

HỆ THỐNG BÃI ĐỖ XE THÔNG MINH

Đề xuất dự án ứng dụng công nghệ AIoT

Ngày 22 tháng 5 năm 2025

Tóm tắt nội dung

Báo cáo này trình bày đề xuất phát triển hệ thống bãi đỗ xe thông minh ứng dụng công nghệ AIoT, nhằm tự động hóa và tối ưu hóa quy trình quản lý bãi đỗ xe. Giải pháp đề xuất kết hợp các công nghệ nhận diện hình ảnh (biển số xe, khuôn mặt người lái) và hệ thống cảm biến IoT để theo dõi tình trạng chỗ đỗ trong thời gian thực, loại bỏ thao tác thủ công và giảm thiểu sai sót trong quá trình vận hành. Các tính năng chính bao gồm: tự động nhận diện phương tiện ra vào, giám sát trạng thái chỗ đỗ, lưu trữ dữ liệu trên nền tảng cloud và cung cấp giao diện dashboard trực quan cho người quản lý. Hệ thống có thể triển khai linh hoạt tại nhiều môi trường khác nhau như trường học, khu dân cư, bãi xe công cộng, góp phần xây dựng hạ tầng đô thị thông minh và hiện đại.

Mục lục

1	Giới thiệu	4
1.1	Bối cảnh	4
1.2	Mục đích	4
1.3	Mục tiêu	5
1.3.1	Nhận diện tự động	5
1.3.2	Đảm bảo hiệu suất và tính thời gian thực	5
1.3.3	Hoàn thiện tính năng quản lý bãi xe quy mô nhỏ	5
1.4	Phát biểu bài toán	6
1.4.1	Dữ liệu đầu vào	6
1.4.2	Kết quả đầu ra	6
1.5	Phạm vi và Giới hạn của Hệ thống	6
1.5.1	Phạm vi triển khai	6
1.5.2	Giới hạn của Hệ thống	7
1.6	Đóng góp chính của Đề tài	7
2	Kiến trúc phần mềm	9
2.1	Thiết bị đầu vào (Input Layer)	9
2.2	Tầng xử lý trung tâm (Processing Layer)	9
2.2.1	Các mô hình Trí tuệ Nhân tạo (AI Models)	9
2.2.2	Luồng xử lý dữ liệu và các thành phần chính	10
2.3	Thiết bị điều khiển - phản hồi (Output Layer)	11
2.4	Nền tảng lưu trữ	12
3	Hệ thống phần cứng	13
3.1	Chi tiết thiết bị	13
3.1.1	ESP32-CAM (AI Thinker ESP32-CAM)	13
3.1.2	ESP32 trung tâm (ESP32 DevKit V1)	15
3.1.3	Breadboard	17
3.1.4	Màn hình OLED	17
3.1.5	Buzzer	18
3.1.6	Servo motor MG90S	18
3.2	Tổng hợp	18

4 Kế hoạch thực hiện	20
4.1 Phân công nhân lực	20
4.2 Timeline	21
4.3 Chi phí thực hiện (chi phí thiết bị, chi phí làm)	21

1 Giới thiệu

1.1 Bối cảnh

Trong bối cảnh đô thị hóa diễn ra mạnh mẽ, việc quản lý hiệu quả các bãi đỗ xe tại các khu vực tập trung đông người như chung cư, trường học, và tòa nhà văn phòng ngày càng trở nên cấp thiết. Hiện nay, nhiều bãi đỗ xe đã ứng dụng công nghệ thẻ từ (RFID) và phần mềm quản lý cơ bản, tuy nhiên, các giải pháp này thường gặp phải những hạn chế đáng kể:

- **Thiếu tự động hóa:** Quy trình vận hành vẫn phụ thuộc nhiều vào thao tác thủ công như quét thẻ, ghi nhận thông tin, dễ dẫn đến sai sót (mất thẻ, quên thẻ, ghi nhận sai) và tốn thời gian.
- **Thiếu thông tin thời gian thực:** Hệ thống không giám sát được tình trạng thực tế của từng vị trí đỗ, khiến người dùng không biết trước bãi xe còn chỗ hay không, gây bất tiện và lãng phí thời gian tìm kiếm.
- **Lưu trữ dữ liệu cục bộ:** Dữ liệu thường được lưu trữ nội bộ, thiếu khả năng đồng bộ lên các nền tảng trực tuyến, gây khó khăn trong việc truy xuất và phân tích khi cần thiết, đặc biệt khi có sự cố.

Những hạn chế này không chỉ làm giảm hiệu quả quản lý mà còn ảnh hưởng tiêu cực đến trải nghiệm của người dùng. Do đó, nhu cầu về một giải pháp quản lý bãi đỗ xe thông minh, tự động hóa cao và cung cấp thông tin chính xác theo thời gian thực là vô cùng lớn. Công nghệ AIoT (Artificial Intelligence of Things), với sự kết hợp giữa Trí tuệ Nhân tạo (AI) và Internet Vạn vật (IoT), mở ra tiềm năng to lớn để giải quyết những thách thức này, hướng tới xây dựng một hệ thống hạ tầng đô thị hiện đại và hiệu quả hơn.

1.2 Mục đích

Đề tài này nhằm mục đích nghiên cứu, thiết kế và xây dựng một mô hình thử nghiệm (prototype) **Hệ thống Bãi đỗ xe Thông minh** ứng dụng công nghệ AIoT. Hệ thống hướng đến việc tự động hóa toàn diện quy trình quản lý phương tiện ra vào, tối ưu hóa việc sử dụng không gian đỗ xe, giảm thiểu sự can thiệp của con người, đồng thời nâng cao trải nghiệm người dùng và hiệu quả vận hành cho đơn vị quản lý.

1.3 Mục tiêu

Để đạt được mục đích đã đề ra, hệ thống tập trung vào ba nhóm mục tiêu chính. Mỗi mục tiêu được cụ thể hóa bằng các tiêu chí định lượng nhằm đảm bảo khả năng kiểm chứng trong quá trình phát triển và đánh giá hệ thống.

1.3.1 Nhận diện tự động

Xây dựng chức năng nhận diện gương mặt, biển số xe hoạt động ổn định trong điều kiện thực tế.

Các tiêu chí đánh giá:

- Tỷ lệ nhận diện chính xác gương mặt và biển số xe đạt trên 80% trong điều kiện ánh sáng ban ngày.
- Tỷ lệ nhận diện chính xác đạt trên 70% trong điều kiện ánh sáng yếu (ví dụ: buổi tối có đèn chiếu sáng của bãi xe).
- Hệ thống được kiểm thử trên một tập dữ liệu 6 lượt xe vào/ra khác nhau.

1.3.2 Đảm bảo hiệu suất và tính thời gian thực

Đảm bảo hệ thống phản hồi nhanh và cập nhật kịp thời các thay đổi từ thực tế.

Các tiêu chí đánh giá:

- Thời gian xử lý trung bình cho một lượt xe (tính từ lúc xe dừng đúng vị trí đến khi hệ thống xử lý xong) không vượt quá 5 giây.
- Trạng thái của các vị trí đỗ xe (trống hoặc có xe) được cập nhật lên giao diện quản lý trong vòng tối đa 3 giây kể từ khi có sự thay đổi thực tế từ cảm biến.

1.3.3 Hoàn thiện tính năng quản lý bãi xe quy mô nhỏ

Xây dựng đầy đủ các chức năng cần thiết để vận hành một mô hình bãi xe nhỏ (5–10 vị trí đỗ xe).

Các tiêu chí đánh giá:

- Xây dựng và kiểm thử đầy đủ 100% các mô-đun chính của hệ thống.
- Ghi nhận thông tin xe vào/ra (biển số, thời gian) một cách tự động.

- Giám sát và hiển thị trực quan trạng thái của từng vị trí đỗ xe.
- Lưu trữ lịch sử các lượt gửi xe trên nền tảng cloud.
- Cung cấp giao diện dashboard cho phép người quản lý theo dõi hoạt động và truy xuất dữ liệu cơ bản.

1.4 Phát biểu bài toán

1.4.1 Dữ liệu đầu vào

- Hình ảnh chất lượng cao từ camera giám sát tại các lối vào và lối ra của bãi xe.

1.4.2 Kết quả đầu ra

- Thông tin nhận diện chính xác (biển số xe, khuôn mặt) tại thời điểm phương tiện vào – ra.
- Bản ghi chi tiết về thời gian vào – ra của từng phương tiện.
- Trạng thái cập nhật liên tục của từng vị trí đỗ (còn trống/đã chiếm chỗ).
- Giao diện dashboard trực quan, dễ sử dụng cho người quản lý và thông tin cho người dùng.
- Cơ sở dữ liệu được lưu trữ an toàn và có khả năng truy xuất hiệu quả trên nền tảng cloud.

1.5 Phạm vi và Giới hạn của Hệ thống

1.5.1 Phạm vi triển khai

Hệ thống được thiết kế với định hướng triển khai tại các tổ chức và doanh nghiệp có nhu cầu kiểm soát an ninh và quản lý phương tiện ra vào một cách tự động và hiệu quả, ví dụ như:

- Công ty, tòa nhà văn phòng.
- Nhà máy, khu công nghiệp.
- Trường học, bệnh viện.

Mô hình hoạt động phù hợp với các đơn vị có sẵn hạ tầng mạng cục bộ ổn định và có khả năng đầu tư cơ bản vào thiết bị như ESP32-CAM, ESP32 điều khiển trung tâm và màn hình hiển thị. Ban đầu, hệ thống sẽ được phát triển dưới dạng một mô hình thử

nghiệm (prototype) cho một bãi đỗ xe quy mô nhỏ (5-10 chỗ) để kiểm chứng tính khả thi và hiệu quả của các giải pháp công nghệ được lựa chọn.

Lưu ý: Trong giai đoạn hiện tại, hệ thống chưa tích hợp chức năng thu phí tự động. Tuy nhiên, đây là một hướng phát triển tiềm năng trong tương lai, đặc biệt khi ứng dụng tại các bãi xe công cộng hoặc dịch vụ.

1.5.2 Giới hạn của Hệ thống

Hệ thống nhận diện và quản lý tự động có thể gặp một số giới hạn sau:

- **Điều kiện môi trường:** Độ chính xác của module nhận diện hình ảnh (biển số, khuôn mặt) có thể bị ảnh hưởng bởi các yếu tố như ánh sáng yếu (ban đêm, khu vực thiếu sáng), điều kiện thời tiết bất lợi (mưa lớn, sương mù dày đặc), hoặc góc quay camera không tối ưu.
- **Chất lượng camera:** Yêu cầu camera có độ phân giải đủ tốt và được lắp đặt ở vị trí phù hợp để đảm bảo chất lượng hình ảnh đầu vào cho quá trình nhận diện.
- **Tài nguyên phần cứng:** Hệ thống cần được cung cấp đủ tài nguyên xử lý (CPU, RAM), dung lượng lưu trữ và băng thông mạng để đảm bảo hoạt động ổn định và đáp ứng yêu cầu thời gian thực.
- **Độ trễ hệ thống:** Mặc dù mục tiêu là giảm thiểu độ trễ, một khoảng trễ nhất định trong quá trình xử lý dữ liệu và phản hồi là khó tránh khỏi, cần được tối ưu để không ảnh hưởng đến trải nghiệm người dùng.

1.6 Đóng góp chính của Đề tài

Đề tài dự kiến mang lại những đóng góp khoa học và thực tiễn sau:

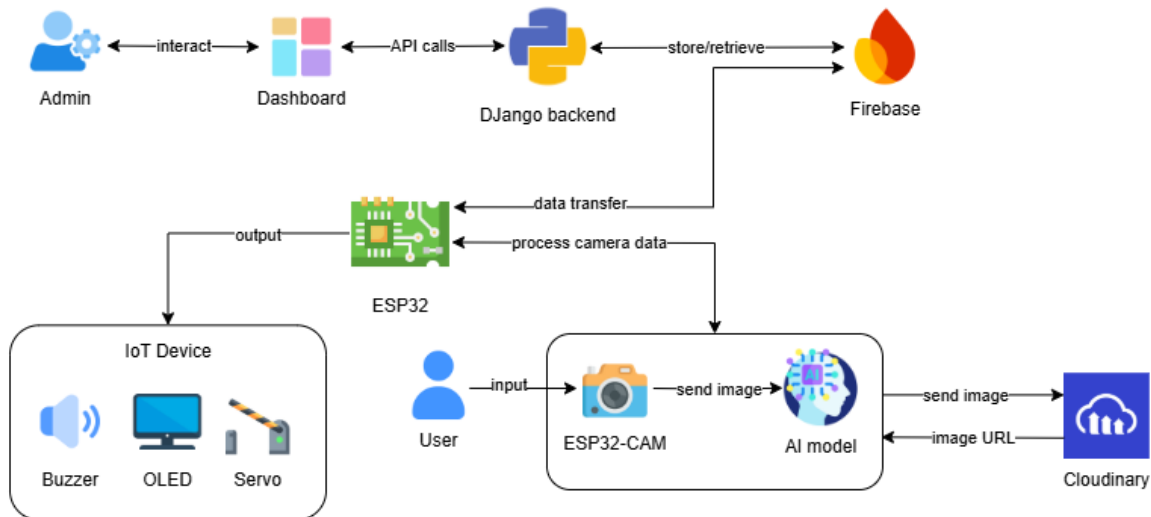
- **Đề xuất mô hình hệ thống bãi đỗ xe thông minh toàn diện:** Tích hợp công nghệ AIoT một cách hiệu quả, có khả năng ứng dụng vào thực tế tại Việt Nam.
- **Tự động hóa quy trình vận hành:** Kết hợp nhận diện hình ảnh (biển số/khuôn mặt) để giảm thiểu tối đa sự can thiệp thủ công, nâng cao độ chính xác.
- **Nâng cao hiệu quả quản lý và trải nghiệm người dùng:** Cung cấp công cụ giám sát trực quan cho người quản lý và thông tin hữu ích cho người dùng (ví dụ: tình trạng chỗ trống).
- **Định hướng phát triển và mở rộng:** Tạo tiền đề cho việc triển khai hệ thống

tại các loại hình bãi đỗ xe khác nhau (trường học, tòa nhà, trung tâm thương mại) và tích hợp thêm các tính năng nâng cao trong tương lai.

- **Giảm thiểu sai sót và tối ưu hóa nguồn lực:** Góp phần giảm thiểu các lỗi thường gặp trong quản lý thủ công và sử dụng hiệu quả hơn nguồn nhân lực.

2 Kiến trúc phần mềm

Hệ thống được thiết kế dựa trên mô hình phân tầng, bao gồm các thành phần chính: tầng thiết bị đầu vào, tầng xử lý trung tâm, và tầng điều khiển - phản hồi.



Hình 1: Sơ đồ kiến trúc hệ thống

2.1 Thiết bị đầu vào (Input Layer)

- Người dùng tương tác trực tiếp với hệ thống thông qua camera.
- Camera OV2640 tích hợp trên ESP32-CAM: Ghi nhận hình ảnh đầu vào (biển số và khuôn mặt).

2.2 Tầng xử lý trung tâm (Processing Layer)

Tầng này chịu trách nhiệm xử lý dữ liệu từ thiết bị đầu vào, thực hiện các tác vụ trí tuệ nhân tạo và điều phối hoạt động của hệ thống.

2.2.1 Các mô hình Trí tuệ Nhân tạo (AI Models)

Hệ thống sử dụng các mô hình AI chuyên biệt cho từng tác vụ, được tóm tắt trong bảng sau:

Tác vụ AI	Mô hình / API sử dụng		Thiết bị triển khai
		Mục đích	
Phát hiện khuôn mặt (có/không + bounding box)	face_detection – mô hình nhúng do ESP-WHO cung cấp	Xác định có khuôn mặt trong ảnh và trả về bounding box để chụp ảnh	ESP32-CAM (chạy trực tiếp trên thiết bị)
Nhận diện khuôn mặt (face matching)	MobileFaceNet	Trích xuất vector đặc trưng và so khớp với cơ sở dữ liệu người dùng	Server (Django backend)
Nhận dạng ký tự biển số (OCR)	Plate Recognizer API (REST API)	Gửi ảnh biển số lên server API và nhận lại chuỗi ký tự đã nhận dạng	Server (Django backend)

Hệ thống áp dụng các mô hình AI nhẹ, phù hợp với thiết bị nhúng và backend, để thực hiện hai nhiệm vụ chính: nhận diện khuôn mặt và nhận dạng biển số xe. Cụ thể:

- **Phát hiện khuôn mặt:** Mô hình `face_detection` từ thư viện ESP-WHO được chạy trực tiếp trên ESP32-CAM. Mô hình này có khả năng phát hiện nhanh khuôn mặt và trả về bounding box, giúp thiết bị xác định thời điểm chụp và gửi ảnh về server.
- **Nhận diện khuôn mặt:** Server sử dụng mô hình MobileFaceNet để trích xuất vector đặc trưng từ ảnh khuôn mặt và so sánh với cơ sở dữ liệu người dùng đã đăng ký.
- **Nhận dạng biển số xe:** Hệ thống sử dụng Plate Recognizer API để nhận diện chuỗi ký tự từ ảnh biển số xe. Ảnh được gửi từ server lên API và trả về kết quả nhận dạng dưới dạng văn bản.

2.2.2 Luồng xử lý dữ liệu và các thành phần chính

1. ESP32-CAM (Thiết bị ghi nhận hình ảnh):

- Chụp ảnh từ camera tích hợp.
- Chạy mô hình `face_detection` (ESP-WHO) để kiểm tra có khuôn mặt hay không.

- Nếu có mặt người, ảnh sẽ được gửi lên Cloudinary, sau đó nhận lại URL ảnh.
- Gửi gói dữ liệu chứa URL ảnh và metadata (thời gian, loại ảnh, thiết bị...) lên Firebase.

2. Server Backend (FastAPI):

- Theo dõi hoặc được kích hoạt từ sự kiện mới trên Firebase.
- Đọc URL ảnh từ Firebase và tiến hành xử lý:
 - **MobileFaceNet**: trích xuất embedding và xác định danh tính người dùng.
 - **Plate Recognizer API**: nhận diện chuỗi ký tự biển số xe từ ảnh.
- Ghi kết quả trả về vào Firebase: gồm thông tin người dùng, biển số xe, trạng thái xác minh (hợp lệ/không hợp lệ), thời gian xử lý, v.v.

3. ESP32 (Thiết bị điều phối trung tâm):

- Theo dõi kết quả từ Firebase.
- Khi có kết quả tương ứng với ảnh mình đã gửi, ESP32:
 - Truy xuất kết quả nhận diện.
 - Điều khiển thiết bị đầu ra (mở barrier, hiển thị OLED, còi, v.v.).
 - Ghi lại trạng thái hoạt động (đã xử lý xong) nếu cần.

4. Dashboard quản trị (ReactJS):

- Lấy dữ liệu trực tiếp từ Firebase để hiển thị:
 - Danh sách người vào/ra
 - Ảnh chụp, biển số, thời gian
 - Lịch sử, trạng thái, báo cáo thống kê

2.3 Thiết bị điều khiển - phản hồi (Output Layer)

- **Thiết bị IoT** gồm:
 - **Servo (Barrier)**: mở/đóng rào chắn tự động.
 - **OLED Display**: hiển thị thông tin trực quan (số ô trống, lỗi, trạng thái...).
 - **Buzzer**: cảnh báo âm thanh cho các trạng thái khác nhau.
- **Dashboard (giao diện quản trị cho Admin)**:
 - Sử dụng các công nghệ phổ biến HTML, CSS, JavaScript để xây dựng giao diện thân thiện, dễ sử dụng.
 - Giao tiếp với backend qua API.

- Hiển thị dữ liệu thời gian thực và lịch sử ra vào.
- Cho phép admin giám sát toàn bộ hệ thống và các thiết bị.

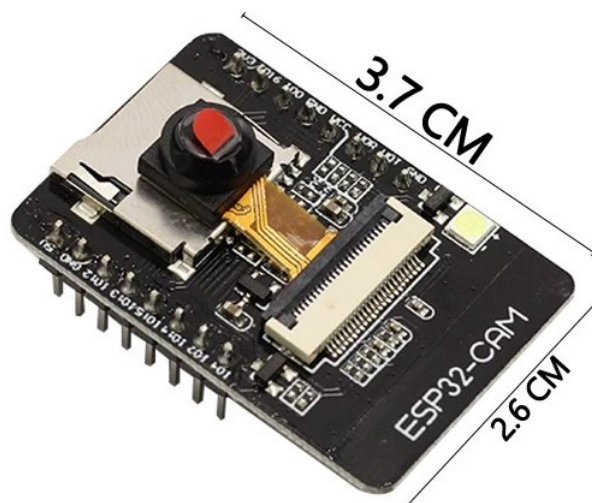
2.4 Nền tảng lưu trữ

- **Firebase:**
 - Lưu trữ dữ liệu hệ thống, nhật ký hoạt động, và thông tin người dùng.
 - Hỗ trợ đồng bộ dữ liệu và hiển thị dashboard.
- **Cloudinary:**
 - Dùng để lưu trữ ảnh chụp từ ESP32-CAM gửi đến và truy xuất nhanh bằng URL.

3 Hệ thống phần cứng

3.1 Chi tiết thiết bị

3.1.1 ESP32-CAM (AI Thinker ESP32-CAM)



Hình 2: ESP32-CAM

ESP32-CAM là một module phát triển nhỏ gọn của hãng **AI. Thinker**, tích hợp sẵn: **Wi-Fi, Bluetooth, camera OV2640, khe cắm thẻ microSD, GPIO** đa năng.

Module này được thiết kế chuyên dụng cho các ứng dụng thị giác máy tính, IoT, nhận diện khuôn mặt, giám sát an ninh, ...

ESP32-CAM là lựa chọn phổ biến nhờ giá rẻ, tính năng mạnh, và khả năng chạy AI trực tiếp trên thiết bị.

Thành phần	Thông tin chi tiết	Ghi chú
Vi xử lý chính	ESP32-WROOM-32	Xử lý chương trình nạp vào
Tốc độ xung nhịp	Lên tới 240 MHz	Xử lý được 240 triệu lệnh mỗi giây
RAM	520 KB SRAM nội	Lưu dữ liệu tạm thời khi chương trình đang chạy, tự động mất khi tắt nguồn
Flash	4 MB (SPI flash)	Lưu trữ chương trình được nạp vào chip, không mất khi tắt nguồn

Thành phần	Thông tin chi tiết	Ghi chú
Camera	OV2640, độ phân giải tối đa 1600x1200 (UXGA), hỗ trợ nhiều chế độ JPEG	Có LED flash trắng tích hợp (nối chân GPIO nếu dùng), không tự động lấy nét, hình bị mờ nếu đặt sai khoảng cách.
GPIO	~9 chân GPIO khả dụng	Các chân đa năng trên chip, cho phép nhận tín hiệu vào và gửi tín hiệu ra
Nguồn hoạt động	3.3V	Khi dùng nguồn 5V, cấp nguồn vào chân VCC/5V, bo mạch ESP32-CAM có sẵn mạch ổn áp AMS1117, chuyển 5V thành 3.3V cấp cho chip
Dòng tiêu thụ	160–250mA khi hoạt động	Nên dùng nguồn ổn định từ MB102 hoặc pin sạc 5V - 1A trở lên
Cổng USB	Không có	Cần dùng USB to UART để nạp code
Kích thước	27 x 40.5 x 4.5 mm	

Hệ thống sử dụng **2 module ESP32-CAM** cho mỗi cổng vào và ra:

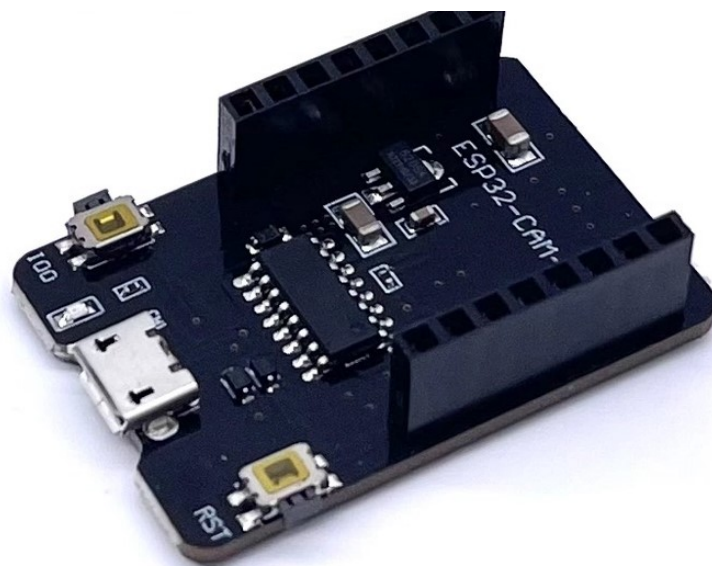
- Một module quét và nhận diện khuôn mặt tài xế
- Một module quét và nhận diện biển số xe

Mỗi ESP32-CAM hoạt động độc lập, được nạp sẵn chương trình, luôn bật camera và model nhận diện, có khả năng:

- Nhận diện sự xuất hiện của xe
- Tự động chụp ảnh
- Thực hiện nhận diện khuôn mặt hoặc biển số trực tiếp trên thiết bị
- Truyền tải thông tin:
 - Gửi hình chụp khuôn mặt và biển số xe tới **Cloudinary** qua giao thức **HTTP POST**.
 - Nhận url hình ảnh từ **Cloudinary**, gửi gói tin gồm url vừa nhận được và kết quả nhận diện vừa xử lý về ESP32 trung tâm qua giao thức **MQTT**.

ESP32 trung tâm sẽ kết nối chung mạng Wi-Fi với 4 ESP32-CAM, đóng vai trò nhận thông tin từ các module này. Sau khi nhận được kết quả nhận diện thông qua Wi-Fi, ESP32 trung tâm xử lý và điều khiển các thiết bị output liên quan như màn hình OLED, còi báo buzzer và servo.

Nạp code:



Hình 3: Đế nạp ESP32-CAM

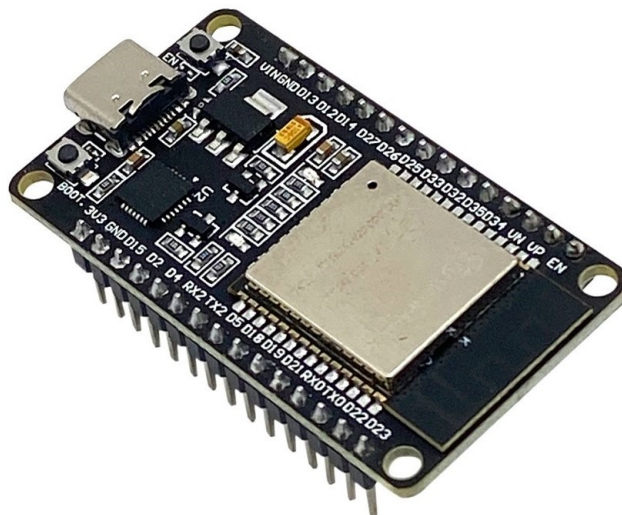
- Vì ESP32-CAM không có cổng USB, nên ta cần thiết bị trung gian để nạp code là **Đế Nạp ESP32-CAM**, sau khi nạp ESP32-CAM sẽ chạy độc lập và không mất code khi tắt nguồn. Đây là module chuyển đổi tín hiệu USB và UART, cho phép kết nối giữa máy tính và các thiết bị không có cổng USB trực tiếp như ESP32-CAM.

Lưu ý khi vừa nạp code, ESP32-CAM đang ở chế độ bootloader, cần ấn nút *reset* để chip khởi động lại chế độ chạy bình thường.

Đối với mô hình AI, sau khi train, ta sẽ chuyển sang định dạng nhẹ và export model ra dạng `.c` hoặc `.h` rồi include vào code.

3.1.2 ESP32 trung tâm (ESP32 DevKit V1)

ESP32 trung tâm là **bộ điều khiển chính** của toàn hệ thống, xử lý và điều phối giữa các thiết bị. Board sử dụng **chip ESP32-WROOM-32**, có thể xử lý nhiều tác vụ song song.



Hình 4: ESP32 DevKit V1

Thành phần	Thông tin chi tiết	Ghi chú
Chip chính	ESP32-WROOM-32	
RAM	520 KB	
Flash	4MB SPI Flash	
Tốc độ xử lý	Lên đến 240MHz	
Điện áp logic	3.3V	Thiết bị gửi tín hiệu 5V vào ESP32 GPIO có thể gây cháy GPIO, cần sử dụng mạch chia áp
USB-to-Serial	Tích hợp sẵn CP2102	Có thể nạp code trực tiếp qua cổng USB-C / microUSB
GPIO khả dụng	~25 chân, đủ dùng cho nhiều ngoại vi đồng thời	
Wi-Fi / Bluetooth	Tích hợp sẵn, dùng được ở chế độ Station và Access Point	Với ESP32 và 4 ESP32-CAM, ta sẽ dùng chế độ Station, nghĩa là cùng kết nối wifi có sẵn

3.1.3 Breadboard

Là bảng mạch kết nối không cần hàn, được sử dụng để phân phối nguồn điện và kết nối các thiết bị với nhau.

Cho phép kết nối linh kiện một cách linh hoạt thông qua dây jumper, giúp xây dựng và tổ chức hệ thống điện tử một cách gọn gàng, dễ bảo trì và dễ mở rộng.

Trong hệ thống này, breadboard sẽ được cấp nguồn trực tiếp từ adapter 5V – 3A thông qua cáp chia nguồn, sử dụng dây jumper để kết nối các thành phần trong mạch breadboard.

Loại dây	Đầu cắm	Ứng dụng
Male-Male	Đầu cắm kim 2 bên	Dùng để cắm từ chân này sang chân khác trên breadboard hoặc từ breadboard đến board mạch (ESP32, MB102...)
Male-Female	Một đầu kim, một đầu lỗ	Dùng để nối module/cắm biến (có chân đực) với breadboard hoặc board mạch
Female-Female	Hai đầu lỗ	Dùng để nối giữa 2 thiết bị đều có chân đực , ví dụ ESP32-CAM và FTDI , hoặc giữa module logic với module khác

3.1.4 Màn hình OLED

Là màn hình đơn sắc kích thước 0.96 inch, giao tiếp bằng chuẩn **I2C (SDA, SCL)** với ESP32 trung tâm (thường là GPIO21 và GPIO22). Trong hệ thống này ta sử dụng màn hình OLED với IC điều khiển SSD1306, điện áp hoạt động 5V (có tương thích 3.3V).

Có chức năng hiển thị:

- Khi xe vào: Hiển thị số chỗ còn trống, hoặc báo “đã hết chỗ” nếu không còn slot đỗ.
- Khi xe ra: Hiển thị lời chào hoặc hiển thị lỗi nếu quá trình nhận diện không hợp lệ.

3.1.5 Buzzer

Sử dụng **2 buzzer loại passive 5V**, mỗi chiếc được điều khiển bởi ESP32 trung tâm thông qua chân GPIO.

Passive buzzer là loại không có mạch dao động tích hợp bên trong, vì vậy **có thể phát được nhiều loại âm thanh khác nhau**. ESP32 sẽ tạo ra các tín hiệu tần số khác nhau (qua lệnh `tone()`), giúp buzzer phát ra các âm báo tùy biến:

- Âm báo thành công
- Âm báo lỗi

Giúp người dùng **phân biệt ngữ cảnh thông qua loại âm phát ra**.

Buzzer passive hoạt động ổn định ở **điện áp 5V**, được cấp nguồn từ breadboard thông qua module MB102, và điều khiển bằng tín hiệu logic 3.3V từ ESP32 mà không cần mạch khuếch đại.

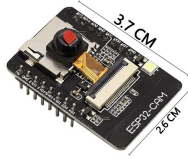
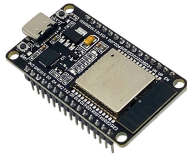





3.1.6 Servo motor MG90S



MG90S là một loại **servo mini** với **moment xoắn**, có cấu trúc **bánh răng kim loại**, bền hơn và chịu lực tốt hơn, phù hợp để điều khiển cơ cấu vật lý như **thanh chắn**.

Servo hoạt động ở **điện áp 5V**, tiêu thụ dòng khoảng **250–400mA khi tải nặng**, do đó cần cấp nguồn ổn định **từ MB102** qua breadboard để tránh sụt áp hoặc làm **ESP32 reset** đột ngột.

ESP32 trung tâm điều khiển servo qua tín hiệu PWM từ một chân GPIO bất kỳ (thường dùng GPIO13 hoặc GPIO14). Tín hiệu PWM xác định góc quay của servo (trong khoảng 0° – 180°).

3.2 Tổng hợp

STT	Tên thiết bị	Số lượng	Giá/cái	Hình ảnh	Nguồn liên kết
1	AI Thinker ESP32-CAM	4	165,000		Xem tại đây
2	ESP32 DevKit V1 (CP2102, Type-C)	1	97,000		Xem tại đây
3	Màn hình OLED 0.96 inch	2	55,000		Xem tại đây
4	Servo motor MG90S	2	37,000		Xem tại đây
5	Buzzer	2	5,000		Xem tại đây
6	Adapter 5V - 3A	1	52,000		Xem tại đây
7	Bộ chia nguồn DC male	1	26,000		Xem tại đây

STT	Tên thiết bị	Số lượng	Giá/cái	Hình ảnh	Nguồn liên kết
8	Jack DC cái có dây 5.5x2.1mm	5	3,000		Xem tại đây
9	Breadboard	1	17,000		Xem tại đây
10	Dây jumper	40 sợi/loại	66,000		Xem tại đây
11	Đế Nạp ESP32-CAM	1	30,000		Xem tại đây
12	Module chuyển mạch I2C	1	33,000		Xem tại đây

4 Kế hoạch thực hiện

4.1 Phân công nhân lực

Thành viên	Vai trò chính	Nhiệm vụ cụ thể
Cao Uyển Nhi	Trưởng nhóm, quản lý tiến độ	Phụ trách AI – Nhận diện biển số/khuôn mặt

Thành viên	Vai trò chính	Nhiệm vụ cụ thể
		Tổng hợp báo cáo, điều phối nhóm, hỗ trợ kỹ thuật
		Thu thập dữ liệu, huấn luyện mô hình, triển khai xử lý ảnh
Võ Lê Việt Tú	Phụ trách IoT – Cảm biến và cloud	Kết nối cảm biến, lập trình ESP32, gửi dữ liệu lên Firebase/ThingsBoard
Trần Thị Cát Tường	Thiết kế phần cứng – Mạch, cảm biến	Lắp ráp hệ thống, bố trí camera và sensor thực tế
Lưu Thanh Thuý	Phần mềm quản lý & báo cáo	Thiết kế dashboard, lập trình hiển thị và làm tài liệu

4.2 Timeline

Tuần	Công việc chính
1–2	Nghiên cứu tài liệu, xác định yêu cầu, phân công nhiệm vụ
3–4	Thu thập dữ liệu khuôn mặt/biển số, khảo sát thiết bị cần thiết
5–6	Triển khai mô hình AI nhận diện, kết nối cảm biến với ESP32
7–8	Thiết kế giao diện dashboard, xây dựng cơ sở dữ liệu trên cloud
9–10	Tích hợp toàn hệ thống: AI + IoT + giao diện
11	Chạy thử, hiệu chỉnh, test toàn bộ quy trình thực tế
12	Viết báo cáo, chuẩn bị slide thuyết trình, hoàn thiện demo

Cụ thể, xem timeline dạng [Gantt Chart](#) tại đây

4.3 Chi phí thực hiện (chi phí thiết bị, chi phí làm)

Tên thiết bị	Số lượng	Đơn giá	Thành tiền
AI Thinker ESP32-CAM	4	165,000	660,000
ESP32 DevKit V1 (CP2102, Type-C)	1	97,000	97,000

Tên thiết bị	Số lượng	Đơn giá	Thành tiền
Màn hình OLED 0.96 inch	2	55,000	110,000
Servo motor MG90S	2	37,000	74,000
Buzzer	2	5,000	10,000
Adapter 5V - 3A	1	52,000	52,000
Bộ chia nguồn DC male	1	26,000	26,000
Jack DC cái có dây 5.5x2.1mm	5	3,000	15,000
Breadboard	1	17,000	17,000
Dây jumper	40 sợi/loại	66,000	66,000
Đế Nạp ESP32-CAM	1	30,000	30,000
Module chuyển mạch I2C	1	33,000	33,000
Chi phí lắp đặt, khung mô hình	-	200,000	200,000
Tổng chi phí			1,390,000