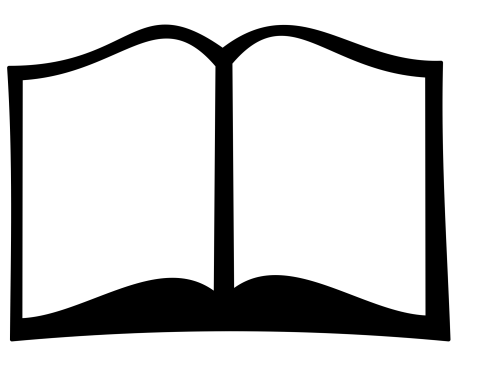
TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC ĐÀ NẴNG

KHOA ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO TIẾN ĐỘ

ĐỒ ÁN TỐT NGHIỆP

**Đề tài:** **THIẾT KẾ VÀ THI CÔNG MÔ HÌNH XE TỰ HÀNH-  
NHẬN DẠNG VÀ DI CHUYỂN THEO LÀN ĐƯỜNG**

Sinh viên thực hiện: Đoàn Minh Thiên Phú 12DT4

Nguyễn Văn Thanh 12DT4

Đặng Quốc Bảo 12DT4

Người hướng dẫn: Phạm Minh Hải FPT Software

TS. Nguyễn Thị Anh Thư DUT

TS. Hồ Phước Tiến DUT

MỤC LỤC

[Chương 1: Giới thiệu các thành phần hệ thống 2](#_Toc481042299)

[1.1 Giới thiệu chương 2](#_Toc481042300)

[1.2 Sơ đồ khối hệ thống 3](#_Toc481042301)

[1.3 Raspberry Pi 3 3](#_Toc481042302)

[1.2.1 Raspberry Pi 3 3](#_Toc481042303)

[1.3.2. Hệ điều hành Raspbian 4](#_Toc481042304)

[1.4 Raspberry PI Camera V2 5](#_Toc481042305)

[1.5. Arduino UNO R3 6](#_Toc481042306)

[1.5.1 Arduino UNO R3 6](#_Toc481042307)

[1.5.2 Arduino IDE 8](#_Toc481042308)

[1.6 Servo 11](#_Toc481042309)

[1.7 ESC (Electronic Speed Control) 11](#_Toc481042310)

[1.8 Động cơ 12](#_Toc481042311)

[1.9 Cảm biến từ trường HALL 13](#_Toc481042312)

[1.10. Kết luận chương: 13](#_Toc481042313)

[Chương 2: Thiết kế các thuật toán. 14](#_Toc481042314)

[2.1. Giới thiệu chương. 14](#_Toc481042315)

[2.2. Lý thuyết và các thuật toán xử lý ảnh 14](#_Toc481042316)

[2.2.1 Thư viện OpenCV 14](#_Toc481042317)

[2.2.2 Thuật toán Canny 15](#_Toc481042318)

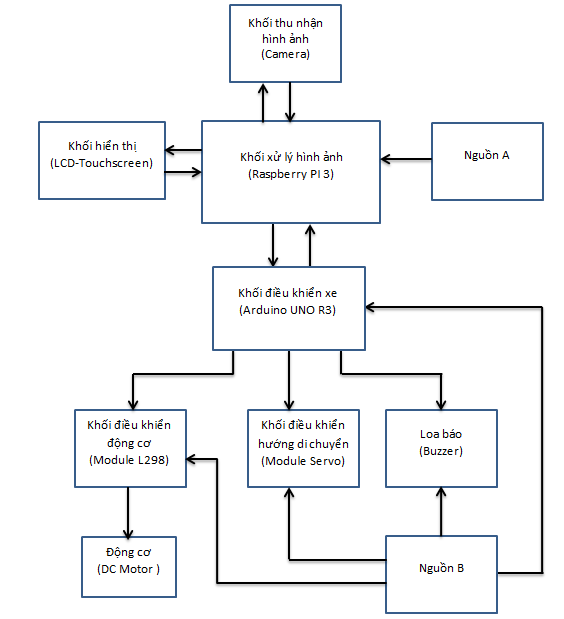
[2.2.3 Thuật toán hough line 17](#_Toc481042319)

[2.2.4 Thuật toán xác định làn đường. 19](#_Toc481042320)

# Chương 1: Giới thiệu các thành phần hệ thống

## 1.1 Giới thiệu chương: chương 1 sẽ giới thiệu các thành phần cấu tạo nên toàn bộ hệ thống.

## 1.2 Sơ đồ khối hệ thống



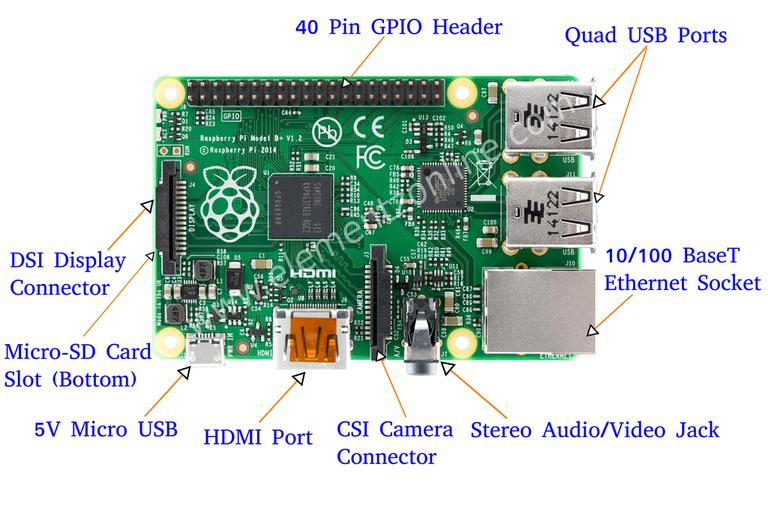
**Hình 1.1**: Sơ đồ khối hệ thống

## 1.3 Raspberry Pi 3

### 1.2.1 Raspberry Pi 3

Raspberry Pi là một máy tính được thiết kế dưới dạng 1 board mạch điện tử có đầy đủ các thành phần cần thiết. Raspberry Pi sử dụng hệ điều hành linux.

Bộ xử lý trung tâm của [Raspberry Pi](http://machtudong.vn/raspberry-pi) là chip SoC (System On Chip) của Broadcom. Chip SoC tích hợp các thành phần cần thiết bao gồm: CPU, GPU, RAM trên duy nhất 1 đế chip tạo điều kiện cho việc thiết kế các hệ thống chạy ổn định nhưng lại yêu cầu kích thước nhỏ.



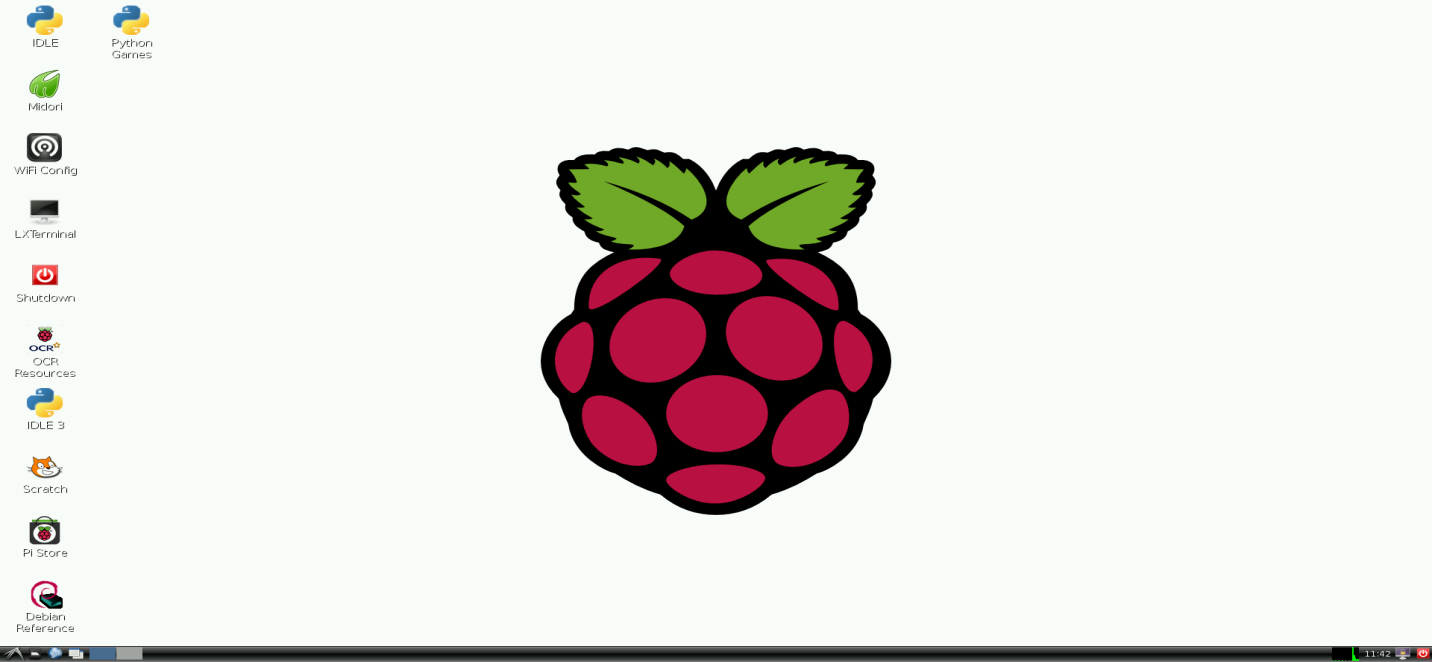
**Hình 1.2**. Mạch Raspberry pi 3

Cấu hình Raspberry Pi B+

* Chip: Broadcom BCM2835 SoC
* Core architecture: ARM11
* CPU: 700 MHz ARM1176JZFS
* GPU: Dual Core VideoCore IV® Multimedia Co-Processor. Hỗ trợ Open GL ES 2.0, tăng tốc phần cứng OpenVG, và bộ giải mã 1080p30 H.264.
* RAM: 512MB SDRAM
* Thẻ nhớ: Sử dụng MicroSD
* Kích thước 85 x 56 x 17mm
* Nguồn: Micro USB 5V, 2A

### 1.3.2. Hệ điều hành Raspbian

Raspbian là một hệ điều hành dễ sử dụng và với sự hỗ trợ tốt của cộng đồng trên thế giới. Hơn thế nữa, Raspbian là phiên bản không chính thức của Debian Wheezy – một  phiên bản Linux có tiếng. Raspbian rất đơn giản và quen thuộc.



**Hình 1.3**. Giao diện hệ điều hành Raspbian

Giống như một chiếc máy tín Windows, Raspbian bao gồm các ứng dụng đa phương tiện và đồ họa (xem ảnh, xem phim, soạn thảo notepad), và nếu bạn cần nhiều hơn, bạn có thể cài thêm các gói phần mềm khác như trình duyệt internet, nhắn tin, bộ phần mềm văn phòng.

## 1.4 Raspberry PI Camera V2

Raspberry Pi Camera Module V2 là một bước nhảy vọt về chất lượng hình ảnh, màu sắc trung thực và hiệu suất ánh sáng thấp . Đặc biệt nó hỗ trợ video lên tới 1080P30, 720P60 và video mode VGA90, cũng như chế độ chụp hình. Nó sử dụng đoạn cáp 15cm qua cổng CSI trên Raspberry Pi.



**Hình 1.4** . Module Camera kết nối với Kit Raspberry pi 3

Thông số kỹ thuật:

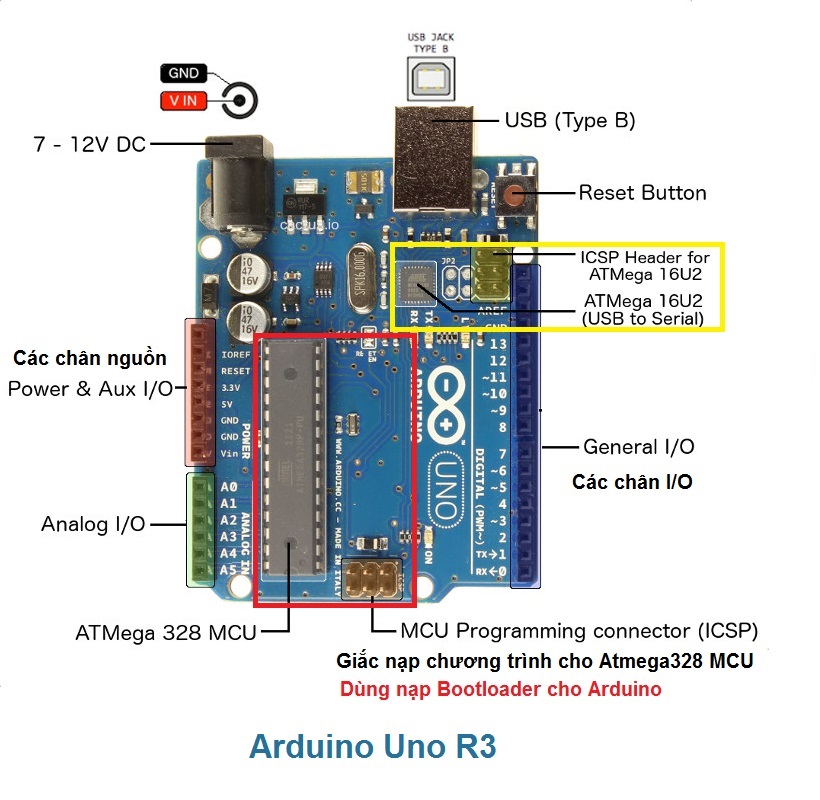
* Ống kính tiêu cự cố định
* Cảm biến độ phân giải 8 megapixel cho khả năng chụp ảnh kích thước 3280 x 2464
* Hỗ trợ video 1080p30, 720p60 và 640x480p90
* Kích thước 25mm x 23mm x 9mm
* Trọng lượng chỉ hơn 3g
* Kết nối với Raspberry Pi thông qua cáp ribbon đi kèm dài 15 cm
* Camera Module được hỗ trợ với phiên bản mới nhất của Raspbian

## 1.5. Arduino UNO R3

### 1.5.1 Arduino UNO R3

Arduino board có rất nhiều phiên bản với hiệu năng và mục đích sử dụng khác nhau như: Arduino Mega, Aruino LilyPad... Trong số đó, Arduino Uno R3 là một trong những phiên bản được sử dụng rộng rãi nhất bởi chi phí và tính linh động của nó.

Arduino Uno được xây dựng với phân nhân là vi điều khiển ATmega328P sử dụng thạch anh có chu kì dao động là 16 MHz. Với vi điều khiển này, ta có tổng cộng 14 pin (ngõ) ra vào được đánh số từ 0 tới 13 (trong đó có 6 pin PWM, được đánh dấu ~ trước mã số của pin). Song song đó, ta có thêm 6 pin nhận tín hiệu analog được đánh kí hiệu từ A0 - A5, 6 pin này cũng có thể sử dụng được như các pin ra / vào bình thường (như pin 0 - 13). Ở các pin được đề cập, pin 13 là pin đặc biệt vì nối trực tiếp với LED trạng thái trên board.



**Hình 1.5**. Mạch Arduino Uno R3

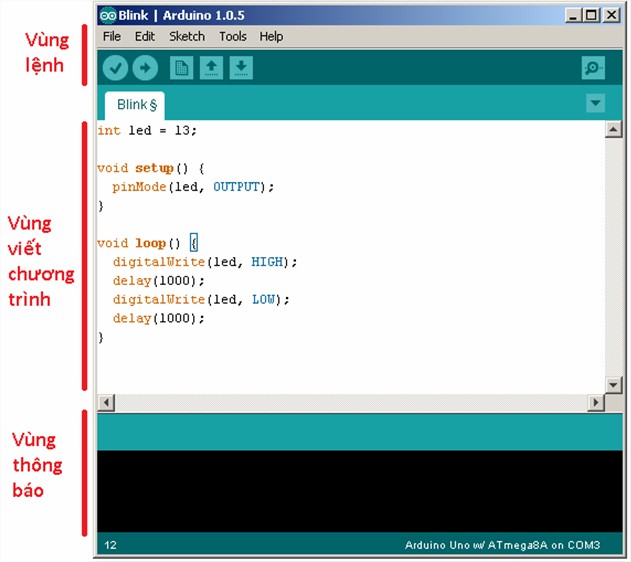
Trên board còn có 1 nút reset, 1 ngõ kết nối với máy tính qua cổng USB và 1 ngõ cấp nguồn sử dụng jack 2.1mm lấy năng lượng trực tiếp từ AC-DC adapter hay thông qua ắc-quy nguồn.

**Thông số của Arduino UNO R3**

|  |  |
| --- | --- |
| 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 khuyên 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) |

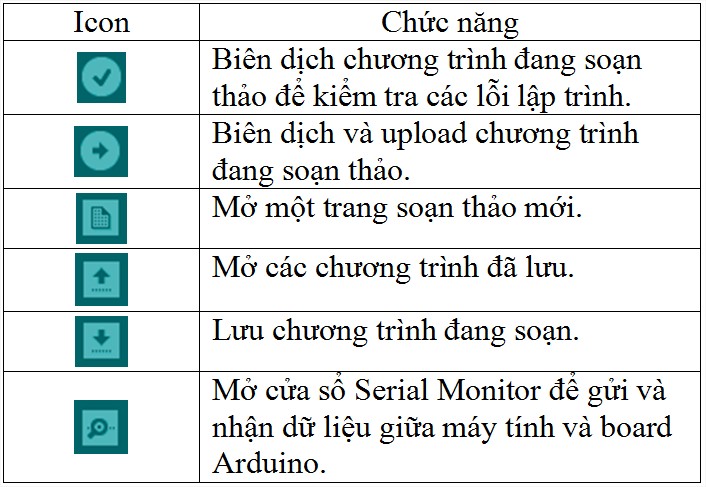
### 1.5.2 Arduino IDE

Để lập trình được cho các board Arduino, các bạn cần phải có một công cụ gọi là **I**ntergrated **D**evelopment **E**nvironment (IDE). Công cụ này được đội ngũ kĩ sư của Arduino phát triển và có thể chạy trên Windows , MAC OS X và Linux



**Hình 1.6**. Giao diện Arduino IDE

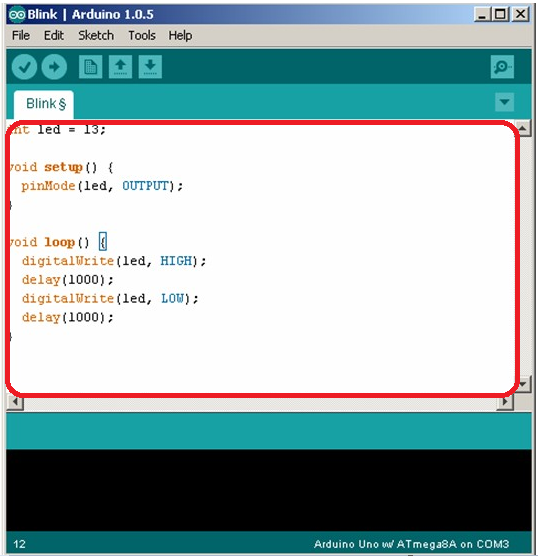
**Vùng lệnh**

[](http://k3.arduino.vn/img/2014/05/24/0/465_8121-1400927938-0--table.jpg)Bao gồm các nút lệnh menu (File, Edit, Sketch, Tools, Help). Phía dưới là các icon cho phép sử dụng nhanh các chức năng thường dùng của IDE được miêu tả như sau:

**Hình 5**. Bảng các nút lệnh trên phần mềm IDE Arduino

**Vùng viết chương trình**

Bạn sẽ viết các đoạn mã của mình tại đây. Tên chương trình của bạn được hiển thị ngay dưới dãy các Icon, ở đây nó tên là “**Blink**”. Để ý rằng phía sau tên chương trình có một dấu “**§**”. Điều đó có nghĩa là đoạn chương trình của bạn chưa được lưu lại.



**Hình 1.7**. Vùng viết chương trình.

**Vùng thông báo (debug)**

[](http://k1.arduino.vn/img/2014/05/24/0/497_1231-1400928031-0--deb.jpg)

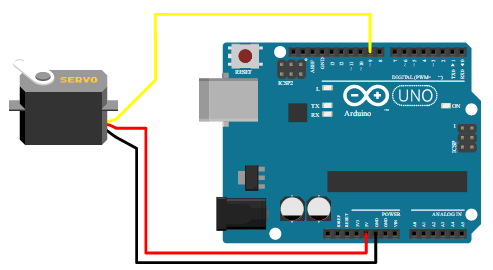
**Hình 1.8**. Vùng thông báo (debug)

Những thông báo từ IDE sẽ được hiển thị tại đây. Để ý rằng góc dưới cùng bên phải hiển thị loại board Arduino và cổng COM được sử dụng. Luôn chú ý tới mục này bởi nếu chọn sai loại board hoặc cổng COM, bạn sẽ không thể upload được code của mình.

## 1.6 Servo

Động cơ servo là loại động cơ cho phép ta điều khiển một cách cực kì chính xác. Vì vậy, khác với động cơ thông thường ta chỉ cần cấp nguồn cho động cơ là có thể vận hành được. Động cơ servo yêu cầu ta phải cấp nguồn (2 dây) và nhận điều khiển từ mạch chính (1 dây), mỗi dây thường được đánh màu như sau:

* Đỏ: nhận điện nguồn, tuỳ vào loại động cơ mà giá trị này có thể khác nhau
* Đen: nối với cực âm của mạch
* Vàng: nhận tín hiệu từ mạch điều khiển



**Hình 1.9**. Arduino kết nối với động cơ servo **Tower Pro MG90S**

Động cơ servo này là loại **Tower Pro MG90S**. Động cơ này quay được 180° (trái 90° và phải 90°).

## 1.7 ****ESC**** (Electronic Speed Control)

ESC là thiết bị điều khiển tốc độ động cơ BLDC thường được ứng dụng rất nhiều trong các mô hình máy bay với đa dạng chủng loại và công suất. Đi đôi với những động cơ BLDC công suất lớn này đồng nghĩa với ESC phải chịu được công suất lớn thường thì phân loại theo dòng chịu đựng có thể từ 10A -> 100A và dĩ nhiên giá cũng hơn cao.



**Hình 1.10**. Bộ điều tốc ESC

Thường thì ESC có 8 dây, trong đó :

-          2 dây là nguồn nuôi động cơ và mạch điều khiển. Nguồn này là nguồn công suất lớn, dòng tiêu thụ có thể lên đến trên 20A tùy loại động cơ. Bạn có thể sử dụng pin lipo, hoặc nếu không mang tính di động bạn có thể sử dụng nguồn xung (nguồn tổ ong) Ampe lớn để thử động cơ.

-          3 dây kết nối với động cơ BLDC

-          3 dây nhận tín hiệu điều khiển sử dụng như RC servo, lưu ý ở RC servo thì ta cần cung cấp nguồn để servo hoạt động nhưng ở ESC trong 3 dây này dây có ký hiệu màu đỏ (hoặc cam) sẽ cấp nguồn 5V ngược ra môi trường.

## 1.8 Động cơ

Động cơ DC không chổi than-BLDC (Brushles Dc motor) là một dạng động cơ đồng bộ tuy nhiên động cơ BLDC kích từ bằng một loại nam châm vĩnh cửu dán trên rotor vàdùng dòng điện DC ba pha cho dây quấn phần ứng stator.

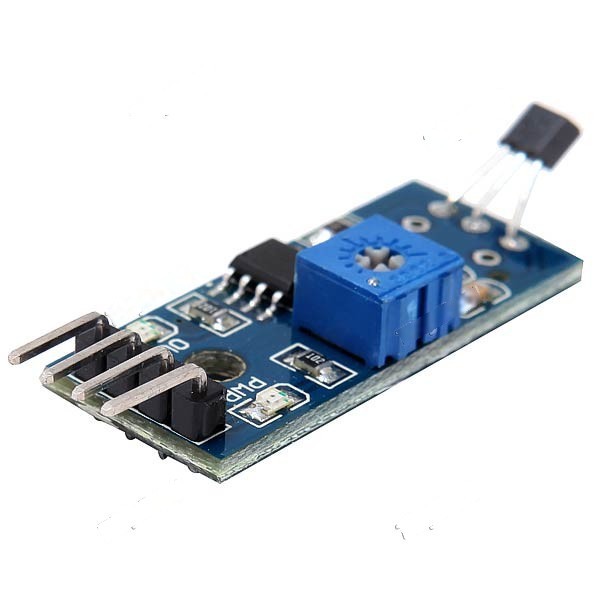


**Hình 1.11.** Động cơ chổi than

Động cơ BLDC cũng giống như động cơ DC chổi than thông thường, gồm các cuộn dây đặt lệch nhau 120 độ trong không gian của stator. Các thanh nam châm được dán chắc chắn vào thân rotor làm nhiệm vụ kích từ cho động cơ.

## 1.9 Cảm biến từ trường HALL

Cảm biến từ trường được ứng dụng để phát hiện từ trường của môi trường xung quanh. Nó được ứng dụng trong bộ đo tốc độ, dùng làm mạch đếm hoặc dùng để phát hiện vị trí của vật,...

****

**Hình 1.12**. Module Cảm biến từ trường HALL

Thông số kỹ thuật:

* Điện áp cung cấp: 5V.
* Đèn led báo tín hiệu đầu ra.
* Sử dụng sensor 3144E và LM393.
* Dạng ngõ ra kiểu số (có thể điều chỉnh độ nhạy bằng biến trở).
* Tín hiệu tích cực ở mức thấp (0V).
* Kích thước: 20 x 32 x 11mm.

## 1.10. Kết luận chương: Sau khi là giới thiệu khái quát các thành phần trong hệ thống, chương tiếp theo sẽ nói về các thuật toán được sử dụng của hệ thống.

# Chương 2: Thiết kế các thuật toán.

## 2.1. Giới thiệu chương: Chương này sẽ giới thiệu các thuật toán được sử dụng trong hệ thống.

## 2.2. Lý thuyết và các thuật toán xử lý ảnh

### 2.2.1 Thư viện OpenCV

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và [máy học](https://techmaster.vn/khoa-hoc/25511/machine-learning-co-ban), và các tính năng tăng tốc GPU trong hoạt động thời gian thực.



**Hình 2.1**. Logo OpenCV

OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.

OpenCV đang được sử dụng rộng rãi trong các ứng dụng bao gồm:

* Hình ảnh street view
* Kiểm tra và giám sát tự động
* Robot và xe hơi tự lái
* Phân tích hình ảnh y tế
* Tìm kiếm và phục hồi hình ảnh/video
* Phim - cấu trúc 3D từ chuyển động
* Nghệ thuật sắp đặt tương tá

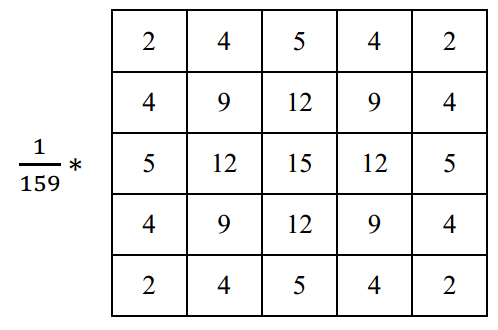
### 2.2.2 Thuật toán Canny

Kỹ thuật phát hiện biên theo phương pháp Canny là phương pháp phát hiện biêntrực tiếp rất hiệu quả áp dụng cho loại ảnh nhiễu.

Phương pháp này do Canny phòng thí nghiệm MIT khởi xướng năm 1986. Canny đã đưa ra tập hợp các mục tiêu của một phương pháp phát hiện biên và đưa ra một phương pháp tối ưu đểthực hiện các mục tiêu đó. Phương pháp này gọi là phương pháp Canny

-Thuật toán được tiến hành qua 4 bước cơ bản:

**B1**: Giảm nhiễu  
Thông thường để giảm nhiễu sử dụng các bộ lọc làm mờ. Có thể sử dụng bộ lọc Gaussian để tích chập với ảnh:

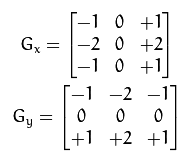


**Hình 2.2**. Bộ lọc Gauss kích thước 5x5

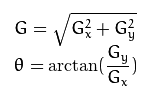
**B2**: Tính độ lớn và góc của Gradient:

Tính đạo hàm Dx ( x,y) và Dy(x,y ) theo chiều x và y của ảnh. Một số bộ lọc như: Roberts, Prewitt, Sobel.

Bộ lọc Sobel 3x3:



Độ lớn Gradient và góc Gradient:



Tính θ’ bằng cách làm tròn từ góc θ vào một trong bốn hướng: 00, 450, 900, 1350.

**B3:** Chặn không cực đại **(**Non-Maximum Surpression**)**  
Bước này chỉ giữ lại những pixel thuộc cạnh mà có độ lớn gradient lớn nhất

Xem xét 3 pixel trong vùng 3 x 3 xung quanh pixel (x,y):

- Nếu θ(x, y) = 00 thì (x+1, y), (x, y) và (x-1, y) được xem xét.

- Nếu θ(x, y) = 900 thì (x, y+1), (x, y) và (x, y-1).

- Nếu θ(x, y) = 450 thì (x+1, y+1), (x, y) và (x-1, y-1).

- Nếu θ(x, y) = 1350 thì (x-1, y+1), (x, y) và (x+1, y-1).

Nếu pixel (x, y) có gradient lớn nhất của 3 pixel xem xét thì pixel đó là cạnh.

**B4:** Ngưỡng Hysteresis (Hysteresis Thresholding)

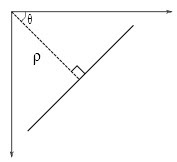
Hysteresis sử dụng 2 ngưỡng, ngưỡng thigh và tlow. Pixel mà có độ lớn gradient D < tlow thì được loại ngay lập tức. Những pixel tlow < D < thigh được giữ lại nếu là một cạnh liên tục với những pixel có độ lớn gradient D > thigh.

****

**Hình 2.3.** Ảnh trước và sau khi dùng thuật toán canny

### 2.2.3 Thuật toán HoughLines

HoughLines để phát hiện đường thẳng, là thuật toán tìm đường thẳng mặc dù có thể bị đứt đoạn trong ảnh. Trong hình học thì đường thẳng có phương trình: y = m.x + c, hoặc là ρ = x.cosθ + y.sinθ (ρ đọc là rho, còn đọc θ là theta nhé), với ρ là khoảng cách từ đường thẳng tới gốc tọa độ, còn θ là góc giữa trục hoành và đoạn thẳng ngắn nhất nối tới gốc tọa độ (đơn vị là radian).

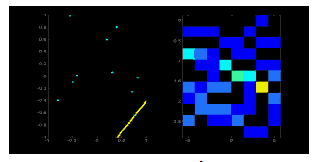


**Hình 2.4**. Hình minh họa trong OpenCV

Có 2 cách để xác định 1 đường thẳng, người ta chọn cách thứ 2 vì chỉ cần 2 tham số (ρ, θ) tiện cho tính toán hơn. Vì θ được tính bằng radian nên có giá trị trong khoảng [0;π] (hay là [0;3.14]). Khi θ bằng 0 hoặc bằng 3.14 thì đường thẳng dựng đứng, còn θ bằng 1.57 (π/2) là nằm ngang.

**MÔ TẢ CÁCH HOẠT ĐỘNG CỦA THUẬT TOÁN**

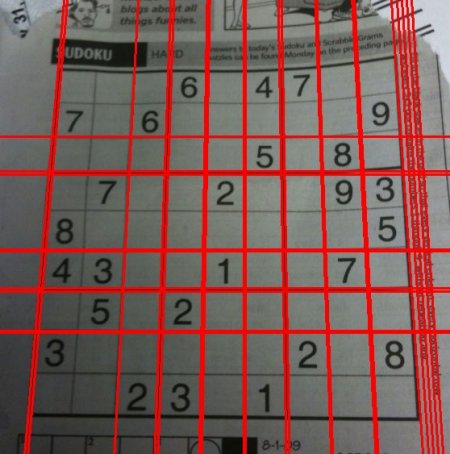
Vì mỗi đường thẳng được xác dịnh bởi 2 giá trị (ρ, θ) nên thuật toán tạo 1 mảng 2 chiều). Dòng ứng với ρ và cột ứng với θ, kích thước của mảng có thể tùy chọn chọn, là mảng càng lớn thì càng chính xác và tính toán càng lâu, còn mảng nhỏ thì nhanh hơn nhưng không chính xác bằng.



**Hình 2.5**. Hình ảnh mô tả hoạt động của thuật toán hough line

Trong Ảnh: bên trái là các điểm trong ảnh, bên phải là mảng 2 chiều

Khi sử dụng người dùng sẽ truyền giá trị ρ và θ họ mong muốn(Thường thì người ta sẽ truyền rho = 1px, theta = 3.14/180 (rad)). Thuật toán sẽ vẽ ρ x θ đường thẳng trên ảnh, với mỗi đường thẳng thuật toán đếm số pixel nằm trên đường thẳng đó, cứ mỗi pixel tìm thấy thì cộng thêm giá trị vào ô ứng với (ρ, θ).

****

**Hình 2.6.** Kết quả thu được sau khi dùng thuật toán HoughLine

### 2.2.4 Thuật toán xác định làn đường.

**Lấy ảnh từ Camera**

**Xác định hướng mà xe cần di chuyển**

**Xác định vạch chỉ làn đường bên trái và phải**

**Xác định các đường thẳng trong khu vực làn đường**

**Loại bỏ các đường thẳng không thể là vạch chỉ làn đường**

**Khởi tạo**

**Thuật toán Xác định các đường thẳng trong khu vực làn đường**

### .

**Từ các biên còn lại xác định các đường thẳng bằng thuật toán HoughLines**

**Xác định các biên bằng thuật toán Canny**

**Loại bỏ các phần bên ngoài khu vực làn đường**

**Chuyển ảnh sang ảnh mức xám**

**Thuật toán Loại bỏ các đường thẳng không thể là vạch chỉ làn đường.**

**Thêm đường thẳng thứ i vào danh sách các đường thẳng có thể là vạch làn đường**

**i++;**

**Góc θ của đường thẳng thứ i > 1100 và < 1500**

**Góc θ của đường thẳng thứ i > 300 và < 700**

**i = 0;**

**i < số đường thẳng còn lại**

**Sai**

**Đúng**

**Sai**

**Sai**

**Đúng**

**Đúng**

**Thuật toán xác định hướng di chuyển của xe.**

**Từ giao điểm(điểm giữa của làn đường) tìm được, tính góc lệch của xe so với làn đường**

**Tìm giao điểm của hai đường thẳng xác định hai vạch chỉ làn đường đã tìm được**

**Các kết quả đã đạt được**

- Xác định được làn đường di chuyển của xe qua việc thu hình ảnh từ Camera và xử lý trên Raspberry PI 3.

- Xác định được hướng mà xe cần di chuyển.

- Truyền dữ liệu (hướng mà xe cần di chuyển) từ Raspberry PI 3 đến Arduino UNO R3 thông qua giao tiếp UART.

- Xác định được tốc độ xe bằng Arduino UNO R3 và cảm biến từ trường Hall.

- Điều khiển được tốc độ xe và hướng di chuyển bằng Arduino UNO R3, module điều khiển động cơ, module Servo và cảm biến từ trường Hall.