

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA KỸ THUẬT ĐIỆN TỬ I

=====o0o=====



BÁO CÁO KẾT QUẢ ĐỀ TÀI

**Thiết kế và xây dựng hệ thống nhà thông minh giám sát và
điều khiển bằng giọng nói ứng dụng mô hình CNN**

Giảng viên hướng dẫn : ThS. Trần Thị Thanh Thủy

Nhóm : 9

Lớp : D22XLTH

Sinh viên thực hiện : Đặng Hữu Thắng-B22DCDT305

Vũ Thị Thanh Thúy-B22DCDT312

Hà Nội, 2026

LỜI CẢM ƠN

Trước tiên, em muốn gửi lời tri ân sâu sắc đến quý thầy cô khoa Điện tử - những người đã tận tâm truyền đạt kiến thức cho em, từ những kiến thức cơ bản nhất cho đến những kiến thức chuyên sâu và phức tạp hơn. Nhờ sự dày công dạy dỗ, hướng dẫn tận tình của các thầy cô, em đã có được nền tảng kiến thức vững chắc, đủ để tiến hành thực hiện và hoàn thành đề tài nghiên cứu quan trọng này một cách thuận lợi. Đặc biệt, em muốn bày tỏ lòng biết ơn chân thành nhất tới cô Trần Thị Thanh Thủy- người đã trực tiếp giảng dạy, hướng dẫn tận tình và luôn sẵn lòng giải đáp mọi thắc mắc của em. Cô đã tạo mọi điều kiện thuận lợi để em có thể hoàn thành đề tài một cách tốt nhất.

Em cũng xin gửi lời cảm ơn chân thành tới các anh chị, các bạn cùng khóa. Trong suốt quá trình thực hiện đề tài, chúng em đã cùng nhau san sẻ, giúp đỡ lẫn nhau, hợp tác một cách hiệu quả. Nhờ sự hỗ trợ quý báu từ các bạn, em đã vượt qua được nhiều khó khăn, thử thách và hoàn thành đề tài đúng tiến độ cũng như đáp ứng được thời gian quy định.

Tuy nhiên, do kiến thức và kinh nghiệm của em vẫn còn hạn chế, nên không tránh khỏi đề tài này vẫn còn nhiều thiếu sót, hạn chế về cả nội dung lẫn hình thức trình bày. Em xin rất mong nhận được sự thông cảm từ quý thầy cô. Đồng thời, em cũng kính mong quý thầy cô sẽ tận tình đóng góp ý kiến quý báu, chỉ ra những hạn chế và gợi ý những cải tiến cần thiết để em có thể hoàn thiện, nâng cao chất lượng các mô hình nghiên cứu trong tương lai, đưa ra những sản phẩm hoàn chỉnh và toàn diện nhất.

Một lần nữa em xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN	i
MỤC LỤC.....	1
DANH MỤC CHỮ VIẾT TẮT.....	3
DANH MỤC HÌNH VẼ.....	4
DANH MỤC BẢNG BIỂU	4
LỜI MỞ ĐẦU	6
CHƯƠNG 1: TỔNG QUAN VỀ IOT, ĐIỆN TOÁN TẠI BIÊN VÀ ĐIỆN TOÁN Đám MÂY	2
1.1. Tổng quan về IoT	2
1.1.1. Khái niệm.....	2
1.1.2. Mô hình hệ thống IoT.....	2
1.1.3. Điện toán đám mây.....	4
1.2. Điện toán biên	5
1.2.1. Khái niệm và kiến trúc.....	5
1.2.2. Ưu và nhược điểm của điện toán biên.....	6
1.3. Mối quan hệ giữa IoT, Điện toán đám mây, Điện toán biên và ứng dụng	6
1.4. Kết luận chương 1.....	8
CHƯƠNG 2: CÁC CÔNG NGHỆ VÀ GIẢI PHÁP ĐƯỢC SỬ DỤNG	9
2.1. Phần cứng sử dụng.....	9
2.1.1. Khối xử lý AI	9
2.1.2. Khối trung tâm điều phối.....	10
2.1.3. Các node cảm biến.....	11
2.1.4. Các cảm biến sử dụng.....	12
2.2. Các giao thức được sử dụng.....	14
2.2.1. Giao thức giao tiếp phần cứng	14

2.2.2. Giao thức giao tiếp không dây	17
2.3. Phần mềm sử dụng.....	21
2.3.1. Công cụ lập trình	21
2.3.2. Các framework phát triển hệ thống	23
2.4. Giải pháp tích hợp mô hình học sâu vào thiết bị nhúng	26
2.5. Kết luận chương 2.....	27
CHƯƠNG 3: ĐỀ XUẤT PHÁT TRIỂN HỆ NHÀ THÔNG MINH GIÁM SÁT VÀ ĐIỀU KHIỂN BẰNG GIỌNG NÓI.....	28
3.1. Mô hình đề xuất.....	28
3.2. Thiết kế và xây dựng Node Edge-AI xử lý giọng nói	30
3.2.1. Đường ống xử lý tín hiệu	30
3.2.2. Quy trình xây dựng và số hóa tập lệnh điều khiển.....	33
3.3. Kết quả và đánh giá	34
3.3.1. Triển khai hệ thống.....	34
3.3.2. Kết quả và đánh giá	38
3.4. Kết luận chương 3.....	39
KẾT LUẬN	40
TÀI LIỆU THAM KHẢO.....	41

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Thuật ngữ tiếng anh	Nghĩa tiếng việt
ADC	Analog-to-Digital Converter	Bộ chuyển đổi tín hiệu tương tự sang số
AEC	Acoustic Echo Cancellation	Khử tiếng vọng âm thanh
AFE	Audio Front-End	Khối tiền xử lý âm thanh
AGC	Automatic Gain Control	Tự động điều chỉnh độ lợi
AI	Artificial Intelligence	Trí tuệ nhân tạo
AIoT	Artificial Intelligence of Things	Trí tuệ nhân tạo vạn vật
AQI	Air Quality Index	Chỉ số chất lượng không khí
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
CRNN	Convolutional Recurrent Neural Network	Mạng nơ-ron tích chập kết hợp hồi quy
DMA	Direct Memory Access	Truy cập bộ nhớ trực tiếp
ESP-IDF	Espressif IoT Development Framework	Khung phát triển IoT của Espressif
ESP-NOW	Espressif Non-Wifi Protocol	Giao thức truyền thông không dây của Espressif
G2P	Grapheme-to-Phoneme	Chuyển đổi từ tự dạng sang âm vị
GPIO	General Purpose Input/Output	Cổng vào/ra đa mục đích
HMI	Human-Machine Interface	Giao diện người - máy
I2C	Inter-Integrated Circuit	Giao thức giao tiếp nối tiếp đồng bộ
I2S	Inter-IC Sound	Giao thức truyền tải âm thanh kỹ thuật số
IoT	Internet of Things	Internet vạn vật
MQTT	Message Queuing Telemetry Transport	Giao thức truyền thông theo mô hình hàng đợi
NS	Noise Suppression	Triệt tiêu nhiễu
PSRAM	Pseudo Static Random Access Memory	Bộ nhớ truy cập ngẫu nhiên giả tĩnh
RISC-V	Reduced Instruction Set Computer - V	Kiến trúc máy tính tập lệnh rút gọn V
SPI	Serial Peripheral Interface	Giao diện ngoại vi nối tiếp
SRAM	Static Random Access Memory	Bộ nhớ truy cập ngẫu nhiên tĩnh

DANH MỤC HÌNH VẼ

Hình 1.1. Mô hình 4 lớp IoT	3
Hình 1.2. Mô hình điện toán đám mây	4
Hình 1.3. Mô hình điện toán biên.....	5
Hình 2.1. ESP32-S3 N16R8.....	10
Hình 2.2 ESP32-WROOM.....	10
Hình 2.3. ESP32-C3 Super mini.....	11
Hình 2.4. Cảm biến DHT11	12
Hình 2.5. Cảm biến ánh sáng BH1750	13
Hình 2.6. Micro INMP441	14
Hình 2.7. Giao thức one-wire	14
Hình 2.8. kết nối I2C.....	15
Hình 2.9. Khung bản tin I2C.	16
Hình 2.10. Cấu trúc khung bản tin I2S	16
Hình 2.11. Cách kết nối sử dụng giao thức UART	17
Hình 2.12. Khung dữ liệu của giao thức UART.....	17
Hình 2.13. Giao thức ESP-NOW.....	19
Hình 2.14. Khung bản tin của giao thức ESP-NOW	20
Hình 2.15. Cấu trúc mạng	21
Hình 2.16. Framework ESP-IDF	23
Hình 2.17. kiến trúc tổng quan hệ thống nhận dạng giọng nói	25
Hình 2.18. Quy trình xử lý tín hiệu khử nhiễu tại khối Audio Front-End.....	25
Hình 2.19. Kiến trúc tổng quát của mạng CRNN.....	27
Hình 3.1. Sơ đồ kết nối các node cảm biến.....	29
Hình 3.2. Quá trình chuyển đổi Grapheme-to-Phoneme	34
Hình 3.3. GateWay	35
Hình 3.4. Node cảm biến.....	36
Hình 3.5. Node Edge-AI	37
Hình 3.6. Giao diện Web Dashboard.....	38

DANH MỤC BẢNG BIỂU

Bảng 2.1 Thông số kỹ thuật.....	9
Bảng 3.1. Hiệu năng trên các mô hình WakeNet	31
Bảng 3.2. Hiệu năng trên các mô hình MultiNet.....	32
Bảng 3.3. Bảng ánh xạ lệnh điều khiển và chuỗi âm vị thực tế	34

LỜI MỞ ĐẦU

Trong Sự phát triển vượt bậc của học sâu và xu hướng TinyML đã mở ra kỷ nguyên mới cho các hệ thống nhúng thông minh. Đặc biệt, việc ứng dụng mạng nơ-ron tích chập vào bài toán nhận dạng từ khóa “Keyword Spotting” cho phép điều khiển thiết bị bằng giọng nói với độ chính xác cao ngay trên phần cứng giới hạn tài nguyên, giảm thiểu độ trễ và sự phụ thuộc vào đám mây.

Tại Việt Nam, nhu cầu về nhà thông minh ngày càng tăng nhưng việc triển khai AI trên các nền tảng giá rẻ vẫn còn nhiều thách thức. Xuất phát từ thực tế đó, đề tài “Thiết kế và xây dựng hệ thống nhà thông minh giám sát và điều khiển bằng giọng nói ứng dụng mô hình CNN” được đề xuất. Mục tiêu là xây dựng một giải pháp toàn diện, kết hợp xử lý AI tại biên và mạng truyền thông hiệu suất cao để đảm bảo sự ổn định và tiện nghi.

Hệ thống được thiết kế sử dụng mô hình CNN để nhận diện lệnh giọng nói, kết hợp giao thức ESP-NOW cho mạng cảm biến thời gian thực và MQTT để quản lý từ xa. Báo cáo đề tài sẽ tập trung vào việc mô tả chi tiết quá trình thiết kế, phát triển và đánh giá hiệu quả của hệ thống, bao gồm:

1. CHƯƠNG 1: TỔNG QUAN VỀ IOT, ĐIỆN TOÁN TẠI BIÊN VÀ ĐIỆN TOÁN ĐÁM MÂY
2. CHƯƠNG 2: CÁC CÔNG NGHỆ VÀ GIẢI PHÁP ĐƯỢC SỬ DỤNG.
3. CHƯƠNG 3: ĐỀ XUẤT PHÁT TRIỂN HỆ NHÀ THÔNG MINH GIÁM SÁT VÀ ĐIỀU KHIỂN BẰNG GIỌNG NÓI.

CHƯƠNG 1: TỔNG QUAN VỀ IOT, ĐIỆN TOÁN TẠI BIÊN VÀ ĐIỆN TOÁN Đám Mây

Hệ thống cân thông minh được phát triển trong đề tài này về bản chất là một thiết bị IoT, ứng dụng học sâu để giải quyết bài toán thực tiễn trong lĩnh vực bán lẻ. Các hệ thống IoT truyền thống thường phụ thuộc vào Điện toán đám mây để xử lý dữ liệu, một mô hình bộc lộ nhiều hạn chế về độ trễ, chi phí băng thông và bảo mật. Để khắc phục những nhược điểm này, kiến trúc Điện toán tại biên đã ra đời, cho phép thực hiện các tác vụ tính toán và suy luận AI ngay tại thiết bị. Chương này sẽ trình bày tổng quan về các khái niệm nền tảng này, làm rõ mối quan hệ tương hỗ giữa IoT, Điện toán đám mây và Điện toán tại biên, qua đó cung cấp cơ sở lý thuyết vững chắc cho giải pháp AIoT được đề xuất.

1.1. Tổng quan về IoT

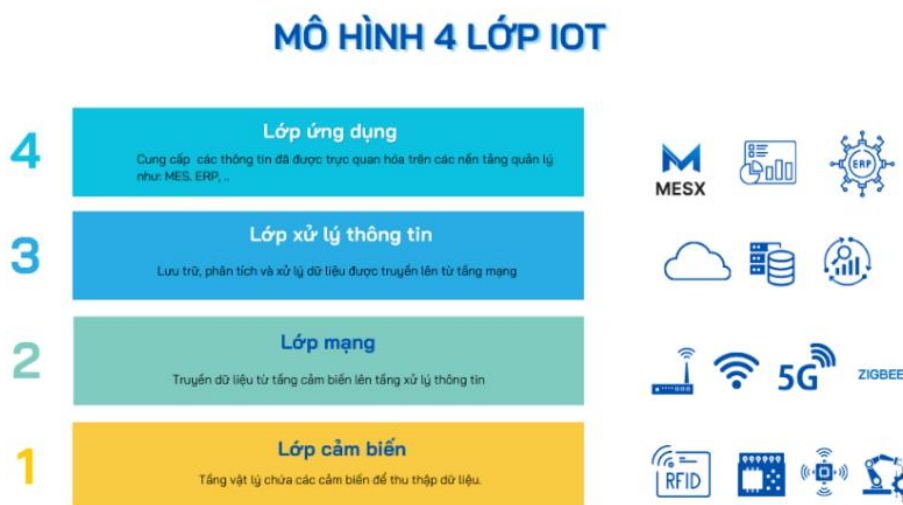
1.1.1. Khái niệm

Internet vạn vật (IoT - Internet of Things) được định nghĩa cơ bản là công nghệ liên kết mọi vật với Internet. Trong hệ thống này, mọi vật thể đều sẽ được cấp những định danh khác nhau, cho phép chúng có khả năng tự truyền thông tin dữ liệu trong một mạng lưới mà không cần thiết có sự giao tiếp giữa máy tính với con người hay giữa con người với con người. Một cách chi tiết hơn, IoT còn được định nghĩa là một hệ thống kết nối mạng lưới các thiết bị thông minh, bao gồm cảm biến, máy tính nhúng, camera, thiết bị điều khiển, máy móc, thiết bị gia dụng và nhiều loại khác, thông qua các giao thức mạng. Hệ thống mạng lưới này được tạo thành với mục đích để thu thập, truyền tải, phân tích và phản hồi thông tin giữa các thiết bị được kết nối trong thời gian thực.

1.1.2. Mô hình hệ thống IoT

Để quản lý hiệu quả toàn bộ vòng đời của dữ liệu, từ thế giới vật lý đến tay người dùng cuối, một kiến trúc hệ thống IoT điển hình thường được phân chia thành bốn tầng chức năng chính. Các tầng này không hoạt động độc lập mà phối hợp chặt chẽ với nhau, tạo thành một luồng xử lý logic và tuần tự. Luồng này bắt đầu từ việc thu thập dữ liệu thô tại nguồn, tiếp đến là truyền tải dữ liệu đó một cách an toàn qua các mạng, sau đó là xử lý và phân tích tập trung để biến dữ liệu thô thành thông tin có giá trị, và cuối cùng

là hiển thị thông tin hữu ích đó cho người dùng thông qua các ứng dụng cụ thể. Bốn tầng này bao gồm:



Hình 1.1. Mô hình 4 lớp IoT

Lớp cảm biến (sensor layer): đây là tầng vật lý, có nhiệm vụ thu thập dữ liệu thô trực tiếp từ môi trường thông qua các cảm biến ví dụ: cảm biến nhiệt độ, độ ẩm, cân nặng, camera, v.v. Dữ liệu sau khi được thu thập sẽ được chuyển tiếp đến Lớp mạng. Hiện nay, nhiều cảm biến được tích hợp công nghệ truyền không dây, cho phép chúng hoạt động ổn định và đồng bộ hóa dữ liệu trong thời gian thực.

Lớp mạng (network layer): lớp này đảm nhiệm vai trò trung gian, chịu trách nhiệm truyền tải dữ liệu một cách an toàn và tin cậy từ Lớp cảm biến đến Lớp xử lý thông tin. Để thực hiện việc này, các thiết bị trong tầng này sử dụng các giao thức truyền thông đa dạng như MQTT, HTTP, Modbus, hoặc PROFINET, nhằm đảm bảo dữ liệu được truyền đi nhanh chóng, chính xác và bảo mật.

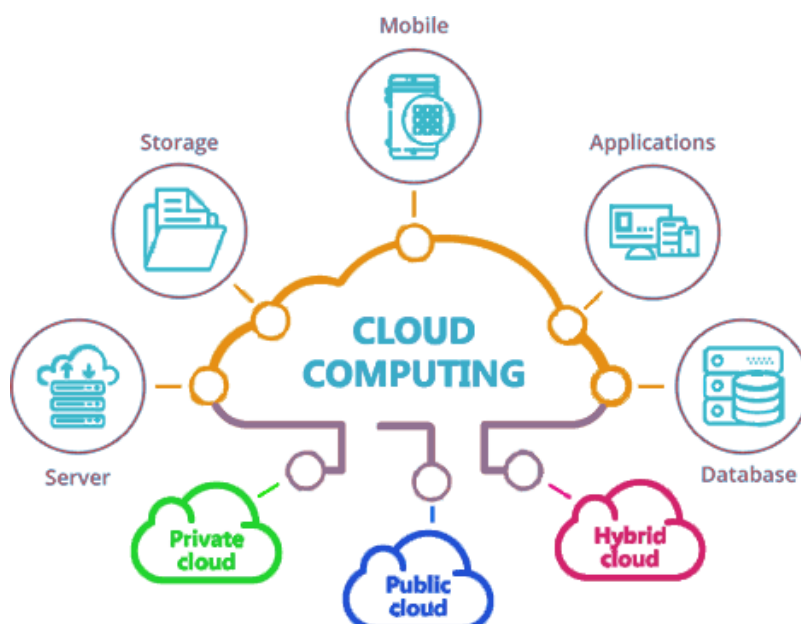
Lớp xử lý thông tin (information processing layer): đây được xem là tầng trung tâm, nơi dữ liệu thô nhận từ lớp mạng được lưu trữ, phân tích và xử lý. Tại đây, các kỹ thuật phân tích dữ liệu thông minh hoặc học máy có thể được áp dụng để trích xuất thông tin hữu ích, thực hiện nhận dạng, dự đoán và đưa ra quyết định. Kết quả sau khi xử lý sẽ được gửi đến lớp ứng dụng.

Lớp ứng dụng (application layer): đây là tầng giao tiếp trực tiếp với người dùng cuối, chịu trách nhiệm hiển thị kết quả, cung cấp các công cụ điều khiển và hỗ trợ ra

quyết định. Người dùng có thể tương tác với hệ thống thông qua các giao diện như ứng dụng web, ứng dụng trên điện thoại thông minh, hoặc giao diện HMI. Mục tiêu của lớp này là cho phép người dùng giám sát và điều khiển hệ thống một cách thuận tiện, từ đó nâng cao trải nghiệm người dùng và tối ưu hóa hiệu quả vận hành.

1.1.3. Điện toán đám mây

Theo định nghĩa của Viện Tiêu chuẩn và Công nghệ Hoa Kỳ (NIST), điện toán đám mây (Cloud Computing) là một mô hình dịch vụ cho phép người dùng truy cập tài nguyên điện toán dùng chung như mạng, máy chủ, lưu trữ, ứng dụng, và dịch vụ thông qua kết nối mạng một cách dễ dàng, mọi lúc, mọi nơi, và theo yêu cầu. Một đặc tính quan trọng là tài nguyên này có thể được thiết lập hoặc hủy bỏ nhanh chóng bởi người dùng mà không cần sự can thiệp trực tiếp từ nhà cung cấp dịch vụ.



Hình 1.2. Mô hình điện toán đám mây

Nhờ các đặc tính này, điện toán đám mây đã trở thành xu hướng tất yếu trong kỷ nguyên chuyển đổi số. Ưu điểm lớn nhất của nó là khả năng cung cấp tài nguyên tính toán linh hoạt và khả năng mở rộng gần như vô hạn. Doanh nghiệp có thể dễ dàng khởi tạo hoặc nâng cấp hệ thống mà không cần đầu tư tốn kém vào hạ tầng phần cứng ban đầu. Thay vào đó, mô hình trả phí theo mức sử dụng (pay-as-you-go) giúp tối ưu hóa chi phí vận hành và quản trị. Khả năng truy cập dữ liệu mọi lúc, mọi nơi và chia sẻ thông tin nhanh chóng cũng thúc đẩy hiệu quả phối hợp. Chính vì vậy, mô hình đám mây là giải pháp truyền thống để lưu trữ và quản lý lượng dữ liệu khổng lồ được tạo ra hàng

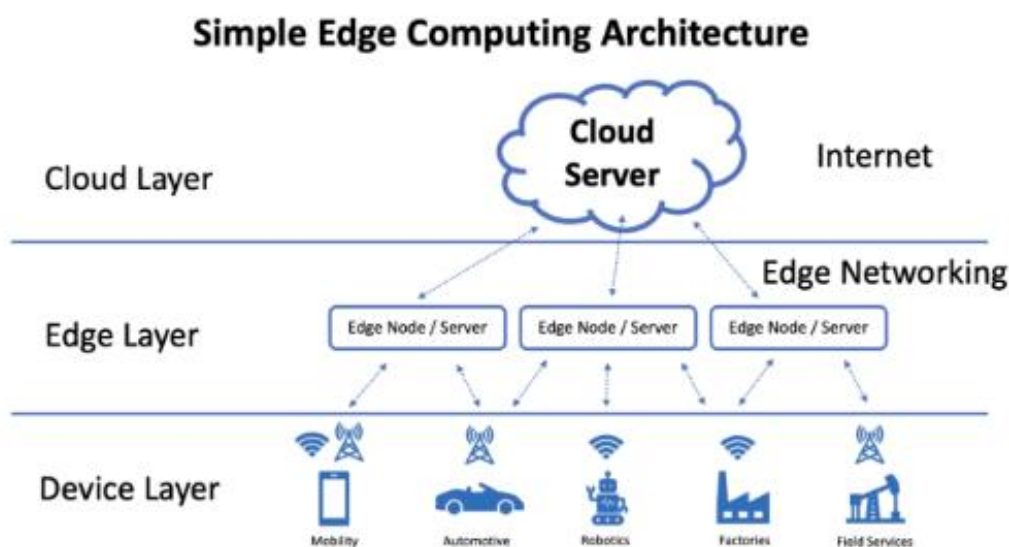
ngày bởi các thiết bị IoT.

Tuy nhiên, bên cạnh những ưu điểm vượt trội, kiến trúc điện toán đám mây cũng tồn tại các nhược điểm cố hữu. Hạn chế lớn nhất là sự phụ thuộc hoàn toàn vào kết nối mạng Internet. Bất kỳ sự cố hoặc gián đoạn đường truyền nào cũng sẽ khiến người dùng mất khả năng truy cập và xử lý dữ liệu. Một thách thức nghiêm trọng khác liên quan đến vấn đề bảo mật và quyền riêng tư. Khi sử dụng dịch vụ đám mây, người dùng buộc phải gửi toàn bộ dữ liệu, bao gồm cả thông tin nhạy cảm, lên máy chủ của một bên thứ ba. Điều này tiềm ẩn rủi ro lớn về việc rò rỉ hoặc đánh cắp dữ liệu nếu hệ thống bảo mật của nhà cung cấp không đủ mạnh.

1.2. Điện toán biên

1.2.1. Khái niệm và kiến trúc

Để khắc phục những nhược điểm cố hữu về độ trễ, băng thông và bảo mật của mô hình đám mây truyền thống, điện toán biên (Edge Computing) đã ra đời như một kiến trúc điện toán phân tán hỗ trợ. Thay vì gửi toàn bộ dữ liệu thô về một trung tâm dữ liệu hoặc đám mây ở xa để xử lý, điện toán biên đưa năng lực tính toán đến gần hơn với nguồn phát sinh dữ liệu. Ví dụ, dữ liệu từ cảm biến camera sẽ được xử lý ngay tại thiết bị hoặc tại một máy chủ biên (gateway) gần đó. Cách tiếp cận này giúp giảm đáng kể độ trễ, tăng tốc độ xử lý thông tin và giảm tải lưu lượng cho mạng lưới trung tâm.



Hình 1.3. Mô hình điện toán biên

Mô hình điện toán biên thường được mô tả qua ba tầng, như trong Hình 1.3, thể hiện sự phối hợp giữa thiết bị, biên và đám mây:

Tầng thiết bị (device layer): Đây là nơi dữ liệu được thu thập từ môi trường thực, bao gồm các cảm biến như camera, cảm biến nhiệt độ độ ẩm, ánh sáng như trong đề tài này, thiết bị thông minh, máy đo. Thay vì gửi ngay lên đám mây, các thiết bị này sẽ gửi dữ liệu đến tầng biên để xử lý cục bộ.

Tầng biên (edge layer): Đây là tầng xử lý trung gian, bao gồm các gateway thông minh, máy trạm biên hoặc trung tâm dữ liệu cục bộ. Tại đây, dữ liệu được xử lý ngay khi vừa thu thập. Các tác vụ yêu cầu độ trễ cực thấp theo thời gian thực trong hệ thống cân này sẽ được thực hiện ngay tại chỗ, giúp giảm tải cho tầng đám mây.

Tầng đám mây (cloud layer): Đám mây không bị thay thế mà chuyển sang vai trò xử lý cấp cao. Nó sẽ tiếp nhận các dữ liệu đã được xử lý sơ bộ từ tầng biên, hoặc đảm nhận các tính toán phức tạp như huấn luyện mô hình AI, phân tích dữ liệu lớn (Big Data) và tối ưu hóa quy trình. Tầng này cung cấp tài nguyên tính toán khổng lồ cho các tác vụ không yêu cầu phản hồi tức thì.

1.2.2. Ưu và nhược điểm của điện toán biên

Điện toán biên (Edge Computing) mang lại nhiều ưu điểm vượt trội so với mô hình điện toán đám mây truyền thống. Việc đặt máy chủ biên gần thiết bị người dùng giúp giảm đáng kể độ trễ truyền dữ liệu, đáp ứng tốt cho các ứng dụng IoT thời gian thực như giám sát hoặc điều khiển. Nhờ xử lý trong mạng cục bộ, hệ thống hoạt động ổn định hơn và an toàn hơn, khi dữ liệu nhạy cảm được xử lý tại chỗ thay vì gửi toàn bộ lên đám mây. Đồng thời, mô hình này giúp giảm tải băng thông, tiết kiệm chi phí truyền dữ liệu và nâng cao tốc độ phản hồi.

Tuy nhiên, Edge Computing vẫn có một số hạn chế như chi phí đầu tư ban đầu cao, yêu cầu thiết bị biên có khả năng xử lý mạnh và quản lý phân tán phức tạp. Ngoài ra, khi xảy ra sự cố hoặc mất kết nối, dữ liệu có thể bị gián đoạn, ảnh hưởng đến tính toàn vẹn và khả năng phục hồi. Dù vậy, với hiệu năng cao và độ trễ thấp, Edge Computing vẫn là giải pháp trọng yếu cho các hệ thống IoT hiện đại, đặc biệt trong các ứng dụng AIoT cho nhà thông minh.

1.3. Mối quan hệ giữa IoT, Điện toán đám mây, Điện toán biên và ứng dụng

Internet vạn vật, điện toán biên và điện toán đám mây là ba công nghệ có mối quan

hệ chặt chẽ, hỗ trợ lẫn nhau, tạo thành một kiến trúc phân tán toàn diện để xây dựng các hệ thống thông minh. Trong mô hình này, IoT đóng vai trò là tầng vật lý, là giác quan của hệ thống, chịu trách nhiệm thu thập dữ liệu thô từ môi trường thực tế. Nó bao gồm các thiết bị cảm biến, thiết bị truyền động và các đối tượng được kết nối, có khả năng giao tiếp qua mạng. Tầng này là nơi dữ liệu được sinh ra trước khi bắt kỳ quá trình xử lý nào diễn ra.

Điện toán biên hoạt động như bộ não cục bộ của kiến trúc. Tầng này nhận dữ liệu thô trực tiếp từ các thiết bị IoT và thực hiện các tác vụ xử lý, phân tích hoặc suy luận AI ngay tại hoặc gần nguồn phát sinh dữ liệu. Mục đích chính của nó là để khắc phục các nhược điểm chí mạng của mô hình đám mây thuần túy, bao gồm giảm độ trễ, tiết kiệm băng thông mạng và tăng cường bảo mật bằng cách xử lý dữ liệu nhạy cảm tại chỗ. Các tác vụ yêu cầu phản hồi tức thì, như trong các ứng dụng thời gian thực, sẽ được ưu tiên xử lý tại biên.

Điện toán đám mây đóng vai trò là bộ nhớ dài hạn của hệ thống. Nó không còn phải xử lý toàn bộ dữ liệu thô, mà thay vào đó, nó nhận các dữ liệu đã được xử lý sơ bộ hoặc các kết quả tóm tắt từ tầng biên. Với tài nguyên tính toán và lưu trữ gần như vô hạn, điện toán đám mây đảm nhận các vai trò xử lý cấp cao như: huấn luyện các mô hình AI phức tạp hay phân tích dữ liệu lớn để tìm ra xu hướng, lưu trữ dữ liệu lâu dài và cung cấp các giao diện quản lý, giám sát tập trung cho người dùng từ xa.

Sự kết hợp ba tầng này mở ra tiềm năng cho nhiều ứng dụng thực tiễn trong các lĩnh vực đòi hỏi cả khả năng phản hồi tức thời (tại biên) và khả năng phân tích sâu (trên đám mây). Các ví dụ điển hình có thể kể đến như trong Công nghiệp 4.0, nơi cảm biến IoT giám sát dây chuyền và máy chủ biên phân tích dữ liệu để cảnh báo lỗi hỏng hóc theo thời gian thực, trong khi hệ thống đám mây phân tích dữ liệu dài hạn để tối ưu hóa kế hoạch bảo trì. Tương tự, trong các thành phố thông minh, camera giao thông (IoT) sử dụng xử lý tại biên để phát hiện tai nạn, và dữ liệu luồng giao thông được đẩy lên đám mây để điều tiết ùn tắc. Trong lĩnh vực bán lẻ thông minh, các cửa hàng tự động dùng camera tại quầy để nhận diện sản phẩm tại biên và gửi hóa đơn cuối cùng lên hệ thống quản lý bán hàng trên đám mây.

Trong bối cảnh đó, hệ thống nhà thông minh được phát triển trong đề tài này chính là một ứng dụng thực tiễn của kiến trúc AIoT. Mỗi quan hệ ba tầng này được ứng

dụng trực tiếp: các cảm biến thu thập dữ liệu hình ảnh và trọng lượng; vi điều khiển ESP32-S3 thực hiện suy luận AI ngay tại chỗ để đảm bảo phản hồi tức thì; và Web Server nhận kết quả cuối cùng qua MQTT để lưu trữ và giao tiếp. Việc thiết kế và triển khai cụ thể kiến trúc này sẽ được phân tích chi tiết trong các chương tiếp theo.

1.4. Kết luận chương 1

Chương 1 đã trình bày tổng quan về các khái niệm công nghệ nền tảng, bao gồm Vạn vật kết nối, Điện toán đám mây và Điện toán tại biên. Chương này cũng đã phân tích ưu, nhược điểm của các kiến trúc xử lý và làm rõ mối quan hệ hỗ trợ lẫn nhau giữa ba công nghệ này, qua đó cung cấp cơ sở lý thuyết vững chắc cho mô hình AIoT được đề xuất trong đề tài. Dựa trên nền tảng lý thuyết đã thiết lập, Chương 2 sẽ đi sâu vào việc phân tích và lựa chọn các công nghệ, giải pháp cụ thể được sử dụng để hiện thực hóa hệ thống cân thông minh. Nội dung sẽ bao gồm các lựa chọn chi tiết về phần cứng, các giao thức giao tiếp, các framework phần mềm, và giải pháp học sâu được tích hợp vào thiết bị.

CHƯƠNG 2: CÁC CÔNG NGHỆ VÀ GIẢI PHÁP ĐƯỢC SỬ DỤNG

Để hiện thực hóa hệ thống nhà thông minh tích hợp khả năng giám sát và điều khiển trực tiếp bằng giọng nói, thách thức công nghệ nền tảng là phải lựa chọn được một kiến trúc phần cứng mạnh mẽ cùng các công cụ phát triển phần mềm tối ưu. Vấn đề cốt lõi nằm ở việc triển khai hiệu quả các mô hình mạng nơ-ron tích chập (CNN) để xử lý tín hiệu âm thanh liên tục và nhận dạng từ khóa theo thời gian thực ngay trên một thiết bị biên có tài nguyên tính toán và năng lượng hạn chế. Chương này sẽ tập trung trình bày và phân tích các nền tảng công nghệ và giải pháp kỹ thuật đã được lựa chọn để giải quyết bài toán này. Nội dung sẽ đi sâu vào đặc tả khối xử lý trung tâm, các giao thức truyền thông độ trễ thấp cho mạng cảm biến, và đặc biệt là hệ sinh thái framework chuyên dụng phục vụ cho việc tối ưu hóa và triển khai ứng dụng nhận dạng giọng nói tại biên.

2.1. Phần cứng sử dụng

2.1.1. Khối xử lý AI

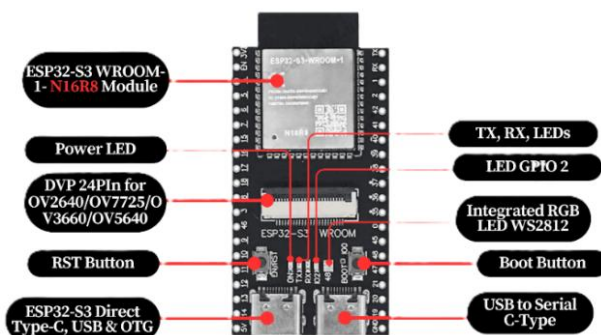
Đây là thành phần phức tạp nhất trong hệ thống, chịu trách nhiệm thu thập tín hiệu âm thanh, tiền xử lý và thực hiện suy luận mô hình học sâu để phát hiện từ khóa điều khiển. Ba dòng vi điều khiển phổ biến thường được cân nhắc là STM32, ESP32 và Raspberry Pi. Mỗi loại có những ưu nhược điểm riêng tùy thuộc vào yêu cầu về hiệu năng, khả năng kết nối và mức tiêu thụ điện.

Bảng 2.1 Thông số kỹ thuật

Tiêu chí	STM32F103C8T6	ESP32-S3-N16R8	Raspberry Pi 4
Hiệu năng xử lý	72 MHz	240 MHz	1.5 GHz
Kết nối mạng	Không hỗ trợ Wi-Fi/BLE	Tích hợp Wi-Fi và BLE 5.0	Tích hợp BLE, Wi-Fi, Ethernet
Bộ nhớ	128 KB Flash, 20 KB RAM	16 MB Flash, 8 MB PSRAM	2–8 GB RAM
Tiêu thụ điện năng	Rất thấp	Thấp – trung bình	Cao

Từ bảng so sánh trên, có thể nhận thấy ESP32-S3-N16R8 là lựa chọn tối ưu nhất nhờ sự cân bằng giữa hiệu năng xử lý và tính di động. Vi điều khiển này sở hữu hai lõi

vi xử lý Xtensa® LX7 32-bit hoạt động song song ở tần số 240 MHz, cho phép phân tách nhiệm vụ một cách hiệu quả. Đặc biệt, so với các dòng vi điều khiển truyền thống như STM32F1, ESP32-S3 vượt trội nhờ được trang bị tập lệnh vector hỗ trợ tăng tốc phần cứng cho các phép toán mạng nơ-ron, giúp giảm đáng kể thời gian suy luận của mô hình nhận dạng từ khóa và đảm bảo độ trễ thấp khi người dùng ra lệnh.



Hình 2.1. ESP32-S3 N16R8

Bên cạnh sức mạnh tính toán, phiên bản N16R8 còn giải quyết triệt để bài toán tài nguyên với 16 MB Flash và 8 MB PSRAM, cung cấp không gian lưu trữ rộng lớn cho mô hình học sâu phức tạp và bộ đệm âm thanh mà bộ nhớ nội SRAM thông thường khó đáp ứng. Đồng thời, khả năng tích hợp sẵn Wi-Fi và Bluetooth giúp thiết bị dễ dàng tham gia vào mạng cảm biến không dây mà không cần module phụ trợ rườm rà hay sự cồng kềnh như Raspberry Pi, từ đó đơn giản hóa thiết kế phần cứng và tối ưu kích thước sản phẩm. Với những ưu điểm vượt trội về cả phần cứng lẫn khả năng kết nối, ESP32-S3-N16R8 hoàn toàn đáp ứng được các yêu cầu khắt khe của một nút xử lý AI tại biên.

2.1.2. Khối trung tâm điều phối

Khối xử lý trung tâm đóng vai trò quan trọng trong toàn bộ hệ thống, chịu trách nhiệm điều phối các tác vụ từ thu thập tín hiệu cảm biến, điều hướng dữ liệu, và truyền nhận dữ liệu qua mạng từ người dùng gửi xuống. Để đáp ứng yêu cầu về khả năng xử lý đa nhiệm với chi phí tối ưu, đề tài lựa chọn sử dụng module ESP32-WROOM-32.



Hình 2.2 ESP32-WROOM

Đây là dòng vi điều khiển được phát triển dựa trên chip ESP32-D0WDQ6, sử dụng kiến trúc lõi kép Xtensa® 32-bit LX6 với tần số xung nhịp có thể đạt tới 240 MHz. Đặc điểm cấu trúc hai lõi này cho phép tối ưu hóa hiệu năng cho thiết bị GateWay: một lõi có thể được phân bổ chuyên biệt để duy trì kết nối Wi-Fi và xử lý các giao thức mạng (TCP/IP), đảm bảo đường truyền dữ liệu luôn thông suốt; trong khi lõi còn lại tập trung thực hiện logic điều khiển, đọc dữ liệu từ các node cảm biến mà không bị gián đoạn bởi các tác vụ mạng. Về khả năng kết nối, ESP32-WROOM tích hợp sẵn chuẩn Wi-Fi 802.11 b/g/n và Bluetooth, giúp thiết bị dễ dàng kết nối với Internet và các thiết bị ngoại vi khác. Với bộ nhớ 520 KB SRAM và 4 MB Flash, vi điều khiển cung cấp đủ không gian bộ nhớ đệm để lưu trữ tạm thời các gói tin dữ liệu trước khi gửi đi, hoặc lưu trữ cấu hình hệ thống. Sự kết hợp giữa khả năng xử lý mạnh mẽ và kết nối đa dạng giúp ESP32-WROOM đảm bảo tính ổn định và độ tin cậy cao trong vai trò trung tâm điều phối dữ liệu của mạng cảm biến.

2.1.3. Các node cảm biến

Trong kiến trúc mạng cảm biến không dây, các node cảm biến đóng vai trò là các điểm thu thập dữ liệu biên, được bố trí phân tán tại nhiều vị trí khác nhau để giám sát các thông số môi trường. Do đặc thù phải hoạt động độc lập, thường xuyên sử dụng nguồn pin và yêu cầu tính linh hoạt trong lắp đặt, việc lựa chọn phần cứng cho các node này đòi hỏi sự cân bằng khắt khe giữa kích thước, hiệu năng và mức tiêu thụ năng lượng. Dựa trên các tiêu chí này, đề tài lựa chọn sử dụng bo mạch ESP32-C3 Super Mini.



Hình 2.3. ESP32-C3 Super mini.

Đây là phiên bản bo mạch phát triển siêu nhỏ gọn dựa trên dòng chip SoC ESP32-C3 của Espressif. Khác với dòng ESP32-WROOM sử dụng kiến trúc Xtensa®, ESP32-

C3 được xây dựng trên kiến trúc RISC-V 32-bit single-core, hoạt động ở tần số lên đến 160 MHz. Sự thay đổi kiến trúc này mang lại ưu thế lớn về hiệu suất trên năng lượng tiêu thụ (performance-per-watt), đặc biệt phù hợp cho các ứng dụng IoT chạy pin. Sự kết hợp giữa kiến trúc tiết kiệm năng lượng và khả năng hỗ trợ phần cứng toàn diện chính là yếu tố then chốt khiến module này trở thành lựa chọn lý tưởng cho các node cảm biến chạy pin. Cụ thể, chế độ năng lượng thấp cho phép thiết lập cơ chế hoạt động ngắt quãng: thiết bị sẽ dành phần lớn thời gian ở trạng thái ngủ sâu để bảo toàn năng lượng và chỉ được đánh thức định kỳ theo bộ định thời để thực hiện thu thập số liệu, gửi về GateWay rồi lập tức quay lại trạng thái ngủ. Bên cạnh đó, dù được tối ưu hóa về kích thước và năng lượng, vi điều khiển này vẫn duy trì đầy đủ các chuẩn giao tiếp ngoại vi tiêu chuẩn như GPIO, I2C, SPI và ADC tương tự các dòng ESP32 thông thường, đảm bảo khả năng tương thích linh hoạt với đa dạng các loại cảm biến đo lường trong hệ thống mà không gặp bất kỳ rào cản nào về phần cứng.

2.1.4. Các cảm biến sử dụng

2.1.4.1. Cảm biến nhiệt độ, độ ẩm

Đối với yêu cầu giám sát các thông số môi trường cơ bản tại các node cảm biến, module DHT11 được lựa chọn nhờ sự cân bằng giữa chi phí thấp và hiệu quả hoạt động trong các ứng dụng dân dụng. Điểm mạnh của cảm biến này là khả năng tích hợp đo đồng thời cả nhiệt độ và độ ẩm trong cùng một linh kiện, giúp tối ưu hóa không gian thiết kế của bo mạch node. Ngoài ra, việc sử dụng giao thức giao tiếp Single-Wire (một dây) giúp đơn giản hóa việc kết nối với vi điều khiển ESP32-C3, tiết kiệm tài nguyên chân GPIO quý giá cho các chức năng khác.



Hình 2.4. Cảm biến DHT11

Về mặt thông số kỹ thuật, DHT11 hoạt động ổn định ở điện áp từ 3.3V đến 5.5V,

hoàn toàn tương thích với mức logic của ESP32. Cảm biến có khả năng đo độ ẩm trong khoảng từ 20% đến 90% RH với sai số $\pm 5\%$, và đo nhiệt độ trong khoảng từ 0°C đến 50°C với sai số $\pm 2^{\circ}\text{C}$. Mặc dù tốc độ lấy mẫu không quá nhanh (khoảng 1Hz), nhưng độ trễ này hoàn toàn chấp nhận được đối với việc giám sát môi trường trong nhà, nơi các thông số nhiệt ẩm không biến đổi đột ngột.

2.1.4.2. Cảm biến cường độ ánh sáng

Để phục vụ cho bài toán tự động hóa hệ thống chiếu sáng thông minh, cảm biến BH1750FVI được ưu tiên lựa chọn thay thế cho các loại quang trở (LDR) truyền thống nhờ khả năng xuất tín hiệu kỹ thuật số trực tiếp. Lý do quan trọng nhất cho sự lựa chọn này là cảm biến sở hữu đặc tính phổ phản hồi gần với độ nhạy của mắt người, giúp hệ thống "cảm nhận" độ sáng thực tế một cách chính xác nhất để đưa ra quyết định bật/tắt đèn. Hơn nữa, giao tiếp qua chuẩn I2C giúp module dễ dàng ghép nối với vi điều khiển và có khả năng chống nhiễu tốt hơn so với tín hiệu analog.



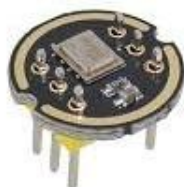
Hình 2.5. Cảm biến ánh sáng BH1750

Xét về thông số kỹ thuật, BH1750 được tích hợp bộ chuyển đổi ADC 16-bit nội bộ, cho phép phân giải cường độ sáng trong dải đo rộng từ 1 đến 65535 Lux. Cảm biến hoạt động ở điện áp thấp (2.4V - 3.6V), phù hợp với nguồn pin của node cảm biến. Đặc biệt, thiết bị có khả năng loại bỏ ảnh hưởng của nhiễu ánh sáng tần số 50Hz/60Hz từ các nguồn sáng nhân tạo, đảm bảo kết quả đo lường luôn ổn định và chính xác ngay cả trong môi trường có nhiễu thiết bị điện.

2.1.4.3. Micro đa hướng

Đóng vai trò là thính giác của hệ thống trong ứng dụng điều khiển bằng giọng nói và nhận dạng từ khóa (Keyword Spotting), module MEMS microphone INMP441

được lựa chọn nhờ khả năng cung cấp tín hiệu âm thanh chất lượng cao chuẩn I2S. Khác với các microphone analog thông thường dễ bị suy hao và nhiễu nhiều trên đường truyền, INMP441 xuất dữ liệu âm thanh dưới dạng kỹ thuật số 24-bit, cho phép kết nối trực tiếp vào giao diện I2S của ESP32 mà không cần bộ codec âm thanh bên ngoài, giúp đơn giản hóa thiết kế phần cứng và tăng độ chính xác cho mô hình xử lý AI.



Hình 2.6. Micro INMP441

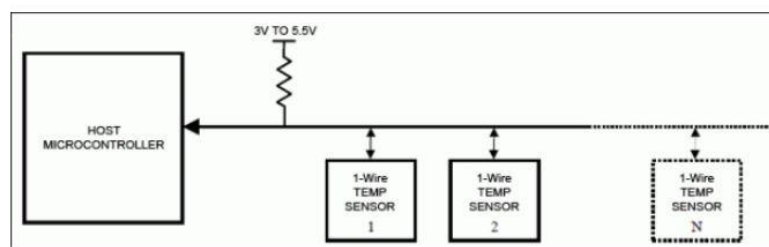
Về thông số kỹ thuật, INMP441 là loại microphone đa hướng với độ nhạy cao 26 dBFS và tỷ lệ tín hiệu trên nhiễu đạt 61 dBA, đảm bảo thu âm rõ ràng ngay cả khi nguồn phát không hướng trực tiếp vào mic. Cảm biến đáp ứng dải tần số rộng từ 60Hz đến 15kHz, bao phủ trọn vẹn dải tần số giọng nói của con người, đây là yếu tố then chốt để mô hình mạng nơ-ron có thể trích xuất đặc trưng chính xác. Ngoài ra, mức tiêu thụ dòng điện thấp khoảng 1.4 mA cũng là một lợi thế lớn khi tích hợp vào các thiết bị IoT.

2.2. Các giao thức được sử dụng

2.2.1. Giao thức giao tiếp phần cứng

2.2.1.1. Giao thức one-wire

Giao thức Single-Wire được sử dụng chuyên biệt để giao tiếp với cảm biến nhiệt độ và độ ẩm DHT11. Đây là một giao thức truyền thông nối tiếp bán song công, đặc trưng bởi khả năng truyền cả dữ liệu và tín hiệu điều khiển chỉ trên một đường dây tín hiệu duy nhất.

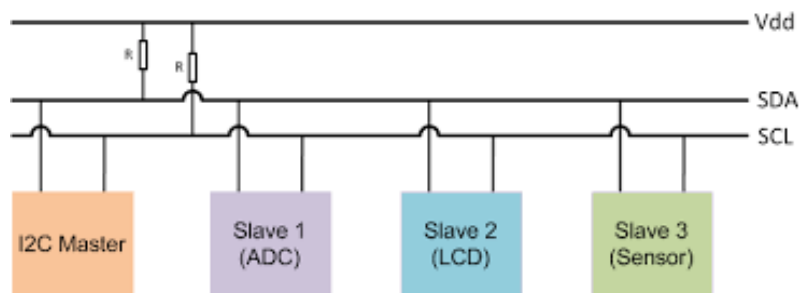


Hình 2.7. Giao thức one-wire

Về nguyên lý hoạt động, giao thức này tuân thủ một quy trình bắt tay nghiêm ngặt về mặt thời gian. Quá trình truyền tin bắt đầu khi vi điều khiển kéo đường dữ liệu xuống mức thấp trong khoảng ít nhất 18ms để gửi tín hiệu "Start". Sau đó, cảm biến sẽ phản hồi bằng một xung thấp và kéo theo chuỗi dữ liệu gồm 40 bit (bao gồm 16 bit độ ẩm, 16 bit nhiệt độ và 8 bit kiểm tra). Mặc dù tốc độ truyền dữ liệu không cao và đòi hỏi vi điều khiển phải quản lý ngặt thời gian chính xác, one-wire là giải pháp tối ưu về mặt kết nối vật lý, giúp tiết kiệm tối đa số lượng chân GPIO cho bo mạch node cảm biến kích thước nhỏ.

2.2.1.2. Giao thức I2C

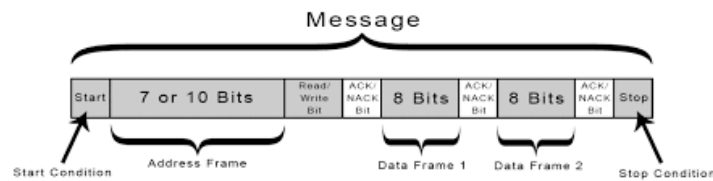
Giao thức I2C thiết lập kết nối giữa vi điều khiển và cảm biến ánh sáng BH1750 dựa trên cấu trúc bus hai dây là SCL (đường xung nhịp) và SDA (đường dữ liệu). Đặc điểm vật lý quan trọng nhất của bus I2C là cấu trúc cực máng hở, nghĩa là các thiết bị chỉ có khả năng kéo mức điện áp xuống thấp mà không thể chủ động đẩy lên mức cao. Do đó, hệ thống bắt buộc phải tích hợp hai điện trở kéo lên nối với nguồn dương để duy trì mức logic 1 mặc định khi đường truyền rảnh, giúp ngăn chặn hiện tượng ngắn mạch và cho phép các thiết bị hoạt động ở các mức điện áp logic khác nhau cùng giao tiếp trên một đường bus.



Hình 2.8. kết nối I2C

Về cấu trúc khung truyền, một gói tin I2C tuân thủ quy trình bắt đầu bằng điều kiện bắt đầu SDA chuyển từ cao xuống thấp khi SCL vẫn ở mức cao, theo sau là 7 bit địa chỉ định danh của cảm biến và 1 bit điều hướng đọc/ghi. Cơ chế kiểm soát lỗi được thực hiện qua bit ACK/NACK tại xung nhịp thứ 9 sau mỗi byte dữ liệu được truyền. Dữ liệu thực tế từ cảm biến sẽ được gửi theo từng byte 8-bit, bắt đầu từ bit trọng số cao nhất, và quá trình truyền chỉ kết thúc khi Master gửi đi điều kiện dừng SDA chuyển từ thấp lên cao khi SCL vẫn ở mức cao, giải phóng đường bus cho các phiên làm việc tiếp

theo.

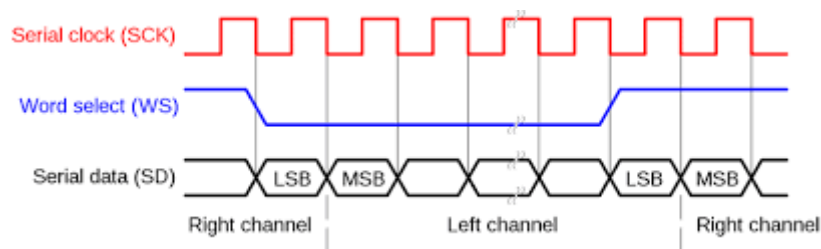


Hình 2.9. Khung bản tin I2C.

2.2.1.3. Giao thức I2S

Đối với microphone INMP441, hệ thống sử dụng giao thức I2S để truyền tải dữ liệu âm thanh kỹ thuật số, đảm bảo độ trung thực cao nhờ cấu trúc tách biệt giữa tín hiệu đồng bộ và dữ liệu. Kết nối vật lý bao gồm ba đường tín hiệu chính: đường SCK cung cấp xung nhịp bit liên tục; đường WS (Word Select) đóng vai trò chọn kênh âm thanh trái hoặc phải; và đường SD mang dữ liệu âm thanh thực tế. Khác với các giao thức điều khiển, I2S cho phép sử dụng cơ chế DMA trên ESP32 để ghi trực tiếp dữ liệu vào bộ nhớ đệm, giúp giảm tải cho CPU trong quá trình thu thập mẫu âm thanh liên tục.

Khung truyền dữ liệu I2S tuân theo chuẩn Philips, với đặc điểm định thời đặc thù là dữ liệu của một mẫu âm thanh luôn bị trễ một chu kỳ xung nhịp so với cạnh biến đổi của tín hiệu WS. Cụ thể, khi đường WS chuyển trạng thái từ thấp lên cao hoặc ngược lại để báo hiệu đổi kênh, bit dữ liệu đầu tiên sẽ không xuất hiện ngay lập tức mà bắt đầu được truyền tại xung nhịp SCK thứ hai. Dữ liệu âm thanh được truyền dưới dạng mã bù hai với độ phân giải 24-bit, đi từ bit trọng số cao nhất đến thấp nhất, đảm bảo bên thu có thể tái tạo chính xác biên độ tín hiệu âm thanh ban đầu.

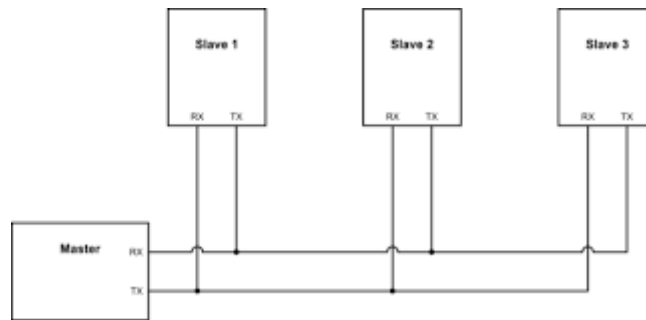


Hình 2.10. Cấu trúc khung bản tin I2S

2.2.1.4. Giao thức UART

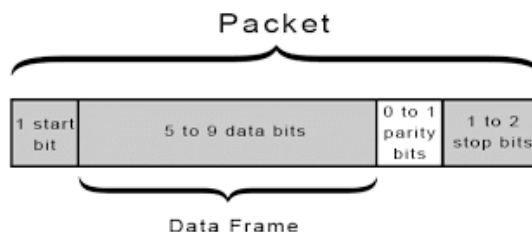
Giao thức UART đóng vai trò là kênh giao tiếp thiết bị không đồng bộ, được sử dụng chủ yếu trong hệ thống để phục vụ quá trình nạp firmware và gỡ lỗi thông qua cổng Serial. Về mặt kết nối vật lý, UART hoạt động theo phương thức song công toàn

phần dựa trên hai đường dây tín hiệu độc lập: đường truyền (TX) và đường nhận (RX). Để thiết lập liên kết, chân TX của thiết bị này phải được đấu nối chéo với chân RX của thiết bị kia và ngược lại, đồng thời hai thiết bị phải có chung mức điện áp tham chiếu (GND). Do đặc thù là giao thức không đồng bộ (không sử dụng đường xung nhịp chung), hai đầu thiết bị bắt buộc phải được cấu hình thống nhất về tốc độ truyền (Baud rate) ví dụ 115200 bps để đảm bảo việc lấy mẫu tín hiệu diễn ra chính xác tại cùng một tần số.



Hình 2.11. Cách kết nối sử dụng giao thức UART

Về cấu trúc khung truyền, dữ liệu UART được đóng gói thành các gói tin tuần tự để đảm bảo tính đồng bộ cho từng byte. Trạng thái mặc định của đường truyền khi rảnh luôn được duy trì ở mức điện áp cao. Quá trình truyền một byte dữ liệu bắt đầu khi bên phát kéo đường tín hiệu xuống mức thấp trong khoảng thời gian của một bit, được gọi là start bit, nhằm đánh thức bên thu và đồng bộ hóa thời gian lấy mẫu. Tiếp theo đó, các data bits thường là 8 bit được truyền đi lần lượt bắt đầu từ bit có trọng số thấp nhất. Tùy thuộc vào cấu hình, một bit kiểm tra chẵn lẻ có thể được thêm vào để phát hiện lỗi truyền. Cuối cùng, khung truyền được khép lại bằng stop bit, trong đó đường tín hiệu được đưa trở lại mức cao trong khoảng thời gian tối thiểu của một hoặc hai bit, đưa đường truyền về trạng thái nghỉ để sẵn sàng cho gói tin kế tiếp.

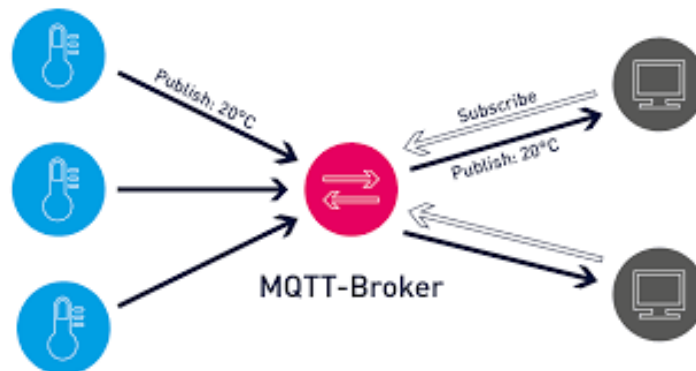


Hình 2.12. Khung dữ liệu của giao thức UART

2.2.2. Giao thức giao tiếp không dây

2.2.2.1. Giao thức MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông điệp được thiết kế cho các ứng dụng Internet of Things (IoT). Đặc điểm cốt lõi của MQTT là một giao thức cực kỳ nhẹ (lightweight), được xây dựng dựa trên mô hình công bố/đăng ký (publish/subscribe). Nó được thiết kế chuyên biệt để hoạt động hiệu quả trên các thiết bị có tài nguyên hạn chế và trên các mạng có băng thông thấp, độ trễ cao hoặc không ổn định. Trong kiến trúc của đề tài, MQTT đóng vai trò là giao thức truyền thông tầng ứng dụng, cho phép hệ thống cân thông minh đồng bộ hóa dữ liệu giao dịch đã xử lý tại biên lên máy chủ trung tâm.



Hình 2.13 Mô hình của giao thức MQTT

Nguyên lý hoạt động của MQTT dựa trên sự tách rời (decoupling) hoàn toàn giữa bên gửi tin nhắn (Publisher) và bên nhận tin nhắn (Subscriber). Thay vì giao tiếp trực tiếp, cả hai thành phần này đều kết nối với một máy chủ trung tâm gọi là Broker (Máy chủ môi giới). Broker chịu trách nhiệm tiếp nhận tất cả các tin nhắn đến, lọc chúng dựa trên chủ đề (topic), và sau đó phân phối các tin nhắn đó đến tất cả các client đã đăng ký nhận chủ đề tương ứng.

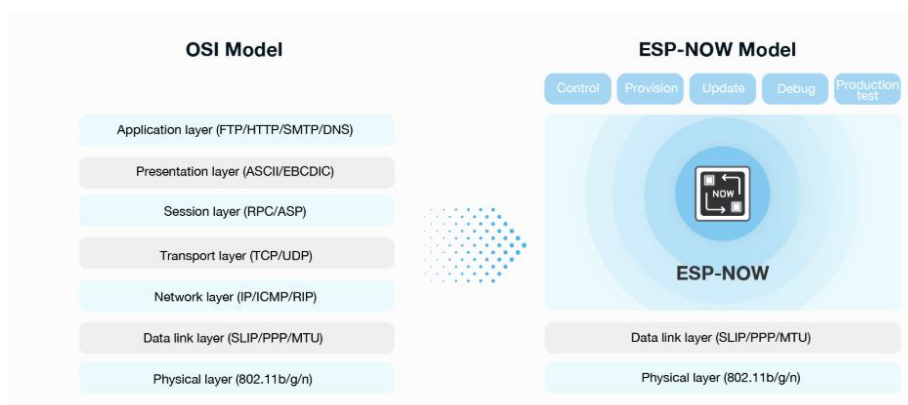
Trong mô hình này, có ba thành phần chính. Publisher (Bên công bố) là client gửi tin nhắn. Trong hệ thống này, vi điều khiển ESP32 chính là một Publisher, có nhiệm vụ công bố dữ liệu về sản phẩm và trọng lượng sau khi xử lý. Subscriber (Bên đăng ký) là client nhận tin nhắn. Trong trường hợp này, Web Server hoặc ứng dụng web thanh toán đóng vai trò là Subscriber, lắng nghe và nhận dữ liệu giao dịch. Broker là máy chủ trung gian quản lý việc chuyển tiếp tin nhắn, vai trò này cũng do Web Server đảm nhiệm.

Cơ chế định tuyến tin nhắn được thực hiện thông qua chủ đề. Topic là một chuỗi ký tự ví dụ: smart_scale/data mà Publisher sử dụng để gắn nhãn cho tin nhắn của mình. Các

Subscriber sẽ đăng ký với Broker về những Topic mà chúng quan tâm. Khi ESP32-S3 (Publisher) gửi một tin nhắn đến một Topic cụ thể, Broker sẽ kiểm tra danh sách Subscriber của Topic đó và ngay lập tức chuyển tiếp tin nhắn đến tất cả các client đã đăng ký. Cơ chế này cho phép hệ thống hoạt động linh hoạt, dễ dàng mở rộng, và đảm bảo việc đồng bộ hóa dữ liệu từ thiết bị biên lên máy chủ được thực hiện một cách hiệu quả và đáng tin cậy.

2.2.2.2. Giao thức ESP-NOW

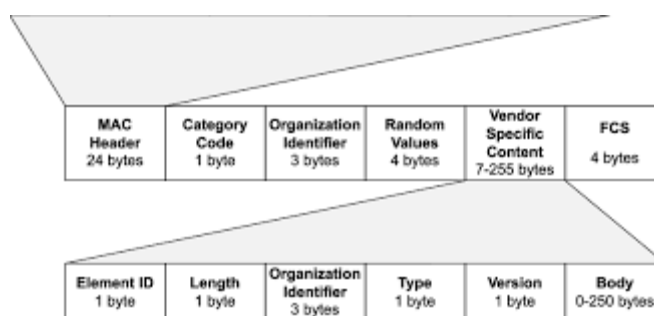
Trong khi MQTT đóng vai trò là xương sống cho việc giao tiếp qua môi trường Internet diện rộng, thì ở cấp độ mạng cục bộ, hệ thống yêu cầu một giải pháp truyền dẫn có độ trễ cực thấp và khả năng phản hồi tức thì để xử lý các lệnh điều khiển bằng giọng nói. Để giải quyết bài toán này, giao thức ESP-NOW đã được lựa chọn làm phương thức giao tiếp nòng cốt giữa các thiết bị trong mạng cảm biến. ESP-NOW là một giao thức giao tiếp không dây tầm ngắn, được phát triển độc quyền bởi Espressif Systems, cho phép các thiết bị vi điều khiển trao đổi dữ liệu trực tiếp với nhau mà không cần thông qua một bộ định tuyến trung gian hay thiết lập các kết nối Wi-Fi truyền thống phức tạp. Về bản chất kỹ thuật, ESP-NOW vận hành dựa trên các khung truyền dẫn theo chuẩn IEEE 802.11, cụ thể là các "Vendor Action Frames". Khác với mô hình OSI 7 lớp truyền thông thường thấy trong các giao thức mạng, ESP-NOW hoạt động chủ yếu ở tầng liên kết dữ liệu. Điều này cho phép giảm thiểu đáng kể các dữ liệu dư thừa của các giao thức tầng trên như TCP/IP, từ đó tối ưu hóa tốc độ truyền tin.



Hình 2.14. Giao thức ESP-NOW

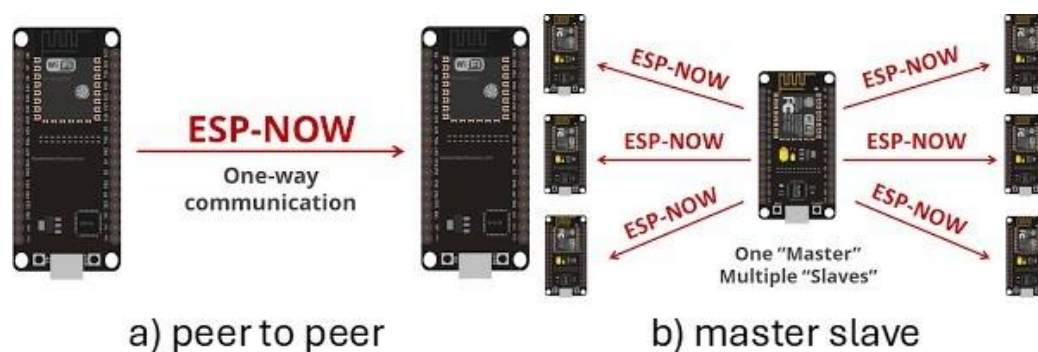
Nguyên lý hoạt động cốt lõi của ESP-NOW là cơ chế truyền thông không kết nối. Trong các giao thức Wi-Fi thông thường, thiết bị cần trải qua quá trình bắt tay và xác

thực với Access Point, tiêu tốn nhiều thời gian và năng lượng. Ngược lại, với ESP-NOW, các thiết bị chỉ cần biết địa chỉ MAC (Media Access Control) của nhau để thiết lập liên kết ngang hàng (Peer-to-Peer). Khi một thiết bị cần gửi dữ liệu, nó sẽ đóng gói thông tin vào một gói tin có cấu trúc xác định và gửi trực tiếp đến địa chỉ MAC đích. Mỗi gói tin ESP-NOW có kích thước tải trọng tối đa là 250 bytes. Mặc dù con số này nhỏ hơn nhiều so với các gói tin HTTP hay MQTT, nhưng nó là hoàn toàn đủ và lý tưởng cho các ứng dụng IoT điều khiển, nơi dữ liệu chủ yếu là các tín hiệu bật/tắt, trạng thái cảm biến hoặc các chuỗi lệnh ngắn. Để đảm bảo tính toàn vẹn của dữ liệu trong môi trường truyền dẫn không dây nhiều nhiễu, ESP-NOW tích hợp sẵn cơ chế ACK (Acknowledgement). Khi cấu hình, thiết bị gửi có thể nhận được phản hồi xác nhận từ thiết bị nhận để biết gói tin đã được chuyển đi thành công hay chưa, giúp tăng độ tin cậy của hệ thống.



Hình 2.15. Khung bản tin của giao thức ESP-NOW

Một trong những ưu điểm khiến ESP-NOW phù hợp với đề tài nhà thông minh là tính linh hoạt trong cấu trúc mạng. Giao thức này hỗ trợ đa dạng các chế độ kết nối, bao gồm giao tiếp một - một (peer-to-peer) và quan trọng hơn là giao tiếp một - nhiều (One-to-Many). Trong kiến trúc của đề tài, mô hình "một - nhiều" được áp dụng triệt để: một thiết bị xử lý trung tâm đóng vai trò là trạm thu nhận, trong khi các nút cảm biến và thiết bị chấp hành đóng vai trò là các trạm gửi. Ngoài ra, ESP-NOW còn hỗ trợ tính năng mã hóa dữ liệu sử dụng công nghệ CCMP (Counter Mode with CBC-MAC Protocol), tuân theo tiêu chuẩn IEEE 802.11i. Tính năng này đóng vai trò quan trọng trong việc bảo mật hệ thống nhà thông minh, ngăn chặn các cuộc tấn công nghe lén hoặc giả mạo lệnh điều khiển trong phạm vi sóng vô tuyến cục bộ.



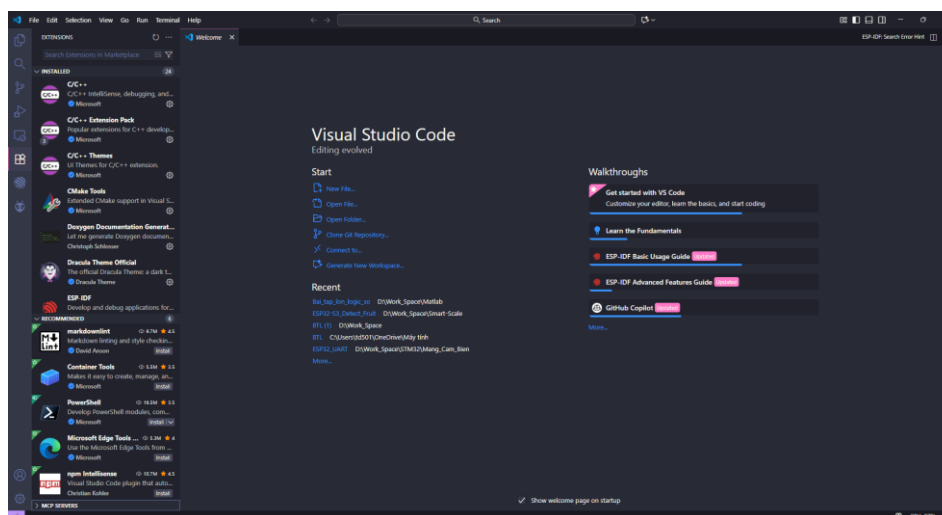
Hình 2.16. Cấu trúc mạng

So sánh với các chuẩn giao tiếp không dây khác như Bluetooth Low Energy hay Zigbee, ESP-NOW thể hiện sự vượt trội về độ trễ và khả năng xuyên tường. Trong các thử nghiệm thực tế, thời gian để một thiết bị ESP32 thức dậy từ chế độ ngủ sâu, gửi một gói tin ESP-NOW và quay lại chế độ ngủ chỉ diễn ra trong vài mili-giây. Đặc tính thời gian thực này là yếu tố then chốt cho hệ thống điều khiển giọng nói, đảm bảo rằng khi người dùng phát lệnh, đèn hoặc thiết bị điện sẽ phản hồi gần như ngay lập tức mà không có độ trễ đáng kể.

Một ưu điểm kỹ thuật khác không thể bỏ qua là khả năng hoạt động song song. Vì điều khiển ESP32 cho phép chạy song song giao thức ESP-NOW và kết nối Wi-Fi. Nhờ đó, thiết bị trung tâm trong hệ thống của chúng tôi có thể thực hiện nhiệm vụ kép: vừa duy trì liên kết thời gian thực với các cảm biến trong nhà qua ESP-NOW, vừa giữ kết nối Internet qua Wi-Fi để đồng bộ trạng thái lên MQTT Broker. Sự kết hợp này tạo nên một kiến trúc AIoT liền mạch, tận dụng sức mạnh xử lý tại biên của ESP-NOW và khả năng lưu trữ, quản lý từ xa của Cloud qua MQTT.

2.3. Phần mềm sử dụng

2.3.1. Công cụ lập trình



Hình 2.17 Phần mềm visual studio code

Visual Studio Code được lựa chọn là môi trường phát triển tích hợp chính để xây dựng, quản lý và biên dịch toàn bộ dự án. VSCode là một trình soạn thảo mã nguồn gọn nhẹ, đa nền tảng do Microsoft phát triển, nhưng sức mạnh thực sự của nó nằm ở hệ sinh thái tiện ích mở rộng (extensions) khổng lồ. Nhờ các tiện ích này, VSCode được chuyển đổi từ một trình soạn thảo đơn thuần thành một IDE mạnh mẽ, hỗ trợ đa dạng ngôn ngữ và nền tảng.

Đối với việc phát triển trên nền tảng ESP32-S3, tiện ích mở rộng PlatformIO IDE tích hợp trong VSCode đã được lựa chọn làm môi trường phát triển chính. PlatformIO là một hệ sinh thái phát triển IoT đa nền tảng, giúp đơn giản hóa và tự động hóa đáng kể các quy trình phức tạp như quản lý thư viện phụ thuộc và cấu hình board mạch. Một trong những ưu điểm lớn nhất của PlatformIO là nó cho phép tích hợp và sử dụng liền mạch bộ công cụ ESP-IDF (Espressif IoT Development Framework) chính thức, giúp nhà phát triển vừa có được sự tiện lợi của PlatformIO, vừa truy cập sâu được vào các tính năng phần cứng và API cấp thấp của vi điều khiển.

Sự kết hợp mạnh mẽ này cho phép thực hiện toàn bộ quy trình phát triển một cách hiệu quả ngay trong VSCode từ việc soạn thảo mã với các tính năng gợi ý thông minh, quản lý cấu hình dự án với file platformio.ini trực quan, biên dịch mã nguồn C/C++, đến việc nạp firmware đã biên dịch xuống vi điều khiển và theo dõi đầu ra qua cổng serial. Việc sử dụng VSCode cùng PlatformIO và ESP-IDF giúp tối ưu hóa đáng kể quy trình làm việc và tăng tốc độ phát triển sản phẩm.

2.3.2. Các framework phát triển hệ thống

Để giải quyết bài toán phức tạp về xử lý tín hiệu âm thanh thời gian thực và triển khai các mô hình học sâu trên thiết bị nhúng có tài nguyên giới hạn, việc lựa chọn nền tảng phát triển phần mềm đóng vai trò then chốt quyết định hiệu năng của toàn bộ hệ thống. Thay vì sử dụng các môi trường lập trình bậc cao với nhiều lớp trừu tượng, hệ thống trong đề tài được xây dựng dựa trên sự kết hợp chặt chẽ giữa khung phát triển hệ thống ESP-IDF và framework trí tuệ nhân tạo chuyên dụng ESP-SR. Sự kết hợp này không chỉ đảm bảo khả năng kiểm soát triệt để các tài nguyên phần cứng cấp thấp mà còn tối ưu hóa hiệu suất xử lý song song, đáp ứng yêu cầu khắt khe về độ trễ trong các ứng dụng điều khiển bằng giọng nói.

2.3.2.1. Khung phát triển hệ thống ESP-IDF

Một đặc điểm kiến trúc nổi bật của ESP-IDF là việc tổ chức mã nguồn theo hướng mô-đun hóa chặt chẽ thông qua cơ chế quản lý thành phần. Thay vì xây dựng ứng dụng theo cấu trúc nguyên khối, toàn bộ dự án được chia nhỏ thành các khối chức năng độc lập được gọi là "Components". Các thành phần này bao gồm nhân hệ điều hành, trình điều khiển ngoại vi, ngăn xếp giao thức mạng, và cả các thư viện mở rộng như ESP-SR hay MQTT. Hệ thống biên dịch dựa trên CMake và Ninja đóng vai trò liên kết tự động, cho phép lập trình viên dễ dàng cấu hình, tích hợp và tái sử dụng mã nguồn giữa các dự án khác nhau. Cách tiếp cận này giúp cô lập các chức năng, đơn giản hóa quá trình bảo trì và cho phép tối ưu hóa dung lượng Firmware cuối cùng bằng cách chỉ biên dịch những thành phần thực sự cần thiết cho hệ thống.



Hình 2.18. Framework ESP-IDF

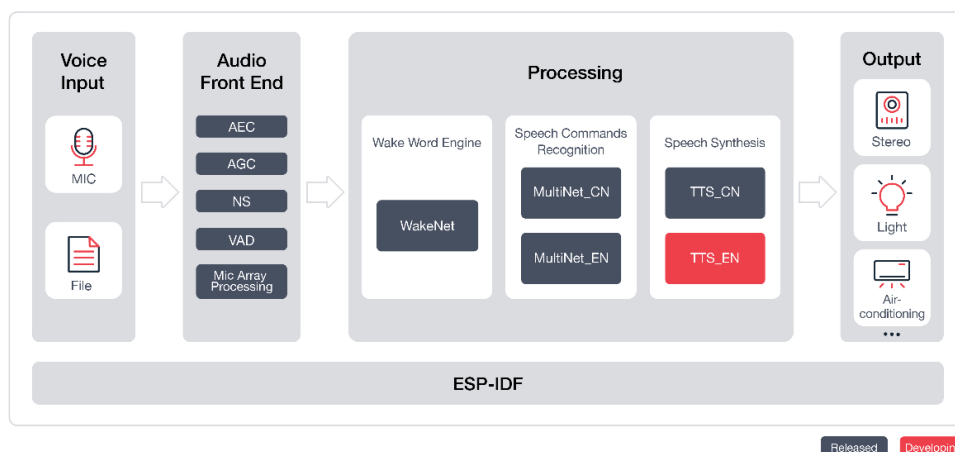
Về phương pháp lập trình, việc phát triển ứng dụng trên ESP-IDF đòi hỏi sự chuyển dịch từ tư duy lập trình tuần tự sang mô hình lập trình đa luồng dựa trên hệ điều hành thời gian thực FreeRTOS. Mã nguồn hệ thống không vận hành trong một vòng lặp vô hạn đơn lẻ mà được tổ chức thành các tác vụ (Tasks) riêng biệt chạy song song. Lập trình viên tương tác với hệ thống thông qua tập hợp các giao diện lập trình ứng dụng

chuẩn của FreeRTOS để khởi tạo tác vụ, quản lý hàng đợi thông điệp để truyền dữ liệu giữa các luồng, và sử dụng Semaphore hoặc Mutex để đồng bộ hóa tài nguyên chia sẻ. Bên cạnh đó, để điều khiển phần cứng, ESP-IDF cung cấp các API trình điều khiển đóng vai trò như một lớp trừu tượng phần cứng (HAL). Các API này cho phép mã ứng dụng tương tác trực tiếp và an toàn với các ngoại vi phức tạp như I2S, Wi-Fi hay Timer thông qua các hàm C tiêu chuẩn mà không cần thao tác thủ công vào từng thanh ghi, giúp cân bằng giữa hiệu năng xử lý cấp thấp và tốc độ phát triển ứng dụng.

2.3.2.2. Framework nhận dạng giọng nói ESP-SR

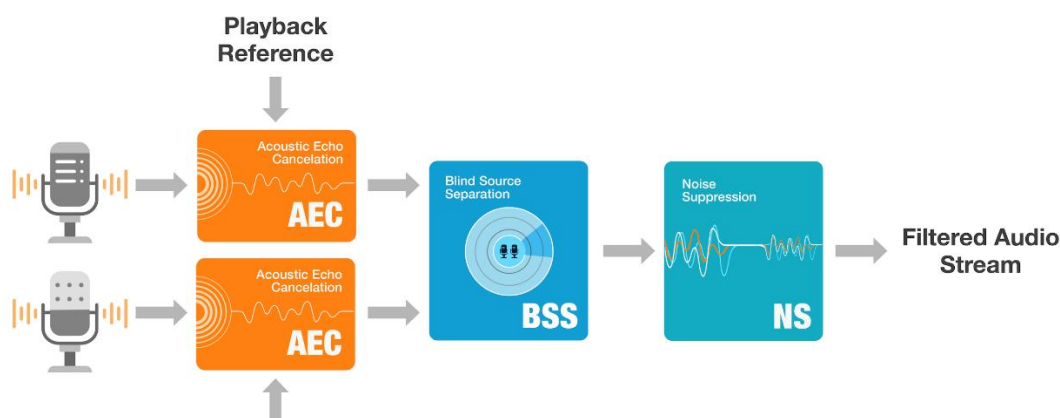
Để hiện thực hóa tính năng điều khiển bằng giọng nói sử dụng mô hình mạng nơron tích chập, đề tài tích hợp ESP-SR (Espressif Speech Recognition Framework). Đây là một giải pháp AIoT toàn diện được tối ưu hóa riêng cho kiến trúc phần cứng của Espressif, cung cấp một đường ống xử lý tín hiệu (signal processing pipeline) khép kín từ khâu thu thập tín hiệu đầu vào đến khâu đưa ra lệnh điều khiển, thay vì chỉ là các thư viện xử lý rời rạc.

Quy trình xử lý bắt đầu tại khối tiền xử lý âm thanh (Audio Front-End - AFE). Tại đây, tín hiệu âm thanh thô thu được từ Microphone qua giao tiếp I2S sẽ được làm sạch thông qua hàng loạt thuật toán xử lý tín hiệu số tiên tiến. Khối AFE thực hiện loại bỏ tiếng vọng âm thanh để triệt tiêu các phản hồi từ loa hệ thống, đồng thời áp dụng thuật toán phân tách nguồn âm và giảm nhiễu để lọc bỏ tạp âm môi trường, làm nổi bật giọng nói người dùng. Bên cạnh đó, thuật toán phát hiện hoạt động giọng nói cũng được tích hợp để hệ thống chỉ kích hoạt các tiến trình xử lý cao cấp khi thực sự phát hiện tiếng người, qua đó tối ưu hóa năng lượng tiêu thụ.



Hình 2.19. kiến trúc tổng quan hệ thống nhận dạng giọng nói

Sau khi tín hiệu đã được làm sạch, luồng dữ liệu được chuyển tiếp đến WakeNet, thành phần đóng vai trò là công cụ phát hiện từ khóa đánh thức (Wake Word Engine). WakeNet sử dụng một mô hình mạng nơ-ron sâu (Deep Neural Network) đã được lượng tử hóa để giảm kích thước, cho phép chạy thường trực trên bộ nhớ đệm của vi điều khiển. Nhiệm vụ của khối này là liên tục lắng nghe môi trường để phát hiện từ khóa định trước ví dụ: "Hi ESP". Chỉ khi WakeNet xác nhận từ khóa với độ tin cậy vượt ngưỡng cho phép, hệ thống mới chuyển sang trạng thái nhận lệnh tích cực, giúp ngăn chặn việc kích hoạt sai do các âm thanh ngẫu nhiên.



Hình 2.20. Quy trình xử lý tín hiệu khử nhiễu tại khối Audio Front-End

Cuối cùng, các câu lệnh điều khiển chi tiết được xử lý bởi MultiNet, mô hình nhận dạng giọng nói đa lệnh (Speech Command Recognition). MultiNet vận hành dựa trên kiến trúc mạng nơ-ron tích chập kết hợp hồi quy (CRNN), có khả năng trích xuất các đặc trưng âm thanh theo thời gian và so sánh chúng với tập lệnh đã được huấn luyện.

Điểm ưu việt của MultiNet trong khuôn khổ framework ESP-SR là khả năng định nghĩa và tùy biến các câu lệnh điều khiển như "Bật đèn", "Tắt quạt" một cách linh hoạt thông qua cấu hình phần mềm mà không yêu cầu quá trình huấn luyện lại phức tạp. Sự kết hợp giữa ESP-IDF và ESP-SR tạo nên một nền tảng vững chắc, cho phép hệ thống hoạt động độc lập tại biên với độ tin cậy cao và tốc độ phản hồi tức thì.

2.4. Giải pháp tích hợp mô hình học sâu vào thiết bị nhúng

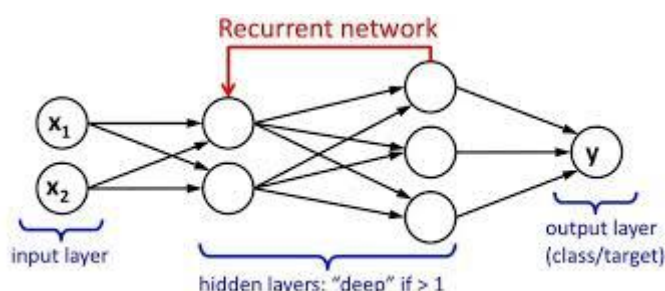
Để giải quyết bài toán nhận dạng giọng nói và điều khiển thiết bị theo thời gian thực ngay trên thiết bị nhúng, việc lựa chọn kiến trúc mô hình học máy là yếu tố then chốt, quyết định tính khả thi và độ ổn định của toàn bộ hệ thống. Các phương pháp tiếp cận sơ khai trong lĩnh vực xử lý tiếng nói thường dựa trên các kỹ thuật thống kê truyền thống như mô hình Markov ẩn (Hidden Markov Models - HMM) kết hợp với mô hình hỗn hợp Gaussian (GMM). Các hệ thống này hoạt động bằng cách khớp các đặc trưng âm thanh thủ công (như MFCC) với các trạng thái xác suất để suy ra từ ngữ. Tuy nhiên, cách tiếp cận này bộc lộ hạn chế lớn khi áp dụng vào môi trường nhà thông minh thực tế, bởi chúng đòi hỏi quá trình căn chỉnh thời gian phức tạp và rất nhạy cảm với tiếng ồn môi trường. Một mô hình thống kê được tối ưu cho phòng thu tĩnh lặng sẽ hoạt động kém hiệu quả khi đối mặt với các tạp âm sinh hoạt thường ngày như tiếng quạt, tiếng TV hay tiếng nói chuyện nền. Hơn nữa, các giải pháp này thường gặp khó khăn trong việc mở rộng tập lệnh điều khiển linh hoạt mà không cần tái cấu trúc lại toàn bộ hệ thống.

Do đó, các giải pháp dựa trên mạng nơ-ron được ưu tiên thay thế, đặc biệt là kiến trúc Mạng nơ-ron tích chập kết hợp hồi quy. Nếu như CNN là kiến trúc chuyên biệt để trích xuất các đặc trưng phổ tần số - thời gian từ tín hiệu âm thanh đầu vào, thì mạng hồi quy (RNN) lại vượt trội trong việc nắm bắt mối quan hệ ngữ cảnh của chuỗi âm thanh theo thời gian. Sự kết hợp này cho phép mô hình tự động học hỏi các đặc trưng phức tạp trực tiếp từ dữ liệu âm thanh thô mà không cần các bước trích xuất đặc trưng thủ công cứng nhắc. Khả năng này biến CRNN trở thành giải pháp tiên tiến và phù hợp nhất cho bài toán nhận dạng từ khóa và các câu lệnh điều khiển rời rạc trong đề tài.

Mặc dù vậy, việc triển khai các mô hình học sâu cho giọng nói cũng đặt ra nhiều thách thức thực tế. Các hệ thống nhận dạng giọng nói hiện đại trên máy chủ như BERT hay Conformer thường có kích thước khổng lồ, yêu cầu hàng gigabyte bộ nhớ và năng

lực tính toán của các GPU chuyên dụng. Việc nhúng các mô hình này xuống một thiết bị biên có tài nguyên cực kỳ hạn chế như ESP32-S3, với bộ nhớ Flash chỉ vài MB và không có hệ điều hành bậc cao, là bất khả thi. Thách thức đặt ra là cần một giải pháp vừa đảm bảo độ chính xác chấp nhận được, vừa tối ưu hóa triệt để về mặt kích thước và năng lượng tiêu thụ.

Đây là lúc các kỹ thuật thuộc lĩnh vực TinyML (Tiny Machine Learning) trở thành chìa khóa giải quyết vấn đề. Thay vì sử dụng các mạng nơ-ron tổng quát, đề tài lựa chọn tiếp cận theo hướng tối ưu hóa phần cứng với framework ESP-SR. Bằng cách sử dụng kỹ thuật lượng tử hóa (Quantization) để chuyển đổi trọng số mô hình từ dạng số thực 32-bit xuống dạng số nguyên 8-bit, kết hợp với cấu trúc mạng nhẹ như WakeNet và MultiNet được thiết kế riêng cho tập lệnh của vi điều khiển, hệ thống có thể giảm đáng kể khối lượng tính toán. Chính sự cân bằng vượt trội giữa khả năng nhận dạng thời gian thực và yêu cầu tài nguyên thấp này là lý do mô hình CRNN trên nền tảng ESP-SR được lựa chọn làm giải pháp lõi cho hệ thống nhà thông minh điều khiển bằng giọng nói, và quy trình triển khai chi tiết sẽ được phân tích sâu ở chương sau.



Hình 2.21. Kiến trúc tổng quát của mạng CRNN

2.5. Kết luận chương 2

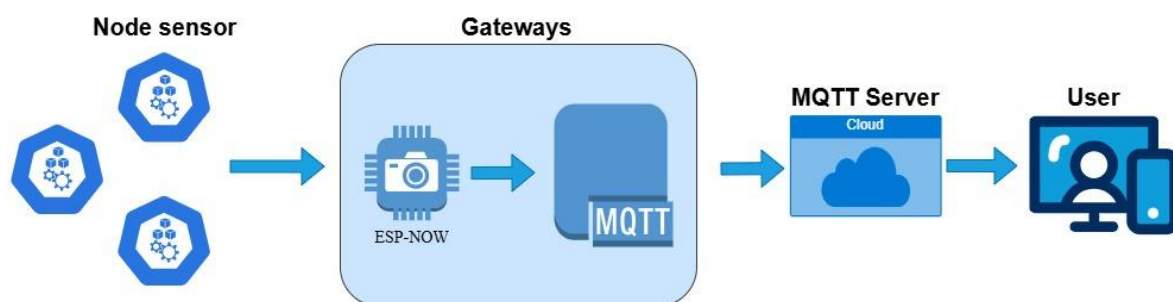
Chương 2 đã thiết lập nền tảng công nghệ vững chắc cho đề tài thông qua việc lựa chọn vi điều khiển ESP32-S3 với kiến trúc lõi kép hỗ trợ AI làm hạt nhân xử lý, kết hợp cùng mô hình truyền thông lai giữa giao thức mạng điều khiển cục bộ thời gian thực và cho quản lý từ xa. Đặc biệt, việc tích hợp khung phát triển hệ thống ESP-IDF trên nền tảng FreeRTOS song song với framework trí tuệ nhân tạo chuyên dụng ESP-SR đã cung cấp một giải pháp toàn diện để quản lý tài nguyên đa luồng và triển khai đường ống xử lý tín hiệu giọng nói khép kín ngay tại biên, tạo cơ sở lý thuyết và kỹ thuật quan trọng để tiến hành thiết kế chi tiết mô hình và thuật toán hệ thống trong Chương 3.

CHƯƠNG 3: ĐỀ XUẤT PHÁT TRIỂN HỆ NHÀ THÔNG MINH GIÁM SÁT VÀ ĐIỀU KHIỂN BẰNG GIỌNG NÓI

Trên cơ sở các nền tảng công nghệ phần cứng và giải pháp phần mềm đã được phân tích tại Chương 2, Chương 3 tập trung đề xuất và hiện thực hóa kiến trúc hệ thống nhà thông minh AIoT toàn diện theo mô hình phân tán. Hệ thống được thiết kế dựa trên cấu trúc mạng hình sao, trong đó nhiệm vụ xử lý được phân tách rõ ràng để tối ưu hiệu năng: vi điều khiển trung tâm đóng vai trò GateWay làm cầu nối chuyển đổi giao thức, quản lý luồng dữ liệu thời gian thực và đồng bộ giám sát từ xa; trong khi đó, các tác vụ tính toán phức tạp được đẩy xuống các thiết bị biên. Từ thiết kế tổng quát này, nội dung chương sẽ đi sâu vào chi tiết sơ đồ nguyên lý phần cứng, lưu đồ thuật toán điều khiển và tiến hành các kịch bản thử nghiệm thực tế nhằm đánh giá độ chính xác cũng như tính ổn định của hệ thống trong điều kiện vận hành cụ thể.

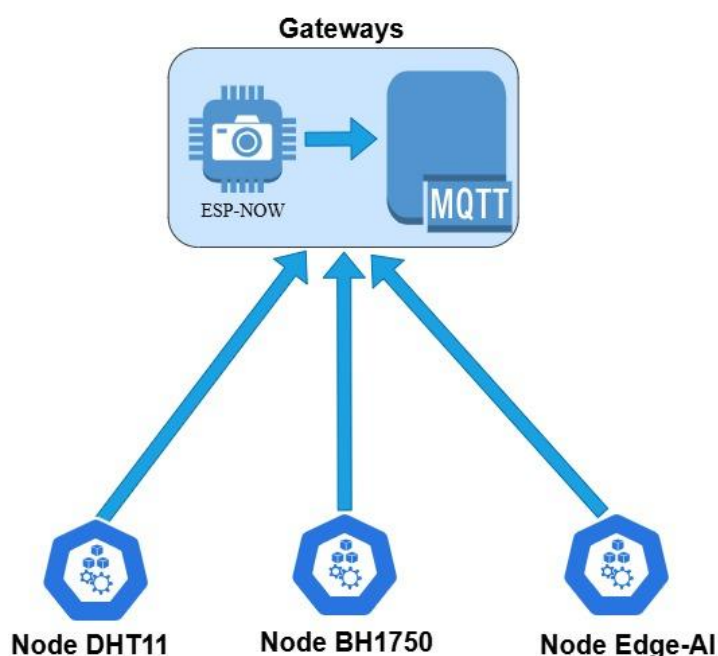
3.1. Mô hình đề xuất

Để giải quyết bài toán xây dựng hệ thống nhà thông minh AIoT đảm bảo sự cân bằng giữa khả năng xử lý tại biên và quản lý từ xa, đề tài đề xuất một kiến trúc hệ thống tổng quát gồm ba tầng chức năng chính như minh họa trong Hình 3.1. Tầng thấp nhất là lớp thiết bị biên, nơi các cảm biến thu thập dữ liệu môi trường và các mô-đun xử lý giọng nói hoạt động. Tầng giữa là lớp trung chuyển, đóng vai trò cầu nối liên kết mạng cục bộ với môi trường Internet. Tầng cao nhất là lớp ứng dụng và lưu trữ, nơi dữ liệu được tổng hợp trên máy chủ MQTT để phục vụ việc giám sát và điều khiển từ xa qua thiết bị di động. Luồng dữ liệu trong hệ thống được thiết kế khép kín: dữ liệu từ các Node được truyền về GateWay, đồng bộ lên Cloud và hiển thị tới người dùng; ngược lại, lệnh điều khiển từ người dùng sẽ đi theo chiều ngược lại để tác động xuống thiết bị cuối.



Hình 3.1 Mô hình đề xuất hệ thống

Ở phía bên trái của sơ đồ là Tầng thu thập và chấp hành, bao gồm mạng lưới các Node Sensor được bố trí tại các vị trí khác nhau trong không gian nhà ở. Các nút này chịu trách nhiệm thu thập dữ liệu môi trường như nhiệt độ, độ ẩm, ánh sáng hoặc điều khiển các thiết bị điện. Để giải quyết bài toán về năng lượng và độ trễ, kết nối giữa các Node Sensor và Gateway không sử dụng Wi-Fi truyền thống mà thông qua giao thức ESP-NOW. Đây là một quyết định thiết kế chiến lược, bởi ESP-NOW cho phép các thiết bị giao tiếp ngang hàng với gói tin nhỏ gọn, giúp giảm thiểu thời gian bắt tay và tiết kiệm năng lượng tối đa cho các cảm biến dùng pin. Mũi tên hướng từ Node Sensor về Gateway trong hình vẽ biểu thị luồng dữ liệu thời gian thực được truyền tải liên tục về trung tâm xử lý với độ ổn định cao.



Hình 3.2. Sơ đồ kết nối các node cảm biến

Trong cấu trúc hình sao này, các nút vệ tinh được phân chia theo chức năng chuyên biệt để tối ưu hiệu năng. Nhóm thứ nhất là các Node cảm biến môi trường như Node DHT11 đo nhiệt độ/độ ẩm, Node BH1750 đo ánh sáng, chúng hoạt động ở chế độ tiết kiệm năng lượng và chỉ thức dậy để gửi dữ liệu định kỳ. Nhóm thứ hai là Node Edge-AI, đây là thành phần tích hợp trí tuệ nhân tạo để xử lý giọng nói. Node Edge-AI hoạt động độc lập, trực tiếp thu nhận và giải mã lệnh điều khiển bằng mô hình CNN, sau đó gửi yêu cầu tới Gateway. Gateway khi đó thực hiện chức năng của một bộ chuyển

đổi giao thức: nó tổng hợp toàn bộ tín hiệu từ các nhánh của mạng ESP-NOW, đóng gói lại và chuyển tiếp lên đám mây thông qua giao thức MQTT. Cách tổ chức này giúp tách biệt hoàn toàn việc xử lý nghiệp vụ tại các Node và việc truyền dẫn tại Gateway, đảm bảo hệ thống vận hành trơn tru và dễ dàng mở rộng thêm các node mới trong tương lai. Phía bên phải sơ đồ là tầng ứng dụng giám sát, nơi dữ liệu rời khỏi mạng cục bộ để đi vào môi trường mạng diện rộng. Sau khi Gateway đóng gói dữ liệu vào các bản tin MQTT, chúng được truyền tới MQTT Server (Broker) trên nền tảng đám mây. MQTT Server đóng vai trò là trung gian phân phối tin nhắn, đảm bảo tính toàn vẹn của dữ liệu ngay cả trong điều kiện đường truyền mạng không ổn định. Cuối cùng, người dùng thông qua giao diện web có thể đăng ký vào các chủ đề trên Broker để nhận cập nhật trạng thái ngôi nhà theo thời gian thực hoặc gửi các lệnh điều khiển từ xa ngược trở lại Gateway. Sự kết hợp chặt chẽ giữa xử lý AI tại biên và quản lý qua đám mây tạo nên một hệ thống khép kín, tối ưu hóa trải nghiệm người dùng cả khi ở nhà lẫn khi đi vắng.

3.2. Thiết kế và xây dựng Node Edge-AI xử lý giọng nói

Trong kiến trúc hình sao đã đề xuất, Node Edge-AI đóng vai trò là thành phần xử lý tại biên. Thay vì chỉ thu thập dữ liệu thụ động, node này chịu trách nhiệm thu nhận tín hiệu âm thanh, tiền xử lý khử nhiễu và thực hiện suy luận mô hình học sâu để giải mã câu lệnh người dùng thành tín hiệu điều khiển số. Phần này sẽ trình bày chi tiết về giải pháp phần mềm lõi được sử dụng để hiện thực hóa chức năng này trên vi điều khiển ESP32-S3.

3.2.1. Đường ống xử lý tín hiệu

Hệ thống sử dụng thư viện ESP-SR hoạt động như một đường ống xử lý dữ liệu khép kín. Việc cấu hình đúng các tham số cho từng khối chức năng trong đường ống là yếu tố quyết định đến độ nhạy và độ chính xác của hệ thống trong môi trường thực tế.

3.2.1.1. khối tiền xử lý âm thanh

Khối AFE là cửa ngõ đầu tiên tiếp nhận dữ liệu thô từ microphone I2S. Trong đề tài này, AFE được khởi tạo và cấu hình thông qua API `esp_afe_sr_open()` với các thông số tối ưu hóa cho mô hình phát hiện từ khóa:

- Chế độ khử tiếng vọng (AEC - Acoustic Echo Cancellation): Được kích hoạt (Enable). Đây là tính năng bắt buộc cho các thiết bị điều khiển bằng giọng nói có loa

phản hồi. AEC sử dụng một bộ lọc thích nghi để so sánh tín hiệu thu được từ microphone với tín hiệu tham chiếu đang phát ra loa, từ đó loại bỏ thành phần âm thanh của chính thiết bị, ngăn chặn hiện tượng tự kích hoạt.

- Chế độ triệt tiêu nhiễu (NS - Noise Suppression): Cấu hình ở mức độ trung bình (Level 3). Mức độ này được lựa chọn dựa trên thực nghiệm để cân bằng giữa việc loại bỏ nhiễu nền như tiếng quạt, điều hòa mà không làm méo dạng tín hiệu giọng nói người dùng, đảm bảo các đặc trưng phổ tần số vẫn được bảo toàn cho khâu nhận dạng phía sau.
- Tự động điều chỉnh độ lợi (AGC - Automatic Gain Control): Được kích hoạt để ổn định biên độ tín hiệu đầu vào. AGC giúp cân bằng âm lượng khi người dùng nói quá nhỏ ở xa hoặc quá to ở gần, giữ cho tín hiệu nằm trong dải động tối ưu của mô hình học sâu.

3.2.1.2. Lựa chọn và cấu hình mô hình WakeNet

Trong kiến trúc của ESP-SR, WakeNet là thành phần hoạt động liên tục để lắng nghe từ khóa kích hoạt. Do đó, yêu cầu đặt ra cho mô hình này là phải đạt được sự cân bằng tối ưu giữa độ chính xác, khả năng chống nhiễu và mức tiêu thụ tài nguyên. Espressif cung cấp nhiều phiên bản WakeNet khác nhau qua các thời kỳ phát triển. Để lựa chọn được mô hình phù hợp nhất cho thiết bị Node Edge-AI chạy trên vi điều khiển ESP32-S3, đề tài đã tiến hành phân tích và so sánh ba phiên bản phổ biến nhất hiện nay là WakeNet 8 và WakeNet 9.

Bảng 3.1. Hiệu năng trên các mô hình WakeNet

Tiêu chí	WakeNet 8 (WN8)	WakeNet9 (WN9_quantized)	Đánh giá
Kiến trúc	DNN truyền thống	DNN + Lượng tử hóa	WN9 hiện đại hơn
Độ chính xác	93% - 94%	> 96%	WN9 chính xác hơn
Khả năng chống nhiễu	Trung bình	Cao	WN9 lọc ồn tốt hơn
Kích thước mô hình	~ 300 - 400 KB	< 200 KB (8-bit)	WN9 nhẹ hơn 50%
Tối ưu phần cứng	Không hỗ trợ đầy đủ	Tối ưu tập lệnh Vector S3	WN9 chạy nhanh hơn

Dựa trên bảng phân tích trên, đề tài quyết định lựa chọn mô hình WakeNet 9 cho hệ

thống vì WN9 cho thấy khả năng vượt trội trong việc xử lý tín hiệu giọng nói ở môi trường thực tế, nơi thường xuyên xuất hiện các loại nhiễu nền như tiếng ồn sinh hoạt hay tiếng vọng âm thanh. Mặc dù có kiến trúc phức tạp hơn, nhưng nhờ áp dụng kỹ thuật lượng tử hóa chuyển đổi trọng số từ số thực sang số nguyên 8-bit, WN9 vẫn đảm bảo kích thước đủ nhỏ để nạp trực tiếp vào bộ nhớ đệm SRAM tốc độ cao của ESP32-S3, giúp giảm thiểu độ trễ xử lý xuống mức thấp nhất.

3.2.1.3. Lựa chọn và cấu hình mô hình MultiNet

Sau khi từ khóa đánh thức được xác nhận, hệ thống chuyển sang giai đoạn phức tạp hơn là giải mã ý định của người dùng. Thành phần chịu trách nhiệm cho tác vụ này là MultiNet. Khác với WakeNet chỉ cần phát hiện sự tồn tại của một từ khóa duy nhất, MultiNet phải có khả năng phân biệt hàng chục, thậm chí hàng trăm câu lệnh khác nhau với độ chính xác cao. Để lựa chọn mô hình phù hợp nhất cho Node Edge-AI, đề tài đã tiến hành khảo sát các phiên bản MultiNet do Espressif cung cấp, tập trung vào hai phiên bản gần nhất là MultiNet 4 và MultiNet 5.

Bảng 3.2. Hiệu năng trên các mô hình MultiNet

Tiêu chí	MultiNet 4 (MN4)	MultiNet 5 (MN5_quantized)
Kiến trúc	CRNN cơ bản.	CRNN + Attention Mechanism (Cơ chế chú ý).
Tính năng thêm lệnh	Phức tạp, hạn chế số lượng lệnh tùy biến.	Dynamic Command Definition: Hỗ trợ thêm/sửa/xóa lệnh linh hoạt qua chuỗi âm vị (Phoneme) mà không cần huấn luyện lại.
Kích thước mô hình	Lớn (> 1.5 MB)	Tối ưu hóa (< 1 MB) nhờ lượng tử hóa, tiết kiệm không gian lưu trữ.
Tốc độ suy luận	Trung bình.	Nhanh, tối ưu hóa cho tập lệnh vector của ESP32-S3.
Hỗ trợ ngôn ngữ	Tiếng Anh/Trung.	Tiếng Anh/Trung (Độ chính xác ngữ âm tốt hơn).

Dựa trên các tiêu chí trên, đề tài quyết định lựa chọn mô hình MultiNet 5 (MN5_quantized) phiên bản tiếng Anh vì MultiNet 5 sử dụng kiến trúc CRNN kết hợp với cơ chế Attention. Trong đó, lớp CNN đóng vai trò trích xuất các đặc trưng không gian và phổ tần số của tín hiệu âm thanh đầu vào. Lớp RNN xử lý chuỗi thời gian để nắm bắt ngữ cảnh, và cơ chế Attention giúp mô hình tập trung vào các đoạn tín hiệu

quan trọng nhất, bỏ qua các đoạn nhiễu hoặc khoảng lặng. Sự kết hợp này mang lại khả năng nhận dạng vượt trội so với các kiến trúc cũ. Ưu điểm lớn nhất của MN5 là khả năng định nghĩa lệnh động. Điều này cho phép hệ thống dễ dàng mở rộng thêm các lệnh điều khiển mới ví dụ: thêm lệnh "Turn on fan" chỉ bằng cách khai báo chuỗi âm vị, giúp giảm thiểu đáng kể thời gian phát triển và triển khai sản phẩm.

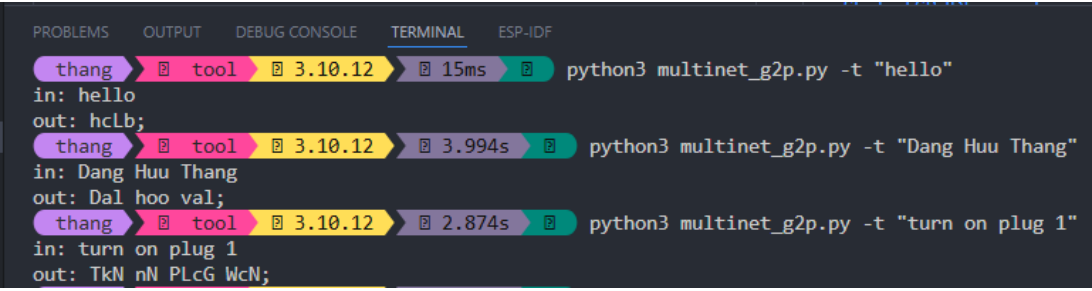
Để đảm bảo trải nghiệm người dùng mượt mà và tối ưu hóa năng lượng tiêu thụ cho thiết bị biên, các tham số vận hành của mô hình MultiNet được cấu hình chặt chẽ với thời gian chờ lệnh là 6000ms và ngưỡng nhận dạng ở mức 0.65. Cụ thể, sau khi từ khóa "Hi ESP" được kích hoạt, hệ thống sẽ mở một cửa sổ lắng nghe trong 6 giây, khoảng thời gian được tính toán dựa trên hành vi thực tế đủ để người dùng phát ra một câu lệnh hoàn chỉnh; nếu quá thời hạn này mà không nhận được lệnh hợp lệ, hệ thống sẽ tự động ngắt MultiNet và quay về trạng thái ngủ chờ để giải phóng tài nguyên CPU cũng như tiết kiệm pin. Đồng thời, mức ngưỡng tin cậy 0.65 được lựa chọn nhằm cân bằng giữa độ nhạy để bắt được các biến thể giọng nói đa dạng và khả năng lọc bỏ các tạp âm hoặc câu nói vu vơ không chủ đích, đảm bảo mọi lệnh điều khiển phát âm rõ ràng đều được thực thi ngay lập tức.

3.2.2. Quy trình xây dựng và số hóa tập lệnh điều khiển

Khác với các hệ thống nhận dạng giọng nói truyền thống yêu cầu huấn luyện lại toàn bộ mô hình khi thay đổi tập lệnh, MultiNet hỗ trợ tính năng Định nghĩa lệnh động (Dynamic Command Definition). Tính năng này cho phép thêm, sửa, xóa các câu lệnh điều khiển ngay trong mã nguồn bằng cách ánh xạ chuỗi văn bản sang chuỗi âm vị. Quy trình thực hiện gồm 3 bước chi tiết như sau:

- **Xác định không gian lệnh (Command Space Definition):** Hệ thống được thiết kế để điều khiển các thiết bị điện cơ bản trong nhà. Mỗi hành động cụ thể được gán một mã định danh duy nhất để phục vụ cho logic điều khiển sau này.
- **Chuyển đổi Grapheme-to-Phoneme (G2P):** Mô hình mạng nơ-ron không hiểu trực tiếp các ký tự văn bản như 'A', 'B', 'C'. Nó hoạt động dựa trên các đơn vị âm thanh cơ bản gọi là âm vị. Do đó, cần thực hiện bước chuyển đổi từ câu lệnh văn bản sang chuỗi âm vị chuẩn ARPABET. Đề tài sử dụng công cụ Python g2p_en dựa trên thư viện NLTK để thực hiện việc này. Cơ chế hoạt động: Công cụ G2P phân tích cấu trúc từ vựng tiếng Anh và tra cứu từ điển phát âm để đưa ra chuỗi ký hiệu phát âm

tương ứng.



Hình 3.3. Quá trình chuyển đổi Grapheme-to-Phoneme

- Tích hợp vào cấu trúc dữ liệu Firmware: Sau khi có được chuỗi âm vị, chúng được nạp vào hệ thống thông qua API của ESP-Skainet.

Bảng 3.3. Bảng ánh xạ lệnh điều khiển và chuỗi âm vị thực tế

ID	Lệnh văn bản	Chuỗi âm vị	Hành động mô tả
0	"Turn on plug one"	TkN nN PLcG WcN	Bật nguồn cho ổ cắm số 1
1	"Turn off plug one"	TkN eF PLcG WcN	Ngắt nguồn cho ổ cắm số 1
2	"Turn on plug two"	TkN nN PLcG To	Bật nguồn cho ổ cắm số 2
3	"Turn off plug two"	TkN eF PLcG To	Ngắt nguồn cho ổ cắm số 2
4	"Turn on plug three"	TkN nN PLcG vRm	Bật nguồn cho ổ cắm số 3
5	"Turn off plug three"	TkN eF PLcG vRm	Ngắt nguồn cho ổ cắm số 3

3.3. Kết quả và đánh giá

Sau quá trình nghiên cứu lý thuyết nền tảng và thiết kế chi tiết giải pháp phần mềm, hệ thống nhà thông minh AIoT đã được hiện thực hóa thành các mô-đun phần cứng hoàn chỉnh và đưa vào vận hành thử nghiệm. Mục này sẽ trình bày chi tiết về cấu trúc vật lý của các thiết bị đã chế tạo, giao diện giám sát trung tâm, đồng thời phân tích sâu các kết quả đo lường hiệu năng kỹ thuật trong điều kiện thực tế.

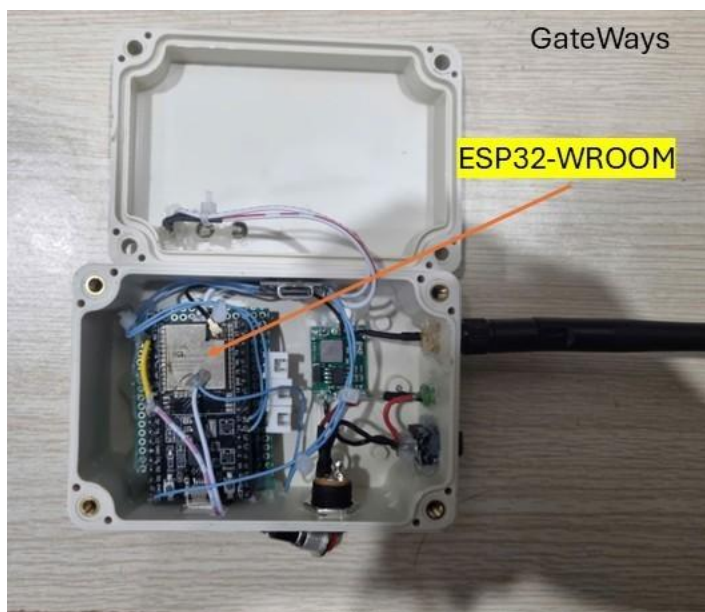
3.3.1. Triển khai hệ thống

Dựa trên kiến trúc mạng hình sao đã đề xuất, hệ thống thực tế được xây dựng bao gồm ba khối chức năng riêng biệt: Gateway trung tâm, Node cảm biến và Node xử lý giọng nói tại biên. Các bo mạch được đóng gói trong các hộp kỹ thuật chuyên dụng hoặc vỏ in 3D thiết kế riêng để đảm bảo tính thẩm mỹ, độ bền cơ học và khả năng tản nhiệt khi hoạt động lâu dài.

3.3.1.1. Khối trung tâm điều phối

Gateway đóng vai trò là trung tâm điều phối và chuyển tiếp dữ liệu của toàn bộ hệ thống truyền thông, thực hiện nhiệm vụ kép: vừa duy trì kết nối Wi-Fi với Internet để đồng bộ dữ liệu lên máy chủ MQTT, vừa lắng nghe liên tục các gói tin ESP-NOW từ các node.

Như thể hiện trong hình ảnh thực tế, Gateway được xây dựng dựa trên module phát triển ESP32-WROOM. Để khắc phục hạn chế về phạm vi phủ sóng của antenna PCB tích hợp sẵn trên mạch (on-board), thiết bị đã được nâng cấp sử dụng antenna ngoài với độ lợi cao. Việc này giúp mở rộng vùng phủ sóng ESP-NOW lên đáng kể, đảm bảo kết nối ổn định xuyên qua các vật cản như tường bê tông trong phạm vi căn hộ.



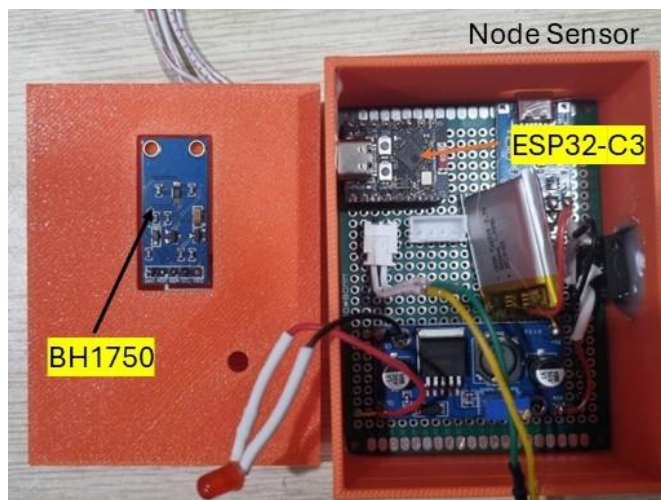
Hình 3.4. GateWay

Toàn bộ mạch điện được đặt trong một hộp kỹ thuật bằng nhựa ABS màu trắng chắc chắn, có khả năng chống bụi và va đập. Về phần nguồn, thay vì phụ thuộc vào nguồn USB của vi điều khiển vốn không ổn định khi tải cao, Gateway sử dụng một mạch giảm áp DC-DC (Buck Converter) chuyên dụng để hạ áp từ nguồn Adapter 12V xuống 5V ổn định cho ESP32. Thiết kế này đảm bảo Gateway có thể hoạt động liên tục 24/7 với độ tin cậy cao nhất, tránh hiện tượng sụt áp gây khởi động lại hệ thống.

3.3.1.2. Node cảm biến không dây

Đối với tầng thu thập dữ liệu, các Node Sensor được thiết kế hướng tới tiêu chí nhỏ gọn và tiết kiệm năng lượng. Phiên bản thực tế sử dụng vi điều khiển ESP32-C3

Super Mini, một biến thể có kích thước vật lý cực nhỏ nhưng vẫn đảm bảo đầy đủ sức mạnh xử lý kiến trúc RISC-V.

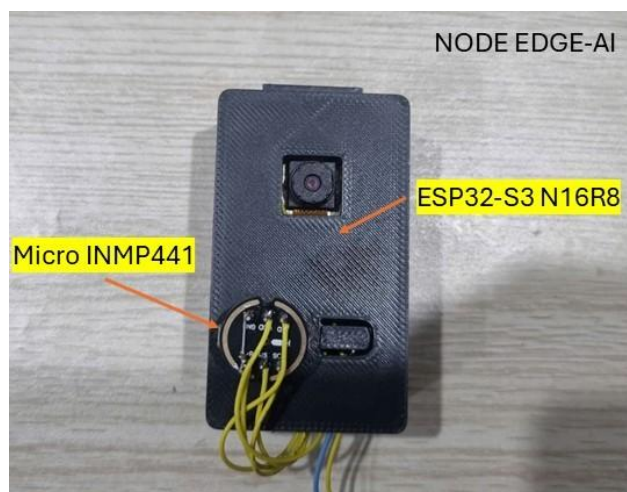


Hình 3.5. Node cảm biến

Node cảm biến được tích hợp module cảm biến ánh sáng kỹ thuật số BH1750 (kết nối qua giao tiếp I2C) và cảm biến nhiệt ẩm DHT11. Toàn bộ hệ thống được cấp nguồn bởi một viên pin Li-Po dung lượng 400mAh. Nhờ sử dụng giao thức ESP-NOW với thời gian truyền tin cực ngắn, thiết bị có thể hoạt động trong thời gian dài chỉ với một lần sạc. Vỏ thiết bị được thiết kế và in 3D bằng nhựa PLA màu cam nổi bật, với các lỗ định vị chính xác cho cảm biến tiếp xúc trực tiếp với môi trường, đảm bảo kết quả đo lường trung thực và nhanh chóng phản ánh sự thay đổi của không khí.

3.3.1.3. Node Edge-AI

Thành phần phức tạp nhất hệ thống là Node Edge-AI, được xây dựng trên nền tảng **ESP32-S3**. Đây là thiết bị chịu trách nhiệm thu nhận và xử lý tín hiệu âm thanh tại chỗ. Vỏ thiết bị được in 3D màu đen với độ hoàn thiện cao, thiết kế tối giản để phù hợp với không gian nội thất phòng khách.

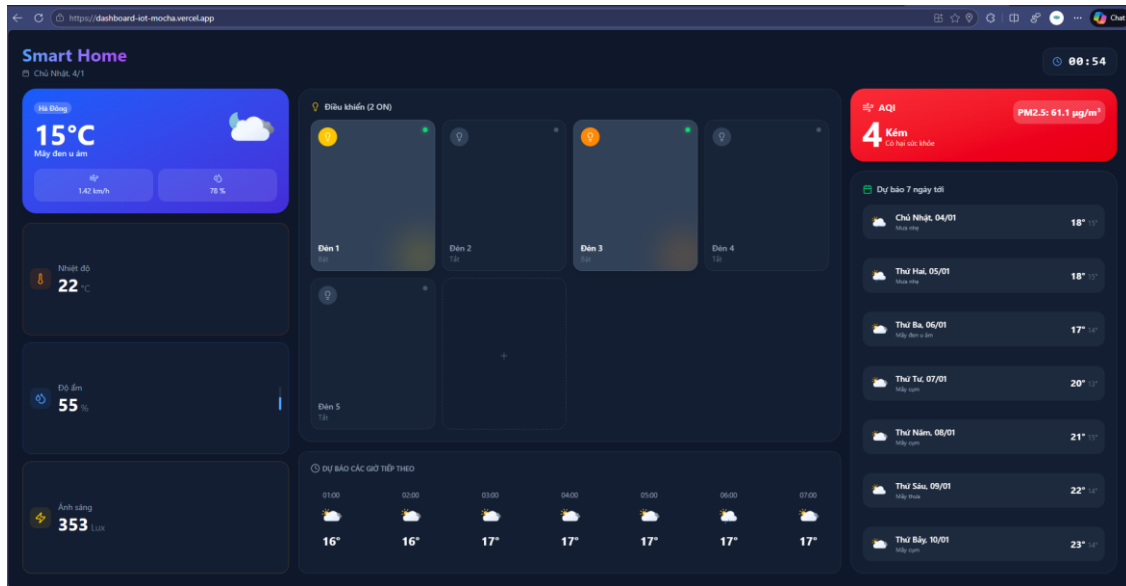


Hình 3.6. Node Edge-AI

Mặt trước của thiết bị bố trí lỗ thu âm cho microphone đa hướng INMP441 và module Camera phục vụ cho các tính năng mở rộng về thị giác máy tính. Microphone INMP441 được kết nối qua giao tiếp I2S kỹ thuật số, cho phép thu được tín hiệu âm thanh trung thực với độ nhiễu thấp, là yếu tố tiên quyết để mô hình học sâu hoạt động chính xác. Bên trong hộp cũng tích hợp loa để phát các âm thanh phản hồi như tiếng bíp xác nhận hoặc thông báo trạng thái, tạo trải nghiệm tương tác hai chiều tự nhiên cho người dùng.

3.3.1.4. Giao diện Web Dashboard giám sát và điều khiển

Song song với phần cứng, một giao diện Web Dashboard trực quan (Smart Home Dashboard) đã được xây dựng để phục vụ công tác giám sát từ xa. Ứng dụng web này kết nối trực tiếp với MQTT Broker, cho phép cập nhật dữ liệu thời gian thực với độ trễ tối thiểu.



Hình 3.7. Giao diện Web Dashboard

Giao diện được thiết kế theo phong cách hiện đại (Dark Mode) với bố cục chia thành các thẻ (Card) chức năng rõ ràng:

- **Thẻ Môi trường:** Hiển thị trực quan các thông số đo được từ Node Sensor như Nhiệt độ, Độ ẩm và Cường độ ánh sáng. Bên cạnh đó là thông tin thời tiết ngoài trời và chỉ số chất lượng không khí (AQI) được lấy từ API dự báo thời tiết, giúp người dùng so sánh môi trường trong và ngoài nhà.
- **Thẻ Điều khiển:** Các nút gạt ảo cho phép người dùng bật/tắt thủ công các thiết bị (Đèn, Quạt, Ổ cắm). Trạng thái của các nút này được đồng bộ hai chiều: khi người dùng ra lệnh bằng giọng nói tại thiết bị Edge-AI, trạng thái trên Web cũng tự động thay đổi theo ngay lập tức.
- **Biểu đồ và Dự báo:** Cung cấp biểu đồ trực quan về xu hướng thay đổi của thời tiết và các thông số môi trường trong 7 ngày tới, hỗ trợ người dùng theo dõi lịch sử hoạt động của ngôi nhà.

3.3.2. Kết quả và đánh giá

Khả năng phản hồi thời gian thực với độ trễ tối thiểu là tiêu chí cốt lõi được ưu tiên hàng đầu trong thiết kế hệ thống. Kết quả thực nghiệm gửi liên tiếp các gói tin từ Node cảm biến về Gateway qua giao thức ESP-NOW ở không gian nhiều nhiễu đã chứng minh sự ổn định của hệ thống. Đáng chú ý, chu trình từ lúc Node cảm biến đánh thức khỏi chế độ ngủ sâu, đo đạc đến khi hoàn tất gửi dữ liệu chỉ tiêu tốn khoảng 100 ms,

không chỉ đảm bảo thông tin trên Web Dashboard được cập nhật tức thì ngay khi môi trường thay đổi mà còn tối ưu hóa đáng kể tuổi thọ pin cho các thiết bị biên.

Đối với khả năng tương tác thông minh, mô hình AI tại biên đã khẳng định được độ tin cậy khi vận hành trong môi trường có tiếng ồn sinh hoạt. Tại phạm vi tương tác phổ biến từ 1 đến 3 mét, hệ thống đạt tỷ lệ nhận diện chính xác ấn tượng từ 88% đến 96%, nhờ sự hỗ trợ đặc lực của các thuật toán xử lý tín hiệu như triệt tiêu tiếng vọng và giảm nhiễu giúp lọc sạch tạp âm nền cho các lệnh điều khiển. Mặc dù hiệu năng có sự suy giảm ở cự ly xa, hệ thống vẫn đảm bảo tính thực tiễn cao cho không gian phòng ở, đồng thời giải quyết triệt để nhược điểm phụ thuộc vào đường truyền Internet thường thấy ở các trợ lý ảo đám mây hiện nay.

Xét trên khía cạnh tích hợp tổng thể, hệ thống đã thể hiện sự đồng bộ cao độ giữa các phân hệ phần cứng và phần mềm. Quy trình xử lý khép kín từ khi người dùng phát lệnh, qua nhận dạng tại Edge AI, truyền dẫn bằng ESP-NOW đến khi thiết bị thực thi chỉ diễn ra trong khoảng 300-400 ms, mang lại trải nghiệm điều khiển mượt mà và gần như tức thì. Song song với đó, trạng thái thực tế của thiết bị (Bật/Tắt) cũng được đồng bộ ngược lên Web Dashboard thông qua giao thức MQTT với độ trễ không đáng kể, đảm bảo tính nhất quán tuyệt đối giữa môi trường vật lý và giao diện giám sát số.

3.4. Kết luận chương 3

Chương 3 đã hoàn thiện quy trình thiết kế và thi công hệ thống nhà thông minh AIoT theo kiến trúc hình sao, hiện thực hóa thành công các node phần cứng chuyên biệt từ cảm biến tiết kiệm năng lượng ESP32-C3, Gateway trung chuyển ESP32-WROOM đến khối xử lý giọng nói hiệu năng cao ESP32-S3. Việc tích hợp sâu framework ESP-SR trên nền tảng FreeRTOS đã giải quyết hiệu quả bài toán xử lý tín hiệu âm thanh phức tạp ngay tại biên, được kiểm chứng qua kết quả thực nghiệm với độ trễ truyền dẫn ESP-NOW ấn tượng dưới 5ms và tỷ lệ nhận dạng lệnh giọng nói duy trì ổn định trên 88% trong môi trường sinh hoạt thực tế. Sự kết hợp đồng bộ giữa khả năng điều khiển cục bộ tin cậy và giám sát từ xa linh hoạt qua Web Dashboard không chỉ khắc phục triệt để nhược điểm phụ thuộc đường truyền Internet của các mô hình truyền thống mà còn khẳng định tính khả thi và hiệu quả của việc ứng dụng công nghệ TinyML trên thiết bị nhúng, tạo tiền đề vững chắc cho việc tối ưu hóa và mở rộng hệ thống trong tương lai.

KẾT LUẬN

Đề tài "Thiết kế và xây dựng hệ thống nhà thông minh giám sát và điều khiển bằng giọng nói ứng dụng mô hình CNN" đã hoàn thành mục tiêu xây dựng một giải pháp AIoT toàn diện, hoạt động độc lập tại biên dựa trên kiến trúc mạng hình sao. Hệ thống được hiện thực hóa đồng bộ với các node phần cứng chuyên biệt: Gateway trung tâm đảm bảo kết nối ổn định, Node cảm biến tối ưu năng lượng và đặc biệt là Node Edge-AI tích hợp vi điều khiển ESP32-S3. Điểm đột phá cốt lõi của nghiên cứu nằm ở việc ứng dụng thành công kỹ thuật TinyML thông qua framework ESP-SR, cho phép vận hành trơn tru các mô hình học sâu phức tạp như WakeNet và MultiNet ngay trên thiết bị nhúng để xử lý tín hiệu âm thanh và nhận dạng lệnh điều khiển hoàn toàn offline mà không phụ thuộc vào điện toán đám mây.

Kết quả thực nghiệm đã khẳng định tính hiệu quả và độ tin cậy vượt trội của mô hình đề xuất trong điều kiện sử dụng thực tế. Hệ thống đạt được sự cân bằng tối ưu giữa tốc độ phản hồi và độ chính xác, với độ trễ truyền dẫn qua giao thức ESP-NOW được kiểm soát và tỷ lệ nhận dạng giọng nói duy trì ổn định trong phạm vi tương tác phổ biến từ 1 đến 3 mét. Thành công này không chỉ chứng minh tính khả thi của việc triển khai trí tuệ nhân tạo trên phần cứng giới hạn tài nguyên mà còn mang lại trải nghiệm người dùng mượt mà, khắc phục triệt để các nhược điểm về độ trễ xử lý và rủi ro mất kết nối Internet thường gặp ở các giải pháp nhà thông minh truyền thống.

Mặc dù đã đạt được những kết quả khả quan, hệ thống vẫn còn tồn tại một số hạn chế nhất định về phạm vi nhận dạng xa trong môi trường nhiều tạp âm và tính linh hoạt của tập lệnh cố định. Do đó, định hướng phát triển trong tương lai sẽ tập trung vào việc nâng cấp phần cứng với mảng microphone kết hợp thuật toán để cải thiện khả năng chống nhiễu, đồng thời mở rộng khả năng tương thích với chuẩn kết nối Matter. Những cải tiến này sẽ giúp hoàn thiện sản phẩm, đưa giải pháp tiếp cận gần hơn với nhu cầu thực tiễn, góp phần thúc đẩy hệ sinh thái thiết bị thông minh an toàn và tiện ích.

TÀI LIỆU THAM KHẢO

- [1] Espressif Systems, "ESP32-S3 Technical Reference Manual," *Espressif Documentation*, v1.1, 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32s3_technical_reference_manual_en.pdf.
- [2] Espressif Systems, "ESP-IDF Programming Guide," *Espressif Documentation*, 2024. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/>.
- [3] Espressif Systems, "ESP-SR Speech Recognition Framework User Guide," *Espressif Documentation*, 2024. [Online]. Available: <https://docs.espressif.com/projects/esp-sr/en/latest/esp32s3/>.
- [4] Espressif Systems, "ESP-Skainet: Intelligent Voice Assistant," *GitHub Repository*, 2024. [Online]. Available: <https://github.com/espressif/esp-skainet>.
- [5] Espressif Systems, "WakeNet: Deep Neural Network for Keyword Spotting on ESP32," *Espressif Blog*, 2021. [Online]. Available: <https://blog.espressif.com/>.
- [6] Espressif Systems, "MultiNet: Offline Speech Command Recognition for ESP32-S3," *Espressif Technical Documents*, 2023.
- [7] K. Park (Kyubyong), "g2p_en: A Simple Python Module for English Grapheme To Phoneme Conversion," *GitHub Repository*, 2019. [Online]. Available: <https://github.com/Kyubyong/g2p>.
- [8] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, Pittsburgh, PA, 2006, pp. 369–376. (Cơ sở lý thuyết của mạng CRNN + CTC dùng trong MultiNet).
- [9] Espressif Systems, "ESP-NOW Wireless Communication Protocol," *Espressif Documentation*, 2024. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html.

[10] OASIS Standard, "MQTT Version 3.1.1," *OASIS Message Queuing Telemetry Transport Technical Committee*, 2014. [Online]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>.

[11] T. Instruments, "INMP441: Omnidirectional Microphone with Bottom Port and I2S Digital Output," *Datasheet*, rev. 1.1, 2014.