

ỦY BAN NHÂN DÂN QUẬN 5
TRƯỜNG TRUNG CẤP NGHỀ KỸ THUẬT CÔNG NGHỆ HÙNG VƯƠNG



GIÁO TRÌNH
PLC cơ bản

Nghề: Điện tử công nghiệp
TRÌNH ĐỘ TRUNG CẤP

LỜI GIỚI THIỆU

Để đáp ứng yêu cầu giảng dạy chương trình đào tạo nghề “Điện tử công nghiệp” cũng như việc cung cấp tài liệu giúp cho sinh viên học tập, khoa Điện tử chúng tôi đã tiến hành biên soạn giáo trình “PLC cơ bản”.

Giáo trình này giúp các bạn có thêm kỹ năng:

- Trình bày được các khái niệm về điều khiển lập trình chính xác theo nội dung đã học
- Trình bày được cấu trúc và phương thức hoạt động của các lệnh cơ bản

Đây là công trình được viết bởi đội ngũ giáo viên đã và đang công tác tại trường TCN KTCN Hùng Vương cùng với sự góp ý và phản biện của các doanh nghiệp trong lĩnh vực liên quan, tuy vậy, cuốn sách chắc chắn vẫn không tránh khỏi những khiếm khuyết. Chúng tôi mong nhận được ý kiến đóng góp của bạn đọc để cuốn sách được hoàn thiện hơn trong lần tái bản.

Xin trân trọng giới thiệu cùng bạn đọc!

Quận 5, ngày tháng năm 2012

Biên soạn

Phạm Phú Thọ

MỤC LỤC

ĐỀ MỤC	TRANG
GIỚI THIỆU MÔ ĐUN	1
1. Bài 1. Đại cương về điều khiển lập trình	3
1.1. Khái quát về điều khiển	3
1.2. Điều khiển nối cứng và điều khiển lập trình.....	5
1.3. So sánh PLC với các thiết bị điều khiển thông thường.....	6
1.4. Các ứng dụng của PLC trong thực tế	8
2. Bài 2. Cấu trúc và phương thức hoạt động của một PLC.	9
2.1. Cấu trúc của một PLC	9
2.2. Thiết bị điều khiển lập trình S7 – 200.....	12
2.3. Địa chỉ các ngõ vào / ra	13
2.4. Cấu trúc bộ nhớ của S7 – 200.....	15
2.5. Xử lý chương trình.....	16
3. Bài 3. Kết nối dây giữa PLC và thiết bị ngoại vi.....	21
3.1. Kết nối dây giữa PLC và thiết bị ngoại vi	21
3.2. Kiểm tra việc nối dây bằng phần mềm.....	30
3.3. Cài đặt và sử dụng phần mềm Step 7-Micro/win 3.2.....	31
4. Bài 4. Các phép toán nhị phân của PLC.....	33
4.1. Các liên kết logic.....	33
4.2. Các lệnh ghi / xóa giá trị cho tiếp điểm.....	45
4.3. Timer.....	50
4.4. Counter.....	62
4.5. Các bài tập ứng dụng.....	68
4.6. Lệnh nhảy và lệnh ọi chương trình con.....	77
5. Bài 5. Các phép toán số của PLC.....	78
5.1. Chức năng truyền dẫn.....	78
5.2. Chức năng so sánh.....	80
5.3. Chức năng dịch chuyển	82
5.4. Chức năng chuyển đổi	84
5.5. Chức năng toán học	92
6. Bài 6. Xử lý tín hiệu analog.....	94
6.1. Tín hiệu analog.....	94
6.2. Biểu diễn các giá trị analog	94
6.3. Kết nối các ngõ vào / ra analog.....	95
6.4. Hiệu chỉnh tín hiệu analog.....	97
7. Bài 7: Các bài tập ứng dụng trong điều khiển động cơ.....	104
7.1. Giới thiệu	104
7.2. Cách kết nối dây.....	105

7.3. Tóm tắt các mô hình và bài tập ứng dụng	106
8. Tài liệu tham khảo	149

GIỚI THIỆU VỀ MÔ ĐUN

Vị trí, tính chất của mô đun:

- * Vị trí của mô đun: Môđun được bố trí dạy cuối chương trình sau khi học xong các môn chuyên môn như điện tử công suất, Kỹ thuật xung – số...
- * Tính chất của mô đun: Là mô đun bắt buộc.

Mục tiêu của mô đun

Sau khi học xong môđun này người học có năng lực:

- * Về kiến thức:
 - Trình bày được các khái niệm về điều khiển lập trình chính xác theo nội dung đã học
 - Trình bày được cấu trúc và phương thức hoạt động của các lệnh cơ bản
- * Về kỹ năng:
 - Thực hiện lập trình các bài tập ứng dụng dùng PLC đạt các yêu cầu về kỹ thuật và công nghệ
 - Kết nối mạch điện theo yêu cầu công nghệ
- * Về thái độ: Rèn luyện tính tỉ mỉ, chính xác và an toàn vệ sinh công nghiệp

Nội dung chính của mô đun:

Mô đun PLC cơ bản gồm các nội dung chính sau:

- Bài 1. Đại cương về điều khiển lập trình.
- Bài 2. Cấu trúc và phương thức hoạt động của một PLC.
- Bài 3. Kết nối dây giữa PLC và thiết bị ngoại vi.
- Bài 4. Các phép toán nhị phân của PLC.
- Bài 5. Các phép toán số của PLC.
- Bài 6. Xử lý tín hiệu analog.
- Bài 7. Các bài tập ứng dụng trong điều khiển động cơ

Bài 1: ĐẠI CƯƠNG VỀ ĐIỀU KHIỂN LẬP TRÌNH

1.1. Khái quát về điều khiển:

Trong ứng dụng các công nghệ khoa học vào sản xuất công nghiệp, mục tiêu tăng năng suất lao động được giải quyết bằng con đường tăng mức độ tự động hóa các quá trình và thiết bị sản xuất nhằm mục đích tăng sản lượng, cải thiện chất lượng và độ chính xác của sản phẩm.

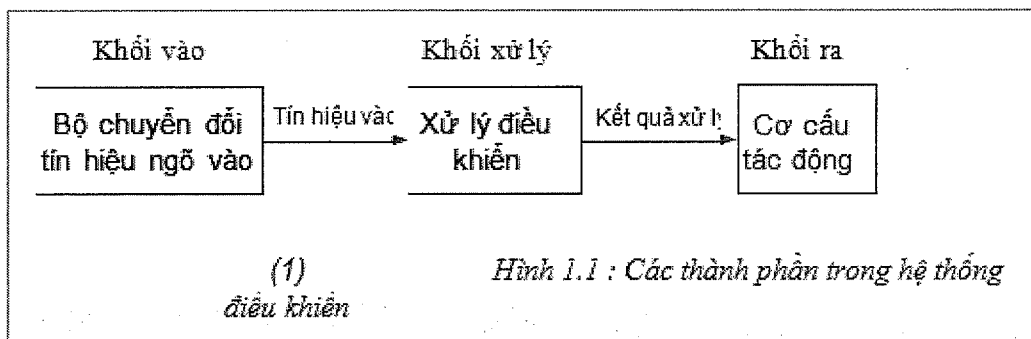
Tự động hóa trong sản xuất nhằm thay thế một phần hoặc toàn bộ các thao tác vật lý của công nhân vận hành máy thông qua hệ thống điều khiển. Những hệ thống điều khiển này có thể điều khiển quá trình sản xuất với độ tin cậy cao, ổn định mà không cần sự tác động nhiều của người vận hành. Điều này đòi hỏi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

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

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

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

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



+ Khối vào:

Để chuyển đổi các đại lượng vật lý thành các tín hiệu điện, các bộ chuyển đổi có thể là các nút nhấn, cảm biến, điện trở đo sức căng.v.v... và tùy theo bộ chuyển đổi mà tín hiệu ra khỏi khối vào có dạng ON/OFF (Binary) hoặc dạng liên tục (Analog).

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

Bảng 1.1: Các dạng tín hiệu vào

+Khối xử lý:

Khối này thay thế người vận hành thực hiện các thao tác đảm bảo quá trình hoạt động. Từ thông tin tín hiệu khối vào hệ thống điều khiển phải tạo ra được những tín hiệu ra cần thiết để đáp ứng yêu cầu điều khiển đã xác định trong phần xử lý. Tín hiệu điều khiển được thực hiện theo 2 cách:

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

+Khối ra:

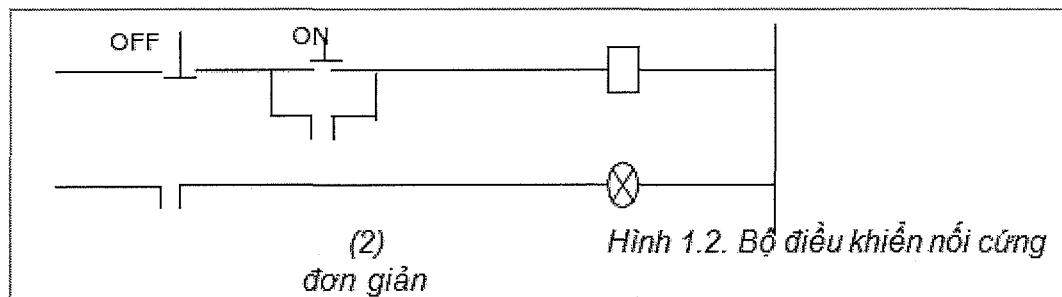
Tín hiệu ra là kết quả của quá trình xử lý của hệ thống điều khiển. Các tín hiệu này được sử dụng để tạo ra những hoạt động đáp ứng cho các thiết bị ở ngõ ra.

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

Bảng 1.2. Các dạng cơ cấu tác động ở ngõ ra.

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

Trong các bộ điều khiển nối cứng, các thành phần chuyển mạch như các rơ le, contactor, các công tắc, đèn báo, động cơ, vv... được nối cố định với nhau. Toàn bộ chức năng điều khiển, cách tiến hành chương trình được xác định qua cách thức nối các rơ le, công tắc, ... với nhau theo sơ đồ thiết kế. Khi muốn thay đổi lại hệ thống thì phải nối dây lại cho hệ thống điều khiển nên đối với hệ thống phức tạp thì việc làm này đòi hỏi tốn nhiều thời gian, chi phí nên hiệu quả đem lại không cao.



Trong công nghiệp, sự ứng dụng các công nghệ khoa học kỹ thuật vào sản xuất nên nhu cầu tự động hóa ngày càng tăng, đòi hỏi kỹ thuật điều khiển phải đáp ứng đủ các yêu cầu:

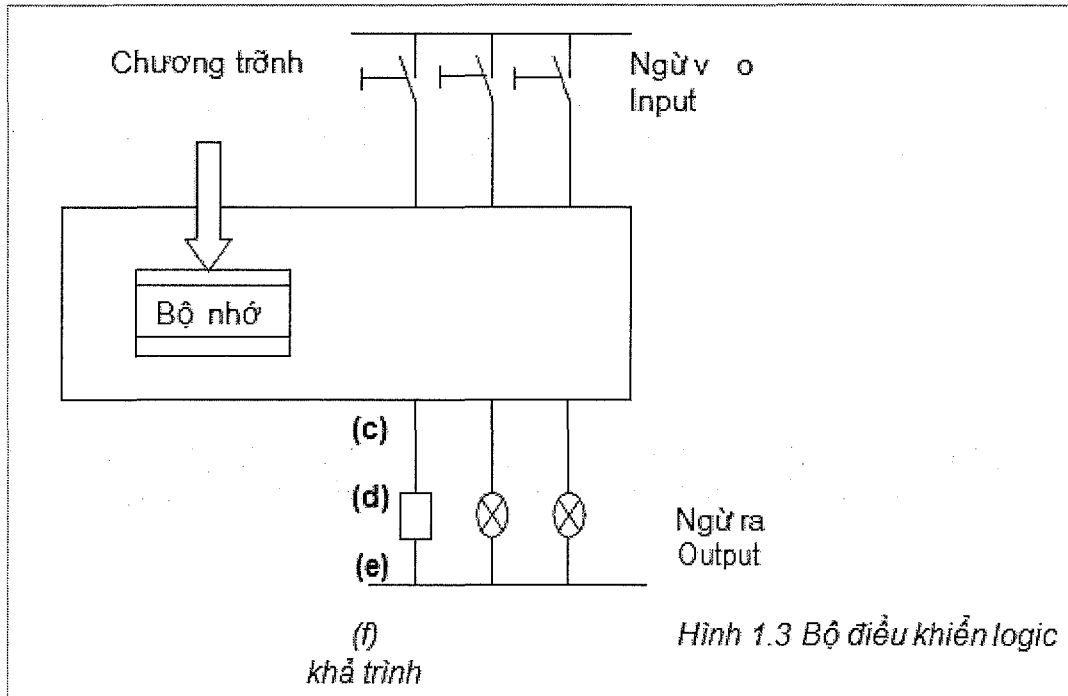
- Dễ dàng thay đổi chức năng điều khiển dựa trên các thiết bị cũ.
- Thiết bị điều khiển dễ dàng làm việc với các dữ liệu, số liệu.
- Kích thước vật lý gọn gàng, dễ bảo quản, dễ sửa chữa.
- Hoàn toàn tin cậy trong môi trường công nghiệp.

Hệ thống điều khiển để đáp ứng được các yêu cầu trên phải sử dụng bộ vi xử lý, bộ điều khiển lập trình, điều khiển qua các cổng giao tiếp với máy tính.

Bộ điều khiển logic khả trình PLC (*Programable Logic Controller*) là loại thiết bị cho phép thực hiện linh hoạt các thuật toán điều khiển thông qua các

ngôn ngữ lập trình. Với chương trình điều khiển của PLC đã tạo cho nó trở thành một bộ điều khiển số nhỏ gọn, dễ dàng thay đổi thuật toán, các số liệu và trao đổi thông tin với môi trường xung quanh.

Các chương trình điều khiển được định nghĩa là tuần tự trong đó các tiếp điểm, cảm biến được sử dụng để từ đó kết hợp với các hàm logic, các thuật toán và các giá trị xuất của nó để điều khiển tác động hoặc không tác động đến các cuộn dây điều hành. Trong quá trình hoạt động, toàn bộ chương trình được lưu vào trong bộ nhớ và tiến hành truy xuất trong quá trình làm việc.



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

Hiện nay, các hệ thống điều khiển bằng PLC đang dần dần thay thế cho các hệ thống điều khiển bằng Relay, Contactor thông thường. Ta hãy thử so sánh ưu khuyết điểm của hai hệ thống trên:

Hệ thống điều khiển thông thường:

- Thô kệch do có quá nhiều dây dẫn và relay trên bảng điều khiển.
- Tốn khá nhiều thời gian cho việc thiết kế, lắp đặt.
- Tốc độ hoạt động chậm.
- Công suất tiêu thụ lớn.
- Mỗi lần muốn thay đổi chương trình thì phải lắp đặt lại toàn bộ, tốn nhiều thời gian.
- Khó bảo quản và sửa đổi.

Hệ thống điều khiển bằng PLC:

- Những dây kết nối trong hệ thống giảm được 80% nên nhỏ gọn hơn.
- Công suất tiêu thụ ít hơn.

- Sự thay đổi các ngõ vào, ra và điều khiển hệ thống trở nên dễ dàng hơn nhờ phần mềm điều khiển bằng máy tính.
- Tốc độ hoạt động của hệ thống nhanh hơn.
- Bảo trì và bảo quản dễ dàng hơn.
- Độ bền và độ tin cậy vận hành cao.
- Giá thành của hệ thống giảm khi số tiếp điểm tăng.
- Có thiết bị chống nhiễu.
- Ngôn ngữ lập trình dễ hiểu.
- Dễ lập trình và có thể lập trình trên máy tính, thích hợp cho việc thực hiện các lệnh tuần tự của nó.
- Các mô đun rời cho phép thay thế hoặc thêm vào khi cần thiết.

Do những lý do trên PLC thể hiện rõ ưu điểm của nó so với thiết bị điều khiển thông thường khác. PLC còn có khả năng thêm vào hay thay đổi các lệnh tùy theo yêu cầu của công nghệ. Khi đó ta chỉ cần thay đổi chương trình của nó, điều này nói lên tính năng điều khiển khá linh động của PLC. Ta có thể so sánh PLC với các hệ thống khác qua bảng tóm tắt sau:

Chỉ tiêu so sánh	Rơ-le	Mạch số	Máy tính	PLC
Giá thành từng chức năng	Khá thấp	Thấp	Cao	Thấp
Kích thước vật lý	Lớn	Rất gọn	Khá gọn	Rất gọn
Tốc độ điều khiển	Chậm	Rất nhanh	Khá nhanh	Nhanh
Khả năng chống nhiễu	Xuất sắc	Tốt	Khá tốt	Tốt
Lắp đặt	Mất thời gian thiết kế và lắp đặt	Mất thời gian thiết kế	Mất nhiều thời gian lập trình	Lập trình và lắp đặt đơn giản
Khả năng điều khiển tác vụ phức tạp	Không	Có	Có	Có
Dễ thay đổi điều khiển	Rất khó	Khó	Khá đơn giản	Rất đơn giản
Công tác bảo trì	Kém; có quá nhiều công tác	Kém; nếu IC được hàn	Kém; có rất nhiều mạch điện tử chuyên dùng	Tốt-các mô-đun được tiêu chuẩn hoá

Theo bảng so sánh, PLC có những đặc điểm về phần cứng và phần mềm làm cho nó trở thành bộ điều khiển công nghiệp được sử dụng rộng rãi

1.4. Các ứng dụng của PLC trong thực tế

Do những đặc điểm nổi bật của PLC trong điều khiển, nên ngày nay nó được sử dụng rất rộng rãi trong các giải pháp tự động hoá trong công nghiệp ở rất nhiều lĩnh vực:

- Điều khiển thang máy, thiết bị nâng, hạ hàng.
- Điều khiển các quy trình sản xuất: đóng gói bao bì, xi măng, bia...v.v
- Tự động hoá các hệ thống dịch vụ: trạm xăng, trạm rửa xe ô tô, máy bơm nước, máy bán nước tự động...v.v
- Tự động hoá các máy công cụ: lò sấy, xi mạ...v.v

Tuy nhiên không phải bất cứ hệ thống điều khiển nào cũng sử dụng PLC mà tùy vào yêu cầu cụ thể và so sánh về yếu tố kinh tế mà ta chọn phương án điều khiển thích hợp.

Bài 2: CẤU TRÚC VÀ PHƯƠNG THỨC HOẠT ĐỘNG CỦA MỘT PLC

2.1. Cấu trúc của một PLC

Thiết bị điều khiển logic khả trình PLC (*Programmable Logic Controller*), là loại thiết bị cho phép thực hiện linh hoạt các thuật toán điều khiển số thông qua một ngôn ngữ lập trình, thay cho việc phải thể hiện thuật toán đó bằng mạch số. Như vậy, với chương trình điều khiển này, PLC trở thành một bộ điều khiển số nhỏ gọn, dễ thay đổi thuật toán và đặc biệt dễ trao đổi thông tin với môi trường xung quanh (với các PLC khác hoặc với máy tính). Toàn bộ chương trình điều khiển được lưu nhớ trong bộ nhớ của PLC dưới dạng các khối chương trình (khối OB, FC hoặc FB) và được thực hiện lặp theo chu kỳ của vòng quét (*Scan*).

Để có thể thực hiện được một chương trình điều khiển, tất nhiên PLC phải có chức năng như một máy tính, nghĩa là phải có bộ vi xử lý (CPU), một bộ điều hành, bộ nhớ để lưu chương trình điều khiển, dữ liệu.... PLC còn phải có các cổng vào/ra để giao tiếp được các đối tượng điều khiển và để trao đổi thông tin với môi trường xung quanh.

Bên cạnh đó, nhằm phục vụ bài toán điều khiển số, PLC còn cần phải có thêm các khối chức năng đặc biệt khác như: bộ đếm (Counter), bộ thời gian (Timer) ... và những khối hàm chuyên dụng.

Thiết bị logic khả trình được lắp đặt sẵn thành bộ. Trước tiên chúng chưa có một nhiệm vụ nào cả. Tất cả các cổng logic cơ bản, chức năng nhớ, timer, counter v.v... được nhà chế tạo tích hợp trong chúng và được kết nối với nhau bằng chương trình cho một nhiệm vụ điều khiển cụ thể nào đó. Có nhiều thiết bị điều khiển và được phân biệt với nhau qua các chức năng sau:

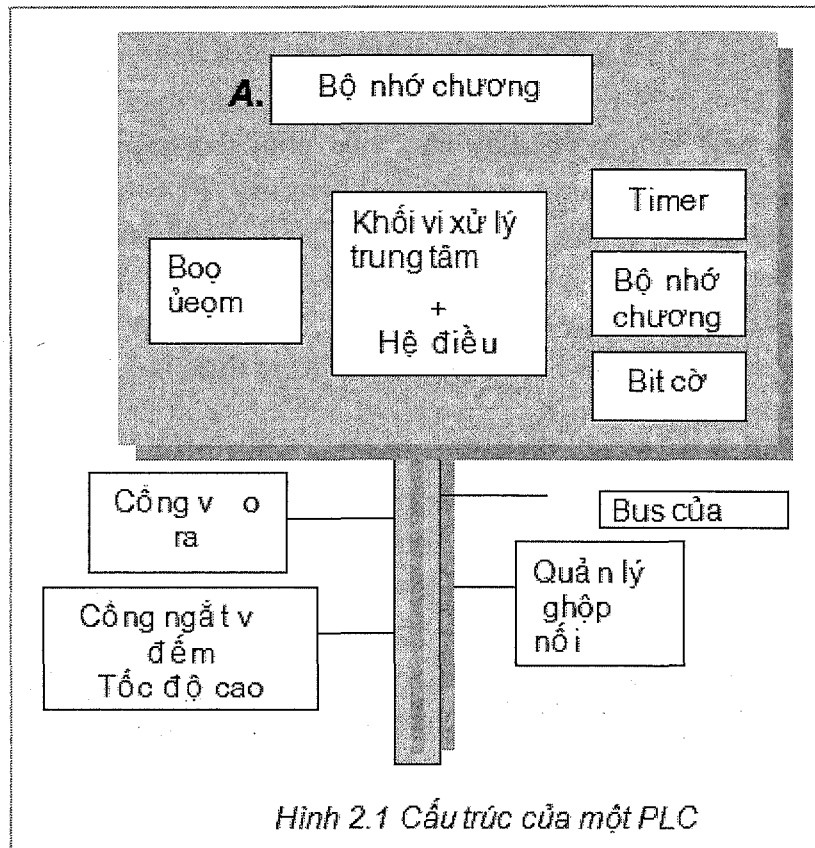
- Các ngõ vào và ra
- Dung lượng nhớ
- Bộ đếm (counter)
- Bộ định thời (timer)
- Bit nhớ
- Các chức năng đặc biệt
- Tốc độ xử lý
- Loại xử lý chương trình.

Các thiết bị điều khiển lớn thì được lắp thành các modul riêng. Đối với các thiết bị điều khiển nhỏ, chúng được lắp đặt chung trong một bộ. Các bộ điều khiển này có số lượng ngõ vào/ra cho trước cố định.

Thiết bị điều khiển được cung cấp tín hiệu bởi các tín hiệu từ các cảm biến ở bộ phận ngõ vào của thiết bị tự động. Tín hiệu này được xử lý tiếp tục thông qua chương trình điều khiển đặt trong bộ nhớ chương trình. Kết quả xử lý được đưa ra bộ phận ngõ ra của thiết bị tự động để đến đối tượng điều khiển hay khâu

điều khiển ở dạng tín hiệu.

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

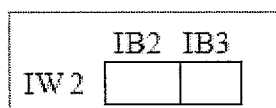


Thông tin xử lý trong PLC được lưu trữ trong bộ nhớ của nó. Mỗi phần tử vi mạch nhớ có thể chứa 1 bit dữ liệu. Bit dữ liệu (Data Binary Digital) là một chữ số nhị phân, chỉ có thể là một trong hai giá trị là 1 hoặc 0. Tuy nhiên các vi mạch nhớ thường được tổ chức thành các nhóm để có thể chứa 8 bit dữ liệu. Mỗi chuỗi 8 bit dữ liệu được gọi là một byte. Mỗi mạch nhớ là một byte (byte nhớ), được xác nhận bởi một con số gọi là địa chỉ (address). Byte nhớ đầu tiên có địa chỉ 0. Dữ liệu chứa trong byte nhớ gọi là nội dung.

Địa chỉ của một byte nhớ là cố định và mỗi byte nhớ trong PLC có một địa chỉ riêng của nó. Địa chỉ của byte nhớ khác nhau sẽ khác nhau, nội dung chứa trong một byte nhớ là đại lượng có thể thay đổi được. Nội dung byte nhớ chính là dữ liệu được lưu trữ tức thời trong bộ nhớ.

Để lưu giữ một dữ liệu mà một byte nhớ không thể chứa hết được thì PLC cho phép một cặp 2 byte nhớ cạnh nhau được xem xét như là một đơn vị nhớ và được gọi là một từ đơn (Word). Địa chỉ thấp hơn trong 2 byte nhớ được dùng làm địa chỉ của từ đơn.

Ví dụ: Từ đơn có địa chỉ là 2 thì các byte nhớ có địa chỉ là 2 và 3 với 2 là địa chỉ byte cao và 3 là địa chỉ của byte thấp.



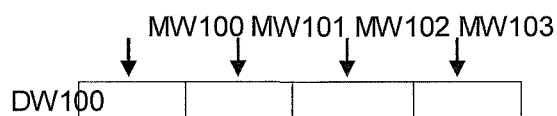
IW là từ đơn có địa chỉ 2

IB2 là byte có địa chỉ 2

IB3 là byte có địa chỉ 3

Trong trường hợp dữ liệu cần được lưu trữ mà một từ đơn không thể chứa hết được, PLC cho phép ghép 4 byte liên nhau được xem xét là một đơn vị nhớ và được gọi là từ kép (Double Word). Địa chỉ thấp nhất trong 4 byte nhớ này là địa chỉ của từ kép.

Ví dụ: Từ kép có địa chỉ là 100 thì các byte nhớ trong từ kép này có địa chỉ là 100, 101, 102, 103 trong đó 103 là địa chỉ byte thấp, 100 là địa chỉ byte cao.



Trong PLC bộ xử lý trung tâm có thể thực hiện một số thao tác như:

- Đọc nội dung các vùng nhớ (bit, byte, word, double word).
- Ghi dữ liệu vào vùng nhớ (bit, byte, word, double word).

Trong thao tác đọc, nội dung ban đầu của vùng nhớ không thay đổi mà chỉ lấy bản sao của dữ liệu để xử lý.

Trong thao tác ghi, dữ liệu được ghi vào trở thành nội dung của vùng nhớ và dữ liệu ban đầu bị mất đi.

Có hai loại bộ nhớ trong CPU của PLC:

- RAM (Random Access Memory): Bộ nhớ có thể đọc và ghi.
- ROM (Read Only Memory): Bộ nhớ chỉ đọc.

Bộ nhớ RAM:

Có một số lượng các ô nhớ xác định. Mỗi ô nhớ có 1 dung lượng nhớ cố định và nó chỉ tiếp nhận một lượng thông tin nhất định. Các ô nhớ được ký hiệu bằng các địa chỉ riêng của nó. Bộ nhớ này chứa các chương trình được sửa đổi hoặc các dữ liệu, kết quả tạm thời trong quá trình tính toán, lập trình.

Đặc điểm của bộ nhớ RAM là nội dung chứa trong các ô nhớ của nó bị mất đi khi mất nguồn điện.

Bộ nhớ ROM:

Chứa các thông tin không có khả năng xoá hoặc không thể thay đổi được, được nhà sản xuất sử dụng chứa các chương trình hệ thống. Chương trình trong bộ nhớ ROM có nhiệm vụ:

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

- Khi bị mất nguồn điện, bộ nhớ ROM vẫn giữ nguyên nội dung của nó và không bao giờ bị mất.

Bộ xử lý trung tâm

Bộ xử lý trung tâm (CPU – Central Processing Unit) điều khiển và quản lý tất cả các hoạt động bên trong PLC. Việc trao đổi thông tin giữa CPU, bộ nhớ và khối vào/ra được thực hiện thông qua hệ thống Bus dưới sự điều khiển của CPU. Một mạch dao động thạch anh cung cấp xung clock tần số chuẩn cho CPU, thường là 1 hay 8 MHz, tùy thuộc vào bộ xử lý sử dụng. Tần số xung clock xác định tốc độ hoạt động của PLC và được dùng để thực hiện sự đồng bộ cho tất cả phần tử trong hệ thống.

Hệ điều hành

Sau khi bật nguồn, hệ điều hành sẽ đặt các counter, timer và bit nhớ với thuộc tính non-retentive (không được nhớ bởi Pin dự phòng) cũng như accu về 0.

Để xử lý chương trình, hệ điều hành đọc từng dòng chương trình từ đầu đến cuối.

Tương ứng hệ điều hành thực hiện chương trình theo các câu lệnh.

Bit nhớ (memory bit)

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

Bộ đệm (Process Image)

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

Accumulator

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

Counter, Timer

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

Hệ thống Bus

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

2.2 Thiết bị điều khiển lập trình S7-200

S7 – 200 là thiết bị điều khiển lập trình loại nhỏ của hãng Siemens (CHLB Đức) có cấu trúc theo kiểu modul và có các modul mở rộng. Thành phần cơ bản

của S7 – 200 là khối vi xử lý **CPU 212** và **CPU 214**. Về hình thức bên ngoài, sự khác nhau của 2 loại CPU này nhờ số đầu vào/ra và nguồn cung cấp.

- **CPU 212** có 8 cổng vào và 6 cổng ra, có khả năng mở rộng thêm 2 mô đun.
- **CPU 214** có 14 cổng vào và 10 cổng ra, có khả năng mở rộng thêm 7 mô đun.

Trong tài liệu này chỉ đề cập đến CPU 214 là chủ yếu.

CPU 214 có những đặc điểm sau:

- 2048 từ nhớ chương trình (chứa trong ROM điện).
- 2048 từ nhớ dữ liệu (trong đó 256 từ chứa trong ROM điện).
- 14 ngõ vào và 19 ngõ ra digital kèm theo trong khối trung tâm.
- Hỗ trợ tối đa 7 mô đun mở rộng kể cả mô đun analog
- Tổng số cổng vào/ra cực đại là 64 cổng vào ra digital.
- 128 Timer chia làm 3 loại theo độ phân giải khác nhau: 4 Timer 1mS, 16 Timer 10mS, 108 Timer có độ phân giải là 100mS.
- 128 bộ đếm chia làm 2 loại: 96 đếm lên và 32 đếm lên xuống.
- 256 ô nhớ nội bộ.
- 688 ô nhớ đặt biệt dùng để thông báo trạng thái và đặt chế độ làm việc.
- Có phép tính số học.
- Ba bộ đếm tốc độ cao với nhịp 2 KHz và 7 KHz
- Hai bộ điều chỉnh tương tự.
- Toàn bộ vùng nhớ không bị mất dữ liệu trong khoảng thời gian 190 giờ khi PLC bị mất nguồn nuôi.

2.3. Địa chỉ các ngõ vào / ra

Địa chỉ ô nhớ trong S7 bao gồm hai phần: Phần chữ và phần số. Ví dụ:

PIW 304	hoặc	I0.0
Phần chữ Phần số		Phần chữ Phần số

2.3.1. hần chữ chỉ vị trí và kích thước của ô nhớ:

- M: Chỉ ô nhớ trong miền các biến cờ có kích thước là 1 bit
- MB: Chỉ ô nhớ trong miền các biến cờ có kích thước là 1 byte (8 bit).
- MW: Chỉ ô nhớ trong miền các biến cờ có kích thước là 2 byte (16 bit).
- MD: Chỉ ô nhớ trong miền các biến cờ có kích thước là 4 byte (32 bit).
- I: Chỉ ô nhớ có kích thước là 1 bit trong miền bộ đệm ngõ vào số.
- IB: Chỉ ô nhớ có kích thước là 1 byte trong miền bộ đệm ngõ vào số.
- IW: Chỉ ô nhớ có kích thước là 2 byte (1 từ) trong miền bộ đệm ngõ vào số.
- ID: Chỉ ô nhớ có kích thước là 4 byte (2 từ) trong miền bộ đệm ngõ vào số.
- Q: Chỉ ô nhớ có kích thước là 1 bit trong miền bộ đệm ngõ ra số.

- QB: Chỉ ô nhớ có kích thước là 1 byte trong miền bộ đếm ngõ ra số.
- QW: Chỉ ô nhớ có kích thước là 2 byte trong miền bộ đếm ngõ ra số.
- QD: Chỉ ô nhớ có kích thước là 4 byte trong miền bộ đếm ngõ ra số.
- T: Chỉ ô nhớ trong miền nhớ của bộ thời gian (Timer).
- C: Chỉ ô nhớ trong miền nhớ của bộ đếm (counter)
- PIB: Chỉ ô nhớ có kích thước là 1 byte thuộc vùng Peripheral Input, thường là địa chỉ cổng vào của các mô đun tương tự.
- PIW: Chỉ ô nhớ có kích thước là 2 byte thuộc vùng Peripheral Input, thường là địa chỉ cổng vào của các mô đun tương tự.
- PID: Chỉ ô nhớ có kích thước là 4 byte thuộc vùng Peripheral Input, thường là địa chỉ cổng vào của các mô đun tương tự.
- PQB: Chỉ ô nhớ có kích thước là 1 byte thuộc vùng Peripheral Output, thường là địa chỉ cổng ra của các mô đun tương tự.
- PQW: Chỉ ô nhớ có kích thước là 2 byte thuộc vùng Peripheral Output, thường là địa chỉ cổng ra của các mô đun tương tự.
- PQD: Chỉ ô nhớ có kích thước là 4 byte thuộc vùng Peripheral Output, thường là địa chỉ cổng ra của các mô đun tương tự.
- PQB: Chỉ ô nhớ có kích thước là 1 byte thuộc vùng Peripheral Output, thường là địa chỉ cổng ra của các mô đun tương tự.
- DBX: Chỉ ô nhớ có kích thước là 1 bit trong khối dữ liệu DB, được mở bằng lệnh OPN DB (Open Data Block).
- DBB: Chỉ ô nhớ có kích thước là 1 byte trong khối dữ liệu DB, được mở bằng lệnh OPN DB (Open Data Block).
- DBW: Chỉ ô nhớ có kích thước là 2 byte trong khối dữ liệu DB, được mở bằng lệnh OPN DB (Open Data Block).
- DBD: Chỉ ô nhớ có kích thước là 4 byte trong khối dữ liệu DB, được mở bằng lệnh OPN DB (Open Data Block).
- DBx.DBX: Chỉ trực tiếp ô nhớ có kích thước là 1 bit trong khối dữ liệu DBx, với x là chỉ số của khối DB. Ví dụ: DB3.DBX1.5
- DBx.DBB: Chỉ trực tiếp ô nhớ có kích thước là 1 byte trong khối dữ liệu DBx, với x là chỉ số của khối DB. Ví dụ: DB4.DBB1.
- DBx.DBW: Chỉ trực tiếp ô nhớ có kích thước là 2 byte trong khối dữ liệu DBx, với x là chỉ số của khối DB. Ví dụ: DB5.DBW1.
- DBx.DBD: Chỉ trực tiếp ô nhớ có kích thước là 4 byte trong khối dữ liệu DBx, với x là chỉ số của khối DB. Ví dụ: DB5.DBD1.
- DIX: Chỉ ô nhớ có kích thước là 1 bit trong khối dữ liệu DB, được mở bằng lệnh OPN DI: (Open instance data block).
- DIB: Chỉ ô nhớ có kích thước là 1 byte trong khối dữ liệu DB, được mở bằng lệnh OPN DI (Open instance data block).
- DIW: Chỉ ô nhớ có kích thước là 2 byte trong khối dữ liệu DB, được mở bằng lệnh OPN DI (Open instance data block).
- DID: Chỉ ô nhớ có kích thước là 4 byte trong khối dữ liệu DB, được mở

bằng lệnh OPN DI (Open instance data block).

2.3.2. Phần số chỉ địa chỉ của byte hoặc bit trong miền nhớ đã xác định:

Nếu ô nhớ đã được xác định thông qua phần chữ là có kích thước 1 bit thì phần số sẽ là địa chỉ của byte và số thứ tự của bit trong byte đó, được tách với nhau bằng dấu chấm. Ví dụ:

I 0.0: Chỉ bit 0 của byte 0 trong miền nhớ bộ đệm ngõ vào số PII. Q 4.1: Chỉ bit 1 của byte 4 của miền nhớ bộ đệm ngõ ra số PIQ. M 10.5: Chỉ bit 5 của byte 10 trong miền các biến cờ M.

Trong trường hợp ô nhớ đã được xác định là byte, từ hoặc từ kép thì phần số sẽ là địa chỉ của byte đầu tiên trong mảng byte của ô nhớ đó.

Ví dụ: DIB 15: Chỉ ô nhớ có kích thước 1 byte (byte 15) trong khối DB đã được mở bằng lệnh OPN DI.

DIW 18: Chỉ ô nhớ có kích thước 1 từ gồm 2 byte 18 và 19 trong khối DB đã được mở bằng lệnh OPN DB.

DB2.DBW15: Chỉ ô nhớ có kích thước 2 byte 15 và 16 trong khối dữ liệu DB2.

M 105: Chỉ ô nhớ có kích thước 2 từ gồm 4 byte 105, 106, 107, 108 trong miền nhớ các biến cờ M.

2.4 Cấu trúc bộ nhớ của S7 – 200

Bộ nhớ của S7 – 200 được chia làm 3 vùng: vùng nhớ chương trình, vùng nhớ dữ liệu và vùng nhớ thông số. Vùng nhớ chương trình, vùng nhớ thông số và một phần vùng nhớ dữ liệu được chứa trong ROM điện EEPROM. Đối với CPU cho phép cắm thêm khối nhớ mở rộng để chứa chương trình mà không cần đến thiết bị lập trình. Phần sau đây mô tả chi tiết về các vùng nhớ.

Vùng nhớ chương trình

Vùng nhớ chương trình chứa các chỉ thị điều khiển vi xử lý để thực hiện yêu cầu điều khiển, chương trình ứng dụng sau khi soạn thảo được nạp vào ROM và vẫn tồn tại khi mất điện.

Vùng nhớ thông số

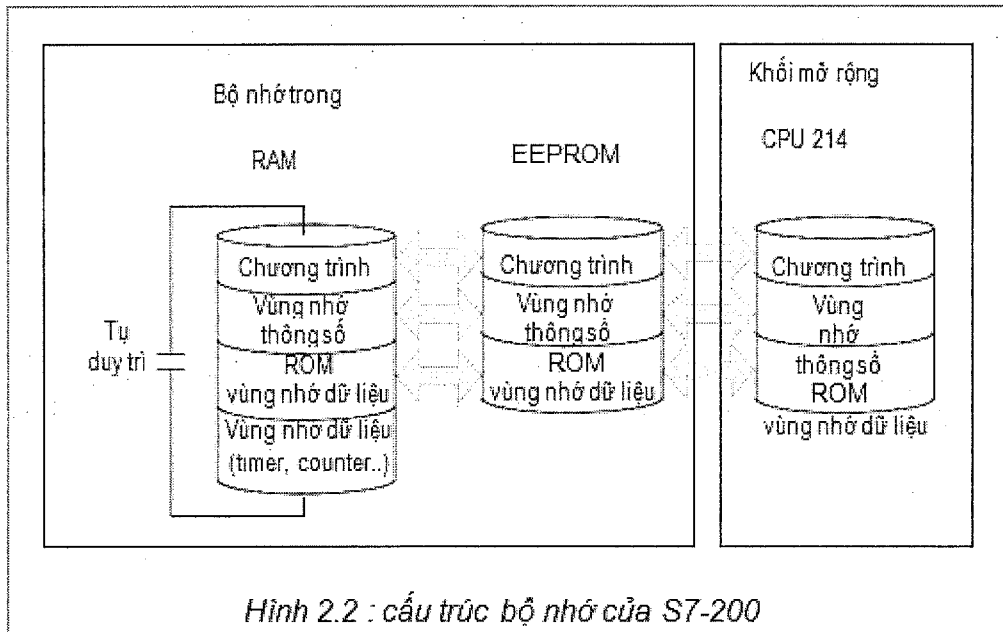
Gồm các ô nhớ chứa các thông số cài đặt, mật khẩu, địa chỉ thiết bị điều khiển và các thông tin về các vùng trống có thể sử dụng. Nội dung của vùng nhớ này được chứa trong ROM giống như vùng chương trình.

Vùng nhớ dữ liệu

Vùng nhớ dữ liệu là nơi làm việc, vùng này gồm các địa chỉ để lưu trữ các phép tính, lưu trữ tạm thời các kết quả trung gian, và chứa các hằng số được sử dụng trong các chỉ dẫn hoặc các thông số điều chỉnh khác. Ngoài ra trong vùng

này còn có các phần tử và đối tượng như: Bộ định thời, bộ đếm, các bộ đếm tốc độ cao và các ngõ vào/ra analog. Một phần tử của vùng nhớ dữ liệu được chứa trong ROM, vì vậy các hằng số cũng như các thông tin khác vẫn được duy trì khi mất điện giống như trong vùng nhớ chương trình. Một phần khác được chứa trong RAM, nội dung trong RAM cũng được duy trì trong khoảng thời gian nhất định khi mất điện bằng một điện dung có độ rỉ thấp.

Vùng dữ liệu gồm các ô biến, vùng đệm của các ngõ vào/ra, vùng nhớ trong và vùng nhớ đặc biệt. Phạm vi của vùng nhớ rất linh hoạt và cho phép đọc cũng như ghi trên toàn bộ vùng nhớ, ngoại trừ một vài ô nhớ đặc biệt chỉ cho phép đọc, các dạng dữ liệu cho phép trong vùng là: Bit, byte, word hoặc double word



2.5. Xử lý chương trình

2.5.1. Chương trình

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

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

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

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

2.5.2. Chương trình của S7 – 200

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

- STEP7 – Micro/DOS
- STEP7 – Micro/WIN

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

Các chương trình cho S7 – 200 phải có cấu trúc bao gồm chương trình chính (main program) và sau đó đến các chương trình con và các chương trình xử lý ngắt được chỉ ra sau đây:

- Chương trình chính được kết thúc bằng lệnh kết thúc chương trình (MEND).
- Chương trình con là một bộ phận của chương trình. Các chương trình con phải được viết sau lệnh kết thúc chương trình chính, đó là lệnh MEND.
- Các chương trình xử lý ngắt là một bộ phận của chương trình. Nếu cần sử dụng chương trình xử lý ngắt phải được viết sau lệnh kết thúc chương trình chính MEND.

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 đằng sau chương trình chính.

<pre>MainProgram : MEND</pre>	Thực hiện trong 1 vòng quét
<pre>SBR 0 Chương trình con thứ 1 : RET</pre>	Thực hiện khi được chương trình chính gọi
<pre>SBR n Chương trình con thứ n+1 : RET</pre>	
<pre>INT 0 Chương trình xử lý ngắt thứ 1 : RETI</pre>	Thực hiện khi có tín hiệu báo ngắt
<pre>INT n Chương trình xử lý ngắt thứ n+1 : RETI</pre>	

2.5.3 Phương pháp lập trình

Cách lập trình cho S7 – 200 nói riêng và cho các PLC hãng Seimens nói chung dựa trên 3 phương pháp cơ bản: **Phương pháp hình thang** (Ladder Logic viết tắt là **LAD**) và **Phương pháp liệt kê lệnh** (Statement List viết tắt là **STL**). Chương này sẽ giới thiệu các thành phần cơ bản của 2 phương pháp trên và các sử dụng chúng trong lập trình. Ngoài ra, còn có **Phương pháp lập trình theo sơ đồ khối** (Function Block Diagramm FBD) nhưng chỉ có trong Version 3.0 của phần mềm STEP 7.

Nếu chương trình được viết theo kiểu LAD, thiết bị lập trình sẽ tự tạo ra một chương trình theo kiểu STL tương ứng. Ngược lại không phải mọi chương trình được viết theo kiểu STL cũng có thể chuyển sang được dạng LAD.

Bộ lệnh của phương pháp STL được trình bày đều có một chức năng tương ứng với mỗi tiếp điểm, các cuộn dây và các hộp dùng trong LAD. Những lệnh này phải đọc và phối hợp được trạng thái của các tiếp điểm để đưa ra một quyết định về giá trị trạng thái đầu ra hoặc một giá trị logic cho phép, hoặc không cho phép thực hiện chức năng của một (hay nhiều) hộp. Để dễ dàng làm quen với các thành phần cơ bản của LAD và của STL cần nắm được các định nghĩa cơ bản sau đây:

Định nghĩa về 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 dùng role. 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 role. Các tiếp điểm đó có thể là thường đóng $\overline{+}$ hay thường $\overline{-}$.

- **Cuộn dây (coil):** là biểu tượng mô tả role được mắc theo chiều dòng điện cung cấp cho rơ le.

- **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 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 các mạch hoàn thiện, đi từ đường nguồn bên trái đến đường nguồn bên phải. Đường nguồn bên trái là dây nóng, đường nguồn bên phải là dây trung hoà (neutral) hay là đường trở về nguồn cung cấp.

Định nghĩa về STL: 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 câu lệnh hình thức biểu diễn một

chức năng của PLC.

Định nghĩa về ngăn xếp logic (logic stack)

Stack 0 – bit đầu tiên hay bit trên cùng của ngăn xếp.

Stack 1 – bit thứ hai của ngăn xếp.

Stack 2 – bit thứ ba của ngăn xếp.

Stack 3 – bit thứ tư của ngăn xếp.

Stack 4 – bit thứ năm của ngăn xếp.

Stack 5 – bit thứ sáu của ngăn xếp.

Stack 6 – bit thứ bảy của ngăn xếp.

Stack 7 – bit thứ tám của ngăn xếp.

Stack 8 – bit thứ chín của ngăn xếp.


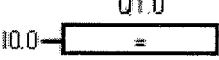
S0
S1
S2
S3
S4
S5
S6
S7
S8

Để tạo ra được một chương trình dạng STL, người lập trình cần phải hiểu rõ phương thức sử dụng 9 bit ngăn xếp logic của S7 – 200. Ngăn xếp logic là một khối gồm 9 bit chồng lên nhau. Tất cả các thuật toán liên quan đến ngăn xếp đều chỉ làm việc với bit đầu tiên hoặc với bit đầu và bit thứ hai của ngăn xếp. Giá trị logic mới đều có thể được gởi (hoặc được nối thêm) vào ngăn xếp. Khi phối hợp hai bit đầu tiên của ngăn xếp, thì ngăn xếp sẽ được kéo lên một bit. Ngăn xếp và tên của từng bit trong ngăn xếp được biểu diễn trong hình trên.

Định nghĩa về FBD : Phương pháp sơ đồ khối sử dụng các “ Hộp ” cho từng chức năng. Ký tự trong hộp cho biết chức năng (thí dụ ký tự & là phép toán logic AND) Ngôn ngữ lập trình này có ưu điểm là 1 người “ không chuyên lập trình “ như 1 kỹ thuật viên công nghệ cũng có thể sử dụng dạng thảo này.

Ví dụ về ladder logic và statement list:

Hình mô tả việc thực hiện lệnh LD (viết tắt của từ tiếng anh Load) đưa giá trị logic của tiếp điểm I0.0 vào trong ngăn xếp theo cách biểu diễn của LAD, STL và FBD.

LAD	STL	FBD
	<pre>LD I0.0 = Q1.0</pre>	

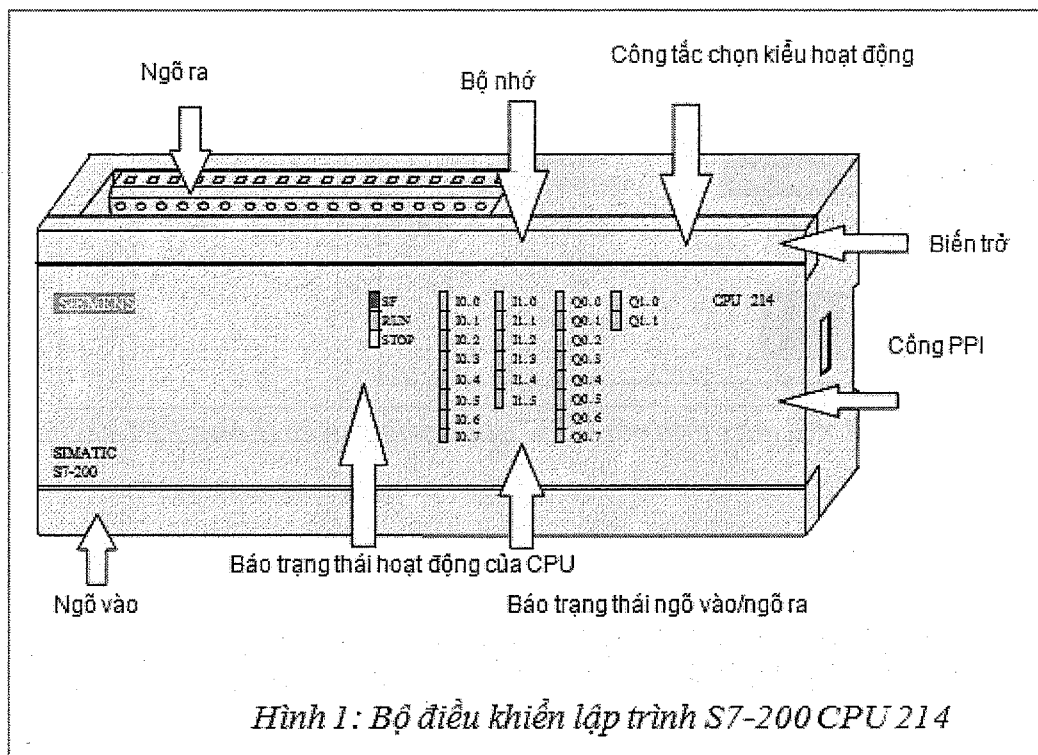
Bài 3. KẾT NỐI DÂY GIỮA PLC VÀ THIẾT BỊ NGOẠI VI

3.1 Kết nối dây giữa PLC và các thiết bị ngoại vi

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

3.1.1. Giới thiệu CPU 214 và cách kết nối với thiết bị ngoại vi.

Sơ đồ bề mặt của bộ điều khiển lập trình S7-200 CPU 214 được cho như hình 3.1

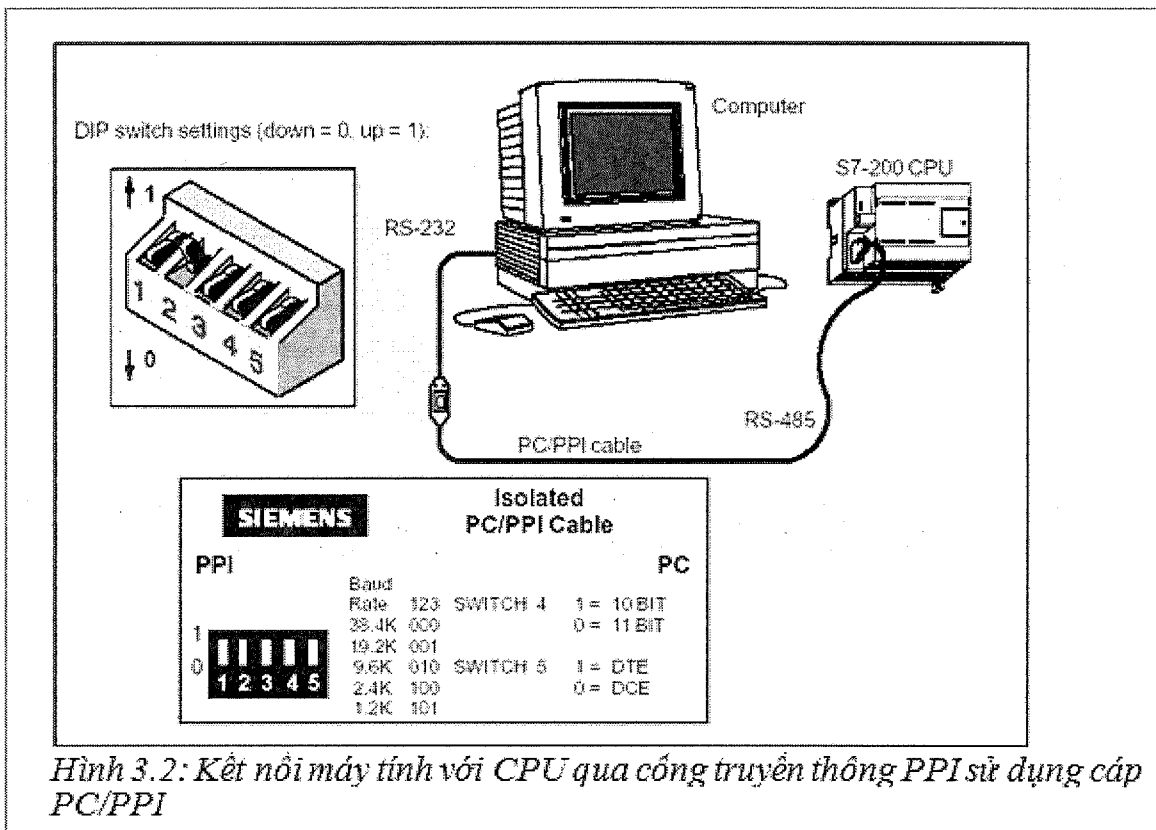


Để cho bộ điều khiển lập trình này hoạt động được thì người sử dụng phải kết nối PLC với nguồn cung cấp và các ngõ vào ra của nó với thiết bị ngoại vi. Muốn nạp chương trình vào CPU, người sử dụng phải soạn thảo chương trình bằng các thiết bị lập trình hoặc máy tính với phần mềm tương ứng cho loại PLC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

vậy, để hệ thống điều khiển khiển bằng PLC hoạt động cũng như lập trình cho nó, cần phải kết nối PLC với máy tính cũng như các ngõ vào ra với ngoại vi.

3.1.1.1 Kết nối với máy tính

Đối với các thiết bị lập trình của hãng Siemens có các cổng giao tiếp PPI thì có thể kết nối trực tiếp với PLC thông qua một sợi cáp. Tuy nhiên đối với máy tính cá nhân cần thiết phải có cáp chuyển đổi PC/PPI. Sơ đồ nối máy tính với CPU thuộc họ S7- 200 được cho như hình 3.2.



Tùy theo tốc độ truyền giữa máy tính và CPU mà các công tắc 1,2,3 được để ở vị trí thích hợp. Thông thường đối với CPU 214 thì tốc độ truyền thường đặt là 9,6 KBaud (tức công tắc 123 được đặt theo thứ tự là 010).

Tùy theo truyền thông là 10 Bit hay 11 Bit mà công tắc 4 được đặt ở vị trí thích hợp. Khi kết nối bình thường với máy tính thì công tắc 4 chọn ở chế độ truyền thông 11 Bit.

Công tắc 5 ở cáp PC/PPI được sử dụng để kết nối port truyền thông RS-232 của một modem với S7-200 CPU. Khi kết nối bình thường với máy tính thì công tắc 5 được đặt ở vị trí data Communications Equipment (DCE). Khi kết nối cáp PC/PPI với một modem thì port RS-232 của cáp PC/PPI được đặt ở vị trí Data Terminal Equipment (DTE).

3.1.1.2 Kết nối vào/ra với ngoại vi

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

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

* Nối nguồn cung cấp cho CPU

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

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

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

Một chiều: 20,4 ... 28,8 VDC

* Kết nối các ngõ vào số với ngoại vi

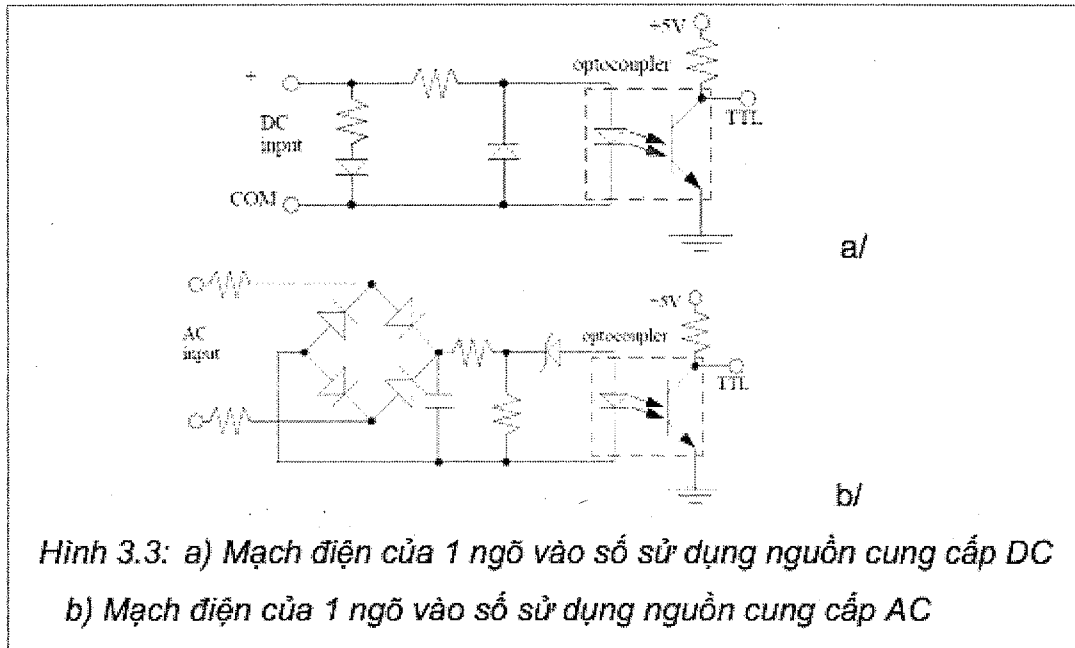
Các ngõ vào của PLC có thể được chế tạo là một khối riêng, hoặc kết hợp với các ngõ ra chung trong một khối hoặc được tích hợp trên khối CPU. Trong trường hợp nào cũng vậy, các ngõ vào cũng phải được cung cấp nguồn riêng với cấp điện áp tùy thuộc vào loại ngõ vào. Cần lưu ý trong một khối ngõ vào cũng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

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

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

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

Sơ đồ mạch điện bên trong của các ngõ vào được cho như hình 3.3a,b



Hình 3.3: a) Mạch điện của 1 ngõ vào số sử dụng nguồn cung cấp DC
 b) Mạch điện của 1 ngõ vào số sử dụng nguồn cung cấp AC

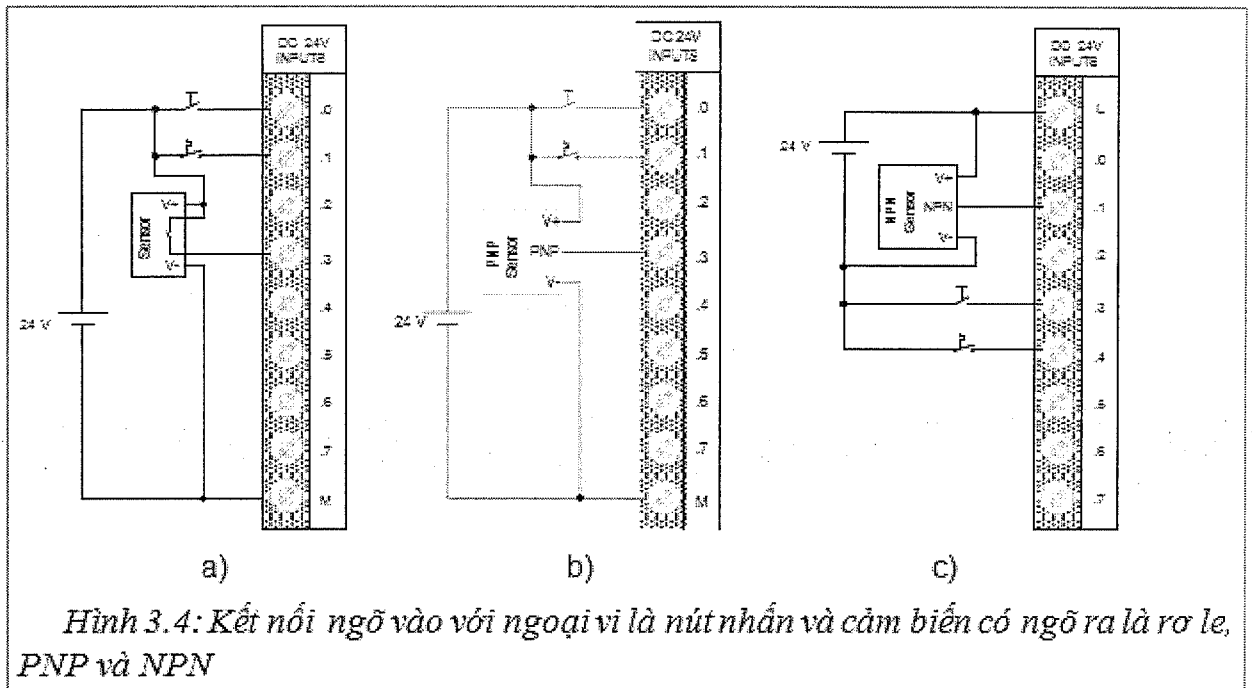
Tùy theo yêu cầu mà có thể quyết định sử dụng loại ngõ vào nào.

- + Ngõ vào DC:
 - Điện áp DC thường thấp do đó an toàn hơn.
 - Đáp ứng ngõ vào DC rất nhanh.
 - Điện áp DC có thể được kết nối với nhiều phần tử hệ thống điện.
- + Ngõ vào AC:
 - Ngõ vào AC yêu cầu cần phải có thời gian. Ví dụ đối với điện áp có tần số 50 Hz phải yêu cầu thời gian đến 1/50 giây mới nhận biết được.
 - Tín hiệu AC ít bị nhiễu hơn tín hiệu DC, vì vậy chúng thích hợp với khoảng cách lớn và môi trường nhiễu (từ).
 - Nguồn AC kinh tế hơn.
 - Tín hiệu AC thường được sử dụng trong các thiết bị tự động hiện hữu.

Đối với các ngõ vào số, khi kết nối với ngoại vi, ngoại trừ các trường hợp đặc biệt thì thông thường mỗi một ngõ vào được kết nối với một bộ tạo tín hiệu nhị phân như: nút nhấn, công tắc, cảm biến tiếp cận . . . Hình 3.4a, b, c minh họa cách kết nối dây các ngõ vào PLC với các bộ tạo tín hiệu nhị phân khác nhau.

Trong ví dụ hình 3.4a có 3 ngõ vào, một là nút nhấn thường hở, hai là tiếp điểm của rơ le nhiệt, và ba là cảm biến tiếp cận với ngõ ra là role. Cả ba bộ tạo tín hiệu này được cung cấp bởi một nguồn 24VDC. Khi tiếp điểm hở hoặc cảm biến phát tín hiệu "0" thì không có điện áp tại các ngõ vào. Nếu các tiếp điểm

được đóng lại hoặc cảm biến phát tín hiệu “1” thì ngõ vào được cấp điện.



Đối với các ngõ vào của CPU 24V DC/DC/DC, theo số tay được kết nối như hình 3.7.

* Kết nối các ngõ ra số với ngoại vi

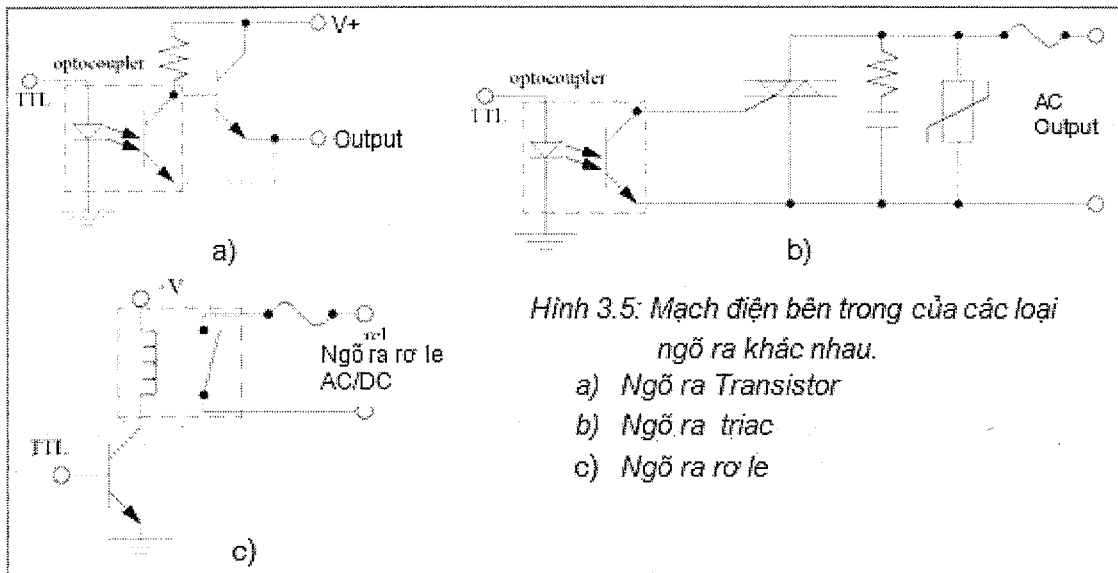
Các ngõ ra của PLC có thể được chế tạo là một khối riêng, hoặc kết hợp với các ngõ ra chung trong một khối hoặc được tích hợp trên khối CPU. Trong trường hợp nào cũng vậy, các ngõ ra cũng phải được cung cấp nguồn riêng với cấp điện áp tùy thuộc vào loại ngõ ra. Cần lưu ý trong một khối ra cũng như các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

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

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

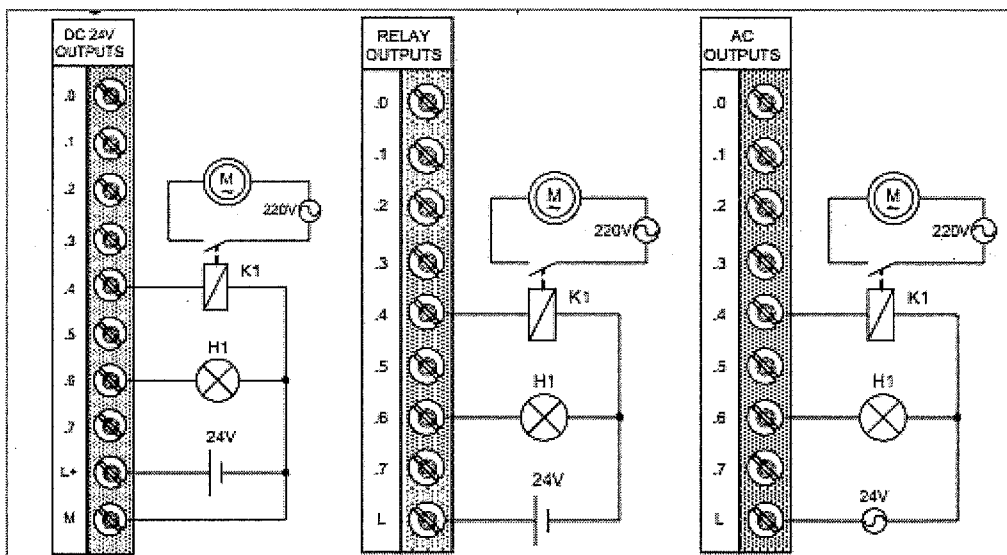
Các khối ra tiêu chuẩn của PLC thường có 8 đến 32 ngõ ra theo cùng loại và có dòng định mức khác nhau. Ngõ ra có thể là rơ le, transistor hoặc triac. Rơ le là ngõ ra linh hoạt nhất. Chúng có thể là ngõ ra AC và DC. Tuy nhiên đáp ứng của ngõ ra rơ le chậm, giá thành cao và bị hư hỏng sau vài triệu lần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

Sơ đồ mạch điện bên trong của các ngõ ra được cho như hình 3.5



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

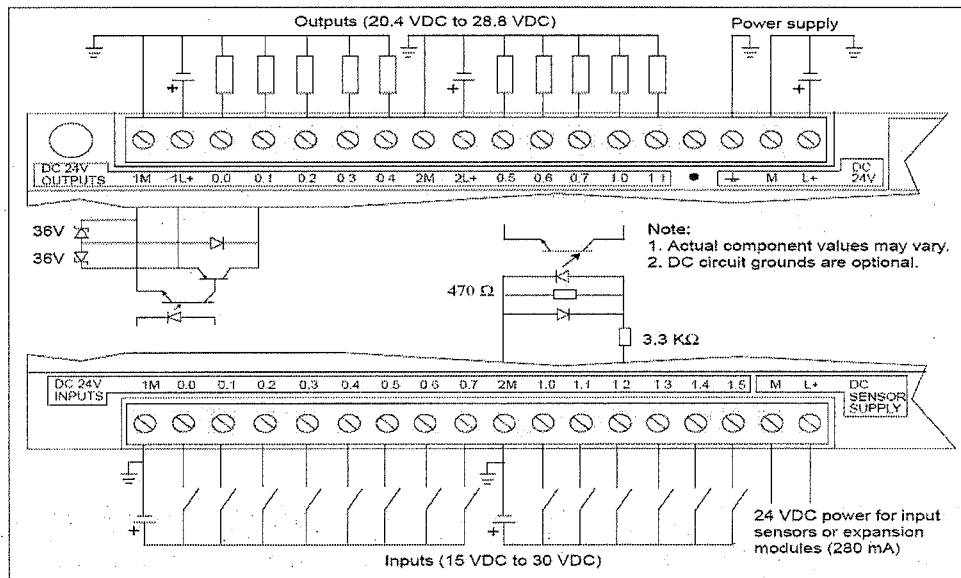
Đối với các ngõ ra số, khi kết nối với ngoại vi, ngoại trừ các trường hợp đặc biệt thì thông thường mỗi một ngõ ra được kết nối với một đối tượng điều khiển nhận tín hiệu nhị phân như: đèn báo, cuộn dây rơ le, chuông báo . . . Hình 3.6 minh họa cách kết nối dây các ngõ ra PLC với các cơ cấu chấp hành.



Hình 3.6a là một ví dụ cho các khối ra sử dụng 24VDC với mass chung. Tiêu biểu cho loại này là ngõ ra transistor. Trong ví dụ này các ngõ ra được kết nối với tải công suất nhỏ là đèn báo và cuộn dây rơ le. Quan sát mạch kết nối

này, đèn báo sử dụng nguồn cung cấp là 24VDC. Nếu ngõ ra .6 ở mức logic “1” (24VDC) thì dòng sẽ chảy từ ngõ ra .6 qua đèn H1 và xuống Mass (M), đèn sáng. Nếu ngõ ra ở mức logic “0” (0V), thì đèn H1 tắt. Nếu ngõ ra .4 ở mức logic “1” thì cuộn dây rơ le có điện, làm tiếp điểm của nó đóng lại cung cấp điện 220 VAC cho động cơ.

Hình 3.6b là một ví dụ ngõ ra rơ le sử dụng nguồn cấp là 24 VDC, và hình 3.6c là ví dụ ngõ ra triac sử dụng nguồn xoay chiều 24 Vac.



Hình 3.7: Cách kết nối ngõ vào/ra của CPU 214 DC/DC/DC với ngoại vi

Một chú ý quan trọng khi kết nối các ngõ ra cần tra cứu số tay khối ngõ ra hiện có để có được thông tin chính xác tránh được những sự cố đáng tiếc xảy ra. Hình 3.7 là ví dụ của CPU 214 với nguồn cung cấp DC, ngõ vào DC và ngõ ra DC được nối dây với ngoại vi (trích từ sổ tay S7-200 Programmable Controller System Manual). Ta nhận thấy mỗi một nhóm ngõ vào cũng như một nhóm ngõ ra và CPU được cung cấp nguồn riêng là 24 VDC. Ngoài ra trên khối CPU còn có nguồn phụ 24 VDC (đến 280 mA) có thể được sử dụng để cung cấp cho các cảm biến hoặc khối mở rộng.

3.1.2. Ví dụ kết nối ngõ vào/ra của PLC từ một sơ đồ điều khiển có tiếp điểm.

Trong nhiều trường hợp, cần cải tạo một hệ thống điều khiển với rơ le và contactor thành hệ thống điều khiển với PLC. Một câu hỏi đặt ra là chúng ta cần giữ lại những phần nào trong hệ thống điều khiển, còn phần nào sẽ loại bỏ đi?

Để dễ dàng trong việc chuyển đổi, có thể áp dụng phương pháp sau để chuyển đổi từ một hệ thống điều khiển cũ sang điều khiển với PLC:

- **Về phần cứng:**

- Xác định các bộ tạo tín hiệu (ví dụ: nút nhấn, công tắc, cảm biến . . .) cần thiết nhất trong hệ thống điều khiển, mỗi bộ tạo tín hiệu tùy theo loại

tạo ra tín hiệu nào nên được kết nối với một ngõ vào của PLC tương ứng, ví dụ nếu bộ tạo ra tín hiệu nhị phân thì được kết nối với các ngõ vào số, còn bộ tạo ra tín hiệu tương tự thì kết nối với ngõ vào tương tự (ngõ vào analog). Còn các bộ tạo tín hiệu còn lại nếu không cần thiết thì có thể bỏ đi và sẽ được thực hiện bằng chương trình trong PLC.

- Tương tự xác định các cơ cấu chấp hành (đối tượng điều khiển) cần thiết nhất, thông thường các đối tượng này là các đèn báo, contactor chính, van từ, .v.v.. Tùy theo loại mà mỗi đối tượng điều khiển có thể kết nối trực tiếp hoặc gián tiếp với các ngõ ra tương ứng, mỗi một đối tượng điều khiển cần một ngõ ra. Nếu các đối tượng điều khiển cần dòng điều khiển lớn thì yêu cầu phải sử dụng rơ le trung gian. Ví dụ như các contactor chính điều khiển các động cơ công suất lớn thì ngõ ra của PLC sẽ được nối với một rơ le trung gian và thông qua tiếp điểm của rơ le trung gian để điều khiển các contactor này. Còn các đối tượng điều khiển không tác động trực tiếp đến quá trình điều khiển mà chỉ đóng vai trò trung gian hỗ trợ cho quá trình điều khiển như rơ le trung gian thì có thể loại bỏ và được thay thế bằng một ô nhớ nào đó trong chương trình của PLC.
- Sau khi đã xác định được số lượng các ngõ vào, ngõ ra cần thiết và hệ thống điện cung cấp cho phần điều khiển thì tiến hành đến việc lựa chọn loại PLC phù hợp.
- Thiết lập bảng xác định các ngõ vào/ra với các ngoại vi tương ứng và chú ý ghi chú lại càng chi tiết càng tốt.
- Thực hiện việc nối dây các ngõ vào, ngõ ra của PLC với các bộ tạo tín hiệu điều khiển và đối tượng điều khiển. Trong quá trình nối dây cần lưu ý đến các nguyên tắc an toàn trong hệ thống điều khiển
- Tất cả việc kết nối dây trong hệ thống điều khiển trước đây sẽ được biến đổi thành chương trình trong PLC.

• Về phần mềm:

Việc viết chương trình có thể thực hiện theo hai cách:

Cách 1: Tùy theo yêu cầu công nghệ mà có thể thiết lập giải thuật điều khiển và viết chương trình theo giải thuật điều khiển này.

Cách 2: Vẫn duy trì hoạt động của hệ thống như cũ, hay nói khác đi là không cần thiết phải lập lại giải thuật điều khiển vì tất cả đã được thiết kế trong sơ đồ điều khiển cứng trước đây mà chỉ cần biến đổi sơ đồ điều khiển này thành chương trình trong PLC. Cách này tương đối dễ dàng và có thể không bị lỗi khi lập trình.

Trong phần này trình bày phương pháp chuyển đổi theo cách 2 theo các bước như sau:

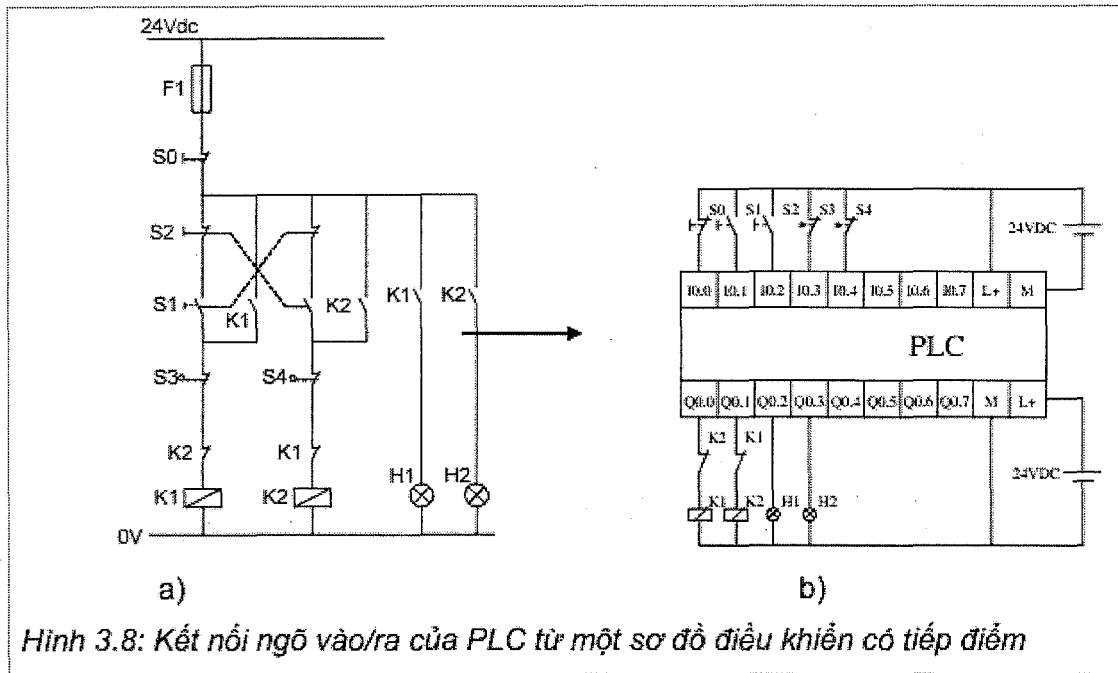
- Thực hiện viết chương trình lần lượt cho mỗi đối tượng điều khiển, mỗi đối tượng điều khiển được viết ở một đoạn chương trình và có ghi chú cụ thể để dễ dàng sửa lỗi.
- Chỉ có các điều kiện cần thiết nhất cho đối tượng điều khiển mới được viết vào đoạn chương trình điều khiển nó.
- Nếu một số đối tượng điều khiển có cùng chung một nhóm điều kiện, thì nhóm điều kiện này nên được viết riêng ở một đoạn chương trình và gán kết quả vào một ô nhớ trong PLC. Nếu đối tượng điều khiển nào cần nhóm điều kiện này thì chỉ cần lấy kết quả được chứa trong ô nhớ. Điều này giúp cho cấu trúc chương trình mạch lạc và việc đọc chương trình trở nên dễ dàng hơn.
- Các đối tượng điều khiển không cần thiết (ví dụ contactor trung gian) sẽ được thay thế bằng một ô nhớ trong PLC. Nếu các đối tượng điều khiển nào cần đến tiếp điểm của rơ le trung gian này thì chỉ cần thay thế bằng tiếp điểm của ô nhớ.
- Tùy theo hệ thống điều khiển có phức tạp hay không mà có thể phân chia thành nhiều khối chương trình để dễ dàng trong quá trình quản lý.

Hình 3.8 là một ví dụ về việc chuyển đổi một sơ đồ điều khiển cửa ra vào cơ quan bằng contactor thành hệ thống điều khiển với PLC (chỉ dừng lại ở việc chuyển đổi kết nối dây, còn chương trình thực hiện ở các chương sau).

Dựa vào các bước trên, ta nhận thấy các nút nhấn, contactor cần thiết được giữ lại như trong bảng xác định kết nối vào/ra với ngoại vi và PLC được chọn ở đây là loại CPU 214 DC/DC/relay. Do contactor K1 và K2 không được phép có điện đồng thời nên theo quan điểm an toàn cần phải khóa chéo hai contactor này lại với nhau.

Bảng xác định kết nối vào/ra với ngoại vi

<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
S0	I0.0	Nút nhấn dừng, thường đóng
S1	I0.1	Nút nhấn mở cửa, thường hở
S2	I0.2	Nút nhấn đóng cửa, thường hở
S3	I0.3	Công tắc hành trình giới hạn cửa mở, thường đóng
S4	I0.4	Công tắc hành trình giới hạn cửa đóng, thường đóng
K1	Q0.0	Cuộn dây contactor K1, điều khiển mở cửa
K2	Q0.1	Cuộn dây contactor K2, điều khiển đóng cửa
H1	Q0.2	Đèn báo cửa đang mở
H2	Q0.3	Đèn báo cửa đang đóng



3.2. Kiểm tra việc kết nối dây bằng phần mềm

Một công việc quan trọng cho người lắp đặt và vận hành là biết được các kết nối của các ngõ vào/ra với ngoại vi có đúng hay không trước khi nạp chương trình điều khiển vào CPU. Hoặc khi một hệ thống đang hoạt động bình thường nhưng một sự cố hư hỏng xảy ra thì các phần ngoại vi nào bị hư và phát hiện nó bằng cách nào. Trong phần mềm Step 7 Micro/Win có trang bị thêm phần này đó là mục **Status Chart**.

a/ Status Chart

Chúng ta có thể sử dụng Status Chart để đọc, ghi hoặc cưỡng bức các biến trong chương trình.

Để có thể truy xuất Status Chart, ta nhấp đúp chuột vào biểu tượng Status Chart trên màn hình.


b/ Đọc và thay đổi biến với Status Chart


Hình dưới đây chỉ một ví dụ về cách sử dụng Status Chart. Để đọc hay ghi các biến chúng ta thực hiện theo các bước sau:


Status Chart				
2 3 4 5 6 7				
	Address	Format	Current Value	New Value
1	I0.0	Bit	2#0	
2	I0.1	Bit	2#0	
3	Q0.0	Bit	2#1	
4		Signed		
5		Signed		
6		Signed		
7		Signed		

Bước 1: Ở ô đầu tiên trong Address chúng ta nhập vào địa chỉ hay tên ký hiệu của 1 biến trong chương trình ứng dụng mà bạn muốn đọc hoặc ghi, sau đó nhấn Enter. Lập lại bước này cho tất cả các biến mà bạn muốn thêm vào biểu đồ.

Bước 2: Nếu biến là một Bit (ví dụ : I, Q hoặc M), thì kiểu biến đặt ở cột Format là Bit. Nếu biến là một Byte, Word hay Double Word thì chọn ở cột Format và nhấp đúp chuột để tìm kiểu biến mong muốn.

Bước 3 : Để xem giá trị hiện hành của một biến trong PLC trong biểu đồ, hãy nhấp chuột vào biểu tượng 

Bước 4: Để dừng việc cập nhật trạng thái thì nhấp chuột vào biểu tượng  trở về vị trí cũ.



Bước 5: Để thay đổi 1 giá trị, hãy nhập một giá trị mới vào cột „New Value“ và nhấp chuột vào biểu tượng  để ghi giá trị vào CPU.


Cưỡng bức biến với Status Chart:


Để cưỡng bức biến trong Status Chart với 1 giá trị xác định, thực hiện các bước sau:


Bước 1: Chọn một ô trong cột Address, vào địa chỉ hay tên của biến cần cưỡng bức.

Bước 2: Nếu biến là một bit (ví dụ: I0.0, Q0.1), thì kiểu biến ở cột Format luôn luôn là bit. Nếu biến là một Byte, Word hay Double Word thì chọn ở cột Format và nhấp đúp chuột để tìm kiểu biến mong muốn.

Bước 3: Để cưỡng bức biến với giá trị hiện hành, trước tiên hãy đọc giá trị hiện hành trong PLC bằng cách nhấp đúp chuột vào biểu tượng . Nhấp hoặc cuộn ô chứa giá trị hiện hành mà bạn muốn cưỡng bức. Nhấp chuột vào biểu tượng  ở trên vị trí giá trị hiện hành để cưỡng bức biến giá trị đó.

Bước 4: Để cưỡng bức một giá trị mới cho một biến, nhập giá trị vào cột “New Value” và nhấp chuột vào biểu tượng .

Bước 5: Để xem giá trị hiện hành của tất cả các biến bị cưỡng bức, kích chuột vào biểu tượng Read Force .

Bước 6: Để cho tất cả các biến trở lại trạng thái bình thường, hãy kích chuột vào biểu tượng Unforce All .

3.3. Cài đặt và sử dụng phần mềm STEP 7 – Micro/win 32

+ Những yêu cầu đối với máy tính PC”

Máy tính cá nhân PC, muốn cài đặt được phần mềm STEP 7-Micro/Win phải thoả mãn những yêu cầu sau đây:

- 640 Kbyte RAM (ít nhất phải có 500 Kbyte bộ nhớ còn trống).

- Màn hình 24 dòng , 80 cột ở chế độ văn bản.
- Còn khoảng 2 Mbyte 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 trực tiếp giữa máy tính và PLC. (truyền thông online)

Truyền thông giữa STEP 7-Micro/Win với S7-200 CPU qua cổng truyền thông ở phía đáy của PLC. Sử dụng cáp có bộ chuyển đổi RS232-RS485, được gọi là cáp PC/PPI, để nối máy tính với PLC tạo thành mạch truyền thông trực tiếp.

Cắm 1 đầu của cáp PC/PPI với cổng truyền thông 9 chân của PLC, còn đầu kia với cổng truyền thông nối tiếp RS-232C của máy PC. Nếu máy PC có cổng truyền thông nối tiếp

RS232 với 25 chân, thì phải ghép nối qua bộ chuyển đổi 25 chân/9 chân để có thể ghép nối với cáp truyền thông PC/PPI.

+ Cài đặt phần mềm lập trình STEP 7-Micro/Win 32.

Sau khi kiểm tra bộ nhớ, ổ đĩa cứng hoàn toàn có đủ khả năng để cài phần mềm STEP 7- Micro/Win vào ổ cứng, thì lần lượt tiến hành các bước:

- 1/ Chèn đĩa CD vào ổ CD máy tính.
- 2/ Kích chuột vào nút “ **Start** “ để mở menu Window.
- 3/ Kích chuột vào mục **Run** của menu.
- 4/ Nếu cài đặt từ:

Disk A: Trong hộp thoại **Run**, gõ **a:\setup** và kích OK hoặc ENTER. CD: Trong hộp thoại **Run**, gõ **e:\setup** và kích OK hoặc ENTER.

5/ Sau đó sẽ nhận được dần dần từng bước các chỉ dẫn thao tác tiếp theo trên màn hình và hoàn thành công việc cài đặt.

- 6/ Khi kết thúc việc cài đặt, hộp thoại setup **PG/PC Interface** tự động xuất hiện.

Kích “**Cancel**” để trở về cửa sổ chính của STEP 7-Micro/Win 32.

Sau khi đã cài đặt xong có thể bắt đầu soạn thảo chương trình nhờ phần mềm STEP 7- Micro/Win 32 bằng cách nhấp đúp chuột vào biểu tượng STEP 7-Micro/Win 32 trên màn hình

Bài 4: CÁC PHÉP TOÁN NHỊ PHÂN CỦA PLC

4.1 Các liên kết logic

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

a/ Lệnh vào/ra

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.

Load Not (LDN):

Lệnh LDN nạp giá trị một 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.


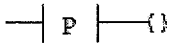
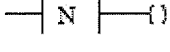
OUTPUT (=):

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.

b/ Các 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 dòng điện cung cấp (giá trị của đỉnh ngăn xếp). LAD sử dụng các tiếp điểm đặc biệt để tác động vào dòng cung cấp. Các tiếp điểm đặc biệt không có toán hạng riêng của chúng và vì thế phải đặt chúng vào vị trí phía trước của cuộn dây hoặc hộp đầu ra. Tiếp điểm chuyển tiếp dương, âm (các lệnh sườn lên và sườn xuống) có nhu cầu về bộ nhớ, bởi vậy đối với CPU 212 chỉ có thể sử dụng nhiều nhất là 128 lệnh và CPU 214 là 256 lệnh.

Các lệnh tiếp điểm đặc biệt được biểu diễn như sau trong LAD và STL

LAD	Mô tả	Toán hạng
	Tiếp điểm đảo trạng thái của dòng cung cấp. Nếu dòng cung cấp có tiếp điểm đảo thì nó bị ngắt mạch, nếu không có tiếp điểm đảo thì nó thông mạch	Không có
	Tiếp điểm chuyển đổi dương cho phép dòng cung cấp thông mạch trong một vòng quét khi sườn xung điều khiển chuyển từ 0 lên 1	Không có
	Tiếp điểm chuyển đổi âm cho phép dòng cung cấp thông mạch trong một vòng quét khi sườn xung điều khiển chuyển từ 1 đến 0	Không có

c/

STL	Mô tả	Toán hạng
NOT	Lệnh đảo giá trị của bit đầu tiên trong ngăn xếp.	Không có
EU	Lệnh nhận biết sự chuyển tiếp trạng thái từ 0 lên 1 trong một vòng quét của đỉnh ngăn xếp. Khi nhận được sự chuyển tiếp như vậy đỉnh ngăn xếp sẽ có giá trị bằng 1 trong một vòng quét.	Không có
ED	Lệnh nhận biết sự chuyển tiếp trạng thái từ 1 xuống 0 trong một vòng quét của đỉnh ngăn xếp. Khi nhận được sự chuyển tiếp như vậy đỉnh ngăn xếp sẽ có giá trị bằng 1 trong một vòng quét.	Không có

Một số bit nhớ đặc biệt

SM0.0 Luôn luôn có giá trị logic bằng 1.

SM0.1 Có giá trị logic bằng 1 ở vòng quét đầu tiên.

SM0.4 Phát xung nhịp 60 giây (0 – cho 30 giây đầu, 1 – cho 30 giây sau). SM0.5 Phát xung nhịp 1 giây (0 – cho 0,5 giây đầu, 1 – cho 0,5 giây sau).

d/ So sánh một số lệnh cơ bản trong PLC của hãng Siemens so với các hãng khác

TÊN LỆNH & TOÁN HẠNG	KLOE CKN ER- MOEL	SIEMENS CPU 214	AEG	MITSUBISI HI
EINGANG INPUT NGÕ VÀO	I0.0	E0.0 hoặc I0.0	E1	X0
AUSGAN OUTPUT NGÕ RA	Q0.0	A0.0 hoặc Q0.0	A1	Y0
MERKER BIT NHỚ	M0.0	M0.0	M1	M0
LADEN LOAD NẠP	L	LD	-	Ld
UND AND VÀ	A	U hoặc A	U	And

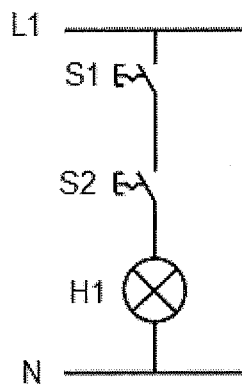
ORDER OR HOẶC	0	0	0	Or
NICHT NOT NGHỊCH ĐẢO	N	N	N	(inverts)
ZUWEISU NG KẾT QUẢ	=	=	=	Out

4.1.2. Lệnh liên kết logic cơ bản a/ 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 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

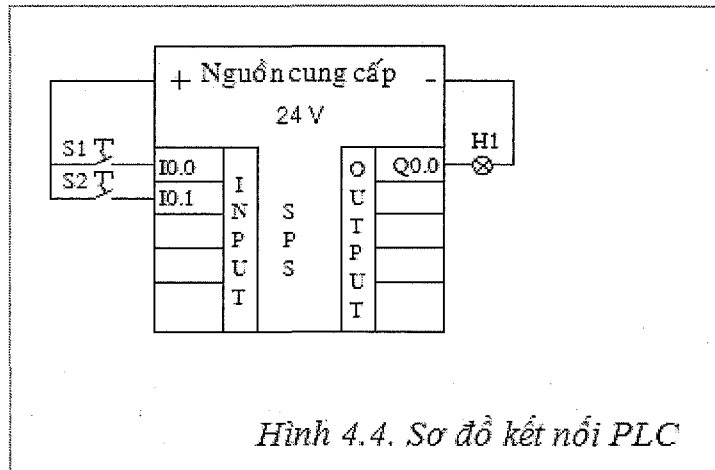
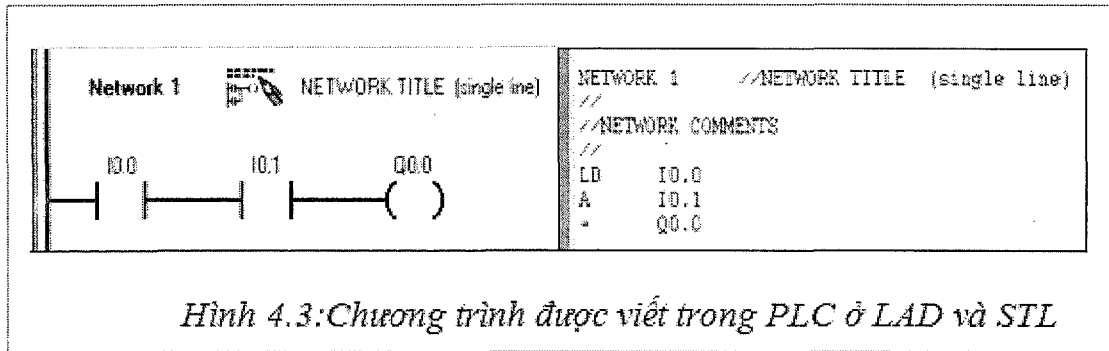
Ví dụ: Hình 4.1 mô tả sơ đồ mạch điện của một liên kết AND. Đèn H1 chỉ sáng khi tất cả các công tắc được đóng lại. Khi 1 công tắc hở mạch thì đèn H1 cũng bị tắt mạch. Liên kết AND có trạng thái 1 khi tất cả các ngõ vào có trạng thái 1.



Hình 4.1 Sơ đồ mạch điện

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.0	Công tắc thường mở
S2	I0.1	Công tắc thường mở
H1	Q0.0	Đèn báo

Hình 4.2: Bảng xác lập vào/ra



Bài tập:

Viết chương trình theo bảng câu lệnh ở hình 4.3 bằng thiết bị lập trình S7-200 và chuyển chương trình vào PLC.

Thử hoạt động của mạch.

b/ Lệnh OR (O)

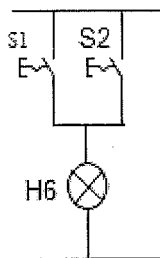
Lệnh OR phối hợp giá trị logic của một tiếp điểm n với giá trị bit đầu tiên của ngăn xếp. Kết quả của phép tính được đặt lại vào bit đầu tiên trong ngăn xếp. Giá trị của các bit còn lại trong ngăn xếp không bị thay đổi.

Cú pháp ở STL: O n

Ví dụ: hình 4.5 mô tả sơ đồ mạch điện của một liên kết OR. Đèn H6 sẽ sáng nếu một hoặc tất cả các công tắc đều đóng mạch. Ngõ ra của liên kết OR luôn luôn có trạng thái 1 nếu 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 ngõ vào/ra để kết nối với PLC.

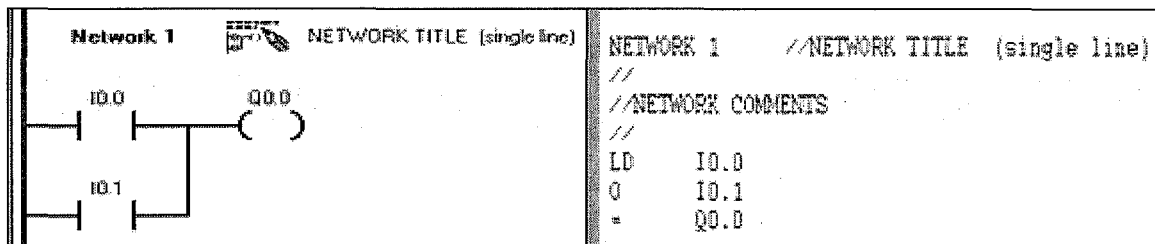
Chương trình được viết trong PLC ở các dạng LAD, và STL được cho như hình 4.7 và sơ đồ kết nối với PLC như hình 4.8



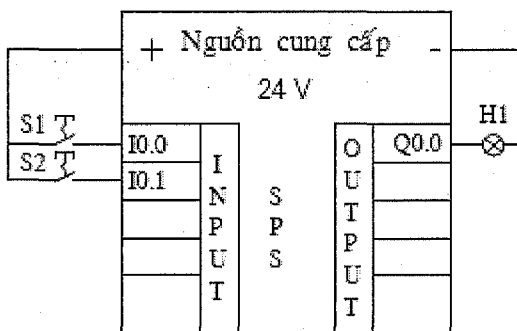
Hình 4.5: Sơ đồ mạch điện

Xác lập vào/ra		
Ký hiệu	Địa chỉ	Chức thích
S1	I0.0	Công tắc thường mở
S2	I0.1	Công tắc thường mở
H6	Q0.0	Đèn báo

Hình 4.6: Bảng xác lập vào/ra



Hình 4.7: chương trình được viết trong PLC ở dạng LAD và STL



Hình 4.8: Sơ đồ kết nối với PLC

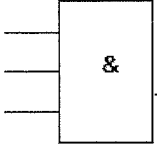
Bài tập:

Viết chương trình theo bảng câu lệnh ở hình 4.7 bằng thiết bị lập trình S7-200 và chuyển chương trình vào PLC.

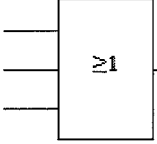
Thử hoạt động của mạch.

c. Ứng dụng cho các cổng logic cơ bản

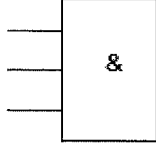
Bài tập: Hãy viết bảng liệt kê lệnh cho các mạch logic sau và kiểm tra chức năng hoạt động.



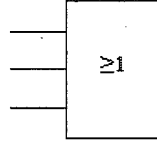
Địa chỉ	Toán hạng
000	
001	
002	
003	



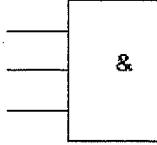
Địa chỉ	Toán hạng
000	
001	
002	
003	



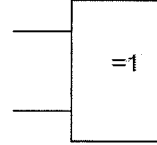
Địa chỉ	Toán hạng
000	
001	
002	
003	



Địa chỉ	Toán hạng
000	
001	
002	
003	



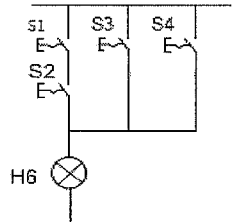
Địa chỉ	Toán hạng
000	
001	
002	
003	



Địa chỉ	Toán hạng
000	
001	
002	
003	

4.1.3. Liên kết các cổng logic cơ bản

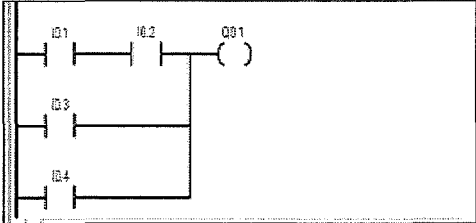
a/ Mạch hỗn hợp 1: AND trước OR



Hình 4.9: Sơ đồ mạch điện

Xác lập vào/ra		
Ký hiệu	Địa chỉ	Chú thích
S 1	IO.1	Công tắc thường mở
S 2	IO.2	Công tắc thường mở
S 3	IO.3	Công tắc thường mở
S 4	IO.4	Công tắc thường mở
H 6	Q0.0	Đèn báo

Hình 4.10: Bảng xác lập vào/ra



Hình 4.11: Chương trình viết trong PLC biểu diễn ở LAD và STL

```

NETWORK 1 //NETWORK TITLE (single line)
//NETWORK COMMENTS
//
LD I0.1
A I0.2
O I0.3
O I0.4
* Q0.1
                    
```

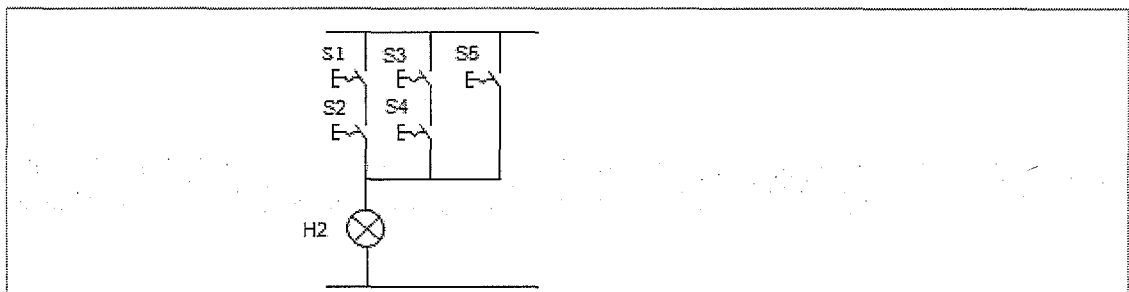
Mô tả:

Ngõ ra Q0.1 chỉ có tín hiệu bằng 1, nếu các ngõ vào I0.1 và I0.2 của cổng AND có tín hiệu bằng 1 hoặc các ngõ vào I0.3 hoặc I0.4 có tín hiệu bằng 1.

Bài tập:

Viết chương trình theo bảng câu lệnh ở hình 4.11 bằng thiết bị lập trình và chuyển vào PLC. Thử chương trình

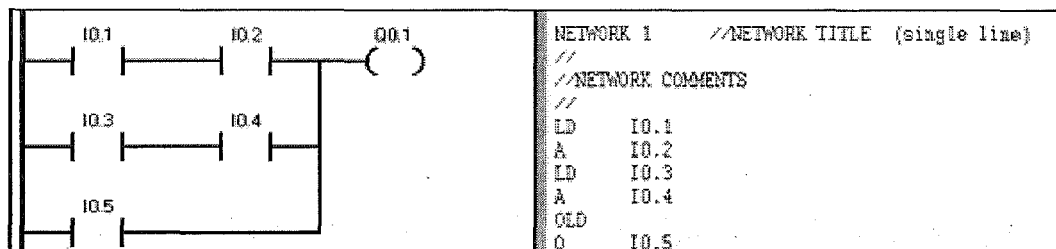
b/ Mạch hỗn hợp 2: AND trước OR



Hình 4.12: Sơ đồ mạch điện

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.1	Công tắc thường mở
S2	I0.2	Công tắc thường mở
S3	I0.3	Công tắc thường mở
S4	I0.4	Công tắc thường mở
S5	I0.5	Công tắc thường mở
H2	Q0.1	Đèn báo

Hình 4.13: Bảng xác lập vào/ra



Hình 4.14: Chương trình viết trong PLC biểu diễn ở LAD và STL

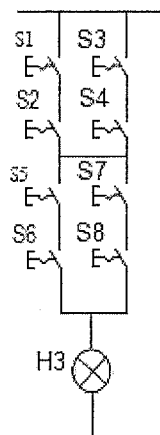
Mô tả: Ngõ ra Q0.1 ở hình 4.14 chỉ có tín hiệu bằng 1, nếu có ít nhất một ngõ vào của các cổng AND hoặc ngõ vào I0.5 có tín hiệu 1.

Bài tập:

Viết chương trình theo bảng câu lệnh ở hình 4.14 bằng thiết bị lập trình và chuyển vào PLC.

Thử chương trình

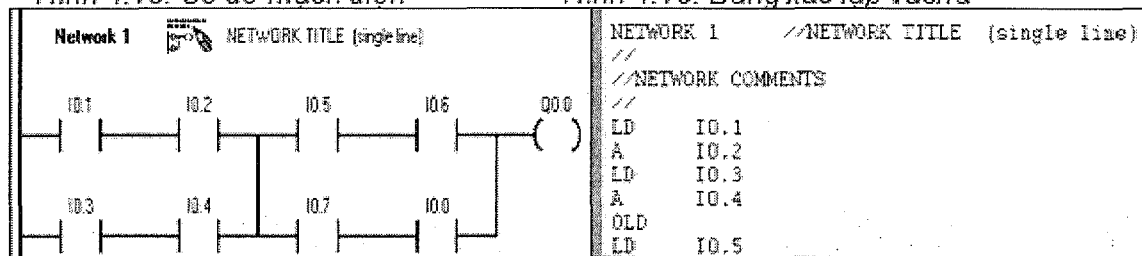
c/ Mạch hỗn hợp 3: AND trước OR



Hình 4.15: Sơ đồ mạch điện

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.1	Công tắc thường mở
S2	I0.2	Công tắc thường mở
S3	I0.3	Công tắc thường mở
S4	I0.4	Công tắc thường mở
S5	I0.5	Công tắc thường mở
S6	I0.6	Công tắc thường mở
S7	I0.7	Công tắc thường mở
S8	I0.8	Công tắc thường mở
H2	Q0.1	Đèn báo

Hình 4.16: Bảng xác lập vào/ra



Hình 4.17: Chương trình viết trong PLC biểu diễn ở LAD và STL

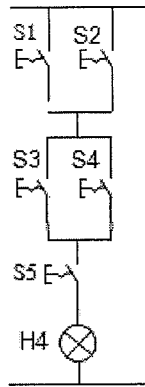
Mô tả: Ngõ ra Q0.1 chỉ có tín hiệu bằng 1, nếu hai cổng AND trước OR có cùng bằng 1.

Bài tập:

Viết chương trình theo bảng câu lệnh ở hình 4.17 bằng thiết bị lập trình và chuyển vào PLC.

Thử chương trình

d/ Mạch hỗn hợp 4: OR trước AND

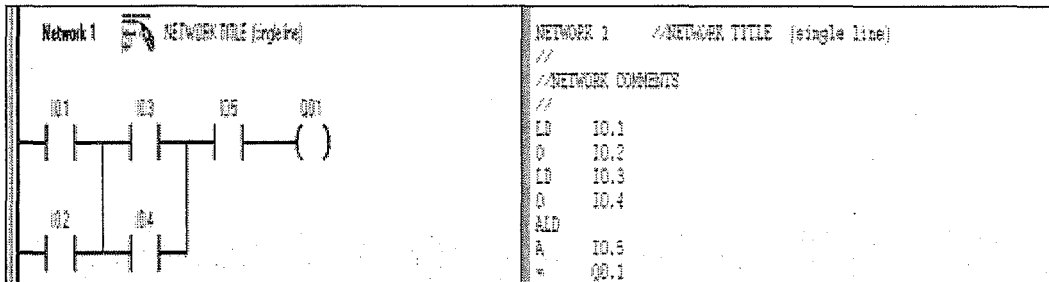


Hình. 18: Sơ đồ mạch điện

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.1	Công tắc thường mở
S2	I0.2	Công tắc thường mở
S3	I0.3	Công tắc thường mở
S4	I0.4	Công tắc thường mở
S5	I0.5	Công tắc thường mở
H4	Q0.1	Đèn báo

1

Hình. 19: Bảng xác lập vào/ra



Hình 4.20: Chương trình viết trong PLC biểu diễn ở LAD và STL

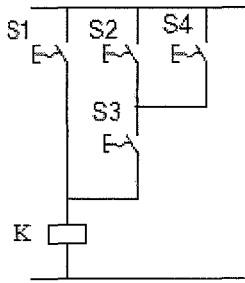
Mô tả: Ngõ ra Q0.1 chỉ có tín hiệu bằng 1, nếu 2 ngõ vào của cổng OR và ngõ vào I0.5 có tín hiệu bằng 1.

Bài tập:

Viết chương trình theo bảng câu lệnh ở hình 4.20 bằng thiết bị lập trình và chuyển vào PLC.

Thử chương trình

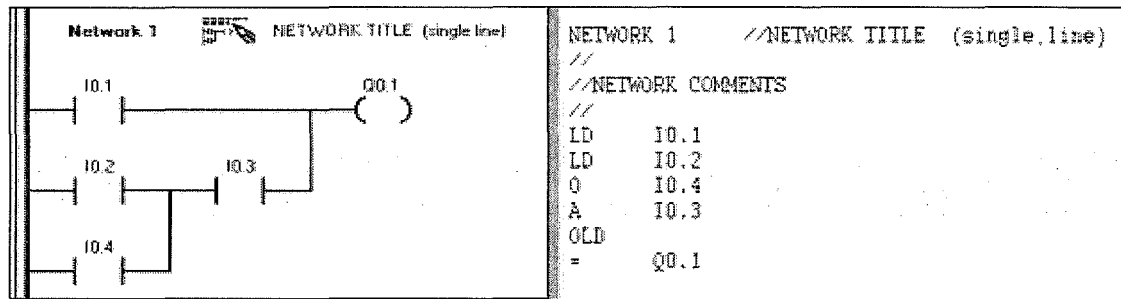
e/ Mạch hỗn hợp 5: OR trước AND



Hình 4.21: Sơ đồ mạch điện

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.1	Công tắc thường mở
S2	I0.2	Công tắc thường mở
S3	I0.3	Công tắc thường mở
S4	I0.4	Công tắc thường mở
K1	Q0.1	Cuộn dây ngắt điện từ

Hình 4.22: Bảng xác lập vào/ra



Hình 4.23. Chương trình viết trong PLC biểu diễn ở LAD và STL

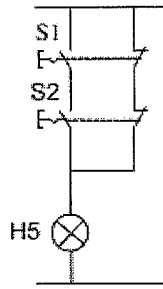
Mô tả: Ngõ ra Q0.1 chỉ có tín hiệu bằng 1, nếu ngõ vào I0.1 hoặc ngõ vào I0.3 và ngõ vào I0.2 hoặc I0.4 có tín hiệu bằng 1.

Bài tập:

Viết chương trình theo bảng câu lệnh ở hình 4.23 bằng thiết bị lập trình và chuyển vào PLC.

Thử chương trình

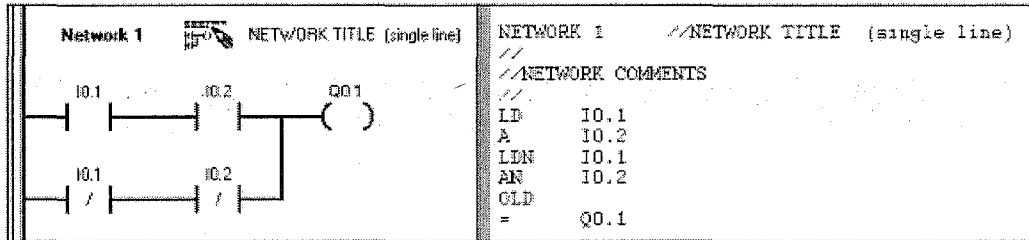
4.1.4. Bài tập ứng dụng**a/ Liên kết****XOR**



Hình 4.24: Sơ đồ mạch điện

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.1	Nút nhấn thường mở
S2	I0.2	Nút nhấn thường mở
H5	Q0.1	Đèn báo

Hình 5.25: Bảng xác lập vào/ra



Hình 4.26: Chương trình viết trong PLC biểu diễn ở LAD và STL

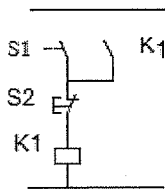
Mô tả: Ngõ ra của loại cổng liên kết này luôn có tín hiệu bằng 1, khi hai ngõ vào có giá trị giống nhau (hoặc 0 hoặc 1).

Bài tập:

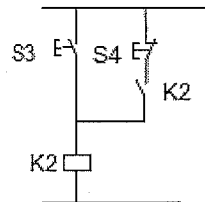
Viết chương trình bằng thiết bị lập trình và chuyển vào PLC.

Thử chương trình.

b/ Mạch tự duy trì (Tự giữ)



Hình 4.27: Sơ đồ mạch điện A



Hình 4.28: Sơ đồ mạch điện B

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.1	Nút nhấn thường mở
S2	I0.2	Nút nhấn thường đóng
S3	I0.3	Nút nhấn thường mở
S4	I0.4	Nút nhấn thường đóng
K1	Q0.1	Khởi động từ
K2	Q0.2	Khởi động từ

Hình 4.29 : Bảng xác lập vào/ra

Mô tả: Mạch A:

Khi nhấn S1, ngõ vào I0.1 có tín hiệu 1, làm cho ngõ ra Q0.1 (K1) có tín hiệu 1. Cuộn dây K1 có điện. Khi nhấn S2 (ngõ vào I0.2) ngõ ra Q0.1 sẽ có tín hiệu 0, cuộn dây K1 mất điện.

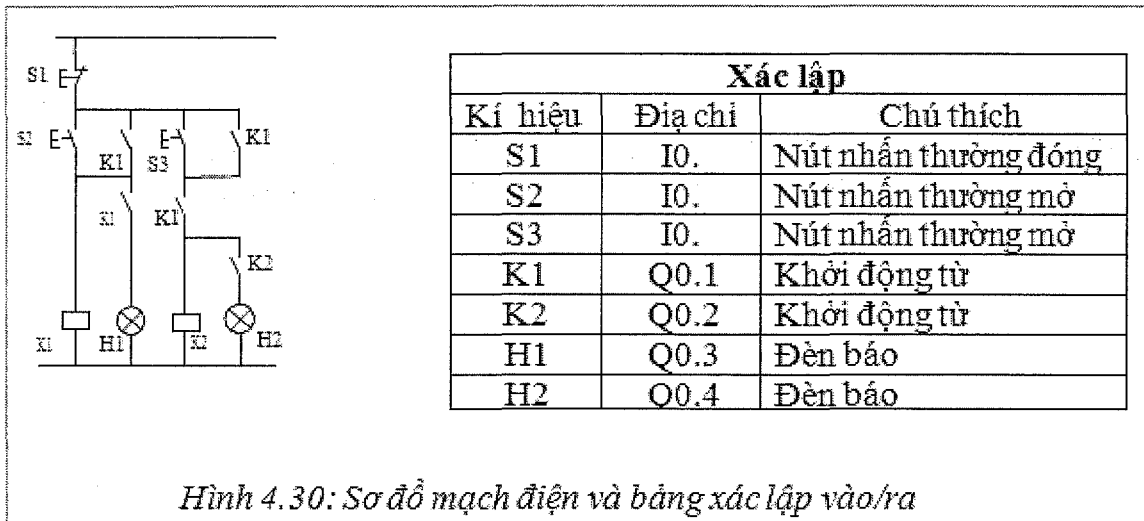
Mạch B:

Khi nhấn S3, ngõ vào I0.3 có tín hiệu 1, làm cho ngõ ra Q0.2 (K2) có tín hiệu 1. Cuộn dây K2 có điện. Khi nhấn S4 (ngõ vào I0.4) ngõ ra Q0.2 sẽ có tín hiệu 0, cuộn dây K2 mất điện.

Bài tập:

Vẽ sơ đồ chức năng và sơ đồ công tắc theo các sơ đồ mạch điện ở hình 4.27 và 4.28. Viết bảng câu lệnh cho hai sơ đồ nói trên.

Viết chương trình bằng thiết bị lập trình và chuyển vào PLC. Thử chương trình.

c/ Mạch điều khiển tuần tự cưỡng bức

Hình 4.30: Sơ đồ mạch điện và bảng xác lập vào/ra

Mô tả:

Khi nhấn S2 thì cuộn dây K1 có điện và nó tự duy trì qua các tiếp điểm thường mở, đèn H1 sáng. Khi nhấn S3 cuộn dây K2 có điện và tự duy trì, đèn H2 sáng. Nhấn S1 mạch bị cắt hoàn toàn khỏi nguồn điện.

K2 chỉ có thể đóng mạch được nếu trước đó K1 đã được đóng.

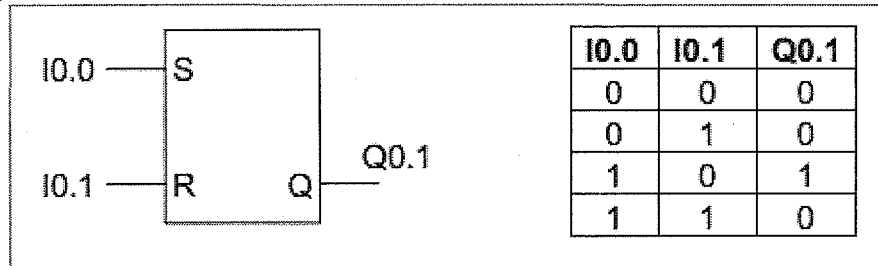
Bài tập:

Vẽ sơ đồ chức năng và sơ đồ công tắc theo sơ đồ hình 4.33. Viết bảng câu lệnh mô tả lệnh.

Viết và thử chương trình.

4.2 CÁC LỆNH GHI/XOÁ GIÁ TRỊ CHO TIẾP ĐIỂM

4.2.1. 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 I0.0 có trạng thái 1 thì ngõ ra có tín hiệu 1. Khi ngõ vào I0.1 có trạng thái tín hiệu 1 thì ngõ ra có tín hiệu 0. Các trường hợp còn lại, ngõ ra đều bằng 0. Người ta gọi mạch này là mạch nhớ tín hiệu (giống như mạch tự giữ trong mạch điều khiển dùng rơ le). Thay đổi trạng thái các ngõ ra: đặt (set) hoặc xoá (reset).

4.2.2. SET (S) và RESET (R) trong S7-200

Trong sơ đồ hình thang, các cuộn dây ra sẽ ở trạng thái đặt (bằng 1) hoặc xoá (bằng 0) phụ thuộc vào các quan hệ logic điều khiển dòng tín hiệu. Khi có dòng chảy đến cuộn dây, một ngõ ra hoặc nhiều ngõ ra sẽ được đặt cũng như xoá bởi các lệnh này.

Trong bảng liệt kê lệnh, các giá trị này sẽ truyền giá trị của đỉnh ngăn xếp đến các ngõ ra tương ứng. Khi đỉnh ngăn xếp bằng 1 thì các ngõ ra sẽ được đặt cũng như xoá bởi các lệnh set và reset (phạm vi cho phép từ 1 đến 255 ngõ ra). Nội dung ngăn không bị thay đổi bởi những lệnh này. Trong cả hai dạng sơ đồ hình thang và liệt kê chỉ thị đều cho phép khả năng truy xuất trực tiếp ngõ ra. Giá trị ngõ ra trong toán hạng được ghi đồng thời vào bộ đệm và các ngõ ra vật lý, khác với các lệnh gián tiếp, giá trị này chỉ được ghi vào bộ đệm.

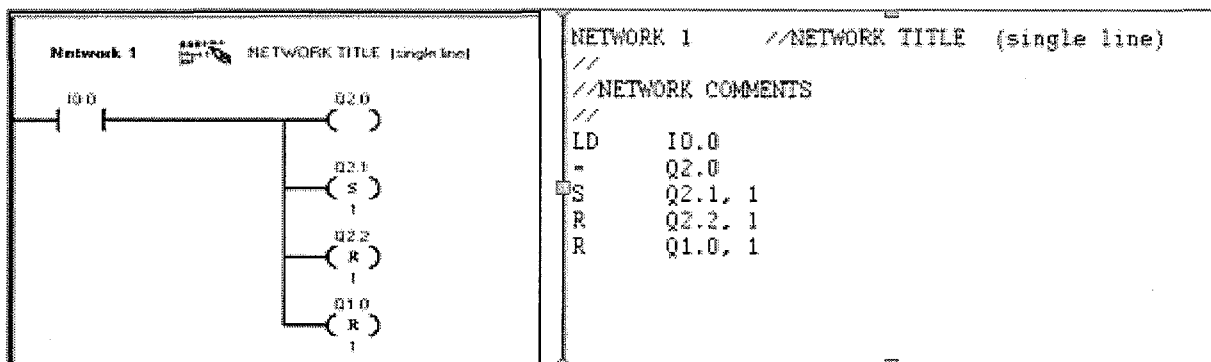
Mô tả lệnh S (Set) và R (Reset) bằng LAD

LAD	Mô tả	Toán hạng
	Đóng một mảng gồm n các tiếp điểm kể từ S_BIT	S_BIT: I,Q,M,SM,,C,V (bit) n :
	Ngắt một mảng gồm n các tiếp điểm kể từ S_BIT. Nếu S_BIT lại chỉ vào Timer hoặc Couter thì lệnh sẽ xoá bit đầu ra của Timer, Counter đó.	IB,QB,MB,SMB, VB,AC,hãng số,*VB,*AC
	Đóng một mảng gồm n các tiếp điểm kể từ S_BIT	S_BIT: Q (bit) n :
	Ngắt tức thời một mảng gồm n các tiếp điểm kể từ địa chỉ S_BIT	IB,QB,MB,SMB, VB,AC,hãng số,*VB,*AC

Mô tả lệnh S (Set) và R (Reset) bằng STL

STL	Mô tả	Toán hạng
S S_BIT n	Ghi giá trị logic vào một mảng gồm n bit kể từ địa chỉ S_BIT	S_BIT: I,Q,M,SM,C,V (bit) n :
R S_BIT n	Xoá một mảng gồm n bit kể từ địa chỉ S_BIT. Nếu S_BIT lại chỉ vào Timer hoặc Couter thì lệnh sẽ xoá bit đầu ra của Timer, Counter đó.	IB,QB,MB,SMB, VB,AC,hãng số,*VB,*AC
SI S_BIT n	Đóng một mảng gồm n các tiếp điểm kể từ S_BIT	S_BIT: Q (bit) n :
RI S_BIT n	Xoá tức thời một mảng gồm n bit kể từ địa chỉ S_BIT	IB,QB,MB,SMB, VB,AC,hãng số,*VB,*AC

Thí dụ : Mô tả việc thực hiện lệnh S (Set) và R (Reset) trong LAD và STL



4.2.3. Ví dụ ứng dụng dùng bộ nhớ

a/ Mạch chốt lẫn nhau của 2 van từ

Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn thường mở
S2	I0.1	Nút nhấn thường đóng
S3	I0.2	Nút nhấn thường mở
Y1	Q0.0	Van từ 1
Y2	Q0.1	Van từ 2

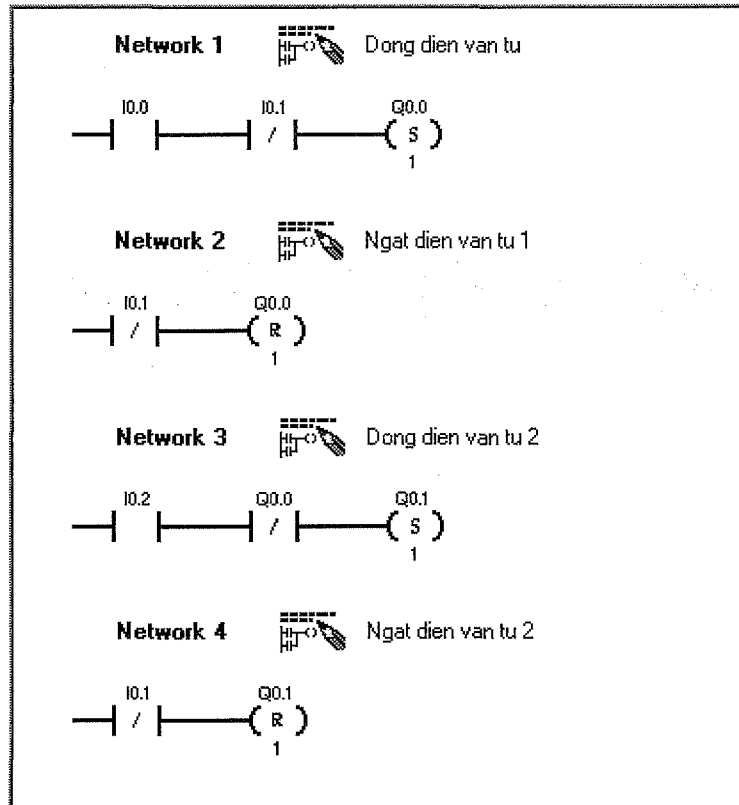
Hình 4.31 Sơ đồ mạch logic

Hình 4.32: Bảng xác lập vào/ra

Mô tả hoạt động:

Qua việc khởi động S1 hoặc S3 các bộ nhớ một (van từ 1) hoặc bộ nhớ hai (van từ 2) sẽ được đặt. Nút nhấn S2 làm nhiệm vụ cắt mạch

Chương trình được viết ở LAD



Chương trình được viết ở STL:

```

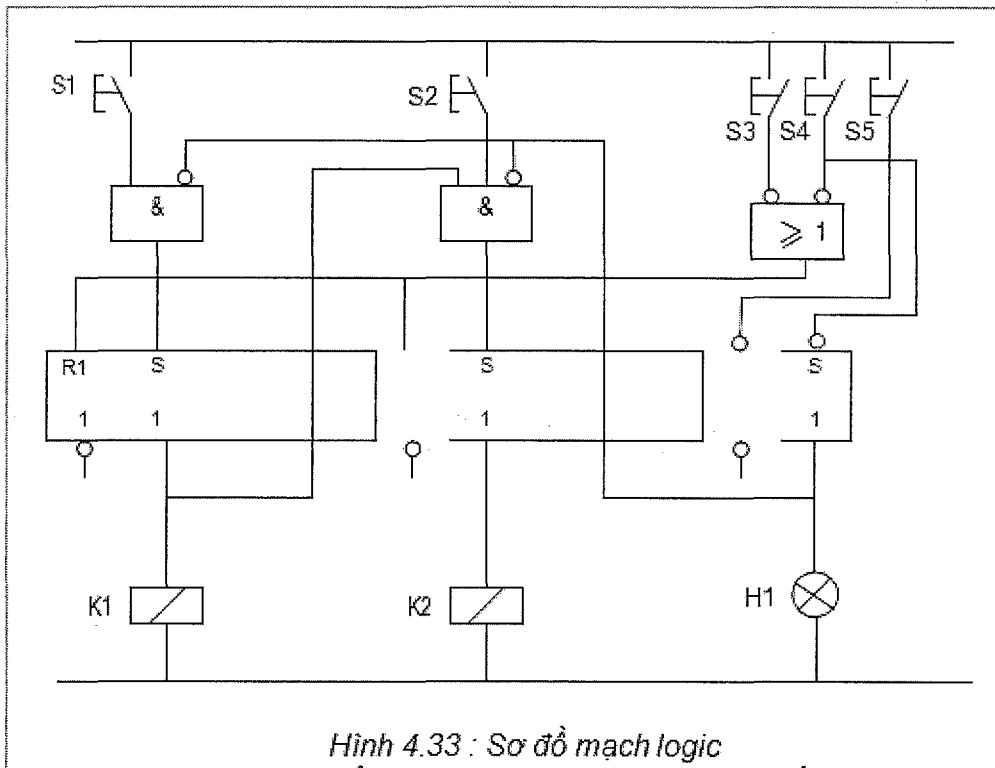
NETWORK 1      //Dong dien van tu
//
//NETWORK COMMENTS
//
LD      IO.0
AN      IO.1
S       Q0.0, 1

NETWORK 2      //Ngat dien van tu 1
LDN     IO.1
R       Q0.0, 1

NETWORK 3      //Dong dien van tu 2
LD      IO.2
AN      Q0.0
S       Q0.1, 1

NETWORK 4      //Ngat dien van tu 2
LDN     IO.1
R       Q0.1, 1
  
```

**b/ Mạch tuần tự cường bức có
báo lỗi Mô tả hoạt động:**



Hình 4.33 : Sơ đồ mạch logic

Qua việc khởi động nhấn S1 thì K1 có điện. Khi nhấn S2 thì K2 có điện. Khi nhấn S3 cả K1 và K2 mất điện. Khi có lỗi thì cả K1 và K2 có thể bị ngắt điện bằng cách nhấn S4. Nút nhấn S5 để phục hồi mạch, khi đó quá trình mới có

thể được bắt đầu.

Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau:

- Lập bảng xác lập vào/ra.
- Vẽ sơ đồ kết nối với PLC.
- Vẽ sơ đồ STL và sơ đồ LAD.
- Viết bảng câu lệnh mô tả mạch.
- Viết và thử chương trình.

c/ Bộ chọn theo

bước Mô tả

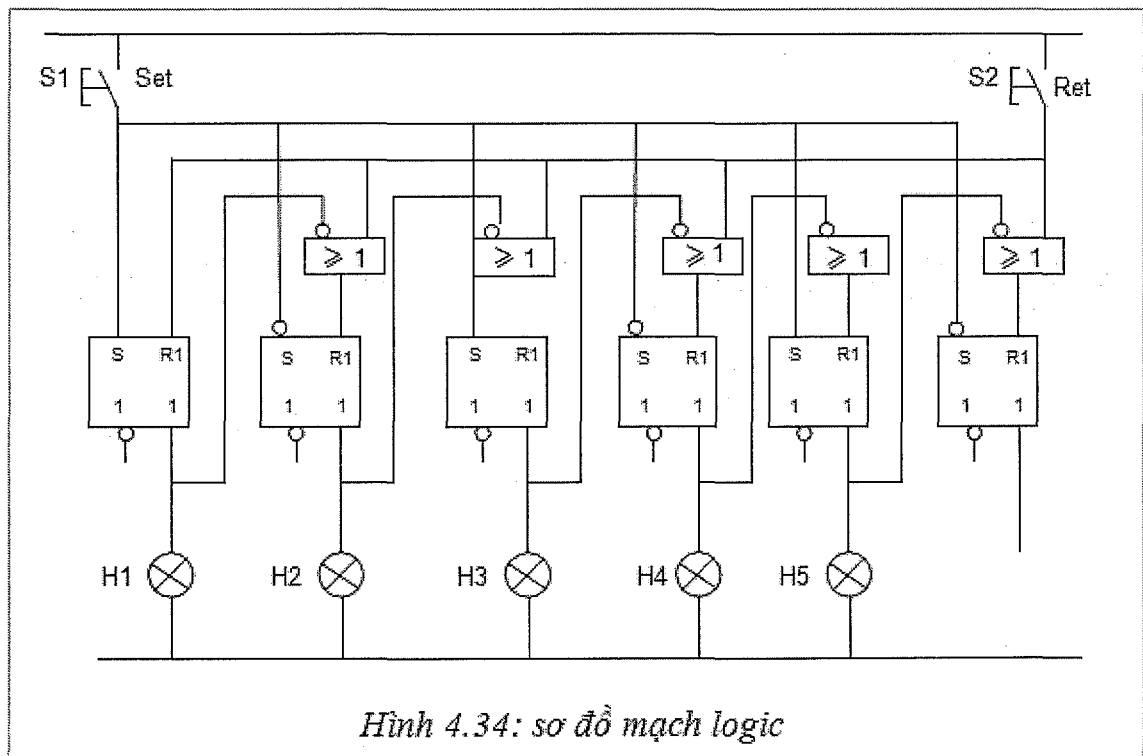
hoạt động:

Qua việc khởi động nút nhấn S1 tín hiệu được thay đổi từ „0“ đến „1“ và đèn H1 sáng. Khi rời tay không nhấn S1 nữa thì tín hiệu thay đổi từ „1“ đến „0“, đèn H2 sáng. Lập lại việc nhấn S1 tín hiệu thay đổi từ „0“ đến „1“ và đèn H3 sáng. Khi nhấn S2 thì tất cả các bộ nhớ được reset. Quá trình có thể được lập lại từ đầu.

Nhiệm vụ:

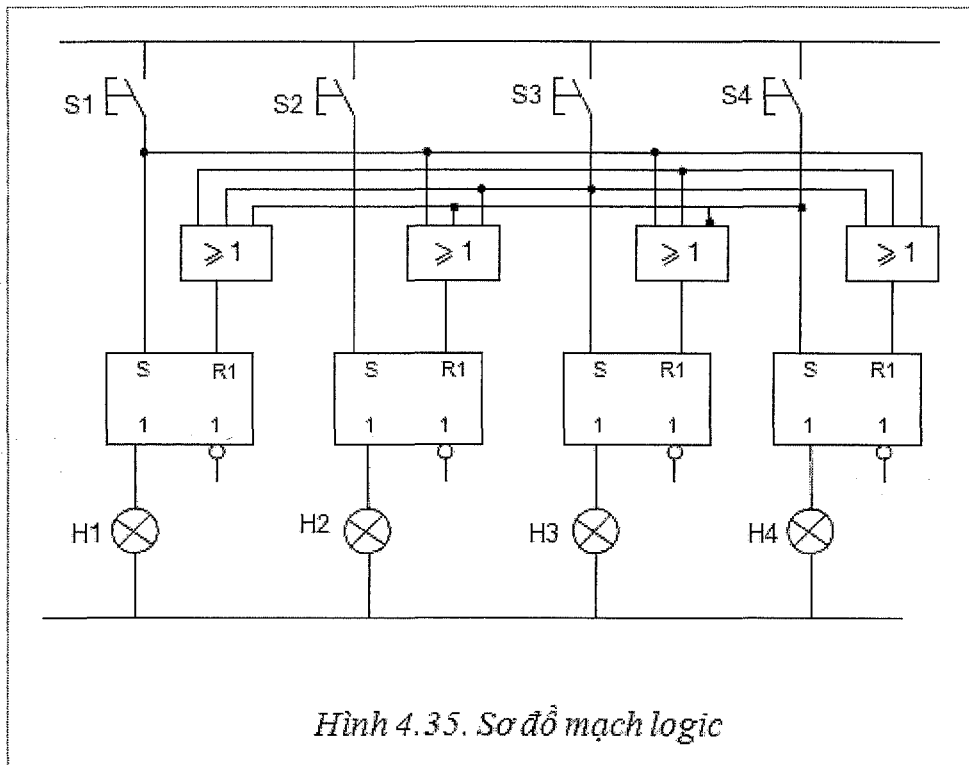
Hãy thực hiện mạch theo các yêu cầu sau:

- Lập bảng xác lập vào/ra.
- Vẽ sơ đồ kết nối với PLC.
- Vẽ sơ đồ STL và sơ đồ LAD.
- Viết bảng câu lệnh mô tả mạch.
- Viết và thử chương trình.



Hình 4.34: sơ đồ mạch logic

d/ Nhớ bằng nút nhấn

**Mô tả hoạt động:**

Khi nhấn các nút nhấn từ S1 đến S4 tín hiệu sẽ thay đổi trạng thái từ 0 đến 1, các bộ nhớ tương ứng sẽ được đặt và tất cả các bộ nhớ khác sẽ bị reset.

Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau:

- Lập bảng xác lập vào/ra.
- Vẽ sơ đồ kết nối với PLC.
- Vẽ sơ đồ STL và sơ đồ LAD.
- Viết bảng câu lệnh mô tả mạch.
- Viết và thử chương trình.

4.3 Timer

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

Bộ điều khiển lập trình S7 - 200 có 128 timer (với CPU 214) được chia làm 2 loại khác nhau:

- Timer tạo thời gian trễ không có nhớ (ON - Delay Timer) kí hiệu là TON.
- Timer tạo thời gian trễ có nhớ (Retentive - ON - Delay Timer) kí hiệu là TONR. Cả hai loại Timer đều có 3 loại với 3 độ phân giải thời gian khác nhau:
 - 1ms
 - 10ms
 - 100ms

Thời gian trễ τ được tạo ra chính là tích của độ phân giải của Timer được chọn và giá trị đặt trước cho Timer.

Ví dụ: một bộ Timer có độ phân giải 10ms, giá trị đặt trước là 30 thì thời gian trễ sẽ là

$$\tau = 300 \text{ ms.}$$

Timer TON và TONR đều có các độ phân giải là 1ms, 10ms, 100ms và các giá trị cực đại là 32,767s, 327,67s, 3276,7s. Độ phân giải và sự hoạt động được đưa ra ở các số hiệu Timer cũng như các giá trị thời gian (PT) có thể là một số (Konstante) cũng có thể là word: VW, T, Z, EW, AW, SMW, SW...

Reset một Timer

Một Timer đang làm việc có thể được đưa lại về trạng thái ban đầu, công việc đó được gọi là reset Timer. Khi reset một bộ Timer, T- word và T- bit của nó đồng thời được xóa và có giá trị bằng 0, như vậy giá trị đếm tức thời được đặt về 0 và tín hiệu đầu ra cũng có logic bằng 0. Có thể reset bất cứ bộ Timer nào của S7 - 200 bằng lệnh R. Có hai phương pháp để reset một Timer TON:

- Xóa tín hiệu đầu vào
- Dùng lệnh reset.

Dùng lệnh R là phương pháp duy nhất để reset các bộ Timer kiểu TON

Cập nhật Timer có độ phân giải là 1ms

CPU của S7 - 200 có các bộ Timer có độ phân giải 1ms cho phép PLC cập nhật và thay đổi giá trị đếm tức thời trong T-word mỗi 1ms một lần. Các bộ Timer có độ phân giải thấp này có khả năng điều khiển chính xác các thao tác.

Ngay sau khi bộ Timer với độ phân giải 1ms được kích, việc cập nhật để thay đổi giá trị đếm tức thời trong T-word hoàn toàn tự động. Chỉ nên đặt giá trị rất nhỏ cho PT của bộ Timer có độ phân giải 1ms. Tần số cập nhật để thay đổi giá trị đếm tức thời và T-bit của một bộ Timer có độ phân giải 1ms không phụ thuộc vào vòng quét (scan) của bộ điều khiển và vòng quét của chương trình đang chạy. Giá trị đếm tức thời và T-bit của bộ Timer này có thể được cập nhật vào bất kỳ thời điểm nào trong vòng quét và được cập nhật nhiều lần trong một vòng quét nếu thời gian vòng quét lớn hơn 1ms.

Thực hiện lệnh R đối với một Timer có độ phân giải 1ms đang ở trạng thái làm việc có nghĩa là đưa Timer đó về trạng thái ban đầu, giá trị đếm tức thời của Timer được đưa về 0 và T-bit nhận giá trị logic 0.

Cập nhật Timer có độ phân giải là 10ms

CPU của S7 - 200 có các bộ Timer với độ phân giải 10ms. Sau khi đã được kích, việc cập nhật T-word và T-bit để thay đổi giá trị đếm tức thời và trạng thái logic đầu ra của các bộ Timer này không phụ thuộc vào chương trình và được tiến hành hoàn toàn tự động mỗi vòng quét một lần và tại thời điểm đầu vòng quét.

Thực hiện lệnh R đối với một bộ Timer có độ phân giải là 10ms đang ở trạng thái làm việc là đưa Timer về trạng thái ban đầu và xoá T-word và T-bit của Timer.

Cập nhật Timer có độ phân giải là 100ms

Hầu hết các bộ Timer của S7 – 200 là các bộ Timer có độ phân giải là 10ms. Giá trị để lưu trữ trong bộ Timer 100ms được tính tại mỗi đầu vòng quét và thời gian để tính sẽ là khoảng thời gian từ đầu vòng quét trước đó.

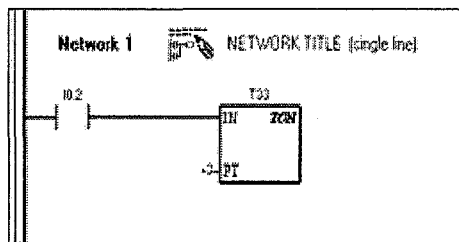
Việc cập nhật để thay đổi giá trị đếm tức thời của Timer chỉ được tiến hành ngay tại thời điểm có lệnh khai báo cho Timer trong chương trình. Bởi vậy quá trình cập nhật giá trị đếm tức thời không phải là quá trình tự động và không nhất thiết phải thực hiện một lần trong mỗi vòng quét ngay cả khi Timer đã được kích.

4.3.1. – Delay Timer (TON)

Địa chỉ của On – Delay Timer ở S7 – 200 được cho theo độ phân giải như sau:

Độ phân giải	CPU 212/214/215/216	CPU 214/215/216	CPU 215/216
1ms	T32	T96	
10ms	T33 đến T36	T97 đến T100	
100ms	T37 đến T63	T101 đến T127	T128 đến T155

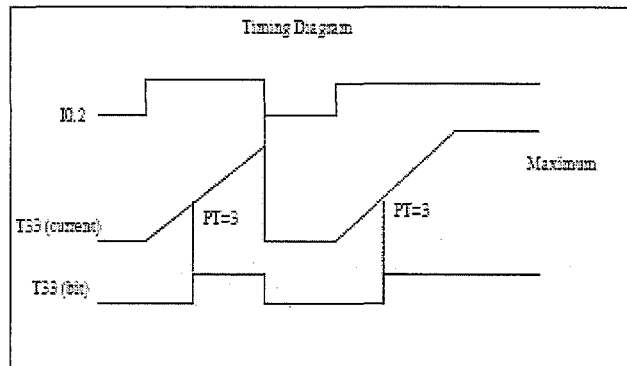
TON được viết trong LAD và STL cũng như giản đồ thời gian của nó



```

NETWORK 1 //NETWORK TITLE (single line)
//
//NETWORK COMMENTS
//
LD I0.2
TON T33, +3

```



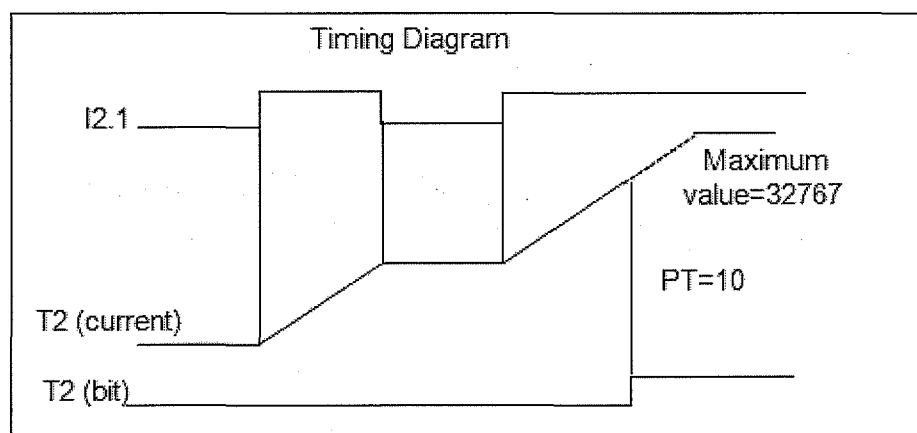
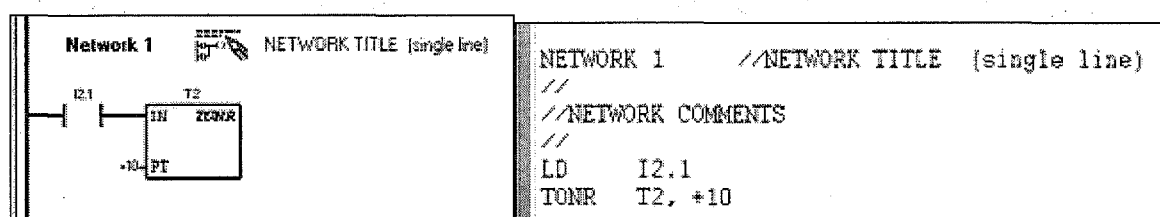
Thời gian đóng mạch chậm khởi động và đếm đến giá trị cao, khi ngõ vào I0.2 đóng mạch. Nếu giá trị đếm tức thời \geq giá trị đặt trước, thì bit thời gian hoạt động (T33 có tín hiệu 1). Nó bị reset khi ngõ vào I0.2 ngắt mạch

4.3.2. Retentive On – Delay Timer (TONR)

Địa chỉ của TONR ở S7 – 200 được cho theo độ phân giải như sau:

Độ phân giải	CPU 212/214	CPU 214	CPU 215/216
1ms	T0	T64	
10ms	T1 đến T4	T65 đến T68	
100ms	T5 đến T31	T69 đến T95	

TONR được viết trong LAD và STL cũng như giản đồ thời gian của nó:

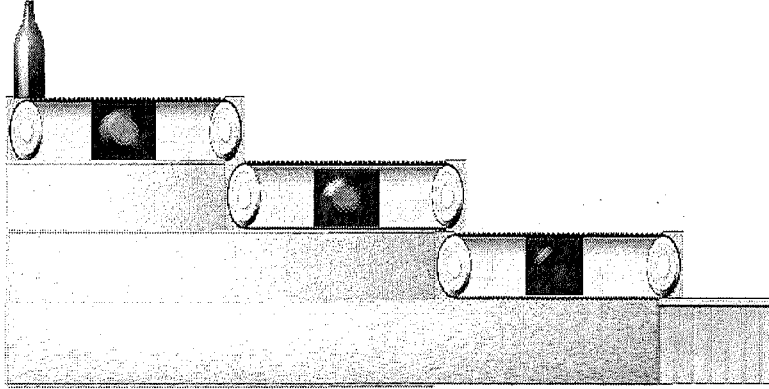


Thời gian đóng mạch chậm khởi động và đếm đến giá trị cao, khi ngõ vào I2.1 đóng mạch. Nếu giá trị đếm tức thời \geq giá trị đặt trước, thì bit thời gian hoạt động (T2 có tín hiệu 1). Giá trị đếm tức thời được lưu lại và không bị thay đổi trong khoảng thời gian tín hiệu đầu vào I2.1 có tín hiệu logic 0. Giá trị

của T-bit không được nhớ mà phụ thuộc hoàn toàn vào kết quả so sánh giữa giá trị đếm tức thời và giá trị đặt trước.

Ví dụ về điều khiển bằng Timer:

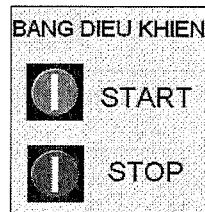
ĐIỀU KHIỂN HỆ THỐNG BĂNG TẢI



Yêu cầu: Hệ thống băng tải gồm 3 động cơ.

Khi khởi động nhấn **START** thì động cơ 1 chạy, sau **3 giây** thì tự động động cơ 2 chạy, tiếp theo **3 giây** kế tự động động cơ 3 chạy. Tương ứng mỗi động cơ chạy thì có đèn sáng.

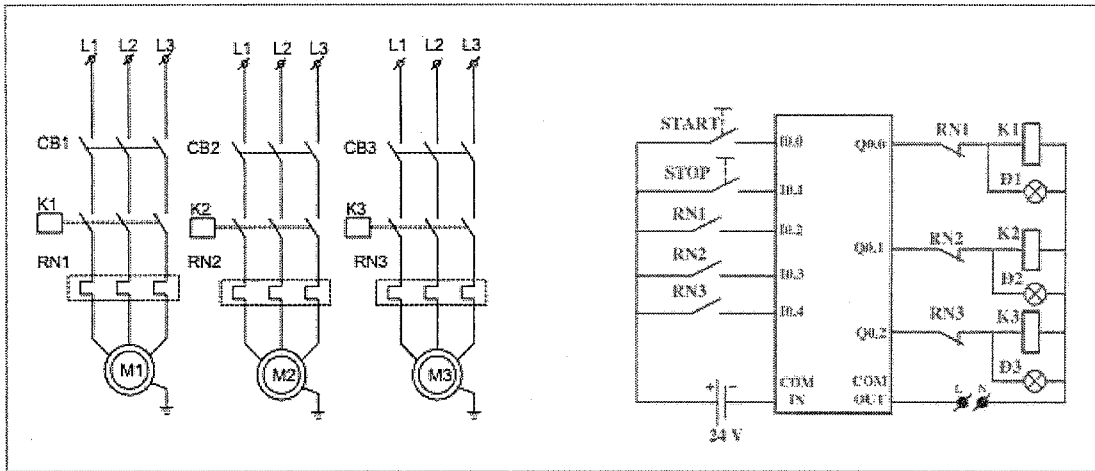
Khi nhấn nút **STOP (dừng)** thì động cơ thứ 1 dừng trước, sau **5 giây** thì tự động động cơ thứ 2 dừng và sau **5 giây** thì tự động động cơ 3 dừng hẳn.



Bài làm:

MẠCH ĐỘNG LỰC:

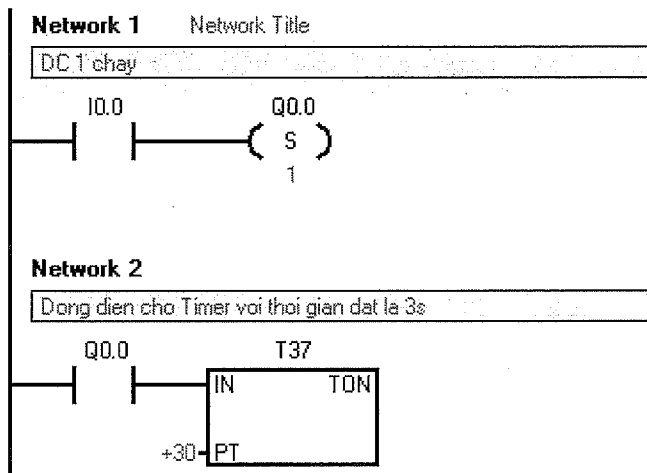
NỘI PLC:



Các thiết bị vào/ra:

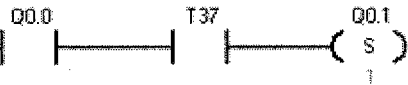
Kí	Địa chỉ	Giải thích
STAR	I0.0	Nút nhấn khởi động
STO	I0.1	Nút nhấn dừng
K1	Q0.0	Ngõ ra điều khiển động cơ kéo băng tải 1
K2	Q0.1	Ngõ ra điều khiển động cơ kéo băng tải 2
K3	Q0.2	Ngõ ra điều khiển động cơ kéo băng tải 3.

Chương trình được viết trong trong PLC ở dạng LAD:



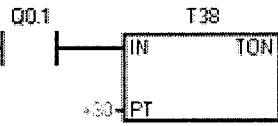
Network 3

Dung DC 2 chuy sau 3s



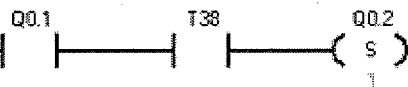
Network 4

Dong dien cho T38 voi thoi gian cai dat la 3s



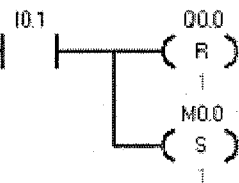
Network 5

Dung DC 3 chuy



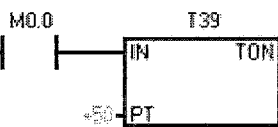
Network 6

Dung DC 1



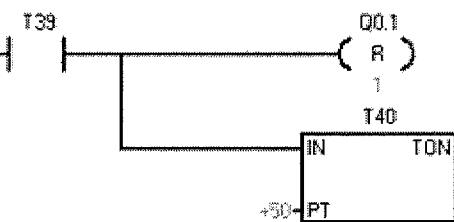
Network 7

Cai dat T39 voi thoi gian dat la 5s



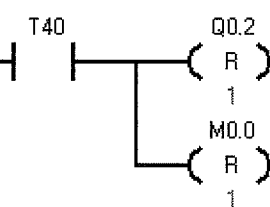
Network 8

Dung DC 2



Network 9

Dung DC 3



Chương trình được viết dưới dạng STL:

Network 1 Network Title <hr/> DC 1 chạy LD I0.0 S Q0.0, 1	Network 7 <hr/> Cài đặt T39 với thời gian đặt là 5s LD M0.0 TON T39, +50
Network 2 <hr/> Đóng điện cho Timer với thời gian đặt là 3s LD Q0.0 TON T37, +30	Network 8 <hr/> Dùng DC 2 LD T39 R Q0.1, 1 TON T40, +50
Network 3 <hr/> DC 2 chạy sau 3s LD Q0.0 A T37 S Q0.1, 1	Network 9 <hr/> Dùng DC 3 LD T40 R Q0.2, 1 R M0.0, 1
Network 4 <hr/> Đóng điện cho T38 với thời gian cài đặt là 3s LD Q0.1 TON T38, +30	
Network 5 <hr/> DC 3 chạy LD Q0.1 A T38 S Q0.2, 1	
Network 6 <hr/> Dùng DC 1 LD I0.1 R Q0.0, 1 S M0.0, 1	

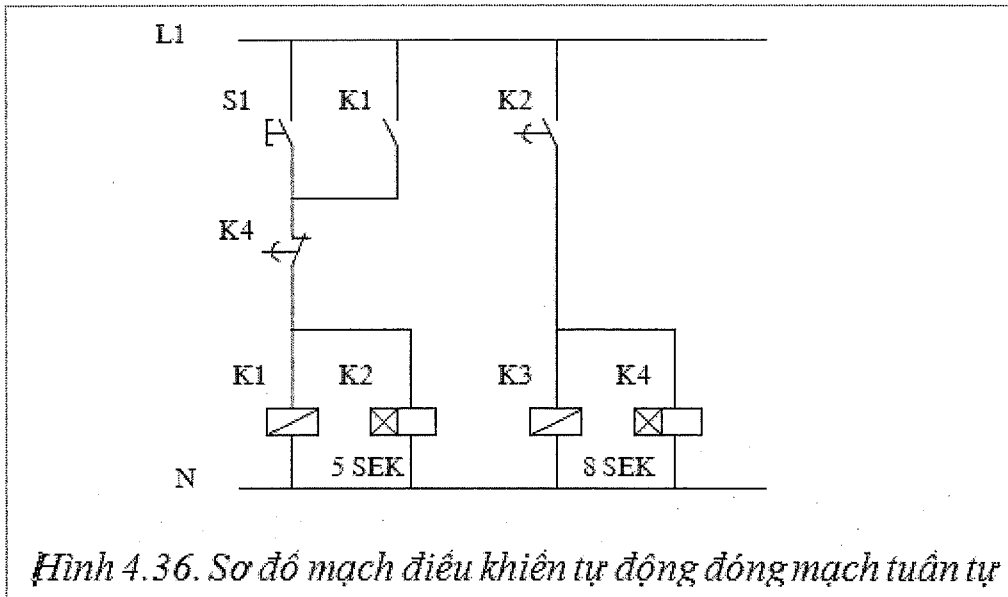
Nguyên lý hoạt động của hệ thống:

Khi nhấn nút I0.0 (START) ở Network1 Q0.0 có điện > công tắc tơ K1 có điện, đóng tiếp điểm K1 trên mạch động lực > động cơ1 (M1) kéo băng tải 1 chạy. Khi Q0.0 có điện sẽ đóng tiếp điểm thường mở Q0.0 ở Network2, sau khoảng thời gian là 3S T1 có điện > Q0.1 có điện > công tắc tơ K2 có điện, đóng tiếp điểm K2 trên mạch động lực > động cơ 2 (M2) kéo băng tải 2 chạy và tiếp điểm thường mở T1 ở Network3 đóng lại, sau khoảng thời gian 3S T2 có điện > Q0.2 có điện > công tắc tơ K3 có điện, đóng tiếp điểm K3 trên mạch động lực > động cơ 3 (M3) có điện kéo băng tải 3 chạy.

Khi nhấn nút I0.1 (STOP) ở Network 1 Q0.0 mất điện, công tắc tơ K1 mất điện, mở tiếp điểm K1 trên mạch động lực > M1 dừng, đồng thời đóng điện cho T39 có điện. Sau 5s, ngắt điện động cơ 2 và đóng điện cho T40. Sau 5s, thì ngừng động cơ 3.

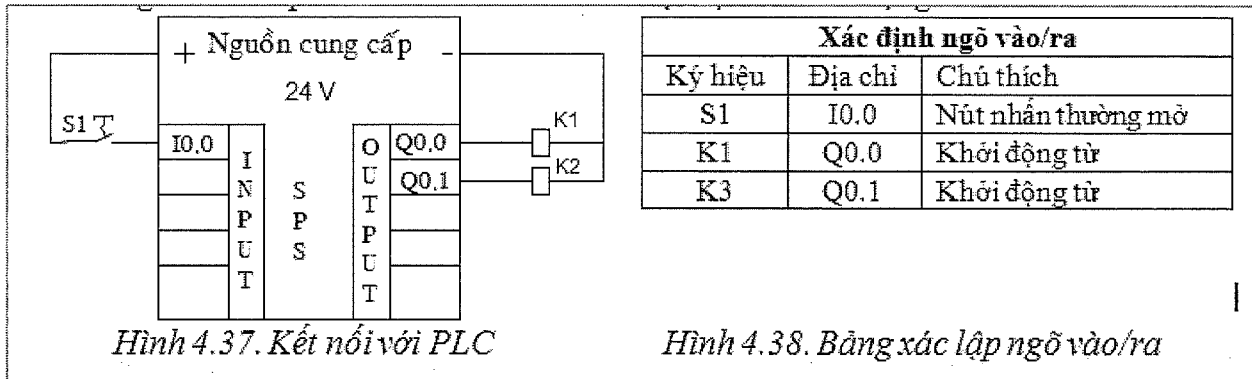
4.3.3. Ứng dụng Timer

a/ Tự động đóng mạch tuần tự:



Mô tả hoạt động:

Nhấn S1, K1 có điện và tự duy trì – đồng thời Role K2 có điện. Sau 1 khoảng thời gian 5s, tiếp điểm K2 của Role thời gian đóng mạch cho K3, K4. Sau 1 khoảng thời gian 8s tiếp điểm K4 hở ra và toàn bộ mạch trở về trạng thái ban đầu.



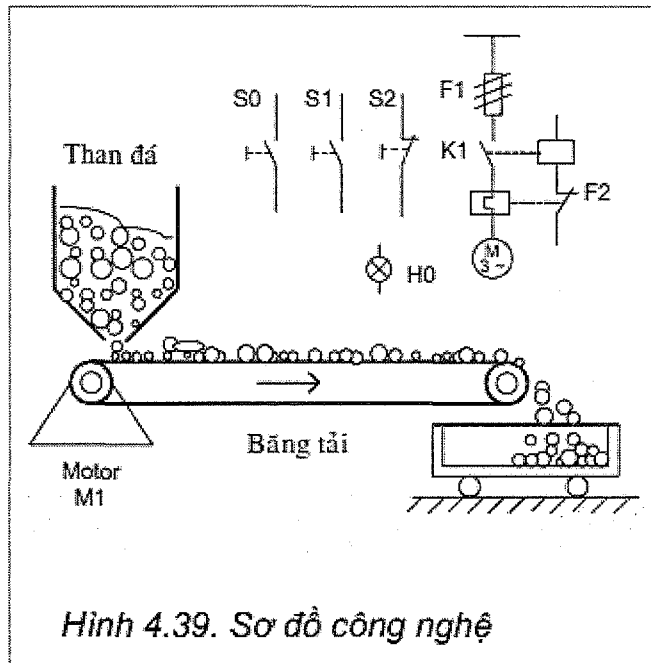
Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau: Vẽ sơ đồ STL và sơ đồ LAD.

Viết bảng câu lệnh mô tả mạch.

Viết và thử chương trình

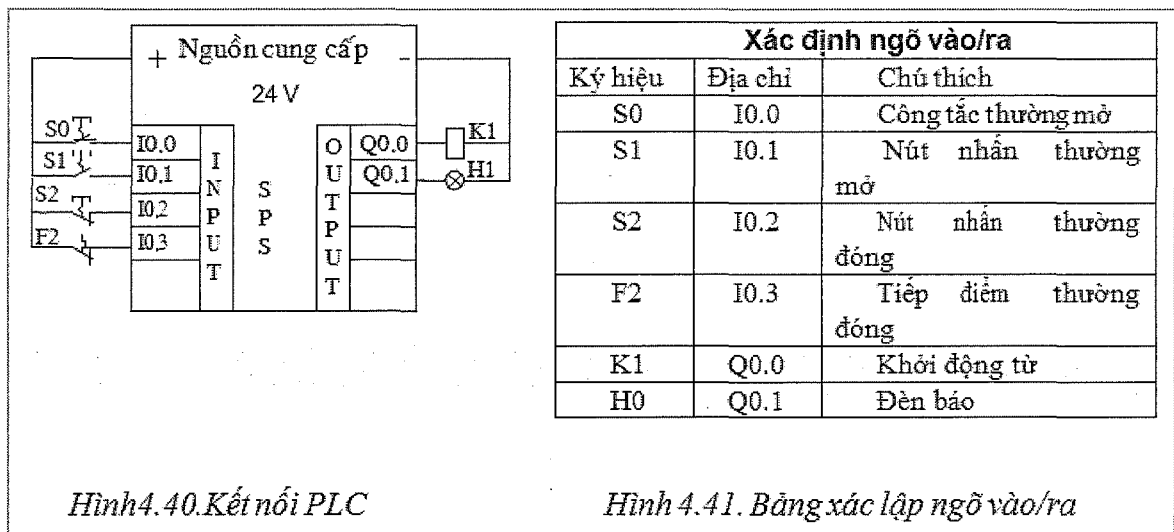
b/ Điều khiển bằng tải



Hình 4.39. Sơ đồ công nghệ

Mô tả hoạt động:

Công tắc S0 dùng để khởi động cho thiết bị và đèn H0 chỉ báo chế độ làm việc. Khi nhấn S1 động cơ M1 khởi động kéo băng tải và than đá trong thùng chứa được vận chuyển theo băng tải. Khi nhấn S2 thì băng tải dừng, nó sẽ được cắt khỏi nguồn qua bộ bảo vệ quá dòng F2



Hình 4.40. Kết nối PLC

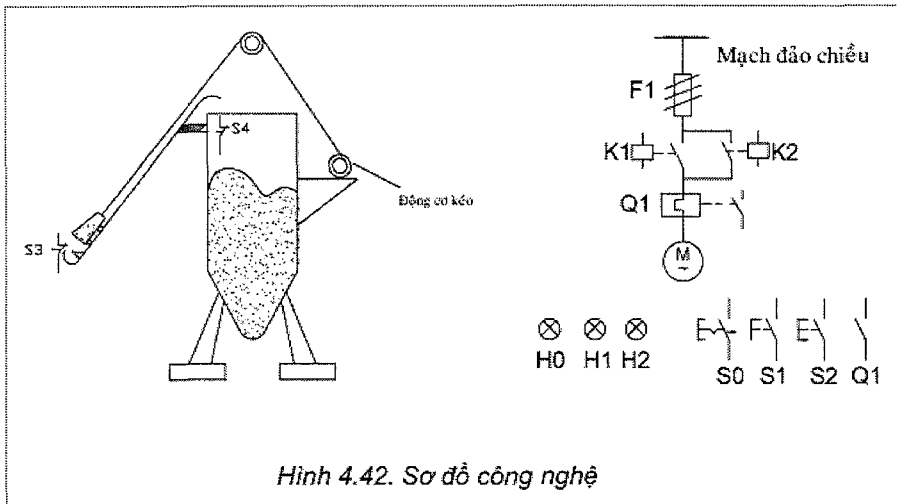
Hình 4.41. Bảng xác lập ngõ vào/ra

Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau:

Vẽ sơ đồ STL và sơ đồ LAD. Viết bảng câu lệnh mô tả mạch. Viết và thử chương trình.

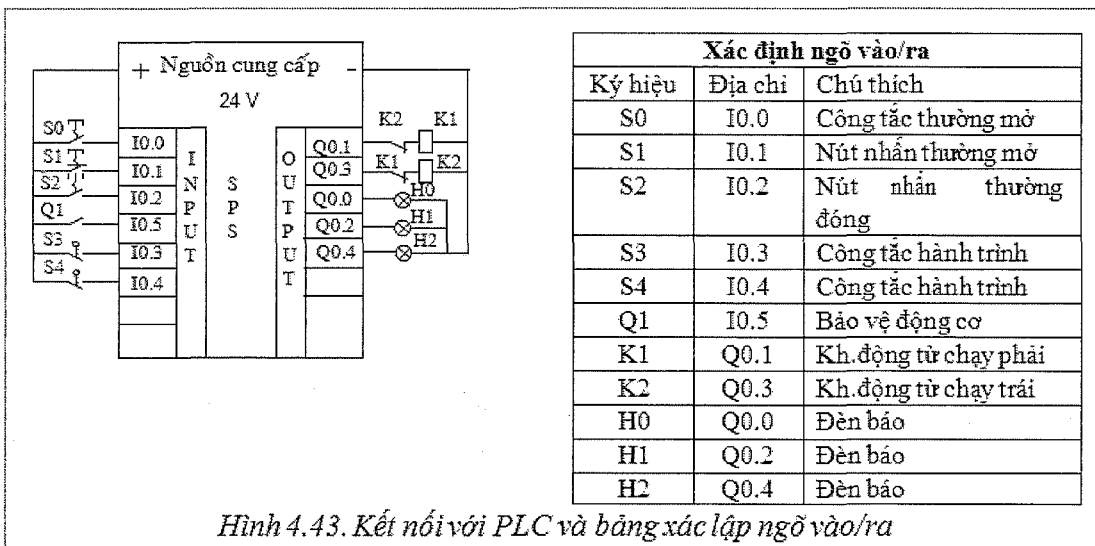
c/ Xe rót vật liệu – bể chứa:



Hình 4.42. Sơ đồ công nghệ

Mô tả hoạt động:

Công tắc S0 dùng để khởi động cho thiết bị, đèn báo H0 chỉ báo chế độ làm việc. Khi động cơ M1 có sự cố đèn H0 chớp với tần số 1Hz. Khi nhấn S1, động cơ M1 có điện và xe di chuyển lên phía trên – đèn H1 chớp với tần số 1Hz. Khi xe lên tới vị trí trên cùng dừng phải công tắc hành trình S4 thì động cơ bị ngắt mạch. Nhấn S2 động cơ có điện trở lại và di chuyển xe xuống phía dưới – đèn báo H2 chớp với tần số 1Hz. Khi xe đến vị trí cuối cùng dừng phải công tắc hành trình S3 thì động cơ bị ngắt mạch. Quá trình mới được khởi động lập lại như ban đầu.



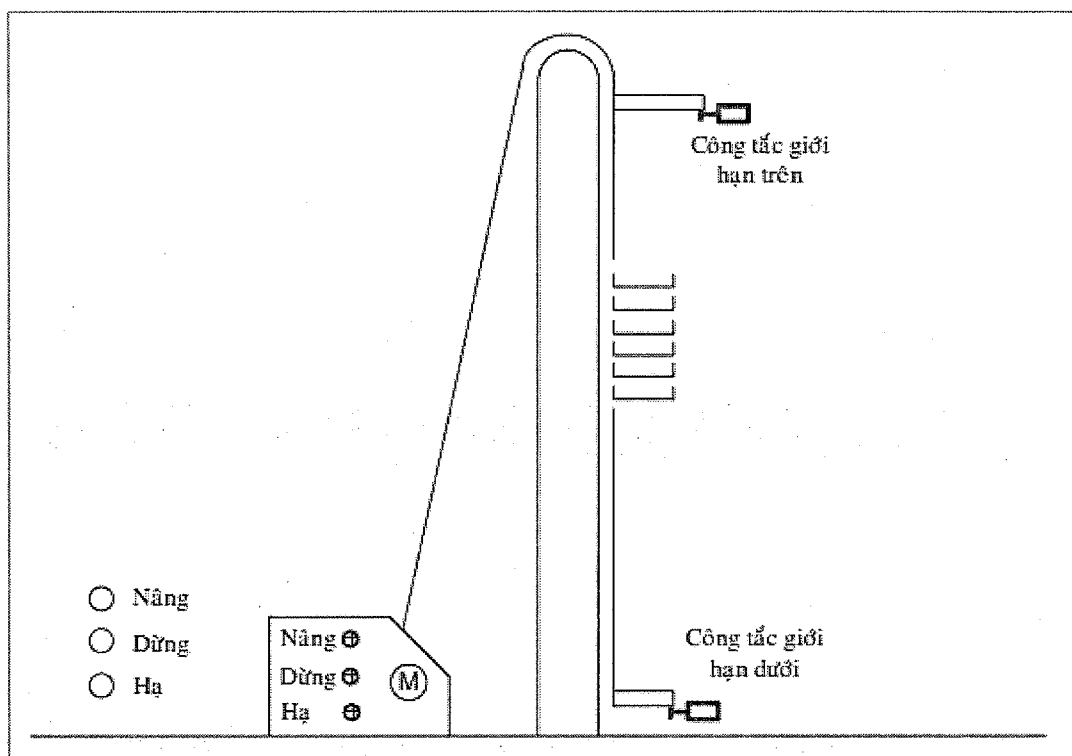
Hình 4.43. Kết nối với PLC và bảng xác lập ngõ vào/ra

Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau: Vẽ sơ đồ STL và sơ đồ LAD.

Viết bảng câu lệnh mô tả mạch. Viết và thử chương trình.

d/ Thang máy xây dựng



Mô tả hoạt động:

Thang máy xây dựng hoạt động như sau:

Khi nhấn nút nhấn nâng thì gàu sẽ chạy lên đến công tắc giới hạn trên thì gàu dừng lại. Khi nhấn nút nhấn hạ thì gàu sẽ hạ xuống đến công tắc giới hạn dưới thì gàu dừng lại. Trong khi đang di chuyển nếu nhấn nút nhấn dừng thì gàu dừng lại và sau đó có thể nâng gàu lên hay hạ gàu xuống theo mong muốn.

Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau: Vẽ sơ đồ STL và sơ đồ LAD.

Viết bảng câu lệnh mô tả mạch. Viết và thử chương trình.

Xác định ngõ vào/ra		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nâng, thường mở
S2	I0.1	Hạ, thường mở
S3	I0.2	Dừng, thường đóng
S4	I0.3	Giới hạn trên thường đóng
S5	I0.4	Giới hạn dưới thường đóng
K1	Q0.0	Gàu chạy lên
K2	Q0.1	Gàu chạy xuống
H0	Q0.2	Đèn báo nâng
H1	Q0.3	Đèn báo hạ
H2	Q0.4	Đèn báo dừng

4.4 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 làm hai loại:

- Bộ đếm lên **CTU (Count Up)**
- Bộ đếm lên và đếm xuống **CTUD (Counter Up/Down)**.

CP 212		CPU 214		CPU 215/216
Z0 – Z63		Z0 – Z127		Z0 – Z255
CTU	CTUD	CTU	CTUD	
0 - 47	48 - 63	0 - 47	48 - 79	
		80 -		

Hình 4.44. Các vùng địa chỉ của bộ đếm

4.4.1. Bộ đếm lên (Counter up)

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

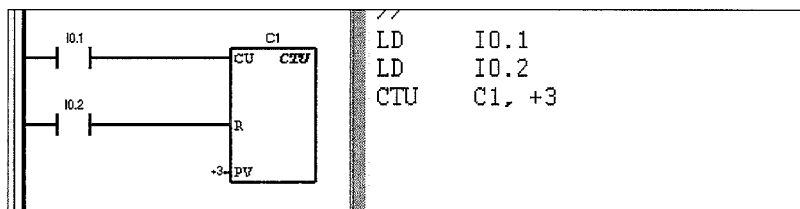
Nội dung của C-word, gọi là giá trị đếm tức thời của bộ đếm, luôn được so sánh với giá trị đặt trước (giá trị tới hạn) của bộ đếm, được ký hiệu là PV (Preset value). Khi giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước này thì bộ đếm báo ra ngoài bằng cách đặt giá trị logic 1 vào một bit đặt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

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

Vùng địa chỉ của bộ đếm được cho trong bảng ở hình 1.

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

Bộ đếm CTU được viết trong LAD, STL cũng như giản đồ thời gian được cho như hình vẽ:



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

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

+32767. Bộ đếm **CTUD** đếm xuống khi gặp sườn lên của xung vào cổng đếm xuống, ký hiệu là **CD** trong LAD. Giá trị đếm của bộ đếm giảm đi 1 ở mỗi sườn xung lên ở ngõ vào **CD**. Bộ đếm chỉ dừng lại, nếu giá trị đếm đạt đến -32767.

Nếu giá trị đếm tức thời \geq giá trị đặt trước ở ngõ vào **PV**, thì **C-bit** có giá trị bằng 1.

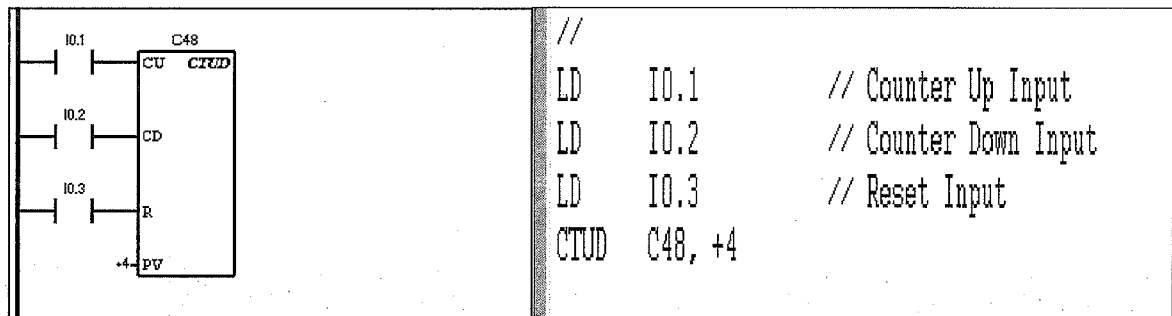
Còn các trường hợp khác **C-bit** có giá trị bằng 0.

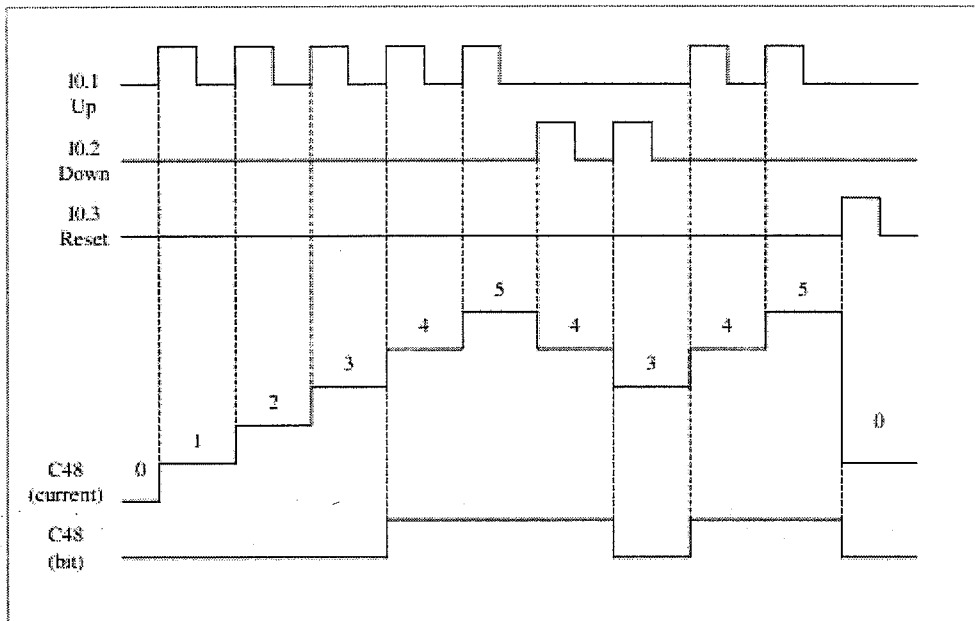
Giống như bộ đếm **CTU**, bộ đếm **CTUD** cũng có thể được đưa về trạng thái khởi phát ban đầu bằng 2 cách:

- Khi ngõ vào **R** có giá trị logic bằng 1
- Dùng lệnh **R** (reset) để reset **C-bit** bộ đếm.

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

Bộ đếm **CTU** được viết trong LAD, STL cũng như giản đồ thời gian được cho như hình vẽ:





Ví dụ:

Chương trình điều khiển máy trộn

Hình dưới là sơ đồ một bình trộn để tạo các màu sơn khác nhau. Trong sơ đồ cho thấy có hai đường ống để đưa hai loại sơn màu khác nhau làm cơ sở cho việc tạo ra màu sơn như mong muốn.

Hai cảm biến được sử dụng để báo mức trong bình:

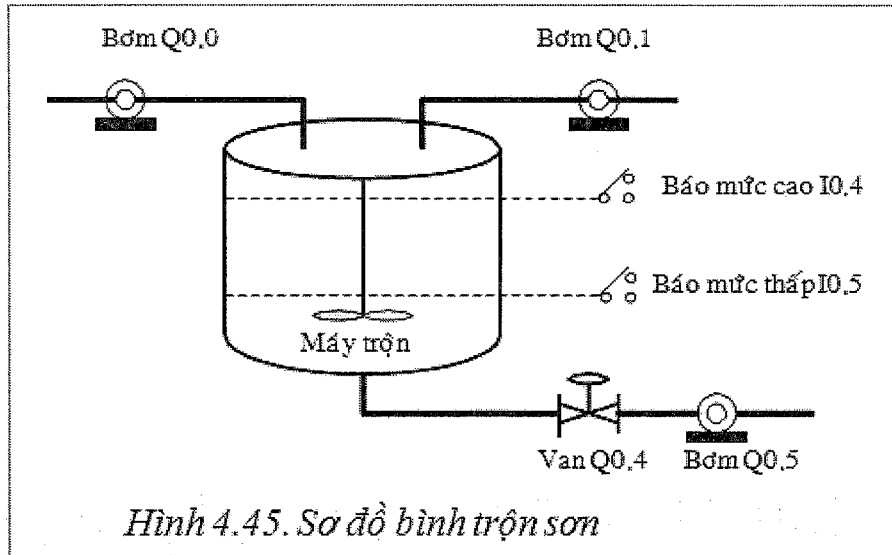
- Báo mức cao (I0.4)
- Báo mức thấp (I0.5)

Một thiết bị trộn được điều khiển bởi động cơ trộn (Q0.2)

Quá trình làm việc được thực hiện như sau: trước tiên bơm hai loại sơn khác màu vào bình, loại sơn thứ nhất được đưa vào bình bằng máy bơm được điều khiển qua Q0.0. loại sơn thứ hai được đưa vào bình bằng máy bơm thứ hai được điều khiển qua Q0.1.

Sau khi hai dung dịch trong bình đạt mức cực đại ($I0.4 = 1$) thì dừng hai máy 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 5s. Sau khi trộn xong, sản phẩm được đưa ra để rót vào các hộp đựng sơn qua van (Q0.4) và máy bơm (Q0.5). Có thể tóm tắt quá trình trộn như sau:

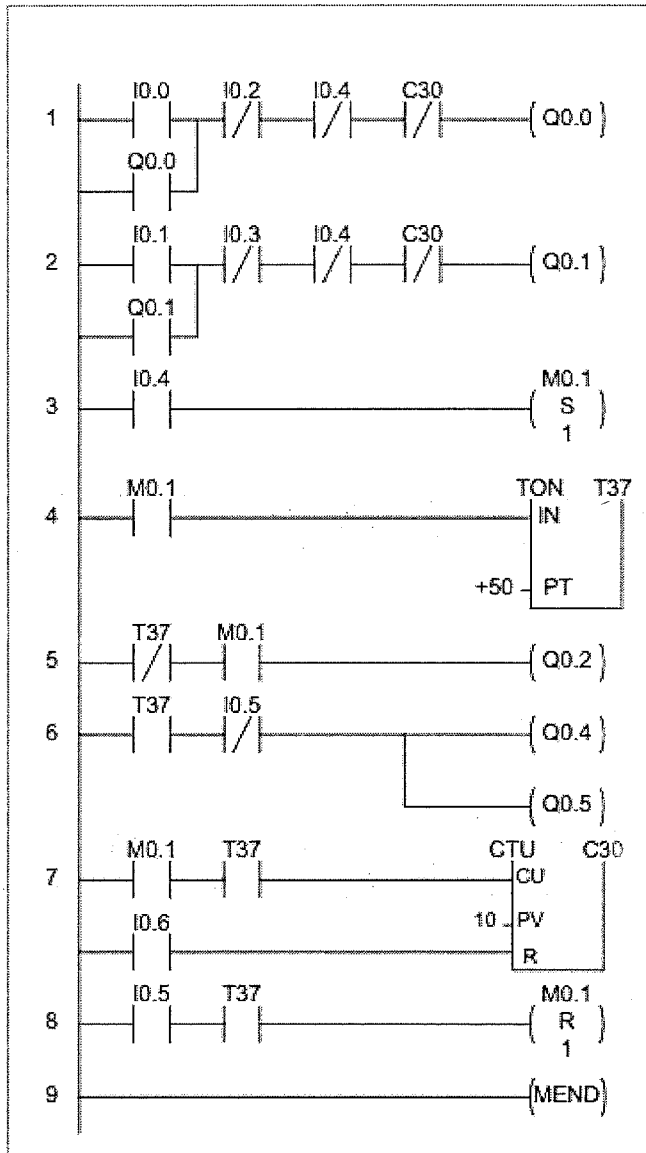
- Bước 1: Rót loại sơn thứ nhất và loại sơn thứ hai vào bình.
- Bước 2: Điều hành chế độ là việc khi đạt mức cao ($I0.4 = 1$).
- Bước 3: Điều khiển động cơ trộn và đặt thời gian trộn.
- Bước 4: Đưa sản phẩm ra khỏi bình trộn.
- Bước 5: Đếm số lần trộn. Nếu đã đủ 10 lần thì dừng sản xuất.
- Bước 6: Quay lại chế độ làm việc ở bước 1.



Chương trình được viết trong PLC ở dạng STL

<p>Network 1 Network Title</p> <hr/> <p>Network Comment</p> <pre> LD I0.0 O Q0.0 AN I0.2 AN I0.4 AN C30 = Q0.0 </pre> <p>Network 2</p> <hr/> <pre> LD I0.1 O Q0.1 AN I0.3 AN I0.4 AN C30 = Q0.1 </pre>	<p>Network 3</p> <hr/> <pre> LD I0.4 S M0.1, 1 </pre> <p>Network 4</p> <hr/> <pre> LD M0.1 TON T37, +50 </pre> <p>Network 5</p> <hr/> <pre> LDN T37 A M0.1 = Q0.2 </pre>	<p>Network 6</p> <hr/> <pre> LD T37 AN I0.5 = Q0.4 = Q0.5 </pre> <p>Network 7</p> <hr/> <pre> LD M0.1 A T37 LD I0.6 CTU C30, +10 </pre> <p>Network 8</p> <hr/> <pre> LD I0.5 A T37 R M0.1, 1 </pre>
---	--	--

Chương trình được viết trong PLC ở dạng LAD:



Xác định ngõ vào/ra

Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Bơm sơn loại 1, thường mở
S2	I0.1	Bơm sơn loại 2, thường mở
S3	I0.2	Dừng bơm sơn loại 1.
S4	I0.3	Dừng bơm sơn loại 2.
CB 1	I0.4	Cảm biến mức cao
CB 2	I0.5	Cảm biến mức thấp
K1	Q0.0	Máy bơm 1
K2	Q0.1	Máy bơm 2
K3	Q0.2	Máy trộn
K4	Q0.4	Van tháo sơn ra ngoài
K5	Q0.5	Máy bơm sơn ra ngoài

4.4.1 Ứng dụng bộ đếm

a/ Kiểm soát dây chuyền đóng hộp

Một băng chuyền đếm táo được cho theo sơ đồ công nghệ như sau:

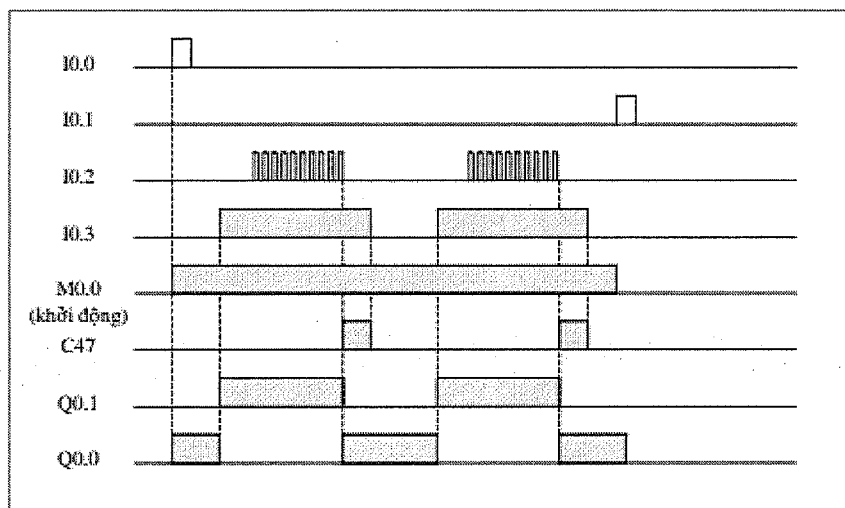
Mô tả hoạt động:

Khi nhấn nút „start“ thì dây chuyền hộp vận hành. Khi hộp đựng công tắc hành trình S3 thì dây chuyền hộp dừng lại, dây chuyền táo bắt đầu chuyển động. Cảm biến S2 được dùng để đếm số lượng táo. Khi đếm được 10 quả táo thì băng chuyền táo dừng lại và dây chuyền hộp lại bắt đầu chuyển động. Bộ đếm được đặt lại và quá trình vận hành lặp lại cho đến khi nhấn nút „stop“.

Bảng xác lập vào/ra và đồ thị thời gian được cho như sau:

Xác định ngõ vào/ra		
Ký hiệu	Địa chỉ	Mô tả
S0	I0.0	Nút khởi động start
S1	I0.1	Nút dừng stop
S2	I0.2	Cảm biến số lượng táo
S3	I0.3	Công tắc hành trình
K1	Q0.0	Đ.cơ băng chuyền thùng
K1	Q0.1	Đ.cơ băng chuyền táo

Đồ thị thời gian



Nhiệm vụ:

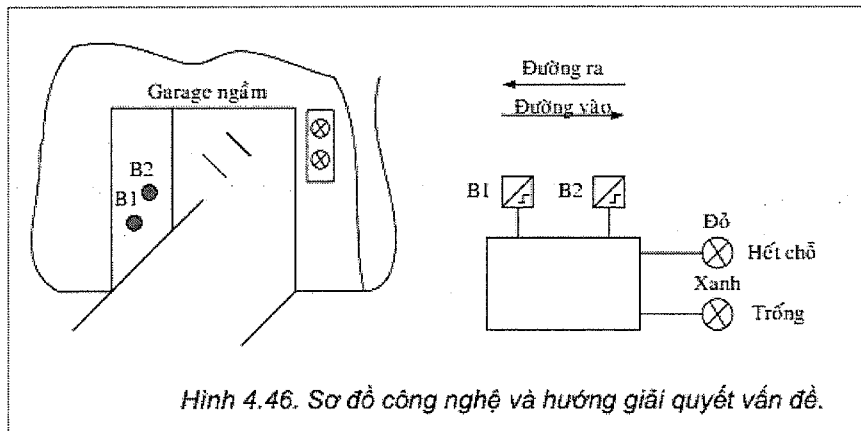
- Vẽ sơ đồ kết nối với PLC Vẽ lưu đồ điều khiển
- Viết chương trình ở dạng LAD, STL
- Nạp vào PLC và kiểm tra chương trình .

b/ Kiểm soát chỗ cho Garage ngầm

ngầm Mô tả hoạt động:

Một Garage ngầm có 20 chỗ đậu xe. Ở ngõ vào có 2 đèn báo: đèn đỏ đã báo hiệu Garage đã hết chỗ, đèn xanh báo hiệu Garage còn trống. Đường vào và đường ra chỉ cho phép một xe chạy.

Sơ đồ công nghệ được cho như hình vẽ sau:



Xác định ngõ vào/ra		
Ký hiệu	Địa chỉ	Mô tả
B1	I0.0	CB quang 1, xe vào
B2	I0.1	CB quang 2, xe ra
H1	Q0.0	Đèn báo đỏ
H2	Q0.1	Đèn báo xanh

Nhiệm vụ:

Vẽ sơ đồ kết nối với PLC Vẽ lưu đồ điều khiển

Viết chương trình ở dạng LAD, STL.

Nạp vào PLC và kiểm tra chương trình .

4.5 Bài tập ứng dụng

4.5.1. Mạch khởi động động cơ xoay chiều 3 pha

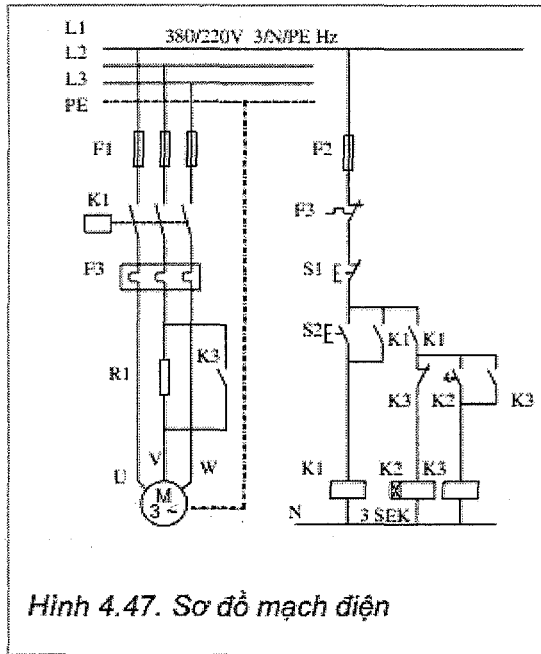
Mô tả hoạt động: Khi nhấn S2, K1 và rơ le K2 có điện – động cơ được khởi động, sau một khoảng thời gian (đã được chỉnh định trước) động cơ đạt tới tốc độ định mức thì K3 có điện và tiếp điểm K3 đóng lại làm ngắn mạch điện trở R1.

Khi nhấn S1 động cơ bị ngắt mạch.

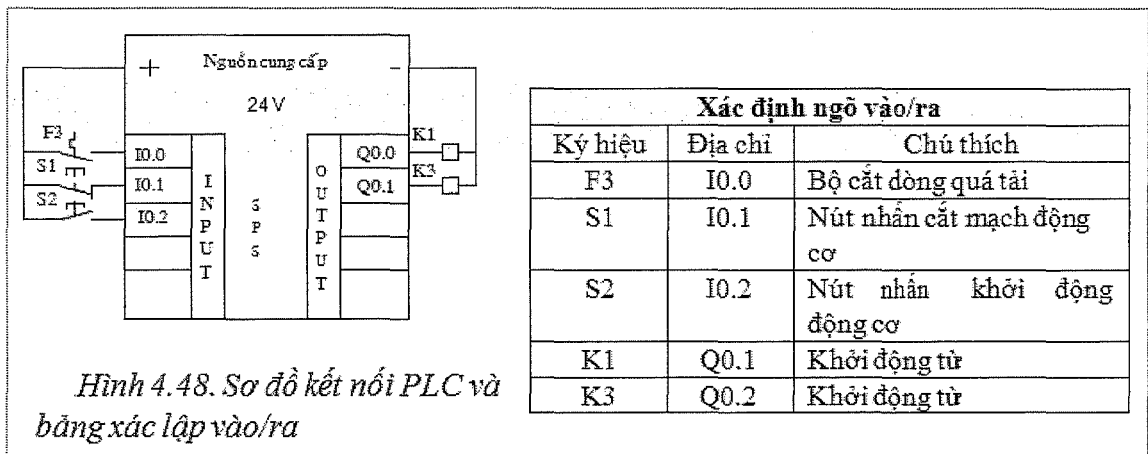
Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau:

Lập sơ đồ STL và sơ đồ LAD. Viết bảng câu lệnh mô tả mạch Nhập và thử chương

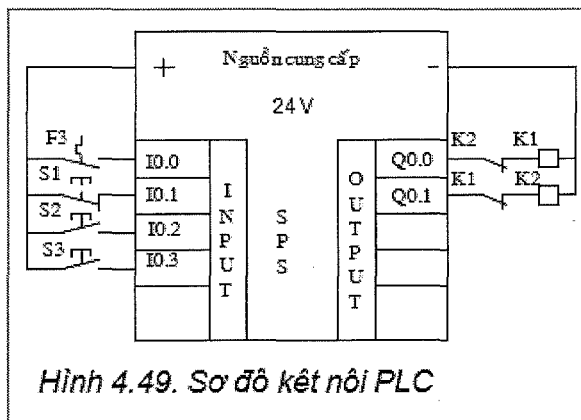


Hình 4.47. Sơ đồ mạch điện

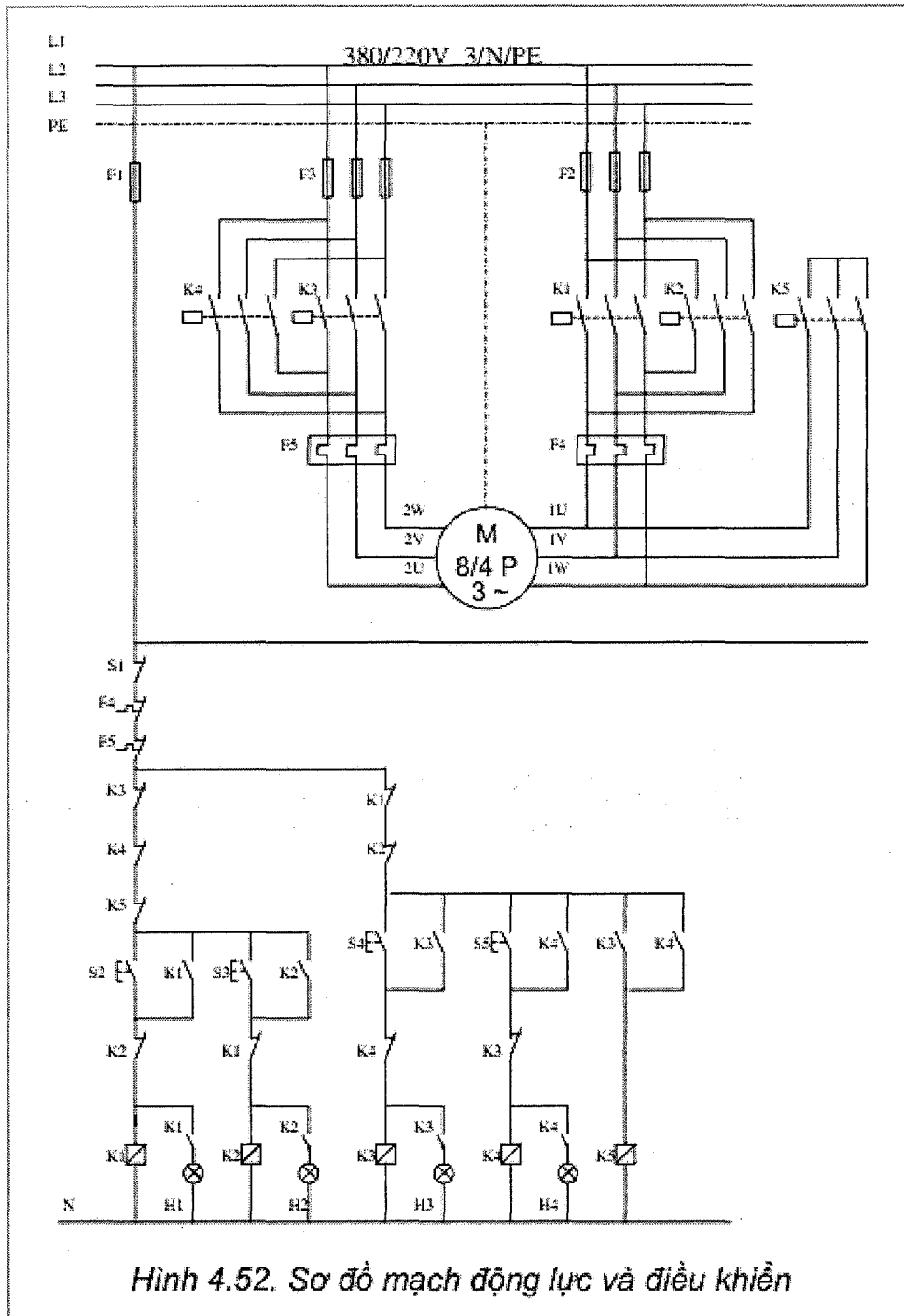


Hình 4.48. Sơ đồ kết nối PLC và bảng xác lập vào/ra

4.5.2. Mạch đảo chiều của động cơ xoay chiều 3 pha Mô tả hoạt động



Hình 4.49. Sơ đồ kết nối PLC



4.5.4. Mạch khởi động sao – tam giác cho 2 chiều quay Mô tả hoạt động

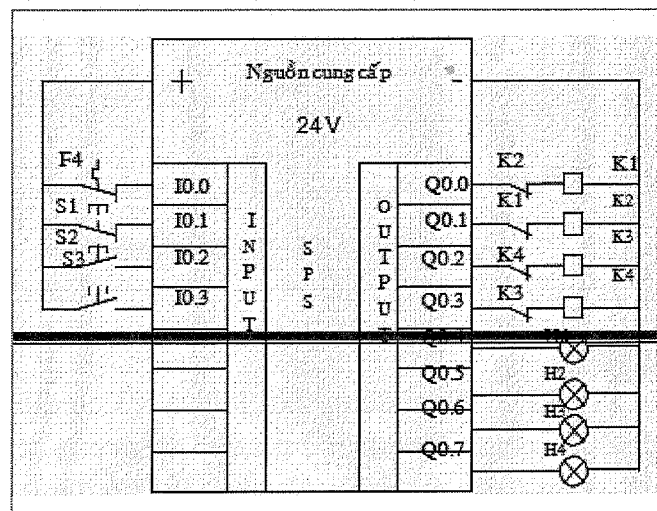
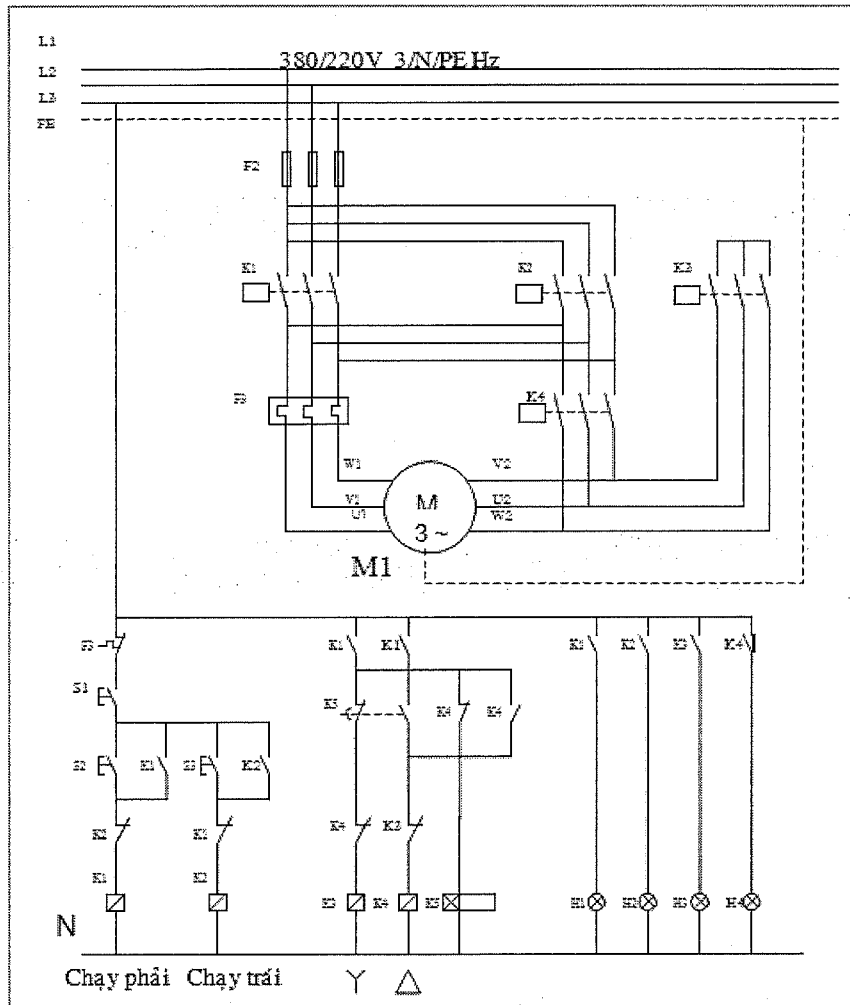
Khi nhấn S2, K1 có điện và tự duy trì, đồng thời role thời gian K5 và K3 có điện, động cơ chạy phải ở chế độ nối Y. Sau một thời gian đã được chỉnh định trước K3 bị hở mạch và K4 có điện, động cơ chạy phải ở chế độ tam giác. Nhấn S1 động cơ được cắt khỏi nguồn điện. Bây giờ nhấn S3, K2 có điện, đồng thời K5 và K3 có điện, động cơ chạy trái ở chế độ nối sao. Quá trình lập lại giống như trường hợp trên. Các khởi động từ K1 và K2 được khoá chéo lẫn nhau.

Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau: Vẽ sơ đồ STL và sơ đồ LAD.

Viết bảng câu lệnh mô tả mạch

Viết và thử chương trình

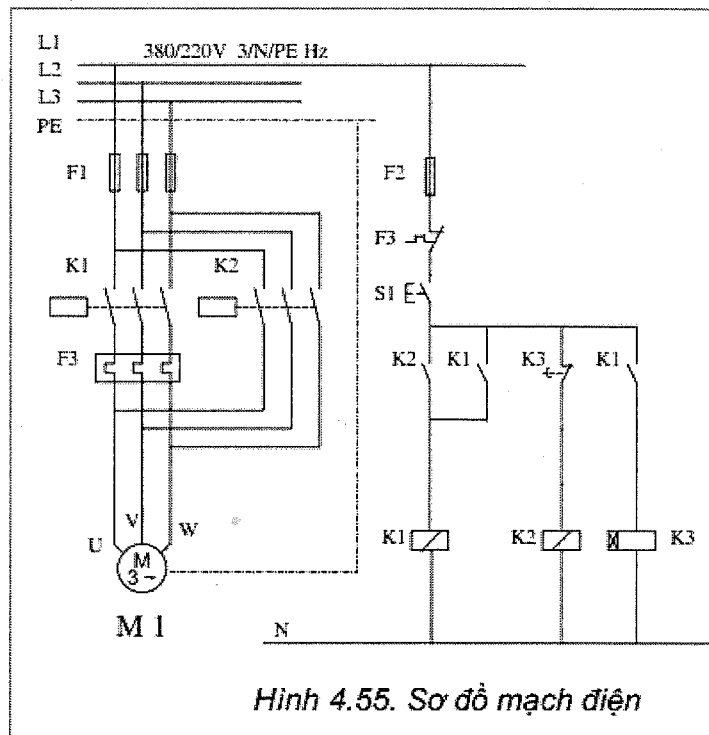


Hình 4.53. Sơ đồ mạch động lực, mạch điều khiển và kết nối PLC

Xác định ngõ vào/ra		
Ký hiệu	Địa chỉ	Mô tả
F3	I0.0	Bộ cắt dòng quá tải
S1	I0.1	Nút nhấn thường đóng – cắt mạch
S2	I0.2	Nút nhấn thường mở – Chạy phải
S3	I0.3	Nút nhấn thường mở – Chạy trái
K1	Q0.0	Khởi động từ chạy phải
K2	Q0.1	Khởi động từ chạy trái
K3	Q0.2	Chạy Y
K4	Q0.3	Chạy tam giác
H1	Q0.4	Đèn báo chạy phải
H2	Q0.5	Đèn báo chạy trái
H3	Q0.6	Đèn báo chạy nối Y
H4	Q0.7	Đèn báo chạy nối tam giác

Hình 4.54. Bảng xác lập trạng thái vào/ra

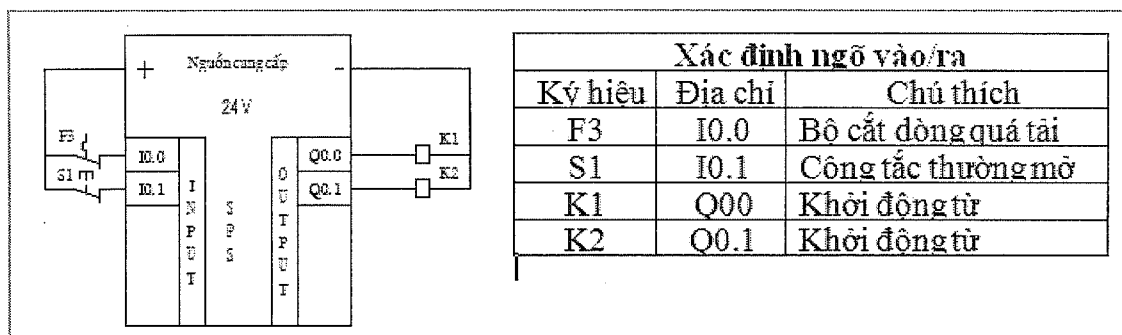
4.5.5. Mạch khởi động cho trường hợp có tải nặng



Hình 4.55. Sơ đồ mạch điện

Mô tả hoạt động:

Khi bật công tắc S1, K2 có điện; đồng thời K1 và K3 có điện, sau một khoảng thời gian đã được chỉnh định trước thì tiếp điểm của role thời gian K3 nhả ra làm cho K2 mất điện. S1 dùng để cắt điện cho động cơ.



Nhiệm vụ:

Hãy thực hiện mạch theo các yêu cầu sau: Vẽ sơ đồ STL và sơ đồ LAD.
Viết bảng câu lệnh mô tả mạch
Viết và thử chương trình

4.5.6. Điều khiển đèn quảng cáo

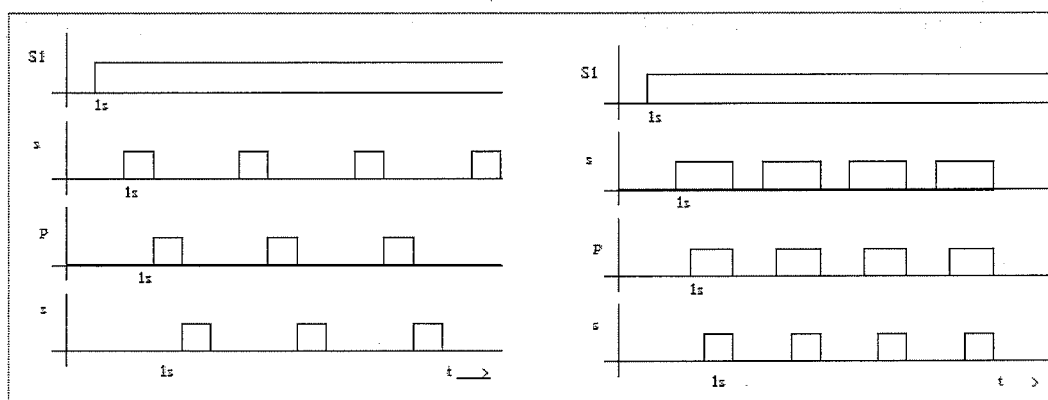
Mô tả hoạt động

Việc chiếu sáng quảng cáo cần được điều khiển theo lập trình, điều khiển các chữ cái PLC sáng theo những yêu cầu sau:

- Hoạt động theo sơ đồ thời gian 1
- Hoạt động theo sơ đồ thời gian 2
- Có thể hoạt động theo hai sơ đồ thời gian trên với công tắc chuyển mạch S2.

Nhiệm vụ:

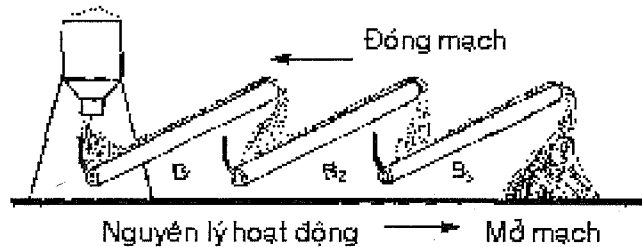
Viết chương trình điều khiển.
Nhập và thử chương trình
Đồ thị thời gian 1:



Đồ thị thời gian 2:

4.5.7. Điều khiển tuần tự: Sự điều khiển một thiết bị vận chuyển rời cần thỏa mãn các điều kiện đóng mạch tuần tự sau:

- Sau 3s Sau 5s
 1. Trình tự đóng mạch: M1 → M2 → M3
 Sau 5s Sau 3s
 2. Trình tự mở mạch: M3 → M2 → M1



Nhiệm vụ:

- Lập bảng xác lập vào/ra.
- Vẽ sơ đồ kết nối với PLC.
- Viết chương trình, nạp chương trình vào PLC và chạy thử.

4.5.8. Điều khiển cửa cuốn

Mô tả hoạt động:

Một cửa cuốn được kéo bởi động cơ M1. Cửa cuốn này có thể được hoạt động ở hai chế độ tay và tự động:

* Ở chế độ tay:

- Khi nhấn nút nhấn nâng S1 thì động cơ sẽ kéo cửa cuốn lên, khi gặp công tắc giới hạn trên S3 thì cửa cuốn dừng lại.
- Khi nhấn nút nhấn hạ S2 thì động cơ sẽ kéo cửa cuốn xuống, khi gặp công tắc giới hạn dưới S4 thì cửa cuốn dừng lại.
- Trong quá trình nâng hay hạ cửa cuốn sẽ được dừng khi nút nhấn dừng S0 được ấn.

* Ở chế độ tự động:

- Sáng: Đúng 7:00 giờ, cửa cuốn tự động kéo lên.
- Chiều: Đúng 17:00 giờ, cửa cuốn tự động đóng xuống.

Nhiệm vụ:

- Lập bảng xác lập vào/ra.
- Vẽ sơ đồ kết nối với PLC.
- Viết chương trình. Nạp chương trình vào PLC và thử chương trình.

Hướng dẫn: Sử dụng hàm đọc đồng hồ thời gian thực: READ_RTC (Read Real-Time Clock).

4.6. Lệnh nhảy và lệnh 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. Chúng cho phép chuyển thứ tự thực hiện, đáng lẽ là lệnh tiếp theo, tới một lệnh bất cứ nào khác của chương trình, trong đó nơi điều khiển chuyển đến phải được đánh dấu trước bằng một nhãn chỉ đích. 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 của chương trình xử lý ngắt được khai báo ở đầu chương trình. Không thể dùng lệnh nhảy JMP để điều khiển từ chương trình chính vào một nhãn bất kỳ trong chương trình con hoặc trong chương trình xử lý ngắt.

5.2. Chức năng so sánh:

Các phép so sánh có thể sử dụng là so sánh $=$, $<$, $>$, \geq , $<=$ và chỉ có thể áp dụng cho Byte, số nguyên I, số nguyên kép DI và số thực R.

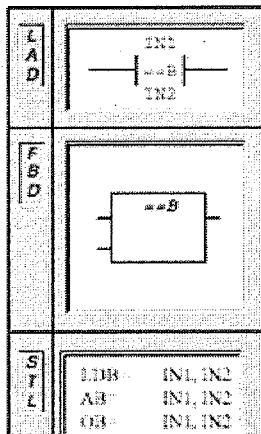
Dữ liệu tại ngõ vào IN1 được so sánh với dữ liệu tại ngõ vào IN2

Trong soạn thảo LAD thì tiếp điểm sẽ ON khi thỏa mãn điều kiện so sánh.

Trong soạn thảo STL các lệnh Load, AND hoặc OR sẽ = 1 khi phép so sánh là

True

So sánh Byte:



$IN1=IN2, IN1 > IN2, IN1 \geq IN2, IN1 < IN2, IN1 \leq IN2,$

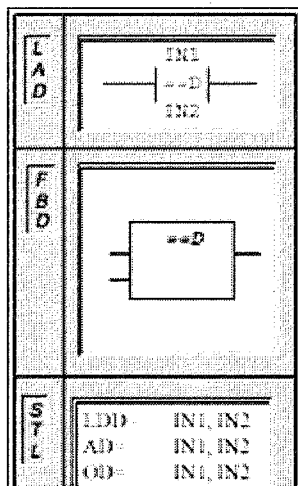
Inputs/Outputs	Operands	Data Types
Inputs	IB, QB, MB, SMB, VB, SB, LB, AC, Constant, *VO, *AC, *LD	BYTE
Outputs (FBD)	I, Q, M, SM, T, C, V, S, L, Power Flow	BOOL

So sánh số nguyên kép Doubleword:

$IN1=IN2, IN1 > IN2, IN1 \geq IN2, IN1 < IN2, IN1 \leq IN2, IN1 \diamond IN2$

So sánh số nguyên cần chú ý đến dấu ($16\#7FFFFFFF > 16\#8000000$) Phạm vi so sánh từ $-2.147.483.647$ đến $+2.147.483.647$.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:



Inputs/Outputs	Operands	Data Types
Inputs	ID, QD, MD, SD, SMD, VD, LD, HC, AC, Constant, *VD, *AC, *LD	DINT
Outputs (FBD)	I, Q, M, SM, T, C, V, S, L, Power Flow	BOOL

Dùng để so sánh 2 số thực : IN1 với IN2. Các phép so sánh có thể sử dụng là: IN1=IN2, IN1 > IN2, IN1 >=IN2, IN1<IN2, IN1<=IN2, IN1<>IN2

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

Inputs/Outputs	Operands	Data Types
Inputs	ID, GD, MD, SD, SMD, VD, LD, AC, Constant, *VB, *AC, *LD	REAL
Outputs (FBD)	I, Q, M, SM, T, C, V, S, L, Power Flow	BOOL

The diagram shows three logic symbols for comparison. The first is a Ladder Logic (LAD) symbol with two inputs labeled IN1 and IN2. The second is a Function Block Diagram (FBD) symbol with a single input and an output labeled =>R. The third is a Statement List (STL) symbol with three lines: LDR- IN1, IN2; AK- IN1, IN2; and OR- IN1, IN2.

Chức năng dịch chuyển:

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

Bảng giới hạn vùng toán hạng và dạng dữ liệu

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

The diagram shows logic symbols for byte shift. The first is a Ladder Logic (LAD) symbol with two versions: SHR_B and SHL_B. Each has an input IN, an output OUT, and a shift count input N. The second is a Function Block Diagram (FBD) symbol with two versions: SHR_B and SHL_B, each with IN, OUT, and N inputs. The third is a Statement List (STL) symbol with two lines: SRB OUT, N and SLB OUT, N. Below the STL symbol are three checkboxes with numbers 221, 222, and 224.

Dịch phải Word SHR_W và Dịch trái Word SHL_W:

L A D	SHR_W
	EN ENO
F B D	IN OUT
	N
SHL_W	
EN ENO	
IN OUT	
N	
S T L	SRW OUT, N
	SLW OUT, N
<input checked="" type="checkbox"/> 221 <input checked="" type="checkbox"/> 222 <input checked="" type="checkbox"/> 224	

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.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

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

5.3. Chức năng chuyển đổi (Converter):

Chuyển đổi số BCD_I và I_BC

L A D	BCD_I
	EN ENO
F B D	IN OUT
I_BCD	
EN ENO	
IN OUT	
S T L	BCDI OUT
	IBCD OUT
<input checked="" type="checkbox"/> 221 <input checked="" type="checkbox"/> 222 <input checked="" type="checkbox"/> 224	

Lệnh chuyển đổi số BCD sang số Integer (BCD_I) sẽ thực hiện việc chuyển đổi số BCD tại ngõ vào IN sang giá trị số nguyên I và chứa kết quả vào địa chỉ xác định tại ngõ ra OUT. Giá trị có thể nhập tại ngõ IN từ 0 đến 9999BCD.

Khi xảy ra lỗi chuyển đổi thì trạng thái ENO = 0.

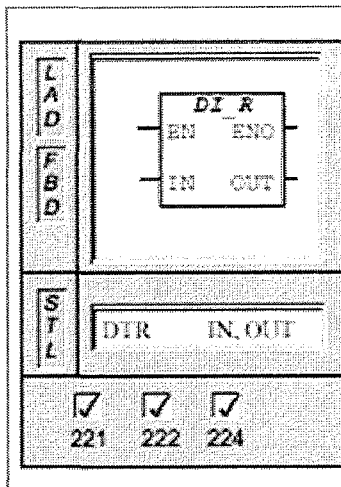
Lệnh chuyển đổi số I sang số BCD sẽ thực hiện việc chuyển đổi số I tại ngõ vào IN sang giá trị số BCD và chứa kết quả vào địa chỉ xác định tại ngõ ra OUT. Giá trị có thể nhập tại ngõ IN từ 0 đến 9999 Integer.

Khi xảy ra lỗi chuyển đổi thì trạng thái ENO = 0.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

Inputs/Outputs	Operands	Data Types
IN	VW, T, C, IW, QW, MW, SMW, LW, AC, AIW, Constant, *VD, *AC, SW, *LD	WORD
OUT	VW, T, C, IW, QW, MW, SMW, LW, AC, *VD, *AC, SW, *LD	WORD

Chuyển đổi số nguyên kép DI sang số thực R:

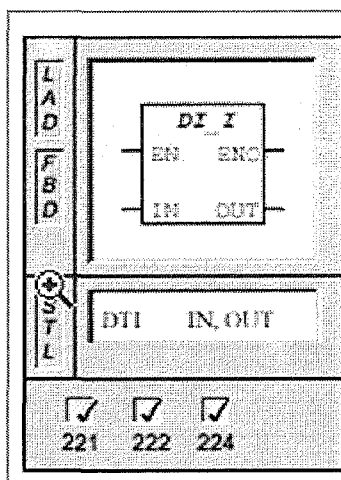


Lệnh chuyển đổi một số nguyên kép DI 32 Bit sang một số thực R và đặt kết quả vào địa chỉ được xác định tại ngõ ra OUT.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

Inputs/Outputs	Operands	Data Types
IN	VD, ID, QD, MD, SMD, AC, LD, HC, Constant, *VD, *AC, SD, *LD	DINT
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	REAL

Chuyển đổi số nguyên kép DI sang số nguyên I:

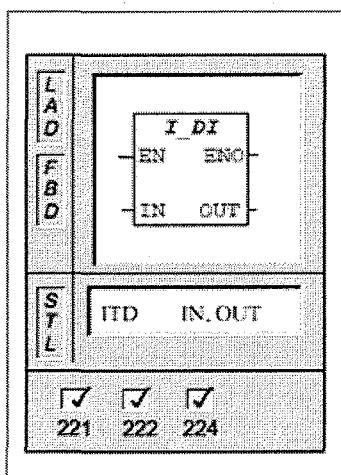


Lệnh chuyển đổi số DI_I chuyển đổi giá trị số DI tại ngõ vào IN sang một giá trị số nguyên I và đặt kết quả tại ngõ OUT, nếu phép chuyển đổi bị tràn (kết quả lớn hơn khả năng chứa của ngõ OUT) thì ngõ ra không thay đổi và trạng thái ENO = 0.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

Inputs/Outputs	Operands	Data Types
IN	VD, ID, QD, MD, SMD, AC, LD, HC, Constant, *VD, *AC, SD, *LD	DINT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	INT

Chuyển đổi số nguyên kép DI sang số nguyên I:

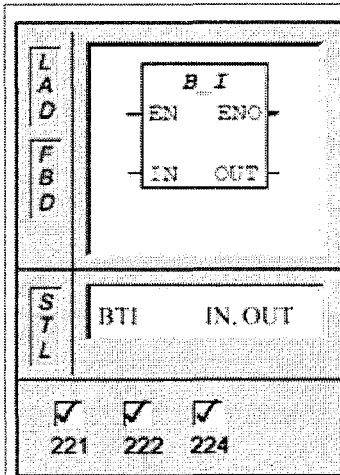


Lệnh chuyển đổi số I_DI chuyển đổi giá trị số I tại ngõ vào IN sang một giá trị số nguyên kép DI và đặt kết quả tại ngõ OUT, nếu phép chuyển đổi bị tràn (kết quả lớn hơn khả năng chứa của ngõ OUT) thì ngõ ra không thay đổi và trạng thái ENO = 0.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

Inputs/Outputs	Operands	Data Types
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, Constant, *AC, *VD, *LD	INT
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	DINT

Chuyển đổi Byte sang số nguyên I:

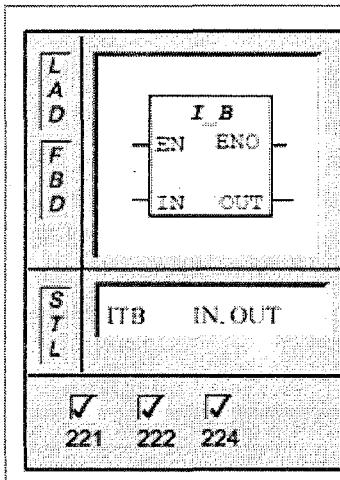


Lệnh chuyển đổi **B_I** chuyển đổi dữ liệu chứa trong Byte có địa chỉ tại ngõ IN sang giá trị số nguyên, kết quả chứa vào biến xác định tại ngõ ra OUT.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

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

Chuyển đổi số nguyên I sang Byte:



Lệnh chuyển đổi **I_B** chuyển đổi giá trị chứa trong word có địa chỉ tại ngõ IN sang giá trị Byte, kết quả chứa vào biến xác định tại ngõ ra OUT. Các số nguyên có thể chuyển đổi là từ 0 đến 255.

Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

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

5.4. Chức năng toán học:

Các lệnh số học dùng để thực hiện các phép tính số học trong chương trình. Trong LAD, bốn hộp toán (math box) thực hiện các phép tính cộng, trừ 16-bit và 32-bit. Hộp nhân (multiply box) nhân hai số nguyên 16-bit và kết quả là một số nguyên 32-bit. Hộp chia (divide box) chia hai số 16-bit, thương là một số 16-bit và dư cũng 16-bit được nạp vào từ ngay trước. Nếu lập trình bằng LAD, có thể tiết kiệm ô nhớ bằng cách sử dụng đầu vào IN1 đồng thời cũng là đầu ra OUT

Trong STL, lệnh thực hiện bốn phép tính số học được qui định cho toán hạng 16-bit và 32-bit. Hộp nhân thực hiện phép nhân hai số nguyên 16-bit và tích số là một số nguyên 32-bit. Lệnh chia thực hiện phép chia một số nguyên 16-bit với 16-bit cuối của một số nguyên 32-bit. Kết quả là một giá trị từ kép (32-bit) trong đó từ thấp (từ bit 0 đến bit 15) là thương số và từ cao (từ bit 16 đến bit 31) là số dư của phép tính.

Phép cộng**ADD_I (LAD)****+I (STL)**

Lệnh thực hiện phép cộng các số nguyên 16-bit IN1 và IN2. Trong LAD kết quả là một số nguyên 16-bit được ghi vào OUT, tức là $IN1 + IN2 = OUT$.

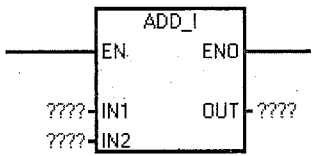
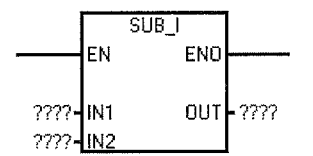
Trong STL, kết quả cũng là một giá trị 16-bit nhưng được ghi lại vào IN2, tức là $IN1 + IN2 = IN2$.

Phép trừ**SUB_I (LAD)****-I (STL)**

Lệnh thực hiện phép trừ các số nguyên 16-bit IN1 và IN2. Trong LAD kết quả là một số nguyên 16-bit và được ghi vào OUT, tức là $IN1 - IN2 = OUT$.

Trong STL, kết quả cũng là một giá trị 16-bit nhưng được ghi lại vào IN2, tức là $IN1 - IN2 = IN2$.

Cú pháp dùng lệnh cộng, trừ hai số nguyên 16-bit trong LAD và STL như sau:

LAD	STL	Toán hạng
	+I IN2	IN1, IN2 VW, T, C, QW, MW, SMW, (word) AC, AIW, hằng số, *VD, *AC
	- I IN2	OUT VW, T, C, QW, MW, SMW, (word) AC, *VD, *AC

ADD_DI (LAD)**+D (STL)**

Lệnh thực hiện phép cộng các số nguyên 32-bit IN1 và IN2. Trong LAD kết quả là một số nguyên 32-bit được ghi vào OUT, tức là $IN1 + IN2 = OUT$.

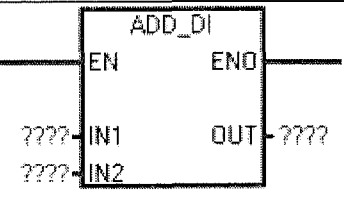
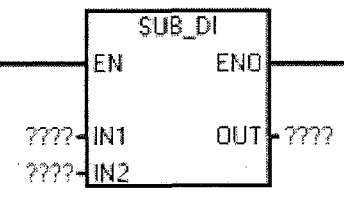
Trong STL, kết quả cũng là một giá trị 32-bit nhưng được ghi lại vào IN2, tức là $IN1 + IN2 = IN2$.

SUB_DI (LAD)**-D (STL)**

Lệnh thực hiện phép trừ các số nguyên 32-bit IN1 và IN2. Trong LAD kết quả là một số nguyên 32-bit và được ghi vào OUT, tức là $IN1 - IN2 = OUT$.

Trong STL, kết quả cũng là một giá trị 32-bit nhưng được ghi lại vào IN2, tức là $IN1 - IN2 = IN2$.

Cú pháp dùng lệnh cộng, trừ hai số nguyên 32-bit trong LAD và STL như sau:

LAD	STL	Toán hạng
	+D IN1 IN2	IN1, IN2 VW, T, C, QW, MW, SMW, (từ kép) AC, AIW, hằng số, *VD, *AC
	-D IN1 IN2	OUT VW, T, C, QW, MW, SMW, (từ kép) AC, *VD, *AC

ADD_R (LAD)**+R (STL)**

Lệnh thực hiện phép cộng các số thực 32-bit IN1 và IN2. Trong LAD kết quả là một số thực 32-bit được ghi vào OUT, tức là $IN1 + IN2 = OUT$.

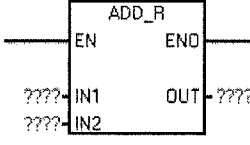
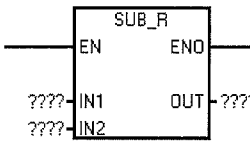
Trong STL, kết quả cũng là một giá trị thực 32-bit nhưng được ghi lại vào IN2, tức là $IN1 + IN2 = IN2$.

SUB_R (LAD)**-R (STL)**

Lệnh thực hiện phép trừ các số thực 32-bit IN1 và IN2. Trong LAD kết quả là một số thực 32-bit và được ghi vào OUT, tức là $IN1 - IN2 = OUT$.

Trong STL, kết quả cũng là một giá trị thực 32-bit nhưng được ghi lại vào IN2, tức là $IN1 - IN2 = IN2$

Cú pháp dùng lệnh cộng, trừ hai số thực 32-bit trong LAD và STL như sau:

LAD	STL	Toán hạng
	+R IN1 IN2	IN1, IN2 VD, ID, QD, MD, SMD, (từ kép) AC, HC, hằng số, *VD, *AC
	-R IN1 IN2	OUT VD, ID, QD, SMD, AC (từ kép) *VD, *AC

Phép nhân

MUL

Trong LAD: Lệnh thực hiện phép nhân hai số nguyên 16-bit IN1 và IN2 và cho ra kết quả 32-bit chứa trong từ kép OUT (4 byte).

Trong STL: Lệnh thực hiện phép nhân giữa số nguyên 16-bit n1 và số nguyên chứa trong từ thấp (từ bit 0 đến bit 15) của toán hạng 32-bit n2 (4 byte). Kết quả 32-bit được ghi lại vào n2.

Cú pháp dùng lệnh nhân hai số nguyên trong LAD và STL như sau:

LAD	STL	Toán hạng
	MUL n1 n2	n1, IN1, IN2 VW, T, C, IW, QW, MW (word) AC, AIW, hằng số, *VD, *AC N2, OUT VD, ID, QD, SMD, AC (từ kép) *VD, *AC

Phép chia

DIV

Trong LAD: Lệnh thực hiện phép chia số nguyên 16-bit IN1 cho số nguyên 16-bit IN2. Kết quả 32-bit chứa trong từ kép OUT (4 byte) gồm thương số ghi trong mảng 16-bit từ bit 0 đến bit 15 (từ thấp) và phần dư cũng 16 bit ghi trong mảng từ bit 16 đến bit 31 (từ cao).

Trong STL: Lệnh thực hiện phép chia số nguyên 16-bit n1 cho số nguyên 16-bit nằm trong từ thấp (từ bit 0 đến bit 15) của toán hạng 32-bit n2 (4 byte). Kết quả 32-bit được ghi lại vào n2 bao gồm thương số ghi trong mảng 16-bit từ bit 0 đến bit 15 (từ thấp) và phần dư ghi trong mảng 16 bit từ bit 16 đến bit 31 (từ cao).

Cú pháp dùng lệnh chia hai số nguyên trong LAD và STL như sau:

LAD	STL	Toán hạng
	DIV n1 n2	n1, IN1, IN2 VW, T, C, IW, QW, MW (word) AC, AIW, hằng số, *VD, *AC N2, OUT VD, ID, QD, SMD, AC (từ kép) *VD, *AC

MUL_R (LAD)

*R (STL)

Trong LAD: Lệnh thực hiện phép nhân hai số thực 32-bit IN1 và IN2 và cho ra kết quả 32-bit chứa trong từ kép OUT (4 byte).

Trong STL: Lệnh thực hiện phép nhân giữa số thực 32-bit IN1 và IN2. Kết quả 32-bit được ghi lại vào IN2.

Cú pháp dùng lệnh nhân hai số thực trong LAD và STL như sau:

LAD	STL	Toán hạng
	*R IN1 IN2	IN1, IN2 VD, ID, QD, MD, SMD, (từ kép) AC, HC, hằng số, *VD, *AC OUT VD, ID, QD, SMD, AC (từ kép) *VD, *AC

DIV_R (LAD)

/R (STL)

Trong LAD: Lệnh thực hiện phép chia hai số thực 32-bit IN1 cho số thực 32-bit IN2 và cho ra kết quả 32-bit chứa trong từ kép OUT (4 byte).

Trong STL: Lệnh thực hiện phép chia số thực 32-bit IN1 cho số thực IN2. Kết quả 32-bit được ghi lại vào IN2.

Cú pháp dùng lệnh chia hai số thực trong LAD và STL như sau:

LAD	STL	Toán hạng
	/R IN1 IN2	IN1, IN2 VD, ID, QD, MD, SMD, (từ kép) AC, HC, hằng số, *VD, *AC OUT VD, ID, QD, SMD, AC (từ kép) *VD, *AC

Phép lấy căn bậc hai

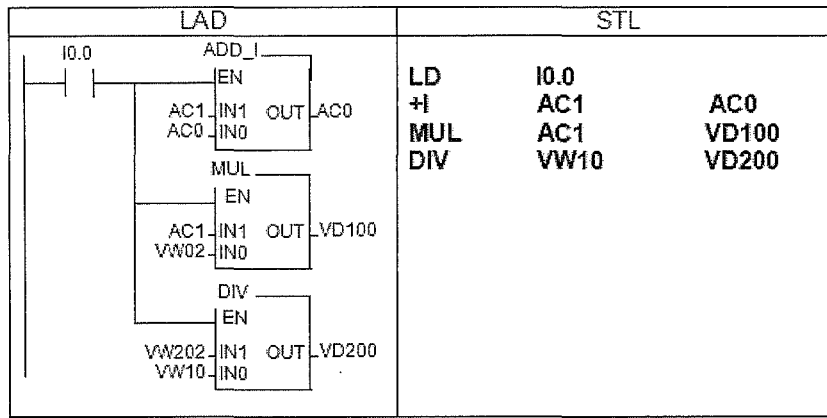
SQRT

Lệnh thực hiện phép lấy căn bậc hai của số thực 32-bit IN. Kết quả cũng là một số 32-bit được ghi vào từ kép OUT (4 byte).

Cú pháp dùng lệnh chia hai số thực trong LAD và STL:

LAD	STL	Toán hạng
	SQRT IN OUT	IN VD, ID, QD, MD, SMD, (từ kép) AC, HC, hằng số, *VD, *AC OUT VD, ID, QD, SMD, AC (từ kép) *VD, *AC

Ví dụ minh họa cách sử dụng các lệnh số học với số nguyên trong LAD và STL



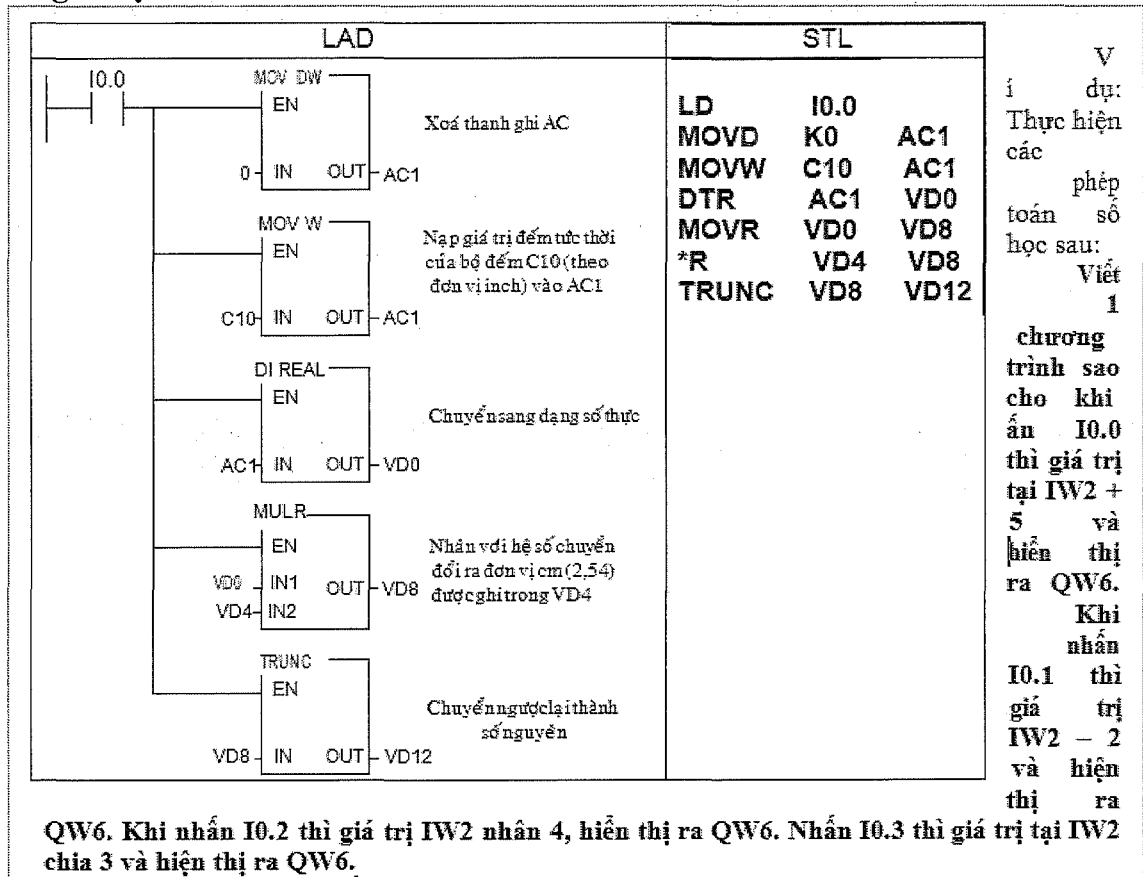
Với các giá trị đầu

AC0 = 6000 AC1 = 4000 VW102 = 200 VW202 = 4000 VW10 = 41

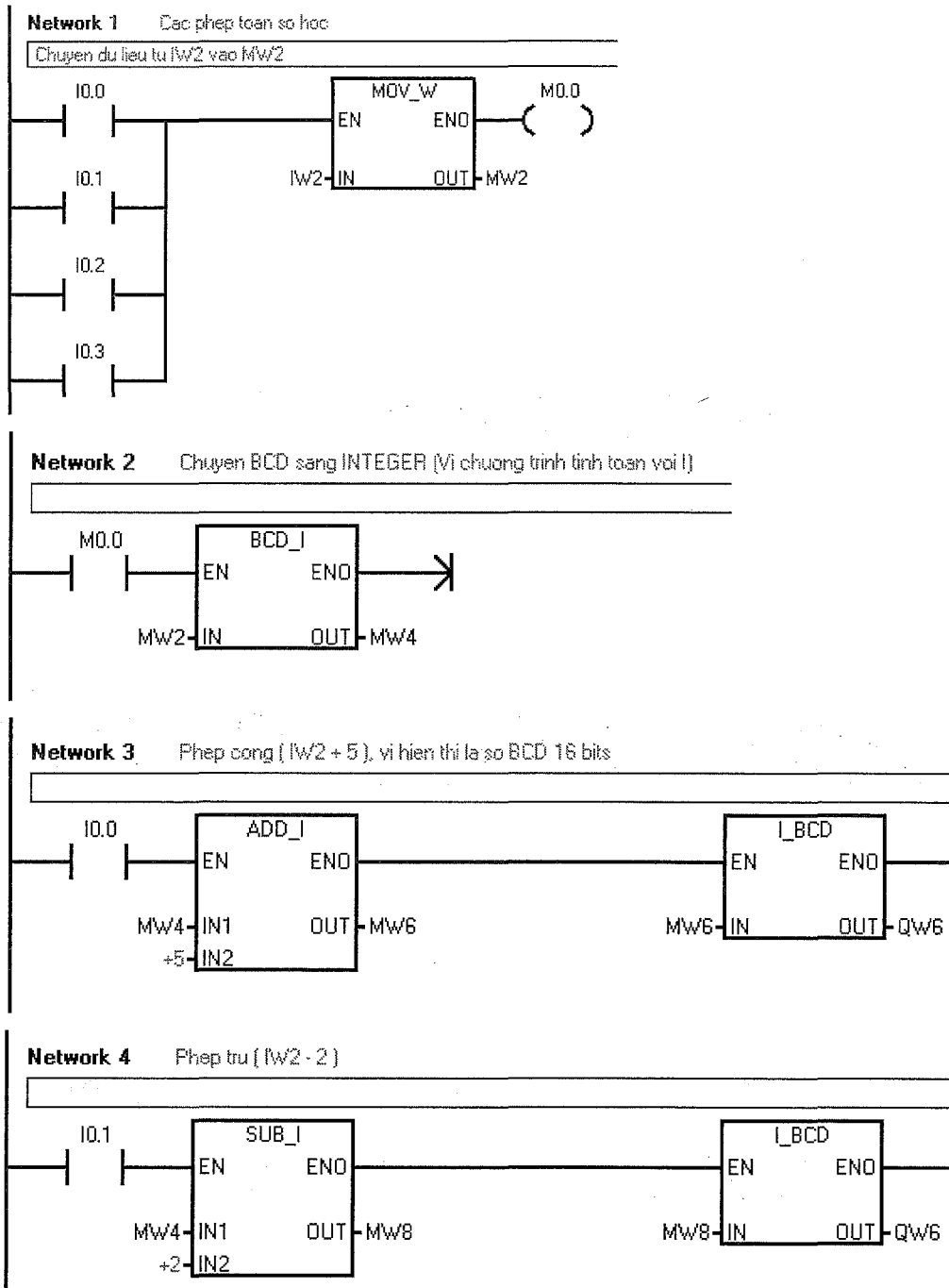
Thì kết quả của ví dụ trên sẽ là:

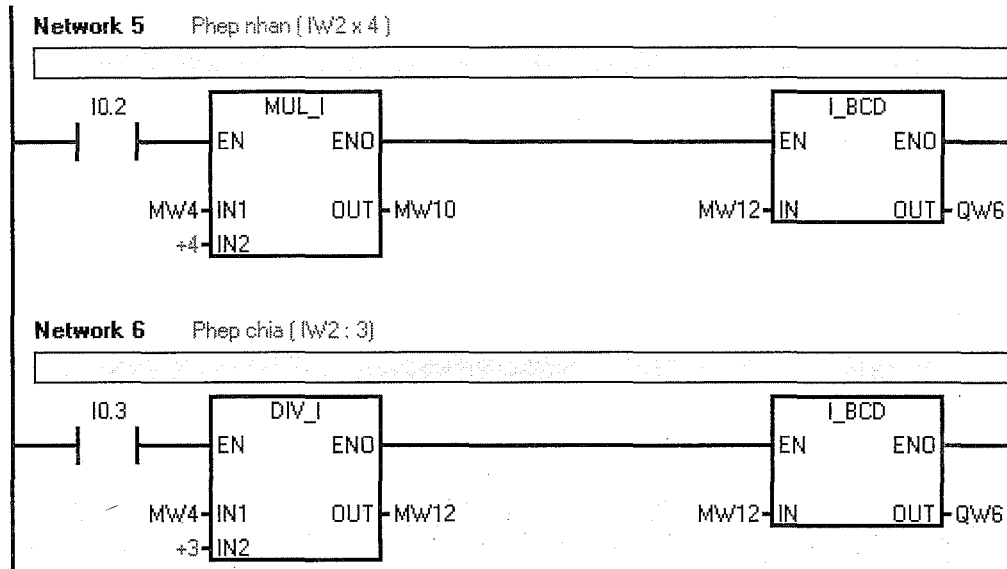
Phép cộng: AC0 = 10000
 Phép nhân: VD100 = 800000
 Phép chia: VD = 23 / 97
 Phần dư: 23, thương: 97

Ví dụ minh họa cách sử dụng các lệnh số học với số thực trong LAD và STL
 Nếu C-word của bộ đếm C10 có giá trị là 101 và từ kép VD4 có giá trị thương là 2,54 thì nội dung của các ô nhớ VD0, VD8 và VD12 sẽ lần lượt có giá trị như sau:



Chương trình được viết trong LAD như sau:





Chương trình được viết trong STL như sau:

<p>Network 1 Các phép toán số học</p> <p>Chuyen du lieu tu Iw2 vao MW2</p> <pre>LD I0.0 O I0.1 O I0.2 O I0.3 MOVW IW2, MW2 AENO = M0.0</pre> <p>Network 2 Chuyen BCD sang INTEGER (Vi chuong trinh tinh toan voi I)</p> <pre>LD M0.0 MOVW MW2, MW4 BCDI MW4</pre> <p>Network 3 Phep cong (Iw2 + 5), vi hien thi la so BCD 16 bits</p> <pre>LD I0.0 MOVW MW4, MW6 AENO +I +5, MW6 AENO MOVW MW6, QW6 IBCD QW6</pre> <p>Network 4 Phep tru (Iw2 - 2)</p> <pre>LD I0.1 MOVW MW4, MW8 AENO -I +2, MW8 AENO MOVW MW8, QW6 IBCD QW6</pre>	<p>Network 5 Phep nhan (Iw2 x 4)</p> <pre>LD I0.2 MOVW MW4, MW10 AENO *I +4, MW10 AENO MOVW MW12, QW6 IBCD QW6</pre> <p>Network 6 Phep chia (Iw2 : 3)</p> <pre>LD I0.3 MOVW MW4, MW12 AENO /I +3, MW12 AENO MOVW MW12, QW6 IBCD QW6</pre>
---	---

5.5. Đồng hồ thời gian thực

Đồng hồ thời gian thực chỉ có với CPU 214. Những giá trị đọc được hoặc ghi được với đồng hồ thời gian thực là các giá trị về ngày, tháng, năm và các giá trị về giờ, phút, giây. Các dữ liệu đọc, ghi với đồng hồ thời gian thực trong LAD và trong STL có độ dài 1 byte và phải được mã hóa theo theo kiểu số nhị thập phân thân BCD (thí dụ 16#95 cho năm 95). Chúng nằm trong bộ đệm gồm 8 byte liên nhau theo thứ tự:

Byte 0	Năm (0 ÷ 99)	
Byte 1	Tháng (0 ÷ 12)	
Byte 2	Ngày (0 ÷ 31)	
Byte 3	Giờ (0 ÷ 23)	
Byte 4	Phút (0 ÷ 59)	
Byte 5	Giây (0 ÷ 59)	
Byte 6	0	Ngày trong tuần
Byte 7		

Riêng giá trị về ngày trong tuần là một số tương ứng

Chủ nhật	Thứ hai	Thứ ba	Thứ tư	Thứ năm	Thứ sáu	Thứ bảy
1	2	3	4	5	6	7

Các lệnh đọc, ghi:

READ_RTC (LAD)

TODR (STL) Lệnh đọc nội dung của đồng hồ thời gian thực vào bộ đệm 8 byte được chỉ thị trong lệnh bằng toán hạng T.

SET_RTC (LAD)

TODW (STL) Lệnh ghi nội dung của bộ đệm 8 byte được chỉ thị trong lệnh bằng toán hạng T vào đồng hồ thời gian thực.

Bài tập 1: Báo giờ làm việc và tan tầm. Mô tả hoạt động;

Một xí nghiệp được trang bị một hệ thống tự động báo giờ làm việc và tan tầm như sau: - Giờ làm việc: Đúng 7:00 giờ sáng, thì một hồi còi vang lên kéo dài trong 1 phút báo hiệu giờ làm việc bắt đầu.

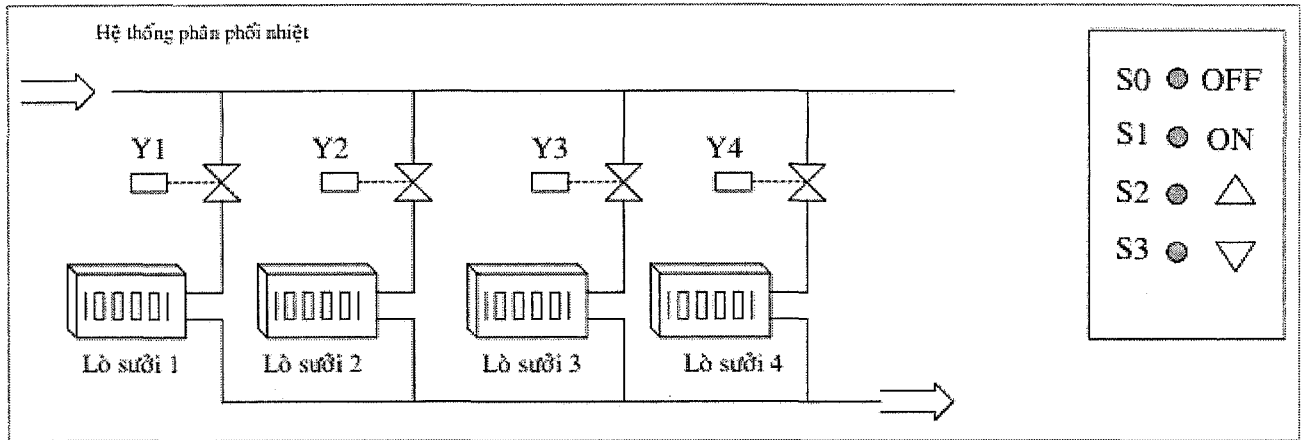
- Giờ tan tầm: Đúng 17:00 giờ, thì một hồi còi vang lên kéo dài trong 1 phút báo hiệu giờ làm việc kết thúc. Nhiệm vụ: - Lập bảng xác lập vào/ra.

- Vẽ sơ đồ kết nối với PLC.

- Viết chương trình

- Nạp chương trình vào PLC và thử chương trình.

Bài tập 2. Điều khiển hệ thống lò sưởi. Để điều khiển nhiệt độ trong phòng người ta dùng PLC điều khiển các van cấp nhiên liệu (hơi ẩm) cho các lò sưởi. Hệ thống gồm 4 lò sưởi, có công suất như sau: lò sưởi 1: 1 kw, lò sưởi 2: 2 kw, lò sưởi 3: 4 kw, lò sưởi 4: 8 kw



Hệ thống ở chế độ sẵn sàng hoạt động khi nhấn S1. Ở chế độ này các lò sưởi có thể chạy hoặc dừng bằng cách đóng hoặc mở các van Y1, Y2, Y3 và Y4 tùy thuộc vào việc cài đặt công suất của hệ thống bằng cách nhấn S2 và S3. Công suất của hệ thống lò sưởi từ 0 đến 15 kw. Một lần nhấn S2 thì công suất của hệ thống tăng 1 kw, 1 lần nhấn S3 thì công suất của lò giảm đi 1 kw. Hệ thống ở chế độ dừng khi nhấn nút S0.

Nhiệm vụ:

- Lập bảng xác lập vào/ra.
- Vẽ sơ đồ kết nối với PLC.
- Viết chương trình.

Hệ thống ở chế độ sẵn sàng hoạt động khi nhấn S1. Ở chế độ này các lò sưởi có thể chạy hoặc dừng bằng cách đóng hoặc mở các van Y1, Y2, Y3 và Y4 tùy thuộc vào việc cài đặt công suất của hệ thống bằng cách nhấn S2 và S3. Công suất của hệ thống lò sưởi từ 0 đến 15 kw. Một lần nhấn S2 thì công suất của hệ thống tăng 1 kw, 1 lần nhấn S3 thì công suất của lò giảm đi 1 kw. Hệ thống ở chế độ dừng khi nhấn nút S0.

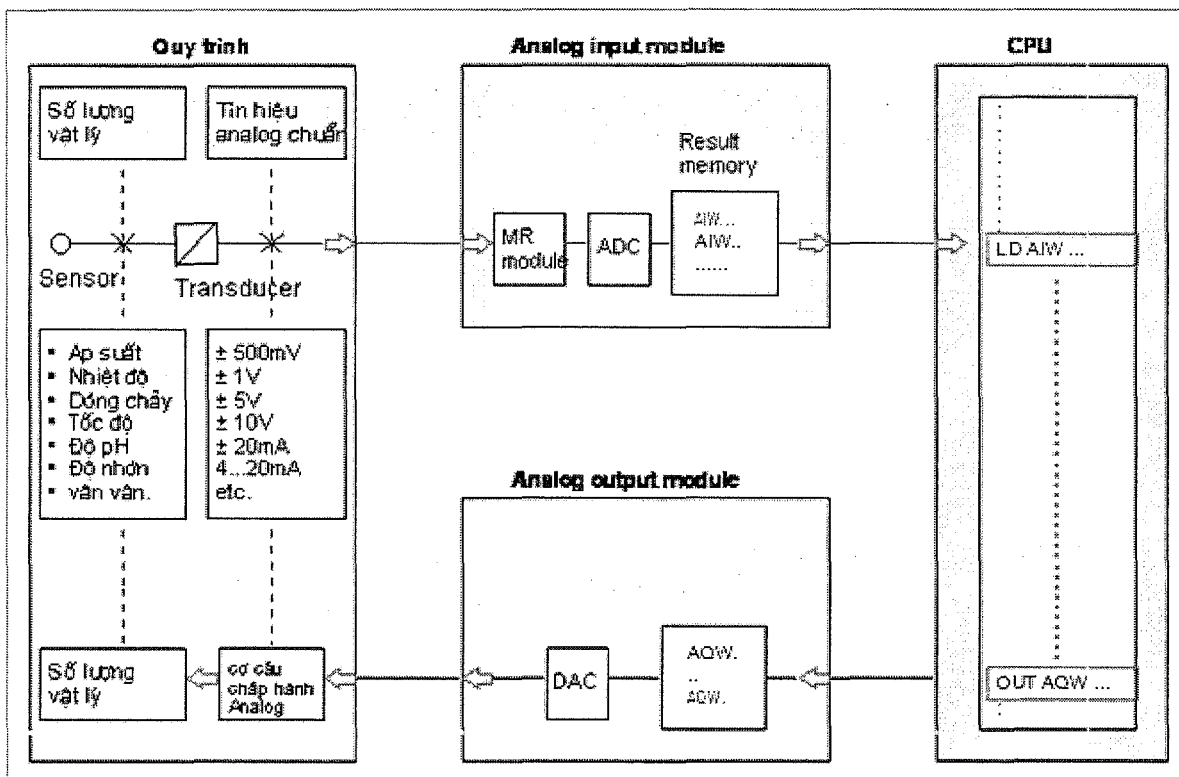
Nhiệm vụ:

- Lập bảng xác lập vào/ra.
- Vẽ sơ đồ kết nối với PLC.
- Viết chương trình.
- Nạp chương trình vào PLC và thử chương trình.

Bài 6: XỬ LÝ TÍN HIỆU ANALOG

6.1. 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.



6.2. 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	0
	9	64	40	*	*	*	*	*	*	*	*	*	1	0	0	0	0	0	0
	10	32	20	*	*	*	*	*	*	*	*	*	*	1	0	0	0	0	0
	11	16	10	*	*	*	*	*	*	*	*	*	*	*	1	0	0	0	0
	12	8	8	*	*	*	*	*	*	*	*	*	*	*	*	1	0	0	0
	13	4	4	*	*	*	*	*	*	*	*	*	*	*	*	*	1	0	0
	14	2	2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	0
	15	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1

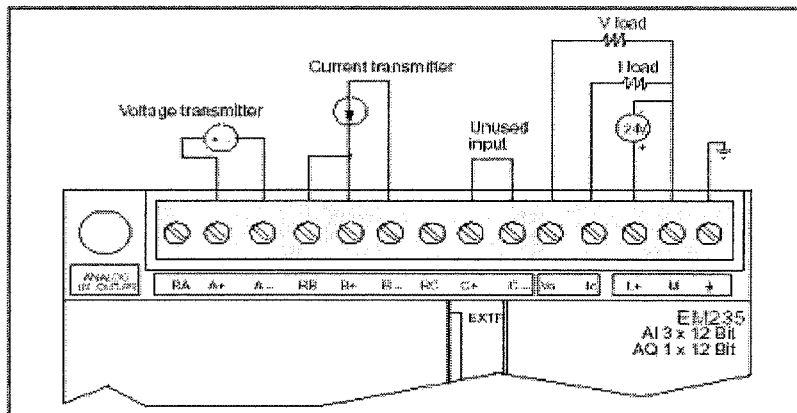
* = 0 or 1

6.3. 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 mô đun (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



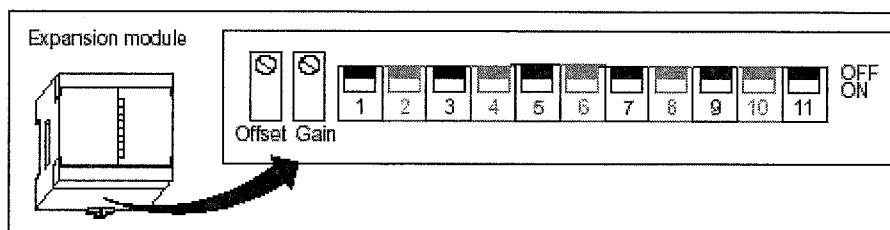
Phương pháp định tỷ lệ ngõ vào analog (Input calibration)

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.

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

- + Tắt nguồn cung cấp cho mô đun, chọn phạm vi ngõ vào mong muốn
- + *Cấp nguồn lại cho CPU và mô đun 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	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.

6.4. Hiệu chỉnh tín hiệu Analog

Trên CPU S7-200 có 2 biến trở (2 biến trở này nằm dưới nắp của mô đun), 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.

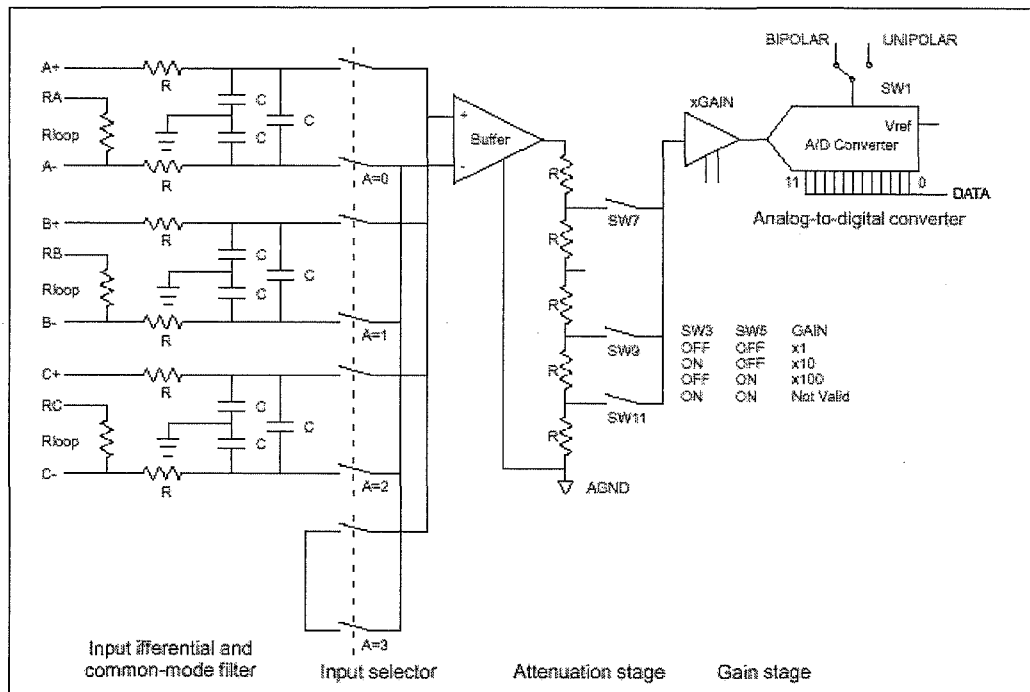
Byte nhớ SMB 28 lưu trữ giá trị số biểu diễn vị trí chỉnh 0. SMB 29 lưu trữ giá trị số biểu diễn vị trí chỉnh 1. Sự điều chỉnh Analog có giới hạn từ 0 tới 255 và độ tin cậy tốt nhất trong phạm vi từ 10 đến 200.

Để thực hiện điều chỉnh này, phải sử dụng một Tuộc 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:

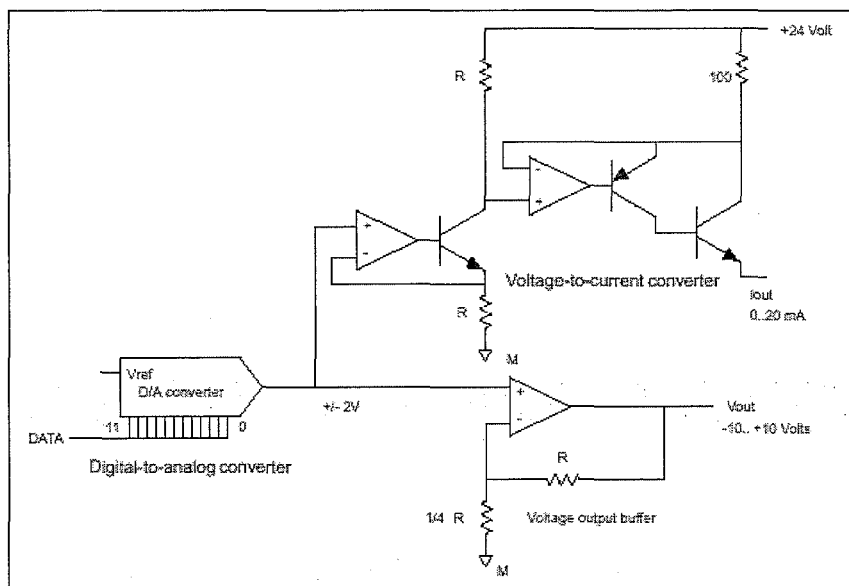
Timer T33 đóng tiếp điểm khi VW 100 đạt giá trị đặt trước

Configuration Switch						Full-Scale Input	Resolution
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

Sau đây là sơ đồ nguyên lý mạch của modul EM 235 3AI/
IAO Sơ đồ mạch ngõ vào :



Sơ đồ mạch

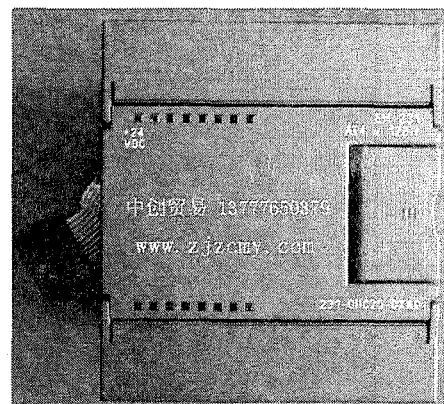


PLC S7 200 có các module analog mở

- EM 231: Gồm có bốn ngõ vào analog.
- EM 232: Gồm có hai ngõ vào analog.
- EM 235: Gồm có bốn ngõ vào analog và 1 ngõ ra analog.

1/ Đặc tính chung

- Trở kháng vào $\geq 10M$
- Bộ lọc đầu vào -3db tại 3.1 KHz.
- Điện áp cực đại cung cấp cho mô đun: 30VDC
- Dòng điện cực đại cung cấp cho mô đun: 32mA.
- Có led báo trạng thái.
- Có nút chỉnh OFFSET và chỉnh độ lợi.

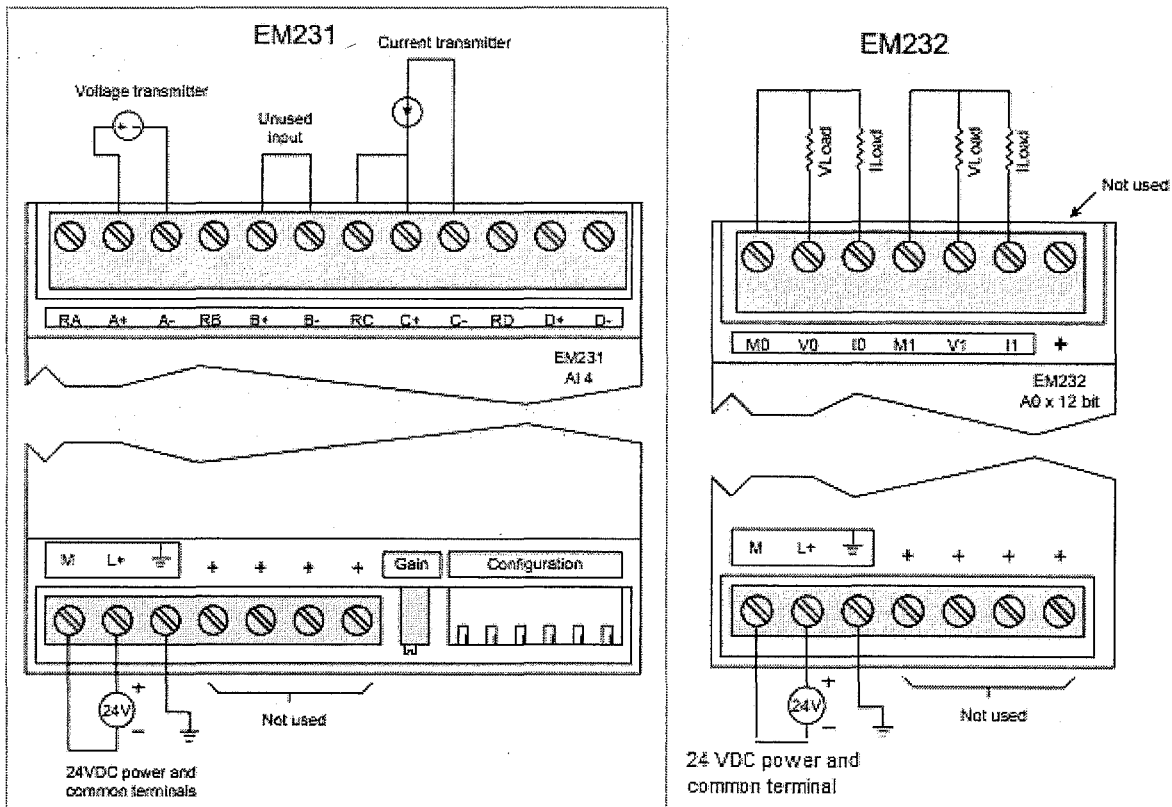


/ Đặc tính kỹ thuật của mô đun Analog EM 231

a/ Đầu vào:

- Số đầu vào: 4, độc lập nhau.
- Chức năng bảo vệ tính: 0
- Phạm vi đầu vào: 0 ~ 50mV // 0 ~ 100mV // 0 ~ 500mV // 0 ~ 1V // 0 ~ 10V // 0 ~ 20mA // 25mA // 50mA // 100mA // 250mA // 500mA // 1V // 2,5V // 5V // 10V.
- Điện áp đầu vào cho phép với đầu vào điện áp, tối đa là 30V.

- Dòng điện đầu vào cho phép với đầu vào dòng điện tối đa là 32mA
- Cách ly điện: không.
- Bộ chuyển đổi: 12 bit.
- Thời gian chuyển đổi từ tương tự sang số : 250 μ s.
- Độ phân giải 12 bit
- Điện áp chế độ chân dung tối đa :
- Triệt nhiễu: 40dB, DC đến 60 Hz (không triệt nhiễu tần số)
- Phạm vi có thể thay thế của các giá trị chuyển



b/ Đầu ra

Số đầu ra: 1

Phạm vi đầu ra:

- Đầu ra điện áp: -10V ~ +10V
- Đầu ra dòng điện 0 ~ 20mA.

Điện trở tải:

- Với đầu ra điện áp nhỏ nhất là: 5k Ω
- Với đầu ra dòng điện lớn nhất là: 0,5k Ω

Độ phân giải:

- Với đầu điện áp nhỏ nhất là: 12 bit

- Với đầu ra dòng điện lớn nhất là: 11 bit

Thời gian đặt:

- Với đầu ra điện áp là $100\mu s$
- Với đầu ra là dòng điện 2ms.

Phạm vi có thể hiển thị được của giá trị chuyển đổi:

- Các tín hiệu đơn cực tính: $-32.000 \sim +32.000$.

Giới hạn lỗi hoạt động ở $60^{\circ}C$

- Điện áp: 2%
- Dòng điện: 2%

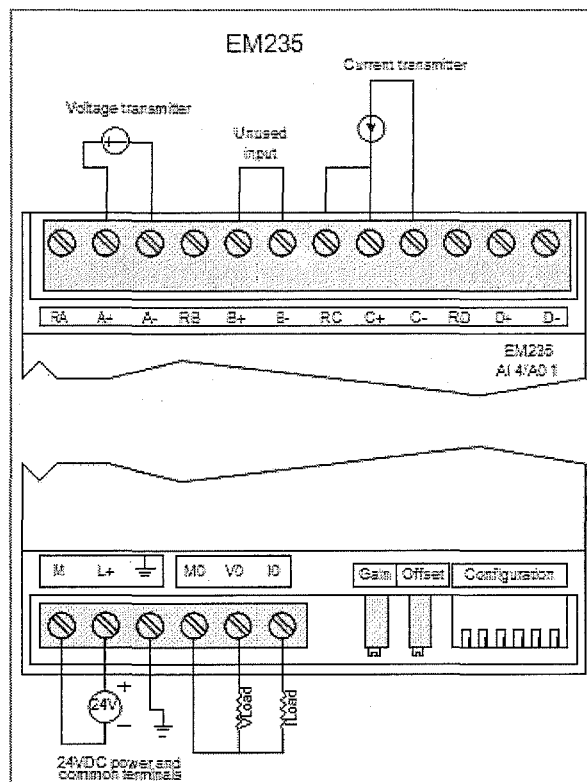
Giới hạn lỗi hoạt động ở $25^{\circ}C$:

- Điện áp: 0,5%
- Dòng điện: 0,5% Tiêu thụ dòng điện:
- Từ bus backplane (VDC): 30mA.
- Từ nguồn cấp sensor hoặc nguồn cấp ngoài: 60W.

Tổn thất công suất: 2W

Kích thước (W*H*D) mm: 71,2*80*62 Trọng lượng: 186g.

Sơ đồ kết nối các thiết bị ngoại vi, sử dụng theo dạng áp và dòng. Các contact (Switch) để lựa chọn phạm vi ngõ vào (contact ở một trong hai vị trí ON và OFF). Contact 1 lựa chọn cực tính áp ngõ vào: ON đối với áp đơn cực, OFF đối với áp lưỡng cực; contact 2, 3, 4, 5, 6, chọn phạm vi điện áp.



c/ Các chú ý khi cài đặt ngõ ra analog:

Chắc chắn là nguồn 24VDC cung cấp không bị nhiễu và ổn định.

Xác định được mô đun.

Dùng dây cảm biến ngắn nhất nếu có t.

Sử dụng dây bọc giáp cho cảm biến và dây chỉ dùng cho một mình cảm biến.

Tránh đặt các dây tín hiệu song song với các dây có năng lượng cao. Nếu hai dây bắt buộc phải gặp nhau thì bắt chéo chúng về phía bên phải.

Kiểm tra các ngõ vào analog.

Với chương trình bất kỳ đang chạy trong PLC thì ta có thể thấy giá trị sau khi biến đổi A/D ở các kênh analog thông qua chức năng Debug > Chart Status với Chart Status có liệt kê AIW0 đến AIW6. Ta chỉnh các biến trở bên ngoài và khảo sát những thay đổi ở các word ATW0 đến AIW6.

Lập lại tín hiệu vào:

bộ thí nghiệm S7 200 đang đặt cấu hình nhập và xuất cùng tỉ lệ, có nghĩa là nếu đọc vào x V và xuất lại ngõ ra thì cũng được x V. (giá trị ngõ vào cho phép là từ 0 đến 10V)

Hãy viết chương trình sử dụng các ngõ vào I0.0 đến I0.3 để chọn ngõ ra lập lại giá trị analog của kênh vào nào (I0.0 ứng với kênh 0, I0.1 ứng với kênh 1, I0.2 ứng với kênh 2 và I0.3 ứng với kênh 3; Nghĩa là tương ứng với các kênh A, B, C và D ở PLC).

Thí dụ: Nhập đoạn chương trình sau thực hiện xuất giá trị analog ra lập lại ở kênh 0.

```
LD I0.0
```

```
MOVW AIW0, AQW0
```

Nếu các ngõ vào I0.0 đến I0.3 sử dụng loại công tắc NO, hãy viết chương trình cho trường hợp này.

Một số xử lý đơn giản trên analog

Gọi giá trị analog là Y volt và analog vào kênh A là XA volt, kênh B là XB volt, kênh C là XC volt, kênh D là XD volt; gọi M (thí dụ sử dụng VW0) là hằng số cần thực hiện với dữ liệu analog và chú ý M là số nguyên.

Thực hiện $Y = M * XA$

Ta chỉ cần viết 1 đoạn chương trình như sau:


```

MOVW    AIW0, MW0    // lấy số liệu XA
*I      M, MW0
MOVW    MW0, AQW0    // xuất ra Y = XA*M

```

Thực hiện $Y = XA / M$

Ta chỉ cần viết 1 đoạn chương trình như sau:

```

MOVW    AIW0, MW0    // lấy số liệu XA
/I      M, MW0
MOVW    MW0, AQW0    // xuất ra Y = XA / M

```

Chú ý đây là phép chia nguyên nên trị số sẽ không chính xác

Thực hiện $Y = (XA + XB + XC + XD) / 4$

Ta chỉ cần viết 1 đoạn chương trình sau:

```

MOVW    AIW0, MW0    // MW0 = XA
+I      AIW2, MW0    // MW0 = XA + XB
+I      AIW4, MW0    // MW0 = XA + XB + XC
+I      AIW6, MW0    // MW0 = XA + XB + XC + XD
/I      4, MW0       // MW0 = (XA + XB + XC + XD) /
MOVW    MW0, AQW0    // xuất ra Y = (XA + XB + XC +

```

Chú ý là việc thực hiện chỉ đúng khi các trị số tính toán không bị tràn.

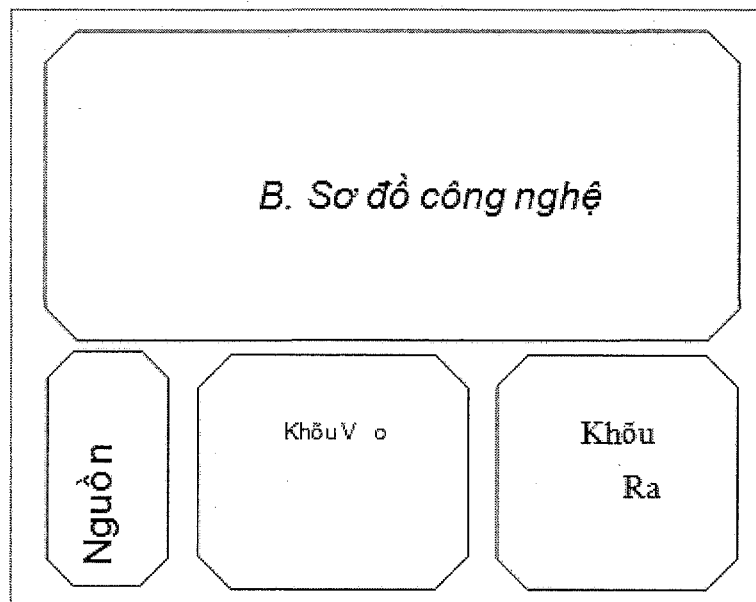
Bài 7. CÁC BÀI TẬP ỨNG DỤNG TRONG ĐIỀU KHIỂN ĐỘNG CƠ

7.1. Giới thiệu

Việc nâng cao chất lượng trong giảng dạy trong kỹ thuật luôn luôn gắn liền với việc học đi đôi với hành. Hiện nay, trong thực hành của học sinh, thì các đồ dùng dạy học với dáng vẻ trang nhã và gọn gàng, bảo đảm an toàn và giúp cho học sinh có một cái nhìn khái quát về những ứng dụng trong thực tế có một ý nghĩa quan trọng. Nó không những giúp cho học sinh có hứng thú trong học tập mà còn giúp cho học sinh có thêm những sáng kiến mới, cũng như cách thức tổ chức trong thực tế.

Từ lý do đó, chúng tôi đã đưa ra một số mô hình đáp ứng được những yêu cầu trên. Đây là các mô hình dùng cho môn học PLC từ cơ bản đến nâng cao lấy từ các ứng dụng trong thực tế được mô phỏng qui trình công nghệ bằng các đèn LED. Mỗi mô hình có thể có nhiều bài tập ứng dụng khác nhau.

Mỗi mô hình có kích thước A4 (210x297 mm) rất phù hợp trong việc lắp ghép trên các bảng tạo thành một mô đun hoàn chỉnh. Cấu trúc chung bề mặt của mô hình được phân bố như sau:



Mục đích của việc phân thành từng cụm riêng là giúp cho học sinh tránh được những nhầm lẫn đáng tiếc trong quá trình thực hành đồng thời cũng giúp cho học sinh tiếp thu thêm được về cách thức tổ chức trong thực hành.

Các mô hình thực tập gồm có:

1. Mô hình thang máy xây dựng
2. **Mô hình điều khiển động cơ sao - tam giác**

3. Mô hình xe chuyển nguyên liệu
4. Mô hình đo chiều dài và sắp xếp vật liệu
5. Mô hình thiết bị nâng hàng hóa
6. Mô hình thiết bị vô nước chai
7. Mô hình thiết bị trộn hóa chất

Các mô hình này đã được sắp xếp theo thứ tự và có các bài tập kèm theo.

Toàn bộ các mô hình sử dụng điện áp 24VDC. Điện áp này có thể được lấy từ nguồn riêng hoặc nguồn 24VDC có sẵn cung cấp cho PLC. Đối với các PLC có ngõ ra là relay thì trên mô hình có thiết kế sẵn nguồn US dùng làm nguồn cung cấp cho các ngõ ra này.

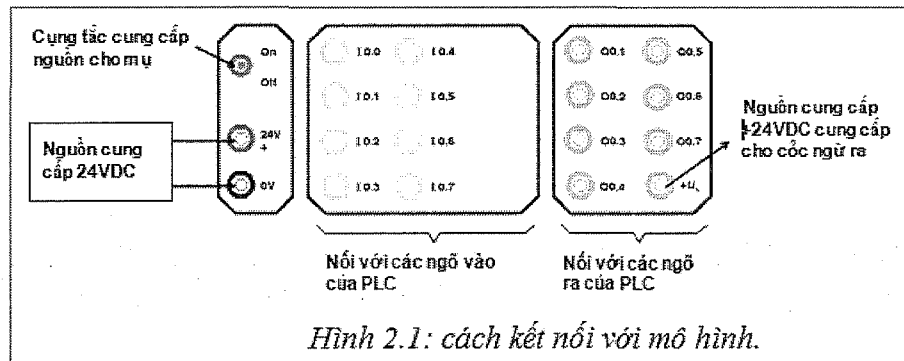
Các mô hình cũng có thể được ứng dụng cho các bộ điều khiển lập trình loại nhỏ như LOGO! của hãng Siemens, EASY của hãng Moeller, ZEN của Omron....

Tùy theo nội dung bài học mà có thể chọn mô hình thích hợp cho bài tập ứng dụng.

Một mô hình có thể sử dụng với nhiều bài tập ứng dụng khác nhau. Lấy ví dụ: *mô hình thang máy xây dựng có thể được sử dụng trong các bài học như ứng dụng điều khiển theo tổ hợp logic, điều khiển với các lệnh ghi/xóa tiếp điểm, sử dụng timer, sử dụng counter, và ứng dụng trong điều khiển trình tự.*

7.2. Cách kết nối dây

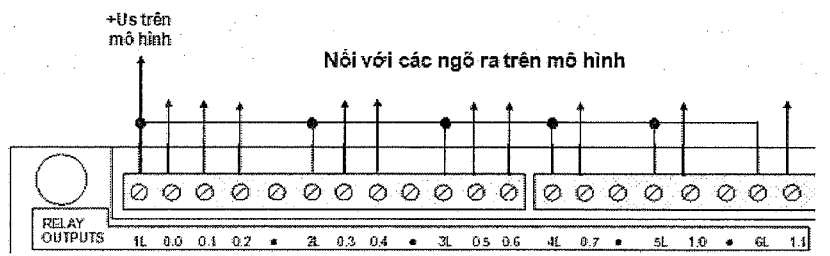
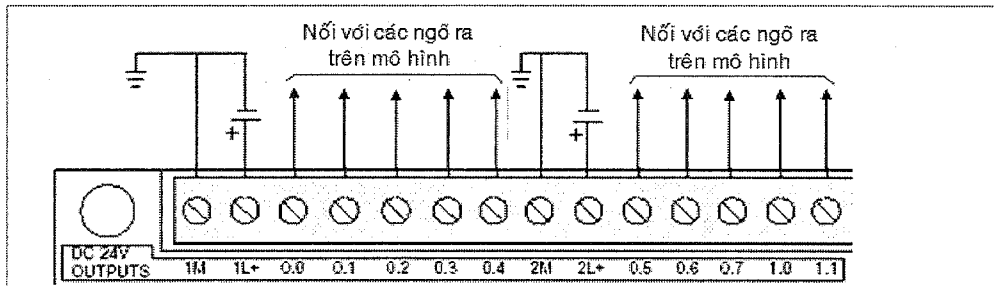
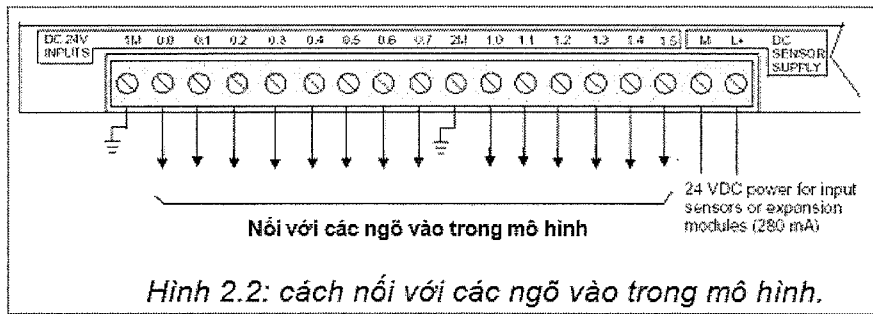
Cách nối dây từ PLC đến mô hình được cho như hình vẽ:



Để kết nối được với PLC, yêu cầu các mô đun vào/ra của PLC như sau:

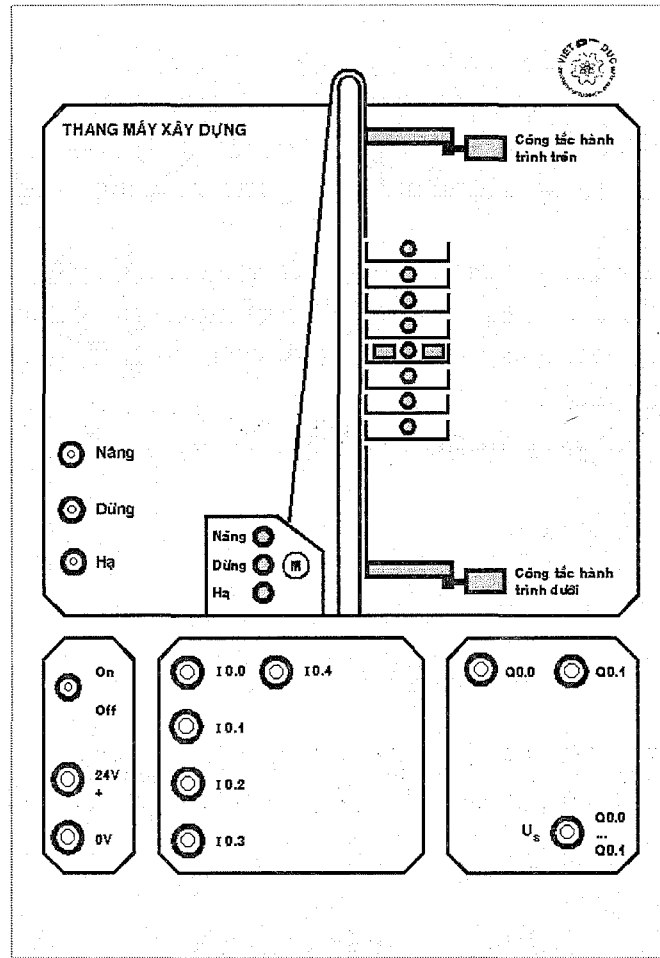
- Sử dụng nguồn áp 24VDC (ổn áp).
- Nguồn cung cấp cho mô đun vào/ra phải được kết nối
- Nếu các ngõ ra là role và chưa có nguồn cung cấp thì đấu chung một đầu lại rồi nối với nguồn +Us ở trên mô hình (hoặc nguồn +24VDC ngoài), còn các đầu còn lại của role nối với ngõ ra trên mô hình.

Các kết nối có thể thực hiện như ví dụ sau:



7.3. Tóm tắt các mô hình và bài tập ứng dụng.

7.3.1. Mô hình thang máy xây dựng



a. Mô Tả:

Mô tả qui trình công nghệ của một thang máy xây dựng. Sự chuyển động của thang được biểu diễn dưới dạng LED. Tín hiệu các công tắc giới hạn được tạo ra tự động.

Mô hình này được ứng dụng trong phần bài tập cơ bản trong môn học PLC (ứng dụng các cổng logic, timer, counter).

Ngoài ra cũng có thể được áp dụng cho phần lập trình nâng cao (điều khiển trình tự).

b. Bảng ký hiệu

<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
Nâng	I0.0	Nút nhấn nâng, thường hở
Hạ	I0.1	Nút nhấn hạ, thường hở
Dừng	I0.2	Nút nhấn dừng, thường đóng
GH_Trên	I0.3	Công tắc hành trình trên, thường đóng
GH_Dưới	I0.4	Công tắc hành trình dưới, thường đóng
K1	Q0.0	Cuộn dây khởi động từ K1, nâng gàu
K2	Q0.1	Cuộn dây khởi động từ K2, hạ gàu

c. Bài tập mẫu

Các bài tập mẫu này được giải với theo phần mềm STEP 7 Micro/win 32 V3.01.

1. Bài tập 1: Ứng dụng cổng logic, các lệnh ghi/xóa tiếp điểm

Viết chương trình điều khiển thang máy xây dựng theo yêu cầu sau::

- Khi nhấn nút nhấn nâng, gàu sẽ được nâng lên đến công tắc giới hạn trên thì dừng lại.
- Khi nhấn nút nhấn hạ, gàu sẽ hạ xuống đến công tắc giới hạn dưới thì dừng lại.
- Khi đang nâng hoặc hạ, nếu nhấn nút nhấn dừng thì gàu dừng lại.

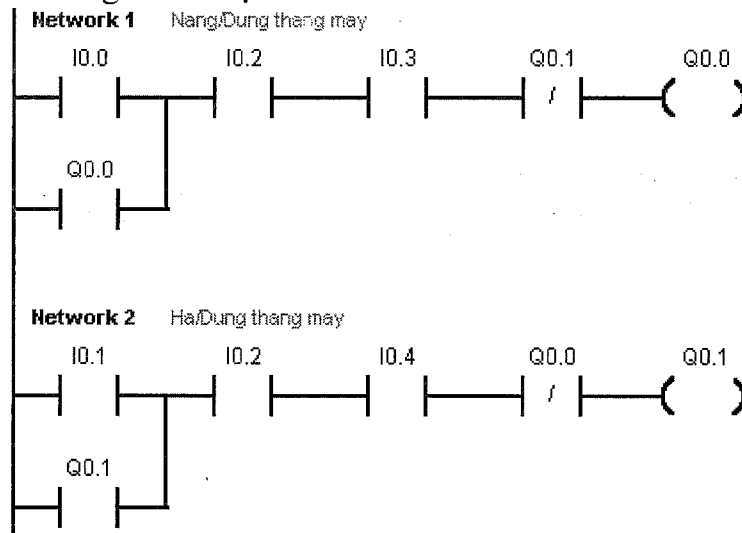
Hãy viết chương trình theo mô tả theo hai cách: ứng dụng cổng logic và sử dụng các lệnh ghi xóa tiếp điểm.

Kiểm tra hoạt động bằng mô hình.

Bài giải: -

Cách 1: ứng dụng cổng logic

Chương trình được viết ở LAD:



Chương trình được viết ở STL:

```

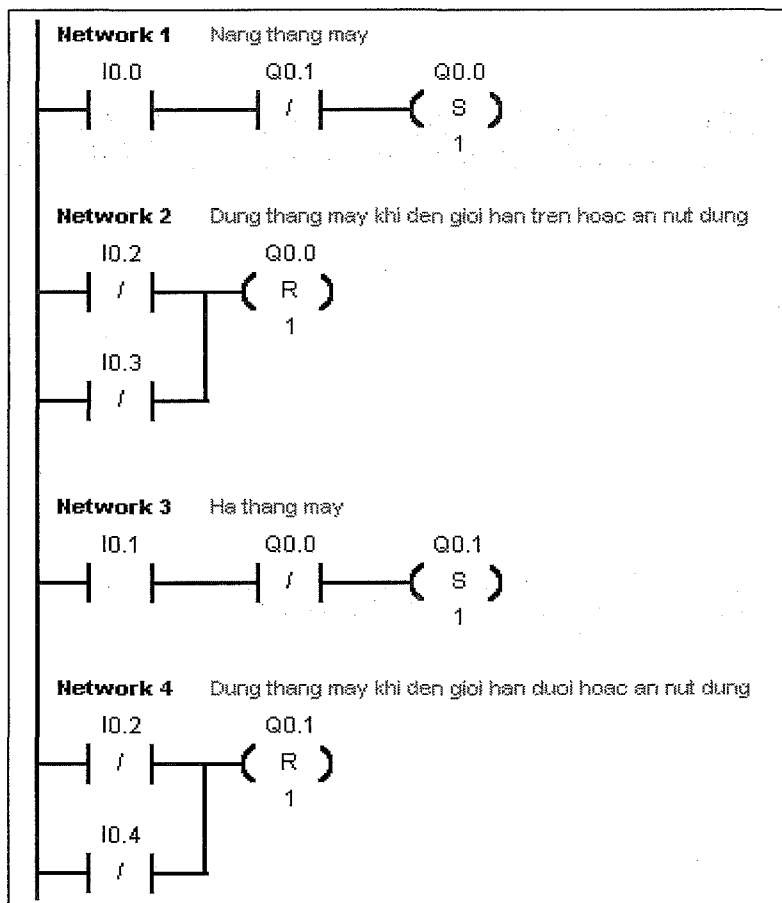
NETWORK 1      //Nang/Dung thang may
//
LD      IO.0
O       Q0.0
A       IO.2
A       IO.3
AN      Q0.1
=       Q0.0

NETWORK 2      //Ha/Dung thang may
//
LD      IO.1
O       Q0.1
A       IO.2
A       IO.4
AN      Q0.0
=       Q0.1

```

Cách 2: Sử dụng lệnh Set/reset:

Chương trình viết ở LAD:



Chương trình viết ở STL:

```

NETWORK 1      //Nâng thang máy
//
LD      I0.0
AN      Q0.1
S      Q0.0, 1

NETWORK 2      //Dừng thang máy khi đến giới hạn trên hoặc an nút dừng
//
LDN     I0.2
ON      I0.3
R      Q0.0, 1

NETWORK 3      //Hạ thang máy
//
LD      I0.1
AN      Q0.0
S      Q0.1, 1

NETWORK 4      //Dừng thang máy khi đến giới hạn dưới hoặc an nút dừng
//
LDN     I0.2
ON      I0.4
R      Q0.1, 1

```

Bài tập 2: Sử dụng Timer.

Viết chương trình điều khiển thang máy xây dựng theo yêu cầu sau:

a/. Khi ấn nút nâng thì gàu được nâng lên, đến giới hạn trên thì dừng lại 5s, sau đó tự động hạ xuống. Đến giới hạn dưới thì dừng.

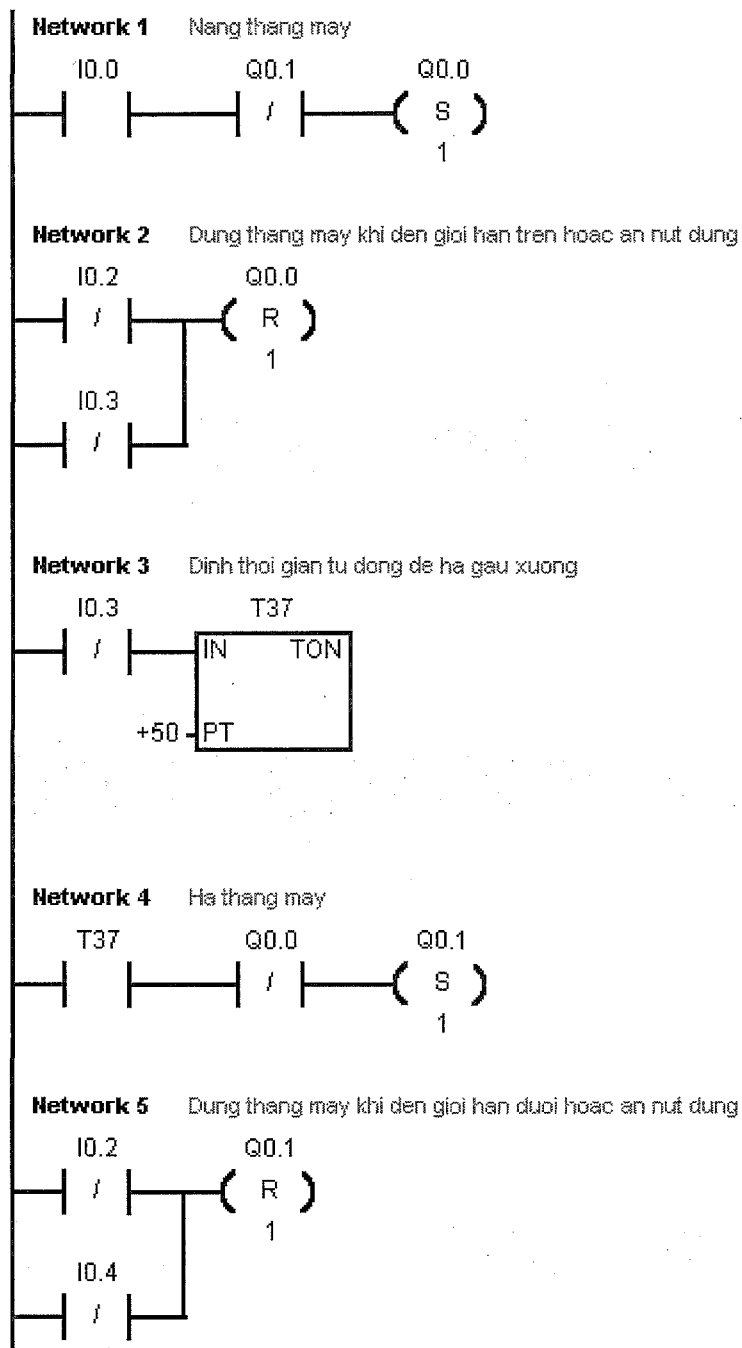
Trong quá trình nâng lên hoặc hạ xuống cũng có thể dừng.

b/. Khi ấn nút nâng thì gàu được nâng lên, đến giới hạn trên thì dừng lại 5s, sau đó tự động hạ xuống đến giới hạn dưới thì dừng lại 10s, sau đó tự động nâng lên.

Thang cũng có thể nâng lên khi chưa hết 10s chờ tự động mà có người ấn nút nâng.

Giải:

Câu a: Chương trình viết ở LAD:



Chương trình viết ở STL:

```

NETWORK 1      //Nang thang may
//
LD      IO.0
AN      Q0.1
S       Q0.0, 1

NETWORK 2      //Dung thang may khi den gioi han tren hoac an nut dung
//
LDN     IO.2
ON      IO.3
R       Q0.0, 1

NETWORK 3      //Dinh thoi gian tu dong da ha gau xuong
//
LDN     IO.3
TON     T37, +50

NETWORK 4      //Ha thang may
//
LD      T37
AN      Q0.0
S       Q0.1, 1

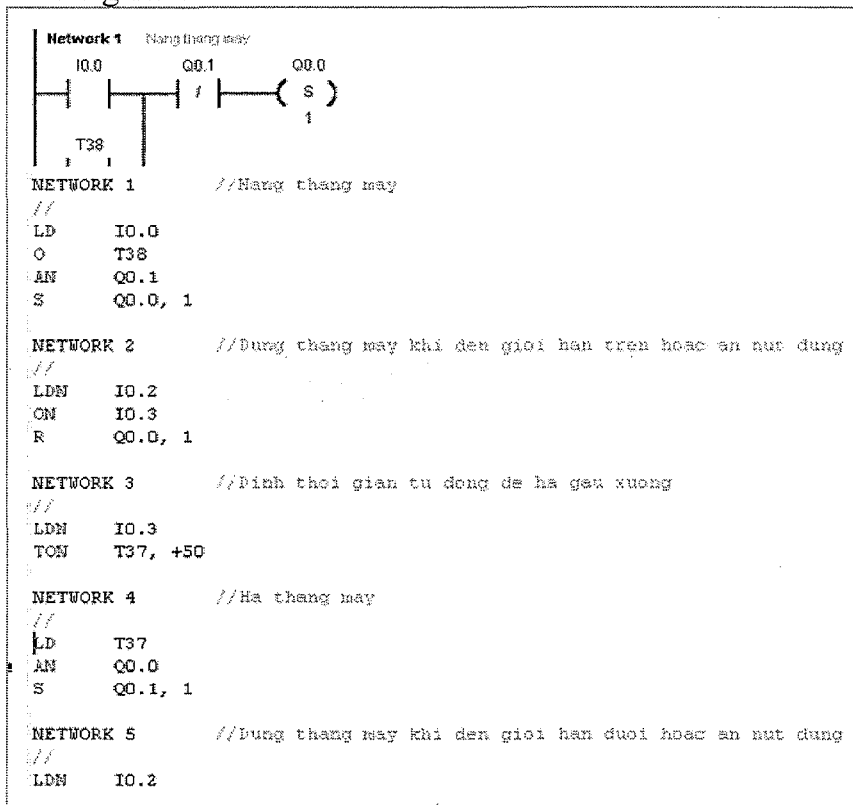
NETWORK 5      //Dung thang may khi den gioi han duoi hoac an nut dung
//
LDN     IO.2
ON      IO.4
R       Q0.1, 1

```

Ghi chú: Nếu bài toán có cho yêu cầu khi ấn nút hạ thì gàu cũng hạ, lúc này chèn thêm một tiếp điểm “NO“ của I0.1 song song với tiếp điểm “NO“ T37 ở Network 4.

Câu b: Chương trình viết ở LAD:

Chương trình viết ở STL:



Ghi chú: Nếu bài toán có cho yêu cầu khi ấn nút hạ thì gàu cũng hạ, lúc này chèn thêm một tiếp điểm “NO” của I0.1 song song với tiếp điểm “NO” T37 ở Network 4.

Bài tập 3: Sử dụng bộ đếm:

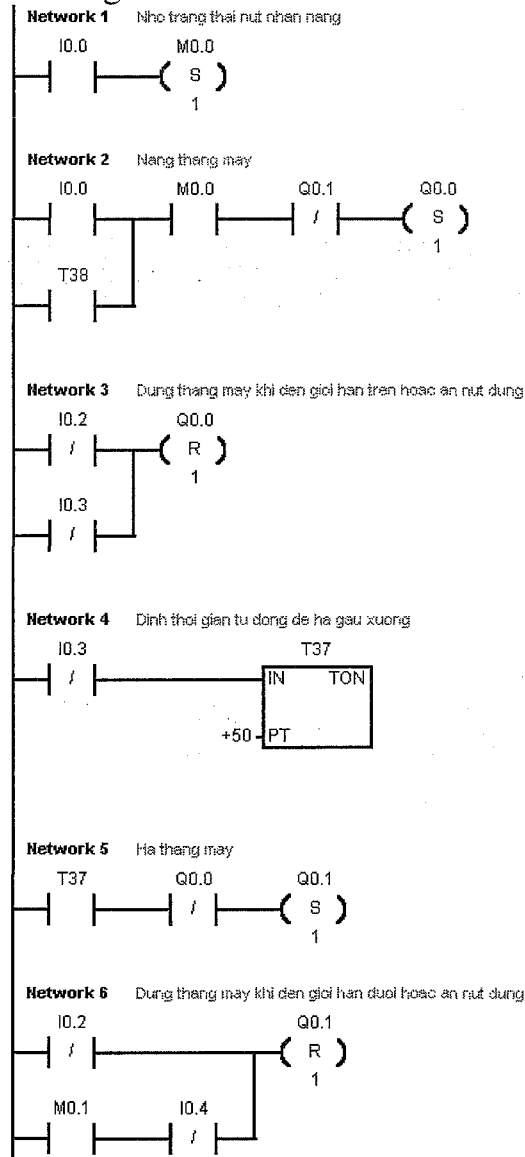
Viết chương trình điều khiển thang máy xây dựng theo yêu cầu sau:

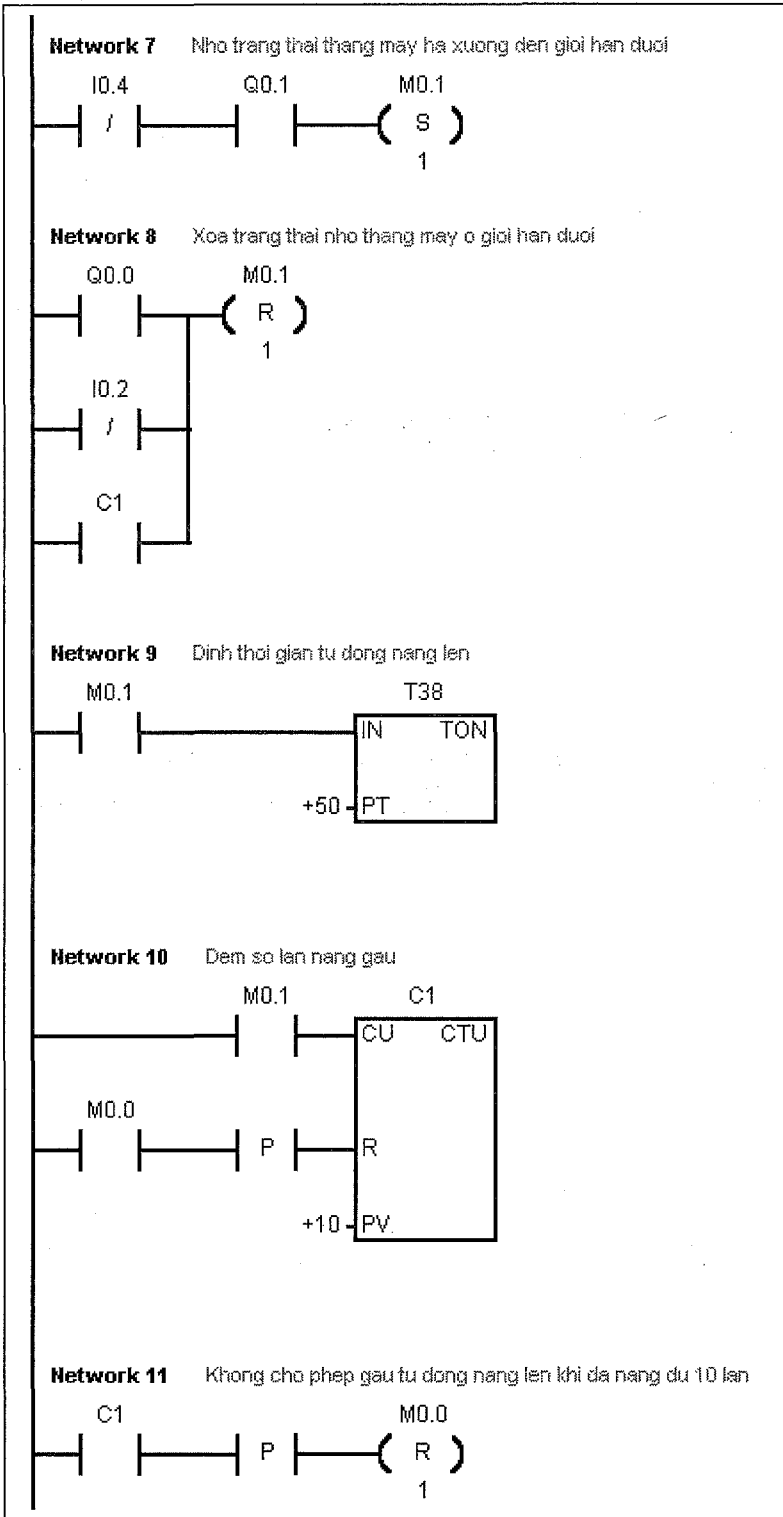
Khi ấn nút nâng thì gàu được nâng lên, đến giới hạn trên thì dừng lại 5s, sau đó tự động hạ xuống đến giới hạn dưới thì dừng lại 10s, sau đó tự động nâng lên. Khi gàu nâng lên được 10 lần thì không nâng lên nữa và sau đó hạ xuống trở về vị trí cơ bản và quá trình lặp lại.

Trong quá trình đang nâng hoặc hạ cũng có thể dừng gàu.

Giải:

Chương trình viết ở LAD:





Ghi chú: Nếu bài toán có cho yêu cầu khi ấn nút hạ thì gàu cũng hạ, lúc này chèn thêm một tiếp điểm “NO“ của I0.1 song song với tiếp điểm “NO“ T37 ở Network 5.

Chương trình viết ở STL:

```

NETWORK 1      //Nho trang thai nut nhan nang
//
LD      IO.0
S      MO.0, 1

NETWORK 2      //Mang thang may
//
LD      IO.0
O      T38
A      MO.0
AN     Q0.1
S      Q0.0, 1

NETWORK 3      //Dung thang may khi den gioi han tren hoac an nut dung
//
LDN     IO.2
ON      IO.3
R      Q0.0, 1

NETWORK 4      //Dinh thoi gian tu dong de ha gau xuong
//
LDN     IO.3
TON     T37, +50

NETWORK 5      //Ha thang may
//
LD      T37
AN      Q0.0
S      Q0.1, 1

NETWORK 6      //Dung thang may khi den gioi han duoi hoac an nut dung
//
LDN     IO.2
LD      MO.1
AN      IO.4
OLD
R      Q0.1, 1

NETWORK 7      //Nho trang thai thang may ha xuong den gioi han duoi
//
LDN     IO.4
A      Q0.1
S      MO.1, 1

NETWORK 8      //Xoa trang thai nho thang may o gioi han duoi
//
LD      Q0.0
ON      IO.2
O      C1
R      MO.1, 1

NETWORK 9      //Dinh thoi gian tu dong nang len
//
LD      MO.1
TON     T38, +50

NETWORK 10     //Dem so lan nang gau
//
LD      MO.1
LD      MO.0
EU
CTU     C1, +10

NETWORK 11     //Khong cho phép gau tu dong nang len khi da nang du 10 lan
//
LD      C1
EU
R      MO.0, 1

```

a. Mô tả

Mô hình này mô phỏng một động cơ mở máy Y/N. Sự chuyển động của rotor và sự đóng cắt của các contactor được mô tả bằng LED.

Mô hình này được ứng dụng trong phần bài tập cơ bản của môn học PLC (ứng dụng PLC trong điều khiển động cơ, các cổng logic, timer).

Có thể dựa vào mô hình băng tải để lập thành các bài tập khác nhau như: Mở máy động cơ đảo chiều, đảo chiều quay động cơ, điều khiển động cơ ở các tốc độ khác nhau

Cách thức nối dây cho mô hình như ở mục II.

Cách vận hành mô hình:

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào ra bằng phần mềm (đối với S7-200 dùng bảng Status chart)) và sau đó thực hiện mô phỏng với mô hình.

Khi động cơ quay phải (trái), các đèn LED sẽ chuyển động theo chiều phải (trái) theo các trạng thái tương ứng: Nếu đèn chuyển động nhanh báo động cơ quay tốc độ nhanh và ngược lại. Các contactor khi được đóng điện được báo bởi các đèn báo đặt ở ký hiệu cuộn dây. Trong động cơ có đặt 3 dây đèn báo tượng trưng cho 3 cuộn dây của động cơ. Nếu động cơ có điện thì các dây đèn này sáng.

b. Bảng ký hiệu

<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
Start	I0.0	Nút nhấn khởi động, thường hở
Stop	I0.1	Nút nhấn dừng, thường đóng
F1	I0.2	CB 3 pha, công tắc
F4	I0.4	Relay nhiệt, công tắc
Right	I0.5	Chọn chiều quay phải, thường hở
Left	I0.6	Chọn chiều quay trái, thường hở
H_Right	I1.0	Quay phải tốc độ nhanh, thường hở
H_Left	I1.1	Quay trái tốc độ nhanh, thường hở
Not aus	I1.2	Dừng khẩn cấp, công tắc
Q1	Q0.0	Khởi động từ K1, quay phải
Q2	Q0.1	Khởi động từ K2, quay trái
Q3	Q0.2	Khởi động từ K3, chạy Y
Q4	Q0.3	Khởi động từ K4, chạy D
R	Q0.4	Đèn báo quay phải
L	Q0.5	Đèn báo quay trái

c. Bài tập mẫu

Các bài tập về điều khiển động cơ có thể sử dụng được với mô hình này. Trong

phần này đưa ra 4 bài tập mẫu như sau:

Bài tập 1 (Sử dụng các công logic, lệnh ghi xóa tiếp điểm): Viết chương trình điều khiển động cơ Y/N bằng tay.

Bài tập 2 (Sử dụng các công logic, lệnh ghi xóa tiếp điểm): Viết chương trình điều khiển tốc độ và đảo chiều quay động cơ.

Bài tập (Sử dụng Timer): Viết chương trình điều khiển mở máy động cơ Y/□.

Bài tập 4 (Sử dụng Timer): Viết chương trình điều khiển bồn trộn.

1. *Bài tập 1* (Sử dụng các công logic, lệnh ghi xóa tiếp điểm): Viết chương trình điều khiển động cơ Y/N bằng tay.

Động cơ được điều khiển theo yêu cầu sau:

Khi ấn nút khởi động “Start” (I0.0), các contactor Q1 và Q3 đóng lại. Động cơ chạy ở chế độ Y. Sau đó nếu nút nhấn “Right” (I0.5) được ấn thì contactor Q3 tắt và contactor Q4 có điện. Động cơ chạy ở chế độ D.

Động cơ được cung cấp điện bởi CB 3 pha, và được bảo vệ bởi rơ le nhiệt và dừng bằng nút nhấn “Stop”..

2. *Bài tập 2* (Sử dụng các công logic, lệnh ghi xóa tiếp điểm): Viết chương trình điều khiển tốc độ và đảo chiều quay động cơ.

Sau khi được cấp điện, động cơ hoạt động như sau:

Trước tiên chọn chiều quay của động cơ bằng các nút nhấn “Right” hoặc “Left”. Khi ấn các nút này tương ứng các contactor Q1 hoặc Q2 đóng lại và các đèn báo quay phải “R” hoặc quay trái “L” sáng. Sau đó ấn nút khởi động “Start” thì động cơ sẽ quay ở tốc độ thấp trước (contactor Q3 có điện). Bây giờ có thể cho động cơ quay ở tốc độ cao hơn bằng cách ấn các nút tương ứng với các chiều quay “H-Right” hoặc “H-Left”. Khi ấn các nút này thì contactor Q4 có điện.

Động cơ được bảo vệ quá nhiệt và có dừng bằng nút nhấn “Stop”.

Ghi chú: Không cho phép đảo chiều trực tiếp, cũng như việc chuyển đổi tốc độ chỉ có thể từ thấp sang cao.

Bài tập (Sử dụng Timer): Viết chương trình điều khiển mở máy động cơ Y/□.

hương trình mở máy động cơ Y/□ theo yêu cầu sau:

a/. Sau khi CB 3 pha được đóng, ấn nút khởi động “Start” thì động cơ hoạt động ở chế độ Y. Sau thời gian 10s thì động cơ được chuyển sang hoạt động ở chế độ tam giác.

(Ghi chú: động cơ quay theo chiều phải).

b/. Khi ấn nút Right hoặc Left thì động cơ sẽ hoạt động ở chế độ sao/tam giác với chiều quay là chiều đã chọn. Tương ứng các đèn báo quay phải hoặc trái sáng.

Bài tập 4 (Điều khiển trình tự): Viết chương trình điều khiển bồn trộn.

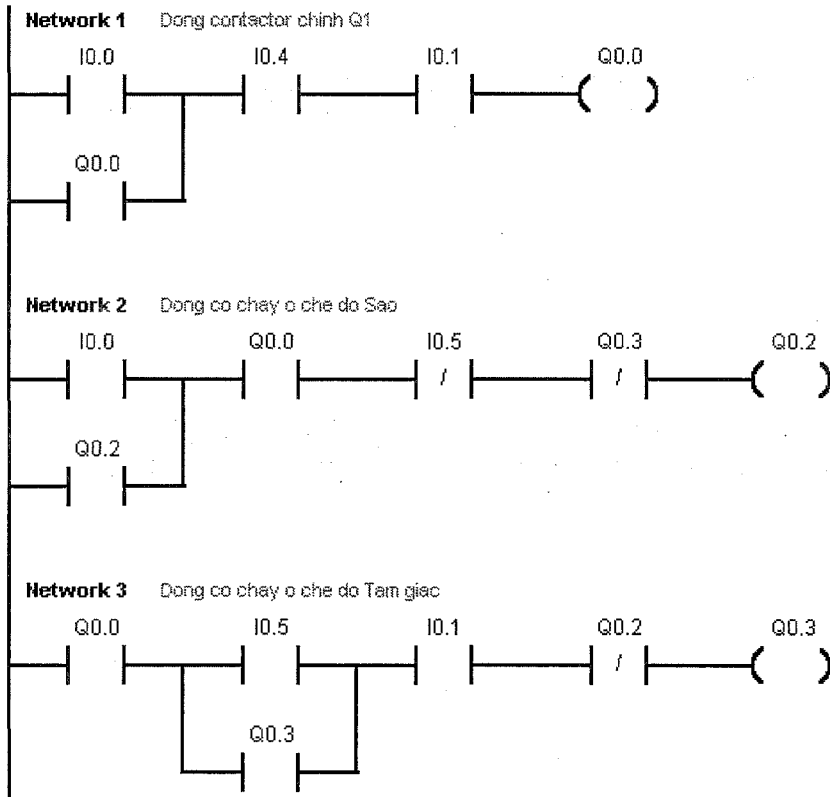
Viết chương trình điều khiển một bồn trộn theo yêu cầu sau: Khi ấn nút

“Start“, thì động cơ quay phải ở tốc độ thấp trong thời gian 10s dừng 5s quay trái 10s dừng 5s. Quá trình cứ thế lặp lại. Sau khoảng 20 lần lặp thì động cơ dừng 10s và sau đó quay phải ở tốc độ thấp được 5s thì chuyển sang quay ở tốc độ cao khoảng 30s thì dừng hẳn.

Bồn trộn có thể dừng bằng cách ấn nút dừng “Stop“.

Bài giải mẫu: (Bài tập 1):

Chương trình viết ở LAD:



Chương trình viết ở STL:

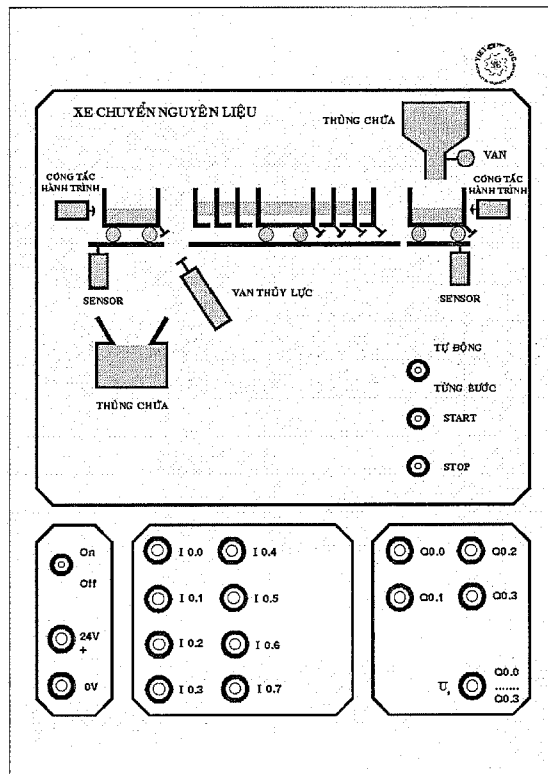

```

NETWORK 1          //Dong contactor chinh Q1
//
LD      IO.0
O       Q0.0
A       IO.4
A       IO.1
=       Q0.0

NETWORK 2          //Dong co chay o che do Sao
//
LD      IO.0
O       Q0.2
A       Q0.0
AN      IO.5
AN      Q0.3
=       Q0.2

NETWORK 3          //Dong co chay o che do Tam giac
//
LD      Q0.0
LD      IO.5
O       Q0.3
ALD
A       IO.1
AN      Q0.2
=       Q0.3
    
```

2.Mô hình xe chuyển nguyên liệu



a. Mô tả

Mô phỏng một xe vận chuyển nguyên liệu từ nơi này đến nơi khác với việc lấy nguyên liệu từ bồn chứa và xả nguyên liệu vào bồn chứa khác bằng các LED với nhiều màu sắc khác nhau. Cũng như các cảm biến và công tắc hành trình đều tạo ra tự động.

- Ứng dụng trong PLC cơ bản: Điều khiển tổ hợp logic.
- Ứng dụng trong PLC nâng cao: Điều khiển trình tự Cách thức nối dây cho mô hình như ở mục II.

b. Cách vận hành mô hình:

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào ra bằng phần mềm (đối với S7-200 dùng bảng Status chart)) và sau đó thực hiện mô phỏng với mô hình.

Nguyên liệu trong bồn chứa khi đang được xả hoặc được rót vào biểu thị bởi những dòng LED chạy, các đèn LED chạy đuổi tượng trưng cho xe đang di chuyển, van thủy lực được biểu thị bởi đèn sáng dần và tắt dần.

Tùy theo yêu cầu bài tập mà có thể lập trình xe chạy ở chế độ tự động, chế độ tay, hoặc hoạt động ở cả hai chế độ

c. Bảng ký hiệu

<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
Start	I0.0	Khởi động hệ thống, thường hở.
End 1	I0.1	Công tắc hành trình ở trạm xả, thường đóng
Fill 1	I0.2	Cảm biến báo xe rỗng, thường đóng.
End 2	I0.3	Công tắc hành trình trạm nạp, thường đóng.
Fill 2	I0.4	Cảm biến báo đầy, thường hở.
Stop	I0.5	Dừng, thường đóng.
Step	I0.6	Chế độ bước, thường hở.
Auto	I0.7	Chế độ tự động, thường hở.
Dir A	O0.0	Xe chạy về hướng A
Dir B	O0.1	Xe chạy về hướng B
Y1	O0.2	Van xả nguyên liệu
Y2	O0.3	Van thủy lực

d. Bài tập mẫu:

Xe vận chuyển nguyên liệu hoạt động như sau:

*Xe vận chuyển nguyên liệu có thể thực hiện qua công tắc chọn chế độ:

- Chế độ tự động: I0.6

- Chế độ bước: I0.7

*Vị trí cơ bản: Xe ở vị trí công tắc hành trình End 2 (I0.3 và xe chưa được làm đầy.

Chế độ tự động:

Khi xe ở vị trí cơ bản và công tắc chọn chế độ đặt ở chế độ tự động, khi nhấn nút khởi động (I0.0) thì van xả Y1 mở, vật liệu được đổ vào xe, cảm biến Fill 2 dùng để nhận biết xe đã được đổ đầy. Khi xe đầy thì van xả Y1 mất điện và xe chạy về hướng B sau thời gian ổn định 5s, xe dừng lại tại B (trạm nhận nguyên liệu) khi chạm công tắc hành trình S2. Xy lanh thủy lực của thiết bị xả được điều khiển và tấm chắn trên xe được mở vật liệu được rót vào bồn chứa. Khi xe xả hết vật liệu cảm biến S4 phát ra tín hiệu 1, pit tông thủy lực của thiết bị xả mất điện, tấm chắn trở về vị trí cũ, xe dừng 5 giây sau đó chạy về hướng A. Chu kỳ hoạt động được lặp lại.

Nếu trong chu kỳ hoạt động mà nút “dừng” được ấn thì quá trình vẫn tiếp tục cho đến khi xe trở về vị trí cơ bản (xe rỗng và ở trạm nhận nguyên liệu) và dừng hẳn.

Chế độ bước:

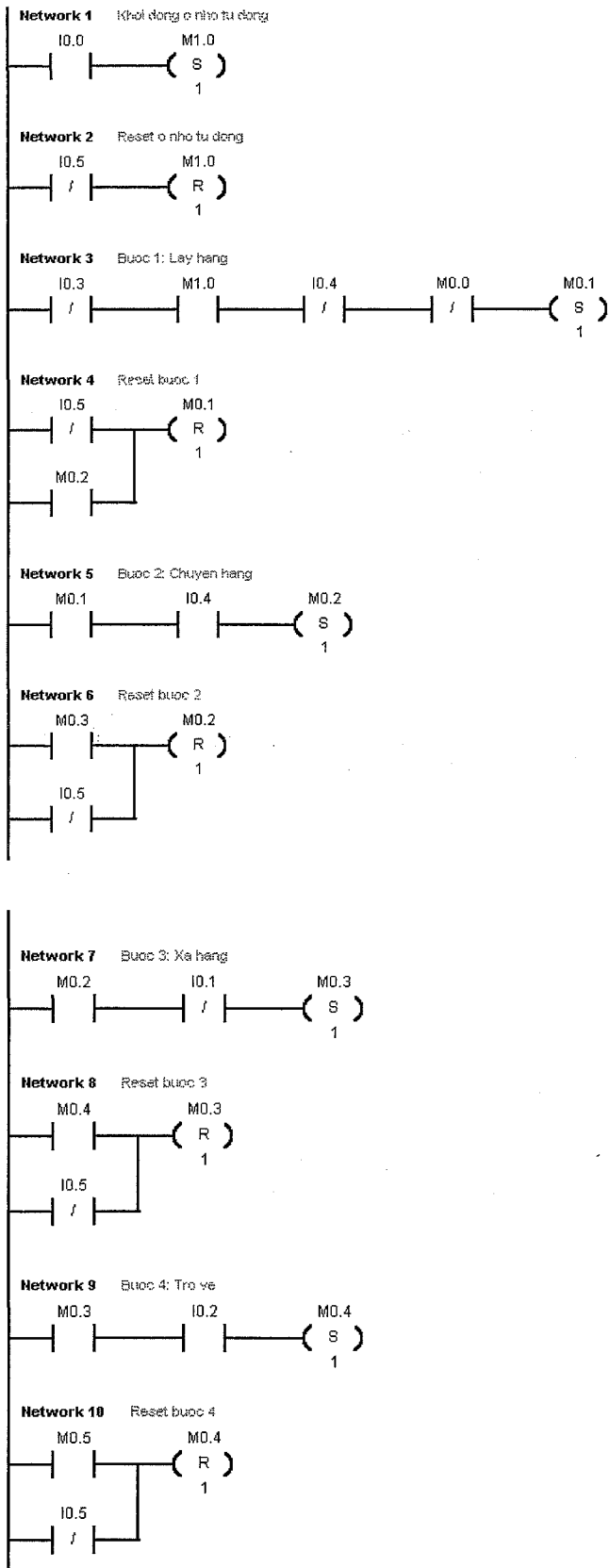
Ở mỗi bước thực hiện phải thông qua nút nhấn “start”.

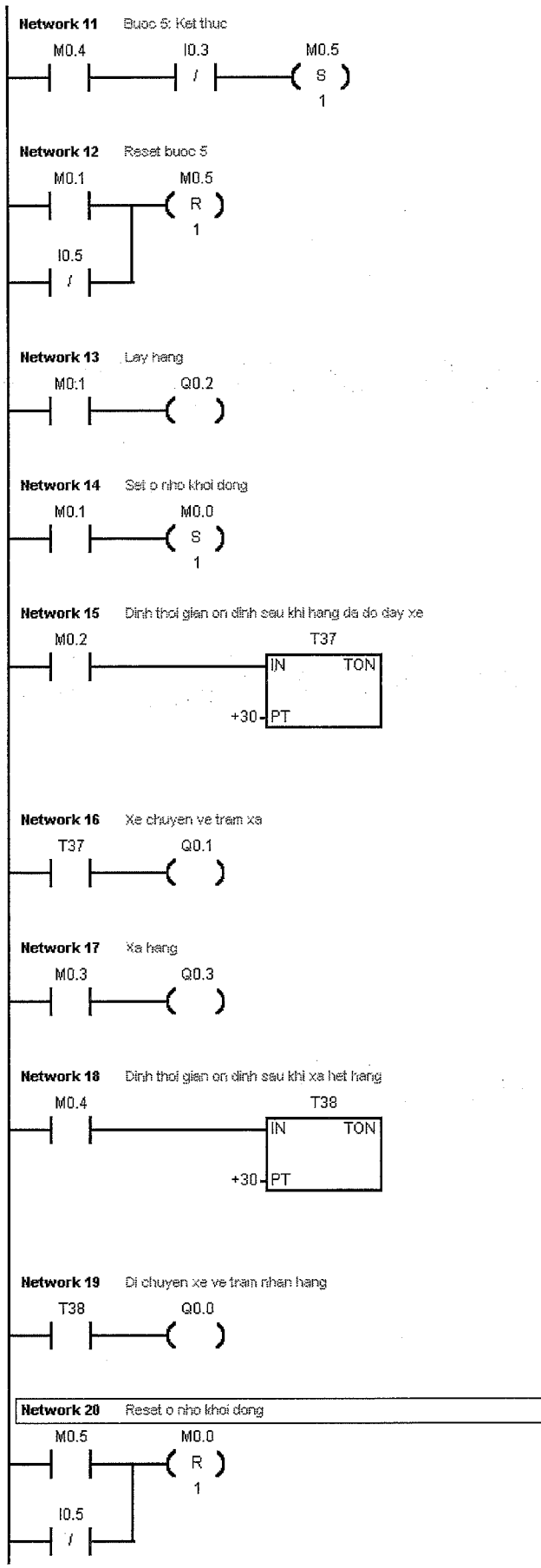
Ví dụ : khi ấn “start” xe đứng vị trí van xả được mở, khi xe đầy thì S3 tác động, van xả đóng lại. Nếu tiếp tục ấn “start” thì xe chạy về hướng B.

Hãy viết chương trình điều khiển xe chuyển nguyên liệu này và sau đó kết nối với thiết bị mô phỏng để kiểm tra hoạt động ở hai dạng:

- a. Điều khiển dùng tổ hợp logic
- b. Điều khiển trình tự

Bài giải mẫu: Chương trình được viết theo điều khiển trình tự
Chương trình được viết ở LAD:





```
NETWORK 1      //Khoi dong o nho tu dong
//
LD      IO.0
S      M1.0, 1

NETWORK 2      //Reset o nho tu dong
//
LDN     IO.5
R      M1.0, 1

NETWORK 3      //Buoc 1: Lay hang
//
LDN     IO.3
A      M1.0
AN     IO.4
AN     MO.0
S      MO.1, 1

NETWORK 4      //Reset buoc 1
//
LDN     IO.5
O      MO.2
R      MO.1, 1

NETWORK 5      //Buoc 2: Chuyen hang
//
LD      MO.1
A      IO.4
S      MO.2, 1

NETWORK 6      //Reset buoc 2
//
LD      MO.3
ON     IO.5
R      MO.2, 1

NETWORK 7      //Buoc 3: Xa hang
//
LD      MO.2
AN     IO.1
S      MO.3, 1

NETWORK 8      //Reset buoc 3
//
LD      MO.4
ON     IO.5
R      MO.3, 1

NETWORK 9      //Buoc 4: Tro ve
//
LD      MO.3
A      IO.2
S      MO.4, 1

NETWORK 10     //Reset buoc 4
//
LD      MO.5
ON     IO.5
R      MO.4, 1
```

```

NETWORK 11      //Buoc 5: Ket thuc
//
LD      MO.4
AN      IO.3
S      MO.5, 1

NETWORK 12      //Reset buoc 5
//
LD      MO.1
ON      IO.5
R      MO.5, 1

NETWORK 13      //Lay hang
//
LD      MO.1
=      QO.2

NETWORK 14      //Set o nha khoi dong
//
LD      MO.1
S      MO.0, 1

NETWORK 15      //Dinh thoi gian on dinh sau khi hang da do day xe
//
LD      MO.2
TON     T37, +30

NETWORK 16      //Xe chuyen ve tram xa
//
LD      T37
=      QO.1

NETWORK 17      //Xe hang
//
LD      MO.3
=      QO.3

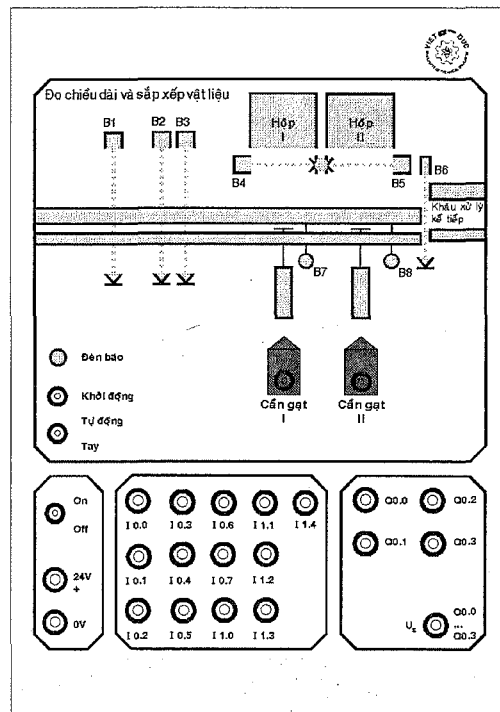
NETWORK 18      //Dinh thoi gian on dinh sau khi xa het hang
//
LD      MO.4
TON     T38, +30

NETWORK 19      //Di chuyen xe ve tram nhan hang
//
LD      T38
=      QO.0

NETWORK 20      //Reset o nha khoi dong
//
LD      MO.5
ON      IO.5
R      MO.0, 1

```

3. Đo chiều dài và sắp xếp vật liệu



a. Mô tả

Mô hình mô phỏng một hệ thống băng tải vận chuyển gỗ và sắp xếp các loại gỗ có chiều dài ngắn khác nhau vào các thùng chứa bằng các cần gạt khí nén. Hàng LED lớn ở trên tượng trưng cho các đoạn gỗ di chuyển trên băng tải, hàng nhỏ ở dưới là băng tải. Gỗ trong thùng chứa được sắp xếp thành hàng

- Ứng dụng trong PLC cơ bản: Điều khiển tổ hợp logic.
- Ứng dụng trong PLC nâng cao: Điều khiển trình tự

b. Cách vận hành mô hình

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào ra bằng phần mềm (đối với S7-200 dùng bảng Status chart)) và sau đó thực hiện mô phỏng với mô hình.

Chỉ có thể đặt gỗ trên băng tải được nếu băng tải hoạt động. Việc đặt những thanh gỗ dài ngắn khác nhau được tạo ra bằng cách ấn nút “Khởi động” lâu hay ngắn. Khi các thanh gỗ này đi qua các cảm biến quang thì các cảm biến này sẽ thay đổi trạng thái. Các cần gạt nếu được kích hoạt thì các thanh gỗ ngay vị trí của nó sẽ biến mất và sau đó một thanh LED trong hộp sáng lên cho biết gỗ đã vào trong hộp. Tùy theo bài tập đặt ra mà có thể phát hiện được người thực hành viết chương trình sai hay đúng.

c. Bảng ký hiệu:

<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
Khởi động	<i>I0.</i>	Khởi động hệ thống, thường hở.
B1	<i>I0.</i>	Cảm biến quang, thường đóng
B2	<i>I0.</i>	Cảm biến quang, thường đóng
B3	<i>I0.</i>	Cảm biến quang, thường đóng
B4	<i>I0.</i>	Cảm biến quang, thường đóng
B5	<i>I0.</i>	Cảm biến quang, thường đóng
B6	<i>I0.</i>	Cảm biến quang, thường đóng
B7	<i>I0.</i>	Công tắc hành trình, thường đóng
B8	<i>II.</i>	Công tắc hành trình, thường đóng
S I	<i>II.</i>	Nút nhấn đưa gỗ vào thùng I
S II	<i>II.</i>	Nút nhấn đưa gỗ vào thùng II
Tự	<i>II.</i>	Công tắc chọn chế độ tự động
Tay	<i>II.</i>	Công tắc chọn chế độ tay
Bã	<i>O0.</i>	Băng tải vận chuyển gỗ
Cà	<i>O0.</i>	Cần gạt đưa gỗ vào thùng I
Cà	<i>O0.</i>	Cần gạt đưa gỗ vào thùng II
Đèn	<i>O0.</i>	Đèn báo băng tải sẵn sàng nhận gỗ

a. Bài tập mẫu:

Mô hình đo chiều dài và sắp xếp vật liệu được dùng để mô phỏng việc sắp xếp các thanh gỗ có chiều dài ngắn khác nhau trên băng tải vào các thùng khác nhau.

Hệ thống có thể hoạt động ở hai chế độ: tự động và tay.

Chế độ tự động:

Khi “đèn báo” sáng báo hiệu hệ thống sẵn sàng làm việc. Bằng cách nhấn nút “Khởi động” thì “đèn báo” tắt và một tín hiệu khởi động được tạo ra. Các thanh gỗ đơn được đặt lên băng tải (bằng nút ấn khởi động) và băng tải chuyển động.

Chiều dài của thanh gỗ được nhận biết bằng các cảm biến quang B1, B2, B3. Điều này có nghĩa:

Cảm biến B1 tác động tương ứng gỗ ngắn.

Cảm biến B1 và B2 tác động tương ứng gỗ trung bình.

Cảm biến B2, B2 và B3 tác động tương ứng gỗ dài.

Khi gỗ ngắn đến cảm biến B7 thì “Tay gạt 1” sẽ đẩy thanh gỗ này vào thùng 1. Khi gỗ trung bình đến cảm biến B8 thì “Tay gạt 2” sẽ đẩy thanh gỗ này vào thùng 2.

Gỗ dài thì được di chuyển tiếp tục đến khâu xử lý kế tiếp. Tay gạt 1 và tay gạt 2 được sử dụng bằng khí nén được điều khiển khoảng 1 giây và sau đó trở về vị trí cơ bản của nó.

Sau khi sự sắp xếp thành công (tương ứng với các cảm biến quang B4, B5 và B6) thì thiết bị tự động phát tín hiệu khởi động tiếp theo và băng tải lại vận chuyển gỗ tiếp tục.

Chế độ tay:

Ở chế độ tay mỗi thanh gỗ được xử lý xong thì yêu cầu khởi động lại hệ thống bằng tay. Tín hiệu khởi động chỉ được phép xử lý nếu việc điều khiển trước đây được báo bằng đèn. Ngay sau khi sắp xếp thành công thì đèn báo lại sáng

Tay gạt I và II được điều khiển bằng tay từ nút nhấn điều khiển.

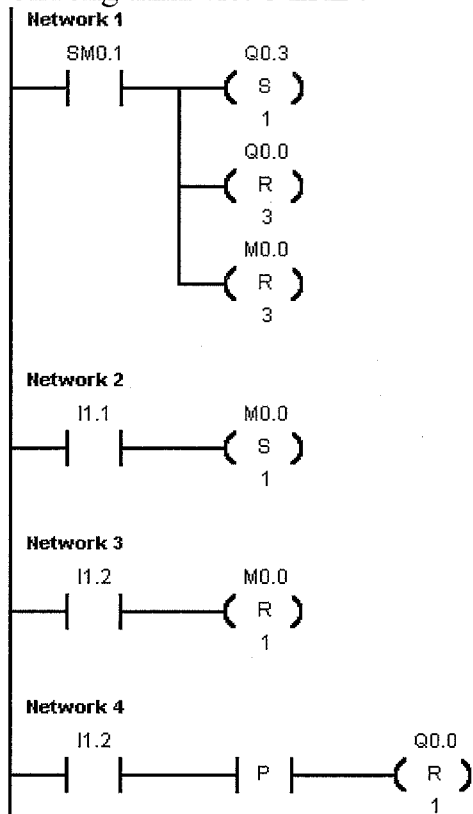
Ghi chú: Đây chỉ là một khâu sắp xếp gỗ, và gỗ được đặt vào băng tải nhờ vào nút nhấn khởi động. Điều này có nghĩa nút nhấn khởi động vừa đóng vai trò khởi động vừa là nơi cung cấp gỗ cho băng tải.

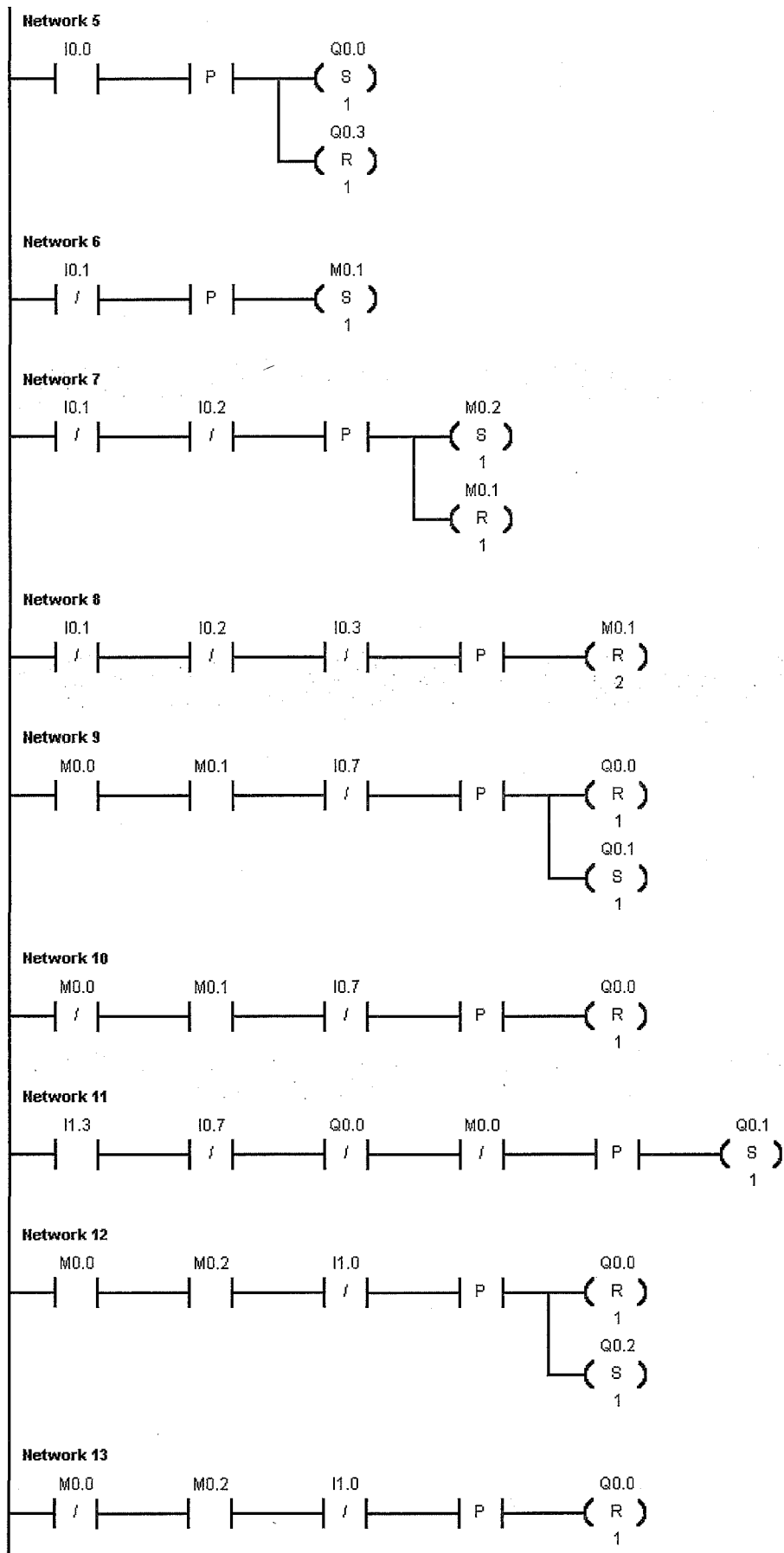
Hãy viết chương trình điều khiển hệ thống này và kiểm tra bằng mô hình ở hai cách:

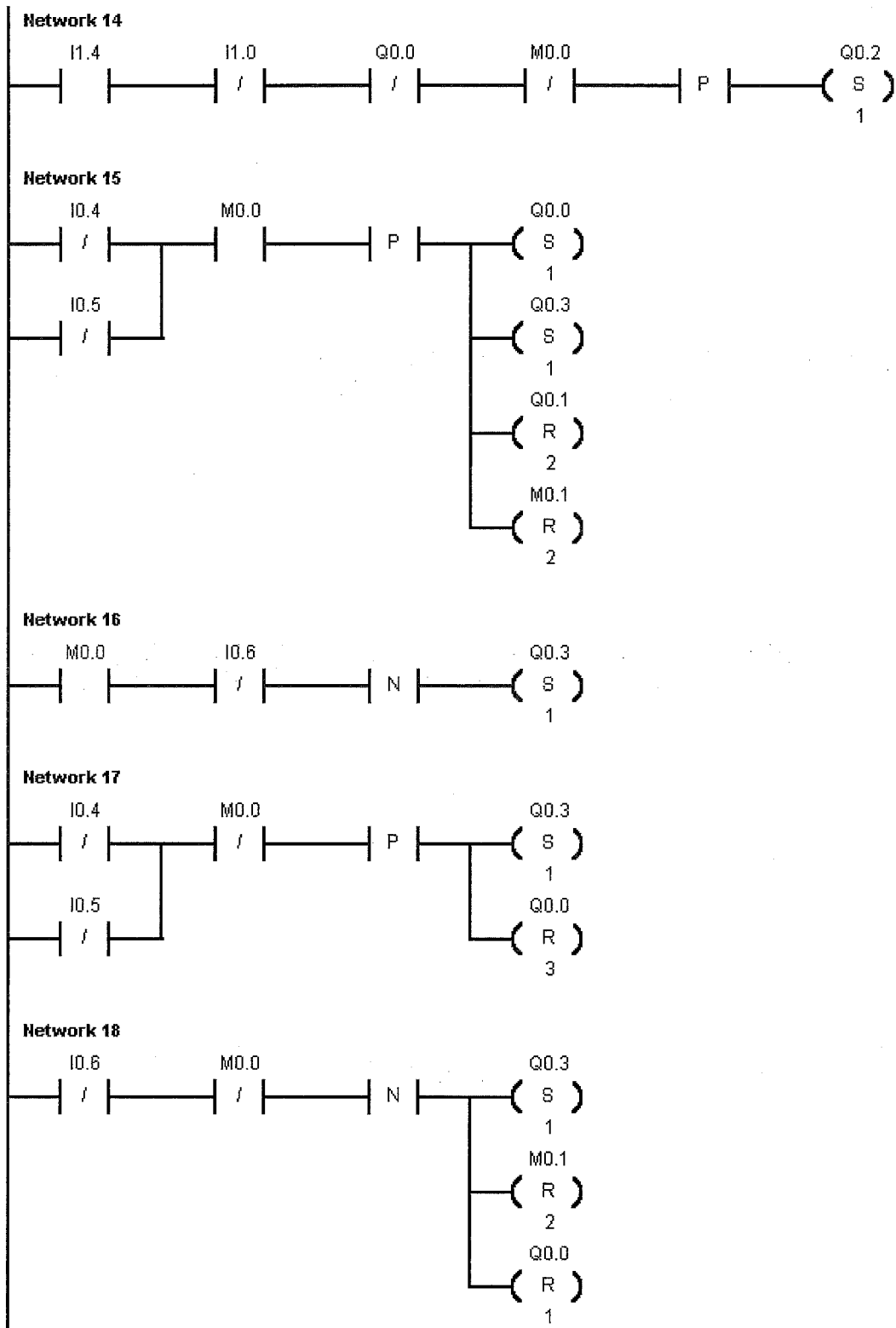
- Dùng tổ hợp logic
- Dùng phương pháp điều khiển trình tự

Bài giải mẫu (dùng tổ hợp logic):

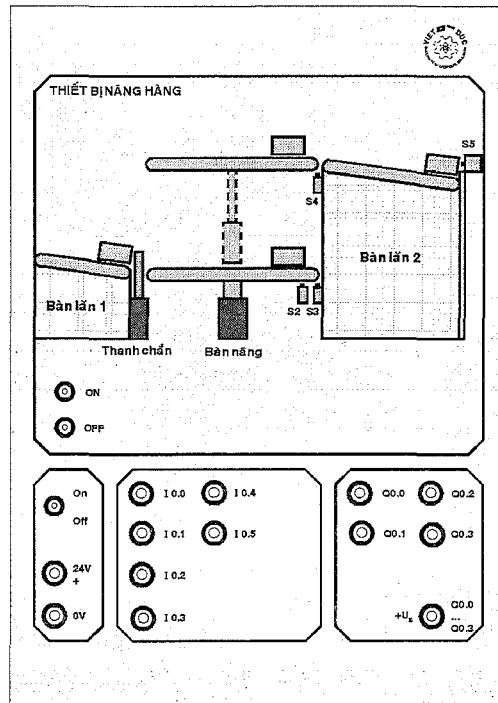
Chương trình viết ở LAD:







4. Thiết bị nâng hàng



a. Mô tả

Mô phỏng một hệ thống nâng hàng bằng các đèn LED với nhiều màu sắc khác nhau. Hàng hóa từ bàn lặn thấp được đưa lên cao sang bàn lặn 2 nhờ vào bàn nâng. Hệ thống này thường thấy trong việc sắp xếp hàng hóa trong kho hoặc đưa hàng hoá vào các khoang chứa hàng của máy bay.

Ứng dụng:

- PLC cơ bản: Điều khiển tổ hợp logic
- PLC nâng cao: Điều khiển trình tự
- Kết nối dây như trình bày ở mục II.

b. Cách vận hành mô hình

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào ra bằng phần mềm (đối với S7-200 dùng bảng Status chart)) và sau đó thực hiện mô phỏng với mô hình.

Vật thể trên bàn lặn 1 xem như do một nơi khác chuyển đến. Cứ sau khi vật thể được đưa sang bàn nâng thì một vật thể trên bàn lặn 1 lại xuất hiện. Vật thể trên bàn nâng được đưa sang đầu bên kia của bàn nâng nhờ vào băng tải trên bàn nâng (trọng trung bởi LED chạy đuổi). Khi vật thể được đưa sang bàn lặn 2 thì nó sẽ lăn đến cuối bàn và dừng lại 1s và sau đó tự biến mất.

c. Bảng ký hiệu

<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
ON	I0.	Khởi động hệ thống, thường hở.
OFF	I0.	Dừng hệ thống, thường đóng
S2	I0.	Báo hàng ở vị trí cuối bàn nâng, thường
S3	I0.	Giới hạn dưới bàn nâng, thường đóng
S4	I0.	Giới hạn trên bàn nâng, thường đóng
S5	I0.	Báo hàng ở cuối bàn lăn 2
Thanh chắn	O0.0	Chặn hàng hóa ở bàn nâng 1
Băng tải	O0.1	Băng tải chuyển hàng
K1	O0.2	Nâng hàng hoá lên
K2	O0.3	Hạ bàn nâng xuống

d. Bài tập mẫu

Thiết bị nâng hàng hoạt động như sau:

Hàng hóa được đặt sẵn trên bàn lăn 1. Bàn nâng ở vị trí giới hạn dưới thì khi ấn nút khởi động “ON”, băng tải trên bàn nâng hoạt động, đồng thời thanh chắn hạ xuống (sử dụng khí nén) khoảng 2s để hàng hóa được đưa sang bàn nâng. Sau đó thanh chắn trở về vị trí cũ.

Khi hàng hóa đến vị trí cuối bàn nâng (S2), thì băng tải dừng. Khởi động từ K1 của động cơ M1 có điện kéo bàn nâng lên. Khi đến giới hạn trên thì bàn nâng dừng lại. Băng tải bắt đầu chuyển động đưa hàng sang bàn lăn 2. Khi hàng đến công tắc hành trình S5 thì băng tải dừng. Khởi động từ K2 của động cơ M1 có điện hạ bàn nâng xuống, đến giới hạn dưới thì dừng.

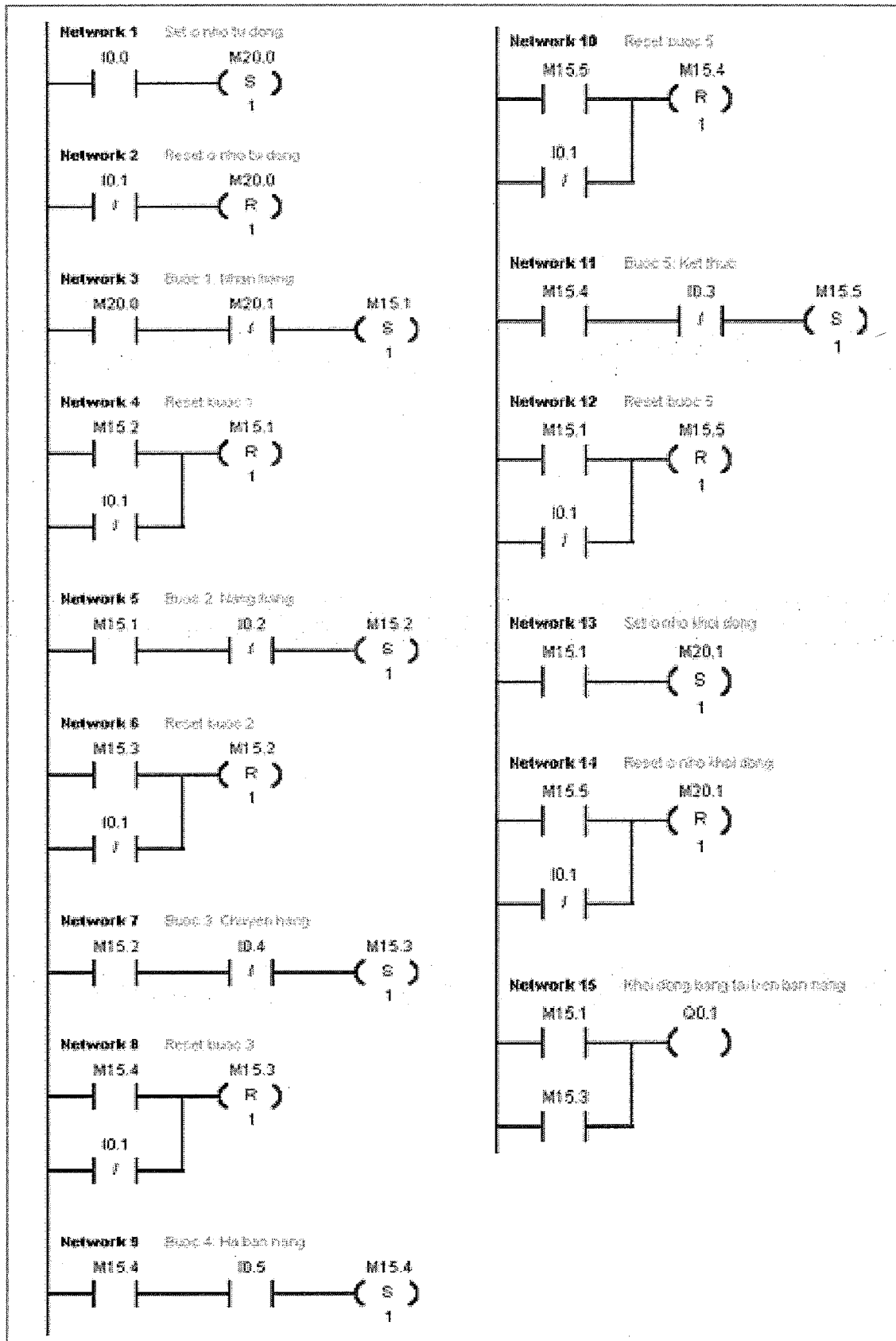
Quá trình mới lại bắt đầu cho đến khi nào nhấn nút dừng “OFF”

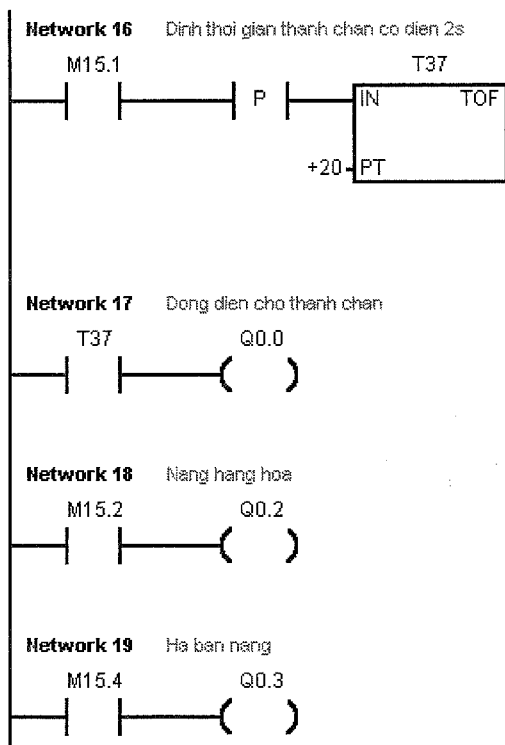
Hãy viết chương trình điều khiển thiết bị nâng trên và kiểm tra bằng mô hình theo hai cách:

- Viết theo tổ hợp Logic
- Viết theo điều khiển trình tự.

Bài giải mẫu (Viết theo phương pháp trình tự):

Chương trình viết ở LAD:





Chương trình viết ở STL:

```

NETWORK 1      //Set o nhỏ tu động
//
LD      IO.0
S       M20.0, 1

NETWORK 2      //Reset o nhỏ tu động
//
LDN     IO.1
R       M20.0, 1

NETWORK 3      //Bước 1: Nhận hàng
//
LD      M20.0
AN      M20.1
S       M15.1, 1

NETWORK 4      //Reset bước 1
//
LD      M15.2
ON      IO.1
R       M15.1, 1

NETWORK 5      //Bước 2: Nâng hàng
//
LD      M15.1
AN      IO.2
S       M15.2, 1

NETWORK 6      //Reset bước 2
//
LD      M15.3
ON      IO.1
R       M15.2, 1

NETWORK 7      //Bước 3: Chuyển hàng
//
LD      M15.2
AN      IO.4
S       M15.3, 1

NETWORK 8      //Reset bước 3
//
LD      M15.4
ON      IO.1
R       M15.3, 1

NETWORK 9      //Bước 4: Hạ ban nâng
//
LD      M15.4
A       IO.5
S       M15.4, 1

NETWORK 10     //Reset bước 5
//
LD      M15.5
ON      IO.1
R       M15.4, 1

```



```
NETWORK 11      //Buoc 5: Ket thuc
//
LD      M15.4
AN      IO.3
S      M15.5, 1

NETWORK 12      //Reset buoc 5
//
LD      M15.1
ON      IO.1
R      M15.5, 1

NETWORK 13      //Set o nho khai dong
//
LD      M15.1
S      M20.1, 1

NETWORK 14      //Reset o nho khai dong
//
LD      M15.5
ON      IO.1
R      M20.1, 1

NETWORK 15      //Khai dong bang tai tren ban nang
//
LD      M15.1
O      M15.3
=      Q0.1

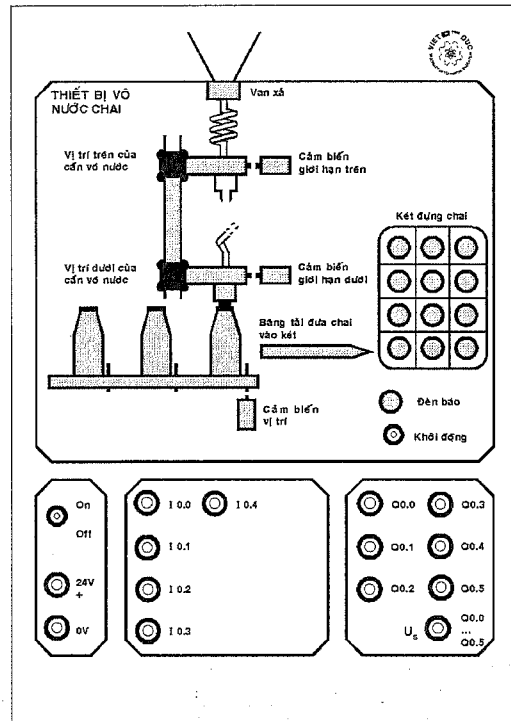
NETWORK 16      //Dinh thoi gian thanh chan co dien 2s
//
LD      M15.1
EU
TOF     T37, +20

NETWORK 17      //Dong dien cho thanh chan
//
LD      T37
=      Q0.0

NETWORK 18      //Nang hang hoa
//
LD      M15.2
=      Q0.2

NETWORK 19      //Ha ban nang
//
LD      M15.4
=      Q0.3
```

5. Thiết bị vô nước chai



a. Mô tả

Mô phỏng một thiết bị vô nước chai có các cảm biến, công tắc hành trình và sự chuyển động bằng các LED

Ứng dụng:

- PLC cơ bản: Điều khiển tổ hợp logic
- PLC nâng cao: Điều khiển trình tự Cách kết nối dây như ở mục II

b. Cách vận hành mô hình

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào ra bằng phần mềm (đối với S7-200 dùng bảng Status chart)) và sau đó thực hiện mô phỏng với mô hình.

Hai chai bìa bên phải được xem là chai rỗng. Chai ở vị trí thứ ba được xem như chai đã được đưa đến đúng vị trí. Nước trong chai dâng lên được mô phỏng bằng đèn LED sáng dần. Tùy vào sự sáng dần này mà có thể định thời gian làm đầy chai. Khi chai đã được đổ đầy nước và nếu băng tải vận chuyển chai hoạt động thì chai đầy tự động được chuyển sang băng tải đưa chai vào két. Một tín hiệu sẽ phát ra nếu chai đã đúng ở vị trí trong két. Khi két đạt đến 12 thì nó không thể tự Reset được. Để có thể xóa các LED trong két này phải ấn nút “Khởi động”.

Để hoạt động giống thực tế thì khi cần vô nước đến miệng chai phải dừng lại 1s để ổn định.

c. Bảng ký hiệu

<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
S1	I0.	Giới hạn trên của cần vô nước, thường đóng
S2	I0.	Giới hạn dưới của cần vô nước, thường đóng
S3	I0.	Cảm biến vị trí chai, thường hở
S4	I0.	Khởi động hệ thống, thường hở
S5	I0.	Chai đúng vị trí trong két, thường hở
K	O0.	Van xả nước
K	O0.	Ha cần vô nước xuống
K	O0.	Nâng cần vô nước lên
K	O0.	Băng tải vận chuyển chai rỗng
K	O0.	Đèn báo két đầy

Bài tập mẫu:

Thiết bị vô nước chai hoạt động như sau:

Trước khi vận hành thiết bị vô nước chai thì các chai rỗng phải được đặt lên băng tải. Nếu sau đó nút nhấn khởi động (I0.3) được tác động, thì băng tải sẽ vận chuyển chai rỗng với thời gian trì hoãn ban đầu là 1s. Băng tải dừng lại khi có một chai đến cảm biến vị trí (I0.2).

Bây giờ cần vô nước sẽ hạ từ trên xuống, khi đến giới hạn dưới (I0.1) thì dừng lại, sau đó 1s thì van xả sẽ được mở đổ nước vào chai, van xả sẽ đóng lại khi chai đầy thời gian làm đầy kéo dài khoảng 3s.

Sau khi van xả đóng lại 1s thì cần vô nước được nâng lên, đến giới hạn trên (I0.0) thì dừng lại. Sau đó 1s thì băng tải vận chuyển chai rỗng lại tiếp tục và quá trình cứ thế lặp lại.

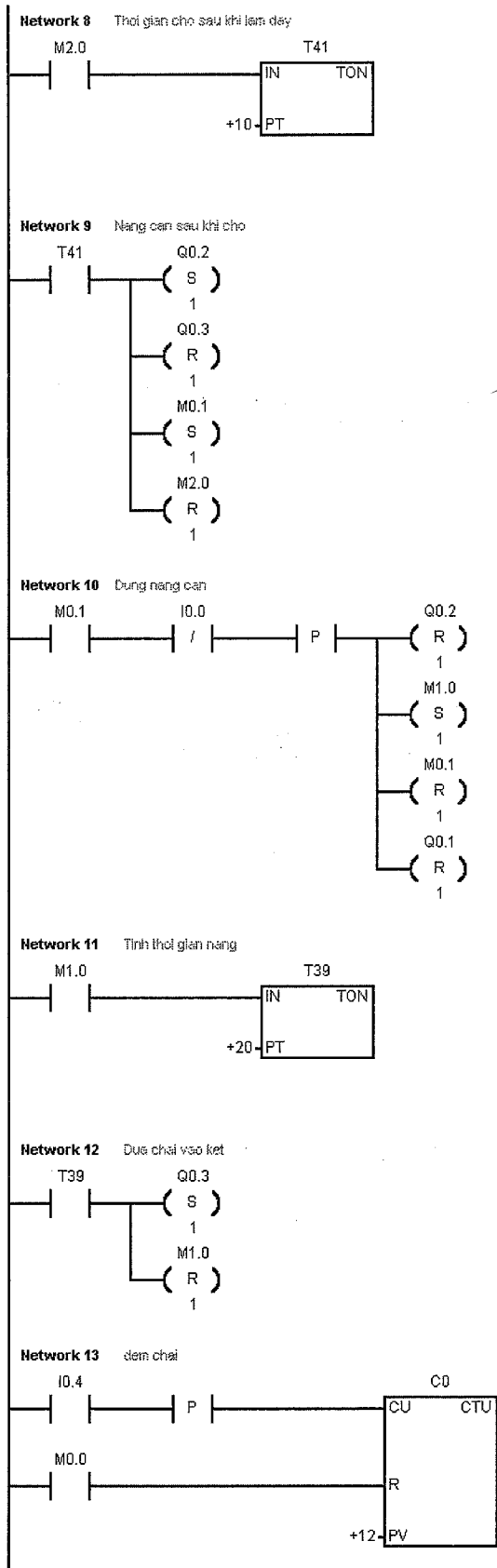
Chai đã đổ đầy nước được đưa sang băng tải đưa chai vào két khi băng tải chai rỗng hoạt động, khi chai đúng vị trí trong két thì có một tín hiệu phát ra (I0.4).

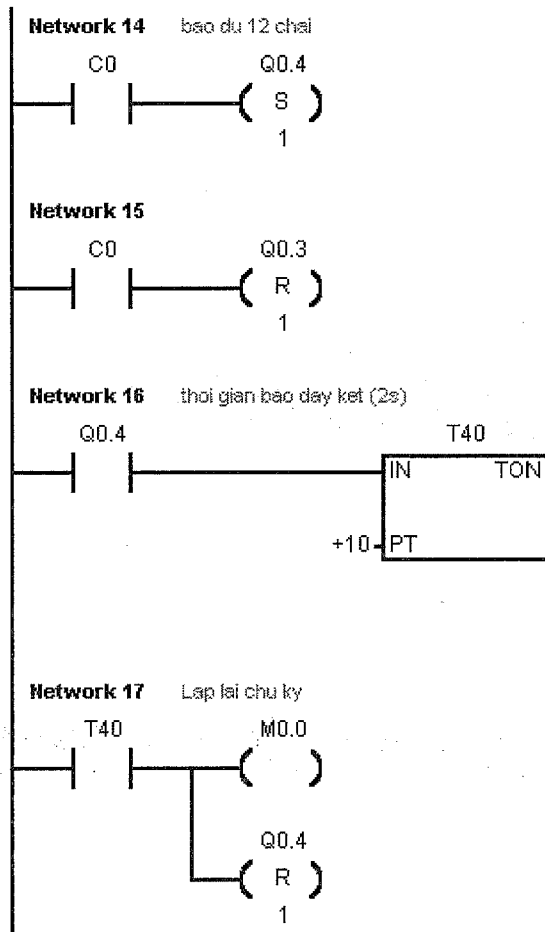
Quá trình được lặp đi lặp lại cho đến khi nào số lượng chai trong két đủ 12 thì đèn báo sáng lên và hệ thống dừng lại. Quá trình mới lại bắt đầu khi nút nhấn khởi động được tác động.

Hãy viết chương trình điều khiển và kiểm tra bằng mô hình.

Bài giải mẫu (Chương trình được viết theo tổ hợp logic):

Chương trình viết ở LAD





Chương trình được viết ở STL:

```

NETWORK 1      //khởi động hệ thống
//Khi nhận IO.3 (NO) thì bảng tài sẽ hoạt động
LD      IO.3
LD      MO.0
EU
OLD
R       Q0.0, 3
S       Q0.3, 1

NETWORK 2      //dung bảng tài-va ha can
//
LD      IO.2
EU
R       Q0.3, 1
S       Q0.1, 1
R       Q0.2, 1

NETWORK 3      //Dung ha can va on dinh 2s
//
LDN     IO.1
EU
R       Q0.1, 1
S       MO.5, 1

NETWORK 4      //thời gian on dinh
//
LD      MO.5
TON     T37, +20

NETWORK 5      //Ho van xa
//
LD      T37
S       Q0.0, 1
R       Q0.3, 1
R       MO.5, 1

NETWORK 6      //Tinh thời gian lam day
//
LD      Q0.0
TON     T38, +40

NETWORK 7      //Dong van xa
//
LDN     Q0.1
A       T38
R       Q0.0, 1
S       M2.0, 1

NETWORK 8      //Thời gian cho sau khi lam day
//
LD      M2.0
TON     T41, +10

```

```

NETWORK 9      //Nang can sau khi cho
//
LD      T41
S       Q0.2, 1
R       Q0.3, 1
S       M0.1, 1
R       M2.0, 1

NETWORK 10     //Dung nang can
//
LD      M0.1
AN      IO.0
EU
R       Q0.2, 1
S       M1.0, 1
R       M0.1, 1
R       Q0.1, 1

NETWORK 11     //Tinh thoi gian nang
//
LD      M1.0
TON     T39, +20

NETWORK 12     //Dua chai vao ket
//
LD      T39
S       Q0.3, 1
R       M1.0, 1

NETWORK 13     //dem chai
//
LD      IO.4
EU
LD      M0.0
CTU     C0, +12

NETWORK 14     //bao du 12 chai
//
LD      C0
S       Q0.4, 1

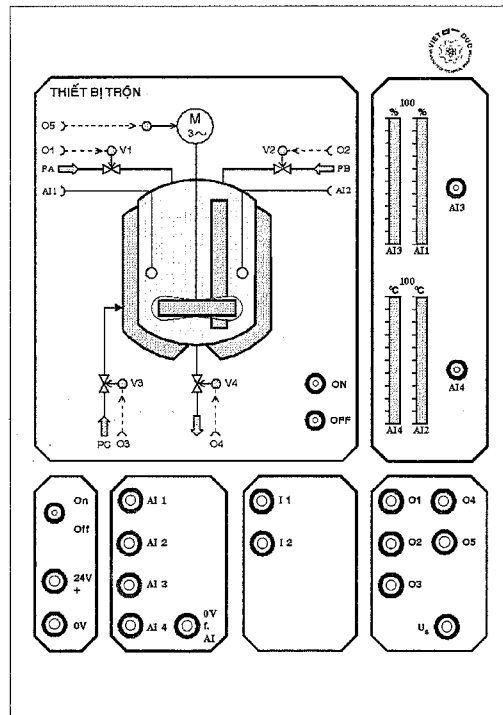
NETWORK 15
LD      C0
R       Q0.3, 1

NETWORK 16     //thoi gian bao day ket (2s)
//
LD      Q0.4
TON     T40, +10

NETWORK 17     //Lap lai chu ky
//
LD      T40
=       M0.0
R       Q0.4, 1

```

6. Thiết bị trộn hóa chất



a. Mô tả

Mô phỏng một thiết bị trộn định lượng, sử dụng các cảm biến analog để đo lượng chất lỏng chứa trong bình và đo nhiệt độ trong bình. Nhiệt độ và lượng chất lỏng trong bình có thể được điều chỉnh trước bằng các phím chỉnh định bên ngoài.

Ứng dụng trong PLC nâng cao (điều khiển tuần tự, xử lý tín hiệu analog, các phép toán, bộ điều chỉnh 2 điểm...)

b. Cách vận hành mô hình

Vài nét về mô hình:

Đây là mô hình thiết bị trộn dùng các cảm biến analog. Mô hình này được thiết kế cho các PLC có gắn các mô đun ngõ vào analog có ngõ vào là điện áp (0..10V). Tùy theo loại PLC và cũng tùy theo các modul ngõ vào analog hiện có mà chúng ta có thể đưa ra một số bài tập phù hợp.

Ghi chú:

Cảm biến đo nhiệt độ ở mô hình này được thiết kế phụ thuộc vào V3. Có nghĩa là nếu V3 được cung cấp điện thì nhiệt độ tăng dần từ khoảng 5°C lên và đến cực đại là 100°C. Nếu V3 mất điện thì nhiệt độ tự động giảm xuống từ từ.

Đối với cảm biến đo lượng chất lỏng, thì nó chỉ đo được tại mỗi thời điểm chỉ với một bơm chất lỏng. Tức là nếu như bơm A hoạt động thì bơm B phải ngưng và ngược lại.

c. Bảng ký hiệu

Ký hiệu	Địa chỉ	Chú thích
AI1	AIW0	Cảm biến nhận biết lượng chất lỏng trong bình
AI2	AIW2	Cảm biến nhận biết nhiệt độ trong bình
AI3	AIW4	Đặt chỉnh trước lượng chất lỏng cần thiết
AI4	AIW6	Đặt chỉnh trước nhiệt độ cần thiết
ON	I0.0	Khởi động hệ thống, thường hở
OF	I0.1	Dừng hệ thống, thường đóng
V1	Q0.0	Bơm chất lỏng A
V2	Q0.1	Bơm chất lỏng B
V3	Q0.2	Cung cấp nhiệt cho bình trộn
V4	Q0.3	Van xả chất lỏng
M1	Q0.4	Quạt trộn hóa chất

d. Bài tập mẫu:

Thiết bị trộn hóa chất hoạt động như sau:

Có hai loại chất lỏng A và B cần được trộn với nhau theo tỷ lệ 1/3. Nhiệt độ cần thiết để trộn hai chất này được đặt chỉnh ở AI4. Lượng chất lỏng muốn trộn được đặt chỉnh ở AI3.

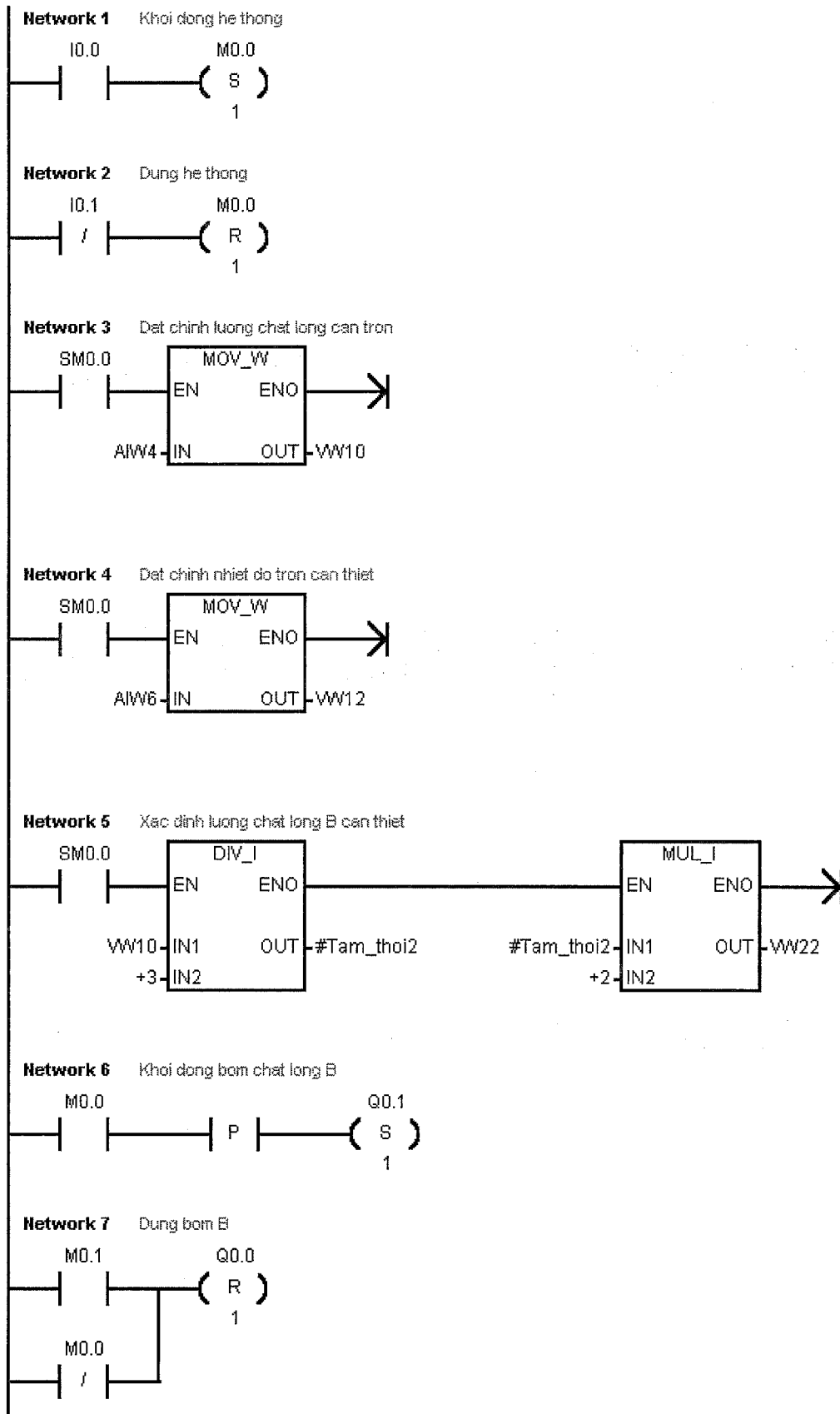
Khi ấn nút khởi động “ON” thì bơm B hoạt động trước. Sau khi chất lỏng A đạt đến mức cần thiết thì dừng lại (được nhận biết bởi cảm biến AI1). Nhiệt độ trong bồn trộn được tăng dần lên đến giá trị đặt (sai số cho phép là $\pm 10\%$, được nhận biết bởi cảm biến AI2) thì bơm chất lỏng A hoạt động, đồng thời quạt trong bồn trộn cũng quay. Khi chất lỏng A đã đổ đúng lượng cho phép thì bơm A dừng. Quạt trộn tiếp tục quay trong khoảng thời gian 10s nữa thì dừng lại. Sau đó van xả tự động mở ra để xả chất lỏng đã trộn vào bồn chứa.

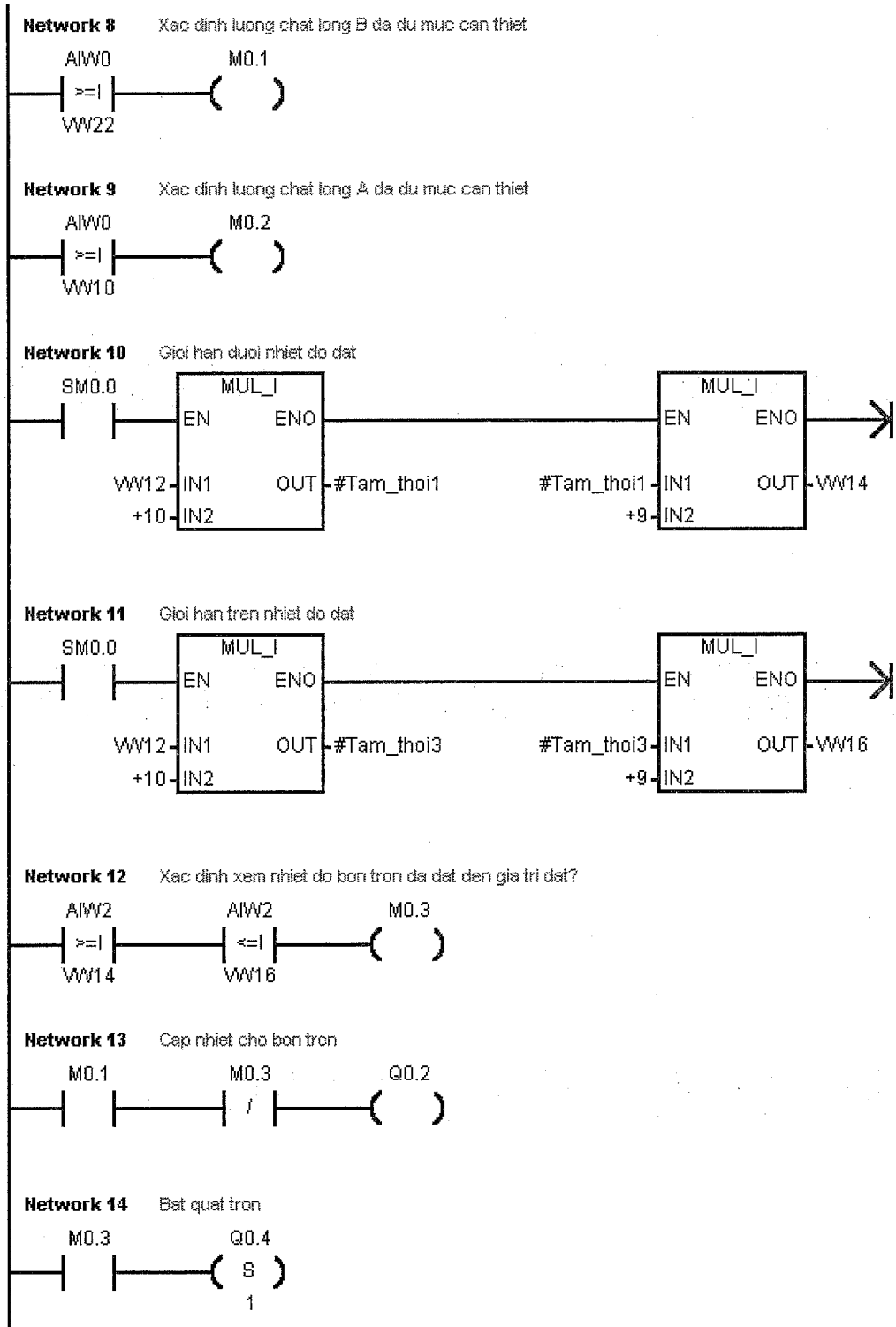
Quá trình mới lại bắt đầu, nếu nút nhấn “ON” được ấn. Hệ thống có thể dừng nếu nhấn nút dừng “OFF”.

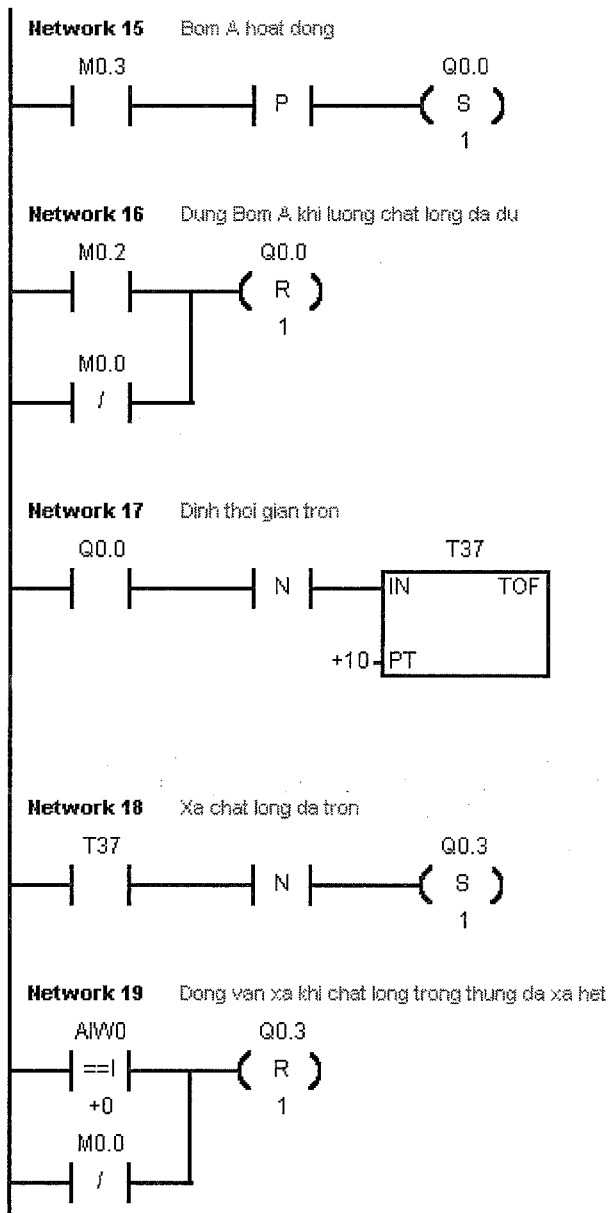
Hãy viết chương trình điều khiển bồn trộn này.

Bài giải mẫu:

Chương trình viết ở LAD:







Chương trình viết ở STL:

```

NETWORK 1      //Khởi động hệ thống
//
LD      IO.0
S      MO.0, 1

NETWORK 2      //Dừng hệ thống
//
LDN     IO.1
R      MO.0, 1

NETWORK 3      //Đặt chỉnh lượng chất lỏng cần trộn
//
LD      SMO.0
MOVW   AIW4, VW10

NETWORK 4      //Đặt chỉnh nhiệt độ trộn cần thiết
//
LD      SMO.0
MOVW   AIW6, VW12

NETWORK 5      //Xác định lượng chất lỏng B cần thiết
//
LD      SMO.0
MOVW   VW10, #Tam_thoi2
AENO
/I     +3, #Tam_thoi2
AENO
MOVW   #Tam_thoi2, VW22
*I     +2, VW22

NETWORK 6      //Khởi động bơm chất lỏng B
//
LD      MO.0
EU
S      QO.1, 1

NETWORK 7      //Dừng bơm B
//
LD      MO.1
ON     MO.0
R      QO.0, 1

NETWORK 8      //Xác định lượng chất lỏng B đã đủ mức cần thiết
//
LDW>=  AIW0, VW22
=      MO.1

NETWORK 9      //Xác định lượng chất lỏng A đã đủ mức cần thiết
//
LDW>=  AIW0, VW10
=      MO.2

NETWORK 10     //Giới hạn dưới nhiệt độ đặt
//
LD      SMO.0
MOVW   VW12, #Tam_thoi1
AENO
*I     +10, #Tam_thoi1
AENO
MOVW   #Tam_thoi1, VW14
*I     +9, VW14

```

```

NETWORK 11      //Giới hạn trên nhiệt độ đất
//
LD      SMO.0
MOVW   VW12, #Tam_thoi3
AENO
*I     +10, #Tam_thoi3
AENO
MOVW   #Tam_thoi3, VW16
*I     +9, VW16

NETWORK 12      //Zac dinh xem nhiệt độ bồn tròn đã đạt đến giá trị đất?
//
LDW>=  AIW2, VW14
AW<=   AIW2, VW16
=      MO.3

NETWORK 13      //Cấp nhiệt cho bồn tròn
//
LD      MO.1
AN      MO.3
=      QO.2

NETWORK 14      //Bắt quạt tròn
//
LD      MO.3
S      QO.4, 1

NETWORK 15      //Bơm & hoạt động
//
LD      MO.3
EU
S      QO.0, 1

NETWORK 16      //Dừng Bơm & khi lượng chất lỏng đã đủ
//
LD      MO.2
ON      MO.0
R      QO.0, 1

NETWORK 17      //Đinh thời gian tròn
//
LD      QO.0
ED
TOF    T37, +10

NETWORK 18      //Xả chất lỏng đã tròn
//
LD      T37
ED
S      QO.3, 1

NETWORK 19      //Dòng van xả khi chất lỏng trong thùng đã xả hết
//
LDW=   AIW0, +0
ON      MO.0
R      QO.3, 1

```

TÀI LIỆU THAM KHẢO

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

