

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ TỰ ĐỘNG HÓA QUÁ TRÌNH SẢN XUẤT

1.1 Giới thiệu	4
1.2 Khái niệm và định nghĩa cơ bản.....	4
1.3 Hệ thống tự động hóa quá trình sản xuất	7

CHƯƠNG 2: CẢM BIẾN VÀ CƠ CẤU CHẤP HÀNH

2.1 Cảm biến	10
2.2 Một số cảm biến phổ biến trong lĩnh vực tự động hóa	14
2.2.1 Cảm biến dịch chuyển thẳng và quay	14
2.2.2 Cảm biến lực	16
2.2.3 Cảm biến khoảng cách	17
2.2.4 Cảm biến quang.....	19
2.2.5 Cảm biến điện dung.....	21
2.2.6 Cảm biến điện cảm (điện từ)	22
2.3 Cơ cấu chấp hành	23
2.3.1 Động cơ điện	23
2.3.2 Hệ thống điều khiển khí nén.....	29

CHƯƠNG 3: BỘ ĐIỀU KHIỂN LẬP TRÌNH PLC

3.1 Giới thiệu	39
3.2 Sự khác nhau giữa hệ điều khiển bằng relay và hệ điều khiển bằng PLC	39
3.3 Cấu trúc của một PLC	42
3.4 Các khối của PLC.....	44
3.4.1 Khối nguồn cung cấp	44
3.4.2 Bộ nhớ chương trình	44
3.4.3 Khối trung tâm (CPU).....	46
3.4.4 Khối vào.....	46
3.4.5 Khối ra.....	46
3.4.6 Các khối đặc biệt	47
3.5 Phương thức thực hiện chương trình trong PLC	47

CHƯƠNG 4: PLC SIMATIC S7-200

4.1 Cấu hình phần cứng.....	49
4.1.1 Khối xử lý trung tâm	49
4.1.2 Khối mở rộng	52
4.1.2.1 Digital module.....	52
4.1.2.2 Analog module	52
4.1.2.3 Intelligent module	53
4.1.2.4 Function module.....	54
4.2 Màn hình điều khiển.....	54
4.3 Các vùng nhớ	55
4.4 Quy ước địa chỉ trong PLC S7-200.....	58
4.4.1 Truy xuất theo bit	58
4.4.2 Truy xuất theo byte (8 bit).....	58
4.4.3 Truy xuất theo word (16 bit)	58
4.4.4 Truy xuất theo 2 word (Double word = 32 bit)	58
4.5 Xử lý chương trình.....	60

CHƯƠNG 5: KẾT NỐI ĐIỆN GIỮA PLC VÀ CÁC THIẾT BỊ NGOẠI VI

5.1 Kết nối dây giữa PLC và các thiết bị ngoại vi.....	62
--	----

5.1.1	Giới thiệu CPU 224 và cách kết nối với thiết bị ngoại vi	62
5.1.2	Kết nối với máy tính.....	62
5.1.3	Nối nguồn cung cấp cho CPU	64
5.1.4	Kết nối vào/ra số với ngoại vi	65
5.1.4.1	Kết nối các ngõ vào số với ngoại vi	65
5.1.4.2	Kết nối các ngõ ra số với ngoại vi	66
CHƯƠNG 6: CÁC PHÉP TOÁN LOGIC VÀ TẬP LỆNH LẬP TRÌNH		
6.1	Ngăn xếp (logic stack) trong S7-200.....	70
6.2	Các phép toán logic cơ bản	70
6.2.1	Phép toán AND	70
6.2.2	Phép toán OR	71
6.2.3	Tổ hợp các cổng AND và OR	71
6.2.3.1	AND trước OR	71
6.2.3.2	OR trước AND	72
6.2.4	Phép toán XOR	73
6.3	Xử lý các tiếp điểm, cảm biến được nối với ngõ vào PLC	73
6.4	Ví dụ ứng dụng các liên kết logic.....	75
6.4.1	Mạch tự duy trì ưu tiên mở máy	75
6.4.2	Mạch tự duy trì ưu tiên dừng máy	76
6.4.3	Điều khiển ON/OFF động cơ có chỉ báo.....	76
6.4.4	Điều khiển đảo chiều quay động cơ	78
6.5	Các lệnh SET, RESET và mạch nhớ RS	80
6.5.1	Lệnh SET	80
6.5.2	Lệnh RESET (R)	80
6.5.3	Mạch nhớ R-S	81
6.5.3.1	Ưu tiên SET (khâu SR)	81
6.5.3.2	Ưu tiên RESET (khâu RS)	82
6.5.4	Các qui tắc khi sử dụng Set và Reset.....	82
6.6	Các lệnh nhận biết cạnh tín hiệu và lệnh NOT	83
6.6.1	Lệnh NOT	83
6.6.2	Các lệnh nhận biết cạnh tín hiệu.....	83
6.7	Các Bit nhớ đặc biệt (Special Memory bits)	84
CHƯƠNG 7: BỘ ĐỊNH THỜI (TIMER) VÀ BỘ ĐẾM (COUNTER)		
7.1	Giới thiệu bộ định thời	85
7.2	Timer đóng mạch chậm TON	85
7.3	Timer đóng mạch chậm có nhớ TONR	86
7.4	Timer mở mạch chậm TOF	87
7.5	Giới thiệu bộ đếm	89
7.6	Bộ đếm lên CTU (Count Up)	89
7.7	Bộ đếm xuống CTD (Count Down)	90
7.8	Bộ đếm lên-xuống CTUD (Count Up/Down)	91
CHƯƠNG 8: PHƯƠNG PHÁP LẬP TRÌNH ĐIỀU KHIỂN TUẦN TỰ		
8.1	Cấu trúc chung của một chương trình điều khiển	93
8.2	Điều khiển tuần tự	93
8.2.1	Giới thiệu	93
8.2.2	Phương pháp lập trình điều khiển tuần tự.....	95
8.3	Các thủ tục tổng quát để thiết kế bài toán tuần tự	96
8.4	Cấu trúc của bài toán điều khiển tuần tự	97
8.4.1	Hệ thống tuần tự nối tiếp.....	98
8.4.2	Hệ thống tuần tự song song.....	100

8.4.3 Hệ thống tuần tự rẽ nhánh có lựa chọn.....	102
8.4.4 Hệ thống tuần tự có vòng lặp... ..	105
CHƯƠNG 9: CÁC CƠ CẤU TỰ ĐỘNG CƠ KHÍ	
9.1 Cơ cấu cấp phối tự động	115
9.2 Bài tập ứng dụng	120

CHƯƠNG 1:

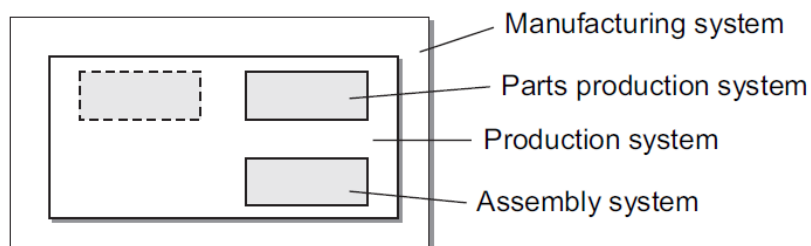
TỔNG QUAN VỀ TỰ ĐỘNG HÓA QUÁ TRÌNH SẢN XUẤT

1.1 Giới thiệu

Những cuộc cách mạng công nghệ đã giúp nâng cao chất lượng cuộc sống. Một trong những khía cạnh ưu việt đó là trong lĩnh vực chế tạo máy và trong sản xuất hàng hóa. Tự động hóa các trang thiết bị trong các công ty, nhà máy, xí nghiệp ngày càng phổ biến và phát triển với qui mô lớn, yêu cầu độ chính xác cao, giúp giải phóng sức lao động, nâng cao cả chất lượng và sản lượng đáp ứng nhu cầu cuộc sống của con người. Việt Nam, một nước đang phát triển lại càng cần thiết sự tự động hóa cao nhất là trong lĩnh vực tự động hóa quá trình sản xuất, giúp người lao động nâng cao hiệu suất làm việc và tránh được những công việc nặng nhọc, nguy hiểm đến sức khỏe. Ở đất nước có cường độ làm việc và độ chính xác cao, con người không thể đảm nhiệm được, lúc đó máy móc sẽ thay con người nhưng dưới sự giám sát của con người. Vì vậy, việc tự động hóa các hệ thống sản xuất với trang thiết bị hiện đại là điều rất cần thiết.

1.2 Khái niệm và định nghĩa cơ bản

Hệ thống: Ngay từ những năm 1990, nhu cầu về một quan điểm toàn diện về hệ thống sản xuất là cần thiết (ví dụ như Rampersad 1994; Wu 1994; Bellgran 1998). Một quan điểm toàn diện về hệ thống sản xuất là hệ thống nên được thiết kế với các bộ phận kỹ thuật và vật lý, con người trong các hệ thống, và cách tổ chức công việc, xem xét (Bennett 1986). Để tạo thuận lợi, quan điểm toàn diện dựa trên lý thuyết hệ thống, các hệ thống sản xuất. Tầm quan trọng tổng thể của cả hệ thống được nhấn mạnh khi một hệ thống quan điểm lý thuyết được áp dụng cho hệ thống sản xuất. Với sự hỗ trợ từ một hệ thống quan điểm lý thuyết tất cả các bộ phận được lấy vào xem xét và tác động lẫn nhau giữa các bộ phận khác nhau của sản xuất. Ngày nay, Các khái niệm về hệ thống đã trở nên ngày càng phổ biến để mô tả hoạt động và hiện tượng trong các tình huống khác nhau (Lind 2001). Do đó, khái niệm hệ thống thường xuất hiện trong sự kết hợp với các thuật ngữ khác, như trong trường hợp hệ thống sản xuất, hệ thống chế tạo (gia công) và hệ thống lắp ráp (hình 1.1). Hệ thống tồn tại ở khắp mọi nơi và bất chấp sự khác biệt tất cả các hệ thống chia sẻ một số cấu trúc cơ bản chung. Như một hệ quả của hệ thống quan điểm lý thuyết đã được phát triển như là một cách giải thích hệ thống một cách khoa học (Wu 1994): "... Các ngành công nghiệp sản xuất đang rời khỏi khái niệm tuổi công nghệ được đặc trưng bằng máy móc, và đang trong quá trình chuyển sang đặc trưng tuổi của các hệ thống. " (Wu 1994). Từ đây có thể thấy nhận thức hệ thống rất hữu ích để tăng sự hiểu biết của một hệ thống sản xuất phức tạp. Để thành công phát triển và vận hành hệ thống sản xuất hiệu biết tốt về các thành phần của một hệ thống sản xuất và làm thế nào các thành phần tương tác là điều cần thiết.

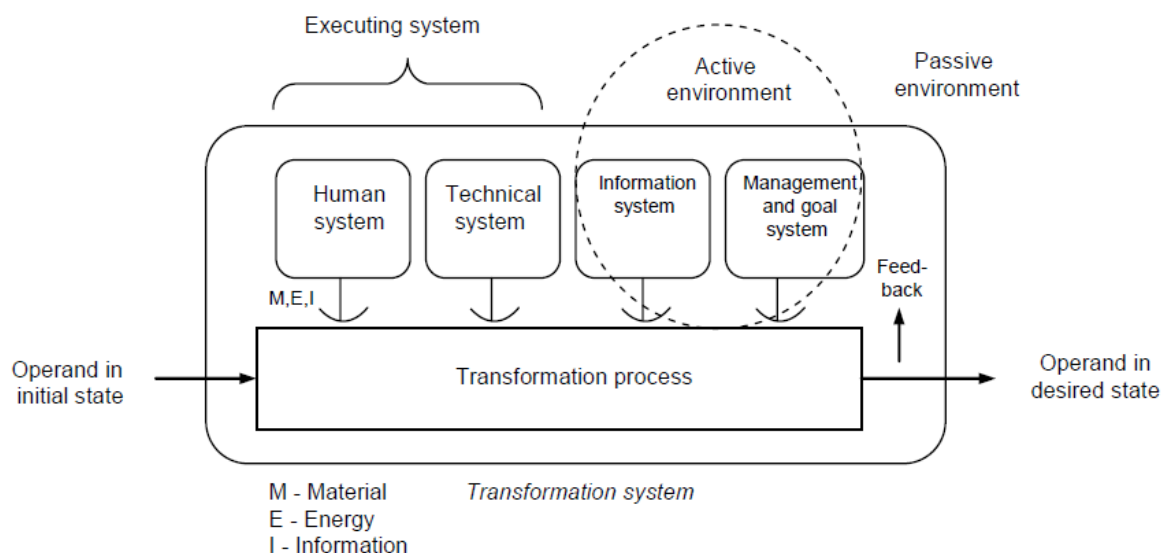


Hình 1.1 Kiến trúc của hệ thống sản xuất

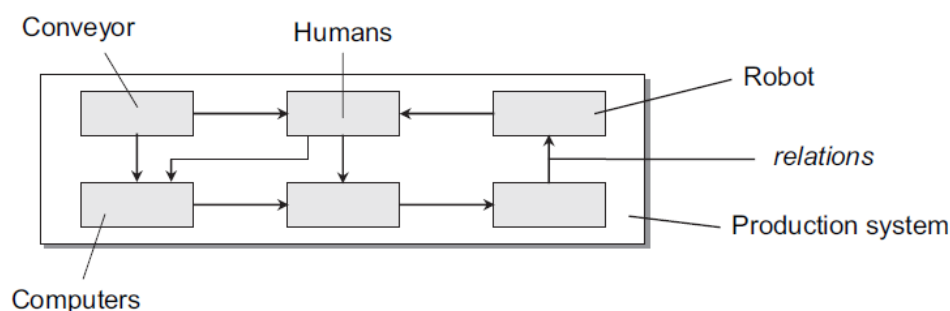
Hệ thống sản xuất

Quá trình tạo ra hàng hóa và / hoặc dịch vụ thông qua một sự kết hợp của vật liệu, công việc, và vốn được gọi là sản xuất. Sản xuất có thể là bất cứ thứ gì từ sản xuất hàng tiêu dùng, sản xuất dịch vụ của một công ty tư vấn, âm nhạc hoặc sản xuất năng lượng. Có mối liên hệ rõ ràng giữa sản xuất hàng hóa và dịch vụ. Sự tiêu thụ tạo động lực cao cấp cho mọi sản xuất. Hàng hóa sản xuất phải bằng cách nào đó được phân phối cho tiêu dùng. Do đó sản xuất hàng hoá thường không quan tâm, nếu không kết hợp với sản xuất dịch vụ, ví dụ như trong khu vực hậu cần (Mattsson và Jonsson 2003). Tuy nhiên loại hình cụ thể của sản xuất được nêu trong tài liệu này là hoạt động sản xuất công nghiệp. Giới hạn trong sản xuất hàng hóa, nơi mà sự chuyển đổi của nguyên liệu thành sản phẩm được thực hiện trong một hệ thống sản xuất như hình 1.2.

Một hệ thống sản xuất bao gồm một số yếu tố mà giữa chúng có quan hệ đối ứng. Yếu tố thường được đề cập đến là các địa điểm, con người, máy móc, và thiết bị (Lofgren 1983). Phần mềm và qui trình có thể được thêm vào các thành phần hệ thống Chapanis (1996). Một quan điểm cấu trúc của các hệ thống sản xuất có thể được sử dụng để mô tả các yếu tố hệ thống khác nhau và mối quan hệ của họ, xem hình 1.3.



Hình 1.2 Mô tả chức năng của một hệ thống sản xuất

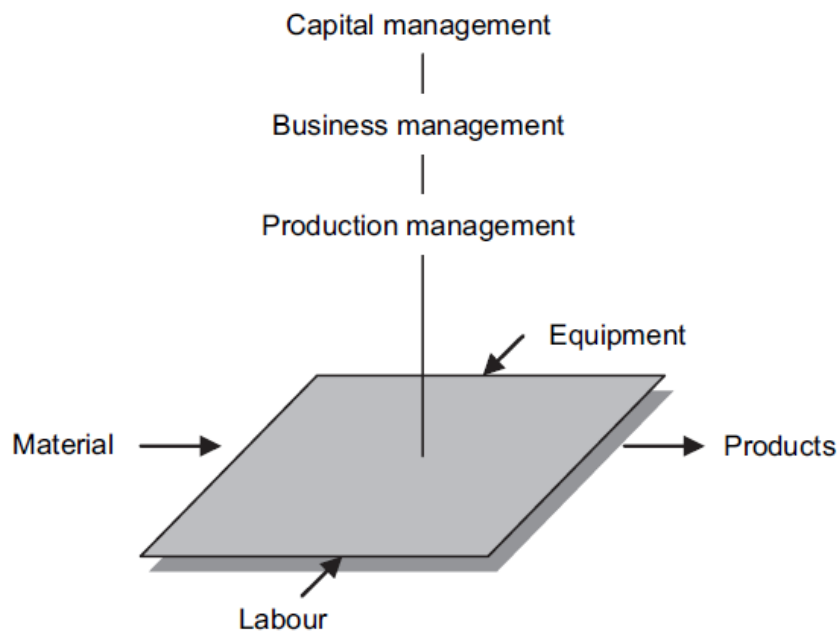


Hình 1.3 Các thành phần cơ bản của một hệ thống sản xuất

Tuy nhiên, một không gian khác có thể được thêm vào mô tả của một hệ thống sản xuất, quá trình ra quyết định. Quá trình ra quyết định cho một hệ thống sản xuất thêm quản lý vốn (chủ sở hữu), quản lý sản xuất kinh doanh quản lý để mô tả một hệ thống sản xuất (Sandkull và Johansson 2000).

Chu kỳ hoạt động của hệ thống sản xuất.

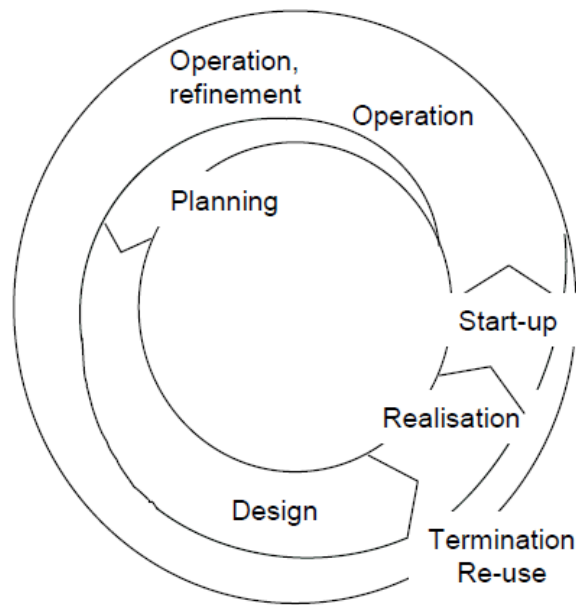
Các hoạt động chính trong một hệ thống sản xuất thường được mô tả dựa trên vòng đời sản phẩm (Foresight kỹ thuật năm 2003):



Hình 1.4 Mô hình của một hệ thống sản xuất bao gồm quá trình ra quyết định

- Các hoạt động thị trường tạo ra nhu cầu về các sản phẩm tạo ra từ hệ thống sản xuất và cung cấp mức độ yêu cầu chất lượng và năng suất hệ thống sản xuất bao gồm điều kiện tiên quyết về thời gian để phát triển, chất lượng sản phẩm và chi phí;
- Các hoạt động kỹ thuật kiểm soát sự phát triển sản phẩm là điều kiện tiên quyết cho các hệ thống sản xuất;
- Các hoạt động sản xuất tạo ra các sản phẩm trong hệ thống sản xuất; các hoạt động phân phối làm cho chắc chắn rằng sản phẩm được giao theo đúng các điều kiện cho khách hàng;
- Các hoạt động dịch vụ nhằm mục đích loại bỏ và ngăn ngừa khuyết tật mà có thể xuất hiện trong sản phẩm;
- Các hoạt động tái chế nhằm tiết kiệm tài nguyên và xử lý vật liệu bị mài mòn nhanh.

Các hệ thống sản xuất cũng có chu kỳ hoạt động riêng. Từ lập kế hoạch ban đầu cho thiết kế đến giai đoạn chế tạo và vận hành. Sự gia tăng nhu cầu từ môi trường bao gồm từ người tiêu dùng và từ các yêu cầu của các tổ chức chính phủ,... có tính pháp lý, đặt ra các yêu cầu cao hơn về tài sử dụng, không chỉ trong những sản phẩm được sản xuất, mà còn của các hệ thống sản xuất. Vì vậy, liên quan đến lập kế hoạch cho nhiều thế hệ sản phẩm cũng như hệ thống thế hệ khi thiết kế hệ thống sản xuất. Các yêu cầu thay đổi từ môi trường đã góp phần vào sự thay đổi có tính tuần tự và song song của vòng đời các hệ thống sản xuất.



Hình 1.5 Chu kỳ hoạt động của một hệ thống sản xuất

Hệ thống sản xuất mới được thiết kế và thực hiện song song với hệ thống cũ vẫn còn hoạt động, trong đó cung cấp các cơ hội tốt để ứng dụng các kinh nghiệm. Bản chất của các hệ thống sản xuất là khác nhau trong vòng đời khác nhau theo các giai đoạn, cũng như các yêu cầu đặt ra cho khả năng đáp ứng của hệ thống. Vì thế, nhận diện của các hệ thống sản xuất vị trí và vai trò hiện tại cũng như đang trong giai đoạn nào trong chu kỳ hệ thống sản xuất để đạt được tính tự động hóa cao. Hiệu quả sản xuất cũng được quan tâm khi vòng đời của một hệ thống sản xuất được xem xét. Hiệu quả sản xuất thì thường được đo trong giai đoạn hoạt động. Nếu hiệu quả sản xuất được xác định ngay từ khi bắt đầu của giai đoạn lập kế hoạch và thiết kế sẽ có nhiều cơ hội nâng cao hiệu quả ngay trong giai đoạn phát triển ban đầu của hệ thống sản xuất.

1.3 Hệ thống tự động quá trình sản xuất

Ngày nay có rất nhiều hệ thống tự động hóa quá trình sản xuất được phân loại dựa trên các khía cạnh khác nhau như dựa trên ứng dụng, loại sản phẩm chế tạo, các dòng sản phẩm của nhà cung cấp và theo xu thế phát triển công nghệ được ứng dụng như hình 1.6-1.8.



Hình 1.6 Dây chuyền lắp ráp xe hơi với các trạm là robot Kuka



Hình1.7 Hệ thống tự động hóa trong nhà máy sản xuất xi măng

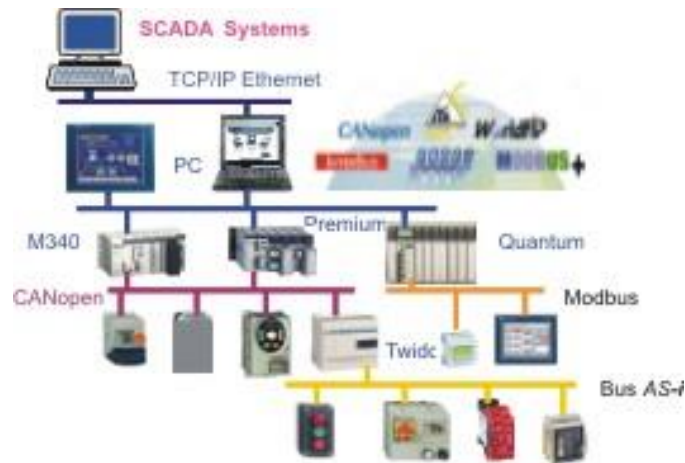


Hình1.8 Hệ thống tự động hóa đóng gói bao bì trong nhà máy sản xuất sản phẩm từ sữa

Tài liệu này giới thiệu một khía cạnh phát triển hệ thống tự động quá trình sản xuất dựa trên sự phát triển của các hệ thống điều khiển nơi mà các kỹ thuật điều khiển cũ được dần thay thế bởi các phương pháp điều khiển ngày càng hoàn thiện cho độ chính, xác, ổn định và tin cậy cao.

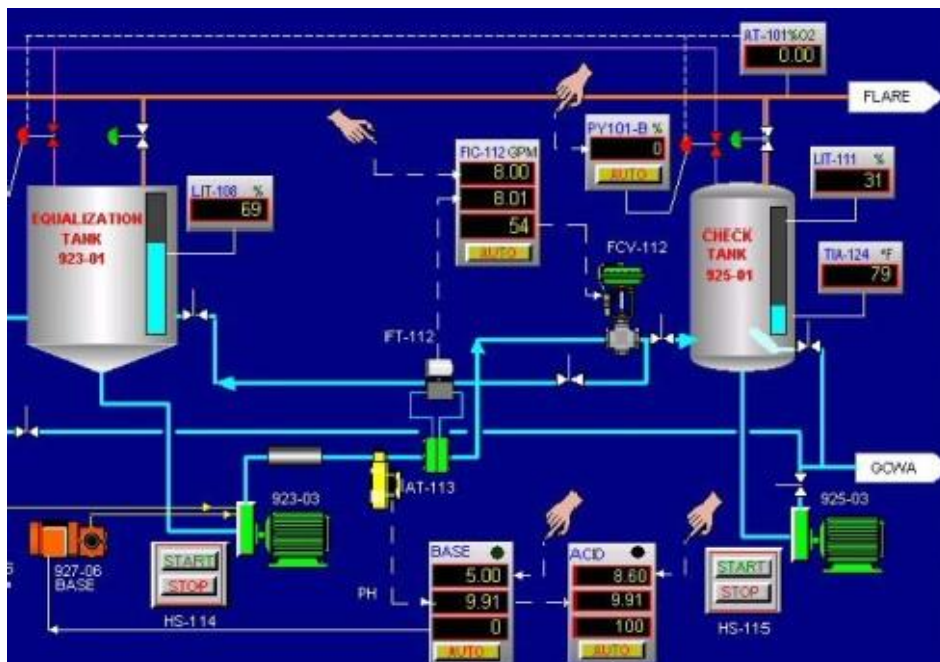


Hình1.9 Bộ điều khiển lập trình (PLC)1500 thế hệ mới của Siemens



Hình1.10 Mô hình cấu trúc mạng hệ thống Đo lường, điều khiển và giám sát (SCADA)

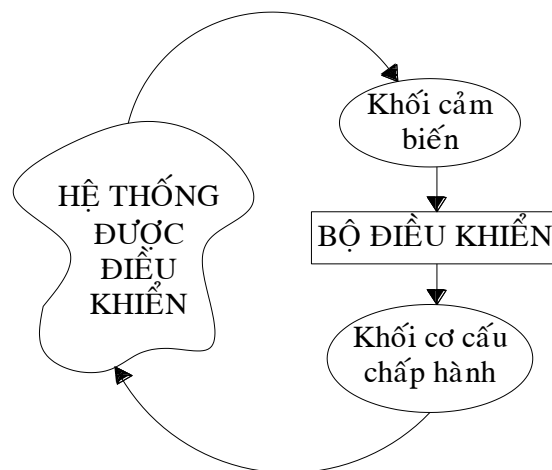
Các thiết bị cơ bản bao gồm các thiết bị cảm biến, cơ cấu chấp hành, mạng truyền thông và các bộ điều khiển luôn được cải tiến, cập nhật và áp dụng các thành tựu mới nhất về công nghệ, như hình 1.9-1.11 nhằm đáp ứng thỏa mãn các yêu cầu phù hợp với hoạt động sản xuất phù hợp với xu thế phát triển của nhân loại.



Hình1.11 Giao diện hệ thống đo lường, điều khiển và giám sát (SCADA) trong công nghiệp

CHƯƠNG 2: CẢM BIẾN VÀ CƠ CẤU CHẤP HÀNH

Cảm biến và cơ cấu chấp hành là hai thành phần cơ bản của một hệ thống điều khiển phổ biến trong các hệ thống tự động hóa sản xuất. Hình 2.1 bao gồm một khối cảm biến, một khối điều khiển và một khối cơ cấu chấp hành. Khối cảm biến có thể đơn giản là một cảm biến đơn lẻ hoặc có thể gồm các thành phần bổ sung như bộ lọc, bộ khuếch đại, bộ điều chế và các bộ biến đổi tín hiệu khác. Khối điều khiển nhận thông tin từ khối cảm biến, đưa ra quyết định dựa trên thuật toán điều khiển và các lệnh tới khối cơ cấu chấp hành. Khối cơ cấu chấp hành bao gồm cơ cấu chấp hành có thể thêm một bộ nguồn và một cơ cấu ghép nối



Hình 2.12 Hệ thống Cơ điện tử thường gặp trong hệ thống tự động hóa.

2.1 Cảm Biến.

Cảm biến là một thiết bị mà khi có một hiện tượng vật lý tác động vào (nhiệt độ, lực, ánh sáng,...) sẽ tạo ra tín hiệu đầu ra (điện, cơ học, từ,...) tỷ lệ. Cảm biến được phân loại thành 2 dạng tương tự hoặc số dựa trên dạng tín hiệu đầu ra. Cảm biến tương tự cung cấp tín hiệu liên tục tỷ lệ với tham số cần đo và cần sự biến đổi tương tự thành số trước khi chuyển cho bộ điều khiển số. Trong khi đó, cảm biến số cung cấp đầu ra số có thể trực tiếp ghép nối với bộ điều khiển số.

Phân loại cảm biến

Một số loại cảm biến thường gặp phân loại theo ứng dụng như:

- Công tắc hành trình
- Cảm biến dịch chuyển thẳng và quay.
- Cảm biến gia tốc.
- Cảm biến lực.
- Cảm biến đo mômen và công suất.
- Cảm biến lưu lượng.
- Cảm biến nhiệt độ.
- Cảm biến đo khoảng cách.
- Các cảm biến nhận biết ánh sáng, hình ảnh và nhận dạng.

Hoặc phân loại theo nguyên lý biến đổi vật lý, hóa học như

- Cảm biến điện dung
- Cảm biến điện từ.
- Cảm biến quang.
- Cảm biến siêu âm.
- Cảm biến hiệu ứng Hall.

Tiêu chuẩn lựa chọn.

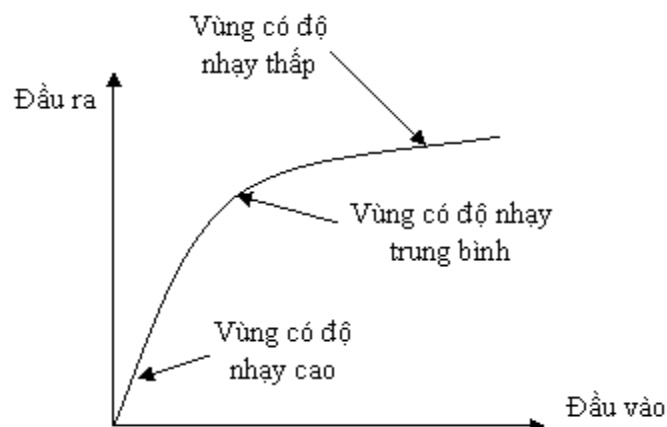
- Dải đo:
- Độ phân giải:
- Độ chính xác:
- Tính chính xác:
- Độ nhạy:
- Thời gian đáp ứng:
- Nhiệt độ hoạt động:
- Vùng chết:

Dải đo: chênh lệch giữa giá trị lớn nhất và giá trị nhỏ nhất của tham số cần đo. Dải đo thường được quy định bởi nhà sản xuất cảm biến.

Độ phân giải: của cảm biến là khoảng thay đổi nhỏ nhất của đầu vào mà cảm biến có thể đo được. Độ phân giải cũng có thể được hiểu là số đếm nhỏ nhất của cảm biến. Ví dụ một bộ mã hóa 1024 ppr (xung/vòng) có độ phân giải là:

$$\frac{360^\circ}{1024 \text{ xung}} = 0.3516 \frac{\text{độ}}{\text{xung}}$$

Độ nhạy: của cảm biến được định nghĩa là tỷ số giữa thay đổi đầu ra trên một đơn vị thay đổi đầu vào. Độ nhạy của cảm biến thường liên quan mật thiết đến độ phân giải. Một cảm biến có đặc tính tuyến tính thì có độ nhạy không đổi trên toàn bộ dải đầu vào. Các cảm biến có đặc tính phi tuyến thì có độ nhạy tăng hoặc giảm khi đầu vào thay đổi như hình 2.2.



Hình 2.13 Độ nhạy của cảm biến.

Sai số: sai số là độ sai khác giữa giá trị đo được và giá trị thực của đầu vào. Có hai loại sai số là sai số hệ thống và sai số ngẫu nhiên. Sai số hệ thống có ở tất cả các phép đo được thực hiện với cảm biến. Các dạng của sai số hệ thống:

- *Sai số lệch không:* là dạng phổ biến của sai số hệ thống khi giá trị đầu ra khác không với đầu vào bằng không.
- *Sai số tải:* do thêm cảm biến vào hệ đo làm thay đổi hệ.
- *Sai số do độ nhạy của cảm biến thay đổi không giống như mong muốn.*

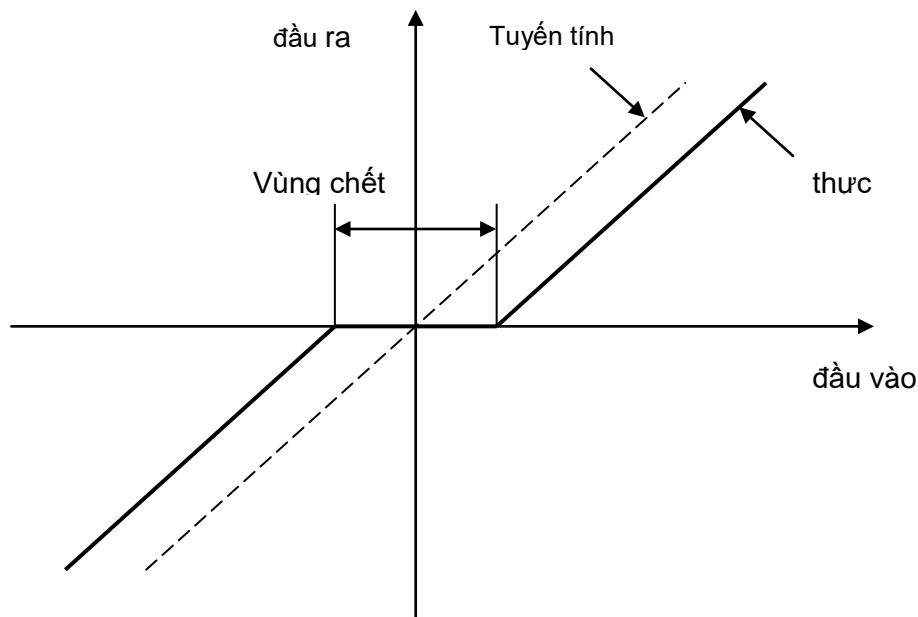
Khả năng lặp lại: là khả năng thu được đầu ra giống nhau với đầu vào giống nhau của một cảm biến. Sai số ngẫu nhiên làm giảm khả năng lặp lại. Sai số ngẫu nhiên có thể khắc phục bằng cách lấy trung bình của một số phép đo. Nhiễu cũng làm giảm khả năng lặp lại.

Vùng chết: là một vùng đầu vào gần điểm không mà đầu ra vẫn giữ giá trị không. Khi đầu vào dịch ra khỏi vùng chết thì đầu ra sẽ thay đổi theo đầu vào (hình 2.3). Vùng chết thường được dùng trong bộ ổn nhiệt gia đình và các bộ điều khiển quá trình.

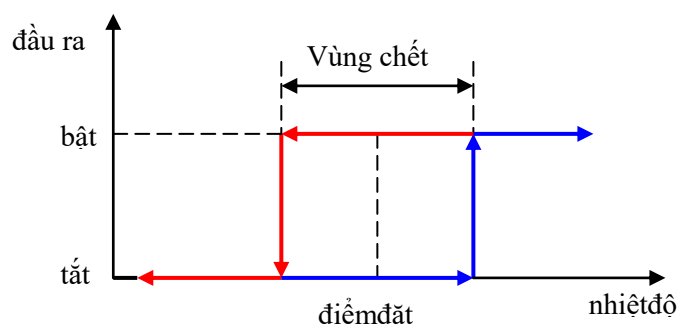
Ví dụ về bộ ổn nhiệt gia đình được mô tả trên hình 2.4.

Khi nhiệt độ đạt điểm đặt (giá trị mong muốn) trên bộ ổn nhiệt, đầu ra vẫn giữ trạng thái tắt. Ngay khi nhiệt độ trong phòng tăng lên tới giá trị điểm đặt cộng một nửa giá trị vùng chết thì đầu ra hệ thống làm mát sẽ bật hoàn toàn.

Khi phòng lạnh đi, đầu ra sẽ giữ trạng thái bật đến khi nhiệt độ trong phòng đạt điểm đặt trừ đi một nửa giá trị vùng chết, đầu ra hệ thống làm lạnh sẽ tắt hoàn toàn.



Hình 2.14 Vùng chết.

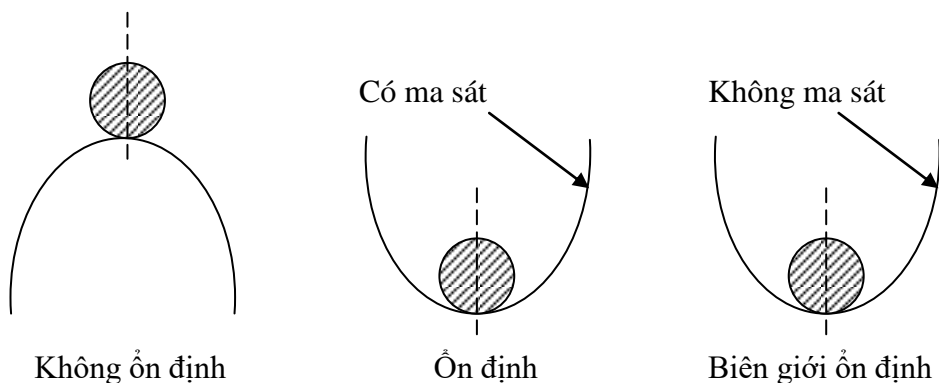


Hình 2.15 Vùng chết của bộ ổn nhiệt.

Tính ổn định: thuật ngữ tính ổn định có nhiều định nghĩa và cách dùng khác nhau nhưng định nghĩa phổ biến nhất là định nghĩa liên quan đến cân bằng. Một hệ cân bằng sẽ giữ được trạng thái không đổi khi không có nhiễu tác động. Một hệ ổn định sẽ trở lại trạng thái cân bằng nếu có nhiễu nhỏ làm dịch chuyển hệ thống khỏi trạng thái ban đầu. Một hệ không ổn định sẽ không trở lại vị trí cân bằng và thường dịch chuyển xa vị trí cân bằng.

Hình 2.5 cho thấy ba điều kiện ổn định với một hệ đơn giản gồm: quả bóng và đồi. Trong mỗi trường hợp, vị trí cân bằng được xác định dễ dàng – trên đỉnh đồi hoặc dưới đáy. Trong trường hợp ổn định, một dịch chuyển nhỏ của quả bóng ra khỏi vị trí cân bằng, nó sẽ quay trở về vị trí cân bằng, có thể sau một vài dao động. Trong trường hợp thứ ba, trường hợp không có ma sát làm quả bóng dao động liên tục quanh vị trí cân bằng sau khi có một chuyển động nhỏ. Trường hợp đặc biệt này gọi là biên ổn định do hệ thống không bao giờ thực sự trở lại vị trí cân bằng.

Hầu hết các cảm biến và cơ cấu chấp hành vốn đã ổn định. Tuy nhiên, việc thêm các hệ điều khiển chủ động có thể làm hệ các thiết bị ổn định trở thành không ổn định. Cần phân tích và kiểm tra cẩn thận để đảm bảo rằng một hệ cơ điện tử hoạt động trong trạng thái ổn định.



Hình 2.16 Tính ổn định của hệ thống.

Thời gian đáp ứng: là thời gian trễ giữa đầu vào và đầu ra. Cảm biến càng tốt thì thời gian đáp ứng càng nhỏ.

Nhiệt độ hệ thống: là khoảng nhiệt độ mà tại đó cảm biến còn có thể hoạt động được.

Việc chọn cảm biến thỏa mãn tất cả các đặc tính kỹ thuật trên là không thực tế. Ví dụ việc tìm cảm biến vị trí với độ phân giải μm trong khoảng một mét gần như không đáp ứng được với hầu hết các loại cảm biến. Do vậy tùy theo hệ Cơ điện tử thiết kế mà ta lựa chọn cảm biến cho phù hợp.

Khi các hệ số chức năng trên được thỏa mãn, người ta có thể đưa ra một danh sách cảm biến. Sự lựa chọn cuối cùng sẽ phụ thuộc vào kích cỡ, độ tin cậy, độ bền vững, khả năng bảo dưỡng và giá thành cảm biến.

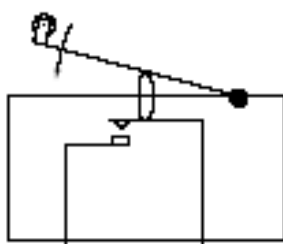
2.2 Một số cảm biến phổ biến trong lĩnh vực tự động hóa

2.2.1 Cảm biến dịch chuyển thẳng và quay.

Hầu hết các chuyển động phổ biến nhất trong các hệ thống cơ khí là chuyển động thẳng dọc theo một trục cố định và quay một góc quanh trục cố định. Những chuyển động phức tạp hơn thường được tạo ra bởi việc kết hợp những chuyển động đơn giản đó. Trong phần này chúng ta sẽ giới thiệu tóm tắt một số công nghệ sẵn có để đo các chuyển động thẳng và chuyển động quay quanh một trục.

Chuyển mạch tiếp xúc (công tắc, công tắc hành trình).

Loại đơn giản nhất của cảm biến dịch chuyển là một công tắc hành trình, nó gửi trả một bit thông tin: chạm hoặc không chạm. Một công tắc hành trình điển hình bao gồm một đòn bẩy, khi nó được tác động nó sẽ tạo ra một tiếp xúc cơ khí bên trong công tắc, từ đó tạo ra một mạch điện kín (hình 2.6). Có thể sử dụng công tắc này như là những cảm biến va chạm.



Hình 2.17 Công tắc hành trình



Hình 2.18 Một số dạng công tắc hành trình

Chuyển mạch tiếp xúc thường có hai dạng là thường hở (normally open) và thường đóng (normally closed). Phần tiếp xúc được gia cố để có thể chịu được lực cơ tác động nhiều lần.

Công tắc từ (Reed Switches)

Công tắc từ thì rất giống relay. Tuy nhiên ở đây một nam châm vĩnh cửu được sử dụng thay thế cho cuộn dây. Khi nam châm ở xa thì tiếp điểm mở. Khi nam châm đến gần thì

tiếp điểm đóng lại. Các cảm biến này thường được dùng làm công tắc hành trình không tiếp xúc cho các xy lanh khí nén.

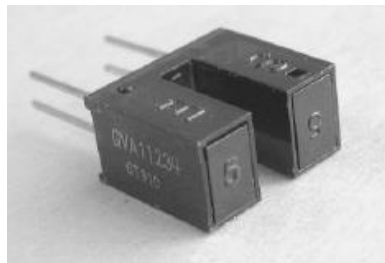


Hình 2.19 Công tắc từ

Tia hồng ngoại.

Tia hồng ngoại có thể được sử dụng để đo các dịch chuyển thẳng hoặc dịch chuyển quay. Thông thường các điôt phát tia hồng ngoại (LED) hoặc là đèn phát quang (Thiết bị phát) được dùng như một nguồn phát sáng và một thiết bị cảm nhận hồng ngoại được dùng để phát hiện tia sáng (thiết bị thu).

Nếu thiết bị phát và thiết bị thu đặt đối diện nhau thì chúng có thể được sử dụng như một bộ ngắt tia để phát hiện bất kỳ vật gì đi qua chúng. Thiết bị này được gọi là *thiết bị ngắt quang* (hình 2.9).



Hình 2.20 Bộ ngắt quang dẫn QVA11234

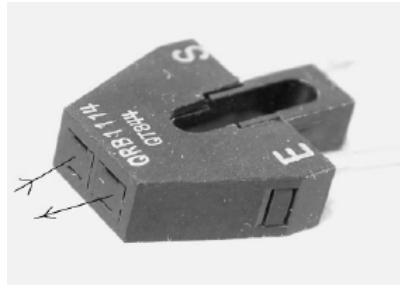
Nếu thiết bị phát và thu di chuyển tự do theo đường thẳng nối chúng, thì cường độ của tín hiệu nhận được có thể được sử dụng để đo khoảng cách giữa chúng. Tuy nhiên thiết bị thu tia hồng ngoại cũng có thể nhạy cảm với cả ánh sáng của môi trường xung quanh. Do vậy trong quá trình thiết kế ta phải để ý đến hiện tượng này để làm cho cảm biến chống nhiễu một cách tốt nhất.

Một bộ phát sáng và bộ dò đặt cùng hướng về một phía có thể đo thô khoảng cách tới một bề mặt gần đó nhờ cường độ của tia phản xạ sau khi va vào bề mặt. Thiết bị đó được gọi là *thiết bị phản xạ quang* (hình 2.9). Một cảm biến như vậy có thể được dùng cho các robot di động để phát hiện các vật cản ở một khoảng cách cho trước.

Các thiết bị ngắt quang và các thiết bị phản xạ quang có thể được đóng gói sẵn hoặc chế tạo riêng biệt từ một LED hồng ngoại và một điôt quang hoặc tranzito quang, sau khi chắc chắn rằng thiết bị thu nhạy cảm với bước sóng sinh ra bởi LED phát hồng ngoại.

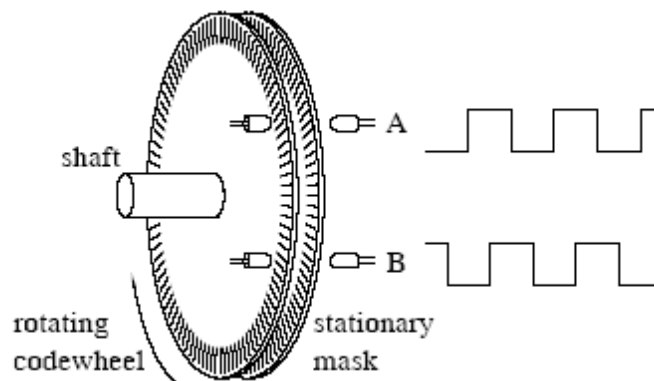
Các bộ mã hóa quang học.

Một bộ mã hóa quang sử dụng một bộ ngắt quang để biến chuyển động thành một chuỗi xung điện. Những chuỗi xung này “mã hóa” chuyển động và các xung được đếm hoặc “được giải mã” bởi một mạch điện để đưa ra số đo dịch chuyển. Chuyển động có thể là thẳng hoặc quay, nhưng chúng ta tập trung vào các bộ mã hóa quang quay thông thường.



Hình 2.21 Cảm biến phản xạ quang bán dẫn QRB1114.

Có hai bộ mã hóa quang quay cơ bản là: mã hóa tương đối và bộ mã hóa tuyệt đối. Trong bộ mã hóa tương đối, một đĩa (đĩa chia vạch) được gắn vào một trục quay tròn giữa hai bộ ngắt quang học (hình 2.10). Vì vậy khi trục quay thì các vạch này sẽ ngăn hoặc cho tia sáng hồng ngoại tới thiết bị dò quang học. Các dãy xung thu được từ các bộ dò này có tần số tương ứng với vận tốc góc của đĩa. Những tín hiệu này được ký hiệu là A và B. Hai tín hiệu A và B này lệch pha nhau $\frac{1}{4}$ chu kỳ.



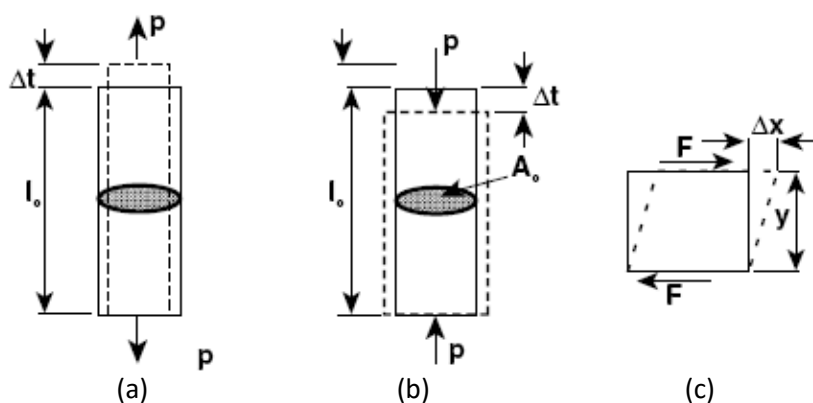
Hình 2.22 Bộ mã hóa tương đối.

Thông qua việc đếm số lượng các xung và số lượng các vạch trên đĩa đã biết, ta có thể đo được tốc độ quay của trục. Hướng quay được xác định dựa trên quan hệ pha của các chuỗi xung A và B. Ví dụ A tăng khi B = 1 ta có thể nói rằng chuyển động của đĩa là chuyển động quay ngược chiều kim đồng hồ. Ngược lại A tăng khi B = 0 thì chuyển động của đĩa là chuyển động quay cùng chiều kim đồng hồ.

2.2.2 Cảm biến lực.

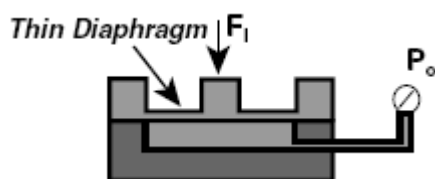
Lực là một đại lượng vector, được định nghĩa như là tác nhân gây ra gia tốc hoặc phản ứng cụ thể của một vật. Trong phần này chúng ta sẽ giới thiệu các phương pháp được ứng dụng để xác định độ lớn của những lực này.

Nhận xét chung: Nếu các lực tác dụng lên một vật mà không sinh ra gia tốc, chúng phải có một dạng hệ lực cân bằng. Hệ này được xem như là một hệ cân bằng tĩnh. Các lực tác dụng lên vật có thể chia làm hai loại: nội lực do các phần tử riêng biệt của vật tác dụng lẫn nhau và ngoại lực.



Hình 2.23 Vật bị kéo dọc trục (a), nén dọc trục (b), biến dạng (c).

Trong kỹ thuật đo hiện có, một nhóm được gọi là cảm biến tải. Nhìn chung cảm biến tải gồm một khung cứng, một môi trường để đo các lực đặt vào và một đầu đo. Các cảm biến lực được sử dụng để đo các lực lớn, tĩnh hoặc biến thiên chậm với độ lệch rất ít và tương đối chính xác. Có thể áp dụng các phương pháp khác nhau để đo các lực tùy thuộc vào thiết kế của cảm biến lực. Ví dụ hình 2.13 minh họa cảm biến lực thủy lực. Cảm biến lực dùng để đo các lực tương đối lớn với giá thành thấp. Cảm biến lực thủy lực dùng một khung rất cứng với một khoang trong chứa đầy chất lỏng. Khi có tải, áp lực của dầu tăng, và một đầu đo chính xác sẽ đọc giá trị này.



Hình 2.24 Cảm biến lực thủy lực.

2.2.3 Cảm biến đo khoảng cách.

Cảm biến này được sử dụng để đo khoảng cách từ một điểm tham chiếu tới một đối tượng. Rất nhiều công nghệ khác nhau đã được ứng dụng để phát triển các loại cảm biến này, tiêu biểu là ánh sáng/quang học, hình ảnh, vi sóng và siêu âm. Cảm biến đo khoảng cách có thể được phân làm hai loại: Tiếp xúc và không tiếp xúc. Ta chủ yếu đề cập đến loại không tiếp xúc.

Cảm biến đo khoảng cách không tiếp xúc là loại cảm biến đo khoảng cách thực từ điểm tham chiếu tới một đối tượng không qua tiếp xúc vật lý. Có thể được phân làm hai loại là: chủ động (phát một số dạng năng lượng vào khu vực cần quan tâm) và bị động (dựa trên năng lượng phát ra từ các đối tượng trong khu vực quan tâm).

Với các loại cảm biến chủ động (phản xạ), khoảng cách đo hiệu quả phụ thuộc không chỉ vào mức năng lượng phát ra mà còn phụ thuộc vào các đặc tính sau của đối tượng:

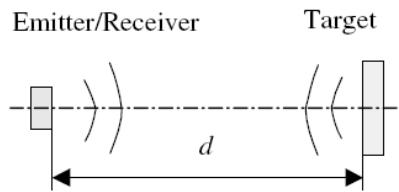
- *Diện tích tiết diện ngang*: xác định lượng năng lượng phát ra tác động vào đối tượng.
- *Hệ số phản xạ*: xác định lượng năng lượng truyền tới được phản xạ so với lượng năng lượng bị hấp thụ hoặc xuyên qua.

- *Độ tập trung*: xác định khả năng phân bố lại của năng lượng phản xạ.

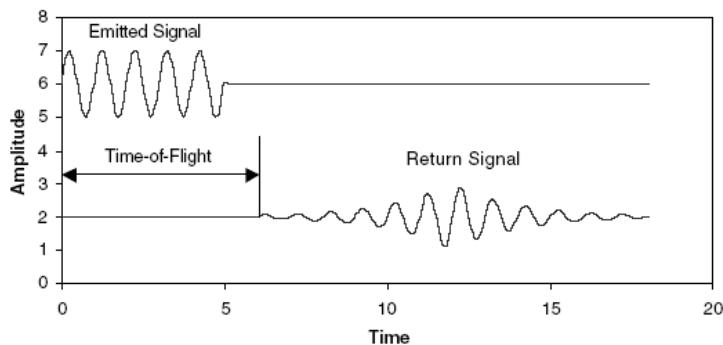
Rất nhiều cảm biến không tiếp xúc hoạt động dựa trên hiện tượng truyền sóng. Sóng được phát ra tại một điểm tham chiếu, khoảng cách được xác định bằng cách đo thời gian truyền từ điểm tham chiếu tới vật hoặc suy giảm của cường độ khi sóng truyền tới vật và quay trở lại điểm tham chiếu. Thời gian truyền sóng thường được đo bằng *phương pháp thời gian truyền* (TOF_Time of flight).

Phương pháp thời gian truyền:

Phương pháp thời gian truyền được minh họa trong hình 2.14 và 2.15. Một bộ phát sóng (phát ra vài chu kỳ) được phát ra và phản xạ từ vật về bộ thu có vị trí gần bộ phát. Bộ phát và bộ thu có thể được tích hợp trên cùng một cảm biến. Bộ thu cũng có thể được gắn trên vật. TOF là thời gian từ khi bắt đầu phát đến khi có tín hiệu trả về. Khoảng cách được xác định bằng công thức $d = c \cdot \text{TOF} / 2$ (khi bộ phát và bộ thu ở cùng một vị trí) và $d = c \cdot \text{TOF}$ (khi bộ thu được gắn trên vật).

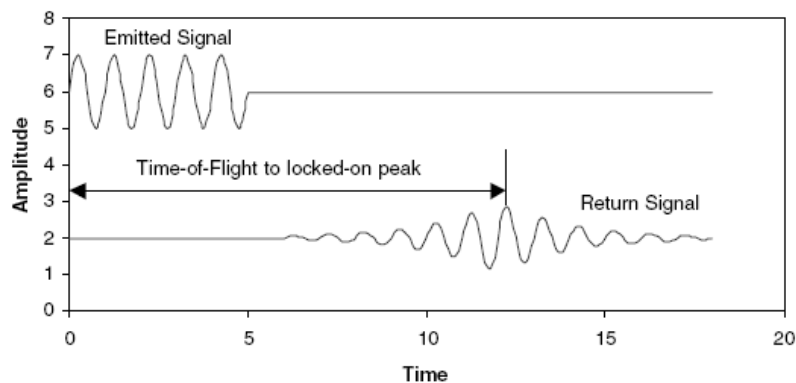


Hình 2.25 Sóng được phát và phản xạ lại từ vật



Hình 2.26 Định nghĩa thời gian truyền sóng.

Độ chính xác của phương pháp này có thể được tăng lên bằng phương pháp dò biên độ cực đại (hình 2.15)



Hình 2.27 TOF tính theo biên độ lớn nhất của tín hiệu phản xạ

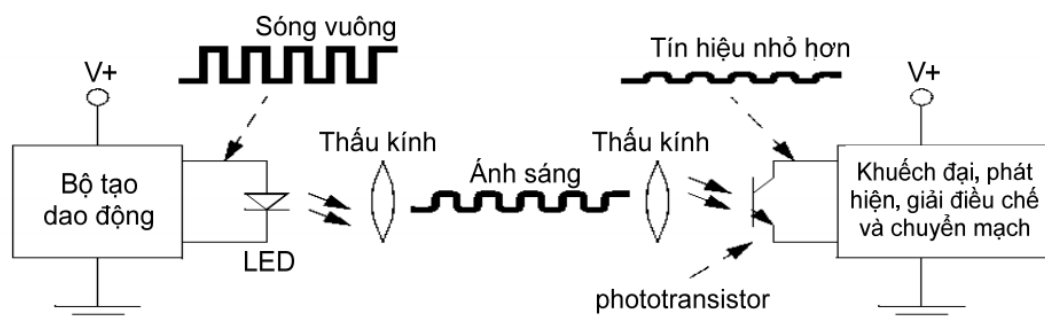
Sai số của phương pháp thời gian truyền có thể do các nguyên nhân sau:

- *Sự thay đổi tốc độ truyền sóng:* đặc biệt là với các hệ thống âm thanh ta phải chú ý tới sự thay đổi này. Vì tốc độ âm thanh chịu ảnh hưởng rõ rệt của nhiệt độ và độ ẩm.
- *Không xác định được chính xác thời gian đến của xung phản xạ.*
- *Sai số của mạch định thời sử dụng để đo thời gian truyền.*
- *Sự tương tác của sóng tới bề mặt đối tượng cần đo khoảng cách:* Khi ánh sáng, âm thanh hoặc sóng radio tác động vào đối tượng, tín hiệu phản hồi đo được chỉ thể hiện một phần nhỏ của tín hiệu ban đầu. Năng lượng còn lại phản xạ theo các hướng khác nhau và có thể được hấp thụ hoặc xuyên qua đối tượng, phụ thuộc vào đặc tính đối tượng và góc tới của chùm sóng.

Các cảm biến dưới đây được phân loại theo nguyên lý hoạt động dựa trên các hiện tượng vật lý khác nhau

2.2.4 Cảm biến quang

Các cảm biến quang yêu cầu có cả hai bộ phận là nguồn sáng (đầu phát) và đầu thu. Các đầu phát (emitter) sẽ phát ra các tia sáng trong phổ nhìn thấy và không nhìn thấy được sử dụng LED và cá diode laser. Đầu thu có cấu tạo là các diode quang (photodiode hoặc transistor quang (phototransistor)). Đầu phát và đầu thu được bố trí để đối tượng xuất hiện sẽ cắt ngang hoặc phản xạ lại tia sáng.



Hình 2.28 Nguyên lý hoạt động của cảm biến quang

Hình trên thể hiện cấu tạo của một cảm biến quang cơ bản. Chùm sáng được phản bên trái tạo ra, qua thấu kính hội tụ và truyền đi. Bên thu sẽ nhận ánh sáng qua một thấu kính hội tụ thứ hai. Ánh sáng thường được tạo ra dưới dạng xung để cảm biến có thể lọc được các nguồn ánh sáng khác trong môi trường bên ngoài. Bên phát sẽ tắt mở ở tần số đặt. Bên thu thường nhận được tín hiệu nhỏ nên cần bộ khuếch đại để tăng tín hiệu lên. Sau đó, bên thu sẽ kiểm tra xem ánh sáng có nằm cùng tần số đặt không (giải điều chế)? Nếu như không có nghĩa là đã có một vật nào chắn ngang, và ngõ ra cảm biến sẽ tác động (chuyển mạch). Có 3 loại cảm biến quang:

a. Cảm biến quang phát hiện đối tượng chắn sáng và phản xạ, phần phát và thu bố trí độc lập



Hình 2.29 Cảm biến thu phát độc lập

Đầu phát và đầu thu được tách riêng thành 2 phần bố trí thẳng hàng với nhau. Cách bố trí này có thể phát hiện được đối tượng **chắn sáng hoặc phản xạ**. Khoảng cách nhận biết có thể lên đến cả trăm mét. Tuy nhiên cách bố trí phần phát và phần thu riêng lẻ làm tăng vấn đề bảo trì và đòi hỏi sự thẳng hàng trong lắp đặt phần phát và phần thu.

b. Cảm biến quang phát hiện đối tượng chắn sáng và phản xạ, phần phát và thu bố trí chung trên một vỏ, sử dụng gương phản xạ

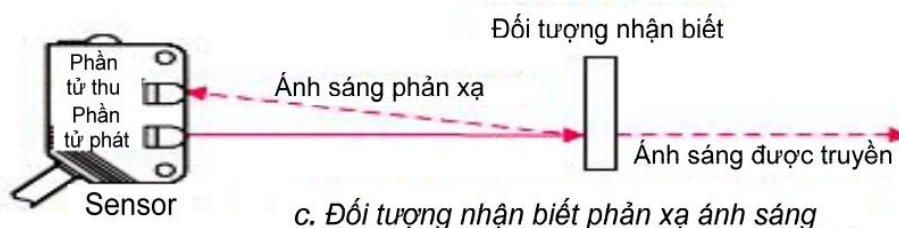


Hình 2.30 Cảm biến quang có gương phản xạ

Đầu phát và đầu thu được bố trí trên cùng một vỏ. Thông thường ánh sáng phát đi nếu không bị chắn ngang sẽ đến được gương phản xạ và hầu hết sẽ quay trở về đầu thu. Nếu có vật chắn sáng chắn ngang qua thì bên thu sẽ không nhận được ánh sáng phản xạ và ngỡ ra cảm biến được tích cực. Vấn đề rủi ro có thể gặp phải là đối tượng phản xạ ánh sáng tốt thì đầu thu cũng sẽ nhận được nhiều ánh sáng phản xạ và không phát hiện được đối tượng. Để khắc phục nhược điểm này, các cảm biến quang dạng này thường được bố trí kính phân cực ánh sáng ở đầu phát, và dùng một bộ lọc phân cực ở đầu thu.

Như vậy cảm biến loại này có thể phát hiện được đối tượng chắn sáng hoặc phản xạ, khoảng cách cũng có thể lên đến hàng trăm mét. Do bố trí đầu phát và đầu thu chung trên một vỏ nên việc bảo trì sẽ đơn giản hơn. Tuy nhiên phải sử dụng gương phản xạ, và yêu cầu lắp đặt giữa gương và cảm biến phải thẳng hàng và vuông góc.

c. Cảm biến quang phát hiện đối tượng phản xạ



Hình 2.31 Cảm biến quang phát xạ

Ánh sáng phát ra từ đầu phát, nếu không gặp vật phản xạ thì sẽ truyền thẳng đi, không quay về đầu thu. Nếu có một đối tượng phản xạ thì ánh sáng này sẽ được quay về đầu thu và ngõ ra của cảm biến sẽ tác động.

Loại cảm biến này chỉ sử dụng được với các đối tượng phản xạ, có kích thước tương đối lớn và khoảng cách chỉ trong vòng vài mét.

2.2.5 Cảm biến điện dung

Cảm biến điện dung được hoạt động dựa trên nguyên tắc hằng số điện môi của không gian xung quang cảm biến sẽ thay đổi khi các đối tượng đượ mang gần đến cảm biến.

Công thức tính điện dung giữa hai bản tụ điện:

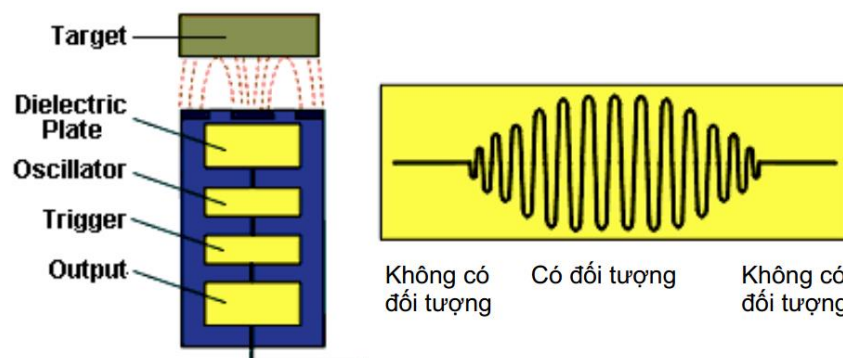
$$C = \frac{\epsilon.A}{d}$$

Trong đó:

- C là điện dung (Fara)
- ϵ : Hằng số điện môi
- A: Diện tích 2 bản tụ điện
- d: Khoảng cách 2 bản tụ điện

Như vậy khi hằng số điện dung thay đổi thì điện dung sẽ thay đổi và sự thay đổi này sẽ được phát hiện để tác động ngõ ra.

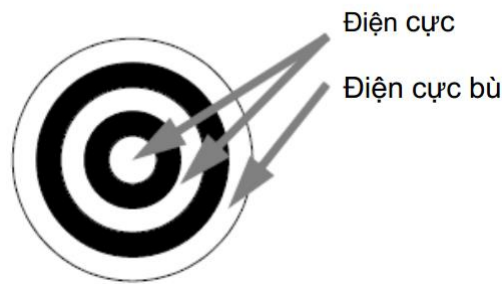
Bề mặt của cảm biến điện dung được hình thành bởi hai điện cực kim loại đồng tâm của một tụ điện. Khi một đối tượng đến gần bề mặt nhận biết, nó sẽ đi vào vùng điện trường của các điện cực và thay đổi điện dung trong mạch dao động. Kết quả biên độ của mạch dao động thay đổi. Một mạch trigger đọc biên độ của dao động. Và khi biên độ đạt đến một mức xác định thì sẽ tác động ngõ ra của cảm biến. Khi đối tượng rời đi thì biên độ của bộ dao động giảm, ngõ ra cảm biến sẽ thôi không tác động nữa.



Hình 2.32 Cảm biến điện dung

Các cảm biến điện dung làm việc tốt với các chất cách điện (có hằng số điện môi cao). Hằng số điện môi càng lớn thì khoảng cách hoạt động càng cao. Do đó, khi hiệu chỉnh đúng, cảm biến điện dung có thể được sử dụng để phát hiện chất lỏng trong bình chứa một cách dễ dàng. Tuy nhiên, cảm biến này cũng làm việc tốt đối với đối tượng kim loại.

Trong thực tế, cảm biến điện dung thường được chế tạo với các vòng (không phải bản cực).

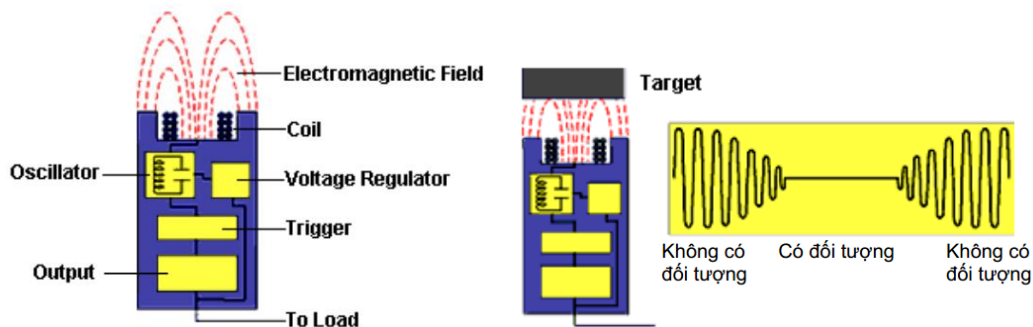


Hình 2.33 Cấu trúc cảm biến điện dung

Trong hình, hai vòng kim loại phía trong là các điện cực của tụ điện. Vòng thứ ba ở ngoài được thêm vào để bù sự thay đổi từ môi trường. Nếu không có vòng này thì cảm biến sẽ rất nhạy với bụi bẩn, dầu, nước, độ ẩm và các chất khác dính trên cảm biến.

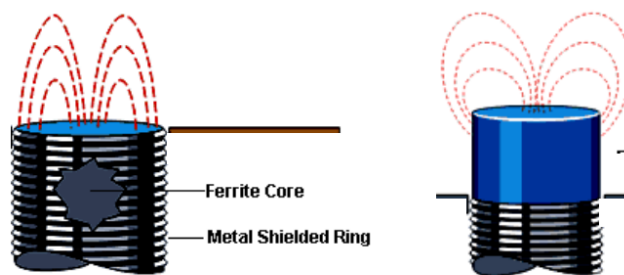
2.2.6 Cảm biến điện cảm

Cảm biến điện cảm sử dụng dòng điện cảm ứng để **phát hiện đối tượng là kim loại**. Một cuộn dây được sử dụng để tạo ra điện từ trường tần số cao. Nếu có một đối tượng là kim loại đến gần làm thay đổi từ trường thì sẽ có một dòng điện xoáy chảy vào đối tượng. Điều này làm tăng tải trong cảm biến, làm giảm biên độ dao động của điện từ trường. Mạch Trigger giám sát biên độ dao động. Khi biên độ giảm đến mức định trước thì nó sẽ tác động ngõ ra của cảm biến. Khi đối tượng di chuyển ra xa thì biên độ dao động tăng lên. Đến một giá trị định trước thì ngõ ra của cảm biến sẽ thôi không tác động nữa.



Hình 2.34 Cảm biến điện cảm

Cảm biến điện cảm được sử dụng để phát hiện các đối tượng là kim loại, khoảng cách nhận biết vào khoảng vài cm. Loại cảm biến không có vỏ bọc kim loại ở đầu cuộn dây thì khoảng cách nhận biết xa hơn, nhưng hướng của đối tượng có thể là bất kỳ. Bằng cách thêm vỏ bọc kim loại ở đầu cuộn dây thì khoảng cách nhận biết sẽ giảm đi nhưng hướng của đối tượng được cải thiện hơn.

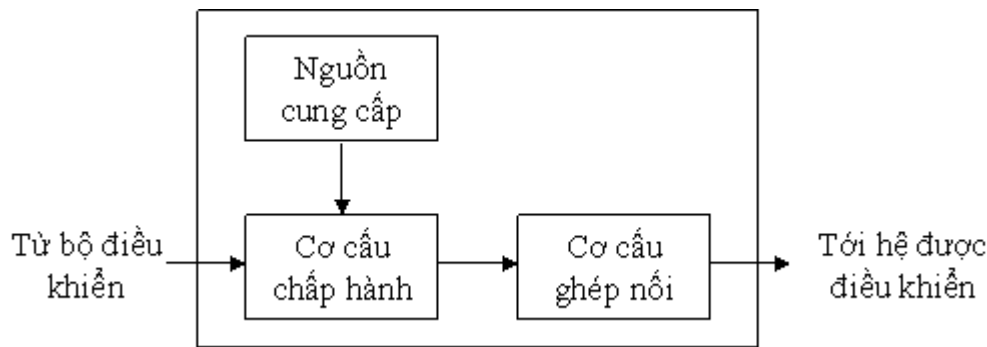


Cảm biến bọc và không bọc vỏ kim loại

Hình 2.23 Cảm biến có bọc đầu cảm ứng và không bọc đầu cảm ứng bằng kim loại

2.3 Cơ cấu chấp hành.

Về cơ bản, các cơ cấu chấp hành là phần phía sau một hệ cơ điện tử, nhận lệnh điều khiển (hầu hết là dạng tín hiệu điện) và gây ra một sự thay đổi trong hệ vật lý bằng cách tạo lực, chuyển động, nhiệt, dòng chảy... Thông thường các cơ cấu chấp hành được sử dụng kết hợp với nguồn nuôi và một cơ cấu ghép nối như trên *hình 3.2*.



Hình 2.35 Thiết bị chấp hành thông thường.

Phân loại:

Cơ cấu chấp hành có thể phân loại dựa trên dạng năng lượng. Về cơ bản có các dạng: điện, điện cơ, điện từ, thủy lực và khí nén. Cơ cấu chấp hành cũng có thể phân loại thành dạng nhị phân và liên tục dựa trên số lượng đầu ra trạng thái ổn định. Một rơle với hai trạng thái ổn định là một ví dụ điển hình của cơ cấu chấp hành dạng nhị phân. Tương tự, một động cơ bước là một ví dụ điển hình cho cơ cấu chấp hành dạng liên tục.

2.3.1 Các động cơ điện.

Động cơ DC.

Trên hình 2.25 khi cho điện áp một chiều U vào hai chổi điện A và B, trong dây quấn phản ứng có dòng điện chịu lực tác dụng tương hỗ lên nhau tạo nên momen tác dụng lên rôto, làm rôto quay. Chiều lực tác dụng được xác định theo qui tắc bàn tay trái (hình 2.25a).

Khi phản ứng quay được $1/2$ vòng, vị trí thanh dẫn ab, cd đổi chỗ nhau (hình 2.25b), nhờ có phiến góp đổi chiều dòng điện, nên dòng điện một chiều biến đổi thành dòng điện xoay chiều đưa vào dây quấn phản ứng, giữ cho chiều lực tác dụng không đổi, do đó lực tác dụng lên rôto cũng theo một chiều nhất định, đảm bảo động cơ có chiều quay không đổi.

ĐIỀU CHỈNH TỐC ĐỘ:

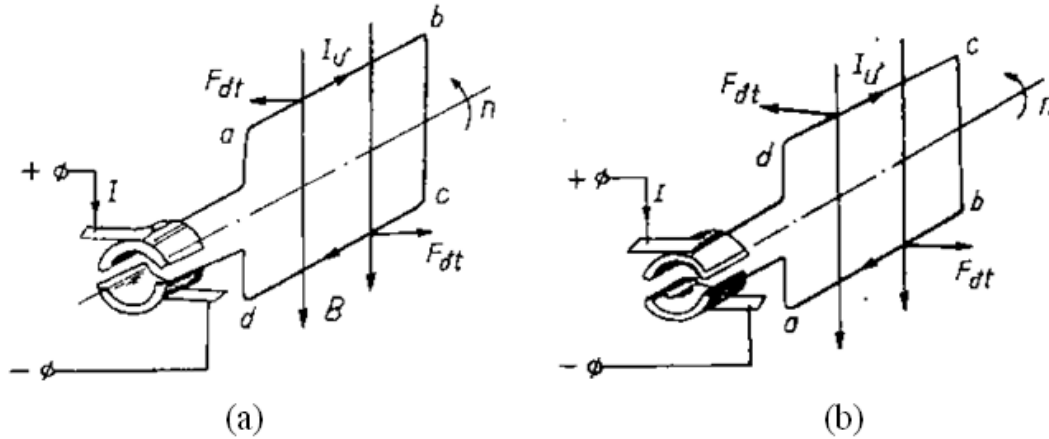
Ứng với mỗi tải khác nhau ta cần có tốc độ khác nhau. Vì vậy để phù hợp với tải, cần phải điều chỉnh tốc độ động cơ lúc có tải.

Ta có phương trình điện áp ở mạch phản ứng là:

$$U = E_u + R_u I_u$$

$$E_u = U - R_u I_u$$

Thay trị số $E_u = K_E \theta n$ ta có phương trình tốc độ là



Hình 2.36 Nguyên lý làm việc của động cơ một chiều

$$n = \frac{U - R_u I_u}{K_E \theta}$$

Nhìn vào phương trình trên, ta thấy rằng muốn điều chỉnh tốc độ ta có các phương pháp sau:

a. Mắc điện trở điều chỉnh vào mạch phản ứng.

Khi mắc thêm điện trở phụ vào mạch phản ứng, điện trở R_u tăng lên đặc tính cơ dốc xuống, tốc độ động cơ giảm dần. Do dòng điện phản ứng lớn nên tổn hao công suất trên điện trở điều chỉnh lớn. Phương pháp này chỉ sử dụng ở động cơ công suất nhỏ.

b. Thay đổi điện áp U.

Dùng nguồn điện một chiều để điều chỉnh điện áp cung cấp cho động cơ. Phương pháp này được sử dụng nhiều.

c. Thay đổi từ thông.

Thay đổi từ thông bằng cách thay đổi dòng điện kích từ.

Khi điều chỉnh tốc độ, ta kết hợp với các phương pháp trên. Ví dụ phương pháp thay đổi từ thông, kết hợp với phương pháp thay đổi điện áp thì phạm vi điều chỉnh rất rộng, đây là ưu điểm lớn của động cơ điện một chiều.

Động cơ AC.

a. Động cơ không đồng bộ.

Nguyên lý làm việc: khi đặt điện áp xoay chiều ba pha có tần số f_1 vào dây quấn stato, trong dây quấn stato sẽ có hệ thống dòng ba pha chạy qua, dòng điện này sẽ tạo ra từ trường quay pđôi cực, quay với tốc độ $n_1 = \frac{60f}{p}$. Từ trường quay cắt các thanh dẫn của dây quấn rôto và cảm ứng các sức điện động. Vì dây quấn rôto nóng ngắn mạch, nên sức điện động cảm ứng sẽ sinh ra dòng trong các thanh dẫn rôto. Lực tác dụng tương hỗ giữa từ trường quay của máy với thanh dẫn mang dòng điện rôto, kéo rôto quay theo chiều với từ trường quay với tốc độ n .

Để minh họa, ta xét từ trường quay B của stato đang quay theo chiều kim đồng hồ với tốc độ n_1 (hình 2.26).

Khi xác định chiều sức điện động cảm ứng theo quy tắc bàn tay phải, ta căn cứ vào chiều chuyển động tương đối của thanh dẫn với từ trường. Nếu coi từ trường đứng yên, thì chiều chuyển động tương đối của thanh dẫn ngược với chiều n_1 , từ đó áp dụng quy tắc bàn tay phải xác định được chiều sức điện động như hình vẽ.

Chiều lực điện từ xác định theo quy tắc bàn tay trái, trùng với chiều quay n_1 .

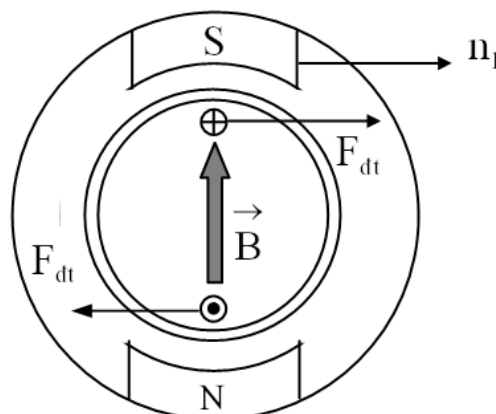
Độ chênh lệch giữa tốc độ từ trường quay và tốc độ của máy gọi là tốc độ trượt n_2 :

$$n_2 = n_1 - n$$

Hệ số trượt của tốc độ là

$$S = \frac{\vec{n}_2}{\vec{n}_1} = \frac{n_1 - n}{n_1}$$

Khi rôto đứng yên ($n = 0$), hệ số trượt $s = 1$. Khi rôto quay định mức $s = 0,02 \div 0,06$. Tốc độ động cơ là:



Hình 2.37 Quá trình tạo momen quay của động cơ không đồng bộ

b. Động cơ đồng bộ.

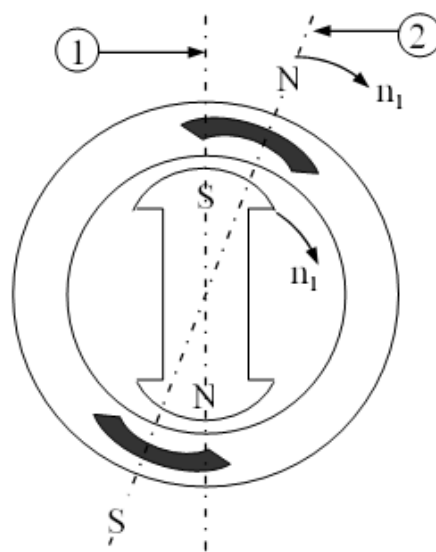
Nguyên lý làm việc:

Khi cho dòng điện ba pha vào dây quấn stato, dòng điện ba pha ở dây quấn stato sẽ sinh ra từ trường quay với tốc độ:

$$n_1 = \frac{60f}{p}$$

Nếu rôto đang đứng yên (hình 2.27), cực Nam S của rôto bị cực Bắc N stato kéo và nó có xu hướng quay theo chiều kim đồng hồ. Nhưng do quán tính và cực Bắc stato quét qua nó quá nhanh, trong khi nó chưa quay tới thì sau nữa chu kỳ nó đã đối diện với cực Nam stato và bị đẩy lùi, nghĩa là rôto có xu hướng quay theo chiều ngược lại. Kết quả là moment (mở máy) trung bình bằng không và rôto không quay được.

Tuy nhiên nếu chúng ta quay trước rôto với tốc độ đồng bộ các cực từ rôto bị “khóa chặt” vào cực từ stato trái dấu. Khi không tải, từ trường stato và rôto cùng quay với tốc độ đồng bộ n_1 và trục của chúng trùng nhau ($\theta = 0$). Lúc có tải trục từ trường rôto đi chậm sau trục từ trường stato một góc θ , tải càng nặng góc θ càng lớn, nhưng cả hai vẫn cùng quay với tốc độ đồng bộ n_1 .



Hình 2.38 Sự tạo ra momen trong động cơ đồng bộ.

Động cơ bước.

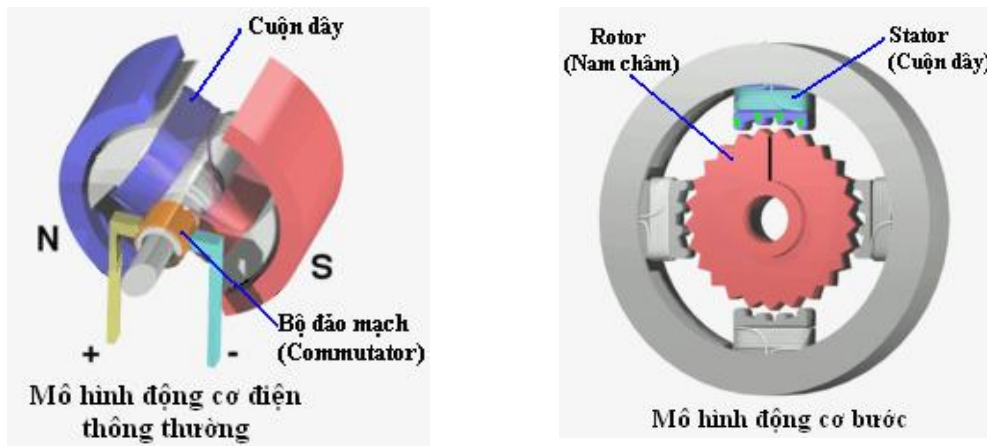
a. Giới thiệu về động cơ bước.

Động cơ bước cơ bản là một động cơ điện không có bộ phận đảo mạch (chổi than). Thông thường động cơ bước có nhiều hơn 2 cuộn dây đóng vai trò stator (phần tĩnh). Rotor (phần động) vẫn là một nam châm vĩnh cửu hoặc là những khối có răng (khía) làm bằng vật liệu nhẹ có từ tính (trường hợp động cơ bước biến từ trở). Sự đảo mạch cho dòng điện trong cuộn dây được thực hiện bằng mạch điện điều khiển bên ngoài. Thông thường, mạch điều khiển này còn có thêm chức năng điều khiển động cơ quay tới, lui hoặc giữ nguyên vị trí xác định.

Động cơ bước thuộc loại động cơ đồng bộ, chúng không thể quay tự do, liên tục mà hoạt động theo nguyên lý quay từng “bước”, mỗi bước động cơ sẽ quay được một góc (ví dụ 1,8°/bước). Một vòng quay của nó gồm nhiều bước nối tiếp nhau. Khi kết hợp với mạch điều khiển bên ngoài, chúng có thể quay đến và dừng lại tại một vị trí xác định một cách chính xác.

b. Các loại động cơ bước.

Dựa vào cấu tạo, ta có ba loại động cơ bước chính là: Động cơ bước nam châm vĩnh cửu, động cơ bước biến từ trở và động cơ bước hỗn hợp (động cơ lai). Ngoài ra còn có cách phân loại dựa vào mạch điều khiển như động cơ bước đơn cực, lưỡng cực.



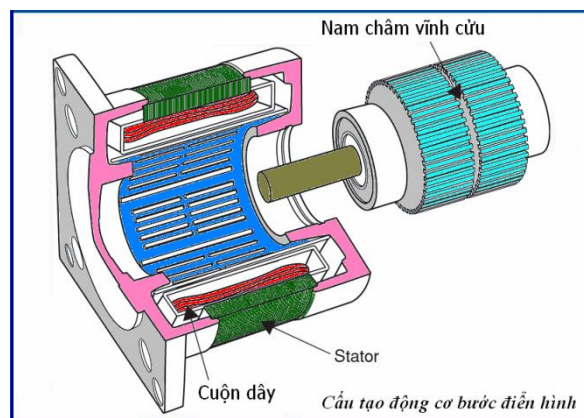
Hình 2.39 Phân biệt động cơ bước và động cơ điện thông thường.

Động cơ bước nam châm vĩnh cửu.

Động cơ bước nam châm vĩnh cửu có rotor là một nam châm vĩnh cửu, không chia thành nhiều răng, cũng có nhiều cuộn dây như loại biến từ trở. Có hai loại thông dụng là động cơ bước đơn cực (còn gọi là độc cực – unipolar) và lưỡng cực (bipolar).

Động cơ bước biến từ trở:

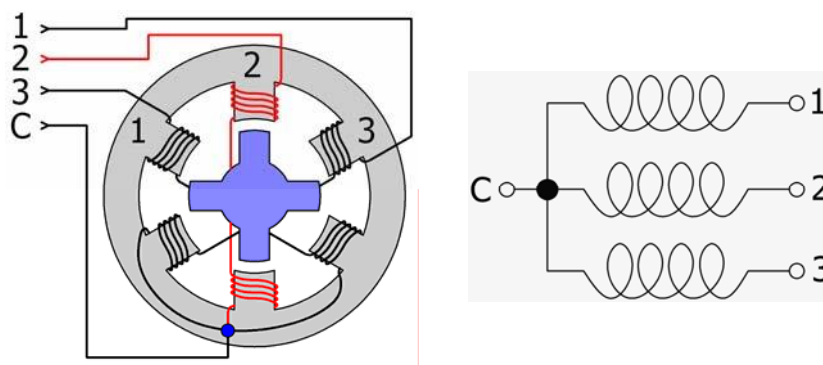
Động cơ bước biến từ trở (hay còn gọi là động cơ bước phản kháng) không dùng nam châm vĩnh cửu để làm rotor mà dùng một khối có răng làm bằng vật liệu nhẹ có từ tính. Phần stator là 3 đến 5 cuộn dây được nối chung với nhau.



Hình 2.40 Cấu tạo bên trong động cơ bước.



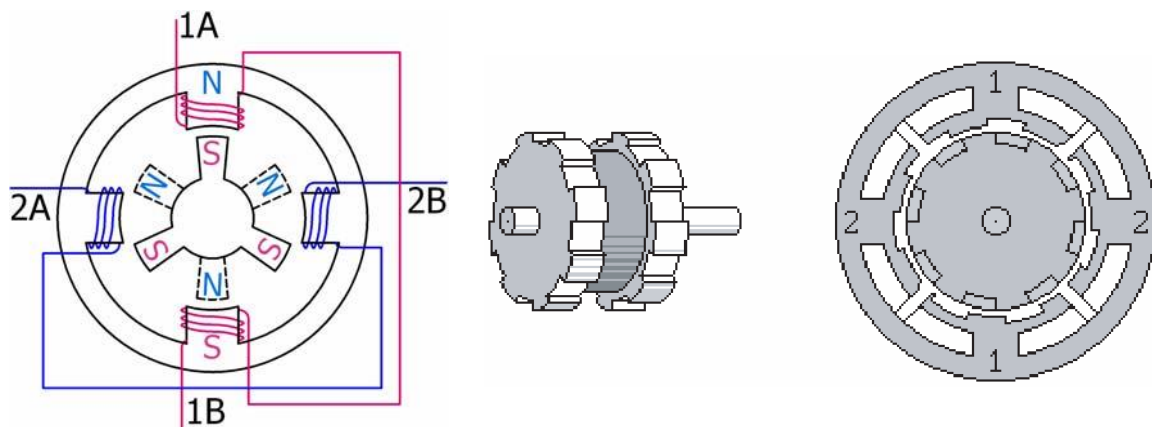
Hình 2.41 Hình dáng bên ngoài của động cơ bước.



Hình 2.42 Động cơ biến từ trở.

Động cơ bước hỗn hợp.

Động cơ bước hỗn hợp (hay còn gọi là động cơ bước cảm ứng) kết hợp ưu điểm của hai loại động cơ trên, nghĩa là chúng có rotor là nam châm vĩnh cửu, có nhiều răng, stator là những cuộn dây. Động cơ hỗn hợp thông dụng là loại có 200 răng và có bước $1,8^0$.



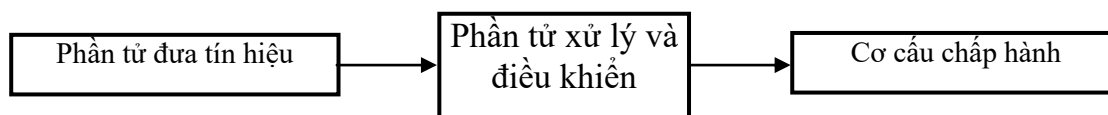
Hình 2.43 Động cơ bước hỗn hợp.

Như ta thấy trong hình, rotor của động cơ bước hỗn hợp gồm hai đĩa bố trí so le nhau. Mỗi đĩa có nhiều răng ứng với các cực.

2.5 Hệ thống điều khiển khí nén.

Điều khiển là quá trình của một hệ thống, trong đó dưới tác dụng của một hay nhiều đại lượng vào, các đại lượng ra thay đổi theo một quy luật nhất định của hệ thống đó.

Một hệ thống điều khiển bao gồm:



- | | | |
|------------------------|-----------------------|--------------------|
| - Công tắc, nút bấm. | - Van đảo chiều. | - Xilanh. |
| - Công tắc hành trình. | - Van chặn. | - Động cơ khí nén. |
| - Cảm biến. | - Van tiết lưu. | |
| | - Van áp suất. | |
| | - Phần tử khuếch đại. | |

Trong phần này ta chỉ xét đến 2 phần: Phần tử xử lý và điều khiển và phần cơ cấu chấp hành.

Phần tử xử lý và điều khiển.

Van đảo chiều.

Van đảo chiều có nhiệm vụ điều khiển dòng năng lượng bằng cách đóng mở hay thay đổi vị trí các cửa van để thay đổi hướng của dòng khí nén.

a. Ký hiệu của van đảo chiều.

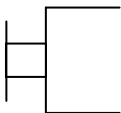
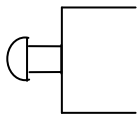
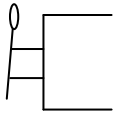
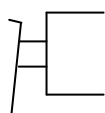
TÊN THIẾT BỊ	KI HIỆU
Van đảo chiều 2/2	
Van đảo chiều 4/2	
Van đảo chiều 5/2	

b. Tín hiệu tác động:

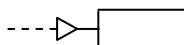
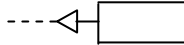
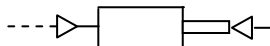
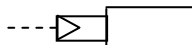
Tín hiệu tác động vào van đảo chiều có 4 loại là: tác động bằng tay, tác động bằng cơ học, tác động bằng khí nén và tác động bằng nam châm điện.

Tín hiệu tác động từ 2 phía (đối với van đảo chiều không có vị trí ‘không’) hay chỉ từ 1 phía (đối với van đảo chiều có vị trí ‘không’).

Tác động bằng tay:


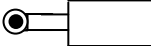
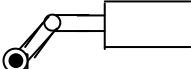

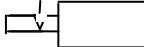
TÊN THIẾT BỊ	KÍ HIỆU
Kí hiệu nút nhấn tổng quát	
Nút bấm	
Tay gạt	
Bàn đạp	

Tác động bằng khí nén:

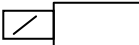
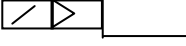
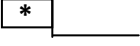
TÊN THIẾT BỊ	KÍ HIỆU
Trực tiếp bằng dòng khí nén vào	
Trực tiếp bằng dòng khí nén ra	
Trực tiếp bằng dòng khí nén vào với đường kính 2 đầu nòng van khác nhau	
Gián tiếp bằng dòng khí nén vào qua van	

phụ trợ	
---------	--

Tác động bằng cơ.

TÊN THIẾT BỊ	KÍ HIỆU
Đầu dò	
Cữ chặn bằng con lăn, tác động 2 chiều	
Cữ chặn bằng con lăn, tác động 1 chiều	
Lò xo	
Nút nhấn có rãnh định vị	

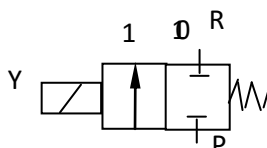
Tác động bằng nam châm điện

TÊN THIẾT BỊ	KÍ HIỆU
Trực tiếp	
Bảng nam châm điện và van phụ trợ	
Tác động theo cách hướng dẫn cụ thể	

c. Van đảo chiều có vị trí ‘không’.

Van đảo chiều có vị trí ‘không’ là loại van tác động bằng cơ – lò xo lên nòng van và ký hiệu lò xo nằm ngay vị trí bên cạnh ô vuông phía bên phải của ký hiệu van. Tác động lên phía đối diện nòng van là tín hiệu tác động bằng cơ, khí nén hay bằng điện. Khi chưa có tín hiệu tác động, vị trí của các cửa nối được biểu diễn trong ô vuông phía bên phải đối với van đảo chiều 2 vị trí. Còn đối với van đảo chiều 3 vị trí thì vị trí ‘không’ nằm ở giữa.

Ví dụ: Van đảo chiều 2/2 tác động bằng nam châm điện.



Van có 2 cửa P và R, 2 vị trí 0 và 1. Tại vị trí 0, cửa P và R bị chặn. Khi cuộn Y có điện, từ vị trí 0 van chuyển sang vị trí 1, cửa P nối với cửa R. Khi cuộn Y mất điện, do tác động của lò xo phía đối diện, van sẽ quay trở về vị trí ban đầu.

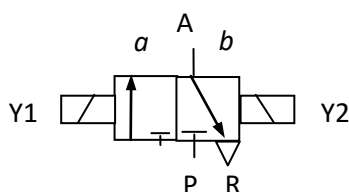
d. Van đảo chiều không có vị trí ‘không’.

Khi không có tín hiệu tác động lên đầu nòng van nữa, thì vị trí của van vẫn được giữ nguyên do tín hiệu tác động từ phía nòng van đối diện. Vị trí tác động ký hiệu a, b, c, ...

Tín hiệu tác động có thể là:

- Tác động bằng tay hay bàn đạp.
- Tác động bằng dòng khí nén điều khiển vào hay ra từ 2 phía nòng van.
- tác động trực tiếp bằng điện từ hay gián tiếp bằng dòng khí nén đi qua van phụ trợ.

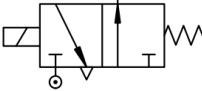
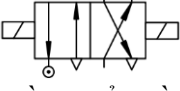


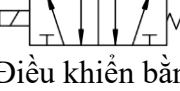
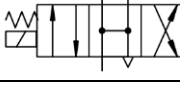
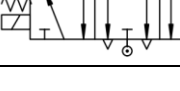
Ví dụ: Van đảo chiều 3/2 tác động bằng nam châm điện.



Khi cuộn Y1 có điện thì cửa P nối với cửa A, cửa R bị chặn. Khi cuộn Y2 có điện thì cửa A nối với cửa R còn cửa P bị chặn.

Bảng tóm tắt các loại van điều khiển bằng Solenoid thường dùng

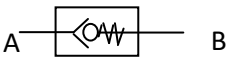
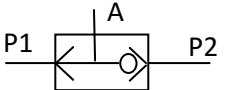
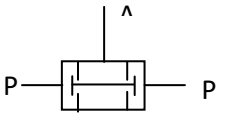
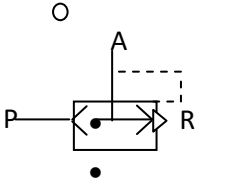
Loại van	Ký hiệu van điều khiển bằng Solenoid	Ứng dụng
2 cửa, 2 vị trí	<p>Thường đóng</p>	Điều khiển mở dòng chảy
	<p>Thường mở</p>	Điều khiển ngắt dòng chảy
3 cửa, 2 vị trí	<p>Thường đóng</p>	Điều khiển cho các xy lanh tác động đơn

	 <p>Thường mở</p>	Điều khiển cho các xy lanh tác động đơn
4 cửa, 2 vị trí	 <p>Điều khiển bằng 2 Solenoid</p>	Điều khiển cho các xy lanh tác động kép
	 <p>Điều khiển bằng 1 Solenoid, trả về bằng lò xo</p>	Điều khiển cho các xy lanh tác động kép
5 cửa, 2 vị trí	 <p>Điều khiển bằng 2 Solenoid</p>	Điều khiển cho các xy lanh tác động kép
	 <p>Điều khiển bằng 1 Solenoid, trả về bằng lò xo</p>	Điều khiển cho các xy lanh tác động kép
4 cửa, 3 vị trí		Điều khiển vị trí cho các xy lanh tác động kép
5 cửa, 3 vị trí		Điều khiển vị trí cho các xy lanh tác động kép

Các cuộn Solenoid điều khiển các van khí nén thường được nối trực tiếp vào các ngõ ra của PLC. Hầu hết các Solenoid công nghiệp sử dụng nguồn cấp 24VDC và dòng định mức vài trăm mA.

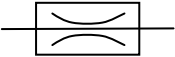


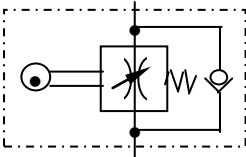
Van chặn: là loại van chỉ cho dòng khí nén đi qua một chiều, chiều còn lại bị chặn. Van chặn gồm có các loại sau:

- Van 1 chiều.
- Van Logic OR.
- Van Logic AND.
- Van xả khí nhanh.

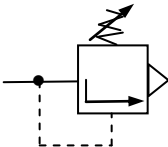
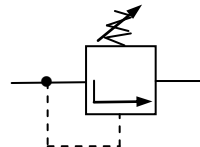
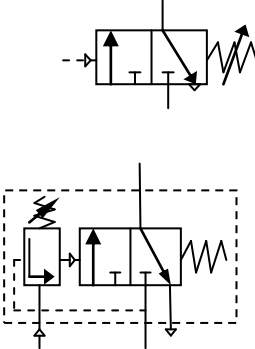
TÊN THIẾT BỊ	KÍ HIỆU
Van một chiều: có tác dụng chỉ cho dòng khí nén đi qua một chiều (từ A qua B), chiều ngược lại bị chặn.	
Van logic OR: khi có dòng khí nén vào từ P1 thì cửa P2 bị chặn và cửa P1 nối với cửa A. Ngược lại khi dòng khí nén vào P2 thì cửa P1 bị chặn, cửa P2 nối với cửa A.	
Van logic AND: khi có dòng khí nén vào P1 thì P1 bị chặn, và ngược lại khi có dòng khí nén vào P2 thì P2 bị chặn. Chỉ khi nào cả P1 và P2 có dòng khí nén vào thì mới có khí nén qua cửa A.	
Van xả khí nhanh: khi dòng khí nén vào cửa P, chặn cửa R, cửa P nối với cửa A. Khi dòng khí nén vào từ A, cửa P bị chặn, cửa A nối với cửa R, khí được xả nhanh ra ngoài.	

Van tiết lưu:

Van tiết lưu có nhiệm vụ thay đổi lưu lượng dòng khí nén, có nghĩa là thay đổi vận tốc của cơ cấu chấp hành.

TÊN THIẾT BỊ	KÍ HIỆU
<p>Van tiết lưu có tiết diện không đổi:</p> <p>Khe hở của van có tiết diện không thay đổi, do đó lưu lượng dòng chảy không thay đổi.</p>	
<p>Van tiết lưu có tiết diện thay đổi:</p> <p>Lưu lượng dòng chảy qua van thay đổi được nhờ vào một vít điều chỉnh làm thay đổi tiết diện của khe hở.</p>	
<p>Van tiết lưu một chiều điều chỉnh bằng tay:</p> <p>Tiết diện A_z thay đổi bằng cách điều chỉnh vít bằng tay. Khi dòng khí nén từ A sang B, lò xo đẩy màng chắn và dòng khí nén chỉ qua tiết diện A_z. Khi dòng khí nén từ B sang A, màng chắn bị đẩy lên và dòng khí đi qua khoảng hở giữa màng chắn và mặt tựa màng chắn, lưu lượng không được điều chỉnh.</p>	
<p>Van tiết lưu một chiều điều chỉnh bằng cỡ chặn:</p> <p>Nguyên lý hoạt động tương tự như van tiết lưu một chiều điều chỉnh bằng tay. Dòng khí nén chỉ có thể đi một chiều từ A sang B, tùy vào vị trí của cỡ chặn mà tiết diện của khe hở của van thay đổi, làm cho lưu lượng dòng chảy thay đổi.</p>	

Van áp suất.

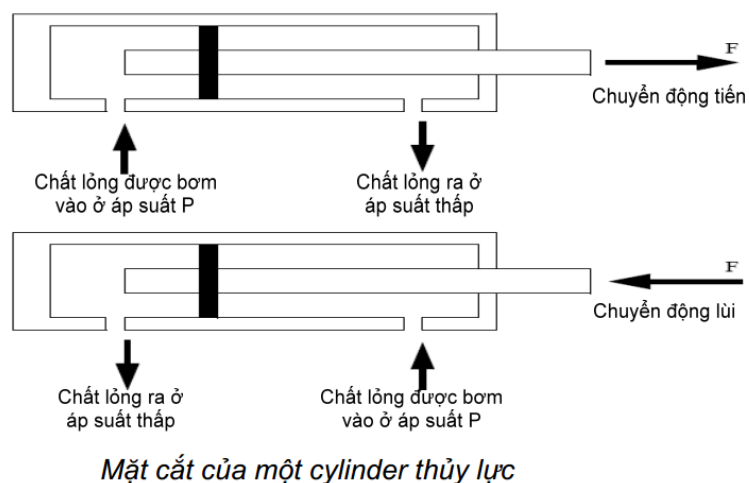
TÊN THIẾT BỊ	KÍ HIỆU
<p>Van an toàn.</p> <p>Bình thường khi áp suất nhỏ hơn hoặc bằng áp suất cho phép, cửa R bị chặn, nhưng khi áp suất lớn hơn áp suất cho phép, cửa R mở ra, khí nén từ cửa P theo cửa R thoát ra ngoài.</p>	
<p>Van tràn: nguyên tắc hoạt động tương tự như van an toàn, nhưng chỉ khác ở chỗ là khi áp suất bằng hoặc lớn hơn áp suất cho phép thì cửa P nối với cửa A, nối với hệ thống điều khiển.</p>	
<p>Van áp suất điều chỉnh từ xa: khi có tín hiệu áp suất tác động, thì cửa P nối với cửa A. Tín hiệu tác động có thể trực tiếp lên van đảo chiều, hay tín hiệu gián tiếp qua van tràn.</p>	

Cơ cấu chấp hành:

Cơ cấu chấp hành có nhiệm vụ biến đổi năng lượng khí nén hoặc thủy lực thành năng lượng cơ học. Cơ cấu chấp hành có thể thực hiện chuyển động thẳng (xi lanh) hoặc chuyển động quay (động cơ khí nén).

Xi lanh.

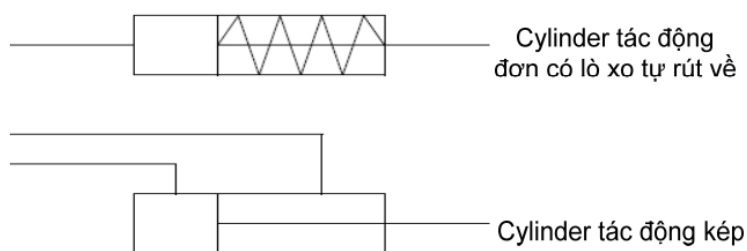
Xi lanh sử dụng áp lực khí nén hoặc chất lỏng để tạo lực/chuyển động tịnh tiến (xy lanh thẳng) hoặc lực/chuyển động quay (xy lanh quay).



Hình 2.44 Mặt cắt 1 xy lanh thủy lực.

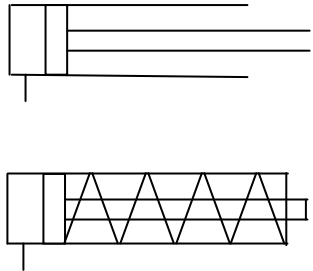
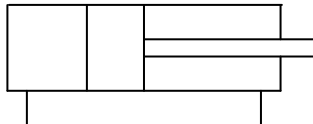
Trong hình 2.33, chất lỏng với áp suất P được bơm vào một phía của xy lanh, làm xy lanh di chuyển về phía còn lại, chất lỏng ở phía này được thoát ra tự do. Lực tác dụng lên xy lanh phụ thuộc vào áp suất và diện tích bề mặt của piston.

Xy lanh tác động đơn chỉ yêu cầu cung cấp lực khi duỗi ra và sử dụng lò xo để co về. Còn xy lanh tác động kép thì cần cung cấp lực ở cả 2 quá trình co và duỗi.



Hình 2.45 Xy lanh tác động đơn và kép.

Xy lanh từ thường được sử dụng trong điều khiển khí nén. Một mảnh nam châm vĩnh cửu được gắn trên đầu piston. Khi nó di chuyển tới các vị trí xác định thì sẽ được các công tắc từ (reed switch) phát hiện.

TÊN THIẾT BỊ	KÍ HIỆU
Xilanh tác dụng đơn (xilanh tác dụng một chiều). Áp lực khí nén chỉ tác dụng vào một phía của xilanh, phía còn lại là do ngoại lực hay lò xo tác dụng.	
Xilanh tác dụng 2 chiều (xilanh tác dụng kép). Áp suất khí nén được dẫn vào 2 phía của xilanh, do yêu cầu điều khiển mà xilanh sẽ đi vào hay đi ra tùy thuộc vào áp lực khí nén vào phía nào.	

Động cơ khí nén:

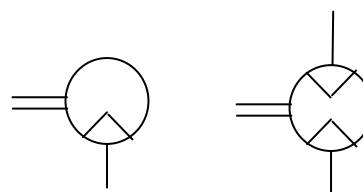
Động cơ khí nén có nhiệm vụ biến đổi năng lượng của khí nén thành năng lượng cơ học (chuyển động quay).

Động cơ khí nén có những ưu điểm sau:

- Điều chỉnh được momen quay và số vòng quay.
- Số vòng quay cao và điều chỉnh vô cấp
- Không hư hỏng khi quá tải
- Giá thành bảo dưỡng thấp

Nhược điểm.

- Giá thành năng lượng cao.
- Số vòng quay thay đổi theo tải trọng.
- Gây tiếng ồn lớn khi xả khí.



- Động cơ quay một chiều
- Động cơ quay hai chiều.

Hình 2.46 Ký hiệu động cơ khí nén

CHƯƠNG 3: BỘ ĐIỀU KHIỂN - LẬP TRÌNH PLC

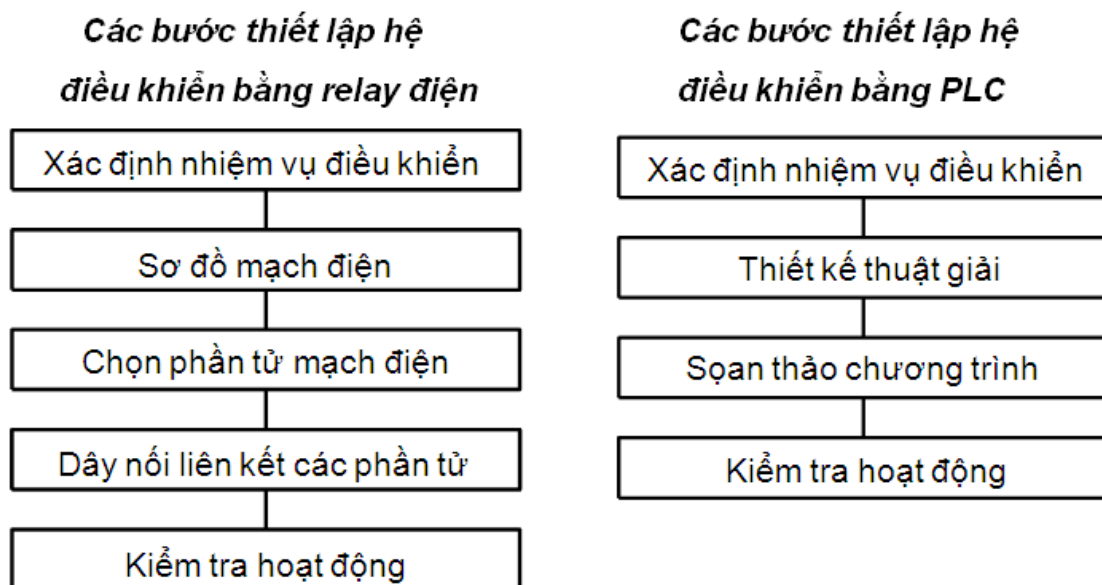
3.1 Giới thiệu

Các thành phần của kỹ thuật điều khiển điện và điện tử ngày càng đóng một vai trò vô cùng to lớn trong lĩnh vực tự động hóa ngày càng cao. Trong những năm gần đây, bên cạnh việc điều khiển bằng Relay và khởi động từ thì việc điều khiển có thể lập trình được càng phát triển với hệ thống đóng mạch điện tử và thực hiện lập trình bằng máy tính. Trong nhiều lĩnh vực, các loại điều khiển cũ đã được thay đổi bởi các bộ điều khiển có thể lập trình được, có thể gọi là các bộ điều khiển logic khả trình, viết tắt trong tiếng Anh là PLC (Programmable Logic Controller).

Sự khác biệt cơ bản giữa điều khiển logic khả trình (thay đổi được qui trình hoạt động) và điều khiển theo kết nối cứng (không thay đổi được qui trình hoạt động) là: Sự kết nối dây không còn nữa, thay vào đó là chương trình.

Có thể lập trình cho PLC nhờ vào các ngôn ngữ lập trình đơn giản. Đặc biệt đối với người sử dụng không cần nhờ vào các ngôn ngữ lập trình khó khăn, cũng có thể lập trình PLC được nhờ vào các liên kết logic cơ bản.

Như vậy thiết bị PLC làm nhiệm vụ thay thế phần mạch điện điều khiển trong khâu xử lý số liệu. Nhiệm vụ của sơ đồ mạch điều khiển sẽ được xác định bởi một số hữu hạn các bước thực hiện xác định gọi là chương trình. Chương trình này mô tả các bước thực hiện gọi một tiến trình điều khiển, tiến trình này được lưu vào bộ nhớ nên được gọi là điều khiển theo lập trình nhớ hay điều khiển khả trình. Trên cơ sở khác nhau ở khâu xử lý số liệu có thể biểu diễn hai hệ điều khiển như sau:



Khi thay đổi nhiệm vụ điều khiển thì người ta thay đổi mạch điều khiển: Lắp lại mạch, thay đổi các phần tử mới ở hệ điều khiển bằng relay điện. Trong khi đó khi thay đổi nhiệm vụ điều khiển ở hệ điều khiển logic khả trình (PLC) thì người ta chỉ thay đổi chương trình soạn thảo.

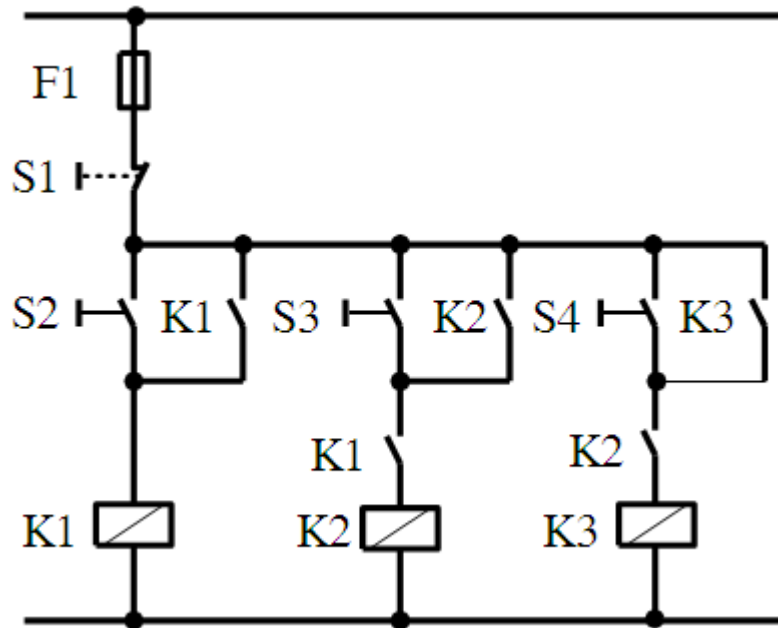
3.2 Sự khác nhau giữa hệ điều khiển bằng relay và hệ điều khiển bằng PLC

Sự khác nhau giữa hệ điều khiển bằng relay và hệ điều khiển bằng PLC có thể minh họa một cách cụ thể như sau:

Điều khiển hệ thống của 3 máy bơm qua 3 khởi động từ K1, K2, K3. Trình tự điều

khởi động như sau: Các khởi động từ chỉ được phép thực hiện tuần tự, nghĩa là K1 đóng trước, tiếp theo K2 đóng và cuối cùng K3 mới đóng.

Để thực hiện nhiệm vụ theo yêu cầu trên mạch điều khiển được thiết kế như sau:



Hình 3.1: Mạch điều khiển trình tự 3 máy bơm

Khởi động từ K2 sẽ đóng khi công tắc S3 đóng với điều kiện là khởi động từ K1 đã đóng trước đó. Phương thức điều khiển như vậy được gọi là điều khiển trình tự. Tiến trình điều khiển này được thực hiện một cách cưỡng bức.

- Bốn nút nhấn S1, S2, S3, S4: Các phần tử nhập tín hiệu.
- Các tiếp điểm K1, K2, K3 và các mối nối liên kết là các phần tử xử lý.
- Các khởi động từ K1, K2, K3 là kết quả xử lý.

Nếu thay đổi mạch điện điều khiển ở phần xử lý bằng hệ PLC ta có thể biểu diễn hệ thống như sau:

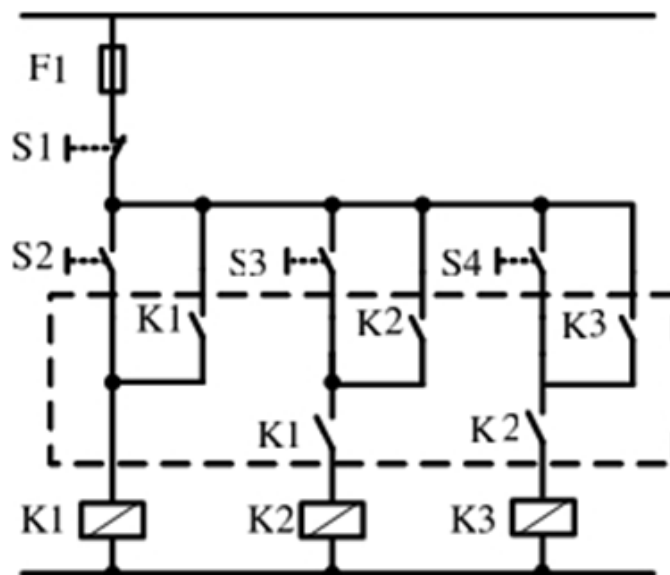
- Phần tử vào: Các nút nhấn S1, S2, S3, S4 vẫn giữ nguyên.
- Phần tử ra: Ba khởi động từ K1, K2, K3, để đóng và mở ba máy bơm vẫn giữ nguyên.
- Phần tử xử lý: Được thay thế bằng PLC.

Sơ đồ kết nối với PLC được cho như ở hình 2.3. Trình tự đóng mở theo yêu cầu đề ra sẽ được lập trình, chương trình sẽ được nạp vào bộ nhớ.

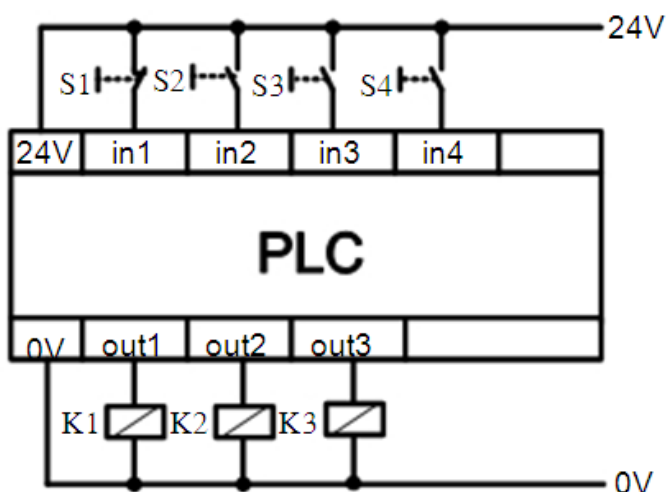
Bây giờ giả thiết rằng nhiệm vụ điều khiển sẽ thay đổi. Hệ thống ba máy bơm vẫn giữ nguyên, nhưng trình tự được thực hiện như sau: chỉ đóng được hai trong ba máy bơm hoặc mỗi máy bơm có thể hoạt động một cách độc lập. Như vậy theo yêu cầu mới đối với hệ thống điều khiển bằng relay điện phải thiết kế lại mạch điều khiển, sơ đồ lắp ráp phải thực hiện lại hoàn toàn mới. Sơ đồ mạch điều khiển biểu diễn như hình 2.4.

Như vậy mạch điều khiển sẽ thay đổi rất nhiều nhưng phần tử đưa tín hiệu vào và ra vẫn giữ nguyên, chi phí cho nhiệm vụ mới sẽ cao hơn.

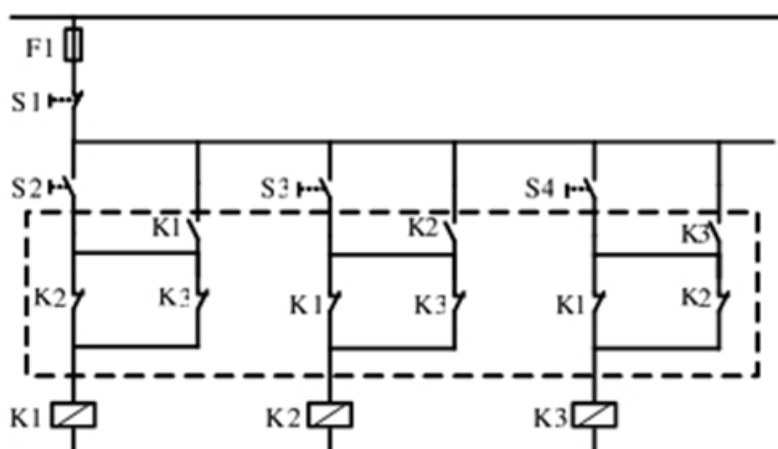
Nếu ta thay đổi hệ điều khiển trên bằng hệ điều khiển lập trình PLC, khi nhiệm vụ điều khiển thay đổi thì thực hiện sẽ nhanh hơn và đơn giản hơn bằng cách thay đổi lại chương trình.



Hình 3.2: Sơ đồ mạch được chuyển thành chương trình trong PLC



Hình 2.3: Sơ đồ kết nối với PLC



Hình 3.4: Sơ đồ mạch điều khiển 3 động cơ đã được thay đổi

Hệ điều khiển lập trình PLC có những ưu điểm sau:

- Thích ứng với những nhiệm vụ điều khiển khác nhau.
- Khả năng thay đổi đơn giản trong quá trình đưa thiết bị vào sử dụng.
- Tiết kiệm không gian lắp đặt.
- Hoạt động ổn định và độ tin cậy cao
- Tiết kiệm thời gian trong quá trình mở rộng và phát triển nhiệm vụ điều khiển bằng cách copy các chương trình.
- Các thiết bị điều khiển theo chuẩn.
- Không cần các tiếp điểm.
- v.v...

Hệ thống điều khiển lập trình PLC được sử dụng rất rộng rãi trong các ngành khác nhau:

- Điều khiển thang máy.
- Điều khiển các quá trình sản xuất khác nhau: sản xuất bia, sản xuất xi măng v.v
- Hệ thống rửa ô tô tự động.
- Thiết bị khai thác .
- Thiết bị đóng gói bao bì, tự động mạ và tráng kẽm v.v ...
- Thiết bị sấy.

3.3 Cấu trúc của một PLC

Các bộ điều khiển PLC được sản xuất theo dòng sản phẩm. Khi mới xuất xưởng, chúng chưa có một chương trình cho một ứng dụng nào cả. Tất cả các cổng logic cơ bản, chức năng nhớ, timer, counter .v.v... được nhà chế tạo tích hợp trong chúng và được kết nối với nhau bằng chương trình được viết bởi người dùng cho một nhiệm vụ điều khiển cụ thể nào đó. Bộ điều khiển PLC có nhiều loại khác nhau và được phân biệt với nhau qua các thành phần sau:

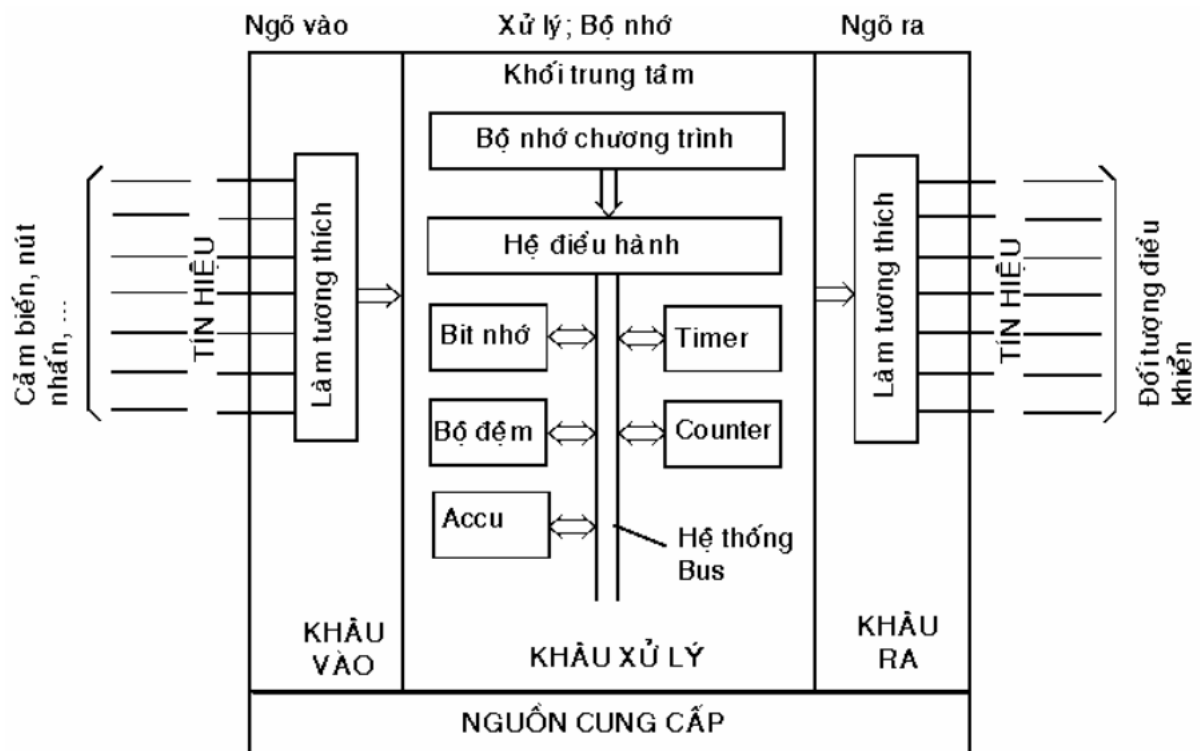
- Các ngõ vào và ra
- Dung lượng nhớ
- Bộ đếm (counter)
- Bộ định thời (timer)
- Bit nhớ
- Các chức năng đặc biệt
- Tốc độ xử lý
- Loại xử lý chương trình.
- Khả năng truyền thông.

Các bộ điều khiển lớn thì các thành phần trên được lắp thành các modul riêng. Đối với các bộ điều khiển nhỏ, chúng được tích hợp trong bộ điều khiển. Các bộ điều khiển nhỏ này có số lượng ngõ vào/ra cho trước cố định.

Bộ điều khiển được cung cấp tín hiệu bởi các tín hiệu từ các cảm biến ở ngõ vào của

nó. Tín hiệu này được xử lý tiếp tục thông qua chương trình điều khiển đặt trong bộ nhớ chương trình. Kết quả xử lý được đưa ra ngõ ra để đến đối tượng điều khiển hay khâu điều khiển ở dạng tín hiệu.

Cấu trúc của một PLC có thể được mô tả như hình vẽ sau:



Hình 3.5: Cấu trúc chung của bộ điều khiển lập trình PLC

* Bộ nhớ chương trình

Bộ nhớ chương trình trong PLC là một bộ nhớ điện tử đặc biệt có thể đọc được. Nếu sử dụng bộ nhớ đọc-ghi được (RAM), thì nội dung của nó luôn luôn được thay đổi ví dụ như trong trường hợp vận hành điều khiển. Trong trường hợp điện áp nguồn bị mất thì nội dung trong RAM có thể vẫn được giữ lại nếu như có sử dụng Pin dự phòng.

Nếu chương trình điều khiển làm việc ổn định, hợp lý, nó có thể được nạp vào một bộ nhớ cố định, ví dụ như EPROM, EEPROM. Nội dung chương trình ở EPROM có thể bị xóa bằng tia cực tím.

* Hệ điều hành

Sau khi bật nguồn cung cấp cho bộ điều khiển, hệ điều hành của nó sẽ đặt các counter, timer, dữ liệu và bit nhớ với thuộc tính non-retentive (không được nhớ bởi Pin dự phòng) cũng như ACCU về 0.

Để xử lý chương trình, hệ điều hành đọc từng dòng chương trình từ đầu đến cuối. Tương ứng hệ điều hành thực hiện chương trình theo các câu lệnh.

* Bit nhớ (Bit memory)

Các bit memory là các phần tử nhớ, mà hệ điều hành ghi nhớ trạng thái tín hiệu.

* Bộ đếm (Process Image)

Bộ đếm là một vùng nhớ, mà hệ điều hành ghi nhớ các trạng thái tín hiệu ở các ngõ vào ra nhị phân.

* Accumulator

Accumulator là một bộ nhớ trung gian mà qua nó timer hay counter được nạp vào hay thực hiện các phép toán số học.

* Counter, Timer

Timer và counter cũng là các vùng nhớ, hệ điều hành ghi nhớ các giá trị đếm trong nó.

* Hệ thống Bus

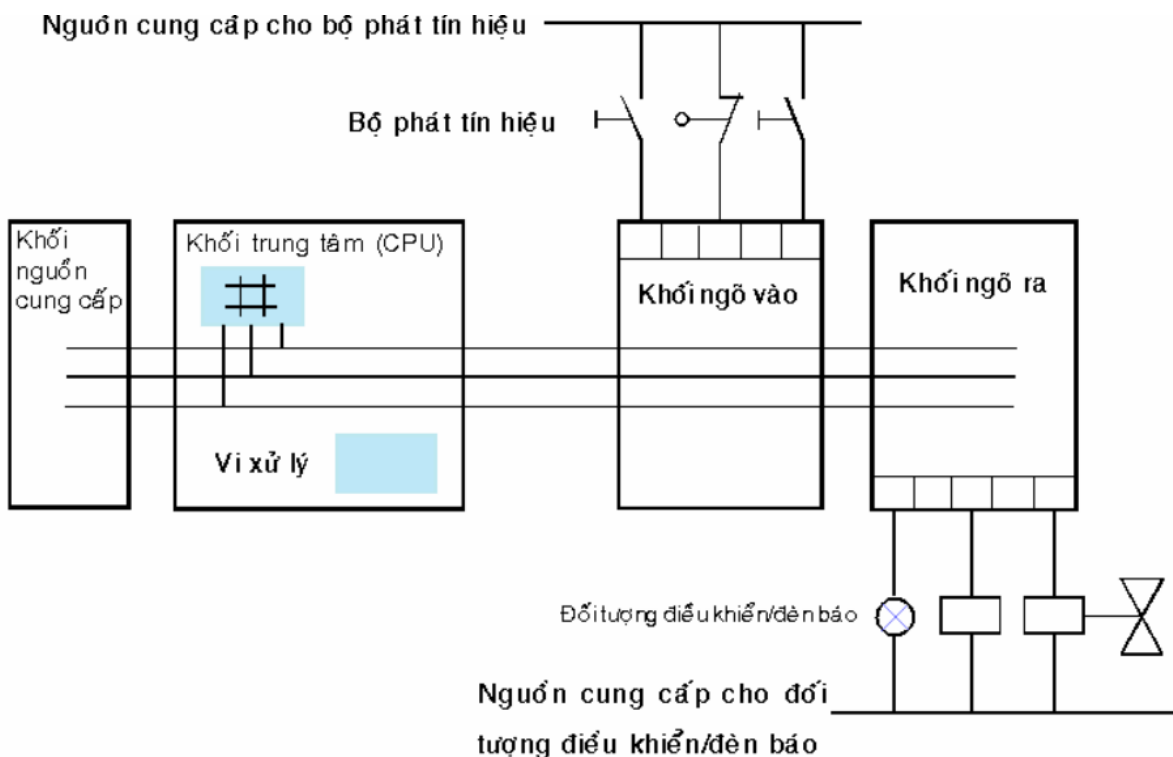
Bộ nhớ chương trình, hệ điều hành và các modul ngoại vi (các ngõ vào và ngõ ra) được kết nối với PLC thông qua Bus nối. Một Bus bao gồm các dây dẫn mà các dữ liệu được trao đổi. Hệ điều hành tổ chức việc truyền dữ liệu trên các dây dẫn này.

3.4 Các khối của PLC

Các khối khác nhau của một PLC được cho như hình 2.6.

3.4.1 Khối nguồn cung cấp

Khối nguồn có nhiệm vụ biến đổi điện áp lưới (110V hay 220V) thành điện áp thấp hơn cung cấp cho các khối của thiết bị tự động. Điện áp này là 24VDC. Các điện áp cho cảm biến, thiết bị điều chỉnh và các đèn báo nằm trong khoảng (24...220V) có thể được cung cấp thêm từ các nguồn phụ ví dụ như biến áp.



Hình 3.6: Các khối trong một PLC

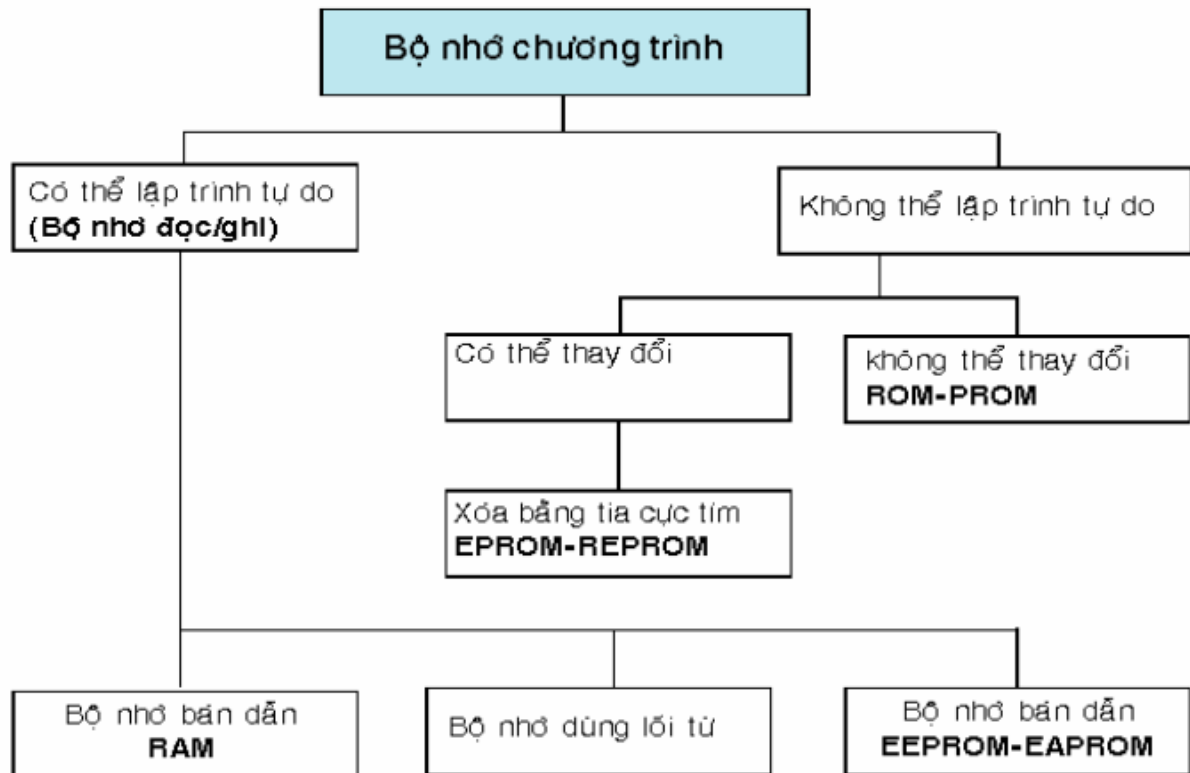
3.4.2 Bộ nhớ chương trình

Các phần tử nhớ là các linh kiện mà thông tin có thể được lưu trữ (được nhớ) trong nó ở dạng tín hiệu nhị phân. Trong PLC các bộ nhớ bán dẫn được sử dụng làm bộ nhớ chương trình. Một bộ nhớ bao gồm 512, 1024, 2048 . . . phần tử nhớ, các phần tử nhớ này sắp đặt theo các địa chỉ từ 0 tới 511, 1023 hoặc 2047 . . . Thông thường số lượng của các phần tử nhớ trong một bộ nhớ cho biết dung lượng của nó là bao nhiêu kilobyte (1kB = 1024 byte). Trong mỗi ô nhớ có thể mô tả một câu lệnh điều khiển nhờ thiết bị lập trình.

Mỗi phần tử nhị phân của một ô nhớ có thể có trạng thái tín hiệu "0" hoặc "1". Sơ đồ của một bộ nhớ chương trình được cho như hình 2.7.

*** Bộ nhớ đọc-ghi RAM (random-access memory)**

Bộ nhớ ghi-đọc có 1 số lượng các ô nhớ xác định. Mỗi ô nhớ có 1 dung lượng nhớ cố định và nó chỉ tiếp nhận 1 lượng thông tin nhất định. Các ô nhớ được ký hiệu bằng các địa chỉ riêng của nó. Bộ nhớ này chứa các chương trình còn sửa đổi hoặc các dữ liệu, kết quả tạm thời trong quá trình tính toán, lập trình. Đặc điểm của loại này là dữ liệu sẽ mất đi khi hệ thống mất điện. RAM được hình dung như một tủ chứa có nhiều ngăn kéo. Mỗi ngăn kéo được đánh số một địa chỉ và người ta có thể cất vào hoặc lấy các dữ liệu ra.



Hình 3.7: Sơ đồ một bộ nhớ chương trình

*** Bộ nhớ cố định ROM (read-only memory)**

Bộ nhớ cố định (ROM) chứa các thông tin không có khả năng xóa được và không thể thay đổi được. Các thông tin này do các nhà sản xuất viết ra và không thể thay đổi được. Chương trình trong bộ nhớ ROM có nhiệm vụ sau:

- Điều khiển và kiểm tra các chức năng hoạt động của CPU. Được gọi là hệ điều hành.
- Dịch ngôn ngữ lập trình thành ngôn ngữ máy.

Một ROM có thể so sánh với một quyển sách. Trong đó nó chứa các thông tin cố định, không thể thay đổi được và ta chỉ đọc các thông tin đó mà thôi. Đặc điểm của loại này là dữ liệu vẫn tồn tại khi mất điện.

*** EPROM (erasable read-only memory)**

EPROM là một bộ nhớ cố định có thể lập trình và xóa được. Nội dung của EPROM có thể xóa bằng tia cực tím và có thể lập trình lại.

*** EEPROM (electrically erasable read-only memory)**

EEPROM là bộ nhớ cố định có thể lập trình và xóa bằng điện. Mỗi ô nhớ trong

EEPROM cho phép lập trình và xóa bằng điện.

3.4.3 Khối trung tâm (CPU)

Khối CPU là loại khối có chứa bộ vi xử lý, hệ điều hành, bộ nhớ, các bộ thời gian, bộ đếm, cổng truyền thông ... và có thể còn có một vài cổng vào ra số. Các cổng vào ra số có trên CPU được gọi là cổng vào/ra onboard.

3.4.4 Khối vào (INPUT)

Các ngõ vào của khối này sẽ được kết nối với các bộ chuyển đổi tín hiệu và biến đổi các tín hiệu này thành tín hiệu phù hợp với tín hiệu xử lý của CPU. Dựa vào loại tín hiệu vào sẽ có các khối ngõ vào tương ứng. Gồm có hai loại khối vào cơ bản sau:

- **Khối vào số (DI: Digital Input):**

Các ngõ vào của khối này được kết nối với các bộ chuyển đổi tạo ra tín hiệu nhị phân như nút nhấn, công tắc, cảm biến tạo tín hiệu nhị phân .v.v... Do tín hiệu tại ngõ vào có thể có mức logic tương ứng với các điện áp khác nhau, do đó khi sử dụng cần phải chú ý đến điện áp cần thiết cung cấp cho khối vào phải phù hợp với điện áp tương ứng mà bộ chuyển đổi tín hiệu nhị phân tạo ra.

Ví dụ: Các nút nhấn, công tắc được nối với nguồn 24VDC thì yêu cầu phải sử dụng khối vào có nguồn cung cấp cho nó là 24VDC.

- **Khối vào tương tự (AI: Analog Input):**

Khối này có nhiệm vụ biến đổi tín hiệu tương tự (hay còn gọi là tín hiệu analog) thành tín hiệu số. Các ngõ vào của khối này được kết nối với các bộ chuyển đổi tạo ra tín hiệu analog như cảm biến nhiệt độ (Thermocouple), cảm biến lưu lượng, ngõ ra analog của biến tần .v.v... Khi sử dụng các khối vào analog cần phải chú ý đến loại tín hiệu analog được tạo ra từ các bộ chuyển đổi (cảm biến)

Ví dụ: Các cảm biến tạo ra tín hiệu analog là dòng điện (4..20 mA) thì phải sử dụng ngõ vào analog là loại nhận tín hiệu dòng điện (4..20 mA). Nếu cảm biến tạo ra tín hiệu analog là điện áp (0..5V) thì phải sử dụng ngõ vào analog nhận tín hiệu là điện áp (0..5V).

3.4.5 Khối ra (OUT PUT)

Khối này có nhiệm vụ khuếch đại các tín hiệu sau xử lý của CPU (được gửi đến vùng đệm ra) cung cấp cho đối tượng điều khiển là cuộn dây, đèn báo, van từ .v.v.. Tùy thuộc vào đối tượng điều khiển nhận tín hiệu dạng nào mà sẽ có các khối ra tương ứng. Gồm có hai loại khối ra tiêu biểu:

- **Khối ra số (DO: Digital Output):**

Các ngõ ra của khối này được kết nối với các đối tượng điều khiển nhận tín hiệu nhị phân như đèn báo, cuộn dây relay .v.v... Vì đối tượng điều khiển nhận tín hiệu nhị phân sử dụng nhiều cấp điện áp khác nhau nên khi sử dụng các khối ra số cần phải chú ý đến điện áp cung cấp cho nó có phù hợp với điện áp cung cấp cho đối tượng điều khiển hay không. Theo loại điện áp sử dụng, ngõ ra số được phân thành hai loại:

- Điện áp một chiều (DC: Direct Current): Gồm có hai loại ngõ ra là Transistor và relay. Thông thường trong công nghiệp điện áp một chiều được sử dụng là 24V.

- Điện áp xoay chiều (AC: Alternative Current): Gồm có hai loại ngõ ra là Relay và TRIAC.

- **Khối ra tương tự (AO: Analog Output):**

Khối này có nhiệm vụ biến đổi tín hiệu số được gửi từ CPU đến đối tượng điều khiển thành tín hiệu tương tự. Các ngõ ra của khối này được kết nối với các đối tượng điều khiển nhận tín hiệu tương tự như ngõ vào analog của biến tần, van tỷ lệ, .v.v... Khi sử dụng các ngõ ra tương tự cần chú ý đến loại tín hiệu tương tự cung cấp cho đối tượng điều khiển có phù hợp với tín hiệu tương tự mà đối tượng điều khiển cần nhận hay không.

Ví dụ: Ngõ vào analog của biến tần nhận tín hiệu là điện áp (0..10V) thì nhất thiết phải sử dụng ngõ ra tương tự tạo ra tín hiệu analog là điện áp (0..10V).

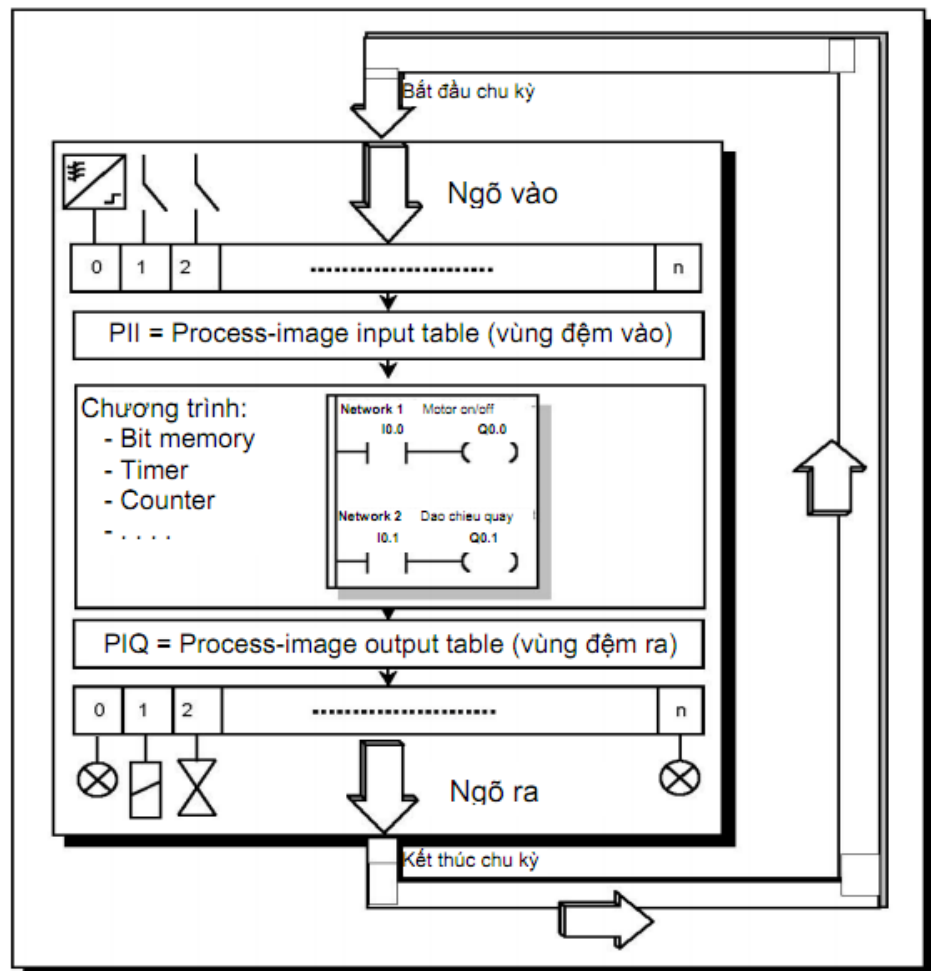
3.4.6 Các khối đặc biệt

Ngoài ra còn có một số khối khác đảm nhận các chức năng đặc biệt như xử lý truyền thông, thực hiện các chức năng đặc biệt như: điều khiển vị trí, điều khiển vòng kín, đếm tốc độ cao .v.v...

Tùy thuộc vào từng loại PLC mà các khối trên có thể ở các dạng module riêng hoặc được tích hợp chung trong khối xử lý trung tâm (CPU).

3.5 Phương thức thực hiện chương trình trong PLC

Hình vẽ minh họa việc xử lý chương trình trong CPU được cho như hình 2.8



Hình 2.8: Chu kỳ quét trong PLC

PLC thực hiện chương trình theo chu trình lặp. Mỗi vòng lặp được gọi là vòng quét (scan). Mỗi vòng quét được bắt đầu bằng giai đoạn chuyển dữ liệu từ các cổng vào số tới vùng bộ đệm ảo ngõ vào (I), tiếp theo là giai đoạn thực hiện chương trình. Trong từng dòng quét, chương trình được thực hiện từ lệnh đầu tiên đến lệnh kết thúc. Sau giai đoạn thực hiện chương trình là giai đoạn chuyển các nội dung của bộ đệm ảo ngõ ra (Q) tới các cổng ra số.

Vòng quét được kết thúc bằng giai đoạn truyền thông nội bộ và kiểm tra lỗi.

Thời gian cần thiết để PLC thực hiện được một vòng quét gọi là thời gian vòng quét (Scan time). Thời gian vòng quét không cố định, tức là không phải vòng quét nào cũng được thực hiện trong một khoảng thời gian như nhau. Có vòng quét thực hiện lâu, có vòng quét thực hiện nhanh tùy thuộc vào số lệnh trong chương trình được thực hiện, vào khối lượng dữ liệu truyền thông ... trong vòng quét đó.

Như vậy giữa việc đọc dữ liệu từ đối tượng để xử lý, tính toán và việc gửi tín hiệu điều khiển tới đối tượng có một khoảng thời gian trễ đúng bằng thời gian vòng quét. Nói cách khác, thời gian vòng quét quyết định tính thời gian thực của chương trình điều khiển trong PLC. Thời gian quét càng ngắn, tính thời gian thực của chương trình càng cao.

Tại thời điểm thực hiện lệnh vào/ra, thông thường lệnh không làm việc trực tiếp với cổng vào/ra mà chỉ thông qua bộ đệm ảo của cổng trong vùng nhớ tham số. Việc truyền thông giữa bộ đệm ảo với ngoại vi do hệ điều hành CPU quản lý. Ở một số module CPU, khi gặp lệnh vào/ra ngay lập tức, hệ thống sẽ cho dừng mọi công việc khác, ngay cả chương trình xử lý ngắt, để thực hiện lệnh trực tiếp với cổng vào/ra.

CHƯƠNG 4: PLC SIMATIC S7-200

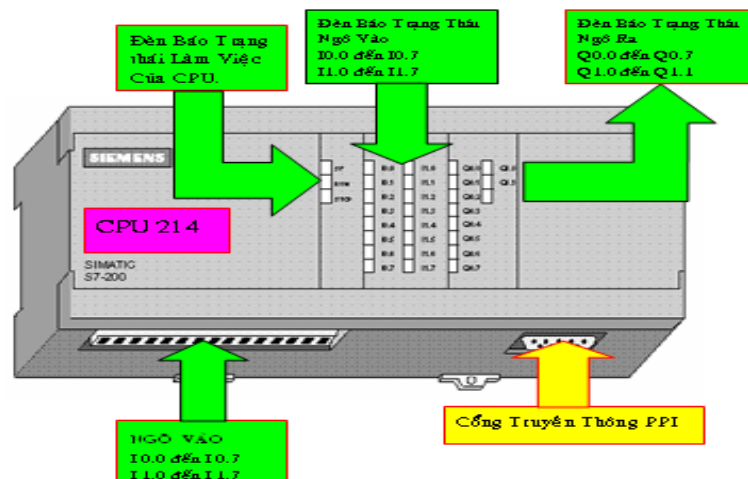
4.1 Cấu hình phần cứng

4.1.1 Khối xử lý trung tâm

PLC S7-200 là thiết bị điều khiển lập trình loại nhỏ (micro PLC) của hãng Siemens (CHLB Đức) có cấu trúc theo kiểu modul và có các modul mở rộng. Thành phần cơ bản của S7 - 200 là khối xử lý trung tâm (CPU: Central Processing Unit) bao gồm hai chủng loại: CPU 21x và CPU 22x. Mỗi chủng loại có nhiều CPU. Loại CPU 21x ngày nay không còn sản xuất nữa, tuy nhiên hiện vẫn còn sử dụng rất nhiều trong các trường học và trong sản xuất. Tiêu biểu cho loại này là CPU 214. CPU 214 có các đặc tính như sau:

- Bộ nhớ chương trình (chứa trong EEPROM): 4096 Byte (4 kByte)
- Bộ nhớ dữ liệu (Vùng nhớ V): 4096 Byte (trong đó 512 Byte chứa trong EEPROM)
- Số lượng ngõ vào: 14
- Số lượng ngõ ra: 10 ngõ ra digital tích hợp trong CPU
- Số module mở rộng: 7 gồm cả module analog
- Số lượng vào/ra số cực đại: 64
- Số lượng Timer : 128 Timer chia làm 3 loại theo độ phân giải khác nhau: 4 Timer 1ms, 16 Timer 10 ms và 108 Timer có độ phân giải 100ms.
- Số lượng Counter: 128 bộ đếm chia làm hai loại: 96 Counter Up và 32 Counter Up/Down.
- Bit memory (Vùng nhớ M): 256 bit
- Special memory (SM) : 688 bit dùng để thông báo trạng thái và đặt chế độ làm việc.
- Có phép tính số học
- Bộ đếm tốc độ cao (High-speed counters): 2 counter 2 KHz và 1 counter 7 KHz
- Ngõ vào analog tích hợp sẵn (biến trở): 2.
- Các chế độ ngắt và xử lý ngắt gồm: ngắt truyền thông, ngắt theo sườn lên hoặc xuống, ngắt thời gian, ngắt của bộ đếm tốc độ cao và ngắt truyền xung.
- Toàn bộ vùng nhớ không bị mất dữ liệu trong khoảng thời gian 190 giờ khi PLC bị mất nguồn nuôi.

Sơ đồ bề mặt của bộ điều khiển logic khả trình S7-200 CPU 214 được cho như hình 4.1.



*** Mô tả các đèn báo trên CPU 214:**

- SF (Đèn đỏ): Đèn đỏ SF báo hiệu hệ thống bị lỗi. Đèn SF sáng lên khi PLC có lỗi.
- RUN (Đèn xanh): cho biết PLC đang ở chế độ làm việc và thực hiện chương trình được nạp vào trong bộ nhớ chương trình của PLC.
- STOP (Đèn vàng): Đèn vàng STOP chỉ định PLC đang ở chế độ dừng. Dừng chương trình đang thực hiện lại.
- I x.x (Đèn xanh): Đèn xanh ở cổng vào chỉ định trạng thái tức thời của cổng (x.x = 0.0 - 1.5). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.
- Qy.y (Đèn xanh): Đèn xanh ở cổng ra chỉ định trạng thái tức thời của cổng (y.y = 0.0 - 1.1). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.

Hiện nay, CPU 22x với nhiều tính năng vượt trội đã thay thế loại CPU 21x và hiện đang được sử dụng rất nhiều. Tiêu biểu cho loại này là CPU 224.

Thông tin về CPU 22x được cho như bảng 4.1 và hình dáng CPU 224 ở hình 4.2.

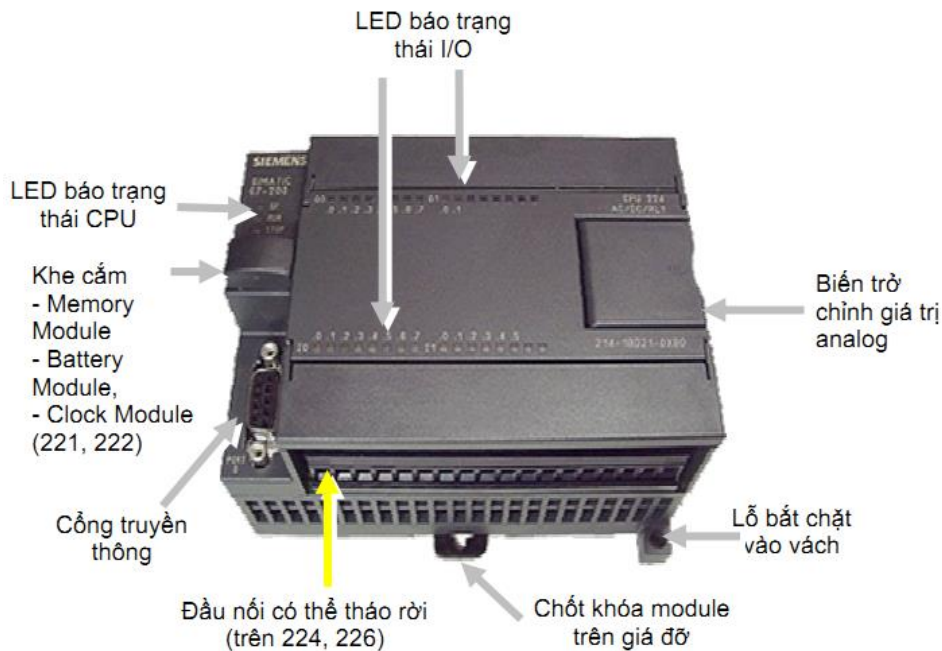
Đặc điểm	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
I/O trên CPU Digital Analog	6DI/4DO -	8DI/6DO -	14DI/10DO -	14DI/10DO 2AI/1AO	24DI/16DO -
Số module mở rộng max.	0	2	7	7	7
Bộ nhớ chương trình	4KB	4KB	8KB	12KB	16KB
Bộ nhớ dữ liệu	2KB	2KB	8KB	10KB	10KB
Thời gian xử lý	0,37 μ s	0,37 μ s	0,37 μ s	0,37 μ s	0,37 μ s
Memory bits/counters/timers	256/256/256	256/256/256	256/256/256	256/256/256	256/256/256
High-speed counters	4 x 30 kHz	4 x 30 kHz	6 x 30 kHz	4 x 30 kHz 2x 200 kHz	6 x 30 kHz
Real-time clock	card	card	Tích hợp	Tích hợp	Tích hợp
Ngõ ra xung	2 x 20 kHz	2 x 20 kHz	2 x 20 kHz	2 x 100 kHz	2 x 20 kHz
Cổng giao tiếp	1x RS-485	1x RS-485	1x RS-485	2x RS-485	2x RS-485
Biến trở analog trên CPU	1	1	2	2	2

Bảng 4.1: Bảng dữ liệu về CPU họ 22x

* Chọn chế độ làm việc cho PLC

Công tắc chọn chế độ làm việc nằm ở phía trên, có ba vị trí cho phép chọn các chế độ làm việc khác nhau của 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.
- STOP: Cưỡng bức PLC dừng chương trình đang chạy và chuyển sang chế độ STOP. Ở chế độ STOP, PLC cho phép hiệu chỉnh, nạp, xóa một chương trình.
- TERM: Cho phép người dùng từ máy tính quyết định chọn một trong hai chế độ làm việc cho PLC hoặc RUN hoặc STOP.



Hình 4.2: Bộ điều khiển lập trình CPU 224

* Cổng truyền thông

S7-200 sử dụng cổng truyền thông nối tiếp RS485 với phích nối 9 chân để phục vụ cho việc ghép nối với thiết bị lập trình hoặc với các trạm PLC khác. Tốc độ truyền cho máy lập trình kiểu PPI là 9600 baud. Tốc độ truyền cung cấp của PLC theo kiểu tự do là từ 300 baud đến 38400 baud.

Để ghép nối S7-200 với máy lập trình PG720 (hãng Siemens) hoặc với các loại máy lập trình thuộc họ PG7xx có thể sử dụng một cáp nối thẳng qua MPI. Cáp đó đi kèm theo máy lập trình.

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

* Card nhớ, pin, clock (CPU 221, CPU222)

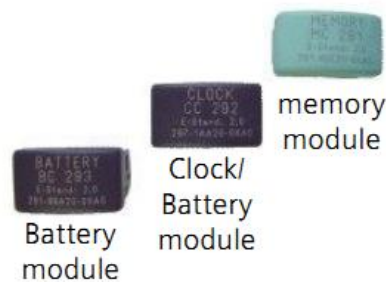
S7-200 cung cấp nhiều biện pháp đảm bảo cho chương trình người dùng, dữ liệu chương trình và cấu hình dữ liệu được duy trì sau:

Một tụ điện với điện dung lớn cho phép nuôi bộ nhớ RAM sau khi bị mất nguồn điện cung cấp. Tùy theo loại CPU mà thời gian lưu trữ có thể kéo dài nhiều ngày. Chẳng hạn ở CPU 224 là khoảng 100 giờ

Vùng nhớ EEPROM cho phép lưu chương trình, các vùng nhớ được người dùng chọn chứa vào EEPROM và cấu hình dữ liệu.

Cho phép gắn thêm Pin để nuôi RAM và cho phép kéo dài thêm thời gian lưu trữ dữ liệu, có thể lên đến 200 ngày kể từ khi mất nguồn điện. Nguồn của Pin sẽ được lấy sau khi tụ điện đã xả hết.

- Card nhớ: được sử dụng để lưu trữ chương trình. Chương trình chứa trong card nhớ bao gồm: program block, data block, system block, công thức (recipes), dữ liệu đo (data logs), và các giá trị cường độ (force values).
- Card pin: dùng để mở rộng thời gian lưu trữ các dữ liệu trong bộ nhớ. Nguồn pin được tự động chuyển sang khi tụ trong PLC cạn. Pin có thể sử dụng đến 200 ngày.
- Card Clock/Battery module: đồng hồ thời gian thực (Real Time Clock) cho CPU 221, 222 và nguồn pin để nuôi đồng hồ và dữ liệu. Thời gian sử dụng đến 200 ngày.



Hình 4.3: Hình dáng các module

* Biến trở chỉnh giá trị analog:

Hai biến trở này được sử dụng như hai ngõ vào analog cho phép điều chỉnh các biến cần phải thay đổi và sử dụng trong chương trình.

4.1.2 Khối mở rộng

Trên các CPU đã tích hợp sẵn một số các ngõ vào và ngõ ra số, chẳng hạn như CPU 224 DC/DC/DC có sẵn 16 ngõ vào và 14 ngõ ra. Tuy nhiên trong thực tế, xuất phát từ yêu cầu điều khiển như: cần nhiều hơn số ngõ vào/ra có sẵn, có sử dụng tín hiệu analog hay có các yêu cầu về truyền thông, nối mạng các PLC...mà ta phải gắn thêm vào CPU các khối mở rộng (Expansion module) có các chức năng khác nhau (bảng 4.2).

4.1.2.1 Digital module

Các module số gắn thêm vào khối CPU để mở rộng số lượng các ngõ vào/ra số.

- Khối ngõ vào số DI (Digital Input): Siemens sản xuất các khối ngõ vào số như: DI8 x 24VDC, DI8 x AC120/230V, DI16 x 24VDC.
- Khối ngõ ra số (Digital Output): Các ngõ ra này được chia ra làm 3 loại là ngõ ra DC, ngõ ra AC và ngõ ra relay. Điện áp ngõ ra có thể là 24Vdc hoặc 230Vac tùy loại, với số lượng ngõ ra có thể là 4 hoặc 8.

Ngoài ra còn có sự kết hợp các ngõ vào và ra số trên cùng một module.

4.1.2.2 Analog module

Ngoại trừ CPU 224XP có tích hợp sẵn 2 ngõ vào và 1 ngõ ra analog (2AI/1AO) để kết nối với ngoại vi nhận và phát tín hiệu analog, thì hầu hết các CPU khác của họ S7-200 đều không có tích hợp sẵn. Vì vậy khi điều khiển với tín hiệu analog thì yêu cầu người sử dụng phải gắn thêm các khối analog.

- Khối ngõ vào tương tự AI (Analog Input): Tín hiệu analog ngõ vào có thể là tín hiệu

điện áp hoặc dòng điện. Tùy thuộc vào tín hiệu analog cần đọc là loại nào mà người sử dụng có thể cài đặt cho phù hợp bằng các công tắc được gắn trên module (Chi tiết xem chương xử lý tín hiệu analog).

Hiện có các khối ngõ vào: 4AI, 8AI. Đối với tín hiệu analog được tạo ra bởi Thermocouple (cặp nhiệt) và RTD thì sử dụng các module đo nhiệt tương ứng (bảng 4.2).

- Khối ngõ ra tương tự AO (Analog Output): Tín hiệu tương tự này có thể là điện áp hoặc dòng điện tùy theo người dùng cài đặt. Tín hiệu ra là điện áp nằm trong khoảng $\pm 10\text{Vdc}$ tương ứng với giá trị số từ -32000 tới + 32000 và tín hiệu dòng điện nằm trong khoảng từ 0 - 20mA tương ứng với giá trị số từ 0 tới +32000.

Ngoài các khối trên còn có các khối có sự kết hợp cả 2 loại tín hiệu vào và ra analog trên cùng một khối.

Các khối mở rộng	Loại			
Digital module				
Input	8 x DC In	8 x AC In	16 x DC In	
Output	4 x DC Out	4 x Relay	8 x Relay	
	8 x DC Out	8 x AC Out		
Tổ hợp	4 x DC In/	8 x DC In/	16 x DC In/	32 x DC In/
	4 x DC Out	8 x DC Out	16x DC Out	32x DC Out
	4 x DC In/	8 x DC In/	16 x DC In/	32 x DC In/
	4 x Relay	8 x Relay	16x DC Out	32x Relay
Analog module				
Input	4 x Analog In	8 x Analog In	4xThermocouple In	
	2 x RTD In	2 x RTD In		
Output	2 x Analog Out	4 x Analog Out		
Tổ hợp	4 x Analog In			
	4 x Analog Out			
Intelligent module				
	Position	Modem	PROFIBUS-DP	
	Ethernet	Ethernet IT		
Các module khác				
	AS-Interface	SIWAREX MS		

Bảng 4.2: Các loại khối mở rộng

4.1.2.3 Intelligent module

Các PLC S7-200 có thể nối vào các loại mạng khác nhau để tăng cường khả năng mở rộng, truyền thông với các thiết bị khác trong hệ thống tự động hóa.

- Master trong mạng AS-Interface: Giao tiếp AS-i (Actuator Sensor Interface) hay giao tiếp actuator/sensor là hệ thống kết nối cho cấp quá trình thấp nhất trong hệ thống tự động hóa nhằm tối ưu hóa việc kết nối cảm biến và cơ cấu chấp hành với thiết bị tự động hóa. Với module CP243-2 cho phép kết nối mạng AS-Interface vào PLC S7-200 và đóng vai trò là master.

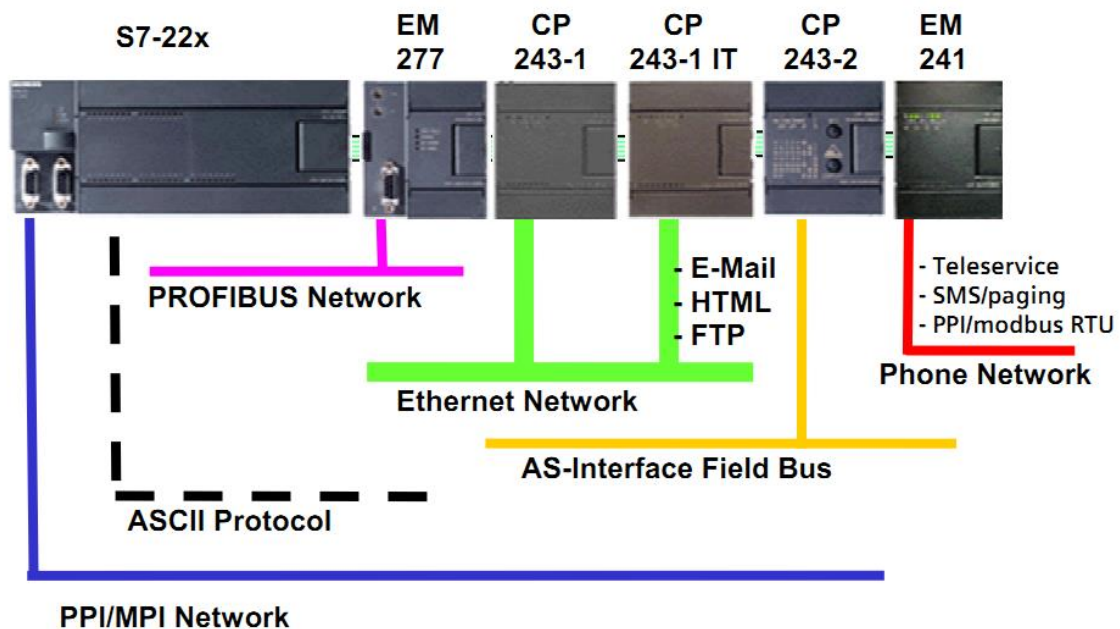
- Kết nối vào mạng PROFIBUS-DP: Các PLC S7-200 có thể kết nối vào mạng Profibus hoạt động như một DP Slave nhờ vào khối mở rộng EM277. Việc sử dụng EM277 cho phép PLC S7-200 có thể kết nối truyền thông với các thiết bị trong mạng Profibus như: PLC S7-300, S7-400, màn hình điều khiển...

- Kết nối vào mạng Ethernet: Để có thể kết nối S7-200 vào mạng Industrial Ethernet thì cần có khối CP 243-1. Đây là khối truyền thông cho phép các PLC S7-200 có thể được cấu hình, lập trình, chẩn đoán từ xa qua Ethernet nhờ phần mềm STEP 7 Micro/win. Giúp cho các CPU S7-200 có thể giao tiếp với các S7-200 khác, S7-300 hay S7-400 qua Ethernet. Các CPU có thể sử dụng là họ CPU 22X. Có thể thực hiện cấu hình cho các CPU vào mạng Ethernet nhờ vào Wizard (Menu Tools → Ethernetwizard).

- Internet Technology: Khối mở rộng CP 243-1 IT cho phép các CPU S7-200 có thể thực hiện các giám sát hay thay đổi qua trình duyệt Web từ một PC có nối mạng. Các thông báo chẩn đoán có thể gửi qua email từ một hệ thống. Sử dụng các chức năng IT cho phép trao đổi các tập tin dữ liệu với các máy tính hay các hệ thống điều khiển khác. Mỗi một khối CP 243-1 IT chỉ nên kết nối cho 2 CPU S7-200.

- Modem module: Cho phép kết nối trực tiếp S7-200 vào đường dây điện thoại, và cung cấp truyền thông giữa S7-200 và Step 7- micro/Win. Với công cụ Modem Expansion wizard cho phép thiết lập một modem ở xa hoặc kết nối S7-200 với một thiết bị ở xa qua modem.

Khả năng truyền thông của S7-200 được cho như hình 4.4.



Hình 4.4: Khả năng truyền thông của PLC S7-200

4.1.2.4 Function module

Là các khối chức năng thực hiện các chức năng đặc biệt như điều khiển vị trí (position module), cân (SIWREX MS).

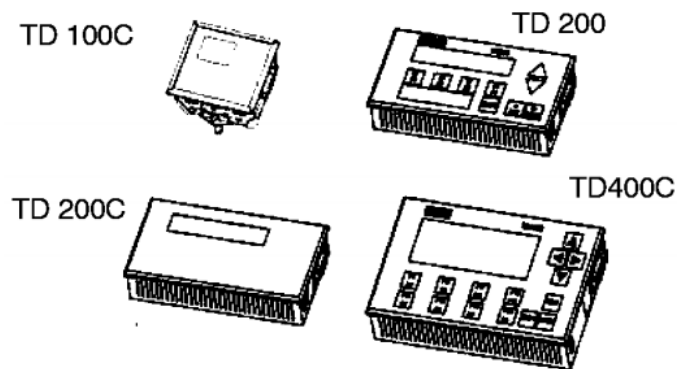
- Position module: Module vị trí được sử dụng để điều khiển tốc độ và vị trí của động cơ bước (stepper motor) hoặc động cơ servo (servo motor). Với công cụ Position Control wizard trong phần mềm STEP 7--Micro/WIN để thiết lập cấu hình cho module điều khiển vị trí. Module điều khiển vị trí thường được sử dụng là EM253.

SIWAREX MS: Là module cân đa năng và linh hoạt, nó được sử dụng với các hệ thống cân hoặc đo lực sử dụng PLC S7-200.

4.2 Màn hình điều khiển

Trong các yêu cầu điều khiển có giám sát thì đối với các PLC S7-200 chúng ta có thể gắn thêm các màn hình để điều khiển và giám sát. Hiện có các loại là: màn hình hiển thị dòng văn bản (Text Display), màn hình điều khiển bằng bàn phím (Operator panel) và màn hình cảm ứng (Touch Panel).

* **Bảng điều khiển hiển thị dòng văn bản (Text Display):** Các màn hình này có giá thành thấp, cho phép người vận hành máy có thể xem, giám sát bằng các dòng văn bản và thay đổi các thông số hay chế độ hoạt động của hệ thống điều khiển bằng các phím trên bảng điều khiển. Gồm có các loại là TD100C, TD200C, TD 200, TD400C (hình 4.5).



Hình 4.5: Bảng điều khiển hiển thị dòng văn bản

Các bảng điều khiển này có thể được thiết lập các thông báo và nút nhấn điều khiển dễ dàng bằng công cụ Text Display wizard (menu lệnh Tools > Text Display Wizard) trong STEP 7--Micro/WIN.

* **Operator Panel và Touch Panel:** Các màn hình được ứng dụng điều khiển và giám sát các máy móc, thiết bị nhỏ. Thời gian thiết lập cấu hình và vận hành nhanh với phần mềm WinCC flexible. Gồm có các loại: OP 73micro, TP 177micro (màn hình này thay thế các màn hình trước TP 070/TP 170micro) (hình 4.6).



Hình 4.6: Màn hình OP 73micro và TP 177mico.

4.3 Các vùng nhớ

Bộ nhớ của các PLC S7-200 được chia ra làm các vùng nhớ như bảng 4.3.

* Vùng nhớ đệm ngõ vào số I:

CPU sẽ đọc trạng thái tín hiệu của tất cả các ngõ vào số ở đầu mỗi chu kỳ quét, sau đó sẽ chứa các giá trị này vào vùng nhớ đệm ngõ vào. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

* Vùng nhớ đệm ngõ ra số Q:

Trong quá trình xử lý chương trình CPU sẽ lưu các giá trị xử lý thuộc vùng nhớ ngõ ra vào đây. Tại cuối mỗi vòng quét CPU sẽ sao chép nội dung vùng nhớ đệm này và chuyển ra các ngõ ra vật lý. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

*** Vùng nhớ biến V:**

Sử dụng vùng nhớ V để lưu trữ các kết quả phép toán trung gian có được do các xử lý logic của chương trình. Cũng có thể sử dụng vùng nhớ để lưu trữ các dữ liệu khác liên quan đến chương trình hay nhiệm vụ điều khiển. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

*** Vùng nhớ M:**

Có thể coi vùng nhớ M như là các relay điều khiển trong chương trình để lưu trữ trạng thái trung gian của một phép toán hay các thông tin điều khiển khác. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

*** Vùng nhớ bộ định thời T:**

S7-200 cung cấp vùng nhớ riêng cho các bộ định thời, các bộ định thời được sử dụng cho các yêu cầu điều khiển cần trì hoãn thời gian. Giá trị thời gian sẽ được đếm tăng dần theo 3 độ phân giải là 1ms, 10ms và 100ms.

*** Vùng nhớ bộ đếm C:**

Có 3 loại bộ đếm là bộ đếm lên, bộ đếm xuống và bộ đếm lên-xuống. Các bộ đếm sẽ tăng hoặc giảm giá trị hiện hành khi tín hiệu tại ngõ vào thay đổi trạng thái từ mức thấp lên mức cao.

*** Vùng nhớ bộ đếm tốc độ cao HC (High speed Counter):**

Các bộ đếm tốc độ cao được sử dụng để đếm các sự kiện tốc độ cao độc lập với vòng quét của CPU. Giá trị đếm là số nguyên 32 bit có dấu. Để truy xuất giá trị đếm của các bộ đếm tốc độ cao cần xác định địa chỉ của bộ đếm tốc độ cao, sử dụng vùng nhớ HC và số của bộ đếm, ví dụ HC0. Giá trị đếm hiện hành của các bộ đếm tốc độ cao là các giá trị chỉ đọc và truy xuất theo double word.

*** Các thanh ghi AC (Accumulators):**

Các thanh ghi AC là các phần tử đọc/ghi mà có thể được dùng để truy xuất giống như bộ nhớ. Chẳng hạn, có thể sử dụng các thanh ghi để truy xuất các thông số từ các chương trình con (Subroutine) và lưu trữ các giá trị trung gian để sử dụng cho tính toán. Các CPU S7-200 có 4 thanh ghi là AC0, AC1, AC2 và AC3. Chúng ta có thể truy xuất dữ liệu trong các thanh ghi này theo Byte, Word, và Doubleword.

*** Vùng nhớ đặc biệt SM (Special Memory):**

Các bit SM là các phần tử cho phép truyền thông tin giữa CPU và chương trình người dùng. Có thể sử dụng các bit này để chọn lựa và điều khiển một số chức năng đặc biệt của CPU, chẳng hạn như bit lên mức 1 trong vòng quét đầu tiên, các bit phát ra các xung có tần số 1Hz...Chúng ta truy xuất vùng nhớ SM theo bit, byte, word, doubleword.

*** Vùng nhớ cục bộ L (Local Memory Area):**

Vùng nhớ này có độ lớn 64 Byte, trong đó 60 byte có thể được dùng như vùng nhớ cục bộ hay chuyển các thông số tới các chương trình con, 4 byte cuối dùng cho hệ thống. Vùng nhớ này tương tự như vùng nhớ biến V chỉ khác ở chỗ các biến vùng nhớ V cho phép sử dụng ở tất cả các khối chương trình còn vùng nhớ L chỉ có tác dụng trong phạm vi soạn thảo của một khối chương trình mà thôi. Vị trí biến thuộc vùng nhớ L trong chương trình chính thì không thể sử dụng ở chương trình con và ngược lại.

*** Vùng nhớ ngõ vào tương tự AI (Analog Inputs):**

Các PLC S7-200 chuyển giá trị một tương tự (chẳng hạn điện áp hay nhiệt độ) thành giá trị số và chứa vào một vùng nhớ 16 bit. Bởi vì các giá trị tương tự chiếm một vùng nhớ word

nên chúng luôn luôn có các giá trị word chẵn, chẳng hạn như AIW0, AIW2, AIW4..và là các giá trị chỉ đọc.

* Vùng nhớ ngõ ra tương tự AQ (Analog Outputs):

Các PLC S7-200 chuyển một giá trị số 16 bit sang giá trị điện áp hoặc dòng điện, tương ứng với giá trị số (digital). Giống như các ngõ vào tương tự chúng ta chỉ có thể truy xuất các ngõ ra tương tự theo word. Và là các giá trị word chẵn, chẳng hạn như AQW0, AQW2, AQW4.

Mô tả	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU226
Kích thước chương trình người dùng	4 KB	4 KB	8 KB	12 KB	16 KB
Kích thước dữ liệu	2 KB	2 KB	8 KB	10 KB	10 KB
Vùng đếm vào số	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ... I15.7
Vùng đếm ra số	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ... Q15.7
Ngõ vào analog	AIW0 ... AIW30	AIW0 ... AIW30	AIW0 ... AIW62	AIW0 ... AIW62	AIW0 ... AIW62
Ngõ ra analog	AQW0 ... AQW30	AQW0 ... AQW30	AQW0 ... AQW62	AQW0 ... AQW62	AQW0 ... AQW62
Vùng nhớ biến (V)	VB0 ... VB2047	VB0 ... VB2047	VB0 ... VB8191	VB0 ... VB10239	VB0 ... VB10239
Vùng nhớ cục bộ (L)	LB0 ... LB63	LB0 ... LB63	LB0 ... LB63	LB0 ... LB63	LB0 ... LB63
Vùng nhớ bit (M)	M0.0 ... M31.7	M0.0 ... M31.7	M0.0 ... M31.7	M0.0 ... M31.7	M0.0 ... M31.7
Vùng nhớ đặc biệt Chỉ đọc (SM)	SM0.0 ... SM179.7 SM0.0 ... SM29.7	SM0.0 ... SM299.7 SM0.0 ... SM29.7	SM0.0 ... SM549.7 SM0.0 ... SM29.7	SM0.0 ... SM549.7 SM0.0 ... SM29.7	SM0.0 ... SM549.7 SM0.0 ... SM29.7
Timer	256 (T0 ... T255)	256 (T0 ... T255)	256 (T0 ... T255)	256 (T0 ... T255)	256 (T0 ... T255)
Retentive on-delay	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
1ms	T1 ... T4, và T65 ... T68	T1 ... T4, và T65 ... T68	T1 ... T4, và T65 ... T68	T1 ... T4, và T65 ... T68	T1 ... T4, và T65 ... T68
10ms	T5 ... T31, và T69 ... T95	T5 ... T31, và T69 ... T95	T5 ... T31, và T69 ... T95	T5 ... T31, và T69 ... T95	T5 ... T31, và T69 ... T95
100ms	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
On/Off delay 1ms	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100
10ms	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255
100ms					
Counter	C0 ... C255	C0 ... C255	C0 ... C255	C0 ... C255	C0 ... C255
Bộ đếm tốc độ cao	HC0 ... HC5	HC0 ... HC5	HC0 ... HC5	HC0 ... HC5	HC0 ... HC5
Bit điều khiển trình tự (S)	S0.0 ... S31.7	S0.0 ... S31.7	S0.0 ... S31.7	S0.0 ... S31.7	S0.0 ... S31.7
Thanh ghi Accu	AC0 ... AC3	AC0 ... AC3	AC0 ... AC3	AC0 ... AC3	AC0 ... AC3
Jumps/Labels	0 ... 255	0 ... 255	0 ... 255	0 ... 255	0 ... 255
Call/Subroutine	0 ... 63	0 ... 63	0 ... 63	0 ... 63	0 ... 127
Interrupt routines	0 ... 127	0 ... 127	0 ... 127	0 ... 127	0 ... 127
Ô nhớ sườn xung (positive/negative)	256	256	256	256	256
PID loops	0 ... 7	0 ... 7	0 ... 7	0 ... 7	0 ... 7
Port	Port 0	Port 0	Port 0	Port 0, Port 1	Port 0, Port 1

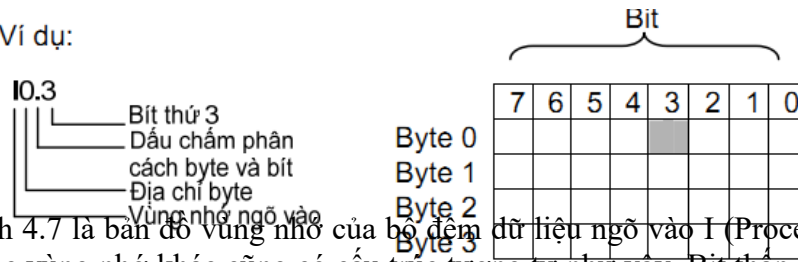
Bảng 4.3: Các vùng nhớ và đặc điểm của CPU S7-200.

4.4 Qui ước địa chỉ trong PLC S7-200

4.4.1 Truy xuất theo bit

Để truy xuất địa chỉ theo dạng Bit chúng ta xác định vùng nhớ, địa chỉ của Byte và địa chỉ của Bit.

Ví dụ:



Trong hình 4.7 là bản đồ vùng nhớ của bộ đệm dữ liệu ngõ vào I (Process Image Input). Bản đồ của các vùng nhớ khác cũng có cấu trúc tương tự như vậy. Bit thấp nhất là bit 0 nằm bên phải và bit cao nhất là bit 7 nằm bên trái. Hình 4.7: Vùng nhớ ngõ vào

4.4.2 Truy xuất theo byte (8 bit)

Khi truy xuất dữ liệu theo byte, chúng ta xác định vùng nhớ, và thứ tự của byte cần truy xuất.

Tương tự như ví dụ ta khai báo cho các vùng nhớ khác, chẳng hạn như IB3, MB2, QB5..

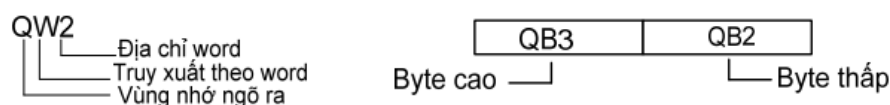
Ví dụ:



4.4.3 Truy xuất theo word (16 bit)

Đối với truy xuất vùng nhớ theo dạng word chúng ta cũng cần xác định vùng nhớ cần truy xuất, khai báo dạng word và địa chỉ của word trong vùng nhớ. Mỗi một vùng nhớ dạng word sẽ gồm 2 byte và được gọi là byte thấp và byte cao.

Ví dụ:



Chú ý:

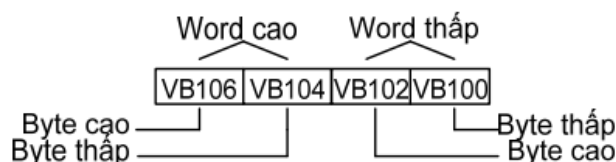
Đối với tín hiệu tương tự (Analog) thì chúng ta chỉ có một dạng truy xuất duy nhất là truy xuất theo word. Điều này là do mỗi tín hiệu tương tự sẽ ứng với một giá trị số nguyên 16 bit. Ví dụ: AIW0, AIW2, AQW0...

- Khi truy xuất địa chỉ theo word thì hai word liền kề nhau bắt buộc cách nhau 2 byte. Ví dụ ta cần chứa 2 dữ liệu dạng số integer vào vùng biến V, thì dữ liệu thứ nhất giả sử chứa vào VW20 thì word kế tiếp lưu dữ liệu thứ hai là VW22.

4.4.4 Truy xuất theo 2 word (Double word = 32 bit)

Khi truy xuất vùng nhớ 32 bit, tương ứng với 4 byte. Trong đó gồm có word thấp, word cao và byte thấp, byte cao.

Ví dụ: VD100



Bảng tóm tắt việc truy xuất các vùng nhớ theo bit, byte, word và double word được cho ở bảng 4.4.

Cách truy xuất	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Truy xuất Bit (byte.bit)					
I	0.0 ... 15.7	0.0 ... 15.7	0.0 ... 15.7	0.0 ... 15.7	0.0 ... 15.7
Q	0.0 ... 15.7	0.0 ... 15.7	0.0 ... 15.7	0.0 ... 15.7	0.0 ... 15.7
V	0.0 ... 2047.7	0.0 ... 2047.7	0.0 ... 8191.7	0.0 ... 10239.7	0.0 ... 10239.7
M	0.0 ... 31.7	0.0 ... 31.7	0.0 ... 31.7	0.0 ... 31.7	0.0 ... 31.7
SM	0.0 ... 165.7	0.0 ... 299.7	0.0 ... 549.7	0.0 ... 549.7	0.0 ... 549.7
S	0.0 ... 31.7	0.0 ... 31.7	0.0 ... 31.7	0.0 ... 31.7	0.0 ... 31.7
T	0 ... 255	0 ... 255	0 ... 255	0 ... 255	0 ... 255
C	0 ... 255	0 ... 255	0 ... 255	0 ... 255	0 ... 255
L	0.0 ... 63.7	0.0 ... 63.7	0.0 ... 63.7	0.0 ... 63.7	0.0 ... 63.7
Truy xuất Byte					
IB	0 ... 15	0 ... 15	0 ... 15	0 ... 15	0 ... 15
QB	0 ... 15	0 ... 15	0 ... 15	0 ... 15	0 ... 15
VB	0 ... 2047	0 ... 2047	0 ... 8191	0 ... 10239	0 ... 10239
MB	0 ... 31	0 ... 31	0 ... 31	0 ... 31	0 ... 31
SMB	0 ... 165	0 ... 299	0 ... 549	0 ... 549	0 ... 549
SB	0 ... 31	0 ... 31	0 ... 31	0 ... 31	0 ... 31
LB	0 ... 63	0 ... 63	0 ... 63	0 ... 63	0 ... 63
AC	0 ... 3	0 ... 3	0 ... 3	0 ... 255	0 ... 255
KB (Constant)	KB (Constant)	KB (Constant)	KB (Constant)	KB (Constant)	KB (Constant)
Truy xuất Word					
IW	0 ... 14	0 ... 14	0 ... 14	0 ... 14	0 ... 14
QW	0 ... 14	0 ... 14	0 ... 14	0 ... 14	0 ... 14
VW	0 ... 2046	0 ... 2046	0 ... 8190	0 ... 10238	0 ... 10238
MW	0 ... 30	0 ... 30	0 ... 30	0 ... 30	0 ... 30
SMW	0 ... 164	0 ... 298	0 ... 548	0 ... 548	0 ... 548
SW	0 ... 30	0 ... 30	0 ... 30	0 ... 30	0 ... 30
T	0 ... 255	0 ... 255	0 ... 255	0 ... 255	0 ... 255
C	0 ... 255	0 ... 255	0 ... 255	0 ... 255	0 ... 255
LW	0 ... 62	0 ... 62	0 ... 62	0 ... 62	0 ... 62
AC	0 ... 3	0 ... 3	0 ... 3	0 ... 3	0 ... 3
AIW	0 ... 30	0 ... 30	0 ... 62	0 ... 62	0 ... 62
AQW	0 ... 30	0 ... 30	0 ... 62	0 ... 62	0 ... 62
KW (Constant)	KW (Constant)	KW (Constant)	KW (Constant)	KW (Constant)	KW (Constant)
T. xuất Double word					
ID	0 ... 12	0 ... 12	0 ... 12	0 ... 12	0 ... 12
QD	0 ... 12	0 ... 12	0 ... 12	0 ... 12	0 ... 12
VD	0 ... 2044	0 ... 2044	0 ... 8188	0 ... 10236	0 ... 10236
MD	0 ... 28	0 ... 28	0 ... 28	0 ... 28	0 ... 28
SMD	0 ... 162	0 ... 296	0 ... 546	0 ... 546	0 ... 546
SD	0 ... 28	0 ... 28	0 ... 28	0 ... 28	0 ... 28
LD	0 ... 60	0 ... 60	0 ... 60	0 ... 60	0 ... 60
AC	0 ... 3	0 ... 3	0 ... 3	0 ... 3	0 ... 3
HC	0 ... 5	0 ... 5	0 ... 5	0 ... 5	0 ... 5
KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)

Bảng 4.4: Truy xuất các vùng nhớ theo địa chỉ bit, byte, word, double word.

Tóm lại, về cơ bản chúng ta có bốn dạng truy xuất dữ liệu như trên. Trong mỗi yêu cầu điều khiển cụ thể chúng ta sẽ chọn truy xuất theo dạng nào.

- Kiểm tra trạng thái của các tín hiệu được tạo ra từ các ngoại vi nối với ngõ vào số như nút nhấn, cảm biến, công tắc hành trình... thì sẽ chọn truy xuất là bit, trong trường hợp này thì chọn địa chỉ ngõ vào tương ứng được kết nối ví dụ như I0.0, I0.5, I1.1...

- Xuất tín hiệu ra các cơ cấu chấp hành nhận tín hiệu nhị phân như relay, đèn báo, van từ ... thì sẽ chọn truy xuất là bit, trong trường hợp này thì chọn địa chỉ ngõ ra tương ứng được kết nối ví dụ như Q0.0, Q0.2, Q1.0...

- Nhận tín hiệu từ các cảm biến tạo ra tín hiệu analog như cảm biến nhiệt độ, áp suất, độ ẩm ... thì sử dụng địa chỉ word, ví dụ: AIW0, AIW2, AIW4...

- Xuất tín hiệu analog ra các cơ cấu chấp hành nhận tín hiệu analog như ngõ vào analog biến tần, van tỉ lệ ... thì sử dụng địa chỉ word, ví dụ: AQW0, AQW2, QW4...

- Trong quá trình thực hiện chương trình cần lưu trữ thông tin ở dạng số 16 bit như đếm

số sản phẩm (số nguyên 16 bit) thì truy cập địa chỉ word, còn ở dạng 32 bit như nhiệt độ, áp suất (số thực) thì truy cập địa chỉ double word...

4.5 Xử lý chương trình

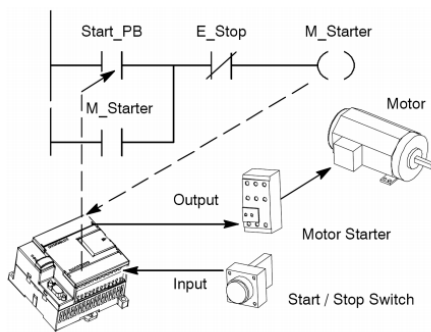
S7-200 thực hiện đọc và ghi dữ liệu theo logic điều khiển trong chương trình liên tục theo chu kỳ.

Hoạt động của S7-200 rất đơn giản:

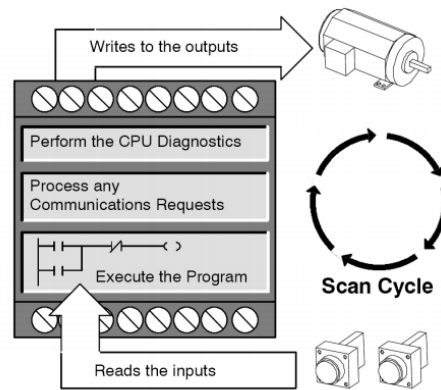
- Đọc trạng thái các ngõ vào
- S7-200 sử dụng các ngõ vào này để thực hiện logic điều khiển theo chương trình được lưu trữ trong nó. Dữ liệu luôn được cập nhật khi chương trình được thực hiện.
- Xuất dữ liệu ra ngõ ra.

Hình 4.8 là một sơ đồ đơn giản chỉ mối quan hệ giữa sơ đồ điện và PLC S7-200. Các nút nhấn khởi động/dừng động cơ được kết nối với ngõ vào. Trạng thái của các ngõ vào tùy thuộc vào nút nhấn. Các trạng thái của ngõ vào sẽ quyết định trạng thái của ngõ ra. Ngõ ra được kết nối với contactor.

Tùy theo trạng thái của ngõ ra mà contactor có điện hay mất điện và tương ứng động cơ sẽ hoạt động hay dừng.



Hình 4.8: Điều khiển ngõ vào và ra



Hình 4.9: Chu kỳ quét S7-200

* Chu kỳ quét trong S7-200

S7-200 thực hiện một loạt các nhiệm vụ theo chu kỳ. Việc thực hiện các nhiệm vụ theo chu kỳ được gọi là chu kỳ quét (scan cycle). Hình 4.9 là ví dụ một chu kỳ quét. S7-200 thực hiện các nhiệm vụ sau trong một chu kỳ quét:

- **Đọc ngõ vào:** S7-200 sao chép trạng thái của các ngõ vào vật lý vào bộ đệm ngõ vào.

Digital inputs: Mỗi chu kỳ quét bắt đầu bằng cách đọc giá trị hiện hành các ngõ vào số và sau đó ghi các giá trị này vào vùng đệm ngõ vào.

Analog inputs: S7-200 không cập nhật các ngõ vào analog từ các module mở rộng nếu là chu kỳ quét bình thường trừ khi có kích hoạt khâu lọc các ngõ vào analog (xem chương xử lý tín hiệu analog). Bộ lọc analog được cung cấp cho phép ta có một tín hiệu ổn định hơn. Có thể cho phép bộ analog ở mỗi điểm ngõ vào analog. Khi một ngõ vào analog được kích hoạt ở bộ lọc, S7-200 cập nhật ngõ vào analog mỗi một lần trong chu kỳ quét và lưu trữ giá trị lọc. Giá trị lọc được cung cấp mỗi khi truy cập ngõ vào analog. Khi bộ lọc analog không được kích hoạt, S7-200 đọc giá trị ngõ vào analog từ module mở rộng mỗi lần chương trình truy xuất ngõ vào analog.

- **Thực hiện theo logic điều khiển trong chương trình:** S7-200 thực hiện các lệnh trong chương trình và lưu giá trị vào vùng nhớ.

Khi thực hiện chu kỳ quét, S7-200 thi hành từ lệnh đầu tiên cho đến lệnh cuối cùng. Các lệnh truy cập I/O tức thì cho phép ta truy xuất ngay lập tức các ngõ vào và ngõ ra khi thực hiện chương trình cũng như chương trình ngắt (interrupt routine).

Nếu có sử dụng các ngắt trong chương trình (chương trình ngắt được gọi bởi các yêu cầu ngắt) thì nó không được thực hiện ở chu kỳ quét bình thường. Nó được thực hiện khi có sự kiện ngắt (có thể xảy ra tại bất kỳ thời điểm nào trong chu kỳ quét).

- **Xử lý bất kỳ yêu cầu truyền thông nào:** S7-200 thi hành bất kỳ nhiệm vụ được yêu cầu cho truyền thông.

Trong giai đoạn xử lý thông tin của chu kỳ quét, S7-200 xử lý bất kỳ thông tin nào nhận được từ cổng truyền thông hoặc từ các module truyền thông (intelligent I/O module).

- **Thực hiện tự chẩn đoán CPU:** S7-200 tự kiểm tra để đảm bảo phần firmware, bộ nhớ chương trình, và bất kỳ các module mở rộng nào cũng đang làm việc đúng.

Trong giai đoạn này, S7-200 kiểm tra cho hoạt động thích hợp của CPU và trạng thái của bất kỳ module mở rộng nào.

- **Xuất ra ngõ ra:** Các giá trị được lưu trong vùng đệm ngõ ra sẽ được xuất ra các ngõ ra vật lý.

Tại cuối mỗi chu kỳ, S7-200 xuất các giá trị được lưu trong bộ đệm ngõ ra đến các ngõ ra số. (Các ngõ ra analog thì được cập nhật ngay lập tức, không phụ thuộc vào chu kỳ quét).

Việc thực hiện chương trình còn tùy thuộc vào S7-200 đang ở chế độ STOP hay chế độ RUN. Ở chế độ RUN thì chương trình được thực hiện; còn ở chế độ STOP thì chương trình không được thực hiện.

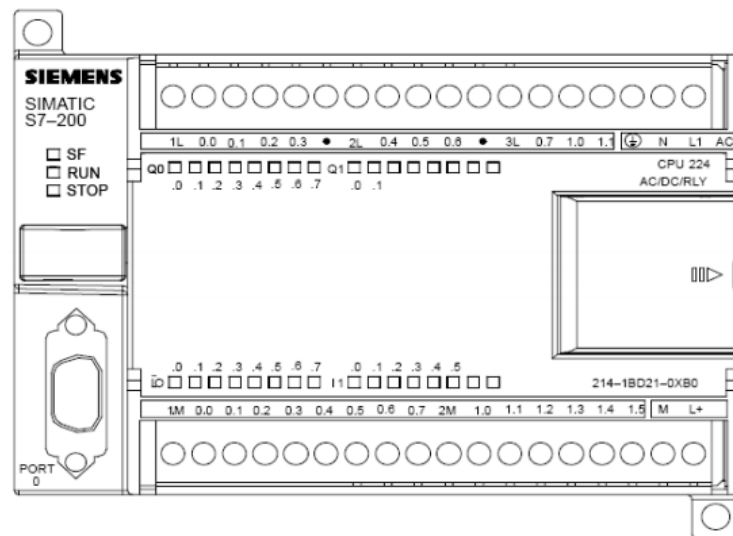
CHƯƠNG 5: KẾT NỐI ĐIỆN GIỮA PLC VÀ CÁC THIẾT BỊ NGOẠI VI

5.1 Kết nối điện giữa PLC và các thiết bị ngoại vi

Việc kết nối dây giữa PLC với ngoại vi rất quan trọng. Nó quyết định đến việc PLC có thể giao tiếp được với thiết bị lập trình (máy tính) cũng như hệ thống điều khiển có thể hoạt động đúng theo yêu cầu được thiết kế hay không. Ngoài ra việc nối dây còn liên quan đến an toàn cho PLC cũng như hệ thống điều khiển.

5.1.1 Giới thiệu CPU 224 và cách kết nối với thiết bị ngoại vi

Sơ đồ bề mặt của bộ điều khiển lập trình S7-200 CPU 224 được cho như hình 5.1.



Hình 5.1: Bộ điều khiển lập trình S7-200 CPU 224

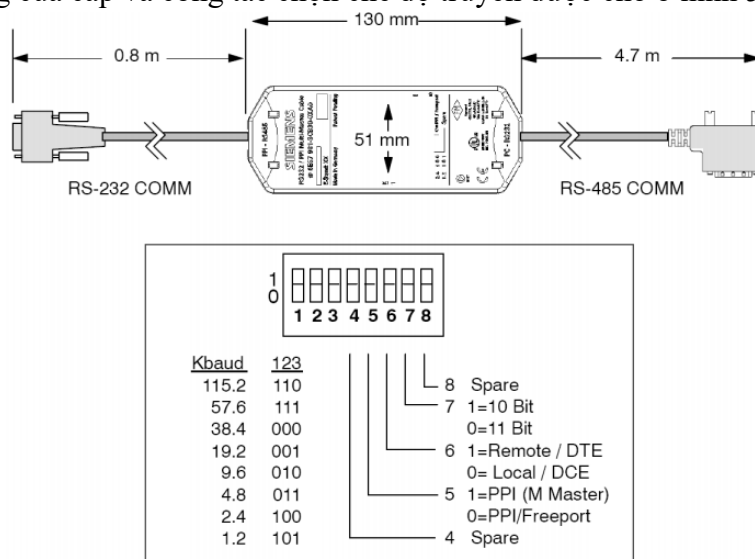
Để cho bộ điều khiển lập trình này hoạt động được thì người sử dụng phải kết nối PLC với nguồn cung cấp và các ngõ vào ra của nó với thiết bị ngoại vi. Muốn nạp chương trình vào CPU, người sử dụng phải soạn thảo chương trình bằng các thiết bị lập trình hoặc máy tính với phần mềm tương ứng cho loại PLC đang sử dụng và có thể nạp trực tiếp vào CPU hoặc copy chương trình vào card nhớ để cắm vào rãnh cắm card nhớ trên CPU của PLC. Thông thường khi lập trình cũng như khi kiểm tra hoạt động của PLC thì người lập trình thường kết nối trực tiếp thiết bị lập trình hoặc máy tính cá nhân với PLC. Như vậy, để hệ thống điều khiển khiển bằng PLC hoạt động cũng như lập trình cho nó, cần phải kết nối PLC với máy tính cũng như các ngõ vào ra với ngoại vi.

5.1.2 Kết nối với máy tính

Đối với các thiết bị lập trình của hãng Siemens có các cổng giao tiếp PPI thì có thể kết nối trực tiếp với PLC thông qua một sợi cáp. Tuy nhiên đối với máy tính cá nhân cần thiết phải có cáp chuyển đổi PC/PPI. Có 2 loại cáp chuyển đổi là cáp RS-232/PPI Multi-Master và cáp USB/PPI Multi-Master.

*** Cáp RS-232/PPI multi-master:**

Hình dáng của cáp và công tắc chọn chế độ truyền được cho ở hình 5.2.



Hình 5.2: Hình dáng cáp RS-232/PPI và các chuyển mạch trên cáp.

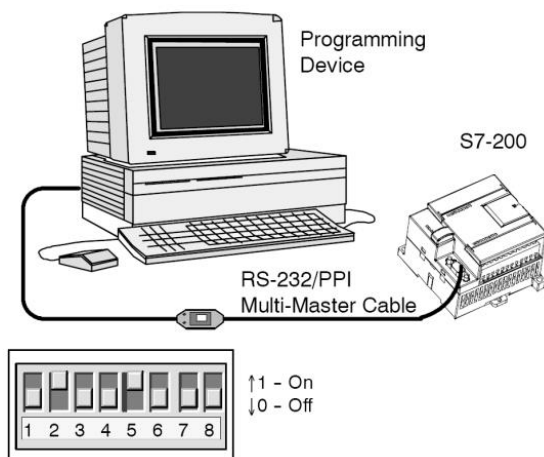
Tùy theo tốc độ truyền giữa máy tính và CPU mà các công tắc 1,2,3 được để ở vị trí thích hợp. Thông thường đối với CPU 22x thì tốc độ truyền thường đặt là 9,6 Kbaud (tức công tắc 123 được đặt theo thứ tự là 010).

Tùy theo truyền thông là 10 Bit hay 11 Bit mà công tắc 7 được đặt ở vị trí thích hợp. Khi kết nối bình thường với máy tính thì công tắc 7 chọn ở chế độ truyền thông 11 Bit (công tắc 7 đặt ở vị trí 0).

Công tắc 6 ở cáp RS-232/PPI Multi-Master được sử dụng để kết nối port truyền thông RS-232 của một modem với S7-200 CPU. Khi kết nối bình thường với máy tính thì công tắc 6 được đặt ở vị trí data Communications Equipment (DCE) (công tắc 6 ở vị trí 0). Khi kết nối cáp PC/PPI với một modem thì port RS-232 của cáp PC/PPI được đặt ở vị trí Data Terminal Equipment (DTE) (công tắc 6 ở vị trí 1).

Công tắc 5 được sử dụng để đặt cáp RS-232/PPI Multi-Master thay thế cáp PC/PPI hoặc hoạt động ở chế độ Freeport thì đặt ở chế độ PPI/Freeport (công tắc 5 ở vị trí 0). Nếu kết nối bình thường là PPI (master) với phần mềm STEP 7 Micro/Win 3.2 SP4 hoặc cao hơn thì đặt ở chế độ PPI (công tắc 5 ở vị trí 1).

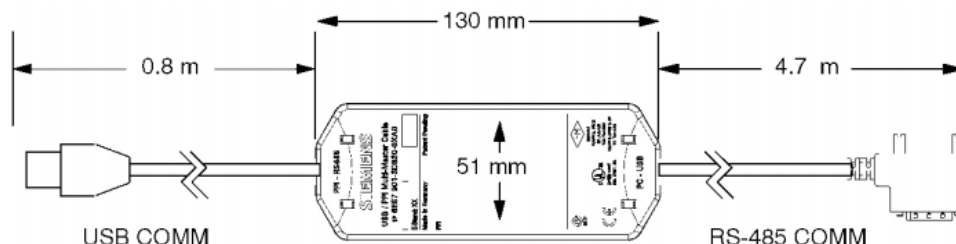
Sơ đồ nối cáp RS-232/PPI Multi-Master giữa máy tính và CPU S7-200 với tốc độ truyền 9,6 Kbaud được cho như hình 5.3.



Hình 5.3: Kết nối máy tính với CPU S7-200 RS-232/PPI Multi-Master

*** Cáp USB/PPI multi-master:**

Hình dáng của cáp được cho ở hình 5.4.



Hình 5.4: Hình dáng cáp USB/PPI.

Cách thức kết nối cáp USB/PPI Multi-Master cũng tương tự như cáp RS-232/PPI Multi-Master. Để sử dụng cáp này, phần mềm cần phải là STEP 7- Micro/WIN 3.2 Service Pack 4 (hoặc cao hơn). Cáp chỉ có thể được sử dụng với loại CPU22x hoặc sau này. Cáp USB không được hỗ trợ truyền thông Freeport và download cấu hình màn TP070 từ phần mềm TP Designer.

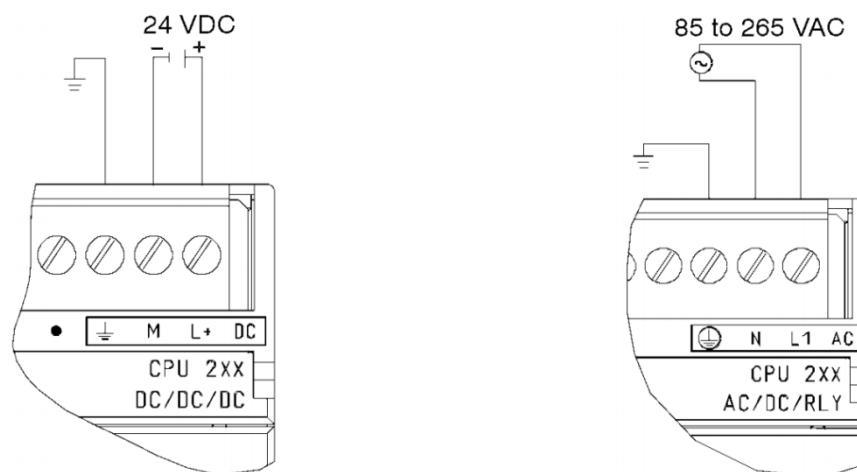
5.1.3 Nối nguồn cung cấp cho CPU

Tùy theo loại và họ PLC mà các CPU có thể là khối riêng hoặc có đặt sẵn các ngõ vào và ra cũng như một số chức năng đặc biệt khác. Hầu hết các PLC họ S7-200 được nhà sản xuất lắp đặt các khâu vào, khâu ra và CPU trong cùng một vỏ hộp. Nhưng nguồn cung cấp cho các khâu này hoàn toàn độc lập nhau. Nguồn cung cấp cho CPU của họ S7-200 có thể là:

Xoay chiều: 20...29 VAC, $f = 47...63$ Hz;

85...264 VAC, $f = 47...63$ Hz

Một chiều: 20,4 ... 28,8 VDC



a. Cấp nguồn cho CPU 2xx loại DC/DC/DC;

b. Cấp nguồn cho CPU 2xx loại AC/DC/RLY

Hình 5.5 a,b là sơ đồ nối dây nguồn cung cấp cho CPU

Để có thể nhận biết việc cấp nguồn cho CPU, khối vào, khối ra số ta căn cứ vào các chữ số đi kèm theo CPU. Các mã số kèm theo CPU 2xx có thể có như sau:

- CPU 2xx DC/DC/DC: Nguồn cấp cho CPU là DC, nguồn cho ngõ vào là DC, nguồn cấp cho ngõ ra là DC.

- CPU 2xx AC/DC/Relay: Nguồn cấp cho CPU là AC, nguồn cho ngõ vào là DC, ngõ ra là Relay có thể cấp nguồn là DC hoặc AC.

5.1.4 Kết nối vào/ra số với ngoại vi

Các ngõ vào, ra của PLC cần thiết để điều khiển và giám sát quá trình điều khiển. Các ngõ vào và ra có thể được phân thành 2 loại cơ bản: số (Digital) và tương tự (analog). Hầu hết các ứng dụng sử dụng các ngõ vào/ra số. Trong bài này chỉ đề cập đến việc kết nối các ngõ vào/ra số với ngoại vi, còn đối với ngõ vào/ra tương tự sẽ trình bày ở chương “xử lý tín hiệu analog”.

Đối với bộ điều khiển lập trình họ S7-200, hãng Siemens đã đưa ra rất nhiều loại CPU với điện áp cung cấp cho các ngõ vào ra khác nhau. Tùy thuộc từng loại CPU mà ta có thể nối dây khác nhau. Việc thực hiện nối dây cho CPU có thể tra cứu sổ tay kèm theo của hãng sản xuất.

5.1.4.1 Kết nối các ngõ vào số với ngoại vi

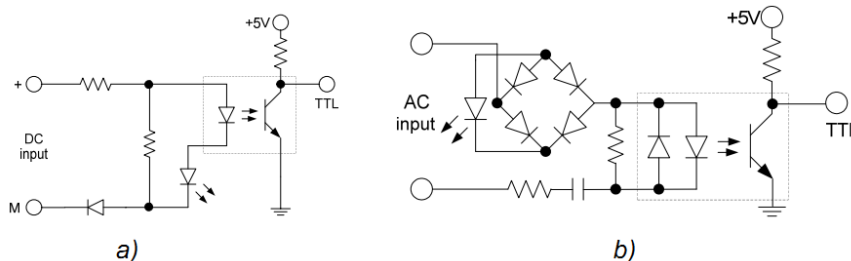
Các ngõ vào số của PLC có thể được chế tạo là một khối riêng, hoặc kết hợp với các ngõ ra chung trong một khối hoặc được tích hợp trên khối CPU. Trong trường hợp nào cũng vậy, các ngõ vào cũng phải được cung cấp nguồn riêng với cấp điện áp tùy thuộc vào loại ngõ vào. Cần lưu ý trong một khối ngõ vào cũng như các ngõ vào được tích hợp sẵn trên CPU có thể có các nhóm được cung cấp nguồn độc lập nhau. Vì vậy cần lưu ý khi cấp nguồn cho các nhóm này. Nguồn cung cấp cho các khối vào của họ S7-200 có thể là:

Xoay chiều: 15...35 VAC, $f = 47...63$ Hz; dòng cần thiết nhỏ nhất 4mA

79...135 VAC, $f = 47...63$ Hz; dòng cần thiết nhỏ nhất 4mA

Một chiều: 15 ... 30 VDC; dòng cần thiết nhỏ nhất 4mA

Sơ đồ mạch điện bên trong của một số ngõ vào được cho như hình 5.6a,b.



Hình 5.6: a) Mạch điện của 1 ngõ vào số sử dụng nguồn cung cấp DC
b) Mạch điện của 1 ngõ vào số sử dụng nguồn cung cấp AC

Tùy theo yêu cầu mà có thể quyết định sử dụng loại ngõ vào nào.

+ Ngõ vào DC:

- Điện áp DC thường thấp do đó an toàn hơn.
- Đáp ứng ngõ vào DC rất nhanh.
- Điện áp DC có thể được kết nối với nhiều phần tử trong hệ thống điện.

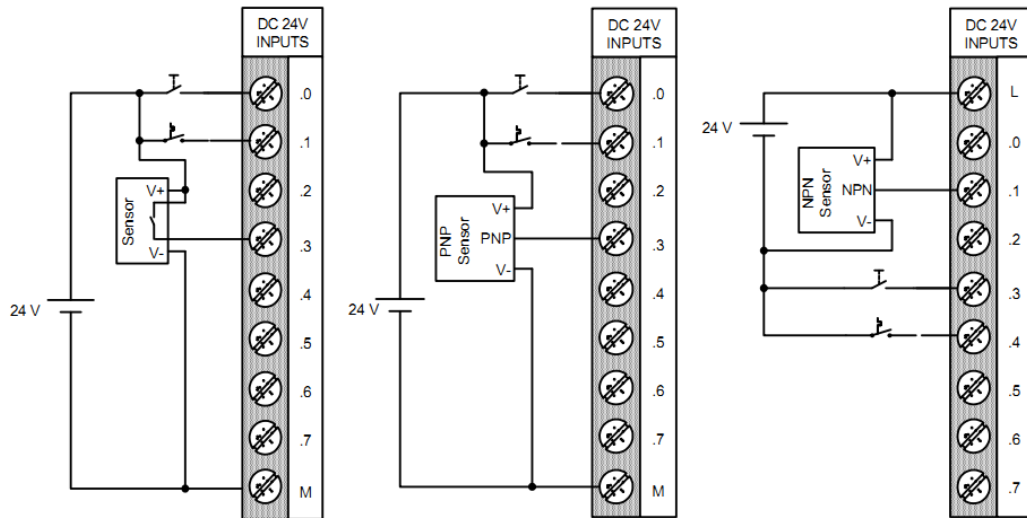
+ Ngõ vào AC:

- Ngõ vào AC yêu cầu cần phải có thời gian. Ví dụ đối với điện áp có tần số 50 Hz phải yêu cầu thời gian đến 1/50 giây mới nhận biết được.
- Tín hiệu AC ít bị nhiễu hơn tín hiệu DC, vì vậy chúng thích hợp với khoảng cách lớn và môi trường nhiễu (từ).
- Nguồn AC kinh tế hơn.
- Tín hiệu AC thường được sử dụng trong các thiết bị tự động hiện

hữu.

Đối với các ngõ vào số, khi kết nối với ngoại vi, ngoại trừ các trường hợp đặc biệt thì thông thường mỗi một ngõ vào được kết nối với một bộ tạo tín hiệu nhị phân như: nút nhấn, công tắc, cảm biến tiếp cận Hình 5.7a,b,c minh họa cách kết nối dây các ngõ vào PLC với các bộ tạo tín hiệu nhị phân khác nhau.

Trong ví dụ hình 5.7a có 3 ngõ vào, một là nút nhấn thường hở, hai là tiếp điểm của relay nhiệt, và ba là cảm biến tiếp cận với ngõ ra là relay. Cả ba bộ tạo tín hiệu này được cung cấp bởi một nguồn 24VDC. Khi tiếp điểm hở hoặc cảm biến phát tín hiệu “0” thì không có điện áp tại các ngõ vào. Nếu các tiếp điểm được đóng lại hoặc cảm biến phát tín hiệu “1” thì ngõ vào được cấp điện.



Hình 5.7: kết nối ngõ vào với ngoại vi

- Nút nhấn và cảm biến có ngõ ra là relay nối với ngõ vào loại sinking.
- Nút nhấn và cảm biến loại PNP nối với ngõ vào loại sinking.
- Nút nhấn và cảm biến loại NPN nối với ngõ vào loại sourcing.

5.1.4.2 Kết nối các ngõ ra số với ngoại vi

Các ngõ ra của PLC có thể được chế tạo là một khối riêng, hoặc kết hợp với các ngõ ra chung trong một khối hoặc được tích hợp trên khối CPU. Trong trường hợp nào cũng vậy, các ngõ ra cũng phải được cung cấp nguồn riêng với cấp điện áp tùy thuộc vào loại ngõ ra. Cần lưu ý trong một khối ra cũng như các ngõ ra được tích hợp sẵn trên CPU có thể có các nhóm được cung cấp nguồn độc lập nhau. Vì vậy cần lưu ý khi cấp nguồn cho các nhóm này. Nguồn cung cấp cho các khối ra của họ S7-200 có thể là:

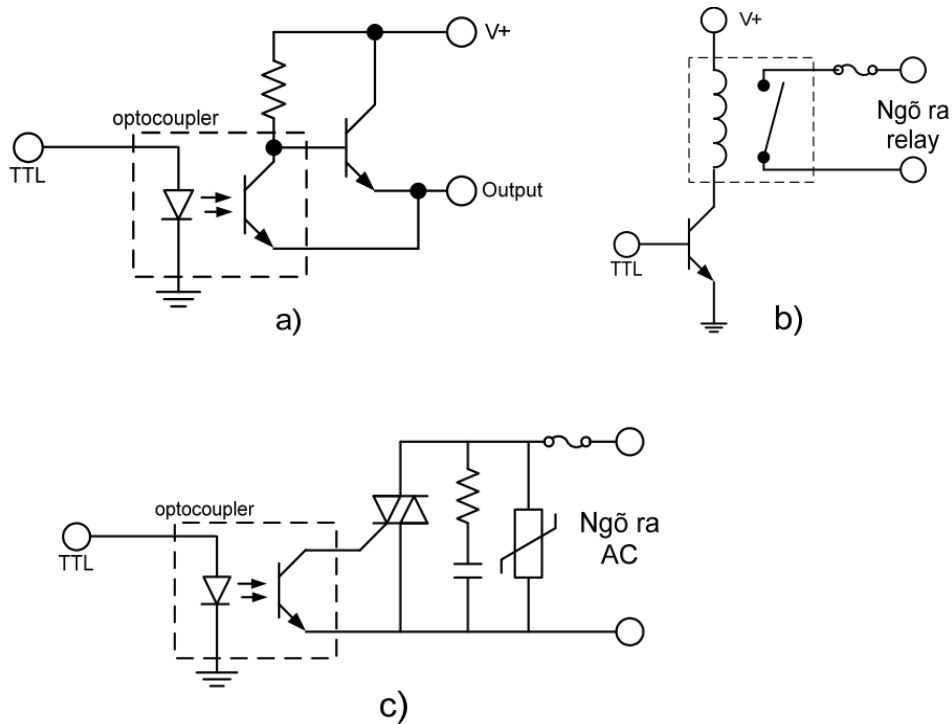
Xoay chiều: 20...264 VAC , $f = 47...63$ Hz;

Một chiều: 5...30 VDC đối với ngõ ra rơ le; 20.4 ... 28.8 VDC đối với ngõ ra transistor;

Các khối ra tiêu chuẩn của PLC thường có 8 đến 32 ngõ ra theo cùng loại và có dòng định mức khác nhau. Ngõ ra có thể là rơ le, transistor hoặc triac. Rơ le là ngõ ra linh hoạt nhất. Chúng có thể là ngõ ra AC và DC. Tuy nhiên đáp ứng của ngõ ra rơ le chậm, giá thành cao và bị hư hỏng sau vài triệu lần đóng cắt. Còn ngõ ra transistor thì chỉ sử dụng với nguồn cung cấp là DC và ngõ ra triac thì chỉ sử dụng được với nguồn AC. Tuy nhiên đáp ứng của các ngõ ra này nhanh hơn.

Sơ đồ mạch điện bên trong của các ngõ ra được cho như hình 5.8.

Cần chú ý khi thiết kế hệ thống có cả hai loại ngõ ra AC và DC. Nếu nguồn AC nối vào ngõ ra DC là transistor, thì chỉ có bán kỳ dương của chu kỳ điện áp được sử dụng và do đó điện áp ra sẽ bị giảm. Nếu nguồn DC được nối với ngõ ra AC là triac thì khi có tín hiệu cho ngõ ra, nó sẽ luôn luôn có điện cho dù có điều khiển tắt bằng PLC.



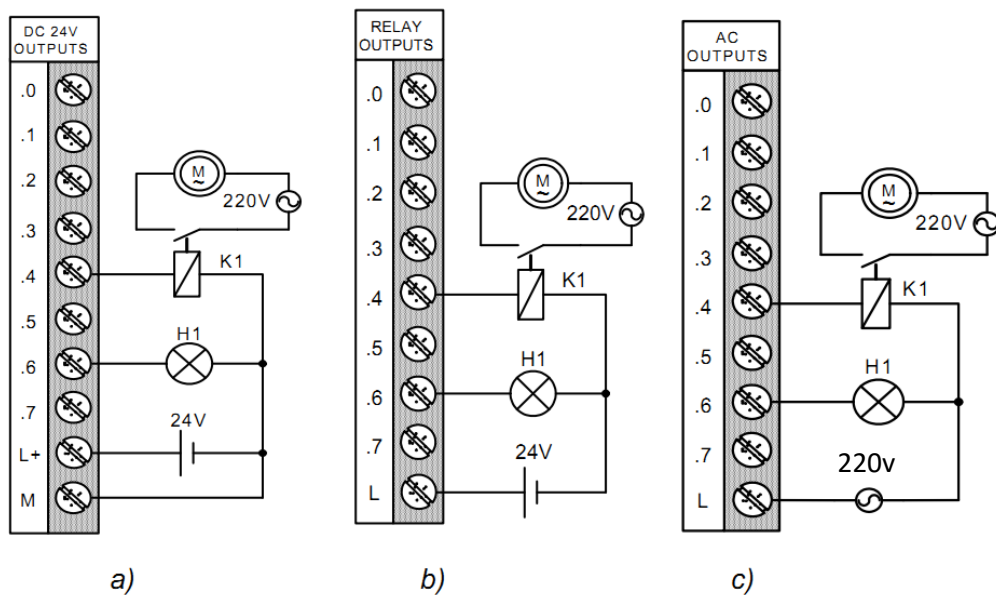
Hình 5.8: Mạch điện bên trong của các loại ngõ ra khác nhau.

a) Ngõ ra transistor ; b) Ngõ ra relay ; c) Ngõ ra triac

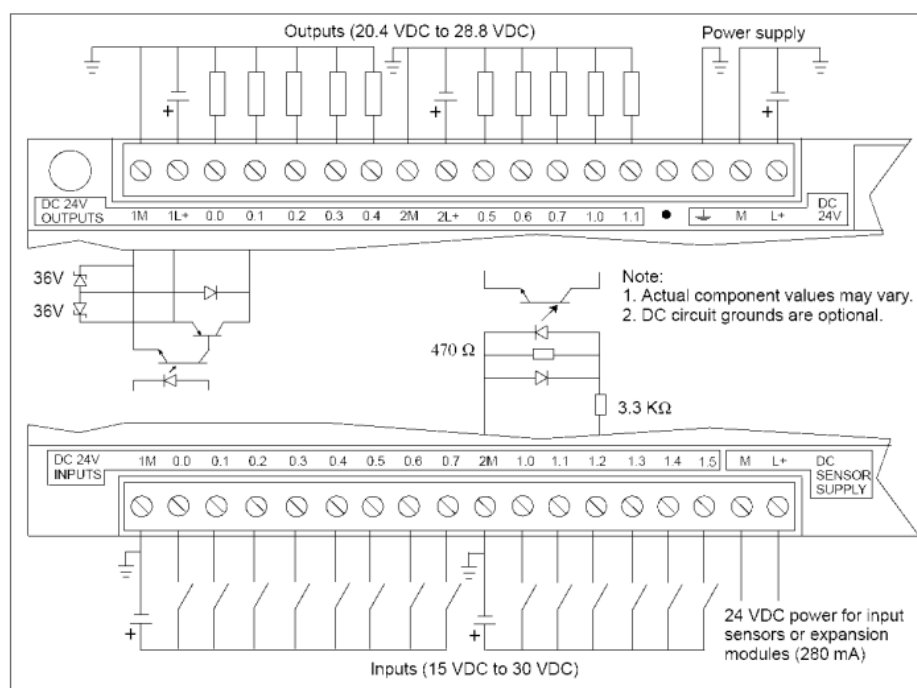
Đối với các ngõ ra số, khi kết nối với ngoại vi, ngoại trừ các trường hợp đặc biệt thì thông thường mỗi một ngõ ra được kết nối với một đối tượng điều khiển nhận tín hiệu nhị phân như: đèn báo, cuộn dây rơ le, chuông báo . . . Hình 5.9 minh họa cách kết nối dây các ngõ ra PLC với các cơ cấu chấp hành. Hình 5.9a là một ví dụ cho các khối ra sử dụng 24Vdc với mass chung. Tiêu biểu cho loại này là ngõ ra transistor. Trong ví dụ này các ngõ ra được kết nối với tải công suất nhỏ là đèn báo và cuộn dây relay. Quan sát mạch kết nối này, đèn báo sử dụng nguồn cung cấp là 24Vdc. Nếu ngõ ra .6 ở mức logic “1” (24Vdc) thì dòng sẽ chảy từ ngõ ra .6 qua đèn H1 và xuống Mass (M), đèn sáng. Nếu ngõ ra ở mức logic “0” (0V), thì đèn H1 tắt. Nếu ngõ ra .4 ở mức logic “1” thì cuộn dây rơ le có điện, làm tiếp điểm của nó đóng lại cung cấp điện 220 Vac cho động cơ.

Hình 5.9b là một ví dụ ngõ ra relay sử dụng nguồn cấp là 24 Vdc, và hình 5.9c là ví dụ ngõ ra triac sử dụng nguồn xoay chiều 220 Vac.

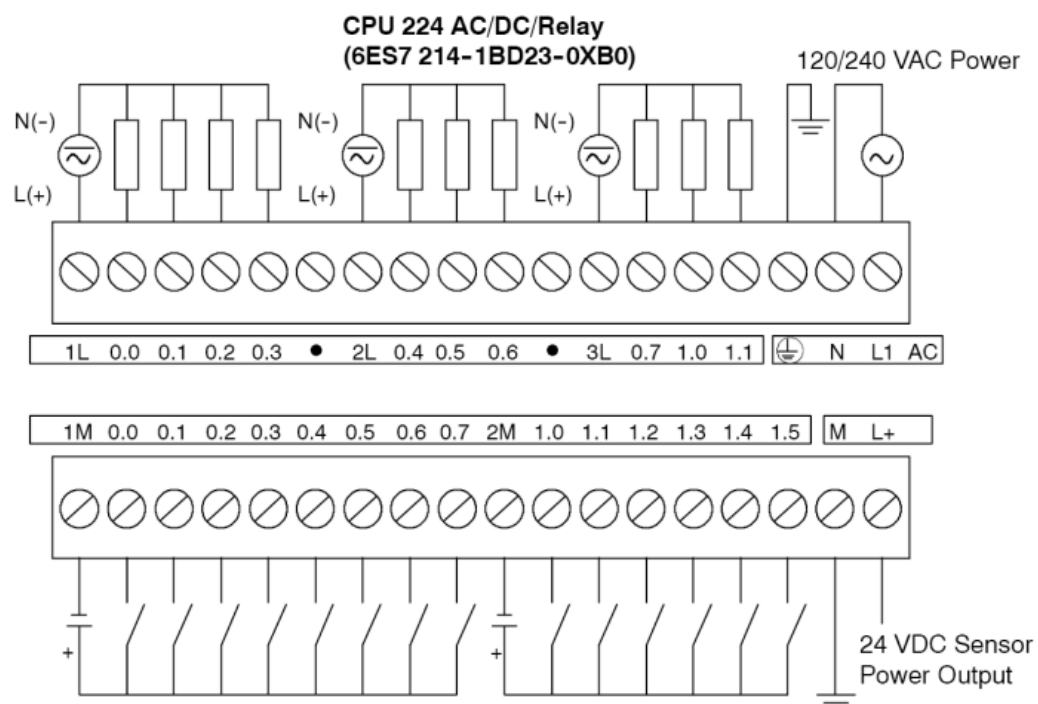
Một chú ý quan trọng khi kết nối các ngõ ra cần tra cứu sổ tay khối ngõ ra hiện có để có được thông tin chính xác tránh được những sự cố đáng tiếc xảy ra. Hình 5.10 là ví dụ của CPU 214 với nguồn cung cấp DC, ngõ vào DC và ngõ ra DC được nối dây với ngoại vi (trích từ sổ tay S7-200 Programmable Controller System Manual). Ta nhận thấy mỗi một nhóm ngõ vào cũng như một nhóm ngõ ra và CPU được cung cấp nguồn riêng là 24 Vdc. Ngoài ra trên khối CPU còn có nguồn phụ 24 Vdc (đến 280 mA) có thể được sử dụng để cung cấp cho các cảm biến hoặc khối mở rộng.



Hình 5.9: Kết nối dây ngõ ra PLC với cơ cấu chấp hành



Hình 5.10: Sơ đồ nối dây CPU 214 DC/DC/DC với nguồn và ngoại vi



Hình 5.11: Sơ đồ nối dây CPU 224 AC/DC/Relay với nguồn và ngoại vi

CHƯƠNG 6: CÁC PHÉP TOÁN LOGIC VÀ TẬP LỆNH LẬP TRÌNH

6.1 Ngăn xếp (logic stack) trong S7-200

Trong các CPU S7-200 có một ngăn xếp gồm 9 bit, chúng được sử dụng cho các câu lệnh mà dữ liệu là dạng bit. Khi viết chương trình dạng STL thì người lập trình cần hiểu rõ về phương thức hoạt động của các bit trong ngăn xếp. Ngăn xếp logic là một khối gồm 9 bit chồng lên nhau. Tất cả các thuật toán liên quan đến ngăn xếp đều chỉ làm việc với bit đầu tiên hoặc với bit đầu và bit thứ hai của ngăn xếp. Giá trị logic mới đều có thể được gửi (hoặc được nối thêm) vào ngăn xếp. Khi phối hợp hai bit đầu tiên của ngăn xếp, thì ngăn xếp sẽ được kéo lên một bit. Ngăn xếp và tên của từng bit trong ngăn xếp được biểu diễn dưới đây:


S0	Stack 0 - bit đầu tiên hay bit trên cùng của ngăn xếp.
S1	Stack 1 - bit thứ hai của ngăn xếp.
S2	Stack 2 - bit thứ ba của ngăn xếp.
S3	Stack 3 - bit thứ tư của ngăn xếp.
S4	Stack 4 - bit thứ năm của ngăn xếp.
S5	Stack 5 - bit thứ sáu của ngăn xếp.
S6	Stack 6 - bit thứ bảy của ngăn xếp.
S7	Stack 7 - bit thứ tám của ngăn xếp.
S8	Stack 8 - bit thứ chín của ngăn xếp.

Trong 9 Stack, thì Stack 0 là ngăn xếp quan trọng nhất. Giá trị logic của nó sẽ là kết quả của phép toán logic. Hay nói khác đi, sau một phép toán logic nhị phân thì kết quả của phép toán sẽ được lưu ở Stack 0. Nếu giá trị logic ở Stack 0 có giá trị là “0” thì kết quả thu được là “0”, tương tự nếu có giá trị là “1” thì kết quả thu được là “1”.

Ngoài ra giá trị logic “1” của Stack 0 còn là điều kiện bắt buộc cho việc thi hành đối với một số lệnh.

6.2 Các phép toán logic cơ bản

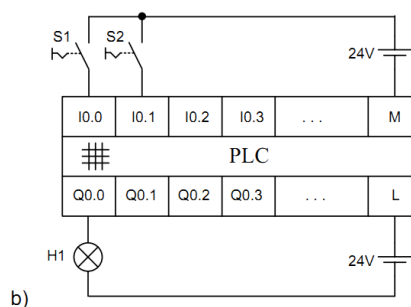
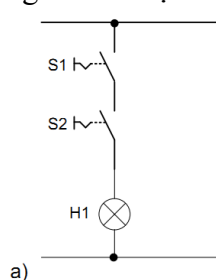
Trong phần này trình bày các phép toán đối với dữ liệu là bit. Trước tiên là phần lý thuyết sau đó tới ví dụ và chương trình. CPU sử dụng trong các ví dụ là loại DC/DC/DC (nguồn cung cấp cho ngõ vào, ra và CPU là 24Vdc).

Chương này chủ yếu trình bày về các phép toán liên quan đến bit hay còn gọi là phép toán nhị phân. Vì vậy khi viết chương trình, ta chỉ lấy các phần tử trong bit logic  Bit Logic)

6.2.1 Phép toán AND

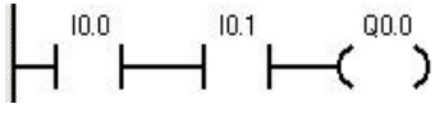
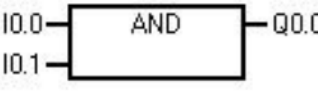
Phép toán AND được sử dụng khi có yêu cầu điều khiển là trạng thái của 2 hay nhiều tín hiệu đồng thời xảy ra thì sẽ thực hiện một nhiệm vụ điều khiển nào đó.

Ví dụ 6.1: Đèn H1 sẽ sáng nếu đồng thời cả 2 công tắc S1 và S2 ở trạng thái đóng mạch. Đèn tắt khi 1 trong 2 công tắc hở mạch.



Hình 6.1: Liên kết AND

+ Chương trình:

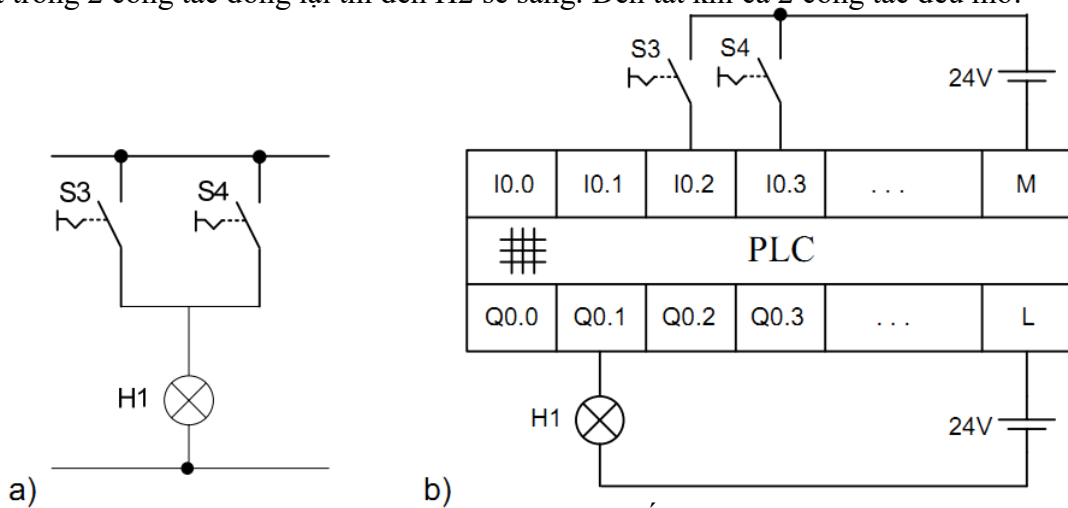
LAD	FBD	STL
		<pre>LD I0.0 A I0.1 = Q0.0</pre>

Hình 6.2: Chương trình biểu diễn dưới 3 dạng ngôn ngữ

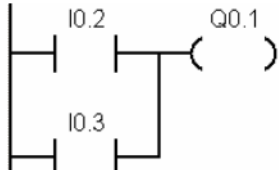
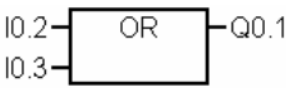
6.1.2 Phép toán OR

Phép toán OR sẽ được sử dụng khi trạng thái của một trong hai (hoặc nhiều) tín hiệu thỏa mãn điều kiện của yêu cầu điều khiển thì sẽ thực hiện một nhiệm vụ điều khiển nào đó.

Ví dụ 6.2: Có 2 công tắc S3 và S4 đều là thường hở. Hãy viết chương trình sao cho nếu một trong 2 công tắc đóng lại thì đèn H2 sẽ sáng. Đèn tắt khi cả 2 công tắc đều mở.



Hình 6.3: Liên kết OR

LAD	FBD	STL
		<pre>LD I0.2 O I0.3 = Q0.1</pre>

Hình 6.4: Chương trình

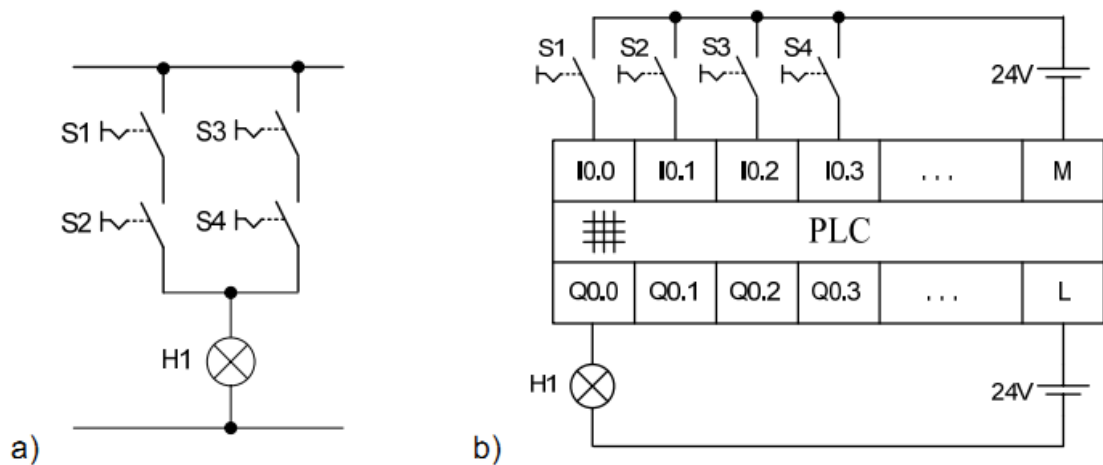
6.2.3 Tổ hợp các cổng AND và OR

Trong thực tế, các đối tượng điều khiển phụ thuộc vào một tổ hợp các liên kết logic AND và OR. Tùy theo liên kết nào đứng trước mà sẽ có các lệnh khác nhau.

6.2.3.1 AND trước OR

Để thực hiện phép OR hai liên kết AND lại với nhau thì trong chương trình viết ở dạng STL phải sử dụng thêm lệnh OLD.

Ví dụ 6.3:



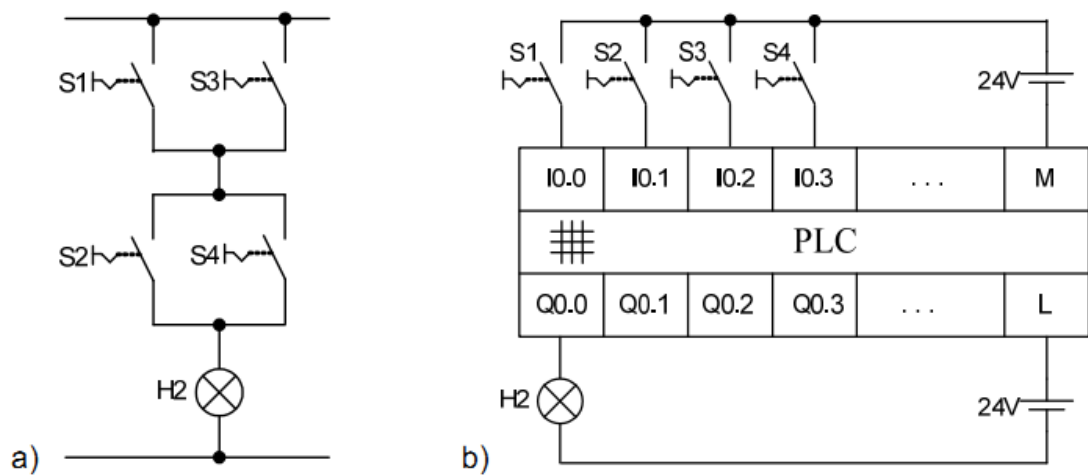
c) chương trình

LAD	FBD	STL
		<pre> LD I0.0 A I0.1 OR I0.2 A I0.3 OLD = Q0.0 </pre>

Hình 6.5: AND trước OR

6.2.3.2 OR trước AND

Để thực hiện phép AND hai liên kết OR lại với nhau thì trong chương trình viết ở dạng STL phải sử dụng thêm lệnh ALD.



c) Chương trình

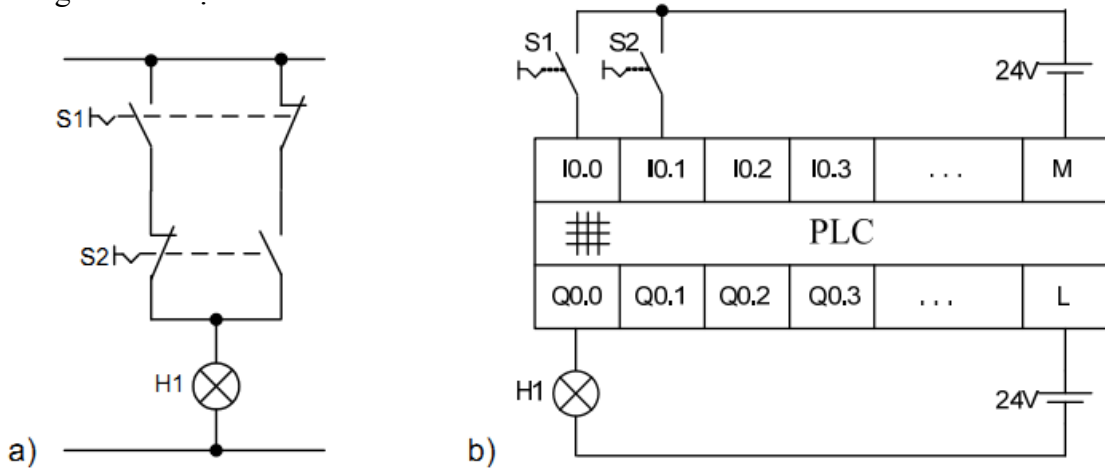
LAD	FBD	STL
		<pre> LD I0.0 O I0.2 LD I0.1 O I0.3 ALD = Q0.0 </pre>

Hình 6.6: OR trước AND

6.2.4 Phép toán XOR

Phép toán XOR được sử dụng khi có 2 tín hiệu mà nếu chúng có cùng trạng thái thì ngõ ra sẽ xuống mức 0 còn nếu 2 tín hiệu này khác trạng thái thì ngõ ra sẽ lên mức 1.

Ví dụ 6.5: Ở sơ đồ hình 6.7a, mỗi một nút nhấn được gắn 2 tiếp điểm (1NO và 1NC), khi tác động nút nhấn thì cả 2 tiếp điểm này tác động theo. Đèn sáng nếu tác động chỉ một trong hai công tắc S1 hoặc S2.



Hình 6.7: Liên kết XOR

LAD	FBD	STL
		<pre> LD I0.0 ON I0.0 LDN I0.1 O I0.1 ALD = Q0.0 </pre>

Hình 6.8: Chương trình

6.3 Xử lý các tiếp điểm, cảm biến được nối với ngõ vào PLC

Một vấn đề quan trọng đối với người mới làm quen với chương trình PLC là việc xác định đúng trạng thái các loại tiếp điểm được viết ở LAD. Đặc biệt là các tiếp điểm ngõ vào.

Các cảm biến, công tắc hoặc nút nhấn thường có hai dạng là thường đóng (NC), hoặc thường hở (NO). Vì các ngõ vào số được nối với các đối tượng này nên các tiếp điểm trong chương trình, tùy theo trường hợp, cũng sẽ có dạng tương ứng. Tuy nhiên, để dễ dàng phân biệt ta không nên gọi các tiếp điểm trong chương trình là thường đóng hoặc thường mở. Qui ước đặt tên cho các tiếp điểm trong chương trình như sau:

- Tiếp điểm : Được gọi là tiếp điểm không đảo trạng thái tín hiệu

- Tiếp điểm : Được gọi là tiếp điểm đảo trạng thái tín hiệu.

Để rõ hơn trạng thái các tiếp điểm được nối với ngõ vào số và kết quả xử lý chương trình trong PLC, ta xem bảng 6.9.

Từ bảng này, ta có một số nhận xét như sau:

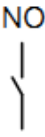



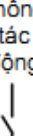
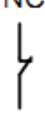

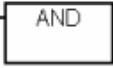
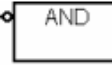
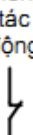
1. Ngõ vào có logic “1” khi ngõ vào có điện áp.
2. Nếu ngõ vào được nối với tiếp điểm thường đóng (NC), thì ngõ vào ở trạng thái bình thường luôn có điện (đèn LED báo ngõ vào tương ứng sáng). Nó chỉ bị mất điện nếu tiếp điểm NC bị tác động.
3. Nếu ngõ vào được nối với tiếp điểm thường hở (NO), thì ngõ vào ở trạng thái bình

thường không có điện (đèn LED báo ngõ vào tương ứng tắt). Nó chỉ có điện khi tác động tiếp điểm NO.

4. Nếu sử dụng tiếp điểm không đảo trạng thái tín hiệu $\begin{array}{|c|} \hline \text{---} \\ \hline \end{array}$, thì kết quả xử lý trong chương trình có cùng trạng thái logic với ngõ vào.

5. Nếu sử dụng tiếp điểm đảo trạng thái tín hiệu $\begin{array}{|c|} \hline \text{---} / \text{---} \\ \hline \end{array}$, thì kết quả xử lý trong chương trình có trạng thái logic ngược với ngõ vào.

6. Không được thay tùy tiện tiếp điểm thường hở (NO) bằng tiếp điểm $\begin{array}{|c|} \hline \text{---} / \text{---} \\ \hline \end{array}$ trong chương trình, cũng như tiếp điểm thường đóng (NC) bằng tiếp điểm $\begin{array}{|c|} \hline \text{---} \\ \hline \end{array}$ mà phải tùy thuộc yêu cầu công nghệ đặt ra.

Bộ tạo tín hiệu nhị phân			Thực hiện trong chương trình PLC				
Cảm biến, nút nhấn là một ...	Cảm biến, nút nhấn bị ...	Điện áp tại ngõ vào PLC	Trạng thái tín hiệu tại ngõ vào	Kiểm tra cho trạng thái tín hiệu "1"		Kiểm tra cho trạng thái tín hiệu "0"	
				Ký hiệu/lệnh	Kết quả kiểm tra	Ký hiệu/lệnh	Kết quả kiểm tra
NO 	tác động 	có	1	LAD:  "tiếp điểm không đảo"	1	LAD:  "tiếp điểm đảo"	0
	không tác động 	không	0		0		1
NC 	tác động 	không	0	FBD: 	0	FBD: 	1
	không tác động 	có	1		1		0

Hình 6.9: Trạng thái các tiếp điểm và xử lý trong chương trình

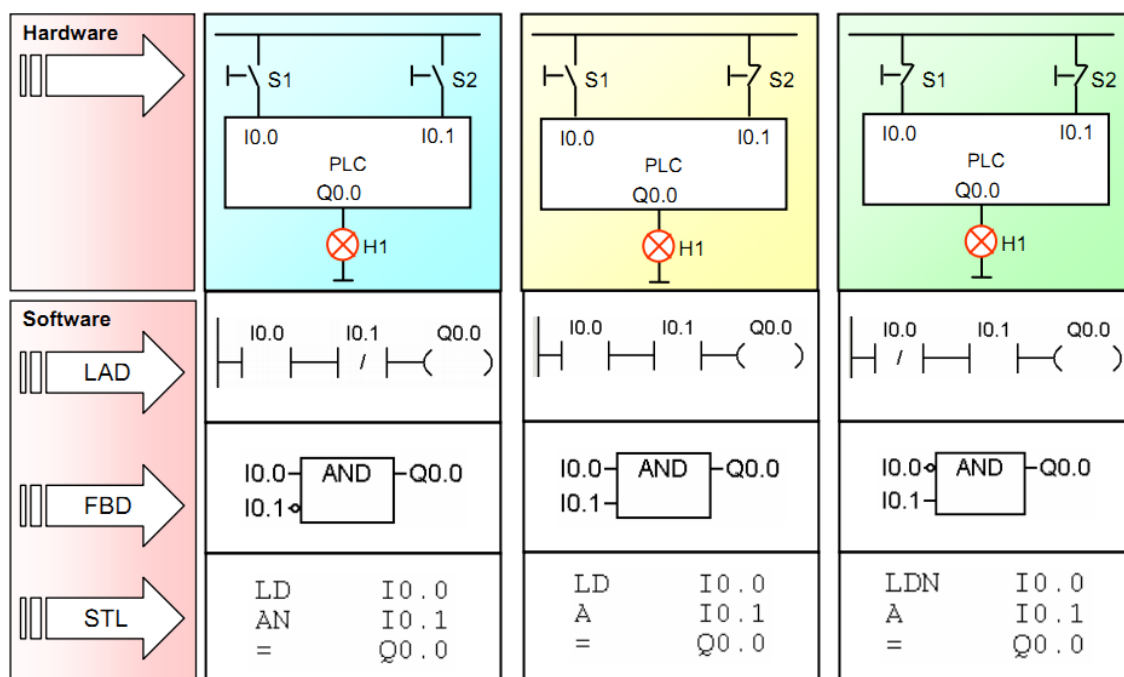
Ví dụ sau đây sẽ làm sáng tỏ hơn về việc xử lý các tiếp điểm nối với ngõ vào.

Ví dụ 6.6: Trong 3 mạch dưới đây (hình 6.10), đèn H1 sẽ sáng khi ấn nút nhấn S1 và không ấn nút nhấn S2.

Từ ví dụ ta nhận thấy dù ngõ vào được nối với loại nút nhấn nào cũng vẫn có thể lập chương trình để thỏa mãn được yêu cầu đặt ra. Tuy nhiên việc sử dụng các tiếp điểm thường mở hoặc thường đóng trong quá trình điều khiển phụ thuộc vào các qui tắc an toàn.

Các tiếp điểm thường đóng luôn luôn được sử dụng cho công tắc hành trình và công tắc an toàn, để không chế sự nguy hiểm nếu dây điện bị đứt trong mạch điện cảm biến.

Các tiếp điểm thường đóng cũng được dùng để tắt máy vì lý do tương tự như trên.



Hình 6.10: Ví dụ xử lý các loại tiếp điểm.

6.4 Ví dụ ứng dụng các liên kết logic

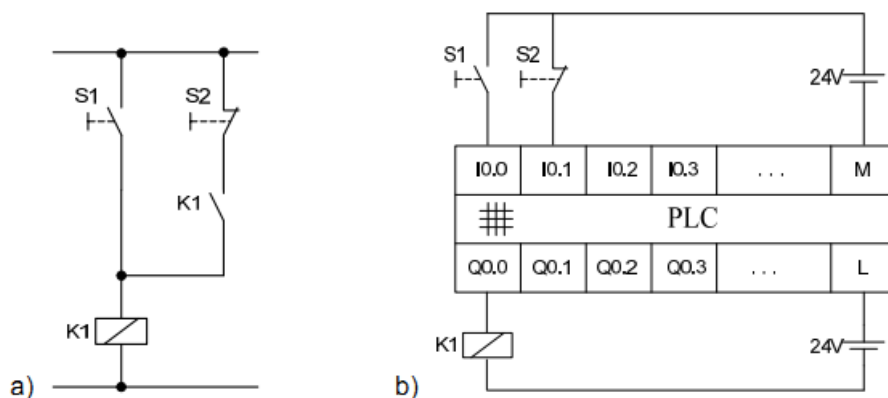
Phần này trình bày một số ví dụ ứng dụng nhỏ sử dụng các liên kết logic. Ở một số ví dụ có trình bày mạch điều khiển thông thường với kiểu nối dây khi không dùng PLC để chúng ta thấy sự giống nhau và khác nhau giữa 2 kiểu điều khiển.

6.4.1 Mạch tự duy trì ưu tiên mở máy

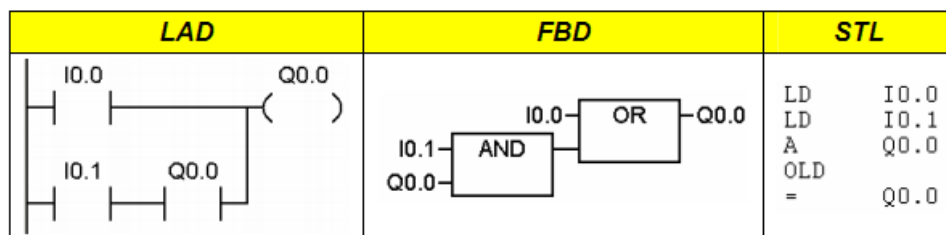
Mạch điều khiển dùng contactor có chức năng nhớ là mạch tự duy trì.

Trong trường hợp nếu cả hai nút nhấn mở máy S1 và dừng S2 cùng tác động mà contactor có điện thì là mạch tự duy trì ưu tiên mở máy.

Bảng ký hiệu		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn mở máy, thường hở (NO)
S2	I0.1	Nút nhấn dừng máy, thường đóng (NC)
K1	Q0.0	Contactor



Hình 7.11 Mạch duy trì ưu tiên mở máy

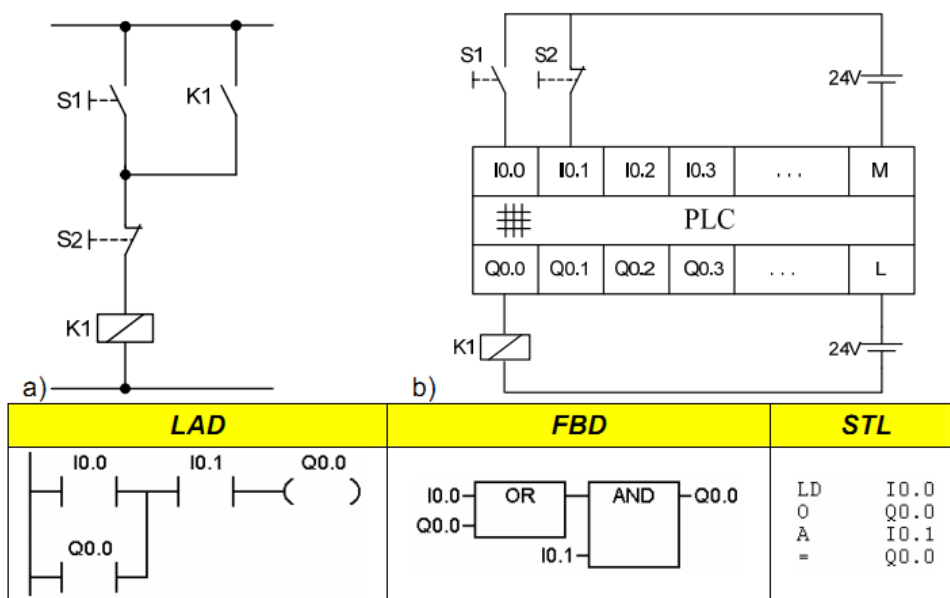


Hình 7.12: Chương mạch duy trì ưu tiên mở máy

6.4.2 Mạch tự duy trì ưu tiên dừng máy

Trong trường hợp nếu cả hai nút nhấn mở máy S1 và dừng S2 cùng tác động mà contactor không có điện thì là mạch tự duy trì ưu tiên dừng máy

Bảng ký hiệu		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn mở máy, thường hở (NO)
S2	I0.1	Nút nhấn dừng máy, thường đóng (NC)
K1	Q0.0	Contactor



Hình 7.13: Mạch duy trì ưu tiên dừng máy

6.4.3 Điều khiển ON/OFF động cơ có chỉ báo

Một động cơ điện 3 pha được điều khiển bằng một PLC S7-200. Khi nhấn nút S2 (thường hở) thì động cơ sẽ chạy. Khi nhấn nút S1 (thường đóng) thì động cơ sẽ dừng lại. Các chế độ hoạt động chạy và dừng được báo bằng 2 đèn báo H1 và H2.

Các thiết bị động lực gồm có:

- Cầu chì 3 pha F1
- CB bảo vệ động cơ (Motor CB) Q1
- Contactor K1

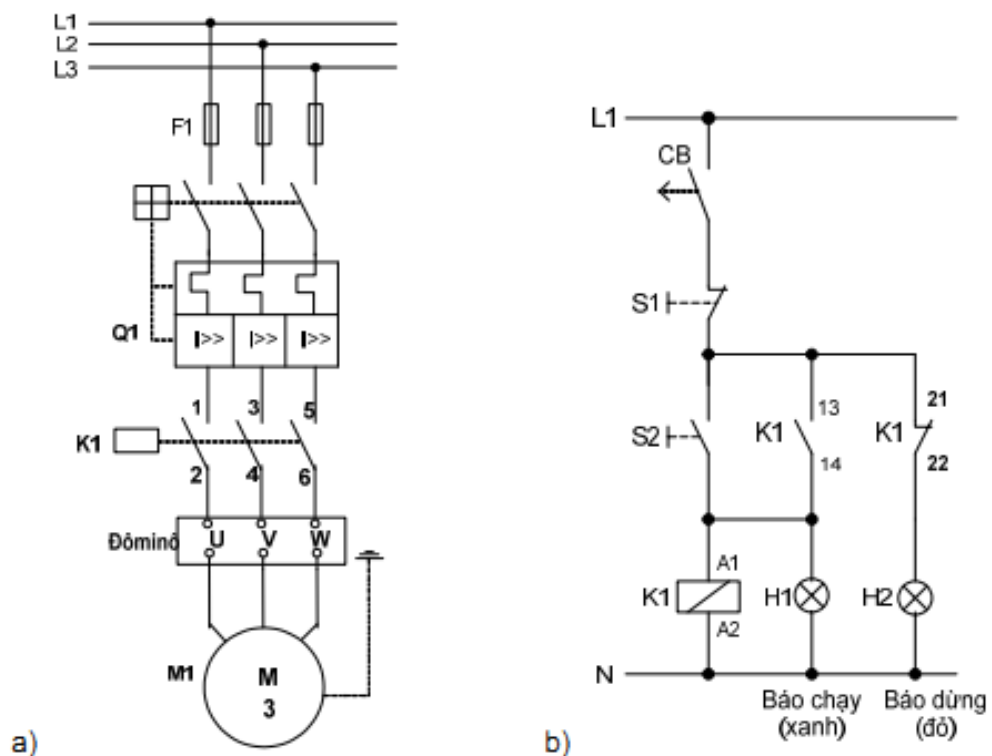
Khi điều khiển dùng PLC thì mạch động lực vẫn giữ nguyên. Phần mạch điều khiển được biến đổi thành chương trình. Cần chú ý rằng các thiết bị điện như nút nhấn, CB, đèn báo đều giữ nguyên không thay đổi.

Nếu ta sử dụng PLC S7-200 loại DC/DC/DC thì ngõ ra của PLC cần phải kết nối với một

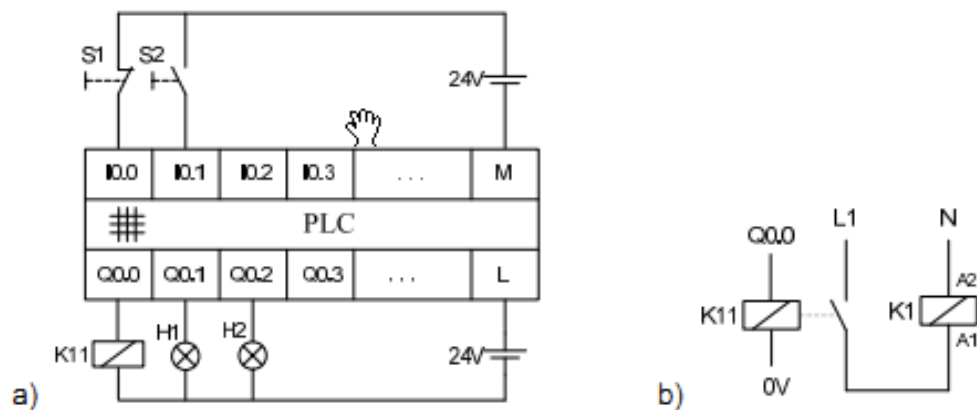
relay trung gian K11 sử dụng nguồn 24Vdc. Relay này được dùng để đóng điện cho cuộn dây contactor K1 (hình 6.15). Riêng các đèn báo ta có thể thay thế bằng loại 24Vdc nhằm tiết kiệm relay trung gian.

Chú ý: Cũng có thể sử dụng loại CPU DC/DC/RLY, thì ngõ ra của nó có thể kết nối trực tiếp với cuộn dây K1. (xem thêm chương 5 về nối dây PLC với ngoại vi).

Bảng ký hiệu		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn dừng máy, thường đóng (NC)
S2	I0.1	Nút nhấn mở máy, thường hở (NO)
K11	Q0.0	Relay trung gian
H1	Q0.1	Đèn báo động cơ hoạt động
H2	Q0.2	Đèn báo động cơ dừng



Hình 6.14: Mạch ON/OFF động cơ dùng contactor



Hình 6.15: Mạch kết nối PLC

+ Chương trình PLC:

Network 1 ON/OFF động cơ

```

LD    I0.0
LD    I0.1
O      Q0.0
ALD
=      Q0.0

```

Network 2 Bảo động cơ hoạt động

```

LD    Q0.0
=      Q0.1

```

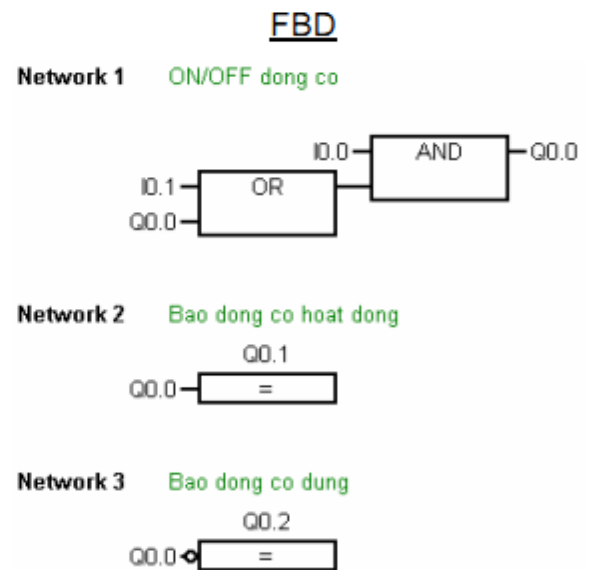
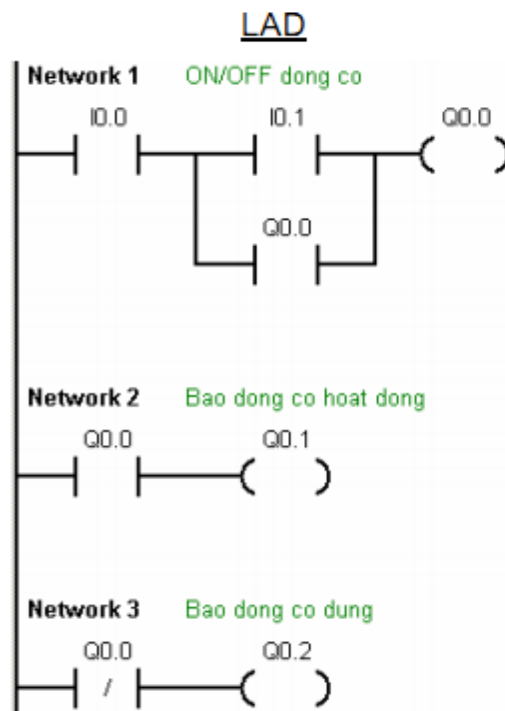
Network 3 Bảo động cơ dừng

```

LDN   Q0.0
=      Q0.2

```

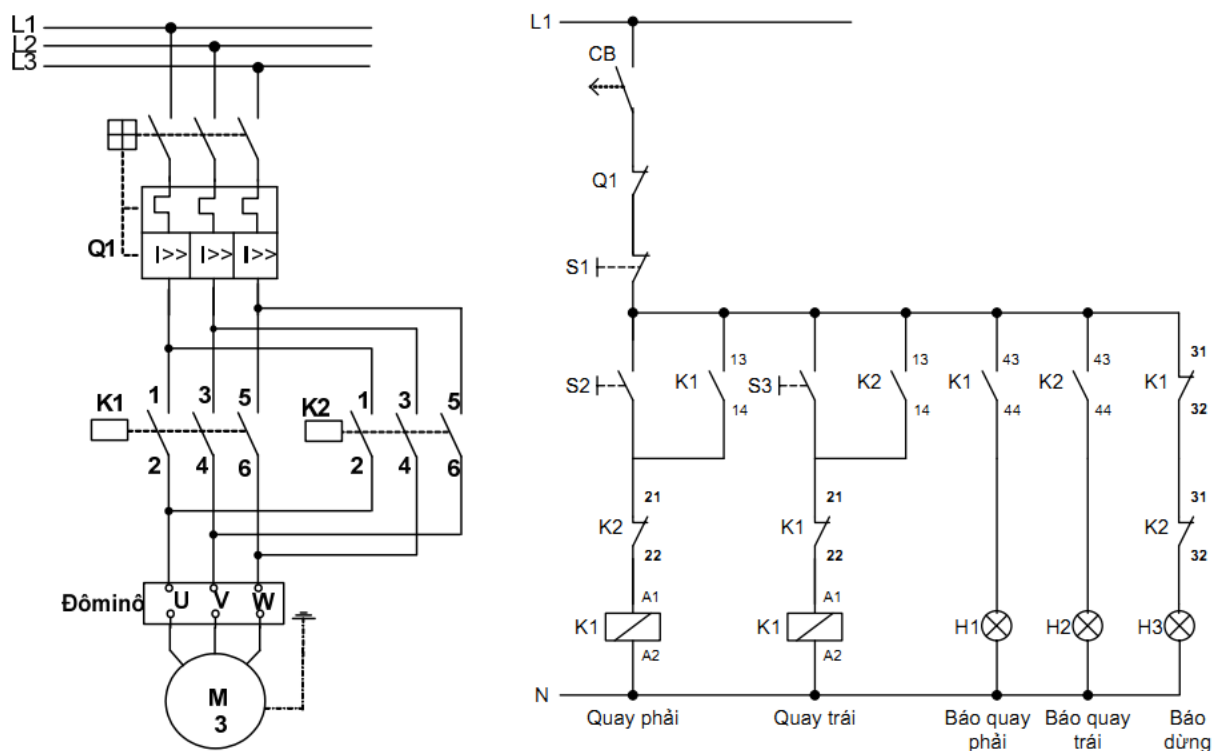
Biểu diễn ở LAD và FBD:



6.4.4 Điều khiển đảo chiều quay động cơ

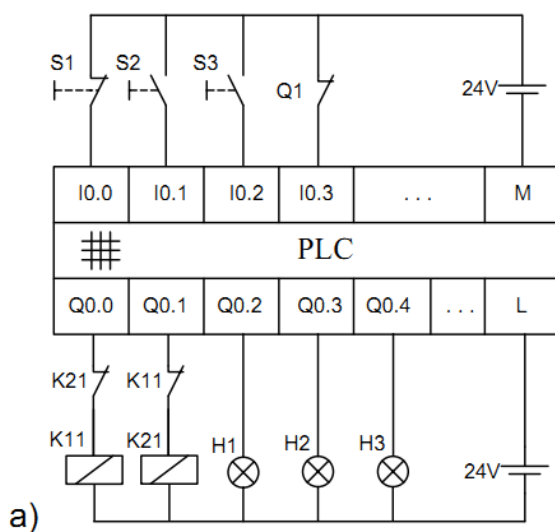
Một động cơ điện 3 pha cần được điều khiển đảo chiều. Khi ấn S1 (thường hở) thì động cơ sẽ quay phải và đèn H1 sáng báo động cơ đang quay phải. Khi nhấn nút S2 (thường hở) thì động cơ quay trái và đèn H2 sáng báo động cơ đang quay trái. Động cơ có thể dừng bất cứ lúc nào nếu ấn nút dừng S3 (thường đóng) hoặc động cơ xảy ra sự cố quá dòng làm cho tiếp điểm của thiết bị bảo vệ Q1 tác động (tiếp điểm 13, 14 của Motor CB). Khi động cơ dừng đèn báo H3 sáng.

Tương tự như mục 6.4.3, ta sử dụng PLC S7-200 loại DC/DC/DC, ngõ ra của PLC điều khiển quay phải kết nối với relay trung gian K11, ngõ ra của PLC điều khiển quay trái kết nối với relay trung gian K21 sử dụng nguồn 24Vdc. Các relay này được dùng để đóng điện cho cuộn dây contactor K1 và K2 (hình 7.16). Riêng các đèn báo ta có thể thay thế bằng loại 24Vdc nhằm tiết kiệm relay trung gian.

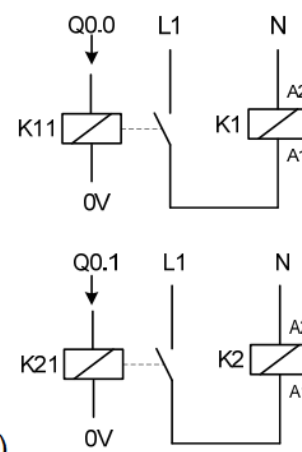


Hình 6.16: Mạch động lực và điều khiển đảo chiều quay động cơ

Bảng ký hiệu		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn dừng máy, thường đóng (NC)
S2	I0.1	Nút nhấn quay phải, thường hở (NO)
S3	I0.2	Nút nhấn quay trái, thường hở (NO)
Q1	I0.3	Tiếp điểm bảo quá dòng, thường đóng (NC)
K11	Q0.0	Relay trung gian điều khiển quay phải
K21	Q0.1	Relay trung gian điều khiển quay trái
H1	Q0.2	Đèn báo động cơ quay phải
H2	Q0.3	Đèn báo động cơ quay trái
H3	Q0.4	Đèn báo động cơ dừng



a)



b)

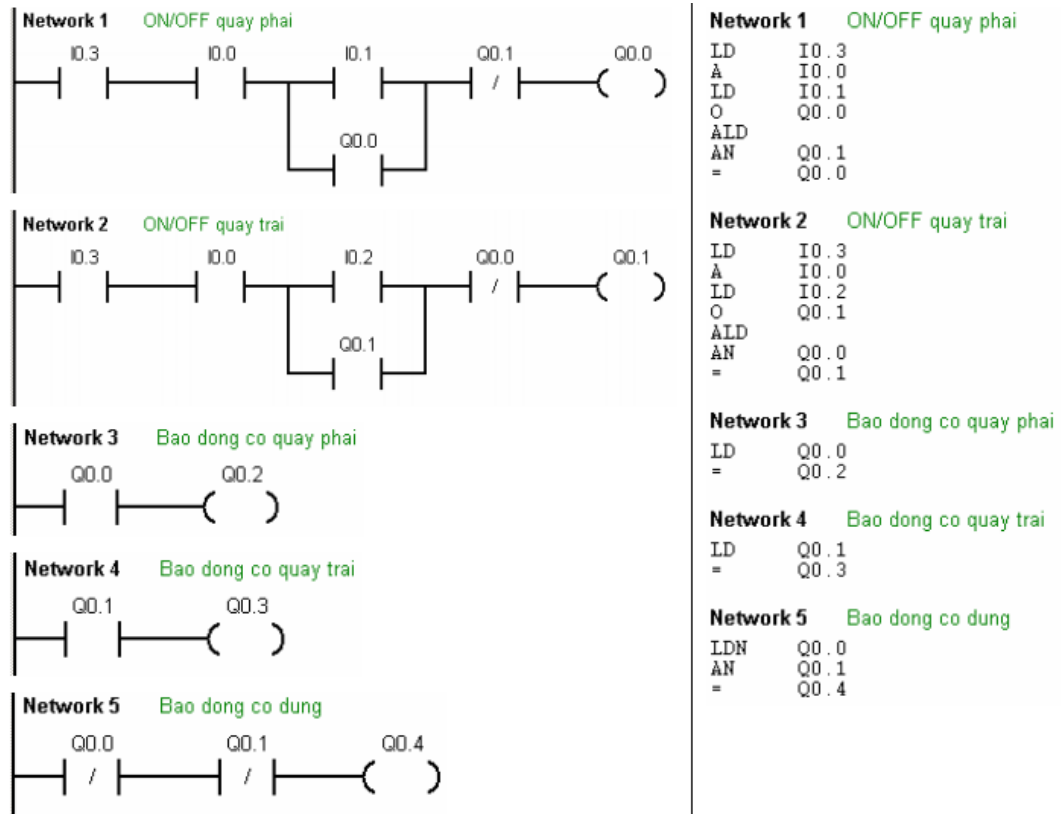
Hình 6.17 a) Sơ đồ đấu dây PLC

b) Nối relay với contacter

Chú ý: Trong các điều khiển có đảo chiều quay thì tại các ngõ ra PLC điều khiển 2 chiều quay của động cơ ta cần phải nối thêm 2 tiếp điểm thường đóng khóa chéo nhau của 2 contactor (hoặc relay) để đảm bảo an toàn.

Chương trình PLC:

Biểu diễn ở LAD và STL:



6.5 Các lệnh SET, RESET và mạch nhớ RS

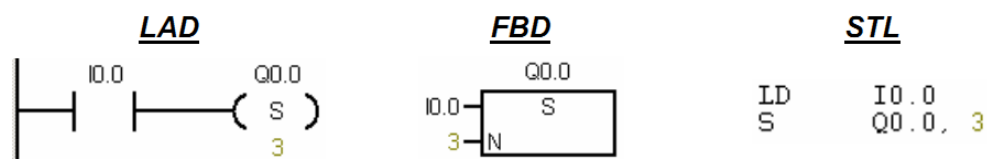
6.5.1 Lệnh SET

Lệnh SET (S) là lệnh thông dụng rất thường được sử dụng và lệnh này đều có trong hầu hết các PLC. Lệnh Set sẽ đặt trạng thái của một hoặc nhiều bit (thuộc vùng nhớ V, M, Q, T, C, SM, L) có địa chỉ liên tục lên mức 1 và duy trì ở trạng thái này cho đến khi bị xóa bằng một lệnh khác. Chúng ta có thể Set một lần tối đa tới 255 bit. Lệnh SET chỉ được thực hiện khi Stack 0 có giá trị logic “1”.

Cú pháp ở STL: S S_Bit, n và ở LAD: $\overline{S}_{n}^{S_Bit}$

Với S_Bit là bit đầu tiên của vùng nhớ cần đặt lên mức logic “1”. và n là số lượng bit bắt đầu từ S_Bit.

Ví dụ: Khi tín hiệu tại I0.0 lên mức 1 thì sẽ set 3 bit từ Q0.0 đến Q0.2. Chương trình ở 3 dạng như sau:



Khi tín hiệu tại I0.0 xuống mức 0 thì 3 ngõ ra Q0.0, Q0.1, Q0.2 vẫn duy trì ở mức 1.

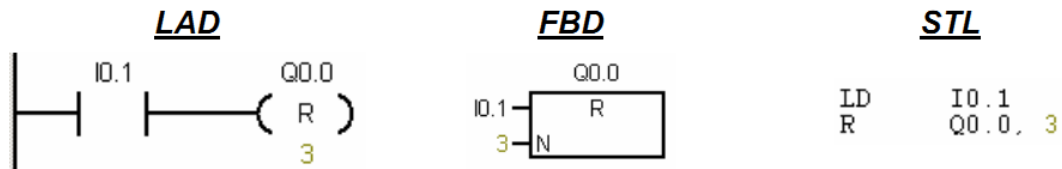
6.5.2 Lệnh RESET (R)

Lệnh Reset (R) đặt trạng thái của một hoặc nhiều bit có địa chỉ liên tục xuống mức 0. Tương tự như lệnh Set chúng ta có thể Reset tới 255 bit nhớ thuộc các vùng nhớ V, M, Q, T, C, SM, L. Lệnh RESET chỉ được thực hiện khi Stack 0 có giá trị logic “1”.

Cú pháp ở STL: R S_Bit, n và ở LAD: 

Với S_Bit là bit đầu tiên của vùng nhớ cần đặt xuống mức logic “0”. và n là số lượng bit bắt đầu từ S_Bit.

Ví dụ: Khi tín hiệu tại I0.1 lên mức 1 thì sẽ reset 3 bit từ Q0.0 đến Q0.2 về logic “0”. Chương trình ở 3 dạng như sau:



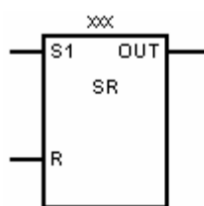
6.5.3 Mạch nhớ R-S

Mạch nhớ là mạch có hai trạng thái ổn định và thông qua tín hiệu ngõ vào mà trạng thái của nó thay đổi. Đối với mạch điều khiển dùng relay và contactor ta có mạch tự duy trì. Còn trong PLC có khâu R-S (viết tắt của Reset và Set).

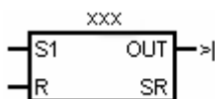
Mạch nhớ R-S là rất cần thiết trong kỹ thuật điều khiển. Nó được xem là một chức năng cơ bản trong hầu hết các loại PLC và được chia thành hai loại là: Ưu tiên SET và ưu tiên RESET.

6.5.3.1 Ưu tiên SET (khâu SR)

Biểu diễn ở LAD:



và FBD:



Với:

xxx: Địa chỉ cần điều khiển

S1: Ngõ vào Set. Ký hiệu ưu tiên Set.

R: Ngõ vào Reset.


OUT: Ngõ ra, có thể nối với một địa chỉ dạng bit

SR: Ký hiệu gọi nhớ khâu SR

Nếu cả hai điều kiện cho S và R lên mức logic “1” thì ngõ ra OUT là “1”.

Bảng sự thật

S1	R	OUT
0	0	Trạng thái trước
0	1	0
1	0	1
1	1	1

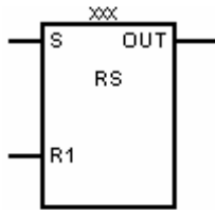
Để lấy khâu SR, ta nhấp chuột vào dấu cộng của  Bit Logic trong cây lệnh, chọn phần tử SR và kéo thả vào network mong muốn.

Khâu SR tương đương với mạch tự duy trì ưu tiên mở máy trong điều khiển dùng

contactor.

6.5.3.2 Ưu tiên RESET (khâu RS)

Biểu diễn ở LAD:



Với:

xxx: Địa chỉ cần điều khiển

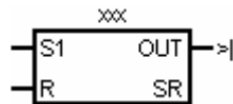
S: Ngõ vào Set.

R1: Ngõ vào Reset. Ký hiệu ưu tiên ReSet.

OUT: Ngõ ra, có thể nối với một địa chỉ dạng bit


RS: Ký hiệu gọi nhớ khâu RS

và FBD:



Bảng sự thật

S1	R	OUT
0	0	Trạng thái trước
0	1	0
1	0	1
1	1	0

Để lấy khâu RS, ta nhấp chuột vào dấu cộng của  Bit Logic trong cây lệnh, chọn phần tử RS và kéo thả vào network mong muốn.

Khâu RS tương đương với mạch tự duy trì ưu tiên dừng máy trong điều khiển dùng contactor.

6.5.4 Các qui tắc khi sử dụng Set và Reset

Khi sử dụng với các lệnh S và R trong chương trình PLC cần chú ý các qui tắc sau:

- Các điều kiện làm cho đối tượng điều khiển ở mức tích cực (logic “1”) được sử dụng với lệnh S.
- Các điều kiện làm cho đối tượng điều khiển ở mức không tích cực (logic “0”) được sử dụng với lệnh R.
- Khi viết lệnh S cho một đối tượng điều khiển thì nhất thiết (tùy theo yêu cầu công nghệ) phải có một lệnh R cho đối tượng điều khiển đó.
- Nếu lệnh S được viết trước lệnh R thì kết quả thu được sẽ là kết quả của lệnh R nếu cả hai điều kiện cho S và R cùng ở mức logic “1” nghĩa là đối tượng điều khiển ở mức logic “0”.
- Nếu lệnh R được viết trước lệnh S thì kết quả thu được sẽ là kết quả của lệnh S nếu cả hai điều kiện cho S và R cùng ở mức logic “1” nghĩa là đối tượng điều khiển ở mức logic “1”.
- Khi đã viết chương trình với lệnh S thì không được sử dụng tiếp điểm tự duy trì (loại bỏ tiếp điểm tự duy trì).
- Tùy theo công nghệ khi sử dụng các điều kiện cho lệnh R thì ở trạng thái bình thường các điều kiện này phải có mức logic “0”.

6.6 Các lệnh nhận biết cạnh tín hiệu và lệnh NOT

Các lệnh nhận biết cạnh tín hiệu và lệnh NOT thực hiện các thuật toán đặc biệt trên bit đầu tiên của ngăn xếp (Stack 0).

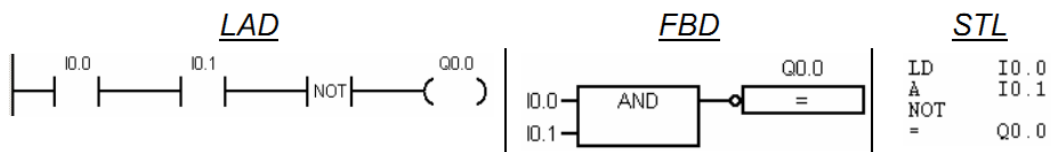
6.6.1 Lệnh NOT

Lệnh NOT đảo giá trị của bit đầu tiên trong ngăn xếp (Stack 0). Nếu sau một phép toán nhị phân mà sử dụng lệnh NOT thì kết quả sẽ bị đảo lại. Nghĩa là nếu kết quả phép toán nhị phân làm cho Stack 0 có giá trị logic “1” thì lệnh NOT sẽ cho kết quả là “0”, và ngược lại.

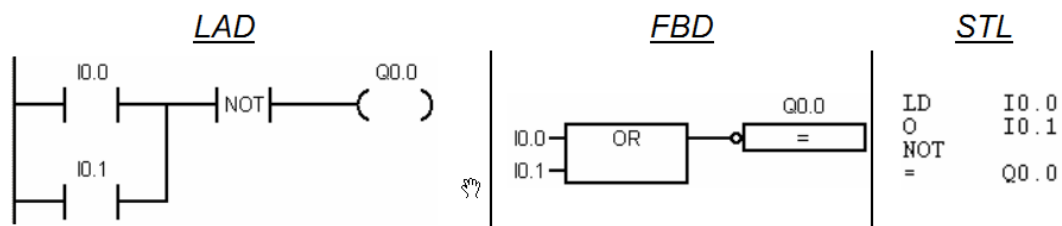
- Kết hợp lệnh NOT sau các cổng logic như OR, AND, XOR ta thu được các cổng NOR, NAND, XNOR.

Ví dụ:

- Cổng NAND với 2 ngõ vào I0.0 và I0.1 và ngõ ra Q0.0 là:



- Cổng NOR với 2 ngõ vào I0.0 và I0.1 và ngõ ra Q0.0 là:



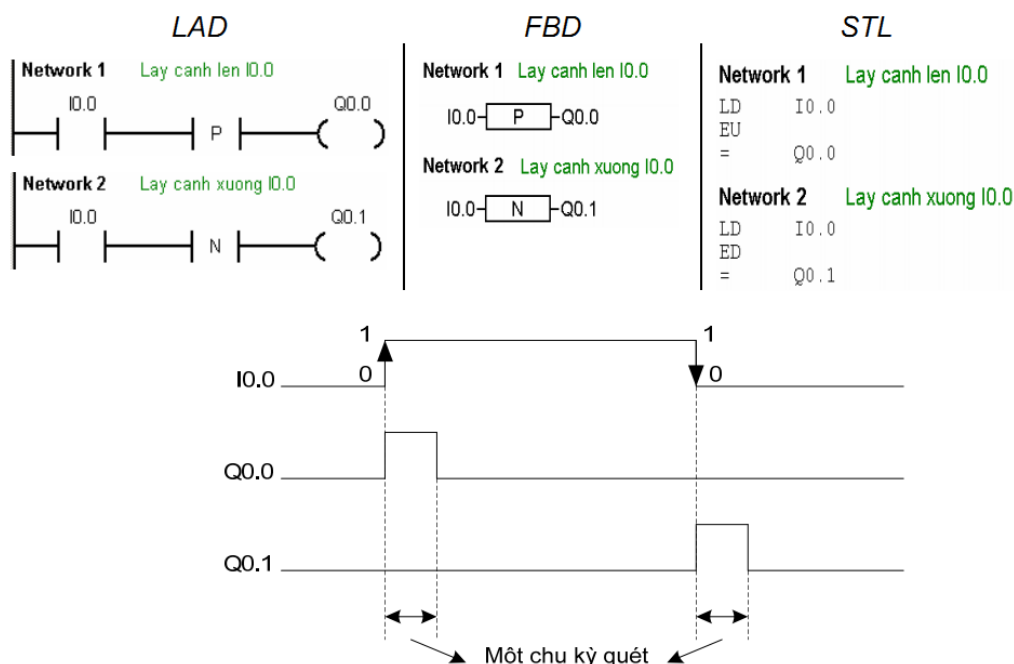
6.6.2 Các lệnh nhận biết cạnh tín hiệu

Hai lệnh nhận biết cạnh tín hiệu là lệnh nhận biết cạnh lên (EU) và nhận biết cạnh xuống (ED).

Lệnh nhận biết cạnh lên (EU) sẽ đặt giá trị logic “1” vào bit đầu tiên của Stack 0 trong một chu kỳ quét chương trình khi phát hiện sự chuyển trạng thái từ 0 lên 1 trong Stack 0. Còn các trường hợp khác nó sẽ đặt Stack 0 về “0”.

Lệnh nhận biết cạnh xuống (ED) sẽ đặt giá trị logic “1” vào bit đầu tiên của Stack 0 trong một chu kỳ quét chương trình khi phát hiện sự chuyển trạng thái từ 1 xuống 0 trong Stack 0. Còn các trường hợp khác nó sẽ đặt Stack 0 về “0”.

Ví dụ: Lấy cạnh lên của I0.0 xuất ra Q0.0, còn cạnh xuống xuất ra Q0.1.



6.7 Các Bit nhớ đặc biệt (Special Memory bits)

Các bit nhớ SM (Special memory bits) cung cấp nhiều chức năng trạng thái và điều khiển, cũng như cung cấp thông tin truyền thông giữa S7-200 và chương trình. Các bit nhớ đặc biệt có thể được sử dụng ở dạng bits, bytes, words và double words. Trong phần này chỉ trình bày các bit trạng thái của SMB0. Còn các bit nhớ SM khác sẽ được trình bày ở mỗi chương tương ứng trong quyển sách này.

SMB0 chứa tám bit trạng thái và được cập nhật ở mỗi chu kỳ quét của S7-200. Đây là các bit nhớ chỉ đọc.

Bit	Chức năng
SM0.0	Bit luôn luôn có trạng thái 1
SM0.1	Bit có trạng thái 1 ở vòng quét đầu tiên của chương trình
SM0.2	Bit báo dữ liệu bị thất lạc (0: dữ liệu còn đủ, 1: dữ liệu bị thất lạc).
SM0.3	Bit báo PLC được đóng nguồn. (1: ở vòng quét đầu tiên, 0: ở các vòng quét còn lại).
SM0.4	Bit tạo ra xung có chu kỳ 1 phút (0: trong 30s đầu, 1 trong 30s sau).
SM0.5	Bit tạo xung có chu kỳ 1s (tần số 1 Hz) (0: trong 0,5s đầu ; 1 trong 0,5 s sau).
SM0.6	Bit lên 1 ở một vòng quét và xuống 0 ở vòng quét tiếp theo. Nó được sử dụng để làm ngõ vào của bộ đếm vòng quét.
SM0.7	Bit báo vị trí của công tắc chọn chế độ làm việc của PLC (0: TERM, 1: RUN).

CHƯƠNG 7: BỘ ĐỊNH THỜI (TIMER) VÀ BỘ ĐẾM (COUNTER)

7.1 Giới thiệu bộ định thời

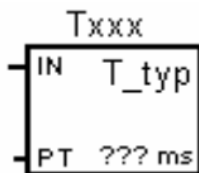
Bộ định thời được sử dụng trong các yêu cầu điều khiển cần trì hoãn về thời gian. Đây là phần tử chức năng cơ bản của các bộ PLC và rất thường được sử dụng trong các chương trình điều khiển. Chẳng hạn như một băng tải khi có tín hiệu hoạt động sẽ chạy trong 10s rồi dừng lại, một van khí nén cần có điện trong 5s, nguyên liệu cần trộn trong thời gian 10 phút... Các PLC S7-200 có 256 Timer có địa chỉ từ T0 đến T255, chia làm 3 loại (xem thêm chương 4 Bộ điều khiển lập trình PLC S7-200) :

- + Timer đóng mạch chậm TON (On-delay Timer).
- + Timer đóng mạch chậm có nhớ TONR (Retentive On-delay Timer).
- + Timer ngắt mạch chậm TOF (Off-delay Timer).

Khi sử dụng một timer chúng ta cần phải xác định các thông số sau:

- Loại timer (TON, TONR hay TOF)
- Độ phân giải của Timer. Có 3 độ phân giải là: 1ms, 10ms và 100ms
- Địa chỉ của timer sẽ sử dụng, ví dụ T0, T37... cần tra bảng để biết loại timer sử dụng tương ứng với các số nào.
- Khai báo hằng số thời gian tương ứng với thời gian cần trì hoãn dựa vào độ phân giải của timer.
- Tín hiệu cho phép bắt đầu tính thời gian.

Ký hiệu chung của Timer S7-200 biểu diễn ở LAD như sau:



Với:

Txxx: Ký hiệu và số thứ tự của timer, ví dụ: T37

IN: Ngõ vào bit, cho phép timer hoạt động

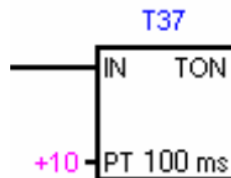
PT: Ngõ vào số Integer, hằng số thời gian.

T_typ: Cho biết loại Timer. Có thể là TON, TONR hay TOF

???ms: Báo độ phân giải của timer, tự động xuất hiện theo Txxx.

Thời gian định thời = [PT] x [???ms].

Ví dụ ta có



Đây là loại On-delay timer, có tên gọi là T37, có độ phân giải là 100ms. Thời gian định thời là : $10 \times 100\text{ms} = 1\text{s}$.

7.2 Timer TON

Các Timer này được sử dụng khi có các yêu cầu trì hoãn một khoảng thời gian. Giá trị hiện hành của TON bị xóa khi ngõ vào IN ở logic “0”.

On-Delay Timer (TON) thực hiện đếm thời gian khi ngõ vào IN ở mức logic “1”. Khi giá trị hiện hành (Txxx) lớn hơn hoặc bằng thời gian đặt trước PT (preset time), thì Timer Bit ở

logic “1”. Giá trị hiện hành của TON bị xóa khi ngõ vào IN ở logic “0”. Timer tiếp tục đếm dù đã đạt đến giá trị đặt PT, và dừng lại khi đếm đến giá trị max. 32767.

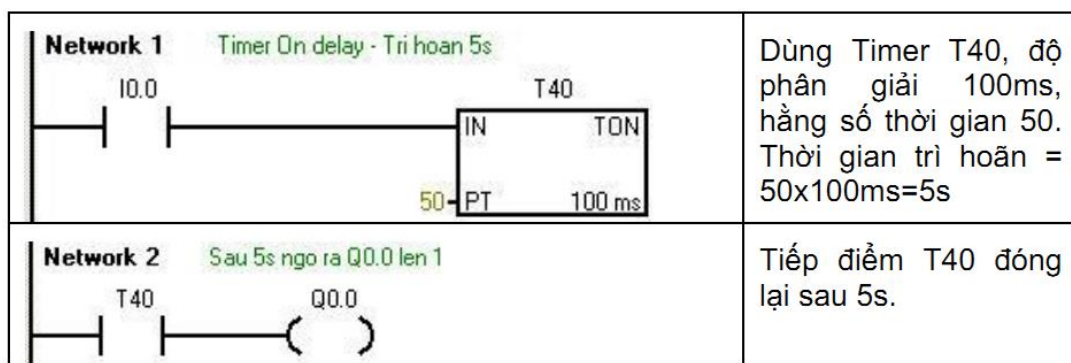
Để xóa timer, có thể sử dụng lệnh Reset (R). Lệnh Reset sẽ làm cho Timer Bit ở mức logic “0” và giá trị hiện hành của timer (Timer Current) =0.

Có 192 timer TON/TOF trong S7-200 được phân chia theo độ phân giải như ở bảng sau:

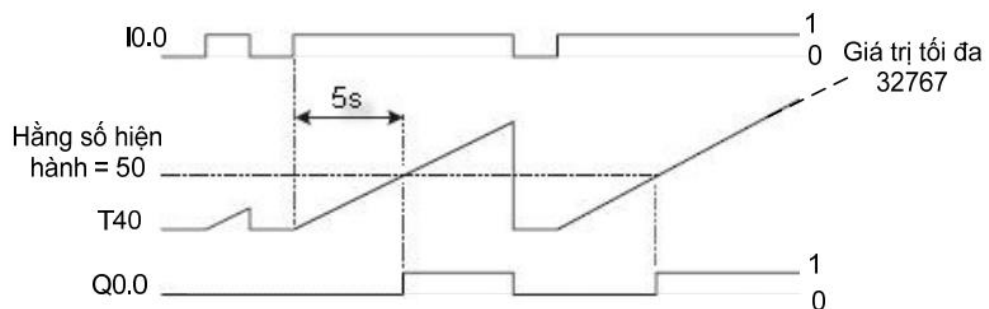
Số Timer	Độ phân giải	Thời gian trì hoãn tối đa
T32, T96	1ms	32,767s
T33 ... T36, T97 ... T100	10ms	327,67s
T37 ... T63, T101 ... T255	100ms	3276,7s

Chú ý: Vì TON và TOF sử dụng cùng địa chỉ timer, nên không thể đặt cho cả hai có cùng số Timer. Ví dụ đã đặt TON là T37 thì không được đặt TOF là T37.

Ví dụ: Bật công tắc I0.0 (NO) thì sau 5s ngõ ra Q0.0 lên mức 1.



Giản đồ thời gian:



Qua giản đồ trên ta nhận thấy để timer TON trì hoãn được hết thời gian đặt trước (ví dụ 5s) thì trạng thái tín hiệu tại ngõ vào IN cần được duy trì ở mức 1 trong suốt khoảng thời gian này. Nếu sau 5s mà ngõ vào IN vẫn duy trì ở mức 1 thì giá trị hằng số thời gian trong timer sẽ tiếp tục tăng cho tới khi đạt giá trị tối đa là 32767.

7.3 Timer TONR

Các Timer này được sử dụng khi cần tích lũy một số khoảng thời gian rời rạc. Giá trị hiện hành TONR chỉ có thể bị xóa bằng lệnh Reset (R).

Timer đóng mạch chậm có nhớ TONR (Retentive On-Delay Timer) thực hiện đếm thời gian khi ngõ vào IN ở mức logic “1”. Khi giá trị hiện hành (Txxx) lớn hơn hoặc bằng thời gian đặt trước PT (preset time), thì Timer Bit ở logic “1”. Giá trị hiện hành của TONR được giữ lại khi ngõ vào IN ở logic “0”. TONR được sử dụng để tích lũy thời gian cho nhiều chu kỳ ngõ vào IN ở mức “1”. Timer này vẫn tiếp tục đếm sau khi đã đạt đến giá trị đặt trước và

dừng lại ở giá trị max. 32767.

Để xóa giá trị hiện hành của TONR và Timer Bit, ta sử dụng lệnh Reset (R).

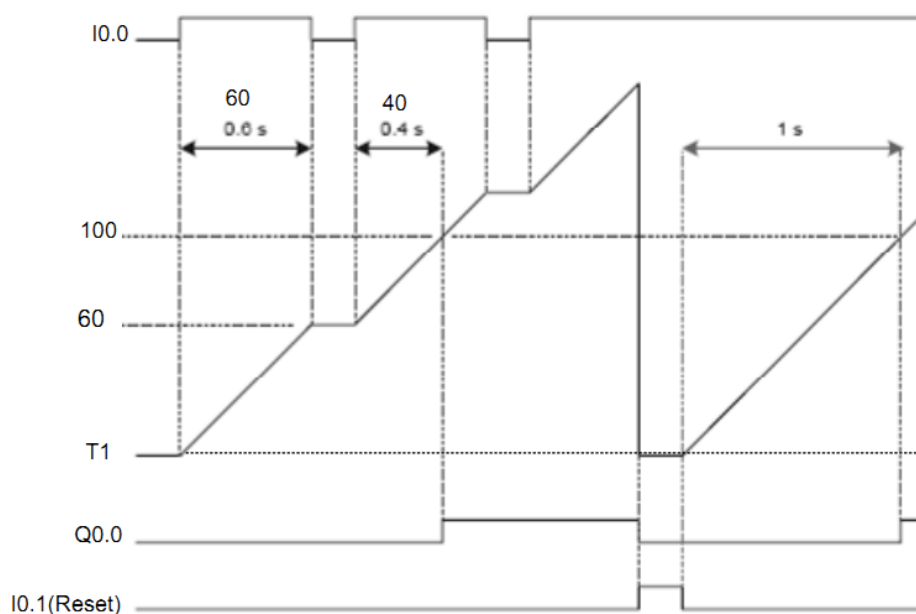
Có 64 timer TONR trong S7-200 được phân chia theo độ phân giải như ở bảng sau:

Số Timer	Độ phân giải	Thời gian trì hoãn tối đa
T0, T64	1 ms	32,767 s
T1 ... T4, T65 ... T68	10 ms	327,67 s
T5 ... T31, T69 ... T95	100 ms	3276,7 s

Ví dụ: Xét đoạn chương trình

Network 1 	Tín hiệu I0.0 kích hoạt timer TONR T1 có độ phân giải 10ms (thời gian = 100 x 10ms = 1s)
Network 2 	Sau 1 s ngõ ra Q0.0 lên mức 1
Network 3 	Tín hiệu I0.1 Reset timer T1

Giản đồ thời gian:



7.4 Timer TOF

Sử dụng timer này khi cần trì hoãn thêm một khoảng thời gian rồi mới tắt ngõ ra kể từ khi tín hiệu ngõ vào IN xuống “0”. Timer TOF chỉ thực hiện đếm thời gian khi IN chuyển từ “1” xuống “0”.

Khi ngõ vào IN của Off-Delay Timer (TOF) ở logic “1”, thì Timer Bit ngay lập tức được đặt lên mức logic “1” và giá trị hiện hành được xóa về 0. Khi ngõ vào IN xuống “0”, thì timer đếm cho đến khi thời gian trôi qua đạt đến giá trị thời gian đặt trước. Khi đạt đến giá trị

đặt trước, Timer Bit được đặt về “0” và giá trị hiện hành dừng đếm. Nếu ngõ vào IN ở “0” trong khoảng thời gian ngắn hơn giá trị đặt trước, thì Timer Bit giữ ở “1”.

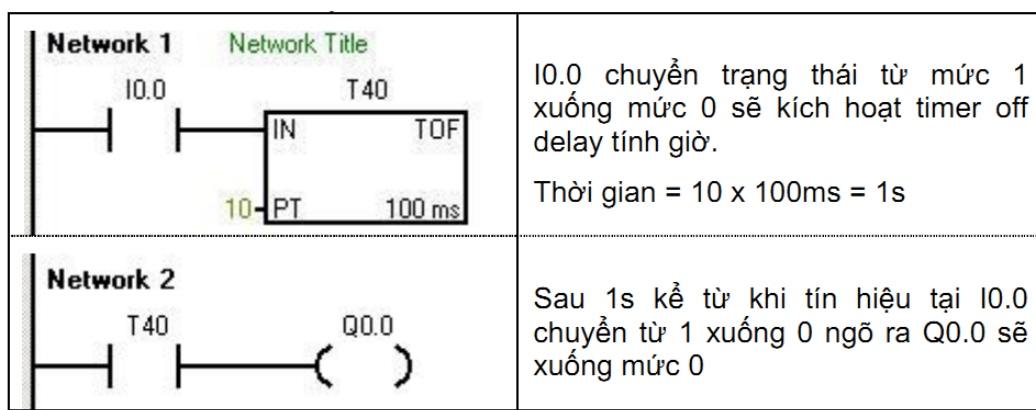
Để xóa timer, có thể sử dụng lệnh Reset (R). Lệnh Reset sẽ làm cho Timer Bit ở mức logic “0” và giá trị hiện hành của timer (Timer Current) =0.

Có 192 timer TON/TOF trong S7-200 được phân chia theo độ phân giải như ở bảng sau:

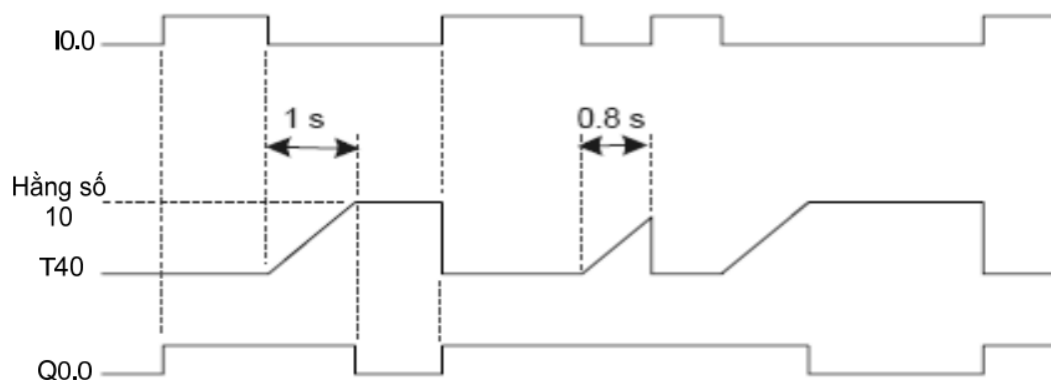
Số Timer	Độ phân giải	Thời gian trì hoãn tối đa
T32, T96	1ms	32,767s
T33 ... T36, T97 ... T100	10ms	327,67s
T37 ... T63, T101 ... T255	100ms	3276,7s

Chú ý: Vì TON và TOF sử dụng cùng số timer, nên không thể đặt cho cả hai có cùng số Timer. Ví dụ đã đặt TON là T37 thì không được đặt TOF là T37.

Ví dụ: Xét đoạn chương trình



Giản đồ thời gian:



7.5 Giới thiệu bộ đếm (counter)

Trong nhiều trường hợp, việc kiểm tra một số lượng xác định phải thông qua tổng các xung. Có thể thực hiện đếm các xung này bằng các bộ đếm. Sử dụng bộ đếm có thể giải quyết được một số vấn đề sau:

- Đếm số lượng
- So sánh với một giá trị đặt trước ở các trường hợp bằng nhau, nhỏ hơn, lớn hơn.
- Kiểm tra sự khác biệt về số lượng.

Trong điều khiển vị trí thì việc sử dụng bộ đếm tốc độ cao là không thể thiếu. Phần điều khiển vị trí và bộ đếm tốc độ cao sẽ được trình bày chi tiết phần sau. Ở chương này chỉ đề cập đến các bộ đếm thông thường.

Bộ đếm cũng có thể sử dụng để thực hiện các nhiệm vụ như: Cộng các xung của bộ phát xung nhịp và dựa vào đó để gọi các giai đoạn điều khiển liên tiếp nhau. Hoặc các yêu cầu điều khiển theo chu kỳ lặp như điều khiển đèn giao thông.

Các PLC thường có 3 loại bộ đếm: bộ đếm lên, bộ đếm xuống, bộ đếm lên-xuống.

Có 256 bộ đếm ở S7-200 có địa chỉ từ C0 đến C255. Chúng cũng có 3 loại bộ đếm là:

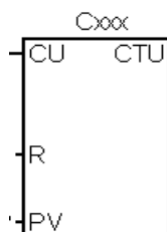
- + Bộ đếm lên CTU (Up Counter).
- + Bộ đếm xuống CTD (Down Counter).
- + Bộ đếm lên-xuống (Up/Down Counter).

Khi sử dụng một counter chúng ta cần phải xác định các thông số sau:

- Loại counter (CTU, CTD hay CTUD)
- Số của counter sẽ sử dụng, không được gán cùng một số counter cho nhiều counter.
- Khai báo giá trị cần đếm cho counter.
- Tín hiệu xung cung cấp cho bộ đếm.
- Tín hiệu xóa bộ đếm.

7.6 Bộ đếm lên CTU (Count Up)

Bộ đếm CTU được biểu diễn ở LAD như sau:



Với:

Cxxx: Ký hiệu và số thứ tự của counter, ví dụ: C10. CTU: Ký hiệu nhận biết bộ đếm lên

CU: Đếm lên. Ngõ vào bit,

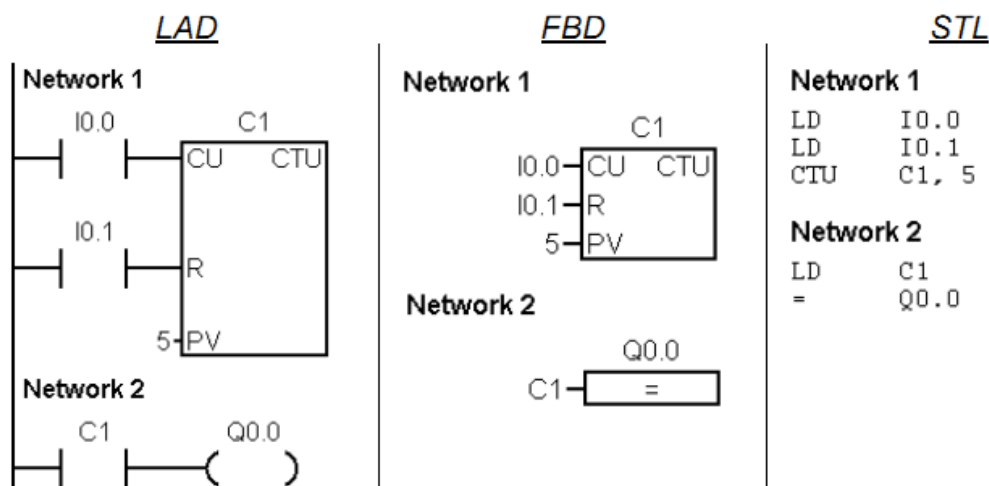
R: Xóa bộ đếm về 0. Ngõ vào bit,

PV: Giá trị đặt trước cho bộ đếm. Biểu diễn ở số Integer.

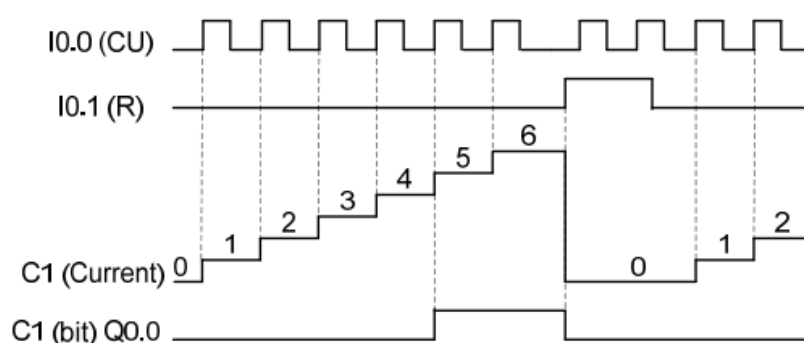
Mỗi khi tín hiệu tại CU từ mức “0” lên “1” thì bộ đếm sẽ tăng giá trị hiện hành của nó lên 1 đơn vị. Khi giá trị hiện hành của bộ đếm (Cxxx) lớn hơn hoặc bằng giá trị đặt trước tại ngõ vào PV (Preset Value) thì ngõ ra bit của counter (counter bit) sẽ lên mức “1”. Giá trị đếm lên tối đa là 32.767. Phạm vi của bộ đếm là C0 đến C255.

Bộ đếm sẽ bị xóa về 0 khi ngõ vào Reset (R) lên mức “1”, hoặc khi sử dụng lệnh Reset để xóa bộ đếm.

Ví dụ: Cứ mỗi xung từ “0” chuyển lên “1” tại ngõ vào I0.0, bộ đếm sẽ tăng 1 đơn vị. Từ xung thứ 5 trở đi ngõ ra Q0.0 sẽ lên “1”. Nếu có xung vào tại ngõ I0.1 thì ngõ ra Q0.0 xuống “0”.

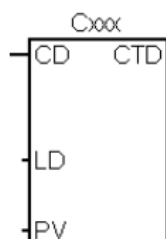


Giải đồ xung:



7.7 Bộ đếm xuống CTD (Count Down)

Bộ đếm xuống CTD được biểu diễn ở LAD như sau:



Với:

Cxxx: Ký hiệu và số thứ tự của counter, ví dụ: C20. CTD: Ký hiệu nhận biết bộ đếm xuống

CD: Ngõ vào đếm xuống. Ngõ vào bit,

LD: Nạp giá trị đặt trước cho bộ đếm xuống. Ngõ vào bit,

PV: Giá trị đặt trước cho bộ đếm. Biểu diễn ở số Integer.

Mỗi khi tín hiệu tại CD từ mức “0” lên “1” thì bộ đếm sẽ giảm giá trị hiện hành của nó xuống 1 đơn vị. Khi giá trị hiện hành của bộ đếm (Cxxx) bằng 0, thì Counter Bit Cxxx lên “1”. Bộ đếm xóa Counter Bit Cxxx và nạp giá trị đặt trước ở PV khi ngõ vào LD (load) lên mức “1”.

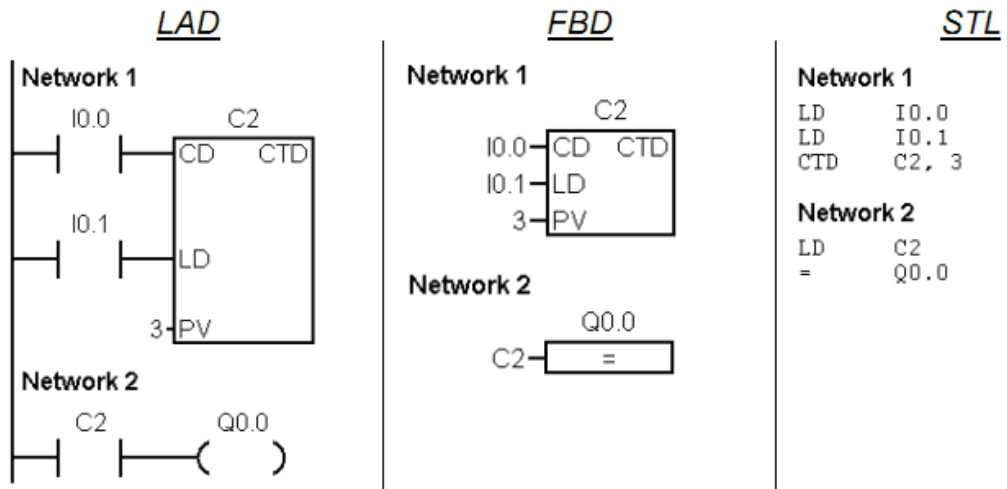
Bộ đếm sẽ dừng đếm khi giá trị hiện hành bằng 0 và counter bit Cxxx lên “1”.

Phạm vi của bộ đếm là C0 đến C255.

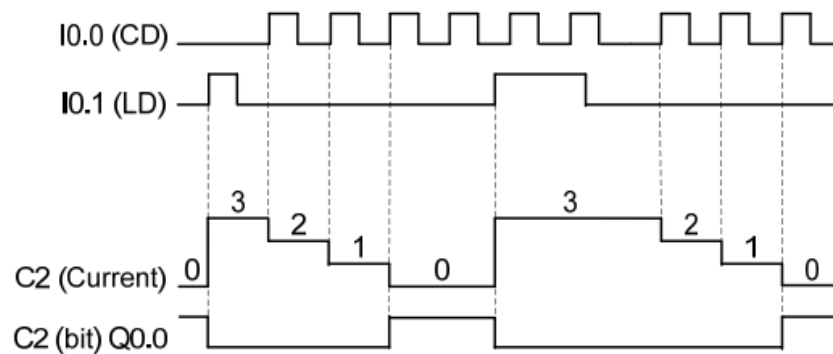
Khi xóa bộ đếm bằng lệnh Reset, counter bit bị xóa và giá trị hiện hành được đặt về 0.

Ví dụ: Sử dụng bộ đếm xuống C2, giá trị hiện hành giảm từ 3 trở về 0. Với I0.1 ở logic

“0” và mỗi lần I0.0 chuyển từ “0” lên “1” thì bộ đếm C2 giảm đi một đơn vị. Khi giá trị hiện hành trong bộ đếm C2 bằng 0 thì ngõ ra Q0.0 lên “1”. Khi I0.1 ở “1” thì bộ đếm được đặt trước giá trị đếm là 3.

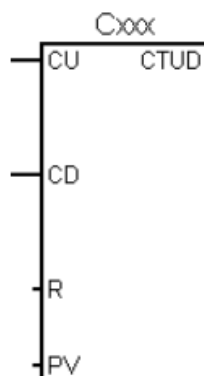


Giản đồ xung:



7.8 Bộ đếm lên-xuống CTUD (Count Up/Down)

Bộ đếm xuống CTUD được biểu diễn ở LAD như sau:



Với:

Cxxx: Ký hiệu và số thứ tự của counter, ví dụ: C0.

CTUD: Ký hiệu nhận biết bộ đếm lên-xuống

CU: Ngõ vào đếm lên. Ngõ vào bit

CD: Ngõ vào đếm xuống. Ngõ vào bit,

R: Xóa bộ đếm về 0. Ngõ vào bit,

PV: Giá trị đặt trước cho bộ đếm. Biểu diễn ở số Integer.

Lệnh đếm lên-xuống (CTUD) sẽ đếm lên mỗi khi ngõ vào đếm lên (CU) từ mức “0” lên “1”, và đếm xuống mỗi khi ngõ vào đếm xuống (CD) chuyển từ “0” lên “1”. Giá trị hiện hành Cxxx giữ giá trị hiện hành của bộ đếm. Giá trị đặt trước PV được so sánh với giá trị hiện hành mỗi khi thực hiện lệnh đếm.

Khi đạt đến giá trị max (32.767), thì ở cạnh lên kế tiếp tại ngõ vào đếm lên bộ đếm sẽ đặt về giá trị min (-32.768).

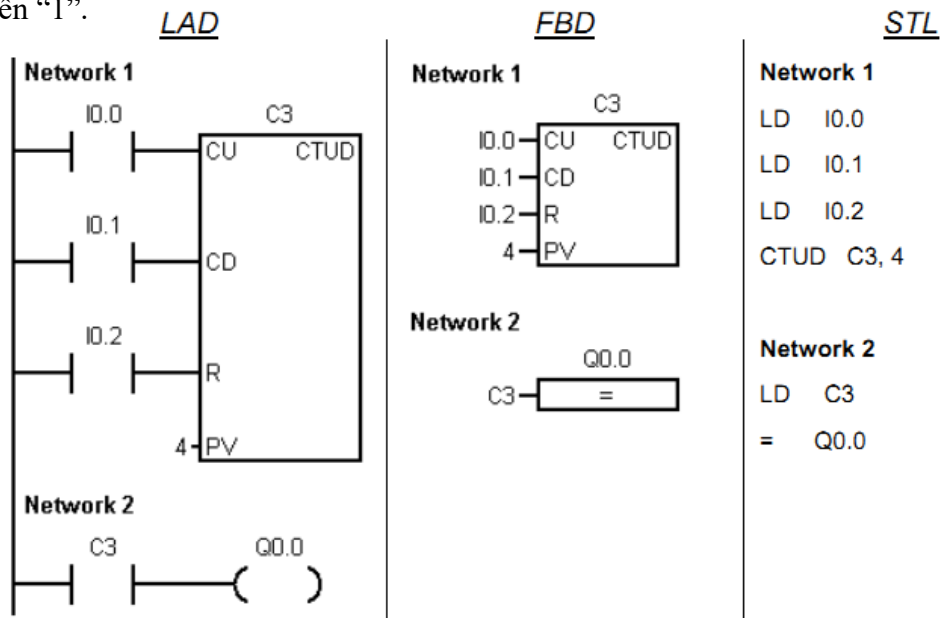
Khi đạt đến giá trị min (-32.768), thì ở cạnh lên kế tiếp tại ngõ vào đếm xuống bộ đếm sẽ đặt về giá trị max (32.767).

Khi giá trị hiện hành Cxxx lớn hơn hoặc bằng giá trị đặt trước PV, thì Counter Bit Cxxx lên “1”. Ngược lại Counter Bit Cxxx bằng “0”.

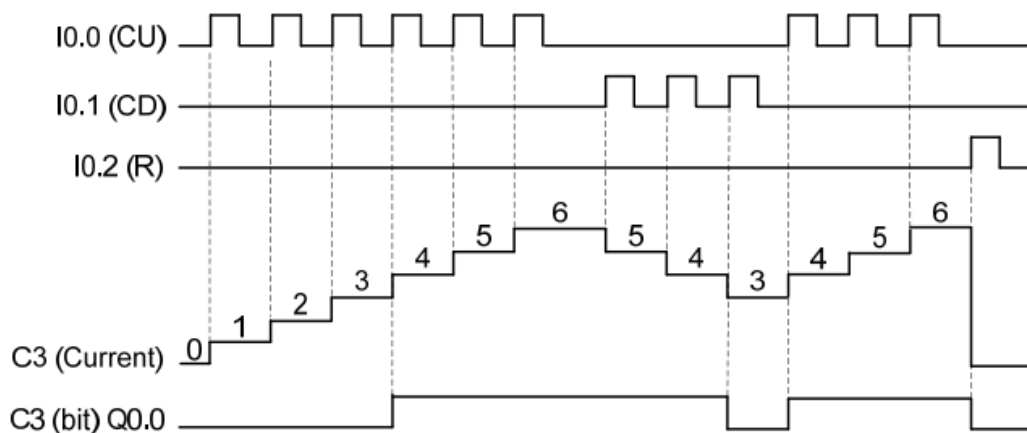
Phạm vi của bộ đếm là C0 đến C255.

Bộ đếm sẽ bị xóa về 0 khi ngõ vào Reset (R) lên mức “1”, hoặc khi sử dụng lệnh Reset để xóa bộ đếm.

Ví dụ: Sử dụng bộ đếm lên-xuống C3. Ngõ vào đếm lên nối với I0.0. Ngõ vào đếm xuống nối với I0.1. Xóa bộ đếm bằng I0.2. Khi bộ đếm có giá trị hiện hành ≥ 4 thì ngõ ra Q0.0 lên “1”.



Giản đồ xung:



CHƯƠNG 8: PHƯƠNG PHÁP LẬP TRÌNH ĐIỀU KHIỂN TUẦN TỰ

8.1 Cấu trúc chung của một chương trình điều khiển

Trong phần này đề cập đến việc tổ chức và cấu trúc cho chương trình PLC, nghĩa là trong chương trình điều khiển gồm các phần có liên quan đến các vấn đề như các chế độ hoạt động, các chức năng cơ bản, trình tự xử lý, kích hoạt các ngõ ra, hiển thị trạng thái theo trình tự sau:

1. Bắt đầu chương trình
2. Các chế độ hoạt động và các chức năng cơ bản
 - Khởi tạo vị trí cơ bản.
 - Các điều kiện cho phép của ngõ ra.
 - Mạch logic điều khiển.
 - Kích hoạt các ngõ ra.
 - Xuất các chỉ thị, chỉ báo.
3. Kết thúc chương trình.

- **Đoạn chương trình điều khiển chế độ hoạt động**

- * Khởi tạo vị trí cơ bản

Các thiết bị vật lý được điều khiển đều có vị trí cơ bản, ví dụ khi các cơ cấu tác động ở các trạng thái OFF và các công tắc hành trình ở vị trí hở. Tất cả các yếu tố này có thể được tổ hợp logic với nhau để báo hiệu và khởi tạo vị trí cơ bản, và được lập trình như là một bước trong chuỗi trình tự.

- Đoạn chương trình chức năng khởi động hay dừng quá trình điều khiển.

Hầu hết các điều khiển trong công nghiệp đều có nút khởi động (START) và nút dừng (STOP) mà có thể lập trình cho hành vi của chúng. Các nút này được lập trình bằng các tiếp điểm logic thực hiện khởi động hay dừng toàn bộ hoạt động điều khiển của PLC. Cũng có thể có một công tắc bằng tay để cho phép hay không cho phép các ngõ ra, dùng khi kiểm tra chương trình.

- **Đoạn chương trình xử lý điều khiển**

Đây là phần chính của chương này, bao gồm việc thiết kế và lập trình các điều khiển dùng cơ chế trình tự hay logic tổ hợp. Các kết quả của sự tổ hợp logic trên thường không trực tiếp kích các cơ cấu chấp hành, mà thông qua các ô nhớ trung gian.

- **Đoạn chương trình kích các ngõ ra**

Các tín hiệu ngõ ra dùng để kích cơ cấu tác động được khoá lần bởi các ô nhớ trung gian hình thành từ các đoạn chương trình xử lý điều khiển.

- **Đoạn chương trình xuất các chỉ thị, chỉ báo**

Các trạng thái của quá trình hoạt động thường được biểu thị bằng đèn, chuông... để người vận hành máy có các quyết định thích hợp.

Việc lập trình theo cấu trúc như trên nhằm làm cho chương trình điều khiển có độ tin cậy cao hơn, dễ hiểu hơn, cho phép xác định lỗi nhanh chóng và rút ngắn được thời gian bảo trì, sửa chữa.

8.2 Điều khiển tuần tự

8.2.1 Giới thiệu

Trong công nghiệp, hầu hết các dự án điều khiển xảy ra một cách tuần tự, khâu xử lý sau

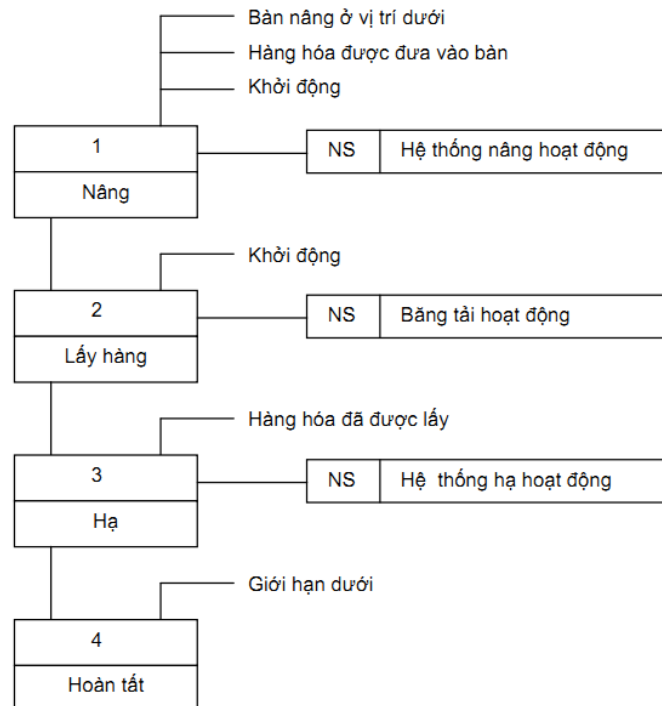
chậm hơn khâu xử lý trước một khoảng thời gian xác định. Ví dụ như quá trình chuyển động mới bắt đầu nếu như một quá trình khác được kết thúc.

Vấn đề này có thể được giải quyết bằng điều khiển liên kết, với việc kết nối cứng các điều kiện trong chương trình. Nhưng ở đây chỉ ra rằng từ một khuôn khổ điều khiển đã biết thì việc giải quyết vấn đề bằng điều khiển liên kết là rất khó đọc chương trình và việc tìm lỗi phải mất nhiều thời gian.

Nếu một dự án được thực hiện theo phương pháp điều khiển trình tự thì cấu trúc chương trình có thể nhận biết một cách dễ dàng và dự án có thể được biểu diễn bằng hình ảnh. Điều khiển trình tự giúp cho người đọc đọc chương trình một cách dễ dàng, chương trình điều khiển được trình bày theo cấu trúc, ưu điểm của nó là giúp cho việc lập trình, thay đổi và tìm lỗi các dự án một cách có hiệu quả.

Để dễ hiểu ta xét Một hệ thống nâng hàng hoạt động như sau :

Bàn nâng ở vị trí dưới và hàng hoá sẽ được đưa vào bàn nâng. Nếu nút khởi động được ấn thì bàn nâng được hệ thống nâng đưa lên cao, khi lên đến giới hạn trên thì hệ thống nâng ngừng lại và băng tải trên bàn nâng hoạt động kéo hàng hoá đưa sang bộ phận khác. Sau khi hàng hoá được lấy xong thì băng tải dừng, lúc này bàn sẽ được hạ xuống khi đến vị trí dưới thì dừng lại, và một quá trình mới lại bắt đầu. Từ yêu cầu công nghệ của hệ thống nâng hàng này ta có thể biểu diễn theo phương pháp điều khiển trình tự như ở hình 9.1.



Hình 9.1: Ví dụ hệ thống nâng hàng được biểu diễn theo sơ đồ chức năng trong điều khiển tuần tự.

Ưu điểm của phương pháp điều khiển tuần tự là:

- Thiết kế, lập trình nhanh và đơn giản.
- Cấu trúc chương trình rõ ràng.
- Thay đổi dễ dàng trình tự thực hiện.
- Nhận biết nhanh chóng các nguyên gây ra lỗi.
- Nhiều kiểu hoạt động khác nhau có thể thực hiện được.

Từ các ưu điểm này mà trong thực tế rất nhiều bài toán điều khiển được giải quyết bằng phương pháp điều khiển trình tự. Điều khiển trình tự có thể chia làm hai loại:

• Điều khiển tuần tự theo thời gian .

Ở điều khiển trình tự theo thời gian thì điều kiện chuyển tiếp chỉ phụ thuộc vào thời gian.

Các khâu định thời, bộ đếm thời gian...để tạo ra điều kiện chuyển tiếp.

- **Điều khiển tuần tự theo quá trình :**

Ở điều khiển trình tự theo quá trình thì điều kiện chuyển tiếp phụ thuộc vào các tín hiệu của thiết bị được điều khiển. Các thông báo về từ các sự kiện của xử lý có thể là vị trí van các bộ giám sát hoạt động, lưu lượng áp suất, nhiệt độ, độ dẫn, độ nhớt ... Trong nhiều trường hợp các thông báo về từ việc xử lý phải được biến đổi thành tín hiệu nhị phân .

Một dạng của điều khiển trình tự phụ thuộc vào quá trình xử lý của điều khiển theo hành trình, điều kiện chuyển tiếp của nó chỉ phụ thuộc vào các tín hiệu hành trình của thiết bị được điều khiển.

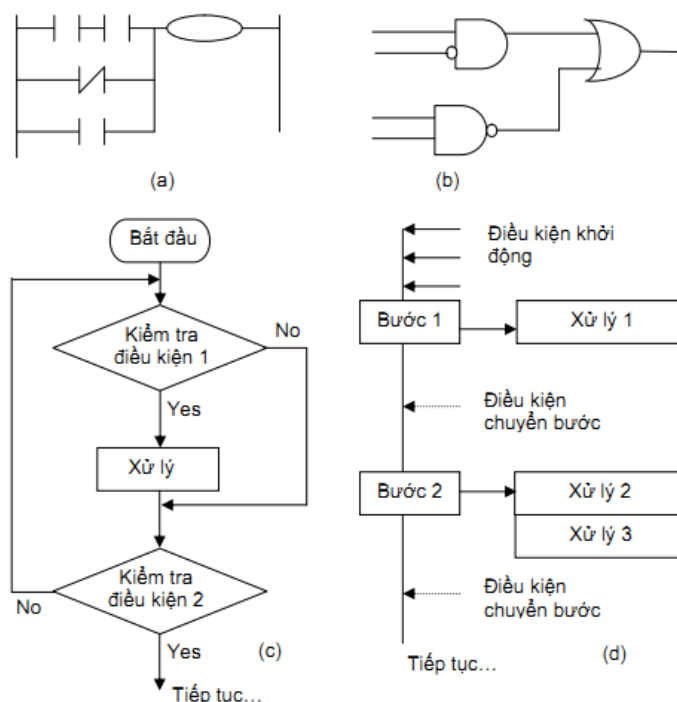
8.2.2 Phương pháp lập trình điều khiển tuần tự

Các bước thiết kế chương trình trình tự cho PLC như sau :

- Quá trình điều khiển được diễn đạt bằng lời.
- Sự mô tả đó được chuyển sang dạng lưu đồ hay sơ đồ chức năng.
- Đến giai đoạn này, các điều kiện logic dễ dàng được xác định, sau đó được chuyển sang biểu thức boolean biểu diễn từng trạng thái của quá trình tuần tự.
- Cuối cùng biểu thức boolean được chuyển đổi sang chương trình trong PLC.

Sự diễn đạt bằng lời hay ghi ra giấy mô tả quá trình điều khiển thường dài, khó theo dõi và không chính xác. Như đã đề cập, toàn bộ quá trình điều khiển sẽ dễ hiểu hơn khi nó chia thành những đơn vị con (sub-units) hay xử lý con (sub- processor). Mỗi đơn vị con sau đó có thể được xây dựng theo dạng trình tự và khóa lẫn để thực hiện một chức năng nào đó theo yêu cầu. Cần có các phương pháp để mô tả hệ thống trình tự như trên sao cho rõ ràng và dễ theo dõi quá trình hoạt động.

Các phương pháp diễn đạt có thể tùy chọn: logic relay (relay logic diagram), công logic (logic schematics), lưu đồ (flowcharts) và sơ đồ chức năng (function charts) như hình 9.2. Các phương pháp này không thay thế cho bước diễn đạt bằng lời mà nó hỗ trợ rất nhiều cho bước này. Việc áp dụng phương pháp nào tùy thuộc chủ yếu vào kinh nghiệm về phương pháp đó. Người phân tích thiết kế hệ thống có kiến thức tốt về kỹ thuật số hay về máy tính thì thường dùng 3 phương pháp sau, còn phương pháp logic relay được dùng đối với những người quen với thiết kế mạch relay.



Hình 9.2: Các phương pháp mô tả hệ thống điều khiển logic

- **Phương pháp logic relay và công logic**

Cả hai phương pháp có liên hệ trực tiếp đến mạch vật lý, nên việc dùng PLC để thay thế hệ thống relay truyền thống là lý tưởng. Các phương pháp này thường dùng cho hệ thống điều khiển dùng tổ hợp các ngõ vào hay các hệ thống trình tự qui mô nhỏ vì sơ đồ biểu diễn cho trình tự qui mô lớn phức tạp và khó theo dõi.

- **Phương pháp biểu diễn theo lưu đồ**

Phương pháp này thường dùng khi thiết kế phần mềm cho máy tính, nhưng lại phổ biến để biểu diễn trình tự hoạt động của hệ thống điều khiển. Lưu đồ có quan hệ trực tiếp đến sự mô tả bằng lời hệ thống điều khiển, chỉ ra từng điều kiện cần kiểm tra từng bước và các xử lý trong các bước đó theo chuỗi trình tự. Các xử lý trong lưu đồ được ghi trong 1 ô chữ nhật, trong khi các điều kiện được ghi vào ô hình thoi. Tuy nhiên, phương pháp này chiếm nhiều không gian khi biểu diễn hệ thống điều khiển lớn và trở nên nặng nề.

- **Phương pháp sơ đồ chức năng**

Phương pháp này ngày càng trở nên phổ biến để biểu diễn các hoạt động trình tự, cho phép thể hiện chi tiết về các xử lý cũng như trình tự các hoạt động trong quá trình điều khiển. Với cách dùng các ký hiệu gọn và cô đọng, phương pháp này có được ưu điểm của các phương pháp trên, việc biểu diễn bước tiến trình hoạt động mạch lạc và rõ ràng. Trong từng bước ta có thể ghi ra các điều kiện set và reset, điều kiện chuyển trạng thái và các tín hiệu điều khiển khác. Sơ đồ chức năng còn thể hiện đặc lực khi kiểm tra và thử hệ thống.

- **Đại số Boolean**

Cho dù dùng phương pháp nào đi nữa, một khi các chức năng đã được đặc tả rõ ràng thì chúng phải được chuyển đổi sang dạng mà từ đó có thể chuyển thành chương trình PLC. Quá trình này được thực hiện bằng cách chuyển đổi các chức năng thành 1 chuỗi liên tiếp biểu thức boolean, và từ đó chuyển thành ngôn ngữ PLC. Một khi quen với kỹ thuật này, ta có thể dễ dàng chuyển đổi sự đặc tả chức năng thành biểu thức boolean bất kể là nó được đặc tả bằng phương pháp nào.

Ta cũng có thể đặc tả toàn bộ hệ thống điều khiển logic bằng biểu thức boolean, mặc dù việc dùng biểu thức Boolean thường kém hiệu quả về mặt thời gian thiết kế và không dễ hiểu đối với những người chưa có kinh nghiệm về các hệ thống điều khiển. Giải pháp dùng Boolean dù sao đi nữa cũng tiết kiệm được không gian biểu diễn trên giấy khi thiết kế.

Trong các phương pháp lập trình cho điều khiển trình tự trên thì phương pháp sơ đồ chức năng có ưu điểm hơn các phương pháp khác. Cho nên chương này chọn phương pháp sơ đồ chức năng để làm cơ sở chính cho việc thiết kế điều khiển trình tự.

8.3 Các thủ tục tổng quát để thiết kế bài toán tuần tự

Trong bài toán điều khiển trình tự, để thực hiện một cách có hệ thống công việc điều khiển và tránh tối đa những thiếu sót, nhầm lẫn thì thủ tục để thiết kế bài toán tuần tự bao gồm các bước như sau:

Bước 1: Xây dựng sơ đồ phối hợp thao tác công nghệ của máy hoặc hệ thống thiết bị cần điều khiển.

Đây là công việc có yêu cầu tương tự như khi bắt tay vào việc thiết kế một máy mới. Người thực hiện sẽ căn cứ vào yêu cầu hoạt động của máy để từ đó hình dung và phân tích ra một trình tự các thao tác thật chi tiết của các khâu chấp hành hoặc từng bộ phận chấp hành của máy cũng như sự hoạt động giữa chúng. Quá trình phân tích và thực hiện việc phối hợp các chuyển động hoặc các thao tác thường được thực hiện dưới dạng một sơ đồ phối hợp. Sơ đồ được thực hiện dưới dạng các dải hình chữ nhật đặt kế tiếp nhau. Mỗi dải tượng trưng cho diễn biến theo thời gian quá trình hoạt động của một khâu chấp hành hoặc một bộ phận chấp hành nhằm thực hiện một thao tác công nghệ nào đó.

Sơ đồ phối hợp các thao tác công nghệ cho phép người thiết kế hình dung toàn bộ quá trình hoạt động của máy hoặc của hệ thống thiết bị bao gồm trình tự các thao tác và thời điểm bắt đầu cũng như kết thúc thực hiện của từng thao tác. Sơ đồ phối hợp này sẽ là cơ sở cho việc soạn thảo chương trình điều khiển trên PLC cũng đồng thời là tài liệu gốc cho việc hiệu chỉnh sự làm việc máy hoặc hệ thống về sau.

Bước 2: Lập sơ đồ khối điều khiển tuần tự.

Căn cứ vào sơ đồ phối hợp các hoạt động hoặc các thao tác của các bộ phận chấp hành trên máy thiết kế, người cán bộ kỹ thuật sẽ thực hiện một công việc tương tự tiếp theo là lập sơ đồ khối điều khiển tuần tự (dạng lưu đồ (flowchart) hoặc sơ đồ chức năng (function-chart)). Công việc này là một bước tiếp cận hơn nữa của quá trình điều khiển. Tùy theo mức độ quen sử dụng cách biểu diễn nào mà người thiết kế sẽ lựa chọn các phương pháp biểu diễn quá trình điều khiển để mô tả chuỗi tuần tự các thao tác công nghệ cũng như các tín hiệu điều khiển cho từng thao tác.

Bước 3: Chuẩn bị phần cứng và mô tả các tham số vào/ra.

Công việc lựa chọn các cơ cấu chấp hành như lựa chọn các loại động cơ, xy lanh khí nén hoặc xy lanh dầu ép, lựa chọn các loại van điều khiển,..., có liên quan mật thiết với quá trình điều khiển đã tổng hợp do nhiều yếu tố như đặc tính kỹ thuật của cơ cấu tác động có phù hợp với máy thiết kế hay không, kết cấu có phù hợp hay không, không gian có cho phép bố trí loại cơ cấu tác động đó hay không; và một yếu tố quan trọng có tính chất quyết định là thời gian và tốc độ đáp ứng của cơ cấu tác động được lựa chọn có phù hợp, thỏa mãn với yêu cầu phối hợp trên máy hay không.

Người thiết kế phải lựa chọn kỹ để tìm kiếm các cơ cấu tác động phù hợp nhất và mô tả đầy đủ các thông số kỹ thuật của cơ cấu tác động, chẳng hạn như các giá trị điện áp, dòng điện tác động vào động cơ điện hay tác động vào các van điện từ điều khiển các van khí nén. Các tín hiệu trên có liên quan mật thiết với các tín hiệu ngõ ra của PLC. Tương tự, các tín hiệu từ các cảm biến; phản ánh trạng thái của cơ cấu tác động, được đưa đến các ngõ vào của PLC.

Thông qua việc lựa chọn và mô tả các tham số vào/ra này, người thiết kế sẽ cung cấp các số liệu cần thiết cho việc thiết kế các mạch giao tiếp giữa PLC với mạch công suất của các cơ cấu tác động, xác định số ngõ vào/ra để lựa chọn PLC thích hợp.

Bước 4: Lập trình.

Với đầy đủ các dữ liệu được cung cấp từ các bước thực hiện ở trên, công việc tiếp theo của người lập trình là soạn thảo chương trình điều khiển cho PLC để thực hiện việc điều khiển máy hoặc hệ thống hoạt động đúng cho chu trình đã thiết kế. Tùy theo khả năng quen sử dụng loại ngôn ngữ lập trình trên PLC nào mà người lập trình sẽ chọn lựa để soạn thảo chương trình. Với các chương trình đơn giản, các phần mềm của các hãng cho phép biên dịch được chương trình được viết từ ngôn ngữ này sang ngôn ngữ khác.

Bước 5: Chạy thử và hoàn chỉnh chương trình.

Đây là công việc hết sức tự nhiên phải thực hiện sau khi lập trình. Việc chạy thử chương trình được thực hiện trong 2 chế độ:

Chế độ giả lập (chế độ offline): Cho chạy chương trình và theo dõi đáp ứng của các ngõ ra thông qua các đèn LED. Đèn LED ở ngõ ra cụ thể sẽ biểu thị cho tín hiệu xuất ở ngõ ra cho cơ cấu tác động và đáp ứng của chúng.

Chế độ thực (chế độ online): Sau khi đã chạy thử và điều chỉnh chương trình trong chế độ giả lập hoàn hảo. Chuyển chế độ hoạt động trên PLC và nối phần mạch giao tiếp với mạch công suất để điều khiển máy chạy trong chế độ thực. Trong chế độ này, với các đáp ứng thực của các cơ cấu tác động khi không tải và khi có tải sẽ giúp cho người lập trình hiệu chỉnh chương trình lần cuối trước khi đưa vào vận hành thực sự trong sản xuất.

8.4 Các kiểu cấu trúc của bài toán điều khiển tuần tự

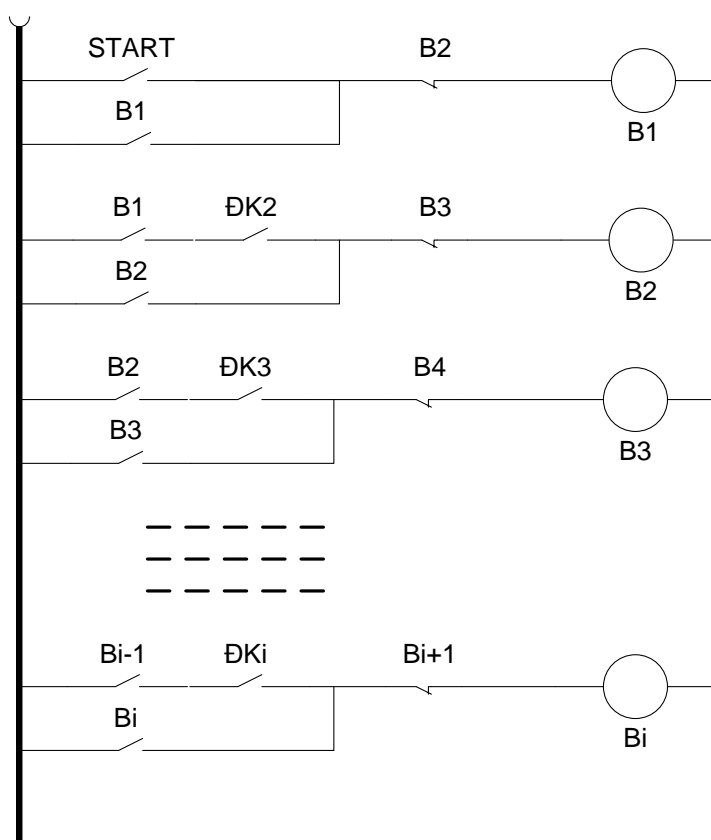
8.4.1 Hệ thống tuần tự nối tiếp

Hệ thống tuần tự nối tiếp đó là quá trình nhiều giai đoạn xảy ra liên tiếp nhau, không đè lên nhau và kết hợp với các phương thức tác động ta có một quá trình công nghệ hoàn chỉnh. (Hình 9.1a)



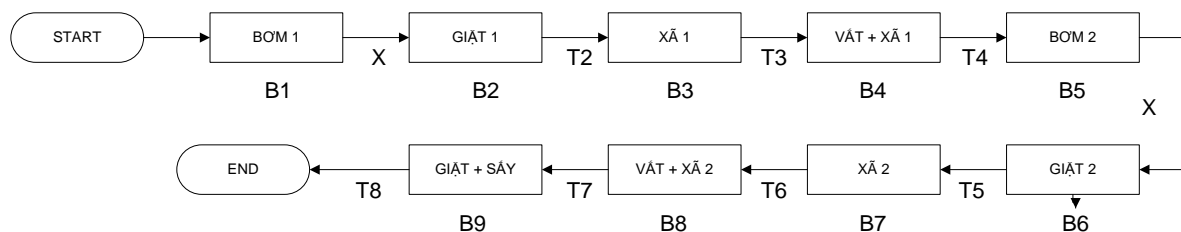
(Hình 9.1a)

Thuật toán giải quyết bài toán điều khiển quá trình tuần tự nối tiếp. Hình 9.1b:



(Hình 9.1b)

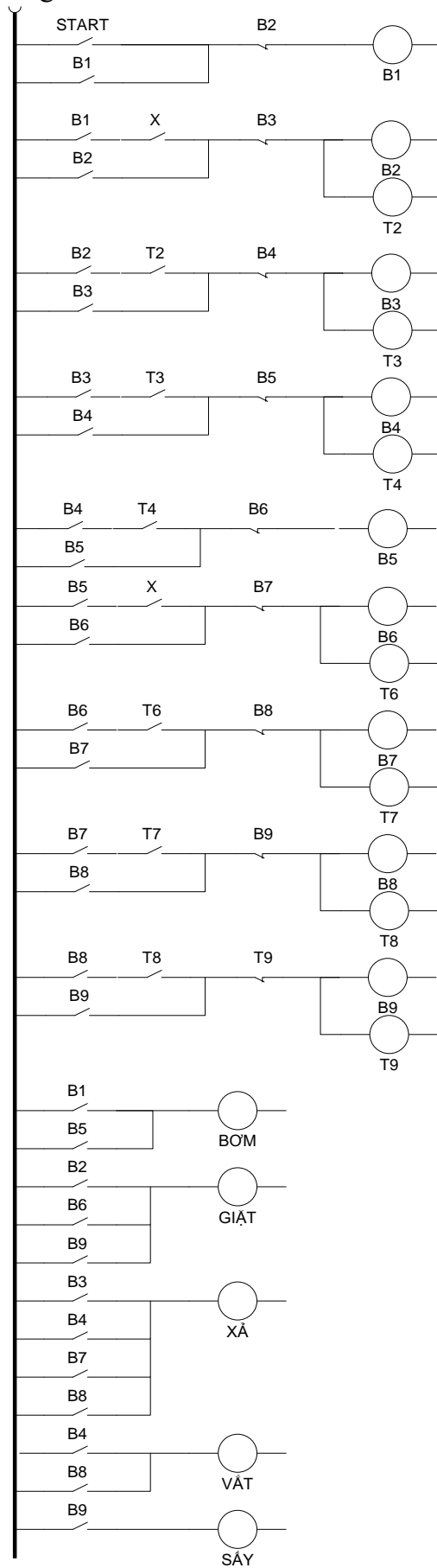
Thí dụ thực hiện một chu trình làm việc của máy giặt như sau:
VÍ DỤ



Y1-Y9: 9 giai đoạn làm việc
X: cảm biến mực nước đầy

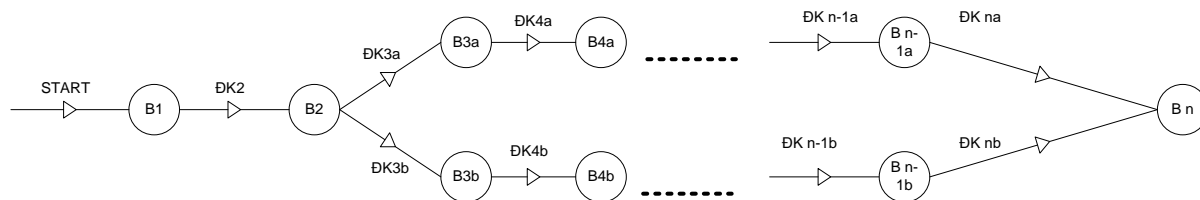
T1-T8: thời gian delay

Chương trình được viết như sau:

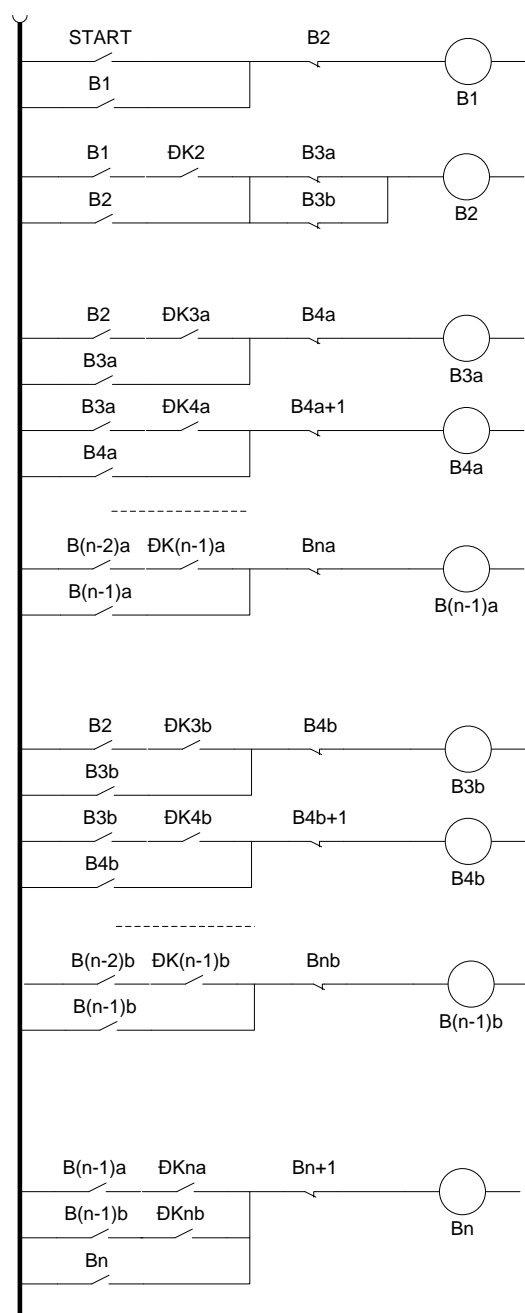


8.4.2 Hệ thống tuần tự song song

Hệ thống tuần tự song song đó là quá trình nhiều giai đoạn xảy ra đồng thời cùng nhau bắt đầu và chờ đợi nhau kết thúc. Kết hợp với các phương thức tác động ta có một quá trình công nghệ hoàn chỉnh.



Thuật toán giải quyết bài toán điều khiển quá tuần tự song song:



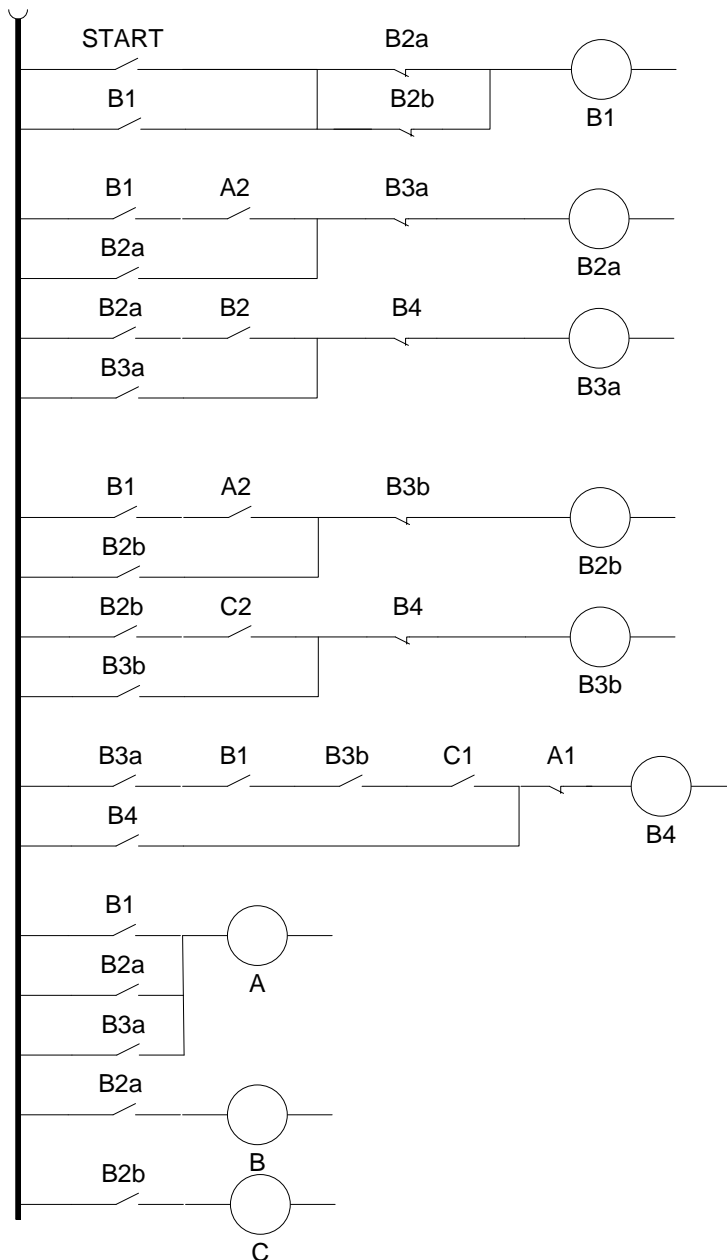
Thí dụ thực hiện chương trình sau:

$Start\ A + \left\{ \begin{matrix} B + B - \\ C + C - \end{matrix} \right\} A -$

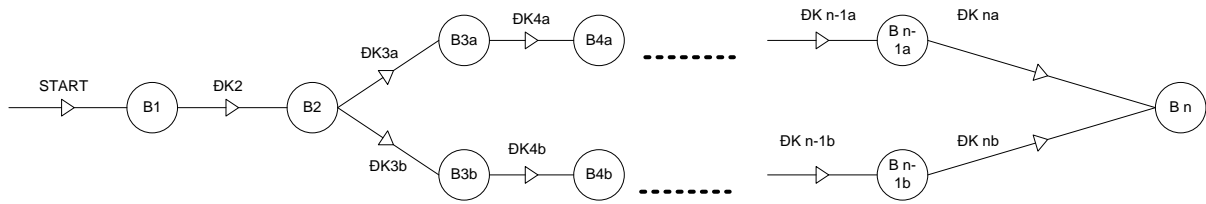
Phân chia giai đoạn

$\begin{matrix} B1 & B2a & B3a & B4 \\ Start\ A + \left\{ \begin{matrix} B + B - \\ C + C - \end{matrix} \right\} A - \end{matrix}$

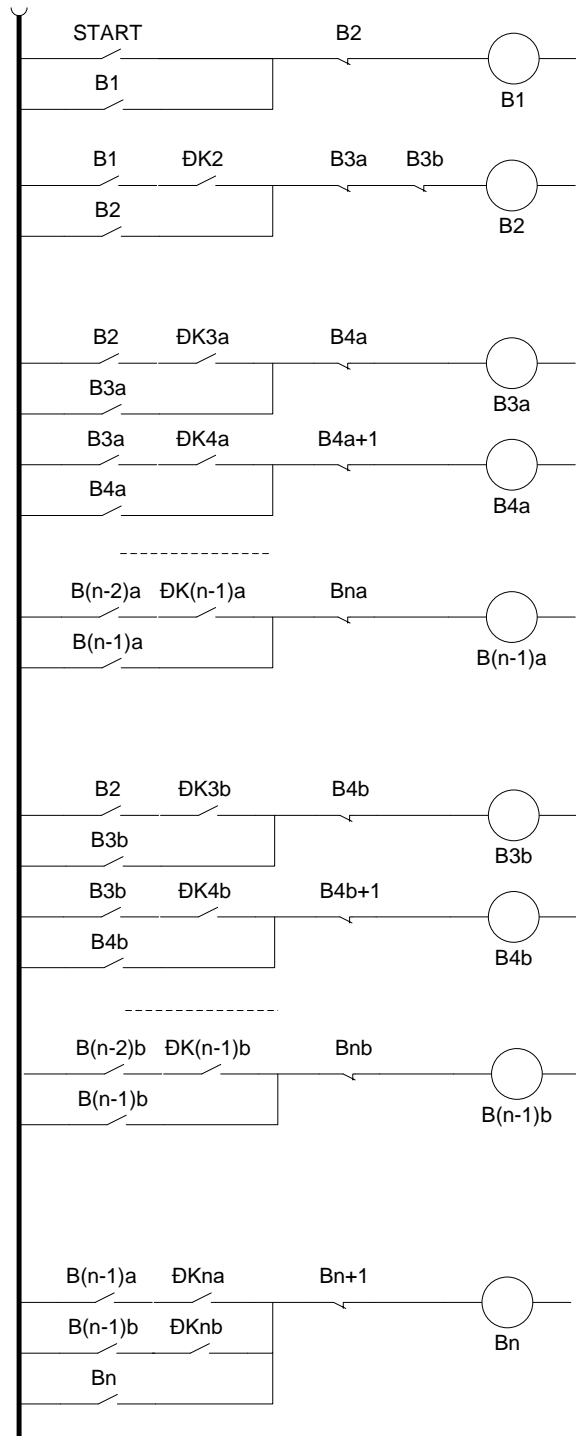
Chương trình thực hiện yêu cầu trên:



8.4.3 Hệ thống tuần tự rẽ nhánh có lựa chọn



Thuật toán giải quyết bài toán điều khiển hệ thống trình tự rẽ nhánh có lựa chọn:



Thí dụ về thực hiện quá trình tuần tự có chọn lựa:

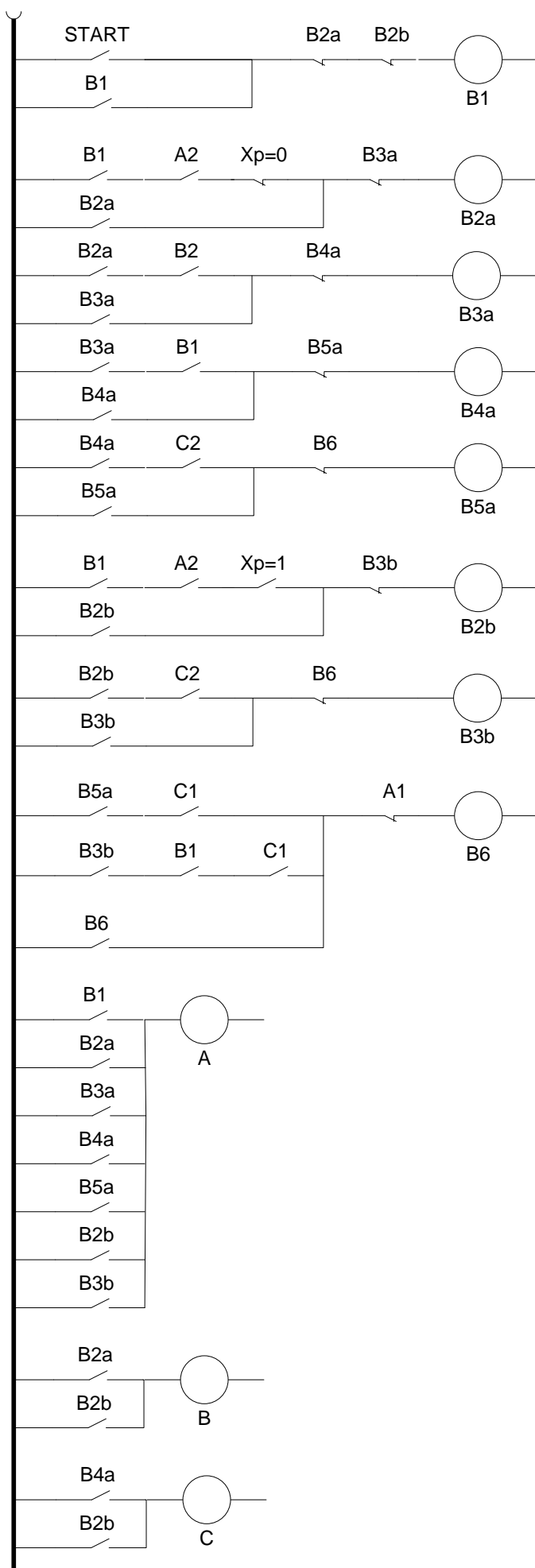
$$Start\ A+ \rightarrow \left[\begin{array}{l} \xrightarrow{Xp=0} \left\{ \begin{array}{l} B+B- \\ C+C- \end{array} \right\} \\ \xrightarrow{Xp=1} \left\{ \begin{array}{l} B+ \\ C+ \end{array} \right\} \left\{ \begin{array}{l} B- \\ C- \end{array} \right\} \end{array} \right] \rightarrow A-$$

Phân chia giai đoạn:

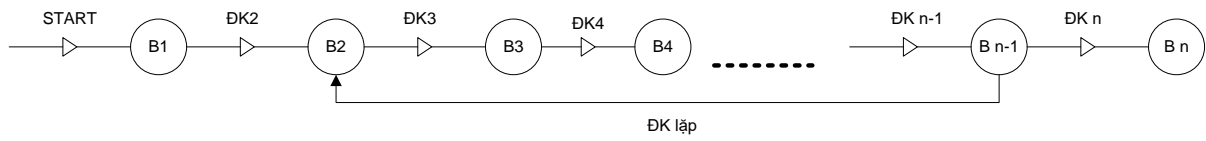
$$\begin{array}{c} \text{B1} \\ \text{Start } A+ \end{array} \xrightarrow{\text{B1}} \left[\begin{array}{l} \xrightarrow{Xp=0} \left\{ \begin{array}{l} \overset{Y}{B+} \overset{B2a}{B-} \\ \underset{Y}{C+} \overset{B3a}{C-} \end{array} \right\} \\ \xrightarrow{Xp=1} \left\{ \begin{array}{l} \overset{B4a}{B+} \overset{B5a}{B-} \\ \underset{Y_{K1}}{C+} \overset{Y_{K2}}{C-} \end{array} \right\} \end{array} \right] \xrightarrow{Y_2} \text{B6} \begin{array}{l} A- \\ \text{B6} \end{array}$$

B2b B3b

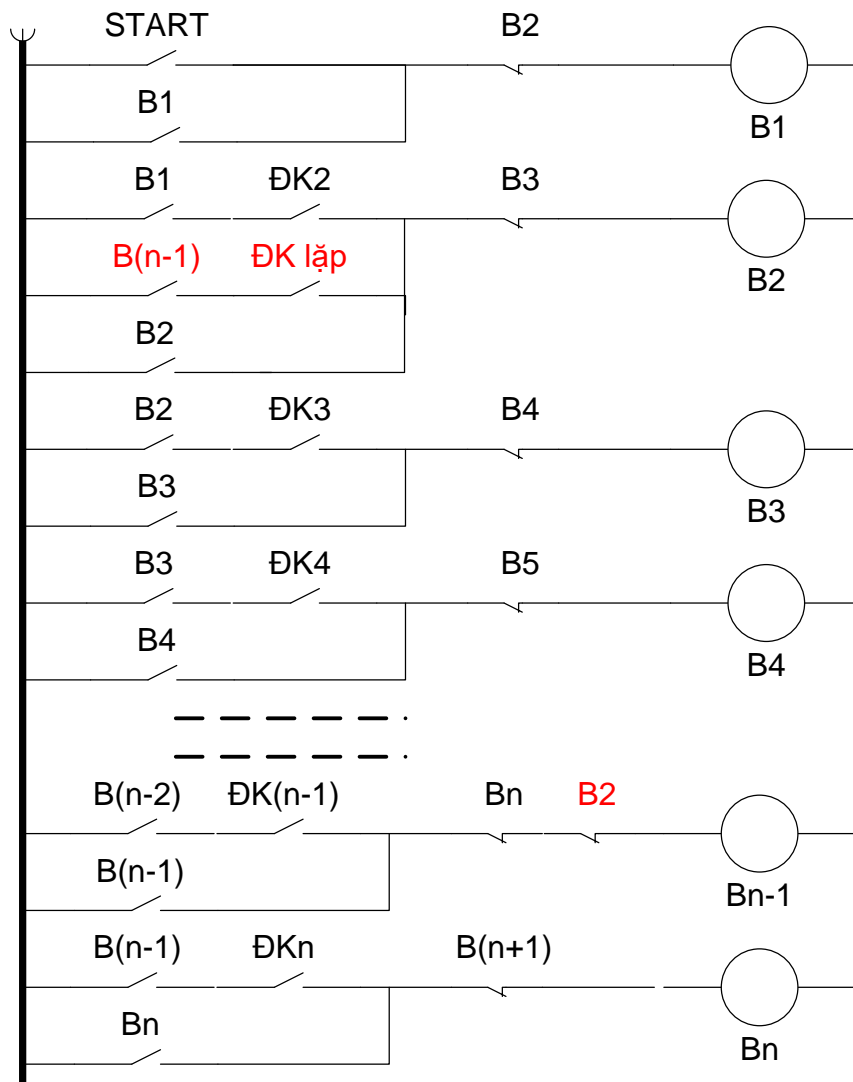
Chương trình thực hiện:



8.4.4 Hệ thống tuần tự có vòng lặp

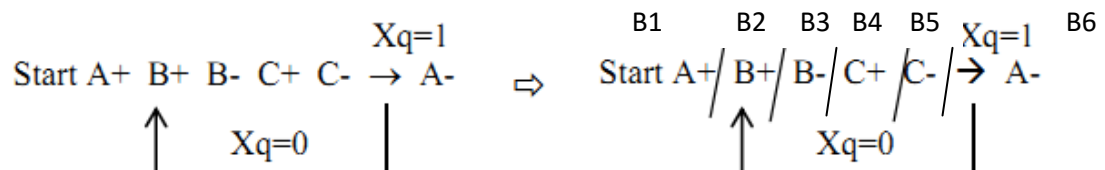


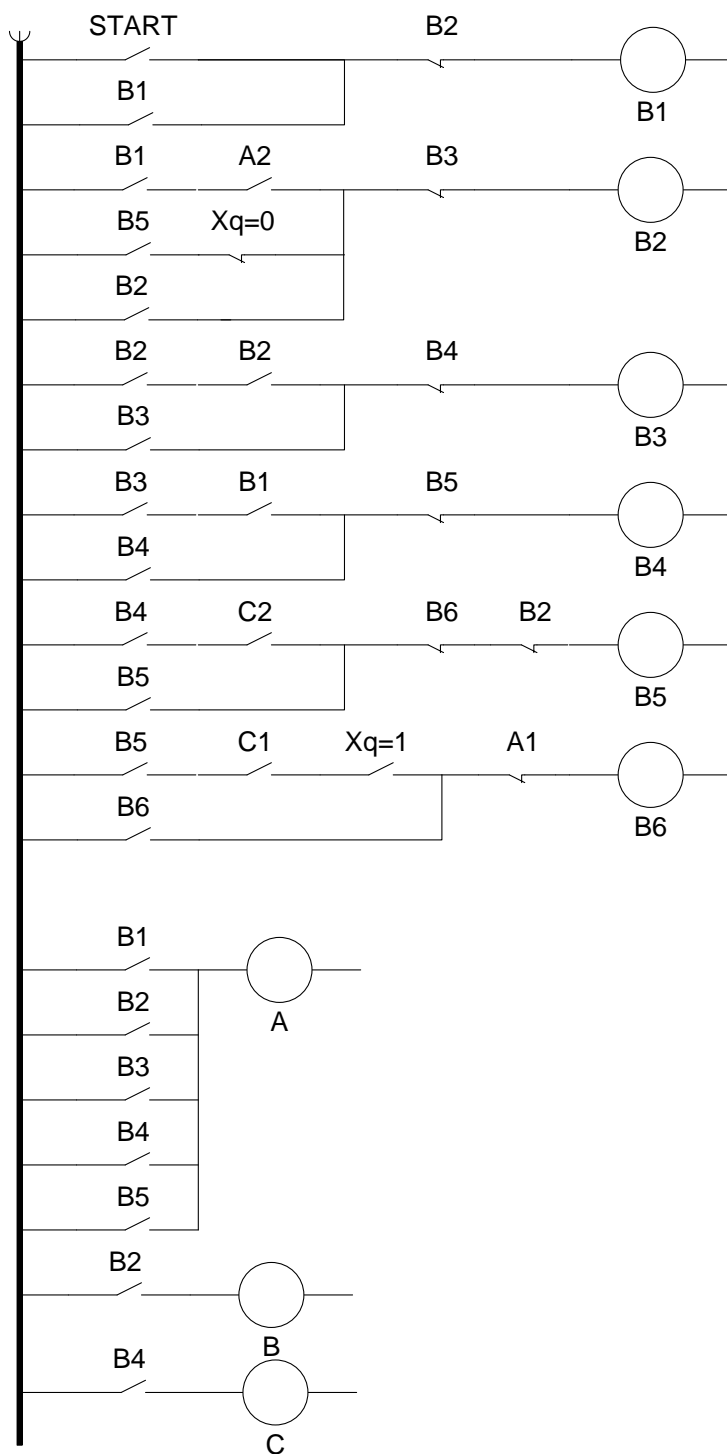
Thuật toán giải quyết bài toán điều khiển hệ thống tuần tự có vòng lặp:



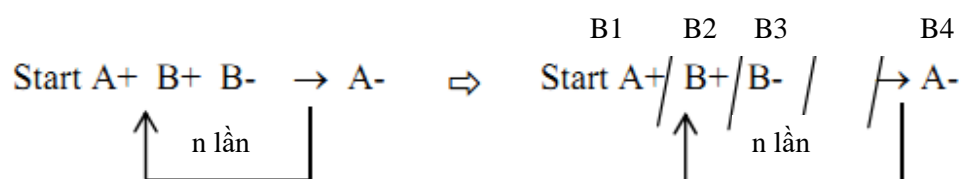
Thí dụ cho quá trình tuần tự có lặp vòng:

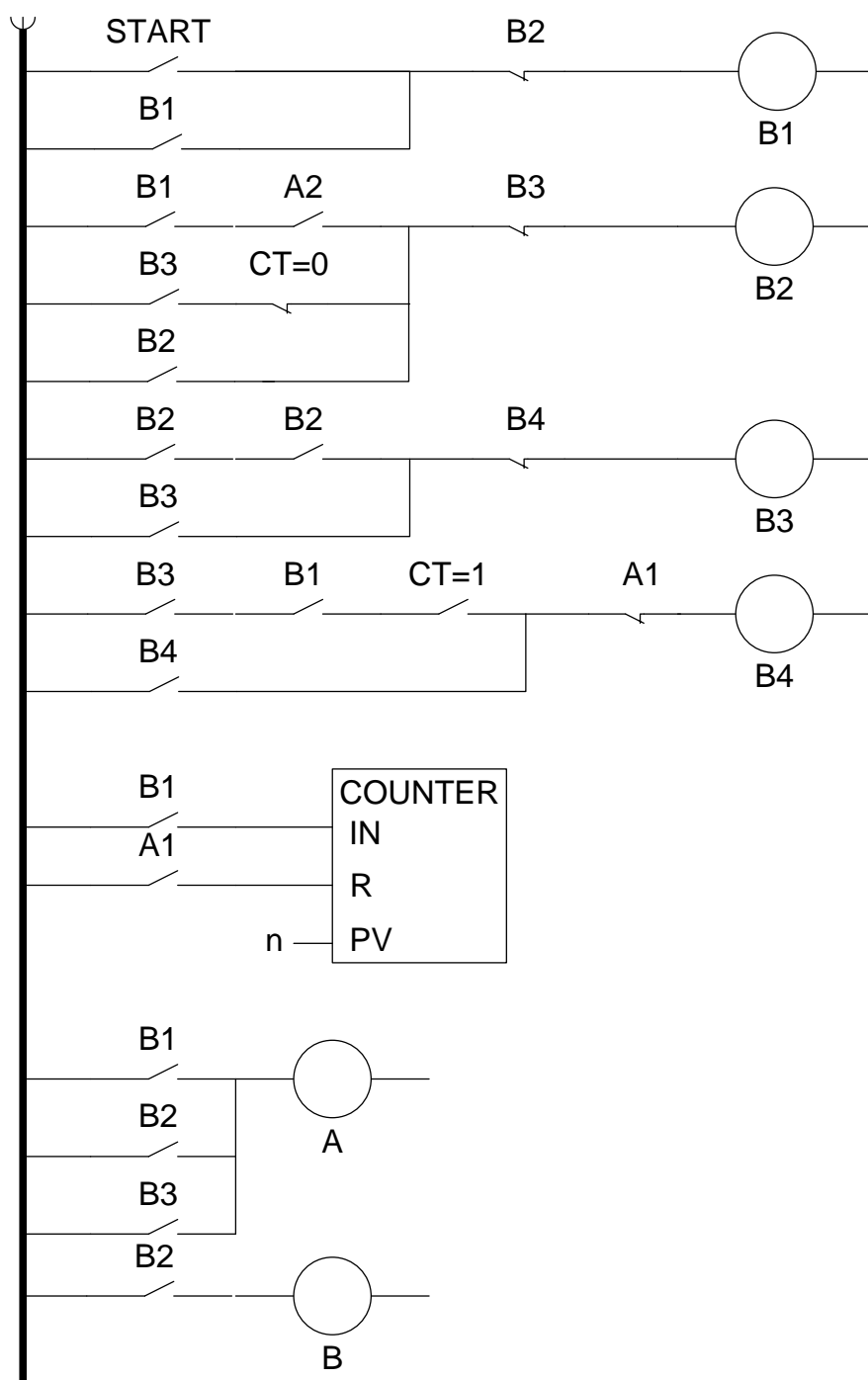
Thí dụ 1: Thực hiện quá trình sau:





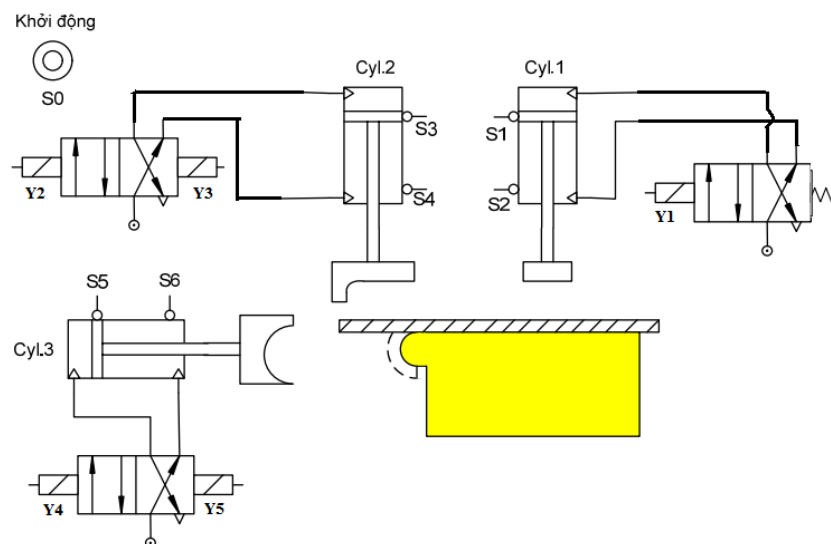
Thí dụ 2: Thực hiện quá trình sau:





8.5 Ví dụ điều khiển tuần tự:

Cho phân mạch khí nén của một hệ thống uốn ống như sau:



Yêu cầu: Thanh kim loại cần được uốn 1 đầu như hình vẽ.

Quy trình hoạt động như sau:

Ban đầu cả 3 xy lanh đều co về, thanh kim loại cần uốn được đặt sẵn lên khuôn uốn.

Nhấn nút khởi động S0, thì xy lanh Cyl.1 đi xuống để kẹp thanh kim loại. Khi thanh kim loại được kẹp chặt (được nhận biết bởi công tắc hành trình S2) thì xy lanh Cyl.2 đi xuống để uốn thanh kim loại vuông góc. Sau khi uốn xong (được nhận biết bởi công tắc hành trình S4) thì xy lanh Cyl.2 rút về.

Sau khi Cyl.2 rút về xong (được nhận biết bởi công tắc hành trình S3) thì xy lanh Cyl.3 đi ra để uốn phần cuối thanh kim loại theo định hình của khuôn. Khi xy lanh này uốn xong (được nhận biết bởi công tắc hành trình S6) thì xy lanh sẽ rút về.

Khi xy lanh Cyl.3 rút về hết hành trình (được nhận biết bởi công tắc hành trình S5) thì xy lanh kẹp thanh kim loại Cyl.1 sẽ rút về để nhả thanh kim loại ra.

Khi xy lanh Cyl.1 hoàn thành quá trình rút về (được nhận biết bởi công tắc hành trình S1) thì quá trình uốn kết thúc. Người dùng có thể lấy thanh kim loại ra và thay một thanh mới vào để bắt đầu lại quy trình.

Các thao tác trong quy trình này được thực hiện một cách tuần tự, xong việc này thì tới việc kia. Kẹp xong thanh kim loại thì uốn vuông góc, uốn xong thì rút về, rút về xong thì uốn định hình....

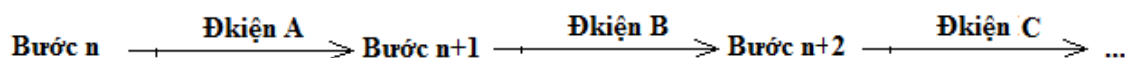
Đây chính là một bài toán điều khiển tuần tự

8.5. 1 Quy trình thực hiện:

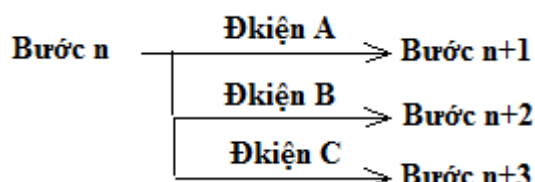
- ✓ Từ yêu cầu hoạt động máy, hình dung và phân tích ra một trình tự các thao tác thật chi tiết của các khâu chấp hành cũng như thời điểm bắt đầu, và kết thúc một thao tác. Mô tả quy trình đó bằng lời sao cho thật cụ thể và chi tiết.
- ✓ Tóm tắt quy trình điều khiển dưới dạng sơ đồ: thông thường, một yêu cầu điều khiển được mô tả bằng lời (như ví dụ ở trên) sẽ phải được tóm tắt thành một sơ đồ đơn giản để thuận tiện cho việc lập trình.

Sơ đồ gồm nhiều bước, mỗi bước thực hiện 1 thao tác hoặc nhiều thao tác cùng lúc với nhau. Các bước được nối với nhau bằng các mũi tên theo trình tự từ trước tới sau (theo thời gian). Phía trên các mũi tên nối giữa hai bước phải ghi rõ những điều kiện để chuyển từ bước trước ra bước sau.

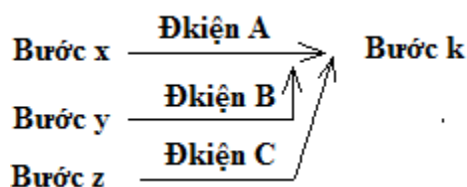
Một đoạn sơ đồ thẳng (**không rẽ nhánh**) được biểu diễn dưới đây.



Từ một bước trước, kết hợp với 2 hoặc nhiều điều kiện khác nhau sẽ có thể chuyển thành 2 hoặc nhiều bước khác nhau (**rẽ nhánh**)



Từ hai hay nhiều bước trước, kết hợp với nhiều điều kiện khác nhau sẽ có thể chuyển về chung 1 bước (**nhập nhánh**)



Như vậy, một sơ đồ đầy đủ sẽ có các bước đi thẳng, các bước rẽ nhánh, các bước nhập nhánh.

Trong ví dụ trên, chúng ta sẽ có 6 bước: 1- kẹp thanh kim loại; 2- uốn vuông góc; 3- rút xy lanh uốn vuông góc về; 4- uốn định hình; 5- rút xy lanh uốn định hình về; 6- nhả thanh kim loại.

Điều kiện để chuyển từ trạng thái đứng yên sang bước 1- kẹp thanh kim loại là nhấn nút S0.

Điều kiện để chuyển từ bước 1- kẹp thanh kim loại sang bước 2- uốn vuông góc là sự tác động của công tắc hành trình S2.

Điều kiện để chuyển từ bước 2- uốn vuông góc sang bước 3- rút xy lanh uốn vuông góc về là sự tác động của công tắc hành trình S4.

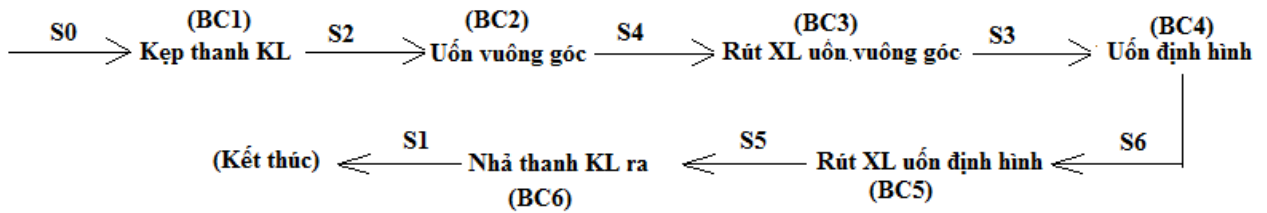
Điều kiện để chuyển từ bước 3- rút xy lanh uốn vuông góc về sang bước 4- uốn định hình là sự tác động của công tắc hành trình S3.

Điều kiện để chuyển từ bước 4- uốn định hình sang bước 5- rút xy lanh uốn định hình về là sự tác động của công tắc hành trình S6.

Điều kiện để chuyển từ bước 5- rút xy lanh uốn định hình về sang bước 6- nhả thanh kim loại là sự tác động của công tắc hành trình S5.

Điều kiện để chuyển từ bước 6- nhả thanh kim loại sang kết thúc quy trình là sự tác động của công tắc hành trình S1.

Rút lại, chúng ta có sơ đồ biểu diễn quy trình trên như sau:

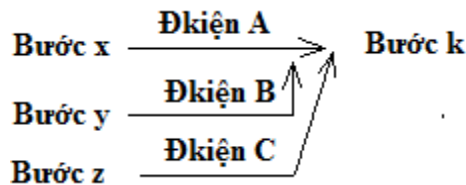


✓ Lập trình chia bước

Để chia bước trong phương pháp lập trình tuần tự, ta có nhiều cách: dùng tiếp điểm duy trì, lệnh SR, RS, MOV... Trong tài liệu này hướng dẫn một trong những phương pháp đó là dùng lệnh MOV.

Bước hiện tại của quy trình sẽ được lưu vào một biến kiểu Byte trên PLC. Ví dụ VB0, MB5... Biến này có thể sử dụng để điều khiển 1 quy trình lên đến 255 bước.

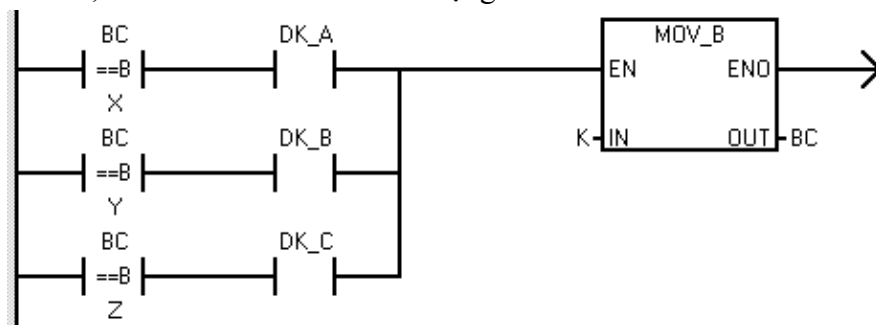
Trong mỗi Network chia bước, ta chỉ cần quan tâm đến điều kiện bắt đầu của bước đó. Ví dụ với sơ đồ như hình vẽ sau:



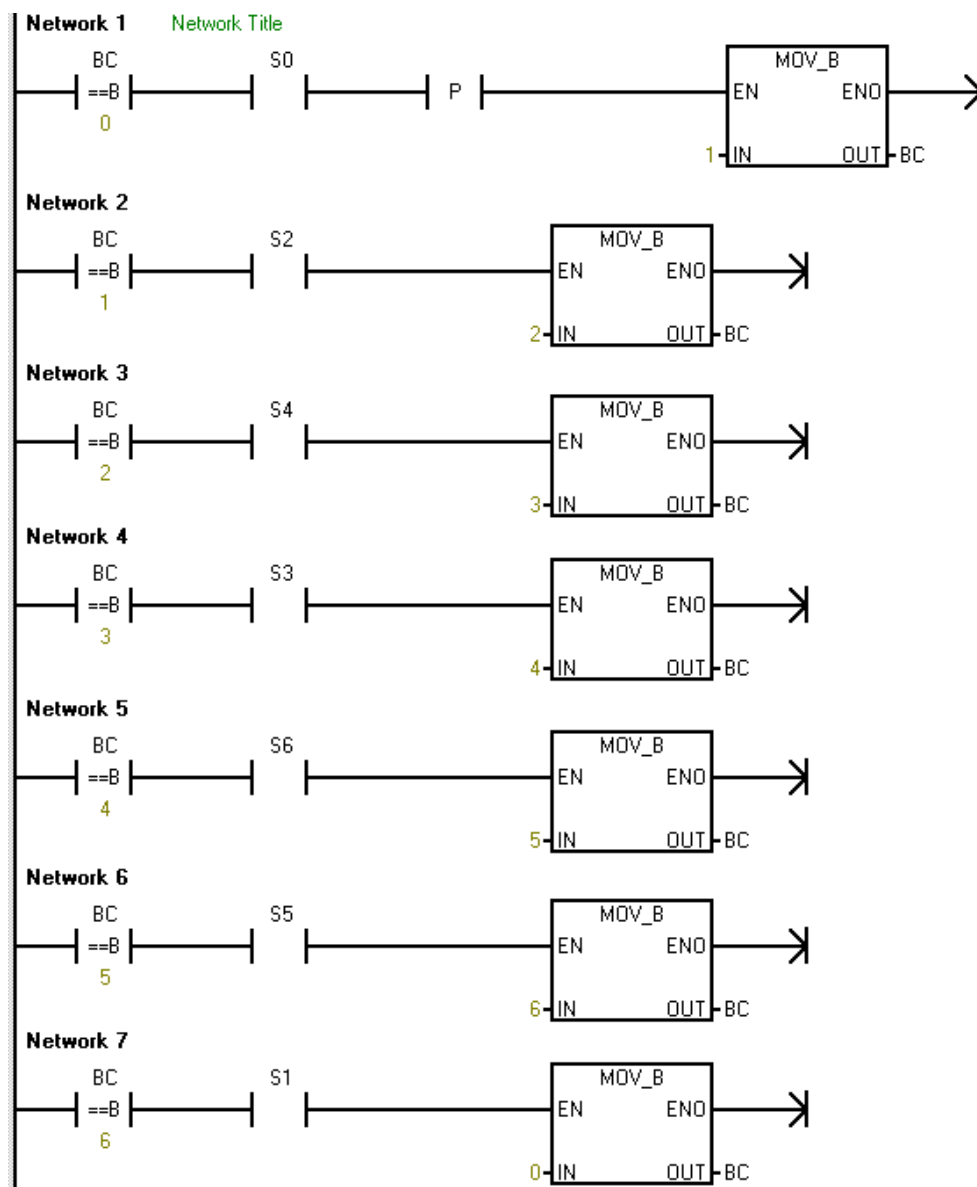
Khi chia bước cho bước k, ta thấy có 3 điều kiện để bắt đầu bước k. Đó là

- Đang ở bước x gặp điều kiện A
- Hoặc đang ở bước y gặp điều kiện B
- Hoặc đang ở bước z gặp điều kiện C.

Khi đó, Network chia bước sẽ có dạng:



Với ví dụ trên, 6 NW chia bước cho 6 bước như sau:

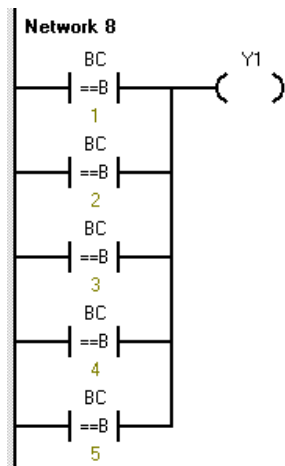


✓ Tổng hợp điều khiển ngõ ra:

Hệ thống có bao nhiêu ngõ ra thì sẽ dùng bấy nhiêu Network để điều khiển.

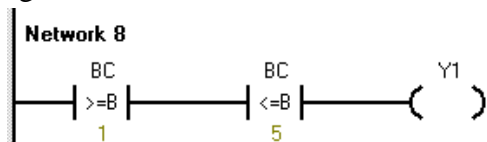
Trong quy trình có bao nhiêu bước tác động đến 1 ngõ ra thì cần tập trung hết bấy nhiêu bước đó (nối song song với nhau) để điều khiển ngõ ra đó.

Ví dụ bước 1, bước 2, bước 3, bước 4, bước 5 cùng tác động đến ngõ ra Y1 thì ta sẽ có chương trình

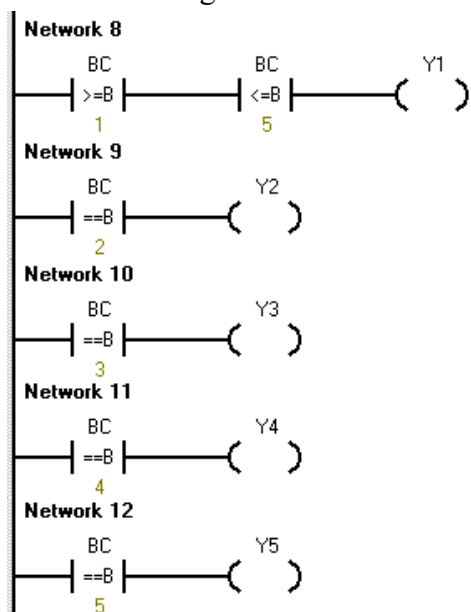


Trong trường hợp có nhiều bước “liên tiếp nhau” cùng tác động đến 1 ngõ ra thì cũng có thể dùng so sánh để điều khiển ngõ ra.



Như trường hợp tất cả các bước liên tiếp nhau từ bước 1 đến bước 5 đều tác động ngõ ra Y1.

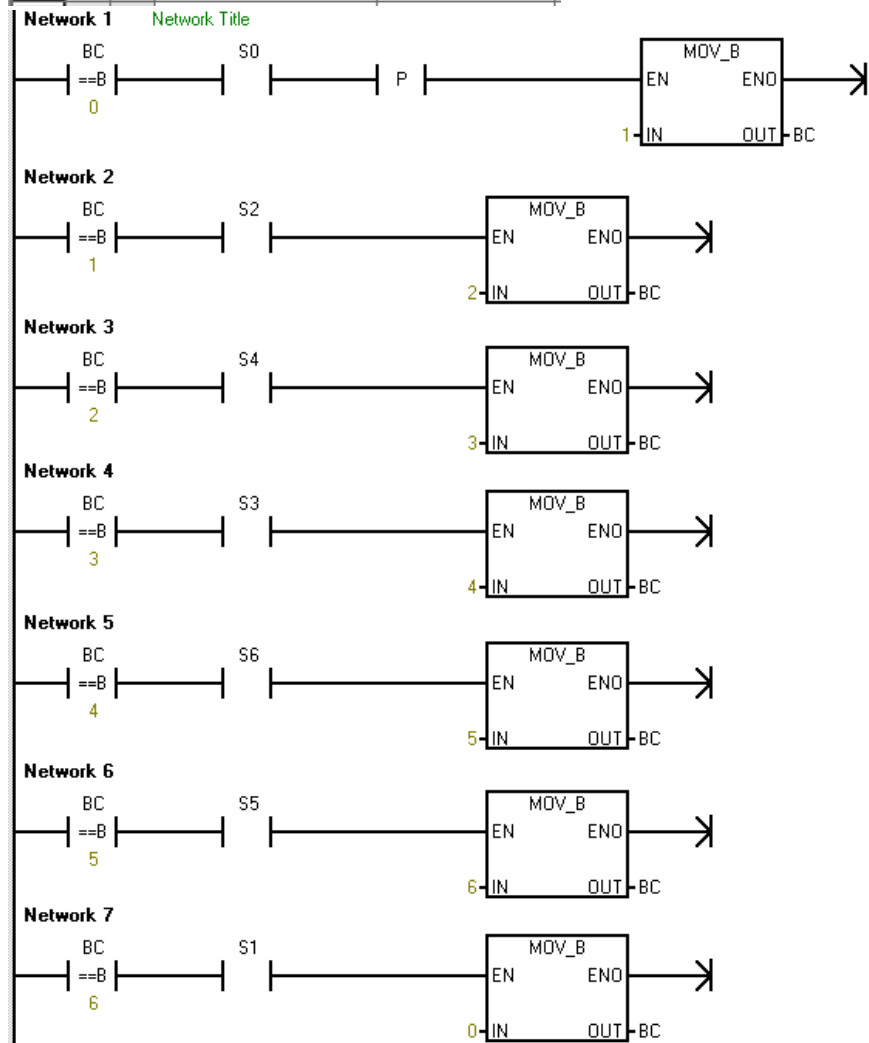


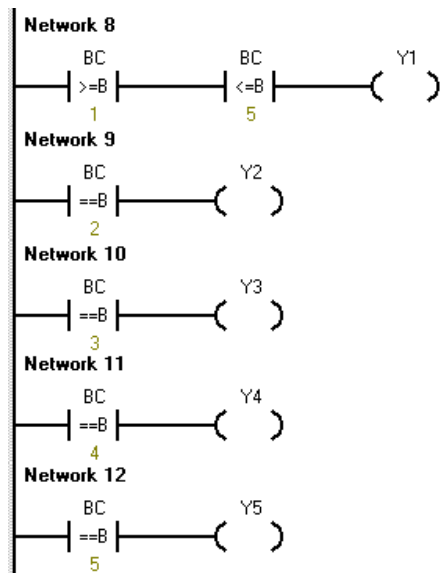
Theo như ví dụ trên, ta có 5 Network (vì có 5 ngõ ra điều khiển 5 cuộn Solenoid) để điều khiển ngõ ra như sau:



Vậy bảng đặt tên biến và toàn bộ chương trình điều khiển của ví dụ uốn thanh kim loại đã cho như sau:

			Symbol	Address
1			BC	VB0
2			S1	I0.0
3			S2	I0.1
4			S3	I0.2
5			S4	I0.3
6			S5	I0.4
7			S6	I0.5
8			Y1	Q0.0
9			Y2	Q0.1
10			Y3	Q0.2
11			Y4	Q0.3
12			Y5	Q0.4
13			S0	I0.6





CHƯƠNG 9: CÁC CƠ CẤU TỰ ĐỘNG CƠ KHÍ

9.1 Cơ cấu cấp phôi tự động

Ý nghĩa: Cấp phôi là quá trình chuyển phôi từ ổ chứa phôi qua máng dẫn và từ một số bộ phận khác tới vị trí gia công. Việc cấp phôi có ý nghĩa to lớn sau:

- Biến máy bán tự động thành máy tự động.
- Dây chuyền sản xuất thành đường dây tự động.
- Mang lại hiệu quả kinh tế nhờ giảm tổn thất về thời gian.
- Cải thiện điều kiện làm việc của công nhân, đặc biệt trong môi trường độc hại, nhiệt độ cao, phôi có trọng lượng lớn

Dựa vào dạng phôi ta có thể chia làm 3 loại:

HT cấp phôi cuộn:

Phôi cuộn là phôi dạng cuộn thép tròn có đường kính nhỏ hoặc những lá thép mỏng được cuộn tròn vào tang. Các cơ cấu cấp phôi cho dạng phôi này bắt buộc phải có cơ cấu kéo --- nắn thẳng phôi --- và cắt thoi theo dung chiều dài yêu cầu.

HT cấp phôi dạng thanh

Phôi thanh là dạng phôi tròn hoặc vuông đã được nắn thẳng, có chiều dài phôi lớn hơn đường kính phôi (hoặc 1 cạnh của mặt cắt hình vuông) từ 20 lần trở lên ($L \geq 20d$). Có 2 phương pháp cấp phôi dạng này:

- Dùng tải trọng để đẩy phôi tới cử chặn (có thể dùng xy lanh khí nén thủy lực, động cơ kéo cơ cấu cam ...)
- Dùng châu phóng phôi (có thể dùng châu kép và dùng cơ cấu tịnh tiến phóng phôi)

HT cấp phôi rời từng chiếc

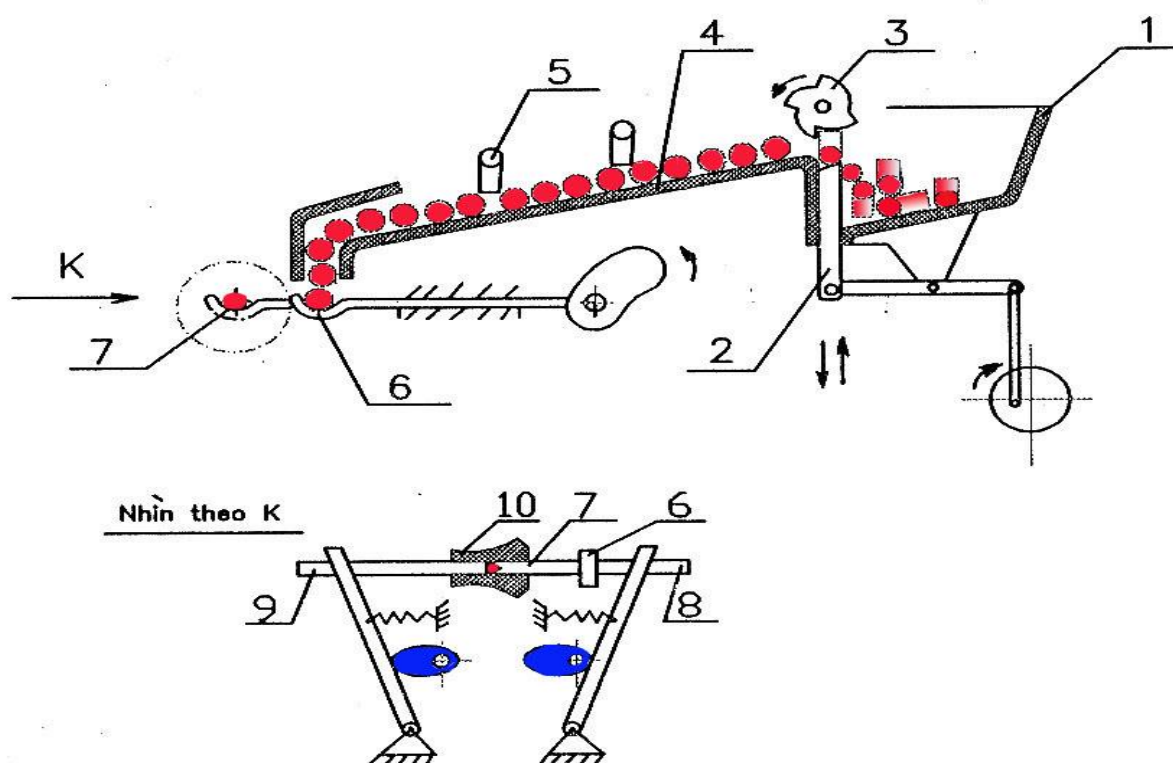
Phôi rời có 3 dạng chủ yếu:

- Chi tiết có trọng lượng lớn có mặt cắt dạng hình chữ nhật hoặc hình vuông. Các chi tiết có trọng lượng lớn cho nên cơ cấu cấp phôi không thể dùng cơ cấu phóng hay dạng phễu, ổ chứa mà dùng cơ cấu có dạng dự trữ chờ phôi.
- Chi tiết có trọng lượng lớn có mặt cắt là hình tròn, cơ cấu cấp phôi cho tiết dạng này thông thường được sắp xếp trong ổ chứa theo chiều dọc chiều dài phôi và có thể dùng cơ cấu xy lanh khí nén thủy lực hoặc cơ cấu tách thoi đơn chiếc, theo trọng lượng và chiều dài phôi sẽ lần xuống vị trí mong muốn.
- Chi tiết nhỏ, hình dáng đơn giản dùng chế tạo các chi tiết tiêu chuẩn như: bu long, đai ốc, chốt trụ, bi bạc, bạc lót ... Với các chi tiết này chúng ta dùng cơ cấu bằng phễu rung và máng dẫn

HT cấp phôi rời dạng tấm

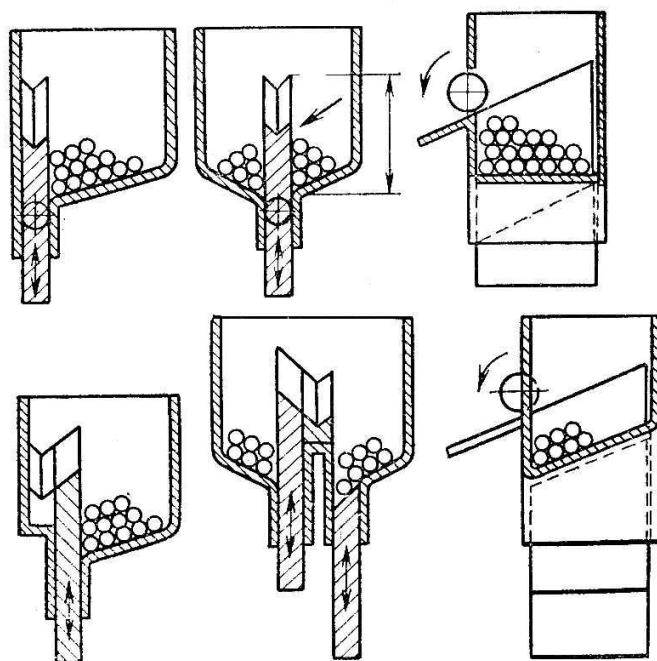
Những phôi dạng tấm như tấm thép tấm, gỗ dạng tấm, kính dạng tấm, đá granite dạng tấm. Trong các dạng phôi này thì nếu phôi nào không cần gia công bề mặt (đã đạt yêu cầu như kính, mặt đá granite) thì chúng ta cần tránh cơ cấu có dạng các chi tiết chồng lên nhau, và chúng ta có thể dùng giác hút chân không để lấy phôi. Còn đối với sắt tấm thì chúng ta có thể dùng nam châm điện để lấy phôi.

Một số cơ cấu cấp phôi

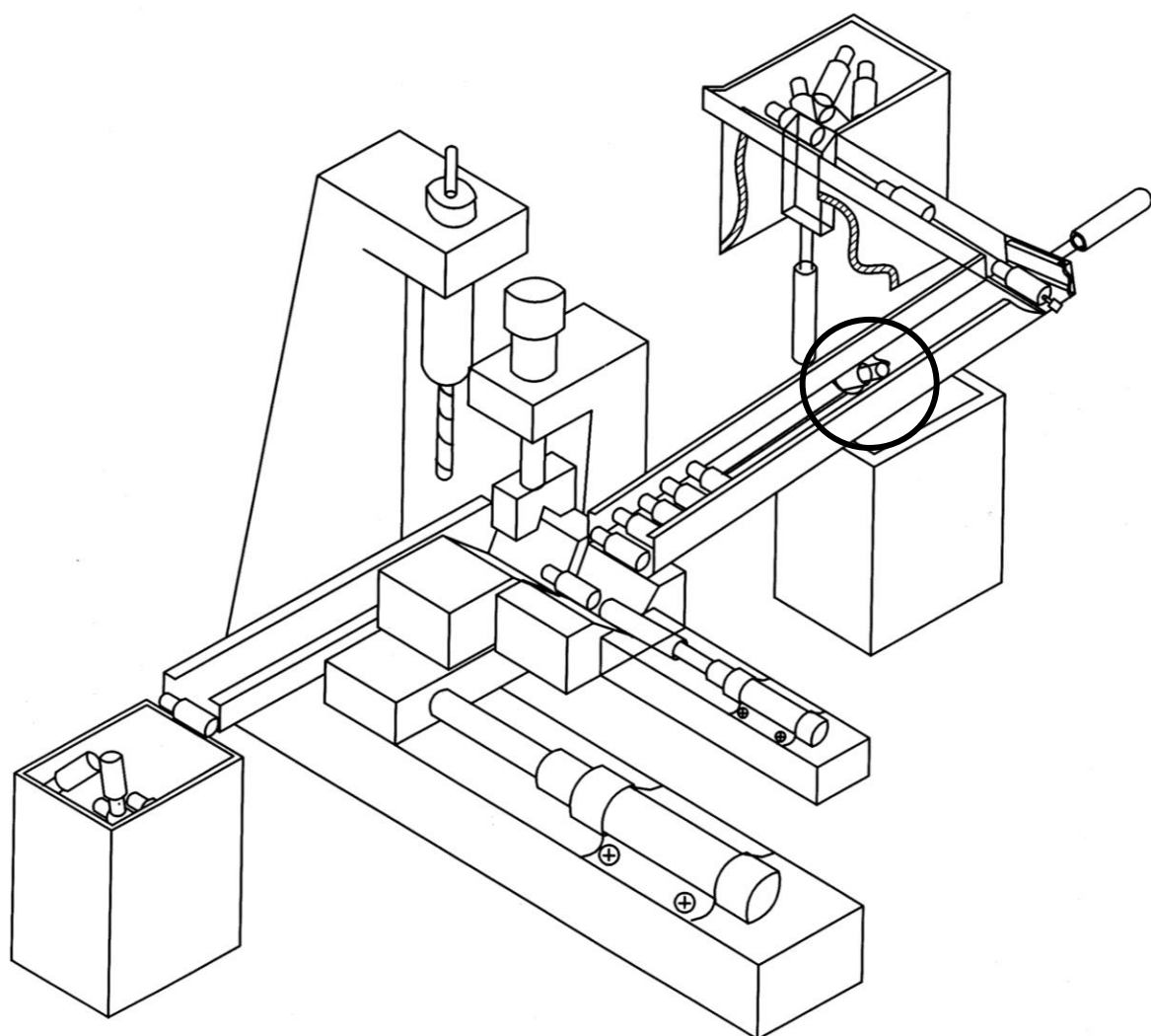


Hình 4.1: Cơ cấu cấp phôi dùng cho chi tiết dạng phôi trụ đối xứng nhau

- 1 – Phễu (ô) chứa phôi
- 2 – Cơ cấu đưa phôi
- 3 – Cơ cấu gạt
- 4 – Băng dẫn phôi
- 5 – Cơ cấu giảm tốc độ
- 6 – Cơ cấu kẹp
- 7 – Phôi

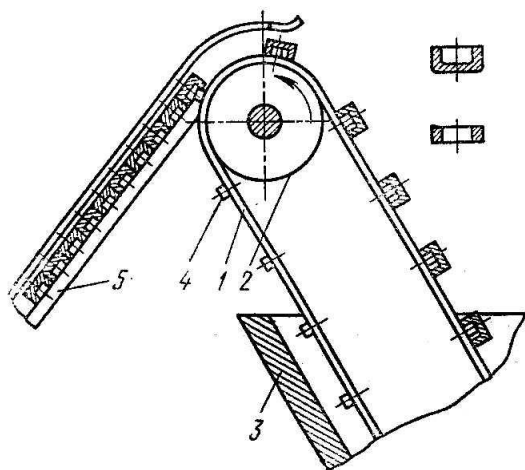


Hình 4.2: Cơ cấu cấp phôi dùng cho chi tiết dạng phôi trụ đối xứng nhau

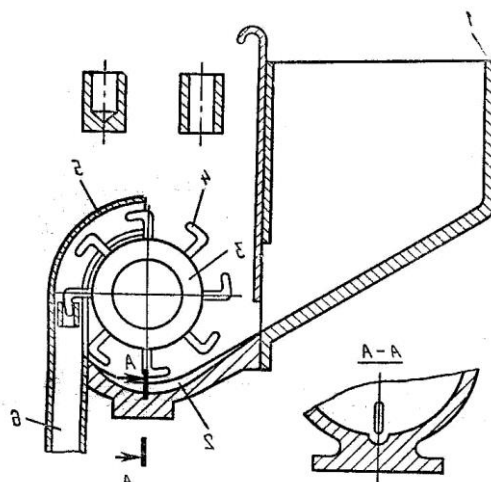


Hình 4.3: Cơ cấu dùng cho chi tiết dạng phôi trụ không đối xứng nhau

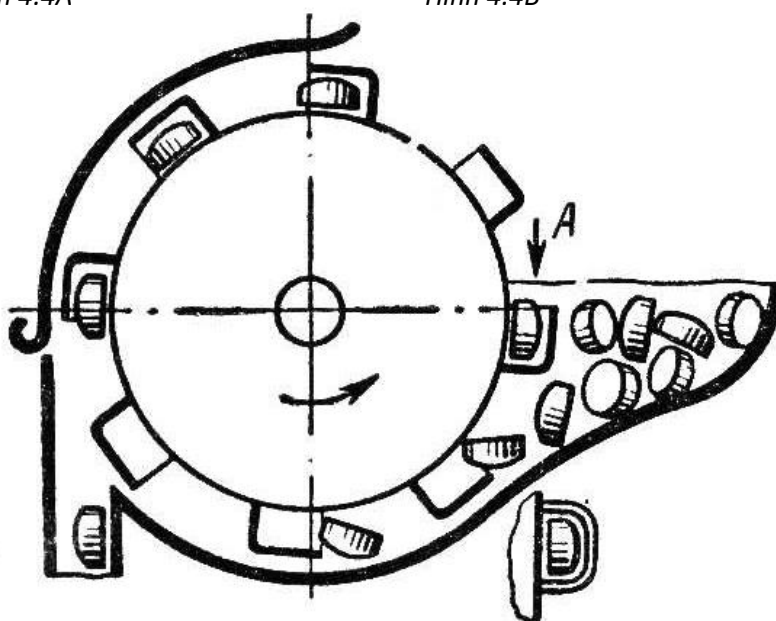
Với những chi tiết dạng phôi trụ không đối xứng nhau, chúng ta có thể phân loại theo chiều bằng cách xẻ rãnh trên máng dẫn, và dựa vào trọng lượng của phôi sẽ rút xuống hoặc được giữ lại trên máng theo hình trên.



Hình 4.4A



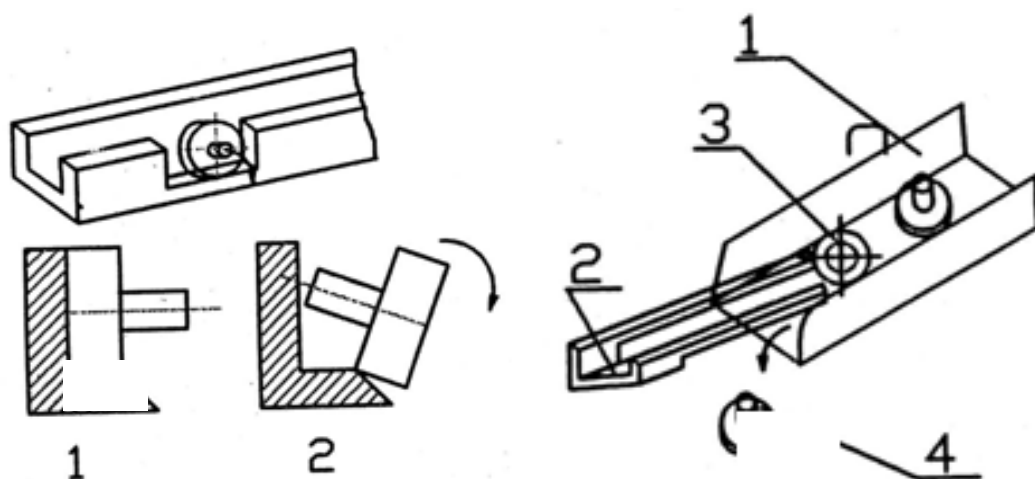
Hình 4.4B



Hình 4.4C

Hình 4.4A, 4.4B và 4.4C là cơ cấu dùng cho các chi tiết dạng trụ rỗng hoặc có lỗ bậc, nên cơ cấu lấy phôi chúng ta dùng dạng móc.

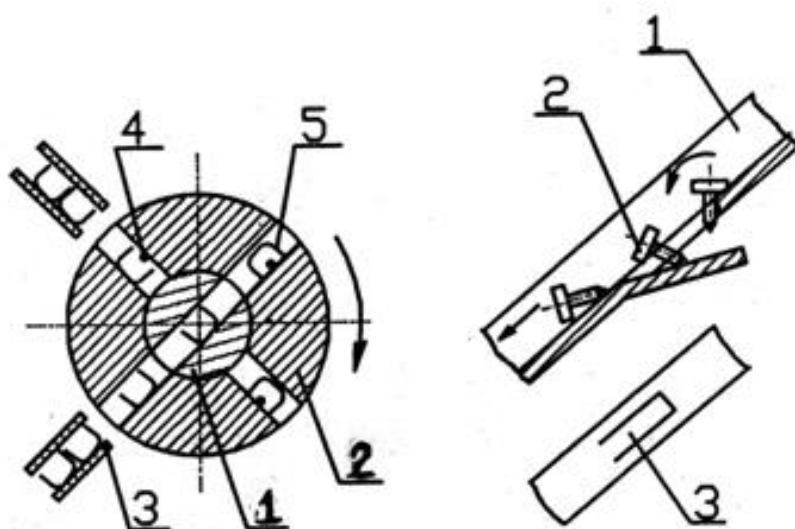
Bên cạnh các cơ cấu cấp phôi thì vấn đề sửa phôi sai hay định hướng phôi, sửa phôi sai rất quan trọng và không thể thiếu đối với loại phôi có hình dạng không đối xứng. Một số cơ cấu sửa phôi sai:

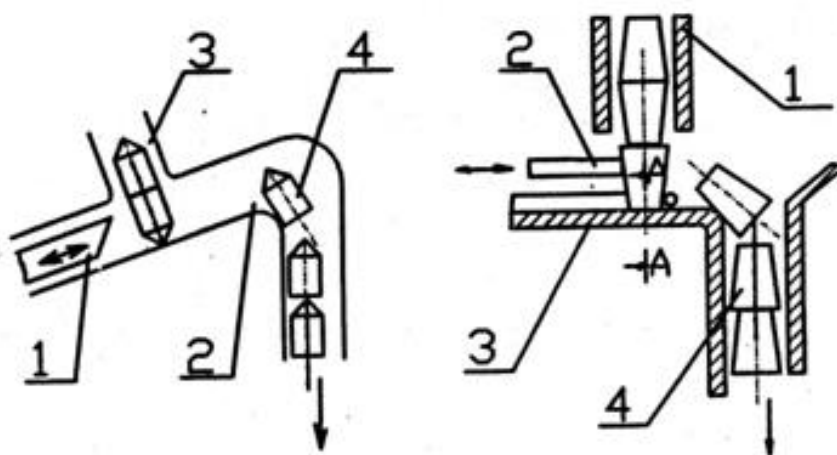


Hình 4.5A

Hình 4.5B

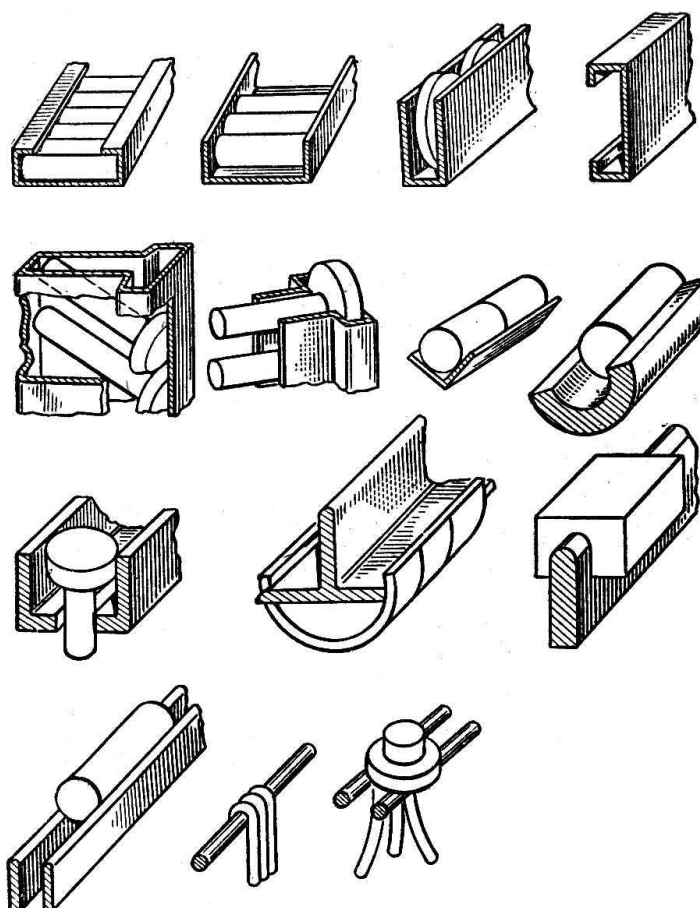
Chúng ta dễ dàng nhận thấy cơ cấu sửa sai được thiết kế theo hình 4.5A, chi tiết hình trụ bậc với phần trọng lượng nặng hơn nằm phía ngoài sẽ được loại ra tương tự đối với hình 4.5B phần to hơn (nặng hơn) nằm phía dưới rãnh. Và các cơ cấu tương tự cùng chức năng theo hình 4.6





Hình 4.6: Một số cơ cấu sửa phôi sai

Ngoài cơ cấu sửa sai hình việc thiết kế máng dẫn đóng vai trò rất quan trọng, quyết định sự ổn định cũng như độ chính xác của cơ cấu.



Hình 4.7: Một số cơ cấu, máng dẫn, định hướng phôi

9.2 Bài tập ứng dụng

9.2.1 Thiết kế cơ cấu cấp phối tự động

Thiết kế cơ cấu cấp phối cho máy đột lỗ phi 6.

Kích thước trước gia công: $D = 40\text{mm}$, chiều dày $t = 4$

Kích thước sau gia công: $D = 40\text{mm}$, $d = 6\text{mm}$, chiều dày $t = 4$

Vật liệu: thép

Thiết kế cơ cấu cấp phối trụ bậc cho máy gia công tiện ren

Kích thước: $D_1 = 50\text{mm}$, $L_1 = 12\text{mm}$, $d_1 = 30\text{mm}$, $l_1 = 40\text{mm}$,

Vật liệu: thép

Thiết kế cơ cấu cấp phối dạng cuộn cho máy uốn định hình

Kích thước trước gia công: $d_1 = 8\text{mm}$, dạng cuộn

Kích thước yêu cầu khi cắt trước khi uốn: $d_1 = 8\text{mm}$, $l = 400\text{mm}$

Vật liệu: thép

Thiết kế cơ cấu cấp phối dạng tấm cho máy chấn

Kích thước: $a = 250\text{mm}$, $b = 450\text{mm}$, $t = 4\text{mm}$

Vật liệu: thép

Thiết kế cơ cấu cấp phối dạng tấm cho máy cắt kính

Kích thước: $a = 2440\text{mm}$, $b = 3660\text{mm}$, $t = 8\text{mm}$

Vật liệu: kính

9.2.2 Điều khiển cơ cấu cấp phối tự động

Thiết kế hệ thống điều khiển cho cơ cấu cấp phối tự động theo hình 4.3

Lưu ý:

- Theo quy trình công nghệ các xylanh được đặt tên theo thứ tự A, B, C, D, E
- Có thể sử dụng công tắc hành trình, cử từ hoặc cảm biến để xác định hành trình của xylanh ...
- Có thể sử dụng van đảo chiều 1 cuộn coil (tác động 1 phía) hoặc 2 cuộn coil (tác động 2 phía) để điều khiển xylanh

a – Lập sơ đồ hành trình bước (quy trình công nghệ)

b – Vẽ thêm các vị trí cảm biến, công tắc hành trình cho hệ thống

c – Vẽ sơ đồ kết nối PLC (tín hiệu đầu vào / đầu ra)

d – Viết chương trình điều khiển cho hệ thống