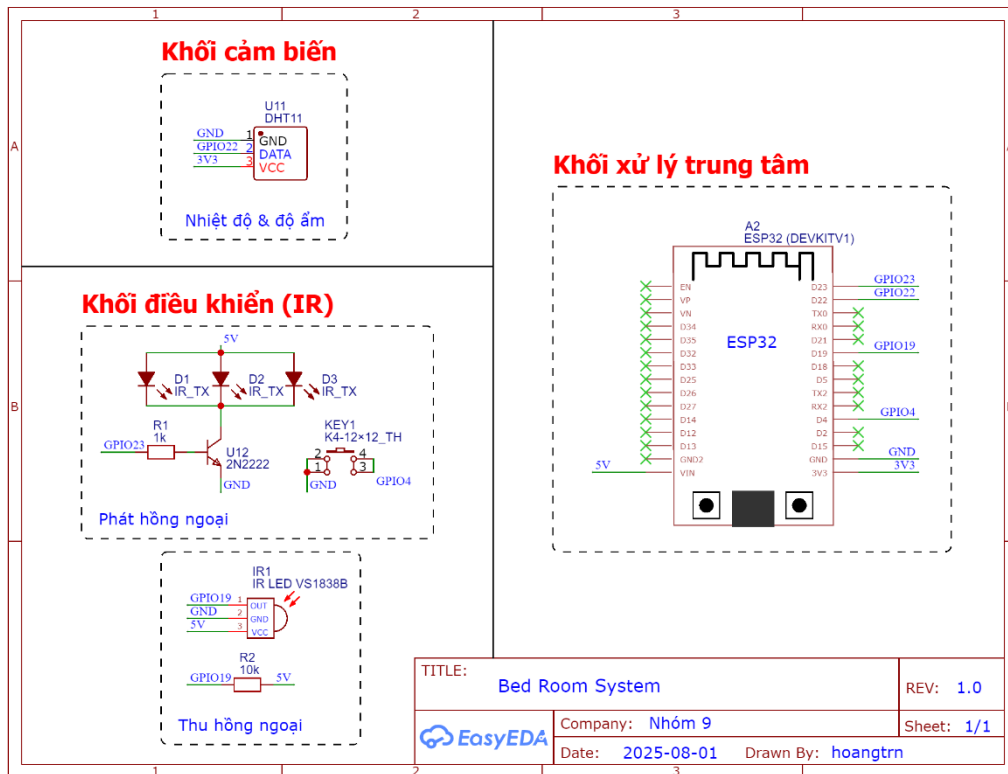
Hình 17: HW-416A

Bảng thông số kỹ thuật cảm biến PIR HW-416A:

Thông số	Giá trị
Điện áp hoạt động	5VDC
Dòng tiêu thụ	< 60 μ A
Khoảng cách phát hiện	3 – 7 mét
Góc phát hiện	110 độ
Thời gian trễ điều chỉnh	5 – 200 giây (có biến trở)
Mức tín hiệu ngõ ra	Cao (HIGH) khi phát hiện
Giao tiếp	Digital (1 chân OUT)
Nhiệt độ hoạt động	-20°C ~ +50°C
Kích thước module	32mm x 24mm x 18mm

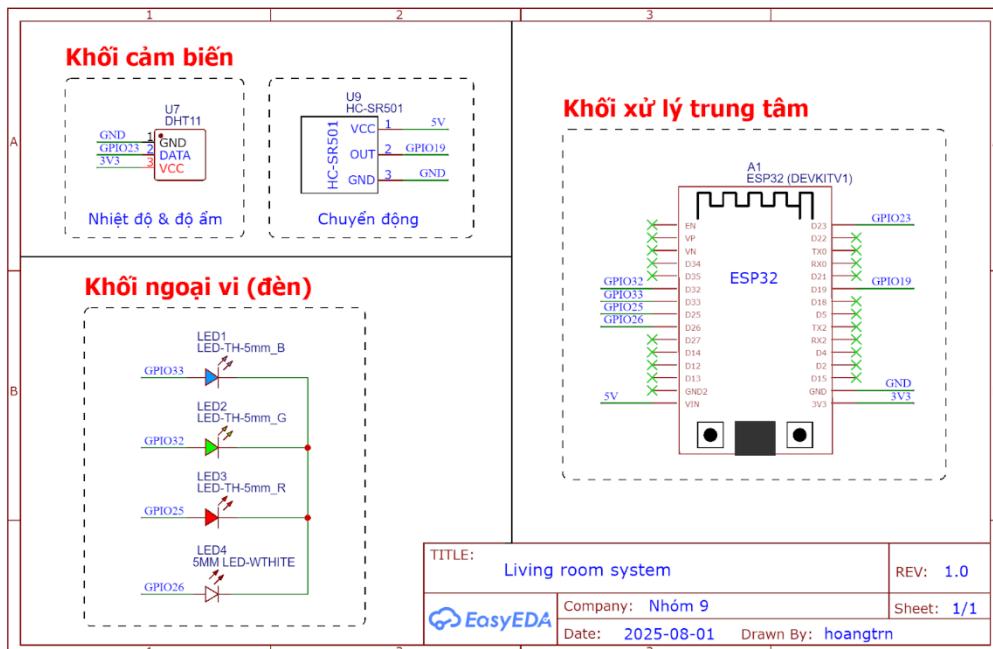
Bảng 10: Thông số kỹ thuật cảm biến PIR HW-416A

3.3.4.3. Sơ đồ nguyên lý Node điều khiển ngoại vi phòng ngủ



Hình 18: Sơ đồ nguyên lý node điều khiển ngoại vi phòng ngủ

3.3.4.4. Sơ đồ nguyên lý Node điều khiển ngoại vi phòng khách



Hình 19: Sơ đồ nguyên lý node điều khiển ngoại vi phòng khách

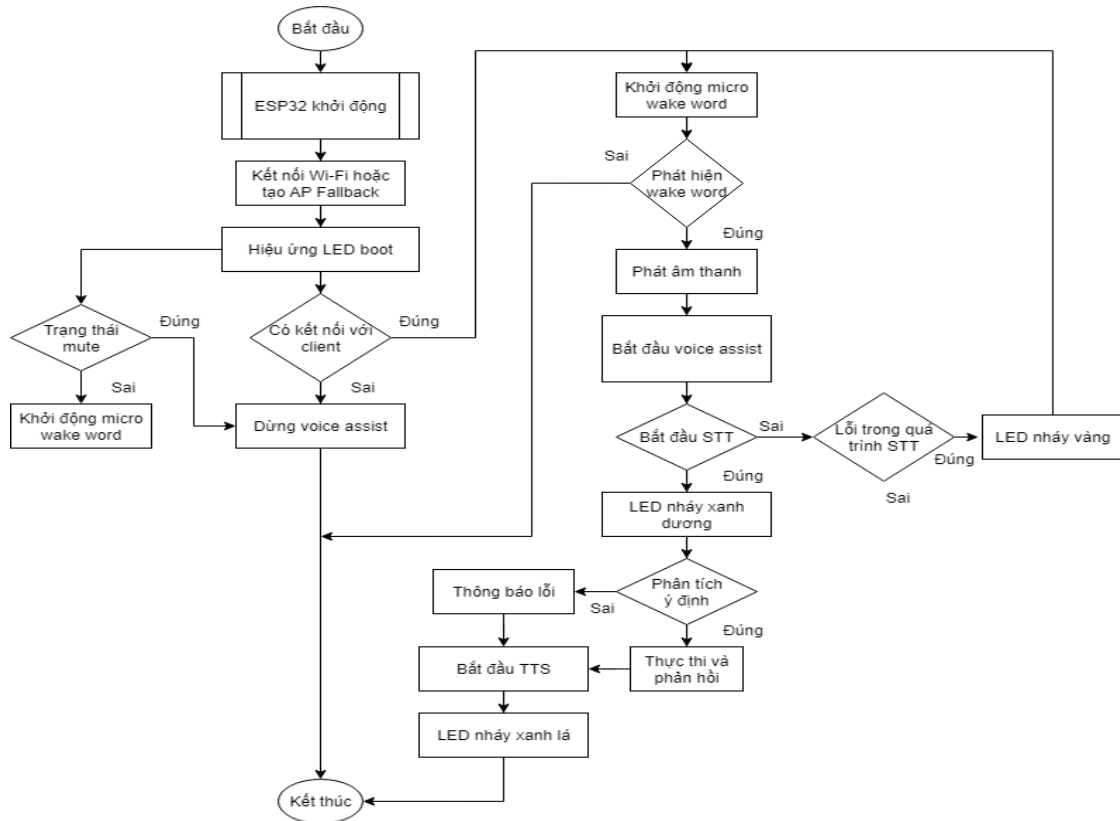
3.4. Thiết kế phần mềm

3.4.1. Lưu đồ giải thuật Node nhận diện giọng nói

Lưu đồ mô tả quy trình hoạt động của Node nhận diện giọng nói trên nền tảng ESP32 trong hệ thống trợ lý giọng nói như sau:

- Khởi động hệ thống: Node ESP32 được khởi động, tiến hành kết nối với mạng Wi-Fi hoặc tạo điểm truy cập dự phòng (AP Fallback) nếu không có mạng. Trong quá trình này, hiệu ứng LED boot được kích hoạt để báo trạng thái khởi động.
- Kiểm tra trạng thái mute: Nếu thiết bị đang ở trạng thái tắt tiếng (mute), hệ thống sẽ khởi động micro để nhận diện wake word (từ khóa đánh thức). Nếu không, hệ thống sẽ dừng chức năng voice assist.
- Phát hiện wake word: Micro liên tục hoạt động để phát hiện từ khóa đánh thức. Khi phát hiện wake word, hệ thống phát âm thanh thông báo kích hoạt và bắt đầu quy trình trợ lý giọng nói (voice assist).
- Chuyển giọng nói thành văn bản (STT): Sau khi khởi động voice assist, hệ thống bắt đầu quá trình chuyển giọng nói thành văn bản (Speech-to-Text - STT). Nếu quá trình này thành công, LED trạng thái sẽ nhấp xanh dương để báo hiệu. Nếu có lỗi trong quá trình này, LED sẽ nhấp vàng báo lỗi.
- Phân tích ý định và thực thi: Văn bản chuyển đổi sẽ được phân tích để nhận diện ý định người dùng. Nếu phân tích đúng, hệ thống thực thi lệnh và phản hồi lại bằng giọng nói. Nếu phân tích lỗi, hệ thống sẽ báo lỗi và phát âm thanh thông báo lỗi qua Text-to-Speech (TTS), đồng thời LED nhấp xanh lá.

- Kết thúc quy trình: Sau khi hoàn tất xử lý hoặc khi có lỗi, hệ thống quay lại trạng thái chờ để nhận diện wake word mới, hoàn thành một chu trình xử lý giọng nói.



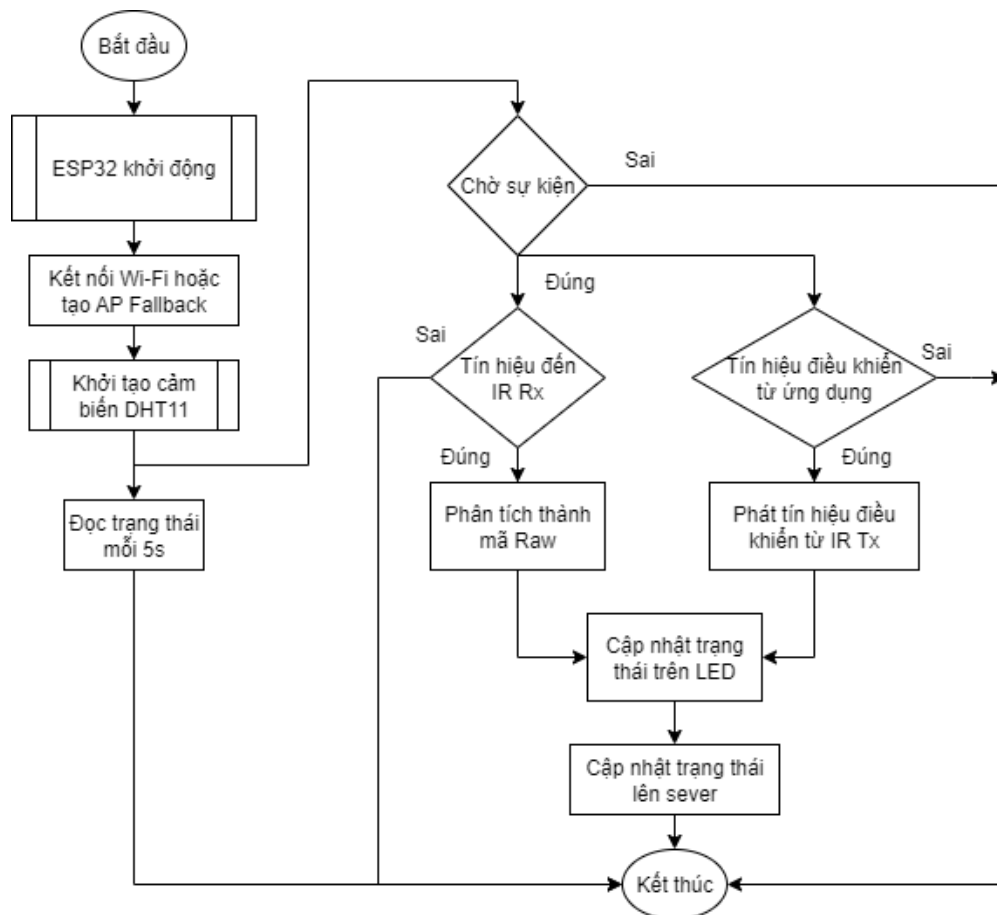
Hình 20: Lưu đồ giải thuật Node nhận diện giọng nói

3.4.2. Lưu đồ giải thuật Node điều khiển ngoại vi phòng ngủ (Bed Room)

Lưu đồ này mô tả quy trình hoạt động của Node điều khiển ngõ ra trong phòng ngủ (Bed Room) dựa trên nền tảng ESP32:

- Khởi động hệ thống: ESP32 khởi động và kiểm tra kết nối Wi-Fi. Nếu thất bại, tạo AP Fallback.
- Kết nối và đọc dữ liệu DHT11: Sau khi kết nối Wi-Fi, hệ thống kết nối và đọc dữ liệu từ cảm biến DHT11.
- Đọc trạng thái từ cảm biến môi trường: Lấy dữ liệu trạng thái từ cảm biến môi trường.

- Chờ sự kiện: Hệ thống chờ tín hiệu từ IR Rx hoặc tín hiệu điều khiển từ đám mây.
- Xử lý sự kiện:
 - Nếu nhận tín hiệu từ IR Rx, phân tích tín hiệu thô (Raw).
 - Nếu nhận tín hiệu điều khiển từ đám mây, phân tích tín hiệu điều khiển từ IR Tx.
- Cấp nguồn sáng cho LED: Dựa trên phân tích, cấp nguồn sáng cho LED.
- Cập nhật trạng thái lên server: Gửi trạng thái cập nhật lên server.
- Kết thúc: Quy trình lặp lại hoặc kết thúc.

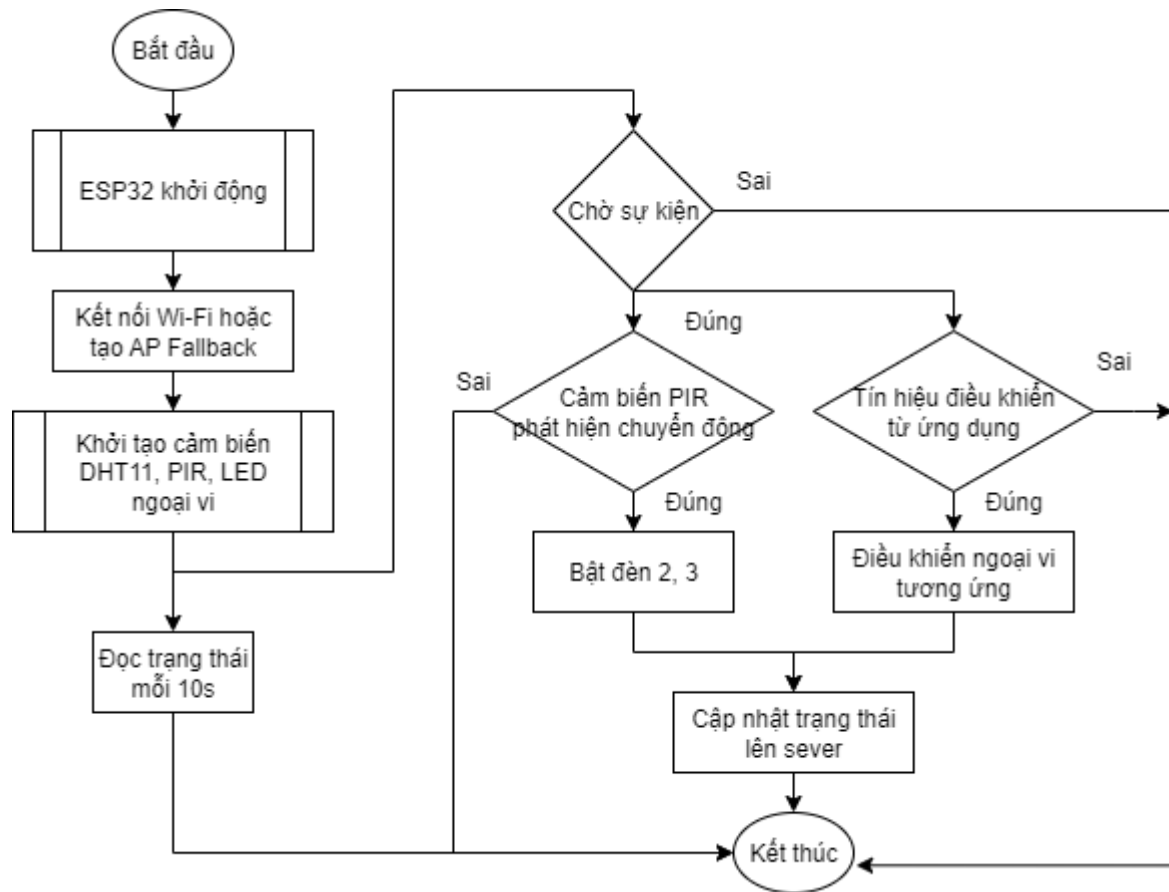


Hình 21: Lưu đồ giải thuật Node điều khiển ngoại vi phòng ngủ (Bed Room)

3.3.3. Lưu đồ giải thuật Node điều khiển ngoại vi phòng khách (Living Room)

Lưu đồ này mô tả quy trình hoạt động của Node điều khiển ngõ ra trong phòng khách (Living Room) dựa trên nền tảng ESP32:

- Khởi động hệ thống: ESP32 khởi động và kiểm tra kết nối Wi-Fi. Nếu thất bại, tạo AP Fallback.
- Kết nối và đọc dữ liệu DHT11, PIR: Sau khi kết nối, hệ thống đọc dữ liệu từ cảm biến DHT11 và PIR, với LED báo hiệu khởi động.
- Đọc trạng thái từ cảm biến trong 10s: Lấy dữ liệu trạng thái từ cảm biến trong 10 giây.
- Chờ sự kiện: Hệ thống chờ tín hiệu từ cảm biến PIR hoặc tín hiệu điều khiển từ đám mây.
- Xử lý sự kiện:
 - Nếu cảm biến PIR phát hiện chuyển động, bật đèn 2, 3.
 - Nếu nhận tín hiệu điều khiển từ đám mây, điều khiển ngõ ra theo ứng dụng.
- Cập nhật trạng thái lên server: Gửi trạng thái cập nhật lên server.
- Kết thúc: Quy trình lặp lại hoặc kết thúc.

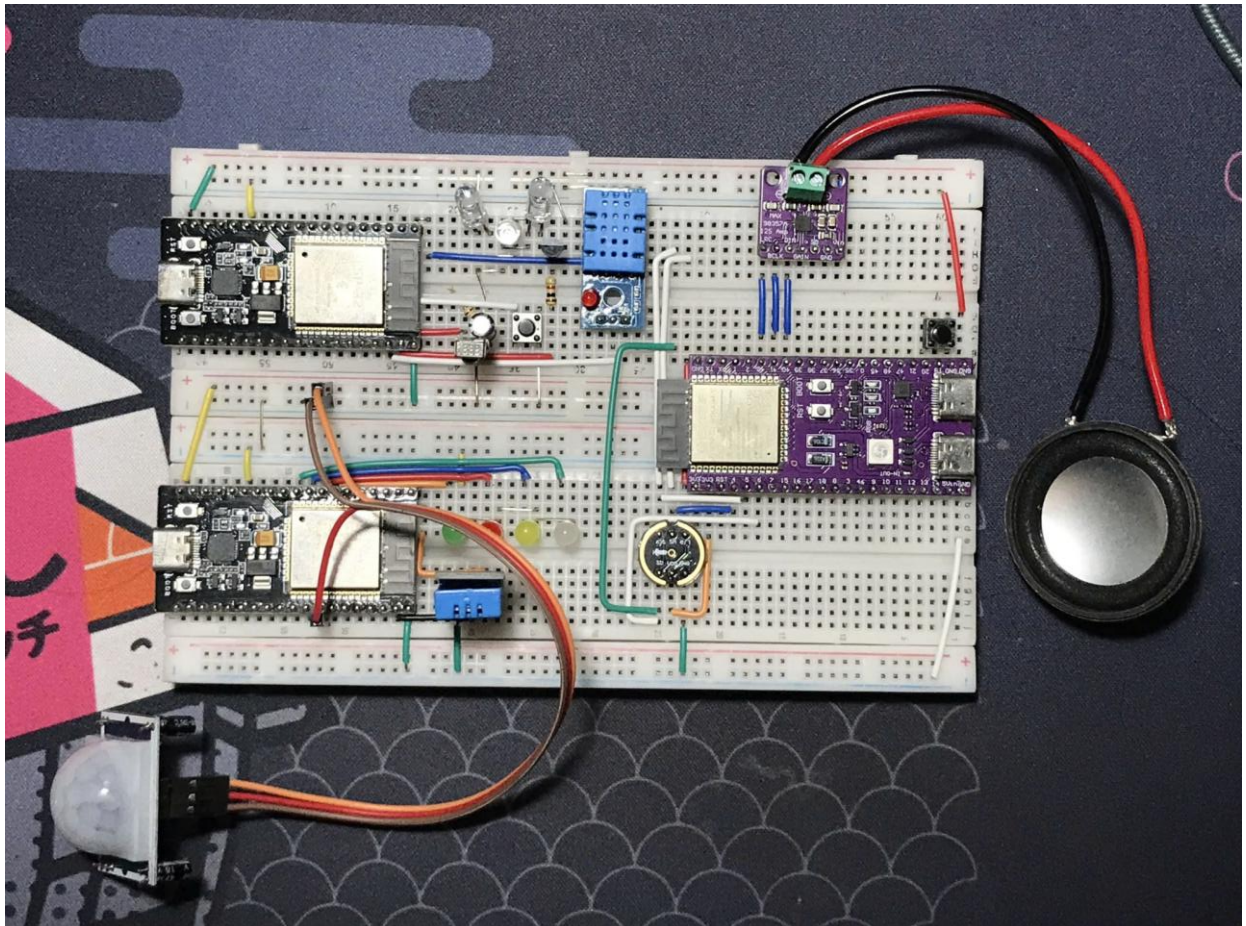


Hình 22: Lưu đồ giải thuật Node điều khiển ngoại vi phòng khách (Living Room)

CHƯƠNG 4: KẾT QUẢ THỰC HIỆN

4.1. Mô hình thực tế

Từ các linh kiện phần cứng và sơ đồ khối đã thiết kế, nhóm tiến hành lắp đặt hệ thống được mô tả tại hình bên dưới:



Hình 23: Mô hình thực tế của hệ thống

4.2. Kết quả hoạt động

4.2.1. Node nhận điện giọng nói

- Khởi động
 - Khi khởi động đèn LED trên board sẽ sáng màu xanh ngọc rồi tắt, hệ thống thiết lập các kết nối đến máy chủ và khởi tạo ngoại vi
 - Thời gian khởi động trung bình khoảng 10 giây

- Nhận diện giọng nói (tỉ lệ nhận diện, thời gian phản hồi)
 - o Hệ thống nhận diện giọng nói được khởi động bằng từ khóa đánh thức "alexa", sau khi phát hiện từ khóa đèn LED trên board nháy trắng kèm theo âm thanh thông báo và quá trình nhận diện câu lệnh bắt đầu
 - o Khi hệ thống phát hiện giọng nói sau khi được đánh thức đèn LED sẽ chuyển sang nháy xanh liên tục cho đến khi kết thúc lời nói từ người dùng. Sau đó âm thanh được gửi về máy chủ để xử lý và thực thi. Trong quá trình nhận diện câu lệnh, nếu xảy ra lỗi thì đèn LED sẽ nháy màu vàng và người dùng có thể đánh thức hệ thống để tiến hành lại.
 - o Sau khi câu lệnh đã được xử lý thì máy chủ sẽ gửi thông điệp đến node dưới dạng văn bản sang lời nói, đèn LED sẽ nháy xanh lá cho đến khi kết thúc thông điệp.
- Để kiểm tra tính ổn định của node khi hoạt động, nhóm tiến hành kiểm thử 10 lần với các câu lệnh điều khiển và kiểm tra trạng thái, quá trình kiểm thử diễn ra với các điều kiện môi trường có tiếng ồn khác nhau từ ~50dB đến ~90dB, 'x' là có phản hồi và giá trị số là thời gian nhận được phản hồi giọng nói.

Bảng kết quả kiểm thử chức năng node nhận diện giọng nói:

Lần thử	~50dB			~90dB		
	Đánh thức	Bật/tắt đèn	Trạng thái	Đánh thức	Bật/tắt đèn	Trạng thái
1	x	4	6	x	lỗi	7
2	x	4	5	x	4	4
3	x	6	5	x	6	5
4	x	lỗi	5	x	lỗi	4
5	x	4	4	x	9	5

6	x	6	7	lỗi	lỗi	lỗi
7	x	6	4	x	5	5
8	x	lỗi	4	x	5	5
9	x	4	4	x	6	5
10	x	4	5	x	lỗi	5
	10/10	4.75s	4.9s	9/10	5.8s	5s

Bảng 11: Kết quả kiểm thử chức năng node nhận diện giọng nói

Qua quá trình kiểm thử thì có thể đánh giá tỉ lệ nhận diện từ đánh thức khá cao, tỉ lệ nhận diện câu lệnh bật tắt tương đối tốt tuy nhiên tỉ lệ lỗi vẫn cao hơn nhận diện lệnh kiểm tra trạng thái, có thể do cú pháp câu lệnh phức tạp hơn dẫn đến nguyên nhân này. Thời gian phản hồi câu lệnh giao động từ 4-10 giây.

4.2.2. Node điều khiển thiết bị/ngoại vi

- Khởi động
 - o Các node tại phòng ngủ và phòng khách có thời gian khởi động trung bình khoảng 10 giây
- Thực thi
 - o Ngoại vi của các node được điều khiển thông qua ứng dụng, web local trên máy chủ hoặc thông qua trợ lý giọng nói. Thời gian thực thi lệnh điều khiển được thể hiện thông qua bảng dưới

Bảng kết quả kiểm thử chức năng node điều khiển thiết bị/ngoại vi:

Lần thử	Bật/tắt đèn	Điều khiển IR	Thu IR
1	<1s	<1s	<1s
2	<1s	<1s	<1s

3	<1s	<1s	<1s
4	<1s	Lỗi	<1s
5	<1s	Lỗi	<1s
6	<1s	<1s	<1s
7	<1s	<1s	<1s
8	<1s	<1s	<1s
9	<1s	<1s	<1s
10	<1s	<1s	<1s

Bảng 12: Kết quả kiểm thử chức năng node điều khiển thiết bị/ngoại vi

Qua kiểm thử có thể đánh giá thời gian thực thi điều khiển các thiết bị/ngoại vi có độ trễ rất ngắn, gần như là tức thì thời gian thực. Các lỗi xảy ra chủ yếu ở bộ phát hồng ngoại do chưa tối ưu được giải thuật phát tín hiệu.

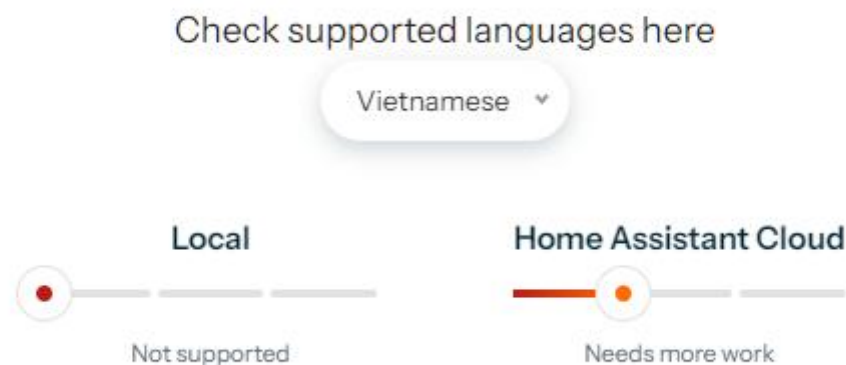
4.3. Nhận xét và đánh giá

- Ưu điểm

Ưu điểm chính là khả năng điều khiển giọng nói với **độ trễ thấp** và **bảo mật cao** nhờ ESP32-S3, cùng với nền tảng quản lý trực quan của Home Assistant. Việc tích hợp các cảm biến như **DHT11**, **IR TX/RX**, và **PIR** còn giúp hệ thống tự động hóa thông minh, điều khiển cả thiết bị cũ, mang lại tiện ích vượt trội và tính ứng dụng thực tiễn cao.

- Nhược điểm

Chưa chính thức hỗ trợ nhận diện giọng nói tiếng Việt, tỉ lệ lỗi cao



Hình 24: Trạng thái hỗ trợ ngôn ngữ tiếng Việt từ Open Home Foundation

- Sử dụng mic mono nên độ phủ chưa cao, tỉ lệ nhận diện giảm đáng kể khi môi trường có tiếng ồn lớn
- Không sử dụng chip xử lý âm thanh ngoài nên chất lượng âm thanh đầu vào chưa đạt hiệu quả tốt nhất
- Module amp Max98357 chỉ có công suất 3W giới hạn công suất loa, ngõ ra tín hiệu còn nhỏ

Bảng thống kê đánh giá tổng quát hệ thống:

Công việc	Số lần thử	Độ chính xác	Sai số	Thời gian đáp ứng	Đánh giá	Chú thích
Cho hệ thống hoạt động liên tục	336h	336h		Ổn định	Đạt	Cần tản nhiệt cho Raspi tốt hơn
Đánh thức trợ lý	20	19/20		<1s	Đạt	Môi trường ảnh hưởng chất lượng thu âm từ mic
Nhận câu lệnh và thực thi	40	33/40		~5.11s	Đạt	

Điều khiển trực tiếp	30	28/30		<1s	Đạt	
-------------------------	----	-------	--	-----	-----	--

Bảng 13: Thống kê đánh giá tổng quát hệ thống

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Sau thời gian nghiên cứu và triển khai, nhóm thực hiện đã cơ bản hoàn thành đề tài với các chức năng đáp ứng được các yêu cầu thiết kế ban đầu. Hệ thống đã được thử nghiệm trong môi trường thực tế và mang lại các kết quả khả quan.

Về phần cứng:

- Hệ thống có thể sử dụng môi trường gia đình.
- Tích hợp nhiều cảm biến và module như microphone I2S, loa, LED WS2812, cảm biến chuyển động PIR, IR thu/phát,... tạo nên hệ thống điều khiển đa dạng.
- Hệ thống hoạt động ổn định trong thời gian dài, tiêu thụ điện năng thấp.
- Các tín hiệu điều khiển trực quan bằng giọng nói hoặc giao diện người dùng, giúp theo dõi trạng thái thiết bị dễ dàng.
- Hệ thống thu và phát tín hiệu hồng ngoại hoạt động hiệu quả với nhiều loại thiết bị.

Về phần mềm:

- Kết nối ổn định, thời gian khởi động và kết nối với Home Assistant dưới 10 giây.
- Tính năng nhận diện từ khóa đánh thức (wake word) hoạt động ổn định, độ nhạy cao.
- Trợ lý giọng nói có thể nhận và xử lý lệnh nhanh chóng với thời gian phản hồi tốt (dưới 10 giây sau khi ghi âm).
- Tích hợp tốt với hệ thống Home Assistant thông qua ESPHome, có thể điều khiển các thiết bị khác trong nhà như đèn, quạt, cảm biến.
- Hệ thống có thể học và lưu nhiều mã điều khiển từ remote IR và gán vào các nút điều khiển ảo (virtual button).

- Có khả năng mở rộng, tùy biến cao nhờ cấu trúc phần mềm linh hoạt và khả năng tích hợp với các dịch vụ mới trong hệ sinh thái Home Assistant.

Tóm lại, đề tài đã chứng minh được tính khả thi và tiềm năng ứng dụng trong các hệ thống nhà thông minh. Hệ thống không chỉ mang lại sự tiện lợi trong điều khiển thiết bị mà còn tạo ra trải nghiệm tương tác tự nhiên hơn thông qua giọng nói. Trong tương lai, khi hệ thống trợ lý giọng nói hỗ trợ tiếng Việt chính thức, hệ thống sẽ càng trở nên thân thiện và dễ sử dụng hơn với người dùng trong nước. Đề tài có tiềm năng phát triển và thương mại hóa, góp phần thúc đẩy ứng dụng công nghệ IoT vào đời sống hàng ngày.

5.2. Hướng phát triển

Thông qua quá trình tìm hiểu và thực hiện đề tài cũng với những kết quả đạt được, nhóm thực hiện đã cơ bản hoàn thiện hệ thống qua đó đưa ra hướng phát triển tiếp theo trong tương lai:

- Tích hợp thêm nhiều thiết bị điều khiển khác như rèm cửa, quạt, máy lạnh thông qua IR hoặc các giao thức khác như Zigbee, Bluetooth, hoặc Matter.
- Mở rộng khả năng nhận diện giọng nói để hỗ trợ nhiều ngữ cảnh phức tạp hơn, chẳng hạn như điều kiện thời gian, trạng thái thiết bị, ngữ nghĩa mệnh lệnh.
- Hỗ trợ sử dụng tiếng Việt làm ngôn ngữ điều khiển chính khi Home Assistant và ESPHome hỗ trợ chính thức các mô hình nhận dạng tiếng Việt hoặc cho phép tích hợp dễ dàng mô hình ngôn ngữ riêng.
- Sử dụng mô hình LLM (Large Language Model) tích hợp cục bộ (on-device) hoặc hybrid (kết hợp cloud + local) để cải thiện độ chính xác và khả năng hiểu ngữ cảnh.
- Nâng cấp giao diện người dùng trực quan hơn trên màn hình OLED hoặc tích hợp ứng dụng di động đơn giản để thiết lập, tùy chỉnh lệnh và kiểm soát thiết bị.

- Tối ưu hóa hiệu suất xử lý âm thanh và tiêu thụ điện năng của ESP32-S3 để tăng thời gian hoạt động liên tục và giảm độ trễ phản hồi.
- Thêm chức năng học lệnh giọng nói cá nhân hoá, cho phép người dùng gán lệnh tùy chỉnh và lưu vào bộ nhớ thiết bị.
- Đồng bộ trạng thái thiết bị giữa remote vật lý, giọng nói và ứng dụng điều khiển nhằm đảm bảo tính nhất quán và phản hồi thời gian thực.

TÀI LIỆU THAM KHẢO

- [1] Open Home Foundation, *Home Assistant*. [Online]. Available: <https://www.home-assistant.io/>. [Accessed: Aug. 4, 2025].
- [2] Open Home Foundation, *S3 Box Voice Assistant Documentation*. [Online]. Available: https://www.home-assistant.io/voice_control/s3_box_voice_assistant/. [Accessed: Aug. 4, 2025].
- [3] Open Home Foundation, *Voice Control Overview*. [Online]. Available: https://www.home-assistant.io/voice_control/. [Accessed: Aug. 4, 2025].
- [4] Open Home Foundation, *Voice Pipeline (Preview Edition)*. [Online]. Available: <https://www.home-assistant.io/voice-pe/>. [Accessed: Aug. 4, 2025].
- [5] Nabu Casa Support, *Getting Started with Home Assistant Preview Edition*. [Online]. Available: <https://support.nabucasa.com/hc/en-us/articles/25918770371229-Getting-started-with-Home-Assistant-Preview-Edition>. [Accessed: Aug. 4, 2025].
- [6] Open Home Foundation, *ESPHome Voice Assistant Component*. [Online]. Available: https://esphome.io/components/voice_assistant.html. [Accessed: Aug. 4, 2025].
- [7] InvenSense, *INMP441 MEMS Microphone Datasheet*. [Online]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/INMP441.pdf>. [Accessed: Aug. 4, 2025].
- [8] Adafruit Industries, *MAX98357 I2S Mono Amp Datasheet*. [Online]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-max98357-i2s-class-d-mono-amp.pdf>. [Accessed: Aug. 4, 2025].
- [9] Adafruit Industries, *PIR Motion Sensor Datasheet*. [Online]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>. [Accessed: Aug. 4, 2025].

- [10] Sivago, *VS1838B Infrared Receiver Module Datasheet*. [Online]. Available: <https://www.sivago.com.cn/upload/pdf/2022/VS1838B.pdf>. [Accessed: Aug. 4, 2025].
- [11] IEEE 802.11 Working Group, *IEEE 802.11 Wireless LANs*. [Online]. Available: <https://www.ieee802.org/11/>. [Accessed: Aug. 4, 2025].
- [12] NXP Semiconductors, *I2S Bus Specification and User Manual (UM11732)*. [Online]. Available: <https://www.nxp.com/docs/en/user-manual/UM11732.pdf>. [Accessed: Aug. 4, 2025].
- [13] Open Home Foundation, *Voice Chapter 8: Assist in the Home*. [Online]. Available: <https://www.home-assistant.io/blog/2024/12/19/voice-chapter-8-assist-in-the-home/>. [Accessed: Aug. 4, 2025].