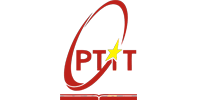
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA KĨ THUẬT ĐIỆN TỬ 1**

****

**BÀI TẬP LỚN**

**ĐỒ ÁN THIẾT KẾ HỆ THỐNG NHÚNG**

**Nhóm: 07**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn** | | : TS Nguyễn Ngọc Minh |
| **Sinh viên thực hiện** | | : |
|  | Vũ Quang Trường | : B19DCDT250 |
|  | Đào Ngọc Anh | : B19DCDT006 |
|  | Lê Ngọc Hoà | : B19DCDT086 |
|  | Lê Quốc Khánh | : B19DCDT121 |

**HÀ NỘI, 2023**

# **LỜI CẢM ƠN**

# Em xin gửi lời cảm ơn đến Học viện Công nghệ Bưu chính viễn thông khi đã đưa môn Thực hành cơ sở vào trong chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn đến giảng viên bộ môn Nguyễn Ngọc Minh đã dạy dỗ, rèn luyện và truyền đạt những kiến thức quý báu cho em trong suốt thời gian kỳ học vừa qua. Trong lớp học của thầy, em đã được tiếp thu thêm nhiều kiến thức bổ ích, học tập được tinh thần làm việc nghiêm túc, hiệu quả. Đây thực sự là những điều rất cần thiết cho quá trình học tập và công tác sau này của em.

# Thời lượng môn học không nhiều, mặc dù đã có cố gắng nhưng những hiểu biết và kỹ năng về môn học này của em còn hạn chế. Do đó bài báo cáo của em có thể còn có những thiếu sót và những chỗ chưa chính xác, kính mong giảng viên bộ môn xem xét và góp ý giúp Bài báo cáo của em được hoàn thiện hơn.

# Em xin chân thành cảm ơn!

# Hà Nội, ngày 13 tháng 05 năm 2023

# **MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc135384913)

[DANH MỤC BẢNG BIỂU, SƠ ĐỒ 4](#_Toc135384915)

[DANH MỤC HÌNH ẢNH 4](#_Toc135384916)

[PHẦN 1: TỔNG QUAN VỀ ĐỀ TÀI 5](#_Toc135384917)

[1.1. Bãi đỗ xe thông minh 5](#_Toc135384918)

[1.2. Nội dung nghiên cứu 5](#_Toc135384919)

[1.3. Các linh kiện cần sử dụng 6](#_Toc135384920)

[PHẦN 2: CẤU TẠO VÀ CHỨC NĂNG CỦA CÁC LINH KIỆN 7](#_Toc135384921)

[2.1. Raspberry Pi 3+ 7](#_Toc135384922)

[2.1.1. Mạch Raspberry Pi 3 b+ là gì? 7](#_Toc135384923)

[2.1.2. Các ứng dụng của mạch Raspberry Pi 3 B+ là gì? 11](#_Toc135384924)

[2.2. Cảm biến hồng ngoại 11](#_Toc135384925)

[2.2.1. Cảm biến hồng ngoại là gì? 11](#_Toc135384926)

[2.2.2. Cấu tạo cảm biến hồng ngoại 12](#_Toc135384927)

[2.2.3. Sơ đồ thiết kế cảm biến hồng ngoại 15](#_Toc135384928)

[2.3. Động cơ servo sg-90 16](#_Toc135384929)

[2.4. Màn hình hiển thị 17](#_Toc135384930)

[2.5. Còi và bàn phím, chuột 18](#_Toc135384931)

[2.6. Ngôn ngữ Python OpenCV 19](#_Toc135384932)

[PHẦN 3: THIẾT KẾ BÃI ĐỖ XE 21](#_Toc135384933)

[3.1. Mô hình bãi đỗ xe 21](#_Toc135384934)

[3.2. Xử lý ảnh 22](#_Toc135384935)

[3.3. Phần điều khiển 27](#_Toc135384936)

[3.4. Kết quả thực nghiệm 31](#_Toc135384937)

\

\

# **DANH MỤC BẢNG BIỂU, SƠ ĐỒ**

[Sơ đồ 2.1. Sơ đồ cảm biến hồng ngoại 16](#_Toc135384834)

[Sơ đồ 3.1. Sơ đồ bãi đỗ xe 21](#_Toc135384844)

# **DANH MỤC HÌNH ẢNH**

[Hình 2.1. Raspberry Pi 3B+ 6](#_Toc135384078)

[Hình 2.2. Chân Pi 8](#_Toc135384079)

[Hình 2.3. Mô tả hoạt động của một cảm biến hồng ngoại 11](#_Toc135384084)

[Hình 2.4. Mô tả hoạt động của một cảm biến hồng ngoại để đo độ sáng 12](#_Toc135384085)

[Hình 2.5. Động cơ servo sg90 13](#_Toc135384089)

[Hình 2.6. Màn hình LCD 13](#_Toc135384091)

[Hình 2.7. Còi báo 14](#_Toc135384093)

# **PHẦN 1: TỔNG QUAN VỀ ĐỀ TÀI**

## **1.1. Bãi đỗ xe thông minh**

Hiện nước ta đang phát triển theo hướng công nghiệp hóa hiện đại hóa, nhu cầu của con người về sự thuận lợi trong cuộc sống ngày càng tăng, ngày mật độ dân cư và xe cộ càng ngày càng đông. Đặc biệt là sự gia tăng về số lượng xe, điều này cũng phản ánh sự phát triển của một quốc gia nhưng cũng hướng dẫn những vấn đề như môi trường, tắc nghẽn giao thông, thiếu bãi đậu xe, cần được giải quyết thiết bị cấp.

Việt Nam là một nước có hệ thống bảo mật, tự động trong các hệ thống quản lí còn thô sơ trong khi đó các nước phát triển đã tiến hành tự động hóa nhằm giảm lực lượng lao động, tăng hiệu quả công việc một cách tối đa. Điển hình là trong một hình thức giữ xe truyền thống, theo phương thức cũ người dân phải bỏ ra nhiều công sức, nhân công lao động để điều hành một hệ thống giữ xe, điều này dẫn đến việc bỏ ra nhiều chi phí cho nhân công và quản lí. Trong khi xã hội ngày càng tiến bộ, kỹ thuật ngày càng tiên tiến hơn, thì nhu cầu đòi hỏi về những thiết bị ứng dụng thông minh, tự động càng nhiều hơn. Và với mong muốn ứng dụng công nghệ vào việc bảo mật và các cảm biến để thực hiện các tác vụ một cách tự động, tiết kiệm sức lao động và đạt được hiệu quả cao, em đã chọn đề tài: “Bãi xe tự động” để thực hiện.

## **1.2. Nội dung nghiên cứu**

- Xác định mục tiêu và giới hạn đề tài.

- Nghiên cứu tài liệu về Raspberry Pi 3+.

- Thiết kế phần xử lý ảnh chuyển đổi biển số sang text.

- Thiết kế, tính toán và thi công cho phần cứng.

- Viết code cho Raspberry Pi 3+.

- Lắp rắp và kết nối giữa khối xử lí trung tâm và các khối khác.

- Hiển thị dữ liệu lên LCD.

- Chạy thử nghiệm hệ thống.

- Chỉnh sửa các lỗi điều khiển, lỗi lập trình và lỗi của các thiết bị.

- Viết báo cáo đồ án.

## **1.3. Các linh kiện cần sử dụng**

- Raspberry Pi 3+

- Camera

- Cảm biến hồng ngoại IR

- Động cơ servo SG -90

- Màn hình hiển thị

- Bàn phím

- Chuột

- Còi báo

# **PHẦN 2: CẤU TẠO VÀ CHỨC NĂNG CỦA CÁC LINH KIỆN**

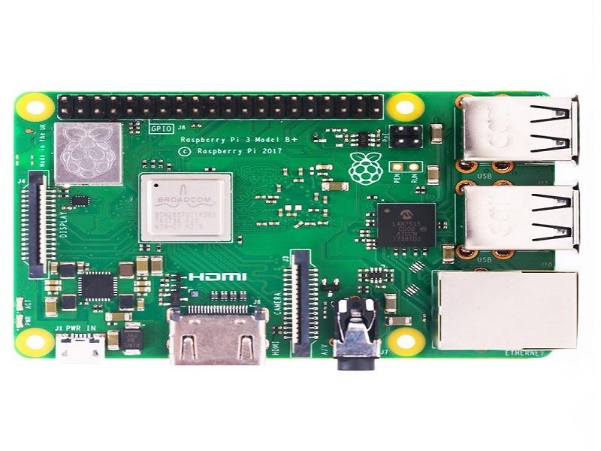
## **2.1. Raspberry Pi 3+**

Raspberry Pi được xem là 1 chiếc máy tính thu nhỏ với đầy đủ phần cứng mạnh mẽ, và hệ điều hành riêng biệt. Ban đầu hãng sản xuất Raspberry Pi nhằm cung cấp cho nhu cầu học tập của trẻ em. Với kích thước nhỏ, ứng dụng được gói gọn và giá cả hợp lý (khoảng mấy chục USD). Nhưng những tính năng trên cũng là điểm hết sức nổi bật của thiết bị so với máy tính thông thường. Nhận thấy tiềm năng, Raspberry Pi được nâng cấp nhiều phiên bản nhanh chóng phát triển rộng rãi. Đến nay trở thành chiếc mini computer nổi bật, và được chú ý nhất. Đã có hàng nghìn ứng dụng được thiết kế và cài đặt riêng cho Raspberry Pi.

### 2.1.1. Mạch Raspberry Pi 3 b+ là gì?

Tuổi đời phát triển của Raspberry Pi còn khá ít khoảng 10 năm. Thiết bị ra mắt đầu tiên vào tháng 2 năm 2012. Tới nay qua sự thay đổi cũng như phát triển đã có nhiều phiên bản mới.

Các phiên bản Raspberry Pi theo thứ tự: Pi A, Pi A+, Pi 1 B, Pi 1 B+, Pi 2B, Pi Zero, Pi 3B, Pi 3 B+, Pi 3 A, Pi 4 B…



#### **Hình 2.1. Raspberry Pi 3B+**

***Thông số kỹ thuật:***

|  |  |
| --- | --- |
| Vi xử lý | Broadcom BCM2837B0Quad-core A53 (ARMv8). 64-bit SoC @1.4GHz |
| RAM | 1GB LPDDR2 SDRAM |
| Kết nối | 2.4GHz and 5GHz (IEEE 802.11 b/g/n/ac wireless LAN) Bluetooth 4.2  BLE  Gigabit Ethernet over USB 2.0 (Tối đa 300Mbps) |
| Cổng USB | 4 x 2.0 |
| Mở rộng | 40-pin GPIO |
| Video và âm thanh | Cổng full-sized HDMI Cổng MIPI DSI Display  Cổng MIPI CSI Camera  Cổng stereo output  Composite video 4 chân |
| Multimedia | H.264, MPEG-4 decode (1080p30) H.264 encode (1080p30)  OpenGL ES 1.1, 2.0 graphics |
| Lưu trữ | MicroSD |
| Nguồn điện sử dụng | 5V/2.5A DC micro USB5V DC chân GPIO |



#### **Hình 2.2. Chân Pi**

Một số cải tiến chính trên Raspberry Pi 3 Model B+ bao gồm các bộ vi xử lý và khả năng kết nối mạng. Model B+ sử dụng vi xử lý Broadcom BCM2837B0 4 nhân 1.4GHz, xung nhịp cao hơn so với BCM2837 1.2GHz trên Pi 3 Model B.

Model B+ cung cấp kết nối không dây Wifi băng tần 2.4GHz và 5GHz, Ethernet gigabit qua cổng USB 2.0 có tốc độ lên đến 300Mbps, gấp 3 lần so với Pi 3 Model B. Thiết bị hỗ trợ Bluetooth 4.2 và Bluetooth Low Energy giúp kết nối tốt hơn với các thiết bị thông minh khác.

Model B+ hỗ trợ Power over Ethernet (PoE), trước đây chưa được hỗ trợ chính thức. Nếu như trước đây các nhà sản xuất linh kiện tạo ra các giải pháp riêng của họ, thì Raspberry Pi Foundation sẽ sớm phát hành PoE HAT cho các nhà sản xuất trong tương lai gần.

Raspberry Pi 3 nâng cấp vi xử lý và kết nối mạng nhanh hơn trên Model B+ nhưng vẫn giữ nguyên được kích thước nhỏ gọn, không thay đổi so với các phiên bản trước đó và giá bán cũng không thay đổi.

### 2.1.2. Các ứng dụng của mạch Raspberry Pi 3 B+ là gì?

- Sử dụng như máy tính để bàn

Do có đầy đủ các phần cứng hỗ trợ nên Raspberry Pi 3 hoàn toàn có thể trở thành 1 chiếc máy tính bình thường. Chuột bàn phím sẽ được kết nối qua cổng USB. Các thiết bị màn hình loa âm thanh có thể kết nối qua Bluetooth hay Wifi. Lựa chọn tải những phần mềm phù hợp yêu cầu bạn sẽ có 1 chiếu máy tính với Case siêu tí hon.

- Làm cổng kết nối cho máy in

Có nhiều người nhiều doanh nghiệp sẽ có 1 lượng lớn máy in cũ, chưa tích hợp tính năng in không dây. Điều này làm cho công việc khá rắc rối. Raspberry Pi 3 dễ dàng xử lý vấn đề này. Chúng ta chỉ cần kết nối máy in để Raspberry Pi 3 trở thành máy chủ điều khiển. Cài một số phần mềm như Samba, CUPS để có thể nhận và chia sẻ dữ liệu từ thiết bị khác. Từ đó mỗi lần muốn in chúng ta chỉ cần truyền dữ liệu qua Raspberry Pi 3.

## **2.1. Cảm biến hồng ngoại**

### 2.1.1. Cảm biến hồng ngoại là gì?

Cảm biến hồng ngoại (IR Sensor) là một thiết bị điện tử đo và phát hiện bức xạ hồng ngoại trong môi trường xung quanh. Bức xạ hồng ngoại đã vô tình được phát hiện bởi một nhà thiên văn học tên là William Herchel vào năm 1800. Trong khi đo nhiệt độ của từng màu ánh sáng (cách nhau bởi một lăng kính), ông nhận thấy rằng nhiệt độ vượt ra ngoài ánh sáng đỏ là cao nhất. IR Sensor là vô hình đối với mắt người, vì bước sóng của nó dài hơn ánh sáng khả kiến ​​(mặc dù nó vẫn nằm trên cùng một phổ điện từ). Bất cứ thứ gì phát ra nhiệt (mọi thứ có nhiệt độ trên năm độ Kelvin) đều phát ra bức xạ hồng ngoại.

Thiết kế cảm biến hồng ngoại nhằm xác định vật thể phát ra bức xạ có nhiệt độ lớn hơn 5 độ K, với độ nhạy cao. Hiện nay, có thể phân loại cảm biến hồng ngoại thành 2 loại:

a. Chủ động (AIR): Có cấu tạo gồm diode phát sáng (LED) và máy thu. Khi một vật thể đến gần cảm biến, thiết bị sẽ chủ động phát ra ánh sáng hồng ngoại từ đèn LED và phản xạ vào vật thể trở lại mà người nhận có thể nhận thấy được. Ứng dụng này thường được sử dụng trong các hệ thống phát hiện chướng ngại vật (như trong robot).

b. Thụ động (PIR): Thiết bị không thể phát ra bức xạ hồng ngoại mà chỉ có thể phát hiện được bức xạ phát ra từ các vật thể khác như người, độc vật hoặc nguồn nhiệt. Cảm biến hồng ngoại thụ động sẽ phát hiện vật thể phát bức xạ hồng ngoại và chuyển tín hiệu thành báo động. Vì vậy người ta gọi đó là thụ động, chỉ phát hiện chứ không phải là nguồn phát ra tia hồng ngoại.

Cấu tạo và nguyên lý hoạt động thiết bị cảm biến hồng ngoại

Cảm biến hồng ngoại hoạt động dựa trên nguyên tắc cảm biến ánh sáng. Đồng thời, thiết bị cảm ứng hồng ngoại có cấu tạo tương tự so với cảm biến ánh sáng.

### 2.1.2. Cấu tạo cảm biến hồng ngoại

Cấu tạo cảm biến hồng ngoại bao gồm các chi tiết sau:

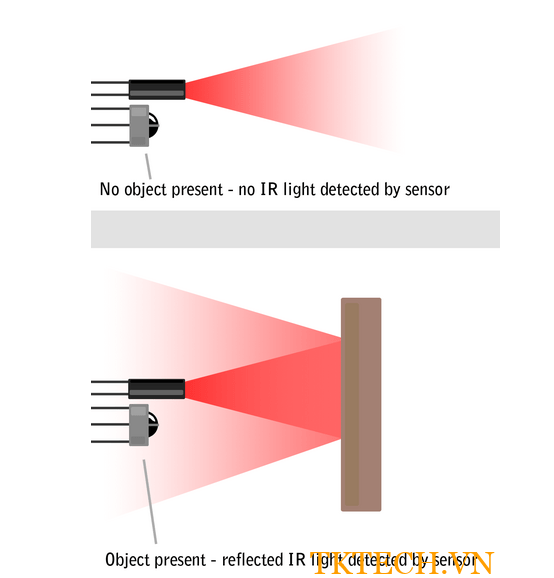
+ Đèn led hồng ngoại - thiết bị phát ra nguồn sáng hồng ngoại.

+ Máy dò hồng ngoại - thiết bị nhận tín hiệu và phát hiện bức xạ hồng ngoại phản xạ trở lại.

+ Điện trở - Thiết bị có tác dụng cản trở cường độ dòng điện quá lớn chạy quá đèn led làm hệ thống chập cháy.

+ Dây điện - Có tác dụng kết nối các chi tiết tạo nên cảm biến hồng ngoại hoạt động ổn định.

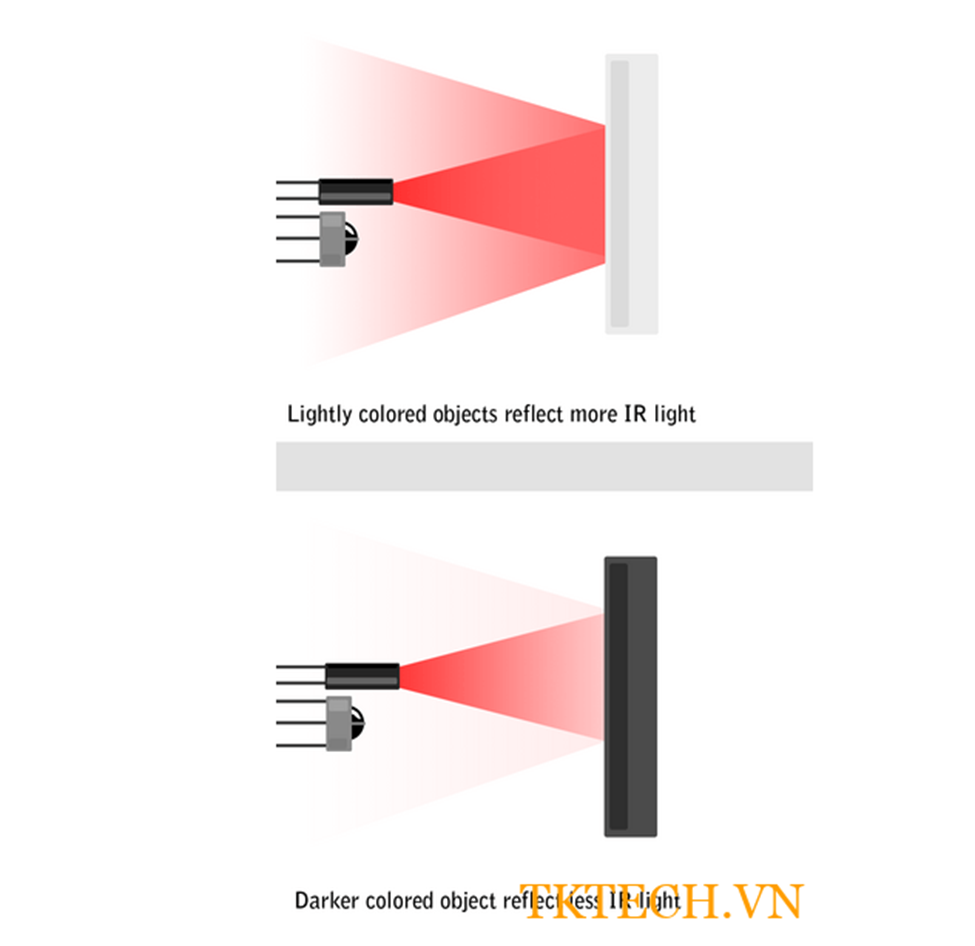
Nguyên lý hoạt động của cảm biến hồng ngoại không quá phức tạp, có thể hiểu đơn giản như sau: Bât kỳ một vật thể nào cũng phát ra các bước xạ hồng ngoại, nhiều hoặc ít. Nên khi một người hoặc vật thể lại đi ngang qua thiết bị cảm biến hồng ngoại sẽ xuất hiện 1 tín hiệu. Tín hiệu này sẽ được cảm biến thu vào và cho vào mạch xử lý, tạo tác dụng điều khiển hay báo động khi cần thiết.



#### **Hình 2.3. Mô tả hoạt động của một cảm biến hồng ngoại**

Phát hiện độ sáng

Vì cảm biến hoạt động bằng cách tìm kiếm ánh sáng phản xạ, có thể có một cảm biến có thể trả về giá trị của ánh sáng phản xạ. Loại cảm biến này sau đó có thể được sử dụng để đo mức độ “sáng” của vật thể. Điều này rất hữu ích cho các nhiệm vụ như theo dõi dòng.

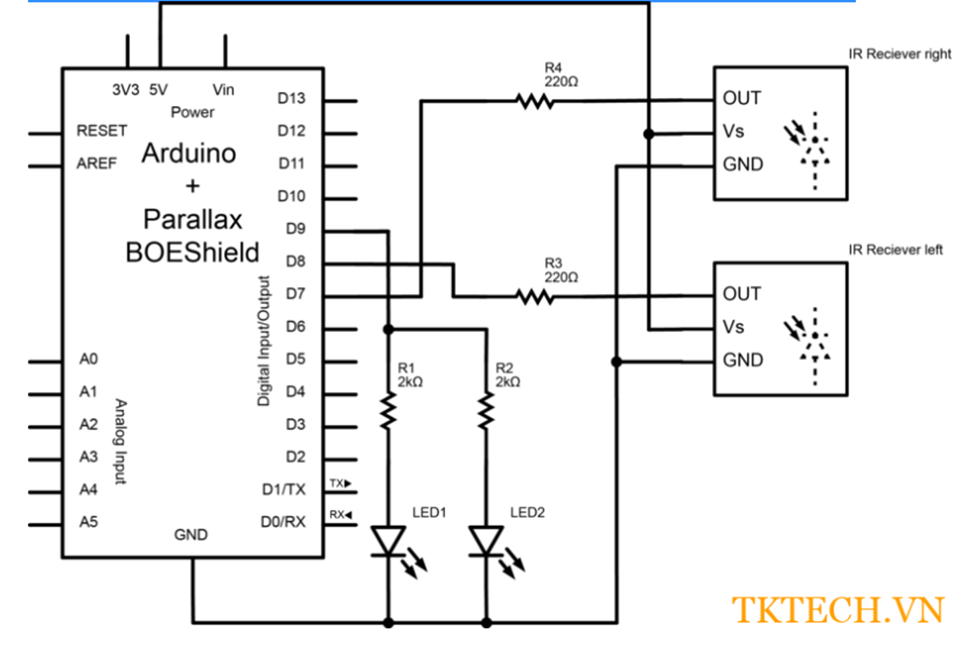


#### **Hình 2.4. Mô tả hoạt động của một cảm biến hồng ngoại để đo độ sáng**

### 

### 2.1.3. Sơ đồ thiết kế cảm biến hồng ngoại

Vì cảm biến hồng ngoại sử dụng cảm biến ánh sáng, nên sơ đồ rất giống với cảm biến ánh sáng. Sự khác biệt duy nhất là việc bổ sung đèn LED hồng ngoại và đầu dò hồng ngoại yêu cầu kết nối với 5V và nối đất.



#### **Sơ đồ 2.1. Sơ đồ cảm biến hồng ngoại**

## **2.3. Động cơ servo sg-90**

- Động cơ RC Servo 9G là động phổ biến dùng trong các mô hình điều khiển nhỏ và đơn giản như cánh tay robot. Động cơ có tốc độ phản ứng nhanh, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

- Đặc điểm:

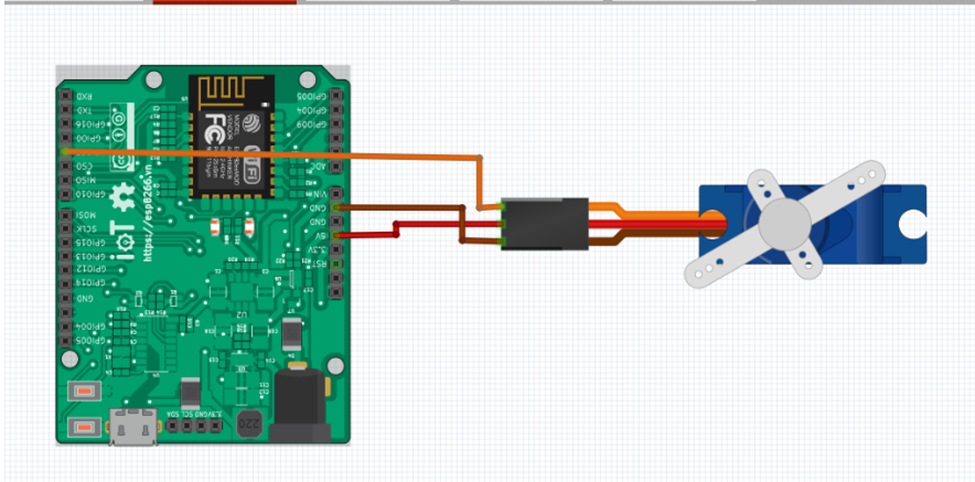
+ Kích thước: 23mm x 12.2mm x 29mm

+ Trọng lượng: 9 gram

+ Điện áp hoạt động: 4.2-6V

+ Nhiệt độ: 0 ℃ 🡪 55 ℃

+ Tốc độ: 0.3 giây/60 độ



#### **Hình 2.5. Động cơ servo sg90**

## **2.4. Màn hình hiển thị**

Màn hình hiển thị: Màn hình hiển thị LCD (Liquid Crystal Display – Màn hình tinh thể lỏng) được sử dụng trong nhiều thiết bị như lò vi sóng, máy tính v.v… Chúng đóng một vai trò rất quan trọng trong các hệ thống nhúng. Màn hình LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: Nó có khả năng hiển thị ký tự đa dạng, trực quan (chữ, số và ký tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn ít tài nguyên hệ thống và giá thành rẽ, … Màn hình có kích thước nhỏ gọn và được kết nối với Raspberry Pi thông qua các giao thức tốc độ cao.



#### **Hình 2.6. Màn hình LCD**

## **2.5. Còi và bàn phím, chuột**

****

#### **Hình 2.7. Còi**

Còi trong mô hình bãi đỗ xe thông minh có thể được sử dụng để cảnh báo cháy hoặc báo tài xế, người đi bộ về những tình huống nguy hiểm, như xe đi ngược chiều hoặc người đi bộ đang đi vào khu vực nguy hiểm. Các cảm biến được lắp đặt trong bãi đỗ xe thông minh có thể giám sát các hoạt động trong bãi đỗ xe và phát hiện ra những tình huống đó.

Bàn phím, chuột: Trong quá trình setup ban đầu cho Raspberry Pi, bàn phím và chuột là hai thiết bị cần thiết để tương tác với hệ điều hành của Raspberry Pi. Đây là hướng dẫn cơ bản để sử dụng bàn phím và chuột trong quá trình setup ban đầu cho Raspberry Pi:

1. Kết nối bàn phím và chuột vào Pi thông qua cổng USB của Raspberry Pi.
2. Khởi động Raspberry Pi bằng cách cắm nguồn vào và chờ cho hệ thống khởi động lên.
3. Trong quá trình khởi động, Raspberry Pi sẽ yêu cầu đăng nhập. Sử dụng bàn phím để nhập tên đăng nhập và mật khẩu vài các trường tương ứng.
4. Sau khi đăng nhập thành công, bạn có thể sử dụng chuột và bàn phím để điều khiển và cài đặt các chương trình và ứng dụng trên Pi.

## **2.6. Ngôn ngữ Python OpenCV**

Python OpenCV là một thư viện dành cho các tác vụ xử lí hình ảnh. Nó là một thư viện phần mềm thị giác máy tính và máy học mã nguồn mở được viết bằng C++ với các liên kết dành cho Python, Java và C. Nó cung cấp các chức năng và thuật toán khác nhau để phân tích hình ảnh và video, phát hiện đối tượng, nhận dạng khuôn mặt và máy móc.

OpenCV ban đầu được phát hiện bởi Intel và sau đó được hỗ trợ bởi Willow Garage và Itseez. Thư viện này có cộng đồng lớn gồm các nhà phát triển và người dùng, nó được sử dụng rộng rãi trong công nghiệp, học viện và nghiên cứu.

Python OpenCV cung cấp bộ công cụ toàn diện để làm việc với hình ảnh và luồng video bao gồm đọc và ghi tệp hình ảnh, thao tác với hình ảnh, áp dụng bộ lọc và chuyển đổi, thực hiện phát hiện và khớp tính năng, v.v. nó cũng cung cấp hỗ trợ xử lí video thời gian thực và điều khiển camera.

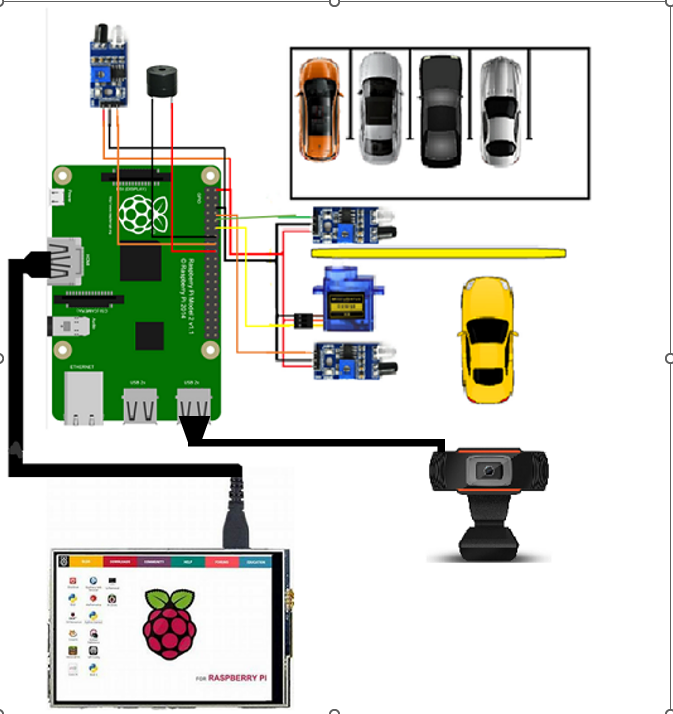
Các chức năng chính của python và open cv:

OpenCV (Open Source Computer Vision) là thư viện thị giác máy tính và máy học phổ biến có hỗ trợ rộng rãi cho Python. đây là một số tính năng chính của Python OpenCV:

1. Xử lý hình ảnh và video: Python OpenCV cung cấp các chức năng và thuật toán khác nhau để xử lý hình ảnh và video, bao gồm các thao tác hình ảnh cơ bản, lọc, phát hiện tính năng và nhận dạng đối tượng.
2. Phát hiện đối tượng: OpenCV có một số bộ phân loại được đào tạo trước cho các tác vụ phát hiện đối tượng như phát hiện khuôn mặt, phát hiện mắt và phát hiện người đi bộ. các bộ phân loại này dựa trên các thuật toán học máy như tầng Haar và mạng lưới thần kinh sâu.
3. Học máy: OpenCV cung cấp bộ công cụ và thư viện cho các nhiệm vụ học máy như phân loại, phân cụm và hồi quy. nó cũng bao gồm hỗ trợ cho các khung máy học phổ biến như TensorFlow và PyTorch.
4. Xử lý thời gian thực: OpenCV có hỗ trợ xử lý các luồng video theo thời gian thực, làm cho nó trở nên hữu ích cho các ứng dụng như giám sát, người máy và phân tích video.
5. Hỗ trợ đa nền tảng: Python OpenCV là đa nền tảng và có thể được sử dụng trên các hệ điều hành khác nhau như Windows, Linux và macOS. nó cũng hỗ trợ các ngôn ngữ lập trình khác nhau như Python, C++ và Java.
6. Tích hợp với các thư viện khác: OpenCV có thể được tích hợp với các thư viện phổ biến khác như NumPy và Matplotlib để thực hiện các tác vụ xử lý hình ảnh và trực quan hóa nâng cao.

**PHẦN 3: THIẾT KẾ BÃI ĐỖ XE**

**3.1. Mô hình bãi đỗ xe**



**Hình 3.1. Sơ đồ bãi đỗ xe**

Bãi đỗ xe gồm 2 phần chính:

+ Phần xử lý ảnh:

Phần xử lý ảnh sẽ xử lý biển số sang dạng text khi có xe ra hoặc vào sau đó lưu vào file csv cùng với thời gian xe ra vào.

+ Phần điều khiển:

Module cảm biến hồng ngoại 1: phát hiện xe vào truyền tín hiệu để mở servo đồng thời trừ chỗ trống trong bãi đỗ xe đi 1.

Module cảm biến hồng ngoại 2: phát hiện xe ra truyền tín hiệu để mở servo đồng thời cộng chỗ trống trong bãi đỗ xe lên 1.

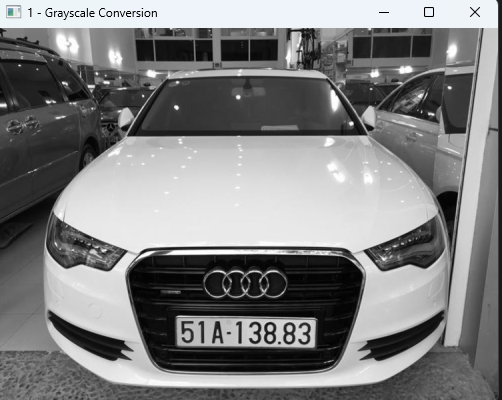
Module cảm biến hồng ngoại 3: phát hiện cháy khi có cháy servo phát tín hiệu còi kêu cảnh báo và gửi thông báo lên màn hình đồng thời mở cho các xe ra.

## **3.2. Xử lý ảnh**

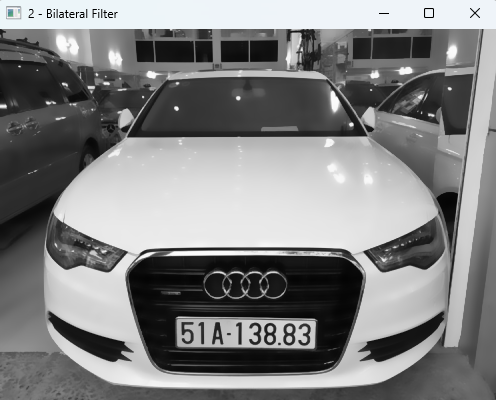
Ngôn ngữ sử dụng python và opencv

Các bước xử lý ảnh:

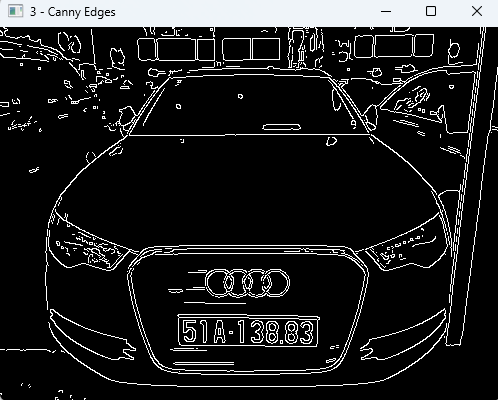
# Chuyển đổi sang ảnh xámgray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

****

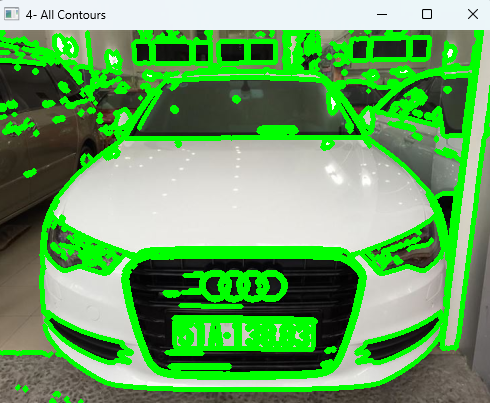
# Lọc ảnh để giảm nhiễugray = cv2.bilateralFilter(gray, 11, 17, 17)

****

# Phát hiện biên cạnh sử dụng thuật toán Cannyedged = cv2.Canny(gray, 170, 200)

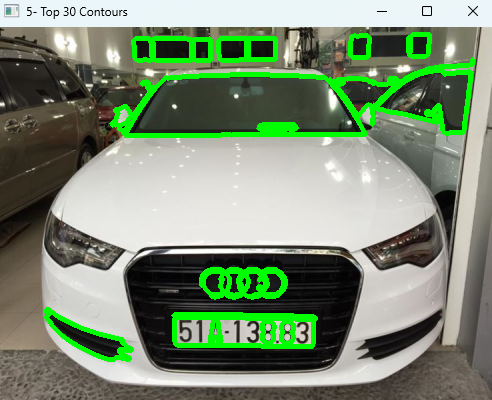


# Tìm các đường viền trong ảnh bằng phương pháp findContourscnts, new  = cv2.findContours(edged.copy(), cv2.RETR\_LIST, cv2.CHAIN\_APPROX\_SIMPLE)# Vẽ tất cả các đường viền lên hình ảnhimg1 = image.copy()cv2.drawContours(img1, cnts, -1, (0,255,0), 3)



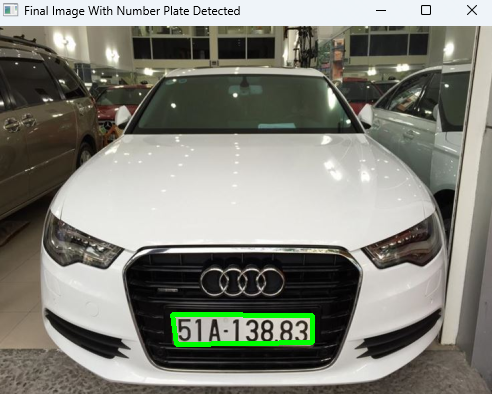
# Sắp xếp các đường viền theo diện tích giảm dần và lấy ra 30 đường viền lớn nhấtcnts=sorted(cnts, key = cv2.contourArea, reverse = True)[:30]NumberPlateCnt = None

# Vẽ 30 đường viền lớn nhất lên hình ảnhimg2 = image.copy()cv2.drawContours(img2, cnts, -1, (0,255,0), 3)



count = 0idx =7for c in cnts:    peri = cv2.arcLength(c, True)    approx = cv2.approxPolyDP(c, 0.02 \* peri, True)

    if len(approx) == 4:          NumberPlateCnt = approx     # Tính toán vị trí và kích thước của đối tượng hình học được phát hiện (c)    x, y, w, h = cv2.boundingRect(c) # Vẽ contour của đối tượng hình học được phát hiện trên ảnh gốc cv2.drawContours(image, [NumberPlateCnt], -1, (0,255,0), 3)



# Cắt ảnh gốc để lấy phần chứa đối tượng được phát hiện    new\_img = gray[y:y + h, x:x + w]



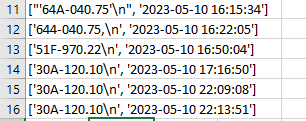
# Nhận dạng ký tự trên ảnh đã cắt sử dụng thư viện Tesseract với ngôn ngữ là tiếng Anh và cấu hình là '--psm 11'text = pytesseract.image\_to\_string(Cropped\_img\_loc, lang='eng', config='--psm 11')



# Lấy thời gian hiện tại    current\_time = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime())

    # Tạo một list chứa thông tin cần lưu    data = [text ,current\_time]

    # Mở file CSV để ghi dữ liệu vào    with open('license\_plates.csv', 'a', newline='') as file:        writer = csv.writer(file)        writer.writerow([data])



## **3.3. Phần điều khiển**

Khai báo các thư viện

import RPi.GPIO as GPIOimport timeimport easygui

GPIO.setmode(GPIO.BOARD)GPIO.setup(8,GPIO.IN)GPIO.setup(10,GPIO.IN)GPIO.setup(16,GPIO.IN)GPIO.setup(12,GPIO.OUT)GPIO.setup(18,GPIO.OUT)pwm=GPIO.PWM(12, 50)pwm.start(0)

Khai báo chỗ trống trong bãi đỗ xe

num\_parking\_spots = 4

Cài đặt góc xoay servo

def SetAngle(angle): duty = angle / 18 + 2 GPIO.output(12, True) pwm.ChangeDutyCycle(duty) time.sleep(1) GPIO.output(12, False) pwm.ChangeDutyCycle(0)



Khi có xe vào cảm biến hồng ngoại 1 gán chân tín hiệu 8 phát hiện vật cản đồng thời trong bãi còn chỗ trống thì servo sẽ mở cho xe vào và trừ chỗ trống còn lại đi 1, nếu hết chỗ trống thì servo sẽ không mở lên nữa và hiện thông báo hết chỗ trống.

while True: if GPIO.input(8) == 0: if num\_parking\_spots > 0: num\_parking\_spots -= 1 print("Cho trong con lai: " + str(num\_parking\_spots)) SetAngle(90) time.sleep(2) SetAngle(0) else: print("Bãi đỗ xe đã hết chỗ trống.") time.sleep(3)



Khi có xe ra cảm biến hồng ngoại 2 gán chân tín hiệu 10 phát hiện vật cản thì servo sẽ mở cho xe ra và cộng chỗ trống còn lại đi 1, nếu hết xe thì hiện thông báo hết xe

elif GPIO.input(10) == 0:

if num\_parking\_spots < 4:

num\_parking\_spots += 1

print("Cho trong con lai: " + str(num\_parking\_spots))

SetAngle(90)

time.sleep(1)

SetAngle(0)

else:

print("Trong bãi không còn xe.")

time.sleep(1)



Khi phát cảm biến phát hiện có cháy thì sẽ phát tín hiệu còi kêu đòng thời gửi thông báo cháy lên màn hình và mở servo cho xe ra.

'''elif GPIO.input(16)==0:

GPIO.output(18,True)

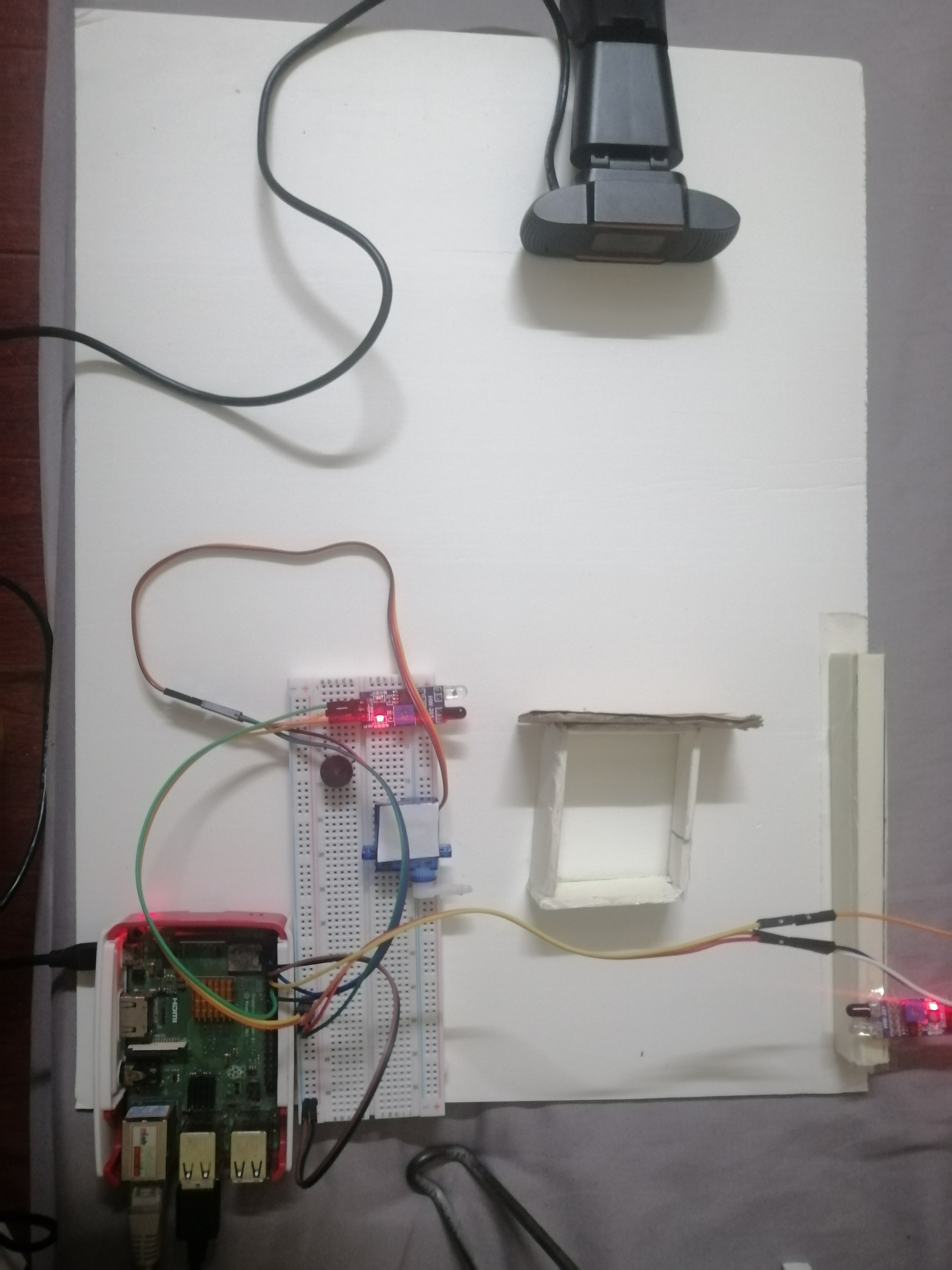
time.sleep(0.1)

SetAngle(90)

easygui.msgbox("co chay", title="canh bao")

elif GPIO.input(16)==1:

## **3.4. Kết quả thực nghiệm**



**Hình 3.2 Sơ đồ thực tế**