

ỦY BAN NHÂN DÂN THÀNH PHỐ HÀ NỘI
TRƯỜNG CAO ĐẲNG NGHỀ VIỆT NAM - HÀN QUỐC THÀNH PHỐ HÀ NỘI

TẠ VĂN BẰNG (Chủ biên)
BÙI VĂN CÔNG – LƯU HUY HẠNH



GIÁO TRÌNH LẬP TRÌNH PLC CƠ BẢN

Nghề: Cơ điện tử

Trình độ: Trung cấp

(Lưu hành nội bộ)

Hà Nội - Năm 2019

LỜI NÓI ĐẦU

Giáo trình Lập trình PLC cơ bản được biên soạn nhằm đáp ứng nhu cầu giảng dạy và học tập của Giảng viên, Sinh viên trường Cao đẳng nghề Việt Nam – Hàn Quốc thành phố Hà Nội. Nội dung giáo trình mang tính lôgic về kiến thức của toàn bộ chương trình đào tạo, đồng thời hướng tới mục tiêu hình thành và phát triển năng lực thực hiện hoạt động nghề nghiệp cho người học. Dạy học tích hợp được lựa chọn trong giáo trình nhằm tạo ra các tình huống liên kết tri thức các môn học, đó là cơ hội phát triển các năng lực của sinh viên. Khi xây dựng các tình huống vận dụng kiến thức người học sẽ phát huy được năng lực tự lực, phát triển tư duy sáng tạo (kiến thức, kỹ năng, và thái độ nghề nghiệp). Giáo trình được trình bày với 4 bài, đi từ lý thuyết cơ sở đến thực hành những kiến thức cơ bản. Đặc biệt trong nội dung giáo trình đã giới thiệu được những nội dung thực hành cơ bản của lĩnh vực Lập trình PLC, đi từ các kiến thức cơ bản của mô đun lập trình PLC .

Mặc dù nhóm biên soạn đã cố gắng phát triển giáo trình sao cho phù hợp và hiệu quả nhất với sinh viên trung cấp nghề Cơ điện tử, nhưng chắc chắn vẫn còn nhiều thiếu sót.

Chúng tôi mong nhận được những ý kiến đóng góp của bạn đọc và đồng nghiệp để giáo trình hoàn thiện hơn.

Xin chân thành cảm ơn!

Hà Nội, ngày..... tháng....năm 2019

Chủ biên: Tạ Văn Bằng

MỤC LỤC

LỜI NÓI ĐẦU	1
MỤC LỤC	2
GIÁO TRÌNH LẬP TRÌNH PLC CƠ BẢN	3
Bài mở đầu	6
Bài 1	9
Tập lệnh cơ bản của PLC	9
1.1. Tổng quát về điều khiển lập trình	9
1.2. Điều khiển nối cứng và điều khiển lập trình.....	11
1.3.Cấu trúc của một PLC	12
1.4.Thiết bị điều khiển lập trình.....	13
1.5. Xử lý chương trình	15
1.6. Cài đặt và sử dụng phần mềm.....	16
Bài 2	22
Các tập lệnh của dữ liệu	22
2.1.Các liên kết logic.....	22
2.2.Các lệnh ghi / xóa giá trị cho tiếp điểm	30
2.3.Timer	32
2.4.Counter	36
Bài 3	43
Các phép toán số của PLC.....	43
3.1. Chức năng truyền dẫn	43
3.2. Chức năng so sánh	46
3.3.Đồng hồ thời gian thực.....	48
Bài 4	51
Lắp đặt mô hình điều khiển bằng PLC.....	51
4.1. Giới thiệu.....	51
4.2.Cách kết nối dây.....	56
4.3. Các mô hình và bài tập ứng dụng	58
Tài liệu cần tham khảo:	68

GIÁO TRÌNH LẬP TRÌNH PLC CƠ BẢN

Tên mô đun: Lập trình PLC cơ bản

Mã số mô đun: MĐ 22

Thời gian của mô đun: 60 giờ (LT: 18 giờ ; TH: 40 giờ; KT: 2 giờ)

I. Vị trí, tính chất của mô đun:

Vị trí :

Mô đun được bố trí dạy cuối chương trình sau khi học xong các môn chuyên môn như điện tử công suất, Kỹ thuật xung – số, kỹ thuật cảm biến, trang bị điện, lắp đặt và điều khiển thiết bị điện công nghiệp....

Tính chất :

Là mô đun chuyên môn nghề.

II. Mục tiêu của mô đun:

Kiến thức:

Trình bày được các khái niệm về điều khiển lập trình chính xác theo nội dung đã học

Trình bày được cấu trúc và phương thức hoạt động của các lệnh cơ bản

Kỹ năng:

Thực hiện lập trình các bài tập ứng dụng dùng PLC đạt các yêu cầu về kỹ thuật và công nghệ

Kết nối mạch điện theo yêu cầu công nghệ

Năng lực tự chủ, trách nhiệm:

Vận dụng được những kiến thức của môn học để tiếp thu các môn học, mô-đun chuyên nghề.

Rèn luyện tính tỉ mỉ, chính xác và an toàn vệ sinh công nghiệp

III. Nội dung mô đun:

1. Nội dung tổng quát và phân bổ thời gian

TT	Tên chương, mục	Thời gian			
		Tổng số	Lý thuyết	Thực hành/ thực tập/thí nghiệm/ bài tập/thảo luận	Kiểm tra
1	Bài mở đầu 1.Khái quát chung về PLC 2.Các bước thiết lập hệ điều khiển bằng rơ le và lập trình nhớ 3.Sự khác nhau giữa hệ điều khiển bằng rơ le điện và hệ điều khiển theo lập trình nhớ. 4.Hệ điều khiển lập trình nhớ PLC có những ưu điểm	1	1	0	0
2	Bài 1:Tập lệnh cơ bản của PLC 1.1.Tổng quan về điều khiển lập trình 1.2. Điều khiển nối cứng và điều khiển lập trình 1.3. Cấu trúc của một PLC 1.4. Thiết bị điều khiển lập trình 1.5. Xử lý chương trình 1.6. Cài đặt và sử dụng phần mềm	12	3	9	
3	Bài 2:Các tập lệnh của dữ liệu 2.1. Các liên kết logic	12	4	7	1

	<p>2.2.Các lệnh ghi/xóa giá trị cho tiếp điểm</p> <p>2.2.1.LệnhSET (S) và RESET (R)</p> <p>2.2.2. Các ví dụ</p> <p>2.2.3. Timer</p> <p>2.2.4.Counter</p> <p>Kiểm tra</p>				
4	<p>Bài 3:Các phép toán số của PLC</p> <p>3.1.Chức năng truyền dẫn</p> <p>3.2.Chức năng so sánh</p> <p>3.3.Đồng hồ thời gian thực</p> <p>Kiểm tra</p>	12	4	8	
5	<p>Bài 4: Lắp đặt mô hình điều khiển bằng PLC</p> <p>4.1. Giới thiệu</p> <p>4.2. Cách kết nối dây</p> <p>4.3. Các mô hình và bài tập ứng dụng</p> <p>4.3.1.Điều khiển van điện từ hai cuộn dây</p> <p>4.3.2.Điều khiển hệ thống cung cấp khí nén</p> <p>4.3.3.Điều khiển hệ thống cung cấp thủy lực</p> <p>4.3.4.Điều khiển hệ thống thông gió</p> <p>4.3.5.Điều khiển động cơ thuận nghịch</p>	23	6	16	1
Cộng		60	18	40	2

Bài mở đầu

Mục tiêu:

Nhận biết các loại điều khiển: Rơ le, PLC

Trình bày được ưu nhược điểm của điều khiển PLC

Tích cực, chủ động và sáng tạo trong học tập.

1. Khái quát chung về PLC

Thiết bị điều khiển lập trình đầu tiên (Programmable Controller) đã được những nhà thiết kế cho ra đời năm 1968 (công ty General Motor - Mỹ). Tuy nhiên, hệ thống này còn khá đơn giản và cồng kềnh, người sử dụng gặp nhiều khó khăn trong việc vận hành hệ thống. Vì vậy các nhà thiết kế từng bước cải tiến hệ thống làm cho hệ thống đơn giản, gọn nhẹ, dễ vận hành. Nhưng việc lập trình cho hệ thống còn khó khăn do lúc này không có các thiết bị lập trình chuyên dùng hỗ trợ cho công việc lập trình. Để đơn giản hóa việc lập trình, hệ thống điều khiển lập trình cầm tay (Programmable Controller Handle) đầu tiên được ra đời vào năm 1969. Điều này đã tạo ra sự phát triển thực sự cho kỹ thuật điều khiển lập trình. Các nhà thiết kế đã từng bước chuẩn hóa ngôn ngữ lập trình, đó là ngôn ngữ lập trình dùng giản đồ hình thang (Ladder Diagram). Các nhà sản xuất liên tục đưa ra các công cụ (cả phần mềm và thiết bị) hỗ trợ cho việc lập trình, giám sát và gỡ rối.

Bộ điều khiển lập trình là ý tưởng của một nhóm kỹ sư hãng General Motors vào năm 1968 nhằm đáp ứng những yêu cầu điều khiển trong công nghiệp. Ban đầu nó mới chỉ được sử dụng để thay thế cho hệ thống điều khiển sử dụng rơle. Bộ điều khiển PLC lúc đầu chỉ là một thiết bị đơn giản. Đầu vào của nó được kết nối với công tắc, cảm biến số...và dựa trên những phép tính logic bên trong mà đầu ra của nó sẽ đóng hoặc mở các thiết bị. Khi mới xuất hiện, bộ điều khiển PLC không tương thích với các hệ thống điều khiển khá phức tạp như điều khiển nhiệt độ, vị trí, áp suất...tuy nhiên, vào những năm kế tiếp nhà sản xuất đã liên tục cải tiến nó.

Hiện nay, PLC đã được nhiều hãng khác nhau sản xuất như: Siemens, Omron, Mitsubishi, Festo, Alan Bradley, Schneider, Hitachi ... Theo xu hướng chuẩn hóa và module hóa thì PLC của các hãng khác nhau đều có cấu trúc phần cứng cũng như tập lệnh tương tự nhau.

2. Các bước thiết lập hệ điều khiển bằng rơ le và lập trình nhớ

Khi bắt đầu xây dựng một hệ thống điều khiển trên cơ sở ứng dụng PLC, một câu hỏi đặt ra là phải thực hiện những công việc theo một quy trình như thế nào?. Có thể đó không phải là một vấn đề lớn khi xây dựng một hệ thống đơn giản. Nhưng đối với những hệ thống phức tạp thì cần phải có một quy trình thiết kế phù hợp. Nó giúp cho người thiết kế kiểm soát được quá trình thực hiện công việc của mình, từ sự mô tả chức năng và yêu cầu của hệ thống cho đến việc lập chương trình điều khiển cho PLC. Trong chương này sẽ đưa ra mô hình hệ thống điều khiển trình tự, đề cập đến phương pháp mô tả chức năng hệ thống điều khiển trình tự và kỹ thuật lập trình điều khiển trình tự ứng dụng PLC.

3. Sự khác nhau giữa hệ điều khiển bằng rơ le điện và hệ điều khiển theo lập trình nhớ.

Một hệ thống điều khiển tự động bao gồm các thành phần chính sau:

- Phần điều khiển: phần điều khiển có chức năng tạo ra các lệnh điều khiển cần thiết tùy thuộc vào thông tin mà nó nhận được. Các thông tin này có thể nhận được từ người điều khiển hoặc thông tin phản hồi từ phần chấp hành thông qua các cảm biến.

- Phần chấp hành: đôi khi còn gọi là phần công suất, nhận lệnh từ phần điều khiển để thực hiện điều khiển đối tượng. Phần chấp hành có thể là các động cơ điện, cuộn dây điện từ, rơle...

Thiết kế hệ thống điều khiển trình tự ứng dụng PLC gồm có hai nhiệm vụ là thiết kế phần cứng và thiết kế chương trình điều khiển. Thiết kế chương trình điều khiển chỉ là một phần trong toàn bộ quá trình thiết kế, nhưng là yếu tố quan trọng vì nó tạo ra các tài liệu cần thiết giúp cho việc lập trình và gỡ rối cũng như lập tài liệu hệ thống để lưu trữ sau này. Ta xét phương pháp tổng quát khi thiết kế hệ thống điều khiển trình tự ứng dụng PLC. Phương pháp này cho phép triển khai, lắp đặt phần cứng và thiết kế chương trình điều khiển được tiến hành độc lập và song song. Nó cũng cho phép trao đổi thông tin bổ xung giữa các quá trình thiết kế nhằm hoàn thiện hệ thống theo hướng tối ưu nhất. Đối với hệ thống điều khiển đơn giản thì ít khi đòi hỏi sự hoạch định và thiết kế chương trình, bởi vì không có nhiều sự liên kết logic giữa các phần trong chương trình. Đối với các hệ thống phức tạp, cần thiết kế chương trình có cấu trúc và theo một quy trình xác định, điều đó làm cho quá trình được kiểm soát, tránh nhầm lẫn và thiếu sót khi thiết kế chương trình, chương trình dễ đọc, hiệu chỉnh, bổ xung, và lập tài liệu thiết kế. Trong quy trình thiết kế hệ thống điều khiển, một vấn đề quan trọng là phải mô tả hệ thống điều khiển một

cách chính xác, khoa học, và được chuẩn hóa. Ngoài ra, cách mô tả hệ thống phải tạo điều kiện thuận lợi cho việc thiết kế chương trình điều khiển.

4. Hệ điều khiển lập trình nhớ PLC có những ưu điểm.

Có thể nêu ra một số ưu điểm chính khi sử dụng PLC như sau:

- Tính linh hoạt: có thể sử dụng một bộ điều khiển cho nhiều đối tượng khác nhau với các thuật toán điều khiển khác nhau.

- Dễ dàng thiết kế và thay đổi logic điều khiển: với các hệ thống điều khiển sử dụng rơle, khi thay đổi logic điều khiển cần có nhiều thời gian để nối lại dây cho các thiết bị và panel điều khiển, và đó là một công việc phức tạp. Với hệ thống điều khiển sử dụng PLC, thay đổi logic điều khiển bằng cách thay đổi chương trình thông qua thiết bị lập trình và ngôn ngữ lập trình chuyên dùng. Điều đó làm giảm đáng kể thời gian thiết kế hệ thống.

- Tối ưu logic điều khiển: được sự hỗ trợ của các công cụ mô phỏng và gỡ rối trực tuyến và trực quan làm cho hệ thống được thiết kế có tính tối ưu hơn.

- Tốc độ thực hiện nhanh.

- Nhỏ, gọn và giá thành thấp.

- Khả năng bảo mật hệ thống khi sử dụng mã khóa.

- Khả năng mở rộng và nâng cấp hệ thống: do được chế tạo dưới dạng các modul được chuẩn hóa cho phép ghép nối các thành phần không chỉ của một nhà sản xuất. Đây là một yêu cầu không thể thiếu trong các hệ thống điều khiển hiện đại.

Bài 1

Tập lệnh cơ bản của PLC

Mục tiêu:

Phát biểu được khái niệm về điều khiển lập trình theo nội dung đã học

So sánh ưu nhược điểm của điều khiển lập trình với các hình thức điều khiển khác theo nội dung đã học.

Trình bày được các ứng dụng của PLC trong thực tế theo nội dung đã học.

Rèn luyện tính tư duy, tác phong công nghiệp

1.1. Tổng quát về điều khiển lập trình

Trong ứng dụng các công nghệ khoa học vào sản xuất công nghiệp yêu cầu tự động hoá ngày càng tăng, đòi hỏi kỹ thuật điều khiển phải đáp ứng được những yêu cầu đó, với mục tiêu tăng năng suất lao động bằng con đường tăng mức độ tự động hóa các quá trình và thiết bị sản xuất nhằm mục đích tăng sản lượng, cải thiện chất lượng và độ chính xác của sản phẩm.

Tự động hóa trong sản xuất nhằm thay thế một phần hoặc toàn bộ các thao tác vật lý của công nhân vận hành máy thông qua hệ thống điều khiển. Những hệ thống điều khiển này có thể điều khiển quá trình sản xuất với độ tin cậy cao, ổn định mà không cần sự tác động nhiều của người vận hành. Điều này đòi hỏi hệ thống điều khiển phải có khả năng khởi động, kiểm soát, xử lý và dừng một quá trình theo yêu cầu hoặc đo đếm các giá trị đã được xác định nhằm đạt được kết quả mong muốn ở sản phẩm đầu ra của máy hay thiết bị. Một hệ thống như vậy được gọi là hệ thống điều khiển.

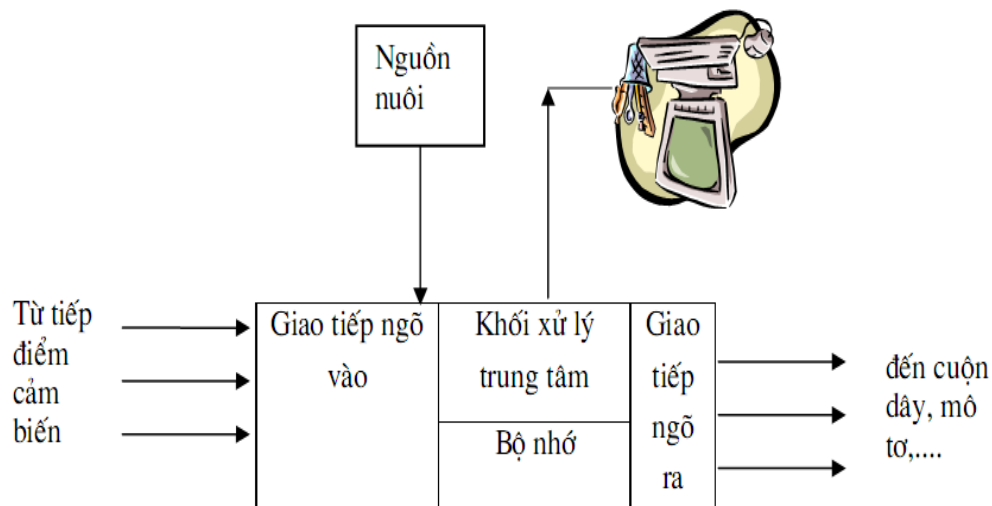
Trong kỹ thuật tự động điều khiển, các bộ điều khiển chia làm 2 loại:

- + Điều khiển nối cứng
- + Điều khiển logic khả trình (PLC)

Một hệ thống điều khiển bất kỳ được tạo thành từ các thành phần:

- + Khối vào
- + Khối xử lý – điều khiển
- + Khối ra

*** Sơ đồ tổng quát của điều khiển lập trình như sau (hình 1.1):**



Hình 1.1

a. Khối vào: (bảng 1.1)

Còn được gọi là giao tiếp ngõ vào có nhiệm vụ biến đổi các đại lượng vật lý đầu vào (từ các tiếp điểm của cảm biến, hay các nút nhấn, điện trở đo sức căng....) thành các mức tín hiệu số ON/OFF (digital) hay tín hiệu liên tục (analog) tùy theo bộ chuyển đổi ngõ vào và cấp vào cho khối xử lý trung tâm (CPU).

Bộ chuyển đổi	Đại lượng đo	Đại lượng ra
Công tắc (Switch)	Sự dịch chuyển/vị trí	Điện áp nhị phân (ON/OFF)
Công tắc hành trình (Limit switch)	Sự dịch chuyển/vị trí	Điện áp nhị phân (ON/OFF)
Bộ điều chỉnh nhiệt (Thermostat)	Nhiệt độ	Điện áp nhị phân (ON/OFF)
Cặp nhiệt điện (Thermocouple)	Nhiệt độ	Điện áp thay đổi
Nhiệt trở (Thermister)	Nhiệt độ	Trở kháng thay đổi
Tế bào quang điện (Photo cell)	Ánh sáng	Điện áp thay đổi (analog)
Tế bào tiệm cận (Proximity cell)	Sự hiện diện của đối tượng	Trở kháng thay đổi
Điện trở đo sức căng (Strain gage)	Áp suất/ sự dịch chuyển	Trở kháng thay đổi

Bảng 1.1

b. Bộ nhớ (Memory):

Lưu chương trình điều khiển được lập trình bởi người dùng và các dữ liệu khác như cờ, thanh ghi tạm, trạng thái đầu vào, lệnh điều khiển đầu ra... Nội dung các bộ nhớ đã được mã hóa dưới dạng mã nhị phân.

c. Khối xử lý – điều khiển:

Là khối xử lý trung tâm (CPU) thay thế người vận hành thực hiện các thao tác đảm bảo quá trình hoạt động. Từ thông tin tín hiệu vào hệ thống điều khiển tuần tự thực thi các lệnh trong chương trình lưu trong bộ nhớ, xử lý các đầu vào và đưa kết quả xuất hoặc điều khiển cho phần giao diện đầu ra (output) như: cuộn dây, mô tơ... Tín hiệu điều khiển được thực hiện theo 2 cách:

- + Dùng mạch điện nối kết cứng
- + Dùng chương trình điều khiển

d. Khối ra: (bảng 1.2)

Còn được gọi là phần giao diện đầu ra. Tín hiệu ra là kết quả của quá trình xử lý của hệ thống điều khiển. Lúc này tín hiệu ngõ vào được biến đổi thành mức tín hiệu vật lý thích hợp bên ngoài như: đóng mở rơle, biến đổi tuyến tính số- tương tự.....

Bảng 1.2

Thiết bị ở ngõ ra	Đại lượng ra	Đại lượng tác động
Động cơ điện	Chuyển động quay	Điện
Xy lanh- Piston	Chuyển động thẳng/áp lực	Dầu ép/ khí ép
Solenoid	Chuyển động thẳng/áp lực	Điện
Lò sấy/ lò cấp nhiệt	Nhiệt	Điện
Van	Tiết diện cửa van thay đổi	Điện/dầu ép/khí ép
Rơle	Tiếp điểm điện/ chuyển động vật lý có giới hạn	Điện

1.2. Điều khiển nối cứng và điều khiển lập trình.

1.2.1. Điều khiển nối cứng.

Thiết bị điều khiển lập trình đầu tiên (Programmable Controller) đã được những nhà thiết kế cho ra đời năm 1968 (công ty General Motor - Mỹ). Tuy nhiên, hệ thống này còn khá đơn giản và cồng kềnh, người sử dụng gặp nhiều khó khăn trong việc vận hành hệ thống. Vì vậy các nhà thiết kế từng bước cải

tiến hệ thống làm cho hệ thống đơn giản, gọn nhẹ, dễ vận hành. Nhưng việc lập trình cho hệ thống còn khó khăn do lúc này không có các thiết bị lập trình chuyên dùng hỗ trợ cho công việc lập trình. Để đơn giản hóa việc lập trình, hệ thống điều khiển lập trình cầm tay (Programmable Controller Handle) đầu tiên được ra đời vào năm 1969. Điều này đã tạo ra sự phát triển thực sự cho kỹ thuật điều khiển lập trình. Các nhà thiết kế đã từng bước chuẩn hóa ngôn ngữ lập trình, đó là ngôn ngữ lập trình dùng giản đồ hình thang (Ladder Diagram). Các nhà sản xuất liên tục đưa ra các công cụ (cả phần mềm và thiết bị) hỗ trợ cho việc lập trình, giám sát và gỡ rối.

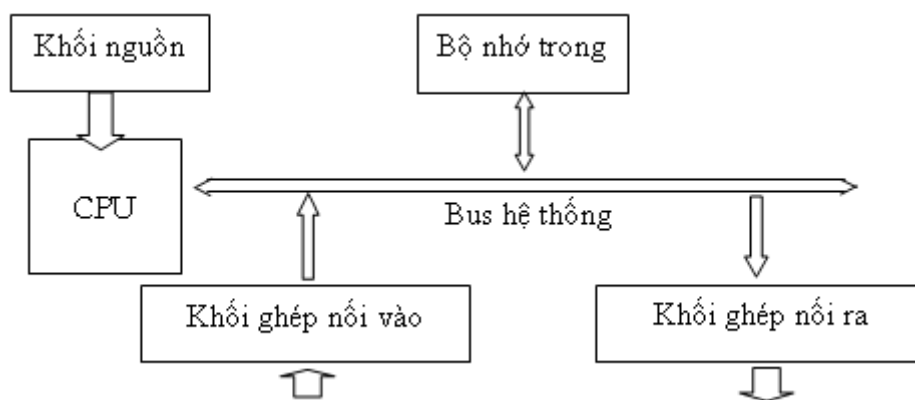
1.2.2. Điều khiển lập trình.

Bộ điều khiển lập trình là ý tưởng của một nhóm kỹ sư hãng General Motors vào năm 1968 nhằm đáp ứng những yêu cầu điều khiển trong công nghiệp. Ban đầu nó mới chỉ được sử dụng để thay thế cho hệ thống điều khiển sử dụng rơle. Bộ điều khiển PLC lúc đầu chỉ là một thiết bị đơn giản. Đầu vào của nó được kết nối với công tắc, cảm biến số...và dựa trên những phép tính logic bên trong mà đầu ra của nó sẽ đóng hoặc mở các thiết bị. Khi mới xuất hiện, bộ điều khiển PLC không tương thích với các hệ thống điều khiển khá phức tạp như điều khiển nhiệt độ, vị trí, áp suất...tuy nhiên, vào những năm kế tiếp nhà sản xuất đã liên tục cải tiến nó.

Hiện nay, PLC đã được nhiều hãng khác nhau sản xuất như: Siemens, Omron, Mitsubishi, Festo, Alan Bradley, Schneider, Hitachi ... Theo xu hướng chuẩn hóa và module hóa thì PLC của các hãng khác nhau đều có cấu trúc phần cứng cũng như tập lệnh tương tự nhau.

1.3.Cấu trúc của một PLC

1.3.1.Sơ đồ khối của PLC



Hình 1.1: Cấu trúc một bộ PLC

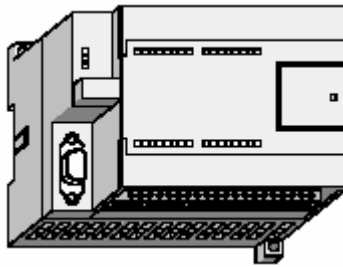
1.3.2.Chức năng các khối

PLC là thiết bị điều khiển dựa trên bộ vi xử lý, các thành phần cơ bản của nó gồm:

- Khối xử lý trung tâm (Central Processing Unit - CPU)
- Bộ nhớ trong (Internal Memory)
- Bus hệ thống (System Bus)
- Khối ghép nối vào/ra (Input/Output Interface)
- Khối nguồn (Power Supply)

1.4.Thiết bị điều khiển lập trình

S7-200 là họ PLC loại nhỏ của hãng SIEMENS (Micro PLC) được cấu trúc theo dạng module. Một PLC S7-200 gồm có một Module CPU S7-200 và có thể có các Module mở rộng. Module CPU S7-200 gồm có một khối xử lý trung tâm (CPU), bộ nhớ trong, nguồn công suất, các đầu vào/ra số tích hợp, cổng truyền thông. Có thể tăng thêm các đầu vào/ra số cũng như các chức năng chuyên dùng khác bằng các module mở rộng.



Hình dáng bên ngoài PLC S7-200

1.4.1.Địa chỉ vào ra

Định địa chỉ trực tiếp các vùng nhớ

CPU S7-200 lưu trữ dữ liệu ở các vị trí khác nhau trong bộ nhớ, mỗi vị trí nhớ có một địa chỉ duy nhất. Chương trình có thể sử dụng địa chỉ của vị trí nhớ để truy nhập dữ liệu trong bộ nhớ. Cách truy nhập này gọi là định địa chỉ trực tiếp.

1.4.2.Phần chữ chỉ vị trí và kích thước của ô nhớ.

Để truy nhập đến một bit trong một vùng nhớ nào đó, chương trình phải chỉ rõ địa chỉ gồm tên vùng nhớ, địa chỉ byte, và địa chỉ bit. Địa chỉ byte và địa

chỉ bit ngăn cách nhau bởi dấu (.). Chế độ địa chỉ này gọi là chế độ định địa chỉ bit.

1.4.3. Phân số chỉ địa chỉ của byte hoặc bit trong miền nhớ đã xác định.

Định dạng địa chỉ:

[Tên vùng nhớ].[Địa chỉ byte].[Địa chỉ bit]

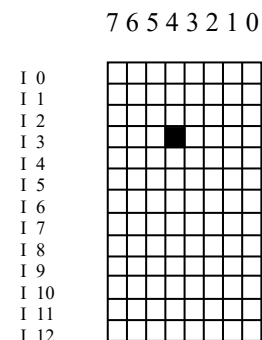
Ví dụ: I3.4

Trong đó:

I: tên vùng nhớ (vùng nhớ đệm vào)

3: địa chỉ byte

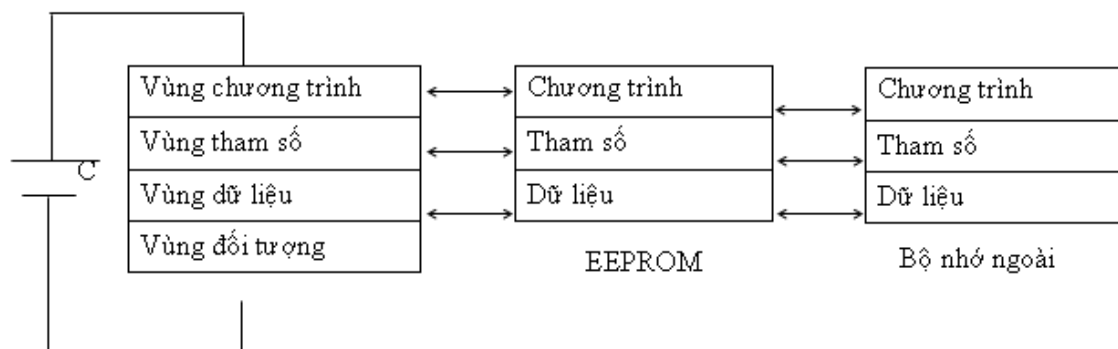
4: địa chỉ bit



Các bộ điều chỉnh tương tự được đặt ở module CPU. Giá trị của các bộ điều chỉnh tương tự được biến đổi thành giá trị số 8 bit và được lưu vào các byte nhớ đặc biệt: SMB28 tương ứng với bộ điều chỉnh tương tự 0; SMB29 tương ứng với bộ điều chỉnh tương tự 1. Có thể sử dụng các bộ điều chỉnh tương tự để nhập giá trị khi cần thiết.

1.4.4. Cấu trúc bộ nhớ của.

Bộ điều khiển lập trình S7-200 được chia thành 4 vùng nhớ. Với 1 tụ có nhiệm vụ duy trì dữ liệu trong thời gian nhất định khi mất nguồn bộ nhớ S7-200 có tính năng động cao đọc và ghi trong phạm vi toàn vùng loại trừ các bit nhớ đặc biệt SM (Special Memory) chỉ có thể truy nhập để đọc.



Hình 1.2. Cấu trúc bộ nhớ trong và ngoài của S7-200

- Vùng chương trình: là vùng bộ nhớ được sử dụng để lưu trữ các lệnh chương trình vùng này thuộc bộ nhớ trong đọc và ghi được

- Vùng tham số: là vùng lưu giữ các tham số như: từ khóa, địa chỉ trạm... cũng giống như vùng chương trình thuộc bộ nhớ trong đọc và ghi được

- Vùng dữ liệu: là vùng nhớ động được sử dụng cất các dữ liệu của chương trình bao gồm các kết quả các phép tính nó được truy cập theo từng bit từng byte vùng này được chia thành những vùng nhớ với các công dụng khác nhau.

1.5. Xử lý chương trình

1.5.1. Vòng quét chương trình.

Chức năng của CPU là thực hiện lặp đi lặp lại một chuỗi công việc. Sự thực hiện các công việc có tính tuần hoàn này gọi là vòng quét (Scan cycle).

1.5.2. Cấu trúc chương trình.

Trong mỗi vòng quét, CPU thực hiện hầu hết hoặc toàn bộ các công việc sau:

Đọc các đầu vào

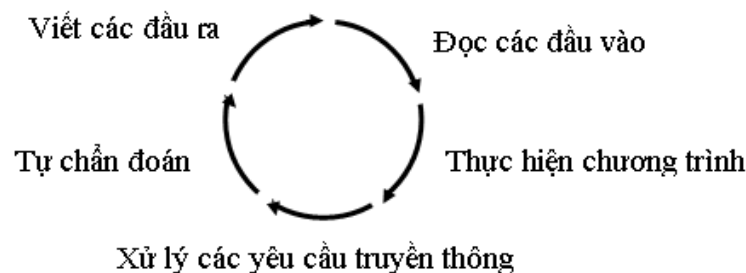
Thực hiện chương trình

Xử lý các yêu cầu truyền thông

Thực hiện tự chẩn đoán

Viết các đầu ra

Có thể biểu diễn vòng quét của CPU như sau:



1.5.3. Phương pháp lập trình.

Như đã nói ở trên thì ngôn ngữ lập trình LAD có ưu điểm là dễ sử dụng, gần với tư duy lập trình logic (tư tưởng chính của PLC), dễ dàng thao tác mà lại không cần phải nhớ tập lệnh. Cho nên với những người mới sử dụng thì việc lựa chọn ngôn ngữ LAD là đúng đắn. Ngay cả với chúng tôi những người đã đi làm thực tế, có nhiều kinh nghiệm thì việc lập trình cho PLC vẫn chủ yếu bằng LAD.

Tuy nhiên các ngôn ngữ STL, FBD vẫn được sử dụng tùy thuộc vào mục đích, mỗi ngôn ngữ lập trình đều có những thế mạnh riêng. Có lúc ta sử dụng ngôn ngữ này thì hợp, có lúc ta sử dụng ngôn ngữ khác lại hợp. Để chọn và sử dụng ngôn ngữ ta bấm vào View, cửa sổ sẽ hiện ra:

1.6. Cài đặt và sử dụng phần mềm.

Để chọn ngôn ngữ lập trình (LAD, STL, FBD) ta click chuột vào ngôn ngữ muốn dùng trong thanh số dọc của View. Bạn mới lập trình thì có thể chọn ngôn ngữ lập trình LAD. Thật ra thì khi bạn viết một chương trình bằng một ngôn ngữ thì bạn có thể chuyển nó thành 2 dạng ngôn ngữ lập trình còn lại, đó là cách thể hiện chương trình của bạn dưới dạng nào. Có thể nói việc lập trình bằng ngôn ngữ nào còn tùy thuộc vào mục đích, đặc điểm của bài toán đặt ra và còn phụ thuộc vào thói quen của người lập trình.

Chương trình của bạn luôn ở trong khối OB1. Trong Instruction tree bạn hãy nhấp chuột vào Program Block, mục Program mở rộng ra, bạn nhấp chuột vào MAIN(OB1), vùng soạn thảo chương trình sẽ hiện ra ở phía bên phải.

Các phần tử trong LAD được định nghĩa khác nhau từ loại này sang loại khác: đầu vào(Input), đầu ra(Output), Timer, Counter,... nếu cùng một loại thì chúng có số thứ tự khác nhau.

1.6.1. Những yêu cầu đối với máy tính PC.

Vào chế độ trợ giúp Help có thể xem được các lệnh cần thiết cho chế độ soạn thảo, có thể cho hiện lên màn hình tất cả các hộp lệnh trong LAD.

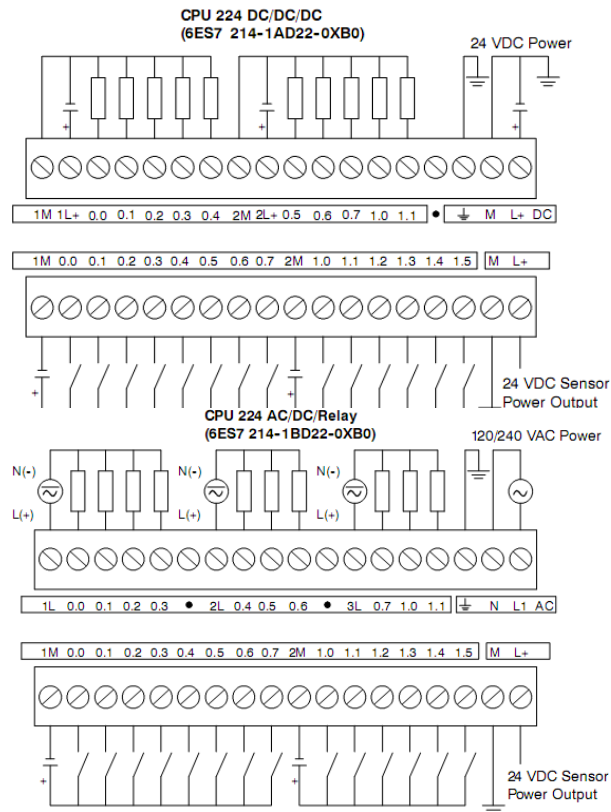
Với cách lập trình bằng LAD thì bạn thể hiện chương trình bằng cách kéo các hộp lệnh và thả vào trong chương trình, và sử dụng các đường nối để nối các hộp lệnh với nhau.

Các hộp lệnh bạn có thể kéo trong vùng Instruction(vùng 2), còn các đường nối dây bạn có thể kéo từ (vùng 1) vào.

1.6.2. Cài đặt phần mềm lập trình ứng dụng.

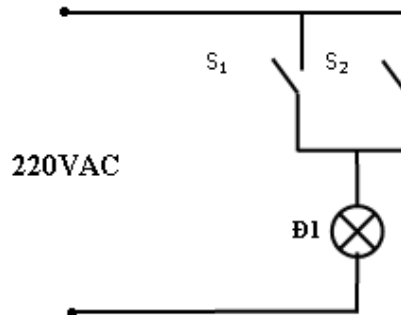
Khi cấp nguồn vào chân của của PLC (chân M và L+ phía đầu ra)

Sơ đồ đấu nối của CPU 224 với loại DC/DC và AC/DC thì bản thân CPU sẽ cung cấp cho ta nguồn 24VDC chuẩn ở phía đầu vào (chân m, L+), nguồn này dùng để nuôi cảm biến nếu ta không có nguồn đầu vào.



Sơ đồ chân của CPU 224

- Lập trình bằng phần mềm sơ đồ nguyên lý mạch sau:



Sơ đồ đấu nối điều khiển đèn chiếu sáng từ hai công tắc.

* Nguyên lý làm việc:

- Đóng công tắc S1 hoặc S2 đều ra đèn sáng.
- Trạng thái tác động này tương đương với trạng thái của mạch OR

* Bước 2: Khai báo địa chỉ và sơ đồ đấu nối vào ra.

Khai báo địa chỉ:

Symbol	Address	Comment
S1	I0.0	Cong tac 1 dieu kien den
S2	I0.1	Cong tac 2 dieu kien den
Den	Q0.0	Den chieu sang D1

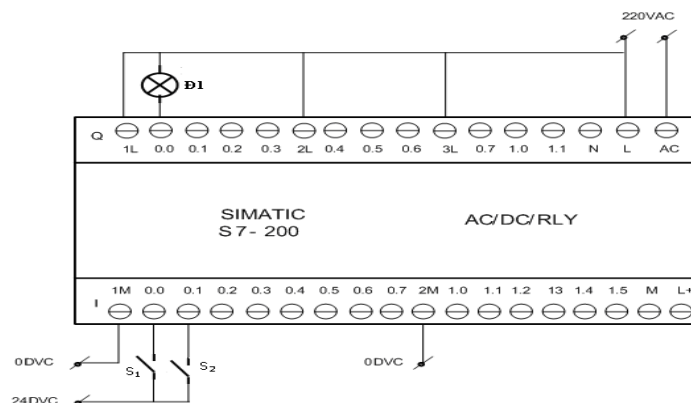
Sơ đồ đấu nối:

- Địa chỉ đầu vào :

I0.0 và I0.1 nối ở đầu vào của blốc: I0.0 đến S1

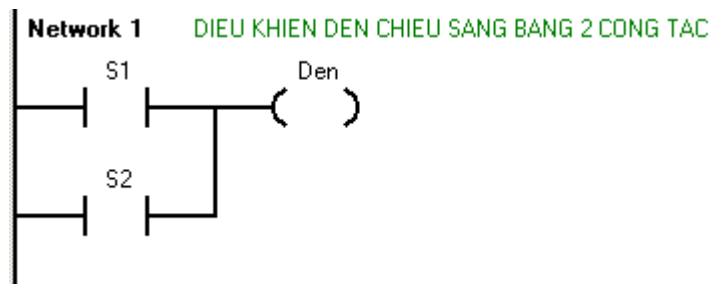
I0.1 đến S2.

- Địa chỉ đầu ra: Đầu ra Q0.0 được nối tới đèn Đ.



Sơ đồ kết nối PLC mạch điều khiển đèn.

* Bước 3: Viết chương trình điều khiển



* Bước 4: Download, chạy thử.

- Nếu bật công tắc S1 (hoặc S2) thì đầu vào I0.0 (hoặc I0.1) được kích hoạt, đầu ra Q0.0 được kích hoạt, đèn được cấp nguồn sáng.

*** Kiểm tra việc kết nối bằng phần mềm.**

Đây là khối cho phép ta theo dõi giá trị của tất cả các biến trong vùng nhớ PLC mà ta sử dụng trong chương trình. Đồng thời ta có thể cho các biến giá trị mới (không kết những biến dạng “Read Only”) để theo dõi hoạt động của chương trình.


- Mở chương trình, vào cửa sổ Status Chart:
- Sẽ xuất hiện màn hình hiển thị như sau

	Address	Format	Current Value	New Value
1		Signed		
2		Signed		
3		Signed		
4		Signed		
5		Signed		


Màn hình hiển thị của Status Chart

- Khi mở cửa sổ Status Chart, có các khối như sau:


	Address	Format	Current Value	New Value
1	Khoidong	Bit		
2	Dung	Bit		
3	Dong_co	Bit		
4		Signed		
5		Signed		
6		Signed		
7		Signed		




Tên biến



Kiểu biến



Giá trị hiện tại



Giá trị mới

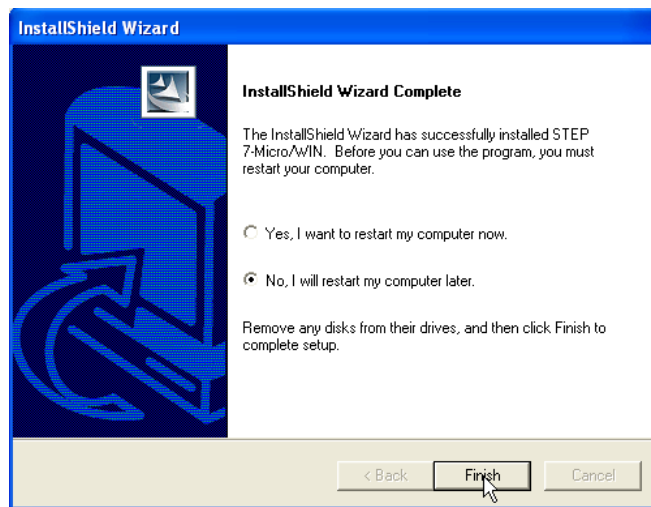
Các khối chức năng trong Status Chart.

- * Cài đặt và sử dụng phần mềm STEP 7 - Micro/win.
- * Trình tự thực hiện cài đặt STEP7-Micro/Win:
 - Cho đĩa CD STEP7-Micro/Win vào trong ổ CDROOM của máy tính. Trình hỗ trợ cài đặt sẽ tự động bắt đầu và nhanh chóng, chỉ cần làm theo các bước để hoàn tất quá trình cài đặt.



Cửa sổ chọn điều kiện License khi cài đặt

Trên cửa sổ này, ta chọn chuẩn PC/PPI cable(PPI) rồi bấm OK. Trên màn hình sẽ hiện ra cửa sổ:




Cửa sổ chọn chế độ khi kết thúc quá trình cài đặt

Trên cửa sổ này, nếu cần khởi động lại PC ngay thì chọn Yes, nếu không thì chọn No, rồi bấm OK. Đến đây quá trình cài đặt kết thúc.

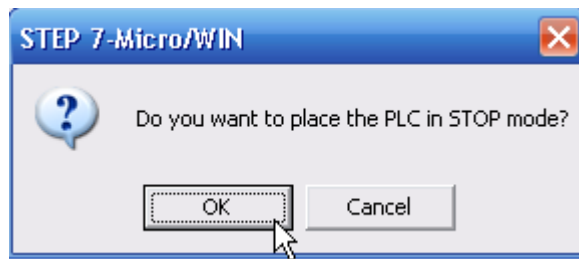
Sau khi cài đặt xong phần mềm, bạn đã có thể tiến hành lập trình cho PLC!. Ở đây chúng tôi giới thiệu với các bạn phần mềm STEP7-Micro/Win phiên bản 4.0 theo chuẩn công nghiệp IEC 1131-3.




***Kiểm tra**

Khi không có thông báo lỗi thì bạn có thể nạp chương trình xuống PLC bằng cách nhấn vào biểu tượng ,

Trên cửa sổ này ta ấn Download rồi ấn OK. Nếu PLC của bạn đang ở chế độ “STOP mode” thì việc download sẽ thực hiện ngay, còn nếu PLC của bạn đang ở chế độ “RUN mode” thì một hộp thoại sẽ hiện ra yêu cầu bạn phải chuyển sang chế độ “STOP mode”, hãy ấn OK:



Cửa sổ xác nhận Download chương trình

Khi download xong bạn muốn cho PLC hoạt động thì ấn vào biểu tượng . Khi hiện ra hộp thoại thông báo, nhấn OK.

Bài 2

Các tập lệnh của dữ liệu

Mục tiêu:

Trình bày được các chức năng của Set (S) và Reset (R), counter, Timer (bộ đếm, Bộ định thời).

Lập trình, kết nối, chạy thử các chức năng của S, R, Timer, counter trong các bài toán thực tế.

Tích cực, chủ động và sáng tạo trong học tập.

2.1.Các liên kết logic

2.1.1. Các lệnh vào/ra và các lệnh tiếp điểm đặc biệt.

a. Giới thiệu các lệnh vào/ ra.

Vùng chứa lệnh: để lấy các lệnh vào/ra ta có thể vào Bit logic để lấy lệnh.

* Lệnh tiếp điểm thường mở: ta vào Bit Logic để lấy lệnh

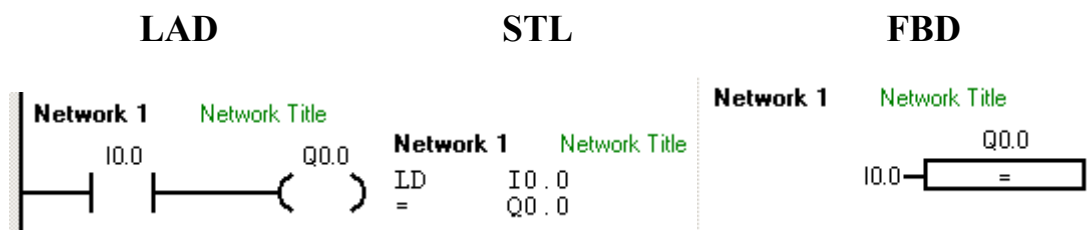
- Trong đó:

Toán hạng của lệnh: I, Q, M, SM, T, C, V, S, L

Kiểu dữ liệu: BOOL

bit: địa chỉ của bit lệnh ứng với các toán hạng trên

- Các kiểu ngôn ngữ của PLC với lệnh tiếp điểm thường mở:



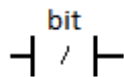
- Tác dụng của lệnh:

+ Trong LAD: Lệnh tác động khi bit địa chỉ có giá trị là 1, khi đó tiếp điểm đóng lại. Khi bit địa chỉ có giá trị là 0 thì lệnh không tác động, khi đó tiếp điểm mở ra.

+ Trong STL: Nạp giá trị Logic của tiếp điểm n vào bit đầu tiên của ngăn xếp

+ Trong FBD: Lệnh tác động khi bit địa chỉ có giá trị là 1.

* **Lệnh tiếp điểm thường đóng:** tương tự như lệnh tiếp điểm thường mở, lệnh tiếp điểm thường đóng cũng vào Bit logic để lấy lệnh.



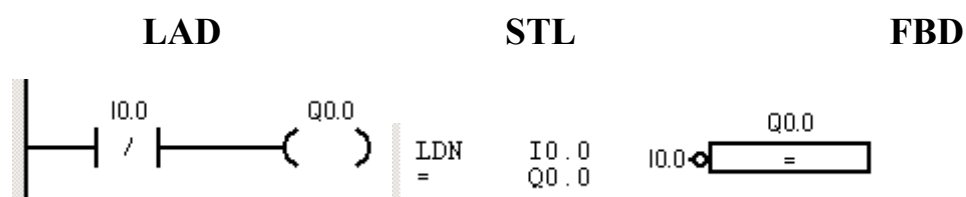
- Trong đó:

Toán hạng của lệnh là: I, Q, M, SM, T, C, V, S, L

Kiểu dữ liệu: BOOL

bit: địa chỉ của bit lệnh ứng với các toán hạng trên

- Các kiểu ngôn ngữ của PLC với lệnh tiếp điểm thường đóng:



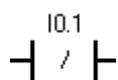
- Tác dụng của lệnh:

+ Trong LAD: Lệnh tác động khi bit địa chỉ có giá trị là 1, khi đó tiếp điểm mở ra. Khi bit địa chỉ có giá trị là 0 thì lệnh không tác động, khi đó tiếp điểm đóng lại.

+ Trong STL: Nạp giá trị nghịch đảo của tiếp điểm n vào bit đầu tiên của ngăn xếp

+ Trong FBD: Lệnh tác động khi bit địa chỉ có giá trị là 1, nó đảo trạng thái cho đầu ra.

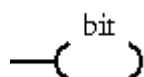
Ví dụ:



Lệnh tiếp điểm thường mở có địa chỉ là I0.1, lệnh sẽ tác động khi bit I0.1 có giá trị là 1.



- Giới thiệu lệnh cuộn dây Output:



Trong đó:

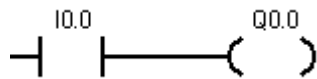
Lệnh cuộn dây là lệnh gán giá trị mới lên bit địa chỉ của lệnh

Toán hạng của lệnh là: I, Q, M, SM, T, C, V, S, L

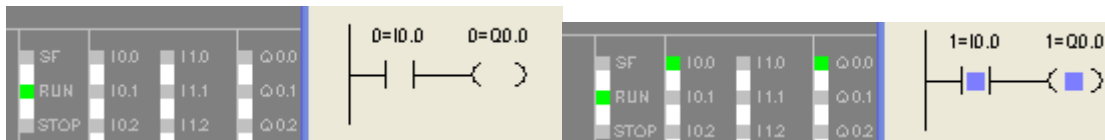
Kiểu dữ liệu: BOOL

bit: địa chỉ của bit lệnh ứng với các toán hạng trên

Ví dụ:

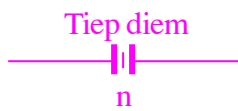


Lệnh cuộn dây có địa chỉ là Q0.0, khi lệnh tiếp điểm thường mở có địa chỉ I0.0 tác động thì lệnh cuộn dây sẽ gán giá trị là 1 vào địa chỉ Q0.0. Khi lệnh tiếp điểm thường mở I0.0 không tác động thì lệnh cuộn dây sẽ gán giá trị là 0 vào địa chỉ Q0.0.

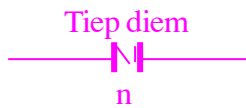


* Các lệnh tiếp điểm tức thời:

- Viết trong LAD:



Tiếp điểm thường mở n được đóng tức thời nếu n = 1



Tiếp điểm thường đóng n được mở tức thời nếu n = 1.



Cuộn dây đầu ra ở trạng thái kích thích (hút)tức thời khi có dòng điện chạy qua

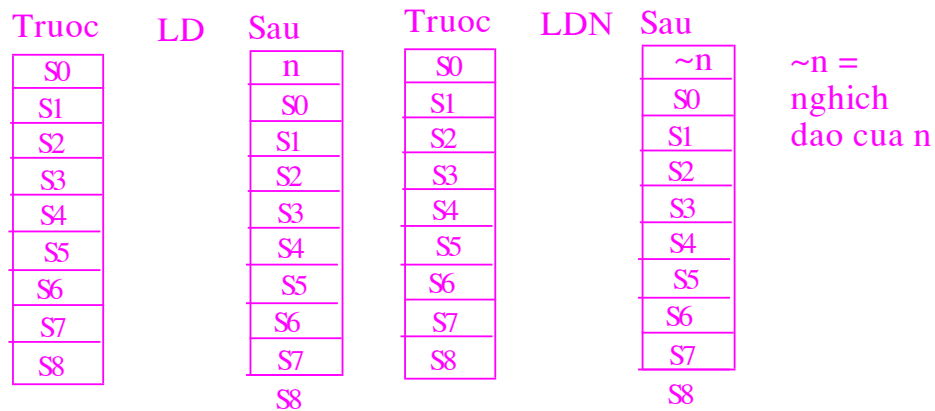
- Viết trong STL:

LDI n Nạp tức thời giá trị Logic của tiếp điểm n vào bit đầu tiên của ngăn xếp

LDNI n Nạp tức thời giá trị Logic nghịch đảo của tiếp điểm n vào bit đầu tiên của ngăn xếp.

=I n Lệnh sao chép tức thời giá trị đỉnh ngăn xếp tới tiếp điểm (đầu ra) n được chỉ trong lệnh

Tác động vào ngăn xếp



b. Các lệnh tiếp điểm đặc biệt.

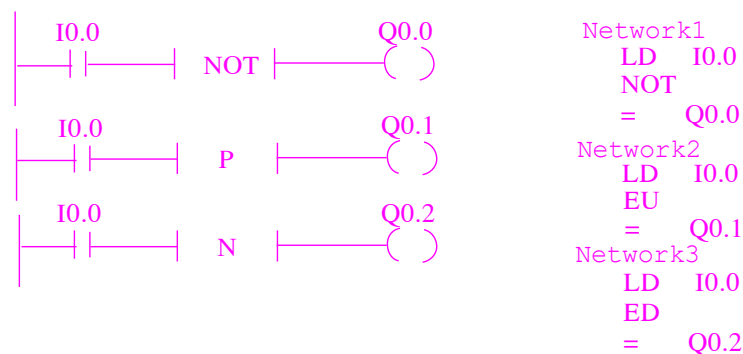
Vùng chứa lệnh: muốn lấy các lệnh tiếp điểm đặc biệt ta cũng vào Bit logic để lấy.

* Lệnh ĐẢO-NOT:

- Tác dụng của lệnh: Lệnh có tác dụng đảo trạng thái tín hiệu đứng trước nó. Khi tín hiệu trước nó ở trạng thái ON thì lệnh sẽ đảo thành OFF.

- Tác dụng của lệnh: Lệnh tiếp điểm phát hiện sườn dương P sẽ phát ra một xung khi tín hiệu trước nó chuyển trạng thái từ mức thấp OFF lên mức cao ON (với lệnh phát hiện sườn dương P) hoặc từ ON xuống OFF (với lệnh phát hiện sườn âm N), độ rộng của xung này bằng thời gian của một chu kỳ quét.

- Ví dụ:



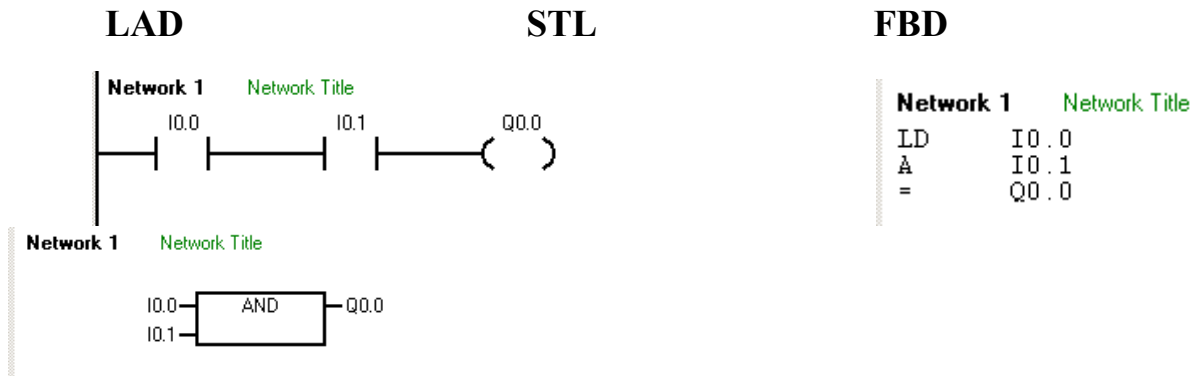
2.1.2. Các lệnh liên kết logic cơ bản.

Các lệnh liên kết logic chỉ được viết trong ngôn ngữ STL và FBD, còn với ngôn ngữ LAD thì tương ứng với mỗi phương pháp tổ hợp cách nối thiết bị

để tạo ra các hàm này. Muốn lấy các hàm logic cơ bản ta cũng vào Bit logic để lấy lệnh.

* Lệnh VÀ-AND.

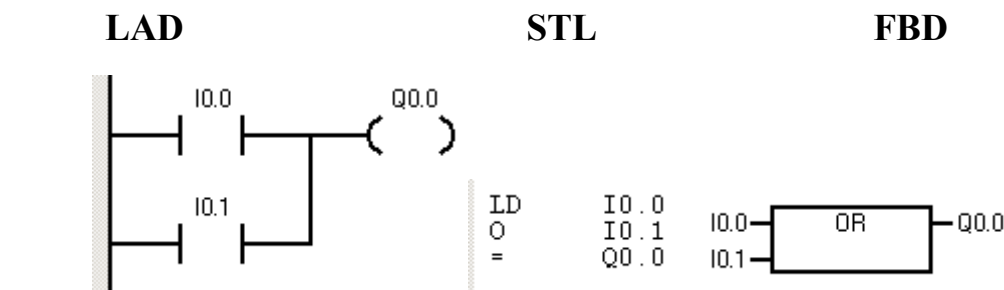
- Cách viết lệnh trong PLC như sau (với ngôn ngữ LAD ta thực hiện nối nối tiếp hai tiếp điểm)



- Lệnh có tác dụng thực hiện phép nhân giữa giá trị logic của tiếp điểm n và giá trị bit đầu tiên trong ngăn xếp.

* Lệnh HOẶC-OR.

- Cách viết lệnh trong PLC như sau (với ngôn ngữ LAD ta thực hiện nối song song hai tiếp điểm).



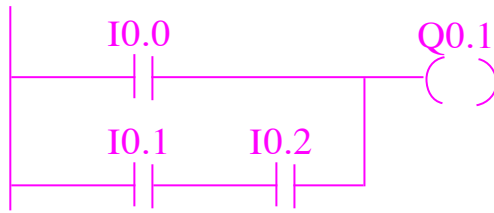
- Lệnh có tác dụng thực hiện phép cộng giữa giá trị logic của tiếp điểm n và giá trị bit đầu tiên trong ngăn xếp.

2.1.3. Liên kết các cổng logic cơ bản.

Là lệnh không có toán hạng . Lệnh thực hiện phép tính logic giữa hai bit đầu của ngăn xếp , nội dung ngăn xếp được kéo lên 1 bit.

Trong LAD :

Trong STL



```
LD      I0.0
LD      I0.1
A       I0.2
OLD
=       Q0.1
```

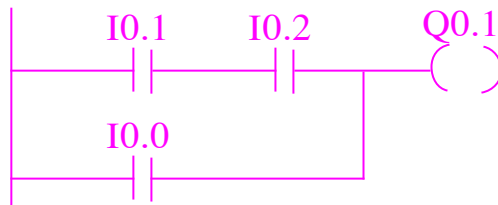
Tác động vào ngăn xếp

Truoc	LD I0.0	Sau	LD I0.1	I0.1	A I0.2	I0.1 A I0.2	OLD	(I0.1 A I0.2) O I0.0
S0		I0.0		I0.1		I0.1 A I0.2		(I0.1 A I0.2) O I0.0
S1		S0		I0.0		I0.0		S0
S2		S1		S0		S0		S1
S3		S2		S1		S1		S2
S4		S3		S2		S2		S3
S5		S4		S3		S3		S4
S6		S5		S4		S4		S5
S7		S6		S5		S5		S6
S8		S7		S6		S6		
		S8		S7				

Ta có thể tối thiểu hóa hàm Logic nhờ vào phép tính trong đại số Bool như sau ở ví dụ trên ta có :

$Q0.1 = I0.0 \vee (I0.1 \wedge I0.2)$, nhờ tính giao hoán của phép hoặc ta có :

$Q0.0 = (I0.1 \wedge I0.2) \vee I0.0$, ta đổi thành :



```
LD      I0.1
A       I0.2
O       I0.0
=       Q0.1
```

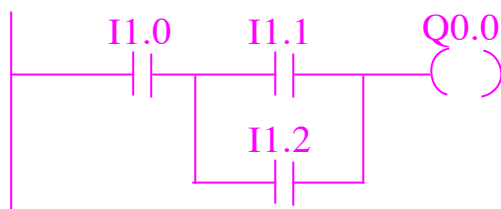
Tương tự ta có :

Lệnh ALD

$Q0.0 = I1.0 \wedge (I1.1 \vee I1.2)$

Ta có :

Trong LAD :

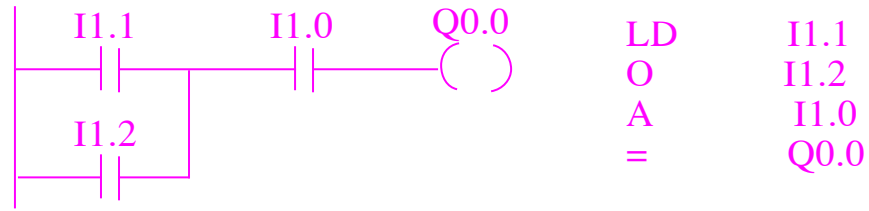


Trong STL

```
LD      I1.0
LD      I1.1
O       I1.2
ALD
=       Q0.0
```

Nhờ tính giao hoán của phép và ta có thể đổi thành :

$Q0.0 = (I1.1 \vee I1.2) \wedge I1.0$, ta đổi thành :



2.1.4. Bài tập ứng dụng.

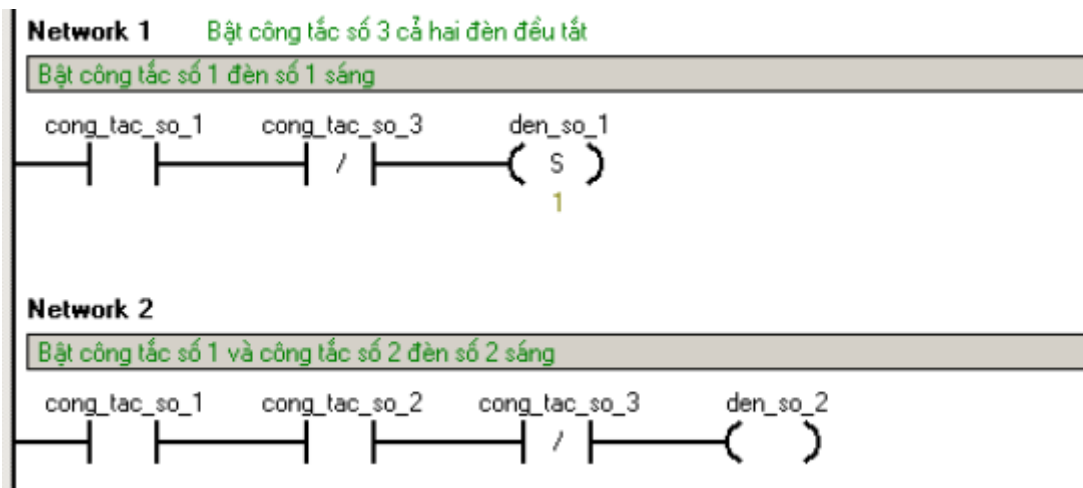
BÀI TẬP SỐ 1.

Đề bài: Viết chương trình điều khiển đèn chiếu sáng theo quy luật sau:

- Bật công tắc số 1 → Đèn số 1 sáng.
- Bật công tắc số 1 và số 2 → Đèn số 2 sáng.
- Bật công tắc số 3 → Hai đèn đều tắt.

THỰC HIỆN CHƯƠNG TRÌNH.

- Mở phần mềm lập trình Micro Win.
- Mở một project mới: File/New.
- Đặt tên biến: Vào Symbol Table trên màn hình rồi đặt tên biến.
- Viết chương trình điều khiển: Vào Program Block/Main để viết chương trình. Vào Bit logic để lấy các tiếp điểm (đầu vào), cuộn dây, đèn (đầu ra), hoặc dùng các phím nóng.



BÀI TẬP SỐ 2.

Đề bài: Viết chương trình điều khiển và vận hành động cơ không đồng bộ ba pha quay một chiều sử dụng nút ấn để điều khiển chạy và dừng.

* Bước 1: Phân tích quy trình làm việc

- Xác định quy trình làm việc của phụ tải:

Mạch điều khiển động cơ không đồng bộ 3 pha quay một chiều có thể mô tả quy trình hoạt động như sau:

Ấn nút Start, động cơ M chạy

Ấn nút Stop, động cơ M dừng

Bảo vệ quá tải cho động cơ dùng rơ le nhiệt RN.

- Xác định mối quan hệ logic của tín hiệu đầu vào và đầu ra

+ Lựa chọn thiết bị điều khiển:

Nút ấn Start: thường mở

Nút ấn Stop: thường đóng

Tiếp điểm rơ le nhiệt RN: thường đóng

* Bước 2: Khai báo địa chỉ đầu vào- đầu ra:

- Địa chỉ đầu vào:

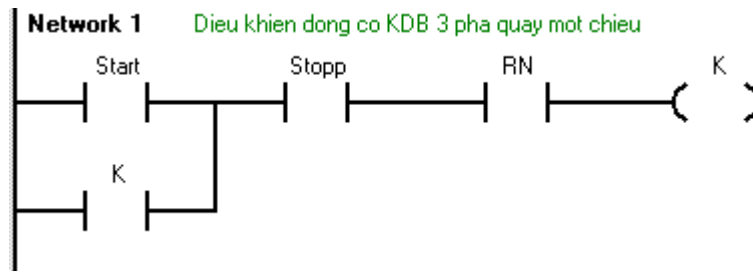
Tín hiệu đầu vào	Địa chỉ	Chức năng
Start	I0.0	Nút ấn mở máy, thường mở, động cơ quay
Stop	I0.1	Nút dừng động cơ, thường đóng
RN	I0.2	Tiếp điểm thường đóng của rơ le nhiệt để bảo vệ quá tải động cơ

Địa chỉ đầu ra:

Tín hiệu đầu ra	Địa chỉ	Chức năng
K	Q0.0	Cuộn dây của công tắc tơ K

* Bước 3: Vẽ sơ đồ thiết kế mạch điều khiển:

Trên cơ sở quy trình làm việc và địa chỉ vào/ra ta tiến hành viết chương trình trên phần mềm Step 7 Micro/win như sau:



Sau khi viết chương trình chúng ta dùng chương trình mô phỏng SIMULINK S7 200 để kiểm tra các chức năng của mạch theo giản đồ thời gian đã có.

* Bước 4: Kết nối cơ cấu chấp hành, nạp chương trình chạy cơ cấu chấp hành

- Kết nối cơ cấu chấp hành.

Để điều khiển đầu ra là cuộn dây công tắc tơ K, ta chọn PLC loại AC/DC/RLY có: các tín hiệu vào là +24VDC ứng với mức logic 1 và 0VDC ứng với mức logic 0, công ra rơ le.

*Bài tập thực hành

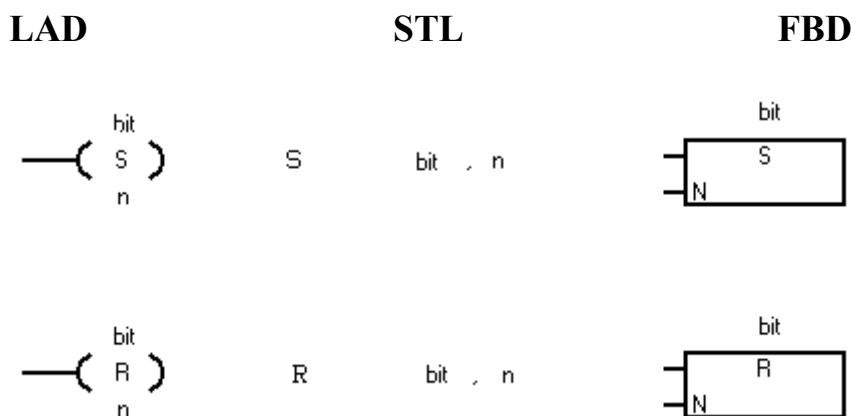
Lập trình PLC điều khiển 1 động cơ không đồng bộ 3 pha quay 1 chiều theo trình tự sau: Ấn nút Start1, động cơ quay với tốc độ chậm -> ấn nút Start2 động cơ quay với tốc độ nhanh -> ấn nút Stop động cơ dừng. Bảo vệ quá tải động cơ bằng rơ le nhiệt.

2.2.Các lệnh ghi / xóa giá trị cho tiếp điểm

2.2.1.Lệnh SET (S) và RESET (R)

- Vùng chứa lệnh: vào Bit logic để lấy lệnh.

- Các ngôn ngữ viết cho lệnh trong PLC.



Toán hạng của lệnh: I, Q, M, SM, T, C, V, S, L

Kiểu dữ liệu: BOOL

bit: địa chỉ của bit lệnh ứng với các toán hạng trên.

n: Số địa chỉ liên tiếp tính từ địa chỉ đầu tiên bit đã cho trong lệnh.

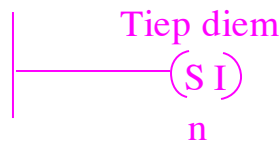
- Tác dụng của lệnh:

+ Trong LAD: Lệnh có tác dụng đóng (với lệnh SET) hoặc ngắt (với lệnh RESET) một mảng gồm n giá trị kể từ tiếp điểm cho trong lệnh.

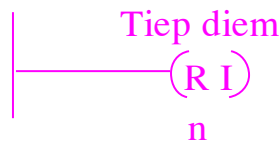
+ Trong STL: Đặt giá trị lô gic 1 (với lệnh S) hoặc đặt giá trị logic 0 (với lệnh R) cho một mảng n bit kể từ bit cho trong lệnh .

* Với lệnh ghi xóa tức thời.

- Trong LAD



Đóng tức thời một mảng gồm n tiếp điểm kể từ tiếp điểm cho trong lệnh



Ngắt tức thời một mảng gồm n tiếp điểm kể từ tiếp điểm cho trong lệnh

- Trong STL:

SI: Tiếp điểm n Đặt tức thời giá trị lô gic 1 cho một mảng n bit kể từ bit cho trong lệnh .

RI: Tiếp điểm n Đặt tức thời giá trị lô gic 0 cho một mảng n bit kể từ bit cho trong lệnh .

*Mạch nhớ R - S.

- Vùng chứa lệnh: vào Bit logic để lấy lệnh.

- Tác dụng của lệnh RS: Lệnh có tác dụng đóng (đặt giá trị logic lên 1) nếu chân S1 tích cực và tín hiệu chân R không tích cực hoặc ngắt nếu tín hiệu chân R là tích cực.

- Tác dụng của lệnh SR: Lệnh có tác dụng đóng (đặt giá trị logic lên 1) nếu chân S1 tích cực và tín hiệu chân R không tích cực hoặc ngắt nếu tín hiệu

chân R là tích cực ở lần tác động đầu tiên. Sau lần đầu tiên, nếu tín hiệu đặt vào chân S là tích cực thì đầu ra luôn luôn ở mức logic 1.

2.2.2. Các ví dụ

Đề bài: Viết chương trình điều khiển hai động cơ hoạt động bằng nút ấn điều khiển dừng và mở, có đèn báo trạng thái làm việc (RUN) và ngừng (STOP) của hai động cơ.

Bước 1: Phân tích yêu cầu công nghệ.

- Ấn M1 → động cơ 1 làm việc, đồng thời đèn báo RUN sáng báo có động cơ đang làm việc.

- Ấn M2 → động cơ 2 làm việc.

- Ấn D → hai động cơ ngừng làm việc, đồng thời đèn báo STOP sáng báo có động cơ đang làm việc.

Bước 2: Qui định địa chỉ vào/ra

Bước 3: Viết chương trình điều khiển.

Bước 4: Mô phỏng, chạy thử.

- Thao tác thử các đầu vào (M1, M2, D) để kiểm tra trạng thái của đầu ra.

2.3. Timer

Các lệnh Timer đều nằm trong tập lệnh Timer hoặc nhóm khối. Dưới đây là một số lệnh tiêu biểu.

2.3.1. On - Delay Timer (TON).

- Vùng chứa lệnh: vào Timer, lấy lệnh TON

* Chức năng của TON là tạo ra một thời gian trễ của tín hiệu ra so với tín hiệu cấp cho TON.

Trong ngôn ngữ LAD:

Trong đó:

IN: là cổng vào nhận tín hiệu dạng BOOL cho phép TON hoạt động

PT: là cổng nhận giá trị tạo thời gian trễ đặt trước cho TON, toán hạng là: IW, VW, QW, MW, SW, SMW, LW, AIW, T, C, *VD, *LD, Constant. Giá trị đặt phải là một số nguyên dương, có giá trị max là 32.767.

???ms: là độ phân giải của TON

Txxx: là số hiệu của TON

Tùy theo số hiệu của TON ta có bảng phân bố độ phân giải sau:

Lệnh	Độ phân giải	Giá trị cực đại	Địa chỉ
TON	1 ms	32,767 s	T0 ÷ T96
	10 ms	327,67 s	T33 ÷ T36, T97 ÷ T100
	100 ms	3276,7 s	T37 ÷ T63, T101 ÷ T127

* Nguyên tắc làm việc của rơ le thời gian TON:

Khi cấp tín hiệu vào cổng IN, TON bắt đầu đếm số nguyên dương. Giá trị đếm được của TON tăng dần từ 0, đến khi \geq giá trị đặt ở PT thì TON tác động, bit tiếp điểm thường đóng mở ra và thường mở đóng lại. Nếu cắt tín hiệu cấp vào IN thì TON ngừng đếm, giá trị đếm được bị reset về 0, TON ngừng tác động. Nếu cấp tín hiệu trở lại IN thì TON đếm lại từ đầu. Nếu duy trì tín hiệu cấp vào IN thì TON sẽ đếm đến giá trị cực đại là 32767 từ dừng đếm và giữ giá trị đếm được là 32767, TON vẫn tác động.

2.3.2. Retentive On - Delay Timer (TONR).

Retentive On - Delay Timer (TONR) là bộ thời gian trễ có nhớ. Trong ngôn ngữ LAD, TONR được viết như sau:

Trong đó:

IN: là cổng vào nhận tín hiệu dạng BOOL cho phép TONR hoạt động

PT: là cổng nhận giá trị tạo thời gian trễ đặt trước cho TONR, toán hạng là: IW, VW, QW, MW, SW, SMW, LW, AIW, T, C, *VD, *LD, Constant.

???ms: là độ phân giải của TONR

Txxx: là số hiệu của TONR

Để tạo thời gian trễ thì đầu vào IN được kích, nếu giá trị đếm tức thời bằng hoặc lớn hơn giá trị đặt trước PT thì bộ TONR có giá trị 1. Chỉ có thể xóa TONR bằng lệnh R, còn khi đầu vào IN được kích từ 1 về 0 thì giá trị của TONR = 0 còn giá trị đếm tức thời vẫn đọc giữ nguyên

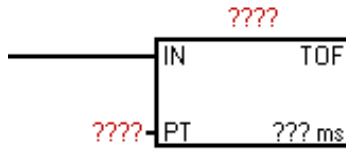
2.3.3. OFF- Delay Timer (TOF).

- TOF là viết tắt của OFF – delay – Timer

Là bộ tạo thời gian trễ cắt đầu ra output với thời gian cố định sau khi đầu vào input = 0 (off)

Khi input = 1 (ON) , đầu ra output = 1 tức thì . Khi input = 0 thời gian bắt đầu được đếm và khi giá trị đếm bằng giá trị đặt thì đầu ra output = 0

Ký hiệu trong LAD là :



Trong đó:

IN: là cổng vào nhận tín hiệu dạng BOOL cho phép TOF hoạt động

PT: là cổng nhận giá trị tạo thời gian trễ đặt trước cho TOF, toán hạng là: IW, VW, QW, MW, SW, SMW, LW, AIW, T, C, *VD, *LD, Constant.

???: là độ phân giải của TOF

Txxx: là số hiệu của TOF

Tùy theo số hiệu của TOF ta có bảng phân bố độ phân giải sau:

Lệnh	Độ phân giải	Giá trị cực đại	Địa chỉ
TON	1 ms	32,767 s	T0 ÷ T96
	10 ms	327,67 s	T33 ÷ T36, T97 ÷ T100
	100 ms	3276,7 s	T37 ÷ T63, T101 ÷ T127

2.3.4. Bài tập ứng dụng Timer.

Trong sản xuất công nghiệp, có rất nhiều các cơ cấu máy móc trong một hệ thống hoặc các thiết bị khác nhau trong một máy sản xuất cần hoạt động tuần tự theo thời gian. Dưới đây là một số ứng dụng cụ thể:

- Động cơ có công suất lớn khi khởi động ở chế độ sao/tam giác, cần không chế để chế độ sao làm việc trước chế độ tam giác trong một khoảng thời gian.

- Trong máy cắt gọt kim loại, động cơ bơm dầu thường làm việc trước động cơ chính để bôi trơn.

* BÀI TẬP ỨNG DỤNG ĐIỀU KHIỂN BẰNG THỜI GIAN TRỄ.

BÀI TẬP SỐ 1:

Yêu cầu: Viết chương trình điều khiển hai động cơ làm việc tuần tự theo thời gian, có đèn báo trạng thái làm việc của hai động cơ.

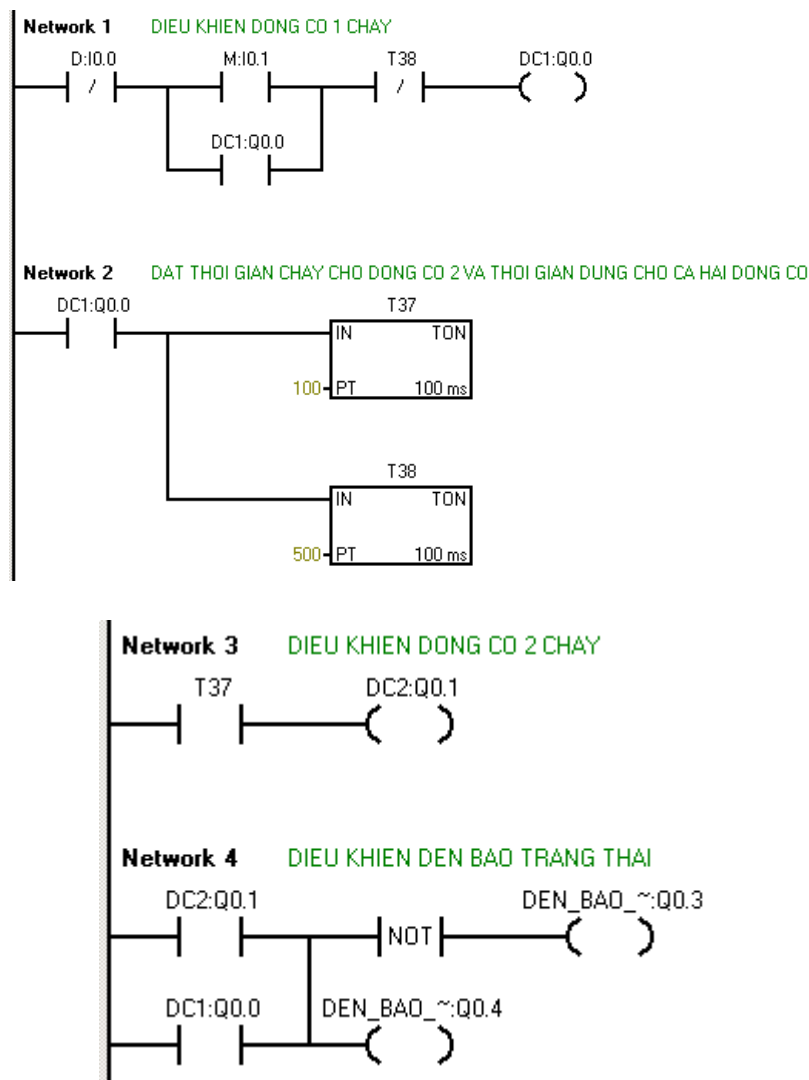
* Bước 1: Phân tích yêu cầu công nghệ.

- Ấn M → động cơ 1 làm việc, đồng thời đèn báo RUN sáng báo có động cơ đang làm việc.
- Sau thời gian đặt → động cơ 2 làm việc.
- Ấn D → hai động cơ ngừng làm việc, đồng thời đèn báo STOP sáng báo có động cơ đang làm việc.

Bước 2: Qui định địa chỉ vào/ra

D	I0.0	NUT AN DIEU KHIEN DUNG
M	I0.1	NUT AN DIEU KHIEN MO MAY
DC1	Q0.0	DONG CO 1
DC2	Q0.1	DONG CO 2
DEN_BAO_STOP	Q0.3	DEN BAO HAI DONG CO KHONG HOAT DONG
DEN_BAO_RUN	Q0.4	DEN BAO HAI DONG CO DANG HOAT DONG

* Bước 3: Viết chương trình điều khiển



2.4.Counter

2.4.1. Bộ đếm lên (Counter up).

* Vùng chứa lệnh: vào Counters.

* Cách khai báo:

- Chân CU: Khai báo tín hiệu đếm tiến theo sườn lên. Dạng dữ liệu BOOL

- PV: Khai báo giá trị đặt trước cho bộ đếm. Dạng dữ liệu WORD. Giá trị đặt là số nguyên dương, có giá trị max là 32.767.

- R: Khai báo tín hiệu xóa hay tín hiệu Reset bộ đếm. Dạng dữ liệu BOOL

- Cxxx: Khai báo địa chỉ cho bộ đếm. Dạng dữ liệu WORD. Địa chỉ của bộ đếm nằm trong dải từ 0 đến 255.

* Nguyên lý hoạt động: Bộ đếm lên (CTU) là bộ đếm số sườn lên của tín hiệu logic đầu vào (CU), tức là đếm số lần thay đổi trạng thái logic từ 0 đến 1 của tín hiệu. Số sườn xung đếm được, được ghi vào thanh ghi 2 byte của bộ đếm, gọi là thanh ghi C-word. Cứ mỗi sườn xung tín hiệu thì giá trị đếm của bộ đếm Cxx tăng 1. Giá trị này có thể tăng đến giá trị cao nhất của nó. Bộ đếm chỉ dừng lại nếu giá trị đếm đạt đến +32767.

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

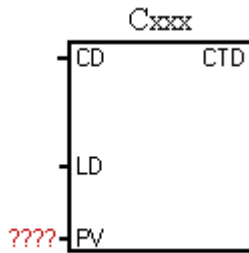
Bộ đếm sẽ được reset (0), nếu ngõ vào đặt tại R của nó được đóng mạnh (bằng 1) hoặc khi lệnh R (reset) được thực hiện với C-bit. Khi bộ đếm được reset, cả C-word và C-bit đều nhận giá trị 0.

Vùng địa chỉ của bộ đếm được trong CPU 214 là từ C0 đến C47 và C80 đến C127.

Giá trị tới hạn giới hạn đếm đặt ở ngõ vào PV đưa ra có thể là hằng số hoặc có thể là từ như sau: VW, T, C, IW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.

2.4.2. Bộ đếm xuống (Counter down).

* Cách khai báo.



- Khai báo tín hiệu đếm lùi theo sườn lên tại chân CD. Dạng dữ liệu BOOL

- Khai báo giá trị đặt trước cho bộ đếm PV. Dạng dữ liệu WORD. Giá trị đặt là số nguyên dương, có giá trị max là 32.767.

- Cxxx: Khai báo địa chỉ cho bộ đếm. Dạng dữ liệu WORD. Địa chỉ của bộ đếm nằm trong dải từ 0 đến 255.

- Khai báo tín hiệu đặt hay tín hiệu Set bộ đếm tại LD. Dạng dữ liệu BOOL

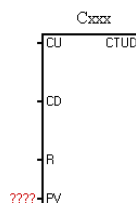
* Nguyên lý hoạt động: Bộ đếm xuống (CTD) sẽ có giá trị đếm bằng giá trị đặt khi gặp sườn xung lên ở ngõ vào LD. Bộ đếm sẽ đếm lùi khi gặp sườn lên của xung vào cổng đếm lùi, ký hiệu là **CD** trong LAD. Giá trị đếm của bộ đếm giảm 1 ở mỗi sườn xung lên ở ngõ vào CD. Giá trị này có thể giảm đến giá trị thấp nhất của nó và bộ đếm chỉ dừng lại nếu giá trị đếm đạt đến +0..

Nếu giá trị đếm tức thời = 0, thì C-bit có giá trị bằng 1. Còn các trường hợp khác C-bit có giá trị bằng 0.

Giá trị tới hạn giới hạn đếm đặt ở ngõ vào PV đưa ra có thể là hằng số hoặc có thể là từ như sau: VW , T, C, IW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.

2.4.3. Bộ đếm lên/ xuống (Counter up - down).

* Cách khai báo.



- Khai báo tín hiệu đếm tiến theo sườn lên tại chân CU. Dạng dữ liệu BOOL

- Khai báo tín hiệu đếm lùi theo sườn lên tại chân CD. Dạng dữ liệu BOOL

- Khai báo giá trị đặt trước cho bộ đếm PV. Dạng dữ liệu WORD. Giá trị đặt là số nguyên dương, có giá trị max là 32.767.

- Cxxx: Khai báo địa chỉ cho bộ đếm. Dạng dữ liệu WORD. Địa chỉ của bộ đếm nằm trong dải từ 0 đến 255.

- Khai báo tín hiệu xóa hay tín hiệu Reset bộ đếm tại R. Dạng dữ liệu BOOL

* Nguyên lý hoạt động: Bộ đếm lên/xuống (**CTUD**) đếm lên khi gặp sườn lên của xung vào cổng đếm lên, ký hiệu là **CU** trong LAD. Giá trị đếm của bộ đếm tăng 1 ở mỗi sườn xung lên ở ngõ vào. Giá trị này có thể tăng đến giá trị cao nhất của nó. Bộ đếm chỉ dừng lại nếu giá trị đếm đạt đến +32767. Bộ đếm **CTUD** đếm xuống khi gặp sườn lên của xung vào cổng đếm xuống, ký hiệu là **CD** trong LAD. Giá trị đếm của bộ đếm giảm đi 1 ở mỗi sườn xung lên ở ngõ vào CD. Bộ đếm chỉ dừng lại, nếu giá trị đếm đạt đến -32767.

Nếu giá trị đếm tức thời \geq giá trị đặt trước ở ngõ vào PV, thì C-bit có giá trị bằng 1. Còn các trường hợp khác C-bit có giá trị bằng 0.

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

- Khi ngõ vào R có giá trị logic bằng 1

- Dùng lệnh R (reset) để reset C-bit bộ đếm.

Giá trị tới hạn giới hạn đếm đặt ở ngõ vào PV đưa ra có thể là hằng số hoặc có thể là từ như sau: VW, T, C, IW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC.

2.4.4. Bài tập ứng dụng

BÀI TẬP SỐ 1:

Yêu cầu: Viết chương trình điều khiển hai băng tải vận chuyển hàng hóa.

* Bước 1: Phân tích yêu cầu công nghệ.

- Ấn M → động cơ 1 làm việc, đồng thời đèn báo RUN sáng báo có động cơ đang làm việc.

- Sau thời gian đặt → động cơ 2 làm việc.

- Nếu số sản phẩm đóng gói vào thùng đã đủ, hai băng tải tạm dừng, chờ lệnh làm việc tiếp theo.

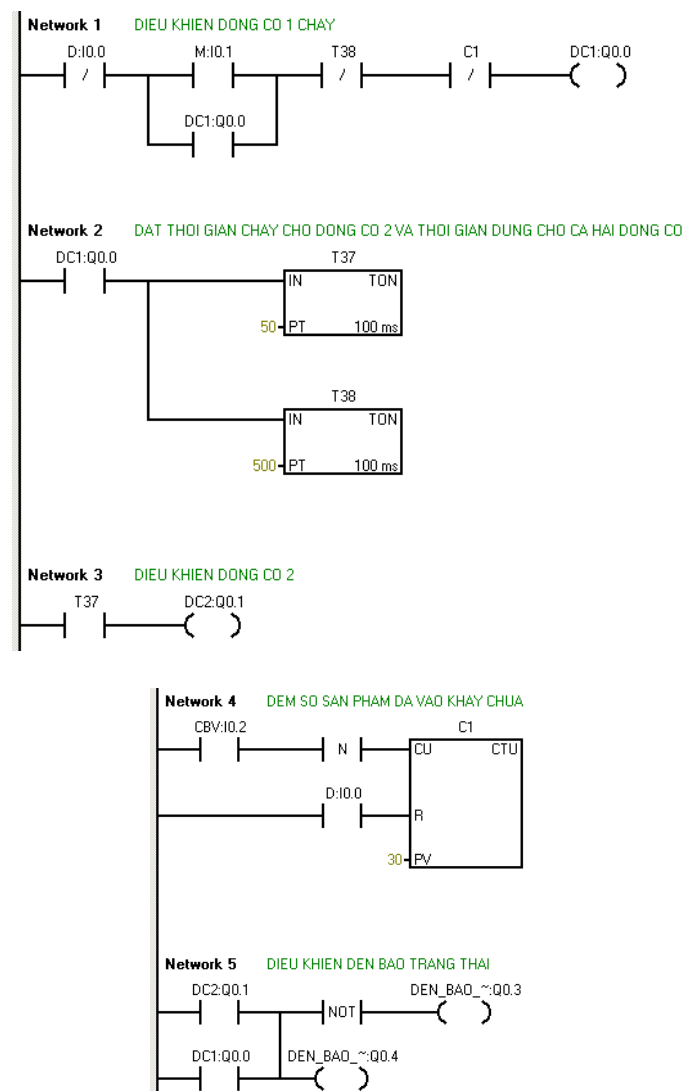
- Sau thời gian đặt trước, nếu số sản phẩm đóng gói vẫn không đủ, hai băng tải cũng tạm dừng chờ sản phẩm.

- Ấn D → hai động cơ ngừng làm việc, đồng thời đèn báo STOP sáng báo có động cơ đang làm việc.

Bước 2: Quy định địa chỉ vào/ra

Symbol	Address	Comment
D	I0.0	NÚT AN ĐIỀU KHIỂN DỪNG
M	I0.1	NÚT AN ĐIỀU KHIỂN MƠ MÁY
CBV	I0.2	CẢM BIẾN PHÁT HIỆN VẬT VÀO HỘP
DC1	Q0.0	ĐỘNG CƠ 1
DC2	Q0.1	ĐỘNG CƠ 2
DEN_BAO_STOP	Q0.3	ĐÈN BÁO HAI ĐỘNG CƠ KHÔNG HOẠT ĐỘNG
DEN_BAO_RUN	Q0.4	ĐÈN BÁO HAI ĐỘNG CƠ ĐANG HOẠT ĐỘNG

Bước 3: Viết chương trình điều khiển.



Bước 4: Mô phỏng, chạy thử.

BÀI TẬP SỐ 2:

Đề bài: Viết chương trình điều khiển các động cơ khởi động và dừng theo trình tự.

Bước 1. Phân tích quy trình làm việc

* Xác định yêu cầu bài toán:

Lập trình PLC điều khiển khởi động trực tiếp và dừng 3 động cơ KĐB 3 pha ro to lồng sóc theo đúng tuần tự sau:

- Ấn nút "khởi động", động cơ 1Đ chạy ngay, sau 5'' động cơ 2Đ chạy, sau 5'' nữa động cơ 3Đ chạy.

- Ấn nút "dừng", động cơ 3Đ dừng ngay, sau 3'' động cơ 2Đ dừng, sau 3'' nữa động cơ 1Đ dừng.

- Các động cơ được bảo vệ quá tải bằng rơ le nhiệt, khi có 1 động cơ quá tải cả 3 động cơ cùng dừng.

- Các nút ấn "dừng" dùng tiếp điểm thường đóng, nút "khởi động" là dùng tiếp điểm thường mở.

- Mạch có nút "dừng khẩn cấp". Khi ấn nút này tất cả các động cơ dừng cùng lúc.

Để điều khiển các động cơ dùng công tắc tơ K1, K2, K3. Đóng – cắt điện và bảo vệ ngắn mạch dùng Aptomat AT. Bảo vệ quá tải dùng rơ le nhiệt RN1, RN2, RN3.

* Xác định mối quan hệ logic của tín hiệu đầu vào và đầu ra

- Lựa chọn thiết bị điều khiển và chấp hành:

Nút ấn "khởi động" M: thường mở

Nút ấn "dừng" D1: thường đóng

Nút ấn "dừng khẩn cấp" D: thường đóng

Tiếp điểm rơ le nhiệt RN: thường đóng

Công tắc tơ K1, K2, K3 điều khiển 3 động cơ

Bước 2: Thiết kế mạch điều khiển bằng phân tử logic:

* Khai báo địa chỉ đầu vào- đầu ra:

- Địa chỉ đầu vào:

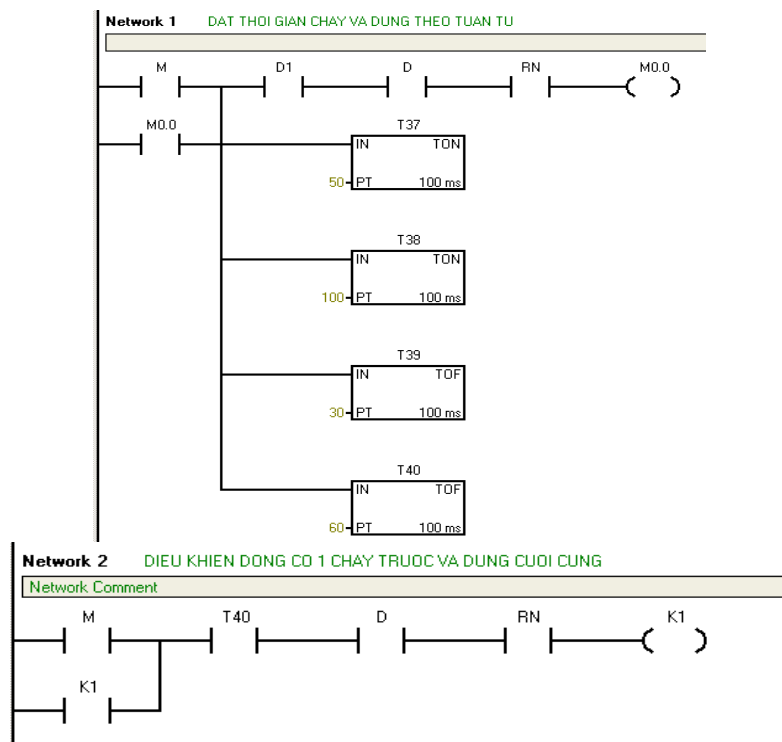
Tín hiệu vào	Địa chỉ	Chức năng
M	I0.0	Nút ấn "khởi động", thường mở
D1	I0.1	Nút "dừng", thường đóng
D	I0.2	Nút "dừng khẩn cấp", thường đóng
RN	I0.3	Tiếp điểm thường đóng của rơle nhiệt để bảo vệ quá tải động cơ

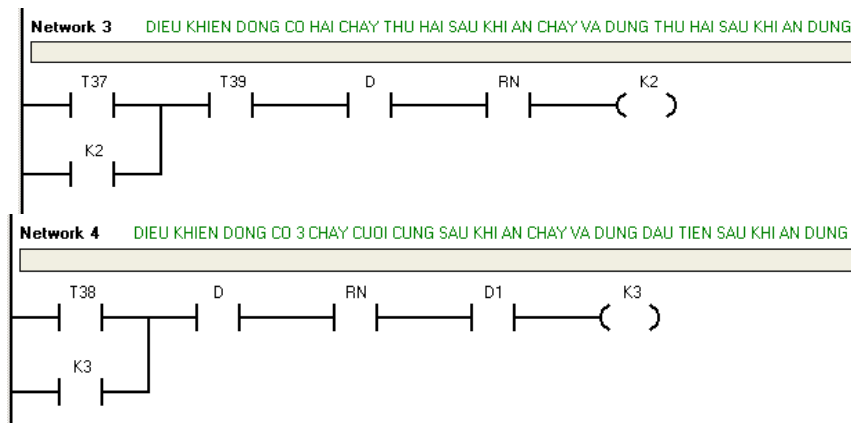
Địa chỉ đầu ra:

Tín hiệu ra	Địa chỉ	Chức năng
K1	Q0.0	Cuộn dây của công tắc tơ K1
K2	Q0.1	Cuộn dây của công tắc tơ K2
K3	Q0.2	Cuộn dây của công tắc tơ K3

* Bước 3: Viết chương trình điều khiển.

Trên cơ sở quy trình làm việc và địa chỉ vào/ra ta tiến hành viết chương trình trên phần mềm Step 7 Micro/win như sau:





Bước 4: Mô phỏng, chạy thử.

BÀI TẬP SỐ 3:

Đề bài: Lập trình PLC điều khiển động cơ KĐB 3 pha roto lồng sóc khởi động trực tiếp, quay hai chiều có hãm dừng.

BÀI TẬP SỐ 3:

Đề bài: Viết chương trình điều khiển băng tải đếm sản phẩm theo yêu cầu công nghệ.

Bước 1. Phân tích quy trình làm việc

* Xác định yêu cầu bài toán:

Lập trình PLC điều khiển một hệ thống băng tải đếm sản phẩm như sau:

- Băng tải 1 có 1 cảm biến: Khi có sản phẩm chạy qua cảm biến phát tín hiệu băng 1.

- Băng tải 2 có 1 cảm biến tương tự băng tải 1

Yêu cầu:

- Ấn nút khởi động băng tải 2 chạy.

- Khi có đủ 12 sản phẩm qua băng tải 1, sau 3'' băng tải 1 tạm dừng, băng tải 2 chạy. Khi cảm biến băng tải 2 phát hiện có sản phẩm, sau 3'' băng tải 2 tạm dừng, băng tải 1 tiếp tục chạy và bộ đếm băng tải 1 đếm từ đầu. Quá trình trên lặp lại liên tục.

- Ấn nút dừng, cả hai băng tải cùng dừng.

- Các băng tải được bảo vệ bằng rơ le nhiệt, khi 1 băng tải quá tải cả hai băng tải đều phải dừng.

Để điều khiển các băng tải ta dùng công tắc tơ K. Để đóng cắt điện và bảo vệ ngắn mạch ta dùng Aptomat AT.

Bài 3

Các phép toán số của PLC

Mục tiêu:

Trình bày được các phép toán so sánh, các phép toán số.

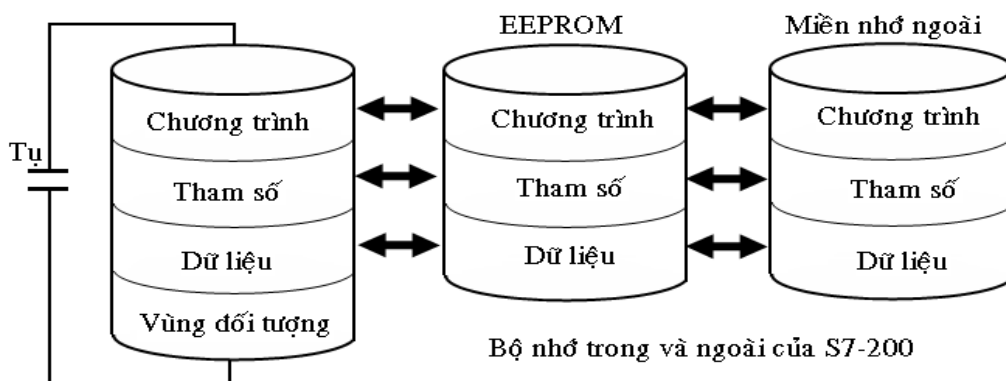
Lập trình, kết nối, chạy thử các phép toán so sánh, phép toán số, trong các bài toán thực tế.

Tích cực, chủ động và sáng tạo trong học tập.

3.1. Chức năng truyền dẫn

3.1.1. Truyền Byte, Word, DoubleWord

Bộ nhớ của S7-200 được chia thành 4 vùng với 1 tụ điện (hình 2.3) có nhiệm vụ duy trì dữ liệu trong 1 khoảng thời gian nhất định khi mất nguồn. Bộ nhớ của S7-200 có tính năng động cao đọc và ghi được trong toàn vùng, loại trừ phần các bit nhớ đặc biệt được ký hiệu bởi SM (special memory) chỉ có thể truy nhập để đọc.



Hình 2.3: bộ nhớ trong và ngoài của S7-200

Vùng chương trình: là miền bộ nhớ được sử dụng để lưu giữ các lệnh chương trình. Vùng này thuộc kiểu non-volatile đọc/ghi được.

Vùng tham số: là miền lưu giữ các tham số như: từ khóa, địa chỉ trạm... Cũng giống như vùng chương trình, vùng tham số thuộc kiểu non-volatile đọc/ghi được.

Vùng dữ liệu: được sử dụng để cất các dữ liệu của chương trình bao gồm các kết quả các phép tính, hằng số được định nghĩa trong chương trình, bộ đệm

truyền thông...Một phần của vùng nhớ này (200 byte đầu tiên đối với CPU 212, 1K byte đầu tiên đối với CPU 214) thuộc kiểu non-volatile đọc/ghi được.

Vùng đối tượng: Timer, bộ đếm, bộ đếm tốc độ cao và các ngõ vào/ ra tương tự được đặt trong vùng nhớ cuối cùng. Vùng này không thuộc kiểu non-volatile nhưng đọc/ghi được.

Hai vùng nhớ cuối có ý nghĩa quan trọng trong việc thực hiện một chương trình, do vậy được trình bày cụ thể như sau.

3.1.2. Truyền mộ vùng nhớ dữ liệu

Vùng dữ liệu là một miền nhớ động. Nó có thể được truy nhập theo từng bit, từng byte, từng từ đơn (word) hoặc theo từng từ kép và được sử dụng làm miền lưu trữ dữ liệu cho các thuật toán, các hàm truyền thông, lập bảng, các hàm dịch chuyển, xoay vòng thanh ghi, con trỏ địa chỉ...

Ghi các dữ liệu kiểu bảng bị hạn chế rất nhiều vì các dữ liệu kiểu bảng thường chỉ được sử dụng cho những mục đích nhất định.

Vùng dữ liệu lại được chia ra thành những miền nhớ nhỏ với các công dụng khác nhau. Chúng được ký hiệu bằng các chữ cái đầu của tên tiếng Anh, đặc trưng cho công dụng riêng của chúng như sau: + V: Variable memory

+ I: Input image register

+ O: Output image register

+ M: Internal memory bits

+ SM: Special memory bits

Tất cả các miền này đều có thể truy nhập được theo từng bit, từng byte, từng từ đơn (word-2byte) hoặc từ kép (2word)

Hình sau mô tả vùng dữ liệu của CPU212 và CPU214 (hình 2.4)

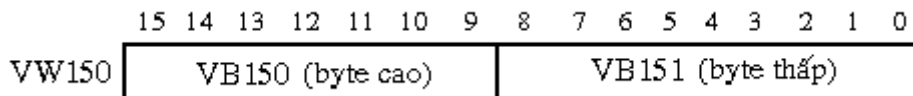
	CPU212 7 6 5 4 3 2 1 0	CPU214 7 6 5 4 3 2 1 0						
Miền V (đọc/ghi)	<table border="1" style="margin: auto;"> <tr><td>V0</td></tr> <tr><td>⋮</td></tr> <tr><td>V1023</td></tr> </table>	V0	⋮	V1023	<table border="1" style="margin: auto;"> <tr><td>V0</td></tr> <tr><td>⋮</td></tr> <tr><td>V4095</td></tr> </table>	V0	⋮	V4095
V0								
⋮								
V1023								
V0								
⋮								
V4095								
Vùng đệm ngõ vào (I) (đọc/ghi)	<table border="1" style="margin: auto;"> <tr><td>I0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>I7.x(x=0÷7)</td></tr> </table>	I0.x(x=0÷7)	⋮	I7.x(x=0÷7)	<table border="1" style="margin: auto;"> <tr><td>I0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>I7.x(x=0÷7)</td></tr> </table>	I0.x(x=0÷7)	⋮	I7.x(x=0÷7)
I0.x(x=0÷7)								
⋮								
I7.x(x=0÷7)								
I0.x(x=0÷7)								
⋮								
I7.x(x=0÷7)								
Vùng đệm ngõ ra (Q) (đọc/ghi)	<table border="1" style="margin: auto;"> <tr><td>Q0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>Q7.x(x=0÷7)</td></tr> </table>	Q0.x(x=0÷7)	⋮	Q7.x(x=0÷7)	<table border="1" style="margin: auto;"> <tr><td>Q0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>Q7.x(x=0÷7)</td></tr> </table>	Q0.x(x=0÷7)	⋮	Q7.x(x=0÷7)
Q0.x(x=0÷7)								
⋮								
Q7.x(x=0÷7)								
Q0.x(x=0÷7)								
⋮								
Q7.x(x=0÷7)								
Vùng nhớ nội (M) (đọc/ghi)	<table border="1" style="margin: auto;"> <tr><td>M0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>M15.x(x=0÷7)</td></tr> </table>	M0.x(x=0÷7)	⋮	M15.x(x=0÷7)	<table border="1" style="margin: auto;"> <tr><td>M0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>M31.x(x=0÷7)</td></tr> </table>	M0.x(x=0÷7)	⋮	M31.x(x=0÷7)
M0.x(x=0÷7)								
⋮								
M15.x(x=0÷7)								
M0.x(x=0÷7)								
⋮								
M31.x(x=0÷7)								
Vùng nhớ đặc biệt (SM) (chỉ đọc)	<table border="1" style="margin: auto;"> <tr><td>SM0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>SM29x(x=0÷7)</td></tr> </table>	SM0.x(x=0÷7)	⋮	SM29x(x=0÷7)	<table border="1" style="margin: auto;"> <tr><td>SM0.x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>SM29x(x=0÷7)</td></tr> </table>	SM0.x(x=0÷7)	⋮	SM29x(x=0÷7)
SM0.x(x=0÷7)								
⋮								
SM29x(x=0÷7)								
SM0.x(x=0÷7)								
⋮								
SM29x(x=0÷7)								
Vùng nhớ đặc biệt (đọc/ghi)	<table border="1" style="margin: auto;"> <tr><td>SM30x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>SM45x(x=0÷7)</td></tr> </table>	SM30x(x=0÷7)	⋮	SM45x(x=0÷7)	<table border="1" style="margin: auto;"> <tr><td>SM30x(x=0÷7)</td></tr> <tr><td>⋮</td></tr> <tr><td>SM85x(x=0÷7)</td></tr> </table>	SM30x(x=0÷7)	⋮	SM85x(x=0÷7)
SM30x(x=0÷7)								
⋮								
SM45x(x=0÷7)								
SM30x(x=0÷7)								
⋮								
SM85x(x=0÷7)								

► Địa chỉ truy nhập được qui ước với công thức:

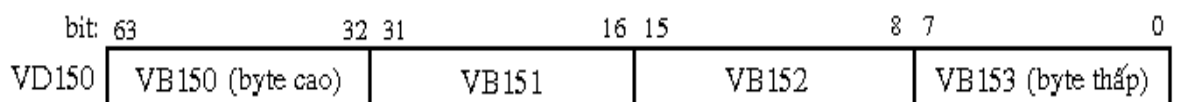
Truy nhập theo bit: Tên miền (+) địa chỉ byte (+) ● (+) chỉ số bit. Ví dụ V105.4 chỉ bit 4 của byte 150 thuộc miền V.

Truy nhập theo byte: Tên miền (+) B (+) địa chỉ của byte trong miền. Ví dụ: VB150 chỉ byte 150 thuộc miền V.

Truy nhập theo từ đơn: Tên miền (+) W (+) địa chỉ byte cao của từ trong miền. Ví dụ: VW150 chỉ từ đơn gồm 2 byte 150 và 151 thuộc miền V, trong đó byte 150 có vai trò là byte cao trong từ.



Truy nhập theo từ kép: Tên miền (+) D (+) địa chỉ byte cao của từ trong miền. Ví dụ: VD150 chỉ từ kép gồm 4 byte 150,151,152 và 153 thuộc miền V, trong đó byte 150 có vai trò là byte cao và byte 153 là byte thấp trong từ kép.



Tất cả các byte thuộc vùng dữ liệu đều có thể truy nhập bằng con trỏ. Con trỏ được định nghĩa trong miền V hoặc các thanh ghi AC1, AC2, AC3. Mỗi con trỏ chỉ địa chỉ gồm 4 byte (từ kép). Quy ước sử dụng con trỏ để truy nhập như sau:

& Địa chỉ byte:(cao) là toán hạng lấy địa chỉ của byte, từ, hoặc từ kép.

Ví dụ: A C1= & VB150, thanh ghi AC1 chứa địa chỉ byte 150 thuộc miền V

VD100= & VW150, từ kép VD100 chứa địa chỉ byte cao (VB150) của từ đơn VW150.

AC2= & VD150, thanh ghi AC2 chứa địa chỉ byte cao (VB150) của từ kép VD150.

+ Con trỏ: là toán hạng lấy nội dung của byte, từ hoặc từ kép mà con trỏ đang chỉ vào.

+ Ví dụ như với phép gán địa chỉ trên thì

+ AC1, lấy nội dung của byte VB150

+ VD100, lấy nội dung của từ đơn VW150

+ AC2, lấy nội dung của từ kép VD150.

Phép gán địa chỉ và sử dụng con trỏ như trên cũng có tác dụng với những thanh ghi 16 bit của Timer, bộ đếm thuộc vùng đối tượng sẽ được trình bày dưới đây.

3.2. Chức năng so sánh

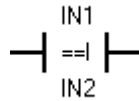
* Cũng như các phép toán so sánh số học, trong PLC S7-200 cũng có tất cả các phép so sánh sau: =, ≥, ≤, <, >, ≠

* Vùng chứa lệnh: vào Compare.

* Tác dụng của hệ lệnh so sánh: hệ lệnh so sánh thực hiện so sánh các số dạng Byte, Word, Interger, Double Word hoặc Real. Nếu kết quả so sánh là đúng thì giá trị logic trả về sẽ là 1, nếu sai sẽ là 0. Trong STL kết quả so sánh được lưu vào bit đầu tiên của ngăn xếp. Các toán hạng trong một phép so sánh phải cùng dạng dữ liệu.

* Giới thiệu lệnh một số lệnh so sánh Compare:

- Lệnh so sánh bằng với số nguyên dương.



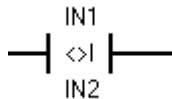
Cách viết lệnh:

Toán hạng: IW, QW, MW, SW, T, C, VW, LW, AC, ...

Kiểu dữ liệu: số nguyên dương

Lệnh tác động khi IN1=IN2.

Lệnh so sánh khác:



- Các lệnh so sánh với dữ liệu kiểu Integer khác: với các lệnh dưới đây chúng ta sử dụng hai số hạng n1, n2 trong các phép so sánh, trong đó n1: là số hạng trên-số hạng bị so sánh, n2 là số hạng dưới-số hạng được so sánh)

	<p>Tiếp điểm này sẽ đóng khi giá trị trong thanh ghi n1 bằng giá trị trong thanh ghi n2.</p> <p>Toán hạng: n1, n2 là 1 word.</p>
	<p>Tiếp điểm này sẽ đóng khi giá trị trong thanh ghi n1 > hoặc = giá trị trong thanh ghi n2.</p> <p>Toán hạng: n1, n2 là 1 word.</p>
	<p>Tiếp điểm này sẽ đóng khi giá trị trong thanh ghi n1 < hoặc = giá trị trong thanh ghi n2.</p> <p>Toán hạng: n1, n2 là 1 word.</p>
	<p>Tiếp điểm này sẽ đóng khi giá trị trong thanh ghi n1 khác giá trị trong thanh ghi n2.</p> <p>Toán hạng: n1, n2 là 1 word.</p>
	<p>Tiếp điểm này sẽ đóng khi giá trị trong thanh ghi n1 lớn hơn giá trị trong thanh ghi n2.</p> <p>Toán hạng: n1, n2 là 1 word.</p>
	<p>Tiếp điểm này sẽ đóng khi giá trị trong thanh ghi n1 nhỏ hơn giá trị trong thanh ghi n2.</p> <p>Toán hạng: n1, n2 là 1 word.</p>

Lệnh tác động khi $IN1 \neq IN2$.

3.2.1. Chức năng dịch chuyển

* Vùng chứa lệnh: Vào Move

* Có các hàm di chuyển dữ liệu như sau: MOV-B, MOV-DB, MOV-W, MOV-D, MOV-R. Các lệnh này có tác dụng dịch chuyển hoặc sao chép dữ liệu trong IN chuyển vào OUT nếu đầu vào EN được kích hoạt.

Toán hạng của lệnh là: Byte, Word, Double Word hoặc Real.

Giá trị đầu vào có thể là các ô nhớ (chứa dữ liệu bên trong) hoặc là hằng số. Đầu ra OUT bắt buộc phải là các ô nhớ với kích thước đúng với lệnh chuyển và kiểu dữ liệu của đầu vào IN. Để hiểu hơn về cách sử dụng các hàm di chuyển dữ liệu ta xét một số hàm dưới đây.

* Tác dụng của hàm MOV-B: lệnh có tác dụng lấy giá trị trong IN và chuyển vào OUT nếu đầu vào EN được kích hoạt.

Toán hạng của lệnh: cả IN, OUT đều là 1Byte.

3.2.2. Chức năng chuyển đổi

* Vùng chứa lệnh: Vào Converter.

* Hệ lệnh này cho phép đổi từ kiểu dữ liệu này sang kiểu dữ liệu khác. Chúng ta có thể sử dụng hệ lệnh này để chuyển đổi các kiểu dữ liệu khác nhau khi cần thực hiện các phép toán học giữa các loại dữ liệu khác nhau.

* Hàm B_I: hàm có tác dụng lấy giá trị trong IN (1 byte) chuyển thành 1 word và chứa ra OUT nếu chân EN được kích hoạt.

Toán hạng của hàm:

Với IN là: VB, IB, QB, MB, SB, SMB, LB, AC, Constant, *AC, *VD, *LD. Dữ liệu kiểu BYTE.

Với OUT là: VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC. Dữ liệu kiểu INT.

- Để hiểu hơn về cách sử dụng hàm ta lấy ví dụ như sau: viết chương trình đưa dữ liệu 255 vào vùng nhớ MW2, dùng lệnh chuyển đổi.

3.3. Đồng hồ thời gian thực

Các bước sử dụng thời gian thực của S7-200 như sau:

Bước 1: Đặt thời gian thực cho CPU của S7-200.

- Vào PLC → Time of Day Clock

Bước 2: Đặt thời gian thực của CPU vào chương trình và lập trình:

- Đọc thời gian thực vào chương trình: vào Clock → Read_RTC

Trong đó: EN – đầu vào kích hoạt.

T – Dữ liệu thời gian đọc được từ CPU về chương trình, khai báo tương ứng như bảng sau:

Thứ tự T-Byte	Dải dữ liệu	Nội dung
0	Năm (0-99)	Dữ liệu năm (giá trị BCD)
1	Tháng (1-12)	Dữ liệu tháng (giá trị BCD)
2	Ngày (1-31)	Dữ liệu ngày (giá trị BCD)
3	Giờ (0-23)	Dữ liệu giờ (giá trị BCD)
4	Phút (0-59)	Dữ liệu phút (giá trị BCD)
5	Giây (0-59)	Dữ liệu giây (giá trị BCD)
6	00	Đưa dữ liệu về 00
7	Ngày trong tuần (1-7)	Dữ liệu ngày trong tuần, mặc định Chủ nhật = 1 (giá trị BCD)

Dữ liệu đầu tiên cần đọc thường là dữ liệu của năm (trong vùng nhớ MB0).

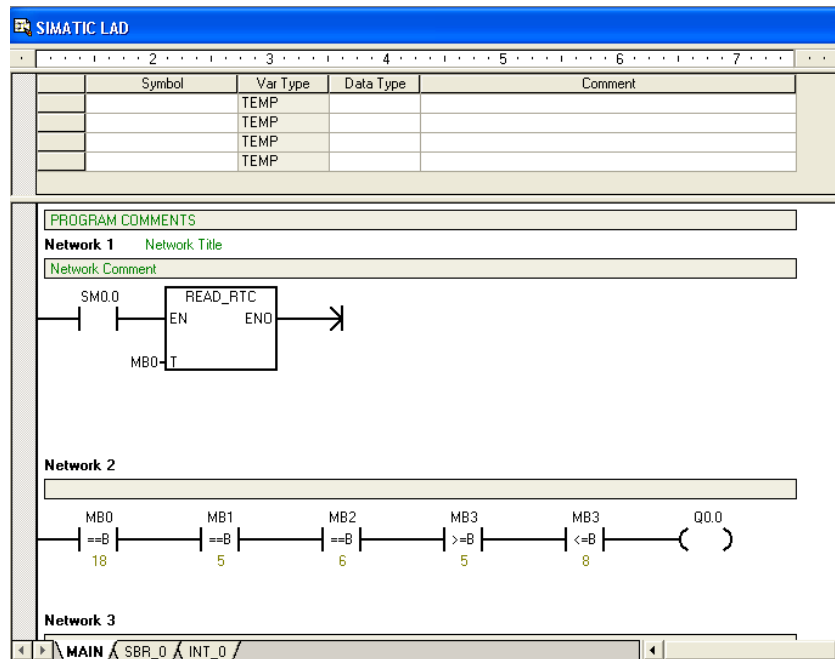
Các dữ liệu trên chương trình đều hiểu là số Hex, vì thế khi sử dụng vào phép toán so sánh ta phải quy đổi ra số thập phân.

BÀI TẬP SỐ 1.

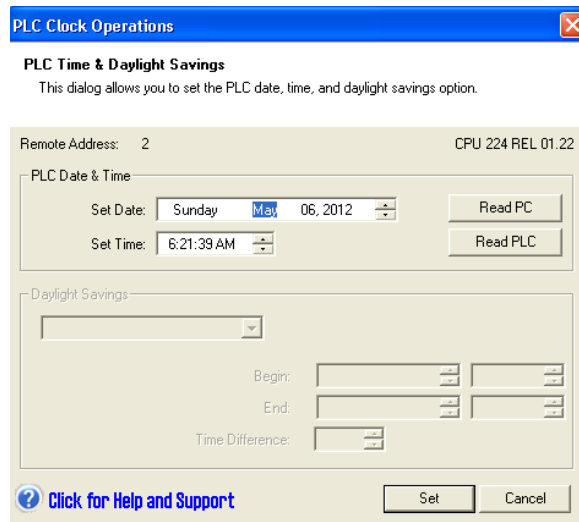
Đề bài: Viết chương trình điều khiển đèn tương ứng với địa chỉ Q0.0 sáng vào thời gian thực từ 5h-8h ngày 6/5/2012.

* Các bước thực hiện:

- Viết chương trình so sánh dữ liệu thời gian:



- Bước 2: Đặt thời gian thực cho CPU



Bài 4

Lắp đặt mô hình điều khiển bằng PLC

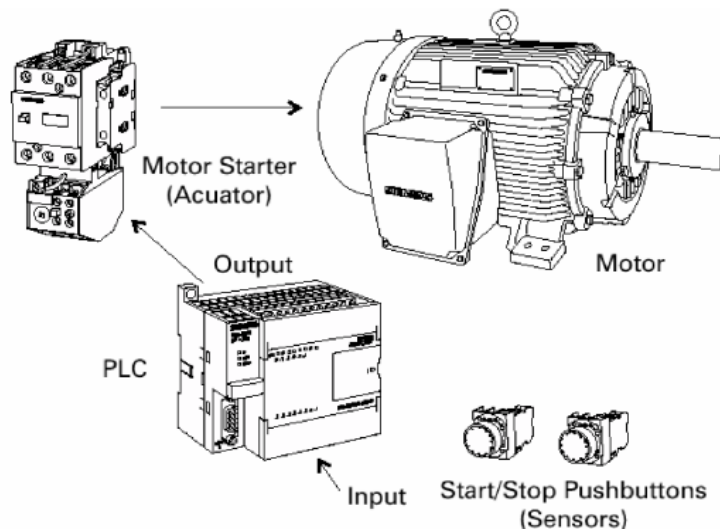
Mục tiêu:

- Phân tích qui trình công nghệ của một số mạch máy sản xuất.
- Lập trình được một số mạch ứng dụng thường gặp trong thực tế.
- Nạp chương trình, vận hành, kiểm tra mạch hoạt động theo yêu cầu kỹ thuật
- Tích cực, chủ động và sáng tạo trong học tập.

4.1. Giới thiệu

Trong nhiều ứng dụng của PLC, phải nói đến ứng dụng PLC trong lĩnh vực trong hệ thống sản xuất công nghiệp, điều khiển robot, điều khiển quá trình, mạng thu nhận dữ liệu, điều khiển trình tự máy phân loại, điều khiển giám sát.... Trong bài này, ta sẽ đi sâu vào nghiên cứu các ứng dụng PLC trong điều khiển động cơ nhằm phục vụ điều khiển các thiết bị trong lĩnh vực sản xuất công nghiệp.

Để điều khiển truyền động điện của thiết bị máy móc nói chung và máy công cụ trong công nghiệp nói riêng, người ta dùng rất nhiều thiết bị và khí cụ điện khác nhau để thực hiện các nhiệm vụ khác nhau. Nhờ dây dẫn điện chúng ta nối liền các bộ phận lại với nhau để tạo nên một dạng sơ đồ chung gọi là sơ đồ điện, nhằm để thực hiện những chức năng theo một yêu cầu nhất định.



Mạch điều khiển động cơ

- Động cơ không đồng bộ rotor lồng sóc :



Động cơ không đồng bộ là máy điện xoay chiều, làm việc theo nguyên lý cảm ứng điện từ, có tốc độ của rotor n khác với tốc độ từ trường quay trong máy n_1 .

Công tắc tơ



Rơ le nhiệt



Rơ le nhiệt dùng để bảo vệ động cơ và mạch điện khỏi bị quá tải, nó không tác động tức thời theo dòng điện mà cần phải có thời gian để phát nóng.

Rơ le nhiệt làm việc theo nguyên lý tác dụng nhiệt của dòng điện, cấu tạo bên trong là phiến kim loại kép: một tấm có hệ số giãn nở bé và một tấm có hệ số giãn nở lớn. Khi đốt nóng do dòng điện I , có thể dùng trực tiếp cho dòng điện đi qua, hoặc dây điện trở bao quanh

Bộ phận đốt nhiệt 1 đầu nối tiếp với mạch điện chính của thiết bị cần bảo vệ (tự động cắt điện). Khi dòng điện chạy trong mạch điện tăng lên quá mức qui định (động cơ bị quá tải) thì nhiệt lượng tỏa ra làm cho phiến kim loại kép 3 cong lên phía trên (về phía có hệ số giãn nở bé). Nhờ lực kéo của lò xo 5, đòn bẩy 4 sẽ quay và mở tiếp điểm 2 làm cho mạch điện tự động cắt điện. Khi bộ phận đốt nóng nguội đi, thanh kim loại kép hết cong, nhấn nút 6 là có thể đưa rotor nhiệt về vị trí cũ, thì tiếp điểm 2 lại đóng lại.

Nút nhấn:

Có các loại nút nhấn sau:

+ Nút nhấn thường mở: khi tác động từ trên xuống thì tiếp điểm đóng lại dẫn điện để mồi mạch điện. Khi bỏ tay ra nhờ lò xo phản, tiếp điểm lại trở về vị trí ban đầu hở mạch.

+ Nút nhấn thường đóng: khi tác động từ trên xuống thì tiếp điểm mở ra để hở mạch điện. Khi bỏ tay ra nhờ lò xo phản, tiếp điểm lại trở về vị trí ban đầu đóng mạch.

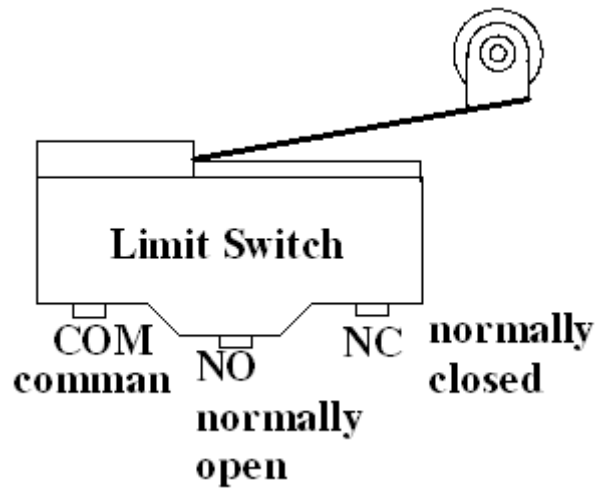
+ Nút nhấn kép: là nút nhấn kết hợp cả nút nhấn thường đóng và thường mở lên trên một nút nhấn.

+ Ngoài ra còn có nút nhấn dừng khẩn cấp có cấu tạo giống như các nút nhấn trên nhưng có thêm bộ phận xoay, dùng để nhấn dừng khẩn khi có sự cố. Nút nhấn này cũng có 1 tiếp điểm thường đóng và một tiếp điểm thường mở.



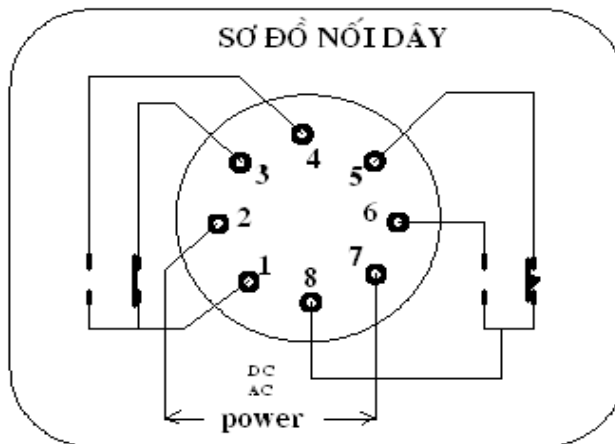
► Công tắc hành trình (Limit Switch)

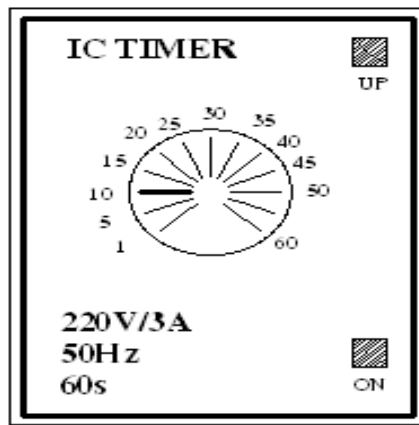
Công tắc hành trình là một loại khí cụ điện, tác động bằng lực cơ học để đóng mở các tiếp điểm thường đóng hay thường mở. Công tắc sẽ tác động (đổi trạng thái đóng, mở của tiếp điểm) khi bộ phận của máy đi qua những vị trí đã xác định trong giới hạn làm việc của nó (gọi là công tắc hành trình), hay gọi chung là công tắc giới hạn.



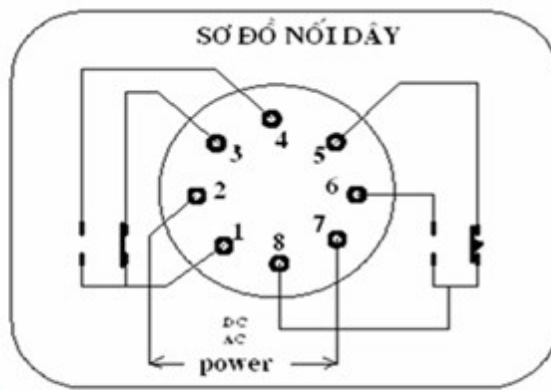
➤ Rơ le thời gian

Rơ le thời gian IC (IC Timer) hiện nay được sử dụng rộng rãi vì có nhiều ưu điểm hơn so với kiểu cơ khí. Rơ le thời gian kiểu IC có kích thước nhỏ gọn, với độ chính xác cao, dễ điều chỉnh và dải điều chỉnh rộng từ 0.05 giây đến 24 giờ tùy theo loại rơ le thời gian. IC Timer cũng dùng được cho cả dòng điện AC và DC.





Sơ đồ và hình dáng của Timer IC

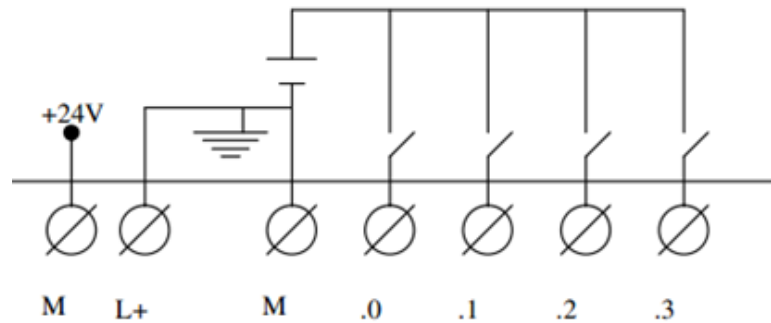


Xi lanh:

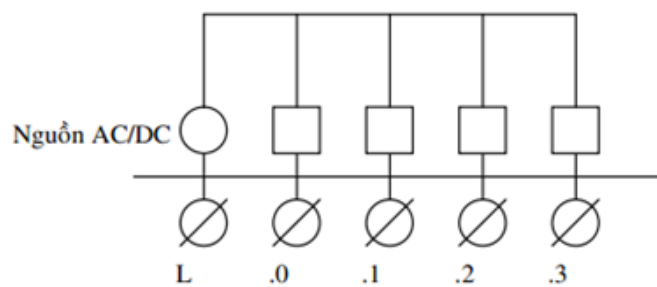


4.2.Cách kết nối dây

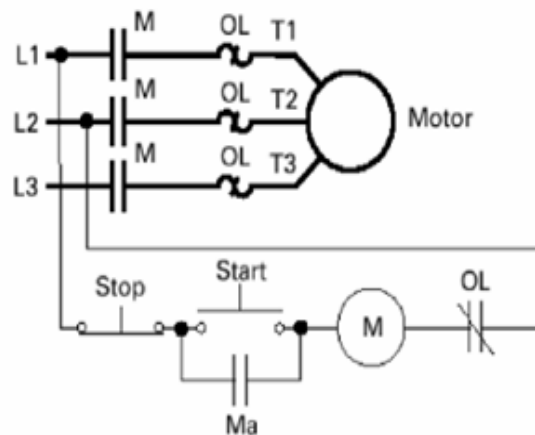
4.2.1.Kết nối ngõ vào:



4.2.2.Kết nối ngõ ra



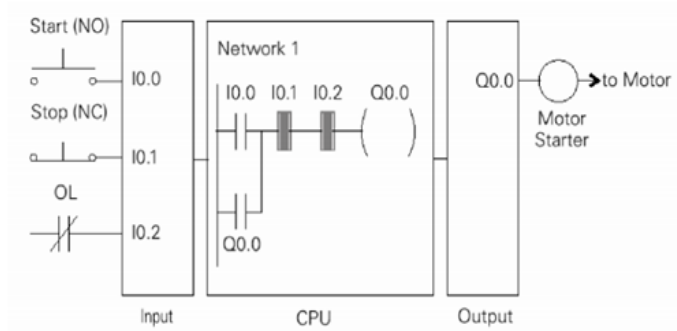
► kết nối bằng mạch rơ le:



Trong thí dụ này động cơ được khởi động (M) được mắc nối tiếp với một nút nhấn bình thường hở NO (nút Start), nút nhấn bình thường đóng NC (Stop) và các tiếp điểm bình thường đóng rơ-le quá tải (OL).

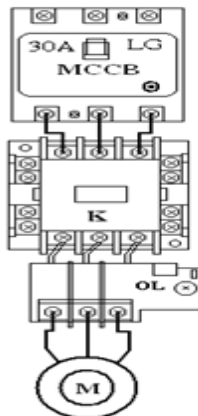
Khi nhấn Start thì có dòng điện đi qua mạch làm khởi động động cơ, nó làm đóng các tiếp điểm M và Ma tương ứng của động cơ. Khi nhấn Start thì động cơ vẫn hoạt động do các tiếp điểm M, Ma đóng. Động cơ sẽ tiếp tục chạy cho đến khi nhấn nút Stop hay khi có quá tải làm mở các tiếp xúc OL.

➤ Chương trình PLC:

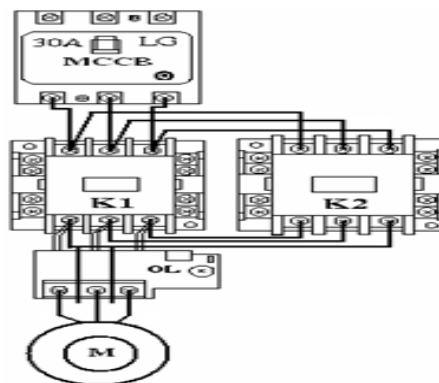


Nút nhấn Start (NO) được nối vào ngõ vào thứ nhất I0.0, nút nhấn Stop (NC) nối vào ngõ vào thứ hai I0.1 và các tiếp điểm rờ le quá tải OL được nối vào ngõ vào thứ ba I0.2. Một mạch AND 3 ngõ vào này tạo nên mạch điều khiển trong Network 1. Bit trạng thái I0.1 ở mức logic 1 vì nút Stop là loại NC; bit trạng thái I0.2 ở mức logic 1 vì các tiếp điểm OL đóng. Bộ điều khiển động cơ được nối vào ngõ ra Q0.0.

➤ Cách nối dây mạch động lực khởi động từ đơn cho động cơ:



➤ Cách nối dây mạch động lực đảo chiều động cơ dùng khởi động từ kép:

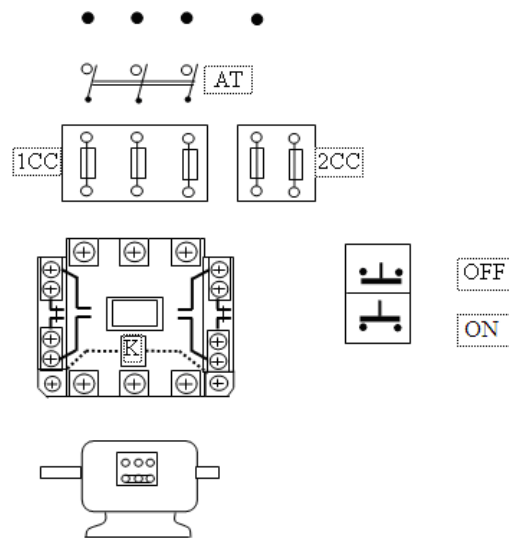


4.3. Các mô hình và bài tập ứng dụng

4.3.1. Điều khiển van điện từ hai cuộn dây.

* Chọn bảng thực hành bằng gỗ (hoặc tử điện), kích thước 0,4m x 0,4m để vẽ sơ đồ gá lắp thiết bị theo tỷ lệ của khổ giấy A4. Dựa trên sơ đồ nguyên lý vẽ sơ đồ lắp đặt thiết bị tương ứng từ trên xuống dưới. Sau đó, dựa trên sơ đồ gá lắp thiết bị dùng máy bắn vít, kim tuốc nơ vít bắt chặt thiết bị vào bảng mạch (bắt các thiết bị có kích thước lớn và ở giữa mạch trước, các thiết bị xung quanh bắt sau).

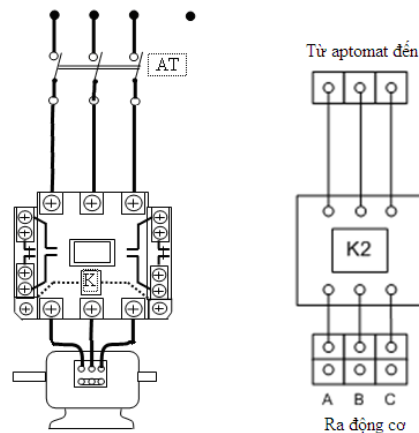
Vậy ta có thể bố trí các thiết bị trên bảng mạch như sau:



Sơ đồ bố trí thiết bị trên bảng mạch thực hành.

* Vẽ sơ đồ đi dây.

- Vẽ phần đi dây động lực (các dây pha đi song song, không chồng chéo lên nhau, màu đậm, các pha chọn các màu khác nhau: pha A- đỏ, pha B: vàng, pha C: xanh): vẽ đi dây từ trên xuống dưới, từ trái qua phải



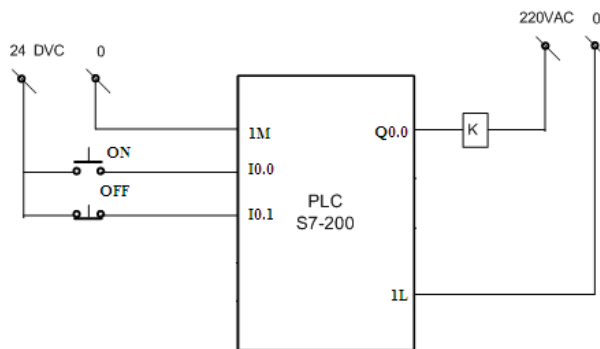
Sơ đồ đi dây mạch động lực.

- Vẽ phần đi dây mạch điều khiển (chọn nét vẽ mảnh, có thể dùng một hoặc hai màu, hạn chế nhiều đường đi dây, nên đi dây theo một số đường để khi lắp ráp dễ dàng bó buộc lại hoặc đi vào trong máng): vẽ từ phần nguồn tới các thiết bị.

Với bài trên ta tiến hành xác định số lượng và chủng loại đầu vào/ra và quy định địa chỉ như sau:

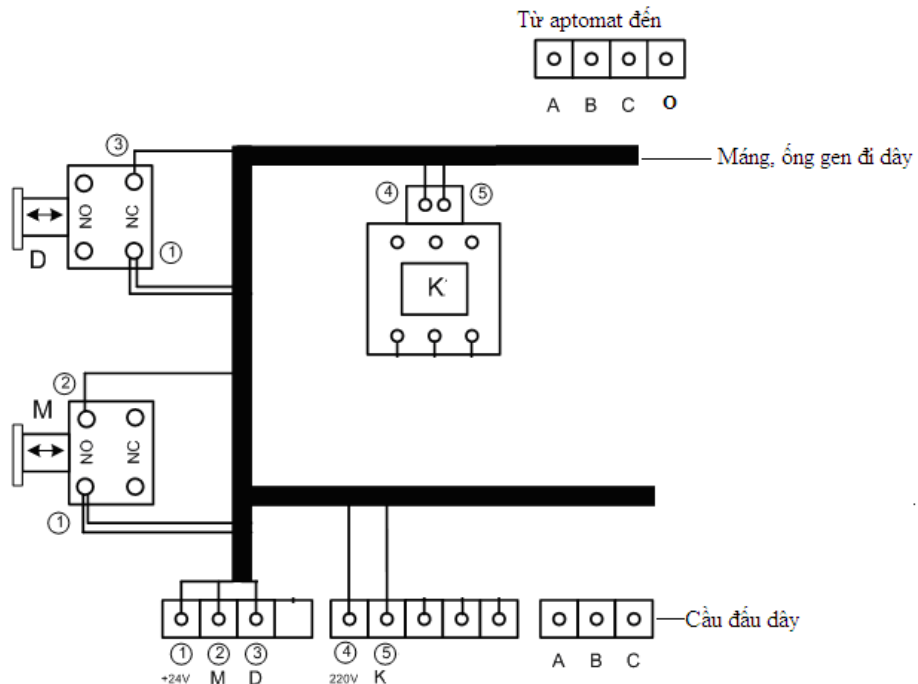
- Đầu vào: Nút ấn điều khiển chạy ON – I0.0 (thường mở)
Nút ấn điều khiển dừng OFF-I0.1 (thường đóng)
- Đầu ra: Cuộn dây công tắc tơ K- Q0.0.

Ta vẽ được sơ đồ nguyên lý đấu nối mạch điều khiển như sau:



Sơ đồ nguyên lý đấu nối từ PLC tới thiết bị chấp hành

Ta vẽ sơ đồ đi dây mạch điều khiển như sau:



Hình 4.4.Sơ đồ đi dây mạch điều khiển từ thiết bị đến phần nối với PLC

* Lắp mạch

* Kiểm tra và lắp các thiết bị vào bảng mạch

* Lắp mạch động lực: đấu nối dây theo nguyên tắc từ trên xuống dưới, từ trái qua phải. Cụ thể: Dùng dây đơn 1,5mm² đi dây từ Aptomat → K → ĐC.

* Với mạch điều khiển:

- Dùng dây đơn hoặc dây bất kỳ đo đo dài giữa các phần cần đi dây. Uốn dây vuông góc tại các điểm gấp khúc và giao nhau để đảm bảo dây đi song song, không chồng chéo.

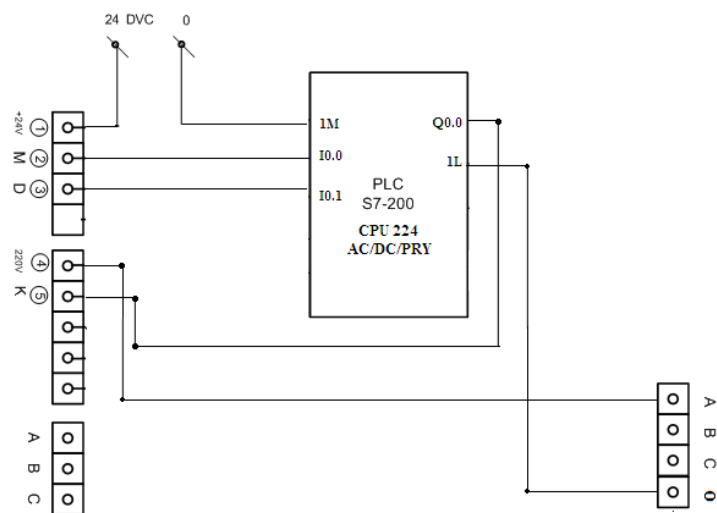
* Kiểm tra trực quan.

Công tắc tơ, nút ấn, cầu chì... không bị nghiêng, dây động lực không bị chồng chéo lên nhau, các đầu cốt không bị hở, không có thiết bị và dây điện thừa..., cảm mạch lên lắc không có thiết bị và dây điện bị bung ra.

+ Đo thông mạch theo sơ đồ.

* Kết nối với PLC và cơ cấu chấp hành.

Khi kết nối PLC với cơ cấu chấp hành cần phải đảm bảo rằng chương trình điều khiển viết bằng phần mềm S7-200 đã được kiểm tra và chạy thử, sau khi thử tác động các trường hợp giả định đạt yêu cầu thì mới tiến hành đấu nối và vận hành. Sơ đồ đấu nối từ PLC tới cơ cấu chấp hành như sau:



Sơ đồ kết nối từ PLC tới cơ cấu chấp hành.

4.3.2. Điều khiển hệ thống cung cấp khí nén.

* Động cơ quay hai chiều được sử dụng rộng rãi trong các máy công nghiệp và dân dụng. Việc điều khiển các máy sản xuất đảo chiều quay hoặc

chuyển động tịnh tiến theo hai chiều thực chất là điều khiển đảo chiều quay các động cơ truyền động cho các cơ cấu của máy, nên cần phải nắm vững cách điều khiển để có thể sử dụng trong thực tế điều khiển. Lập trình PLC giúp điều khiển đảo chiều quay động cơ không đồng bộ 3 pha dễ dàng và thuận tiện.

Các bước giải quyết bài toán:

*** Bước 1: Phân tích quy trình làm việc**

- Xác định quy trình làm việc của phụ tải:

Mạch điều khiển động cơ không đồng bộ 3 pha quay hai chiều có thể mô tả quy trình hoạt động như sau:

Nếu ấn nút MT thì động cơ M quay thuận hoặc ấn nút MN động cơ M quay ngược. Ấn nút D động cơ M dừng. Để điều khiển động cơ M ta dùng công tắc tơ K và cấp điện và bảo vệ ngắn mạch ta dùng Aptomat TA.

Bước 2: Thiết kế mạch điều khiển bằng phân tử logic

* Khai báo địa chỉ đầu vào- đầu ra

- Địa chỉ đầu vào:

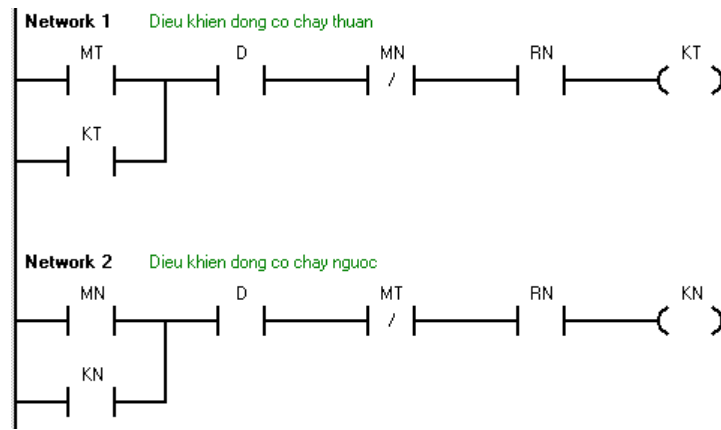
Tín hiệu đầu vào	Địa chỉ	Chức năng
MT	I0.0	Nút ấn mở máy, thường mở, động cơ quay thuận
MN	I0.1	Nút ấn mở máy, thường mở, động cơ quay ngược
D	I0.2	Nút dừng động cơ, thường đóng
RN	I0.3	Tiếp điểm thường đóng của rơle nhiệt để bảo vệ quá tải động cơ

- Địa chỉ đầu ra:

Tín hiệu đầu ra	Địa chỉ	Chức năng
KT	Q0.0	Cuộn dây của công tắc tơ KT
KN	Q0.1	Cuộn dây của công tắc tơ KN

* Vẽ sơ đồ thiết kế mạch điều khiển:

Trên cơ sở Quy trình làm việc và địa chỉ vào/ra ta tiến hành viết chương trình trên phần mềm Step 7 Micro/win như sau:



Sau khi viết chương trình chúng ta dùng chương trình mô phỏng SIMULINK S7 200 để kiểm tra các chức năng của mạch theo giản đồ thời gian đã có.

Bước 3: Kết nối cơ cấu chấp hành, nạp chương trình chạy cơ cấu chấp hành

* Kết nối cơ cấu chấp hành

Để điều khiển đầu ra là cuộn dây công tắc tơ K, ta chọn PLC loại AC/DC/RLY có: các tín hiệu vào là +24VDC ứng với mức logic 1 và 0VDC ứng với mức logic 0, công ra rơ le.

4.3.3. Điều khiển hệ thống cung cấp thủy lực.

Các thiết bị vận tải liên tục dùng để vận chuyển các vật liệu thể hạt, thể cục kích thước nhỏ, các chi tiết ở dạng thành phẩm và bán thành phẩm, hoặc vận chuyển hành khách theo một cung đường nhất định. Thiết bị vận tải liên tục gồm: băng tải, băng chuyền....

Băng tải thường được dùng để vận chuyển vật liệu thể bột mịn, hạt hoặc kích thước nhỏ theo phương nằm ngang hoặc nghiêng với góc nhỏ hơn 30^0 với các cơ cấu kéo đa dạng như băng vải, cao su...

****Bước 1: Phân tích quy trình làm việc***

* Xác định quy trình làm việc của phụ tải .

+ Ấn M, hệ thống chuẩn bị làm việc.

+ Khi có sản phẩm trên băng tải 1 để cảm biến CB1 tác động và có hộp để cảm biến CB3 tác động thì băng tải 1 chạy chuyển sản phẩm vào hộp.

+ Khi cảm biến CB2 đếm được 15 sản phẩm (hộp đầy) thì băng tải 1 dừng, băng tải 2 chạy để đưa hộp ra ngoài. Nếu đặt hộp mới vào vị trí thì băng

tải 2 dừng, đồng thời nếu có sản phẩm qua CB1 thì băng tải 1 chạy. Sau đó hoạt động của dây chuyền lặp lại như trên.

- Ấn nút dừng, cả hai băng tải cùng dừng.

- Các băng tải được bảo vệ bằng rơ le nhiệt, khi 1 băng tải quá tải cả hai băng tải đều phải dừng.

Bước 2: Thiết kế mạch điều khiển bằng phần tử logic.

* Khai báo địa chỉ đầu vào- đầu ra:

- Địa chỉ đầu vào:

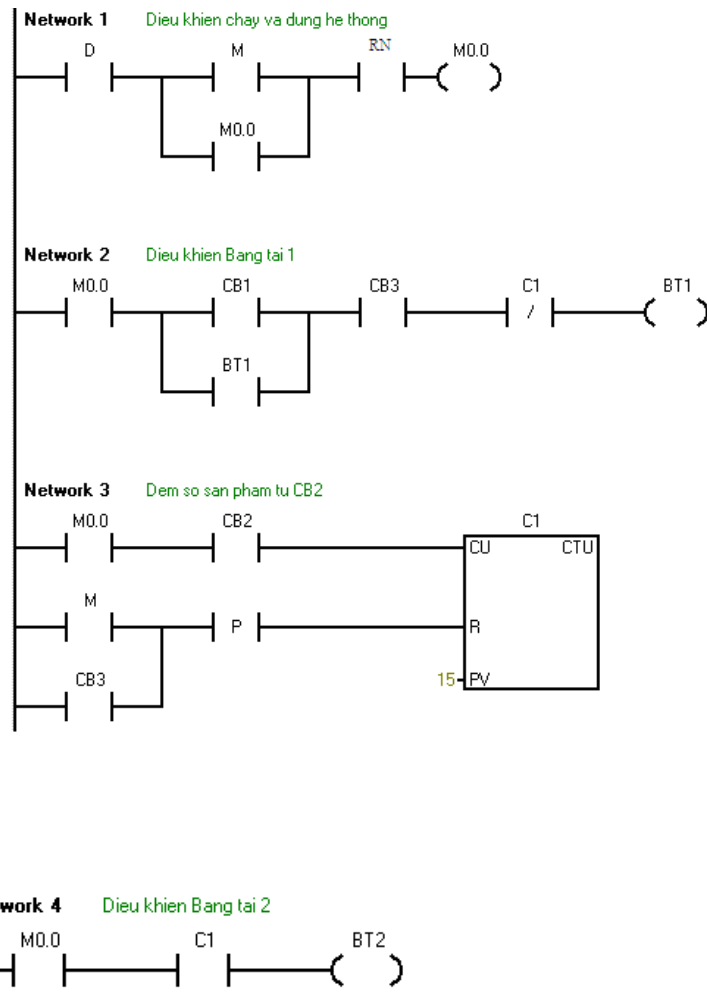
Tín hiệu đầu vào	Địa chỉ	Chức năng
M	I0.0	Nút ấn "chạy", thường mở
D	I0.1	Nút "dừng", thường đóng
RN	I0.4	Tiếp điểm thường đóng của rơle nhiệt để bảo vệ quá tải động cơ
CB1	I0.2	Cảm biến 1
CB2	I0.3	Cảm biến 2
CB3	I0.4	Cảm biến 3

- Địa chỉ đầu ra:

Tín hiệu đầu ra	Địa chỉ	Chức năng
BT1	Q0.0	Cuộn dây của công tắc tơ điều khiển BT1
BT2	Q0.1	Cuộn dây của công tắc tơ điều khiển BT1

* Vẽ sơ đồ thiết kế mạch điều khiển:

Trên cơ sở quy trình làm việc và địa chỉ vào/ra ta tiến hành viết chương trình trên phần mềm Step 7 Micro/win như sau:



Sau khi viết chương trình chúng ta dùng chương trình mô phỏng SIMULINK S7 200 để kiểm tra các chức năng của mạch theo giản đồ thời gian đã có.

*** Bước 3: Kết nối cơ cấu chấp hành, nạp chương trình chạy cơ cấu chấp hành**

- Kết nối cơ cấu chấp hành

Để điều khiển đầu ra là cuộn dây công tắc tơ K, ta chọn PLC loại AC/DC/RLY có: các tín hiệu vào là +24VDC ứng với mức logic 1 và 0VDC ứng với mức logic 0, công ra rơ le.

4.3.4. Điều khiển hệ thống thông gió.

Động cơ không đồng bộ 3 pha công suất lớn khi khởi động có thể gây ra sụt áp và làm giảm chất lượng của nguồn điện. Phương pháp đổi nối Y- Δ thường được sử dụng để giảm dòng khi khởi động nhằm khắc phục nhược điểm này. Đây là phương pháp đơn giản, chắc chắn và sử dụng tin cậy. Chúng ta nên nắm vững cách điều khiển để có thể sử dụng trong thực tế điều khiển. Lập trình

PLC giúp khởi động đôi nối Y- Δ động cơ không đồng bộ 3 pha dễ dàng và thuận tiện.

Các bước giải quyết bài toán:

*** Bước 1: Phân tích quy trình làm việc**

- Xác định quy trình làm việc của phụ tải

Bước 2: Thiết kế mạch điều khiển bằng phần tử logic.

* Khai báo địa chỉ đầu vào- đầu ra

- Địa chỉ đầu vào:

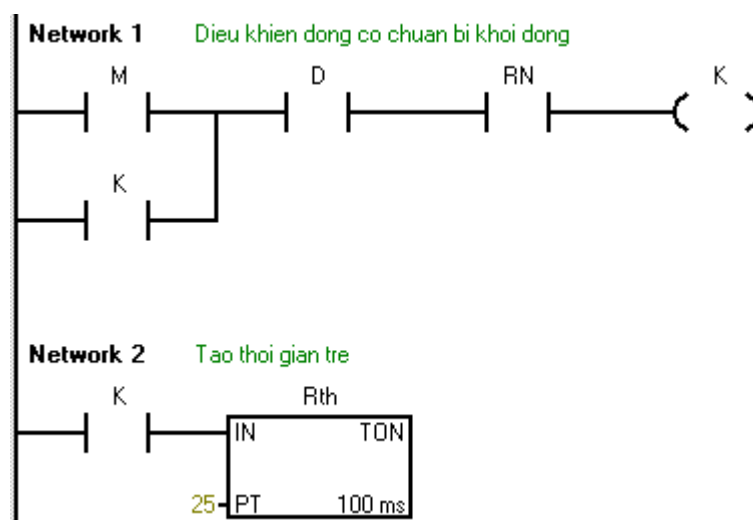
Tín hiệu đầu vào	Địa chỉ	Chức năng
M	I0.0	Nút ấn mở máy, thường mở, động cơ quay
D	I0.1	Nút dừng động cơ, thường đóng
RN	I0.2	Tiếp điểm thường đóng của rơle nhiệt để bảo vệ quá tải động cơ

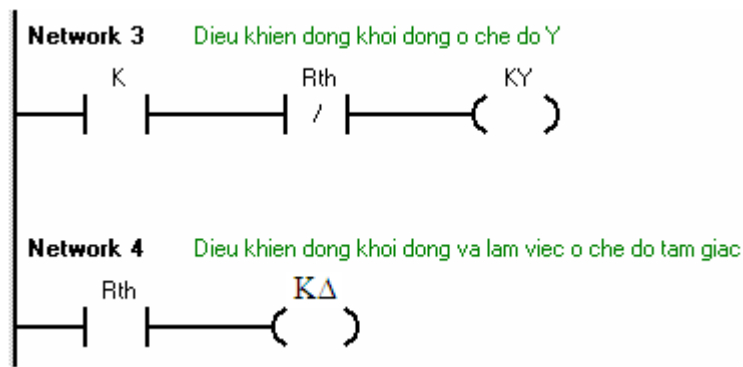
Địa chỉ đầu ra:

Tín hiệu đầu ra	Địa chỉ	Chức năng
K	Q0.0	Cuộn dây của công tắc tơ K
KY	Q0.1	Cuộn dây của công tắc tơ KY
KΔ	Q0.2	Cuộn dây của công tắc tơ KΔ

* Vẽ sơ đồ thiết kế mạch điều khiển

Trên cơ sở Quy trình làm việc và địa chỉ vào/ra ta tiến hành viết chương trình trên phần mềm Step 7 Micro/win như sau:





* Mô phỏng chạy thử:

***Bước 3: Kết nối cơ cấu chấp hành, nạp chương trình chạy cơ cấu chấp hành**

- Kết nối cơ cấu chấp hành: Để điều khiển đầu ra là cuộn dây công tắc tơ, ta chọn PLC loại AC/DC/RLY có: các tín hiệu vào là +24VDC ứng với mức logic 1 và 0VDC ứng với mức logic 0, công ra rơ le.

Bài tập thực hành

Lập trình PLC điều khiển 2 động cơ không đồng bộ 3 pha theo yêu cầu sau: Ấn nút M, động cơ 1 chạy ngay. Sau 5 giây, động cơ 2 khởi động đổi nối Y- Δ với thời gian khởi động là 3 giây. Ấn nút D hai động cơ cùng dừng.

4.3.5. Điều khiển động cơ thuận nghịch

Động cơ quay hai chiều được sử dụng rộng rãi trong các máy công nghiệp và dân dụng. Việc điều khiển các máy sản xuất đảo chiều quay hoặc chuyển động tịnh tiến theo hai chiều thực chất là điều khiển đảo chiều quay các động cơ truyền động cho các cơ cấu của máy, nên cần phải nắm vững cách điều khiển để có thể sử dụng trong thực tế điều khiển. Lập trình PLC giúp điều khiển đảo chiều quay động cơ không đồng bộ 3 pha dễ dàng và thuận tiện.

Các bước giải quyết bài toán:

***Bước 1: Phân tích quy trình làm việc**

- Xác định quy trình làm việc của phụ tải:

Mạch điều khiển động cơ không đồng bộ 3 pha quay hai chiều có thể mô tả quy trình hoạt động như sau:

Nếu ấn nút MT thì động cơ M quay thuận hoặc ấn nút MN động cơ M quay ngược. Ấn nút D động cơ M dừng. Để điều khiển động cơ M ta dùng công tắc tơ K và cấp điện và bảo vệ ngắn mạch ta dùng Aptomat TA.

Bước 2: Thiết kế mạch điều khiển bằng phân tử logic

* Khai báo địa chỉ đầu vào- đầu ra

- Địa chỉ đầu vào:

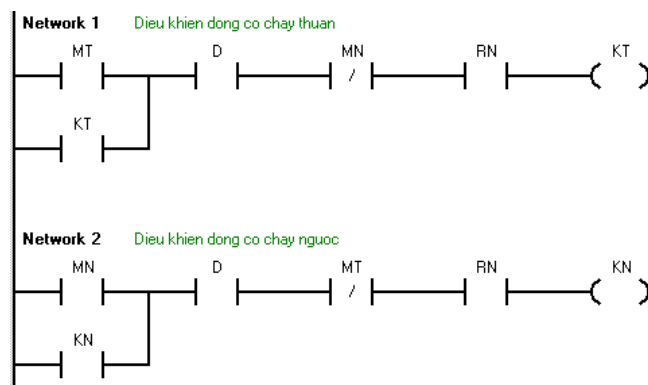
Tín hiệu đầu vào	Địa chỉ	Chức năng
MT	I0.0	Nút ấn mở máy, thường mở, động cơ quay thuận
MN	I0.1	Nút ấn mở máy, thường mở, động cơ quay ngược
D	I0.2	Nút dừng động cơ, thường đóng
RN	I0.3	Tiếp điểm thường đóng của rơle nhiệt để bảo vệ quá tải động cơ

- Địa chỉ đầu ra:

Tín hiệu đầu ra	Địa chỉ	Chức năng
KT	Q0.0	Cuộn dây của công tắc tơ KT
KN	Q0.1	Cuộn dây của công tắc tơ KN

* Vẽ sơ đồ thiết kế mạch điều khiển:

Trên cơ sở Quy trình làm việc và địa chỉ vào/ra ta tiến hành viết chương trình trên phần mềm Step 7 Micro/win như sau:



Sau khi viết chương trình chúng ta dùng chương trình mô phỏng SIMULINK S7 200 để kiểm tra các chức năng của mạch theo giản đồ thời gian đã có.

Bước 3: Kết nối cơ cấu chấp hành, nạp chương trình chạy cơ cấu chấp hành

* Kết nối cơ cấu chấp hành

Để điều khiển đầu ra là cuộn dây công tắc tơ K, ta chọn PLC loại AC/DC/RLY có: các tín hiệu vào là +24VDC ứng với mức logic 1 và 0VDC ứng với mức logic 0, công ra rơ le.

Tài liệu cần tham khảo:

- [1]. Đề cương môđun/môn học nghề Sửa chữa thiết bị điện tử công nghiệp”, Dự án Giáo dục kỹ thuật và Dạy nghề (VTEP), Tổng cục Dạy Nghề, Hà Nội, 2003
- [2]. Automatisieren mit sps - Guenter, Wellenreuther, Dieter Zastrow. nxb Viweg
- [3]. Steuerung von – ELWE
- [4]. Tự động hóa với simatic s7-200. Nguyễn Doãn Phước. nxb nông nghiệp
- [5]. Kỹ thuật điều khiển lập trình. Trung tâm Việt Đức Trường ĐHSPKT