

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HCM**  
**KHOA CÔNG NGHỆ ĐIỆN TỬ**



**ĐỒ ÁN 2**

**THIẾT KẾ BỘ ĐIỀU KHIỂN**  
**THANG MÁY 4 TẦNG DÙNG PLC**

Giảng viên hướng dẫn : **Ths. VŨ THỊ HỒNG NGÀ**

Sinh viên thực hiện: **NÌM TRÍ MINH -07730171**

**VŨ CÔNG NGUYỄN -07733571**

**VŨ QUANG TÙNG -07732331**

Lớp: **CDDT9A**

Niên khoá: **2007 - 2010**

**TP. Hồ Chí Minh, tháng 01 năm 2010**

## **Chương 1: TỔNG QUAN VỀ THANG MÁY**

### **1.1 Khái niệm chung về thang máy**

Thang máy là một thiết bị chuyên dùng để vận chuyển người, hàng hóa, vật liệu, v.v... theo phương thẳng đứng hoặc nghiêng một góc nhỏ hơn 150 so với phương thẳng đứng theo một tuyến đã định sẵn.

Thang máy thường được dùng trong các khách sạn, công sở, chung cư, bệnh viện, các đài quan sát, tháp truyền hình, trong các nhà máy, công xưởng v.v... Đặc điểm vận chuyển bằng thang máy so với các phương tiện vận chuyển khác là thời gian của một chu kỳ vận chuyển bé, tần suất vận chuyển lớn, đóng mở máy liên tục. Ngoài ý nghĩa về vận chuyển, thang máy còn là một trong những yếu tố làm tăng vẻ đẹp và tiện nghi công trình.

Thang máy là một thiết bị vận chuyển đòi hỏi tính an toàn nghiêm ngặt, nó liên quan trực tiếp đến tài sản và tính mạng con người. Vì vậy, yêu cầu chung đối với thang máy khi thiết kế, chế tạo, lắp đặt, vận hành, sử dụng và sửa chữa là phải tuân thủ một cách nghiêm ngặt các yêu cầu về kỹ thuật an toàn được quy định trong các tiêu chuẩn quy trình, quy phạm.

### **1.2 Phân loại thang máy**

#### *1.2.1. Phân loại theo chức năng:*

- Thang máy chuyên chở người.
- Thang máy chuyên chở hàng nhưng có người đi kèm.
- Thang máy chuyên chở người nhưng có hàng đi kèm.
- Thang máy bệnh viện.
- Thang máy chuyên chở hàng không có người đi kèm.

#### *1.2.2. Phân loại theo hệ thống dẫn động:*

- Thang máy dẫn động điện.
- Thang máy thủy lực.
- Thang máy khí nén.

### 1.2.3. Phân loại theo hệ thống điều khiển:

- Điều khiển bằng relay.
- Điều khiển bằng PLC.
- Điều khiển bằng máy tính.

### 1.2.4. Phân loại theo trọng tải:

- Thang máy loại nhỏ  $Q < 160$  kg.
- Thang máy trung bình  $Q = 500-2000$  kg.
- Thang máy loại lớn  $Q > 2000$  kg.

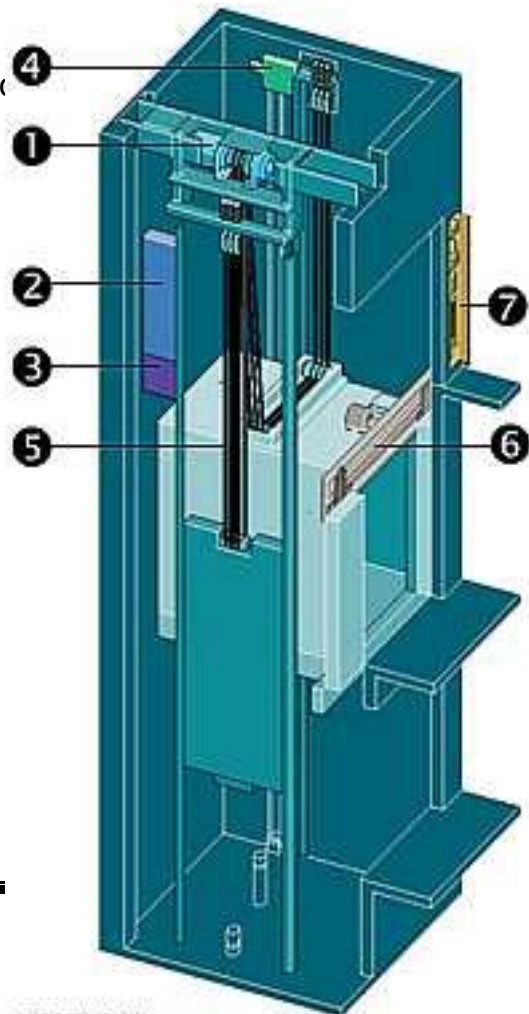
### 1.2.5. Phân loại theo độ dịch chuyển:

- Thang máy chạy chậm  $V = 0,5$  m/s.
- Thang máy tốc độ trung bình  $V = 0,75 - 1,5$  m/s.
- Thang máy cao tốc  $V = 2,5 - 5$  m/s.

## 1.3 Cấu tạo chung

➤ Cấu tạo: Thang máy có nhiều loại, nhưng cấu tạo chung đều gồm có các bộ phận chính như sau:

- Nguồn
- Motor kéo
- Dây cáp
- Thanh ray
- Cabin
- Đối trọng
- Bộ phận lò xo giảm xóc
- Nút nhấn gọi thang bên ngoài



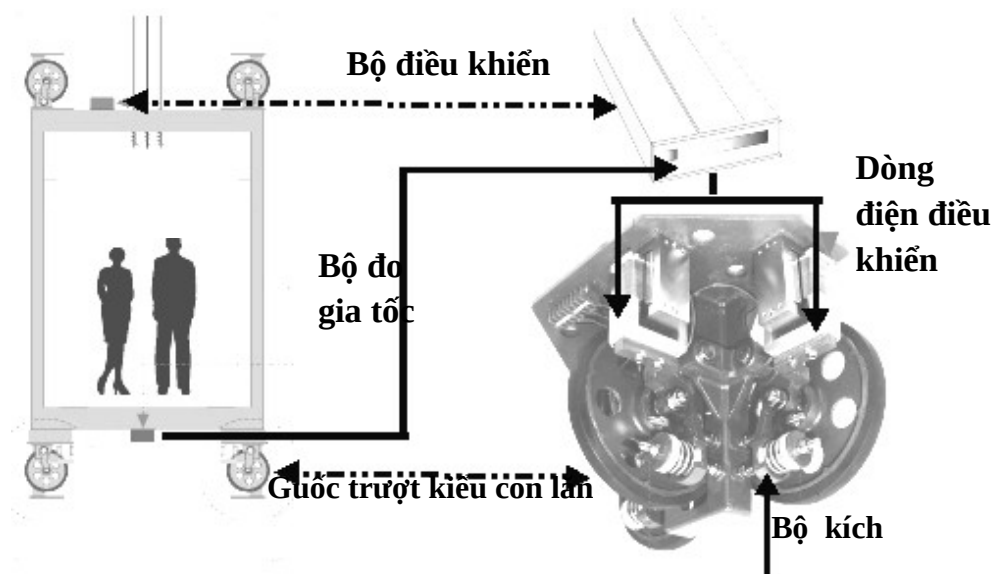
- Nút nhấn gọi thang bên trong

Hình 1.1. Cấu tạo chung của thang máy

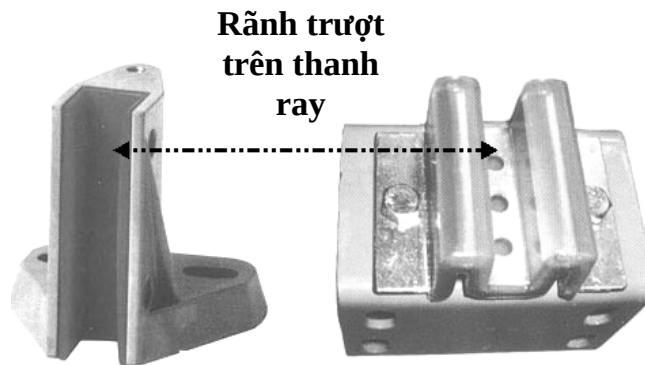
➤ Cabin trong đó có chứa người hoặc hàng hóa, chuyển động trên ray dẫn hướng thẳng đứng. Cáp nâng trên đó có treo cabin và đối trọng. Trọng lượng thang máy và trọng lượng vật nâng được cân bằng bởi đối trọng treo trên các dây cáp đi ra từ pulley dẫn cáp. Buồng thang máy và đối trọng khi di chuyển sẽ trượt trên thanh ray dẫn hướng.



Hình 1.2. Một số dạng cabin thang máy



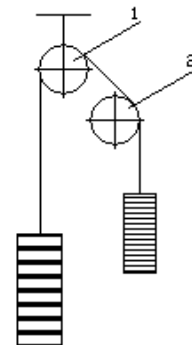
Hình 1.3. Biên dạng guốc trượt kiểu lăn của hãng MITSUBISHI



Hình 1.4. Biên dạng guốc kiểu trượt của hãng NINGBO XINGDA

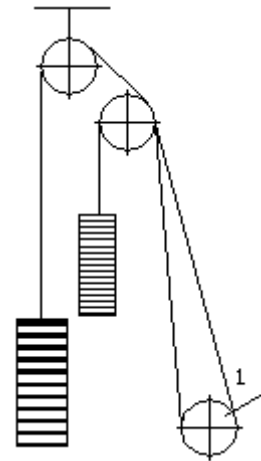
- Một số sơ đồ thang máy thường gặp:

+ Thang máy có puli dẫn hướng: Có lắp thêm puli phụ (2) để dẫn hướng cáp đối trọng. Sơ đồ này thường được dùng khi kích thước cabin lớn, cáp đối trọng không thể dẫn hướng từ puli dẫn cáp (hoặc tang) một cách trực tiếp xuống dưới.



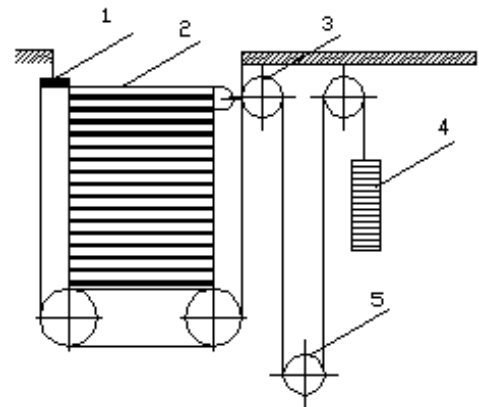
Hình 1.5a

+ Thang máy có sự bố trí bộ tời bên dưới có bộ tời (1) được bố trí ở phần bên hông hoặc phần dưới của đáy giếng, nhờ đó có thể làm giảm tiếng ồn của thang máy khi làm việc. Dùng sơ đồ này sẽ làm tăng tải trọng tác dụng lên giếng thang, cũng như tăng chiều dài và số điểm uốn của cáp nâng, dẫn đến tăng độ mòn của cáp nâng. Kiểu bố trí bộ tời như thế này chỉ sử dụng trong trường hợp đặc biệt khi mà buồng máy không thể bố trí được phía trên giếng thang và khi có yêu cầu cao về giảm độ ồn khi thang máy làm việc.



Hình 1.5b

+ Thang máy kiểu đẩy: cáp nâng (1) tên đó có tời cabin (2), được uốn qua các puli (6) lắp trên khung cabin, sau đó đi qua puli phía trên (3) đến puli dẫn cáp (5) dẫn cáp (5) của bộ tời nâng Trọng lượng của cabin và một phần vật nâng được cân bằng bởi đối trọng(4). Các dây cáp của đối trọng uốn qua puli dẫn hướng phụ.



Hình 1.5c

Hình 1.5 a,b,c. sơ đồ một loại thang máy.

#### 1.4 Nguyên lý hoạt động và sử dụng thang máy

Thang máy hoạt động theo các nguyên tắc sau:

##### 1.4.1 Reset buồng thang khi đóng nguồn:

Dù cho buồng thang đang ở bất kỳ vị trí hoặc trạng thái nào, thì khi đóng nguồn đều được reset và đưa về tầng trệt.

##### 1.4.2 Nguyên tắc di chuyển lên xuống, đóng và mở cửa.

- \_ Buồng thang chỉ hoạt động khi cửa đã hoàn toàn đóng.
- \_ Cửa chỉ mở khi buồng thang dừng đúng tầng.
- \_ Cửa sẽ tự động mở hoặc đóng sau khi nhận được các yêu cầu.

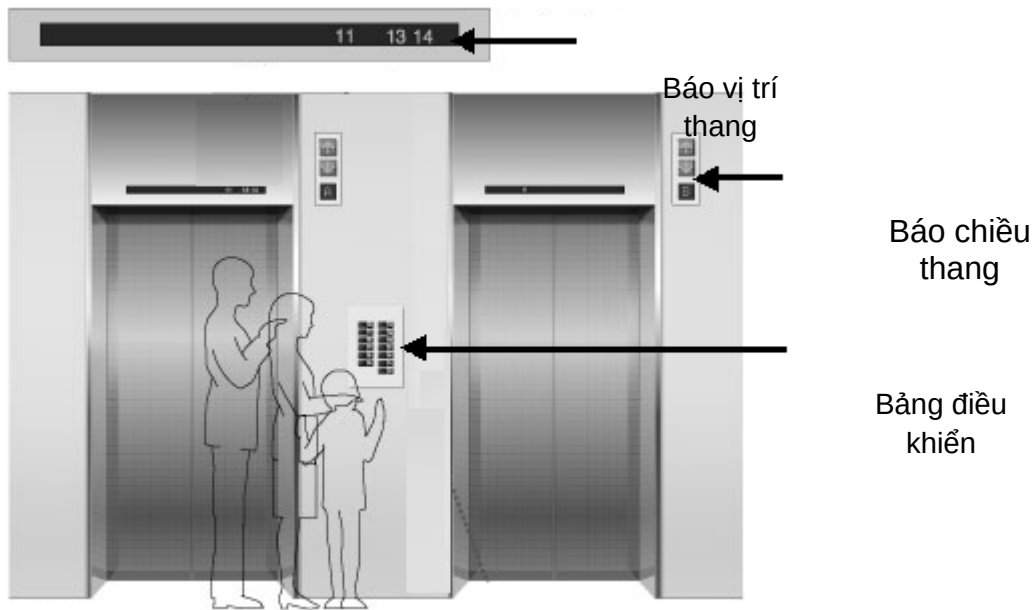
\_ Cửa buồng thang sẽ ở chế độ mở thường trực khi thang không hoạt động.

#### 1.4.3 Nguyên tắc đến tầng:

Để xác định vị trí hiện tại của thang nhờ cảm biến ở mỗi cửa tầng. Khi buồng thang ở tầng nào thì cảm biến nhận tín hiệu ở tầng đó và đưa về điều khiển.



#### 1.4.4 Sử dụng thang máy:

- Gọi thang máy từ bên ngoài buồng thang (ở các tầng)



Hình 1.6. Mô hình điều khiển thang máy từ bên ngoài buồng thang

\_ Gọi thang: ở mỗi tầng mà thang phục vụ, gần ngay cửa tầng đều có bảng điều khiển (Hall Call Panell), còn gọi là hộp Button tầng mục đích phục vụ cho việc gọi thang bao gồm:

+ Hai nút ấn: Một nút để gọi cho thang đi lên , một nút để gọi thang đi xuống . Riêng ở tầng dưới cùng chỉ có một nút (là đi lên hoặc đi xuống).

+ Đèn báo tầng và báo chiều cho biết vị trí và chiều hoạt động hiện của cabin thang máy. Khi muốn gọi thang, hành khách chỉ cần ấn vào nút gọi tầng theo chiều muốn đi, tín hiệu đèn sẽ sáng lên, đèn báo hiệu hệ thống đã ghi nhận lệnh gọi.

\_ Đáp ứng của thang sau lệnh gọi: Nếu buồng thang đang ở một vị trí nào đó khác với tầng mà hành khách vừa gọi, thang sẽ di chuyển đến tầng đó theo thứ tự ưu tiên như sau :

+ Nếu thang di chuyển cùng chiều với lệnh gọi thang và di chuyển ngang qua tầng mà hành khách khách đang đứng gọi, thì khi đến tầng được gọi, thang sẽ dừng lại và đón khách.

+ Nếu thang đang di chuyển theo chiều ngược với chiều hành khách muốn đi, hoặc cùng chiều nhưng không đi ngang qua, thì sau khi đáp ứng hết các nhu cầu của chiều đó, thang sẽ quay trở lại đón khách.

+ Nếu buồng thang đang ở ngay tại tầng mà hành khách vừa gọi, buồng thang sẽ mở cửa đón khách.

➤ Gọi thang từ bên trong buồng thang: Trong buồng thang có bảng điều khiển phục vụ cho việc đi thang của khách (Car Operating Panel) còn gọi là hộp Button Car. Bao gồm các nút có chức năng sau:





Hình 1.7. Bảng điều khiển bên trong thang máy

+ Các nút mang số : Đại diện cho các tầng mà thang phục vụ.





+ Nút  (**DO** – Door Open): Dùng để mở cửa (chỉ có tác dụng khi thang dừng tại tầng).

+ Nút  (**DC** – Door Close): Dùng để đóng cửa (chỉ có tác dụng khi thang dừng tại tầng).

+ Nút **Interphone** hoặc **Alarm** : Dùng để liên lạc với bên ngoài khi thang gặp các sự cố về điện, hoặc đứt cáp treo.

+ Công tắc E.Stop (Emergency Stop) nếu có: Để dừng thang khẩn cấp khi có sự cố xảy ra.

\_ Khi đã vào bên trong buồng thang, muốn đến tầng nào, khách ấn nút chỉ định tầng đó, thang máy sẽ lập tức di chuyển và tuần tự dừng tại các tầng mà nó đi qua. Cửa buồng thang và cửa tầng được thiết kế đóng mở tự động. Khi buồng thang di chuyển đến một tầng nào đó, sau khi ngừng hẳn, cửa buồng thang và cửa tầng sẽ tự động mở để khách có thể ra (vào) buồng thang, sau vài giây cửa sẽ tự động đóng lại.

\_ Sau đó thang máy sẽ thực hiện lệnh tiếp theo. Nếu không muốn chờ hết khoảng thời gian cửa đóng lại, khách có thể ấn nút DC để đóng cửa buồng thang. Trong trường hợp khẩn cấp muốn dừng thang, khách có thể ấn nút E.Stop (nếu có) trên bảng điều khiển trong buồng thang. Khi có sự cố mất điện, khách ấn vào nút Interphone hoặc Alarm để yêu cầu giúp đỡ từ bên ngoài.

### 1.5 Các thông số của thang máy

Các thông số này bao gồm:

+ Tải trọng định mức: Được xác định theo khối lượng tính toán lớn nhất mà thang máy có thể vận chuyển được không kể đến khối lượng của buồng thang và các thiết bị bố trí trong đó.

+ Tốc độ định mức: Là tốc độ chuyển động của buồng thang theo tính toán thiết kế. Trong thực tế vận hành tốc độ có thể sai lệch khoảng 10%.

+ Chiều cao nâng, hạ.

+ Năng suất của thang máy: Là lượng người hay số lượng hàng hóa mà thang máy có thể vận chuyển được trong một giờ theo một hướng. Công suất của

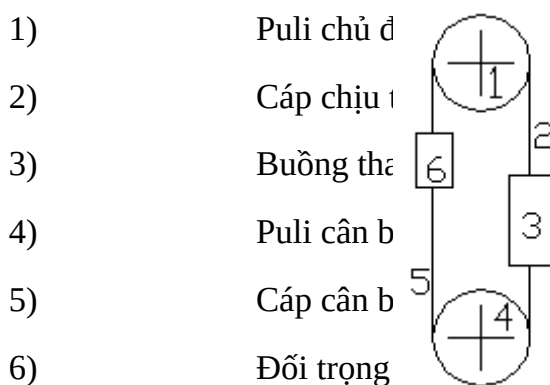
thang máy có thể tính theo công thức:

$$N = \frac{3600 \cdot E \cdot \gamma}{2 \frac{H}{V} + \sum t_i}$$

Trong đó:

- N: công suất thang máy
- $\gamma$ : hệ số mang tải của buồng thang
- E: sức chứa tính toán định mức của buồng thang
- H: chiều cao nâng – hạ
- V: tốc độ của buồng thang (m/s)
- $\sum t_i$ : thời gian tổn cộng để đóng – mở cửa buồng thang, thời gian ra vào của hành khách, thời gian mở máy và hãm máy.

**Việc sử dụng đối trọng và cáp cân bằng là để giảm phụ tải của cơ cấu, tức là độ mất cân bằng khi nâng hoặc hạ buồng thang đến các vị trí biên, do đó giảm được cơ cấu truyền động**



Hình 1.8. Sơ đồ thang máy có cáp cân bằng.

\_ Nếu không có cáp cân bằng, lực tác động lên puli chủ động theo hai nhánh của dây cáp sẽ là:

$$F_1 = G_0 + G - g_c \cdot x \quad (N)$$

$$F_2 = G_{dt} - g_c.(H - x) \quad (N)$$

Trong đó:

- $G_0$ : trọng lượng buồng thang (N)
- $G$ : trọng lượng tải trọng (N)
- $G_{dt}$ : trọng lượng đối trọng (N)
- $g_0$ : trọng lượng của 1 đơn vị dài dây cáp (N)
- $H$ : chiều cao nâng hạ (m)
- $x$ : khoảng cách từ buồng thang đến puli chủ động (m)

\_ Khi đó lực tác động lên puli chủ động khi nâng hạ tải là:

$$F_n = F_1 - F_2 = G_0 + G - g_c.x - [G_{dt} - g_c.(H - x)] = G_0 + G - g_c.(2.x - H - x) - G_{dt}$$

$$F_h = F_2 - F_1 = G_{dt} + g_c.(H - x) - (G_0 + G - g_c.x) = G_{dt} + g_c.(H - 2.x) - G$$

\_ Từ hai biểu thức trên ta thấy lực tác dụng lên puli phụ thuộc vào khoảng cách  $x$ , nếu khoảng cách này là cực đại thì sẽ gây ra phụ tải cực đại cho động cơ, nếu khoảng cách này là cực tiểu thì gây ra non tải cho động cơ. Điều này không có lợi cho động cơ và cho toàn bộ cơ cấu. Do đó việc sử dụng cáp cân bằng là để khắc phục nhược điểm này, cáp cân bằng có thể chọn cùng loại với cáp nâng hạ. Khi sử dụng cáp cân bằng, các thành phần liên quan đến  $x$  trong biểu thức trên sẽ bị triệt tiêu:

$$F_n = F_1 - F_2 = G_0 + G - G_{dt}$$

$$F_h = F_2 - F_1 = G_{dt} - (G_0 + G)$$

\_ Việc chọn khối lượng cho đối trọng nhằm mục đích cân bằng để đảm bảo có thể chọn động cơ có công suất nhỏ nhất:

$$G_{dt} = G_0 + a.G_{dm}$$

Trong đó:

- $A$ : là hệ số cân bằng ( $a=0, 34 \div 0, 6$ )
- $G_{dm}$ : là trọng lượng định mức

\_ Từ khối lượng đối trọng, tính được lực tác động lên puli chủ động khi nâng và hạ như sau:

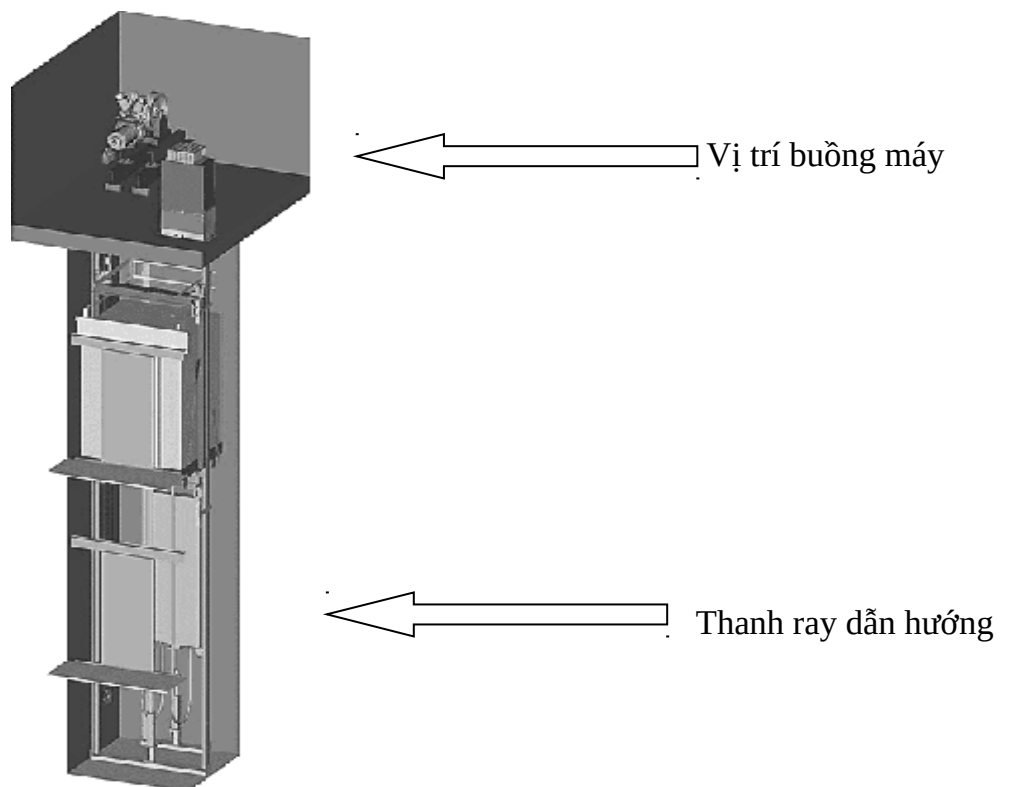
$$F_n = G + a.G_{dm}$$

$$F_h = -G + a.G_{dm}$$

## 1.6 Các yêu cầu đối với thang máy

### 1.6.1 Vị trí buồng máy

Vị trí buồng máy có thể đặt bên trên hoặc bên dưới đường hầm tùy theo yêu cầu và diện tích cho phép của buồng máy.



Hình 1.9. Mô hình giếng thang với buồng máy được đặt bên trên.

### 1.6.2 Thanh ray dẫn hướng

Trong khi chuyển động, buồng thang và đối trọng sẽ trượt dọc trên thanh ray dẫn hướng. Ray dẫn hướng đảm bảo cho cabin và đối trọng luôn nằm và chuyển động theo đúng vị trí đã được thiết kế trong giếng thang, không cho chúng dịch

chuyển tho phương ngang trong quá trình dịch chuyển. Ray dẫn hướng được lắp đặt ở hai bên cabin và đối trọng với độ chính xác theo yêu cầu cần thiết (đòi hỏi độ chính xác về độ thẳng đứng của ray, khoảng cách các đầu ray...).

### 1.6.3 Công tắc hành trình

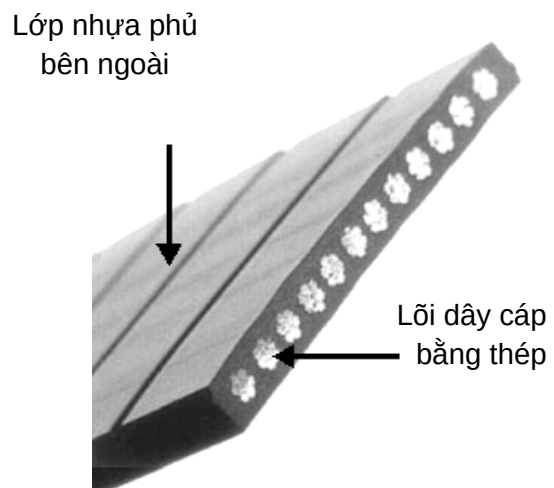
Để đảm bảo an toàn cho người sử dụng và các thiết bị trong mạch điều khiển, người ta bố trí các thiết bị bảo vệ liên động, các tiếp điểm hành trình để đảm bảo cho thang máy dừng chính xác, không vượt khỏi phạm vi giới hạn (các loại công tắc hạn chế hành trình trên, hạn chế hành trình dưới, công tắc chuyển đổi tầng, công tắc đến tầng...).

### 1.6.4 Cáp nâng cabin và đối trọng

Phải đảm bảo chịu lực nâng và lực ma sát với puli theo đúng tiêu chuẩn an toàn cho phép trong lắp đặt thang máy. Có thể dùng cáp thép hoặc cáp thép có phủ nhựa bên ngoài để kéo cabin thang máy.

\_ Cáp thép phủ nhựa có sự linh hoạt và khả năng kéo tải tốt hơn so với loại cáp thép thông thường.

\_ Đối với loại cáp thép truyền thống, sự hao mòn gây ra là bởi nhiều yếu tố, đó là ảnh hưởng của sự mài mòn của các sợi cáp khi chúng bị chèn vào bên trong và bị kéo ra khỏi rãnh kéo, do có sự bám bụi trên sợi cáp nên càng làm tăng thêm sự mài mòn sợi cáp, giảm thời gian sử dụng của cáp rất đáng kể.



Hình 1.10. Cáp thép phủ nhựa của hãng OTIS

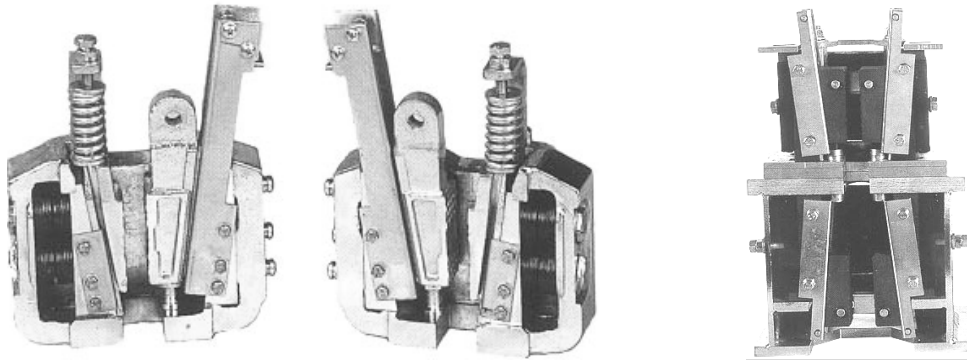
\_ Đối với loại cáp thép phủ nhựa, nhờ có lớp nhựa nên nó bám chặt bánh đà, tạo nên sự ma sát thích hợp, không có sự mài mòn nào gây ra thêm giữa các rãnh, các sợi cáp thép được phủ nhựa nên tránh được bụi bám, nhờ đó tránh bị hao mòn. Tuy nhiên sự giảm khả năng chịu lực của dây thép theo thời gian sử dụng vẫn xảy ra, nhưng ta có thể biết trước được sự giảm tuổi thọ của cáp nhờ vào tính toán và do nhà sản xuất cung cấp.

### 1.6.5 Hệ thống phanh bảo hiểm

Buồng thang còn được trang bị thêm các bộ phận phanh bảo vệ phòng khi cáp treo bị đứt, bị mất điện, khi tốc độ buồng thang vượt quá  $20\% \div 40\%$  tốc độ định mức, phanh sẽ tác động. Thường có 3 loại phanh:

- Phanh kiểu nêm.
- Phanh kiểu lệch tâm.
- Phanh bảo hiểm kiểu kìm.

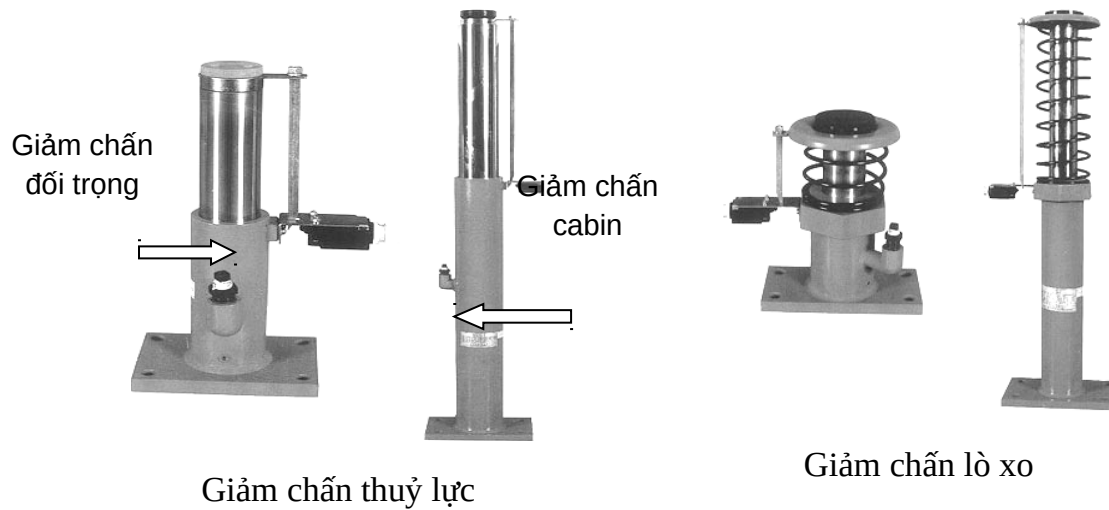
Trong đó, phanh bảo hiểm kiểu kìm được sử dụng rộng rãi hơn, nó bảo đảm cho buồng thang dừng tốt hơn so với các loại phanh khác. Phanh bảo hiểm thường được đặt phía dưới buồng thang, có gọng kìm trượt theo thanh dẫn hướng.



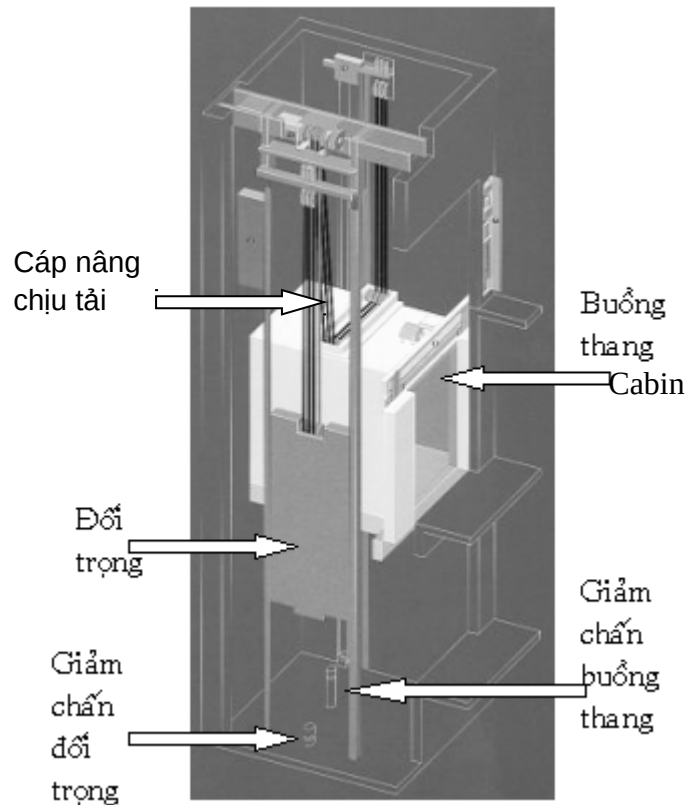
Hình 1.11. Phanh bảo hiểm kiểu kìm.

### 1.6.6 Bộ giảm chấn

Dưới đáy giếng có bố trí thêm các bộ giảm chấn nhằm tránh hiện tượng va đập quá mạnh khi công tắc hạn chế hành trình không tác động, hoặc khi thang bị đứt cáp treo..., dùng để chống sóc hoặc va chạm mạnh gây ảnh hưởng đến an toàn cho hành khách đang sử dụng thang máy, đồng thời tránh hư hỏng cho cabin và đối trọng thang máy.



Hình 1.12. Một số loại giảm chấn.



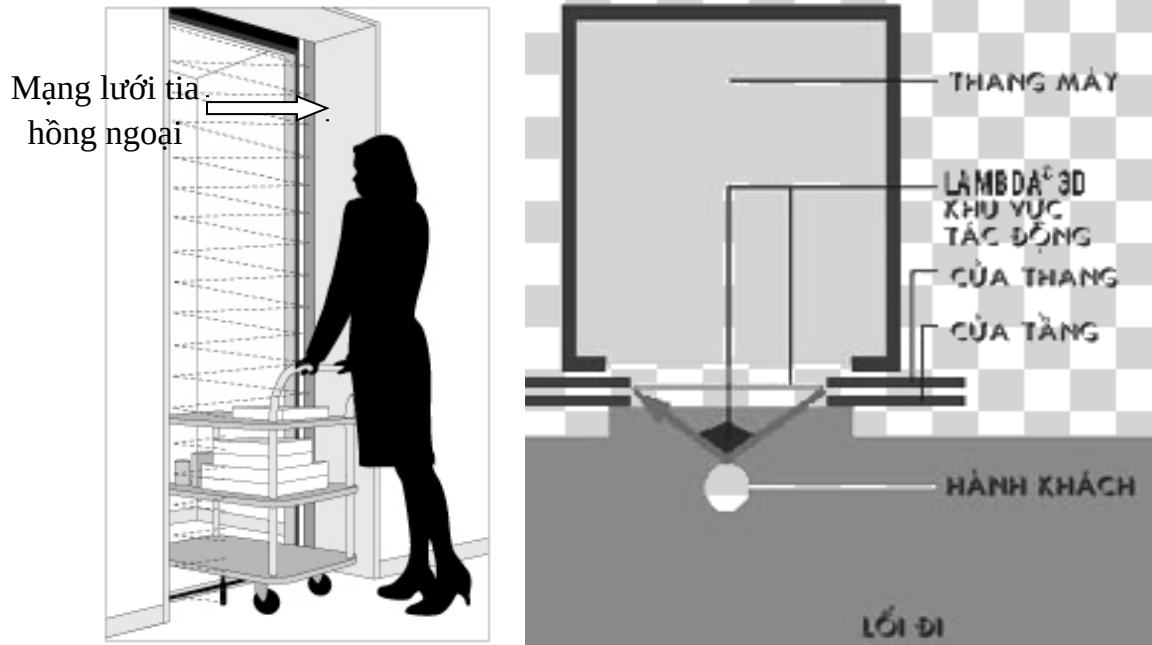
Hình 1.13. Vị trí lắp đặt hệ thống giảm chấn trong giếng thang

Chuyển động của buồng thang phải êm, không gây sốc, gây cảm giác khó chịu cho hành khách. Phải dừng chính xác đến tầng để không gây nguy hiểm và trở ngại cho hành khách khi ra vào buồng thang

#### 1.6.7 Hệ thống cảm biến cửa

Hệ thống cảm biến cửa là mạng lưới tia hồng ngoại bao phủ ngay vị trí cửa ra vào cabin, điều khiển hoạt động của cửa nhằm bảo vệ an toàn cho hành khách và

hàng hóa khi ra vào buồng thang. Ngoài ra nó còn làm giảm sự hư hỏng của thang trong trường hợp di vận chuyển vật nặng hoặc di chuyển ra vào chậm. Tăng cường khả năng tin cậy của hệ thống.

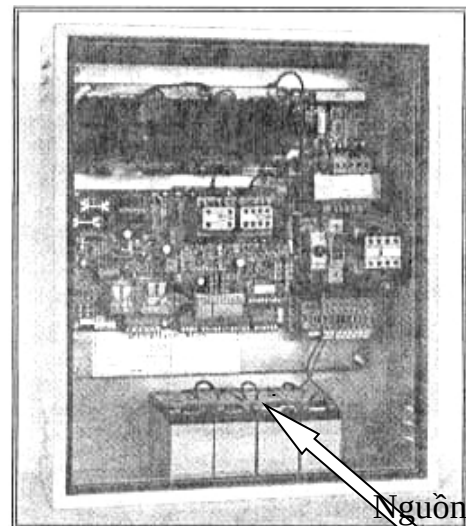


Hình 1.14. Mô hình hệ thống cảm biến cửa.

\_ Đặc tính: Hệ thống cảm biến cửa sử dụng thiết bị thu và phát tia hồng ngoại tạo ra một mạng lưới cắt ngang khung cửa, hệ thống quét liên tục để phát hiện bất cứ tia hồng ngoại nào bị gián đoạn, nếu có, hệ thống sẽ mở cửa ngay lập tức và không gây va chạm cho hành khách (hoặc hàng hóa) với cửa.

#### 1.6.8 Hệ thống tự động bảo vệ bằng điện (Automatic Rescue Divide):

Khi thang máy có sự cố hoặc gặp lỗi không mong muốn, hành khách có thể bị mắc kẹt bên trong buồng thang. Khi đó thiết bị bảo vệ tự động sẽ tác động ngay lập tức, nó được cấp nguồn từ nguồn điện dự trữ (hệ thống acqui, pin ...), buồng thang khi đó sẽ được điều khiển đưa đến tầng gần nhất và hệ thống cửa sẽ được tự động mở ra.\_ Lĩnh vực



Nguồn acqui tự  
cấp



ứng dụng: Bộ ARD được dùng vận hành cho trường hợp khẩn cấp cần bảo vệ tự động cho thang máy, được kết nối với hộp số thang máy (dùng nguồn 3 pha AC), cùng các bộ phanh (dùng nguồn DC). Tùy theo yêu cầu,

*Hình1.15. Tủ điện ARD*

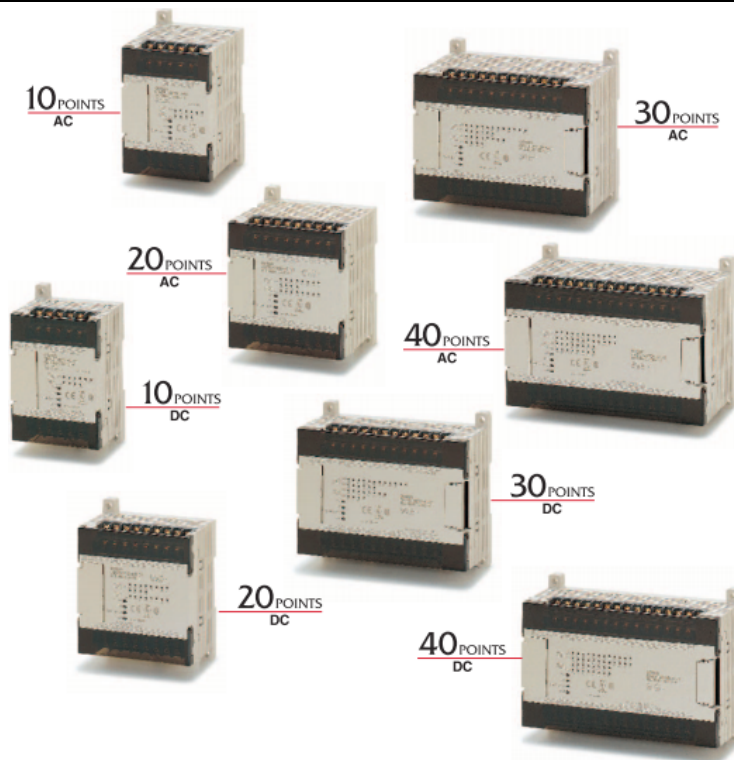
hệ thống truyền động mở cửa có thể vận hành bằng dòng điện AC hoặc DC.

\_ Nguyên lý hoạt động: Bộ ARD tự hoạt động khi thang máy bị mất điện, khi đó nó sẽ điều khiển tay quay của hộp số đưa cabin thang máy về đến tầng gần nhất và tự động mở cửa buồng thang.

## **Chương 2: KHÁI QUÁT VỀ PLC.**

### **2.1 Giới thiệu.**

**PLC (Programmable Logic Controller)** là thiết bị có thể lập trình được, được thiết kế chuyên dùng trong công nghiệp để điều khiển các tiến trình xử lý từ đơn giản đến phức tạp, tùy thuộc vào người điều khiển mà nó có thể thực hiện một loạt các chương trình và sự kiện sự kiện này được kích hoạt bởi các tác nhân kích thích (hay còn gọi là ngõ vào) tác động vào PLC hay còn gọi là bộ định thì (Timer) hay các sự kiện được đếm qua bộ đếm. Khi một sự kiện được kích hoạt nó sẽ bật ON hay OFF hoặc phát ra một chuỗi xung ra, các thiết bị bên ngoài được gắn vào ngõ ra của PLC. Như vậy, nếu ta thay đổi các chương trình được cài đặt trong PLC là ta có thể thực hiện các chức năng khác nhau. Hiện nay PLC đã được nhiều hãng khác nhau sản xuất như Siemens, Omron, Misubishi, Festo, Alan Bradley, Shneider, Hitachi...



Hình 2.1. Một số loại PLC của hãng OMRON.

Mặt khác, ngoài PLC cũng đã bổ sung thêm các thiết bị mở rộng khác như công mở rộng AI (Analog Input), DI (digital Input), các thiết bị hiển thị, các bộ nhớ Cartridge thêm vào.

Thiết bị điều khiển lập trình đầu tiên (Programmable Controller) đã được những nhà thiết kế cho ra đời năm 1968 (Công Ty Motor Mỹ). Tuy nhiên, hệ thống này còn khá đơn giản và cồng kềnh, người sử dụng gặp nhiều khó khăn trong việc vận hành hệ thống. Vì vậy, các nhà thiết kế từng bước cải tiến hệ thống, và cho hệ thống đơn giản, gọn nhẹ, dễ vận hành, nhưng việc lập trình cho hệ thống còn khó khăn, do lúc này không có thiết bị lập trình ngoại vi hỗ trợ cho công việc lập trình.

Để đơn giản hoá việc lập trình bằng tay (Programmable Controller Handle) đầu tiên được ra đời năm 1969. Điều này đã tạo ra được một sự phát triển thực sự cho kỹ thuật điều khiển lập trình. Trong giai đoạn này, các hệ thống điều khiển lập trình (PLC) chỉ đơn giản nhằm thay thế hệ thống relay và dây nối trong hệ thống điều khiển cổ điển. Qua quá trình vận hành, các nhà thiết kế đã từng bước tạo ra một tiêu chuẩn mới cho hệ thống, tiêu chuẩn đó là: dạng lập trình dùng

giản đồ hình thang (The Diagram Format). trong những năm đầu thập niên 1970, những hệ thống PLC còn có khả năng vận hành với những thuật toán hỗ trợ (arithmetic) “vận hành với các dữ liệu cập nhật” (data manipulation), do sự phát triển của loại màn hình dùng cho máy tính Cathode RayTube (CRT), nên việc giao tiếp giữa người điều khiển lập trình cho hệ thống càng trở nên thuận tiện hơn.

Sự phát triển của hệ thống phần cứng và phần mềm từ năm 1975 cho đến nay, đã làm cho hệ thống PLC phát triển mạnh hơn với các chức năng mở rộng: Hệ chương trình tăng lên.

Hơn 128000 từ bộ nhớ ( word of memory), có thể gắn thêm nhiều module bộ nhớ để có thể tăng thêm kích thước chương trình. Ngoài ra, các nhà thiết kế còn tạo ra kỹ thuật nối với các hệ thống PLC riêng lẻ thành một hệ thống PLC chung, kết nối với các hệ thống máy tính, tăng khả năng điều khiển của từng hệ thống riêng lẻ. Tốc độ xử lý của hệ thống được cải thiện, chu kỳ quét (scan) nhanh hơn, làm cho hệ thống PLC xử lý tốt với những chức năng phức tạp, số lượng cổng ra / vào lớn hơn. Một số thuật toán cơ bản dùng cho điều khiển cũng được tích hợp vào phần cứng như điều khiển PID (cho điều khiển nhiệt độ, cho điều khiển tốc độ động cơ, cho điều khiển vị trí), điều khiển mờ, lọc nhiễu ở tín hiệu đầu vào,...

Trong tương lai hệ thống PLC không chỉ giao tiếp với các hệ thống khác thông qua CIM ( Computer Intergrated Manufacturing ) để điều khiển hệ thống robot, Cad / Cam,...Ngoài ra, các nhà thiết kế còn đang xây dựng các loại PLC với các chức năng điều khiển “ thông minh” (Intelligence) còn gọi là các siêu PLC (Super PLC) cho tương lai.

## **2.2 Đặc điểm của hệ thống lập trình PLC**

Nhu cầu về một bộ điều khiển để sử dụng linh hoạt và có giá thành thấp, đã thúc đẩy sự phát triển những hệ thống điều khiển lập trình (Programmable Controller Systems). Hệ thống sử dụng CPU và bộ nhớ để điều khiển máy móc hay quá trình hoạt động. Trong bối cảnh đó, bộ điều khiển lập trình PLC (Programmable Logic Controller) được thiết kế nhằm thay thế phương pháp

truyền thống dùng rơ le và thiết bị rời công kênh, nó tạo ra một khả năng điều khiển thiết bị dễ dàng, linh hoạt dựa trên việc lập trình trên các lệnh logic cơ bản.

Ngoài ra, PLC còn có thể thực hiện những tác vụ khác như định thì, đếm, ....Làm tăng khả năng điều khiển cho những hoạt động phức tạp, ngay cả với loại PLC nhỏ nhất.

Hoạt động của PLC là kiểm tra tất cả những trạng thái tín hiệu ngõ vào, được đưa về từ quá trình điều khiển, thực hiện logic được tập trung trong chương trình và kích ra tín hiệu để điều khiển cho thiết bị bên ngoài tương ứng. Với các mạch giao tiếp chuẩn ở khối vào và khối ra của PLC cho phép nó kết nối trực tiếp đến những cơ cấu tác động (actuators) có công suất nhỏ ở ngõ ra, và những mạch chuyển đổi tín hiệu (transducers) ở ngõ vào, mà không cần có các mạch giao tiếp hay rơ le trung gian. Tuy nhiên, cần phải có mạch điện tử công suất trung gian khi PLC điều khiển những thiết bị có công suất lớn.

Việc sử dụng PLC cho phép chúng ta hiệu chỉnh hệ thống mà không cần có sự thay đổi về mặt kết nối dây, sự thay đổi chỉ là thay đổi chương trình điều khiển trong bộ nhớ thông qua thiết bị lập trình chuyên dùng. Hơn nữa, chúng còn có ưu điểm là thời gian lắp đặt và đưa vào hoạt động nhanh hơn so với những hệ thống các thiết bị rời.

Về phần cứng PLC tương tự như máy tính “truyền thống”, và chúng có các đặc điểm thích hợp cho mục đích điều khiển trong công nghiệp:

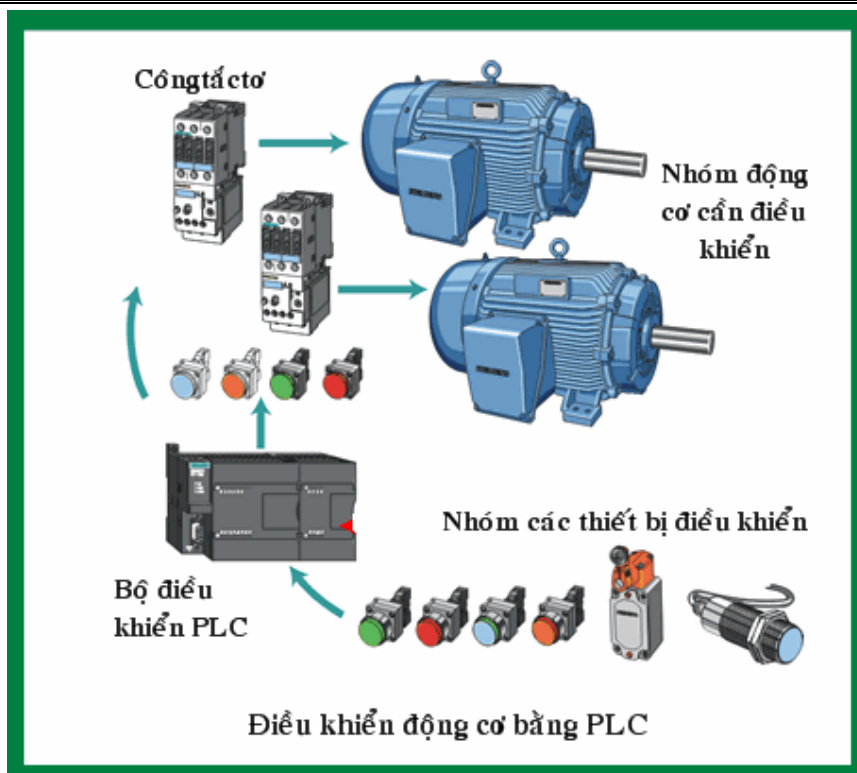
Khả năng kháng nhiễu tốt.

Ngôn ngữ lập trình chuyên dùng : Ladder, TTL....dễ hiểu và dễ sử dụng.

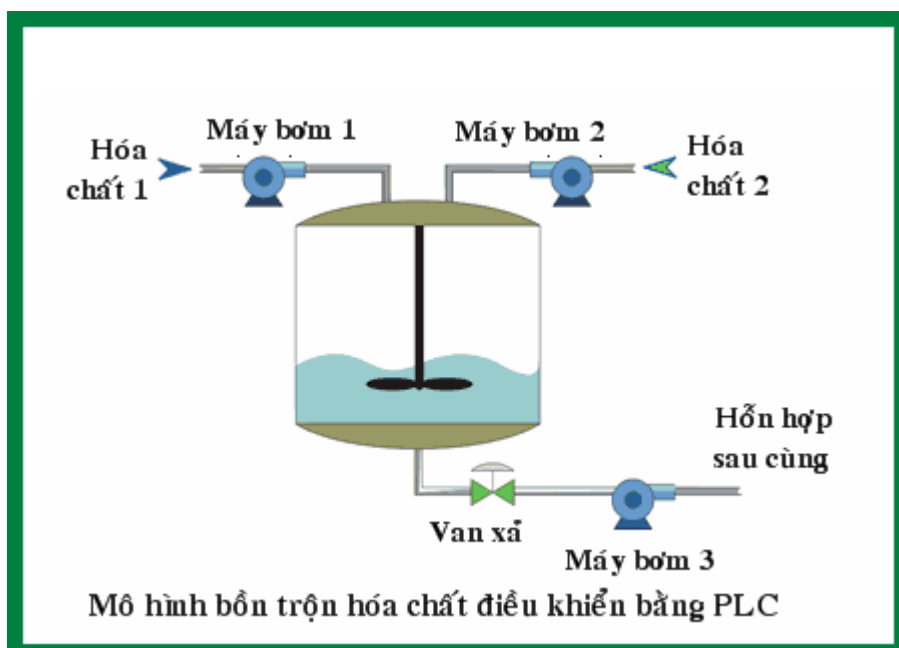
Thay đổi chương trình dễ dàng.

Những đặc điểm trên làm cho PLC được sử dụng nhiều trong việc điều khiển các máy móc công nghiệp và trong quá trình điều khiển quá trình (process-control).

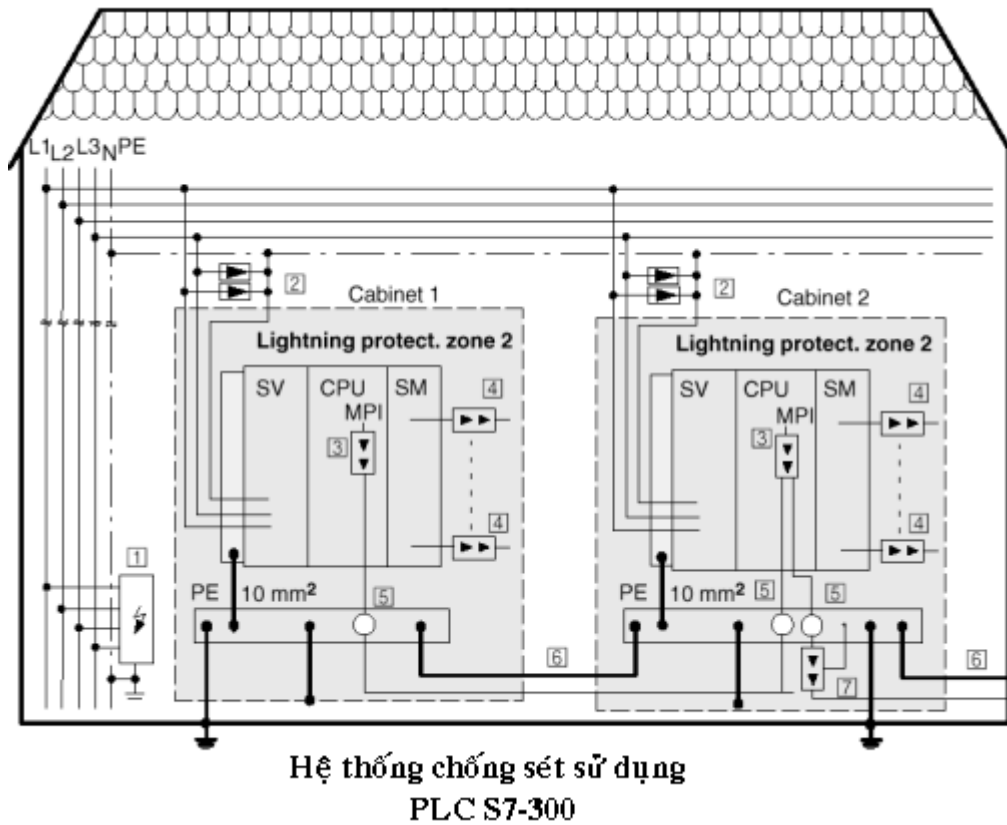
Một số ứng dụng cụ thể điều khiển bằng PLC thông dụng:



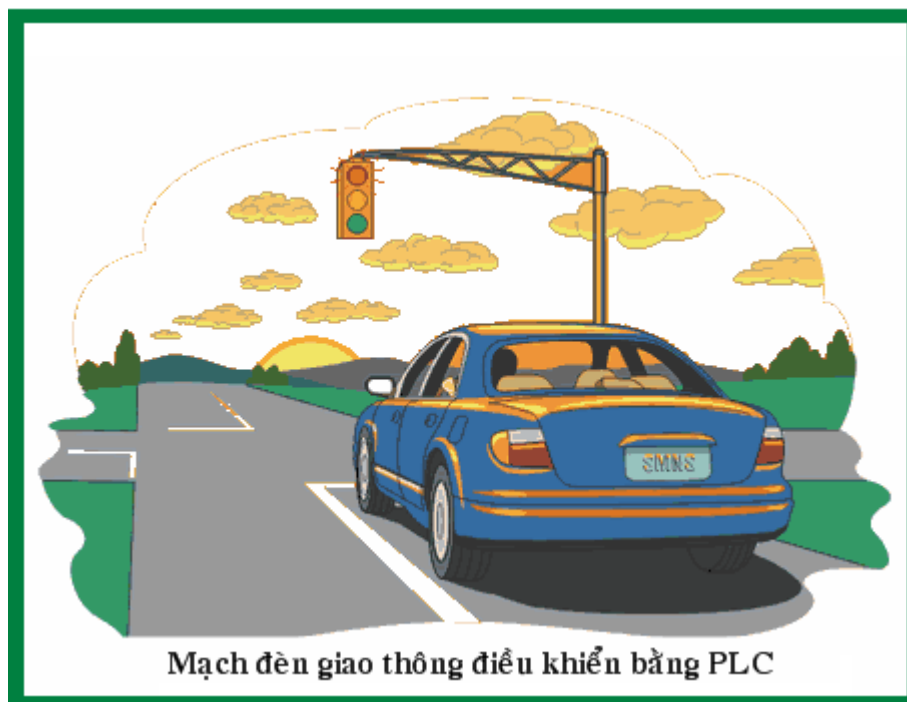
Hình 2.2. Điều khiển động cơ bằng PLC.



Hình 2.3. Mô hình bồn trộn hóa chất điều khiển bằng PLC.



Hình 2.4. Hệ thống chống sét dùng PLC S7-300.



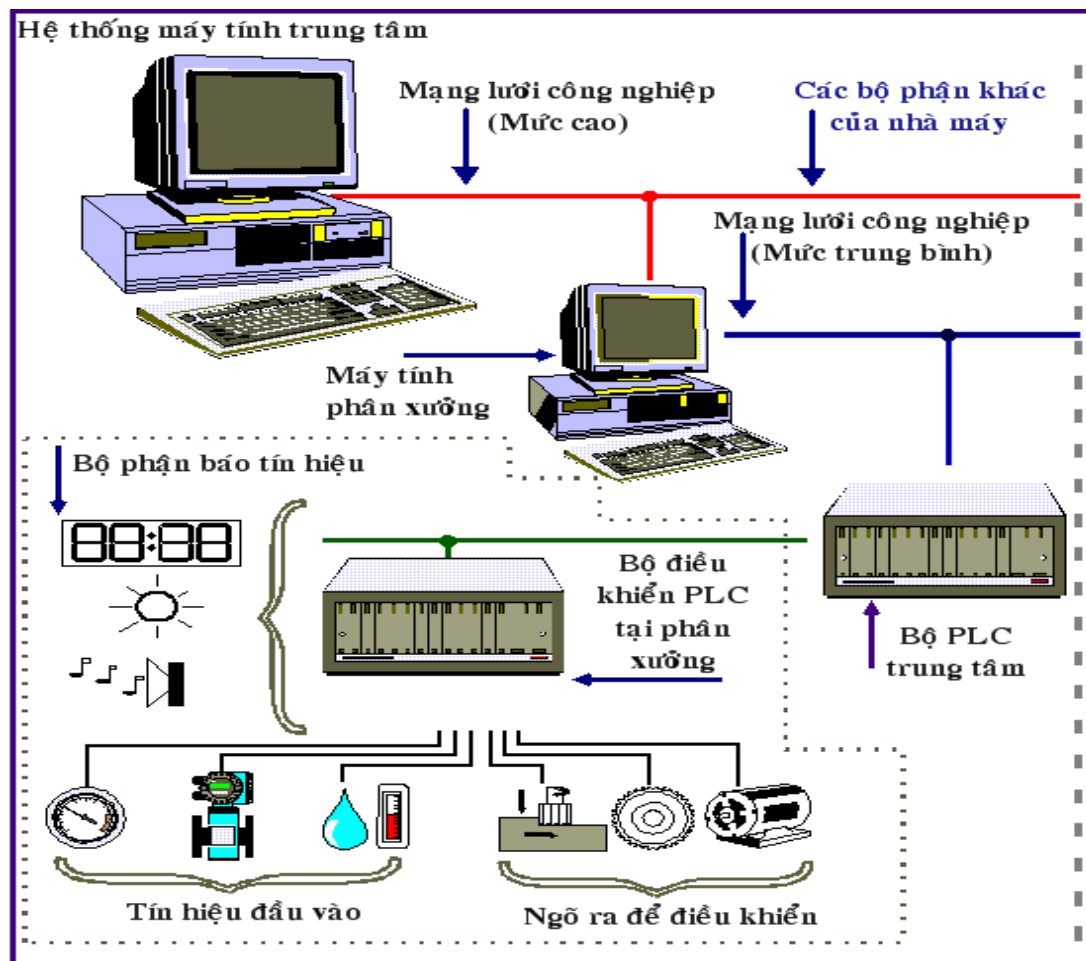
Hình 2.5. Điều khiển đèn giao thông dùng PLC.

Mỗi phần tử, hoặc thiết bị của một hệ thống điều khiển công nghiệp bất chấp kích thước của nó là nhỏ hay lớn đều có vai trò hết sức quan trọng trong quá

trình điều khiển. Chẳng hạn như, nếu không có thiết bị cảm biến, bộ PLC sẽ không biết chính xác cái gì đang xảy ra trong quá trình.

Trong hệ thống tự động hóa, bộ điều khiển PLC là phần tử trung tâm của cả hệ thống điều khiển. Bằng việc thực hiện các chương trình đã được lưu trữ trong bộ nhớ, PLC còn liên tục theo dõi trạng thái của cả hệ thống thông qua các tín hiệu được đưa vào. Dựa vào các thuật toán logic được thực hiện bên trong chương trình, PLC sẽ xác định những hoạt động nào cần thiết đưa ra cung cấp cho các thiết bị.

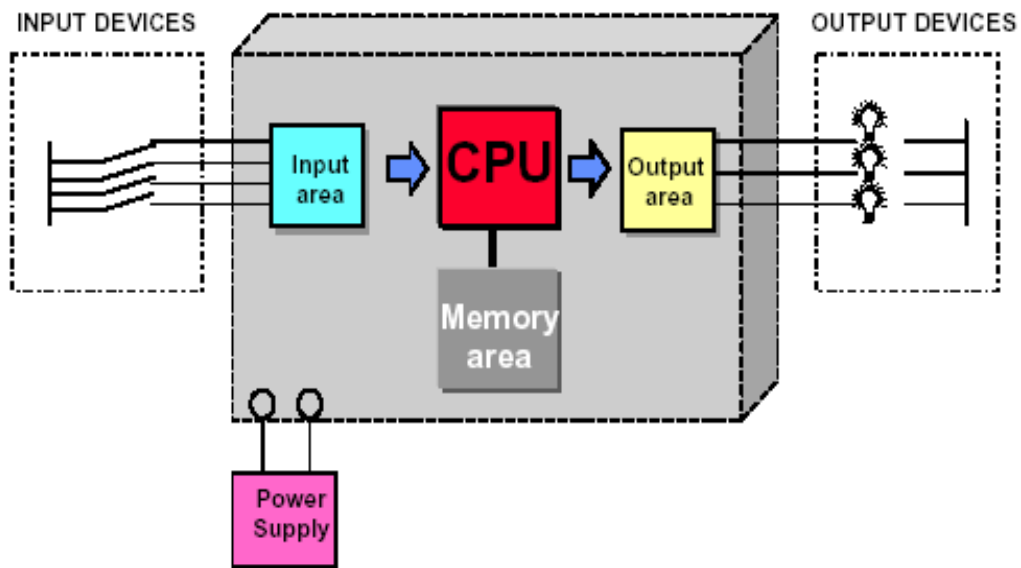
Nếu muốn các hoạt động phức tạp cao cấp hơn, cần có nhiều bộ PLC kết nối với máy tính trung tâm.



Hình 2.6 Sơ đồ mạng lưới điều khiển công nghiệp dùng PLC

### 2.3 Cấu tạo chính của bộ lập trình PLC





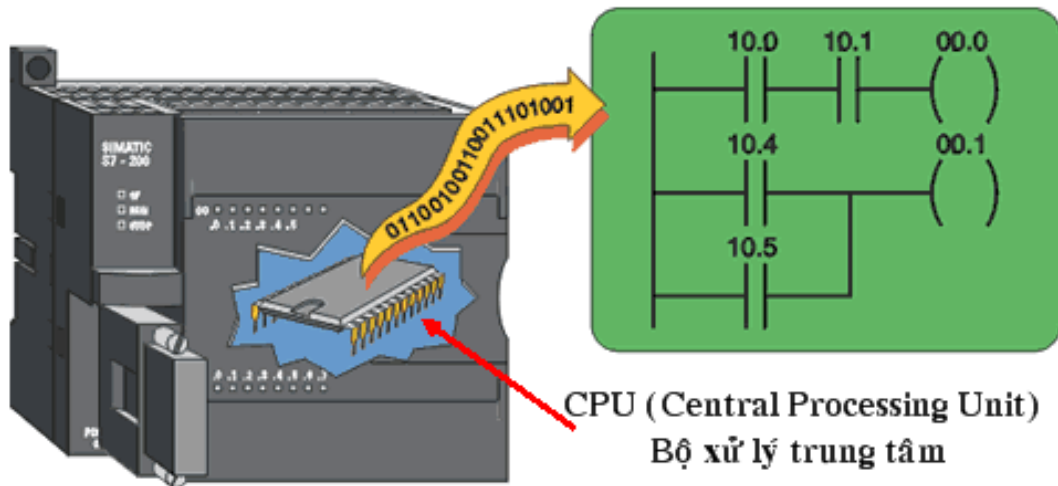
Hình 2.7. Cấu tạo cơ bản của PLC.

PLC gồm có 5 thành phần cơ bản sau:

### 2.3.1. Đơn vị xử lý trung tâm CPU (Central Processing Unit):

\_ Đơn vị xử lý trung tâm được xem như là não của bộ điều khiển PLC. Thông thường, đơn vị xử lý trung tâm là một loại vi điều khiển, như vi điều khiển 8051 có 8 bit và ngày nay chúng lên đến 16 bit hay 32 bit.

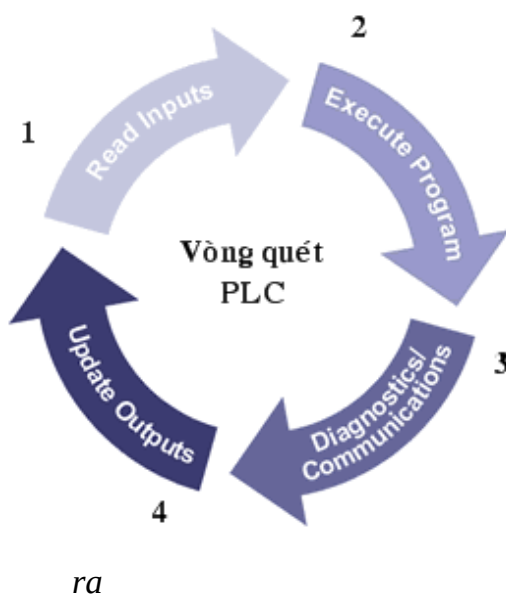
\_ Đơn vị xử lý trung tâm chú trọng phần truyền thông giữa các bộ phận của bộ điều khiển PLC với nhau như việc lập trình, quản lý bộ nhớ, quan sát trạng thái ngõ vào và ngõ ra. Đơn vị xử lý trung tâm thường thực hiện việc kiểm tra vùng nhớ của bộ điều khiển PLC để bảo đảm rằng bộ nhớ không bị lỗi, không bị hỏng, nhờ đó mà các lỗi nếu có sẽ sớm được phát hiện.



Hình 2.8. Bộ xử lý trung tâm của PLC.

Nguyên lý vận hành của một đơn vị xử lý trung tâm CPU được mô tả như sau:

\_ Các thông tin được lưu trữ trong bộ nhớ chương trình được gọi lên tuần tự vì đã được điều khiển và kiểm soát bằng bộ đếm chương trình do đơn vị xử lý trung tâm khống chế. Bộ xử lý liên kết các tín hiệu riêng lẻ lại với nhau và từ đó cho ra kết quả điều khiển tại ngõ ra. Sự thao tác tuần tự của chương trình tạo nên một khoảng thời gian trễ gọi là thời gian quét, vì tính tuần tự của nó nên ta có thể gọi là chu kỳ quét, chu kỳ quét này phụ thuộc vào của chương trình (số lượng ngõ vào, ngõ ra, và những thông tin yêu cầu khác). Chính đơn vị xử lý trung tâm quyết định thời gian quét, chức năng và khả năng của một bộ PLC



1. Đọc trạng thái ngõ vào
2. Thực hiện chương trình
3. Kiểm tra thông tin
4. Truyền dữ liệu ở ngõ ra

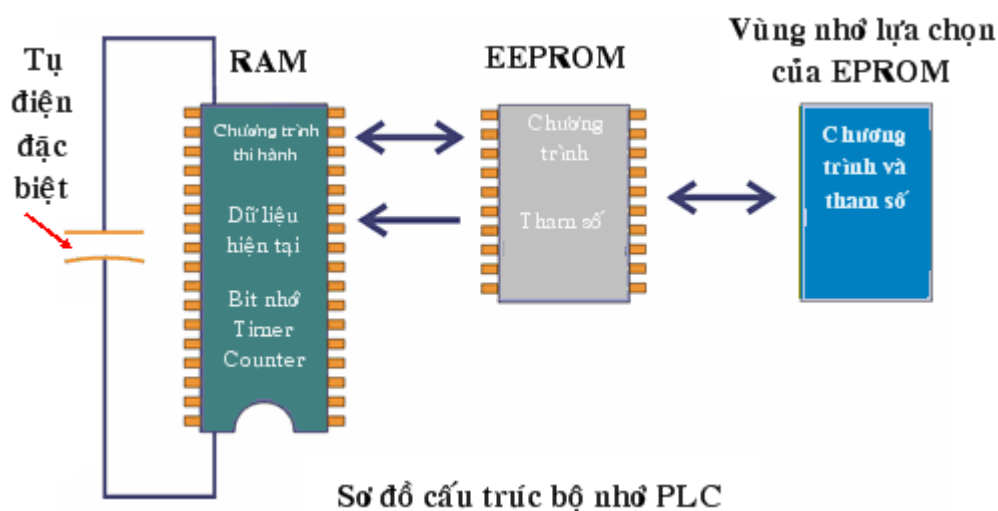
Hình 2.9. Vòng quét PLC

### 2.3.2. Vùng nhớ (Memory).

\_ Vùng nhớ được PLC sử dụng cho một quá trình điều khiển công nghiệp. Người ta thường sử dụng bộ nhớ EPROM với việc lập trình cần được thực hiện trên máy lập trình còn khi muốn xóa bỏ dữ liệu cần chiếu tia tử ngoại vào, hoặc dữ liệu cũng có thể được xóa bằng năng lượng điện. Nếu muốn phát triển thêm các ứng dụng khác của chương trình điều khiển ta chỉ cần thực hiện lập trình lại cho bộ điều khiển thông qua sợi cáp kết nối.

\_ Bộ nhớ thường được chia thành các khối với những chức năng đặc biệt khác nhau. Một số vùng của bộ nhớ dùng cho việc lưu trữ trạng thái ngõ vào và ngõ ra. Trạng thái của một ngõ vào được lưu trữ dưới dạng bit nhớ đặc biệt, thường là '1' hoặc '0'. Bit nhớ của mỗi trạng thái vào hoặc ra cũng sẽ có trạng thái tương tự.

\_ Những phần khác của bộ nhớ được dùng cho việc lưu trữ các nội dung có giá trị cần thiết cho việc lập trình, ví dụ như giá trị role thời gian, giá trị counter... được lưu trữ trong phần này. Thế mạnh của bộ điều khiển lập trình PLC đó là có bộ nhớ có thể thay đổi một cách nhanh chóng.



Hình 2.10. Cấu trúc bộ nhớ PLC.

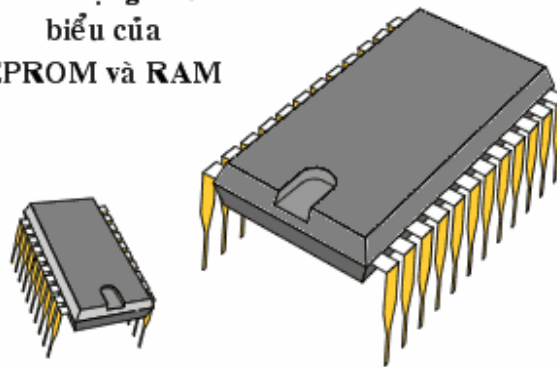
\_ Tụ điện đặc biệt: được gọi là tụ điện đặc biệt do nó có khả năng tích trữ năng lượng trong một thời gian dài, giúp lưu trữ dữ liệu trong RAM khi không có điện. Loại RAM thông thường có thời gian lưu trữ dữ liệu đến 50 giờ, cũng có thể lên đến 72 giờ.

\_ RAM (Random Access Memory): RAM được sử dụng như 1 vùng nhớ tạm thời. Vùng nhớ của RAM không ổn định, các dữ liệu lưu trên nó sẽ bị mất đi khi bị mất điện. Do đó, thường có 1 nguồn dự trữ để đề phòng trường hợp RAM bị mất điện trong 1 thời gian dài.

\_ EPROM (Erasable Programmable Read Only Memory): Nó được thiết kế sao cho dữ liệu có thể được đọc dễ dàng, nhưng khó có thể thay đổi. Để thay đổi dữ liệu của EPROM cần có phương pháp đặc biệt. Đối với UVEEPROM, dữ liệu có thể được thay đổi bằng cách chiếu tia cực tím (Ultraviolet Light) vào. Nhưng đối với EPROM thông dụng thì có thể dùng điện để xóa dữ liệu.

\_ Firmware: Là một phần mềm đặc biệt để đưa dữ liệu vào EPROM. Do đó EPROM có thể được xem như một bộ phận của phần cứng của PLC, nó cho phép PLC sử dụng các chức năng cơ bản của nó.

Hình dạng tiêu  
biểu của  
EPROM và RAM



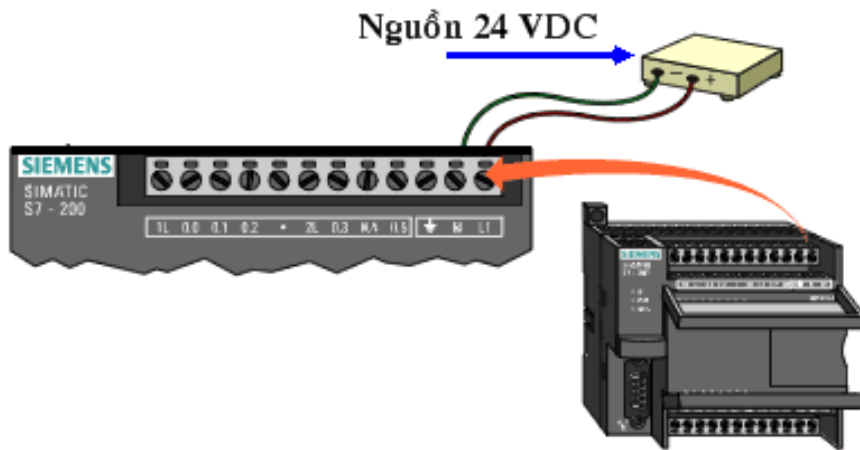
Hình 2.11. Hình dạng tiêu biểu của RAM và EPROM.

### 2.3.3 Nguồn điện cung cấp

Điện cấp vào được dùng cho đơn vị xử lý trung tâm CPU, đa số các bộ điều khiển PLC sử dụng nguồn điện 24 VDC hoặc 220 VAC.

Người sử dụng cần nắm rõ số lượng đầu vào và đầu ra để bảo đảm thiết bị được cấp điện một cách chính xác. Mỗi modul khác nhau thì khả năng sử dụng điện khác nhau. Nguồn điện cung cấp này không được dùng để khởi động cho các

thiết bị kết nối phía bên ngoài tại ngõ vào, hoặc ngõ ra. Người sử dụng phải cấp điện cho các thiết bị tại đầu vào hoặc đầu ra phải được tiến hành một cách riêng biệt. Có như vậy mới bảo đảm được rằng những ảnh hưởng của các thiết bị máy móc dùng trong công nghiệp không gây hư hại cho bộ điều khiển PLC. Đối với một số bộ điều khiển PLC loại nhỏ, chúng cấp nguồn cho các thiết bị kết nối tại ngõ vào bằng điện áp được lấy từ một nguồn nhỏ đã được tích hợp vào bộ điều khiển PLC.

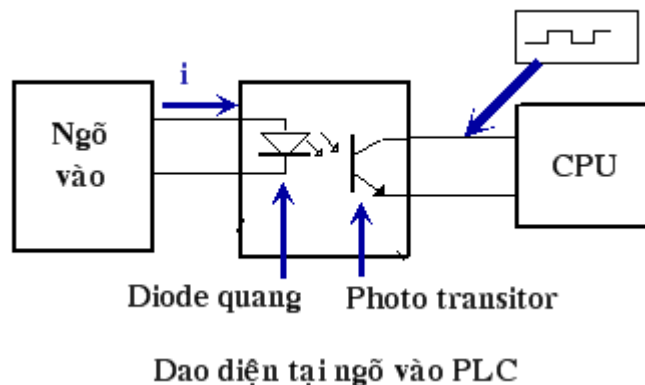


Hình 2.12. Cấp nguồn cho PLC.

### 2.3.4 Module ngõ vào

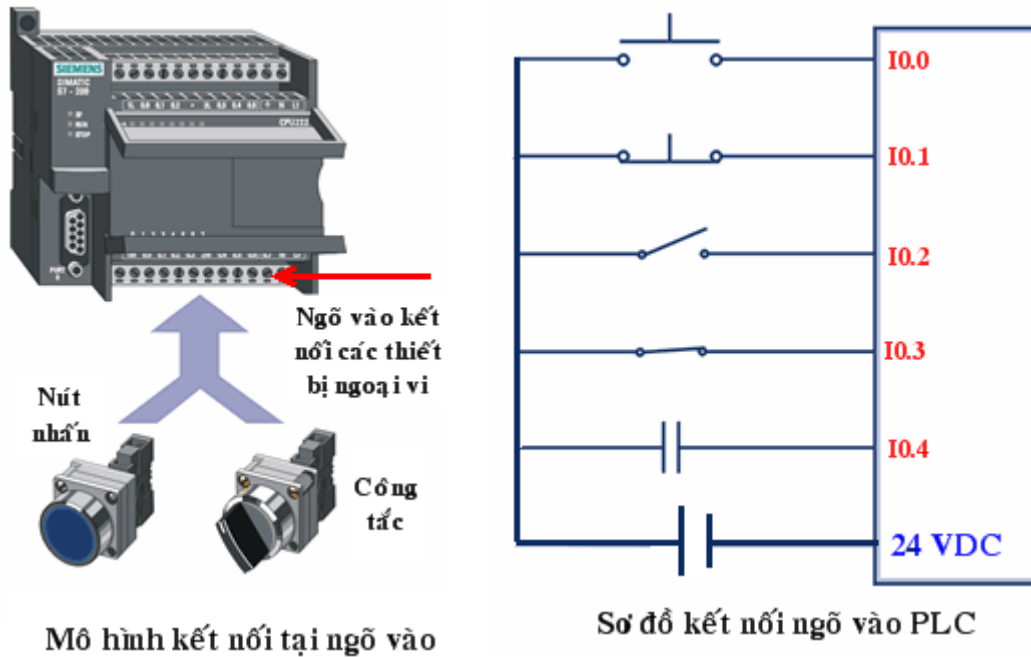
\_ Làm nhiệm vụ khối ghép, chuyển đổi tín hiệu đầu vào thành tín hiệu số bên trong PLC. Kết quả của việc xử lý được lưu trữ trong vùng nhớ của ngõ vào.

\_ Mạch đầu vào được cách ly về điện với các mạch bên trong PLC nhờ các diode quang và photo diode (thường gặp là diode 4N28) do đó mọi hư hỏng ở mạch đầu vào đều không ảnh hưởng đến hoạt động của PLC.



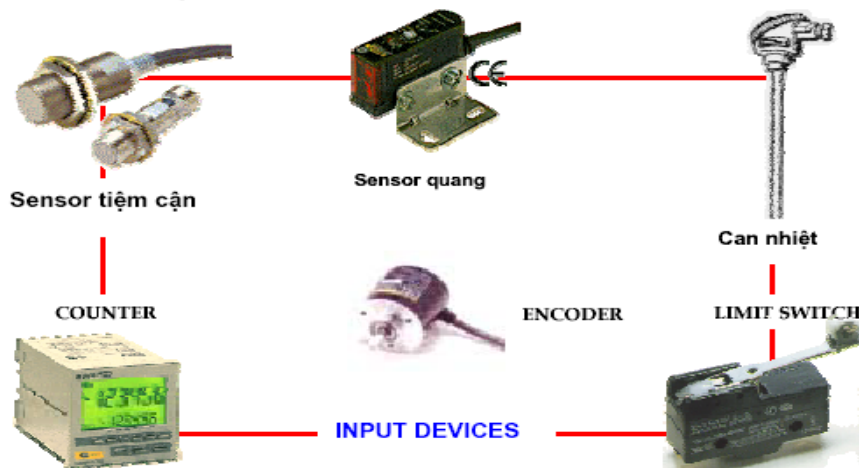
Hình 2.13. Giao diện ngõ vào của PLC.

\_ Các thiết bị đầu vào có thể là nút nhấn, công tắc, công tắc hành trình, tiếp điểm (thường mở, thường đóng), các bộ cảm biến...Ký hiệu: I0.0 là ngõ vào thứ 1, I0.1 là ngõ vào thứ 2..... Một bộ PLC có thể có nhiều ngõ vào.



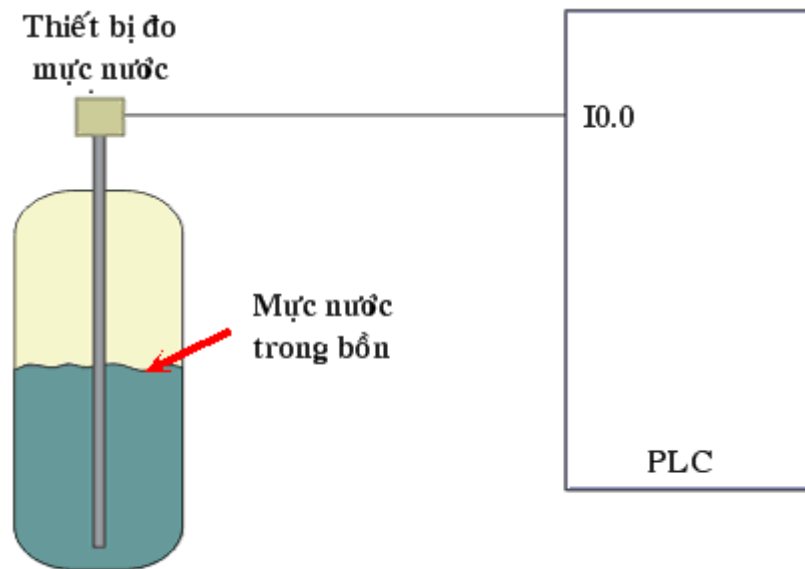
Hình 2.14. Kết nối ngõ vào của PLC.

➤ Các thiết bị vào ra thường gặp



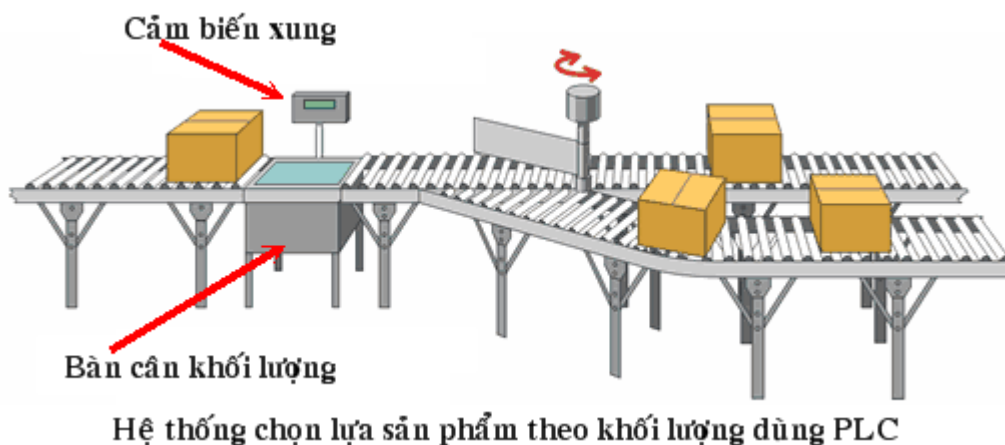
Hình 2.15. Các thiết bị vào thường gặp trong PLC

Ngõ vào cũng có thể được điều khiển bằng trạng thái của các yêu cầu cụ thể của một chương trình điều khiển. Như các hệ thống bên dưới đây:



Điều khiển bơm chất lỏng dùng PLC

Hình 2.15. Điều khiển bơm chất lỏng dùng PLC.



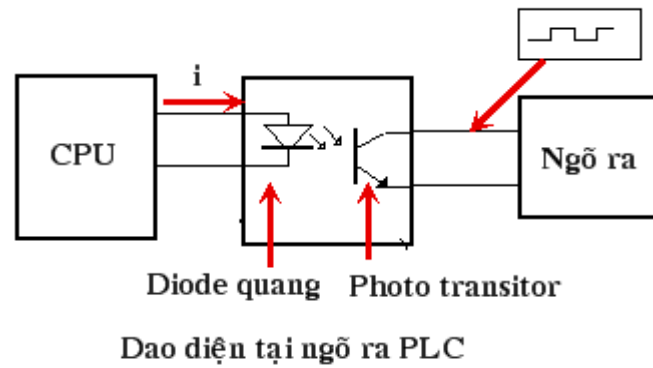
Hệ thống chọn lựa sản phẩm theo khối lượng dùng PLC

Hình 2.15. Hệ thống phân loại sản phẩm theo khối lượng dùng PLC.

Trong các sơ đồ trên, tín hiệu của các thiết bị (thiết bị đo mực nước, cảm biến xung, bàn cân khối lượng) đều được kết nối với tín hiệu ngõ vào của bộ PLC.

### 2.3.5 Module ngõ ra:

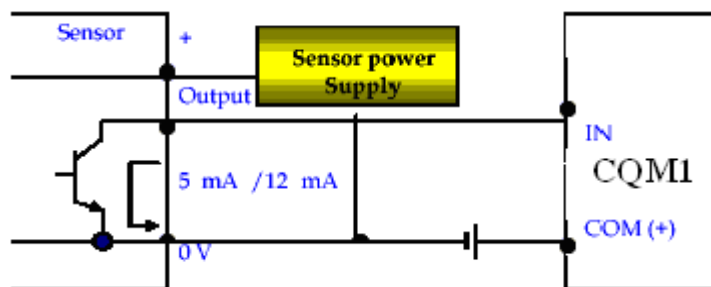
Làm nhiệm vụ biến đổi các mức logic bên trong PLC thành các tín hiệu điều khiển đưa ra bên ngoài. Tương tự như tại ngõ vào, ngõ ra của PLC cũng được cách ly về điện đối với các thiết bị bên ngoài bằng bộ opto (diode quang và photo transistor), tiếp điểm rơle (Relay) hoặc transistor.



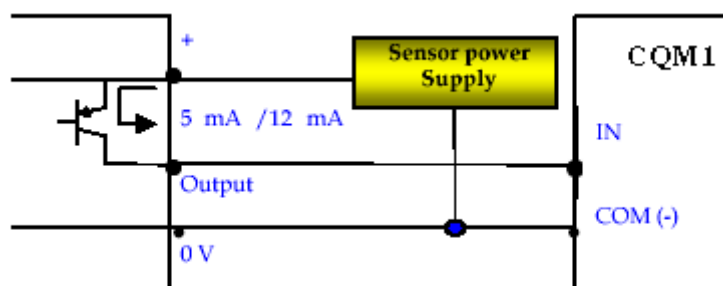
Hình 2.16. Giao diện ngõ ra của PLC.



Hình 2.17. Đầu vào loại tiếp điểm rơle (Relay).

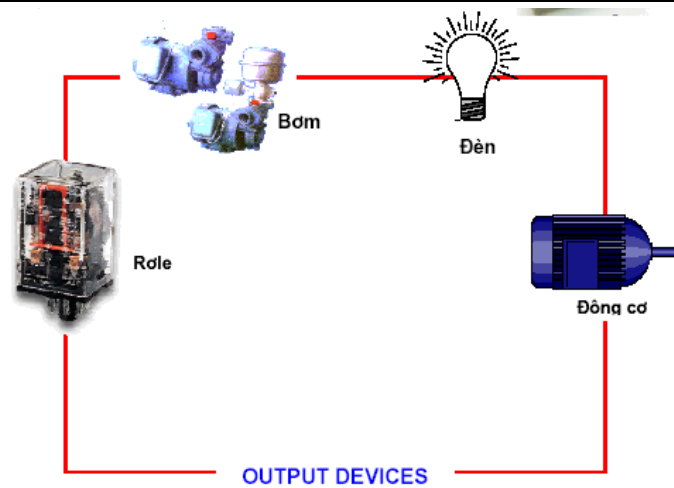


Hình 2.18. Đầu vào loại transistor kiểu NPN.



Hình 2.19. Đầu vào loại transistor kiểu PNP.

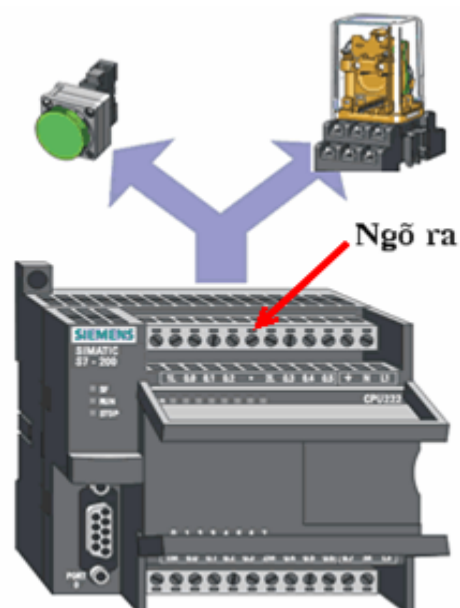




Hình 2.20. Các thiết bị ra thường gặp trong PLC.

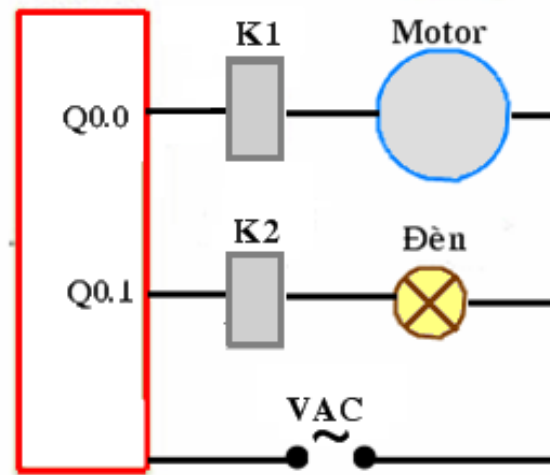
Ngõ ra của PLC được gọi là ngõ ra kỹ thuật số, nó kết nối với các thiết bị cần điều khiển như van điện từ, cuộn dây công tắc tơ (bộ khởi động từ), bóng đèn...

Ký hiệu tại mỗi ngõ ra của PLC là Q0.0, Q0.1, Q0.2 ...



Mô hình kết nối tại ngõ ra PLC

Hình 2.21. Mô hình kết nối ngõ ra PLC.



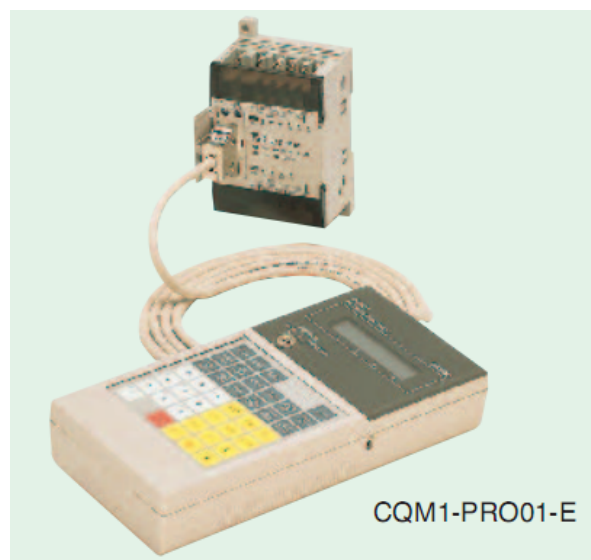
Sơ đồ kết nối ngõ ra PLC

Hình 2.21. Sơ đồ kết nối ngõ ra PLC.

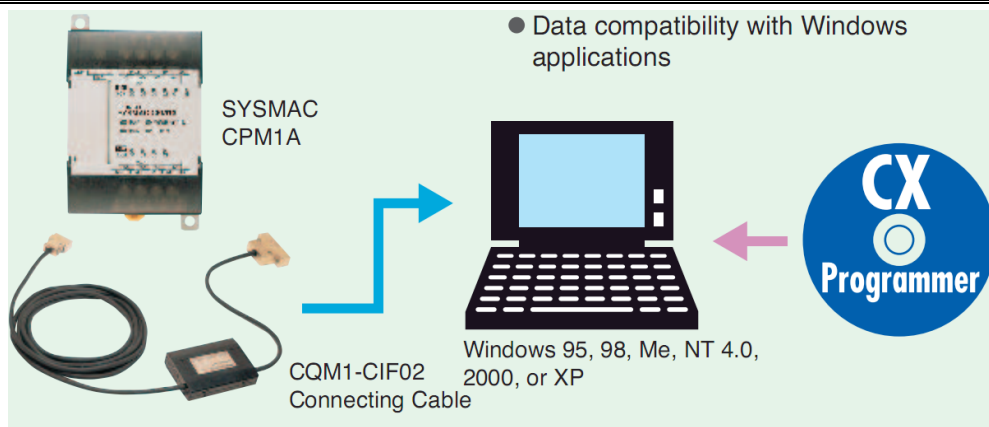
Số lượng ngõ vào và ngõ ra của mỗi bộ PLC là có giới hạn, nhưng ta có thể tăng số lượng của chúng lên nhờ các bộ module mở rộng, tức là ta có thể ghép nối giữa các bộ PLC với nhau.

### 2.3.6 Lập trình cho bộ điều khiển PLC

Cách thông dụng nhất để lập trình cho bộ điều khiển PLC là thông qua máy tính đã được cài đặt phần mềm lập trình hoặc các thiết bị lập trình tích hợp sẵn do nhà sản xuất cung cấp sẵn.

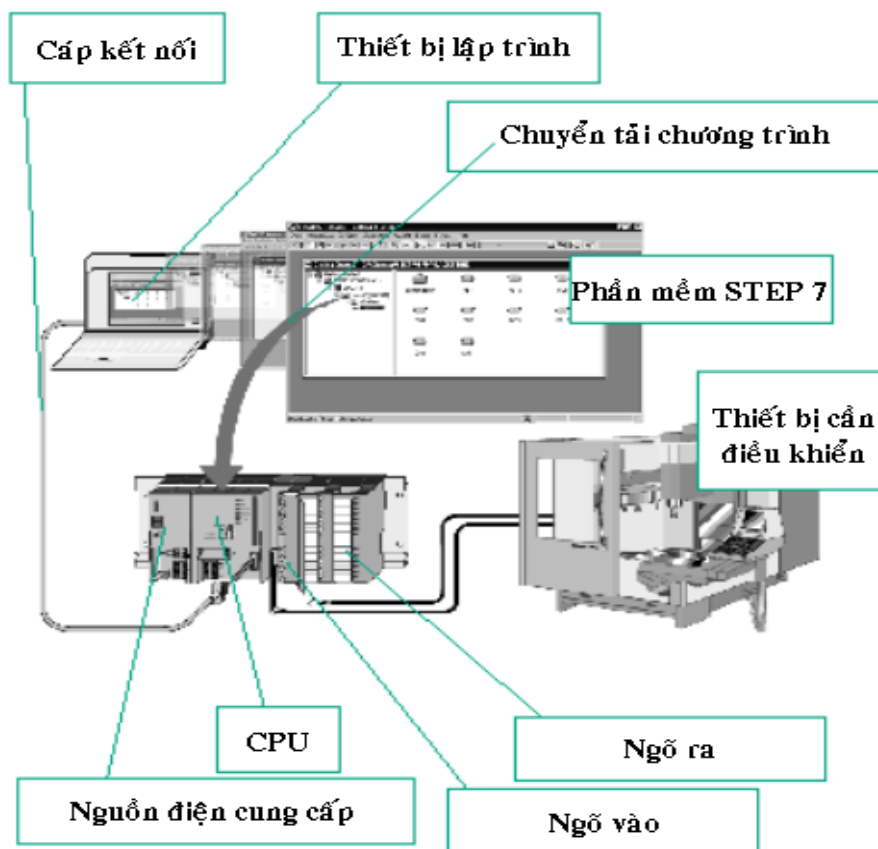


Hình 2.22. Bộ lập trình PLC bằng tay của OMRON.



Hình 2.23. Lập trình PLC OMRON thông qua máy tính.

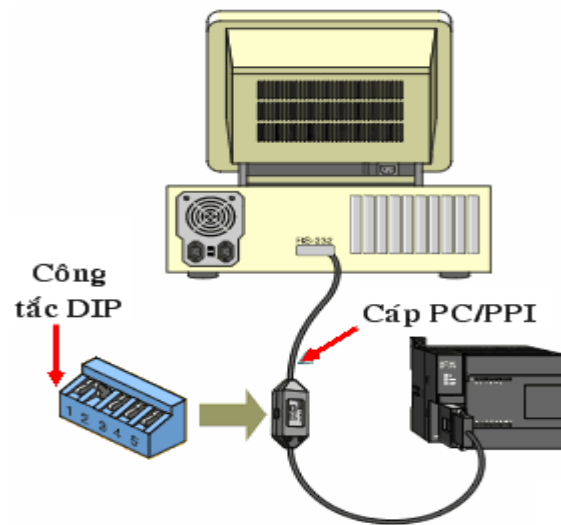
Ngày nay, việc sử dụng máy tính cho việc lập trình và tải lập trình cho những bộ PLC dùng trong các nhà máy rất phổ biến, nó giữ vai trò rất quan trọng trong công nghiệp.



Hình 2.24. Sơ đồ kết nối của bộ PLC S7-300.

### 2.3.7 Một số bộ phận phụ.

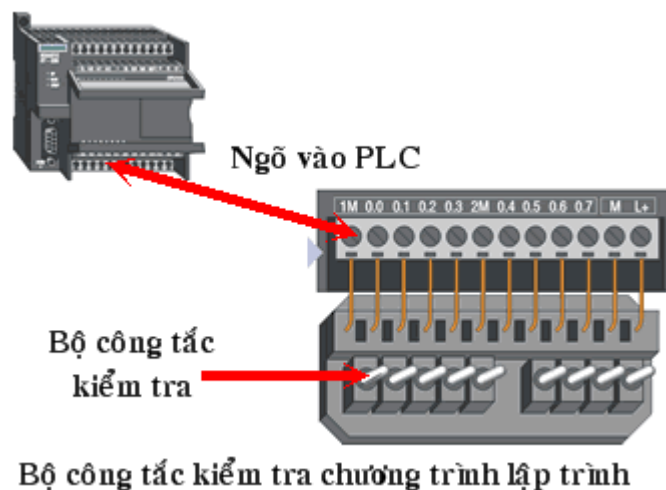
**Cáp nối (PC/PPI):** Dùng để kết nối giao diện máy tính với giao diện bộ PLC. Trên cáp, có công tắc DIP cho phép chọn lựa tốc độ truyền thông tin thích hợp giữa máy tính và bộ PLC.



Sơ đồ cáp nối máy tính với bộ PLC

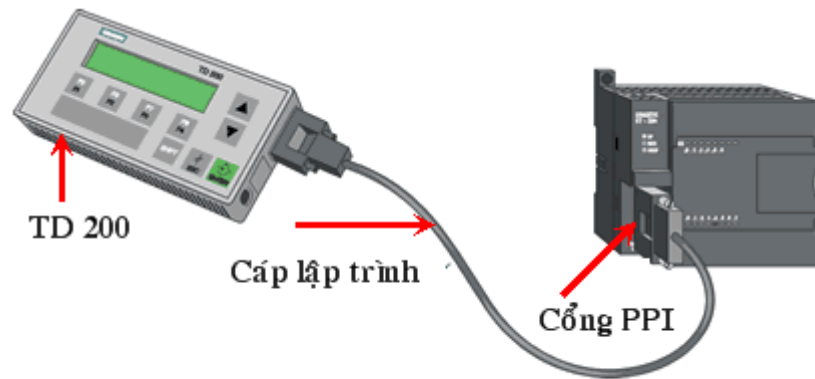
Hình 2.25. Sơ đồ cáp nối máy tính với bộ PLC.

Việc kiểm tra trạng thái của chương trình lập trình PLC không cần thiết phải lắp các thiết bị điều khiển tại đầu vào, ta có thể tiến hành kiểm tra bằng bộ công tắc thay thế cho các tiếp điểm trên thực tế. Bộ công tắc này sẽ cung cấp mức logic 0/1 cho bộ PLC.



Hình 2.26. Bộ công tắc kiểm tra chương trình lập trình.

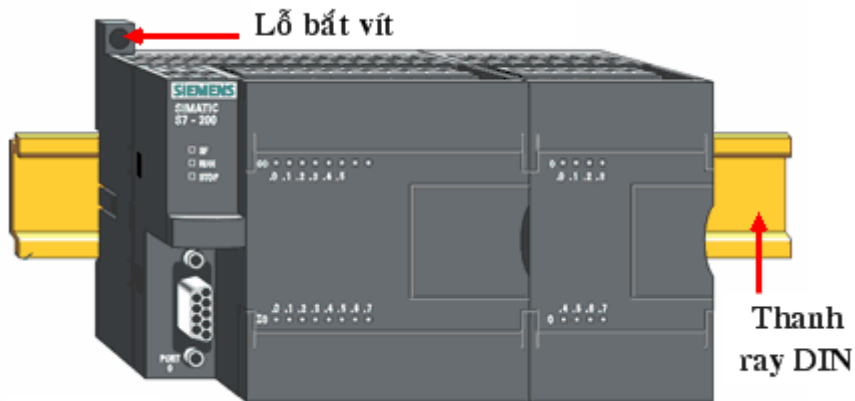
Ngoài việc dùng máy tính để lập trình cho bộ PLC, ta cũng có thể tiến hành lập trình hoặc thay đổi chương trình của bộ PLC bằng một số thiết bị cầm tay.



Bộ lập trình cầm tay TD 200 cho PLC S7-200

Hình 2.27. Bộ lập trình TD 200 cho S7-200.

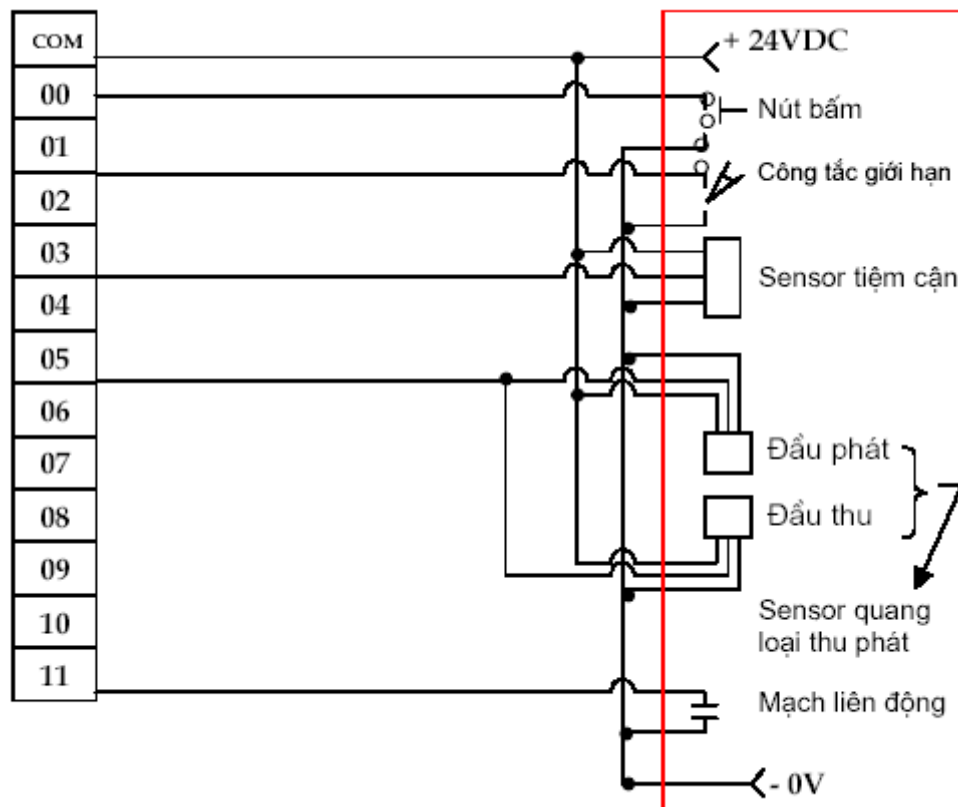
**Lắp đặt:** Để lắp đặt PLC lên bảng điện, ta có thể dùng thanh ray DIN hoặc bắt vít trực tiếp lên bảng điện.



Hình 2.28. Lắp đặt bộ PLC.

## 2.4 Nối dây đầu vào PLC switch

Dưới đây là 1 ví dụ khi đấu dây đầu vào với các thiết bị có trong thực tế thay cho công tắc mô phỏng.



Hình 2.29. sơ đồ đấu dây cho các ngõ vào của PLC

\_ Trong thiết kế hệ thống điều khiển PLC trong công nghiệp, trước hết cần phải lựa chọn những thiết bị hay hệ thống cần điều khiển. Hệ thống tự động hóa có thể chỉ có 1 máy hoặc cũng có thể là cả một dây chuyền gồm nhiều máy, nhưng chúng được gọi chung là hệ thống điều khiển quy trình công nghiệp. Hàm của một hệ thống điều khiển quy trình công nghiệp đó là quan sát các tín hiệu ngõ vào để cung cấp tín hiệu cho bộ điều khiển PLC. Đáp lại, bộ điều khiển PLC sẽ xuất tín hiệu điều khiển ra ngoài để vận hành các thiết bị.

\_ Kế tiếp, cần phải biết rõ số ngõ vào và ngõ ra của thiết bị sẽ kết nối với bộ điều khiển PLC. Ngõ vào có thể là các tín hiệu khác nhau, công tắc, cảm biến... Ngõ ra sẽ điều khiển các thiết bị như cuộn dây, van điện từ, động cơ, rơle, các thiết bị tạo âm thanh và ánh sáng.

\_ Tiếp theo việc xác định các thiết bị tại ngõ vào và ngõ ra đó là gán cho bộ PLC những thiết bị nào thuộc ngõ vào, ngõ ra. Sự phân chia trạng thái ngõ vào và ngõ ra cho bộ điều khiển PLC phải tương ứng với hệ thống đã được thiết kế.

\_ Sau cùng, tạo sơ đồ hình thang (dạng Ladder) cho chương trình theo bước đầu đã nêu. Cuối cùng chương trình sẽ được tải vào bộ điều khiển PLC. Khi việc nạp chương trình kết thúc, nó sẽ kiểm tra và phát hiện các lỗi có trong chương trình (bằng cách sử dụng chức năng dò lỗi). Nếu có thể, toàn bộ chương trình sẽ được chạy mô phỏng. Trước khi chương trình bắt đầu, ta cần phải kiểm tra lại các ngõ vào và ngõ ra để bảo đảm chúng đã được kết nối đúng. Kế tiếp, cấp nguồn và cho hệ thống hoạt động.

\_ Việc thường xuyên kiểm tra dữ liệu trong bộ điều khiển để bảo đảm chương trình trong đó không bị thay đổi là hết sức cần thiết, nó giúp tránh được các tình trạng nguy hiểm xảy ra trong sản xuất. Do đó, một số thiết bị tự động được thiết lập trong mạng lưới thông tin của nhà máy nhằm thực hiện việc kiểm tra các chương trình PLC để bảo đảm rằng chương trình của chúng không mắc lỗi.

\_ Người lập trình có thể đánh dấu, đặt tên cho các thiết bị ngõ vào và ngõ ra. Điều này sẽ tạo được thuận lợi khi xác định lỗi, hoặc khi bảo trì cho hệ thống. Việc đặt tên và hướng dẫn khi lập trình phải bảo đảm sao cho mọi nhân viên kỹ thuật đều có thể hiểu, vận hành và sửa chữa được hệ thống (đối với sơ đồ hình thang), chứ không chỉ đối với người viết chương trình cho hệ thống. Tránh tình trạng người lập trình giữ bí mật chương trình của mình khiến cho những nhân viên kỹ thuật khác không thể hiểu để có thể vận hành và sửa chữa.

## 2.5 Bộ điều khiển logic lập trình PLC S7-200 của SIEMENS

S7-200 là thiết bị điều khiển logic khả trình loại nhỏ của hãng Siemens (CHLB Đức) có cấu trúc theo kiểu module và có các module mở rộng. Các module này sử dụng cho các ứng dụng lập trình khác nhau. Thành phần cơ bản của S7-200 là khối vi xử lý với các dòng CPU 212, CPU 214, CPU 215, CPU 216, CPU 221, CPU 222, CPU 224, CPU 224XP, CPU 226, CPU 226XP. Ứng với mỗi loại CPU sẽ có số lượng đầu ra, đầu vào hoặc khả năng mở rộng của các module khác nhau.

PLC S7-200 đáp ứng cho các ứng dụng vừa và nhỏ, mục đích nhằm giảm chi phí đầu tư thiết bị. Thực ra, S7-200 đáp ứng rất tốt các yêu cầu kỹ thuật về

điều khiển.

PLC S7-200 có khả năng kết nối với các mô đun mở rộng EM: vào ra số DI/DO, vào ra tương tự AI/AO, kết nối truyền thông, điều khiển động cơ bước, đo lường chính xác, ...

Có thể kết nối với màn hình giao diện như: TD 200, TP070.

Có truyền thông điều khiển converter.

Có tập hợp lệnh mạnh, dễ sử dụng.

Miễn phí phần mềm lập trình.

Có thể kết nối nhà lập trình thứ 3 thông qua thư viện hoặc ứng dụng được siemens hỗ trợ.

Hỗ trợ truyền thông mạnh: AS\_i, Internet, PPI, Profibus-DP, có thể truyền file theo giao thức FTP, có email client, truyền thông qua mạng GSM, ...

Có điều khiển ổn định vòng kín PID.

Kết nối được với S7-300, S7-400.

Mở rộng được thẻ nhớ, pin.

Có đồng hồ thời gian thực.



Hình 2.30. Module PLC S7-200 của SIEMENS.

### 2.5.1 Các thành phần cơ bản của PLC SIEMENS S7-200.



*Hình 2.31. Các thành phần cơ bản của PLC SIEMENS S7-200.*

Các led trạng thái(Status LEDs):

- **SF (đèn đỏ):** Đèn đỏ SF báo hiệu khi PLC có hỏng hóc.
- **RUN (đèn xanh):** Đèn xanh sáng báo hiệu PLC đang ở chế độ làm việc và thực hiện chương trình nạp ở trong máy.
- **STOP (đèn vàng):** Đèn vàng sáng báo hiệu PLC đang ở chế độ dừng, không thực hiện chương trình hiện có.

Các led I/O:

- **Ix.x (đèn xanh):** chỉ trạng thái logic tức thời của cổng Ix.x. đèn sáng tương ứng mức logic là 1.
- **Qx.x (đèn xanh):** chỉ trạng thái logic tức thời của cổng Ix.x. đèn sáng tương ứng mức logic là 1.

Công tắc chọn chế độ làm việc cho PLC:

Công tắc chọn chế độ nằm ở phía bên tay phải, có 3 vị trí cho phép chọn các chế độ làm việc khác nhau cho PLC.

- RUN: cho phép PLC thực hiện chương trình trong bộ nhớ. PLC S7-200 sẽ rời khỏi chế độ RUN và chuyển sang chế độ STOP nếu trong máy có sự cố, hoặc trong chương trình gặp lệnh STOP, thậm chí ngay cả khi công tắc ở chế độ RUN. Nên quan sát trạng thái thực tại của PLC theo đèn báo.

- STOP: cưỡng bức PLC dừng công việc thực hiện chương trình đang chạy và chuyển sang chế độ STOP. Ở chế độ STOP, PLC cho phép hiệu chỉnh lại chương trình hoặc nạp chương trình mới.

- TERM: cho phép máy lập trình tự quyết định một trong chế độ làm việc cho PLC ở RUN hoặc STOP.

### **Cổng truyền thông (Communicationsport)**

S7-200 sử dụng cổng truyền thông nối tiếp RS-485 với phích nối 9 chân để phục vụ cho việc ghép nối với thiết bị lập trình hoặc với các trạm PLC khác.

### **Module số (Terminalconnector)**

Đặc tính chung:

- Kích thước (DxRxC) (mm) : 90x80x62, 90x80x62, 120.5x80x62, 140x80x62, 190x80x62

- Công suất tiêu thụ : 2W.

Ngõ vào:

- Tầm khi ON từ 15-30VDC, dòng tối thiểu là 4mA. Bình thường, điện áp ngõ vào là 24VDC hoặc 220VAC.

- Thời gian đáp ứng: 3.5ms.

Ngõ ra:

- Tầm điện áp ngõ ra là 5-30VDC.

- Dòng tải ngõ ra tối đa là 2A.

- Thời gian trì hoãn ngõ ra là tối đa 10ms.

*Bảng 1. So sánh giữa các loại CPU S7-200.*

Feature	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Physical size (mm)	90 x 80 x 62	90 x 80 x 62	120.5 x 80 x 62	140 x 80 x 62	190 x 80 x 62
Program memory: with run mode edit without run mode edit	4096 bytes 4096 bytes	4096 bytes 4096 bytes	8192 bytes 12288 bytes	12288 bytes 16384 bytes	16384 bytes 24576 bytes
Data memory	2048 bytes	2048 bytes	8192 bytes	10240 bytes	10240 bytes
Memory backup	50 hours typical	50 hours typical	100 hours typical	100 hours typical	100 hours typical
Local on-board I/O Digital Analog	6 In/4 Out -	8 In/6 Out -	14 In/10 Out -	14 In/10 Out 2 In/1 Out	24 In/16 Out -
Expansion modules	0 modules	2 modules <sup>1</sup>	7 modules <sup>1</sup>	7 modules <sup>1</sup>	7 modules <sup>1</sup>
High-speed counters Single phase  Two phase	4 at 30 kHz  2 at 20 kHz	4 at 30 kHz  2 at 20 kHz	6 at 30 kHz  4 at 20 kHz	4 at 30 kHz 2 at 200 kHz 3 at 20 kHz 1 at 100 kHz	6 at 30 kHz  4 at 20 kHz
Pulse outputs (DC)	2 at 20 kHz	2 at 20 kHz	2 at 20 kHz	2 at 100 kHz	2 at 20 kHz
Analog adjustments	1	1	2	2	2
Real-time clock	Cartridge	Cartridge	Built-in	Built-in	Built-in
Communications ports	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485
Floating-point math	Yes				
Digital I/O image size	256 (128 in, 128 out)				
Boolean execution speed	0.22 microseconds/instruction				

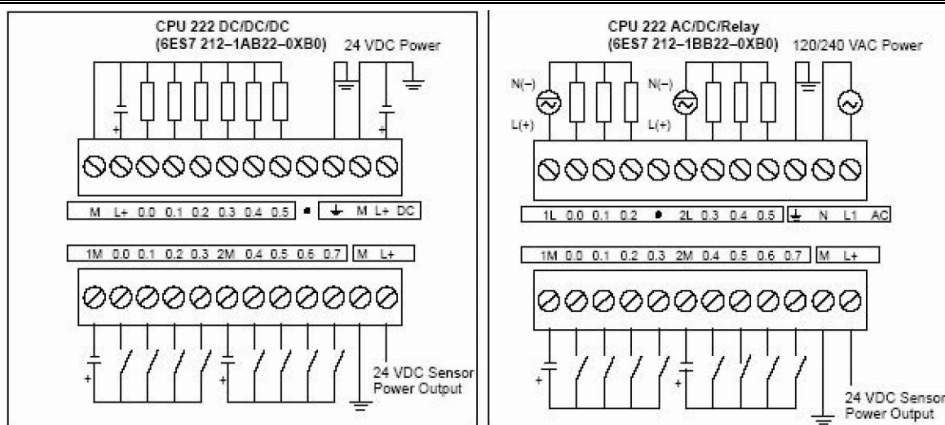
### 2.5.2 Thiết bị lập trình.

Thiết bị lập trình có thể là máy tính cá nhân (PC) có cài phần mềm Step7-Win32 hoặc bộ lập trình bằng tay chuyên dụng của hãng Siemens. Các máy tính cá nhân được lập cấu hình như các trạm làm việc phát triển chương trình. Các hệ thống cầm tay có bàn phím nhỏ, màn hình tinh thể lỏng. Các thiết bị lập trình cầm tay thường có đủ bộ nhớ để lưu giữ chương trình trong khi di chuyển từ vị trí này đến vị trí khác.

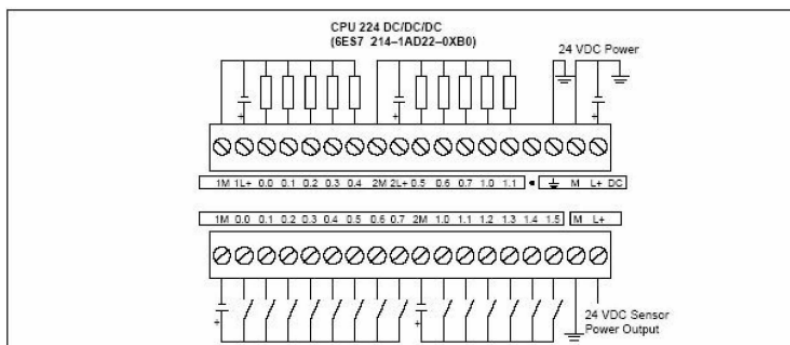
#### ➤ Cáp nối thiết bị lập trình và PLC.

Ghép S7-200 với máy lập trình qua cáp nối PPI. Cáp đó đi kèm theo máy lập trình.

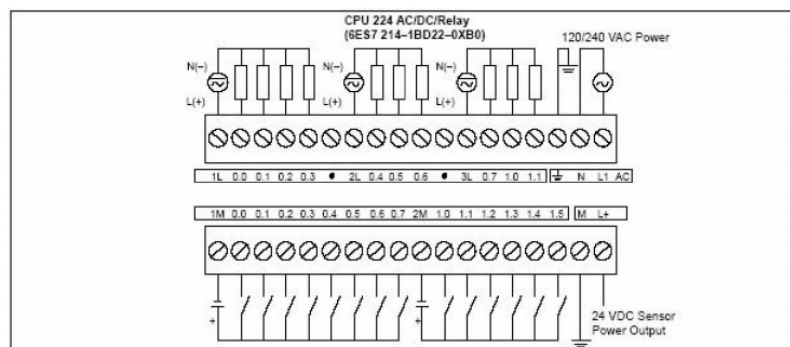
Ghép S7-200 với máy tính PC qua cổng RS-232 cần có cáp nối PC/PPI với bộ chuyển đổi RS232/RS485.



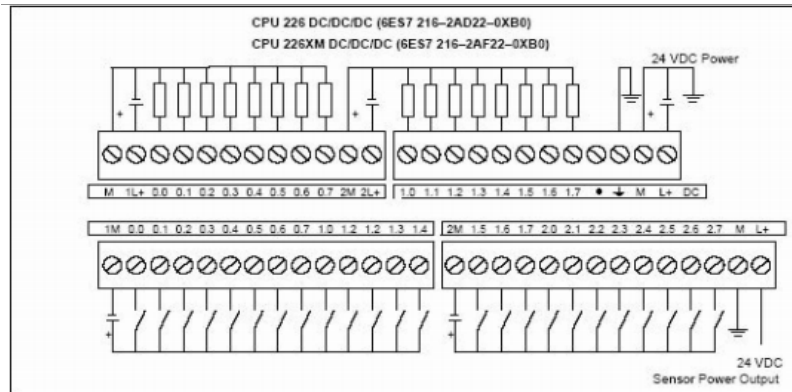
Sơ đồ đấu nối CPU221



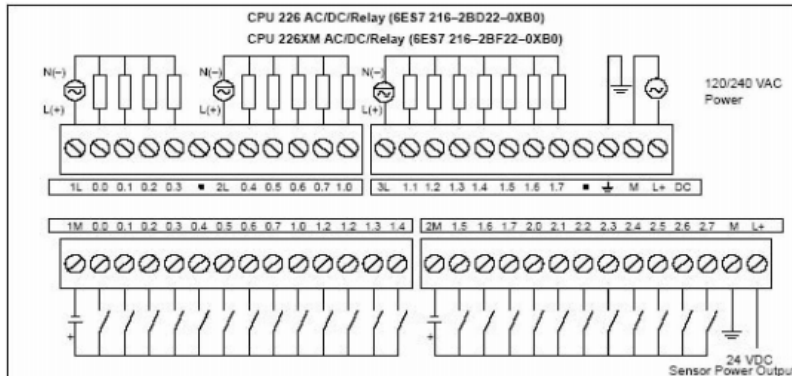
Sơ đồ đấu nối CPU224-DC



Sơ đồ đấu nối CPU224-AC



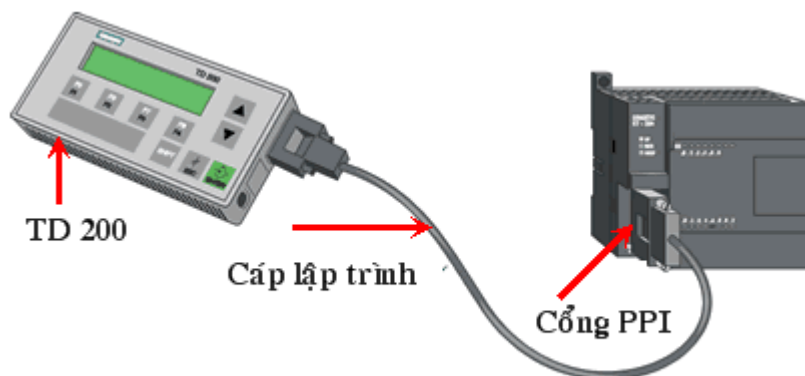
Sơ đồ đấu nối CPU226-DC



Sơ đồ đấu nối CPU226-AC

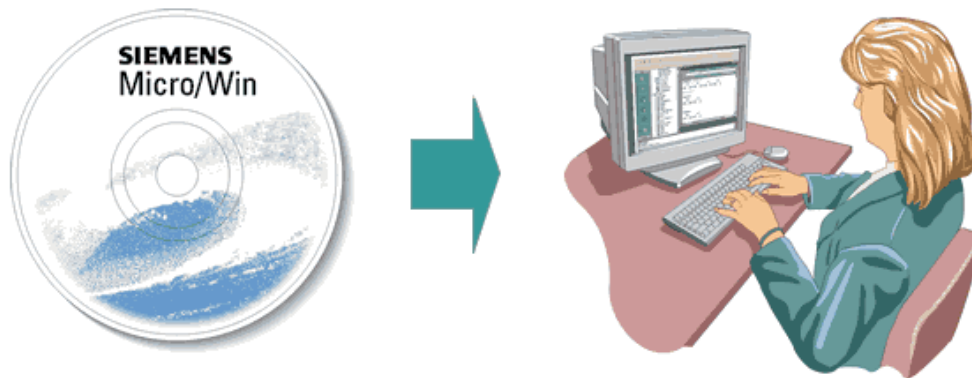
Hình 2.32. Sơ đồ đấu nối một số loại CPU của S7-200.

Để lập trình cho PLC S7-200 ta có thể dùng bộ lập trình tích hợp sẵn của Siemens hoặc chương trình Step7-Microwin được cài đặt trên máy tính.



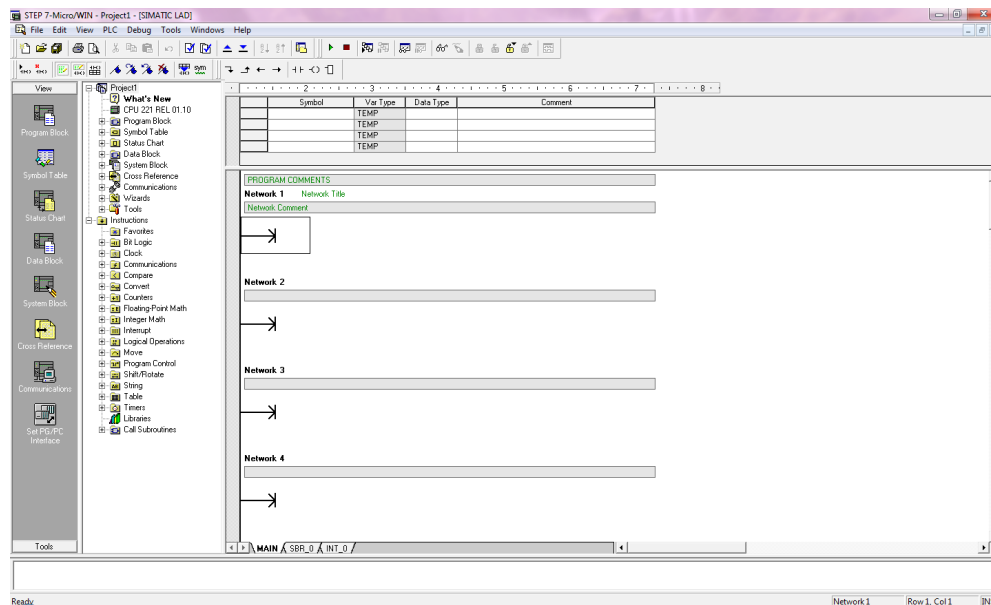
Bộ lập trình cầm tay TD 200 cho PLC S7-200

Hình 2.33. Bộ lập trình cầm tay TD 200.



Phần mềm lập trình cho bộ PLC SIEMENS S7-200

Hình 2.34 Lập trình thông qua chương trình Step7-Microwin



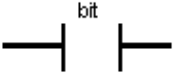
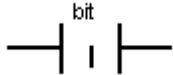
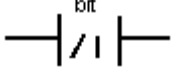

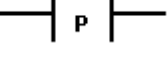
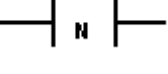
Hình 2.35 Giao diện chương trình Step7-Microwin.

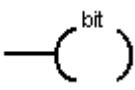
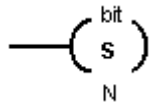
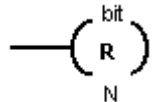
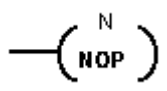
## 2.6 Một số tập lệnh cơ bản của PLC S7-200

### 2.6.1. Nhóm lệnh Bit logic.

Bảng 2. Nhóm lệnh Bit logic.

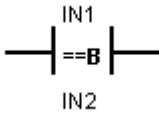
Dạng Lệnh		Mô tả chức năng lệnh
L		Tiếp điểm thường đóng sẽ mở khi giá trị logic bit bằng 1.
A		Toán hạng:
D		Bit : I, Q, M, SM, T, C, V(n)
STL	LDN n	

L		Tiếp điểm thường mở sẽ đóng khi giá trị logic bit bằng 1.
A		Toán hạng:
D		Bit : I, Q, M, SM, T, C, V(n)
STL	LD n	
L		Tiếp điểm thường mở sẽ đóng tức thời khi giá trị logic bit bằng 1.
A		Toán hạng:
D		Bit : I, Q, M, SM, T, C, V(n)
STL	LDI n	
L		Tiếp điểm thường đóng sẽ mở tức thời khi giá trị logic bit bằng 1.
A		Toán hạng:
D		Bit : I, Q, M, SM, T, C, V(n)
STL	LDNI n	
L		Tiếp điểm đảo trạng thái của dòng cung cấp. Nếu dòng cung cấp có tiếp điểm đảo thì nó ngắt mạch. Còn nếu không có tiếp điểm đảo thì nó thông mạch.
A		
D		
STL	NOT	
L		Tiếp điểm chuyển đổi dương cho phép dòng cung cấp thông mạch trong một vòng quét khi sườn xung điều khiển chuyển từ 0 lên 1.
A		
D		
STL	EU	
L		Tiếp điểm chuyển đổi âm cho phép dòng cung cấp thông mạch trong một vòng quét khi sườn xung điều khiển chuyển từ 1 xuống 0.
A		
D		
STL	ED	

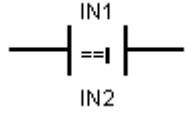
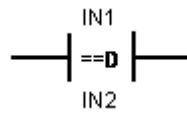
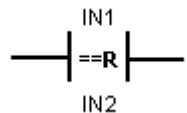
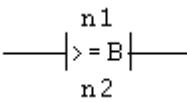
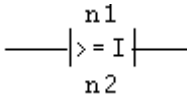
L A D		Cuộn dây đầu ra ở trạng thái kích thích khi có dòng điều khiển đi qua.
STL	= n	
L A D		Dùng để đóng một mảng gồm n tiếp điểm kể từ giá trị ban đầu bit.  Toán hạng:
STL	S bit n	Bit: I, Q, M, SM, T, C, V, IB, QB, MB, SMB, VB, AC, *VD, *AC, Constant.
L A D		Dùng để ngắt một mảng gồm n tiếp điểm kể từ giá trị ban đầu bit.  Toán hạng:
STL	R bit n	Bit: I, Q, M, SM, T, C, V, IB, QB, MB, SMB, VB, AC, *VD, *AC, Constant.
L A D		Lệnh này không có hiệu lực trong chương trình. Toán hạng:  N: Là một số từ 0 đến 255.
STL	NOP	

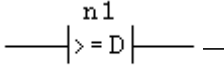
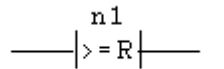
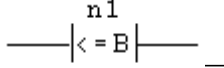
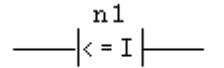
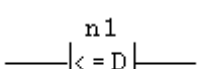
### 2.6.2. Nhóm lệnh so sánh.

Bảng 3. Nhóm lệnh so sánh.

Dạng Lệnh		Mô Tả Chức Năng Lệnh
L A D		Lệnh so sánh bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Byte) và ngược lại.  Toán hạng:



STL	LDB= IN1 IN2	IN1, IN2: VB, IB, QB, MB, SMB, AC, Constant, *VD, *AC.
L A D		Lệnh so sánh bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Word) và ngược lại.  Toán hạng:
STL	LDW= IN1 IN2	IN1, IN2: VW, IW, MW, SMW, AC, Constant, T, C, AIW, *VD, *AC.
L A D		Lệnh so sánh bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Double Word) và ngược lại.  Toán hạng:
STL	LDD= IN1 IN2	IN1, IN2: VD, ID, MD, SMD, AC, Constant, HC, *VD, *AC.
L A D		Lệnh so sánh bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Real) và ngược lại.  Toán hạng: IN1, IN2: VD, ID, QD, MD, SMD, AC, HC, *AC Constant, *VD.
STL	LDR= IN1 IN2	
L A D		Lệnh so sánh lớn hơn hoặc bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Byte)  Toán hạng: IN1, IN2: VB, IB, QB, MB, SMB, AC, Constant, *VD, *AC.
STL	LDB >= IN1 IN2	
L A D		Lệnh so sánh lớn hơn hoặc bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Word)  Toán hạng:
STL	LDW >= IN1 IN2	IN1, IN2: VW, IW, MW, SMW, AC, Constant, T, C, AIW, *VD, *AC.

L		Lệnh so sánh lớn hơn hoặc bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Dword)
A		
D		Toán hạng:
STL	LDD >= IN1 IN2	IN1, IN2: VD, ID, MD, SMD, AC, Constant, HC, *VD, *AC.
L		Lệnh so sánh lớn hơn hoặc bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Real)
A		
D		Toán hạng: IN1, IN2: VD, ID, QD, MD, SMD, AC, HC, *AC, Constant, *VD.
STL	LDR >= IN1 IN2	
L		Lệnh so sánh nhỏ hơn hoặc bằng sẽ làm cho tiếp điểm đóng IN1 bằng IN2 (IN1, IN2 kiểu Byte).
A		
D		Toán hạng: IN1, IN2: VB, IB, QB, MB, SMB, AC, Constant, *VD, *AC.
STL	LDB <= IN1 IN2	
L		Lệnh so sánh nhỏ hơn hoặc bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Word).
A		
D		Toán hạng:
STL	LDW <= IN1 IN2	IN1, IN2: VW, IW, MW, SMW, AC, Constant, T, C, AIW, *VD, *AC.
L		Lệnh so sánh nhỏ hơn hoặc bằng sẽ làm cho tiếp điểm đóng khi IN1 bằng IN2 (IN1, IN2 kiểu Dword).
A		
D		Toán hạng: IN1, IN2: VD, ID, MD, SMD, AC, Constant, HC, *VD, *AC.
STL	LDD <= IN1 IN2	

### 2.6.3. Lệnh nhảy và chương trình con.

- Các lệnh của chương trình, nếu không có những lệnh điều khiển riêng, sẽ được thực hiện theo thứ tự từ trên xuống dưới trong một vòng quét. Lệnh điều

chương trình cho phép thay đổi thứ tự thực hiện lệnh. Chúng cho phép nhảy đến một vị trí bất kỳ trong chương trình, trong đó nơi điều khiển chuyển đến phải được đánh dấu trước bằng một nhãn chỉ đích. Nhóm lệnh điều khiển chương trình gồm: lệnh nhảy, lệnh gọi chương trình con. Nhãn chỉ đích hay gọi đơn giản là nhãn phải được đánh dấu trước khi thực hiện lệnh nhảy hay lệnh gọi chương trình con.

- Việc đặt nhãn cho lệnh nhảy phải nằm trong chương trình. Nhãn của chương trình con hoặc trong chương trình xử lý ngắt được khai báo ở đầu chương trình. Không thể dùng lệnh nhảy JMP để chuyển điều khiển từ chương trình chính vào một nhãn bất kỳ trong chương trình con hoặc trong chương trình xử lý ngắt. Tương tự như vậy, cũng không thể từ một chương trình con hay chương trình xử lý ngắt nhảy vào bất cứ một nhãn nào nằm ngoài các chương trình đó.

- Lệnh gọi chương trình con là lệnh chuyển điều khiển đến chương trình con. Khi chương trình con thực hiện xong các phép tính của mình thì việc điều khiển lại được chuyển trở về lệnh tiếp theo trong chương trình chính (ngay sau lệnh gọi chương trình con). Từ một chương trình con có thể gọi được một chương trình con khác trong nó, có thể gọi như vậy nhiều nhất là tám lần trong S7\_200. Để quy (trong một chương trình con có lệnh gọi đến chính nó) về nguyên tắc không bị cấm song phải để ý đến giới hạn trên.

- Nếu lệnh nhảy hay lệnh gọi chương trình con được thực hiện thì đỉnh ngăn xếp luôn có giá trị logic 1. Bởi vậy trong chương trình con các lệnh có điều kiện được thực hiện như các lệnh không điều kiện. Sau các lệnh LBL (đặt nhãn) và SBR, lệnh LD trong STL sẽ bị vô hiệu hoá.

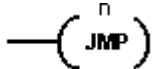
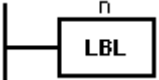
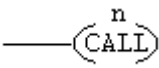
- Khi một chương trình con được gọi, toàn bộ nội dung của ngăn xếp sẽ được cất đi, đỉnh của ngăn xếp nhận giá trị logic mới là 1, các bit khác của ngăn xếp nhận giá trị logic 0 và điều khiển được chuyển đến chương trình con đã được gọi. Khi thực hiện xong chương trình con và trước khi điều khiển được chuyển trở lại chương trình đã gọi nó, nội dung ngăn xếp đã được cất giữ trước đó sẽ được chuyển trở lại ngăn xếp.

- Nội dung của thanh ghi AC không được cất giữ khi gọi chương trình con nhưng khi một chương trình xử lý ngắt được gọi, nội dung của thanh ghi AC sẽ được cất giữ trước khi thực hiện chương trình xử lý ngắt và sẽ được nạp lại khi chương trình xử lý ngắt đã được thực hiện xong. Bởi vậy, chương trình xử lý ngắt có thể tự do sử dụng bốn thanh ghi AC của S7\_200.

- Lệnh nhảy JMP và lệnh gọi chương trình con SBR cho phép chuyển điều khiển từ vị trí này đến vị trí khác trong chương trình. Cú pháp của lệnh nhảy và lệnh gọi chương trình con trong LAD và STL đều có toán hạng là nhãn chỉ đích (nơi nhảy đến, nơi chứa chương trình con).

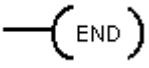


- Lệnh nhảy, lệnh gọi chương trình con, lệnh khai báo nhãn và lệnh thoát khỏi chương trình con được biểu diễn trong LAD và trong STL như sau:

Bảng 4. Nhóm lệnh nhảy.

Dạng Lệnh		Mô Tả Chức Năng Lệnh
L A D		Lệnh nhảy thực hiện việc chuyển điều khiển đến nhãn n trong một chương trình  n: 0-255
STL	JMP Kn	
L A D		Lệnh khai báo nhãn n trong một chương trình.  n: 0-255
STL	LBL Kn	
L A D		Lệnh gọi chương trình con, thực hiện phép chuyển điều khiển đến chương trình con có nhãn là n.  n: 0-255
STL	CALL Kn	

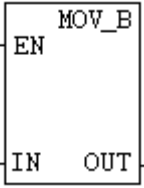
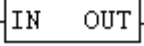
#### 2.6.4 Nhóm lệnh can thiệp vào thời gian vòng quét.

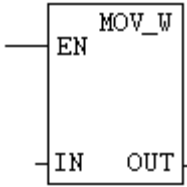
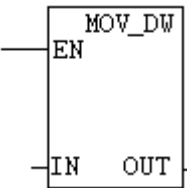
Bảng 5. Nhóm lệnh can thiệp vào thời gian vòng quét.

Dạng Lệnh		Mô Tả Chức Năng Lệnh
L A D		Lệnh END dùng để kết thúc chương trình chính hiện hành có điều kiện.
STL	END	
L A D		Lệnh STOP dùng để dừng chương trình hiện hành và chuyển sang chế độ STOP
STL	STOP	
L A D		Lệnh WDR dùng để khởi tạo lại đồng hồ quan sát.
STL	WDR	

#### 2.6.5 Nhóm lệnh dịch chuyển nội dung ô nhớ.

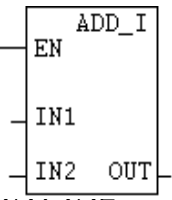
Bảng 6. Nhóm lệnh dịch chuyển nội dung ô nhớ.

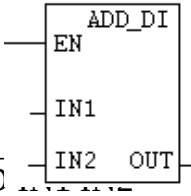
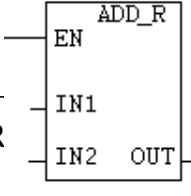
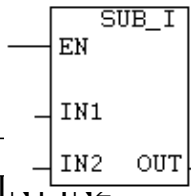
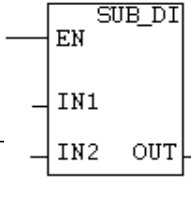
Dạng Lệnh		Mô Tả Chức Năng Lệnh
L A D		Lệnh sao chép nội dung của byte IN sang byte OUT.  Toán hạng: IN: VB, IB, QB, MB, SMB, SB, AC, Constant, *VD,*AC.OUT:VB, IB, QB, MB, SMB, SB, AC, *VD, *AC.
STL	M  T	

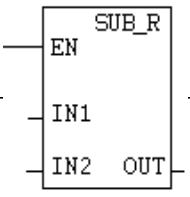
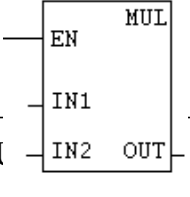
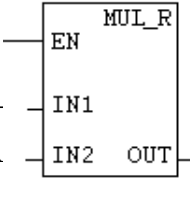
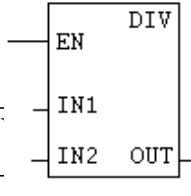
L A D		Lệnh sao chép nội dung của từ đơn IN sang từ đơn OUT. Toán hạng: IN: VW, T, C, IW, QW, MW, SMW, AC, AIW, Constant, *VD, *AC. OUT: VW, T, C, IW, QW, MW, SMW, SW, AC, AQW, *VD, *AC.
STL	M	JT
L A D		Lệnh sao chép nội dung của từ kép IN sang từ kép OUT. Toán hạng: IN: VD, ID, QD, MD, SMD, HC, *VD, *AC, &VB, &IB, &QB, &MB, &T, &C, &SB, Constant. OUT: VD, ID, QD, MD, SD, SMD, AC, *VD, *AC.
STL	M	JT

### 2.6.6 Nhóm lệnh số học.

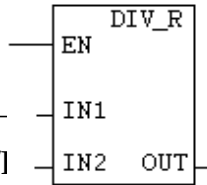
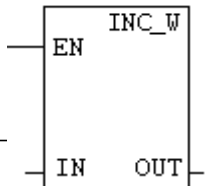
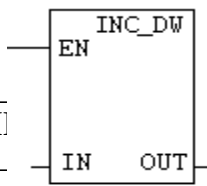
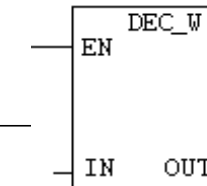
Bảng 7. Nhóm lệnh số học.

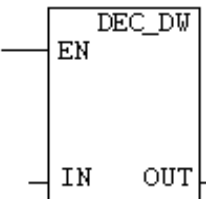
Dạng Lệnh		Mô Tả Chức Năng Lệnh
L A D		Lệnh thực hiện phép cộng hai số nguyên 16 bit IN1 và IN2. Kết quả là một số nguyên 16 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2. Toán hạng: IN1, IN2 :VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constan, *VD, *AC. OUT :VW, T, C, IW, QW, MW, SMW, SW, AC, *VD, *AC.
STL	+I	

L A D		Lệnh thực hiện phép cộng hai số nguyên 32 bit IN1 và IN2. Kết quả là một số nguyên 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.  Toán hạng: IN1, IN2: VD, ID, QD, MD, SMD, SD, AC, HC, Constant, *VD, *AC.  OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.
STL	+D	
L A D		Lệnh thực hiện phép cộng hai số thực 32 bit IN1 và IN2. Kết quả là một số thực 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.  Toán hạng: IN1, IN2: VD, ID, QD, MD, SMD, SD, AC, Constant, *VD, *AC. OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.
STL	+R	
L A D		Lệnh thực hiện phép trừ hai số nguyên 16 bit IN1 và IN2. Kết quả là một số nguyên 16 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.  Toán hạng: IN1, IN2: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.  OUT : VW, T, C, IW, QW, MW, SMW, SW, AC, *VD, *AC.
STL	-I	
L A D		Lệnh thực hiện phép trừ hai số nguyên 32 bit IN1 và IN2. Kết quả là một số nguyên 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.  Toán hạng: IN1, IN2: VD, ID, QD, MD, SMD, SD, AC, HC, Constant, *VD, *AC.  OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.
STL	-D	

L A D			<p>Lệnh thực hiện phép trừ hai số thực 32 bit IN1 và IN2. Kết quả là một số thực 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.</p> <p>Toán hạng: IN1, IN2: VD, ID, QD, MD, SMD, SD, AC, Constan, *VD, *AC.</p> <p>OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.</p>
STL	-R		
L A D			<p>Lệnh thực hiện phép nhân hai số nguyên 16 bit IN1 và IN2. Kết quả là một số 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.</p> <p>Toán hạng: IN1, IN2: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC. OUT : VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.</p>
STL	MI		
L A D			<p>Lệnh thực hiện phép nhân hai số thực 32 bit IN1 và IN2. Kết quả là một số 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.</p> <p>Toán hạng: IN1, IN2: VD, ID, QD, MD, SMD, SD, AC, Constan, *VD, *AC. OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.</p>
STL	*R		
L A D			<p>Lệnh thực hiện phép chia hai số nguyên 16 bit IN1 và IN2. Kết quả là một số 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.</p> <p>Toán hạng: IN1, IN2: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.</p> <p>OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.</p>
STL	DI		



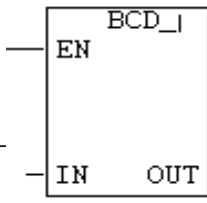
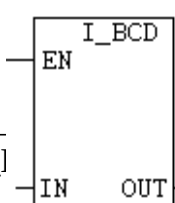
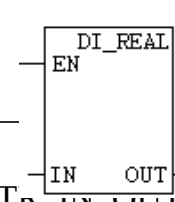
L A D		Lệnh thực hiện phép chia hai số thực 32 bit IN1 và IN2. Kết quả là một số 32 bit và được ghi vào OUT. Trong STL thì kết quả được ghi vào IN2.
STL	/I	Toán hạng: IN1, IN2: VD, ID, QD, LD, SD, SMD, AC, Constant, *VD, *AC.  OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.
L A D		Lệnh tăng giá trị của từ đơn IN lên một đơn vị. Kết quả được ghi vào OUT. Trong STL kết quả được ghi vào IN.
STL	INCW IN	Toán hạng: IN: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.  OUT: VW, T, C, IW, QW, MW, SMW, SW, AC, *VD, *AC.
L A D		Lệnh tăng giá trị của từ kép IN lên một đơn vị. Kết quả được ghi vào OUT. Trong STL kết quả được ghi vào IN.
STL	I	Toán hạng: IN: VD, ID, QD, MD, SMD, SD, AC, HC, Constant, *VD, *AC.  OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.
L A D		Lệnh giảm giá trị của từ đơn IN xuống một đơn vị. Kết quả được ghi vào OUT. Trong STL kết quả được ghi vào IN.
STL	DECW IN	Toán hạng: IN: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.  OUT: VW, T, C, IW, QW, MW, SMW, SW, AC, *VD, *AC.

L			Lệnh giảm giá trị của từ kép IN lên một đơn vị. Kết quả được ghi vào OUT. Trong STL kết quả được ghi vào IN.
A			Toán hạng: IN: VD, ID, QD, MD, SMD, SD, AC, HC, Const, *VD, *AC.
D			OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.
STL	I		

### 2.6.7 Nhóm lệnh biến đổi kiểu dữ liệu.

Nhóm lệnh biến đổi kiểu dữ liệu cho phép thực hiện các phép biến đổi dữ liệu từ dạng này sang dạng khác.

Bảng 8. Nhóm lệnh biến đổi kiểu dữ liệu.

Dạng Lệnh		Mô Tả Chức Năng Lệnh
L A D		<p>Lệnh thực hiện phép biến đổi một số thuộc hệ nhị thập phân 16 bit thành số nguyên 16 bit và ghi kết quả vào OUT.</p> <p>Toán hạng:</p> <p>IN: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.</p> <p>OUT: VW T,C, IW, QW, MW, SMW, SW, AC, *VD, *AC.</p>
STL	BCD_I OUT	
L A D		<p>Lệnh thực hiện phép biến đổi một số nguyên 16 bit thành số nhị thập phân 16 bit và ghi kết quả vào OUT.</p> <p>Toán hạng: IN: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.</p> <p>OUT: VW, T, C, IW, QW, MW, SMW, SW, AC, *VD, *AC.</p>
STL	I_	
L A D		<p>Lệnh thực hiện phép biến đổi một số nguyên 32 bit có dấu IN thành số thực 32 bit và ghi vào OUT.</p> <p>Toán hạng:</p> <p>IN: VD, ID, QD, MD, SMD, SD, AC, HC, Constant, *VD, *AC.</p> <p>OUT: VD, ID, QD, MD, SMD, SD, AC, *VD, *AC.</p>
STL	DI_REAL OUT	

## 2.7 Giới Thiệu Về Timer Và Các Lệnh Điều Khiển Timer.

- Timer là bộ tạo thời gian trễ giữa tín hiệu vào và tín hiệu ra nên trong điều khiển thường được gọi là khâu trễ. S7-200 (CPU 224) có 256 Timer và được chia làm hai loại khác nhau, đó là:

+ Timer tạo thời gian trễ không có nhớ (On – Delay Timer), ký hiệu là TON.

+ Timer tạo thời gian trễ có nhớ (Retentive On – Delay Timer), ký hiệu là TONR.

- Hai kiểu Timer của S7-200 (TON và TONR) phân biệt với nhau ở phản ứng của nó đối với trạng thái tín hiệu đầu vào.

- Cả hai Timer kiểu TON và TONR cùng bắt đầu tạo thời gian trễ tín hiệu kể từ thời điểm có sườn lên ở tín hiệu đầu vào, tức là khi tín hiệu đầu vào chuyển trạng thái logic từ 0 lên 1, được gọi là thời điểm Timer được kích, và không tính khoảng thời gian khi đầu vào có giá trị logic 0 vào thời gian trễ tín hiệu được đặt trước.

- Khi đầu vào có giá trị logic bằng 0, TON tự động Reset còn TONR thì không tự Reset. Timer TON được dùng để tạo thời gian trễ trong một khoảng thời gian, còn với TONR thời gian trễ sẽ được tạo trong nhiều khoảng thời gian khác nhau.

- Timer TON và TONR bao gồm 3 loại với 3 độ phân giải khác nhau, độ phân giải 1ms, 10 ms và 100 ms. Thời gian trễ được tạo ra chính là tích của độ phân giải của bộ Timer được chọn và giá trị đặt trước cho Timer.

- Timer Của S7\_200 Có Những Tính Chất Cơ Bản Như Sau:

- Các bộ Timer được điều khiển bởi một cổng vào và giá trị đếm tức thời. Giá trị đếm tức thời của Timer được nhớ trong thanh ghi 2 byte (gọi là T-word) của Timer, xác định khoảng thời gian trễ kể từ khi Timer được kích. Giá trị đặt trước của các bộ Timer được ký hiệu trong LAD và STL là PT. Giá trị đếm tức

thời của thanh ghi T-word thường xuyên được so sánh với giá trị đặt trước của Timer.

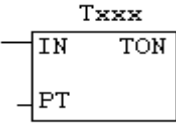
- Mỗi bộ Timer, ngoài thanh ghi 2 byte T-word lưu giá trị đếm tức thời, còn có một bit, ký hiệu bằng T-bit, chỉ thị trạng thái logic đầu ra. Giá trị logic của bit này phụ thuộc vào kết quả so sánh giữa giá trị đếm tức thời với giá trị đặt trước.

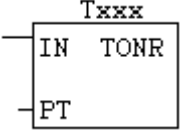
- Trong khoảng thời gian tín hiệu x (t) có giá trị logic 1, giá trị đếm tức thời trong T-word luôn được cập nhật và thay đổi tăng dần cho đến khi nó đạt giá trị cực đại. Khi giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước, T-bit có giá trị logic 1.

- Các loại Timer của S7-200 (đối với CPU 214) chia theo TON, TONR và độ phân giải bao gồm:

Lệnh	Độ phân giải	Giá trị cực đại	Tên Timer
TON	1 ms	32,767 s	T32;T96
	10 ms	327,67 s	T33→T36;T97→T100
	100 ms	3276,7 s	T37→T63;T101→T255
TONR	1 ms	32,767 s	T0;T64
	10 ms	327,67 s	T1→T4; T65→T68
	100 ms	3276,7 s	T5→T31; T69→T95

Bảng 9. Lệnh Khai Báo Sử Dụng Timer.

Dạng Lệnh		Mô Tả Chức Năng Lệnh
L A D		<p>Khai báo Timer số hiệu xxx kiểu TON để tạo thời gian trễ tính từ khi giá trị đầu vào IN được kích. Nếu giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước PT thì T-bit có giá trị bằng 1. Có thể Reset Timer kiểu TON bằng lệnh R hoặc bằng giá trị logic 0 tại đầu vào IN.</p> <p>Toán hạng:</p>

	STL	TON T <sub>xxx</sub> PT	PT: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.
L	A D		Khai báo Timer số hiệu xxx kiểu TONR để tạo thời gian trễ tính từ khi giá trị đầu vào IN được kích. Nếu giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước PT thì T-bit có giá trị bằng 1. Chỉ có thể Reset Timer kiểu TONR bằng lệnh R.
	STL	TONR T <sub>xxx</sub> PT	Toán hạng:  PT: VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.

## 2.8 Giới Thiệu Về Counter Và Các Lệnh Điều Khiển Counter.

- Counter là bộ đếm hiện chức năng đếm sườn xung trong S7-200. Các bộ đếm của S7-200 được chia ra làm 2 loại: bộ đếm tiến (CTU) và bộ đếm tiến / lùi (CTUD).

+ Bộ đếm tiến (CTU) đếm số sườn lên của tín hiệu logic đầu vào, tức là đếm số lần thay đổi trạng thái logic từ 0 lên 1 của tín hiệu. Số sườn xung đếm được được ghi vào thanh ghi 2 byte của bộ đếm, gọi là thanh ghi C-word.

\* Nội dung của C-word, gọi là giá trị đếm tức thời của bộ đếm luôn được so sánh với giá trị đặt trước của bộ đếm và được ký hiệu là PV. Khi giá trị đếm tức thời bằng hoặc lớn hơn giá trị đặt trước này thì bộ đếm báo ra ngoài bằng cách đặt giá trị logic 1 vào một bit đặc biệt của nó, được gọi là C-bit. Trường hợp giá trị đếm tức thời nhỏ hơn giá trị đặt trước thì C-bit có giá trị logic là 0.

\* Khác với các bộ Counter, các bộ đếm CTU đều có chân nối với tín hiệu điều khiển xóa để thực hiện việc đặt lại chế độ khởi phát ban đầu (Reset) cho bộ đếm, được ký hiệu bằng chữ cái R trong LAD hay được qui định là trạng thái logic của bit đầu tiên của ngăn xếp trong STL. Bộ đếm được Reset khi tín hiệu xóa này có mức logic là 1 hoặc khi lệnh R (Reset) được thực hiện với C-bit. Khi bộ đếm được Reset, cả C-word và C-bit đều nhận giá trị 0.

+ Bộ đếm tiến / lùi (CTUD) đếm tiến khi gặp sườn lên của xung vào cổng đếm, ký hiệu là CU trong LAD hoặc bit thứ 3 của ngăn xếp trong STL, và đếm lùi khi gặp sườn lên của xung vào cổng đếm lùi, được ký hiệu là CD trong LAD hoặc bit thứ 2 của ngăn xếp trong STL.

\* Giống như bộ đếm CTU, bộ đếm CTUD cũng được đưa về trạng thái khởi phát ban đầu bằng 2 cách:

+ Khi đầu vào logic của chân xóa, ký hiệu bằng R trong LAD hoặc bit thứ nhất của ngăn xếp trong STL có giá trị logic là 1.

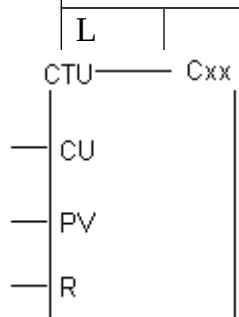
+ Bằng lệnh R (reset) với C-bit của bộ đếm.

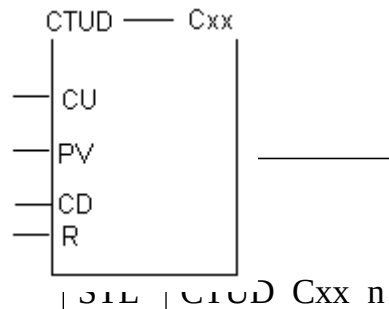
\* CTUD có giá trị đếm tức thời đúng bằng giá trị đang đếm và được lưu trong thanh ghi 2 byte C-word của bộ đếm. Giá trị đếm tức thời luôn được so sánh với giá trị đặt trước PV của bộ đếm. Nếu giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước thì C-bit có giá trị logic bằng 1. Còn các trường hợp khác C-bit có giá trị logic bằng 0.

- Bộ đếm tiến CTU có miền giá trị đếm tức thời từ 0 đến 32.767. Bộ đếm tiến/ lùi CTUD có miền giá trị đếm tức thời là  $-32.767 \div 32.767$ .

- Các bộ đếm được đánh số từ 0 đến 255 (đối với CPU 224) và ký hiệu bằng Cxx, trong đó xx là số thứ tự của bộ đếm. Ký hiệu Cxx đồng thời cũng là địa chỉ hình thức của C-word và của C-bit. Mặc dù dùng địa chỉ hình thức, song C-word và C-bit vẫn được phân biệt với nhau nhờ kiểu lệnh sử dụng làm việc với từ hay với tiếp điểm (bit).

Lệnh Khai Báo Sử Dụng Bộ Đếm Trong LAD Như Sau:

Dạng Lệnh		Mô Tả Chức Năng Lệnh
		Khai báo bộ đếm tiến theo sườn lên của CU. Khi giá trị đếm tức thời C-word Cxx lớn hơn hoặc bằng giá trị đặt trước PV, C-bit (cxx) có giá trị logic bằng 1 và ngừng đếm khi C-word Cxx đạt

STL	CTU Cxx n	<p>giá trị cực đại 32.767. Bộ đếm được Reset khi đầu vào R có giá trị logic bằng 1.</p> <p>Toán hạng:</p> <p>PV: VW, T, C, IW, QW, MW, SMW, AC, AIW, Constant, *VD, *AC.</p>
L A	 <p>CTUD Cxx n</p>	<p>Khai báo bộ đếm tiến / lùi, đếm tiến theo sườn lên của CU và đếm lùi theo sườn lên của CD. Khi giá trị đếm tức thời C-word Cxx lớn hơn hoặc bằng giá trị đặt trước PV, C-bít (cxx) có giá trị logic bằng 1. Bộ đếm ngừng đếm tiến khi C-word đạt giá trị cực đại 32.767 và ngừng đếm lùi khi C-word đạt giá trị cực tiểu 32.767. CTUD được Reset khi đầu vào R có giá trị logic bằng 1.</p> <p>Toán hạng:</p> <p>PV: VW, T, C, IW, QW, MW, SMW, AC, AIW, Constant, *VD, *AC.</p>



## **Chương 3: THIẾT KẾ VÀ THI CÔNG THANG MÁY 4 TẦNG**

### **3.1 Thiết kế mô hình thang máy 4 tầng.**

Mô hình thang máy 4 tầng mô phỏng gần giống hệ thống thang máy thực gồm các bộ phận sau:

- Động cơ kéo thang máy : đặt ở trên cùng của thang máy.
- Động cơ đóng mở cửa cabin : đặt trên cabin.
- Cabin được treo bằng sợi cáp vắt qua hệ thống ròng rọc và pulley của Motor kéo, phần còn lại của sợi cáp được nối với đối trọng.
- Cabin và đối trọng trượt trên các thanh ray.

Khung mô hình được làm bằng sắt để tiết kiệm chi phí và đảm bảo tính vững chắc khi vận hành.

### **3.2 Thi công mô hình.**

Mô hình được thiết kế và thi công trong suốt quá trình làm đồ án, các chi tiết của mô hình đã hoàn thành đúng tiến độ. Trong thời gian thực hiện mô hình tuy gặp phải những trục trặc về kỹ thuật nhưng mô hình đã kịp thời được đưa vào vận hành thử trước thời gian bảo vệ đồ án.

Mô hình sau khi hoàn chỉnh đã vận hành thử và hoạt động tốt.

Sau đây là mô hình đã được hoàn thành.



Hình 3.1 Mô hình thang máy 4 tầng



*Hình 3.2 Mô hình chi tiết Cabin, đối trọng, ray dẫn hướng các thiết bị khác.*



*Hình 3.3 Hình PLC, Motor DC và các thiết bị khác.*

### 3.2 Các thành phần chính của mô hình

#### 3.2.1 Khối nguồn

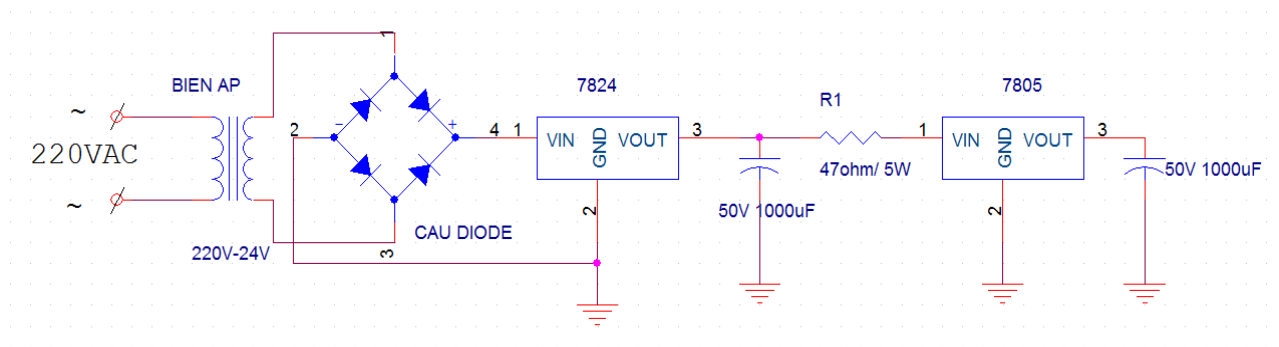
Phần nguồn được thiết kế để cung cấp đủ công suất cho động cơ DC, mạch điều khiển gồm Role và cho khối hiển thị.

Các linh kiện trong phần nguồn:

Biến thế 5A đưa điện thế 24V cấp cho motor.

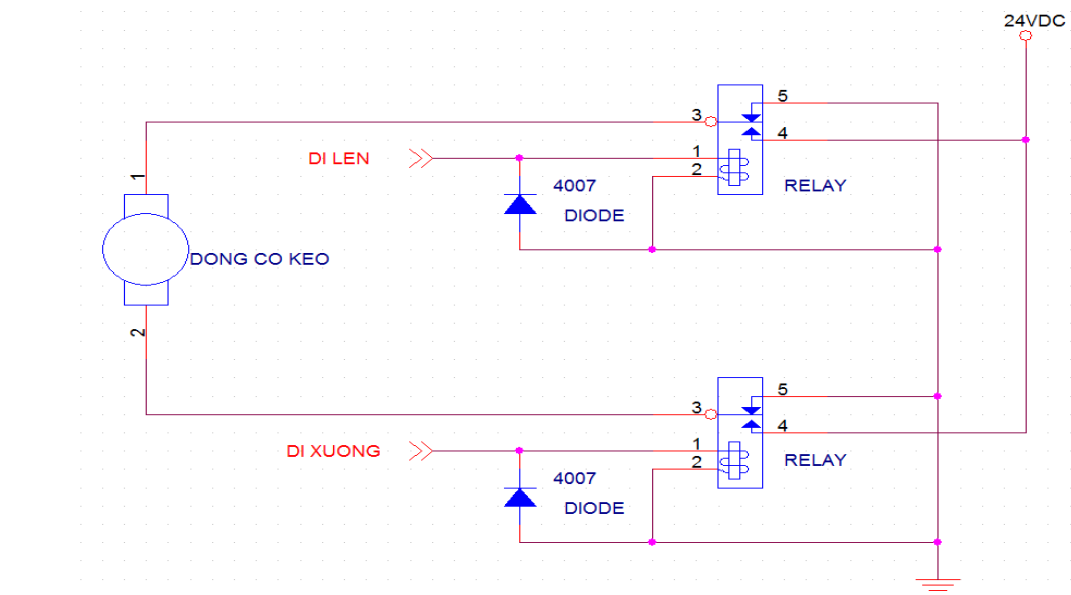
Tụ lọc nguồn 220 $\mu$ F/ 50V

Mạch cấp nguồn 5-24VDC

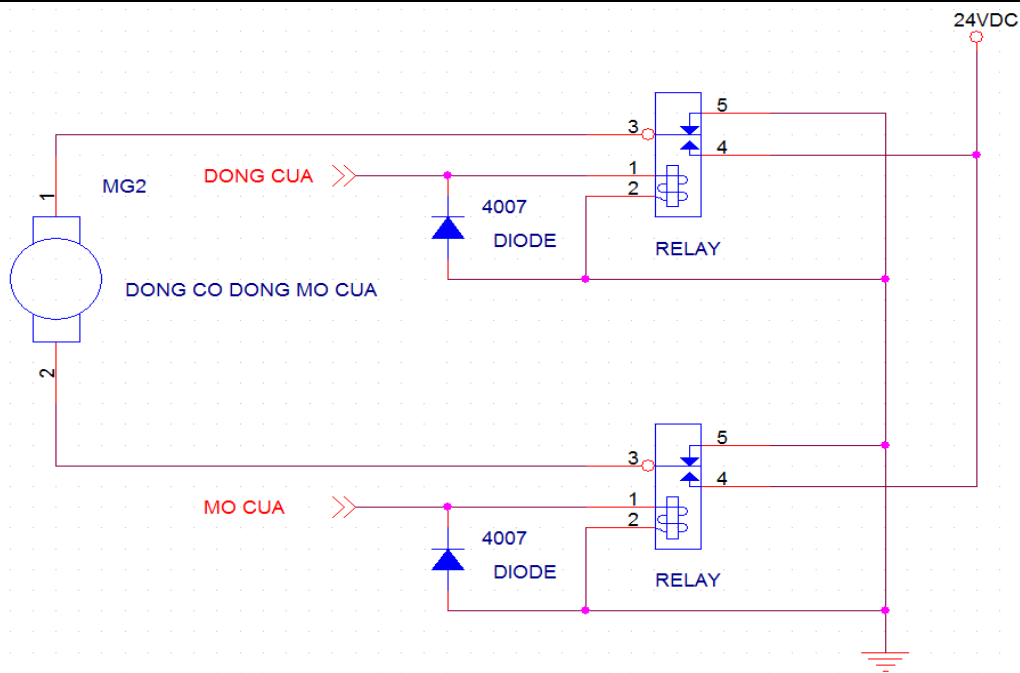


Hình 3.4 Bộ biến áp, chỉnh lưu dòng và ổn áp dòng

#### 3.2.2 Mạch đảo chiều động cơ.



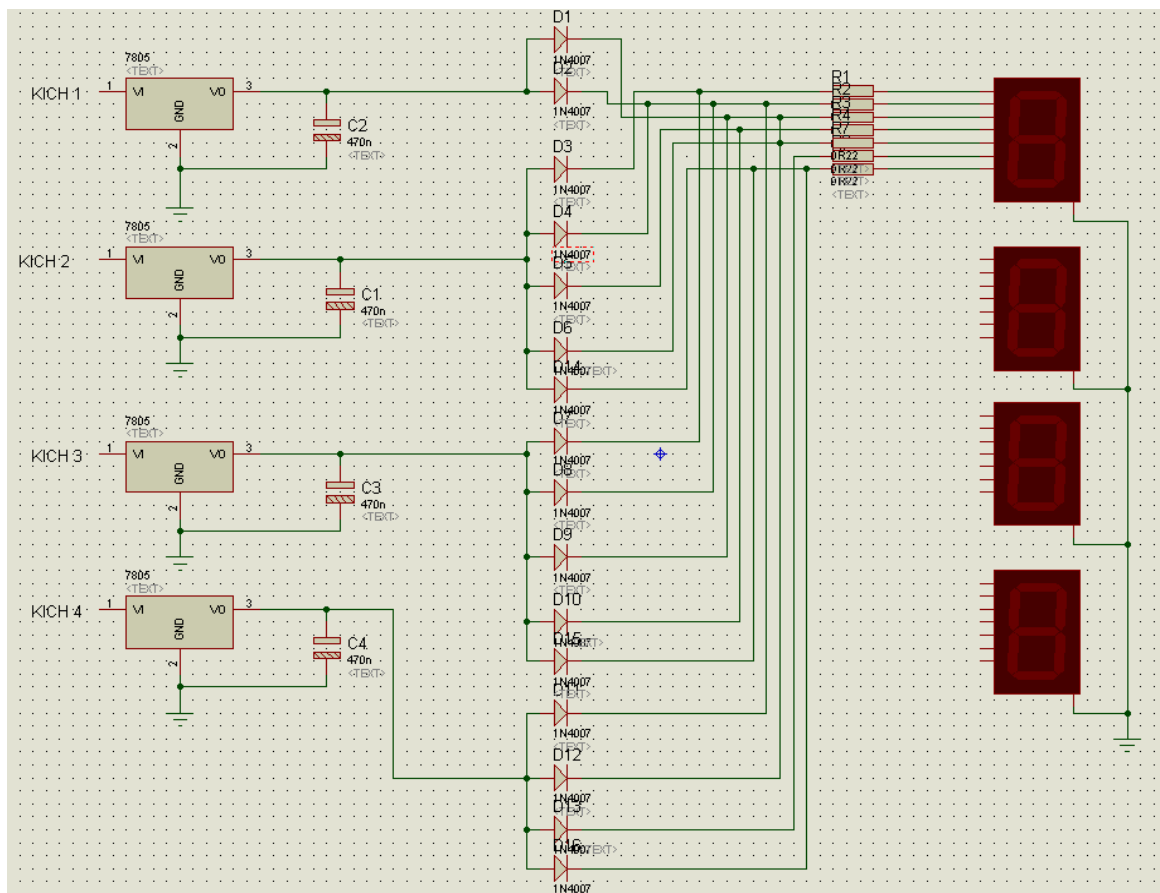
Hình 3.5 Mạch đảo chiều động cơ điều khiển thang máy lên-xuống



Hình 3.6 Mạch đảo chiều động cơ điều khiển thang máy đóng-mở cửa

### 3.2.3 Khối hiển thị

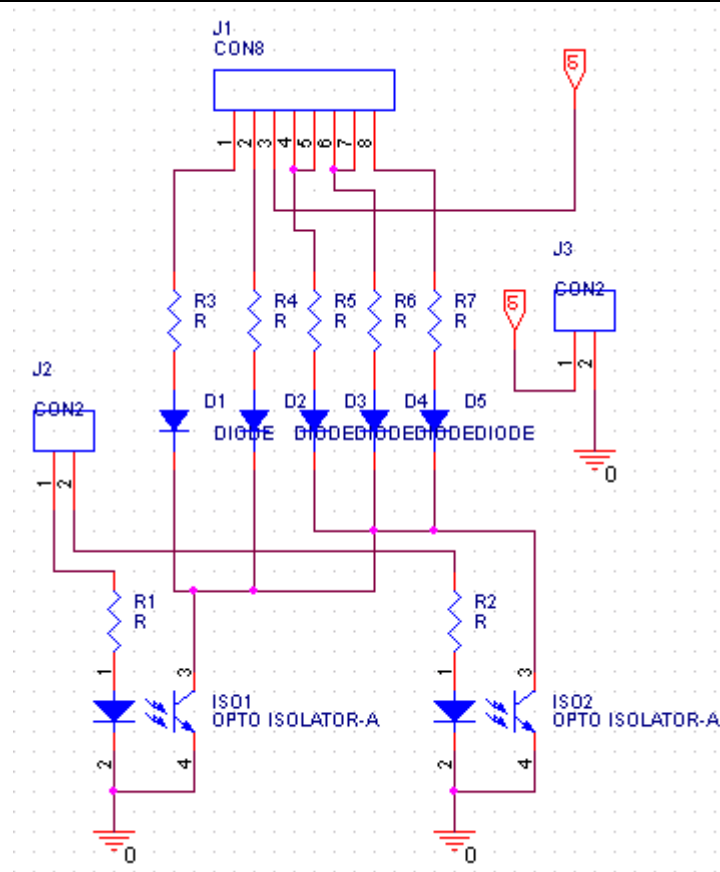
- ❖ Mạch hiển thị tầng dùng led 7 đoạn.



Hình 3.7. Mạch hiển thị tầng dùng led 7 đoạn.

- ❖ Mạch hiển thị trạng thái lên-xuống của cabin.



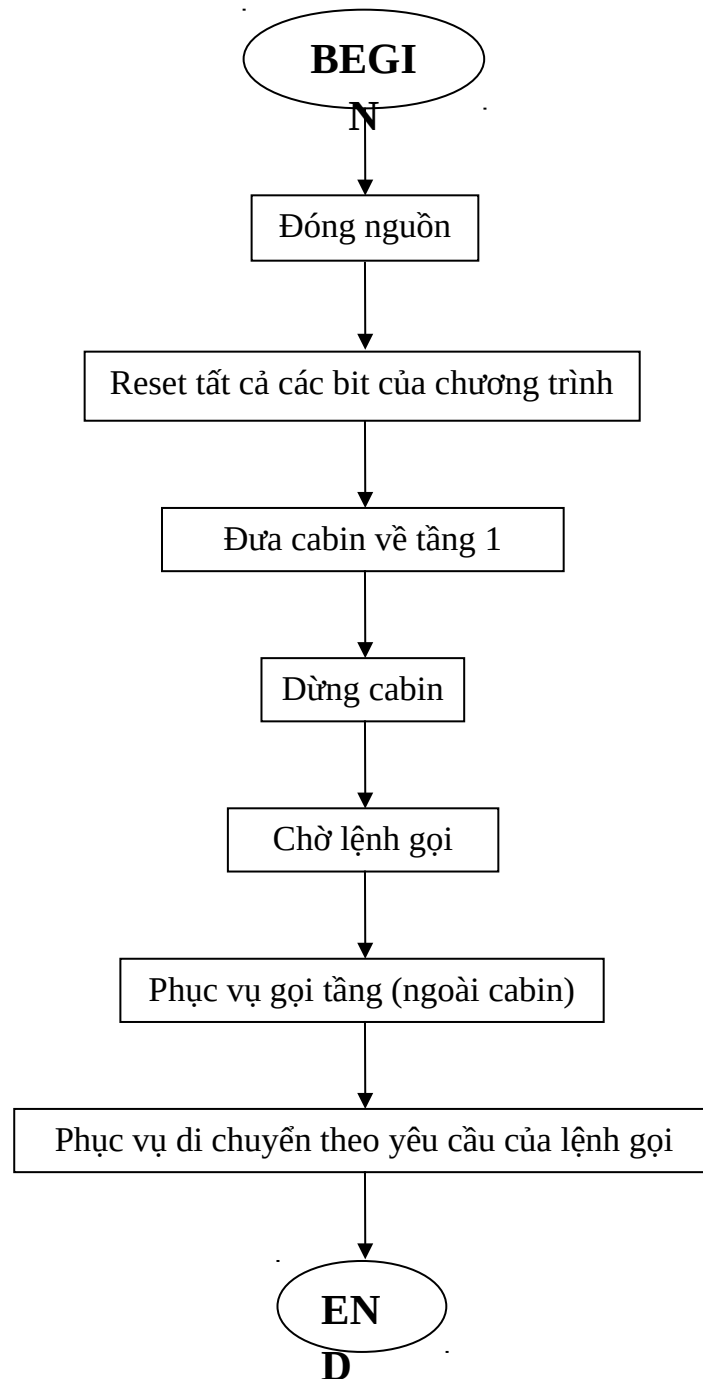


Hình 3.8. Mạch hiển thị lên-xuống dùng led dạng mũi tên.

## Chương 4: THIẾT KẾ CHƯƠNG TRÌNH ĐIỀU KHIỂN THANG MÁY 4 TẦNG

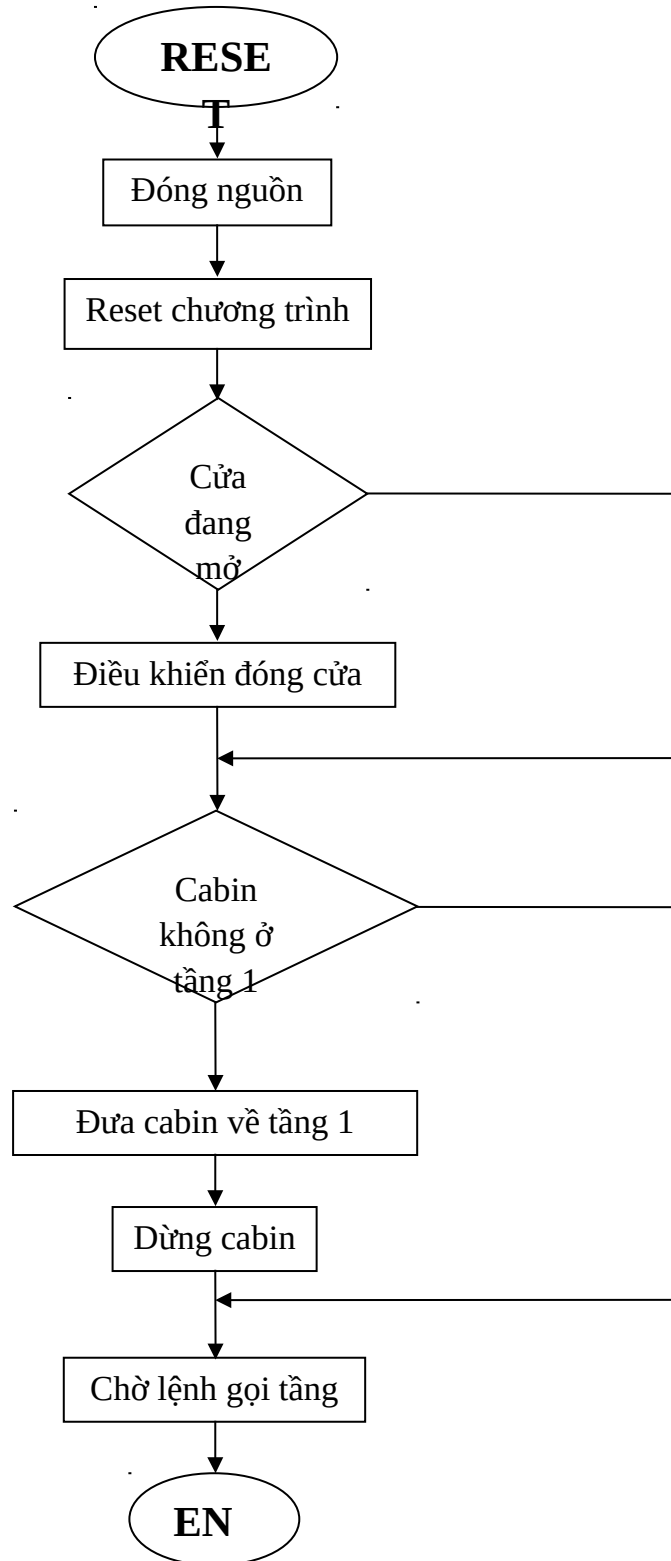
### 4.1 Lưu đồ giải thuật.

#### 4.1.1 Lưu đồ giải thuật điều khiển thang máy.



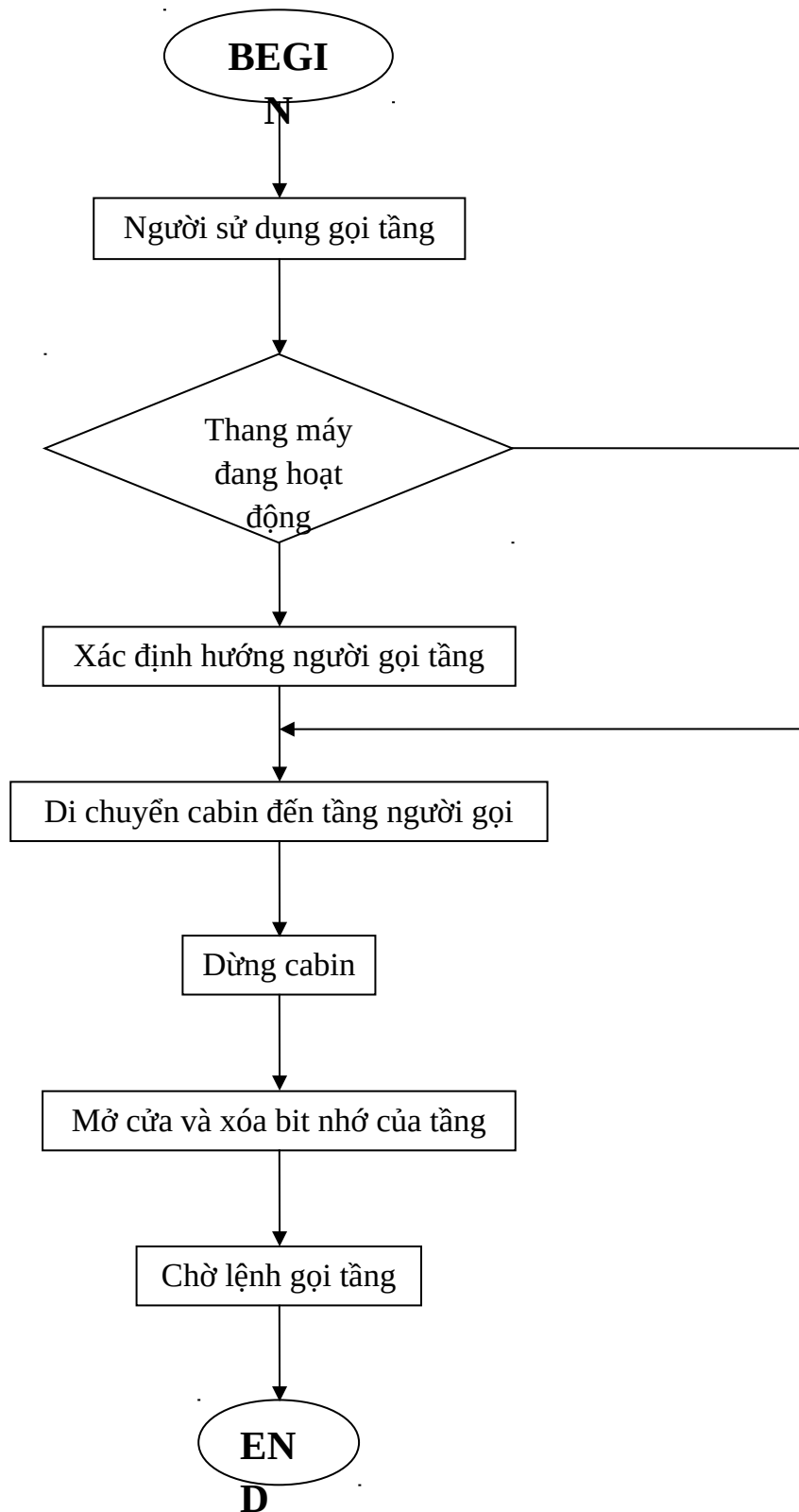
Hình 4.1. Lưu đồ giải thuật điều khiển tổng quát.



**4.1.2 Lưu đồ giải thuật phần RESET.**

Hình 4.2. Lưu đồ giải thuật phần RESET.

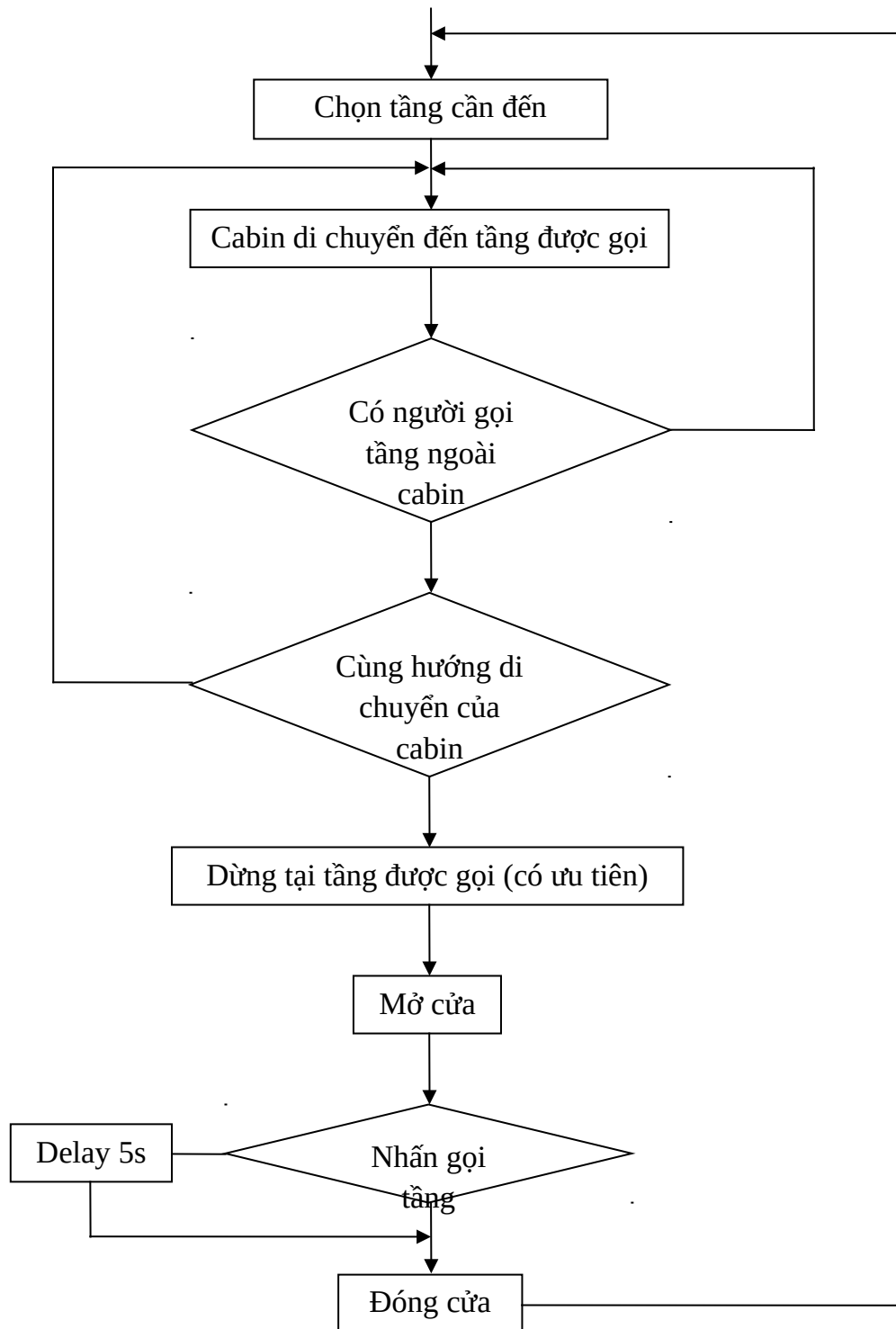
**4.1.3 Lưu đồ giải thuật gọi tầng bên ngoài cabin.**



Hình 4.3. Lưu đồ giải thuật gọi tầng bên ngoài cabin.

#### 4.1.4 Lưu đồ giải thuật gọi tầng bên trong cabin.





Hình 4.4. Lưu đồ giải thuật gọi tầng bên trong cabin.

## 4.2 Kết quả thực hiện.

Dù thời gian có hạn và gặp một số trục trặc nhỏ trong suốt thời gian thi công mô hình, nhưng với sự nỗ lực của bản thân và sự giúp đỡ tận tình của giáo viên hướng dẫn nên mô hình đã được hoàn tất và vận hành thử trước thời gian bảo vệ.

Hệ thống đã đáp ứng mọi yêu cầu của đề tài đặt ra Điều khiển thang máy dùng PLC S7-200 với những

- Thang máy hoạt động tốt.
- Hệ thống Cabin dừng tương đối chính xác nhờ công tắc hành trình.
- Hệ thống mạch điện điều khiển gọn nhẹ do dùng thiết bị PLC.
- Kết cấu phần cứng gọn nhẹ, chắc chắn.

### **4.3 Đánh giá và hướng phát triển đề tài.**

Mặc dù hệ thống thi công đã hoạt động tốt, tuy nhiên do kinh phí có hạn nên mô hình thực hiện chưa được tốt, để hoàn chỉnh hơn cần phát triển thêm một số chức năng như sau:

- Thiết kế nguồn phụ khi có sự cố mất điện.
- Để thêm phần tiện nghi trang bị thêm quạt thông gió và đèn chiếu sáng
- Thiết kế phần báo sự cố khi thang máy lỗi (kẹt thang máy, cửa cabin không đóng mở được..)
- Thêm vào bộ phận giảm chấn khi cabin di chuyển lên xuống.
- Thiết kế bộ phanh bảo hiểm phòng trường hợp cáp bị đứt.
- Lắp đặt thêm phần cảm biến ngay cửa ra vào cabin để cảm biến được vật cản để hoạt động chính xác hơn.

1. Qua thời gian thực hiện đề tài, chúng em có những nhận xét như sau:

- Về phần cứng: theo em thì đây là phần khó khăn nhất khi thiết kế và thi công thang máy bởi vì mô hình thang máy có rất nhiều chi tiết cho nên việc tính toán để đặt các chi tiết này lên mô hình cụ thể một cách hợp lý là không dễ dàng. Vì vậy đây cũng là phần tốn nhiều thời gian nhất trong quá trình làm đề tài này.

- Về phần mềm: với những kiến thức đã được học trong suốt những năm học vừa qua cộng với đây là một đề tài không mới nên việc tìm tài liệu tham khảo để hỗ trợ cho việc viết phần mềm để điều khiển thang máy cũng tương đối thuận lợi.

2. Đánh giá kết quả: Tuy gặp những khó khăn trên nhưng với sự cố gắng của nhóm cộng với sự giúp đỡ và hướng dẫn của quý thầy cô và các bạn, đặc biệt là giáo viên hướng dẫn thì nhóm đã hoàn thành mô hình đáp ứng được mục tiêu đã đề ra, cụ thể: Thang máy được điều khiển tương đối ổn định, các thiết bị hiển thị hoạt động tốt, cabin dừng đúng tầng mặc dù độ chính xác chưa cao.

**Quy ước các địa chỉ ngõ vào và ngõ ra cho chương trình điều khiển.****Bảng 10.** Quy ước các địa chỉ ngõ vào và ngõ ra cho chương trình điều khiển.

STT	Tên ngõ ra out put	Địa chỉ	Giải thích
1	start	I0.0	bat dau: reset va dua cabin ve tang 1
2	stop_	I0.1	dung toan bo he thong
3	goi_l1	I0.2	goi len khi dang o ngoai tang 1
4	goi_l2	I0.3	goi len khi dang o ngoai tang 2
5	goi_l3	I0.4	goi len khi dang o ngoai tang 3
6	goi_x2	I0.5	goi xuong khi dang o ngoai tang 2
7	goi_x3	I0.6	goi xuong khi dang o ngoai tang 3
8	goi_x4	I0.7	goi xuong khi dang o ngoai tang 4
9	t_1	I1.0	=toi tang 1:goi tang bang nut dieu khien trong cabin
10	t_2	I1.1	=toi tang 2:goi tang bang nut dieu khien trong cabin
11	t_3	I1.2	=toi tang 3:goi tang bang nut dieu khien trong cabin
12	t_4	I1.3	=toi tang 4:goi tang bang nut dieu khien trong cabin
13	open	I1.4	lenh mo cua
14	close	I1.5	lenh dong cua
15	opened	I2.0	cong tac hanh trinh xac dinh cua mo
16	closed	I2.1	cong tac hanh trinh xac dinh cua dong
17	ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
18	ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
19	ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
20	ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
21		M0.0	bit nho dieu khien cho toan he thong hoat dong
22	nho_x2	M0.1	bit nho goi xuong khi dang o tang 2
23	nho_x3	M0.2	bit nho goi xuong khi dang o tang 3
24	nho_x4	M0.3	bit nho goi xuong khi dang o tang 4

25	nho_l1	M0.4	bit nho gọi lên khi đang ở tầng 1
26	nho_l2	M0.5	bit nho gọi lên khi đang ở tầng 2
27	nho_l3	M0.6	bit nho gọi lên khi đang ở tầng 3
28	nho_lên	M0.7	bit nho đi lên
29	nho_xuong	M1.0	bit nho đi xuống
30	nho1	M1.1	bit nho chọn tầng 1
31	nho2	M1.2	bit nho chọn tầng 2
32	nho3	M1.3	bit nho chọn tầng 3
33	nho4	M1.4	bit nho chọn tầng 4
34	nho_open	M1.5	bit nho mở cửa
35	xoaled_l	M1.6	xóa bit nho led lên
36	xoaled_x	M1.7	xóa bit nho led xuống
37	xoa1	M3.0	xóa bit nho tầng 1
38	xoa2	M3.1	xóa bit nho tầng 2
39	xoa3	M3.2	xóa bit nho tầng 3
40	xoa4	M3.3	xóa bit nho tầng 4
41	nho_xuong_free	M3.4	bit nho đưa cabin xuống khi thang máy rảnh
42	xoa_nho_xuong_free	M3.5	xóa bit nho cabin xuống khi thang máy đang rảnh
43	reset_T5	M4.0	
44	nho1_	M4.1	bit nho chọn tầng 1
45	nho2_	M4.2	bit nho chọn tầng 2
46	nho3_	M4.3	bit nho chọn tầng 3
47	nho4_	M4.4	bit nho chọn tầng 4
48	nho_x2_	M4.5	bit nho gọi xuống khi đang ở tầng 2
49	nho_x3_	M4.6	bit nho gọi xuống khi đang ở tầng 3
50	nho_x4_	M4.7	bit nho gọi xuống khi đang ở tầng 4
51	nho_reset_mocua	M5.0	
52	nho_l1_	M5.1	bit nho gọi lên khi đang ở tầng 1
53	nho_l2_	M5.2	bit nho gọi lên khi đang ở tầng 2
54	nho_l3_	M5.4	bit nho gọi lên khi đang ở tầng 3
55	nho_close	M5.5	bit nho đóng cửa
56	dung_tang	M5.6	bit nho dừng tầng
57	reset	M5.7	bit nho dừng cho phần reset
58	nho_close_reset	M6.1	bit nho đóng cửa (phần reset)
59	nho_xuong_reset	M6.2	bit nho kéo xuống (phần reset)
60	keo_lên	Q0.0	điều khiển động cơ kéo lên
61	keo_xuong	Q0.1	điều khiển động cơ kéo xuống
62	mo_cua	Q0.2	điều khiển động cơ mở cửa
63	dong_cua	Q0.3	điều khiển động cơ đóng cửa

64	led1	Q0.4	hien thi led bao dang o tang 1
65	led2	Q0.5	hien thi led bao dang o tang 2
66	led3	Q0.6	hien thi led bao dang o tang 3
67	led4	Q0.7	hien thi led bao dang o tang 4
68	dang_len	Q1.0	hien thi led bao dang di len
69	dang_xuong	Q1.1	hien thi led bao dang di xuong

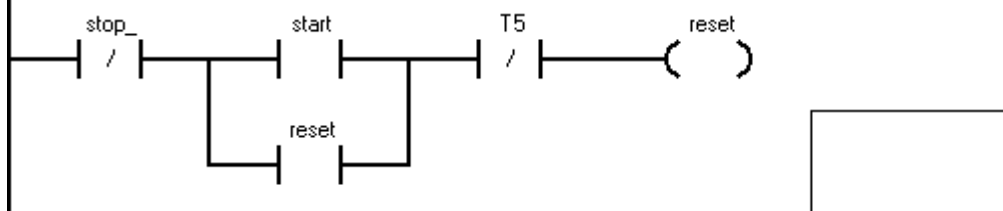
**Chương trình điều khiển thang máy 4 tầng:**



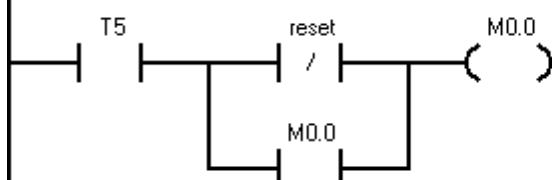
## CHƯƠNG TRÌNH ĐIỀU KHIỂN THANG MÁY 4 TẦNG

**Network 1**      phản điều khiển start (tin hiệu reset) và stop

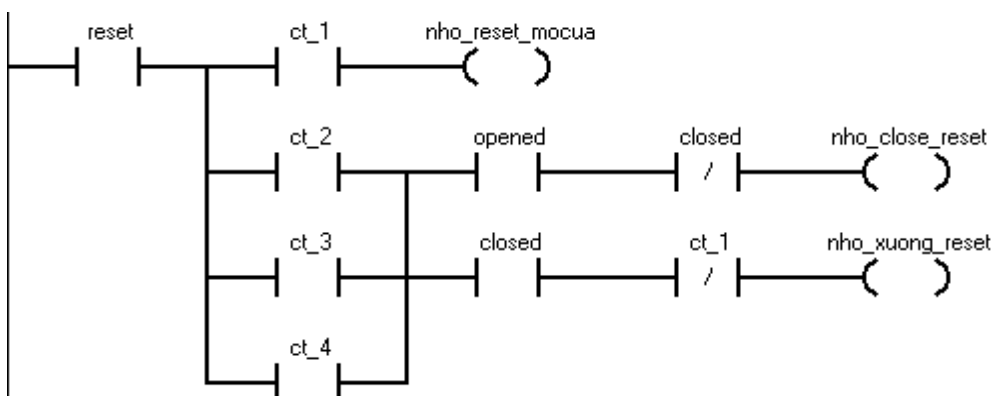
phản điều khiển start (tin hiệu reset) và stop



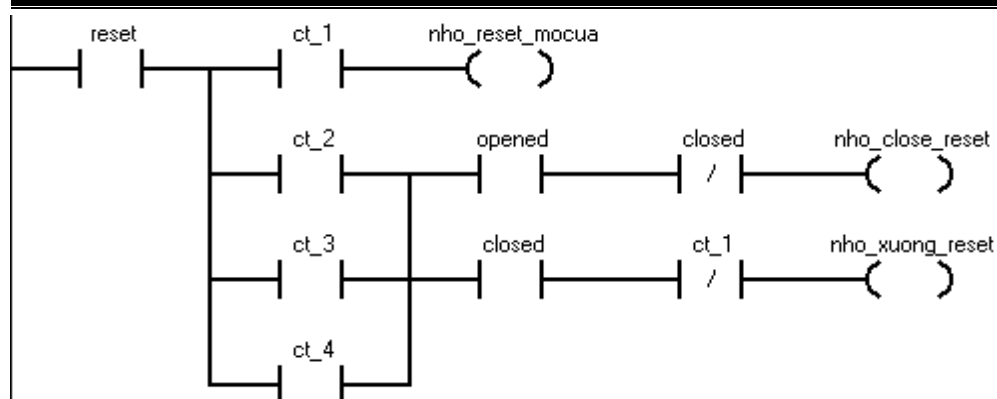
Symbol	Address	Comment
reset	M5.7	bit nhớ dùng cho phản reset
start	I0.0	bắt đầu: reset và đưa cabin về tầng 1
stop_	I0.1	dừng toàn bộ hệ thống

**Network 2**      khi chương trình reset kết thúc thì kích hoạt bit nhớ điều khiển cho toàn hệ thống sẵn sàng hoạt động

Symbol	Address	Comment
reset	M5.7	bit nhớ dùng cho phản reset

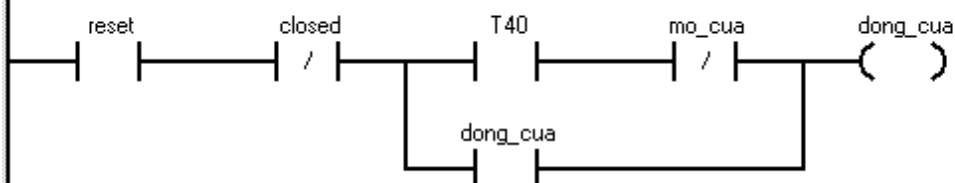
**Network 3**      chương trình con đưa cabin về tầng 1 khi có lệnh reset

Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định cửa đóng
ct_1	I2.2	công tắc hành trình xác định cabin ở tầng 1
ct_2	I2.3	công tắc hành trình xác định cabin ở tầng 2
ct_3	I2.4	công tắc hành trình xác định cabin ở tầng 3
ct_4	I2.5	công tắc hành trình xác định cabin ở tầng 4



Symbol	Address	Comment
closed	I2.1	cong tac hanh trinh xac dinh cua dong
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
nho_close_reset	M6.1	bit nho dong cua (phan reset)
nho_reset_mocua	M5.0	
nho_xuong_reset	M6.2	bit nho keo xuong (phan reset)
opened	I2.0	cong tac hanh trinh xac dinh cua mo
reset	M5.7	bit nho dung cho phan reset

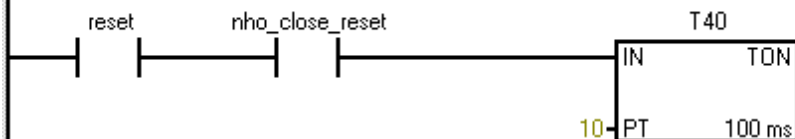
Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định cửa đóng
ct_1	I2.2	công tắc hành trình xác định cabin ở tầng 1
ct_2	I2.3	công tắc hành trình xác định cabin ở tầng 2
ct_3	I2.4	công tắc hành trình xác định cabin ở tầng 3
ct_4	I2.5	công tắc hành trình xác định cabin ở tầng 4
nho_close_reset	M6.1	bit nhớ đóng cửa (phan reset)
nho_reset_mocua	M5.0	
nho_xuong_reset	M6.2	bit nhớ kéo xuống (phan reset)
opened	I2.0	công tắc hành trình xác định cửa mở
reset	M5.7	bit nhớ dùng cho phan reset

**Network 4**    lệnh đóng cửa

Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định cửa đóng
dong_cua	Q0.3	điều khiển động cơ đóng cửa
mo_cua	Q0.2	điều khiển động cơ mở cửa
reset	M5.7	bit nhớ dùng cho phan reset

**Network 5**

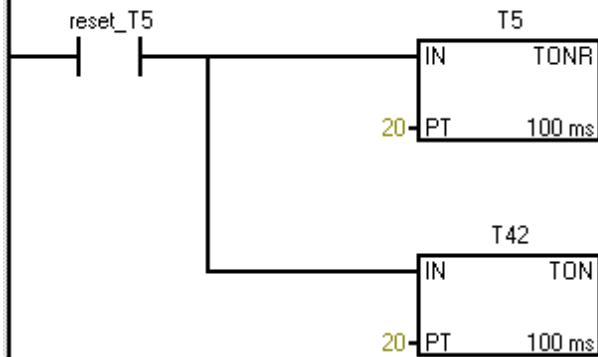
sau 1s thì cửa đóng lại



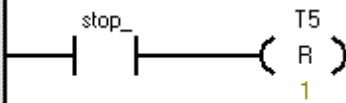
Symbol	Address	Comment
nho_close_reset	M6.1	bit nhớ đóng cửa (phan reset)
reset	M5.7	bit nhớ dùng cho phan reset

**Network 6**      lệnh mở cửa và lệnh chuyển sang trạng chờ đo phục vụ sau khi reset

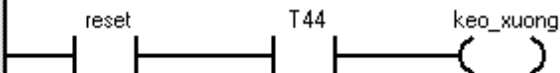
cabin tới tầng 1 thì 2s sau cửa tự động mở  
sau 5s chương trình reset được ngắt, thang máy sẵn sàng phục vụ



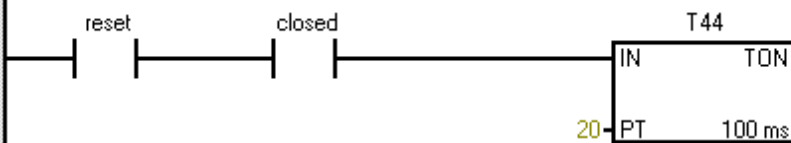
Symbol	Address	Comment
reset_T5	M4.0	

**Network 7** reset lại T5 khi nhan stop

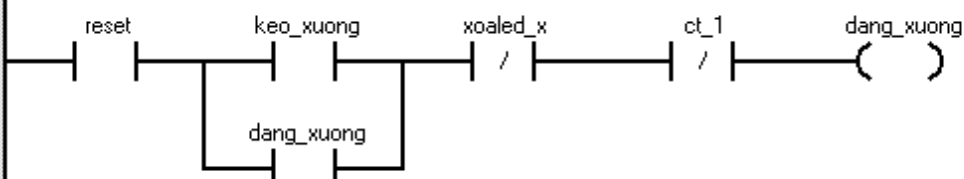
Symbol	Address	Comment
stop_	I0.1	dung toan bo he thong

**Network 8** lenh keo cabin xuong

Symbol	Address	Comment
keo_xuong	Q0.1	dieu khien dong co keo xuong
reset	M5.7	bit nho dung cho phan reset

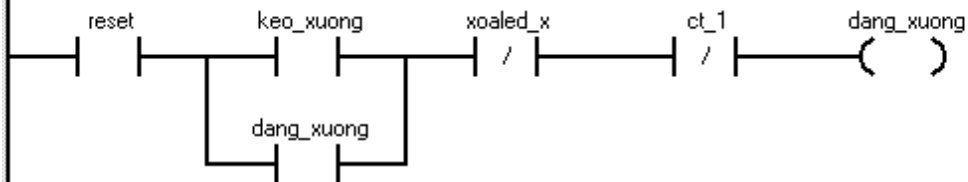
**Network 9**

Symbol	Address	Comment
closed	I2.1	cong tac hanh trinh xac dinh cua dong
reset	M5.7	bit nho dung cho phan reset

**Network 10** led bao dang xuong

Symbol	Address	Comment
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
dang_xuong	Q1.1	hien thi led bao dang di xuong
keo_xuong	Q0.1	dieu khien dong co keo xuong
reset	M5.7	bit nho dung cho phan reset

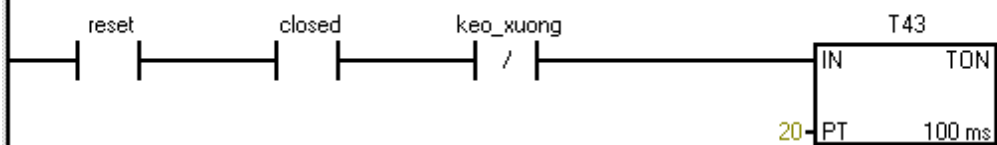
Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định của dòng
reset	M5.7	bit nhớ dùng cho phản reset

**Network 10** led báo đang xuống

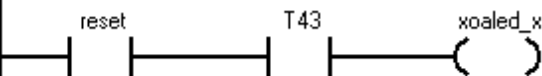
Symbol	Address	Comment
ct_1	I2.2	công tắc hành trình xác định cabin ở tầng 1
dang_xuong	Q1.1	hiện thị led báo đang đi xuống
keo_xuong	Q0.1	điều khiển động cơ kéo xuống
reset	M5.7	bit nhớ dùng cho phản reset
xoaled_x	M1.7	xóa bit nhớ led xuống

**Network 11**

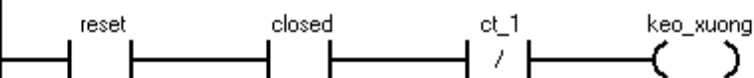
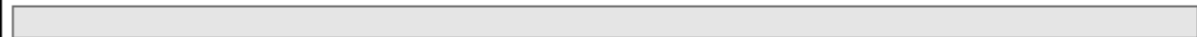
sau khi động cơ kéo dừng ngang tầng thì 2s sau xóa led báo lên



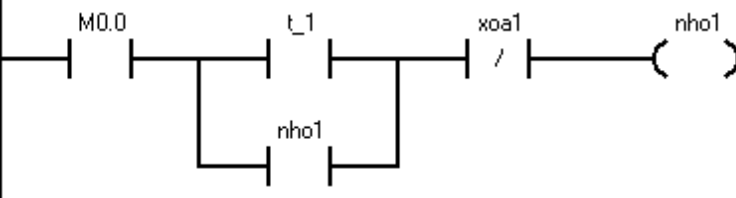
closed	I2.1	cong tac hanh trinh xac dinh cua dong
keo_xuong	Q0.1	dieu khien dong co keo xuong
reset	M5.7	bit nho dung cho phan reset

**Network 12**

Symbol	Address	Comment
reset	M5.7	bit nho dung cho phan reset
xoaled_x	M1.7	xoa bit nho led xuong

**Network 13** khi den tang 1 cabin dung lai

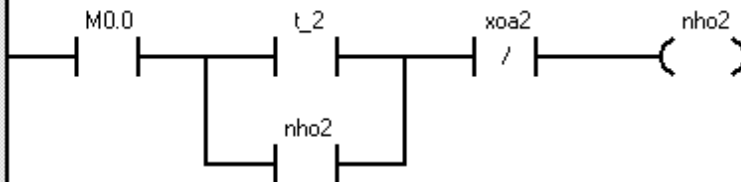
Symbol	Address	Comment
closed	I2.1	cong tac hanh trinh xac dinh cua dong
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
keo_xuong	Q0.1	dieu khien dong co keo xuong
reset	M5.7	bit nho dung cho phan reset

**Network 14** dat bit nho cho cac nut gọi

Symbol	Address	Comment
nho1	M1.1	bit nhớ chọn tầng 1
t_1	I1.0	=tôi tầng 1: gọi tầng bằng nút điều khiển trong cabin
xoa1	M3.0	xóa bit nhớ tầng 1

**Network 15**

bit gọi tầng 2

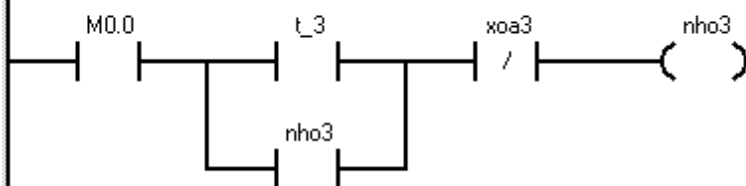


Symbol	Address	Comment
nho2	M1.2	bit nhớ chọn tầng 2
t_2	I1.1	=tôi tầng 2: gọi tầng bằng nút điều khiển trong cabin
xoa2	M3.1	xóa bit nhớ tầng 2



**Network 16**

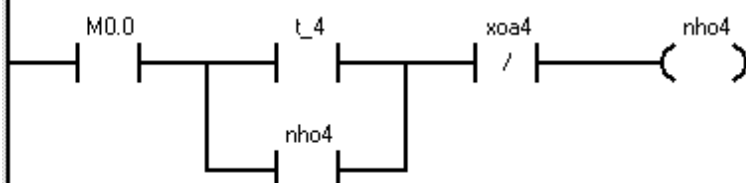
bit gọi tầng 3



Symbol	Address	Comment
nho3	M1.3	bit nhớ chọn tầng 3
t_3	I1.2	=tôi tầng 3: gọi tầng bằng nút điều khiển trong cabin
xoa3	M3.2	xóa bit nhớ tầng 3

**Network 17**

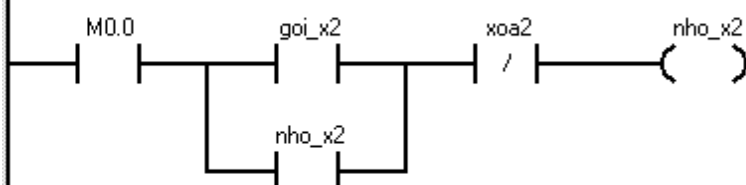
bit gọi tầng 4



Symbol	Address	Comment
nho4	M1.4	bit nhớ chọn tầng 4
t_4	I1.3	=tôi tầng 4: gọi tầng bằng nút điều khiển trong cabin
xoa4	M3.3	xóa bit nhớ tầng 4

**Network 18**

bit gọi xuống khi đang ở tầng 2

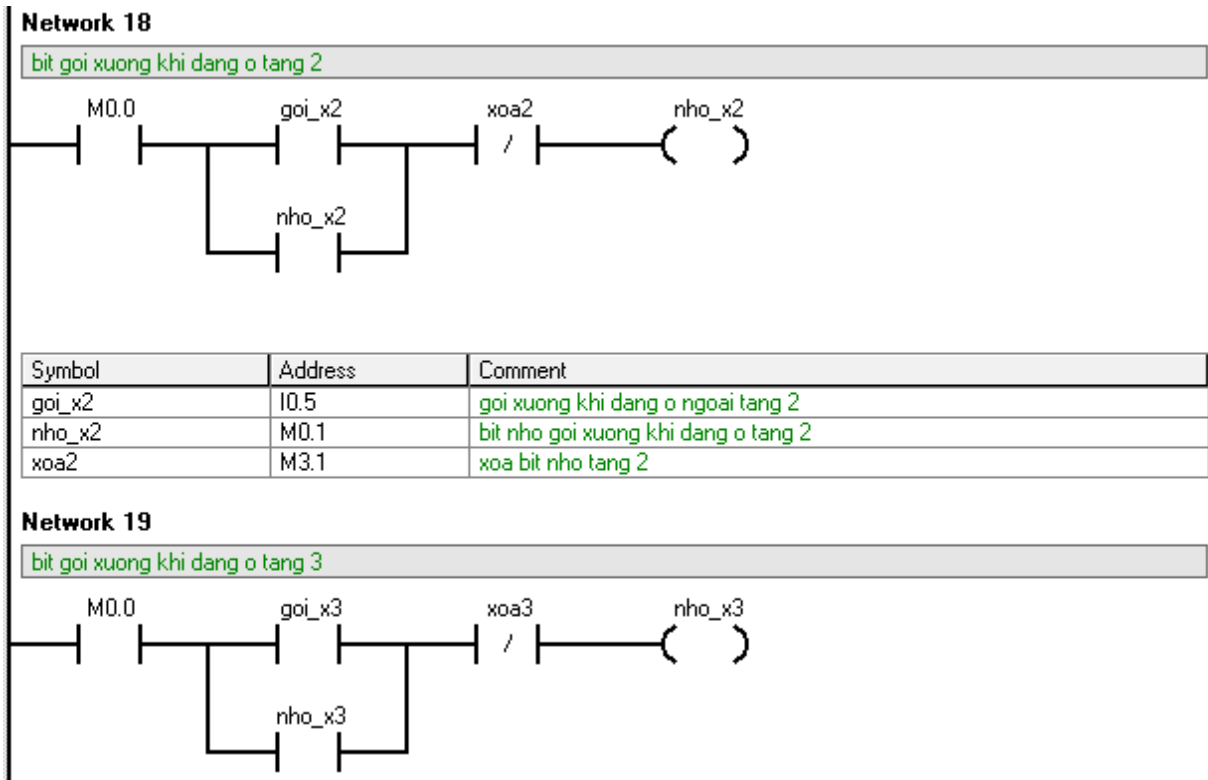


Symbol	Address	Comment
gọi_x2	I0.5	gọi xuống khi đang ở ngoài tầng 2
nho_x2	M0.1	bit nhớ gọi xuống khi đang ở tầng 2
xoa2	M3.1	xóa bit nhớ tầng 2

**Network 19**

bit gọi xuống khi đang ở tầng 3

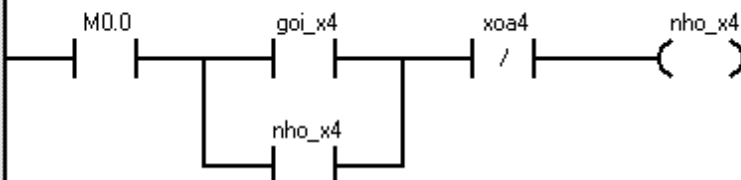




Symbol	Address	Comment
goi_x3	I0.6	goi xuống khi đang ở ngoài tầng 3
nho_x3	M0.2	bit nhớ gọi xuống khi đang ở tầng 3
xoa3	M3.2	xóa bit nhớ tầng 3

**Network 20**

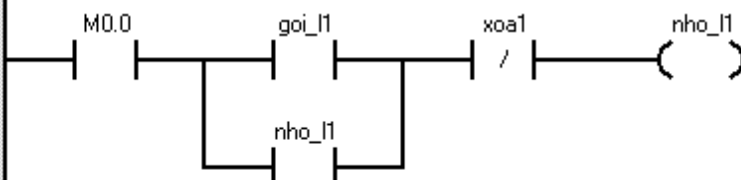
bit gọi xuống khi đang ở tầng 4



Symbol	Address	Comment
goi_x4	I0.7	goi xuống khi đang ở ngoài tầng 4
nho_x4	M0.3	bit nhớ gọi xuống khi đang ở tầng 4
xoa4	M3.3	xóa bit nhớ tầng 4

**Network 21**

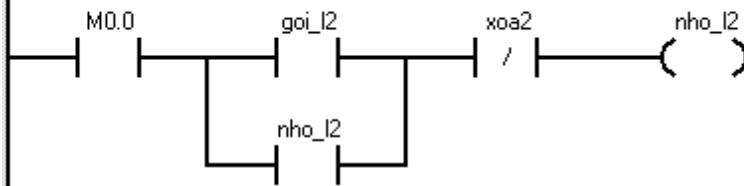
bit gọi lên khi đang ở tầng 1



Symbol	Address	Comment
goi_l1	I0.2	goi lên khi đang ở ngoài tầng 1
nho_l1	M0.4	bit nhớ gọi lên khi đang ở tầng 1
xoa1	M3.0	xóa bit nhớ tầng 1

**Network 22**

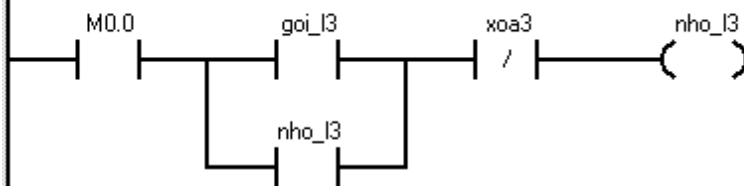
bit gọi lên khi đang ở tầng 2



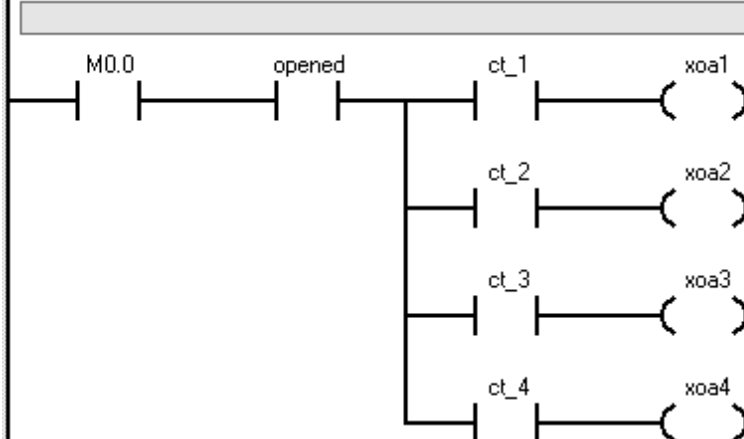
Symbol	Address	Comment
goi_l2	I0.3	gọi lên khi đang ở ngoài tầng 2
nho_l2	M0.5	bit nhớ gọi lên khi đang ở tầng 2
xoa2	M3.1	xóa bit nhớ tầng 2

**Network 23**

bit gọi lên khi đang ở tầng 3

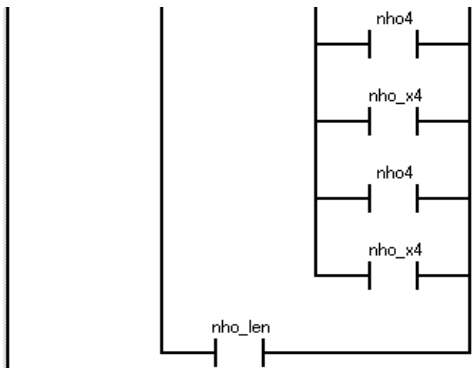
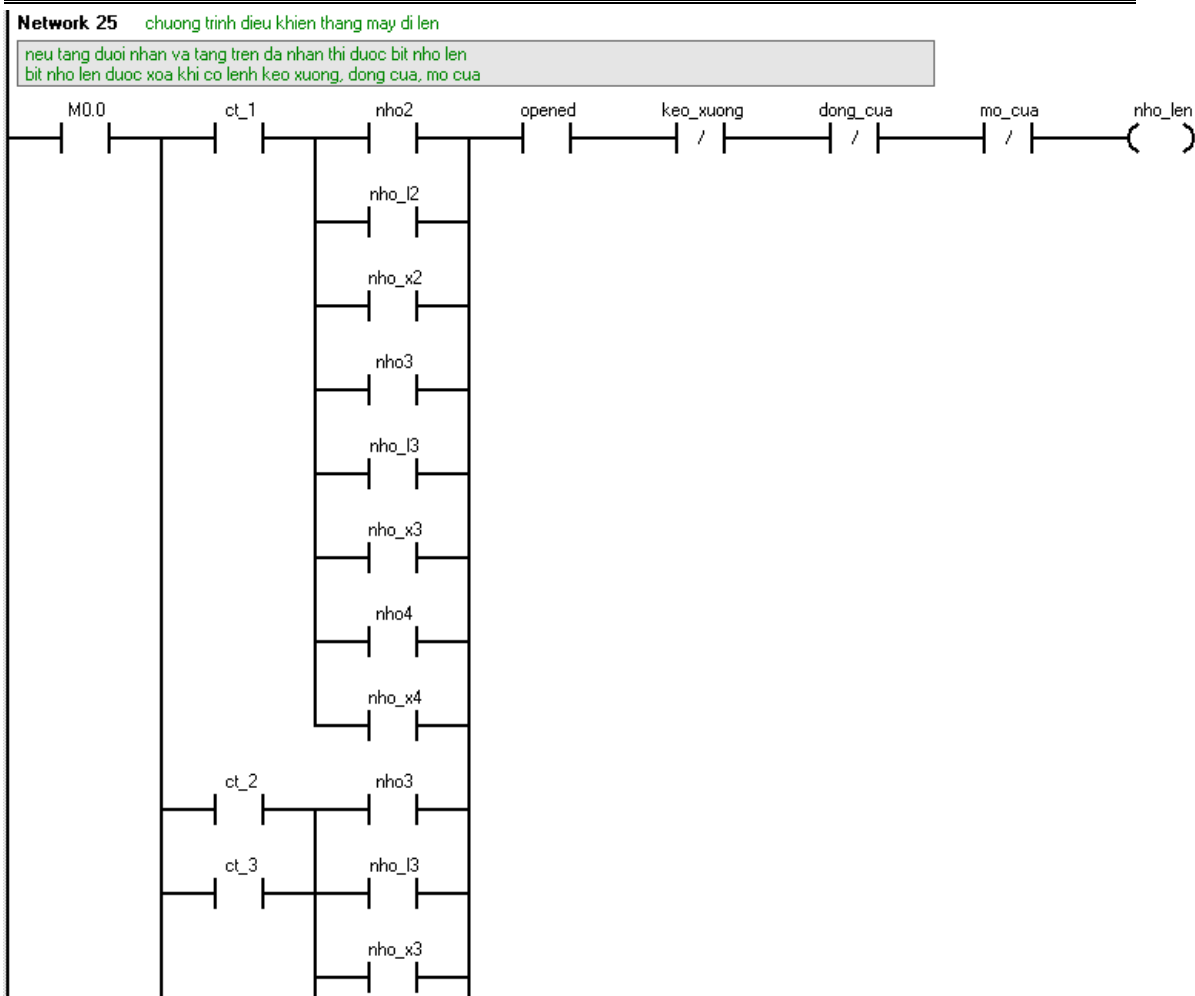


Symbol	Address	Comment
goi_l3	I0.4	gọi lên khi đang ở ngoài tầng 3
nho_l3	M0.6	bit nhớ gọi lên khi đang ở tầng 3
xoa3	M3.2	xóa bit nhớ tầng 3

**Network 24** Xóa bit nhớ

Symbol	Address	Comment
ct_1	I2.2	công tắc hành trình xác định cabin ở tầng 1
ct_2	I2.3	công tắc hành trình xác định cabin ở tầng 2
ct_3	I2.4	công tắc hành trình xác định cabin ở tầng 3
ct_4	I2.5	công tắc hành trình xác định cabin ở tầng 4
opened	I2.0	công tắc hành trình xác định cửa mở
xoa1	M3.0	xóa bit nhớ tầng 1
xoa2	M3.1	xóa bit nhớ tầng 2
xoa3	M3.2	xóa bit nhớ tầng 3
xoa4	M3.3	xóa bit nhớ tầng 4

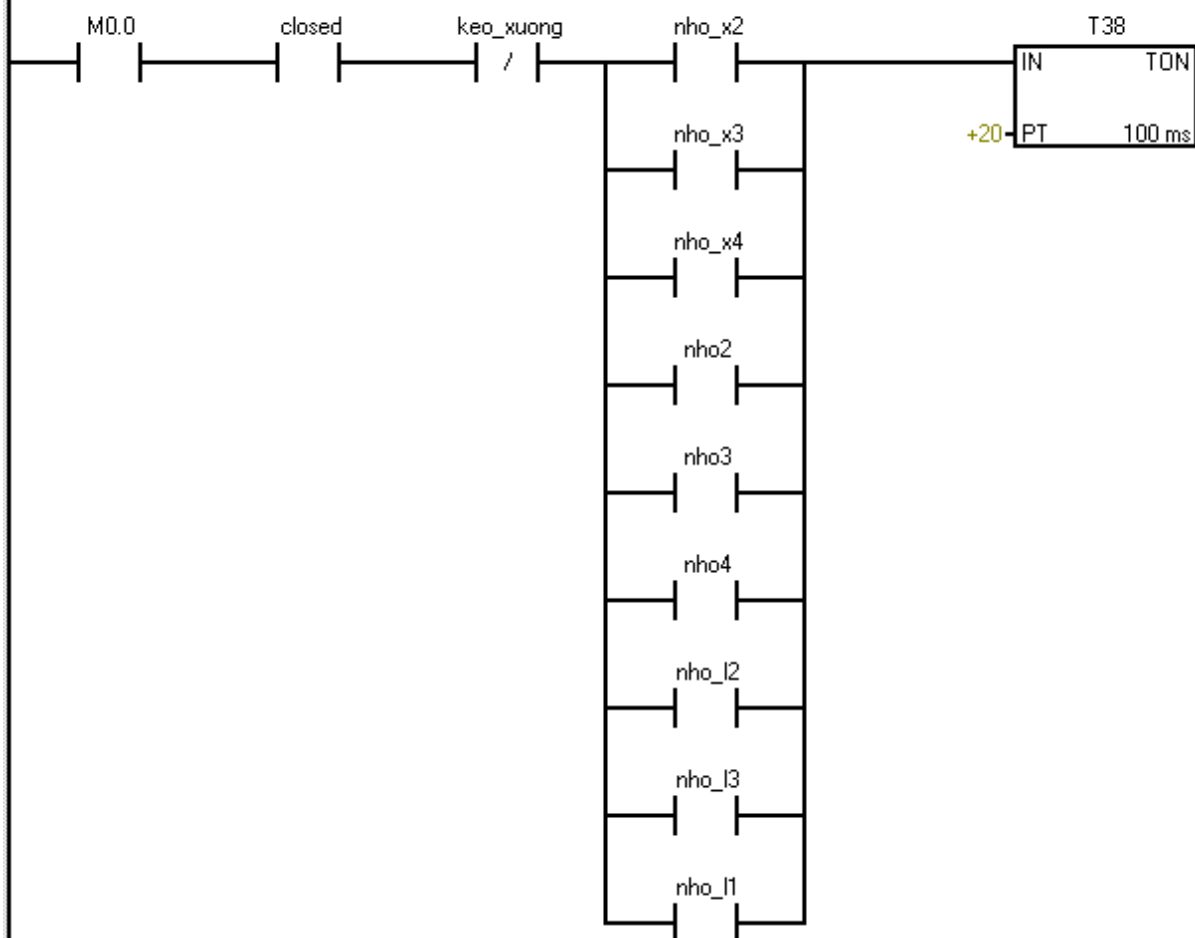




Symbol	Address	Comment
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
dong_cua	Q0.3	dieu khien dong co dong cua
keo_xuong	Q0.1	dieu khien dong co keo xuong
mo_cua	Q0.2	dieu khien dong co mo cua
nho2	M1.2	bit nho chon tang 2
nho3	M1.3	bit nho chon tang 3
nho4	M1.4	bit nho chon tang 4
nho_l2	M0.5	bit nho goi len khi dang o tang 2
nho_l3	M0.6	bit nho goi len khi dang o tang 3
nho_len	M0.7	bit nho di len
nho_x2	M0.1	bit nho goi xuong khi dang o tang 2



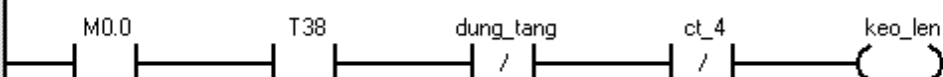
**Network 26** sau khi nạp bit nhớ thì khoảng 2s động cơ bắt đầu kéo cabin lên;  
có bit nhớ gọi tầng thì mới được kéo cabin



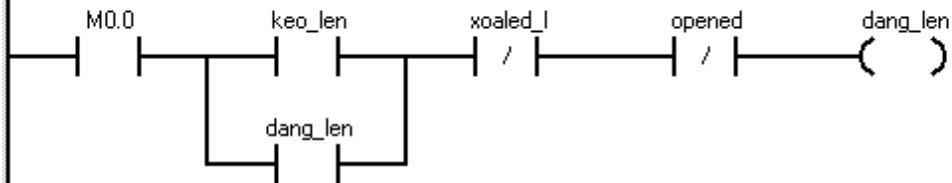


Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định của dòng
keo_xuong	Q0.1	điều khiển động cơ kéo xuống
nho2	M1.2	bit nhớ chọn tầng 2
nho3	M1.3	bit nhớ chọn tầng 3
nho4	M1.4	bit nhớ chọn tầng 4
nho_l1	M0.4	bit nhớ gọi lên khi đang ở tầng 1
nho_l2	M0.5	bit nhớ gọi lên khi đang ở tầng 2
nho_l3	M0.6	bit nhớ gọi lên khi đang ở tầng 3
nho_x2	M0.1	bit nhớ gọi xuống khi đang ở tầng 2
nho_x3	M0.2	bit nhớ gọi xuống khi đang ở tầng 3
nho_x4	M0.3	bit nhớ gọi xuống khi đang ở tầng 4

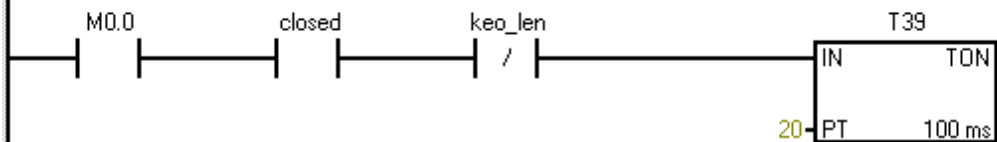
**Network 27** khi cabin di chuyển không qua giới hạn là tầng 4



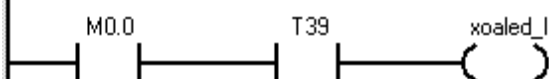
Symbol	Address	Comment
ct_4	I2.5	công tắc hành trình xác định cabin ở tầng 4
dung_tang	M5.6	bit nhớ dừng tầng
keo_len	Q0.0	điều khiển động cơ kéo lên

**Network 28**    *hien thi led di len**khi dong co keo len thi den bao led len**khi dong co dung ngang tang de don khach nhung van dang theo chieu di len thi hien thi led len*

Symbol	Address	Comment
dang_len	Q1.0	<i>hien thi led bao dang di len</i>
keo_len	Q0.0	<i>dieu khien dong co keo len</i>
opened	I2.0	<i>cong tac hanh trinh xac dinh cua mo</i>
xoaled_l	M1.6	<i>xoa bit nho led len</i>

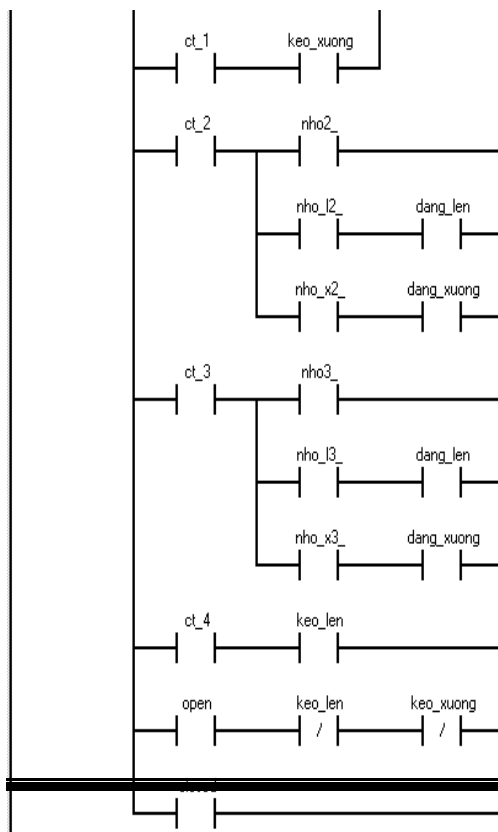
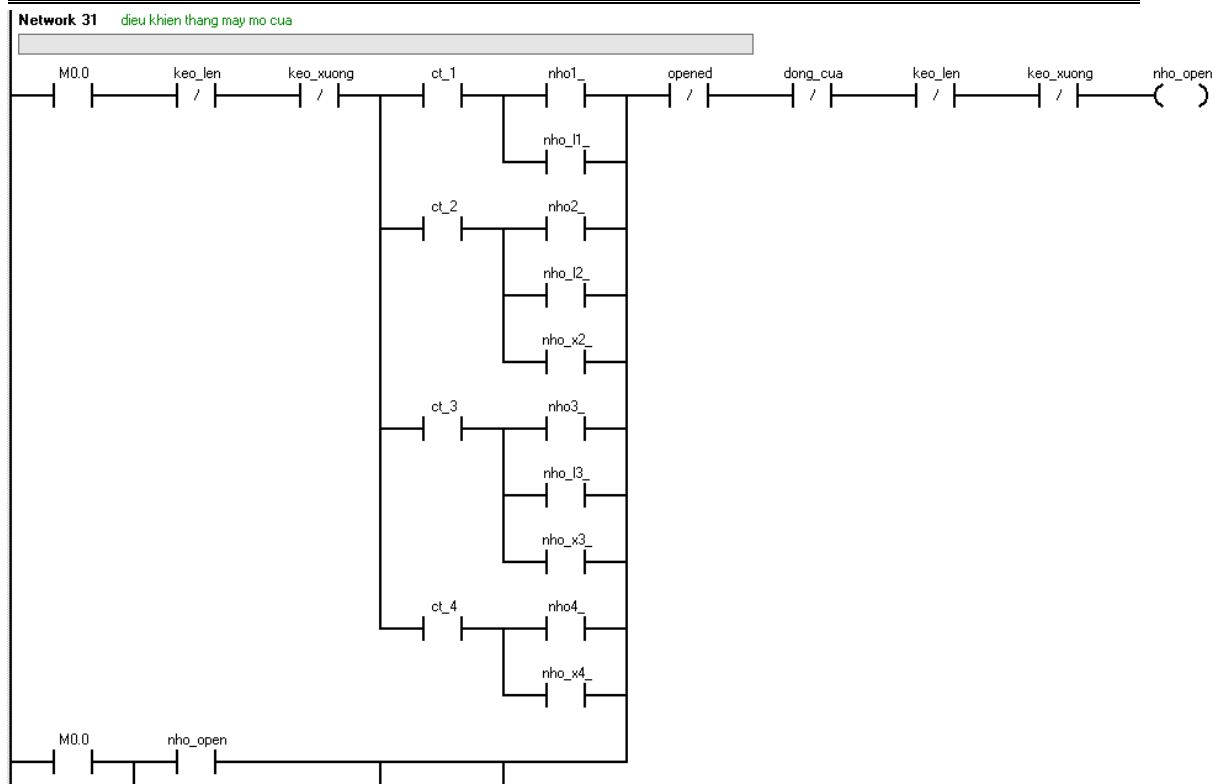
**Network 29**    *sau khi dong co keo dung ngang tang thi 2s sau xoa led bao len*

Symbol	Address	Comment
closed	I2.1	<i>cong tac hanh trinh xac dinh cua dong</i>
keo_len	Q0.0	<i>dieu khien dong co keo len</i>

**Network 30**

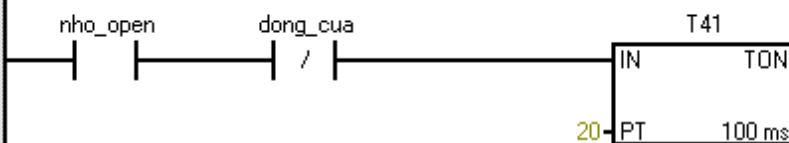
Symbol	Address	Comment
xoaled_l	M1.6	<i>xoa bit nho led len</i>



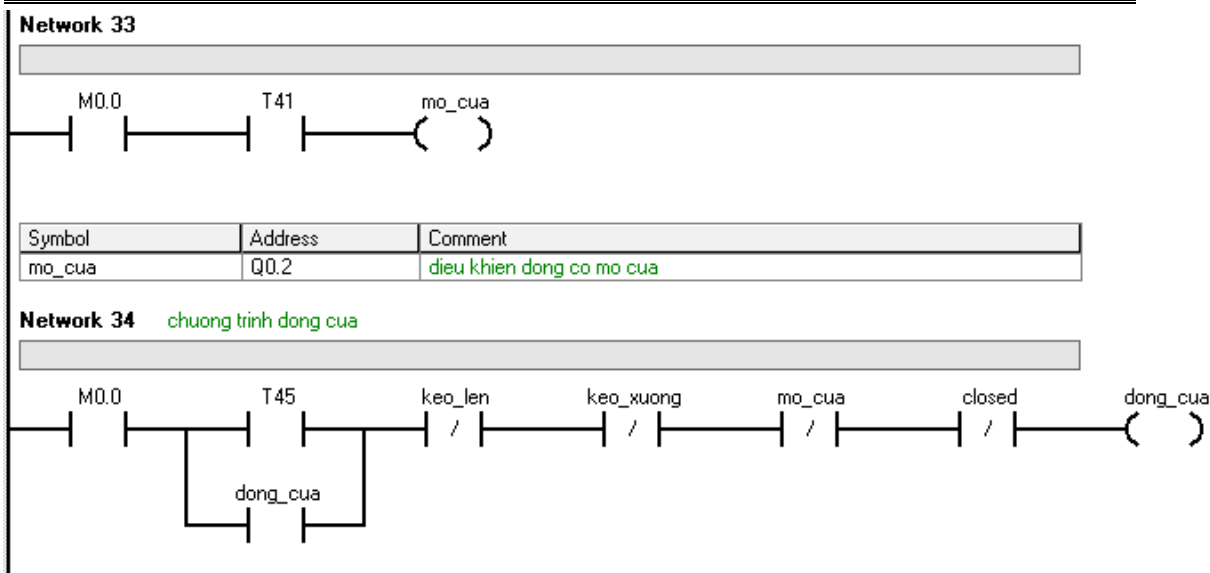




Symbol	Address	Comment
closed	I2.1	cong tac hanh trinh xac dinh cua dong
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
dang_len	Q1.0	hien thi led bao dang di len
dang_xuong	Q1.1	hien thi led bao dang di xuong
dong_cua	Q0.3	dieu khien dong co dong cua
keo_len	Q0.0	dieu khien dong co keo len
keo_xuong	Q0.1	dieu khien dong co keo xuong
nho1_	M4.1	bit nho chon tang 1
nho2_	M4.2	bit nho chon tang 2
nho3_	M4.3	bit nho chon tang 3
nho4_	M4.4	bit nho chon tang 4
nho_l1_	M5.1	bit nho goi le khi dang o tang 1
nho_l2_	M5.2	bit nho goi le khi dang o tang 2
nho_l3_	M5.4	bit nho goi le khi dang o tang 3
nho_open	M1.5	bit nho mo cua
nho_x2_	M4.5	bit nho goi xuong khi dang o tang 2
nho_x3_	M4.6	bit nho goi xuong khi dang o tang 3
nho_x4_	M4.7	bit nho goi xuong khi dang o tang 4
open	I1.4	lenh mo cua
opened	I2.0	cong tac hanh trinh xac dinh cua mo

**Network 32** delay thời gian mở cửakhi có bit `nho_open` thì sau 2s của mô

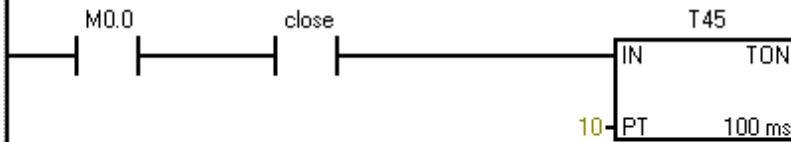
Symbol	Address	Comment
dong_cua	Q0.3	dieu khien dong co dong cua
nho_open	M1.5	bit nho mo cua



Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định của dồng
dong_cua	Q0.3	điều khiển dồng có dồng của
keo_len	Q0.0	điều khiển dồng có kéo lên
keo_xuong	Q0.1	điều khiển dồng có kéo xuống
mo_cua	Q0.2	điều khiển dồng có mở cửa

#### Network 35

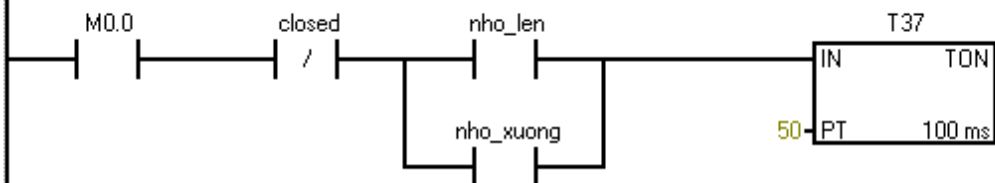
nhận close sau 1s thì của dồng lại



Symbol	Address	Comment
close	I1.5	lệnh dồng của

#### Network 36 lệnh dồng của

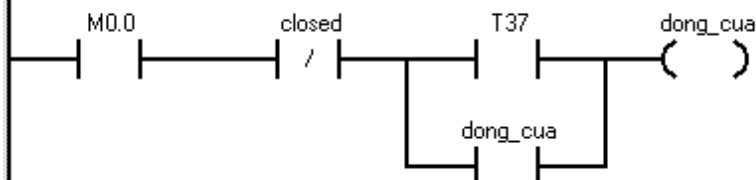
sau 2s khi có lệnh gọi thì của dồng lại



Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định của dồng
nho_len	M0.7	bit nhớ đi lên
nho_xuong	M1.0	bit nhớ đi xuống

#### Network 37

Q0.7: lệnh dồng của thay thế tạm thời

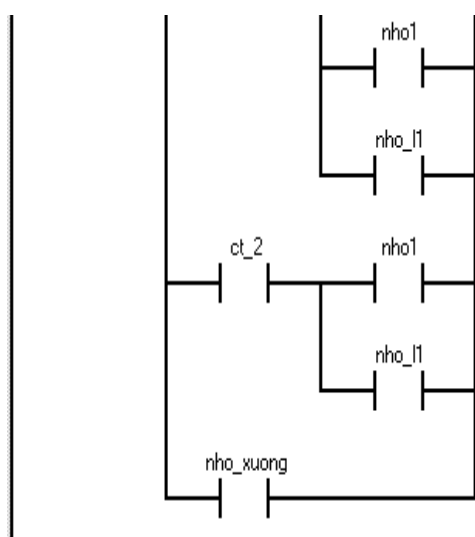
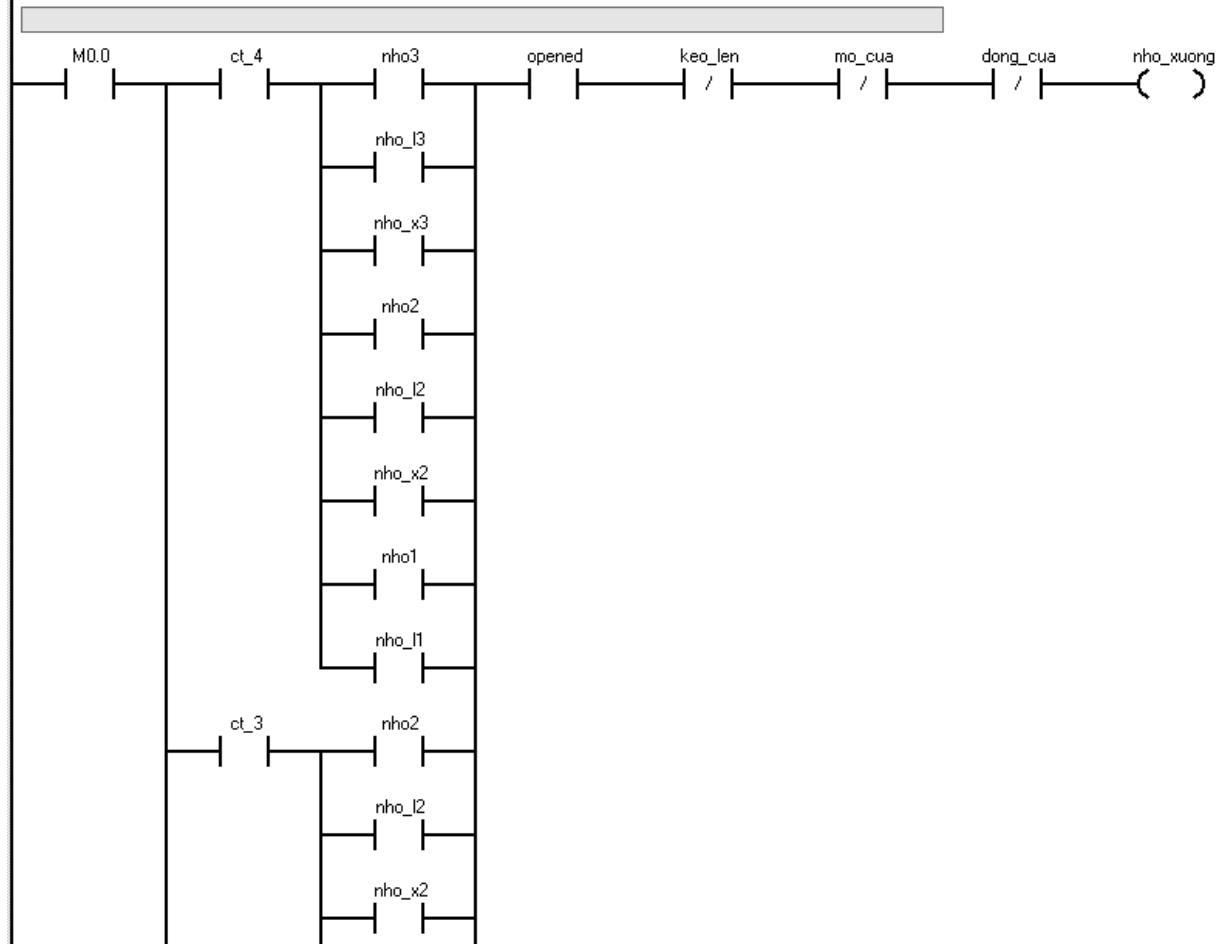


Symbol	Address	Comment
closed	I2.1	công tắc hành trình xác định của dồng
dong_cua	Q0.3	điều khiển dồng có dồng của





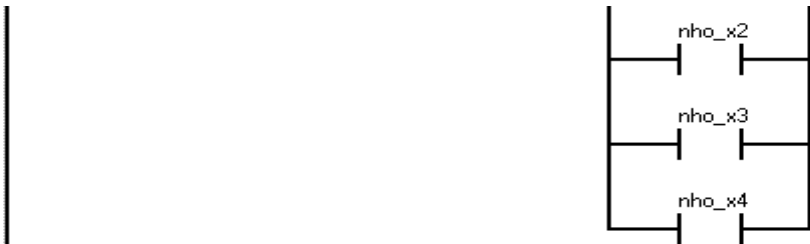
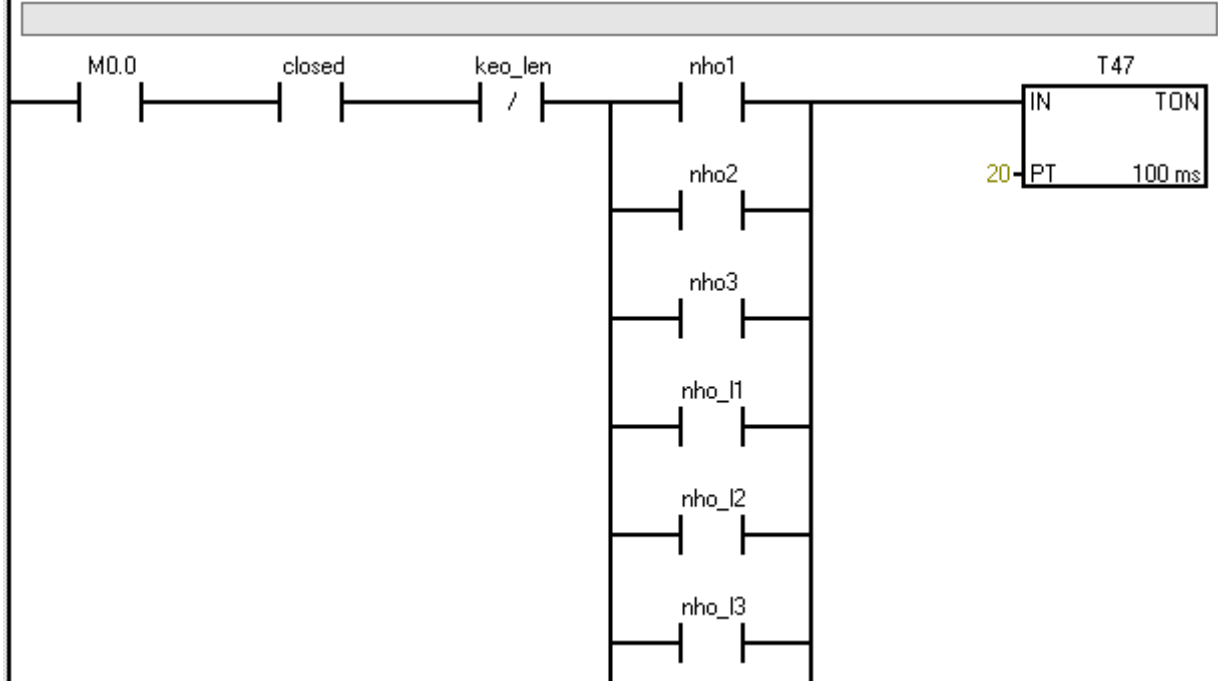
**Network 38** chương trình điều khiển thang máy đi xuống



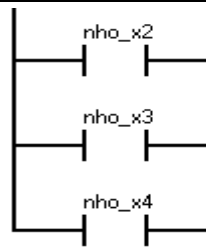


Symbol	Address	Comment
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
dong_cua	Q0.3	dieu khien dong co dong cua
keo_len	Q0.0	dieu khien dong co keo len
mo_cua	Q0.2	dieu khien dong co mo cua
nho1	M1.1	bit nho chon tang 1
nho2	M1.2	bit nho chon tang 2
nho3	M1.3	bit nho chon tang 3
nho_l1	M0.4	bit nho goi len khi dang o tang 1
nho_l2	M0.5	bit nho goi len khi dang o tang 2
nho_l3	M0.6	bit nho goi len khi dang o tang 3
nho_x2	M0.1	bit nho goi xuong khi dang o tang 2
nho_x3	M0.2	bit nho goi xuong khi dang o tang 3
nho_xuong	M1.0	bit nho di xuong
opened	I2.0	cong tac hanh trinh xac dinh cua mo

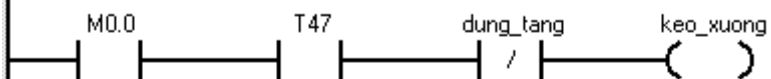
**Network 39** sau khi nap bit nho thi khoang 2s dong co bat dau keo cabin xuong;  
co bit nho goi tang thi moi duoc keo cabin



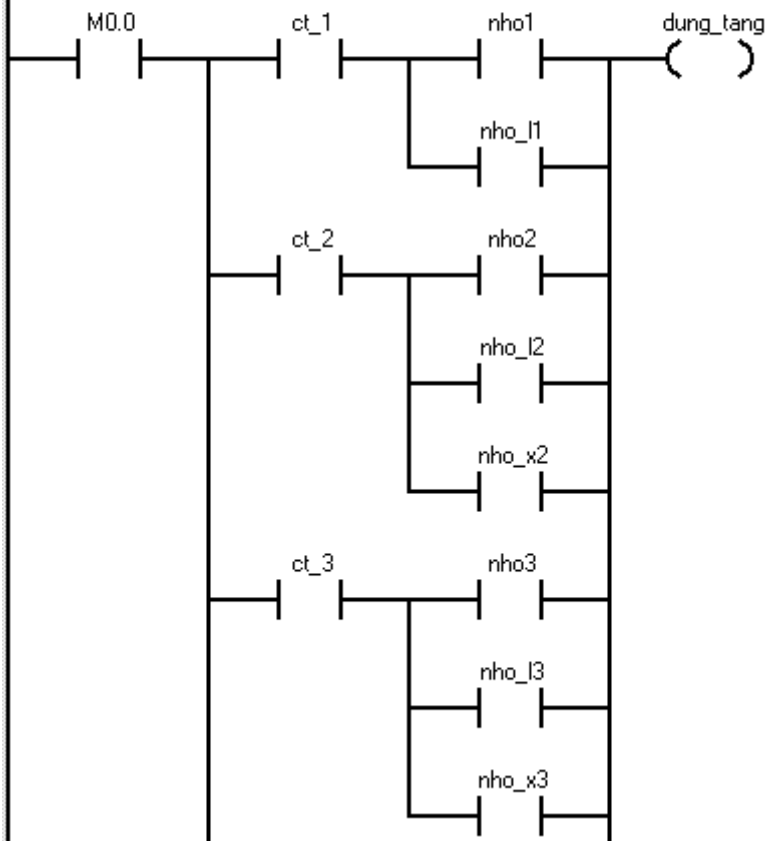
Symbol	Address	Comment
closed	I2.1	cong tac hanh trinh xac dinh cua dong
keo_len	Q0.0	dieu khien dong co keo len
nho1	M1.1	bit nho chon tang 1
nho2	M1.2	bit nho chon tang 2
nho3	M1.3	bit nho chon tang 3
nho_l1	M0.4	bit nho goi len khi dang o tang 1
nho_l2	M0.5	bit nho goi len khi dang o tang 2
nho_l3	M0.6	bit nho goi len khi dang o tang 3

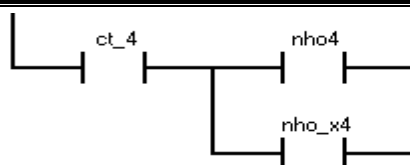


Symbol	Address	Comment
closed	I2.1	cong tac hanh trinh xac dinh cua dong
keo_len	Q0.0	dieu khien dong co keo len
nho1	M1.1	bit nho chon tang 1
nho2	M1.2	bit nho chon tang 2
nho3	M1.3	bit nho chon tang 3
nho_l1	M0.4	bit nho goi len khi dang o tang 1
nho_l2	M0.5	bit nho goi len khi dang o tang 2
nho_l3	M0.6	bit nho goi len khi dang o tang 3
nho_x2	M0.1	bit nho goi xuong khi dang o tang 2
nho_x3	M0.2	bit nho goi xuong khi dang o tang 3
nho_x4	M0.3	bit nho goi xuong khi dang o tang 4

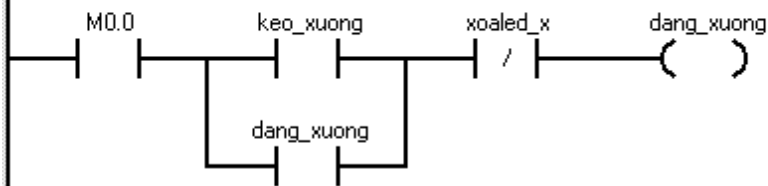
**Network 40**

Symbol	Address	Comment
dung_tang	M5.6	bit nhớ dung tầng
keo_xuong	Q0.1	điều khiển động cơ kéo xuống

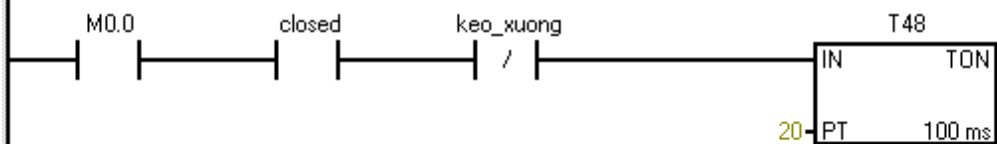
**Network 41**



Symbol	Address	Comment
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
dung_tang	M5.6	bit nho dung tang
nho1	M1.1	bit nho chon tang 1
nho2	M1.2	bit nho chon tang 2
nho3	M1.3	bit nho chon tang 3
nho4	M1.4	bit nho chon tang 4
nho_l1	M0.4	bit nho goi len khi dang o tang 1
nho_l2	M0.5	bit nho goi len khi dang o tang 2
nho_l3	M0.6	bit nho goi len khi dang o tang 3
nho_x2	M0.1	bit nho goi xuong khi dang o tang 2
nho_x3	M0.2	bit nho goi xuong khi dang o tang 3
nho_x4	M0.3	bit nho goi xuong khi dang o tang 4

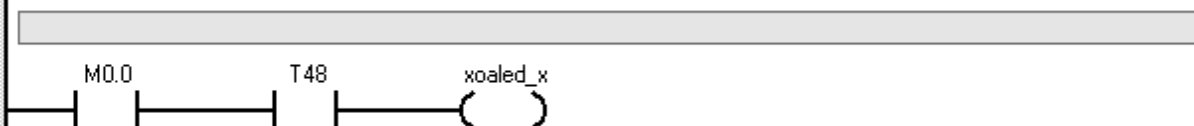
**Network 42**    *hien thi led đi xuống**khi dong co keo len thi den bao led xuống**khi dong co dung ngang tang de don khách nhưng van dang theo chieu đi xuống thì hien thi led xuống*

Symbol	Address	Comment
dang_xuong	Q1.1	<i>hien thi led bao dang đi xuống</i>
keo_xuong	Q0.1	<i>điều khiển động cơ kéo xuống</i>
xoaled_x	M1.7	<i>xóa bit nhớ led xuống</i>

**Network 43**    *sau khi động cơ kéo dừng ngang tầng thì 2s sau xóa led bao lên*

Symbol	Address	Comment
closed	I2.1	<i>công tắc hành trình xác định của động</i>
keo_xuong	Q0.1	<i>điều khiển động cơ kéo xuống</i>

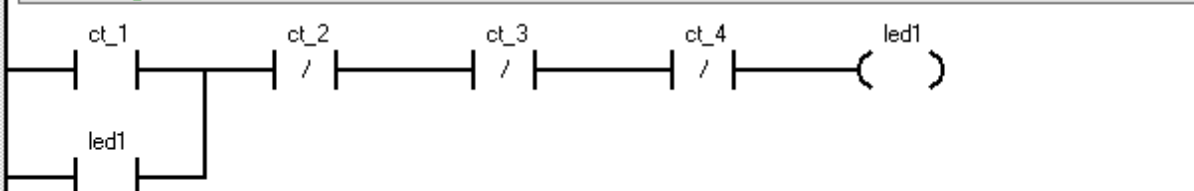


**Network 44**

Symbol	Address	Comment
xoaled_x	M1.7	xoa bit nho led xuong

**Network 45** chương trình điều khiển led hiển thị tầng

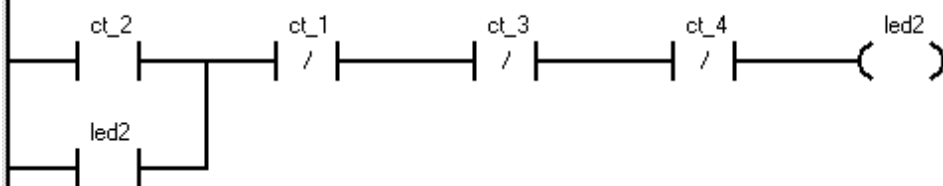
hien thi tang 1



Symbol	Address	Comment
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
led1	Q0.4	hien thi led bao dang o tang 1

**Network 46**

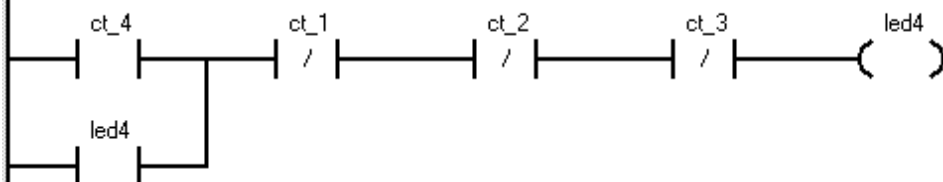
hien thi tang 2



Symbol	Address	Comment
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
led2	Q0.5	hien thi led bao dang o tang 2

**Network 47**

hien thi tang 4



Symbol	Address	Comment
ct_1	I2.2	cong tac hanh trinh xac dinh cabin o tang 1
ct_2	I2.3	cong tac hanh trinh xac dinh cabin o tang 2
ct_3	I2.4	cong tac hanh trinh xac dinh cabin o tang 3
ct_4	I2.5	cong tac hanh trinh xac dinh cabin o tang 4
led4	Q0.7	hien thi led bao dang o tang 4

**Danh mục tài liệu tham khảo:**

1. Tự động hóa trong công nghiệp với S7 & Protool, Phạm Quang Huy-Trần Thu Hà, NXB Hồng Đức, 2008.
2. Lập trình với S7 & WINCC, Trần Thu Hà- Trần Thị Hoàng Oanh- Phạm Quang Huy, NXB Hồng Đức.S7-200 USER'S GUIDE.
3. Tự động hóa với SIMATIC S7-200, Phan Xuân Minh-Nguyễn Duy Phước.
4. OPC Data Access Standard Version 1.0A. OPC Foundation. Austin, Texas, September 11,1997.
5. OPC Data Access Automation Interface Standard Version 2.02. OPC Foundation. Austin, Texas, February 4, 1999.
6. S7-200 Catalogue
7. S7-200 Programmable Controller, SystemManual, SIMATIC.
8. OMRON- Programming.
9. [www.google.com](http://www.google.com)
10. <http://www.Mitsubishi-elevator.com>
11. <http://www.plc-siemens.com>
12. <http://www.support.automation.siemens.com>
13. <http://www.hiendaihoa.com>
14. <http://www.vn.wikipedia.net>
15. Đồ án tham khảo của các khóa trước.