

BÀI 1: ĐẠI CƯƠNG VỀ ĐIỀU KHIỂN LẬP TRÌNH.

1. Tổng quát về điều khiển lập trình.

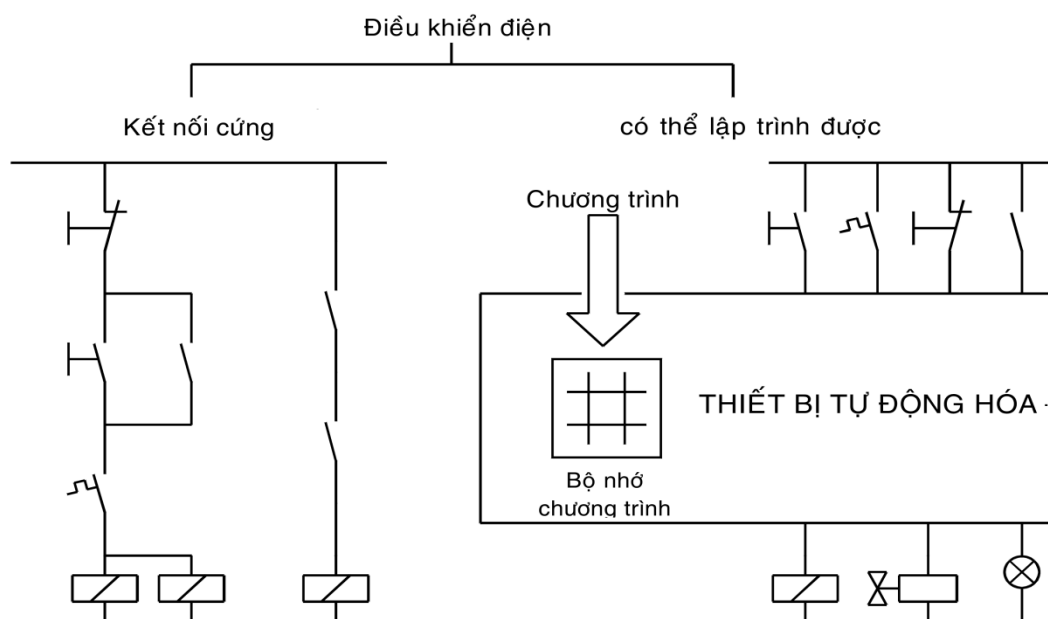
1.1. Điều khiển nối cứng và điều khiển lập trình.

1.1.1. Điều khiển kết nối cứng

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

1.1.2. Điều khiển logic khả trình (PLC)

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



Hình 1.5: Điều khiển kết nối cứng và điều khiển logic khả trình

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

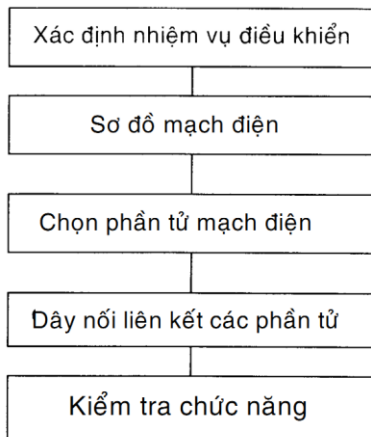
Trong công nghiệp, yêu cầu tự động hóa ngày càng tăng, đòi hỏi kỹ thuật điều khiển phải đáp ứng được các yêu cầu đó. Trong những năm gần đây, bên cạnh việc điều khiển bằng relais và khởi động từ thì việc điều khiển có thể lập trình được càng phát triển với hệ thống đóng mạch điện tử và thực hiện lập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

Trong nhiều lĩnh vực, các loại điều khiển cũ đã được thay đổi bởi điều khiển có thể lập trình được, có thể gọi là điều khiển logic khả trình. Viết tắt trong tiếng Anh là PLC (Programmable Logic Controller), tiếng Đức là SPS (Speicherprogrammierbare Steuerung). Sự khác biệt cơ bản giữa điều khiển logic lập trình (thay đổi được qui trình hoạt động) và điều khiển theo kết nối cứng (không thay đổi được qui trình hoạt động) là: Sự kết nối dây không còn nữa, thay vào đó là chương trình.

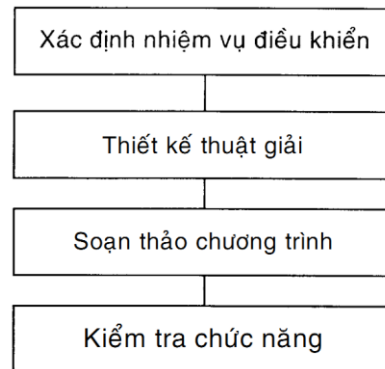
Có thể lập trình cho PLC nhờ vào các ngôn ngữ lập trình đơn giản. Đặc biệt đối với người sử

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

Các bước thiết lập hệ điều khiển bằng Rơle điện



Các bước thiết lập hệ điều khiển theo lập trình có nhớ (PLC)

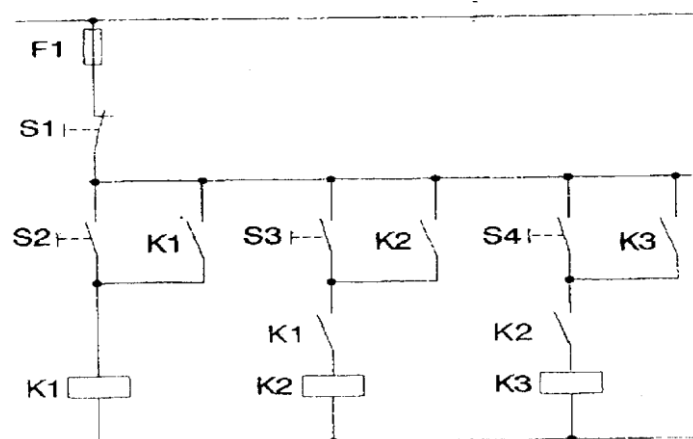


Khi thay đổi nhiệm vụ điều khiển thì người ta thay đổi mạch điều khiển: Lắp lại mạch, thay đổi các phần tử mới ở hệ điều khiển bằng relais điện. Trong khi đó khi thay đổi nhiệm vụ điều khiển ở hệ điều khiển logic khả trình (PLC) thì người ta chỉ thay đổi chương trình soạn thảo.

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

Điều khiển hệ thống của 3 máy bơm qua 3 khởi động từ K1, K2, K3. Trình tự điều khiển như sau: Các khởi động từ chỉ được phép thực hiện tuần tự, nghĩa là K1 đóng trước, tiếp theo K2 đóng và cuối cùng K3 mới đóng.

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



Error! Bookmark not defined. Hình 1.1: Mạch điều khiển tuần tự 3 máy bơm

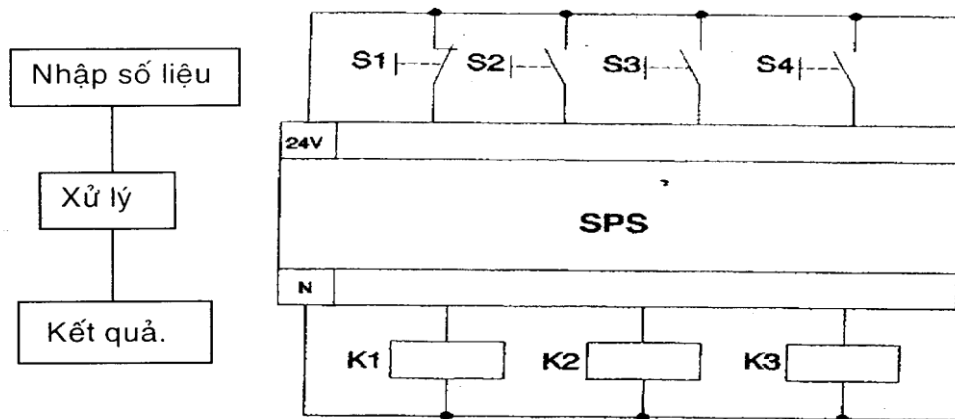
Khởi động từ K2 sẽ đóng khi công tắc S3 đóng với điều kiện là khởi động từ K1 đã đóng trước đó. Phương thức điều khiển như vậy được gọi là điều khiển tuần tự. Tiến trình điều khiển này được thực hiện một cách cưỡng bức.

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

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

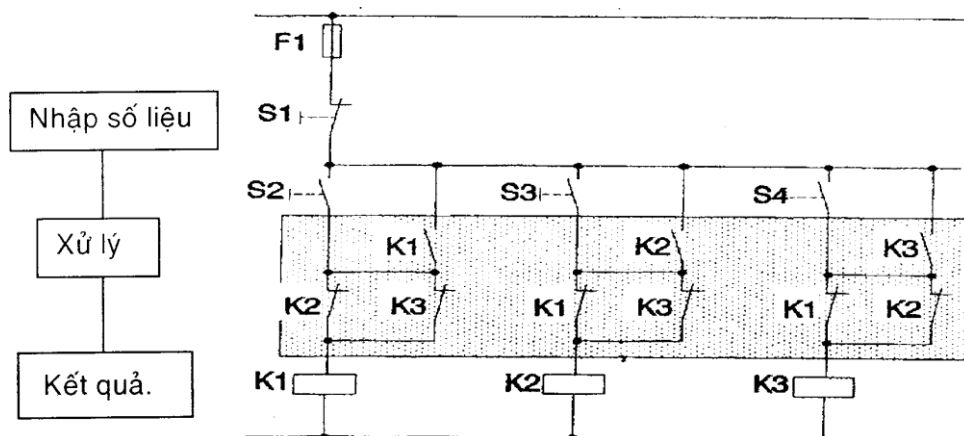
- Phần tử vào: Các nút nhấn S1, S2, S3, S4 vẫn giữ nguyên.
- Phần tử ra: Ba khởi động từ K1, K2, K3, để đóng và mở ba máy bơm vẫn giữ nguyên.
- Phần tử xử lý: được thay thế bằng PLC.

Sơ đồ kết nối với PLC được cho như ở hình 1.2. Tuần tự đóng mở theo yêu cầu đề ra sẽ được lập trình, chương trình sẽ được nạp vào bộ nhớ.



Hình 1.2: Sơ đồ kết nối với PLC

Bây giờ giả thiết rằng nhiệm vụ điều khiển sẽ thay đổi. Hệ thống ba máy bơm vẫn giữ nguyên, nhưng trình tự được thực hiện như sau: chỉ đóng được hai trong ba máy bơm hoặc mỗi máy bơm có thể hoạt động một cách độc lập. Như vậy theo yêu cầu mới đối với hệ thống điều khiển bằng rơ le điện phải thiết kế lại mạch điều khiển, sơ đồ lắp ráp phải thực hiện lại hoàn toàn mới. Sơ đồ mạch điều khiển biểu diễn như hình 1.3.



Hình 1.3: Sơ đồ mạch điều khiển 3 động cơ đã được thay đổi.

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

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

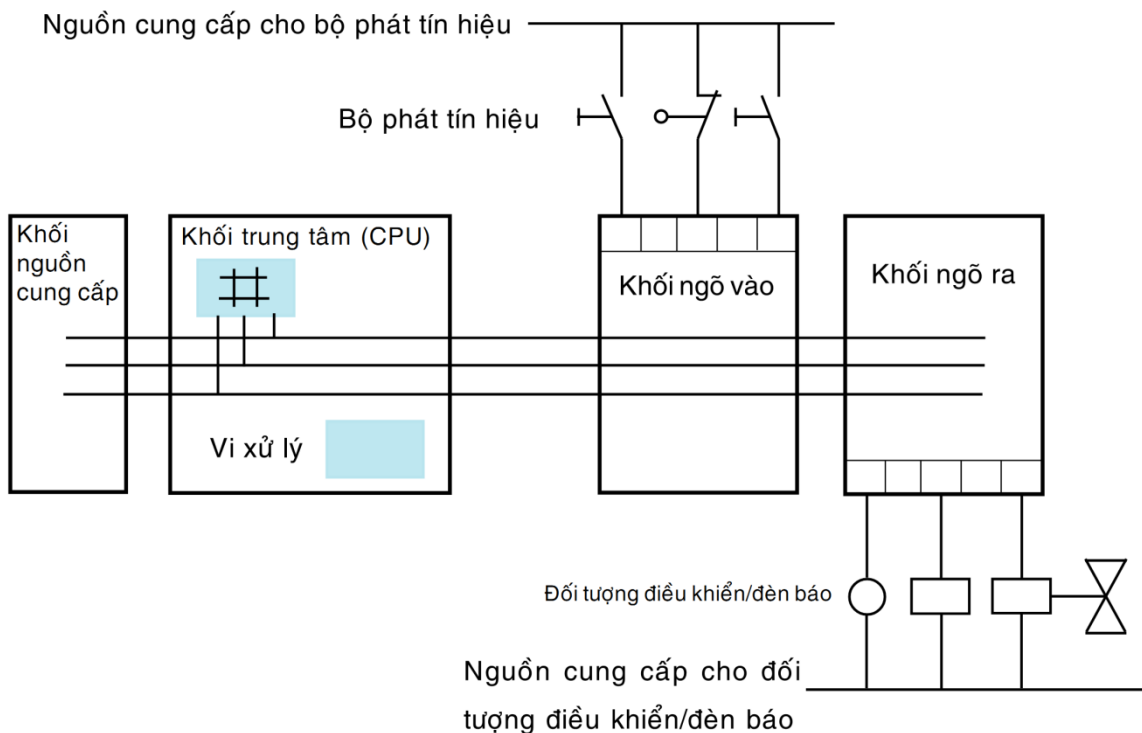
Hệ điều khiển lập trình có nhớ (PLC) có những ưu điểm sau:

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

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

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

2. Cấu trúc của một PLC.



Hình 2.1: Các khối trong một PLC

- Khối nguồn nuôi: nguồn trong các PLC thường là 24VDC.
- Module CPU: (cũng có bộ PLC sử dụng nguồn 220VAC. Những PLC không có module nguồn thì được cấp nguồn bên ngoài CPU: central processing unit: đơn vị xử lý trung tâm) bao gồm: bộ vi xử lý và bộ nhớ.
- Module xuất nhập (I/O module).
 - + Module nhập (input module) được nối với các công tắc, nút ấn, các bộ sensor ... để điều khiển từ chương trình bên ngoài.
 - + Module xuất (output module) được nối với các tải ở ngõ ra như cuộn dây của relay, contactor, đèn tín hiệu, các bộ ghép quang ...
- Hệ thống bus truyền tín hiệu: hệ thống bus truyền tín hiệu gồm nhiều đường tín hiệu song song:
 - Tuyến địa chỉ (address bus): chọn địa chỉ trên các khối khác nhau.
 - Tuyến dữ liệu (data bus): mang dữ liệu từ khối này đến khối khác.
 - Tuyến điều khiển (control bus): chuyển, truyền các tín hiệu định thì và điều khiển để đồng bộ các hoạt động trong PLC .

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

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

3.1. Địa chỉ ngõ vào/ ra.

_ Địa chỉ ngõ vào:

- + Ngõ vào số: I – Input image register.
- + Ngõ vào tương tự: AIW – Analog Input (Word).

_ Địa chỉ ngõ ra:

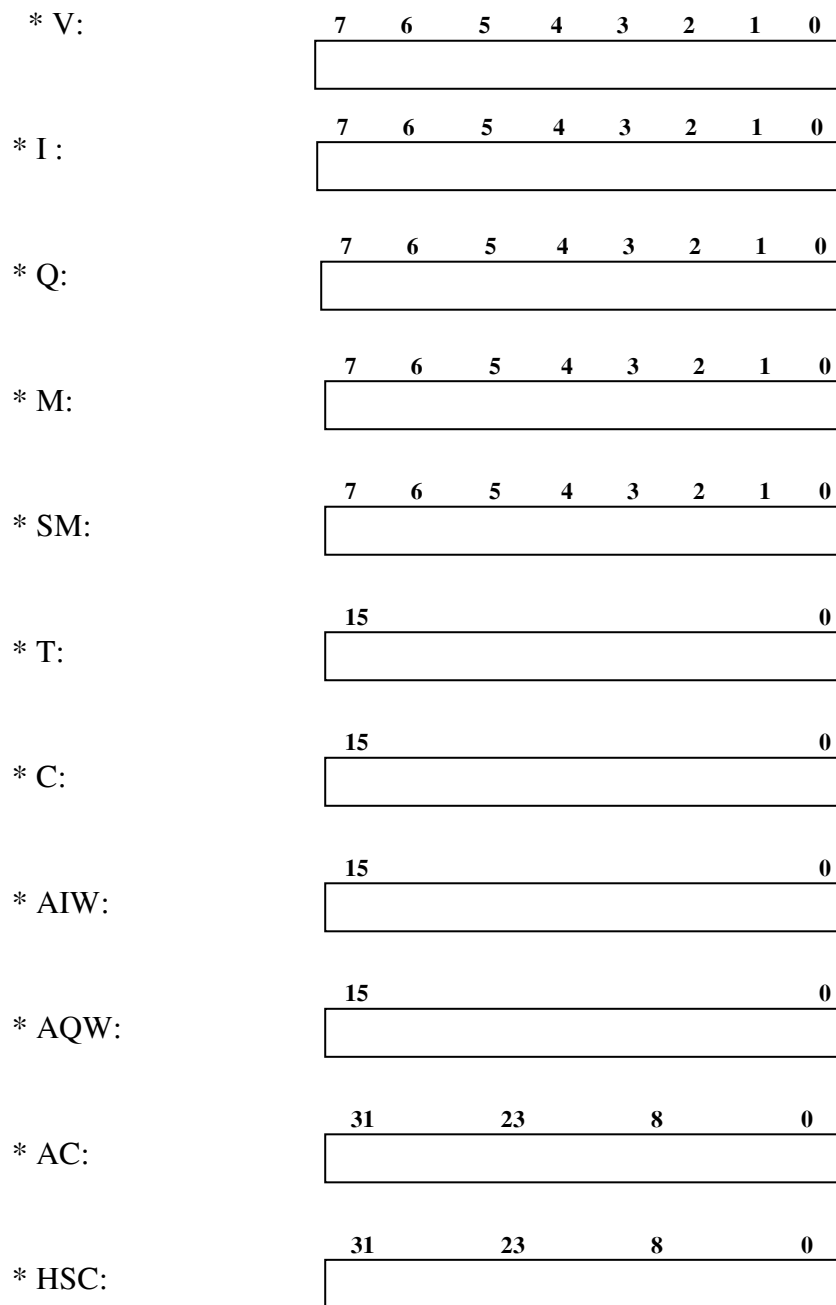
- + Ngõ ra số: Q– Output image register.
- + Ngõ ra tương tự: AQW – Analog Output (Word).

3.2. Phần chữ chỉ địa chỉ và kích thước của ô nhớ.

_ Phần chữ chỉ địa chỉ:

- * V- Variable memory: Vùng nhớ biến.
- * I – Input image register: vùng đệm cổng vào (I) (đọc/ ghi).
- * Q – Output image register: vùng đệm cổng ra (Q) (đọc/ ghi).
- * M – Internal memory bits: Vùng nhớ nội (M) (đọc/ ghi).
- * SM – Special memory bits: vùng nhớ đặc biệt.
- * T – Timer: điều khiển thời gian (đọc/ ghi).
- * C – Counter: bộ đếm (đọc/ ghi).
- * AIW – Analog Input (word): bộ đệm cổng vào tương tự (chỉ đọc).
- * AQW – Analog Output (word): bộ đệm cổng ra tương tự (chỉ ghi).
- * AC – Accumulator: thanh ghi (đọc/ghi).
- * HSC – (): bộ đếm tốc độ cao (đọc/ghi).

_ Kích thước của ô nhớ:



3.3. Phần số chỉ địa chỉ byte hoặc bit trong miền nhớ đã xác định.

_ Truy nhập theo bit:

Tên miền (+) địa chỉ byte (+) • (+) chỉ số bit.

Ví dụ: **I1.0** – Tên miền I, thứ tự byte 1, chỉ số bit 0.

_ Truy nhập theo byte:

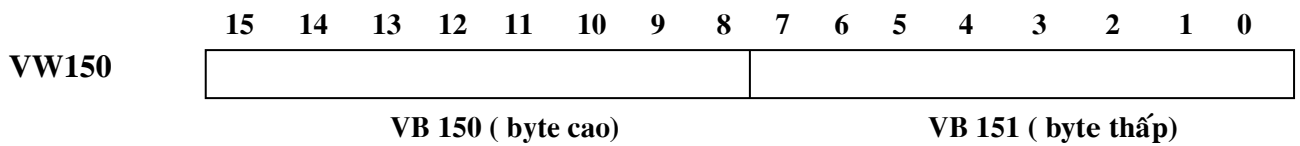
Tên miền (+) B (+) địa chỉ của byte trong miền.

Ví dụ: **VB150** – Tên miền V, B, thứ tự byte 150.

_ Truy nhập theo từ đơn (Word) , (1 word = 2 byte):

Tên miền (+) W (+) địa chỉ byte cao của từ trong miền.

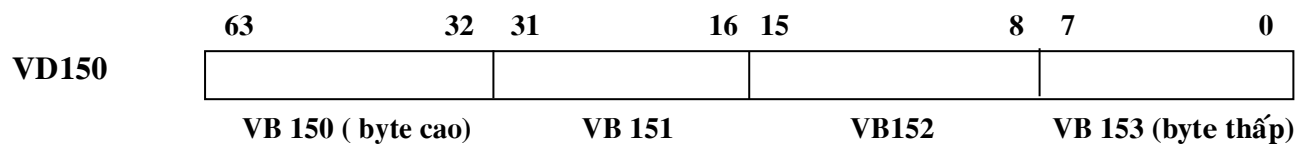
Ví dụ: VW150 – chỉ từ đơn gồm 2 byte 150 và 151 thuộc miền V, trong đó byte 150 có vai trò là byte cao trong từ.



_ Truy nhập theo từ kép(Double Word) , (1 double word = 2 word):

Tên miền (+) D (+) địa chỉ byte cao trong miền.

Ví dụ: VD150 – chỉ từ kép gồm 4 byte 150, 151, 152 và 153 thuộc miền V, trong đó byte 150 có vai trò là byte cao và byte 153 là byte thấp trong từ kép.



3.4. Cấu trúc của một bộ nhớ S7 – 200.

Bộ nhớ của S7 – 200 được chia thành 4 vùng:

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

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

_ **Vùng dữ liệu:** được sử dụng để cất các dữ liệu của chương trình bao gồm các kết quả, các phép tính, hằng số được định nghĩa trong chương trình, bộ đệm truyền thông...Một phần của bộ nhớ này thuộc kiểu non-volatile đọc/ ghi được.

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

o V – Variable memory:

o I – Input image register: vùng đệm cổng vào (I) (đọc/ ghi).

o Q – Output image register: vùng đệm cổng ra (Q) (đọc/ ghi).

o M – Internal memory bits: Vùng nhớ nội (M) (đọc/ ghi).

o SM – Special memory bits: vùng nhớ đặc biệt.

_ **Vùng đối tượng:** Timer (điều khiển thời gian), counter (bộ đếm), bộ đếm tốc độ cao và các cổng vào ra tương tự được đặt trong vùng nhớ cuối cùng. Vùng này không thuộc kiểu non-volatile nhưng đọc/ ghi được.

o T – Timer: điều khiển thời gian (đọc/ ghi).

o C – Counter: bộ đếm (đọc/ ghi).

o AIW – Analog Input (word): bộ đệm cổng vào tương tự (chỉ đọc).

o AQW – Analog Output (word): bộ đệm cổng ra tương tự (chỉ ghi).

o AC – Accumulator: thanh ghi (đọc/ghi).

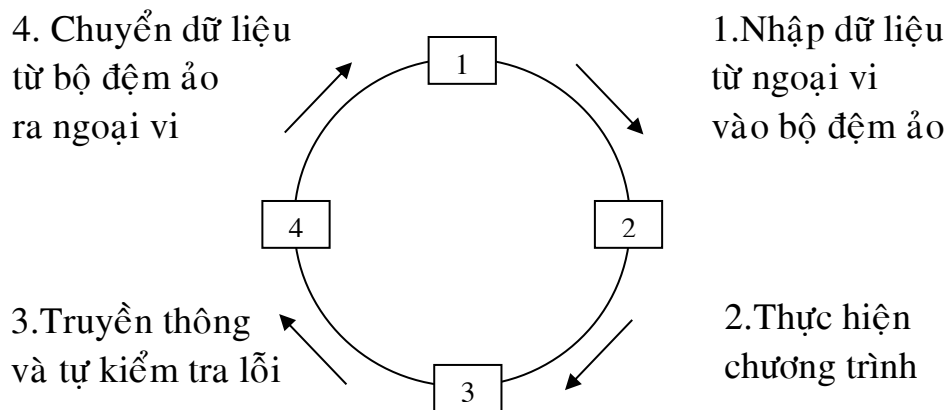
o HSC – (): bộ đếm tốc độ cao (đọc/ghi).

4. Xử lý chương trình

4.1. Vòng quét chương trình

PLC thực hiện chương trình theo chu kỳ lặp. Mỗi vòng lặp được gọi là vòng quét (scan). Như hình (H.6)

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



(H.6)

4.2. Cấu trúc chương trình S7 – 200:

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

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

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

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

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

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

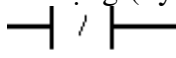
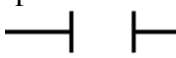
Main Program . . END	Thực hiện trong một vòng quét
SBR 0 Chương trình con thứ nhất . . RET	Thực hiện khi được chương trình chính gọi
SBR n Chương trình con thứ n+1 . . RET	
INT 0 Chương trình xử lý ngắt thứ nhất . . RETI	Thực hiện khi có tín hiệu báo ngắt
INT n Chương trình xử lý ngắt thứ n+1 . . RETI	

4.3. Phương pháp lập trình:


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

4.3.1. Phương pháp hình thang (LAD):

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

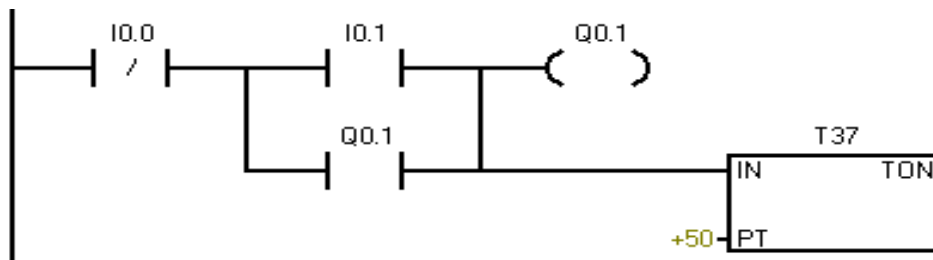
+ Tiếp điểm: Là biểu tượng (Symbol) mô tả các tiếp điểm của relay. các tiếp điểm có thể là thường đóng  hoặc thường mở .

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

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

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

Ví dụ:



Hình 1.3.1: Mô tả mạng LAD và các phần tử

4.3.2. Phương pháp lập trình FBD:

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

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

4.3.3. Phương pháp lập trình STL:

STL là ngôn ngữ lập trình dưới dạng tập hợp các câu lệnh. Mỗi câu lệnh trong chương trình, kể cả những lệnh hình thức biểu diễn một chức năng của PLC.

Ví dụ:

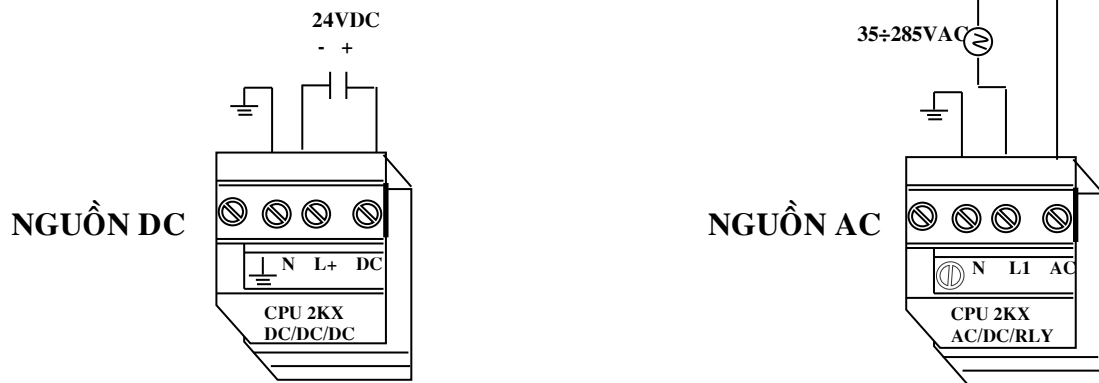
LAD	STL
	<pre>LD I0.1 A I0.2 O I0.0 = Q0.1</pre>

Tên lệnh	Mô tả
LD n	Nạp giá trị logic của điểm n chỉ dẫn trong lệnh vào bit đầu tiên của ngăn xếp.
A n	Giá trị bit đầu tiên được thực hiện với bằng phép tính AND với điểm n chỉ dẫn trong lệnh. Kết quả được ghi lại vào bit đầu tiên của ngăn xếp.
O n	Thực hiện toán tử OR giữa bit đầu tiên của ngăn xếp với điểm n chỉ dẫn trong lệnh. Kết quả được ghi lại vào bit đầu trong ngăn xếp.
= n	Giá trị của bit đầu tiên trong ngăn xếp được sao chép sang điểm n được chỉ dẫn trong lệnh.

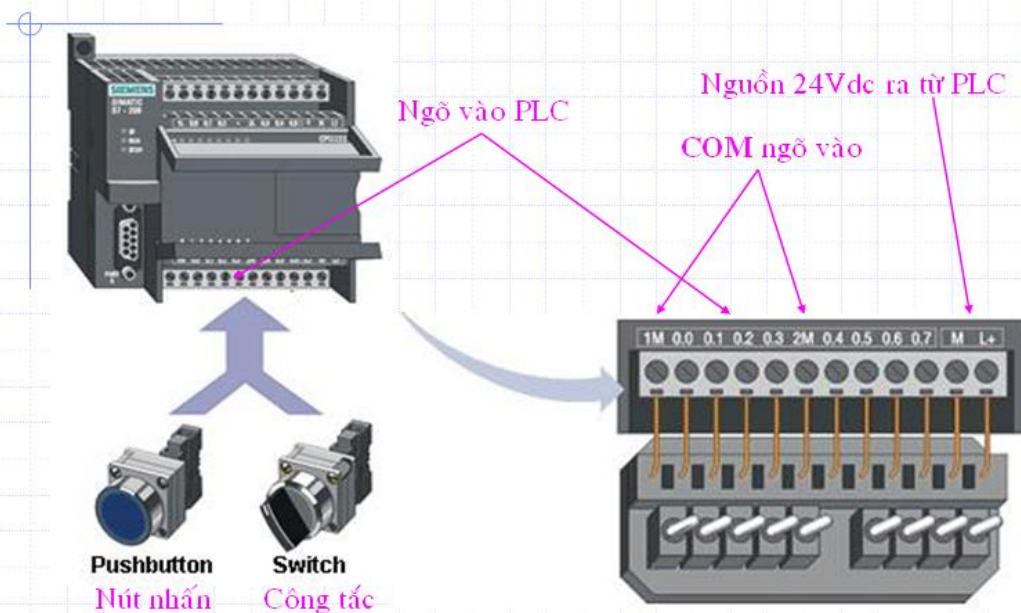
5. Kết nối PLC với các thiết bị ngoại vi:

5.1. Cấp nguồn:

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

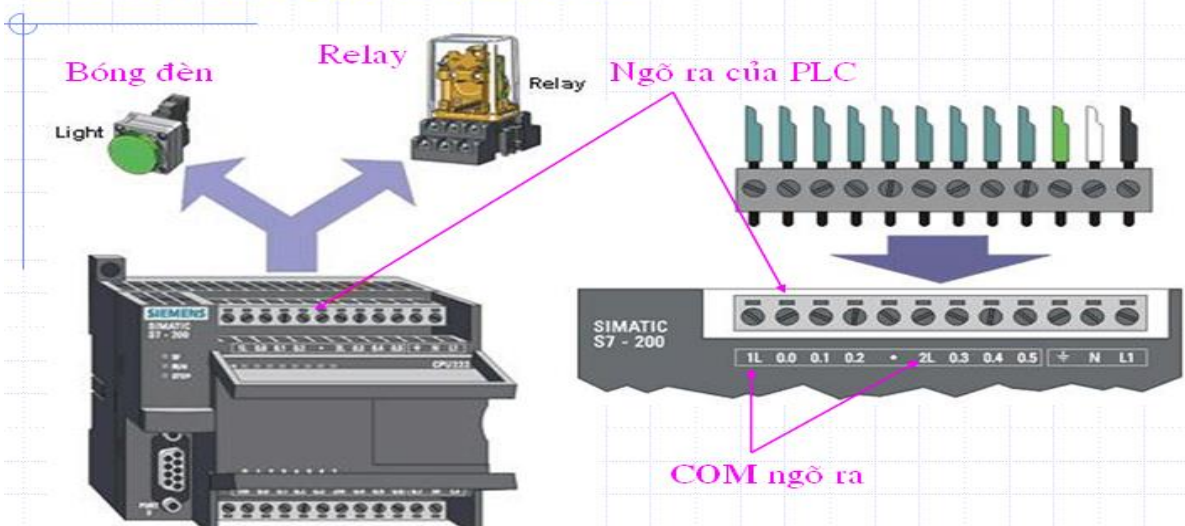


Kết nối tín hiệu ngõ vào cho PLC



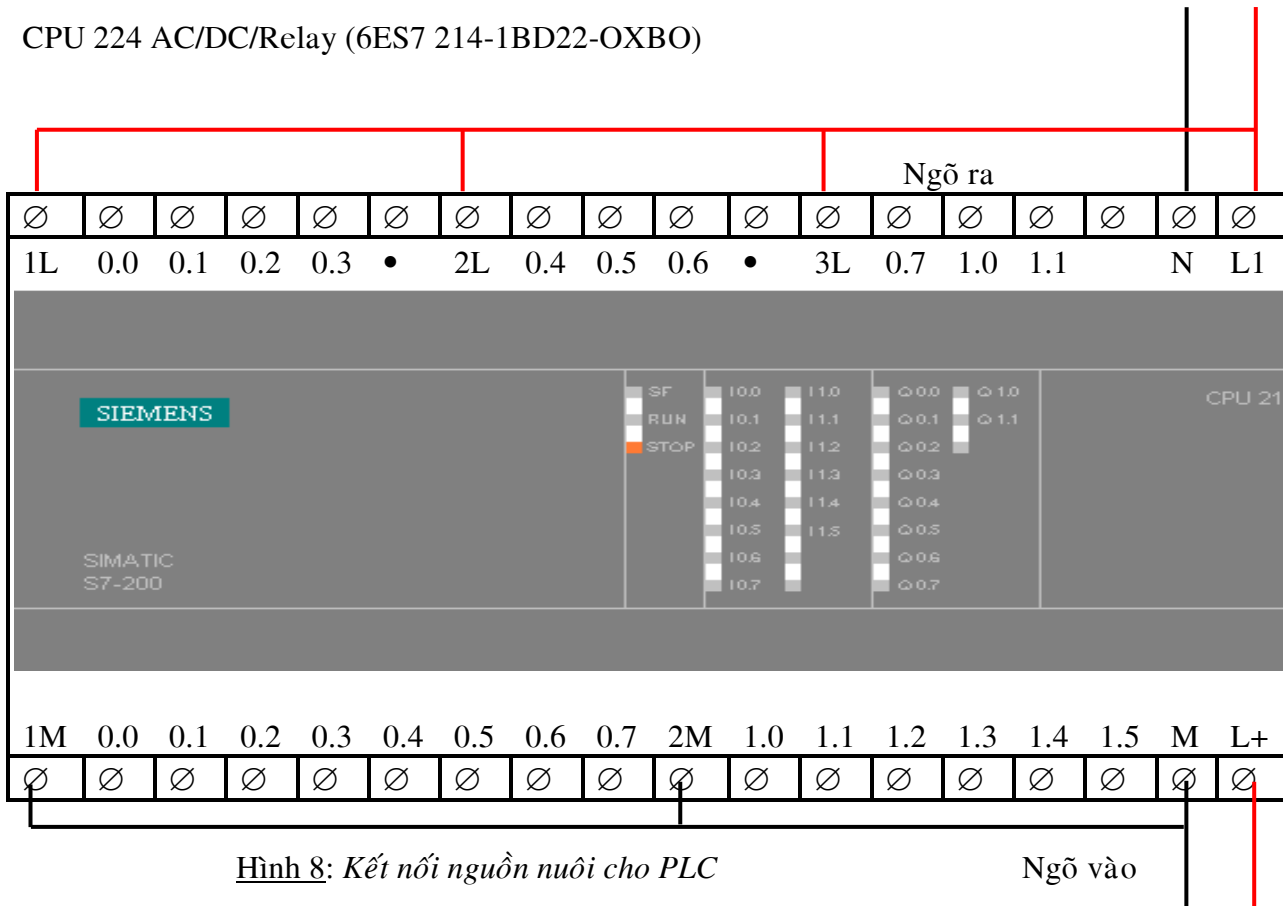
(H.7a)

Kết nối tín hiệu ngõ ra của PLC



(H.7b)

CPU 224 AC/DC/Relay (6ES7 214-1BD22-0XBO)



Hình 8: Kết nối nguồn nuôi cho PLC

Ngõ vào

Ngõ vào cấp nguồn 24 VDC

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

Nguồn dương (+) nối vào chân L+.

Ngõ ra cấp nguồn 240 VAC

Dây trung tính nối vào chân N

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

Dây bảo vệ PE nối vào chân

5.2. Kết nối thiết bị ngoại vi:

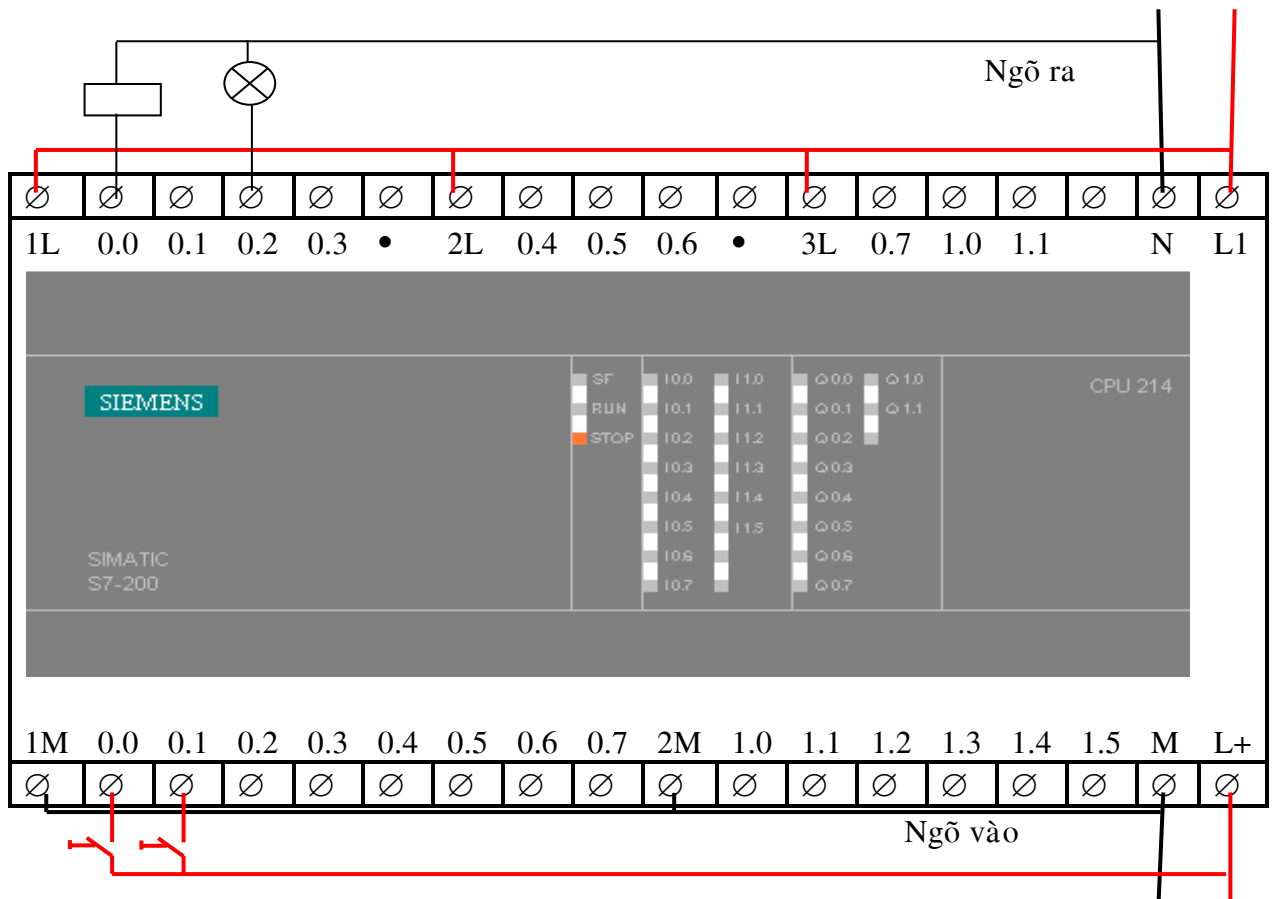
Kết nối thiết bị ngoại vi là kết nối giữa PLC với các thiết bị ngõ vào và thiết bị ngõ ra.

- Kết nối thiết bị ngõ vào:

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

- Kết nối thiết bị ngõ ra:

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



Hình 9: Kết nối ngõ vào/ ngõ ra cho PLC

5.3. Kết nối CPU đến thiết bị lập trình: (H.10)

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

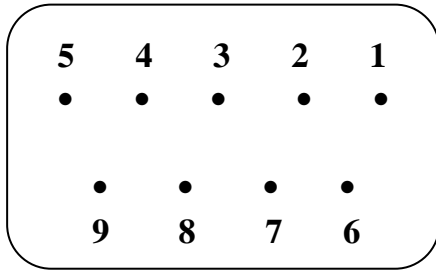
- Kết nối đầu RS232(được ký hiệu là PC) của cáp RS232/PPI Multi – Master Cable đến thiết bị lập trình.
- Kết nối đầu RS485(được ký hiệu là PPI) của cáp RS232/PPI Multi – Master Cable đến S7 – 200.
- Kiểm tra những SWITCHS chọn chế độ phải đúng.



(H.10)

*** Cổng truyền thông:**

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



Hình 1.5.1: Sơ đồ chân của cổng truyền thông

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

Để ghép nối S7 – 200 với máy lập trình PG702 hoặc với các loại máy lập trình thuộc họ PG7xx có thể sử dụng một cáp nối thẳng qua MPI. Cáp đó đi kèm theo máy lập trình.

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

*** Mô tả đèn báo trạng thái:**

Mô tả các đèn báo trên S7 – 200, CPU 214:

- SF Đèn đỏ SF báo hiệu hệ thống bị hỏng. Đèn SF sáng lên khi PLC bị hỏng.
(đèn đỏ)
- RUN Đèn xanh RUN chỉ định PLC đang ở chế độ làm việc và thực hiện chương trình được nạp vào trong máy.
(đèn xanh)
- STOP Đèn vàng STOP chỉ định rằng PLC đang ở chế độ dừng. Dừng chương trình đang thực hiện lại.
(đèn vàng)
- Ix.x Đèn xanh ở cổng vào chỉ định trạng thái tức thời của cổng Ix.x (x.x=0.0÷1.5)
(đèn xanh)
- Qy.y Đèn xanh ở cổng ra báo hiệu trạng thái tức thời của cổng Qy.y (y.y=0.0÷1.1)
(đèn xanh)

*** Công tắc chọn chế độ làm việc cho PLC:**

Công tắc chọn chế độ làm việc cho nằm phía trên, bên cạnh các cổng ra của S7 – 200 có ba vị trí cho phép chọn các chế độ làm việc khác nhau cho PLC.

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

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

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

*** Chỉnh định tương tự:** Điều chỉnh tương tự (1bộ trong CPU 212 và 2 bộ trong CPU 214) cho phép điều chỉnh các biến cần thay đổi và sử dụng trong chương trình. Núm điều chỉnh analog được lắp đặt dưới nắp đậy bên cạnh các cổng ra. Thiết bị chỉnh định có thể quay 270 độ.

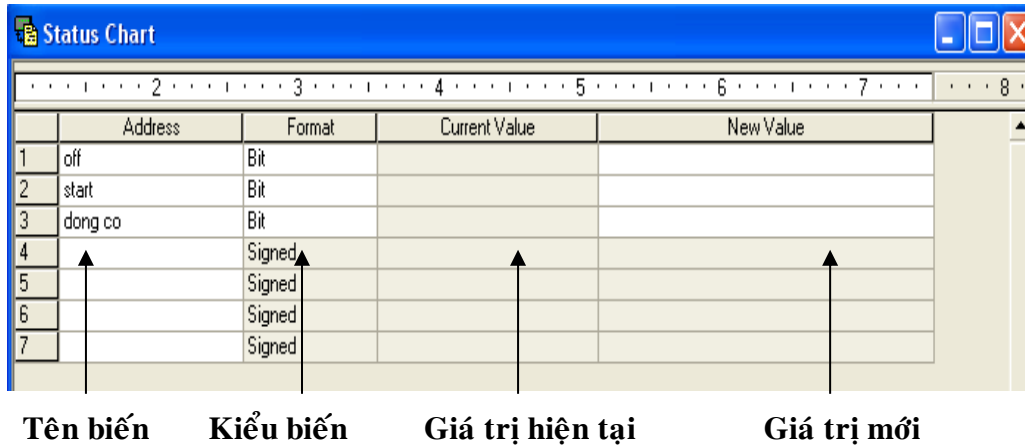
*** Pin và nguồn nuôi bộ nhớ:**

Nguồn nuôi dùng để ghi chương trình hoặc nạp một chương trình mới.

Nguồn pin có thể được sử dụng để mở rộng thời gian lưu giữ cho các dữ liệu có trong bộ nhớ. Nguồn pin tự động được chuyển sang trạng thái tích cực nếu như dung lượng tụ nhớ bị cạn kiệt và nó phải thay thế vào vị trí đó để dữ liệu trong bộ nhớ không bị mất đi.

6. Kiểm tra việc nối dây bằng phần mềm.

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



(H.11)

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

7.1. Những yêu cầu đối với máy tính PC:

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

- 640 Kb RAM (ít nhất phải có 500 Kb bộ nhớ còn trống)
- Màn hình 24 dòng, 80 cột ở chế độ văn bản
- Còn khoảng 2Mb trống trong ổ đĩa cứng
- Có hệ điều hành MS-DOS ver 5.0 hoặc cao hơn
- Bộ chuyển đổi RS232 – RS485 phục vụ ghép nối truyền thông giữa PC và PLC

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

7.2.1. Cài đặt:

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



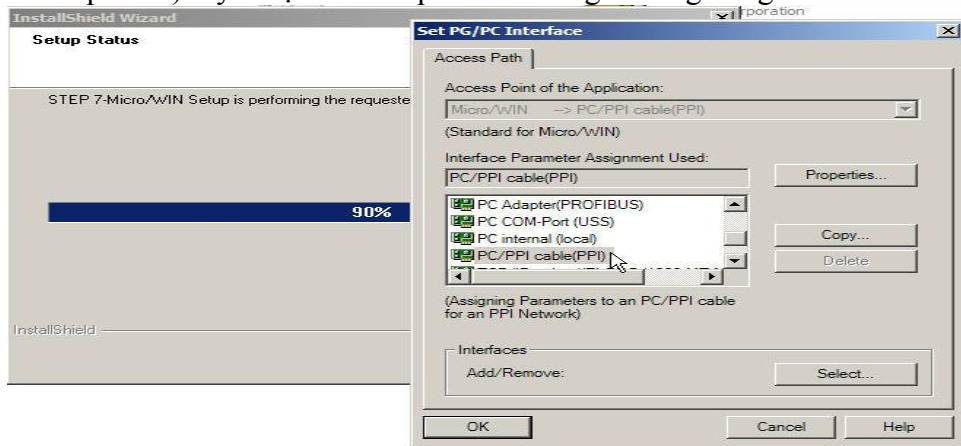
Hình 12: sau khi chạy file setup.exe



(Hình 13)

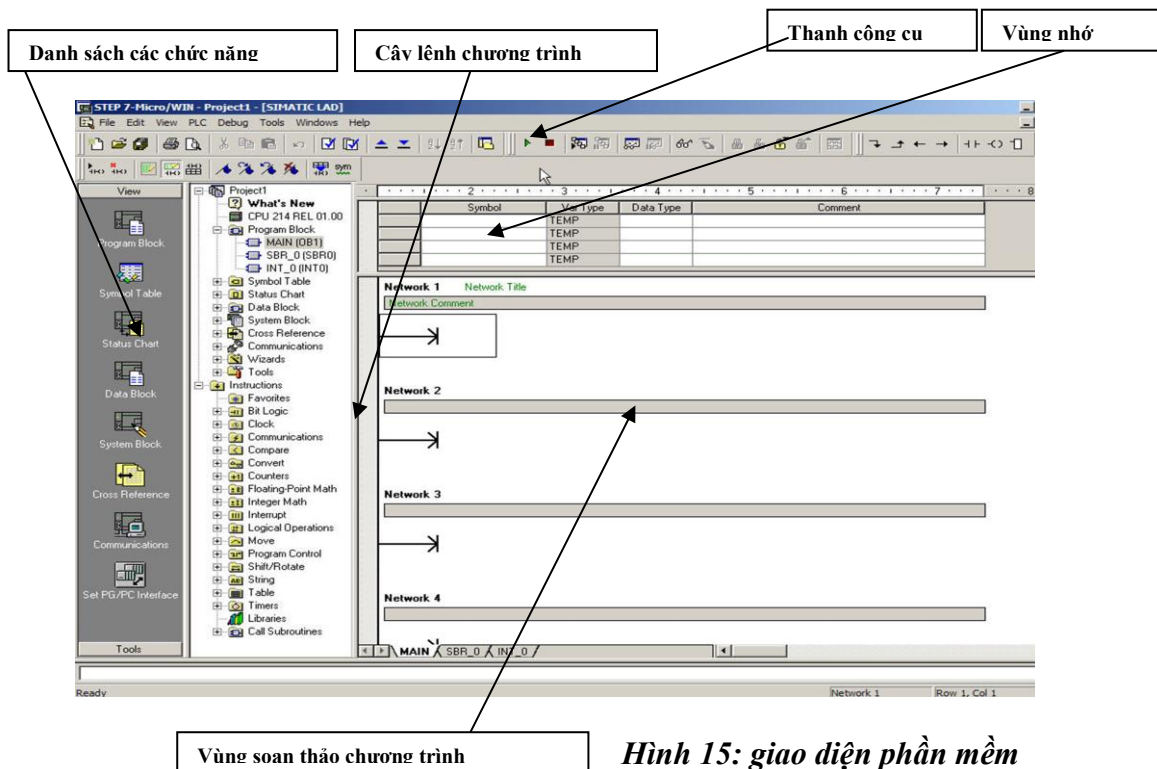
Chọn Next....

Sau đó chương trình sẽ tự động cài đặt các file cần thiết. Lưu ý khi tới phần chọn giao tiếp máy tính ta nhớ chọn giao tiếp là PC/PPI, sau đó có thể chọn cổng COM hoặc USB, tốc độ truyền...(Tab Properties) tùy thuộc vào adapter mà chúng ta đang dùng. Xem hình 14:



Hình 14: chọn cài đặt giao tiếp với máy tính.

7.2.2. Giao diện của MicroWin32:



Hình 15: giao diện phần mềm

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

7.2.3. Soạn thảo:

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

a. Khởi động chương trình:

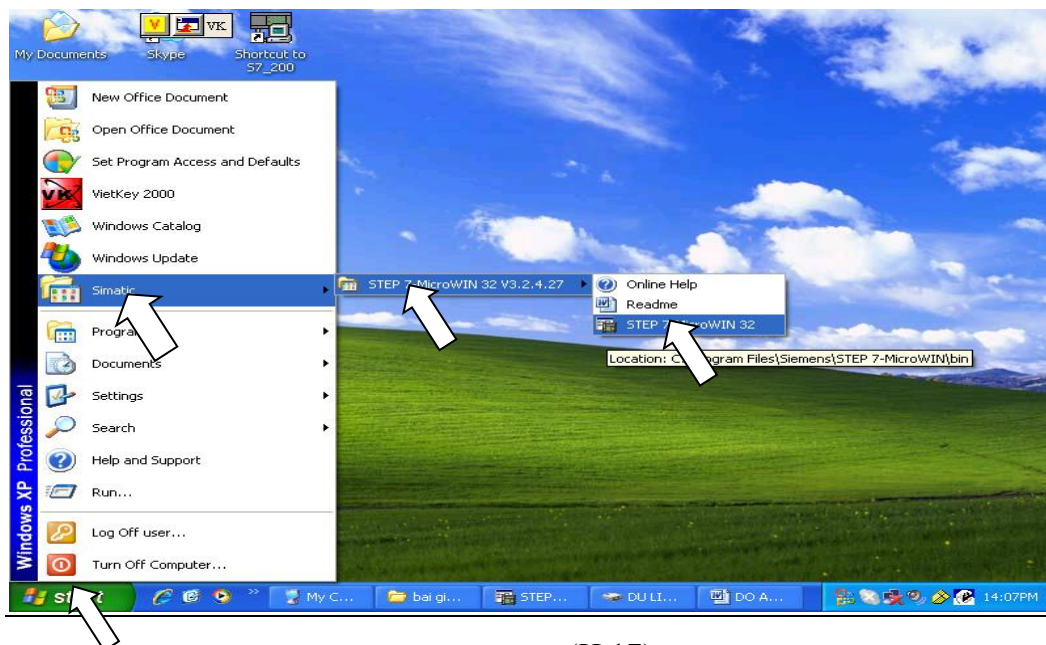
- Cách 1: Click chuột vào biểu tượng STEP 7 - MicroWin V3.2 hay V4.0 ở màn hình (Desktop) để bắt đầu soạn thảo như (H.16).



(H.16)

- Cách 2:

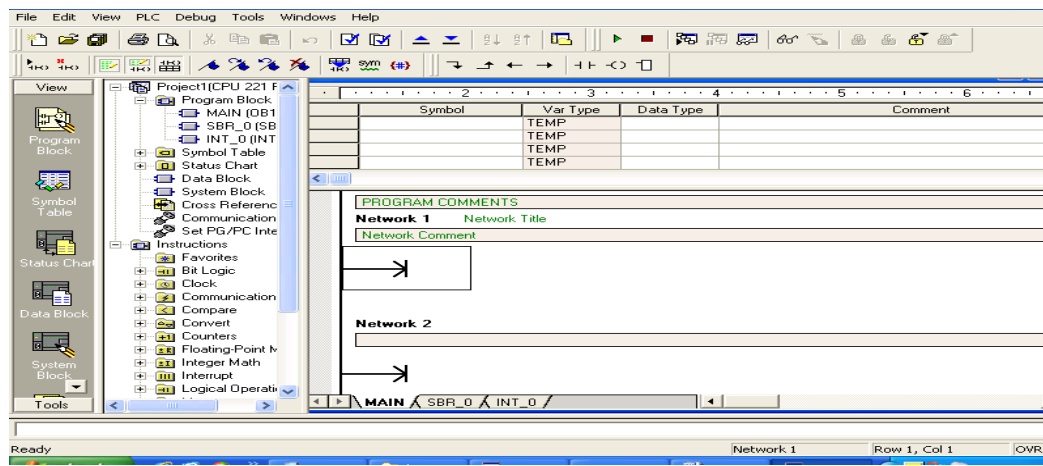
Nhấp Start » Simatic » STEP 7 - MicroWin V3.2 hay V4.0 như (H.17).



(H.17)

b. Soạn thảo chương trình:

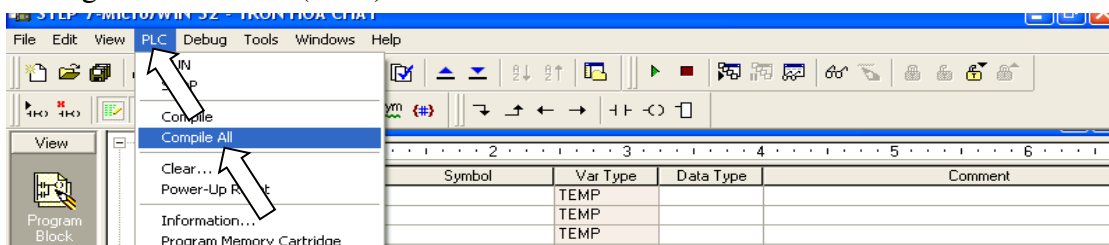
Khi khởi động màn hình soạn thảo sẽ xuất hiện như hình (H.18). Sau đó ta tiến hành soạn thảo chương trình theo yêu cầu (ngôn ngữ dạng LAD).



(H.18)

c. Kiểm tra lỗi:

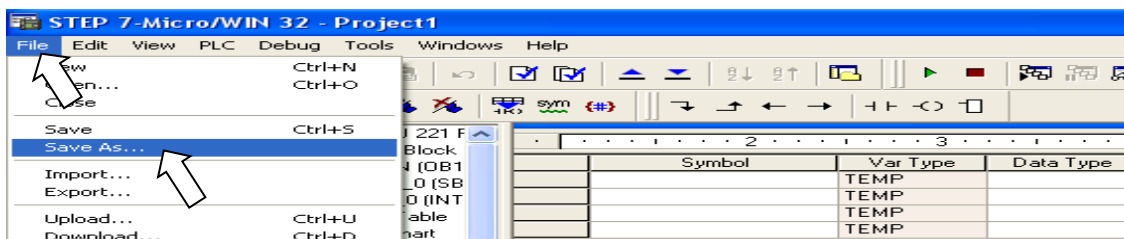
Sau khi soạn thảo chương trình xong ta nhấp vào PLC/ Compile All để kiểm tra toàn bộ chương trình như hình (H.19):



(H.19)

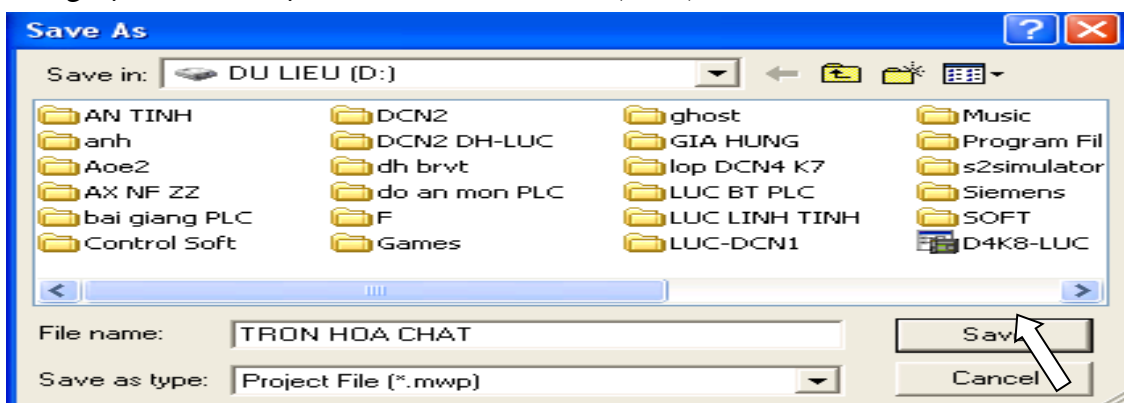
d. Lưu chương trình:

Sau khi soạn thảo chương trình xong và đã kiểm tra lỗi, chương trình đúng yêu cầu ta vào File/ Save As để lưu chương trình như hình (H.20) .



(H.21)

Trong hộp thoại Save As ta chọn tên ổ đĩa, tên File. Ví dụ tên File là TRỘN HOÁ CHẤT chẳng hạn, sau đó chọn Save để lưu như hình (H.22).



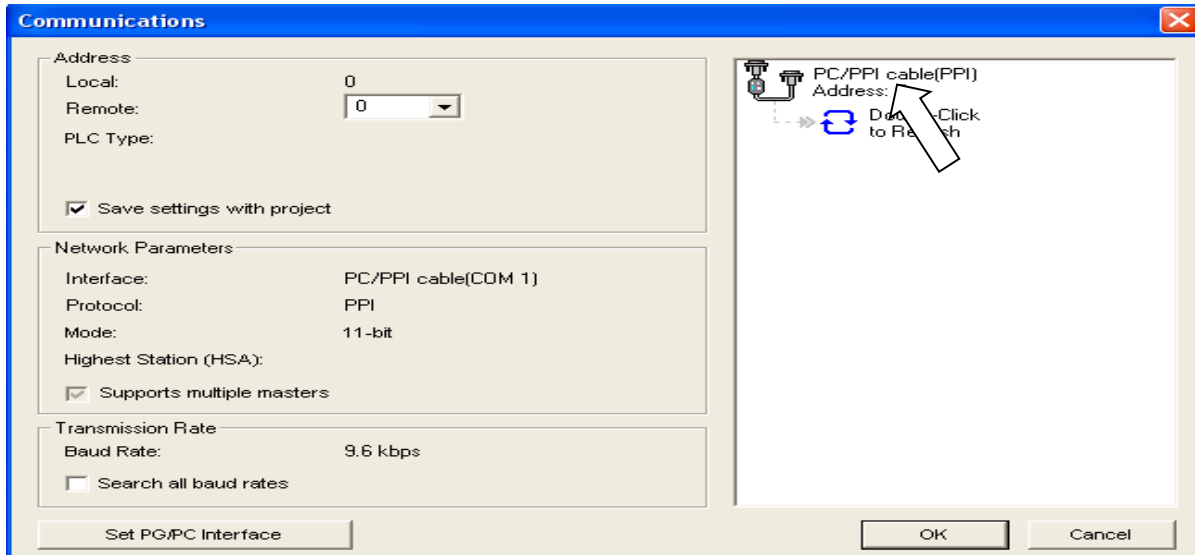
(H.22)

e. Thiết lập thông số cho hộp thoại Communications:

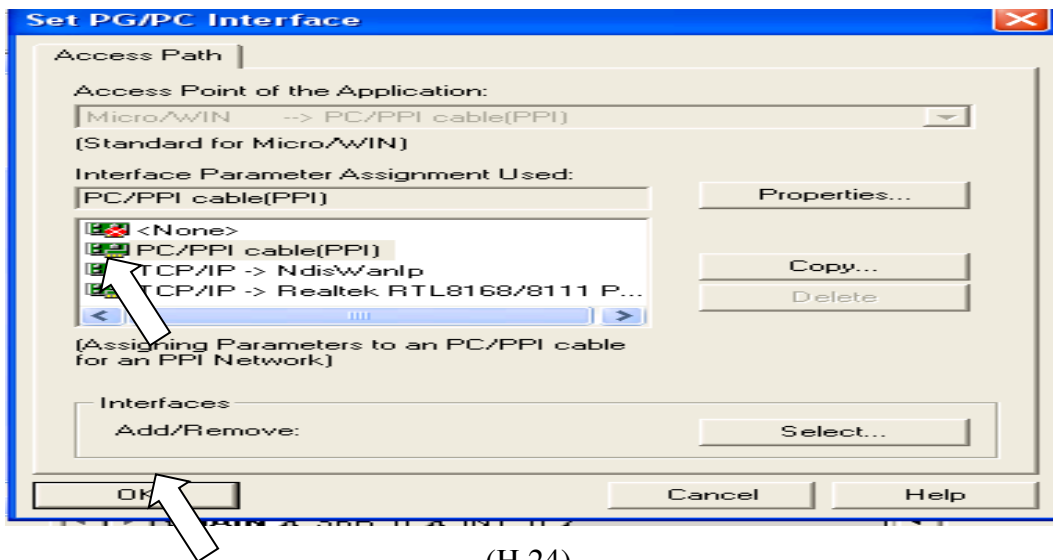


Click vào biểu tượng ở các khối chức năng bên trái màn hình soạn thảo để thiết lập giao tiếp cho Click – MicroWin. Khối chức năng sẽ hiện thị như hình (H.16):

* Click đúp vào PC/PPI cable(PPI), hoặc kích vào Set PG/PC Interface chọn PC/PPI cable(PPI) ⇒ OK như hình (H.23), (H.24)



(H.23)



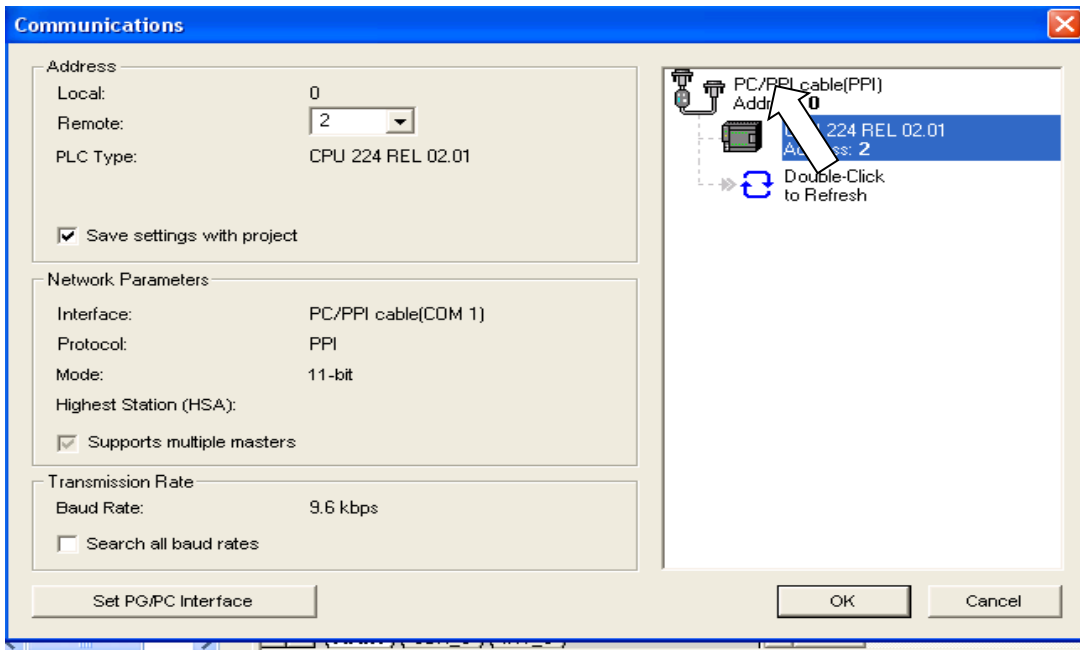
(H.24)

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

- Thiết lập địa chỉ của cáp PC/PPI Cable bằng 0
- Thiết lập cổng giao tiếp của cáp PC/PPI Cable là COM 1
- Thiết lập tốc độ truyền là 9.6 kbps

f. Thiết lập sự kết nối với S7 – 200:


Để thiết lập sự kết nối giữa thiết bị lập trình và S7 – 200 ta Click đúp vào Refresh trong hộp kết nối. STEP 7 sẽ tìm và hiển thị những CPU được kết nối. như hình (H.125), sau đó nhấn OK.

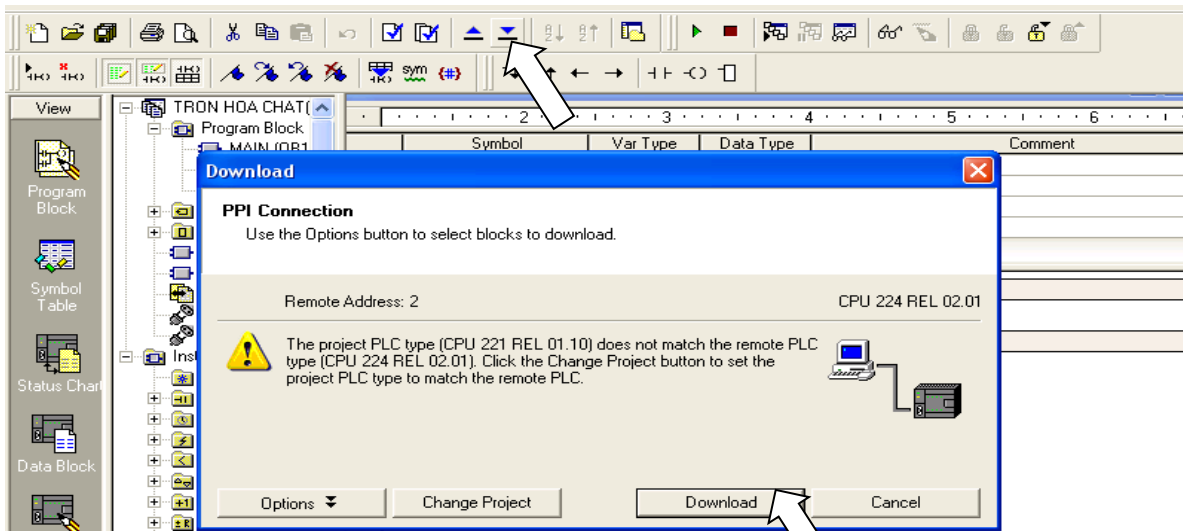


(H.15)

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


g. Download chương trình:

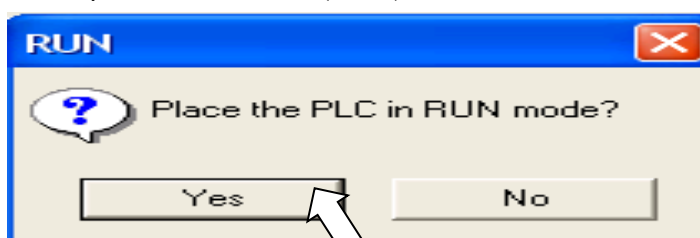
Để Download chương trình đến S7 – 200 ta Click vào biểu tượng  , màn hình sẽ hiện thị hộp thoại và ta chọn Download để load chương trình đến PLC, như hình (H.26):



(H.26)


h. Chạy chương trình:

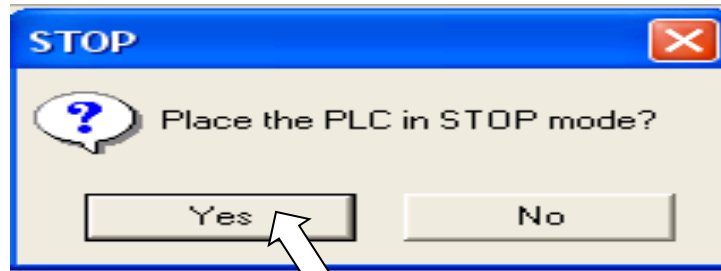
Khi đã Download chương trình thành công ta nhấp vào PLC⇒RUN hoặc biểu tượng , hộp thoại xuất hiện ta chọn Yes, như hình (H.27):



(H.27)

i. Dừng chương trình:

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



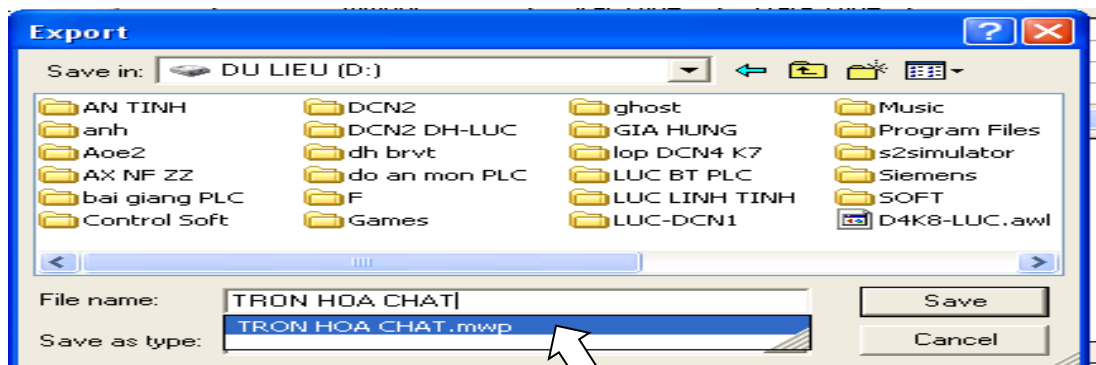
(H.28)

III.4 . Lập trình mô phỏng trên máy tính:

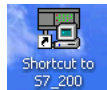
Chương trình mô phỏng S7-200 – Simulator dùng để người học thực tập khi không có PLC thực. Để sử dụng phần mềm S7-200 – Simulator trong máy tính ngoài cần phải có phần mềm lập trình MicroWin V3.2, hay V4.0 còn cần thêm phần mềm mô phỏng S7-200 – Simulator.

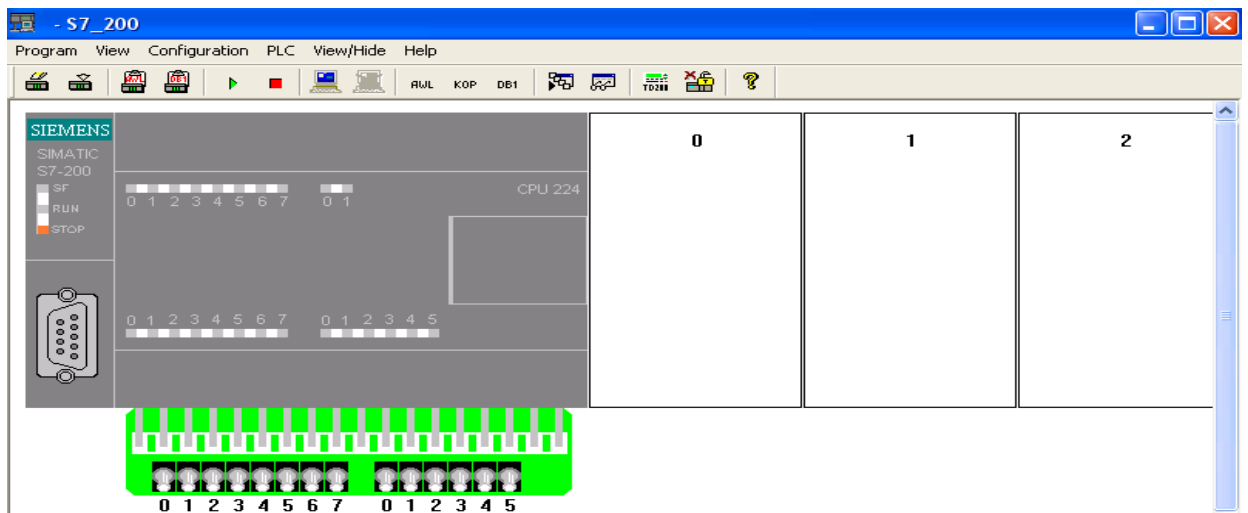
Quá trình mô phỏng được thực hiện theo trình tự sau:

- Soạn thảo chương trình trên phần mềm MicroWin V3.2, hay V4.0. sau đó vào File/ Export để lưu file vào ổ đĩa (file có đuôi .awl) như hình (H.29):




(H.29)

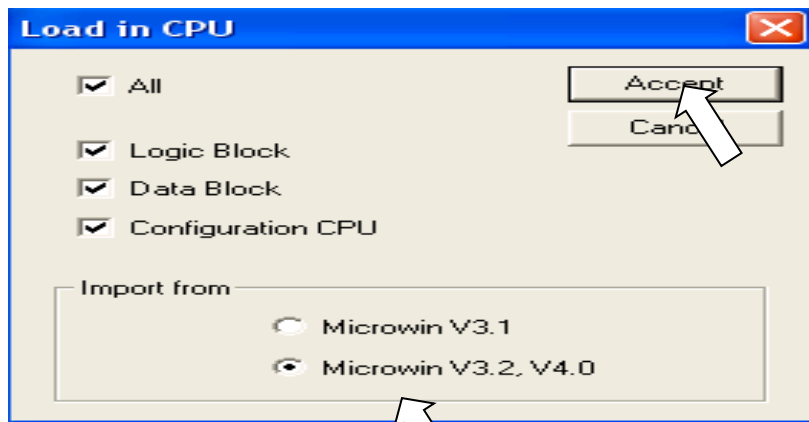
- Khởi động phần mềm mô phỏng bằng cách nhấp đúp vào biểu tượng .
- Giao diện mô phỏng sẽ xuất hiện như hình (H.30):



(H.30)

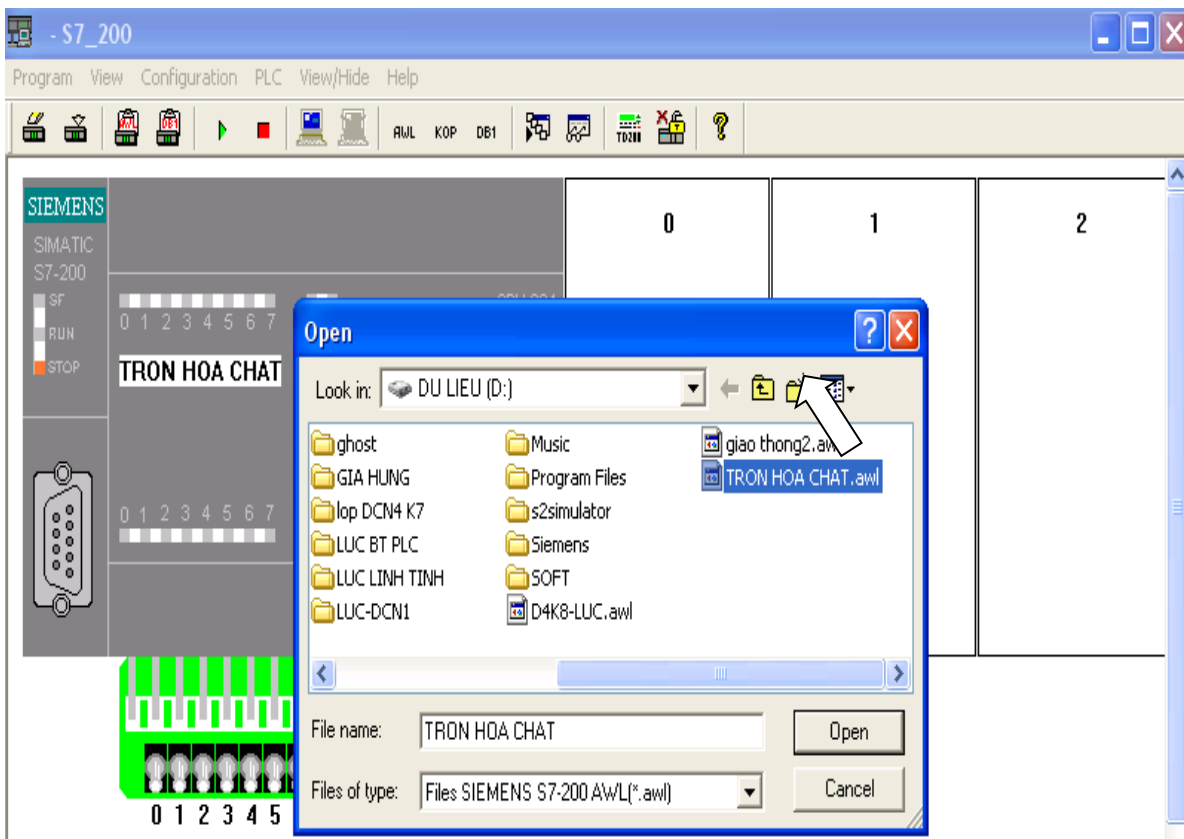
Trường hợp nếu xuất hiện hộp thoại thì ta nhập mã bảo vệ 6596, sau đó Click OK.

Khi màn hình mô phỏng sẽ xuất hiện ta nhấp vào biểu tượng , sau đó xuất hiện hộp thoại ta chọn MicroWin V3.2, V4.0 và nhấp vào Accept như hình (H.31) .



(H.31)

Hộp thoại yêu cầu nhập tên bài cần mô phỏng. Ta chọn tên bài mô phỏng và Click Open. Sau đó chọn RUN để bắt đầu mô phỏng như hình (H.32).



(H.32)

BÀI 2: CÁC PHÉP TOÁN NHỊ PHÂN CỦA PLC

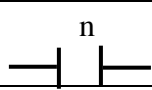
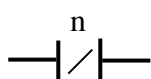
1. Các Liên Kết Logic

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

_ Các lệnh vào/ ra:

* **Lệnh vào** (ngõ vào):

Mô tả lệnh vào bằng LAD như sau

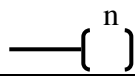
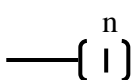
LAD	Mô tả	Toán hạng
 n	Tiếp điểm thường mở sẽ được đóng nếu n=1	n(bit): I, Q, M, SM, T, C, V
 n	Tiếp điểm thường đóng sẽ mở khi n=1	

Mô tả lệnh vào bằng STL như sau:

STL	Mô tả	Toán hạng
LD n	Lệnh nạp giá trị logic của điểm n vào bit đầu tiên trong ngăn xếp	n(bit): I, Q, M, SM, T, C, V
LDN n	Lệnh nạp giá trị logic nghịch đảo của điểm n vào bit đầu tiên của ngăn xếp.	

* **Lệnh ra** Output (ngõ ra):

Mô tả lệnh ra bằng LAD như sau:

LAD	Mô tả	Toán hạng
 n	Cuộn dây đầu ra ở trạng thái kích thích khi có dòng điều khiển đi qua.	n(bit): I, Q, M, SM, T, C, V
 n	Cuộn dây đầu ra được kích thích tức thời khi có dòng điều khiển đi qua.	

Mô tả lệnh ra bằng STL như sau:

LAD	Mô tả	Toán hạng
= n	Lệnh = sao chép giá trị của đỉnh ngăn xếp tới tiếp điểm n được chỉ dẫn trong lệnh	n(bit): I, Q, M, SM, T, C, V
=I n	Lệnh =I (immediate) sao chép tức thời giá trị của đỉnh stack tới tiếp điểm n được chỉ dẫn trong lệnh.	

_ Các lệnh tiếp điểm đặc biệt:

Có thể dùng các lệnh tiếp điểm đặt biệt để phát hiện sự chuyển tiếp trạng thái của xung (sườn xung) và đảo lại trạng thái của dòng cung cấp (giá trị của đỉnh ngăn xếp). LAD sử dụng các tiếp điểm đặt biệt để tác động vào dòng cung cấp. Các tiếp điểm đặt 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.

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

LAD	Mô tả	Toán hạng
	Tiếp điểm đảo trạng thái của dòng cung cấp. Nếu dòng cung cấp có tiếp điểm đảo thì nó được 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] xuống [0]	Không có

Ví dụ minh họa:

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

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

O nhớ	Mô tả
SM0.0	Luôn có giá trị logic bằng 1
SM0.1	Có giá trị logic bằng 1 ở vòng quét đầu tiên
SM0.2	Bit báo dữ liệu bị thất lạc (0- dữ liệu còn đủ; 1- dữ liệu bị thất lạc)
SM0.3	Bit báo PLC được đóng nguồn(1- ở vòng quét đầu tiên; 0- ở vòng quét tiếp theo)
SM0.4	Phát nhịp 60 giây (0- cho 30 giây đầu; 1- cho 30 giây sau)
SM0.5	Phát nhịp 1 giây (0- cho 0.5 giây đầu; 1- cho 0.5 giây sau)
SM0.6	Nhịp vòng quét (1- cho vòng quét luân phiên)
SM0.7	Bit chọn chế độ cho PLC (0- TERM; 1- RUN)

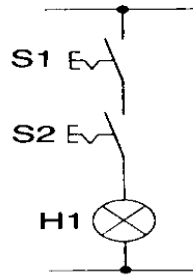
1.2. Các lệnh liên kết logic cơ bản

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 của phép tính được đặt lại vào bit đầu tiên trong ngăn xếp. Giá trị của các bit còn lại trong ngăn xếp không bị thay đổi

Cú pháp STL: A n

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

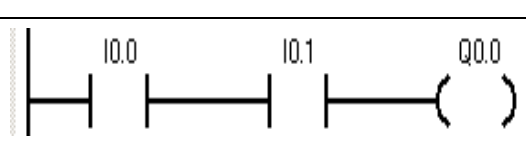
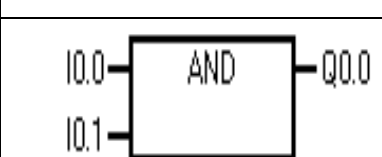


Đèn H1 chỉ sáng khi tất cả các công tắc được đóng lại. Khi 1 công tắc hở mạch thì đèn H1 cũng bị cắt mạch. Liên kết AND có trạng thái 1 khi tất cả các ngõ vào có trạng thái 1.

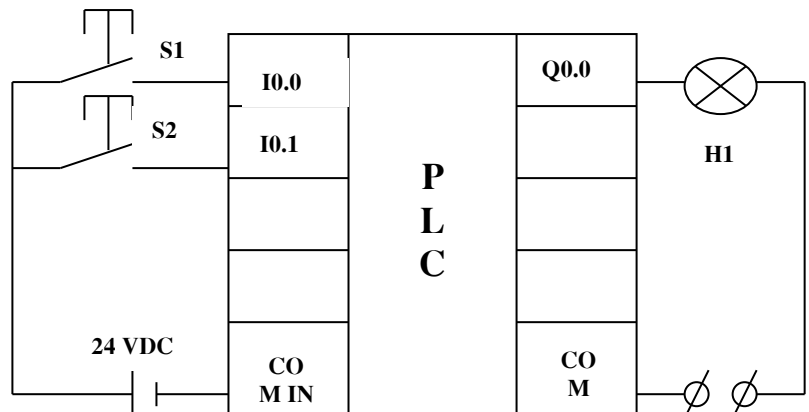
Để giải quyết vấn đề này, trước tiên ta cần phải lập một bảng xác lập các ngõ vào/ra (hay bảng trạng thái) để kết nối với PLC.

BẢNG TRẠNG THÁI		
KÝ HIỆU	ĐỊA CHỈ (TOÁN HẠNG)	GIẢI THÍCH (MÔ TẢ)
S1	I0.0	Công tắc thường hở
S2	I0.1	Công tắc thường hở
H1	Q0.0	Đèn báo

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

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

Sơ đồ kết nối với PLC như hình vẽ

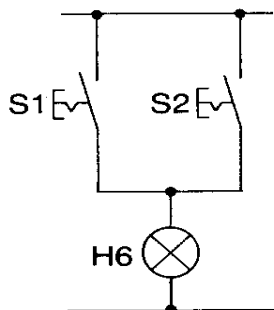


b/.Lệnh OR (O)

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

Cú pháp ở STL: O n

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



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

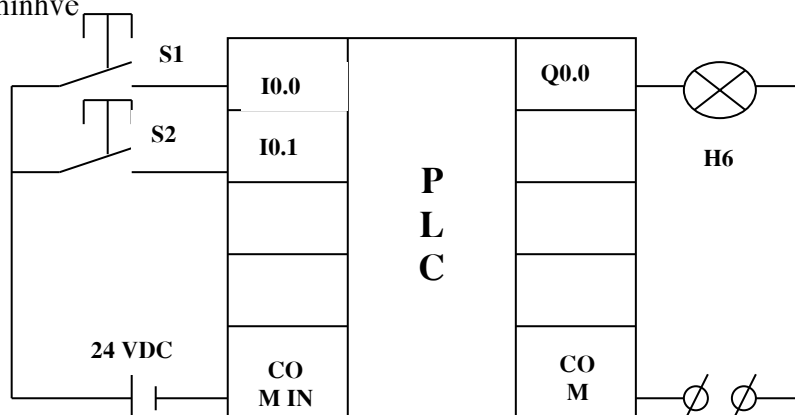
Để giải quyết vấn đề này, trước tiên ta cần phải lập một bảng xác lập các ngõ vào/ra (hay bảng trạng thái) để kết nối với PLC.

BẢNG TRẠNG THÁI		
KÝ HIỆU	ĐỊA CHỈ (TOÁN HẠNG)	GIẢI THÍCH (MÔ TẢ)
S1	I0.0	Công tắc thường hở
S2	I0.1	Công tắc thường hở
H6	Q0.0	Đèn báo

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

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

Sơ đồ kết nối với PLC như hình vẽ



1.3. Liên kết các cổng logic cơ bản

a/. Liên kết AND trước OR

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

b/. Liên kết OR trước AND

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

1.4. Các bài tập ứng dụng

Bài tập 1: Mạch duy trì tự giữ.

Thiết lập PLC điều khiển 1 động cơ theo yêu cầu:

- Nhấn Start, động cơ khởi động và làm việc.
- Nhấn Stop, động cơ ngừng
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải

Bài tập 2: Mạch đảo chiều quay động cơ.

Thiết lập PLC điều khiển 1 động cơ theo yêu cầu:

- Nhấn FDW, động cơ khởi động và quay theo chiều thuận.
- Nhấn REV, động cơ khởi động và quay theo chiều nghịch.
- Nhấn Stop, động cơ ngừng
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải

Bài tập 3: Mạch mạch điều khiển tuần tự cưỡng bức.

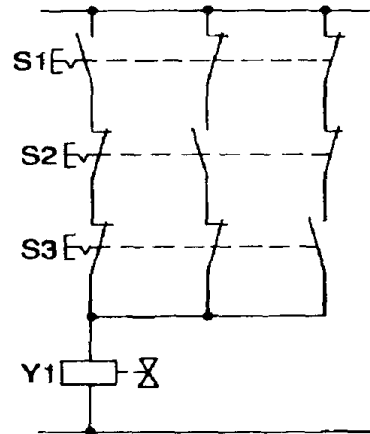
Thiết lập PLC điều khiển 2 động cơ theo yêu cầu:

- Khi mở máy thì động cơ 1 mới mở máy trước sau đó mới mở máy được động cơ 2.
- Khi dừng máy thì cả 2 động cơ cùng dừng
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải

Bài tập 4: Mạch lựa chọn các khả năng.(1 trong 3 khả năng)

Thiết lập PLC điều khiển yêu cầu: Chỉ cần nhấn 1 trong 3 nút nhấn đóng thì Y1 hoạt động.

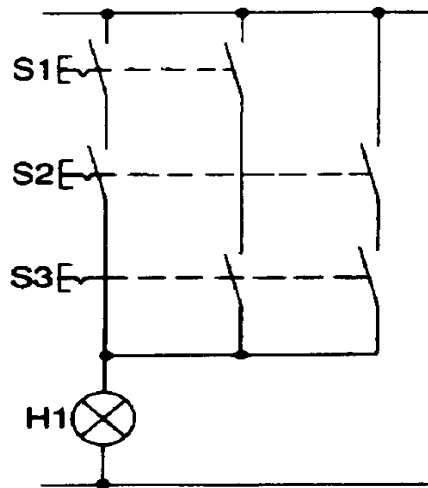
Sơ đồ nối cứng.



Bài tập 4: Mạch lựa chọn các khả năng.(2 trong 3 khả năng)

Thiết lập PLC điều khiển theo yêu cầu: Chỉ cần nhấn 2 trong 3 nút nhấn đóng thì H1 sáng.

Sơ đồ nối cứng



Bài tập 5: Mạch mạch điều khiển tuần tự cưỡng bức.

Thiết lập PLC điều khiển 2 động cơ theo yêu cầu:

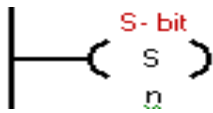
- Khi mở máy thì động cơ 1 mới mở máy trước sau đó mới mở máy được động cơ 2.
- Khi dừng máy thì động cơ 2 dừng trước sau đó mới dừng được động cơ 1
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải

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

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 có trạng thái 1 thì ngõ ra có tín hiệu 1, khi ngõ vào có trạng thái 1 thì ngõ ra có tín hiệu 0. người ta gọi mạch này là mạch nhớ tín hiệu (giống như mạch tự giữ trong điều khiển dùng rơle). Thay đổi trạng thái ngõ ra: Đặt (**Set**) hoặc Xóa (**Reset**)

2.2.lệnh Set (S): Lệnh dùng để đóng các điểm gián đoạn đã được thiết kế. Trong LAD, logic điều khiển dòng điện đóng các cuộn dây đầu ra. Khi dòng điều khiển đến các cuộn dây thì các cuộn dây đóng các tiếp điểm. Trong STL, lệnh truyền trạng thái bit đầu tiên của ngăn xếp đến các điểm thiết kế. Nếu bit này có giá trị bằng 1, các lệnh S đóng một

tiếp điểm hoặc một dãy các tiếp điểm (giới hạn từ 1 đến 255). Nội dung của ngăn xếp không bị thay đổi bởi các lệnh này.



Trong đó: + S: Set

+ S-bit: Star bit (bit bắt đầu)

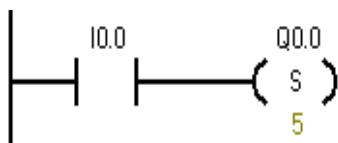
+ n: Số bit của chuỗi

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

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

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

+ Dạng LAD:

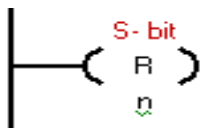


+ Dạng STL:

LD I0.0

S Q0.0, 5

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



Trong đó: + R: Reset

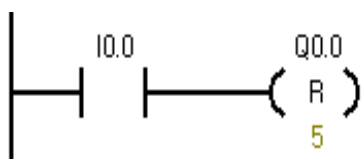
+ S-bit: Star bit (bit bắt đầu)

+ n: Số bit của chuỗi

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

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

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



LD I0.0

R Q0.0, 5

2.4. Các ví dụ ứng dụng bộ nhớ

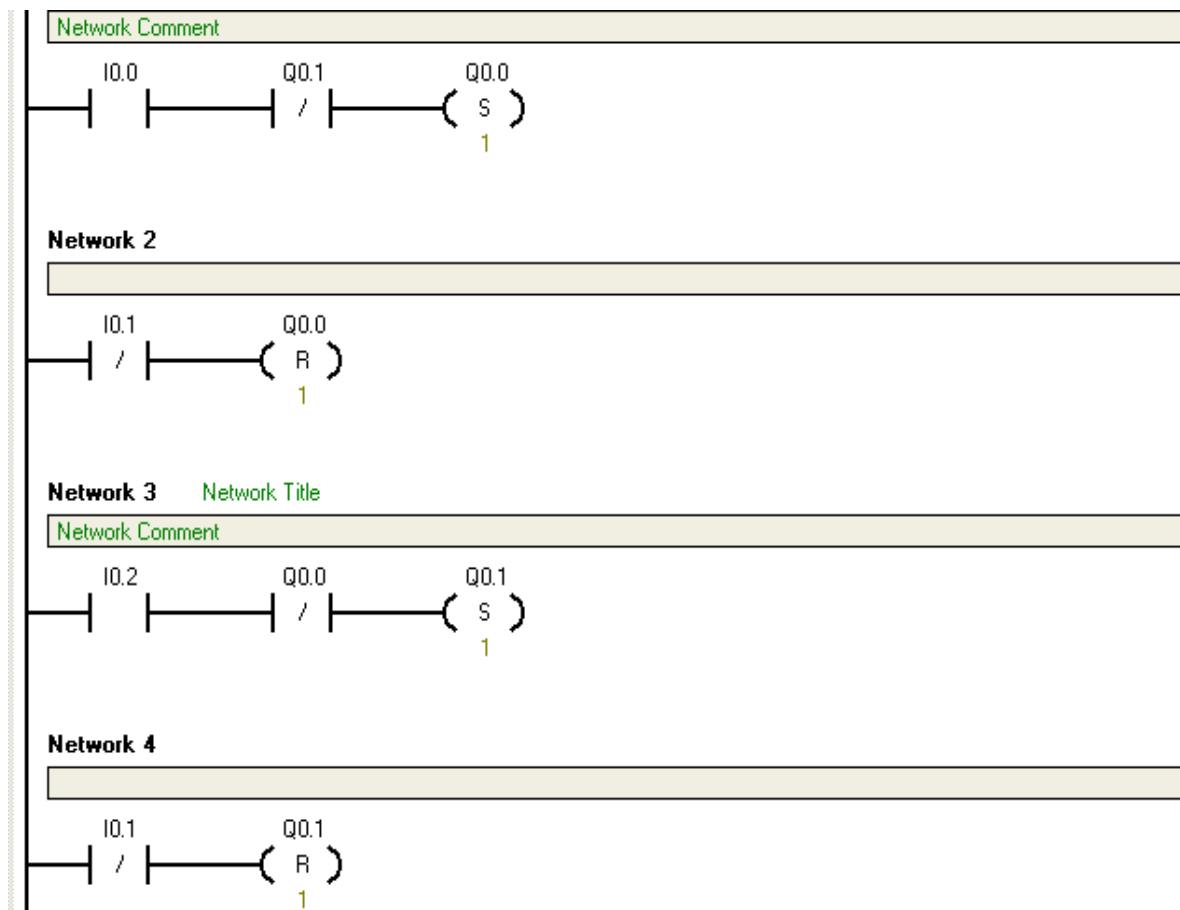
* VD1: Mạch chốt lẫn nhau bằng hai van từ

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

- Bảng Trạng Thái:

BẢNG TRẠNG THÁI		
KÝ HIỆU	ĐỊA CHỈ (TOÁN HẠNG)	GIẢI THÍCH (MÔ TẢ)
S1	I0.0	Nút nhấn thường hở
S2	I0.1	Nút nhấn thường đóng
S3	I0.3	Nút nhấn thường hở
Y1	Q0.0	Van từ 1
Y2	Q0.1	Van từ 2

- Chương trình (LAD):



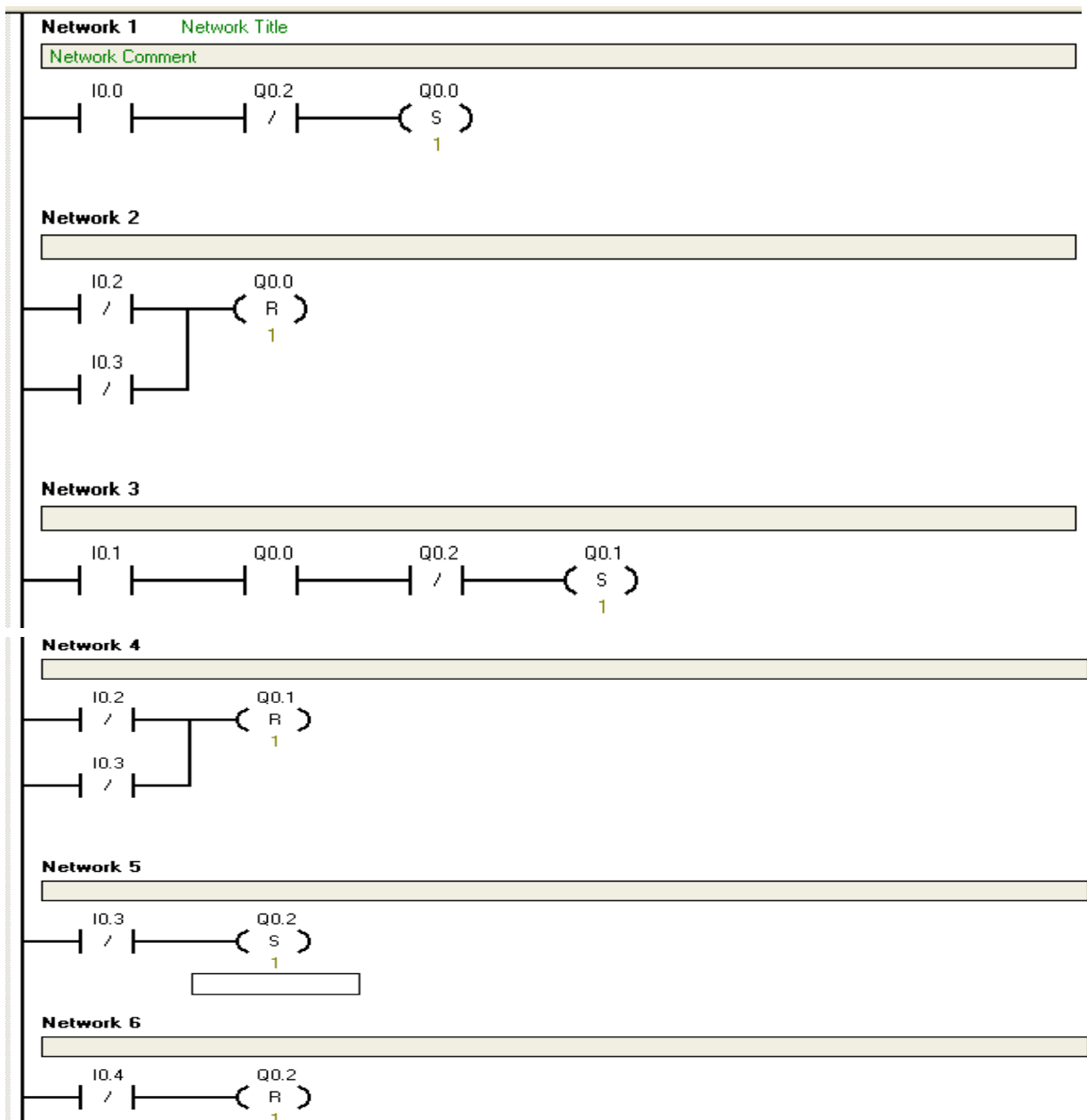
Ví dụ 2: Mạch tuần tự cưỡng bức báo lỗi

Qua việc khởi động nút nhấn S1 (I0.0) ngõ ra Q0.0 sẽ được đặt. Khi nhấn S2 (I0.1) ngõ ra Q0.1 cũng được đặt. Bằng nút nhấn S3 cả hai ngõ ra sẽ được đặt ngược lại. Khi có sự cố thì cả hai bộ nhớ Q0.0 và Q0.1 cũng được đặt ngược lại qua nút nhấn thường đóng S4 (I0.3). Nút nhấn thường đóng S5(I0.4) để phục hồi mạch, khi đó quá trình mới có thể bắt đầu.

- Bảng Trạng Thái:

BẢNG TRẠNG THÁI		
KÝ HIỆU	ĐỊA CHỈ (TOÁN HẠNG)	GIẢI THÍCH (MÔ TẢ)
S1	I0.0	Nút nhấn thường hở
S2	I0.1	Nút nhấn thường hở
S3	I0.2	Nút nhấn thường đóng
S4	I0.3	Nút nhấn thường đóng
S5	I0.4	Nút nhấn thường đóng
K1	Q0.0	Khởi động từ
K2	Q0.1	Khởi động từ
H1	Q0.2	Đèn báo

-Chương trình(LAD)



3. Timer (Bộ định thời)

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

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

- Timer tạo thời gian trễ không có nhớ

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

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

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

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

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

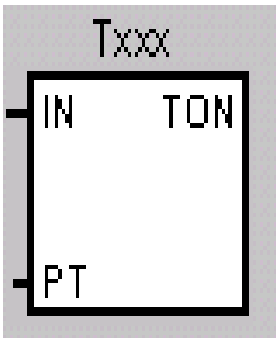
- Khi đầu vào có giá trị logic bằng 0, TON tự động reset còn TONR thì không. Timer TON được dùng để tạo thời gian trễ trong một khoảng thời gian (miền liên thông), còn với TONR thời gian trễ sẽ được tạo ra trong nhiều khoảng thời gian khác nhau.

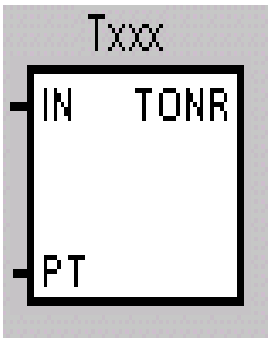
- Timer TON và TONR bao gồm ba loại với ba độ phân giải khác nhau, độ phân giải 1ms, 10ms, và 100ms. Thời gian trễ được tạo ra chính là tích của độ phân giải của bộ timer được chọn và giá trị đặt trước cho timer.

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

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

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

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

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

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

Khi reset một bộ timer, T-word và T-bit của nó đồng thời được 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 trạng thái logic bằng 0.

* Bài tập ứng dụng Timer

Bài tập 6: Mạch tự động đảo chiều quay động cơ.

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

- Chạy thuận (nghịch) 10 giây.
- Ngừng 2 giây.
- Chạy nghịch (thuận) 10 giây.
- Quá trình lặp lại liên tục cho đến khi nhấn nút dừng.
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

Bài tập 7: Mạch khởi động động cơ ba pha bằng phương pháp đổi nối sao-tam giác.

Lập trình PLC điều khiển động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc khởi động sao-tam giác theo yêu cầu sau:

- Khởi động ở chế độ sao 15 giây.
- Đổi nối 3 giây, sau đó làm việc định mức ở chế độ tam giác.
- Có thể chọn được chiều quay.
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

Bài tập 8: Mạch tự động khởi động tuần tự các động cơ.

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

- Khởi động động cơ Đ1 chạy 10 giây, sau đó động cơ Đ2 tự động chạy, 15 giây sau động cơ Đ3 tự động chạy tiếp.
- Khi dừng thì ngược lại.
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

Bài tập 9: Mạch tự động khởi động tuần tự bốn động cơ trong dây chuyền sản xuất liên tục.

Lập trình PLC điều khiển hệ thống động cơ trong dây chuyền sản xuất liên tục (gồm 4 động cơ) theo yêu cầu sau:

- Nhấn ON động cơ Đ1 chạy 10 giây, sau đó động cơ Đ2 tự động chạy 10 giây, sau động cơ Đ3 tự động chạy tiếp 10 giây thì đến động cơ Đ4 tự động chạy
- Khi nhấn OFF thì ngược lại.
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

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 ra làm 2 loại: Bộ đếm tiến (CTU) và Bộ đếm tiến/ lùi (CTUD).

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

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

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

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

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

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

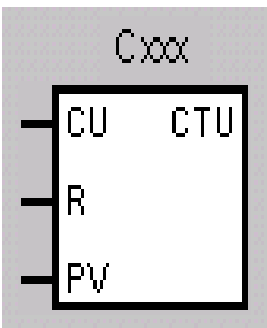
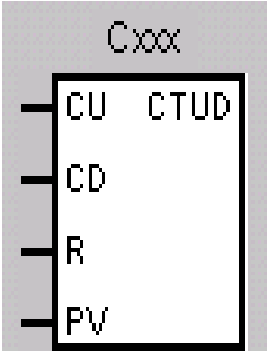
CTUD có giá trị đếm tức thời đúng bằng giá trị đang đếm và được lưu trong thanh ghi 2 byte C-word của bộ đếm. Giá trị đếm tức thời luôn được so sánh với giá trị đặt trước PV

của bộ đếm. Nếu giá trị đếm tức thời lớn hơn bằng giá trị đặt trước thì C-bit có giá trị logic bằng 1. Còn các trường hợp khác C-bit có giá trị logic bằng 0.

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

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

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

LAD	Mô tả	Toán hạng
	<p>Khai báo bộ đếm tiến theo sườn lên của CU. Khi giá trị đếm tức thời C-word Cxxx lớn hơn hoặc bằng giá trị đặt trước PV, C-bit (cxxx) có giá trị logic bằng 1. Bộ đếm được reset khi đầu vào R có giá trị logic bằng 1. Bộ đếm ngừng đếm khi C-word Cxx đạt giá trị cực đại (32.767).</p>	<p>Cxxx (word) CPU 224: C0 ÷ C47 và C80 ÷ C127 PV(word): VW, T, C, IW, QW, MW, SMW, AC, VD, *AC, AIW, Const.</p>
	<p>Khai báo bộ đếm tiến/lùi, đếm tiến theo sườn lên của CU và đếm lùi theo sườn lên của CD. Khi giá trị đếm tức thời C-word lớn hơn hoặc bằng giá trị đặt trước PV, C-bit (Cxxx) có giá trị logic bằng 1. Bộ đếm ngừng đếm tiến khi C-word đạt giá trị cực đại (32.767) và ngừng đếm lùi khi C-word đạt giá trị cực tiểu (-32.767) CTUD reset khi đầu vào R có giá trị logic bằng 1.</p>	<p>Cxxx (word) CPU 224: C48 ÷ C79 PV(word): VW, T, C, IW, QW, MW, SMW, AC, VD, *AC, AIW, Const.</p>

*** Bài tập ứng dụng bộ đếm:**

Bài tập 10: Lập trình PLC điều khiển động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc làm việc theo yêu cầu sau:

- Chạy thuận (nghịch) 7giây.
- Ngừng 3 giây.
- Chạy nghịch (thuận) 7giây.
- Quá trình trên lặp lại 3 lần.
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

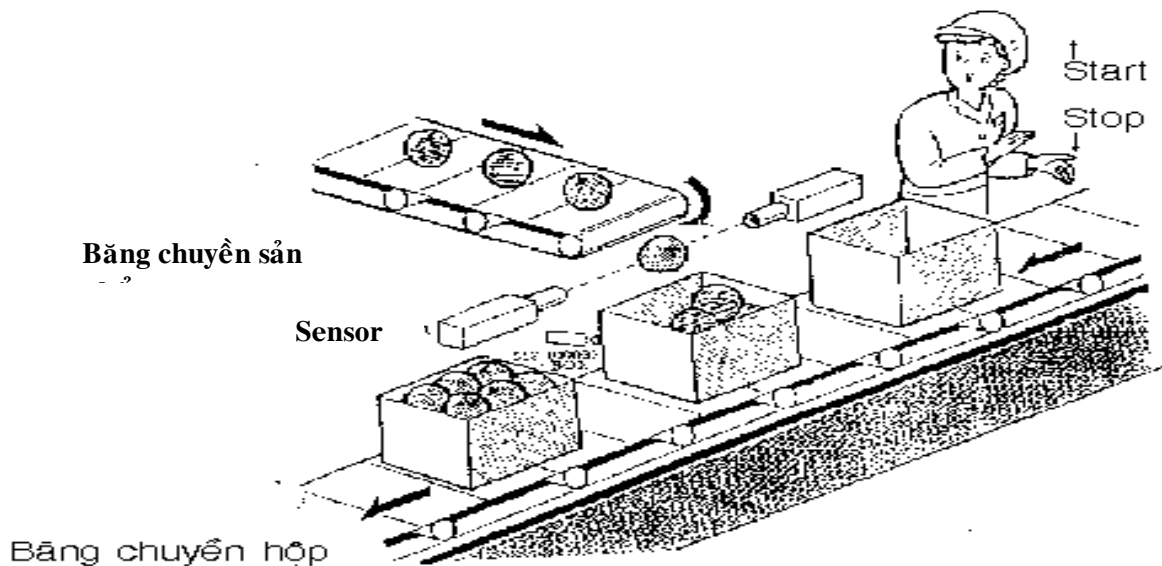
5. Bài tập tổng hợp

Bài tập 11: Lập trình PLC điều khiển hai động cơ làm việc theo yêu cầu sau:

- Động cơ Đ1 làm việc 5 giây rồi ngừng, sau đó đến động cơ Đ2 tự động làm việc 5 giây rồi ngừng 5 giây. Động cơ Đ2 lặp lại 3 lần như vậy, kể đến chu kỳ làm việc của hai động cơ lặp lại 5 lần rồi nghỉ.
- Muốn làm việc nữa thì khởi động lại.
- Mạch có bảo vệ các sự cố ngắn mạch, quá tải.

Bài tập 12: Lập trình PLC điều khiển kiểm soát dây chuyền đóng hộp, với yêu cầu sau:

- Khi nhấn START dây chuyền hộp vận hành. Khi hộp đựng công tắc hành trình thì dây chuyền hộp dừng lại, dây chuyền sản phẩm bắt đầu hoạt động. Cảm biến đầu vào bộ đếm sẽ đếm 10 sản phẩm. Khi đếm đủ sản phẩm thì dây chuyền sản phẩm dừng 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 STOP.
- Mạch có bảo vệ sự cố ngắn mạch, quá tải.
- Sơ đồ công nghệ, bảng trạng thái như sau:



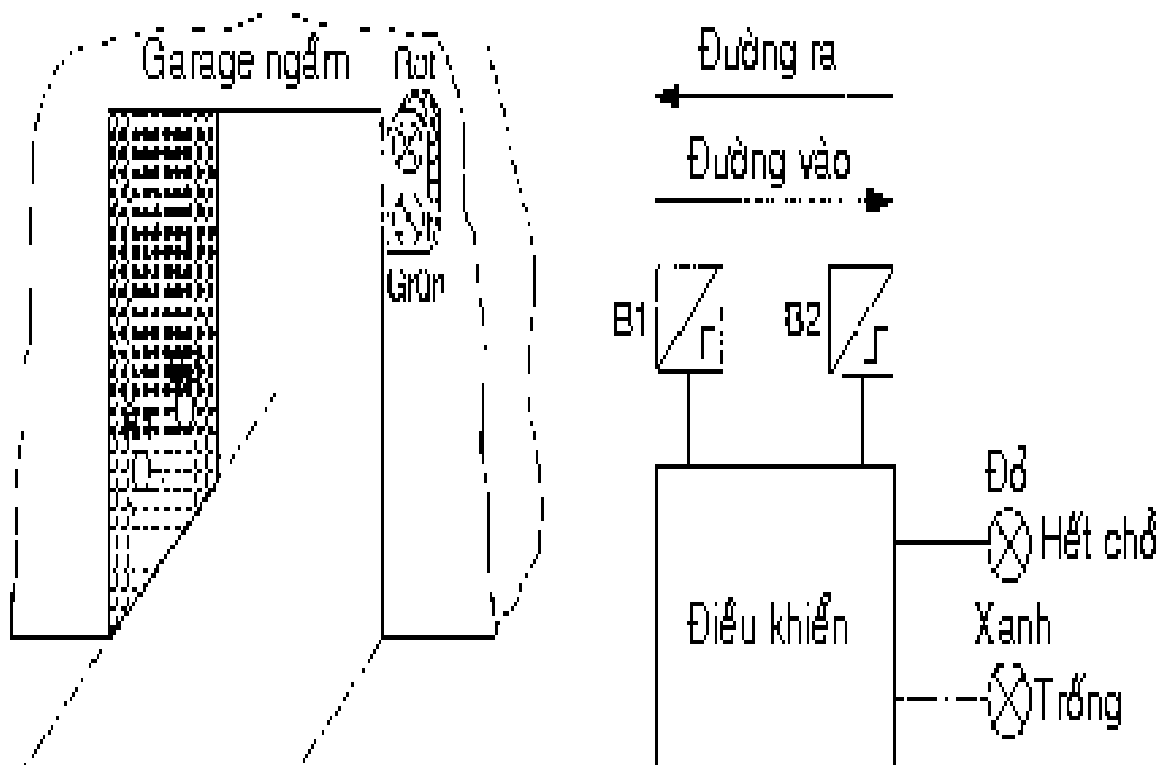
- Bảng trạng thái

BẢNG TRẠNG THÁI		
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 sản phẩm
S3	I0.3	Công tắc hành trình
K1	Q0.1	Động cơ băng chuyền thùng
K2	Q0.0	Động cơ băng chuyền sản phẩm

Bài tập 13: Lập trình PLC điều khiển kiểm soát chỗ cho GARAGE, với yêu cầu như sau:

- GARAGE cho tổng 50 xe đậu.
- Ở ngõ vào có hai đèn báo; đèn đỏ báo GARAGE hết chỗ, đèn xanh báo GARAGE còn chỗ trống.
- Đường vào và đường ra chỉ cho phép một xe chạy.
- Mạch có bảo vệ sự cố ngắn mạch, quá tải.

- Sơ đồ công nghệ, bảng trạng thái như sau:



- Bảng trạng thái

BẢNG TRẠNG THÁI		
Ký hiệu	Địa chỉ	Mô tả
B1	I0.0	Tế bào quang điện 1
B2	I0.1	Tế bào quang điện 2
H1	Q0.0	Đèn báo đỏ
H2	Q0.1	Đèn báo xanh

6. Lệnh nhảy và gọi chương trình con

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

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

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

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

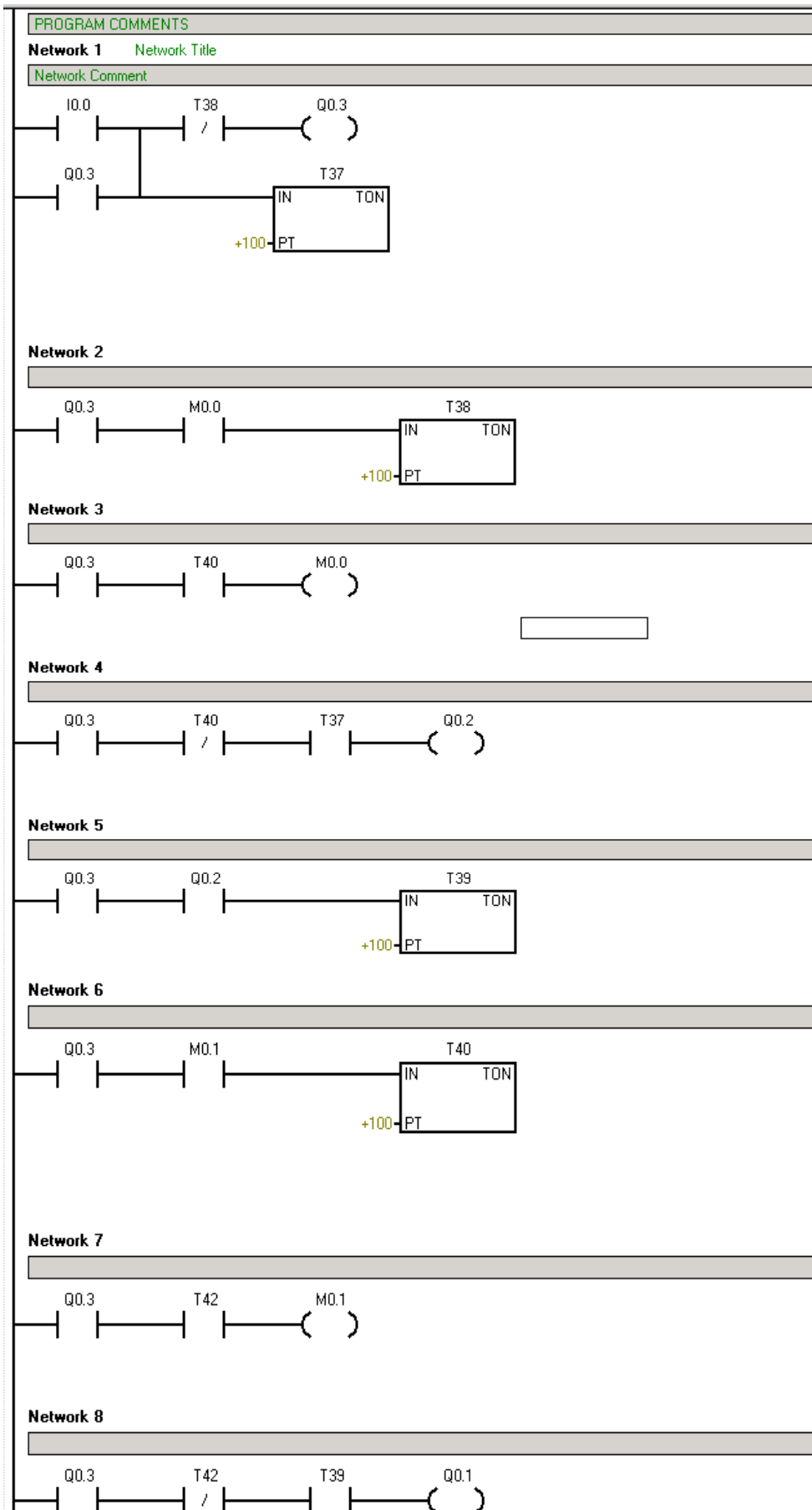
Khi một chương trình con được gọi, toàn bộ nội dung của ngăn xếp sẽ được cất đi, đỉnh của ngăn xếp nhận giá trị logic mới là 1, các bit khác còn lại của ngăn xếp nhận giá trị logic 0 và điều khiển được chuyển đến chương trình con đã được gọi. Khi thực hiện xong chương trình con và trước khi điều khiển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

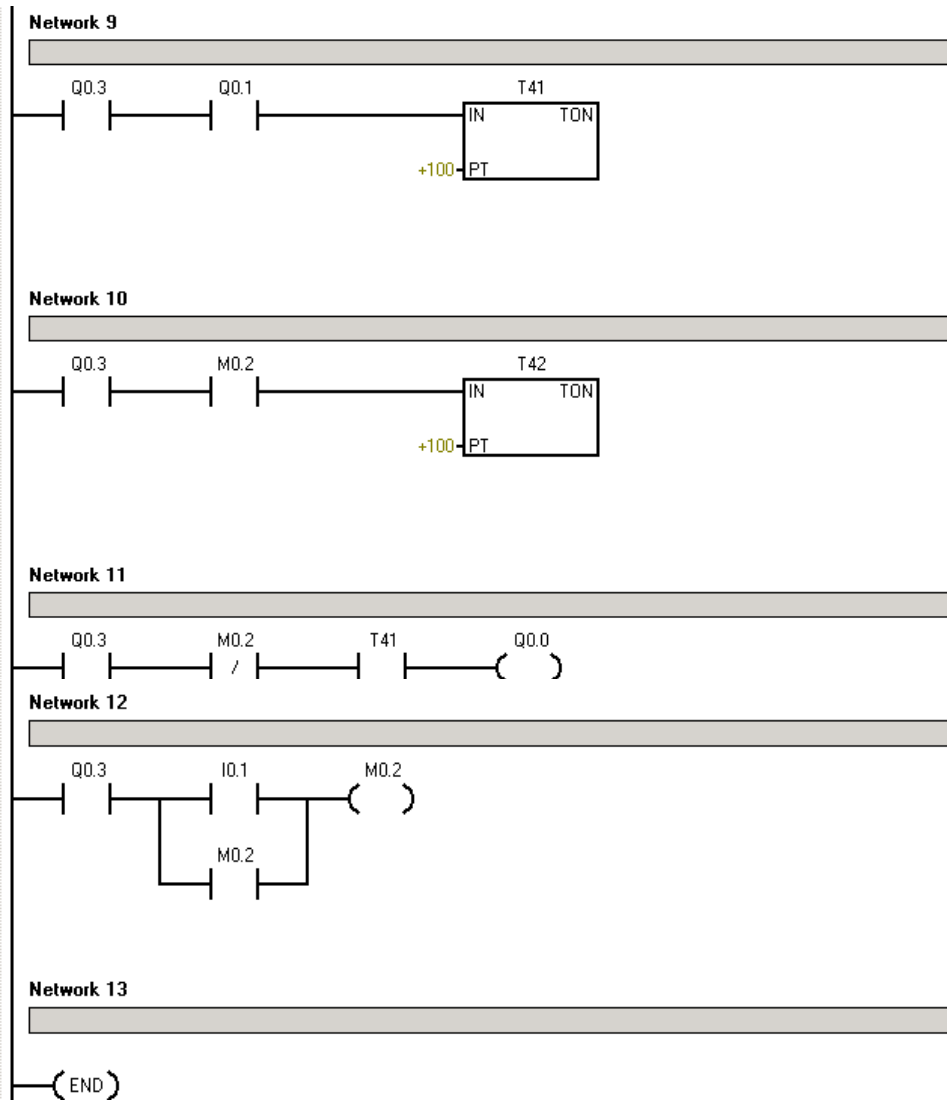
Nội dung của thanh ghi AC không được cất giữ khi gọi chương trình con, nhưng khi một chương trình xử lý ngắt được gọi, nội dung của thanh ghi AC sẽ được cất giữ trước khi thực hiện chương trình xử lý ngắt và nạp lại khi chương trình xử lý ngắt đã được thực hiện xong. Bởi vậy chương trình xử lý ngắt có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

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

LAD	STL	Mô tả	Toán hạng
$\left[\begin{array}{c} n \\ \text{JPM} \end{array} \right]$	JPM n	Lệnh nhảy thực hiện việc chuyển điều khiển đến nhãn n trong một chương trình	n: CPU 212: 0 ÷ 63 CPU 214: 0 ÷ 255 (224)
$\left[\begin{array}{c} \text{LBL: n} \end{array} \right]$	LBL n	Lệnh khai báo nhãn n trong một chương trình	
$\left[\begin{array}{c} n \\ \text{CALL} \end{array} \right]$	CALL n	Lệnh gọi chương trình con, thực hiện phép chuyển điều khiển đến chương trình con có nhãn là n	n: CPU 212: 0 ÷ 15 CPU 214: 0 ÷ 255 (224)
$\left[\begin{array}{c} \text{SBR: n} \end{array} \right]$	SBR n	Lệnh gán nhãn cho một chương trình con	
$\left[\begin{array}{c} \text{CRET} \end{array} \right]$	CRET	Lệnh trở về chương trình đã gọi chương trình con có điều kiện (bit đầu của ngăn xếp có giá trị logic bằng 1).	Không có
$\left[\begin{array}{c} \text{RET} \end{array} \right]$	RET	Lệnh trở về chương trình đã gọi chương trình con không điều kiện	

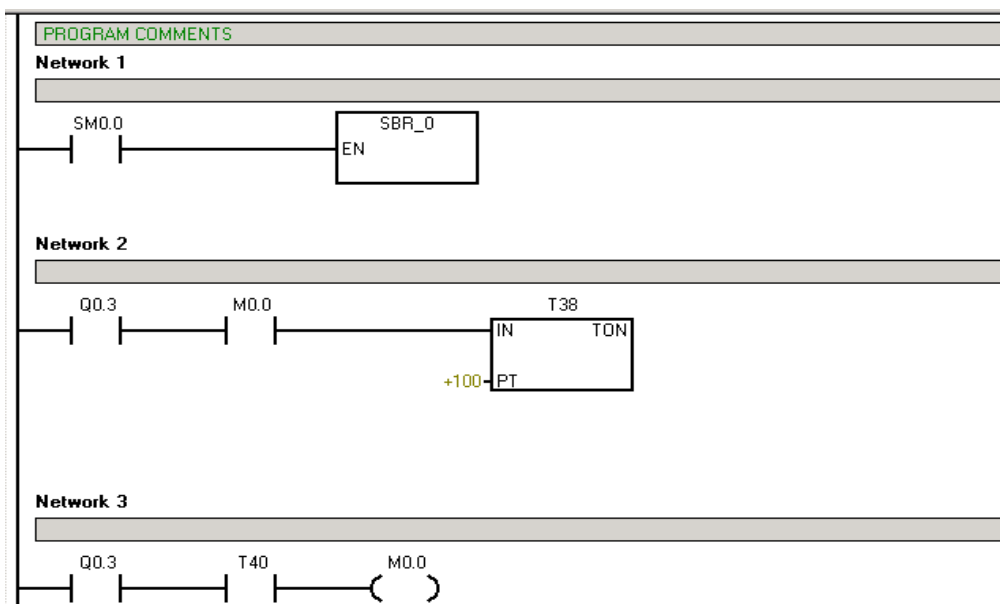
* Ví dụ chương trình phức tạp, quá dài ta có thể cắt NETWORK 1 bỏ vào SBR-0 (Chương trình con).

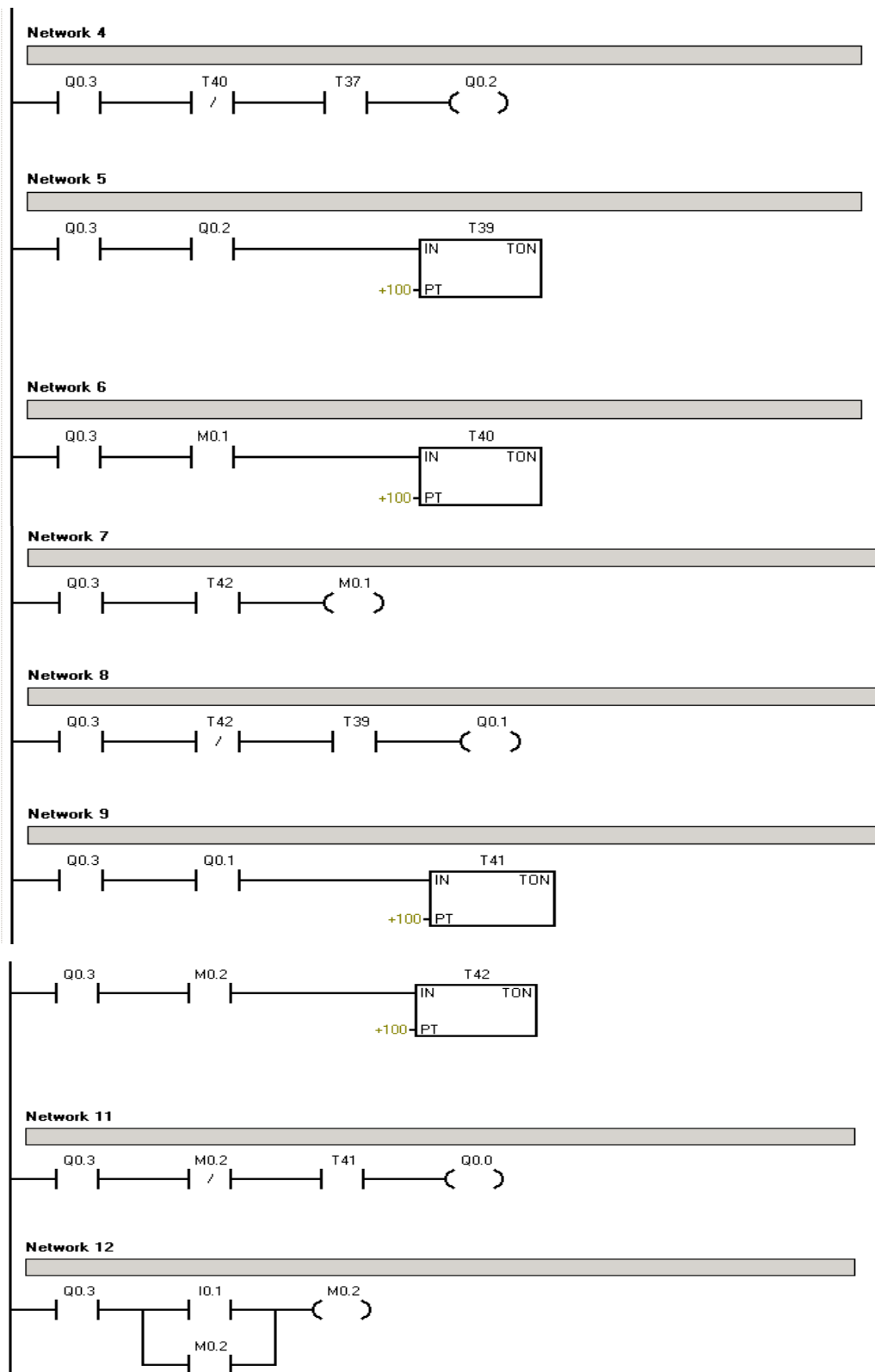




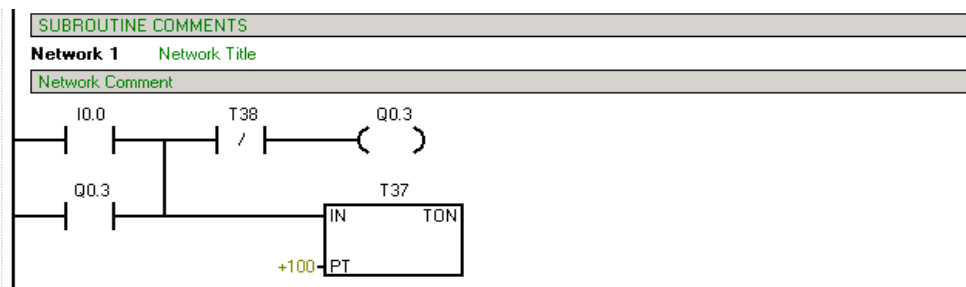
Cách thực hiện ta được

Ở chương trình chính (MAIN):





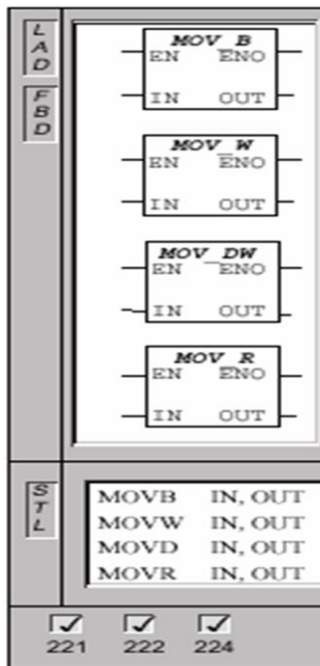
Ở chương trình con (SBR-0):



BÀI 3: CÁC PHÉP TOÁN SỐ CỦA PLC

1. Chức năng truyền dẫn

a/. Truyền dẫn Byte; Word; Doubleword



Phép truyền **Move Byte** sẽ thực hiện copy dữ liệu Byte tại ngõ vào IN và truyền tới Byte tại ngõ ra OUT.

Phép truyền **Move Word** sẽ thực hiện copy dữ liệu Word tại ngõ vào IN và truyền tới Word tại ngõ ra OUT.

Phép truyền **Move DoubleWord** sẽ thực hiện copy dữ liệu doubleword tại ngõ vào IN và truyền tới doubleWord tại ngõ ra OUT.

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

Khi xảy ra lỗi thì ngõ ENO bị SET = 0

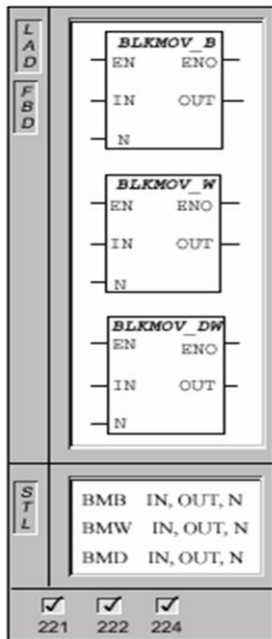
Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ

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

b/.Truyền một vùng nhớ dữ liệu

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

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

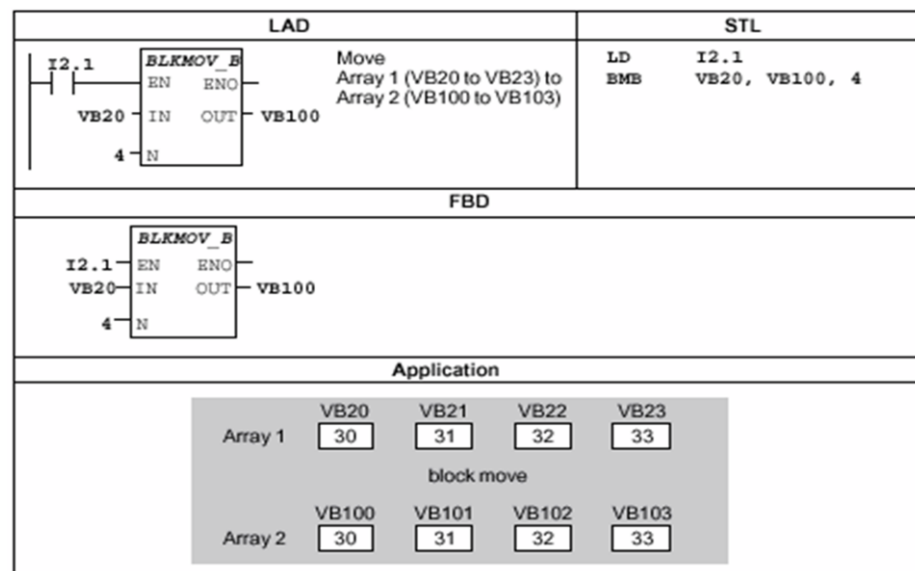


Bảng giới hạn vùng toán hạng và dạng dữ liệu hợp lệ:

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

Ví dụ về truyền một mảng dữ liệu BLKMOV:

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

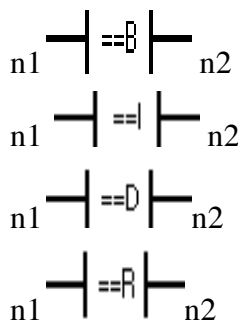
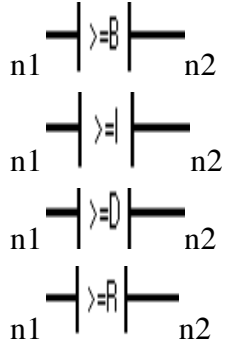
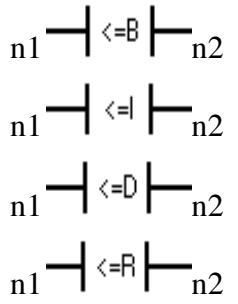


2. Chức năng so sánh

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

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

Biểu diễn các lệnh so sánh trong LAD:

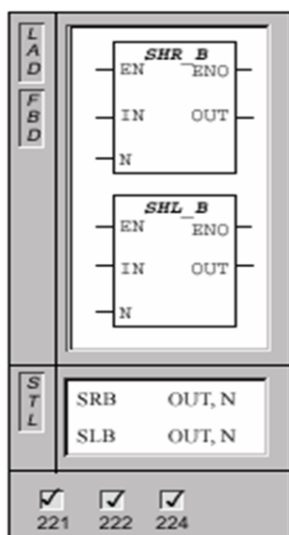
LAD	Mô tả	Toán hạng
	<p>Tiếp điểm đóng khi n1=n2</p> <p>B = byte I = Integer = Word D = Double Integer R = Real</p>	n1. n2(byte): VB, IB, QB, MB, SMB, AC, Const, *VD, *AC
	<p>Tiếp điểm đóng khi n1>n2</p> <p>B = byte I = Integer = Word D = Double Integer R = Real</p>	n1. n2(Word): VW, T, QW, MW, SMW,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

2.1. Chức năng dịch chuyển

- Dịch phải Byte SHR_B và Dịch trái Byte SHL_B:

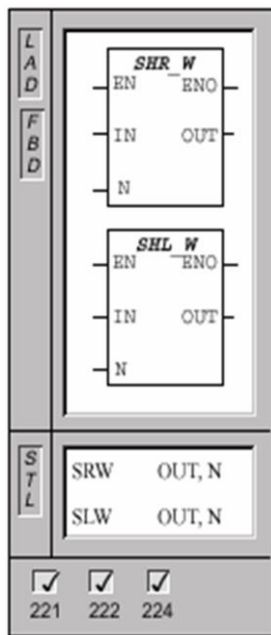
Các lệnh SHR_B và SHL_B sẽ dịch dữ liệu tại Byte ngõ vào IN sang phải hoặc sang trái với số vị trí dịch được nhập tại N, kết quả được chứa vào Byte ngõ ra OUT. Ở các lệnh SHIFT thì tại vị trí các Bit bị dịch sẽ lấp đầy bằng số 0. Số vị trí Bit cần dịch được nhập tại ngõ N <=8. Việc ảnh hưởng đến các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

- **Dịch phải Word SHR_W và Dịch trái Word SHL_W:**

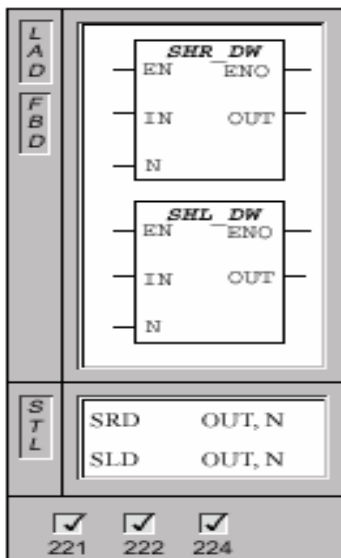


Các lệnh SHR_W và SHL_W sẽ dịch dữ liệu tại Word ngõ vào IN sang phải hoặc sang trái với số vị trí dịch được nhập tại N, kết quả được chứa vào Word có địa chỉ tại ngõ ra OUT. Tại vị trí các Bit bị dịch sẽ lấp đầy bằng số 0. Số vị trí Bit cần dịch được nhập tại ngõ $N \leq 16$. Việc ảnh hưởng đến các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

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

- **Dịch phải DoubleWord SHR_DW và Dịch trái Word SHL_DW:**



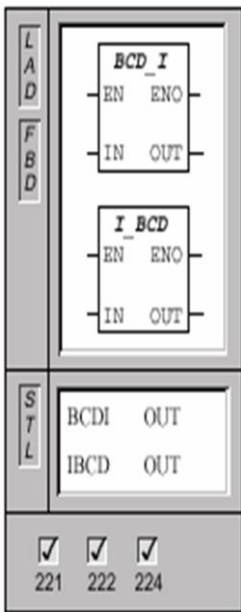
Các lệnh SHR_DW và SHL_DW sẽ dịch dữ liệu tại DoubleWord ngõ vào IN sang phải hoặc sang trái với số vị trí dịch được nhập tại N, kết quả được chứa vào DoubleWord có địa chỉ tại ngõ ra OUT. Tại vị trí các Bit bị dịch sẽ được lấp đầy bằng số 0. Số vị trí Bit cần dịch được nhập tại ngõ $N \leq 32$. Việc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

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

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

2.2. Chức năng chuyển đổi (Converter):

*Chuyển đổi số BCD_I và I_BCD



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.

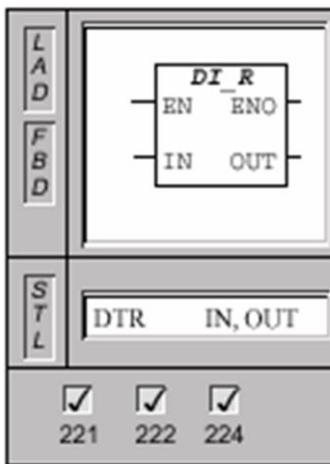
Lệnh chuyển đổi số I sang số BCD sẽ thực hiện việc chuyển đổi số I tại ngõ vào IN sang giá trị số BCD và chứa kết quả vào địa chỉ xác định tại ngõ ra OUT. Giá trị có thể nhập tại ngõ IN từ 0 đến 9999 Integer.

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:

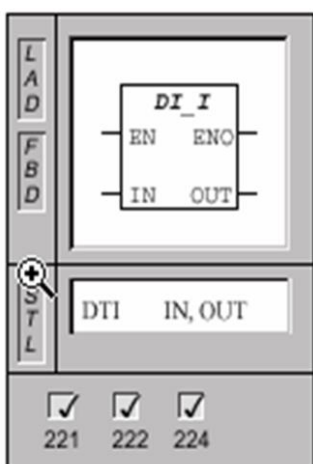


Lệnh chuyển đổi một số nguyên kép DI 32 Bit sang một số thực R và đặt kết quả vào địa chỉ được xác định tại ngõ ra OUT.

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:

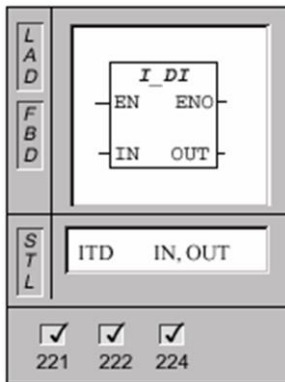


Lệnh chuyển đổi số DI_I chuyển đổi giá trị số DI tài nguyên vào IN sang một giá trị số nguyên I vào đầu kết quả tài nguyên OUT, nếu phép chuyển đổi bị trượt (kết quả vượt hạn khả năng chứa của tài nguyên OUT) thì ngõ ra không thay đổi và trạng thái ENO = 0.

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 I sang số nguyên kép DI:**

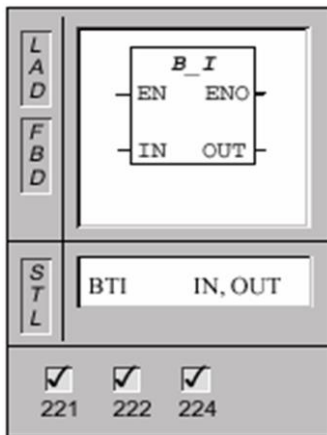


Lệnh chuyển đổi số I_DI chuyển đổi giá trị số I tài nguồn vào IN sang một giá trị số nguyên kép DI và kết quả quay tài nguồn OUT, nếu phép chuyển đổi bị trượt (kết quả luận hạn khả năng chứa của nguồn OUT) thì nguồn ra không thay đổi và trạng thái ENO = 0.

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:**

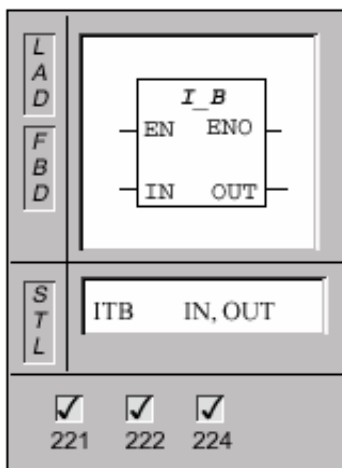


Lệnh chuyển đổi B_I chuyển đổi dữ liệu chứa trong Byte cụ thể của tài nguồn IN sang giá trị số nguyên, kết quả chứa vào biến xác định tài nguồn ra OUT.

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

Inputs/Outputs	Operands	Data Types
IN	VB, IB, QB, 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:**



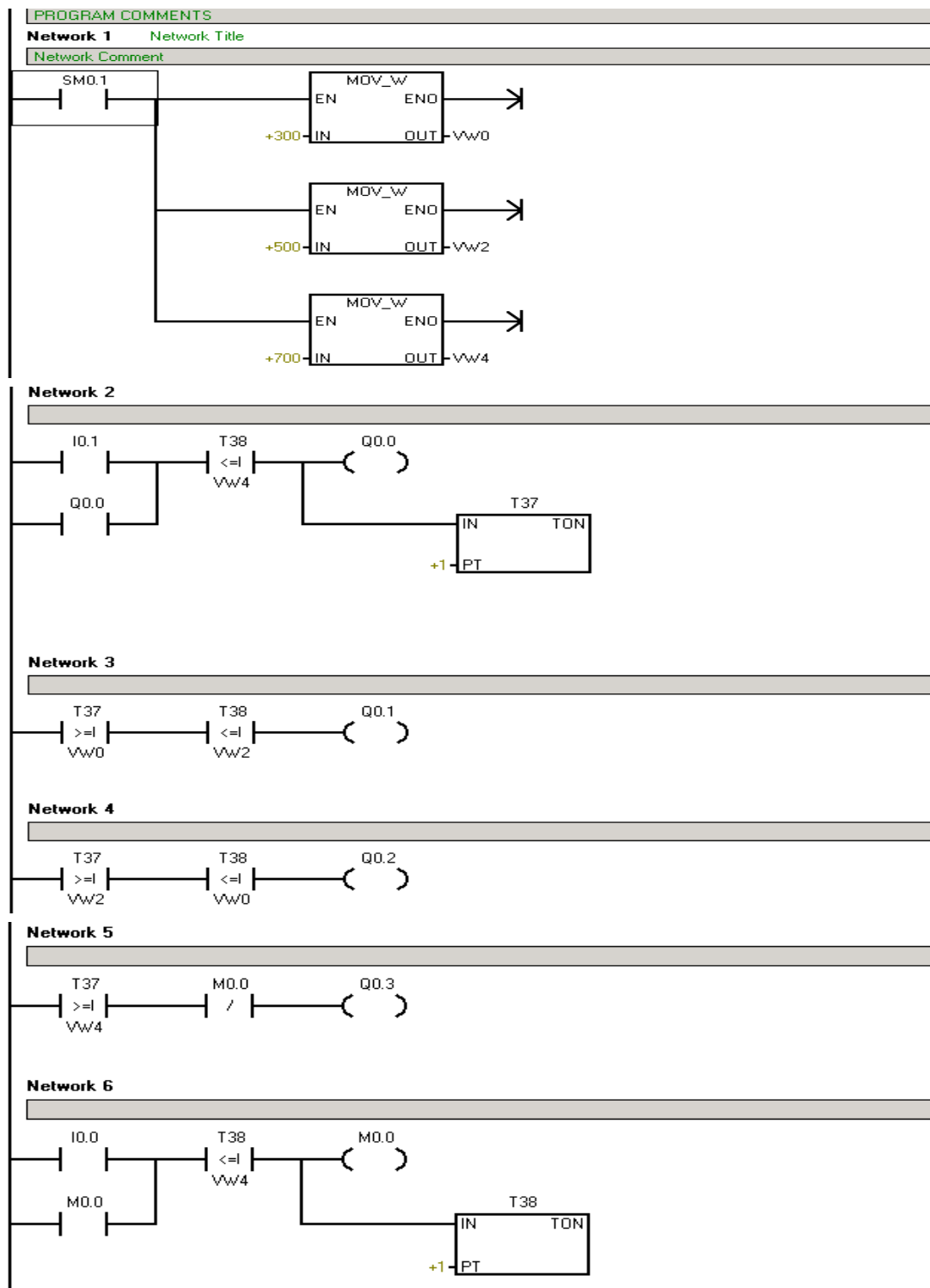
Lệnh chuyển đổi I_B chuyển đổi giá trị chứa trong word cụ thể của tài nguồn IN sang giá trị Byte, kết quả chứa vào biến xác định tài nguồn ra OUT. Các số nguyên cụ thể chuyển đổi lưu trữ từ 0 đến 255.

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, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE

Bài tập ứng dụng: Dùng lệnh MOV kết hợp các lệnh đã học viết chương trình điều khiển hệ thống gồm 4 động cơ làm việc theo trình tự từ động cơ đầu đến động cơ cuối, và dừng thì ngược lại.

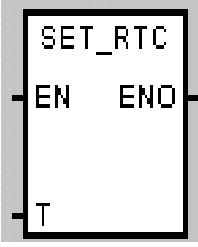
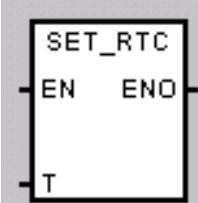
Chương trình(LAD):



3. Đồng hồ thời gian thực

Trong thiết bị khả trình S7-200 kể từ CPU 214 trở đi thì trong CPU có một đồng hồ ghi giá trị thời gian thực gồm các thông số về năm, tháng, ngày, giờ, phút, giây, và các ngày trong tuần. Đồng hồ này được cấp điện liên tục bởi nguồn pin 3 volt.

Khi thực hiện lập trình cho các hệ thống tự động điều khiển cần cập nhật đồng hồ thời gian thực này ta phải thông qua hai lệnh sau: **Lệnh đọc** và **Lệnh ghi**

Lệnh	LAD	STL	ý nghĩa	Vùng nhớ
Lệnh đọc		TODR T	Lệnh này đọc nội dung của đồng hồ thời gian thực rồi chuyển sang mã BCD và lưu vào bộ đệm 8 byte liên tiếp nhau. Trong đó byte đầu tiên được chỉ định bởi toán hạng T trong câu lệnh.	T: VB, IB, QB, MB, SMB, *VD, *AC.
Lệnh ghi		TODW T	Lệnh này ghi nội dung của bộ đệm 8 byte với byte đầu tiên được chỉ định trong toán hạng T vào đồng hồ thời gian thực.	

Các dữ liệu đọc, ghi với đồng hồ thời gian thực trong LAD và trong STL có độ dài một byte và phải được mã hoá theo kiểu số nhị phân BCD

Đồng hồ thời gian thực trong S7-200 sử dụng 2 chữ số cuối cho năm ví dụ : năm 2000 sẽ là 00, giá trị số năm thay đổi từ 00 đến 99.

Chúng nằm trong bộ đệm 8 byte liên tiếp nhau theo thứ tự:

Byte 1	Năm (0 ÷ 99)	VB0
Byte 2	Tháng (1 ÷ 12)	VB1
Byte 3	Ngày (1 ÷ 31)	VB2
Byte 4	Giờ (0 ÷ 23)	VB3
Byte 5	Phút (0 ÷ 59)	VB4
Byte 6	Giây (0 ÷ 59)	VB5
Byte 7	0	VB6
Byte 8	Ngày trong tuần(1 ÷ 7)	VB7

Các kiểu dữ liệu hợp lệ:

Năm (yy)	Tháng(mm)	Ngày(dd)	Giờ(hh)	Phút(mm)	Giây(ss)
(0 ÷ 99)	(1 ÷ 12)	(1 ÷ 31)	(0 ÷ 23)	(0 ÷ 59)	(0 ÷ 59)

Riêng giá trị về các ngày trong tuần là một số tương ứng với nội dung của nibble thấp (4 bit) trong byte theo kiểu:

Chủ nhật	Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7
1	2	3	4	5	6	7

Chú ý : CPU S7-200 không thực hiện kiểm tra là ngày tháng nhập có đúng với tháng tương ứng hay không, ví dụ: nếu nhập ngày 30 tháng 2 vẫn được chấp nhận. không được sử dụng các lệnh TODR và TODW trong cả chương trình chính (Main) và chương trình con (Subroutine), nếu xảy ra trường hợp này thì Bit SM 4.3 bị SET.

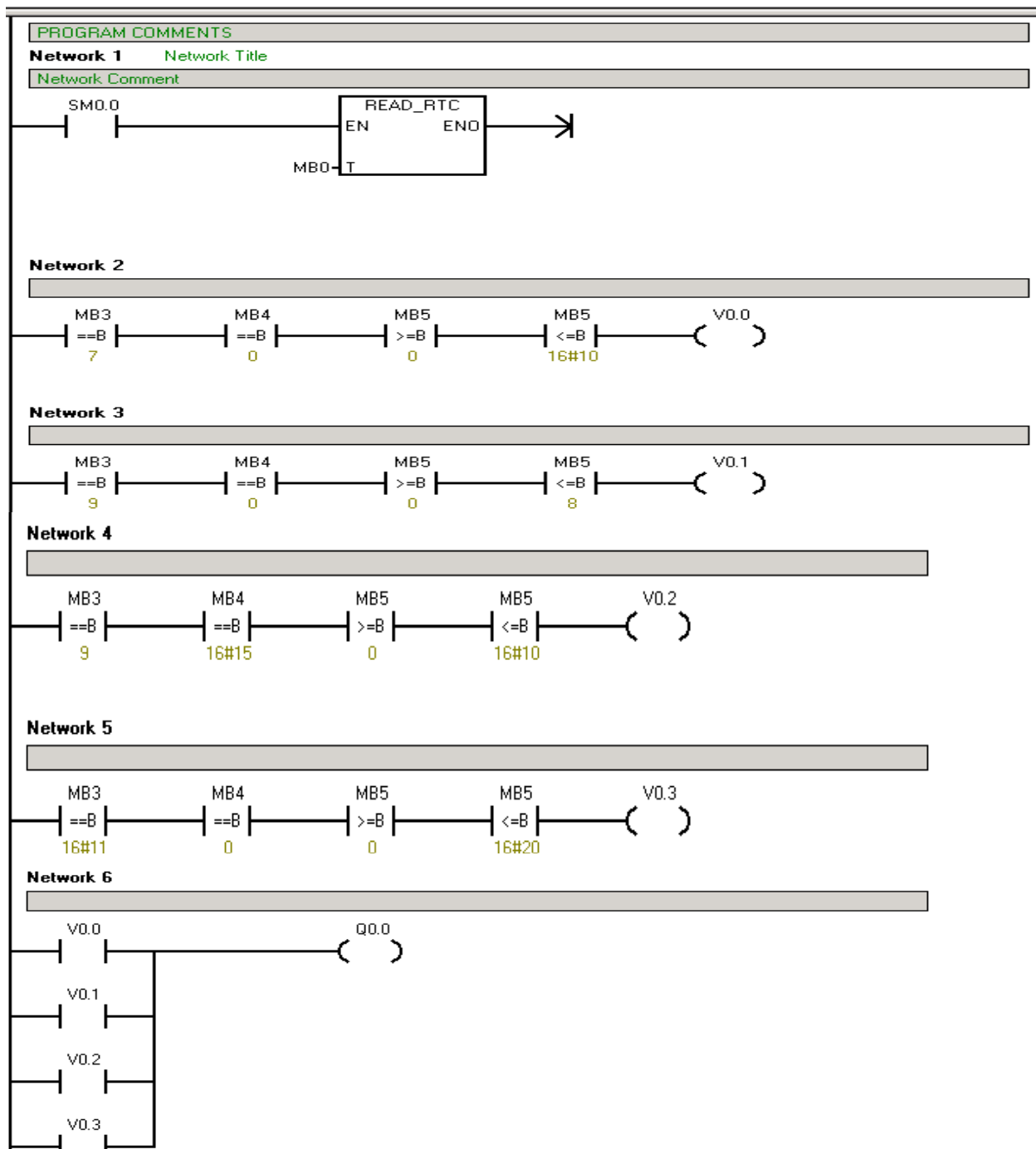
Do không sử dụng 2 chữ số đầu của năm nên CPU không bị ảnh hưởng khi chuyển sang thế kỷ mới.

Khi viết chương trình trên máy tính phải chuyển đồng hồ của máy tính về kiểu 24h.

* **Bài tập ứng dụng:** Viết chương trình PLC điều khiển báo giờ học cho trường học có yêu cầu:

- Từ 7h00' đến 7h00'10": chuông báo vào học
- Từ 9h00' đến 9h00'08": chuông báo giờ giải lao
- Từ 9h15' đến 9h15'10": chuông báo vào tiết học
- Từ 11h00' đến 11h00'20": chuông báo kết thúc buổi học

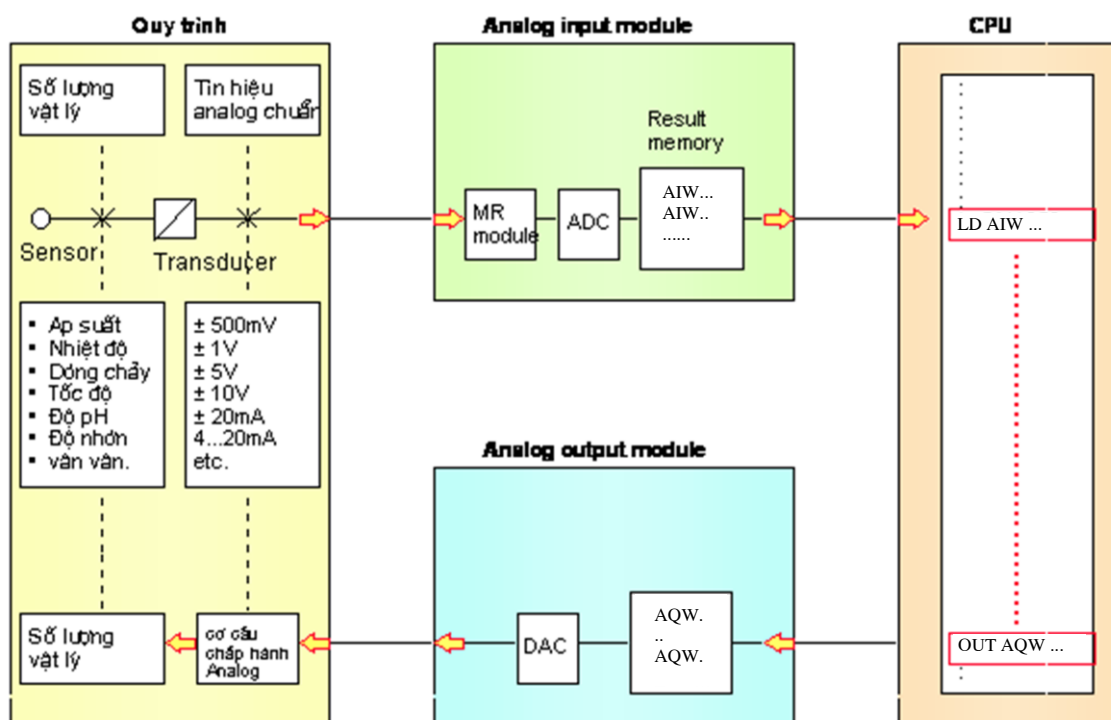
-Chương trình (LAD):



BÀI 4: XỬ LÝ TÍN HIỆU ANALOG

1. Tín hiệu Analog

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



2. Biểu diễn các giá trị Analog

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

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

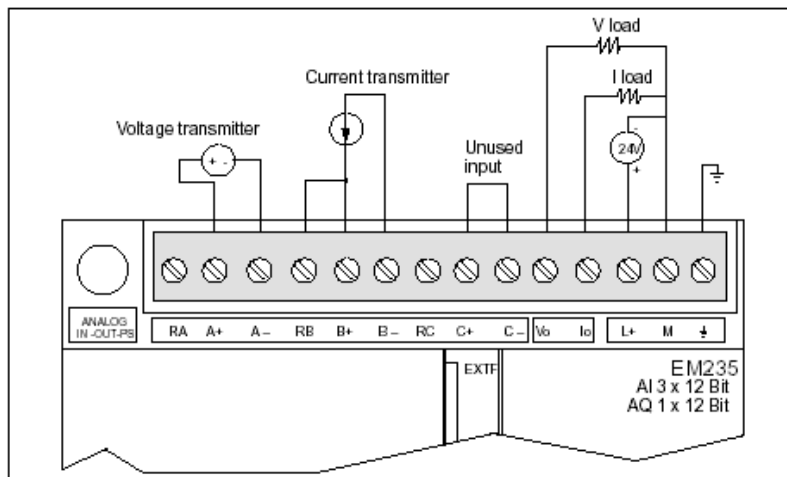
* = 0 or 1

3. Kết nối ngõ vào-ra Analog:

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

- + Đảm bảo rằng điện áp 24 VDC cấp nguồn cho Sensor không bị ảnh hưởng bởi nhiễu và ổn định .
- + Định tỷ lệ cho module (được mô tả bên dưới).
- + Dây nối cho Sensor cần để ngắn nhất tới mức có thể.
- + Sử dụng cặp 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

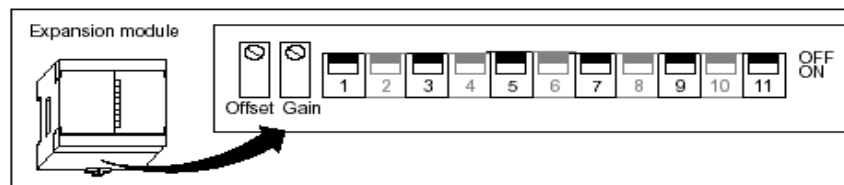


Phương pháp định tỷ lệ ngõ vào Analog (Input calibration)

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 modul, chọn phạm vi ngõ vào mong muốn
- + Cấp nguồn lại cho CPU và modul có AI
- + Sử dụng một Transmitter, một nguồn aụp, hay một nguồn dòng vaút ủaỏt vaút giao trữ 0 cho một trong các ngõ vaút.
- + Đọc giá trị mà CPU nhận được tại ngõ vào tương ứng đó.
- + Điều chỉnh biến trở đặt giá trị OFFSET cho tới khi giá trị đọc được là 0.
- + Điều chỉnh để tăng giá trị đặt vào tới định mức và xem giá trị mà CPU nhận được.
- + Điều chỉnh biến trở GAIN cho tới khi giá trị nhận được là 32000 hoặc tới 1 giá trị số mong muốn.
- + Lặp lại các bước trên nếu cần.



Điều chỉnh các Switch và biến trở chỉnh GAIN

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

Configuration Switch						Full-Scale Input	Resolution
1 ¹	3	5	7	9	11		
ON	ON	OFF	ON	OFF	OFF	0 to 50 mV	12.5 μ V
ON	ON	OFF	OFF	ON	OFF	0 to 100mV	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.

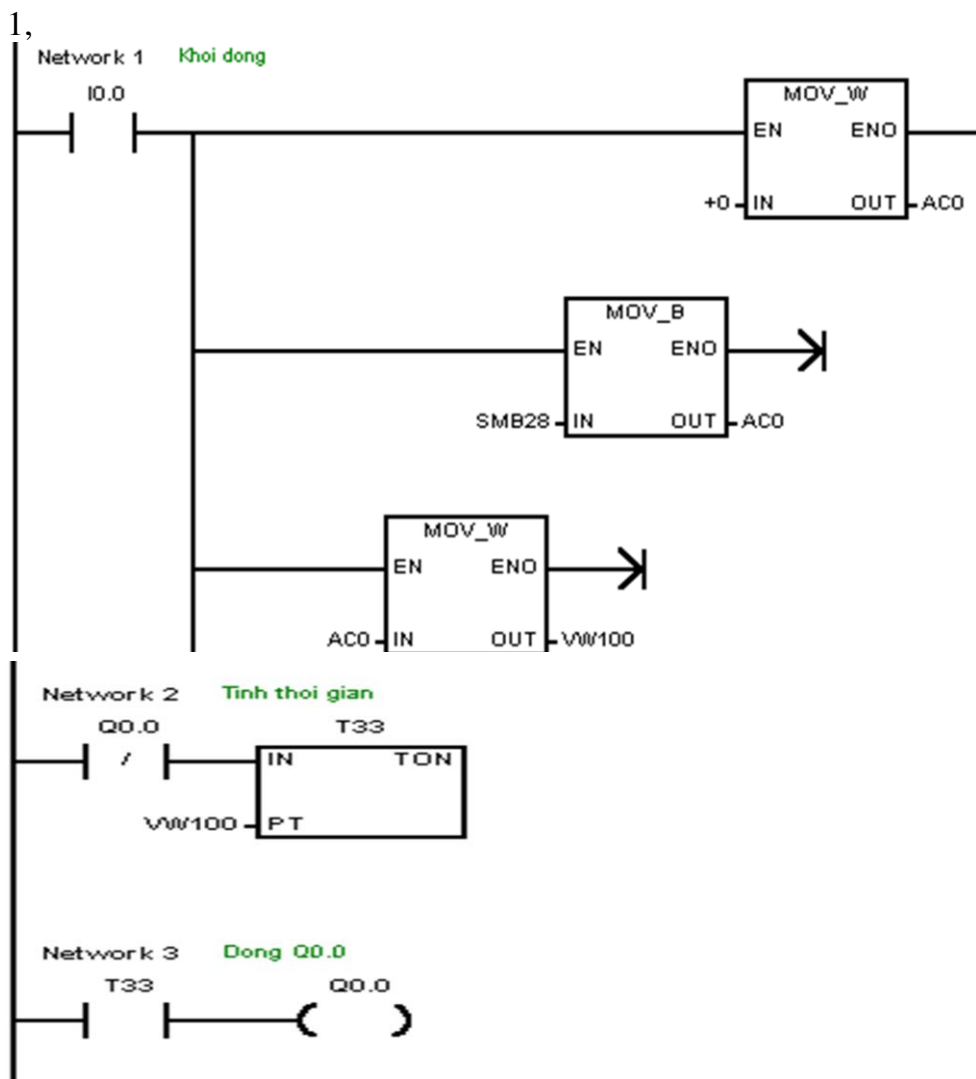
4. Hiệu chỉnh tín hiệu Analog

Trên CPU S7-200 có 2 biến trở (2 biến trở này nằm dưới nắp của module), có thể sử dụng 2 biến trở này để tăng hoặc giảm giá trị được lưu trữ trong các Byte của vùng nhớ Special Memory (SMB 28 và SMB 29). Các giá trị chỉ đọc trong 2 Byte này có thể được sử dụng cho nhiều chức năng khác nhau. Chẳng hạn, dùng để cập nhật giá trị hiện hành cho 1 Timer, một Counter, thay đổi giá trị đặt trước, đặt các giá trị giới hạn.

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

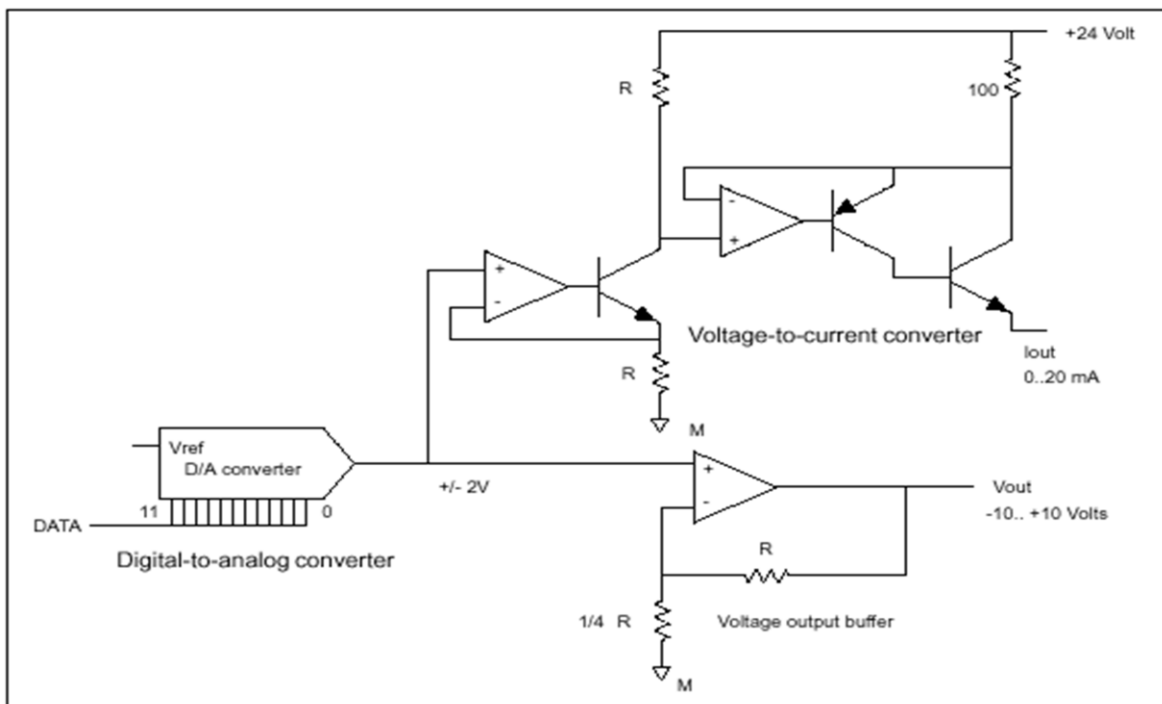
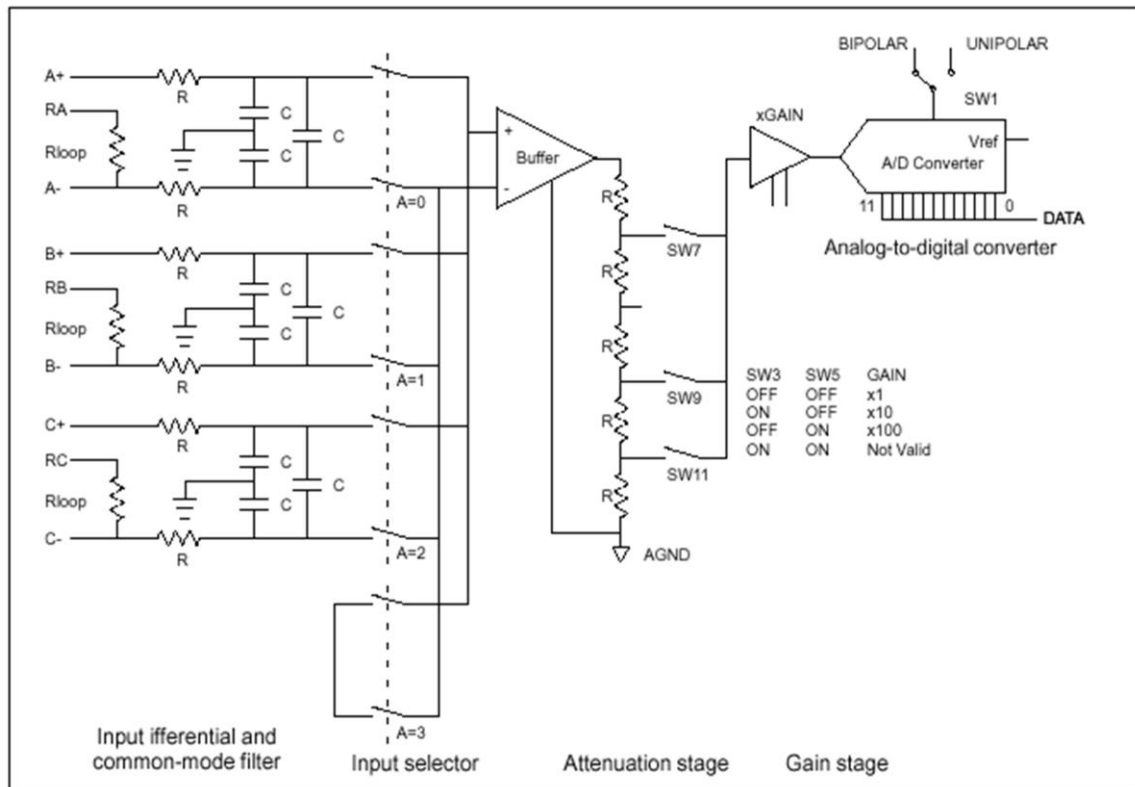
Để thực hiện điều chỉnh này, phải sử dụng một Tuốc vít nhỏ: nếu xoay biến trở sang phải là tăng giá trị, còn xoay sang trái là giảm giá trị.

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

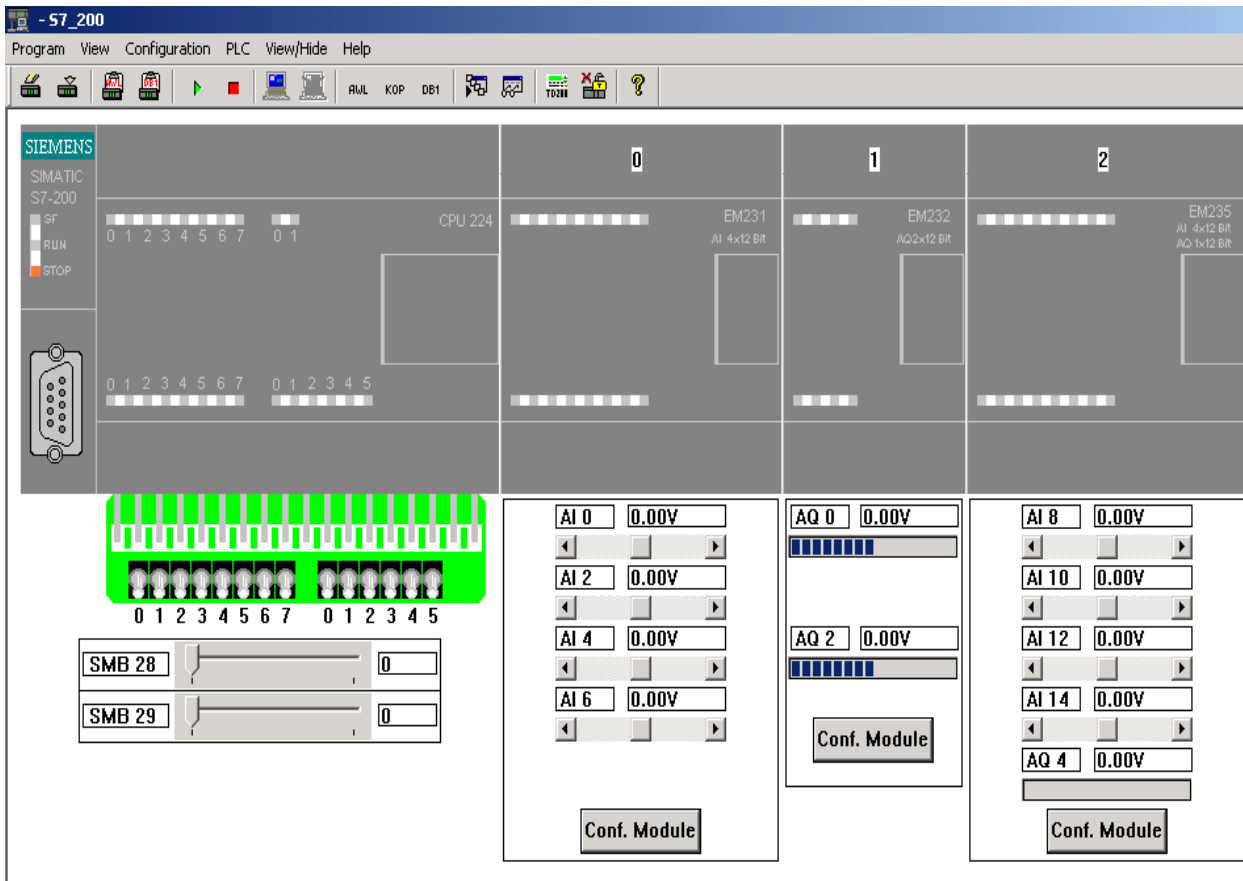


Sau đây là sơ đồ nguyên lý mạch của modul EM 235 3AI/ 1AO

Sơ đồ mạch ngõ vào :



EM235 4 I × 12 Bits
 1 Q × 12 Bits



Tín hiệu Analog là các tín hiệu tương tự (0 – 10VDC, hoặc 4-20mA.....), Hầu hết các ứng dụng của chương trình PLC Siemens nói riêng hay các ứng dụng khác đều cần phải đọc các tín hiệu analog. Tín hiệu analog có thể là tín hiệu từ các cảm biến đo khoảng cách, cảm biến áp suất, cảm biến đo trọng lượng.....

Các bước đọc tín hiệu Analog:

a/Đọc tín hiệu analog từ Modul EM231:

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

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

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

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

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

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

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

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

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

Cho phép ngắt (ENI)

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

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

Ví dụ :

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

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

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

□ Thông thường các tín hiệu Analog đọc vào bao giờ người sử dụng cũng mong muốn đọc được chính giá trị mong muốn (Ví dụ: giá trị khối lượng trong đọc đầu cân Loadcell, giá trị áp suất trong đọc tín hiệu từ cảm biến áp suất....)

□ Phương pháp đọc Analog trong trường hợp này ta sẽ không cần quan tâm nhiều đến chế độ đơn cực hay lưỡng cực, mà chỉ cần xác định được 2 điểm, từ đó lập được phương trình đường thẳng (Giá trị mong muốn đọc theo AIW)

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

□ Bước 1: Ta cần xác định 2 điểm:

Điểm 1: Ta online trên máy tính, đọc giá trị AIW0 là x1, trong trường hợp ở điểm 1 (Điểm 1 là điểm ta đặt quả cân chuẩn 1: có khối lượng m1 lên bàn cân), Tương tự ta có thể xác định được điểm 2 (tương ứng x2 và m2).

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

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

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

Đó chính là phương trình khối lượng theo AIW.

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

Phương trình lập:

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

$Y = 1 \times X / 32$

Vậy : Khối lượng = AIW / 32

b/ Xuất tín hiệu analog qua modul EM232:

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

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

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

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

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

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

c/ Modul EM235:

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

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

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

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

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

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

BÀI 5: PLC CỦA CÁC HÃNG KHÁC

Trong bài này giới thiệu một số đặc điểm của các PLC của các hãng khác như Omron, Mitsubishi của Nhật Bản.

1 PLC của hãng Omron:

* Các PLC họ CPM1A

Các PLC loại này là các PLC có kích thước nhỏ gọn, cấu trúc đồng nhất một khối, trên mỗi CPU có sẵn từ 10, 20, 30, 40 ngõ I/O. Tất cả các CPU dạng này đều có ngõ ra Relay.

Để dữ liệu trong RAM không bị mất khi PLC bị mất điện thì người dùng có thể sử dụng một Card nhớ Flash Memory.

Khả năng mở rộng:

Các modul CPU có thể mở rộng thêm 3 modul mở rộng, mỗi module mở rộng có từ 30-40 I/O

Chức năng lọc tín hiệu ngõ vào:

Các ngõ vào có khả năng đáp ứng nhanh, nhận biết trạng thái tín hiệu ngõ vào trong vòng 0,2 ms và có khả năng chống nhiễu

Bộ đếm tốc độ cao

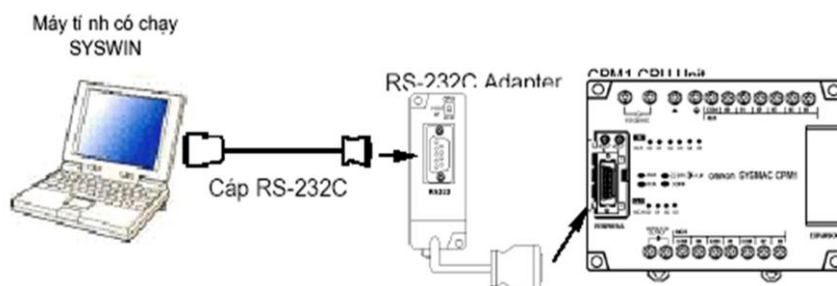
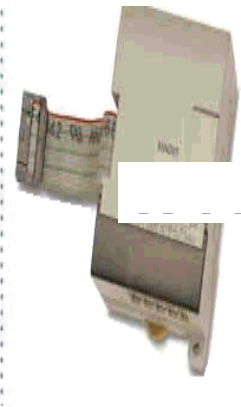
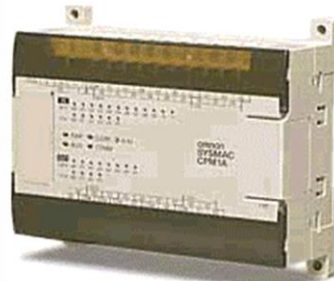
Bộ đếm tốc độ cao cho phép tăng khả năng đếm lên hoặc xuống và không bị ảnh hưởng bởi chu kỳ quét của CPU.

Timer

Các Timer có khả năng trì hoãn từ 0.5 ms đến 999,9s. có tổng số 128 Timer.

Chức năng chỉnh định tín hiệu Analog

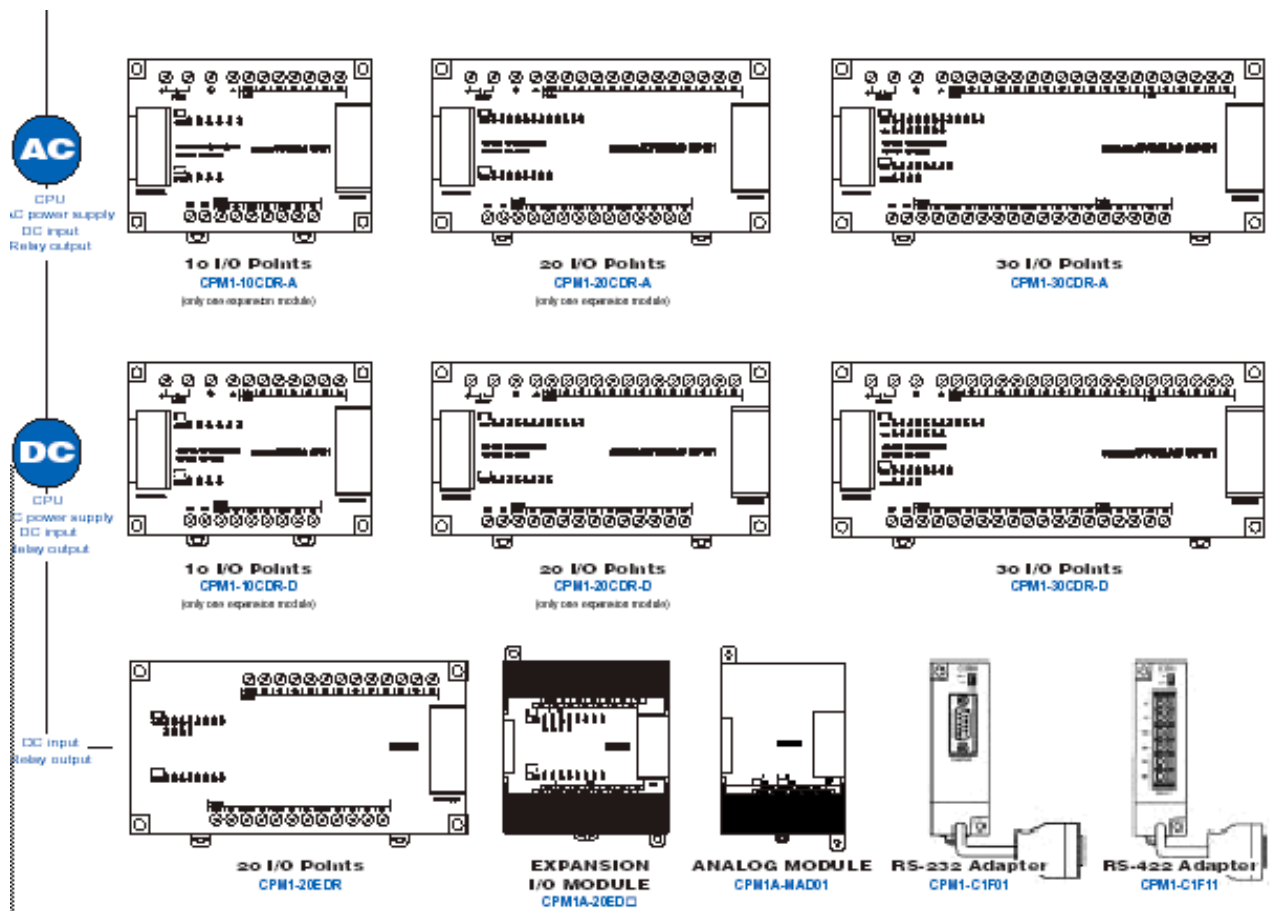
Trên CPU có 2 kênh chỉnh định tín hiệu Analog



Truyền thông

Các PLC CPM1A có khả năng kết nối với các máy tính cá nhân PC, chuyển đổi giao tiếp RS232 cho truyền thông 1-1 và bộ chuyển đổi giao tiếp RS-422 cho truyền thông 1-n.

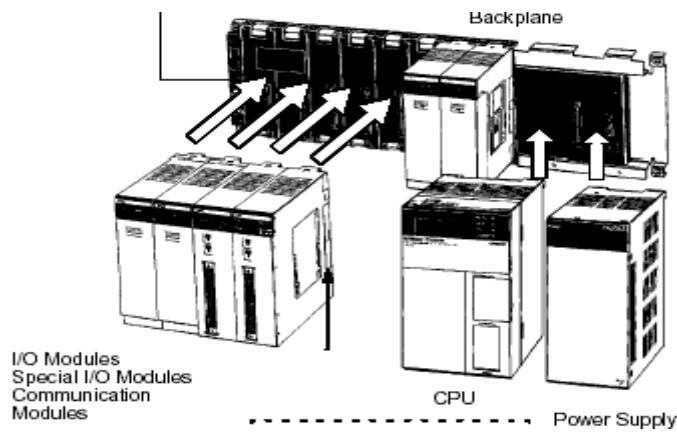
Dưới đây là toàn bộ các Module họ CPM1A



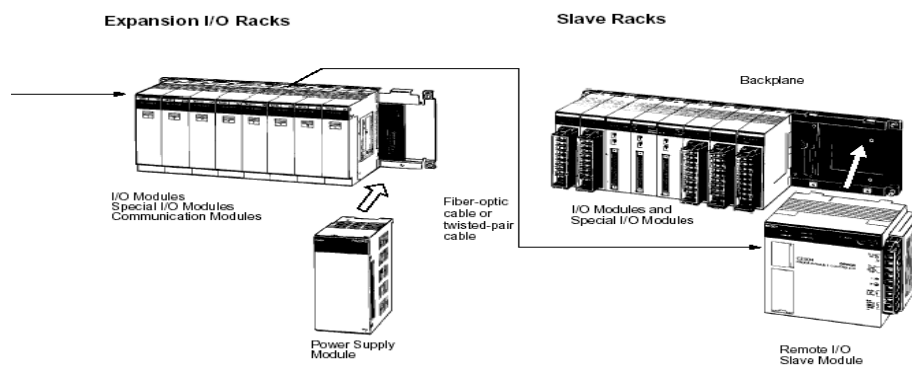
* Các CPU họ C200Hα

Các PLC họ C200Hα là họ các PLC cỡ trung bình, được phát triển dựa trên các PLC họ C200H. Các CPU C200Hα có nhiều ưu điểm như bộ nhớ được mở rộng hơn, tốc độ xử lý nhanh hơn, được hỗ trợ Protocol Marco (thủ tục truyền thông cho các Modul ASCII và Basic Modul) và có thể tùy chọn gắn thêm các card PCMCIA. Dưới đây là một số đặc điểm của các PLC thuộc họ này.

- + Có 11 loại CPU thuộc họ này
- + Nguồn cung cấp là module tách rời với CPU
- + Tổng số I/O: 1184
- + Tốc độ xử lý: 0.1 μs/lệnh
- + Khả năng mở rộng là 3 Backplane(rack mở rộng)
- + Các chức năng tích hợp cho phép các PLC họ này giao tiếp với nhau một cách dễ dàng.
- + Khả năng truyền thông với các bảng điều khiển vận hành (OP), các bộ đọc mã vạch,... sử dụng Devicenet cho phép chúng có thể kết nối dễ dàng với các thiết bị của các hãng khác như các bộ biến tần hay các thiết bị Analog.
- + Sử dụng phần mềm SYSMAC V1.2 hoặc SYSWIN V3.0 trở lên.



Cấu hình của 1 PLC họ C200Hα



Backplane và các modul mở rộng

2 PLC của hãng Mitsubishi:

Trong phần này chỉ đề cập tới các CPU họ FX1S, FX 1N, FX2N, FX2NC , cụ thể là các CPU họ FX 2NC.

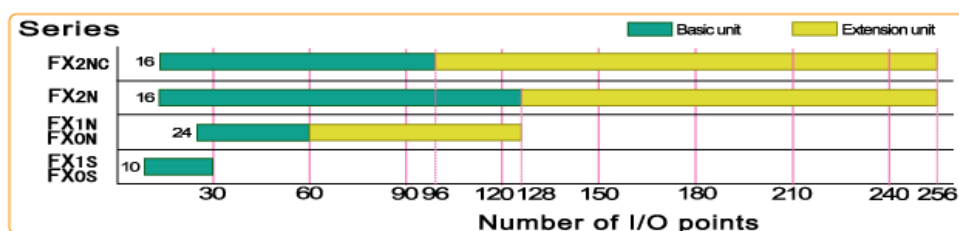
Các CPU loại này sử dụng phần mềm FX-WIN và GPP-WIN, chúng có một số đặc điểm sau:

- + Tính hiệu quả cao
- + Có thể soạn thảo chương trình ở 3 dạng là LAD, STL và FBD
- + kết nối: có khả năng kết nối với tất cả các CPU của Mitsubishi, CC Link, Profibus, AS-i và các mạng khác.
- + sử dụng trong các lĩnh vực điều khiển có số lượng ngõ vào ra tới 256 I/O



CPU họ FX

Dưới đây là phạm vi mở rộng I/O của các CPU họ FX



BÀI 6 : LẬP ĐẶT MÔ HÌNH ĐIỀU KHIỂN BẰNG PLC.

1. MÔ HÌNH ĐIỀU KHIỂN ĐỘNG CƠ SAO – TAM GIÁC

* Yêu cầu công nghệ:

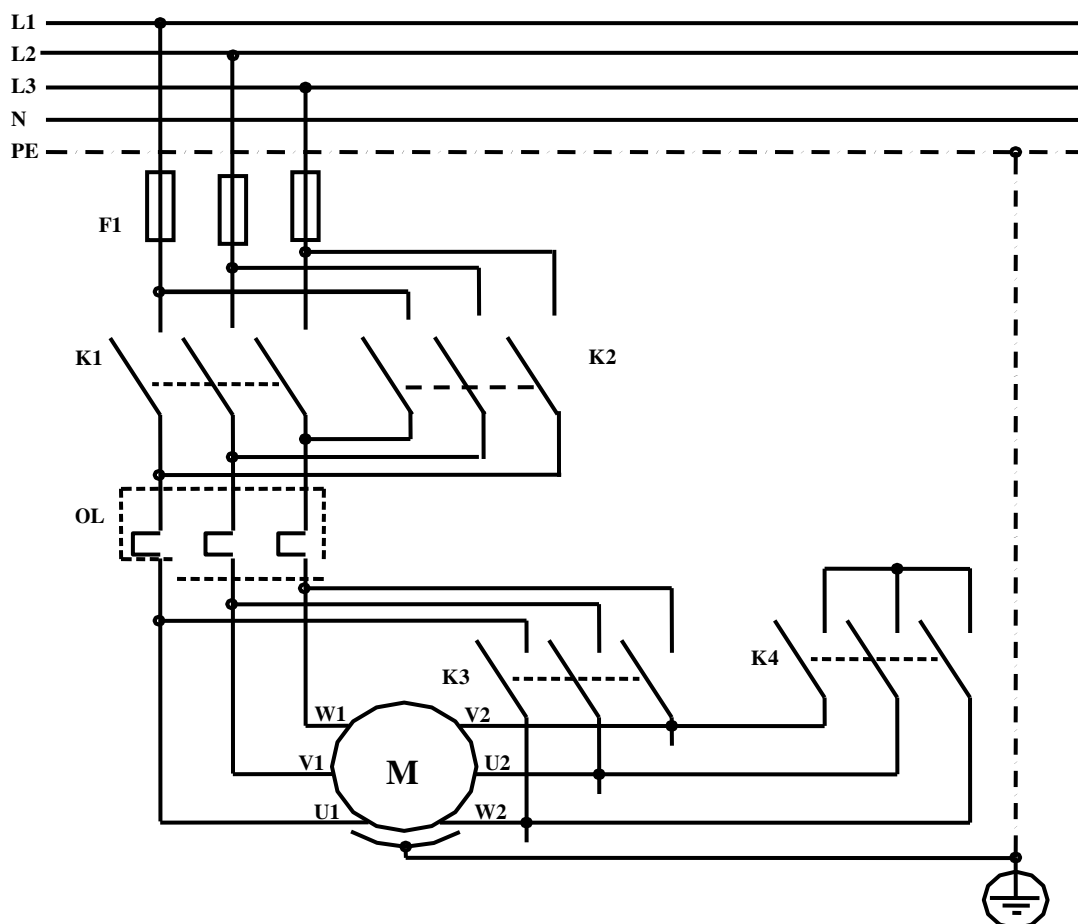
Lập trình PLC chương trình điều khiển Đ_{KĐB} AC 3 pha rotor lồng sóc bằng phương pháp đổi nối sao – tam giác. Có yêu cầu như sau:

- Nhấn FWD động cơ khởi động và quay theo chiều thuận với cuộn dây stator nối hình sao, sau 5 giây động cơ tự động đổi nối sang hình tam giác, động cơ tiếp tục khởi động và làm việc định mức ở chế độ tam giác.
- Nhấn REV, quá trình khởi động và làm việc tương tự với chiều quay ngược lại.
- Nhấn OFF động cơ dừng tự do.

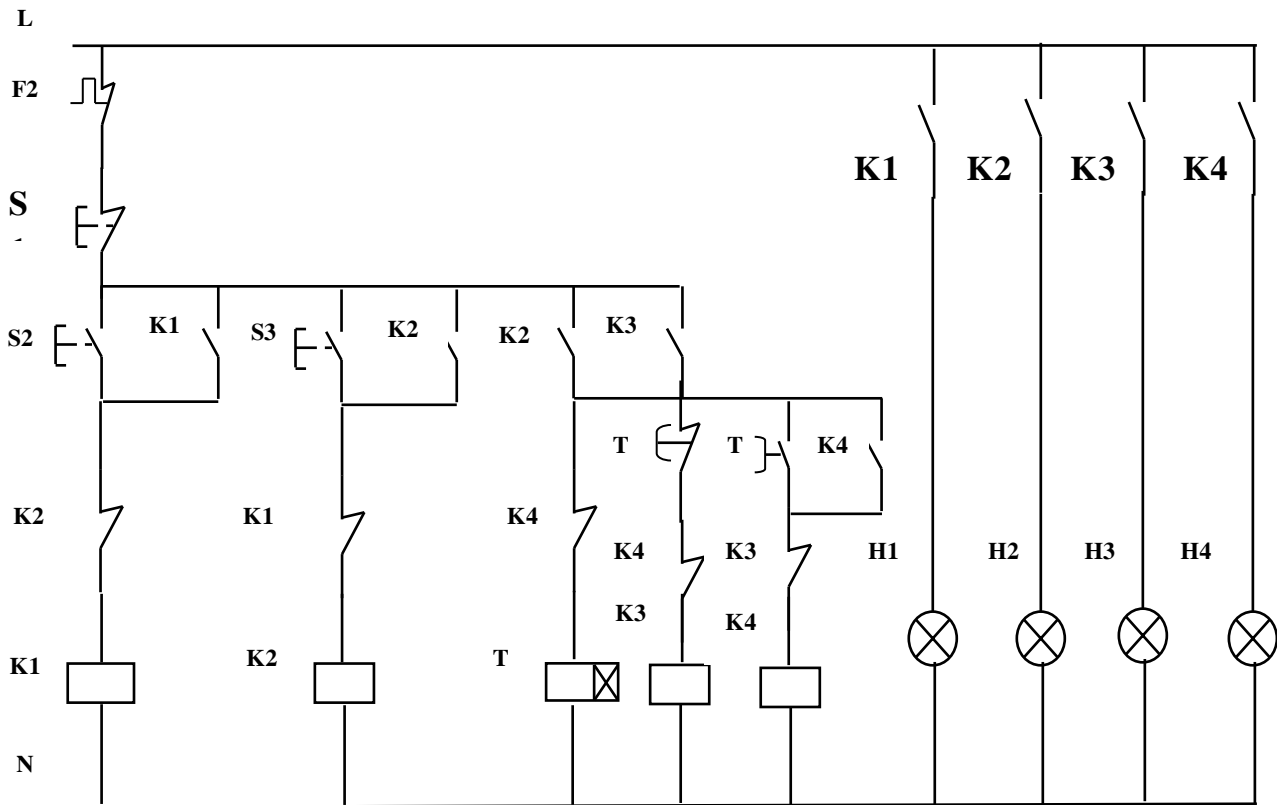
* Nhiệm vụ:

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

* Mạch động lực:



*** Sơ đồ nối cứng:**



*** Mô tả hoạt động:**

Khi nhấn S2, K1 có điện và tự duy trì, đồng thời Rơle thời gian T 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 nối tam giác.

Nhấn S1, động cơ được cắt khỏi nguồn điện. Bây giờ nhấn S3, K2 có điện, đồng thời T và K3 có điện, động cơ chạy trái ở chế độ nối Y. Quá trình lập lại giống như ở trường hợp trên.

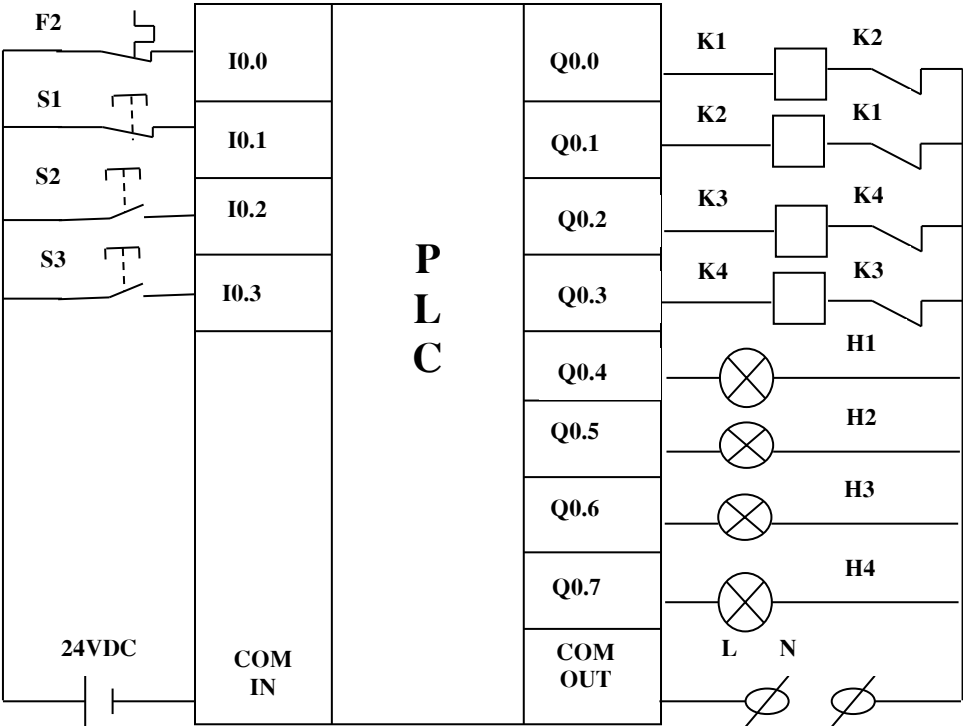
Các khởi động từ K1 và K2 được khóa chéo lẫn nhau.

*** Bảng trạng thái:**

Kí hiệu	Địa chỉ	Mô tả
F2	I0.0	Bộ cắt dòng quá tải
S1	I0.1	Nút nhấn cắt mạch - thường đóng
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	Khởi động từ chạy sao (Y)
K4	Q0.3	Khởi động từ chạy tam giác(Δ)

H1	Q0.4	Đèn báo chế độ chạy phải
H2	Q0.5	Đèn báo chế độ chạy trái
H3	Q0.6	Đèn báo chế độ chạy nối sao
H4	Q0.7	Đèn báo chế độ chạy nối tam giác

*** Nối dây PLC:**



*** Chương trình:**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. MÔ HÌNH ĐIỀU KHIỂN DÂY CHUYỀN SẢN XUẤT LIÊN TỤC

* Yêu cầu công nghệ:

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

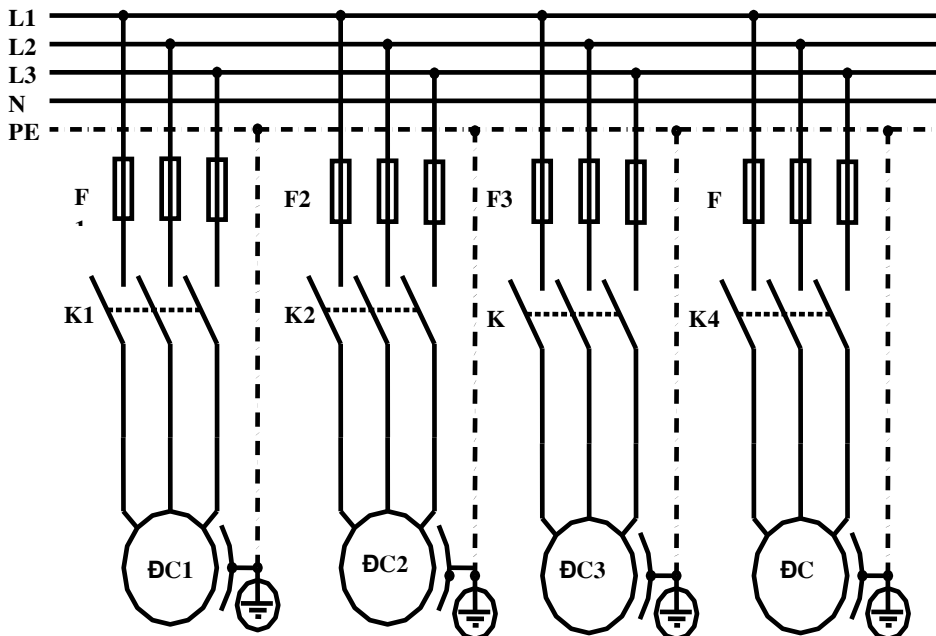
Khi nhấn nút STOP: Hệ thống động cơ dừng từ động cơ đầu dây chuyền đến động cơ cuối dây chuyền lần lượt cách nhau 10 giây.

Hệ thống có bảo vệ sự cố ngắn mạch, quá tải.

* Nhiệm vụ:

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

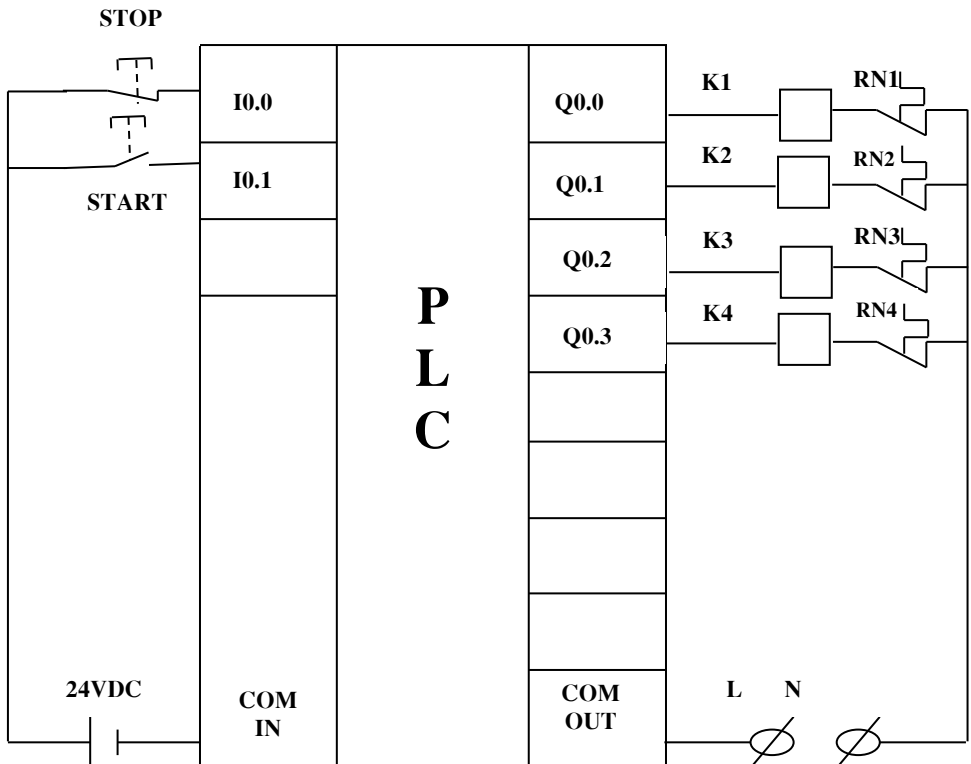
* Mạch động lực:



* Bảng trạng thái:

Xác định ngõ vào/ra		
Ký hiệu	Toán hạng	Mô tả
STOP	I0.0	Dừng hệ thống
START	I0.1	Khởi động hệ thống
K1	Q0.0	Contactơ khống chế động cơ 1
K2	Q0.1	Contactơ khống chế động cơ 2
K3	Q0.2	Contactơ khống chế động cơ 3

*** Nối dây PLC:**



*** Chương trình:**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

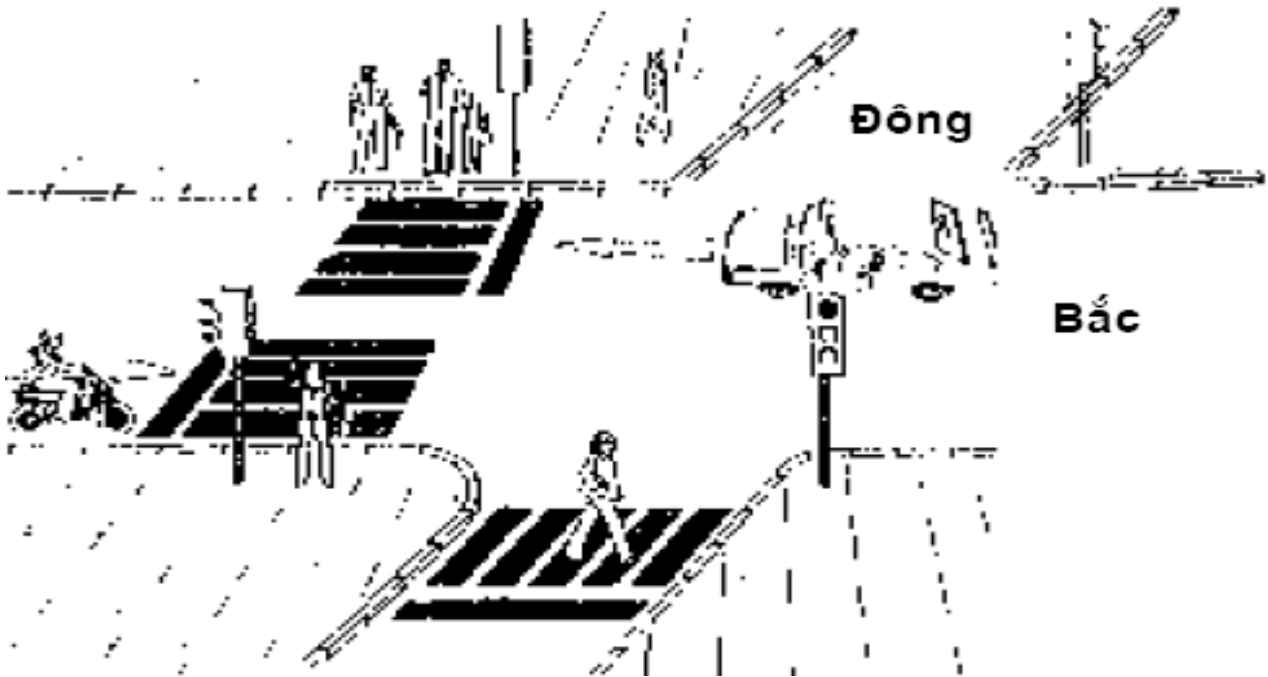
.....

3. MÔ HÌNH ĐÈN GIAO THÔNG CHO MỘT NGÃ TƯ

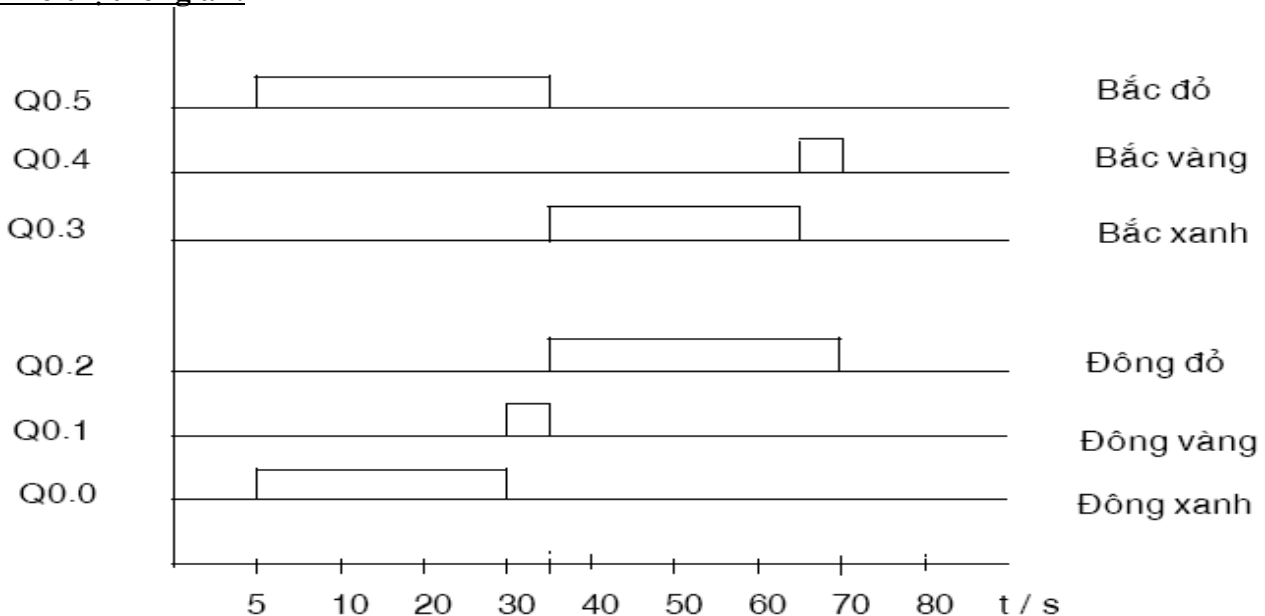
* **Yêu cầu:** Lập trình PLC điều khiển đèn giao thông cho ngã tư, với yêu cầu như sau:

- Đèn xanh sáng trong khoảng thời gian 25 giây.
- Đèn vàng sáng trong khoảng thời gian 5 giây.
- Đèn đỏ sáng trong khoảng thời gian 30 giây.
- Mạch có bảo vệ sự cố ngắn mạch, quá tải.

* **Sơ đồ công nghệ:**



* **Đồ thị thời gian:**



* **Nhiệm vụ:**

- Vẽ mạch động lực.
- Lập bảng xác lập ngõ vào/ra.
- Vẽ sơ đồ nối dây PLC.

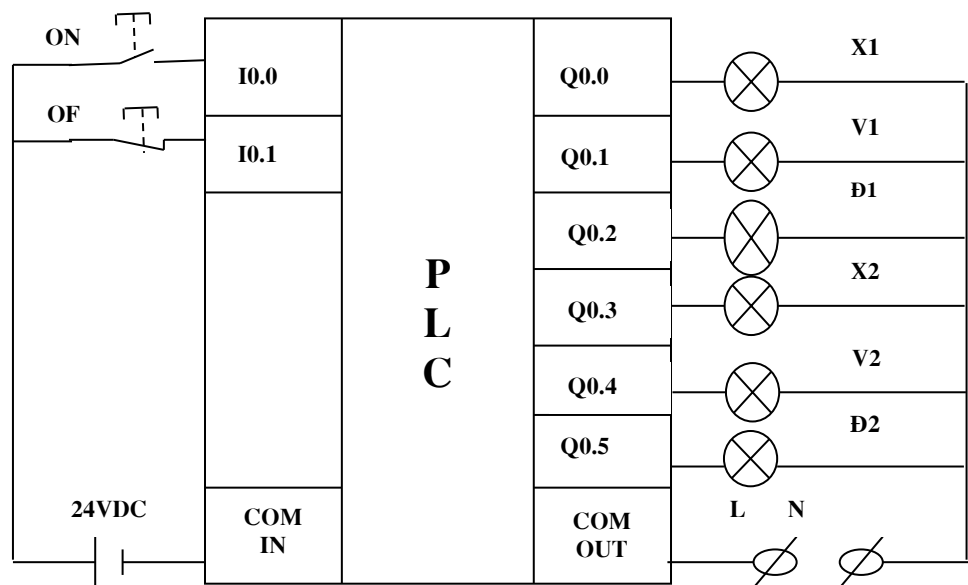
- Viết chương trình PLC S7-200 theo ngôn ngữ LAD trên phần mềm STEP7-Microwin V3.2 hoặc V4.0.

- Kết nối thiết bị ngoại vi, download, vận hành chương trình trên S7-200, CPU 224.

*** Bảng trạng thái:**

Kí hiệu	Địa chỉ	Mô tả
ON	I0.0	Khởi động hệ thống
OFF	I0.1	Dừng hệ thống
X1	Q0.0	Đèn báo xanh hướng đông
V1	Q0.1	Đèn báo vàng hướng đông
Đ1	Q0.2	Đèn báo đỏ hướng đông
X2	Q0.3	Đèn báo xanh hướng bắc
V2	Q0.4	Đèn báo vàng hướng bắc
Đ2	Q0.5	Đèn báo đỏ hướng bắc

*** Nối dây PLC:**



*** Chương trình:**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

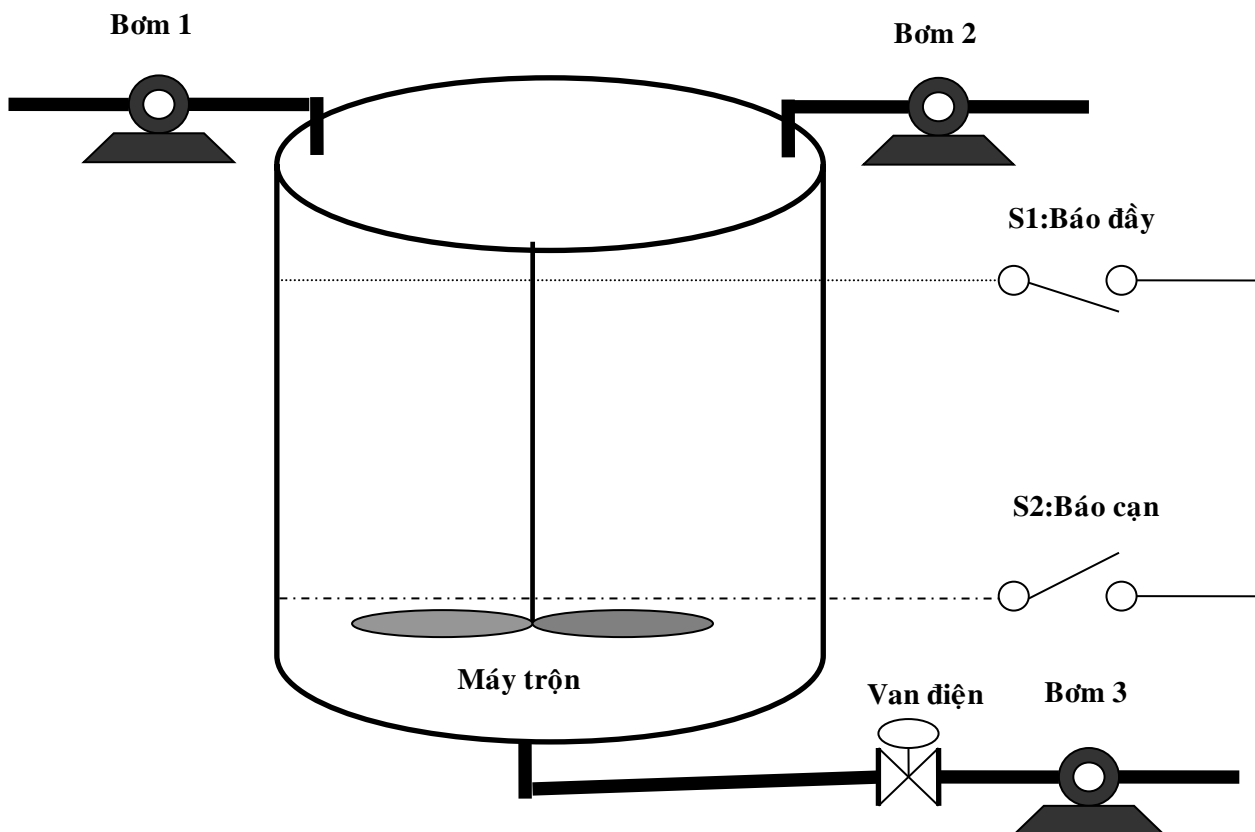
.....

4. MÔ HÌNH TRỘN HOÁ CHẤT

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

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

* **Sơ đồ công nghệ:**



* **Mô tả:**

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

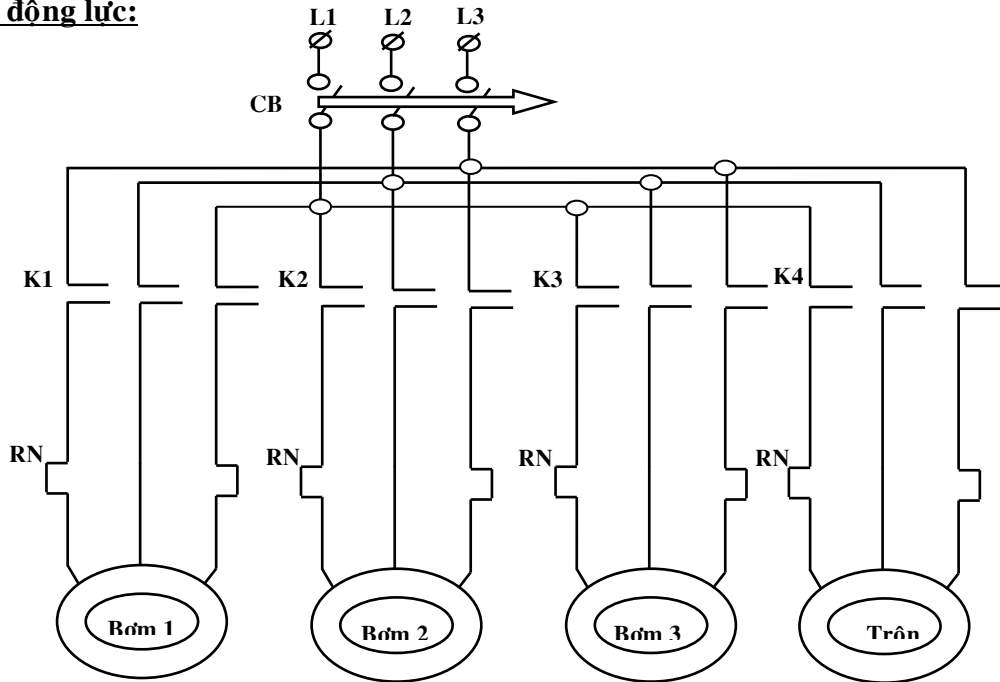
Quá trình được làm việc như sau: Khi nhấn nút START thì bơm 1 và bơm 2 điều khiển qua (Q0.0) và (Q0.1) hoạt động để đưa hai loại hoá chất khác nhau vào bình. Khi dung dịch trong bình đã đạt mức cực đại thì cảm biến báo đầy (I0.4) tác động dừng hai bơm và bắt đầu quá trình trộn, quá trình này được điều khiển bởi động cơ trộn (Q0.2) và thời gian trộn cần thiết là 5 giây. Sau khi trộn xong sản phẩm được đưa ra qua van (Q0.3) và bơm 3 (Q0.4). Khi xả

hết sản phẩm thì cảm biến báo cạn (I0.5) tác động đóng van và dừng bơm 3. Đồng thời lúc đó bơm 1 và bơm 2 tự động hoạt động trở lại cho chu kỳ mới, chu trình lặp lại 5 lần thì dừng luôn. Trong quá trình hoạt động có sự cố hoặc nhấn nút STOP thì hệ thống dừng ngay.

*** Nhiệm vụ:**

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

*** Mạch động lực:**

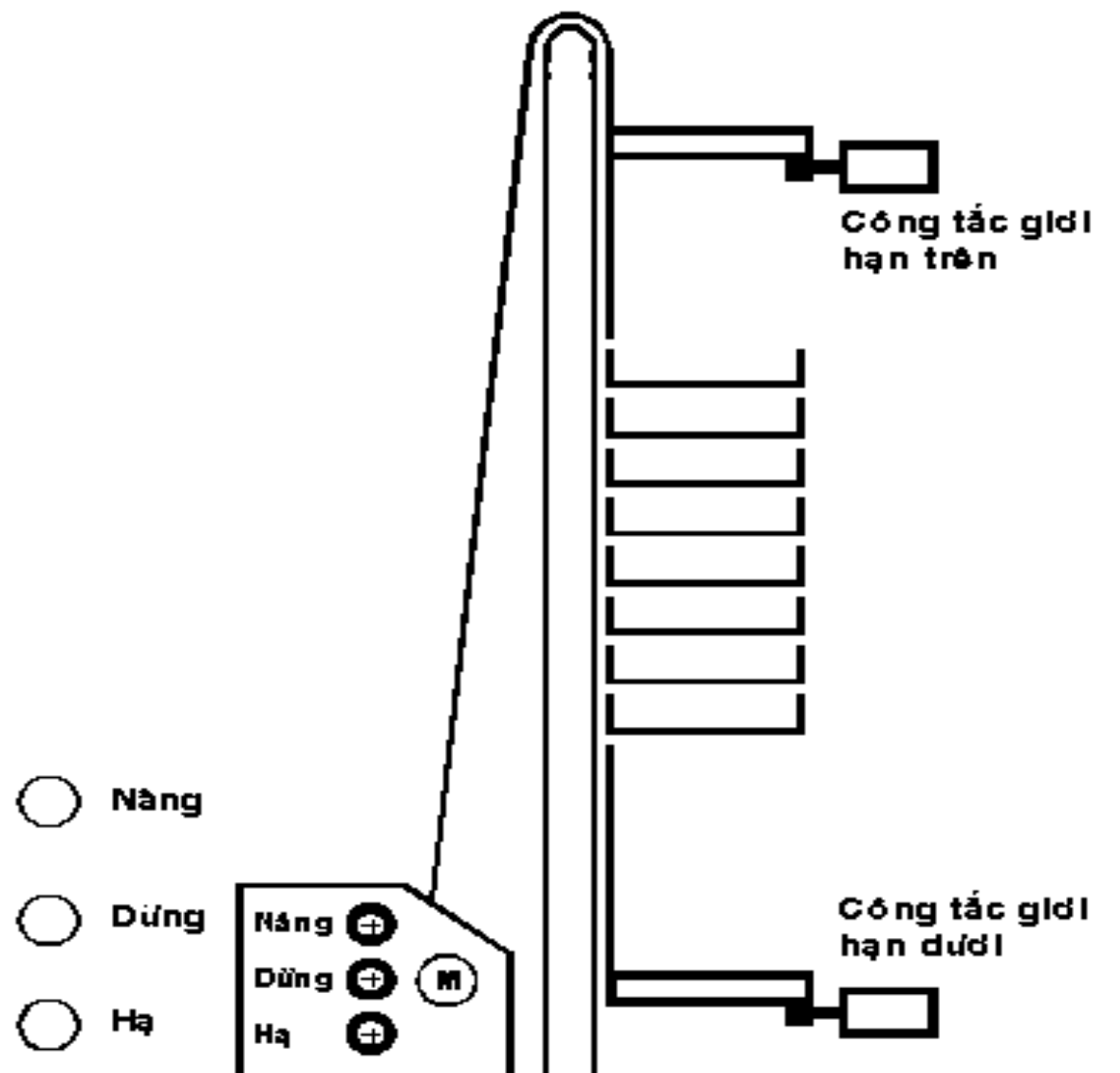


*** Bảng trạng thái:**

Kí hiệu	Địa chỉ	Mô tả
START	I0.0	Khởi động hệ thống
STOP	I0.1	Dừng hệ thống
CB	I0.2	Bảo vệ ngắn mạch
RN1234	I0.3	Bảo vệ quá tải
S1	I0.4	Cảm biến báo đầy
S2	I0.5	Cảm biến báo cạn
K1	Q0.0	Điều khiển bơm 1
K2	Q0.1	Điều khiển bơm 2
K3	Q0.2	Điều khiển động cơ trộn
K4	Q0.3	Điều khiển bơm 3
V	Q0.4	Điều khiển van xả

5. MÔ HÌNH THANG MÁY XÂY DỰNG

* Sơ đồ công nghệ.



* 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.

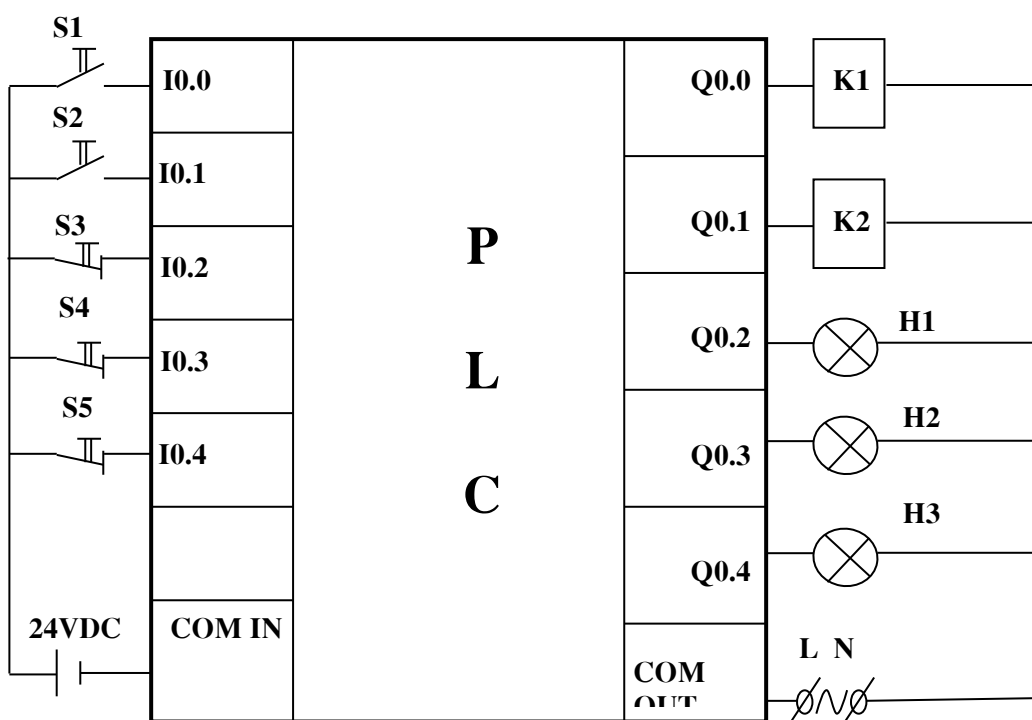
Các trạng thái nâng lên, hạ xuống hoặc dừng đều được thông báo bằng đèn báo.

* Bảng trạng thái:

Kí hiệu	Địa chỉ	Mô tả
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	Điều khiển gàu chạy lên
K2	Q0.1	Điều khiển gàu chạy xuống
H1	Q0.2	Đèn báo nâng
H2	Q0.3	Đèn báo hạ
H3	Q0.4	Đèn báo dừng

*** Nối dây PLC:**



*** Chương trình:**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TÀI LIỆU THAM KHẢO

1. Giáo trình PLC S7-200
Ngô Quang Hà- Trần Văn Trọng - Trung Tâm Việt Đức
2. Tài liệu thực hành PLC S7-200 - Trung Tâm Việt Đức
3. Hướng dẫn thiết kế mạch và lập trình PLC
Trần Thế San (biên dịch)– NXB Đà Nẵng-2005
4. Điều khiển logic lập trình PLC
Tăng Văn Mùi(biên dịch)– NXB Thống Kê-2006
5. Tự động hoá với simatic S7-200 – NXB Nông Nghiệp Hà Nội
Nguyễn Doãn Phước
Phan Xuân Minh
6. Speicher
7. Automatisieren mit SPS Theory und Praxis.
8. S7-200 Programmable Controller System Manual -
Siemens C79000-G7076-C230-02
9. PLC Omron Catalogs & manuals V5 (8-01)
10. Mitsubishi-Sổ tay hướng dẫn lập trình họ FX
JY- 922d48301