# CHƯƠNG 4 THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN DẪN HƯỚNG BẰNG TÍN HIỆU HÌNH ẢNH

## Giới thiệu về giải thuật xử lí ảnh và bộ thư viện xử lí ảnh mã nguồn mở OpenCV

### Giải thuật xử lí ảnh

Hiện tại, cùng với sự phát triển vượt bậc của phần cứng máy tính, xử lí ảnh và đồ họa máy tính phát triển rất mạnh mẽ, có nhiều ứng dụng trong thực tiễn. Xử lí ảnh và đồ họa vì thế cũng có vai trò quan trọng đối với lĩnh vực tương tác người máy, máy học (Machine Learning) và trí tuệ nhân tạo (Artificial Intelligence).

Về cơ bản, quá trình xử lí ảnh xem như là thao tác xử lí ảnh đầu vào sao cho ra kết quả mong muốn, kết quả ở đây có thể là một ảnh “tốt hơn” hoặc là những kết luận (dựa trên những so sánh thực nghiệm).

Hình 4.1 Quá trình xử lí ảnh

Ảnh có thể được xem là tập hợp của những điểm ảnh (pixels), mỗi điểm ảnh được xem là đặc trưng cường độ sáng hoặc dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian, nó có thể xem như hàm n đến P(c1, c2, c3, … cn). Vì thế, ảnh trong quá trình xử lí ảnh có thể xem như ảnh n chiều.

Miêu tả ảnh Đối tượng ảnh Đặc trưng ảnh

Xử lý phân tích ảnh

Dữ liệu ảnh

Số hóa ảnh

Tín hiệu ảnh

Thiết bị cảm biến (Camera, Sensor, Scanner)

Ảnh số

Ảnh liên tục

Khung cảnh đối tượng ảnh quang học

Nén, lưu trữ và truyền ảnh

##### Hình 4.2 Các khối cơ bản trong một qui trình xử lí ảnh

Về cơ bản chúng có các khối xử lí chính sau:

* **Thu nhận**: Bao gồm các thiết bị thu nhận như camera, cảm biến ảnh,.. có nhiệm vụ tiếp nhận hình ảnh hoặc video ngõ vào.
* **Tiền xử lí**: Khối này có nhiệm vụ tiền xử lí ảnh bằng các tác vụ nâng cao chất lượng ảnh như giảm nhiễu, tìm biên, phần vùng màu,…
* **Trích chọn đặc điểm**: Khối này có nhiệm vụ trích chọn các đặc trưng của ảnh đã được tiền xử lí để sử dụng trong Hệ quyết định. Đặc điểm của một ảnh có thể là tọa độ tâm đối tượng, màu sắc, đường nét,..
* **Khối hậu xử lí**: Xử lí các đặc điểm đã trích chọn: đường biên ảnh, tọa độ tâm ảnh, mức xám,…. Các đặc trưng này có thể được lược bớt hoặc chuyển đổi để phù hợp hơn với các kĩ thuật lựa chọn cụ thể trong Hệ quyết định.
* **Khối hệ quyết định và lưu trữ**: Khối này có nhiệm vụ đưa ra những lựa chọn phù hợp với yêu cầu xử lí trên lượng dữ liệu lấy được trong khối lưu trữ. Khối lưu trữ có thể là một vùng nhớ trên RAM, ROM hoặc là một ổ cứng, bộ nhớ nhất định.
* **Khối kết luận**: Đưa ra quyết định dựa vào quyết định của khối quyết định. Khối này có thể được liên kết thêm với các khối ngoại vi hoặc phần cứng khác nhau để phục vụ các nhu cầu thực tế: xử lí dữ liệu, điều khiển robot, quản lí kho bãi, hàng hóa,….

Trên thực tế, để có thể thu gọn và hệ thống hóa các thao tác xử lí ảnh và tận dụng sức mạnh tính toán của phần cứng thì cần có những thuật toán xử lí chi tiết. Các thuật toán này có thể đơn thuần là những thao tác như mở ảnh, mở camera,…hay thậm chí như chuyển màu ảnh từ màu RGB sang màu đơn sắc, khữ nhiễu ảnh răng cưa,….

Thêm vào đó, việc trình bày rõ ràng và giải thích chi tiết các đặc điểm xử lí cũng đòi hỏi khả năng tư duy toán học và mô hình hóa hệ thống cao. Chúng tôi đưa ra 2 giải pháp để nghiên cứu về giải thuật: tự viết giải thuật hoặc sử dụng các bộ thư viện có sẵn. Giải pháp tự viết giải thuật đem đến khả năng tự chủ về tư duy thiết kế hệ thống, khả năng linh hoạt, gọn nhẹ trong các dự án riêng biệt tuy nhiên đòi hỏi tư duy về cấu trúc dữ liệu và giải thuật tốt. Giải pháp sử dụng các bộ thư viện mã nguồn có sẵn sẽ giúp giảm thiểu thời gian nghiên cứu giải thuật, chuẩn hóa các quy trình bởi các kĩ sư phần mềm trên cộng đồng, tuy nhiên nhược điểm cố hữu là các bộ thư viện luôn bao hàm nhiều gói cài đi kèm cho rất nhiều chức năng khác nhau, chúng thường có dung lượng nặng và đòi hỏi nhiều về tài nguyên phần cứng khi xử lí, khiến tốc độ xử lí chậm hơn nếu phần cứng không đủ tốt.

Dưới đây mô tả yêu cầu phần cứng hoạt động cho một hệ thống xử lí ảnh số hoạt động ổn định trên nền tảng nhúng Linux (theo đánh giá trên forum https://forum.opencv.org/ ).

* Kiến trúc xử lí CPU (Central Processing Unit): Intel® CoreTM2 Quad Processor Q9650

Số lõi CPU: 4

Số threads: 4

Xung nhịp: 3 GHz

Tốc độ FSB: 1333 MHz

RAM DDR2: 8GB

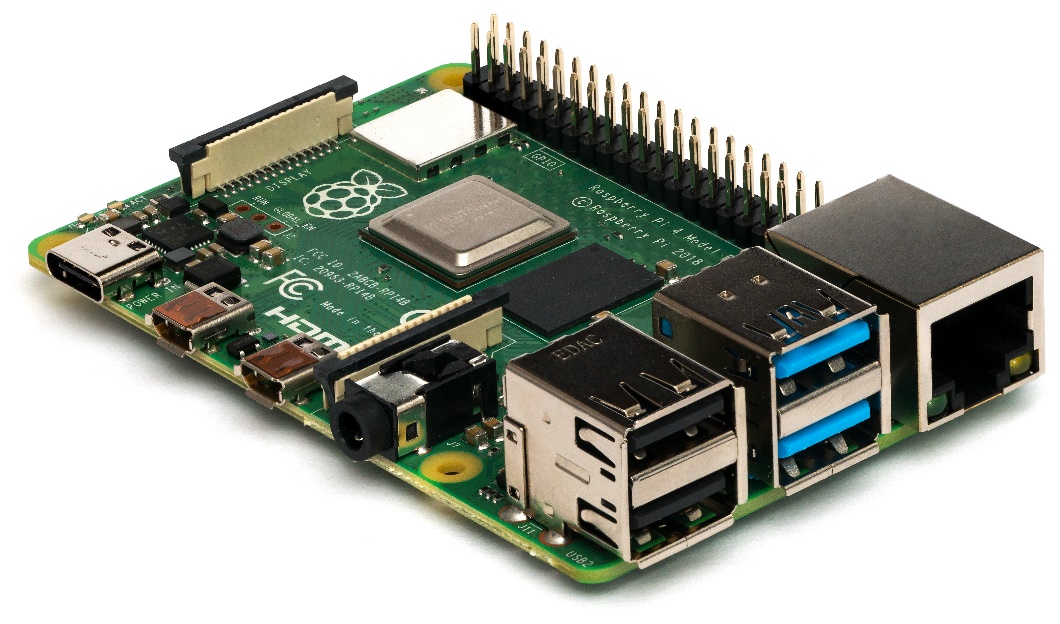
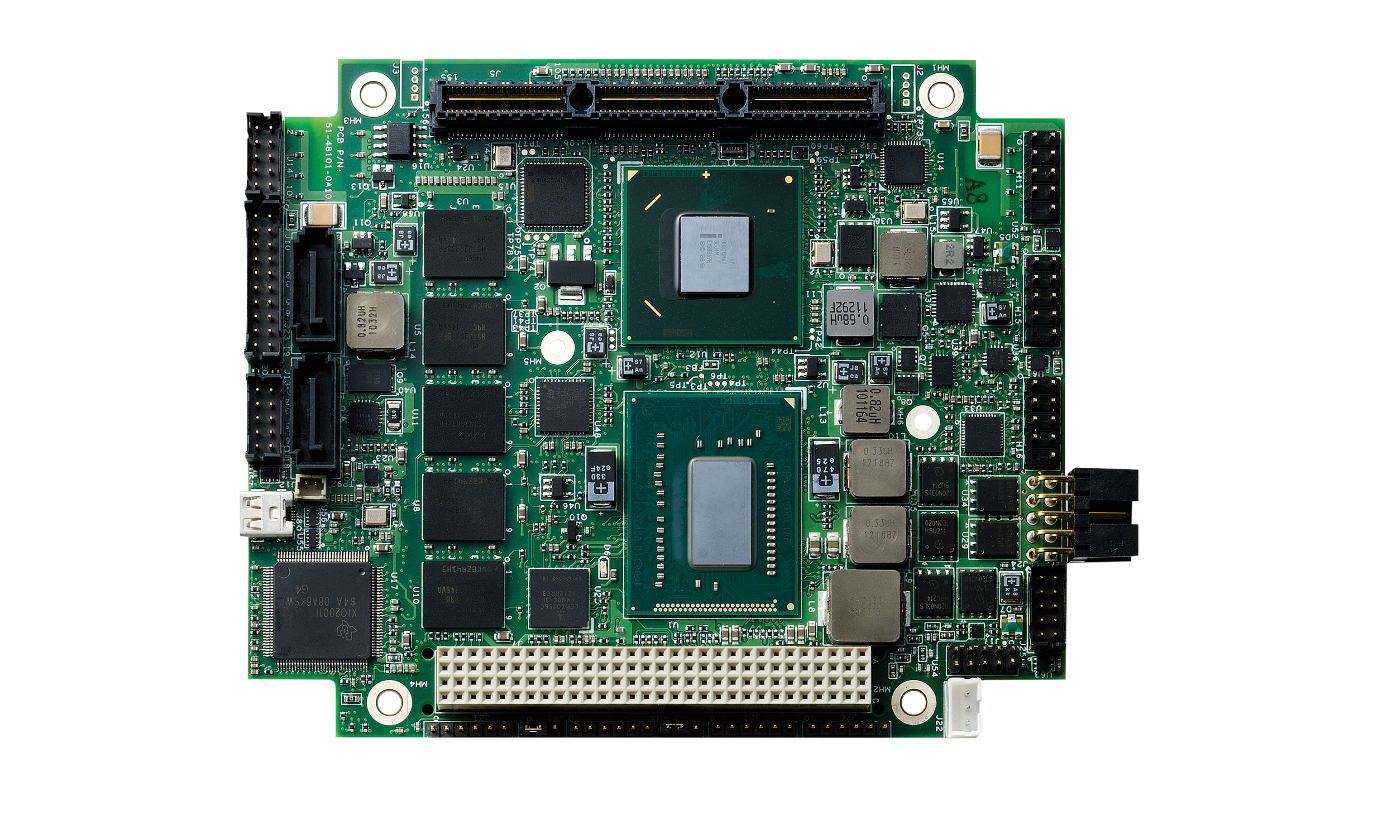
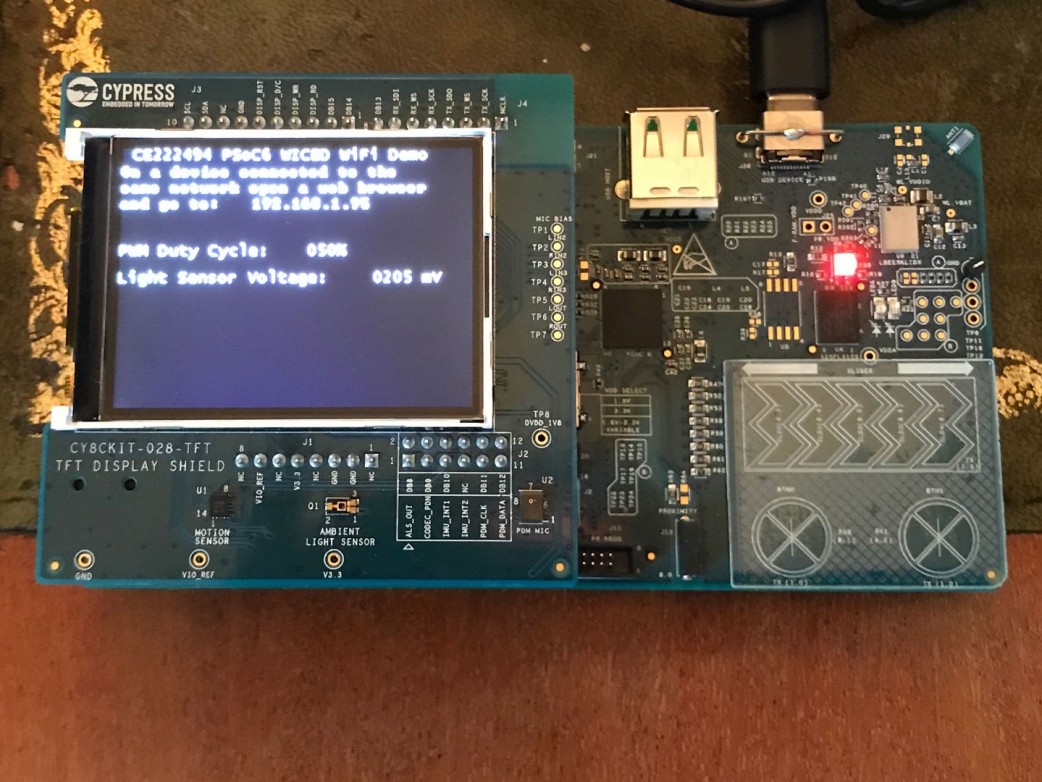
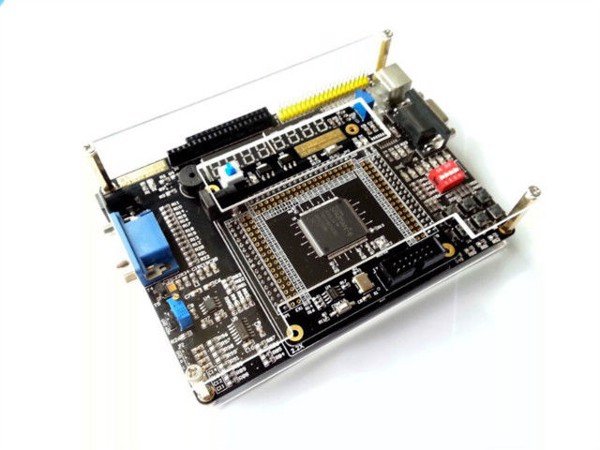
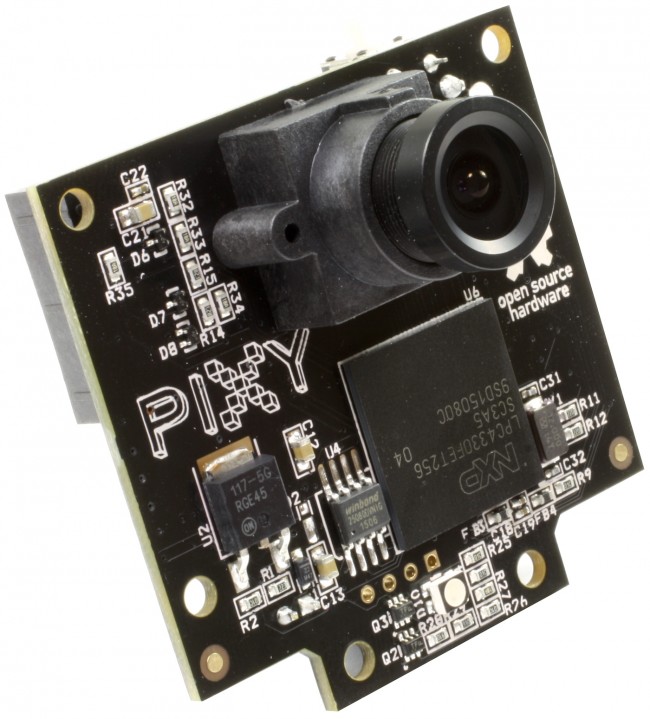
* Kiến trúc đồ họa GPU (Graphics Processing Unit): NVDIA Tesla C1060

Số lõi GPU: 240

Xung nhịp: 1296 GHz

RAM Internal: 4 GB

Yêu cầu phần cứng nêu trên có thể tùy chỉnh được tùy theo phần cứng của các dòng vi mạch chuyên dụng DSP (Digital Signal Processing), mạch vi điều khiển hoặc các máy tính cá nhân Personal Computer (PC). Ngoài nhiệm vụ thu thập dữ liệu, các thiết bị này còn phải xử lí lượng lớn dữ liệu thu thập được bằng các chương trình nhúng trong bộ xử lí hoặc chạy trên PC. Thông thường, các vi điều khiển này là các vộ điều khiển nhúng (Embeeded Controller) vì chúng và các mạch điện tử hỗ trợ được tích hợp ngay bên trong thiết bị. Các dòng thiết bị nhúng thường gặp có thể kể đến PSoC, FPGA, SBC, CMUCAMI1 2 3, DSP,..và các máy tính nhúng. Hiện này các dòng máy tính nhúng cá nhân giá rẻ như Raspberry Pi3, Pi4, NVIDIA Jetson Nano Kit,…thường được ứng dụng hầu hết cả với sự bền bỉ, hiệu năng cao.



(Các mạch nhúng (SBC PsoC, FPGA, CMUCAM, DSP, máy tính công nghiệp PC 104, máy tính nhúng Jetson Nano Kit, Raspberry Pi)).

Hiện nay, các dòng máy tính nhúng cá nhân như Raspberry Pi vừa có thể lập trình và xử lí tín hiệu, đồng thời có khả năng lưu trữ dữ liệu không hề thua kém các máy tính công nghiệp hoặc máy tính cá nhân. Chúng cũng được thiết kế nhỏ gọn và sử dụng nguồn điện năng hợp lí cho các ứng dụng di động như thiết kế, chế tạo robot, thiết bị cầm tay,… Do vậy, trong khuôn khổ đề tài, chúng tôi lựa chọn dòng máy tính nhúng Raspberry Pi để xử lí và có thể can thiệp trực tiếp trong suốt quá trình điều khiển. Như vậy, việc lựa chọn máy tính nhúng là phù hợp với yêu cầu đề tài.

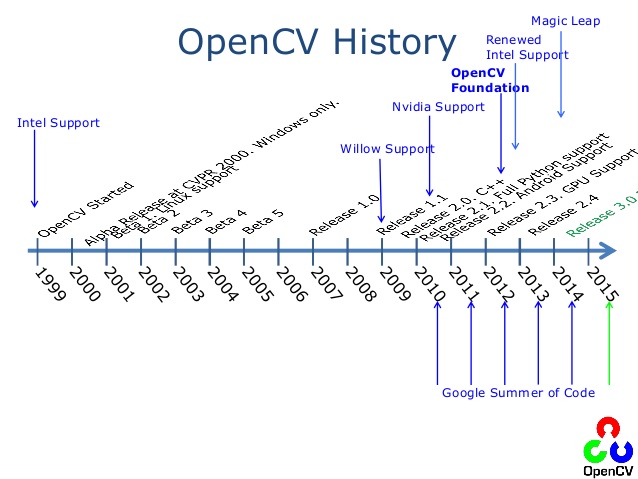
### Bộ thư viện xử lí ảnh mã nguồn mở OpenCV

#### Giới thiệu

OpenCV (Open Source Computer Vision Library) là một thư viện phần mềm máy tính và thị giác máy tính mã nguồn mở1. Chúng được xây dựng để cung cấp một cơ sở hạ tầng chung cho các ứng dụng thị giác máy tính nhằm tăng tốc việc sử dụng nhận thức máy móc trong các sản phẩm thương mại với hơn 500 hàm ứng dụng thị giác máy tính. OpenCV là một sản phẩm được cấp phép BSD1 (Berkelay Software Distribution - giấy phép mã nguồn mở) và GPL1 (General Public License), giúp các doanh nghiệp dễ dàng sử dụng và sửa đổi mã nguồn. OpenCV có trang chủ chính thức tại *https://opencv.org/*, hoặc có thể tải về trên *https://sourceforge.net/*, thư viện này cho phép tải tài liệu và mã nguồn theo giấy phép của BSD và GPL.

(Bộ nhận diện thương hiệu của thư viện mã nguồn mở OpenCV)

Hiện nay, OpenCV bao gồm hơn 2500 thuật toán được tối ưu hóa, trong đó bao gồm một bộ toàn diện của cả thuật toán máy học và thị giác máy tính cổ điển cũng như hiện đại. OpenCV có hơn 47 nghìn người dùng cộng đồng, lượng kĩ sư và cộng đồng phát triển cũng tăng lên không ngừng nghỉ. Bộ thư viện này đã tung ra phiên bản mới nhất là OpenCV 4.5 và OpenCV Contrib 4.5, ước tính hiện tại có hơn 18 triệu lượt tải về tính trên tất cả phiên bản của mã nguồn. OpenCV cũng là một trong số những bộ thư viện an toàn để sử dụng rộng rãi trong các công ty, nhóm nghiên cứu và các cơ quan chính phủ.



(Lịch sử hình thành bộ thư viện OpenCV (theo https://opencv.org/))

Thư viện OpenCV cũng rất phù hợp sử dụng cho các nhà nghiên cứu, sinh viên, học sinh trên toàn thế giới bởi rất dễ tiếp cận từ các hàm xử lí đơn giản cho đến phức tạp. Trong bộ môn học máy (Machine Learning), OpenCV cũng được đưa vào giảng dạy tại hầu hết các trường Đại học tại Việt Nam.

#### Cấu trúc thư viện OpenCV

(Cấu trúc thư viện OpenCV1 (theo LearningOpenCV) – vẽ lại)

**CV (Computer Vision)**: Là bộ phận chứa những thuật toán xử lí hình ảnh và các giải thuật cao cấp liên quan đến thị giác máy tính.

**MLL (Machine Learning Layers)**: Bao gồm thuật toán cho việc học máy, bao gồm nhiều lớp thư viện động và công cụ điều hướng.

**HighGUI (High Graphical User Interface):** Chứa các luồng vào ra I/O cơ bản và các hàm cho việc lưu trữ và truyền tải video, hình ảnh

**CXCore**: Chứa các định nghĩa về cấu trúc dữ liệu cơ sở của cả thư viện. Một vài chức năng về đồ họa cũng được đặt ở bộ phận này.

#### Tại sao lựa chọn thư viện OpenCV

Thị giác máy tính ngày càng được mở rộng và liên tục phát triển. Việc sử dụng những thư viện tiêu chuẩn sẽ tạo ra các ứng dụng mới và các giải pháp xử lí dễ dàng. Các ứng dụng hiện nay cũng thường được cập nhật thường xuyên và thực hiện các thuật toán khó như dò tìm đối tượng, nhận diện khuôn mặt,.. OpenCV cũng được rất nhiều công ty công nghệ lớn trên toàn cầu sử dụng như: Intel, NVIDIA, Microsoft, Siemens, Google,…và các viện nghiên cứu AI như: Stanford, MIT, Cambridge,…

Hiện nay, OpenCV đã và đang góp phần vô cùng lớn trong lĩnh vực thị giác máy tính. Đơn cử tại Việt Nam, bộ phận nghiên cứu FGA.AI (FPT Global Automotive Artificial Intelligence) của FPT Software kết hợp với Yamaha đã và đang triển khai thành công dòng xe tự hành chạy từ cuối năm 2017 trên máy nhúng NVIDIA Jetson có sử dụng thư viện OpenCV tại Khu công nghệ cao TP.HCM, Tòa nhà F-Comlex (Đà Nẵng) và Khu công nghệ cao Hòa Lạc (Hà Nội).

(Xe tự hành an toàn cấp độ 3 (theo thang đo của SAE) triển khai cuối năm 2017 của FPT Software)

Chính vì những lẽ đó, chúng tôi đã lựa chọn sử dụng bộ thư viện mã nguồn mở OpenCV để giải quyết các bước tính toán và rút ngắn thời gian và độ khó trong suốt quá trình xử lí ảnh số, có thể nói, nếu không có sự hỗ trợ từ bộ thư viện OpenCV, chúng tôi sẽ rất khó tiếp cận và hoàn thành được yêu cầu đặt ra của đề tài.

#### Cài đặt OpenCV 3.2 trên hệ điều hành GNU/Linux

OpenCV đều có thể được cài đặt trên rất nhiều nền tảng hệ điều hành (OS – Operation System) khác nhau như Windows, MacOS, các bản phân phối GNU/Linux,.. tuy nhiên để thuận tiện và thống nhất cho ứng dụng, chúng tôi chọn cách cài đặt OpenCV trên hệ điều hành GNU/Linux, cụ thể là bản phân phối Raspbian (Raspberry Pi OS) trên board máy tính nhúng Raspberry Pi 3 và Ubuntu trên máy tính cá nhân.

**a. Cài đặt môi trường và các gói cài cần thiết**

- Mở terminal, tiến hành cài các gói cài sau:

***$ sudo apt install build-essential cmake git pkg-config libgtk-3-dev \ libavcodec-dev libavformat-dev libswscale-dev libv4l-dev \ libxvidcore-dev libx264-dev libjpeg-dev libpng-dev libtiff-dev \ gfortran openexr libatlas-base-dev python3-dev python3-numpy \ libtbb2 libtbb-dev libdc1394-22-dev***

**b. Clone OpenCV và kho đóng góp OpenCV**

- Tạo ra folder **opencv\_build** và đến đường dẫn đó.

***$ mkdir opecv\_build && cd ~/opencv\_build***

Đối với tôi khi làm mình sẽ tạo ra đường dẫn như sau: **/home/covv/workspace/src/opencv\_build**

- Có hai cách lấy gói cài về máy tính: Clone trực tiếp từ github hoặc tải trực tiếp từ trang chủ OpenCV.

+ Clone trực tiếp:

***$ git clone https://github.com/opencv/opencv.git***

***$ git clone https://github.com/opencv/opencv\_contrib.git***

Nếu bạn muốn cài các phiên bản cũ hơn của opencv và opencv\_contrib, bạn có thể dùng lệnh:

***$ git checkout <opencv-version>***

+ Tải trực tiếp từ trang chủ:

Truy cập **https://opencv.org/releases** và tải về các gói cài opencv.

Truy cập **https://github.com/opencv/opencv\_contrib** để tải về gói cài opencv\_contrib.

**c. Sau khi tải về, tạo ra một folder build tạm và chuyển đến folder đó.**

- Copy hoặc di chuyển 2 gói cài opencv và opencv\_contrib vừa tải về vào thư mục opencv\_build vừa tạo ở bước 2.

- Thực hiện dòng lệnh:

***$ cd ~/opencv\_build/opencv***

***$ mkdir build && cd build***

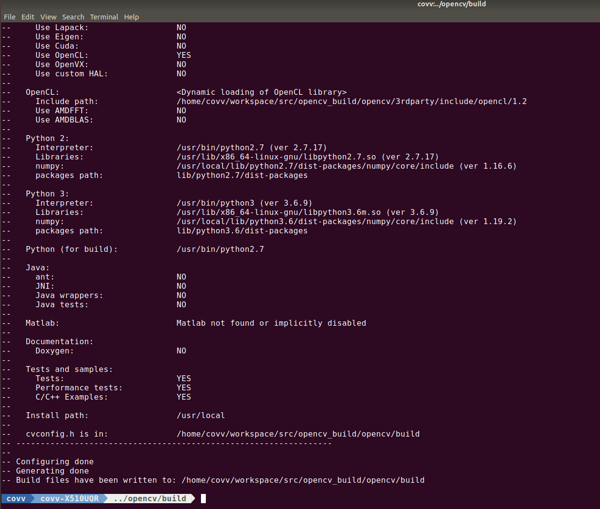
- Setup môi trường build OpenCV với CMake:

**$ cmake -D CMAKE\_BUILD\_TYPE=RELEASE \  
-D CMAKE\_INSTALL\_PREFIX=/usr/local \  
-D INSTALL\_C\_EXAMPLES=ON \  
-D INSTALL\_PYTHON\_EXAMPLES=ON \  
-D OPENCV\_GENERATE\_PKGCONFIG=ON \  
-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_build/opencv\_contrib/modules \  
-D BUILD\_EXAMPLES=ON ..**

Các bạn chú ý đến OPENCV\_EXTRA\_MODULES\_PATH phải là đường dẫn chính xác module của bạn.

Đối với máy tính của mình thì đường dẫn là: **/workspace/src/opencv\_build/opencv\_contrib/modules \**

Sau khi setup môi trường thành công, màn hình terminal thông thường sẽ hiển thị như sau:



**d. Tiến hành biên dịch**

Gõ dòng lệnh:

***$ make -j4*** (có thể là make -j8, -j16,..) tùy thuộc vào số thread CPU máy tính của bạn, số thread càng cao thì biên dịch càng nhanh.

Gõ: ***$ sudo nproc*** để biết được số thread.

**đ. Tải xuống OpenCV**

**$ sudo make install**

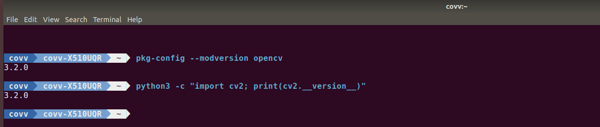
**f. Kiểm tra version đã được cài thành công trên máy tính.**

***$ pkg-config --modversion opencv***

**3.2.0 -->**Đây là bản mình cài cho máy tính của mình

***$ python3 -c "import cv2; print(cv2.\_\_version\_\_)"***

**3.2.0 -->** Đây là bản mình cài cho máy tính của mình



Như vậy về cơ bản chúng ta đã có thể sử dụng bộ thư viện trong các chương trình cơ bản trên máy tính như C/C++, Python, MATLAB mà không bị lỗi.

### Giải thuật xử lí hình ảnh CamShift

#### Cơ sở toán học của thuật toán CamShift

CamShift (Continously Adaptive Meanshift) được kế thừa và phát huy từ thuật toán Meanshift, chúng bao gồm thuật toán cơ sở MeanShift với các thích ứng (Adaptive) bước thay đổi của vùng màu. Trọng tâm là một hàm bước đơn giản nằm trong bản đồ xác suất. Bản đồ xác suất của mỗi điểm ảnh được tính toán dựa trên màu sắc thông qua phương pháp cân bằng Histogram Backprojection. Màu sắc được đưa ra tính toán có thể là các thuộc tính của màu sắc như Hue (Tông màu), Saturation (Độ bão hòa màu), Value (Giá trị màu) từ mô hình màu Hue Saturation Value (HSV).

Ở phương pháp Mean Shift, mỗi một lần lặp thì chỉ đơn giản là giá trị trung bình x và y của bản đồ xác suất đóng góp trong vùng hiện tại. Điều này được xác định bằng cách chia moment thứ nhất của vùng này cho moment thứ không ở mỗi lần lặp và dịch chuyển vùng đến trọng tâm xác suất. Sau khi Mean Shift hội tụ đến một vị trí (x,y), phạm vi của vùng được cập nhật dựa trên giá trị hiện tại của moment thứ 0. Giá trị của bản cập nhật này có thể thay đổi, phạm vi tuyến tính của nó được giả định là tỉ lệ thuận với căn bậc 2 của tổng của các khoản đóng góp xác suất của khu vực hiện tại (ví dụ. Moment thứ không). Chiều rộng và chiều dài tại thời điểm thứ i sẽ được tính thông qua giá trị tại thời điểm thứ i-1, nghĩa là giá trị trước đó.

Trong thuật toán CamShift, một bức ảnh về xác suất phân bố màu sắc của bức ảnh trong chuỗi video được tạo ra. Đầu tiên nó tạo ra một mẫu mô tả Hue sử dụng một biểu đồ màu sắc (color Histogram) và sử dụng không gian màu HSV được tham chiếu từ không gian màu tiêu chuẩn RGB. Vì phân bố màu sắc của các bức ảnh trong video thay đổi theo thời gian, nên thuật toán CamShift đã được sửa đổi để dễ dàng thích ứng với sự thay đổi phân bố xác suất màu sắc của mục tiêu nó theo dõi.

#### Các bước thực thi thuật toán CamShift:

Bước 1: Chọn vị trí ban đầu của cửa sổ tìm kiếm (frame camera)

Bước 2: Thuật toán Meanshift (lặp một hoặc nhiều lần), lưu moment thứ 0.

Bước 3: Kích thước của cửa sổ tìm kiếm được đặt bằng một hàm của moment thứ 0 vừa được tìm trong bước 2.

Bước 4: Lặp lại bước 2 và bước 3 cho đến khi hội tụ (nghĩa là vị trí dịch chuyển nhỏ hơn ngưỡng thiết lập trước đó).

#### Lưu đồ thuật toán Camshift

Đầu tiên là một Histogram được tạo ra, Histogram này chứa các thuộc tính liên quan đến màu sắc. Sau khi tạo ra histogram, tiếp theo tâm và kích cỡ của mục tiêu được tính toán để theo dõi mục tiêu khi hình dạng và kích cỡ của nó thay đổi. Tính xác suất phân bố mục tiêu căn cứ vào Histogram nhận được. Dịch chuyển đến vị trí mới với mỗi khung hình vừa nhận được từ video. Camshift sẽ dịch chuyển đến vị trí mà nó ước lượng trên đối tượng tập trung nhiều điểm sáng nhất trong bức ảnh xác suất. Tìm vị trí mới bắt đầu từ vị trí trước đó và tính toán giá trị trọng tâm vừa tìm được. Cả hai thuật toán trên đều có thư viện các hàm tính toán trong OpenCV.

Đối với các khâu xử lí tính toán trong báo cáo, chúng tôi sử dụng thao tác dịch chỉnh Histogram (dịch chỉnh H, S, V) để thực hiện lấy mẫu thử nhiều lần cho vật chuẩn là màn hình LED sáng từ điện thoại. Quá trình thử và lấy mẫu dựa trên dịch chỉnh các hệ số nhiều lần.

Chương trình xử lý ảnh viết trên Qt Creator có sử dụng thư viện OpenCV của Intel được biên dịch chéo trên máy tính Linux sau đó được truyền xuống board nhúng chạy độc lập là Raspberry Pi B+.

Chuyển đổi không gian màu

Xác suất phân bố màu

CamShift

Tính toán Histogram

Thiết lập tâm và kích thước cửa sổ hiển thị ảnh

X, Y

Thiết lập lại vùng tính toán

##### Hình 4.3 Sơ đồ hệ thống Camshift

### Lưu đồ giải thuật của hệ thống

**a. Chương trình chính**

HOME

CHƯƠNG TRÌNH ĐIỀU KHIỂN TỰ ĐỘNG

CHƯƠNG TRÌNH ĐIỀU KHIỂN BẰNG TAY

##### Hình 4.6 Sơ đồ giải thuật chương trình chính

**b. Chương trình điều khiển tự động**

ĐK ĐỘNG CƠ DC

BẬT MÁY BƠM

XỬ LÝ VỊ TRÍ KHOẢNG CÁCH

THU THẬP HÌNH ẢNH MỤC TIÊU ĐỊNH VỊ

S

XỬ LÝ MÀU SẮC

ĐẠT KHOẢNG CÁCH

S

ĐẠT NGƯỠNG HSV MÀU CHUẨN

Đ

STOP: DỪNG THU HÌNH ẢNH, DỪNG MÁY BƠM

HỒI TIẾP DÒNG

Đ

XỬ LÝ TOẠ ĐỘ LẤY DỘ LỆCH MỤC TIÊU

CHƯƠNG TRÌNH ĐIỀU KHIỂN BẰNG TAY

ĐỘ LỆCH TOẠ ĐỘ MỤC TIÊU

##### Hình 4.7 Sơ đồ giải thuật chương trình điều khiển tự động

**c. Chương trình điều khiển bằng tay**

THU NHẬN TÍN HIỆU TAY CẦM ROMOTE

ĐIỀU KHIỂN ĐỘNG CƠ

##### Hình 4.4 Sơ đồ điều khiển bằng tay

### Tìm khoảng cách mục tiêu từ robot đến mục tiêu

Theo lưu đồ giải thuật chương trình điều khiển đã trình bày ở trên và phương thức hoạt động của robot đã trình bày ở chương số 1. Chúng tôi cần phải tìm ra một phương pháp xác định khoảng cách của đối tượng mục tiêu so với vị trí lắp đặt camera của robot nhằm chuyển phương pháp điều khiển robot từ tay cầm sang chế độ tự động.

Trên thực tế, có rất nhiều phương pháp xác định khoảng cách mục tiêu đã được phát triển,… Dưới đây mô tả các phương pháp xác định khoảng cách phổ biến hiện nay như: sử dụng đồng hồ lazer xác định vị khoảng cách đối tượng, sử dụng cảm biến siêu âm và sử dụng khẩu độ camera.

**Đồng hồ đo khoảng cách lazer**: Nguyên lí cơ bản của các dòng đồng hồ đo lazer là sử dụng 2 đầu thu phát tín hiệu độc lập và tính khoảng cách gián tiếp thông qua đinh lí Pythagore. Đồng hồ đo đạc và xử lí thời gian phản hồi cũng như bước sóng để xác định khoảng cách. Các dạng đồng hồ đo lazer dạng này có giá thành và độ chính xác đa đạng tùy vào nhu cầu sử dụng dân sự như: xây dựng, thể thao,.. hoặc phục vụ cho quân đội.



(đồng hồ đo lazer SNDWAY phục vụ đo đạc cơ bản)

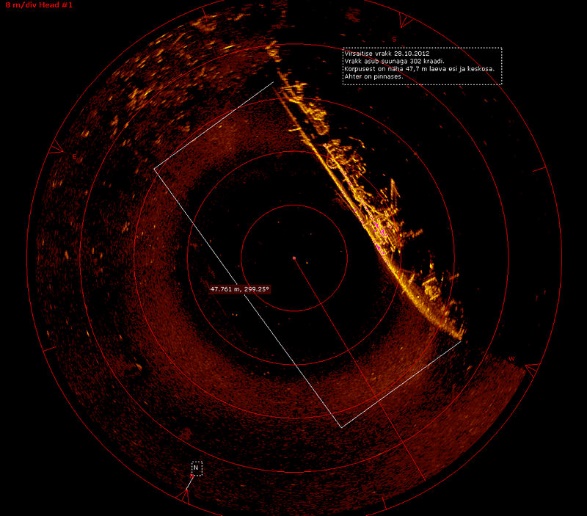


(ống nhòm đo khoảng cách lazer APPROACH Z82 (Garmin) phục vụ chơi Goft )

**Cảm biến siêu âm**: Cảm biến siêu âm cũng có nguyên lí hoạt động tương tự như các dòng đồng hồ lazer. Cảm biến siêu âm hoạt động trên nguyên lí thu thập định thời gian và tốc độ của tín hiệu phản hồi để tính khoảng cách. Cảm biến siêu âm ngày nay ứng dụng phổ biến trong nhiều lĩnh vực dân sự, y tế, hàng hải, hàng không, quân sự,…



(một vài dòng cảm biến siêu âm xác định khoảng cách trong công nghiệp)

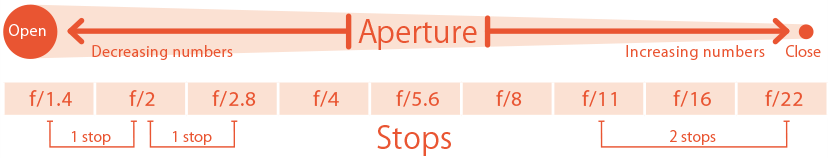


(hình ảnh quan trắc khoảng cách thu được từ sonar siêu âm về tàu ngầm T-297 của Hải quân Liên Xô cũ bị đắm tại đảo Keri, vùng biển Estonia – theo Wikipedia)

**Đo đạc khoảng cách sử dụng tiêu cự và khẩu độ camera**: Tiêu cự camera là lượng ánh sáng đi vào ống kính của camera. Khẩu độ là sự đóng mở ống kính camera. Giá trị tiêu cự được kí hiệu là f. Các số f tiêu chuẩn là f/1.4, f/2, f/2.8, f/4, f/5.6, f/8… (theo https://snapshot.canon-asia.com/vietnam/). Khẩu độ của các camera chuyên nghiệp thông thường sẽ thay đổi được trong phạm vi tùy theo ống kính. Tuy nhiên, các loại camera thông thường phục vụ xử lí ảnh số sẽ không có ống kính gắn kèm khẩu độ và được nhà sản xuất chế tạo theo một thông số cụ thể do đó hiếm khi thay đổi được số f.



(điều chỉnh phạm vi lấy nét ảnh bằng việc điều chỉnh khẩu độ)



(độ mở của ống kính có mối quan hệ mật thiết đến tiêu cự của camera)

Tiêu cự của camera thường được ứng dụng đo đạc khoảng cách thông qua việc xử lí hình ảnh hoặc video streaming thu về bằng các thuật toán hình học. Chúng cũng phổ biến trong lĩnh vực thị giác máy tính.



(tiêu cự camera thường được sử dụng làm thông số tính toán khoảng cách trong học máy (ảnh từ https://www.pyimagesearch.com/))

So sánh các phương pháp xác định khoảng cách đối tượng trong mục tiêu ứng dụng của đề tài:

|  |  |  |
| --- | --- | --- |
| **Phương pháp** | **Ưu điểm** | **Nhược điểm** |
| Sử dụng đồng hồ đo khoảng cách lazer | Giá thành tùy thuộc vào độ chính xác của cảm biến.  Sẵn có, không cần tính toán quá nhiều. | Cần mua thêm thiết bị để trích dẫn dữ liệu, không tận dụng được sức mạnh của camera.  Khó để đọc được dữ liệu hồi tiếp từ đồng hồ do thiết bị đã được lập trình sẵn. |
| Sử dụng cảm biến siêu âm | Giá thành tùy thuộc vào độ chính xác của cam biến. | Phải có phương pháp hoặc thiết bị giải mã dữ liệu thu thập được. |
| Sử dụng tiêu cự camera | Hợp lí, tối ưu do đã sử dụng camera sẵn có.  Dễ dàng đọc và xử lí tín hiệu hồi tiếp. | Độ chính xác phụ thuộc hầu hết vào chất lượng camera và giải thuật tính toán. |

Trên cơ sở phân tích và lựa chọn phương pháp xác định khoảng cách, nhóm chúng tôi quyết định lựa chọn sử dụng phương pháp lấy thông số tiêu cự camera để xác định khoảng cách của đối tượng so với robot. Qua đây, chúng tôi cũng tận dụng luôn camera sẵn có trên robot và máy tính nhúng tích hợp sẵn để xử lí tính toán các giá trị thu được từ camera thông qua các giải thuật.

### Giải thuật tam giác đồng dạng

Giải thuật tìm khoảng cách được lựa chọn là phương pháp đồng dạng tam giác (Triangle Similarity for Object/Marker). Phương pháp này được chúng tôi tham khảo từ các công trình nghiên cứu của Adrian Rosebrock, tiến sĩ ngành khoa học máy tính, Đại học Maryland, Baltimore. Adrian Rosebrock đã ứng dụng giải thuật này trong rất nhiều công trình nghiên cứu và sản phẩm thương mại của mình. Phương pháp này được trình bày chi tiết tại bài viết của ông trên trang: https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/.

Về cơ bản, sự đồng dạng của tam giác có thể xem xét như sau: Chúng ta có điểm đánh dấu của đối tượng mục tiêu với chiều rộng là trên thực tế. Sau đó, chúng ta đặt điểm đánh dấu này cách xa máy ảnh một khoảng . Tiến hành chụp hình ảnh và đo chiều rộng biểu kiến của hình ảnh bằng pixel là . Điều này cho phép chúng ta tìm ra được tiêu cự cảm nhận được của máy ảnh thông qua công thức sau:

Ví dụ, chúng ta có một tấm giấy có tiêu chuẩn là 21.59 27.94 cm (kích thước bề rộng là = 27.94 cm). Khoảng cách thí nghiệm từ tấm giấy tiêu chuẩn đến camera là = 60 cm. Tiến hành đo lường chiều rộng của tấm giấy trên hình ảnh thu được từ camera là = 248 pixels.

Vậy, tiêu cự camera tính toán được là:

Giờ đây, khi ta di chuyển camera tiến gần hơn và xa hơn đối tượng/điểm đánh dấu, chúng ta có thể áp dụng tính chất đồng dạng của tam giác để phát hiện khoảng cách của đối tượng đến camera thông qua công thức:

trong đó: là khoảng cách thực tế thu được từ đối tượng đến camera

Cụ thể, giả sử chúng ta di chuyển camera ra xa tấm giấy khoảng 90 cm và chụp ảnh lại tấm giấy bằng camera. Thông qua việc xử lí, tính toán hình ảnh, chúng ta đo được chiều rộng tấm giấy trên hình ảnh thu được là 170 pixels. Thay các giá trị này vào công thức (4.2), ta được:

So sánh với kích thước thí nghiệm là 90cm, chúng ta nhận thấy sai số đo đạc nằm trong khoảng 2.7%. Khi chúng tôi chụp ảnh trong ví dụ, thước đo đạc của chúng tôi bị sai số cố định 1%. Hơn nữa, các bức ảnh cũng được chụp thông qua một camera cá nhân bình thường và không chính xác 100% cho các tọa độ pixel. Tuy nhiên, nếu chất lượng hình ảnh được cải thiện tốt và khoảng cách thí nghiệm được đo đạc lại một cách chính xác, phương pháp đồng dạng tam giác sẽ đúng 100% trong các thí nghiệm trên.

Với kết quả tính toán như trên, chúng tôi quyết định lựa chọn Công thức (4.2) để áp dụng đo đạc và tính toán khoảng cách thu được từ camera đến đối tượng trong đề tài.

### Máy tính nhúng Raspberry Pi B+ và camera Pi NoIR

#### Máy tính nhúng Raspberry Pi 3 Model B+

Raspberry Pi là máy tính nhúng gía rẻ chỉ từ 35USD, có kích cỡ siêu nhỏ gọn và chạy hệ điều hành Linux. Raspberry Pi có khả năng lập trình được, vốn dĩ phù hợp sử dụng dành cho học sinh, sinh viên nhưng hiện tại đã được sử dụng bởi nhiều đối tượng khác nhau với nhiều mục đích khác nhau từ nghiên cứu, chế tạo sản phẩm thương mại hóa.

Mặc dù có bản chất như một máy tính, Raspberry Pi không thể thay thế hoàn toàn cho hệ thống máy tính để bàn hoặc máy tính xách tay. Chúng cũng không hỗ trợ Windows mà chỉ hỗ trợ các bản phân phối của GNU/Linux. Tuy nhiên, Raspberry Pi hoàn hảo cho những hệ thống điện tử, DIY (Do It Yourseft),…

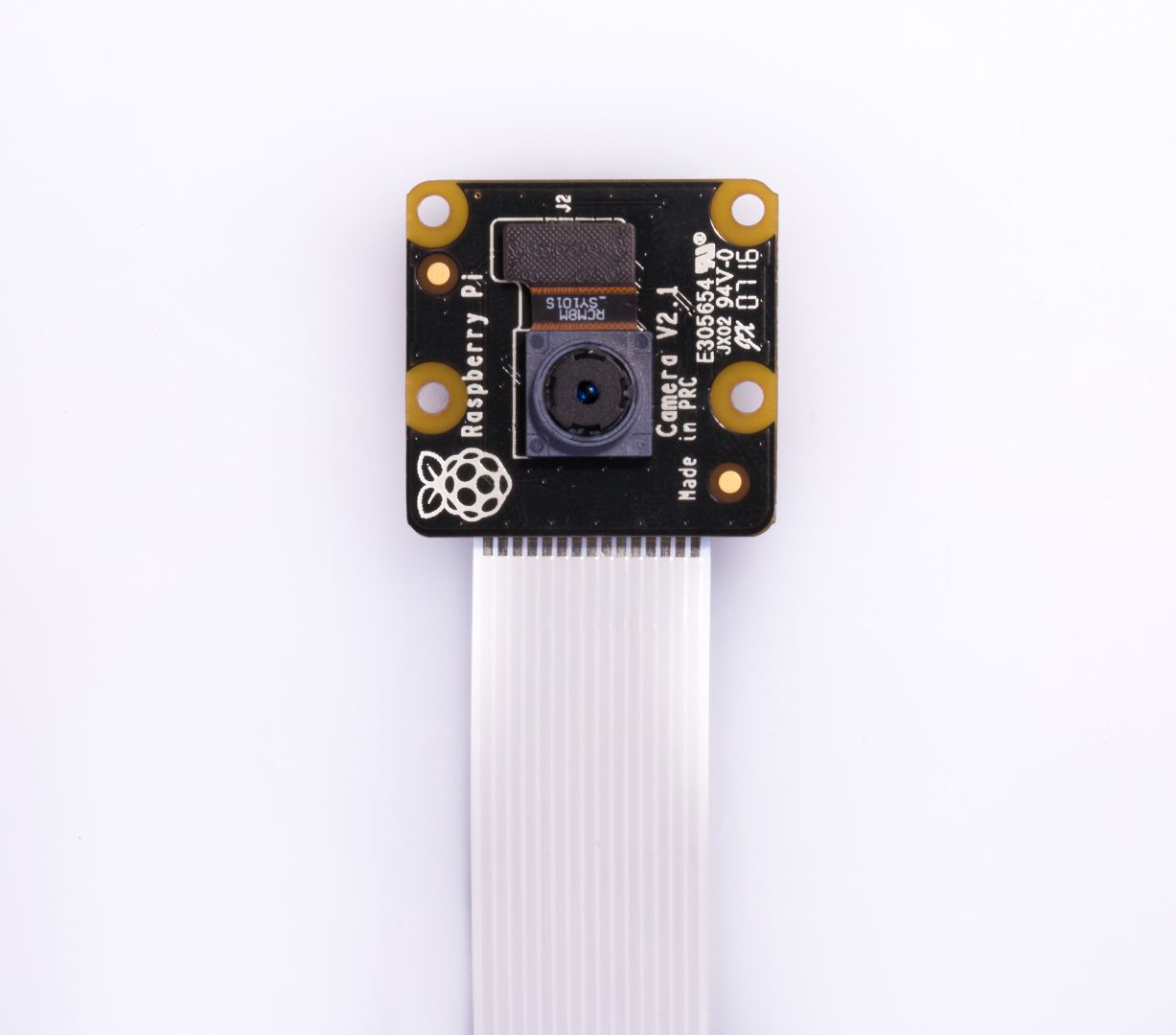
Chính vì những lẽ đó, nhóm chúng tôi sử dụng phiên bản Raspberry Pi 3 Model B+ cho mục đích xử lí tính toán hình ảnh thu thập được từ camera và truyền về board nhúng theo như sơ đồ giải thuật.

**Thông số kĩ thuật chi tiết máy tính nhúng Raspberry Pi 3 Model B+:**

|  |  |
| --- | --- |
| Vi xử lí CPU | Broadcom BCM2837B0 4 nhân A53 (ARMv8) 64-bit xung nhịp 1.4GHz |
| Vi xử lí đồ họa GPU | GPU Broadcom Videocore-IV |
| Bộ nhớ RAM | 1GB LPDDR2 SDRAM |
| Giao tiếp mạng | Gigabit Ethernet (thông qua cổng USB)  Hỗ trợ WiFi 802.11b/g/n/ac băng tần 2.4GHz và 5GHz |
| Giao tiếp bluetooth | Bluetooth 4.2, Bluetooth Low Energy (BLE) |
| Bộ nhớ | Micro SD |
| Giao tiếp ngoại vi GPIO | 40 chân cắm header GPIO |
| Cổng kết nối | HDMI  Jack âm thanh analog 3.5mm  4x USB 2.0  Ethernet  Camera Serial Interface (CSI)  Display Serial Interface (DSI) |
| Kích thước | 82mm x 56mm x 19.5mm, |
| Trọng lượng | 50g |

Mở rộng hơn nữa, với tác vụ tính toán nghiêng về đồ họa nặng cho thư viện OpenCV như hiện nay, có thể nâng cấp lên các phiên bản cao cấp hơn của Raspberry như Raspberry Pi 4 hoặc dùng các dòng máy tính nhúng chuyên dụng đồ họa AI như NVIDIA Jetson Nano Developer Kit để tăng hiệu năng xử lí hình ảnh.

#### Camera Pi NoIR V2 8MP



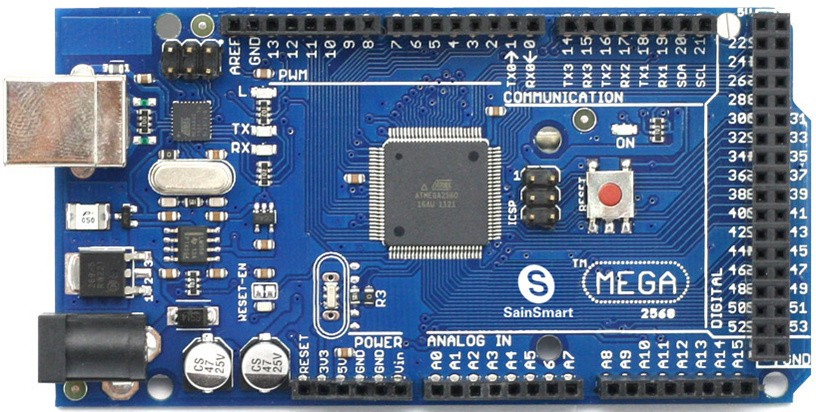
Camera Raspberry Pi NoIR V2 8MP là phiên bản Camera Module dành cho Raspberry Pi mới nhất sử dụng cảm biến ảnh IMX219 8-megapixel từ Sony thay cho cảm biến cũ là OV5647. Với cảm biến IMX219 8-megapixel từ Sony, Camera Module cho Raspberry Pi đã có được sự nâng cấp vượt trội về cả chất lượng hình ảnh, video cũng như độ bền.

Camera Raspberry Pi NoIR V2 8MP có thể sử dụng với Raspberry Pi để chụp hình, quay phim trong điều kiện thiếu sáng với chất lượng HD 1080p30, 720p60 hoặc VGA90, cách sử dụng cũng như lập trình với Camera Module trên Raspberry Pi cũng rất dễ dàng, chỉ cần cắm vào cổng Camera CSI trên Raspberry Pi và cấu hình là có thể sử dụng được. Camera này cũng có rất nhiều bộ thư viện được cộng đồng Raspberry Pi phát triển trên C/C++, Python giúp cho việc tìm hiểu và sử dụng trở nên dễ dàng hơn rất nhiều.

**Thông số kĩ thuật chi tiết camera Raspberry Pi NoIR V2 8MP**

|  |  |
| --- | --- |
| Cảm biến ảnh | IMX219 (Sony) |
| Số điểm ảnh | 8MP |
| Lens | Fixed focus |
| Độ phân giải camera | 3280 x 2464 pixel |
| Độ phân giải video quay được | HD 1080p30, 720p60, 640x480p90 |
| Giao tiếp | Camera Serial Interface (CSI) |
| Chân kết nối | Ribbon |
| Chức năng đặc biệt | Chụp ảnh, quay video thiếu sáng |
| Kích thước | 25mm x 23mm x 9mm |
| Trọng lượng | 3g |

### Board nhúng Arduino Mega 2560



Board nhúng Arduino Mega là board phát triển nhúng đơn giản, phù hợp cho những người mới, học sinh, sinh viên nghiên cứu các sản phẩm ứng dụng cho điện tử, công nghệ máy tính theo hướng đơn giản hóa câu lệnh và phương thức sử dụng. Board chính hãng của Arduino.cc được sản xuất và phân phối từ Italia. Arduino Mega chính hãng có giá thành tuy không rẻ nhưng có độ bền bỉ và khả năng ứng dụng tốt, do đó được rất nhiều người dùng ưa thích. Địa chỉ webstite trang chủ của nhà sản xuất: https://www.arduino.cc/

**Thông số kĩ thuật chi tiết board nhúng Arduino Mega 2560**

|  |  |
| --- | --- |
| Vi điều khiển | ATmega2560 |
| IC nạp và giao tiếp UART | Atmega 16U2 |
| Nguồn cung cấp | 5VDC từ cổng micro USB  5VDC từ jack Barrel |
| Số chân Digital I/O | 54 (trong đó 15 chân có khả năng xuất xung PWM) |
| Số chân Analog Input | 16 |
| Dòng điện DC Current trên mỗi chân I/O | 20mA |
| Dòng điện DC Current chân 3.3V | 50mA |
| Flash Memory | 256 KB |
| SRAM | 8KB |
| EEPROM | 4KB |
| Clock Speed | 16MHz |
| Kích thước | 101.52 x 53.3 mm |
| Khối lượng | 30g |

Ngoài ra, với mức ứng dụng của đề tài, chúng tôi có thể sử dụng các dòng board nhúng cao cấp với các MCU mạnh mẽ hơn như STM32F4 Discovery Board (STMicroelectronics), Tiva Launchpad TI Board (Texas Instruments),…. Tuy nhiên, với độ khó của toàn bộ đề tài, lựa chọn sử dụng board Arduino Mega giúp giảm thiểu gánh nặng về lập trình, tập trung tốt hơn cho giải thuật điều khiển.

### Bộ tay cầm điều khiển từ xa FlySky FS-TH9X

Tay cầm điều khiển từ xa FlySky FS-TH9X được phát triển cho những người có sở thích và đam về điều khiển vô tuyến radio. Bộ điều khiển bao gồm tay cầm (Handler) và đầu thu (Receiver). Sản phẩm có thể được ứng dụng điều khiển hầu hết dạng robot, xe địa hình, máy bay không người lái, thuyền, cano không người lái.



##### Hình 4.9 Tay cầm điều khiển từ xa FlySky FS-TH9X

FS-TH9X có 8 kênh mặc định, tuy nhiên có thể nâng cấp lên 14 kênh bằng mô đun RF (Radio Frequency). Hệ thống này được phát triển đặc biệt cho tất cả các mô hình điều khiển Radio, cung cấp khả năng chống nhiễu siêu chủ động và thụ động, tiêu thụ điện năng rất thấp và độ nhạy thu cao. Với sự kiểm tra nghiêm ngặt của các kỹ sư và nghiên cứu thị trường trong nhiều năm, FLYSKY hiện được coi là một trong những hệ thống tốt nhất hiện có trên thị trường. Địa chỉ trang chủ của nhà sản xuất: https://www.flysky-cn.com/

**Thông số kĩ thuật chi tiết tay cầm điều khiển từ xa FlySky FS-TH9X**

|  |  |
| --- | --- |
| Chế độ nạp mã nguồn | Hỗ trợ nạp mã nguồn mở |
| Tần số sóng RF | 2.4GHz |
| Năng lượng truyền dẫn | < 20dB |
| Khoảng cách thu phát | 500 ~ 1500m (trong không khí) |
| Chuẩn giao tiếp dữ liệu | Phone Jack (PPM)  PWM |
| Số kênh giao tiếp | 8 kênh RF PPM, 2 kênh RF PWM |
| Module RF hỗ trợ | AFHDS/AFHDS 2A |
| Nguồn cung cấp | 8 pin AA |
| Tiêu chuẩn | CE, RCM, FCC ID |

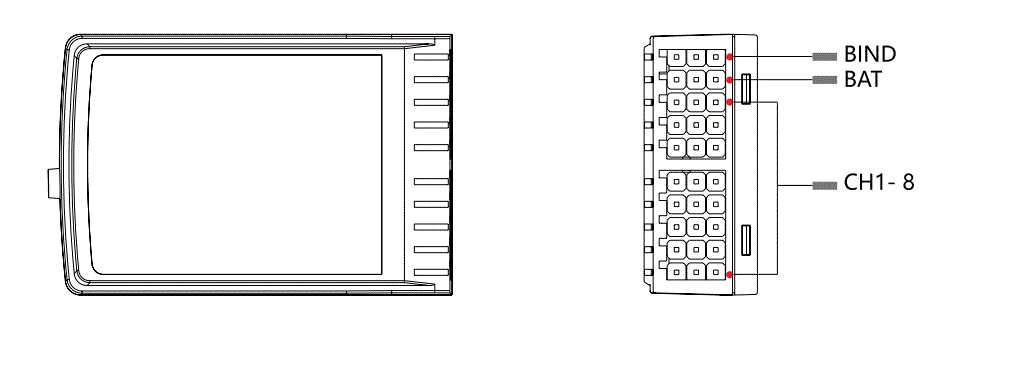
Trong đề tài, chúng em sử dung 3 kênh tín hiệu của tay cầm đề điều khiển các động cơ và bộ phận sau: Động cơ tải chính, Động cơ bẻ lái và động cơ máy bơm.

Động cơ DC giảm tốc

Động cơ DC Servo

Driver mạch cầu BTS7960

Driver mạch cầu BTS7960



Relay 5V

Động cơ máy bơm

(sơ đồ kết nối bộ thu Receiver với mạch điều khiển động cơ)

### Các board mạch và phần cứng khác

Ngoài các board mạch chính sử dụng trong bộ phận xử lí, chúng tôi cũng sử dụng các board mạch và phần cứng khác như mạch cầu H BTS7960, mạch giảm áp 24V – 5V DC, acquy,…

## Sơ đồ kết nối thiết bị

Camera Pi NoIR

Nguồn DC 5V 3A

Raspberry Pi 3 B+

Receiver FS-TH9X

Nguồn DC 5V 2A

Nguồn DC 5V 2A

Mega 2560

Driver mạch cầu H BTS7960

Driver mạch cầu H BTS7960

(Sơ đồ kết nối phần cứng)

Động cơ DC Servo

Động cơ DC giảm tốc

Relay 5V

Động cơ máy bơm

Acquy DC 24V