

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HỒ CHÍ MINH

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



ĐỒ ÁN MÔN HỌC ĐIỆN Ô TÔ

**ĐỀ TÀI: XE GIÁM SÁT DÒ LINE KẾT HỢP XỬ LÝ
ẢNH NHẬN ĐIỆN KHUÔN MẶT**

GVHD: PGS.TS ĐỖ VĂN DŨNG

SVTH:

NGUYỄN KHẮC THÀNH ĐẠT

MSSV: 18145332

HUỲNH ĐỨC THÀNH

MSSV: 18145231

Tp. Hồ Chí Minh, tháng 6 năm 2021

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Tên đề tài: Xe giám sát dò line kết hợp xử lý ảnh nhận diện khuôn mặt

Họ và tên Sinh viên: Nguyễn Khắc Thành Đạt

MSSV: 18145332

Huỳnh Đức Thành

MSSV: 18145231

Ngành: Công nghệ Kỹ thuật ô tô

I. NHẬN XÉT

1. Về hình thức trình bày và tính hợp lý của cấu trúc đề tài:

.....
.....

2. Về nội dung (đánh giá chất lượng đề tài, ưu/khuyết điểm và giá trị thực tiễn)

.....
.....
.....

II. NHỮNG NỘI DUNG CẦN ĐIỀU CHỈNH, BỔ SUNG

.....
.....
.....

III. ĐỀ NGHỊ VÀ ĐÁNH GIÁ

1. Đề nghị

.....

2. Điểm đánh giá (theo thang điểm 10)

.....

Tp. Hồ Chí Minh, ngày tháng... năm 2021

Giảng viên hướng dẫn

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

Tên đề tài: Xe giám sát dò line kết hợp xử lý ảnh nhận diện khuôn mặt

Họ và tên Sinh viên: Nguyễn Khắc Thành Đạt

MSSV: 18145332

Huỳnh Đức Thành

MSSV: 18145231

Ngành: Công nghệ Kỹ thuật ô tô

I. NHẬN XÉT

1. Về hình thức trình bày và tính hợp lý của cấu trúc đề tài:

.....
.....

2. Về nội dung (đánh giá chất lượng đề tài, ưu/khuyết điểm và giá trị thực tiễn)

.....
.....
.....

II. NHỮNG NỘI DUNG CẦN ĐIỀU CHỈNH, BỔ SUNG

.....
.....
.....

III. ĐỀ NGHỊ VÀ ĐÁNH GIÁ

1. Đề nghị

.....

2. Điểm đánh giá (theo thang điểm 10)

.....

Tp. Hồ Chí Minh, ngày tháng... năm 2021

Giảng viên phản biện

XÁC NHẬN HOÀN THÀNH ĐỒ ÁN

Tên đề tài: Xe giám sát dò line kết hợp xử lý ảnh nhận diện khuôn mặt

Họ và tên Sinh viên: Nguyễn Khắc Thành Đạt

MSSV: 18145332

Huỳnh Đức Thành

MSSV: 18145231

Ngành: Công nghệ Kỹ thuật ô tô

Sau khi tiếp thu và điều chỉnh theo góp ý của Giảng viên hướng dẫn, Giảng viên phản biện và các thành viên trong Hội đồng bảo vệ. Đồ án môn học đã được hoàn chỉnh đúng theo yêu cầu về nội dung và hình thức.

Chủ tịch Hội đồng: _____

Giảng viên hướng dẫn: _____

Giảng viên phản biện: _____

Tp. Hồ Chí Minh, ngày tháng... năm 2021

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	8
1.1. Giới thiệu	8
1.2. Lý do chọn đề tài	8
1.3. Mục tiêu thực hiện đề tài	9
1.4. Phương pháp nghiên cứu	10
1.5. Phạm vi của đề tài	10
1.6. Nhiệm vụ của đề tài	10
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	11
2.1. Cảm biến vật cản hồng ngoại	11
2.2. Arduino Uno R3	12
2.3. ESP32 – CAM	16
2.4. Nhận diện khuôn mặt bằng đặc trưng Haar	17
2.5. Mạch cầu H đảo chiều động cơ	19
CHƯƠNG 3: THỰC HIỆN SẢN PHẨM	20
3.1. Khái quát về hoạt động của “Xe giám sát dò line kết hợp xử lý ảnh nhận dạng khuôn mặt”	20
3.2. Thực hiện	20
3.2.1. Các linh kiện cần thiết	20
3.2.2. Phần xe dò line	21
3.2.2.1. Thiết kế xe dò line và chọn loại động cơ	21
3.2.2.2. Sơ đồ đấu dây và nguyên lý hoạt động xe dò line	23
3.2.2.3. Lập trình xe dò line	24
3.2.3. Phần xử lý ảnh và cảnh báo người dùng	26
3.2.3.1. Sơ đồ kết nối	26
3.2.3.2. Cách sử dụng module ESP32-CAM để mở camera trên webserver	27
3.2.3.3. Lập trình nhận dạng khuôn mặt	30
3.3. Kết quả	33

DANH MỤC CÁC HÌNH

Hình 1.2: Xếp hạng độ hiệu quả của các đối tượng phát hiện trộm cắp (năm 2012) đơn vị %	9
Hình 2.1a: Nguyên lý hoạt động cảm biến vật cản hồng ngoại	11
Hình 2.1b: Sơ đồ mạch điện cảm biến vật cản hồng ngoại	11
Hình 2.2a: Board Arduino Uno R3	12
Hình 2.2b: Sơ đồ chân Arduino Uno	15
Hình 1.3: Sơ đồ chân ESP32-CAM	16
Hình 2.4a: Minh họa cho đặc trưng Haar	17
Hình 2.4b: Ảnh lý tưởng (trái) và ảnh thực tế (phải)	18
Hình 2.5: Mạch cầu H đảo chiều động cơ	19
Hình 3.2.2.1a: Bản vẽ thiết kế xe dò line	21
Hình 3.2.2.1b: Các lực tác dụng lên xe	22
Hình 3.2.2.2: Sơ đồ nối dây của xe dò line	23
Hình 3.2.3.1: Sơ đồ kết nối của các bộ phận xử lý ảnh và cảnh báo người dừng	26
Hình 3.2.3.2a: Thiết lập cho ESP32-CAM	28
Hình 3.2.3.2b: Một địa chỉ IP dẫn đến Webserver	28
Hình 3.2.3.2c: Sau khi dán link truy cập vào trình duyệt, một video stream sẽ hiện ra	29
Hình 3.2.3.3: Python tự động tạo 1 hình chữ nhật bao lấy khuôn mặt	32
Hình 3.3a: Sản phẩm sau khi hoàn thiện	33
Hình 3.3b: Hoạt động của xe dò line đi thẳng và quẹo trái	34
Hình 3.3c: Hoạt động của bộ phận xử lý ảnh, cảnh báo người dừng	34

DANH MỤC CÁC BẢNG

Bảng 2.2: Thông số kỹ thuật Arduino Uno	14
Bảng 3.2.2.1: Bảng khối lượng các linh kiện thành phần	22
Bảng 3.2.2.2: Kết nối các chân giữa L298N, Arduino, cảm biến vật cản hồng ngoại	24
Bảng 3.2.3.1a: Kết nối các chân giữa ESP32-CAM và mạch chuyển PL2303	27
Bảng 3.2.3.1b: Kết nối các chân giữa Arduino UNO và còi	27

CHƯƠNG 1: TỔNG QUAN

1.1. Giới thiệu

Ngày nay, xe tự hành là một thuật ngữ không còn xa lạ với con người. Xe tự hành được sử dụng nhiều nhất là mục đích vận chuyển người, đồ vật trong khoảng cách xác định và trong an ninh giám sát. Đặc biệt, đối với mục đích an ninh, xe tự hành giám sát được các công ty lớn đầu tư nghiên cứu rất nhiều và có những bước phát triển nổi bật, thậm chí đã có sản phẩm được đưa vào thương mại hóa rộng rãi. Ở các nước phát triển, việc chế tạo và sử dụng các xe giám sát thông minh từ lâu đã trở thành nhu cầu tất yếu của các doanh nghiệp, các hộ gia đình. Tuy nhiên ở Việt Nam, xe giám sát vẫn còn quá xa lạ đối với mọi người cho nên phần lớn các doanh nghiệp ngoài dựa vào camera được bố trí sẵn, họ vẫn còn phải thuê rất nhiều nhân viên bảo vệ dẫn đến tốn kém chi phí. Sản phẩm “Xe giám sát dò line kết hợp xử lý ảnh nhận diện khuôn mặt” của nhóm mong muốn sẽ giải quyết vấn đề giảm nhân lực bảo vệ trong các tòa nhà, công ty, thậm chí nhà ở mà vẫn đảm bảo chất lượng về an ninh.

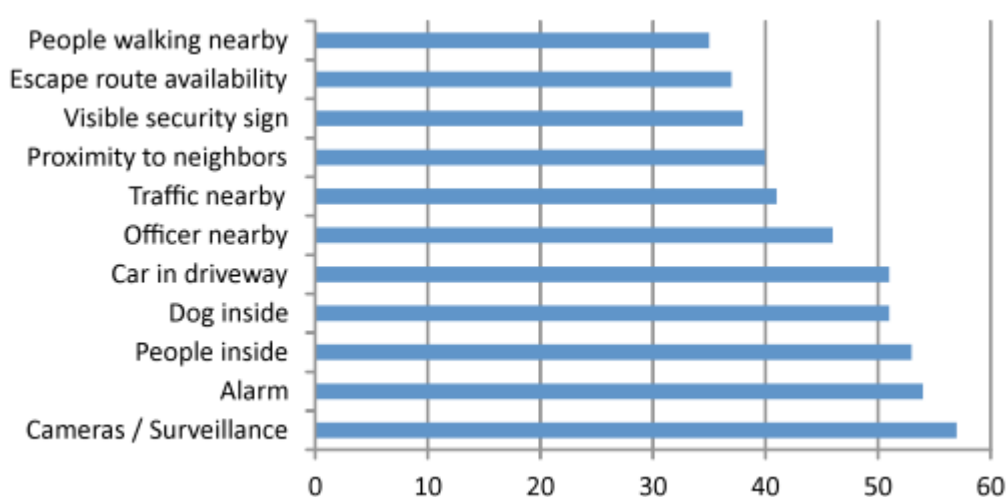
1.2. Lý do chọn đề tài

Ở Việt Nam hiện nay, các vụ trộm cắp xảy ra rất thường xuyên. Nếu chỉ xét riêng về các vụ đột nhập vào nhà dân, vào doanh nghiệp trộm cắp tài sản thì trong năm 2019, trên địa bàn tỉnh Mỹ Tho là 218 vụ^[1], địa bàn tỉnh Tiền Giang là 97 vụ^[2],... Để giải quyết vấn đề giám sát địa bàn, nơi ở, nhiều người đã lắp các camera an ninh. Tuy nhiên giải pháp này không thật sự tốt đối với các khu vực rộng lớn vì lắp quá nhiều camera sẽ gây tốn kém. Bên cạnh đó, camera nào cũng có điểm mù nên khi người đột nhập nắm được điều này thì việc giám sát sẽ không có ý nghĩa gì nữa.

Theo số liệu thống kê năm 2012 được thực hiện bởi trường Đại học North Carolina về hiệu quả trong việc phát hiện trộm cắp của các đối tượng khác nhau thì camera giám sát dẫn đầu (57%), theo sau là các phương thức giúp báo động cho người dùng (54%). Và đáng chú ý ở cuối bài khảo sát này, họ đã kết luận việc kết

hợp giữa giám sát và báo động sẽ giúp ngăn ngừa mọi rủi ro về an ninh^[1]. Vì lẽ đó mà công nghệ giám sát ở nước phát triển tiến bộ vượt bậc, thậm chí là bỏ xa các nước đang phát triển.

Qua các số liệu được phân tích ở trên, nhóm đã thấy được tính cấp thiết trong việc tạo ra một sản phẩm có khả năng giám sát, báo động cho người dùng kết hợp khả năng tự động di chuyển đến từng khu vực. Nhóm mong muốn sản phẩm sẽ là bước khởi đầu trong việc giải quyết một số vấn đề về an ninh.



Hình 1.2: Xếp hạng độ hiệu quả của các đối tượng phát hiện trộm cắp (năm 2012) đơn vị %^[3]

1.3. Mục tiêu thực hiện đề tài

- Tìm hiểu về Arduino, lập trình vi điều khiển Atmega328p.
- Tìm hiểu và cải thiện thuật toán điều khiển cho xe dò line.
- Tìm hiểu cách sử dụng board ESP32 – CAM để mở camera từ Webserver.
- Tìm hiểu cơ bản về lập trình Python trong nhận dạng khuôn mặt.
- Thiết kế sản phẩm thực tế xe dò line có gắn camera nhận diện khuôn mặt.

1.4. Phương pháp nghiên cứu

Việc nghiên cứu dựa trên các kiến thức đã được học trước đó, các tài liệu giấy và tài liệu trên mạng, giáo trình tại thư viện trường và đồ án liên quan trước và nghiên cứu thực nghiệm

1.5. Phạm vi của đề tài

- Về nội dung: Đề tài tập trung lập thiết kế, làm sản phẩm, lập trình xe dò line và áp dụng xử lý ảnh nhận diện khuôn mặt.
- Về không gian: Sản phẩm được sử dụng ở những nơi có bề mặt phẳng, kết nối Wi-Fi ổn định.
- Thời gian: Khoảng 3 tháng (03/2021 đến 06/2021).

1.6. Nhiệm vụ của đề tài

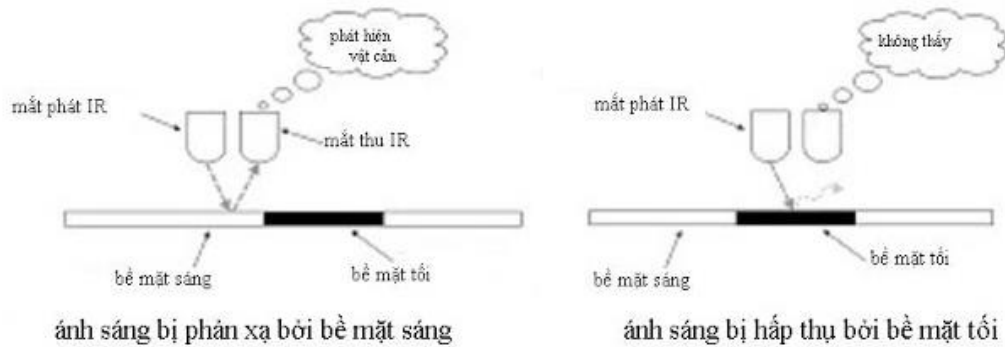
Đề tài này tập trung nghiên cứu tình hình phát triển sản phẩm xe giám sát trong, ngoài nước, ứng dụng kiến thức đã học về lập trình vi điều khiển, tìm hiểu kiến thức mới trong lập trình xử lý ảnh và làm sản phẩm có thể vận hành được ở mức cơ bản.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

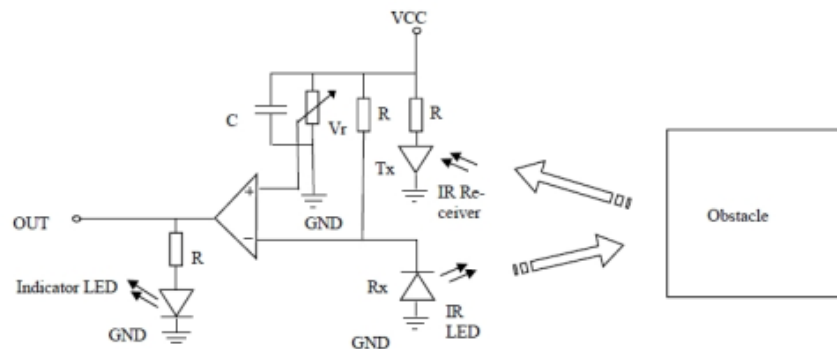
2.1. Cảm biến vật cản hồng ngoại

Cảm biến vật cản hồng ngoại có khả năng nhận biết vật cản với một cặp LED thu phát hồng ngoại để truyền và nhận dữ liệu hồng ngoại.

Nguyên lý hoạt động: Tia hồng ngoại phát ra với tần số nhất định, khi có vật cản trên đường truyền của LED phát (Rx) nó sẽ phản xạ vào LED thu (Tx) hồng ngoại, khi đó LED báo vật cản trên module sẽ sáng, khi không có vật cản, LED sẽ tắt^[9]. Như **Hình 2.1a**, nếu tia hồng ngoại từ mắt phát, đến bề mặt sáng và bị chiếu ngược lại mắt nhận, chân OUT sẽ trả về giá trị 1. Ngược lại, đối với bề mặt tối, tia hồng ngoại từ mắt phát bị hấp thụ, chân OUT sẽ trả về giá trị 0. Cảm biến gồm có 3 chân: VCC, GND, OUT như **Hình 2.1b**.



Hình 2.1a: Nguyên lý hoạt động cảm biến vật cản hồng ngoại



Hình 2.1b: Sơ đồ mạch điện cảm biến vật cản hồng ngoại^[4]

2.2. Arduino Uno R3

Arduino Uno R3 là một board mạch vi điều khiển được phát triển bởi Arduino.cc, một nền tảng điện tử mã nguồn mở chủ yếu dựa trên vi điều khiển AVR Atmega328P.

- Phiên bản hiện tại của Arduino Uno R3 đi kèm với giao diện USB, 6 chân đầu vào analog, 14 cổng kỹ thuật số I/O được sử dụng để kết nối với các mạch điện tử, thiết bị bên ngoài. Trong đó có 14 cổng I/O, 6 chân đầu ra xung *PWM* cho phép các nhà thiết kế kiểm soát và điều khiển các thiết bị mạch điện tử ngoại vi một cách trực quan.
- Arduino Uno R3 được kết nối trực tiếp với máy tính thông qua USB để giao tiếp với phần mềm lập trình IDE, tương thích với Windows, MAC hoặc Linux Systems, tuy nhiên, Windows thích hợp hơn để sử dụng. Các ngôn ngữ lập trình như C và C++ được sử dụng trong IDE.
- Ngoài USB, người dùng có thể dùng nguồn điện ngoài để cấp nguồn cho bo mạch.
- Các bo mạch Arduino Uno khá giống với các bo mạch khác trong các loại Arduino về mặt sử dụng và chức năng, tuy nhiên các bo mạch Uno không đi kèm với chip điều khiển FTDI USB to Serial.
- Có rất nhiều phiên bản bo mạch Uno, tuy nhiên, Arduino Nano V3 và Arduino Uno là những phiên bản chính thức nhất đi kèm với vi điều khiển Atmega328 8 bit AVR Atmel trong đó bộ nhớ RAM là 32KB.
- Khi tính chất và chức năng của nhiệm vụ trở nên phức tạp, thẻ nhớ SD Micro có thể được kết nối thêm vào Arduino để lưu trữ được nhiều thông tin hơn.



Hình 2.2a: Board Arduino Uno R3

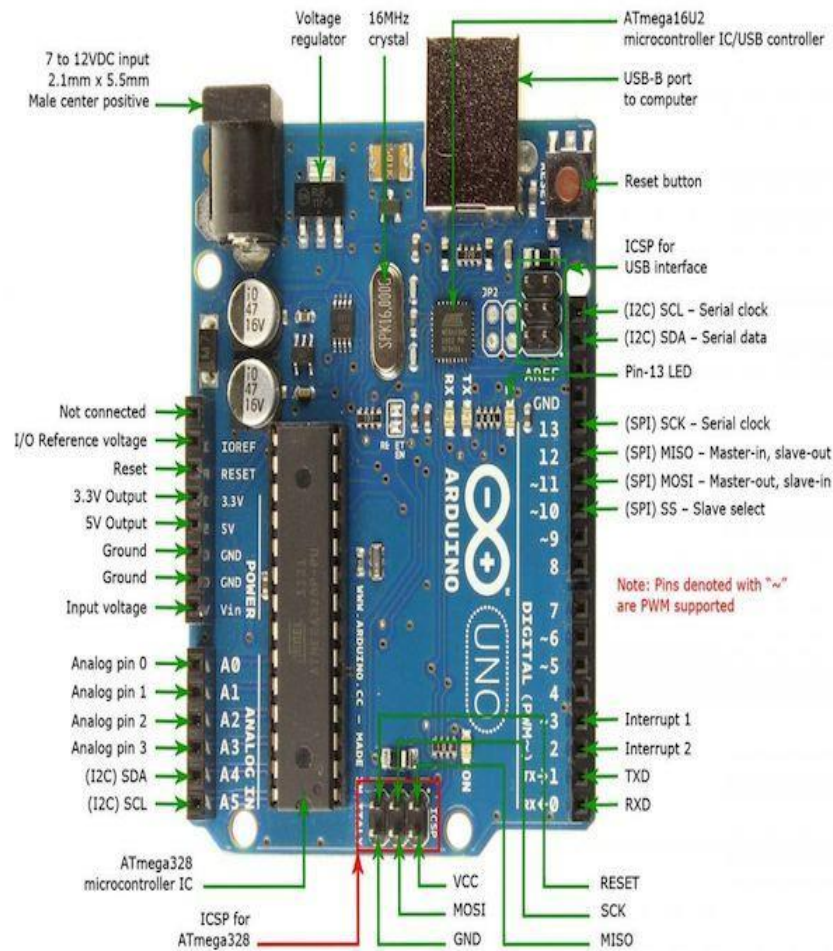
- Arduino Uno đi kèm với giao diện USB tức là cổng USB được thêm vào bo mạch Arduino để phát triển giao tiếp nối tiếp với máy tính.
- Bộ vi điều khiển Atmega328 sử dụng trên bo mạch đi kèm với một số tính năng như hẹn giờ, bộ đếm, ngắt, chân PWM, CPU, chân I/O và dựa trên xung nhịp 16 MHz giúp tạo ra nhiều tần số và số lệnh hơn trong mỗi chu kỳ.
- Đây là một nền tảng mã nguồn mở, nơi mọi người có thể sửa đổi và tối ưu hóa bảng dựa trên số lượng hướng dẫn và nhiệm vụ muốn đạt được.
- Arduino đi kèm với một tính năng điều chỉnh tích hợp giúp giữ điện áp trong tầm kiểm soát khi thiết bị được kết nối với thiết bị bên ngoài.
- Chân reset trên Arduino để thiết lập lại toàn bộ và đưa chương trình đang chạy trở về ban đầu. Chân reset này hữu ích khi Arduino bị treo khi đang chạy chương trình
- Có 14 chân I/O digital và 6 chân analog được tích hợp trên Arduino cho phép kết nối bên ngoài với bất kỳ mạch nào với Arduino. Các chân này cung cấp sự linh hoạt và dễ sử dụng cho các thiết bị bên ngoài có thể được kết nối thông qua các chân này.
- 6 chân analog được đánh dấu là A0 đến A5 và có độ phân giải 10 bit. Các chân này đo từ 0 đến 5V, tuy nhiên, chúng có thể được cấu hình ở phạm vi cao bằng cách sử dụng chức năng analogReference () và chân ISF.
- Bộ nhớ flash 13KB được sử dụng để lưu trữ số lượng hướng dẫn dưới dạng mã.
- Chỉ cần nguồn 5V để sử dụng với Arduino, hoặc lấy nguồn trực tiếp từ cổng USB. Arduino có thể hỗ trợ nguồn điện bên ngoài lên đến 12 V có thể được điều chỉnh và giới hạn ở mức 5 V hoặc 3,3 V dựa trên yêu cầu của projects.

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

Bảng 2.2: Thông số kỹ thuật Arduino Uno^[5]

- Vin: Đây là điện áp đầu vào được cung cấp cho board mạch Arduino. Khác với 5V được cung cấp qua cổng USB. Pin này được sử dụng để cung cấp điện áp toàn mạch thông qua jack nguồn, thông thường khoảng 7-12VDC
- 5V: Chân 5V được sử dụng để cung cấp điện áp đầu ra. Arduino được cấp nguồn bằng ba cách đó là USB, chân Vin của bo mạch hoặc giắc nguồn DC.
- USB: Hỗ trợ điện áp khoảng 5V trong khi Vin và Power Jack hỗ trợ dải điện áp trong khoảng từ 7V đến 20V.
- GND: Chân mass chung cho toàn mạch Arduino
- Reset: Chân reset để thiết lập lại về ban đầu
- IOREF: Chân này rất hữu ích để cung cấp tham chiếu điện áp cho Arduino
- PWM: PWM được cung cấp bởi các chân 3,5,6,9,10, 11. Các chân này được cấu hình để cung cấp PWM đầu ra 8 bit.
- SPI: Chân này được gọi là giao diện ngoại vi nối tiếp. Các chân 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) cung cấp liên lạc SPI với sự trợ giúp của thư viện SPI.
- AREF: Chân này được gọi là tham chiếu tương tự, được sử dụng để cung cấp điện áp tham chiếu cho các đầu vào tương tự.

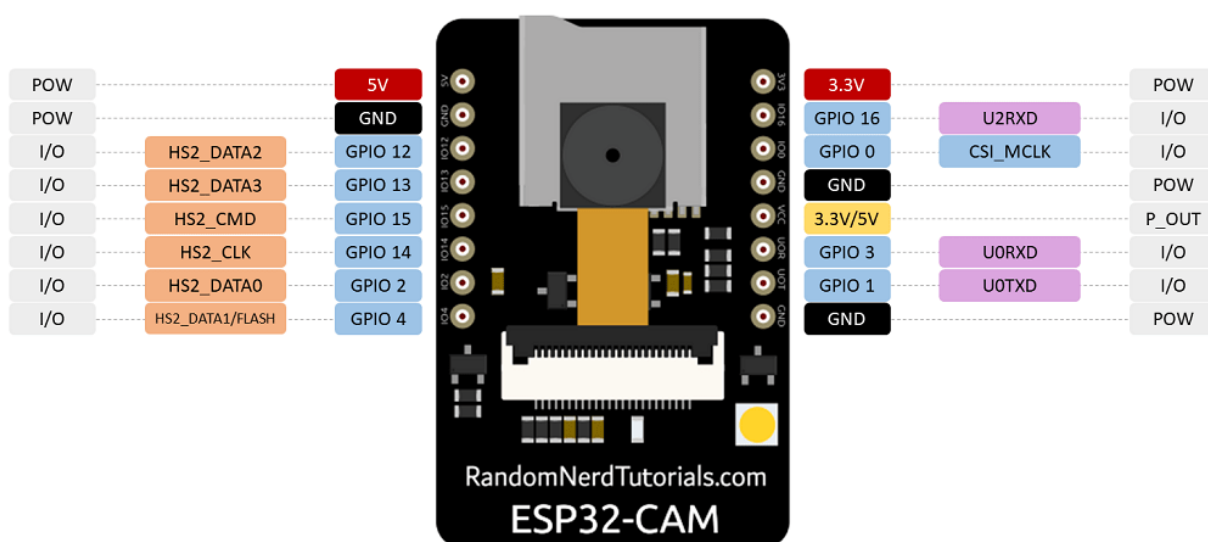
- TWI: Chân Giao tiếp TWI được truy cập thông qua thư viện dây. Chân A4 và A5 được sử dụng cho mục đích này.
- Serial Communication: Giao tiếp nối tiếp được thực hiện thông qua hai chân 0 (Rx) và 1 (Tx).
- Rx: Chân này được sử dụng để nhận dữ liệu trong khi chân Tx được sử dụng để truyền dữ liệu.
- External Interrupts (Ngắt ngoài): Chân 2 và 3 được sử dụng để cung cấp các ngắt ngoài.



Hình 2.2b: Sơ đồ chân Arduino Uno

2.3. ESP32 – CAM

ESP32-CAM là một board mạch phát triển với chip điều khiển ESP32-S, camera OV2640 (có thể mở video stream kích thước 1284x1024 điểm ảnh, tốc độ khung hình là 6 fps, tốc độ nhanh hơn ở kích thước nhỏ hơn), khe cắm thẻ nhớ microSD và 9 GPIO để kết nối với thiết bị ngoại vi.



Hình 2.3: Sơ đồ chân ESP32-CAM^[6]

Vấn đề cấp nguồn:

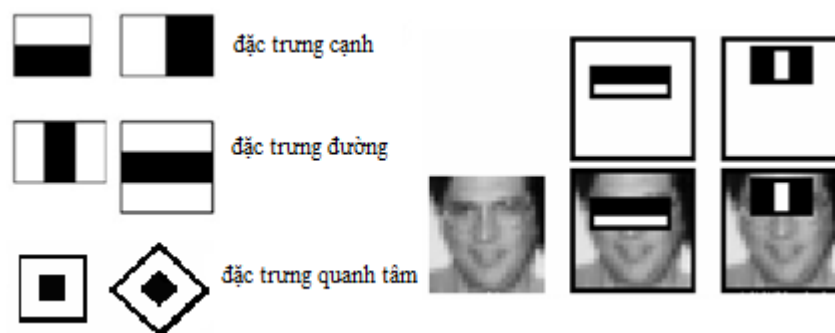
- Chân nguồn (power pin): ESP32-CAM có 3 chân GND (màu đen trong sơ đồ chân), 1 chân 3.3V và 1 chân 5V (màu đỏ trong sơ đồ chân).
- ESP32-CAM có thể hoạt động ở cả 3.3V và 5V, tuy nhiên với nguồn 3.3V, ESP32-CAM có thể sẽ hoạt động không ổn định, chúng ta nên dùng nguồn 5V để đảm bảo ESP32-CAM hoạt động ổn định.
- Trên **Hình 2.3**, có thể thấy 1 chân màu vàng 3.3/5V, trên ESP32-CAM chân này được kí hiệu là VCC. Chúng ta không nên sử dụng chân này để cấp nguồn cho ESP32-CAM. Để cấp nguồn cho board ESP32-CAM, ta nên dùng một nguồn 5V ổn định (sạc dự phòng điện thoại chẳng hạn) kết hợp với 1 mạch chuyển trung gian serial.

Chân serial:

- GPIO 1 và GPIO 3 là 2 chân serial của ESP32-CAM (GPIO 1 đóng vai trò là TX, GPIO 3 là RX). ESP32-CAM không được tích hợp sẵn mạch giao tiếp serial với máy tính, cần kết nối 2 chân này với một mạch giao tiếp serial trung gian mới có thể nạp chương trình từ máy tính. Trong bài này là mạch PL2303.
- Sử dụng GPIO 1 và GPIO 3 để kết nối với các module ngoại vi sau khi nạp chương trình xong.
- GPIO 0 được nối với một điện trở kéo lên, có chức năng chọn chế độ hoạt động cho ESP32-CAM. Khi điện áp GPIO 0 ở mức 0, ESP32-CAM sẽ hoạt động ở chế độ flash, chúng ta nạp chương trình cho ESP32-CAM ở chế độ này. Khi điện áp chân này ở mức 1, ESP32-CAM sẽ thoát khỏi chế độ flash. Khi cần nạp chương trình cho ESP32, ta kết nối GPIO 0 với chân GND, và ngắt kết nối giữa GPIO 0 với GND khi cần ESP32-CAM hoạt động bình thường. Mỗi lần kết nối hoặc ngắt kết nối giữa GPIO 0 và GND cần nhấn nút RST trên board để ESP32-CAM cập nhật lại chế độ hoạt động.

2.4. Nhận diện khuôn mặt bằng đặc trưng Haar

Đặc trưng Haar nhận biết khuôn mặt thông qua các điểm ảnh trắng và đen trong một hình chữ nhật. Có tất cả 3 đặc trưng Haar: Đặc trưng cạnh, đặc trưng đường, đặc trưng quanh tâm.



Hình 2.4a: Minh họa cho đặc trưng Haar^[7]

Dựa vào 3 đặc trưng Haar, trong **Hình 2.4a**, đối với việc nhận diện vùng chân mày – mắt, ta dùng đặc trưng cạnh với các điểm ảnh đen tượng trưng cho chân

mày và điểm ảnh trắng tượng trưng cho phần dưới chân mày. Hoặc nhận dạng phần mũi và hai bên hốc mắt, ta dùng đặc trưng đường, với hai bên hốc mắt ứng với các điểm ảnh đen và mũi ứng với điểm ảnh trắng.

Tuy nhiên trong thực tế, không bao giờ có những bức ảnh trắng tuyệt đối (mức 0) hoặc đen tuyệt đối (mức 1), từ đó phương pháp nhận diện Violas-Jones được xây dựng để xác định các đặc tính Haar trên một bức ảnh xám thực tế.



Hình 2.4b: Ảnh lý tưởng (trái) và ảnh thực tế (phải)

Giả sử có một bức ảnh xám thực tế (phải), công thức Viola-Jones sẽ so sánh độ sáng giữa những điểm ảnh tối và điểm ảnh sáng của bức ảnh thực tế đó với bức ảnh lý tưởng (trái) qua công thức:

$$\Delta = \frac{1}{n} \sum_1^n (\text{độ sáng các điểm tối}) - \frac{1}{m} \sum_1^m (\text{độ sáng các điểm sáng}) \quad (1)$$

$$= \frac{1}{9} (0.3 + 0.6 \times 3 + 0.8 \times 4 + 0.9) - \frac{1}{7} (0.1 \times 3 + 0.2 \times 4) = 0.53$$

Trong đó

- Δ : Độ lệch màu của các điểm ảnh tối và điểm ảnh sáng.
- n, m : Lần lượt là số lượng điểm ảnh có độ sáng gần đen (trên 0.3) và gần trắng (không quá 0.3) trong bức ảnh.

Δ càng gần bằng 1 thì đặc trưng Haar càng rõ ràng, nhận diện các bộ phận trên khuôn mặt theo đặc trưng Haar càng dễ dàng. Ưu điểm của cách nhận dạng khuôn mặt này là dễ hiểu, dễ tiếp cận cho người mới học. Tuy nhiên trong thực tế, rất nhiều vật thể có thể mang những đặc trưng Haar, nên chỉ dựa vào đặc trưng

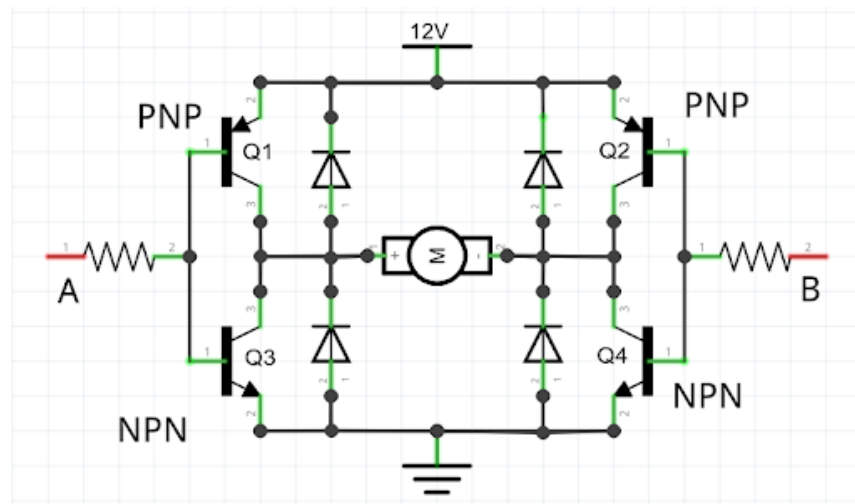
Haar thì việc nhận dạng khuôn mặt sẽ không tránh khỏi sai sót, chương trình sẽ dễ bị đánh lừa. Vì những khuyết điểm của phương pháp này, ngày nay người ta sử dụng trí tuệ nhân tạo, deep learning để cải thiện việc nhận dạng khuôn mặt. Nhưng trong bài báo cáo này, nhóm vẫn áp dụng phương pháp Haar vì tính dễ dàng, có sẵn của nó.

2.5. Mạch cầu H đảo chiều động cơ

Nguyên lý đảo chiều động cơ được mô tả dựa vào **Hình 2.5** (với Q1,Q2 là các transistor PNP và Q3,Q4 là các transistor NPN):

Khi điểm A có điện áp mức thấp, B có điện áp mức cao thì Q1, Q4 đóng và Q2,Q3 ngắt, dòng điện đi từ nguồn dương 12V → tiếp điểm EC Q1 → Động cơ → tiếp điểm CE Q4 → mass. Động cơ quay theo một chiều.

Khi điểm A có điện áp mức cao, B có điện áp mức thấp thì Q2, Q3 đóng và Q1,Q4 ngắt, dòng điện đi từ nguồn dương 12V → tiếp điểm EC Q2 → Động cơ → tiếp điểm CE Q3 → mass. Động cơ quay theo chiều ngược lại.



Hình 2.5: Mạch cầu H đảo chiều động cơ

CHƯƠNG 3: THỰC HIỆN SẢN PHẨM

3.1. Khái quát về hoạt động của “Xe giám sát dò line kết hợp xử lý ảnh nhận dạng khuôn mặt”

Sản phẩm “Xe giám sát dò line kết hợp xử lý ảnh nhận dạng khuôn mặt” được chia làm 2 phần chính:

- Phần xe dò line
- Phần bộ phận xử lý ảnh và cảnh báo người dừng.

Khi xe giám sát vận hành, nó sẽ đi có khả năng tự đi đúng vạch đường đã chỉ định. Đồng thời, 1 camera sẽ liên tục gửi hình ảnh về cho máy tính bằng Wi-Fi để máy tính xử lý. Cứ mỗi lần nhận thấy có khuôn mặt người trong bức ảnh, một chiếc còi báo động ở máy tính sẽ kêu lên. Kết quả là người sử dụng biết được có người xuất hiện ở khu vực của xe giám sát.

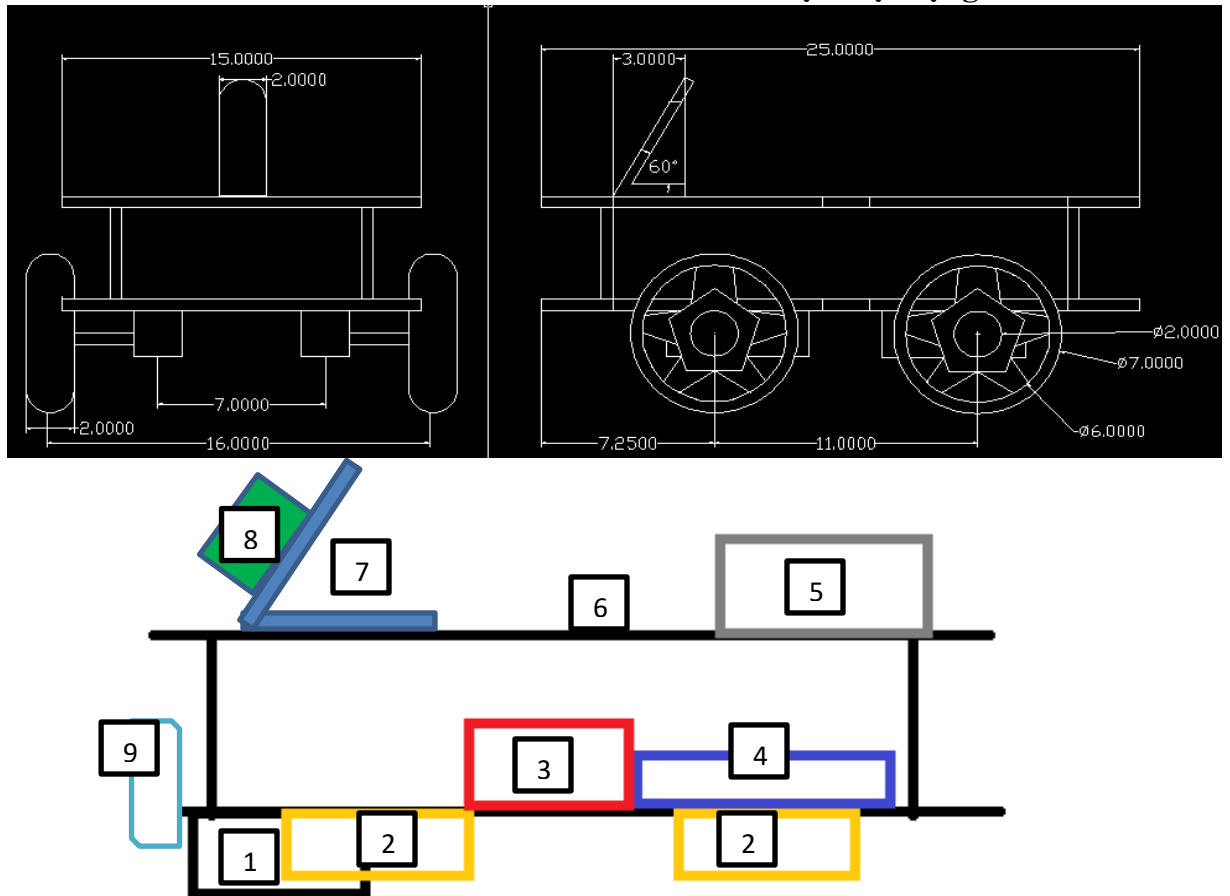
3.2. Thực hiện

3.2.1. Các linh kiện cần thiết

- 2 pin Lithium 18650 4.2V 4200mAh.
- 2 Arduino UNO.
- 1 board ESP32 – CAM kèm mạch chuyển USB UART PL2303 .
- 3 cảm biến vật cản hồng ngoại.
- 1 công tắc.
- Còi 5V.
- 4 motor giảm tốc vàng.
- Dây điện.
- 1 sạc dự phòng 5V – 2A.
- Khung xe, 1 thanh chữ L.
- 1 dây cáp nạp USB cho Arduino.

3.2.2. Phần xe dò line

3.2.2.1. Thiết kế xe dò line và chọn loại động cơ



Hình 3.2.2.1a: Bản vẽ thiết kế xe dò line

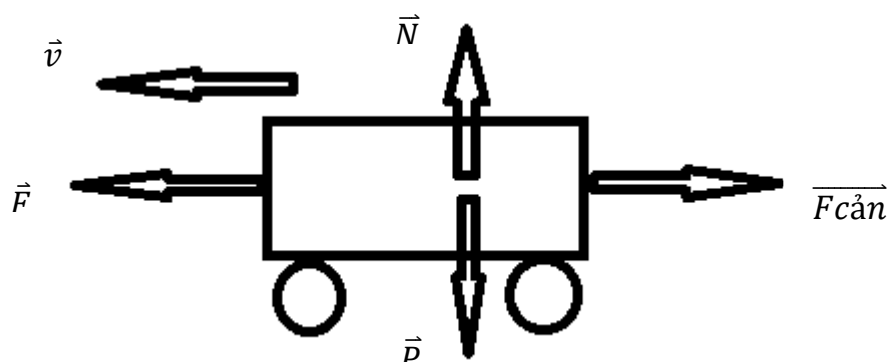
(1- Khay và pin 18650; 2 - Motor; 3 - L298N; 4 - Arduino Uno; 5 – Pin sạc dự phòng; 6 – Khung xe; 7 – Thanh chữ L; 8 – ESP32-CAM; 9 – Cảm biến hồng ngoại)

Tính toán chọn động cơ cho mô hình:

Tên linh kiện	Khối lượng (đơn vị gam)
Arduino Uno	25
L298N	48
ESP32 - CAM	20
Thanh chữ L	32
Khung xe (2 miếng)	540
Khay pin + 2 pin 18650	96
Pin sạc dự phòng	422g
4 motor (không tính bánh xe)	$0.8 \times 4 = 3.2$
Cảm biến hồng ngoại	21g

Bảng 3.2.2.1: Bảng khối lượng các linh kiện thành phần

Những yêu cầu chung: Xe chạy thẳng đều trên mặt phẳng nằm ngang, bề mặt làm từ thép và bánh xe làm bằng cao su nên hệ số cản lăn $f = 0.3$. Xe thường đi ở vận tốc $v = 35 \text{ cm/s}$ và không leo dốc.



Hình 3.2.2.1b: Các lực tác dụng lên xe

Khối lượng toàn bộ:

$$\begin{aligned}
 m &= m_{\text{ARDUINO}} + m_{\text{L298N}} + m_{\text{ESP}} + m_{\text{THANH L}} + m_{\text{KHUNG}} + m_{\text{KHAY}} + m_{\text{PIN DỰ PHÒNG}} + m_{\text{4MOTOR}} + m_{\text{CB}} \\
 &= 25 + 20 + 48 + 32 + 540 + 96 + 422 + 4 \times 0.8 + 21 = 1207 \text{ (g)} = 1.2 \text{ (kg)}
 \end{aligned}
 \tag{2}$$

Vì ta muốn tìm ra công suất tối thiểu để mô hình thẳng được các thành phần lực cản ($P_{\text{KÉO}} \geq P_{\text{CẢN}}$), ta dùng phương trình cân bằng công suất:

$$P_k = P_f + P_j + P_w + P_i \tag{3}$$

$$P_k = P_f = fNv = fmgv$$

$$P_k = 0.3 \times 1.2 \times 9.81 \times 0.35 = 1.2 \text{ (W)}$$

Với:

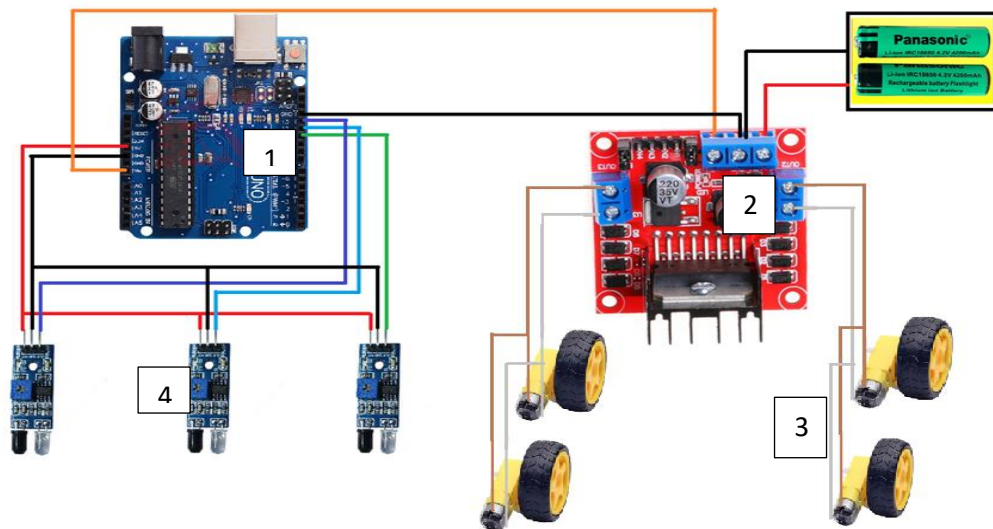
- P_k : Công suất cần thiết của động cơ để kéo cả mô hình (W).
- P_f : Công suất tiêu hao để thắng lực cản lăn (W).
- P_j : Công suất tiêu hao để thắng lực quán tính (W). Vì xe chuyển động đều nên $P_j=0$.
- P_w : Công suất tiêu hao để thắng lực cản gió (W). Vì xe chạy rất chậm, diện tích cản chính diện rất nhỏ nên bỏ qua ảnh hưởng lực này $P_w=0$.
- P_i : Công suất tiêu hao để thắng lực cản dốc (W). Vì xe chạy trên đường bằng nên $P_i=0$.

Động cơ được chọn có moment xoắn trục ra $0.5 \text{ kG.cm} = 0.05 \text{ Nm}$, tốc độ 200 vòng/phút ở điện áp 8V, nên công suất đầu ra của 1 động cơ là:

$$P = M\omega = 0.05 \times \frac{2 \times 3.14 \times 200}{60} = 1 \text{ (W)} \quad (4)$$

Khi gắn vào mô hình, công suất đầu ra của motor còn bị ảnh hưởng do pin dẫn đến khoảng số liệu này thực tế sẽ thay đổi. Tuy nhiên nhìn chung, ở điều kiện pin đầy thì việc chọn 4 động cơ sẽ đảm bảo tổng công suất lớn hơn P_k , việc kéo mô hình là khả thi.

3.2.2.2. Sơ đồ đấu dây và nguyên lý hoạt động xe dò line



Hình 3.2.2.2: Sơ đồ nối dây của xe dò line

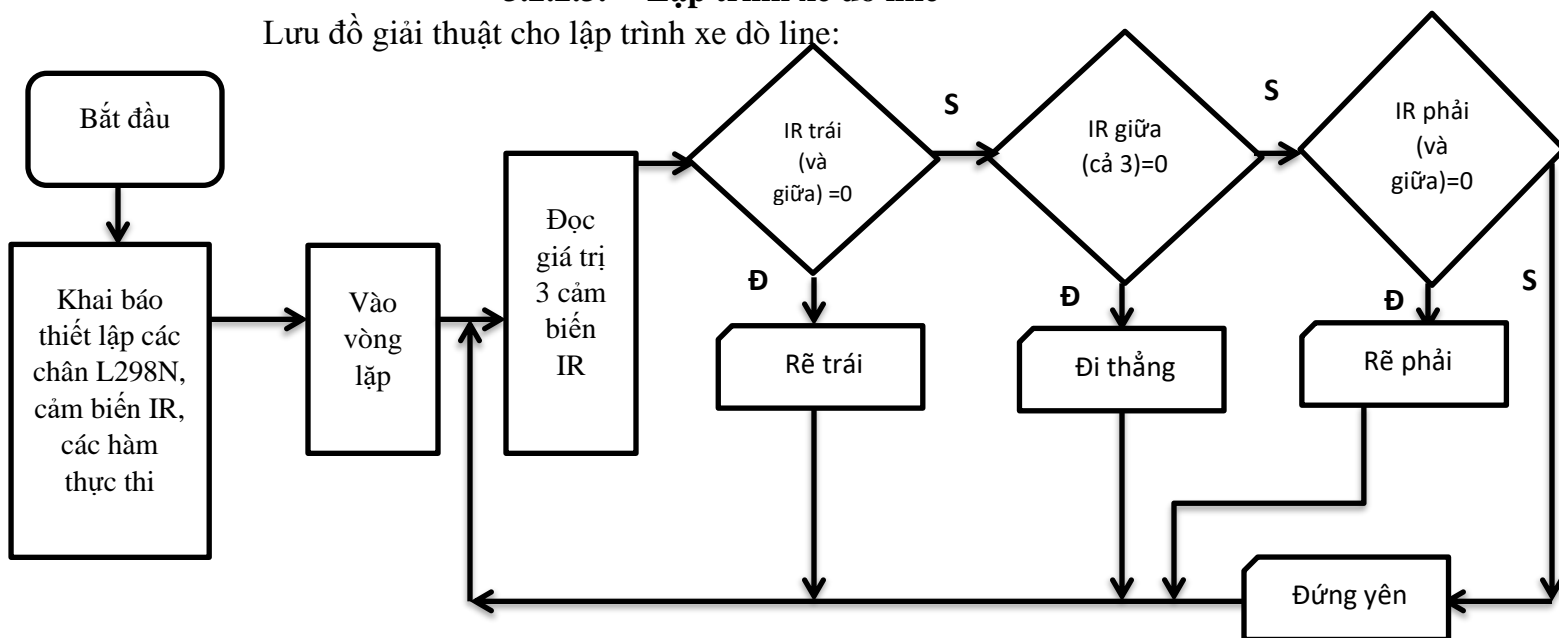
(1-Arduino Uno; 2 - L298N; 3 - Bánh xe; 4 – Cảm biến vật cản hồng ngoại)

Arduino	Chân tín hiệu cảm biến	Module L298N
Chân 2		in3
Chân 3		enA
Chân 4		in4
Chân 5		enB
Chân 6		in2
Chân 7		in1
Chân 11	IR1	
Chân 12	IR2	
Chân 13	IR3	

Bảng 3.2.2.2: Kết nối các chân giữa L298N, Arduino, cảm biến vật cản hồng ngoại

3.2.2.3. Lập trình xe dò line

Lưu đồ giải thuật cho lập trình xe dò line:




```

int IR1=11;
int IR2=12;
int IR3=13;
int enA=3;
int in1=7;
int in2=6;
int in3=2;
int in4=4;
int enB=5;

void setup() {
  pinMode(IR1, INPUT);
  pinMode(IR2, INPUT);
  pinMode(IR3, INPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // doc gia tri cam bien hong ngoai
  int val1=digitalRead(IR1);
  int val2=digitalRead(IR2);
  int val3=digitalRead(IR3);

  //0 co lane, 1 ko co lane
  if(val1==0 && val2==0 && val3==1){queotrai();}

  else if(val1==0 && val2==1 && val3==1){queotrai();}

  else if(val1==0 && val2==1 && val3==0){dithang();}

  else if(val1==0 && val2==0 && val3==0){dithang();}

  else if(val1==1 && val2==0 && val3==0){queophai();}

  else if(val1==1 && val2==1 && val3==0){queophai();}

  else {dungyen();}
}

void dithang(){
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  analogWrite(enA,110);
  analogWrite(enB,110);
}

void queotrai(){
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  analogWrite(enA,110);
  analogWrite(enB,110);
}

void queophai(){
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  analogWrite(enA,150);
  analogWrite(enB,0);
}

void dungyen(){
  analogWrite(enA,0);
  analogWrite(enB,0);
  digitalWrite(in1,LOW);
  digitalWrite(in2,LOW);
  digitalWrite(in3,LOW);
  digitalWrite(in4,LOW);
}

```

Lập trình cho xe dò line tương đối đơn giản. Phần lập trình này được thực hiện trên phần mềm Arduino IDE 1.8.13. Phần khởi tạo (trên void setup) lần lượt khai báo các chân in1->4, enA, enB của L298N và chân OUT1->3 của các cảm biến vật cản hồng ngoại. Trong void setup, lần lượt thiết lập các chân in1 đến in4, enA, enB là OUTPUT để điều khiển, OUT1 đến OUT3 là INPUT để đọc tín hiệu. Trong vòng lặp void loop, bước đầu là lần lượt đọc các giá trị cảm biến vật cản hồng ngoại (0 là có line, 1 là chệch line). Tiếp theo là điều khiển tốc độ bánh xe, các trường hợp điều khiển đã được trình bày ở **Bảng 3.2.1.1b**.

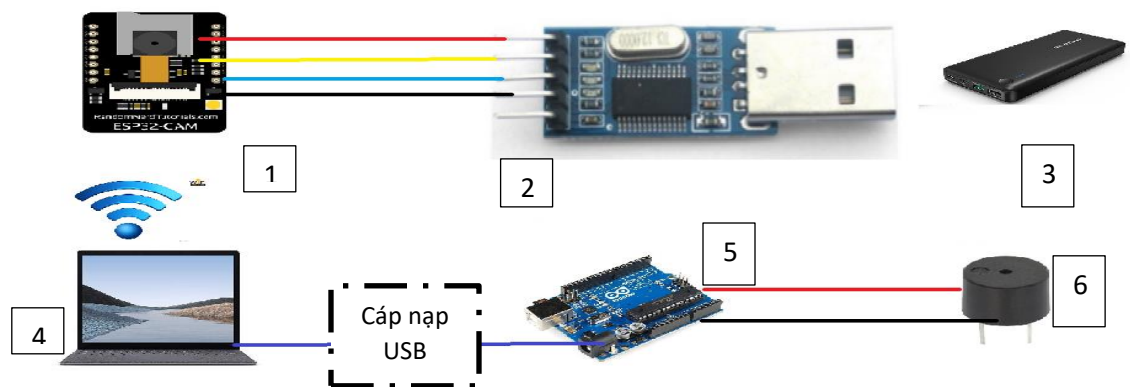
Cuối cùng là định nghĩa các hàm `dithang()`, `queotrai()`, `queophai()`, `dungyen()` ngoài void loop. Ở đây nhóm cho tốc độ di chuyển với PWM rất thấp (110) để phù hợp với tốc độ ghi hình của camera. Chênh lệch tốc độ các bánh xe 2 bên khi quay vòng lớn để xe thực hiện quay vòng nhanh hơn.

3.2.3. Phần xử lý ảnh và cảnh báo người dùng

3.2.3.1. Sơ đồ kết nối

ESP32-CAM (1) là một module có tích hợp camera (tốc độ 6 khung hình trên giây), có bộ kit RF thu phát Wi-Fi. Để cấp nguồn cho board này, ta cần 1 mạch chuyển USB UART (2) và cấp cho nguồn 5V ổn định (3), ở đây sử dụng sạc dự phòng điện thoại 5V-2A. ESP32-CAM khi được nạp code, kích hoạt sẽ gửi hình ảnh lên Webserver. Ta có thể truy cập Webserver bằng điện thoại hoặc máy tính. ESP32-CAM sẽ được gắn trên thân xe dò line.

Máy tính (4) sẽ có vai trò nhận hình ảnh gửi về và xử lý hình ảnh nhận dạng khuôn mặt (bằng OpenCV trên Python) và điều khiển Arduino (5) cấp điện cho còi (6) kêu khi phát hiện khuôn mặt (lập trình Arduino bằng Python qua thư viện Firmata). Giao tiếp giữa Python và Arduino là giao tiếp Serial, nên ta sử dụng 1 dây cáp nạp USB.



Hình 3.2.3.1: Sơ đồ kết nối của các bộ phận xử lý ảnh và cảnh báo người dùng

(1-ESP32 CAM; 2-Mạch chuyển USB UART PL2303; 3-Sạc dự phòng; 4-Máy tính; 5-Arduino Uno; 6-Còi)

ESP32-CAM	Mạch chuyển PL2303
5V	5V
UOR	TX
UOT	RX
GND	GND

Bảng 3.2.3.1a: Kết nối các chân giữa ESP32-CAM và mạch chuyển PL2303

Arduino UNO	Còi
11	(+)
GND	(-)

Bảng 3.2.3.1b: Kết nối các chân giữa Arduino UNO và còi

3.2.3.2. Cách sử dụng module ESP32-CAM để mở camera trên webserver

Để sử dụng được board ESP32-CAM, trước tiên ta cần vào Tools→Board Manager→ Tìm: esp32 → Tải về. Sau đó thiết lập các mục như **Hình 2.2.2a**. Kết nối board ESP32 cùng với mạch chuyển PL2303. ESP32 hoạt động với baud rate 115200 (115200 Bd) nhanh hơn rất nhiều so với mức 9600 Bd, nên khi chọn tốc độ 9600 Bd như thường lệ, tín hiệu đọc được từ máy tính sẽ bị sai dẫn đến không thấy đường link truy cập vào webserver. Để kích hoạt camera trên board, ta chọn File → Examples → Webserver. Sau khi chỉnh sửa tên và mật khẩu Wi-Fi ở 2 biến ssid và password như **Hình 3.2.2.2a**, ta upload cho board. Lưu ý trong suốt quá trình nạp code cho ESP32-CAM, ta phải nối tắt GPIO 0 với GND.

```

// #define CAMERA_MODEL_M5STACK_FSKAM
// #define CAMERA_MODEL_M5STACK_WIDE
// #define CAMERA_MODEL_AI_THINKER

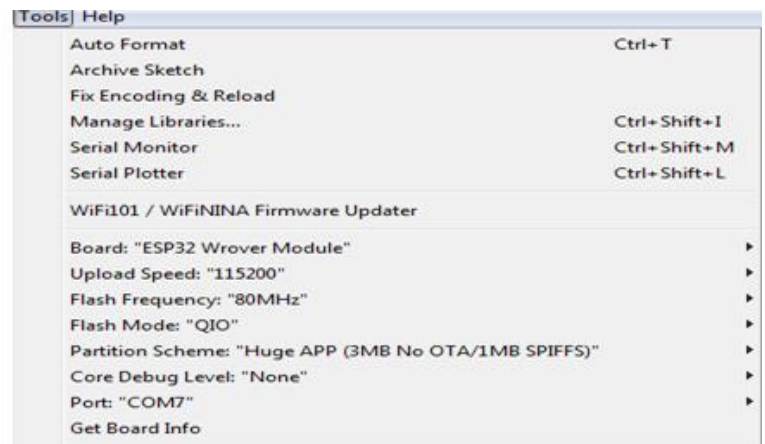
#include "camera_pins.h"

const char* ssid = "Dat";
const char* password = "0977167456";

void startCameraServer();

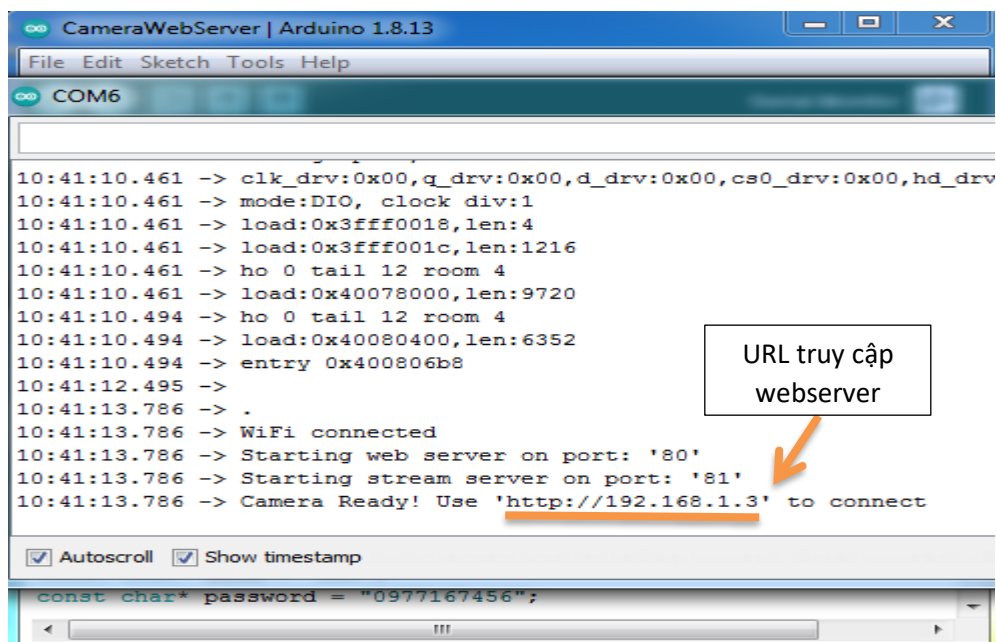
void setup() {
    Serial.begin(115200);
}

```

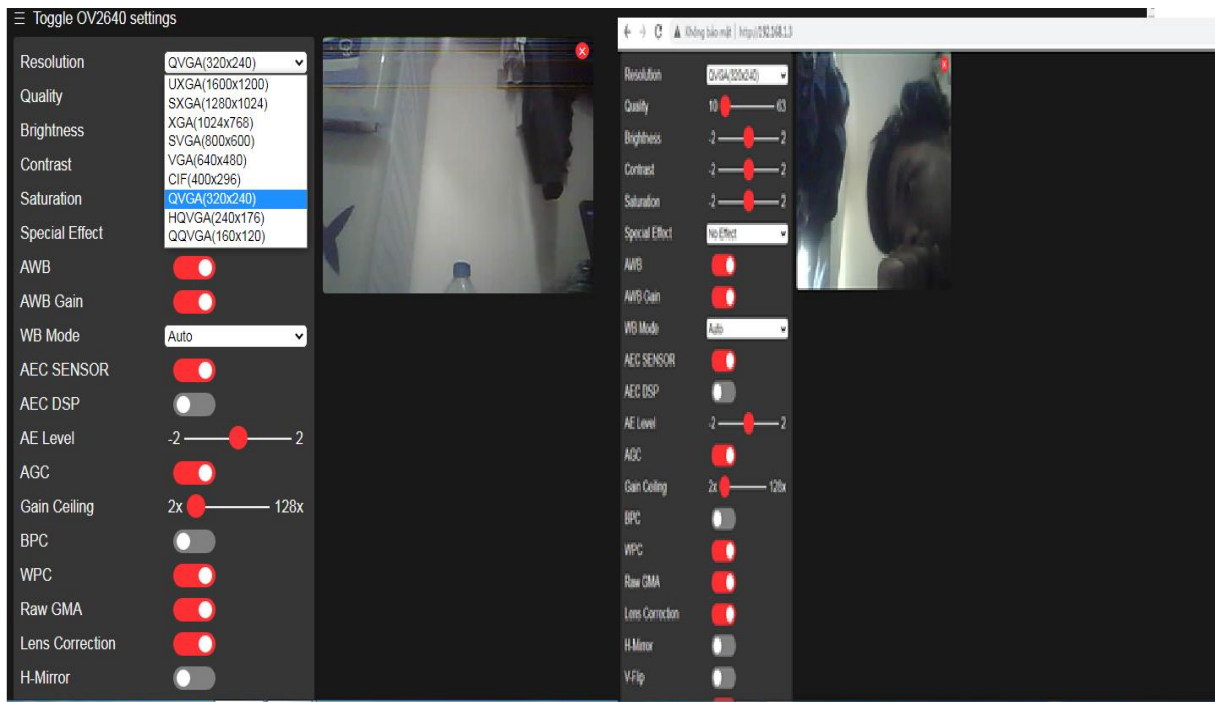


Hình 3.2.3.2a: Thiết lập cho ESP32-CAM

Sau khi nạp code thành công, ta rút chân GPIO 0 ra khỏi GND, nhấn nút reset trên board ESP32-CAM để kích hoạt camera. Một đường dẫn sẽ hiện ra, ta copy đường link đó dán vào trình duyệt web để vào webserver. Cuối cùng, một video stream sẽ hiện ra.



Hình 3.2.3.2b: Một địa chỉ IP dẫn đến Webserver



Hình 3.2.3.2c: Sau khi dán link truy cập vào trình duyệt, một video stream sẽ hiện ra

Tuy nhiên đối với video stream này, chúng ta không thể làm gì đối với nó vì vậy chúng ta sử dụng Python như một công cụ nhận dạng khuôn mặt.

3.2.3.3. Lập trình nhận dạng khuôn mặt

```
1  import cv2
2  import urllib.request
3  import pyfirmata #thư viện giao tiếp giữa arduino và python
4  import time
5
6  face_cascade=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
7
8  #thiết lập giao tiếp qua cổng COM[tự chọn]
9  board = pyfirmata.Arduino('COM3')
10
11 #đan duong link den video stream
12 #đường dẫn này không cố định nên phải
13 #thay đổi mỗi lần reset camera
14 url='http://192.168.1.3/capture?_cb=1621482935943'
15
16 #tao 1 cua so de load hinh anh vao ben trong no
17 #window_autosize la đối số mặc định của hàm
18 #cho phép thay đổi kích thước cửa sổ tự động theo kích thước ảnh
19 cv2.namedWindow("gotcha",cv2.WINDOW_AUTOSIZE)
20
21 while True:
22     #khi bắt đầu, cho còi (chân 13 arduino) OFF
23     board.digital[11].write(0)
24
25     #để lấy hình ảnh từ đường dẫn, dùng urllib
26     #tham số là biến lưu url (chứa đường link)
27     imgResponse=urllib.request.urlopen(url)
28
29     #chuyển hình ảnh về môi trường màu xám (trắng đen)
30     gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

```

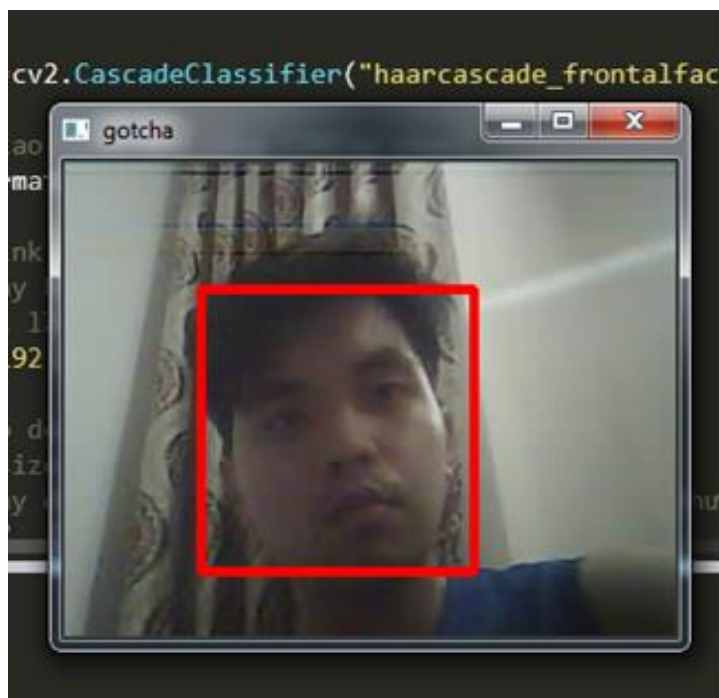
29 #chuyển hình ảnh về môi trường màu xám (trắng đen)
30 gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
31
32 #gọi hàm cho phép phát hiện 1 lúc nhiều khuôn mặt
33 face=face_cascade.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5)
34
35 #nếu phát hiện 1 khuôn mặt thì vẽ 1 hình chữ nhật quanh mặt
36 for x,y,w,h in face:
37     img=cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),3)
38     print("Human detected")
39     #nếu phát hiện khuôn mặt, còi ON
40     board.digital[11].write(1)
41     time.sleep(2)
42 #hiện video trên màn hình window
43 cv2.imshow("gotcha",img)
44 key=cv2.waitKey(1)
45
46 #nhấn phím 'q' để tắt cửa sổ
47 if key==ord('q'):
48     break
49 2.destroyAllWindows

```

Phần lập trình được thực hiện trên Python 3.8 với trình soạn thảo Sublime Text. Từ dòng 1 đến dòng 4, ta khai báo các thư viện cv2, urllib.request, pyfirmata, time. Thư viện cv2 bao gồm tất cả các hàm trong Open CV phục vụ người lập trình trong việc nhận dạng hình ảnh. Thư viện urllib.request phục vụ thao tác với một địa chỉ URL được khai báo vào dòng 14 và mở địa chỉ URL đó trong dòng 27. Thư viện pyfirmata giúp Python giao tiếp được với Arduino, và giao tiếp qua cổng COM3 như dòng 9. Thư viện time được khai báo để sử dụng hàm time.sleep(giây) ở dòng 41, hàm này tương đương delay(mili giây) trong Arduino IDE. Dòng 19 cho phép tạo 1 cửa sổ window mới đặt tên là “gotcha” để chứa video kết quả, đối số cv2.AUTO_SIZE giúp tự động thay đổi kích thước cửa sổ window tùy theo kích thước khung hình video stream.

Kể từ dòng 21 trở đi, đoạn code sẽ chạy theo một vòng lặp while (tương đương với void loop() trong Arduino IDE). Ở dòng 23, Python điều khiển điện áp chân 11 của Arduino ở mức thấp, còi không kêu. Chương trình chuyển ảnh nhận được về ảnh xám để bắt đầu xử lý ở dòng 30 với hàm cvtColor và đối số là COLOR_BGR2GRAY. Ở dòng 33, ta gọi hàm giúp phát hiện cùng lúc nhiều gương mặt detectMultiScale và lưu các tọa độ khuôn mặt vào các biến face. Phương pháp nhận dạng khuôn mặt

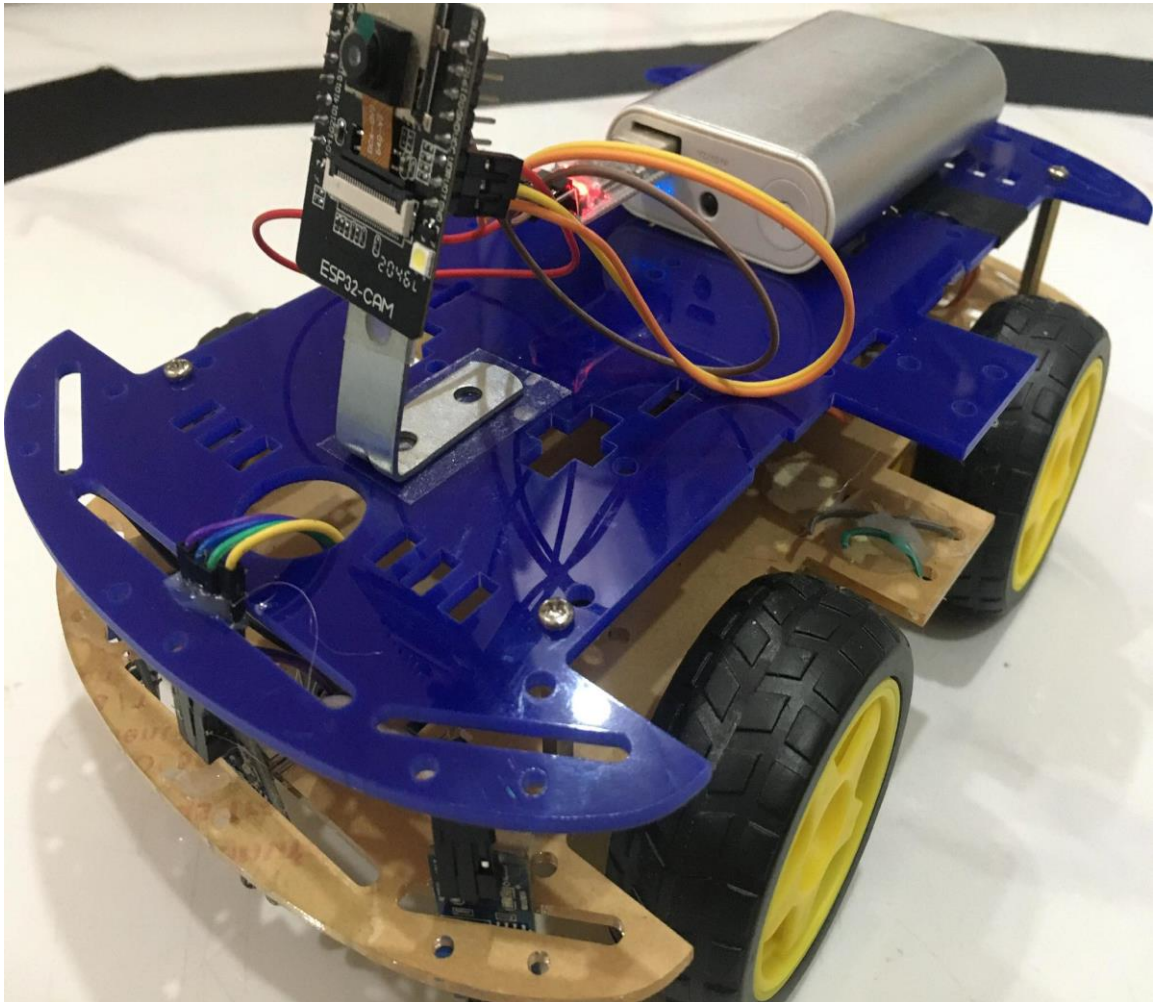
CascadeClassifier là phương pháp đặc trưng Haar được tác giả train trong tệp haarcascade_frontalface_default.xml (đây là một tệp được viết sẵn nên ta download về, để cùng thư mục với file code Python) khai báo ở dòng 6. Ở dòng 37 và 38, khi phát hiện khuôn mặt thì Python tự động tạo 1 hình chữ nhật phù hợp với kích thước khuôn mặt đó (**Hình 3.2.2.3**) bằng hàm rectangle trong thư viện cv2 với đối số img là biến lưu hình ảnh hiện tại, (x,y) và (x+w,y+h) là tọa độ bắt đầu (ở trên bên trái) và kết thúc (ở dưới bên phải) của hình chữ nhật bao lấy khuôn mặt được nhận dạng, đối số thứ 3 là màu của hình chữ nhật (ở đây chọn màu đỏ), đối số cuối cùng là độ dày. Dòng 46 in ra màn hình dòng chữ “Human detected”. Dòng 40, Python điều khiển điện áp chân 11 của Arduino ở mức thấp, còi báo động và delay trong 2 giây bởi hàm time.sleep(2) dòng 41. Dòng 43, 44 để hiển thị ra màn hình video stream. Dòng 47, 48 cho biết nhấn nút “q” để tắt cửa sổ video stream. Dòng 49 giúp dọn dẹp tất cả cửa sổ video hiển thị kết quả sau khi thoát chương trình.



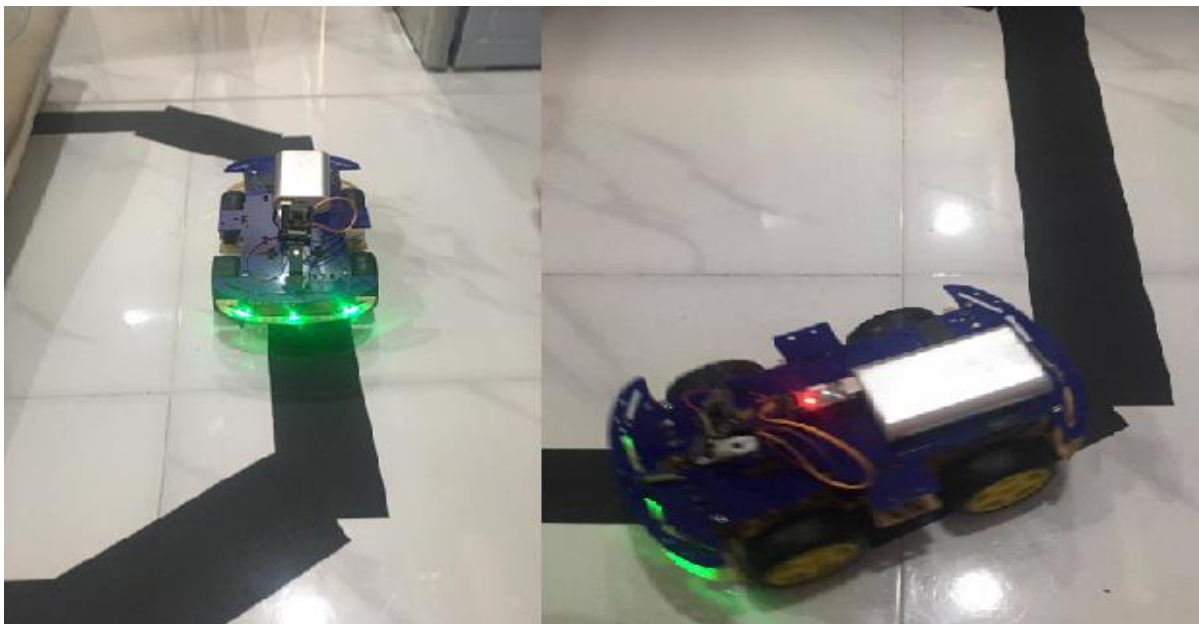
Hình 3.2.3.3: Python tự động tạo 1 hình chữ nhật bao lấy khuôn mặt

3.3. Kết quả

Sau khi hoàn thành bước lập trình cho Python để nhận dạng khuôn mặt, nhóm gắn board ESP32-CAM lên thanh chữ L cùng mạch chuyển PL2303, pin sạc dự phòng trên xe dò line, tiến hành cho xe chạy trên line và xem kết quả ghi hình và xử lý ảnh của camera gửi về bằng Python.



Hình 3.3a: Sản phẩm sau khi hoàn thiện



Hình 3.3b: Hoạt động của xe dò line đi thẳng và quẹo trái



Hình 3.3c: Hoạt động của bộ phận xử lý ảnh, cảnh báo người dùng

CHƯƠNG 4: KẾT LUẬN

Sau gần 3 tháng tìm hiểu, nảy ra ý tưởng và tự học những kiến thức mới về xử lý ảnh, lập trình Python... nhóm đã hoàn thành sản phẩm “Xe giám sát dò line kết hợp xử lý ảnh nhận diện khuôn mặt”. Sản phẩm xe giám sát của nhóm là ý tưởng gói gọn trong cụm từ “Camera biết di chuyển”, là sự kết hợp giữa khả năng di chuyển của xe dò line và khả năng giám sát từ xa của camera Wi-Fi. Hai chức năng này độc lập với nhau, tạo điều kiện để phát triển các sản phẩm khác.

Nhóm em xin cảm ơn Thầy đã hỗ trợ nhóm trong việc làm đồ án môn học và đốc thúc nhóm, việc kịp thời đưa ra những ý kiến đóng góp để bài làm nhóm em được hoàn thiện hơn.

Bài làm của nhóm không tránh khỏi thiếu sót vì có một vài kiến thức quá mới mẻ nhưng thời gian để học lại rất ngắn dẫn đến bài này kiến thức không quá sâu và cặn kẽ mà chỉ nghiêng về hướng ứng dụng, làm mô hình. Vì vậy nên nhóm em mong các thầy thông cảm và đóng góp để nhóm em cải thiện hơn.

TÀI LIỆU THAM KHẢO

- [1] *Công an tỉnh Tiền Giang*. (2021, 5). Retrieved from <http://congan.tiengiang.gov.vn/chi-tiet-tin/?thuc-trang-trom-tai-san-va-bien-phap-phong-ngua/18962379>
- [2] *Công an tỉnh Tiền Giang*. (2021, 5). Retrieved from <http://congan.tiengiang.gov.vn/chi-tiet-tin/?thong-bao-tinh-hinh-trat-tu-xa-hoi-mot-so-thu-oan-hanh-vi-pham-toi-thang-3-2021-huong-dan-noi-dung-phong-chong-toi-pham-tai-te-nan-xa-hoi-thang-4-2021/30054284>
- [3] *Arcdyn*. (2021, 5). Retrieved from <https://www.arcdyn.com/articles/do-surveillance-cameras-actually-deter-criminals/>
- [4] (2021, 5). Retrieved from Store: <https://store.arduino.cc/usa/arduino-uno-rev3>
- [5] (2021, 5). Retrieved from DatasheetPDF: <https://datasheetpdf.com/pdf/1105513/Vishay/TSOP4438/1>
- [6] *Ohtech*. (2021, 5). Retrieved from <https://ohtech.vn/all-courses/lap-trinh-esp32-cam-voi-arduino-ide/lessons/cac-chan-gpio-cua-esp32-cam/>
- [7] OpenCV. Retrieved 5 2021, from https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html