

TUYÊN BỐ BẢN QUYỀN

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

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

LỜI GIỚI THIỆU

Tự động hóa công nghiệp và dân dụng ngày càng phát triển. Bộ não trong các hệ thống tự động hóa là các bộ điều khiển lập trình. Việc học tập nghiên cứu các bộ điều khiển lập trình cũng như vận hành nó đang là nhu cầu cấp thiết đối với học sinh, sinh viên các ngành kỹ thuật.

Hiện nay tài liệu về giảng dạy lập trình về bộ điều khiển lập trình có rất nhiều tuy nhiên những giáo trình này viết còn khá chung chung, mang nặng tính lý thuyết và chủ yếu dành cho các đối tượng sinh viên đại học. Quyển giáo trình này ra đời với mục tiêu giúp cho các đối tượng học sinh, sinh viên học nghề có thể tiếp cận dễ dàng hơn với bộ điều khiển khả trình này.

Tài liệu được chia làm 4 bài:

Bài 1: Tổng quan về điều khiển

Bài 2: Kết nối PLC S7-200 và phần mềm STEP7 MICROWIN

Bài 3: Các lệnh lập trình cơ bản

Bài 4: Thực hành lắp đặt PLC S7-200

Bài 5: Các chức năng cơ bản của LOGO

Bài 6: Các chức năng đặc biệt của LOGO

Bài 7: Lập trình trực tiếp trên LOGO.

Mỗi bài ngoài phần lý thuyết cơ bản còn bổ sung thêm các ví dụ minh họa và các bài toán điều khiển ngoài thực tế giúp cho học sinh sinh viên nắm rõ hơn về loại PLC này.

Mặc dù đã cố gắng tổ chức biên soạn để đáp ứng được mục tiêu đào tạo nhưng không tránh được những khiếm khuyết. Rất mong nhận được đóng góp ý kiến của các thầy, cô giáo, bạn đọc để nhóm biên soạn sẽ hiệu chỉnh hoàn thiện hơn. Các ý kiến đóng góp xin gửi về Trường Trung cấp nghề Tổng hợp Hà Nội, địa chỉ số 21 Bùi Bằng Đoàn – Q. Hà Đông – TP. Hà Nội

Hà Nội, ngày tháng năm 2023

Nhóm tác giả

MỤC LỤC

TUYÊN BỐ BẢN QUYỀN	1
LỜI GIỚI THIỆU	3
CHƯƠNG TRÌNH MÔ ĐUN ĐÀO TẠO	9
PLC CƠ BẢN.....	9
BÀI 1: TỔNG QUAN VỀ ĐIỀU KHIỂN.....	11
1. Khái niệm chung về điều khiển.....	11
1.1. Khái niệm.....	11
1.2. Cấu trúc một quy trình điều khiển	12
2. Phân loại điều khiển.....	13
3. Bộ điều khiển lập trình PLC	15
3.1. Khái niệm.....	15
3.2. Ứng dụng.....	15
3.3. Phân loại.....	17
BÀI 2: KẾT NỐI PLC S7-200 VÀ PHẦN MỀM STEP7 MICROWIN	19
1. Cấu trúc PLC S7-200	19
1.1. Hình dáng bên ngoài	19
1.2. Cấu trúc phần cứng	21
1.3. Cấu trúc bộ nhớ.....	22
2. Kết nối dây giữa PLC và thiết bị ngoại vi	22
2.1. Kết nối nguồn cung cấp PLC	22
2.2. Kết nối ngõ vào số với ngoại vi	23
2.3. Kết nối ngõ ra số với ngoại vi.....	24
2.4. Giới thiệu kết nối CPU 214, CPU 224 với thiết bị ngoại vi	26
3. Các vùng nhớ và quy ước địa chỉ.....	27
3.1. Phân chia bộ nhớ:.....	27
3.2. Các khái niệm xử lý thông tin	27
3.3. Quy ước địa chỉ.....	28
4. Kết nối PLC S7-200 với máy tính	30
5. Phần mềm STEP7 MCROWIN.....	31
5.1. Giới thiệu phần mềm STEP7 MCROWIN	31
5.2. Các bước thiết lập một dự án mới.....	35
6. Ngôn ngữ lập trình	37
6.1. Dạng hình thang LAD(Ladder logic).....	37

6.2. Dạng khối chức năng: FBD(Function Block Diagram).....	38
6.3. Dạng liệt kê lệnh STL (StaTement List).....	39
BÀI 3: CÁC LỆNH LẬP TRÌNH CƠ BẢN	41
1. Các lệnh xử lý Bit.....	41
1.1. Lệnh tiếp điểm thường mở	41
1.2. Lệnh tiếp điểm thường đóng	42
1.3. Lệnh cuộn dây	44
1.4. Bài toán ứng dụng	45
1.5. Lệnh Set, Reset.....	49
2. Timer	50
2.1. Giới thiệu.....	50
2.2. Timer đóng mạch chậm TON	50
2.3. Timer đóng mạch chậm có nhớ TONR.....	52
3. Counter	53
3.1. Giới thiệu:	53
3.2. Bộ đếm lên CTU	54
3.2. Bộ đếm lên xuống CTUD	56
BÀI 4: THỰC HÀNH LẮP ĐẶT PLC S7-200	59
1. Điều khiển đóng cắt động cơ.....	59
1.1. Yêu cầu công nghệ	59
1.2. Các bước tiến hành.....	59
2. Điều khiển đảo chiều quay trực tiếp động cơ.....	65
2.1. Yêu cầu công nghệ	65
2.2. Các bước tiến hành.....	65
3. Điều khiển đảo chiều quay gián tiếp động cơ	68
3.1. Yêu cầu công nghệ	68
3.2. Các bước tiến hành.....	68
4. Điều khiển đèn giao thông	71
4.1. Yêu cầu công nghệ	71
4.2. Các bước tiến hành.....	72
5. Điều khiển khởi động động cơ chế độ sao – tam giác	75
5.1. Yêu cầu công nghệ	75
5.2. Các bước tiến hành.....	75
BÀI 5: CÁC CHỨC NĂNG CƠ BẢN CỦA LOGO!	79
1. Kết nối Logo với thiết bị ngoại vi	79

1.1. Nối nguồn cung cấp	79
1.2. Kết nối ngõ vào	80
1.3. Kết nối ngõ ra.....	80
2. Các loại hàm trong LOGO!	81
3. Danh sách Co	81
3.1. Ngõ vào số	81
3.2. Ngõ ra số	81
3.3. Ngõ vào analog	81
3.4. Ngõ ra analog	81
3.5. Cờ Start up	82
3.6. Thanh ghi dịch bit	82
3.7. Mức hằng số.....	82
3.8. Không kết nối.....	82
4. Các hàm sử dụng trong Logo!.....	82
4.1. Hàm OR	82
4.2. Hàm AND	83
4.3. Hàm NOT.....	84
4.4. Hàm NAND	85
4.5. Hàm NOR	86
4.6. Hàm XOR	87
BÀI 6: CÁC CHỨC NĂNG ĐẶC BIỆT CỦA LOGO!	89
1. Latching Relay (Relay chốt)	89
2. Counter Up and Down (Bộ đếm lên xuống)	90
3. Timer ON delay.....	91
BÀI 7: LẬP TRÌNH TRỰC TIẾP TRÊN LOGO!.....	93
1. Bốn quy tắc sử dụng phím trên Logo!	93
1.1. Quy tắc 1: thay đổi chế độ hoạt động	93
1.2. Quy tắc 2: các ngõ vào và ngõ ra.....	93
1.3. Quy tắc 3: Con trỏ và di chuyển con trỏ	94
1.4. Quy tắc 4: Lưu trữ chương trình.....	94
2. Cách gọi các chức năng.....	94
2.1. Chế độ lập trình (Programming mode)	94
2.2. Chế độ thiết lập thông số (Parameter assignment mode)	95
3. Phương pháp kết nối các khối chức năng	95
3.1. Biểu diễn các khối trong LOGO!.....	95

3.2. Soạn thảo chương trình	96
3.3. Các thao tác cơ bản	103
4. Lưu trữ vào thẻ nhớ và chạy chương trình.....	104
5. Bài tập ứng dụng	106
5.1. Điều khiển đóng cắt động cơ	106
5.2. Điều khiển đảo chiều quay động cơ.....	109
TÀI LIỆU THAM KHẢO	113

CHƯƠNG TRÌNH MÔ ĐUN ĐÀO TẠO PLC CƠ BẢN

Tên mô đun: PLC Cơ bản

Mã mô đun: MĐ 20

Thời gian thực hiện mô đun: 120 giờ; (Lý thuyết: 36 giờ, Thực hành, thí nghiệm, thảo luận, bài tập: 80 giờ, kiểm tra: 04 giờ)

VỊ TRÍ, TÍNH CHẤT MÔN HỌC:

- Vị trí :

Trước khi học mô đun này cần hoàn thành các môn học cơ sở và các mô đun chuyên môn.

- Tính chất:

Là mô đun chuyên môn nghề, thuộc mô đun đào tạo nghề bắt buộc

MỤC TIÊU MÔN HỌC:

- Kiến thức:

Nắm rõ cấu tạo, hình dáng, thông số kỹ thuật của PLC S7-200 và LOGO.

- Kỹ năng:

+ Sử dụng thành thạo phần mềm Step7 microwin, ngôn ngữ lập trình dạng LAD.

+ Đấu nối thành thạo PLC S7-200 và LOGO với nguồn và các thiết bị ngoại vi.

+ Thiết lập được một dự án nhỏ từ khâu lắp đặt cho đến viết chương trình chạy thử và kiểm tra lỗi .

- Năng lực tự chủ và trách nhiệm:

+ Cẩn thận, bình tĩnh, thực hiện đúng thao tác khi tiếp xúc với điện thế cao.

+ Rèn luyện tính kiên trì, cẩn thận, chính xác và tính kỷ luật cao.

+ Có ý thức bảo quản, giữ gìn sắp xếp đồ dùng học tập gọn gàng, ngăn nắp.

NỘI DUNG MÔN HỌC:

BÀI 1: TỔNG QUAN VỀ ĐIỀU KHIỂN

Mã Bài: MĐ 20 - 01

Giới thiệu:

Trong xu thế hội nhập toàn cầu, đất nước ta đang bước vào giai đoạn thực hiện công nghiệp hóa hiện đại hóa, bất cứ ngành nghề kỹ thuật nào cũng cần đến điều khiển tự động hóa. Kỹ thuật điều khiển đã có một cơ sở nền tảng khoa học vững chắc, đảm bảo cho việc điều khiển một cách nhanh chóng, chính xác đạt hiệu suất cao với các dây chuyền sản xuất phức tạp. Với sự ra đời của các mạch điều khiển điện tử, các cảm biến tự động, thủy lực, khí nén... người ta có đủ cơ sở và công cụ để tăng lên mức tự động hóa của các máy móc công nghiệp. Tuy vậy, với xu thế phát triển của xã hội, một vấn đề đặt ra là nhu cầu của con người thay đổi quá nhanh, nhu cầu sản xuất sản phẩm thay đổi liên tục. Mỗi lần thay đổi sản xuất sản phẩm mới là mỗi lần phải thay đổi lại toàn bộ các máy móc thiết bị, dẫn đến các hệ thống sản xuất dễ bị lỗi thời. Yêu cầu bức thiết đặt ra là làm sao để một dây chuyền có thể sản xuất linh hoạt với nhiều chủng loại sản phẩm khác nhau mà không cần phải thay thế, làm lại các thiết bị máy móc. Với sự ra đời tiếp theo của PLC và máy tính cùng với sự phát triển khoa học điều khiển... hệ thống sản xuất linh hoạt như yêu cầu ở trên đã trở thành hiện thực và trở nên phổ biến.

Mục tiêu:

- Trình bày được khái niệm về điều khiển và phân loại điều khiển.
- Phân biệt điều khiển nối cứng và điều khiển khả trình.
- Kể tên các loại PLC phổ biến trên thị trường
- Rèn luyện đức tính tích cực, chủ động và sáng tạo

Nội dung chính:

1. Khái niệm chung về điều khiển

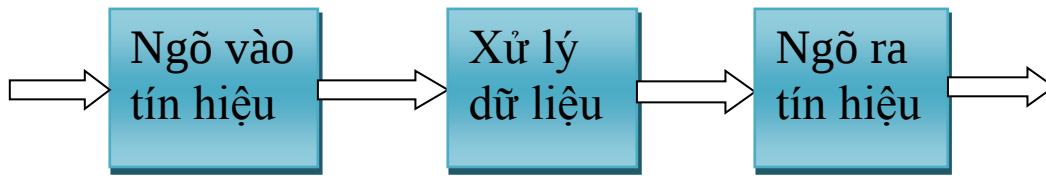
1.1. Khái niệm

Điều khiển có nhiệm vụ thực hiện các chức năng riêng của một máy móc hay một thiết bị theo một trình tự hoạt động định trước phụ thuộc vào trạng thái của máy móc hay các tín hiệu đầu vào.

Ví dụ:

- Điều khiển bật tắt bóng đèn theo thời gian
- Điều khiển khởi động động cơ, đảo chiều quay động cơ.
- Điều khiển chuông báo.
- Các hệ thống điều khiển công nghiệp....

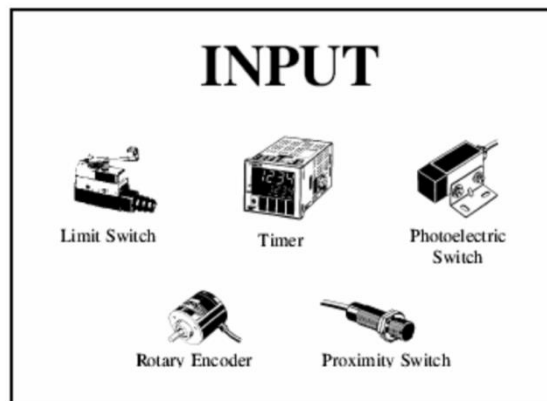
1.2. Cấu trúc một quy trình điều khiển



Hình 1.1. Cấu trúc một quy trình điều khiển

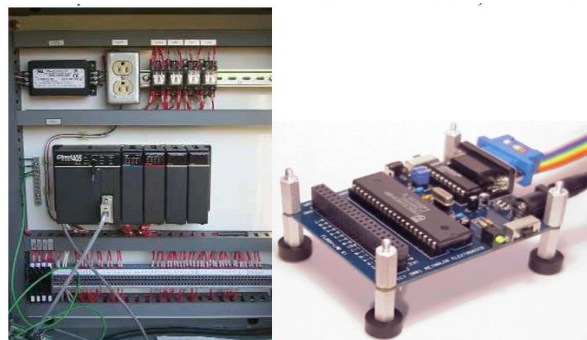
Bao gồm 3 bộ phận hợp thành:

- Ngõ vào tín hiệu: Các tín hiệu vào thường qua bộ chuyển đổi để chuyển đổi các đại lượng vật lý thành các tín hiệu điện. Các bộ chuyển đổi có thể là: Nút nhấn, công tắc, cảm biến... Tùy theo các loại bộ chuyển đổi mà tín hiệu ra có dạng số (Digital) hoặc tương tự (Analog).



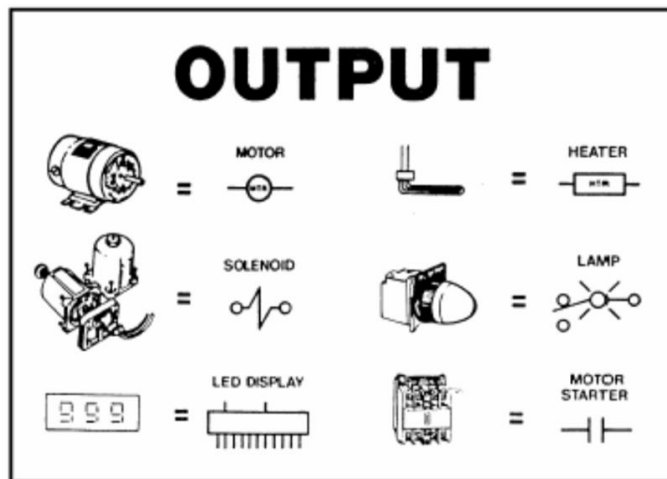
Hình 1.2: Tín hiệu vào

- Xử lý dữ liệu: Là phần chính của tất cả các hệ thống điều khiển, Có thể là các kỹ thuật điều khiển có tiếp điểm như rơ le trung gian, rơ le thời gian, công tắc tơ... hay các các bộ điều khiển lập trình được như PLC, hệ thống máy tính, Vi điều khiển...



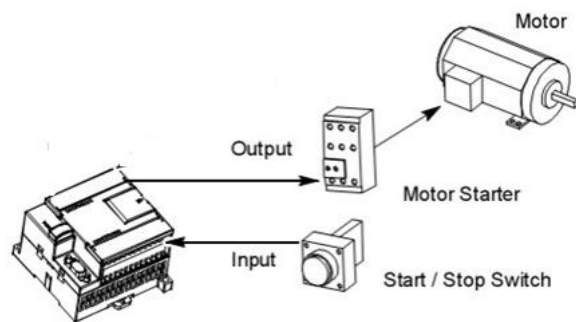
Hình 1.3: Bộ điều khiển

- Ngõ ra tín hiệu: Tín hiệu ra là kết quả của quá trình xử lý của hệ thống điều khiển. Các tín hiệu này được sử dụng để tạo những đáp ứng cụ thể cho các cơ cấu chấp hành như: động cơ, xi lanh, relay,....



Hình 1.4. Tín hiệu ra

- Mô hình điều khiển thực tế



Hình 1.5. Mô hình điều khiển thực tế

2. Phân loại điều khiển

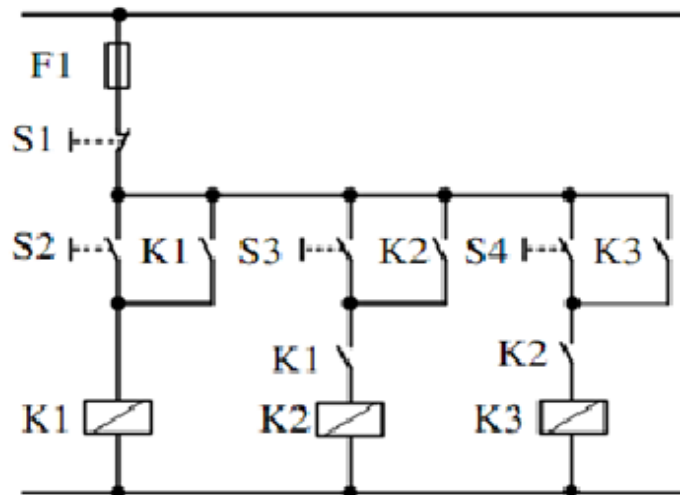
Trong kỹ thuật điều khiển cũng như tự động hóa, người ta chia làm hai loại điều khiển: Điều khiển kết nối cứng và điều khiển khả trình.

- Điều khiển nối cứng là loại điều khiển mà các chức năng của máy móc, thiết bị được điều khiển bằng những role điện từ như các bộ định thời, tiếp điểm, bộ đếm, relay điện từ. Những thiết bị này được liên kết với nhau để trở thành một hệ thống hoàn chỉnh bằng vô số các dây điện bố trí chằng chịt bên trong panel điện (tủ điều khiển).

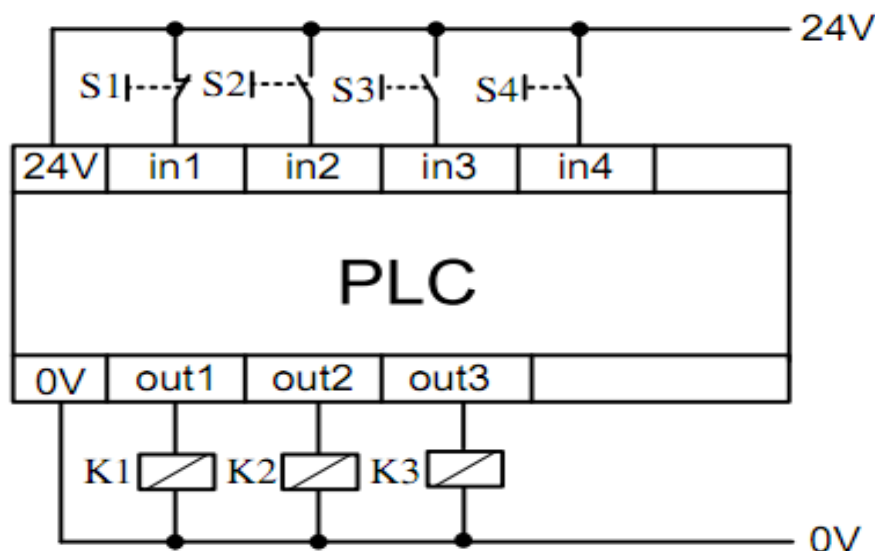
Như vậy, với 1 hệ thống có nhiều trạm làm việc và nhiều tín hiệu vào/ra thì tủ điều khiển rất lớn. Điều đó dẫn đến hệ thống cồng kềnh, sửa chữa khi hư hỏng rất phức tạp và khó khăn. Hơn nữa, các role tiếp điểm nếu có sự thay đổi yêu cầu điều khiển thì bắt buộc thiết kế lại từ đầu.

- Điều khiển khả trình là loại điều khiển mà chức năng của máy móc, thiết bị được điều khiển bằng chương trình lập trình sẵn. Nếu muốn thay đổi chức năng điều khiển của máy móc, thiết bị thì chỉ cần viết lại chương trình điều khiển. Điều khiển khả trình có thể được thực hiện bởi PLC hoặc các hệ thống máy tính...

Ví dụ:



Hình 1.6. Mạch điều khiển trình tự 3 máy bơm sử dụng phương pháp nối cứng.



Hình 1.7. Sơ đồ điều khiển 3 động cơ bơm sử dụng PLC

Qua hai ví dụ đơn giản ta thấy điều khiển khả trình có nhiều ưu điểm so với điều khiển kết nối cứng. Đó là:

- Tiết kiệm các tiếp điểm rơ le trung gian.
- Mạch điều khiển dễ dàng chuyển đổi phù hợp với yêu cầu công nghệ bằng cách lập trình lại chương trình.

- Thi công lắp đặt đơn giản.
- Tính ổn định cao.

Hạn chế điều khiển khả trình so với điều khiển kết nối cứng

- Giá thành cao.
- Bị ảnh hưởng khi môi trường có nhiệt độ cao, rung lắc mạnh.
- Các yêu cầu cố định đơn giản thì không cần PLC.

Ngày nay hệ thống điều khiển khả trình được sử dụng rộng rãi trong các lĩnh vực sản xuất công nghiệp và dân dụng.

Ví dụ:

- Hệ thống điều khiển thang máy
- Hệ thống chiếu sáng tự động
- Hệ thống băng chuyền phân loại sản phẩm
- Hệ thống an ninh tòa nhà
- Các hệ thống sản xuất công nghiệp như bia, xi măng, giấy, bao bì, sữa....

3. Bộ điều khiển lập trình PLC

3.1. Khái niệm

PLC viết tắt của **Programmable Logic Control**, là thiết bị điều khiển Logic lập trình được, hay khả trình, cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua một ngôn ngữ lập trình.

3.2. Ứng dụng

- Trong sản xuất công nghiệp, các hệ thống tự động hóa
- Trong các lĩnh vực cuộc sống như chiếu sáng, thang máy, sản xuất nông nghiệp công nghệ cao
- Trong lĩnh vực an ninh quốc phòng, hàng không, vũ trụ...

Ví dụ:



Hình 1.8. Hệ thống thang máy



Hình 1.9. Băng chuyền sản phẩm



Hình 1.10. Robot hàn



Hình 1.11. Hệ thống rót sản phẩm



Hình 1.12. Băng chuyền phân loại

3.3. Phân loại

Hiện nay trên thị trường có rất nhiều PLC của các hãng khác nhau

- Mỹ: Allen Bradley, General Electric, Square D, Texas Instruments.

- Đức: Siemens, Boost, Festo...

- Hàn Quốc: LG

- Nhật: Mitsubishi, Omron, Panasonic, Fanuc, Mashushita, Fuzi, Koyo...

Và nhiều chủng loại khác, Các sản phẩm như: Logo!, Easy, Zen, ... cũng được chế tạo ra để đáp ứng những yêu cầu điều khiển đơn giản.

Câu hỏi và bài tập

Câu 1: Phân biệt điều khiển nối cứng và điều khiển khả trình, cho ví dụ minh họa.

Câu 2: Kể tên các loại PLC được dùng phổ biến trên thị trường.

BÀI 2: KẾT NỐI PLC S7-200 VÀ PHẦN MỀM STEP7 MICROWIN

Mã Bài: MĐ 20-02

Giới thiệu:

Khi thiết kế một hệ thống điều khiển logic khả trình nói chung hay bộ điều khiển PLC S7-200 nói riêng việc đầu tiên mà một kỹ sư cần làm là thiết kế sơ đồ kết nối phần cứng trong đó sẽ thể hiện

- Nguồn cấp PLC là một chiều, xoay chiều, vị trí các chân cấp nguồn.
- Tín hiệu đầu vào như các nút nhấn, cảm biến... sẽ được quy định địa chỉ vào cụ thể
- Tín hiệu đầu ra như các rơ le, contactor, van điện từ... sẽ được quy định địa chỉ ra cụ thể

Để hoàn thiện một thiết kế cho hệ thống PLC với kết nối phần cứng không là chưa đủ cần có một chương trình điều khiển đi kèm với mỗi thiết kế. Mỗi dòng PLC đi kèm là phần mềm lập trình riêng, PLC S7-200 của hãng Siemens cũng vậy phần mềm lập trình đi kèm là STEP7 MICROWIN đây là công cụ lập trình giúp đưa ý tưởng của người lập trình viên vào trong thiết kế của mình.

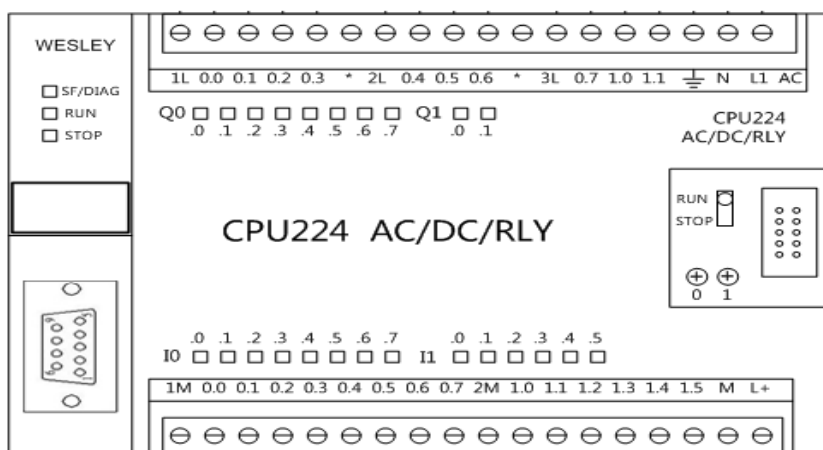
Mục tiêu:

- Nhận biết và đấu nối được nguồn nuôi cho PLC
- Định địa chỉ cho các ngõ vào, ra
- Kết nối PLC với máy tính.
- Thiết lập được dự án mới với step7 microwin.
- Rèn luyện đức tính cẩn thận, tỉ mỉ, tư duy sáng tạo và khoa học.

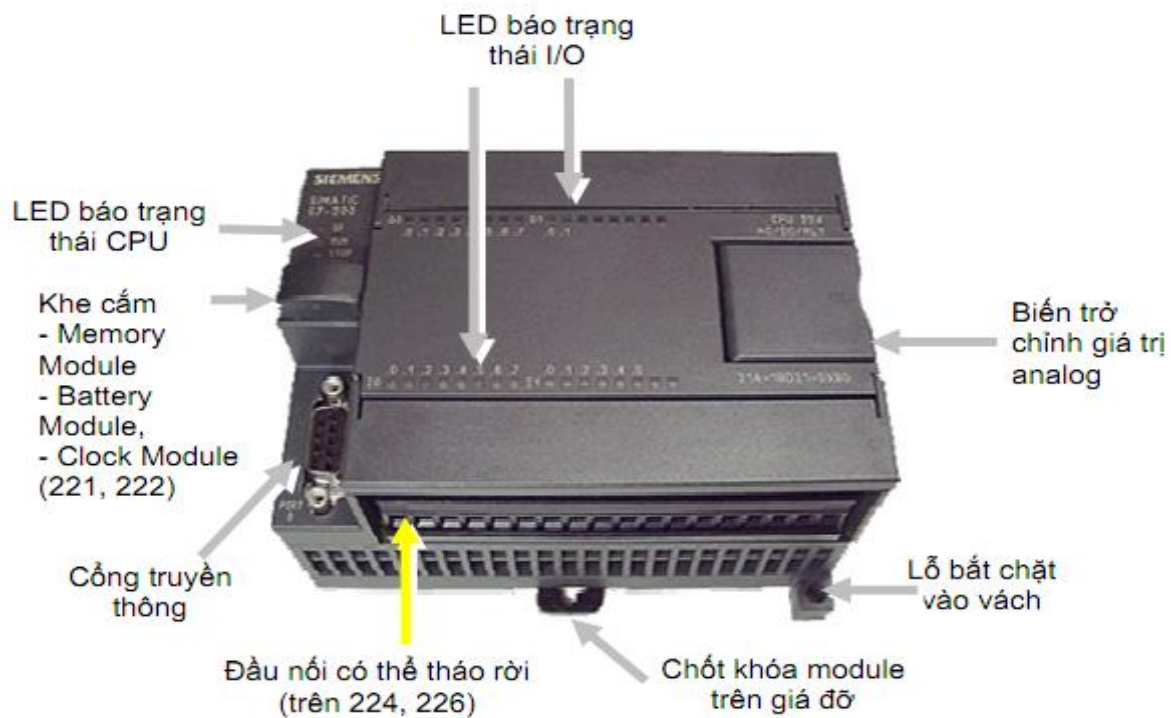
Nội dung chính:

1. Cấu trúc PLC S7-200

1.1. Hình dáng bên ngoài



Hình 2.2. Hình dáng PLC S7-200 CPU 224



Hình 2.1. Hình dáng thực tế PLC S7-200 CPU 224

**Mô tả các đèn báo trên CPU 224*

- SF (Đèn đỏ): báo hiệu hệ thống đang bị lỗi. Đèn SF sáng lên khi hệ thống báo lỗi
- RUN (Đèn xanh): Cho biết PLC đang ở chế độ làm việc thực hiện các lệnh bên trong bộ nhớ chương trình.
- STOP (Đèn vàng): Cho biết PLC đang ở chế độ dừng
- Ix.x (Đèn xanh): Báo hiệu trạng thái hiện thời của cổng vào (x.x=0.0-0.7 và 1.0-1.5)
- Qy.y (Đèn xanh): Báo hiệu trạng thái hiện thời của cổng ra (y.y=0.0-0.7 và 0.0-0.1)

**Chọn chế độ làm việc cho PLC*

RUN: Cho phép PLC thực hiện chương trình nạp sẵn trong bộ nhớ

STOP: Cưỡng bức PLC dừng thực hiện chương trình. Ở chế độ này cho phép hiệu chỉnh, nạp, xóa chương trình.

TERM: Cho phép máy tính chọn chế độ cho run hoặc stop cho PLC

**Cổng truyền thông*

S7200 sử dụng cổng truyền thông nối tiếp RS485 để phục vụ ghép nối với máy tính hoặc với các PLC khác. Để kết nối với máy tính cần cáp nối PC/PPI qua cổng com hoặc PC/MPI qua cổng usb.

**Các khối mở rộng*

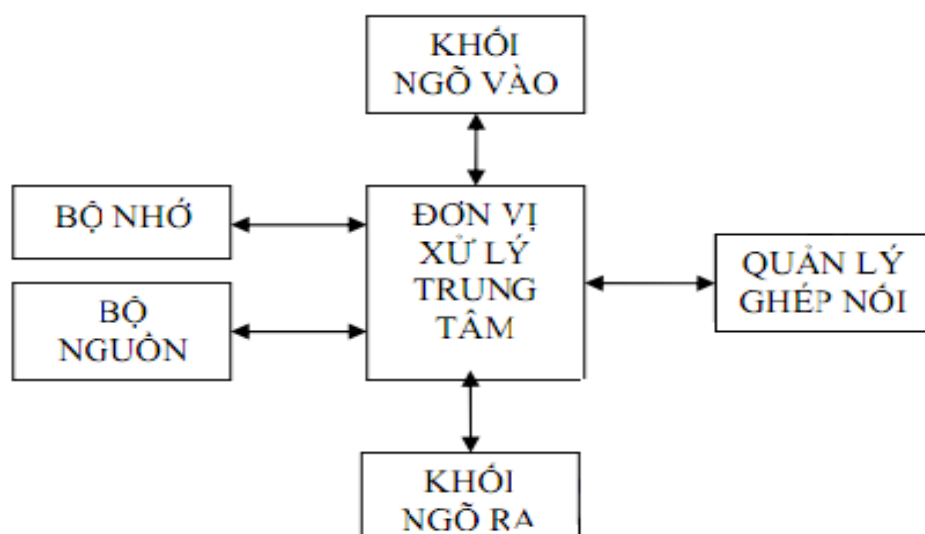
Xuất phát từ yêu cầu điều khiển cần nhiều ngõ vào ra số, sử dụng tín hiệu analog hoặc kết nối truyền thông công nghiệp mà ta phải kết nối PLC với các mô đun mở rộng.

Các khối mở rộng bao gồm:

- Khối vào/ ra số
- Khối vào/ ra analog
- Các khối kết nối mạng truyền thông công nghiệp: AS-interface, Profibus, ethernet...
- Các mô đun chuyên dụng

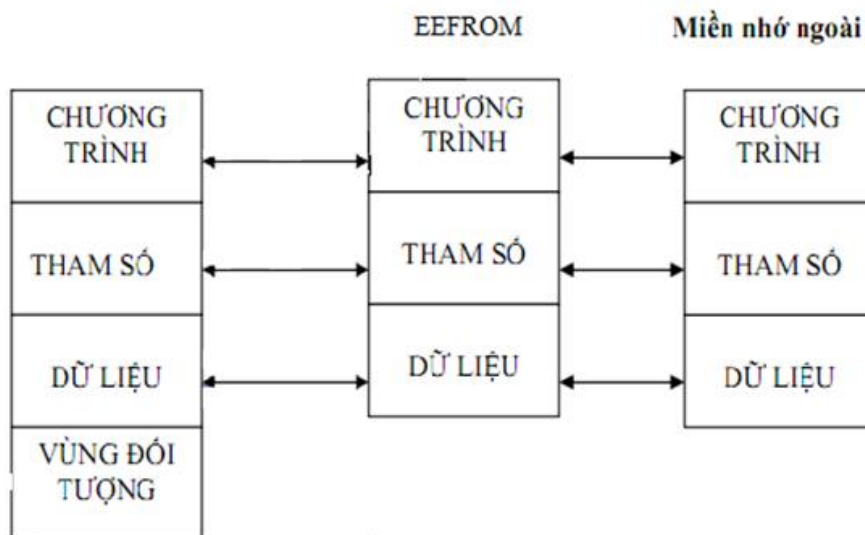
1.2. Cấu trúc phần cứng

- Mô đun nguồn: cấp nguồn ổn định cho PLC hoạt động
- Mô đun đầu vào: nhận các tín hiệu vào
- Mô đun đầu ra: xuất các tín hiệu điều khiển
- Mô đun đơn vị xử lý trung tâm CPU: thực hiện xử lý thông tin dữ liệu
- Mô đun bộ nhớ: nơi lưu trữ thông tin và dữ liệu của CPU
- Mô đun quản lý ghép nối: dùng để ghép nối PLC với máy tính và các thiết bị lập trình, mạng truyền thông công nghiệp.



Hình 2.3. Cấu trúc phần cứng PLC

1.3. Cấu trúc bộ nhớ

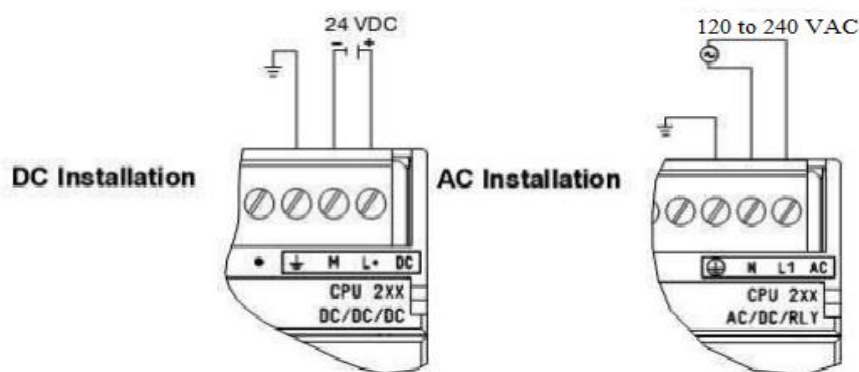


Hình 2.4. Cấu trúc bộ nhớ PLC

- Vùng nhớ dữ liệu
- Vùng nhớ chương trình
- Vùng nhớ đối tượng
- Vùng nhớ tham số

2. Kết nối dây giữa PLC và thiết bị ngoại vi

2.1. Kết nối nguồn cung cấp PLC



Hình 2.5 Sơ đồ đấu nối nguồn cho PLC

Trước khi sử dụng ta phải quan tâm xem mã PLC mà mình đang sử dụng căn cứ vào các chữ số đi kèm theo CPU. Ví dụ với mã số kèm theo CPU 2xx có thể có như sau:

- CPU 2xx DC/DC/DC: Nguồn cấp cho CPU là DC, nguồn cấp cho ngõ vào là DC, nguồn cấp cho ngõ ra là DC.
- CPU 2xx AC/DC/RLY: Nguồn cấp cho CPU là AC, nguồn cấp cho ngõ vào là DC, ngõ ra là rơ le có thể cấp nguồn DC hoặc AC.

Nguồn cấp cho PLC có thể là nguồn một chiều hoặc nguồn xoay chiều

- Nguồn 1 chiều 24V và kí hiệu là M, L+
- Nguồn xoay chiều 120 đến 240V và là N, L1.

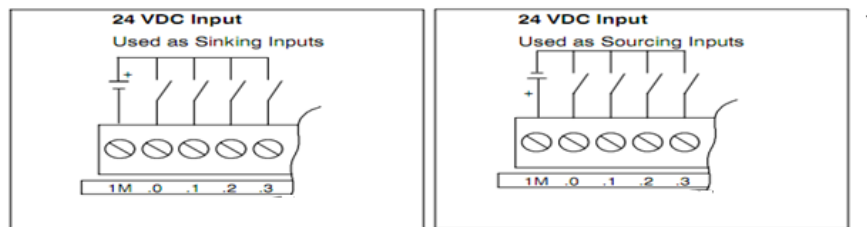
Order Number	CPU Model	Power Supply (Nominal)	Digital Inputs	Digital Outputs	Comm Ports	Analog Inputs	Analog Outputs	Removable Connector
6ES7 211-0AA23-0XB0	CPU 221	24 VDC	6 x 24 VDC	4 x 24 VDC	1	No	No	No
6ES7 211-0BA23-0XB0	CPU 221	120 to 240 VAC	6 x 24 VDC	4 x Relay	1	No	No	No
6ES7 212-1AB23-0XB0	CPU 222	24 VDC	8 x 24 VDC	6 x 24 VDC	1	No	No	No
6ES7 212-1BB23-0XB0	CPU 222	120 to 240 VAC	8 x 24 VDC	6 x Relay	1	No	No	No
6ES7 214-1AD23-0XB0	CPU 224	24 VDC	14 x 24 VDC	10 x 24 VDC	1	No	No	Yes
6ES7 214-1BD23-0XB0	CPU 224	120 to 240 VAC	14 x 24 VDC	10 x Relay	1	No	No	Yes
6ES7 214-2AD23-0XB0	CPU 224XP	24 VDC	14 x 24 VDC	10 x 24 VDC	2	2	1	Yes
6ES7 214-2AS23-0XB0	CPU 224XPSi	24 VDC	14 x 24 VDC	10 x 24 VDC	2	2	1	Yes
6ES7 214-2BD23-0XB0	CPU 224XP	120 to 240 VAC	14 x 24 VDC	10 x Relay	2	2	1	Yes
6ES7 216-2AD23-0XB0	CPU 226	24 VDC	24 x 24 VDC	16 x 24 VDC	2	No	No	Yes
6ES7 216-2BD23-0XB0	CPU 226	120 to 240 VAC	24 x 24 VDC	16 x Relay	2	No	No	Yes

Hình 2.6 Điện áp nguồn cấp cho S7-200

Tuy nhiên điện áp nguồn cấp một chiều có thể dao động từ 20.4-28.8 VDC và điện áp nguồn cấp xoay chiều dao động từ 85-265 VAC ở tần số 47-63 Hz thì PLC vẫn hoạt động bình thường được.

2.2. Kết nối ngõ vào số với ngoại vi

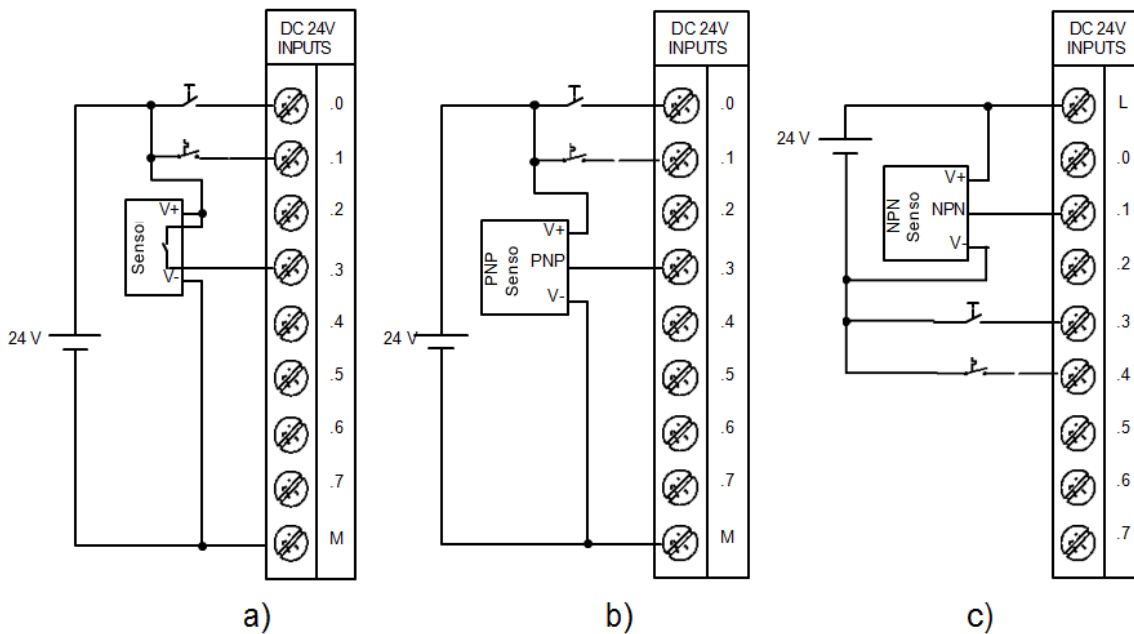
- Sơ đồ đấu nối các loại ngõ vào PLC



Hình 2.7 Sơ đồ đấu nguồn ngõ vào

Nguồn ngõ vào là nguồn một chiều danh định 24VDC tuy nhiên có thể sử dụng dao động từ 15-30 VDC tùy theo tiếp điểm

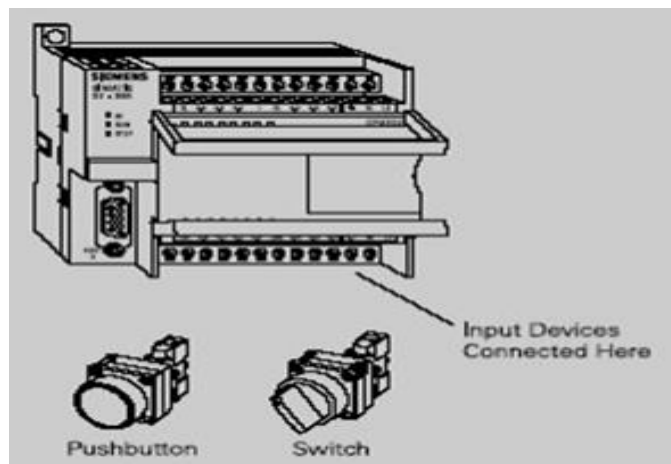
Trong ví dụ hình 5.7a có 3 ngõ vào, một là nút nhấn thường hở, hai là tiếp điểm của relay nhiệt, và ba là cảm biến tiếp cận với ngõ ra là relay. Cả ba bộ tạo tín hiệu này được cung cấp bởi một nguồn 24VDC. Khi tiếp điểm hở hoặc cảm biến phát tín hiệu “0” thì không có điện áp tại các ngõ vào. Nếu các tiếp điểm được đóng lại hoặc cảm biến phát tín hiệu “1” thì ngõ vào được cấp điện.



Hình 2.8. Kết nối ngõ vào với ngoại vi.

- Nút nhấn và cảm biến có ngõ ra là relay nối với ngõ vào loại sinking.
- Nút nhấn và cảm biến loại PNP nối với ngõ vào loại sinking.
- Nút nhấn và cảm biến loại NPN nối với ngõ vào loại sourcing.

Giả sử cần kết nối 1 công tắc, hoặc 1 nút nhấn cho ngõ vào PLC s7-200 CPU 224 AC/DC/RLY



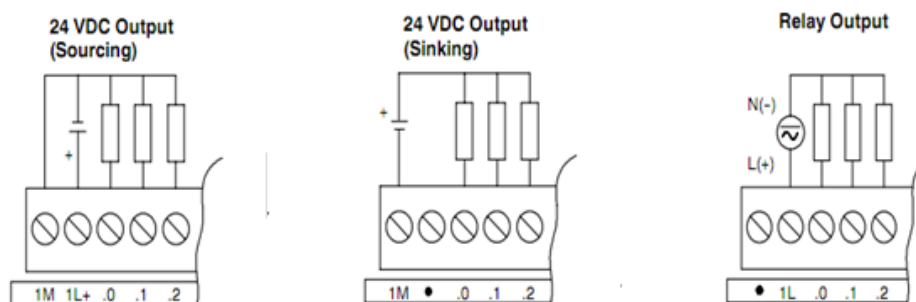
Hình 2.9. Kết nối nút nhấn, công tắc với PLC

Chân 1M, 2M nối nguồn âm 24vdc (GND).

Nguồn dương 24vdc nối vào 1 đầu của nút nhấn hay công tắc, đầu còn lại của tiếp điểm nối vào các ngõ vào I trên PLC.

2.3. Kết nối ngõ ra số với ngoại vi

- Sơ đồ đấu nối các loại ngõ ra PLC



Hình 2.10. Sơ đồ đầu nguồn ngõ ra

Nguồn ngõ ra có thể là:

- 24V một chiều đối với ngõ ra là MOSFET
- 24V một chiều hoặc 250V xoay chiều đối với ngõ ra rơ le

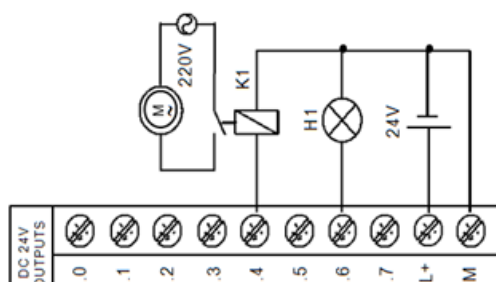
General	24 VDC Output (CPU 221, CPU 222, CPU 224, CPU 226)	24 VDC Output (CPU 224XP)	24 VDC Output (CPU 224XPSi)	Relay Output
Type	Solid State-MOSFET (Sourcing)		Solid State-MOSFET (Sinking)	Dry contact
Rated voltage	24 VDC	24 VDC	24 VDC	24 VDC or 250 VAC
Voltage range	20.4 to 28.8 VDC	5 to 28.8 VDC (Q0.0 to Q0.4) 20.4 to 28.8 VDC (Q0.5 to Q1.1)	5 to 28.8 VDC	5 to 30 VDC or 5 to 250 VAC
Surge current (max.)	8 A for 100 ms			5 A for 4 s @ 10% duty cycle
Logic 1 (min.)	20 VDC at maximum current	L+ minus 0.4 V at max. current	External Voltage Rail minus 0.4V with 10K pullup to External Voltage Rail	-

Hình 2.11. Chi tiết điện áp cấp ngõ vào

- Ngõ vào có thể đấu với nút nhấn, công tắc, công tắc hành trình, cảm biến, tiếp điểm rơ le nhiệt...

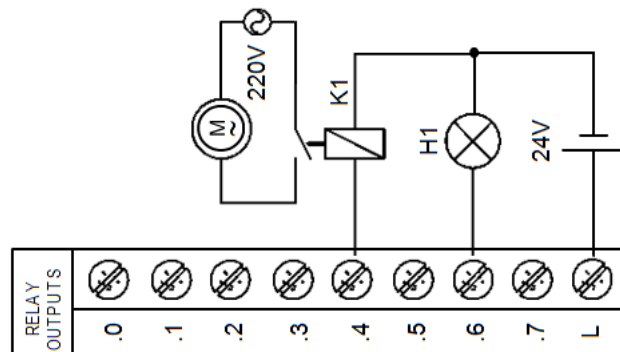
- Ngõ ra có thể đấu với rơ le trung gian, rơ le điều khiển công tắc tơ, đèn...

Trong hình 2.11 kết nối ngõ ra Transistor cho đèn báo, relay sử dụng nguồn 24Vdc



Hình 2.12a: Kết nối dây ngõ ra Transistor với cơ cấu chấp hành đèn báo,

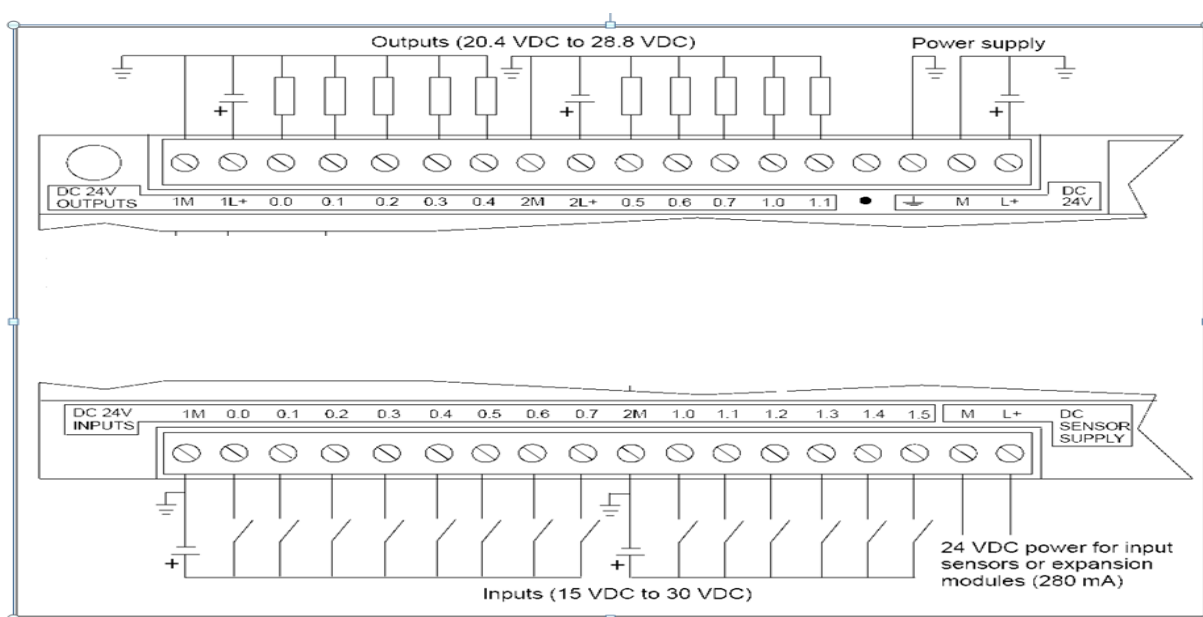
Trong hình 2.12 kết nối ngõ ra relay cho đèn báo, relay sử dụng nguồn 24Vdc



Hình 2.12b. Kết nối dây ngõ ra rơ le với cơ cấu chấp hành đèn báo,

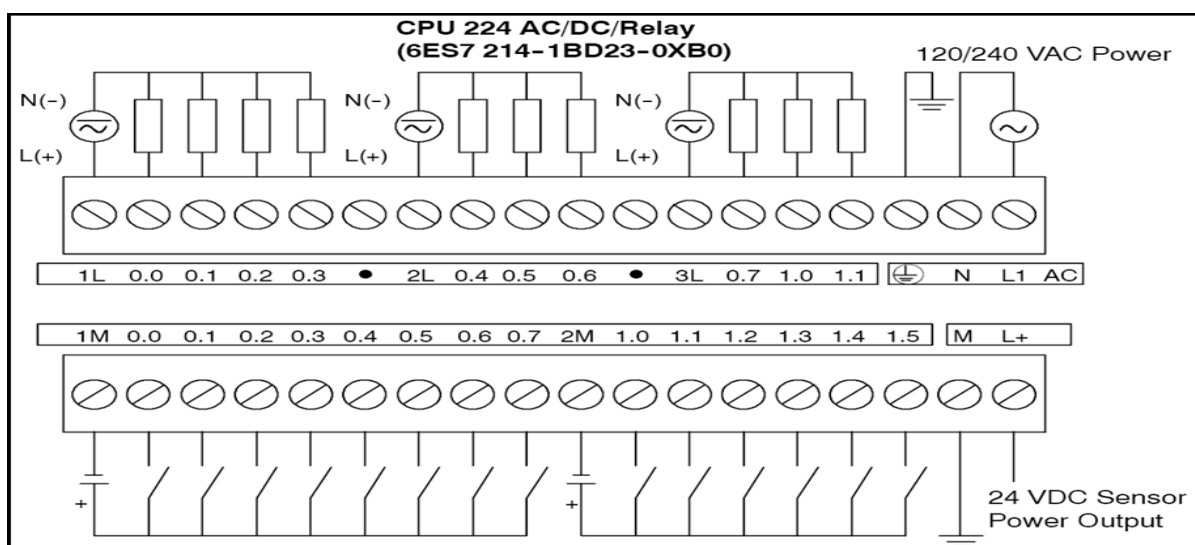
2.4. Giới thiệu kết nối CPU 214, CPU 224 với thiết bị ngoại vi

- Sơ đồ nối dây CPU 214



Hình 2.13: Sơ đồ nối dây CPU 214 DC/DC/DC với nguồn và ngoại vi

- Sơ đồ nối dây CPU 224



Hình 2.14. Sơ đồ nối dây CPU 224 AC/DC/Relay với nguồn và ngoại vi

3. Các vùng nhớ và quy ước địa chỉ

3.1. Phân chia bộ nhớ:

Bộ nhớ S7 – 200 chia làm 4 vùng nhớ:

- Vùng chương trình có dung lượng 4 Kwords được sử dụng để lưu giữ các lệnh chương trình.
- Vùng tham số: miền lưu giữ các từ khóa, địa chỉ trạm.
- Vùng dữ liệu: lưu giữ dữ liệu chương trình: kết quả phép tính, hằng số được định nghĩa trong chương trình. Là 1 vùng nhớ động. Nó có thể truy nhập theo từng bit, byte, word hoặc double word.
- Miền V (Variable): V0.x – V5119.7 (x = 0 - 7)
- Vùng đếm cổng vào (I): I 0.x → I 15.x (x = 0 - 7)
- Vùng đếm cổng ra (Q): Q 0.x → Q15.x (x = 0 - 7)
- Vùng nhớ nội (M): M 0.x → M 31.x (x = 0 - 7)
- Vùng nhớ đặc biệt(SM): SM 0.x → SM 179.x (x = 0 – 7)

Vùng đối tượng:

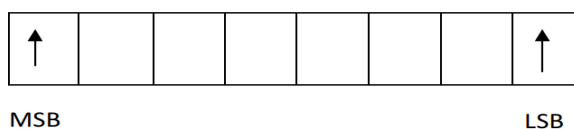
- Timer: T0 → T255
- Counter: C0 → C255
- Bộ đếm cổng vào tương tự: AIW 0 – AIW 30
- Bộ đếm cổng ra tương tự: AQW 0 – AQW 30
- Thanh ghi (Accumulator): AC 0, AC1, AC2, AC3
- Bộ đếm tốc độ cao: HSC0 → HSC5

3.2. Các khái niệm xử lý thông tin

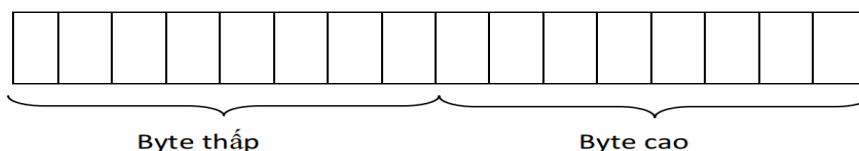
Trong PLC, hầu hết các khái niệm xử lý thông tin cũng như dữ liệu đều được sử dụng như: Bit, Byte, Word, Double Word.

Bit: là 1 ô nhớ có giá trị logic là 0 hoặc 1.

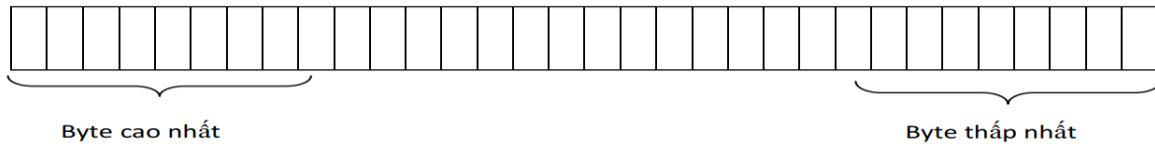
Byte gồm 8 bit



Word(từ đơn): 1 từ gồm có 2 byte.



Double word: gồm có 4 byte



$$1 \text{ Kb(Kílobyte)} = 2^{10} \text{ bits}$$

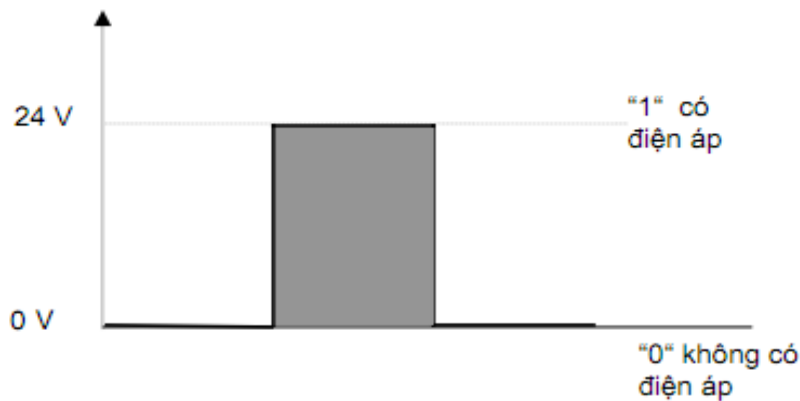
$$1 \text{ Mb(Megabyte)} = 2^{20} \text{ bits}$$

$$1 \text{ Gb(Gigabyte)} = 2^{30} \text{ bits}$$

3.3. Quy ước địa chỉ

3.3.1. Truy xuất theo bit

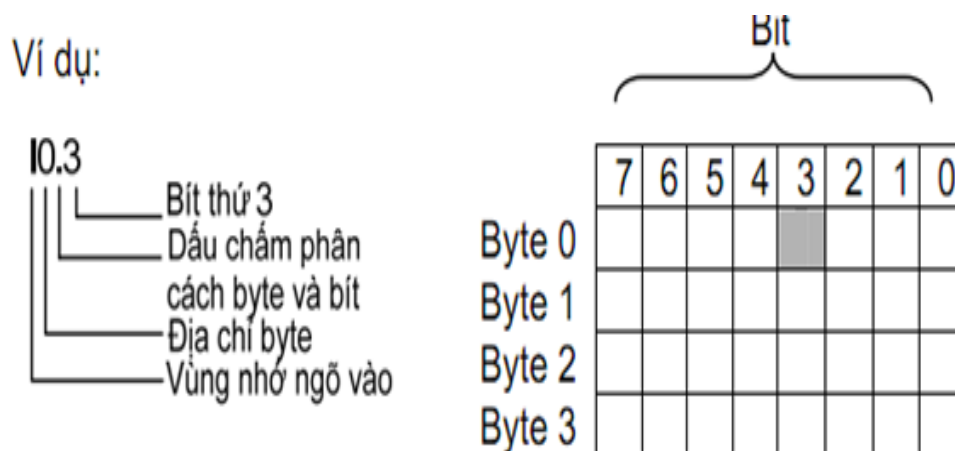
- Bit là đơn vị thông tin nhị phân nhỏ nhất có thể có 2 giá trị là 0 hoặc 1.



Hình 2.15. Giá trị có thể có của một bit

- Để truy xuất địa chỉ theo dạng bit ta xác định vùng nhớ, địa chỉ của byte và địa chỉ bit.

Tên miền + địa chỉ byte + . + chỉ số bit



Hình 2.16. Truy xuất theo bit

Ví dụ: I0.0, Q0.2, M0.1...

3.3.2 Truy xuất theo byte(8 bit)

- Khi truy xuất theo byte ta xác định vùng nhớ, thứ tự byte cần truy xuất

Tên miền + B + địa chỉ byte

Ví dụ:



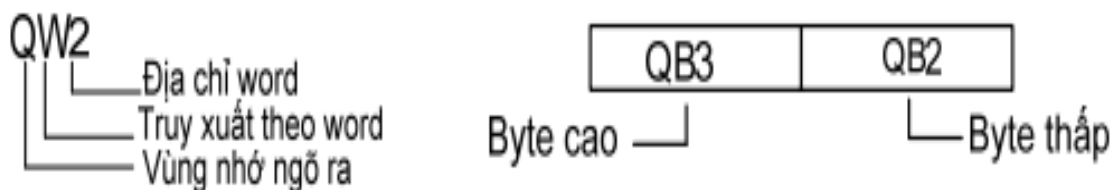
Hình 2.17. Truy xuất theo byte

Ví dụ: QB0, MB1, VB2....

3.3.3 Truy xuất theo word (16 bit)

- Đối với truy xuất vùng nhớ theo dạng word chúng ta cần xác định vùng nhớ cần truy xuất, khai báo dạng word và địa chỉ của word trong vùng nhớ. Mỗi word gồm 2 byte là byte cao và byte thấp.

Tên miền + W + địa chỉ byte cao của word trong miền



Hình 2.18. Truy xuất theo word

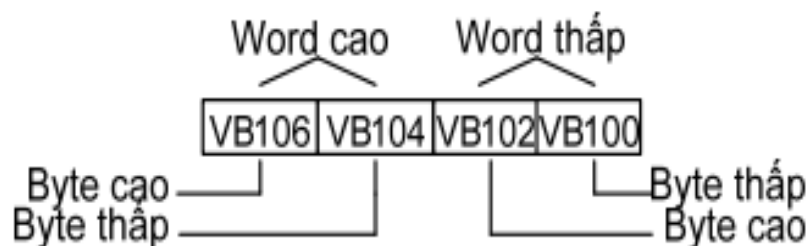
Ví dụ: MW0, QW1, IW2...

3.3.4 Truy xuất theo double word (32 bit)

Tên miền + D + địa chỉ byte cao nhất của một double word trong miền.

Khi truy xuất theo double word tương ứng ta sẽ có 2 word hoặc 4 byte

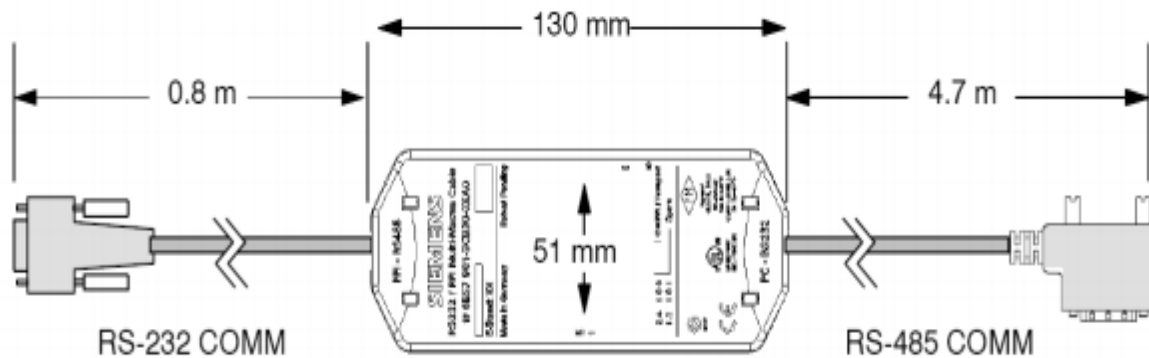
Ví dụ: VD100



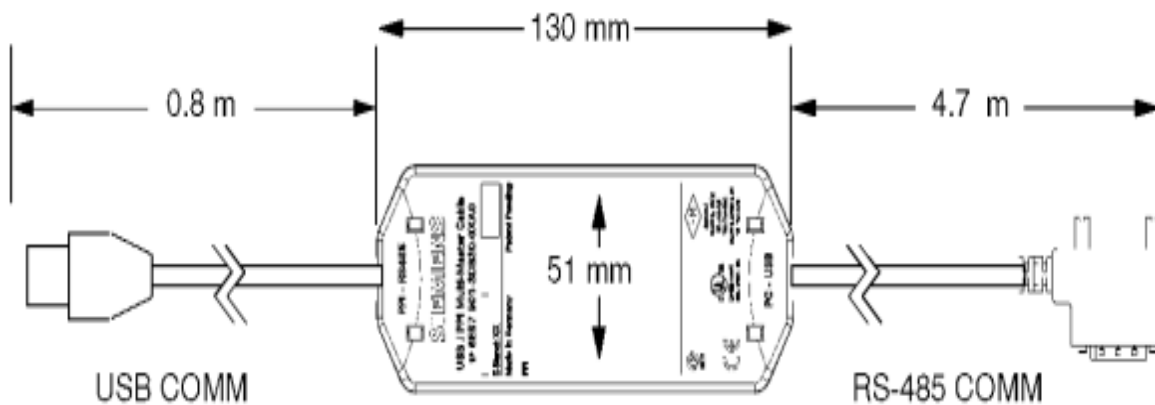
Hình 2.19 Truy xuất theo double word

4. Kết nối PLC S7-200 với máy tính

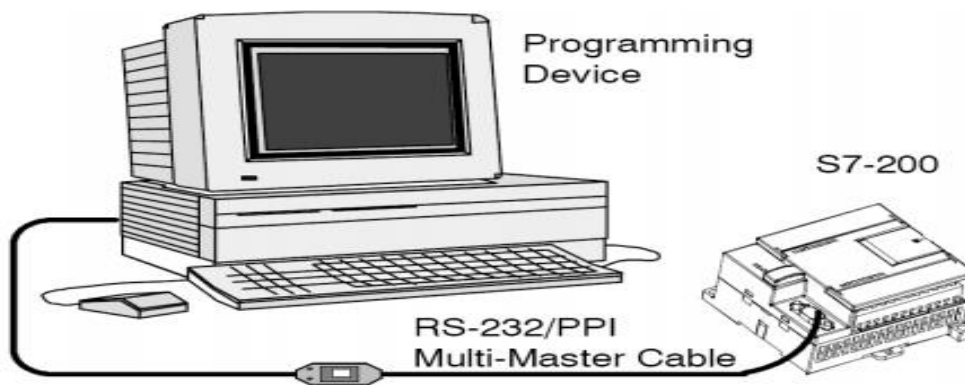
S7-200 sử dụng cổng giao tiếp PPI chuẩn RS485 vì vậy để kết nối với máy tính ta cần cáp chuyển đổi. Có hai loại là RS-232/PPI Multi-Master và USB/PPI Multi-master



Hình 2.20. Cáp RS-232/PPI Multi-Master



Hình 2.21. Cáp USB/PPI Multi-Master



Hình 2.22. Kết nối PLC S7-200 với máy tính

5. Phần mềm STEP7 MCROWIN

5.1. Giới thiệu phần mềm STEP7 MCROWIN

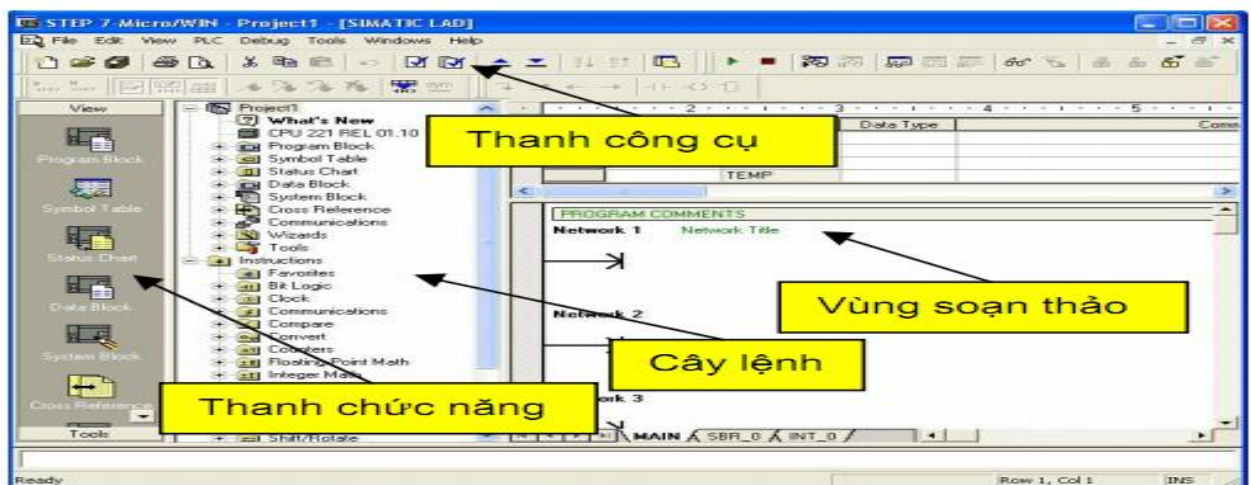
5.1.1. Cài đặt

Thực hiện các bước sau

- Đóng tất cả các ứng dụng
- Chèn đĩa CD step7 microwin vào ổ đĩa. Chương trình có thể được tự động cài đặt hoặc ta có thể khởi động chương trình cài đặt bằng cách kích đúp vào file “Setup.exe” trong đĩa CD
- Sau đó ta tiếp tục làm theo các chỉ dẫn cài đặt trên màn hình và hoàn thành công việc cài đặt.
- Khi cài đặt xong hộp hội thoại “Set PG/PC interface” tự động xuất hiện. Ta nhấn nút “Cancel” để thoát
- Khởi động lại máy tính để hoàn tất việc cài đặt.

5.1.2. Làm quen với giao diện phần mềm

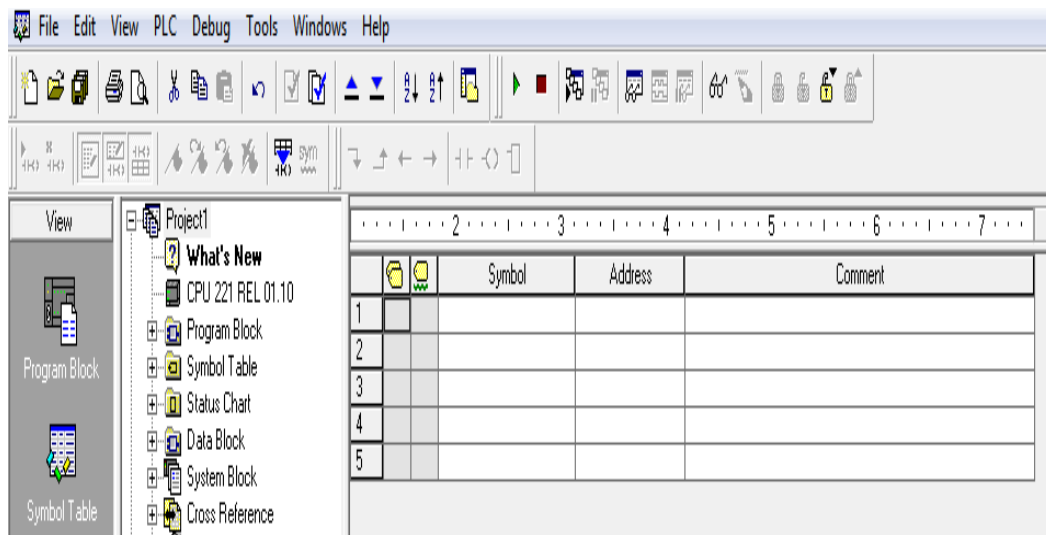
Kích đúp vào biểu tượng  để vào màn hình soạn thảo chính xuất hiện cửa sổ



Hình 2.23. Giao diện chính

Giao diện chính bao gồm:

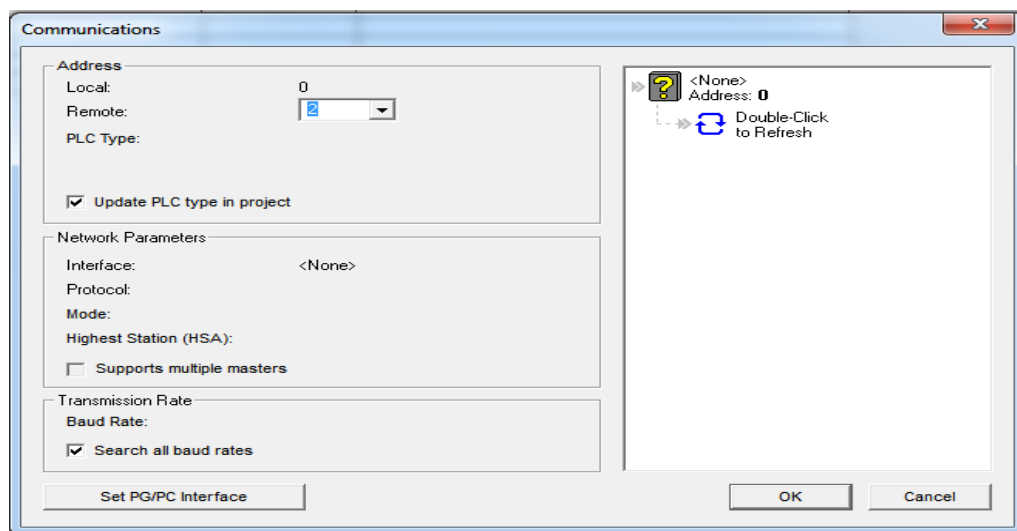
- Thanh chức năng gồm nhiều khối: Program block, Symbol table, Status chart, Data block, System block, Cross reference, Communications, Set PG/PC interface. Tuy nhiên ta chỉ quan tâm tới hai khối chính là Program block và Symbol table.
- + Program block: chính là màn hình soạn thảo chính.
- + Symbol table: dùng để khai báo địa chỉ vào ra, chú thích



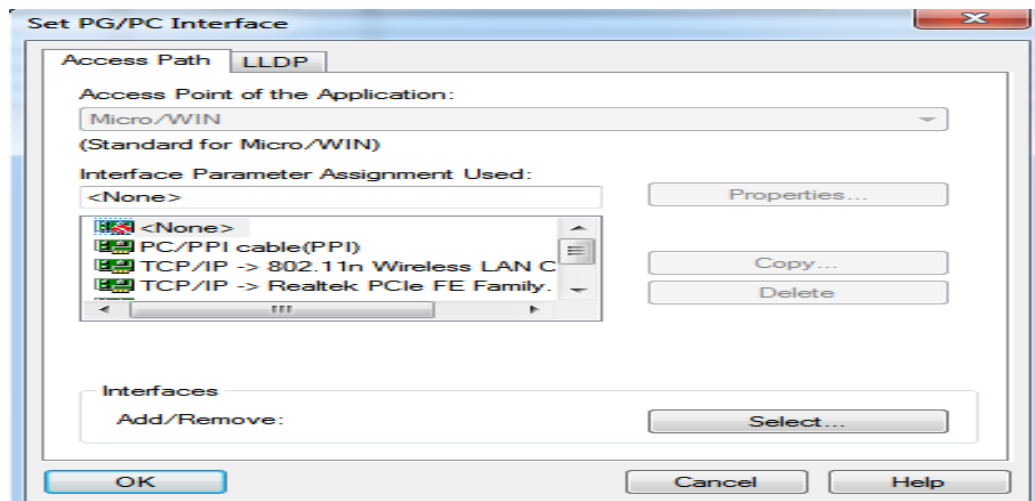
Hình 2.24. Giao diện khối Symbol table

Ngoài ra khi kết nối với PLC ta cần quan tâm thêm hai khối

- + Communications
- + Set PG/PC interface



Hình 2.25. Giao diện khối communications



Hình 2.26. Giao diện khối set PG/PC interface

Hai khối này cho phép ta cài đặt giao tiếp PLC với máy tính như: cổng giao tiếp, địa chỉ CPU, tốc độ truyền...

- Thanh công cụ: Gồm các chức năng hỗ trợ người lập trình để dàng sử dụng :



New Project (File menu): Khởi động một dự án mới



Open Project (File menu): Mở một dự án tồn tại



Save Project (File menu): Lưu dự án



Print (File menu): In chương trình và tài liệu dự án



Print Preview (File menu): Xem trước khi in



Cut (Edit menu): Cắt phần chọn và đưa vào clipboard



Copy (Edit menu): Copy phần được chọn vào clipboard



Paste (Edit menu): Dán nội dung clipboard vào cửa sổ được kích hoạt



Undo (Edit menu): Khôi phục lại phần bị xóa trước



Compile (PLC menu): Biên dịch cửa sổ được kích hoạt (Program Block hoặc Data Block).



Compile All (PLC menu): Biên dịch tất cả các phần tử dự án (Program Block, Data Block, and System Block)



Upload (File menu): Lấy (Upload) các phần tử dự án từ PLC vào màn hình soạn thảo chương trình



Download (File menu): Nạp (download) các phần tử dự án từ STEP7-MicroWin vào PLC.



Option (Tools menu): Truy cập menu Options



RUN (PLC menu): Đặt PLC ở chế độ RUN



STOP (PLC menu): Đặt PLC ở chế độ STOP



Program Status (Debug menu): ON/OFF trạng thái chương trình trong PLC.

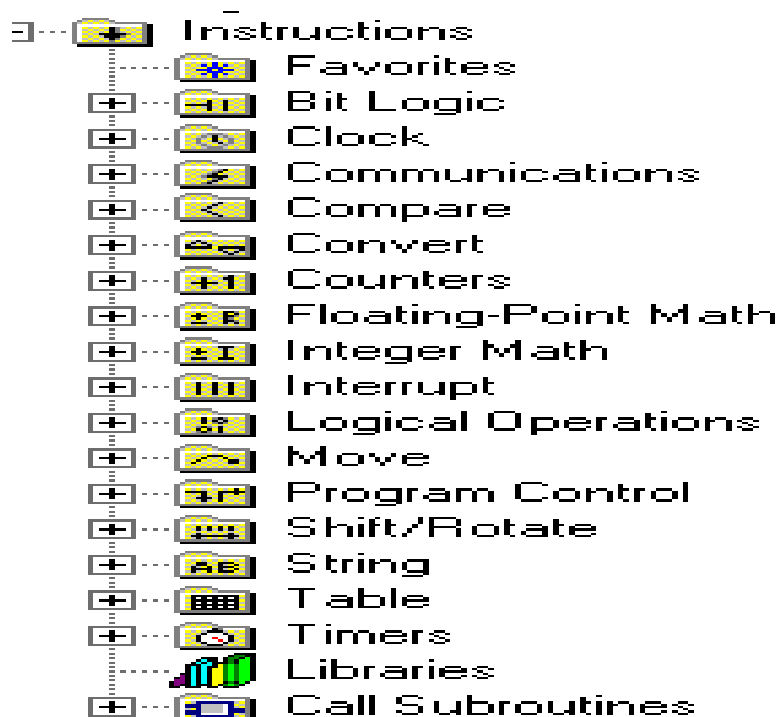


Pause Program Status (Debug menu): Dừng ON/OFF trạng thái chương trình trong PLC.

-  Chart Status (Debug menu): ON/OFF hiển thị trạng thái dữ liệu trong bảng Status chart.
-  Trend View (View menu): ON/OFF xem trạng thái dữ liệu trong PLC ở dạng đồ thị
-  Pause Trend View: Dừng việc vẽ đồ thị dữ liệu
-  Single Read (Debug menu): Sử dụng Single Read để cập nhật một lần tất cả các giá trị trong bảng Status Chart.
-  Write All (Debug menu): Ghi tất cả các giá trị ở cột New Value trong bảng Status Chart vào PLC.
-  Force (Debug menu): Cường bức dữ liệu PLC
-  Unforce For (Debug menu): Gỡ bỏ cường bức dữ liệu PLC
-  Unforce All (Debug menu): Gỡ bỏ tất cả các cường bức trong bảng Status Chart.
-  Read All Forced (Debug menu): Đọc tất cả các giá trị cường bức trong Status Chart.

Hình 2.27. Các chức năng trên thanh công cụ

- Vùng cây lệnh: Bao gồm tất cả các lệnh của PLC



Hình 2.28. Vùng cây lệnh

- Vùng soạn thảo: Là nơi soạn thảo chương trình chính



Hình 2.29 Vùng soạn thảo chính

5.2. Các bước thiết lập một dự án mới

5.2.1. Tạo dự án mới

Để tạo một dự án mới ta vào giao diện chính của phần mềm chọn menu **file=>new** hoặc ấn tổ hợp phím **Ctrl+N** hoặc nhấn vào biểu tượng  trên thanh toolbar để mở vùng soạn thảo mới.

5.2.2. Soạn thảo chương trình

5.2.3. Lập bảng phân công vào/ra

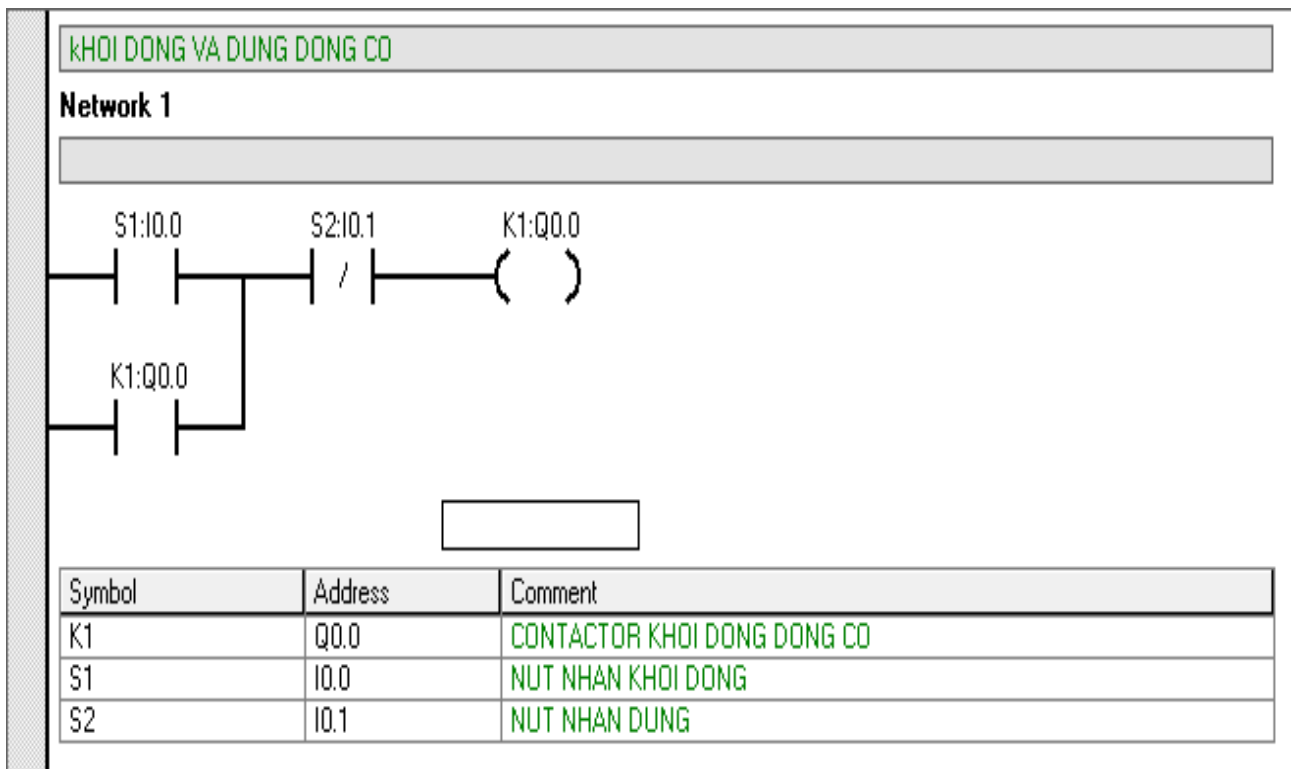
Trong thanh chức năng bấm vào biểu tượng  để vào giao diện symbol table. Tại đây ta khai báo kí hiệu, địa chỉ, chú thích cho các đầu vào ra.

			Symbol	Address	Comment
1			S1	I0.0	NUT NHAN KHOI DONG
2			S2	I0.1	NUT NHAN DUNG
3			K1	Q0.0	CONTACTOR KHOI DONG DONG CO
4					
5					

Hình 2.30. Bảng phân công vào ra

5.2.4. Viết chương trình điều khiển

Bấm vào biểu tượng  để quay lại màn hình chính. Sử dụng các lệnh trong cây lệnh để viết chương trình

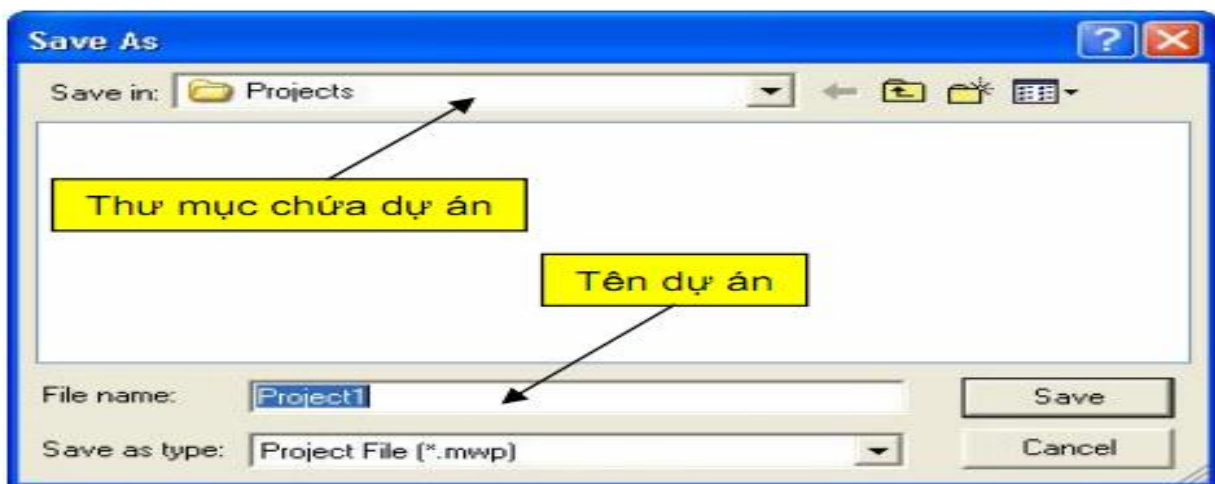


Hình 2.31. Một chương trình đơn giản

Ta viết chương trình bắt đầu từ network 1 và tiếp tục các network tiếp theo.

5.2.5. Lưu dự án

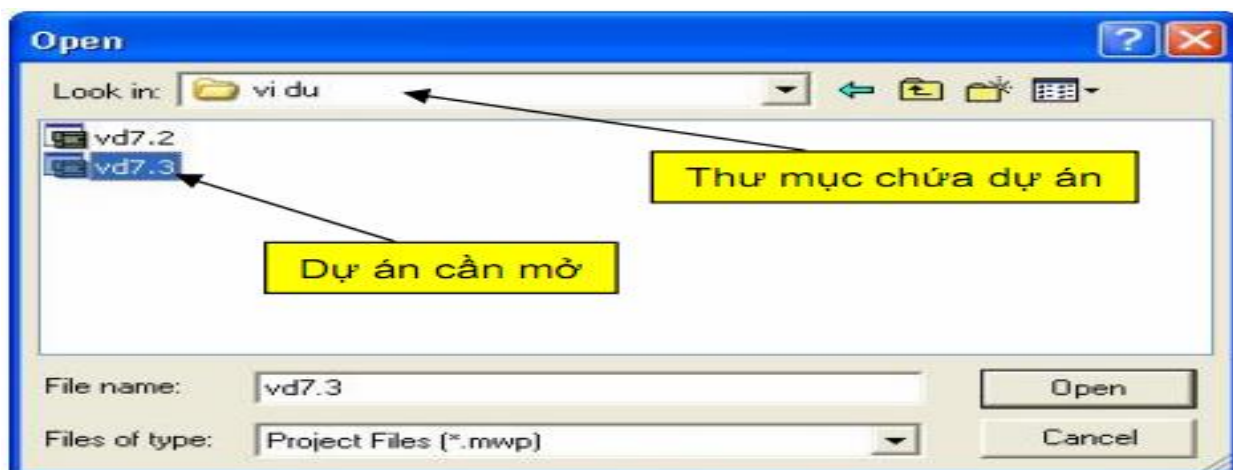
Để lưu dự án ta nhấp chuột vào biểu tượng  hoặc vào menu **File>save** cửa sổ xuất hiện chọn thư mục cần lưu, điền tên dự án và bấm **Save**



Hình 2.32. Lưu dự án

5.2.6. Mở một dự án

Để mở một dự án đang có sẵn ta nhấp chuột vào biểu tượng  hoặc vào menu **File>open** hoặc bấm tổ hợp phím **Ctrl+O**. Cửa sổ màn hình xuất hiện, chọn thư mục chứa chương trình, chọn tên dự án và bấm **Open**



Hình 2.33. Mở một dự án có sẵn

5.2.7. Nạp một dự án vào PLC

Khi cho phép kết nối giữa PLC với PC ta có thể download chương trình xuống PLC

- Trước khi download cần kiểm tra xem PLC đã ở chế độ dừng chưa thông qua đèn báo STOP. Nếu chưa ta chuyển về chế độ STOP

- Nhấp chuột vào biểu tượng download  trên thanh toolbar hoặc chọn File>download. Hộp hội thoại download xuất hiện

- + Chọn các khối cần download, thông thường là chọn hết

- + Nhấn OK để bắt đầu quá trình tải xuống

- + Sau khi download thành công sẽ có thông báo download successful

- + Nhấn biểu tượng RUN  trên thanh toolbar để đưa PLC về chế độ hoạt động. (Chú ý: lúc này trên PLC công tắc chọn chế độ phải ở vị trí TERM hoặc RUN)

- + Trong lúc chương trình đang chạy ấn biểu tượng Program status  để theo dõi trạng thái của các tiếp điểm, cuộn dây, timer, counter...

- + Nhấn biểu tượng STOP  để dừng chương trình đang chạy.

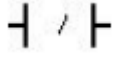
6. Ngôn ngữ lập trình

Có 3 dạng soạn thảo thông dụng là LAD, FBD, STL. Việc sử dụng dạng nào là do người lập trình tự chọn lựa.

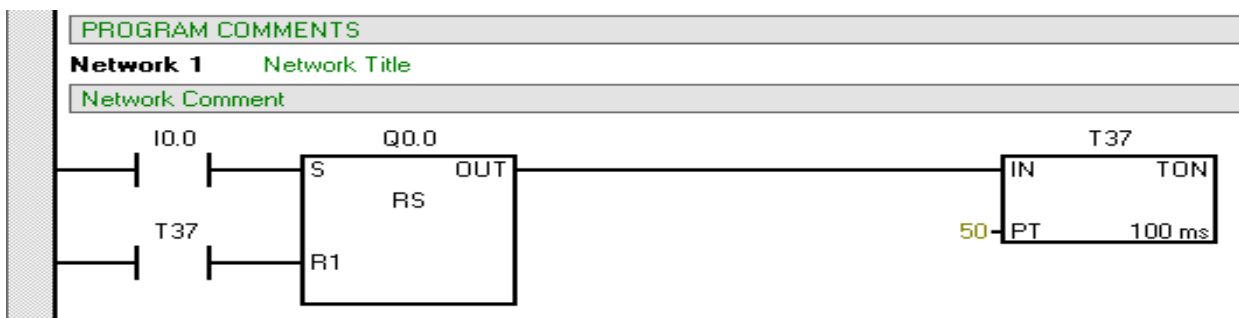
6.1. Dạng hình thang LAD(Ladder logic)

Ở dạng soạn thảo này chương trình được hiển thị gần giống với sơ đồ nối dây một mạch trang bị điện dùng rơ le, contactor. Chúng ta xem như có một dòng điện chạy qua một loạt tiếp điểm ngõ vào từ trái qua phải để tới ngõ ra. Chương trình điều khiển được chia thành nhiều network, mỗi network thực hiện

một nhiệm vụ nhỏ và cụ thể. Các network được xử lý từ trên xuống dưới, từ trái sang phải. Các phần tử chủ yếu dùng trong dạng này là:

- + Tiếp điểm thường hở : 
- + Tiếp điểm thường đóng : 
- + Ngõ ra : 
- + Các hộp chức năng như RS,SR,timer,counter...

Ví dụ:



Hình 2.34. Chương trình dạng LAD

Ưu điểm:

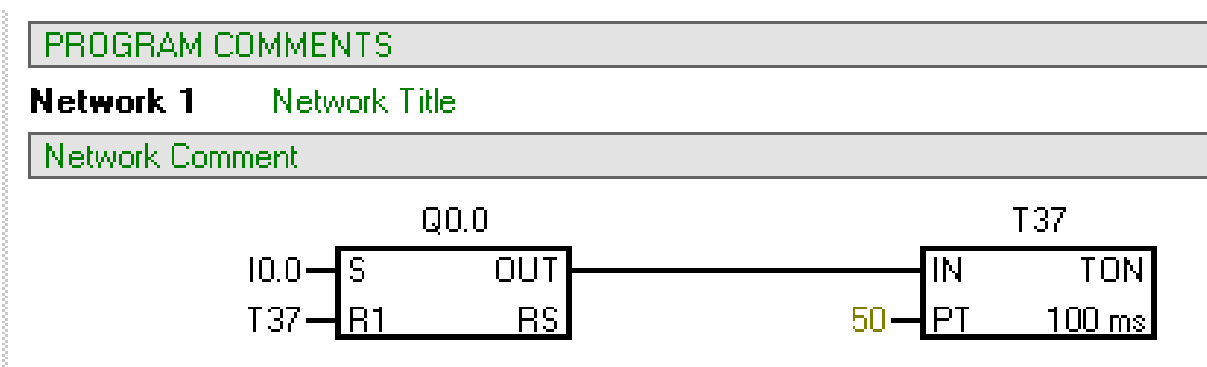
- + Dễ dàng tiếp cận đối với những người mới bắt đầu lập trình
- + Biểu diễn dạng đồ họa dễ hiểu và thông dụng
- + Có thể chuyển qua các dạng khác một cách dễ dàng.

Chú ý: Trong giáo trình này ta sẽ tập trung trình bày ngôn ngữ lập trình theo dạng LAD này.

6.2. Dạng khối chức năng: FBD(Function Block Diagram)

Chương trình ở trên có thể chuyển sang dạng FBD như sau:

- Vào menu **View** tích vào dạng **FBD**, ở màn hình chính chương trình từ dạng LAD chuyển sang dạng FBD

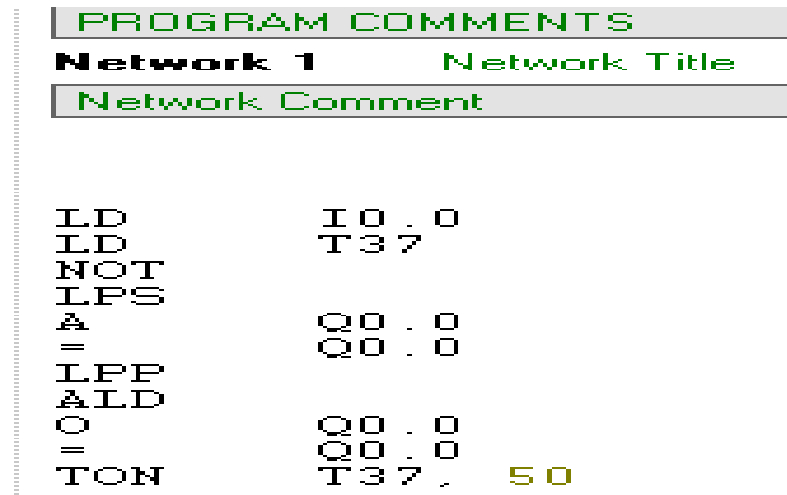


Hình 2.35. Chương trình dạng FBD

6.3. Dạng liệt kê lệnh STL (StaTement List)

Đây là dạng soạn thảo tập hợp các câu lệnh. Có thể chuyển chương trình trên sang dạng STL như sau:

+ Vào menu **View** chọn **STL** màn hình chính sẽ thay đổi thành



Hình 2.36. Chương trình dạng STL

CÂU HỎI VÀ BÀI TẬP

Câu 1: Mô tả cấu trúc bên ngoài của PLC S7-200 CPU 224 DC/DC/DC

Câu 2: Nêu ý nghĩa tên của CPU 224 DC/DC/DC và 222 AC/DC/RLY

Câu 3: Vẽ cách đấu nối nguồn đối với CPU 224 DC/DC/DC

Câu 4: Cho các ví dụ về truy xuất theo bit,byte,word.

Câu 5: Lập bảng địa chỉ cho các ngõ vào/ra sau: Nút nhấn S1, S2, S3, Công tắc hành trình CTHT_T,CTHT_P, Tiếp điểm rơ le nhiệt RN, Cảm biến CB1, CB2, Đèn D1, D2, D3, Coctactor K1, K2, K3, Van điện V1, V2

Câu 6: Nêu các bước thiết lập một dự án mới sử dụng phần mềm Step 7 microwin

Câu 7: Nêu các dạng của ngôn ngữ lập trình PLC

BÀI 3: CÁC LỆNH LẬP TRÌNH CƠ BẢN

Mã Bài: MĐ 20 - 03

Giới thiệu:

Trong PLC có 3 ngôn ngữ lập trình cơ bản LAD là ngôn ngữ lập trình phù hợp các lập trình viên chuyên sâu về điện, FBD dành cho lập trình viên chuyên ngành điện tử, STL ngôn ngữ dành cho các kỹ sư tin học. trong chương trình môn học PLC cơ bản các câu lệnh lập trình chủ yếu tập chung vào cấu trúc ngôn ngữ LAD. Thông qua các câu lệnh cơ bản người lập trình viên ứng dụng luôn vào các bài tập điều khiển trong thực tế sản xuất điều khiển đóng cắt động cơ, điều khiển đảo chiều quay động cơ, điều khiển đổi nối sao tam giác, điều khiển hãm động năng, điều khiển đèn giao thông ở ngã tư...

Mục tiêu:

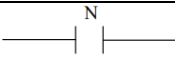
- Biết cách sử dụng các lệnh lập trình cơ bản
- Viết được chương trình cho một dự án nhỏ
- Sửa lỗi chương trình
- Rèn luyện tính cẩn thận, tinh thần làm việc nhóm cho sinh viên.

Nội dung chính :

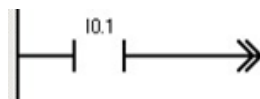
* Ngôn ngữ lập trình được sử dụng là dạng LAD

1. Các lệnh xử lý Bit

1.1. Lệnh tiếp điểm thường mở

LAD		Tiếp điểm thường hở sẽ đóng khi giá trị của bit có địa chỉ là N bằng 1. Toán hạng N: I, Q, M, SM, T, C, V.
-----	---	---

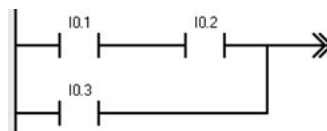
Ví dụ:



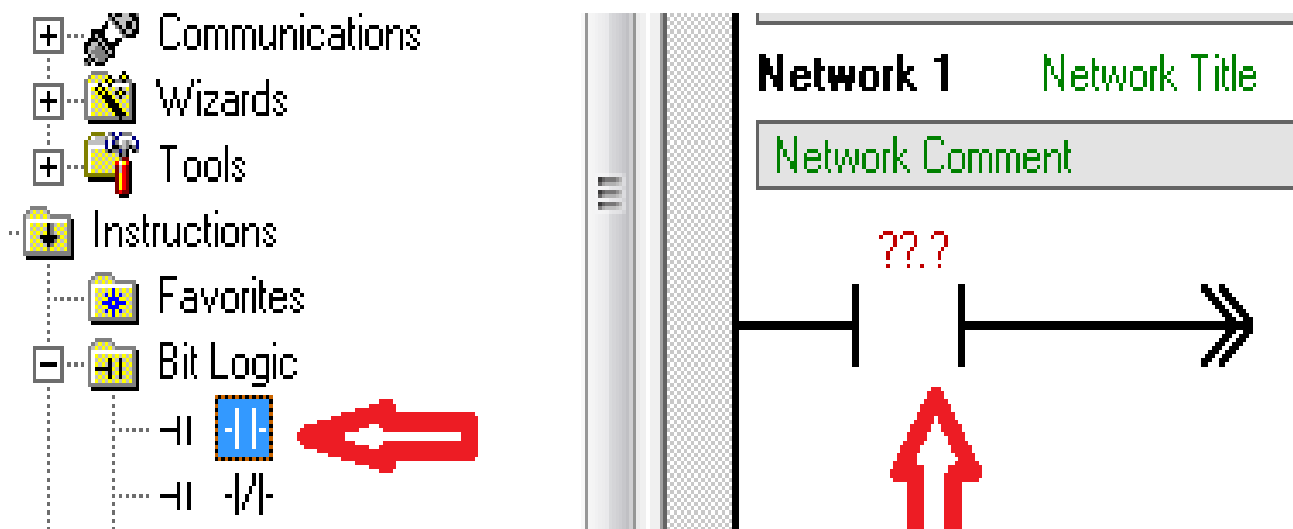
Bình thường tại tiếp điểm này sẽ hở mạch, khi có tín hiệu mức 1 (24VDC) vào I0.1 thì làm tiếp điểm này đóng lại.

Sử dụng không hạn định số lệnh tiếp điểm trên cùng 1 địa chỉ.

Có thể mắc nối tiếp hoặc song song nhiều lệnh tiếp điểm.

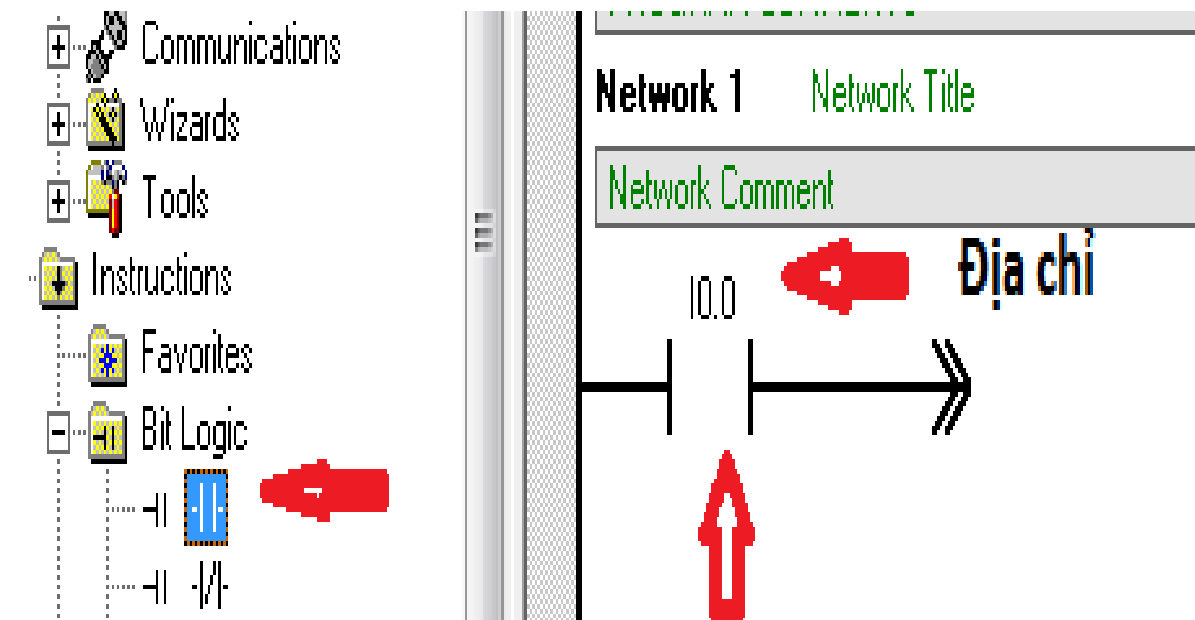


Cách lấy lệnh: Đưa con trỏ chuột vào vùng cây lệnh, mở rộng mục bit logic , và kích đúp vào lệnh . Màn hình chính xuất hiện



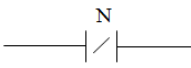
Hình 3.1. Lấy lệnh tiếp điểm thường mở

Phân dấu chấm hỏi ta điền địa chỉ. Địa chỉ có thể là địa chỉ ngõ vào (nút nhấn, công tắc, cảm biến...): I0.0, I0.1, I1.0... hoặc các địa chỉ trung gian: M0.0, M0.1, Q0.0...

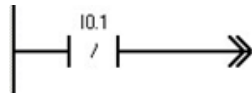


Hình 3.2. Nhập địa chỉ Bit

1.2. Lệnh tiếp điểm thường đóng

LAD		Tiếp điểm thường đóng sẽ đóng khi giá trị của bit có địa chỉ là N bằng 0. Toán hạng N: I, Q, M, SM, T, C, V.
-----	---	---

Ví dụ:



Bình thường tại tiếp điểm này sẽ nóng mạch, khi có tín hiệu mức 1 (24VDC) đưa vào I0.1 thì làm tiếp điểm này sẽ hở ra.

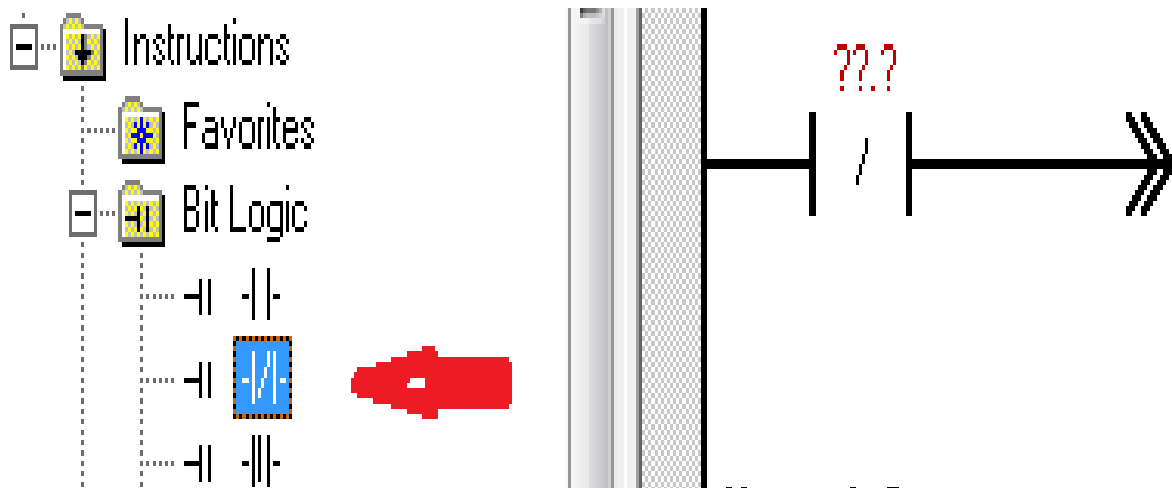
Sử dụng không hạn định số lệnh tiếp điểm trên cùng 1 địa chỉ.

Có thể mắc nối tiếp hoặc song song nhiều lệnh tiếp điểm.



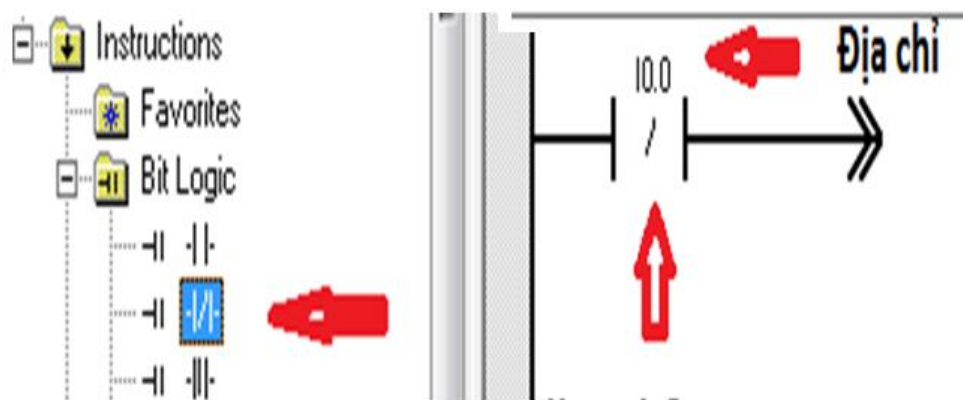
Cách lấy lệnh: Đưa con trỏ chuột vào vùng cây lệnh, mở rộng mục bit

logic Bit Logic, và kích đúp vào lệnh . Màn hình chính xuất hiện



Hình 3.3. Lấy lệnh tiếp điểm thường đóng

- Phần địa chỉ cũng giống như trên.



Hình 3.4. Nhập địa chỉ Bit

- Về kết quả xử lý tiếp điểm trong PLC ta có bảng sau :

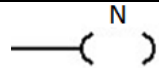
Bộ tạo tín hiệu nhị phân			Thực hiện trong chương trình PLC				
Cảm biến, nút nhấn là một ...	Cảm biến, nút nhấn bị ...	Điện áp tại ngõ vào PLC	Trạng thái tín hiệu tại ngõ vào	Kiểm tra cho trạng thái tín hiệu "1"		Kiểm tra cho trạng thái tín hiệu "0"	
				Ký hiệu/lệnh	Kết quả kiểm tra	Ký hiệu/lệnh	Kết quả kiểm tra
NO 	tác động 	có	1	LAD:  "tiếp điểm không đảo"	1	LAD:  "tiếp điểm đảo"	0
	không tác động 	không	0		0		1
NC 	tác động 	không	0		0		1
	không tác động 	có	1		1		0

Hình 3.5. Kết quả xử lý tiếp điểm NO, NC

Nhận xét

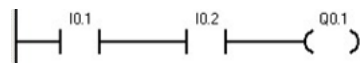
- + Ngõ vào có mức logic "1" khi ngõ vào có điện áp.
- + Nếu ngõ vào được nối với tiếp điểm thường đóng thì ngõ vào ở trạng thái bình thường luôn có điện. Nó chỉ mất điện khi tiếp điểm thường đóng tác động.
- + Nếu ngõ vào được nối với tiếp điểm thường hở (NO - công tắc, nút nhấn, cảm biến) thì ngõ vào ở trạng thái bình thường không có điện. Nó chỉ có điện khi tiếp điểm NO bị tác động
- + Nếu sử dụng  trong chương trình thì trạng thái tiếp điểm sẽ có cùng trạng thái logic với ngõ vào.
- + Nếu sử dụng  trong chương trình thì trạng thái tiếp điểm sẽ ngược trạng thái logic với ngõ vào.

1.3. Lệnh cuộn dây

LAD		Giá trị của bit có nĩa chỉ là n sẽ bằng 1 khi nĩa vào của lệnh này bằng 1 và ngược lại Toán hạng N: Q, M, SM, T, C, V. Chỉ sử dụng 1 lệnh Out cho 1 nĩa chỉ.
-----	---	---

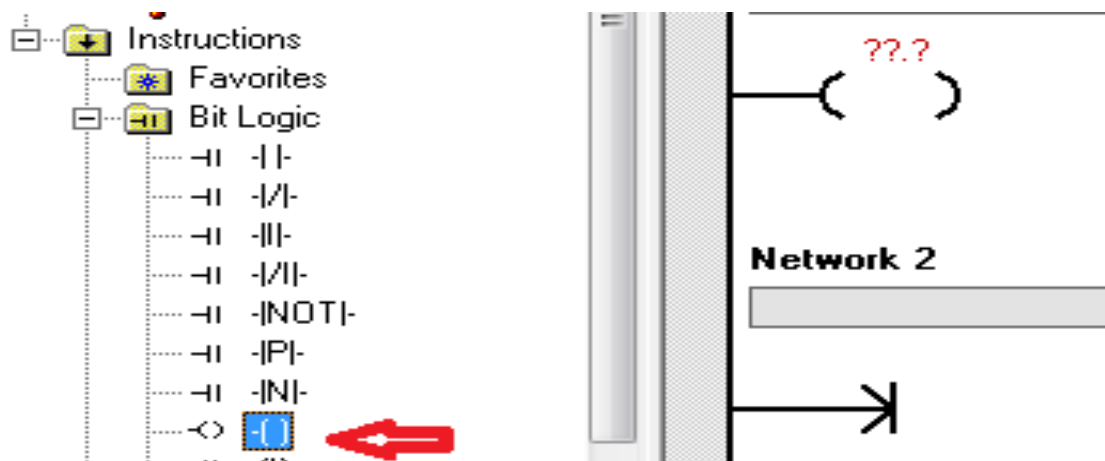
- Cuộn dây $()$ có thể đại diện cho bóng đèn, contactor, van điện từ...là những thiết bị chấp hành.

Ví dụ:



- Ngỏ ra bằng 0 khi chỉ 1 trong 2 ngỏ vào = 0. Bằng 1 khi cả 2 ngỏ vào này = 1

- Cách lấy lệnh: vào mục bit logic kích đúp biểu tượng $()$.



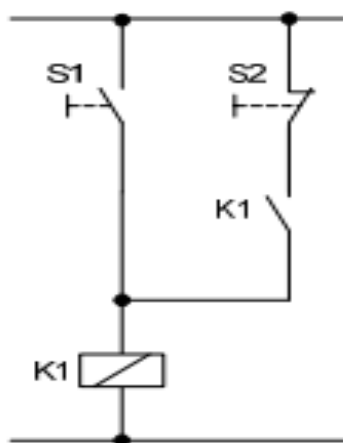
Hình 3.6. Lấy lệnh cuộn dây

1.4. Bài toán ứng dụng

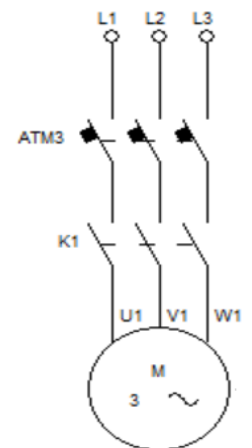
1.4.1. Mạch tự duy trì ưu tiên mở máy

Lập trình PLC S7-200 điều khiển động cơ không đồng bộ 3 pha rotor lồng sóc đảm bảo yêu cầu sau:

- Ấn S1 động cơ chạy
- Ấn S2 động cơ dừng
- Nếu ấn cùng lúc S1 và S2 thì ưu tiên mở máy.



Mạch điều khiển trang bị điện



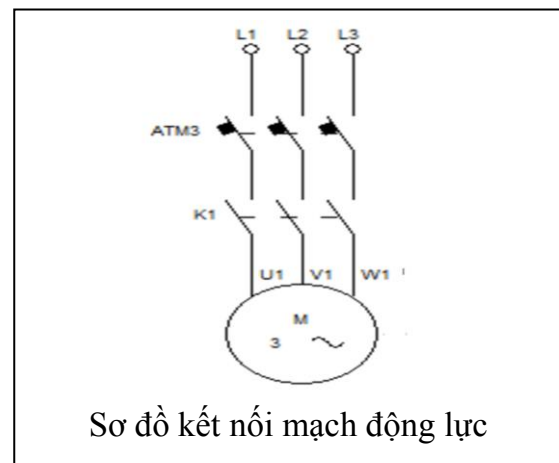
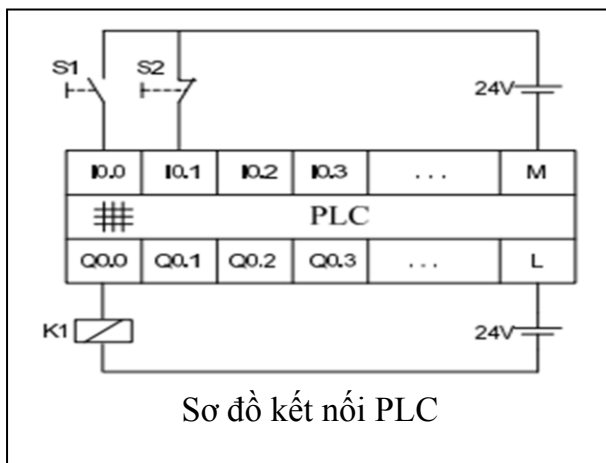
Mạch động lực

Đáp án:

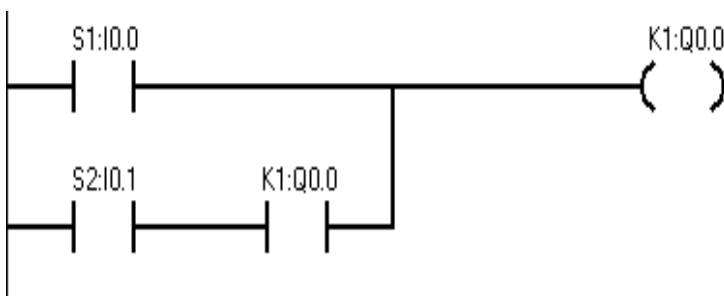
Bước 1: Lập bảng địa chỉ vào ra

Kí hiệu	Địa chỉ	Ghi chú
S1	I0.0	Nút nhấn mở máy(NO)
S2	I0.1	Nút nhấn dừng máy(NC)
K1	Q0.0	Công tắc tơ điều khiển

Bước 2: Sơ đồ kết nối



Bước 3: Chương trình PLC

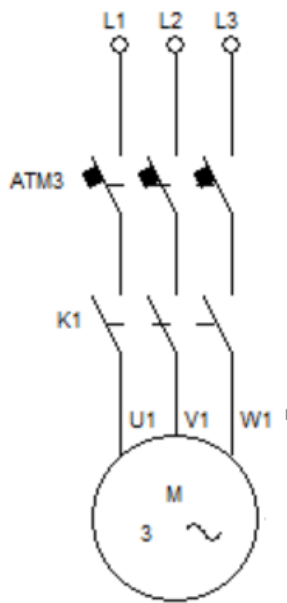


Symbol	Address	Comment
K1	Q0.0	CONTACTOR DIEU KHIEN
S1	I0.0	NUT NHAN MO MAY (NO)
S2	I0.1	NUT NHAN DUNG MAY (NC)

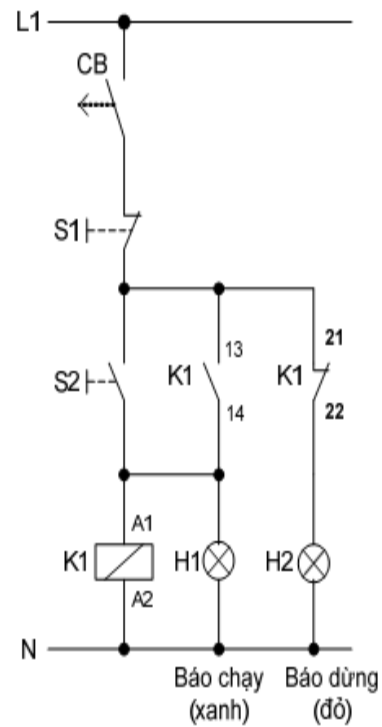
1.4.2. Điều khiển động cơ có chỉ báo

Lập trình PLC S7-200 điều khiển động cơ không đồng bộ 3 pha rotor lồng sóc đảm bảo yêu cầu sau:

- Ấn S2 động cơ chạy đồng thời đèn H1 sáng báo động cơ đang hoạt động
- Ấn S1 động cơ dừng đồng thời đèn H2 sáng báo động cơ dừng



Mạch động lực



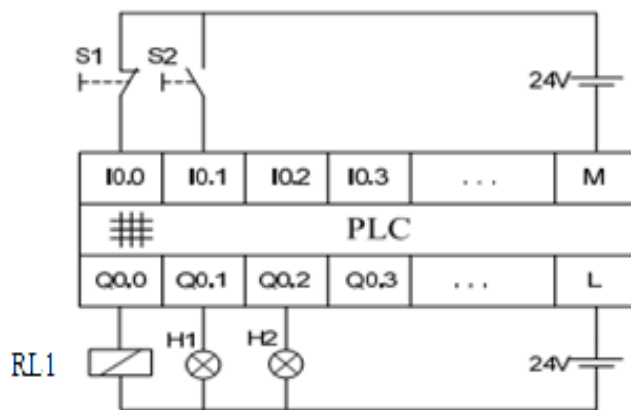
Mạch điều khiển trang bị điện

Đáp án:

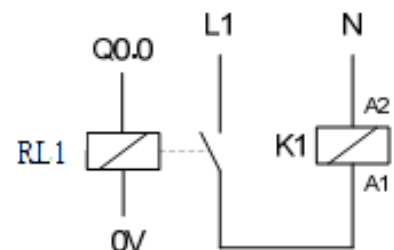
Bước 1: Lập bảng địa chỉ vào ra

Kí hiệu	Địa chỉ	Ghi chú
S1	I0.0	Nút nhấn dừng máy(NC)
S2	I0.1	Nút nhấn mở máy(NO)
RL1	Q0.0	Rơ le trung gian 1
H1	Q0.1	Đèn báo động cơ hoạt động
H2	Q0.2	Đèn báo động cơ dừng hoạt động

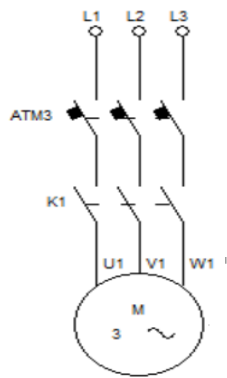
Bước 2: Sơ đồ kết nối



Sơ đồ kết nối PLC

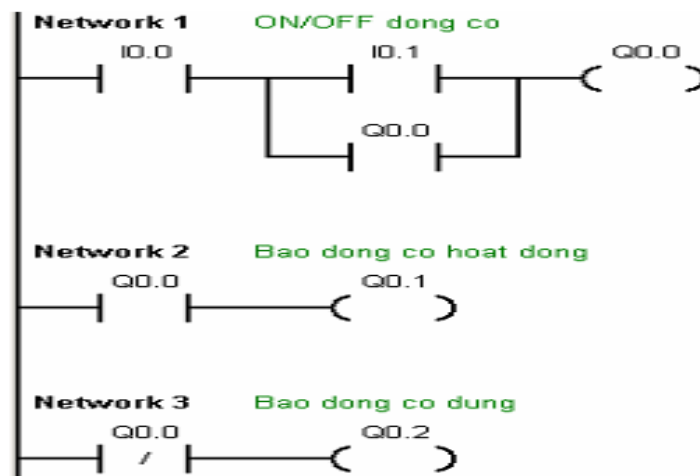


Sơ đồ kết nối rơ le trung gian với công tắc tơ



Sơ đồ kết nối mạch động lực

Bước 3: Viết chương trình

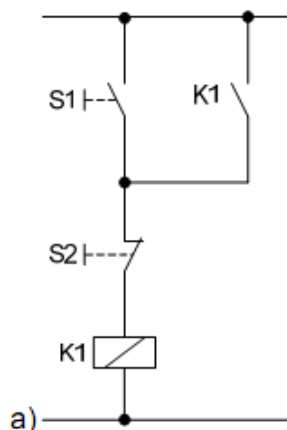


* Bài tập

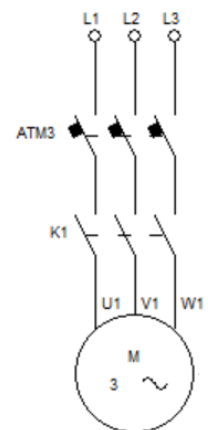
Cho mạch ưu tiên dừng máy như hình.

Lập trình PLC S7-200 điều khiển động cơ không đồng bộ 3 pha rotor lồng sóc đảm bảo yêu cầu sau:

- Ấn S1 động cơ chạy
- Ấn S2 động cơ dừng
- Nếu ấn cùng lúc S1 và S2 thì ưu tiên dừng máy.



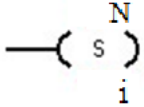
Sơ đồ kết nối mạch điều khiển



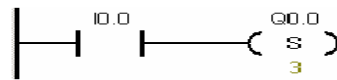
Sơ đồ kết nối mạch động lực

1.5. Lệnh Set, Reset

1.5.1. Lệnh Set

LAD		Giá trị của các bit có địa chỉ đầu tiên là N sẽ bằng 1 khi đầu vào của lệnh này bằng 1. Khi đầu vào của lệnh bằng 0 thì các bit N này vẫn giữ nguyên trạng thái. i là số bit cần thiết lập giá trị lên 1. Toán hạng n: Q, M, SM, T, C, V. i: IB, QB, MB, SMB, VB, AC, hằng số.
-----	---	---

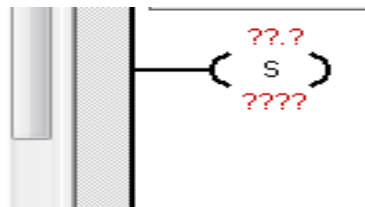
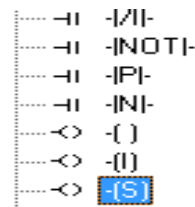
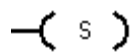
Ví dụ:



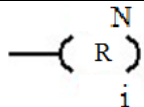
+ Khi ngõ vào I0.0=1 thì sẽ set bit Q0.0, Q0.1, Q0.2 lên mức 1.

+ Khi I0.0=0 thì ngõ ra Q0.0, Q0.1, Q0.2 vẫn ở mức 1.

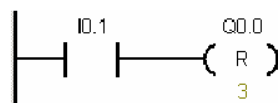
- Cách lấy: Vào cây lệnh, mở rộng mục bit logic, kích đúp vào biểu tượng



1.6.2. Lệnh Reset

LAD		Giá trị của các bit có địa chỉ đầu tiên là N sẽ bằng 0 khi đầu vào của lệnh này bằng 1. Khi đầu vào của lệnh bằng 0 thì các bit N này vẫn giữ nguyên trạng thái. i là số bit cần xóa giá trị về 0. Toán hạng n: Q, M, SM, T, C, V. i: IB, QB, MB, SMB, VB, AC, hằng số.
-----	---	--

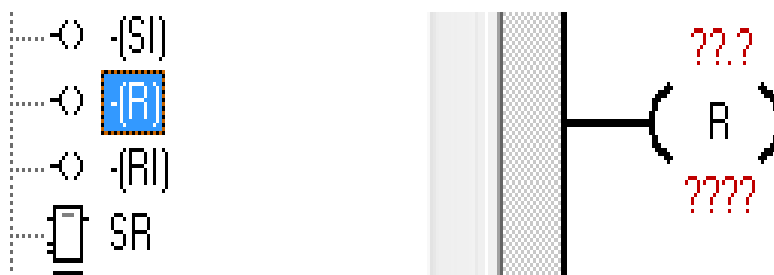
- Ví dụ:



Khi ngõ vào I0.1 lên “1” thì 3 ngõ ra Q0.0, Q0.1, Q0.2 xuống mức “0” và giữ nguyên trạng thái này cho đến khi gặp lệnh được set bằng một lệnh khác.

- Cách lấy: Vào vùng cây lệnh mở rộng mục **bit logic**, kích đúp vào lệnh

—(R)



* **Bài tập:** làm lại các bài tập ở phần trên bằng các lệnh Set, Reset

2. Timer

2.1. Giới thiệu

Bộ định thời (Timer) được sử dụng trong các ứng dụng điều khiển cần trì hoãn thời gian.

- S7-200 có 256 bộ Timer từ T0-T255 chia làm 3 loại:

- + Timer đóng mạch chậm TON (On-delay Timer)
- + Timer đóng mạch chậm có nhớ TONR (Retentive on-delay timer)
- + Timer ngắt mạch chậm TOF (Off-delay timer)

- Khi sử dụng một loại timer ta cần xác định các thông số sau:

- + Loại timer (TON, TONR hay TOF)
- + Độ phân giải của timer (Có 3 độ phân giải là 1ms, 10ms, 100ms)
- + Số của timer cần sử dụng (T0, T1, ..T37...). Cần tra bảng để biết loại timer cần sử dụng tương ứng với loại nào
- + Khai báo thời gian trì hoãn dựa vào độ phân giải của timer
- + Tín hiệu cho phép bắt đầu tính thời gian

* **Chú ý:** Trong giáo trình này ta chỉ cần quan tâm đến timer đóng mạch chậm TON.

2.2. Timer đóng mạch chậm TON

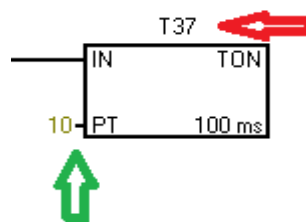
LAD		Khi ngõ vào chân IN lên mức 1 thì bộ định thời Ton hoạt động đếm thời gian, khi thời gian đếm được lớn hơn hoặc bằng giá trị đặt trước PT thì các tiếp điểm của bộ định thời tác động. Các tiếp điểm thường mở sẽ đóng và các tiếp điểm thường đóng sẽ mở ra. Khi ngõ vào chân IN xuống mức 0 thì thời gian đếm
-----	--	---

		được sẽ bị reset đi, bộ Ton ngừng hoạt động và tiếp điểm của bộ định thời thường mở sẽ mở ra và tiếp điểm thường đóng sẽ đóng. Toán hạng: Txxx Ton 1ms T32, T96 10ms T33 -> T36; T97 -> T100 100ms T37 -> T63; T101 -> T255
--	--	--

- Tính thời gian trễ:

Thời gian trì hoãn = [PT] * độ phân giải

Ví dụ:



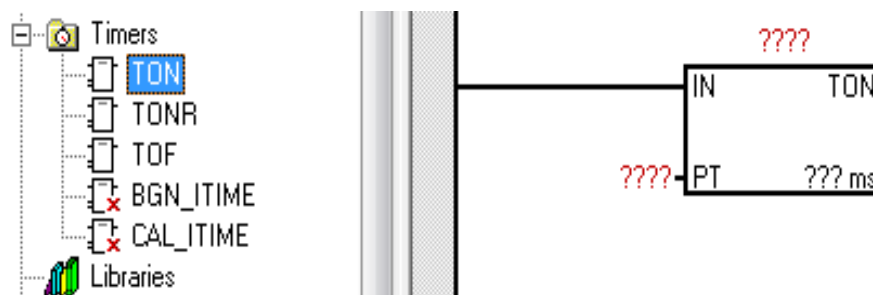
+ Số timer sử dụng là T37

+ Độ phân giải là 100 ms

+ Thời gian trì hoãn = $10 \times 100\text{ms} = 1000\text{ms} = 1\text{s}$.

- Cách lấy lệnh: vào vùng cây lệnh ta nhấp chuột vào dấu (+) ở biểu tượng

Timers sau đó kích đúp vào biểu tượng TON.



Hình 3.7. Lấy lệnh TON

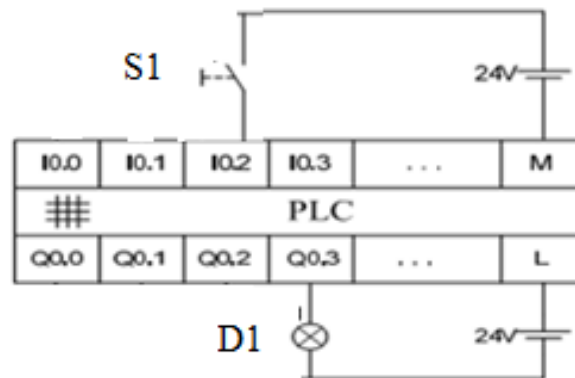
Ví dụ ứng dụng:

Bài toán: Ấn S1 sau 1 giây đèn D1 sáng, nhả nút nhấn đèn D1 tắt ngay

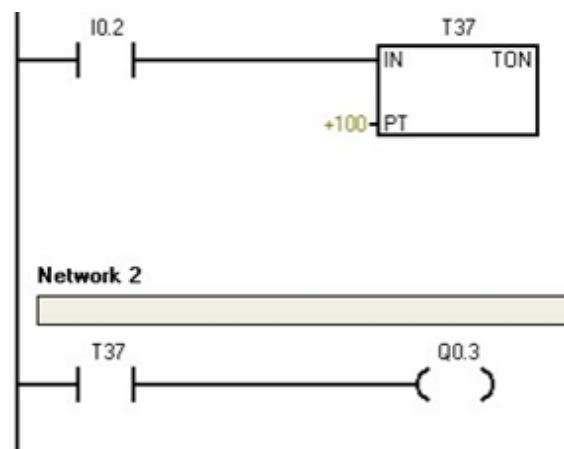
Bước 1: Lập bảng phân công vào ra

Kí hiệu	Địa chỉ	Ghi chú
S1	I0.2	Nút nhấn
D1	Q0.3	Đèn D1

Bước 2: Sơ đồ kết nối



Bước 3: Chương trình PLC

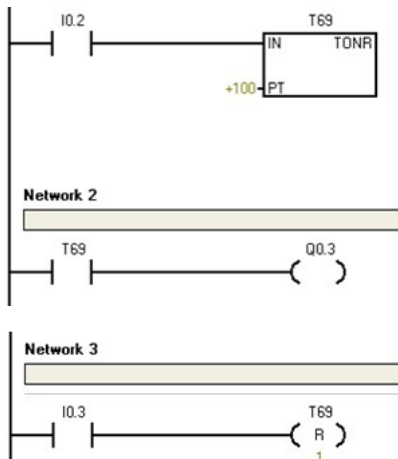


2.3. Timer đóng mạch chậm có nhớ TONR

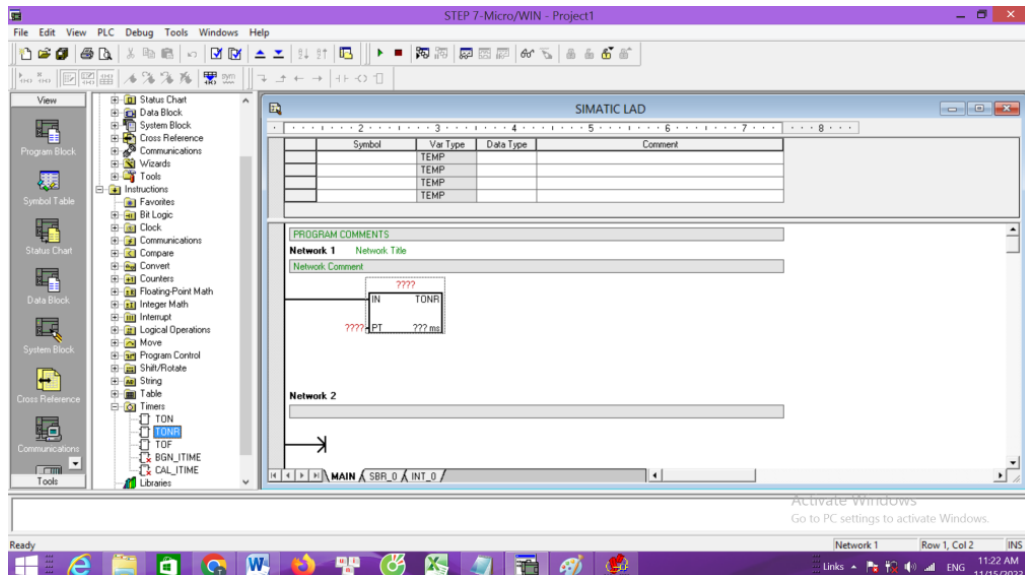
LAD		Khi ngõ vào chân IN lên mức 1 thì bộ định thời TONR hoạt động đếm thời gian, khi thời gian đếm được lớn hơn hoặc bằng giá trị đặt trước PT thì các tiếp điểm của bộ định thời tác động. Các tiếp điểm thường mở sẽ đóng và các tiếp điểm thường đóng sẽ mở ra. Khi ngõ vào chân IN xuống mức 0 thì thời gian đếm được sẽ không bị reset đi, bộ TONR ngừng hoạt động và các tiếp điểm của bộ định thời giữ nguyên trạng thái. Khi nào chân IN lên mức 1 thì thời gian tiếp tục được đếm kể từ lúc dừng đếm. Toán hạng: Txxx TONR 1ms T0, T64 10ms T1 -> T4; T65 -> T68 100ms T5 -> T31; T69 -> T95
-----	--	---

Ví dụ:

Phân tích yêu cầu công nghệ của chương trình PLC sau



- Cách lấy lệnh: vào vùng cây lệnh ta nhấp chuột vào dấu (+) ở biểu tượng  Timers sau đó kích đúp vào biểu tượng TONR .



Hình 3.8. Lấy lệnh TONR

3. Counter

3.1. Giới thiệu:

S7-200 có 256 bộ đếm địa chỉ từ C0-C255. Có 3 loại bộ đếm là:

- Bộ đếm lên CTU (Up counter)
- Bộ đếm xuống CTD (Down counter)
- Bộ đếm lên xuống CTUD (Up/Down counter)

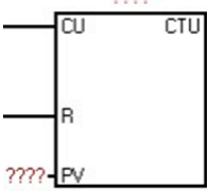
Khi sử dụng một loại counter ta phải chú ý đến các thông số sau:

- Loại counter (CTU, CTD hay CTUD)
- Số của counter sẽ sử dụng, không được gán cùng một số counter cho nhiều counter

- Khai báo giá trị cần đếm
- Tín hiệu xung cấp cho bộ đếm
- Tín hiệu xóa bộ đếm

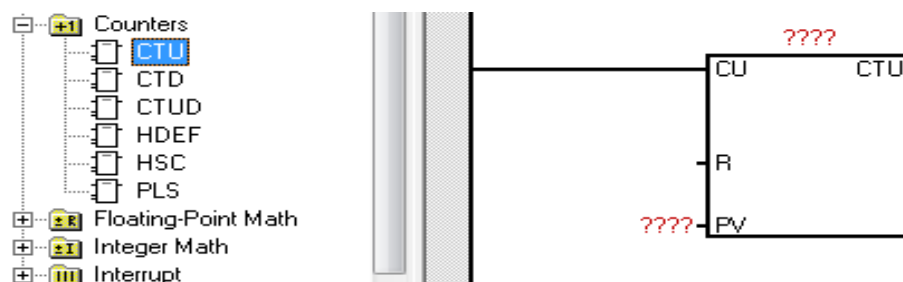
*** Trong giáo trình này ta sẽ tập trung nghiên cứu bộ đếm lên CTU**

3.2. Bộ đếm lên CTU

LAD		Mỗi khi tín hiệu ngõ vào CU chuyển từ 0 lên 1 thì bộ đếm sẽ tăng giá trị hiện hành của nó lên một đơn vị. Khi giá trị hiện hành của bộ đếm tăng lên lớn hơn hoặc bằng giá trị đặt trước PV thì ngõ ra counter bit sẽ được set lên “1” khi đó các tiếp điểm thường hở sẽ đóng và các tiếp điểm thường đóng sẽ mở ra. Khi đầu vào chân R lên mức 1 sẽ reset các giá trị đếm được và các tiếp điểm thường hở sẽ hở và các tiếp điểm thường đóng sẽ đóng. Toán hạng: Cxxx = C0 C47, C80 C127
-----	---	---

- Cách lấy lệnh: Vào vùng cây lệnh bấm vào dấu (+) của biểu tượng

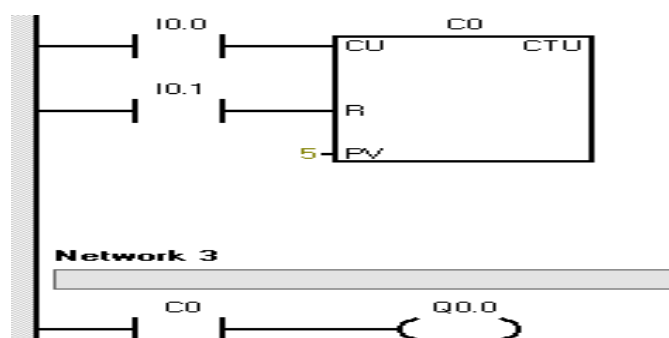
 Counters sau đó kích đúp vào biểu tượng  CTU, màn hình sẽ xuất hiện như hình:



Hình 3.9. Lấy lệnh CTU

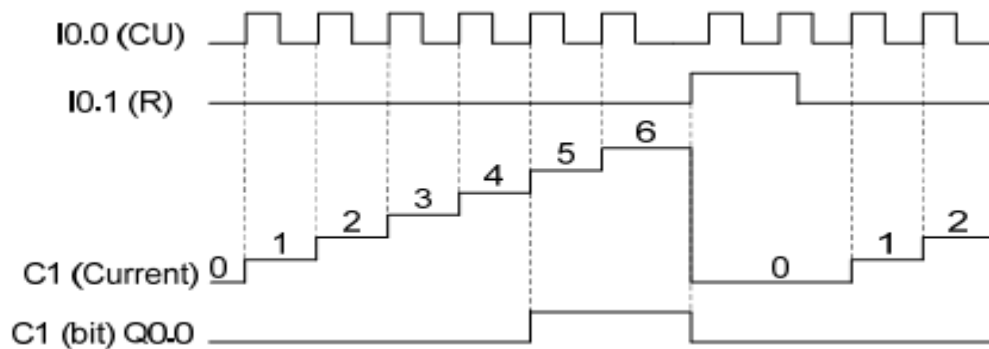
Điền các thông tin, giá trị ở CU, PV, R và số counter

Ví dụ:



- Phân tích:

Khi ngõ vào I0.0 chuyển từ “0” lên “1” 5 lần thì bit C0 sẽ được set lên 1 đồng thời đưa ngõ ra Q0.0 lên 1. Nếu ngõ vào I0.1 chuyển từ 0 lên 1 thì bit C0 bị reset đồng thời ngõ ra Q0. cũng mất điện. Sau đây là giản đồ xung của C0



Hình 3.10. Giản đồ xung của CTU

Ví dụ ứng dụng:

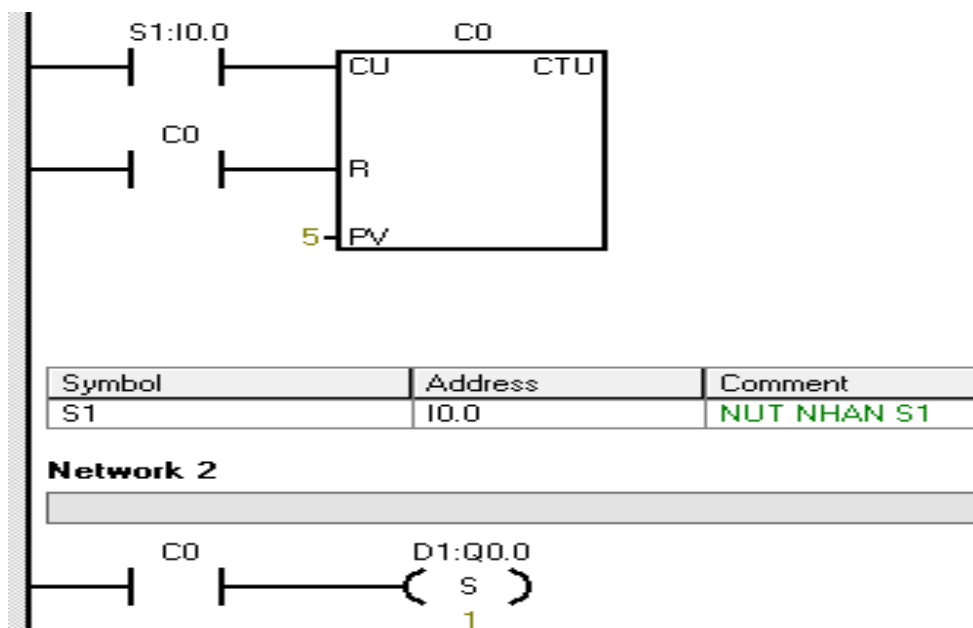
Bài toán: Nhấn 5 lần S1 thì đèn D1 sáng. Sau khi đèn sáng nhấn 4 lần S2 thì đèn tắt

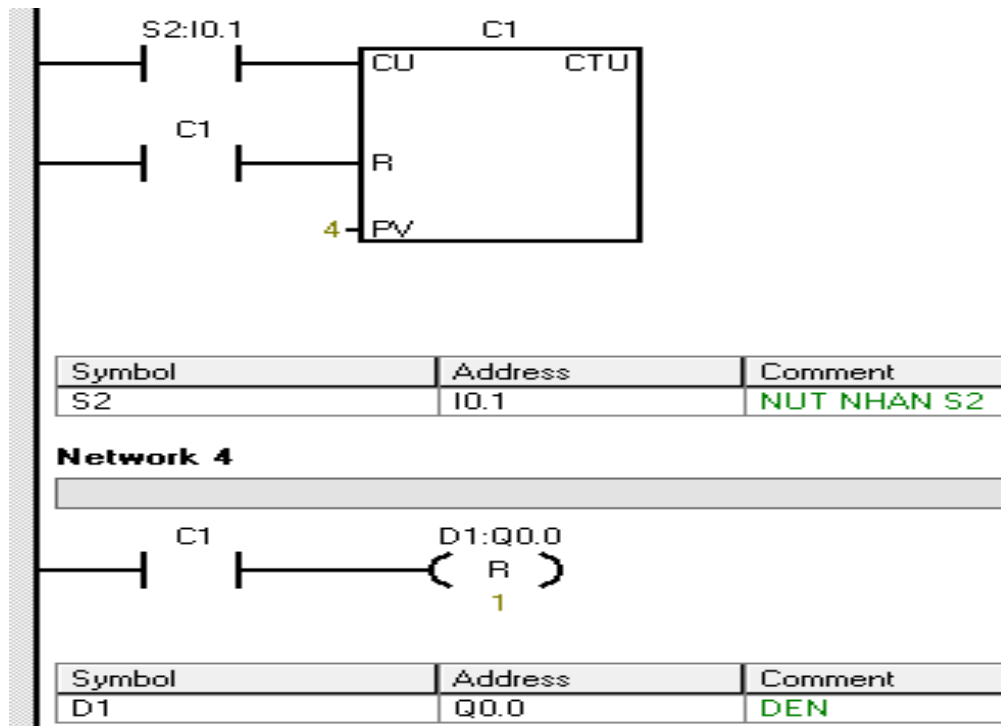
Các bước thực hiện như sau:

Bước 1: Lập bảng quy định địa chỉ vào ra

			Symbol	Address	Comment
1			S1	I0.0	NUT NHAN S1
2			S2	I0.1	NUT NHAN S2
3			D1	Q0.0	DEN

Bước 2: Viết chương trình



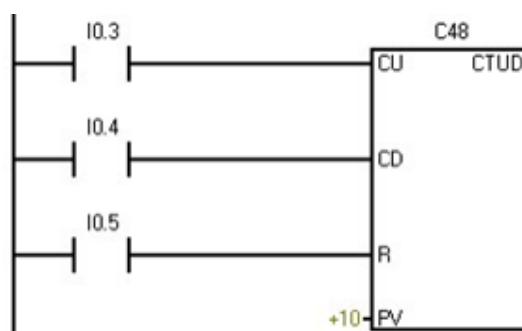


Nhận xét: Ở đây ta dùng chính C0 và C1 để reset bộ đếm tương ứng, tức là khi C0 đếm đến 5 bit C0 được set lên 1 thì ngay lập tức nó reset giá trị bộ đếm về 0.

3.2. Bộ đếm lên xuống CTUD

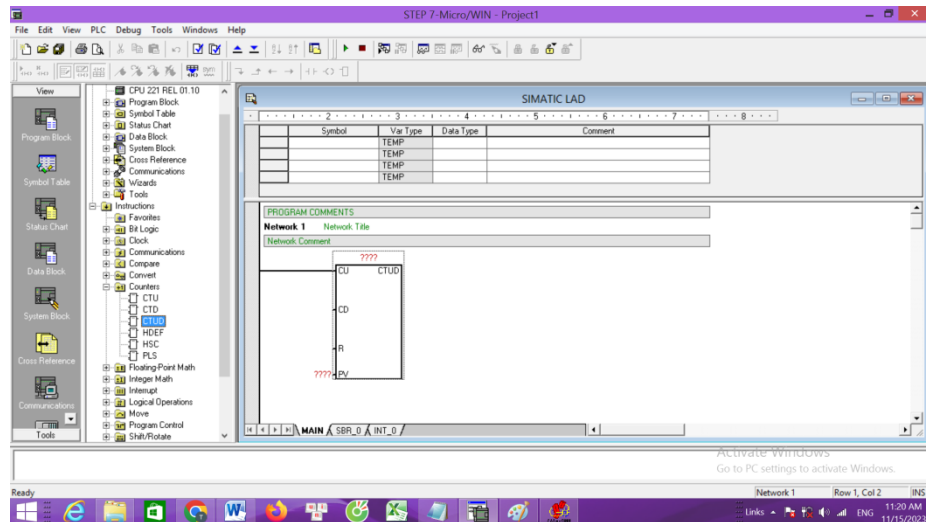
LAD		CTUD đếm lên số sườn lên đầu vào chân CU CTUD đếm xuống số sườn lên đầu vào CD. Khi giá trị đếm được lớn hơn hoặc bằng giá trị đặt trước PV thì các tiếp điểm của bộ CTU này tác động. Các tiếp điểm thường hở sẽ đóng và các tiếp điểm thường đóng sẽ hở. Khi đầu vào chân R lên mức 1 sẽ reset các giá trị đếm được và các tiếp điểm thường hở sẽ hở và các tiếp điểm thường đóng sẽ đóng. Toán hạng: C48 C79
-----	--	---

Ví dụ:



- Cách lấy lệnh: Vào vùng cây lệnh bấm vào dấu (+) của biểu tượng

 sau đó kích đúp vào biểu tượng  , màn hình sẽ xuất hiện như hình:

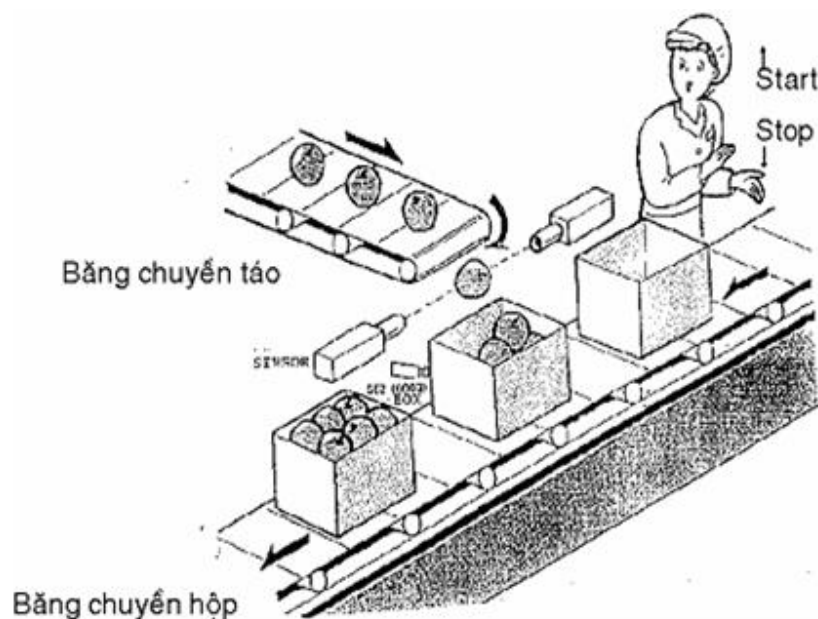


Hình 3.11. Lấy lệnh CTUD

* Bài tập

Lập trình PLC S7-200 điều khiển băng chuyền đóng gói sản phẩm đảm bảo yêu cầu sau:

Nhấn nút Start thì băng chuyền thùng hoạt động, khi thùng đi tới công tắc hành trình S3(NO) thì băng chuyền thùng dừng lại, băng chuyền sản phẩm đã đóng gói bắt đầu chuyển động. Cảm biến S2(NC) được dùng để đếm số lượng sản phẩm. Khi đếm được 12 sản phẩm thì băng chuyền sản phẩm dừng và băng chuyền thùng lại bắt đầu chuyển động. Bộ đếm được đặt lại và quá trình vận hành lặp lại cho đến khi ấn nút Stop(NC).



Hình 3.12. Băng chuyền đóng gói sản phẩm

BÀI 4: THỰC HÀNH LẮP ĐẶT PLC S7-200

Mã Bài: MĐ 20 - 04

Giới thiệu:

Sau khi học xong phần lý thuyết trong các bài từ bài 1 đến bài 3 các phần này sẽ trang bị cho người học các kiến thức đầy đủ từ cách kết nối phần cứng, các câu lệnh cơ bản, phương pháp lập trình để giải quyết các bài toán điều khiển. bài 4 này sẽ giúp người học vận dụng các kiến thức lý thuyết trên vào các bài tập điều khiển trong thực tế như điều khiển đóng cắt động cơ không đồng bộ ba pha rotor lồng sóc, điều khiển đảo chiều quay động cơ không đồng bộ ba pha rotor lồng sóc, đảo chiều đóng cắt động cơ có hãm động năng, điều khiển đèn giao thông ngã tư, điều khiển thang máy xây dựng, điều khiển led 7 đoạn...

Mục tiêu:

- Lắp đặt và đấu nối thành thạo PLC, mạch động lực, mạch điều khiển
- Đảm bảo an toàn con người và thiết bị

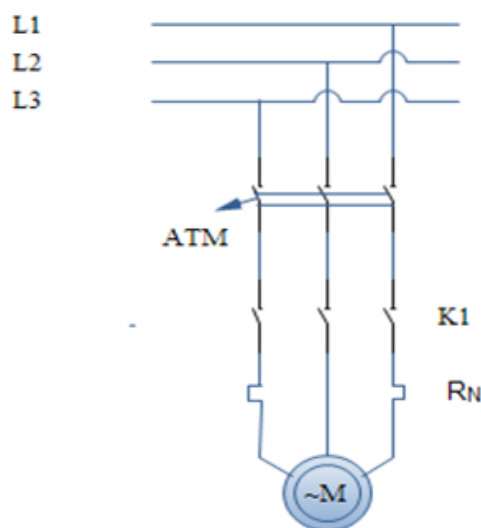
Nội dung chính:

1. Điều khiển đóng cắt động cơ

1.1. Yêu cầu công nghệ

Lập trình PLC S7-200 điều khiển đóng cắt động cơ không đồng bộ ba pha roto lồng sóc đảm bảo các yêu cầu sau:

- Nhấn nút Start contactor K1 đóng lại động cơ chạy.
- Nhấn nút Stop contactor K1 mở ra động cơ dừng.



1.2. Các bước tiến hành

Bước 1: Chuẩn bị dụng cụ, vật tư, bàn thực hành



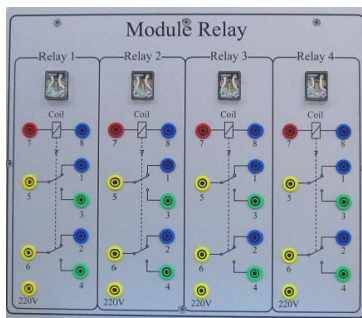
Module nguồn



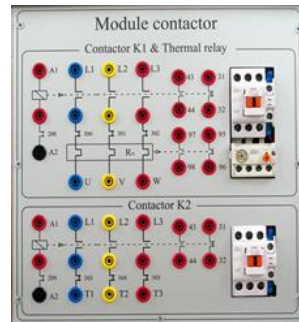
Module nút nhấn



Module PLC



Module Re lay



Module công tắc tơ K1, K2



Động cơ



Dây dẫn



Bàn thực hành gá module



Bộ dụng cụ

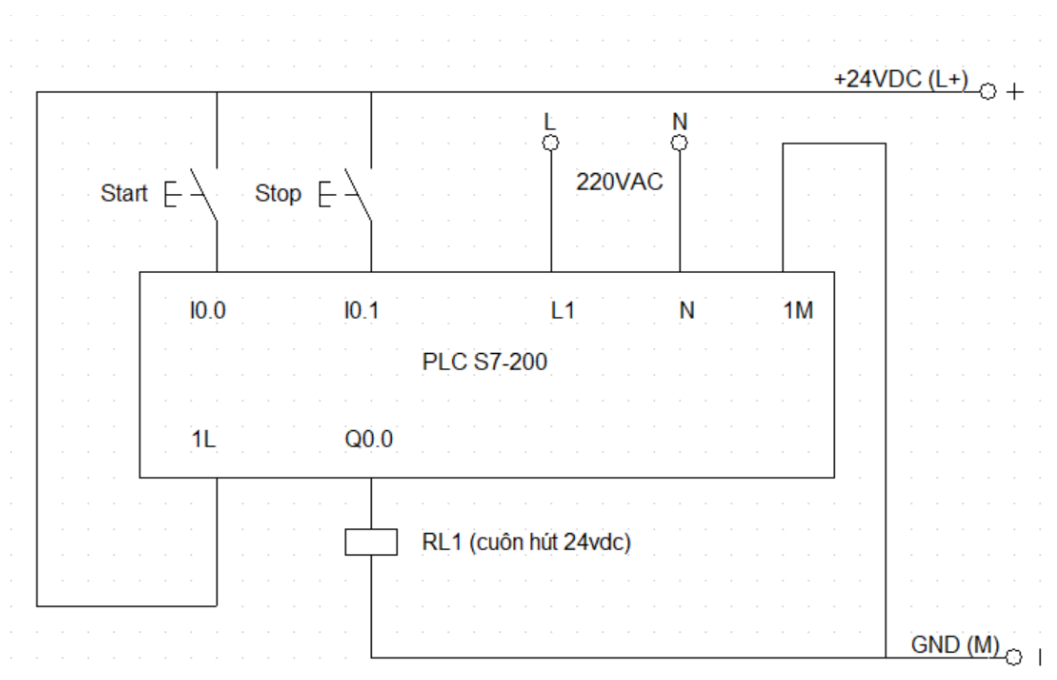
- Gá lắp các thiết bị lên bàn thực hành sau đó kết nối dây theo sơ đồ

Bước 2: Lập bảng quy định địa chỉ vào/ra:

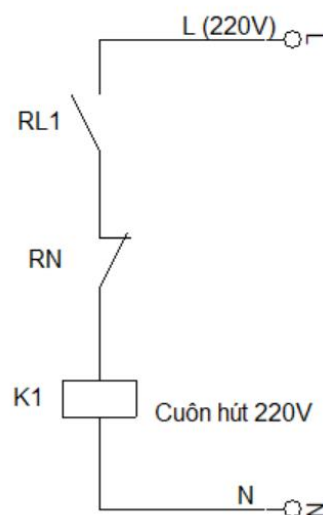
Tên	Địa chỉ	Ghi Chú
Start	I0.0	Nút nhấn khởi động động cơ
Stop	I0.1	Nút dừng động cơ
RL1	Q0.0	Role RL1 cuộn hút 24VDC dùng đóng cắt Contactor K1 cuộn hút 220VAC

Bước 3: Sơ đồ kết nối

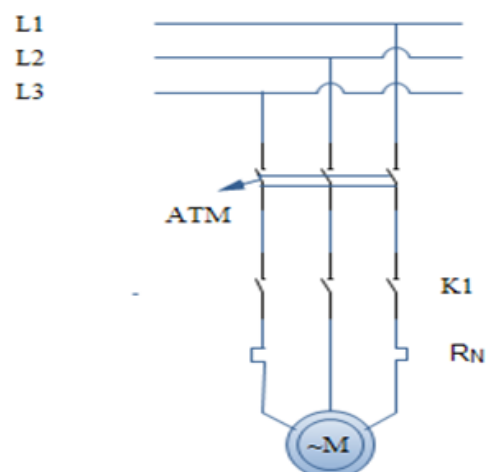
- Sơ đồ kết nối PLC



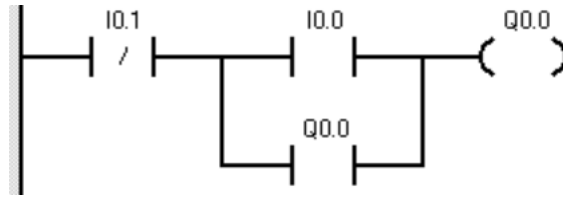
- Sơ đồ kết nối contactor



Sơ đồ mạch động lực: Động cơ làm việc có thể chạy sao hoặc tam giác ở đây trong trường hợp này chọn phương án đấu sao



Bước 4: Soạn thảo chương trình PLC



Bước 5: Vận hành

Hướng dẫn soạn thảo chương trình PLC trên máy tính:

- Khởi động chương trình:

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



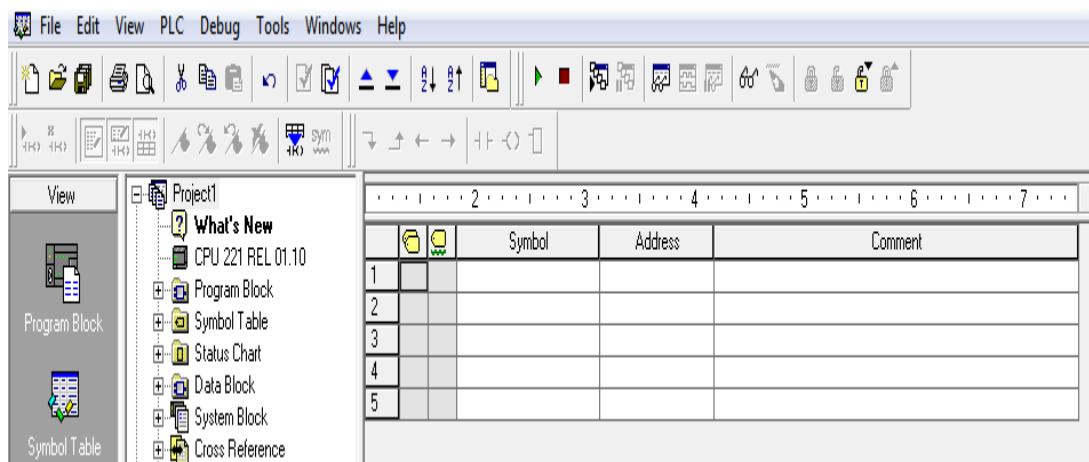
Giao diện khởi động chương trình

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



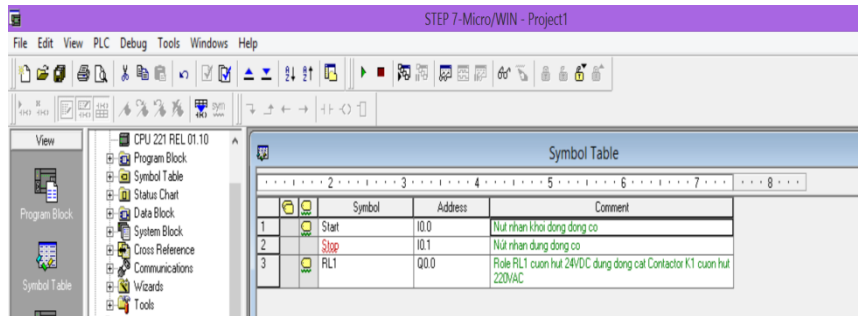
Giao diện khởi động chương trình

- **Khai báo bảng địa chỉ vào/ra :** Khi khởi động màn hình soạn thảo sẽ xuất hiện như hình click chuột vào biểu tượng Symbol Table



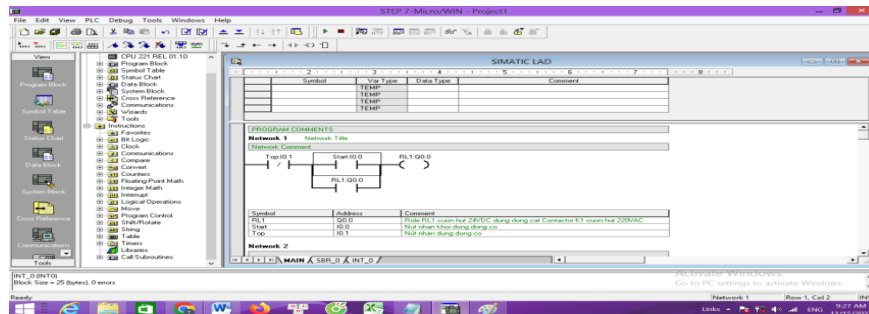
Giao diện Symbol Table

Nhập bảng quy định địa chỉ vào/ra



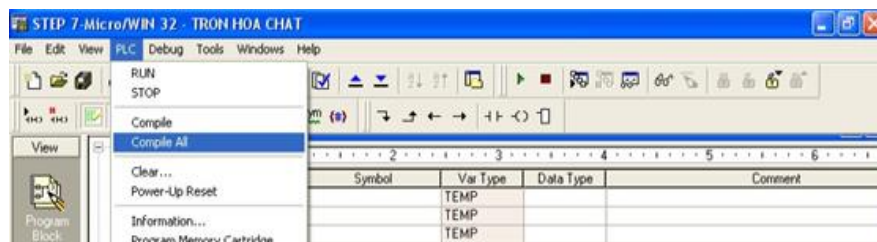
Nhập bảng quy định địa chỉ

- **Soạn thảo chương trình PLC:** Click chuột vào biểu tượng Program Block  giao diện màn hình soạn thảo hiện ra sau đó ta tiến hành soạn thảo chương trình theo yêu cầu (ngôn ngữ dạng LAD)



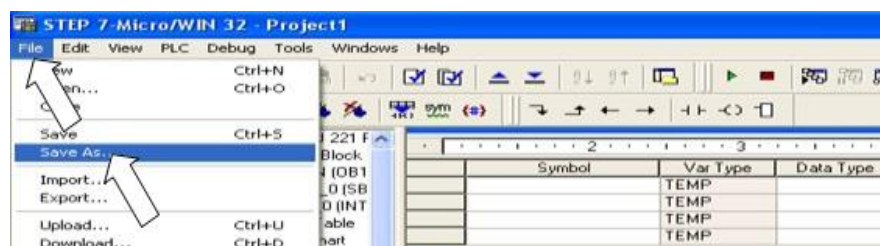
Giao diện soạn thảo

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



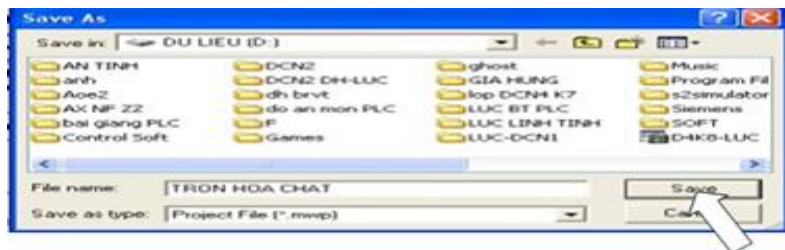
Giao diện kiểm tra

- Lưu chương trình:** Sau khi soạn thảo chương trình xong và đã kiểm tra lỗi, chương trình đúng yêu cầu ta vào File/ Save As để lưu chương trình như hình



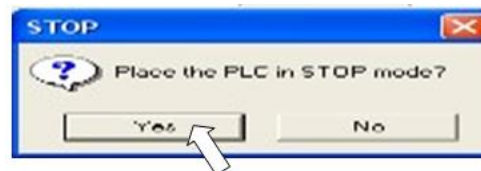
Giao diện lưu chương trình

Trong hộp thoại Save As ta chọn tên ổ đĩa, chọn Folder(thư mục), đánh tên File. Ví dụ tên File là “Hoten_dong cat dong co” được lưu trong folder(thư mục) “Dien cong nghiep K39A1: chẳng hạn, sau đó chọn Save để lưu như hình



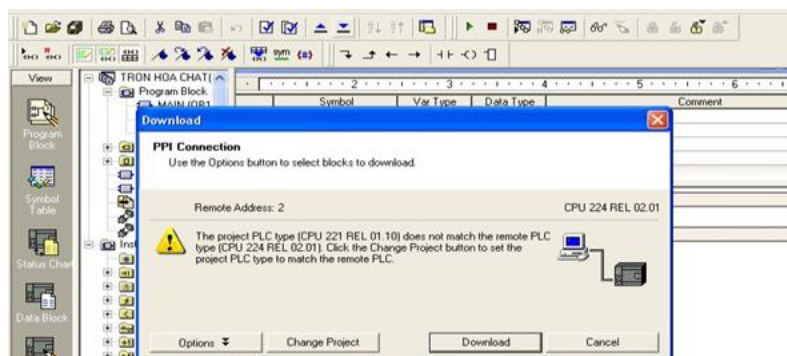
Giao diện lưu chương trình đóng cắt động cơ

Dừng chương trình: Trước khi tải chương trình từ máy tính xuống PLC ta cần đưa PLC về chế độ Stop. Muốn Dừng chương trình nhấp vào PLC→STOP hoặc biểu tượng  hộp thoại xuất hiện ta chọn Yes, như hình



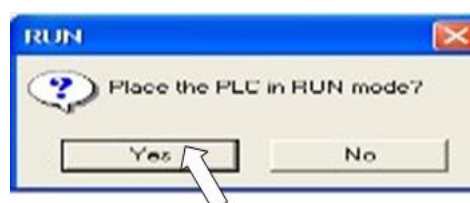
Thiết lập chế độ STOP

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



Thiết lập download

- **Chạy chương trình:** Khi đã Download chương trình thành công ta nhấp vào PLC→RUN hoặc biểu tượng , hộp thoại xuất hiện ta chọn Yes



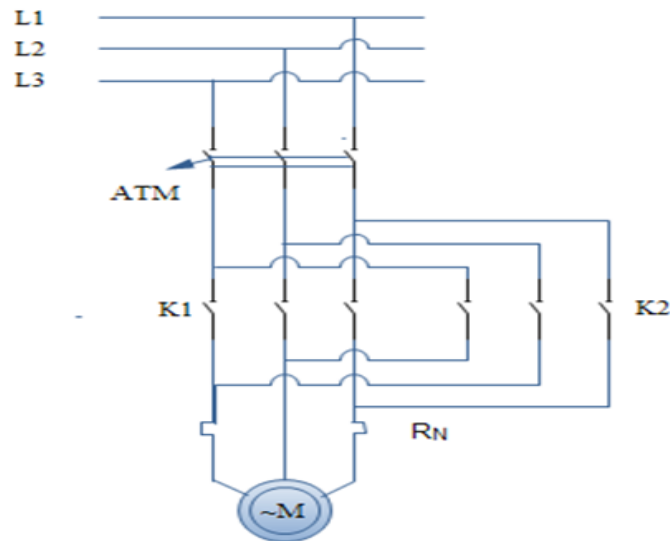
Thiết lập chế độ RUN

2. Điều khiển đảo chiều quay trực tiếp động cơ

2.1. Yêu cầu công nghệ

Lập trình PLC S7-200 điều khiển động cơ không đồng bộ ba pha roto lồng sóc đảo chiều trực tiếp đảm bảo các yêu cầu sau:

- Nhấn nút MT(**Mở Thuận**) động cơ chạy theo chiều thuận, nhấn nút MN(**Mở Ngược**) động cơ chạy theo chiều ngược.
- Khi động cơ đang chạy muốn dừng động cơ nhấn nút D (**Dừng**).
- Khi động cơ đang chạy ngược, nhấn nút MT thì động cơ sẽ chuyển sang chạy thuận và ngược lại



2.2. Các bước tiến hành

Bước 1: Chuẩn bị dụng cụ, vật tư, bàn thực hành



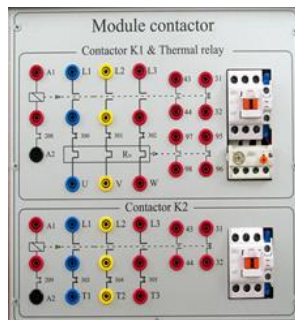
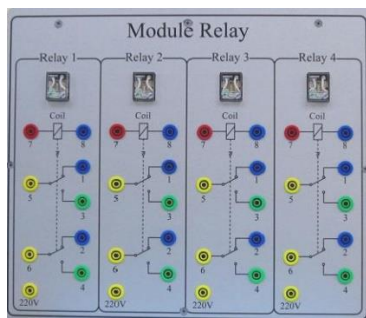
Module nguồn



Module nút nhấn



Module PLC



Module Re lay



Dây dẫn

Module công tắc tơ K1, K2



Bàn thực hành gá module

Động cơ



Bộ dụng cụ

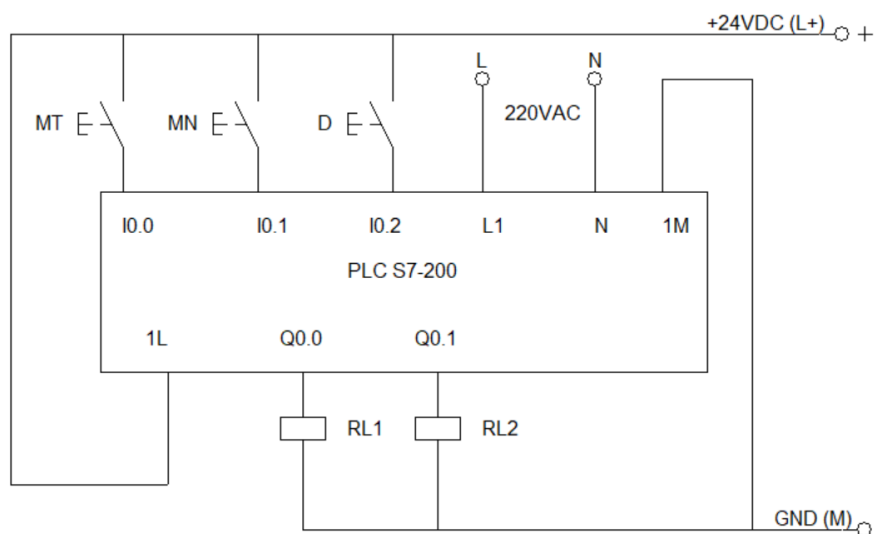
- Gá lắp các thiết bị lên bàn thực hành sau đó kết nối dây theo sơ đồ

Bước 2: Lập bảng quy định địa chỉ vào/ra

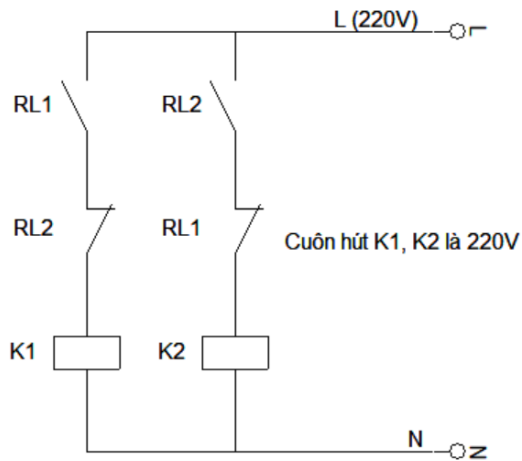
Tên	Địa chỉ	Ghi Chú
MT	I0.0	Nút nhấn MT điều khiển động cơ chạy thuận
MN	I0.1	Nút nhấn MN điều khiển động cơ chạy ngược
D	I0.2	Nút nhấn D để dừng động cơ
RL1	Q0.0	Role 1 cuộn hút 24VDC dùng để đóng cắt contactor K1 cuộn hút 220VAC
RL2	Q0.3	Role 2 cuộn hút 24VDC dùng để đóng cắt contactor K2 cuộn hút 220VAC

Bước 3: Sơ đồ kết nối

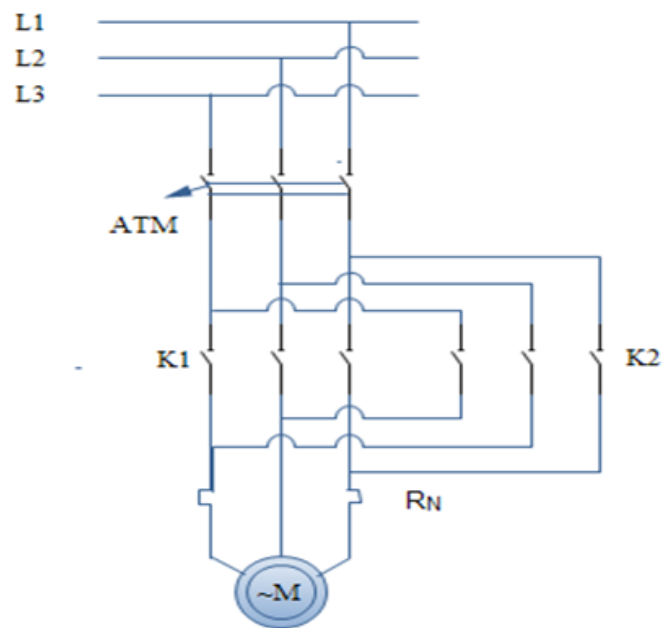
- Sơ đồ kết nối PLC



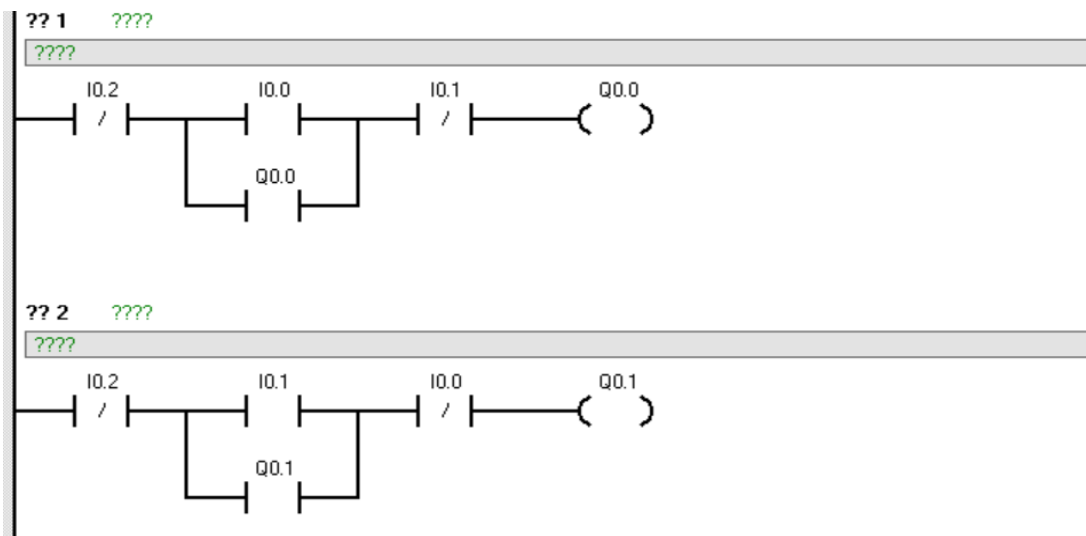
- Sơ đồ kết nối Contactor



- Sơ đồ mạch động lực:



Bước 4: Soạn thảo chương trình PLC(xem hướng dẫn bài điều khiển đóng cắt động cơ)



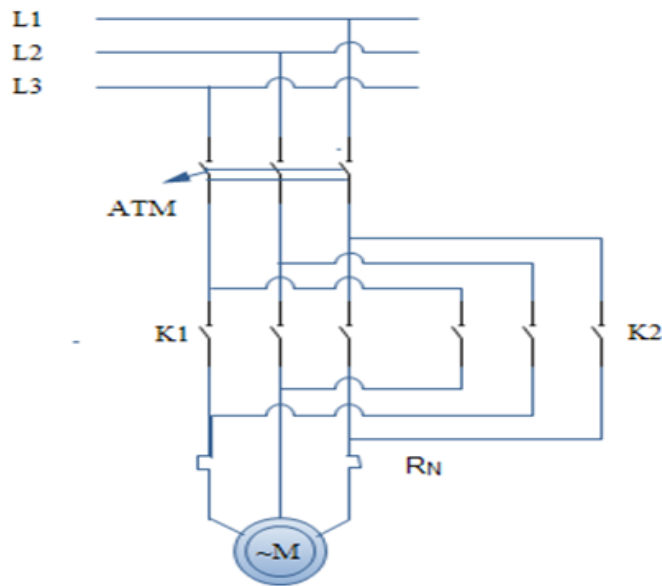
Bước 5: Vận hành

3. Điều khiển đảo chiều quay gián tiếp động cơ

3.1. Yêu cầu công nghệ

Lập trình PLC S7-200 điều khiển động cơ không đồng bộ ba pha roto lồng sóc đảo chiều gián tiếp đảm bảo các yêu cầu sau:

- Nhấn nút MT (**Mở Thuận**) động cơ chạy theo chiều thuận, nhấn nút MN (**Mở Ngược**) động cơ chạy theo chiều ngược.
- Khi động cơ đang chạy muốn dừng động cơ nhấn nút D (**Dừng**).
- Khi động cơ đang chạy ngược, muốn chuyển sang chạy thuận phải ấn nút **Dừng** rồi ấn nút Mở thuận và ngược lại.



3.2. Các bước tiến hành

Bước 1: Chuẩn bị dụng cụ, vật tư, bàn thực hành



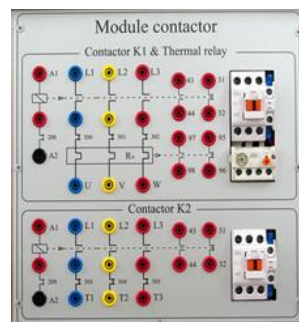
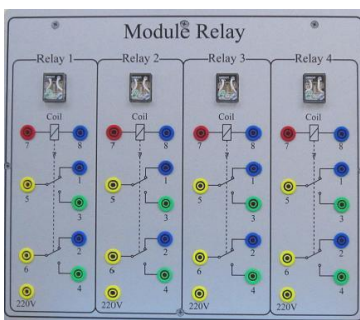
Module nguồn



Module nút nhấn



Module PLC



Module Re lay



Dây dẫn

Module công tắc tơ K1, K2



Bàn thực hành gá module

Động cơ



Bộ dụng cụ

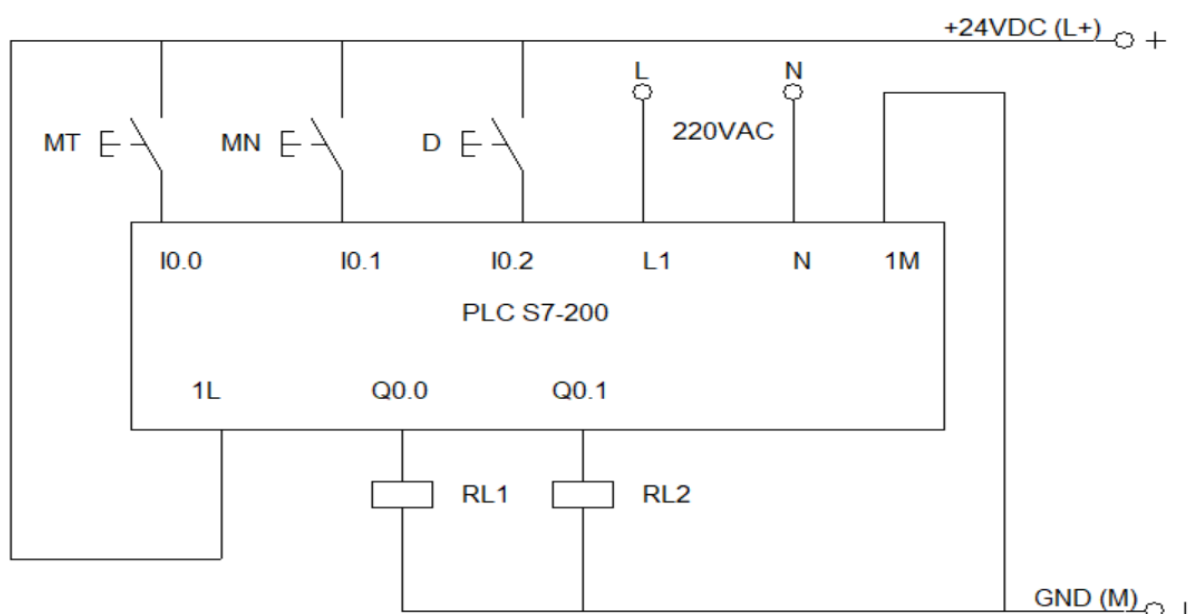
- Gá lắp các thiết bị lên bàn thực hành sau đó kết nối dây theo sơ đồ

Bước 2: Lập bảng quy định địa chỉ vào/ra

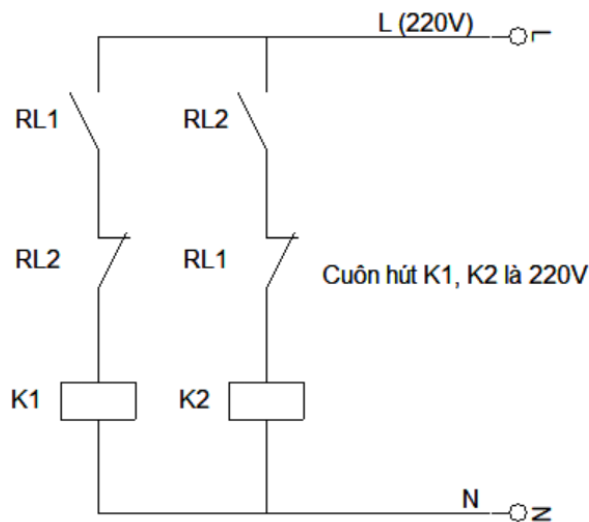
Tên	Địa chỉ	Ghi Chú
MT	I0.0	Nút nhấn MT điều khiển động cơ chạy thuận
MN	I0.1	Nút nhấn MN điều khiển động cơ chạy ngược
D	I0.2	Nút nhấn D để dừng động cơ
RL1	Q0.0	Role 1 cuộn hút 24VDC dùng để đóng cắt contactor K1 cuộn hút 220VAC
RL2	Q0.3	Role 2 cuộn hút 24VDC dùng để đóng cắt contactor K2 cuộn hút 220VAC

Bước 3: Sơ đồ kết nối

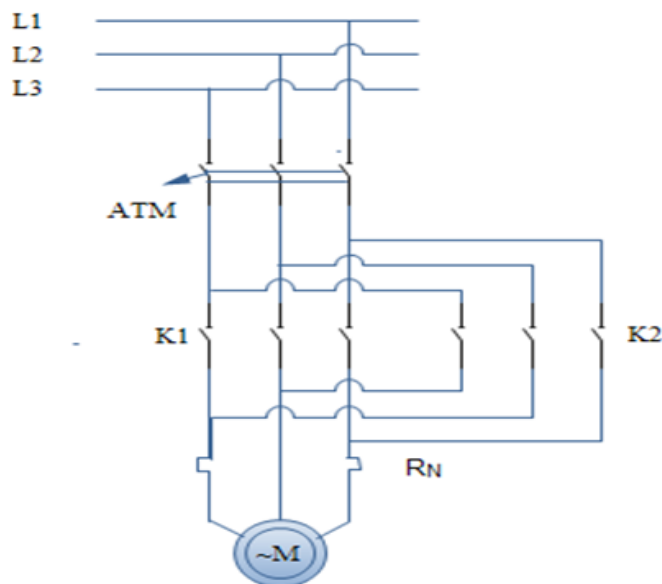
- Sơ đồ kết nối PLC



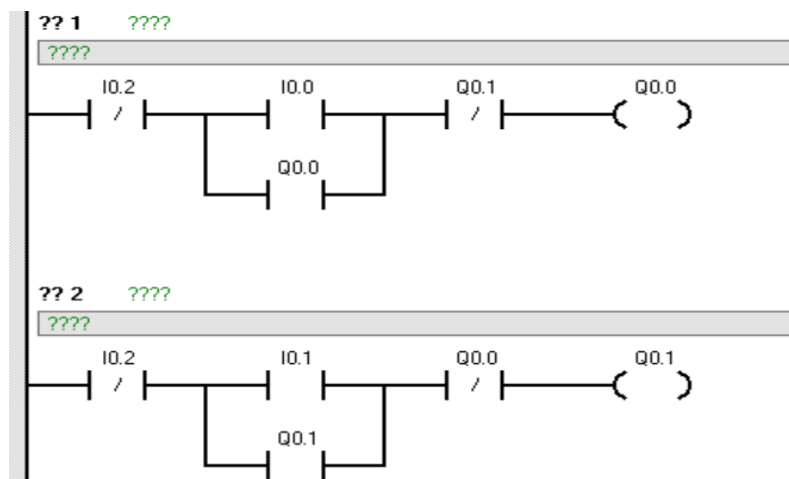
- Sơ đồ kết nối contactor



- Sơ đồ mạch động lực: Động cơ làm việc có thể chạy sao hoặc tam giác ở đây trong trường hợp này chọn phương án đấu sao



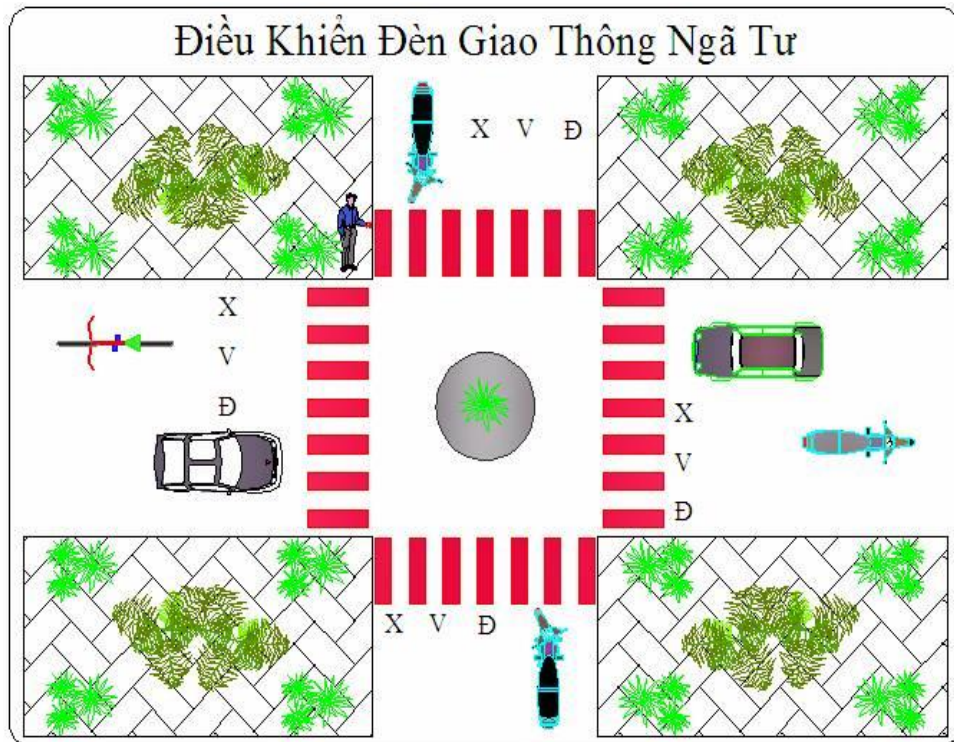
Bước 4: Soạn thảo chương trình PLC(xem hướng dẫn bài điều khiển đóng cắt động cơ)



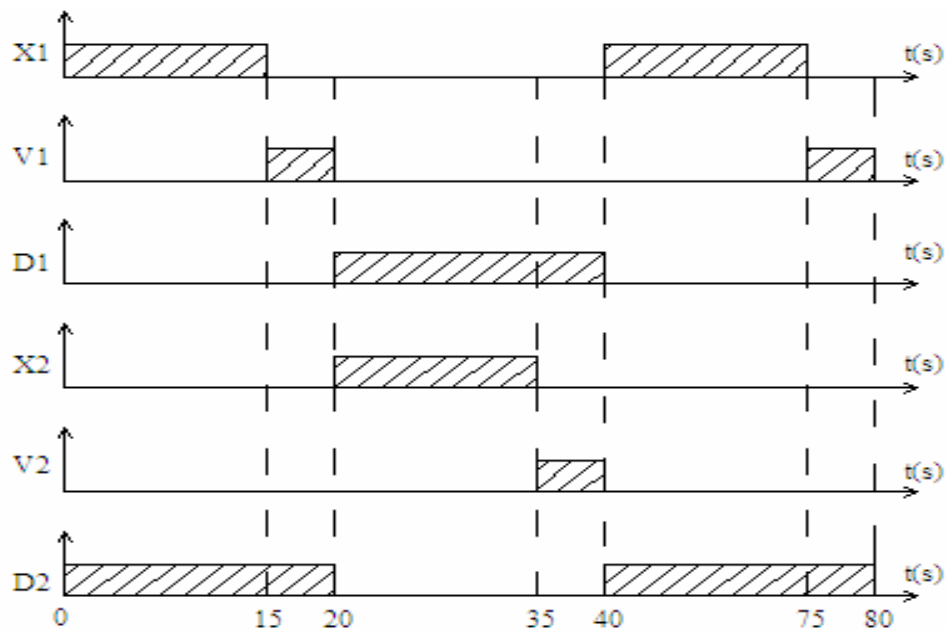
Bước 5: Vận hành

4. Điều khiển đèn giao thông

4.1. Yêu cầu công nghệ



Giản đồ thời gian



Lập trình PLC s7-200 cho một hệ thống điều khiển đèn giao thông ở một ngã tư đảm bảo yêu cầu sau:

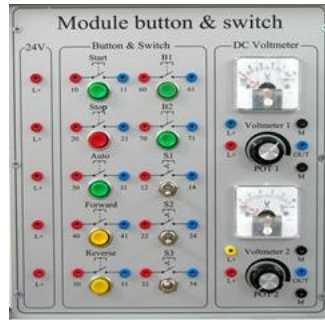
- Khi bật công tắc nguồn hệ thống bắt đầu hoạt động. Đèn Xanh sáng 15s sau đó chuyển sang đèn Vàng sáng 5s, tiếp đó đèn Đỏ sáng 20s. Chu trình lặp đi lặp lại.

4.2. Các bước tiến hành

Bước 1: Chuẩn bị dụng cụ, vật tư, bàn thực hành



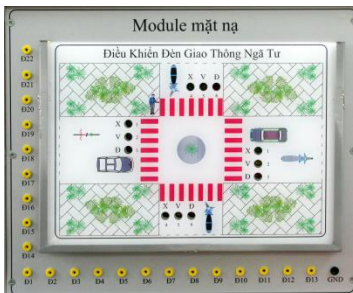
Module nguồn



Module nút nhấn



Module PLC



Module đèn giao thông



Dây dẫn



Bàn thực hành gá module



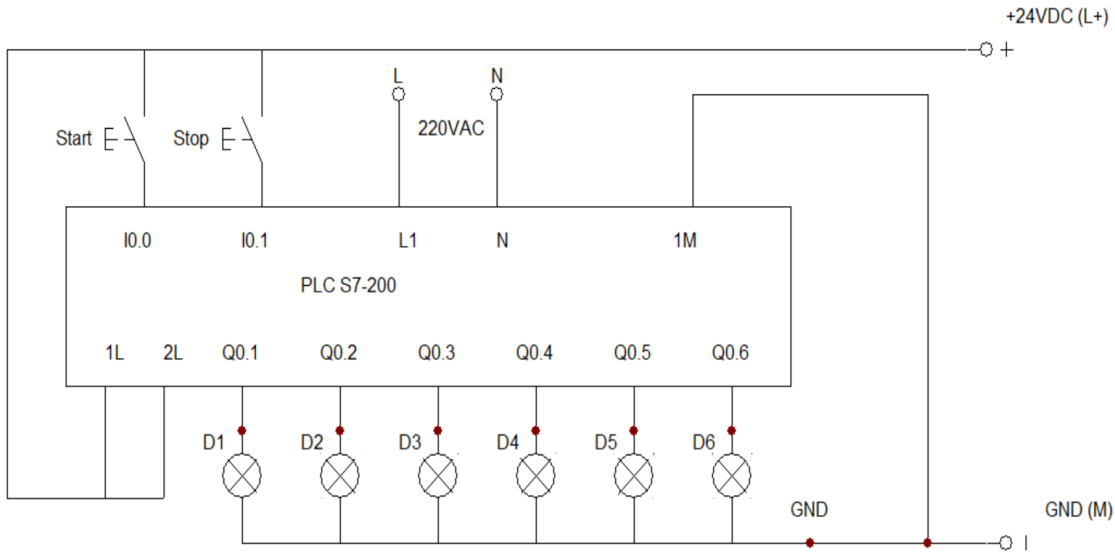
Bộ dụng cụ

- Gá lắp các thiết bị lên bàn thực hành sau đó kết nối dây theo sơ đồ

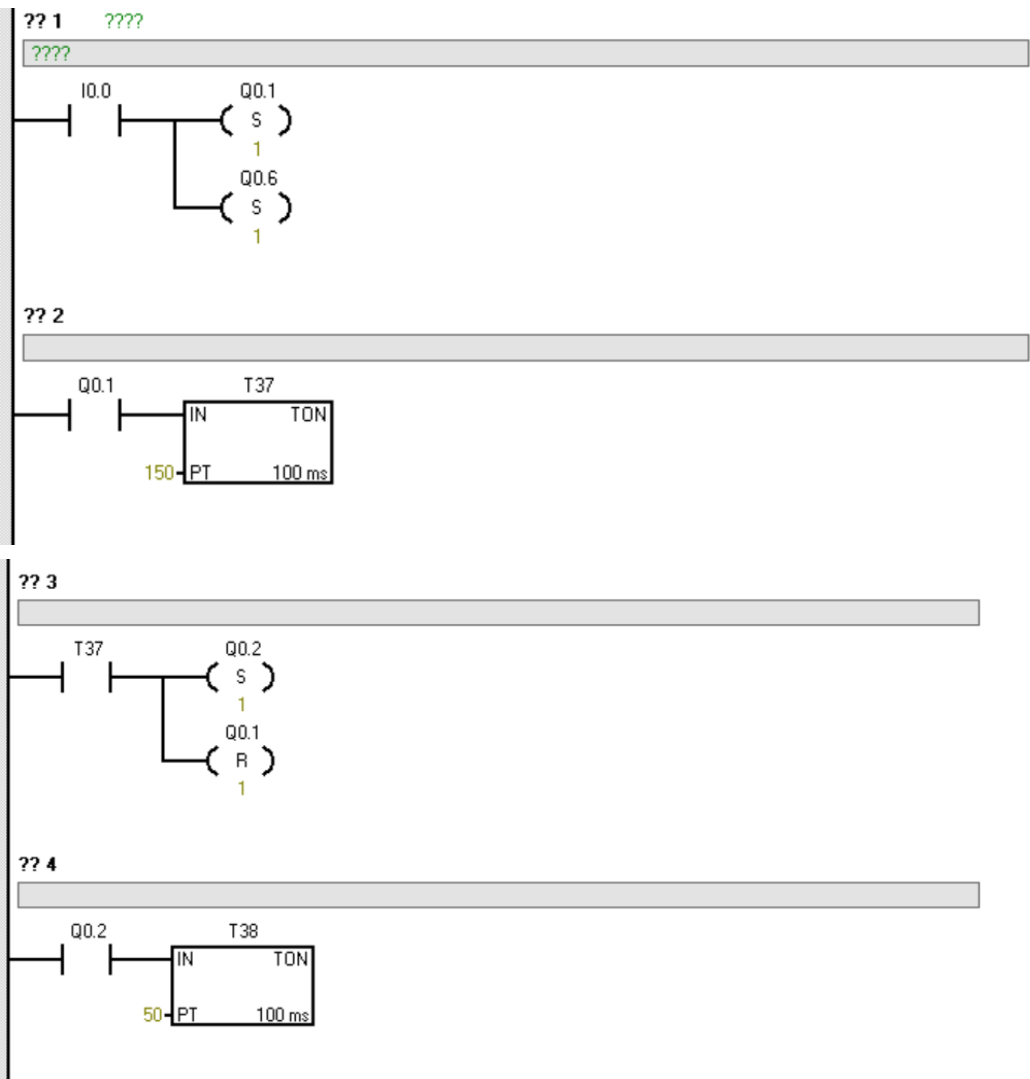
Bước 2: Bảng quy định địa chỉ vào/ra.

Tên	Địa chỉ	Ghi Chú
Xanh 1	Q0.1	Đèn xanh 1
Vàng 1	Q0.2	Đèn vàng 1
Đỏ 1	Q0.3	Đèn đỏ 1
Xanh 2	Q0.4	Đèn xanh 2
Vàng 2	Q0.5	Đèn vàng 2
Đỏ 2	Q0.6	Đèn đỏ 2

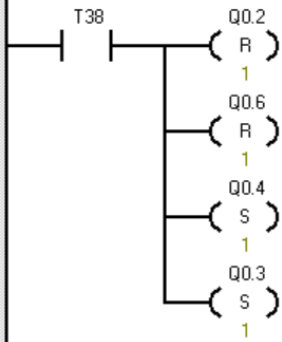
Bước 3: Sơ đồ kết nối



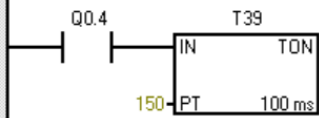
Bước 4: Soạn thảo chương trình PLC(xem hướng dẫn bài điều khiển đóng cắt động cơ)



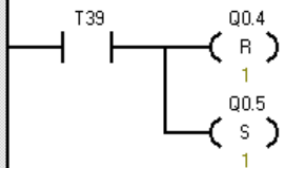
?? 5



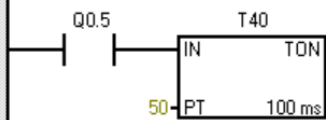
?? 6



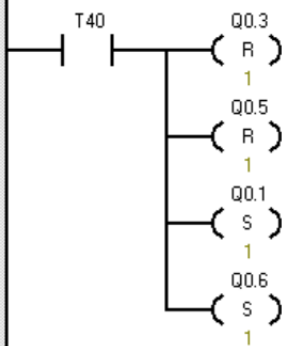
?? 7



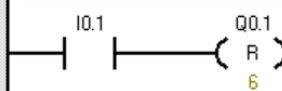
?? 8



?? 9



?? 10



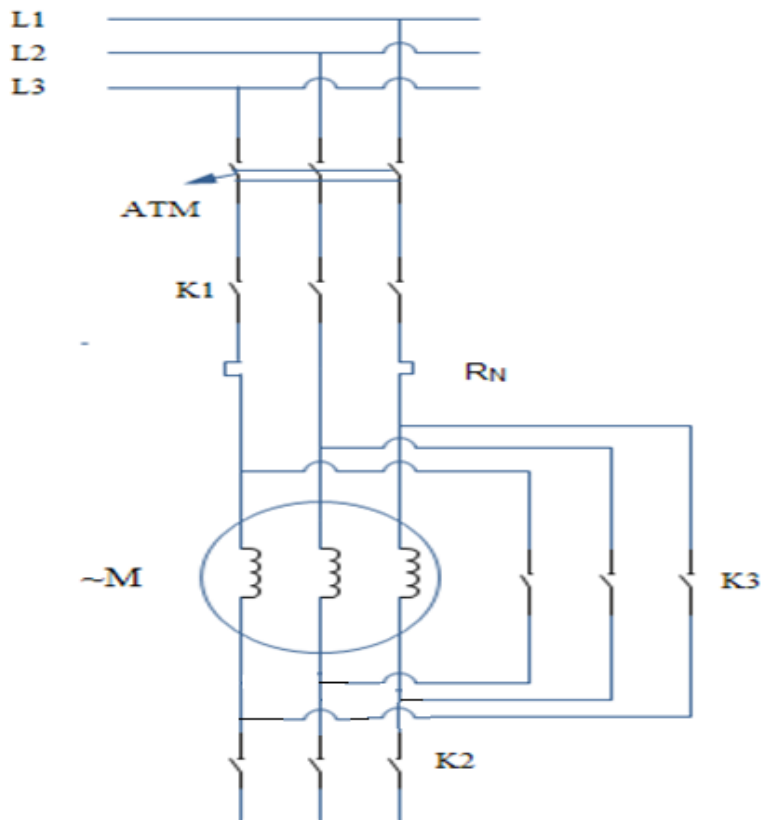
Bước 5: Vận hành

5. Điều khiển khởi động động cơ chế độ sao – tam giác

5.1. Yêu cầu công nghệ

Lập trình PLC S7-200 điều khiển động cơ không đồng bộ ba pha roto lồng sóc đảm bảo các yêu cầu sau:

- Nhấn nút Start động cơ khởi động chế độ sao(K1 và K2 đóng), sau 5s động cơ chạy ở chế độ tam giác(K1 và K3 đóng).
- Nhấn nút Stop động cơ dừng lại(K1, K2, K3 mở ra).



5.2. Các bước tiến hành

Bước 1: Chuẩn bị dụng cụ, vật tư, bàn thực hành



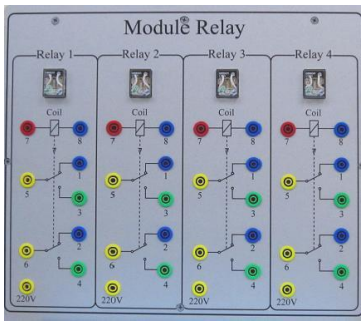
Module nguồn



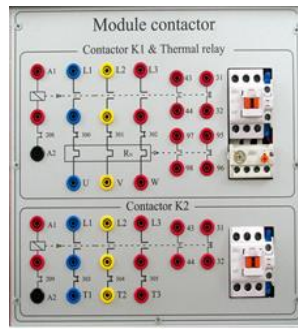
Module nút nhấn



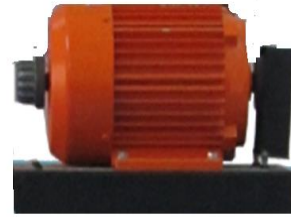
Module PLC



Module Re lay



Module công tắc tơ K1,K2



Động cơ



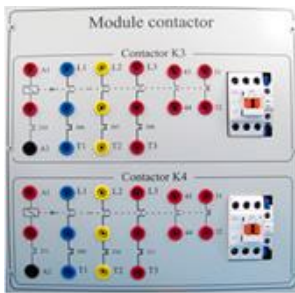
Dây dẫn



Bàn thực hành gá module



Bộ dụng cụ



Module công tắc tơ K3,K4

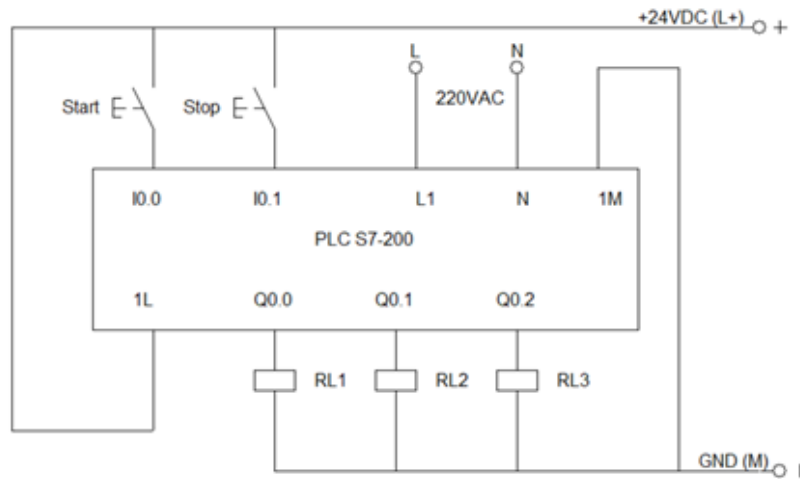
- Gá lắp các thiết bị lên bàn thực hành sau đó kết nối dây theo sơ đồ

Bước 2: Lập bảng quy định địa chỉ vào/ ra

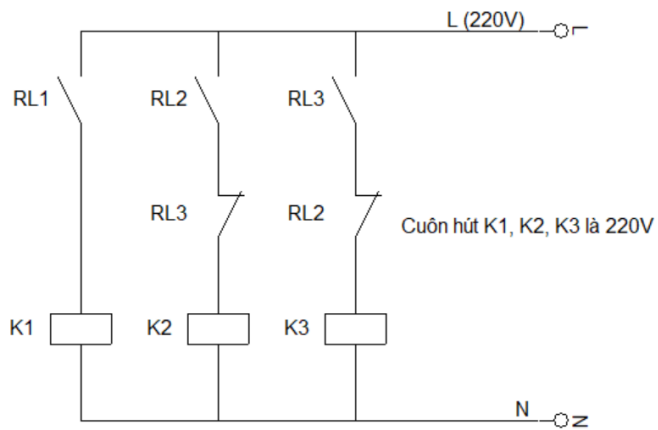
Tên	Địa chỉ	Ghi Chú
Start	I0.0	Nút khởi động động cơ
Stop	I0.1	Nút dừng động cơ
RL1	Q0.0	Rơ le RL1 cuộn hút 24VDC dùng để đóng cắt contactor K1 cuộn hút 220VAC
RL2	Q0.2	Rơ le RL2 cuộn hút 24VDC dùng để đóng cắt contactor K2 cuộn hút 220VAC
RL3	Q0.3	Rơ le RL3 cuộn hút 24VDC dùng để đóng cắt contactor K3 cuộn hút 220VAC

Bước 3: Sơ đồ kết nối

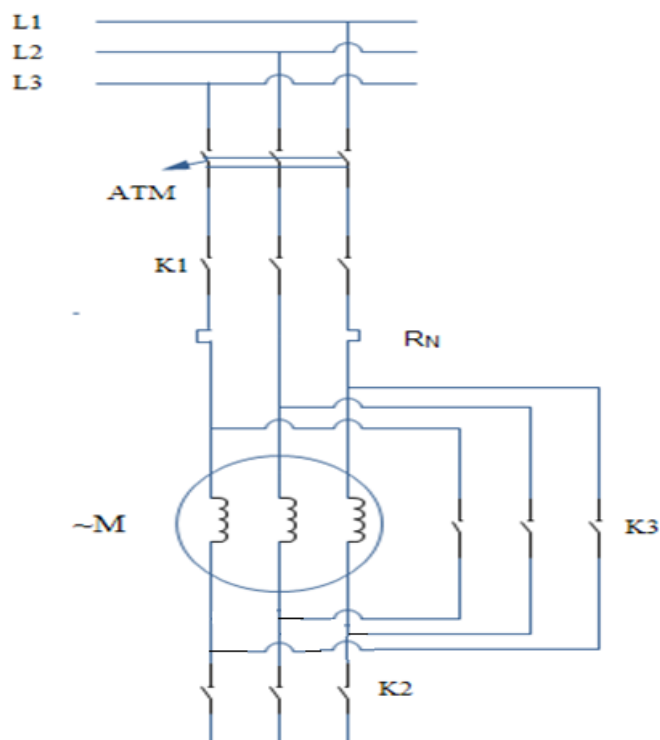
- Sơ đồ kết nối PLC



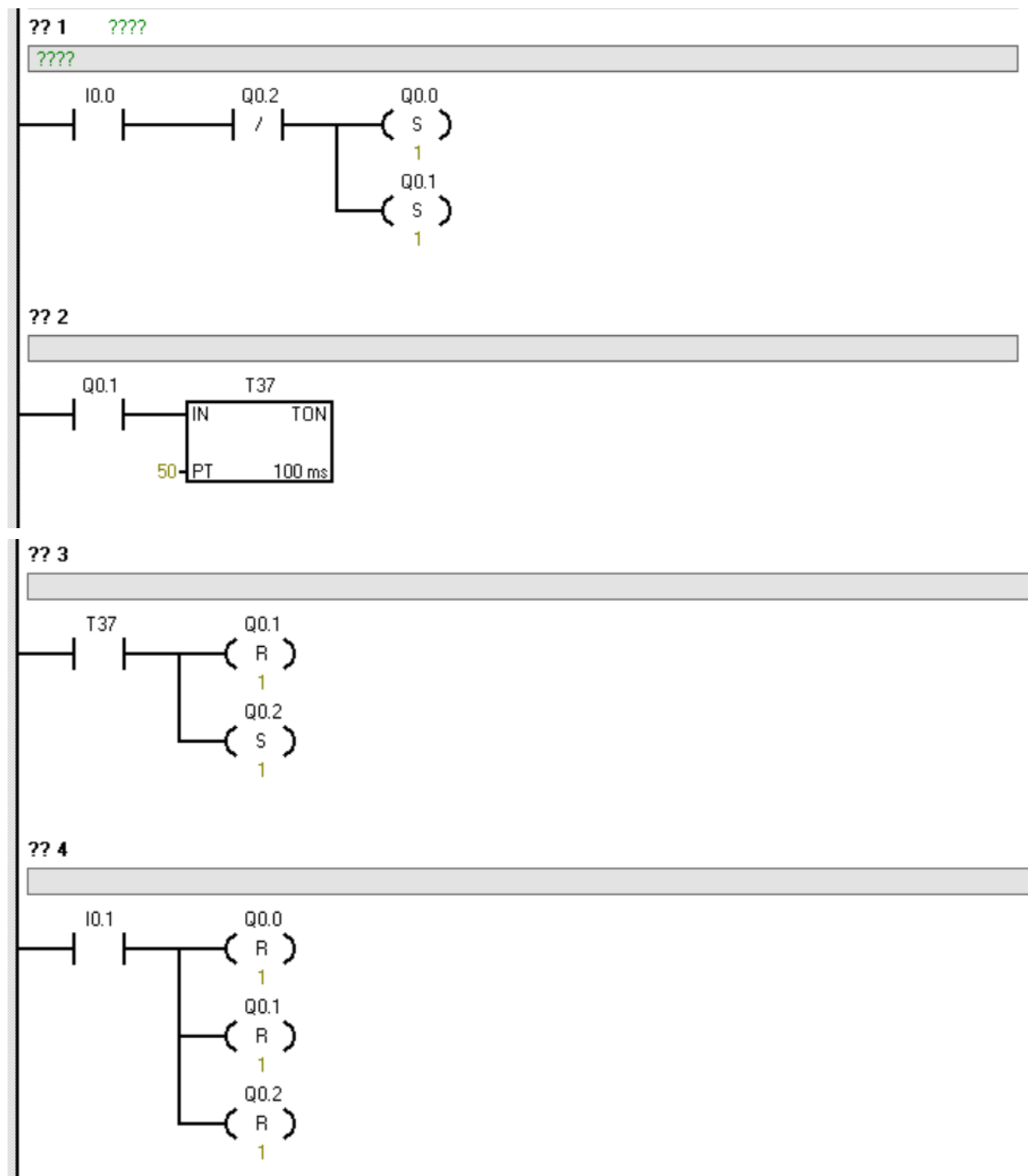
- Sơ đồ kết nối Contactor



- Sơ đồ mạch động lực:



Bước 4: Soạn thảo chương trình PLC(xem hướng dẫn bài điều khiển đóng cắt động cơ)



Bước 5: Vận hành

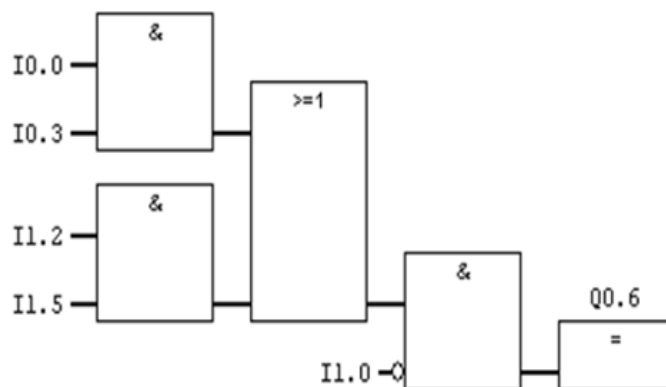
BÀI 5: CÁC CHỨC NĂNG CƠ BẢN CỦA LOGO!

Mã bài: MĐ 20 - 05

Giới thiệu:

PLC Logo là một bộ lập trình cỡ nhỏ cung cấp cho người dùng 2 nhóm lệnh đó là nhóm lệnh chức năng cơ bản và nhóm lệnh chức năng đặc biệt, với người mới bắt đầu hay trong các ứng dụng đơn giản người lập trình có thể chỉ cần sử dụng đến các câu lệnh chức năng cơ bản để giải quyết bài toán lập trình.

Trong Logo người ta dùng các khối kí hiệu cho các chức năng khác nhau, tương tự sơ đồ logic trong mạch số hay trang bị điện không tiếp điểm. Cách này được viết tắt là CSF (Control System Flowchart: Lưu đồ hệ thống điều khiển) hay FBD (Function Block Diagram: Sơ đồ khối chức năng).



Hình 5.1. Sơ đồ biểu diễn kiểu FBD

Để lập trình cho Logo phải sử dụng các đầu nối ở ngõ vào, các chức năng cơ bản, các chức năng đặc biệt. Trong bài học này giúp cho người học bước đầu tiếp cận với các câu lệnh cơ bản để viết chương trình.

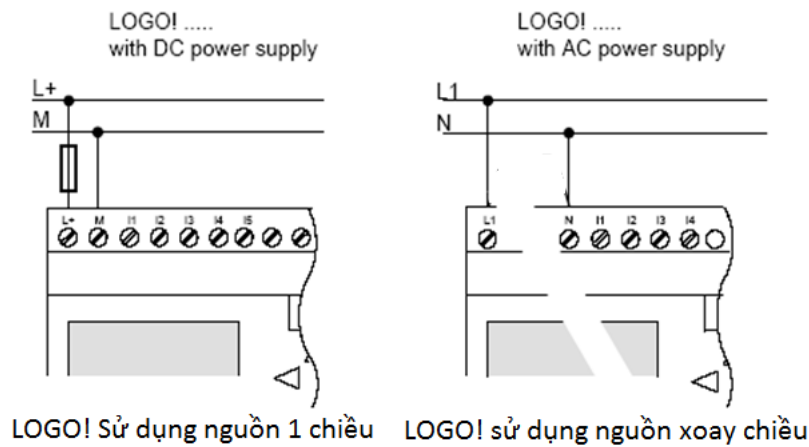
Mục tiêu:

- Sử dụng, khai thác đúng chức năng các hàm cơ bản của LOGO!.
- Viết các chương trình ứng dụng các hàm cơ bản theo từng yêu cầu cụ thể.
- Rèn luyện tính cẩn thận, tỉ mỉ, chính xác, tư duy khoa học và sáng tạo.

Nội dung chính:

1. Kết nối Logo với thiết bị ngoại vi

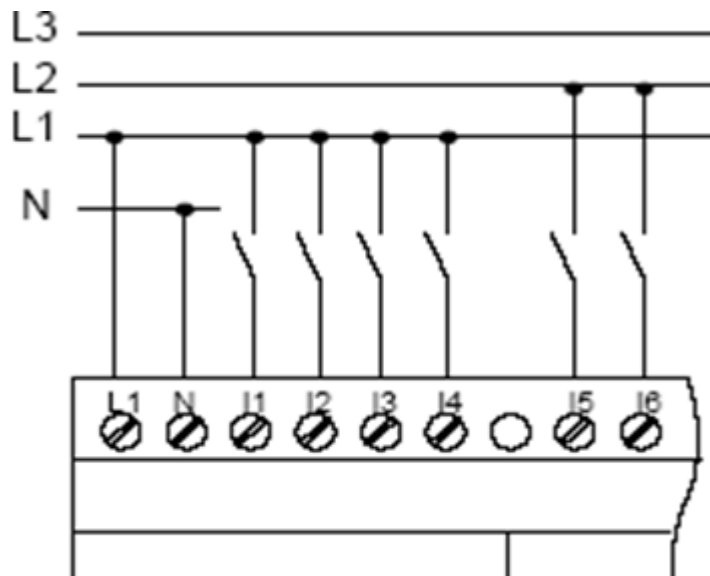
1.1. Nối nguồn cung cấp



Hình 5.2. Nối nguồn cung cấp

1.2. Kết nối ngõ vào

Kết nối đầu vào LOGO! 230...



Hình 5.3: Kết nối đầu vào Logo! 230

Việc đi dây cho các đầu vào được chia thành hai nhóm, mỗi nhóm 4 ngõ vào. Các đầu vào trong cùng một nhóm chỉ có thể cấp cùng một pha điện áp. Các đầu vào trong hai nhóm có thể cấp cùng pha hoặc khác pha điện áp. VD I1 đến I4 nối đến pha 1 (L1) và I5 đến I8 nối đến pha 2 (L2)

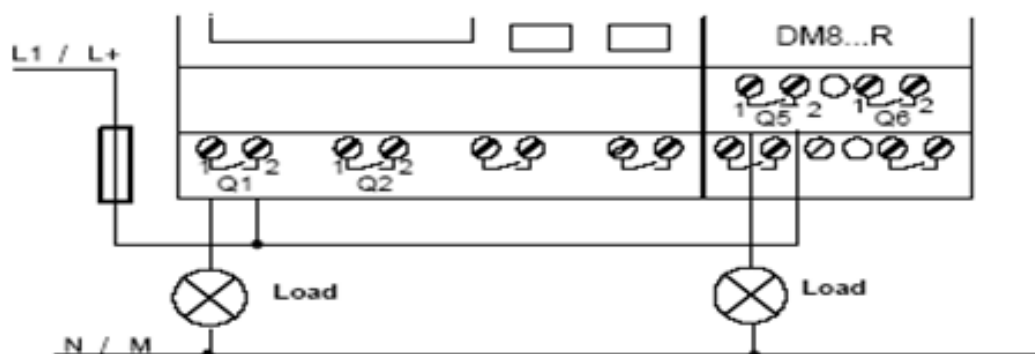
Các ngõ vào của LOGO! DM8 230R không được kết nối khác pha nhau.

1.3. Kết nối ngõ ra

Đối với ngõ ra dạng relay

Ta có thể kết nối nhiều dạng tải khác nhau vào ngõ ra. Ví dụ: đèn, motor, contactor, relay

Tải thuần trở: tối đa 10A. Tải cảm: tối đa 3A. Sơ đồ kết nối như sau:



Hình 5.4. Sơ đồ kết nối ngõ ra Relay

2. Các loại hàm trong LOGO!

Các hàm lập trình trong LOGO được chia thành 4 danh sách sau đây:

Tên hàm	Chức năng
Co	danh sách các điểm liên kết (bit M, các ngõ input, output...), các hằng số.
GF	danh sách các hàm cơ bản như AND, OR...
SF	danh sách các hàm đặc biệt.
BN	danh sách các block đã được sử dụng trong sơ đồ mạch.

Bảng 5.1: Các loại hàm trong Logo

3. Danh sách Co

3.1. Ngõ vào số

Ngõ vào số được xác định bởi ký tự bắt đầu là I. Số thứ tự của các ngõ vào (I1, I2, ...) tương ứng với ngõ vào kết nối trên LOGO.

3.2. Ngõ ra số

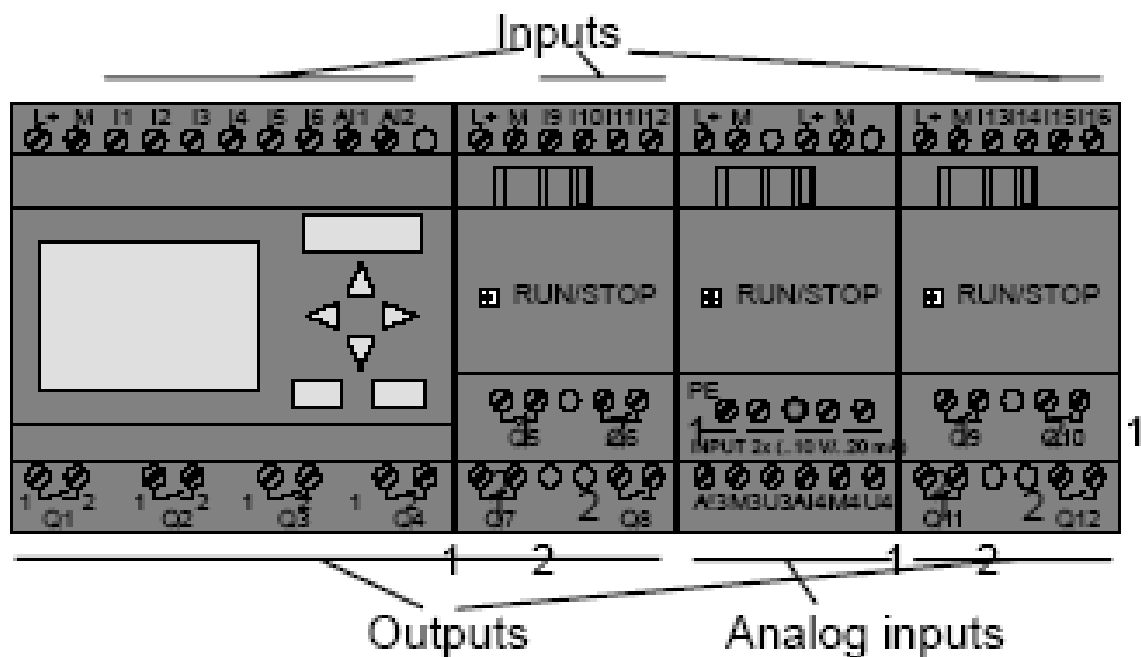
Ngõ ra số được xác định bởi ký tự bắt đầu là Q (Q1, Q2, ... Q16).

3.3. Ngõ vào analog

Đối với các version LOGO! 24, LOGO! 24o, LOGO! 12/24RC và LOGO! 12/24Rco, các ngõ vào I7, I8 có thể được lập trình để sử dụng như hai kênh vào analog AI1, AI2.

3.4. Ngõ ra analog

Ngõ ra analog được bắt đầu bởi ký tự AQ, LOGO chỉ cho phép tối đa 2 ngõ ra analog là AQ1 và AQ2



Hình 5.5. Kết nối Module mở rộng

3.5. Cờ Start up

Trong LOGO, bit M8 tự động được set lên 1 trong chu kỳ quét đầu tiên. Vì vậy, ta có thể sử dụng bit này như 1 cờ Start up. Sau chu kỳ quét đầu tiên, bit M8 sẽ được reset về 0. Ngoài ra, bit M8 cũng có thể được sử dụng như một bit nhớ thông thường trong chương trình.

3.6. Thanh ghi dịch bit

LOGO! cung cấp 8 thanh ghi dịch bit từ S1 đến S8. Đây là các thanh ghi chỉ đọc. Nội dung của thanh ghi dịch bit chỉ có thể được định nghĩa lại bằng hàm đặc biệt (SF) “shift register”.

3.7. Mức hằng số

Mức tín hiệu được thiết kế ở 2 mức: **Hi** và **Lo** với:

Hi = 1: mức cao

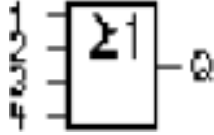
Lo = 0: mức thấp.

3.8. Không kết nối

Các kết nối không sử dụng có thể được định nghĩa bởi gán x

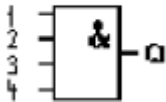
4. Các hàm sử dụng trong Logo!

4.1. Hàm OR

Ký hiệu	Bảng Logic cổng OR				
 Ngõ ra bằng 1 nếu có ít nhất một ngõ vào bằng 1 Ngõ vào không sử dụng ta có thể dùng ký hiệu x (x=0)	1	2	3	4	Q
	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	1
	0	0	1	1	1
	0	1	0	0	1
	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	1
	1	0	0	0	1
	1	0	0	1	1
	1	0	1	0	1
	1	0	1	1	1
	1	1	0	0	1
	1	1	0	1	1
	1	1	1	0	1
	1	1	1	1	1

Bảng 5.2. Bảng logic cổng OR

4.2. Hàm AND

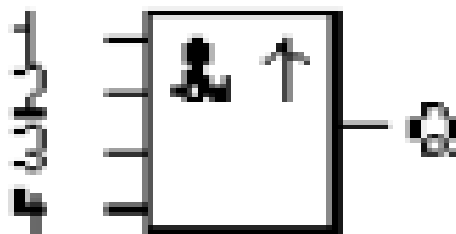
Ký hiệu	Bảng Logic cổng AND				
Cổng AND:  Ngõ ra của hàm AND bằng 1 khi tất cả các ngõ vào bằng 1.	1	2	3	4	Q
	0	0	0	0	0
	0	0	0	1	0
	0	0	1	0	0
	0	0	1	1	0
	0	1	0	0	0
	0	1	0	1	0
	0	1	1	0	0
	0	1	1	1	0
	1	0	0	0	0
	1	0	0	1	0
	1	0	1	0	0

	1	0	0	1	0
	1	0	1	0	0
	1	0	1	1	0
	1	1	0	0	0
	1	1	0	1	0
	1	1	1	0	0
	1	1	1	1	1

Bảng 5.3. Bảng logic cổng AND

Cổng AND lấy cạnh xung lên:

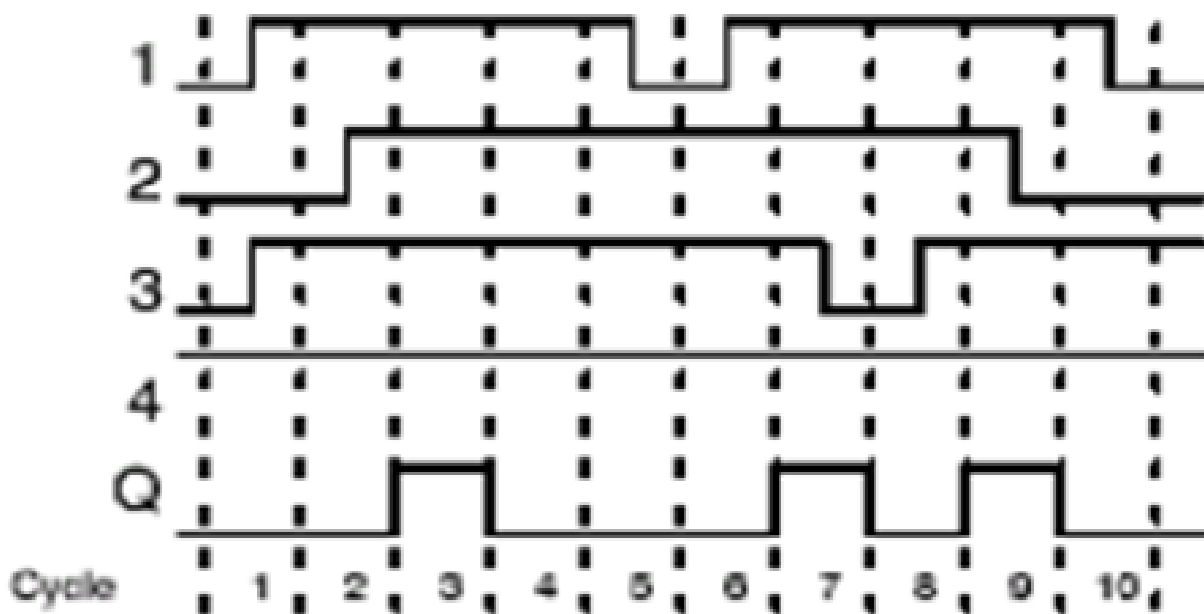
- Ký hiệu:



Hình 5.6. Ký hiệu cổng AND lấy cạnh xung lên

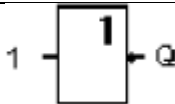
Ngõ ra bằng 1 trong 1 chu kỳ quét tại thời điểm đầu tiên mà cả 4 ngõ vào cùng bằng 1.

Ngõ vào không sử dụng ta có thể sử dụng ký hiệu x ($x=1$).



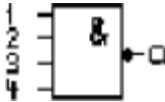
Hình 5.7. Giản đồ thời gian

4.3. Hàm NOT

Ký hiệu	Bảng logic	
	1	Q
	0	1
	1	0

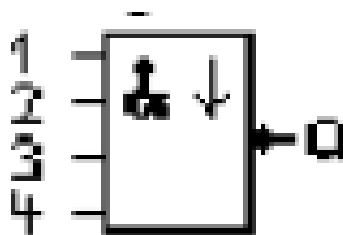
Bảng 5.4. Bảng logic cổng NOT

4.4. Hàm NAND

Ký hiệu	Bảng Logic cổng NAND				
 Ngõ ra cổng NAND chỉ bằng 0 khi tất cả ngõ vào cùng bằng 1.	1	2	3	4	Q
	0	0	0	0	1
	0	0	0	1	1
	0	0	1	0	1
	0	0	1	1	1
	0	1	0	0	1
	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	1
	1	0	0	0	1
	1	0	0	1	1
	1	0	1	0	1
	1	0	1	1	1
	1	1	0	0	1
	1	1	0	1	1
	1	1	1	0	1
	1	1	1	1	0

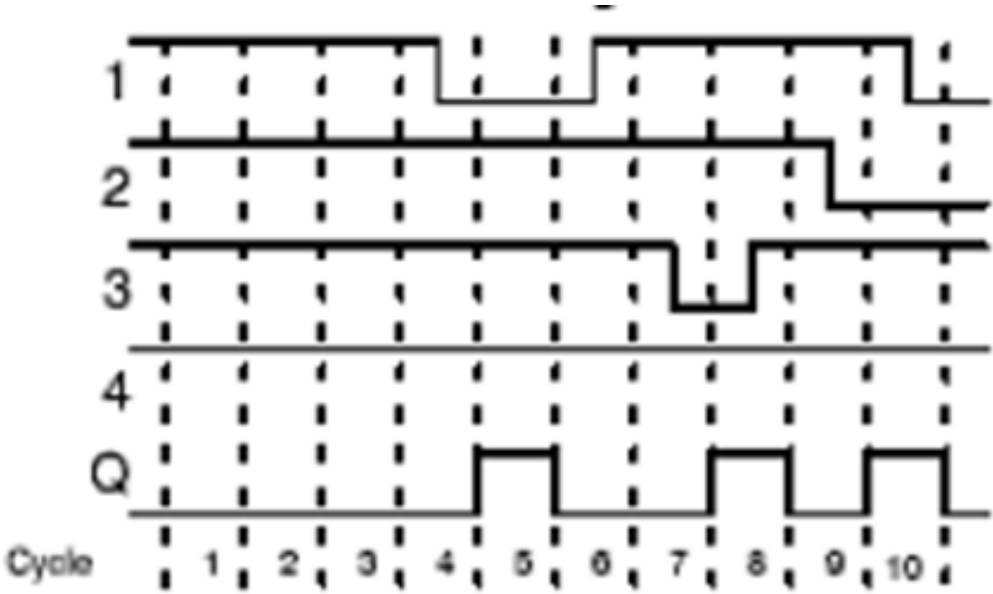
Bảng 5.5. Bảng logic cổng NAND

Cổng NAND lấy cạnh xung lên:



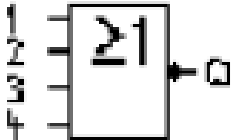
Hình 5.8. Cổng NAND lấy cạnh xung lên:

Ngõ ra của cổng NAND lấy cạnh xung lên bằng 1 trong 1 chu kỳ máy tại thời điểm đầu tiên mà một trong các ngõ vào bằng 0.



Hình 5.9. Giải đồ thời gian

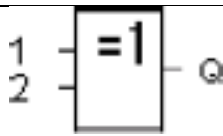
4.5. Hàm NOR

Ký hiệu	Bảng Logic cổng NAND				
 Ngõ ra cổng NOR bằng 1 nếu tất cả các ngõ cùng bằng 0. Ngõ vào không sử dụng ta có thể dùng ký hiệu x (x=0)	1	2	3	4	Q
	0	0	0	0	1
	0	0	0	1	0
	0	0	1	0	0
	0	0	1	1	0
	0	1	0	0	0
	0	1	0	1	0
	0	1	1	0	0
	0	1	1	1	0
	1	0	0	0	0
	1	0	0	1	0
	1	0	1	0	0
	1	0	1	1	0
	1	1	0	0	0
	1	1	0	1	0
	1	1	1	0	0

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

Bảng 5.6. Bảng logic cổng NOR

4.6. Hàm XOR

Ký hiệu	Bảng logic		
 Ngõ ra cổng XOR bằng 1 khi mức logic của 2 ngõ vào khác nhau. Ngõ vào không sử dụng ta có thể dùng ký hiệu x (x = 0).	1	2	Q
	0	0	0
	0	0	0
	0	0	1
	0	0	1

Bảng 5.7. Bảng logic cổng XOR

BÀI 6: CÁC CHỨC NĂNG ĐẶC BIỆT CỦA LOGO!

Mã bài: MĐ 20 - 06

Giới thiệu:

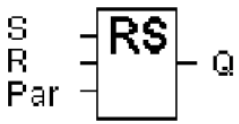
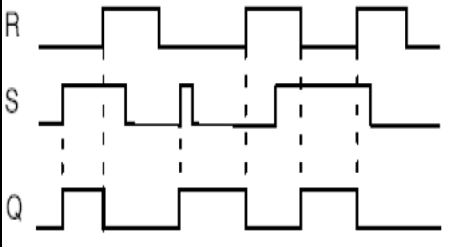
Để thực hiện các bài toán điều khiển tự động theo chương trình số bên cạnh những lệnh logic cơ bản Logo còn có các hàm chức năng như Timer, Counter, các hàm tạo xung, đồng hồ thời gian thực... các hàm này sẽ giúp người lập trình có thể giải quyết các bài toán phức tạp, hay đôi khi sử dụng các câu lệnh chức năng đặc biệt sẽ là giải pháp biến các bài toán phức tạp thành đơn giản, chương trình trở lên ngắn gọn hơn.

Mục tiêu:

- Sử dụng, khai thác đúng chức năng các hàm đặc biệt của LOGO!.
- Viết các chương trình ứng dụng các hàm cơ bản theo từng yêu cầu cụ thể.
- Rèn luyện tính cẩn thận, tỉ mỉ, chính xác, tư duy khoa học và sáng tạo.
- Đảm bảo an toàn cho người và thiết bị.

Nội dung chính:

1. Latching Relay (Relay chốt)

Ký hiệu	Kết nối	Mô tả	Giản đồ thời gian
	Input S	Tín hiệu mức 1 ngõ này sẽ set ngõ ra Q	
	Input R	Tín hiệu mức 1 ngõ này sẽ reset ngõ ra Q	
	Output Q	Ngõ ra Q được set với tín hiệu S và được reset với tín hiệu R.	

Hình 6.1: Hàm LATCHING

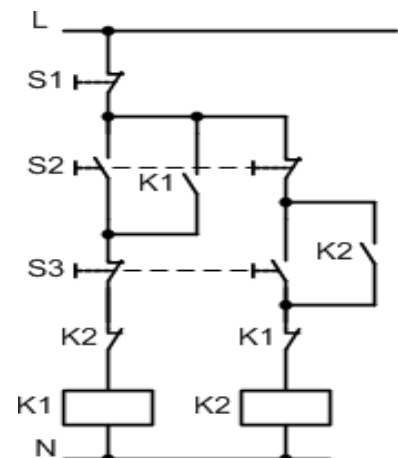
Trong trường hợp cả hai ngõ S và R đều bằng 1 thì ngõ ra sẽ được reset. (reset có mức ưu tiên cao).

Bài tập : Cho mạch điện như hình vẽ.

Mô tả hoạt động: Nhấn nút S2 thì cuộn dây K1 có điện và tự giữ khởi động động cơ chạy thuận. Nhấn S3 thì cuộn dây K1 mất điện và cuộn dây K2 có điện và tự giữ khởi động động cơ chạy nghịch.

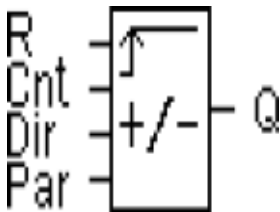
Nhiệm vụ:

- Vẽ sơ đồ động lực.



- Lập bảng xác lập ngõ vào/ra.
- Vẽ sơ đồ kết nối LOGO!.
- Viết chương trình bằng thiết bị lập trình ở dạng FBD và thử chương trình.
- Lập bảng liệt kê lệnh.

2. Counter Up and Down (Bộ đếm lên xuống)

Ký hiệu	Kết nối	Mô tả
	Input R	Tín hiệu mức 1 ngõ R sẽ reset giá trị đếm về 0.
	Input Cnt	Cạnh lên của chân này sẽ thực hiện chức năng đếm. Sử dụng: - Ngõ vào I5/I6 được dùng cho đếm tốc độ cao (chỉ đối với version LOGO! 12/24 RC/RCo và LOGO! 24/24o), tối đa 2Khz. - Các ngõ vào còn lại được dùng cho đếm tần số thấp (trong vòng 4Hz)
	Input Dir	Chọn chiều đếm: 0: đếm lên 1: đếm xuống
	Parameter	On: ngưỡng On của ngõ ra Q (giá trị từ 0...999999) Off: ngưỡng Off của ngõ ra Q (giá trị từ 0...999999)
	Output Q	Ngõ ra được set hay reset phụ thuộc vào giá trị đếm và các ngưỡng đặt.

Hình 6.2. Bộ đếm lên xuống



Hình 6.3. Giải đồ xung bộ đếm lên xuống

Mô tả:

Giá trị đếm sẽ được tăng hoặc giảm một đơn vị ứng với mỗi cạnh lên của ngõ vào **Cnt** và ngõ vào **Dir**. Giá trị đếm được **reset** về 0 khi ngõ vào **R** lên 1. Ngõ ra được **set** hoặc **reset** theo quy luật sau đây:

Trường hợp ngưỡng On $\geq$ ngưỡng Off $Q = 1$, nếu $Cnt \geq On$ $Q = 0$, nếu $Cnt < Off$.

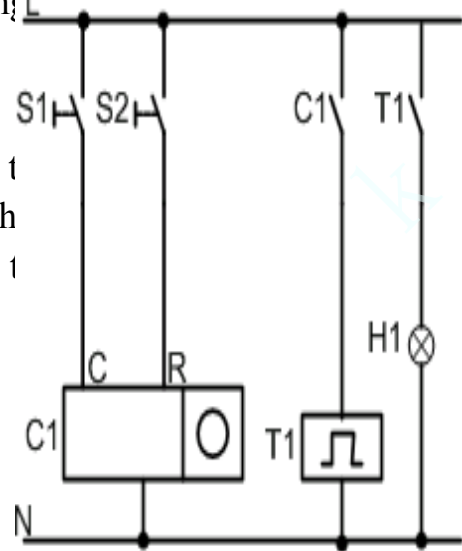
Trường hợp ngưỡng On $<$ ngưỡng Off, nếu $Cnt \geq On$ $Q = 1$ khi : On $< Cnt < Off$

Bài tập : Cho mạch điện như hình vẽ.

Mô tả hoạt động: Đóng/mở nút nhấn S1 5 lần t đóng tiếp điểm C1 cuộn dây T1 có điện sau thời gian 2s thì đóng tiếp điểm T1 đèn sáng chớp 1 theo thời gian đóng mở của T1.

Nhiệm vụ:

- Lập bảng xác lập ngõ vào/ra.
- Vẽ sơ đồ kết nối LOGO!.
- Viết chương trình bằng thiết bị lập trình ở dạng FBD và thử chương trình.
- Lập bảng liệt kê lệnh.



3. Timer ON delay

Ký hiệu	Kết nối	Mô tả	Giải đồ thời gian
	Input Trg	Ngõ vào khởi động thời gian delay on	
	Parameter T	Khoảng thời gian delay	
	Output Q	Ngõ ra sẽ lên 1 sau thời gian T kể từ khi ngõ Trg lên 1.	

Hình 6.4 Timer On Delay

Hoạt động:

- Thời gian **Ta** được khởi động khi ngõ vào **Trg** chuyển từ 0 lên 1. (**Ta**: thời gian hiện hành của LOGO)
- Nếu trạng thái ngõ vào **Trg** duy trì mức 1 trong suốt khoảng thời gian **T** thì ngõ ra **Q** được lên mức 1 cho đến khi ngõ vào chuyển từ 1 xuống 0.
- Nếu trong khoảng thời gian **T** mà ngõ vào chuyển từ 1 xuống 0 thì ngõ ra cũng xuống 0 và timer bị **Reset**.

- Nếu tính năng **Retentive** không được set thì khi mất nguồn, ngõ ra **Q** và thời gian **Ta** bị reset.

BÀI 7: LẬP TRÌNH TRỰC TIẾP TRÊN LOGO!

Mã bài: MĐ 20- 07

Giới thiệu:

PLC S7-200 khi muốn lập trình một ứng dụng nào đó dù lớn hay nhỏ yêu cầu bắt buộc người kỹ thuật viên cần phải trang bị máy tính, cable lập trình vì thế khi sản xuất Logo nhà sản xuất đã xác định đó là một bộ lập trình cỡ nhỏ đòi hỏi phải có tính linh hoạt có thể lập trình mọi lúc mọi nơi do đó trên PLC Logo một số dòng nhà sản xuất siemens tích hợp thêm màn hình hiển thị để người lập trình trực tiếp bằng tay mà không bắt buộc phải lập trình trên máy tính. Nội dung bài này giới thiệu kỹ năng lập trình trên LOGO thông qua các phím trực tiếp bố trí trên cấu hình phần cứng của nó

Mục tiêu:

- Thực hiện đúng các nguyên tắc lập trình, các phương pháp kết nối của LOGO!.
- Viết các chương trình ứng dụng theo từng yêu cầu cụ thể. Sử dụng, khai thác đúng chức năng các vùng nhớ, card nhớ của LOGO!.
- Tính toán, chọn lựa chính xác dung lượng, chức năng của bộ nhớ theo từng yêu cầu cụ thể.
- Rèn luyện tính cẩn thận, tỉ mỉ, chính xác, tư duy khoa học và sáng tạo.
- Đảm bảo an toàn cho người và thiết bị.

1. Bốn quy tắc sử dụng phím trên Logo!

1.1. Quy tắc 1: thay đổi chế độ hoạt động

- Để tạo chương trình trong chế độ lập trình (**programming mode**). Sau khi bật nguồn và khi hiển thị “**No program/Press ESC**” thì nhấn phím **ESC** để đăng nhập chế độ lập trình.

- Giá trị các tham số và timer của một chương trình có thể được sửa một trong hai nơi là **parameter assingnment mode** hoặc **programming mode**. Trong **parameter assgnment** thì LOGO! ở chế độ **RUN**. Để làm việc trong chế độ **Programming mode** thì cần dừng chương trình (Stop).

- Lựa chọn lệnh “**Start**” trên menu chính để chạy chương trình.

- Khi hệ thống đang hoạt động có thể trở lại chế độ **Parameter assignment mode** bởi nhấn phím **ESC**.

- Khi ở chế độ thiết lập các thông số (**Parameter assignment mode**) được mở và muốn trở lại chế độ lập trình (**Programming mode**), chọn lệnh **Stop** từ menu **parameter assignment** và xác nhận “**Stop prg**”

1.2. Quy tắc 2: các ngõ vào và ngõ ra

- Luôn luôn soạn thảo chương trình bởi viết từ ngõ ra trở về ngõ vào.
- Có thể nối một ngõ ra đến nhiều ngõ vào nhưng không được kết nối cùng một ngõ vào đến một vài ngõ ra.
- Trong cùng một dòng chương trình không thể kết nối một ngõ ra đến dòng bên trên. Nên liên kết thông qua flags hoặc ngõ ra.

1.3. Quy tắc 3: Con trỏ và di chuyển con trỏ

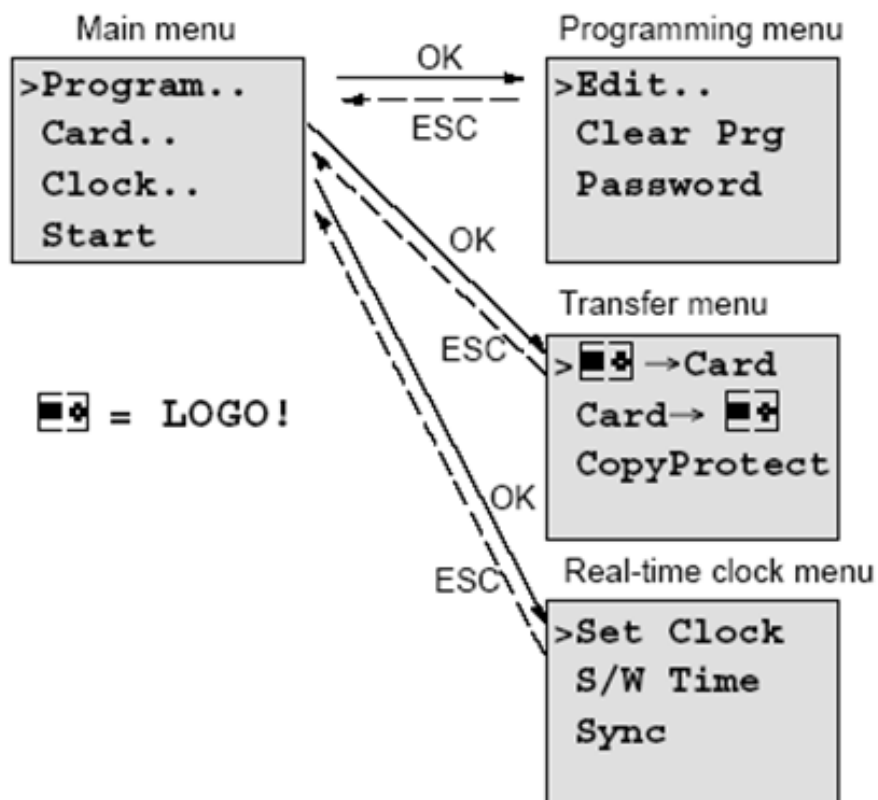
- Chỉ có thể di chuyển con trỏ khi xuất hiện dạng gạch dưới:
 - + Nhấn các phím ▲, ►, ▼, ◀, để di chuyển trong chương trình.
 - + Nhấn **OK** để thay đổi **khởi/ kết nối** đã lựa chọn.
 - + Nhấn **ESC** để thoát chế độ soạn thảo.
- Lựa chọn một kết nối/khởi
- Khi con trỏ xuất hiện hình vuông
 - + Nhấn phím ▼, ▲ để lựa chọn kết nối hoặc khởi.
 - + Xác nhận lựa chọn bằng **OK**
 - + Nhấn **ESC** để trở lại bước trước đó.

1.4. Quy tắc 4: Lưu trữ chương trình

- Logo chỉ lưu trữ chương trình khi đã hoàn thiện

2. Cách gọi các chức năng

2.1. Chế độ lập trình (Programming mode)

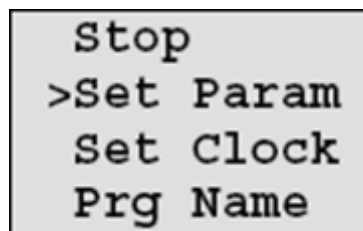


Hình 7.1. Giao diện màn hình Logo

2.2. Chế độ thiết lập thông số (Parameter assignment mode)

Để lựa chọn các thông số trước tiên ta phải dừng hoạt động chương trình (chọn **Stop**) sau đó thực hiện theo các bước sau:

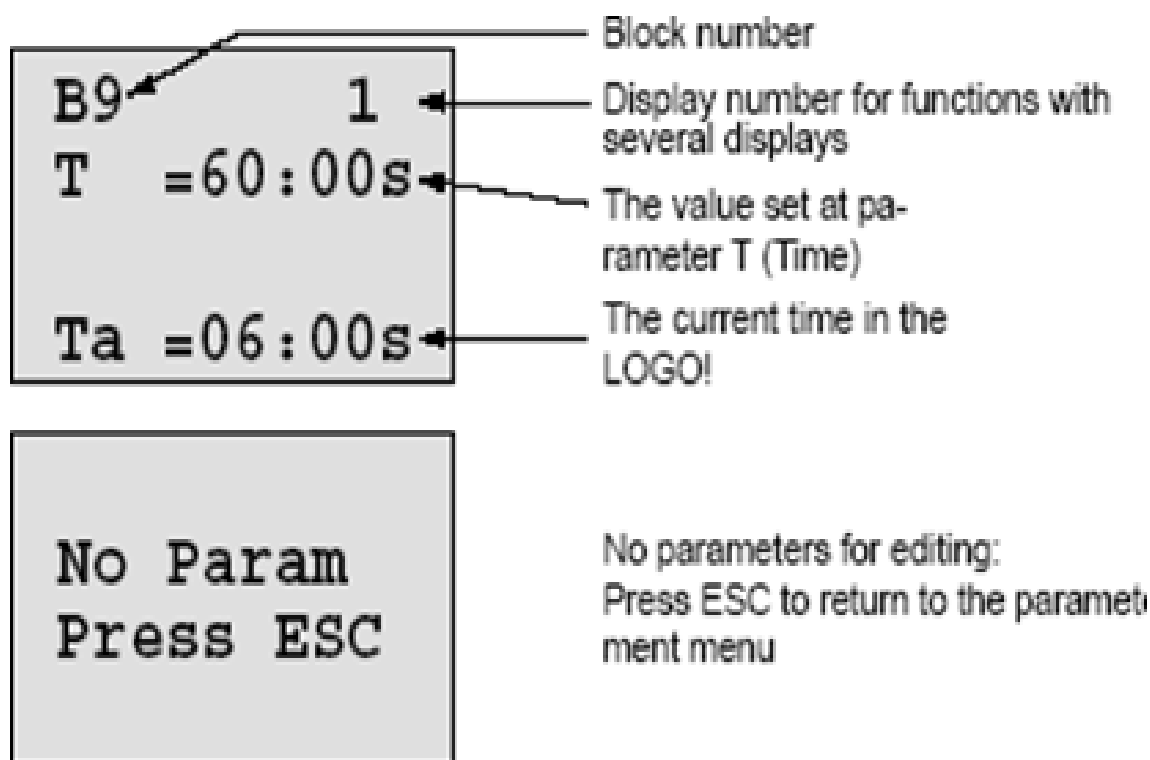
- Chọn **Set Param** bằng phím **▲**, **▼**



Hình 7.2. Giao diện Set Param

- Xác nhận lựa chọn với phím **OK**

LOGO! Sẽ hiển thị tham số đầu tiên, nếu không có tham số nào có thể thiết lập thì nhấn **ESC** để trở lại menu **Parameter assignment**

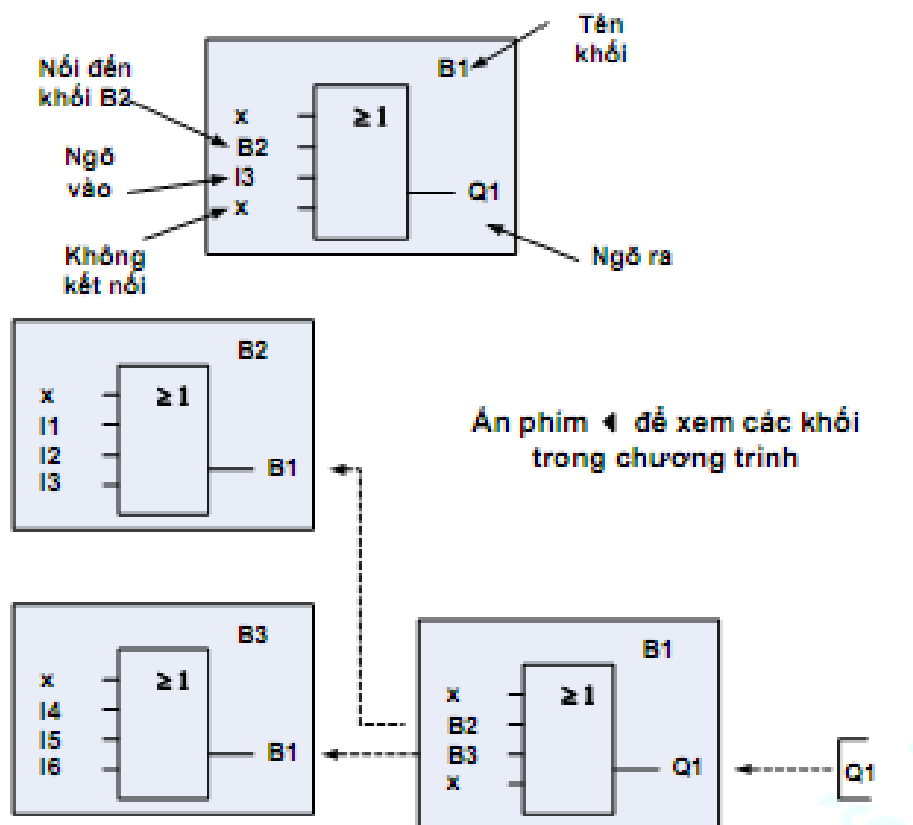


Hình 7.3. Giao diện cài đặt các tham số cơ bản

- Lựa chọn các tham số mong muốn bằng phím **▼**, **▲**
- Nhấn **OK** để xác nhận lựa chọn

3. Phương pháp kết nối các khối chức năng

3.1. Biểu diễn các khối trong LOGO!

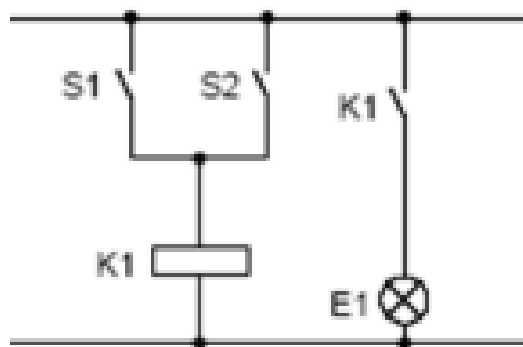


Hình 7.4. Biểu diễn các khối trong Logo

3.2. Soạn thảo chương trình

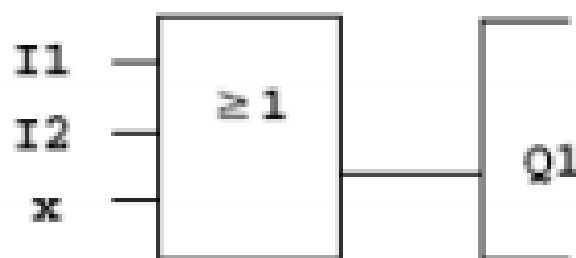
3.2.1. Ví dụ 1

Viết chương trình hoạt động theo mạch điện sau:



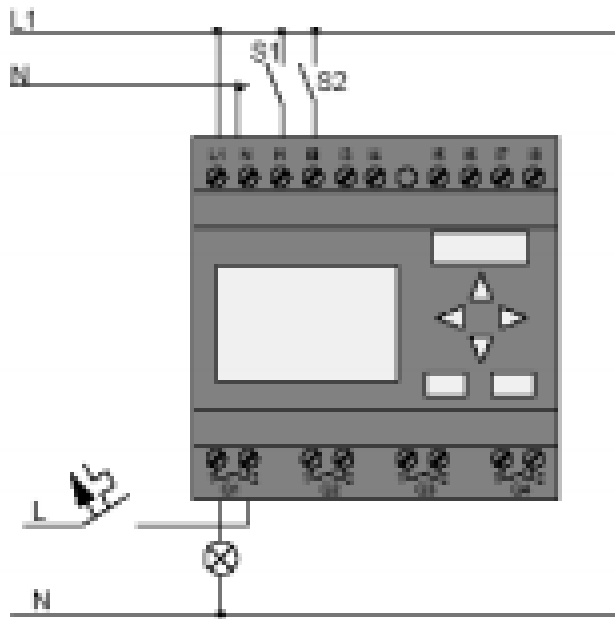
Hình 7.5: Sơ đồ mạch điện

Mạch điện trên được biểu diễn bằng hàm logic sau:



Hình 7.6. chương trình logic

Kết nối mạch điện qua LOGO!



Hình 7.7. Sơ đồ kết nối Logo

Soạn thảo chương trình:

Lưu ý: Khi viết chương trình trong LOGO! Ta thực hiện viết từ ngõ ra ngược trở lại ngõ vào.

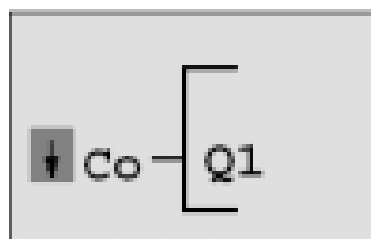
Nhập ngõ ra:

- Đăng nhập vào chế độ lập trình chế độ lập trình, sẽ xuất hiện việc soạn thảo chương trình với ngõ ra Q1. Nhấn phím ▼ con trỏ sẽ di chuyển sang bên trái.



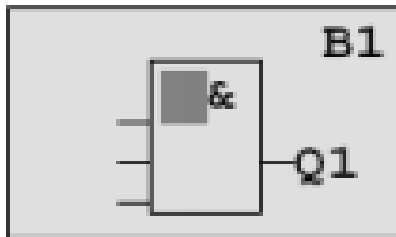
Hình 7.8. Giao diện nhập ngõ ra

- Nhấn OK để lựa chọn chế độ sửa. Con trỏ chuyển từ dạng gạch chân sang dạng hình vuông nhấp nháy, lúc này ta có thể thay đổi các tùy chọn.



Hình 7.9. Giao diện sau nhấn OK

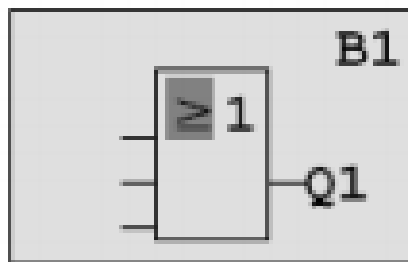
- Lựa chọn GF (basic functions) bằng cách nhấn phím ▼ đến khi GF xuất hiện, lúc này khối đầu tiên từ nhóm GF xuất hiện:



The AND is the first block of the basic functions list. The solid square cursor prompts you to select a block.

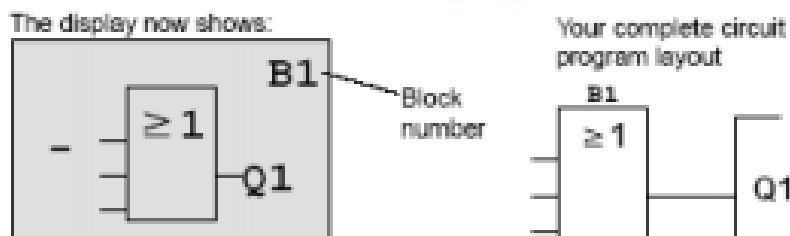
Hình 7.10. Giao diện lựa chọn hàm GF

- Nhấn phím ▲ hoặc ▼ cho đến khi khối OR xuất hiện, con trỏ vuông vẫn được nằm trên khối.



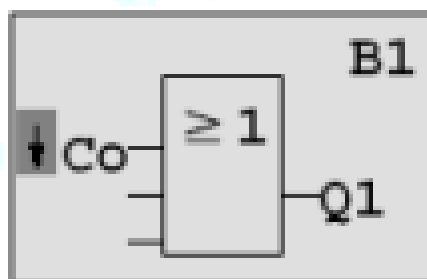
Hình 7.11. Giao diện hàm OR

- Nhấn OK để xác nhận hàm OR và nhập ngõ vào:



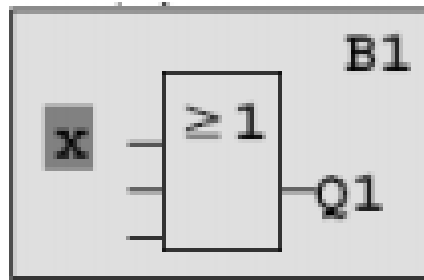
Hình 7.12. Giao diện sau khi xác nhận hàm OR

- Đến đây đã hoàn thành việc nhập khối đầu tiên, mỗi khối mới được gán với một số. Nhấn OK màn hình sẽ hiển thị



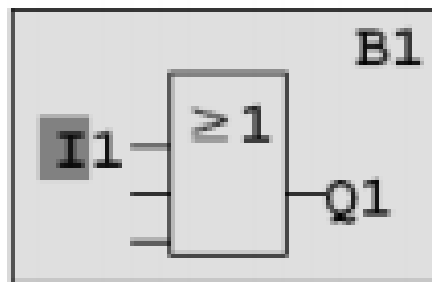
Hình 7.13. Giao diện lấy danh sách Co

Lựa chọn Co, nhấn OK, màn hình sẽ hiển thị



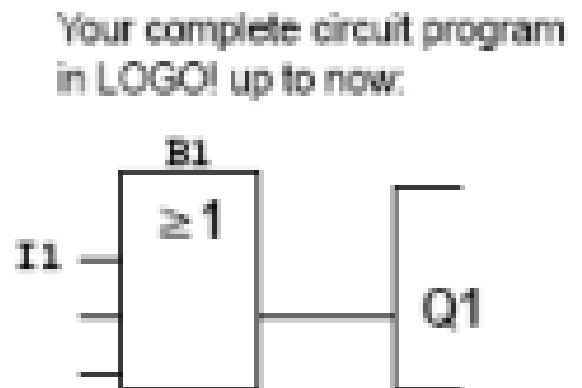
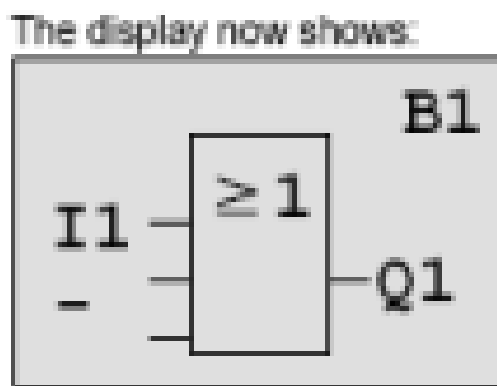
Hình 7.14. Giao diện vào danh sách Co

- Nhấn phím ▼ để bắt đầu các ngõ vào



Hình 7.15. Giao diện lấy I1

- Nhấn OK thì I1 được nối đến ngõ vào cổng OR. Con trỏ nhảy đến vị trí tiếp theo của ngõ vào cổng OR.

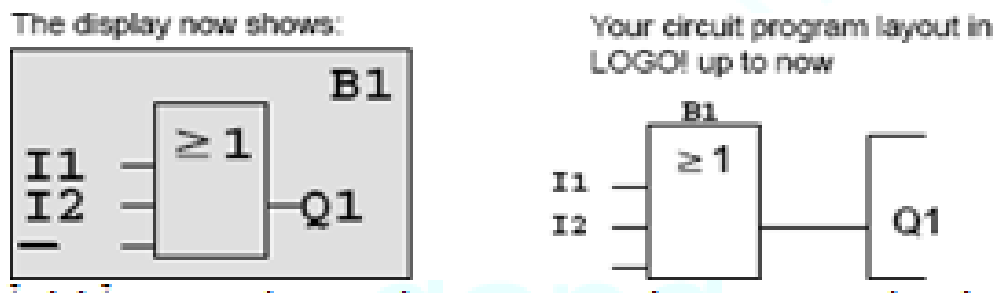


Hình 7.16. Hoàn thiện lấy I1

Để nối ngõ vào I2 đến ngõ vào cổng OR. Thực hiện từng bước sau:

- Chuyển đến chế độ lập trình: nhấn **OK**
- Để lựa chọn **Co**: nhấn phím ▲ hoặc ▼
- Để xác nhận **Co**: nhấn phím **OK**
- Để lựa chọn **I2**: nhấn phím ▲ hoặc ▼
- Để xác nhận **I2**: nhấn **OK**

Lúc này I2 được nối đến ngõ vào cổng OR

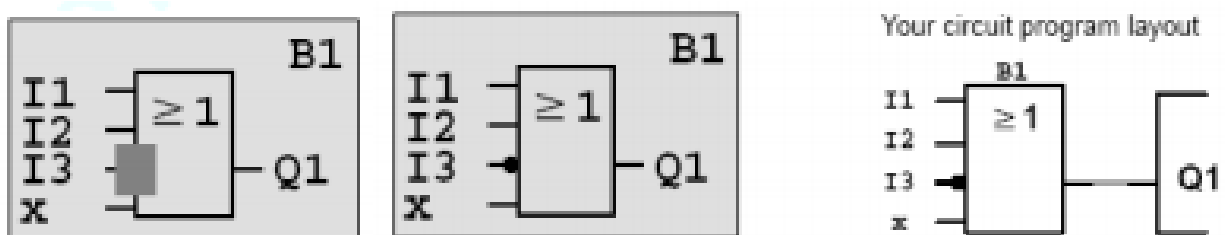


Hình 7.17. Hoàn thiện lấy I1 và I2

Để kiểm tra hoạt động của chương trình chuyển LOGO! Về chế độ RUN

- Để trở về menu chính: nhấn **ESC**
- Di chuyển con trỏ đến vị trí Start: nhấn phím **▲** hoặc **▼**
- Để xác nhận “Start”: nhấn **OK**

Chú ý: Có thể đảo ngược các ngõ vào riêng lẻ. Ngõ vào mức 1 sẽ được chuyển thành mức 0 ở chương trình, mức 0 được đảo thành mức 1. Để thực hiện thì di chuyển con trỏ đến vị trí thích hợp sau đó nhấn phím **▲** hoặc **▼**. Sau đó nhấn **OK** để xác nhận

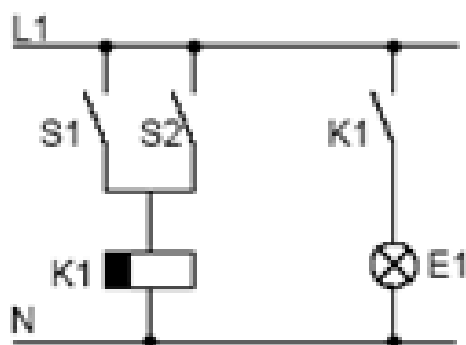


Hình 7.18. Lấy ngõ vào đảo

3.2.2. Ví dụ 2

Ví dụ 1 trình bày cách nhập một hàm đơn giản. Với ví dụ 2 sẽ thực hiện:

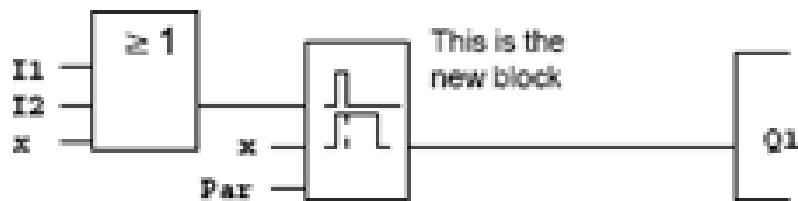
- Thêm một khối vào chương trình hiện có.
- Chọn 1 khối cho một hàm đặc biệt.
- Thiết lập các tham số.



Hình 7.19. Sơ đồ điện

S1 và S2 đóng cắt điện cho relay K1 dùng để chuyển mạch cho tải E1 và cắt nguồn cung cấp sau 12 phút.

Chương trình được viết trong LOGO!



Hình 7.20. Chương trình logic

Sửa chương trình:

- Chuyển LOGO! Đến chế độ lập trình
- Trên menu chính chọn “**Program**”
- Trên **programming menu**, chọn “**Edit**” → **OK** → **Edit prg** → **OK**

Thêm một khối đến chương trình hiện có:

- Di chuyển con trỏ đến B của khối B1



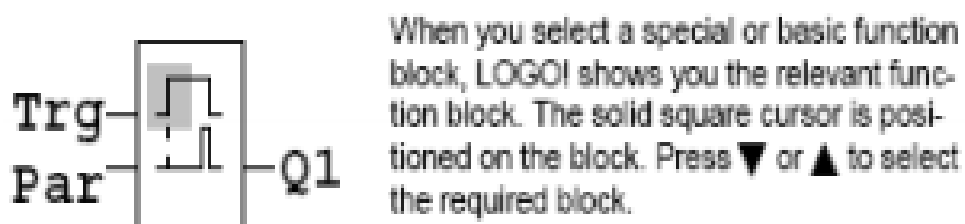
Hình 7.21. Di chuyển con trỏ tới khối B1

- Nhấn OK để chèn thêm khối mới ở điểm đã chọn



Hình 7.22. Xác nhận chèn thêm khối

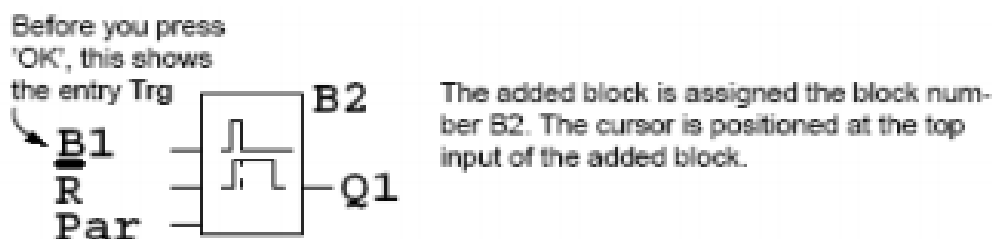
- Ấn ▼ để chọn SF list
- Ấn OK → khối đầu tiên của SF xuất hiện



Hình 7.24. Giao diện khi vào khối hàm SF

- Chọn khối Off delay và nhấn OK

Khối được thêm được gán với số khối là B2. Con trỏ được đặt ở ngõ vào bên trên của khối.

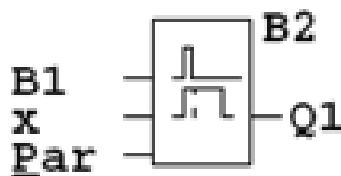


Hình 7.25. Giao diện hàm Off delay

Khối B1 được nối đến trước Q1 một cách tự động đến ngõ vào bên trên của khối mới. Chú ý rằng chỉ có thể liên kết các ngõ vào số với các ngõ ra số hoặc các ngõ vào tương tự với các ngõ ra tương tự. Nếu không thì khối cũ sẽ mất.

Khối Off-delay có ba ngõ vào. ở trên là ngõ vào trigger (Trg) sử dụng để bắt đầu thời gian Off-delay. Trong ví dụ này khối B1 là Trigger của Off-delay. Để Reset thời gian và ngõ ra sử dụng ngõ vào R (Reset). Thiết lập thời gian Off-delay ở ngõ vào Par.

Ở ví dụ 2 do không sử dụng ngõ vào Reset của hàm Off-delay nên sẽ được gán kết nối “x”.

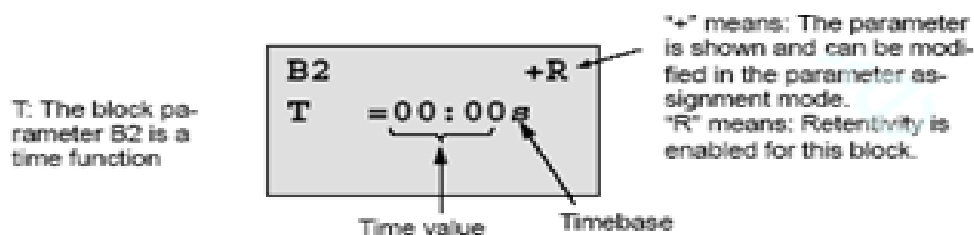


Hình 7.26. Gán X đầu vào Reset

Ấn định các tham số của khối

- Di chuyển con trỏ đến **Par**, nếu không xuất hiện ở vị trí này: nhấn ▼ hoặc ▲
- Chuyển đến chế độ sửa: Nhấn OK

LOGO! hiển thị các tham số trong cửa sổ ấn định các tham số:



Hình 7.27. Đặt thời gian cho hàm Off deley

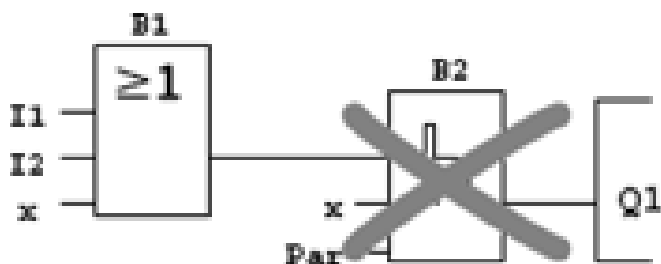
Để thay đổi giá trị thời gian:

- Nhấn ◀ và ▶ để đặt vị trí con trỏ.
- Ấn ▼ và ▲ để sửa đổi giá trị ở vị trí thích hợp
- Xác nhận sửa đổi bằng phím **OK**

3.3. Các thao tác cơ bản

3.3.1. Xóa 1 khối

Giả sử để xóa khối B2 và nối B1 trực tiếp Q1

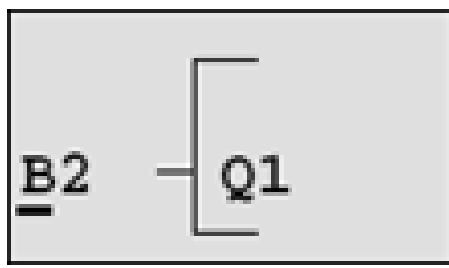


Hình 7.28. Xóa khối B2

Thực hiện theo các bước sau:

Chuyển LOGO! đến chế độ lập trình.

- Chọn **Edit**
- Chọn **Edit prg.**
- Di chuyển con trỏ con trỏ đến B2, nhấn ◀



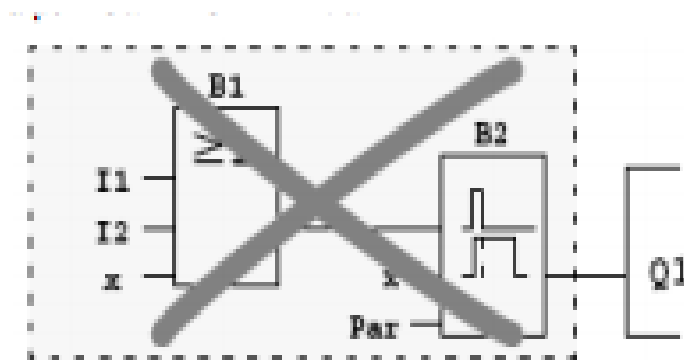
Hình 7.29. Đưa con trỏ về vị trí B2

- Xác nhận lựa chọn bằng phím **OK**.
- Thay thế khối B2 với khối B1 ở ngõ ra Q1. Thực hiện:
 - + Chọn **BN list**: nhấn ▼ hoặc ▲
 - + Xác nhận chọn: nhấn **OK**
 - + Chọn **B1**
 - + Xác nhận lựa chọn **B1**: Nhấn **OK**

Kết quả là khối B2 được xóa bởi vì không còn sử dụng trong chương trình. Khối B1 đã được nối trực tiếp đến ngõ ra.

3.3.2. Xóa các nhóm khối

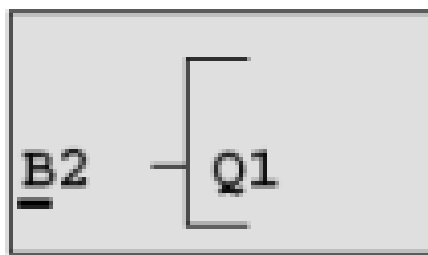
Ví dụ: xóa khối B1 và B2



Hình 7.30. Xóa nhiều khối

Thực hiện theo các bước sau:

- Chuyển LOGO! về chế độ lập trình.
- Chọn **Edit**
- Chọn **Edit prg.**
- Di chuyển con trỏ con trỏ đến B2, nhấn



Hình 7.31. Đưa con trỏ về khối B2

- Xác nhận lựa chọn bằng phím **OK**.
- Chọn kết nối 'x' ở ngõ ra Q1 thay vì khối B2. Thực hiện các thao tác sau:

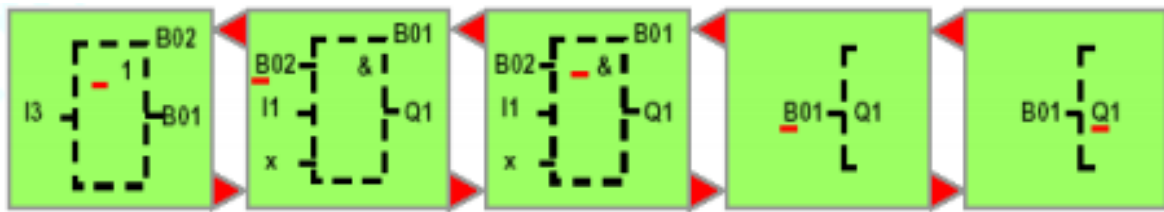
- + Chọn **Co list**: nhấn ▲ hoặc ▼
- + Xác nhận chọn: nhấn **OK**
- + Chọn 'x': nhấn ▲ hoặc ▼
- + Xác nhận chọn: nhấn **OK**

4. Lưu trữ vào thẻ nhớ và chạy chương trình

Sau khi lập trình xong, nhấn **OK** màn hình sẽ hiện lại ngõ ra cuối cùng được lập trình.

- Khi nhập chương trình xong, ấn **OK** màn hình sẽ hiện lại ngõ ra cuối cùng được lập trình.

- Dùng phím ◀, ▶ để kiểm tra chương trình nhập đúng hay chưa



Hình 7.32. Kiểm tra lại chương trình

- Chương trình được lưu tự động vào card nhớ nếu trong logo! đã gắn card nhớ.

- Nếu không có chương trình trong logo! hay card nhớ thì logo! hiển thị thông báo: **No program**.



Hình 7.33. Không có thẻ nhớ hoặc chưa có chương trình

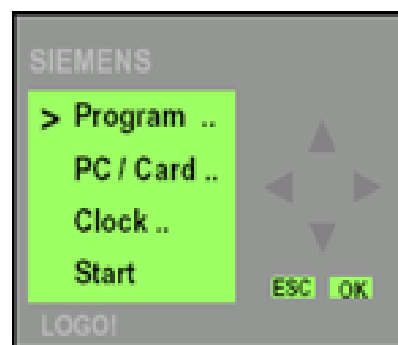
- Nếu có chương trình trong card nhớ, nó tự động chép vào logo!. nếu trong logo! đã có chương trình thì nó sẽ chép đè lên chương trình cũ.

- Nếu có chương trình trong logo! hay card nhớ thì logo! sẽ nhận trạng thái trước khi mất nguồn.

- Muốn chạy chương trình nhấn phím **ESC** 2 lần để thoát ra menu chính và con trỏ chuyển thành hình ">".

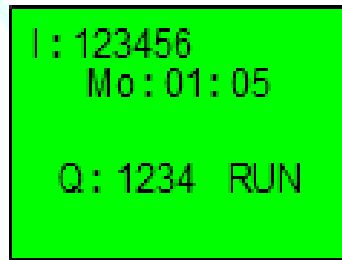
- Dùng phím ▼ di chuyển con trỏ xuống **Start**

- Chấp nhận lựa chọn nhấn **OK**.



Hình 7.34. Giao diện màn hình chính

- Logo! chuyển sang chế độ Run. ở chế độ này logo! hiển thị số ngõ vào, ngõ ra, thời gian hiện hành.



Hình 7.35. Giao diện màn hình chế độ Run

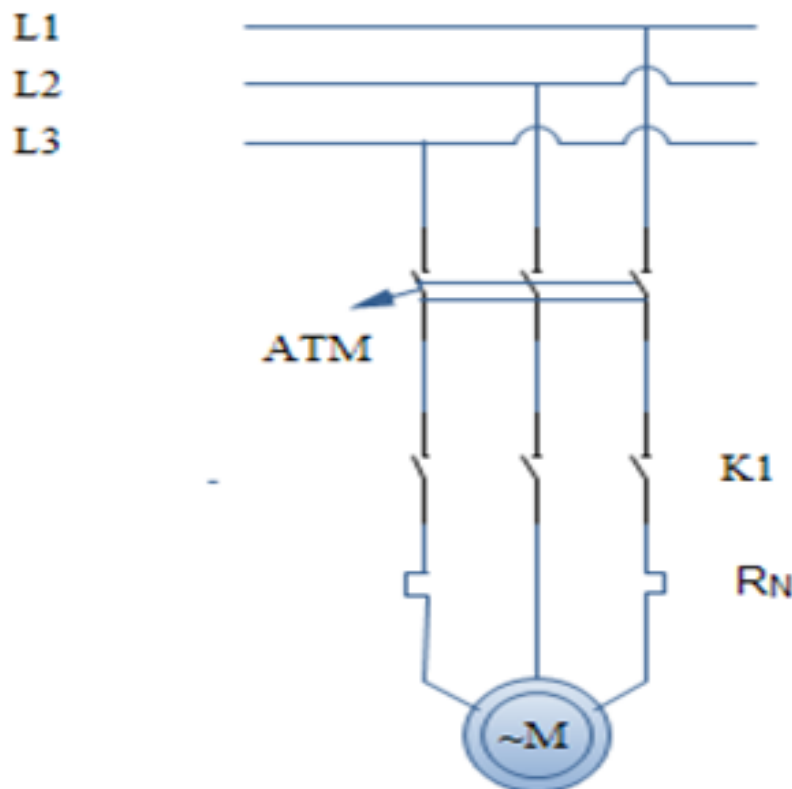
5. Bài tập ứng dụng

5.1. Điều khiển đóng cắt động cơ

Yêu cầu hệ thống:

Lập trình LOGO điều khiển đóng cắt động cơ không đồng bộ ba pha roto lồng sóc đảm bảo các yêu cầu sau:

- Nhấn nút Start contactor K1 đóng lại động cơ chạy.
- Nhấn nút Stop contactor K1 mở ra động cơ dừng.



Các bước tiến hành:

Bước 1: Chuẩn bị dụng cụ, vật tư, bản thực hành



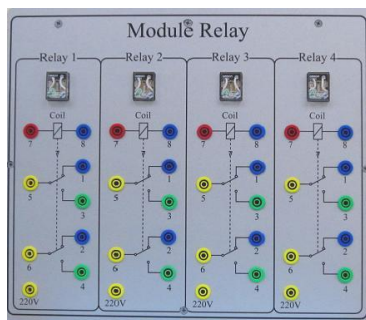
Module nguồn



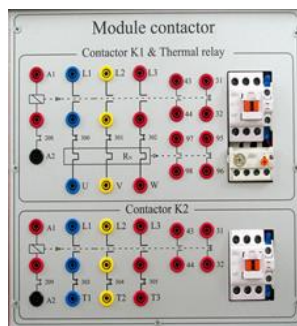
Module nút nhấn



Module PLC LOGO



Module Re lay



Module công tắc tơ K1, K2



Động cơ



Dây dẫn



Bàn thực hành gá module



Bộ dụng cụ

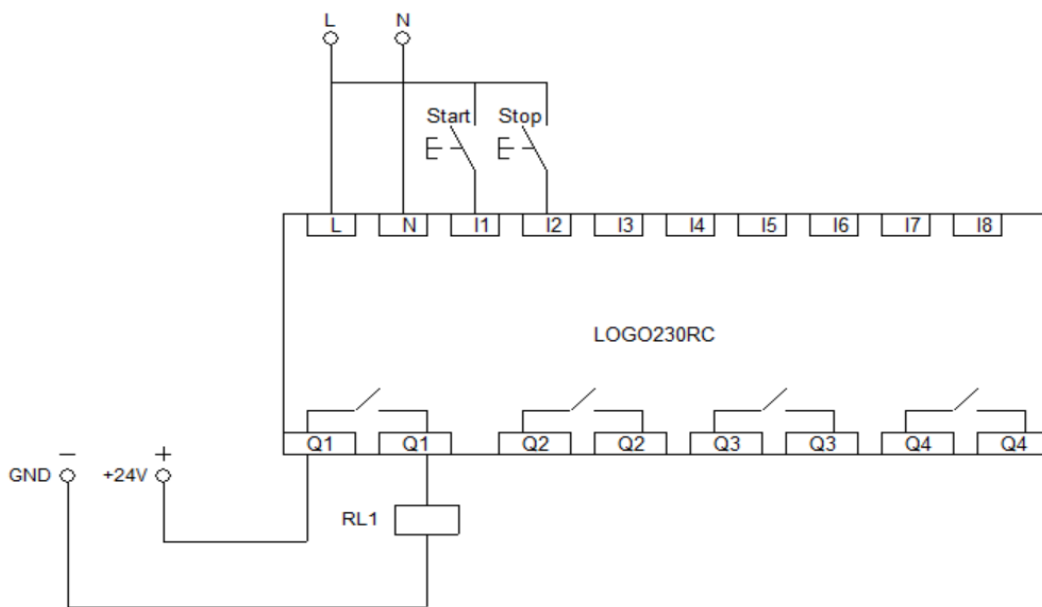
- Gá lắp các thiết bị lên bàn thực hành sau đó kết nối dây theo sơ đồ

Bước 2: Lập bảng quy định địa chỉ vào/ra:

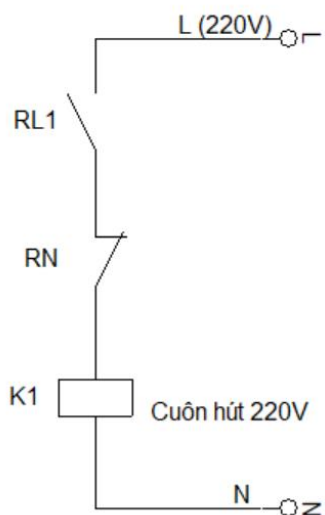
Tên	Địa chỉ	Ghi Chú
Start	I1	Nút mở điều khiển cho động cơ chạy
Stop	I2	Nút dừng
RL1	Q1	Rơ le RL1 cuộn hút 24VDC đóng cắt công tắc tơ K1 cuộn hút 220VAC

Bước 3: Sơ đồ kết nối

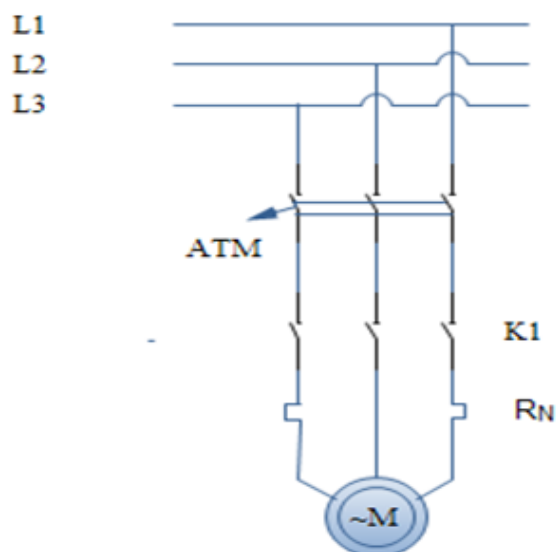
+ Sơ đồ kết nối LOGO



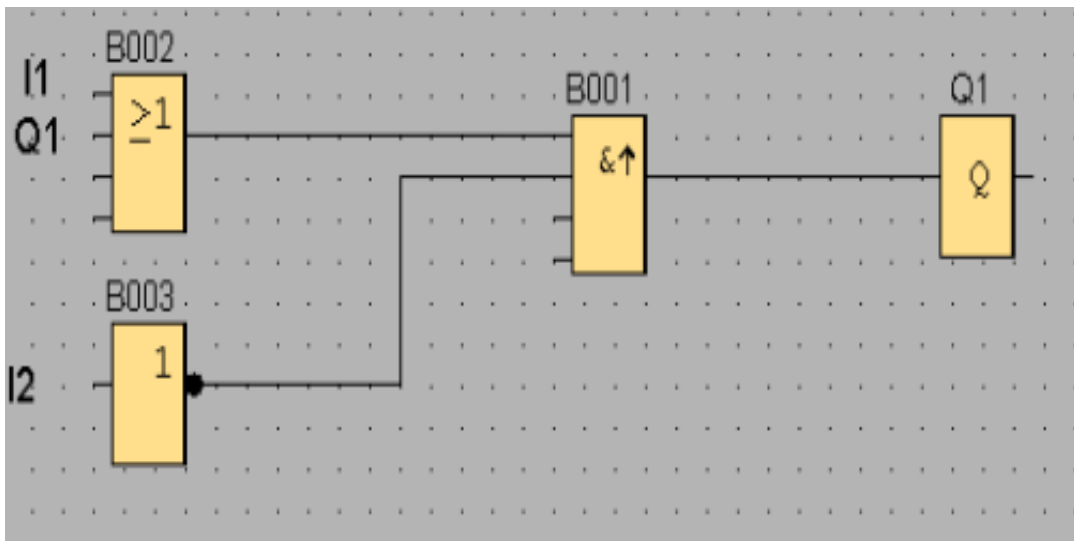
+ Sơ đồ kết nối mạch điều khiển



+ Sơ đồ kết nối mạch động lực



Bước 4: Lập trình Chương trình điều khiển trên Logo



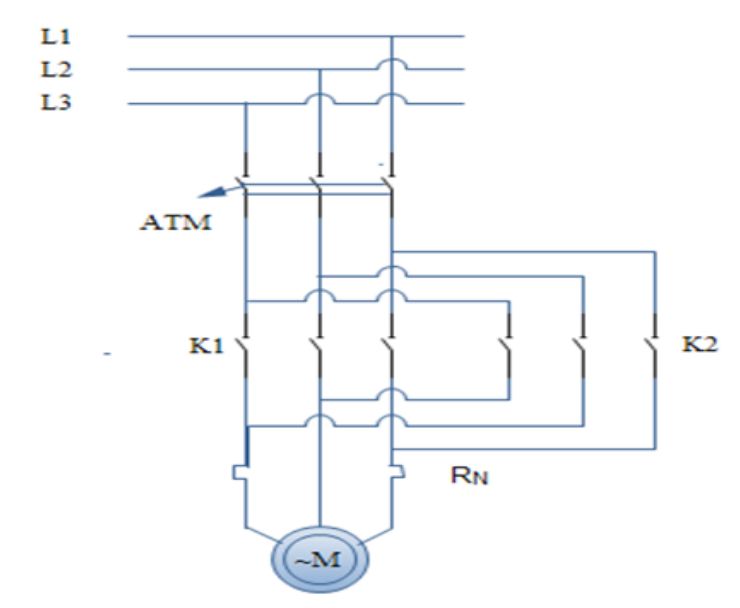
Bước 5: Vận hành

5.2. Điều khiển đảo chiều quay động cơ

Yêu cầu hệ thống:

Lập trình PLC điều khiển động cơ không đồng bộ ba pha roto lồng sóc đảm bảo các yêu cầu sau:

- Nhấn nút **Mở Thuận** động cơ chạy theo chiều thuận, nhấn nút **Mở Ngược** động cơ chạy theo chiều ngược.
- Khi động cơ đang chạy muốn dừng động cơ nhấn nút **Dừng**.
- Khi động cơ đang chạy thuận (ngược), muốn chuyển sang chạy ngược (thuận) ta phải ấn nút **Dừng** rồi ấn nút **Mở Ngược** (**Mở Thuận**).



Các bước tiến hành:

Bước 1: Chuẩn bị dụng cụ, vật tư, bàn thực hành



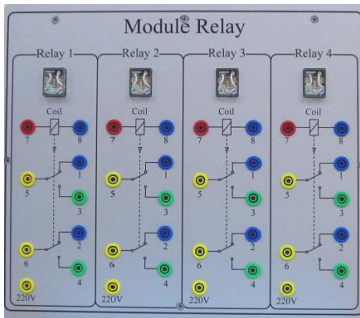
Module nguồn



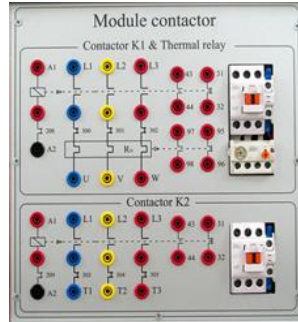
Module nút nhấn



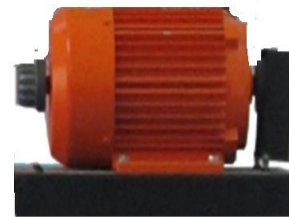
Module PLC LOGO



Module Re lay



Module công tắc tơ K1, K2



Động cơ



Dây dẫn



Bàn thực hành gá module



Bộ dụng cụ

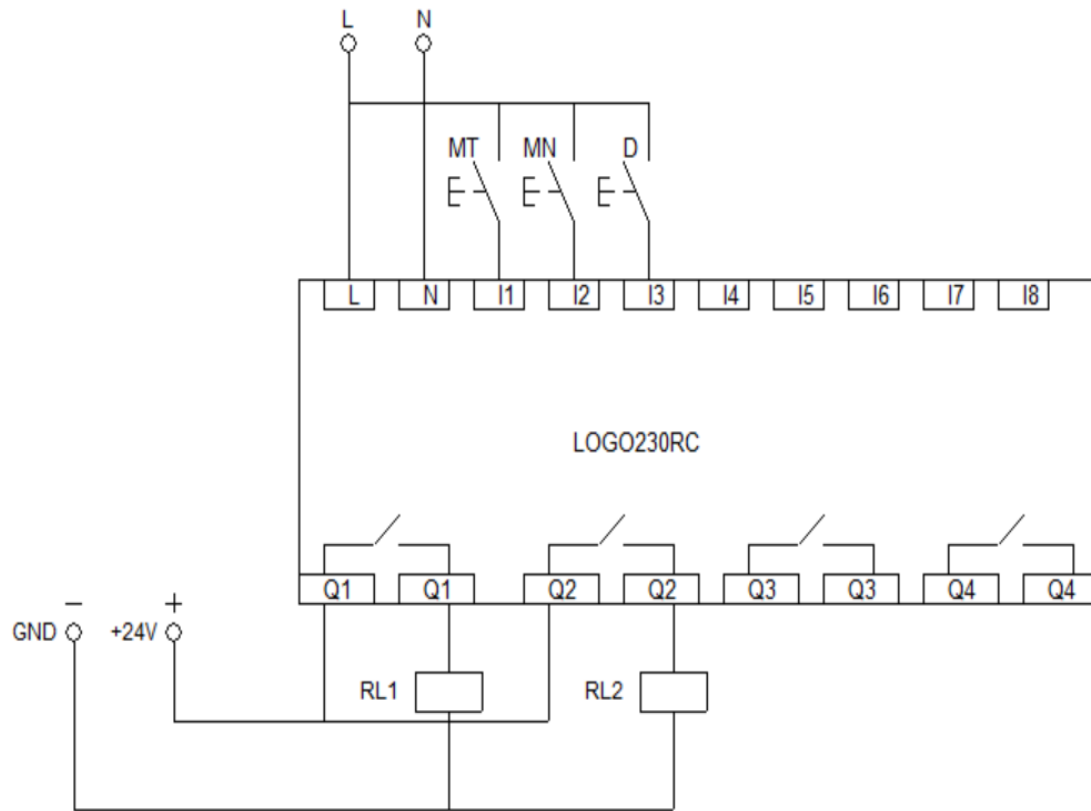
- Gá lắp các thiết bị lên bàn thực hành sau đó kết nối dây theo sơ đồ

Bước 2: Lập bảng quy định địa chỉ vào/ ra:

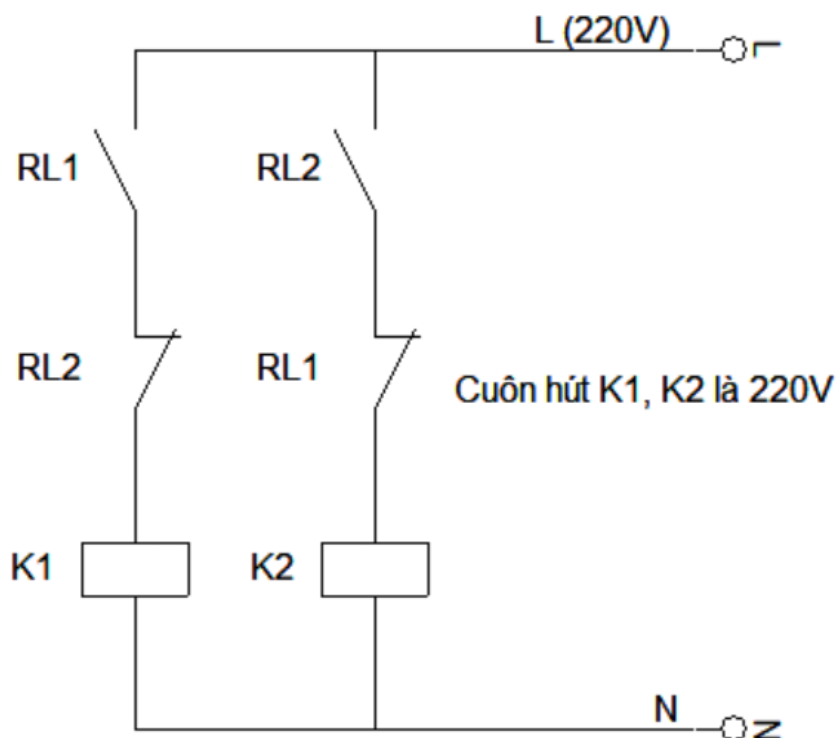
Tên	Địa chỉ	Ghi Chú
Mở Thuận	I1	Nút nhấn mở thuận điều khiển động cơ chạy thuận
Mở Ngược	I2	Nút nhấn mở ngược điều khiển động cơ chạy ngược
Dừng	I3	Nút nhấn dừng
RL1	Q1	Rơ le RL1 cuộn hút 24VDC đóng cắt công tắc tơ K1(chạy thuận) cuộn hút 220VAC
RL2	Q2	Rơ le RL1 cuộn hút 24VDC đóng cắt công tắc tơ K2(chạy ngược) cuộn hút 220VAC

Bước 3: Sơ đồ kết nối

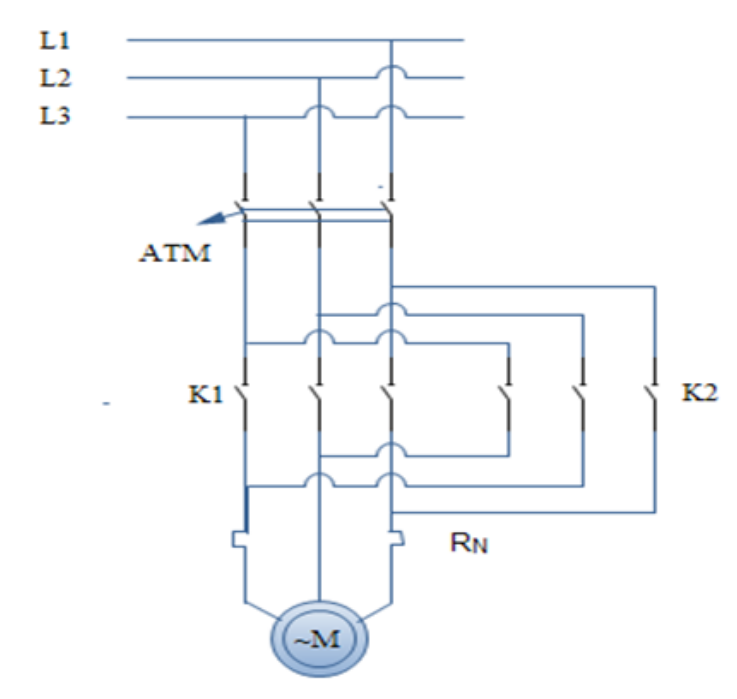
+ Sơ đồ kết nối LOGO



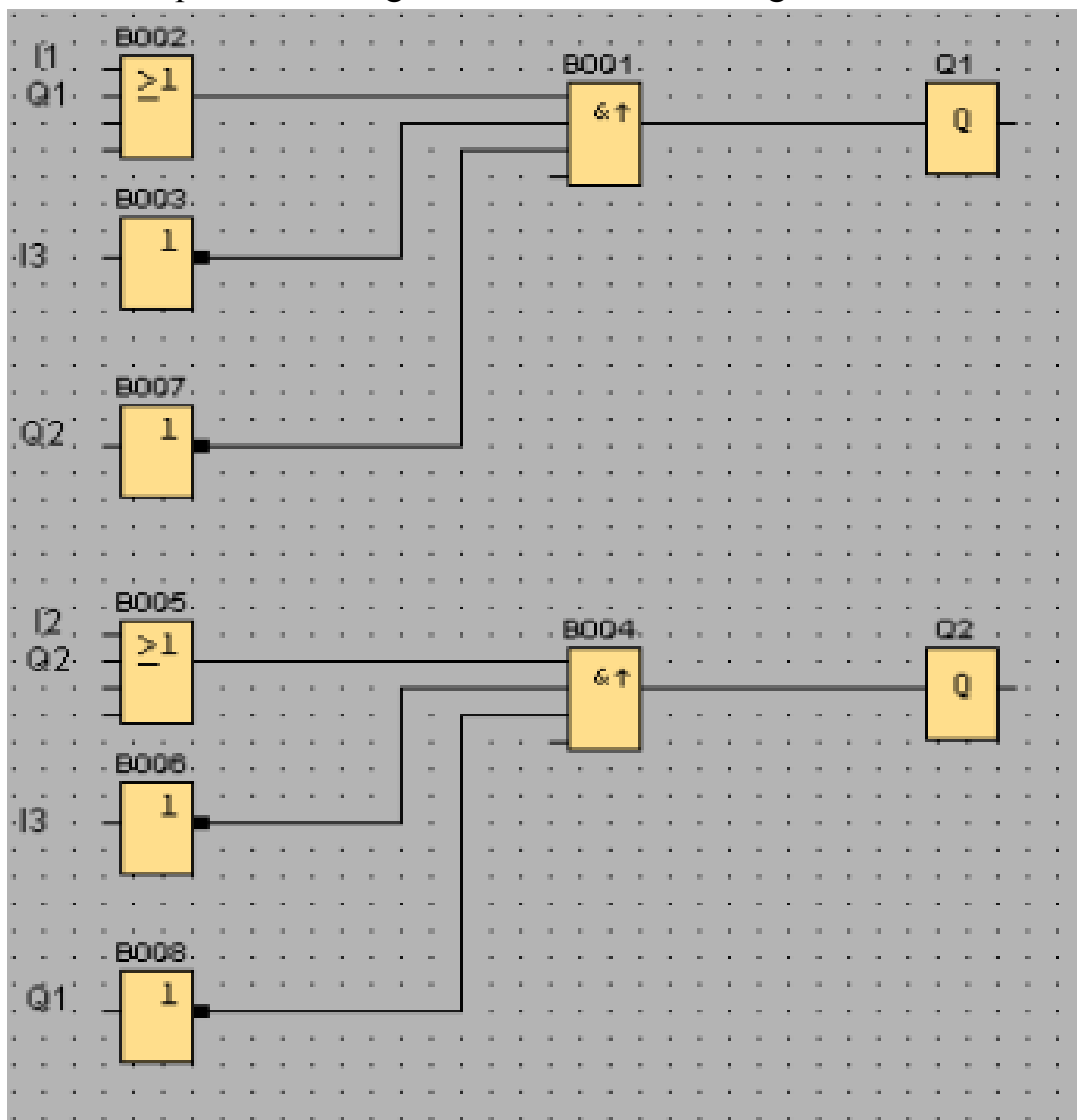
+ Sơ đồ kết nối công tắc tơ



+ Sơ đồ kết nối mạch động lực



Bước 4: Lập trình Chương trình điều khiển trên Logo



Bước 5: Vận hành

TÀI LIỆU THAM KHẢO

- [1]- Nguyễn Trọng Thuận, *Điều khiển logic và ứng dụng*, NXB Khoa học kỹ thuật 2016
- [2]- Trần Thế San, *Hướng dẫn thiết kế mạch và lập trình PLC*, NXB Đà Nẵng 2019
- [3]- Tăng Văn Mùi, *Điều khiển logic lập trình PLC*, NXB Thống kê 2017