

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM
KHOA ĐIỆN TỬ

BÀI GIẢNG:

ĐIỀU KHIỂN LẬP TRÌNH 1

BIÊN SOẠN:

GV ThS Nguyễn Tấn Đời

TP HỒ CHÍ MINH, NĂM 2007

MỤC LỤC

	TRANG
CHƯƠNG 1: MỞ ĐẦU	01
1.1 GIỚI THIỆU.	01
1.2 LOGIC BẬC THANG. Ladder Logic	01
1.3 LẬP TRÌNH. Programming	04
1.4 KẾT NỐI PLC. PLC Connections	06
1.5 NGÕ VÀO LOGIC BẬC THANG. Ladder Logic Inputs	06
1.6 NGÕ RA LOGIC BẬC THANG Ladder Logic Outputs	07
CHƯƠNG 2: CẤU TRÚC VÀ HOẠT ĐỘNG CỦA PLC	09
2.1 CẤU TRÚC PHẦN CỨNG PLC. PLC Hardware	09
2.1.1 Giới Thiệu.	09
2.1.2 Ngõ Vào và Ngõ Ra.	10
2.1.3 Relay.	16
2.1.4 Sơ Đồ Nối Dây.	17
2.2 HOẠT ĐỘNG CỦA PLC. PLC Operation	18
2.2.1 Giới Thiệu.	18
2.2.2 Hoạt Động Tuần Tự.	19
2.2.3 Trạng Thái PLC.	20
2.2.4 Bộ Nhớ.	20
CHƯƠNG 3: CẢM BIẾN	22
3.1 GIỚI THIỆU.	22
3.2 CẢM BIẾN DÂY NỐI. Sensor Wiring	22
3.2.1 Công Tác.	22
3.2.2 TTL.	23
3.2.3 Rút Dòng và Cấp Dòng. Sinking/Sourcing	23
3.2.4 Tiếp điểm Relay Solid State Relay.	23
3.3 CẢM BIẾN TIÊM CẬN. Presence Detection	24
3.3.1 Công Tác Tiếp Xúc.	24
3.3.2 Công tác Lưỡi Gà.	24
3.3.3 Cảm Biến Quang.	25
3.3.4 Cảm Biến Điện Dung.	25
3.3.5 Cảm Biến Điện Cảm.	26
3.3.6 Dòng Chất lỏng.	27
CHƯƠNG 4: THIẾT BỊ CHẤP HÀNH	28
4.1 GIỚI THIỆU.	28
4.2 CUỘN DÂY. Solenoid	28
4.3 VAL Valve	28
4.4 XY LẠNH Cylinder	29
4.5 THỦY LỰC. Hydraulic	30
4.6 KHÍ NÉN. Pneumatic	31
4.7 ĐỘNG CƠ Motor	31
CHƯƠNG 5: THIẾT KẾ CHƯƠNG TRÌNH THEO LƯU ĐỒ	35
5.1 GIỚI THIỆU.	35
5.2 PHƯƠNG PHÁP BLOCK LOGIC.	37
5.3 PHƯƠNG PHÁP SEQUENCE BIT.	42

CHƯƠNG 6: PLC S7 – 200	46
6.1 CẤU TRÚC PHẦN CỨNG.	46
6.1.1 Đặc điểm chung.	46
6.1.2 Các đèn báo.	46
6.1.3 Các ngõ vào.	46
6.1.4 Các ngõ ra.	46
6.1.5 Nguồn cung cấp.	47
6.1.6 Cổng truyền thông.	47
6.1.7 Các module mở rộng.	48
6.2 NGUYÊN LÝ HOẠT ĐỘNG.	49
6.2.1 Đơn vị xử lý trung tâm CPU.	49
6.2.2 Hệ thống BUS.	49
6.2.3 Bộ nhớ.	50
6.3 CẤU TRÚC BỘ NHỚ.	50
6.3.1 Phân chia bộ nhớ.	50
6.3.2 Vùng dữ liệu.	51
6.3.3 Vùng đối tượng.	51
6.3.4 Phương thức truy cập bộ nhớ.	52
6.4 PHƯƠNG PHÁP LẬP TRÌNH.	53
6.4.1 Quan hệ giữa chương trình và các ngõ vào/ra	53
6.4.2 Khái niệm về ngôn ngữ lập trình	53
6.4.3 Phương pháp STL.	54
6.4.4 Phương pháp LAD.	55
6.4.5 Phương pháp FBD.	56
CHƯƠNG 7: TẬP LỆNH S7 – 200	57
7.1 NHÓM LỆNH VỀ TIẾP ĐIỂM.	57
7.2 NHÓM LỆNH VỀ TIMER VÀ COUNTER.	61
7.2.1 Lệnh Timer.	61
7.2.2 Lệnh Counter.	66
7.3 NHÓM LỆNH SO SÁNH.	69
7.4 NHÓM LỆNH VỀ CÔNG LOGIC.	71
7.4.1 Lệnh AND.	71
7.4.2 Lệnh OR.	72
7.5 NHÓM LỆNH VỀ CÁC PHÉP TOÁN LOGIC.	73
7.6 NHÓM LỆNH DI CHUYỂN VÀ BIẾN ĐỔI DỮ LIỆU.	75
7.6.1 Lệnh Di chuyển.	75
7.6.2 Lệnh Tăng Giảm.	77
7.6.3 Lệnh Chuyển đổi.	81
7.7 LỆNH VỀ ĐỒNG HỒ THỜI GIAN THỰC.	83
BÀI TẬP	86
PHỤ LỤC:	96
PHẦN MỀM LẬP TRÌNH Step 7 MicroWIN 3.2/4.0	
PHẦN MỀM MÔ PHỎNG S7-200 Simulator 2.0	

ĐIỀU KHIỂN LẬP TRÌNH 1**CHƯƠNG 1: MỞ ĐẦU****Topics:**

- PLC History
- Ladder Logic and Relays
- PLC Programming
- PLC Operation
- An Example

Objectives:

- Know general PLC issues
- To be able to write simple ladder logic programs
- Understand the operation of a PLC

1.1 GIỚI THIỆU.

Kỹ thuật điều khiển đã được phát triển trong thời gian rất lâu. Trước kia việc điều khiển hệ thống chủ yếu do con người thực hiện. Gần đây, việc điều khiển được thực hiện nhờ vào các ứng dụng của ngành điện, thực hiện bằng việc đóng ngắt tiếp điểm relay. Các relay sẽ cho phép đóng ngắt công suất không cần dùng công tắc cơ khí. Ta thường sử dụng relay để tạo nên các thao tác điều khiển đóng ngắt logic đơn giản. Sự xuất hiện của máy tính điện tử đã tạo một bước tiến mới trong điều khiển – Kỹ thuật điều khiển lập trình PLC. PLC xuất hiện vào những năm 1970 và nhanh chóng trở thành sự lựa chọn cho việc điều khiển sản xuất.

PLC có nhiều lợi thế trong nhà máy, bao gồm:

- Giảm giá thành đối với các hệ thống phức tạp.
- Mềm dẻo và dễ thay thế khi cần thay đổi hệ thống điều khiển.
- Khả năng kết hợp với máy tính cho phép điều khiển các hệ thống tinh vi.
- Khả năng hỗ trợ xử lý sự cố làm cho việc lập trình dễ dàng và nhanh chóng.
- Kết cấu chắc chắn và chính xác làm cho hệ thống hoạt động ổn định và tin cậy.

1.2 LOGIC BẬC THANG.**Ladder Logic**

Logic bậc thang là phương pháp lập trình chính cho PLC. Logic bậc thang được phát triển để thay thế cho việc điều khiển bằng logic relay. Do đã có sơ đồ điều khiển bằng relay nên khi chọn Logic bậc thang làm phương pháp lập trình chính cho PLC thì việc huấn luyện cho các kỹ sư và người sử dụng sẽ giảm đi rất nhiều.

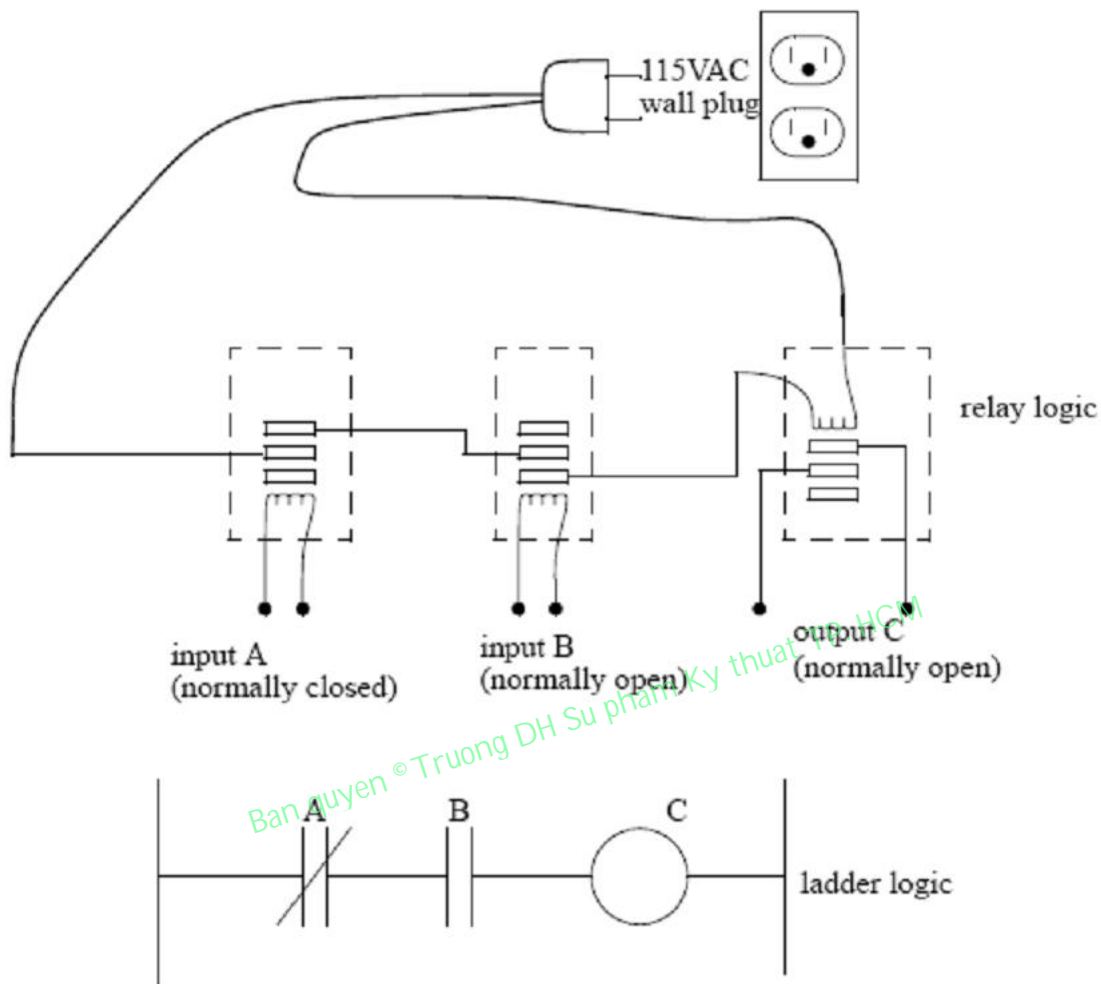
Các hệ thống điều khiển hiện đại ngày nay vẫn còn sử dụng relay, nhưng chúng không được dùng để tạo ra mức logic mà hoạt động như một thiết bị điện tử dùng để đóng mở tiếp điểm.

Các relay được dùng để đóng mở các nguồn điện công suất lớn dựa vào nguồn năng lượng nhỏ, vẫn giữ cách ly các nguồn này.

Hệ thống điều khiển đơn giản có sử dụng relay được minh họa trên hình 1.1. Relay bên trái sử dụng tiếp điểm thường đóng cho dòng điện qua đến khi có điện áp cấp vào đầu dây A. Relay ở giữa sử dụng tiếp điểm thường hở nên không cho dòng điện qua đến

ĐIỀU KHIỂN LẬP TRÌNH 1

khi đầu dây B có điện. Nếu dòng điện qua 2 tiếp điểm của relay A và B rồi vào cuộn dây của relay C thì sẽ đóng tiếp điểm đầu ra C.



Hình 1.1: Hệ thống điều khiển dùng relay

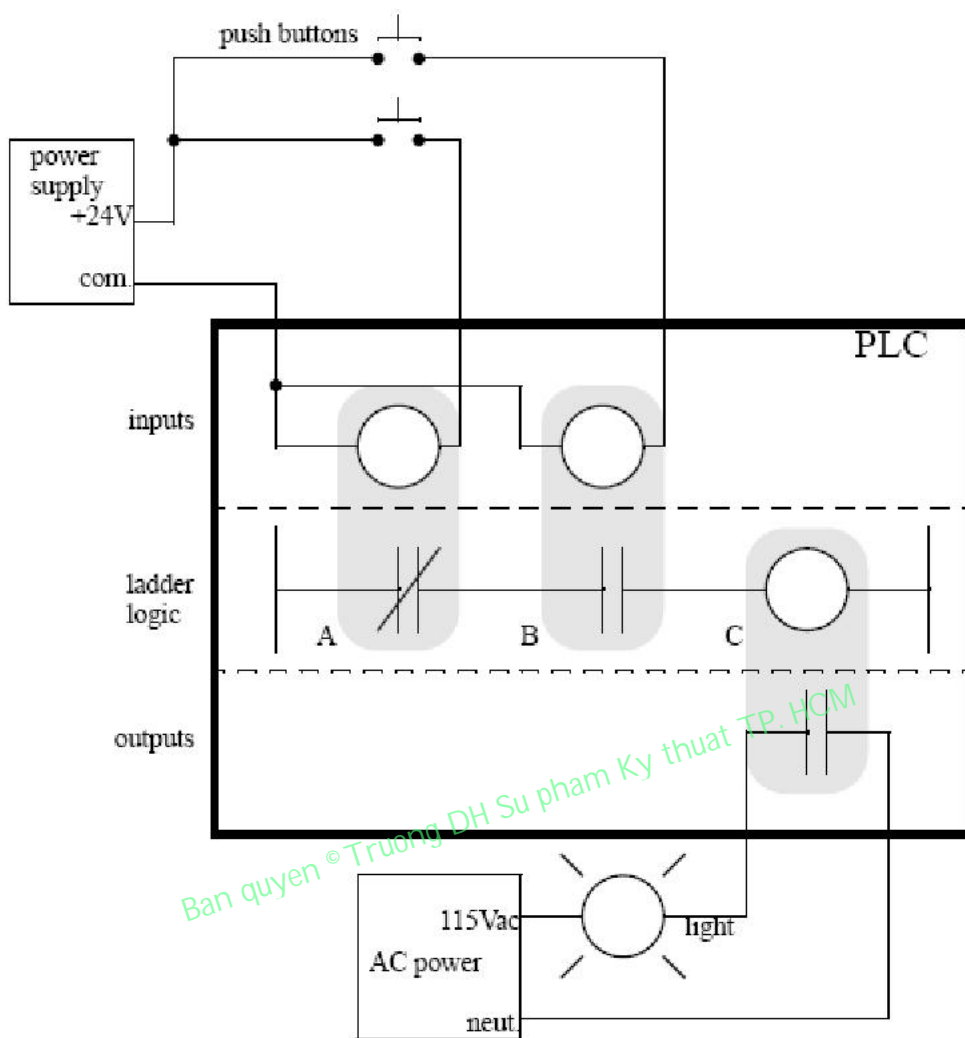
Mạch điện được vẽ lại ở dạng sơ đồ logic bậc thang bên dưới trong hình 1.1.

Trạng thái logic được đọc là: C đóng nếu A mở và B đóng.

Hình vẽ này không phải là toàn bộ hệ thống điều khiển, chỉ là sơ đồ logic. Khi xem xét một PLC, ngoài sơ đồ logic còn có các ngõ vào/ra, minh họa trong hình 1.2.

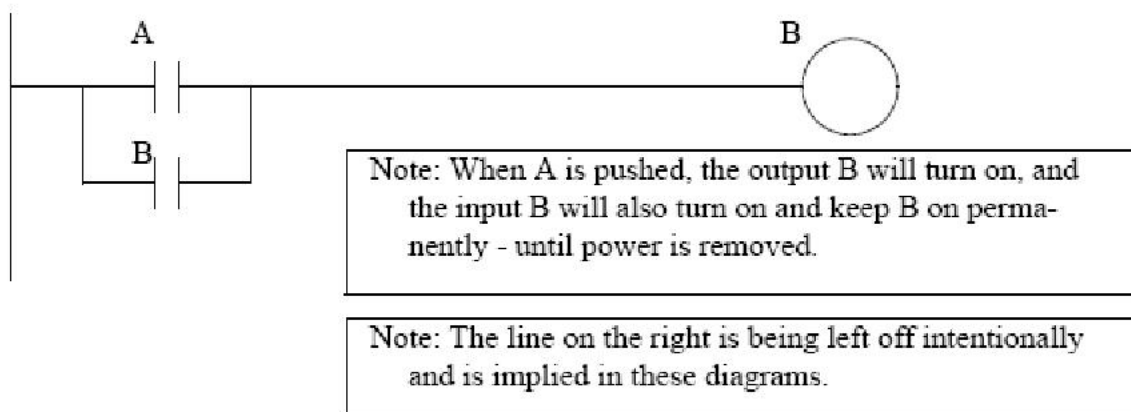
Có 2 ngõ vào nút nhấn, giả sử sẽ tác động các cuộn dây relay bên trong PLC, làm ngõ ra relay đóng cấp nguồn 115VAC cho đèn sáng. Lưu ý là với các PLC thực tế, ngõ vào không sử dụng relay, nhưng ngõ ra có thể sử dụng relay. Logic bậc thang trong PLC thường là các chương trình do người dùng viết và hiệu chỉnh trên máy tính. Cả 2 ngõ vào PLC là nút nhấn thường hở, nhưng logic bậc thang bên trong PLC có thể sử dụng một thường đóng và một thường hở, không nhất thiết logic bậc thang này phải phù hợp với trạng thái các ngõ vào/ ra.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 1.2: PLC có sử dụng relay

Một số relay có nhiều ngõ ra nên có thể sử dụng một ngõ ra relay như một ngõ vào tức thời, tạo thành mạch duy trì như trong hình 1.3.



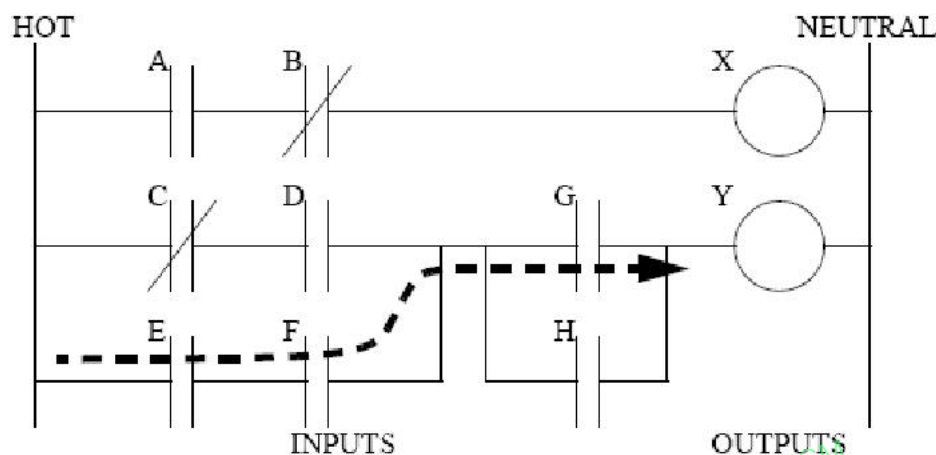
Hình 1.3: Mạch duy trì

Trong mạch này, dòng điện có thể chạy qua cả 2 nhánh là các công tắc A và B. Ngõ vào B chỉ đóng khi ngõ ra B có điện. Nếu B mất điện, đóng ngõ vào A thì B sẽ có điện, làm ngõ vào B đóng. Khi đó cho dù ngõ vào A mở ra nhưng B vẫn có điện nhờ ngõ vào B duy trì. Sau khi ngõ vào B đã đóng thì không thể tắt điện B được.

ĐIỀU KHIỂN LẬP TRÌNH 1

1.3 LẬP TRÌNH. Programming

Các PLC trước kia được lập trình bằng kỹ thuật sử dụng các sơ đồ nối dây relay. Do đó không cần phải hướng dẫn nhiều cho các thợ điện, kỹ thuật viên, kỹ sư cách lập trình trên máy tính, nên đây cũng là kỹ thuật lập trình thông dụng cho PLC ngày nay. Xét ví dụ trên hình 1.4.



Hình 1.4: Sơ đồ logic bậc thang đơn giản

Giả sử nguồn nối với đường dây bên trái HOT, gọi là dây nóng, bên phải là dây trung tính. Sơ đồ có 2 nhánh, mỗi nhánh là một tổ hợp các ngõ vào và ngõ ra.

Nếu các ngõ vào đóng hoặc mở thì công suất sẽ chạy từ dây nóng qua các ngõ vào, kết hợp với dây trung tính cấp điện cho ngõ ra.

Ngõ vào PLC có thể được kết nối với các cảm biến hoặc công tắc. Ngõ ra PLC sẽ nối với các thiết bị trung gian đóng ngắt các tải bên ngoài như đèn, động cơ.

Trong nhánh trên, công tắc A thường hở và B thường đóng, nghĩa là nếu A đóng và B mở thì dòng điện sẽ chạy qua công tắc A và B tác động đến ngõ ra X, các trạng thái khác của A và B sẽ làm X mất điện.

Giải thích tương tự cho hoạt động của nhánh bên dưới.

Có nhiều phương pháp lập trình khác nhau cho PLC. Một trong những kỹ thuật đó là sử dụng lệnh gọi nhớ. Các lệnh này xuất phát trực tiếp từ sơ đồ logic bậc thang và được nhập vào PLC bằng một thiết bị lập trình. Xét ví dụ trên hình 1.5.

Trong ví dụ này, các lệnh được đọc lần lượt từ trên xuống dưới.

Dòng 00000 có lệnh LDN (input load not) cho ngõ vào 00001. Lệnh này xác định một ngõ vào nối với PLC, nếu nó mở thì sẽ tạo một giá trị 1, và ngược lại sẽ tạo giá trị 0.

Dòng tiếp theo 00001 sử dụng lệnh LD (input load) để xác định giá trị ngõ vào, nếu ngõ vào này mở thì tạo giá trị 0 và ngược lại sẽ tạo giá trị 1.

Lệnh AND sử dụng lại 2 số được tạo ra bên trên, nếu chúng cùng bằng 1 thì sẽ tạo ra giá trị 1, còn có một ngõ vào bằng 0 thì tạo giá trị 0. Giá trị này sẽ thay thế cho 2 kết quả trên và lúc này chỉ còn một kết quả của lệnh AND được giữ lại.

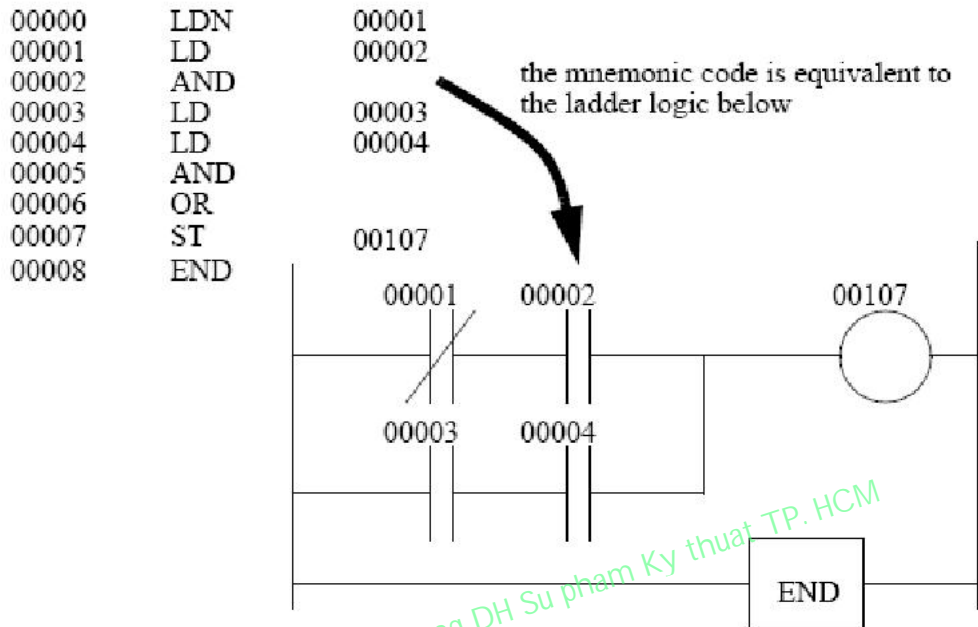
Quá trình này sẽ lặp lại với các hàng 00003 và 00004, sau khi thực hiện xong sẽ có 3 số được lưu lại.

Lệnh AND trong hàng 00005 sẽ AND kết quả của hàng 00003 và 00004, tạo ra 1 kết quả mới.

Lệnh OR trong hàng 00006 sẽ OR kết quả của 2 lệnh AND ở các hàng trên. Lúc này chỉ còn 1 kết quả lưu lại.

ĐIỀU KHIỂN LẬP TRÌNH 1

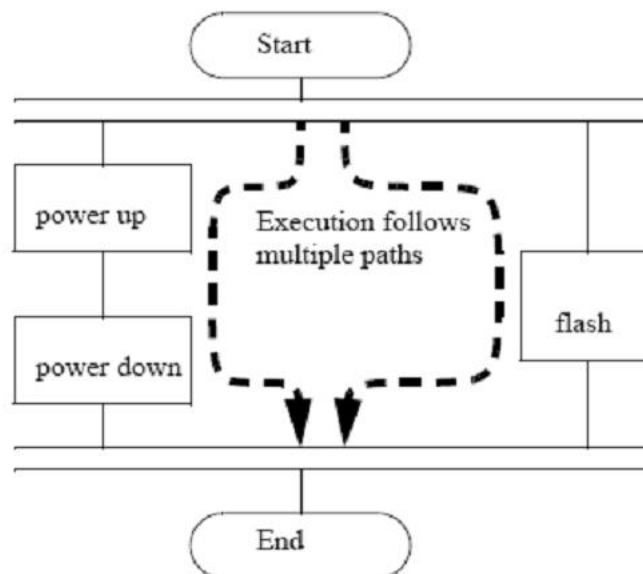
Lệnh ST (store output) trong hàng 00007 sẽ lưu lại kết quả sau cùng. Nếu kết quả này bằng 1 thì ngõ ra 00107 sẽ tác động, ngược lại ngõ ra này không tác động. Chương trình logic bậc thang trong hình 1.5 tương đương với chương trình gọi nhớ vừa phân tích trên. Thậm chí nếu ta đã lập trình cho PLC bằng logic bậc thang thì nó có thể sẽ được chuyển về dạng gọi nhớ trước khi được PLC sử dụng.



Hình 1.5: Chương trình gọi nhớ và Sơ đồ logic bậc thang tương đương

Ngoài ra, đồ thị hàm dãy SFC (Sequential Function Chart) được sử dụng để hỗ trợ việc lập trình cho những hệ thống phức tạp hơn. SFC tương tự như lưu đồ nhưng hiệu quả hơn rất nhiều. Xét ví dụ trong hình 1.6.

Ví dụ này thực hiện 2 việc khác nhau. Để đọc lưu đồ, ta bắt đầu từ vị trí Start. PLC sẽ bắt đầu sau 2 đường ngang bên dưới, thực hiện độc lập và cùng lúc trên 2 nhánh trái phải. Nhánh bên trái có 2 hàm là power up và power down, nhánh phải có hàm flash. Mỗi hàm là 1 chương trình logic bậc thang. Phương pháp này khác với lưu đồ ở chỗ nó không thực hiện theo 1 đường từ khối này đến khối kia.



Hình 1.6: Lưu đồ hàm dãy

ĐIỀU KHIỂN LẬP TRÌNH 1

Cuối cùng, chương trình dạng văn bản được phát triển và được xem là cách lập trình hiện đại nhất. Cấu trúc nó tương tự như ngôn ngữ Basic.

Xét ví dụ sau:

```
N7:0 :=0;
REPEAT
N7:0 :=N7:0 +1;
UNTIL N7:0 >=10
END_REPEAT;
```

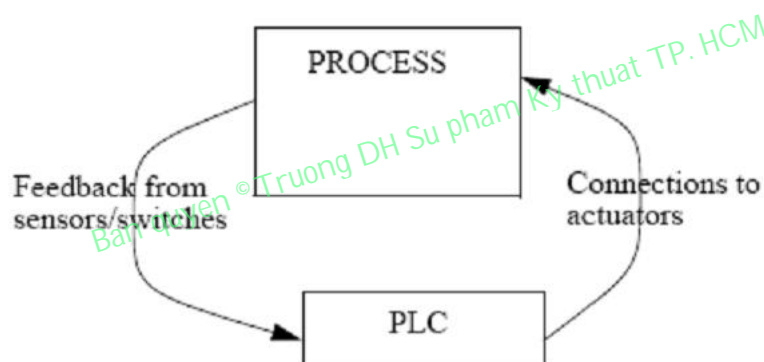
Ví dụ này sử dụng vùng nhớ N7:0 của PLC. Đây là vùng nhớ chứa các số nguyên.

Lệnh đầu tiên đặt các giá trị vùng nhớ về 0.

Các lệnh tiếp theo sẽ tăng giá trị vùng nhớ lên 1 đơn vị, khi giá trị này lớn hơn hoặc bằng 10 thì thoát khỏi vòng lặp.

1.4 KẾT NỐI PLC. PLC Connections

Khi sử dụng PLC để điều khiển 1 quá trình nào đó, ta sử dụng các cảm biến nối với ngõ vào PLC, ngõ ra PLC sẽ điều khiển các thiết bị chấp hành, như hình 1.7.



Hình 1.7: Kết nối PLC

Đây là quá trình xử lý thực, thay đổi liên tục theo thời gian. Các thiết bị chấp hành sẽ làm hệ thống thay đổi sang các trạng thái mới, có nghĩa là hệ thống sẽ được giới hạn điều khiển bởi các cảm biến đầu vào. Nếu 1 ngõ vào không tác động thì bộ điều khiển không thể nhận biết được trạng thái hệ thống.

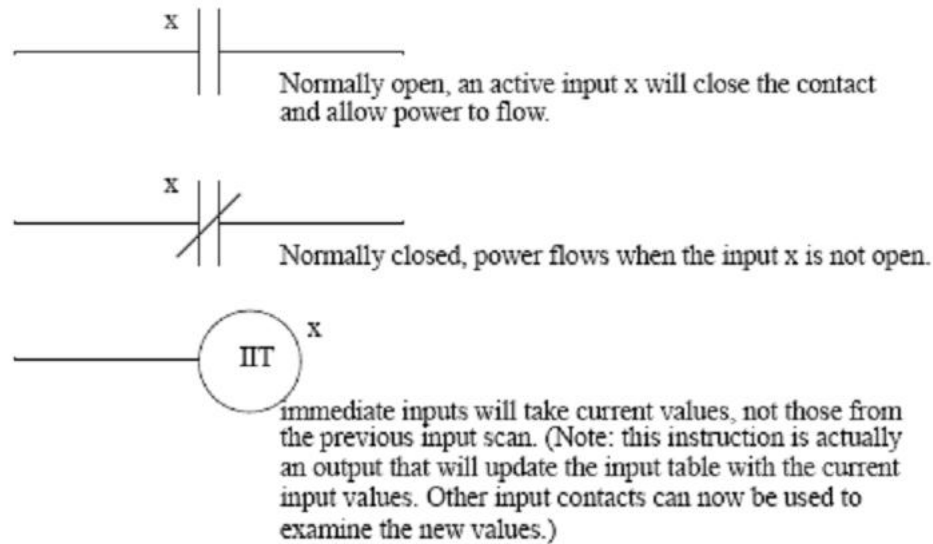
Vòng điều khiển này là 1 chu kỳ liên tục của PLC, gồm việc đọc các dữ liệu đầu vào, thực hiện logic bậc thang và làm thay đổi ngõ ra theo ngõ vào.

1.5 NGÕ VÀO LOGIC BẬC THANG. Ladder Logic Inputs

Các ngõ vào của PLC được biểu diễn dễ dàng trong chương trình logic bậc thang.

Xét hình 1.8, có 3 loại ngõ vào gồm 2 công tắc thường hở và thường đóng và 1 hàm IIT (Immediate Input). Hàm IIT cho phép đọc ngõ vào ngay sau khi nó được quét, trong khi logic bậc thang vẫn đang quét. Điều này cho phép logic bậc thang xác định các giá trị đầu vào thường xuyên hơn trong mỗi chu kỳ.

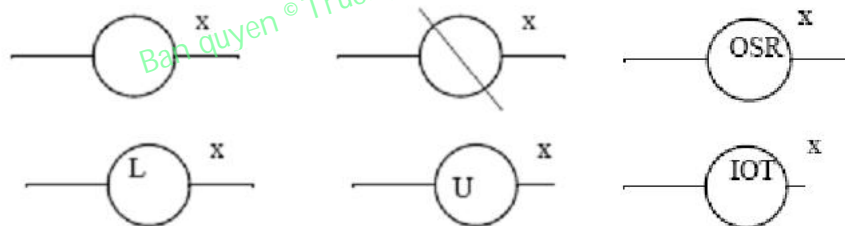
ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 1.8: Ngõ vào PLC

1.6 NGÕ RA LOGIC BẬC THANG Ladder Logic Outputs

Trong logic bậc thang, có nhiều loại ngõ ra khác nhau nhưng chúng không phù hợp cho tất cả các PLC. Một số ngõ ra kết nối bên ngoài PLC, nhưng phần lớn chúng sử dụng các vùng nhớ bên trong PLC. Có 6 loại ngõ ra trình bày trong hình 1.9.



Hình 1.8: Ngõ ra PLC

Ngõ ra X-OSR (one shot relay) sẽ đóng trong 1 vòng quét khi phát hiện trạng thái ngõ vào đóng.

Khi cuộn dây L được cấp điện nó sẽ chuyển X sang trạng thái đóng cho đến khi cuộn dây U được cấp điện. Hoạt động này tương tự 1 FF, và nó vẫn giữ trạng thái cho dù ta tắt PLC.

Một số PLC cho phép sử dụng lệnh xuất ngõ ra tức thời IOT (Immediate Output) mà không cần chờ quét xong chương trình.

TÓM TẮT:

- Normally open and closed contacts.
- Relays and their relationship to ladder logic.
- PLC outputs can be inputs, as shown by the seal in circuit.
- Programming can be done with ladder logic, mnemonics, SFCs, and structured text.
- There are multiple ways to write a PLC program.

ĐIỀU KHIỂN LẬP TRÌNH 1

VÍ DỤ:

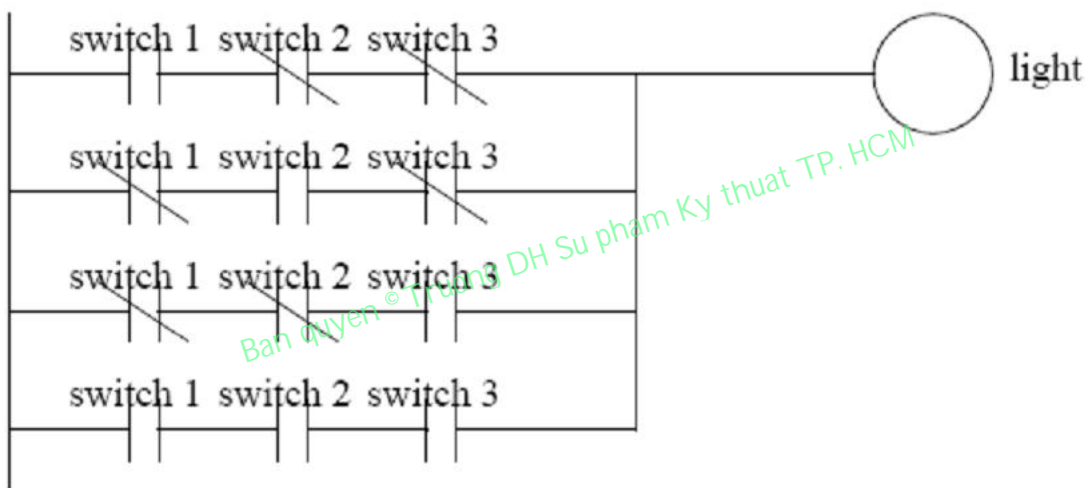
Vẽ sơ đồ điều khiển relay sử dụng 3 công tắc điều khiển 1 bóng đèn.

Có 2 cách thực hiện:

- **Cách 1:**



- **Cách 2:**



ĐIỀU KHIỂN LẬP TRÌNH 1

CHƯƠNG 2: CẤU TRÚC VÀ HOẠT ĐỘNG CỦA PLC

Topics:

- PLC hardware configurations
- Input and outputs types
- Electrical wiring for inputs and outputs
- Relays
- Electrical Ladder Diagrams and JIC wiring symbols

Objectives:

- Be able to understand and design basic input and output wiring.
- Be able to produce industrial wiring diagrams.

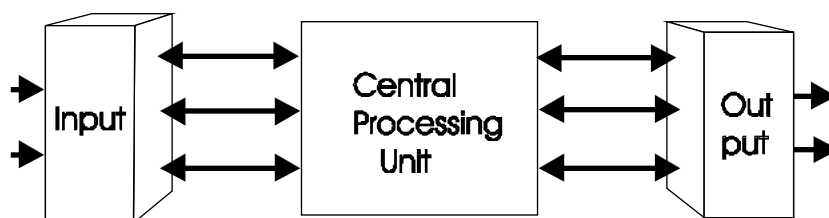
2.1 CẤU TRÚC PHẦN CỨNG PLC. PLC Hardware

2.1.1 Giới Thiệu.

PLC có nhiều cấu hình khác nhau, tuy nhiên chúng đều có chung các thành phần sau:

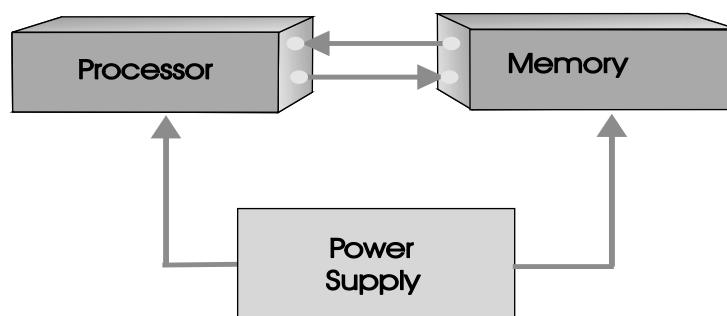
- Nguồn cung cấp: có thể tích hợp sẵn bên trong PLC hoặc làm riêng bên ngoài. Có nhiều cấp điện áp khác nhau tùy loại PLC, gồm 110VAC hoặc 220VAC hoặc 24VDC.
- CPU (Central Processing Unit) đây là bộ xử lý trung tâm làm việc như 1 máy tính, dùng để lưu trữ và xử lý chương trình logic bậc thang.
- I/O (Input/Output): phải kết nối các ngõ vào/ra để PLC có thể giám sát các quá trình và đưa ra các tác động thích hợp.
- Đèn báo: dùng để chỉ báo trạng thái PLC, gồm nguồn, chạy chương trình, lỗi hệ thống. Các cảnh báo này rất cần thiết trong chẩn đoán sự cố.

Cấu trúc tổng quát PLC như hình 2.1.



Hình 2.1: Cấu trúc PLC

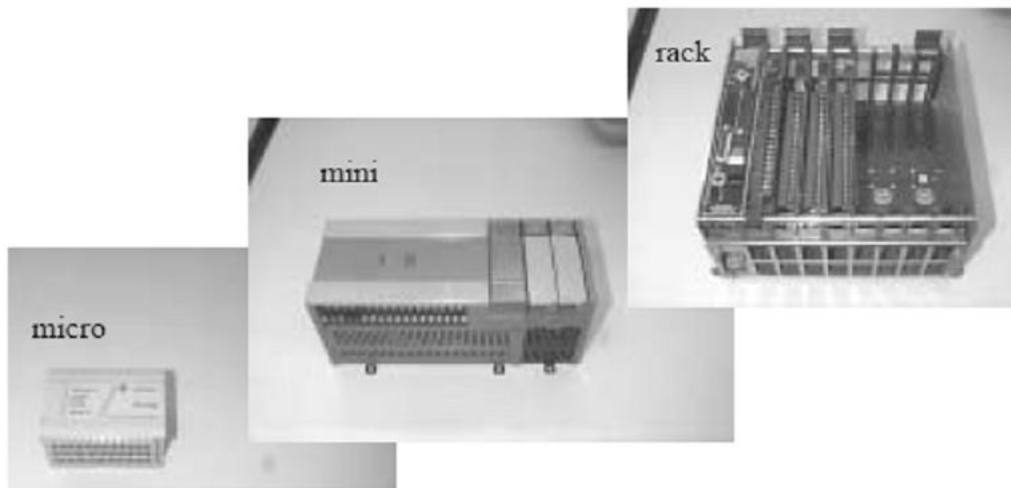
Một CPU bao gồm các thành phần như hình 2.2.



Hình 2.2: Cấu trúc CPU

ĐIỀU KHIỂN LẬP TRÌNH 1

Cấu hình của PLC liên quan đến các thành phần của nó, cấu hình tiêu biểu cho các loại khác nhau cho trên hình 2.3.



Hình 2.3: Các loại PLC

- Rack: đây là loại lớn, có thể gắn nhiều card khác nhau và có thể kết nối nhiều rack với nhau. Loại này giá thành cao nhưng linh hoạt và dễ bảo dưỡng.
- Mini: tương tự các PLC thực hiện từng chức năng của rack nhưng có kích thước nhỏ hơn.
- Micro: đây là loại nhỏ, thường có các ngõ vào/ra cố định và khả năng có hạn, giá thành thấp.

2.1.2 Ngõ Vào và Ngõ Ra.

PLC nhận các ngõ vào và tác động đến ngõ ra để giám sát và điều khiển các quá trình. Các ngõ vào và ngõ ra có thể phân chia thành 2 loại tiêu biểu: logic và liên tục.

Ví dụ xét 1 bóng đèn, nếu nó chỉ được tắt mở thì ta nói bóng đèn được điều khiển logic, nếu bóng đèn được chỉnh độ sáng tối khác nhau thì ta nói nó được điều khiển liên tục.

Các giá trị liên tục phụ thuộc nhiều vào trực giác, nên điều khiển logic vẫn được sử dụng nhiều hơn do nó cho kết quả xác định và dễ điều khiển hơn. Dĩ nhiên chọn loại nào thì còn tùy thuộc vào yêu cầu điều khiển.

Phần lớn PLC sử dụng các ngõ vào/ra logic cho các ứng dụng điều khiển.

Ngõ ra PLC được kết nối với các thiết bị chấp hành để điều khiển hệ thống, các thiết bị này bao gồm: solenoid valve, light, motor starter, servo motor. Ngõ ra PLC thường sử dụng relay hoặc các transistor cho tải DC và Triac cho tải AC. Còn các ngõ ra liên tục cần có card chuyển đổi giữa tương tự và số.

Ngõ vào PLC nhận tín hiệu từ các cảm biến. Cảm biến có nhiệm vụ chuyển đổi tín hiệu vật lý thành tín hiệu điện. Các loại cảm biến khác nhau gồm: công tắc tiếp xúc, công tắc, chiết áp,...

- Ngõ vào:

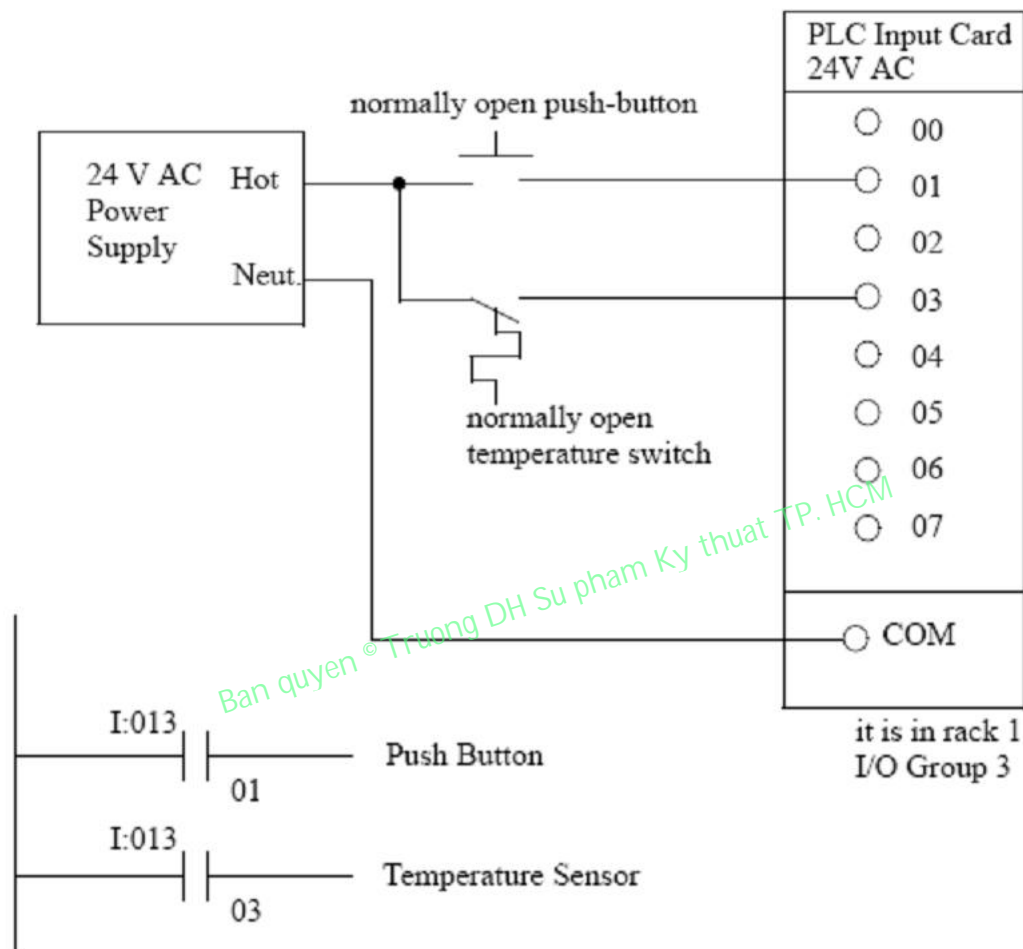
Các PLC loại nhỏ, ngõ vào thường được tích hợp bên trong và được xác định khi mua PLC. Các PLC lớn hơn, các ngõ vào được gắn ở dạng module hoặc card mở rộng.

Điện áp ngõ vào PLC gồm nhiều dải khác nhau tùy loại PLC, bao gồm:

ĐIỀU KHIỂN LẬP TRÌNH 1

- 12 – 24 VDC 24VDC
- 100 – 120VAC 48VDC
- 12 – 24 VAC/DC 200 – 240 VAC

Card ngõ vào PLC không hỗ trợ nguồn nên phải có mạch nguồn bên ngoài cấp cho ngõ vào và cảm biến. Xét ví dụ mạch kết nối card AC với ngõ vào trên hình 2.4.



Hình 2.4: Card ngõ vào AC và Logic bậc thang

Mạch trên có 2 ngõ vào gồm 1 công tắc thường hở và 1 công tắc nhiệt, đều được nối với dây HOT của nguồn cung cấp 24VAC. Khi công tắc hở, không có nguồn cấp cho ngõ vào của card, khi có 1 trong 2 công tắc đóng thì ngõ vào 01 hoặc 03 sẽ được cấp điện. Card này sẽ so sánh giá trị điện áp ở ngõ vào 01 và 03 với chân chung (được nối với dây trung tính), nếu có sự lệch áp giữa các ngõ vào này và chân chung thì ngõ vào sẽ đóng.

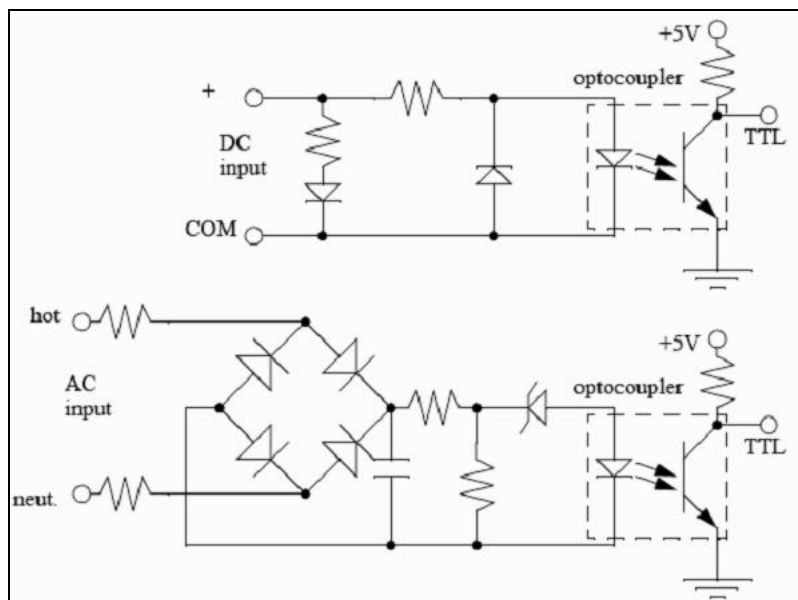
Logic bậc thang của 2 ngõ vào được vẽ bên dưới trong hình 2.4.

Có nhiều yếu tố khác nhau ảnh hưởng đến việc chọn loại card vào nào, bao gồm:

- Điện áp nguồn DC thường có giá trị thấp 12 – 24V nên an toàn hơn.
- Ngõ vào DC đáp ứng nhanh hơn AC.
- Điện áp DC có thể kết nối với nhiều hệ thống lớn hơn.
- Tín hiệu AC miễn nhiễu tốt hơn so với DC nên phù hợp cho đường truyền dài, môi trường nhiễu.
- Nguồn AC dễ thực hiện và giá thấp hơn.
- Tín hiệu AC thông dụng hơn trong các thiết bị tự động.

ĐIỀU KHIỂN LẬP TRÌNH 1

Mạch điện tiêu biểu cho ngõ vào PLC minh họa trong hình 2.5.



Hình 2.5: Mạch điện ngõ vào PLC

- Ngõ ra:

WARNING - ALWAYS CHECK RATED VOLTAGES AND CURRENTS FOR PLC's AND NEVER EXCEED!

Tương tự như ngõ vào, ngõ ra PLC không dùng để cung cấp nguồn mà nó hoạt động như 1 công tắc. Nguồn bên ngoài sẽ cấp cho card ngõ ra và nó sẽ đóng ngắt cho từng ngõ ra. Điện áp tiêu biểu cho ngõ ra có nhiều giá trị khác nhau:

12 – 48VAC	5VDC (TTL)
120VAC	24VDC
230VAC	12 – 48VDC

Card ngõ ra có thể sử dụng relay, transistor hoặc triac.

Ngõ ra relay là dạng linh hoạt nhất cho việc sử dụng thiết bị. Chúng có khả năng đáp ứng cho cả tải AC và DC nhưng khả năng đáp ứng chậm (10ms), kích thước lớn, chi phí cao, tuổi thọ ngắn và gây nhiễu.

Ngõ ra transistor chỉ sử dụng cho tải DC.

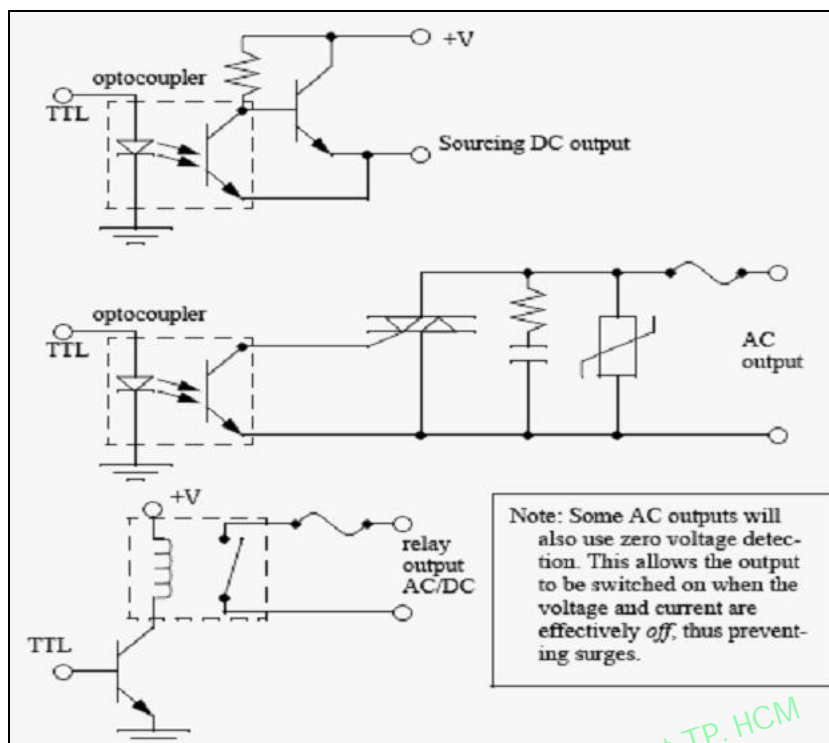
Ngõ ra triac chỉ sử dụng cho tải AC.

Mạch điện ngõ ra PLC được vẽ trên như hình 2.6.

Cần lưu ý khi kết nối hệ thống với tải AC và DC. Nếu nguồn AC được nối với ngõ ra DC dùng transistor thì nó chỉ đóng trong bán kỳ + làm cho điện áp ra tải bị giảm đi. Nếu nguồn DC được nối với ngõ ra AC sử dụng triac thì nó sẽ đóng tải làm việc và ta không thể tắt ngõ ra nếu không tắt PLC.

Một vấn đề lớn cần quan tâm đối với ngõ ra là sử dụng các nguồn kết hợp. Sẽ rất tốt nếu ta cách ly được các nguồn và giữa các chân mass riêng biệt nhau, nhưng rất khó làm việc này. Các ngõ ra relay cho phép chúng sử dụng chân mass riêng. Nhưng các loại card ra khác yêu cầu các ngõ ra trên mỗi card sử dụng chung chân mass.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 2.6: Mạch điện ngõ ra PLC

Hình vẽ 2.7 minh họa card ra 24VDC sử dụng chân mass chung.

Card loại này thường dùng transistor làm ngõ ra rút dòng.

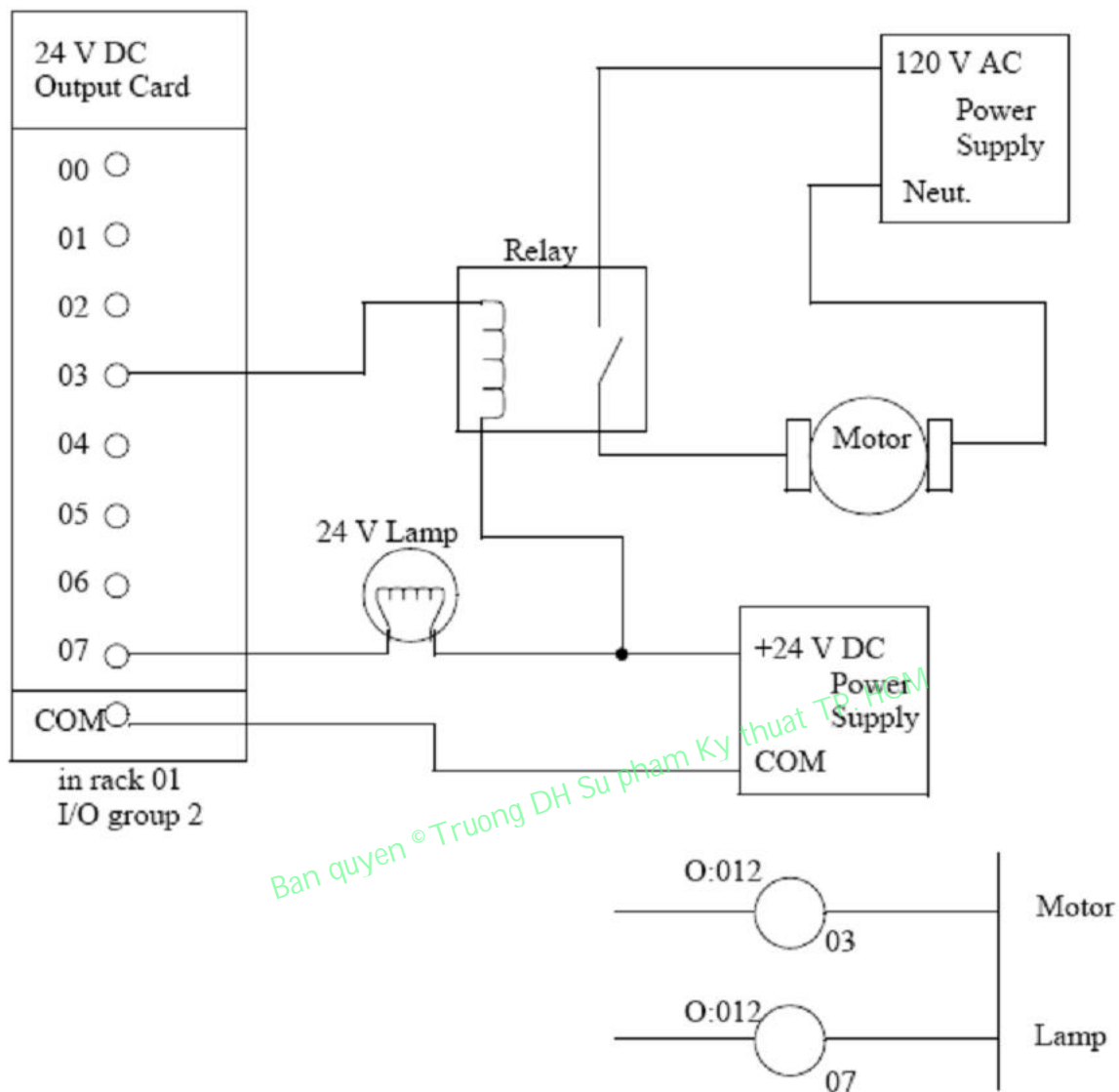
Trong mạch này, ngõ ra của card được nối với bóng đèn công suất nhỏ và một cuộn dây của relay.

Xét mạch điều khiển bóng đèn: nếu ngõ ra 07 đóng thì dòng điện chạy từ nguồn 24V qua đèn vào ngõ ra 07 rồi về mass – đèn sáng. Nếu ngõ ra mở thì hở mạch, đèn tắt.

Xét mạch điều khiển động cơ: cuộn dây của relay được nối với ngõ ra 03 tương tự như bóng đèn, nên khi ngõ ra đóng thì cuộn dây relay có điện đóng tiếp điểm relay cấp nguồn 120VAC làm động cơ hoạt động.

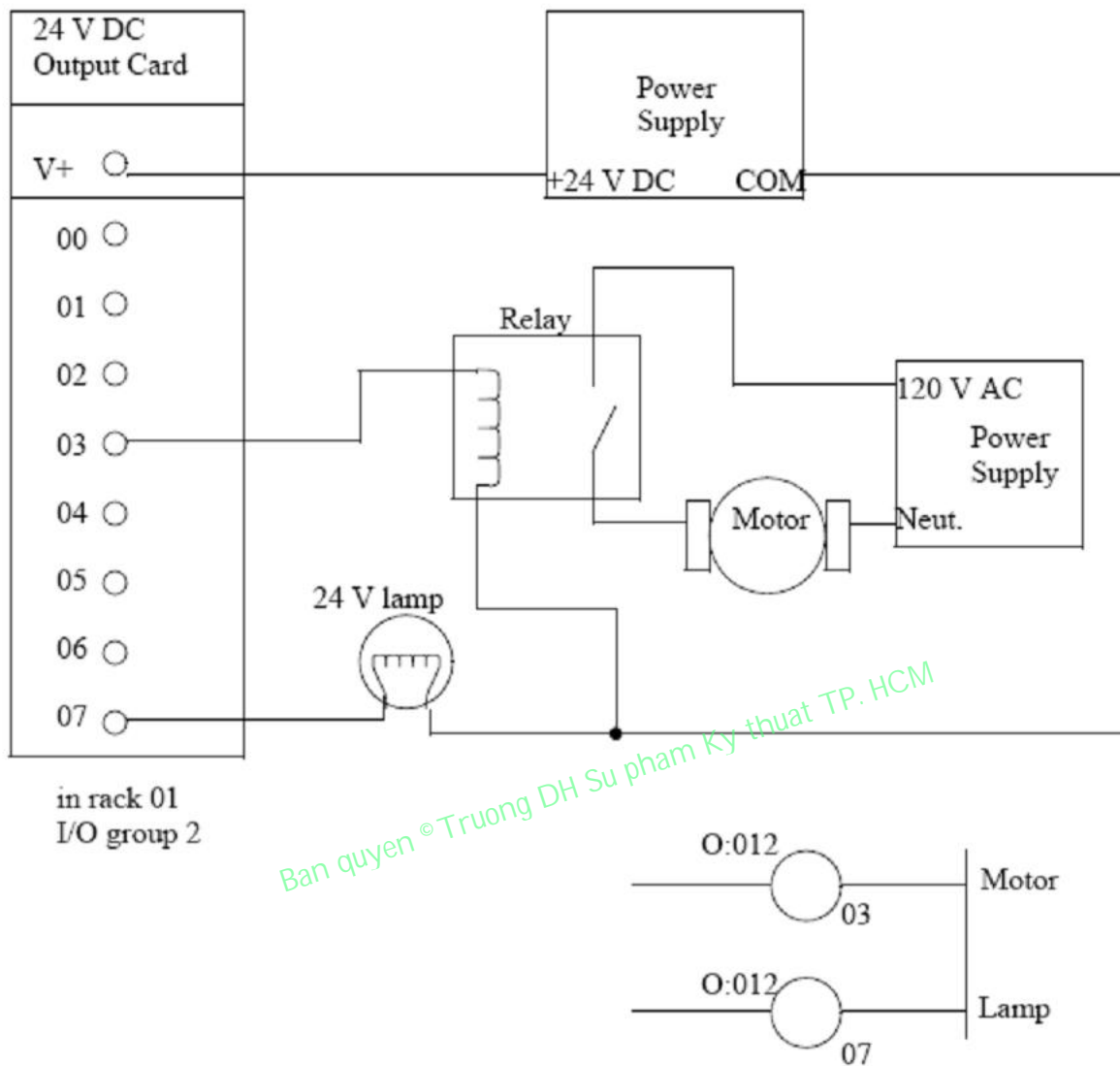
Tương tự cho các ngõ ra cấp dòng và ngõ ra relay trong các hình 2.8 và 2.9.

ĐIỀU KHIỂN LẬP TRÌNH 1



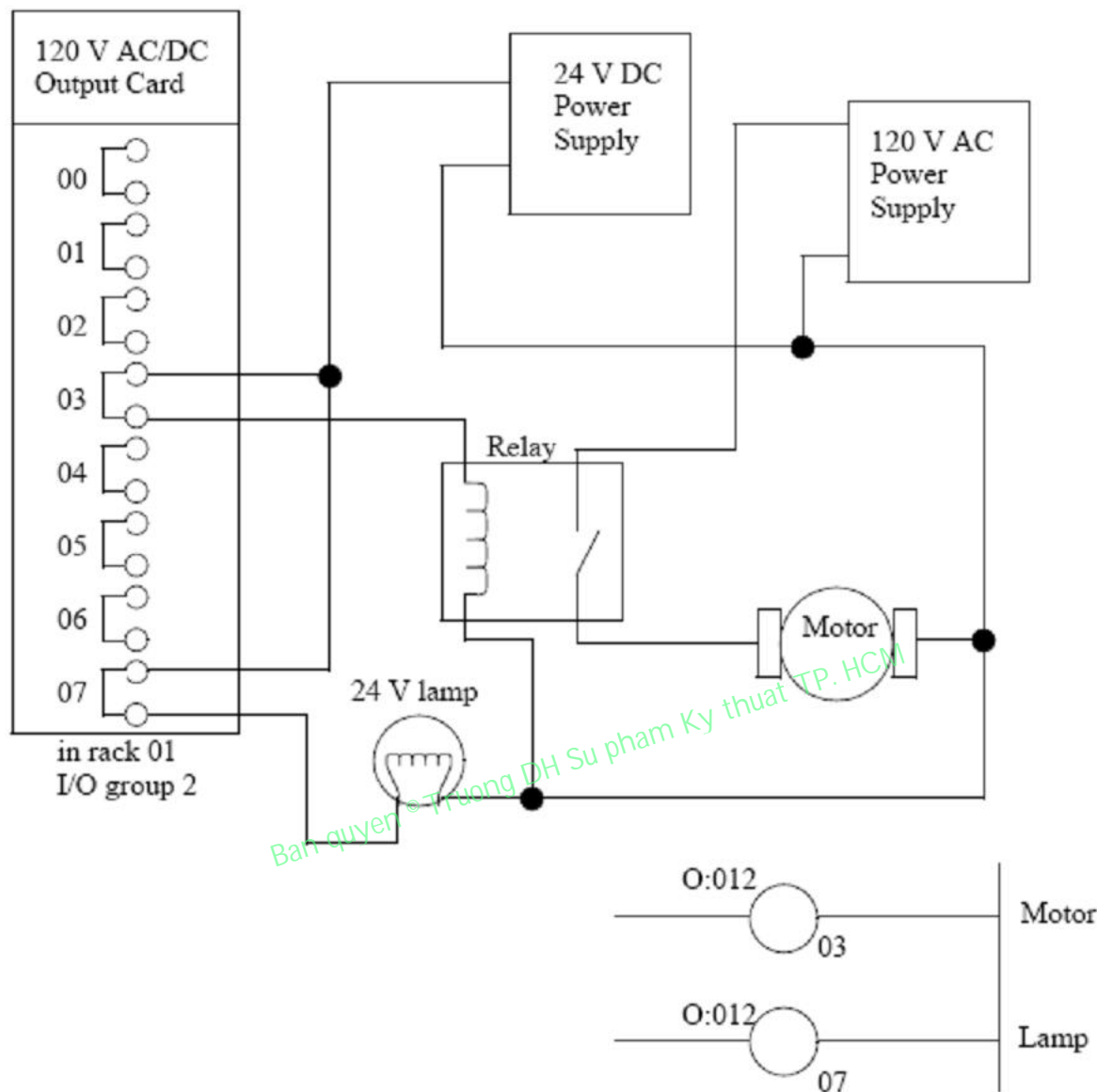
Hình 2.7: Card ngõ ra 24VDC (Rút dòng)

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 2.8: Card ngõ ra 24VDC (Cấp dòng)

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 2.9: Card ngõ ra Relay

2.1.3 Relay.

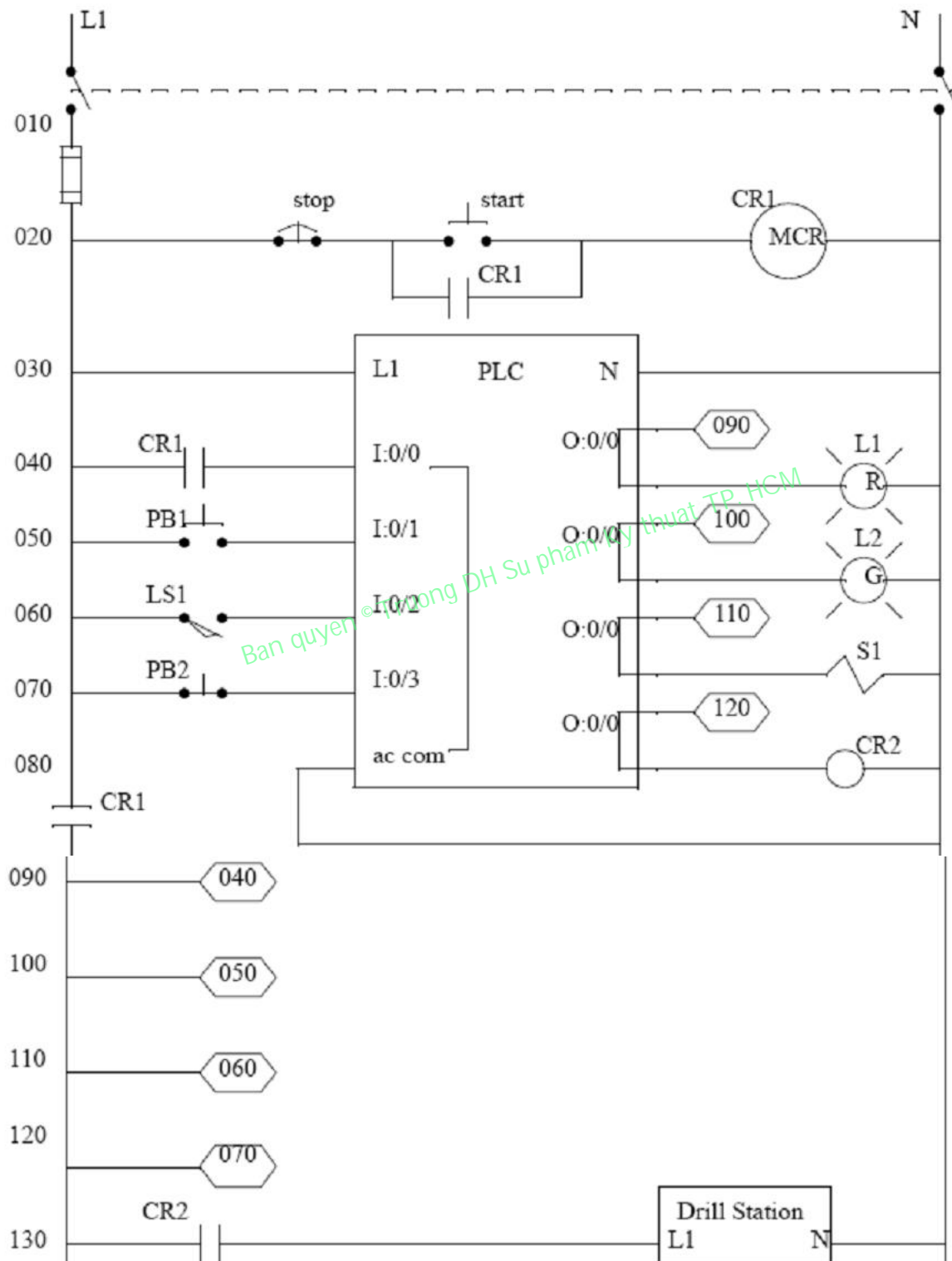
Mặc dù relay ít được dùng trong điều khiển logic nhưng chúng vẫn hiệu quả trong đóng mở các tải công suất lớn. Một số thuật ngữ quan trọng cần quan tâm trong điều khiển relay:

- Contactor: là loại relay đặc biệt dùng đóng ngắt các tải có dòng rất lớn.
- Motor starter: sử dụng contactor mắc nối tiếp với relay chịu quá tải để cắt khi có dòng quá lớn đi qua.
- Arc suppression: khi một relay đóng ngắt sẽ xuất hiện hồ quang, đặc biệt với các tải có dòng lớn. Đối với nguồn AC có thể khắc phục bằng cách đóng ngắt ngay thời điểm 0V. Còn với nguồn DC ta có thể giảm hồ quang bằng cách thổi khí áp suất cao qua tiếp điểm đang đóng mở.
- AC coils: khi cuộn dây relay được điều khiển bằng nguồn AC thì tiếp điểm của nó sẽ đóng mở dao động với cùng tần số nguồn AC. Khắc phục hiện tượng này bằng cách thêm vào relay một cực **shading**.

ĐIỀU KHIỂN LẬP TRÌNH 1

2.1.4 Sơ Đồ Nối Dây.

Khi thiết kế một tủ điều khiển, sơ đồ logic bậc thang sẽ được xây dựng để làm tài liệu cho việc nối dây. Sơ đồ nối dây được vẽ trên hình 2.10.



Hình 2.10: Sơ đồ nối dây bậc thang

Hệ thống được cấp nguồn 110VAC hoặc 220VAC tại các đường L1 và N. Các đường của sơ đồ sẽ được đánh số và các số này sẽ được dùng để đánh số dây dẫn khi thực hiện nối dây phần điện cho hệ thống.

ĐIỀU KHIỂN LẬP TRÌNH 1

Công tắc trước đường 010 là công tắc chính cấp nguồn cho hệ thống. Cầu chì dùng để giới hạn dòng điện cho hệ thống. Đường 020 của sơ đồ dùng để điều khiển công suất đến ngõ ra. Nút nhấn thường đóng Stop, nút nhấn thường hở Start. PLC nhận công suất từ đường 30 trên sơ đồ. Ngõ vào của PLC đều là AC từ đường 040 đến 070.

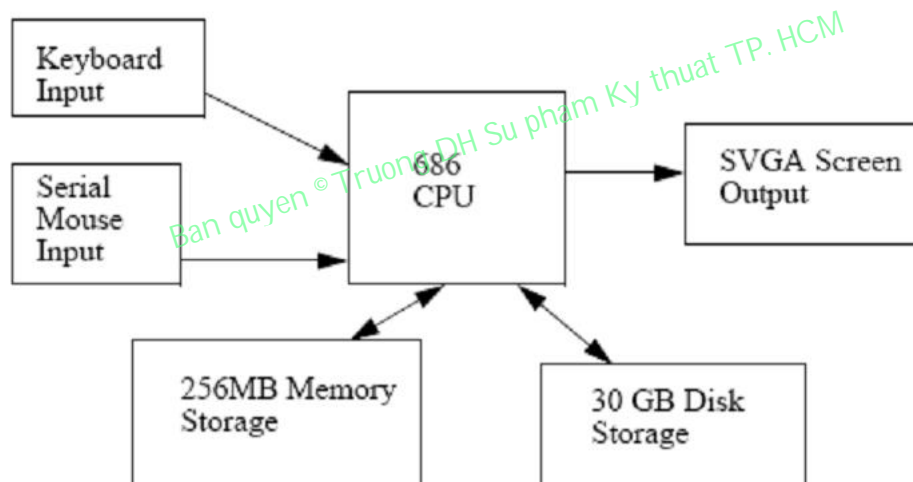
Trong sơ đồ nối dây, việc chọn các nút nhấn thường đóng và thường hở cho nút Stop và Start là có chủ ý. Xét đường 020, nếu nút Stop được nhấn nó sẽ mở tiếp điểm cắt đường cấp nguồn đến relay nên ngõ ra sẽ tắt. Nếu nút nhấn này bị hư vẫn không có nguồn cấp đến ngõ ra nên hệ thống vẫn an toàn. Nếu nút nhấn này là thường hở, khi nó bị hư sẽ đóng lại làm hệ thống sẽ hoạt động và ta không thể tắt hệ thống.

2.2 HOẠT ĐỘNG CỦA PLC. PLC Operation

2.2.1 Giới Thiệu.

Mô hình điều khiển relay của PLC chỉ phù hợp với các chương trình đơn giản. Với các chương trình phức tạp cần có các mô hình phức tạp hơn cho PLC. Khi đó, mô hình PLC tương tự như một máy tính.

Xét sơ đồ máy tính như hình 2.11.



Hình 2.11: Cấu trúc đơn giản của máy tính

Tín hiệu ngõ vào từ Keyboard và Mouse, tín hiệu ngõ ra được đưa đến Screen, Disk và Memory được sử dụng lưu trữ cho cả ngõ vào và ngõ ra.

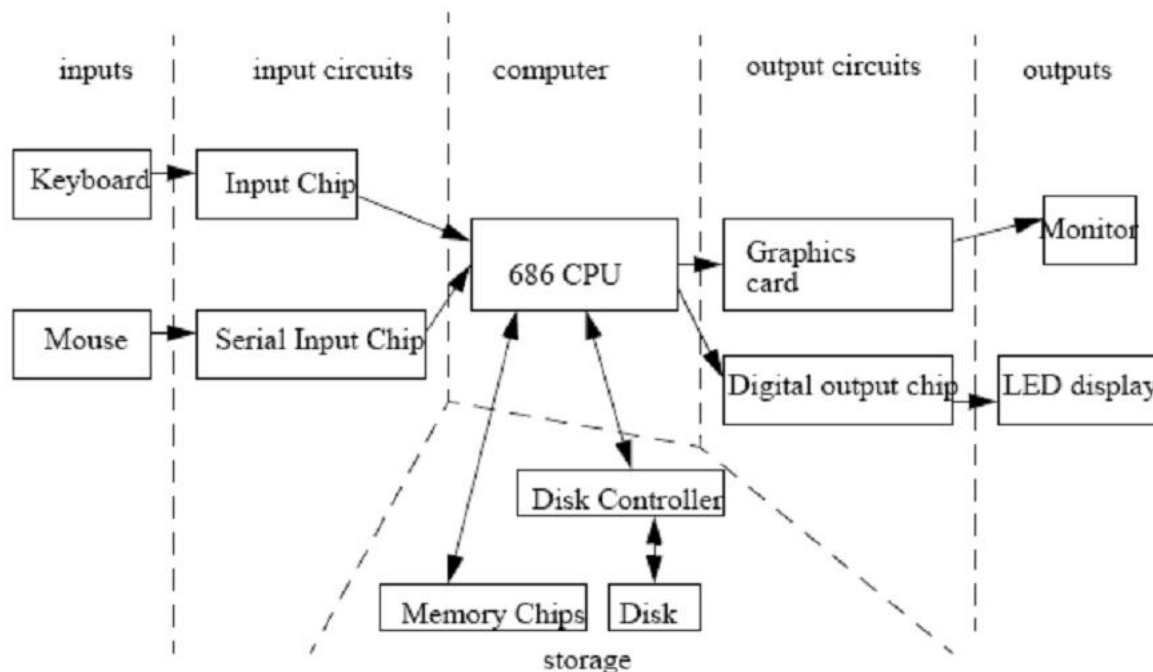
Sơ đồ này được vẽ lại rõ hơn như hình 2.12.

Trong sơ đồ này, dữ liệu đưa vào hệ thống từ bên trái, qua các ngõ vào. Sau đó dữ liệu qua mạch đệm để vào CPU. Sau khi xử lý, CPU sẽ xuất dữ liệu đến ngõ ra qua mạch đệm ngõ ra. Đĩa cứng và bộ nhớ được dùng lưu trữ dữ liệu.

PLC hoạt động tương tự máy tính, được so sánh như sau:

- Ngõ vào: bàn phím tương tự với các công tắc tiệm cận.
- Mạch điện ngõ vào tương tự card ngõ vào.
- CPU máy tính tương tự CPU của PLC.
- Mạch điện ngõ ra tương tự card ngõ ra.
- Ngõ ra: màn hình máy tính tương tự đèn báo PLC.

ĐIỀU KHIỂN LẬP TRÌNH 1



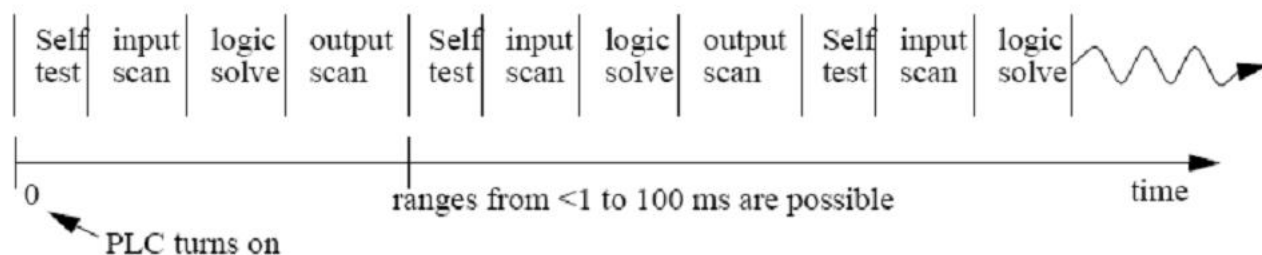
Hình 2.12: Cấu trúc theo ngõ vào và ngõ ra

2.2.2 Hoạt Động Tuần Tự.

Tất cả PLC đều hoạt động theo chu trình lập, mỗi chu trình hoạt động gồm 4 giai đoạn.

- Giai đoạn 1: sau khi bật nguồn, PLC sẽ tự kiểm tra lỗi phần cứng và phần mềm.
- Giai đoạn 2: nếu không có lỗi nó sẽ đọc toàn bộ giá trị ngõ vào và chứa vào bộ nhớ, giai đoạn này gọi là đọc ngõ vào.
- Giai đoạn 3: với dữ liệu trong bộ nhớ này, chương trình logic bậc thang sẽ được thực hiện một lần, giai đoạn này gọi là thực hiện chương trình.
- Giai đoạn 4: trong khi thực hiện chương trình logic bậc thang các giá trị ngõ ra chỉ được thay đổi tạm thời trong bộ nhớ. Sau khi quét xong chương trình, dữ liệu ngõ ra sẽ được cập nhật từ các giá trị tạm thời, giai đoạn này gọi là xuất dữ liệu ngõ ra.

Sau đó PLC sẽ khởi động lại quá trình bằng cách khởi động việc kiểm tra lỗi. Quá trình này lặp lại từ 10 đến 100 lần mỗi s, như trên hình 2.10.



Hình 2.10: Một chu kỳ vòng quét.

Tự kiểm tra lỗi: kiểm tra lỗi trên tất cả phần cứng.

Đọc dữ liệu vào: đọc dữ liệu từ card vào, copy chúng vào bộ nhớ. Việc này giúp cho PLC hoạt động nhanh hơn và tránh trường hợp một ngõ vào thay đổi từ lúc khởi đầu cho đến lúc kết thúc hương trình. Có những hàm đặc biệt của PLC cho phép đọc trực tiếp dữ liệu vào.

ĐIỀU KHIỂN LẬP TRÌNH 1

Thực hiện chương trình logic: dựa vào bảng dữ liệu vào trong bộ nhớ, chương trình sẽ được thực hiện từng bước, và cập nhật ngõ ra.

Xuất dữ liệu ngõ ra: bảng dữ liệu ngõ ra sẽ được copy từ bộ nhớ đến card ra, và xuất ra điều khiển thiết bị.

Ban đầu khi PLC được bật nguồn thì các ngõ ra sẽ ở trạng thái tắt, điều này không ảnh hưởng đến các giá trị ngõ vào.

2.2.3 Trạng Thái PLC.

Điều dễ nhận thấy ở PLC là nó thiếu bàn phím và các thiết bị vào ra khác. Mặt trước PLC thường có các đèn chỉ trạng thái, bao gồm:

- Đèn báo nguồn.
- Đèn chạy chương trình.
- Đèn báo sự cố.

Các đèn này thường dùng cho việc sửa lỗi.

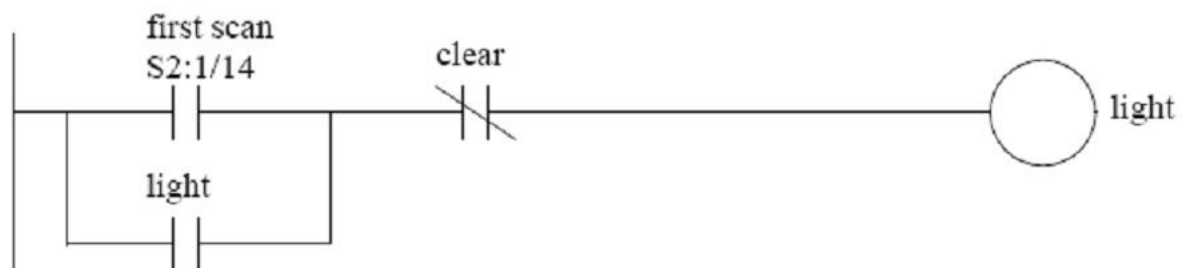
Ngoài ra phần cứng PLC còn có các nút nhấn, phổ biến nhất là nút chạy chương trình.

PLC không cần các công tắc On/Off, hoặc nút nhấn Reset. Chúng thường được thiết kế bên trong hệ thống.

Ta cũng có thể nhận biết trạng thái của PLC bằng chương trình logic bậc thang, bằng cách kiểm tra xem chúng có được thực thi trong vòng quét ban đầu hay không.

Xét hình vẽ 2.11.

Ngõ vào “first scan” sẽ đúng trong lần đầu thực hiện các lệnh logic bậc thang, các vòng quét khác sẽ sai. Trong trường hợp này, địa chỉ cho ngõ vào “first scan” là S2:1/14. Với chương trình logic này, vòng quét đầu sẽ thực hiện việc đóng đèn, cho đến khi tiếp điểm “clear” đóng. Vì vậy đèn sẽ sáng sau khi PLC đóng, và nó sẽ tắt nếu “clear” được đóng.



Hình 2.11: Chương trình kiểm tra vòng quét đầu tiên

2.2.4 Bộ Nhớ.

Các loại bộ nhớ được dùng phổ biến hiện nay bao gồm: RAM, ROM, EPROM, EEPROM. (tương tự như phân bộ nhớ trong Kỹ thuật số).

Tất cả PLC đều sử dụng RAM cho CPU và dùng ROM để lưu hệ điều hành cho PLC.

Khi bật nguồn, nội dung của RAM sẽ được giữ lại. Nhưng vấn đề cần quan tâm là chuyện gì xảy ra khi bộ nhớ này mất nguồn. Các PLC trước kia sử dụng RAM có nguồn pin nên dữ liệu RAM không bị mất khi mất điện. Phương pháp này vẫn còn sử dụng nhưng không nhiều. Ngày nay người ta sử dụng EPROM làm bộ nhớ cho PLC. Bộ nhớ này được lập trình bên ngoài PLC sau đó đặt vào PLC. Khi PLC hoạt động chương trình này sẽ được nạp vào PLC và thực hiện. Phương pháp này chính xác nhưng việc lập trình và xóa bộ nhớ sẽ mất nhiều thời gian. Bộ nhớ EEPROM trở thành

ĐIỀU KHIỂN LẬP TRÌNH 1

một phần cố định của PLC, các chương trình lưu trong EEPROM tương tự như lưu trong EPROM. Hiện nay giá thành các bộ nhớ đã giảm đáng kể, và người ta còn phát triển thêm các bộ nhớ khác như Flash ROM.

TÓM TẮT:

- A PLC and computer are similar with inputs, outputs, memory, etc.
- The PLC continuously goes through a cycle including a sanity check, input scan, logic scan, and output scan.
- While the logic is being scanned, changes in the inputs are not detected, and the outputs are not updated.
- PLCs use RAM, and sometime EPROMs are used for permanent programs.

Bản quyền © Truong DH Su pham Ky thuat TP. HCM

ĐIỀU KHIỂN LẬP TRÌNH 1

CHƯƠNG 3: CẢM BIẾN

3.1 GIỚI THIỆU.

Các cảm biến cho phép một PLC nhận biết các trạng thái của một quá trình hoạt động. Các cảm biến logic chỉ nhận biết một trạng thái đúng hoặc sai, một số hiện tượng vật lý được nhận biết bao gồm:

- Có một vật kim loại ở gần hay không?
- Có một vật điện môi ở gần hay không?
- Có vật che hoặc phản xạ ánh sáng hay không?
- ...

3.2 CẢM BIẾN DÂY NÓI. Sensor Wiring

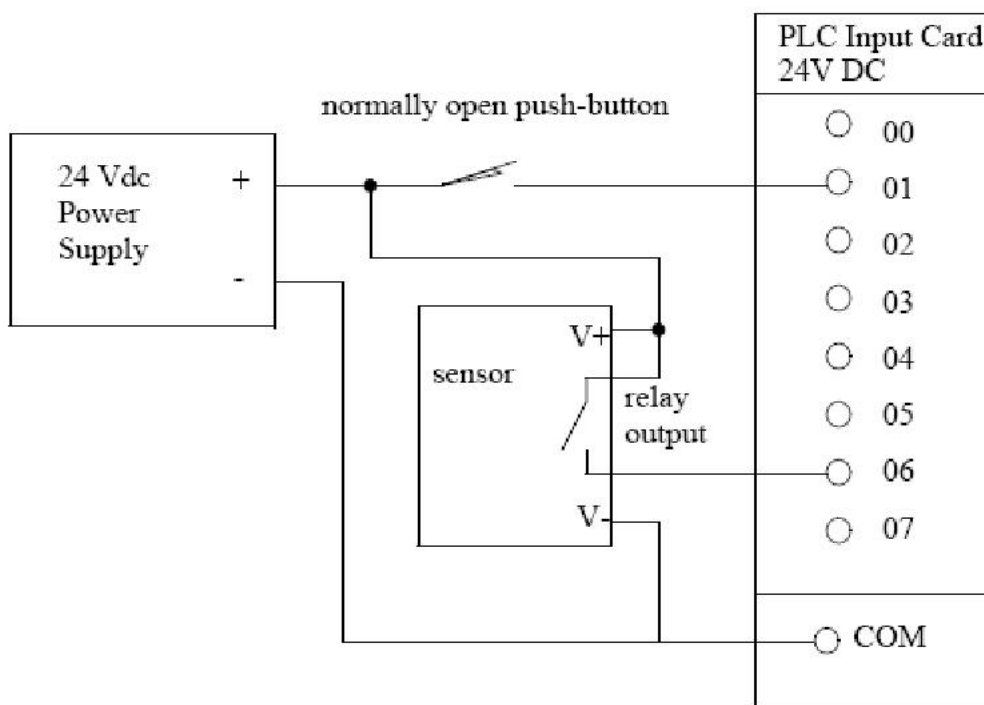
Khi một cảm biến nhận biết có sự thay đổi logic thì nó sẽ báo cho PLC biết sự thay đổi này bằng cách đóng/ ngắt một điện áp hoặc dòng điện đến PLC. Trong một số trường hợp ngõ ra của cảm biến sẽ đóng ngắt trực tiếp tải.

Ngõ ra của cảm biến (ngõ vào PLC) bao gồm:

- Ngõ ra cấp dòng hoặc rút dòng.
- Các công tắc để đóng ngắt điện áp.
- Các tiếp điểm relay để đóng ngắt ngõ ra AC.
- Ngõ ra TTL chỉ mức logic 0 hoặc 5V.

3.2.1 Công Tắc.

Ví dụ đơn giản nhất của ngõ ra cảm biến là các công tắc hoặc tiếp điểm relay, minh họa trong hình vẽ 3.1.



Hình 3.1: Cảm biến đóng ngắt.

Hình vẽ này bao gồm công tắc thường hở NO (Normal Open) được nối đến ngõ vào I0.1, cảm biến có ngõ ra relay được cấp nguồn +/-V.

ĐIỀU KHIỂN LẬP TRÌNH 1

Ngõ ra cảm biến sẽ tác động khi xảy ra một hiện tượng nào đó định trước. công tắc bên trong cảm biến sẽ đóng lại cấp điện áp đến ngõ vào I0.6 của PLC.

3.2.2 Ngõ ra TTL.

Ngõ ra TTL dựa vào 2 mức điện áp 0V và 5V (có cho phép sai số như trong môn học Kỹ thuật số). Phương pháp này rất nhạy với nhiễu điện trong nhà máy nên chỉ được sử dụng khi thật sự cần thiết.

Các mạch điều khiển điện tử và máy tính thường có ngõ ra TTL, khi nối với các thiết bị khác cần thêm mạch trigger để cải thiện tín hiệu.

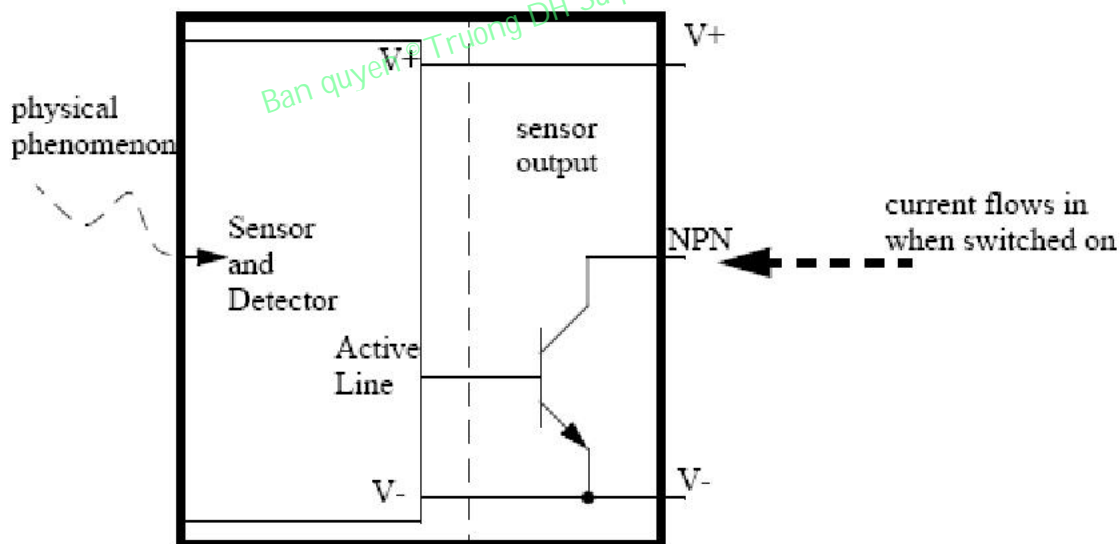
Nếu cảm biến có ngõ ra TTL thì PLC phải dùng card ngõ vào để đọc các giá trị TTL. Nếu sử dụng cảm biến ngõ ra TTL cho các ứng dụng khác thì lưu ý dòng max ngõ ra là 20mA.

3.2.3 Ngõ ra Rút Dòng và Cấp Dòng. Sinking/Sourcing

Cảm biến rút dòng cho phép dòng chạy vào cảm biến về mass, còn cảm biến cấp dòng từ nguồn Vcc chạy ra cảm biến.

Trong cả 2 trường hợp, ta chỉ quan tâm đến dòng điện, không quan tâm điện áp nên giảm được ảnh hưởng của nhiễu điện.

Ngõ ra của cảm biến sử dụng transistor đóng ngắt (có tổn hao điện áp). Loại PNP dùng cho ngõ ra rút dòng, loại NPN ngõ ra cấp dòng. Minh họa trên hình vẽ 3.2.



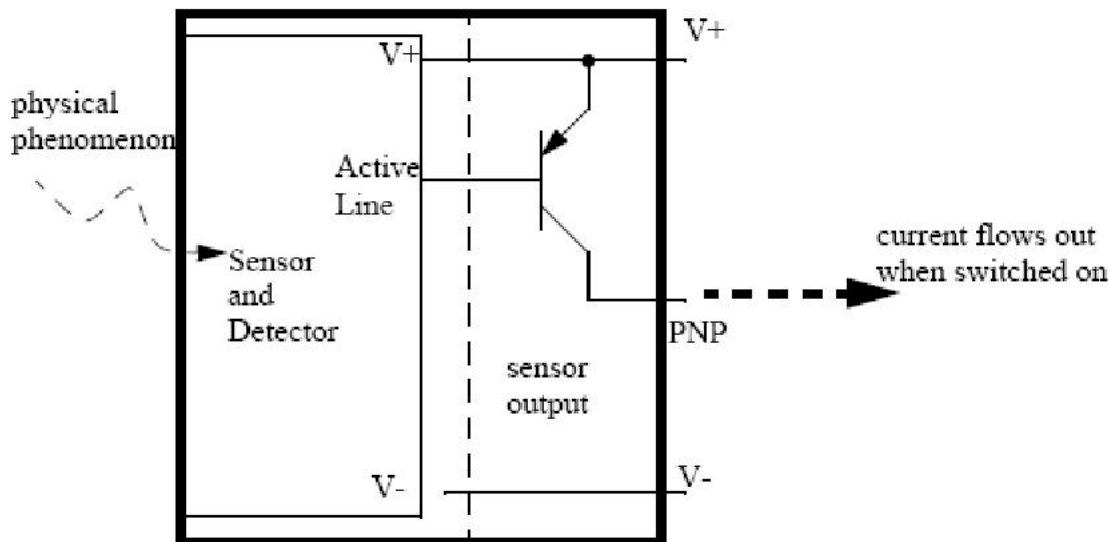
Hình 3.2: Ngõ ra Rút dòng

Cảm biến có bộ phận đầu dò để nhận biết các hiện tượng vật lý xảy ra. Với nguồn cung cấp +/-V cảm biến sẽ nhận biết các hiện tượng xảy ra và tác động vào chân B của transistor NPN.

Nếu chân B có 0V thì transistor ngưng dẫn, nếu chân B có 5V/12V thì transistor dẫn bảo hòa rút dòng bên ngoài vào.

Hoạt động tương tự cho cảm biến cấp dòng PNP ở hình 3.3.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 3.3: Ngõ ra Cấp dòng

3.2.4 Tiếp điểm Relay Solid State Relay.

Sử dụng cảm biến có ngõ ra relay để đóng cắt dòng AC, phù hợp cho các tải lớn, yêu cầu giá thành thấp. Một số cảm biến và thiết bị còn sử dụng relay làm ngõ ra.

3.3 CẢM BIẾN TIỆM CẬN. Presence Detection

Có 2 cách để phát hiện sự xuất hiện của một vật: tiếp xúc và tiệm cận.

Tiếp xúc có nghĩa là dùng công tắc cơ khí và tạo ra lực giữa vật và cảm biến.

Tiệt cận nghĩa là có vật ở gần nhưng không cần công tắc.

Những phần tiếp theo sẽ trình bày về các loại cảm biến dùng để nhận biết sự hiện diện của một vật. Các cảm biến này sẽ giải thích cho phần lớn các ứng dụng của cảm biến.

3.3.1 Công Tắc Tiếp Xúc.

Công tắc tiếp xúc có dạng thường hở hoặc thường đóng. Vỏ của chúng được gia cố nên chúng có thể lặp lại các tác động đóng mở liên tục.

3.3.2 Công tắc Lưỡi Gà.

Tương tự relay nhưng sử dụng nam châm vĩnh cửu thay cho cuộn dây. Nam châm ở xa thì công tắc mở và ngược lại. chúng thường được sử dụng trong các bảng thông báo và cửa nhà vì bền hơn các cảm biến khác.

Hình dạng công tắc lưỡi gà cho trên hình 3.4.

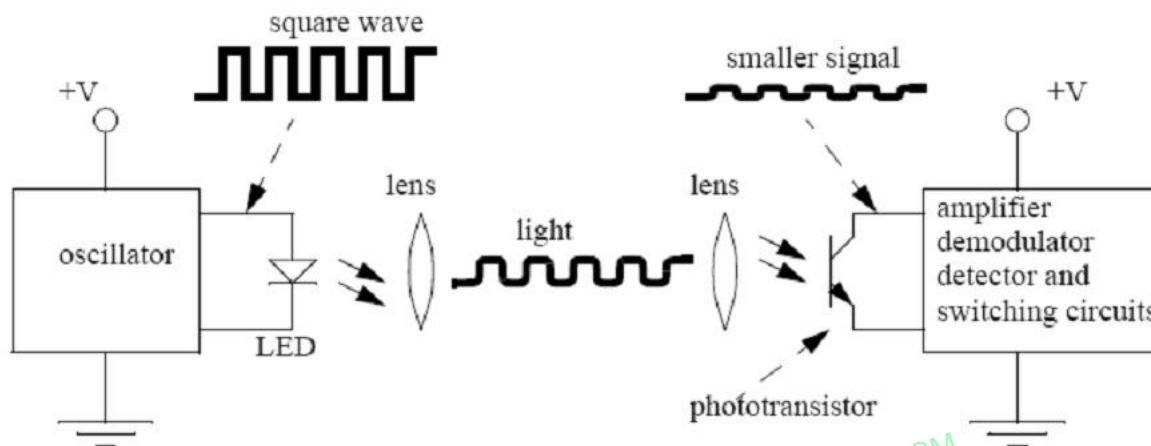


Hình 3.4: Công tắc lưỡi gà

ĐIỀU KHIỂN LẬP TRÌNH 1

3.3.3 Cảm Biến Quang.

Cảm biến quang được dùng rất lâu, bao gồm một nguồn phát quang và một bộ thu quang. Nguồn quang sử dụng LED hoặc LASER phát ra ánh sáng thấy hoặc không thấy tùy theo bước sóng. Bộ thu quang sử dụng diode hoặc transistor quang. Ta đặt bộ thu và phát sao cho vật cần nhận biết có thể che chắn hoặc phản xạ ánh sáng khi vật xuất hiện. Sơ đồ sử dụng cảm biến quang cho trên hình 3.5.



Hình 3.5: Cảm biến quang

Ánh sáng do LED phát ra được hội tụ qua thấu kính. Ở phần thu ánh sáng từ thấu kính tác động đến transistor thu quang.

Nếu có vật che chắn thì chùm tia sẽ không tác động đến bộ thu được. Sóng dao động dùng để bộ thu loại bỏ ảnh hưởng của ánh sáng trong phòng. Ánh sáng của mạch phát sẽ tắt và sáng theo tần số mạch dao động.

Phương pháp sử dụng mạch dao động làm cho cảm biến thu phát xa hơn và tiêu thụ ít công suất hơn.

3.3.4 Cảm Biến Điện Dung.

Cảm biến điện dung có thể nhận biết các vật ở khoảng cách lên đến vài cm.

Công thức tính điện dung:

$$C = A.K/D$$

C: điện dung (F)

D: hằng số điện môi.

A: diện tích bản cực.

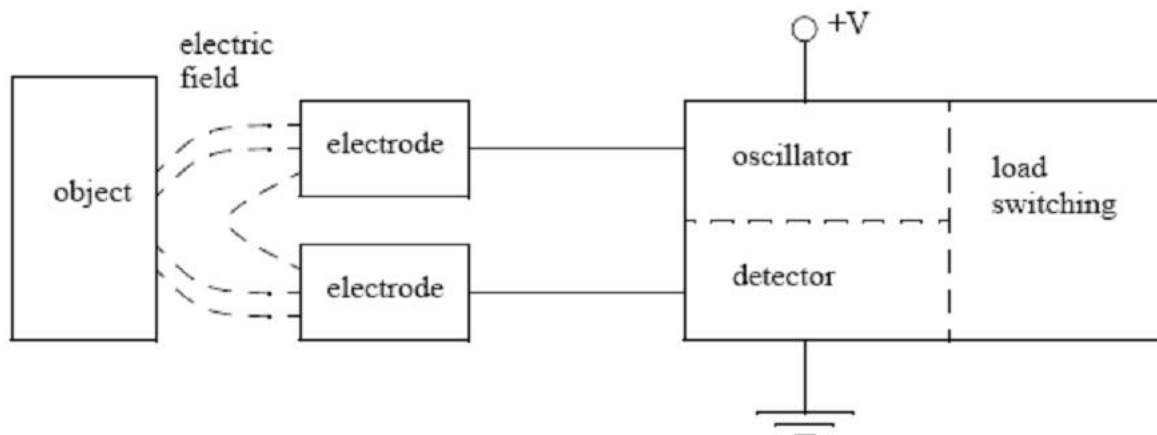
K: khoảng cách 2 bản cực.

Đối với cảm biến điện dung thì diện tích và khoảng cách 2 bản cực là cố định, nhưng hằng số điện môi của môi trường xung quanh 2 bản cực sẽ thay đổi khi có các vật khác nhau đến gần.

Hình vẽ cảm biến điện dung cho trên hình 3.6.

Điện dung của 2 bản cực sẽ được xác định bởi một trường thay đổi. Khi có vật đến gần làm thay đổi điện môi giữa 2 bản cực sẽ làm thay đổi điện dung đến giá trị đặt trước nên cảm biến sẽ tác động đóng cắt tải.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 3.6: Cảm biến điện dung

Các cảm biến này làm việc tốt đối với vật cách điện vì chúng có hệ số điện môi lớn nên điện dung lớn. Chúng cũng làm việc tốt đối với kim loại vì các vật dẫn điện tốt xuất hiện sẽ giống như các bản cực lớn hơn nên cũng làm tăng điện dung.

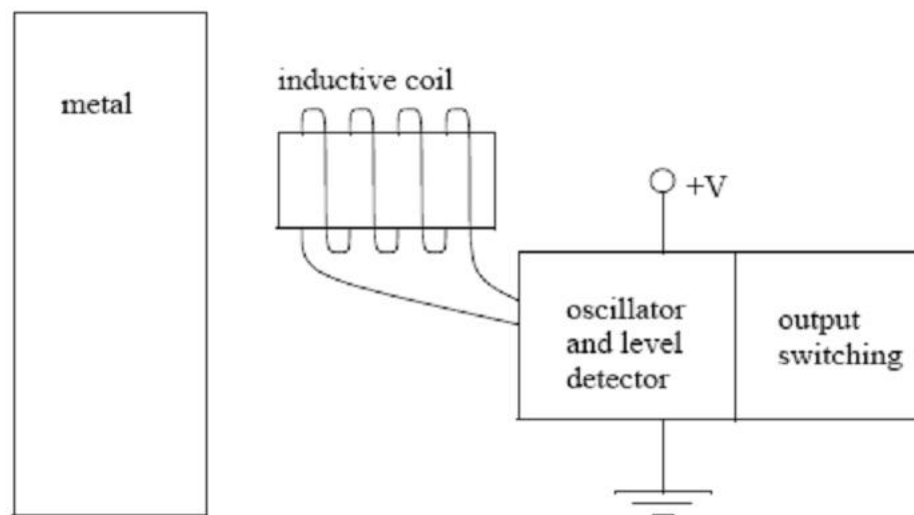
Hình 3.7 minh họa ảnh hưởng của vật cách điện và vật dẫn điện đối với cảm biến.



Hình 3.7: Vật cách điện và dẫn điện làm tăng điện dung

3.3.5 Cảm Biến Điện Cảm.

Cảm biến điện cảm sử dụng các từ trường cảm ứng để nhận biết các vật kim loại ở gần. Nó sử dụng cuộn cảm để tạo ra từ trường tần số cao như hình vẽ 3.8.



Hình 3.8: Cảm biến điện cảm.

ĐIỀU KHIỂN LẬP TRÌNH 1

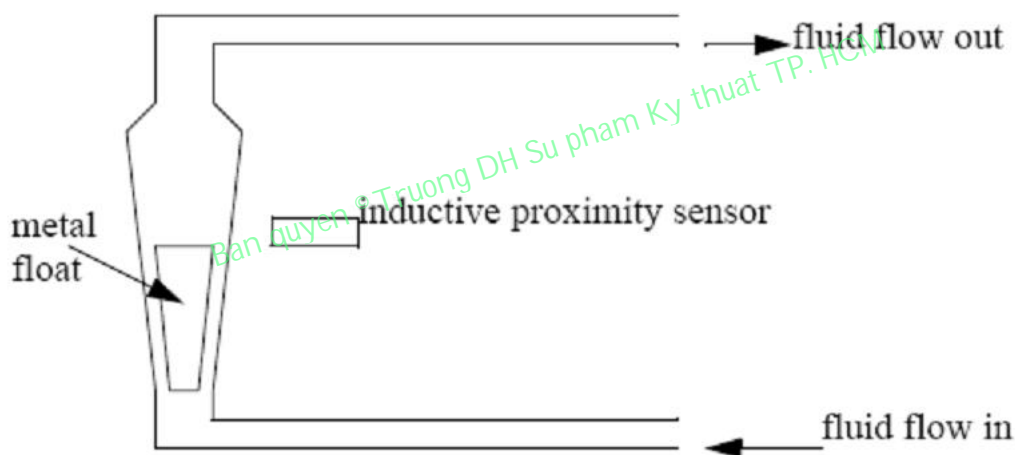
Nếu có một vật kim loại đến gần thì từ trường sẽ thay đổi tạo ra dòng điện qua vật. Dòng điện này tạo ra từ trường mới ngược với từ trường ban đầu nên làm thay đổi cảm kháng cuộn dây bên trong cảm biến. Bằng cách đo điện cảm cảm biến sẽ nhận biết khi có vật kim loại đến gần nó.

Cảm biến loại này có thể dùng để nhận biết bất kỳ vật kim loại nào. Khi cần nhận biết nhiều vật thì ta sử dụng nhiều cảm biến.

3.3.6 Dòng Chất Lỏng.

Ta có thể đặt các cảm biến phức tạp bên ngoài các cảm biến đơn giản hơn. Hình vẽ 3.9 mô tả một ứng dụng này.

Người ta đặt một cái phao kim loại bên trong một ống hình thang. Khi tốc độ dòng chất lỏng tăng thì áp lực tác động lên phao cũng tăng theo. Phao có dạng hình thang để tạo vị trí cân bằng tỉ lệ với tốc độ chảy. Một cảm biến điện cảm được đặt bên ngoài ống chất lỏng sao cho nó có thể nhận biết sự thay đổi độ cao của phao bên trong và xác định được tốc độ chảy trong ống.



Hình 3.8: Nhận biết tốc độ chảy bằng cảm biến điện cảm

TÓM TẮT:

- Sourcing sensors allow current to flow out from the V+ supply.
- Sinking sensors allow current to flow in to the V- supply.
- Photo-optical sensors can use reflected beams (retroreflective), an emitter and detector (opposed mode) and reflected light (diffuse) to detect a part.
- Capacitive sensors can detect metals and other materials.
- Inductive sensors can detect metals.
- Hall effect and reed switches can detect magnets.
- Ultrasonic sensors use sound waves to detect parts up to meters away.

ĐIỀU KHIỂN LẬP TRÌNH 1

CHƯƠNG 4: THIẾT BỊ CHẤP HÀNH

Topics:

- Solenoids, valves and cylinders
- Hydraulics and pneumatics
- Other actuators

Objectives:

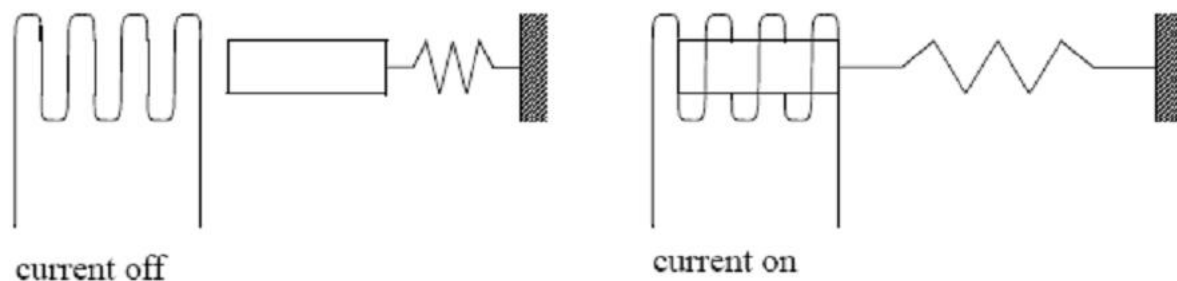
- Be aware of various actuators available.

4.1 GIỚI THIỆU.

Thiết bị chấp hành là các thiết bị nhận tín hiệu điều khiển từ ngõ ra của PLC và tác động đến chuyển động của các thiết bị cơ khí – quá trình chuyển hóa cơ/điện.

4.2 SOLENOID.

Solenoid là một trong những thiết bị chấp hành thông dụng nhất. Nguyên lý hoạt động dựa vào chuyển động của lõi sắt (gọi là piston) trong lòng cuộn dây. Thông thường piston được giữ bên ngoài cuộn dây bằng lò xo. Khi có dòng điện chạy qua cuộn dây sẽ tạo ra từ trường hút piston và kéo nó vào trong cuộn dây. Piston có thể tạo ra lực tuyến tính. Hình vẽ 4.1 minh họa một solenoid.



Hình 4.1: Solenoid

Solenoid công nghiệp sử dụng điện áp 24VDC tạo ra dòng điện vài trăm mA.

4.3 VALVE.

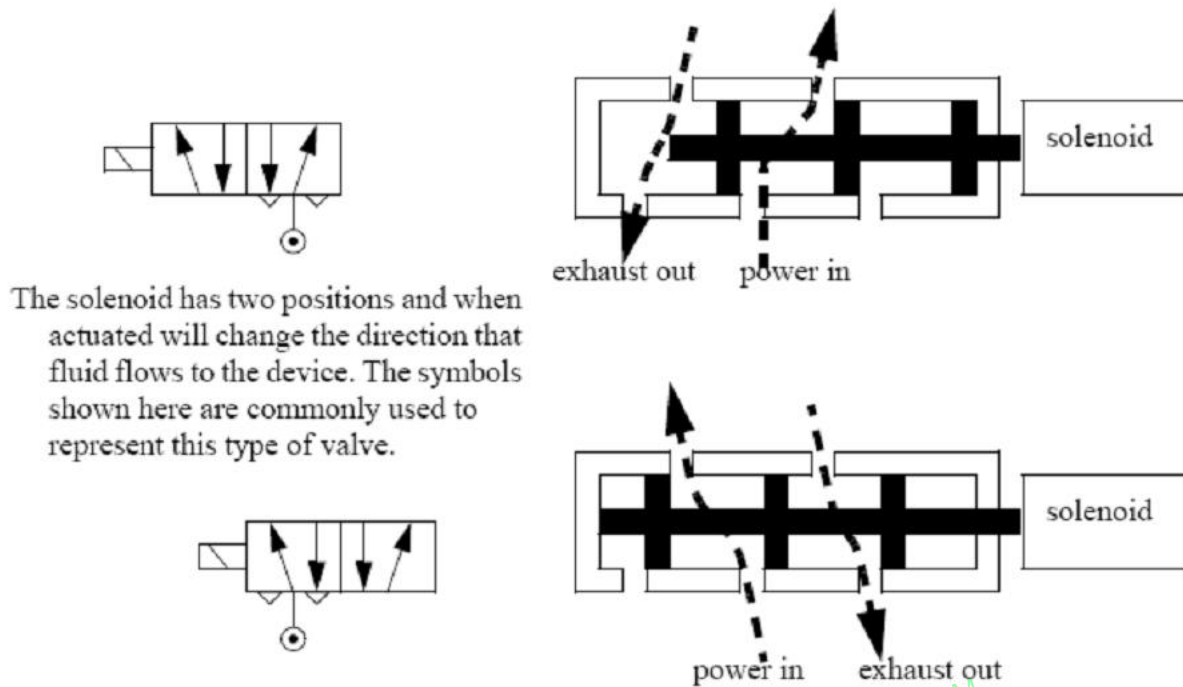
Dòng chảy của chất lỏng và chất khí sẽ được điều khiển bằng các Valve, được đóng ngắt bằng Solenoid. Xem hình vẽ 4.1.

Solenoid được gắn vào một mặt phẳng. Khi được kích thích nó sẽ tác động đến ống bên trái. Phía trên của valve có 2 port nối với các thiết bị, như xy lanh chất lỏng. Phía dưới của valve có 1 đường áp suất ở tâm và 2 ống xả 2 bên.

Trong hình vẽ trên, công suất đi qua tâm đến port phải của xy lanh. Port trái của xy lanh cho phép thoát qua ống xả.

Trong hình vẽ dưới, Solenoid ở vị trí mới nên áp suất được cấp đến port trái, còn port phải dùng để xả.

ĐIỀU KHIỂN LẬP TRÌNH 1



The solenoid has two positions and when actuated will change the direction that fluid flows to the device. The symbols shown here are commonly used to represent this type of valve.

Hình 4.2: Solenoid điều khiển Valve 5 port, 4 đường, 2 vị trí

Các loại Valve bao gồm:

- 2 đường thường đóng: 1 đường ra, 1 đường vào, bình thường valve đóng, khi có điện sẽ mở. Ứng dụng trong cho phép dòng chảy.
- 2 đường thường mở: dùng trong ngắt dòng chảy.
- 3 đường thường đóng: 1 vào, 1 ra, 1 ống xả, bình thường port ra nối với ống xả, khi có điện port vào sẽ nối với port ra.
- ...

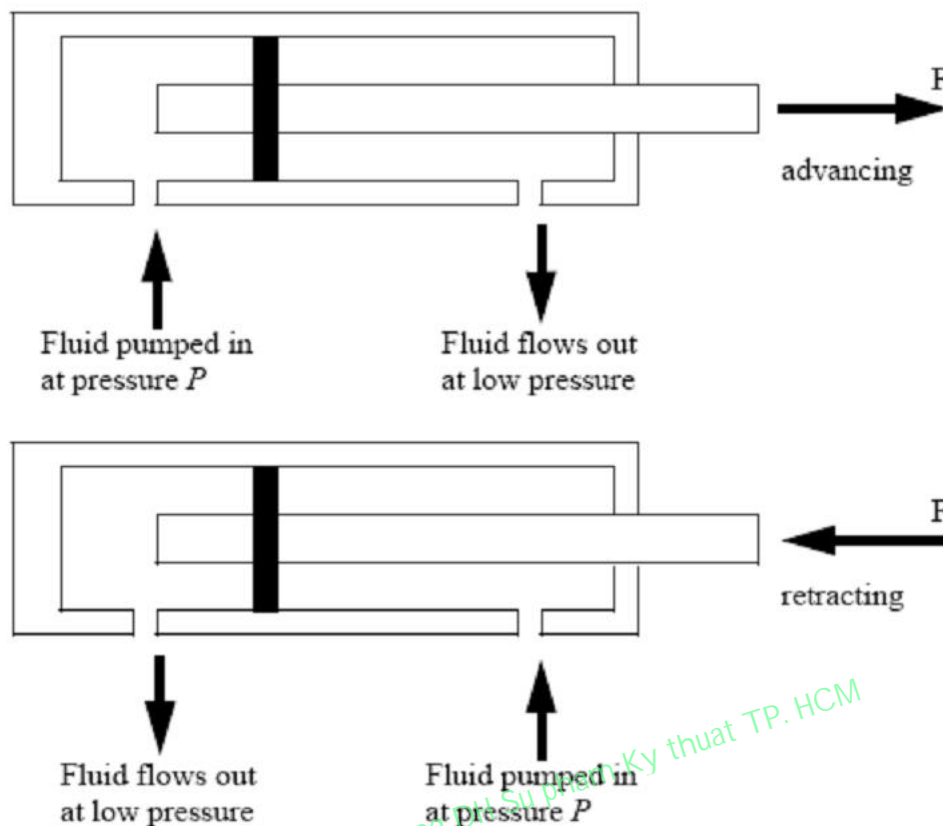
Khi chọn Valve cần quan tâm các vấn đề sau:

- Kích cỡ ống dẫn.
- Tốc độ dòng chảy.
- Áp suất vận hành.
- Điện áp Solenoid.
- Thời gian đáp ứng. 5-150ms

4.4 XY LANH Cylinder

Xy lanh dùng khí hoặc chất lỏng đã nén áp suất để tạo ra lực tuyến tính, như hình 4.3. Chất lỏng được bơm vào 1 bên của xy lanh làm nó nở ra đẩy piston. Chất lỏng sẽ thoát tự do ở mặt bên kia. Lực do xy lanh tạo ra tỉ lệ với tiết diện mặt cắt ngang của xy lanh. Xy lanh đơn tạo ra lực khi nở ra và có lò xo kéo piston lại. Xy lanh đôi tạo lực ở cả 2 hướng.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 4.3: Mặt cắt của xy lanh chất lỏng.

4.5 THỦY LỰC. Hydraulic

Thủy lực sử dụng các chất lỏng chưa nén để cung cấp lực rất lớn ở tốc độ chậm và di chuyển có giới hạn. Hệ thống thủy lực (thường là dầu) được nén bằng máy bơm, co qua valve để điều khiển xy lanh.

Hệ thống thủy lực bao gồm:

- Chất lỏng.
- Bồn chứa dầu.
- Bơm tạo áp suất.
- Valve điều khiển.
- Piston và xy lanh để tác động đến hệ thống cơ khí.

Chất lỏng thường là dầu không ăn mòn được chọn sao cho có thể bôi trơn máy móc. Dầu được chứa trong bồn như hình vẽ 4.4.

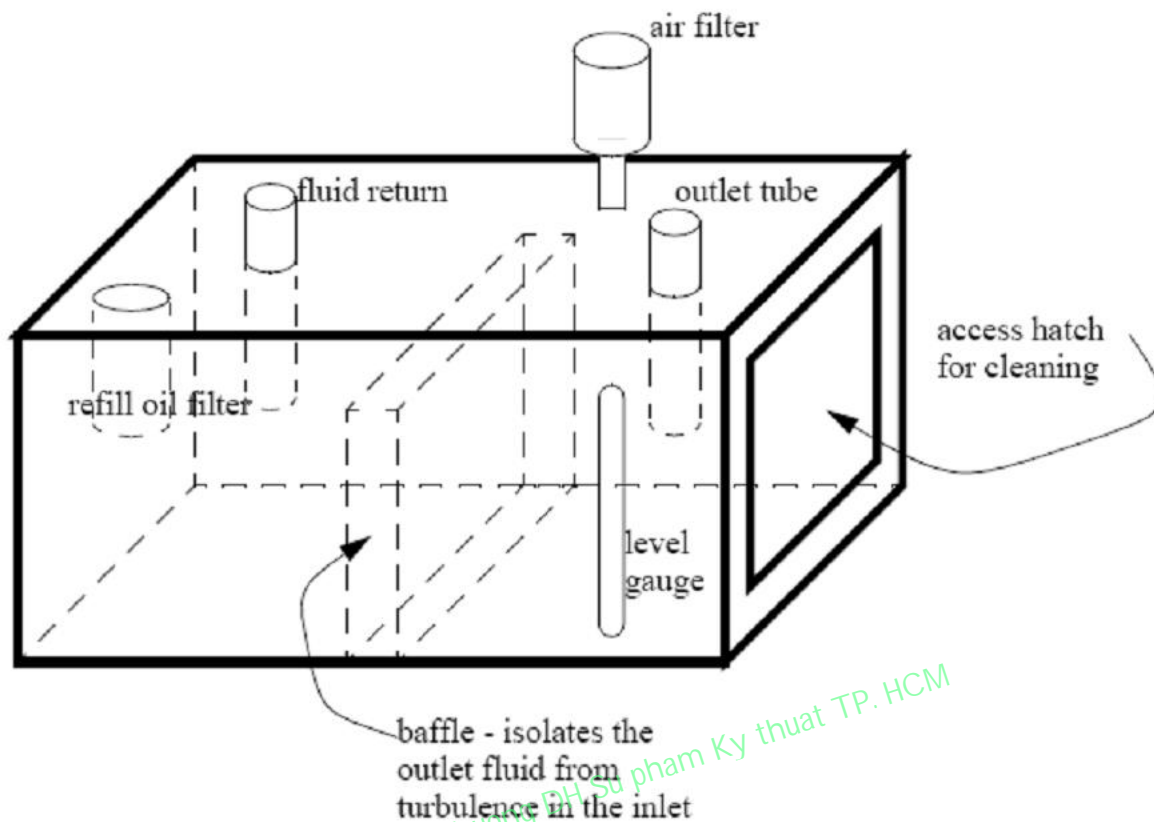
Chất lỏng được đưa đến bơm nén áp suất. Thường sử dụng bơm thay đổi số sao cho tạo chất lỏng áp suất cao và tốc độ chảy ổn định. Bộ điều tốc được đặt ở ngõ ra áp suất cao của bơm.

Nếu chất lỏng không chảy vào các phần khác của hệ thống thì nó sẽ quay về bồn chứa để giảm tổn hao trên bơm.

Chất lỏng áp suất cao được đưa qua Valve để có thể điều khiển đóng ngắt dòng chảy.

Hệ thống thủy lực rất hiệu quả đối với các ứng dụng công suất lớn nhưng nguy hiểm, bẩn và ồn do sử dụng chất lỏng áp suất cao.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 4.4: Bồn chứa chất lỏng

4.6 KHÍ NÉN. Pneumatic

Hệ thống khí nén rất thông dụng và có nhiều điểm chung với thủy lực. Tuy nhiên vẫn có một số khác biệt: không sử dụng bồn chứa vì không cần chứa thu nhận khí trở lại, sử dụng không khí nên dễ nén và không cần bộ điều tốc.

Hệ thống khí nén có đáp ứng rất nhanh, được sử dụng trong các ứng dụng lực nhỏ ở nhiều nơi trong nhà máy.

Một số đặc tính cơ bản của hệ thống khí nén bao gồm:

- Kích thước từ vài mm đến vài m.
- Thiết bị chấp hành nhỏ, có tính đàn hồi.
- Áp suất thường khoảng 85 psi.
- Xy lanh có khối lượng nhẹ.
- Thiết bị hỗ trợ có sẵn như máy nén khí và thiết bị quang.
- Sử dụng bộ chống rung và giảm va chạm ở các đầu xy lanh.

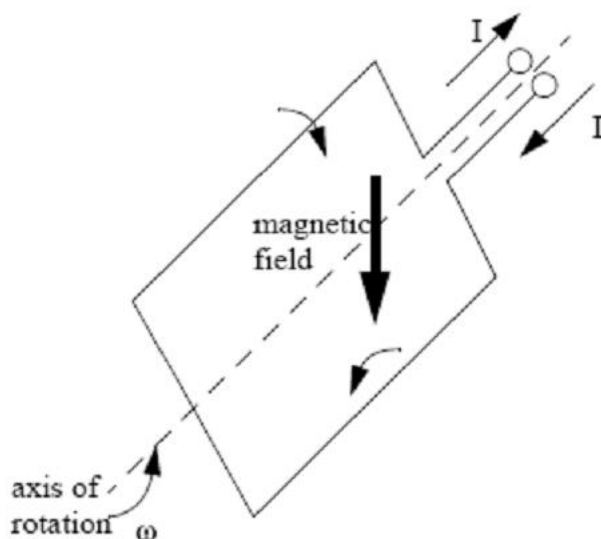
4.7 ĐỘNG CƠ Motor

4.7.1 Động Cơ DC Chổi Quét:

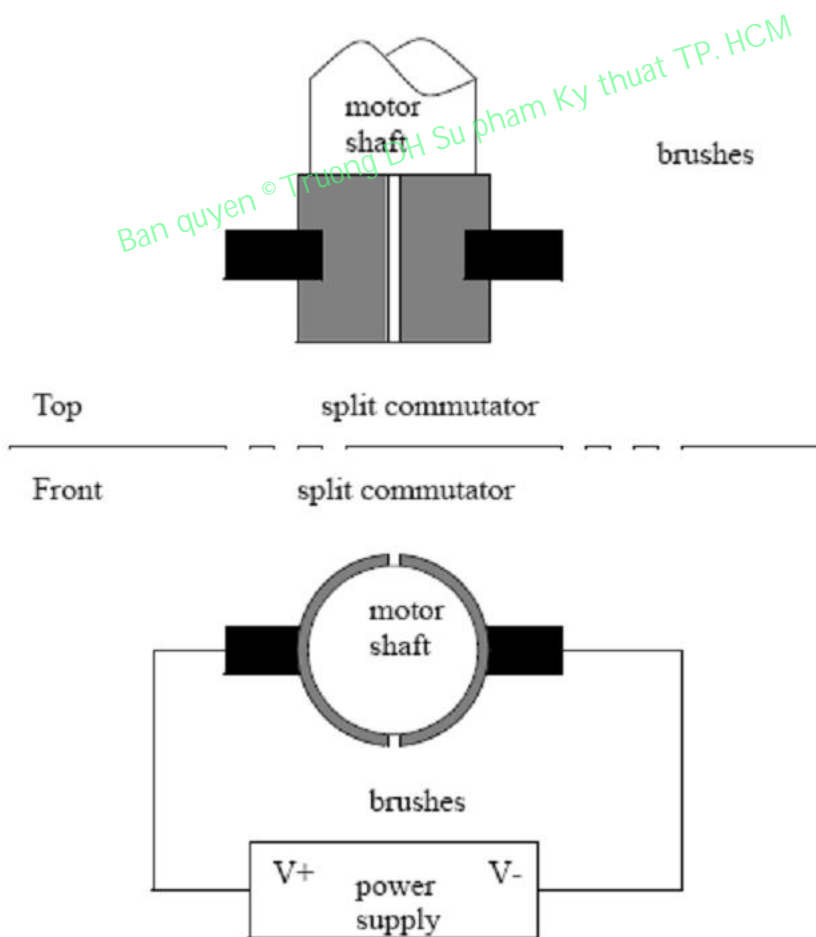
Động cơ DC có bộ dây quấn trên rotor quay bên trong startor, như hình 4.5.

Khi có dòng điện chạy qua vòng dây sẽ tạo ra một từ trường, năng lượng cấp cho rotor qua bộ chuyển mạch và chổi quét., như hình 4.6.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 4.5: Rotor của động cơ DC



Hình 4.6: Bộ chuyển mạch.

Bộ chuyển mạch được chia sao cho mỗi nửa vòng quay làm cho cực tính điện áp trên rotor và từ trường cảm ứng ngược với nam châm vĩnh cửu. Chiều quay động cơ được xác định bằng cực tính điện áp nguồn, tốc độ tỉ lệ với giá trị điện áp. Dùng bộ điều khiển hồi tiếp để xác định vị trí và điều chỉnh tốc độ động cơ.

ĐIỀU KHIỂN LẬP TRÌNH 1

4.7.2 Động Cơ AC Đồng Bộ.

Động cơ đồng bộ có dây quấn trên Startor, Rotor lồng sóc. Lồng sóc là 1 lõi nhôm khi đặt vào từ trường thay đổi sẽ tạo ra trường ngược lại. khi cấp nguồn AC đến cuộn dây Startor sẽ tạo ra một từ trường AC. Lồng sóc sẽ tạo ra từ trường ngược lại và tạo ra moment xoắn làm động cơ quay.

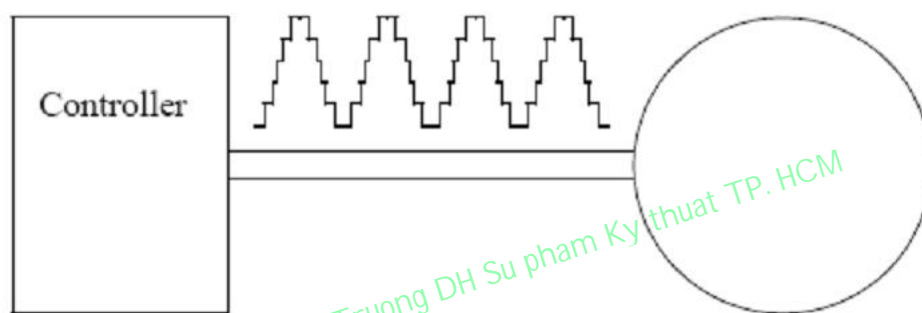
Động cơ gọi là đồng bộ vì có nó quay với tần số gần bằng tần số nguồn.

Ta điều khiển tốc độ động cơ AC bằng cách điều khiển tần số nguồn AC. Tín hiệu điều khiển tốc độ động cơ AC có dạng như hình vẽ 4.7.

Thực tế, động cơ đạt moment quay max nhỏ hơn tốc độ đồng bộ.

Ví dụ động cơ có 2 cực có tốc độ đồng bộ $2 \times 60 \times 60 / 2 = 3600$ vòng/phút nhưng chỉ đạt vận tốc 3520 vòng/phút.

$$\text{Tốc độ vòng/phút} = f \cdot 120 / p \qquad \text{Tốc độ lý tưởng.}$$



Hình 4.7: Điều khiển tốc độ động cơ AC

4.7.3 Động Cơ DC Không Có Chổi Quét.

Động cơ DC không dùng chổi quét mà sử dụng nam châm điện trên rotor, ta phải quấn dây cho startor, nên không cần chổi quét và bộ chuyển mạch. Không sử dụng chổi quét nên không cần bảo dưỡng như động cơ chổi quét.

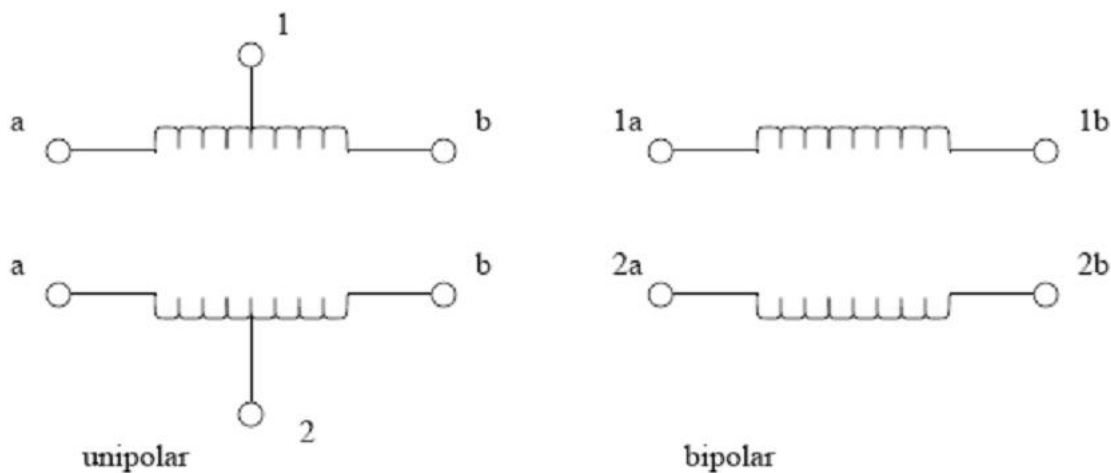
Để động cơ quay liên tục dòng điện của cuộn dây bên ngoài phải thay đổi liên tục. Nếu nguồn cấp là AC thì động cơ sẽ hoạt động như động cơ AC.

4.7.4 Động Cơ Bước.

Động cơ bước được thiết kế trong điều khiển định vị trí. Chúng di chuyển mỗi lần một bước, thường có góc quay là 1.8° – tương ứng 200 bước/vòng. Những loại khác có góc quay $2^{\circ}, 2.5^{\circ}, 5^{\circ}, 15^{\circ}, 30^{\circ}$.

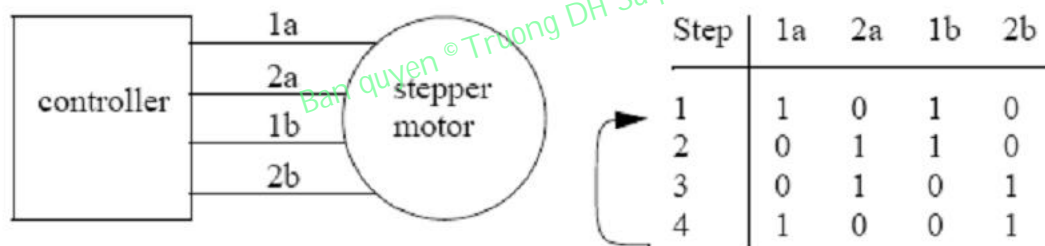
Có 2 loại động cơ bước: cực đơn và cực kép như hình vẽ 4.8.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 4.8: Sơ đồ dây quấn động cơ bước.

Loại đơn cực sử dụng cuộn dây có nhánh rẽ ở tâm và dùng nguồn đơn. Loại cực kép có cấu tạo đơn giản hơn nhưng phải dùng nguồn đôi, mạch điều khiển phức tạp hơn. Động cơ bước quay được nhờ điện áp lệch ở các đầu dây. Các giá trị điện áp khác nhau cấp cho động cơ đơn cực minh họa ở hình 4.9



To turn the motor the phases are stepped through 1, 2, 3, 4, and then back to 1. To reverse the direction of the motor the sequence of steps can be reversed, eg. 4, 3, 2, 1, 4, If a set of outputs is kept on constantly the motor will be held in position.

Hình 4.8: Tuần tự điều khiển động cơ bước đơn cực.

Khi động cơ mở, điện áp cung cấp các đầu dây có giá trị theo hàng 1. Để động cơ quay, ta phải thay đổi áp theo đường 2, rồi đường 3, đường 4, lặp lại. Nếu đảo tuần tự này động cơ sẽ quay theo chiều ngược lại. Động năng của động cơ và tải sẽ giới hạn tốc độ cực đại của các chuyển mạch đóng ngắt điện áp các đầu dây. Thông thường khoảng vài ngàn bước mỗi giây.

Khi không thay đổi, điện áp các đầu dây sẽ giữ động cơ ở 1 vị trí nào đó. Động cơ bước không cần bộ phận hồi tiếp, trừ khi sử dụng trong các ứng dụng có độ tin cậy rất cao.

Động cơ bước sẽ bị lệch khi moment giữ bị vượt hoặc khi nó tăng tốc quá nhanh. Khi động cơ bị lệch nó sẽ quay một góc nào đó không xác định từ vị trí hiện tại, muốn xác định giá trị này cần có hệ thống hồi tiếp vị trí.

ĐIỀU KHIỂN LẬP TRÌNH 1

CHƯƠNG 5: THIẾT KẾ CHƯƠNG TRÌNH THEO LƯU ĐỒ

Topics:

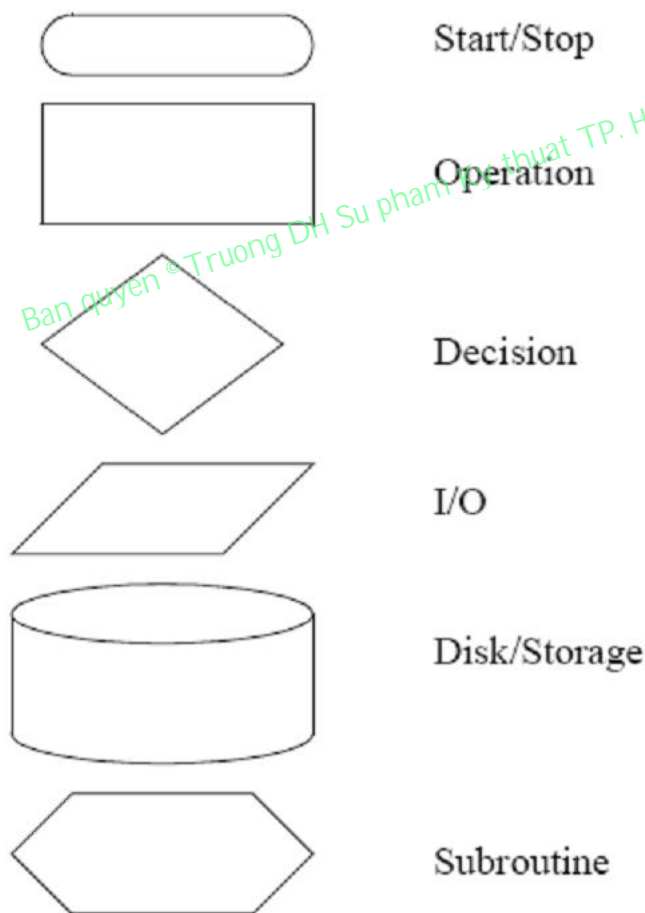
- Describing process control using flowcharts
- Conversion of flowcharts to ladder logic

Objectives:

- Be able to describe a process with a flowchart.
- Be able to convert a flowchart to ladder logic.

5.1 GIỚI THIỆU.

Một quá trình có các bước xử lý tuần tự sẽ thích hợp khi sử dụng lưu đồ để thiết kế chương trình. Các bước trong lưu đồ được thực hiện theo một trình tự đơn giản. các ký hiệu dùng trong lưu đồ bao gồm:



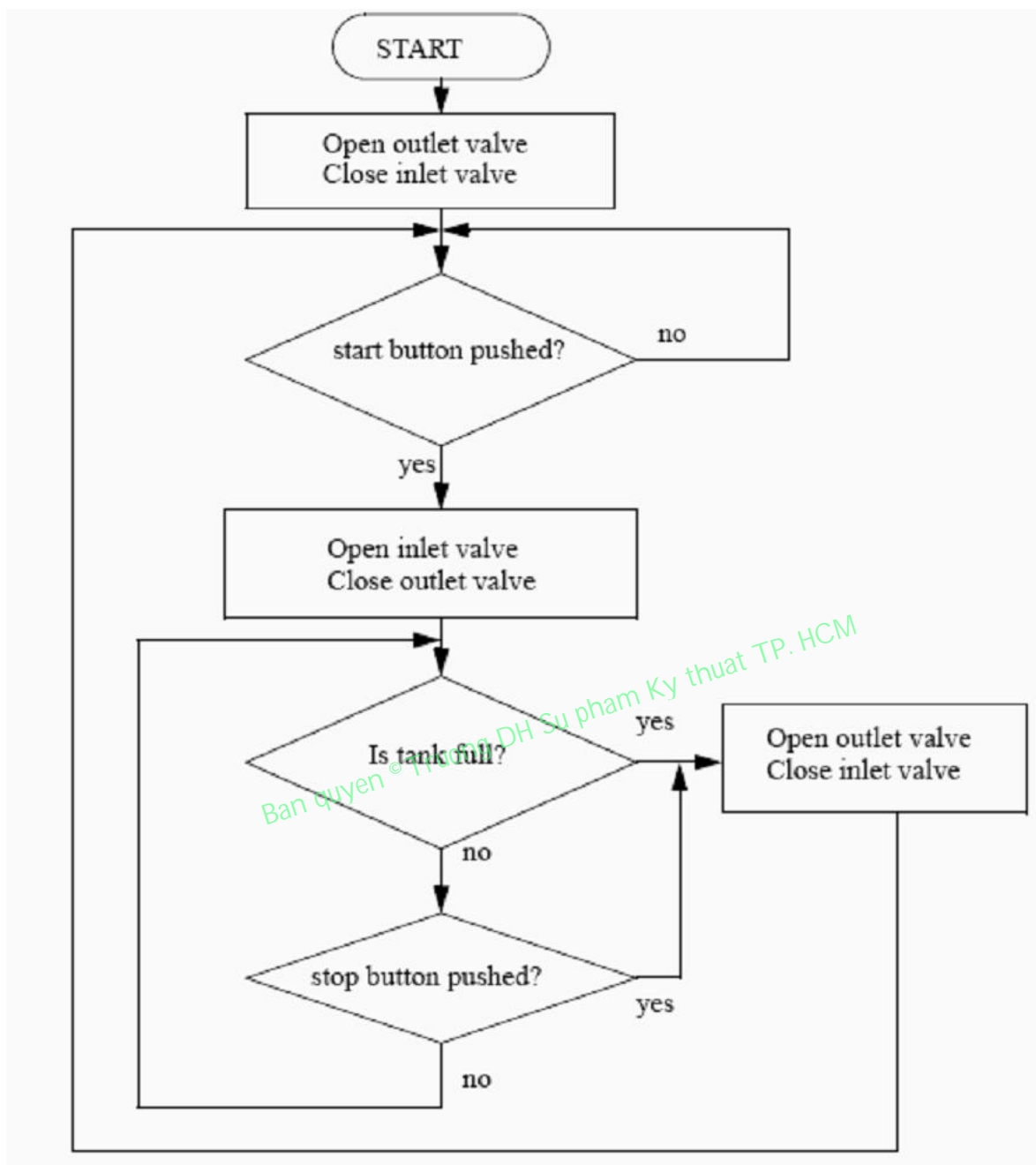
Hình 5.1: Ký hiệu dùng trong lưu đồ

Các khối được nối với nhau bằng các mũi tên nhằm chỉ ra các bước thực hiện tuần tự. Các khối khác nhau diễn tả các lệnh khác nhau.

Chương trình PLC luôn bắt đầu bằng khối Start và ít khi sử dụng khối Stop vì chương trình luôn chạy liên tục.

Hình 5.2 mô tả lưu đồ của hệ thống điều khiển một bồn nước.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 5.2: Lưu đồ điều khiển bồn nước

Khi nhấn nút Start, bồn bắt đầu cho nước vào và tắt đường chảy ra. Khi bồn đầy nước, hoặc nhấn nút Stop sẽ mở đường chảy ra và đóng đường chảy vào. Trong lưu đồ, quá trình bắt đầu từ trên cùng. Đầu tiên là mở van ngõ ra và đóng van ngõ vào. Tiếp theo, khối Decision sẽ chờ xem có nút nào được nhấn không. Nếu có nút được nhấn, theo nhánh Yes sẽ mở van vào và đóng van ra. Tiếp theo đến một vòng gồm hai khối Decision để chờ đến khi bồn đầy hoặc nhấn nút Stop. Nếu một trong hai trường hợp xảy ra thì đóng van vào và mở van ra. Và hệ thống sẽ quay lại chờ nút Start nhấn lần nữa. Khi vận hành, chương trình sẽ luôn chạy nên chỉ cần khối Start.

ĐIỀU KHIỂN LẬP TRÌNH 1

PHƯƠNG PHÁP CHUNG ĐỂ XÂY DỰNG LƯU ĐỒ:

- Hiểu quá trình hoạt động của hệ thống.
- Xác định các hoạt động chính, vẽ thành các khối.
- Xác định tuần tự vận hành, vẽ bằng các mũi tên.
- Khi tuần tự này thay đổi thì sử dụng các khối Decision để rẽ nhánh.

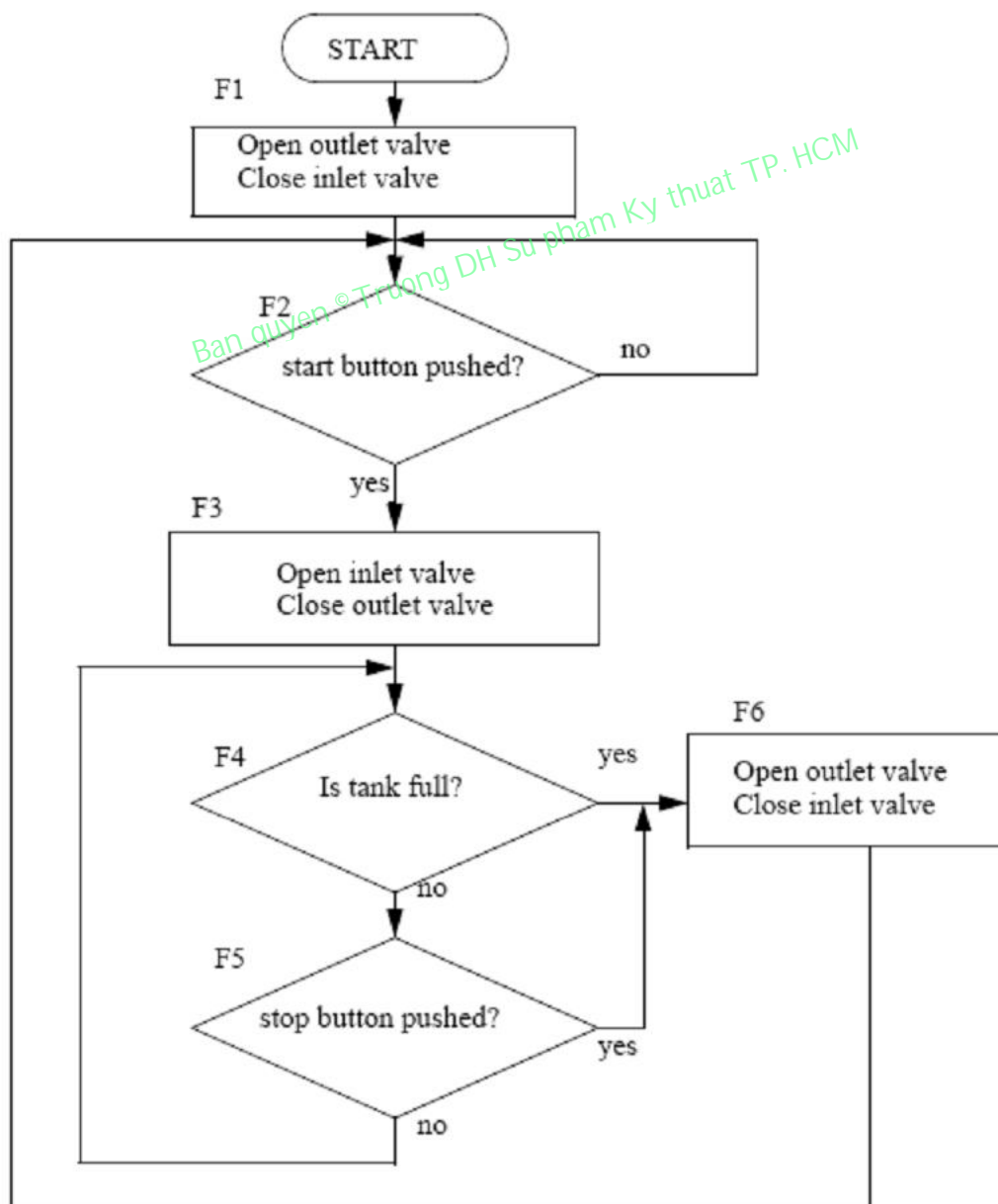
Mỗi lưu đồ sẽ được viết thành một chương trình LAD. Có 2 kỹ thuật cơ bản được sử dụng cho việc này:

- Sử dụng các khối mã logic bậc thang.
- Sử dụng logic bậc thang thông thường.

5.2 PHƯƠNG PHÁP BLOCK LOGIC.

Bước đầu tiên của phương pháp này là đặt tên các khối trong lưu đồ, như hình 5.3. Mỗi bước đã được đánh dấu sẽ được chuyển thành chương trình logic bậc thang.

BƯỚC 1: ĐẶT TÊN CÁC KHỐI TRONG LƯU ĐỒ



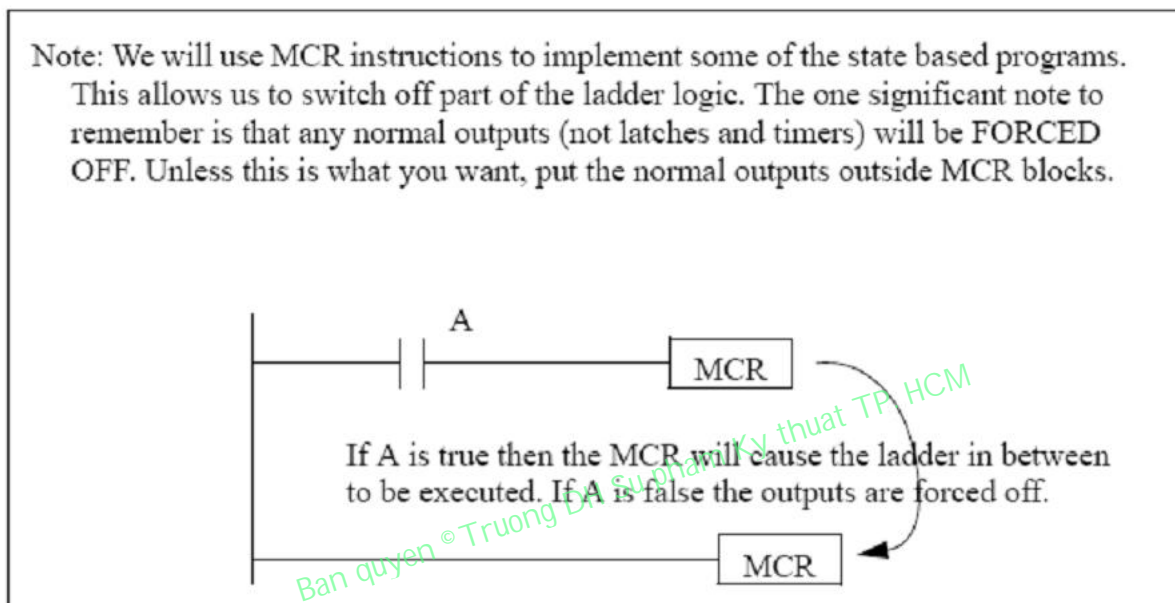
Hình 5.3: Đặt tên các khối trong lưu đồ.

ĐIỀU KHIỂN LẬP TRÌNH 1

Mỗi khối trong lưu đồ được chuyển thành một khối của logic bậc thang. Để thực hiện việc này ta sử dụng lệnh MCR (Master Control Relay). Lệnh này được trình bày ở hình 5.4, bao gồm một cặp ngõ ra MCR.

Nếu dòng MCR đầu tiên trong lệnh này đúng thì các logic bậc thang trong các dòng tiếp theo sẽ được quét bình thường cho đến lệnh MCR thứ hai.

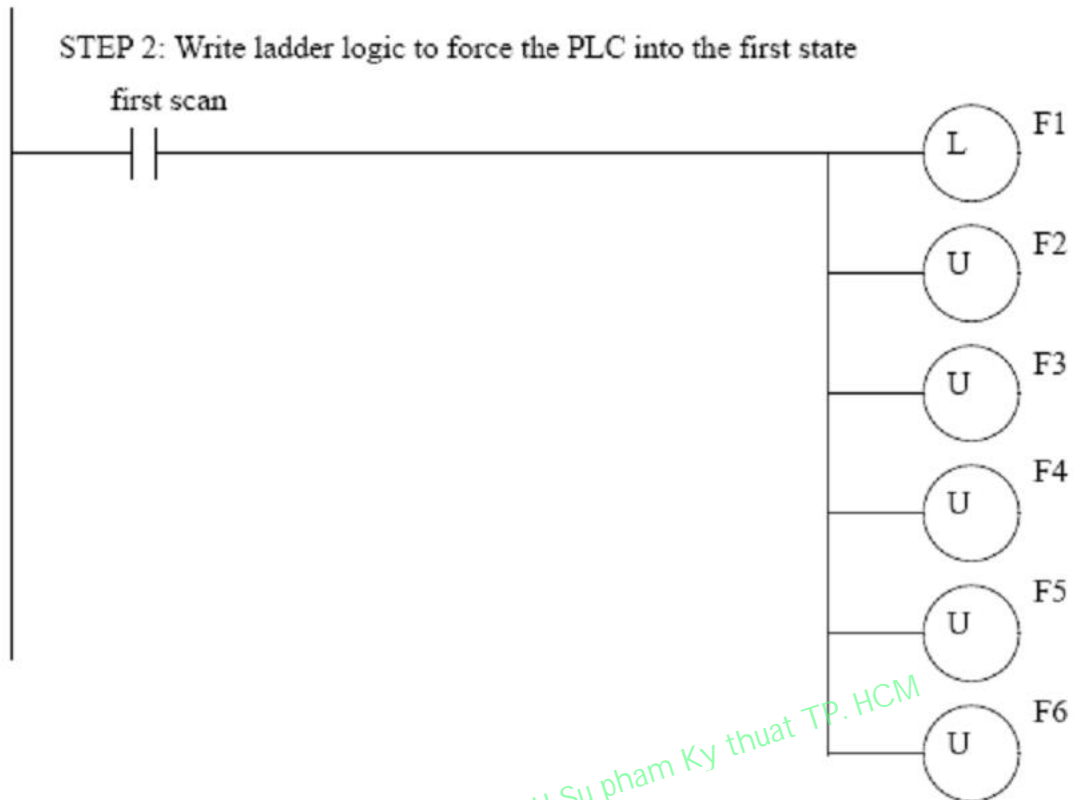
Nếu dòng MCR đầu sai thì các dòng logic bậc thang tiếp theo sẽ tắt. Nếu trong khối MCR có sử dụng một ngõ ra bình thường thì nó cũng bị tắt, nên ta phải sử dụng các lệnh chốt đối với phương pháp này.



Hình 5.4: Lệnh MCR

BƯỚC 2: VIẾT LOGIC BẬC THANG ĐỂ PLC Ở TRẠNG THÁI ĐẦU TIÊN
 Như hình 5.5.

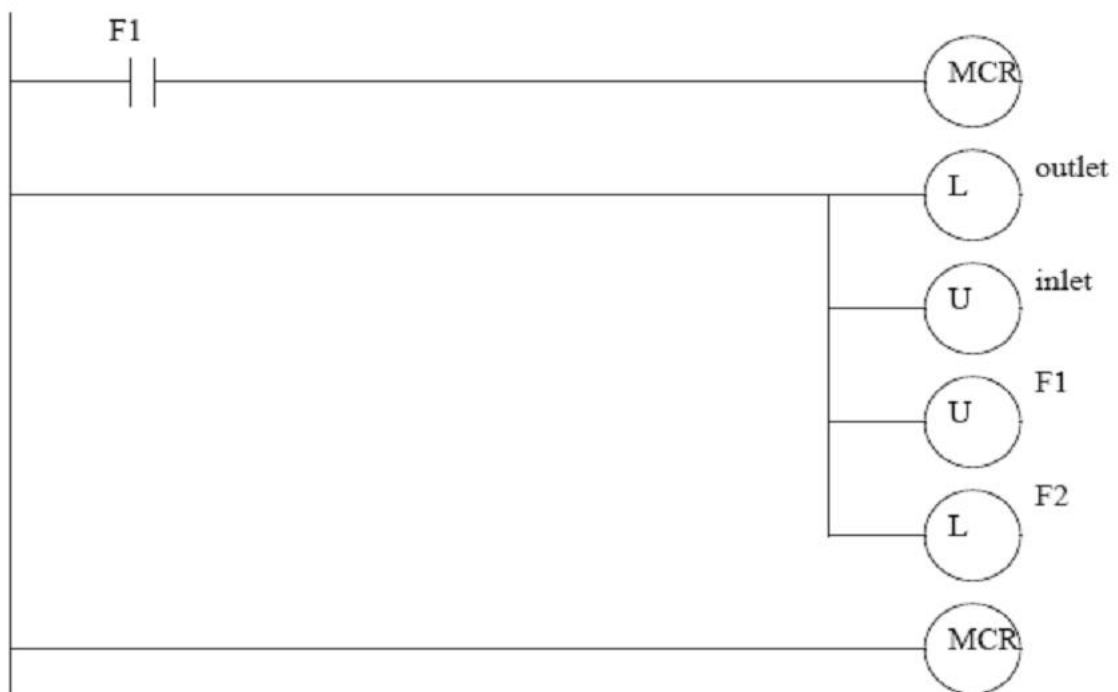
ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 5.5: Đặt logic bậc thang vào trạng thái đầu tiên

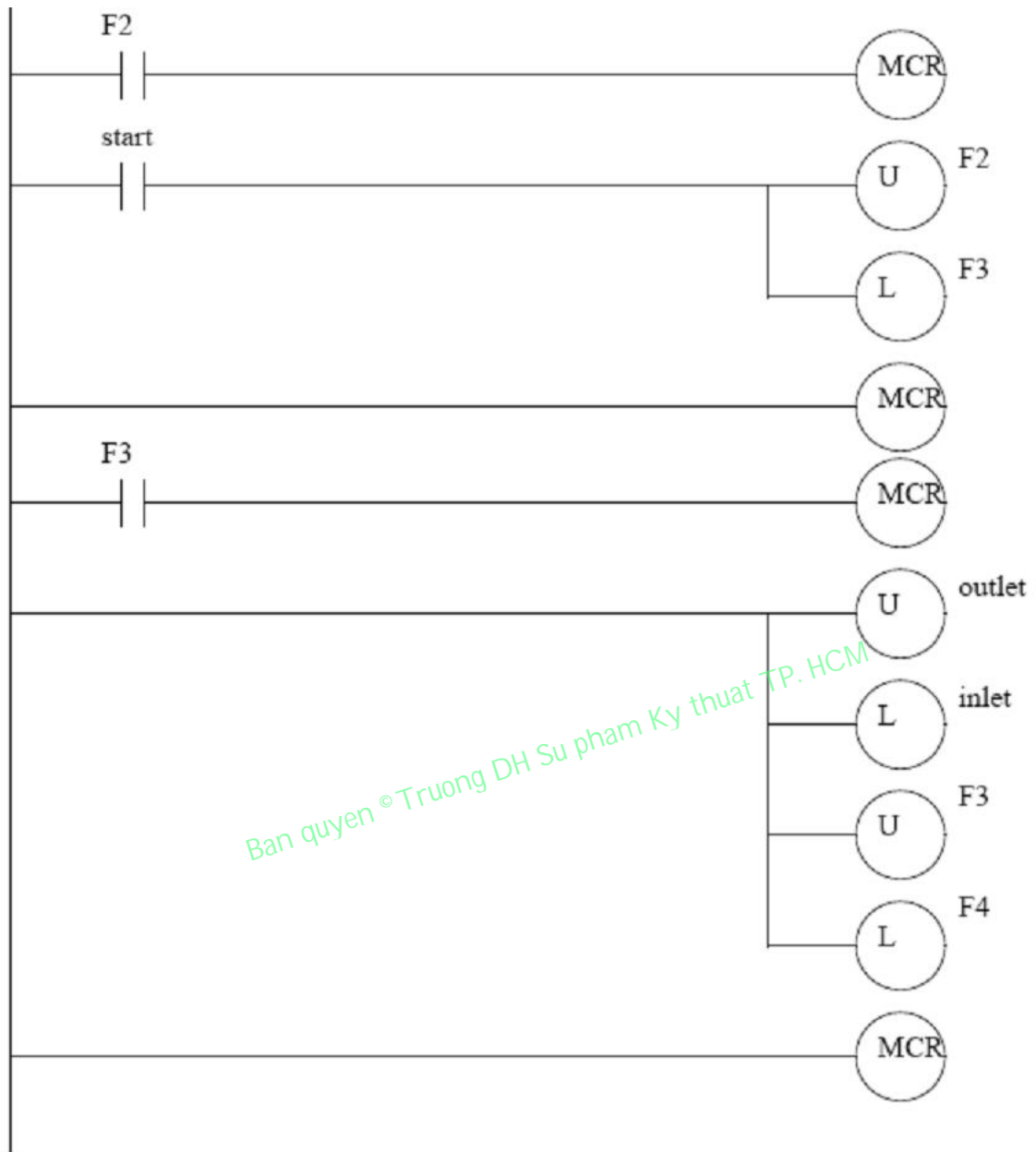
BƯỚC 3: VIẾT LOGIC BẬC THANG CHO CÁC HÀM TRONG LƯU ĐỒ

STEP 3: Write ladder logic for each function in the flowchart



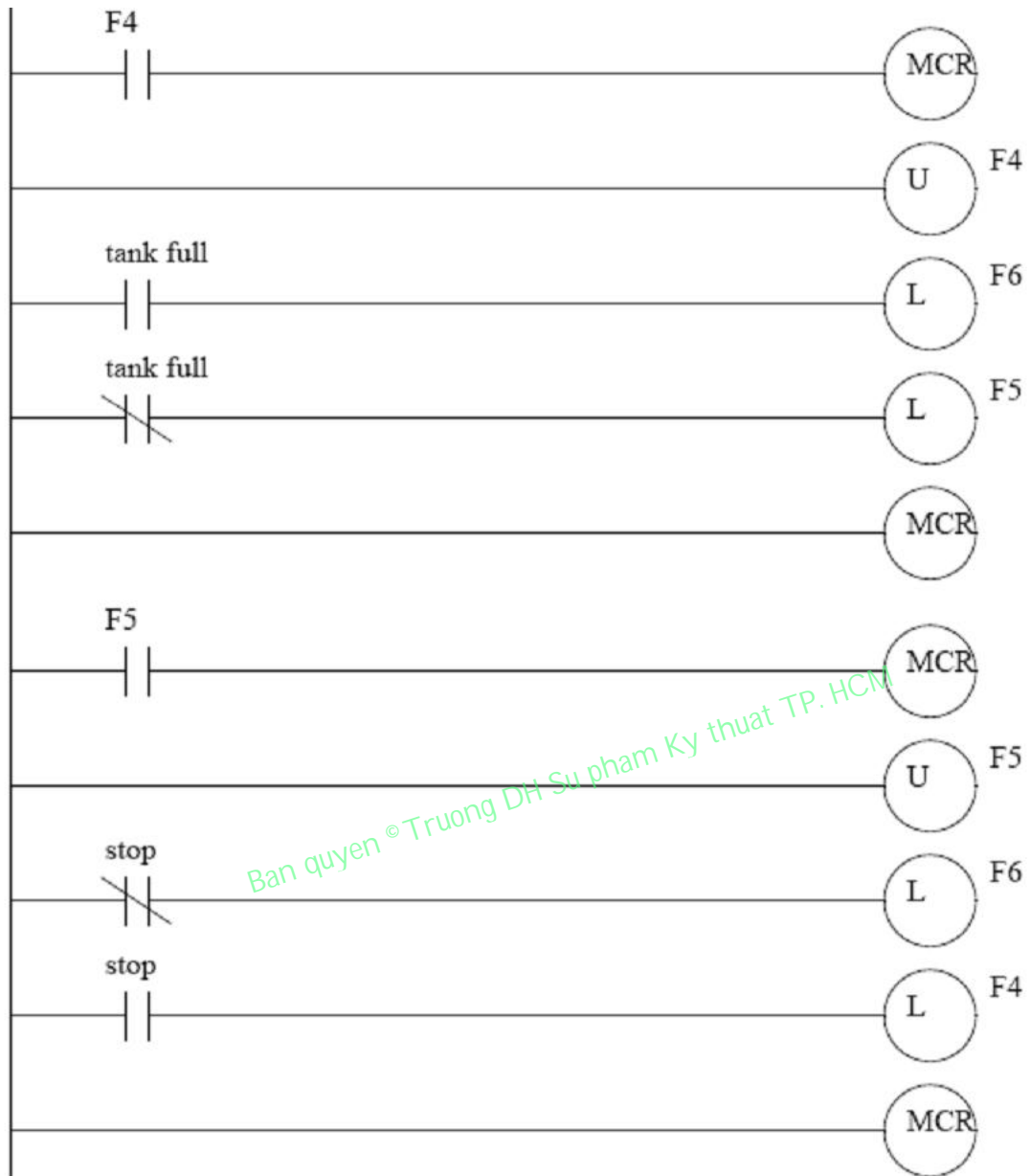
Hình 5.6: Logic bậc thang cho bước F1.

ĐIỀU KHIỂN LẬP TRÌNH 1



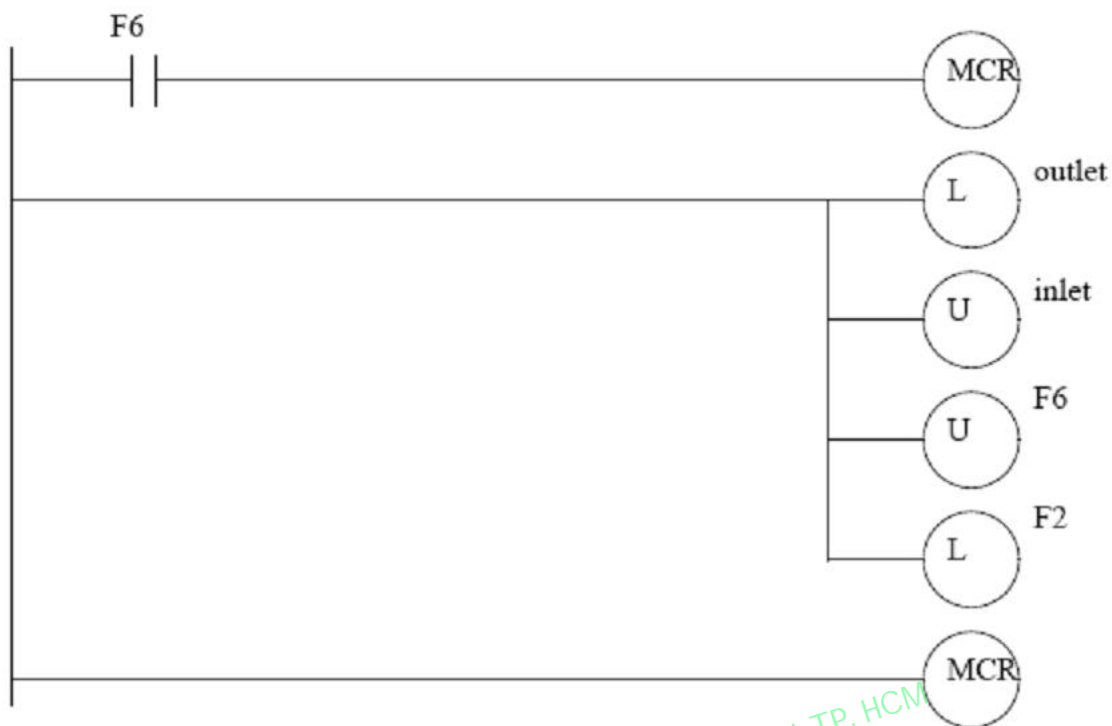
Hình 5.7: Logic bậc thang cho bước F2 và F3.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 5.8: Logic bậc thang cho bước F4 và F5

ĐIỀU KHIỂN LẬP TRÌNH 1



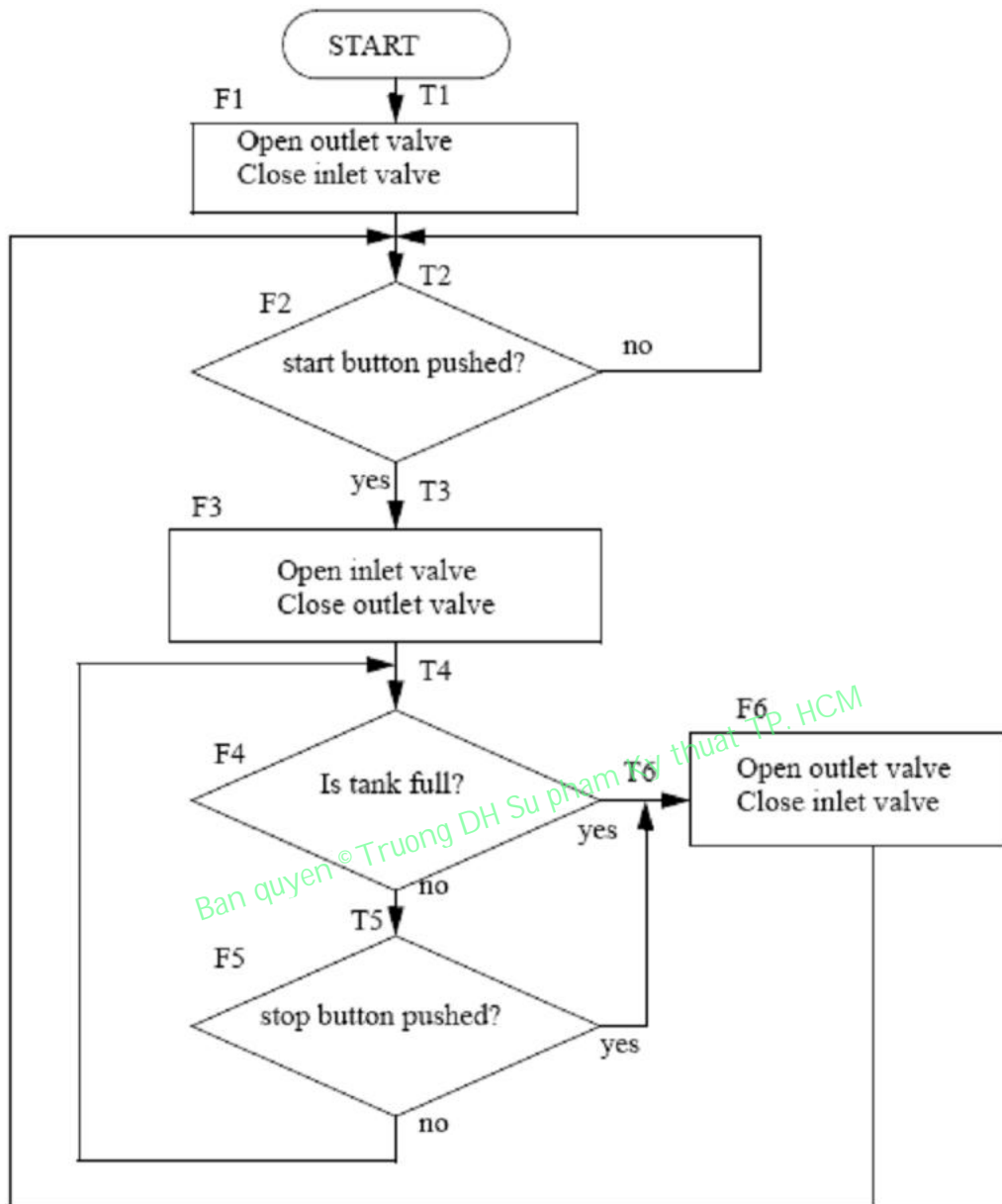
Hình 5.8: Logic bậc thang cho bước F6

5.3 PHƯƠNG PHÁP SEQUENCE BIT.

Ta thường chọn cách điều khiển không sử dụng lệnh MCR . Lưu đồ trong các ví dụ trước có thể được thực hiện theo cách khác như sau:

Bước đầu tiên được vẽ trên hình 5.9.

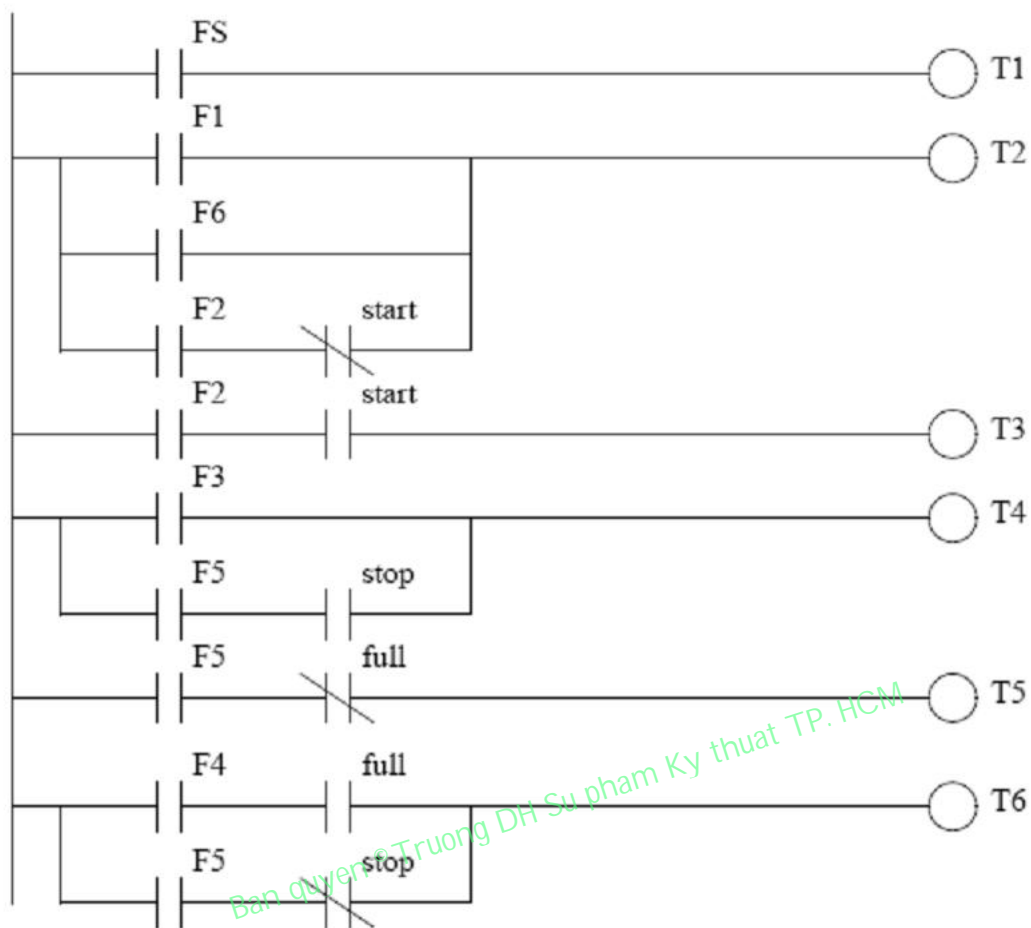
ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 5.9: Đặt tên các khối và mũi tên cho lưu đồ.

Đoạn chương trình logic bậc thang đầu tiên được cho trên hình 5.10.

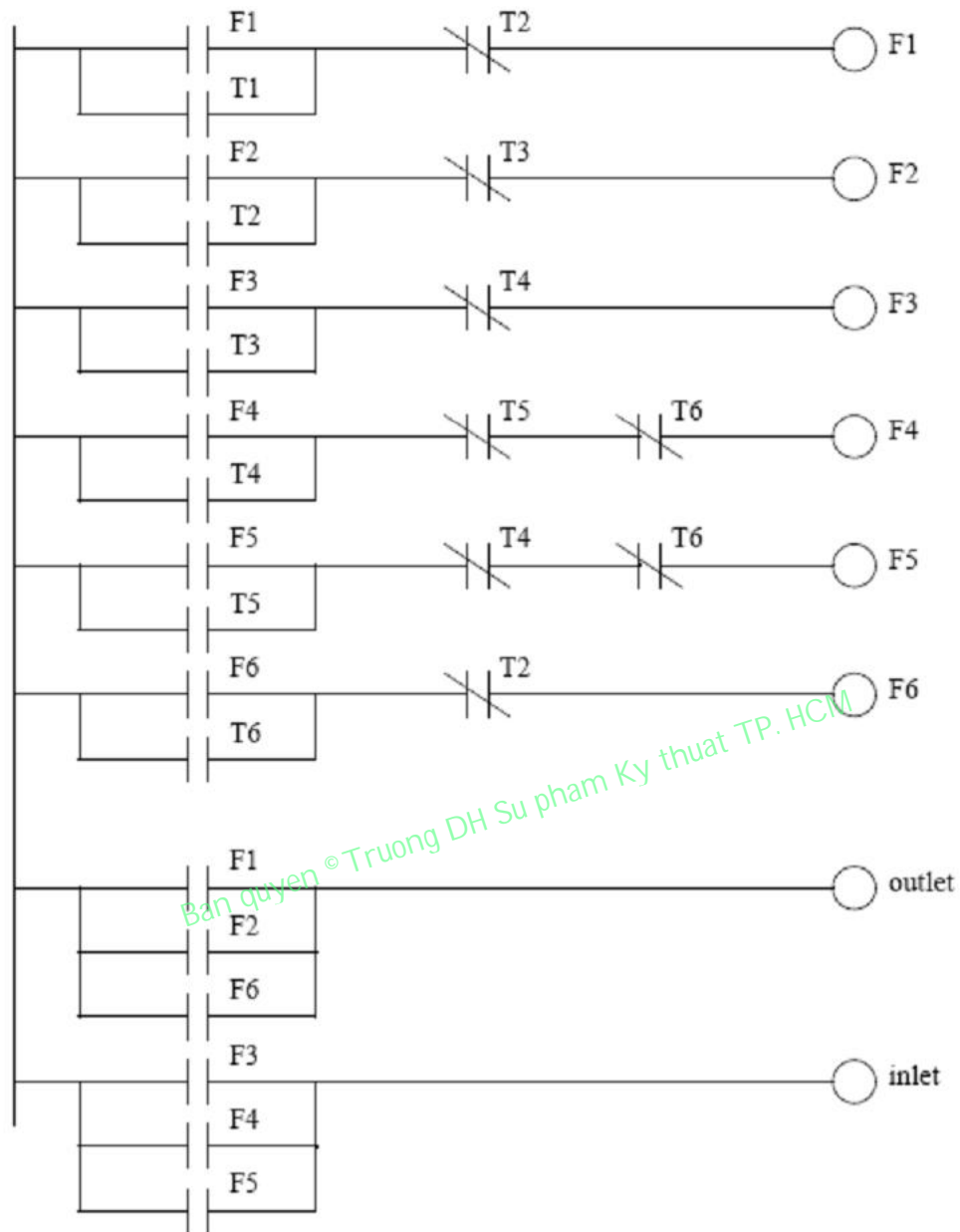
ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 5.10: Logic chuyển đổi.

Logic bậc thang trong hình 5.11 sẽ kích hoạt một hàm hoặc chuyển tới hàm tiếp theo.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 5.11: Hàm logic bậc thang và Các ngõ ra.

ĐIỀU KHIỂN LẬP TRÌNH 1

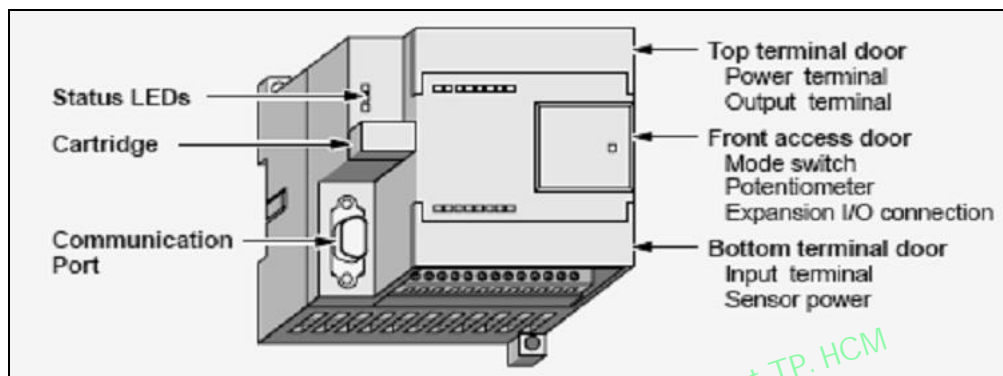
CHƯƠNG 6: PLC S7 – 200

6.1 CẤU TRÚC PHẦN CỨNG.

6.1.1 Đặc điểm chung.

S7-200 là thiết bị điều khiển logic lập trình loại nhỏ của hãng Siemens (Đức), có cấu trúc theo kiểu module và có các module mở rộng. Các module này được sử dụng cho nhiều ứng dụng lập trình khác nhau.

Hình dạng bên ngoài của PLC S7-200 được mô tả như hình 6.1.



Hình 6.1: Bộ điều khiển lập trình S7-200

Đặc điểm và thông số của các loại PLC S7-200 khác nhau được giới thiệu trong bảng sau:

Đặc trưng	CPU 221	CPU 222	CPU 224	CPU 226
Kích thước(mm)	90x80x62	90x80x62	120.5x80x62	190x80x62
Bộ nhớ chương trình	2048 words	2048words	4096words	4096words
Bộ nhớ dữ liệu	1024 words	1024words	2560words	2560words
Cổng logic vào	6	8	14	24
Cổng logic ra	4	6	10	16
Modul mở rộng	None	2	7	7
Digital I/O cục đại	128/128	128/128	128/128	128/128
Analog I/O cục đại	None	16In/16Out	32In/32Out	32In/32Out
Bộ đếm (Counter)	256	256	256	256
Bộ định thì (Timer)	256	256	256	256
Tốc độ thực thi lệnh	0.37μs	0.37μs	0.37μs	0.37μs
Khả năng lưu trữ khi mất điện	50 giờ	50 giờ	190 giờ	190 giờ

6.1.2 Các đèn báo.

- SF (System Failure): Đèn đỏ SF báo hiệu hệ thống bị hỏng.
- RUN: Đèn xanh RUN chỉ định PLC đang ở chế độ làm việc và thực hiện chương trình được nạp vào trong máy.
- STOP: Đèn vàng STOP chỉ định rằng PLC đang ở chế độ dừng chương trình.

6.1.3 Các ngõ vào.

- Kiểu đầu vào IEC 1131-2 hoặc SIMATIC.
- Điện áp mức logic 1: 15-30VDC, dòng nhỏ nhất 4mA; 35VDC ở thời gian tức thời 500ms.

ĐIỀU KHIỂN LẬP TRÌNH 1

- Trạng thái mức logic 1 chuẩn: 24 VDC, 7mA.
- Trạng thái mức logic 0: Tối đa 5 VDC, 1mA.
- Đáp ứng thời gian lớn nhất ở các chân I0.0 đến I1.5: có thể chỉnh từ 0,2 đến 8,7ms. Thời gian mặc định 0,2ms.
- Sự cách ly về quang 500VAC.

6.1.4 Các ngõ ra.

- Kiểu đầu ra: Relay hoặc Transistor cấp dòng điện.
- Điện áp mức 1: 24.4 đến 28.8VDC.
- Dòng tải tối đa: 2A/ điểm; 8A/common.
- Quá dòng: 7A với contact đóng.
- Điện trở cách ly: nhỏ nhất 100 MΩ.
- Thời gian chuyển mạch: tối đa 10ms.
- Thời gian sử dụng: 10.000.000 lần với công tắc cơ khí; 100.000 lần với tốc độ tải.
- Điện trở công tắc: tối đa 200 mΩ.
- Chế độ bảo vệ ngắn mạch: không có.

6.1.5 Nguồn cung cấp.

- Điện áp nguồn cung cấp: 20.4 đến 24.8VDC
- Dòng vào max load: 900mA tại 24VDC
- Cách ly điện ngõ vào: không có
- Thời gian duy trì khi mất nguồn: 10ms ở 24 VDC
- Cầu chì bên trong: 2A, 250V

6.1.6 Chế độ làm việc.

PLC có 3 chế độ làm việc:

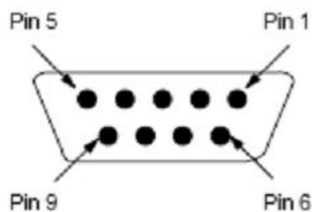
- **RUN**: cho phép PLC thực hiện chương trình từ bộ nhớ, PLC sẽ chuyển từ RUN sang STOP nếu trong máy có sự cố hoặc trong chương trình gặp lệnh STOP.
- **STOP**: cưỡng bức PLC dừng chương trình đang chạy và chuyển sang chế độ STOP.
- **TERM**: cho phép máy lập trình tự quyết định chế độ hoạt động cho PLC ở chế độ RUN hoặc STOP.

6.1.7 Cổng truyền thông.

S7-200 sử dụng cổng truyền thông nối tiếp RS485 với đầu 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 (Point to Point Interface) là 9600 bauds. Tốc độ truyền của PLC theo kiểu tự do là 300 ÷ 38.400 bauds.

Sơ đồ chân cổng truyền thông vẽ trên hình 6.2.

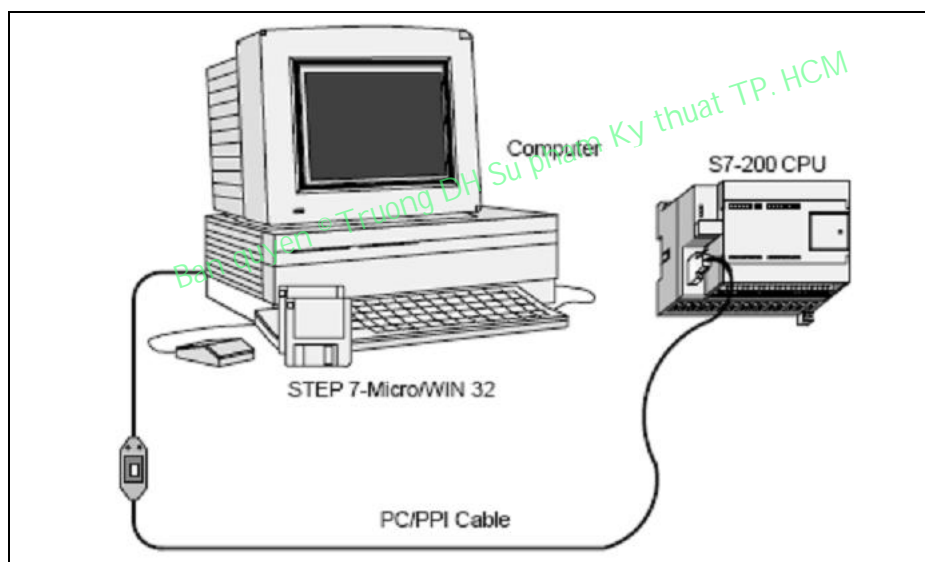
ĐIỀU KHIỂN LẬP TRÌNH 1



Chân	Chức năng
1	GND
2	24 VDC
3	Tín hiệu A của RS485 (RxD/TxD+)
4	RTS (theo mức TTL)
5	GND
6	+5 VDC
7	Nguồn cấp 24 VDC 120mA max
8	Tín hiệu B RS485 (RxD/TxD-)
9	Chọn lựa cách giao tiếp

Hình 6.2: Sơ đồ chân của cổng truyền thông

Để ghép S7-200 với các máy tính PC qua cổng RS-232 cần có cáp nối PC/PPI với bộ chuyển đổi từ RS232 sang RS485, theo hình vẽ 6.3.



Hình 6.3: Ghép nối S7-200 với máy tính qua cổng RS232

6.1.8 Các module mở rộng.

Có thể mở rộng ngõ vào/ra của PLC bằng cách ghép nối thêm vào PLC các module mở rộng về phía bên phải của CPU. CPU 224 có thể ghép nhiều nhất 7 module theo bảng 6.1.

Các module mở rộng Digital hay Analog đều chiếm chỗ trong bộ đệm, tương ứng với số đầu vào/ ra của các module.

Các module 5 và 6 dùng để kết nối mạng Profibus và AS-Interface.

ĐIỀU KHIỂN LẬP TRÌNH 1

	Module 0	Module 1	Module 2	Module 3	Module 4
CPU 224	4 In / 4 Out	8 In	4 AI/ 1 AQ	8 Out	4 AI/ 1 AQ
Process-image I/O register assigned to physical I/O:					
I0.0 Q0.0	I2.0 Q2.0	I3.0	AIW0 AQW0	Q3.0	AIW8 AQW4
I0.1 Q0.1	I2.1 Q2.1	I3.1	AIW2	Q3.1	AIW10
I0.2 Q0.2	I2.2 Q2.2	I3.2	AIW4	Q3.2	AIW12
I0.3 Q0.3	I2.3 Q2.3	I3.3	AIW6	Q3.3	AIW14
I0.4 Q0.4		I3.4		Q3.4	
I0.5 Q0.5		I3.5		Q3.5	
I0.6 Q0.6		I3.6		Q3.6	
I0.7 Q0.7		I3.7		Q3.7	
I1.0 Q1.0					
I1.1 Q1.1					
I1.2					
I1.3					
I1.4					
I1.5					

Bảng 2.1: Định địa chỉ cho các module mở rộng

6.2 NGUYÊN LÝ HOẠT ĐỘNG.

6.2.1 Đơn vị xử lý trung tâm CPU.

CPU điều khiển các hoạt động bên trong PLC. Bộ xử lý sẽ đọc và kiểm tra chương trình được chứa trong bộ nhớ, sau đó sẽ thực hiện tuần tự từng lệnh trong chương trình, kết quả làm đóng hay ngắt các ngõ ra. Các trạng thái ngõ ra ấy được đưa đến tác động các thiết bị bên ngoài. Toàn bộ các hoạt động này đều phụ thuộc vào chương trình điều khiển được lưu trữ trong bộ nhớ.

6.2.2 Hệ thống BUS.

Hệ thống Bus là tuyến truyền tín hiệu, hệ thống gồm nhiều đường tín hiệu song song:

- Address Bus: Bus địa chỉ dùng để truyền địa chỉ đến các Module khác nhau.
- Data Bus: Bus dùng để truyền dữ liệu.
- Control Bus: Bus điều khiển dùng để truyền các tín hiệu định thời và điều khiển đồng bộ các hoạt động trong PLC

Trong PLC dữ liệu được trao đổi giữa bộ vi xử lý và các module vào ra thông qua Data Bus. Address Bus và Data Bus gồm 8 đường, ở cùng thời điểm cho phép truyền đồng thời 8 bit của 1 byte, còn gọi là truyền song song 8 bit.

Nếu một module vào nhận được địa chỉ của nó trên Address Bus, nó sẽ chuyển tất cả trạng thái ngõ vào của nó vào Data Bus. Nếu một địa chỉ byte của 8 ngõ ra xuất hiện trên Address Bus, modul ra tương ứng sẽ nhận được dữ liệu từ Data bus. Control Bus sẽ chuyển các tín hiệu điều khiển vào chu trình hoạt động của PLC.

Các địa chỉ và dữ liệu được chuyển lên các Bus tương ứng trong thời gian hạn chế.

Hệ thống Bus sẽ làm nhiệm vụ trao đổi thông tin giữa CPU, bộ nhớ và I/O. Bên cạnh đó, CPU được cung cấp một xung Clock có tần số từ 1÷8 MHz. Xung này quyết định tốc độ hoạt động của PLC và cung cấp các yếu tố về định thời, đồng hồ của hệ thống.

ĐIỀU KHIỂN LẬP TRÌNH 1

6.2.3 Bộ nhớ.

PLC thường sử dụng bộ nhớ trong các trường hợp:

- Làm bộ lưu trữ tạm thời các bảng trạng thái I/O.
- Làm bộ đếm trạng thái các chức năng trong PLC như định thời, đếm, thanh ghi.

Mỗi lệnh của chương trình có một vị trí riêng trong bộ nhớ, tất cả mọi vị trí trong bộ nhớ đều được đánh số, những số này chính là địa chỉ trong bộ nhớ.

Địa chỉ của từng ô nhớ sẽ được trở đến bởi một bộ đếm địa chỉ ở bên trong bộ vi xử lý.

Bộ vi xử lý sẽ tăng giá trị bộ đếm này lên một trước khi xử lý lệnh tiếp theo. Với một địa chỉ mới, nội dung của ô nhớ tương ứng sẽ xuất hiện ở đầu ra, quá trình này được gọi là quá trình đọc.

Bộ nhớ bên trong PLC được tạo bởi các vi mạch bán dẫn, mỗi vi mạch này có khả năng chứa 2000 ÷ 16000 dòng lệnh, tùy loại vi mạch. Trong PLC các bộ nhớ như RAM, EEPROM đều được sử dụng.

- RAM (Random Access Memory) có thể nạp chương trình, thay đổi hay xóa bỏ nội dung bất kỳ lúc nào. Nội dung của RAM sẽ bị mất nếu nguồn điện nuôi bị mất. Để tránh tình trạng này các PLC đều được trang bị một pin khô, có khả năng cung cấp năng lượng dự trữ cho RAM từ vài tháng đến vài năm. Trong thực tế RAM được dùng để khởi tạo và kiểm tra chương trình. Khuynh hướng hiện nay dùng CMOSRAM nhờ khả năng tiêu thụ năng lượng thấp và tuổi thọ lớn.
- EEPROM (Electrically Erasable Programmable Read Only Memory) là bộ nhớ mà người sử dụng bình thường chỉ có thể đọc chứ không ghi nội dung vào được. Nội dung của EEPROM không bị mất khi mất nguồn, nó được gắn sẵn trong máy, đã được nhà sản xuất nạp và chứa hệ điều hành sẵn. Nếu người sử dụng không muốn mở rộng bộ nhớ thì chỉ dùng EEPROM gắn bên trong PLC.
- Ngoài ra PLC còn cho phép gắn thêm bộ nhớ mở rộng như RAM, EPROM ở khe Cartridge.

6.3 CẤU TRÚC BỘ NHỚ.

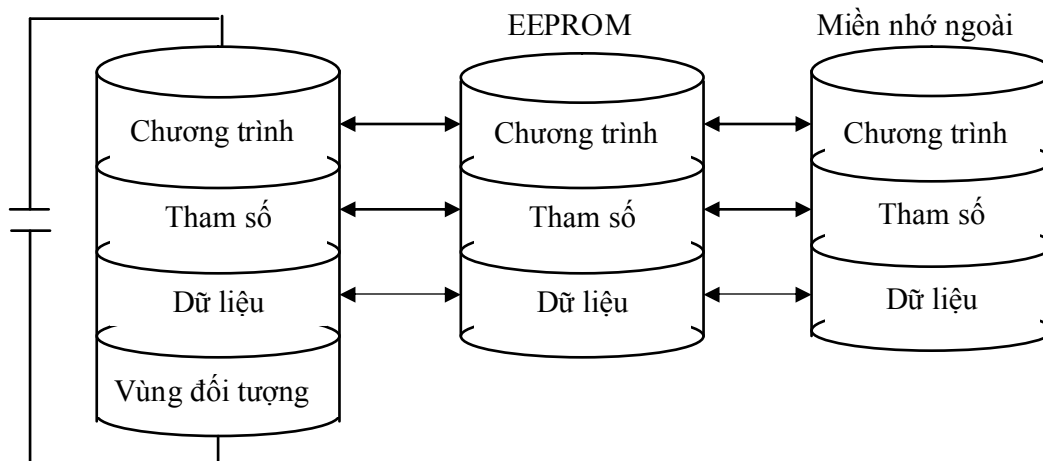
6.3.1 Phân chia bộ nhớ.

Bộ nhớ của S7-200 được chia thành 4 vùng có một tụ điện làm nhiệm vụ duy trì dữ liệu trong một khoảng thời gian nhất định khi mất nguồn. Bộ nhớ S7-200 có tính năng động cao, có thể đọc ghi được trong toàn vùng, ngoại trừ các bit nhớ đặc biệt SM (Special memory) chỉ có thể truy nhập để đọc. Hình vẽ 6.4 mô tả bộ nhớ trong và ngoài của PLC, bao gồm:

- Vùng chương trình: miền bộ nhớ được sử dụng để lưu trữ các lệnh chương trình.
- Vùng tham số: miền lưu trữ các tham số như từ khóa, địa chỉ trạm,... cũng giống như vùng chương trình.
- Vùng dữ liệu: được sử dụng để cất các dữ liệu của chương trình bao gồm các kết quả các phép tính, bộ đếm truyền thông...
- Vùng đối tượng: Timer, counter, bộ đếm tốc độ cao và các cổng vào/ra tương tự được đặt trong vùng nhớ cuối cùng.

Vùng nhớ dữ liệu và vùng nhớ đối tượng có ý nghĩa quan trọng trong việc thực hiện một chương trình.

ĐIỀU KHIỂN LẬP TRÌNH 1



Hình 6.4: Phân chia bộ nhớ của PLC S7-200

6.3.2 Vùng dữ liệu.

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

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

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

- V: Variable memory
- I: Input image register
- O: Output image register
- M: Internal memory bits
- SM: Special memory bits

6.3.3 Vùng đối tượng.

Vùng đối tượng được sử dụng để lưu trữ dữ liệu cho các đối tượng lập trình như các giá trị tức thời, giá trị đặt trước của bộ đếm Counter, bộ định thời Timer. Dữ liệu kiểu đối tượng bao gồm các Timer, Counter, các bộ đếm tốc độ cao, bộ đệm vào/ra tương tự và các thanh ghi Accumulator(AC).

Kiểu dữ liệu đối tượng bị hạn chế rất nhiều vì các dữ liệu kiểu đối tượng chỉ được ghi theo mục đích cần sử dụng đối tượng đó.

Phân chia vùng nhớ và các toán hạng tương ứng cho từng loại PLC cho ở bảng 6.2:

Vùng nhớ		CPU 221	CPU 222	CPU 224	CPU 226
Vùng dữ liệu	V	V0.0÷V2047.7	V0.0÷V2047.7	V0.0÷V5119.7	V0.0÷V5119.7
	I	I0.0÷I15.7	I0.0÷I15.7	I0.0÷I15.7	I0.0÷I15.7
	Q	Q0.0÷Q15.7	Q0.0÷15.7	Q0.0÷Q15.7	Q0.0÷Q15.7
	M	M0.0÷M31.7	M0.0÷M31.7	M0.0÷M31.7	M0.0÷M31.7
	SM	SM0.0÷SM179.7	SM0.0÷SM179.7	SM0.0÷SM179.7	SM0.0÷SM179.7
	S	S0.0÷S31.7	S0.0÷S31.7	S0.0÷S31.7	S0.0÷S31.7
	L	L0.0÷L63.7	L0.0÷L63.7	L0.0÷63.7	L0.0÷L63.7
Vùng	Timer	T0÷T255	T0÷T255	T0÷T255	T0÷T255

ĐIỀU KHIỂN LẬP TRÌNH 1

đối tượng	Counter	C0÷C255	C0÷C255	C0÷C255	C0÷C255
	Analog inputs	None	AIW0÷AIW30	AIW0÷AIW62	AIW0÷AIW62
	Analog outputs	None	AQW0÷AQW30	AQW0÷AQW62	AQW0÷AQW62
	Thanh nghi ACC	AC0÷AC3	AC0÷AC3	AC0÷AC3	AC0÷AC3
Bộ đếm tốc độ cao	HC0,HC3,HC4,HC5	HC0,HC3,HC4,HC5	HC0÷HC5	HC0÷HC5	

Bảng 6.2: Phân chia vùng nhớ và toán hạng PLC S7-200

6.3.4 Phương thức truy cập bộ nhớ.

- Truy cập theo bit: tên miền(+) địa chỉ byte (+) • (+) chỉ số bit.
- Truy cập theo byte: tên miền (+) B (+) địa chỉ của byte trong miền.
- Truy cập theo từ: tên miền (+) W (+) địa chỉ byte cao của từ trong miền.
- Truy cập theo từ kép: tên miền (+) D (+) địa chỉ byte cao của từ trong miền.

Access Method	CPU 221	CPU 222	CPU 224	CPU 226
Bit access (byte.bit)	V0.0÷V2047.7 I0.0÷I15.7 Q0.0÷Q15.7 M0.0÷M31.7 SM0.0÷SM179.7 S0.0÷S31.7 T0÷T255 C0÷C255 L0.0÷L63.7	V0.0÷V2047.7 I0.0÷I15.7 Q0.0÷Q15.7 M0.0÷M31.7 SM0.0÷SM179.7 S0.0÷S31.7 T0÷T255 C0÷C255 L0.0÷L63.7	V0.0÷V5119.7 I0.0÷I15.7 Q0.0÷Q15.7 M0.0÷M31.7 SM0.0÷SM179.7 S0.0÷S31.7 T0÷T255 C0÷C255 L0.0÷L63.7	V0.0÷V5119.7 I0.0÷I15.7 Q0.0÷Q15.7 M0.0÷M31.7 SM0.0÷SM179.7 S0.0÷S31.7 T0÷T255 C0÷C255 L0.0÷L63.7
Byte access	VB0÷VB2047 IB0÷IB15 QB0÷QB15 MB0÷MB31 SMB0÷SMB179 AC0÷AC3 SB0÷SB31 LB0÷LB63 Constant	VB0÷VB2047 IB0÷IB15 QB0÷QB15 MB0÷MB31 SMB0÷SMB179 AC0÷AC3 SB0÷SB31 LB0÷LB63 Constant	VB0÷VB5119 IB0÷IB15 QB0÷QB15 MB0÷MB31 SMB0÷SMB179 AC0÷AC3 SB0÷SB31 LB0÷LB63 Constant	VB0÷VB5119 IB0÷IB15 QB0÷QB15 MB0÷MB31 SMB0÷SMB179 AC0÷AC3 SB0÷SB31 LB0÷LB63 Constant
Word access	VW0÷VW2046 T0÷T255 C0÷C255 IW0÷IW14 QW0÷QW14 MW0÷MW30 SMW0÷SMW178 AC0÷AC3 LW0÷LW62 SW0÷SW30 Constant	VW0÷VW2046 T0÷T255 C0÷C255 IW0÷IW14 QW0÷QW14 MW0÷MW30 SMW0÷SMW178 AC0÷AC3 AIW0÷AIW30 AQW0÷AQW30 LW0÷LW62 SW0÷SW30 Constant	VW0÷VW5118 T0÷T255 C0÷C255 IW0÷IW14 QW0÷QW14 MW0÷MW30 SMW0÷SMW178 AC0÷AC3 AIW0÷AIW62 AQW0÷AQW62 LW0÷LW62 SW0÷SW30 Constant	VW0÷VW5118 T0÷T255 C0÷C255 IW0÷IW14 QW0÷QW14 MW0÷MW30 SMW0÷SMW178 AC0÷AC3 AIW0÷AIW62 AQW0÷AQW62 LW0÷LW62 SW0÷SW30 Constant
Double word access	VD0÷VD2044 ID0÷ID12 QD0÷QD12	VD0÷VD2044 ID0÷ID12 QD0÷QD12	VD0÷VD5116 ID0÷ID12 QD0÷QD12	VD0÷VD5116 ID0÷ID12 QD0÷QD12

ĐIỀU KHIỂN LẬP TRÌNH 1

MD0÷MD28 SMD0÷SMD176 AC0÷AC3 HC0,3,4,5 SD0÷SD28 LD0÷LD60 Constant	MD0÷MD28 SMD0÷SMD176 AC0÷AC3 HC0,3,4,5 SD0÷SD28 LD0÷LD60 Constant	MD0÷MD28 SMD0÷SMD176 AC0÷AC3 HC0÷HC5 SD0÷SD28 LD0÷LD60 Constant	MD0÷MD28 SMD0÷SMD176 AC0÷AC3 HC0÷HC5 SD0÷SD28 LD0÷LD60 Constant
---	---	---	---

Bảng 6.3: Phương pháp truy cập vùng nhớ PLC S7-200

6.4 PHƯƠNG PHÁP LẬP TRÌNH.

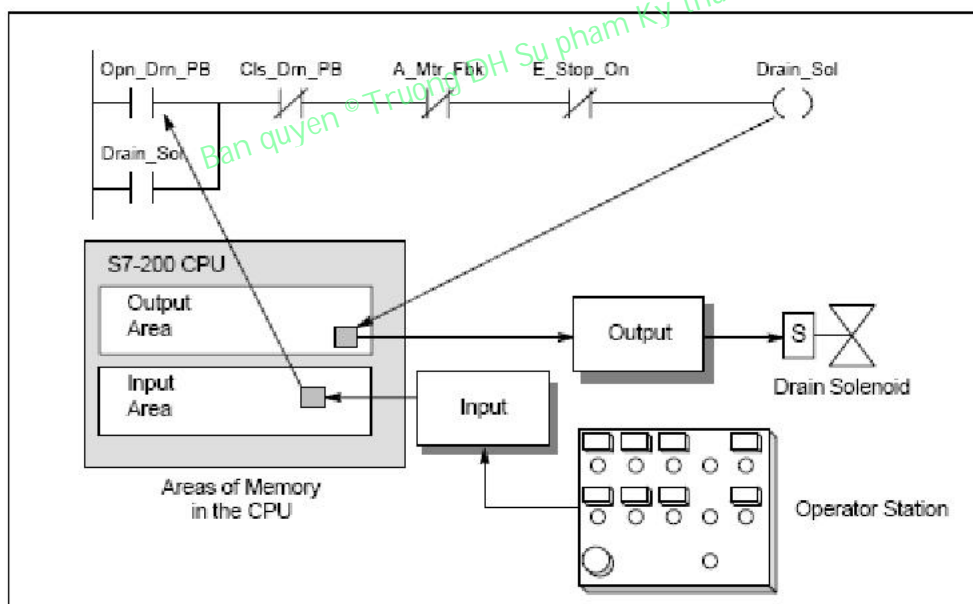
6.4.1 Quan hệ giữa chương trình và các Ngõ vào/ra.

Hoạt động cơ bản của PLC bao gồm các bước:

- CPU đọc trạng thái các ngõ vào.
- Thực hiện chương trình logic chứa trong bộ nhớ.
- CPU xuất dữ liệu đến ngõ ra.

Chương trình của PLC bao gồm một dãy các tập lệnh. PLC S7-200 thực hiện chương trình bắt đầu từ lệnh lập trình đầu tiên và kết thúc ở lệnh lập trình cuối trong một vòng quét.

Quan hệ giữa chương trình PLC và các ngõ vào, ngõ ra như hình 6.5



Hình 6.5: Quan hệ giữa chương trình và các ngõ vào/ra PLC

6.4.2 Khái niệm về ngôn ngữ lập trình.

Các CPU S7-200 có nhiều loại lệnh khác nhau cho phép ta giải quyết nhiều công việc tự động hóa. Có 2 tập lệnh cơ bản trong CPU S7-200: SIMATIC và IEC1131-3. Phần mềm lập trình cho PLC trên máy tính cũng cho phép ta lập trình bằng các loại lệnh này.

Có 2 vấn đề ta cần quan tâm khi viết chương trình cho PLC:

- Chọn loại tập lệnh nào: SIMATIC hay IEC1131-3
- Chọn ngôn ngữ lập trình nào: LAD, STL hay FBD

Quan hệ giữa tập lệnh và ngôn ngữ lập trình cho ở bảng 6.4.

ĐIỀU KHIỂN LẬP TRÌNH 1

SIMATIC Instruction Set	IEC 1131-3 Instruction Set
Statement List (STL) Editor	STL not available
Ladder Logic (LAD) Editor	Ladder Logic (LAD) Editor
Function Block Diagram (FBD) Editor	Function Block Diagram (FBD) Editor

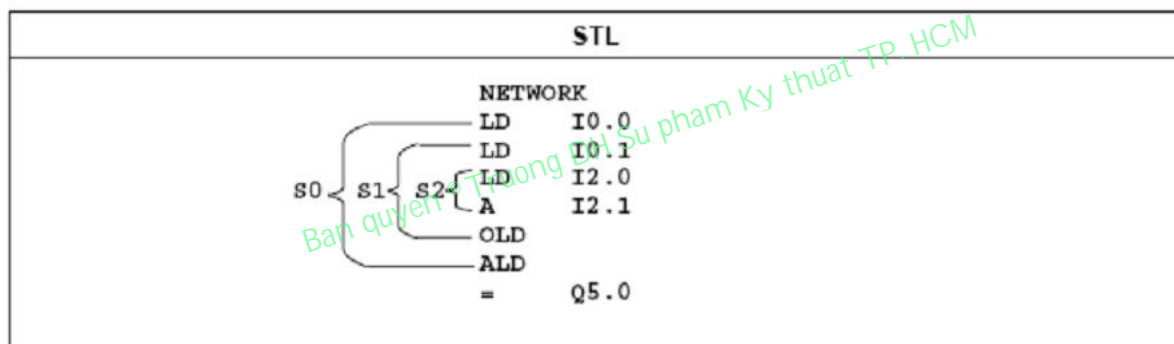
Bảng 6.4: Quan hệ giữa Tập lệnh và Ngôn ngữ lập trình

6.4.3 Ngôn ngữ lập trình STL (Statement List)

Soạn thảo chương trình theo phương pháp STL cho phép ta viết chương trình điều khiển bằng các lệnh gọi nhớ. Nói chung soạn thảo bằng STL phù hợp cho người có kinh nghiệm lập trình và đã quen với PLC cũng như cách lập trình logic.

Soạn thảo bằng ngôn ngữ STL cũng cho phép ta tạo ra các chương trình mà các ngôn ngữ LAD và FBD không thực hiện được. Vì STL là cách lập trình theo ngôn ngữ tự nhiên của CPU, trong khi các phương pháp khác là lập trình đồ họa.

Một chương trình ví dụ theo ngôn ngữ STL như sau:



Chương trình này tương tự như lập trình bằng ngôn ngữ Assembler. CPU thực hiện chương trình bằng cách chạy các lệnh từ trên xuống dưới, rồi lặp lại.

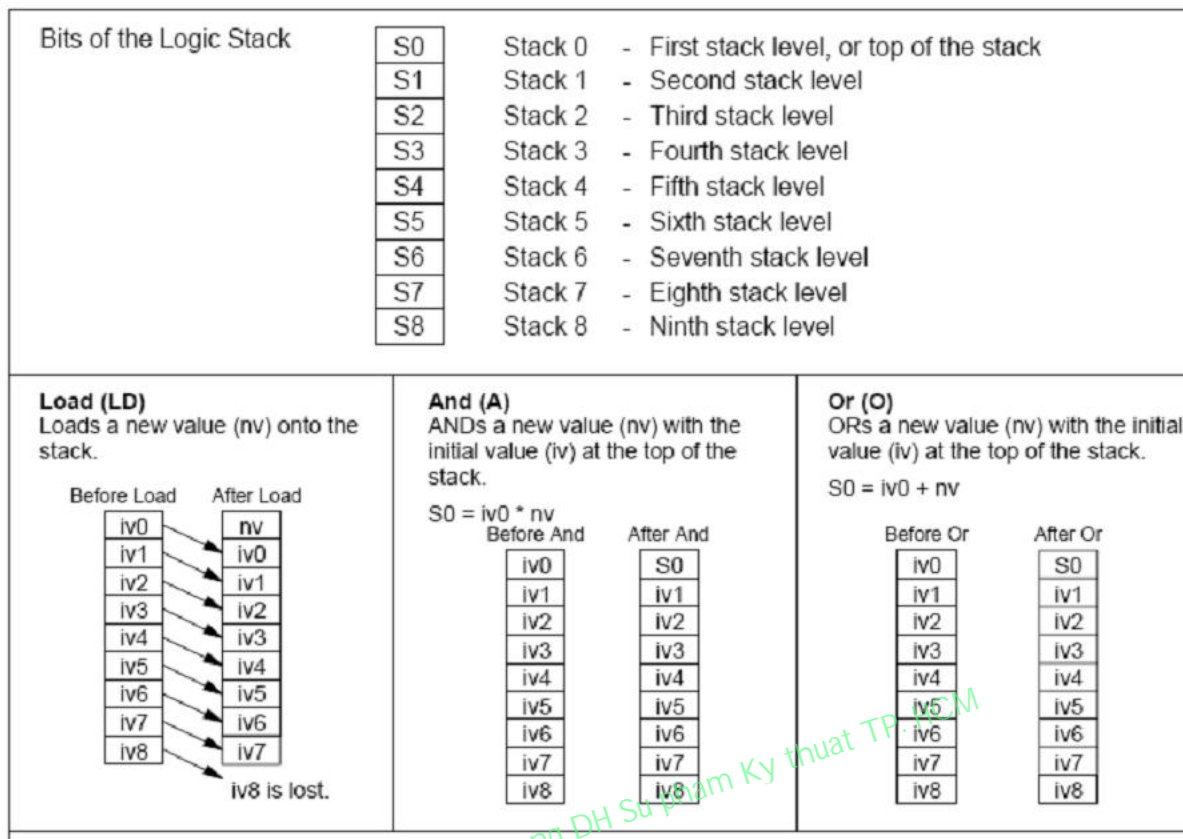
PLC S7-200 sử dụng ngăn xếp logic để giải quyết các lệnh logic, xem hình 6.6.

Ngôn ngữ LAD và FBD tự động thêm các lệnh cần thiết để quản lý hoạt động của ngăn xếp logic. Đối với ngôn ngữ STL, ta phải thêm vào các lệnh quản lý ngăn xếp.

Các điểm chính cần quan tâm khi chọn ngôn ngữ lập trình STL:

- STL thích hợp cho những người lập trình kinh nghiệm.
- STL cho phép ta giải quyết các điều khiển phức tạp mà LAD và FBD không thực hiện được.
- STL chỉ thực hiện với tập lệnh SIMATIC.
- Có thể chuyển từ chương trình STL sang LAD và FBD nhưng ngược lại thì sẽ bị giới hạn.

ĐIỀU KHIỂN LẬP TRÌNH 1



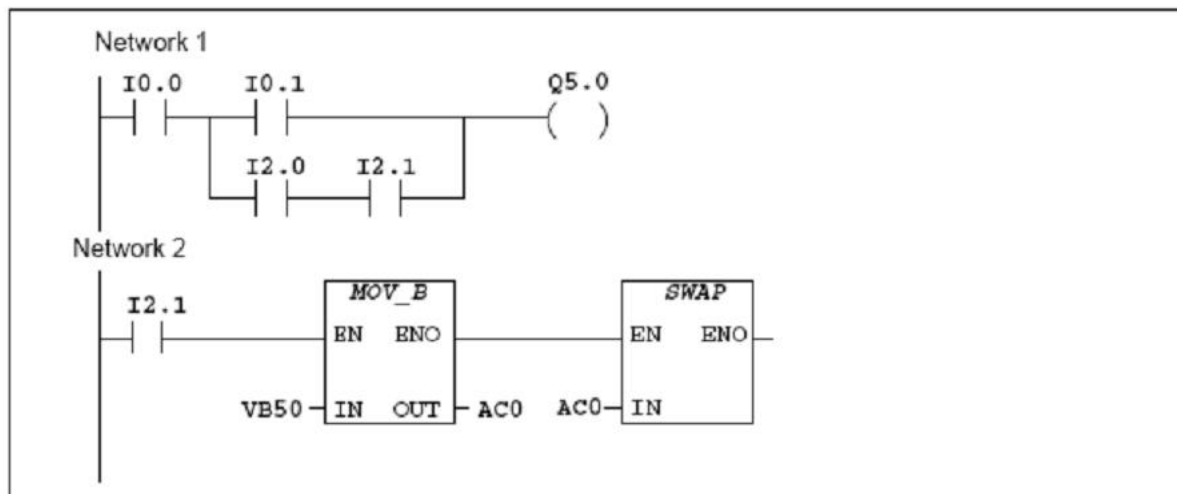
Hình 6.6: Ngăn xếp logic của PLC S7-200

6.4.4 Ngôn ngữ lập trình LAD (Ladder Loic)

Ngôn ngữ LAD cho phép ta viết chương trình tương tự như mạch tương đương của sơ đồ nối dây mạch điện. Rất nhiều người lập trình và các nhân viên kỹ thuật chọn lựa sử dụng phương pháp này.

Chương trình LAD cho phép CPU mô phỏng di chuyển của dòng điện từ nguồn, qua một loạt các điều kiện ngõ vào để tác động đến ngõ ra.

Hình 6.7 minh họa chương trình LAD.



Hình 6.7:Chương trình LAD của PLC S7-200

ĐIỀU KHIỂN LẬP TRÌNH 1

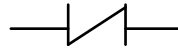
Các lệnh khác nhau được biểu diễn bằng các ký hiệu đồ họa, gồm 3 dạng cơ bản:

- Tiếp điểm: biểu diễn các điều kiện logic ngõ vào, như các công tắc, nút nhấn, trạng thái của cảm biến, ...

Tiếp điểm thường mở



Tiếp điểm thường đóng



- Cuộn dây (coil): —() biểu diễn cho kết quả logic ngõ ra, như đèn, động cơ, cuộn dây của 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 gồm các bộ thời gian (Timer), bộ đếm (counter) và các hàm toán học. Cuộn dây và các hộp phải mắc đúng chiều dòng điện.
- **Mạch LAD**: Là đường nối các phần tử thành một mạch hoàn chỉnh, đi từ đường nguồn bên trái sang đường nguồn bên phải. Đường nguồn bên trái là dây pha, đường nguồn bên phải là dây trung tính và cũng là đường trở về nguồn cung cấp. Dòng điện chạy từ trái qua tiếp điểm đến đóng các cuộn dây hoặc các hộp trở về bên phải nguồn.

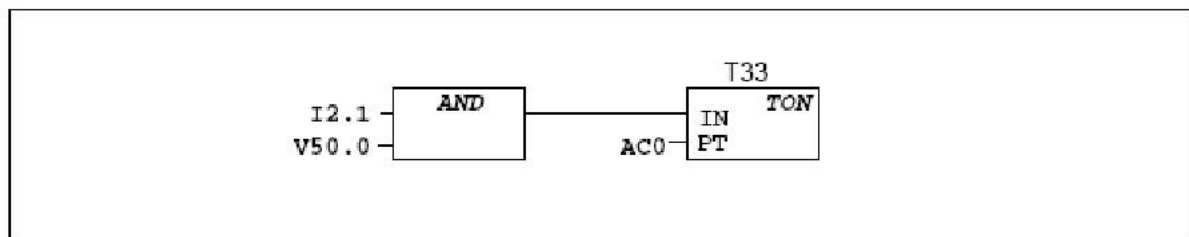
Các vấn đề chính cần quan tâm khi sử dụng ngôn ngữ LAD:

- LAD thích hợp cho người mới bắt đầu lập trình.
- Biểu diễn đồ họa dễ hiểu và thông dụng hơn.
- LAD sử dụng được các tập lệnh SIMATIC và IEC1131-3.
- Luôn chuyển từ dạng LAD sang STL.

6.4.5 Ngôn ngữ FBD (Function Block Diagram)

Ngôn ngữ FBD cho phép ta xem các lệnh như là các hộp logic, tương tự như sơ đồ cổng logic. Không có các tiếp điểm và cuộn dây, nhưng sẽ có các hộp. Chương trình logic sẽ được tạo ra bằng việc kết nối các hộp, ngõ ra lệnh này sẽ tác động đến ngõ vào lệnh kia tạo thành chương trình điều khiển logic. Phương pháp kết nối này cho phép ta giải quyết được nhiều bài toán logic khác nhau.

Ví dụ minh họa chương trình FBD trên hình 6.8.



Hình 6.8:Chương trình FBD của PLC S7-200

Các vấn đề chính cần quan tâm khi lập trình FBD.

-
- FBD sử dụng được các tập lệnh SIMATIC và IEC1131-3.
- Luôn chuyển đổi từ chương trình FBD sang STL.

ĐIỀU KHIỂN LẬP TRÌNH 1

CHƯƠNG 7: TẬP LỆNH S7 – 200

7.1 NHÓM LỆNH VỀ TIẾP ĐIỂM.

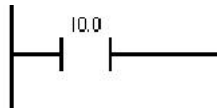
7.1.1 Lệnh Vào/Ra.

- LOAD (LD) :

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

Toán hạng gồm I, Q, M, SM, V, C, T.

- o Dạng LAD : tiếp điểm thường mở I0.0 sẽ đóng nếu ngõ vào PLC có địa chỉ I0.0 tác động.



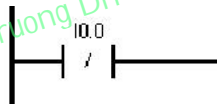
- o Dạng STL : LD I0.0

- LOAD NOT (LDN) :

Lệnh LDN nạp giá trị logic của một tiếp điểm vào trong bit đầu tiên của ngăn xếp, các giá trị còn lại trong ngăn xếp bị đẩy lùi xuống một bit.

Toán hạng gồm : I, Q, M, SM, V, C, T.

- o Dạng LAD : tiếp điểm thường đóng sẽ mở khi ngõ vào PLC có địa chỉ I0.0 tác động.



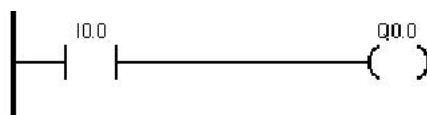
- o Dạng STL : LDN I0.0

- OUTPUT (=) :

Lệnh sao chép nội dung của bit đầu tiên trong ngăn xếp vào bit được chỉ định trong lệnh. Nội dung ngăn xếp không bị thay đổi.

Toán hạng bao gồm : I,Q,M,SM,T,C (bit)

- o Mô tả lệnh OUTPUT bằng LAD như sau :
Nếu tiếp điểm I0.0 đóng thì cuộn dây Q0.0 sẽ được cấp điện (làm cho ngõ ra của PLC có địa chỉ Q0.0=1)



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

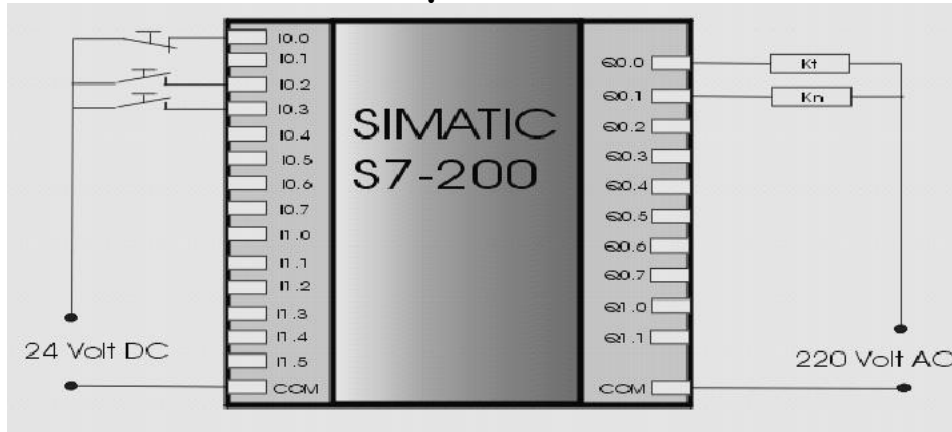
LD I0.0
= Q0.0

ĐIỀU KHIỂN LẬP TRÌNH 1

VÍ DỤ:

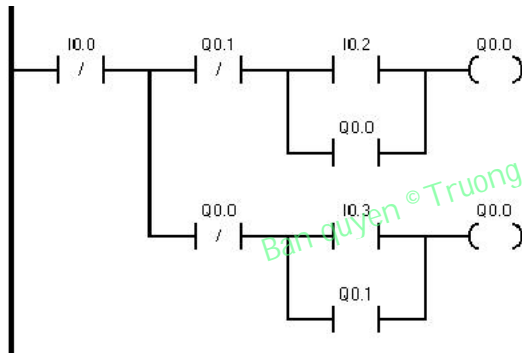
Lập trình PLC điều khiển đảo chiều động cơ.

- Sơ đồ kết nối PLC với các thiết bị :



- Lập trình dạng LAD :

Chú thích :



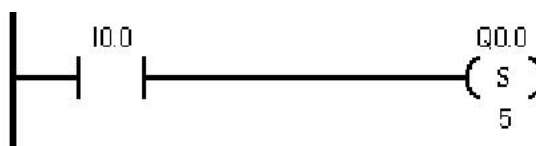
- IO.0** : nút nhấn dừng
- IO.2** : nút nhấn mở máy thuận
- IO.3** : nút nhấn mở máy nghịch
- Q0.0** : ngõ ra nối với cuộn dây Kt.
- Q0.1** : ngõ ra nối với cuộn dây Kn.

7.1.2 Lệnh Ghi/ Xóa.

- SET (S) :

Lệnh dùng để đóng các điểm liên tục đã được chọn trước. Trong LAD, logic ngõ ra sẽ điều khiển đóng dòng điện 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 sẽ đóng 1 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.

- o Dạng LAD : khi tiếp điểm IO.0 đóng lệnh SET sẽ đóng một mảng gồm n các tiếp điểm kể từ địa chỉ Q0.0.
Toán hạng bao gồm I, Q, M, SM,T, C,V (bit)



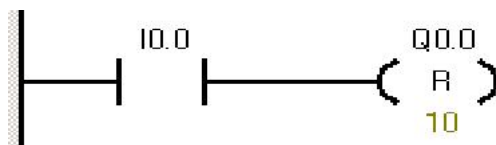
- o Dạng STL : ghi giá trị logic vào một mảng gồm n bit từ địa chỉ Q0.0
LD IO.0
S Q0.0, 5

ĐIỀU KHIỂN LẬP TRÌNH 1

- RESET (R):

Lệnh dùng để ngắt các điểm liên tục được chọn trước. Trong LAD, logic ngõ ra sẽ điều khiển ngắt dòng điện 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 1 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.

- o Dạng LAD : ngắt một mảng gồm 10 các tiếp điểm kể từ tiếp điểm có địa chỉ Q0.0. Toán hạng bao gồm I, Q, M, SM,T, C,V (bit)



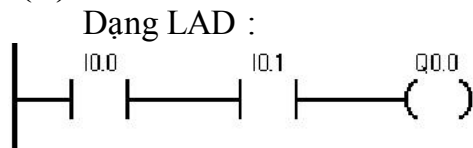
- o Dạng STL : xóa một mảng gồm 10 bit kể từ địa chỉ Q0.0.

```
LD I0.0
R Q0.0, 10
```

7.1.3 Lệnh Đại số Bool.

Các lệnh tiếp điểm đại số Bool cho phép tạo lập các mạch logic (không có nhớ). Trong LAD các lệnh này được biểu diễn thông qua cấu trúc mạch, mắc nối tiếp hay song song các tiếp điểm thường đóng hay các tiếp điểm thường mở.

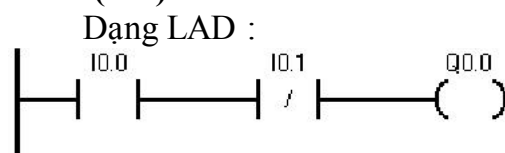
- AND (A) :



Dạng STL :

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

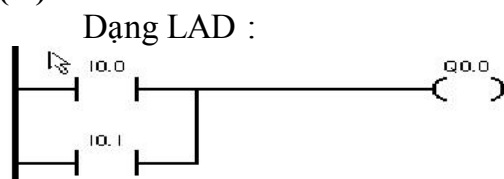
- AND NOT(AN) :



Dạng STL :

```
LD I0.0
AN I0.1
= Q0.0
```

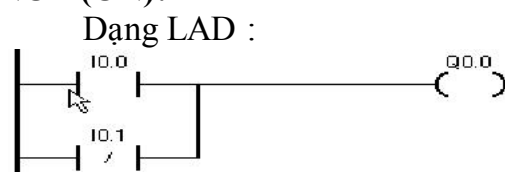
- OR (O):



Dạng STL :

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

- OR NOT (ON):



Dạng STL :

```
LD I0.0
ON I0.1
= Q0.0
```

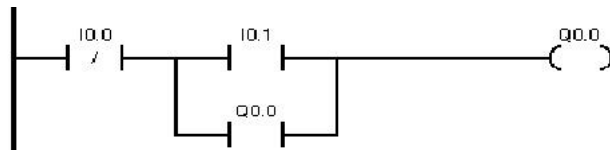
ĐIỀU KHIỂN LẬP TRÌNH 1

Ví dụ :

Viết chương trình điều khiển mở máy động cơ bằng PLC.

Lập trình LAD

Ghi chú :



- I0.0 : Nút nhấn dừng
- I0.1 : Nút nhấn mở
- Q0.0 : Cuộn dây KĐT
- Q0.0 : Tiếp điểm duy trì

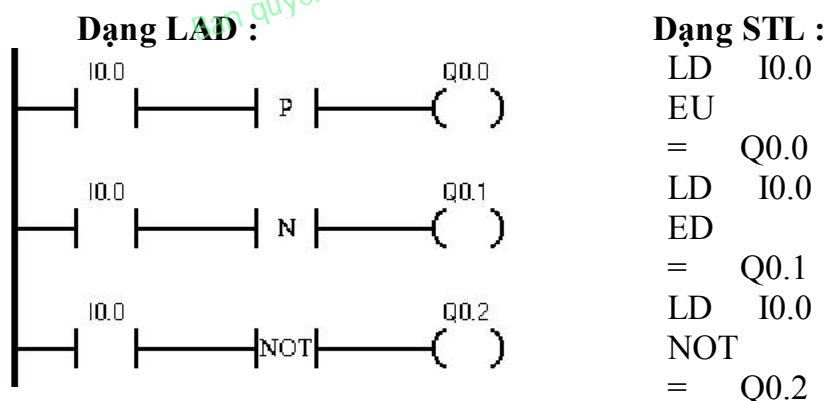
7.1.4 Lệnh Tiếp điểm đặc biệt.

- Tiếp điểm đảo, tác động cạnh xuống, tác động cạnh lên :

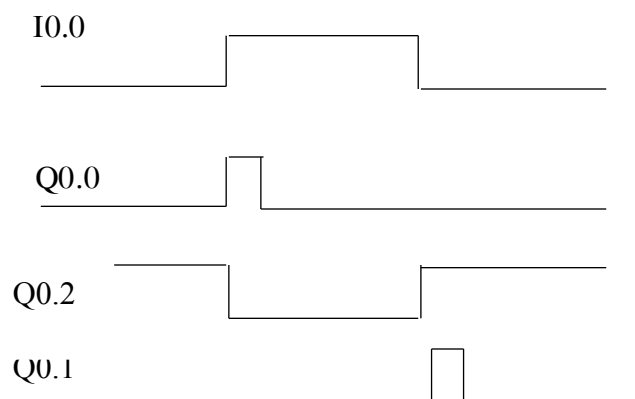


Có thể dùng các lệnh tiếp điểm đặc biệt để phát hiện sự chuyển tiếp trạng thái của xung (sườn xung) và đảo lại trạng thái của dòng cung cấp (giá trị đỉnh của ngõ xấp). LAD sử dụng các tiếp điểm đặc biệt này để tác động vào dòng cung cấp. Các tiếp điểm đặc biệt không có toán hạng riêng của nên ta phải đặt chúng phía trước cuộn dây hoặc hộp đầu ra. Tiếp điểm chuyên tiếp dương/âm (các lệnh sườn trước và sườn sau) có yêu cầu về bộ nhớ, bởi vậy đối với CPU 214 có thể sử dụng nhiều nhất là 256 lệnh.

Ví dụ minh họa:



Biểu đồ thời gian :



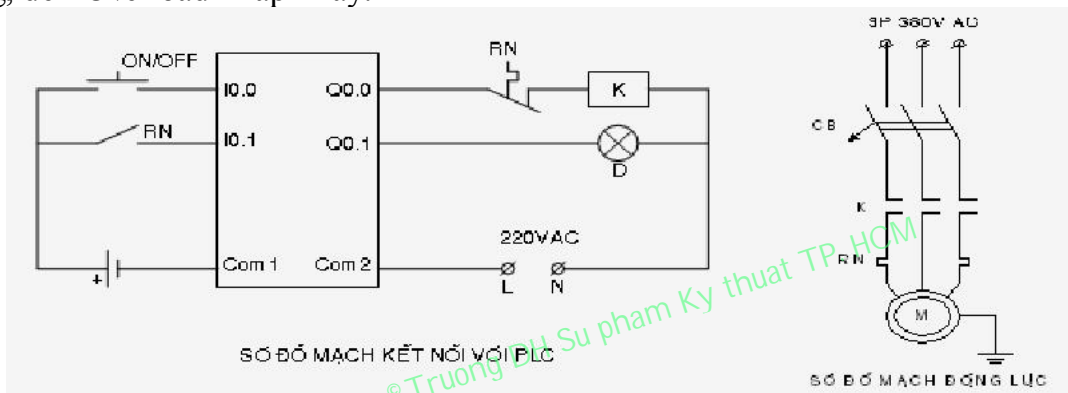
ĐIỀU KHIỂN LẬP TRÌNH 1

- Tiếp điểm trong vùng nhớ đặc biệt :

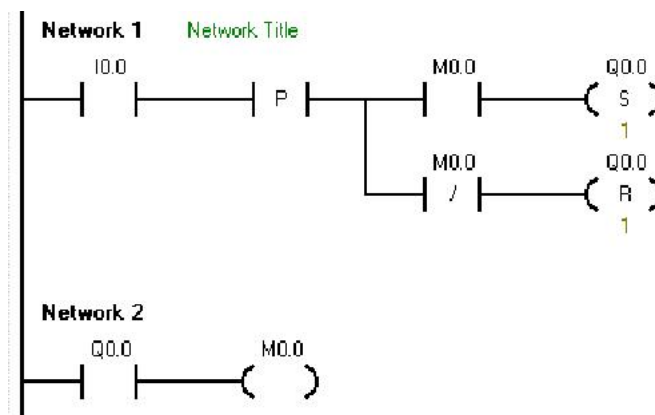
- **SM0.0** : Ngược lại với SM0.1, vòng quét đầu tiên thì mở nhưng từ vòng quét thứ hai trở đi thì đóng.
- **SM0.1** : Vòng quét đầu tiên tiếp điểm này đóng, kể từ vòng quét thứ hai thì mở ra và giữ nguyên trong suốt quá trình hoạt động.
- **SM0.4** : Tiếp điểm tạo xung với nhịp xung với chu kỳ là 1 phút.
- **SM0.5** : Tiếp điểm tạo xung với nhịp xung với chu kỳ là 1s

VÍ DỤ:

Viết chương trình điều khiển động cơ bằng 1 nút nhấn ON/OFF. Nhấn lần 1 động cơ hoạt động, nhấn lần 2 động cơ dừng. Khi bị quá tải relay nhiệt thì động cơ tự động dừng, đèn Overload nhấp nháy.



Chương trình tham khảo:



7.2 NHÓM LỆNH VỀ TIMER VÀ COUNTER.

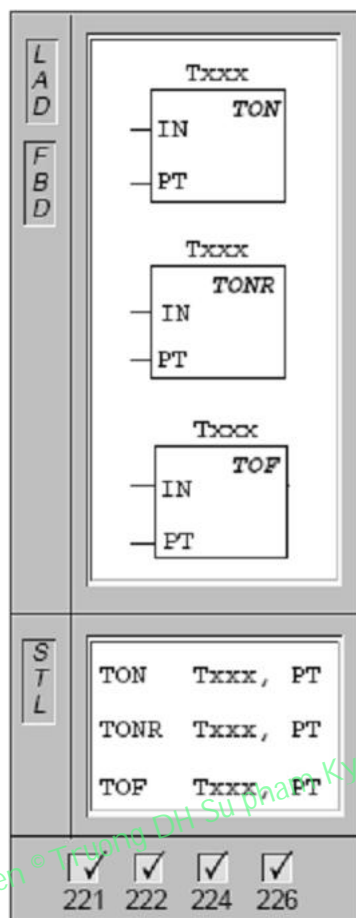
7.2.1 Lệnh Timer.

Timer là bộ tạo thời gian trễ giữa tín hiệu vào và tín hiệu ra nên trong điều khiển thường được gọi là khâu trễ. Nếu ký hiệu tín hiệu (logic) vào là $x(t)$ và thời gian trễ tạo ra bằng Timer là τ thì tín hiệu đầu ra của Timer đó sẽ là $x(t - \tau)$

S7-200 có 64 bộ Timer (với CPU 212) hoặc 128 Timer (với CPU 214) được chia làm 3 loại khác nhau:

- Timer đóng chậm không có nhớ (On-Delay Timer), ký hiệu là TON.
- Timer mở chậm không nhớ (Off-Delay Timer), ký hiệu TOF.
- Timer đóng chậm có nhớ (Retentive On-Delay Timer), ký hiệu TONR.

ĐIỀU KHIỂN LẬP TRÌNH 1



Ba loại Timer của S7-200 phân biệt với nhau ở phản ứng của nó đối với trạng thái logic đầu vào.

Cả hai Timer kiểu TON và TONR cùng bắt đầu đếm thời gian kể từ thời điểm có sườn lên ở tín hiệu đầu vào, đầu vào chuyển trạng thái logic từ 0 lên 1, được gọi là thời gian Timer được kích. Khi giá trị thời gian của Timer lớn hơn hoặc bằng giá trị đặt trước thì tiếp điểm của Timer sẽ đóng lại. Lúc này Timer vẫn tiếp tục đếm thời gian cho đến giá trị max.

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 TOF dùng để mở chậm một ngõ ra sau thời gian đặt trước nào đó, kể từ khi logic ngõ vào chuyển từ 1 xuống 0. Khi ngõ vào Timer lên 1, tiếp điểm của nó đóng ngay lập tức và đặt giá trị thời gian bằng 0. Khi ngõ vào xuống 0, Timer sẽ đếm thời gian cho đến giá trị đặt trước thì tiếp điểm Timer sẽ mở ra.

Inputs/Outputs	Operands	Data Types
Txxx	Constant	WORD
IN (LAD)	Power Flow	BOOL
IN (FBD)	I, Q, M, SM, T, C, V, S, L, Power Flow	BOOL
PT	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, Constant, *VD, *AC, *LD	INT

ĐIỀU KHIỂN LẬP TRÌNH 1

Timer TON, TOF và TONR đều có 3 loại với 3 độ phân giải khác nhau, độ phân giải 1ms, 10ms và 100ms. Thời gian trễ τ được tạo ra chính là tích của độ phân giải của bộ Timer được chọn và giá trị đặt trước cho Timer. Ví dụ Timer có độ phân giải 10ms và giá trị đặt trước 50 thì thời gian trễ là 500ms.

Độ phân giải các loại Timer của S7-200, loại CPU 214, được trình bày trong bảng bên dưới.

Lệnh	Độ phân giải	Giá trị cực đại	CPU 214
TON, TOF	1 ms	32,767 s	T32 và T96
	10 ms	327,67 s	T33 ÷ T36, T97 ÷ T100
	100 ms	3276,7 s	T37 ÷ T63, T101 ÷ T127
TONR	1 ms	32,767 s	T0 và T64
	10 ms	327,67 s	T1 ÷ T4, T65 ÷ T68
	100 ms	3276,7 s	T5 ÷ T31, T69 ÷ T95

Cú pháp khai báo sử dụng Timer như sau :

LAD	Mô tả	Toán hạng
TON-Txx IN PT	Khai báo Timer số hiệu xx 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.	Txx (word) CPU214:32÷63 96÷127 PT: VW, T, (word) C, IW, QW, MW, SMW, C, IW, hằng số
TONR-Txx IN PT	Khai báo Timer số hiệu xx 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 Timer kiểu TON bằng lệnh R cho T-bit.	Txx (word) CPU214:0÷31 64 ÷95 PT: VW, T, (word) C, IW, QW, MW, SMW, AC, AIW, hằng số

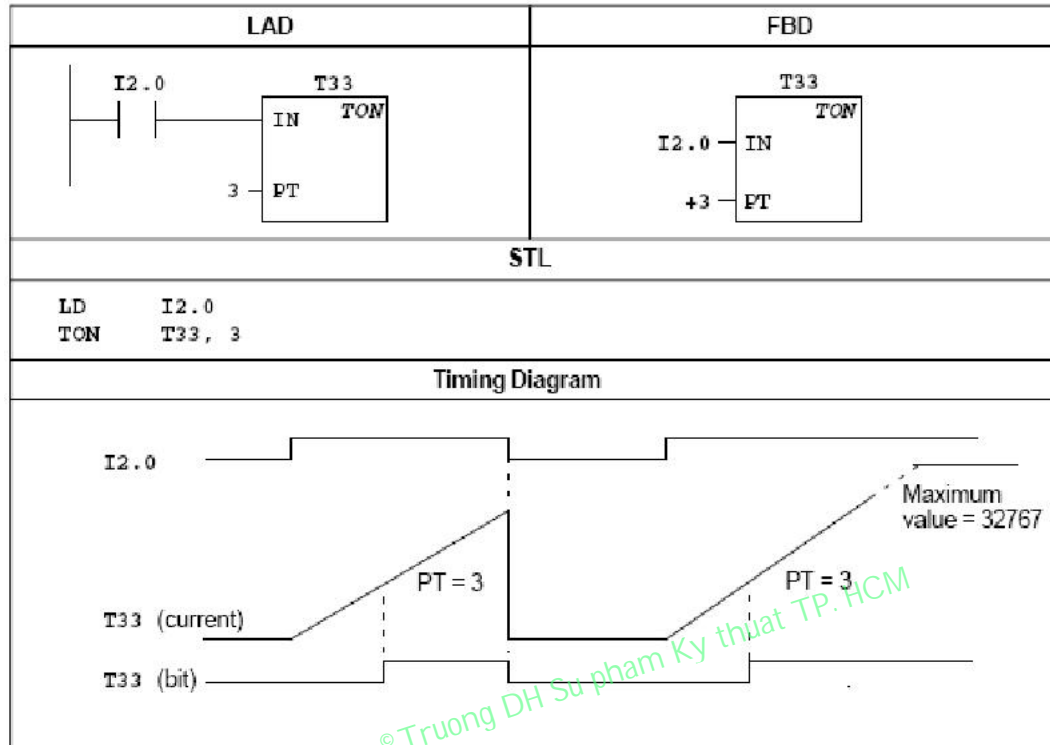
Khi sử dụng Timer TONR, giá trị đếm tức thời được lưu lại và không bị thay đổi trong khoảng thời gian khi tín hiệu đầu vào có logic 0. Giá trị của T-bit không được nhớ mà hoàn toàn phụ thuộc vào kết quả so sánh giữa giá trị đếm tức thời và giá trị đặt trước.

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 0.

ĐIỀU KHIỂN LẬP TRÌNH 1

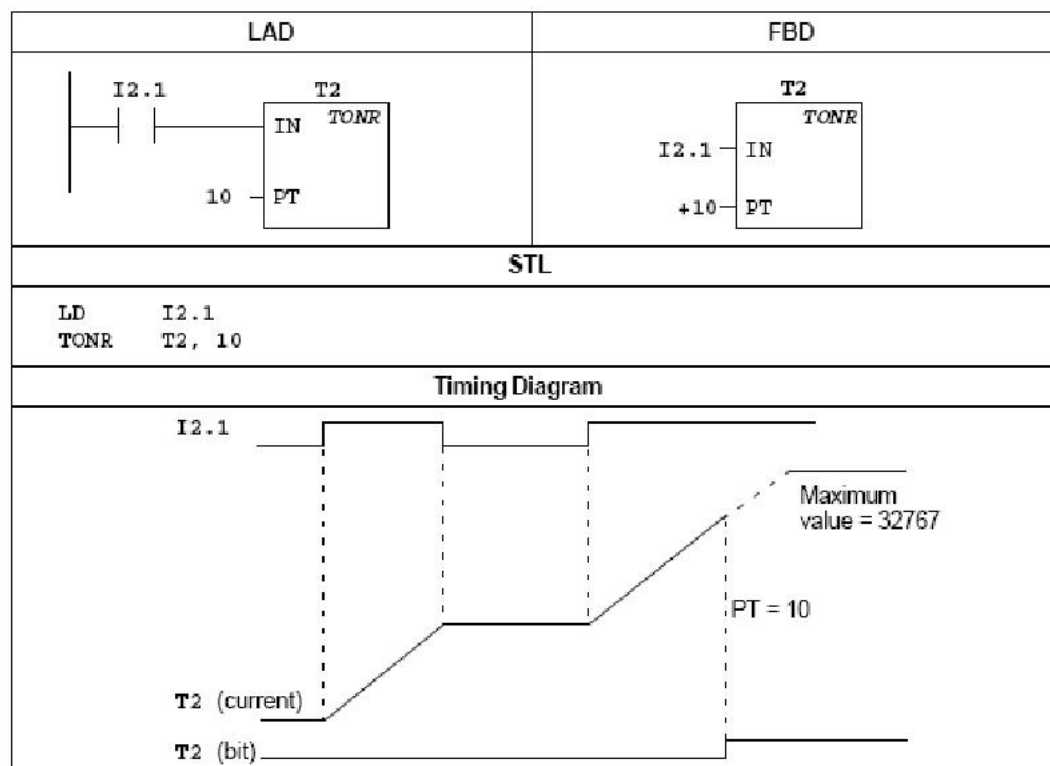
Ví dụ:

Sử dụng Timer kiểu TON



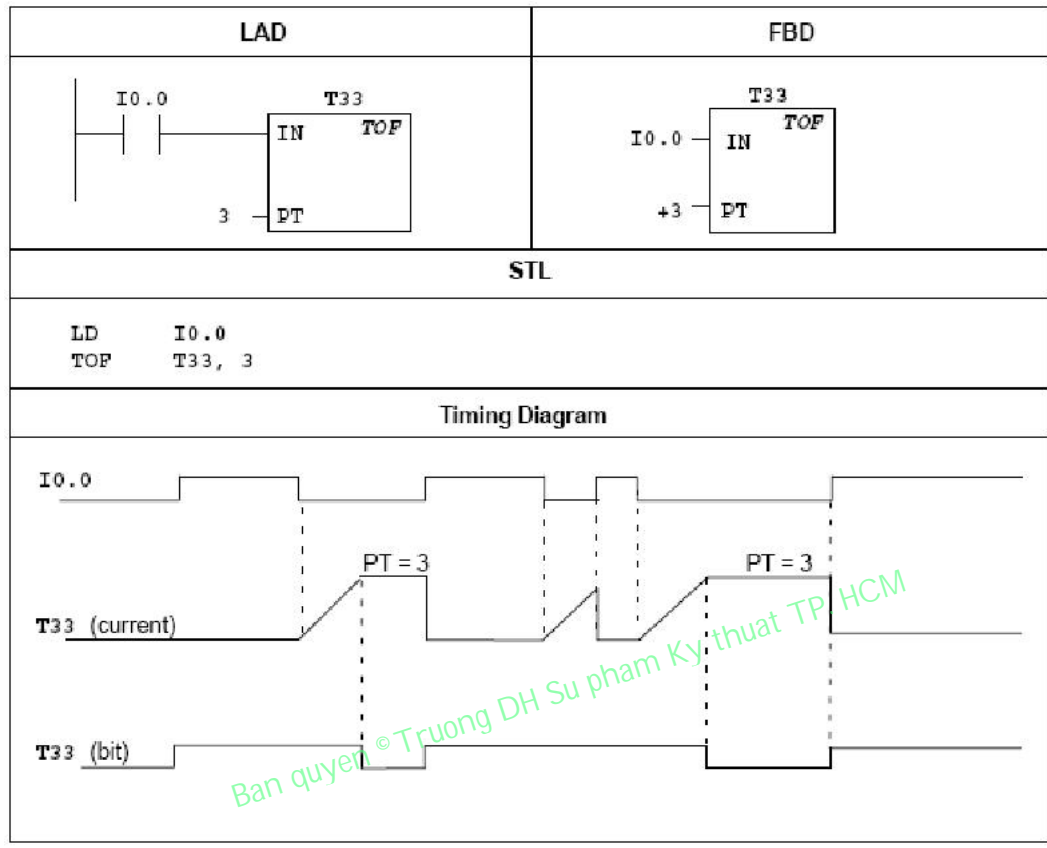
Ví dụ:

Sử dụng timer kiểu TONR



ĐIỀU KHIỂN LẬP TRÌNH 1

**Ví dụ:
Sử dụng timer kiểu TOF**

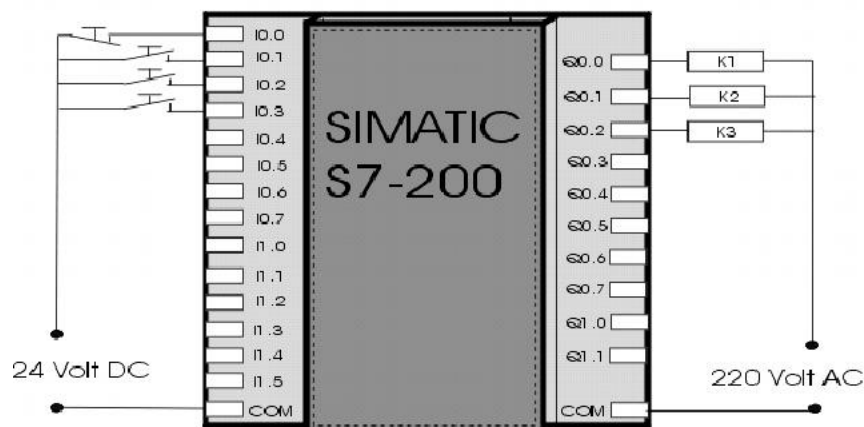


VÍ DỤ MINH HỌA:

Lập trình logic cho PLC điều khiển dây chuyền sản xuất gồm 3 động cơ hoạt động theo yêu cầu sau :

- Nhấn nút khởi động cho động cơ Đ1 chạy, sau 5s cho phép vận hành Đ2.
- Nhấn nút khởi động cho động cơ Đ2 chạy đồng thời lúc này động cơ Đ1 ngừng, sau 10s thì cho phép vận hành động cơ Đ3.
- Nhấn nút khởi động cho động cơ Đ3 chạy đồng thời động cơ Đ2 ngừng.
- Nhấn nút dừng thì bất kỳ động cơ nào chạy cũng phải ngừng.

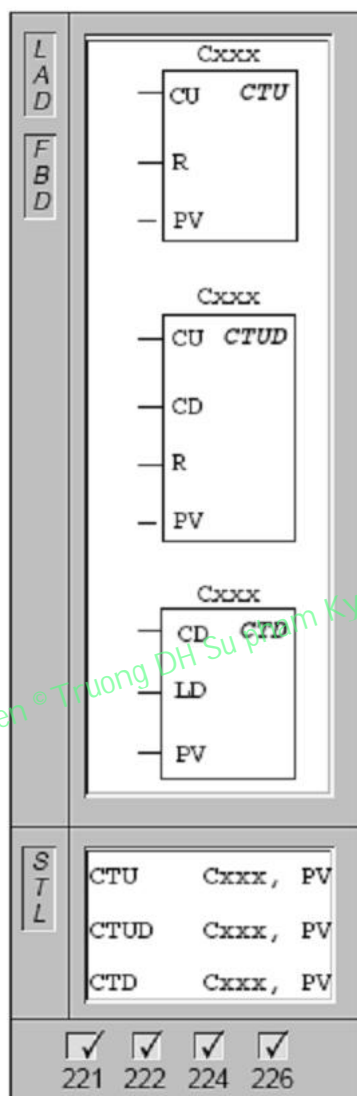
Sơ đồ kết nối PLC :



ĐIỀU KHIỂN LẬP TRÌNH 1

7.2.2 Lệnh Counter.

Counter là bộ đếm thực hiện chức năng đếm sườn xung, trong S7-200 các bộ đếm được chia làm 3 loại : bộ đếm lên (CTU), bộ đếm xuống (CU) và bộ đếm lên/xuống (CTUD).



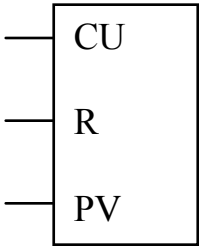
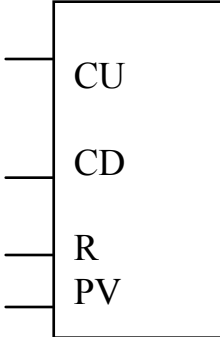
- Bộ đếm lên CTU đếm số sườn lên của tín hiệu logic đầu vào, tức là đếm số lần thay đổi trạng thái logic từ 0 lên 1 của tín hiệu. Số xung đếm được ghi vào thanh ghi 2 byte của bộ đếm, gọi là thanh ghi C-word. Nội dung của thanh ghi C-word, gọi là giá trị đếm tức thời của bộ đếm, luôn được so sánh với 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.
- Bộ đếm lên/xuống CTUD đếm lên khi gặp sườn lên của xung vào cổng đếm lên, ký hiệu là CU hoặc bit thứ 3 của ngăn xếp trong STL, và đếm xuống khi gặp sườn lên của xung vào cổng đếm xuống, ký hiệu là CD trong LAD hoặc bit thứ 2 của ngăn xếp trong STL.
 Khác với các bộ Timer, các bộ đếm CTU và CTUD đều có chân nối với tín hiệu điều khiển xóa để thực hiện việc đặt lại chế độ ban đầu (reset) cho bộ đếm, được

ĐIỀU KHIỂN LẬP TRÌNH 1

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 1 hoặc khi lệnh R (reset) được thực hiện với C-bit. Khi bộ đếm được reset, cả C-word và C-bit đều có giá trị 0.

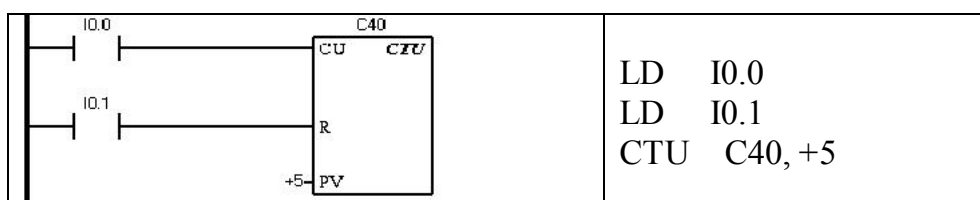
Bộ đếm lên CTU có miền giá trị đếm tức thời từ 0 đến 32.767. Bộ đếm lên/xuống CTUD có miền giá trị đếm tức thời từ -32.768 đến 32.767.

- Bộ đếm xuống CU đếm xuống từ giá trị đặt trước khi có sườn lên tác động vào cổng đếm xuống. Khi giá trị đếm bằng 0 thì dừng đếm và tiếp điểm của nó sẽ đóng. Khi chân Load LD tác động thì bộ đếm xuống sẽ mở tiếp điểm và nạp giá trị đặt trước vào.

LAD	Mô tả	Toán hạng
<p>CTU – Cxx</p> 	<p>Khai báo bộ đếm tiến theo sườn lên của CU. Khi giá trị đếm tức thời C-word Cxx lớn hơn hoặc bằng giá trị đặt trước PV, C-bit (Cxx) 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 được giá trị cực đại.</p>	<p>Cxx : (word) CPU 214 : 0 ÷ 47 80 ÷ 127 PV(word) : VW, T, C, IW, QW, MW, SMW, AC, AIW, hằng số, *VD, *AC</p>
<p>CTD-Cxx</p> 	<p>Khai báo bộ đếm tiến/lùi, đếm tiến theo sườn lên của CU, đếm lùi theo sườn lên của CD. Khi giá trị đếm tức thời C-word Cxx lớn hơn hoặc bằng giá trị đặt trước PV, C-bit (Cxx) có giá trị logic bằng 1. Bộ đếm ngừng đếm tiến khi C-word Cxx đạt được giá trị cực đại 32.767 và ngừng đếm lùi khi C-word Cxx đạt được giá trị cực đại - 32.768. CTUD reset khi đầu vào R có giá trị logic bằng 1.</p>	<p>Cxx (word) CPU 214 : 48 ÷ 79 PV(word) : VW, T, C, IW, QW, MW, SMW, AC, AIW, hằng số, *VD, *AC</p>

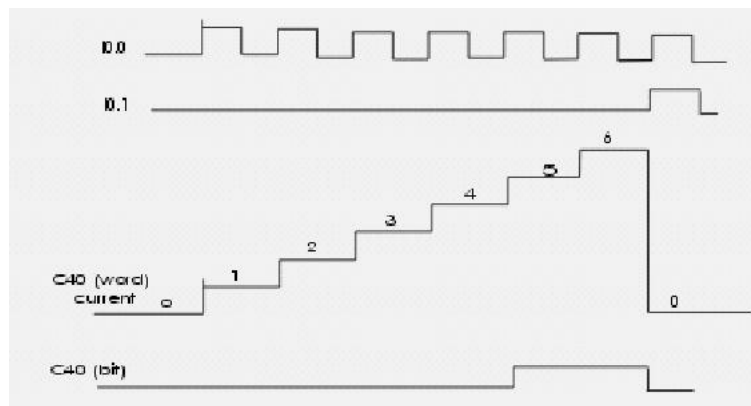
Ví dụ:

- o **Sử dụng bộ đếm CTU :**

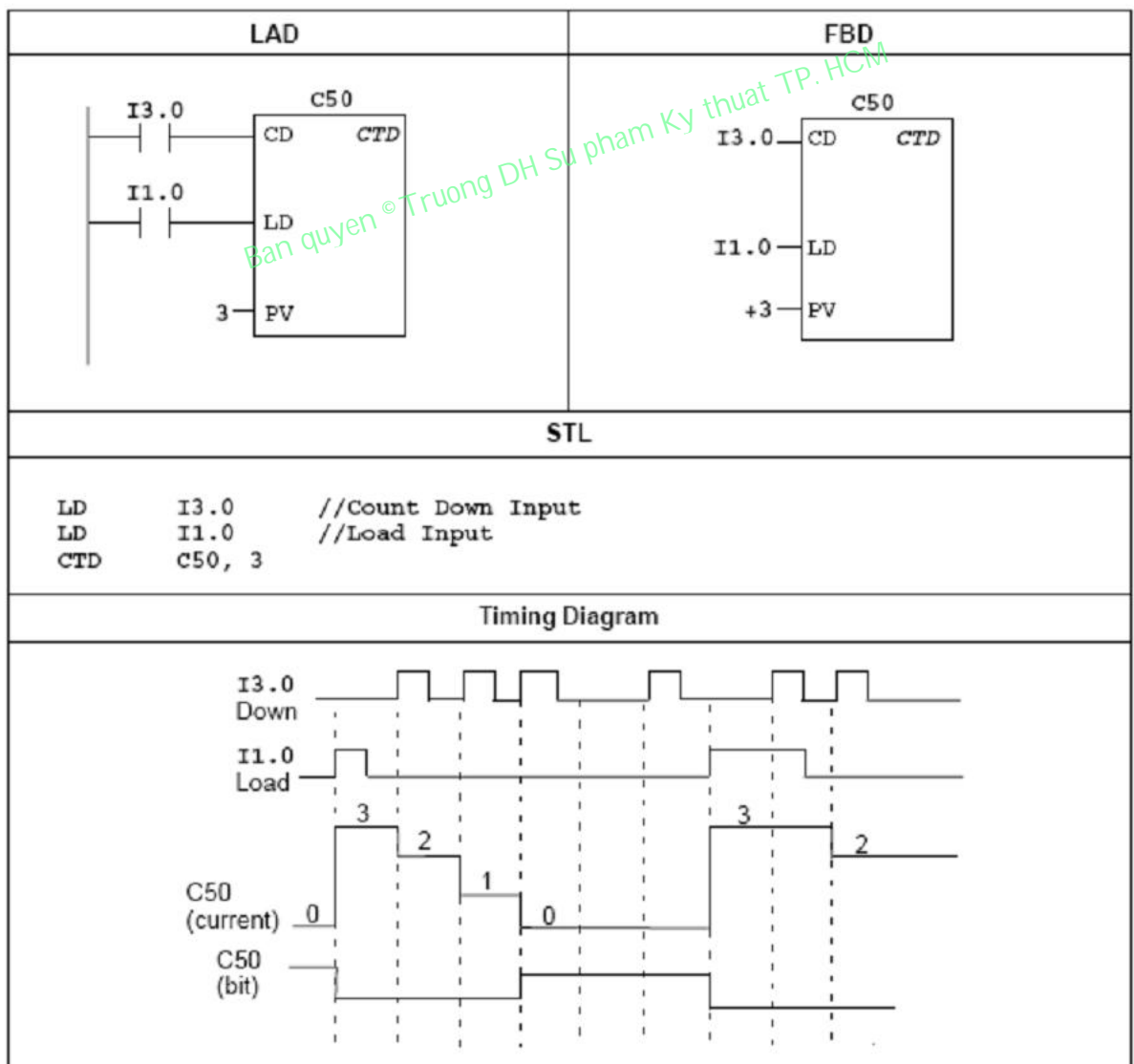


ĐIỀU KHIỂN LẬP TRÌNH 1

Giải đồ thời gian :

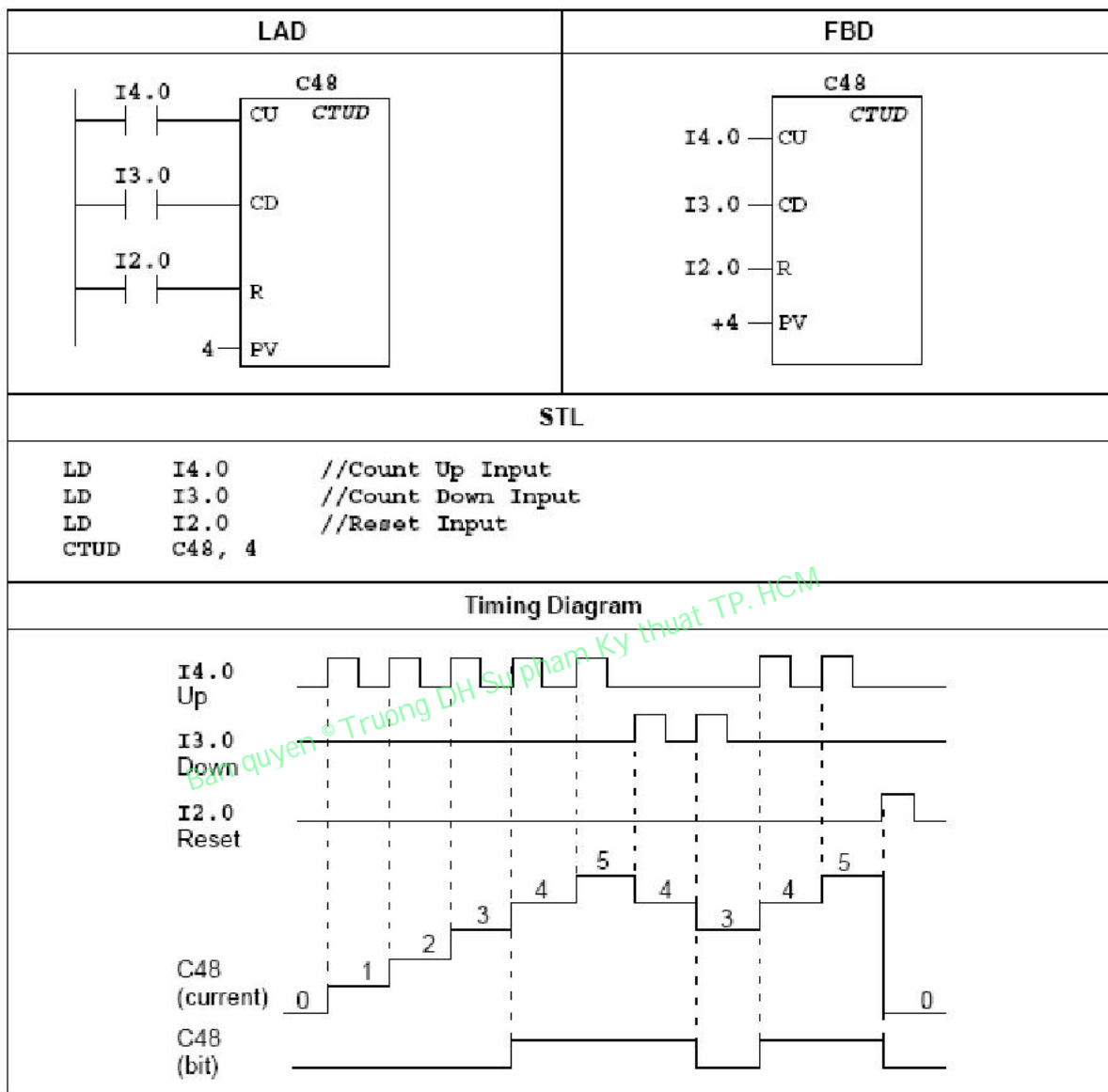


○ **Sử dụng bộ đếm CTD:**



ĐIỀU KHIỂN LẬP TRÌNH 1

- Sử dụng bộ đếm CTUD :



7.3 NHÓM LỆNH SO SÁNH.

Khi lập trình, nếu 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 theo Byte, Word hay DWord của S7-200. Những lệnh so sánh thường là: so sánh nhỏ hơn hoặc bằng (<=); so sánh bằng (=) và so sánh lớn hơn hoặc bằng (>=).

Khi so sánh giá trị của byte thì không cần phải để ý đến dấu của toán hạng, ngược lại khi so sánh các từ hay từ kép với nhau thì phải để ý đến dấu của toán hạng là bit cao nhất trong từ hoặc từ kép.

Trong STL những lệnh so sánh thực hiện phép so sánh byte, Word hay DWord. Căn cứ vào kiểu so sánh (<=, =, >=), kết quả của phép so sánh có giá trị bằng 0 (nếu đúng) hoặc bằng 1 (nếu sai) nên nó có thể được kết hợp cùng các lệnh LD, A, O. Để tạo ra được các phép so sánh mà S7-200 không có lệnh so sánh tương ứng (như so sánh

ĐIỀU KHIỂN LẬP TRÌNH 1

không bằng nhau \diamond , so sánh nhỏ hơn $<$, hoặc so sánh lớn hơn $>$) ta có thể kết hợp lệnh **NOT** với các lệnh đã có ($=$, $>=$, $<=$)

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

Ví dụ:

- Sử dụng lệnh về Timer và So sánh, Viết chương trình điều khiển đèn giao thông tại ngã 4 theo yêu cầu: Xanh 4s, Vàng 1s, Đỏ 5s. Có 2 nút nhấn ở chế độ bình thường và ưu tiên.
- Thực hiện tương tự với lệnh Counter.

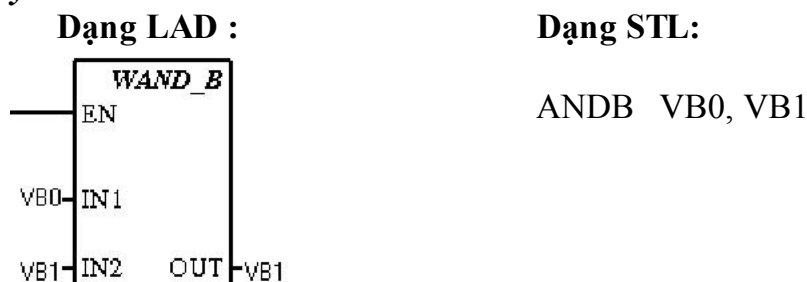
ĐIỀU KHIỂN LẬP TRÌNH 1

7.4 NHÓM LỆNH VỀ CÔNG LOGIC.

Ngoài những lệnh ghép nối tiếp, song song và tổng hợp các tiếp điểm thì tập lệnh của S7-200 còn cung cấp các công logic AND, OR, EXOR thực hiện đối với byte (8 bit hay 8 tiếp điểm), word (16 bit hay 16 tiếp điểm) và double word (32 bit hay 32 tiếp điểm). Sau đây là chi tiết của từng công:

7.4.1 Lệnh AND.

- **Lệnh AND byte :**



Ý nghĩa:

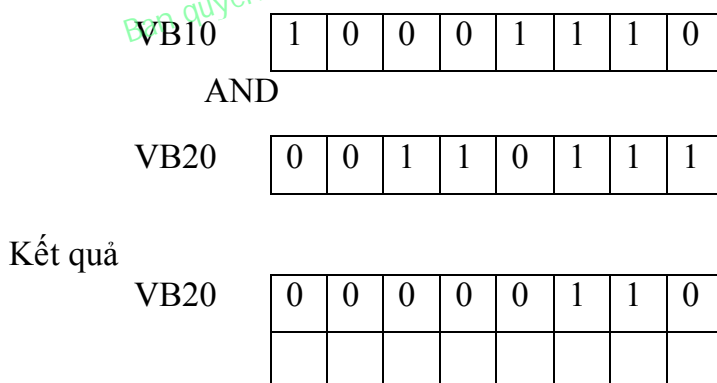
Lệnh thực hiện phép AND từng bit của hai byte ngõ vào IN1 và IN2, kết quả được ghi vào 1 byte ở ngõ ra OUT, địa chỉ ngõ ra có thể giống ngõ vào.

Toán hạng trong câu lệnh thuộc một trong các vùng địa chỉ sau :

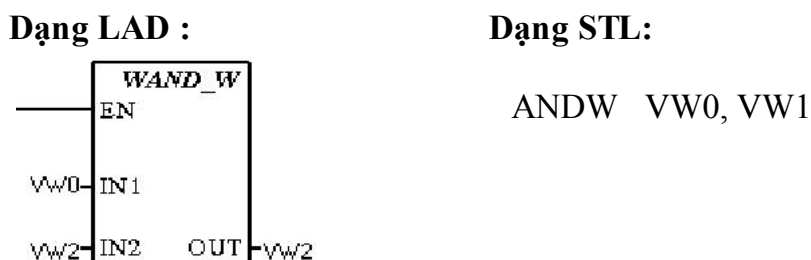
IN1 : VB, T, C, IB, QB, SMB, AC, const

IN2 : VB, T, C, IB, QB, SMB, AC

Ví dụ:



- **Lệnh AND Word :**



Ý nghĩa :

Lệnh thực hiện phép AND từng bit của hai Word ngõ vào IN1 và IN2, kết quả được ghi vào 1 Word ở ngõ ra OUT, địa chỉ ngõ ra có thể giống ngõ vào.

Toán hạng trong câu lệnh thuộc một trong các vùng địa chỉ sau

IN1 : VW, T, C, IW, QW, SMW, AC, const

IN2 : VW, T, C, IW, QW, SMW, AC

ĐIỀU KHIỂN LẬP TRÌNH 1

Ví dụ :

VW10

1	0	1	0	1	1	1	0	0	1	1	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

AND

VW12

1	0	0	0	1	0	1	0	1	1	0	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

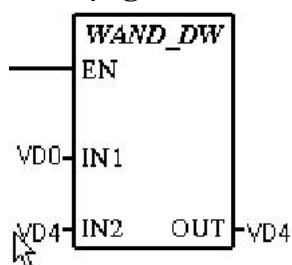
Kết quả

VW12

1	0	0	0	1	0	1	0	0	1	0	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- **Lệnh AND DWord :**

Dạng LAD :



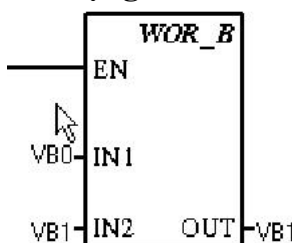
Dạng STL:

ANDD VD0, VD4

7.4.2 Lệnh OR.

- **Lệnh OR byte :**

Dạng LAD :



Dạng STL:

ORB VB0, VB1

Ý nghĩa :

Lệnh thực hiện phép OR từng bit của hai byte ngõ vào IN1 và IN2, kết quả được ghi vào 1 byte ở ngõ ra OUT, địa chỉ ngõ ra có thể giống ngõ vào.

Toán hạng trong câu lệnh thuộc một trong các vùng địa chỉ sau

IN1 : VB, T, C, IB, QB, SMB, AC, const

IN2 : VB, T, C, IB, QB, SMB, AC

Ví dụ:

VD10

1	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

OR

VD20

0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---

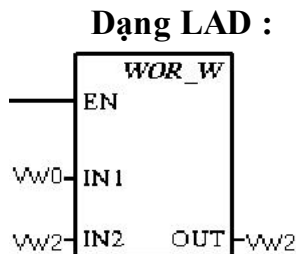
Kết quả

VD20

1	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---

ĐIỀU KHIỂN LẬP TRÌNH 1

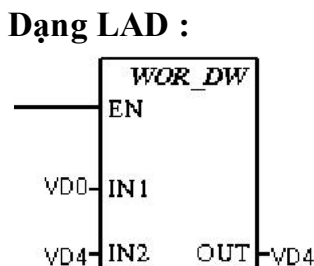
- **Lệnh OR word :**



Dạng STL:

ORW VW0, VW2

- **Lệnh OR Double word :**



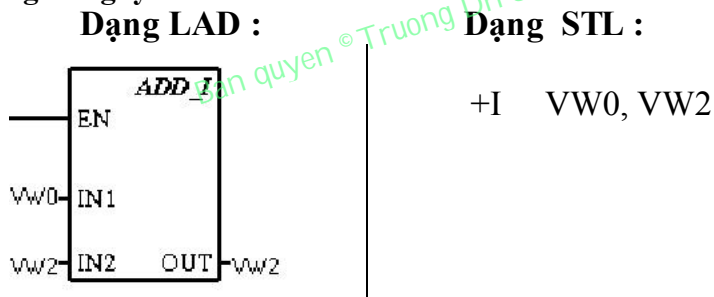
Dạng STL:

ORDW VD0, VD4

7.5 NHÓM LỆNH VỀ CÁC PHÉP TOÁN LOGIC.

7.5.1 Lệnh Cộng.

Lệnh cộng số nguyên 16 bit :



Ý nghĩa :

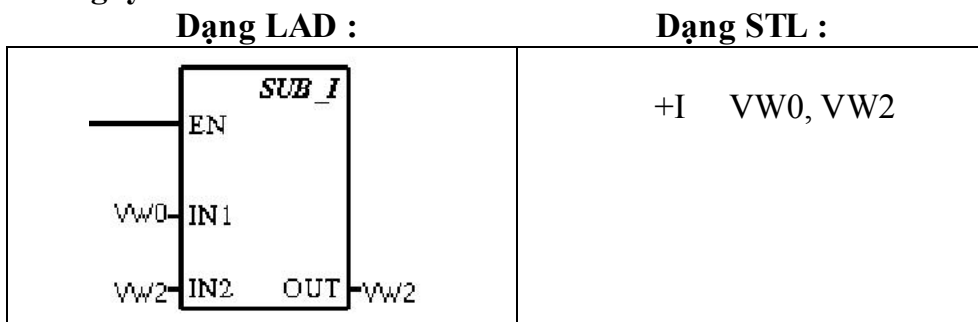
Lệnh thực hiện cộng các số nguyên 16 bit IN1 và IN2 , kết quả là số nguyên 16 bit được ghi vào OUT, $IN1 + IN2 = OUT$, IN2 và OUT có thể cùng địa chỉ, thuộc các vùng nhớ sau

IN1, IN2 : VW, T, C, IW, QW, MW, SMW, AC, AIW, const.

OUT : VW, T, C, IW, QW, MW, SMW, AC, AIW

7.5.2 Lệnh Trừ.

Lệnh trừ số nguyên 16 bit



ĐIỀU KHIỂN LẬP TRÌNH 1

Ý nghĩa :

Lệnh được thực hiện phép trừ các số nguyên 16 bit IN1 và IN2 , kết quả là số nguyên 16 bit và được ghi vào OUT, tức là : $IN1 - IN2 = OUT$, địa chỉ thuộc các vùng nhớ sau

IN1, IN2 : VW,T ,C, IW, QW, MW, SMW, AC, AIW, const.

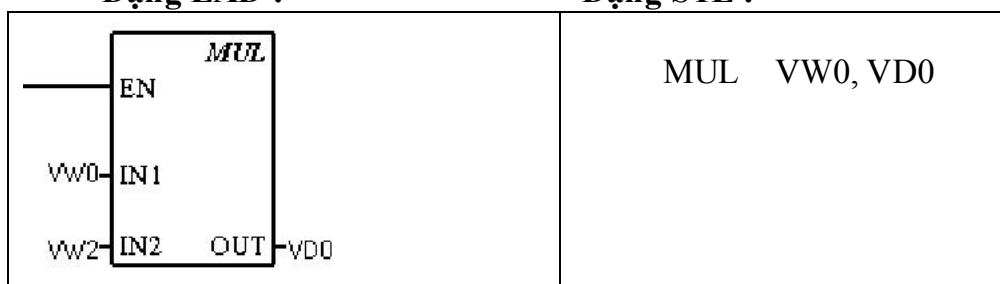
OUT : VW,T ,C, IW, QW, MW, SMW, AC, AIW.

7.5.3 Lệnh Nhân.

Nhân số nguyên 16 bit :

Dạng LAD :

Dạng STL :



Ý nghĩa :

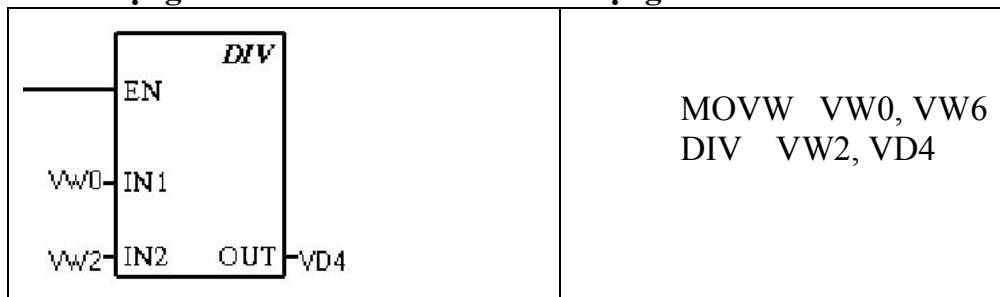
Lệnh thực hiện phép nhân 2 số nguyên 16bit IN1 và IN2. Kết quả 32 bit chứa trong từ kép OUT (4 byte).

7.5.4 Lệnh Chia.

Chia số nguyên 16 bit :

Dạng LAD :

Dạng STL :



Ý nghĩa :

Lệnh thực hiện phép chia số nguyên 16 bit IN1 cho số nguyên 16 bit IN2. Kết quả 32 bit chứa trong từ kép OUT (4 byte) gồm thương số ghi trong mảng 16 bit từ bit 0 đến bit 15 (từ thấp) và phần dư cũng 16 bit ghi trong mảng từ bit 16 đến bit 31 (từ cao). Trong lệnh này có sử dụng các bit nhớ đặc biệt sau để báo trạng thái.

Kết quả tính	SM1.0	SM1.1	SM1.2	SM1.3
= 0	1			
Báo tràn		1		
Số âm			1	
Mẫu = 0				1

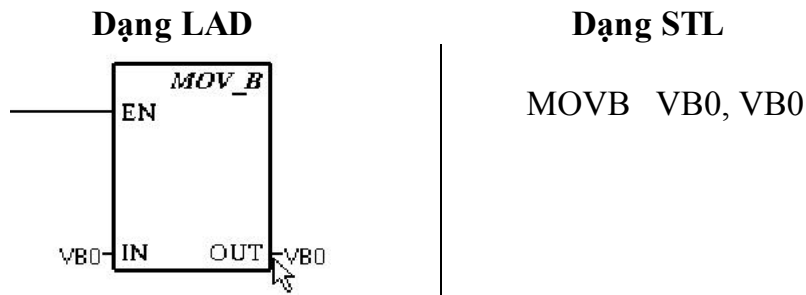
ĐIỀU KHIỂN LẬP TRÌNH 1

7.6 NHÓM LỆNH DI CHUYỂN VÀ BIẾN ĐỔI DỮ LIỆU.

7.6.1 Lệnh Di chuyển.

Các lệnh di chuyển thực hiện việc di chuyển hoặc sao chép dữ liệu từ vùng này sang vùng khác trong bộ nhớ. Lệnh dịch chuyển thực hiện việc di chuyển hay sao chép nội dung một byte, một từ đơn, hoặc một từ kép từ vùng này sang vùng khác trong bộ nhớ. Lệnh trao đổi nội dung của hai byte trong một từ đơn thực hiện việc chuyển nội dung của byte thấp sang byte cao và ngược lại chuyển nội dung của byte cao sang byte thấp của từ đó. Sau đây là chi tiết của từng lệnh.

- **MOV_B :**



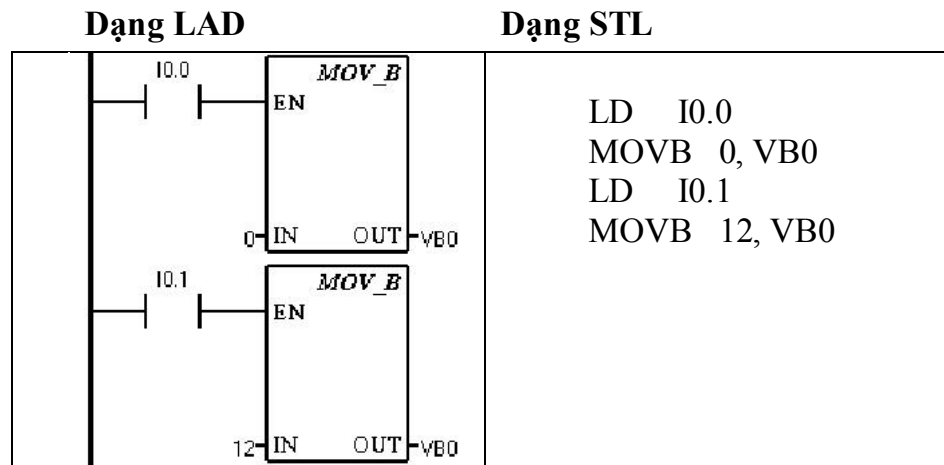
Ý nghĩa:

Lệnh sao chép nội dung của byte ở địa chỉ ngõ vào IN sang byte có địa chỉ ở ngõ ra OUT. Địa chỉ của byte ngõ vào IN và địa chỉ byte ngõ ra OUT có thể giống nhau, thuộc các vùng sau:

IN : VB, IB, QB, MB, SMB, AC, const

OUT: VB, IB, QB, MB, SMB, AC,

Ví dụ :



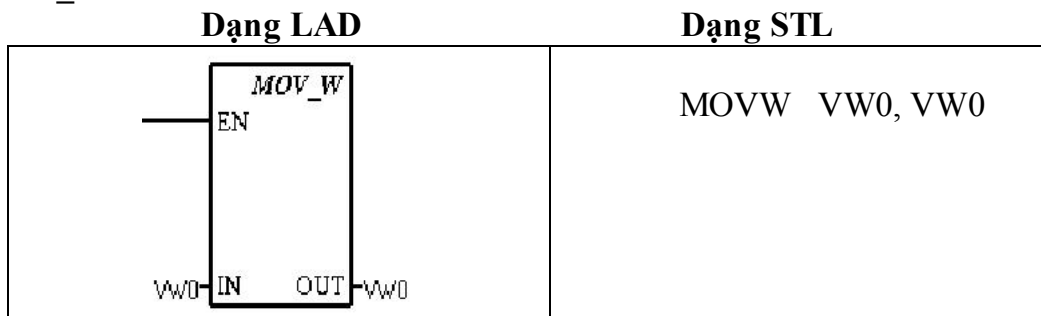
Giải thích :

Nếu tiếp điểm I0.0 đóng thì lấy giá trị 0 ghi vào byte VB0 (xóa VB0)

Tiếp theo đóng tiếp điểm I0.1 thì lấy số 12 ghi vào VB0. Kết quả địa chỉ byte VB0 có giá trị bằng 12 (nhị phân).

ĐIỀU KHIỂN LẬP TRÌNH 1

- **MOV_W :**



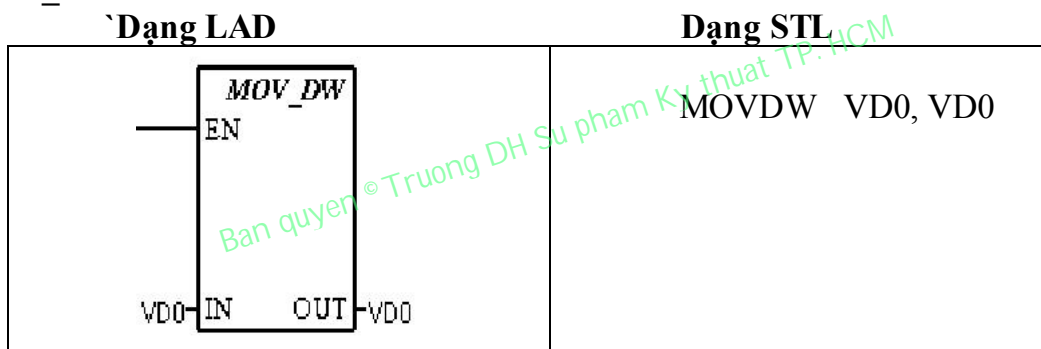
Ý nghĩa :

Lệnh sao chép nội dung của Word ở địa chỉ ngõ vào IN sang Word có địa chỉ ở ngõ ra OUT, địa chỉ ngõ ra có thể giống ngõ vào, nằm trong các vùng sau:

IN: VW, IW, QW, MW, SMW, AC, const

OUT: VW, IW, QW, MW, SMW, AC

- **MOV_DW :**



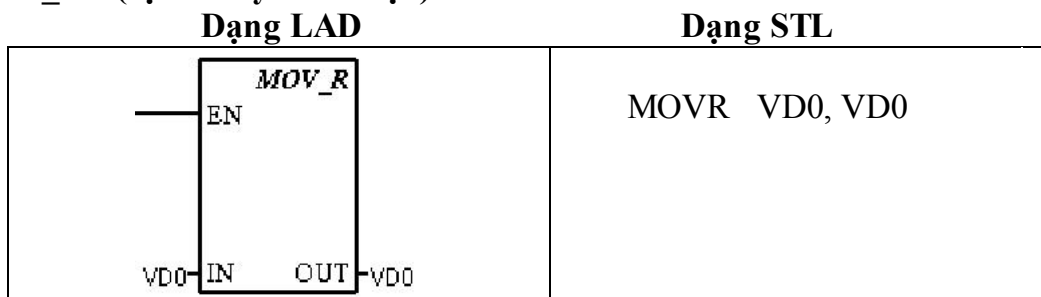
Ý nghĩa :

Lệnh sao chép nội dung của DWord ở địa chỉ ngõ vào IN sang DWord có địa chỉ ở ngõ ra OUT, địa chỉ ngõ ra có thể giống ngõ vào, nằm trong các vùng sau:

IN: VD, ID, QD, MDW, SMD, AC, const

OUT: VD, ID, QD, MD, SMD, AC

- **MOV_R : (dịch chuyển số thực)**



ĐIỀU KHIỂN LẬP TRÌNH 1

Ý nghĩa :

Lệnh sao chép nội dung của số thực chứa trong double word có địa chỉ ở ngõ vào IN sang double word có địa chỉ ở ngõ vào OUT, địa chỉ ngõ ra có thể giống ngõ vào, thường nằm trong các vùng sau:

IN: VD, ID, QD, MD, SMD, AC, const

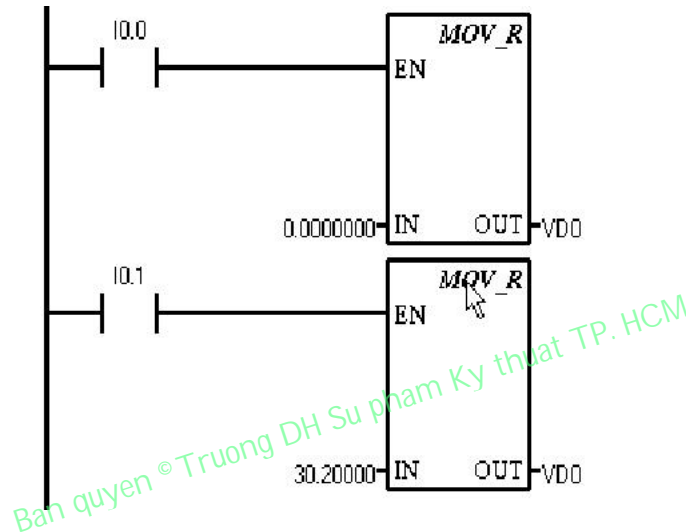
OUT: VD, ID, QD, MD, SMD, AC

Dữ liệu được ghi vào trong các địa chỉ này theo nguyên tắc sau :

Phần nguyên ghi vào word thấp

Phần thập phân ghi vào word cao

Ví dụ :



Giải Thích :

Tiếp điểm I0.0 đóng thì xóa double word 0 (VD0), tiếp điểm I0.1 đóng thì ghi số thực 30,2 vào double word VD0, kết quả như sau :

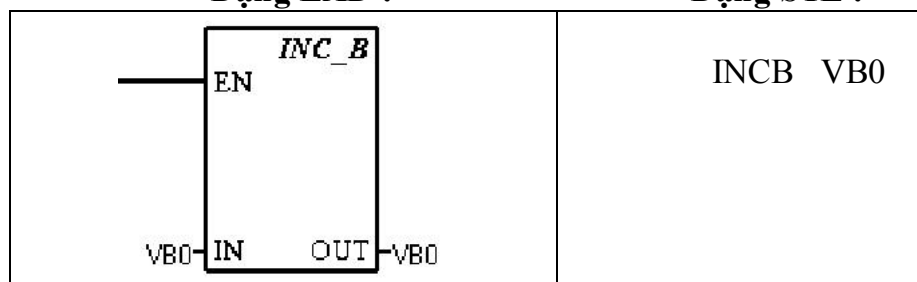


7.6.2 Lệnh Tăng Giảm.

- Lệnh cộng số nguyên 1 vào nội dung byte :

Dạng LAD :

Dạng STL :



Ý nghĩa :

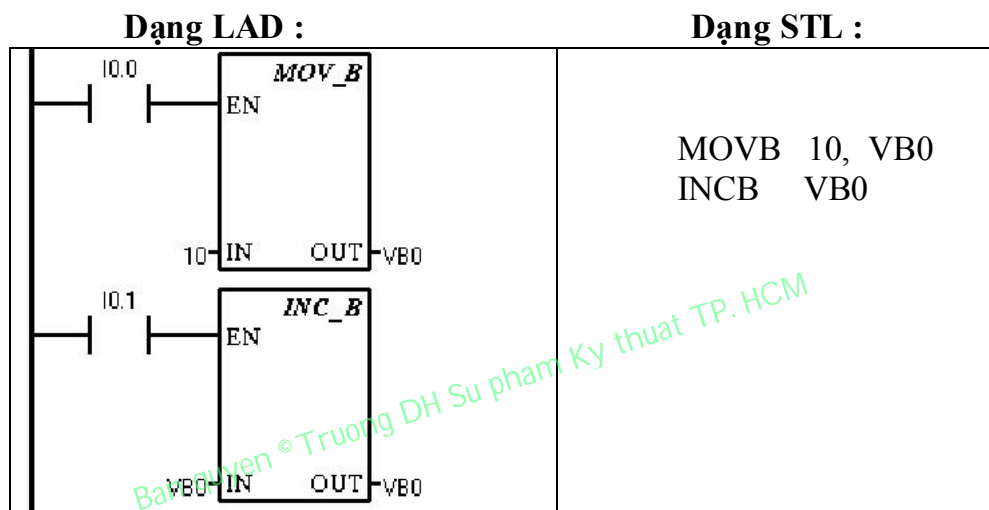
Lệnh này có tác dụng cộng số nguyên 1 đơn vị với nội dung của byte có địa chỉ ở ngõ vào IN, kết quả được ghi vào byte có địa chỉ ở ngõ ra OUT. Byte IN và byte OUT có

ĐIỀU KHIỂN LẬP TRÌNH 1

thể cùng địa chỉ. Lệnh này có sử dụng các bit nhớ đặc biệt SM1.0, SM1.1, SM1.2 để báo trạng thái kết quả phép tính theo nguyên tắc như sau :

Kết quả tính	SM1.0	SM1.1	SM1.2
= 0	1		
Số âm			1
> byte		1	

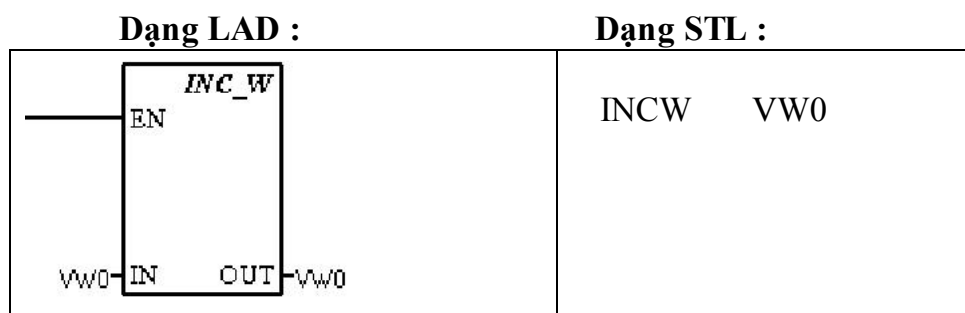
Ví dụ :



Giải thích :

Tiếp điểm I0.0 đóng thì số 10 được ghi vào VB0, tiếp điểm I0.1 đóng thì nội dung của VB0 tăng lên 1 đơn vị và kết quả được lưu lại VB0. Lúc này VB0 = 11.

- **Lệnh cộng số nguyên 1 vào nội dung word :**



Ý nghĩa :

Lệnh này có tác dụng cộng số nguyên 1 đơn vị với nội dung của word có địa chỉ ở ngõ vào IN, kết quả được ghi vào word có địa chỉ ở ngõ ra OUT , word IN và word OUT có thể cùng địa chỉ và ở lệnh này có sử dụng các bit nhớ đặc biệt SM1.0, SM1.1, SM1.2 để báo trạng thái kết quả phép tính theo nguyên tắc như sau:

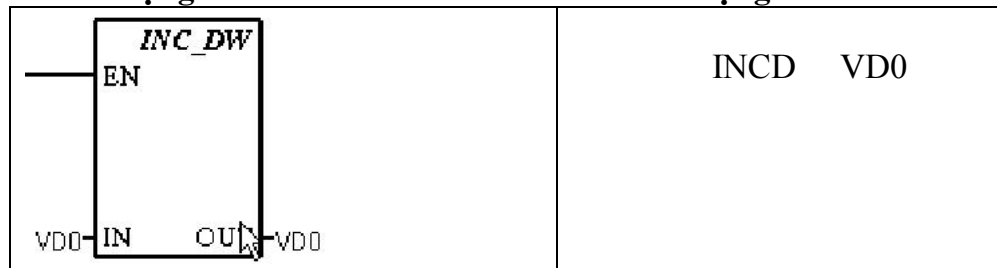
ĐIỀU KHIỂN LẬP TRÌNH 1

Kết quả tính	SM1.0	SM1.1	SM1.2
= 0	1		
Số âm			1
> byte		1	

- **Lệnh cộng số nguyên 1 vào nội dung double word :**

Dạng LAD :

Dạng STL :



Ý nghĩa :

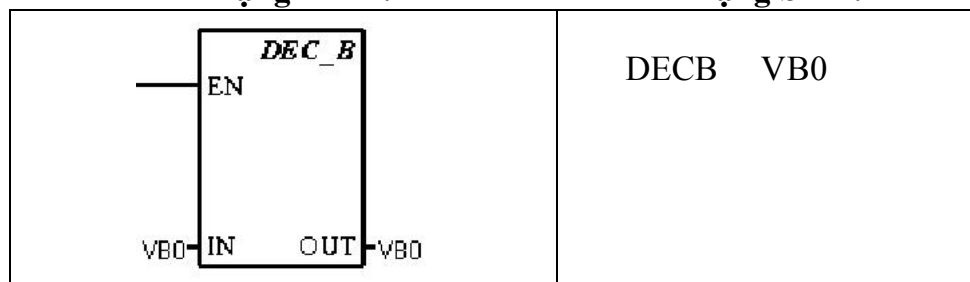
Lệnh này có tác dụng cộng số nguyên 1 đơn vị với nội dung của double word có địa chỉ ở ngõ vào IN, kết quả được ghi vào double word có địa chỉ ở ngõ ra OUT, double word IN và double word OUT có thể cùng địa chỉ. Lệnh này cũng sử dụng các bit nhớ đặc biệt SM1.0, SM1.1, SM1.2 để báo trạng thái kết quả phép tính theo nguyên tắc như sau :

Kết quả tính	SM1.0	SM1.1	SM1.2
= 0	1		
Số âm			1
> byte		1	

- **Lệnh trừ nội dung của byte đi 1 đơn vị :**

Dạng LAD :

Dạng STL :



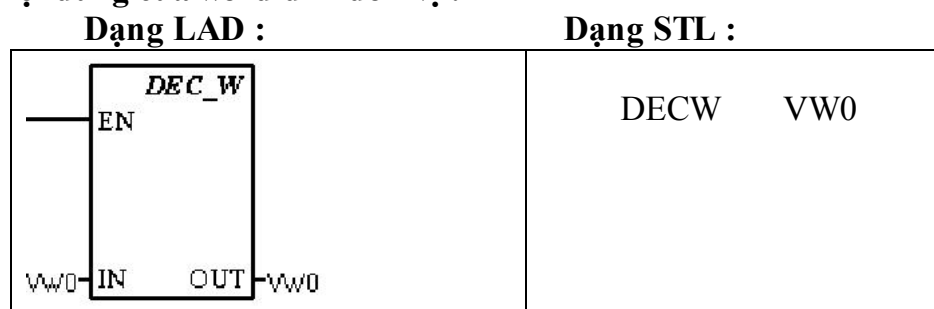
Ý nghĩa :

Lệnh này có tác dụng lấy nội dung của byte có địa chỉ ở ngõ vào IN trừ đi 1 đơn vị, kết quả được ghi vào byte có địa chỉ ở ngõ ra OUT, byte IN và byte OUT có thể cùng địa chỉ và ở lệnh này cũng sử dụng các bit nhớ đặc biệt SM1.0, SM1.1, SM1.2 để báo trạng thái kết quả phép tính theo nguyên tắc như sau :

ĐIỀU KHIỂN LẬP TRÌNH 1

Kết quả tính	SM1.0	SM1.1	SM1.2
= 0	1		
Số âm			1
> byte		1	

- **Lệnh trừ nội dung của word đi 1 đơn vị :**

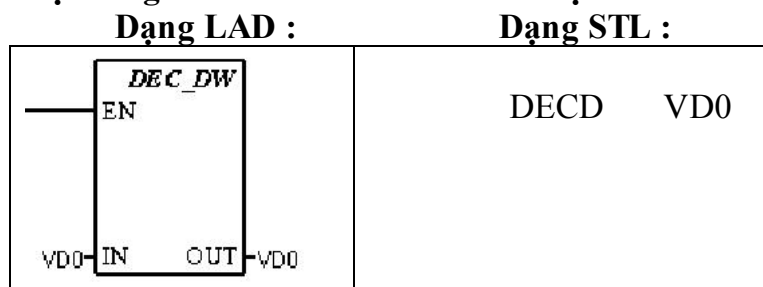


Ý nghĩa :

Lệnh này có tác dụng lấy nội dung của word có địa chỉ ở ngõ vào IN trừ đi 1 đơn vị , kết quả được ghi vào word có địa chỉ ở ngõ ra OUT, trong lệnh word IN và word OUT có thể cùng địa chỉ và ở lệnh này cũng sử dụng các bit nhớ đặc biệt SM1.0, SM1.1, SM1.2 để báo trạng thái kết quả phép tính theo nguyên tắc như sau:

Kết quả tính	SM1.0	SM1.1	SM1.2
= 0	1		
Số âm			1
> byte		1	

- **Lệnh trừ nội dung của double word đi 1 đơn vị :**



Ý nghĩa :

Lệnh này có tác dụng lấy nội dung của double word có địa chỉ ở ngõ vào IN trừ đi 1 đơn vị , kết quả được ghi vào double word có địa chỉ ở ngõ ra OUT , trong lệnh double word IN và double word OUT có thể cùng địa chỉ và ở lệnh này cũng sử dụng các bit nhớ đặc biệt SM1.0, SM1.1, SM1.2 để báo trạng thái kết quả phép tính theo nguyên tắc như sau :

ĐIỀU KHIỂN LẬP TRÌNH 1

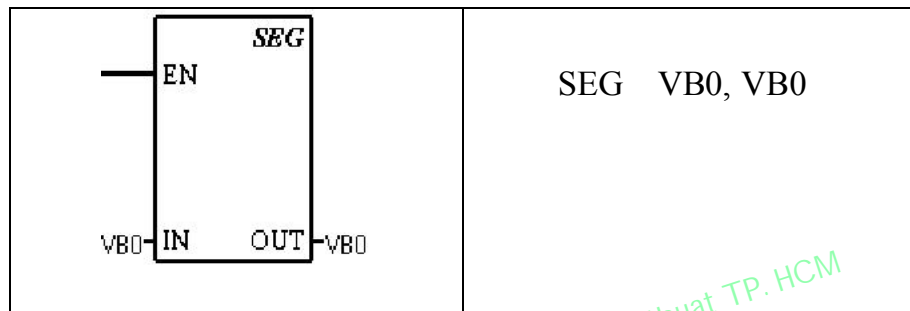
Kết quả tính	SM1.0	SM1.1	SM1.2
= 0	1		
Số âm			1
> byte		1	

7.6.3 Lệnh Chuyển đổi.

- Chuyển đổi số nguyên sang mã led 7 đoạn:

Dạng LAD :

Dạng STL :



Ý nghĩa :

Lệnh này có tác dụng chuyển đổi các số trong hệ thập lục phân từ 0 đến F (dạng nhị phân) chứa trong 4 bit thấp của byte có địa chỉ ở ngõ vào IN thành giá trị BIT chứa trong 8 bit của byte có địa chỉ ở ngõ ra OUT tương ứng với thanh led 7 đoạn CK, địa chỉ ngõ ra có thể giống ngõ vào, nằm trong những vùng sau:

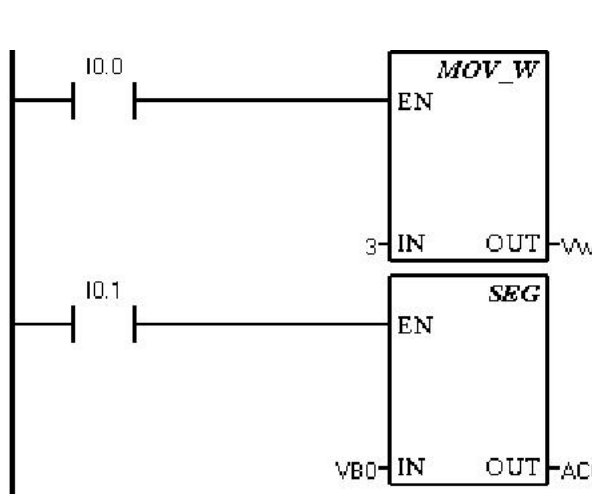
IN: VB, IB, QB, MB, SMB, AC, const

OUT: VB, IB, AB, MB, SMB, AC

Ví dụ :

Dạng LAD:

Dạng STL:

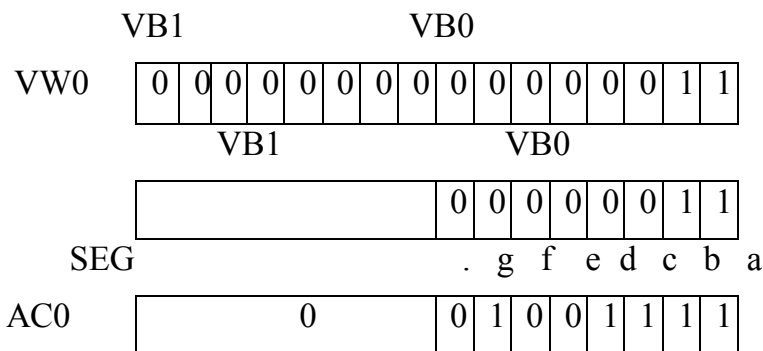


```
LD I0.0
MOVW +3, VW0
LD I0.1
SEG VB0, AC0
```

Giải thích :

ĐIỀU KHIỂN LẬP TRÌNH 1

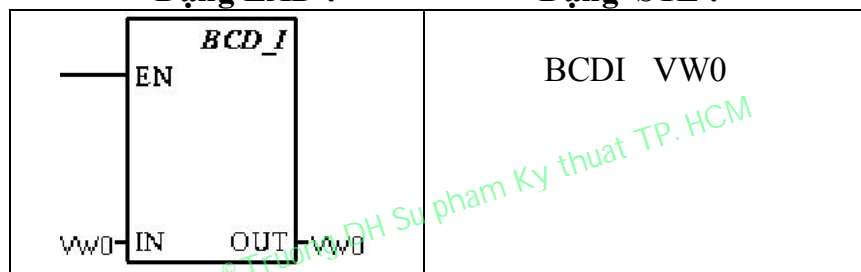
Khi tiếp điểm I0.0 đóng thì số 7 được ghi vào VW0, sau đó tiếp điểm I0.1 đóng thì giá trị chứa trong 4 bit thấp của byte VB0 chuyển thành 8 bit chứa trong thanh ghi AC0. Ta có thể minh họa theo bit như sau :



- Lệnh chuyển đổi mã BCD sang số nguyên:

Dạng LAD :

Dạng STL :



Ý nghĩa:

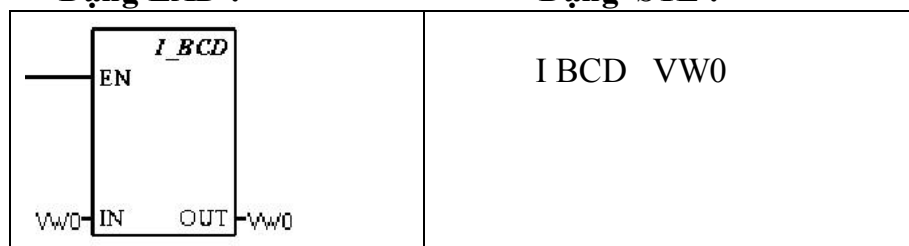
Lệnh này thực hiện phép biến đổi một số dạng mã BCD 16 bit chứa trong word có địa chỉ ở ngõ vào IN sang số nguyên dạng nhị phân 16 bit chứa trong word có địa chỉ ở ngõ ra OUT, địa chỉ ngõ ra có thể giống ngõ vào, thường nằm trong các vùng sau :

- IN: VW, T, C, IW, QW, MW, SMW, AC, AIW, const
- OUT: VW, T, C, IW, QW, MW, SMW, AC.

- Lệnh chuyển đổi số nguyên sang mã BCD:

Dạng LAD :

Dạng STL :



Ý nghĩa :

Lệnh này thực hiện phép biến đổi một số nguyên dạng nhị phân 16 bit chứa trong word có địa chỉ ở ngõ vào IN sang mã BCD 16 bit chứa trong word có địa chỉ ở ngõ ra OUT, địa chỉ ngõ ra có thể giống ngõ vào, thường nằm trong các vùng sau :

- IN : VW, T, C, IW, QW, MW, SMW, AC, AIW, const
- OUT : VW, T, C, IW, QW, MW, SMW, AC.

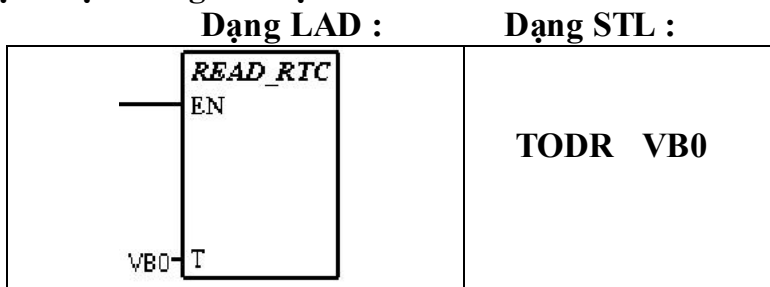
ĐIỀU KHIỂN LẬP TRÌNH 1

7.7 LỆNH VỀ ĐỒNG HỒ THỜI GIAN THỰC.

Trong thiết bị điều khiển lập trình PLC 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 ngày trong tuần. Đồng hồ này được cấp điện liên tục bởi nguồn pin 3V.

Khi thực hiện lập trình cho các hệ thống tự động điều khiển cần cập nhật giá trị đồng hồ thời gian thực này ta sử dụng 2 lệnh sau :

- **Lệnh đọc thời gian thực:**



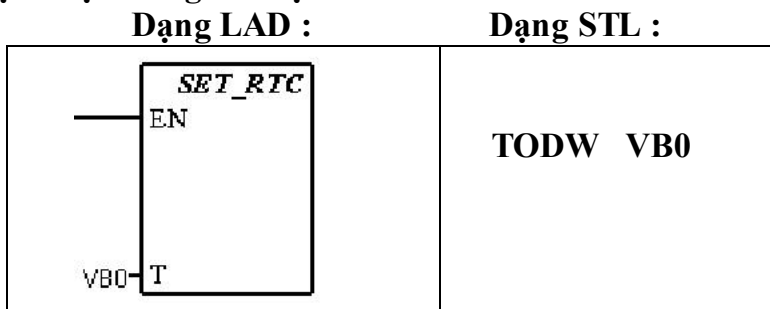
Ý nghĩa :

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 theo thứ tự như sau:

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

Trong đó byte đầu tiên được chỉ định bởi toán hạng T trong câu lệnh, byte 7 chỉ sử dụng 4 bit thấp để lưu giá trị các ngày trong tuần.

- **Lệnh đặt thời gian thực:**



ĐIỀU KHIỂN LẬP TRÌNH 1

Ý nghĩa :

Lệnh này có tác dụng 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. Trong đó T thuộc 1 trong những vùng nhớ sau : VB, IB, QB, MB, SMB.

Nếu cần điều chỉnh các thông số về năm, tháng, ngày, giờ, phút, giây, ngày trong tuần thì điều chỉnh các byte như sau :

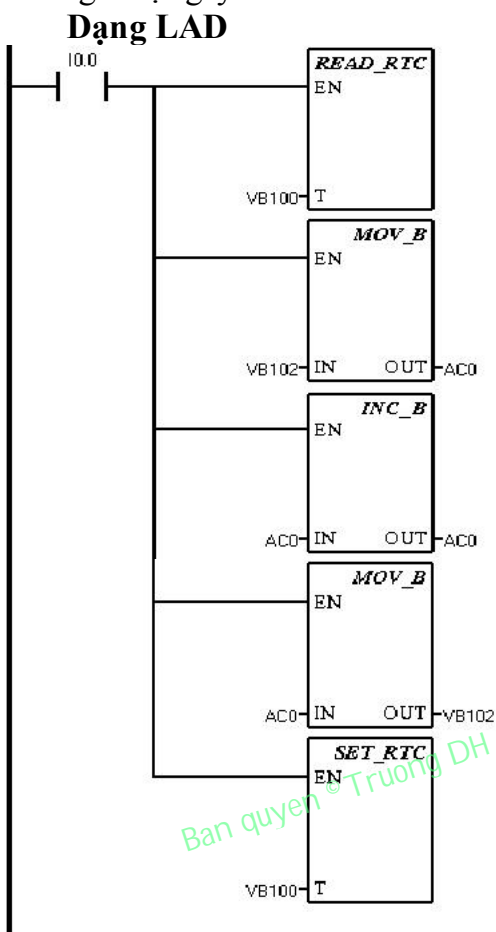
Byte 0	Năm (0 → 99)
Byte 1	Tháng (1 → 12)
Byte 2	Ngày (1 → 31)
Byte 3	Giờ (0 → 23)
Byte 4	Phút (0 → 59)
Byte 5	Giây (0 → 59)
Byte 6	0
Byte 7	Ngày trong tuần (1 → 7)

Ban quyền © Truong DH Su pham Ky thuat TP. HCM

ĐIỀU KHIỂN LẬP TRÌNH 1

Ví dụ :

Để điều chỉnh giá trị ngày ta làm như sau :



Dạng STL

```
LD I0.0
TODR VB100
MOVB VB102, AC0
INCB AC0
MOVB AC0, VB102
TODW VB100
```

Giá trị trước xử lý :

VB100	2003
B101	02
VB102	30
B103	09
VB104	20
VB105	35
VB106	0
VB107	4

Sau khi xử lý :

VB100	2003
VB101	02
VB102	31
VB103	09
VB104	20
VB105	35
VB106	0
VB107	4

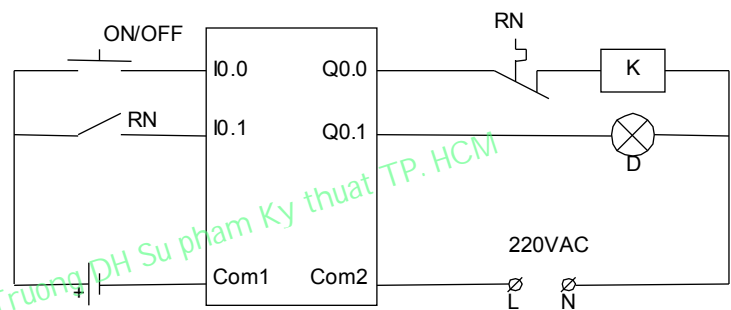
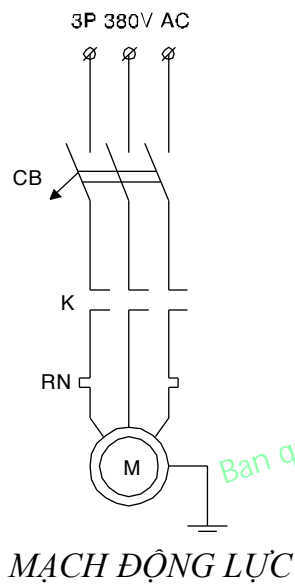
ĐIỀU KHIỂN LẬP TRÌNH 1

BÀI TẬP

BT1:

Lập trình mở máy động cơ dùng 1 nút nhấn ON/OFF, hoạt động theo yêu cầu sau:

- Nhấn nút ON/OFF lần 1 động cơ chạy, đèn D sáng.
- Nhấn nút ON/OFF lần 2 động cơ dừng, đèn D tắt
- Khi có sự cố như quá tải, rò le nhiệt RN tác động ,động cơ dừng và đèn D sáng nhấp nháy liên tục.
- Sau khi xử lý sự cố xong nhấn nút RESET ở rò le nhiệt, sau đó nhấn ON động cơ chạy lại bình thường.

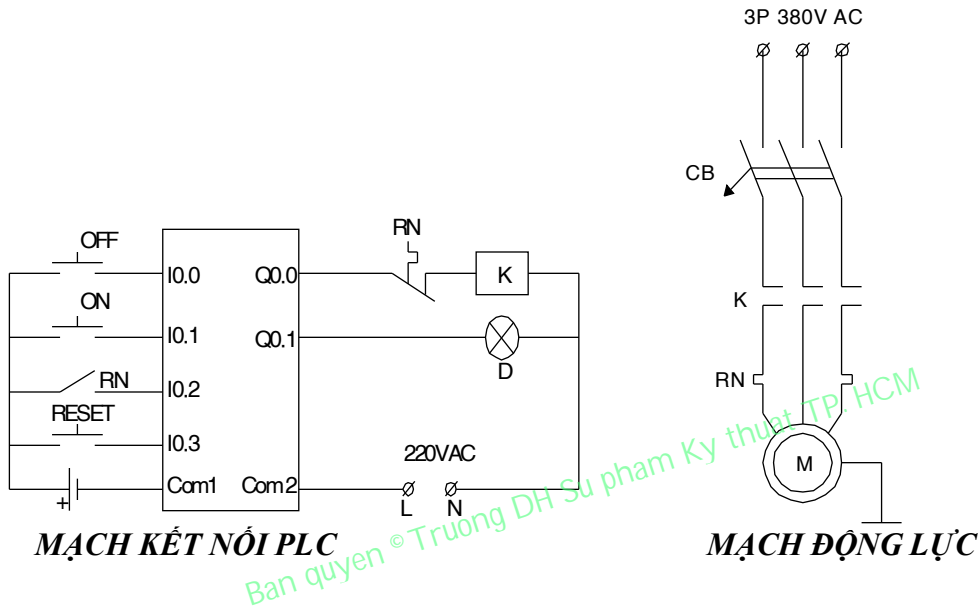


ĐIỀU KHIỂN LẬP TRÌNH 1

BT2 :

Lập trình mở máy động cơ dùng 2 nút nhấn ON/OFF, hoạt động theo yêu cầu sau:

- Nhấn nút ON động cơ sẽ chạy, đèn D sáng.
- Nhấn nút OFF động cơ dừng, đèn D tắt.
- Khi có sự cố như qua tải rò le nhiệt RN tác động , động cơ dừng và đèn D sẽ nhấp nháy liên tục.
- Sau khi xử lý sự cố xong nhấn nút RESET ở rò le nhiệt và bảng điều khiển, động cơ chạy lại bình thường

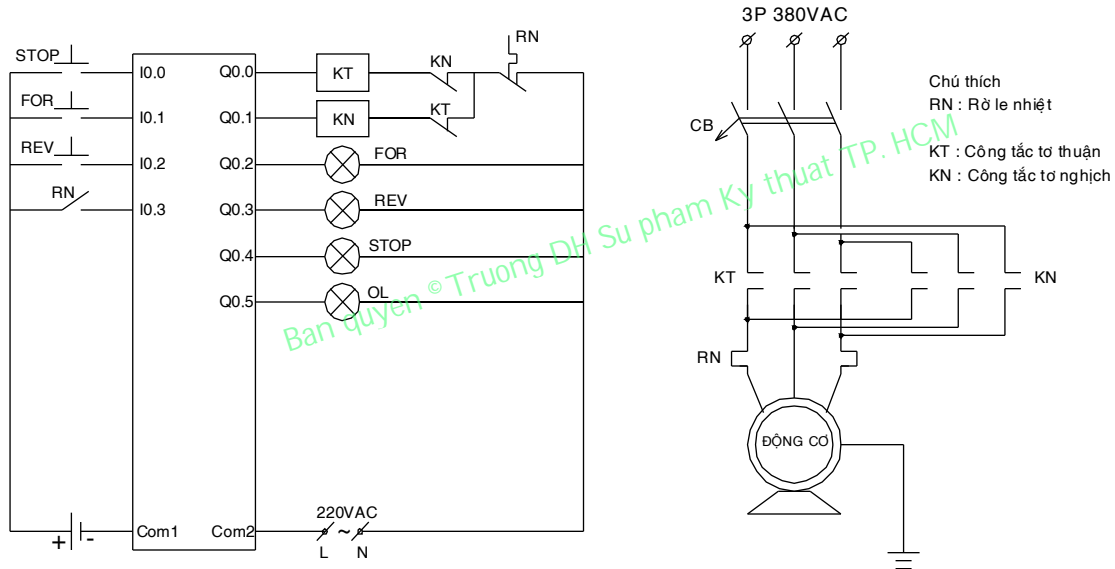


ĐIỀU KHIỂN LẬP TRÌNH 1

BT3 :

Mở máy động cơ 3 pha quay thuận nghịch dùng nút nhấn FOR,REV,OFF, hoạt động theo yêu cầu sau:

- Đèn STOP sáng báo hiệu động cơ đang dừng.
- Nút nhấn FOR , động cơ chạy thuận, đèn FOR sáng, muốn dừng động cơ nhấn STOP, đèn FOR tắt.
- Nút nhấn REV, động cơ chạy nghịch, đèn REV sáng, muốn dừng động cơ nhấn STOP, đèn REV tắt.
- Khi động cơ đang chạy muốn chuyển đổi chiều quay động cơ nhấn STOP trước, sau đó nhấn FOR hay REV.
- Khi có sự cố như quá tải, rờ le nhiệt RN tác động ,động cơ dừng, đèn FOR và REV tắt, đèn OL sáng nhấp nháy liên tục.
- Sau khi xử lý sự cố xong nhấn nút RESET ở rờ le nhiệt, sau đó nhấn FOR hay REV động cơ chạy lại bình thường.

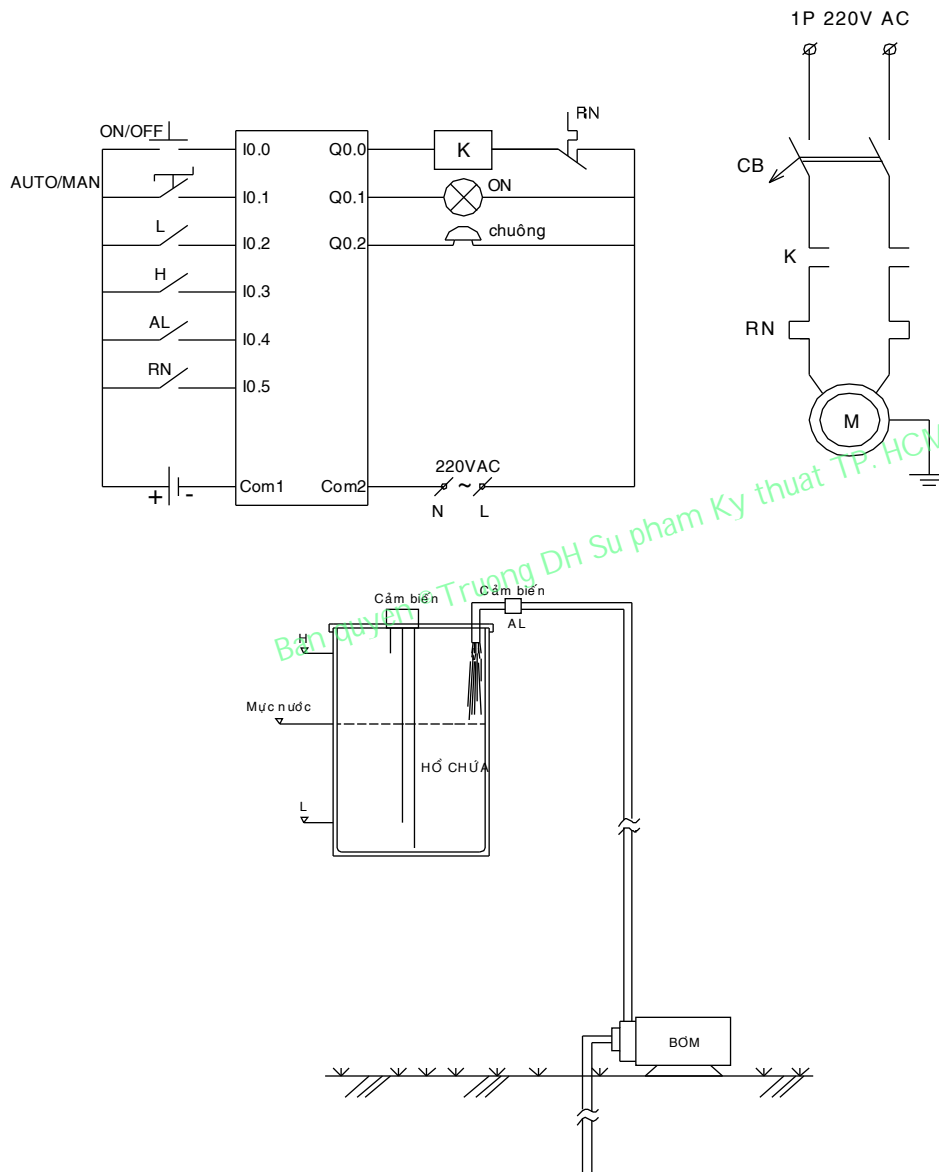


ĐIỀU KHIỂN LẬP TRÌNH 1

BT4 :

Điều khiển động cơ 1 pha bơm nước giếng lên hồ chứa với yêu cầu sau :

- Nước hồ cạn động cơ chạy
- Nước hồ đầy động cơ dừng.
- Khi động cơ đang chạy nếu nước trong ống không có 30 giây sau động cơ dừng lại, chuông kêu báo mỗi nước.



ĐIỀU KHIỂN LẬP TRÌNH 1**BT5 :**

Lập trình PLC điều khiển dây chuyền sản xuất gồm 3 động cơ hoạt động như sau :

- Nhấn nút khởi động cho động cơ Đ1 chạy, sau 5s cho phép vận hành Đ2.
- Nhấn nút khởi động cho động cơ Đ2 chạy, đồng thời lúc này động cơ Đ1 ngừng, sau 10s thì cho phép vận hành động cơ Đ3.
- Nhấn nút khởi động động cơ Đ3 chạy đồng thời lúc này động cơ Đ2 ngừng.
- Nhấn nút dừng thì bất kỳ động cơ nào chạy cũng phải ngừng.

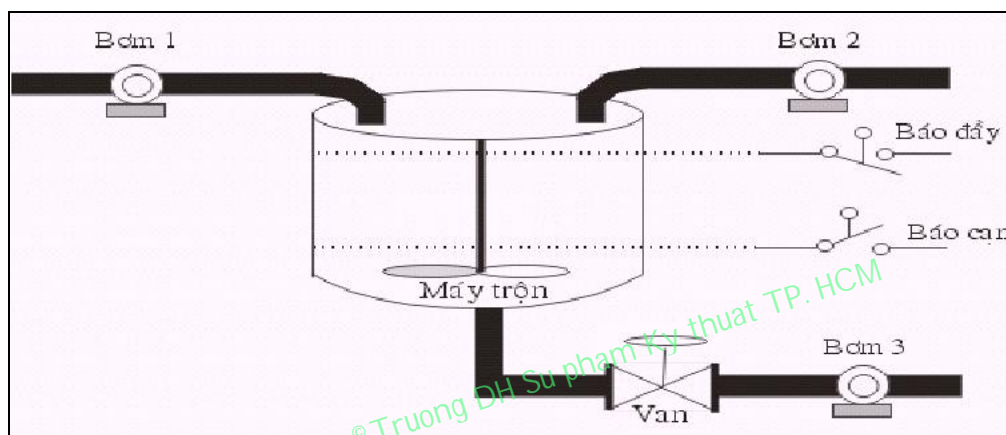
Bản quyền © Truong DH Su pham Ky thuat TP. HCM

ĐIỀU KHIỂN LẬP TRÌNH 1

BT6 :

Lập trình PLC điều khiển bồn trộn hóa chất từ 2 loại khác nhau hoạt động như sau: Nhấn nút khởi động, bơm 1 và 2 làm việc bơm 2 loại hóa chất vào bồn trộn, khi hóa chất đã đầy thì 2 bơm ngưng và máy trộn hoạt động trong vòng 5 phút. Khi trộn xong thì van xả và bơm 3 hoạt động bơm hoá chất để sử dụng. Khi sử dụng hết thì van xả và bơm 3 ngưng làm việc động thời lúc đó bơm 1 và 2 hoạt động trở lại cho chu kỳ mới. Nếu trong quá tình hoạt động có sự cố hoặc bấm nút dừng thì hệ thống sẽ dừng ngay. Trong lúc đó bơm 1 và 2 cũng có 2 nút dừng riêng.

Sơ đồ bồn trộn :



ĐIỀU KHIỂN LẬP TRÌNH 1

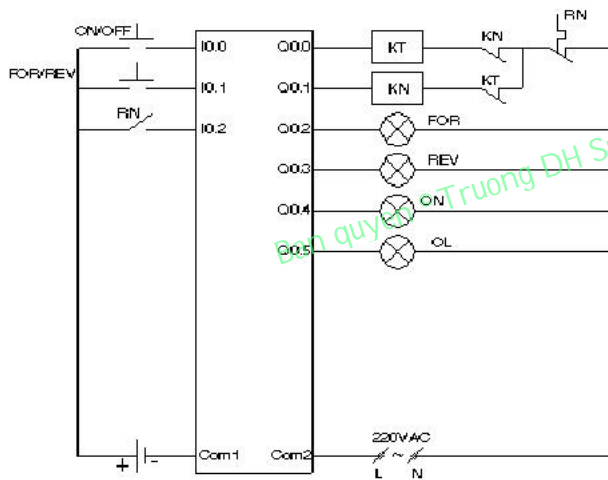
BÀI TẬP VỀ NHÀ

BT 1:

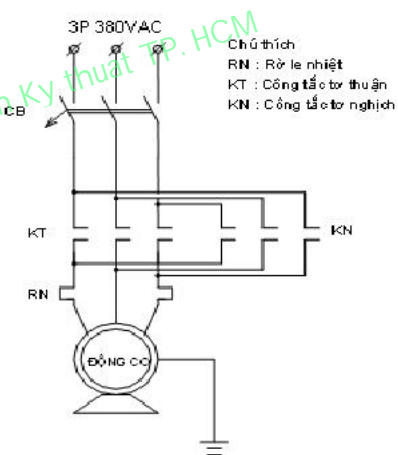
Mở máy động cơ 3 pha thuận nghịch dùng 2 nút nhấn ON/OFF và FOR/REV theo yêu cầu sau:

- Nếu đèn FOR sáng thì khi nhấn nút ON/OFF lần 1 đèn ON sáng báo hiệu động cơ đang chạy thuận. Nếu đèn REV sáng thì nhấn nút ON/OFF lần 1, đèn ON sáng báo hiệu động cơ đang chạy nghịch.
- Khi động cơ đang chạy muốn đổi chiều quay của động cơ, phải nhấn nút ON/OFF lần 2 để dừng động cơ và nhấn nút FOR/REV sau đó nhấn lại nút ON/OFF để chạy động cơ.
- Khi có sự cố quá tải, rờ le RN tác động làm động cơ dừng, đèn ON sẽ tắt, đèn FOR hay REV sẽ sáng báo hiệu chiều quá tải và đèn OL sẽ sáng nhấp nháy.
- Khi sự cố được khắc phục, nhấn RESET ở rờ le nhiệt và nhấn nút ON/OFF để chạy động cơ.

Chú ý xem đèn hiển thị FOR và REV để biết chiều quay động cơ trước khi nhấn ON/OFF..



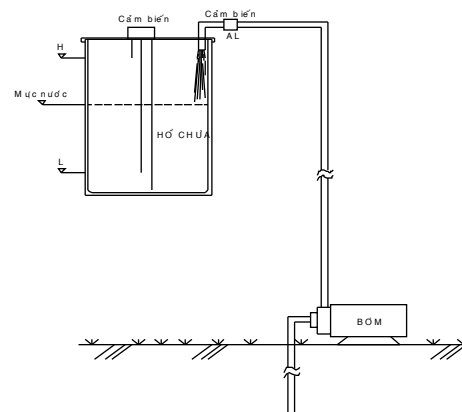
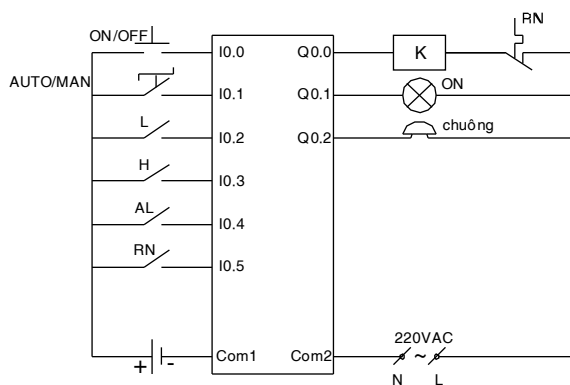
SƠ ĐỒ KẾT NỐI PLC



SƠ ĐỒ MẠCH ĐỘNG LỰC

BT 2:

Chương trình điều khiển động cơ 1 pha bơm nước giếng lên hồ chứa theo hình vẽ, với yêu cầu:



SƠ ĐỒ LẮP ĐẶT THIẾT BỊ

- Ở chế độ điều khiển bằng tay:

ĐIỀU KHIỂN LẬP TRÌNH 1

Đề công tắc A/M về phía M, nhấn nút ON/OFF lần 1 động cơ chạy, đèn ON sáng. Nhấn nút ON/OFF lần 2 thì động cơ dừng, đèn ON tắt.

- Ở chế độ tự động:

Đề công tắc A/M về phía A.

Khi nước cạn dưới mức L thì động cơ chạy, bơm nước lên hồ.

Khi nước dâng lên đến mức H hay trên mức H thì động cơ dừng.

Khi mực nước ở giữa mức L và mức H thì động cơ sẽ giữ nguyên trạng thái. Nếu đang bơm thì tiếp tục bơm, nếu đang dừng thì tiếp tục dừng.

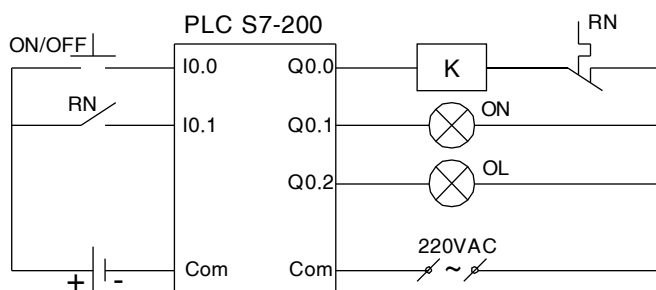
Khi động cơ đang chạy mà trong ống không có nước trên 30 giây thì dừng, chuông sẽ kêu. Chuyển công tắc về phía MAN chuông hết kêu, nhấn nút ON/OFF lần 1 để bơm chạy mỗi nước. Nếu đã có nước lên hồ thì nhấn nút ON/OFF lần 2 để dừng bơm và chuyển công tắc AUTO/MAN về phía AUTO để cho bơm chạy ở chế độ tự động.

Ghi chú : Cảm biến L, H, AL tác động khi đầu dò có nước và ngược lại.

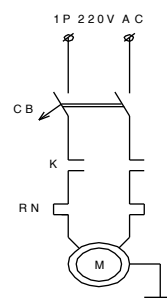
BT 3:

Điều khiển động cơ chạy 5s dừng 10s, lặp lại 5 lần rồi dừng hẳn, dùng 1 nút nhấn ON/OFF để điều khiển

- Nhấn nút ON/OFF lần 1 động cơ chạy, đèn sáng trong 5s rồi động cơ dừng, đèn ON tắt trong 10s và lặp đi lặp lại 5 lần theo chu kỳ rồi dừng hẳn. Sau đó nếu nhấn tiếp nút nhấn ON/OFF động lại chạy 5s và dừng 10s lặp lại theo chu kỳ 5 lần rồi dừng hẳn.
- Nếu động cơ đang hoạt động theo chương trình mà ta nhấn ON/OFF động cơ dừng hẳn, đèn ON tắt.
- Nếu có sự cố quá tải, rò rỉ nhiệt RN tác động, động cơ dừng hẳn, đèn ON tắt, đèn OL nhấp nháy.
- Khi khắc phục xong sự cố nhấn RESET rò rỉ nhiệt RN, mạch hoạt động lại bình thường.



Sơ đồ kết nối PLC



Sơ đồ mạch động lực

BT 4:

Chương trình điều khiển đèn giao thông theo yêu cầu sau:

- Ở chế độ tay:

Chuyển nút AUTO-MAN về phía MAN thì đèn Đỏ 1 và Xanh 2 sáng trước, đèn Đỏ 2 và Xanh 1 tắt.

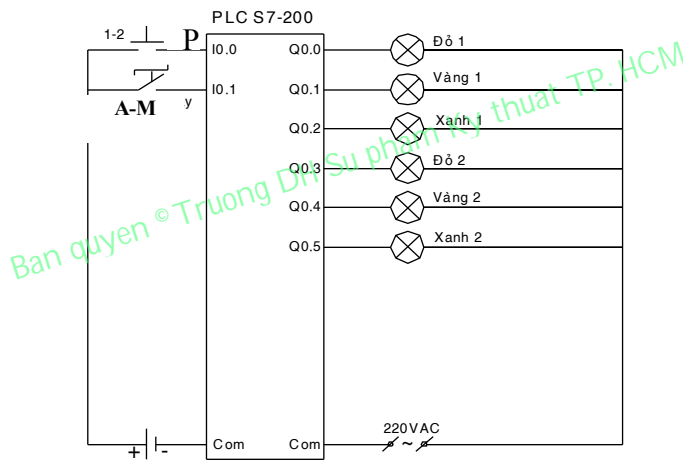
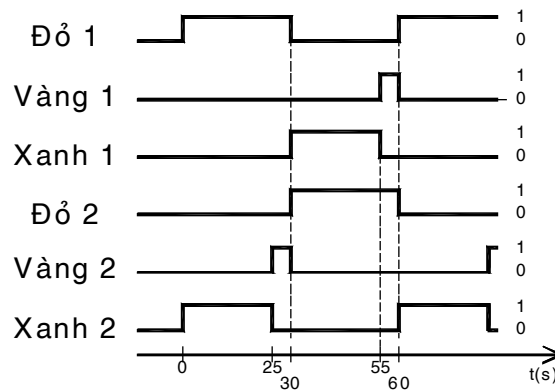
Nhấn nút P lần 1 đèn Đỏ 1 và Xanh 2 tắt, đèn Đỏ 2 và Xanh 1 sáng. Nếu nhấn tiếp nút P thì hoạt động của các đèn trở lại trạng thái ban đầu.

- Ở chế độ tự động:

Chuyển nút AUTO-MAN về phía AUTO thì hoạt động của các đèn được biểu diễn như hình vẽ sau:

ĐIỀU KHIỂN LẬP TRÌNH 1

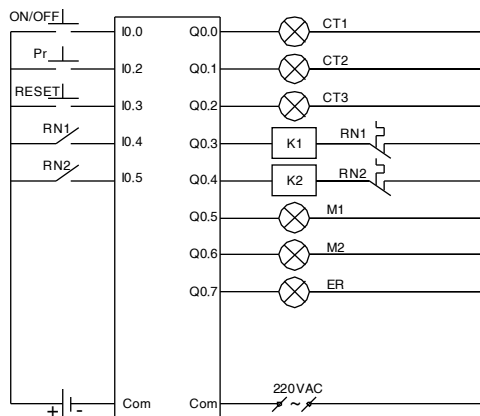
Các đèn hoạt động theo hình trên trong 10 chu kỳ, sau đó đèn Vàng 1 và Vàng 2 nhấp nháy 5 chu kỳ.



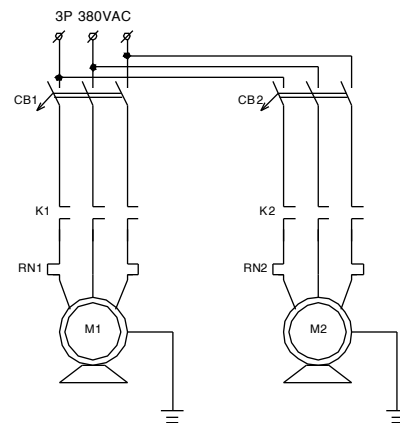
Sơ đồ kết nối PLC

BT 5:

Viết chương trình điều khiển 2 động cơ theo yêu cầu sau:



Sơ đồ kết nối PLC



Sơ đồ Mạch động lực

ĐIỀU KHIỂN LẬP TRÌNH 1

- Đầu tiên nhấn nút Pr chọn chương trình, nếu không nhấn nút chọn chương trình mà nhấn nút ON/OFF đèn Er sẽ nhấp nháy, nhấn nút RESET để phục hồi lại trạng thái cũ.
- Nhấn nút Pr lần 1, đèn CT1 sáng báo hiệu chương trình 1 đã sẵn sàng. Nhấn nút ON điều khiển 2 động cơ chạy theo chương trình 1. Nhấn nút OFF, 2 động cơ dừng.
- Nhấn nút Pr lần 2 đèn CT2 sáng báo hiệu chương trình 2 đã sẵn sàng. Nhấn nút ON điều khiển 2 động cơ chạy theo chương trình 2. Nhấn nút OFF, 2 động cơ dừng.
- Nhấn nút Pr lần 3 đèn CT3 sáng báo hiệu chương trình 3 đã sẵn sàng. Nhấn nút ON điều khiển 2 động cơ chạy theo chương trình 1. Nhấn nút OFF, 2 động cơ dừng.
- Nếu có sự cố xảy ra 2 động cơ sẽ dừng, động cơ nào bị sự cố quá tải thì rờ le của động cơ đó tác động và đèn M của động cơ đó nhấp nháy. Sau khi sự cố được khắc phục đèn M sẽ hết nhấp nháy.
- Đèn M1 và M2 sẽ sáng khi động cơ M1 và M2 chạy và tắt khi động cơ dừng.

Chương trình 1 : K1 và K2 hoạt động khi nhấn ON và OFF

Chương trình 2 : K1 chạy 5s rồi dừng, K2 chạy 10s rồi dừng và lặp lại.

Chương trình 3 : K1 và K2 cùng chạy 5s, dừng 10s, lặp lại theo chu kỳ 5 lần rồi dừng.

BT 6:

Chương trình điều khiển máy trộn bê tông, theo yêu cầu sau:

- Nhấn nút ON, các van V1 mở 5s, V2 mở 10s và V3 mở 15s cho vật liệu vào máy. (V1 chứa nước, V2 chứa xi măng, V3 chứa cát)
- Sau đó động cơ Đ1 hoạt động 10s để trộn.
- Van V4 mở ra và động cơ Đ2 hoạt động bơm bê tông ra ngoài.
- Khi hết bê tông trong máy, sẽ bắt đầu qui trình mới.
- Nhấn nút OFF, dừng quá trình trộn sau khi hết bê tông trong máy.
- Nếu có sự cố thì nhấn nút E để dừng ngay.

BT 7:

Chương trình điều khiển 2 bồn trộn hóa chất theo yêu cầu sau:

- Bảng điều khiển có 5 nút nhấn: ON, OFF, M1, M2, RESET.
- Nếu nhấn nút ON thì cả 2 bồn trộn hoạt động trong 30s.
- Nếu nhấn M1 thì bồn 1 hoạt động 15s, bồn 2 dừng.
- Nếu nhấn M2 thì bồn 2 hoạt động 15s, bồn 1 dừng.
- Khi đang trộn, nếu van của 1 trong 2 bồn bị hở phải báo chuông và dừng trộn ngay. Sau khi khắc phục sự cố, nhấn nút RESET để trả lại trạng thái bình thường.
- Nhấn nút OFF sẽ dừng hoạt động các bồn trộn.

BT 8:

Chương trình đặt trước giá trị cho hệ thống đếm sản phẩm từ 0 đến 999 bằng các nút nhấn, theo yêu cầu:

- Nhấn nút ON để khởi động.
- Chọn giá trị cài đặt bằng các nút nhấn từ 0 đến 9.
- Kết quả cài đặt được hiển thị trên 3 led 7 đoạn.

ĐIỀU KHIỂN LẬP TRÌNH 1

- Cài đặt xong, nhấn nút SET để kết thúc. Hệ thống bắt đầu đếm theo số đặt trước.
- Khi đang đếm, muốn thay đổi giá trị cài đặt thì nhấn nút ADJ.
- Muốn dừng hệ thống, nhấn nút OFF.

BT 9:

Chương trình điều khiển hệ thống sấy sản phẩm theo yêu cầu sau:

- Sau khi đưa sản phẩm vào bồn sấy, nhấn nút ON để khởi động.
- Đặt thời gian sấy bằng các phím từ 0 đến 9.
- Thời gian sấy hiển thị trên 2 led 7 đoạn, tính theo s.
- Đặt thời gian xong, nhấn nút START để máy sấy hoạt động.
- Nếu không đặt thời gian thì khi nhấn START sẽ mặc định thời gian là 10s.
- Sau khi sấy xong, máy sấy tự dừng, đèn báo màu xanh sáng.
- Nếu có sự cố, hay muốn dừng sấy thì nhấn STOP, máy sấy dừng, đèn đỏ sáng.

BT 10:

Chương trình điều khiển động cơ 3 pha theo yêu cầu sau:

- Nhấn nút ON/OFF lần 1 để bắt đầu.
- Chọn số chu kỳ làm việc bằng cách nhấn các phím số 1,2,3,4.
- Muốn chọn lại số chu kỳ làm việc thì nhấn nút RESET và chọn lại các phím số.
- Động cơ đang hoạt động, muốn dừng lại thì nhấn nút ON/OFF lần 2.
- Một chu kỳ hoạt động gồm các bước sau:
 - o Chạy thuận 2s, đèn T sáng.
 - o Dừng 3s, đèn STOP sáng.
 - o Chạy ngược 3s, đèn N sáng.

BT 11:

Chương trình điều khiển động cơ bước hoạt động theo yêu cầu sau:

- Nhấn nút Start đèn xanh sáng cho phép hệ thống hoạt động.
- Nhấn nút F, động cơ quay thuận, đèn F sáng.
- Nhấn nút R, động cơ quay ngược, đèn R sáng.
(Muốn đảo chiều động cơ phải cho động cơ dừng trước)
- Nhấn F1, động cơ quay thuận đúng 1 vòng, đèn F nhấp nháy.
- Nhấn R1, động cơ quay ngược đúng 1 vòng, đèn R nhấp nháy.
- Nhấn Stop thì động cơ dừng, đèn tắt.

BT 12:

Chương trình điều khiển động cơ bước hoạt động theo yêu cầu sau:

- Nhấn Start cho phép hệ thống hoạt động.
- Nhấn nút F thì động cơ quay thuận tự do, đèn F sáng.
- Nhấn nút R thì động cơ quay ngược tự do, đèn R sáng.
- Muốn chọn góc quay động cơ thì phải nhấn các số lựa chọn trên bảng. Trên bảng có các số chọn góc quay: 45, 90, 135, 180, 270, 360, sau đó nhấn F hoặc R.
- Nhấn số 45 thì động cơ quay 45^0 ,... Góc quay động cơ được hiển thị trên 3 led 7 đoạn.
- Nhấn Stop thì động cơ dừng.

ĐIỀU KHIỂN LẬP TRÌNH 1

BT 13:

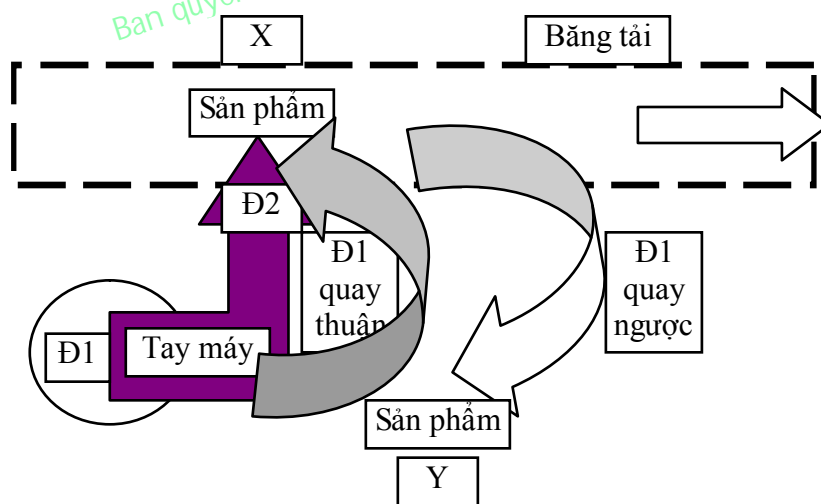
Chương trình điều khiển thang máy hoạt động theo yêu cầu sau:

- Chọn tầng bằng các phím GND, 1, 2 và 3. Có các cảm biến vị trí V0, V1, V2 và V3 để biết thang đang ở tầng nào.
- Nếu chọn phím 1 thì cho động cơ hoạt động kéo thang từ tầng trệt lên tầng 1 và dừng lại, chuông kêu, cửa mở trong vòng 10s rồi đóng lại. Tương tự cho các phím khác.
- Nếu không có phím chọn khác thì động cơ hoạt động đưa thang về tầng trệt chờ.
- Động cơ đi lên = quay thuận. Động cơ đi xuống = quay ngược.

BT 14:

Chương trình điều khiển cánh tay máy và băng tải hoạt động theo yêu cầu sau:

- Nhấn nút Start, động cơ Đ chạy, kéo băng tải chuyển động mang sản phẩm đi chuyển.
- Khi sản phẩm đến vị trí X thì băng tải dừng, động cơ Đ1 quay thuận đưa cánh tay máy đến sản phẩm.
- Khi cánh tay quay đến vị trí X thì động cơ Đ1 dừng, động cơ Đ2 quay thuận gấp sản phẩm trong 2s.
- Sau đó băng tải tiếp tục di chuyển mang sản phẩm mới đến X, đồng thời động cơ Đ1 quay ngược đưa sản phẩm đến điểm Y.
- Khi đến Y, Đ1 dừng, Đ2 quay ngược nhà sản phẩm ra.
- Muốn dừng hệ thống thì nhấn nút Stop.



b

BT 15:

Viết chương trình điều khiển báo chuông tiết học theo yêu cầu:

- Chuông báo từ tiết 1 đến tiết 12 trong ngày,
- Chủ nhật nghỉ.

ĐIỀU KHIỂN LẬP TRÌNH 1

PHỤ LỤC

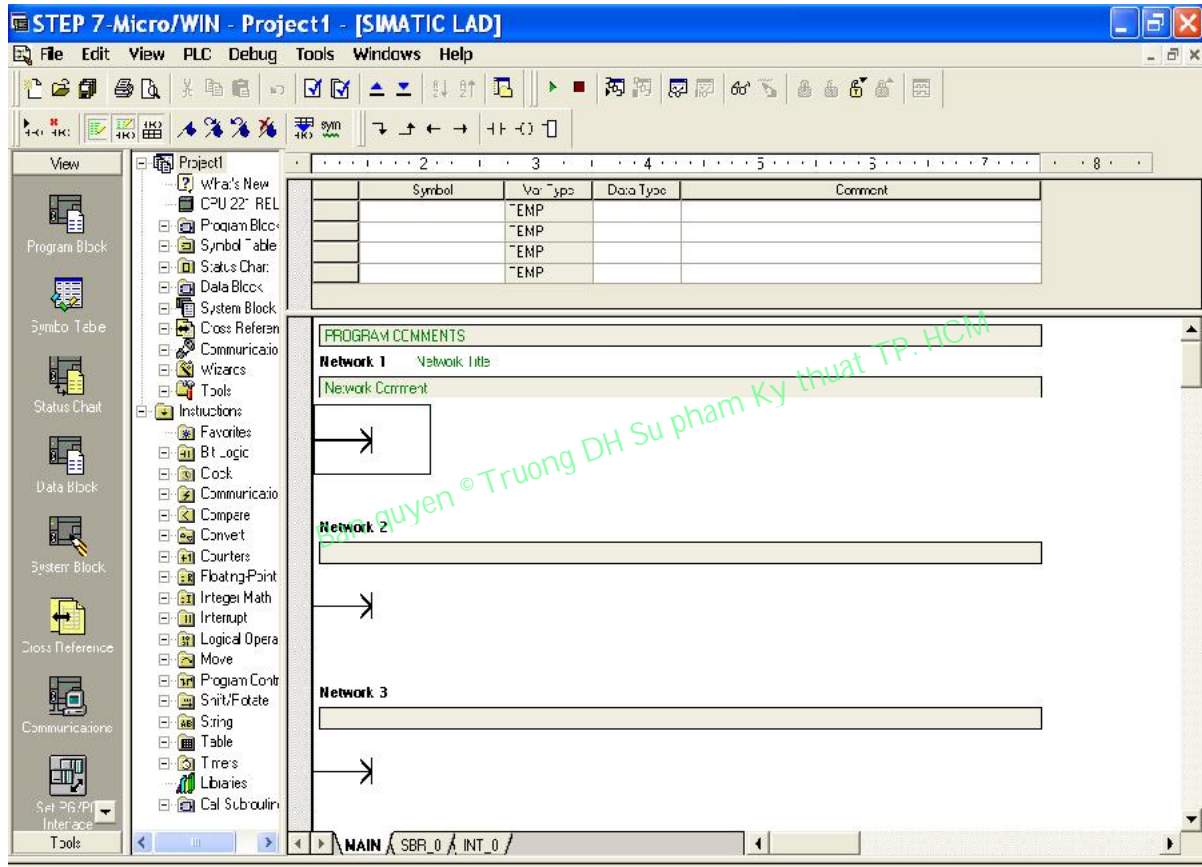
PL1: PHẦN MỀM LẬP TRÌNH Step 7 MicroWIN 3.2/4.0

Step7 MicroWin 3.2 hoặc 4.0 là phần mềm dùng để lập trình cho PLC S7-200.

Khởi động :

Cách 1 : Start, Simatic, Step7-Microwin

Cách 2 : Doubleclick vào biểu tượng Step7-Microwin.



Soạn thảo chương trình

Cách 1 : Chọn Project _ New Hoặc

Cách 2 : Chọn biểu tượng trên cửa sổ chính



Chọn : Read CPU type nếu đã nối giữa máy tính và PLC để phần mềm tự xác lập loại CPU đang giao tiếp.

Sau đó vào màn hình soạn thảo chương trình .

Step7 MicroWIN cho phép chứa nhiều network (tối đa là 100). Mỗi một Network tương đương một câu lệnh, nếu tồn tại 2 câu lệnh trở lên thì chương trình sẽ báo lỗi khi biên dịch.

ĐIỀU KHIỂN LẬP TRÌNH 1

Ta có thể dùng chuột để chọn các biểu tượng mong muốn ở mỗi trong thư viện lệnh và đặt chúng vào các vị trí trong Network.

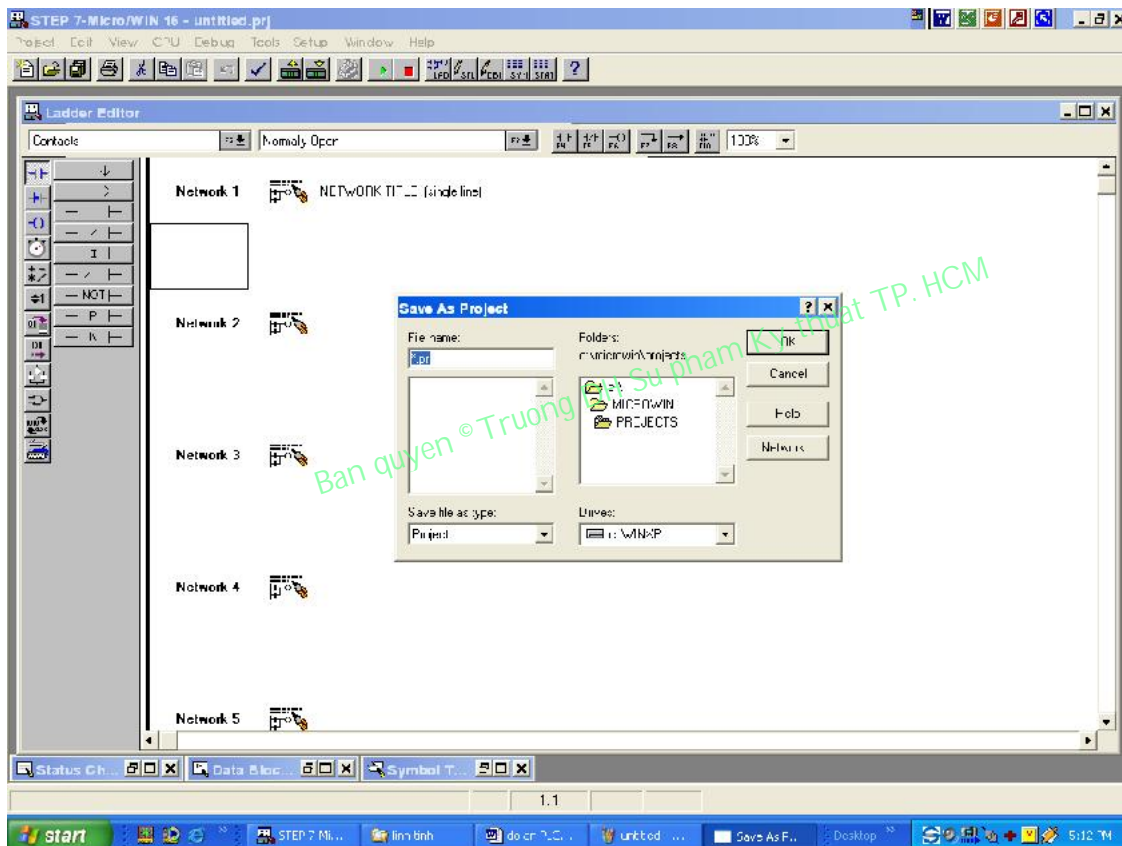
Lưu ý:

Mỗi lệnh phải được gắn trực tiếp vào đường bên trái

Khi con trỏ (hình ô vuông) ở vị trí nào thì khi truy suất thì các toán hạng sẽ đặt tại vị trí đó

Lưu chương trình:

- Lưu project theo tên đã đặt trước : Project _ save all _ tên đặt _ OK
 - Lưu project tên khác : Chọn Project _ save as _ đặt tên _ OK
- Hoặc chọn biểu tượng hình đĩa mềm trên cửa sổ chính _ đặt tên _ OK



Mở chương trình có sẵn:

- Cách 1 : Chọn menu Project _ open _ chọn tên project _ open
- Cách 2 : Chọn biểu tượng open.



Nạp chương trình vào PLC :

- Cách 1 : Chọn Project _ Download _ OK
- Cách 2 : Chọn biểu tượng Download.



ĐIỀU KHIỂN LẬP TRÌNH 1

Lưu ý :

Công tắc chọn chế độ làm việc của PLC phải ở vị trí TERM hoặc ở chế độ STOP
Màn hình báo Download successful thì chương trình đã nạp thành công

Chạy chương trình :

- Cách 1 : Chọn CPU _ RUN _ yes .
- Cách 2: Chọn biểu tượng Run.



Dừng chương trình

- Cách 1 : Chọn menu CPU _ Stop _ Yes
- Cách 2 : Chọn biểu tượng Stop



Lưu ý : Công Tắc chọn chế độ làm việc của PLC phải ở vị trí TERM.

Hiển thị các Chương trình ladder: (quan sát quá trình hoạt động của chương trình)

- Chọn menu : Debug _ ladder Satus on
- Chọn View _ StatusChart

Đọc chương trình của PLC:

Chọn menu Project _ up load _ OK _ Yes
Chọn biểu tượng Upload.



Ok _ Yes

Màn hình hiển thị upload Succesful mới thành công

Xóa hàng , cột , network:

- Cách 1 : Chọn menu Edit _ Delete _ chọn Row Hoặc column hoặc network
- Cách 2 : Nhấn Shift + delete

Chèn hàng , cột , network:

- Cách 1 : Chọn menu Edit _ Inset _ Chọn tiếp Row hoặc column hoặc network
- Cách 2 : Nhấn Shift + Insert

ĐIỀU KHIỂN LẬP TRÌNH 1

PHẦN MỀM MÔ PHỎNG S7-200 Simulator 2.0

Đây là phần mềm dùng để mô phỏng hoạt động của PLC nhằm thay thế PLC thực tế. c
Ta có thể mô phỏng chương trình đã viết bằng cách sử dụng phần mềm này mà không cần đến PLC. Để chạy mô phỏng, ta chỉ cần thực thi file S7-200.exe, gồm các bước sau:

- Viết chương trình bằng phần mềm STEP7
- Biên dịch chương trình: File/Export
- Đặt tên tập tin và chọn Save (*.awl)
- Chạy phần mềm mô phỏng S7_200.exe
- Nhập mã : 6596
- Chọn loại CPU: Configuration/PLC Type
- Mở file cần mô phỏng: Program/Load Program, chọn Accept, chọn file *.awl
- Chạy mô phỏng: Run
- Thay đổi trạng thái các ngõ vào bằng các công tắc trên bảng điều khiển màu xanh.
- Quan sát các đèn báo Input và Output trên PLC

