

ĐẶNG VĂN THÁI

NGÀNH CNKT ĐIỆN TỬ VIỄN THÔNG

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH CNKT ĐIỆN TỬ VIỄN THÔNG
SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI SERVO ĐỂ
ĐIỀU KHIỂN VÀ PHÂN LOẠI

CBHD: TS. Phạm Xuân Thành

Sinh viên: Đặng Văn Thái

Mã số sinh viên: 2018605396

Hà Nội - 2022

LỜI CẢM ƠN

Trước tiên em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất của mình tới TS. Phạm Xuân Thành, người đã hướng dẫn tận tình và hiệu quả, thường xuyên động viên chúng em trong quá trình hoàn thiện đề tài. Người đã dành cho em sự ưu ái nhất trong thời gian học tập, nghiên cứu cũng như quá trình hoàn thành thực tập tốt nghiệp.

Em xin cảm ơn các Thầy giáo, Cô giáo trong khoa Điện Tử trường Đại học Công Nghiệp Hà Nội cùng tất cả thành viên lớp Điện tử 04 – K13 đã tạo điều kiện và đóng góp ý kiến để em hoàn thành tốt đồ án tốt nghiệp.

Mặc dù em đã cố gắng để hoàn thành thực tập nhưng do kiến thức cũng như khả năng còn hạn hẹp nên quá trình thực hiện đề tài còn có sai sót. Rất mong nhận được sự góp ý và chỉ bảo của quý thầy cô.

Em xin chân thành cảm ơn!

Hà Nội, Ngày... tháng ... năm 2022

Sinh viên thực hiện

Đặng Văn Thái

MỤC LỤC

LỜI CẢM ƠN	I
DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU	IV
DANH MỤC TỪ VIẾT TẮT.....	V
LỜI MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI SERVO ĐỂ ĐIỀU KHIỂN VÀ PHÂN LOẠI	3
1.1 Đặt vấn đề	3
1.2 Tổng quan về đề tài.....	3
1.3 Tổng quan về các hệ thống phân loại trên thị trường.....	4
1.4 Kết luận chương 1	7
CHƯƠNG 2: CÁC CÔNG NGHỆ SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI, SERVO ĐỂ ĐIỀU KHIỂN VÀ PHÂN LOẠI.....	8
2.1 Giới thiệu về Arduino Nano	8
2.2 Module cảm biến vật cản hồng ngoại.....	11
2.3 Động cơ RC Servo Futaba S3003	12
2.4 Băng tải	14
2.5 Module L298N.....	16
2.6 Ngôn ngữ lập trình Python và Open CV	17
2.7 Phần mềm thiết kế Altium Designer	18
2.8 Phần mềm Arduino IDE.....	20
2.9 Kết luận chương 2.....	22
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI SERVO ĐỂ ĐIỀU KHIỂN VÀ PHÂN LOẠI.....	23

3.1 Sơ đồ khối của hệ thống	23
3.2 Sơ đồ nguyên lý toàn mạch	24
3.2.1 Khối nguồn và ổn áp nguồn	25
3.2.2 Khối xử lý trung tâm	25
3.2.3 Khối cảm biến	27
3.2.4 Khối servo	27
3.2.5 Khối đầu ra.....	28
3.2.6 Khối điều khiển động cơ băng tải	28
3.3 Thiết kế phần cứng	29
3.4 Thiết kế phần mềm	30
3.4.1 Lưu đồ thuật toán	30
3.4.2 Phần mềm đọc mã QR	31
3.5 Mô hình thực tế	32
KẾT LUẬN	33
TÀI LIỆU THAM KHẢO	35
PHỤ LỤC	36

DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU

Hình 1.1: Một hệ thống phân loại sản phẩm	4
Hình 1.2: Cảm biến màu Autonics BC15-LDT-C-P.....	5
Hình 1.3: Camera công nghiệp trong phân loại sản phẩm.....	5
Hình 1.4: Hệ thống phân loại sản phẩm theo chiều cao.....	6
Hình 1.5: Phân loại sản phẩm theo hình dạng	6
Hình 2.1: Arduino Nano.....	10
Hình 2.2: Module cảm biến vật cản hồng ngoại	11
Hình 2.3: Servo Futaba S3003	12
Hình 2.4: Băng tải	14
Hình 2.5: Module L298N.....	16
Hình 2.6: Python và OpenCV	17
Hình 2.7: Altium Designer 2022.....	18
Hình 2.8: Giao diện Arduino IDE.....	20
Hình 3.1: Sơ đồ khối của hệ thống.....	23
Hình 3.2: Sơ đồ nguyên lý của hệ thống.....	24
Hình 3.3: Khối nguồn và ổn áp nguồn	25
Hình 3.4: Khối MCU.....	26
Hình 3.5. Khối cảm biến	27
Hình 3.6: Khối Servo	27
Hình 3.7: Khối đầu ra.....	28
Hình 3.8: Khối điều khiển động cơ băng tải	28
Hình 3.9: Sơ đồ mạch in của hệ thống	29
Hình 3.10: Lưu đồ thuật toán	30
Hình 3.11: Giao diện đọc mã QR.....	31
Hình 3.12: Mô hình thực tế và khối điều khiển	32
Hình 3.13: Tổng quát mô hình thực tế	32

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Tiếng anh	Tiếng việt
1	PWM	Pulse-width modulation	Điều chế độ rộng xung
2	RAM	Random Access Memory	Bộ nhớ khả biến
4	DC	Direct Current	Dòng điện một chiều
5	LCD	Liquid Crystal Display	Màn hình tinh thể lỏng
6	IDE	Integrated Development Environment	Môi trường phát triển tích hợp

LỜI MỞ ĐẦU

Ngày nay, xã hội ngày càng phát triển, công nghiệp hóa, hiện đại hóa ngày càng được nâng cao để phát triển đất nước và cải thiện cuộc sống của người dân. Vì vậy việc ứng dụng khoa học kỹ thuật ngày càng rộng rãi, phổ biến và mang lại hiệu quả cao trong hầu hết các lĩnh vực.

Xét điều kiện cụ thể ở nước ta trong công cuộc công nghiệp hóa, hiện đại hóa sử dụng ngày càng nhiều thiết bị hiện đại để điều khiển tự động các quá trình sản xuất, gia công và chế biến sản phẩm... Và một khâu quan trọng ảnh hưởng đến chất lượng sản phẩm đó là phân loại sản phẩm. Hệ thống phân loại sản phẩm nhằm chia sản phẩm ra các nhóm có cùng thuộc tính với nhau để thực hiện đóng gói hay loại bỏ sản phẩm hỏng. Hiện nay để phân loại sản phẩm người ta thường sử dụng các loại cảm biến với các chức năng khác nhau để phân loại sản phẩm theo mong muốn như cảm biến phân loại theo cân nặng, màu sắc, ...

Nếu như phương pháp phân loại sản phẩm truyền thống yêu cầu không gian làm việc rộng hơn cho số lượng người tham gia phân loại lớn, thời gian phân loại lâu và dễ sai sót thì nay, với dây chuyền phân loại sản phẩm tự động nhờ camera, cảm biến cân nặng... số lượng nhân công đã giảm xuống đáng kể (tới 80%) khi năng suất tăng lên từ 3-5 lần cùng tỷ lệ nhầm lẫn, sai sót được kiểm soát.

Qua quá trình học tập và tìm hiểu một số môn học ở trên trường, em đã quyết định chọn đề tài: **“Sử dụng cảm biến quét mã QR, bằng tải servo để điều khiển và phân loại”** là đề tài đồ án tốt nghiệp.

Trong đề tài này, em sẽ thực hiện theo dạng mô hình mô phỏng dây chuyền băng tải phân loại sản phẩm theo dạng đọc mã QR. Thông qua việc xử lý mã QR để nhận biết được cân nặng của các sản phẩm, từ đó vi điều khiển điều khiển cơ cấu phân loại sản phẩm vào từng ngăn phân loại theo cân nặng của từng loại

Câu hỏi đặt ra là: Tại sao chúng ta cần phải hệ thống phân loại sản phẩm? Bởi vì đối với phương pháp phân loại sản phẩm truyền thống yêu cầu không gian rộng hơn vì số lượng người tham gia vào công đoạn phân loại khá lớn, thời gian phân loại sản phẩm lâu, dễ xảy ra sai sót. Với sự ra đời và phổ biến rộng rãi của băng tải công nghiệp, chúng ta gần như có thể tối ưu hóa mọi lĩnh vực, trong đó có việc chế biến và phân loại nông sản. Thông qua việc tự động hóa ở các khâu, sử dụng những công nghệ, sản phẩm của khoa học kỹ thuật, băng tải công nghiệp có thể giảm thiểu sai sót, giảm bớt nhân công, cho thời gian làm việc liên tục cũng như hiệu suất làm việc cao.

Ý nghĩa

- Nâng cao chất lượng sản phẩm khi đưa ra thị trường.
- Giúp quản lý sản phẩm sản xuất được.
- Giảm kinh phí, tăng năng suất lao động.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI SERVO ĐỂ ĐIỀU KHIỂN VÀ PHÂN LOẠI

1.1 Đặt vấn đề

Ngày nay, cùng với sự phát triển của nền kinh tế thị trường, công nghiệp hóa, hiện đại hóa của nước ta cũng đang phát triển mạnh mẽ. Trên cơ sở những tiềm năng và chiến lược phát triển mà nhà nước đã đề ra, Việt Nam ngày càng khẳng định được vị thế ở cả thị trường trong nước và quốc tế, trở thành một trong những nước chiếm tỉ trọng lớn trong ngành xuất khẩu, đóng góp một phần không nhỏ vào nền kinh tế quốc dân.

Các loại sản phẩm và mặt hàng luôn luôn cần được phân loại thành loại 1, loại 2 ... với những tiêu chuẩn khác nhau về trọng lượng, màu sắc, hình dáng ... để dễ dàng trong việc quản lý và sản xuất. Nhưng với phương pháp phân loại bằng tay truyền thống thì mất rất nhiều thời gian và công sức nhưng có thể không đáp ứng chính xác các yêu cầu tiêu chuẩn được.

Với mục đích có thể tạo ra một hệ thống phân loại sản phẩm theo dây chuyền, thân thiện với người dùng, khả năng tùy biến cao và giá thành phải chăng, em xin giới thiệu đề tài: **“Sử dụng cảm biến quét mã QR, băng tải servo để điều khiển và phân loại”**. Với mô hình này, chúng ta có thể phát triển thêm thành một hệ thống hoàn chỉnh, có thể giúp con người dễ dàng sử dụng, vận hành cũng như điều chỉnh, thay thế lao động chân tay giúp giảm nhân công, tiền bạc, tăng năng suất làm việc, đồng thời giảm bớt chi phí lắp đặt và vận hành khi phải đầu tư một hệ thống lớn.

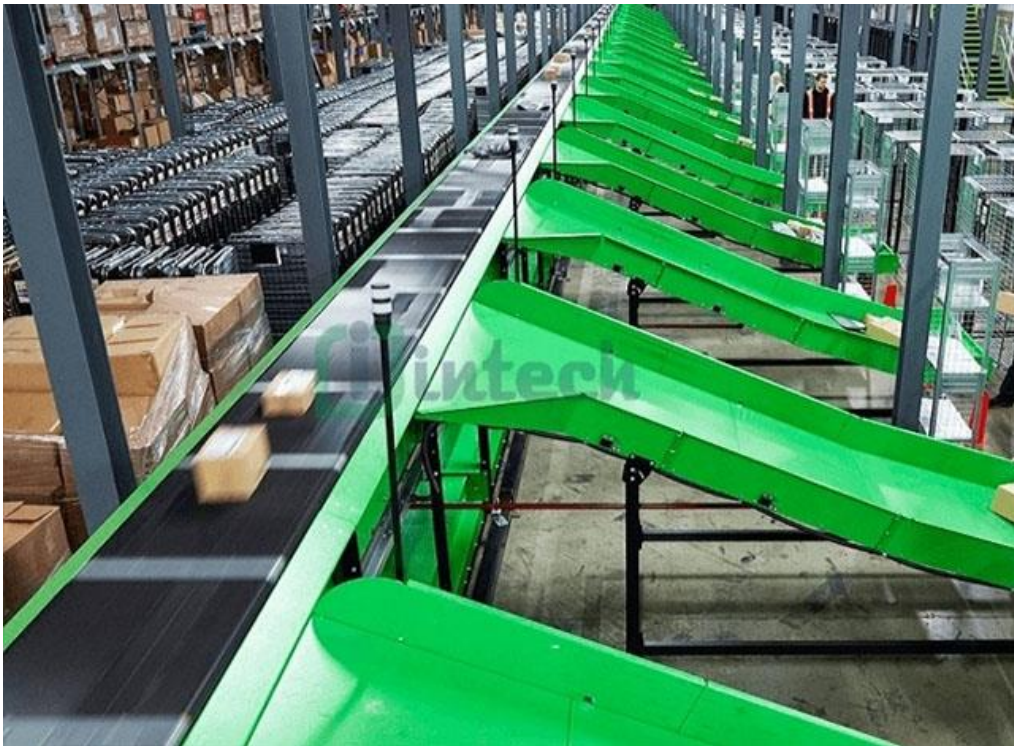
1.2 Tổng quan về đề tài

Trong đề tài này, em sẽ thực hiện theo dạng mô hình mô phỏng dây chuyền băng tải phân loại sản phẩm theo mã QR. Thông qua việc xử lý mã QR, xử lý ảnh mã QR để nhận biết được dữ liệu trong mã QR, từ đó vi điều khiển điều khiển cơ cấu phân loại sản phẩm vào từng ngăn phân loại theo mã QR của từng loại. Kết thúc quá trình có thể phân loại ra được các loại mã QR theo các ô khác nhau.

1.3 Tổng quan về các hệ thống phân loại trên thị trường

Các loại hệ thống phân loại sản phẩm đang hiện hành hiện nay:

Nhằm phục vụ nhiệm vụ hiện đại hóa quy trình sản xuất, hệ thống phân loại sản phẩm ra đời là một công cụ hiệu quả giúp thay thế con người trong công việc phân loại, nó đã góp phần nâng cao hiệu quả trong công việc. Một hệ thống hoàn chỉnh có thể phân loại các sản phẩm với độ tin cậy cao, hoạt động liên tục và giảm tối đa thời gian trì hoãn hệ thống. Hơn thế nữa, đối với những công việc đòi hỏi sự tập trung cao và có tính tuần hoàn, nên các công nhân khó đảm bảo được sự chính xác trong công việc. Điều đó ảnh hưởng trực tiếp đến chất lượng sản phẩm và uy tín của nhà sản xuất. Vì vậy, hệ thống tự động nhận dạng và phân loại sản phẩm ra đời nhằm đáp ứng nhu cầu cấp bách này.



Hình 1.1: Một hệ thống phân loại sản phẩm

Hệ thống phân loại sản phẩm hiện nay có rất nhiều trong ứng dụng thực tế ở nhiều ngành nghề khác nhau. Trên thế giới có rất nhiều hệ thống phân loại sản phẩm như: phân loại sản phẩm theo kích thước, chiều cao, màu sắc...

Hệ thống phân loại sản phẩm theo màu sắc

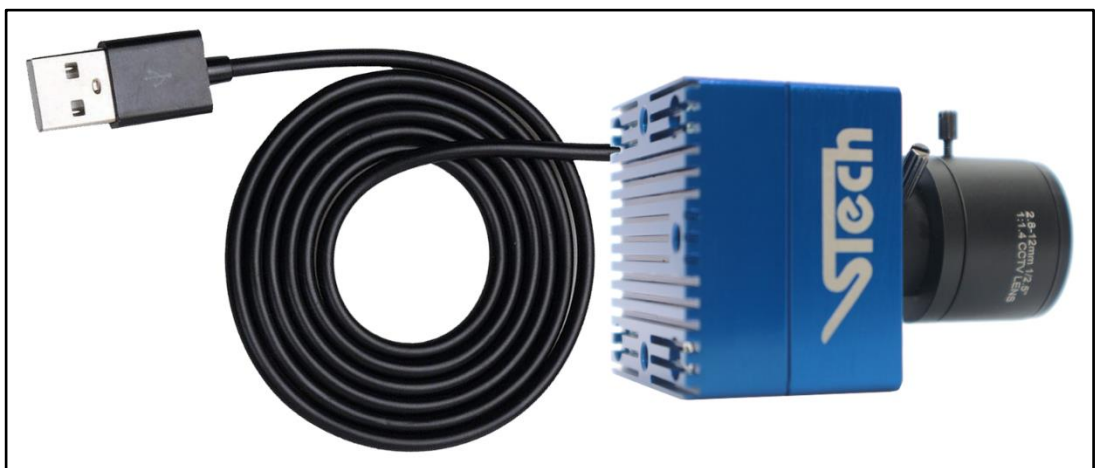


Hình 1.2: Cảm biến màu Autonics BC15-LDT-C-P

Hệ thống phân loại sản phẩm theo màu sắc được ứng dụng rất nhiều trong các dây chuyền sản xuất.

Hệ thống sử dụng các cảm biến có chức năng phát hiện các loại màu khác nhau, từ đó gửi các tín hiệu màu sắc về bộ xử lý trung tâm để xử lý các tín hiệu về màu sắc, từ đó phân loại ra các loại sản phẩm theo màu sắc khác nhau.

Ngoài các cảm biến màu sắc, hệ thống còn có thể dùng các loại camera công nghiệp để phân loại theo màu sắc và các thông số khác.



Hình 1.3: Camera công nghiệp trong phân loại sản phẩm

Ứng dụng phân loại sản phẩm theo màu sắc trong các ngành nông nghiệp, thực phẩm, dầu, hóa chất, công nghiệp dược phẩm, linh kiện điện tử...

Phân loại sản phẩm theo chiều cao



Hình 1.4: Hệ thống phân loại sản phẩm theo chiều cao

Hệ thống phân loại sản phẩm theo chiều cao được ứng dụng rất nhiều trong các ngành công nghiệp:

- Ứng dụng trong các ngành công nghiệp thực phẩm như bánh kẹo, hoa quả...
- Ứng dụng trong công nghiệp sản xuất bia, nước giải khát.

Phân loại sản phẩm theo hình dạng

Phân loại sản phẩm theo hình dạng (hình 1.2)



Hình 1.5: Phân loại sản phẩm theo hình dạng

Ứng dụng:

Hệ thống phân loại sản phẩm theo hình dạng được ứng dụng trong rất nhiều ngành công nghiệp:

- Ứng dụng trong công nghiệp kiểm tra và phân loại sản phẩm có hình dáng khác nhau như: gạch, ngói, thực phẩm tiêu dùng...
- Ứng dụng trong kiểm tra và phân loại Nông Sản.

Như vậy, ngoài những loại hệ thống phân loại sản phẩm trên, còn có hệ thống phân loại sản phẩm khác theo đặc tính của sản phẩm. Ví dụ như phân loại sản phẩm theo trọng lượng, kích thước... Hầu như cấu tạo và nguyên lý hoạt động của chúng khá tương tự nhau, chỉ khác nhau ở bộ phận đẩy sản phẩm phân loại: có thể là xilanh, piston hoặc động cơ bước. Các bộ phận nhận dạng sản phẩm: các loại cảm biến như màu sắc, cảm biến quang thu phát...

1.4 Kết luận chương 1

Dựa vào nhu cầu sử dụng cũng như sự tiện lợi hệ thống phân loại sản phẩm trong thực tiễn ngày càng nhiều, yêu cầu công nghệ được cải tiến ngày càng hiện đại và hệ thống có nhiều ưu điểm nên em đã lựa chọn đề tài này để làm đồ án tốt nghiệp.

CHƯƠNG 2: CÁC CÔNG NGHỆ SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI, SERVO ĐỂ ĐIỀU KHIỂN VÀ PHÂN LOẠI

2.1 Giới thiệu về Arduino Nano

Giới thiệu về Arduino

[5] Arduino thực sự đã gây sóng gió trên thị trường người dùng DIY (là những người tự chế ra sản phẩm của mình) trên toàn thế giới trong vài năm gần đây, gần giống với những gì Apple đã làm được trên thị trường thiết bị di động. Số lượng người dùng cực lớn và đa dạng với trình độ trải rộng từ bậc phổ thông lên đến đại học đã làm cho ngay cả những người tạo ra chúng phải ngạc nhiên về mức độ phổ biến.

Arduino là gì mà có thể khiến ngay cả những sinh viên và nhà nghiên cứu tại các trường đại học danh tiếng như MIT, Stanford, Carnegie Mellon phải sử dụng, hoặc ngay cả Google cũng muốn hỗ trợ khi cho ra đời bộ kit Arduino Mega ADK dùng để phát triển các ứng dụng Android tương tác với cảm biến và các thiết bị khác.

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 18 bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm một cổng giao tiếp USB, sáu chân đầu vào analog, mười bốn 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 nhiệt 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 Arduino bằng ngôn ngữ C hoặc C++.

Một mạch Arduino bao gồm một vi điều khiển AVR với nhiều linh kiện bổ sung giúp dễ dàng lập trình và có thể mở rộng với các mạch khác.

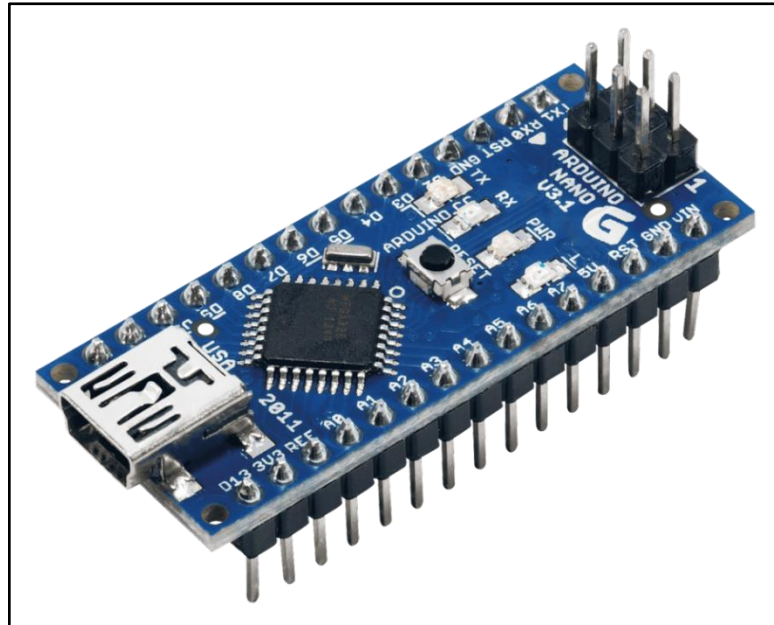
Một khía cạnh quan trọng của Arduino là các kết nối tiêu chuẩn của nó, cho phép người dùng kết nối với CPU của board với các module thêm vào có thể dễ dàng chuyển đổi, được gọi là shield. Vài shield truyền thông với board Arduino trực tiếp thông qua các chân khác nhau, nhưng nhiều shield được định địa chỉ thông qua serial bus I2C - nhiều shield có thể được xếp chồng và sử dụng dưới dạng song song.

Arduino chính thức thường sử dụng các dòng chip megaAVR, đặc biệt là ATmega8, ATmega168, ATmega328, ATmega1280, và ATmega2560. Một vài các bộ vi xử lý khác cũng được sử dụng bởi các mạch Arduino tương thích. Hầu hết các mạch gồm một bộ điều chỉnh tuyến tính 5V và một thạch anh dao động 16 MHz (hoặc bộ cộng hưởng ceramic trong một vài biến thể), mặc dù một vài thiết kế như LilyPad chạy tại 8 MHz và bỏ qua bộ điều chỉnh điện áp onboard do hạn chế về kích cỡ thiết bị.

Một vi điều khiển Arduino cũng có thể được lập trình sẵn với một boot loader cho phép đơn giản là upload chương trình vào bộ nhớ flash on-chip, so với các thiết bị khác thường phải cần một bộ nạp bên ngoài. Điều này giúp cho việc sử dụng Arduino được trực tiếp hơn bằng cách cho phép sử dụng một máy tính gốc như là một bộ nạp chương trình.

Giới thiệu về Arduino Nano

Board Arduino Nano có cấu tạo, số lượng chân vào ra là tương tự như board Arduino Uno tuy nhiên đã được tối giản về kích thước cho tiện sử dụng hơn. Do được tối giản rất nhiều về kích thước nên Arduino Nano chỉ được nạp code và cung cấp điện bằng duy nhất 1 cổng mini USB.



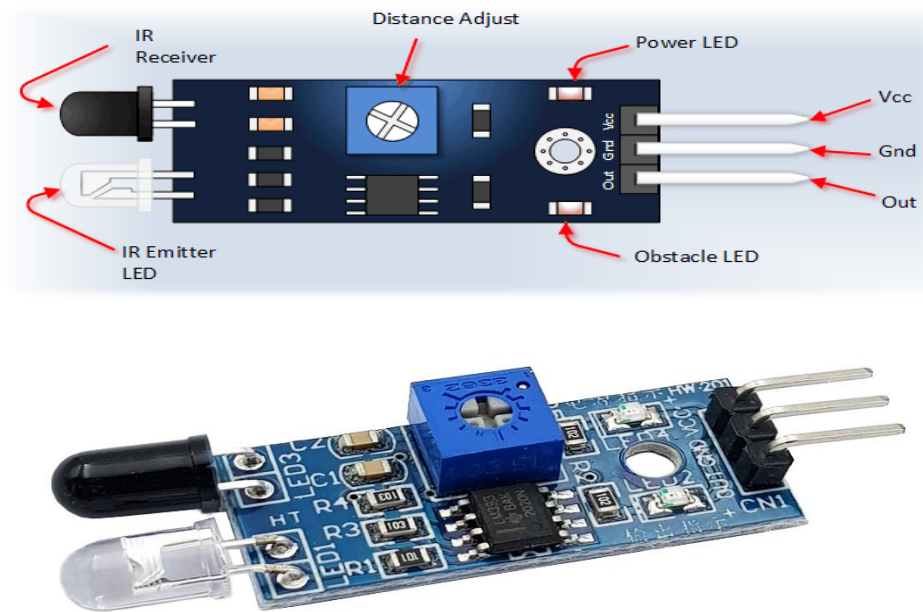
Hình 2.1: Arduino Nano

Thông số kỹ thuật chi tiết:

- Vi xử lý ATmega328 (phiên bản v3.0).
- Điện áp hoạt động 5 V.
- Điện áp đầu vào (khuyến nghị) 7-12 V.
- Điện áp đầu vào (giới hạn) 6-20 V.
- Chân vào/ra số 14 (6 chân có khả năng xuất ra tín hiệu PWM).
- Chân vào tương tự 8.
- Dòng điện mỗi chân vào/ra 40 mA.
- Bộ nhớ 16 KB (ATmega168), 32 KB (ATmega328) trong đó 2 KB dùng để nạp bootloader.
- SRAM 1 KB (ATmega168) hoặc 2 KB (ATmega328).
- EEPROM 512 bytes (ATmega168) hoặc 1 KB (ATmega328).
- Xung nhịp 16 MHz + Kích thước 0.73" x 1.70".

2.2 Module cảm biến vật cản hồng ngoại.

Module cảm biến phát hiện vật cản hồng ngoại có một cặp mắt phát hiện tia hồng ngoại với một tần số nhất định, khi phát hiện ra tia hồng ngoại của một vật cản bề mặt phản xạ sẽ nhận tín hiệu và xử lý đèn báo màu xanh lá cây sẽ sáng đồng thời đầu cho tín hiệu ra ở mức thấp.



Hình 2.2: Module cảm biến vật cản hồng ngoại

Cảm biến vật cản hồng ngoại có khả năng thích nghi với môi trường, có một cặp truyền và nhận tia hồng ngoại. Tia hồng ngoại phát ra một số tần số, khi phát hiện định hướng có vật cản (phản xạ), phản xạ vào đèn thu hồng ngoại, sau khi so sánh, đèn màu xanh sẽ sáng lên, đồng thời bắt đầu tín hiệu đầu số hiệu (một bậc tín hiệu).

Thông số kỹ thuật:

- Sử dụng 1 led phát hồng ngoại và 1 led thu hồng ngoại.
- Biến trở điều chỉnh độ chỉnh VR vuông 10K.
- Điện áp hoạt động: 3.3 - 5VDC.
- Tích hợp biến trở chỉnh khoảng cách nhận biết vật cản.
- Kích thước: 3.2 x 1.4cm.

2.3 Động cơ RC Servo Futaba S3003

Động cơ servo là một thiết bị tự động có sử dụng lỗi cảm biến phản hồi âm để điều chỉnh hành động của một cơ cấu. Động cơ RC Servo là loại động cơ có tốc độ thấp, mô-men xoắn cao và có nhiều kích thước to nhỏ khác nhau tùy vào thiết bị.

Động Cơ Servo Futaba S3003 có các bánh răng được làm bằng nhựa và trọng lượng nhẹ, có cách điều khiển tương tự như các servo phổ biến trên thị trường, thích hợp cho các ứng dụng cần mức chi phí rẻ, trọng lượng nhẹ phù hợp với các dự án vừa và nhỏ

Cấu tạo của động cơ [6]:

- Động cơ chính
- Board điều khiển tín hiệu hồi tiếp
- Dây nguồn (đỏ)
- Dây tín hiệu vào (vàng hoặc trắng)
- Dây mass (đen)
- Volt kế
- Trục/ bánh răng output
- Horn/Wheel/Arm gắn kèm
- Vỏ servo
- Chip điều khiển chính



Hình 2.3: Servo Futaba S3003

Trong đó động cơ và vôn kế được nối với mạch điều khiển tạo thành mạch hồi tiếp vòng kín. Cả mạch và động cơ đều được cấp nguồn DC. Để quay động cơ tín hiệu số được gửi tới mạch điều khiển. Tín hiệu này sẽ khởi động

động cơ thông qua chuỗi bánh răng nối với vôn kế. Vị trí của trục vôn kế cho biết vị trí trục ra của servo. Khi vôn kế đạt được vị trí yêu cầu thì mạch điều khiển sẽ tự động ngắt động cơ.

Thay vì quay liên tục như DC servo, động cơ RC servo được thiết kế quay với giới hạn trong khoảng 90, 180 và 270 độ.

Nguyên lý hoạt động:

Theo như nghiên cứu và giám sát cho thấy động cơ RC servo hoạt động dựa trên nguyên lý PWM. Cụ thể:

Servo sẽ đáp ứng một dãy xung số ổn định. Chi tiết hơn là mạch điều khiển sẽ đáp ứng một số tín hiệu ứng với các xung biến đổi từ 1-2ms. Các xung này sẽ được gửi đi 50 lần/ giây - đồng nghĩa với việc cung cấp xung mỗi 20ms một lần. Động cơ RC servo đòi hỏi 30- 60 xung/giây. Nếu số này quay không đáp ứng đủ điều kiện và quay quá thấp sẽ dẫn đến độ chính xác và công suất servo giảm.

Chiều dài (độ rộng) của các xung sẽ quyết định đến vị trí góc trục của động cơ:

- Độ rộng của xung 1.5ms thì cho trục động cơ quay đến vị trí góc 90 độ
- Độ rộng xung nhỏ hơn 1.5ms thì cho trục động cơ quay ở vị trí góc 0 độ.
- Độ rộng xung lớn hơn 1.5ms thì trục động cơ sẽ quay đến vị trí góc 180 độ.

Các servo khác nhau ở góc quay được cùng với tín hiệu 1 - 2ms thì các servo chuẩn được thiết kế để quay tới và lui từ 90 - 180 độ. Lúc này sẽ được cung cấp toàn bộ chiều dài xung. Hầu hết các servo đều có thể quay được 180 độ hay gần 180 độ.

Nếu cố khiến servo quay quá giới hạn cơ học thì trục ra của động cơ sẽ đụng vật cản bên trong khiến các bánh răng bị mài mòn. Nếu cứ để hiện tượng kéo dài lâu ngày động cơ bánh răng sẽ bị hủy.

Ứng dụng

Các servo RC thường được sử dụng trong các hệ thống cơ khí khác nhau như: Hệ thống lái của xe hơi, các bề mặt điều khiển trên máy bay hay các bánh lái của một chiếc thuyền.

Ngoài ra còn được sử dụng trong các mô hình điều khiển bằng radio

2.4 Băng tải

Băng tải (băng chuyền) là một máy cơ khí dùng để vận chuyển các đồ vật từ điểm này sang điểm khác, từ vị trí A sang vị trí B. Thay vì vận chuyển sản phẩm bằng công nhân vừa tốn thời gian, chi phí nhân công lại tạo ra môi trường làm việc lộn xộn thì hiện nay người ta lựa chọn băng chuyền tải có thể giải quyết điều đó. Băng tải giúp tiết kiệm sức lao động, số lượng nhân công, giảm thời gian và tăng năng suất lao động.



Hình 2.4: Băng tải

Với với hệ thống phân loại sản phẩm thì băng tải giúp vận chuyển các sản phẩm cần phân loại thuận tiện hơn, giúp các quy trình công nghệ và quy trình về tự động hóa trở nên đơn giản hơn. Hình 2.4 đưa ra một phần của băng tải trong các hệ thống thực tế.

Cấu tạo của băng tải bao gồm:

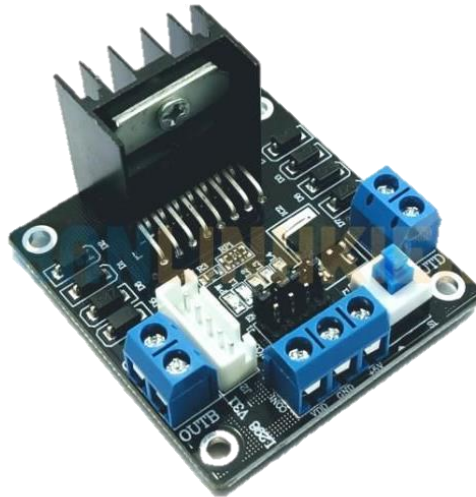
- Khung băng tải: thường được làm bằng nhôm định hình, thép sơn tĩnh điện hoặc inox.
- Dây băng tải
- Động cơ chuyển động
- Bộ điều khiển băng chuyển: thường gồm có biến tần, sensor, timer,...
- Cơ cấu truyền động gồm có: Rulo kéo, con lăn đỡ, không xích...
- Hệ thống bàn thao tác trên băng chuyển thường bằng gỗ, thép hoặc inox trên mặt có dán thảm cao su chống tĩnh điện.
- Hệ thống đường khí nén và đường điện có ổ cắm để lấy điện cho các máy dùng trên băng chuyển.

Ứng dụng:

Băng tải được ứng dụng rộng rãi ở tất cả các ngành sản xuất, nông nghiệp, công nghiệp,... Với hệ thống băng tải giúp tối ưu hóa được chi phí, tiết kiệm thời gian, hạn chế nguồn nhân lực, hiệu quả kinh tế cao đồng thời còn giúp cho hệ thống sản xuất ngày càng được tự động hóa theo hướng hiện đại.

2.5 Module L298N

IC L298 là một IC tích hợp gồm hai mạch cầu H. Với điện áp làm tăng công suất đầu ra từ 5V – 47V, dòng lên đến 2A, L298 rất thích hợp trong những ứng dụng công suất nhỏ.



Hình 2.5: Module L298N

[3] L298 gồm các chân:

- Chân 12V, 5V: 2 chân cấp nguồn trực tiếp đến động cơ.
- Chân GND: GND của nguồn cấp cho động cơ.
- 2 Jump A enable và B enable.
- Gồm có 4 chân Input. IN1, IN2, IN3, IN4.
- Output A: nối với động cơ A.

Thông số kỹ thuật:

- Điện áp điều khiển: +5 V ~ +12 V
- Dòng tối đa cho mỗi cầu H là: 2A
- Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
- Dòng của tín hiệu điều khiển: 0 ~ 36mA

Ứng dụng

Module L298 có thể điều khiển nhiều loại motor khác nhau như step motor, servo motor, motor DC,...

2.6 Ngôn ngữ lập trình Python và Open CV

Ngôn ngữ lập trình Python

[8] Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học và được dùng rộng rãi trong phát triển trí tuệ nhân tạo.

Ban đầu, Python được phát triển để chạy trên nền Unix. Theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.

Open CV



Hình 2.6: Python và OpenCV

[9] OpenCV là tên viết tắt của open source computer vision library – có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực.

OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các ứng dụng về máy học.

Ứng dụng của OpenCV trong đề tài được dùng để lập trình xử lý ảnh mã QR để đọc các nội dung từ mã QR code

2.7 Phần mềm thiết kế Altium Designer

Altium Designer trước kia có tên gọi quen thuộc là Protel DXP, là một trong những công cụ vẽ mạch điện tử mạnh nhất hiện nay. Được phát triển bởi hãng Altium Limited. Altium designer là một phần mềm chuyên ngành được sử dụng trong thiết kế mạch điện tử. Nó là một phần mềm mạnh với nhiều tính năng thú vị, tuy nhiên phần mềm này còn được ít người biết đến so với các phần mềm thiết kế mạch khác như Orcad hay Proteus [3].



Hình 2.7: Altium Designer 2022

Altium Designer có một số đặc trưng sau:

- Giao diện thiết kế, quản lý và chỉnh sửa thân thiện, dễ dàng biên dịch, quản lý file, quản lý phiên bản cho các tài liệu thiết kế.
- Hỗ trợ mạnh mẽ cho việc thiết kế tự động, đi dây tự động theo thuật toán tối ưu, phân tích lắp ráp linh kiện. Hỗ trợ việc tìm các giải pháp thiết kế hoặc chỉnh sửa mạch, linh kiện, netlist có sẵn từ trước theo các tham số mới.
- Mở, xem và in các file thiết kế mạch dễ dàng với đầy đủ các thông tin linh kiện, netlist, dữ liệu bản vẽ, kích thước, số lượng...
- Hệ thống các thư viện linh kiện phong phú, chi tiết và hoàn chỉnh bao gồm tất cả các linh kiện nhúng, số, tương tự...

- Đặt và sửa đổi tượng trên các lớp cơ khí, định nghĩa các luật thiết kế, tùy chỉnh các lớp mạch in, chuyển từ schematic sang PCB, đặt vị trí linh kiện trên PCB.
- Mô phỏng mạch PCB 3D, đem lại hình ảnh mạch điện trung thực trong không gian 3 chiều, hỗ trợ MCAD-ECAD
- Hỗ trợ thiết kế PCB sang FPGA và ngược lại.

Từ đó, chúng ta thấy Altium Designer có nhiều điểm mạnh so với các phần mềm khác như đặt luật thiết kế, quản lý đề tài mô phỏng dễ dàng...

Việc thiết kế mạch điện tử trên phần mềm Altium Designer có thể được tóm tắt gồm các bước như sau:

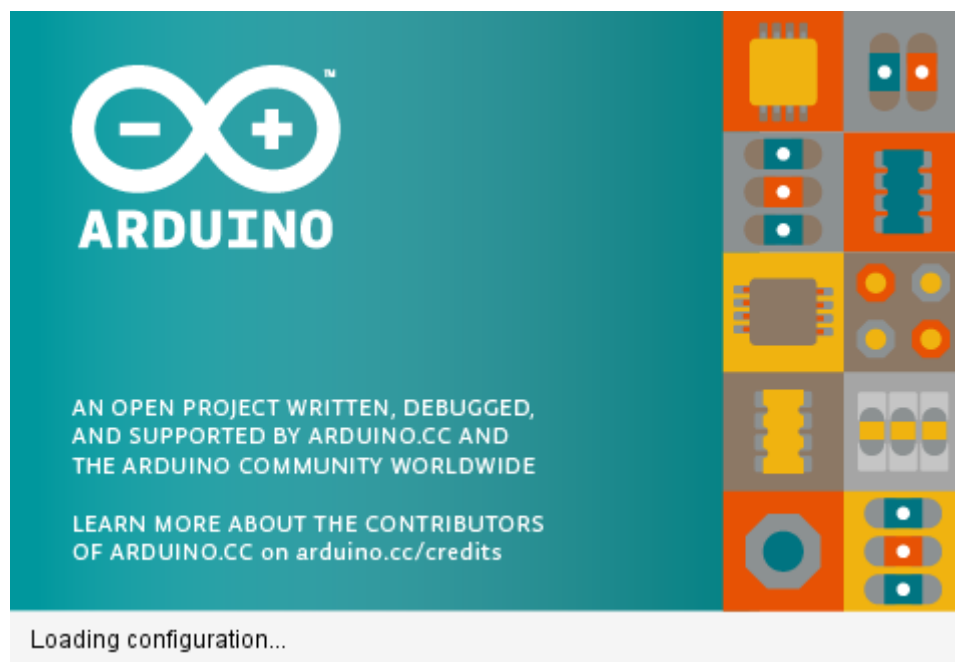
- Đặt ra các yêu cầu bài toán.
- Lựa chọn linh kiện.
- Thiết kế mạch nguyên lý.
- Lựa chọn các chân linh kiện để chuyển sang mạch in
- Update mạch nguyên lý sang mạch in.
- Lựa chọn kích thước mạch in
- Sắp xếp các vị trí các loại linh kiện như điện trở, tụ điện, IC...
- Đặt kích thước các loại dây nối.
- Đi dây trên mạch.
- Kiểm tra toàn mạch.
- Kiểm tra toàn bộ lỗi về mạch in và linh kiện
- Cho phép xem 2D và 3D của tất cả linh kiện và board mạch

2.8 Phần mềm Arduino IDE

[1] Sử dụng phần mềm Arduino IDE để lập trình cho Arduino Nano.

Arduino IDE là môi trường để lập trình và nạp code cho các dòng Arduino. Arduino IDE được xây dựng trên nền tảng Java nên hỗ trợ hầu hết các hệ điều hành hiện nay.

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.









Hình 2.8: Giao diện Arduino IDE

Khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và gửi đến bo mạch bằng cáp USB. Mỗi bo Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết.

Giao diện của Arduino IDE được chia thành 3 vùng chính:

Vùng 1: Các phím chức năng được nêu như bảng ...:

Bảng 2-1: Bảng chức năng của Altium Designer

Icon	Chức năng
	Biên dịch chương trình soạn thảo để kiểm tra các lỗi.
	Biên dịch và upload chương trình đang soạn thảo.
	Mở một trang soạn thảo mới.
	Mở các chương trình đã lưu.
	Lưu chương trình đang soạn thảo.
	Mở cửa sổ Serial Monitor để gửi và nhận dữ liệu giữa máy tính và board vi điều khiển.

Vùng 2: Cửa sổ để viết chương trình

- Chương trình Code sẽ được viết tại đây. Ở đây có hai hàm quan trọng là `setup()` và `loop()`.
- Hàm `setup()` được khởi chạy một lần duy nhất. Chức năng của hàm này dùng để khởi tạo các biến, khai báo chức năng các chân, khởi tạo các thông số ban đầu

- Hàm `loop()` là nơi chương trình được chạy lặp đi lặp lại đến khi ngắt vi điều khiển.
- Các dấu “//” dùng để tạo chú thích, giúp cho việc đọc code được dễ dàng hơn
- Trong lập trình có phân biệt ký tự hoa, thường, tuyệt đối phải đánh chính xác, đồng thời cuối mỗi câu lệnh cần phải có dấu chấm phẩy (;), trừ lệnh khai báo thư viện.

Vùng 3: Hiển thị các thông tin liên quan đến chương trình

Là cửa sổ để hiển thị về việc build chương trình, nạp chương trình thành công xuống vi điều khiển và các cảnh báo khác liên quan đến chương trình và điều khiển của chúng ta. Lưu ý, mọi thông báo và trạng thái của cả quá trình viết chương trình (write code), xây dựng chương trình (build code) và nạp chương trình (program code) đều được hiển thị tại đây. Cửa sổ này được gọi là cửa sổ debug.

2.9 Kết luận chương 2

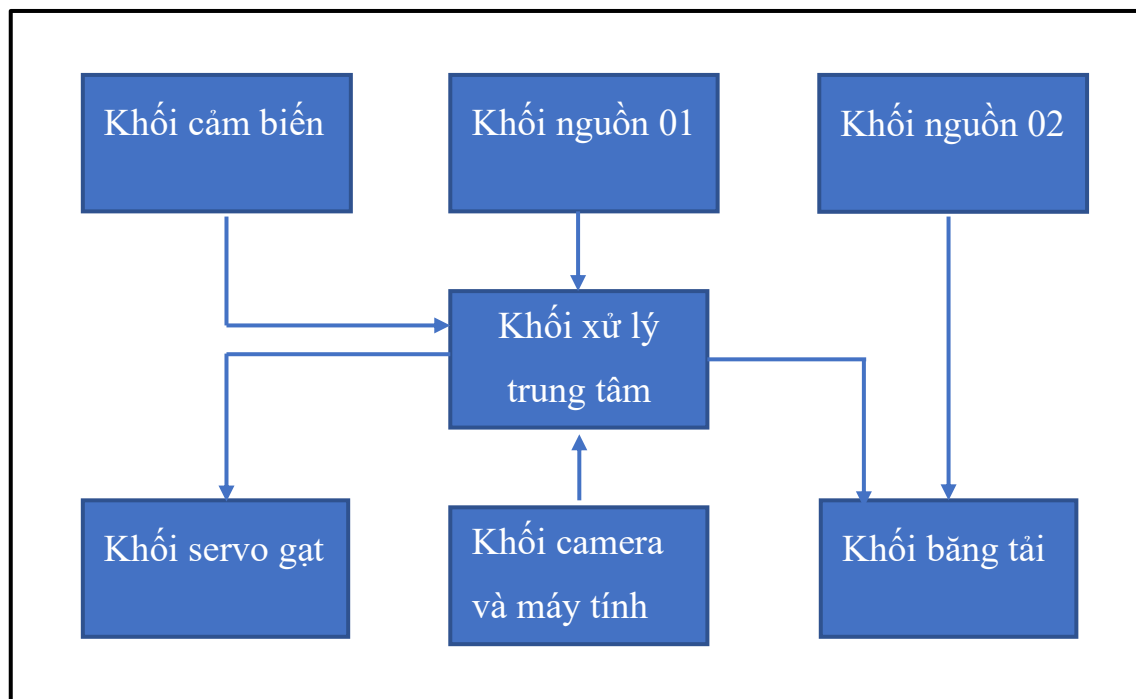
Chương 2 nói về việc sử dụng các thiết bị, các phần mềm và các công cụ đã dùng trong quá trình hoàn thành sản phẩm và các kiến trúc tổng quan phục vụ cho việc giải bài toán được đặt ra. Cùng với đó, chương 2 giới thiệu cũng như cách sử dụng các công cụ để lập trình và debug chương trình vi điều khiển Arduino Nano

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI SERVO ĐỂ ĐIỀU KHIỂN VÀ PHÂN LOẠI

3.1 Sơ đồ khối của hệ thống

Trong đề tài này, em sẽ thực hiện theo dạng mô hình mô phỏng dây truyền băng tải phân loại sản phẩm theo mã QR. Thông qua việc đọc qr code từ máy tính và camera để nhận biết được dữ liệu trong mã QR, từ đó vi điều khiển điều khiển cơ cấu phân loại sản phẩm vào từng ngăn phân loại theo mã QR của từng loại.

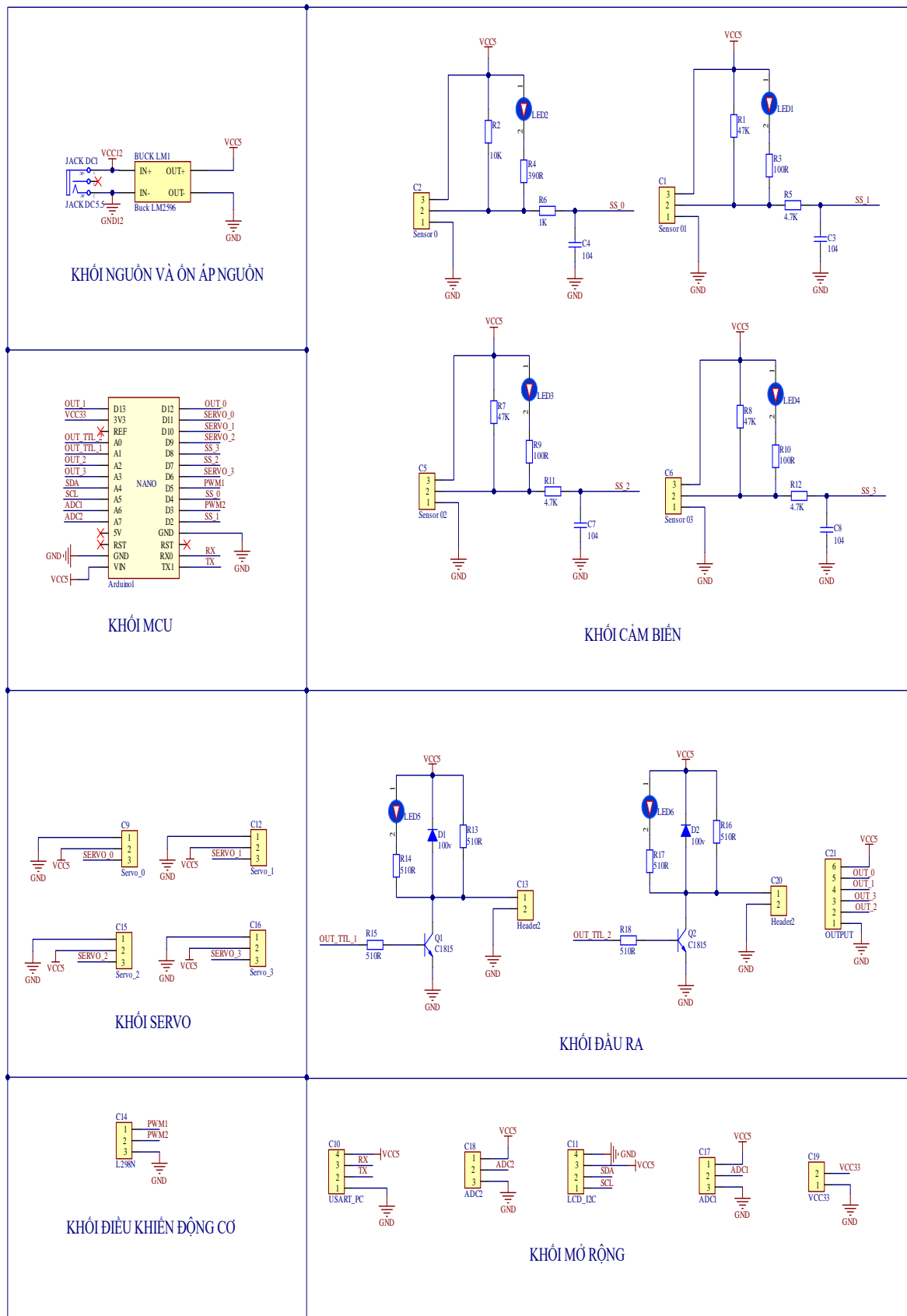
Sơ đồ khối của hệ thống.



Hình 3.1: Sơ đồ khối của hệ thống

- Khối nguồn: Cấp nguồn cho mạch điều khiển và băng tải
- Khối xử lý trung tâm: Xử lý các tín hiệu
- Khối băng tải: Di chuyển vật dựa vào sự quay của động cơ
- Khối camera và máy tính: Xử lý QR code và gửi xuống mạch điều khiển
- Khối cảm biến: phát hiện vật và trả về tín hiệu cho mạch điều khiển
- Khối servo gạt: Tiếp nhận tín hiệu và gạt các vật cần phân loại

3.2 Sơ đồ nguyên lý toàn mạch



Hình 3.2: Sơ đồ nguyên lý của hệ thống

3.2.1 Khối nguồn và ổn áp nguồn

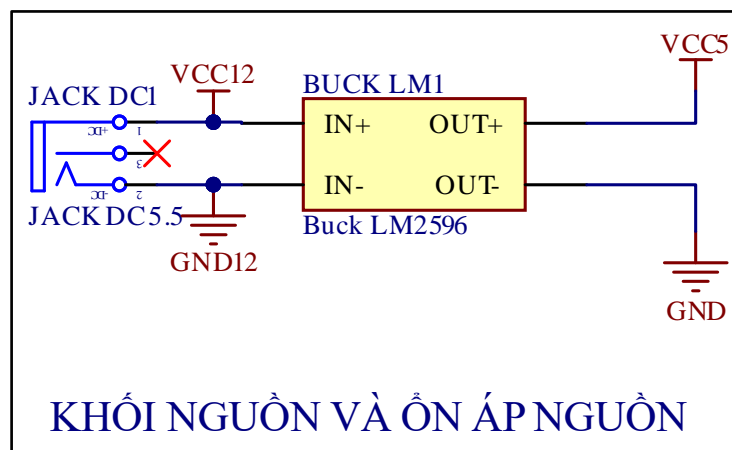
Khối nguồn

Mạch sử dụng 2 nguồn 12V và 24V riêng biệt được nối chung GND.

- Nguồn 12V dùng để cấp cho mạch điều khiển
- Nguồn 24V dùng để cấp cho động cơ của băng tải

Việc sử dụng 2 nguồn nhằm cách ly mạch điều khiển và mạch điều khiển động cơ để tránh gây nhiễu từ việc điều khiển động cơ vào mạch điều khiển.

Khối ổn áp nguồn



Hình 3.3: Khối nguồn và ổn áp nguồn

Mạch sử dụng nguồn điện 12V để cấp cho mạch điều khiển. Tuy nhiên khối xử lý cần nguồn 5V. Vì vậy cần một khối ổn áp chứa module nguồn LM2569 có tác dụng biến đổi nguồn cấp 12V về 5V để cấp cho mạch điều khiển.

3.2.2 Khối xử lý trung tâm

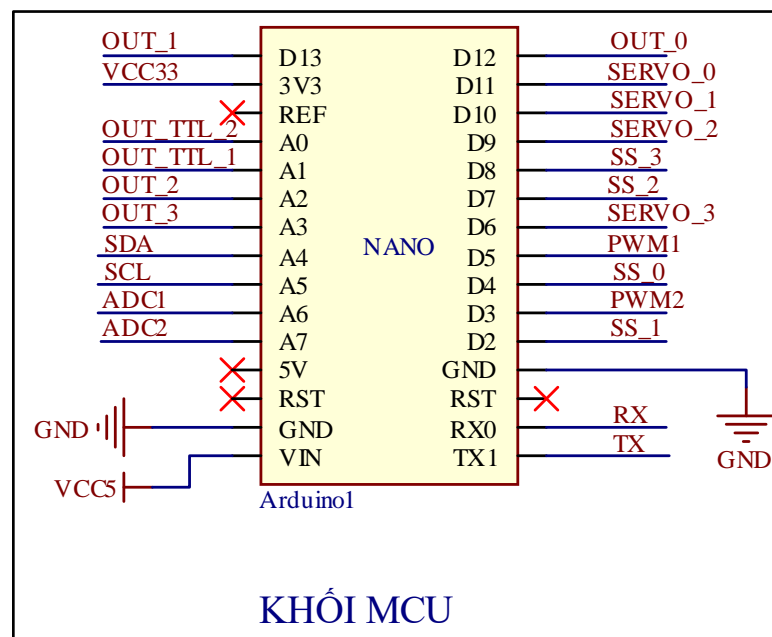
Sử dụng Aduino Nano để điều khiển toàn bộ hoạt động của hệ thống.

Arduino Nano thực hiện các nhiệm vụ nhận dữ liệu từ máy tính thông qua giao tiếp USART để điều khiển các thiết bị: băng tải, servo để gạt vào đúng máng cần phân loại.

Ngoài ra khối xử lý trung tâm còn nhận dữ liệu từ cảm biến để dừng vật cần đọc QR đúng vị trí góc camera để việc đọc QR trở nên chính xác hơn.

Các chân được sử dụng trong đề tài:

- VIN và GND : Cấp nguồn 5V từ khối nguồn và ổn áp nguồn
- Các chân D2, D4, D7 và D8: Sử dụng 1 trong số các chân để làm đầu vào của cảm biến quang
- Các chân D12, D13, A2, A3: Sử dụng 1 trong số các chân để làm đầu ra cho led chiếu sáng



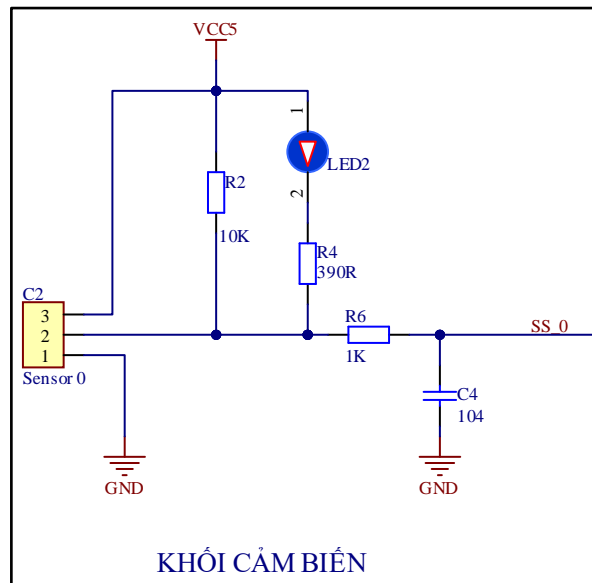
Hình 3.4: Khối MCU

- Chân D6, D9, D10 và D11: Sử dụng để điều khiển động cơ Servo. Trong đề tài sử dụng chân D10 và D11 để điều khiển 2 động cơ Servo
- Các chân D3 và D5: xuất ra đầu ra 2 kênh PWM để điều khiển động cơ quay với một tốc độ cố định và dừng động cơ
- Chân RX0 và TX1: đây là 2 chân USART của vi điều khiển, dùng để giao tiếp với máy tính

Một số các chân khác để sử dụng cho việc mở rộng điều khiển sau này như:

- Chân A4 và A5: Chân SDA và SCL để kết nối I2C
- Chân ADC1 và ADC2: mở rộng cho việc kết nối ADC

3.2.3 Khối cảm biến



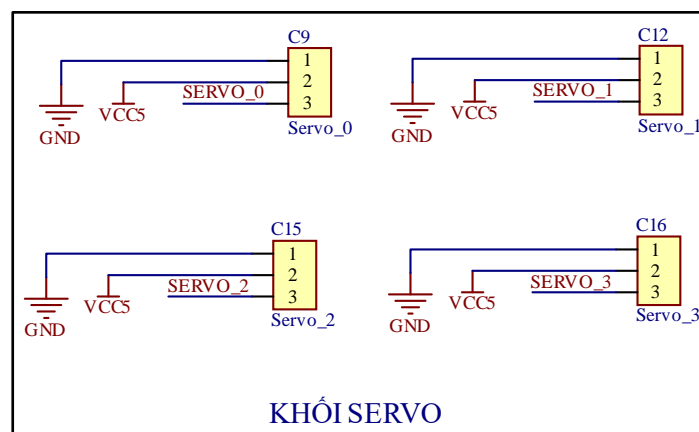
Hình 3.5. Khối cảm biến

Khối cảm biến được gắn một điện trở kéo lên có giá trị 10K, ngoài ra còn có các led 3mm báo tín hiệu.

Ngoài ra mạch còn được trang bị thêm mạch lọc thông cao RC nhằm tránh việc nhiễu từ tín hiệu cảm biến .

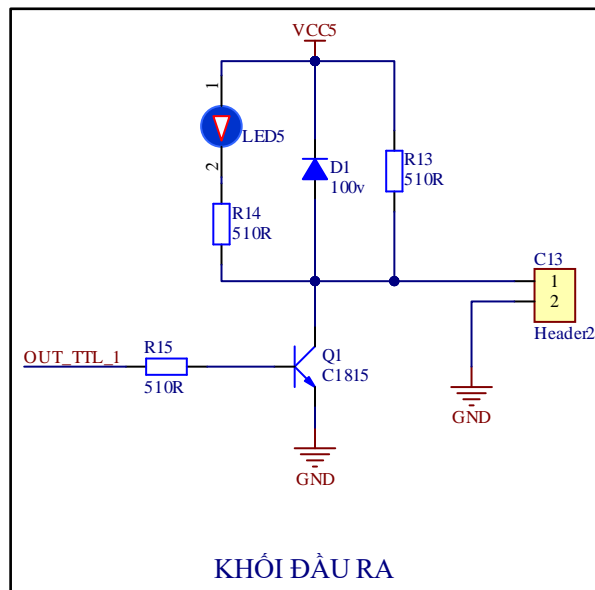
Cảm biến được sử dụng trong đề tài là cảm biến hồng ngoại dùng để phát hiện vật để dừng băng tải đúng vị trí camera có thể quét được nhằm đọc mã QR chính xác.

3.2.4 Khối servo



Hình 3.6: Khối Servo

3.2.5 Khối đầu ra



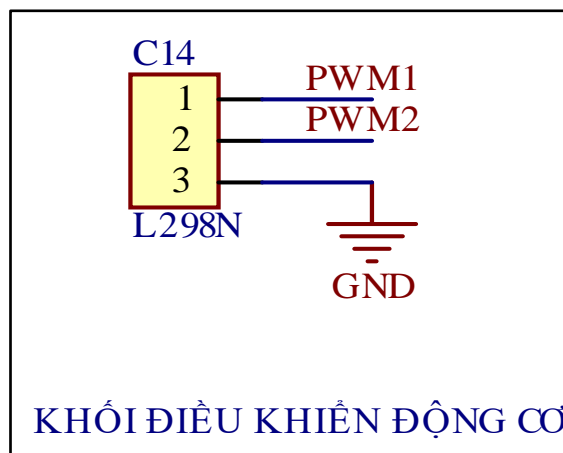
Hình 3.7: Khối đầu ra

Khối đầu ra được lấy tín hiệu từ một chân GPIO của vi xử lý để kích mở Transistor. Transistor được sử dụng trong mạch là C1815.

Diode D1 là một diode chuyển mạch 1N148 có tác dụng chống dòng ngược khi tải mắc với relay.

Khối đầu ra trong đề tài được mắc với một LED chiếu sáng có tác dụng chiếu sáng mã qr để camera có thể đọc được tốt hơn trong điều kiện môi trường là trời tối.

3.2.6 Khối điều khiển động cơ bằng tải



Hình 3.8: Khối điều khiển động cơ bằng tải

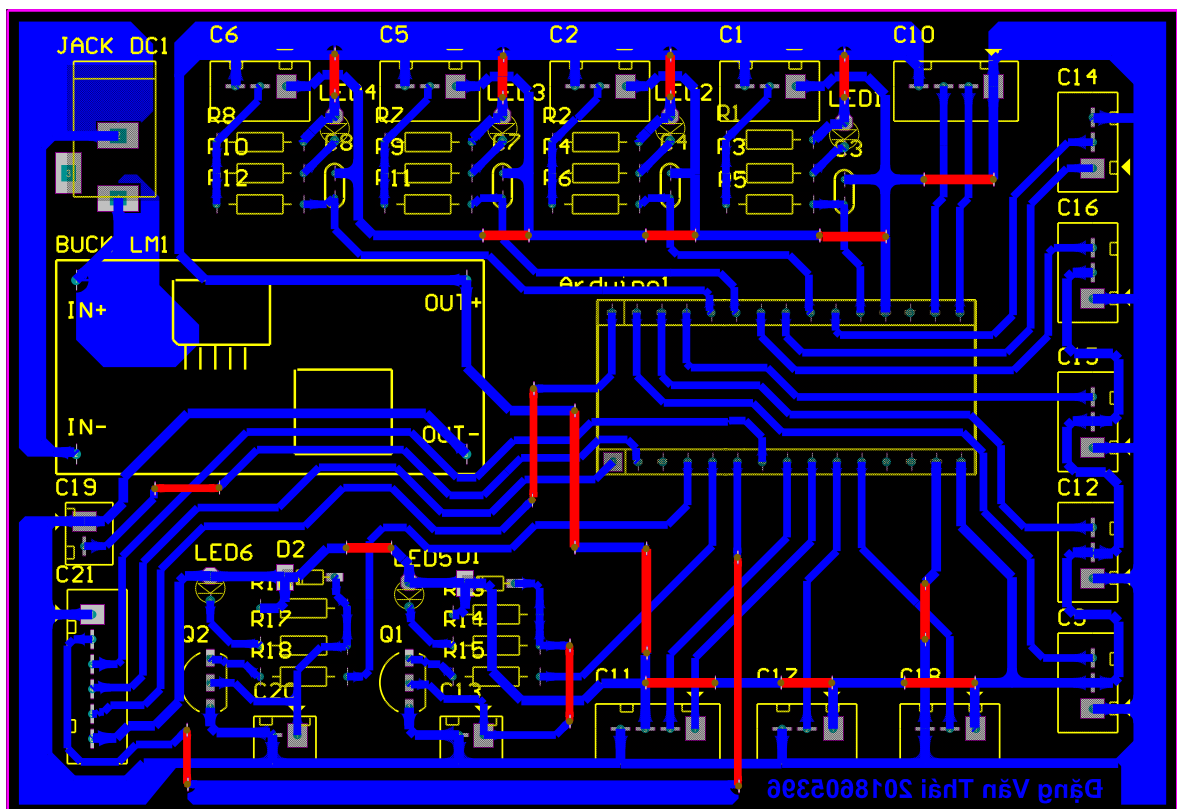
Khởi điều khiển động cơ bằng tải sử dụng module L298N với 2 kênh PWM. Module L298N cho phép cấp 4 kênh PWM. Do vậy Module điều khiển được tối đa 2 động cơ với dòng tối đa 2A. Trong đề tài sử dụng 1 động cơ của băng tải để dừng và quay với một tốc độ cố định.

3.3 Thiết kế phần cứng

Đặt luật cho mạch in:

- Khoảng cách giữa các thành phần (clearance): 1mm
- Chiều rộng dây tín hiệu: 0.8mm
- Chiều rộng dây GND: 1mm
- Chiều rộng dây VCC: 1mm

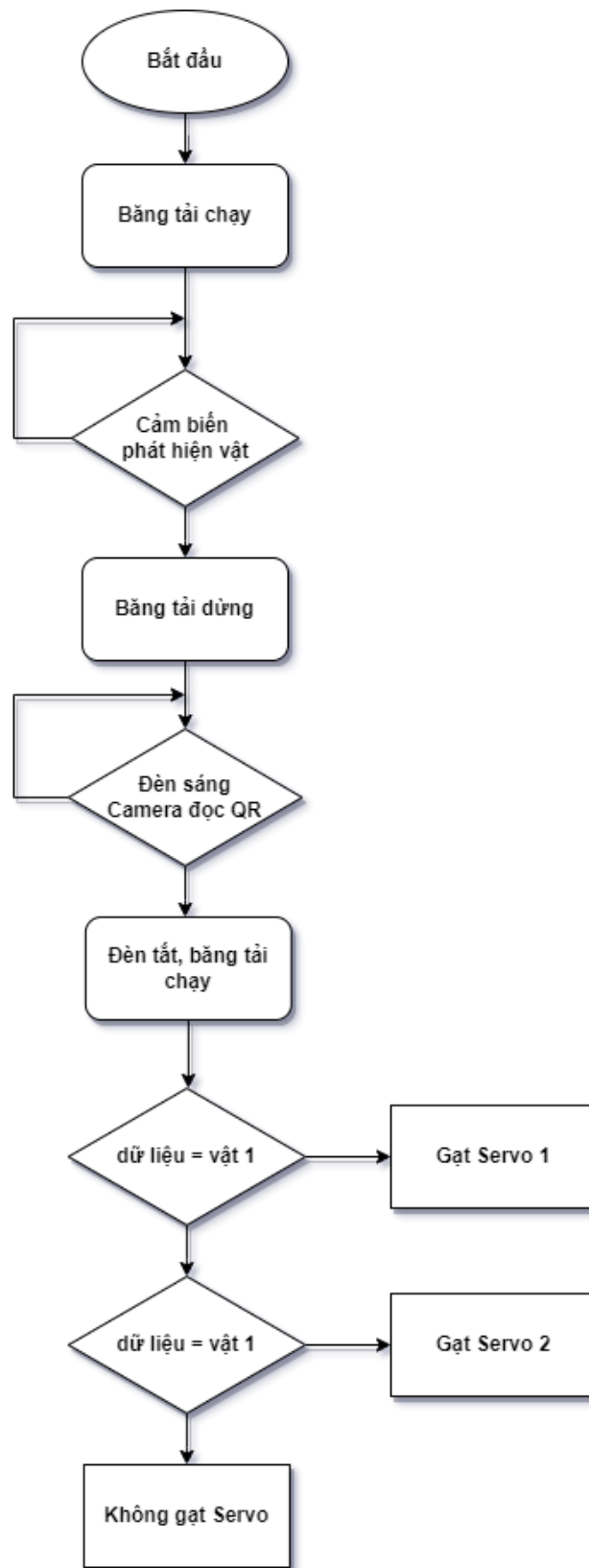
Sơ đồ mạch in



Hình 3.9: Sơ đồ mạch in của hệ thống

3.4 Thiết kế phần mềm

3.4.1 Lưu đồ thuật toán



Hình 3.10: Lưu đồ thuật toán

3.4.2 Phần mềm đọc mã QR

[2] Sử dụng OpenCV để đọc mã QR và sử dụng Tkinter của Python để tạo ra một giao diện đơn giản hiển thị góc quét của Camera



Hình 3.11: Giao diện đọc mã QR

Giao diện có sự kết hợp của OpenCV và Tkinter để tạo nên [11] :

Các thành phần chính của giao diện:

Khởi tạo camera để đọc QR code: Sử dụng OpenCV để khởi tạo, hàm khởi tạo camera:

```
cap = cv2.VideoCapture(2)
```

Hàm đọc QR code của OpenCV: Dùng để đọc QR code từ video lấy được từ camera:

```
dataImage, bboxImage, _ = detector.detectAndDecode(img)
```

Một số các nhãn như tên, mã sinh viên được tạo bằng Tkinter.

3.5 Mô hình thực tế



Hình 3.13: Tổng quát mô hình thực tế



Hình 3.12: Mô hình thực tế và khối điều khiển

KẾT LUẬN

Qua quá trình thực hiện làm đồ án, em đã trình bày các cơ sở lý thuyết liên quan và chạy thành công hệ thống : **“Sử dụng cảm biến quét mã qr, băng tải Servo để điều khiển và phân loại”**. Tuy thời gian làm đồ án thực sự không quá dài nhưng được sự giúp đỡ tận tình của TS. Phạm Xuân Thành cùng với sự nỗ lực và cố gắng của bản thân, sự chỉ bảo của các Thầy Cô trong khoa Điện tử em đã hoàn thành đề tài theo yêu cầu và đúng thời gian quy định với những nội dung sau:

- Nghiên cứu và tìm hiểu về các hệ thống phân loại sản phẩm trên thực tế, ưu điểm và nhược điểm của từng hệ thống
- Nghiên cứu và tìm hiểu về vi điều khiển Arduino Nano, các ngoại vi và lập trình Arduino Nano
- Tìm hiểu về ngôn ngữ lập trình Python, các thư viện xử lý ảnh và đọc QR code của OpenCV
- Nắm được các hệ thống điều khiển động cơ DC và động cơ Servo
- Xây dựng được giao diện đọc QR code
- Thiết kế và vận hành thành công mạch điều khiển động cơ DC và động cơ Servo cho băng tải

Hướng phát triển đề tài:

- Tạo giao diện phong phú và đa dạng hơn
- Nâng cấp mô hình với việc tự động hoàn toàn, tự động cấp phối
- Phát triển thêm nhiều tính năng về xử lý ảnh để nhận diện các vật khác nhau không chỉ là quét mã QR
- Điều khiển và giám sát thêm các thiết bị khác, các loại thiết bị có công suất lớn
- Tạo ra một dây chuyền tự động để ứng dụng đề tài vào trong một dây chuyền cụ thể, có tính ứng dụng cao trong nhà máy và các cơ sở sản xuất

Thông qua quá trình làm đồ án, em đã được vận dụng những kiến thức chuyên ngành trong 4 năm học. Qua đó đã giúp cho em rèn luyện được kỹ năng, cách tiếp cận với các vấn đề, các bài toán thực tế phức tạp tại các doanh nghiệp, nhà máy khi ra trường làm việc.

Việc xây dựng mô hình đã đáp ứng được yêu cầu đặt ra, tuy nhiên do trình độ và kinh nghiệm thực tiễn còn hạn chế nên không thể tránh khỏi sai sót và thiếu hoàn chỉnh. Rất mong được đón nhận sự đóng góp ý kiến từ thầy cô và các bạn.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- [1] Phạm Quang Huy, Lê Cảnh Trung(2016), Lập trình điều khiển với Arduino, NXB Khoa học & Kỹ thuật.
- [2] Nguyễn Văn Hậu, Nguyễn Duy Tân(2019), *Python cơ bản*, NXB Đại Học Quốc Gia Hà Nội
- [3] <https://www.altium.com/altium-designer>
- [4] <http://arduino.vn/bai-viet/1219-tap-lenh-voi-arduino>
- [5] <https://en.wikipedia.org/wiki/Arduino>
- [6] <https://hshop.vn/products/dong-co-rc-servo-futaba-s3003-2>
- [7] <https://intech-group.vn/thong-tin-tap-doan.htm>
- [8] <https://www.python.org/>
- [9] <https://en.wikipedia.org/wiki/OpenCV>
- [10] <https://pyimagesearch.com/2018/05/21/an-opencv-barcode-and-qr-code-scanner-with-zbar/>
- [11] <https://lindevs.com/detect-and-decode-qr-code-in-image-using-opencv/>

PHỤ LỤC**Code Arduino Nano**

```
#include <Servo.h>

Servo servo1;

Servo servo2;

String readString;

int sv1AngleFirst = 0;

int sv1AngleLast = 150;

int sv2AngleFirst = 0;

int sv2AngleLast = 150;

int servo1Pin = 10;

int servo2Pin = 6;

void servoInit();

void setup() {

    servoInit();

    pinMode(3, OUTPUT);

    pinMode(5, OUTPUT);

    pinMode(8, OUTPUT);

    Serial.begin(9600);

}

bool flagsr1 = false;

void loop()
```

```

{
    while(!Serial.available()) {}

    while (Serial.available())
    {
        if (Serial.available() >0)
        {
            String c = Serial.readStringUntil('@');
            Serial.println(c);
            if(c == "Sản phẩm 01")
            {
                digitalWrite(13, LOW);
                servo2.write(50);
            }
            else if(c == "Sản phẩm 02")
            {
                digitalWrite(13, HIGH);
                digitalWrite(13, LOW);
                servo2.write(0);
                delay(1000);
            }
        }
    }
}

void servoInit()

```

```

{
servo1.attach(servo1Pin);

servo1.write(sv1AngleFirst);

servo2.attach(servo2Pin);

servo2.write(sv2AngleFirst);

}

```

Code Python:

```

from time import sleep

import cv2

import numpy as np

from tkinter import *

from PIL import Image, ImageTk

import serial

from threading import Thread

import threading

import time

try:

    ser = serial.Serial('COM10')

except:

    print("An exception occurred")

i = 0

```

```

global bbox

global data

dataImage = ""

bboxImage = ""

_ = ""

flagVar = 0

root = Tk()

detector = cv2.QRCodeDetector()

root.geometry("700x700")

root.configure(bg="black")

Label(root, text="Đặng Văn Thái", font=("time new roman", 30, "bold"),
bg="black", fg = "blue").pack()

Label(root, text="Đọc QR code", font=("time new roman", 30, "bold"),
bg="black", fg = "red").pack()

f1 = LabelFrame(root, bg = "red")

f1.pack()

L1 = Label(f1, bg = "red")

L1.pack()

cap = cv2.VideoCapture(2)

def thread_function():

    while True:

        print ("Message from arduino: ")

        try:

```

```

    msg = ser.readline()

except:

    print("An exception occurred")

    print(msg.decode())

def thread_function_2():

    while True:

        if bboxImage is not None and dataImage != "":

            try:

                ser.write(dataImage.encode())

            except:

                print("An exception occurred")

                time.sleep(5)

if __name__ == "__main__":

    t1 = threading.Thread(target=thread_function)

    t1.start()

    t2 = threading.Thread(target=thread_function_2)

    t2.start()

    while True:

        img = cap.read()[1]

        dataImage, bboxImage, _ = detector.detectAndDecode(img)

        if flagVar == 0:

            flagVar = 1

```

```
img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
```

```
img = ImageTk.PhotoImage(Image.fromarray(img))
```

```
L1['image'] = img
```

```
root.update()
```