

## Phát triển xe tự hành nhận diện biển báo và vật cản

Phan Minh Thuận<sup>1</sup>, Huỳnh Nhật Nam<sup>2</sup>, Nguyễn Thái Văn<sup>3</sup>, Đặng Thế Kỳ<sup>4</sup>

<sup>1</sup> Đại học sư phạm Kỹ thuật Tp. Hồ Chí Minh

<sup>2</sup> Đại học sư phạm Kỹ thuật Tp. Hồ Chí Minh

<sup>3</sup> Đại học sư phạm Kỹ thuật Tp. Hồ Chí Minh

<sup>4</sup> Đại học sư phạm Kỹ thuật Tp. Hồ Chí Minh

### THÔNG TIN BÀI VIẾT

Ngày nhận bài:  
Ngày hoàn thiện:  
Ngày chấp nhận đăng:  
Ngày đăng:

### TỪ KHÓA

Xe tự hành;  
Dò line;  
Cảm biến;  
Thị giác máy tính;  
Trí tuệ nhân tạo.

### TÓM TẮT

Đề tài "Phát triển xe tự hành nhận diện biển báo và vật cản" được chọn vì mong muốn tìm hiểu theo xu hướng về công nghệ tự động hóa hiện nay, cũng như tìm hiểu về các sự mới mẻ của công nghệ tự động hóa và cũng vì xe tự hành đang là một chủ đề nghiên cứu thu hút sự chú ý toàn cầu. Mục tiêu của chúng tôi là phát triển một mẫu thử xe tự hành có khả năng thực hiện các nhiệm vụ chính như dò line, bật coi khi phát hiện và né tránh vật cản, cũng như dừng nhận diện được biển báo "Stop". Việc phát triển mẫu thử xe tự hành giúp hiểu rõ hơn về xu hướng xe tự hành hiện nay, sinh viên có cơ hội áp dụng những môn đã học về lập trình, trí tuệ nhân tạo vào đề tài. Khi tham gia vào đề tài này, sinh viên sẽ được tiếp cận các công nghệ mới từ đó mở rộng tầm hiểu biết của mình về lĩnh vực tự động hóa. Đặc biệt là luyện tập kỹ năng làm việc nhóm cũng như các kỹ năng về nghiên cứu khoa học.

Doi: <https://doi.org/10.54644/jtexxxxxx>

Copyright © JTE. This is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-NonCommercial 4.0 International License](#) which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

## 1. Giới thiệu

### 1.1. Lý do chọn đề tài

Xu hướng xe tự hành hiện nay trên thế giới đang tiếp tục phát triển mạnh mẽ và thu hút sự quan tâm lớn từ cả ngành công nghiệp ô tô và các công ty công nghệ hàng đầu. Các công nghệ và ứng dụng mới liên tục được ra đời, từ các hệ thống tự lái bậc thấp cho đến những công nghệ tự hành hoàn toàn. Các công ty hàng đầu như Tesla, Waymo, và Baidu đang đầu tư mạnh vào nghiên cứu và phát triển các giải pháp xe tự lái. Tesla, với công nghệ Full Self-Driving (FSD), đã thử nghiệm các tính năng tự lái bậc thấp trên toàn cầu, trong khi Waymo và Baidu đã thành công trong việc triển khai dịch vụ taxi tự hành tại một số thành phố. Ngoài ra, cả Uber và Lyft đang tích cực phát triển công nghệ xe tự lái để cung cấp dịch vụ taxi tự lái trong tương lai. Sự cạnh tranh giữa các công ty công nghệ và các nhà sản xuất ô tô đang tạo ra một sân chơi cạnh tranh sôi động và thú vị, đồng thời mở ra những cơ hội mới cho sự đổi mới và tiến bộ trong ngành công nghiệp giao thông và di chuyển toàn cầu.

Tại Việt Nam, xu hướng xe tự hành đang dần trở nên rõ ràng và thu hút sự quan tâm từ các nhà sản xuất công nghệ và khách hàng. Cụ thể, các tổ chức nghiên cứu, trường đại học và các doanh nghiệp công nghệ đang tích cực nghiên cứu và phát triển các giải pháp xe tự hành. Có nhiều dự án đã được khởi tạo, tập trung vào việc áp dụng trí tuệ nhân tạo và các công nghệ cảm biến để cải thiện khả năng tự lái của các loại xe. Trong đó, VinGroup là một trong những tập đoàn lớn và đa ngành hàng đầu tại Việt Nam, đã bắt đầu đầu tư và nghiên cứu về công nghệ xe tự hành. VinFast, công ty con của VinGroup chuyên sản xuất ô tô và xe máy, đã công bố kế hoạch phát triển xe tự lái vào năm 2019. VinFast đang hợp tác với các đối tác trong và ngoài nước để nghiên cứu và phát triển các giải pháp xe tự hành, từ hệ thống cảm biến đến trí tuệ nhân tạo và phần mềm điều khiển. Sự đầu tư mạnh mẽ của VinGroup vào lĩnh vực này cho thấy cam kết

của họ trong việc thúc đẩy sự phát triển của ngành công nghiệp ô tô và công nghệ tự động hóa tại Việt Nam. Ngoài VinGroup thì Viettel và FPT cũng là những tập đoàn lớn đều đang đầu tư vào nghiên cứu và thử nghiệm các công nghệ mới trong lĩnh vực này.

Giới trẻ cụ thể là sinh viên Việt Nam đối với xe tự hành hiện nay thể hiện sự quan tâm và hứng thú ngày càng gia tăng, cùng với sự sẵn lòng tham gia và đóng góp vào việc phát triển của ngành công nghiệp và công nghệ tự động hóa. Sinh viên thể hiện sự quan tâm này thông qua việc tham gia vào các cuộc thi và sự kiện liên quan đến xe tự hành, ngoài ra còn vô số cuộc thi trong nước như Robocon, cuộc thi Robot dò line... Nhằm bắt kịp xu hướng về công nghệ tự động hóa hiện nay, cũng như tìm hiểu về các sự mới mẻ của công nghệ tự động hóa nhóm em đã quyết định chọn đề tài "Phát triển xe tự hành nhận diện biển báo và vật cản".

## **1.2. Mục tiêu**

Mục tiêu của dự án này là phát triển một mẫu thử về xe tự hành có khả năng thực hiện nhiều nhiệm vụ trong môi trường mô phỏng. Xe tự hành được thiết kế để tự động xác định và đi theo line trên một môi trường mô phỏng đường phố, đồng thời có khả năng nhận diện và phản ứng với biển báo "stop" bằng cách dừng lại. Ngoài ra, xe sẽ được trang bị hệ thống cảnh báo là còi báo hiệu và vượt qua vật trên đường mô phỏng, đồng thời trở lại line ban đầu sau khi vượt qua vật cản.

## **2. Phương pháp tiếp cận**

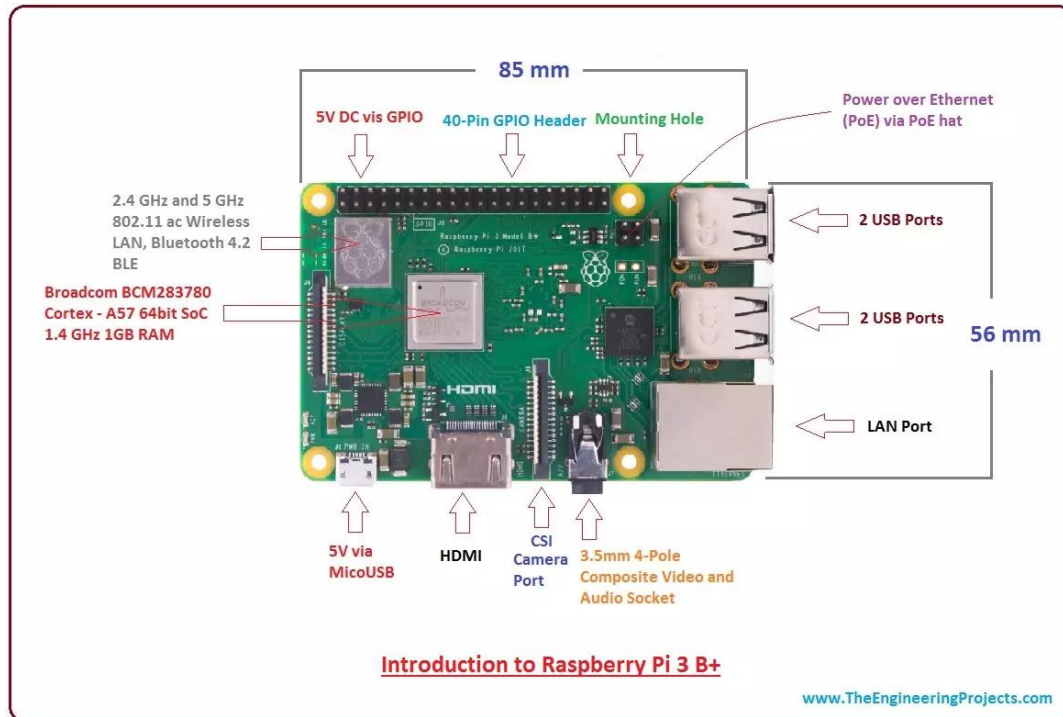
### **2.1. Link kiện**

#### **2.1.1 Raspberry Pi 3 B+**



Raspberry Pi 3 B+ được Raspberry Pi Foundation giới thiệu vào ngày 14 tháng 3 năm 2018. Đây là phiên bản nâng cao của Raspberry Pi 3 B được giới thiệu vào năm 2016.

Nó là một bo mạch máy tính nhỏ đi kèm với CPU, GPU, cổng USB, chân I/O, Wifi, Bluetooth và Network Boot có khả năng thực hiện một số chức năng như một bộ máy tính thông thường.



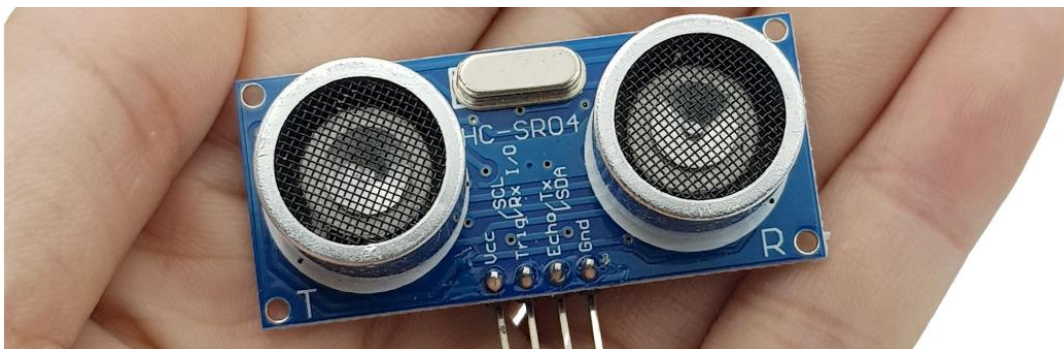
### 2.1.2. Camera Raspberry: picamera2

Camera cho Raspberry Pi có thể được sử dụng để quay video độ phân giải cao, cũng như ảnh tĩnh. Nó dễ sử dụng cho người mới bắt đầu, nhưng cũng có thể cung cấp cho người dùng nâng cao nếu bạn đang muốn mở rộng kiến thức của mình.



### 2.1.3. Cảm biến siêu âm: HSRC04

Cảm biến siêu âm HC-SR04 (Ultrasonic Sensor) được sử dụng rất phổ biến để xác định khoảng cách vì RẺ và CHÍNH XÁC. Cảm biến HC-SR04 sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 -> 300cm, với độ chính xác gần như chỉ phụ thuộc vào cách lập trình.



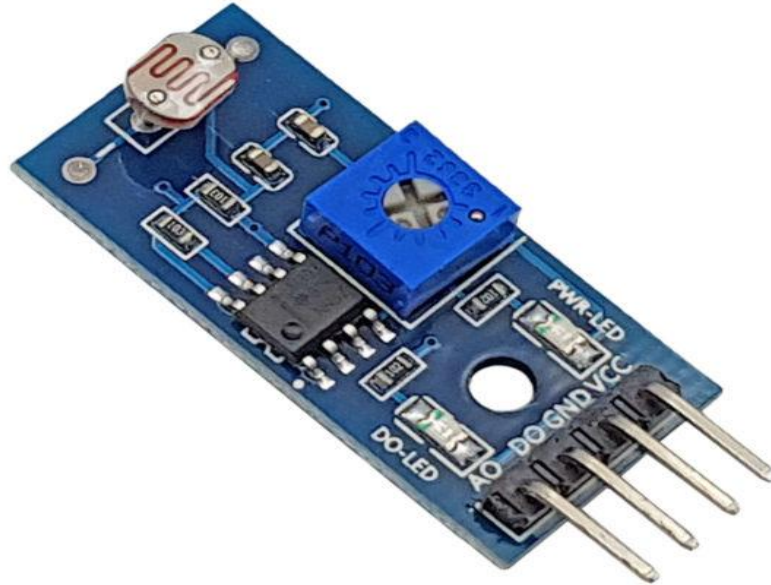
2.1.4. Driver LS98N



Driver L298N là module điều khiển động cơ trong các xe DC và động cơ bước. Module điều khiển động cơ L298N có IC điều khiển động cơ L298, Bộ điều chỉnh điện áp 78M05, điện trở, tụ điện, LED nguồn, jumper 5V tích hợp.. Module L298N có thể điều khiển tối đa 4 động cơ DC hoặc 2 động cơ DC với khả năng điều khiển hướng và tốc độ.



### 2.1.5. Cảm biến ánh sáng



Cảm biến cường độ ánh sáng phát hiện cường độ ánh sáng, sử dụng bộ cảm biến photoresistor loại nhạy cảm, cho tín hiệu ổn định, rõ ràng và chính xác hơn so với quang trở độ nhạy có thể tùy chỉnh. Thiết kế đơn giản nhưng hiệu quả và độ tin cậy cao, độ nhiễu thấp do được thiết kế mạch lọc tín hiệu trước khi so sánh với ngưỡng.

## 2.2. Các thuật toán

### 2.2.1. Xử lý ảnh phát hiện line

Sử dụng động cơ hoạt động. Thông qua hình ảnh từ camera, hệ thống có thể nhận diện được làn đường và từ đó tính toán, điều chỉnh hướng di chuyển của xe.

Đầu tiên, tạo các thư viện OpenCV, Picamera2 và Numpy để xử lý hình ảnh được trích xuất từ camera và điều khiển

mặt nạ tách nền và chuyển đổi ảnh sang không gian màu HSV.

```
imgBorder = cv2.GaussianBlur(img, (5, 5), cv2.BORDER_DEFAULT)
imgHsv = cv2.cvtColor(imgBorder, cv2.COLOR_BGR2HSV)

h_min = cv2.getTrackbarPos("HUE Min", "HSV")
h_max = cv2.getTrackbarPos("HUE Max", "HSV")
s_min = cv2.getTrackbarPos("SAT Min", "HSV")
s_max = cv2.getTrackbarPos("SAT Max", "HSV")
v_min = cv2.getTrackbarPos("VAL Min", "HSV")
v_max = cv2.getTrackbarPos("VAL Max", "HSV")
```

```
lowerWhite = np.array([h_min, s_min, v_min])
upperWhite = np.array([h_max, s_max, v_max])

lowerWhite = np.array([0, 0, 0])
upperWhite = np.array([179, 255, 115])
maskWhite = cv2.inRange(imgHsv, lowerWhite, upperWhite)
```

Tiếp theo, trích xuất góc từ trên cao

```
hT, wT = img.shape[:2]
wrapYTopLeft = 20
wrapYTopRight = 20
wrapXTopLeft = 110
wrapXTopRight = 70

pts1 = [[0, height], [width, height], [width-wrapXTopRight, wrapYTopLeft],
[wrapXTopLeft, wrapYTopRight]]
pts2 = [[0, height], [width, height], [width, 0], [0, 0]]

src = np.float32(pts1)
des = np.float32(pts2)

matrix = cv2.getPerspectiveTransform(src, des)
imgWarp = cv2.warpPerspective(maskWhite, matrix, (wT, hT))
```

Tiếp theo, tính độ lệch giữa trung tâm đường và tâm ảnh

```
minPer = 0.5
region = 4

h, w = imgWarp.shape[:2]
histValues = np.sum(imgWarp[-h//region:, :], axis=0)

maxValue = np.max(histValues)
minValue = minPer * maxValue

indexArray = np.where(histValues >= minValue)
RoadCenter = int(np.average(indexArray))
imgHist = np.zeros((h, w, 3), np.uint8)
```

```
imgCenter = 240  
dist = (RoadCenter - imgCenter)
```

Cuối cùng, làm mịn giá trị khoảng cách, áp dụng phép biến đổi logarit và căn bậc hai để làm mịn giá trị.

```
def smoothed(dist):  
    normalizedDist = dist/240  
    n = np.sqrt(np.log10(abs(normalizedDist)/4+1))  
    return n*(dist>15)-n*(dist<-15)
```

### 2.2.2. Dùng cảm biến siêu âm phát hiện vật cản

Tính khoảng cách bằng cách phát ra sóng âm thanh và sau đó lắng nghe tiếng vang. Nó đo thời gian giữa việc gửi và nhận sóng âm, tính toán khoảng cách dựa trên thời gian này và trả về khoảng cách tính bằng centimet.

```
def measure_distance():  
    gpio.output(trig, False)  
    time.sleep(0.1)  
    gpio.output(trig, True)  
    time.sleep(0.00001)  
    gpio.output(trig, False)  
    while gpio.input(echo) == 0:  
        pulse_start = time.time()  
    while gpio.input(echo) == 1:  
        pulse_end = time.time()  
    pulse_duration = pulse_end - pulse_start  
    distance = pulse_duration * 17000  
    distance = round(distance, 3)  
    return distance
```

### 2.2.3. Nhận diện biển báo giao thông

Sau khi train dữ liệu, thu được file xml

```
<?xml version="1.0"?>
<opencv_storage>
<cascade>
  <stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>24</height>
  <width>13</width>
  <stageParams>
    <boostType>GAB</boostType>
    <minHitRate>9.9500000476837158e-01</minHitRate>
    <maxFalseAlarm>5.0000000000000000e-01</maxFalseAlarm>
    <weightTrimRate>9.4999998807907104e-01</weightTrimRate>
    <maxDepth>1</maxDepth>
    <maxWeakCount>100</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount>
    <featSize>1</featSize>
    <mode>BASIC</mode></featureParams>
  <stageNum>13</stageNum>
  <stages>
    <!-- stage 0 -->
    <_>
      <maxWeakCount>3</maxWeakCount>
      <stageThreshold>-7.9856824874877930e-01</stageThreshold>
      <weakClassifiers>
        <_>
          <internalNodes>
            0 -1 23 -2.4647466838359833e-01</internalNodes>
          <leafValues>
            9.9553900957107544e-01 -8.7567567825317383e-01</leafValues></_>
        <_>
          <internalNodes>
            0 -1 30 -3.8283371832221746e-03</internalNodes>
          <leafValues>
            8.3999037742614746e-01 -8.1806915998458862e-01</leafValues></_>
        <_>
          <internalNodes>
```

Sau đó nhận diện biển báo và trả về kết quả tương ứng, nhóm chúng em chỉ làm trên biển báo Stop.

```
stop_sign = cv2.CascadeClassifier('cascade_stop_sign.xml')

def detect_stop_sign(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```



```
stop_sign_scaled = stop_sign.detectMultiScale(gray, 1.3, 5)

    if len(stop_sign_scaled) > 0: # Nếu có biển báo stop được phát hiện
        for (x, y, w, h) in stop_sign_scaled:
            # Vẽ khung xanh xung quanh biển báo
            img = cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 3)
            # Thêm văn bản "Stop Sign" dưới biển báo
            img = cv2.putText(img, "Stop Sign", (x, y+h+30),
                               cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2,
                               cv2.LINE_4)
        return "stop"
    else:
        return "go"
```

### 3. Kết quả và thảo luận

Nhận diện line: mẫu thử xe tự hành có thể nhận diện 2 line và đi theo 2 line đó.

Nhận diện biển báo: Mẫu xe tự hành này có thể nhận diện biển báo STOP, khi gặp biển STOP, xe sẽ báo còi và dừng lại.

Nhận diện vật cản: Mẫu xe tự hành có thể nhận diện vật cản nhờ vào cảm biến siêu âm HC-SR04, khi phát hiện có vật cản, xe sẽ báo còi và vượt qua vật cản.

### 4. Kết luận

#### Lời cảm ơn

Chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Xuân Sâm, người đã tận tâm hướng dẫn và truyền đạt kiến thức quý báu trong suốt quá trình thực hiện dự án này. Sự hỗ trợ nhiệt tình và những lời khuyên chân thành của thầy đã giúp chúng em vượt qua nhiều khó khăn và hoàn thành công việc một cách tốt đẹp. Chúng em cũng xin trân trọng cảm ơn các tác giả của những công trình nghiên cứu và bài báo mà chúng em đã tham khảo. Những kiến thức và thông tin từ các công trình này đã đóng góp quan trọng vào sự phát triển của dự án. Chúng em thật sự biết ơn sự giúp đỡ vô giá này và hy vọng rằng trong tương lai, chúng em sẽ có cơ hội tiếp bước thầy và các tác giả, để tiếp tục hỗ trợ và truyền cảm hứng cho các thế hệ nghiên cứu tiếp theo, góp phần phát triển lĩnh vực này ngày càng thịnh vượng.

#### Xung đột lợi ích

Các tác giả tuyên bố không có xung đột lợi ích trong bài báo này.

#### Tuyên bố dữ liệu sẵn có

Dữ liệu hỗ trợ cho các khám phá của nghiên cứu này khi độc giả yêu cầu một cách hợp lý sẽ được tác giả liên hệ cung cấp.

#### TÀI LIỆU THAM KHẢO

[1] *Introduction to Raspberry Pi 3 B+, David Watson*

<https://www.theengineeringprojects.com/2018/07/introduction-to-raspberry-pi-3-b-plus.html>