# MỤC LỤC

[MỤC LỤC 1](#_Toc533771189)

[DANH MỤC HÌNH ẢNH 3](#_Toc533771190)

[CHƯƠNG 1 . GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 5](#_Toc533771191)

[1.1 Tình trạng thiếu bãi đỗ xe ở Việt Nam 5](#_Toc533771192)

[1.2. Đối tượng và phạm vi nghiên cứu 5](#_Toc533771193)

[1.3. Nhiệm vụ của luận văn 6](#_Toc533771194)

[1.4 Phân tích và lựa chọn mô hình bãi đỗ xe tự động 6](#_Toc533771195)

[1.4.1 Các giải pháp bãi giữ xe ô tô tự động 6](#_Toc533771196)

[CHƯƠNG 2 :THIẾT KẾ HỆ THỐNG CƠ KHÍ 10](#_Toc533771197)

[2.1. Nguyên lý hoạt động 10](#_Toc533771198)

[2.2 Tính toán thiết kế hệ dẫn động 11](#_Toc533771199)

[2.2.1. Các thông số thiết kế : 11](#_Toc533771200)

[2.2.3. Tính toán bộ truyền đai : 18](#_Toc533771201)

[2.3 Các thành phần cơ bản và thông số kỹ thuật của hệ thống 22](#_Toc533771202)

[2.3.1. Bộ khung thép 22](#_Toc533771203)

[2.3.2. Pallet để xe 23](#_Toc533771204)

[2.3.3. Bánh răng quay 23](#_Toc533771205)

[2.3.4. Bàn xe di động 24](#_Toc533771206)

[2.3.5. Cơ cấu di chuyển ngang 24](#_Toc533771207)

[2.3.6. Cơ cấu di chuyển đứng 24](#_Toc533771208)

[2.3.7. Dây đai. 25](#_Toc533771209)

[2.3.8. Động cơ 26](#_Toc533771210)

[CHƯƠNG 3 : THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN 27](#_Toc533771211)

[3.1 Các thành phần thiết bị sử dụng 27](#_Toc533771212)

[3.1.1 Arduino Mega 2560 27](#_Toc533771213)

[3.1.2 Modul A4988 28](#_Toc533771214)

[3.1.3 Cảm biến khoảng cách hồng ngoại MH-B 30](#_Toc533771215)

[3.1.4. Công tắc hành trình 31](#_Toc533771216)

[3.2 Thuật toán điều khiển 33](#_Toc533771217)

[3.2.1 Tổng quan về hệ thống điều khiển 33](#_Toc533771218)

[3.2.2. Thuật toán điều khiển Arduino 34](#_Toc533771219)

[3.2.3 Thuật toán điều khiển từ máy tính 35](#_Toc533771220)

[3.3 Công nghệ xử lý ảnh số 37](#_Toc533771221)

[3.4 Công nghệ thẻ từ RFID 40](#_Toc533771222)

[3.5 Xây dựng chương trình máy tính 42](#_Toc533771223)

[3.5.1 Thiết kế giao diện người dùng 42](#_Toc533771224)

[3.5.2 Cơ sở dữ liệu bãi đỗ xe 44](#_Toc533771225)

[3.5.3 Hệ thống giám sát 46](#_Toc533771226)

[CHƯƠNG 4:MÔ HÌNH THỰC TẾ 47](#_Toc533771227)

[4.1. Thi công mô hình 47](#_Toc533771228)

[CHƯƠNG 5 : KẾT LUẬN 53](#_Toc533771229)

[5.1. Phương hướng và định hướng phát triển của đê tài 53](#_Toc533771230)

[5.2 . Ứng dụng vào thực nghiệm 53](#_Toc533771231)

[TÀI LIỆU THAM KHẢO 54](#_Toc533771232)

[CODE CHƯƠNG TRÌNH 55](#_Toc533771233)

# DANH MỤC HÌNH ẢNH

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

*Hình1.1: Giải pháp "bãi đỗ xe xếp chồng"*

*Hình1. 2: Giải pháp "bãi giữ xe otô tự động tòa nhà hình trụ"*

*Hình 1.3. Giải pháp “bãi giữ xe ô tô tự động dạng tòa nhà vuông”*

CHƯƠNG 2: THIẾT KẾ HỆ THỐNG CƠ KHÍ

*Hình 2.1: Biểu đồ hoạt động*

*Hình 2.2: Bảng thông số chọn động cơ*

*Hình 2.3: Động coq bước Nema 17*

*Hình 2.4: Sơ đồ bộ truyền đai*

*Hình 2.5: Tra thông số khối lượng đai*

*Hình 2.6: Bộ truyền đai với thông số sơ bộ*

*Hình 2.7: Dây đai*

*Hình 2.8: Kết cấu và thông số kích thước*

*Hình 2.9: Thông số kích thước Pallet*

*Hình 2.10: Thông số bánh răng*

*Hình 2.11: Thông số hình học của bàn xe*

*Hình 2.12: Sơ đồ động cơ cấu di chuyển ngang*

*Hình 2.13: Sơ đồ động cơ cấu di chuyển đứng*

*Hình 2.14: Động cơ*

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN

*Hình 3.1: Ardunio Mega 2560*

*Hình 3.2: Modul A4988*

*Hình 3.3: Sơ đồ nối mạch*

*Hình 3.4: Cảm biến MH-B*

*Hình 3.5: Sơ đồ đấu dây công tắc hành trình*

*Hình 3.6: Tổng quan hệ thống*

*Hình 3.7: Sơ đồ vị trí cảm biến và công tắc hành trình*

*Hình 3.8: Thuật toán xử lý ảnh*

*Hình 3.9: Ảnh biển số sau khi nhị phân hóa*

*Hình 3.9: Ảnh biển số sau khi tìm contour*

*Hình 3.10: Ảnh sau khi đã cắt kí tự*

*Hình 3.11: Sơ đồ nguyên lý hoạt động của RFID*

*Hình 3.12: Một loại thẻ chip RFID phổ biến*

*Hình 3.13: Giao diện điều khiển trên máy tính*

*Hình 3.14: Cơ sở dữ liệu*

CHƯƠNG 4: MÔ HÌNH THỰC TẾ

*Hình 4.1: Mô hình thực tế*

*Hình 4.2: Hệ thống nút nhấn*

*Hình 4.3: Tủ điện của hệ thống*

*Hình 4.4: Đầu đọc thẻ từ RFID*

*Hình 4.4: Động cơ bước Nema 17*

*Hình 4.5: Camera của hệ thống*

# CHƯƠNG 1 . GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## 1.1 Tình trạng thiếu bãi đỗ xe ở Việt Nam

Cùng với sự phát triển của một nền kinh tế năng động đòi hỏi một nền kỹ thuật công nghệ cũng phải phát triển, đáp ứng kịp thời và đầy đủ những nhu cầu cần thiết của sự phát triển đó. Ngày nay ở các trung tâm thành phố lớn với sự phát triển mật độ dân cư và sẽ có người càng đông đúc.Đặc biệt là sự gia tăng về số lượng xe ô tô ngày càng nhiều điều này phần nào cũng phản ánh sự phát triển của một quốc gia. Song song với sự phát triển đó,người ta đặt vấn đề là xây dựng những bãi giữ xe để phục vụ cho người dân trong công việc cũng như trong việc đi lại của họ vì thế ngày nay trên các nước tiên tiến trên thế giới như Nhật Bản,Hàn Quốc ở những thành phố chật hẹp,người ta xây dựng hệ thống bãi giữ xe ôtô tự động được trang bị thiết bị nâng để di chuyển ôtô từ mặt đất lên điểm đỗ cao đây là những giải pháp giúp tăng hơn 100 lần số lượng xe trên một diện tích truyền thống,cho phép giải quyết tình trạng thiếu mặt bằng xây dựng.Mặc dù trên thế giới, nhiều bãi đậu xe có kỹ thuật cao và hiện đại đã được đưa vào ứng dụng trong thực tế. Tuy nhiên, với mong muốn đưa ý tưởng từ những kiến thức đã học vào trong thực tế giải quyết khó khăn này, chúng em quyết định đưa ra và thi công đề tài: “ Thiết kế và thi công mô hình bãi giữ xe ô tô tự động”

## 1.2. Đối tượng và phạm vi nghiên cứu

Việc nghiên cứu thành công đề tài mang lại nhiều ý nghĩa sâu sắc. Bất kỳ một ứng dụng hay thử nghiệm nào cũng dựa trên những cơ sở lý thuyết khoa học hay một giả định khoa học. Kết quả của đề tài một lần nữa đã đưa những ứng dụng từ những thiết bị tự động vào trong công nghiệp xuất phát từ cơ sở lý thuyết điều khiển tự động. Đề tài cũng cho thấy những bộ khả lập trình có thể đưa vào các công trình tự động lớn một cách chính xác và hoàn hảo chẳng hạn từ bộ khả trình Arduino có thể lập trình điều khiển khung nâng hoạt động một cách tinh tế. Ngoài ra, đề tài còn cho người sử dụng thấy được tính tự động hóa cao trong kỹ thuật thông qua công nghệ quét thẻ từ, kết hợp nhận dạng biển số xe, hay đến việc giám sát hoạt động của hệ thống thông qua máy vi tính.

Về mặt thực tiễn, đề tài là một mô hình thực tế nhằm đưa ra hướng giải quyết tình trạng thiếu bãi đỗ xe tại các trung tâm kinh tế ở các thành phố lớn như Hà Nội, Hồ Chí Minh. Đồng thời, quá trình thực hiện đề tài là một quá trình trải nghiệm thực tế hết sức bổ ích. Thông qua quá trình làm đề tài, khả năng lập trình với Arduino được vững vàng hơn, hiểu sâu hơn về các truyền thông giữa Arduino với máy tính,… và bên cạnh đó môi trường công nghiệp cũng gần gũi hơn qua tiêu chí lựa chọn từng phương pháp thực hiện và thiết bị cụ thể.

## 1.3. Nhiệm vụ của luận văn

Phân tích các giải pháp bãi giữ xe ô tô tự động trên thế giới để lựa chọn phù hợp với đặc tính kinh tế, quỹ đất xây dựng, kỹ thuật công nghệ... của Việt Nam. Từ đó, lựa chọn phương án và thực hiện thiết kế và thi công mô hình bãi giữ xe ô tô tự động, và chính từ sự thành công và tính khả thi trên mô hình để các nhà đầu tư phát triển các dự án xây dựng công trình thực tế.

## 1.4 Phân tích và lựa chọn mô hình bãi đỗ xe tự động

### 1.4.1 Các giải pháp bãi giữ xe ô tô tự động

#### 1.4.1.1 Giải pháp “bãi giữ xe ô tô xếp chồng”

Ưu điểm: hệ thống gọn nhẹ, dễ vận chuyển, lắp đặt nhanh.

Nhược điểm: Sử dụng hệ thống thủy lực hoặc động cơ để nâng tối đa bốn ôtô xếp cạnh nhau lên một tầm cao, để dành chỗ cho những xe khác ở bên dưới. Tuy nhiên, giải pháp này có hiệu quả kinh tế không cao, chỉ phù hợp với qui mô nhỏ: một hoặc vài hộ gia đình. Ngoài ra, chưa có tối ưu hóa tính tự động.



*Hình1.3: Giải pháp "bãi đỗ xe xếp chồng"*

#### 1.4.1.2. Giải pháp “bãi giữ xe ô tô tự động dạng tòa nhà hình trụ”

Ưu điểm: diện tích lớn, số lượng xe nhiều, các pallet hoạt động lên xuống và theo các góc nhất định giảm thiểu thời gian lấy và gửi xe, xe hoàn toàn tắt máy nên không gây ô nhiễm khói bụi.

Nhược điểm: Để có số lượng chỗ để xe nhiều cần phải xây dựng tòa nhà thật sự rộng và cao. Điều này tốn nhiều quỹ đất xây dựng, cũng như phần nền móng vững chắc. Tuy nhiên, do các thành phố lớn sự đông đúc và chật chội vấn đề diện tích đất để xây dựng là vô cùng khó khăn. Ngoài ra, yếu tố kinh tế, và đòi hỏi về kỹ thuật công nghệ cao để xây dựng cũng gặp nhiều khó khăn.



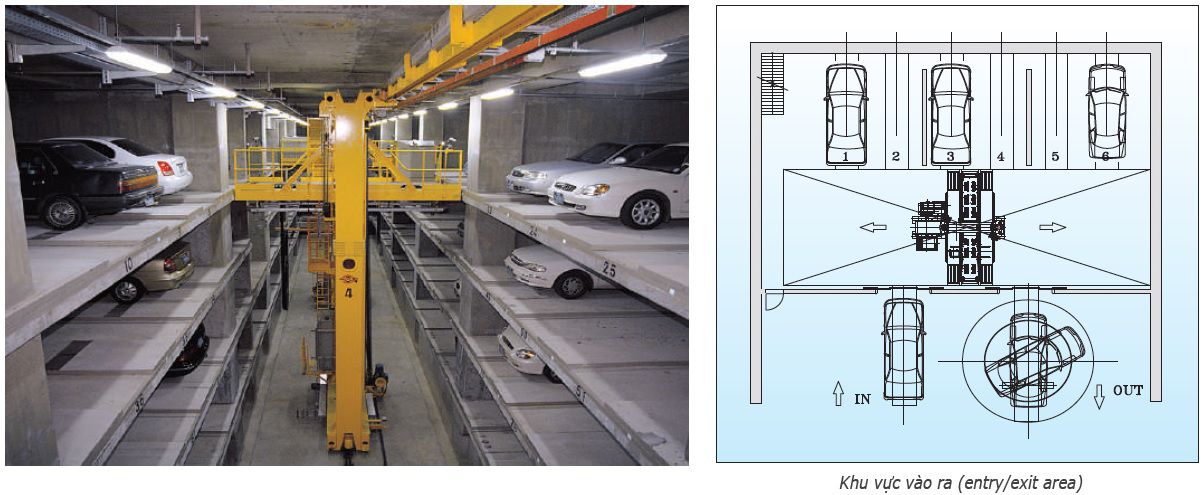
*Hình1. 4: Giải pháp "bãi giữ xe o tô tự động tòa nhà hình trụ"*

#### 1.4.1.3. Giải pháp “bãi giữ xe ô tô tự động dạng tòa nhà vuông

Ưu điểm: Chiếm diện tích đất xây dựng, chi phí xây dựng và kỹ thuật công nghệ không đòi hỏi quá lớn. Số lượng xe nhiều, tối ưu hóa số lượng chỗ để

xe do xây dựng các tầng đối xứng. Các xe phải hoàn toàn tắt máy nên không ô nhiễm khói bụi.

Nhược điểm: So với dạng tòa nhà hình trụ thì số lượng xe ít hơn, do chỉ có một pallet đảm nhiệm lấy và cất nên không tối ưu về thời gian.



*Hình 1.3. Giải pháp “bãi giữ xe ô tô tự động dạng tòa nhà vuông”*

#### 1.4.2. Lựa chọn phương án

Qua phân tích ưu và nhược điểm các Giải pháp bãi giữ xe ô tô tự động, nhóm thực hiện luận văn đã chọn phương án theo các tiêu chí sau: Quỹ đất xây dựng nơi các thành phố lớn như Hà Nội, thành phố Hồ Chí Minh không nhiều. Phù hợp với kinh phí xây dựng các tòa nhà, cũng như cơ sở vật chất kỹ thuật công nghệ. Do đó, nhóm luận văn đã lựa chọn và thực hiện đề tài “Thiết kế và thi công mô hình bãi giữ xe ô tô tự động”, với cấu trúc dạng tòa nhà vuông.

Tuy nhiên, trong phạm vi thực hiện đề tài có những hạn chế về mặt cơ khí, nên chỉ thiết kế một bên khung để xe, và dễ dàng phát triển hai khung để xe đối xứng nhau.

# CHƯƠNG 2 :THIẾT KẾ HỆ THỐNG CƠ KHÍ

## 2.1. Nguyên lý hoạt động

Quy trình diễn ra như sau : Hành khách muốn gửi xe thì lái xe vào vị trí in Khi đó hệ thống sẽ kiểm tra xem có xe chưa thông qua cảm biến , khi có xe thì quẹt thẻ mới được chấp nhận qua hệ thống nhận diện biển số đồng thời nhận được vị trí đỗ, tiếp theo đó tay khung nâng thực hiện đưa xe vào khung nâng , khung nâng di chuyển đến vị trí pallet bằng 1 động cơ để đưa chiếc xe vào đúng vị trí. Khi lấy xe ra, bàn nâng cũng di chuyển đến vị trí đó để đưa chiếc xe được yêu cầu về mặt đất và người dùng có thể lấy xe.

**Nguyên lý cất giữ xe :**

Bước 1 : Chủ xe quẹt thẻ và nhận số ô mình để xe

Bước 2 : xe sẽ được cho vào khung nâng qua cơ cấu tay nâng

Bước 3 : Hệ thống cơ cấu nâng hoạt động đưa xe tới vị trí cất trữ (theo ô nhớ)

trên pallet

Bước 4 : Cơ cấu nâng và bàn xe di động hoạt động đồng thời để đưa xe đến

đúng vị trí

**Nguyên lý lấy xe ra :**

Bước 1 : Chủ xe bấm nút ô số mình vừa gửi xe

Bước 2 : Hệ thống nâng và bàn di động cùng hoạt động để đưa xe ra chỗ vị trí

lấy xe

Bước 3 : Cửa nhà xe mở ra.

Bước 4 : Xe được lấy ra và kết thúc chu trình

## 2.2 Tính toán thiết kế hệ dẫn động

### 2.2.1. Các thông số thiết kế :

Kích thước xe tối đa : D 150 mm, R 100 mm, C 50 mm

Kích thước ô chứa : D 200 mm, R 150 mm, C 150 mm

Trọng lượng xe tối đa : 2kg = 20 N

Trọng lượng cơ cấu nâng : 2kg = 20 N

Tổng trọng lượng xe và 1 ô chứa : 40 N

Vận tốc bàn xe chạy V = 0,2 m/s

Số lượng xe bãi đỗ : 3 tầng = 9 xe

*- Tính chọn động cơ kéo cơ cấu khung nâng dọc :*

Chọn sơ bộ thông số : m = 3kg

µ = 0,05

l = 250 mm

t0 = 2,67s

Các bước tính toán chọn động cơ

Yêu cầu tính ra moomen xoắn của động cơ:

Bước 2**.** Tính số xung

A=\*

l : Quãng đường chuyển động [m]

lrev : Quãng đường dịch chuyển trên 1 vòng động cơ

 : Bước góc

Bước 3**.** Tính tần số hoạt động f2

Thời gian tăng tốc : t1 = t0.0,125

Tần số hoạt động :

Bước 4**.** Tính tốc độ hoạt động của động cơ

Nm(r/min)

Bước 5**.** Tính Mômen tải

(Do kết cấu của trục X, Y gần giống nhau chỉ khác nhau 1 chút về hành trình nên ta đồng cơ trục X làm động cơ trục Y)

Trục X

Chọn sơ bộ thông số:

Khối lượng tải sơ bộ : m = 3 kg

Hệ số ma sát cụm trục : µ = 0,05

Chọn sơ bộ bước góc động cơ 

Đường kính ngoài pulley : D = 10 mm

Chiều cao pulley : L = 15 mm

Quãng đường dịch chuyển : Sv = 600 mm

Thời gian làm việc hết một chu trình : t0 = 0,067s

Hệ số công suất 

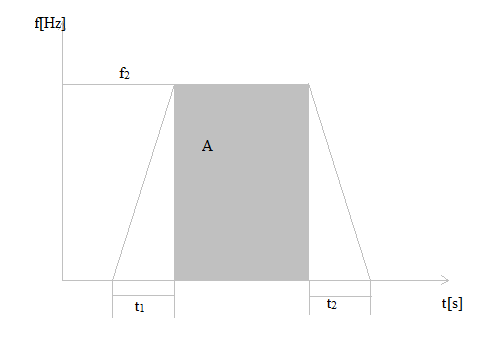
Tỉ số truyền : i = 1

Khối lượng riêng của vật liệu làm pulley ρ = 8,0×103 [kg/m3]

Hệ số an toàn

Kết quả tính toán thông số chọn động cơ trục X:

Từ kết quả đăng ký trên thì ta chọn động cơ trên thị trường của hãng Nema 17 công với hãng Nema 16 đê từ đó sử dụng chúng từu kết quả trên thi trường mà ta chọn được .qua kết quả tính toán ở trên thì nhận thấy rằng ta cùng co thể chọn hãng động cơ trên thì có thể chọn 3 động cơ giông nhau trên thi trường đó



Hình 2.1: Biểu đồ hoạt động

Vị trí tương đối của chuyển động của kết cấu khi động cơ đi được bước :

Số xung :



Tần số hoạt động :

=

Tốc độ hoạt động của đông cơ :

Tính mô men yêu cầu :

Momen tải :

Mômen tăng tốc Ta

Tính mô men quán tính của :

Pulley :

Tải :

Với kết quả tính toán trên tra datasheet của hãng ***PBC*** cũng như kết quả khảo sát động cơ trên thị trường Việt Namta chọn động cơ bước ***Nema 17*** cho 2 trục X, Y và động cơ ***Nema 16*** cho trục Z không những phù hợp với kết quả tính trọn mà còn dễ dàng tìm kiếm được chúng trên thị trường.

P = 11,76 (w)

A = = = 0,2

Trong đó :

A là công suất tính theo (A)

là khối lượng của cơ cấu` nâng : 2kg

G là khối lượng của vật nâng : 2kg

V tốc độ nâng (m/s) : 0,2m/s

g : gia tốc trọng trường(m/s) : 9,8 m/s2

hiệu xuất của cơ cấu nâng : 0,8%

K : hệ số ma sát : 1,2

*- Tính mô men của động cơ khi không có tải trọng*

M == 6,5 (Nm)

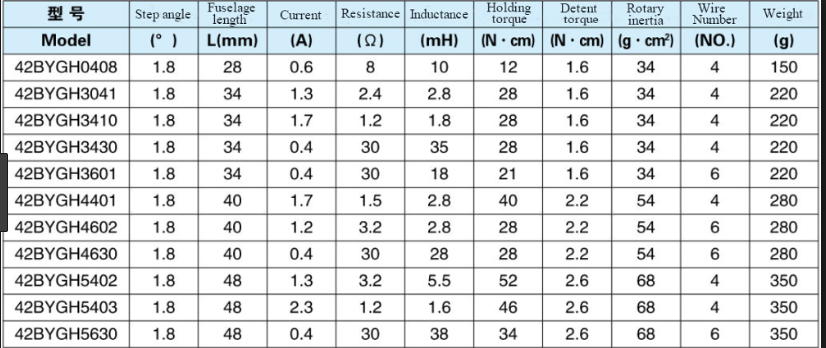
Trong đó :

P là trọng lượng của các pallet

W là tốc độ góc của động cơ

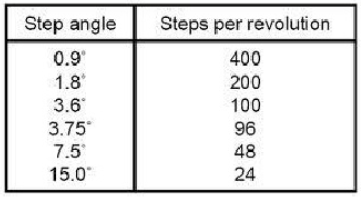
1. *Điều kiện chọn động cơ :*

Động cơ được chọn phải có công suất Pđc và số vòng quay nđc thỏa mãn



*Hình 2.2: Bảng thông số chọn động cơ*

***-*** *Bảng step của một số động cơ bước thông dụng* ***:***



**Chọn động cơ :** Động cơ bước

Loại động cơ : Nema 17 – 1.2A – 42BYGH4602

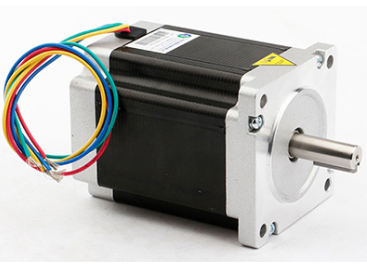
Kích thước : 42 x 42 x40 mm

Điện áp : 12 V

Mômen xoắn : 0.55 Nm

Trọng lượng : 280 gam

Góc bước : 1,8



*Hình 2.3: Động cơ bước Nema 17*

*- Tính toán chọn động cơ cho bàn xe trượt ngang:*

- Trọng lượng bàn xe : Go = 3 kg = 30 N.

- Trọng lượng cơ cấu nâng : Q = 2 kg = 20 N.

- Tốc độ di chuyển : vn = 0,4m/s.

Bàn xe chạy thì động cơ được gắn vào dây đai khi đó dây đai sẽ gắn vào bàn

xe và kéo chạy ngang.

P = 29,4 (w)

A = = = 0,5 (A)

Trong đó :

A là công suất tính theo (A)

là khối lượng của cơ cấu nâng : 2kg

G là khối lượng của bàn xe : 3kg

V tốc độ nâng (m/s) : 0,2m/s

g : gia tốc trọng trường(m/s) : 9,8 m/s2

hiệu xuất của cơ cấu nâng : 0,8%

K : hệ số ma sát : 1.2

- Chọn động cơ

Cần chọn 1 động cơ gắn vào dây đai và dây đai gắn với bàn xe di động

Loại động cơ : Nema 17 – 42BYGH4401

Điện áp : 12V

Công suất : 30W Góc bước : 1.8

### 2.2.3. Tính toán bộ truyền đai :

Thông thường động cơ NEMA17 sử dụng loại đai răng GT2 và đai MXL. Đây là các loại đai đã được tiêu chuẩn hóa đi kèm puly tiêu chuẩn dễ dàng mua được phù hợp với khả năng tiếp cận của sinh viên. So với MXL puly GT2 chống phản ứng dữ dội (anti-backlash) tốt hơn. Vì vậy, ta chọn loại đai răng GT2.



Hình 2.4: Sơ đồ bộ truyền đai

a. Quy trình tính toán bộ truyền đai :

- Lực kéo cần thiết :

Trong đó :

.g

.g.s

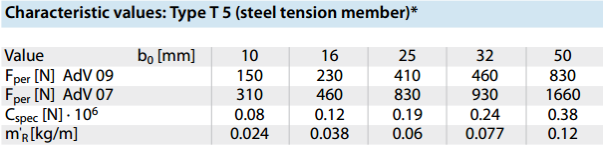
- Tính công suất bộ truyền : [kw]

- Tính chiều dài đai l [mm] và khối lượng đai  [kg]:

với i = 1

*l=+2e+*với i  1

; : tra bảng sau



Bảng 2.5. Tra thông số khối lượng đai Mr

*Tính toán bánh đai : Pulley*

- Tính số răng : Chọn theo tiêu chuẩn

- Đường kính vòng chia : ****

*Tính các thông số của bộ truyền:*

- Sau khi có mô đun của bộ truyền. Ta tiến hành tính thông số còn lại của bộ

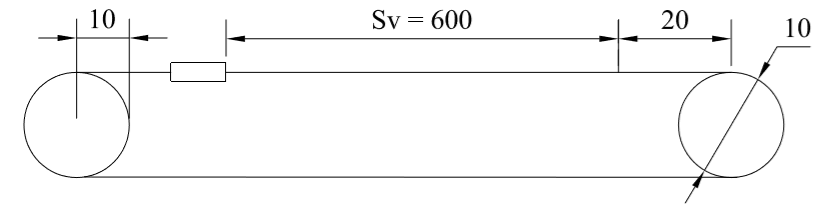
truyền :

- Tính số răng bánh bị động thông qua tỉ số truyền :

- Số răng đai :

- Số răng đồng thời ăn khớp trên đai nhỏ :

Trong đó  : góc ôm trên bánh đai nhỏ



Hình 2.6: Bộ truyền đai với thông số sơ bộ.

Quãng đường chuyển động :  = 600mm

Vận tốc cắt : 5m/ph

Gia tốc : 0,75 m/s2

Khối lượng tải : = 2 kg

Hệ số ma sát : μ = 0,03

Khối lượng đai sơ bộ : m = 0,01 kg

Chọn sơ bộ đường kính bánh đai : D = 10 mm

Tỉ số truyền : 

Tốc độ quay bánh chủ động : 

Các khoảng cách trục sơ bộ được lấy kích thước trên hình vẽ.

Tính chọn đai :

Bước đai :

Tham khảo catolog của hãng **SDP/SI** chọnp = 2mm

Bề rộng đai : 

Tham khảo catolog của hãng **SDP/SI** chọnb = 6mm

Chọn puley đai :

Bước pulley = bước đai p = 2mm

Đường kính vòng chia :

Theo như sơ bộ : Ta chọn D = 10 mm Bước puley p = 2 mm nên ta tính được số răng sơ bộ z = 15 răng.

Suy ra ta chọn z = 20theo tiêu chuẩn mà hãng ***SPD/SI*** đã thiết kế

Tính lại đường kính vòng chia :



## 2.3 Các thành phần cơ bản và thông số kỹ thuật của hệ thống

### 2.3.1. Bộ khung thép

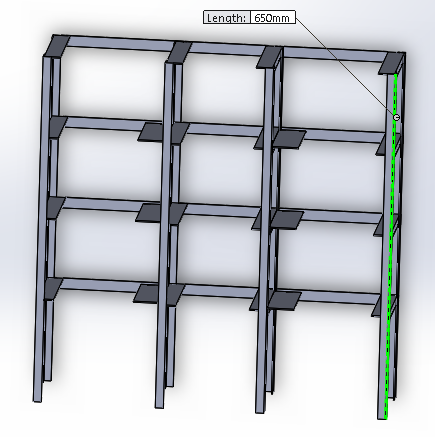
- Được làm bằng thép tấm 2 cạnh vuông góc thông số : 20 x 2 mm

Đặc điểm : thép tấm 2 cạnh, rất tiện lợi để hàn, hơn nữa lại chịu lực

tốt và bền bỉ , cho nên được sửu dụng rộng dãi trong các ngành công nghiệp

trên toàn thế giới. Chính vì vậy nhóm chúng em chọn khung thép để hoàn

lên mô hình này.



*Hình 2.8. Kết cấu và thông số kích thước*

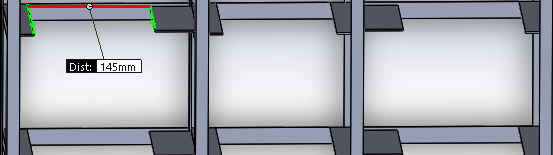
*- Kiểm nghiệm bền cho khung thép khi chịu tải trọng :*

Vì tải trọng tác dụng lên khung xe và pallet là không đáng kể nên với lựa chọn

thông số kích thước của mô hình thì hoàn toàn đảm bảo yêu cầu bền

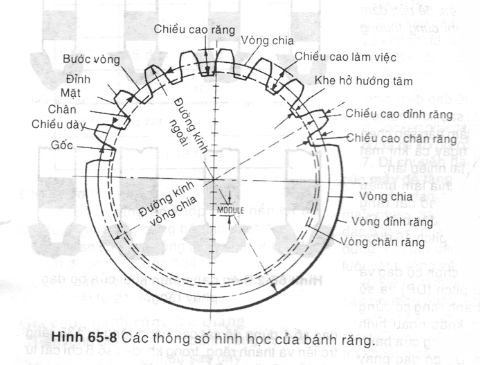
### 2.3.2. Pallet để xe

Sử dụng 2 tấm nhôm được hàn với nhau và gắn với khung.



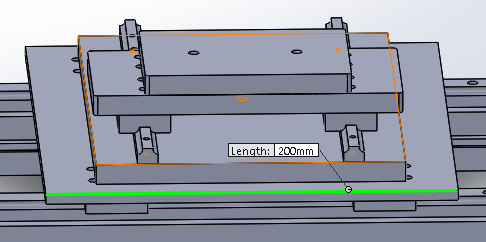
*Hình 2.9: Thông số kích thước Pallet*

### 2.3.3. Bánh răng quay



*Hình 2.10: Thông số bánh răng*

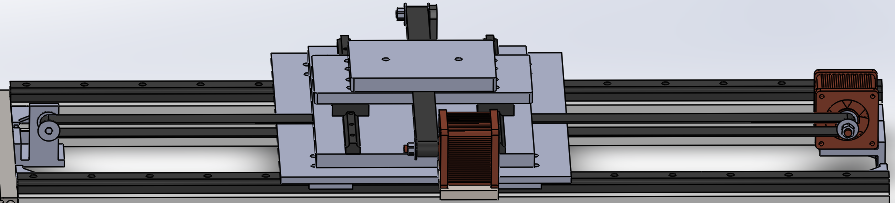
### 2.3.4. Bàn xe di động



*Hình 2.11: Thông số hình học của bàn xe*

### 2.3.5. Cơ cấu di chuyển ngang

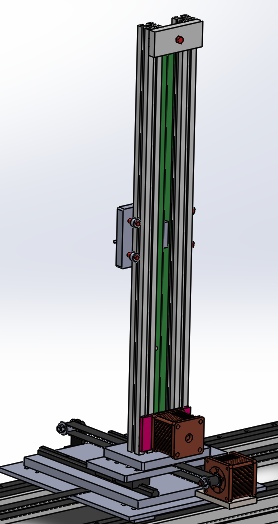
Bao gồm những chi tiết như : Động cơ , bàn xe di động , con trượt, pulley



*Hình 2.12: Sơ đồ động cơ cấu di chuyển ngang*

### 2.3.6. Cơ cấu di chuyển đứng

Bao gồm những chi tiết như : Động cơ, dây đai , bánh xe, bàn trượt, pulley..



*Hình 2.13: Sơ đồ động cơ cấu di chuyển đứng*

### 2.3.7. Dây đai.

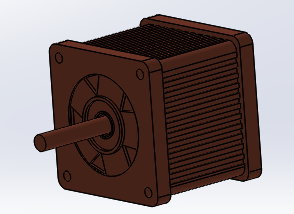
Chất liệu cao su, được chọn theo tiêu chuẩn



*Hình 7: Dây đai*

### 2.3.8. Động cơ

Chọn động cơ bước hãng Nema theo tiêu chuẩn



*Hình 2.14: Động cơ*

# CHƯƠNG 3 : THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN

## 3.1 Các thành phần thiết bị sử dụng

### 3.1.1 Arduino Mega 2560

Arduino Mega 2560 R3 là phiên bản nâng cấp của Arduino Uno R3 với số chân giao tiếp, ngoại vi và bộ nhớ nhiều hơn, mạch được thiết kế và sử dụng các linh kiện tương đương với phiên bản chính hãng trên Arduino.cc, phù hợp cho các ứng dụng cần nhiều bộ nhớ hoặc nhiều chân, cổng giao tiếp hơn so với Arduino Uno.

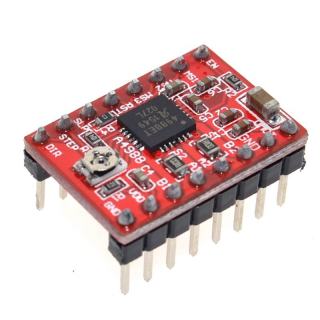


*Hình 3.1: Arduino Mega 2560*

**Thông số kỹ thuật:**

* Vi điều khiển chính: ATmega2560
* IC nạp và giao tiếp UART: ATmega16U2.
* Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC (khuyên dùng 7-9VDC để đảm bảo mạch hoạt động tốt. Nếu bạn cắm 12V thì IC ổn áp rất dễ chết và gây hư hỏng mạch).
* Số chân Digital: 54 (15 chân PWM)
* Số chân Analog: 16
* Giao tiếp UART : 4 bộ UART
* Giao tiếp SPI : 1 bộ ( chân 50 -> 53 ) dùng với thư viện SPI của Arduino
* Giao tiếp I2C : 1 bộ
* Ngắt ngoài : 6 chân
* Bộ nhớ Flash: 256 KB, 8KB sử dụng cho Bootloader
* SRAM: 8 KB
* EEPROM: 4 KB
* Xung clock: 16 MHz

### 3.1.2 Modul A4988

[](https://iotmaker.vn/images/detailed/2/3_1n6s-c6.jpg)

*Hình 3.2: Modul A4988*

Top of Form

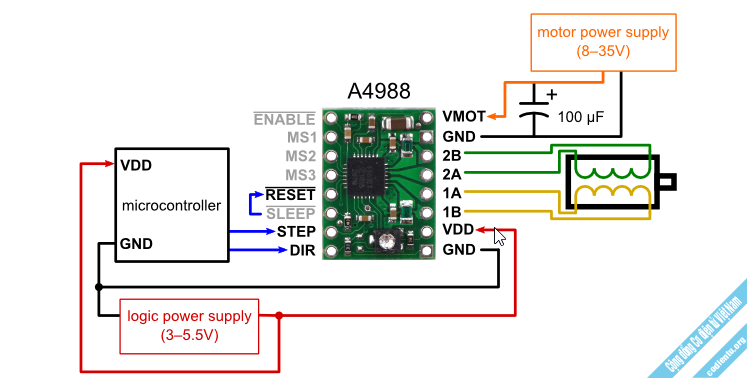
A4988 là một bộ điều khiển DMOS cực nhỏ với bộ chuyển đổi và bảo vệ quá dòng. A4988 có thể điều khiển được động cơ bước lưỡng cực với dòng điện lên đến 2A với mỗi cuộn dây.

Dưới đây là một số tính năng chính của sản phẩm:

* Dễ dàng điều khiển hướng quay và số bước quay
* 5 chế độ điều khiển: full step, haft step, 1/4, 1/8, 1/16
* Có thể điều chỉnh dòng tối đa thông qua một biến trở cho phép động cơ bước hoạt động với công suất tối đa
* Ngắt bảo vệ khi quá nhiệt, quá áp và quá dòng
* Bảo vệ ngắn mạch

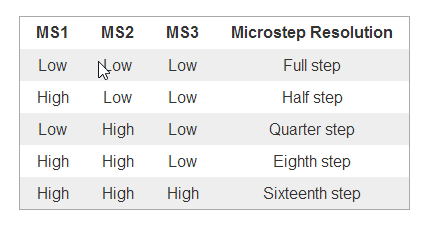
**Thông số kỹ thuật**

* Điện áp hoạt động: 8V~35V
* Dòng liên tục trên mỗi pha: 1A~2A
* Điện áp logic: 3V-5.5V
* Kích thước: 15,24 x 20,32cm



*Hình 3.3: Sơ đồ nói mạch*

* Lựa chọn chế độ full hay 1/2 hay 1/4.. sẽ được thông qua 3 pin MS1 MS2 MS3. Mình thường nối thẳng 3 pin này với công tắc bit 3p để dễ thiết lập từ trên phần cứng. Lưu ý là nếu thả nổi 3 pin này tức là mode full step.

​

* Bật tắt động cơ thì thông qua pin ENABLE, mức LOW là bật module, mức HIGH là tắt module.
* Điều khiển chiều quay của động cơ thông qua pin DIR.
* Điều khiển bước của động cơ thông qua pin STEP, mỗi xung là tương ứng với 1 bước ( hoặc vi bước).
* Hai chân Sleep với Reset nối với nhau.

### 3.1.3 Cảm biến khoảng cách hồng ngoại MH-B



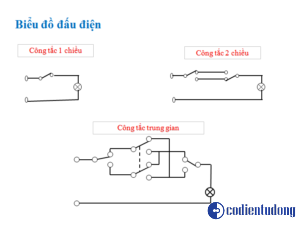
*Hình 3.4: Cảm biến MH-B*

Cảm biến có khả năng thích nghi với môi trường, có một cặp truyền và nhận tia hồng ngoại. Tia hồng ngoại phát ra một tần số nhất định, khi phát hiện hướng truyền có vật cản (mặt phản xạ), phản xạ vào đèn thu hồng ngoại, sau khi so sánh, đèn màu xanh sẽ sáng lên, đồng thời đầu cho tín hiệu số đầu ra (một tín hiệu bậc thấp). Khoảng cách làm việc hiệu quả 2 ~ 5cm, điện áp làm việc là 3.3 V đến 5V. Độ nhạy sáng của cảm biến được điều chỉnh bằng chiết áp, cảm biến dễ lắp ráp, dễ sử dụng,...Có thể được sử dụng rộng rãi trong robot tránh chướng ngại vật, xe tránh chướng ngại vật và dò đường...

### 3.1.4. Công tắc hành trình

  Công tắc hành trình là thiết bị chuyển đổi chuyển động cơ thành tín hiệu điện.Tín hiệu của công tắc hành trình phục vụ cho quá trình điều khiển và giám sát.

Có nhiều chủng loại công tắc hành trình tùy theo ứng dụng riêng biệt có thể phù hợp với từng ứng dụng về kích thước, chức năng, và môi trường hoạt động. Hiện nay trên thị trường có rất nhiều hãng sản xuất công tắc hành trình.



*Hình 3.5: Sơ đồ đấu dây công tắc hành trình*

*Nguyên lý hoạt động:*

Dùng để đóng cắt mạch dùng ở lưới điện hạ áp Nó có tác dụng giống như nút ấn động tác ấn bằng tay được thay thế bằng động tác va chạm của các bộ phận cơ khí, làm cho quá trình chuyển động cơ khí thành tín hiệu điện.

Công tắc hành trình là thiết bị chuyển đổi chuyển động cơ thành tín hiệu điện.Tín hiệu của công tắc hành trình phục vụ cho quá trình điều khiển và giám sát.

Có nhiều chủng loại công tắc hành trình tùy theo ứng dụng riêng biệt có thể phù hợp với từng ứng dụng về kích thước, chức năng, và môi trường hoạt động. Hiện nay trên thị trương có rất nhiều hãng sản xuất công tắc hành trình. Tuy nhiên, trên thị trường Việt Nam phổ biến 2 hãng công tắc hành trình là: Omron, Hanyoung

Chúng ta cùng tìm hiểu những ưu điểm của các hãng sản xuất công tắc hành trình:

Công tắc hành trình Omron:

Ưu điểm:

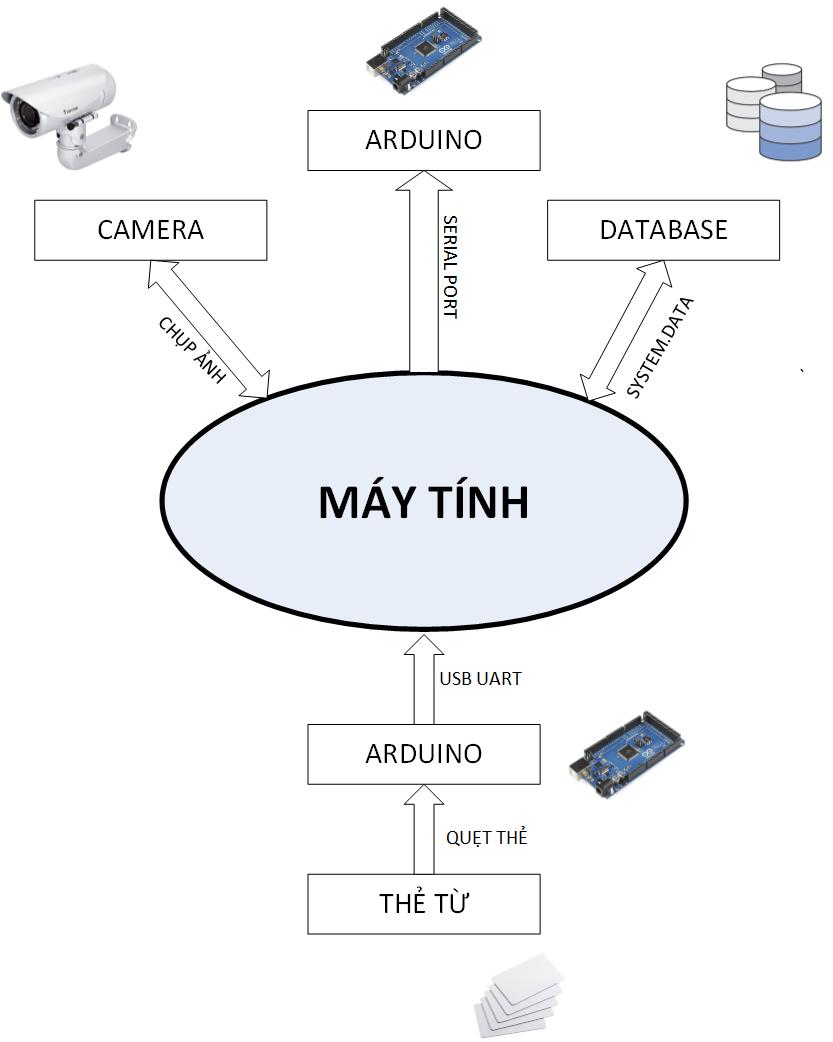
– Có mặt trên thị trường sớm, phổ biến trong công nghiệp

– Có nhiều loại công tắc hành trình với kích thước khác nhau

– Độ an toàn cao, tuổi thọ cao, xuất xứ Nhật

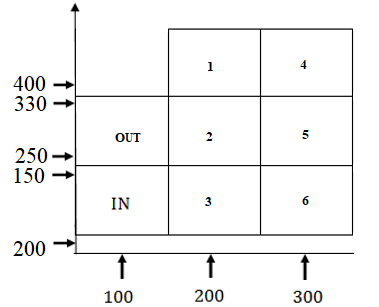
## 3.2 Thuật toán điều khiển

### 3.2.1 Tổng quan về hệ thống điều khiển

****

*Hình 3.6: Tổng quan hệ thống*

### 3.2.2. Thuật toán điều khiển Arduino

****

*Hình 3.7 Sơ đồ vị trí cảm biến và công tắc hành trình*

Các cảm biến được gắn trên trục Z và trục X. Như vậy để lấy được xe và gửi được xe cần dựa vào các cảm biến này. Các tọa độ của xe theo cảm biến là:

- Vị trí 1: (200;340;400)

- Vị trí 2: (200;150;250)

- Vị trí 3: (200; 0;200)

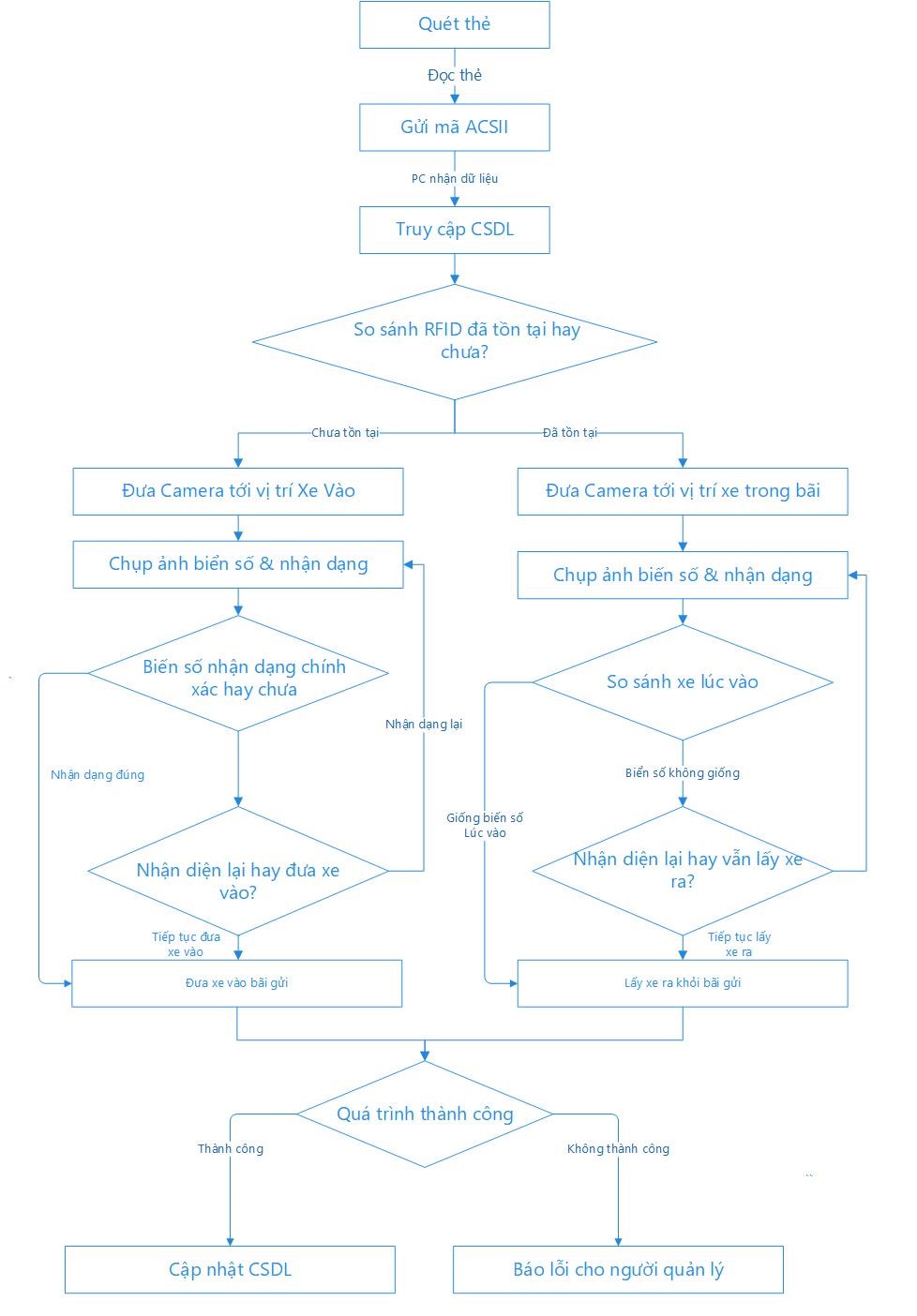
- Vị trí 4: (300;340;400)

- Vị trí 5: (300;150;250)

- Vị trí 6: (300; 0;200)

Arduino sẽ điều khiển các động cơ theo trục X;Y; Z với các INPUT là các cảm biến và công tắc hành trình theo các tọa độ ở trên. Các thông số đầu vào là cảm biến và công tắc hành trình sẽ là điều kiện để phục vụ cho việc dừng hay chạy các động cơ trên các trục X; Y; Z.

### 3.2.3 Thuật toán điều khiển từ máy tính

****

*Giải thích:*

1. Thẻ từ được quét vào thiết bị đọc thẻ, thiết bị sẽ có được chuỗi dữ liệu gồm 7 mã ACSII để lên máy tính.

2. Thiết bị gữi chuỗi mã ACSII và máy tính nhận dữ liệu.

3. Từ chuỗi mã ACSII, rồi tiến hành truy cập cơ sở dữ liệu.

4. Trong cơ sở dữ liệu, máy tính tiến hành kiểm tra xem mã thẻ đã tồn tại trước đó hay chưa? (có nghĩa là chiếc thẻ từ đã được quét trước đó và tiến hành quá trình gữi xe thành công, thì mã thẻ sẽ được lưu trong cơ sở dữ liệu và gắn liền với chiếc xe đó).

5. Nếu chiếc thẻ chưa được liên kết với chiếc xe nào trong bãi. Hệ thống Arduino sẽ ra lệnh cho khung nâng đưa camera tới vị trí xe vào.

6. Quá trình chụp ảnh và nhận dạng biển số xe bắt đầu.

7. Nếu nhận dạng ảnh sai (không thể chụp được hình, nhận dạng không đủ ký tự…).

8. Máy tính sẽ đưa ra lựa chọn: có muốn tiếp tục đưa xe vào hay nhận dạng lại một lần nữa?

9. Máy tính ra lệnh cho Arduino đưa xe vào vị trí còn trống trong bãi xe.

10. Nếu chiếc thẻ đã được liên kết với một chiếc xe nào đó trong bãi (đã được quét trước đó và đưa xe vào thành công). Hệ thống Arduino sẽ ra lệnh cho khung nâng đưa camera tới vị trí xe hiện tại trong bãi.

11. Quá trình chụp ảnh và nhận dạng biển số xe bắt đầu.

12. Nếu biển số xe nhận dạng trùng với biển số xe lúc đưa vào?

13. Nếu biển số không giống thì máy tính đưa ra lựa chọn: tiếp tục lấy xe ra hay nhận dạng lại xe ra?

14. Máy tính ra lệnh cho Arduino lấy xe ra khỏi bãi giữ xe.

15. Nếu quá trình đưa xe vào hoặc lấy xe ra thành công.

16. Máy tính sẽ cập nhật dữ liệu cho cơ sở dữ liệu.

17. Nếu không thành công thì máy tính sẽ thông báo đã xãy ra lỗi trong quá trìnnh vận hành.

## 3.3 Công nghệ xử lý ảnh số

CHỤP ẢNH

TÁCH HÌNH BIỂN SỐ

CẮT KÝ TỰ

NHẬN DẠNG KÝ TỰ

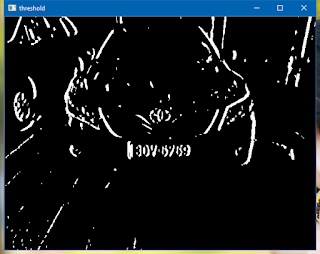
SẮP XẾP KÝ TỰ

*Hình 3.8 Thuật toán xử lý ảnh*

*Trình tự thực hiện:*

1. Camera nhận lệnh chụp ảnh từ C# và tiến hành chụp hình. Hình ảnh sẽ được thiết bị truyền máy tính và được lưu trữ trong chương trình C#.

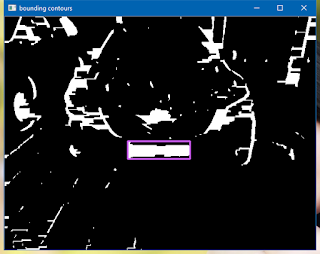
2. Sau khi có hình ảnh biển số xe, bằng các bước tiền xử lý ảnh trên thư viện OpenCV là biến đổi ảnh xám, nhị phân hóa ảnh ta được hình dạng ảnh như sau:



*Hình 3.9: Ảnh biển số sau khi nhị phân hóa*

Vì biển số xe có đặc trưng là các ký tự màu đen trên hình chữ nhật nền trắng do đó sau khi chúng ta nhị phân hóa, ảnh sẽ được các vùng có độ tương phản cao nhất.

3. Từ biển số xe ta tiến hành lấy đường viền ảnh có khả năng cao nhất là biển số. Ta sử dụng thuật toán Morphology để làm nhòe các vùng có ký tự. Tuy nhiên do trong ảnh có rất nhiều vùng được làm nhòe nên ta phải định nghĩa một kích thước có thể là biển số nhất. Áp dụng những đặc trưng trên ta tìm được Contour biển số.



*Hình 3.9: Ảnh biển số sau khi tìm contour*

4. Trong vùng biển số đã tìm được, ta tiếp dùng câu lệnh Contour để tìm các contour là ký tự của biển. Từ các đường viền này ta cắt riêng lẻ từng ký tự của biển số xe. Thường các biển số xe hiện tại thường có 7 hoặc 8 ký tự nên chúng ta chỉ quan tâm tới 7-8 ký tự đó. Do đó để để biết contours nào đang chứa ký tự thì dựa vào các quy luật gồm:

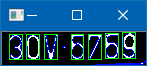
– Width luôn nhỏ hơn height.

– Chiều cao tối thiểu 15 pixels.

– Tỉ lệ aspect (width/height) phải lớn hơn 0.2 và nhỏ hơn 0.5. Đây là tỉ lệ với biển số nhìn trực diện, nếu có biển số nằm nghiêng phải thay đổi cho phù hợp

– Tỉ lệ pixel màu trắng trong mỗi ký tự phải nhỏ hơn 80%.

Dựa vào đặc trưng trên ta cắt được các ký tự ra để phục vụ bước tiếp theo:



*Hình 3.10: Ảnh sau khi đã cắt kí tự*

5. Với những mẫu ký tự được cắt ra, ta đưa chúng vào thư viện nhận dạng ký tự có sẵn trong EmguCV. Từ đó ta sẽ có được các ký tự được chuyển từ dạng tương tự (hình ảnh) sang dạng số (mã ACSII).

6. Mặc dù đã nhận dạng được các ký tự, nhưng lúc này thứ tự của chúng vẫn rất lộn xộn. Do đó bước cuối cùng là sắp xếp chúng theo thứ tự dựa vào vị trí cắt của các ký tự ở bước 4. Và cuối cùng ta có được kết quả là các chữ số và chữ cái chính xác, cũng như sắp xếp hợp lý như biển số đã chụp.

## 3.4 Công nghệ thẻ từ RFID

Công nghệ thẻ từ cho phép một thiết bị đọc thông tin chứa trong chip không cần tiếp xúc trực tiếp ở khoảng cách xa. Công nghệ này cho ta phương pháp truyền nhận dữ liệu từ một điểm đến điểm khác.

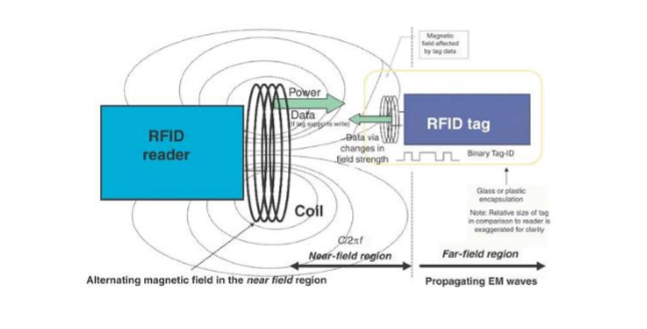
Các thành phần cơ bản của hệ thống thẻ từ:

- *Tag:* là thành phần không thể thiếu trong hệ thống, còn gọi là thẻ từ.

- *Reader:* là thành phần bắt buộc trong hệ thống, bên ngoài còn gọi là bộ phận đọc thẻ từ.

- *Reader anten:* cũng là thành phần bắt buộc của hệ thống, và ngày nay thì bộ phận này tích hợp sẵn trong bộ reader. Ngoài ra, còn có các bộ phận quan trọng trong hệ thống như: mạch điều khiển, cảm biến, các thiết bị truyền thông, …

Dạng đơn giản nhất được sử dụng hiện nay là hệ thống thẻ từ bị động làm việc như sau: reader truyền một tín hiệu tần số vô tuyến điện từ qua anten của nó đến một con chip. Reader nhận thông tin trở lại từ chip và gửi nó đến máy tính điều khiển đầu đọc và xử lý thông tin lấy được từ chip. Các chip không tiếp xúc, không tích điện, chúng hoạt động bằng cách sử dụng năng lượng nhận từ tín hiệu được gửi bởi reader. Và nguyên lý hoạt động được thể hiện ở sơ đồ:



*Hình 3.11: Sơ đồ nguyên lý hoạt động của RFID*

- Thiết bị RFID reader phát ra sóng điện từ ở một tần số nhất định, khi thiết bị RFID tag trong vùng hoạt động sẽ cảm nhận được sóng điện từ này và thu nhận năng lượng từ đó phát lại cho thiết bị RFID Reader biết mã số của mình. Từ đó thiết bị RFID reader nhận biết được tag nào đang trong vùng hoạt động.



*Hình 3.12: Một loại thẻ chip RFID phổ biến*

*Ứng dụng:*

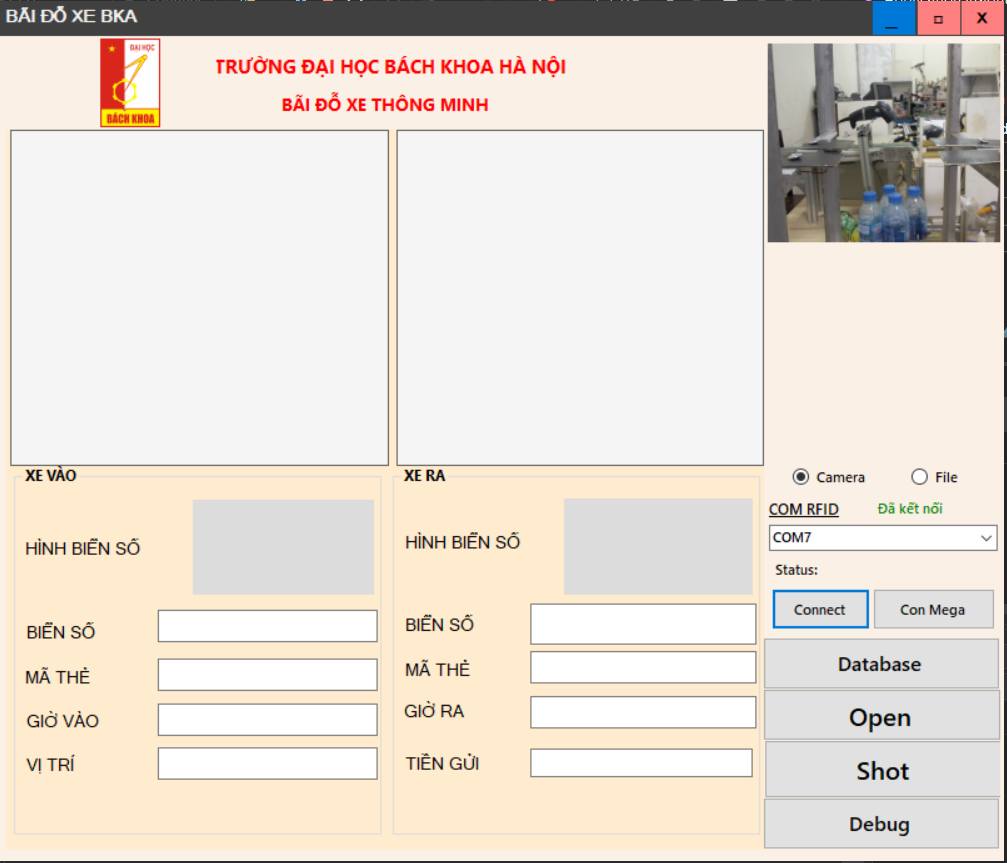
Ứng dụng tiêu biểu nhất của công nghệ này là chống mất trộm hàng hóa trong siêu thị. Thiết bị chip RFID (tag) được gắn với các hàng hóa trong đó. Thiết bị Reader và antenna được gắn bên ngoài cửa kiểm soát. Nếu một đồ vật chưa được tháo chip đi qua cửa kiểm soát thì thiết bị Reader dễ dàng nhận dạng thấy và phát cảnh bảo.

Một ứng dụng khác cũng được áp dụng công nghệ này mang đến lợi ích rất lớn là ứng dụng trong việc sản xuất khóa chống trộm xe máy.

## 3.5 Xây dựng chương trình máy tính

### 3.5.1 Thiết kế giao diện người dùng

Giao diện này giúp người dùng dễ dàng quan sát các hình ảnh chụp về từ camera, thực hiện nhận dạng biển số và hiển thị biển số cùng với mã thẻ được quét, thực hiện báo lỗi khi không nhận dạng được biển số xe. Đồng thời dễ dàng quản lý: Xe vào bãi, biển số, mã thẻ, giờ vào, vị trí gửi; Xe ra bãi: biển số, mã thẻ, vị trí lấy, tiền gửi.



*Hình 3.13: Giao diện điều khiển trên máy tính*

*Camera*: hiển thị hình ảnh liên tực lấy từ Camera.

*Hình chụp lúc xe ra vào*: mỗi lần xe ra hay xe vào hệ thống đều chụp lại hình ảnh nhằm lưu vào hệ thống vào dùng để nhận dạng biên số xe, nên đây là vùng để hiển thị ảnh chụp đó.

*Ô hiển thị thông tin của xe:* hiển thị đầy đủ thông tin liên quan đến chiếc xe đang ra hoặc vào bãi.

*Hình cắt biển số:* là hình biển số được cắt ra khi đưa hình ảnh chụp chiếc xe vào hệ thống xử lý ảnh.

Các nút bấm chức năng bao gồm:

*- Shot:* Nút bấm chụp ảnh hiện tại để đưa vào hệ thống.

*- Database:* Mở cơ sở dữ liệu để người sử dụng có thể dễ dàng quản lý xe trong bãi.

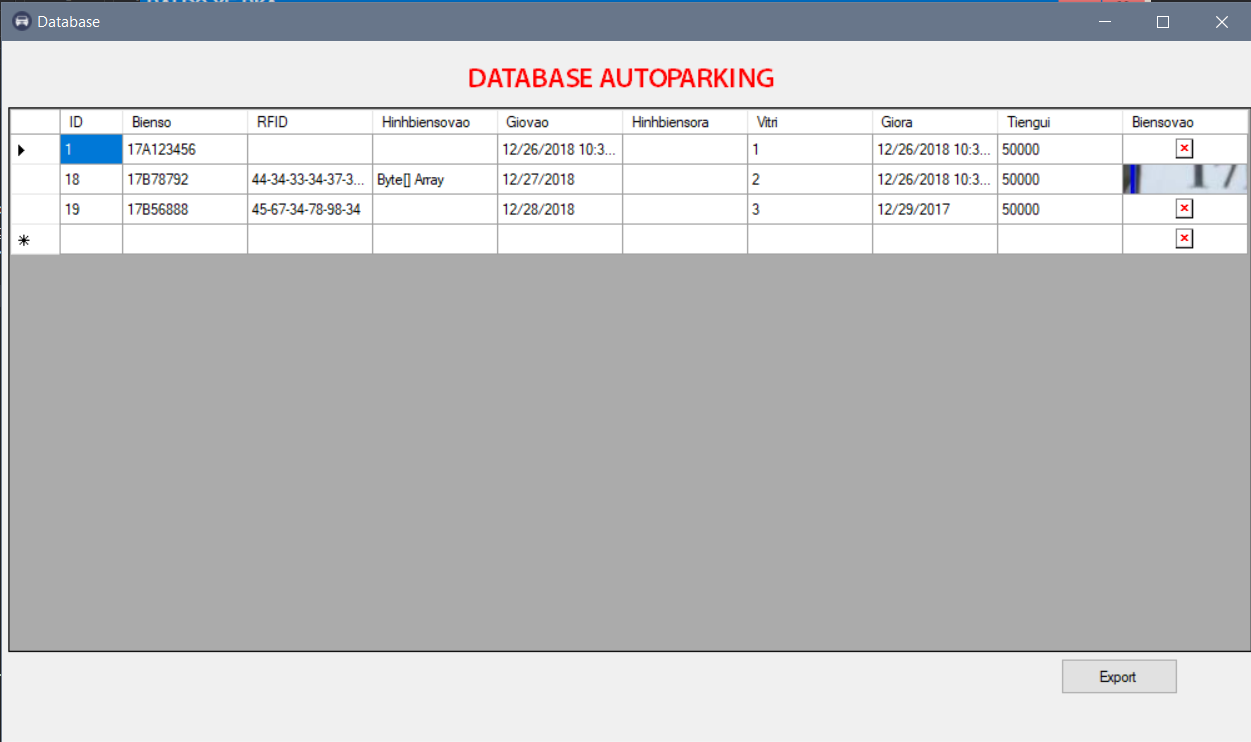
*- Cổng Connect:* để người sử dung có thể dễ dàng kết nối với các thành phần điều khiển như RFID hay hệ thống điều khiển Arduino.

Các tùy chọn cho người sử dụng là nhận diện bằng camera và nhận từ việc mở file ảnh có sẵn trên máy tính.

### 3.5.2 Cơ sở dữ liệu bãi đỗ xe

*Cơ sở dữ liệu* là một tập hợp những thông tin được tổ chức để dễ dàng trong việc tạo lập, cập nhập và khai thác thông tin. Cơ sở dữ liệu được duy trì dưới dạng một tập hợp các tập tin trong hệ điều hành hay được lưu trữ trong các hệ quản trị cơ sở dữ liệu.

Cơ sở bãi đỗ xe thông minh là cơ sở dữ liệu hướng đối tượng: dữ liệu cũng được lưu trữ trong các bảng dữ liệu nhưng các bảng có bổ sung thêm các tính năng hướng đối tượng như lưu trữ thêm các hành vi, nhằm thể hiện hành vi của đối tượng. Mỗi bảng xem như một lớp dữ liệu, một dòng dữ liệu trong bảng là một đối tượng. Các hệ quản trị có hỗ trợ cơ sở dữ liệu hướng đối tượng như: MS SQL Server, Oracle, Postgres. Ở đồ án này em xây dựng cơ sở dữ liệu trên hệ SQL Server của hãng Microsoft.

****

*Hình 3.14: Cơ sở dữ liệu*

*Cơ sở dữ liệu:*

Gồm các thuộc tính:

- *Biensovao* kiểu dữ liệu string, lưu các biển số mà chương trình nhận diện.

- *RFID* kiểu NVChar lưu các mã RFID quét được.

- *Hinhbiensovao* kiểu dữ liệu Byte dạng Array lưu các hình biển số vào.

- *Giovao* dạng DateTime lưu chính xác Giờ vào

- *Hinhbiensora* kiểu dữ liệu Byte dạng Array lưu các hình biển số ra.

- *Vitri* kiểu dữ liệu Int, lưu vị trí xe.

- *Giora* dạng DateTime lưu chính xác Giờ ra

- *Tiengui* kiểu dữ liệu Int, lưu số tiền khách đã trả.

- *Biensovao* kiểu dữ liệu Image lưu ảnh biển số được cắt ra từ ảnh chụp camera.

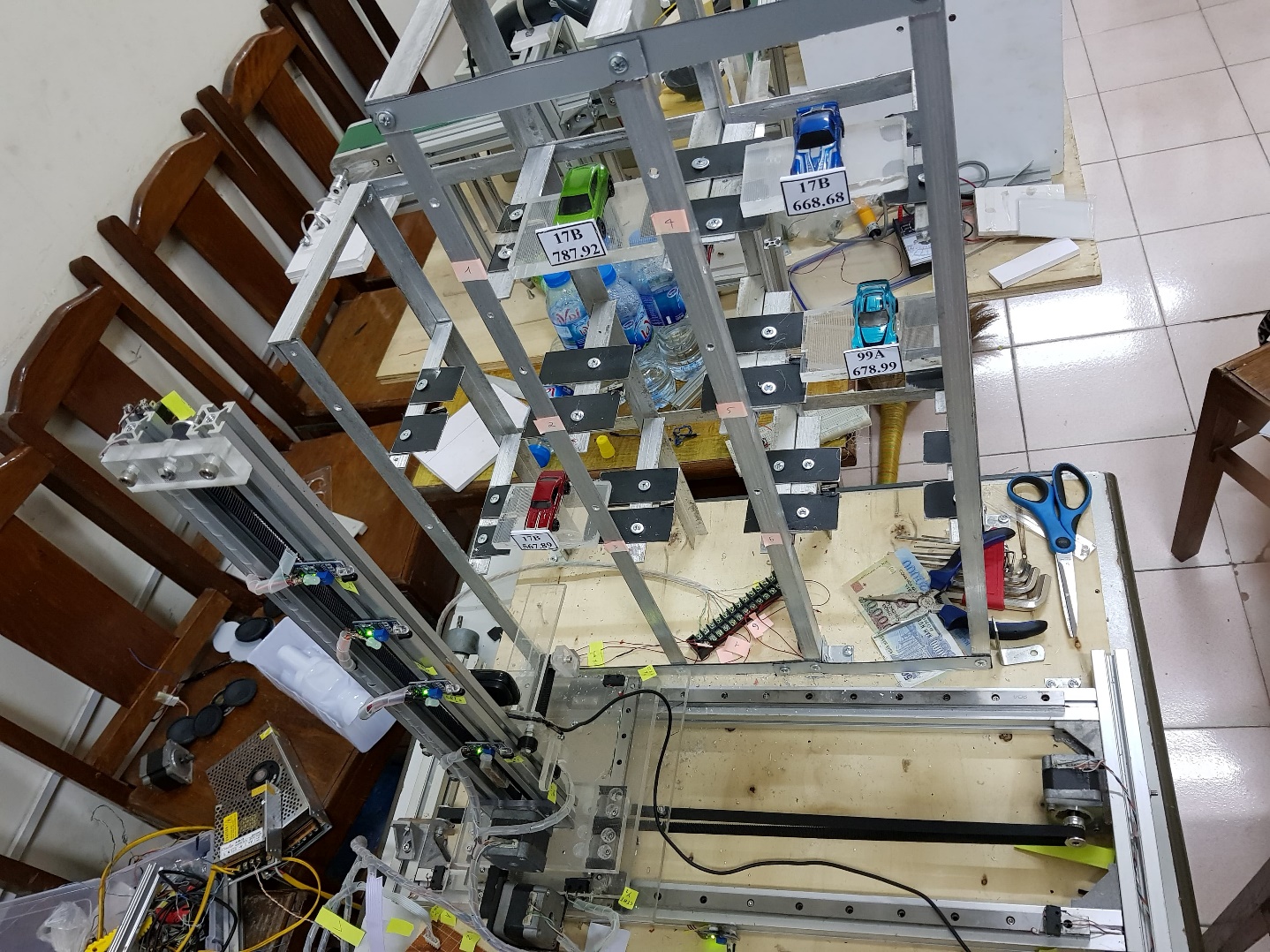
### 3.5.3 Hệ thống giám sát



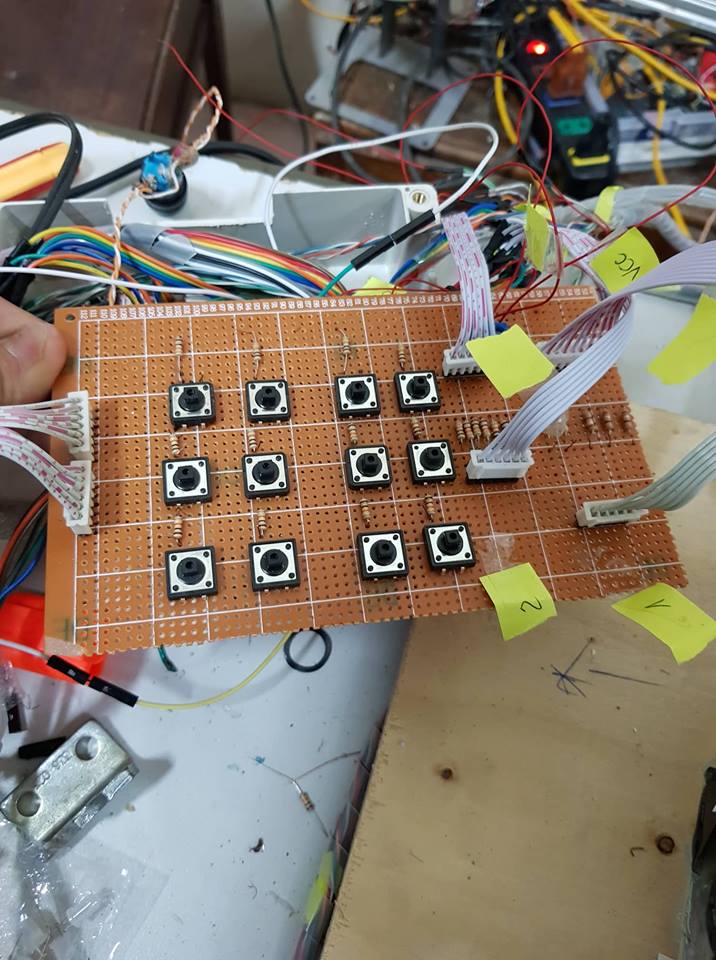
- Hiển thị toàn bộ xe, vị trí để nhân viên bãi đỗ xe có thể dễ dàng quản lý xe trong bãi, cũng như biết được số vị trí còn trống hiện tại trong bãi.

# CHƯƠNG 4:MÔ HÌNH THỰC TẾ

## 4.1. Thi công mô hình

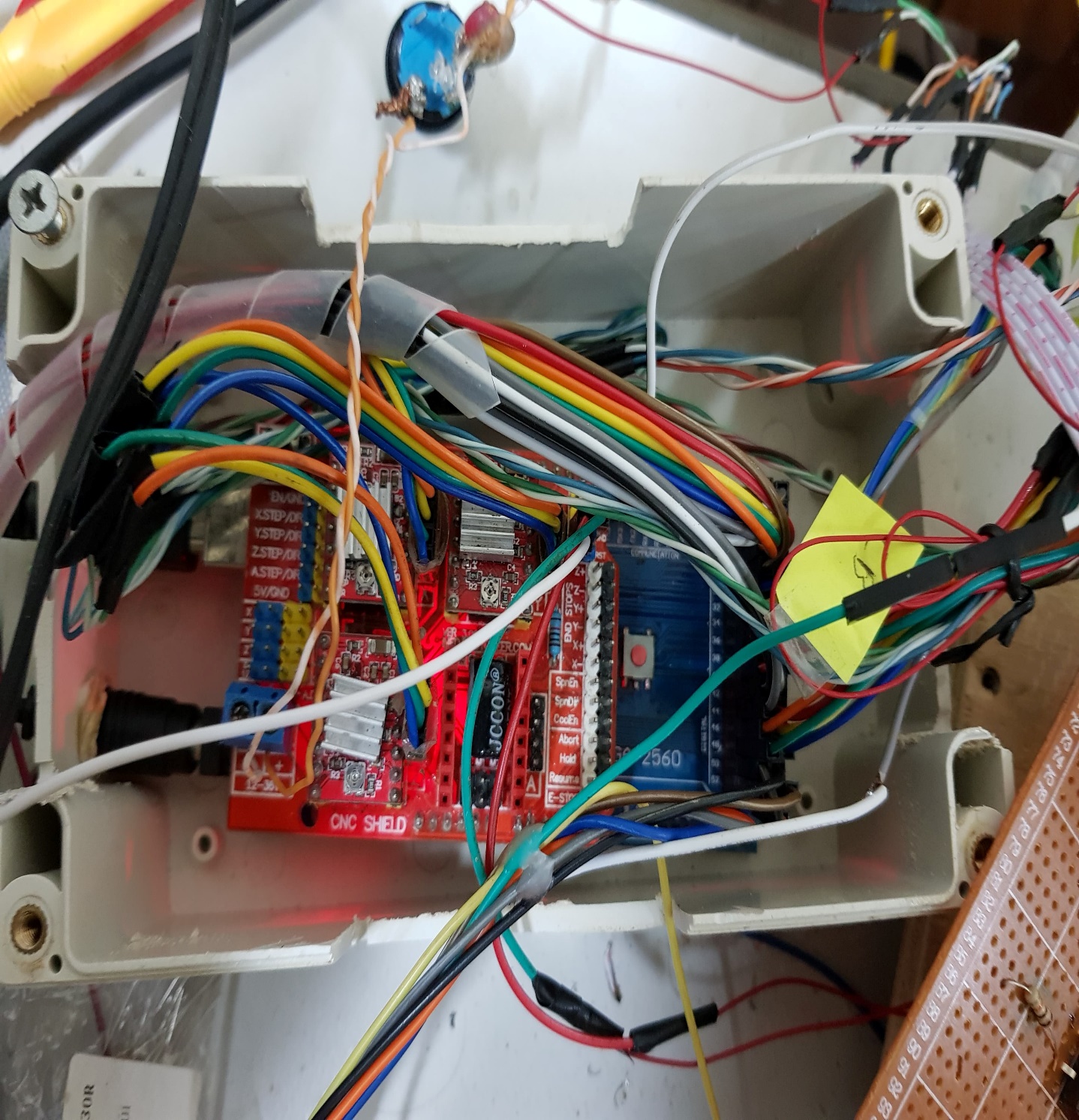


Hình 4.1: Mô hình thực tế



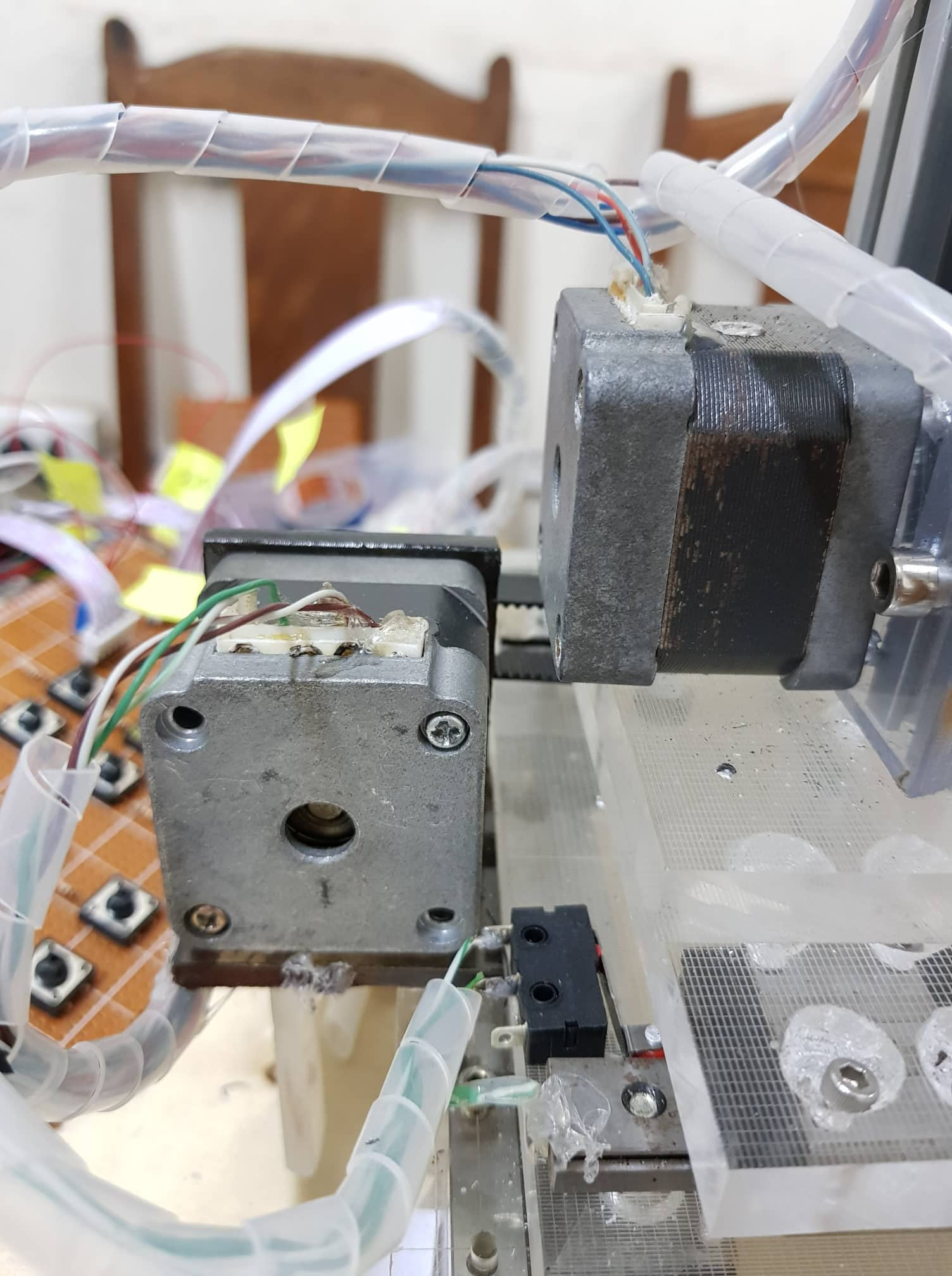
Hình 4.2: Hệ thống nút nhấn

Khi khách hàng mất thẻ, sau khi chứng minh được xe trong bãi là của mình, nhân viên bãi gửi sẽ tiến hành khởi động chế độ điều khiển bằng tay đưa xe của khách ra ngoài, đồng thời xóa xe trong CSDL.



Hình 4.3: Tủ điện của hệ thống

Hình 4.4: Đầu đọc thẻ từ RFID



Hình 4.4: Động cơ bước Nema 17



Hình 4.5: Camera của hệ thống

# CHƯƠNG 5 : KẾT LUẬN

## 5.1. Phương hướng và định hướng phát triển của đê tài

Để tối ưu hóa diện tích mặt bằng, chúng ta có thể xây dựng một hệ thống khung đôi, mỗi khung có nhiều tầng, khung nâng có thể lấy xe ở cả hai hướng. Cơ cấu nâng gầm xe chưa thực sự hoàn hảo, bởi một số dòng xe có kích thước gầm khác xa so với các dòng xe thông thường. Do đó, chúng em mong muốn phát triển cơ cấu nâng gầm thành cơ cấu nâng bánh. Vì cơ cấu nâng bánh đảm bảo tính chính xác, và lúc khung nâng di chuyển xe ổn định hơn cơ cấu nâng gầm. Ngoài hệ thống nhận diện bằng thẻ từ kếp hợp nhận dạng biển số, chúng ta còn có thể phát triển hệ thống quản lý bằng dấu vân tay, kết hợp với hệ thống nhận diện bảng số. Kết hợp hệ thống chuyển nguồn tự động ATS để phòng trường hợp mất điện lưới gây sự cố cho hệ thống. Sử dụng nhiều cảm biến tăng thêm độ chính xác và tính an toàn cho hệ thống.

## 5.2 . Ứng dụng vào thực nghiệm

Đề tài:“Thiết kế và thi công mô hình bãi giữ xe ô tô tự động” là một mô hình thực tế. Phần quan trọng của đề tài là thiết kế và thi công được mô hình bãi giữ xe ô tô cất và lấy xe một cách tự động có sử dụng hệ thống bảo mật bằng thẻ từ kết hợp nhận dạng biển số xe. Để hoàn thành đề tài nhóm thực hiện đã sử dụng được những tính năng của arduinocũng như các thiết bị, phương pháp truyền gửi dữ liệu …. Việc ứng dụng đề tài vào thực tế là rất cần thiết đối với tình hình thực tế hiện nay của nước ta tại các thành phố lớn.Tuy nhiên, đề tài khi áp dụng thực tế cần xem xét đến kinh phí xây dựng vì việc thi công mô hình đòi hỏi kinh phí cao.

# TÀI LIỆU THAM KHẢO

\*Tài liệu tiếng Việt:

[1] Trịnh Chất, Lê Văn Uyển .Tính toán thiết kế dẫn động cơ khí tập 1 & 2

[2] Lương Mạnh Bá, Nguyễn Thanh Thủy (2002). Nhập Môn Xử lý ảnh số, nhà xuất bản khoa học và Kỹ thuật.

\*Tài liệu từ các Website:

[3] http//www.Arduino.vn.

[4] http//www.Arduino.cc.

# CODE CHƯƠNG TRÌNH

#include <TimerOne.h>

#include <SPI.h>

#include <MFRC522.h>

#define RST\_PIN 15

#define SS\_PIN 53

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

MFRC522::MIFARE\_Key key;

//RFID

# define EN 8 // stepper motor enable , active low

# define X\_DIR 5 // X -axis stepper motor direction control

# define Y\_DIR 6 // y -axis stepper motor direction control

# define Z\_DIR 7 // z axis stepper motor direction control

# define X\_STP 2 // x -axis stepper control

# define Y\_STP 3 // y -axis stepper control

# define Z\_STP 4 // z -axis stepper control

# define sensor1 37

# define sensor2 38

# define sensor3 39

# define sensor4 40

# define sensor5 41

# define TIMER\_US 10000000

void RFID(){

if ( ! mfrc522.PICC\_IsNewCardPresent())

return;

if ( ! mfrc522.PICC\_ReadCardSerial())

return;

dump\_byte\_array(mfrc522.uid.uidByte, mfrc522.uid.size);

mfrc522.PICC\_HaltA();

mfrc522.PCD\_StopCrypto1();

x1 = digitalRead(limitSW1);

x2 = digitalRead(limitSW2);

x3 = digitalRead(limitSW3);

x4 = digitalRead(limitSW4);

x5 = digitalRead(limitSW5);

//x6 = digitalRead(limitSW6);

x = x1 + x2\*2 + x3\*2\*2 + x4\*2\*2\*2 + x5\*2\*2\*2\*2;

}

void Guixeo(){

if (digitalRead(limitSWy) == LOW && digitalRead(limitSWx) == LOW && digitalRead(limitSWz) == LOW){

while (digitalRead(limitSWy2) == HIGH){

digitalWrite(Y\_DIR,true);

digitalWrite(Y\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Y\_STP,LOW);

delayMicroseconds(200);

}}

if (digitalRead(limitSWx) == LOW && digitalRead(limitSWz) == LOW){

while ( digitalRead(sensor1) == HIGH){

digitalWrite(Z\_DIR,true);

digitalWrite(Z\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Z\_STP,LOW);

delayMicroseconds(200);

}

//if (digitalRead(sensor1) == LOW){

while (digitalRead(limitSWy) == HIGH){

digitalWrite(Y\_DIR,false);

digitalWrite(Y\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Y\_STP,LOW);

delayMicroseconds(200);

// }

}

//if (digitalRead(limitSWy) == LOW && digitalRead(sensor1) == LOW) {

while (digitalRead(sensor5) == HIGH){

digitalWrite(X\_DIR,false);

digitalWrite(X\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(X\_STP,LOW);

delayMicroseconds(200);

// }

}

// if (digitalRead(sensor5) == LOW && digitalRead(sensor1) == LOW){

while ( digitalRead(limitSWz2) == HIGH){

digitalWrite(Z\_DIR,true);

digitalWrite(Z\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Z\_STP,LOW);

delayMicroseconds(200);

}

}

if ( digitalRead(limitSWz2) == LOW){

while (digitalRead(limitSWy2) == HIGH){

digitalWrite(Y\_DIR,true);

digitalWrite(Y\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Y\_STP,LOW);

delayMicroseconds(200);

}}

if ( digitalRead(limitSWy2) == LOW){

while (digitalRead(sensor4) == HIGH ){

digitalWrite(Z\_DIR,false);

digitalWrite(Z\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Z\_STP,LOW);

delayMicroseconds(200);

}

}

if (digitalRead(sensor4) == LOW && digitalRead(sensor5) == LOW ){

while (digitalRead(limitSWy) == HIGH){

digitalWrite(Y\_DIR,false);

digitalWrite(Y\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Y\_STP,LOW);

delayMicroseconds(200);

}}

void Layxeo(){

if (digitalRead(limitSWy) == LOW && digitalRead(limitSWx) == LOW && digitalRead(limitSWz) == LOW){

while (digitalRead(sensor5) == HIGH ){

digitalWrite(X\_DIR,false);

digitalWrite(X\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(X\_STP,LOW);

delayMicroseconds(200);

}

}

if (digitalRead(sensor5) == LOW & digitalRead(limitSWz) == LOW){

while(digitalRead(sensor4) == HIGH){

digitalWrite(Z\_DIR,true);

digitalWrite(Z\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Z\_STP,LOW);

delayMicroseconds(200);

}

if (digitalRead(sensor4) == LOW){

while ( digitalRead(limitSWy2) == HIGH){

digitalWrite(Y\_DIR,true);

digitalWrite(Y\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Y\_STP,LOW);

delayMicroseconds(200);

}

}

while ( digitalRead(limitSWz2) == HIGH){

digitalWrite(Z\_DIR,true);

digitalWrite(Z\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Z\_STP,LOW);

delayMicroseconds(200);

}

if (digitalRead(limitSWz2) == LOW && digitalRead(sensor5) == LOW){

while (digitalRead(limitSWy) == HIGH){

digitalWrite(Y\_DIR,false);

digitalWrite(Y\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Y\_STP,LOW);

delayMicroseconds(200);

}

}

if ( digitalRead(limitSWy) == LOW){

while (digitalRead(limitSWx) == HIGH){

digitalWrite(X\_DIR,true);

digitalWrite(X\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(X\_STP,LOW);

delayMicroseconds(200);

}

}

if (digitalRead(limitSWx) == LOW && digitalRead(limitSWz2) == LOW && digitalRead(limitSWy) == LOW){

while (digitalRead(sensor3) == HIGH){

digitalWrite(Z\_DIR,false);

digitalWrite(Z\_STP,HIGH);

delayMicroseconds(200);

digitalWrite(Z\_STP,LOW);

delayMicroseconds(200);

}

}