IoT Course

Capstone Project   
Action Plan

For students (instructor’s review required)

ⓒ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.

| Course | IoT Course |
| --- | --- |
| Team Name | Nhóm 9 |
| Team Leader/  Members | Văng Công Quỳnh /  Hồ Minh Tuấn, Trần Nguyễn Khánh Hoàng, Phan Minh Tân, Võ Anh Trường |
| Project Title | Thiết kế hệ thống quản lý và điều khiển thiết bị sử dụng trợ lý giọng nói dùng ESP32-S3 và Home Assistant |
| Goal |  |
| - Quản lý và điều khiển thiết bị trên đa nền tảng (web, app, điện thoại, máy tính bảng,...)  - Điều khiển và kiểm tra trạng thái hệ thống bằng giọng nói (voice assist)  - Có khả năng tùy biến (tự xây dựng thiết bị cuối) và mở rộng thêm các phần tử IoT khác từ bên thứ 3 (cảm biến, camera,...)  - Tăng cường khả năng phản hồi theo ngữ cảnh (tự động hóa theo các thiết lập ngữ cảnh)  - Thiết lập hệ thống có khả năng dễ dàng cấu hình và bảo trì hệ thống  - Hệ thống được chạy local để tối ưu tính bảo mật và sự riêng tư | |
| Abstract |  |
| 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. | |
| Method |  |
| Để hiện thực hóa hệ thống quản lý và điều khiển thiết bị sử dụng trợ lý giọng nói, đề tài áp dụng phương pháp thiết kế theo hướng phân tán với kiến trúc tổng thể gồm một trung tâm điều phối (central hub) và các thiết bị ngoại vi hoạt động độc lập (node). Quá trình triển khai được chia thành các giai đoạn chính như sau:   * Xây dựng trung tâm điều khiển (Home Assistant Core): Hệ thống trung tâm được triển khai trên nền tảng Home Assistant OS cài đặt trên Raspberry Pi 5 thực hiện các chức năng điều phối, lưu trữ trạng thái thiết bị, thiết lập tự động hóa, đồng thời tiếp nhận và phản hồi các lệnh điều khiển từ người dùng. * Thiết kế và lập trình các node ngoại vi sử dụng ESP32/ESP32-S3: dễ dàng cấu hình và kết nối trực tiếp đến Home Assistant thông qua giao thức API nội bộ. ESPHome cho phép lập trình bằng YAML kết hợp các khối logic điều kiện, đồng thời hỗ trợ cập nhật OTA và tương tác thời gian thực. * Tích hợp chức năng điều khiển bằng giọng nói: tích hợp microphone và DAC (digital to analog converter) cùng với loa trên ESP32-S3 để nhận lệnh và phản hồi. Sử dụng từ đánh thức (wake word) được nhận diện cục bộ, sau đó toàn bộ âm thanh được truyền tới Home Assistant trên Pi để xử lý. * Thiết lập các tự động hóa và giao diện người dùng (Dashboard): Các quy tắc tự động hóa được xây dựng trực tiếp trong Home Assistant dựa trên điều kiện môi trường (thời gian, cảm biến hiện diện, trạng thái thiết bị...) và ngữ cảnh sử dụng. Giao diện người dùng được tùy chỉnh để dễ dàng truy cập từ điện thoại, máy tính bảng hoặc máy tính. Dashboard có thể được tối ưu và thiết lập riêng cho từng thiết bị. * Đảm bảo bảo mật và quyền riêng tư: Hệ thống không phụ thuộc vào các dịch vụ đám mây (có thể mở rộng nhưng vẫn đảm bảo các yếu tố trên), toàn bộ dữ liệu điều khiển và xử lý giọng nói được thực hiện cục bộ. Các kết nối giữa node và Home Assistant được giới hạn trong mạng nội bộ. * Thử nghiệm và đánh giá: Sau khi hoàn thiện triển khai phần cứng và cấu hình phần mềm, hệ thống được kiểm tra hoạt động trong các kịch bản sử dụng thực tế: điều khiển đèn, phản hồi trạng thái, nhận diện lệnh thoại, và tính ổn định khi hoạt động lâu dài. Các thông số về độ trễ, độ chính xác, hiệu năng và tính mở rộng được ghi nhận để đánh giá hiệu quả hệ thống. | |

| Data |  |
| --- | --- |
| 1. **Dữ liệu giọng nói**   Dữ liệu âm thanh từ người dùng được thu thập thông qua microphone I2S (INMP441) gắn trên các node ESP32-S3. Quá trình thu âm được kích hoạt bằng từ khóa đánh thức (wake word) đã cấu hình sẵn. Sau khi phát hiện wake word, toàn bộ lệnh thoại được ghi lại và truyền đến Home Assistant để xử lý nội bộ thông qua dịch vụ Voice Assistant.   * Nguồn dữ liệu: Đầu vào từ microphone gắn trên node ESP32-S3 * Cách xử lý: Dữ liệu âm thanh được truyền trực tiếp tới Home Assistant để chuyển thành văn bản (speech-to-text) và giọng nói (text-to-speech) bằng các add-on giọng nói cục bộ. * Cách sử dụng: Văn bản sau xử lý được phân tích để thực thi lệnh điều khiển thiết bị (bật/tắt đèn, điều chỉnh độ sáng, bật quạt, v.v.)  1. **Dữ liệu trạng thái thiết bị và cảm biến**   Các node ESP32 truyền dữ liệu trạng thái thiết bị (bật/tắt, mức độ sáng, trạng thái IR, v.v.) và dữ liệu cảm biến (phát hiện chuyển động, nhiệt độ, ánh sáng...) về Home Assistant thông qua kết nối Wi-Fi nội bộ.   * Nguồn dữ liệu: Cảm biến PIR, cảm biến môi trường, trạng thái GPIO trên các node,... * Cách xử lý: Dữ liệu được gửi định kỳ hoặc theo sự kiện về Home Assistant thông qua ESPHome. * Cách sử dụng: Dùng để hiển thị trạng thái thiết bị, làm điều kiện kích hoạt các kịch bản tự động hóa hoặc phản hồi giọng nói theo ngữ cảnh.  1. **Dữ liệu điều khiển từ người dùng**   Ngoài giọng nói, người dùng còn có thể thao tác qua giao diện điều khiển trên ứng dụng di động hoặc trình duyệt web. Các thao tác này sẽ ghi nhận lệnh điều khiển và cập nhật trạng thái tương ứng lên Home Assistant.   * Nguồn dữ liệu: UI của Home Assistant (Lovelace Dashboard, Companion App) * Cách xử lý: Lệnh được xử lý cục bộ, gửi đến các thiết bị thông qua API nội bộ của ESPHome. * Cách sử dụng: Thay đổi trạng thái thiết bị, kiểm tra nhật ký hoạt động, tạo hoặc sửa các tự động hóa. | |
| Expected  Outcome |  |
| Dự án hướng đến việc xây dựng thành công một hệ thống quản lý và điều khiển thiết bị thông minh dựa trên nền tảng Home Assistant, trong đó người dùng có thể tương tác với các thiết bị trong nhà thông qua giọng nói và giao diện điều khiển trực quan. Hệ thống phải đảm bảo khả năng hoạt động ổn định, độ trễ thấp, phản hồi chính xác, dễ mở rộng và đặc biệt là xử lý cục bộ để bảo vệ quyền riêng tư người dùng.  Nếu đạt được mục tiêu trên, hệ thống sẽ mang lại nhiều lợi ích thiết thực như:   * Tăng tính tiện lợi và tương tác tự nhiên: Người dùng có thể điều khiển thiết bị chỉ bằng giọng nói mà không cần thao tác vật lý, phù hợp với nhiều tình huống sử dụng trong sinh hoạt hàng ngày. * Tối ưu khả năng tự động hóa và giám sát: Home Assistant cho phép thiết lập các quy tắc điều khiển tự động theo ngữ cảnh như thời gian, hiện diện, ánh sáng… giúp giảm tiêu thụ điện năng và tăng tính linh hoạt. * Bảo vệ quyền riêng tư và bảo mật dữ liệu: Toàn bộ hệ thống được triển khai và xử lý trong mạng nội bộ, không phụ thuộc vào dịch vụ đám mây bên ngoài, giúp loại bỏ nguy cơ rò rỉ dữ liệu cá nhân. * Khả năng nhân rộng và tùy biến: Nhờ sử dụng phần mềm mã nguồn mở và phần cứng chi phí thấp (ESP32, Raspberry Pi), hệ thống có thể được triển khai linh hoạt cho nhiều mô hình nhà thông minh khác nhau với chi phí hợp lý, hỗ trợ nhiều giao thức kết nối. | |
| Role by  Member |  |
| | Vai trò | | Người thực hiện | | | | | | --- | --- | --- | --- | --- | --- | --- | | Quỳnh | Tuấn | Hoàng | Trường | Tân | | Phân tích yêu cầu hệ thống | Thu thập yêu cầu chức năng |  |  | x |  |  | | Lập sơ đồ hệ thống |  |  | x |  |  | | Lựa chọn phần cứng |  |  | x | x |  | | Tham khảo nguồn tài liệu và dự án có sẵn | | x |  | x | x |  | | Triển khai nền tảng trung tâm (Home Assistant) | |  |  | x |  |  | | Thiết kế và lập trình node ngoại vi | |  |  | x |  |  | | Tích hợp chức năng điều khiển bằng giọng nói | |  |  | x |  |  | | Thiết lập tự động hóa và giao diện người dùng | |  |  | x |  |  | | Thử nghiệm và hiệu chỉnh hệ thống | |  |  | x |  |  | | Viết tài liệu, báo cáo | File báo cáo | x | x | x | x |  | | File thuyết trình | x |  |  | x |  | | |

| Schedule  Summary |  |
| --- | --- |
| - Bảng chi tiết kế hoạch công việc được trình bày tại trang sau:  [SIC\_IoT\_Capstone Project\_Work Breakdown Structure\_Nhóm 9.xlsx](https://docs.google.com/spreadsheets/d/1W4ETmPFs6LTyOTaSt-Kpna-LxZOwYUyF/edit?usp=sharing&ouid=107500679383481647489&rtpof=true&sd=true) | |
| Comment &  Assessment |  |
| <Comment and assessment **by the instructor.**> | |