

ỦY BAN NHÂN DÂN TỈNH BÀ RỊA – VŨNG TÀU

TRƯỜNG CAO ĐẲNG NGHỀ



GIÁO TRÌNH

MÔ ĐUN: LẬP TRÌNH PLC

NGHỀ: CƠ ĐIỆN TỬ

TRÌNH ĐỘ: CAO ĐẲNG NGHỀ VÀ TRUNG CẤP

Ban hành kèm theo Quyết định số 427A/QĐ-CDN ngày 01 tháng 09 năm 2015 của Hiệu trưởng trường Cao đẳng nghề tỉnh BR - VT

BR –VT , năm 2015

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Giáo trình **Lập Trình PLC** này được biên soạn theo chương trình chi tiết chuyên ngành Cơ Điện Tử, dùng cho hệ cao đẳng nghề và trung cấp. Tài liệu này là loại giáo trình nội bộ dùng trong nhà trường với mục đích làm tài liệu giảng dạy cho giáo viên và tài liệu học tập cho học sinh, sinh viên. Giáo trình trình bày những vấn đề cốt lõi nhất của mô đun **Lập Trình PLC**. Các bài học được trình bày ngắn gọn, dễ hiểu. Các kiến thức trong giáo trình được tham khảo từ rất nhiều nguồn khác nhau.

Chúng tôi mong rằng các sinh viên tự tìm hiểu trước mỗi vấn đề và kết hợp với bài giảng trên lớp của giáo viên để việc học môn này đạt hiệu quả.

Trong quá trình giảng dạy và biên soạn giáo trình này, chúng tôi đã nhận được sự động viên của quý thầy, cô trong Ban Giám Hiệu nhà trường cũng như những ý kiến của các đồng nghiệp trong khoa Điện . Chúng tôi xin chân thành cảm ơn và hy vọng rằng giáo trình này sẽ giúp cho việc dạy và học môđun **Lập Trình PLC** của trường chúng ta ngày càng tốt hơn.

Mặc dù đã rất nỗ lực, song không thể không có thiếu sót. Do đó chúng tôi rất mong nhận được những góp ý sửa đổi bổ sung thêm để giáo trình ngày càng hoàn thiện.

Bà Rịa – Vũng Tàu, ngày 31 tháng 08 năm 2015

Tham gia biên soạn

Đào Danh Tài

NỘI DUNG

	trang
Lời giới thiệu	3
Bài 1: Đại cương về điều khiển lập trình	11
1. Tổng quát về điều khiển lập trình.....	11
1.1. Điều khiển nối cứng và điều khiển lập trình.	11
1.1.1. Điều khiển kết nối cứng.....	11
1.1.2.Điều khiển logic khả trình (PLC).....	11
2. So sánh PLC với các thiết bị điều khiển thông thường khác.	12
2.1. Cấu trúc của một PLC.	16
3. Thiết bị điều khiển lập trình S7-200.....	17
3.1. Cấu trúc phần cứng.....	17
3.2. Mô tả đèn báo trạng thái trên S7 – 200, CPU 214 (224).....	17
3.3. Cổng truyền thông	18
3.4. Công tắc chọn chế độ của PLC	19
3.5. Vùng nhớ.....	19
3.6. Mở rộng ngõ vào/ ra.....	21
4. Xử lý chương trình	22
4.1. Vòng quét chương trình	22
4.2. Cấu trúc chương trình S7 – 200	23
4.3. Phương pháp lập trình	24
5. Kết nối PLC với các thiết bị ngoại vi.....	24
5.1. Cấp nguồn.....	24
5.2. Kết nối thiết bị ngoại vi.....	26
5.3. Kết nối CPU đến thiết bị lập trình.....	27
6. Kiểm tra việc nối dây bằng phần mềm.....	28

7. Cài đặt và sử dụng phần mềm STEP7-Micro/Win 32.....	28
7.1. Những yêu cầu đối với máy tính PC	28
7.2. Cài đặt và sử dụng phần mềm STEP 7 – Micro/Win 32	28
7.2.1. Cài đặt.....	28
Bài 2: Điều khiển on/off động cơ không đồng bộ ba pha	30
1.Giao diện của MicroWin 4.0	30
2.Soạn thảo	30
2.1.Khởi động chương trình	30
2.2. Soạn thảo chương trình	31
2.3. Kiểm tra lỗi	32
2.4. Lưu chương trình	32
2.5. Thiết lập thông số cho hộp thoại Communications:(truyền thông).....	33
2.6. Thiết lập sự kết nối với S7 – 200.....	34
2.7. Dowload chương trình	34
2.8. Chạy chương trình	34
2.9. Dừng chương trình	35
3. Lập trình mô phỏng trên máy tính.....	35
4. Các Liên Kết Logic	37
4.1. Lệnh vào/ ra và các lệnh tiếp điểm đặc biệt:	37
4.2. Load (LD)	37
4.3. Load Not (LDN)	37
4.4. OUTPUT	38
4.5. Lệnh tiếp điểm đặc biệt	38
4.6. Một số tiếp điểm trong vùng nhớ đặc biệt.....	39
5. Các lệnh liên kết logic cơ bản	39
5.1. Lệnh AND (A).....	39
5.2.Lệnh OR (O).....	40
6. Liên kết các công logic cơ bản	42
6.1. Liên kết AND trước OR	42

6.2. Liên kết OR trước AND	42
7. Lập trình mạch điện điều khiển động cơ không đồng bộ 3 pha quay 1 chiều.	42
7.1. Yêu cầu công nghệ	42
7.2. Nhiệm vụ	42
7.3. Sơ đồ mạch động lực	43
7.4.Lập bảng địa chỉ ngõ vào - ngõ ra	43
7.5.Sơ đồ kết nối PLC với thiết bị ngoại vi	44
7.6.Chương trình	44
7.7.Kết nối các thiết bị ngoại vi, download chương trình, chạy chương trình.	
7.7.1. Thiết lập thông số cho hộp thoại Communications:(truyền thông).....	44
7.7.2. Thiết lập sự kết nối với S7 – 200	45
7.7.3. Dowload chương trình	46
7.7.4. Chạy chương trình	47
7.7.5. Dừng chương trình	47
8. Câu hỏi ôn tập.....	47
Bài 3: Điều khiển động cơ 1 chiều	48
1. Các lệnh ghi / xóa (set/ reset) giá trị cho tiếp điểm.....	48
1.1. Mạch nhớ R-S	48
1.2.lệnh Set (S)	48
1.3.lệnh ReSet (R)	49
2. Điều khiển động cơ không đồng bộ 3 pha quay 2 chiều.	49
2.1. Yêu cầu công nghệ	49
2.2. Nhiệm vụ	49
2.2.1. Sơ đồ mạch động lực	50
2.2.2. Lập bảng địa chỉ ngõ vào - ngõ ra	50
2.2.3.Sơ đồ kết nối PLC với thiết bị ngoại vi	51
2.2.4.Chương trình	51
3.Bài tập	51
Bài 4: Điều khiển hệ thống động cơ khởi động tuần tự	55

1. Timer (Bộ định thời).....	55
2. Điều khiển 4 động cơ không đồng bộ 3 pha khởi động tuần tự	57
2.1. Yêu cầu công nghệ	57
2.2. Nhiệm vụ	57
2.2.1. Mạch động lực	57
2.2.2. Lập bảng địa chỉ ngõ vào - ngõ ra	58
2.2.3. Sơ đồ kết nối PLC với thiết bị ngoại vi	58
2.2.4. Chương trình	59
3. Bài tập ứng dụng Timer	59
Bài 5: Điều khiển dây chuyền đóng gói sản phẩm	61
1. Counter (Bộ đếm)	61
1.1. Bộ đếm tiến (CTU):	61
1.2. Bộ đếm xuống (CTD)	62
2. Điều khiển dây chuyền đóng gói sản phẩm	63
2.1. Yêu cầu công nghệ	63
2.2. Lập bảng địa chỉ ngõ vào - ngõ ra	64
3. Bài tập	66
Bài 6: Điều khiển trạm khí nén có hai xy lanh	67
1. Chức năng truyền dẫn	67
1.1. Truyền dẫn Byte; Word; Doubleword	67
1.2. Truyền một vùng nhớ dữ liệu	68
2. Chức năng dịch chuyển	68
2. Dịch phải Byte SHR_B và Dịch trái Byte SHL_B	68
2.1. Dịch phải Byte SHR_B và Dịch trái Byte SHL_B	68
2.2. Dịch phải Word SHR_W và Dịch trái Word SHL_W	69
2.3. Dịch phải Doubleword SHR_DW và Dịch trái SHL_DW	69
3. Chức năng so sánh	70
4. Yêu cầu	71

5. Sơ đồ kết nối plc.....	71
6. Các bước thực hiện:	72
7. Bài tập áp dụng.....	72
Bài 7: Điều khiển hệ thống động cơ dùng hàm chương trình con	75
1. Lệnh nhảy và gọi chương trình con	75
2.Ví dụ tạo chương trình con.....	76
3. Bài tập.....	78
3.1. Yêu cầu:.....	78
3.2. Sơ đồ công nghệ.....	78
3.2.1. Mô tả.....	79
3.2.2. Nhiệm vụ	79
3.3. Mạch động lực.....	79
3.4. Lập bảng địa chỉ ngõ vào - ngõ ra	80
3.5.Nối dây PLC	80
3.6. Chương trình	81
Bài 8: Điều khiển tốc độ động cơ dùng hàm analog	82
1.Tín hiệu Analog	82
2.Biểu diễn các giá trị Analog	83
3. <i>Kết nối ngõ vào-ra Analog</i>	83
3.1. <i>Phương pháp định tỷ lệ ngõ vào Analog (Input calibration).....</i>	84
3.1.1.Để thực hiện việc định tỷ lệ cần theo các bước sau	84
3.1.2. <i>Hiệu chỉnh tín hiệu Analog.....</i>	85
4.5. <i>Giới thiệu module Analog S7-200.....</i>	87
4.5.1. Đọc tín hiệu analog từ Modul EM231	87
4.5.2. Xuất tín hiệu analog qua modul EM232	89
4.5.3. Modul EM235	89

5. Điều khiển mô hình hệ thống điều khiển nhiệt độ	90
5.1 Điều khiển nhiệt độ của lò.....	90
5.2. Chương trình.....	90
5.2.1: chương trình con.....	90
5.2.2. chương trình chính	92
Tài liệu tham khảo	93

CHƯƠNG TRÌNH MÔ ĐUN LẬP TRÌNH PLC

Mã số mô đun: MĐ 09

VỊ TRÍ, TÍNH CHẤT CỦA MÔ ĐUN:

- Mô đun này bao gồm các bài tập riêng biệt để điều khiển một phần hoặc toàn bộ hệ thống.
- Mô đun này cần được thực hiện theo định hướng thực hành. Qua đó học viên được đào tạo các kỹ năng tự lập kế hoạch, tự thực hiện và tự kiểm tra.
- Mỗi bài tập đều có sự kết hợp giữa lý thuyết và thực hành.
- Xuất phát từ yêu cầu công nghệ, học viên phải phân tích được quá trình, vẽ sơ đồ mạch, viết chương trình, nạp chương trình vào PLC, lắp ráp hệ thống, kiểm tra hoạt động, vận hành hệ thống và thực hiện các công việc tìm và sửa lỗi.
- Để học được mô đun này, người học phải có các kiến thức cơ bản về kỹ thuật cơ khí, đặc biệt là kỹ thuật tháo lắp, lắp đặt điện và điều khiển khí nén.

MỤC TIÊU MÔ ĐUN:

-Kiến thức chuyên môn:

- + Các lệnh logic.
- + Lệnh P, N, SET, RESET.
- + Hàm Timer, hàm Counter.
- + Lệnh so sánh, các lệnh xử lý thanh ghi.
- + Chương trình con, thời gian thực, tín hiệu Analog.

Kỹ năng nghề:

- + Lập trình cho plc dùng phần mềm Step7 Microwin.
- + Thiết kế các chương trình điều khiển dùng PLC ở mức độ đơn giản và trung bình.

Thái độ lao động:

+ Cẩn thận, chính xác trong công việc.

Các kỹ năng cần thiết khác:

+ Phối hợp tốt trong làm việc nhóm.

NỘI DUNG TỔNG QUÁT VÀ PHÂN BỐ THỜI GIAN

TT	Tên các bài trong mô đun	Thời gian	Hình thức giảng dạy
1	Đại cương về điều khiển lập trình	10	Lý thuyết
2	Điều khiển on/off động cơ không đồng bộ ba pha	12	Tích hợp
3	Điều khiển động cơ 1 chiều	10	Tích hợp
4	Điều khiển hệ thống động cơ khởi động tuần tự	15	Tích hợp
	Kiểm tra bài 4	1	
5	Điều khiển dây chuyền đóng gói sản phẩm	14	Tích hợp
	Kiểm tra bài 5	2	
6	Điều khiển hệ thống động cơ làm việc luân phiên	14	Tích hợp
	Kiểm tra bài 6	2	
7	Điều khiển hệ thống động cơ dùng hàm chương trình con.	16	Tích hợp
	Kiểm tra bài 7	2	
8	Điều khiển tốc độ động cơ dùng hàm analog	20	Tích hợp
	Kiểm tra bài 8	2	
	TỔNG SỐ	120	

BÀI 1

ĐẠI CƯƠNG VỀ ĐIỀU KHIỂN LẬP TRÌNH

Mục tiêu :

- Trình bày được các ưu điểm của điều khiển lập trình so với các loại điều khiển khác và các ứng dụng của chúng trong thực tế.
- Trình bày được cấu trúc và nhiệm vụ các khối chức năng của PLC.
- Thực hiện được sự kết nối giữa PLC và các thiết bị ngoại vi.
- Lắp đặt các thiết bị bảo vệ cho PLC theo yêu cầu kỹ thuật.

Nội dung chính :

1. Tổng quát về điều khiển lập trình.

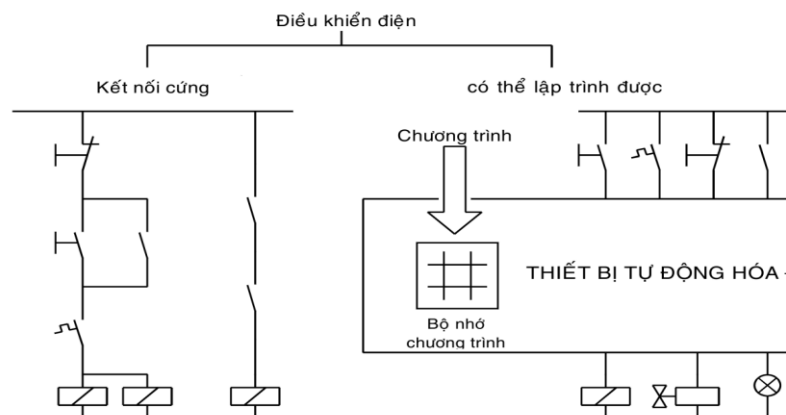
1.1. Điều khiển nối cứng và điều khiển lập trình.

1.1.1. Điều khiển kết nối cứng

Điều khiển kết nối cứng là loại điều khiển mà các chức năng của nó được đặt cố định (nối dây). Nếu muốn thay đổi chức năng điều đó có nghĩa là thay đổi kết nối dây. Điều khiển kết nối cứng có thể thực hiện với các tiếp điểm (Relais, khởi động từ, v.v.) hay điện tử (mạch điện tử).

1.1.2. Điều khiển logic khả trình (PLC)

Điều khiển logic khả trình là loại điều khiển mà chức năng của nó được đặt cố định thông qua một chương trình còn gọi là bộ nhớ chương trình. Các phần tử nhập tín hiệu được nối ở ngõ vào của bộ điều khiển, các phần tử này khởi động các cuộn dây đặt ở ngõ ra. Quá trình điều khiển ở đây được thực hiện bằng một chương trình đã soạn thảo theo mục đích, yêu cầu của việc điều khiển thiết bị. Nếu chức năng điều khiển cần được thay đổi, thì chỉ phải thay đổi chương trình bằng thiết bị lập trình cho đối tượng điều khiển tương ứng hay cắm một bộ nhớ chương trình đã lập trình khác vào trong bộ điều khiển.



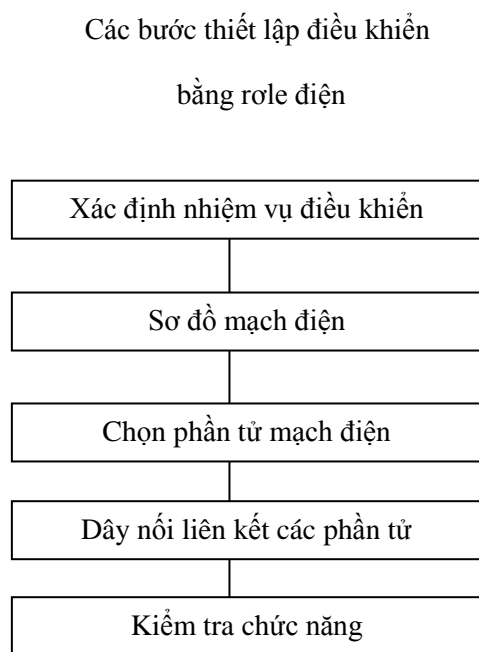
Hình 1-1: Điều khiển kết nối cứng và điều khiển logic

2. So sánh PLC với các thiết bị điều khiển thông thường khác.

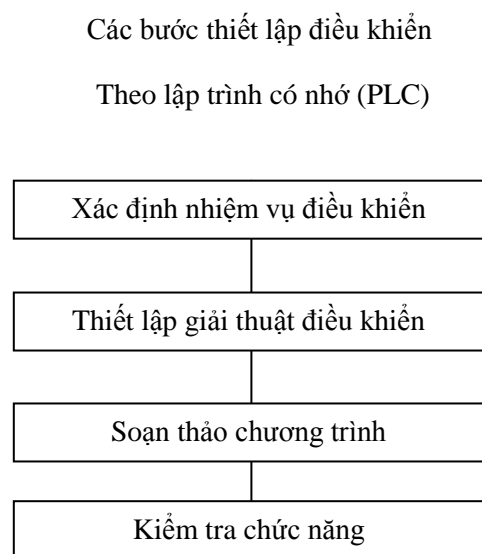
Trong công nghiệp, yêu cầu tự động hóa ngày càng tăng, đòi hỏi kỹ thuật điều khiển phải đáp ứng được các yêu cầu đó. Trong những năm gần đây, bên cạnh việc điều khiển bằng relay và khởi động từ thì việc điều khiển có thể lập trình được càng phát triển với hệ thống đóng mạch điện tử và thực hiện lập trình bằng máy tính.

Trong nhiều lĩnh vực, các loại điều khiển cũ đã được thay đổi bởi điều khiển có thể lập trình được, có thể gọi là điều khiển logic khả trình. Viết tắt trong tiếng Anh là PLC(Programmable Logic Control), tiếng Đức là SPS (Speicherprogrammierbare Steuerung). Sự khác biệt cơ bản giữa điều khiển logic lập 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 là 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:



Hình 1-2: Điều khiển bằng role



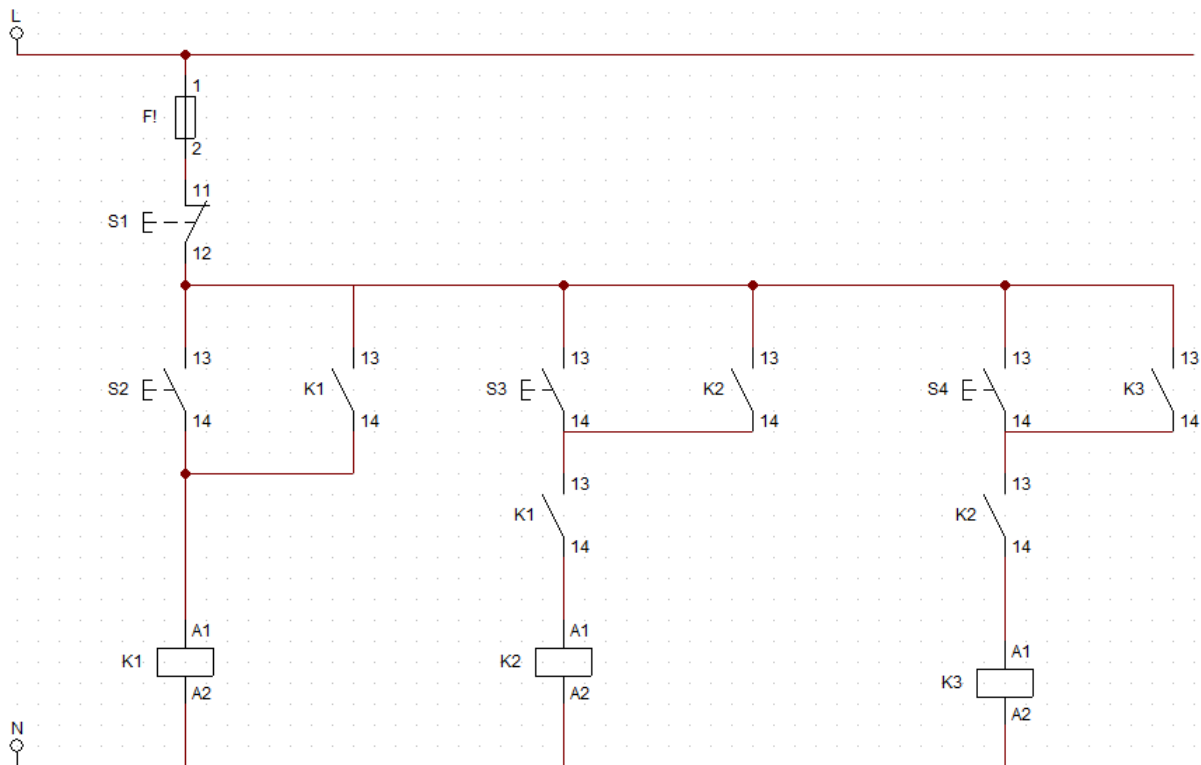
Hình 1-3: Điều khiển bằng PLC

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.

** Sự khác nhau giữa hệ điều khiển bằng rơ le điện và hệ điều khiển logic khả trình có thể minh họa 1 cách cụ thể như sau:*

Điều khiển hệ thống của 3 máy bơm qua 3 khởi động từ K1, K2, K3. Trình tự điều khiển 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 1-4: Mạch điều khiển tuần 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 tuần 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.

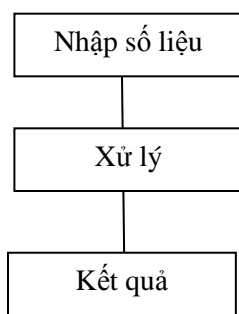
Các tiếp điểm K1, K2, K3 và các mối nối liên kết là các phần tử xử lý. Các khởi động từ

K1, K2, K3 là kết quả xử lý.

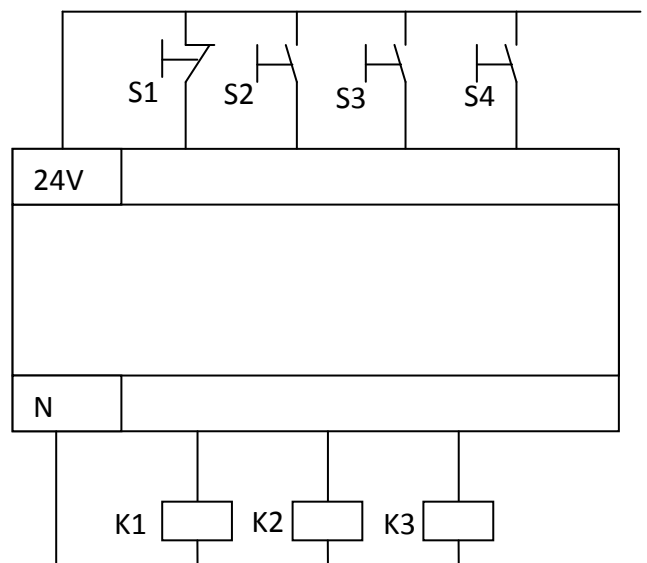
Nếu thay đổi mạch điện điều khiển ở phần xử lý bằng hệ PLC ta có thể biểu diễn hệ thống như sau:

- 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, tuần tự đóng mở theo yêu cầu đề ra sẽ được lập trình, chương trình sẽ được nạp vào bộ nhớ.

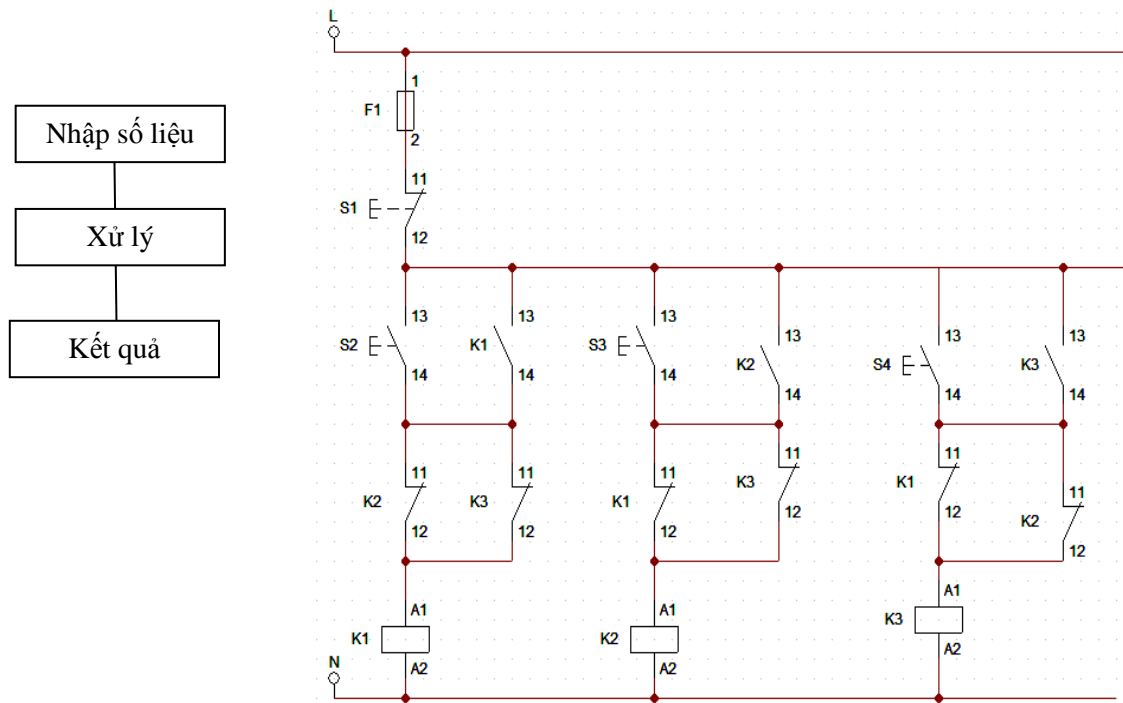


Hình 1-5: Lưu đồ xử lý bằng PLC



Hình 1- 6: Sơ đồ kết nối cứng PLC

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 rơ le 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 1-7



Hình 1-7: Sơ đồ mạch điều khiển 3 động cơ đã được thay đổi.

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 có nhớ 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

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

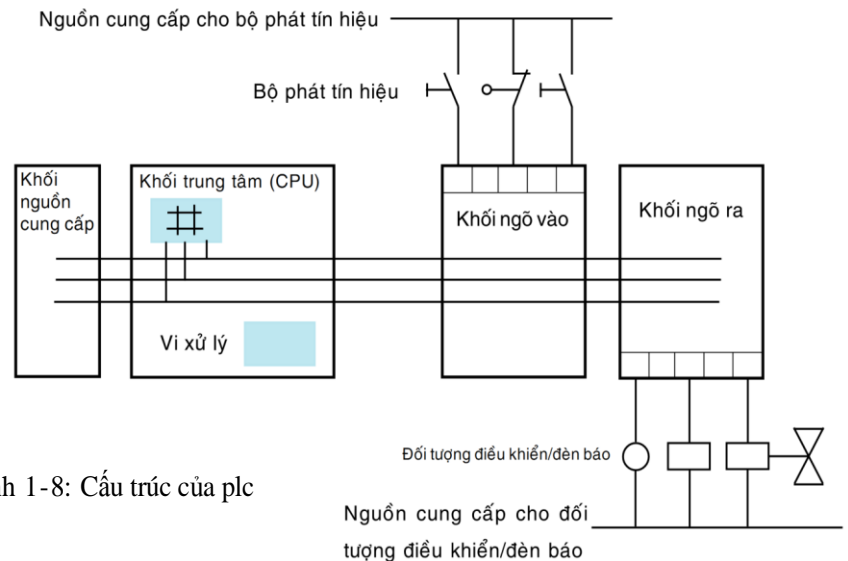
- Thích ứng với những nhiệm vụ điều khiển khác nhau.
- Khả năng thay đổi đơn giản trong quá trình đưa thiết bị vào sử dụng.
- Nhu cầu mặt bằng ít.
- Tiết kiệm thời gian trong quá trình mở rộng và phát triển nhiệm vụ điều khiển bằng cách copy các chương trình.
- Các thiết bị điều khiển chuẩn.
- Không cần các tiếp điểm.

Hệ thống điều khiển theo lập trình có nhớ 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.

2.1. Cấu trúc của một PLC.



Hình 1-8: Cấu trúc của plc

Khối nguồn nuôi: nguồn trong các PLC thường là 24VDC.

Module CPU: (cũng có bộ PLC sử dụng nguồn 220VAC. Những PLC không có module nguồn thì được cấp nguồn bên ngoài CPU: central processing unit: đơn vị xử lý trung tâm) bao gồm: bộ vi xử lý và bộ nhớ.

Module xuất nhập (I/O module).

+ **Module nhập (input module)** được nối với các công tắc, nút ấn, các bộ sensor ... để điều khiển từ chương trình bên ngoài.

+ **Module xuất (output module)** được nối với các tải ở ngõ ra như cuộn dây của relay, contactor, đèn tín hiệu, các bộ ghép quang ...

Hệ thống bus truyền tín hiệu: hệ thống bus truyền tín hiệu gồm nhiều đường tín hiệu song song:

- **Tuyến địa chỉ (address bus):** chọn địa chỉ trên các khối khác nhau.
- **Tuyến dữ liệu (data bus):** mang dữ liệu từ khối này đến khối khác.
- **Tuyến điều khiển (control bus):** chuyển, truyền các tín hiệu định thì và điều khiển để đồng bộ các hoạt động trong PLC .

Chương trình điều khiển được nạp vào bộ nhớ nhờ bộ lập trình cầm tay (programming console) hay bằng một máy tính. Hiện nay đã có một số loại PLC được thiết kế có các phím bấm để có thể lập trình trực tiếp mà không cần bộ lập trình cầm tay hay máy vi tính.

3. Thiết bị điều khiển lập trình S7-200.

3.1. Cấu trúc phần cứng.

S7-200 là thiết bị điều khiển logic khả trình loại nhỏ của Hãng SIEMENS (CHLB Đức) có cấu trúc theo kiểu Modul và các modul mở rộng. Các modul này được sử dụng cho nhiều ứng dụng lập trình khác nhau. Thành phần cơ bản của S7-200 là khối vi xử lý CPU 212, CPU 214, CPU 215, CPU 216, CPU 221, CPU 222, CPU 224, CPU 224XP, ... Về hình thức bên ngoài, sự khác nhau của các loại CPU này nhận biết nhờ số đầu vào/ra và nguồn cung cấp.

Ví dụ:

_ CPU 212 có 8 cổng vào và 6 cổng ra và có khả năng được mở rộng thêm bằng 2 modul mở rộng.

_ CPU-214(224) bao gồm 14 ngõ vào và 10 ngõ ra, có khả năng thêm 7 modul mở rộng.

+ Tổng số ngõ vào / ra cực đại là 64 ngõ vào và 64 ngõ ra.

+ 128 Timer chia làm 3 loại theo độ phân giải khác nhau: 4 Timer 1ms, 16 Timer 10ms và 108 Timer 100ms.

+ 128 bộ đếm chia làm 2 loại: chỉ đếm tiến và vừa đếm tiến vừa đếm lùi.

+ 86 byte nhớ đặc biệt (SM) dùng để thông báo trạng thái và đặt chế độ làm việc.

+ 4696 byte nhớ đa dụng(V).

+ Toàn bộ vùng nhớ không bị mất dữ liệu trong khoảng thời gian 190 giờ kể từ khi PLC bị mất nguồn cung cấp.

3.2. Mô tả đèn báo trạng thái trên S7 – 200, CPU 214 (224):

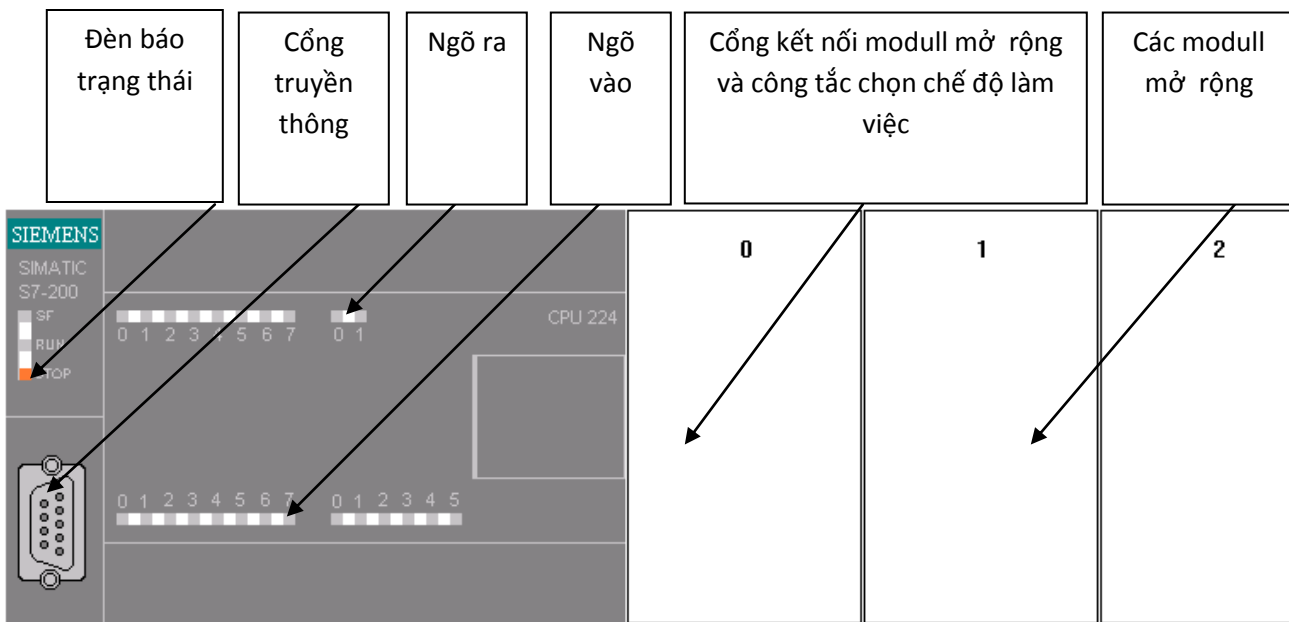
- SF (Đèn đỏ): Đèn báo hiệu hệ thống bị hỏng. Đèn SF sáng lên khi PLC bị hỏng hóc.

- RUN(Đèn xanh): Đèn chỉ định PLC đang ở chế độ làm việc và thực hiện chương trình được nạp trong máy.

- STOP(Đèn vàng): Đèn chỉ định PLC đang ở chế độ dừng. Dừng chương trình đang thực hiện lại.

- Ix.x(Đèn xanh): đèn ở cổng vào chỉ định trạng thái tức thời của cổng Ix.x(x.x = 0.0 ÷ 1.5). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị của cổng ngõ vào.

- Qy.y(Đèn xanh): đèn ở cổng ra chỉ định trạng thái tức thời của cổng Qy.y(y.y = 0.0 ÷1.1). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị của cổng ngõ ra.

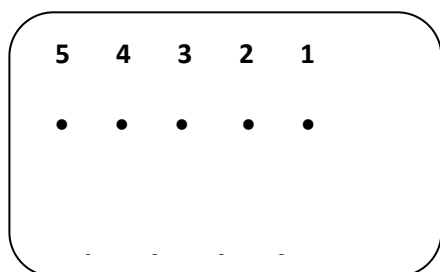


Hình 1-9: Các cổng vào ra của PLC S7-200

3.3. 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 các thiết bị lập trình hoặc với các trạm PLC khác.

Để ghép S7 – 200 với máy tính PC qua cổng RS232 cần có cáp nối PC/PPI với bộ chuyển đổi từ RS232 sang RS485. Tốc độ truyền cho máy lập trình kiểu PPI là 9600 baud. Tốc độ truyền cung cấp cho của PLC theo kiểu tự do là từ 300 đến 38400.



Sơ đồ chân của cổng truyền thông

Chân	Giải thích
1	Đất
2	24 VDC
3	Truyền và nhận dữ liệu
4	Không sử dụng
5	Đất
6	5 VDC (điện trở trong 100Ω)
7	24 VDC (120mA tối đa)

- 8 Truyền và nhận dữ liệu
- 9 Không sử dụng.

Để ghép nối S7 – 200 với máy lập trình PG702 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.

3.4. Công tắc chọn chế độ của PLC:

Công tắc chọn chế độ làm việc nằm phía trên, bên cạnh cổng kết nối modul mở rộng, có ba vị trí cho phép chọn các chế độ làm việc khác nhau cho PLC.

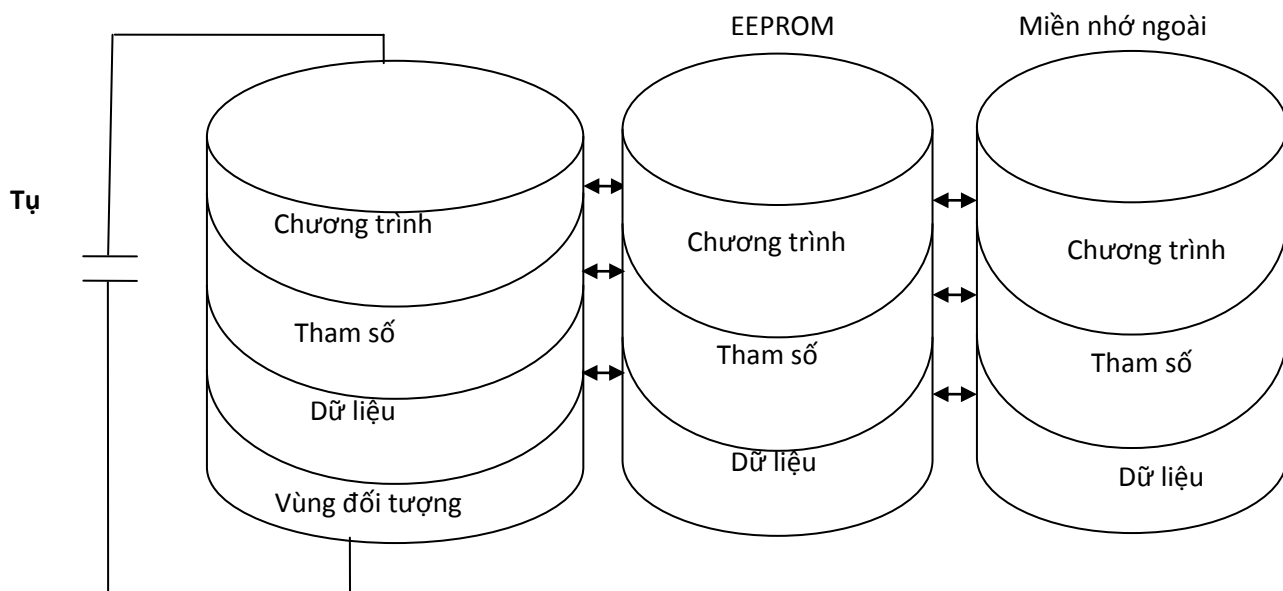
RUN: cho phép PLC thực hiện chương trình trong bộ nhớ. PLC S7 – 200 sẽ rời khỏi chế độ RUN và chuyển sang chế độ STOP nếu trong máy có sự cố hoặc trong chương trình gặp lệnh STOP, thậm chí ngay cả khi công tắc ở chế độ RUN. Nên quan sát trạng thái thực tại của PLC theo đèn báo.

STOP: Cường bức PLC dừng thực hiện chương trình đang chạy và chuyển sang chế độ STOP. Ở chế độ STOP PLC cho phép hiệu chỉnh lại chương trình hoặc nạp một chương trình mới.

TERM: Cho phép máy lập trình tự quyết định một trong các chế độ làm việc cho PLC (hoặc ở chế độ RUN hoặc ở chế độ STOP)

3.5. Vùng nhớ:

Bộ nhớ của S7 – 200 được chia thành các vùng nhớ như hình vẽ:



Hình 1-10: Bộ nhớ trong v ngoài của S7-200

Trong PLC có một tụ điện có nhiệm vụ duy trì dữ liệu trong khoảng thời gian nhất định khi bị mất nguồn. Bộ nhớ S7-200 có tính năng động cao, đọc và ghi trong toàn vùng, trừ các bit nhớ đặc biệt SM(special memory) chỉ có thể truy cập để đọc.

Vùng chương trình: vùng nhớ này sử dụng để lưu các lệnh của chương trình, nó thuộc kiểu đọc/ghi(non/volatile)

Vùng tham số: là vùng nhớ để lưu trữ các tham số như: từ khoá, địa chỉ trạm, vùng tham số này cũng thuộc kiểu đọc/ghi.

Vùng dữ liệu: được sử dụng để cất giữ các dữ liệu của chương trình bao gồm các kết quả của các phép tính, hằng số được định nghĩa trong chương trình, bộ đếm truyền thông... vùng nhớ này có một phần thuộc kiểu đọc/ghi được.

Vùng dữ liệu được chia ra thành những miền nhớ nhỏ với các công dụng khác nhau:

- + *I* - Input image registet: Vùng đệm cổng vào
- + *V* - Variable memory: Vùng nhớ biến
- + *Q* - Output image registet: Vùng đệm cổng ra
- + *M* - Internal memory: Vùng nhớ nội
- + *SM* - Special memory: Vùng nhớ đặc biệt

Vùng đối tượng: Bao gồm các times, counter, high speed counter, các cổng vào ra tương tự được đặt trong vùng nhớ cuối cùng, tham số này cũng thuộc kiểu đọc/ghi.

- + *T* - Times: Điều khiển thời gian
- + *C* – Counter: Bộ đếm
- + *HSC* - High Speed Counter: Bộ đếm tốc độ cao
- + *AIW* - Analog Input: Cổng vào tương tự
- + *AQW* - Analog Output: Cổng ra tương tự
- + *AC* – Accumulator: Thanh ghi

Địa chỉ truy nhập được qui ước với công thức:

Truy nhập theo bit: Tên miền (+) địa chỉ byte (+).(+) chỉ số bit

Ví dụ: V150.4: Chỉ bit 4 của byte 150 thuộc miền V

Truy nhập theo byte: Tên miền (+) B (+) địa chỉ số byte trong miền.

Ví dụ: VB150: Chỉ của byte 150 thuộc miền V

Truy nhập theo từ (Word): Tên miền (+) W (+) địa chỉ byte cao của từ trong miền.

Ví dụ: VW150: Chỉ từ đơn gồm hai byte 150 và 151 thuộc miền V. Trong đó byte 150 có vai trò là byte cao trong từ

- Truy nhập theo từ kép (Double Word): Tên miền (+) D (+) địa chỉ byte cao của từ trong miền.

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

VW150	VB150 (byte cao)	VB151 (byte thấp)
-------	--------------------	---------------------

Ví dụ: VD150: Chỉ từ kép gồm bốn 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

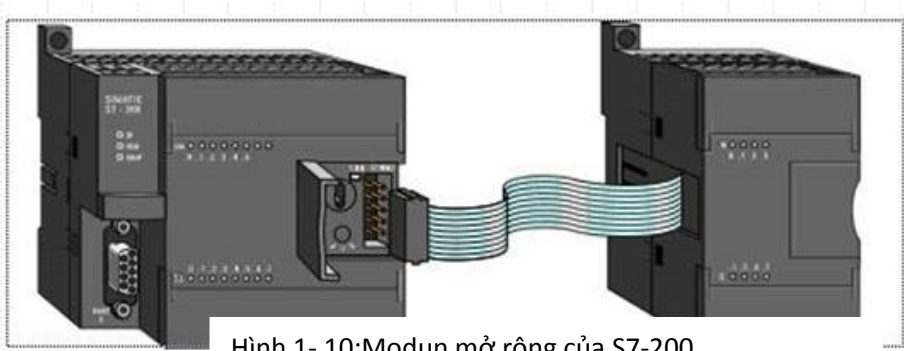
Bit 63 32 31 16 15 8 7 0

VD150	VB150(byte cao)	VB151	VB152	VB153(byte thấp)
-------	-----------------	-------	-------	------------------

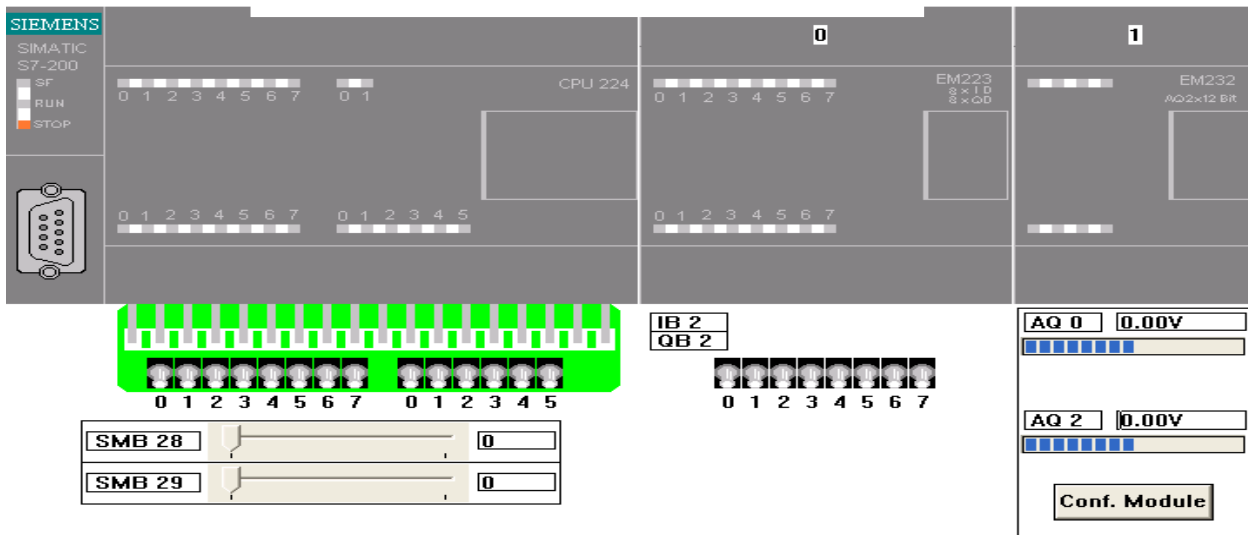
3.6. Mở rộng ngõ vào/ ra:

Có thể mở rộng ngõ vào/ ra của PLC bằng cách ghép nối thêm vào PLC các modull mở rộng về phía bên phải của CPU (CPU 224 có thể ghép nhiều nhất 7 modull mở rộng), làm thành một móc xích, bao gồm các modull có cùng kiểu.

Các modull mở rộng số hay tương tự đều chiếm chỗ trong bộ đệm, tương ứng với số ngõ vào/ ra của các modull.



Hình 1- 10:Modun mở rộng của S7-200



Hình 1- 11:Modun mở rộng của S7-200

Sau đây là một ví dụ về cách đặt địa chỉ cho các modul mở rộng trên CPU 214 (CPU224):

CPU 214	Modul 0	Modul 1	Modul 2	Modul 3	Modul 4
	(4 vào/4 ra)	8 vào	3 vào analog/ 1 ra analog	8 ra	3 vào analog/ 1 ra analog

I0.0	Q0.0	I2.0	I3.0	AIW0	Q3.0	AIW8
I0.1	Q0.1	I2.1	I3.1	AIW2	Q3.1	AIW10
I0.2	Q0.2	I2.2	I3.2	AIW4	Q3.2	AIW12
I0.3	Q0.3	I2.3	I3.3		Q3.3	
I0.4	Q0.4		I3.4	AQW0	Q3.4	AQW4
I0.5	Q0.5	Q2.0	I3.5		Q3.5	
I0.6	Q0.6	Q2.1	I3.6		Q3.6	
I0.7	Q0.7	Q2.2	I3.7		Q3.7	
I1.0	Q1.0	Q2.3				
I1.1	Q1.1					
I1.2						
I1.3						
I1.4						
I1.5						

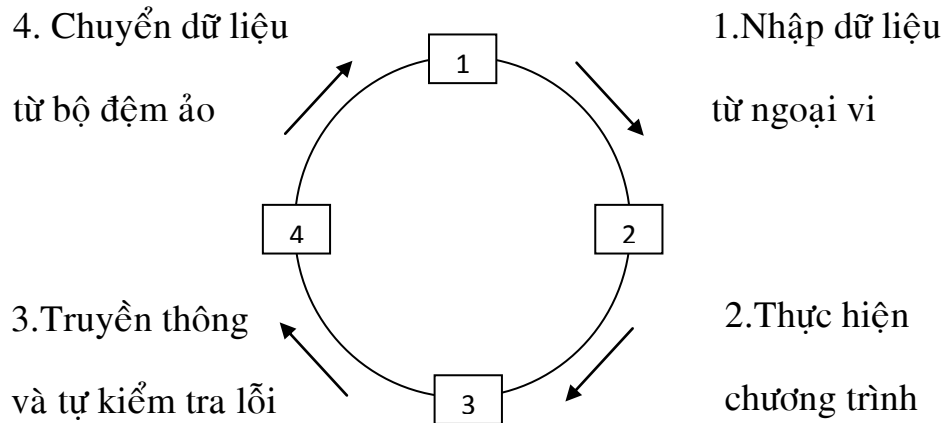
4. Xử lý chương trình

4.1. Vòng quét chương trình

PLC thực hiện chương trình theo chu kỳ lặp. Mỗi vòng lặp được gọi là vòng quét (scan). Như hình (H.12)

Mỗi vòng quét bắt đầu bằng giai đoạn đọc dữ liệu từ các ngõ vào (contact, sensor, relay...) vào vùng bộ đệm ảo, tiếp theo là giai đoạn thực hiện chương trình. Trong từng vòng quét, chương trình được thực hiện bằng lệnh đầu tiên và kết thúc tại lệnh END (MEND). Sau giai đoạn thực hiện chương trình là giai đoạn truyền thông nội bộ và kiểm

tra lỗi. Vòng quét được kết thúc bằng giai đoạn chuyển các nội dung của bộ đệm ảo tới các ngõ ra. Như vậy, tại thời điểm thực hiện lệnh vào/ ra, lệnh này không trực tiếp làm việc với cổng vào/ ra mà chỉ thông qua bộ đệm ảo của cổng trong vùng tham số. Việc truyền thông giữa bộ đệm ảo với thiết bị ngoại vi trong giai đoạn 1 và 4 là do CPU quản lý. Khi gặp lệnh vào/ ra ngay lập tức thì hệ thống sẽ cho dừng mọi công việc khác, ngay cả chương trình xử lý ngắt để thực hiện lệnh này trực tiếp với cổng vào/ ra.



Hình 1-12: Một vòng quét của PLC.

4.2. Cấu trúc chương trình S7 – 200:

Có thể lập trình cho PLC S7 – 200 bằng cách sử dụng một trong những phần mềm sau:

- _ STEP 7- Micro/ DOS.
- _ STEP 7- Micro/ WIN.

Những phần mềm này đều có thể cài đặt được trên các máy lập trình họ PG7xx và các máy tính cá nhân (PC).

Các chương trình của S7 – 200 phải có cấu trúc bao gồm:

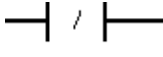



- _ Chương trình chính (main program)
- _ Chương trình con: là bộ phận của chương trình. Các chương trình con phải được viết sau lệnh kết thúc chương trình chính
- _ Các chương trình xử lý ngắt là một bộ phận của chương trình. Nếu cần sử dụng chương trình xử lý ngắt phải viết sau lệnh kết thúc chương trình chính.


Các chương trình con được nhóm lại thành một nhóm ngay sau chương trình chính. Sau đó đến ngay các chương trình xử lý ngắt. Bằng cách viết như vậy, cấu trúc chương trình được rõ ràng và thuận tiện hơn trong việc đọc chương trình sau này. Có thể tự do trộn lẫn các chương trình con và chương trình xử lý ngắt đăng sau chương trình chính.


4.3. Phương pháp lập trình:

Cách lập trình cho S7-200 nói riêng và cho các PLC nói chung dựa trên các phương pháp cơ bản. Phương pháp hình thang (Ladder, viết tắt là LAD), phương pháp liệt kê lệnh (Statement list, viết tắt là STL), và phương pháp FBD (Function Block Diagram).

Phương pháp hình thang (LAD): LAD là một ngôn ngữ lập trình bằng đồ họa, những thành phần cơ bản dùng trong LAD tương ứng với các thành phần của bảng điều khiển bằng relay. Trong chương trình LAD, các phần tử cơ bản dùng để biểu diễn lệnh logic như sau:

+ Tiếp điểm: Là biểu tượng (Symbol) mô tả các tiếp điểm của relay. các tiếp điểm có thể là thường đóng  /  hoặc thường mở  .

+ Cuộn dây (coil): Là biểu tượng  mô tả relay được mắc theo chiều dòng điện cung cấp cho relay.

+ Hộp (Box):  Là biểu tượng mô tả các hàm khác nhau, nó làm việc khi có dòng điện chạy đến hộp. Những dạng hàm thường được biểu diễn bằng hộp là các bộ thời gian (Timer), bộ đếm (counter) và các hàm toán học. Cuộn dây và các hộp phải mắc đúng chiều dòng điện.

+ Mạng LAD: Là đường nối các phần tử thành một mạch hoàn thiện, đi từ đường nguồn bên trái sang đường nguồn bên phải. Đường nguồn bên trái là dây nóng, đường nguồn bên phải là dây trung hoà hay là đường dây trở về nguồn cung cấp.

Phương pháp liệt kê lệnh (STL): Là phương pháp thể hiện chương trình dưới dạng tập hợp các câu lệnh. Mỗi câu lệnh trong chương trình, kể cả những lệnh hình thức biểu diễn một chức năng của PLC.

Phương pháp FBD (Function Block Diagram):

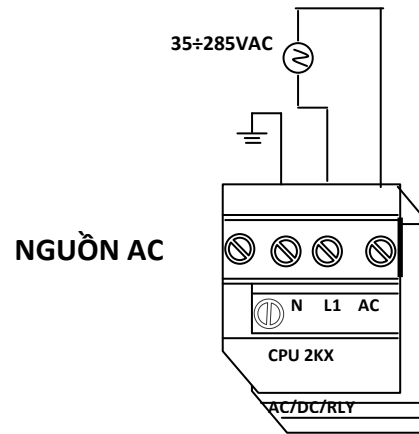
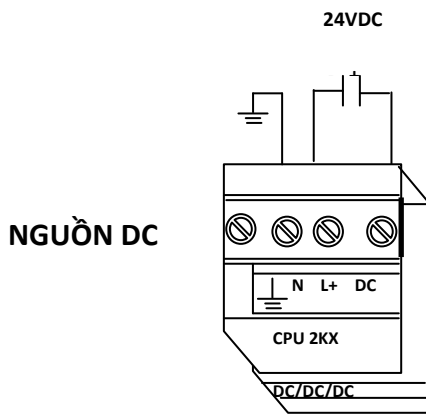
FBD là ngôn ngữ lập trình bằng các cổng logic. Trong chương trình FBD các phần tử cơ bản dùng để biểu diễn lệnh là:

- + Các tiếp điểm mắc nối tiếp với nhau bằng cổng AND
- + Các tiếp điểm hở ghép song song được thay bằng cổng OR
- + Các tiếp điểm thường đóng thì có cổng NOT
- + Hai tiếp điểm đối ngược nhau ghép nối tiếp dùng cổng XOR
- + Các tiếp điểm thường đóng ghép song song được thay bằng cổng NAND

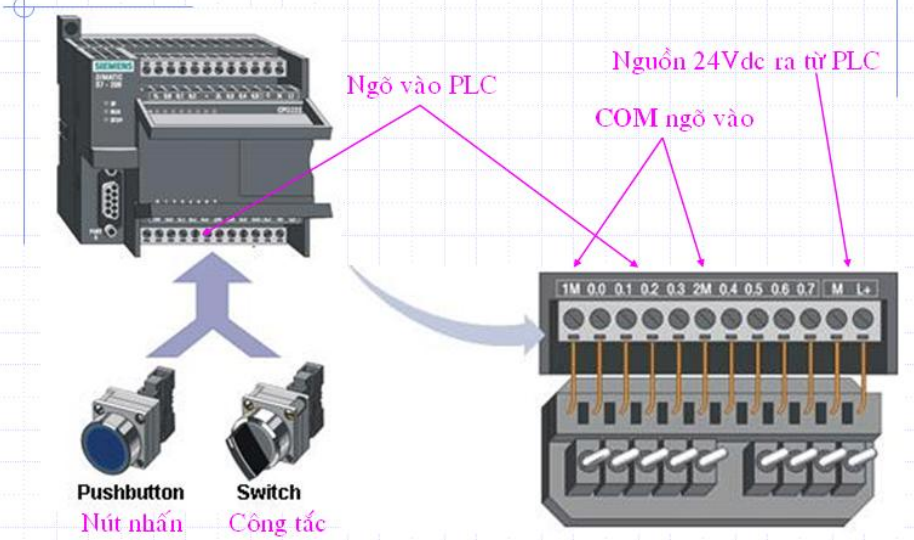
5. Kết nối PLC với các thiết bị ngoại vi:

5.1. Cấp nguồn:

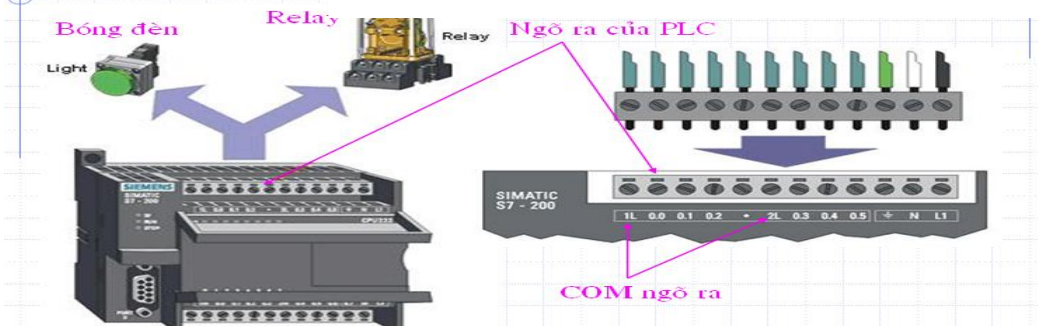
Tùy theo CPU sử dụng loại nguồn nào mà ta kết nối nguồn cho phù hợp



Kết nối tín hiệu ngõ vào cho PLC



Kết nối tín hiệu ngõ ra của PLC



Hình 1-14: Kết nối tín hiệu ngõ vào của PLC.

Hình 1-15: Kết nối tín hiệu ngõ ra của PLC.

Ngõ vào cấp nguồn 24 VDC

Nguồn âm (-) nối vào các chân 1M, 2M và M.

Nguồn dương (+) nối vào chân L+.

Ngõ ra cấp nguồn 240 VAC

Dây trung tính nối vào chân N

Dây pha nối vào chân 1L, 2L, 3L và L1.

Dây bảo vệ PE nối vào chân

5.2. Kết nối thiết bị ngoại vi:

Kết nối thiết bị ngoại vi là kết nối giữa PLC với các thiết bị ngõ vào và thiết bị ngõ ra.

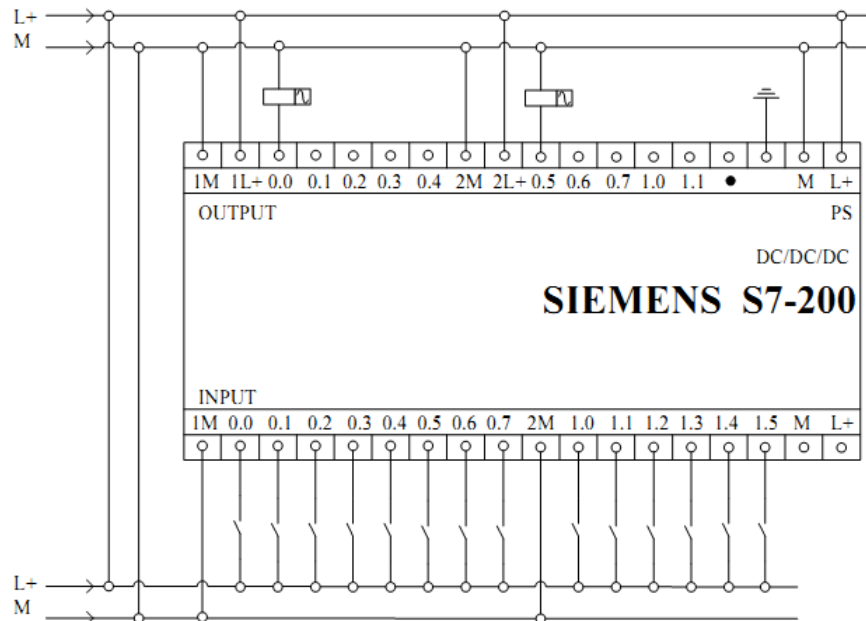
- Kết nối thiết bị ngõ vào:

Ngõ vào gồm: các công tắc, cảm biến, tiếp điểm, công tắc hành trình....

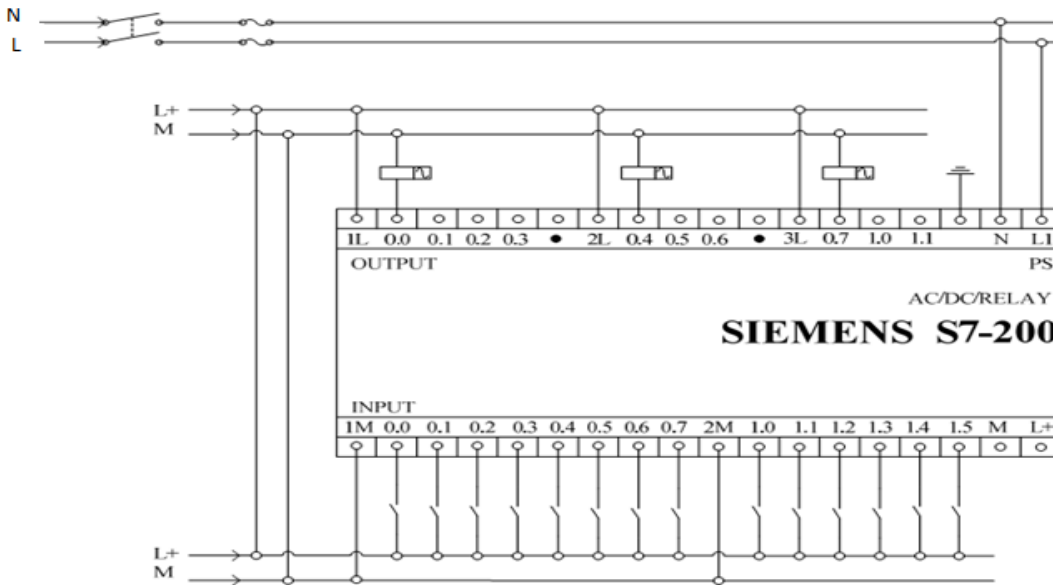
- Kết nối thiết bị ngõ ra:

Ngõ ra gồm: relay, công tắc tơ (contactor), van điện (Solenoid), đèn tín hiệu, động cơ...

CPU 224 DC/DC/DC



Hình 1-16: Kết nối ngõ vào/ ngõ ra cho PLC



Hình 17: Kết nối ngõ vào/ ngõ ra cho PLC

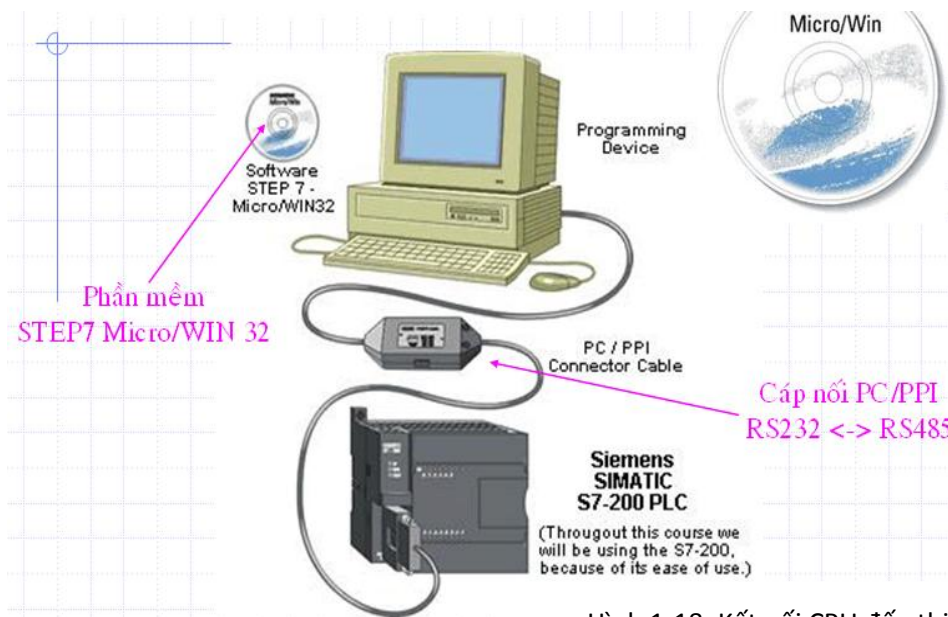
5.3. Kết nối CPU đến thiết bị lập trình:

Để kết nối S7 – 200 đến thiết bị lập trình ta dùng cáp RS232/PPI Multi – Master Cable theo trình tự:

Kết nối đầu RS232(được ký hiệu là PC) của cáp RS232/PPI Multi – Master Cable đến thiết bị lập trình.

Kết nối đầu RS485(được ký hiệu là PPI) của cáp RS232/PPI Multi – Master Cable đến S7 – 200.

Kiểm tra những switch chọn chế độ phải đúng.



Hình 1-18: Kết nối CPU đến thiết bị lập

6. Kiểm tra việc nối dây bằng phần mềm.

- Status Chart: Cho phép ta theo dõi giá trị của tất cả các biến trong vùng nhớ của PLC mà ta sử dụng trong chương trình. Đồng thời ta có thể cho các biến giá trị mới (không thể kể những biến dạng “Read Only”) để theo dõi hoạt động của chương trình.

	Address	Format	Current Value	New Value
1	off	Bit		
2	start	Bit		
3	dong co	Bit		
4		Signed		
5		Signed		
6		Signed		
7		Signed		

Tên biến Kiểu biến Giá trị hiện tại Giá trị mới

Hình 1-19: Bảng trạng thi

7. Cài đặt và sử dụng phần mềm STEP7-Micro/Win 32

7.1. Những yêu cầu đối với máy tính PC:

Máy tính cá nhân PC muốn cài đặt được phần mềm STEP7-Micro/Win phải thỏa mãn những yêu cầu sau:

640 Kb RAM (ít nhất phải có 500 Kb bộ nhớ còn trống)

Màn hình 24 dòng, 80 cột ở chế độ văn bản

Còn khoảng 2Mb trống trong ổ đĩa cứng

Có hệ điều hành MS-DOS ver 5.0 hoặc cao hơn

Bộ chuyển đổi RS232 – RS485 phục vụ ghép nối truyền thông giữa PC và PLC

7.2. Cài đặt và sử dụng phần mềm STEP 7 – Micro/Win 32

7.2.1. Cài đặt:

Kích đúp chuột vào file setup.exe để cài đặt chương trình, việc cài đặt diễn ra bình thường và gần giống với các phần mềm ứng dụng khác. Dưới đây mô tả cách cài đặt.



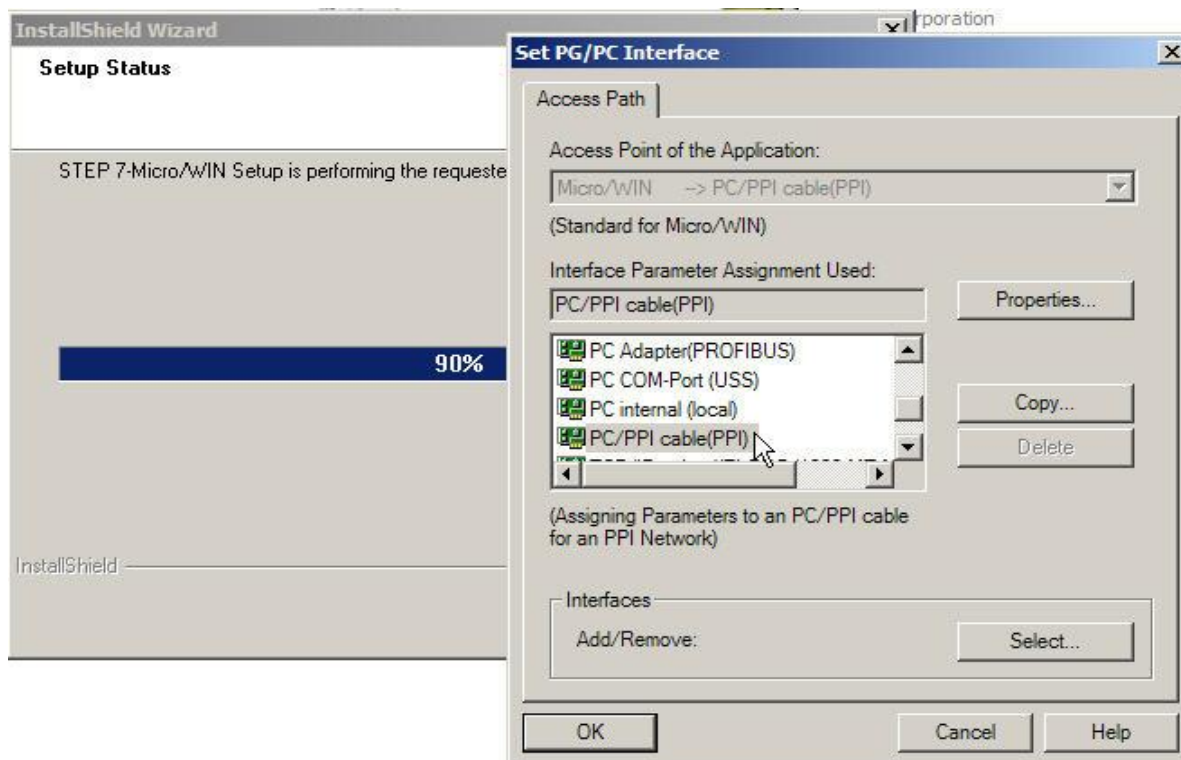
Hình 1- 19.1: Cài đặt



Chọn Next....

Hình 1-19.1: Cài đặt

Sau đó chương trình sẽ tự động cài đặt các file cần thiết. Lưu ý khi tới phần chọn giao tiếp máy tính ta nhớ chọn giao tiếp là PC/PPI, sau đó có thể chọn cổng COM hoặc USB, tốc độ truyền...(Tab Properties) tùy thuộc vào adapter mà chúng ta đang dùng.



Hình 1-19.2: Cài đặt

BÀI 2

ĐIỀU KHIỂN ON - OFF ĐỘNG CƠ KHÔNG ĐỒNG BỘ BA PHA

Mục tiêu:

- Trình bày được các chức năng của ngõ vào, ngõ ra số.
- Ứng dụng linh hoạt các chức năng trong các bài toán thực tế.
- Lập trình, kết nối, vận hành.
- Giải quyết các công việc một cách hệ thống theo nhóm.

Nội dung chính:

1.Giao diện của MicroWin 4.0:

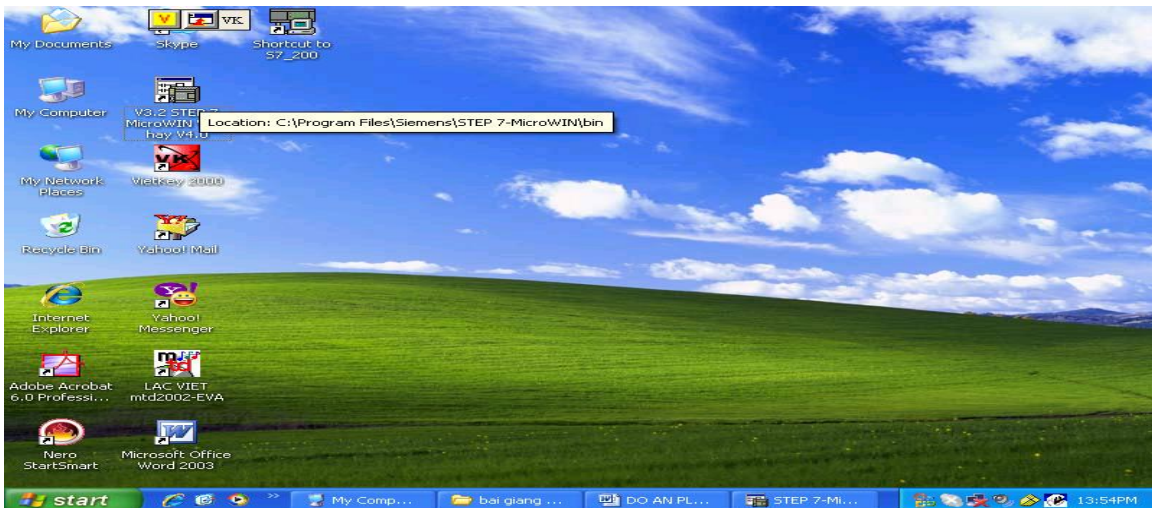
Tạo mới một Project: nhấp chuột vào menu File →New để mở một Project mới. Sau đó chọn Save as để đặt tên cho Project

2.Soạn thảo

Để bắt đầu quá trình soạn thảo ta tiến hành theo trình tự sau:

2.1.Khởi động chương trình:

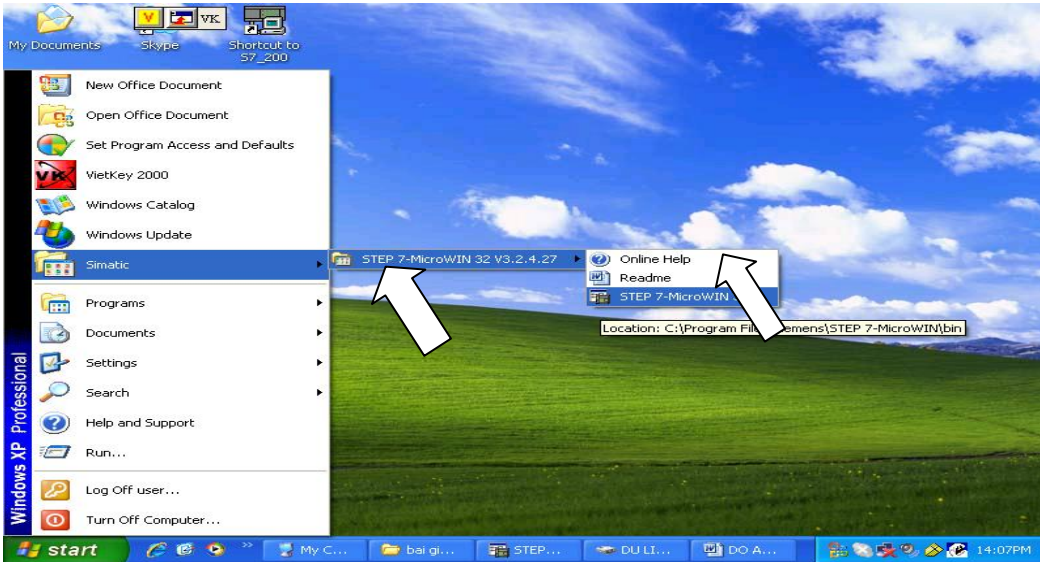
Cách 1: Click chuột vào biểu tượng STEP 7 - MicroWin V3.2 hay V4.0 ở màn hình(Desktop) để bắt đầu soạn thảo như .



Hình 2-1 : Khởi động chương trình

Cách 2:

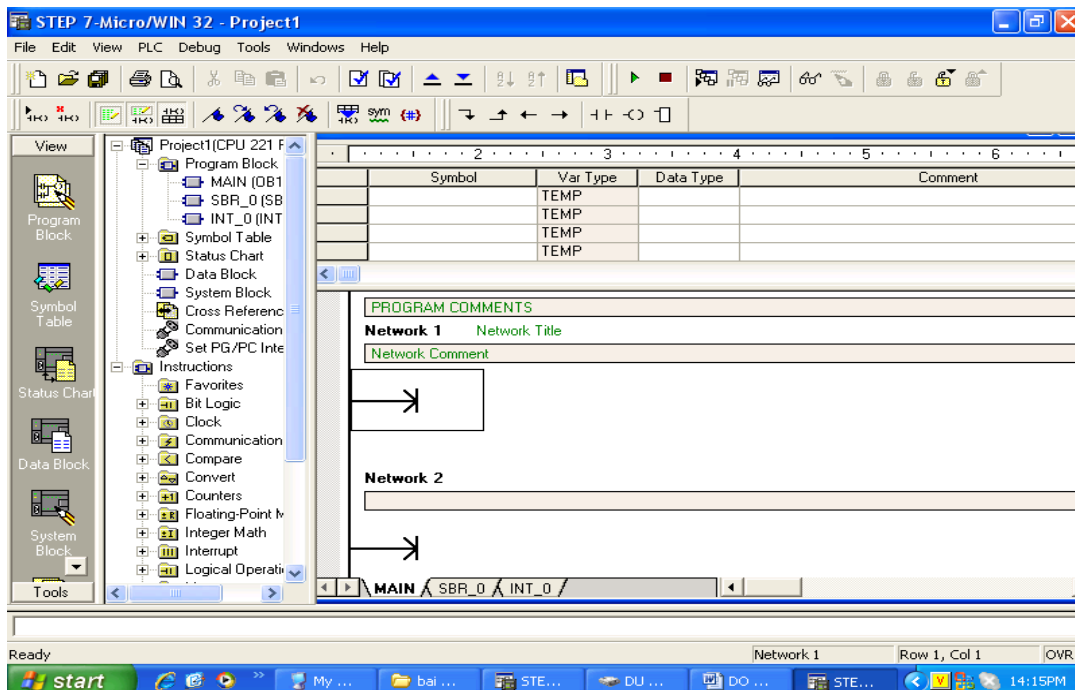
Nhấp Start » Simatic » STEP 7 - MicroWin V3.2 hay V4.0 như .



Hình 2-2 : Khởi động chương trình

2.2. Soạn thảo chương trình:

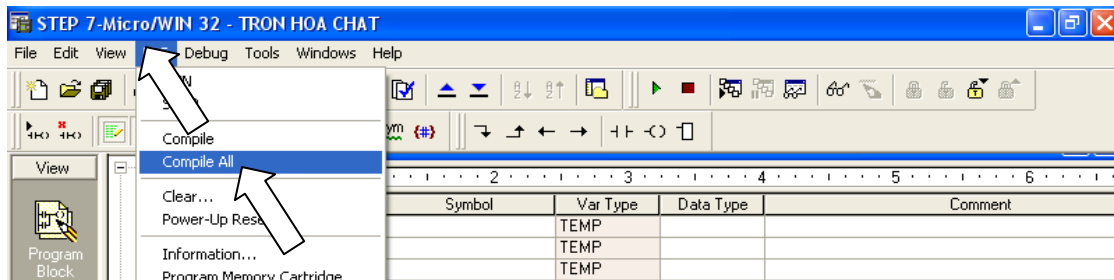
Khi khởi động màn hình soạn thảo sẽ xuất hiện như hình . Sau đó ta tiến hành soạn thảo chương trình theo yêu cầu (ngôn ngữ dạng LAD).



Hình 2-3 : Giao diện soạn thảo

2.3. Kiểm tra lỗi:

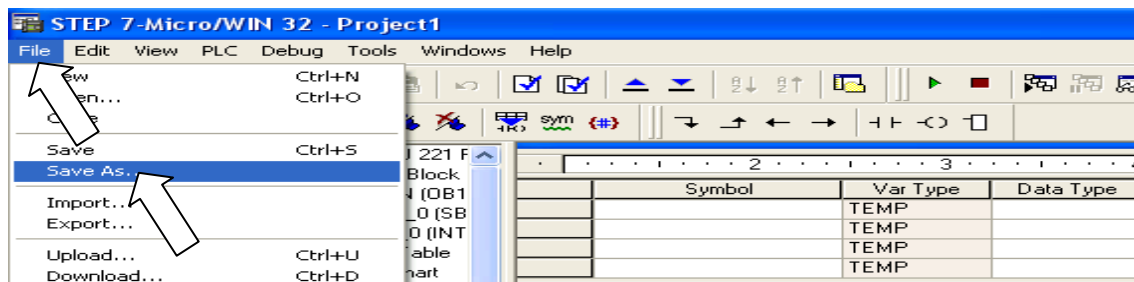
Sau khi soạn thảo chương trình xong ta nhấp vào PLC/ Compile All để kiểm tra toàn bộ chương trình như hình:



Hình 2-4 : Giao diện kiểm tra

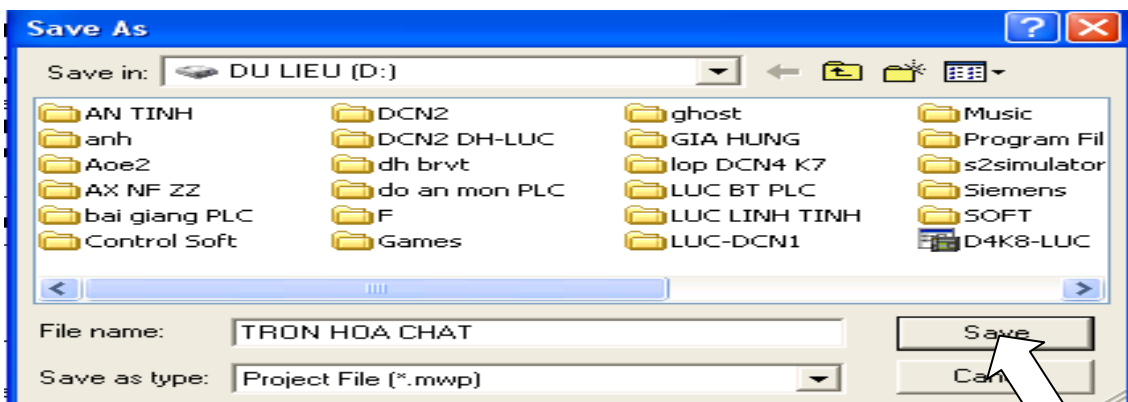
2.4. Lưu chương trình:

Sau khi soạn thảo chương trình xong và đã kiểm tra lỗi, chương trình đúng yêu cầu ta vào File/ Save As để lưu chương trình.



Hình 2-5 : Lưu chương trình


Trong hộp thoại Save As ta chọn tên ổ đĩa, tên File. Ví dụ tên File là TRỘN HOÁ CHẤT chẳng hạn, sau đó chọn Save để lưu như hình.



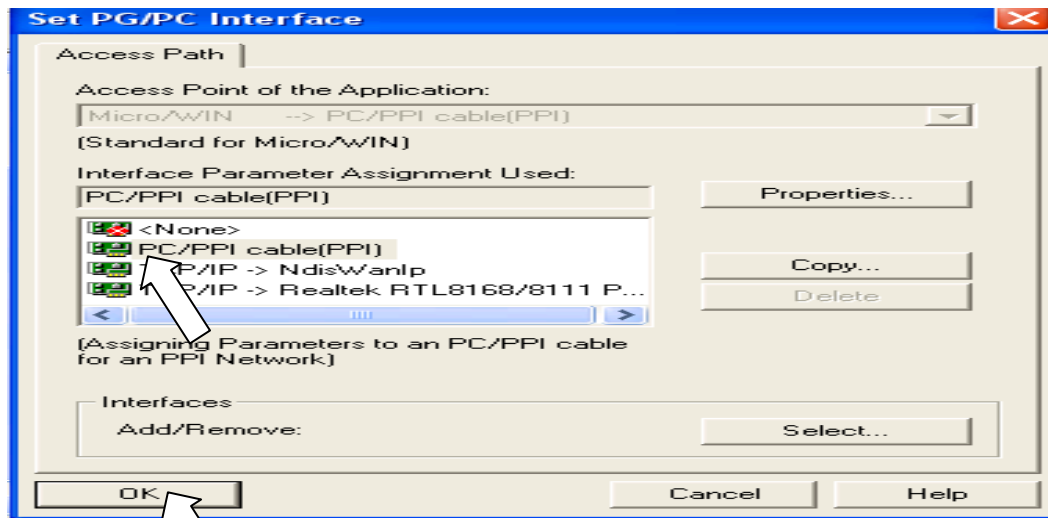
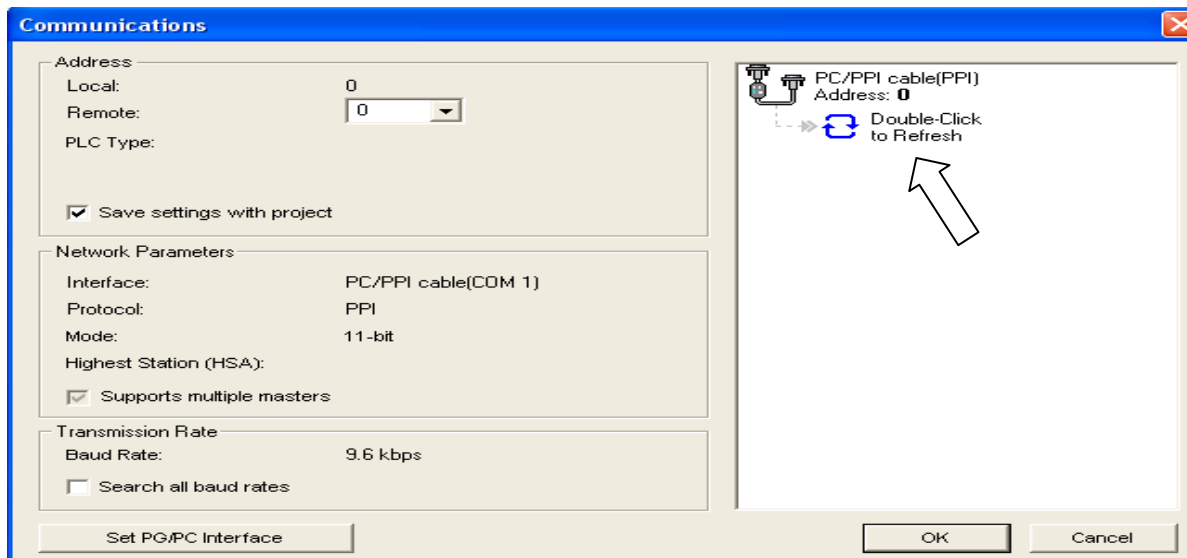
Hình 2-6 : Lưu chương trình trộn hóa chất

2.5. Thiết lập thông số cho hộp thoại Communications:(truyền thông)



Click vào biểu tượng  ở các khối chức năng bên trái màn hình soạn thảo để thiết đặt giao tiếp cho Click – MicroWin. Khối chức năng sẽ hiện thị như hình (H.16):

* Click đúp vào PC/PPI cable(PPI), hoặc kích vào Set PG/PC Interface chọn PC/PPI cable(PPI) ⇒ OK



Hình 2-8 : Thiết lập truyền thông

* Trong hộp thiết lập thông số giao tiếp ta kiểm tra các thông số sau đây:

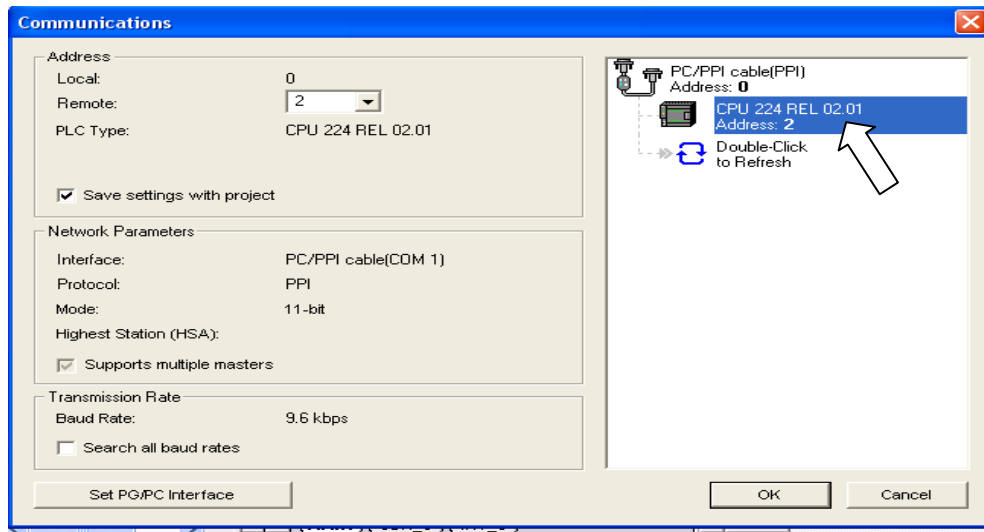
Thiết lập địa chỉ của cáp PC/PPI Cable bằng 0

Thiết lập cổng giao tiếp của cáp PC/PPI Cable là COM 1

Thiết lập tốc độ truyền là 9.6 kbps

2.6. Thiết lập sự kết nối với S7 – 200:


Để thiết lập sự kết nối giữa thiết bị lập trình và S7 – 200 ta Click đúp vào Refresh trong hộp kết nối. STEP 7 sẽ tìm và hiển thị những CPU được kết nối., sau đó nhấn OK.

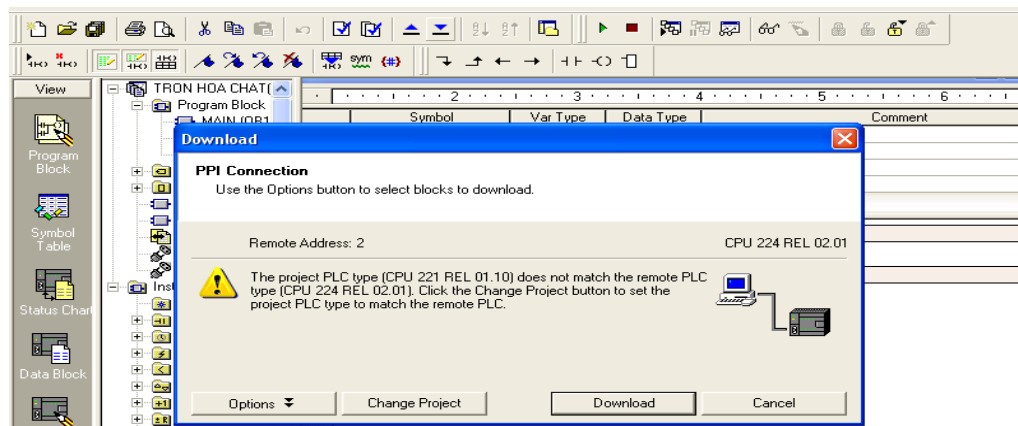


Hình 2-9 : Thiết lập kết nối

Chú ý: Nếu không tìm thấy ta kiểm tra lại việc lập các thông số kết nối và làm lại bước này.


2.7. Dowload chương trình:

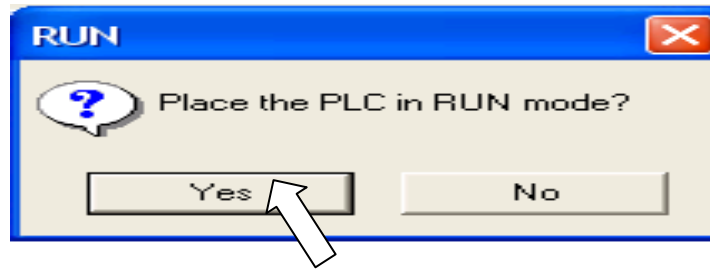
Để Dowload chương trình đến S7 – 200 ta Click vào biểu tượng  , màn hình sẽ hiển thị hộp thoại và ta chọn Dowload để load chương trình đến PLC:



Hình 2-10 : Thiết lập download

2.8. Chạy chương trình:

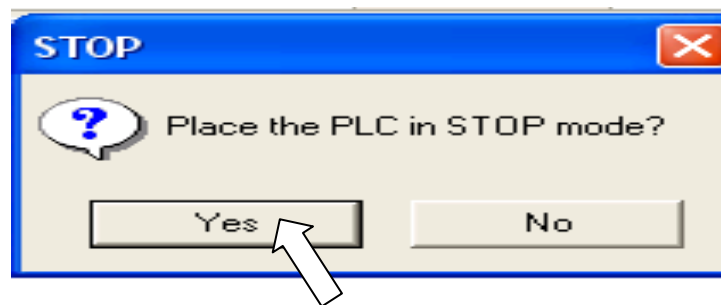
Khi đã Dowload chương trình thành công ta nhập vào PLC⇒RUN hoặc biểu tượng  , hộp thoại xuất hiện ta chọn Yes.



Hình 2-11 : Thiết lập chế độ RUN

2.9. Dừng chương trình:

Muốn Dừng chương trình nhấp vào PLC \Rightarrow STOP hoặc biểu tượng , hộp thoại xuất hiện ta chọn Yes, như hình (H.28):



Hình 2-12 : Thiết lập chế độ stop

3. Lập trình mô phỏng trên máy tính:

Chương trình mô phỏng S7-200 – Simulator dùng để người học thực tập khi không có PLC thực. Để sử dụng phần mềm S7-200 – Simulator trong máy tính ngoài cần phải có phần mềm lập trình MicroWin V3.2, hay V4.0 còn cần thêm phần mềm mô phỏng S7-200 – Simulator.

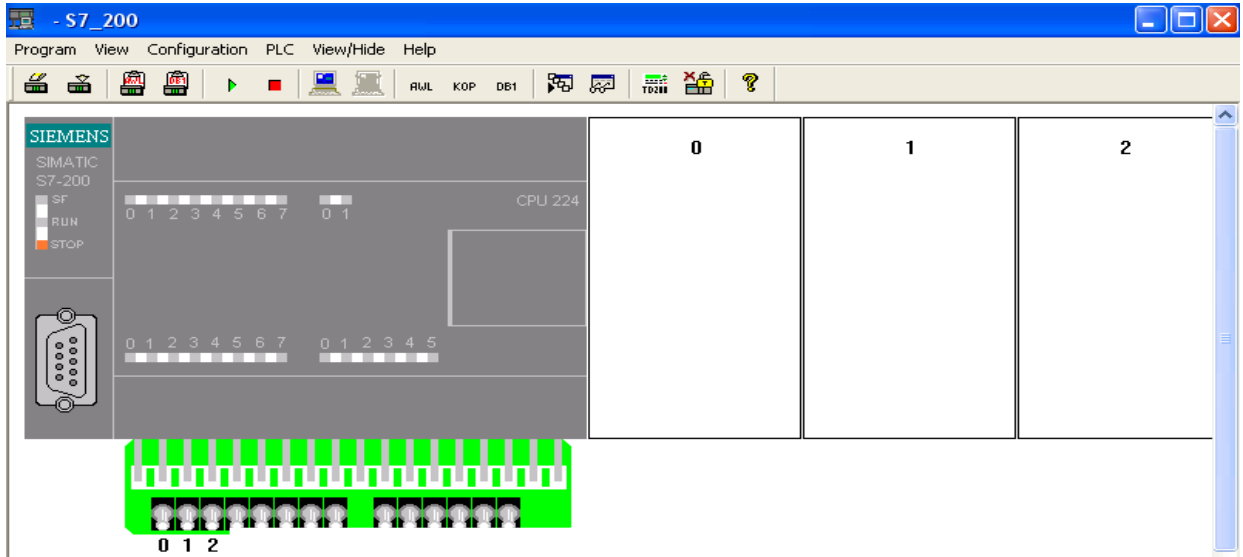
Quá trình mô phỏng được thực hiện theo trình tự sau:

Soạn thảo chương trình trên phần mềm MicroWin V3.2, hay V4.0. sau đó vào File/Export để lưu file vào ổ đĩa (file có đuôi .awl) Khởi động phần mềm mô phỏng bằng cách

nhấp đúp vào biểu tượng




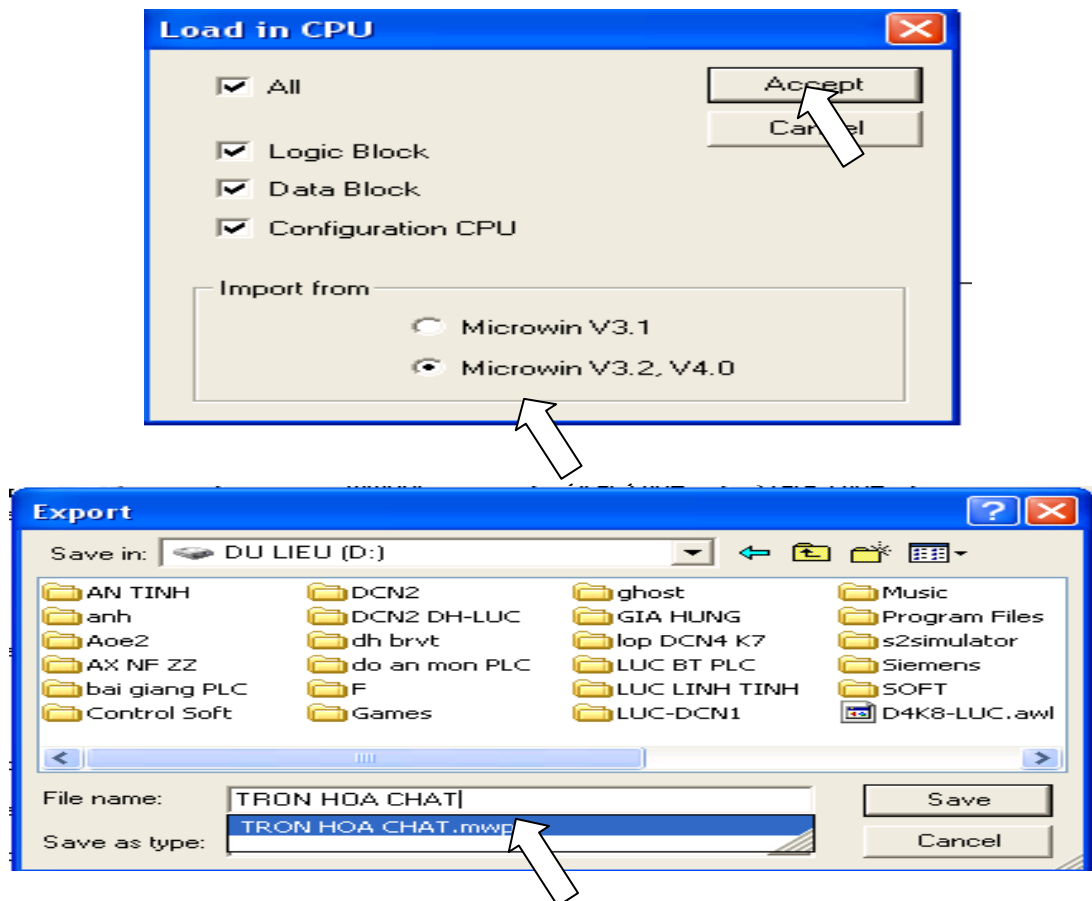
Giao diện mô phỏng sẽ xuất hiện.



Hình 2-13 : Giao diện mô phỏng

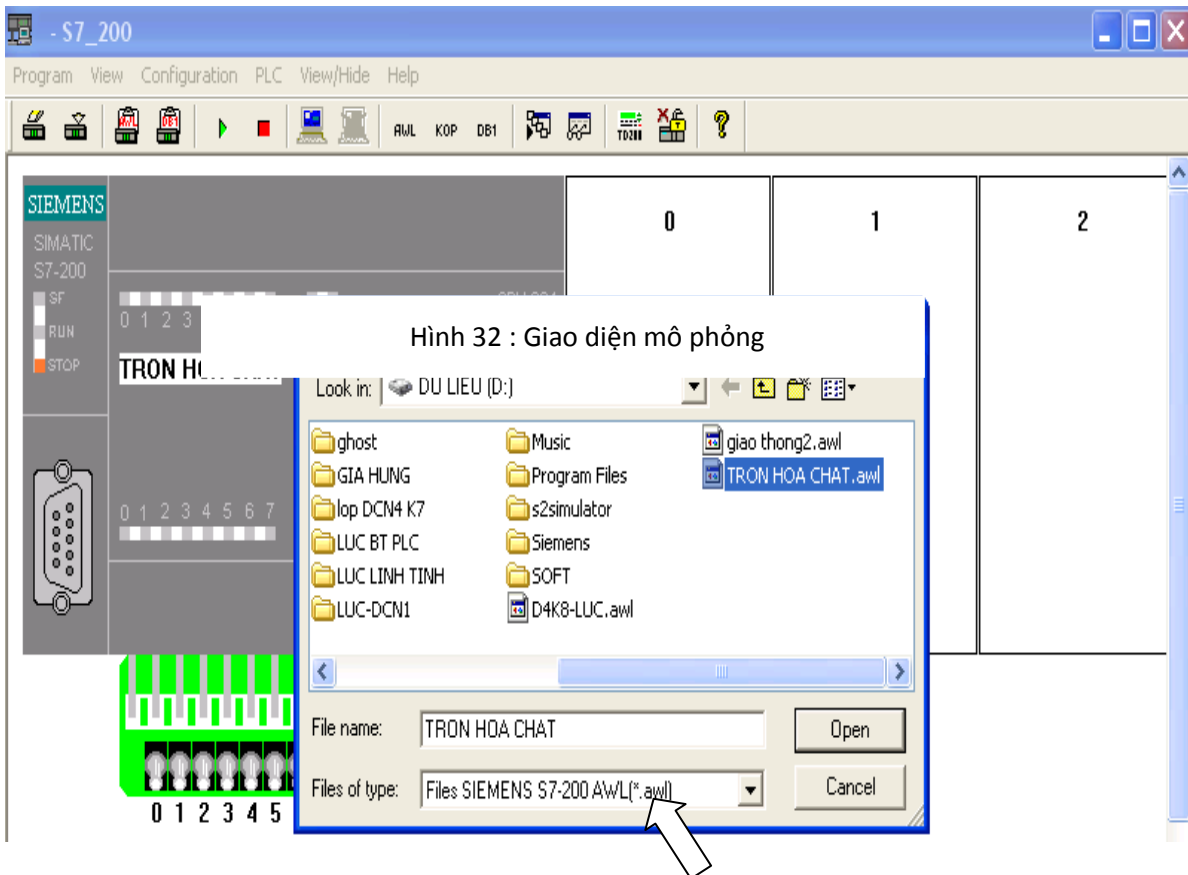
Trường hợp nếu xuất hiện hộp thoại thì ta nhập mã bảo vệ 6596, sau đó Click OK.

Khi màn hình mô phỏng sẽ xuất hiện ta nhấp vào biểu tượng , sau đó xuất hiện hộp thoại ta chọn MicroWin V3.2, V4.0 và nhấp vào Accept như hình.



Hình 2-15 : Lưu file mô phỏng

Hộp thoại yêu cầu nhập tên bài cần mô phỏng. Ta chọn tên bài mô phỏng và Click Open. Sau đó chọn RUN để bắt đầu mô phỏng.



Hình 2-16: Giao diện mô phỏng

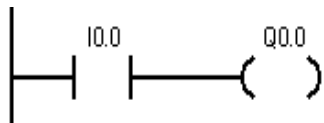
4. Các Liên Kết Logic

4.1. Lệnh vào/ra và các lệnh tiếp điểm đặc biệt:

4.2. Load (LD): Lệnh LD nạp giá trị logic của một tiếp điểm vào trong bit đầu tiên của ngăn xếp, các giá trị cũ còn lại trong ngăn xếp bị đẩy lùi xuống một bit.

Toán hạng gồm I, Q, M, SM, V, C, T.

+ Dạng LAD: Tiếp điểm thường mở sẽ đóng nếu I0.0 = 1



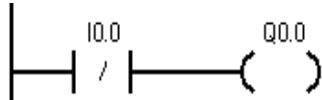
+ Dạng STL: LD I0.0

= Q0.0

4.3. Load Not (LDN): Lệnh LDN nạp giá trị logic nghịch đảo của một tiếp điểm vào trong bit đầu tiên của ngăn xếp, các giá trị cũ còn lại trong ngăn xếp bị đẩy lùi xuống một bit.

Toán hạng gồm I, Q, M, SM, V, C, T.

+ Dạng LAD: Tiếp điểm thường đóng sẽ mở nếu I0.0=1

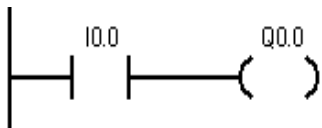


+ Dạng STL: LD I0.0

= Q0.0

4.4. OUTPUT (=): Lệnh sao chép nội dung của bit đầu tiên trong ngăn xếp vào bit được chỉ định trong lệnh. Nội dung của ngăn xếp không bị thay đổi.

+ Dạng LAD: Nếu I0.0=1 thì Q0.0=1



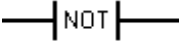
+ Dạng STL: Giá trị logic I0.0 được đưa vào bit đầu tiên của ngăn xếp , và bit này được sao chép vào bit ngõ ra Q0.0

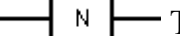
LD I0.0

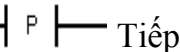
= Q0.0

4.5. Lệnh tiếp điểm đặc biệt:

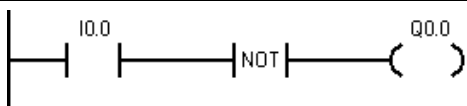
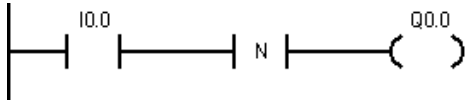
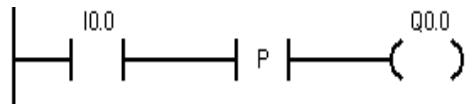
Có thể dùng các lệnh tiếp điểm đặc biệt để phát hiện sự chuyển tiếp trạng thái của xung (sườn xung) và đảo lại trạng thái của dòng cung cấp (giá trị đỉnh của ngăn xếp). LAD sử dụng các tiếp điểm đặc biệt này để tác động vào dòng cung cấp. Các tiếp điểm đặc biệt không có toán hạng riêng của chúng vì thế phải đặt chúng trước cuộn dây hoặc hộp đầu ra.

- Tiếp điểm đảo:  Tiếp điểm đảo trạng thái của dòng điện cung cấp. Nếu dòng cung cấp có tiếp điểm đảo thì nó bị ngắt mạch.

- Tiếp điểm tác động cạnh xuống:  Tiếp điểm chuyển đổi âm cho phép dòng cung cấp thông mạch trong một vòng quét thì sườn xung điều khiển từ 0 lên 1

- Tiếp điểm tác động cạnh lên:  Tiếp điểm chuyển đổi dương cho phép dòng cung cấp thông mạch trong một vòng quét thì sườn xung điều khiển từ 1 xuống 0

Ví dụ minh họa:

Dạng LAD	Dạng STL
	LD I0.0 NOT = Q0.0
	LD I0.0 ED = Q0.0
	LD I0.0 EU = Q0.0

4.6. Một số tiếp điểm trong vùng nhớ đặc biệt:

ô nhớ	Mô tả
SM0.0	Luôn có giá trị logic bằng 1
SM0.1	Có giá trị logic bằng 1 ở vòng quét đầu tiên
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 vòng quét tiếp theo)
SM0.4	Phát nhịp 60 giây (0- cho 30 giây đầu; 1- cho 30 giây sau)
SM0.5	Phát nhịp 1 giây (0- cho 0.5 giây đầu; 1- cho 0.5 giây sau)
SM0.6	Nhịp vòng quét (1- cho vòng quét luân phiên)
SM0.7	Bit chọn chế độ cho PLC (0- TERM; 1- RUN)

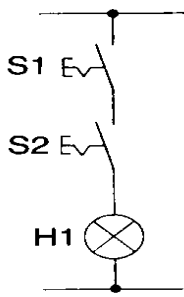
5. Các lệnh liên kết logic cơ bản

5.1. Lệnh AND (A)

Lệnh A phối hợp giá trị logic của một tiếp điểm n với giá trị bit đầu tiên của ngăn xếp. Kết quả của phép tính được đặt lại vào bit đầu tiên trong ngăn xếp. Giá trị của các bit còn lại trong ngăn xếp không bị thay đổi

Cú pháp STL: A n

Ví dụ: Hình vẽ mô tả sơ đồ mạch điện của một liên kết AND



Đèn H1 chỉ sáng khi tất cả các công tắc được đóng lại. Khi 1 công tắc hở mạch thì đèn H1 cũng bị cắt mạch. Liên kết AND có trạng thái 1 khi tất cả các ngõ vào có trạng thái 1.

Để giải quyết vấn đề này, trước tiên ta cần phải lập một bảng xác lập các ngõ vào/ra để kết nối với PLC.

Lập bảng địa chỉ ngõ vào - ngõ ra		
KÝ HIỆU	ĐỊA CHỈ (TOÁN HẠNG)	GIẢI THÍCH (MÔ TẢ)
S1	I0.0	Công tắc thường hở
S2	I0.1	Công tắc thường hở
H1	Q0.0	Đèn báo

Chương trình được viết trong PLC ở các dạng LAD, FBD và STL được cho như sau

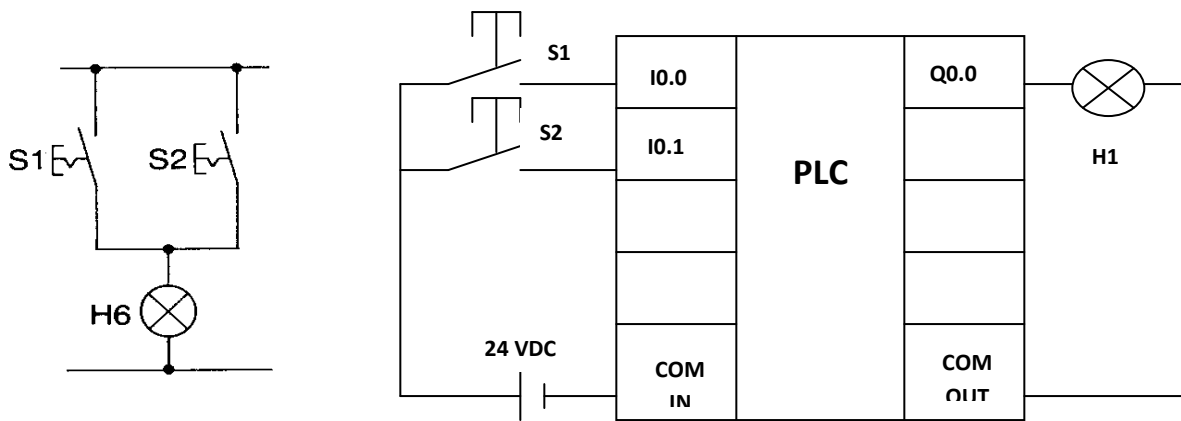
LAD	FBD	STL
		<pre>LD I0.0 A I0.1 = Q0.0</pre>

5.2.Lệnh OR (O)

Lệnh OR phối hợp giá trị logic của một tiếp điểm n với giá trị bit đầu tiên của ngăn xếp. Kết quả phép tính được đặt lại vào bit đầu tiên trong ngăn xếp. Giá trị của các bit còn lại trong ngăn xếp không bị thay đổi.

Cú pháp ở STL: O n

Ví dụ: Hình vẽ mô tả sơ đồ mạch điện của một liên kết OR



Hình 2-18 : Sơ đồ kết nối plc

Đèn H6 sáng khi một hoặc tất cả các công tắc đều đóng mạch. Ngõ ra của liên kết OR có trạng thái 1 khi ít nhất một trong các ngõ vào có trạng thái 1.

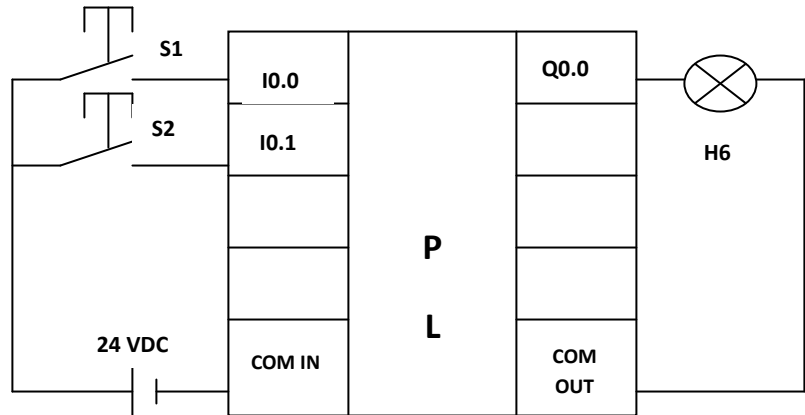
Để giải quyết vấn đề này, trước tiên ta cần phải lập một bảng xác lập các ngõ vào/ra để kết nối với PLC.

Lập bảng địa chỉ ngõ vào - ngõ ra		
Ký hiệu	Địa chỉ (toán hạng)	Giải thích (mô tả)
S1	I0.0	Công tắc thường hở
S2	I0.1	Công tắc thường hở
H6	Q0.0	Đèn báo

Chương trình được viết trong PLC ở các dạng LAD, FBD và STL được cho như sau

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

Sơ đồ kết nối với PLC như hình vẽ



Hình 2-19 :Sơ đồ kết nối plc

6. Liên kết các cổng logic cơ

6.1. Liên kết AND trước OR

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

6.2. Liên kết OR trước AND

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

7. Lập trình mạch điện điều khiển động cơ không đồng bộ 3 pha quay 1 chiều.

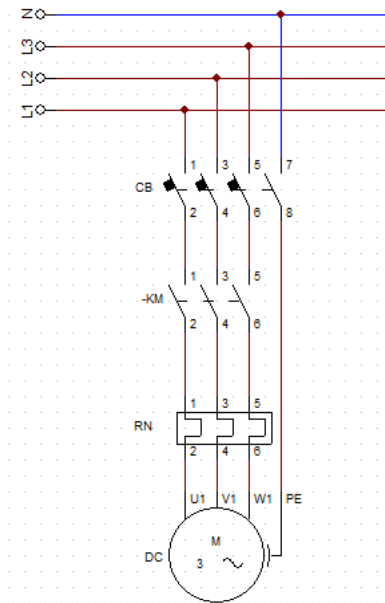
7.1. Yêu cầu công nghệ

- Nhấn Start, động cơ khởi động và làm việc.
- Nhấn Stop, động cơ ngừng
- Mạch có bảo vệ quá tải

7.2. Nhiệm vụ:

- Vẽ được sơ đồ mạch động lực.
- Lập bảng trạng thái.
- Vẽ sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển PLC S7-200.
- Lập trình trên phần mềm PLC S7 – 200.
- Kết nối các thiết bị ngoại vi, down load chương trình, chạy chương trình.
- Kiểm tra lỗi đảm bảo hoạt động đúng chương trình.

7.3. Sơ đồ mạch động lực:

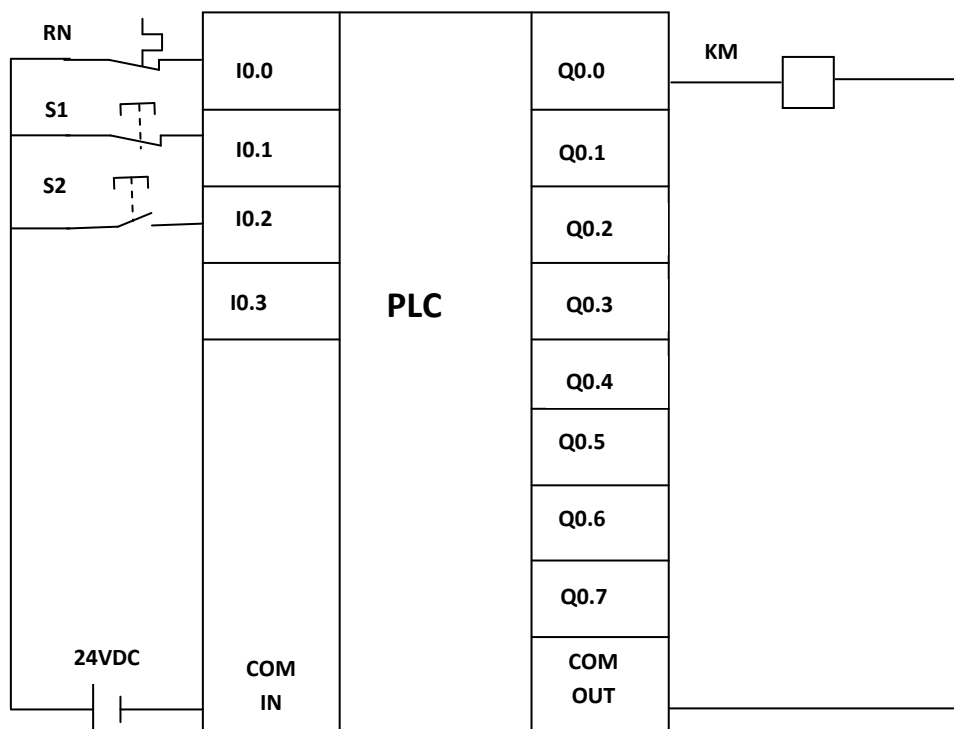


Hình 2-20 :Sơ đồ mạch động lực

7.4.Lập bảng địa chỉ ngõ vào - ngõ ra.

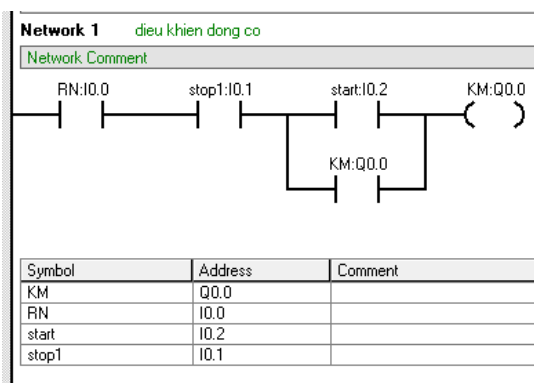
Kí hiệu	Địa chỉ	Mô tả
RN	I0.0	Bảo vệ quá tải
S1	I0.1	Stop
S2	I0.2	Start
KM	Q0.0	Khởi động từ không chế động cơ

7.5. Sơ đồ kết nối PLC với thiết bị ngoại vi



Hình 2-21 :Sơ đồ kết nối PLC

7.6. Chương trình



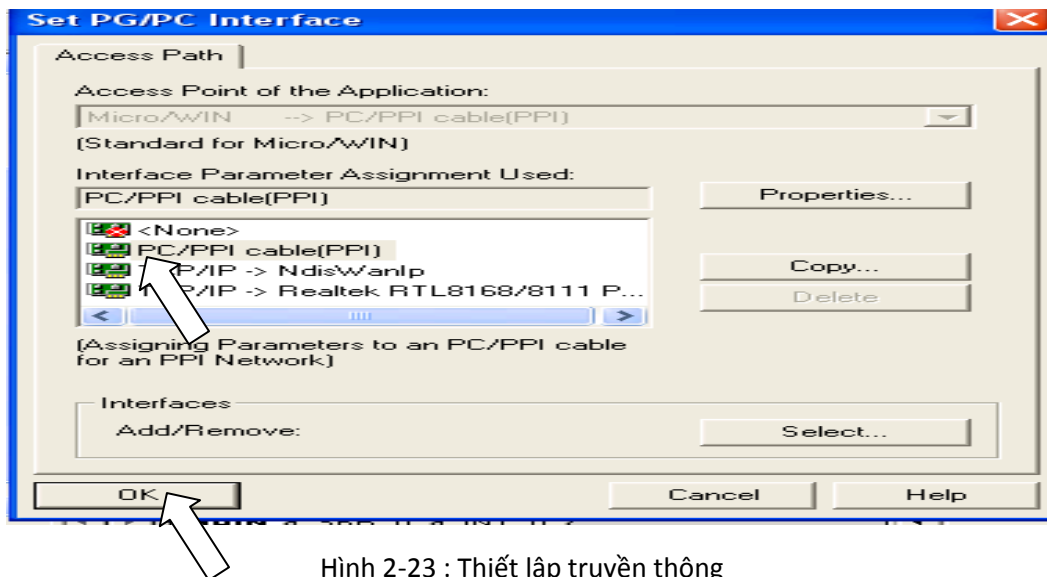
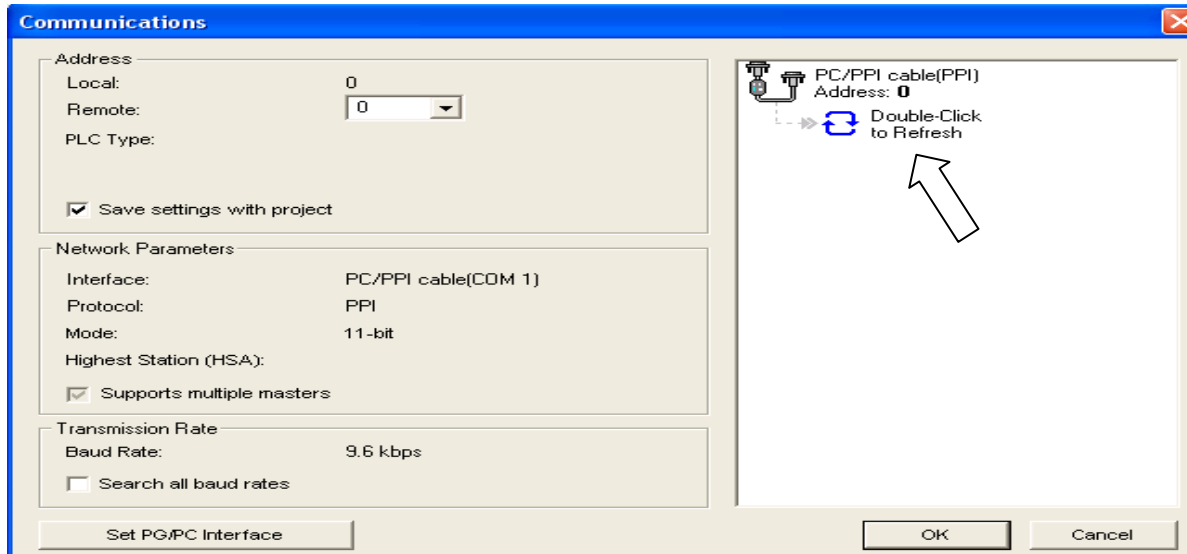
7.7. Kết nối các thiết bị ngoại vi, download chương trình, chạy chương trình.

7.7.1. Thiết lập thông số cho hộp thoại Communications:(truyền thông)



Click vào biểu tượng Communications ở các khối chức năng bên trái màn hình soạn thảo để thiết lập giao tiếp cho Click – MicroWin. Khối chức năng sẽ hiện thị như hình (H.16):

* Click đúp vào PC/PPI cable(PPI), hoặc kích vào Set PG/PC Interface chọn PC/PPI cable(PPI) ⇒ OK như hình (H.26.1), (H.26.2)



Hình 2-23 : Thiết lập truyền thông

* Trong hộp thiết lập thông số giao tiếp ta kiểm tra các thông số sau đây:

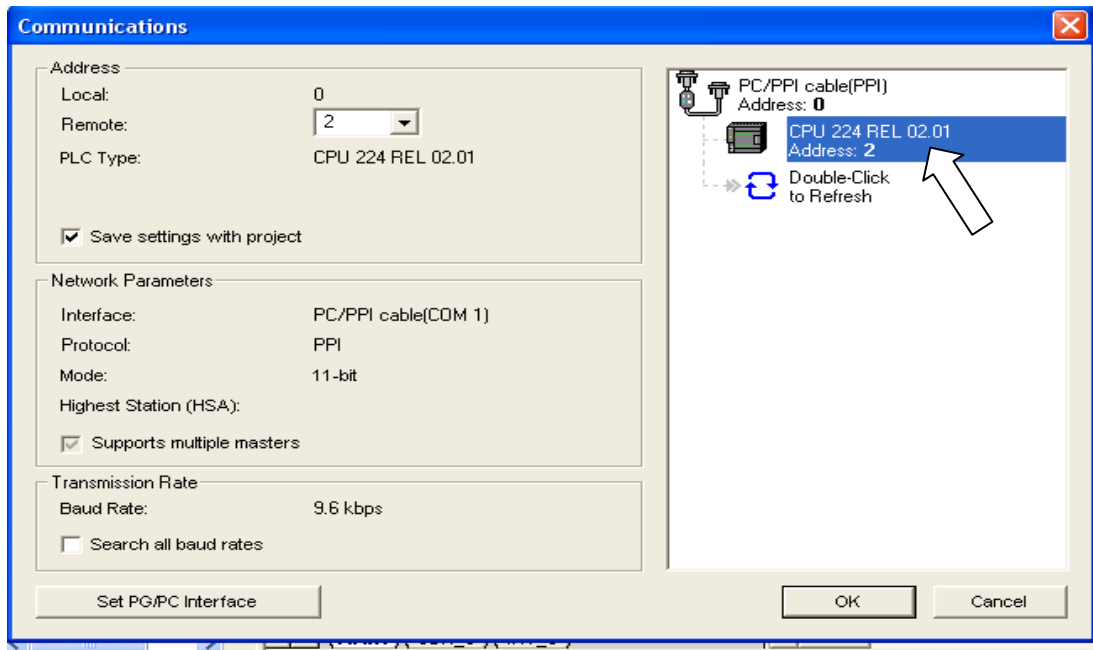
Thiết lập địa chỉ của cáp PC/PPI Cable bằng 0

Thiết lập cổng giao tiếp của cáp PC/PPI Cable là COM 1

Thiết lập tốc độ truyền là 9.6 kbps

7.7.2. Thiết lập sự kết nối với S7 – 200:


Để thiết lập sự kết nối giữa thiết bị lập trình và S7 – 200 ta Click đúp vào Refresh trong hộp kết nối. STEP 7 sẽ tìm và hiển thị những CPU được kết nối., sau đó nhấn OK.

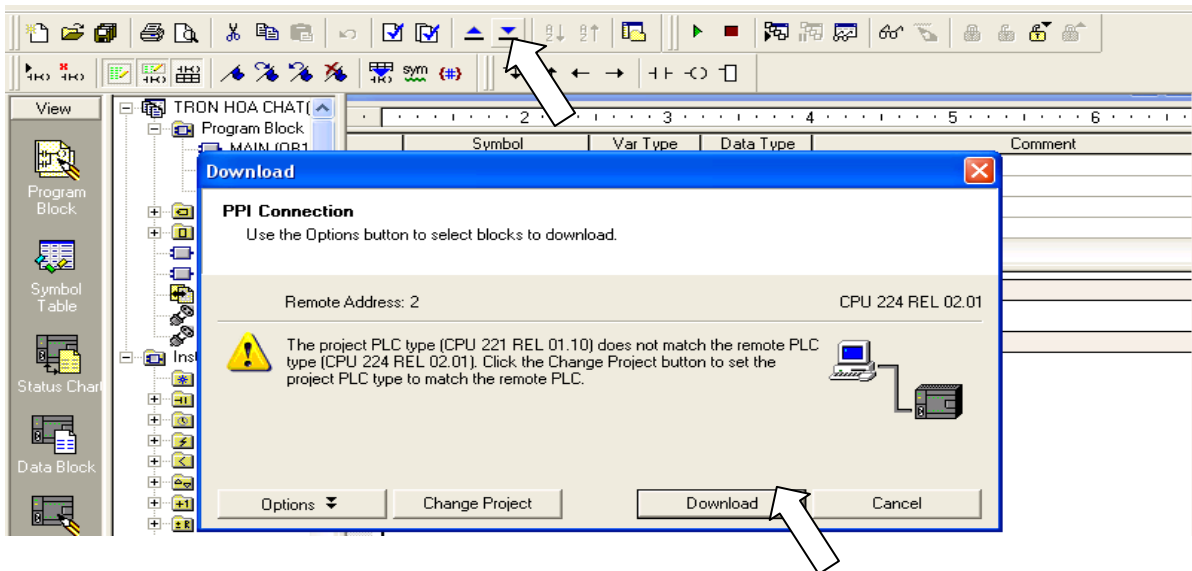


Hình 2-24 : Thiết lập kết nối

Chú ý: Nếu không tìm thấy ta kiểm tra lại việc lập các thông số kết nối và làm lại bước này.


7.7.3. Download chương trình:

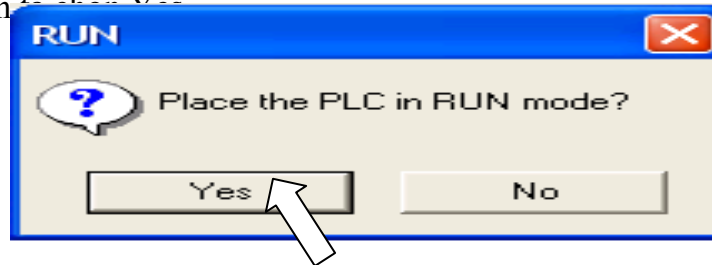
Để Download chương trình đến S7 – 200 ta Click vào biểu tượng  , màn hình sẽ hiện thị hộp thoại và ta chọn Download để load chương trình đến PLC:



Hình 2-25 : Thiết lập download

7.7.4. Chạy chương trình:

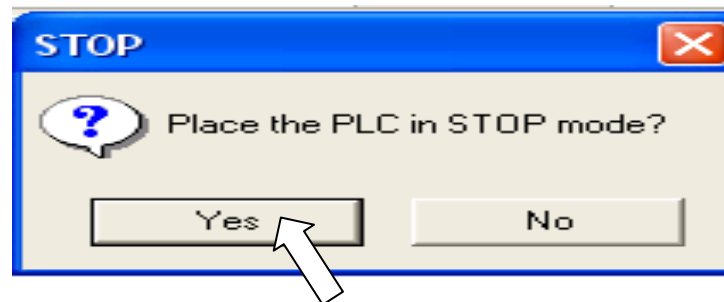
Khi đã Dowload chương trình thành công ta nhấp vào PLC⇒RUN hoặc biểu tượng , hộp thoại xuất hiện ta chọn Yes



Hình 2-26 : Thiết lập chế độ RUN

7.7..5. Dừng chương trình:

Muốn Dừng chương trình nhấp vào PLC⇒STOP hoặc biểu tượng , hộp thoại xuất hiện ta chọn Yes, như hình :



Hình 2-27 : Thiết lập chế độ stop

8. Câu hỏi ôn tập

Bài tập: Lập trình mạch điện điều khiển hai động cơ không đồng bộ 3 pha quay 1 chiều.

- Nhấn Start 1, động cơ 1 khởi động và làm việc.
- Nhấn Start 2, động cơ 2 khởi động và làm việc.
- Nhấn Stop, 2 động cơ ngừng
- Mạch có bảo vệ quá tải cho 2 động cơ

Bài 3:

ĐIỀU KHIỂN ĐỘNG CƠ 1 CHIỀU

Mục tiêu:

- Trình bày được các chức năng của hàm đặc biệt.
- Ứng dụng linh hoạt các chức năng của đặc biệt trong các bài toán thực tế
- Lập trình, kết nối, vận hành.
- Giải quyết các công việc một cách hệ thống theo nhóm.

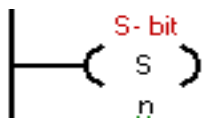
Nội dung chính :

1. Các lệnh ghi / xóa (set/ reset) giá trị cho tiếp điểm

1.1. Mạch nhớ R-S: Mạch này có hai trạng thái tín hiệu ở ngõ ra tương ứng với các trạng thái tín hiệu đặt ở ngõ vào. Nếu ngõ vào có trạng thái 1 thì ngõ ra có tín hiệu 1, khi ngõ vào có trạng thái 1 thì ngõ ra có tín hiệu 0. người ta gọi mạch này là mạch nhớ tín hiệu (giống như mạch tự giữ trong điều khiển dùng role). Thay đổi trạng thái ngõ ra: Đặt (**Set**) hoặc Xoá (**Reset**)

1.2.lệnh Set (S):

Lệnh dùng để đóng các điểm gián đoạn đã được thiết kế. Trong LAD, logic điều khiển dòng điện đóng các cuộn dây đầu ra. Khi dòng điều khiển đến các cuộn dây thì các cuộn dây đóng các tiếp điểm. Trong STL, lệnh truyền trạng thái bit đầu tiên của ngăn xếp đến các điểm thiết kế. Nếu bit này có giá trị bằng 1, các lệnh S đóng một tiếp điểm hoặc một dãy các tiếp điểm (giới hạn từ 1 đến 255). Nội dung của ngăn xếp không bị thay đổi bởi các lệnh này.



Trong đó: + S: Set

+ S- bit: Star bit (bit bắt đầu)

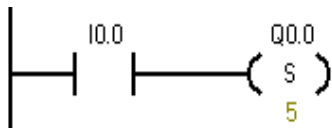
+ n: Số bit của chuỗi

+ Dạng LAD: Đóng một mảng bao gồm n các tiếp điểm kể từ địa chỉ S-bit, Toán hạng gồm I, Q, M, SM, V, C, T. (bit)

+ Dạng STL: Ghi giá trị logic vào một mảng gồm n bit kể từ địa chỉ S-bit

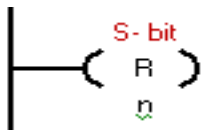
VD:Đưa 5 bit lên 1 bắt đầu từ Q0.0 đến Q0.4

+ Dạng LAD:



+ Dạng STL: LD I0.0
S Q0.0, 5

1.3.lệnh ReSet (R): Lệnh dùng để ngắt các điểm gián đoạn đã được thiết kế. Trong LAD, logic điều khiển dòng điện ngắt các cuộn dây đầu ra. Khi dòng điều khiển đến các cuộn dây thì các cuộn dây mở các tiếp điểm. Trong STL, lệnh truyền trạng thái bit đầu tiên của ngăn xếp đến các điểm thiết kế. Nếu bit này có giá trị bằng 1, các lệnh R sẽ ngắt một tiếp điểm hoặc một dãy các tiếp điểm (giới hạn từ 1 đến 255). Nội dung của ngăn xếp không bị thay đổi bởi các lệnh này.



Trong đó: + R: Reset

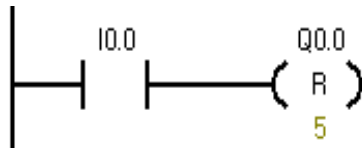
+ S-bit: Star bit (bit bắt đầu)

+ n: Số bit của chuỗi

+ Dạng LAD: Ngắt một mảng bao gồm n các tiếp điểm kể từ địa chỉ S-bit. Nếu S-bit lại chỉ vào Timer hoặc Counter thì lệnh sẽ xóa bit đầu ra của Timer hoặc Counter đó. Toán hạng gồm I, Q, M, SM, V, C, T. (bit)

+ Dạng STL: Xóa một mảng gồm n bit kể từ địa chỉ S-bit. Nếu S-bit lại chỉ vào Timer hoặc Counter thì lệnh sẽ xóa bit đầu ra của Timer hoặc Counter đó.

VD:Đưa 5 bit từ 1 xuống 0 bắt đầu từ Q0.0 đến Q0.4



LD I0.0
R Q0.0, 5

2. Điều khiển động cơ không đồng bộ 3 pha quay 2 chiều.

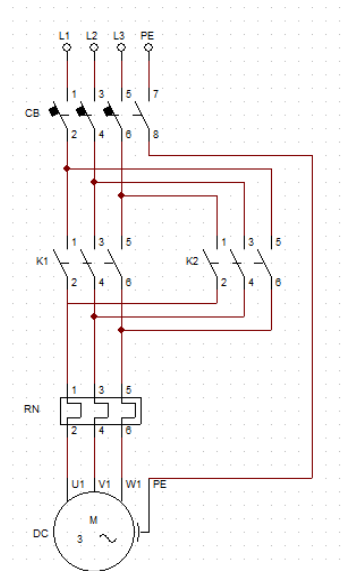
2.1. Yêu cầu công nghệ

- Nhấn MT động cơ khởi động và làm việc quay theo chiều thuận.
- Nhấn MN động cơ khởi động và làm việc quay theo chiều ngược.
- Nhấn D động cơ ngừng
- Mạch có bảo vệ quá tải

2.2. Nhiệm vụ:

- Vẽ được sơ đồ mạch động lực.
- Lập bảng trạng thái.
- Vẽ sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển PLC S7-200.
- Lập trình trên phần mềm PLC S7 – 200.
- Kết nối các thiết bị ngoại vi, down load chương trình, chạy chương trình.
- Kiểm tra lỗi đảm bảo hoạt động đúng chương trình.

2.2.1. Sơ đồ mạch động lực:

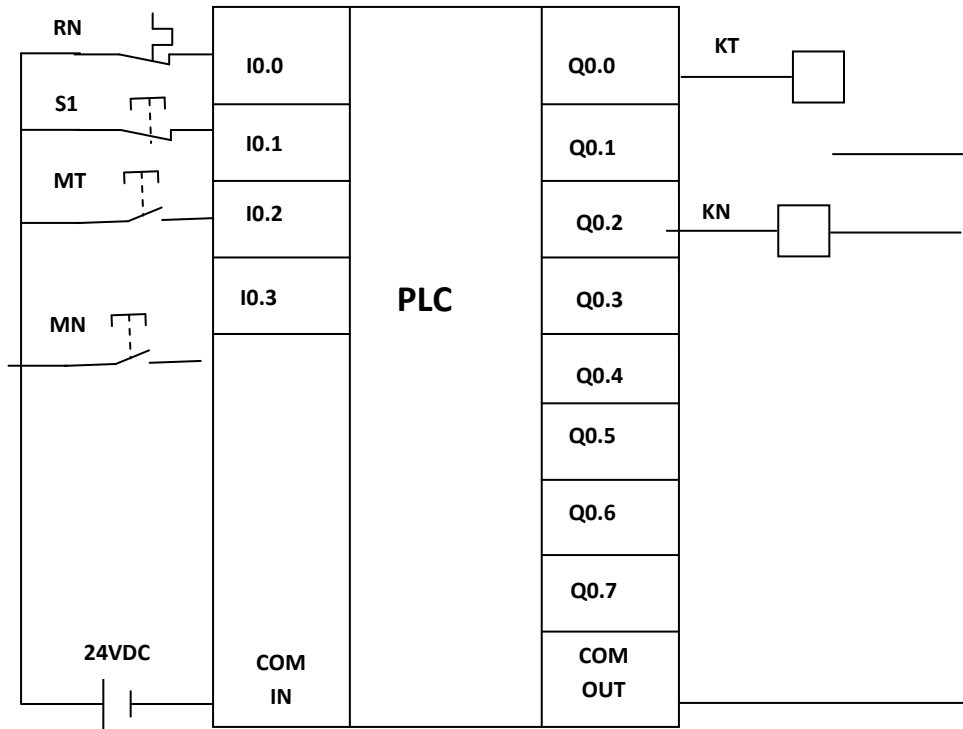


Hình 3-1 : Sơ đồ mạch động lực

2.2.2. Lập bảng địa chỉ ngõ vào - ngõ ra.

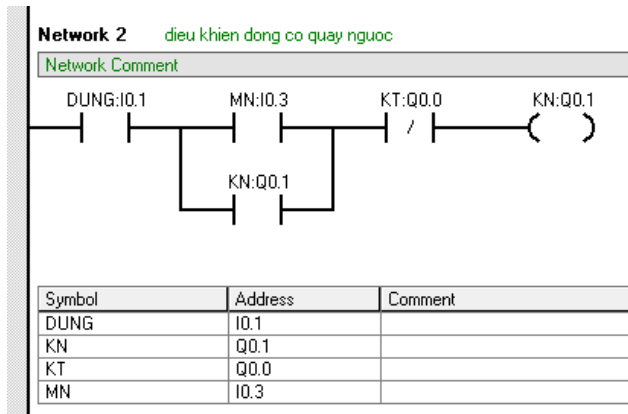
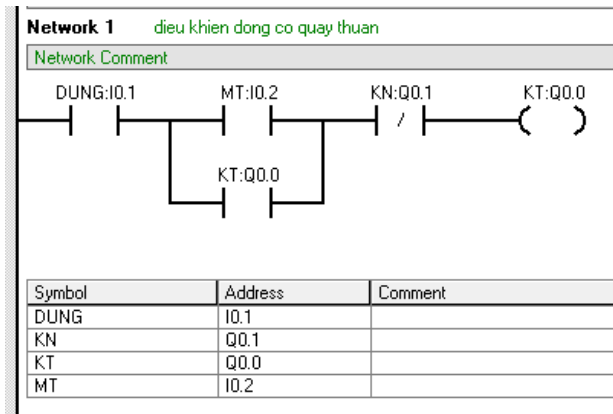
Kí hiệu	Địa chỉ	Mô tả
RN	I0.0	Bảo vệ quá tải
S1	I0.1	Dừng
MT	I0.2	Điều khiển quay thuận
MN	I0.3	Điều khiển quay ngược
KT	Q0.0	Khởi động từ khống chế động cơ quay thuận
KN	Q0.1	Khởi động từ khống chế động cơ quay ngược

2.2.3.Sơ đồ kết nối PLC với thiết bị ngoại vi



Hình 3-2 :Sơ đồ kết nối PLC

2.2.4.Chương trình



3.Bài tập

Bài tập 1:

- Nhấn MT động cơ khởi động và làm việc quay theo chiều thuận.
- Nhấn MN động cơ khởi động và làm việc quay theo chiều ngược.

- Nhấn D động cơ ngừng
- Mạch có bảo vệ quá tải

Bài tập 2:

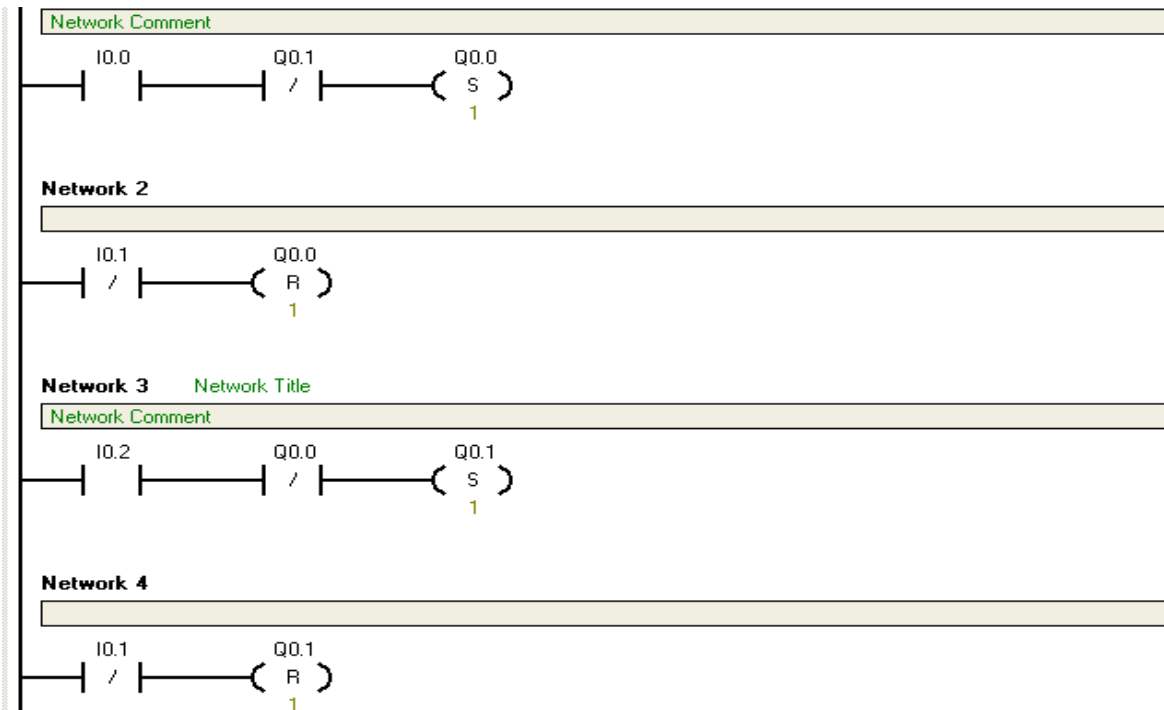
Mạch chốt lẫn nhau bằng hai van từ

Qua việc khởi động S1 hoặc S3 các bộ nhớ 1(van từ 1) hoặc bộ nhớ 2(van từ 2) sẽ được đặt. Nút nhấn s2 làm nhiệm vụ cắt mạch.

- Lập bảng địa chỉ ngõ vào - ngõ ra.

ký hiệu	Địa chỉ (toán hạng)	Giải thích (mô tả)
S1	I0.0	Nút nhấn thường hở
S2	I0.1	Nút nhấn thường đóng
S3	I0.3	Nút nhấn thường hở
Y1	Q0.0	Van từ 1
Y2	Q0.1	Van từ 2

Chương trình (LAD):



Bài 3: Mạch tuần tự cường bức báo lỗi

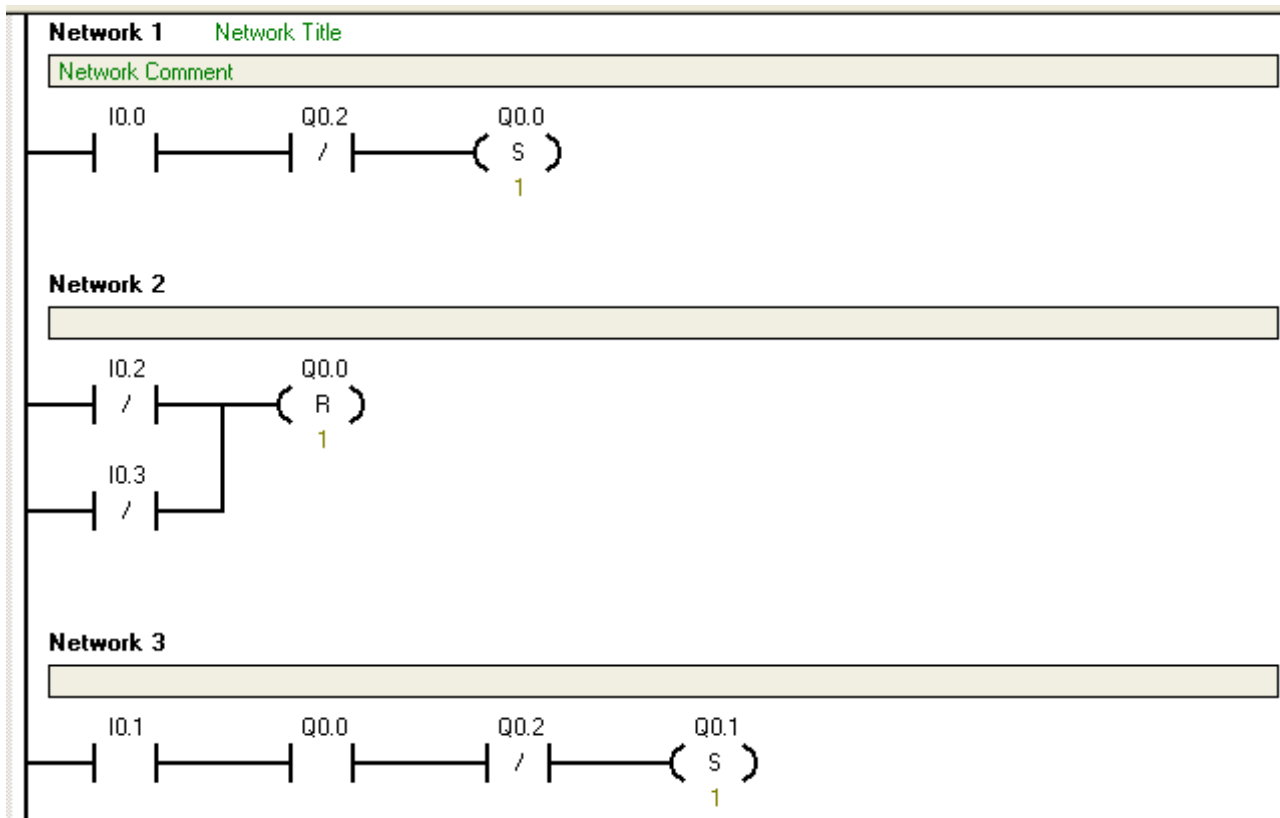
Qua việc khởi động nút nhấn S1 (I0.0) ngõ ra Q0.0 sẽ được đặt. Khi nhấn S2 (I0.1) ngõ ra Q0.1 cũng được đặt. Bằng nút nhấn S3 cả hai ngõ ra sẽ được đặt ngược lại. Khi có sự cố thì cả hai bộ nhớ Q0.0 và Q0.1 cũng được đặt ngược lại qua nút nhấn thường đóng S4

(I0.3). Nút nhấn thường đóng S5(I0.4) để phục hồi mạch, khi đó quá trình mới có thể bắt đầu.

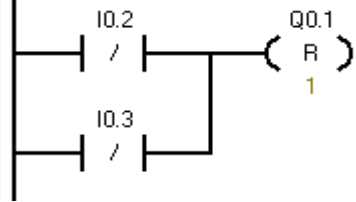
- Lập bảng địa chỉ ngõ vào - ngõ ra.

ký hiệu	địa chỉ (toán hạng)	Giải thích (mô tả)
S1	I0.0	Nút nhấn thường hở
S2	I0.1	Nút nhấn thường hở
S3	I0.2	Nút nhấn thường đóng
S4	I0.3	Nút nhấn thường đóng
S5	I0.4	Nút nhấn thường đóng
K1	Q0.0	Khởi động từ
K2	Q0.1	Khởi động từ
H1	Q0.2	Đèn báo

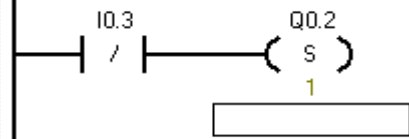
-Chương trình(LAD)



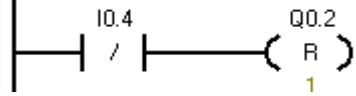
Network 4



Network 5



Network 6



BÀI 4:

ĐIỀU KHIỂN HỆ THỐNG ĐỘNG CƠ KHỞI ĐỘNG TUẦN TỰ .

Mục tiêu:

- Trình bày được các chức năng của Timer
- Ứng dụng linh hoạt các chức năng của Timer trong các bài toán thực tế
- Lập trình, kết nối, vận hành.
- Giải quyết các công việc một cách hệ thống theo nhóm.

Nội dung chính:

1. Timer (Bộ định thời)

- Timer là bộ tạo thời gian trễ giữa tín hiệu vào và tín hiệu ra nên trong điều khiển vẫn thường gọi là khâu trễ.

- S7-200 có 128 Timer (CPU-214; CPU-224) được chia làm 2 loại khác nhau,

đó là:

- Timer tạo thời gian trễ không có nhớ

+ On delay Timer - ký hiệu là TON: Timer mở chậm không có nhớ

+ Off delay Timer - ký hiệu là TOF: Timer đóng chậm không có nhớ

- Timer tạo thời gian trễ có nhớ (Timer on delay retentive), ký hiệu là TONR.

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

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

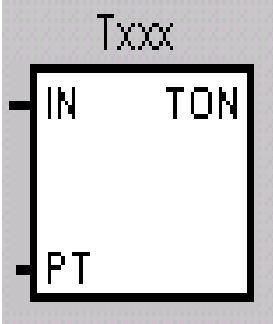
- Khi đầu vào có giá trị logic bằng 0, TON tự động reset còn TONR thì không. Timer TON được dùng để tạo thời gian trễ trong một khoảng thời gian (miền liên thông), còn với TONR thời gian trễ sẽ được tạo ra trong nhiều khoảng thời gian khác nhau.

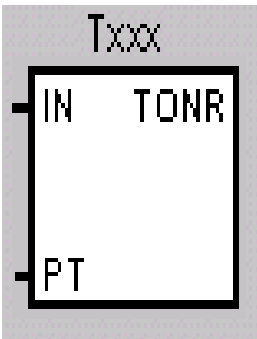
- Timer TON và TONR bao gồm ba loại với ba độ phân giải khác nhau, độ phân giải 1ms, 10ms, và 100ms. *Thời gian trễ được tạo ra chính là tích của độ phân giải của bộ timer được chọn và giá trị đặt trước cho timer.*

- Các loại Timer của S7-200 (đối với CPU-214; CPU-224) chia theo TON, TONR và độ phân giải bao gồm:

Lệnh	Độ phân giải	Giá trị cực đại	CPU 224
TON	1 ms	32,767s	T32 và T96
	10 ms	327,67s	T33÷T36; T97÷ T100
	100 ms	3276,7s	T37÷T63; T101÷ T127
TONR	1 ms	32,767s	T0 và T64
	10 ms	327,67s	T1÷T4; T65÷T68
	100 ms	3276,7s	T5÷T31; T69÷T95

-Cú pháp khai báo sử dụng Timer trong LAD như sau:

LAD	Mô tả	Toán hạng
	<p>Khai báo Timer số hiệu xxx kiểu TON để tạo thời gian trễ tính từ khi đầu vào IN được kích. Nếu như giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước PT thì T-bit có giá trị logic bằng 1. Có thể reset Timer kiểu TON bằng lệnh R hoặc bằng giá trị logic 0 tại đầu vào IN</p>	<p>Txxx (word) CPU 214;CPU 224: T32 ÷ T63 và T96 ÷ T127 PT: VW, T, (word), C, IW, QW, MW, SMW, AC, VD, *AC, AIW, Const.</p>

LAD	Mô tả	Toán hạng
	<p>Khai báo Timer số hiệu xxx kiểu TONR để tạo thời gian trễ tính từ khi đầu vào IN được kích. Nếu như giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước PT thì T-bit có giá trị logic bằng 1. Chỉ có thể reset kiểu TONR bằng lệnh R cho T-bit</p>	<p>Txxx (word) CPU 214;CPU 224: T0 ÷ T31 và T64 ÷ T95 PT: VW, T, (word), C, IW, QW, MW, SMW, AC, VD, *AC, AIW, Const.</p>

Khi sử dụng timer kiểu TONR, giá trị đếm tức thời được lưu lại và không bị thay đổi trong khoảng thời gian khi tín hiệu đầu vào có giá trị logic 0. giá trị của T-bit không được nhớ mà hoàn toàn phụ thuộc vào kết quả so sánh giữa giá trị đếm tức thời và giá trị đặt trước.

Khi reset một bộ timer, T-word và T-bit của nó đồng thời được xoá và có giá trị bằng 0, như vậy giá trị đếm tức thời được đặt về 0 và tín hiệu đầu ra cũng có trạng thái logic bằng 0.

2. Điều khiển 4 động cơ không đồng bộ 3 pha khởi động tuần tự.

2.1. Yêu cầu công nghệ:

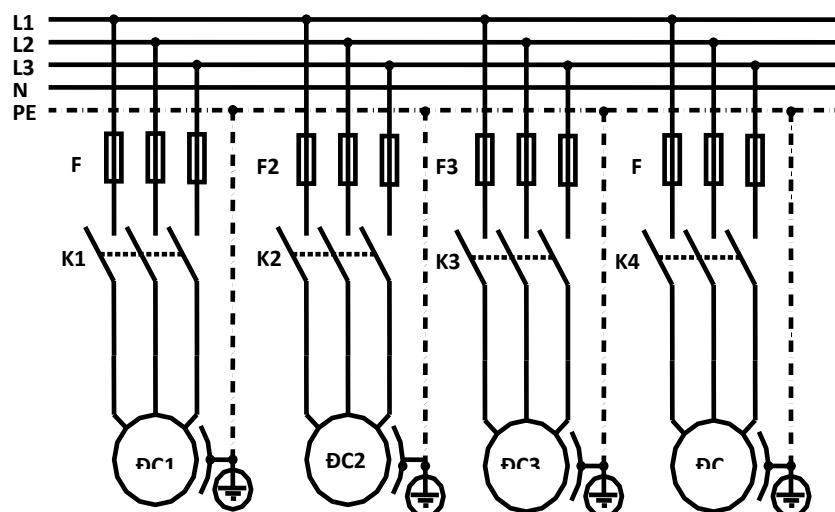
Khi nhấn START: Hệ thống động cơ hoạt động từ động cơ cuối dây chuyền đến động cơ đầu dây chuyền lần lượt cách nhau 5 giây.

Khi nhấn nút STOP: Hệ thống động cơ

2.2. Nhiệm vụ:

- Vẽ mạch động lực.
- Lập bảng xác lập ngõ vào/ra.
- Vẽ sơ đồ nối dây PLC.
- Viết chương trình PLC S7-200 theo ngôn ngữ LAD trên phần mềm STEP7-Microwin V3.2 hoặc V4.0.
- Kết nối thiết bị ngoại vi, download, vận hành chương trình trên S7-200, CPU 224.

2.2.1. Mạch động lực:

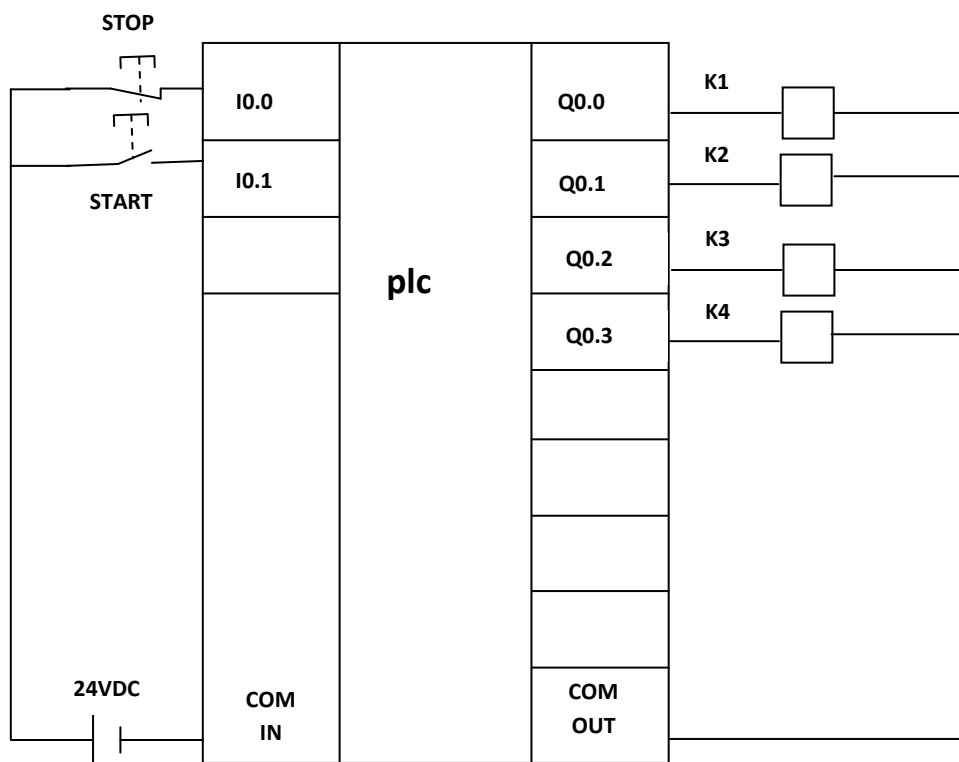


Hình 4-1 :Sơ đồ mạch động lực

2.2.2. Lập bảng địa chỉ ngõ vào - ngõ ra.

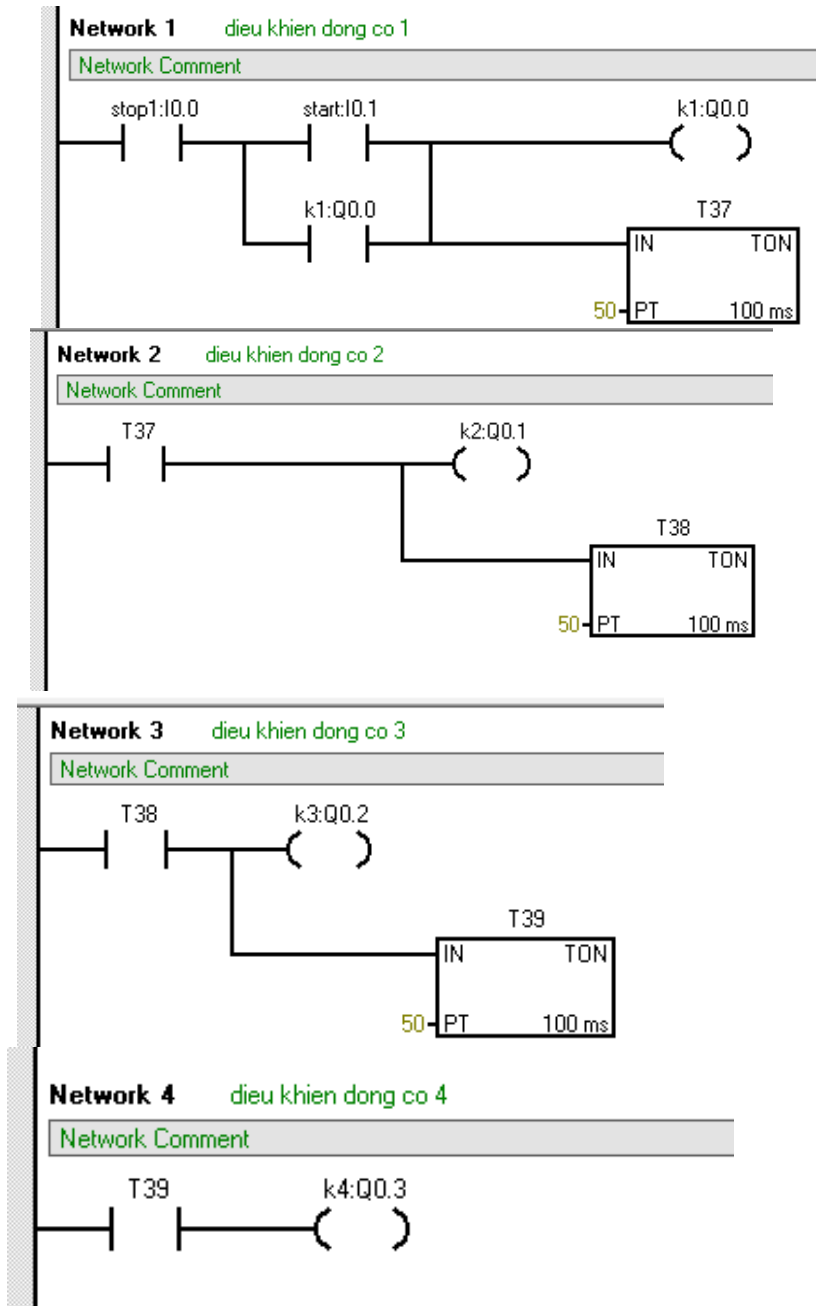
Ký hiệu	Toán hạng	Mô tả
STOP	I0.0	Dừng hệ thống
START	I0.1	Khởi động hệ thống
K1	Q0.0	Contactơ không chế động cơ 1
K2	Q0.1	Contactơ không chế động cơ 2
K3	Q0.2	Contactơ không chế động cơ 3
K4	Q0.3	Contactơ không chế động cơ 4

2.2.3. Sơ đồ kết nối PLC với thiết bị ngoại vi



Hình 4-2 :Sơ đồ kết nối PLC

2.2.4. Chương trình:



3. Bài tập ứng dụng Timer

Bài 1 : Khi nhấn START: Hệ thống động cơ hoạt động từ động cơ cuối dây chuyền đến động cơ đầu dây chuyền lần lượt cách nhau 05 giây.

Khi nhấn nút STOP: Hệ thống động cơ dừng từ động cơ đầu dây chuyền đến động cơ cuối dây chuyền lần lượt cách nhau 10 giây.

Bài 2 : Khi nhấn START: Hệ thống động cơ hoạt động từ động cơ cuối dây chuyền đến động cơ đầu dây chuyền lần lượt cách nhau 05 giây.

Khi nhấn nút STOP: Hệ thống động cơ dừng từ động cơ đầu dây chuyền đến động cơ cuối dây chuyền lần lượt cách nhau 10 giây.

Khi có sự cố quá tải của động cơ nào thì đèn của động cơ đó sáng.

BÀI 5

ĐIỀU KHIỂN DÂY CHUYỀN ĐÓNG GÓI SẢN PHẨM

Mục tiêu:

- Trình bày được các chức năng của counter.
- Ứng dụng linh hoạt các chức năng của counter trong các bài toán thực tế
- Lập trình, kết nối, vận hành.
- Giải quyết các công việc một cách hệ thống theo nhóm.

Nội dung chính :

1. Counter (Bộ đếm)

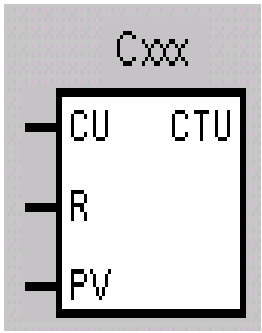
Counter là bộ đếm thực hiện chức năng đếm sườn xung trong S7-200. Các bộ đếm của S7-200 được chia ra làm 2 loại: Bộ đếm tiến (CTU) và Bộ đếm tiến/ lùi (CTUD).

1.1. Bộ đếm tiến (CTU):

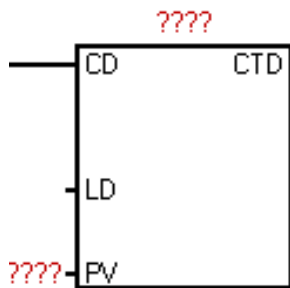
Bộ đếm tiến đếm số sườn lên của tín hiệu logic đầu vào, tức là đếm số lần thay đổi trạng thái logic từ 0 lên 1 của tín hiệu. Số xung đếm được ghi vào thanh ghi 2byte của bộ đếm, gọi là thanh ghi C-word.

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

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

LAD	Mô tả	Toán hạng
	<p>Khai báo bộ đếm tiến theo sườn lên của CU. Khi giá trị đếm tức thời C-word Cxxx lớn hơn hoặc bằng giá trị đặt trước PV, C-bit (cxxx) có giá trị logic bằng 1. Bộ đếm được reset khi đầu vào R có giá trị logic bằng 1. Bộ đếm ngừng đếm khi C-word Cxx đạt giá trị cực đại (32.767).</p>	<p>Cxxx (word) CPU 224: C0 ÷ C47 và C80 ÷ C127 PV(word): VW, T, C, IW, QW, MW, SMW, AC, VD, *AC, AIW, Const.</p>

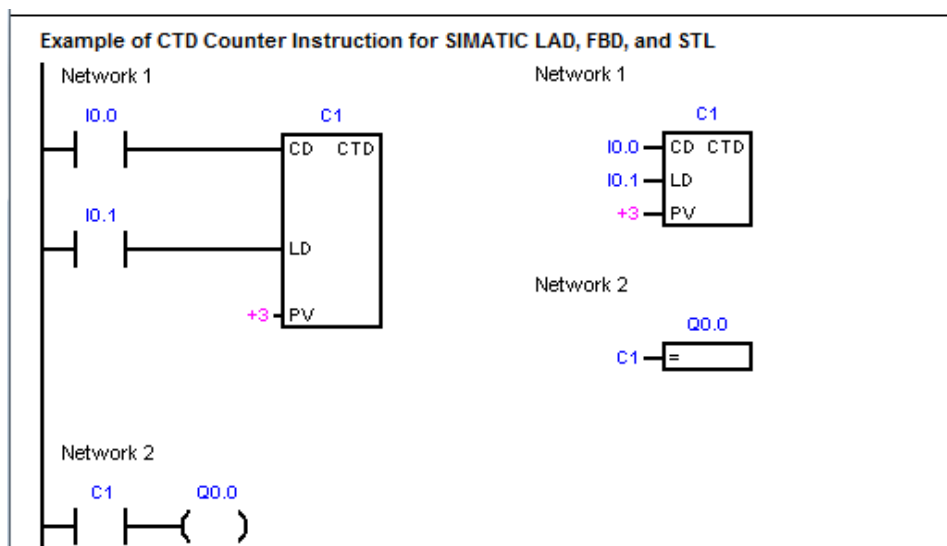
1.2. Bộ đếm xuống (CTD):



– Khi tín hiệu preset ở chn LD (CTD) thì giá trị đếm của Cxxx được set về giá trị đặt PV .

CD : ngõ vào đếm xuống

– Đối với bộ đếm xuống CTD: Khi giá trị đếm của Cxxx = 0 thì trạng thái Cxxx là ON (1), ngược lại OFF(0).



1.3. Bộ đếm tiến / lùi (CTUD):

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

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

Khi đầu vào logic của chân xóa, ký hiệu bằng R trong LAD hoặc bit thứ nhất của ngăn xếp trong STL, có giá trị logic là 1 hoặc bằng lệnh R (reset) với C-bit của bộ đếm.

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

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

Các bộ đếm được đánh số từ 0 đến 127 (đối với CPU 224) và ký hiệu bằng Cxxx, trong đó xxx là số thứ tự của bộ đếm. Ký hiệu Cxxx đồng thời cũng là địa chỉ hình thức của C-word và của C-bit. Mặc dù dùng địa chỉ hình thức, song C-word và C-bit vẫn được phân biệt với nhau nhờ kiểu lệnh sử dụng làm việc với từ hay với tiếp điểm (bit).

Lệnh khai báo sử dụng bộ đếm trong LAD như sau:

LAD	Mô tả	Toán hạng
	<p>Khai báo bộ đếm tiến/lùi, đếm tiến theo sườn lên của CU và đếm lùi theo sườn lên của CD. Khi giá trị đếm tức thời C-word lớn hơn hoặc bằng giá trị đặt trước PV, C-bit (Cxxx) có giá trị logic bằng 1. Bộ đếm ngừng đếm tiến khi C-word đạt giá trị cực đại (32.767) và ngừng đếm lùi khi C-word đạt giá trị cực tiểu (-32.767) CTUD reset khi đầu vào R có giá trị logic bằng 1.</p>	<p>Cxxx (word) CPU 224: C48 ÷ C79 PV(word): VW, T, C, IW, QW, MW, SMW, AC, VD, *AC, AIW, Const.</p>

2. Điều khiển dy chuyên đóng gói sản phẩm.

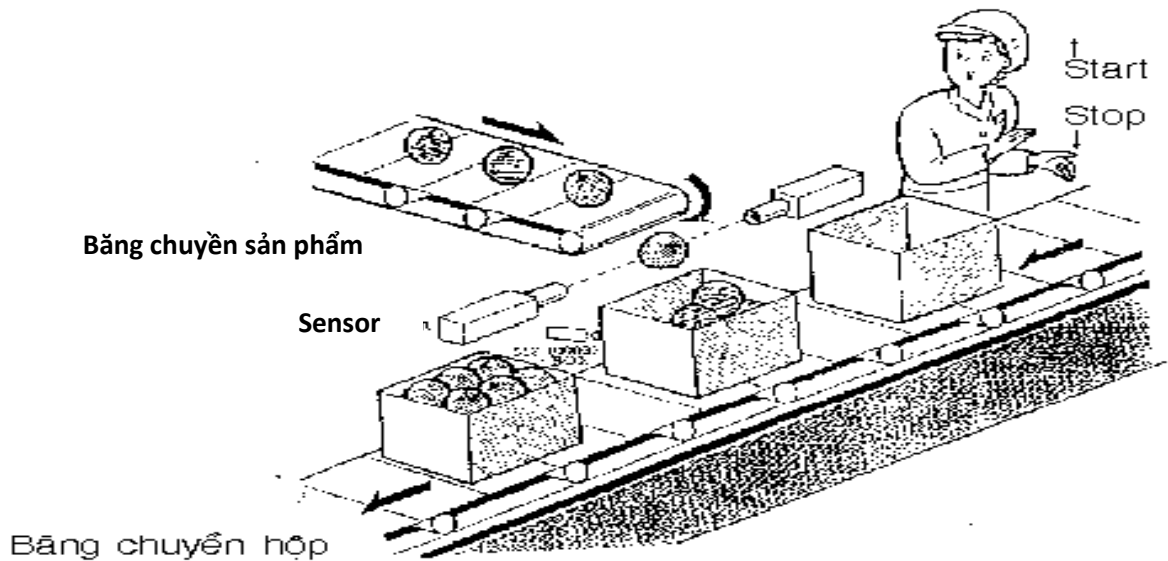
2.1. Yêu cầu công nghệ:

Lập trình PLC điều khiển kiểm soát dây chuyền đóng hộp, với yêu cầu sau:

- Khi nhấn *START* dây chuyền hộp vận hành. Khi cảm biến hộp phát hiện thì dây chuyền hộp dừng lại, dây chuyền sản phẩm bắt đầu hoạt động. Khi cảm biến sản phẩm phát hiện thì bộ đếm tăng lên 1. Khi đếm đủ 10 sản phẩm thì dây chuyền sản phẩm dừng và dây chuyền hộp lại bắt đầu chuyển động.

Khi nhấn *STOP* hệ thống dừng.

Sơ đồ công nghệ, bảng trạng thái như sau:

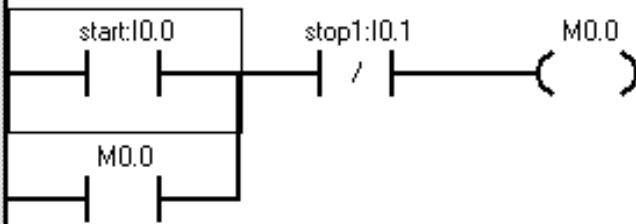


2.2. Lập bảng địa chỉ ngõ vào - ngõ ra.

Ký hiệu	Địa chỉ	Mô tả
<i>S0</i>	<i>I0.0</i>	Nút khởi động <i>START</i>
<i>S1</i>	<i>I0.1</i>	Nút dừng <i>STOP</i>
<i>S2</i>	<i>I0.2</i>	Cảm biến hộp
<i>S3</i>	<i>I0.3</i>	Cảm biến sản phẩm
<i>K1</i>	<i>Q0.1</i>	Động cơ băng chuyền thùng
<i>K2</i>	<i>Q0.0</i>	Động cơ băng chuyền sản phẩm

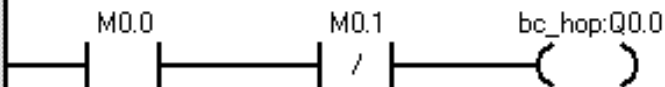
Network 1 khởi động hệ thống

Network Comment

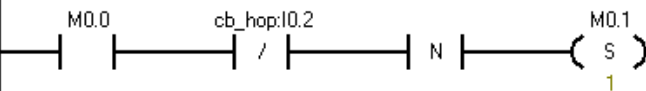


Symbol	Address	Comment
start	I0.0	
stop1	I0.1	

Network 2 bảng chuyển hop hoạt động

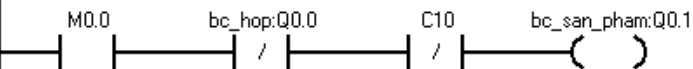


Network 3 cảm biến hop phát hiện hop phát ra 1 xung để dùng bảng chuyển hop

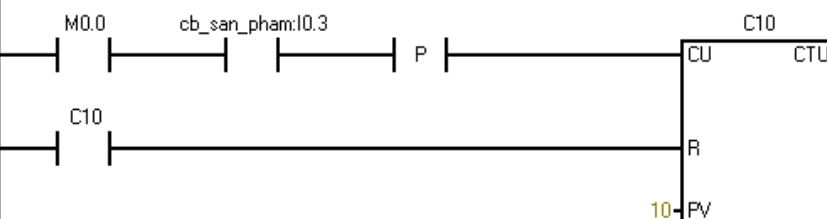


Symbol	Address	Comment
cb_hop	I0.2	

Network 4 bảng chuyển sản phẩm hoạt động



Network 5 đếm sản phẩm



3. Bài tập:

Bài tập 1:

Lập trình PLC điều khiển động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc làm việc theo yêu cầu sau:

Chạy thuận (nghịch) 7giây.

Ngừng 3 giây.

Chạy nghịch (thuận) 7giây.

Quá trình trên lặp lại 3 lần.

Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

Bài tập 2:

Lập trình PLC điều khiển hai động cơ làm việc theo yêu cầu sau:

Động cơ Đ1 làm việc 5 giây rồi ngừng, sau đó đến động cơ Đ2 tự động làm việc 5 giây rồi ngừng 5 giây. Động cơ Đ2 lặp lại 3 lần như vậy, kể đến chu kỳ làm việc của hai động cơ lặp lại 5 lần rồi nghỉ.

Muốn làm việc nữa thì khởi động lại.

Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

BÀI 6

ĐIỀU KHIỂN TRẠM KHÍ NÉN CÓ HAI XY LANH

Mục tiêu :

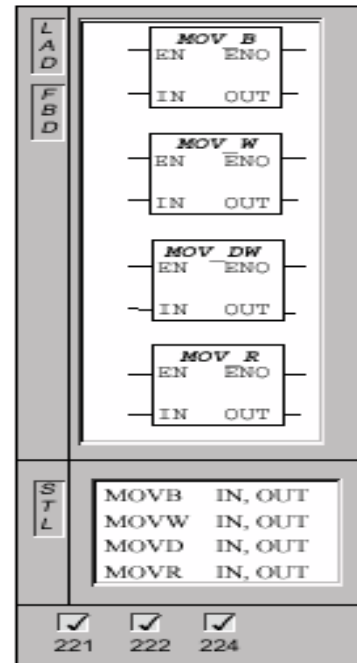
- Ứng dụng chúng trong các bài toán thực tế
- Lập trình, kết nối, vận hành.
- Giải quyết các công việc một cách hệ thống theo nhóm.

Nội dung chính :

1. Chức năng truyền dẫn

1.1. Truyền dẫn Byte; Word; Doubleword

- Phép truyền **Move Byte** sẽ thực hiện copy dữ liệu Byte tại ngõ vào IN và truyền tới Byte tại ngõ ra OUT.
- Phép truyền **Move Word** sẽ thực hiện copy dữ liệu Word tại ngõ vào IN và truyền tới Word tại ngõ ra OUT.
- Phép truyền **Move DoubleWord** sẽ thực hiện copy dữ liệu doubleword tại ngõ vào IN và truyền tới doubleWord tại ngõ ra OUT.
- Phép truyền **Move Real** sẽ thực hiện copy một số thực 32 bit tại Double Word ngõ vào IN và truyền tới doubleWord tại ngõ ra OUT.



- Khi xảy ra lỗi thì ngõ ENO bị SET = 0

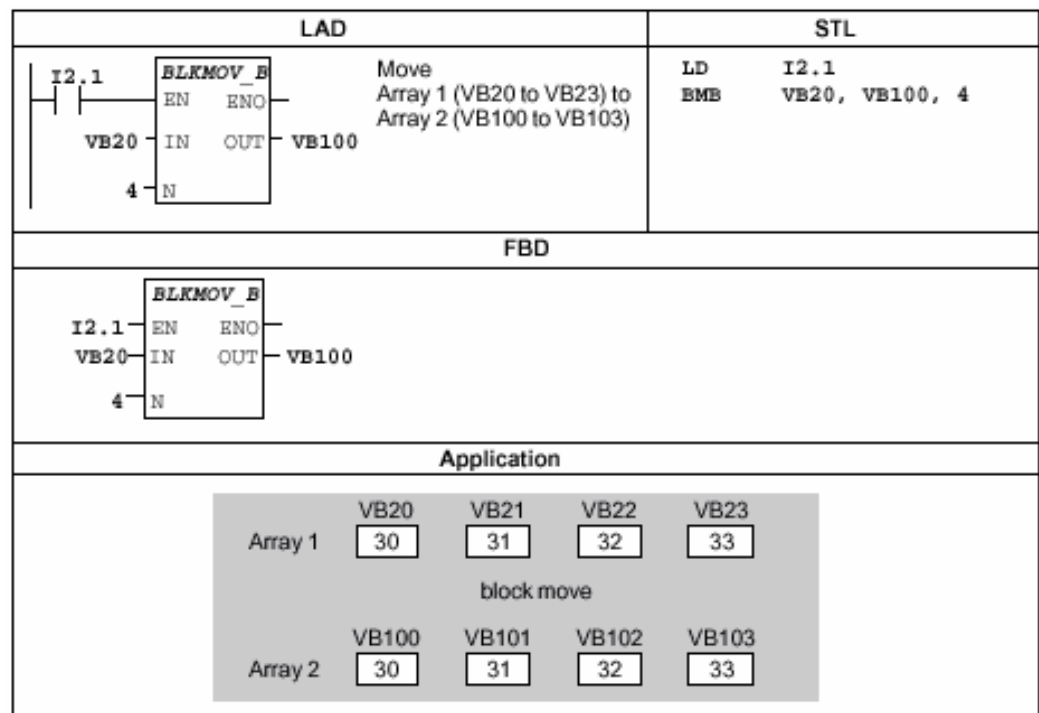
Move...	Inputs/Outputs	Operands	Data Types
Byte	IN	VB, IB, QB, MB, SB, SMB, LB, AC, Constant, *VD, *AC, *LD	BYTE
	OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
Word	IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, Constant, AC *VD, *AC, *LD	WORD, INT
	OUT	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	WORD, INT
Double Word	IN	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, AC, Constant, *VD, *AC, *LD	DWORD, DINT
	OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DWORD, DINT
Real	IN	VD, ID, QD, MD, SD, SMD, LD, AC, Constant, *VD, *AC, *LD	REAL
	OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	REAL

1.2. Truyền một vùng nhớ dữ liệu

- Phép truyền Block Move Byte, Block Move word, Block Move Doubleword sẽ thực hiện truyền một số lượng Byte (N) có địa chỉ Byte đầu tại ngõ vào IN sang vùng nhớ có địa chỉ đầu tại ngõ ra OUT.

N là số lượng Byte có giới hạn từ 1 đến 255.

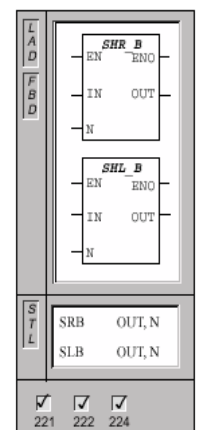
Ví dụ về truyền một mảng dữ liệu BLKMOV: Trong ví dụ này một mảng dữ liệu thứ nhất gồm 4 Byte (N= 4) thuộc vùng nhớ V có địa chỉ đầu từ VB0 được truyền đến một vùng nhớ V có địa chỉ đầu từ VB 100 (mảng 2). Dữ liệu tại mảng 1 vẫn không đổi.



2. Chức năng dịch chuyển

2.1. Dịch phải Byte SHR_B và Dịch trái Byte SHL_B:

Các lệnh SHR_B và SHL_B sẽ dịch dữ liệu tại Byte ngõ vào IN sang phải hoặc sang trái với số vị trí dịch được nhập tại N, kết quả được chứa vào Byte ngõ ra OUT. Ở các lệnh SHIFT thì tại vị trí các Bit bị dịch sẽ lấp đầy bằng số 0. Số vị trí Bit cần dịch được nhập tại ngõ N <=8. Việc ảnh hưởng đến các Bit nhớ đặc biệt (SM1.0 và SM1.1) xin xem thêm trong sổ tay.



Inputs/Outputs	Operands	Data Types
IN, OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
N		

Bảng giới hạn vùng toán hạng và dạng dữ liệu

2.2. Dịch phải Word SHR_W và Dịch trái Word SHL_W:

Các lệnh SHR_W và SHL_W sẽ dịch dữ liệu tại Word ngõ vào IN sang phải hoặc sang trái với số vị trí dịch được nhập tại N, kết quả được chứa vào Word có địa chỉ tại ngõ ra OUT. Tại vị trí các Bit bị dịch sẽ lấp đầy bằng số 0. Số vị trí Bit cần dịch được nhập tại ngõ $N \leq 16$. Việc ảnh hưởng đến các Bit nhớ đặc biệt (SM1.0 và SM1.1) xin xem thêm trong sổ tay.

Inputs/Outputs	Operands	Data Types
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, Constant, *VD, *AC, *LD	WORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, Constant, *VD, *AC, *LD	BYTE
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp

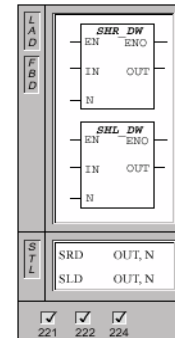
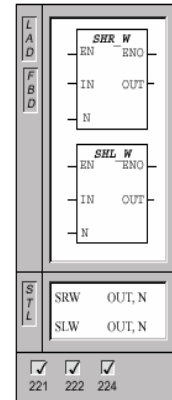
2.3. Dịch phải Doubleword SHR_DW và Dịch trái SHL_DW:

Dịch phải DoubleWord SHR_DW và Dịch trái Word SHL_W:

Các lệnh SHR_DW và SHL_DW sẽ dịch dữ liệu tại DoubleWord ngõ vào IN sang phải hoặc sang trái với số vị trí dịch được nhập tại N, kết quả được chứa vào DoubleWord có địa chỉ tại ngõ ra OUT. Tại vị trí các Bit bị dịch sẽ được lấp đầy bằng số 0. Số vị trí Bit cần dịch được nhập tại ngõ $N \leq 32$. Việc ảnh hưởng đến các Bit nhớ đặc biệt (SM1.0 và SM1.1) xin xem thêm trong sổ tay.

Inputs/Outputs	Operands	Data Types
IN	VD, ID, QD, MD, SD, SMD, LD, AC, HC, Constant, *VD, *AC, *LD	DWORD
N	VB, IB, QB, MB, SB, SMB, LB, AC, Constant, *VD, *AC, *LD	BYTE
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DWORD

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp



3. Chức năng so sánh

Khi lập trình, nếu có các quyết định về điều khiển được thực hiện dựa trên kết quả của việc so sánh thì có thể sử dụng lệnh so sánh cho byte, từ hay từ kép của **S7-200**.

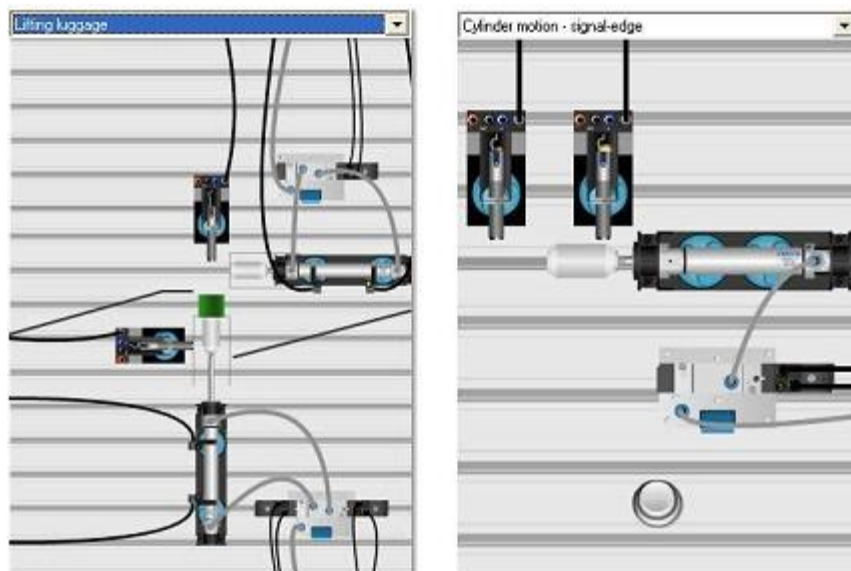
LAD sử dụng lệnh so sánh để so sánh các giá trị của byte, từ và từ kép (giá trị thực hoặc nguyên). Những lệnh so sánh thường là so sánh nhỏ hơn hoặc bằng (<=); so sánh bằng (=) và so sánh lớn hơn hoặc bằng (>=).

Biểu diễn các lệnh so sánh trong LAD:

LAD	Mô tả	Toán hạng
$n1 \text{ --- } \text{==} \begin{matrix} \text{B} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{==} \begin{matrix} \text{I} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{==} \begin{matrix} \text{D} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{==} \begin{matrix} \text{R} \\ \\ \text{---} \end{matrix} \text{---} n2$	<p>Tiếp điểm đóng khi n1=n2</p> <p>B = byte I = Integer = Word D = Double Integer R = Real</p>	<p>n1. n2(byte): VB, IB, QB, MB, SMB, AC, Const, *VD, *AC</p>
$n1 \text{ --- } \text{>} \begin{matrix} \text{B} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{>} \begin{matrix} \text{I} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{>} \begin{matrix} \text{D} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{>} \begin{matrix} \text{R} \\ \\ \text{---} \end{matrix} \text{---} n2$	<p>Tiếp điểm đóng khi n1>n2</p> <p>B = byte I = Integer = Word D = Double Integer R = Real</p>	<p>n1. n2(Word): VW, T, QW, MW, SMW, AC, AIW, Const, *VD, *AC</p>
$n1 \text{ --- } \text{<} \begin{matrix} \text{B} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{<} \begin{matrix} \text{I} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{<} \begin{matrix} \text{D} \\ \\ \text{---} \end{matrix} \text{---} n2$ $n1 \text{ --- } \text{<} \begin{matrix} \text{R} \\ \\ \text{---} \end{matrix} \text{---} n2$	<p>Tiếp điểm đóng khi n1<n2</p> <p>B = byte I = Integer = Word D = Double Integer R = Real</p>	<p>n1. n2(Dword): VD, ID, QD, MD, SMD, AC, HC, Const, *VD, *AC</p>

4. Yêu cầu

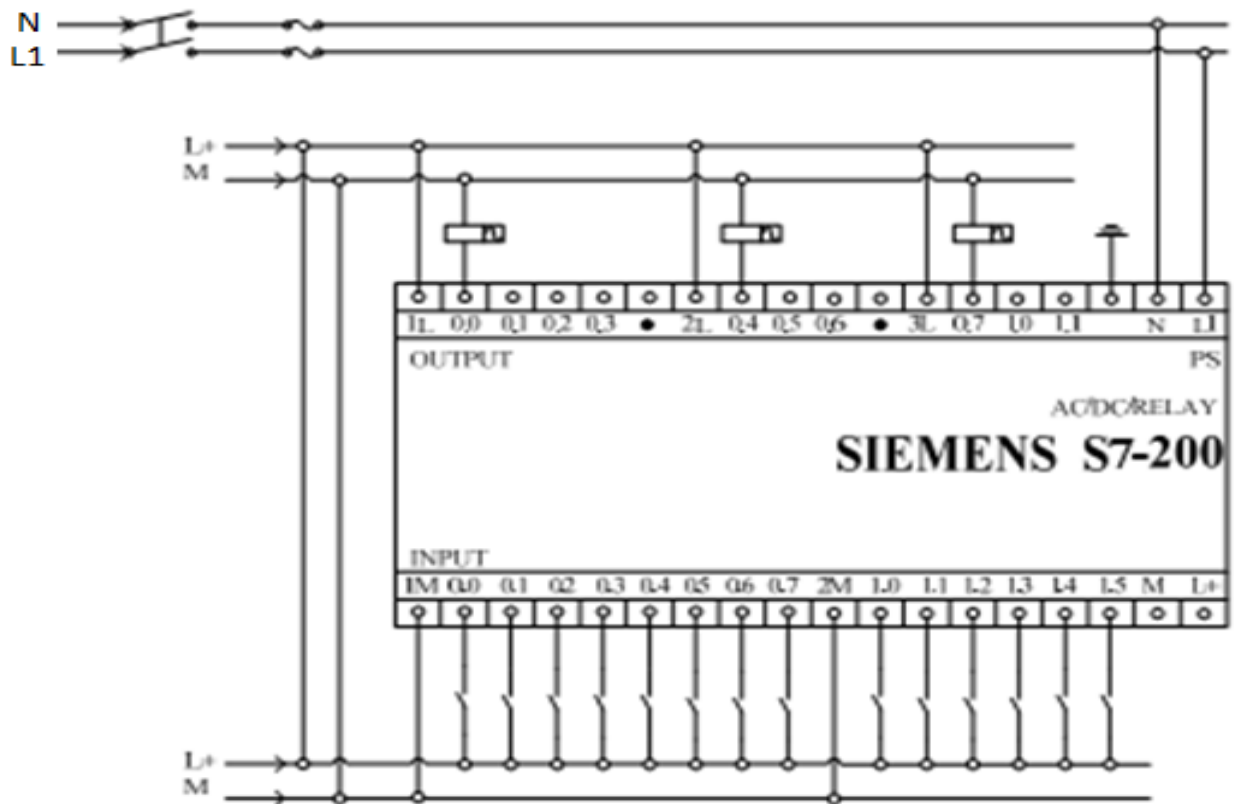
Hệ thống gồm các xylanh được thiết kế ở hình bên.



Hình 5-1 :Sơ đồ bố trí thiết bị

1. Khi gạt công tắc thì xylanh chạy tới, khi gạt công tắc trở lại thì xylanh tự rút về vị trí ban đầu.
2. Khi nhấn nút Start thì xylanh chạy tới, khi nhấn nút Stop thì xylanh tự rút về vị trí ban đầu.
3. Nhấn nút PB1 thì xylanh chạy ra, gập cảm biến ngoài thì xylanh tự dừng.
Nhấn nút PB2 thì xylanh chạy vào, gập cảm biến trong thì xylanh tự dừng.
4. Mỗi lần nhấn nút khởi động Start, xylanh chạy ra, gập hành trình ngoài thì xylanh tự rút về. Gập hành trình trong thì xylanh tự dừng. (Điều khiển xylanh tương tự như đảo chiều quay động cơ, dùng 2 ngõ ra.)
5. Khi nhấn nút Start thì xylanh chạy tới, sau 5 giây xylanh tự rút về.
6. Khi nhấn nút Start thì xylanh chạy tới, gập cảm biến Gh1 thì xylanh tự rút về, gập Gh2 thì chu kỳ mới tiếp tục. Xylanh chỉ dừng khi nhấn nút Stop.
7. Khi nhấn nút Start thì xylanh_1 chạy tới, gập cảm biến Gh1 thì xylanh_2 chạy tới, gập cảm biến Gh3 thì xylanh_2 chạy lùi, gập cảm biến Gh4 thì xylanh_1 chạy lùi.

5. Sơ đồ kết nối plc



Hình 5-2 : Sơ đồ kết nối plc

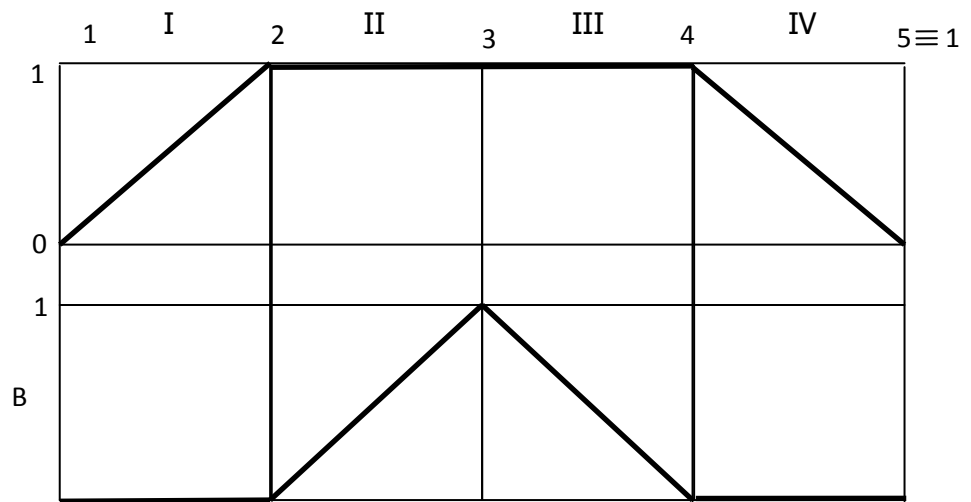
6. Các bước thực hiện:

1. Mở máy tính
2. Nối cáp PC/PPI vào PLC và máy tính
3. Cấp nguồn cho PLC
4. Mở phần mềm Step7-Microwin
5. Kiểm tra kết nối giữa PLC và máy tính
6. Viết chương trình sử dụng các lệnh cơ bản (thường hở, thường đóng, out)
7. Nạp chương trình
8. Chạy chương trình
9. Kiểm tra chương trình hoạt động
10. Lập lại bước 4 khi tiếp tục.

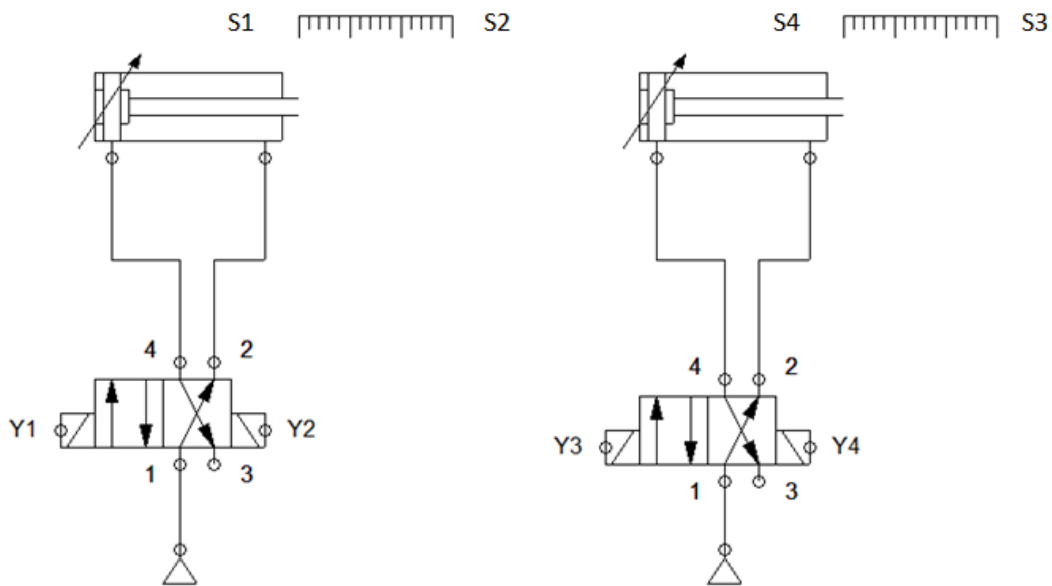
7. Bài tập áp dụng:

Lập trình plc điều khiển xy lanh theo sơ đồ sau:

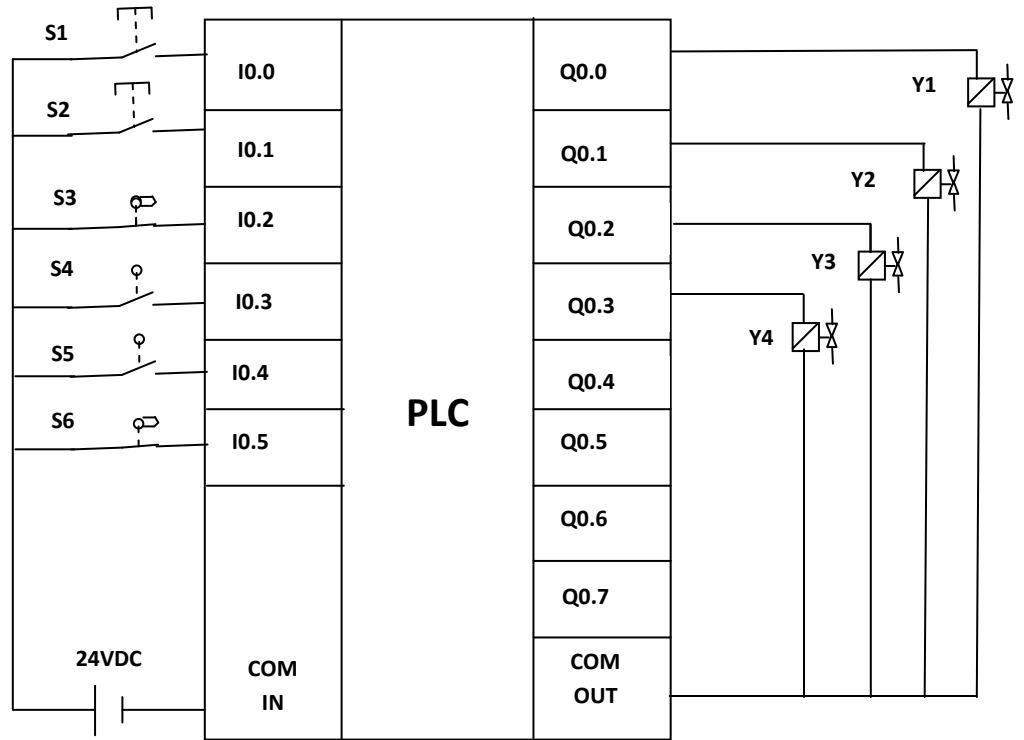
Hành trình bước:



Sơ đồ mạch động lực:



Sơ đồ kết nối plc với xy lanh



Hình 5-3 : Sơ đồ kết nối PLC

BÀI 7

ĐIỀU KHIỂN HỆ THỐNG ĐỘNG CƠ DÙNG HÀM CHƯƠNG TRÌNH CON.

Mục tiêu:

- Trình bày được cách khai báo chương trình con
- Ứng dụng chúng trong các bài toán thực tế
- Lập trình, kết nối, vận hành.
- Giải quyết các công việc một cách hệ thống theo nhóm.

Nội dung chính :

1. Lệnh nhảy và gọi chương trình con

Các lệnh của chương trình, nếu không có những lệnh điều khiển riêng, sẽ được thực hiện theo thứ tự từ trên xuống dưới trong một vòng quét. Lệnh điều khiển chương trình cho phép thay đổi thứ tự thực hiện lệnh. Chúng cho phép chuyển thứ tự thực hiện, đáng lẽ ra là lệnh tiếp theo, tới một lệnh bất kỳ nào khác của chương trình, trong đó nơi điều khiển chuyển đến phải được đánh dấu bằng một nhãn chỉ đích. Thuộc nhóm lệnh điều khiển chương trình gồm: lệnh nhảy; lệnh gọi chương trình con. Nhãn chỉ đích, hay gọi đơn giản là nhãn, phải được đánh dấu trước khi thực hiện lệnh nhảy hay lệnh gọi chương trình con.

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


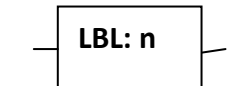
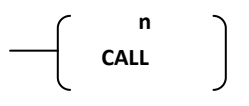
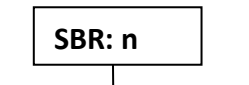


Lệnh gọi chương trình con là lệnh chuyển điều khiển đến chương trình con. khi chương trình con thực hiện xong các phép tính của mình thì việc điều khiển lại được chuyển về lệnh tiếp theo trong chương trình chính nằm ngay sau lệnh gọi chương trình con. Từ một chương trình con có thể gọi được một chương trình con khác trong nó, có thể gọi lặp lại nhiều nhất là 8 lần trong S7-200. Trong một chương trình con có lệnh gọi đến chính nó, về nguyên tắc không bị cấm song phải để ý đến giới hạn trên.

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

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

Nội dung của thanh ghi AC không được cất giữ khi gọi chương trình con, nhưng khi một chương trình xử lý ngắt được gọi, nội dung của thanh ghi AC sẽ được cất giữ trước khi thực hiện chương trình xử lý ngắt và nạp lại khi chương trình xử lý ngắt đã được thực hiện xong. Bởi vậy chương trình xử lý ngắt có thể tự do sử dụng 4 thanh ghi AC của S7-200

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

LAD	STL	Mô tả	Toán hạng
	JPM n	Lệnh nhảy thực hiện việc chuyển điều khiển đến nhãn n trong một chương trình	n: CPU 212: 0 ÷63 CPU 214: 0 ÷255 (224)
	LBL n	Lệnh khai báo nhãn n trong một chương trình	
	CALL n	Lệnh gọi chương trình con, thực hiện php chuyển điều khiển đến chương trình con chỉ nhn ln	n: CPU 212: 0 ÷15 CPU 214: 0 ÷255 (224)
	SBR n	Lệnh gñ nhn cho một chương trình con	
	CRET	Lệnh trở về chương trình đ gọi chương trình con cũ điều kiện (bit đầu của ngăn xếp cũ gi trị logic bằng 1).	Không có
	RET	Lệnh trở về chương trình đ gọi chương trình con không điều kiện	

2.Ví dụ tạo chương trình con:

Lập trình điều khiển 2 động cơ làm việc như sau:

Nhấn start 1 động cơ 1 làm việc sau 5s động cơ 2 làm việc

Nhấn stop 1 2 động cơ dừng

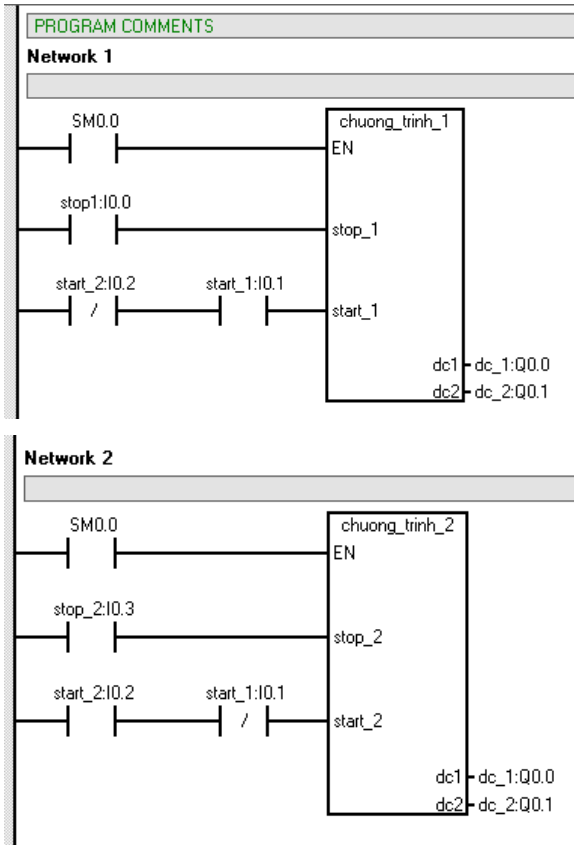
Lập trình điều khiển 2 động cơ làm việc như sau:

Nhấn start 2 động cơ 1 làm việc sau 10 s động cơ 2 làm việc

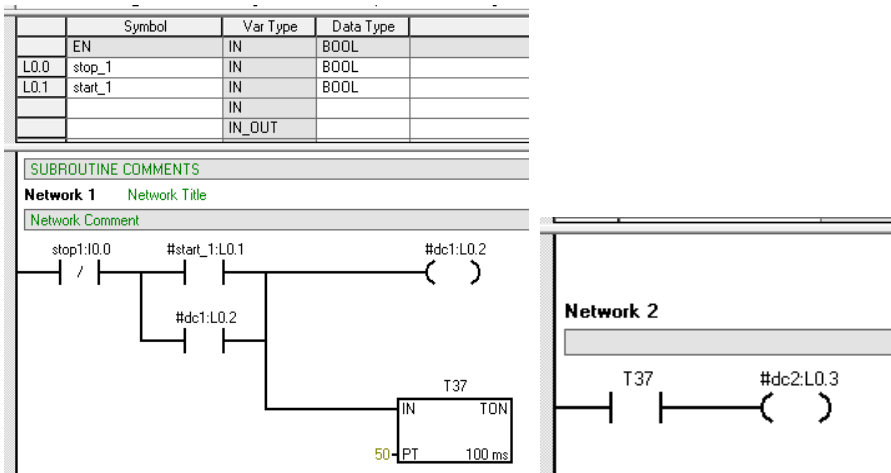
Nhấn stop 2 2 động cơ dừng

Yêu cầu tạo 2 chương trình con

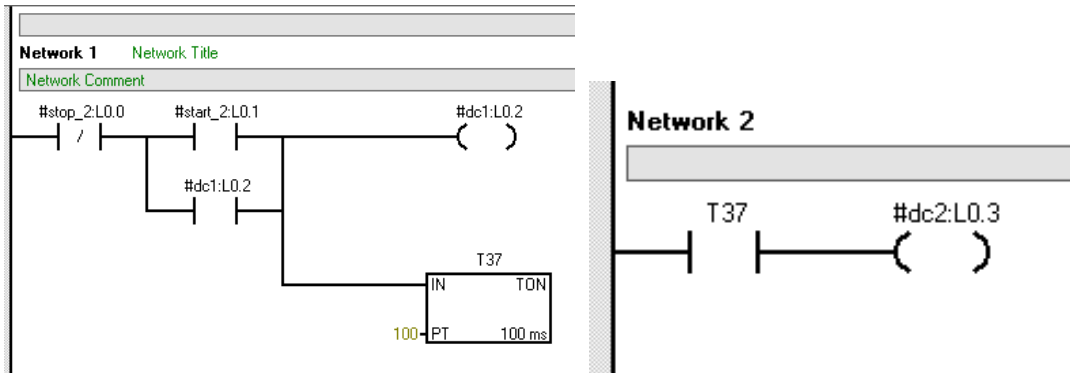
Chương trình chính (main)



Chương trình con 1:



Chương trình con 2:

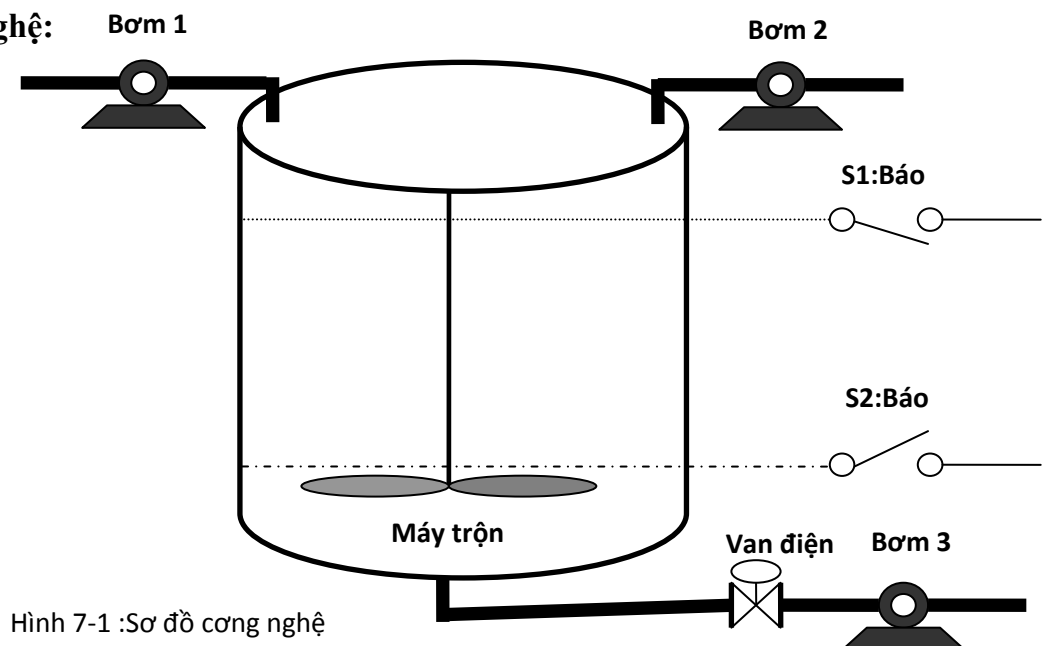


3. Bài tập:

3.1. Yêu cầu: Lập trình PLC điều khiển bồn trộn hóa chất từ hai loại hóa chất khác nhau, theo yêu cầu:

- Khi nhấn START thì bơm 1 và bơm 2 hoạt động để đưa hai loại hóa chất bồn chứa.
- Khi hóa chất trong bồn đầy thì bơm 1 và bơm 2 tự động dừng, đồng thời động cơ trộn hoạt động để trộn hóa chất.
- Sau 10 giây, động cơ trộn tự động dừng, đồng thời van điện mở và bơm 3 tự động hoạt động đưa sản phẩm ra ngoài.
- Khi hóa chất trong bồn cạn, bơm 3 tự động dừng và van điện tự động đóng lại, đồng thời bơm 1 và bơm 2 tự động hoạt động trở lại bắt đầu chu kỳ mới, quá trình lặp lại 5 lần thì dừng luôn.
- Khi nhấn STOP thì hệ thống dừng hoạt động.

3.2. Sơ đồ công nghệ:



Hình 7-1 :Sơ đồ công nghệ

3.2.1. Mô tả:

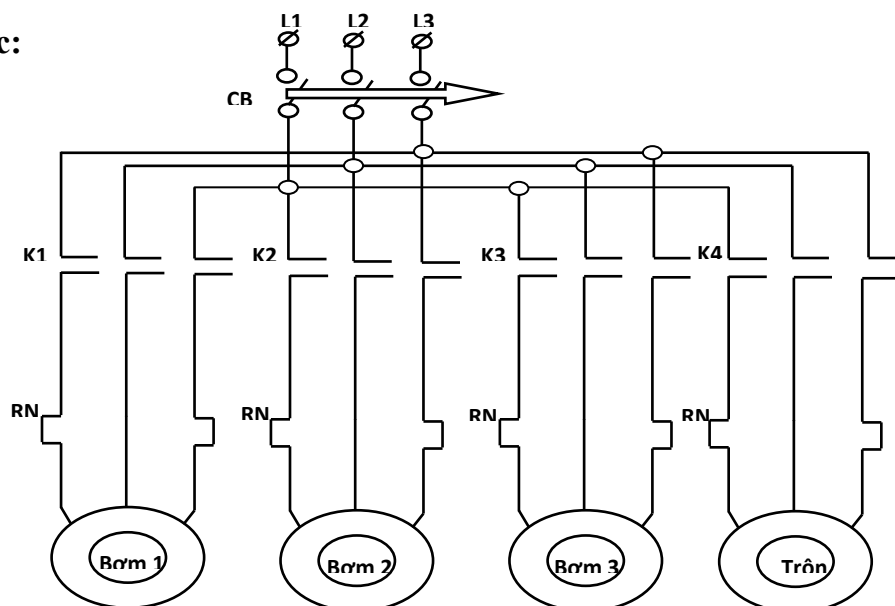
Trên sơ đồ cho thấy có hai đường ống để đưa hai loại hoá chất khác nhau điều khiển bằng bơm 1 và bơm 2 vào bình trộn điều khiển bởi máy trộn, sản phẩm đưa ra bởi van và bơm 3. Theo giới mức hoá chất bằng cảm biến báo đầy và cảm biến báo cạn.

Quá trình được làm việc như sau: Khi nhấn nút START thì bơm 1 và bơm 2 điều khiển qua (Q0.0) và (Q0.1) hoạt động để đưa hai loại hoá chất khác nhau vào bình. Khi dung dịch trong bình đã đạt mức cực đại thì cảm biến báo đầy (I0.4) tác động dừng hai bơm và bắt đầu quá trình trộn, quá trình này được điều khiển bởi động cơ trộn (Q0.2) và thời gian trộn cần thiết là 5 giây. Sau khi trộn xong sản phẩm được đưa ra qua van (Q0.3) và bơm 3 (Q0.4). Khi xả hết sản phẩm thì cảm biến báo cạn (I0.5) tác động đóng van và dừng bơm 3. Đồng thời lúc đó bơm 1 và bơm 2 tự động hoạt động trở lại cho chu kỳ mới, chu trình lặp lại 5 lần thì dừng luôn. Trong quá trình hoạt động có sự cố hoặc nhấn nút STOP thì hệ thống dừng ngay.

3.2.2. Nhiệm vụ:

- Vẽ mạch động lực.
- Lập bảng xác lập ngõ vào/ra.
- Vẽ sơ đồ nối dây PLC.
- Viết chương trình PLC S7-200 theo ngôn ngữ LAD trên phần mềm STEP7-Microwin V3.2 hoặc V4.0.
- Kết nối thiết bị ngoại vi, download, vận hành chương trình trên S7-200, CPU 224.

3.3. Mạch động lực:

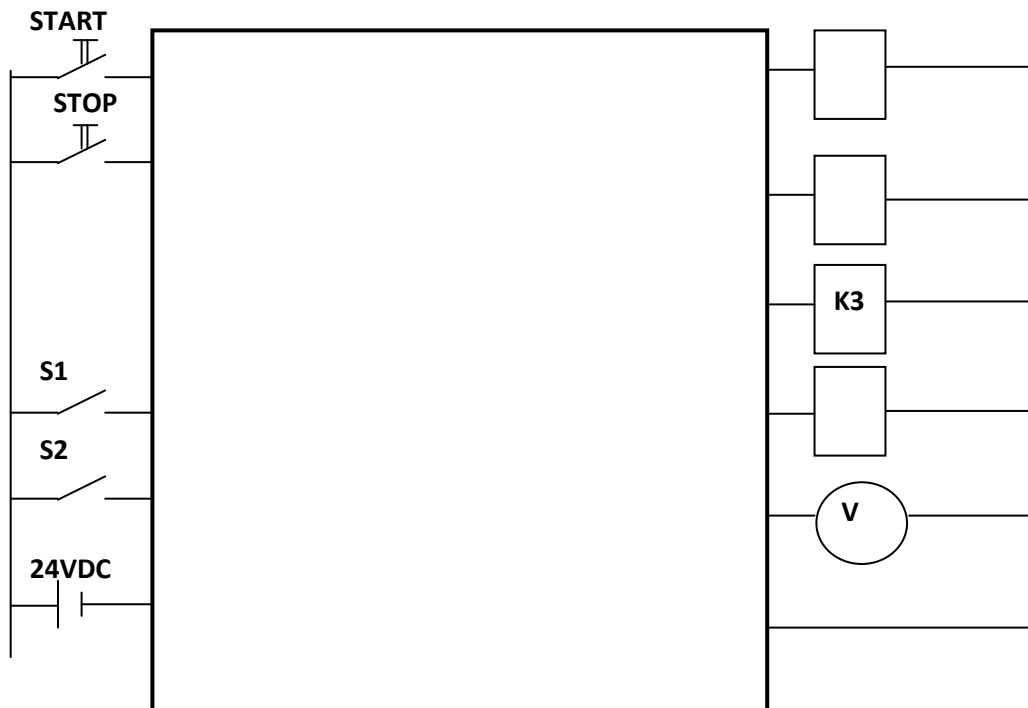


Hình 7-2 :Sơ đồ mạch động lực

3.4. Lập bảng địa chỉ ngõ vào - ngõ ra.

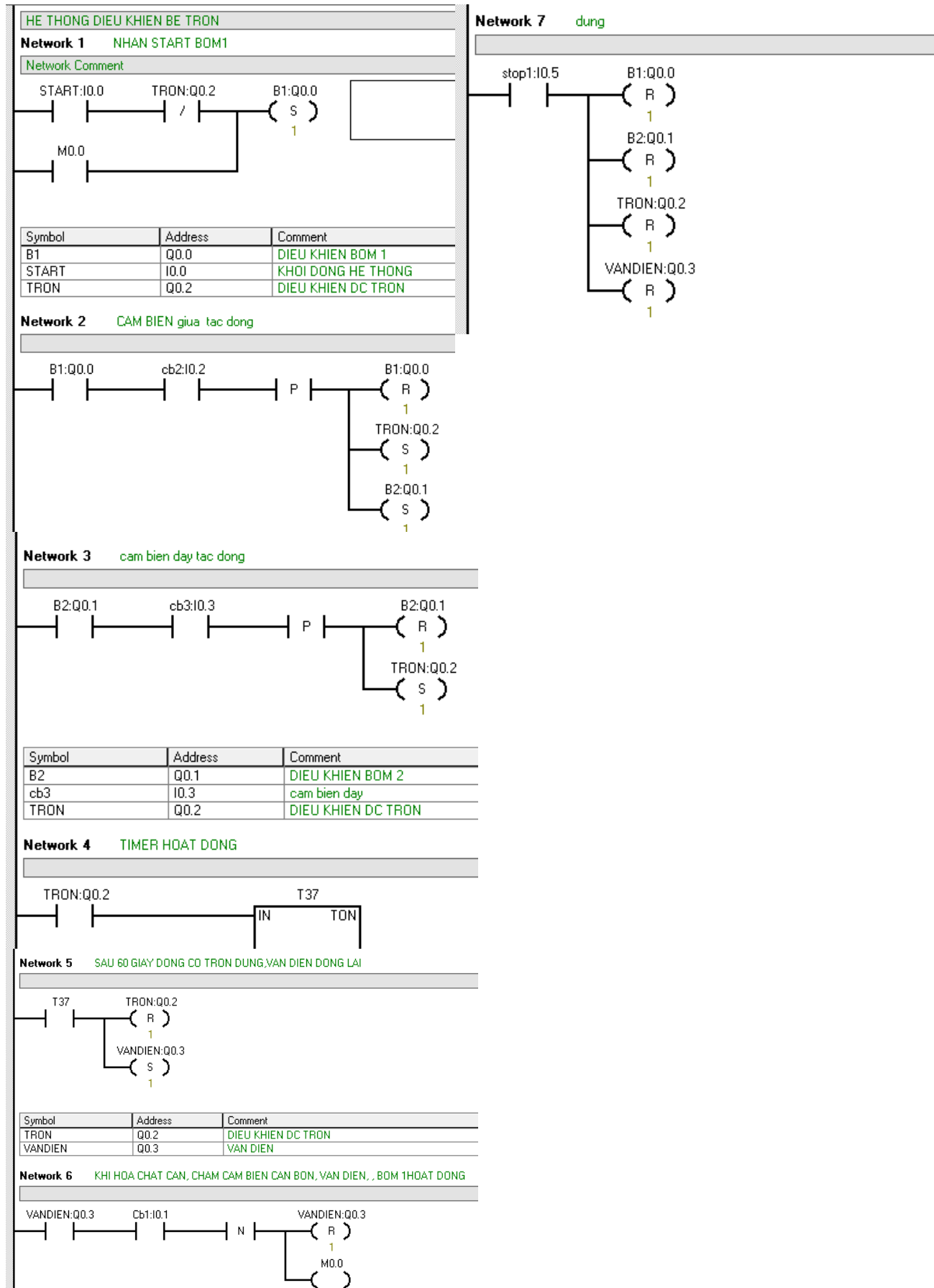
Kí hiệu	Địa chỉ	Mô tả
START	I0.0	Khởi động hệ thống
STOP	I0.1	Dừng hệ thống
S1	I0.4	Cảm biến báo đầy
S2	I0.5	Cảm biến báo cạn
K1	Q0.0	Điều khiển bơm 1
K2	Q0.1	Điều khiển bơm 2
K3	Q0.2	Điều khiển động cơ trộn
K4	Q0.3	Điều khiển bơm 3
V	Q0.4	Điều khiển van xả

3.5. Nối dây PLC:



Hình 7-2 : Sơ đồ kết nối PLC

3.6. Chương trình:



BÀI 8

ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ MỘT CHIỀU BẰNG TÍN HIỆU ANALOG

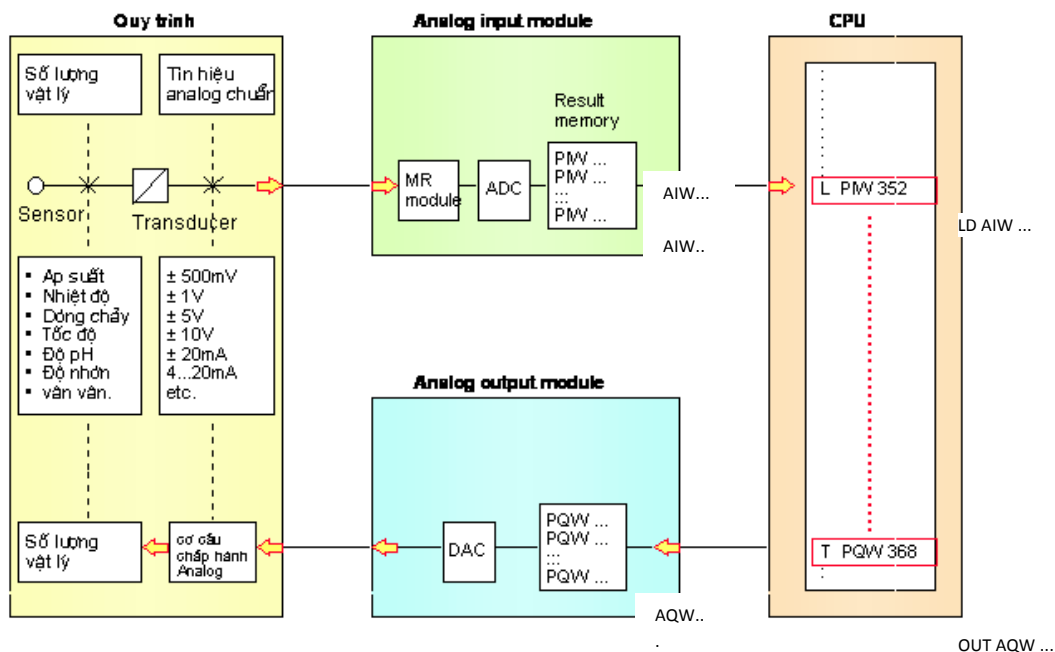
Mục tiêu:

- Trình bày được các bộ chuyển đổi đo.
- Ứng dụng chúng trong các bài toán thực tế
- Lập trình, kết nối, vận hành.
- Giải quyết các công việc một cách hệ thống theo nhóm.

Nội dung chính:

1. Tín hiệu Analog

Trong quá trình điều khiển một hệ thống tự động hoá có thể có các yêu cầu điều khiển liên quan đến việc xử lý các tín hiệu Analog. Các đại lượng vật lý như : nhiệt độ, áp suất, tốc độ, dòng chảy, độ PH... cần phải được các bộ Transducer chuẩn hoá tín hiệu trong phạm vi định mức cho phép trước khi nối tín hiệu vào ngõ vào Analog . Ví dụ : chuẩn của tín hiệu điện áp là từ 0 đến 10 VDC hoặc chuẩn của tín hiệu Analog là dòng điện từ 4 đến 20 mA. Các Modul ngõ vào Analog (AI) bên trong có các bộ chuyển đổi ADC (Analog Digital Converter) để chuyển đổi các tín hiệu Analog nhận được thành các tín hiệu số đưa về CPU qua Bus dữ liệu. Các Modul ngõ ra Analog (AO) bên trong có bộ chuyển đổi DAC (Digital-Analog Converter) chuyển các tín hiệu số nhận được từ CPU ra các giá trị Analog có thể là áp hoặc dòng.



2. Biểu diễn các giá trị Analog

Mỗi một tín hiệu ngõ vào Analog sau khi qua bộ chuyển đổi ADC trong module AI được chuyển thành các số nguyên Integer 16 bit có giá trị từ 0 đến ± 27648 . Do đó địa chỉ vùng nhớ chứa giá trị này là 1 Word. Độ chính xác của phép chuyển đổi này phụ thuộc vào độ phân giải của Modul Analog hiện có, phạm vi độ phân giải là từ 8 đến 15 Bits. Modul Analog có độ phân giải càng cao thì giá trị chuyển đổi càng chính xác. Việc chuyển đổi từ tín hiệu Analog sang tín hiệu số là tỷ lệ thuận và có dạng đường thẳng. Các giá trị Analog sau khi được chuyển đổi thành giá trị số sẽ được chứa vào một Word 16 Bit và lấp đầy các bit trong word này theo thứ tự từ bên trái sang, các Bit trống sẽ bị lấp đầy bằng số 0. (chú ý Bit thứ 15 là Bit dấu : = 0 khi giá trị chuyển đổi là số nguyên dương và = 1 khi giá trị chuyển đổi là số nguyên âm).

Bit số	Các đơn vị		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Giá trị bit	Dec.	Hex.	VZ	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Độ phân giải tính bằng bit có dấu	8	128	80	*	*	*	*	*	*	*	*	1	0	0	0	0	0	0
	9	64	40	*	*	*	*	*	*	*	*	*	1	0	0	0	0	0
	10	32	20	*	*	*	*	*	*	*	*	*	*	1	0	0	0	0
	11	16	10	*	*	*	*	*	*	*	*	*	*	*	1	0	0	0
	12	8	8	*	*	*	*	*	*	*	*	*	*	*	*	1	0	0
	13	4	4	*	*	*	*	*	*	*	*	*	*	*	*	*	1	0
	14	2	2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1
	15	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* = 0 or 1

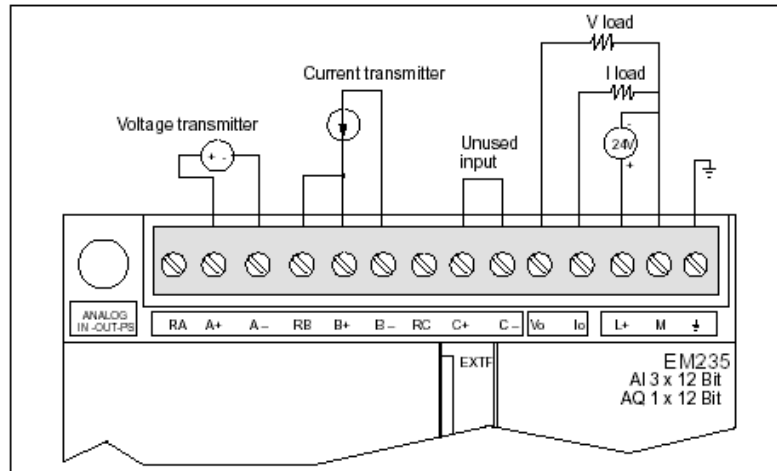
3. Kết nối ngõ vào-ra Analog:

Để đảm bảo tín hiệu Analog có được độ chính xác cao và ổn định cần tuân thủ các điều kiện sau:

- Đảm bảo rằng điện áp 24 VDC cấp nguồn cho Sensor không bị ảnh hưởng bởi nhiễu và ổn định .
- Định tỷ lệ cho module (được mô tả bên dưới).
- Dây nối cho Sensor cần để ngắn nhất tới mức có thể.
- Sử dụng cáp đôi dây xoắn cho sensor.
- Tất cả các ngõ vào không sử dụng phải được nối tắt.
- Tránh bẻ cong dây dẫn thành những góc nhọn.
- Sử dụng máng đi dây hay các ống đi dây cho tuyến dây.

- Tránh đặt các đường dây tín hiệu Analog gần với các đường dây có điện áp cao, nếu 2 đường dây này cắt nhau phải đặt chúng vuông góc với nhau.

Ví dụ về kết nối tín hiệu AI và AO vào Modul analog



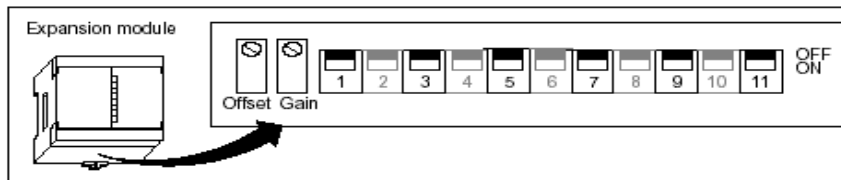
3.1. Phương pháp định tỷ lệ ngõ vào Analog (Input calibration)

Việc định tỷ lệ ngõ vào analog có ảnh hưởng đến tất cả các ngõ vào của modul EM có AI. Để định tỷ lệ ngõ vào một cách chính xác, cần sử dụng một chương trình được thiết kế để tính trung bình các giá trị đọc được từ Modul. Có thể sử dụng Analog Input Filtering wizard trong STEP7-MicroWIN để tạo ra chương trình này. Nên sử dụng 64 giá trị lấy mẫu hoặc hơn để tính giá trị trung bình của tín hiệu Analog.

3.1.1. Để thực hiện việc định tỷ lệ cần theo các bước sau:

- Tắt nguồn cung cấp cho modul, chọn phạm vi ngõ vào mong muốn
- Cấp nguồn lại cho CPU và modul có AI
- Sử dụng một Transmitter, một nguồn áp, hay một nguồn dòng và đặt vào giá trị 0 cho một trong các ngõ vào.
- Đọc giá trị mà CPU nhận được tại ngõ vào tương ứng đó.
- Điều chỉnh biến trở đặt giá trị OFFSET cho tới khi giá trị đọc được là 0.
- Điều chỉnh để tăng giá trị đặt vào tới định mức và xem giá trị mà CPU nhận được.

- Điều chỉnh biến trở GAIN cho tới khi giá trị nhận được là 32000 hoặc tới 1 giá trị số mong muốn.
- Lặp lại các bước trên nếu cần.



Điều chỉnh các Switch và biến trở chỉnh GAIN

Việc chỉnh định các công tắc (Switch) trên modul Analog EM sẽ thay đổi các phạm vi đo lường định mức và độ phân giải của Modul. Các phạm vi và độ phân giải được cho ở bảng dưới đây :

Configuration Switch						Full-Scale Input	Resolution
1 ¹	3	5	7	9	11		
ON	ON	OFF	ON	OFF	OFF	0 to 50 mV	12.5 μ V
ON	ON	OFF	OFF	ON	OFF	0 to 100 mV	25 μ V
ON	OFF	ON	ON	OFF	OFF	0 to 500 mV	125 μ V
ON	OFF	ON	OFF	ON	OFF	0 to 1 V	250 μ V
ON	OFF	OFF	ON	OFF	OFF	0 to 5 V	1.25 mV
ON	OFF	OFF	ON	OFF	OFF	0 to 20 mA ²	5 μ A
ON	OFF	OFF	OFF	ON	OFF	0 to 10 V	2.5 mV
OFF	ON	OFF	ON	OFF	OFF	\pm 25 mV	12.5 μ V
OFF	ON	OFF	OFF	ON	OFF	\pm 50 mV	25 μ V
OFF	ON	OFF	OFF	OFF	ON	\pm 100 mV	50 μ V
OFF	OFF	ON	ON	OFF	OFF	\pm 250 mV	125 μ V
OFF	OFF	ON	OFF	ON	OFF	\pm 500 mV	250 μ V
OFF	OFF	ON	OFF	OFF	ON	\pm 1 V	500 μ V
OFF	OFF	OFF	ON	OFF	OFF	\pm 2.5 V	1.25 mV
OFF	OFF	OFF	OFF	ON	OFF	\pm 5 V	2.5 mV
OFF	OFF	OFF	OFF	OFF	ON	\pm 10 V	5 mV

Sơ đồ công tắc, chỉnh định phạm vi đo định mức và độ phân giải phụ thuộc vào từng Modul Analog. Các thông tin này được lấy từ sổ tay phần cứng của Modul.

3.1.2. Hiệu chỉnh tín hiệu Analog

Trên CPU S7-200 có 2 biến trở (2 biến trở này nằm dưới nắp của module), có thể sử dụng 2 biến trở này để tăng hoặc giảm giá trị được lưu trữ trong các Byte của vùng nhớ Special Memory (SMB 28 và SMB 29). Các giá trị chỉ đọc trong 2 Byte này có thể

được sử dụng cho nhiều chức năng khác nhau. Chẳng hạn, dùng để cập nhật giá trị hiện hành cho 1 Timer, một Counter, thay đổi giá trị đặt trước, đặt các giá trị giới hạn.

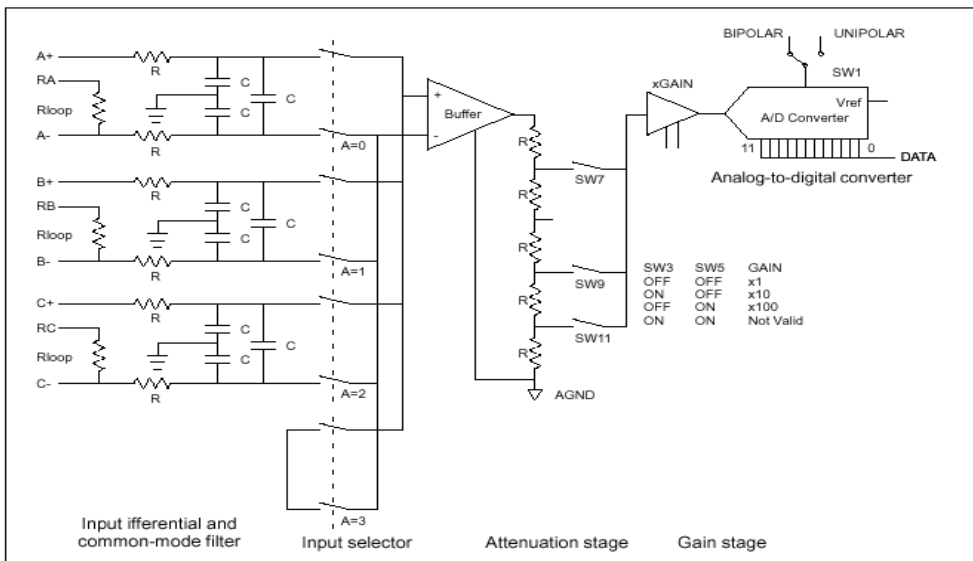
Byte nhớ SMB 28 lưu trữ giá trị số biểu diễn vị trí chỉnh 0. SMB 29 lưu trữ giá trị số biểu diễn vị trí chỉnh 1. Sự điều chỉnh Analog có giới hạn từ 0 tới 255 và độ tin cậy tốt nhất trong phạm vi từ 10 đến 200.

Để thực hiện điều chỉnh này, phải sử dụng một Tuốc vít nhỏ: nếu xoay biến trở sang phải là tăng giá trị, còn xoay sang trái là giảm giá trị.

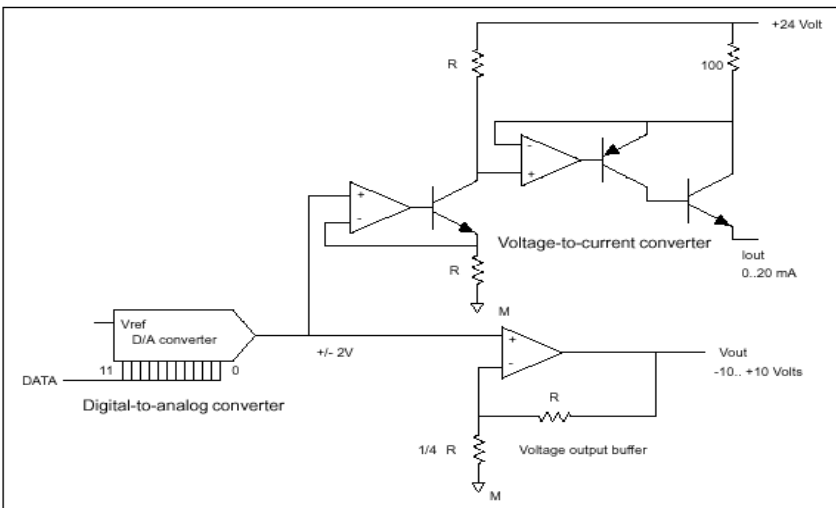
Dưới đây là một ví dụ ứng dụng: Timer T33 đóng tiếp điểm khi VW 100 đạt giá trị đặt trước

Sau đây là sơ đồ nguyên lý mạch của modul EM 235 3AI/ 1AO

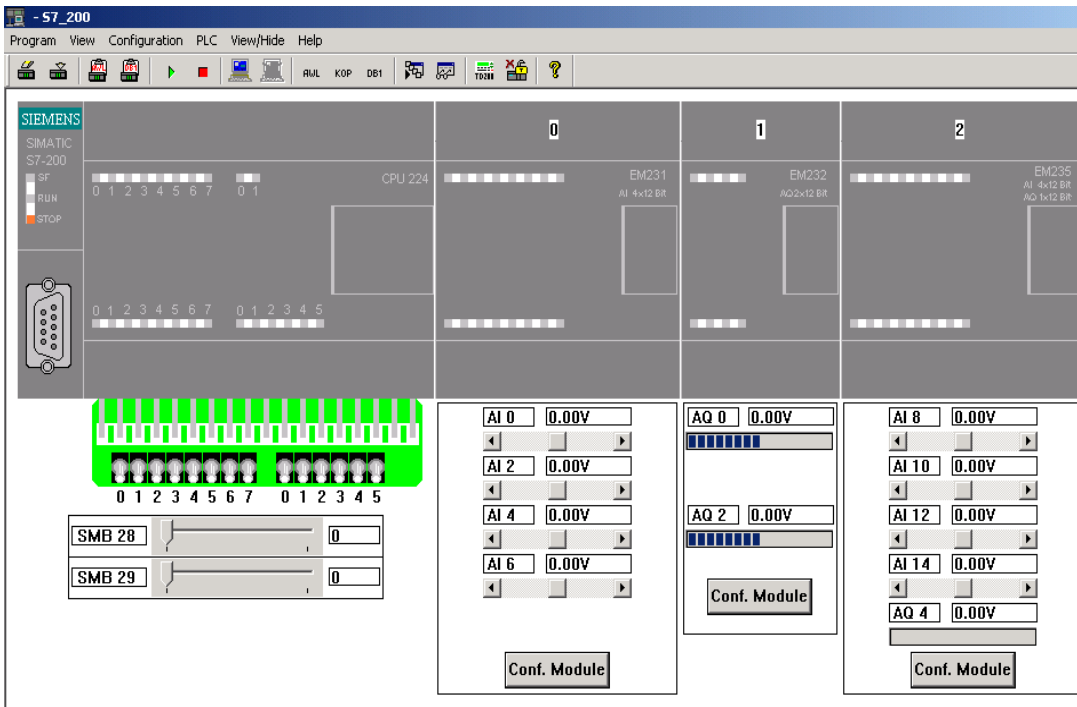
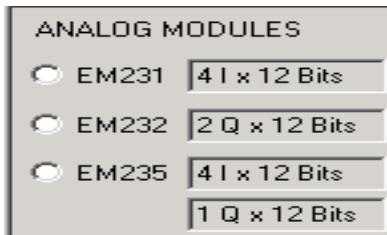
Sơ đồ mạch ngõ vào :



Sơ đồ mạch ngõ ra :



4.5. Giới thiệu module Analog S7-200:



Tín hiệu Analog là các tín hiệu tương tự (0 – 10VDC, hoặc 4-20mA.....), Hầu hết các ứng

dụng của chương trình PLC Siemens nói riêng hay các ứng dụng khác đều cần phải đọc các tín hiệu analog. Tín hiệu analog có thể là tín hiệu từ các cảm biến đo khoảng cách, cảm biến áp suất, cảm biến đo trọng lượng.....

Các bước đọc tín hiệu Analog:

4.5.1. Đọc tín hiệu analog từ Modul EM231:

Các tín hiệu có thể đọc được từ Modul EM231 (tùy thuộc việc chọn các Switch trên modul):

Tín hiệu đơn cực (Tín hiệu điện áp): 0-10VDC, 0-5VDC

Tín hiệu lưỡng cực (tín hiệu điện áp): -5VDC – 5VDC, -2.5VDC – 2.5VDC

Tín hiệu dòng điện :0 – 20mA (có thể đọc được 4-20mA)

Tín hiệu Analog sẽ được đọc vào AIW0,AIW2 tương ứng,tùy thuộc vào vị trí của tín hiệu đưa vào modul

Modul EM231 có 4 ngõ vào Analog,do vậy vị trí các ngõ vào tương ứng là: AIW0,AIW2,AIW4,AIW6

Tín hiệu analog là tín hiệu điện áp ,tuy nhiên giá trị mà AIW đọc vào không phải là giá trị điện áp ,mà là giá trị đã được quy đổi tương ứng 16bit.

Trường hợp đơn cực : Giá trị từ 0 – 64000 tương ứng với (0-10V,0-5V hay 0-20mA)

Liên kết sự kiện ngắt số 8 với chương trình ngắt INT_0 (Khi xảy ra sự kiện số 8 thì chương trình INT_0 được thực thi)

Cho phép ngắt (**ENI**)

Kết thúc sự kiện ngắt số 8,sự kiện ngắt số 8 sẽ được cho phép lại khi có lệnh **ENI**

Trường hợp lưỡng cực : Giá trị từ -32000 – 32000 tương ứng với (-5VDC – 5VDC hay -2.5VDC – 2.5VDC).

Ví dụ :

Trường hợp đơn cực: giá trị đọc vào của AIW0 = 32000,khi đó giá trị điện áp tương ứng là : $(32000 \times 10VDC / 64000) = 5VDC$ (Tầm chọn 0 – 10VDC)

Trường hợp lưỡng cực : Giá trị đọc vào của AIW0 = 16000,khi đó giá trị điện áp tương ứng là : $(16000 \times 5VDC / 32000) = 2.5VDC$ (Tầm đo -2.5VDC – 2.5VDC)

Do vậy căn cứ vào giá trị đọc vào của AIW ta có thể dùng quy tắc “tam suất”,từ đó có thể tính được giá trị điện áp tương ứng.Từ giá trị điện áp ta có thể suy ra giá trị mong muốn.

- Thông thường các tín hiệu Analog đọc vào bao giờ người sử dụng cũng mong muốn

đọc được chính giá trị mong muốn (Ví dụ: giá trị khối lượng trong đọc đầu cân Loadcell, giá trị áp suất trong đọc tín hiệu từ cảm biến áp suất.....)

- Phương pháp đọc Analog trong trường hợp này ta sẽ không cần quan tâm nhiều đến

chế độ đơn cực hay lưỡng cực,mà chỉ cần xác định được 2 điểm,từ đó lập được phương trình đường thẳng (Giá trị mong muốn đọc theo AIW)

- Ví dụ: Để đọc khối lượng từ đầu cân :Ta xây dựng hàm Khối lượng theo AIW(là tín

hiệu đọc vào)

- Bước 1: Ta cần xác định 2 điểm:

Điểm 1: Ta online trên máy tính, đọc giá trị AIW0 là x1, trong trường hợp ở điểm 1 (

Điểm 1 là điểm ta đặt quả cân chuẩn 1: có khối lượng m1 lên bàn cân), Tương tự ta có thể xác định được điểm 2 (tương ứng x2 và m2).

Từ đó ta có 2 điểm : Điểm 1 (x1,m1) , Điểm 2 (x2,m2).

Phương trình đường thẳng đi qua 2 điểm 1,2 có dạng:

$(X-X1/X2-X1) = (Y-Y1/Y2-Y1)$, Từ đó rút Y theo X

Đó chính là phương trình khối lượng theo AIW.

Ví dụ cụ thể: Điểm 1 (0,0), điểm 2 (32000,1000)

Phương trình lập:

$(X-0/32000-0) = (Y-0/1000-0)$ Từ đó suy ra:

$Y= 1xX/ 32$

Vậy : Khối lượng = AIW / 32

4.5.2. Xuất tín hiệu analog qua modul EM232:

Các tín hiệu có thể xuất ra Modul EM232(tùy thuộc việc chọn các Switch trên modul):

Tín hiệu đơn cực (Tín hiệu dòng điện): 0-20mA

Tín hiệu lưỡng cực (tín hiệu điện áp): -10VDC – 10VDC

Tín hiệu 0 -20mA tương ứng với giá trị 0 – 32000

Tín hiệu -10VDC – 10VDC tương ứng -32000 – 32000

Giá trị xuất ra Modul EM232 được đưa vào ô nhớ AQW tương ứng.

4.5.3. Modul EM235:

Các tín hiệu có thể đọc được thông qua Modul EM235 (Tùy theo Switch chọn trên Modul):

Đơn cực : 0 – 50mV , 0 – 100mV , 0 – 500mV , 0 – 1V , 0 – 5VDC , 0 – 20mA , 0 – 10VDC.

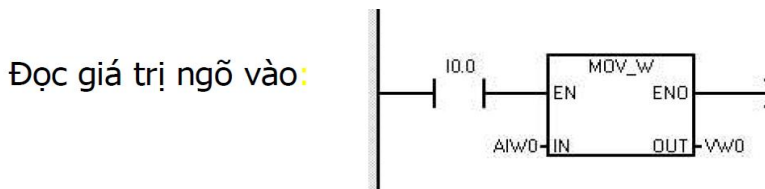
Lưỡng cực : +25mV , +50mV , +100mV , +250mV , +500mV , +1VDC , +2.5VDC , +5VDC , +10VDC

Giá trị tương ứng cho chế độ đơn cực : Từ 0 – 64000

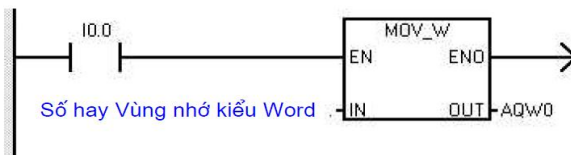
Giá trị tương ứng cho chế độ lưỡng cực : -32000 – 32000

Ngoài ra Modul EM235 còn có 2 Ngõ ra Analog output tương ứng : +10VDC, 0 –20mA

Ví dụ:



Xuất giá trị ngõ ra analog:



5. Điều khiển mô hình hệ thống điều khiển nhiệt độ

5.1 Điều khiển nhiệt độ của lò:

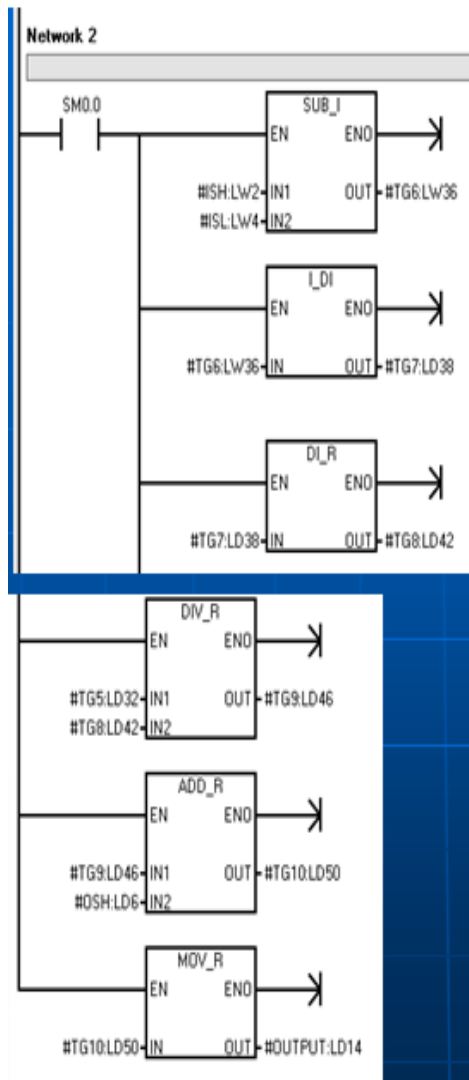
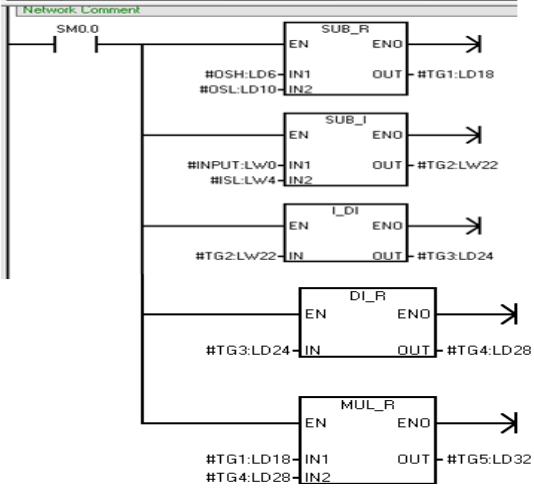
Khi nhiệt độ lớn hơn hoặc bằng 100 độ thì ngưng cấp nhiệt cho lò

Khi nhiệt độ giảm nhỏ hơn hoặc bằng 70 độ thì cấp nhiệt cho lò

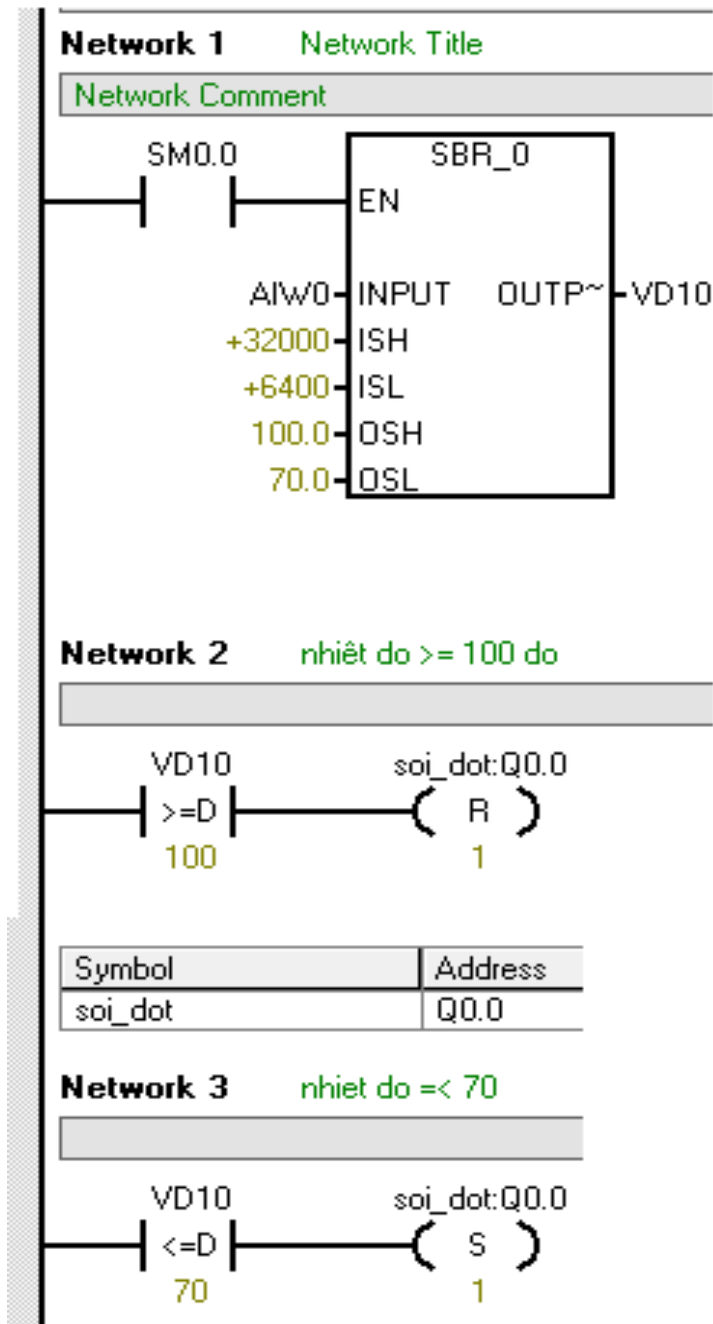
5.2. Chương trình:

5.2.1: chương trình con:

	EN	IN	BOOL
LW0	INPUT	IN	INT
LW2	ISH	IN	INT
LW4	ISL	IN	INT
LD6	OSH	IN	REAL



5.2.2. chương trình chính



TÀI LIỆU THAM KHẢO

- Tài liệu thực hành PLC-S7 200 – Trung tâm Việt Đức – Trường ĐH Sư phạm Kỹ thuật TPHCM.
- Hướng dẫn thiết kế mạch và lập trình PLC – Trần Thế San (biên dịch) – NXB Đà Nẵng – 2005.
- Điều khiển logic lập trình PLC – Tăng Văn Mùi (biên dịch) – NXB Thống kê – 2006.
- Các tạp chí, tài liệu kỹ thuật có liên quan.