HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA CÔNG NGHỆ THÔNG TIN I**

**BỘ MÔN IOT VÀ ỨNG DỤNG**

**⎯⎯⎯⎯⎯⎯⎯⎯⎯**



**BÁO CÁO GIỮA KÌ**

**MÔN HỌC: IOT VÀ ỨNG DỤNG**

**Nhóm lớp học: Nhóm 02**

**Nhóm bài tập lớn: Nhóm 01**

**Đề tài: Xe trông nhà thông minh**

**Họ và tên Mã sinh viên**

**Nguyễn Mạnh Cường B22DCCN100**

**Trần Đức Phương B22DCCN640**

**Đỗ Thành Đạt B21DCCN184**

**Bùi Văn Hiến B22DCCN292**

**Giảng viên giảng dạy: Kim Ngọc Bách**

**HÀ NỘI – 2025**

Nội dung

[A. Xác định mục tiêu và phạm vi hệ thống 3](#_Toc211108494)

[I.Mục tiêu hệ thống 3](#_Toc211108495)

[1. Vấn đề thực tế 3](#_Toc211108496)

[2. Mục tiêu IoT 3](#_Toc211108497)

[3. Kỳ vọng 4](#_Toc211108498)

[II.Phạm vi triển khai 4](#_Toc211108499)

[1.Số lượng thiết bị: 4](#_Toc211108500)

[2.Môi trường hoạt động: 4](#_Toc211108501)

[3.Khu vực thử nghiệm: 4](#_Toc211108502)

[III.Tiêu chí thành công 4](#_Toc211108503)

[1.Độ chính xác: 4](#_Toc211108504)

[2.Độ trễ: 5](#_Toc211108505)

[3.Độ tin cậy: 5](#_Toc211108506)

[4.Hiệu suất hệ thống: 5](#_Toc211108507)

[5.Chi phí: 5](#_Toc211108508)

[6.Khả năng mở rộng: 5](#_Toc211108509)

[IV.Kết quả mong đợi 5](#_Toc211108510)

[B. Thu thập yêu cầu từ các bên liên quan 6](#_Toc211108511)

[I. Bối cảnh vấn đề 6](#_Toc211108512)

[II. Yêu cầu từ các bên liên quan 6](#_Toc211108513)

[1. Người dùng cuối (Nhân viên bảo vệ & Quản lý an ninh) 6](#_Toc211108514)

[2. Quản lý kỹ thuật (Đội ngũ phát triển & Bảo trì) 7](#_Toc211108515)

[3. Ban lãnh đạo (Giám đốc & Quản lý cấp cao) 7](#_Toc211108516)

[III. Phương pháp thu thập yêu cầu 8](#_Toc211108517)

[1.Phỏng vấn: 8](#_Toc211108518)

[2.Khảo sát: 8](#_Toc211108519)

[3.Quan sát thực tế: 8](#_Toc211108520)

[4.Nghiên cứu tài liệu: 8](#_Toc211108521)

[5.Thử nghiệm pilot: 8](#_Toc211108522)

[C. Xác định yêu cầu chức năng 9](#_Toc211108523)

[I. Các chức năng cần có của hệ thống 9](#_Toc211108524)

[1.Thu thập dữ liệu: 9](#_Toc211108525)

[2.Truyền dữ liệu: 9](#_Toc211108526)

[3.Xử lý và phân tích dữ liệu: 9](#_Toc211108527)

[4.Điều khiển/Ra lệnh: 9](#_Toc211108528)

[5.Cảnh báo: 9](#_Toc211108529)

[II.Đặc tả công việc 10](#_Toc211108530)

[D.Xác định yêu cầu phi chức năng 10](#_Toc211108531)

[I.Các yêu cầu phi chức năng 10](#_Toc211108532)

[1. Hiệu năng 10](#_Toc211108533)

[2. Bảo mật 11](#_Toc211108534)

[3. Độ tin cậy 11](#_Toc211108535)

[4. Khả năng mở rộng 11](#_Toc211108536)

[5. Chi phí và Năng lượng 12](#_Toc211108537)

[7. Khả năng bảo trì 12](#_Toc211108538)

[E. Ràng buộc pháp lý 12](#_Toc211108539)

[F. Phân tích ràng buộc kỹ thuật và môi trường 13](#_Toc211108540)

[I. Môi trường hoạt động 13](#_Toc211108541)

[II. Ràng buộc pháp lý 14](#_Toc211108542)

[III. Tài nguyên thiết bị 14](#_Toc211108543)

[G. Các lý thuyết và công nghệ áp dụng 15](#_Toc211108544)

[I. Kiến trúc IoT 15](#_Toc211108545)

[II. Trí tuệ nhân tạo và Thị giác máy tính 15](#_Toc211108546)

[III. Hệ thống nhúng và Vi điều khiển 16](#_Toc211108547)

[IV. Giao thức truyền thông 16](#_Toc211108548)

[V. Cảm biến và Điều khiển 17](#_Toc211108549)

[VI. Thuật toán và Xử lý 18](#_Toc211108550)

[VII. Cập nhật Firmware OTA 18](#_Toc211108551)

[VIII. Bảo mật 19](#_Toc211108552)

[IX. Công nghệ phát triển ứng dụng 19](#_Toc211108553)

# A. Xác định mục tiêu và phạm vi hệ thống

## I.Mục tiêu hệ thống

### 1. Vấn đề thực tế

Hiện nay, công tác giám sát và tuần tra an ninh gặp nhiều hạn chế:

* Chi phí nhân lực cao: Cần nhiều nhân viên bảo vệ làm việc 24/7 để giám sát các khu vực quan trọng.
* Hạn chế con người: Nhân viên có thể mệt mỏi, mất tập trung, không quan sát được toàn diện.
* Nguy hiểm: Tuần tra các khu vực nguy hiểm (nhà máy, kho hóa chất) gây rủi ro cho nhân viên.
* Phát hiện chậm: Camera truyền thống chỉ ghi hình, không tự động phát hiện và cảnh báo xâm nhập.
* Thiếu linh hoạt: Camera cố định không thể di chuyển để theo dõi hoặc khảo sát nhiều khu vực.

Do đó, cần một giải pháp giám sát thông minh, tự động, có khả năng di chuyển và phát hiện xâm nhập kịp thời.

### 2. Mục tiêu IoT

* Giám sát tự động: Thu thập và theo dõi các thông số quan trọng như khoảng cách đến vật cản, mức pin, tốc độ di chuyển, và truyền video real-time về laptop.
* Điều khiển từ xa đa kênh: Điều khiển xe qua Bluetooth (smartphone) cho chế độ Manual, đồng thời nhận lệnh cảnh báo từ laptop qua WiFi khi AI phát hiện người.
* Navigation tự động thông minh: Tự động tránh vật cản dựa trên dữ liệu cảm biến siêu âm, điều chỉnh tốc độ và hướng di chuyển phù hợp với môi trường.
* Nhận diện và cảnh báo AI: Sử dụng YOLO trên laptop để nhận diện người trong video stream, tự động kích hoạt cảnh báo âm thanh trên cả laptop và xe khi phát hiện xâm nhập.
* Cập nhật firmware từ xa: Hỗ trợ OTA (Over-The-Air) update qua WiFi, cho phép nâng cấp và sửa lỗi mà không cần can thiệp vật lý.

### 3. Kỳ vọng

* Triển khai tại kho hàng, văn phòng, khu công nghiệp
* Giảm 30-50% chi phí nhân lực giám sát
* Phát hiện và cảnh báo xâm nhập kịp thời
* Giám sát từ xa các khu vực nguy hiểm

## II.Phạm vi triển khai

### 1.Số lượng thiết bị:

* 1 xe robot tự hành (prototype) tích hợp ESP32-CAM, động cơ DC, cảm biến siêu âm, và module cảnh báo.
* 1 laptop xử lý AI (YOLO) để nhận diện người từ video stream.
* 1 smartphone (Android/iOS) để điều khiển xe qua Bluetooth.
* 1 router WiFi hoặc laptop hotspot để kết nối ESP32-CAM với laptop.

### 2.Môi trường hoạt động:

* Ngoài trời (outdoor) → điều kiện thời tiết bình thường, nhiệt độ 20-35°C, không mưa, ánh sáng tự nhiên ban ngày.
* Yêu cầu: Mặt đất tương đối phẳng (sân, vỉa hè, sân trường), không có vật cản phức tạp, khoảng cách WiFi <15m để đảm bảo streaming ổn định.
* Nguồn điện: Pin Li-Po 7.4V (2200mAh) cho phép hoạt động liên tục 1-2 giờ.
* Lưu ý: Các linh kiện điện tử cần được bảo vệ khỏi bụi bẩn và nắng trực tiếp, tránh nhiệt độ quá cao ảnh hưởng đến ESP32-CAM và pin.

### 3.Khu vực thử nghiệm:

* Diện tích: 50-100m² (sân trường, sân nhà, bãi đỗ xe).
* Điều kiện: Có nguồn điện để sạc pin, WiFi ổn định hoặc sử dụng laptop hotspot di động, đủ không gian cho xe di chuyển và thử nghiệm chế độ tự động tránh vật cản.

## III.Tiêu chí thành công

### 1.Độ chính xác:

* Độ chính xác nhận diện người (AI) > 85%.
* Sai số đo khoảng cách cảm biến siêu âm < ±2cm.
* Tỷ lệ false positive (cảnh báo nhầm) < 10%.

### 2.Độ trễ:

* Độ trễ video streaming < 1s.
* Độ trễ điều khiển Bluetooth (Manual mode) < 200ms.
* Thời gian phản ứng cảnh báo khi phát hiện người < 1s.

### 3.Độ tin cậy:

* Tỷ lệ truyền video thành công > 95%.
* Xe tránh vật cản thành công 100%, không va chạm trong chế độ Auto.
* Kết nối WiFi/Bluetooth ổn định, tự động reconnect khi mất kết nối.

### 4.Hiệu suất hệ thống:

* FPS video streaming: 10-15 frames/giây.
* FPS xử lý AI trên laptop > 10 frames/giây.
* Thời gian hoạt động liên tục: 1-2 giờ với 1 lần sạc pin.

### 5.Chi phí:

* Tổng chi phí đầu tư ban đầu cho 1 xe prototype < 2 triệu VNĐ (bao gồm ESP32-CAM, động cơ, cảm biến, khung xe, pin).

### 6.Khả năng mở rộng:

* Kiến trúc sẵn sàng mở rộng thành đội robot tuần tra (multi-robot system) trong tương lai.

## IV.Kết quả mong đợi

* Tự động hóa giám sát an ninh → giảm 30-50% chi phí nhân lực bảo vệ, tăng hiệu quả tuần tra 24/7.
* Phát hiện xâm nhập thông minh → cảnh báo kịp thời khi phát hiện người trong khu vực cấm, ghi lại bằng chứng video real-time.
* Điều khiển linh hoạt từ xa → người vận hành có thể giám sát và điều khiển xe qua smartphone, theo dõi trạng thái (pin, khoảng cách, chế độ hoạt động) mọi lúc mọi nơi.
* Nền tảng nghiên cứu IoT & AI → dữ liệu video thu thập liên tục giúp cải thiện thuật toán AI, phát triển thêm tính năng nhận diện khuôn mặt, theo dõi đối tượng, phân tích hành vi.
* Giải pháp an toàn → giám sát các khu vực nguy hiểm (nhà máy, kho hóa chất) mà không cần con người tiếp cận trực tiếp.
* Khả năng mở rộng → làm nền tảng để phát triển hệ thống đội robot tuần tra tự động, tích hợp với hệ thống báo động và camera giám sát hiện có.

# B. Thu thập yêu cầu từ các bên liên quan

## I. Bối cảnh vấn đề

* Các khu vực cần giám sát an ninh như kho hàng, nhà máy, văn phòng thường có diện tích rộng với nhiều góc khuất.
* Hiện nay, việc giám sát chủ yếu dựa vào nhân viên bảo vệ tuần tra thủ công hoặc camera cố định → dễ bỏ sót khu vực, phát hiện xâm nhập chậm trễ, chi phí nhân lực cao, nguy hiểm khi tuần tra khu vực nguy hiểm.
* Hệ thống camera giám sát truyền thống chỉ ghi hình thụ động, không có khả năng di chuyển linh hoạt hay phát hiện và cảnh báo tự động khi có người xâm nhập trái phép.
* Ban quản lý muốn ứng dụng IoT kết hợp AI để xây dựng robot tuần tra tự động, có khả năng nhận diện người xâm nhập, cảnh báo kịp thời và giảm thiểu rủi ro cho nhân viên bảo vệ.

## II. Yêu cầu từ các bên liên quan

### 1. Người dùng cuối (Nhân viên bảo vệ & Quản lý an ninh)

Hiển thị:

* Muốn xem video streaming real-time từ xe trên laptop hoặc smartphone để giám sát khu vực.
* Xem số lượng người phát hiện, vị trí người (bounding box), confidence score của AI.

Cảnh báo:

* Nhận thông báo tức thì (âm thanh + visual alert) khi AI phát hiện người xâm nhập trong khu vực giám sát.
* Cảnh báo khi xe gặp sự cố (mất kết nối WiFi/Bluetooth, vật cản chặn đường quá lâu).

Điều khiển:

* Có quyền điều khiển xe từ xa qua smartphone (chế độ Manual): tiến/lùi/rẽ trái/rẽ phải.
* Có nút Emergency Stop để dừng xe khẩn cấp từ xa khi phát hiện rủi ro.
* Chuyển đổi chế độ Manual/Auto linh hoạt theo tình huống.

### 2. Quản lý kỹ thuật (Đội ngũ phát triển & Bảo trì)

Giám sát hệ thống:

* Theo dõi telemetry data (FPS video, FPS AI, độ trễ mạng, nhiệt độ ESP32-CAM) để đánh giá hiệu suất.

Bảo trì và nâng cấp:

* Hỗ trợ OTA firmware update để sửa lỗi và nâng cấp tính năng từ xa mà không cần thu hồi xe.
* Khả năng rollback firmware tự động nếu update thất bại để đảm bảo hệ thống luôn hoạt động.

Mở rộng:

* Kiến trúc cho phép thêm cảm biến mới (nhiệt độ, độ ẩm, GPS, gas sensor) mà không cần thay đổi firmware lớn.
* Hỗ trợ tích hợp với hệ thống báo động và camera giám sát hiện có.

### 3. Ban lãnh đạo (Giám đốc & Quản lý cấp cao)

Hiệu quả kinh tế:

* Giảm 30-50% chi phí nhân lực bảo vệ bằng cách sử dụng robot tuần tra tự động.
* Tăng độ phủ giám sát mà không cần đầu tư thêm camera cố định đắt tiền.

Báo cáo và phân tích:

* Dashboard tổng quan: tổng số lần phát hiện xâm nhập, thời gian hoạt động, tỷ lệ cảnh báo đúng/sai.
* Báo cáo định kỳ (ngày/tuần/tháng) về tình hình an ninh, xu hướng xâm nhập.

Khả năng mở rộng:

* Hệ thống có thể nhân rộng thành đội robot tuần tra (fleet management) trong tương lai.
* Nền tảng nghiên cứu để phát triển thêm tính năng AI nâng cao (nhận diện khuôn mặt, phân tích hành vi).

## III. Phương pháp thu thập yêu cầu

### 1.Phỏng vấn:

* Làm việc với nhân viên bảo vệ và quản lý an ninh để hiểu rõ khó khăn trong công tác tuần tra, giám sát (khu vực khó tiếp cận, thời gian tuần tra, tần suất phát hiện xâm nhập).
* Trao đổi với đội ngũ kỹ thuật về khả năng tích hợp với hệ thống camera và báo động hiện có.

### 2.Khảo sát:

* Gửi bảng hỏi cho ban quản lý để thu thập nhu cầu về tính năng (chế độ tự động/thủ công, loại cảnh báo, báo cáo an ninh).
* Thu thập ý kiến về ngân sách đầu tư, thời gian triển khai và yêu cầu đào tạo sử dụng.

### 3.Quan sát thực tế:

* Đội ngũ phát triển đến trực tiếp khu vực giám sát (kho hàng, sân trường, nhà máy) để:
  + Khảo sát địa hình (mặt đất phẳng/gồ ghề, vật cản cố định).
  + Kiểm tra phủ sóng WiFi, điểm mù tín hiệu.
  + Đánh giá điều kiện ánh sáng (ban ngày/ban đêm) ảnh hưởng đến camera và AI.
  + Xác định vị trí đặt trạm sạc pin và điểm xuất phát của xe.

### 4.Nghiên cứu tài liệu:

* Tham khảo các dự án robot tuần tra, xe tự hành tương tự để học hỏi kinh nghiệm.
* Nghiên cứu tiêu chuẩn an ninh và quy định pháp lý về giám sát tự động.

### 5.Thử nghiệm pilot:

* Triển khai prototype trong môi trường thực tế với quy mô nhỏ (1 xe, 1 khu vực).
* Thu thập feedback từ người dùng để cải tiến tính năng trước khi triển khai rộng.

# C. Xác định yêu cầu chức năng

## I. Các chức năng cần có của hệ thống

### 1.Thu thập dữ liệu:

* Thu thập video/hình ảnh real-time từ camera OV2640 trên ESP32-CAM.
* Đo khoảng cách đến vật cản liên tục từ cảm biến siêu âm HC-SR04.

### 2.Truyền dữ liệu:

* Streaming video về laptop qua WiFi (MJPEG format, 10-15 FPS).
* Nhận lệnh điều khiển từ smartphone qua Bluetooth.
* Nhận lệnh cảnh báo từ laptop qua WiFi/HTTP.

### 3.Xử lý và phân tích dữ liệu:

* Xử lý AI trên laptop: nhận diện người từ video stream bằng YOLO.
* Xử lý logic tránh vật cản trên ESP32-CAM dựa trên dữ liệu cảm biến siêu âm.
* Phân tích và lọc dữ liệu: loại bỏ nhiễu cảm biến, áp dụng NMS (Non-Maximum Suppression) cho kết quả AI.

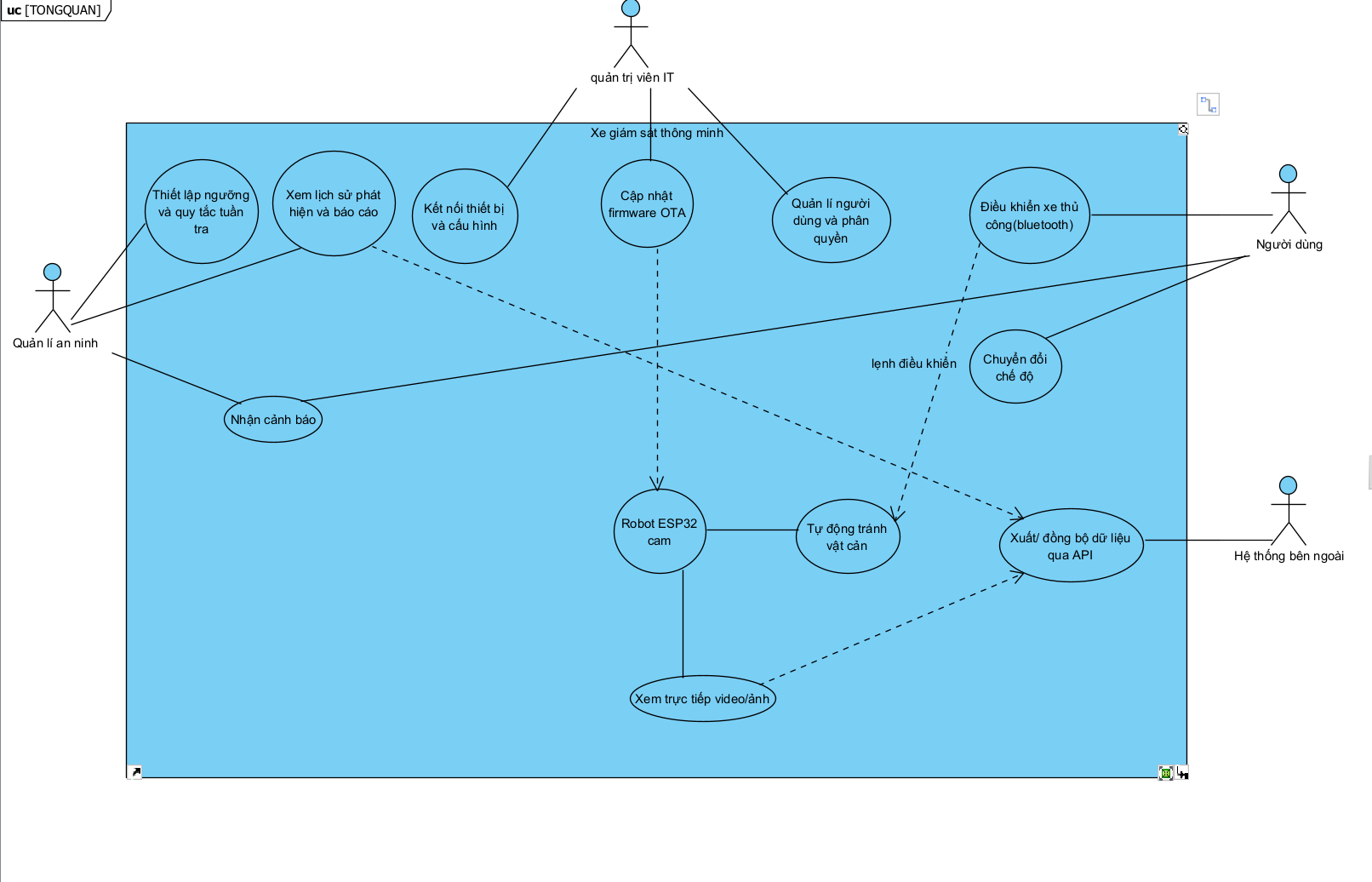
### 4.Điều khiển/Ra lệnh:

* Điều khiển di chuyển xe (tiến/lùi/trái/phải/dừng) từ smartphone qua Bluetooth.
* Chuyển đổi chế độ Manual/Auto từ app.
* Gửi lệnh Emergency Stop từ app hoặc laptop.
* Laptop gửi lệnh cảnh báo (bật/tắt buzzer) về ESP32-CAM khi phát hiện/mất người.
* Cập nhật firmware từ xa qua OTA.

### 5.Cảnh báo:

* Kích hoạt buzzer/loa trên xe khi nhận lệnh cảnh báo từ laptop.
* Hiển thị visual alert (màu đỏ, chữ "ALERT!") trên UI laptop/app.

## II.Đặc tả công việc



# D.Xác định yêu cầu phi chức năng

## I.Các yêu cầu phi chức năng

### 1. Hiệu năng

Độ trễ tối đa:

* Truyền video: độ trễ < 500ms từ ESP32-CAM đến laptop
* Điều khiển Bluetooth: độ trễ < 200ms từ khi nhấn nút đến khi xe phản ứng
* Cảnh báo AI: thời gian từ phát hiện người đến kích hoạt còi < 500ms
* Cập nhật firmware OTA: tốc độ tải lên tối thiểu 50KB/s

Tần suất lấy mẫu:

* Camera: 10-15 khung hình/giây (FPS)
* Cảm biến siêu âm: đo khoảng cách mỗi 100ms (10 lần/giây)
* Dữ liệu giám sát: gửi mỗi 1-2 giây (pin, khoảng cách, tốc độ, chế độ)
* Xử lý AI: xử lý tối thiểu 10 khung hình/giây để đảm bảo phát hiện kịp thời

Khả năng chịu tải:

* Hệ thống phải xử lý được 1 luồng video HD (640x480) đồng thời với điều khiển và giám sát
* Laptop AI phải xử lý được YOLO với mức sử dụng CPU < 70%
* ESP32-CAM phải xử lý được đồng thời: truyền video, Bluetooth, WiFi, đọc cảm biến, điều khiển động cơ
* Băng thông WiFi tối thiểu: 2 Mbps cho truyền video ổn định

### 2. Bảo mật

Mã hóa:

* Kết nối WiFi: sử dụng mã hóa WPA2/WPA3
* Bluetooth: ghép nối bảo mật với mã PIN hoặc khóa truy cập
* Cập nhật firmware OTA: xác thực checksum (MD5/SHA256) trước khi cài đặt
* Truyền dữ liệu nhạy cảm: sử dụng HTTPS/TLS

### 3. Độ tin cậy

Cơ chế dự phòng:

* Dừng khẩn cấp tự động khi phát hiện vật cản < 20cm (không phụ thuộc WiFi/Bluetooth)
* Bộ đếm thời gian giám sát: tự động khởi động lại ESP32-CAM nếu firmware bị treo
* Giám sát pin: cảnh báo và dừng xe an toàn khi pin < 20%
* Khôi phục OTA: tự động quay lại firmware cũ nếu cập nhật thất bại hoặc khởi động không thành công

### 4. Khả năng mở rộng

Hỗ trợ thêm thiết bị mới:

* Kiến trúc module cho phép thêm cảm biến mới (nhiệt độ, độ ẩm, khí gas, GPS) chỉ bằng cách thêm module firmware
* Các chân GPIO dự phòng trên ESP32-CAM để kết nối thêm thiết bị
* Bus I2C/SPI có thể kết nối nhiều cảm biến cùng lúc
* Firmware hỗ trợ đăng ký cảm biến động (không cần biên dịch lại)

### 5. Chi phí và Năng lượng

Quản lý năng lượng:

* Pin Li-Po 7.4V 2200mAh: đủ cho 1-2 giờ hoạt động liên tục
* Chế độ ngủ khi không sử dụng: giảm 80% tiêu thụ điện
* Tự động giảm khung hình camera khi không cần thiết (từ 15 FPS xuống 5 FPS)
* Đèn LED báo hiệu tắt tự động sau 30 giây để tiết kiệm pin
* Chế độ tiết kiệm năng lượng: tắt truyền video WiFi khi chỉ cần điều khiển Bluetooth

6. Khả năng sử dụng

Dễ sử dụng:

* Ứng dụng di động: giao diện trực quan, dễ hiểu, thời gian học < 5 phút
* Giao diện laptop: hiển thị rõ ràng video, trạng thái, cảnh báo
* Kết nối một chạm: tự động tìm và kết nối ESP32-CAM
* Phản hồi trực quan: đèn LED, còi báo, màu sắc giao diện để người dùng biết trạng thái

### 7. Khả năng bảo trì

Dễ bảo trì:

* Mã nguồn có chú thích đầy đủ, tuân thủ quy ước lập trình
* Thiết kế module: mỗi module độc lập, dễ sửa lỗi
* Hệ thống nhật ký: ghi lại các sự kiện quan trọng để gỡ lỗi
* Cập nhật OTA: sửa lỗi và nâng cấp từ xa mà không cần thu hồi thiết bị

# E. Ràng buộc pháp lý

Quy định tần số vô tuyến:

* WiFi 2.4GHz và Bluetooth được phép sử dụng tự do tại Việt Nam theo Thông tư 26/2016/TT-BTTTT với công suất phát tối đa 100mW. ESP32-CAM mặc định tuân thủ tiêu chuẩn này, không cần xin giấy phép sử dụng tần số.

Bảo mật dữ liệu cá nhân:

* Theo Luật An ninh mạng 2018 và Nghị định 13/2023/NĐ-CP, hệ thống ghi hình và nhận diện người cần có biển báo "Khu vực có camera giám sát" và chỉ lưu trữ video trong thời gian cần thiết (khuyến nghị < 30 ngày). Giải pháp: không lưu video dài hạn, chỉ streaming real-time, không sử dụng nhận diện khuôn mặt, mã hóa dữ liệu khi truyền.

Quyền riêng tư và giám sát:

* Theo Luật Dân sự 2015, không được giám sát khu vực riêng tư và cần có sự đồng ý của chủ sở hữu. Giải pháp: chỉ triển khai tại khu vực được phép (kho hàng, sân trường, khu công nghiệp) và có văn bản cho phép từ ban quản lý.

An toàn lao động:

* Theo Nghị định 44/2016/NĐ-CP, robot di động phải đảm bảo an toàn cho người xung quanh. Giải pháp: giới hạn tốc độ tối đa 0.5-1 m/s, cảm biến tránh vật cản bắt buộc, còi báo khi di chuyển, và nút dừng khẩn cấp.

# F. Phân tích ràng buộc kỹ thuật và môi trường

## I. Môi trường hoạt động

Nhiệt độ:

* Nhiệt độ hoạt động tối ưu: 20-35°C (môi trường ngoài trời điều kiện bình thường). ESP32-CAM có thể quá nhiệt khi streaming video liên tục ở nhiệt độ > 40°C, cần sử dụng tản nhiệt nhôm và tự động giảm FPS khi phát hiện quá nóng.

Độ ẩm:

* Độ ẩm hoạt động: 30-70% RH. Các linh kiện điện tử không chống nước có thể bị oxy hóa hoặc ngắn mạch ở độ ẩm cao, cần đóng gói trong hộp nhựa/acrylic và sử dụng gói hút ẩm silica gel.

Nhiễu sóng:

* Khoảng cách WiFi/Bluetooth bị giới hạn: 10-15m cho WiFi, 5-10m cho Bluetooth trong môi trường có vật cản. Nhiễu điện từ từ động cơ DC và thiết bị 2.4GHz khác có thể ảnh hưởng chất lượng tín hiệu, cần sử dụng tụ lọc chống nhiễu và chọn kênh WiFi ít chen lấn.

Nguồn cấp:

* Thời gian hoạt động: 1-2 giờ với pin Li-Po 7.4V 2200mAh. Cần sử dụng module BMS để bảo vệ pin, giám sát điện áp liên tục và tự động dừng xe khi pin < 15% để tránh xả sâu gây hỏng pin.

Ánh sáng:

* Chỉ hoạt động ban ngày hoặc môi trường có đủ ánh sáng (> 50 lux). Camera OV2640 không có đèn hồng ngoại nên chất lượng hình ảnh giảm mạnh trong điều kiện thiếu sáng, có thể bổ sung đèn LED trắng nếu cần hoạt động trong môi trường tối.

## II. Ràng buộc pháp lý

Quy định tần số vô tuyến:

* Sử dụng tự do không cần giấy phép. WiFi 2.4GHz và Bluetooth được phép sử dụng tại Việt Nam theo Thông tư 26/2016/TT-BTTTT với công suất tối đa 100mW, ESP32-CAM mặc định tuân thủ.

An toàn lao động:

* Tốc độ tối đa 0.5-1 m/s và có cảm biến tránh vật cản bắt buộc. Theo Nghị định 44/2016/NĐ-CP, robot phải đảm bảo an toàn với cơ chế dừng khẩn cấp khi phát hiện người/vật < 20cm.

## III. Tài nguyên thiết bị

Bộ nhớ ESP32-CAM:

* RAM 520KB + 4MB PSRAM không đủ để chạy AI. Bộ nhớ phải chia sẻ cho WiFi, Bluetooth, camera buffer và chương trình, do đó AI phải xử lý trên laptop thay vì trên ESP32-CAM.

CPU ESP32-CAM:

* Dual-core 240MHz không đủ mạnh cho xử lý AI real-time. Core 0 dành riêng cho WiFi/Bluetooth, Core 1 xử lý application, cần sử dụng RTOS task để quản lý đa nhiệm và ưu tiên task quan trọng.

GPIO giới hạn:

* Chỉ còn ~9 GPIO khả dụng sau khi camera và flash chiếm dụng. Cần thiết kế mạch cẩn thận để phân bổ GPIO cho động cơ, cảm biến, buzzer, và dự phòng cho mở rộng sau này.

# G. Các lý thuyết và công nghệ áp dụng

## I. Kiến trúc IoT

Kiến trúc 3 tầng (Three-tier Architecture):

* Tầng Perception (Cảm nhận): ESP32-CAM, camera OV2640, cảm biến siêu âm HC-SR04 thu thập dữ liệu video và khoảng cách từ môi trường thực tế.
* Tầng Network (Mạng): WiFi 802.11n (2.4GHz) truyền video streaming, Bluetooth 4.2 truyền lệnh điều khiển, HTTP/JSON protocol trao đổi dữ liệu giữa ESP32-CAM và laptop.
* Tầng Application (Ứng dụng): Laptop xử lý AI, mobile app điều khiển, dashboard hiển thị trạng thái và cảnh báo.

Edge Computing:

* Xử lý phân tán: logic tránh vật cản, điều khiển động cơ, đọc cảm biến được xử lý ngay trên ESP32-CAM (edge device) để giảm độ trễ và không phụ thuộc vào kết nối mạng.
* Cloud/Server processing: các tác vụ nặng như AI nhận diện người được xử lý trên laptop (computing node) có CPU/GPU mạnh hơn.

## II. Trí tuệ nhân tạo và Thị giác máy tính

YOLO (You Only Look Once) - Object Detection:

* Mô hình CNN (Convolutional Neural Network) để nhận diện đối tượng real-time trong video.
* Sử dụng YOLOv5 hoặc YOLOv8 đã được pre-trained trên dataset COCO (80 classes bao gồm class "person").
* Single-stage detector: xử lý cả bounding box và classification trong một lần forward pass, tốc độ nhanh (10-30 FPS).
* NMS (Non-Maximum Suppression): loại bỏ các bounding box trùng lặp, chỉ giữ lại detection có confidence cao nhất.

Computer Vision với OpenCV:

* Frame processing: decode MJPEG stream, convert color space (JPEG → BGR), resize/crop frame.
* Drawing utilities: vẽ bounding box, text label, màu cảnh báo trên video.
* Video I/O: đọc/ghi video stream, hiển thị trên màn hình.

Image Processing trên ESP32-CAM:

* JPEG encoding: nén frame từ raw format sang JPEG để giảm dung lượng truyền qua WiFi (từ ~300KB xuống ~30-50KB/frame).
* Image quality adjustment: điều chỉnh brightness, contrast, saturation tự động theo điều kiện ánh sáng.

## III. Hệ thống nhúng và Vi điều khiển

ESP32 Microcontroller:

* Dual-core Xtensa LX6 32-bit @ 240MHz: core 0 cho WiFi/Bluetooth stack, core 1 cho application.
* FreeRTOS: hệ điều hành thời gian thực để quản lý đa nhiệm (task scheduling, semaphore, queue).
* DMA (Direct Memory Access): truyền dữ liệu camera trực tiếp vào bộ nhớ mà không qua CPU, giảm tải xử lý.

PWM (Pulse Width Modulation):

* Điều khiển tốc độ động cơ DC: thay đổi duty cycle (0-100%) để điều chỉnh tốc độ quay.
* Tần số PWM: 1-20 kHz, duty cycle càng cao thì động cơ quay càng nhanh.
* Sử dụng LEDC (LED Controller) peripheral của ESP32 để tạo tín hiệu PWM.

RTOS Task Management:

* Multi-tasking: camera task, WiFi task, Bluetooth task, motor task, sensor task chạy đồng thời.
* Priority scheduling: task quan trọng (motor control, emergency stop) có priority cao hơn telemetry.
* Inter-task communication: sử dụng queue, semaphore để truyền dữ liệu giữa các task an toàn.

## IV. Giao thức truyền thông

WiFi 802.11n (2.4GHz):

* CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance): cơ chế truy cập kênh truyền, tránh xung đột gói tin.
* TCP/IP stack: đảm bảo gói tin được truyền đầy đủ và đúng thứ tự.
* HTTP server: ESP32-CAM chạy web server để stream MJPEG và nhận lệnh qua HTTP GET/POST request.

Bluetooth Classic (BR/EDR):

* SPP (Serial Port Profile): mô phỏng kết nối serial qua Bluetooth để truyền lệnh điều khiển ASCII.
* Master-Slave architecture: smartphone là master, ESP32-CAM là slave.
* Pairing and bonding: ghép nối bảo mật với PIN code, lưu thông tin để tự động kết nối lần sau.

MJPEG (Motion JPEG) Streaming:

* HTTP multipart/x-mixed-replace: gửi liên tục các JPEG frame qua HTTP connection duy nhất.
* Format: --frame\r\nContent-Type: image/jpeg\r\nContent-Length: <size>\r\n\r\n<JPEG data>.
* Lợi ích: đơn giản, không cần codec phức tạp, browser có thể xem trực tiếp.

JSON (JavaScript Object Notation):

* Format trao đổi dữ liệu giữa laptop và ESP32-CAM: {"alert": "on", "person\_count": 2}.
* Nhẹ, dễ parse, human-readable, hỗ trợ tốt trên cả Python và C/C++.

## V. Cảm biến và Điều khiển

Cảm biến siêu âm HC-SR04:

* Nguyên lý: phát xung siêu âm 40kHz, đo thời gian phản hồi từ vật cản, tính khoảng cách = (thời gian × tốc độ âm thanh) / 2.
* Trigger pulse: gửi xung 10μs để kích hoạt, Echo pulse trả về thời gian proportional với khoảng cách.
* Độ chính xác: ±3mm, phạm vi 2-400cm, góc phát hiện 15 độ.

Motor Driver L298N/TB6612FNG:

* H-Bridge circuit: cho phép đảo chiều quay động cơ DC bằng cách đổi cực tính điện áp.
* Input: 2 chân điều khiển hướng (IN1, IN2), 1 chân PWM điều khiển tốc độ (ENA).
* Logic: IN1=HIGH, IN2=LOW → quay thuận; IN1=LOW, IN2=HIGH → quay nghịch; PWM → tốc độ.

Camera OV2640:

* CMOS image sensor 2MP (1600×1200 max), hỗ trợ nhiều resolution (QVGA, VGA, SVGA, UXGA).
* Interface: DVP (Digital Video Port) kết nối trực tiếp với ESP32-CAM qua 8-bit parallel bus.
* Auto-exposure, auto-white balance: tự động điều chỉnh theo điều kiện ánh sáng.

## VI. Thuật toán và Xử lý

Thuật toán tránh vật cản:

* Rule-based logic:
  + Nếu distance > 50cm → tiến với tốc độ bình thường (80% PWM)
  + Nếu 20cm < distance ≤ 50cm → giảm tốc (40% PWM)
  + Nếu → dừng, quay phải 90°, kiểm tra khoảng cách, nếu clear thì tiến
* State machine: Idle → Moving → Detecting Obstacle → Avoiding → Moving.

Bộ lọc Kalman:

* Lọc nhiễu cảm biến siêu âm: trung bình động 5-10 mẫu gần nhất để loại bỏ giá trị nhiễu đột biến.
* Ví dụ: filtered\_distance = (d1 + d2 + d3 + d4 + d5) / 5.

## VII. Cập nhật Firmware OTA

OTA (Over-The-Air) Update:

* Partition scheme: Flash 4MB chia thành 2 app partition (mỗi ~1.8MB), khi update ghi vào partition không active.
* Process: Download firmware → Validate checksum (MD5/SHA256) → Write to OTA partition → Set boot flag → Reboot.
* Rollback mechanism: nếu boot fail 3 lần liên tiếp, ESP32 tự động boot lại từ partition cũ (rollback).

HTTP Firmware Upload:

* Laptop gửi file .bin qua HTTP POST multipart/form-data.
* ESP32-CAM nhận từng chunk (~4KB), ghi vào flash, trả về progress %.
* Update complete: reboot và verify firmware version mới.

## VIII. Bảo mật

WPA2/WPA3 Encryption:

* Mã hóa kết nối WiFi với AES-128 hoặc AES-256.
* Pre-shared key (PSK): password WiFi được hash thành session key.
* Ngăn chặn sniffing và man-in-the-middle attack trên kênh WiFi.

Checksum Validation:

* MD5/SHA256 hash: xác thực firmware không bị sửa đổi trong quá trình truyền.
* ESP32-CAM tính hash của file nhận được, so sánh với hash gốc, nếu khác thì reject update.

## IX. Công nghệ phát triển ứng dụng

Python với OpenCV, YOLO:

* OpenCV (cv2): thư viện xử lý ảnh và video mã nguồn mở.
* Ultralytics YOLOv8: framework Python để load model, inference, và visualize kết quả.
* PyTorch/ONNX: backend để chạy mô hình deep learning.

Arduino Framework cho ESP32:

* C/C++ programming với Arduino API: digitalWrite(), analogWrite(), Serial.println().
* ESP-IDF (Espressif IoT Development Framework): framework chính thức của ESP32, mạnh hơn Arduino.
* PlatformIO: IDE đa nền tảng, quản lý thư viện và build automation.

Mobile App Development:

* Flutter (Dart): cross-platform framework để build app cho Android và iOS từ một codebase.
* React Native (JavaScript): alternative cho Flutter, sử dụng React syntax.
* Bluetooth plugin: flutter\_blue, react-native-ble-plx để kết nối Bluetooth.

Communication Libraries:

* Requests (Python): gửi HTTP request từ laptop đến ESP32-CAM.
* WebSocket (optional): kết nối real-time 2 chiều thay thế HTTP polling.
* MQTT (optional): publish/subscribe protocol cho IoT, nhẹ hơn HTTP.