

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

**XÂY DỰNG HỆ THỐNG RỬA XE TỰ ĐỘNG TRÊN NỀN  
VI ĐIỀU KHIỂN HỌ PIC**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

**XÂY DỰNG HỆ THỐNG RỬA XE TỰ ĐỘNG TRÊN NỀN  
VI ĐIỀU KHIỂN HỘ PIC**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

Sinh viên : Nguyễn Đức Hùng

Giáo viên hướng dẫn: Thạc sỹ Nguyễn Trọng Thắng

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

ĐỘC LẬP – TỰ DO – HẠNH PHÚC

-----o0o-----

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

## **NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên: Nguyễn Đức Hùng

Mã sinh viên: 1013102019

Lớp: ĐCL401

Ngành: Điện tự động công nghiệp

Tên đề tài: *Xây dựng hệ thống rửa xe tự động trên nền vi điều khiển họ PIC*

## NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp:

.....

## CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất :

Họ và tên : Nguyễn Trọng Thắng

Học hàm, học vị : Thạc sỹ

Cơ quan công tác : Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn : Toàn bộ đề tài

Người hướng dẫn thứ hai :

Họ và tên :

Học hàm, học vị :

Cơ quan công tác :

Nội dung hướng dẫn :

Đề tài tốt nghiệp được giao ngày 08 tháng 8 năm 2012

Yêu cầu phải hoàn thành trước ngày 09 tháng 11 năm 2012

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

Nguyễn Đức Hùng

Th.S Nguyễn Trọng Thắng

*Hải Phòng, ngày.....tháng.....năm 2012*

**HIỆU TRƯỞNG**

**GS.TS.NGŨT *Trần Hữu Nghị***

## PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

.....  
.....  
.....  
.....  
.....

2. Đánh giá chất lượng của Đ.T.T.N ( so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận thực tiễn, tính toán giá trị sử dụng chất lượng các bản vẽ...)

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

3. Cho điểm của cán bộ hướng dẫn:

*(Điểm ghi bằng số và chữ)*

Ngày ... tháng ... năm 2012

Cán bộ hướng dẫn chính

*(Họ tên và chữ ký)*

**NHẬN XÉT ĐÁNH GIÁ CỦA NGƯỜI CHẤM PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP**

1. Đánh giá chất lượng của Đ.T.T.N về các mặt thu thập và phân tích số liệu ban đầu, cơ sở lý luận chọn phương án tối ưu, cách tính toán chất lượng thuyết minh các bản vẽ giá trị lý luận và thực tiễn đề tài:

.....

.....

.....

.....

.....

2. Cho điểm của cán bộ chấm phản biện:

*(Điểm ghi bằng số và chữ)*

Ngày ..... tháng ..... năm 2012

Người chấm phản biện

# MỤC LỤC

LỜI NÓI ĐẦU .....	1
CHƯƠNG 1.TỔNG QUAN VỀ HỆ THỐNG RỬA XE TỰ ĐỘNG	
1.1.Các phương pháp rửa xe	
1.1.1.Rửa xe thủ công	
1.1.2.Rửa xe bán tự động	
1.1.3.Rửa xe tự động	
1.2.Nhà rửa xe trên thực tế Metro Wash 85 của công ty Hanna	
1.2.1.Giới thiệu về nhà rửa xe Metro Wash 85	
1.2.2.Mô tả chi tiết	
CHƯƠNG 2.THIẾT KẾ PHẦN CỨNG HỆ THỐNG	
2.1.Mô hình cơ khí	
2.1.1.Phần khung dưới	
2.1.2.Phần khung trên	
2.2. Các cơ cấu chấp hành	
2.2.1. Động cơ	
2.2.2. Bơm	
2.3.Thiết kế tủ động lực	
2.3.1.Điều kiện để lựa chọn thiết bị	
2.3.2.Aptomát	
2.3.3.Role	
2.3.4.Thiết bị cảm biến	
2.3.5.Công tắc hành trình	
2.3.6.Cảm biến Encoder	
2.3.7.Nút ấn	



## 2.4. Giới thiệu về bộ điều khiển: Vi điều khiển PIC

2.4.1. PIC là gì ?

2.4.2. Ưu điểm của PIC

2.4.3. Các dòng PIC và cách lựa chọn vi điều khiển PIC

2.4.5. Mạch nạp PIC

2.4.4. Ngôn ngữ lập trình cho PIC

2.4.5. Mạch nạp PIC

2.4.6. Vi điều khiển PIC16F877A

## 2.5. Màn hình LCD

2.5.1. Hoạt động của LCD

2.5.2. Mô tả các chân của LCD

## 2.6. Các khối mạch điều khiển

2.6.1. Sơ đồ

2.6.2. Chức năng từng khối

# CHƯƠNG 3. THIẾT KẾ THUẬT TOÁN VÀ CHƯƠNG TRÌNH ĐIỀU KHIỂN CHO NHÀ RỬA XE TỰ ĐỘNG

## 3.1. Nguyên lý hoạt động của hệ thống

## 3.2. Thuật toán PID trong việc điều khiển tốc độ động cơ

3.2.1. Sơ đồ khối bộ điều chỉnh PID động cơ một chiều bằng vi điều khiển

3.2.2. Các luật điều khiển số

3.2.3. Xây dựng bộ PID dùng vi điều khiển PIC16F877A

## 3.3. Giới thiệu về ngôn ngữ lập trình C

3.3.1. Tổng quan về ngôn ngữ lập trình C

3.3.2.Cấu trúc của một chương trình C

3.4.Lưu đồ thuật giải

3.5.Chương trình điều khiển

## LỜI NÓI ĐẦU

Theo đà phát triển nhanh chóng của khoa học kỹ thuật làm cho cuộc sống của con người ngày càng văn minh và tiện lợi hơn. Công nghệ tự động hóa đã được áp dụng tại nhiều quốc gia tiên tiến trên thế giới trong nhiều lĩnh vực, không chỉ trong công nghiệp mà cả trong sinh hoạt hàng ngày. Đối với thực tế cuộc sống hàng ngày thì “Rửa xe tự động” không thể thiếu ở các quốc gia phát triển với mật độ lớn xe ô tô. Hệ thống rửa xe tự động ra đời đáp ứng được tính chuyên nghiệp của dịch vụ rửa xe, đáp ứng nhu cầu ngày càng cao của cuộc sống và tiết kiệm thời gian nhưng cũng không kém phần hiệu quả so với các dịch vụ rửa xe tự động.

Với mặt bằng của Việt Nam hiện nay thì dịch vụ này còn khá mới, nó vẫn chưa phổ biến rộng rãi, nhưng trong một thời gian không xa nữa cùng với đà phát triển của nền kinh tế quốc dân thì ngành giao thông vận tải cũng phát triển theo, làm cho xuất hiện nhiều xe ô tô thay thế các phương tiện hiện có là xe máy và xe thô sơ. Điều này dẫn tới nhu cầu cấp thiết của những hệ thống rửa xe tự động nhằm tiết kiệm thời gian, tiết kiệm nhân công mà chất lượng thì vượt xa các phương pháp rửa xe truyền thống.

Sau thời 5 năm học tập tại trường và sau thời gian thực tập tốt nghiệp em đã được giao đề tài :

### ***“Xây dựng hệ thống rửa xe tự động trên nền vi điều khiển họ PIC”***

Nội dung cơ bản của đề án tốt nghiệp được chia làm 3 chương cơ bản như sau:

Chương 1: **Tổng quan về hệ thống rửa xe tự động.**

Chương 2: **Thiết kế phần cứng của hệ thống.**

Chương 3: **Thiết kế thuật toán và chương trình điều khiển cho hệ thống.**

Trong thời gian làm đề án tốt nghiệp, em đã được sự chỉ bảo tận tình của thầy giáo hướng dẫn Th.S Nguyễn Trọng Thắng và sự giúp đỡ thầy cô trong khoa và các bạn mà đề án của em đã hoàn thành.

Mặc dù có nhiều cố gắng nhưng do hạn chế về kiến thức và kinh nghiệm nên đề án của em còn không ít thiếu sót và nhiều phần còn chưa tìm hiểu được sâu. Em

rất mong được sự chỉ bảo của toàn thể thầy cô và bạn bè để đề án của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn thầy giáo hướng dẫn Th.S Nguyễn Trọng Thắng cùng các thầy cô trong khoa và toàn thể các bạn đã giúp đỡ em hoàn thành đề án này.

Em xin chân thành cảm ơn !

Hải Phòng, ngày tháng năm 2012

Sinh viên thực hiện

Nguyễn Đức Hùng

# CHƯƠNG 1

## TỔNG QUAN VỀ HỆ THỐNG RỬA XE TỰ ĐỘNG

### 1.1.Các phương pháp rửa xe.

#### 1.1.1.Rửa xe thủ công.



**Hình 1.1:** Phương pháp rửa xe thủ công.

Với phương pháp này thì hầu như con người đảm nhiệm 100% công việc từ phun nước, phun chất tẩy rửa, làm sạch bụi bẩn và xì khô...

-Ưu điểm:

- Xe được rửa khá sạch, chỗ nào bẩn nhiều thì được rửa kỹ hơn.
- Vốn đầu tư cho máy móc không lớn.
- Giá thành tương đối thấp.

-Nhược điểm:

- Thời gian rửa xe lâu

- Chi phí nhân công tốn kém

### 1.1.2. Rửa xe bán tự động.



**Hình 1.2:** Phương pháp rửa xe bán tự động

Phương pháp kết hợp giữa máy móc và con người. Con người tham gia vào quá trình rửa xe như phun nước, tẩy rửa. Sau đó xe được đưa qua hệ thống phun nước tẩy rửa, bụi bẩn và qua hệ thống sấy khô tự động.

-Ưu điểm:

- Vốn đầu tư cho thiết bị không quá lớn
- Phù hợp với địa hình nhiều quốc gia
- Giảm bớt sức lao động của con người

-Nhược điểm:

- Thời gian tuy đã tiết kiệm được nhưng chưa tối ưu.
- Chưa kết hợp các quá trình thành một hệ thống hoàn chỉnh.

### 1.1.3. Rửa xe tự động.



**Hình 1.3:** Phương pháp rửa xe tự động.

Toàn bộ quá trình rửa xe được thực hiện bởi các thiết bị tự động dưới sự giám sát và điều khiển của con người.

-Ưu điểm:

- Tiết kiệm thời gian rửa xe
- Sử dụng máy móc tự động làm tiết kiệm nhân công

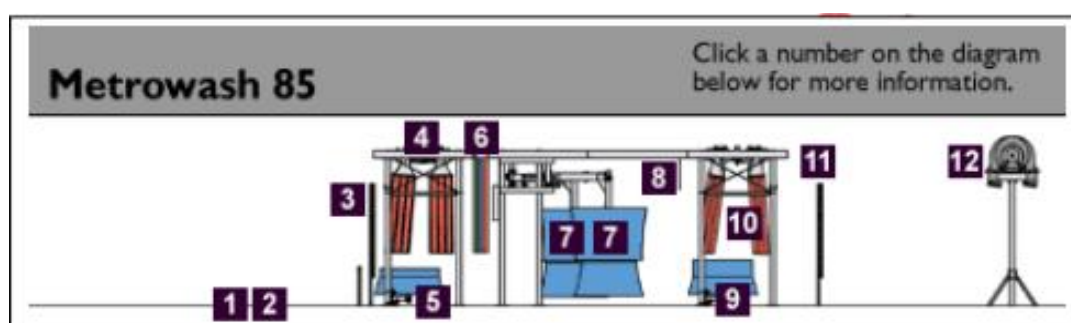
-Nhược điểm:

- Vốn đầu tư ban đầu, chi phí lắp đặt, bảo dưỡng lớn
- Giá thành rửa xe cao

## 1.2. Nhà rửa xe trên thực tế Metro Wash 85 của công ty Hanna.

### 1.2.1. Giới thiệu về nhà rửa xe Metro Wash 85 (hình 1.4)

Kiểu Nhà Rửa Xe Metro Wash 85 có công suất lớn, rửa được 85 chiếc xe trong mỗi giờ. Khả năng tẩy rửa của hệ thống rất cao. Loại bỏ các bụi bẩn bám trên bề mặt trước-sau và các góc cạnh của xe. Hệ thống không chỉ rửa sạch các bụi bẩn mà còn có thêm chức năng đánh bóng xe nhờ hệ thống chổi vải mềm đặc biệt.



*Hình 1.4:* Mô hình tổng thể.

*Chú thích:*

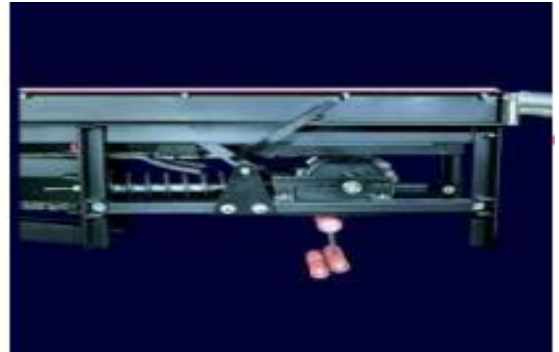
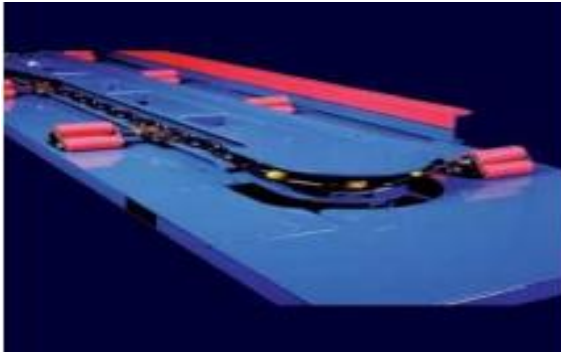
1. Con lăn bánh xe cho xe di chuyển vào hệ thống.
2. Đường ray để đưa bánh xe vào vị trí con lăn.
3. Gồm các vòi phun nước áp suất cao để loại bỏ sơ bụi bám bên ngoài và phun chất tẩy rửa.
- 4-10. Hệ thống cặp chổi làm sạch phần phía trên của mũ xe.
- 5-9. Hệ thống chổi nhỏ làm sạch 4 bánh xe.
7. Hệ thống cặp chổi lớn làm sạch phần trước – sau thân xe.
8. Phun nước làm sạch chất tẩy.
11. Phun nước áp suất cao để loại bỏ hoàn toàn bụi bám.
12. Hệ thống sấy khô xe.

### 1.2.2. Mô tả chi tiết.

#### 1.2.2.1. Hệ thống Con lăn (hình 1.5)



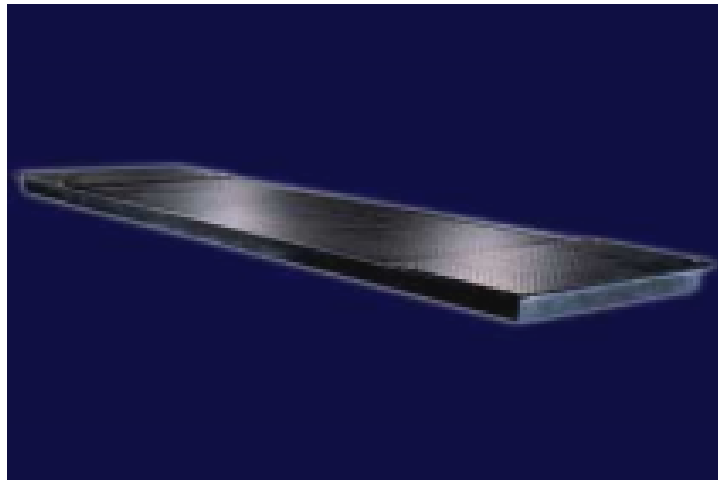
Gồm có sợi dây sên dài gắn nhiều con lăn có trục nghiêng khoảng  $45^{\circ}$ . Khi dây sên di chuyển sẽ đẩy xe về phía trước. Hệ thống con lăn chỉ nằm 1 bên của xe.



**Hình 1.5:** Hệ thống con lăn.

#### 1.2.2.2. Đường ray (hình 1.6)

Khi xe di chuyển vào, hệ thống đường ray sẽ giữ hướng di chuyển của xe trên một đường thẳng. Giúp đưa bánh xe vào hệ thống con lăn.



**Hình 1.6:** Hệ thống đường ray.

#### 1.2.2.3. Vòng phun nước (hình 1.7)

Gồm nhiều vòi phun nước khác nhau. Bố trí đều ở vòng phun nước. Phun ra lượng lớn nước có áp suất cao. Loại bỏ bụi bẩn bám ở mặt ngoài của xe.



**Hình 1.7:** Hệ thống vòng phun nước.

#### 1.2.2.4. Hệ thống cặp chổi (hình 1.8)

Gồm nhiều dây vải ghép nối với nhau vào thanh chuyển động tạo thành cặp chổi. Cặp chổi này được gắn vào trục quay lệch tâm của động cơ thông qua cơ cấu cơ khí. Khi động cơ quay sẽ làm cặp chổi chuyển động tới lui theo chiều chuyển động của xe. Có tác dụng làm sạch ở phần thân trên của xe.



**Hình 1.8:** Hệ thống chổi lớn.

#### *1.2.2.5. Hệ thống chổi nhỏ (hình 1.9)*

Gồm nhiều dây vải nhỏ đan vào trục quay của động cơ tạo thành chùm. Quay ở một vị trí cố định. Có tác dụng làm sạch bụi bám vào phần thân dưới và bánh xe.



**Hình 1.9:** Hệ thống chổi nhỏ.

#### *1.2.2.6. Hệ thống phun chất tẩy rửa (hình 1.10)*

Gồm vòi phun hóa chất tẩy rửa:



**Hình 1.10:** Hệ thống phun chất tẩy rửa.

#### 1.2.2.7. Hệ thống chổi lớn (Hình 1.11)

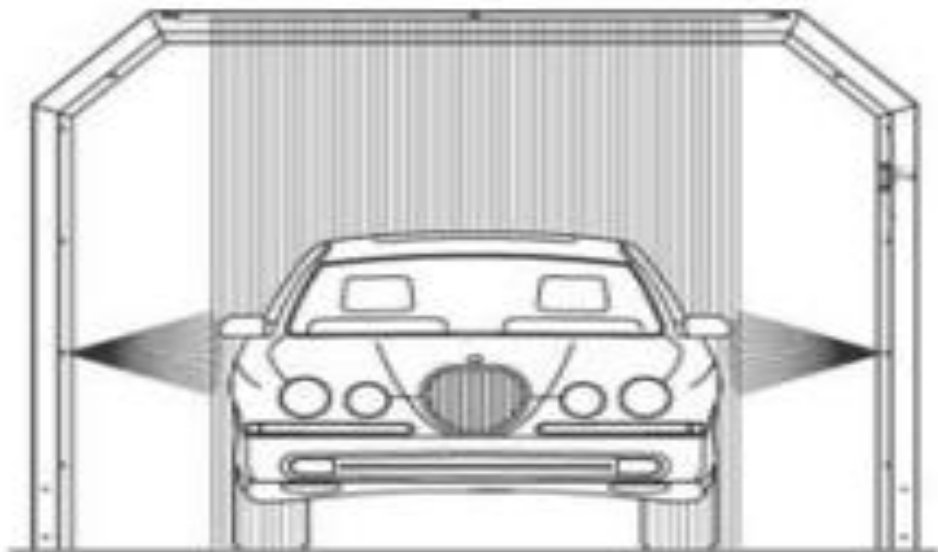
Gồm mảnh vải lớn và dày gắn vào một trục xoay dài. Hệ thống chổi lớn này có thể di chuyển vào ra dễ dàng nhờ cơ cấu bánh nhông. Để lau chùi hết các vị trí bên hông của thân xe.



**Hình 1.11:** Hệ thống chổi thân xe.

#### 1.2.2.8. Phun nước làm sạch (hình 1.12)

Sau khi qua hệ thống chổi, tương đối xe cũng đã được làm tẩy sạch. Hệ thống phun nước này có tác dụng rửa sạch các hóa chất phun lúc ban đầu.



**Hình 1.12:** Hệ thống phun nước làm sạch.

### 1.2.2.9. Sấy khô

Sau khi xe đã chùi rửa xong. Xe sẽ được sấy khô trong khoảng thời gian ngắn.



**Hình 1.13:** Hệ thống sấy

## CHƯƠNG 2

# THIẾT KẾ PHẦN CỨNG HỆ THỐNG

### 2.1. Mô hình cơ khí

#### 2.1.1. Phần khung dưới:

- Hệ thống nhông xích kéo xe được gắn 1 động cơ 24VDC có khả năng kéo tải từ 20-30kg.
- Các động cơ bơm (hình 2.1) và bể chứa nước tẩy, nước sạch cũng như nước thải được bố trí gần nhau và được đặt tránh các động cơ, mạch điện để đảm bảo hạn chế nước tràn lan.

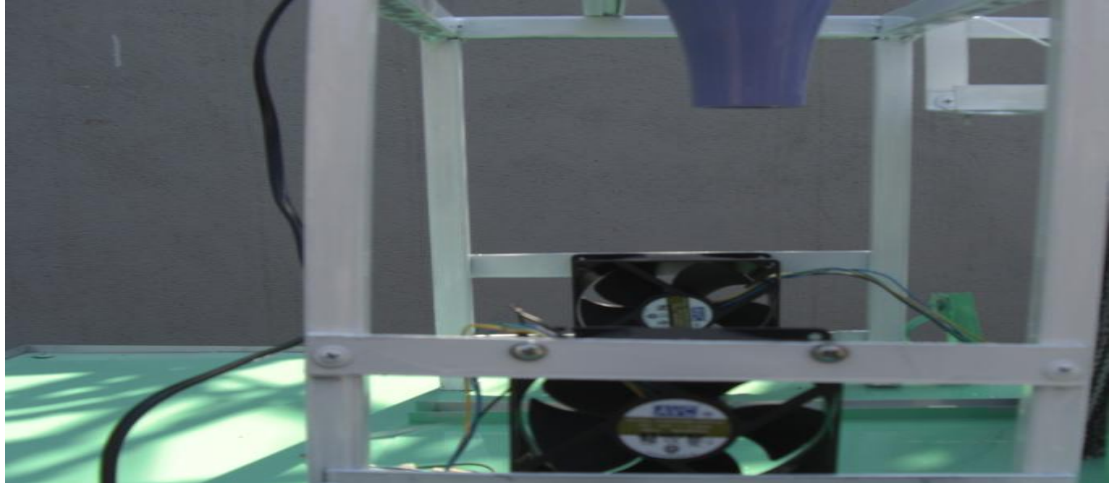


*Hình 2.1:* Các hệ thống bơm.

#### 2.1.2. Phần khung trên:

- Bên trên khung gắn các vòi phun nước gồm các vòi phun nước làm sạch và các vòi phun chất tẩy rửa.
- Hệ thống chổi được làm từ các chất liệu mềm để không làm xước sơn cũng như đảm bảo có độ cọ để có thể làm sạch được các vết bẩn bám trên xe.
- Hệ thống sấy khô gồm hệ thống sấy phía trên và hệ thống sấy 2 bên thân xe như hình vẽ 2.3.
- Các mạch động lực và các mạch điều khiển được lắp trong 1 tủ điện đặt bên cạnh khung mô hình.





*Hình 2.2:* Hệ thống sấy.

## **2.2.Các cơ cấu chấp hành: Động cơ, bơm...**

### **2.2.1.Động cơ**

#### *2.2.1.1.Khái quát về động cơ điện*

Là máy điện chuyển đổi năng lượng điện sang năng lượng cơ. Trong hầu hết các hệ thống máy móc công nghiệp, người ta chủ yếu sử dụng động cơ điện làm thiết bị dẫn động với những ưu điểm sau:

- Ít gây tiếng ồn, ít gây rung động.
- Dễ điều chỉnh tốc độ.
- Dễ sử dụng, dễ vận hành, điều khiển.
- Dễ tự động quá trình sản xuất.
- Hiệu quả cao.

#### *2.2.1.2.Phân loại*

Động cơ điện xoay chiều: Động cơ điện xoay chiều được sản xuất với nhiều kiểu và công suất khác nhau. Theo sơ đồ nối điện có thể phân ra làm 2 loại: động cơ 3 pha và 1 pha, và nếu theo tốc độ có động cơ đồng bộ và động cơ không đồng bộ.

### 2.2.1.3. Động cơ điện một chiều

Động cơ điện một chiều là động cơ điện hoạt động với dòng điện một chiều. Động cơ điện 1 chiều được dùng rất phổ biến trong công nghiệp, giao thông vận tải và nói chung ở những thiết bị cần điều chỉnh tốc độ quay liên tục trong một phạm vi rộng.

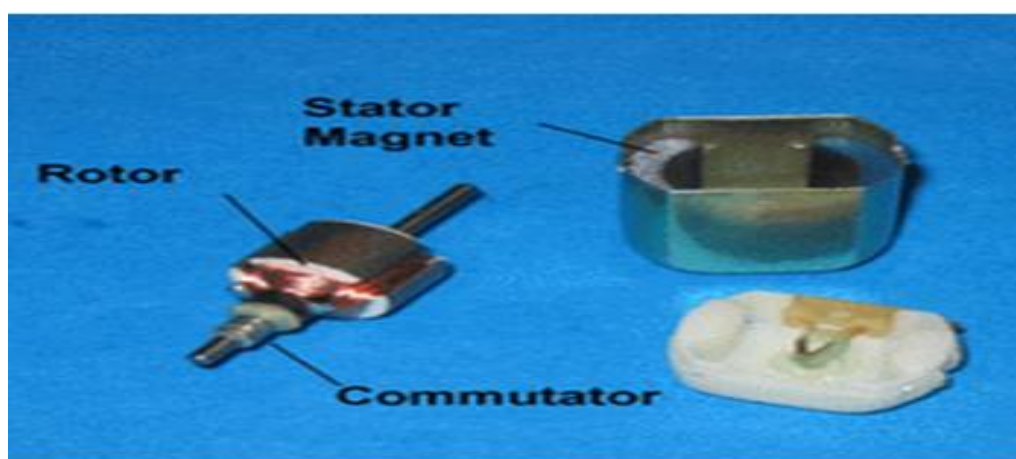
### 2.2.1.4. Cấu tạo

Phần chính gồm Stato (phần đứng yên) với các cực từ (bằng nam châm vĩnh



**Hình 2.3:** Động cơ điện một chiều

cửu hoặc nam châm điện), Roto với các cuộn dây quấn, cổ góp cùng chổi điện. Nắp động cơ được dùng để gá lắp ổ bi đỡ trục động cơ và gắn giá đỡ chổi than.



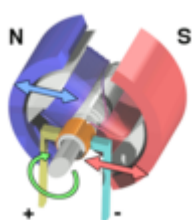
**Hình 2.4:** Cấu tạo động cơ điện 1 chiều



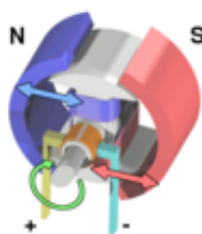
Một phần quan trọng của động cơ điện một chiều là bộ phận chỉnh lưu, nó có nhiệm vụ là đổi chiều dòng điện trong cuộn rotor trong khi chuyển động quay của rotor là liên tục. Thông thường bộ phận này là bộ phận gồm có một bộ cổ góp và một bộ chổi than tiếp xúc với cổ góp. Đây cũng chính là nhược điểm chính của động cơ điện một chiều: cổ góp làm cho cấu tạo phức tạp, đắt tiền, kém tin cậy và nguy hiểm trong môi trường dễ nổ, khi sử dụng phải có nguồn điện một chiều kèm theo hoặc bộ chỉnh lưu.

#### 2.2.1.5. Nguyên lý làm việc

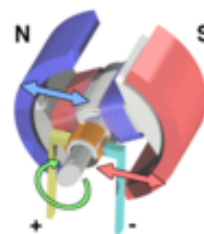
Stator của động cơ điện 1 chiều thường là 1 hay nhiều cặp nam châm vĩnh cửu, hay nam châm điện, rotor có các cuộn dây quấn và được nối với nguồn điện một chiều, 1 phần quan trọng khác của động cơ điện 1 chiều là bộ phận chỉnh lưu, nó có nhiệm vụ là đổi chiều dòng điện trong khi chuyển động quay của rotor là liên tục. Thông thường bộ phận này gồm có một bộ cổ góp và một bộ chổi than tiếp xúc với cổ góp.



**Pha 1:** Từ trường của rotor cùng cực với stator, sẽ đẩy nhau tạo ra chuyển động quay của rotor.



**Pha 2:** Rotor tiếp tục quay.



**Pha 3:** Bộ phận chỉnh điện sẽ đổi cực sao cho từ trường giữa stator và rotor cùng dấu, trở lại pha 1.

### 2.2.1.6. Phương trình đặc tính cơ của động cơ điện một chiều.

Phương trình cân bằng điện áp như sau:

$$U_u = E_u + (R_u + R_{fu}) I_u$$

Trong đó:

- $U_u$  là điện áp nguồn đặt vào phần ứng (V)
- $E_u$  là sức phản điện động của phần ứng động cơ (V), nó tỷ lệ với từ thông  $\Phi$  và tốc độ quay của động cơ  $\omega$  theo biểu thức  $E_u = K\Phi\omega$
- $K$  là hệ số tỷ lệ phụ thuộc vào cấu tạo của động cơ:  $K = pN/2\pi a$
- $p$  là số đôi cực từ chính.
- $N$  là số thanh dẫn tác dụng của cuộn dây phần ứng.
- $a$  là số mạch nhánh đầu song song của cuộn dây phần ứng.
- $R_u = r_u + r_{cf} + r_{cb} + r_{ct}$ : là điện trở mạch phần ứng của động cơ, đơn vị là  $\Omega$ , bao gồm điện trở cuộn dây phần ứng  $r_u$ , điện trở cực từ phụ  $r_{cf}$ , điện trở cuộn bù  $r_{cb}$ , điện trở tiếp xúc của chổi than trên cổ góp  $r_{ct}$ .
- $R_{fu}$  là điện trở phụ trong mạch phần ứng, ( $\Omega$ ).
- $I_u$  là dòng điện mạch phần ứng, (A).

Ta có “ phương trình đặc tính cơ điện ” của động cơ như sau:

$$\omega = \{U_u - (R_u + R_{fu})I_u\}/K\Phi$$

Phương trình trên biểu thị mối quan hệ giữa đại lượng cơ học  $\omega$  và đại lượng  $I_u$  của động cơ.

Mặt khác, mômen điện từ của động cơ tỷ lệ với từ thông  $\Phi$  và dòng điện phần ứng  $I_u$

$$M = K\Phi I_u$$

Từ đó, ta có “phương trình đặc tính cơ” của động cơ như sau:

$$\omega = \frac{U_u}{K\Phi} - \frac{(R_u + R_{fu})}{K^2\Phi^2} M$$

## 2.2.2. Bơm

### 2.2.2.1. Khái niệm

Máy bơm là một loại máy thủy lực, nhận năng lượng từ bên ngoài ( cơ năng, điện năng, thủy năng ..vv.. ) và truyền năng lượng cho dòng chất lỏng, nhờ vậy đưa chất lỏng lên một độ cao nhất định hoặc dịch chuyển chất lỏng theo hệ thống đường ống.

### 2.2.2.2. Phân loại

Người ta chia máy bơm ra nhiều loại dựa vào những đặc điểm như: nguyên lý tác động của cánh bơm vào dòng nước, dạng năng lượng làm chạy máy bơm, kết cấu máy bơm, mục đích bơm, loại chất lỏng cần bơm ... Bơm được chia làm 3 loại bơm sau: Bơm bánh răng, bơm cánh gạt, bơm pittong.

#### a) Bơm bánh răng

Bơm bánh răng là sự thay đổi thể tích của buồng hút (A) tăng, bơm dầu hút, thực hiện chu kỳ hút, và khi thể tích giảm, bơm đẩy dầu ra buồng (B), thực hiện chu kỳ nén. Nếu trên đường đi của dầu ta đặt một vật cản thì dầu sẽ bị chặn lại tạo nên một áp suất nhất định phụ thuộc vào độ lớn của sức cản và kết cấu của bơm.

#### b) Bơm cánh gạt

Bơm cánh gạt được dùng rộng rãi hơn bơm bánh răng do ổn định và lưu lượng, hiệu suất thể tích cao hơn. Lưu lượng bơm có thể thay đổi bằng cách thay đổi độ lệch tâm

### c) Bom pittong

- Bom pittong có khả năng làm kín tốt hơn so với bơm cánh gạt và bánh răng, bởi vậy bơm pittong được sử dụng rộng rãi trong hệ thống thủy lực làm việc ở áp suất cao. Phụ thuộc vào vị trí của pittong đối với roto, có thể phân biệt chúng thành hướng kính và hướng trục.
- Bơm dầu pittong hướng kính có các pittong chuyển động hướng tâm với trục quay của roto. Tùy thuộc vào số pittong ta có lưu lượng khác nhau
- Bơm pittong hướng trục là loại bơm có các pittong đặt song song với trục roto và được truyền bằng khớp nối với trục quay của động cơ điện. Bơm pittong hướng trục có ưu điểm là kích thước nhỏ gọn và hầu hết đều chỉnh được nhờ điều chỉnh góc nghiêng của kết cấu trục đĩa nghiêng ở trong bơm.

#### 2.2.2.3. Hệ thống bơm trong mô hình.

##### a) Hệ thống nước sạch

Hệ thống nước sạch gồm 2 khối:

- 1 khối ở đầu quá trình rửa xe để phun nước cho bớt bụi bẩn của xe khi bắt đầu quá trình rửa. Bơm dùng cho khối này là bơm DC 24V-5W.
- 1 khối ở gần cuối quá trình rửa xe, khi đó chất bẩn đã được chổi lau làm sạch, hệ thống phun nước này giúp cho sạch bụi bẩn và chất tẩy rửa trên xe. Bơm dùng cho khối này là bơm 220V-25W.

##### b) Hệ thống nước tẩy rửa

- Chất tẩy là những nguyên liệu có độ tẩy vừa phải để làm sạch xe và không gây ảnh hưởng tới lớp sơn của xe có thể là dung dịch xà phòng phun bọt tuyết hoặc dung dịch xà phòng thông thường (không có bọt tuyết).

- Hệ thống nước tẩy được đặt sau hệ thống nước đầu mô hình với nhiệm vụ phun chất tẩy rửa trên toàn bề mặt xe để bắt đầu mang xe vào hệ thống chổi để lau.
- Bơm dùng cho mô hình ở hệ thống chất tẩy rửa là bơm 12V.
- Hệ thống vòi phun nước gắn quanh khung để đảm bảo xả nước được hết các phần của xe.

## **2.3.Thiết kế tử động lực**

### **2.3.1.Điều kiện để lựa chọn thiết bị**

Khi thiết kế mạch lực cho một hệ thống điều khiển tự động thì nhiệm vụ của người thiết kế là lựa chọn phương án cung cấp điện tốt nhất cho hệ thống, vừa thỏa mãn các yêu cầu kỹ thuật đã đề ra lại vừa rẻ về vốn đầu tư và chi phí vận hành. Do vậy, cũng có nhiều phương án đưa ra để tính toán so sánh để chọn được phương án tốt nhất.

Điều kiện đầu tiên để các phương án được liệt vào danh sách các phương án đem so sánh là chúng phải đảm bảo những yêu cầu kỹ thuật cơ bản. Nếu một phương án thiết kế đưa ra tuy rẻ tiền nhưng không đảm bảo những yêu cầu kỹ thuật cơ bản về chất lượng, độ tin cậy, vấn đề an toàn khi vận hành hệ thống thì phương án đó không có ý nghĩa và phải được loại trừ từ đầu. Ta chỉ cần nhấn mạnh đến việc thỏa mãn các yêu cầu cơ bản vì trong thực tế ít có phương án nào thỏa mãn được toàn bộ những yêu cầu kỹ thuật đề ra. Mỗi phương án đều có mặt mạnh và mặt yếu của nó, việc quyết định lựa chọn phương án nào phải kết hợp với tính toán so sánh về kinh tế kỹ thuật.

Theo yêu cầu công nghệ của máy, cơ cấu sản xuất, các hệ thống truyền động điện tự động đều được thiết kế, tính toán để làm việc ở những trạng thái xác định. Các trạng thái hay sự cố, hư hỏng thông thường phải được dự đoán khi thiết kế, tính toán chúng để áp dụng những thiết bị và biện pháp bảo vệ cần thiết. Những trạng thái làm việc của hệ thống truyền động điện tự động có thể được đặc trưng

bằng các thông số như: tốc độ làm việc của các động cơ truyền động hay của cơ cấu chấp hành máy sản xuất, dòng điện phần ứng của động cơ hay của phần kích thích của động cơ điện một chiều, mômen phụ tải trên trục động cơ truyền động ... Tùy theo quá trình công nghệ yêu cầu mà các thông số trên có thể lấy các giá trị khác nhau. Việc chuyển từ giá trị này đến giá trị khác được thực hiện tự động nhờ hệ thống điều khiển.

Trong quá trình vận hành, hệ thống điều khiển tự động truyền động điện (ĐKTĐ – TĐĐ) có thể có các tác động ngẫu nhiên hoặc chủ quan của nhân viên vận hành dẫn đến các sự cố hoặc chế độ làm việc xấu cho hệ thống. Nếu các chế độ xấu hoặc sự cố không được loại trừ kịp thời thì sẽ dẫn đến hư hỏng máy móc, thiết bị, rối loạn quá trình sản xuất hoặc thậm chí có thể gây ra những tai họa nguy hiểm cho người. Vì vậy, vấn đề bảo vệ và tín hiệu hóa trong hệ thống là không thể thiếu được. Nó còn có nhiệm vụ loại trừ và đề phòng các sự cố và chế độ xấu có thể xảy ra, đảm bảo vận hành an toàn cho máy móc, thiết bị cũng như con người.

Các thiết bị điện, sứ cách điện và các bộ phận dẫn điện khác của hệ thống điện trong điều kiện vận hành có thể ở một trong ba chế độ cơ bản sau:

- Chế độ làm việc lâu dài (tức làm việc bình thường ).
- Chế độ quá tải (đối với một số thiết bị điện có thể cho phép quá tải đến 1,4 lần định mức).
- Chế độ chịu dòng điện ngắn mạch.

Trong chế độ làm việc lâu dài, các thiết bị điện, sứ cách điện và các bộ phận dẫn điện khác sẽ làm việc tin cậy nếu chúng được chọn theo đúng điện áp và dòng điện định mức. Trong chế độ quá tải, dòng điện qua thiết bị và các bộ phận dẫn điện khác lớn hơn so với dòng điện định mức. Nếu mức quá tải không vượt quá giới hạn cho phép thì các thiết bị điện vẫn làm việc tin cậy.

Trong tình trạng ngắn mạch thì các khí cụ điện, sứ cách điện và các bộ phận dẫn điện khác vẫn đảm bảo làm việc tin cậy nếu quá trình lựa chọn chúng có các thông số

theo đúng điều kiện ổn định động và ổn nhiệt. Tất nhiên khi xảy ra ngắn mạch, để hạn chế tác hại của nó thì cần phải nhanh chóng loại trừ tình trạng ngắn mạch.

Các dạng ngắn mạch thường xảy ra trong hệ thống cung cấp điện là ngắn mạch ba pha, hai pha và một pha nối đất. Trong đó ngắn mạch ba pha là nghiêm trọng nhất. Vì vậy, thường người ta căn cứ vào dòng điện ngắn mạch ba pha để lựa chọn các thiết bị điện. Trong hệ thống này ta dùng Aptomat, cầu chì để bảo vệ ngắn mạch cho động cơ của các máy bơm.

Do đó, khi thiết kế hệ động lực cho hệ thống thì việc quan trọng bậc nhất đối với người thiết kế là phải lựa chọn ra được thiết bị, khí cụ điện phù hợp, đảm bảo các yêu cầu kỹ thuật của hệ thống, có ý nghĩa quan trọng trong việc đảm bảo cho hệ thống vận hành an toàn, tin cậy và kinh tế.

### 2.3.2. Aptomat (hình 2.5)

Áp tô mát là thiết bị điện dùng để tự động cắt mạch điện bảo vệ quá tải, ngắn mạch, sụt áp,... hồ quang được dập trong không khí.



**Hình 2.5:** Các loại aptomat.

#### 2.3.2.1. Tính chất và đặc điểm

Là khí cụ điện dùng để đóng cắt mạch điện, bảo vệ quá tải, ngắn mạch, sụt áp. Các yêu cầu đối với Aptomat:

- Chế độ làm việc ở định mức của Aptômat phải là chế độ làm việc dài hạn nghĩa là trị số dòng điện định mức chạy qua Aptômat lâu bao nhiêu cũng được. Mặt khác, mạch dòng điện của Aptômat phải chịu được dòng điện lớn (khi có ngắn mạch) lúc các tiếp điểm của nó đã đóng hay đang đóng
- Aptômat phải ngắt được trị số dòng điện ngắn mạch cao, có thể lên đến vài chục kilôampe. Sau khi ngắt dòng điện ngắn mạch Aptômat phải đảm bảo vẫn làm việc tốt ở trị số dòng điện định mức.
- Để nâng cao tính ổn định nhiệt và điện động của các thiết bị điện, hạn chế sự phá hỏng của dòng điện ngắn mạch gây ra, Aptômat phải có thời gian cắt bé. Muốn vậy thường phải kết hợp lực thao tác cơ học với thiết bị dập hồ quang bên trong Aptômat.
- Để thực hiện các yêu cầu thao tác bảo vệ có chọn lọc, Aptômat cần phải có khả năng điều chỉnh chỉ số dòng điện tác động và thời gian tác động.

#### 2.3.2.2. Nguyên lý làm việc của Aptômat

Ở trạng thái bình thường, sau khi đóng điện, Aptômat được giữ ở trạng thái đóng tiếp điểm nhờ móc răng 1 khớp với cần răng 5 cùng một cụm tiếp điểm động. Khi mạch điện quá tải hay ngắn mạch nam châm điện sẽ hút phần ứng xuống làm nhả móc ra, cần được tự do, kết quả các tiếp điểm được mở ra dưới tác dụng của lò xo, mạch điện bị ngắt.

Theo kết cấu chia của Aptômat ra làm 3 loại: một cực, hai cực, ba cực.

Theo thời gian thao tác chia Aptômat ra các loại : Aptômat cực đại theo dòng điện, Aptômat cực tiểu theo dòng điện, Aptômat cực tiểu theo điện áp, Aptômat dòng điện ngược v.v...

Trong một vài trường hợp có yêu cầu bảo vệ tổng hợp (cực đại theo dòng điện và cực tiểu theo điện áp) người ta có loại Aptômat vạn năng.



### 2.3.2.3. Lựa chọn aptômat

Việc lựa chọn Aptômat dựa vào:

- Dòng điện tính toán đi trong mạch.
- Dòng điện quá tải.
- Tính thao tác có chọn lọc.

Ngoài ra lựa chọn Aptômat còn phải căn cứ vào đặc tính làm việc của phụ tải là Aptômat không được phép cắt khi có quá tải ngắn hạn thường xảy ra trong điều kiện làm việc bình thường như dòng điện khởi động, dòng điện đỉnh trong phụ tải công nghệ.

Yêu cầu chung là dòng điện định mức của móc bảo vệ  $I_{ap}$  không được nhỏ hơn dòng điện tính toán  $I_{tt}$  của mạch:

$$I_{ap} \geq I_{tt}$$

Tùy theo đặc tính và điều kiện làm việc cụ thể của phụ tải, người ta thường chọn dòng điện định mức của móc bảo vệ bằng 125%, 150% hay lớn hơn nữa so với dòng điện tính toán mạch.

Khi mạch cung cấp điện cho phụ tải động lực thì cần xác định các dữ liệu sau:

Xác định dòng điện lớn nhất (dòng điện khởi động lớn nhất)  $I_{KD}$  theo công thức:

$$I_{KD} = K_{mm} \cdot I_{dm}$$

Trong đó:

$I_{dm}$  là dòng điện định mức của động cơ

$K_{mm}$  bội số dòng điện khởi động và dòng điện định mức

Đối với động cơ không đồng bộ loại roto lồng sóc:  $K_{mm} = 4 \div 8$

Loại roto dây quấn:  $K_{mm} = 2$

Đối với nhiều động cơ điện đặt trên cùng một tuyến, nhưng khởi động riêng rẽ:

$$I_{kd} = \sum I_{dm} + (k-1) \cdot I_{dm \max}$$

Trong đó:

$\sum I_{dm}$  : tổng dòng điện định mức của tất cả các động cơ.

$I_{dm \max}$  : dòng điện định mức của động cơ có công suất lớn nhất đồng thời có hệ số K lớn nhất

### 2.3.3. Role (hình 2.6)

#### 2.3.3.1. Khái niệm

Role là loại khí cụ điện hạ áp tự động mà tín hiệu đầu ra thay đổi nhảy cấp khi tín hiệu đầu vào đạt những giá trị xác định. Role được sử dụng rất rộng rãi trong mọi lĩnh vực của khoa học công nghệ và đời sống hàng ngày. Role có nhiều chủng loại với nguyên lý làm việc, chức năng khác nhau như role điện từ, role phân cực, role cảm ứng, role nhiệt, role điện tử tương tự, role điện tử số, role dòng điện, role điện áp, role tổng trở, role áp lực....

#### 2.3.3.2. Cấu tạo và nguyên lý làm việc



**Hình 2.6:** Hình ảnh Role.

Đặc tính cơ bản của role (hình 2.8) là đặc tính “vào-ra”. Khi đại lượng đầu vào X tăng đến giá trị tác động  $X_2$ , đại lượng đầu ra Y thay đổi nhảy cấp từ 0 ( $Y_{\min}$ ) đến 1 ( $Y_{\max}$ ).

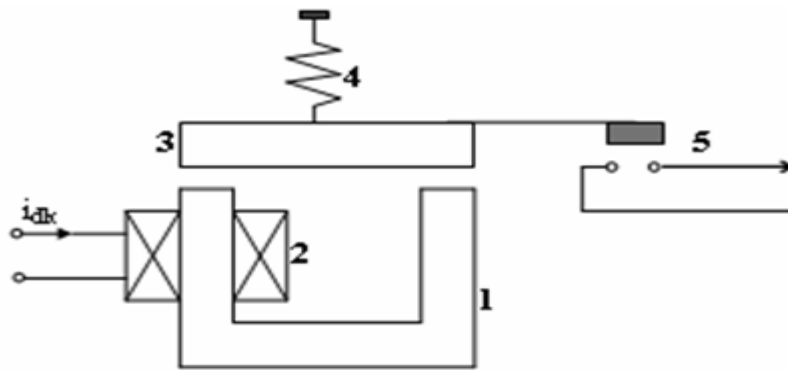
Theo chiều giảm của  $X$ , đến trị số nhỏ  $X_1$  thì đại lượng đầu ra sẽ nhảy cấp từ 1 xuống 0. Đây là quá trình nhả của role.

Hệ số nhả của role :

$$K_{nh} = X_1/X_2$$

Trong đó:

$X_1$  : trị số nhỏ của đại lượng đầu vào



**Hình 2.7:** Đặc tính cơ bản của Role.

$X_2$  : trị số tác động của đại lượng đầu vào.

Từ đặc tính “vào-ra” của role thấy  $K_{nh} < 1$ . Hệ số nhả lớn thường dùng cho role bảo vệ, còn hệ số nhả bé thường dùng cho role điều khiển.

Hệ số dự trữ : 
$$K_{dt} = \frac{X_{lv}}{X_2}$$

Trong đó:

$X_{lv}$  trị số làm việc dài hạn của đại lượng đầu vào.

Nếu  $K_{dt}$  càng lớn, thiết bị làm việc càng an toàn.

Hệ số điều khiển (hệ số khuếch đại) của role :

$$K_{dk} = \frac{P_{ra}}{P_{vào}}$$

Trong đó:

- $P_{ra\_}$  công suất lớn nhất phía đầu ra của role.
- $P_{vào\_}$  công suất tác động của đầu vào.
- $P_{vào}$  khoảng từ cỡ mW đến vài W, còn  $P_{ra}$  cỡ từ vài chục W đến hàng ngàn W, do đó mà  $K_{dk}$  của role có trị số khá lớn, đạt  $10^6$ .

Thời gian tác động của role: là khoảng thời gian từ khi có  $X_{tt}$  đến khi đạt được  $Y_{max}$ , hoặc từ khi  $X=X_{nh}$  đến khi đầu ra đạt  $Y_{min}$ . Đây là một tham số quan trọng của role. Tùy theo chức năng của role mà có thời gian tác động nhanh ( $t < 10^{-3}s$ ), tác động bình thường (cỡ  $10^{-2}s$ ), tác động chậm ( $10^{-1} \div 1s$ ) và role thời gian ( $t > 1s$ ).

Role trung gian được sử dụng rất rộng rãi trong các sơ đồ bảo vệ hệ thống điện và các sơ đồ điều khiển tự động. Đặc điểm của role trung gian là số lượng tiếp điểm lớn (cả tiếp điểm thường đóng và tiếp điểm thường mở) với khả năng chuyển mạch lớn và công suất nuôi cuộn dây bé nên nó được dùng để truyền và khuếch đại tín hiệu, hoặc chia tín hiệu của role chính đến nhiều bộ phận khác nhau của mạch điều khiển và bảo vệ.

Nguyên lý và cấu tạo của một role trung gian như sau: Nếu cuộn dây của role được cấp điện áp định mức (qua tiếp điểm của role chính), sức từ động do dòng điện trong cuộn dây sinh ra ( $i_w$ ) sẽ tạo ra trong mạch từ từ thông, hút nắp làm các tiếp điểm thường mở đóng lại và các tiếp điểm thường đóng mở ra. Khi cắt điện của cuộn dây, lò xo nhả sẽ đưa nắp và các tiếp điểm về vị trí ban đầu. Do dòng điện qua tiếp điểm có giá trị nhỏ (5A) nên hồ quang khi chuyển mạch không đáng kể nên không cần bù dòng hồ quang.

Role trung gian có kích thước nhỏ, gọn, số lượng tiếp điểm đến 4 cặp thường đóng và thường mở liên động, công suất tiếp điểm cỡ 5A, 250VAC, 28VDC, công suất tiêu thụ cuộn dây cỡ ( $1 \div 2W$ ), có LED đỏ báo trạng thái làm việc của role. Nó được chế tạo cho nguồn một chiều 12, 24 VDC và 110, 220VAC,

hệ số nhả của role nhỏ hơn 0,4; thời gian tác động dưới 0,05s; tuổi thọ tiếp điểm đạt  $10^6 \div 10^7$  lần đóng cắt, cho phép tần số thao tác tới 1200 lần/h.

### 2.3.3.3. Các thông số kỹ thuật và cách lựa chọn Role

Dòng điện định mức trên role trung gian là dòng điện lớn nhất cho phép role làm việc trong thời gian dài mà không bị hư hỏng. Khi chọn role trung gian thì dòng điện định mức của nó không được nhỏ hơn dòng điện tính toán của phụ tải. Dòng điện này chủ yếu do tiếp điểm của role trung gian quyết định:

$$I_{dm} = (1,2 \div 1,5) I_{tt}$$

Điện áp làm việc của role trung gian là mức điện áp mà role có khả năng đóng cắt. Điện áp này không được chọn nhỏ hơn điện áp cực đại của lưới điện :

$$U_{lv} > U_l = 380V$$

Dòng điện làm việc của role trung gian phải lớn hơn dòng điện định mức của động cơ :  $I_{lv} > 15,6 A$

Điện áp định mức cấp cho cuộn hút của role là mức điện áp mà khi đó role sẽ hoạt động. Điện áp này phải phù hợp với điện áp của mạch điều khiển.

Ký hiệu của role trung gian trên sơ đồ điện:



Cuộn hút

Tiếp điểm thường mở

Tiếp điểm thường đóng

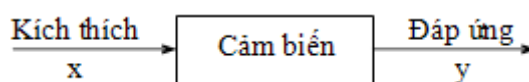
## 2.3.4. Thiết bị cảm biến

### 2.3.4.1. Định nghĩa

Cảm biến trong tiếng Anh gọi là “sensor”, xuất phát từ chữ “sense” theo nghĩa Latinh là cảm nhận. Cảm biến được định nghĩa theo nghĩa rộng là thiết bị cảm nhận và đáp ứng với các tín hiệu và kích thích.

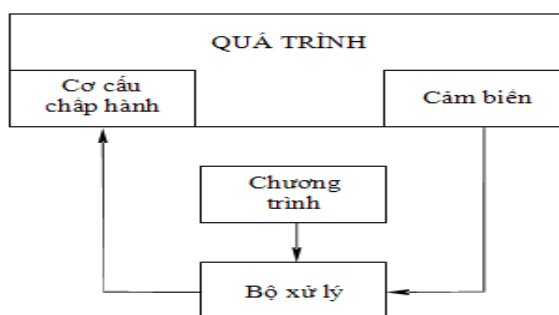
Trong hệ thống đo lường – điều khiển, mọi quá trình đều được đặc trưng bởi các biến trạng thái: nhiệt độ, áp suất, tốc độ, moment... Các biến trạng thái này thường là các đại lượng không điện. Tuy nhiên, trong các quá trình đo lường – điều khiển, thông tin được truyền tải và xử lý dưới dạng điện. Do đó, cảm biến được định nghĩa như những thiết bị dùng để biến đổi các đại lượng vật lý và các đại lượng không điện cần đo thành các đại lượng điện có thể đo được (như dòng điện, điện thế, điện dung, trở kháng v.v...).

Trong mô hình mạch điện, ta có thể coi cảm biến như một mạch hai cửa. Trong đó cửa vào là biến trạng thái cần đo  $x$  và cửa ra là đáp ứng  $y$  của bộ cảm biến với kích thích đầu vào  $x$ .



Phương trình quan hệ:  $y = f(x)$  thường rất phức tạp.

Sơ đồ điều khiển tự động quá trình:



- Bộ cảm biến đóng vai trò cảm nhận, đo đạc và đánh giá các thông số hệ thống.
- Bộ xử lý làm nhiệm vụ xử lý thông tin và đưa ra tín hiệu điều khiển quá trình.

#### 2.3.4.2. Phân loại

a) *Phân loại theo nguyên lý chuyển đổi giữa đáp ứng và kích thích*

- Vật lý: nhiệt điện, quang điện, điện từ, từ điện,...
- Hóa học: hóa điện, phở,...
- Sinh học: sinh điện, ...

b) *Phân loại theo dạng kích thích*: âm thanh, điện, từ, quang, cơ, nhiệt,...

c) *Phân loại theo tính năng*: độ nhạy, độ chính xác, độ phân giải, độ tuyến tính...

d) *Phân loại theo phạm vi sử dụng*: công nghiệp, nghiên cứu khoa học, môi trường, thông tin, nông nghiệp...

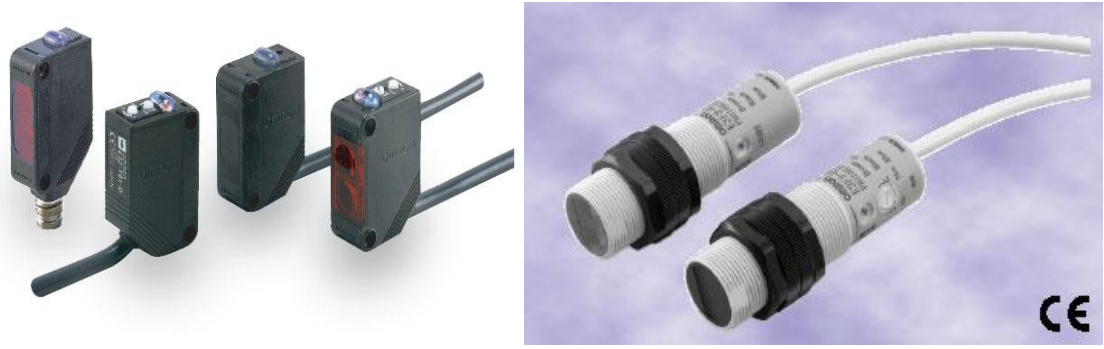
e) *Phân loại theo thông số của mô hình thay thế*:

- Cảm biến tích cực (có nguồn) ngõ ra là nguồn áp hoặc nguồn dòng.
- Cảm biến thụ động (không có nguồn): R, L, C, tuyến tính, phi tuyến.

#### 2.3.4.3. Cảm biến quang (hình 2.8)

a) *Định nghĩa*

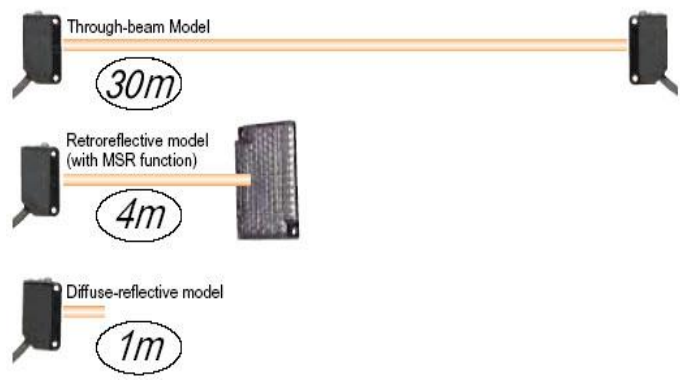
Cảm biến quang dẫn được sử dụng để chuyển thông tin từ ánh sáng nhìn thấy hoặc tia hồng ngoại (IR) và tia tử ngoại (UV) thành tín hiệu điện. Cảm biến này được sử dụng rộng rãi trong các hệ thống tự động vì tuổi thọ cao và có thể đặt vào các không gian nhỏ hẹp.



**Hình 2.8:** Cảm biến quang.

*b) Phân loại*

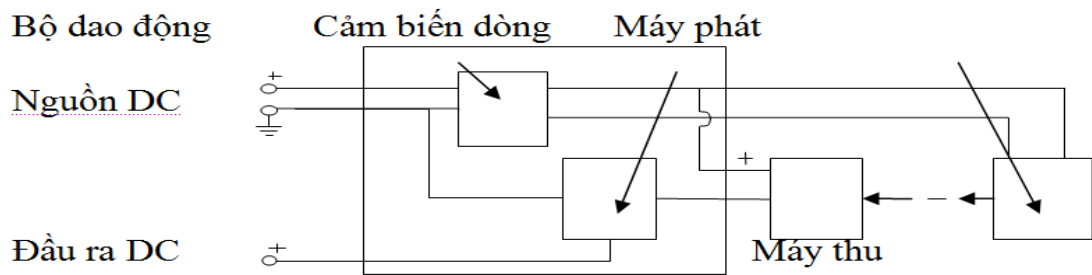
Các cảm biến quang có 2 dạng: cảm biến ánh sáng dạng ống có cực thu phát rời nhau hoặc ghép cùng trên một cụm. Nguồn phát sáng cho các cảm biến quang có tần số sao cho cảm biến có thể phát hiện được tốt nhất, thông thường là tia hồng ngoại. Các cảm biến quang có thể dùng cho khoảng cách vài cm đến vài m, một số loại có thể đạt tới vài chục m.



Loại cảm biến ống thường là một cặp thu và phát tia hồng ngoại. Loại này dùng để phát hiện các vật thể lạ thâm nhập vào hệ thống, có thể được nối với bộ nhập.

Loại đầu thu phát phản hồi thì kết cấu gọn hơn. Nguồn sáng sau khi đập vào vật cản sẽ phản hồi lại vào đầu thu. Khoảng cách đo của loại này nhỏ và cũng dùng để nhận diện chi tiết và dùng trong các hệ thống bảo vệ.





### 2.3.5. Công tắc hành trình (hình 2.9)

Công tắc hành trình trước tiên là cái công tắc tức là làm chức năng đóng mở mạch điện, và nó được đặt trên đường hoạt động của một cơ cấu nào đó sao cho khi cơ cấu đến 1 vị trí nào đó sẽ tác động lên công tắc. Hành trình có thể là tịnh tiến hoặc quay.

Khi công tắc hành trình được tác động thì nó sẽ làm đóng hoặc ngắt một mạch điện do đó có thể ngắt hoặc khởi động cho một thiết bị khác. Người ta có thể dùng công tắc hành trình vào các mục đích như:

- Giới hạn hành trình ( khi cơ cấu đến vị trí dới hạn tác động vào công tắc sẽ làm ngắt nguồn cung cấp cho cơ cấu → nó không thể vượt qua vị trí giới hạn)
- Hành trình tự động: Kết hợp với các role, PLC hay VDK để khi cơ cấu đến vị trí định trước sẽ tác động cho các cơ cấu khác hoạt động (hoặc chính cơ cấu đó).



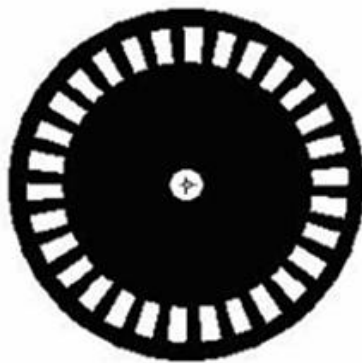
**Hình 2.9:** Công tắc hành trình.

Công tắc hành trình được dùng nhiều trong các dây chuyền tự động... Các công tắc hành trình có thể là các nút nhấn (button) thường đóng, thường mở, công tắc 2 tiếp điểm và cả công tắc quang.

### 2.3.6. Cảm biến Encoder

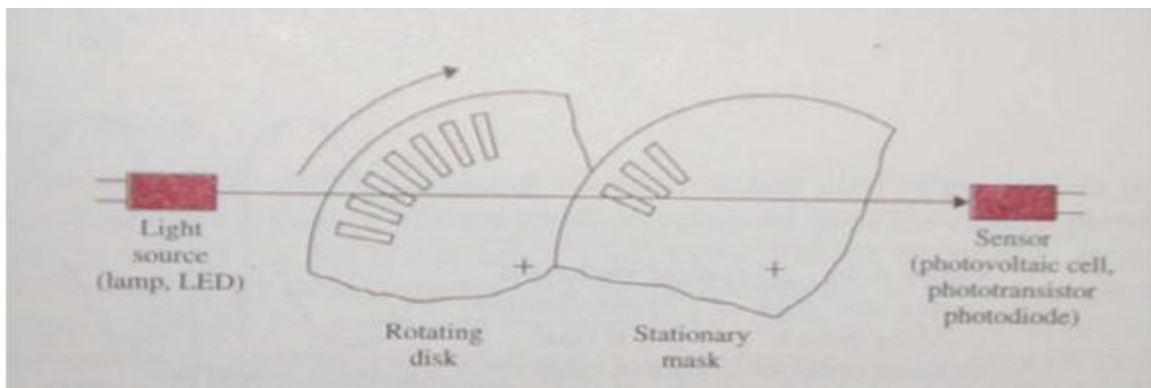
Nguyên lý hoạt động của cảm biến encoder : có rất nhiều loại encoder khác nhau như: Encoder tiếp xúc, Encoder từ trường, Encoder quang (Encoder quang tương đối và Encoder tuyệt đối). Mỗi loại lại có một nguyên lý hoạt động khác nhau, trong phạm vi báo cáo em chỉ xin trình bày phần nguyên lý loại encoder sử dụng là Encoder quang tương đối (incremental encoder).

Encoder thứ 1



**Hình 2.10:** Encoder thứ 1.

Incremental encoder về cơ bản là 1 đĩa tròn quay một trục được đục lỗ như hình vẽ trên.

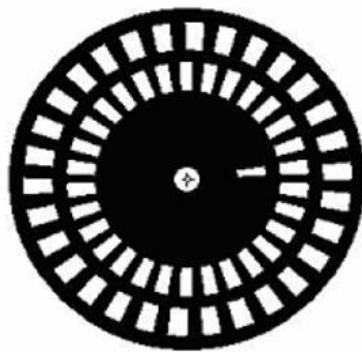


**Hình 2.11:** Phương thức hoạt động Encoder quang tương đối.

Ở 2 bên mặt của cái đĩa tròn, sẽ có 1 bộ thu phát quang. Trong quá trình encoder quay quanh trục, nếu gặp lỗ rộng thì ánh sáng chiếu qua được, nếu gặp vành chắn thì tia sáng không chiếu qua được. Do đó tín hiệu nhận từ sensor quang là 1 chuỗi xung. Mỗi encoder được chế tạo sẽ biết sẵn 1 số xung trên 1 vòng. Do đó ta có thể dùng vi xử lý điều khiển đếm số xung đó trong 1 đơn vị thời gian và tính ra tốc độ động cơ.

Encoder mà em sử dụng trong mô hình hoàn toàn giống với encoder trên. Tuy nhiên có nhược điểm lớn là không thể xác định được động cơ quay trái hay quay phải, dạng xung đưa ra là giống nhau. Ngoài ra điểm bắt đầu của động cơ cũng không thể xác định được.

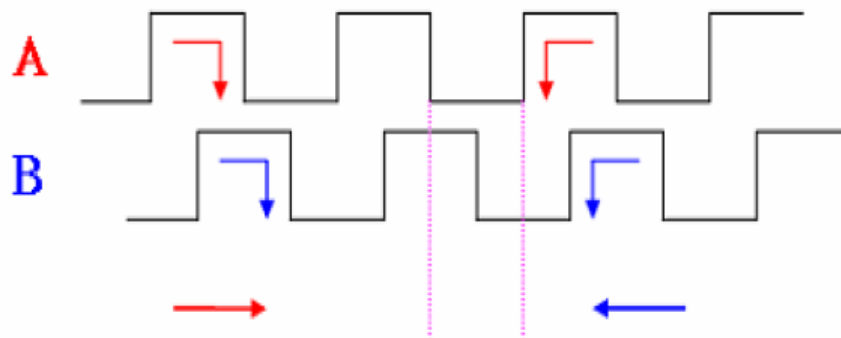
Cải tiến encoder 1 bằng encoder 2 như sau :



**Hình 2.12:** Encoder 2

Người ta đục tất cả là 2 vòng lỗ. Vòng ngoài cũng giống như encoder 1, vòng giữa pha so với vòng ngoài là 90 độ. Khi dạng xung ra từ 2 vòng trên như sau :

Hai xung đưa ra từ 2 vòng lệch nhau 90 độ, nếu vòng ngoài nhanh hơn vòng trong thì chắc chắn động cơ quay từ trái sang phải và ngược lại. Một lỗ ở vòng trong cùng dùng để phát hiện điểm bắt đầu của động cơ. Có thể viết chương trình cho vi điều khiển nhận biết : nếu có một xung phát ra từ vòng trong cùng này, tức là động cơ đã quay đúng một vòng. Với những đặc tính trên, encoder dùng phổ biến trong việc xác định vị trí góc của động cơ....



**Hình 2.13:** Sơ đồ xung của Encoder tương đối

Những loại động cơ có gắn encoder thường bị chết cặp mắt quang 2 bên encoder để tạo xung và không có khả năng thay thế.

### 2.3.7.Nút ấn (hình 2.14)

#### 2.3.7.1. Khái niệm

Nút ấn còn gọi là nút điều khiển là một loại khí cụ điện dùng để đóng ngắt từ xa các thiết bị điện từ khác nhau; các dụng cụ báo hiệu và cũng để chuyển đổi các mạch điện điều khiển, tín hiệu liên động bảo vệ ...

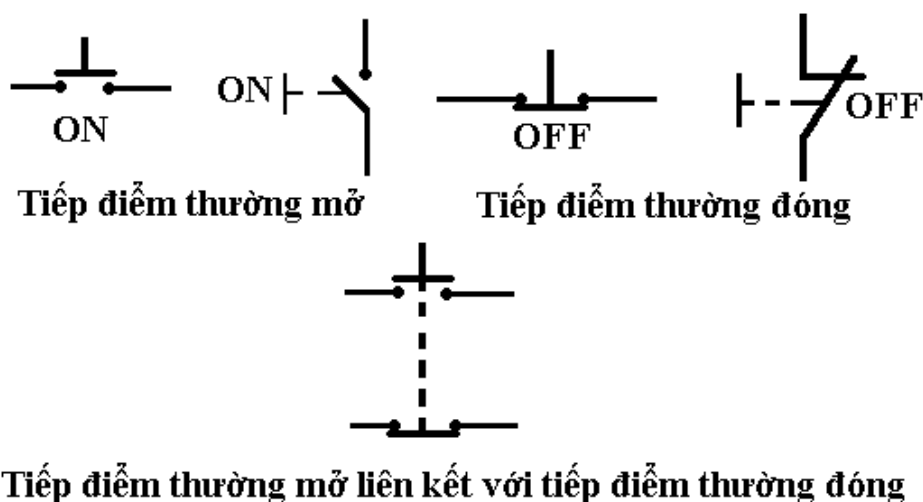


**Hình 2.14:** Hình ảnh nút ấn.

#### 2.3.7.2. Cấu tạo và nguyên lý làm việc

Nút ấn gồm hệ thống lò xo, hệ thống các tiếp điểm thường mở và thường đóng và vỏ bảo vệ. Khi tác động vào nút ấn, các tiếp điểm chuyển trạng thái và khi không còn tác động, các tiếp điểm trở về trạng thái ban đầu. Nút ấn thường được

đặt trên bảng điều khiển, ở tủ điện, trên hộp nút nhấn. Nút nhấn thường được nghiên cứu, chế tạo làm việc trong môi trường không ẩm ướt, không có hơi hóa chất và bụi bẩn. Các loại nút ấn thông dụng có dòng điện định mức là 5A, điện áp định mức là 400V, tuổi thọ điện đến 200.000 lần đóng cắt, tuổi thọ cơ đến 1.000.000 lần đóng cắt. Ở mạch điện một chiều điện áp đến 440V và mạch điện xoay chiều điện áp 500V, tần số 50HZ; 60HZ, nút ấn thông dụng để khởi động, đảo chiều quay động cơ điện bằng cách đóng và ngắt các cuộn dây của contactor nối cho động cơ. Nút ấn màu đỏ thường dùng để dừng máy, còn màu xanh dùng cho khởi động máy.



### 2.3.7.3. Phân loại

Nút ấn được phân loại theo các yếu tố sau:

+ Phân loại theo chức năng trạng thái hoạt động của nút ấn, có các loại:

- Nút ấn đơn: Mỗi nút ấn chỉ có một trạng thái (ON hoặc OFF).
- Trong thực tế, để dễ dàng sử dụng vào tháo ráp lắp lẫn trong quá trình sửa chữa, thường người ta dùng nút ấn kép, ta có thể dùng nó như là dạng nút nhấn ON hay OFF.

+ Phân loại theo hình dạng bên ngoài, người ta chia nút ấn ra thành 4 loại:

- Loại hở.
  - Loại bảo vệ.
  - Loại bảo vệ chống nước và chống bụi. Nút ấn kiểu bảo vệ chống nước được đặt trong một hộp kín khí để tránh nước lọt vào. Nút ấn kiểu bảo vệ chống bụi nước được đặt trong một vỏ cacbon đút kín khí để chống âm và bụi lọt vào.
  - Loại bảo vệ khỏi nổ. Nút ấn kiểu chống nổ dùng trong các hầm lò, mỏ than hoặc ở nơi có các khí nổ lẫn trong không khí. Cấu tạo của nó đặc biệt kín khí không lọt được tia lửa ra ngoài và đặc biệt vững chắc để không bị phá vỡ khi nổ.
- + Theo yêu cầu điều khiển người ta chia nút ấn ra 3 loại: một nút, hai nút, ba nút.
- + Theo kết cấu bên trong:
- Nút ấn loại có đèn báo.
  - Nút ấn loại không có đèn báo.
- \* Các thông số kỹ thuật của nút ấn:
- $U_{dm}$  : điện áp định mức của nút ấn.
  - $I_{dm}$  : dòng điện định mức của nút ấn.

## **2.4. Giới thiệu về bộ điều khiển: Vi điều khiển PIC**

### **2.4.1. PIC là gì?**

PIC là viết tắt của “Programmable Intelligent Computer”, có thể tạm dịch là “máy tính thông minh khả trình” do hãng General Instrument đặt tên cho vi điều khiển đầu tiên của họ: PIC1650 được thiết kế để dùng làm các thiết bị ngoại vi cho vi điều khiển CP1600. Vi điều khiển này sau đó được nghiên cứu phát triển thêm và từ đó hình thành nên dòng vi điều khiển PIC ngày nay.

### **2.4.2.Ưu điểm của PIC**

Hiện nay trên thị trường có rất nhiều họ vi điều khiển như 8051, Motorola 68HC, AVR, ARM,... Ngoài họ 8051 được hướng dẫn một cách căn bản ở môi trường đại học, bản thân chúng em đã chọn họ vi điều khiển PIC để mở rộng vốn kiến thức và phát triển các ứng dụng trên công cụ này vì các nguyên nhân sau:

- Họ vi điều khiển này có thể mua dễ dàng tại thị trường Việt Nam.
- Giá thành không quá đắt (khoảng 60-70 nghìn đồng/ PIC 16F877A)
- Có đầy đủ các tính năng của một vi điều khiển khi hoạt động độc lập.
- Là một sự bổ sung rất tốt về kiến thức cũng như về ứng dụng cho họ vi điều khiển mang tính truyền thống: họ vi điều khiển 8051.
- Số lượng người sử dụng họ vi điều khiển PIC: Hiện nay tại Việt Nam cũng như trên thế giới, họ vi điều khiển này được sử dụng khá rộng rãi. Điều này tạo nhiều thuận lợi trong quá trình tìm hiểu và phát triển các ứng dụng như: số lượng tài liệu, số lượng các ứng dụng mở đã được phát triển thành công, dễ dàng trao đổi, học tập, dễ dàng tìm được sự chỉ dẫn khi gặp khó khăn,...
- Sự hỗ trợ của nhà sản xuất về trình biên dịch, các công cụ lập trình, nạp chương trình từ đơn giản đến phức tạp,...
- Các tính năng đa dạng của vi điều khiển PIC, và các tính năng này không ngừng được phát triển.

### **2.4.3.Các dòng PIC và cách lựa chọn vi điều khiển PIC**

Các kí hiệu của vi điều khiển PIC:

PIC12xxxx: độ dài lệnh 12 bit

PIC16xxxx: độ dài lệnh 14 bit

PIC18xxxx: độ dài lệnh 16 bit

C: PIC có bộ nhớ EPROM (chỉ có 16C84 là EEPROM)

F: PIC có bộ flash

LF: PIC có bộ nhớ flash hoạt động ở điện áp thấp

LV: tương tự như LF, đây là kí hiệu cũ

Bên cạnh đó một số vi điều khiển có kí hiệu xxFxxx là EEPROM, nếu có thêm chữ A ở cuối là flash (ví dụ PIC16F877 là EEPROM, còn PIC16F877A là flash). Ngoài ra còn có thêm một dòng vi điều khiển PIC mới là dsPIC.

Ở Việt Nam phổ biến nhất là các họ vi điều khiển PIC do hãng Microchip sản xuất.

Cách lựa chọn một vi điều khiển PIC phù hợp:

- Trước hết cần chú ý đến số chân của vi điều khiển cần thiết cho ứng dụng. Có nhiều vi điều khiển PIC với số lượng chân khác nhau, thậm chí có vi điều khiển chỉ có 8 chân, ngoài ra còn có các vi điều khiển 28,40,44,... chân.
- Cần chọn cho vi điều khiển PIC có bộ nhớ flash để có thể nạp xóa chương trình được nhiều lần hơn.
- Tiếp theo cần chú ý đến các khối chức năng được tích hợp sẵn trong vi điều khiển, các chuẩn giao tiếp bên trong.
- Sau cùng cần chú ý đến bộ nhớ chương trình mà vi điều khiển cho phép.

#### **2.4.4. Ngôn ngữ lập trình cho PIC**

Ngôn ngữ lập trình cho PIC rất đa dạng. Ngôn ngữ lập trình cấp thấp có MPLAB (được cung cấp miễn phí bởi nhà sản xuất Microchip), các ngôn ngữ lập trình cấp cao hơn bao gồm C, Basic, Pascal, ... Ngoài ra còn có một số ngôn ngữ lập trình được phát triển dành riêng cho PIC như PICBasic, MikroBasic, ...



### 2.4.5. Mạch nạp PIC

Có thể sử dụng mạch nạp được cung cấp bởi nhà sản xuất là hãng Microchip như: PICSTART plus, MPLAB ICD 2, MPLAB 3, PRO MATE II. Có thể dùng sản phẩm này để nạp cho vi điều khiển khác thông qua chương trình MPLAB. Dòng sản phẩm chính thống này có ưu thế là nạp được tất cả các vi điều khiển PIC, tuy nhiên giá thành lại rất cao và thường gặp rất nhiều khó khăn trong quá trình mua sản phẩm.

Ngoài ra do tính năng cho phép nhiều chế độ nạp khác nhau, còn có rất nhiều mạch nạp được thiết kế dành cho vi điều khiển PIC. Có thể sơ lược một số mạch nạp cho PIC như sau:

JDM programmer: mạch nạp này dùng chương trình nạp Icprog, cho phép nạp các vi điều khiển PIC có hỗ trợ tính năng nạp chương trình điện áp thấp ICSP (In Circuit Serial Programming). Hầu hết các mạch nạp đều hỗ trợ tính năng nạp chương trình này.

WARP-13A và MCP-USB: hai mạch này giống với mạch PICSTART plus do nhà sản xuất Microchip cung cấp, tương thích với trình biên dịch MPLAB, nghĩa là ta có thể trực tiếp dùng chương trình MPLLAB để nạp cho vi điều khiển PIC mà không cần sử dụng một chương trình nạp khác, chẳng hạn như ICprog.

P16PRO40: mạch nạp này do Nigel thiết kế và cũng khá nổi tiếng. Ông còn thiết kế cả chương trình nạp, tuy nhiên ta cũng có thể sử dụng chương trình nạp Icprog.

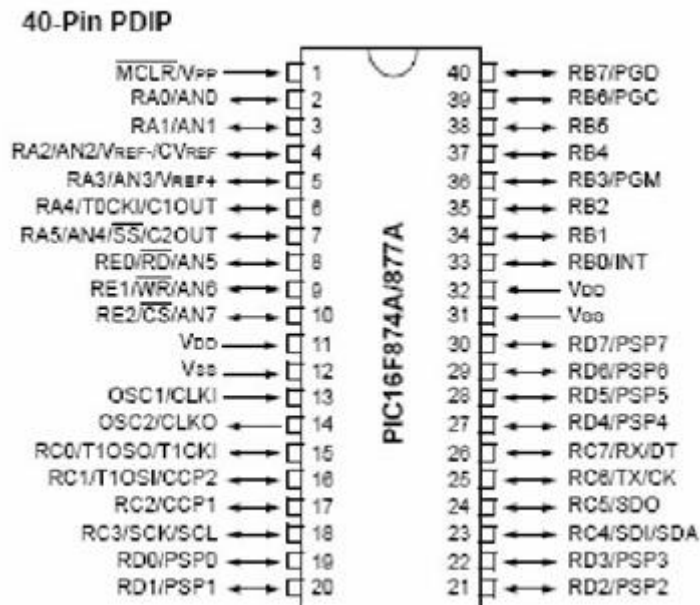
Mạch nạp Universal của Williem: đây không phải là mạch nạp chuyên dụng cho PIC như P16PRO40.

Các mạch nạp kể trên có ưu điểm rất lớn là đơn giản, rẻ tiền, hoàn toàn có thể tự lắp ráp một cách dễ dàng, và mọi thông tin về sơ đồ mạch nạp, cách thiết kế, thi công, kiểm tra và chương trình nạp đều dễ dàng tìm được và download miễn

phí thông qua mạng internet. Tuy nhiên các mạch nạp trên có nhược điểm là hạn chế về số vi điều khiển được hỗ trợ, bên cạnh đó mỗi mạch nạp cần được sử dụng với một chương trình nạp thích hợp.

## 2.4.6. Vi điều khiển PIC16F877A

### 2.4.6.1. Sơ đồ chân vi điều khiển PIC16F877A ( hình 2.15)



**Hình 2.15:** Sơ đồ chân của PIC16F877A.

Ngoài dạng sơ đồ chân như hình 2.12 PIC16F877A còn có dạng 2 dạng sơ đồ 44 chân khác.

### 2.4.6.2. Một vài thông số về vi điều khiển PIC16F877A

Đây là vi điều khiển thuộc họ PIC16Fxxx với tập lệnh gồm 35 lệnh có độ dài 14 bit.

Mỗi lệnh đều được thực thi trong một chu kỳ xung clock. Tốc độ hoạt động tối đa cho phép là 20 MHz với một chu kỳ lệnh là 200ms. Bộ nhớ chương trình 8K x 14bit, bộ nhớ dữ liệu là 368 x 8byte RAM và bộ nhớ dữ liệu EEPROM với dung lượng 256 x 8byte. Số PORT I/O là 5 với 33 pin I/O

Các đặc tính ngoại vi bao gồm các khối chức năng sau:

- Timer0: bộ đếm 8 bit với bộ chia tần số 8 bit.
- Timer1: bộ đếm 16 bit với bộ chia tần số, có thể thực hiện chức năng đếm dựa vào xung ngoại vi ngay khi điều khiển hoạt động ở chế độ sleep.
- Timer2: bộ đếm 8 bit với bộ chia tần số, bộ postcaler.
- Hai bộ Capture/so sánh/điều chế độ rộng xung.
- Các chuẩn giao tiếp nối tiếp SSP (Synchronous Serial Port), SPI và I2C.
- Chuẩn giao tiếp nối tiếp USART với 9 bit địa chỉ.
- Cổng giao tiếp song song PSP (Parallel Slave Port) với các chân điều khiển RD, WR, CS ở bên ngoài.

Các đặc tính Analog:

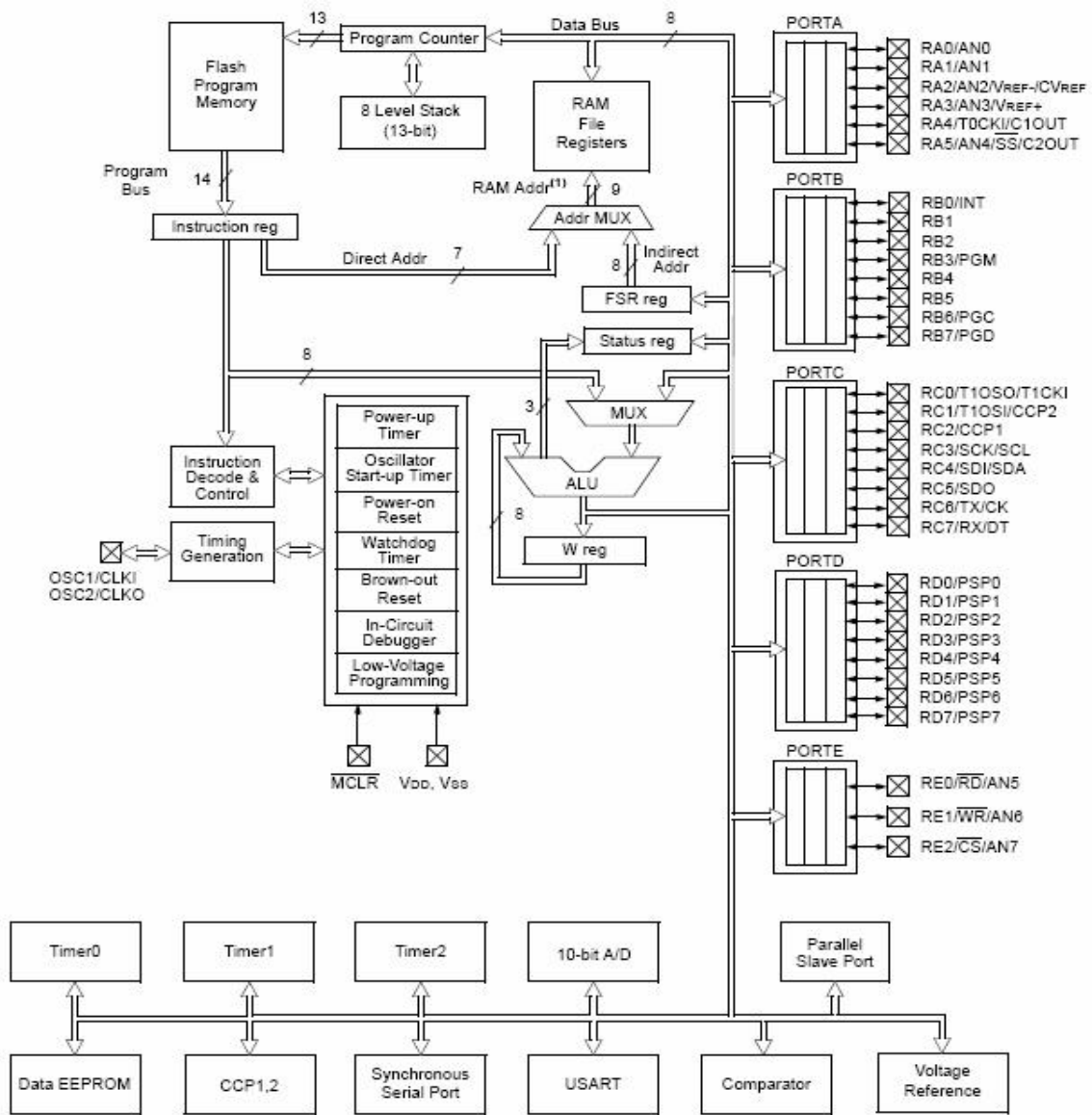
- 8 kênh chuyển đổi ADC 10 bit.
- Hai bộ so sánh.

Bên cạnh đó là một vài đặc tính khác của vi điều khiển như:

- Bộ nhớ flash với khả năng ghi xóa được 100.000 lần.
- Bộ nhớ EEPROM với khả năng ghi xóa được 1.000.000 lần.
- Dữ liệu bộ nhớ EEPROM có thể lưu trữ trên 40 năm.
- Khả năng tự nạp chương trình với sự điều khiển của phần mềm.
- Nạp được chương trình ngay trên mạch điện ICSP (In Circuit Serial Programming) thông qua 2 chân.
- Watchdog Timer với bộ dao động trong.
- Chức năng bảo mật chương trình.

- Chế độ Sleep.
- Có thể hoạt động với nhiều dạng Oscillator khác nhau.

### 2.4.6.3. Sơ đồ khối vi điều khiển PIC16F877A (hình 2.16)



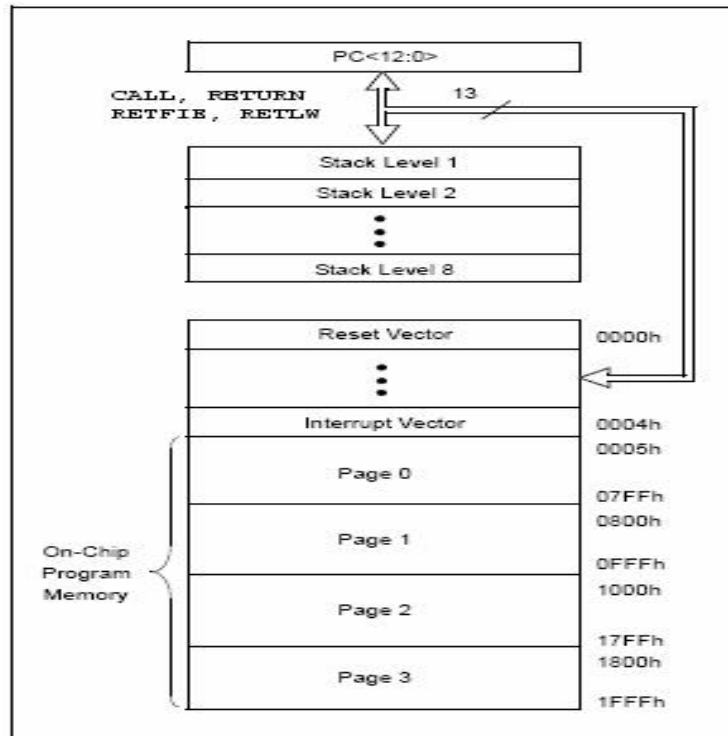
**Hình 2.16:** Sơ đồ khối vi điều khiển PIC16F877A.

### 2.4.6.4. Tổ chức bộ nhớ

Cấu trúc bộ nhớ của vi điều khiển PIC16F877A bao gồm bộ nhớ chương trình (Program memory) và bộ nhớ dữ liệu (Data Memory).

a) Bộ nhớ chương trình (Program memory) (hình 2.17)

Bộ nhớ chương trình của vi điều khiển PIC16F877A là bộ nhớ flash, dung lượng bộ nhớ 8K word (1 word =14 bit) và được phân thành nhiều trang (từ page 0 đến page 3). Như vậy bộ nhớ chương trình có khả năng chứa được  $8 \times 1024 = 8192$  lệnh (vì một lệnh sau khi mã hóa sẽ có dung lượng 1 word (14bit))



**Hình 2.17:** Bộ nhớ chương trình

Để mã hóa được địa chỉ của 8K word bộ nhớ chương trình, bộ đếm chương trình có dung lượng 13 bit(PC<12:0>).

Khi vi điều khiển được reset, bộ đếm chương trình sẽ chỉ đến địa chỉ 0000h (Reset vector). Khi có ngắt xảy ra, bộ đếm chương trình sẽ chỉ đến địa chỉ 0004h (Interrupt vector).

Bộ nhớ chương trình không bao giờ gồm bộ nhớ stack và không được địa chỉ hóa bởi bộ đếm chương trình.

b) Bộ nhớ dữ liệu (Data Memory)

Bộ nhớ dữ liệu của PIC là bộ nhớ EEPROM được chia ra làm nhiều bank. Đối với PIC16F877A bộ nhớ dữ liệu được chia ra làm 4 bank. Mỗi bank có dung lượng 128 byte, bao gồm các thanh ghi có chức năng đặc biệt SFG (Special Function Register) nằm ở các vùng địa chỉ thấp và các thanh ghi mục đích chung GFR (General Purpose Register) nằm ở vùng địa chỉ còn lại trong bank. Các thanh ghi SFG thường xuyên được sử dụng (ví dụ như thanh ghi STATUS) sẽ được đặt ở tất cả các bank của bộ nhớ dữ liệu giúp thuận tiện trong quá trình truy xuất và làm giảm bớt lệnh của chương trình.

Sơ đồ cụ thể của bộ nhớ dữ liệu PIC16F877A như hình 2.18

File Address		File Address		File Address		File Address	
Indirect addr. <sup>(*)</sup>	00h	Indirect addr. <sup>(*)</sup>	80h	Indirect addr. <sup>(*)</sup>	100h	Indirect addr. <sup>(*)</sup>	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved <sup>(2)</sup>	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved <sup>(2)</sup>	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch	CMCON	9Ch		11Ch		19Ch
CCP2CON	1Dh	CVRCON	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes	
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h - 7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.

**Note 1:** These registers are not implemented on the PIC16F876A.  
**Note 2:** These registers are reserved; maintain these registers clear.

**Hình 2.18:** Bộ nhớ dữ liệu PIC16F877A.

c) Các thanh ghi chức năng đặc biệt SFR và thanh ghi mục đích chung GPR

d) Stack

Stack không nằm trong bộ nhớ chương trình hay bộ nhớ dữ liệu mà là một vùng nhớ đặc biệt không cho phép đọc hay ghi.

Bộ nhớ stack trong vi điều khiển PIC họ 16F87xA có khả năng chứa được 8

địa chỉ và hoạt động theo cơ chế xoay vòng.

Không có cờ hiệu nào cho biết trạng thái stack, do đó ta không biết được khi nào stack tràn. Bên cạnh đó tập lệnh của vi điều khiển dòng PIC cũng không có lệnh POP hay PUSH, các thao tác với bộ nhớ stack sẽ hoàn toàn được điều khiển bởi CPU.

*e) Các cổng xuất nhập của PIC16F877A*

Cổng xuất nhập (I/O port) chính là phương tiện mà vi điều khiển dùng để tương tác với thế giới bên ngoài.

Vi điều khiển PIC 16F877A có 5 cổng xuất nhập, bao gồm PORTA, PORTB, PORTC, PORTD và PORTE.

PORTA (RPA) bao gồm 6 I/O pin. Đây là các chân “hai chiều” (bidirectional pin), nghĩa là có thể xuất và nhập được. Chức năng I/O này được điều khiển bởi thanh ghi TRISA (địa chỉ 85h).

PORTB (RPB) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISB. Bên cạnh đó một số chân của PORTB còn được sử dụng trong quá trình nạp chương trình cho vi điều khiển với các chế độ nạp khác nhau. PORTB còn liên quan đến ngắt ngoại vi và bộ Timer0. PORTB còn được tích hợp chức năng điện trở kéo lên được điều khiển bởi chương trình.

PORTC (RPC) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISC. Bên cạnh đó PORTC còn chứa các chân chức năng của bộ so sánh, bộ Timer1, bộ PWM và các chuẩn giao tiếp nối tiếp 12C, SPI, SSP, USART.

PORTD (RPD) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISD. PORTD còn là cổng xuất dữ liệu của chuẩn giao tiếp PSP (Parallel Slave Port).

PORTE (RPE) gồm 3 chân I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISE. Các chân của PORTE có ngõ vào analog, bên cạnh đó PORTE còn là các chân điều khiển của chuẩn giao tiếp PSP.

*f) Các bộ định thời timer0, timer1, timer2, ADC, comparator, ...*

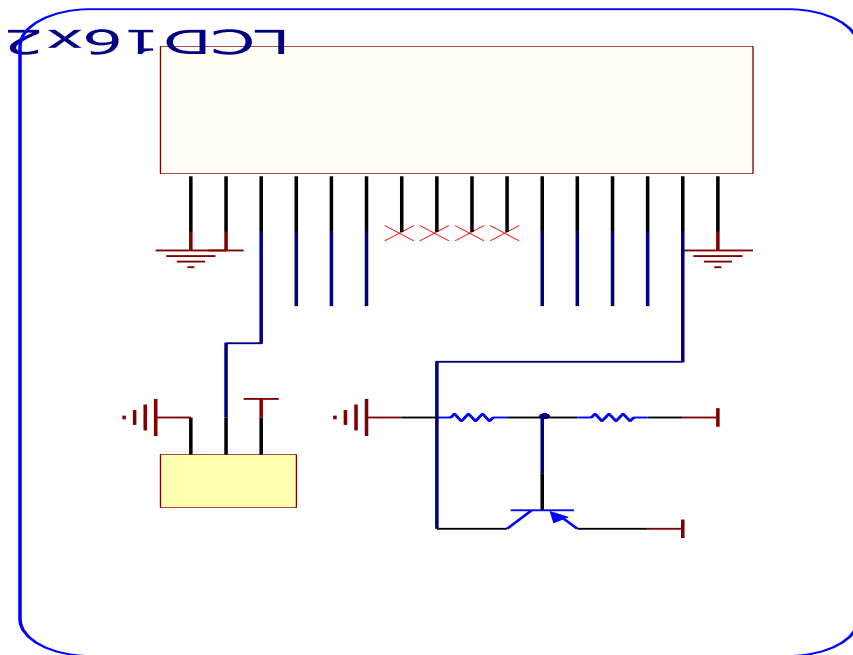
## **2.5.Màn hình LCD (hình 2.19)**

### **2.5.1.Hoạt động của LCD**



Trong những năm gần đây LCD đang ngày càng được sử dụng rộng rãi thay thế dần cho các đèn LED (các đèn LED 7 đoạn hay nhiều đoạn). Đó là kết quả dựa trên những ưu điểm của LCD như sau:

- Các LCD có giá thành hạ.
- Khả năng hiển thị các số, các ký tự và đồ họa tốt hơn nhiều so với các đèn LED (vì các đèn LED chỉ hiển thị được các số và một số ký tự).
- Nhờ kết hợp một bộ điều khiển làm tươi vào LCD làm giải phóng cho CPU công việc làm tươi LCD. Trong khi đèn LED phải được làm tươi bằng CPU (hoặc bằng cách nào đó) để duy trì việc hiển thị dữ liệu.
- Dễ dàng lập trình cho các ký tự và đồ họa.



**Hình 2.19:** LCD1602M là màn hình LCD 2 dòng và 16 kí tự.

### 2.5.2. Mô tả các chân của LCD

Mô tả các chân, loại 14 chân:

- VCC: cấp nguồn dương

- VSS: cấp nguồn âm
- VEE: điều khiển độ tương phản của LCD

Chân chọn thanh ghi RS( Register Select). Có hai thanh ghi rất quan trọng trong LCD, chân RS được dùng để chọn thanh ghi này như sau: Nếu  $RS = 0$  thì thanh ghi mà lệnh được chọn để cho phép người dùng gửi một lệnh chẳng hạn như xóa màn hình, đưa con trỏ về đầu dòng...Nếu  $RS=1$  thì thanh ghi dữ liệu được chọn cho phép người dùng gửi dữ liệu cần hiển thị trên LCD.

- Chân đọc/ghi (R/W): Đầu đọc/ghi cho phép người dùng ghi thông tin lên LCD khi  $R/W=0$  hoặc đọc thông tin từ nó khi  $R/W=1$ .
- Chân cho phép E (Enable): Chân cho phép E được sử dụng bởi LCD để chốt thông tin hiện hữu trên chân dữ liệu của nó. Khi dữ liệu được cung cấp đến chân dữ liệu thì một xung mức cao xuống thấp phải được áp đến chân này để LCD chốt dữ liệu trên các chân dữ liệu. Xung này phải rộng tối thiểu là 450 ns.
- Chân D0~D7: Đây là 8 chân dữ liệu 8 bit, được dùng để gửi thông tin trên LCD hoặc đọc nội dung của các thanh ghi trong LCD.

Để hiển thị các chữ cái và các con số, chúng ta gửi các mã ASCII của các chữ cái từ A đến Z, a đến f và các con số từ 0~9 đến các chân này khi bật  $RS=1$ . Cũng như các mã lệnh mà có thể được gửi đến LCD để xóa màn hình hoặc đưa con trỏ về vị trí đầu dòng hoặc nhấp nháy con trỏ. Chúng ta cũng sử dụng  $RS=0$  để kiểm tra bit cờ bận để xem LCD có sẵn sàng nhận thông tin. Cờ bận là D7 và có thể được đọc khi  $R/W=1$  và  $RS=0$  như sau:

Nếu  $R/W=1$ ,  $RS=0$  khi  $D7=1$ (cờ bận 1) thì LCD bận bởi các công việc bên trong và sẽ không nhận bất kì thông tin mới nào. Khi  $D7$  thì LCD sẵn sàng nhận thông tin mới. Lưu ý chúng ta nên kiểm tra cờ bận trước khi ghi bất kì dữ liệu nào lên LCD.

Chân	Kí hiệu	I/O	Mô tả
1	VSS	-	Đất
2	VCC	-	Dương 5V
3	VEE	-	Cấp nguồn điều khiển phản
4	RS	I	RS=0 chọn thanh ghi lệnh. RS=1 chọn thanh ghi dữ liệu.
5	R/W	I	R/W=1 đọc dữ liệu. R/W=0 ghi
6	E	I/O	Cho phép
7	DB0	I/O	Các bit dữ liệu
8	DB1	I/O	Các bit dữ liệu
9	DB2	I/O	Các bit dữ liệu
10	DB3	I/O	Các bit dữ liệu
11	DB4	I/O	Các bit dữ liệu
12	DB5	I/O	Các bit dữ liệu
13	DB6	I/O	Các bit dữ liệu
14	DB7	I/O	Các bit dữ liệu

Các mã lệnh LCD:

Mã Hex	Lệnh đến thanh ghi của LCD
1	Xóa màn hình hiển thị
2	Trở về đầu dòng
4	Giảm con trỏ (dịch con trỏ sang trái)
5	Tăng con trỏ (dịch con trỏ sang phải)
6	Dịch hiển thị sang phải
7	Dịch hiển thị sang trái
8	Tắt con trỏ, tắt hiển thị
A	Tắt hiển thị, bật con trỏ
C	Bật hiển thị, tắt con trỏ
E	Bật hiển thị, nhấp nháy con trỏ
F	Tắt hiển thị, nhấp nháy con trỏ
10	Dịch vị trí con trỏ sang trái
14	Dịch vị trí con trỏ sang phải
18	Dịch toàn bộ hiển thị sang trái
1C	Dịch toàn bộ hiển thị sang phải
80	Ép con trỏ Vũ đầu dòng thứ nhất
C0	Ép con trỏ Vũ đầu dòng thứ hai
38	Hai dòng ma trận

## 2.6.Các khối mạch điều khiển

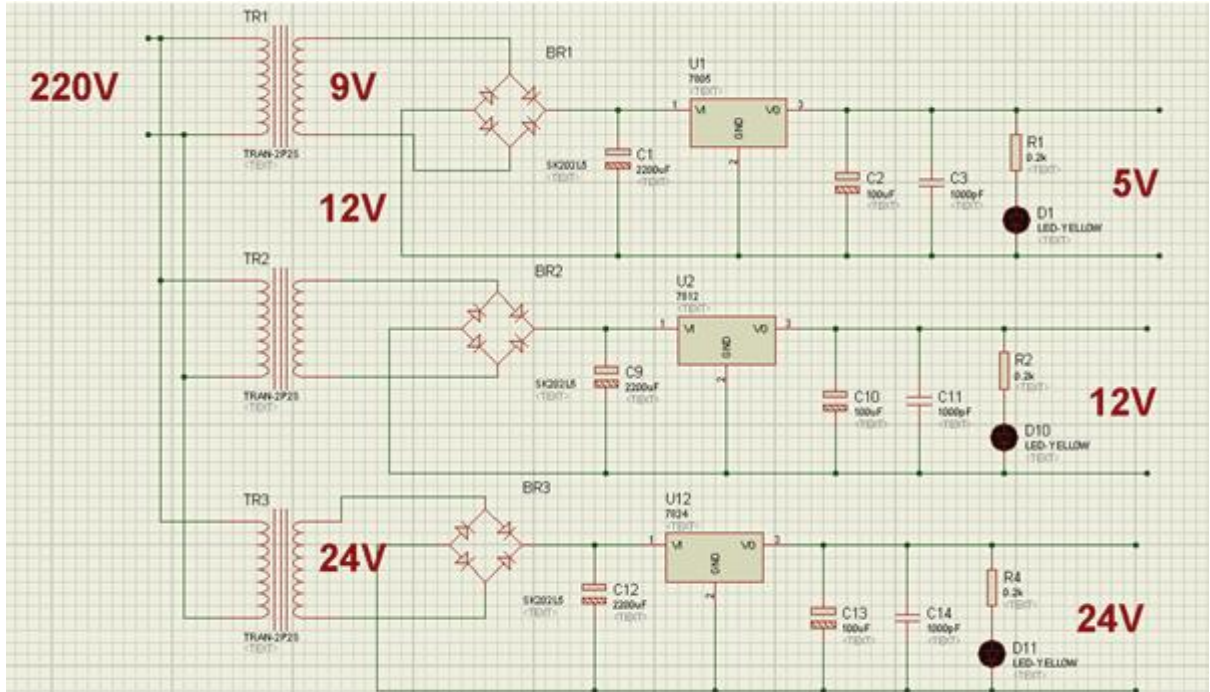
### 2.6.1.Sơ đồ

Danh sách các linh kiện sử dụng

Thứ tự	Linh kiện	Số lượng
1	PIC16F877A	2
2	ATC89C52	1
3	LCD16x2	1
4	Nút bấm	2
5	Biến trở	2
6	Thạch anh 12M	3
7	Tụ 33p	6
8	Role 24V	2
9	IC DS1307	1
10	Opto 4 chân	6
11	Diode Zener	3
12	Diode thường	9
13	Led	1
14	Transitor C408	6
15	Transitor C1815	6
16	Điện trở 100 $\Omega$	10
17	Điện trở 560 $\Omega$	10
18	Điện trở 1.5 $\Omega$	10
19	Điện trở 2.2 $\Omega$	10

## 2.6.2. Chức năng từng khối

### 2.6.2.1. Khối nguồn (hình 2.20)



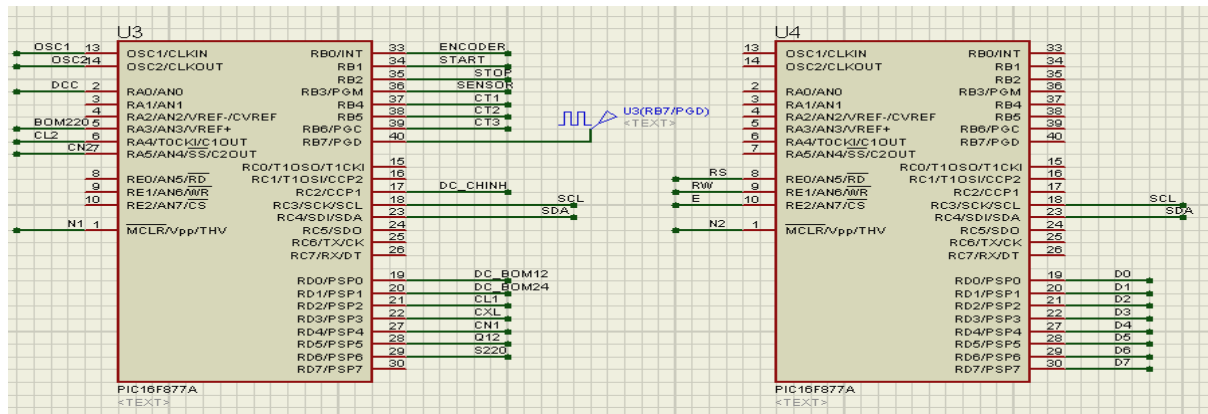
**Hình 2.20:** Sơ đồ khối nguồn

Cung cấp nguồn nuôi cho mạch động lực và mạch điều khiển.

Khối nguồn được thiết kế để cung cấp mức điện áp ổn định 6V, 12V, 24V cho mô hình. Dùng biến áp để chuyển từ nguồn 220VAC sang 24VAC, 12VAC, 6VAC cung cấp vào module mạch nguồn trên. Cầu diode Bridge BR1, BR2, BR3 dùng để dòng xoay chiều thành một chiều. Khối mạch gồm các linh kiện C1, C2, C3, C9, C10, C11, C12, C13, C14, R1, R2, R4. Các tụ trong mạch này đều có tác dụng lọc nhiễu nguồn vào, cũng như lọc nhiễu nguồn đầu ra. Làm cho nguồn tương đối ổn định, không bị ảnh hưởng nhiều bởi tải (tải nhỏ). Các IC 7824, 7812, 7805 là 3 IC ổn áp, lần lượt ổn áp 24V, 12V và 5V ở đầu ra, Led D1, D10, D11 để hiển thị ở đầu ra.

## 2.6.2.2. Khối vi điều khiển PIC16F877A

### a) Sơ đồ nguyên lý (hình 2.21)



**Hình 2.21:** Sơ đồ nguyên lý.

### b) Giải thích

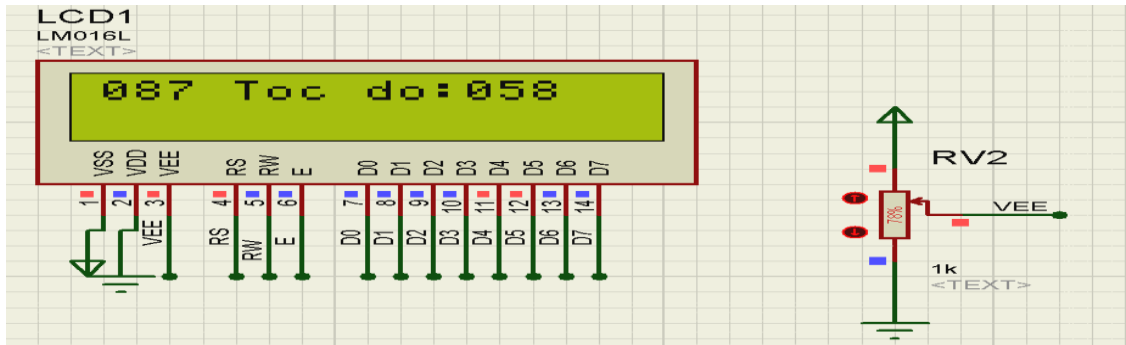
Vi điều khiển 16F877A hoạt động khi chân số 1 MCLR (Master clear) ở mức 1 và bị reset khi bị clear về mức 0.

Trung tâm điều khiển các chế độ làm việc của động cơ và toàn bộ mọi quá trình hoạt động khác: điều khiển động cơ chạy ở tốc độ mong muốn, điều khiển hoạt động mạch hiển thị LCD, đo tốc độ động cơ chính.

Khối gồm có IC 16F877A, mạch tạo xung và reset.

Với mạch reset ở trên, bình thường phím nhấn hờ ra, chân reset mức 1, vi điều khiển hoạt động. Khi nhấn phím xuống, chân 1 chạm đất bị clear về 0, vi điều khiển được reset trở lại trạng thái ban đầu.

### 2.6.2.3. Khối hiển thị (hình 2.22)



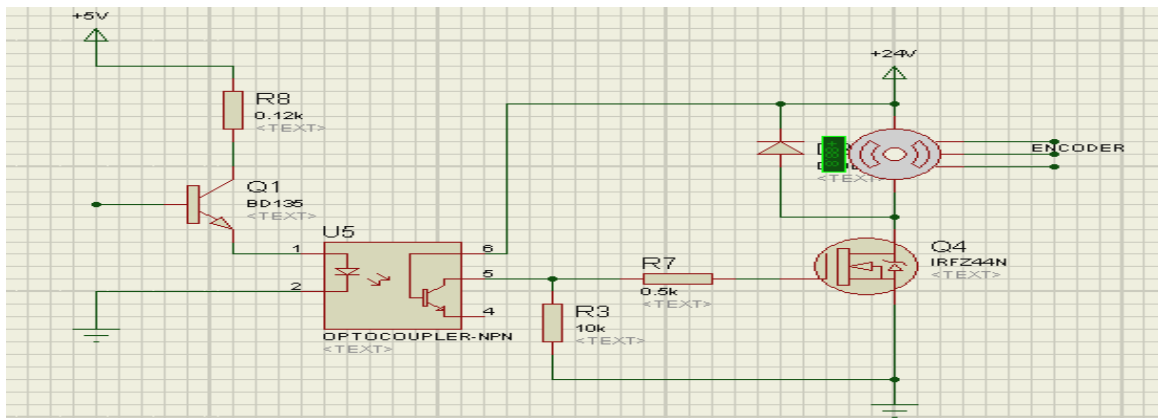
**Hình 2.22:** Khối hiển thị màn hình LCD.

Dùng để hiển thị tốc độ đo của động cơ trong mô hình mạch thực tế.

Khối này là 1 màn hình LCD 16x2 được lấy tín hiệu và nhận dữ liệu trực tiếp từ Vi điều khiển.

### 2.6.2.4. Khối mạch động lực

#### a) Động cơ chính (hình 2.23)

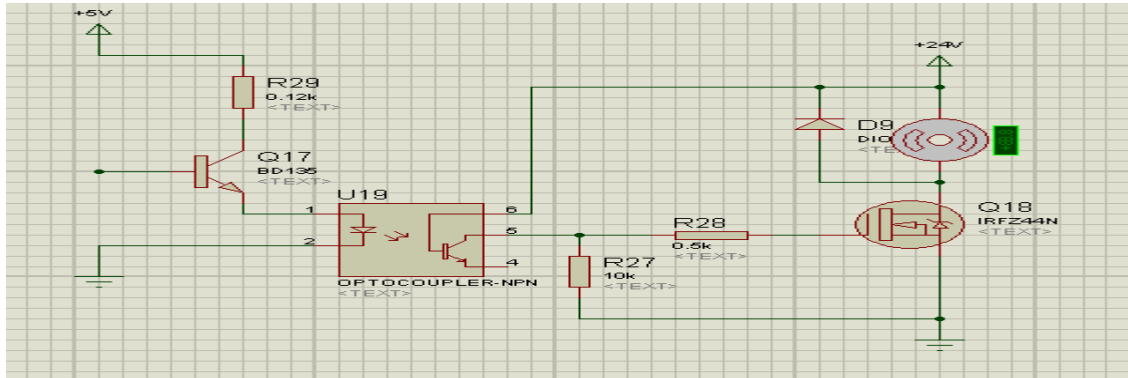


**Hình 2.23:** Mạch động cơ chính.

Khối này gồm mạch động lực cung cấp công suất cho động cơ kéo. Bộ phận đo tốc độ là 1 encoder 100 xung lấy tín hiệu đưa về Vi điều khiển để đo tốc độ và tự động điều chỉnh cũng như ổn định ở tốc độ động cơ ở tốc độ đặt.



b) Động cơ phụ (hình 2.24)



**Hình 2.24:** Mạch động cơ phụ.

Khối này gồm mạch động lực cung cấp công suất cho các động cơ chổi lắ, chổi xoay lớn, chổi xoay nhỏ, bơm ....

## CHƯƠNG 3

# THIẾT KẾ THUẬT TOÁN VÀ CHƯƠNG TRÌNH ĐIỀU KHIỂN CHO NHÀ RỬA XE TỰ ĐỘNG

### 3.1. Nguyên lý hoạt động của hệ thống

Mô hình nhà rửa xe tự động hoạt động nhờ hệ thống các động cơ được vi điều khiển PIC16F877A đã được lập trình cấp tín hiệu để hoạt động.

Khi ấn nút Start trên tủ điện, hệ thống bắt đầu hoạt động, động cơ chính quay kéo đường ray chuyển động cùng chiều làm ô tô chuyển động dọc theo đường ray. Khi ô tô đi qua thiết bị cảm biến, sau 2 giây hệ thống vòi phun nước bắt đầu hoạt động phun nước, đồng thời sau 2,5 giây hệ thống vòi phun chất tẩy rửa cũng được hoạt động. Hệ thống bơm này sẽ bơm nước và chất tẩy rửa trong vòng 4s.

Sau khi ô tô đi qua giàn phun nước và chất tẩy rửa công tắc hành trình số 1 được đóng lại khi chạm vào thành xe làm chổi lắc số 1, cặp chổi nhỏ số 1 và cặp chổi lớn hoạt động. Thời gian hoạt động của chổi lắc số 1 là 4 giây, cặp chổi nhỏ số 1 là 4 giây, cặp chổi lớn là 5 giây. Khoảng thời gian này chính là khoảng thời gian ô tô được làm sạch sơ bộ bằng hệ thống các cặp chổi.

Sau đó, ô tô được đưa đến giàn phun nước làm sạch, hệ thống nước này được hoạt động khi công tắc hành trình số 2 đóng. Hoạt động đồng thời sau đó là cặp chổi nhỏ số 2 và chổi lắc số 2. Toàn bộ hệ thống này hoạt động trong 4 giây, và đây là khoảng thời gian xe được rửa sạch nhất.

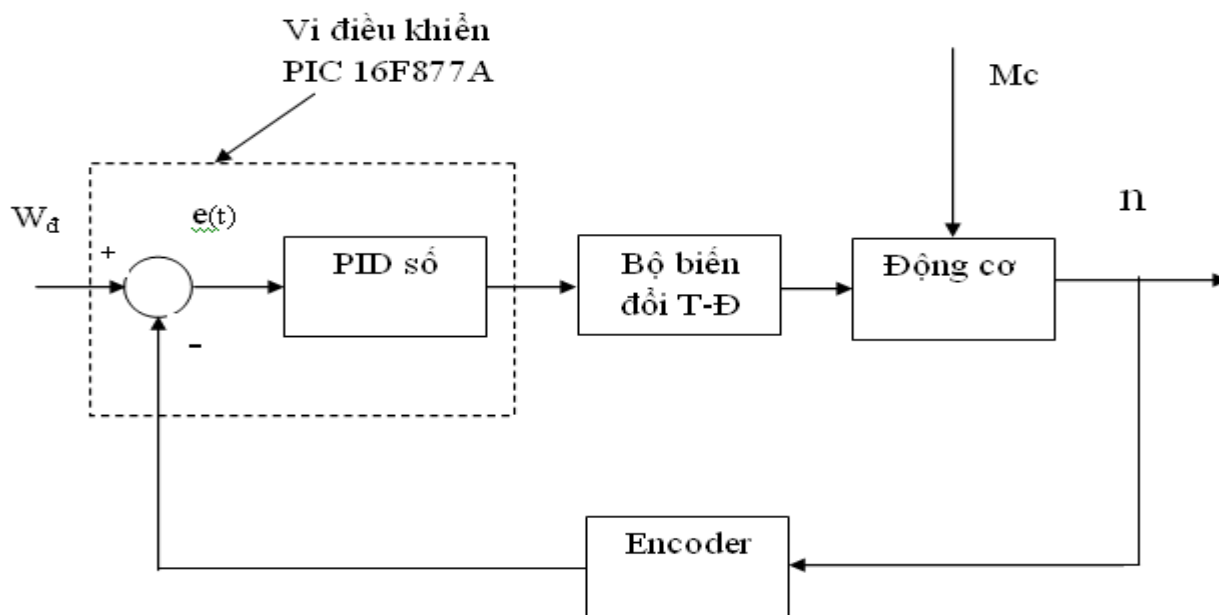
Công đoạn cuối cùng là làm khô xe. Thành xe ô tô chạm vào công tắc hành trình số 3 sau khi đi qua cặp chổi nhỏ số 2, làm cho 2 quạt gió và máy sấy khô hoạt động trong vòng 4 giây.

Kết thúc quá trình này, ô tô đã được rửa sạch hoàn toàn, hệ thống lại sẵn sàng cho lần rửa xe tiếp theo.

## 3.2. Thuật toán PID trong việc điều khiển tốc độ động cơ

### 3.2.1. Sơ đồ khối bộ điều chỉnh PID động cơ một chiều bằng vi điều khiển

Sơ đồ khối chức năng hệ thống điều khiển động cơ một chiều bằng vi xử lý như sau:



**Hình 3.1:** Sơ đồ khối hệ thống điều khiển

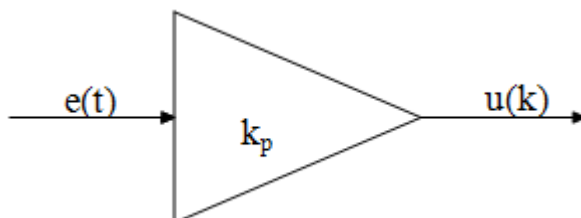
Hệ thống lấy giá trị đặt  $W_d$  do người sử dụng thiết lập từ bên ngoài. Giá trị thực tế của động cơ đo được là tốc độ thông qua cảm biến tốc độ (encoder), sau đó bộ điều khiển sẽ tính sai lệch  $e(t)$  giữa giá trị đặt và giá trị đạt được rồi tính toán đầu ra của bộ điều khiển theo luật PID để xuất tín hiệu điều khiển đối tượng. Hệ thống với thuật toán luật điều khiển PID số sẽ có xu hướng luôn đưa sai lệch  $e(t)$  về giá trị 0, tức là giá trị đạt được sau 1 thời gian sẽ bằng giá trị đặt. Do đó tốc độ của động cơ luôn ổn định.

### 3.2.2. Các luật điều khiển số

Yêu cầu thiết kế được đặt ra là bộ PID số phải có tính linh hoạt cao, có nghĩa là phải có giao điều khiển các đối tượng công nghiệp theo luật PID và có thể lựa chọn tham số của các luật phù hợp với đối tượng thiết kế. Luật PID số phải được

thiết kế gọn gàng, giao diện thân thiện với người sử dụng. Thông qua HMI, người sử dụng có thể chọn luật điều khiển dễ dàng. Ví dụ như có thể gian xử lý lệnh phải nhanh để làm tăng tính thời gian thực cho thiết bị điều khiển.

### 3.2.2.1. Luật điều khiển tỷ lệ số

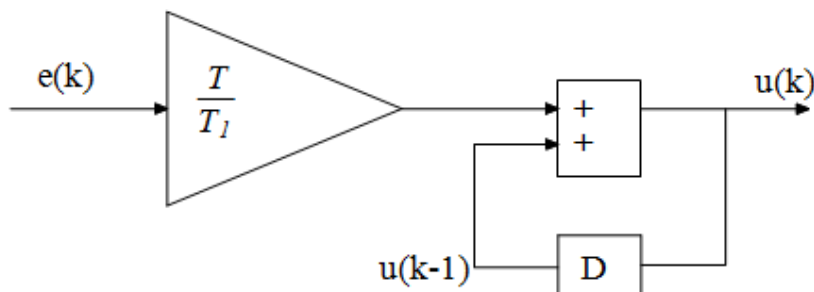


**Hình 3.2:** Cấu trúc luật P số.

Đây là luật điều khiển có thể thiết kế đơn giản nhất. Dãy  $u(k)$  được tính tuwf dãy  $e(k)$  theo công thức:

$$u(k) = k_p e(k) \quad k=0,1,2 \dots \quad (3.1)$$

### 3.2.2.2. Luật điều khiển tích phân số



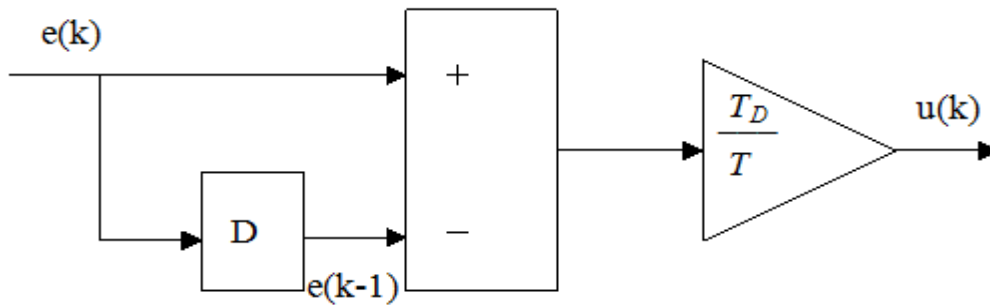
**Hình 3.3:** Cấu trúc luật I số.

Ta có phương trình sai phân:

$$u(k) = \frac{T}{T_i} e(k) + u(k-1) \quad (3.2)$$

Trong đó  $T$  là thời gian trích mẫu (Sample Time)

### 3.2.2.3. Luật điều khiển vi phân số



**Hình 3.4:** Cấu trúc luật D số.

Thường các bộ điều khiển theo luật vi phân số được cài đặt theo các phương trình sai phân sau:

$$u(k) = \frac{T_D}{T} [e(k) - e(k-1)] \quad (3.3)$$

Trong đó T là thời gian trích mẫu.

### 3.2.2.4. Luật điều khiển PID số (hình 3.5)

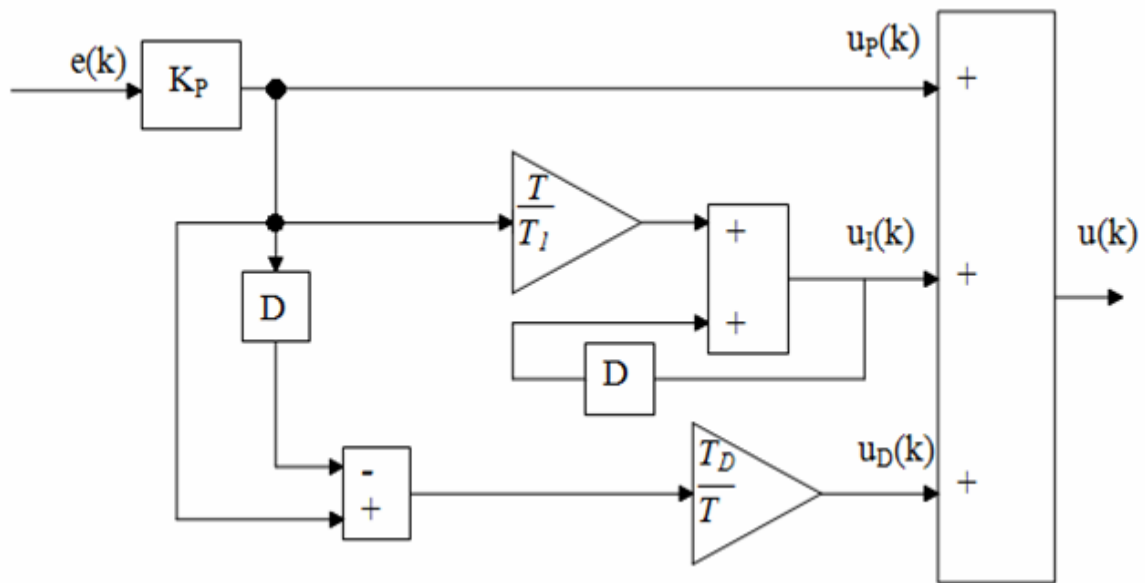
Luật điều khiển PID số trong công thức trên được lựa chọn để cài đặt cho bộ điều khiển được chế tạo trên chip PIC.

Từ cấu trúc PID số trong Hình 4.5, ta có:

$$u(k) = k_p \left\{ e(k) + \frac{T}{T_i} e(k) + u_i(k-1) + \frac{T_D}{T} [e(k) - e(k-1)] \right\} \quad (3.4)$$

$$u(k) = k_p \left\{ \left(1 + \frac{T_D}{T}\right) e(k) - \frac{T_D}{T} e(k-1) + \frac{T}{T_i} e(k) + u_i(k-1) \right\} \quad (3.5)$$

$$u(k) = k_p \left\{ \left(1 + \frac{T_D}{T} + \frac{T}{T_i}\right) e(k) - \frac{T_D}{T} e(k-1) + u_i(k-1) \right\} \quad (3.6)$$



**Hình 3.5:** Cấu trúc luật PID số.

### 3.2.3. Xây dựng bộ PID dùng vi điều khiển PIC16F877A

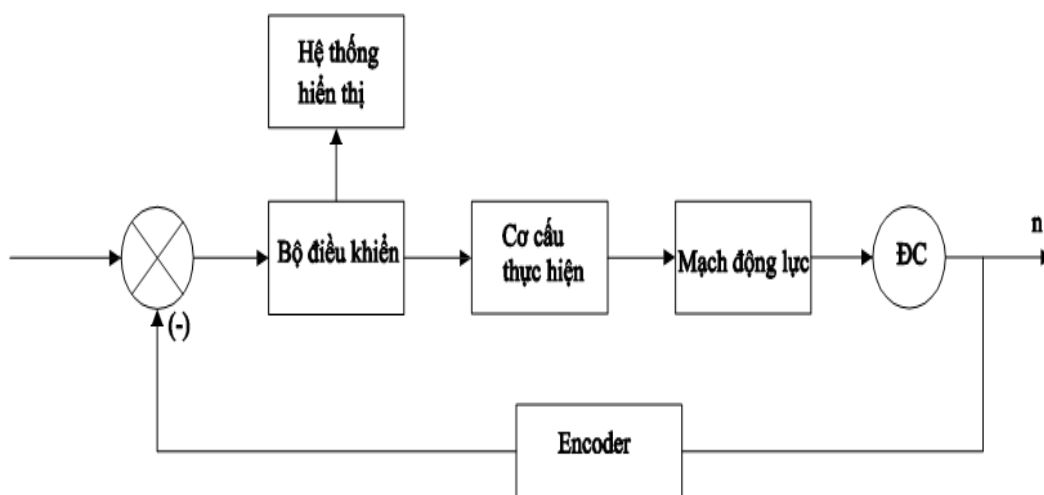
Giá trị đặt  $W_d$  do người sử dụng đưa vào bộ điều khiển qua hệ thống điều chỉnh (biến trở), giá trị đặt được thể hiện trên hệ thống hiển thị. Tốc độ thực tế của động cơ thông qua cảm biến tốc độ (encoder) cũng được đưa vào bộ điều khiển và thể hiện trên hệ thống hiển thị. Bộ điều khiển sẽ tính sai lệch  $e$  giữa giá trị đặt và giá trị đạt được rồi tính toán đầu ra của bộ điều khiển theo luật PID để xuất tín hiệu điều khiển đối tượng thông qua mạch động lực.

Hệ thống với thuật toán luật điều khiển PID số sẽ có xu hướng luôn đưa sai lệch  $e(t)$  về giá trị 0, tức là giá trị đạt được sau 1 thời gian sẽ bằng giá trị đặt.

Do đó tốc độ của động cơ luôn ổn định. Trên hệ thống hiển thị sẽ thể hiện giá trị đặt và giá trị tốc độ động cơ đạt được là bằng nhau.

Ta sử dụng bộ điều chế độ rộng xung (PWM) tích hợp sẵn bên trong PIC với 2 ngõ ra xung tại hai chân CCP1(17) và CCP2 (16) . Tại các chân này khi hoạt động sẽ xuất chuỗi xung vuông , độ rộng điều chỉnh được dễ dàng. Xung ra này dùng để tạo tín hiệu đóng ngắt Trasiistor trong mạch động lực, với độ rộng xác định

sẽ tạo ra một điện áp trung bình xác định. Thay đổi độ rộng xung sẽ thay đổi điện áp trung bình và do đó thay đổi được tốc độ động cơ.



*Hình 3.6:* Sơ đồ khối bộ PID dùng vi điều khiển PIC16F877A.

### 3.3. Giới thiệu về ngôn ngữ lập trình C

#### 3.3.1. Tổng quan về ngôn ngữ lập trình C

C là ngôn ngữ lập trình cấp cao, được sử dụng rất phổ biến để lập trình hệ thống cùng với Assembler và phát triển các ứng dụng.

Ngôn ngữ lập trình C là một ngôn ngữ lập trình hệ thống rất mạnh và rất “mềm dẻo”, có một thư viện gồm rất nhiều các hàm (function) đã được tạo sẵn. Người lập trình có thể tận dụng các hàm này để giải quyết các bài toán mà không cần phải tạo mới. Hơn thế nữa, ngôn ngữ C hỗ trợ rất nhiều phép toán nên phù hợp cho việc giải quyết các bài toán kỹ thuật có nhiều công thức phức tạp. Ngoài ra, C cũng cho phép người lập trình tự định nghĩa thêm các kiểu dữ liệu trừu tượng khác. Tuy nhiên, điều mà người mới vừa học lập trình C thường gặp “rắc rối” là “hơi khó hiểu” do sự “mềm dẻo” của C. Dù vậy, C được phổ biến khá rộng rãi và đã trở thành một công cụ lập trình khá mạnh, được sử dụng như là một ngôn ngữ lập trình chủ yếu trong việc xây dựng những phần mềm hiện nay.

Ngôn ngữ C có những đặc điểm cơ bản sau:

- Tính cô đọng (compact): C chỉ có 32 từ khóa chuẩn và 40 toán tử chuẩn, nhưng hầu hết đều được biểu diễn bằng những chuỗi ký tự ngắn gọn.
- Tính cấu trúc (structured): C có một tập hợp những chỉ thị của lập trình như cấu trúc lựa chọn, lặp... Từ đó các chương trình viết bằng C được tổ chức rõ ràng, dễ hiểu.
- Tính tương thích (compatible): C có bộ tiền xử lý và một thư viện chuẩn vô cùng phong phú nên khi chuyển từ máy tính này sang máy tính khác các chương trình viết bằng C vẫn hoàn toàn tương thích.
- Tính linh động (flexible): C là một ngôn ngữ rất uyển chuyển và cú pháp, chấp nhận nhiều cách thể hiện, có thể thu gọn kích thước của các mã lệnh làm chương trình chạy nhanh hơn.
- Biên dịch (compile): C cho phép biên dịch nhiều tập tin chương trình riêng rẽ thành các tập tin đối tượng (object) và liên kết (link) các đối tượng đó lại với nhau thành một chương trình có thể thực thi được (executable) thống nhất.

### **3.3.2.Cấu trúc của một chương trình C**

#### *3.3.2.1.Tiền xử lý và biên dịch*

Trong C, việc dịch (translation) một tập tin nguồn được tiến hành trên hai bước hoàn toàn độc lập với nhau:

- Tiền xử lý.
- Biên dịch.

Hai bước này trong phần lớn thời gian được nối tiếp với nhau một cách tự động theo cách thức mà ta có ấn tượng rằng nó đã được thực hiện như là một xử lý duy nhất.

Nói chung, ta thường nói đến việc tồn tại của một bộ tiền xử lý (preprocessor)



nhằm chỉ rõ chương trình thực hiện việc xử lý trước. Ngược lại, các thuật ngữ trình biên dịch hay sự biên dịch vẫn còn nhập nhằng bởi vì nó chỉ ra khi thì toàn bộ hai giai đoạn, khi thì lại là giai đoạn thứ hai.

Bước tiền xử lý tương ứng với việc cập nhật trong văn bản của chương trình nguồn, chủ yếu dựa trên việc diễn giải các mã lệnh rất đặc biệt gọi là các chỉ thị dẫn hướng của bộ tiền xử lý (destination directive of preprocessor); các chỉ thị này được nhận biết bởi chúng bắt đầu bằng ký hiệu (symbol) #.

Hai chỉ thị quan trọng nhất là:

- Chỉ thị sự gộp vào của các tập tin nguồn khác: #include
- Chỉ thị việc định nghĩa các macros hoặc ký hiệu: #define

Chỉ thị đầu tiên được sử dụng trước hết là nhằm gộp vào nội dung của các tập tin cần có (header file), không thể thiếu trong việc sử dụng một cách tốt nhất các hàm của thư viện chuẩn, phổ biến nhất là:

```
#include <stdio.h>
```

Chỉ thị thứ hai rất hay được sử dụng trong các tập tin thư viện (header file) đã được định nghĩa trước đó và thường được khai thác bởi các lập trình viên trong việc định nghĩa các ký hiệu như là:

```
#define NB_COUPS_MAX 100
```

```
#define SIZE 25
```

### 3.3.2.2. Cấu trúc một chương trình C

Một chương trình C bao gồm các phần như: Các chỉ thị tiền xử lý, khai báo biến ngoài, các hàm tự tạo, chương trình chính (hàm main).

Cấu trúc có thể như sau :

## Các chỉ thị tiền xử lý (Preprocessor directives)

```
#include <Tên tập tin thư viện>
```

```
#define ....
```

**Định nghĩa kiểu dữ liệu** (phần này không bắt buộc): dùng để đặt tên lại cho một kiểu dữ liệu nào đó để gọi nhớ hay đặt 1 kiểu dữ liệu cho riêng mình dựa trên các kiểu dữ liệu đã có.

**Cú pháp: typedef** <Tên kiểu cũ> <Tên kiểu mới>

*Ví dụ:* typedef int SoNguyen; // Kiểu SoNguyen là kiểu int

**Khai báo các prototype** (tên hàm, các tham số, kiểu kết quả trả về,... của các hàm sẽ cài đặt trong phần sau, phần này không bắt buộc): phần này chỉ là các khai báo đầu hàm, không phải là phần định nghĩa hàm.

**Khai báo các biến ngoài** (các biến toàn cục) *phần này không bắt buộc*: phần này khai báo các biến toàn cục được sử dụng trong cả chương trình.

**Chương trình chính phần này bắt buộc phải có**

```
<Kiểu dữ liệu trả về> main()
```

```
{
```

**Các khai báo cục bộ trong hàm main:** Các khai báo này chỉ tồn tại trong hàm mà thôi, có thể là khai báo biến hay khai báo kiểu.

**Các câu lệnh dùng để định nghĩa hàm main**

```
return <kết quả trả về>; // Hàm phải trả về kết quả
```

```
}
```

**Cài đặt các hàm**

```
<Kiểu dữ liệu trả về> function1( các tham số)
```

{

Các khai báo cục bộ trong hàm.

Lưu ý: Một số tập tin header thường dùng:

}

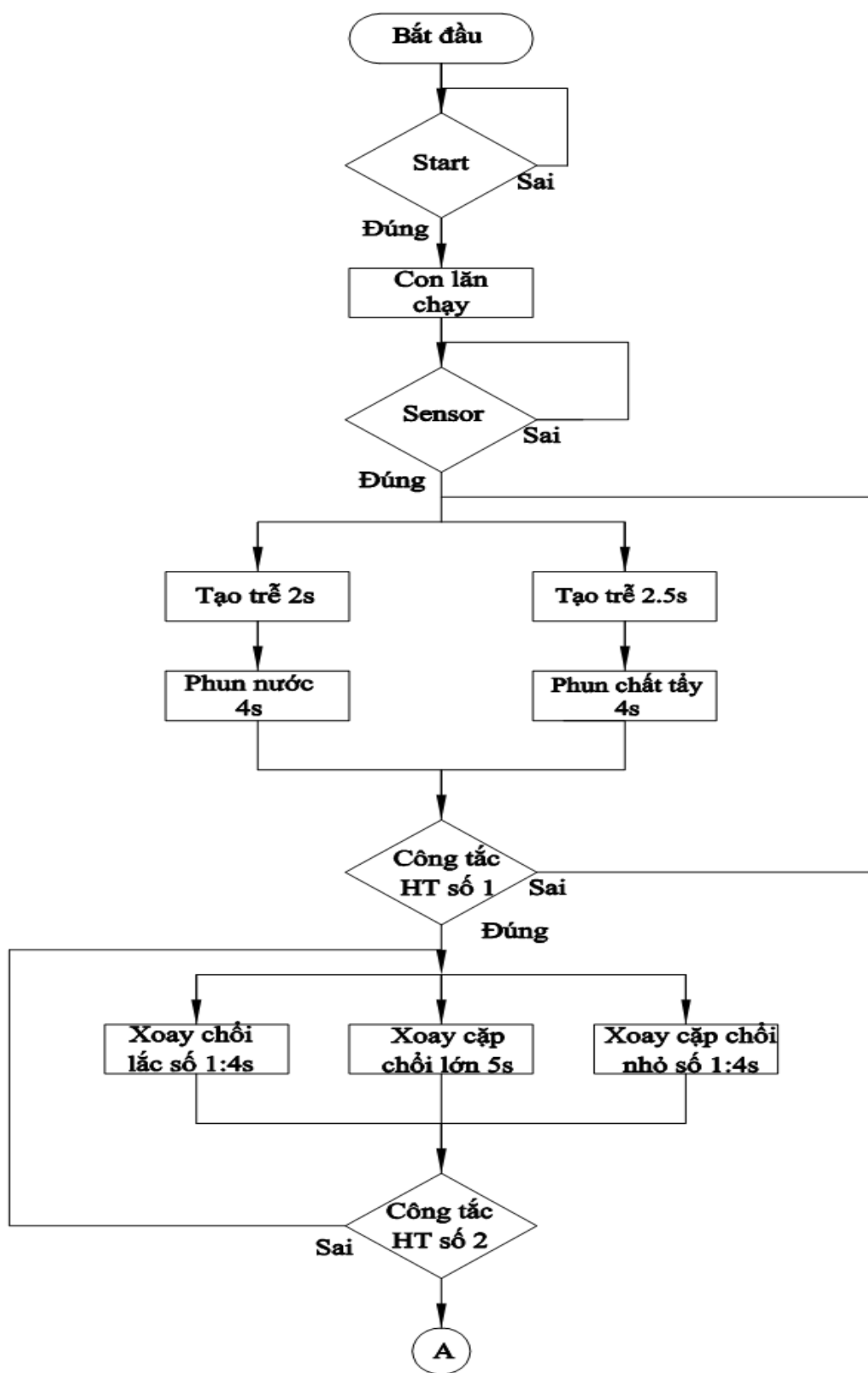
...

Một chương trình C bắt đầu thực thi từ hàm main (thông thường là từ câu lệnh đầu tiên đến câu lệnh cuối cùng).

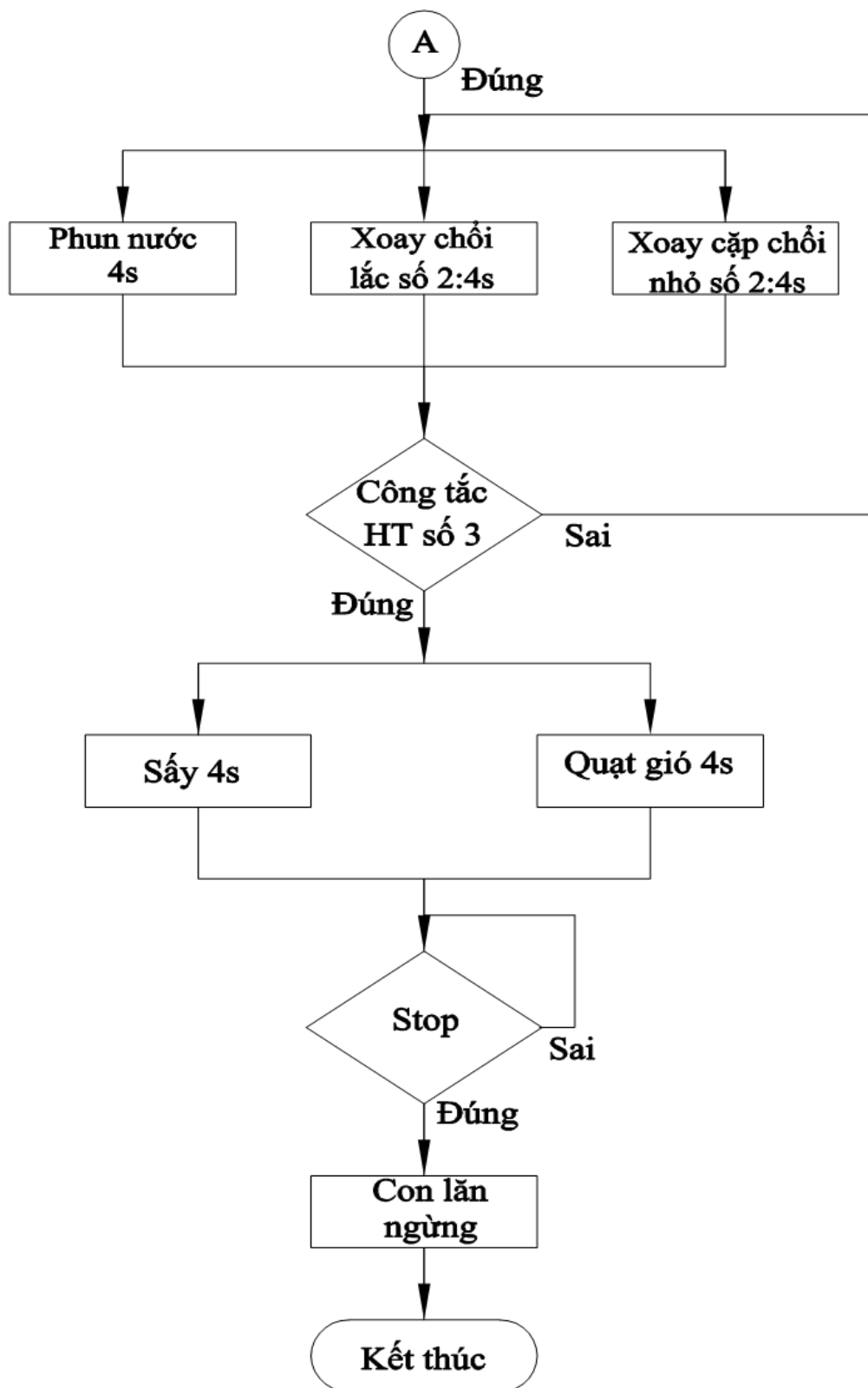
### **3.4.Lưu đồ thuật giải (hình 3.7.1 và 3.7.2)**

Dựa theo nguyên lý hoạt động của hệ thống như phần 3.1, ta xây dựng lưu đồ thuật giải.

Án start, động cơ chính được khởi động sẽ kéo hệ thống con lăn chạy và di chuyển xe. Khi xe đi qua sensor quang sẽ phát tín hiệu về cho Vi điều khiển (VĐK), VĐK sẽ xử lý và xuất xung cho động cơ phun nước và động cơ phun chất tẩy rửa hoạt động trong 4s. Khi xe đi qua công tắc hành trình (CTHT) số 1 thì CTHT sẽ gửi tín hiệu về VĐK, từ đó VĐK sẽ xử lý và xuất xung điều khiển điều khiển động cơ xoay các chổi. Tương tự như vậy khi xe qua CTHT số 2, động cơ xoay chổi và phun nước làm sạch hoạt động. CTHT số 3, máy sấy và quạt gió hoạt động. Sau 4s thì kết thúc quá trình và đưa tín hiệu cho động cơ chính dừng con lăn. Quá trình kết thúc.

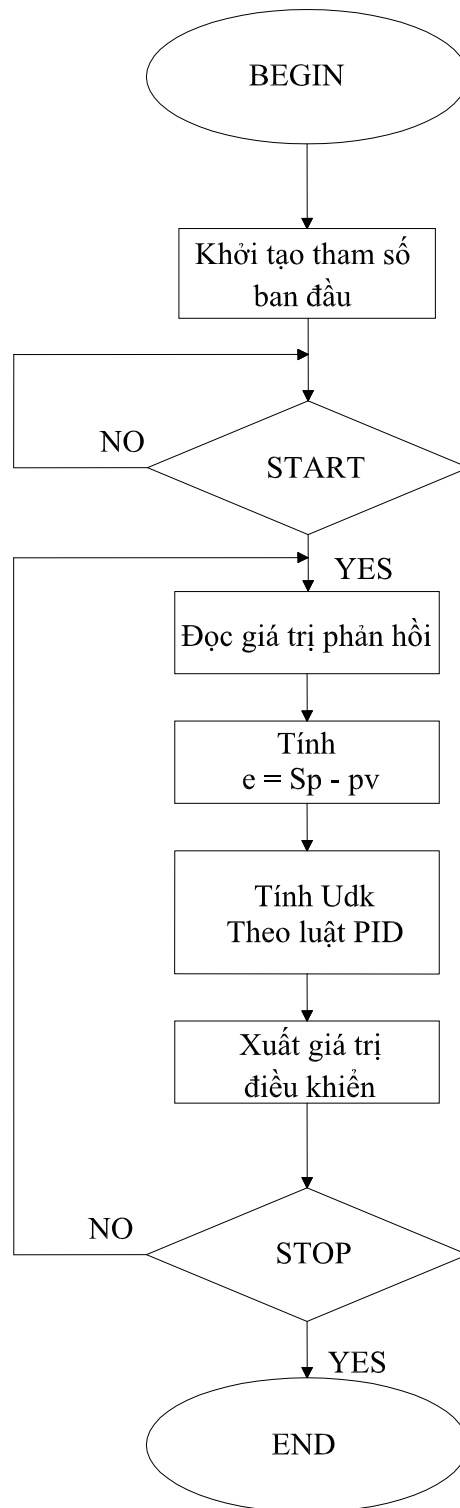


Hình 3.7.1: Lưu đồ thuật giải hoạt động của hệ thống.



*Hình 3.7.2:* Lưu đồ thuật giải hoạt động của hệ thống.

Lưu đồ thuật toán điều khiển tốc độ động cơ chính (hình 3.8) :



**Hình 3.8:** Sơ đồ thuật toán điều khiển động cơ chính.

Khi chương trình được bắt đầu ta “khởi tạo các tham số ban đầu” (*nhập giá trị đặt*) bằng cách nhập dữ liệu từ phím ấn. Sau đó nếu ta ấn nút “Start” động cơ sẽ hoạt động, lúc này bộ điều khiển sẽ “đọc giá trị phản hồi” (*giá trị thực của tốc độ động cơ*) và “tính  $e = Sp - pv$ ) sau đó nó sẽ “tính Uđk theo luật PID” và “xuất giá trị điều khiển” qua mạch công suất cấp điện cho động cơ. Nếu giá trị tốc độ động cơ vẫn sai lệch với giá trị đặt, hệ thống sẽ quay lại từ bước “đọc giá trị phản hồi” và cứ như vậy cho đến khi giá trị tốc độ động cơ bằng với giá trị đặt, lúc này động cơ sẽ quay với tốc độ ổn định bằng đúng giá trị đặt. Khi ta ấn nút “Stop” động cơ sẽ ngừng hoạt động.

### **3.5.Chương trình điều khiển**

```
#include <16F877A.h>

#device *=16 ADC=8

#FUSES HS, NOWDT, NOPROTECT, NOPUT

#use delay(clock=12000000)

#use i2c(master, sda=PIN_C4, scl=PIN_C3)

#use fast_io(b)

#use fast_io(d)

#use fast_io(e)

#define START input(pin_b1)

#define STOP input(pin_b2)

#define SENSOR input(pin_b3)

#define CT1 input(pin_b4)

#define CT2 input(pin_b5)

#define CT3 input(pin_b6)
```

```
#define RB7 input(pin_b7)

int16 a,b,j,k;

int8 ecd,necd;

int8 x;

int8 start_stop;

int1 clock;

int16 timer,timerdo;

int16 kt_sensor;

#INT_EXT          //ngat ngoai

void ngat_RB0()

{

    ecd=ecd+1;

}

#INT_RB

void ngat_RB7()

{

    if(RB7==0)

    {

        necd=ecd;

        ecd=0;

        clock=1;

    }

}
```



```
i2c_start();

i2c_write( 0x10);

i2c_write(x);////////DC

i2c_stop();

}

}

#int_timer1

void interrupt_timer0()

{

setup_timer_2(T2_DIV_BY_1,0, 1);

}

void khoitao()

{

output_low(pin_e0);

output_low(pin_e1);

output_d(0x38);

output_high(pin_e2);

output_low(pin_e2);
```

```
delay_us(450);  
  
output_d(0x0C);  
  
output_high(pin_e2);  
output_low(pin_e2);  
  
delay_us(50);  
  
}  
  
void hienthi()  
  
{  
  
a=a+48;  
  
output_high(pin_e0);  
output_low(pin_e1);  
output_d(a);  
output_high(pin_e2);  
output_low(pin_e2);  
  
delay_us(50);  
  
}  
  
void lenh()  
  
{  
  
output_low(pin_e0);  
output_low(pin_e1);  
output_d(b);
```

```
output_high(pin_e2);
```

```
output_low(pin_e2);
```

```
delay_us(50);
```

```
}
```

```
void kitu()
```

```
{
```

```
output_high(pin_e0);
```

```
output_low(pin_e1);
```

```
output_d(a);
```

```
output_high(pin_e2);
```

```
output_low(pin_e2);
```

```
delay_us(50);
```

```
}
```

```
void gia_tri_doc_ADC()
```

```
{
```

```
a=(x/1000);
```

```
if(a==1)
```

```
{
```

```
x=(x-1000);
```

```
}
```

```
hienthi();
```

```
a=(x/100);  
hienthi();  
j=((x/100)*100);  
k=((x/10)*10);  
a=(k-j);  
a=(a/10);  
hienthi();  
k=((x/10)*10);  
a=(x-k);  
hienthi();  
a=0x20;  
kitu();  
a=0x20;  
kitu();  
a=0x20;  
kitu();  
b=0x80; //quay lai  
lenh();  
}  
void main()  
{
```

```
set_tris_a(0b000001);
```

```
set_tris_b(0xFF);
```

```
set_tris_d(0);
```

```
set_tris_e(0);
```

```
output_float(pin_c3);
```

```
output_float(pin_c4);
```

```
enable_interrupts(GLOBAL); //cho phep tat ca cac ngat hoat dong
```

```
enable_interrupts(INT_EXT); //cho phep ngat ngoai hoat dong
```

```
enable_interrupts(INT_RB); //cho phep ngat RB hoat dong
```

```
ext_int_edge(H_to_L);
```

```
setup_timer_1(T1_INTERNAL|T1_DIV_BY_8); //Cai dat cho Time1
```

```
set_timer1(6250); //Dat thoi gian ngat cho time1
```

```
enable_interrupts(INT_TIMER1); //Cho phep time1 hoat dong
```

```
SETUP_ADC(ADC_CLOCK_INTERNAL); //cai dat cho module ADC hoat dong
```

```
SETUP_ADC_PORTS(AN0); //Su dung chan A0 lam dau vao analog bien doi  
ADC
```

```
delay_us(20); //Tao che cho module ADC thiet lap
```

```
setup_ccp1(CCP_PWM); //cai dat module PWM hoat dong
```

```
setup_timer_2(T2_DIV_BY_1,255, 1); //lay time 2 lam mau chuan tao xung
```



```
clock=0;

timer=0;

timerdo=0;

set_pwm1_duty(x); //tao xung voi duty=0

//////////////////////////////////////chuong trinh kiem tra o to di vao

if(SENSOR==1)

{

do

{

if(SENSOR==1)

{

kt_sensor=kt_sensor+1;

delay_ms(10);

}

else{ kt_sensor=0;}

}

while (kt_sensor<100);

//////////////////////////////////////

}

//////////////////////////////////////Chuong trinh cho bom

if(kt_sensor>98&&clock==1)
```

```
{  
  
clock=0;  
output_high(pin_d0); // Dong co bom 24V hoat dong  
output_high(pin_d1); //Dong co bom 12V hoat dong  
do  
{  
if(clock==1)  
{  
timer=timer+1;  
timerdo=timerdo+1;  
}  
if(timer==3)  
{  
output_low(pin_d0); // Dong co bom 24V dung  
output_low(pin_d1); // Dong co bom 12V dung  
timer=0;  
}  
}  
while(timerdo<3);  
timerdo=0;  
clock=0;
```



```
}
```

```
//////////////////////////////////////Chương trình cho công tác hành trình 1
```

```
while(clock!=1){; } //Cho đề bắt đầu
```

```
if(CT1==0&&clock==1)
```

```
{
```

```
output_high(pin_d2); //Động cơ chơi lạc 1 hoạt động
```

```
output_high(pin_d3); // Động cơ chơi xoay lớn hoạt động
```

```
output_high(pin_d4); //Động cơ chơi nhỏ 1 hoạt động
```

```
do
```

```
{
```

```
if(clock==1)
```

```
{
```

```
timer=timer+1;
```

```
timerdo=timerdo+1;
```

```
}
```

```
if(timer==3)
```

```
{
```

```
output_low(pin_d2); //Động cơ chơi lạc 1 dừng
```

```
output_low(pin_d3); // Động cơ chơi xoay lớn dừng
```

```
output_low(pin_d4); //Động cơ chơi nhỏ 1 dừng
```

```
timer=0;
```

```
}
```

```
}  
while(timerdo<3);  
timerdo=0;  
clock=0;  
}
```

```
//////////////////////////////////////Chuong trinh cho cong tac hanh trinh 2
```

```
while(clock!=1){;} //Cho de bat dau  
if(CT2==0&&clock==1)  
{  
output_high(pin_a3); //Dong co bom 220V hoat dong  
output_high(pin_a4); // Dong co choi lac 2 hoat dong  
output_high(pin_a5); //Dong co choi nho 2 hoat dong  
do  
{  
if(clock==1)  
{  
timer=timer+1;  
timerdo=timerdo+1;  
}  
if(timer==3)  
{
```

```

output_low(pin_a3); //Dong co bom 220V dung
output_low(pin_a4); // Dong co choi lac 2 dung
output_low(pin_a5); //Dong co choi nho 2 dung
timer=0;

}

}

while(timerdo<3);

timerdo=0;

clock=0;

}

```

//////////////////////////////////////Chuong trinh cho cong tac hanh trinh 3

```

while(clock!=1){; } //Cho de bat dau

if(CT3==0&&clock==1)

{

output_high(pin_d5); //Dong co quat 12V hoat dong

output_high(pin_d6); // May say hoat dong

do

{

if(clock==1)

{

timer=timer+1;

```

```
timerdo=timerdo+1;

}

if(timer==3)

{

output_low(pin_d5); //Dong co quat 12V dung

output_low(pin_d6); // May say dung

timer=0;

}

}

while(timerdo<3);

timerdo=0;

clock=0;

}

////////////////////////////////////

////////////////////////////////////

}

////////////////////////////////////

}

}
```

```
#include <16F877A.h>

#device *=16

#FUSES NOWDT, HS, NOPUT, NOPROTECT, NODEBUG, BROWNOUT,
LVP, NOCPD, NOWRT //No Watch Dog Timer

#use delay(clock=12000000)

#define device_SDA PIN_C4

#define device_SCL PIN_C5

#use i2c(SLAVE, SDA=PIN_C4, SCL=PIN_C3, address=0x10)

#use fast_io(b)

#use fast_io(d)

#use fast_io(e)

int8 state, address;

int8 x1;

int16 ecd;

int16 a,b,j,k;

#INT_SSP

void ssp_interrupt ()

{
```

```
state = i2c_isr_state();

if(state == 0)
{
    address = i2c_read();

}

if(state < 0x80)
{
    if(state==1)
    {
        x1 = i2c_read();

    }

    if(state==2)
    {
        ecd = i2c_read();

    }

}
```

```
}  
  
void khoitao()  
{  
output_low(pin_e0);  
output_low(pin_e1);  
output_d(0x38);  
//delay_ms(1);  
output_high(pin_e2);  
delay_us(5);  
output_low(pin_e2);  
//delay_ms(1);  
delay_us(100);  
output_d(0x0C);  
output_high(pin_e2);  
delay_us(5);  
output_low(pin_e2);  
//delay_ms(1);  
delay_us(100);  
}  
  
void hienthi()
```

```
{  
a=a+48;  
output_high(pin_e0);  
output_low(pin_e1);  
output_d(a);  
delay_us(5);  
output_high(pin_e2);  
delay_us(5);  
output_low(pin_e2);  
delay_us(100);  
}
```

```
void lenh()
```

```
{  
output_low(pin_e0);  
output_low(pin_e1);  
output_d(b);  
delay_us(5);  
output_high(pin_e2);  
delay_us(5);  
output_low(pin_e2);  
delay_us(100);
```



```
}  
  
void kitu()  
{  
    output_high(pin_e0);  
    output_low(pin_e1);  
    output_d(a);  
    delay_us(5);  
    output_high(pin_e2);  
    delay_us(5);  
    output_low(pin_e2);  
    delay_us(100);  
}  
  
void htthongso()  
{  
  
    ////////////////////////////////////////////toc do dc 1  
  
    a=(x1/100);  
  
    hienthi();  
  
    j=((x1/100)*100);  
    k=((x1/10)*10);  
    a=(k-j);  
}
```

```
a=(a/10);
```

```
hienthi();
```

```
k=((x1/10)*10);
```

```
a=(x1-k);
```

```
hienthi();
```

```
a=0x20;
```

```
kitu();
```

```
////////////////////////////////////
```

```
////////////////////////////////////
```

```
b=0xC0; //xuong dong 2
```

```
lenh();
```

```
////////////////////////////////////nhiet do dong co 1
```

```
a=(ecd/100);
```

```
hienthi();
```

```
j=((ecd/100)*100);
```

```
k=((ecd/10)*10);
```

```
a=(k-j);
```

```
a=(a/10);
```

```
hienthi();
```

```
k=((ecd/10)*10);
```

```
a=(ecd-k);
```

```
hienthi();
```

```
a=0x20;

kitu();

b=0x80; //quay lai dong 1

lenh();

}

void main()

{

set_tris_b(0);

set_tris_d(0);

set_tris_e(0);

output_float(pin_c4);

output_float(pin_c5);

enable_interrupts(int_SSP);

enable_interrupts(global);

khoitao();

While(true)

{

htthongso();

}

}
```

## KẾT LUẬN

Sau thời gian tìm hiểu và thực nghiệm cứu mô hình nhà rửa xe tự động, em đã hoàn thành đề tài tốt nghiệp: ***“Xây dựng mô hình nhà rửa xe tự động trên nền vi điều khiển họ PIC”***, nhưng chưa có điều kiện để thực hiện mô hình.

Trong cuốn đồ án này em đã thực hiện được những nội dung sau:

- Tìm hiểu quy trình rửa xe tự động.
- Tìm hiểu phương pháp lựa chọn thiết bị cho hệ thống.
- Đã thiết kế điều khiển cho hệ thống rửa xe tự động trên nền vi điều khiển họ PIC.

Khi tìm hiểu thực tế và thực hiện đồ án, em thấy hệ thống này có tính khả thi cao, có khả năng ứng dụng trong thực tế.

Trong quá trình thực hiện đồ án với sự nỗ lực của bản thân và đặc biệt là sự hướng dẫn, chỉ bảo tận tình của thầy giáo hướng dẫn Th.S Nguyễn Trọng Thắng cùng sự giúp đỡ của các thầy cô giáo trong tổ bộ môn mà đồ án tốt nghiệp của em đã hoàn thành theo đúng yêu cầu.

Mặc dù đã cố gắng hết sức nhưng do trình độ còn hạn chế, thời gian làm đồ án có hạn, hơn nữa đây cũng là đề tài thực tế còn khá mới mẻ ở nước ta... vậy nên đồ án của em còn rất nhiều thiếu sót và nhiều phần chưa tìm hiểu sâu. Rất mong sự chỉ dạy và góp ý của các thầy cô và các bạn để bản đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

### **\* Hướng mở rộng của đề tài**

Do thời gian thực hiện có hạn, trình độ còn hạn chế và thiếu kinh nghiệm thực tế vậy nên đề án của em mới chỉ dừng lại ở việc thiết kế hệ thống cho các xe cỡ nhỏ (từ 4- 16 chỗ), hệ thống cấp nước còn sơ sài, các chế độ hoạt động vẫn còn hạn chế và tính thẩm mỹ chưa cao. Trong thời gian tới em hi vọng các sinh viên khóa sau sẽ mở rộng được mô hình với những tính năng tiên tiến hơn :

- Mô hình nghiên cứu có thể làm việc với những xe có kích cỡ khác nhau và giải pháp công nghệ toàn diện hơn.
- Xây dựng được hệ thống cấp nước tiên tiến hơn: với hệ thống xử lý nước thải, tiết kiệm nguồn nước...

...

## TÀI LIỆU THAM KHẢO

[1] Phạm Công Ngô : “ Lý thuyết điều khiển tự động ” – NXB Khoa học và kỹ thuật – 1996.

[2] Phạm Văn Át : “ Kỹ thuật lập trình C - cơ sở và nâng cao ” – NXB Khoa học và kỹ thuật – 1999.

[3] Nguyễn Tăng Cường – Phan Quốc Thắng : “ Cấu trúc và lập trình họ vi điều khiển 8051 ” – NXB Khoa học và kỹ thuật – 2004

Website:

<http://www.picvietnam.com/forum/forumdisplay.php?f=8>

<http://webdien.com/d/index.php>

<http://www.ebook.edu.vn/?page=1.5>

<http://tailieu.vn/tag/tai-lieu/lap-trinh-C-can-ban.html>