

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM
KHOA CNKT ĐIỆN – ĐIỆN TỬ



ĐỒ ÁN HỌC PHẦN 1
ĐỀ TÀI:
CÁNH TAY ROBOT PHÂN LOẠI
SẢN PHẨM THEO MÀU SẮC

GVHD:

SVTH :

Lớp :

MSSV :

...., tháng... năm...

LỜI NÓI ĐẦU

Đồ án học phần 1

Cùng với sự phát triển không ngừng của các ngành khoa học kỹ thuật, các ngành công nghiệp cũng phát triển nhanh chóng. Việc áp dụng các máy móc hiện đại vào sản xuất là một yêu cầu không thể thiếu trong các nhà máy nhằm tăng năng suất, tăng chất lượng và giảm giá thành sản phẩm.

Song song với sự phát triển đó, công nghệ chế tạo Robot cũng phát triển nhanh chóng đặc biệt là ở các nước phát triển nhằm đáp các nhu cầu về sản xuất, sinh hoạt, quốc phòng... Robot có thể thực hiện những công việc mà con người khó thực hiện và thậm chí không thực hiện được như: làm những công việc đòi hỏi độ chính xác cao, làm việc trong môi trường nguy hiểm (như lò phản ứng hạt nhân, dò phá mìn trong quân sự), thám hiểm không gian vũ trụ...

Trong các họ Robot, chúng ta không thể không nhắc đến ROBOT CÔNG NGHIỆP với những đặc thù mà những Robot khác không có. Cánh tay robot phân loại sản phẩm theo màu với khả năng linh hoạt, độ chính xác cực cao, được ứng dụng rộng rãi trên khắp các nước thay thế con người làm những công việc ở trong môi trường độc hại hay chế tạo Robot khác với độ chính xác mà con người không thể làm bằng tay được.

Sau một thời gian làm việc, nghiên cứu, tham khảo chúng em đã hoàn thành đề tài cánh tay robot phân loại sản phẩm theo màu, bài làm còn dựa trên nhiều lý thuyết, vì vậy chúng em đang hoàn thiện và ứng dụng trong thực tế.

Em xin chân thành cảm ơn thầy ***** đã giúp em hoàn thành đồ án này.

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

- NỘI DUNG ĐỒ ÁN:

.....

.....

.....

.....

.....

- Ý KIẾN ĐÁNH GIÁ:

.....

.....

.....

.....

.....

.....

Ngày tháng năm 2017

Giáo viên hướng dẫn

Ký tên

MỤC LỤC:

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	5
1. LÝ DO CHỌN ĐỀ TÀI:	5
2. LÝ DO CHỌN BOARD ARDUINO ĐỂ ĐIỀU KHIỂN CÁNH TAY ROBOT PHÂN LOẠI SẢN PHẨM:	6
CHƯƠNG 2: TỔNG QUAN VỀ CÁNH TAY ROBOT PHÂN LOẠI SẢN PHẨM THEO MÀU SẮC....	7
2.1. GIỚI THIỆU:	7
2.2. LỊCH SỬ PHÁT TRIỂN:	8
2.3. PHÂN LOẠI ARM ROBOT:	10
2.4. ỨNG DỤNG:	13
CHƯƠNG 3: TỔNG QUAN THIẾT BỊ VÀ PHẦN MỀM SỬ DỤNG	14
3.1 GIỚI THIỆU CHUNG VỀ ARDUINO:	14
1.2.1 Arduino Uno:	18
2.2.1 Arduino Mega 2560:	21
3.2 GIỚI THIỆU CHUNG VỀ CẢM BIẾN:	26
3.2.1 Giới thiệu cảm biến:	26
3.2.2 Cảm biến màu sắc TCS3200:	26
3.3 ĐỘNG CƠ SERVO:	28
3.3.1 Động cơ RC Servo Digital RC FR1510:	28
3.3.2 Động cơ RC Servo 9G:	29
3.4 KHUNG CÁNH TAY ROBOT:	30
3.5 GIỚI THIỆU PHẦN MỀM IDE:	31
CHƯƠNG 4: KẾT NỐI LINH KIỆN, NGUYÊN LÝ HOẠT ĐỘNG VÀ LẬP TRÌNH	34
4.1. KẾT NỐI LINH KIỆN:	34
4.1.1. Kết nối Servo với Arduino:	34
4.1.2. Kết nối Cảm biến màu sắc với Arduino:	35
4.2. NGUYÊN LÝ HOẠT ĐỘNG:	36
4.3. LẬP TRÌNH:	38
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	43
5.1. KẾT LUẬN:	43

Đồ án học phần 1

5.1.1.	Những nhiệm vụ đã thực hiện:	43
5.1.2.	Những thuận lợi và khó khăn khi thực hiện đề tài:	43
5.2.	HƯỚNG PHÁT TRIỂN ĐỀ TÀI:	44
<i>TÀI LIỆU THAM KHẢO:</i>		45

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1. LÝ DO CHỌN ĐỀ TÀI:

Hiện nay với nền công nghiệp hóa hiện đại hóa, với sự phát triển của các kỹ thuật hiện đại, nền công nghiệp 4.0 cần sản xuất rất nhiều linh kiện hay chế tạo máy móc, hay trong nền công nghiệp sản xuất hàng hóa cũng cần hiện đại hóa. Việc sắp xếp và phân loại là một công đoạn hết sức thiết yếu, để giảm tiết kiệm thời gian, công sức và tiền bạc, việc chế tạo các tay máy để phục vụ nhu cầu này trong công nghiệp đã và đang được phát triển rộng rãi ở nước ta.

Với ngành công nghiệp của Việt Nam thì robot chưa được xuất hiện nhiều trong các dây chuyền sản xuất. Vì sản phẩm này còn quá đắt đối với thị trường Việt Nam. Nhằm nội địa hóa sản phẩm, cũng như nghiên cứu chuyên sâu về robot, em chọn đề tài “Cánh tay robot phân loại sản phẩm theo màu sắc”. Đề tài này hướng tới có thể thay thế các bộ điều khiển của các công ty nước ngoài và xây dựng thuật điều khiển tối ưu cho các đối tượng sản xuất, mà các đối tượng này thích hợp với điều kiện sản xuất ở nước ta.

Với các phòng thí nghiệm, đây là một mô hình để sinh viên thực nghiệm và nghiên cứu, để hướng tới cho các bạn sinh viên một cái nhìn cụ thể, thực tiễn hơn về robot.

Mục đích của đề tài này là nghiên cứu về cấu tạo và các phương pháp điều khiển thích hợp trên cơ sở ứng dụng arduino để hiểu được bộ điều khiển cánh tay robot phân loại sản phẩm theo màu sắc. có thể áp dụng vào phòng thí nghiệm của các trường cao đẳng, đại học cũng như ứng dụng trong sản xuất công nghiệp.

2. LÝ DO CHỌN BOARD ARDUINO ĐỂ ĐIỀU KHIỂN CÁNH TAY ROBOT PHÂN LOẠI SẢN PHẨM:

- Về phần cứng :

Arduino được thiết kế chuyên biệt dành cho những người không chuyên về điện tử vẫn có thể làm được. Bạn không phải mất thời gian ở giai đoạn “làm mạch”, mọi thứ đã có sẵn và bạn chỉ cần tập trung cho phần điều khiển mà thôi. Nếu có gì đó không ổn xảy ra, bạn cũng sẽ đỡ mất thời gian hơn cho việc rà soát lỗi ở phần mạch, mọi lỗi sẽ nằm ở code của bạn.

Arduino rất dễ sử dụng, trực quan, trên mạch có ký hiệu rất rõ ràng, đầy đủ các chân, cực kỳ thuận tiện trong quá trình sử dụng.

- Về phần mềm :

Môi trường phát triển tích hợp (IDE) của Arduino là một ứng dụng cross-platform (nền tảng) được viết bằng Java, và từ IDE này sẽ được sử dụng cho Ngôn ngữ lập trình xử lý (Processing programming language) và project Wiring. Nó được thiết kế để dành cho những người mới tập làm quen với lĩnh vực phát triển phần mềm. Nó bao gồm một chương trình code editor với các chức năng như đánh dấu cú pháp, tự động brace matching, và tự động canh lề, cũng như compile (biên dịch) và upload chương trình lên board chỉ với 1 cú nhấp chuột. Một chương trình hoặc code viết cho Arduino được gọi là một sketch.

Với những ưu điểm vượt trội trên em đã chọn Board arduino để làm bộ điều khiển cho đề tài cánh tay robot phân loại sản phẩm theo màu sắc.

CHƯƠNG 2: TỔNG QUAN VỀ CÁNH TAY ROBOT PHÂN LOẠI SẢN PHẨM THEO MÀU SẮC

2.1. GIỚI THIỆU:

Đề tài cánh tay robot phân loại sản phẩm là một phân nhánh trong Robot công nghiệp vì vậy em xin giới thiệu tổng quát về lịch sử ra đời từ Robot công nghiệp.

Thuật ngữ Robot xuất hiện lần đầu tiên vào năm 1922 trong tác phẩm “Rossum’s Universal Robot” của Karel Capek. Trong tác phẩm nhân vật Rossum và con trai đã tạo ra chiếc máy giống con người để phục vụ cho con người.

Năm 40 nhà văn viễn tưởng Nga, Issac Asimov, mô tả robot là một chiếc máy tự động, mang diện mạo con người được điều khiển bằng một hệ thần kinh khả trình Positron, do chính con người lập trình. Asimov đặt tên cho ngành nghiên cứu về robot là robotics, trong đó có 3 nguyên tắc cơ bản:

- Robot không được xúc phạm con người và không gây tổn hại cho con người.
- Hoạt động của robot phải tuân thủ các quy tắc do con người đặt ra. Các quy tắc này không được vi phạm nguyên tắc thứ nhất.
- Một robot cần phải bảo vệ sự sống của mình, nhưng không được vi phạm hai nguyên tắc trước.

Robot là một tác nhân cơ khí, nhân tạo, ảo, thường là một hệ thống cơ khí-điện tử. Với sự xuất hiện và chuyển động của mình, robot gây cho người ta cảm giác rằng nó giác quan giống như con người. Từ "robot" thường được hiểu với hai nghĩa: robot cơ khí và phần mềm tự hoạt động. Về lĩnh vực Robot, Mỹ và Nhật Bản là những nước đi đầu thế giới về lĩnh vực này.

Tay robot có thể được thiết kế để thực hiện bất kỳ công việc mong muốn như hàn, nắm, quay, vv, tùy thuộc vào ứng dụng. Ví dụ, cánh tay robot trong dây chuyền lắp ráp ô tô thực hiện nhiều công việc khác nhau như hàn và xoay linh kiện và lắp đặt trong quá trình lắp ráp. Trong một số trường hợp, sự mô phỏng chặt chẽ của bàn tay con người là mong muốn, như trong robot được thiết kế để tiến hành giải trừ vũ khí và thải bỏ bom.

2.2. LỊCH SỬ PHÁT TRIỂN:

Sau sự xuất hiện vào năm 1922, hơn 20 năm sau, sau chiến tranh thế giới thứ 2 Hình dạng Robot xuất hiện đầu tiên ở nước Hoa Kỳ, là loại tay máy chép hình dung trong phòng thí nghiệm vật liệu phóng xạ. Vào những năm 50 của thế kỷ trước, bên cạnh các loại tay máy chép hình cơ khí, các loại tay máy chép hình thủy lực điện tử đã xuất hiện.

Chiếc robot công nghiệp được đưa vào ứng dụng đầu tiên. Năm 1961. ở một nhà máy ô tô của General Motors tại Trenton, New Jersey Hoa Kỳ. Năm 1967 Nhật Bản mới nhập chiếc robot công nghiệp đầu tiên từ Công ty AMF của Hoa Kỳ (American Machine and Foundry Company).

Nên năm 1990 có hơn 40 công ty Nhật Bản, trong đó có những công ty không lẽ như Công ty Hitachi và Công ty Mitsubishi, đã đưa ra thị trường quốc tế nhiều loại robot nổi tiếng. Từ những năm 70 việc nghiên cứu nâng cao tính năng của robot đã chú ý nhiều đến sự lắp đặt thêm các cảm biến ngoại tín hiệu để nhận biết môi trường làm việc. Tại trường Đại học Tổng hợp Stanford người ta đã tạo ra loại robot lắp ráp tự động điều khiển bằng máy vi tính trên cơ sở xử lý thông tin từ các cảm biến lực và thị giác. Vào thời gian này công ty IBM đã chế tạo loại robot có các cảm biến xúc giác và cảm biến lực. điều khiển bằng máy tính để lắp ráp các máy in gồm 20 cụm chi tiết.

Theo Viện Kỹ Thuật robot của Hoa Kỳ định nghĩa robot là loại tay máy nhiều chức năng, với chương trình làm việc thay đổi được, dùng để thực hiện một số thao tác sản xuất. Có nhiều tài liệu khi định nghĩa robot rất lưu ý đến tiêu chí điều khiển bằng máy tính nhưng trong phân loại robot công nghiệp theo tiêu chuẩn của Nhật Bản (JISB 0134-1979) có cả nhóm tay máy điều khiển bằng tay.

Theo ISO (International Standards Organization) thì: “robot công nghiệp là một tay máy đa mục tiêu, có một số bậc tự do, dễ dàng lập trình, điều khiển tự động, dùng để tháo lắp phôi, dụng cụ hoặc các vật dụng khác. Do chương trình thao tác có thể thay đổi nên thực hiện nhiều nhiệm vụ đa dạng”. Tuy nhiên, robot công nghiệp được định nghĩa như thế là chưa hoàn toàn thỏa đáng.

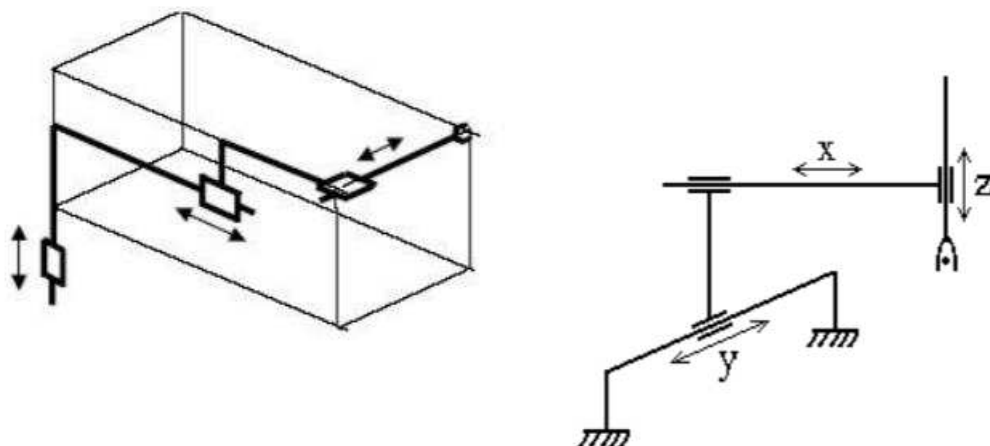
Robot công nghiệp có thể được hiểu là những thiết bị tự động linh hoạt, bắt chước được các chức năng lao động công nghiệp của con người. Nói đến thiết bị tự động linh hoạt là nhấn mạnh đến khả năng thao tác với nhiều bậc tự do, được điều

Đồ án học phần 1

khởi trợ động và lập trình thay đổi được. Còn nói đến sự bắt chước các chức năng lao động công nghiệp của con người là có ý nói đến sự không hạn chế từ các chức năng lao động chân tay đơn giản đến trí khôn nhân tạo, tùy theo loại hình công việc lao động cần đến chức năng đó hay không. Đồng thời cũng nói đến mức độ cần thiết bắt chước được như con người hay không.

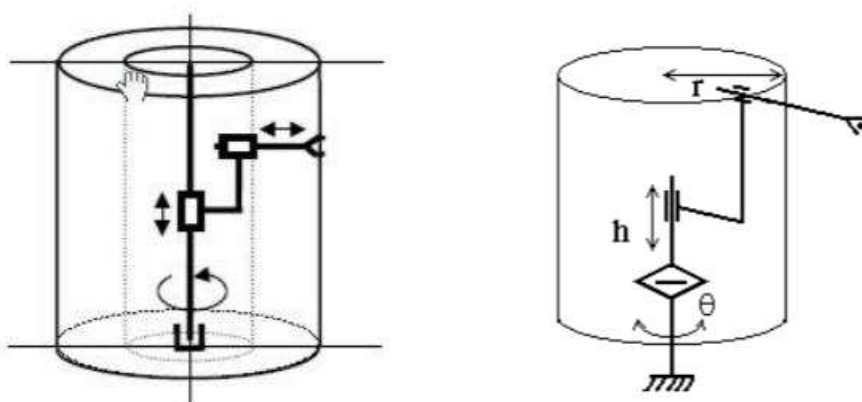
2.3. PHÂN LOẠI ARM ROBOT:

- 1 Robot Cartesian / Gantry Robot (tọa độ DECAC): (Hình 2.3.1) Được sử dụng để chọn và đặt công việc, ứng dụng sealant, hoạt động lắp ráp, xử lý các máy công cụ và hàn hồ quang. Đó là một robot có cánh tay có ba khớp lồng trụ, có trục khớp.



Hình 2.3.1 : Robot hoạt động theo tọa độ DECAC.

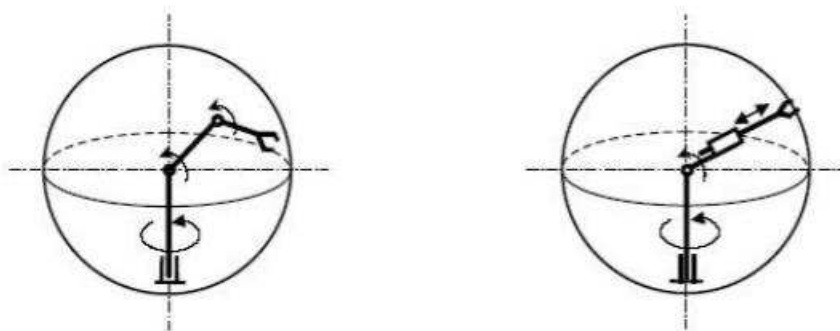
- 2 Robot trụ : (Hình 2.3.2) Được sử dụng cho các hoạt động lắp ráp, xử lý tại các máy công cụ, hàn điểm, và xử lý tại các máy đúc. Đó là một robot có trục tạo thành một hệ trục tọa độ.



Hình 2.3.2 : Robot hoạt động theo hệ tọa độ trụ.

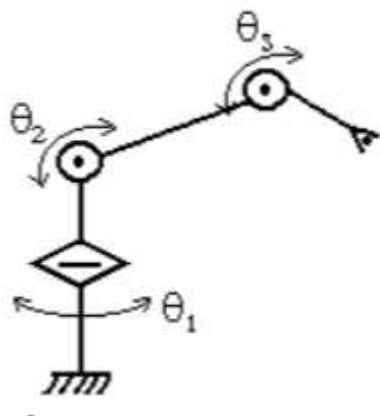
Đồ án học phần 1

- 3 Robot hình cầu / Robot Polar : (Hình 2.3.3) Được sử dụng để xử lý các máy công cụ, hàn điểm, máy đúc, máy hàn nhiệt, hàn hồ quang và hàn hồ quang. Đó là một robot có trục tạo thành một hệ tọa độ cực.



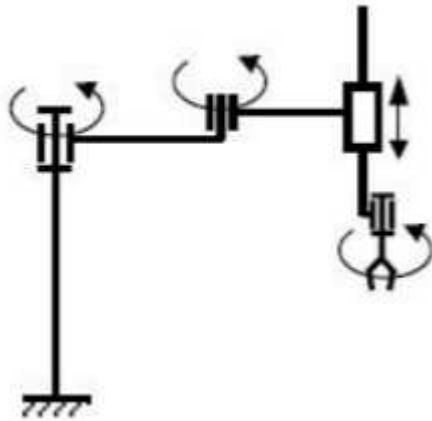
Hình 2.3.3 : Robot hoạt động theo hệ tọa độ cầu.

- 4 SCARA robot : (Hình 2.3.4) Được sử dụng để chọn và đặt công việc, ứng dụng sealant, hoạt động lắp ráp và xử lý các máy công cụ. Robot này có hai khớp nối song song để cung cấp sự tuân thủ trong máy bay.



Hình 1.3.4 : Robot kiểu SCARA.

- 5 Robot khớp nối : (Hình 2.3.5) Được sử dụng cho các hoạt động lắp ráp, diecasting, fettling máy móc, khí hàn, hàn hồ quang và phun sơn. Đó là một robot có cánh tay có ít nhất ba khớp quay.



Hình 2.3.5 : Robot khớp nối

- 6 Robot song song : (Hình 2.3.6) Một sử dụng là một nền tảng di động xử lý mô phỏng bay buồng lái. Đó là một robot có cánh tay có khớp nối lăng trụ hoặc khớp quay đồng thời.

Hình 2.3.6 : Robot song song (3-PRS)

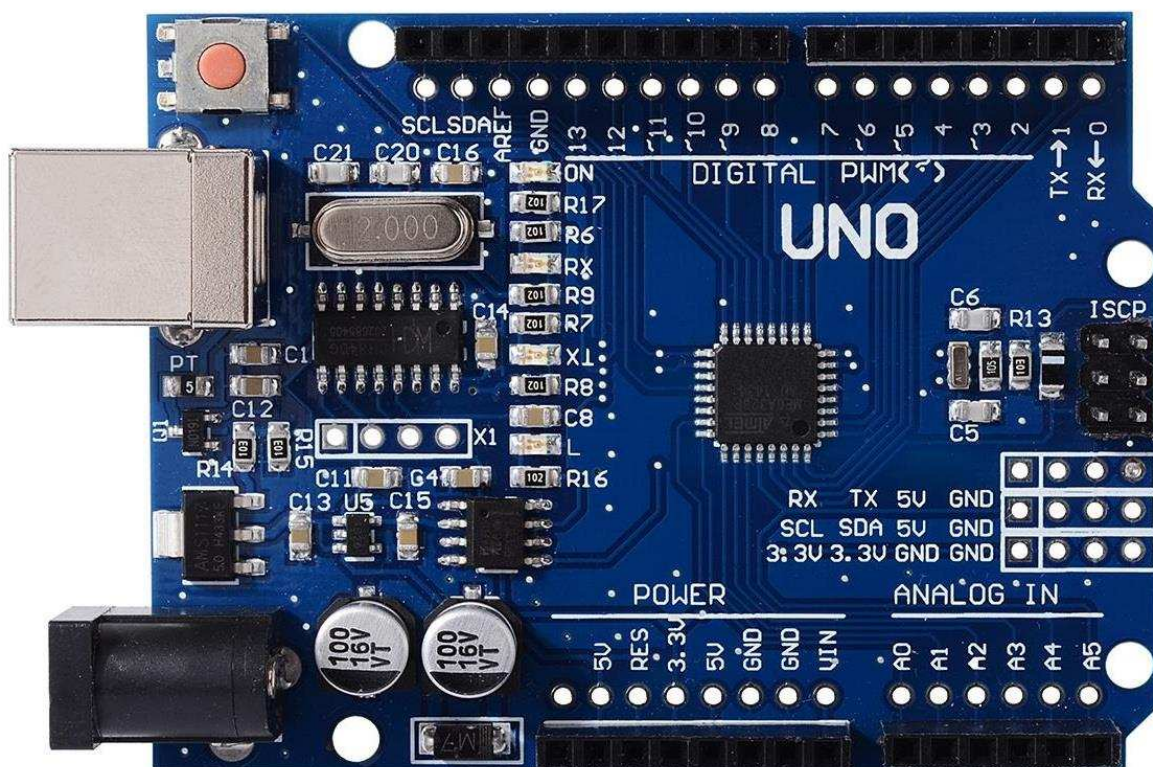


2.4. ỨNG DỤNG:

Robot cánh tay phân loại sản phẩm theo màu được ứng dụng trong các hệ thống phân loại các sản phẩm có màu khác nhau đã biết trước, khi phát triển đề tài hơn có thể thay vì màu sắc có thể theo kích thước hay mã vạch, mã hiệu được cung cấp trước trong hệ thống.

CHƯƠNG 3: TỔNG QUAN THIẾT BỊ VÀ PHẦN MỀM SỬ DỤNG

3.1 GIỚI THIỆU CHUNG VỀ ARDUINO:



Hình 3.1.1: Board mạch Arduino Uno.

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.

Được giới thiệu vào năm 2005, những nhà thiết kế của Arduino cố gắng mang đến một phương thức dễ dàng, không tốn kém cho những người yêu thích, sinh viên và giới chuyên nghiệp để tạo ra những thiết bị có khả năng tương tác với môi trường thông qua các cảm biến và các cơ cấu chấp hành. Những ví dụ phổ biến cho những người yêu thích mới bắt đầu bao gồm các robot đơn giản, điều khiển

Đồ án học phần 1

nhệt độ và phát hiện chuyển động. Đi cùng với nó là một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Aduino bằng ngôn ngữ C hoặc C++.

Arduino cơ bản là một nền tảng tạo mẫu mở về điện tử (open-source electronics prototyping platform) được tạo thành từ phần cứng lẫn phần mềm. Về mặt kỹ thuật có thể coi Arduino là 1 bộ điều khiển logic có thể lập trình được. Đơn giản hơn, Arduino là một thiết bị có thể tương tác với ngoại cảnh thông qua các cảm biến và hành vi được lập trình sẵn. Với thiết bị này, việc lắp ráp và điều khiển các thiết bị điện tử sẽ dễ dàng hơn bao giờ hết. Một điều không hề dễ dàng cho những ai đam mê công nghệ và điều khiển học nhưng là người ngoại đạo và không có nhiều thời gian để tìm hiểu sâu hơn về về kỹ thuật lập trình và cơ điện tử.

Hiện tại có rất nhiều loại vi điều khiển và đa số được lập trình bằng ngôn ngữ C/C++ hoặc Assembly nên rất khó khăn cho những người có ít kiến thức sâu về điện tử và lập trình. Nó là trở ngại cho mọi người muốn tạo riêng cho mình một món đồ mang tính công nghệ. Do vậy đó là lí do Arduino được phát triển nhằm đơn giản hóa việc thiết kế, lắp ráp linh kiện điện tử cũng như lập trình trên vi xử lí và mọi người có thể tiếp cận dễ dàng hơn với thiết bị điện tử mà không cần nhiều về kiến thức điện tử và thời gian. Sau đây là những thế mạnh của Arduino so với các nền tảng vi điều khiển khác:

- Chạy trên đa nền tảng: Việc lập trình Arduino có thể thực hiện trên các hệ điều hành khác nhau như Windows, Mac Os, Linux trên Desktop, Android trên di động.
- Ngôn ngữ lập trình đơn giản dễ hiểu.
- Nền tảng mở: Arduino được phát triển dựa trên nguồn mở nên phần mềm chạy trên Arduino được chia sẻ dễ dàng và tích hợp vào các nền tảng khác nhau.
- Mở rộng phần cứng: Arduino được thiết kế và sử dụng theo dạng module nên việc mở rộng phần cứng cũng dễ dàng hơn.
- Đơn giản và nhanh: Rất dễ dàng lắp ráp, lập trình và sử dụng thiết bị.
- Dễ dàng chia sẻ: Mọi người dễ dàng chia sẻ mã nguồn với nhau mà không lo lắng về ngôn ngữ hay hệ điều hành mình đang sử dụng.

Đồ án học phần 1

Arduino có rất nhiều module, mỗi module được phát triển cho một ứng dụng. Về mặt chức năng, các bo mạch Arduino được chia thành hai loại: loại bo mạch chính có chip Atmega và loại mở rộng thêm chức năng cho bo mạch chính. Các bo mạch chính về cơ bản là giống nhau về chức năng, tuy nhiên về mặt cấu hình như số lượng I/O, dung lượng bộ nhớ, hay kích thước có sự khác nhau. Một số bo có trang bị thêm các tính năng kết nối như Ethernet và Bluetooth. Các bo mở rộng chủ yếu mở rộng thêm một số tính năng cho bo mạch chính. Ví dụ như tính năng kết nối Ethernet, Wireless, điều khiển động cơ.

Arduino được chọn làm bộ não xử lý của rất nhiều thiết bị từ đơn giản đến phức tạp. Trong số đó có một vài ứng dụng thực sự chứng tỏ khả năng vượt trội của Arduino do chúng có khả năng thực hiện nhiều nhiệm vụ rất phức tạp. Sau đây là danh sách một số ứng dụng nổi bật của Arduino như trong công nghệ in 3D, robot dò đường theo hướng có nguồn nhiệt, tạo một thiết bị nhấp nháy theo âm thanh và đèn laser hay là một thiết bị báo cho khách hàng biết khi nào bánh mì ra lò. Đồ án tốt nghiệp SVTH: Nguyễn Đức Thịnh 22 Arduino đã và đang được sử dụng rất rộng rãi trên thế giới, và ngày càng chứng tỏ được sức mạnh của chúng thông qua vô số ứng dụng độc đáo của người dùng trong cộng đồng nguồn mở (open-source). Tuy nhiên tại Việt Nam Arduino vẫn còn chưa được biết đến nhiều.

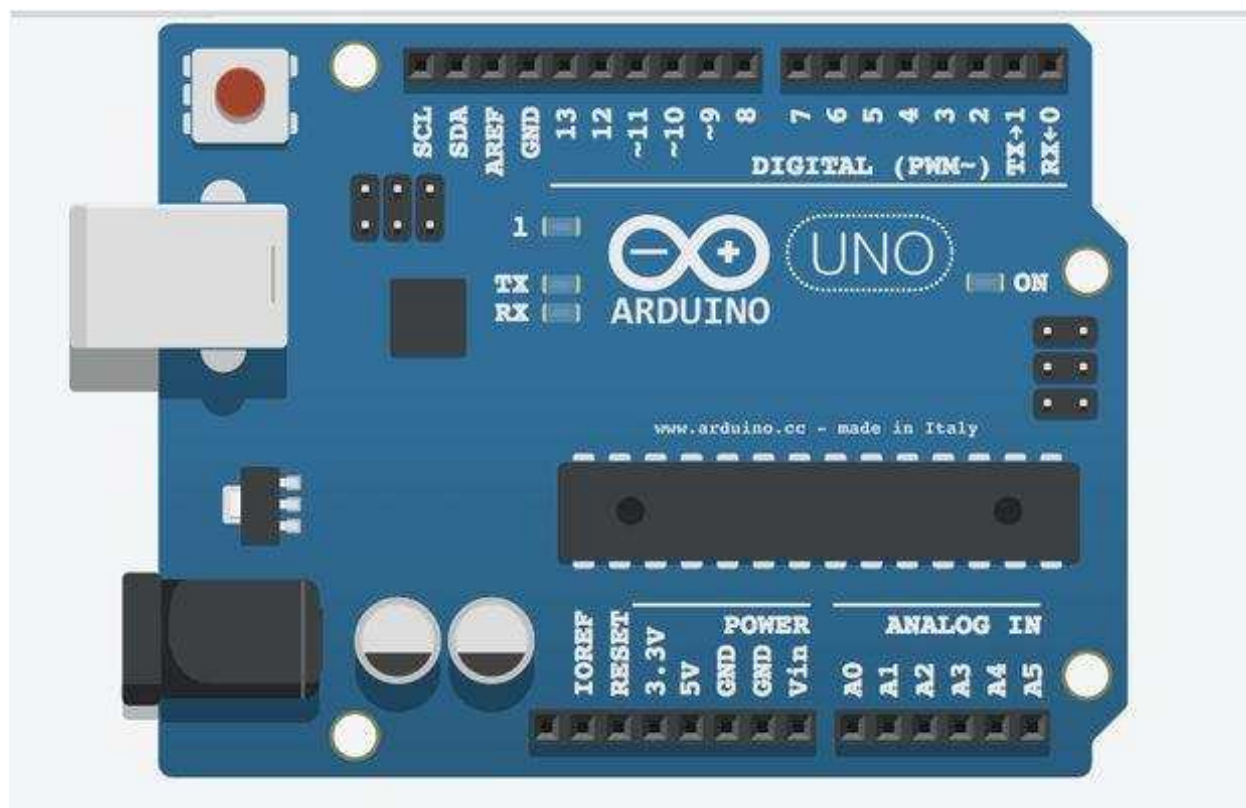
Đồ án học phần 1

ứng dụng của arduino trong thực tế:



Hình 3.1.3: Hệ thống cánh tay robot.

1.2.1 Arduino Uno:



Hình 3.1.1.1: Board mạch Arduino Uno.

Arduino Uno là một bo mạch vi điều khiển dựa trên chip Atmega328. Nó có 14 chân vào ra bằng tín hiệu số, trong đó có 6 chân có thể sử dụng để điều chế độ rộng xung. Có 6 chân đầu vào tín hiệu tương tự cho phép chúng ta kết nối với các bộ cảm biến bên ngoài để thu thập số liệu, sử dụng một dao động thạch anh tần số dao động 16MHz, có một cổng kết nối bằng chuẩn USB để chúng ta nạp chương trình vào bo mạch và một chân cấp nguồn cho mạch, một ICSP header, một nút reset. Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển, nguồn cung cấp cho Arduino có thể là từ máy tính thông qua cổng USB hoặc là từ bộ nguồn chuyên dụng được biến đổi từ xoay chiều sang một chiều hoặc là nguồn lấy từ pin.

Arduino có thể được hỗ trợ thông qua kết nối USB hoặc với một nguồn cung cấp điện bên ngoài. Các nguồn năng lượng được lựa chọn tự động. Hệ thống vi điều khiển có thể hoạt động bằng một nguồn cung cấp bên ngoài từ 6V đến 20V. Tuy nhiên, nếu cung cấp với ít hơn 7V, chân 5V có thể cung cấp ít hơn 5V và hệ

Đồ án học phần 1

thống vi điều khiển có thể không ổn định. Nếu cấp nhiều hơn 12V, bộ điều chỉnh điện áp có thể quá nóng và gây nguy hiểm cho bo mạch. Phạm vi khuyến nghị là 7V đến 12V.

- Chân Vin: Điện áp đầu vào Arduino khi chúng ta dùng nguồn điện bên ngoài (khác với nguồn 5V lấy từ USB hoặc nguồn thông qua jack cắm nguồn riêng). Chúng ta có thể cung cấp nguồn thông qua chân này. 25
- Chân 5V: Cung cấp nguồn vi điều khiển và các bộ phận khác trên bo mạch và cung cấp nguồn cho các thiết bị ngoại vi khi kết nối tới bo mạch.
- Chân 3,3V: Cung cấp nguồn cho các thiết bị cảm biến.
- Chân GND: Chân nối đất.
- Chân Aref: Tham chiếu điện áp đầu vào analog.
- Chân IOREF: Cung cấp điện áp cho các vi điều khiển hoạt động. Một shield được cấu hình đúng có thể đọc điện áp chân IOREF và lựa chọn nguồn thích hợp hoặc kích hoạt bộ chuyển đổi điện áp để là việc ở mức 5V hoặc 3,3V.

Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

- 32KB bộ nhớ Flash: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader .
- 2KB cho SRAM (Static Random Access Memory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM.
- 1KB EEPROM (Electrically Erasable Programmable Read Only Memory): đây giống như một chiếc ổ cứng mini – nơi bạn có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

Arduino Uno có 14 chân digital (chân 0 – 13) và 6 chân analog (chân A0 – A5).

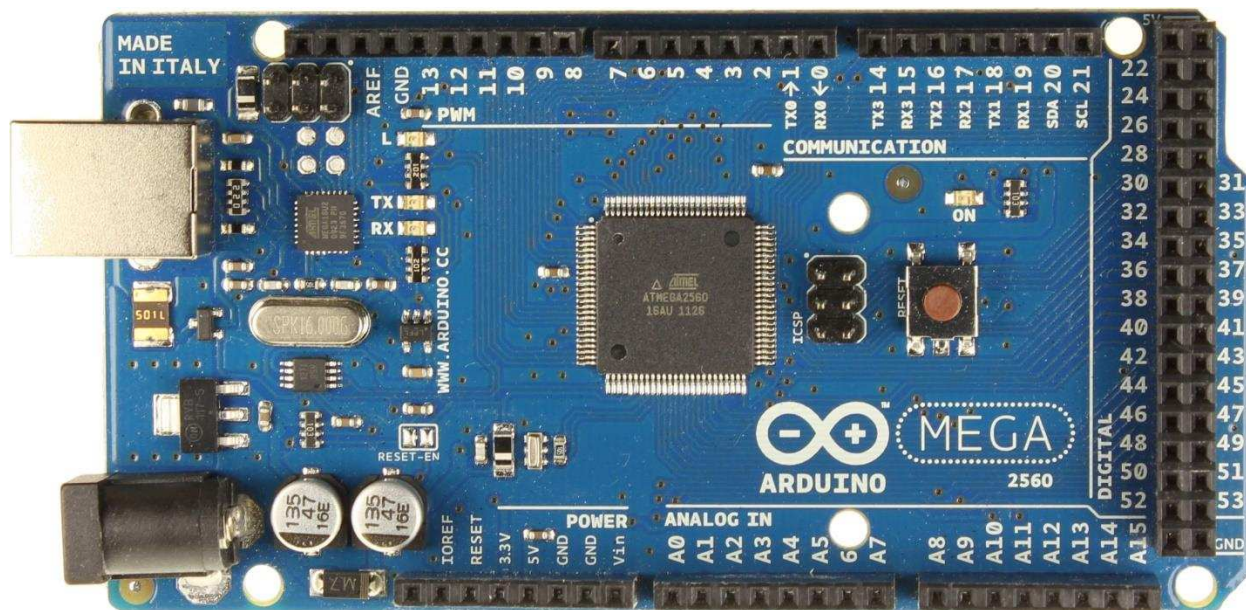
Các chân digital chúng ta có thể cấu hình để làm chân nhận dữ liệu vào từ các thiết bị ngoại vi hoặc làm chân để truyền tín hiệu ra các thiết bị ngoại vi. Bằng cách sử dụng các hàm pinMode(), digitalWrite() và digitalRead(). Mỗi chân có thể cung

Đồ án học phần 1

cấp hoặc nhận một dòng điện tối đa 40mA và có một điện trở kéo nội (mặc định không nối) 20 - 50 kOhms. Ngoài ra có một số chân có chức năng đặc biệt:

- Chân 0 (Rx): Chân được dùng để nhận dữ liệu nối tiếp.
- Chân 1 (Tx): Chân được dùng để truyền dữ liệu nối tiếp.
- Chân 2 và 3: Chân ngắt ngoài.
- Chân 3, 5, 6, 9, 10 và 11: Chân vào/ra số hoặc để điều chế độ rộng xung (chân 13 được nối với một LED đơn, sáng tắt tương ứng với mức logic của chân này).
- Chân 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK): Chuẩn giao tiếp SPI.
- Các chân analog có độ phân giải 10 bit (tương ứng với 1024 mức giá trị khác nhau) ứng với mức từ 0 – 5V. Ngoài ra một số chân có các chức năng đặc biệt:
- Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

2.2.1 Arduino Mega 2560:



Hình 3.1.2.1: Board mạch Arduino Mega 2560.

Arduino Mega 2560 là một bo mạch vi điều khiển được xây dựng dựa trên chip Atmega2560. Nó có 54 chân vào/ra số (trong đó có 15 chân có thể sử dụng để điều chế độ rộng xung), có 16 chân đầu vào tín hiệu tương tự, sử dụng một dao động thạch anh tần số dao động 16MHz, có một cổng kết nối USB, chân nguồn, một ICSP header, một nút reset. Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển, nguồn cung cấp cho Arduino có thể là từ máy tính thông qua cổng USB hoặc là từ bộ nguồn chuyên dụng được biến đổi từ xoay chiều sang một chiều hoặc là nguồn lấy từ pin. Arduino Mega tương thích với hầu hết các shield thiết kế cho Arduino Duemilanove hay Diecimila.

Arduino Mega 2560 là bản cập nhật từ Arduino Mega.

Arduino Mega 2560 khác so với các bo mạch trước đó ở chỗ nó không sử dụng chip điều khiển FTDI USB-to-serial. Thay vào đó, các tính năng của Atmega16U2 (ATmega8U2 trong phiên bản 1 và phiên bản 2 bảng) được lập trình như một bộ chuyển đổi USB-to-serial).

Phiên bản 2 của bo mạch Mega2560 có 1 điện trở kéo đường 8U2 HWB xuống đất, làm cho nó dễ đặt chế độ DFU hơn.

Đồ án học phần 1

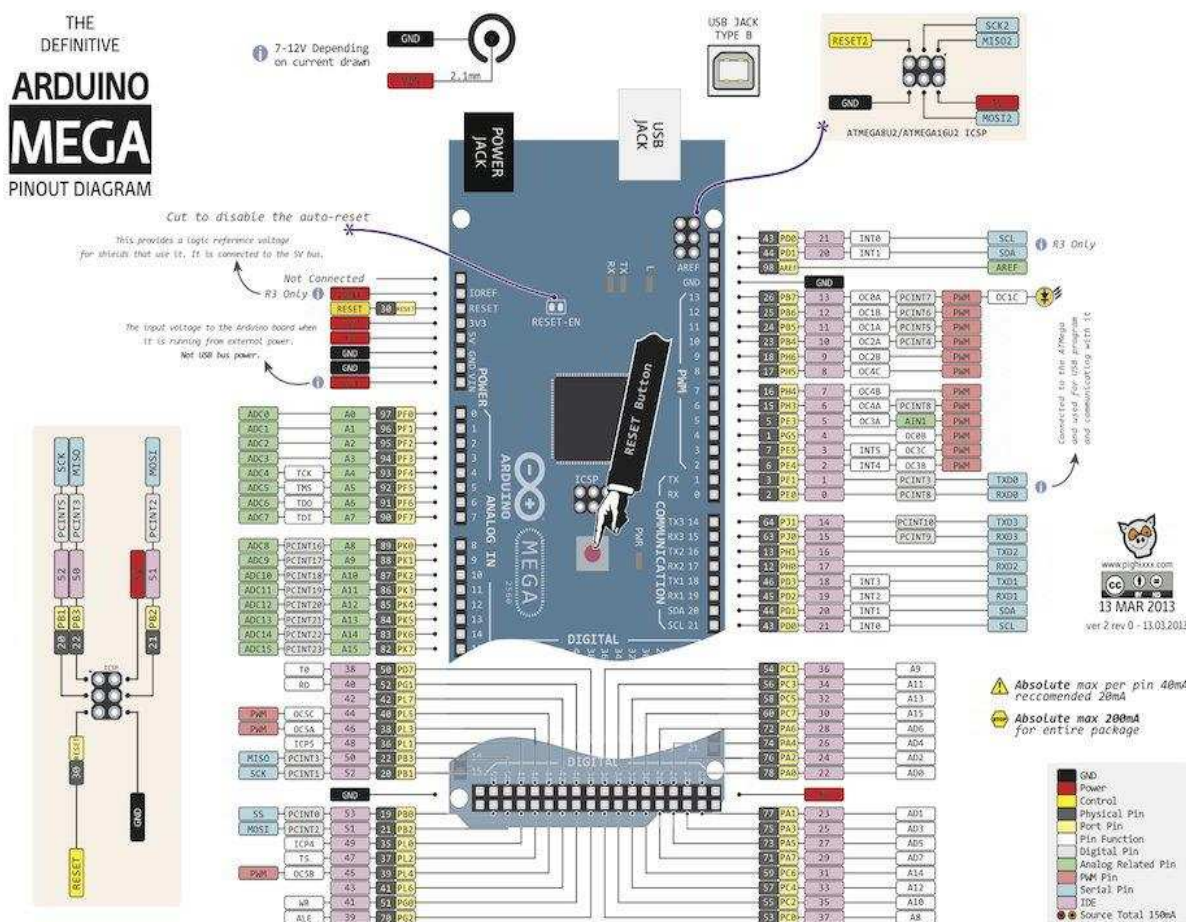
Phiên bản 3 của bo mạch có các đặc tính mới sau:

Thêm chân SDA và SCL gần chân AREF và 2 chân mới được đặt gần chân RESET, IOREF cho phép các shield tương thích với điện áp được cung cấp từ bo mạch. Trong tương lai, các shield sẽ tương thích với cả hai bo mạch sử dụng AVR mà hoạt động với nguồn 5V và Arduino Due hoạt động ở mức 3,3V. Chân thứ 2 không kết nối dành cho các mục đích sau này.

Đặc tính của board Atmega2560:

Vi điều khiển	ATmega2560
Điện áp hoạt động	5V
Điện áp đầu vào (được đề nghị)	7-12V
Điện áp đầu vào (giới hạn)	6-20V
Digital I/O Pins	54 chân (15 chân có thể sử dụng như các chân PWM)
Analog Input Pins	16 chân
DC hiện tại mỗi I/O Pin	40 mA
DC hiện tại cho 3.3V Pin	50 mA
Bộ nhớ Flash	256 KB trong 8 KB sử dụng bởi bộ nạp khởi động
SRAM	8 KB
EEPROM	4 KB
Clock Speed (thạch anh)	16 MHz

Đồ án học phần 1



Hình 3.1.2.2: Sơ đồ chân Board Arduino Mega 2560.

Khởi nguồn:

- Arduino Mega có thể được cấp nguồn thông qua kết nối USB hoặc nguồn ngoài. Nguồn nuôi được chọn một cách tự động.
- Nguồn ngoài (không phải USB) có thể lấy từ bộ chuyển đổi AC sang DC hoặc từ pin. Bo mạch có thể hoạt động với nguồn ngoài từ 6 – 20V. Tuy nhiên, nếu nguồn cấp nhỏ hơn 7V thì chân 5V có thể cấp không đủ 5V và bo mạch có thể chạy không ổn định. Nếu cấp lớn hơn 12V, bộ biến áp có thể bị nóng và ảnh hưởng tới mạch. Điện áp khuyến nghị là 7 – 12V.
- Các chân cấp nguồn:

Đồ án học phần 1

- VIN: Điện áp vào mạch Arduino khi nó sử dụng nguồn ngoài (khác với nguồn 5V từ kết nối USB hoặc nguồn khác). Ta có thể cấp nguồn qua chân này hoặc cấp thông qua jack cắm nguồn.
- 5V: Chân cấp điện áp ra 5V từ bộ điều chỉnh điện áp của bo mạch.
- 3V3: Chân cấp điện áp ra 3,3V từ bộ điều chỉnh điện áp.
- GND: Chân nối đất.
- IOREF: Chân này cấp điện áp tham chiếu cho vi điều khiển hoạt động. Một shield được cấu hình đúng có thể đọc điện áp chân IOREF và chọn nguồn điện phù hợp hoặc cho phép biến đổi điện áp để làm việc với 5,5V hoặc 3,3V.

Bộ nhớ: Atmega2560 có 256KB bộ nhớ flash để lưu trữ mã chương trình (trong đó có 8KB được sử dụng cho bootloader), 8KB SRAM và 4KB EEPROM (có thể đọc và viết bằng thư viện EEPROM).

Thông số kỹ thuật khác:

- Mỗi chân trong 54 chân digital trên bo Mega có thể sử dụng làm chân input hoặc output, bằng cách sử dụng hàm pinMode(), digitalWrite(), digitalRead(). Chúng hoạt động ở 5V. Mỗi chân có thể cấp hoặc nhận tối đa 40mA và có một trở kéo nội (mặc định không nối) từ 20 – 50Kohm. Ngoài ra, một số chân có các chức năng đặc biệt
- Nối tiếp: 0 (RX) và 1 (TX);
- Serial 1: 19 (RX) and 18 (TX);
- Serial 2: 17 (RX) and 16 (TX);
- Serial 3: 15 (RX) and 14 (TX).
- Sử dụng để nhận (RX) và truyền (TX) dữ liệu nối tiếp kiểu TTL.
- Ngắt ngoài:
- 2 (ngắt 0),
- 3 (ngắt 1),

Đồ án học phần 1

- 18 (ngắt 5),
 - 19 (ngắt 4),
 - 20 (ngắt 3),
 - 21 (ngắt 2).
- Những chân này có thể cấu hình để kích hoạt một ngắt ở mức thấp, sườn lên hoặc sườn xuống.
- PWM: 2 - 13 và 44 - 46. Cung cấp 8-bit PWM đầu ra với hàm `analogWrite()` - SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Những chân này hỗ trợ truyền thông SPI.
- LED: 13. Có một LED được nối với chân 13. Khi chân này ở mức cao thì đèn sáng, khi ở mức thấp thì đèn tắt.
- TWI: 20 (SDA) và 21 (SCL). Hỗ trợ truyền thông TWI. Lưu ý rằng những chân này không cùng số chân TW trên *Duemilanove* hay *Diecimila*.
- Mega2560 có 16 chân đầu vào analog, mỗi chân có độ phân giải 10 bit (1024 mức khác nhau). - Có một cặp chân khác trên bo:
- AREF: Tham chiếu điện áp cho đầu vào analog. Sử dụng hàm `analogReference()`.
- RESET: Nối đường dây xuống LOW để reset vi xử lý.

3.2 GIỚI THIỆU CHUNG VỀ CẢM BIẾN:

3.2.1 Giới thiệu cảm biến:

Cảm biến – sensor: xuất phát từ chữ “sense” nghĩa là giác quan – do đó nó như các giác quan trên cơ thể con người. Nhờ cảm biến mà mạch điện, hệ thống điện có thể thu thập thông tin từ bên ngoài. Từ đó hệ thống máy móc, điện tử tự động mới có thể tự động hiển thị thông tin về đối tượng hay điều khiển quá trình định trước có khả năng thay đổi một cách uyển chuyển theo môi trường hoạt động.

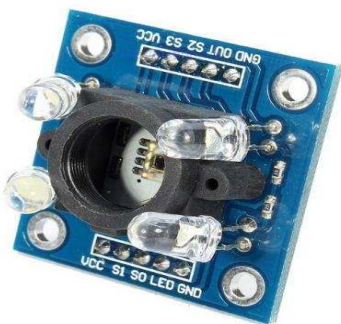
Cảm biến là thiết bị dùng để cảm nhận biến đổi các đại lượng vật lý và các đại lượng không có tính chất điện cần đo thành các đại lượng điện có thể đo và xử lý được.

Các bộ cảm biến đóng vai trò cực kỳ quan trọng trong lĩnh vực đo lường và điều khiển. Chúng cảm nhận và đáp ứng theo các kích thích thường là đại lượng không điện, chuyển đổi các đại lượng này thành các đại lượng điện và truyền các thông tin về hệ thống đo lường điều khiển, giúp chúng ta nhận dạng đánh giá và điều khiển mọi biến trạng thái của đối tượng.

Hiện nay, con người đã sản xuất được rất nhiều các loại cảm biến khác nhau như: cảm biến ánh sáng, cảm biến độ ẩm, cảm biến nhiệt độ, cảm biến quang...

Các cảm biến có ứng dụng rộng rãi và phạm vi sử dụng lớn, ví dụ như trong công nghiệp, khoa học kỹ thuật, giám sát thiết bị, viễn thông, giao thông, dân dụng, quân sự...

3.2.2 Cảm biến màu sắc TCS3200:



Hình 3.2.2.1: Cảm biến màu sắc TCS3200

Đồ án học phần 1

TCS3200 Color Sensor là một máy dò màu hoàn chỉnh, bao gồm một chip cảm biến TAOS TCS3200 RGB và 4 đèn LED trắng. Các TCS3200 có thể phát hiện và đo lường một phạm vi gần như vô hạn của màu sắc nhìn thấy được. Các ứng dụng bao gồm đọc dải thử, phân loại theo màu sắc, cảm biến và hiệu chuẩn ánh sáng xung quanh, và kết hợp màu, chỉ để chỉ một vài điểm.

TCS3200 có một dãy các bộ lọc quang, mỗi bộ lọc có màu đỏ, xanh lá cây hoặc xanh hoặc không có bộ lọc (rõ ràng). Các bộ lọc của mỗi màu được phân bố đều trong toàn bộ mảng để loại bỏ sự thiên vị vị trí giữa các màu sắc. Bên trong thiết bị là bộ dao động tạo ra một đầu ra của sóng vuông có tần số tỉ lệ với cường độ của màu đã chọn.

- Hoạt động cung cấp duy nhất (2.7V đến 5.5V)
- Chuyển đổi độ phân giải cao sang Tần số
- Màu lập trình được và Tần số đầu ra Quy mô Toàn
- Tính năng tắt nguồn
- Giao tiếp trực tiếp với vi điều khiển
- S0 ~ S1: đầu ra lựa chọn lựa chọn tần số đầu ra
- S2 ~ S3: Đầu vào lựa chọn loại Photodiode
- OUT Pin: Tần số đầu ra
- OE Pin: Tần số đầu ra cho phép pin (hoạt động thấp), có thể là sắp xảy ra khi sử dụng
- Hỗ trợ điều khiển ánh sáng đèn LED
- Kích thước: 28.4 x 28.4mm

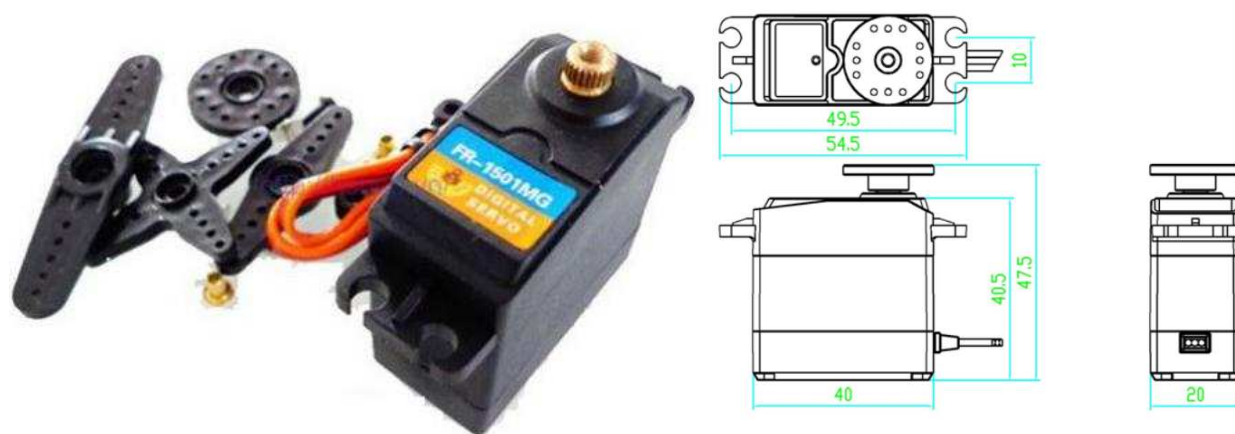
3.3 ĐỘNG CƠ SERVO:

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0° - 180° . Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô hình), có loại thì sở hữu một momen lực vài chục Newton/m, hoặc có loại thì khỏe và không sắc chắc chắn,...

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bất kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.

Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.

3.3.1 Động cơ RC Servo Digital RC FR1510:



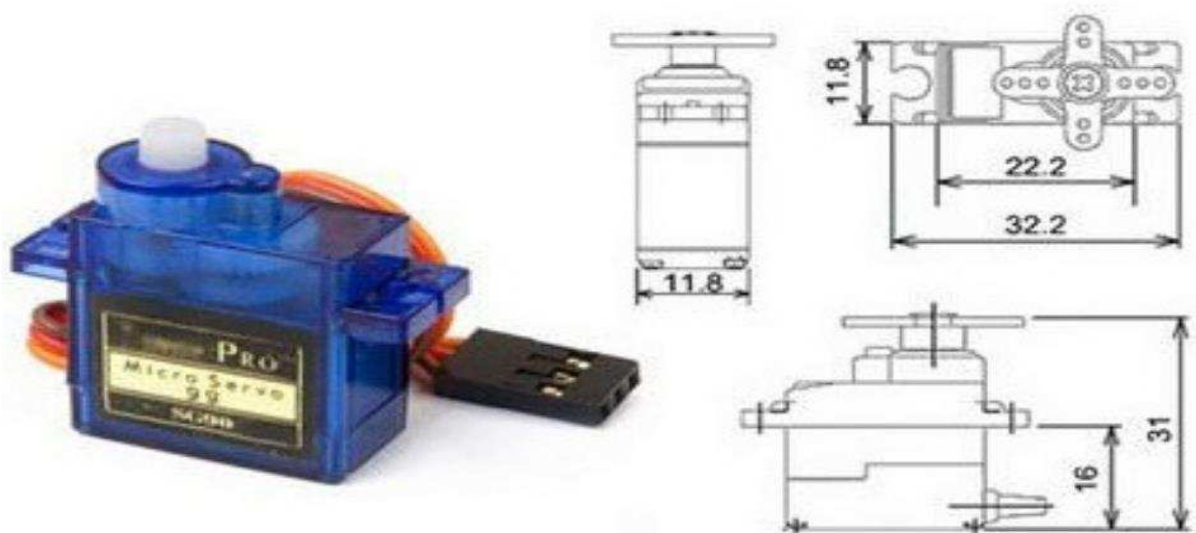
Hình 3.3.1.1: Động cơ RC Servo Digital RC FR1510

Đồ án học phần 1

Động cơ RC Servo Digital RC FR1510 có cấu tạo một trục xoay giống như servo truyền thống giúp bạn dễ ứng dụng cho các thiết kế robot của mình, ngoài ra chất lượng của loại động cơ này rất tốt, động cơ này có bánh răng kim loại, lực kéo mạnh, xoay êm, không rung, giữ vị trí tốt

- Trọng lượng: 60g.
- Kích thước: 40*20*40.5mm.
- Tốc độ: 0.16s/60° tại 4.8VDC đến 7.2VDC.
- Lực kéo: 17kg/cm tại 4.8VDC đến 7.2VDC.
- Dòng điện tiêu thụ: >600mA.
- Chiều dài cáp: 32cm.

3.3.2 Động cơ RC Servo 9G:



Hình 3.3.2.1: Động cơ RC Servo 9G.

Động cơ RC Servo 9G có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng. Động cơ RC Servo 9G có tốc độ phản ứng nhanh, động cơ RC Servo 9G có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

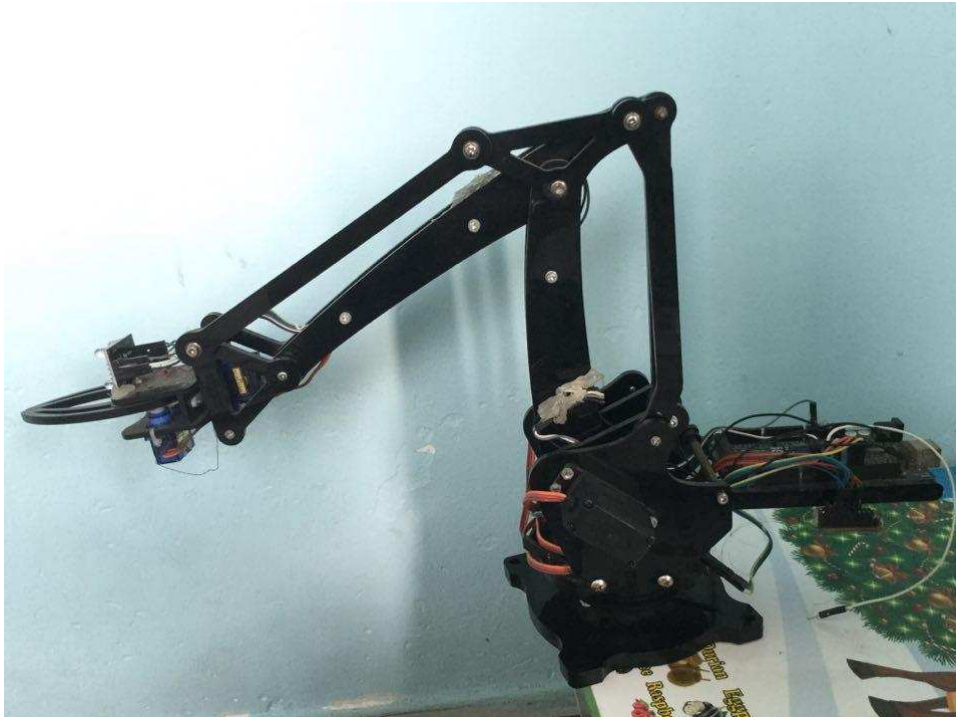
Đồ án học phần 1

Lưu ý: Các bánh răng được làm bằng nhựa nên cần lưu ý khi nâng tải nặng vì có thể làm hư bánh răng.

THÔNG SỐ KỸ THUẬT:

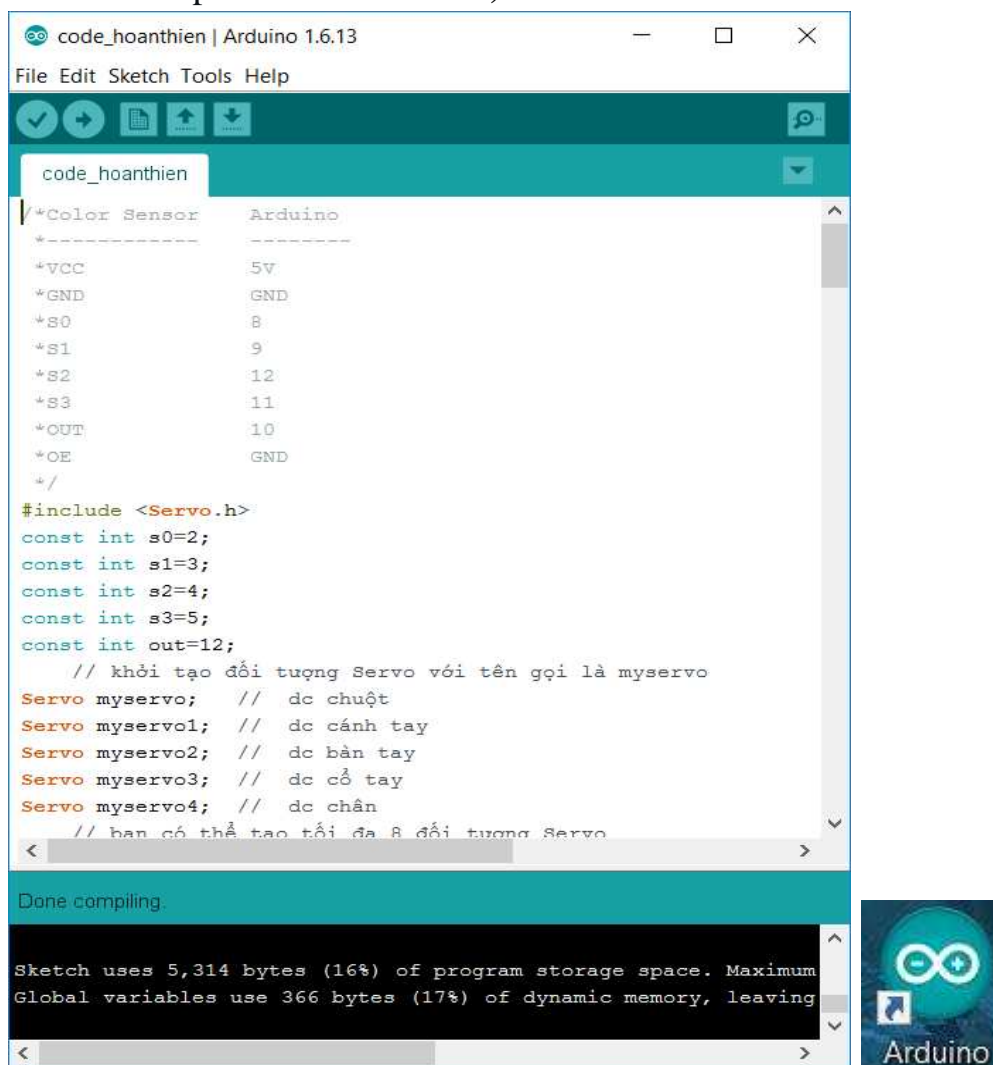
- Khối lượng : 9g
- Kích thước: 22.2×11.8.32 mm
- Momen xoắn: 1.8kg/cm
- Tốc độ hoạt động: 60 độ trong 0.1 giây
- Điện áp hoạt động: 4.8V(~5V)
- Nhiệt độ hoạt động: 0 °C – 55 °C

3.4 KHUNG CÁNH TAY ROBOT:



Hình 3.4.1: Khung robot phân loại sản phẩm theo màu sắc.

3.5 GIỚI THIỆU PHẦN MỀM IDE: (Integrated Development Environment)



Hình 3.5.1: Giao diện phần mềm IDE.

Môi trường phát triển tích hợp (IDE) của Arduino là một ứng dụng cross-platform (nền tảng) được viết bằng Java, và từ IDE này sẽ được sử dụng cho Ngôn ngữ lập trình xử lý (Processing programming language) và project Wiring. Nó được thiết kế để dành cho những người mới tập làm quen với lĩnh vực phát triển phần mềm. Nó bao gồm một chương trình code editor với các chức năng như đánh dấu cú pháp, tự động brace matching, và tự động canh lề, cũng như compile(biên dịch) và upload chương trình lên board chỉ với 1 cú nhấp chuột. Một chương trình hoặc code viết cho Arduino được gọi là một *sketch*.

Đồ án học phần 1

Các chương trình Arduino được viết bằng C hoặc C++. Arduino IDE đi kèm với một thư viện phần mềm được gọi là "Wiring", từ project Wiring gốc, có thể giúp các thao tác input/output được dễ dàng hơn. Người dùng chỉ cần định nghĩa 2 hàm để tạo ra một chương trình vòng thực thi (cyclic executive) có thể chạy được:

- `setup()`: hàm này chạy mỗi khi khởi động một chương trình, dùng để thiết lập các cài đặt
- `loop()`: hàm này được gọi lặp lại cho đến khi tắt nguồn board mạch

Một chương trình điển hình cho một bộ vi điều khiển đơn giản chỉ là làm cho một bóng đèn Led sáng/tắt. Trong môi trường Arduino, ta sẽ phải viết một chương trình giống như sau:

```
#define LED_PIN 13|
void setup () {
    pinMode (LED_PIN, OUTPUT); // Đặt chân 13 làm đầu ra digital
}
void loop () {
    digitalWrite (LED_PIN, HIGH); // Bật LED on
    delay (1000); // chờ trong 1 giây (1000 mili giây)
    digitalWrite (LED_PIN, LOW); // Tắt LED off
    delay (1000); // chờ trong 1s
}
```

Hình 3.5.2: Code sáng tắt led trên phần mềm IDE.

Một đặc điểm của hầu hết các board Arduino là chúng có một đèn LED và điện trở nối giữa chân 13 với đất; một đặc điểm thuận tiện cho nhiều ứng dụng đơn giản.

Đoạn code ở trên không thể đọc được bởi một compiler C++ chuẩn như là một chương trình đúng, vì vậy khi ta click vào nút "Upload to I/O board" trong IDE này, một bản copy của đoạn code này sẽ được ghi vào một file tạm với một

Đồ án học phần 1

extra include header ở phía trên cùng và một hàm main () đơn giản nằm ở phía đáy, để làm cho thàn một chương trình C++ khả dụng.

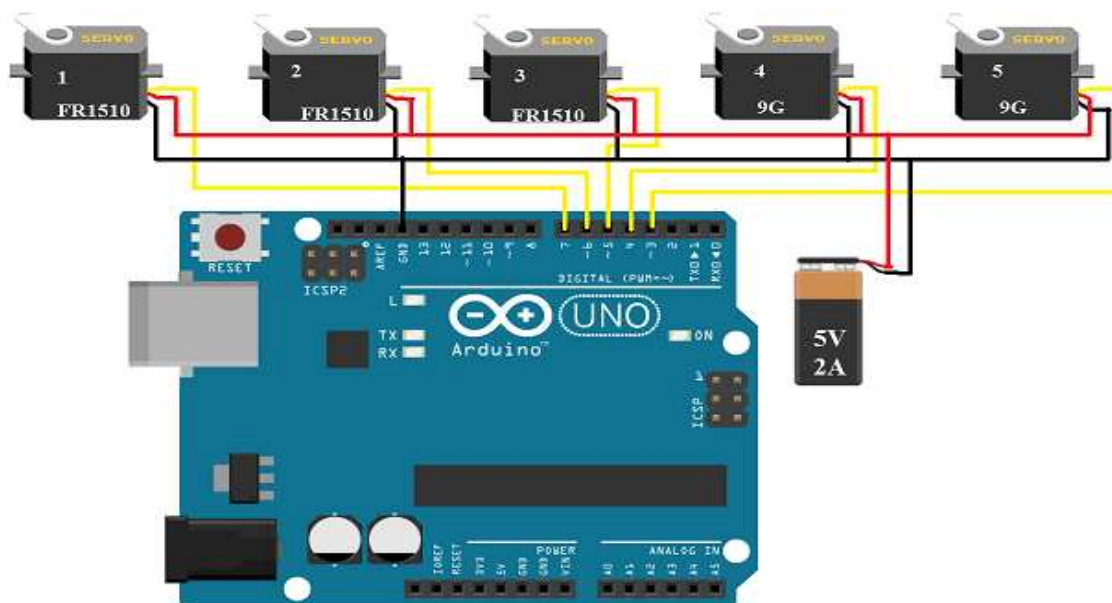
Arduino IDE này sử dụng GNU toolchain và AVR Libc để biên dịch chương trình, và sử dụng avrdude để upload chương trình lên board.

Vì nền tảng của Arduino là các vi điều khiển của Atmel, cho nên môi trường phát triển của Atmel, AVR Studio hoặc các phiên bản Atmel Studio mới hơn, cũng có thể được sử dụng để làm phần mềm phát triển cho Arduino.

CHƯƠNG 4: KẾT NỐI LINH KIỆN, NGUYÊN LÝ HOẠT ĐỘNG VÀ LẬP TRÌNH

4.1. KẾT NỐI LINH KIỆN:

4.1.1. Kết nối Servo với Arduino:



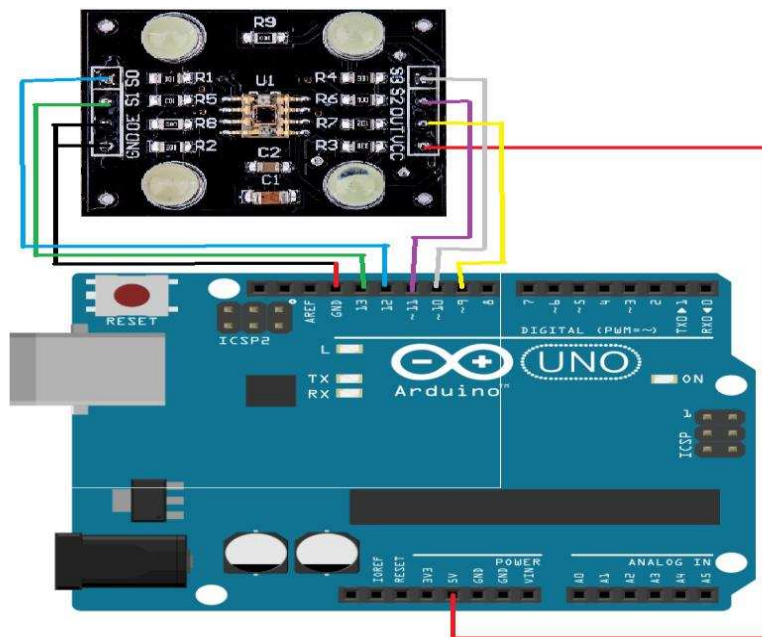
Hình 4.1.1: Kết nối Servo với arduino.

	ARDUINO UNO	PIN
SERVO 1	Dây cam -7, dây đen-GND	Dây đỏ -VCC
SERVO 2	Dây cam -6, dây đen-GND	Dây đỏ -VCC
SERVO 3	Dây cam -5, dây đen-GND	Dây đỏ -VCC
SERVO 4	Dây cam -4, dây đen-GND	Dây đỏ -VCC
SERVO 5	Dây cam -3, dây đen-GND	Dây đỏ -VCC
PIN	GND	

Bảng 4.1.1: Kết nối servo với arduino.

Đồ án học phần 1

4.1.2. Kết nối Cảm biến màu sắc với Arduino:

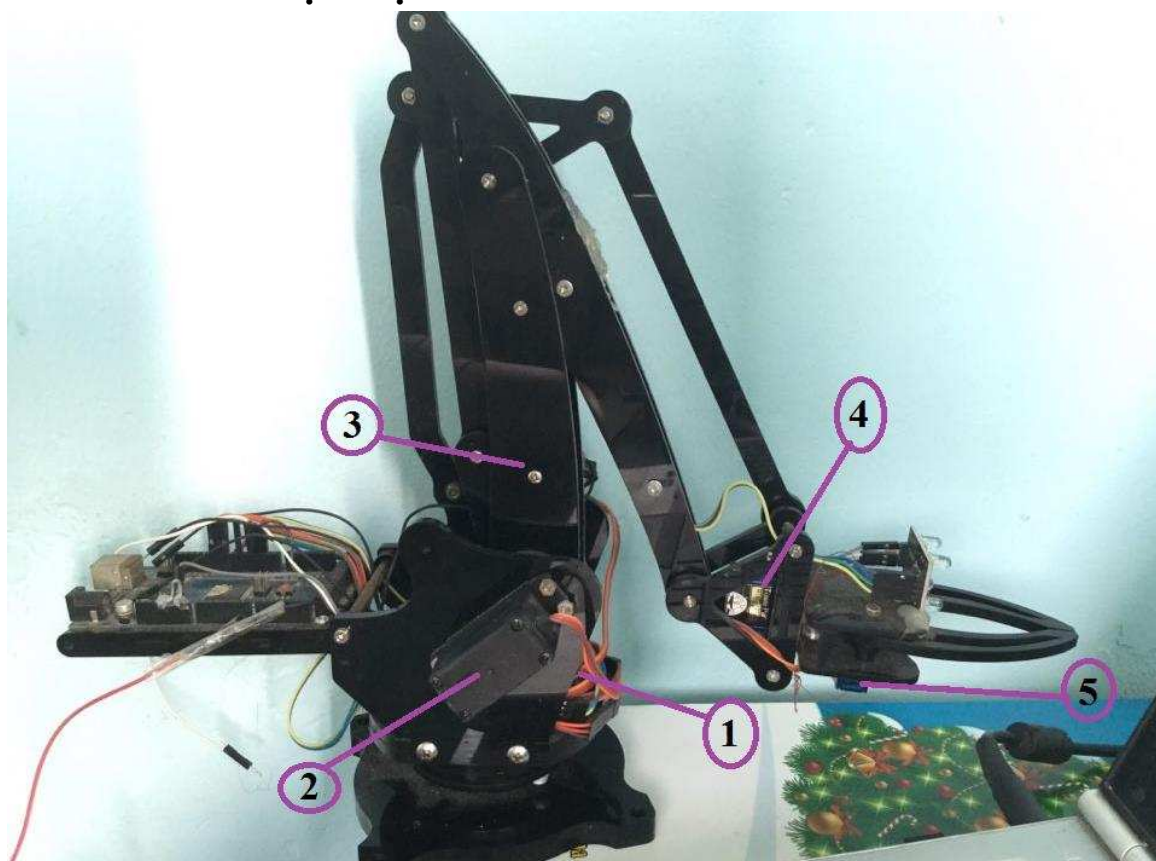


Hình 4.1.2: Kết nối cảm biến màu sắc với arduino.

TCS3200	ARDUINO UNO
VCC	5V
GND	GND
S0	12
S1	13
S2	11
S3	10
OUT	9
OE	GND

Bảng 4.1.2: Kết nối cảm biến màu sắc với arduino.

4.2. NGUYÊN LÝ HOẠT ĐỘNG:



Hình 4.2.1: Vị trí servo trên cánh tay robot.

- Đầu tiên, khi khởi động cánh tay robot được set về vị trí ban đầu.
- Động cơ Servo(sv) 1 hoạt động, làm cánh tay xoay 1 góc 180 độ đến (vị trí làm việc 1).
- Khi đến vị trí làm việc 1, động cơ sv2 và sv3 hoạt động đưa cánh tay về phía trước (vị trí làm việc 2), đồng thời sv4 và sv5 hoạt động đưa kẹp gấp về vị trí như hình, và kẹp gấp được mở ra nhờ sv5.
- Khi đến vị trí làm việc 2, cảm biến màu sắc được hoạt động. Khi này sản phẩm có các màu phân biệt là đỏ, xanh lá và vàng được cảm biến phân biệt rồi đưa dữ liệu về arduino xử lý.
- Khi arduino xử lý nhận được màu sắc từ cảm biến sẽ đưa ra tín hiệu điều khiển đến các động cơ. Tùy theo màu mà cánh tay sẽ đưa sản phẩm đến 1 vị trí khác nhau:

- Màu đỏ : vị trí 3.
 - Màu xanh lá : vị trí 4.
 - Màu vàng : vị trí 5.
- Chú ý: vị trí làm việc là của động cơ, vị trí là nơi đã được xác định để cánh tay đến.
 - Khi nhận được tín hiệu màu rồi, sv5 sẽ hoạt động kẹp chặt vật lại. Sv 2 và sv3 sẽ hoạt động đưa sản phẩm lên cao(vị trí làm việc 6).
 - Đến vị trí làm việc 6, sv1 sẽ hoạt động, tùy theo màu sắc mà cảm biến nhận được mà sv1 đưa cánh tay xoay đến vị trí 3, vị trí 4 hay vị trí 5.
 - Khi đến các vị trí làm việc 3,4 hoặc 5. Cánh tay được đưa về phía trước nhờ động cơ sv2 và sv3(vị trí làm việc 2).
 - Khi đến vị trí làm việc 2, sv 5 sẽ hoạt động thả vật ra . Rồi sv2 và sv3 hoạt động đưa cánh tay về vị trí làm việc 6, rồi sv5 hoạt động kẹp lại.
 - Đến vị trí làm việc 6 của động cơ sv2 và sv3, sv1 sẽ hoạt động đưa cánh tay đến vị trí làm việc 1.
 - Tiếp theo sẽ là vòng lặp từ vị trí làm việc 1 của các động cơ đến khi hết sản phẩm, khi đó cánh tay luôn ở vị trí làm việc chờ. Đến khi cảm biến nhận được dữ liệu thì lại đưa cánh tay đến vị trí làm việc 2.
 - Trước khi tắt máy nên nhấn nút Reset, động cơ sẽ về vị trí ban đầu.

4.3. LẬP TRÌNH:

```
/*Color Sensor      Arduino
 *-----
 */
#include <Servo.h> |
const int s0=12;
const int s1=13;
const int s2=11;
const int s3=10;
const int out=9;
// khởi tạo đối tượng Servo với tên gọi là myservo
Servo myservo;    // dc chuột
Servo myservo1;   // dc cánh tay
Servo myservo2;   // dc bàn tay
Servo myservo3;   // dc cổ tay
Servo myservo4;   // dc chân
// bạn có thể tạo tối đa 8 đối tượng Servo
// LED pins connected to Arduino
int redLed=A0;
int greenLed=A1;
int blueLed=A2;
// Variables
int red=0;
int green =0;
int blue  =0;
int gr=0;

int r=0;
int b=0;
// biến pos dùng để lưu tọa độ các Servo
int pos = 0;
int m=70;
bool isColor=false;
unsigned long interval;
unsigned long intervalr;
unsigned long intervalgr;
unsigned long intervalb;
void setup() {
    // put your setup code here, to run once sensor
    Serial.begin(9600);
    pinMode(s0,OUTPUT);
    pinMode(s1,OUTPUT);
    pinMode(s2,OUTPUT);
    pinMode(s3,OUTPUT);
    pinMode(out,INPUT);
    pinMode(redLed,OUTPUT);
    pinMode(greenLed,OUTPUT);
    pinMode(blueLed,OUTPUT);
    digitalWrite(s0,HIGH);
    digitalWrite(s1,HIGH);
    // arm
```

```
// arm
pinMode(7, OUTPUT);
pinMode(6, OUTPUT);
pinMode(5, OUTPUT);
pinMode(4, OUTPUT);
pinMode(3, OUTPUT);
myservo.attach(7); // attach (đính) servo ở chân digital 9
myservo1.attach(6); // bạn có thể thay bằng bất cứ chân digital/analog nào khác
myservo2.attach(5);
myservo3.attach(4);
myservo4.attach(3);
}

void loop() {
    // put your main code here, to run repeatedly:
    myservo4.write(70);
    myservo.write(106);
    myservo1.write(2);
    myservo2.write(160);
    mouselen();
    armxuong();
    while(!isColor){
        color();
    }
    if(isColor){
        nam();
    }

    armlen();
    xoay();
    armxuong();
    tha();
    armlen();
    xoayve();
    mousexuong();
    }
    isColor=false;
}

void color()
{
    digitalWrite(s2, LOW);
    digitalWrite(s3, LOW);
    //count OUT, pRED, RED
    red=pulseIn(out, digitalRead(out)==HIGH?LOW:HIGH);
    digitalWrite(s3, HIGH);
    //count OUT, pBLUE, BLUE
    blue=pulseIn(out, digitalRead(out)==HIGH?LOW:HIGH);
    digitalWrite(s2, HIGH);
    //count OUT, pGREEN, GREEN
    green=pulseIn(out, digitalRead(out)==HIGH?LOW:HIGH);
    Serial.print("R Intensity: ");
    Serial.print(red, DEC);
```

```
Serial.print("    G Intensity: ");
Serial.print(green, DEC);
Serial.print("    B Intensity: ");
Serial.print(blue, DEC);
if (red < blue && red < green && red > 50 && red < 70)
{
    Serial.print(" - (Red Color)");
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    digitalWrite(blueLed, LOW);
    delay(100);
    if (r > 0 && (millis() - intervalr > 500)) {r = 0;}
    r++;
    if (r == 5) {isColor = true; r = 0; m = 120;}
    intervalr = millis();
}
if (blue < red && blue < green && blue > 40 && blue < 100)
{
    Serial.print(" - (Blue Color)");
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, LOW);
    digitalWrite(blueLed, HIGH);
    delay(100);

    if (b > 0 && (millis() - intervalb > 500)) {b = 0;}
    b++;
    if (b == 5) {
        isColor = true; b = 0; m = 150;}
    intervalb = millis();
}
if (green < red && green < blue && green > 8 && green < 100)
{
    Serial.println(" - (Green Color)");
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    digitalWrite(blueLed, LOW);
    delay(100);
    if (gr > 0 && (millis() - intervalgr > 500)) {gr = 0;}
    gr++;
    if (gr == 5) {
        isColor = true; gr = 0; m = 180;}
    intervalgr = millis();
}
else{
    Serial.println();
}
delay(100);
digitalWrite(redLed, LOW);
```

Đồ án học phần 1

```
digitalWrite(greenLed,LOW);
digitalWrite(blueLed,LOW);
}
/*****
Servo: DC chuột / servo1: DC cánh tay / servo2: DC bàn tay / servo3: cổ tay / servo4: chân
*****/
void mouselen(){
  for(int i=106;i<160;i+=1)
  {
    myservo.write(i);      //Servo: DC chuột lên
    delay(20);
  }
}
void mousexuong(){
  for(int i=160;i>106;i-=1)
  {
    myservo.write(i);      //Servo: DC chuột xuống
    delay(20);
  }
}
void armlen(){
  for(int k=90;k>=2;k-=1)
  {
    myservo1.write(k);     // servo1: DC cánh tay lên
    delay(25);
  }
}
void armxuong(){
  for(int k=2;k<=90;k+=1)
  {
    myservo1.write(k);     // servo1: DC cánh tay xuống
    delay(25);
  }
}
void xoay(){
  for(int j=70;j<=m;j+=1)
  {
    myservo4.write(j);     // servo4: chân xoay
    delay(30);
  }
}
void xoayve(){
  for(int j=m;j>=70;j-=1)
  {
    myservo4.write(j);     // servo4: chân xoay
    delay(30);
  }
}
```

Đồ án học phần 1

```
void nam() {  
    for(int h=160;h>90;h-=1)  
    {  
        myservo2.write(h);          // servo2: DC bàn tay nắm  
        delay(20);  
    }  
}  
void tha() {  
    for(int h=90;h<160;h+=1)  
    {  
        myservo2.write(h);          // servo2: DC bàn tay thả  
        delay(20);  
    }  
}
```

Kết quả: chương trình đã chạy được và không bị lỗi. Vì chỉ làm trên mặt lý thuyết nên khả năng sai khi chạy thực tế vẫn sẽ có... và tùy vào độ sáng của môi trường nên cảm biến màu sắc có thể sai.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT LUẬN:

5.1.1. Những nhiệm vụ đã thực hiện:

- Tìm hiểu được quá trình hình thành và phát triển của Robot.
- Tìm hiểu về arduino.
- Nghiên cứu, tìm hiểu phân cứng của đề tài.
- Suru tầm, tìm kiếm tài liệu tham khảo phục vụ cho việc nghiên cứu.
- Nghiên cứu, học hỏi các phương pháp lập trình arduino.
- ...

5.1.2. Những thuận lợi và khó khăn khi thực hiện đề tài:

Thuận lợi:

- Được thầy và các bạn hỗ trợ hết mình.
- Được tiếp xúc với môi trường năng động ở xưởng thực hành làm tăng niềm đam mê tìm tòi học hỏi.
- ...

Khó khăn:

- Vì vừa nghiên cứu và phải đảm bảo việc học trên trường nên sự phân bố thời gian chưa được nhiều
- Tài liệu chủ yếu về đề tài chủ yếu là tiếng anh nên việc dịch có những chỗ chưa chuẩn xác.
- ...

5.2. HƯỚNG PHÁT TRIỂN ĐỀ TÀI:

Với những gì mà em tìm hiểu được, có thể phát triển hơn nữa đề tài cánh tay robot phân loại sản phẩm theo màu thành nhiều phiên bản hơn nữa. ví như cánh tay robot phân loại sản phẩm theo kích thước hay nhiệt độ.

Ngoài việc phân loại sản phẩm đề tài này có thể phát triển thành cánh tay robot dùng để gấp linh kiện, hay hàn, cắt hay khoan...

Ngoài những hướng phát triển mà em nghĩ mình có khả năng có thể làm được trong tương lai với lượng kiến thức đã nghiên cứu, và cần phải tìm hiểu hơn nữa. Vì Robot phát triển đa dạng nên mong thầy có thể bổ sung và hướng dẫn em nhiều hơn nữa.

Đồ án học phần 1

TÀI LIỆU THAM KHẢO:

[1]<https://arduinoockc.blogspot.com/2016/10/arduino-loai-bo-ieu-khien-on.html>

[2]https://en.wikipedia.org/wiki/Robotic_arm

[3]<http://arduino.vn/bai-viet/602-canh-tay-robot-4-bac-tu-do-dieu-khien-bang-may-tinh-dung-labview>

[4]<http://tailieu.vn/doc/ky-thuat-robot-chuong-1-tong-quan-ve-robot-695344.html>