TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 2**

**XÂY DỰNG HỆ THỐNG TỰ XOAY CAMERA THEO CHUYỂN ĐỘNG CỦA VẬT THỂ**

*Người hướng dẫn*: **TS MAI NGỌC THẮNG**

*Người thực hiện*: **VĂN HỒNG QUÂN - 51303138**

**TRẦN LÊ NHẬT LINH - 51303090**

Lớp **: 13050302**

Khoá  **: ĐH-17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2016**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 2**

**XÂY DỰNG HỆ THỐNG TỰ XOAY CAMERA THEO CHUYỂN ĐỘNG CỦA VẬT THỂ**

*Người hướng dẫn*: **TS MAI NGỌC THẮNG**

*Người thực hiện*: **VĂN HỒNG QUÂN - 51303138**

**TRẦN LÊ NHẬT LINH - 51303090**

Lớp **: 13050302**

Khoá  **: ĐH-17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2016**

LỜI CẢM ƠN

Trong quá trình nghiên cứu và hoàn thành đồ án chúng em đã nhận được rất nhiều sự giúp đỡ ân cần của quý Thầy Cô. Đặc biệt, chúng em xin được gửi lời cảm ơn sâu sắc tới Thầy-TS.Mai Ngọc Thắng, người đã trực tiếp hướng dẫn rất tận tình để chúng em có đủ kiến thức để hoàn thành đồ án này. Một lần nữa chúng em xin chân thành cảm ơn Thầy đã hướng dẫn và truyền lại những kiến thức quý giá để chúng em có thể hoàn thành đồ án này.

Đề tài “Xây dựng hệ thống camera tự xoay theo chuyển động của vật thể” khá là mới đối với chúng em nên không tránh được những sai sót và khiếm khuyết. Chúng em rất mong nhận được ý kiến đóng góp của quý thầy cô và bạn bè để hệ thống ngày càng hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng em và được sự hướng dẫn của TS.Mai Ngọc Thắng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Lê Nhật Linh*

*Văn Hồng Quân*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Đây là tài liệu giới thiệu về hệ thống nhúng, Arduino, OpenCV, cách xây dựng một hệ thống được điều khiển bởi Arduino, cụ thể đó là xây dựng “ Hệ thống tự xoay camera theo chuyển động của vật thể ”.

Xem xong tài liệu chúng ta có thể hiểu rõ hơn về Arduino, cách nhận diện khuôn mặt với sự hỗ trợ của OpenCV, cách điều khiển động cơ Servo bằng Arduino.

Biết cách sử dụng và vận hành và xây dựng hệ thống camera tự xoay theo chuyển động của vật thể, cụ thể ở đây camera chuyển động theo khuôn mặt.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc470087498)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc470087499)

[TÓM TẮT iv](#_Toc470087500)

[MỤC LỤC 1](#_Toc470087501)

[CHƯƠNG 1 – MỞ ĐẦU 5](#_Toc470087502)

[CHƯƠNG 2 – TỔNG QUAN 6](#_Toc470087503)

[2.1 Hệ Thống Nhúng 6](#_Toc470087504)

[2.1.1 Khái niệm hệ thống nhúng 6](#_Toc470087505)

[2.1.2 Kiến trúc hệ thống nhúng 6](#_Toc470087506)

[2.1.3 Các đặc điểm của hệ thống nhúng 7](#_Toc470087507)

[2.2 Giới Thiệu Về Arduino. 8](#_Toc470087508)

[2.2.1 Arduino là gì? 8](#_Toc470087509)

[2.2.2 Khả năng kết nối của Arduino. 9](#_Toc470087510)

[2.2.3 Arduino Uno 10](#_Toc470087511)

[2.2.4 Arduino IDE 11](#_Toc470087512)

[2.3 OpenCV 13](#_Toc470087513)

[2.3.1 OpenCV là gì? 13](#_Toc470087514)

[2.3.2 Các ứng dụng OpenCV 13](#_Toc470087515)

[CHƯƠNG 3 – XÂY DỰNG HỆ THỐNG 14](#_Toc470087516)

[3.1 Yêu cầu phần cứng 14](#_Toc470087517)

[3.2 Yêu cầu phần mềm 14](#_Toc470087518)

[3.3 Sự tương tác giữa OpenCV, Arduino và Servo 15](#_Toc470087519)

[CHƯƠNG 4 – HỆ THỐNG 16](#_Toc470087520)

[4.1 Điều khiển động cơ Servo 16](#_Toc470087521)

[4.1.1 Thiết lập một số thông số cơ bản của servo 16](#_Toc470087522)

[4.1.2 Xác định vị trí của khuôn mặt để điều khiển Servos 18](#_Toc470087523)

[4.1.3 Nạp code vào Arduino 19](#_Toc470087524)

[4.2 Nhận diện khuôn mặt bằng OpenCV 22](#_Toc470087525)

[4.3 Lắp đặt phần cứng 34](#_Toc470087526)

[CHƯƠNG 5: KẾT LUẬN 39](#_Toc470087527)

[5.1 Những vấn đề đã giải quyết được : 39](#_Toc470087528)

[5.2 Những vấn đề chưa giải quyết được và khó khăn gặp phải: 39](#_Toc470087529)

[5.3 Những kiến thức có được khi hoàn thành đồ án: 39](#_Toc470087530)

[CHƯƠNG 6: TÀI LIỆU THAM KHẢO 40](#_Toc470087531)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

**DANH MỤC HÌNH ẢNH**

[Hình 1 : Kiến trúc tổng thể của hệ thống nhúng 6](#_Toc470087532)

[Hình 2: Một số loại arduino 9](#_Toc470087533)

[Hình 4: Giao diện chính của Arduino IDE 11](#_Toc470087534)

[Hình 5: Một số nút lệnh trong Arduino IDE. 12](#_Toc470087535)

[Hình 3 : Sự tương tác giữa OpenCV, Arduino và Servo. 15](#_Toc470087536)

[Hình 6: Thiết lập cổng kết nối COM3 cho Arduino. 19](#_Toc470087537)

[Hình 7 : Chọn Board Ardunio trong Arduino IDE. 20](#_Toc470087538)

[Hình 8 : Tải chương trình lên mạch Arduino. 21](#_Toc470087539)

[Hình 9 : Nhận diện dựa vào đặc trưng Haar-like 22](#_Toc470087540)

[Hình 10: Ví dụ nhận diện ảnh dựa vào đặc trưng Haar-like 23](#_Toc470087541)

[Hình 11: Ví dụ nhận diện ảnh dựa vào đặc trưng Haar-like 23](#_Toc470087542)

[Hình 12 : Nhận diện ảnh dựa vào đặc trung Haar-like 24](#_Toc470087543)

[Hình 13 : Nhận diện khuôn mặt bằng OpenCV. 33](#_Toc470087544)

[Hình 14: Các thiết bị cần có 34](#_Toc470087545)

[Hình 15: Cách nối Arduino vào Breadboard 35](#_Toc470087546)

[Hình 16: Cách nối dây động cơ Servo 36](#_Toc470087547)

[Hình 17: Cách nối dây động cơ Servo vào Breadboard và Arduino 37](#_Toc470087548)

[Hình 18: Hệ thống camera xoay theo chuyển động. 38](#_Toc470087549)

**DANH MỤC BẢNG BIỂU**

[Bảng 1 : Thông số của Arduino Uno (R3) 10](#_Toc470052875)

CHƯƠNG 1 – MỞ ĐẦU

**-Đặt vấn đề:**

Cuộc sống ngày một phát triển, kéo theo sự phát triển nhanh chóng của công nghệ nhằm đáp ứng nhu cầu ngày càng tăng cao của con người. Tự động hóa đã được ứng dụng ở rất nhiều lĩnh vực, mục đích thường rất đa dạng và tùy theo nhu cầu đặc thù của từng lĩnh vực, tuy nhiên điểm chung nhất vẫn là giảm nhân lực, chi phí, thời gian và sai sót. Ngày nay, chúng ta có thể dễ dàng bắt gặp thiết bị camera giúp hỗ trợ quan sát ở khắp mọi nơi: ở trường học, cơ quan, bệnh viện, cửa hàng,… nhưng hầu hết đều là những camera cố định. Và điểm yếu của camera này là bị giới hạn không gian mà nó có thể quan sát, tạo nên những góc chết mà thiết bị không thể đạt đến. Để giảm thiểu tối đa những góc chết này thì cần nhiều camera đặt các ở vị trí khác nhau, nhưng điều này cũng dẫn đến việc tăng lên về chi phí.

Một giải pháp đặt ra cho vấn đề này là xây dựng một hệ thống camera có khả năng tự xoay theo chuyển động của vật thể. Camera này có thể nhận diện được chuyển động của con người hay các vật thể khác và tự động xoay theo chuyển động. Hệ thống này giúp tiết kiệm chi phí vì giảm thiểu được số lượng camera lắp đặt và tính tự động giúp nó hoạt động chính xác và hiệu quả hơn so với camera cố định. Hơn nữa ứng dụng còn có thể phát triển thêm để áp dụng vào các dự án như : Robot đi theo chuyển động, điều khiển các thiết bị trong ngôi nhà thông minh…Đó là một trong những lí do mà chúng ta xây dựng một hệ thống camera có thể tự xoay theo chuyển động.

CHƯƠNG 2 – TỔNG QUAN

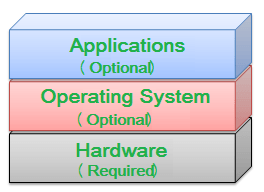
2.1 Hệ Thống Nhúng

2.1.1 Khái niệm hệ thống nhúng**:** là một hệ thống được tích hợp cả phần cứng và phần mềm phục vụ cho các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, gia dụng, y tế, tự động hóa điều khiển .

Hệ thống này đòi hỏi độ ổn định và tự động hóa cao. Do sử dụng cho các nhiệm vụ chuyên biệt và được sản xuất với số lượng lớn nên chúng được thiết kế một cách tối ưu nhằm giảm thiểu kích thước cũng như giá thành sản xuất. Độ phức tạp là khác nhau theo yêu cầu của công việc mà chúng đảm nhận, hệ thống nhúng có thể rất đơn giản với một vi điều khiển hoặc rất phức tạp với nhiều đơn vị, các thiết bị ngoại vi và mạng lưới được nằm gọn trong một lớp vỏ máy lớn.

****2.1.2 Kiến trúc hệ thống nhúng****

Mỗi hệ thống nhúng đều có một kiến trúc thổng thể như sau:



Hình 1 : Kiến trúc tổng thể của hệ thống nhúng

* + 1. Các đặc điểm của hệ thống nhúng

-Yêu cầu chất lượng, ổn định và độ tin cậy cao.

Nhiều loại thiết bị nhúng có những yêu cầu rất cao về chất lượng, tính ổn định và độ tin cậy. Lỗi của hệ thống nhúng có thể gây ra tai nạn khủng khiếp: Như hệ thống điều khiển máy bay, tên lửa, hệ thống điều khiển động cơ ô tô…Vì vậy việc phát triển hệ thống nhúng yêu cầu quy trình kiểm tra, kiểm thử rất cẩn thận.

-Chuyên dụng.

Hệ thống nhúng được thiết kế để thực hiện một chức năng chuyên biệt nào đó. Đây là điểm khác biệt so với các hệ thống máy tính khác như máy tính cá nhân hoặc các siêu máy tính có thể thực hiện nhiều chức năng khác nhau với những phép tính phức tạp. Chuyên dụng giúp nâng cao tính dễ sử dụng và tiết kiệm tài nguyên.

-Có tài nguyên giới hạn

Các hệ thống nhúng bị giới hạn nhiều hơn về phần cứng và chức năng phần mềm so với máy tính cá nhân. Giới hạn phần cứng có thể bao gồm giới hạn về khả năng xử lý, tiêu thụ điện năng, bộ nhớ, chức năng phần cứng,… Còn giới hạn phần mềm thường liên quan đến việc hỗ trợ ít ứng dụng, ứng dụng bị thu gọn tính năng, không có hệ điều hành hoặc hệ điều hành có nhiều hạn chế.

-Một số ví dụ điển hình về hệ thống nhúng

Các thiết bị gia dụng: tủ lạnh, lò vi sóng, lò nướng,…

Các thiết bị kết nối mạng: router, hub, gateway,…

Các thiết bị văn phòng: máy photocopy, máy fax, máy in, máy scan,…

Các thiết bị y tế: máy thẩm thấu, máy điều hòa nhịp tim,…

Các máy trả lời tự động.

Dây chuyền sản xuất tự động trong công nghiệp, robots.

2.2 Giới Thiệu Về Arduino.

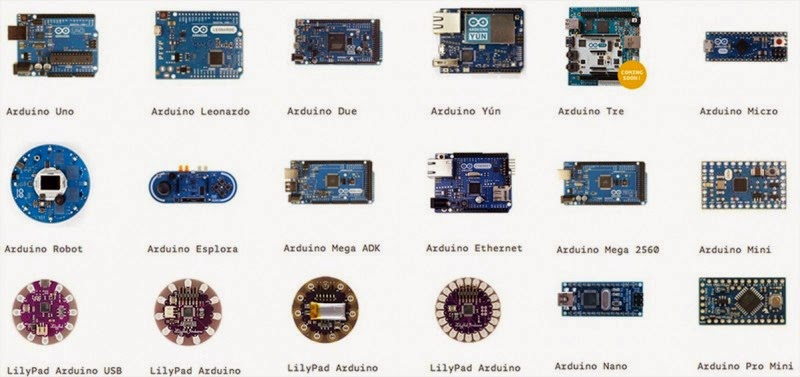
****2.2.1 Arduino là gì?****

**Arduino** : là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.

Trường hợp cụ thể, nếu muốn bật tắt một bóng đèn một cách tự động, đảo hướng một mô tơ một cách dễ dàng, đọc tín hiệu của các cảm biến thì chúng ta có thể sử dụng một mạch Arduino. Hãy nghĩ một cách đơn giản, nếu muốn điều khiển, quản lý, ... thứ gì liên quan đến điện tử thì Arduino có thể giúp chúng ta.

Arduino hiện nay đã được biết đến một cách rộng rãi tại Việt Nam, và trên thế giới thì nó đã quá phổ biến. Sức mạnh của chúng ngày càng được chứng tỏ theo thời gian với vô vàn các ứng dụng mở (open source) độc đáo được chia sẻ rộng rãi.

Với Arduino chúng ta có thể ứng dụng vào những mạch đơn giản như mạch cảm biến ánh sáng bật tắt đèn, mạch điều khiển động cơ,... hoặc cao hơn nữa có thể làm những sản phẩm như: máy in 3D, Robot, khinh khí cầu, máy bay không người lái...



Hình 2: Một số loại arduino

2.2.2 Khả năng kết nối của Arduino.

Một hệ thống Arduino có thể cung cấp cho bạn rất nhiều sự tương tác với môi trường xung quanh :

* Các thiết bị hiển thị (màn hình LCD, đèn LED,…).
* Các module chức năng hỗ trợ kết nối có dây với các thiết bị khác hoặc các kết nối không dây thông dụng (3G, GPRS, Wifi, Bluetooth, 315/433Mhz, 2.4Ghz,…), …
* Hệ thống cảm biến đa dạng về chủng loại (đo đạc nhiệt độ, độ ẩm, gia tốc, vận tốc, cường độ ánh sáng, màu sắc vật thể, lưu lượng nước, phát hiện chuyển động, phát hiện kim loại, khí độc,…),…
* Định vị GPS, nhắn tin SMS,…

2.2.3 Arduino Uno

-Thông số của Arduino Uno (R3).

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5V |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp vào khuyên dùng | 7-12V DC |
| Điện áp vào giới hạn | 5-20V DC |
| Số chân Digital I/O | 14 |
| Số chân Analog | 6 |
| 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 |

Bảng : Thông số của Arduino Uno (R3)

2.2.4 Arduino IDE

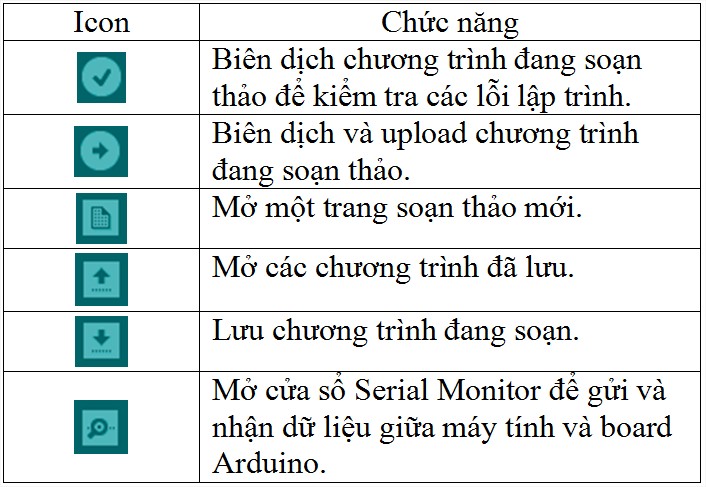
-Arduino Integrated Development Environment: là một trình soạn thảo văn bản, giúp bạn viết code để nạp vào bo mạch arduino.



Hình : Giao diện chính của Arduino IDE

-Vùng lệnh của Arduino IDE

-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:

[](http://k3.arduino.vn/img/2014/05/24/0/465_8121-1400927938-0--table.jpg)

Hình : Một số nút lệnh trong Arduino IDE.

2.3 OpenCV

2.3.1 OpenCV là gì?

**OpenCV** (Open Computer Vision) là một thư viện mã nguồn mở chuyên dùng để xử lý các vấn đề liên quan đến thị giác máy tính. Nhờ một hệ thống các giải thuật chuyên biệt, tối ưu cho việc xử lý thị giác máy tính, vì vậy tính ứng dụng của OpenCV là rất lớn, có thể kể đến như:

* Nhận dạng ảnh: nhận dạng khuôn mặt, các vật thể …
* Xử lý ảnh: khử nhiễu, điều chỉnh độ sáng …
* Nhận dạng cử chỉ.

Ta có thể sử dụng nhiều ngôn ngữ lập trình khác nhau để làm việc với OpenCV như C++, Java, Python, C# ...

2.3.2 Các ứng dụng OpenCV

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

* 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.

CHƯƠNG 3 – XÂY DỰNG HỆ THỐNG

3.1 Yêu cầu phần cứng  
1. Servo Motors x 2  
2. Webcam (bất kỳ loại webcam nào) x 1  
3. Arduino Uno x 1  
4. BreadBoard x 1  
5. Dây nối

( Có thể cấp thêm nguồn bên ngoài 5V cho Arduino để nó hoạt động ổn định hơn )

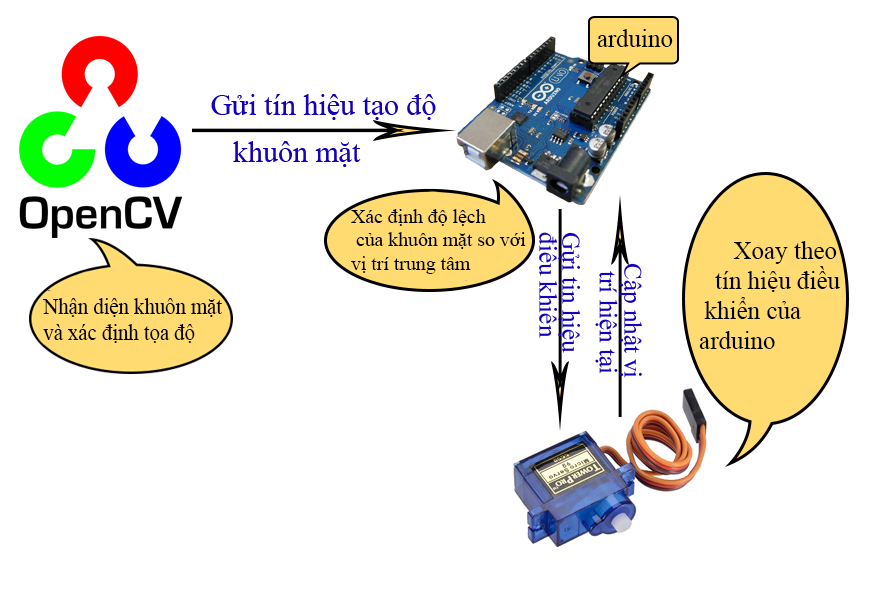
3.2 Yêu cầu phần mềm

1. [Arduino IDE](http://arduino.cc/en/Main/Software) ( Dùng để nạp code cho Arduino)  
2. [OpenCV 3.1](http://sourceforge.net/projects/opencvlibrary/) ( Tích hợp vào Visual Studio)  
3. [Visual Studio 2015](http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express)

3.3 Sự tương tác giữa OpenCV, Arduino và Servo

-Khi OpenCV nhận diện được khuôn mặt thì sẽ gửi thông tin tọa độ về cho Arduino, Arduino sẽ xác định độ lệch của khuôn mặt so với vị trí trung tâm từ đó gửi tín hiệu điều khiển tới động cơ servo. Sau khi động cơ servo nhận được tín hiệu từ Arduino, nó sẽ xoay sau đó cập nhật vị trí lại cho Arduino.

-OpenCV sẽ liên tục gửi tọa độ của khuôn mặt xuống cho Arduino để Arduino xác định độ lệch và điều khiển servo cho tới lúc hình ảnh khuôn mặt nằm ở vị trí trung tâm.



Hình 3 : Sự tương tác giữa OpenCV, Arduino và Servo.

CHƯƠNG 4 – HỆ THỐNG

4.1 Điều khiển động cơ Servo

4.1.1 Thiết lập một số thông số cơ bản của servo

#include <Servo.h>

// Title: Đồ Án 2 - Hệ Thống Camera Xoay Theo Chuyển Động

// Subject: Điều khiển hai động cơ servo để đưa hình ảnh vào trung tâm của màn hình.

// Date: 2016

#define servomaxx 180 // Thiết lập Servo ngang (x) có thể quay được tối đa là 180 độ.

#define servomaxy 180 // Thiết lập Servo dọc (y) có thể quay được tối đa là 180 độ.

#define screenmaxx 320 // Độ phân giải chiều rộng của màn hình là 320

#define screenmaxy 240 // Độ phân giải chiều dọc của màn hình là 240

#define servocenterx 90 // Thiết lập vị trị ban đầu của Servo X là 90

#define servocentery 90 // Thiết lập vị trị ban đầu của Servo Y là 90

#define servopinx 9 // Thiết lập chân điều khiển cho Servo X trong arduino là 9

#define servopiny 10 // Thiết lập chân điều khiển cho Servo Y trong arduino là 10

#define baudrate 57600 // Tốc độ của cổng com. Phải thiết lập trùng với setting với code của xử lý ảnh bên C++.

#define distancex 1 // Bước xoay của Servo X

#define distancey 1 // Bước xoay của Servo Y

void setup() {

Serial.begin(baudrate); // Mở cổng giao tiếp Serial tại baudrate = 57600

Serial.println("Starting Cam-servo Face tracker");

pinMode(servopinx,OUTPUT); // Thông báo đèn trên Arduino cho servo x

pinMode(servopiny,OUTPUT); // Thông báo đèn trên Arduino cho servo y

servoy.attach(servopiny);

//Cài đặt chức năng điều khiển cho chân servopiny(10)

servox.attach(servopinx);

//Cài đặt chức năng điều khiển cho chân servopinx(9)

servox.write(servocenterx);

// Cho servo quay một góc là servocenterx (90 độ)

delay(200);

servoy.write(servocentery);

// Cho servo quay một góc là servocentery (90 độ)

delay(200);

}

4.1.2 Xác định vị trí của khuôn mặt để điều khiển Servos

// Đọc vị trí mới nhất của servo

posx = servox.read();

posy = servoy.read();

//Kiểm tra vị trí của khuôn mặt bị lệch qua bên trái so với vị trí trung tâm.

if(valx < (screenmaxx/2 - incx)){

posx += distancex; //Di chuyển servo sang trái.

}

//Kiểm tra vị trí của khuôn mặt bị lệch qua bên phải so với vị trí trung tâm.

else if(valx > screenmaxx/2 + incx){

posx -=distancex; //Di chuyển servo sang phải.

}

//Kiểm tra vị trí của khuôn mặt bị lệch lên trên so với vị trí trung tâm.

if(valy < (screenmaxy/2 - incy)){

posy += distancey; //Di chuyển servo lên trên.

}

//Kiểm tra vị trí của khuôn mặt bị lệch xuống dưới so với vị trí trung tâm.

else if(valy > (screenmaxy/2 + incy)){

posy -= distancey; //Di chuyển servo xuống dưới.

}

// Servos sẽ xoay cho phù hợp

servox.write(posx);

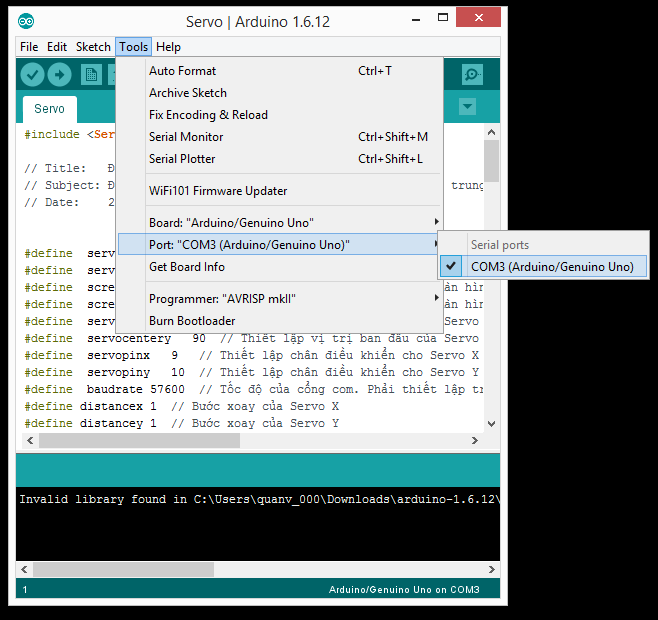
servoy.write(posy); }

4.1.3 Nạp code vào Arduino

-Đầu tiên, phải kiểm tra cổng kết nối giữa máy tính với Arduino.

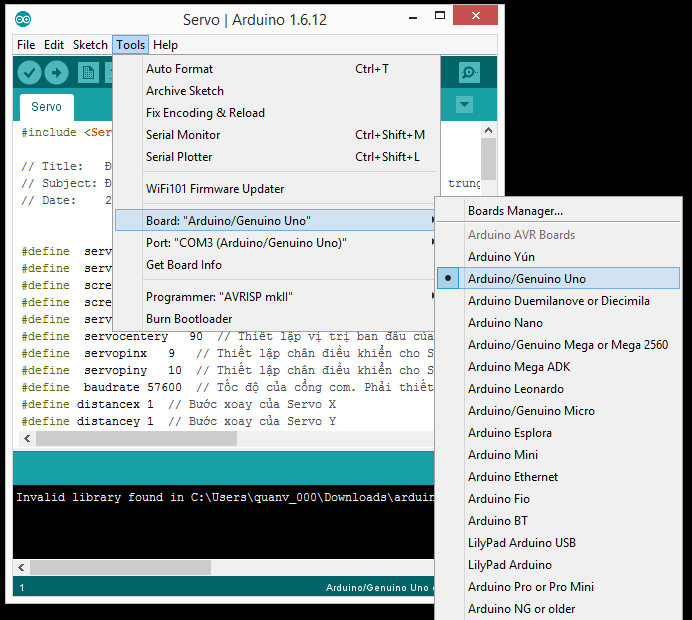
-Khi Arduino kết nối với máy tính, nó sẽ sử dụng một cổng COM (Communication port - cổng dữ liệu ảo) để máy tính và bo mạch có thể truyền tải dữ liệu qua lại thông qua cổng này. Windows có thể quản lí đến 256 cổng COM. Ở đây, mặc định đó là cổng COM3.

-Để kiểm tra, vào Tools 🡪 Port 🡪 Thấy cổng COM3 đã kết nối với Arduino.(Hoặc có thể kiểm tra ở góc phải màn hình của Arduino IDE “Arduino/Genuino Uno on COM3).



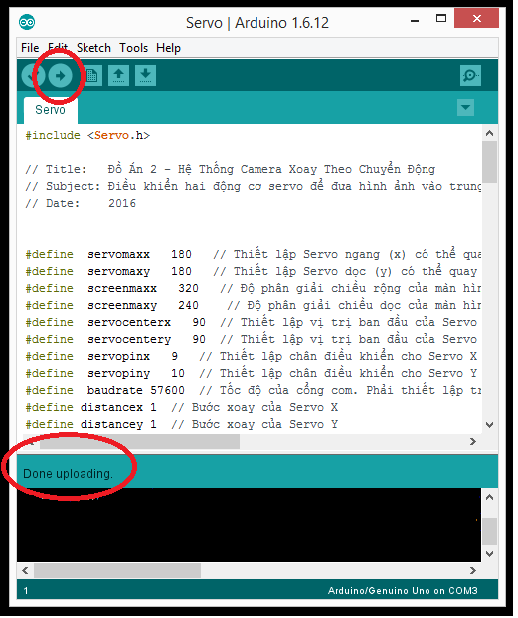
Hình 6: Thiết lập cổng kết nối COM3 cho Arduino.

-Tiếp đó, vào Tools 🡪 Board:”Arduino/Genuino Uno” 🡪 Chọn Arduino/Genuino Uno.



Hình 7 : Chọn Board Ardunio trong Arduino IDE.

-Bước cuối cùng, để tiến hành nạp code cho Arduino, chúng ta nhấn vào Upload (mũi tên ở trên), khi màn hình thông báo hiển thị Done Uploading nghĩa là chúng ta đã nạp code thành công cho Arduino.

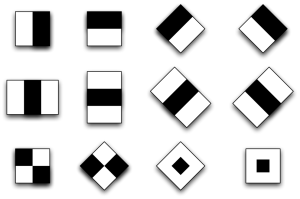


Hình 8 : Tải chương trình lên mạch Arduino.

-Vậy là hoàn thành quá trình nạp code cho Arduino.

4.2 Nhận diện khuôn mặt bằng OpenCV

-Giới thiệu phương pháp nhận diện khuôn mặt dựa vào đặc trưng Haar-like.Các đặc trưng Haar-Like(Haar-Like feature) là những hình chữ nhật được phân thành các vùng khác nhau như hình sau :



Hình 9 : Nhận diện dựa vào đặc trưng Haar-like

-Để nhận dạng khuôn mặt bằng đặc trưng Haar like thì đầu tiên ta phải đưa ảnh về xám hóa:

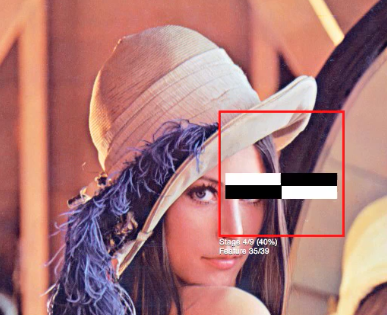
/\*\*Chuyển đổi hình ảnh từ BGR sang định dạng Grayscale,

có nghĩa là chuyển từ ảnh có 3 kênh màu thông dụng Red,Green,Blue

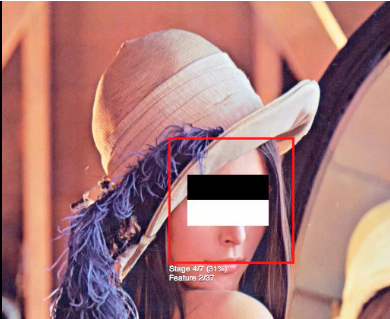
thành ảnh 8 bits Grayscale, dùng hàm: cvtColor(InputArray, OutputArray, int code(code là mã màu)) \*/

cvtColor(frame, frame\_gray, CV\_BGR2GRAY);

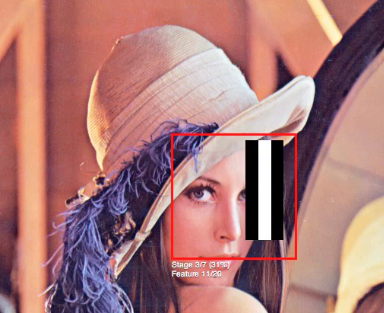
-Sau đó cho các đặc trưng haar-like chạy khắp bức ảnh, những khu vực so sánh được cho là giống với nhiều đặc chưng haar like nhất sẽ được đánh dấu lại. Đây chính là phương pháp cơ bản để nhận dạng khuôn mặt.



Hình 10: Ví dụ nhận diện ảnh dựa vào đặc trưng Haar-like



Hình 11: Ví dụ nhận diện ảnh dựa vào đặc trưng Haar-like



Hình 12 : Nhận diện ảnh dựa vào đặc trung Haar-like

-Hiểu một cách đơn giản: Phương pháp này sẽ chèn đặc trưng Haar-like lên toàn bộ bức hình. Khu vực nào giống thì sẽ nhận diện ở đó là khuôn mặt. Vậy nên có rất nhiều khu vực trong hình nó sẽ nhận là khuôn mặt. Sau đó nó sẽ dùng các ảnh ko giống để loại trừ các vùng này. Cho nên trong quá trình training một bộ nhận dạng của Adaboost, số lượng ảnh sai luôn phải tương đối nhiều để nó loại trừ. Mean shift là thuật toán tracking. Tức là sau khi nó phát hiện được đối tượng nằm đâu nó chỉ theo dõi chuyển động của đối tượng đó. Ở đây là khuôn mặt, sau khi xác định khuôn mặt, nếu khuôn mặt có lắc lư thì nó cũng sẽ theo dõi, chứ ko phải dùng lại thuật toán để nhận dạng.

-Các đặc trưng Haar-like sau khi huấn luyện sẽ được lưu vào file xml. Thư viện OpenCV đã có sẵn bộ huấn luyện nhận dạng khuôn mặt vì thế chúng ta không cần phải xây dựng thêm nữa.

-Để chạy được chương trình này cần copy file haarcascade\_frontalface\_alt.xml vào cùng thư mục với file thực thi hoặc sao chép đường dẫn từ máy tính.

-Dưới đây là toàn bộ code nhận diện khuôn mặt.

-Khai báo thư viện.

#include "opencv2/objdetect/objdetect.hpp"

#include "opencv2/highgui/highgui.hpp"

#include "opencv2/imgproc/imgproc.hpp"

#include <iostream>

#include "Tserial.h"

using namespace std;

using namespace cv;

/\*\* Hàm detectAndDisplay là hàm dùng để nhận dạng tọa độ của khuôn mặt và hiển thị ra màn hình\*/

/\*\* Function Headers \*/

void detectAndDisplay(Mat frame);

/\*\* Global variables (Các biến toàn cục) \*/

/\*\* Sao chép đường dẫn file haarcascade\_frontalface\_alt.xml từ opencv/data/haarscascades trên máy tính của bạn, \*/

//-- Note, either copy these files from opencv/data/haarscascades to your current folder, or change these locations

String face\_cascade\_name = "E:\\Arduino IDE\\openvc3\\opencv\\sources\\data\\haarcascades\\haarcascade\_frontalface\_alt.xml";

-Khai báo biến.

/\*\*Sử dụng lớp CascadeClassifier để phát hiện các đối tượng trong một luồng video. Chúng ta sẽ sử dụng hàm

detectMultiScale để tải hình ảnh lên và thực hiện việc phát hiện\*/

CascadeClassifier face\_cascade;

//Name of Window (tên của cửa sổ camera)

string window\_name = "Capture - Face detection";

// Serial to Arduino global declarations (khai báo cổng nối tiếp toàn cục của Arduino)

int arduino\_command;

Tserial \*arduino\_com;

unsigned char MSB = 0;

unsigned char LSB = 0;

short MSBLSB = 0;

-Qua 2 đoạn code trên chúng ta đã khai báo các biến toàn cục cần thiết để chạy chương trình.

-Sau đây sẽ là hàm quan trọng nhất, hàm xử lý nhận diện khuôn mặt.

void detectAndDisplay(Mat frame)

{

std::vector<Rect> faces;//Khai báo 1 mảng để lưu các giá trị tọa độ x và y

Mat frame\_gray; //Đọc vào đối tượng Mat của OpenCV

/\*\*Chuyển đổi hình ảnh từ BGR sang định dạng Grayscale,

có nghĩa là chuyển từ ảnh có 3 kênh màu thông dụng Red,Green,Blue

thành ảnh 8 bits Grayscale, dùng hàm: cvtColor(InputArray, OutputArray, int code(code là mã màu)) \*/

cvtColor(frame, frame\_gray, CV\_BGR2GRAY);

/\*\*cân bằng biểu đồ của một hình ảnh màu xám, đầu vào là 1 ảnh

màu xám 8 bit. Chức năng của nó là sử dụng các thuật toán để

bình thường hóa độ sáng và tăng độ tương phản của hình ảnh\*/

equalizeHist(frame\_gray, frame\_gray);

//-- Detect faces (nhận dạng khuôn mặt)

/\*\* hàm detectMultiScale: Phát hiện các đối tượng kích thước

khác nhau trong hình ảnh nhập vào. Phát hiện các đối tượng được

trả về như là một danh sách các hình chữ nhật.\*/

face\_cascade.detectMultiScale(frame\_gray, faces, 1.1, 3, 0 | CV\_HAAR\_SCALE\_IMAGE, Size(30, 30));

for (int i = 0; i < faces.size(); i++)

{

//hàm Point cung cấp tọa đô trung tâm x,y của hình cung

/\*Point center;

center.x = faces[i].x + (int)faces[i].width / 2;

center.y = faces[i].y + (int)faces[i].height / 2;\*/

Point center(faces[i].x + (int)faces[i].width\*0.5, faces[i].y + (int)faces[i].height\*0.5);

//hàm vẽ 1 vòng cung đơn giản đặt ngay chỗ khuôn mặt của bạn

ellipse(frame, center, Size(faces[i].width\*0.5, faces[i].height\*0.5), 0, 0, 360, Scalar(155, 100, 65), 2, 8, 0);

cout << "X:" << faces[i].x << " y:" << faces[i].y << endl;

// send X,Y of face center to serial com port (gửi x,y của trung tâm khuôn mặt tới cổng nối)

// send X axis (gửi trục x)

// read least significant byte (đọc những byte gần nhất)

LSB = faces[i].x & 0xff;

// read next significant byte (đọc những byte tiếp theo)

MSB = (faces[i].x >> 8) & 0xff;

arduino\_com->sendChar(MSB);

arduino\_com->sendChar(LSB);

// Send Y axis (gửi trục y) nó cũng hoạt động tương tự trục x

LSB = faces[i].y & 0xff;// he 16 (hex)

MSB = (faces[i].y >> 8) & 0xff;

arduino\_com->sendChar(MSB);

arduino\_com->sendChar(LSB);

// serial com port send

Mat faceROI = frame\_gray(faces[i]);

}

//hiển thị ảnh

imshow(window\_name, frame);

}

**-Giải thích:**

-HàmctvColor:

std::vector<Rect> faces;//Khai báo 1 mảng để lưu các giá trị tọa độ x và y

Mat frame\_gray; //Đọc vào đối tượng Mat của OpenCV

/\*\*Chuyển đổi hình ảnh từ BGR sang định dạng Grayscale,

có nghĩa là chuyển từ ảnh có 3 kênh màu thông dụng Red,Green,Blue

thành ảnh 8 bits Grayscale, dùng hàm: cvtColor(InputArray, OutputArray, int code(code là mã màu)) \*/

cvtColor(frame, frame\_gray, CV\_BGR2GRAY);

-Ở đoạn code trên chúng ta đã chuyển đổi hình ảnh từ BGR sang định dạng ảnh xám để áp dụng thuật toán trainning. Dưới đây sẽ là code nhận diện khuôn mặt:

-Hàm dectectMultiScale:

//-- Detect faces (nhận dạng khuôn mặt)

/\*\* hàm detectMultiScale: Phát hiện các đối tượng kích thước

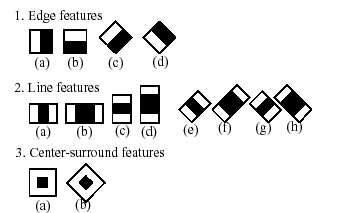
khác nhau trong hình ảnh nhập vào. Phát hiện các đối tượng được

trả về như là một danh sách các hình chữ nhật.\*/

face\_cascade.detectMultiScale(frame\_gray, faces, 1.1, 3, 0 | CV\_HAAR\_SCALE\_IMAGE, Size(30, 30));

-Chức năng của hàm **detectMultiScale**: Phát hiện các đối tượng kích thước khác nhau trong hình ảnh nhập vào. Phát hiện các đối tượng được trả về như là một danh sách các hình chữ nhật.

* frame\_gray: ma trận thuộc kiểu CV\_8U có chứa một hình ảnh của các đối tượng được phát hiện, đọc đối tượng từ camera
* faces: Vector mảng của các hình chữ nhật mà mỗi hình chữ nhật có chứa các đối tượng được phát hiện.



* 1.1: Các yếu tố mà cửa sổ tìm kiếm được chia tỷ lệ giữa các lần quét tiếp theo, 1,1 có nghĩa là cửa sổ ngày càng tăng 10%
* 3: Số lượng tối thiểu (trừ 1) hình chữ nhật gần đó để duy trì đối tượng. Nếu giá trị là 0, hàm trả về tất cả các hình chữ nhật được phát hiện, nó sẽ chồng đề lên nhau.
* 0: thông số với cùng 1 ý nghĩa tương tự cho một cascade cũ trong hàm cvHaarDetectObjects. Nó không được sử dụng cho một cascade mới.
* CV\_HAAR\_SCALE\_IMAGE: là phương thức tiện lợi hơn so với DMA (direct memory access / truy cập bộ nhớ trực tiếp) . Mặc định phương thức (CV\_HAAR\_DO\_CANNY\_PRUNING) sử dụng thuật toán để mở rộng hình ảnh hơn cả 1 máy dò (phân tích).
* Size(Width, Heigh): Tham số cuối cùng cv::Size(width, height) chính là kích thước đầu vào của các đặc trưng Haar-Like, các ảnh gần các bạn chỉnh tham số này lớn lên, và với những ảnh kích thước bé, khuôn mặt trong ảnh bé các bạn nên chỉnh kích thước này nhỏ lại., số càng cao thì tốc độ load ảnh càng mượt nhưng khả năng nhân diện những khuôn mặt ở xa rất thấp, số càng thấp thì khả năng nhận diễn tốt nhưng tốc độ load ảnh chậm.

for (int i = 0; i < faces.size(); i++)

{

//hàm Point cung cấp tọa đô trung tâm x,y của hình cung

/\*Point center;

center.x = faces[i].x + (int)faces[i].width / 2;

center.y = faces[i].y + (int)faces[i].height / 2;\*/

Point center(faces[i].x + (int)faces[i].width\*0.5, faces[i].y + (int)faces[i].height\*0.5);

//hàm vẽ 1 vòng cung đơn giản đặt ngay chỗ khuôn mặt của bạn

ellipse(frame, center, Size(faces[i].width\*0.5, faces[i].height\*0.5), 0, 0, 360, Scalar(155, 100, 65), 2, 8, 0);

cout << "X:" << faces[i].x << " y:" << faces[i].y << endl;

-Ta dùng 1 vòng lặp để duyệt qua từng hình chữ nhật có trong mảng của các hình chữ nhật mà mỗi hình chữ nhật có chứa các đối tượng được phát hiện.

Giải thích các tham số truyền vào của hàm **ellipse**:

* frame: hình ảnh từ camera
* center: tọa độ trung tâm của hình elip
* Size: một nữa kích thước của các trục chính trên hình elip
* 0: góc quay của hình elip đơn vị là độ
* 0: góc quay bắt đầu của vòng cung elip đơn vị là độ
* 360: Góc kết thúc của vòng cung elip đơn vị là độ
* Scala (R, G, B): màu của hình elip bao gồm 3 tham số RED, GREEN, BLUE
* 2: Độ dày của viền hình elip
* 8: Ranh giới của hình elip, thường là 8
* 0: Số phân đoạn bit ở tọa độ của các trung tâm và các giá trị của trục.

-Thuật toán xử lý lấy tọa độ trung tâm của khuôn mặt:

// send X,Y of face center to serial com port (gửi x,y của trung tâm khuôn mặt tới cổng nối)

// send X axis (gửi trục x)

// read least significant byte (đọc những byte gần nhất)

LSB = faces[i].x & 0xff;

// read next significant byte (đọc những byte tiếp theo)

MSB = (faces[i].x >> 8) & 0xff;

arduino\_com->sendChar(MSB);

arduino\_com->sendChar(LSB);

// Send Y axis (gửi trục y) nó cũng hoạt động tương tự trục x

LSB = faces[i].y & 0xff;// he 16 (hex)

MSB = (faces[i].y >> 8) & 0xff;

arduino\_com->sendChar(MSB);

arduino\_com->sendChar(LSB);

// serial com port send

Mat faceROI = frame\_gray(faces[i]);

-Vòng for sẽ thực hiện các bước lặp để lấy ra được tọa độ x và y sau đó gửi qua cho Arduino để xử lý.

-Cuối cùng là ta cho nó hiển thị ảnh từ camera ra màn hình window:

//hiển thị ảnh

imshow(window\_name, frame);

-Bên trên là hàm xử lý nhận diện khuôn mặt, còn đây là hàm **main** để chạy chương trình:

int main(int argc, const char\*\* argv)

{

VideoCapture capture;//Khai báo lớp để lấy ảnh từ các tập tin video, chuỗi mô phỏng hình ảnh hoặc camera

Mat frame;//Đọc vào đối tượng Mat của OpenCV

arduino\_com = new Tserial();// khởi tạo lớp Tserial

if (arduino\_com != 0) {

//sử dụng hàm connect từ lớp Tserial đã khởi tạo

arduino\_com->connect("COM3", 57600, spNONE);

}

//// serial to Arduino setup (thiết lập kết nối tới Arduino)

//-- 1. Load the cascades (Nạp ảnh)

if (!face\_cascade.load(face\_cascade\_name)) { printf("--(!)Error loading face cascade\n"); return -1; };

//-- 2. Read the video stream (Đọc các luồng video)

capture.open(0);

if (capture.isOpened())

{

for (;;)

{

capture >> frame;//đọc các mô phỏng hình ảnh đối tượng từ camera

if (!frame.empty())

{

detectAndDisplay(frame);

}

if (waitKey(30) >= 0) // Pause key

break;

}

}

// Serial to Arduino - shutdown (ngắt kết nối với Arduino)

arduino\_com->disconnect();

delete arduino\_com;

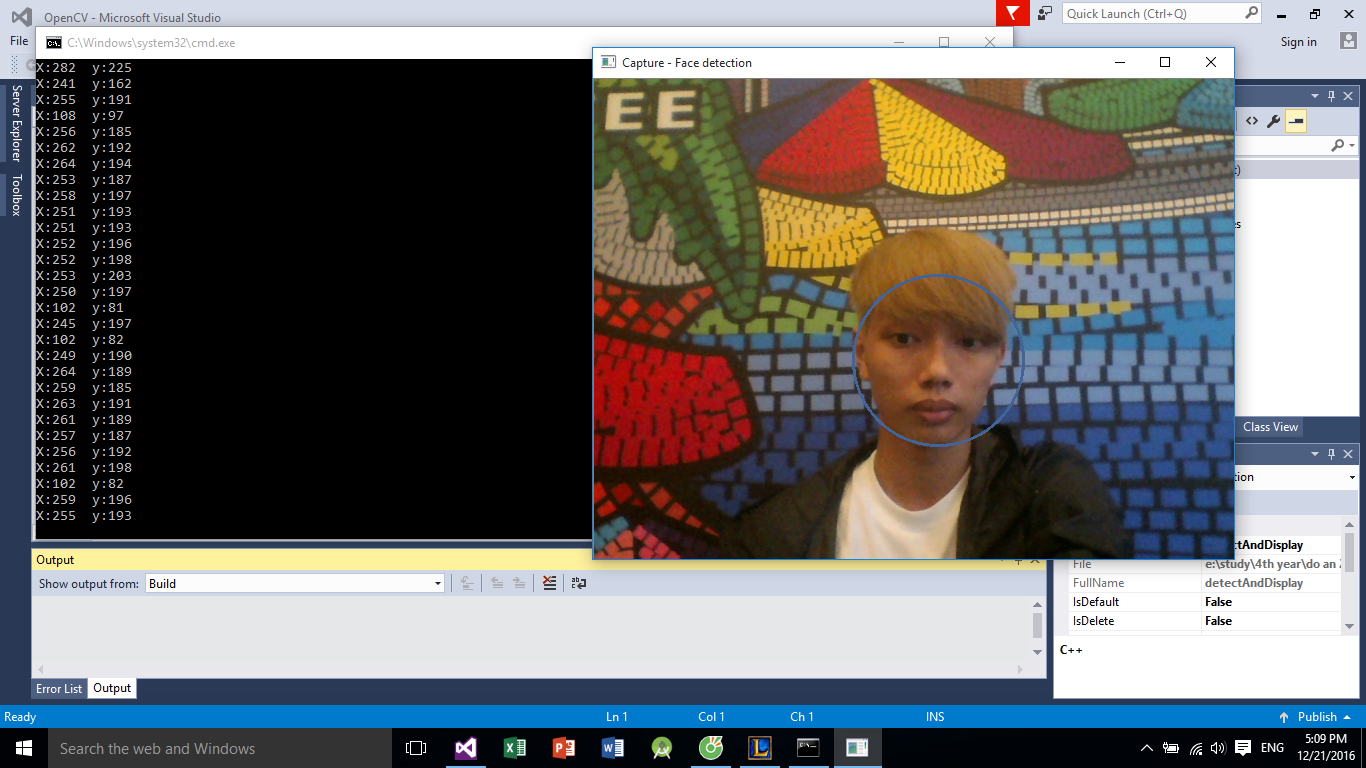
arduino\_com = 0;

return 0;

}

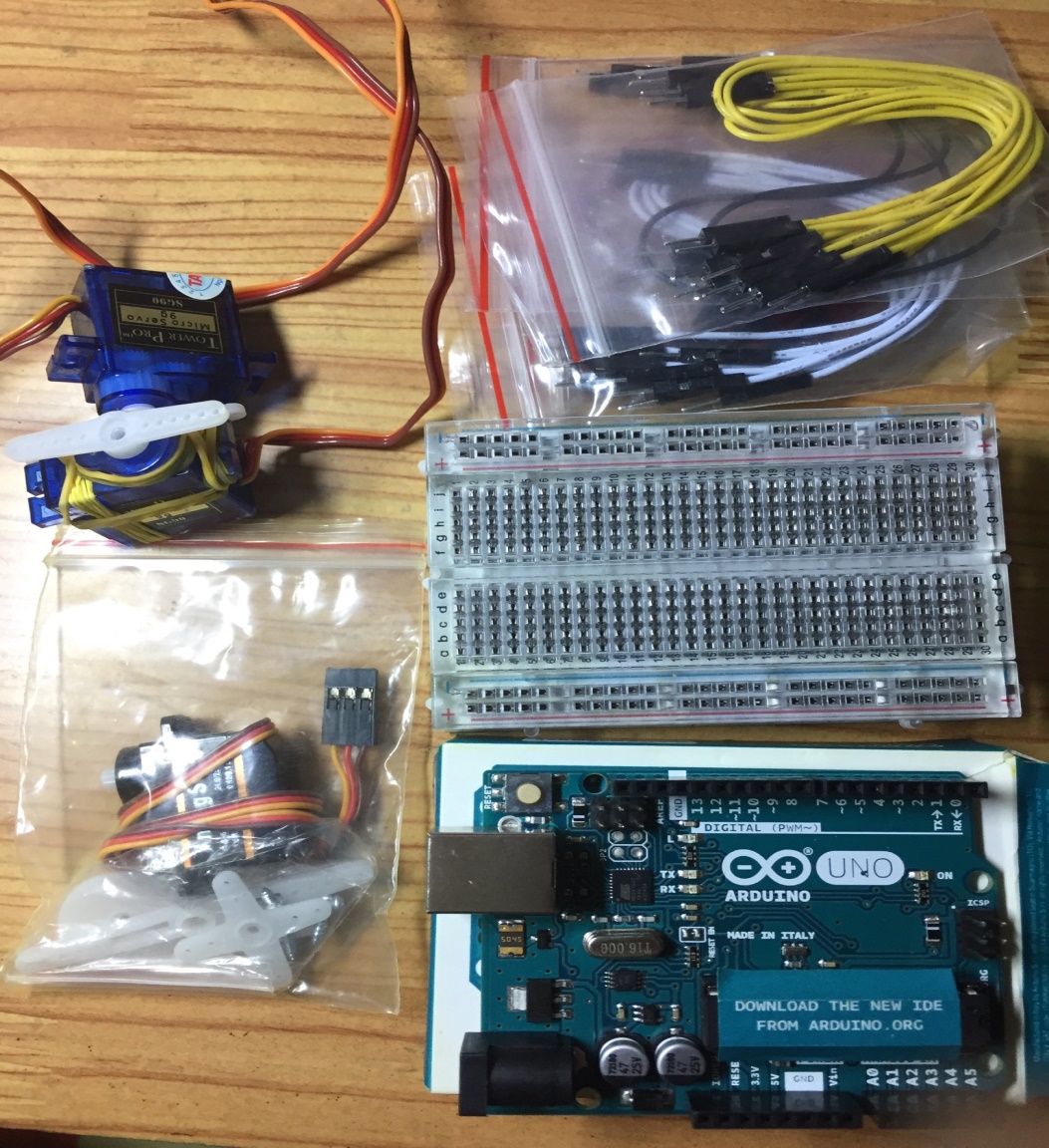
-Vậy là xong phần code nhận diện khuôn mặt, tuy nhiên để nó kết nối được với server của Arduino để nhận dữ liệu và thực hiện xoay camera cần phải có thêm lớp Tserial.cpp. File source của lớp này đã đính kèm trong bộ tài liệu gồm tất cả các file source code bao gồm cả code điều khiển động cơ nạp cho Arduino.

-Đây là thành quả sau khi đã thực hiện thành công, khuôn mặt đã được nhận diện bởi sự hỗ trợ của OpenCV.



Hình 13 : Nhận diện khuôn mặt bằng OpenCV.

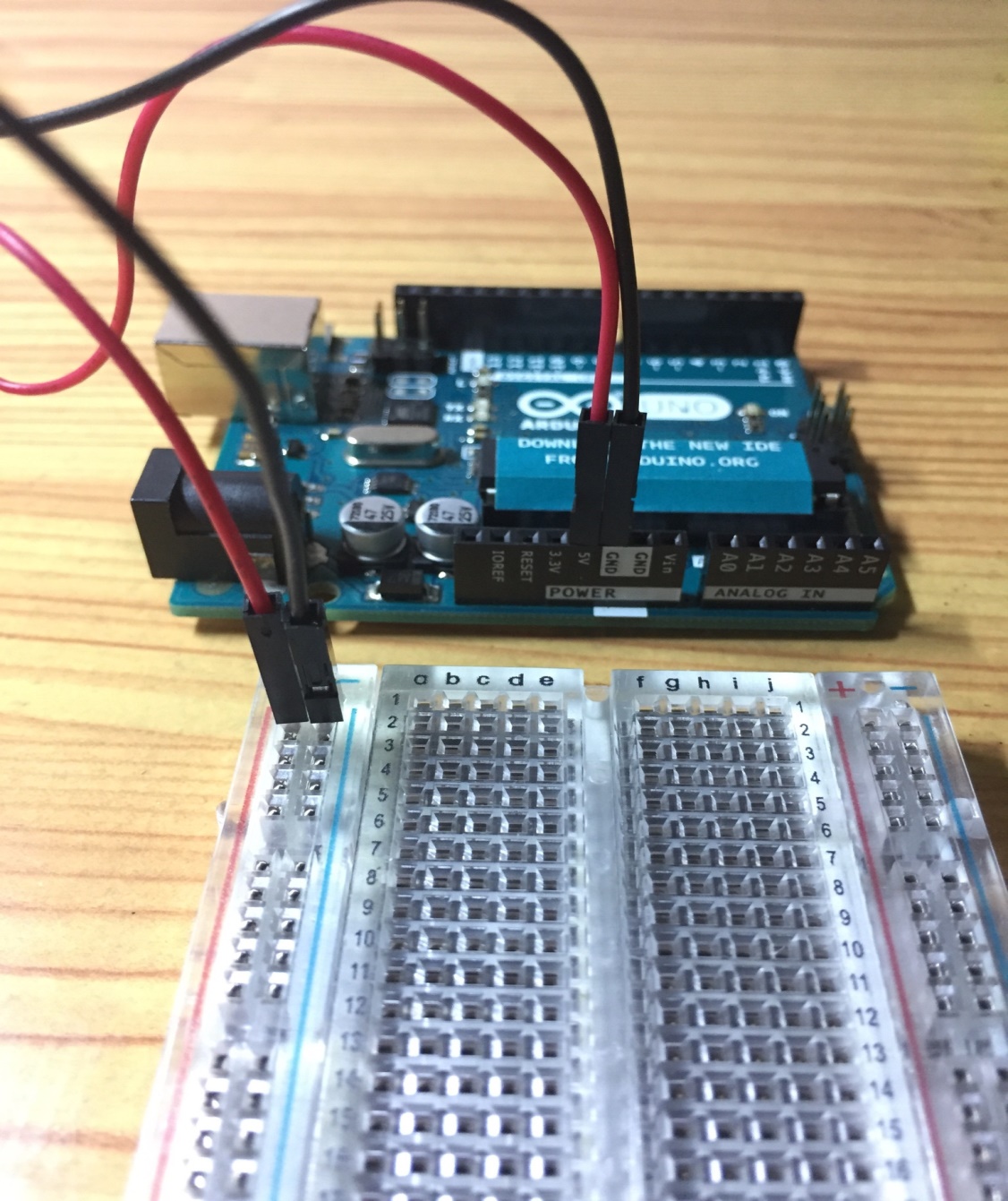
4.3 Lắp đặt phần cứng



Hình 14: Các thiết bị cần có

1. Servo Motors x 2

2. Webcam (bất kỳ loại webcam nào) x 1  
3. Arduino Uno x 1  
4. BreadBoard x 1  
5. Dây nối



Hình 15: Cách nối Arduino vào Breadboard

-Cắm dây từ chân 5V của arduino và nối vào cột + (Màu đỏ trong) Breadboard.

-Cắm dây vào chân GND của arduino và nối vào cột - (Màu xanh trong) Breadboard.

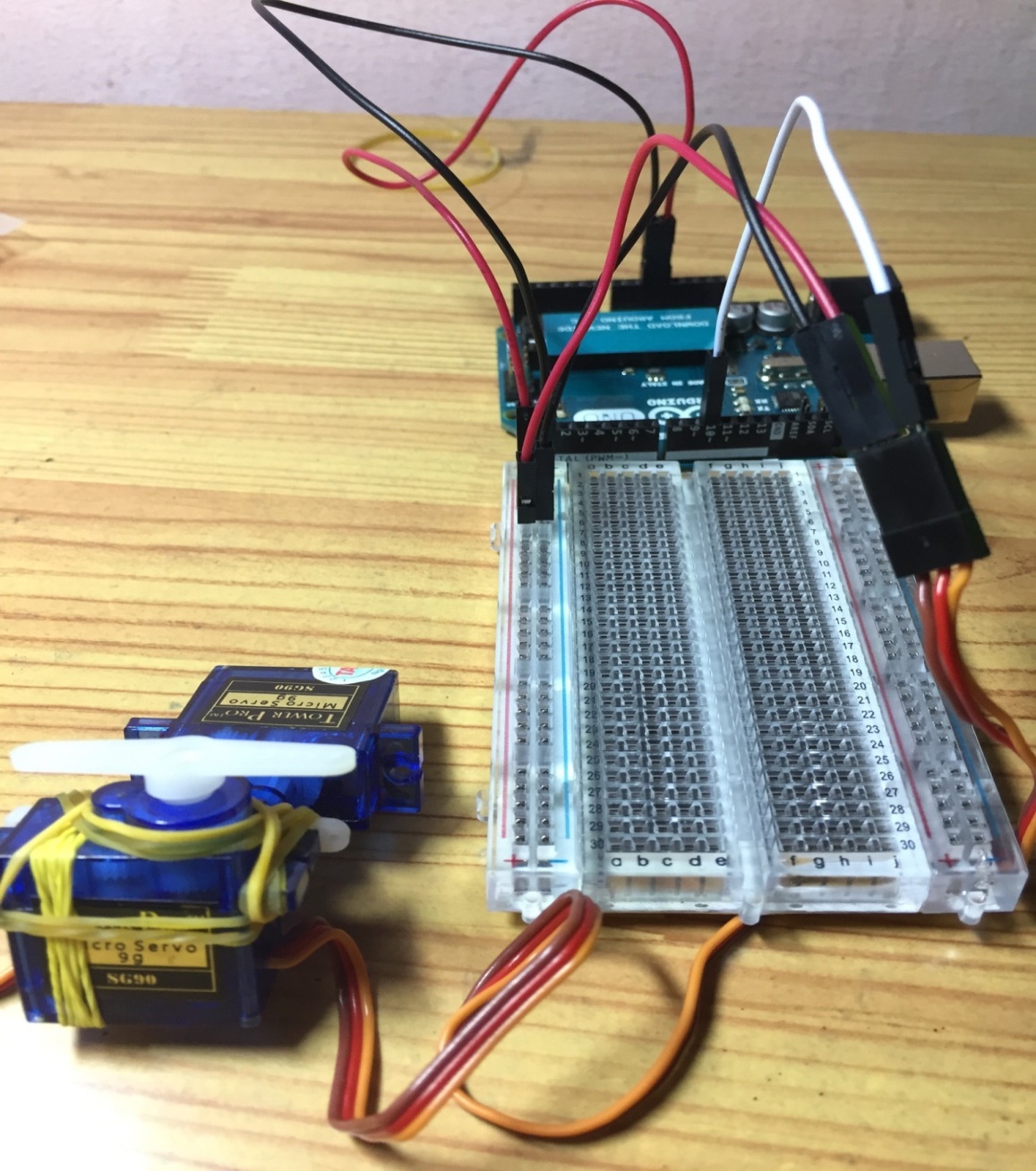
-Cách lắp mạch cho động cơ Servo.

-Nguyên tắc: Động cơ servo có 3 dây màu đỏ, nâu và vàng.

+Gắm dây màu nâu vào cột – (Màu xanh) trong Breadboard.

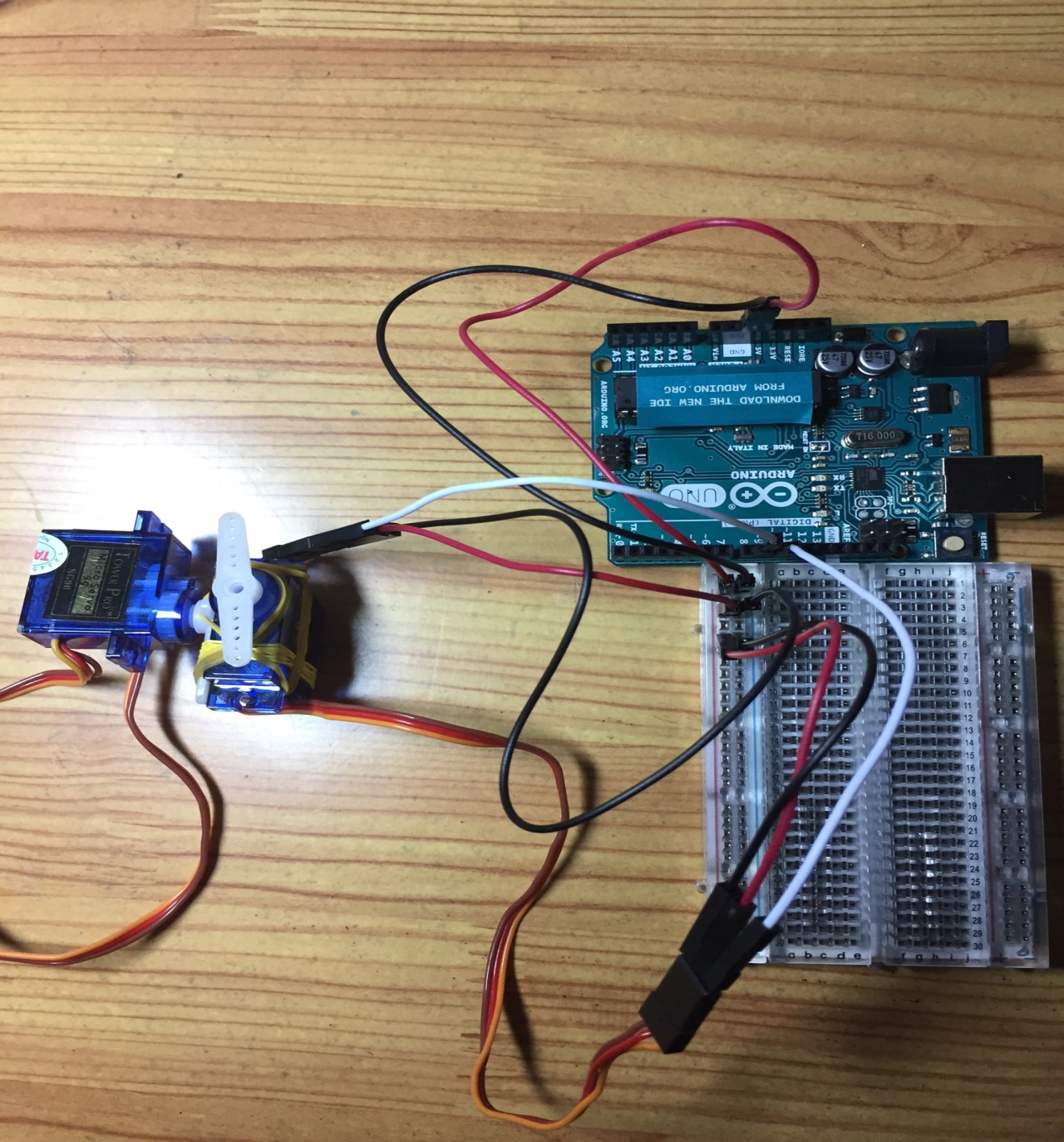
+Gắm dây màu đỏ vào cột + (Màu đỏ) trong Breadboard.

+Gắm dây màu vàng vào chân số của Arduino(Tùy vào cách khai báo trong Arduino mà ta cắm dây- đối với bài này khai báo hai chân số 9 và 10).



Hình 16: Cách nối dây động cơ Servo

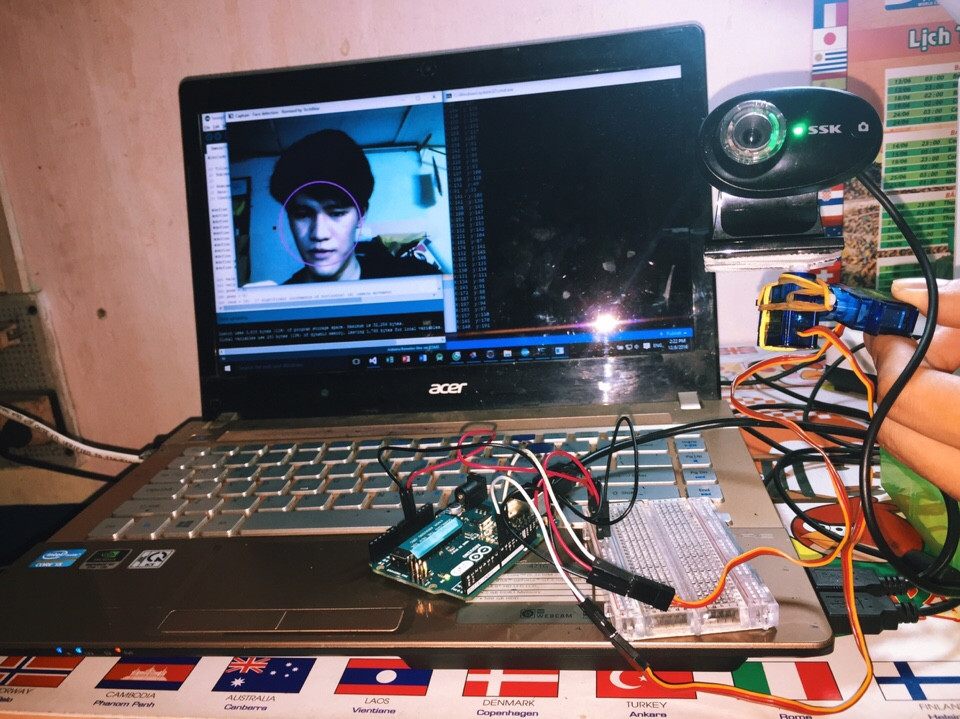
-Đây là 2 động cơ servo sau khi lắp. Hai động cơ servo lắp vào chân số 9 và số 10 của Arduino.



Hình 17: Cách nối dây động cơ Servo vào Breadboard và Arduino

-Dùng cáp nối Arduino vào máy tính để lấy nguồn, mở OpenCV lên để bắt đầu nhận diện hình ảnh, OpenCV sẽ gửi tín hiệu về cho Arduino, Arduino sẽ điều khiển hai động cơ Servo gắn theo camera xoay theo chuyển động.

* Đây là hệ thống sau khi hoàn thành.



Hình 18: Hệ thống camera xoay theo chuyển động.

CHƯƠNG 5: KẾT LUẬN

5.1 Những vấn đề đã giải quyết được :

+ Cài đặt, tích hợp OpenCV vào Visual Studio.

+ Lắp mạch điều khiển động cơ Servo

+ Nạp code điều khiển động cơ cho Arduino.

+ Xây dựng được hệ thống camera xoay theo chuyển động.

5.2 Những vấn đề chưa giải quyết được và khó khăn gặp phải:

+ Hệ thống nhúng, OpenCV, Servo và Arduino là những mảng kiến thức mới nên chưa tìm hiểu sâu, code tham khảo trên mạng vẫn chưa hiểu tường tận.

+ Chưa xây dựng được một hệ thống camera tự động thật sự hiệu quả, việc xử lý ảnh và gửi tín hiệu về Arduino vẫn chưa hoạt động thật sự trơn tru.

+ Phương hướng khắc phục: Để tăng tính ổn định cho Arduino và động cơ chúng ta có thể cấp nguồn riêng bên ngoài hoặc có thể thay thế động cơ servo lớn hơn (Trong bài dùng servo SG90 9g Micro hơi nhỏ so với kích thước của camera), nâng cao khả năng nhận dạng chuyển động trong OpenCV.

5.3 Những kiến thức có được khi hoàn thành đồ án:

+ Có kiến thức cơ bản về hệ thống nhúng.

+ Sử dụng OpenCV để xử lý hình ảnh.

+ Biết cấu tạo và cách sử dụng Arduino.

+ Biết cách điều khiển động cơ Servo.

+ Biết nguyên lí lắp mạch điện cho Arduino và Servo.

CHƯƠNG 6: TÀI LIỆU THAM KHẢO

* 1. Getting Started with Arduino – Tác giả : Massimo Banzi,

tái bản lần 2 năm 2011.

* 1. Beginning Arduino – Tác giả : Micheal McRoberts, xuất bản ngày 22 tháng 12 năm 2010.
  2. Learning OpenCV – Tác giả : Adrian Kaehler và Gary Rost Bradski
  3. Practical OpenCV - Tác giả : Samarth Brahmbhatt, xuất bản ngày 19 tháng 11 năm 2013.