

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 và Tính cấp thiết	4
1.2	Mục đích và Mục tiêu của Đề tài	4
1.2.1	Mục đích	4
1.3	Mục tiêu	5
1.3.1	Nhận diện phương tiện 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	Mô hình hệ thống và luồng hoạt động	8
2.1	Mô hình hệ thống	8
2.1.1	Thiết bị đầu vào (Input Layer)	8
2.1.2	Tầng xử lý trung tâm (Processing Layer)	8
2.1.3	Thiết bị điều khiển - phản hồi (Output Layer)	8
2.1.4	Nền tảng lưu trữ và giám sát	9
2.2	Luồng hoạt động chính của hệ thống	10
2.2.1	Entry flow (luồng hoạt động ở cổng vào)	10
2.2.2	Exit flow (luồng hoạt động ở cổng ra)	12
3	Công nghệ	12
3.1	Công nghệ sử dụng	12
3.1.1	Hệ thống phần cứng	12
3.1.2	Ứng dụng AI phân tích hình ảnh	20
3.1.3	Giao diện người dùng và quản trị (chưa quyết định)	20
3.1.4	Kết nối và nền tảng lưu trữ (chưa quyết định)	21

3.2	Thiết bị	21
4	Kế hoạch thực hiện	23
4.1	Phân công nhân lực	23
4.2	Timeline	23
4.3	Chi phí thực hiện (chi phí thiết bị, chi phí làm)	23

1 Giới thiệu

1.1 Bối cảnh và Tính cấp thiết

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 và Mục tiêu của Đề tài

1.2.1 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 phương tiện tự động

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

Các tiêu chí đánh giá:

- Tỷ lệ nhận diện chính xác biển số xe đạt trên 90% trong điều kiện ánh sáng ban ngày.
- Tỷ lệ nhận diện chính xác biển số xe 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 gồm ít nhất 100 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/video 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 nếu có) 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 (nếu có).
- 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, camera, cảm biến 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) và cảm biến IoT 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 Mô hình hệ thống và luồng hoạt động

2.1 Mô hình hệ thống

Hệ thống được thiết kế theo kiến trúc phân tầng gồm ba thành phần chính: tầng thiết bị đầu vào (ESP32-CAM), tầng xử lý trung tâm (AI model và backend Django), và tầng điều khiển - phản hồi (thiết bị IoT và Dashboard).

2.1.1 Thiết bị đầu vào (Input Layer)

- Hệ thống sử dụng các camera ESP32-CAM được lắp đặt tại cổng vào và cổng ra:
 - ESP32-CAM_1 và ESP32-CAM_3 dùng để quét **biển số xe**.
 - ESP32-CAM_2 và ESP32-CAM_4 dùng để quét **khuôn mặt tài xế**.
- Dữ liệu hình ảnh thu được từ các camera này sẽ được gửi đến mô hình AI để nhận dạng.

2.1.2 Tầng xử lý trung tâm (Processing Layer)

- **AI Model (local hoặc cloud):**
 - Nhận hình ảnh đầu vào từ ESP32-CAM.
 - Thực hiện xử lý nhận dạng khuôn mặt và biển số xe.
 - Trả về kết quả nhận diện (ID khuôn mặt, chuỗi biển số).
- **Backend Django:**
 - Nhận kết quả từ mô hình AI.
 - Kiểm tra tính hợp lệ và đối chiếu với dữ liệu đã lưu (đối với trường hợp xe ra).
 - Gửi lệnh điều khiển thiết bị (servo, buzzer, OLED).
 - Giao tiếp với Dashboard thông qua API để hiển thị thông tin trạng thái, nhật ký hệ thống.
 - Lưu dữ liệu tạm thời hoặc đồng bộ dữ liệu với hệ thống lưu trữ (Firebase/ThingsBoard).

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

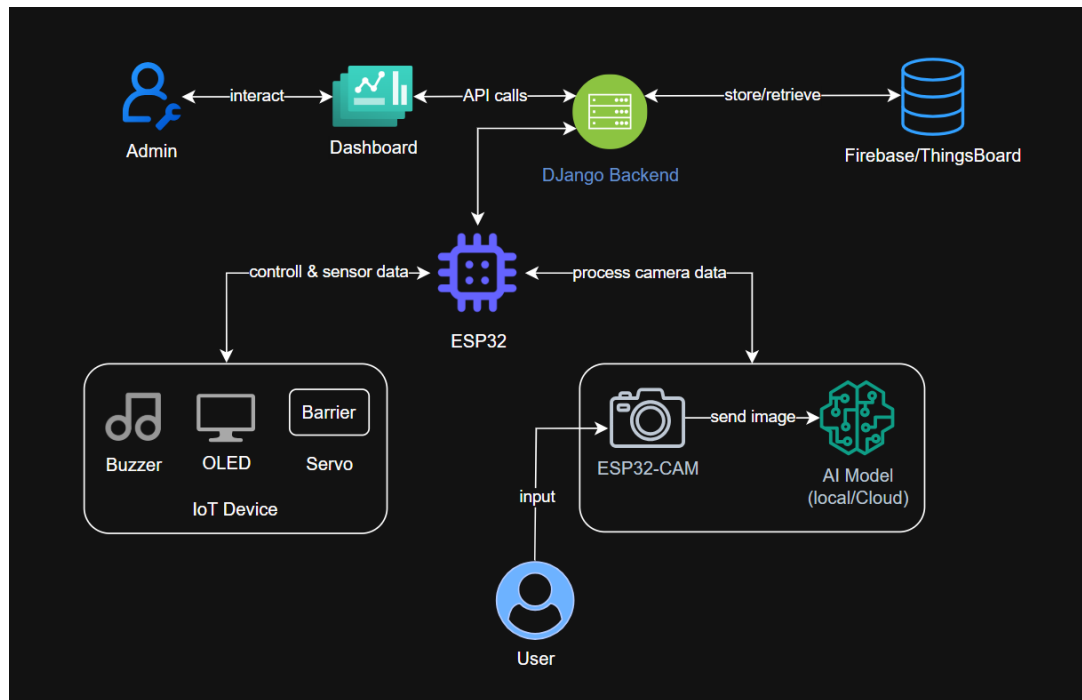
- **Servo (Barrier):** Tự động mở/đóng barrier cho xe ra vào sau khi xác thực.

- **OLED Display:** Hiển thị thông tin thời gian thực như số ô trống còn lại, lỗi nhận diện, cảnh báo...
- **Buzzer:** Phát âm báo tùy vào trạng thái: thành công, cảnh báo nhẹ, hoặc vi phạm.
- **Dashboard (giao diện web cho Admin):**
 - Hiển thị trạng thái thiết bị và số liệu thời gian thực.
 - Cho phép admin xem nhật ký vào/ra, giám sát trạng thái các camera và thiết bị IoT.
 - Tương tác trực tiếp với Backend qua các API.

2.1.4 Nền tảng lưu trữ và giám sát

- **Firebase hoặc ThingsBoard:**
 - Lưu trữ dữ liệu nhật ký hệ thống (log), trạng thái thiết bị, và thông tin người dùng.
 - Hỗ trợ hiển thị biểu đồ, số liệu và cảnh báo qua giao diện quản lý.

Tổng thể, hệ thống hoạt động theo nguyên tắc nhận diện kết hợp (biển số và khuôn mặt) nhằm đảm bảo độ chính xác và an toàn cao trong việc kiểm soát ra vào tại các khu vực như bãi xe, cổng tòa nhà, v.v.

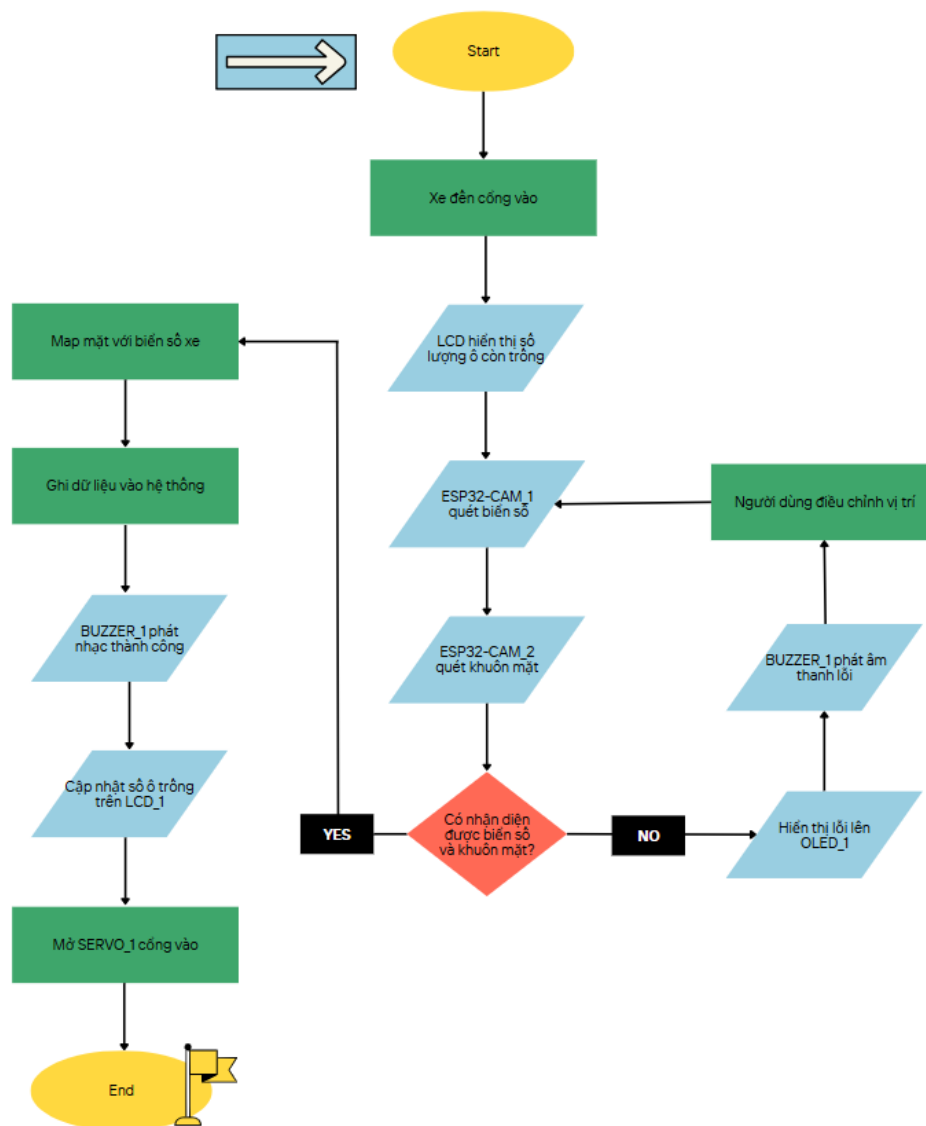


Hình 1: Sơ đồ kiến trúc hệ thống

2.2 Luồng hoạt động chính của hệ thống

2.2.1 Entry flow (luồng hoạt động ở cổng vào)

- ESP32-CAM_1 quét **biển số xe**.
- ESP32-CAM_2 quét **khuôn mặt tài xế**.
- Hệ thống xử lý và kiểm tra:
 - Nếu **cả hai đều nhận diện thành công**:
 - * Map dữ liệu: `face_id - license_plate`.
 - * Lưu vào cơ sở dữ liệu tạm thời.
 - * Servo mở barrier cho xe vào.
 - * Buzzer phát âm báo thành công.
 - * Màn hình OLED hiển thị số ô trống hiện tại.
 - Nếu **không nhận diện được**:
 - * Hiển thị thông báo lỗi trên màn hình OLED.
 - * Buzzer phát cảnh báo nhẹ.
 - * Chờ người dùng điều chỉnh vị trí xe để quét lại.



Hình 2: Flowchart cho entry flow

2.2.2 Exit flow (luồng hoạt động ở cổng ra)

- ESP32-CAM_3 quét **biển số xe**.
- ESP32-CAM_4 quét **khuôn mặt tài xế**.
- Hệ thống kiểm tra:
 - Biển số và khuôn mặt có được nhận diện không?
 - Có khớp với dữ liệu đã lưu khi vào không?
 - Nếu **có khớp**:
 - * Mở barrier cho xe ra.
 - * Buzzer phát âm báo thành công.
 - * Xóa dữ liệu mapping khỏi hệ thống.
 - * Cập nhật số lượng ô trống trên màn hình OLED ở cổng vào.
 - Nếu **không khớp**:
 - * Barrier không mở.
 - * Buzzer phát âm báo nguy hiểm để cảnh báo vi phạm.
 - * Màn hình OLED hiển thị thông tin lỗi.

3 Công nghệ

3.1 Công nghệ sử dụng

3.1.1 Hệ thống phần cứng

1. Camera ESP32-CAM (AI Thinker 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.

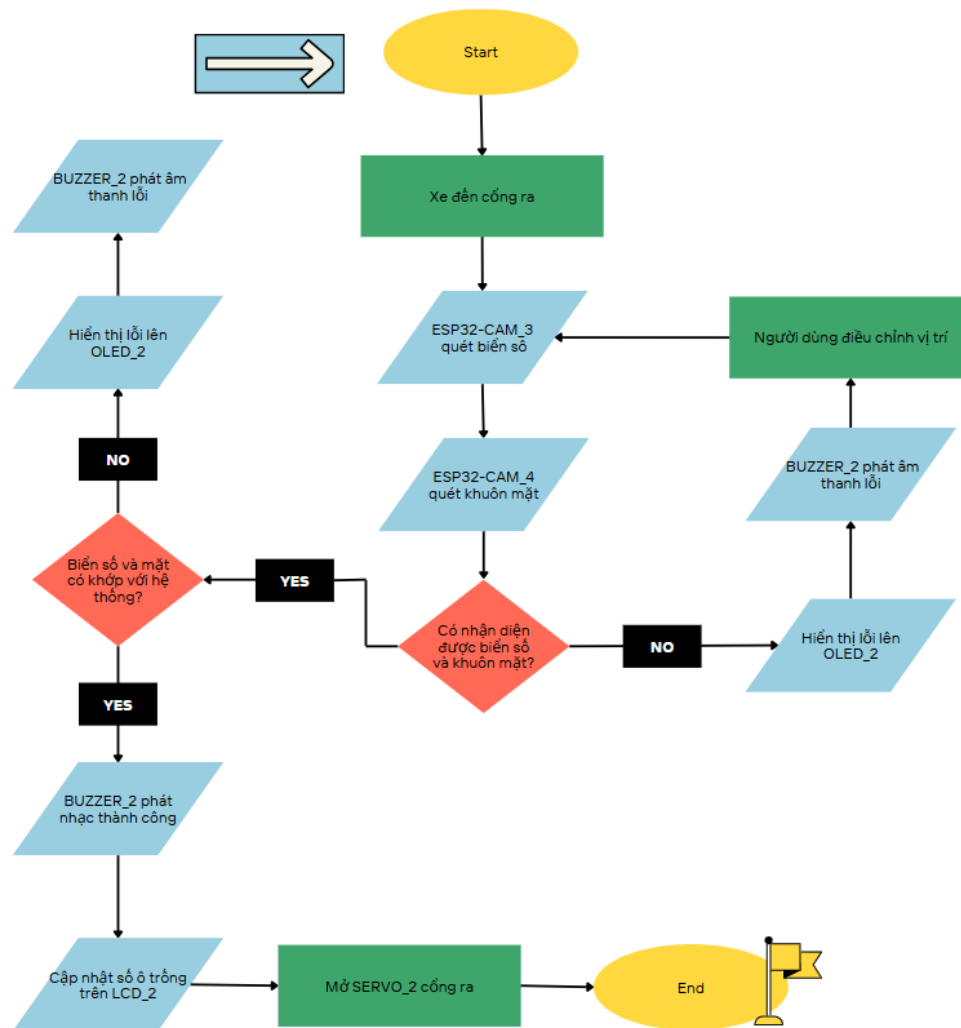
Module này được thiết kế chuyên dụng cho các ứng dụng thị giác máy tính, IoT, nhận diện khuôn mặt, giám sát an ninh, ...

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

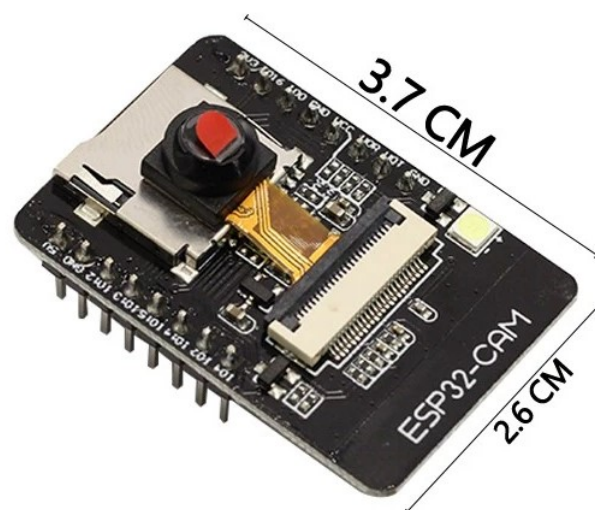
Thành phần	Thông tin chi tiết	Ghi chú
Vi xử lý chính	ESP32-WROOM-32	Xử lý chương trình nạp vào

Thành phần	Thông tin chi tiết	Ghi chú
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
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):



Hình 3: Flowchart cho exit flow



Hình 4: ESP32-CAM

- 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
- Gửi kết quả nhận diện kèm timestamp về ESP32 trung tâm qua giao thức như **HTTP POST** hoặc **MQTT**

ESP32 trung tâm và 4 ESP32-CAM sẽ kết nối cùng một mạng wifi nội bộ, ESP32 trung tâm sẽ chạy server nhẹ để lắng nghe và mỗi ESP32-CAM đóng vai trò client.

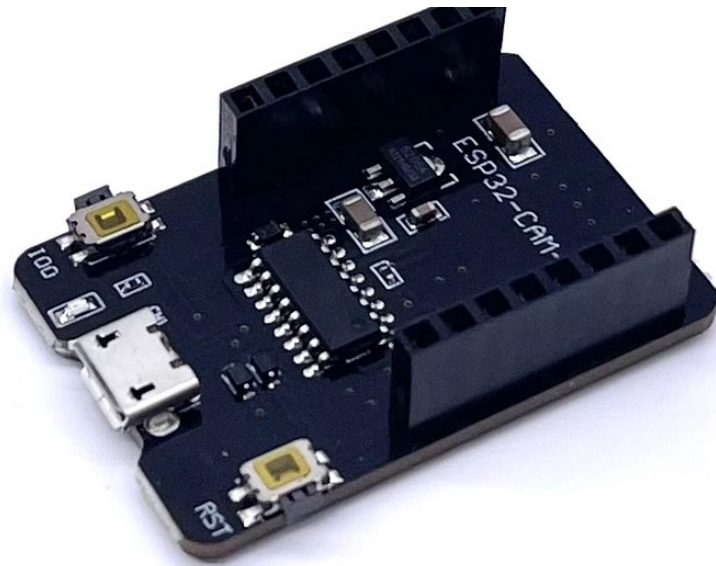
Cấp nguồn: ESP32-CAM khi bật camera và Wi-Fi có thể tiêu thụ **160–300mA**, vì vậy cần tránh cấp nguồn chung cho cả 5 module. Ta sẽ sử dụng **sạc dự phòng** để cấp nguồn riêng cho ESP32-CAM. Ưu tiên sử dụng loại sạc dự phòng có 2 cổng USB-A, mỗi ESP32-CAM sẽ được cấp nguồn từ 1 cổng, cụ thể:

- Dây đỏ (VCC): nối từ cổng USB-A của sạc dự phòng đến chân VCC/5V của ESP32-CAM
- Dây đen (GND): nối từ cổng USB-A của sạc dự phòng đến chân GND của ESP32-CAM
- Chân GND khác của ESP32-CAM sẽ nối dây đến GND rail của breadboard

Nạp code:

- 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.
- Ngoài ra, khi cắm cáp USB từ sạc dự phòng vào đế, ESP32-CAM sẽ được cấp nguồn điện, không cần dây USB-to-dupont.

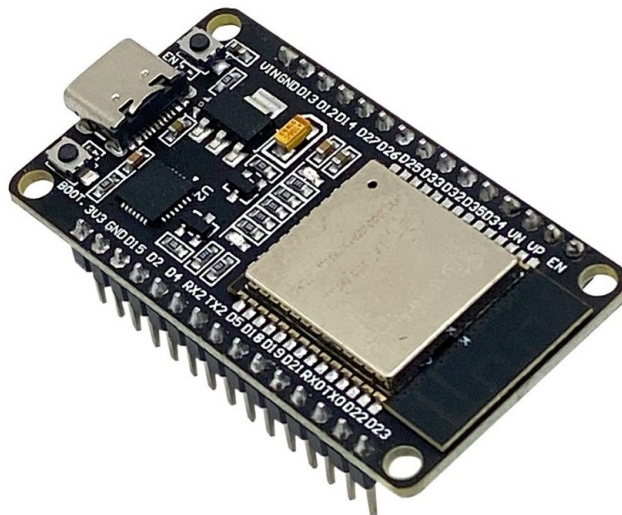
Lưu ý khi vừa nạp code, ESP32-CAM đang ở chế độ bootloader, cần ấn nút *reset* để chip khởi động lại chế độ chạy bình thường.



Hình 5: Đế nạp ESP32-CAM

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.

2. ESP32 trung tâm (ESP32 DevKit V1)



Hình 6: 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.

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. 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.

Cấp nguồn:

- Trong hệ thống này, breadboard sẽ được cấp nguồn trực tiếp từ adapter 5V – 3A thông qua cáp chia nguồn.
- Mỗi đầu ra được gắn jack DC cái chuyển sang dây jumper:
 - Dây đỏ (VCC) được nối vào rail 5V của breadboard
 - Dây đen (GND) được nối vào rail GND

- Dây jumper là loại dây cắm sẵn đầu, chuyên dùng để kết nối các thành phần trong mạch điện tử mà không cần hàn, đặc biệt là trong hệ thống kết nối qua 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/cảm 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

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

Chân OLED	Kết nối đến	Ghi chú
VCC	VCC của breadboard	Cấp điện 5V từ MB102 / breadboard

Chân OLED	Kết nối đến	Ghi chú
GND	GND của breadboard	Mass
SCL	GPIO22 của ESP32	Tạo xung đồng bộ, do ESP32 điều khiển, cho OLED biết khi nào nhận 1 bit
SDA	GPIO21 của ESP32	Truyền dữ liệu đọc và ghi

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.

Chân buzzer	Kết nối đến	Ghi chú
Chân dương (+)	5V từ breadboard (MB102)	Cấp nguồn hoạt động; không nên lấy từ ESP32
Chân âm (-)	GPIO bất kỳ của ESP32	Dùng để phát âm bằng tín hiệu <code>tone()</code>
GND chung	GND breadboard	GND của buzzer và ESP32 phải nối chung

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°).

Chân servo	Kết nối đến	Ghi chú
VCC (đỏ)	5V từ MB102 qua rail breadboard	Cấp nguồn chính
GND (nâu)	GND chung trên breadboard	Cực âm
Signal (vàng)	GPIO13 (hoặc bất kỳ) trên ESP32	Điều khiển PWM

3.1.2 Ứng dụng AI phân tích hình ảnh

- **Nhận diện khuôn mặt**

- **Phát hiện khuôn mặt:** Sử dụng mô hình **YOLOv6** để phát hiện vùng chứa khuôn mặt (bounding box) trong ảnh. Mô hình có tốc độ nhanh, độ chính xác cao, phù hợp xử lý ảnh thời gian thực.
- **Nhận diện khuôn mặt:** Dựa trên các vector đặc trưng được tạo bởi mô hình **FaceNet**, so sánh với database để xác định danh tính.

- **Nhận diện biển số xe**

- **Phát hiện biển số xe:** Sử dụng mô hình **YOLOv8** để phát hiện vùng chứa biển số xe trong ảnh.
- **Nhận dạng ký tự biển số:** Cắt vùng biển số từ ảnh dựa trên kết quả phát hiện, áp dụng thuật toán nhận dạng ký tự **Tesseract OCR** để trích xuất chuỗi ký tự.

3.1.3 Giao diện người dùng và quản trị (chưa quyết định)

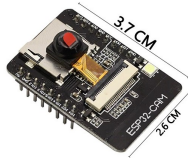



- **Frontend:** Sử dụng các công nghệ phổ biến HTML, CSS, JavaScript để xây dựng giao diện thân thiện, dễ sử dụng cho tài xế và quản trị viên.

- **Backend:** Sử dụng Django làm framework backend, xử lý các logic nghiệp vụ như xác thực người dùng, quản lý dữ liệu xe và vị trí bãi đỗ, cung cấp API cho frontend và thiết bị IoT kết nối.

3.1.4 Kết nối và nền tảng lưu trữ (chưa quyết định)

- MQTT (Message Queuing Telemetry Transport):
- Firebase:

3.2 Thiết bị

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

STT	Tên thiết bị	Số lượng	Giá/cái	Hình ảnh	Nguồn liên kết
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
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

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 Tổng hợp báo cáo, điều phối nhóm, hỗ trợ kỹ thuật Thu thập dữ liệu, huấn luyện mô hình, triển khai xử lý ảnh
Trần Thị Cát Tường	Phụ trách IoT – Cảm biến và cloud	Kết nối cảm biến, lập trình ESP32, gửi dữ liệu lên Firebase/ThingsBoard
Võ Lê Việt Tú	Thiết kế phần cứng – Mạch, cảm biến	Lắp ráp hệ thống, bố trí camera và sensor thực tế
Lưu Thanh Thuý	Giao diện & 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

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)	2	97,000	194,000
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
Module nguồn MB102	2	17,000	34,000
Breadboard	2	17,000	34,000
Dây jumper	40 sợi	66,000	66,000
Đế Nạp ESP32-CAM	4	30,000	120,000
Pin Sạc Dự Phòng	4	245,000	980,000
Vật tư lắp đặt, khung mô hình	-	-	200,000
Tổng chi phí			2,482,000