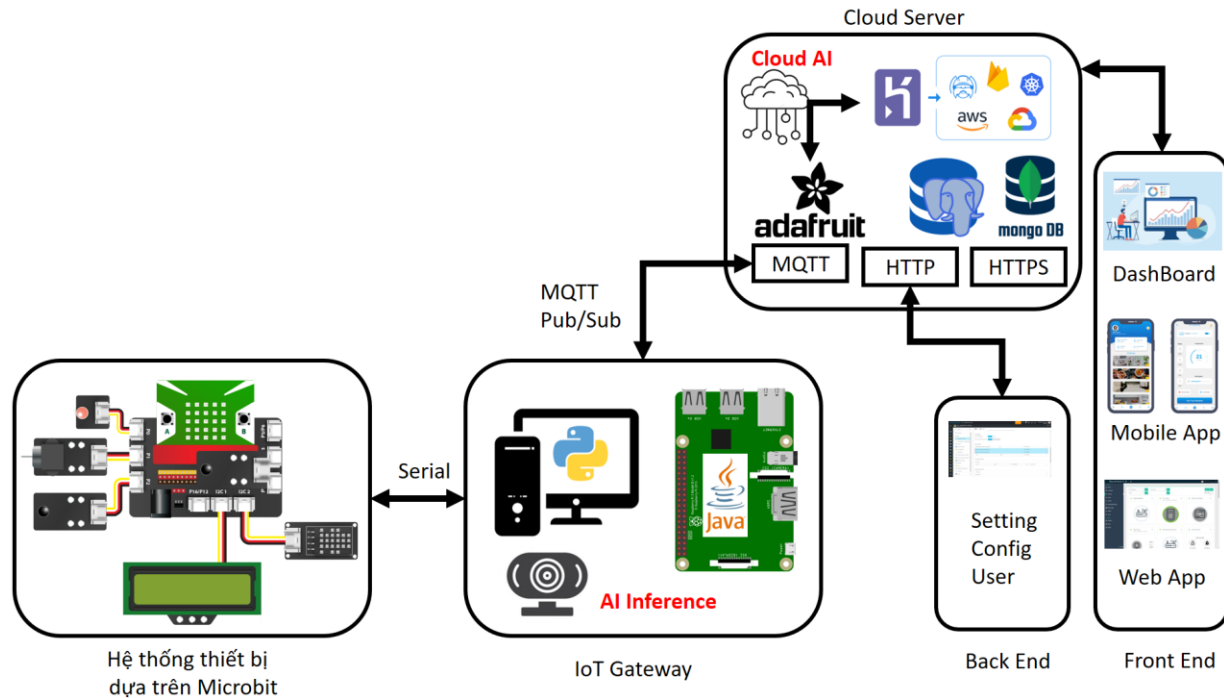


1) Kiến trúc tổng quan



Kiến trúc tổng quan được đề xuất với đủ các thành phần như IoT, AI, Hệ thống thông tin và Công nghệ phần mềm. Tùy vào đề tài của sinh viên, 1 số thành phần có thể được lược bỏ hoặc làm đơn giản. Ví dụ như sau:

- Đề tài về hướng công nghệ phần mềm, sẽ hiện thực Mobile App và/hoặc Web App, công nghệ server và database sử dụng MongoDB, server điều khiển thiết bị Adafruit IO, IoT Gateway và phần hệ thống cảm biến.
- Đề tài về hướng IoT sẽ hiện thực hệ thống cảm biến, IoT Gateway, server Adafruit IO, App Mobile và Dashboard (có thể không có phần AI và Database)

Sinh viên cần xác nhận với giảng viên hướng dẫn về các thành phần mà mình sẽ hiện thực. Sinh viên cũng có thể đề xuất phát triển thêm tính năng cho mỗi khối chức năng tùy theo dự án mà mình đề xuất.

2) Hệ thống cảm biến

Đây là thành phần liên quan nhiều đến thiết bị phần cứng. Tuy nhiên, khi đã dựa trên nền tảng board mạch Microbit, với ngôn ngữ lập trình cấp cao, sinh viên chủ yếu làm việc với ngôn ngữ lập trình và thư viện hỗ trợ sẵn.

Các thiết bị phần cứng sẽ được triển khai tại phòng thí nghiệm 202C5 và 610H6. Bắt đầu từ thứ 3, 16/03/2022, thiết bị sẽ sẵn sàng tại phòng 202C5 và sau đó là 11/03/2022 cho phòng 610H6.

Tại mỗi phòng thí nghiệm, sẽ có 2 mô hình được kết nối và lập trình sẵn, minh họa cho ứng dụng nhà thông minh và vườn rau thông minh. Sinh viên không được phép thay đổi chương trình trong 2 mô hình này mà chỉ được dùng nó để tích hợp và kiểm tra với Gateway của mình. Với mỗi mô hình sẽ có đường link hướng dẫn trực tuyến. Các câu hỏi và giải đáp cũng sẽ được hỗ trợ trực tiếp cho sinh viên.

Tại mỗi phòng thí nghiệm, sinh viên sẽ được hỗ trợ sẵn mạch Microbit, mạch mở rộng, các cảm biến và dây kết nối. Sinh viên tự chọn thiết bị cho dự án của mình, tự kết nối và lập trình.

Một hướng phát triển thêm cho hệ thống cảm biến, là xây dựng nó thành nhiều hệ thống cảm biến, giao tiếp không dây với nhau (sử dụng tính năng Radio có sẵn trên Microbit).

Bản thân mạch Microbit đã có sẵn 2 cảm biến thông dụng, là nhiệt độ và cường độ ánh sáng, rất phù hợp cho các ứng dụng quan trắc trong nhà.

Tài liệu hướng dẫn và danh sách các thiết bị có thể tìm thấy ở link sau đây:

<https://drive.google.com/file/d/1EUze00NjyJdFRy9nF8QQxjB7dwrBWjGI/view?usp=sharing>

Video hướng dẫn tương ứng với tài liệu có thể tìm thấy ở link sau đây:

https://www.youtube.com/watch?v=aFjTyc0k8xo&list=PLB5bcMnwbMmxAlaT1_I4ubvrGwtgOwhE3

Đối với các đề tài liên quan đến Robot tự hành, Robot sẽ chỉ triển khai ở phòng 202C5. Tài liệu hướng dẫn cho Robot sinh viên có thể tìm thấy ở link sau đây:

<https://drive.google.com/file/d/1vl7jIWks51x5fy34MqluStgAA4r-nDi/view?usp=sharing>

Về căn bản, Robot cũng như 1 hệ thống cảm biến, nhận lệnh từ một Gateway (camera + AI để xử lý) để điều khiển tốc độ của 4 bánh, vận hành theo 1 quy luật nào đó. Tập lệnh để điều khiển 4 bánh có thể được tìm thấy trong bài hướng dẫn đầu tiên của giáo trình.

3) IoT Gateway

IoT Gateway có 2 tính năng chính là giao tiếp với hệ thống cảm biến và server điều khiển thiết bị, hỗ trợ giao thức MQTT.

- Giao tiếp với hệ thống cảm biến: Được dựa trên dây nối USB từ mạch Microbit và Máy tính (hoặc máy tính nhúng). Đây là giao tiếp COM ảo, tốc độ 115200 (8 bits data, 1 stop bit, no parity). Sinh viên cần định nghĩa định dạng dữ liệu cho giao tiếp giữa Gateway và Hệ thống cảm biến. Định dạng dữ liệu này bắt buộc phải có kí tự bắt đầu và kí tự kết thúc.
- Giao tiếp với server MQTT: server Adafruit được khuyến khích sử dụng trong đề án. Tuy nhiên sinh viên cũng có thể đề xuất server khác (cloud MQTT, Thingsboard, v.v...) hoặc thậm chí tự hiện thực. Server MQTT phải hỗ trợ đăng nhập từ nhiều thiết bị, có cơ chế Pub/Sub một hoặc nhiều topics đồng thời. Sinh viên cũng phải tự định nghĩa định dạng dữ liệu khi publish lên server và khi nhận từ server. Thông thường, dữ liệu này sẽ ở dạng JSON.

Tại IoT gateway, khi nhận được dữ liệu từ server, thông thường sẽ cần bước phân tích dữ liệu. Sau đó đóng gói nó lại để gửi qua hệ thống cảm biến. Tương tự cho chiều ngược lại, dữ liệu từ hệ thống cảm biến tới gateway, cũng được phân tích và thay đổi format trước khi publish lên server hoặc gọi API tới các server khác, phục vụ cho việc lưu trữ thông tin trong Database.

Đối với hướng ứng dụng AI, gateway IoT có thể tích hợp thêm camera để thực hiện bài toán AI Inference. Một ví dụ cho tính năng này là sử dụng camera để nhận dạng lá cây bị héo úa. Sinh viên có thể tự train một AI Model hoặc sử dụng một Model đã có sẵn.

IoT Gateway có thể được hiện thực bằng bất kì nền tảng và bất kì ngôn ngữ lập trình nào, có hỗ trợ tối thiểu kết nối Serial và MQTT.

Tài liệu hướng dẫn sử dụng Teachable Machine của Google:

https://drive.google.com/file/d/1d-VGIM5m_jFh9WkA38kg6v0o8zmF85v5/view?usp=sharing

Video hướng dẫn sử dụng Teachable Machine của Google:

<https://www.youtube.com/watch?v=tq6R5tXGqls&list=PLB5bcMnwbMmy-E0nYL-BH20cJ2r-Y1P5d>

Một số Model AI hỗ trợ từ Google:

<https://coral.ai/models/>

4) Cloud Server

Sinh viên chọn một công nghệ cho việc hiện thực các tính năng lưu trữ thông tin trong cơ sở dữ liệu do sinh viên thiết kế.

Trong trường hợp sử dụng 2 server (1 cho database và Adafruit IO), khi thiết bị đầu cuối (Front End) cần tương tác đến 2 server, sinh viên cần xử lý dưới dạng transaction để đảm bảo tính toàn vẹn của dữ liệu. Một API sẽ được sinh viên hiện thực, xử lý lưu trữ trong database sau đó mới gửi MQTT Pub/Sub tới Adafruit

Sinh viên cũng có thể tự hiện thực MQTT và không bắt buộc phải sử dụng Adafruit IO

Sinh viên có thể hiện thực Cloud AI trên server của mình (ví dụ cho bài toán FaceID)

5) Front End và Back End

Tùy đồ án mà sinh viên lựa chọn, các hình thức cho Front End và Back End có thể sẽ khác nhau.

Theo hướng IoT, Dashboard và Mobile App sẽ rất cần thiết cho việc quan trắc và điều khiển thiết bị

Theo hướng database, Web App và các trang để quản trị sẽ cần được ưu tiên.

Sinh viên cần chọn lựa giao thức phù hợp giữa Front End và Back End với server. Thông thường Dashboard sẽ cần MQTT để có thể cập nhật dữ liệu tức thì. Tuy nhiên các tác vụ cập nhật thông tin vào database thì chỉ cần HTTP GET/POST là đủ.

Theo hướng công nghệ phần mềm, Front End và Back End cần được phát triển theo 1 pattern của công nghệ phần mềm.