

**UBND HUYỆN CỬ CHI
TRƯỜNG TRUNG CẤP NGHỀ CỬ CHI**

GIÁO TRÌNH

MÔN HỌC/MÔ ĐUN: LẬP TRÌNH VÀ ĐIỀU KHIỂN PLC

NGÀNH/NGHỀ: KỸ THUẬT MÁY LẠNH VÀ ĐIỀU HÒA KHÔNG KHÍ

TRÌNH ĐỘ: TRUNG CẤP NGHỀ

*Ban hành kèm theo Quyết định số: 48/QĐ-TCNCC ngày 04 tháng 10 năm 2021 của
Hiệu trưởng Trường Trung Cấp Nghề Cử Chi*

Tp. Hồ Chí Minh, năm 2021

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

Để thực hiện biên soạn giáo trình đào tạo nghề Điện tử công nghiệp ở trình độ Trung Cấp Nghề, giáo trình PLC cơ bản là một trong những giáo trình môn học đào tạo chuyên ngành được biên soạn theo nội dung chương trình khung được Bộ Lao động Thương binh Xã hội và Tổng cục Dạy Nghề phê duyệt. Nội dung biên soạn ngắn gọn, dễ hiểu, tích hợp kiến thức và kỹ năng chặt chẽ với nhau, logic.

Khi biên soạn, nhóm biên soạn đã cố gắng cập nhật những kiến thức mới có liên quan đến nội dung chương trình đào tạo và phù hợp với mục tiêu đào tạo, nội dung lý thuyết và thực hành được biên soạn gắn với nhu cầu thực tế trong sản xuất đồng thời có tính thực tiễn cao.

Nội dung giáo trình được biên soạn với dung lượng thời gian đào tạo 90 giờ gồm các bài:

Bài 1: Điều khiển lập trình PLC S7-1200

Bài 2: Tính toán và các phép toán nhị phân của PLC S7-1200

Bài 3: Lắp đặt mô hình điều khiển bằng plc

Bài 4: Lập trình các lệnh trên PLC S7 1200

Bài 5: Hoạt động của PLC LOGO, Biến tần và Motor Sevor

Trong quá trình sử dụng giáo trình, tùy theo yêu cầu cũng như khoa học và công nghệ phát triển có thể điều chỉnh thời gian và bổ sung những kiến thức mới cho phù hợp. Trong giáo trình, chúng tôi có đề ra nội dung thực tập của từng bài để người học cũng có và áp dụng kiến thức phù hợp với kỹ năng. Tuy nhiên, tùy theo điều kiện cơ sở vật chất và trang thiết bị, các trường có thể sử dụng cho phù hợp.

Tuy nhiên, tùy theo điều kiện cơ sở vật chất và trang thiết bị, các trường có thể sử dụng cho phù hợp. 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.

Lời cảm ơn của các cơ quan liên quan, các đơn vị và cá nhân đã tham gia.

....., ngày.....tháng.....năm.....

Tham gia biên soạn

Mục Lục

	TRANG
Lời nói đầu	2
Lời giới thiệu	3
Bài 1. Thiết bị cấp nhiệt	5
1. Bàn là điện	5
2. Bếp điện	6
3. Nồi cơm điện	9
4. Siêu điện, phích đun nước điện	10
5. Bếp từ	14
6. Lò nướng viba	16
7. Bình nước nóng	17
Bài 2. Máy biến áp gia dụng	18
1. Khái niệm chung	18
2. Cấu tạo, nguyên lý làm việc của máy biến áp	20
3. Sử dụng, sửa chữa máy biến áp một pha thông dụng	27
Bài 3. Động cơ điện gia dụng	36
1. Đại cương	36
2. Cấu tạo, nguyên lý làm việc của động cơ không đồng bộ	37
3. Sử dụng và sửa chữa động cơ điện một pha	44
4. Một số ứng dụng điển hình của động cơ điện	51
Bài 4. Các loại đèn gia dụng và trang trí	57
1. Đèn sợi đốt	59
2. Đèn huỳnh quang	62
3. Đèn thủy ngân	64
4. Các mạch đèn cơ bản	65
Tài liệu tham khảo	69

CHƯƠNG TRÌNH MÔĐUN

Tên môđun: LẬP TRÌNH VÀ ĐIỀU KHIỂN PLC

Mã số mô đun: MĐ 20

Thời gian mô đun: 90 giờ; (Lý thuyết: 10 giờ; Thực hành, thí nghiệm, thảo luận, bài tập : 78 giờ. Kiểm tra: 2 giờ)

I. Vị trí, tính chất của môđun:

- Vị trí: Là mô đun đóng vai trò quan trọng trong các môđun đào tạo nghề. Mô đun này đòi hỏi học sinh phải có khả năng tư duy, kiên trì nắm vững được kiến thức đã được học trong các môn học cơ sở. 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

II. Mục tiêu mô đun:

Học xong mô đun này học sinh có khả năng:

- Kiến thức:

- + Giải thuật phù hợp đơn giản, ngắn gọn.
- + Phân tích, khảo sát các đặc điểm, đặc tính của các loại máy điện nói trên
- + Cấu tạo, cấu trúc chương trình, nguyên tắc nạp trình cho Logo

-Kỹ năng:

- + Nạp trình thành thạo, kiểm tra sửa chữa lỗi khi nạp trình.
 - + Sử dụng đúng các khối chức năng, các lệnh cơ bản (các phép toán nhị phân các phép toán số của PLC, xử lý tín hiệu analog).
 - + Sử dụng, khai thác thành thạo phần mềm mô phỏng. Thực hiện kết nối tốt với PC.
 - + Lắp ráp thành thạo mạch động lực đảm bảo kỹ thuật và an toàn.
 - + Thao tác nạp trình trực tiếp, dùng các phần mềm tương ứng
 - + Kỹ năng kiểm tra, phát hiện sai lỗi của chương trình và sửa chữa khắc phục.
- Năng lực tự chủ và trách nhiệm:
- + Rèn luyện thái độ nghiêm túc, tỉ mỉ, chính xác trong học tập và thực hiện công việc.....

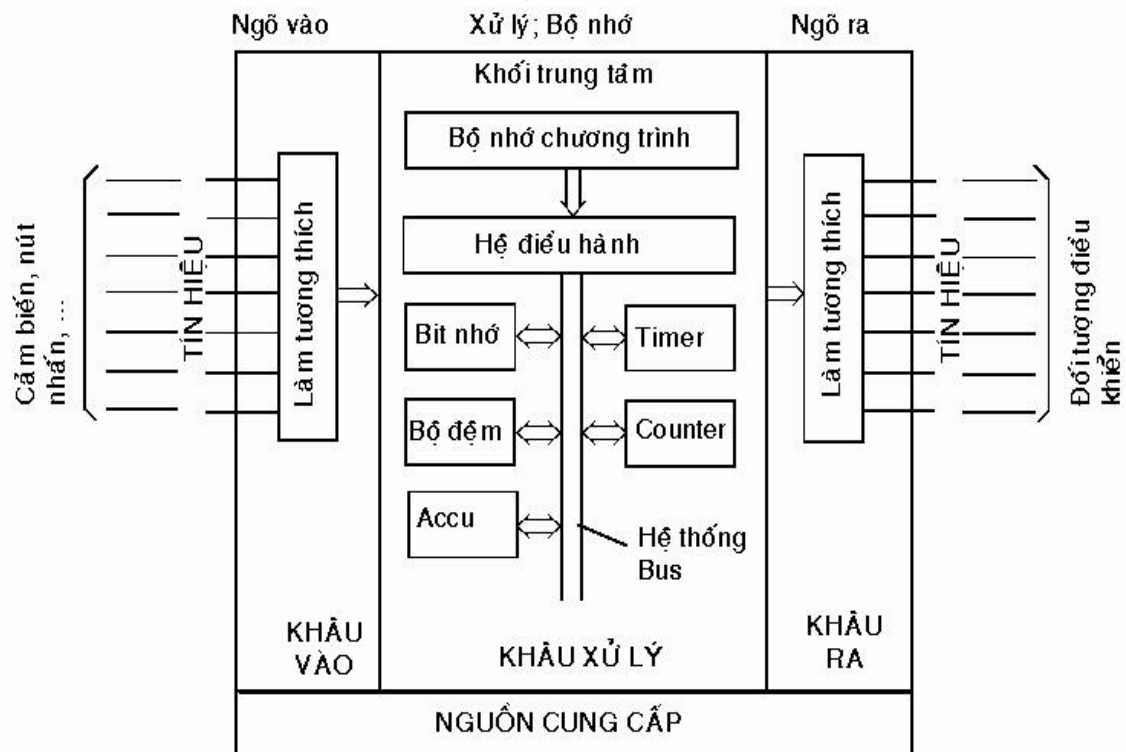
Bài 1: : Điều khiển lập trình PLC S7-1200

Thời gian: 24 giờ

Mục tiêu của bài:

- Trình bày được các ưu điểm của điều khiển lập trình so với các loại điều khiển khác và các ứng dụng của chúng trong thực tế.
- Trình bày được cấu trúc và nhiệm vụ các khối chức năng của PLC.
- Thực hiện được sự kết nối giữa PLC và các thiết bị ngoại vi.
- Lắp đặt được các thiết bị bảo vệ cho PLC theo yêu cầu kỹ thuật
- Rèn luyện đức tính cẩn thận, tỉ mỉ, tư duy sáng tạo và khoa học.

1. TÌM HIỂU VỀ CẤU TRÚC CỦA MỘT PLC



Hình 1.1. Cấu trúc chung của bộ điều khiển lập trình PLC

PLC gồm ba khối chức năng cơ bản: bộ xử lý, bộ nhớ và khối vào/ra. Trạng thái ngõ vào của PLC được phát hiện và lưu vào bộ nhớ đệm. PLC thực hiện các lệnh Logic trên các trạng thái của chúng và thông qua chương trình trạng thái ngõ ra được cập nhật và lưu vào bộ nhớ đệm; sau đó, trạng thái ngõ ra trong bộ nhớ đệm được dùng để đóng mở các tiếp điểm kích hoạt các thiết bị tương ứng. Như vậy, sự hoạt động của các thiết bị được điều khiển hoàn toàn tự động theo chương trình trong bộ nhớ. Chương trình được nạp vào PLC thông qua thiết bị lập trình chuyên dùng.

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

1.1 Xử lý trung tâm (CPU Central Processing Unit)

CPU điều khiển và quản lý mọi hoạt động bên trong PLC. Việc trao đổi thông tin giữa CPU, bộ nhớ và khối vào/ra được thực hiện thông qua hệ thống BUS dưới sự điều khiển của CPU.

** Nguyên lý hoạt động:*

- Thông tin lưu trữ trong bộ nhớ chương trình được gọi lên tuần tự (do đã được điều khiển và kiểm soát bởi bộ đếm chương trình do đơn vị xử lý trung tâm không chế).

- Bộ xử lý liên kết các tín hiệu (dữ liệu) đơn lẻ (theo một quy định nào đó - do thuật toán điều khiển) và rút ra kết quả là các lệnh cho đầu ra.

- Sự thao tác tuần tự của chương trình đi qua một chu trình đầy đủ rồi sau đó lại bắt đầu lại từ đầu, thời gian đó gọi là "thời gian quét".

- Đo thời gian mà bộ xử lý xử lý 1 kbyte chương trình để làm chỉ tiêu đánh giá giữa các PLC.

Như vậy bộ vi xử lý quyết định khả năng và chức năng của PLC.

1.2 Đọc / ghi và bộ nhớ

Bộ nhớ chương trình trong PLC là một bộ nhớ điện tử đặc biệt có thể ghi/ đọc được. Nếu sử dụng bộ nhớ đọc-ghi được (RAM), thì nội dung của nó luôn được thay đổi như trong trường hợp vận hành điều khiển. Trong trường hợp điện áp nguồn bị mất thì nội dung trong RAM có thể vẫn được giữ lại nhờ có sử dụng nguồn Pin dự phòng. Để có các vùng nhớ chuyên dùng khác nhau, tất cả PLC đều dùng các loại bộ nhớ: RAM, ROM, EEPROM.

Bộ nhớ được thiết kế thành dạng modul để cho phép dễ dàng thích nghi với các chức năng điều khiển với các kích cỡ khác nhau. Muốn mở rộng bộ nhớ chỉ cần cắm thẻ nhớ vào rãnh cắm chờ sẵn trên modul CPU.

1.3 Kết nối khối ngõ vào / ra

Hoạt động xử lý tín hiệu bên trong PLC: 5VDC, 15VDC (điện áp cho họ TTL và CMOS). Trong khi đó tín hiệu điều khiển bên ngoài có thể lớn hơn, khoảng 24VDC đến 240VDC hay 110VAC đến 220VAC với dòng lớn.

Khối giao tiếp vào/ra có vai trò giao tiếp giữa mạch vi điện tử của PLC với mạch công suất bên ngoài. Thực hiện chuyển mức điện áp tín hiệu và cách ly bằng mạch ghép quang (Opto-isolator) trên các khối vào/ra. Cho phép tín hiệu nhỏ đi qua và ghim các tín hiệu có mức cao xuống mức tín hiệu chuẩn. Tác dụng chống nhiễu tốt khi chuyển công tắc bảo vệ quá áp từ nguồn cung cấp điện lên đến điện áp 1500V.

+ Ngõ vào: nhận trực tiếp tín hiệu từ cảm biến.

Khi một cảm biến phát hiện một sự thay đổi trạng thái logic thì nó phải truyền trạng thái thay đổi này đến PLC. Tiêu biểu là việc đóng hoặc ngắt dòng điện hay điện áp. Trong một vài trường hợp, ngõ ra của cảm biến sử dụng để đóng mạch trực tiếp cho tải mà không thông qua PLC. Các ngõ ra tiêu biểu của cảm biến là:

- Đóng hoặc ngắt dòng điện

- Đóng hoặc ngắt điện áp
- Chuyển mạch AC
- Sử dụng điện áp 0V và 5V để chỉ thị mức logic.
- + Ngõ ra: là các transistor, role hay triac vật lý

2. ĐIỀU KHIỂN LẬP TRÌNH S7-1200

2.1 Khái quát chung

PLC viết tắt của programmable logic controller là thiết bị điều khiển logic cho phép thực hiện linh hoạt các thuật toán điều khiển logic qua một ngôn ngữ lập trình, bộ điều khiển thỏa mãn các yêu cầu:

- Lập trình dễ dàng vì ngôn ngữ lập trình dễ học;
- Gọn nhẹ, dễ dàng bảo quản, tu sửa;
- Dung lượng bộ nhớ lớn để có thể chứa được những chương trình phức tạp;
- Hoàn toàn tin cậy trong môi trường công nghiệp;
- Giao tiếp với các thiết bị thông tin máy tính, nối mạng các module; mở rộng;
- Giá cả phù hợp.

Bộ điều khiển lập trình PLC được thiết kế nhằm thay thế phương pháp điều khiển truyền thống dùng Rơ le và thiết bị công kênh nó tạo ra một khả năng điều khiển thiết bị dễ dàng và linh hoạt dựa trên việc lập trình trên các lệnh logic cơ bản. PLC còn thực hiện các tác vụ định thì và đếm làm tăng khả năng điều khiển, thực hiện logic được lập trong chương trình và kích ra tín hiệu điều khiển cho thiết bị bên ngoài tương ứng. S7-1200 là thiết bị điều khiển logic khả trình loại nhỏ của hãng Siemens (CHLB Đức), với cấu trúc theo kiến module và có các module mở rộng được sử dụng cho nhiều ứng dụng lập trình khác nhau.

Bộ điều khiển *logic* khả trình (PLC) S7-1200 mang lại tính linh hoạt và sức mạnh để điều khiển nhiều thiết bị đa dạng hỗ trợ các yêu cầu về điều khiển tự động. Sự kết hợp giữa thiết kế thu gọn, cấu hình linh hoạt và tập lệnh mạnh mẽ đã khiến cho S7-1200 trở thành một giải pháp hoàn hảo dành cho việc điều khiển nhiều ứng dụng đa dạng khác nhau.

Kết hợp một bộ vi xử lý, một bộ nguồn tích hợp, các mạch ngõ vào và mạch ngõ ra trong một kết cấu thu gọn, CPU trong S7-1200 đã tạo ra một PLC mạnh mẽ. Sau khi người dùng tải xuống một chương trình, CPU sẽ chứa mạch logic được yêu cầu để giám sát và điều khiển các thiết bị nằm trong ứng dụng. CPU giám sát các ngõ vào và làm thay đổi ngõ ra theo *logic* của chương trình người dùng, có thể bao gồm các hoạt động như *logic* Boolean, việc đếm, định thì, các phép toán phức hợp và việc truyền thông với các thiết bị thông minh khác.

Một số tính năng bảo mật giúp bảo vệ việc truy xuất đến cả CPU và chương trình điều khiển:

- Mỗi CPU cung cấp một sự bảo vệ bằng mật khẩu cho phép người dùng cấu hình việc truy xuất đến các chức năng của CPU.

Người dùng có thể sử dụng chức năng “know-how protection” để ẩn mã nằm trong một khối xác định.

CPU cung cấp một cổng PROFINET để giao tiếp qua một mạng PROFINET. Các *module* truyền thông là có sẵn dành cho việc giao tiếp qua các mạng RS232 hay RS485.

Họ S7-1200 cung cấp một số lượng lớn các *module* tín hiệu và bảng tín hiệu để mở rộng dung lượng của CPU. Người dùng còn có thể lắp đặt thêm các *module* truyền thông để hỗ trợ các giao thức truyền thông khác.

2.1. Địa chỉ các ngõ vào/ ra.

Về cơ bản S7-200 có bốn dạng truy xuất dữ liệu *theo Bit, theo Byte, theo từ đơn(Word) và theo từ kép(Double Word)*. Trong mỗi yêu cầu điều khiển cụ thể chúng ta sẽ chọn truy xuất theo dạng nào.

- Kiểm tra trạng thái của các tín hiệu được tạo ra từ các thiết bị ngoại vi nối với ngõ vào số như nút nhấn, cảm biến, công tắc hành trình... thì sẽ chọn truy xuất là bit, trong trường hợp này thì chọn địa chỉ ngõ vào tương ứng được kết nối ví dụ như I_{0.0}, I_{0.5}, I_{1.1}...

- Xuất tín hiệu ra các cơ cấu chấp hành nhận tín hiệu nhị phân như relay, đèn báo, van từ ... thì sẽ chọn truy xuất là bit, trong trường hợp này thì chọn địa chỉ ngõ ra tương ứng được kết nối ví dụ như Q_{0.0}, Q_{0.2}, Q_{1.0}...

- Nhận tín hiệu từ các cảm biến tạo ra tín hiệu analog như cảm biến nhiệt độ, áp suất, độ ẩm ... thì sử dụng địa chỉ word, ví dụ: AIW₀, AIW₂, AIW₄...

- Xuất tín hiệu analog ra các cơ cấu chấp hành nhận tín hiệu analog như ngõ vào analog biến tần, van tỉ lệ... thì sử dụng địa chỉ word, ví dụ: AQW₀, AQW₂, AQW₄...

- Trong quá trình thực hiện chương trình cần lưu trữ thông tin ở dạng số 16 bit như đếm số sản phẩm (số nguyên 16 bit) thì truy cập địa chỉ word, còn ở dạng 32 bit như nhiệt độ, áp suất (số thực) thì truy cập địa chỉ double word...

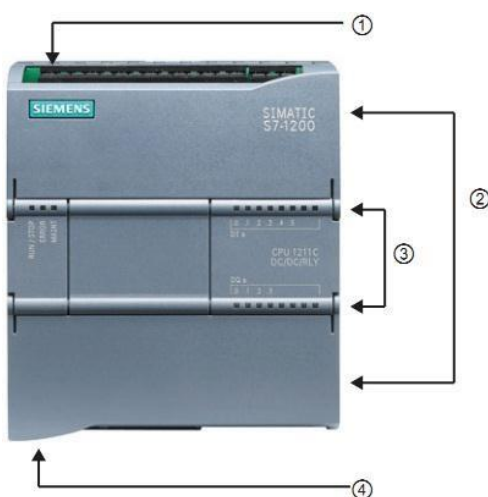
Tóm lại, các ngõ vào và ra của PLC S7-1200 đặt theo thứ tự hệ bát phân. Trong đó CPU có sẵn 14 vào và 10 ra số.

Có thể mở rộng ngõ vào ra của PLC bằng cách ghép nối thêm vào nó các Modul mở rộng phía sau. Địa chỉ của các ngõ vào/ra của mô đun mở rộng thêm được xác định theo vị trí của mô đun trong dãy PLC.

Module		Chỉ ngõ vào	Chỉ ngõ ra	Kết hợp In/Out
Module tín hiệu (SM)	Kiểu số	8 x DC In	8 x DC Out 8 x Relay Out	8 x DC In / 8 x DC Out 8 x DC In / 8 x Relay Out
		16 x DC In	16 x DC Out 16 x Relay Out	16 x DC In / 16 x DC Out 16 x DC In / 16 x Relay Out
	Kiểu tương tự	4 x Analog In 8 x Analog In	2 x Analog In 4 x Analog In	4 x Analog In / 2 x Analog Out

Bảng tín hiệu (SB)	Kiểu số	-	-	2 x DC In / 2 x DC Out
	Kiểu tương tự	-	1 x Analog In	-
Module truyền thông (CM)				
RS485				
RS232				

Bảng 1.1 Địa chỉ vào/ra và modul mở rộng



- ① Bộ phận kết nối nguồn
- ② Các bộ phận kết nối dây của người dùng có thể tháo được (phía sau các nắp che)
- ③ Khe cắm thẻ nhớ nằm dưới cửa phía trên
- ④ Các LED trạng thái dành cho I/O tích hợp
- ⑤ Bộ phận kết nối PROFINET (phía trên của CPU).

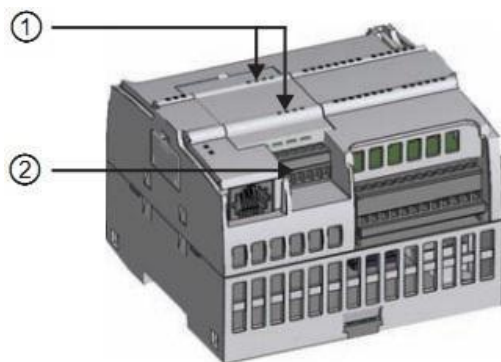
Hình 1.2. Các chú thích trên PLC S7 -1200

Các kiểu CPU khác nhau cung cấp một sự đa dạng các tính năng và dung lượng giúp cho người dùng tạo ra các giải pháp có hiệu quả cho nhiều ứng dụng khác nhau.

2.2. Phần chữ chỉ vị trí và kích thước của ô nhớ.

Một bảng tín hiệu (SB) cho phép người dùng thêm vào I/O cho CPU. Người dùng có thể thêm một SB với cả I/O kiểu số hay kiểu tương tự. SB kết nối vào phía trước của CPU.

- SB với 4 I/O kiểu số (ngõ vào 2 x DC và ngõ ra 2 x DC)
- SB với 1 ngõ ra kiểu tương tự.

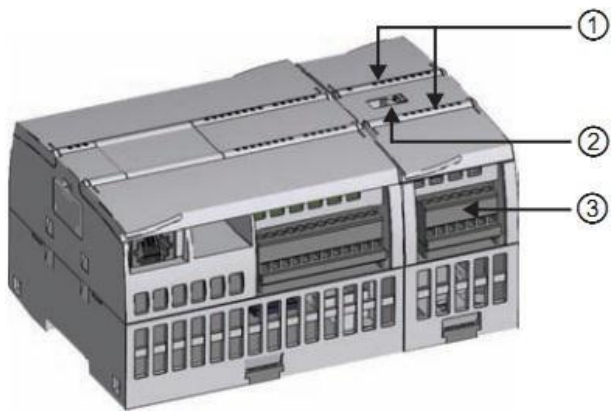


- ① Các LED trạng thái trên SB
- ② Bộ phận kết nối dây của người dùng có thể tháo ra

Hình 1.3. Chú thích kết nối trên PLC S7 -1200

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

Người dùng có thể sử dụng các module tín hiệu để thêm vào CPU các chức năng. Các module tín hiệu kết nối vào phía bên phải của CPU.

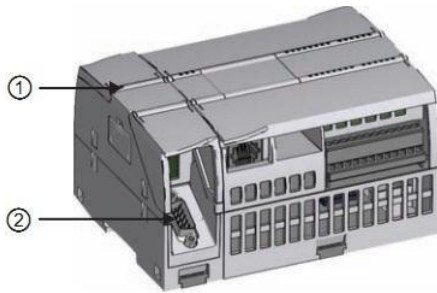


- ① Các LED trạng thái dành cho I/O của module tín hiệu
- ② Bộ phận kết nối đường dẫn
- ③ Bộ phận kết nối nối dây của người dùng có thể tháo ra

Hình 4. Chú thích kết nối tín hiệu trên PLC S7 -1200

Họ S7-1200 cung cấp các *module* truyền thông (CM) dành cho các tính năng bổ sung vào hệ thống. Có 2 *module* truyền thông: RS232 và RS485.

- CPU hỗ trợ tối đa 3 module truyền thông
- Mỗi CM kết nối vào phía bên trái của CPU (hay về phía bên trái của một CM khác)



- ① Các LED trạng thái dành cho module truyền thông
- ② Bộ phận kết nối truyền thông

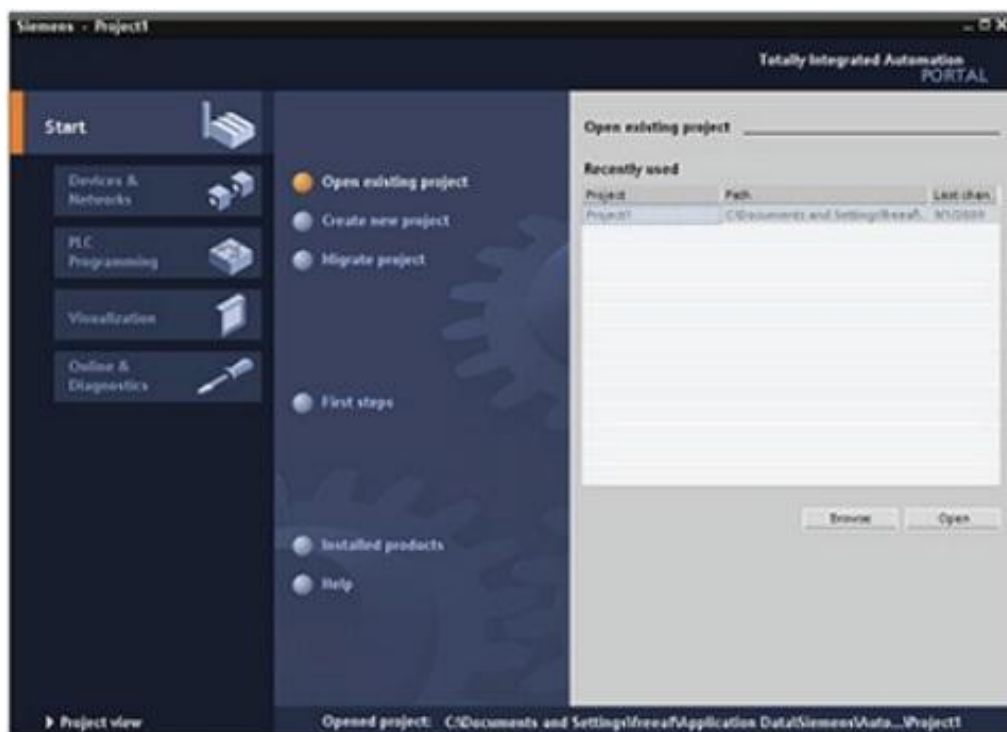
Hình 5. Chú thích kết nối truyền thông trên PLC S7 -1200

Để cài đặt STEP 7 Basic trên một máy tính cá nhân dùng hệ điều hành Windows 2000, Windows XP hay Windows Vista, người dùng cần phải đăng nhập với quyền hạn Administrator.

Nhằm giúp gia tăng hiệu suất công việc, phần Totally Intergrated Automation Portal cung cấp hai kiểu xem thiết lập công cụ khác nhau: một là thiết lập được định hướng theo công việc, thiết lập này được tổ chức trong chức năng của các công cụ (kiểu xem Portal), hai là kiểu xem được định hướng theo đề án gồm các phần tử bên trong đề án (kiểu xem Project). Người dùng cần chọn kiểu xem nào giúp làm việc với hiệu quả tốt nhất. Với một cú nhấp chuột, người dùng có thể chuyển đổi giữa kiểu xem Portal và kiểu xem Project.

Kiểu xem Portal cung cấp một kiểu xem theo chức năng đối với các nhiệm vụ và tổ chức chức năng của các công cụ theo nhiệm vụ để được hoàn thành, như là tạo ra việc cấu hình các thành phần và các mạng phần cứng.

Người dùng có thể dễ dàng xác định cách thức để tiến hành và nhiệm vụ để chọn.



Hình 1.6. Giao diện phần mềm cài đặt PLC S7-1200

Các đèn báo trên S7-1200

- SF (đèn đỏ) : đèn đỏ SF báo hiệu hệ thống bị hỏng. Đèn SF sáng lên khi PLC bị hỏng hóc.
- Run (đèn xanh) : đèn xanh chỉ định PLC đang ở chế độ làm việc và thực hiện chương trình được nạp vào trong máy.
- Stop (đèn vàng): đèn vàng chỉ định PLC đang ở chế độ dừng.
- I_{x.x} (đèn xanh) : đèn xanh ở cổng vào chỉ định ở trạng thái tức thời của cổng I_{x.x} (X.X = 0.0 ÷ 1.5)
- Q_{y.y} (đèn xanh): đèn xanh ở cổng ra báo hiệu trạng thái tức thời của cổng Q_{y.y} (y.y = 0.0 ÷ 1.1)

Chọn chế độ làm việc cho PLC

Công tắc chọn chế độ làm việc nằm ở phía trên, có ba vị trí cho phép chọn các chế độ làm việc khác nhau của PLC:

- RUN: Cho phép PLC thực hiện chương trình trong bộ nhớ. PLC S7-1200 sẽ rời khỏi chế độ RUN và chuyển sang chế độ STOP nếu trong máy có sự cố hoặc trong chương trình gặp lệnh STOP.
- STOP: Cưỡng bức PLC dừng chương trình đang chạy và chuyển sang chế độ STOP. Ở chế độ STOP, PLC cho phép hiệu chỉnh, nạp, xóa một chương trình.
- TERM: Cho phép người dùng từ máy tính quyết định chọn một trong hai chế độ làm việc cho PLC hoặc RUN hoặc STOP.

Ghép nối S7-1200 với máy tính PC qua cổng LAN cần có cáp nối qua cổng LAN/USB của máy tính.

2.4. CẤU TRÚC BỘ NHỚ CỦA S7-1200

Bộ nhớ của S7-1200 được chia thành ba vùng nhớ chính:

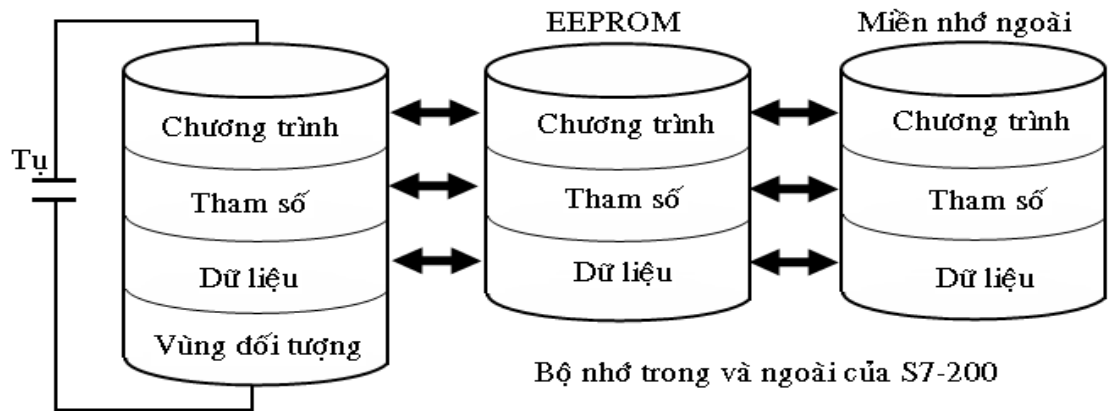
- + **Vùng chứa chương trình ứng dụng**: Vùng chứa chương trình được chia thành 3 miền:
 - MAIN (Organisation block): miền chứa chương trình tổ chức, chứa chương trình chính, các lệnh trong khối này luôn được quét.
 - SBR_0 (Subroutine): Miền chứa chương trình con, được tổ chức thành hàm và có biến hình thức để trao đổi dữ liệu, chương trình con này sẽ được thực hiện khi nó được gọi trong chương trình chính.
 - INT_0 (Interrupt): Miền chứa chương trình ngắt, được tổ chức thành hàm và có khả năng trao đổi dữ liệu với bất cứ 1 khối chương trình nào khác. Chương trình này sẽ được thực hiện khi có sự kiện ngắt xảy ra. Có rất nhiều sự kiện ngắt như: Ngắt thời gian, ngắt xung tốc độ cao...
- + **Vùng chứa tham số của hệ điều hành**: Chia thành 5 miền khác nhau
 - I (Process image input): Miền dữ liệu các cổng vào số, trước khi bắt đầu thực hiện chương trình, PLC sẽ đọc giá trị logic của tất cả các cổng đầu vào và cất giữ chúng trong vùng nhớ I. Thông thường chương trình ứng dụng không đọc trực tiếp trạng thái logic của cổng vào số mà chỉ lấy dữ liệu của cổng vào từ bộ đệm I.
 - Q (Process Image Output): Miền bộ đệm các dữ liệu cổng ra số. Kết thúc giai đoạn thực hiện chương trình, PLC sẽ chuyển giá trị logic của bộ đệm Q tới các cổng ra số. Thông thường chương trình không trực tiếp gán giá trị tới tận cổng ra mà chỉ chuyển chúng tới bộ đệm Q.
 - M (Miền các biến cờ): Chương trình ứng dụng sử dụng những biến này để lưu giữ các tham số cần thiết và có thể truy nhập nó theo Bit (M), byte (MB), từ (MW) hay từ kép (MD).

- T (Timer): Miền nhớ phục vụ bộ thời gian (Timer) bao gồm việc lưu trữ giá trị thời gian đặt trước (PV-Preset Value), giá trị đếm thời gian tức thời (CV –Current Value) cũng như giá trị Logic đầu ra của bộ thời gian.

- C (Counter): Miền nhớ phục vụ bộ đếm bao gồm việc lưu trữ giá trị đặt trước (PV- Preset Value), giá trị đếm tức thời (CV _ Current Value) và giá trị logic đầu ra của bộ đếm.

- DB(Data Block): Miền chứa dữ liệu được tổ chức thành khối. Kích thước cũng như số lượng khối do người sử dụng quy định, phù hợp với từng bài toán điều khiển. Chương trình có thể truy nhập miền này theo từng bit (DBX), byte (DBB), từ (DBW) hoặc từ kép (DBD).

+ **Vùng chứa các khối dữ liệu:**



Hình 1.7 Vùng nhớ dữ liệu

Vùng nhớ	Miêu tả	Ép buộc	Loại giữ
I	Được sao chép từ các ngõ vào vật lý tại điểm bắt đầu của chu trình quét	Không	Không
I:P	Việc đọc ngay lập tức của các điểm ngõ vào trên CPU, SB và SM	Có	Không
Ngõ vào vật lý			
Q	Được sao chép đến các ngõ ra vật lý tại điểm bắt đầu của chu trình quét	Không	Không
Q:P	Việc ghi ngay lập tức đến các điểm ngõ ra vật lý trên CPU, SB và SM	Có	Không
Ngõ ra vật lý			
M	Bộ nhớ dữ liệu và điều khiển	Không	Có
Bộ nhớ bit			
L	Dữ liệu tạm thời cho một khối, một bộ phận của khối đó	Không	Không
Bộ nhớ tạm thời			
DB	Bộ nhớ dữ liệu và còn là bộ nhớ thông số dành cho các FB	Không	Có
Khối dữ liệu			

Bảng 2.2 Địa chỉ vùng nhớ các cpu PLC S7-200

- L (Local data block): Miền dữ liệu địa phương, được các khối chương trình chính, chương trình con, chương trình ngắt tổ chức và sử dụng cho các biến nhập tức thời và trao đổi dữ liệu của biến hình thức với những khối chương trình gọi nó. Nội dung của một khối dữ liệu trong miền nhớ này sẽ bị xoá khi kết thúc chương trình tương ứng trong chương trình chính, chương trình con, chương trình ngắt. Miền này có thể được truy nhập từ chương trình theo bit (L), byte (LB) từ (LW) hoặc từ kép (LD).

3. XỬ LÝ CHƯƠNG TRÌNH

3.1. Vòng quét chương trình.

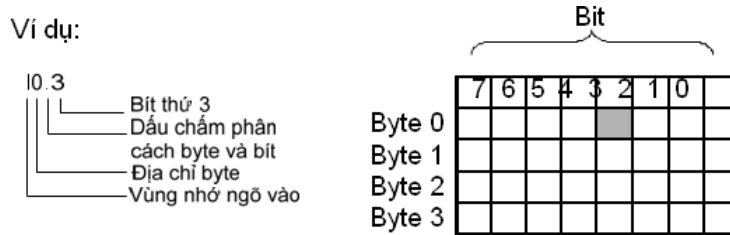
3.1.1. Truy xuất dữ liệu

Việc xử lý chương trình trong S7-1200 thông qua việc truy xuất dữ liệu. PLC có thể truy xuất dữ liệu theo bit, theo byte, theo từ đơn và từ kép.

Truy xuất theo bit

Để truy xuất địa chỉ theo dưới dạng Bit chúng ta xác định vùng nhớ, địa chỉ của Byte và địa chỉ của Bit.

Trong hình là bản đồ vùng nhớ của bộ đệm dữ liệu ngõ vào I(Process Image Input). Bản đồ của các vùng nhớ khác cũng có cấu trúc tương tự như vậy. Bit thấp nhất là bit 0 nằm bên phải và bit cao nhất là bit 7 nằm bên trái. Do đó chúng ta hoàn toàn có thể khai báo tương tự như ví dụ trên, chẳng hạn như: Q1.0, V5.2, M0.1... Dung lượng của các vùng nhớ phụ thuộc vào loại CPU mà chúng ta sử dụng.



Hình 1.8: Vùng nhớ ngõ vào I

Truy xuất theo byte (8 bit)

Khi truy xuất dữ liệu theo byte, chúng ta xác định vùng nhớ, và thứ tự của byte cần truy xuất.



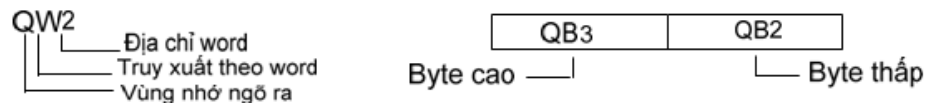
Hình 1.9: Truy xuất vùng nhớ byte

Tương tự như ví dụ ta khai báo cho các vùng nhớ khác, chẳng hạn như IB3, MB2, QB5.

Truy xuất theo word (16 bit)

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

Ví dụ:



Hình 1.10 Truy xuất vùng nhớ word

Đối với tín hiệu tương tự (Analog) thì chúng ta chỉ có một dạng truy xuất duy nhất là truy xuất theo word. Điều này là do mỗi tín hiệu tương tự sẽ ứng với một giá trị số nguyên 16 bit. Ví dụ: AIW0, AIW2, AQW0....

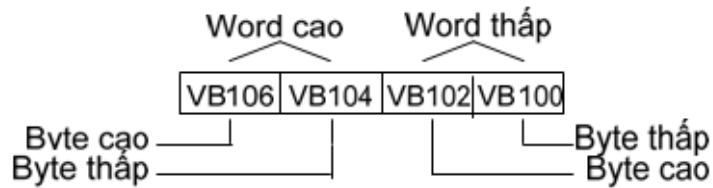
Khi truy xuất địa chỉ theo word thì hai word liền kề nhau bắt buộc cách nhau 2 byte. Ví dụ ta cần chứa 2 dữ liệu dạng số interger vào vùng biến V, thì dữ liệu thứ nhất giả sử chứa vào VW20 thì word kế tiếp lưu dữ liệu thứ hai là VW22.

Truy xuất theo 2 word (Double word = 32 bit)

Khi truy xuất vùng nhớ 32 bit, tương ứng với 4 byte. Trong đó gồm có word thấp, word cao

và byte thấp, byte cao.

Ví dụ: VD100



Hình 1.11 Truy xuất vùng nhớ Double word

Tóm lại, về cơ bản chúng ta có bốn dạng truy xuất dữ liệu như trên. Trong mỗi yêu cầu điều khiển cụ thể chúng ta sẽ chọn truy xuất theo dạng nào.

- Kiểm tra trạng thái của các tín hiệu được tạo ra từ các ngoại vi nối với ngõ vào số như nút nhấn, cảm biến, công tắc hành trình... thì sẽ chọn truy xuất là bit, trong trường hợp này thì chọn địa chỉ ngõ vào tương ứng được kết nối ví dụ như I0.0, I0.5, I1.1...

- Xuất tín hiệu ra các cơ cấu chấp hành nhận tín hiệu nhị phân như relay, đèn báo, van từ ... thì sẽ chọn truy xuất là bit, trong trường hợp này thì chọn địa chỉ ngõ ra tương ứng được kết nối ví dụ như Q0.0, Q0.2, Q1.0....

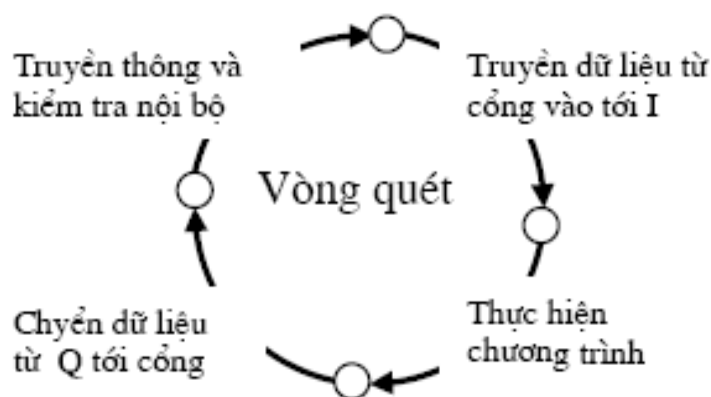
- Nhận tín hiệu từ các cảm biến tạo ra tín hiệu analog như cảm biến nhiệt độ, áp suất, độ ẩm ... thì sử dụng địa chỉ word, ví dụ: AIW0, AIW2, AIW4....

- Xuất tín hiệu analog ra các cơ cấu chấp hành nhận tín hiệu analog như ngõ vào analog biến tần, van tỉ lệ ... thì sử dụng địa chỉ word, ví dụ: AQW0, AQW2, AQW4...

- Trong quá trình thực hiện chương trình cần lưu trữ thông tin ở dạng số 16 bit như đếm số sản phẩm (số nguyên 16 bit) thì truy cập địa chỉ word, còn ở dạng 32 bit như nhiệt độ, áp suất (số thực) thì truy cập địa chỉ double word...

3.1.2. Vòng quét chương trình.

PLC thực hiện chương trình theo chu trình lặp, mỗi vòng lặp được gọi là vòng quét. Mỗi vòng quét bắt đầu bằng giai đoạn đọc dữ liệu từ các ngõ vào (contact, sensor, relay...) vào vùng bộ đệm ảo, tiếp theo là giai đoạn thực hiện chương trình. Trong từng vòng quét, chương trình được thực hiện bằng lệnh đầu tiên và kết thúc tại lệnh cuối cùng. Sau đó là giai đoạn truyền thông nội bộ và kiểm tra lỗi. Vòng quét được kết thúc bằng giai đoạn chuyển các nội dung của bộ đệm ảo tới các ngõ ra.



Hình 1.12 Quá trình hoạt động của một vòng quét

Như vậy, tại thời điểm thực hiện lệnh vào/ra, PLC sẽ không trực tiếp làm việc với cổng vào/ra mà thông qua bộ đệm ảo của cổng trong vùng tham số. Việc truyền thông giữa bộ đệm ảo với thiết bị ngoại vi trong giai đoạn 1 và 4 là do CPU quản lý.

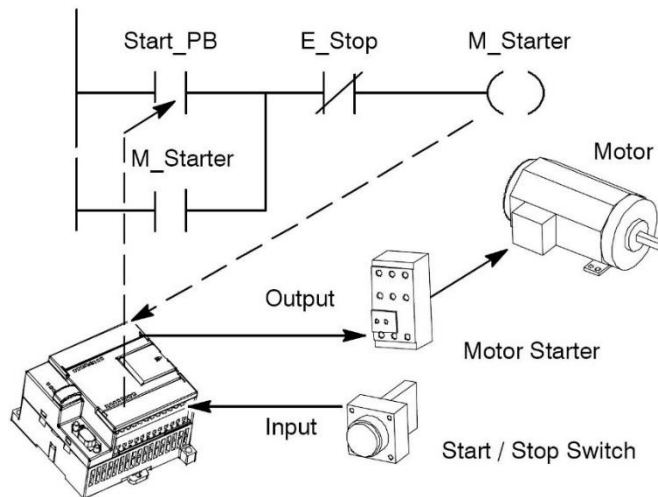
Thường việc thực hiện một vòng quét xảy ra với một thời gian rất ngắn, một vòng quét đơn có thời gian thực hiện từ 1ms tới 100ms. Việc thực hiện một chu kỳ quét dài hay ngắn còn phụ thuộc vào độ dài của chương trình và cả mức độ giao tiếp giữa PLC với các thiết bị ngoại vi. Vì

xử lý có thể đọc được tín hiệu ở ngõ vào, chỉ khi nào tín hiệu này tác động với khoảng thời gian lớn hơn một chu kỳ quét thì vi xử lý coi như không có tín hiệu này. Tuy nhiên trong thực tế sản xuất, thường các hệ thống chấp hành là các hệ thống cơ khí nên tốc độ quét như trên có thể đáp ứng được các chức năng của dây chuyền sản xuất

Tóm lại, hoạt động của S7-1200 rất đơn giản và thực hiện theo trình tự sau:

- Đọc trạng thái các ngõ vào.
- S7-1200 sử dụng các ngõ vào này để thực hiện logic điều khiển theo chương trình được lưu trữ trong nó. Dữ liệu luôn được cập nhật khi chương trình được thực hiện.
- Xuất dữ liệu ra ngõ ra.

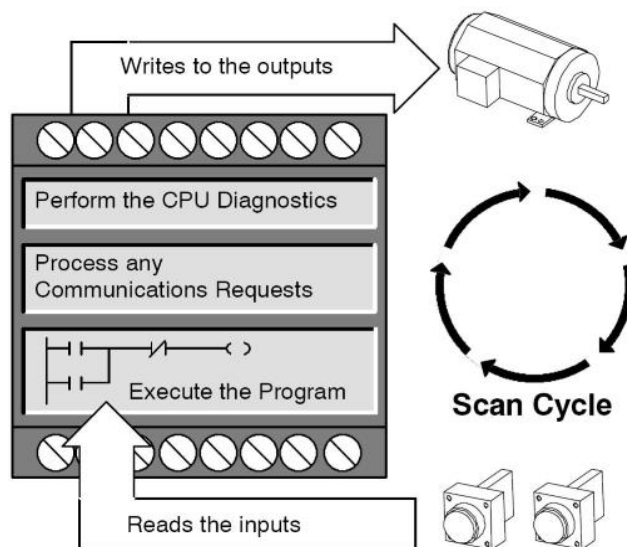
Hình 1.13 là một sơ đồ đơn giản chỉ mối quan hệ giữa sơ đồ điện và PLC S7-1200. Các nút nhấn khởi động/dừng động cơ được kết nối với ngõ vào.



Hình 1.13 Điều khiển ngõ vào và ra

Trạng thái của các ngõ vào tùy thuộc vào nút nhấn. Các trạng thái của ngõ vào sẽ quyết định trạng thái của ngõ ra. Ngõ ra được kết nối với contactor. Tùy theo trạng thái của ngõ ra mà contactor có điện hay mất điện và tương ứng động cơ sẽ hoạt động hay dừng.

Hình 1.14 là ví dụ một chu kỳ quét. S7-1200 thực hiện các nhiệm vụ sau trong một chu kỳ quét:



Hình 1.14 Chu kỳ quét S7-1200

***Đọc ngõ vào:** S7-1200 sao chép trạng thái của các ngõ vào vật lý và chứa trong bộ đệm ngõ vào.

- *Digital inputs:* Mỗi chu kỳ quét bắt đầu bằng cách đọc giá trị hiện hành các ngõ vào số và sau đó ghi các giá trị này vào vùng đệm ngõ vào.

- *Analog inputs*: S7-1200 không cập nhật các ngõ vào analog từ các module mở rộng nếu là chu kỳ quét bình thường trừ khi có kích hoạt khâu lọc các ngõ vào analog (xem chương xử lý tín hiệu analog). Bộ lọc analog được cung cấp cho phép ta có một tín hiệu ổn định hơn. Có thể cho phép bộ analog ở mỗi điểm ngõ vào analog. Khi một ngõ vào analog được kích hoạt ở bộ lọc, S7-1200 cập nhật ngõ vào analog mỗi một lần trong chu kỳ quét và lưu trữ giá trị lọc. Giá trị lọc được cung cấp mỗi khi truy cập ngõ vào analog. Khi bộ lọc analog không được kích hoạt, S7-1200 đọc giá trị ngõ vào analog từ module mở rộng mỗi lần chương trình truy xuất ngõ vào analog.

* **Thực hiện theo logic điều khiển trong chương trình**: S7-1200 thực hiện các lệnh trong chương trình và lưu giá trị vào vùng nhớ.

Khi thực hiện chu kỳ quét, S7-1200 thi hành từ lệnh đầu tiên cho đến lệnh cuối cùng. Các lệnh truy cập I/O tức thì cho phép ta truy xuất ngay lập tức các ngõ vào và ngõ ra khi thực hiện chương trình cũng như chương trình ngắt (interrupt routine). Nếu có sử dụng các ngắt trong chương trình (chương trình ngắt được gọi bởi các yêu cầu ngắt) thì nó không được thực hiện ở chu kỳ quét bình thường. Nó được thực hiện khi có sự kiện ngắt (có thể xảy ra tại bất kỳ thời điểm nào trong chu kỳ quét).

Xử lý bất kỳ yêu cầu truyền thông nào: S7-1200 thi hành bất kỳ nhiệm vụ được yêu cầu cho truyền thông.

Trong giai đoạn xử lý thông tin của chu kỳ quét, S7-1200 xử lý bất kỳ thông tin nào nhận được từ cổng truyền thông hoặc từ các module truyền thông (intelligent I/O module).

Thực hiện tự chẩn đoán CPU: S7-1200 tự kiểm tra để đảm bảo phần firmware, bộ nhớ chương trình, và bất kỳ các module mở rộng nào cũng đang làm việc đúng. Trong giai đoạn này, S7-1200 kiểm tra cho hoạt động thích hợp của CPU và trạng thái của bất kỳ module mở rộng nào.

Xuất ra ngõ ra: Các giá trị được lưu trong vùng đệm ngõ ra sẽ được xuất ra các ngõ ra vật lý. Tại cuối mỗi chu kỳ, S7-1200 xuất các giá trị được lưu trong bộ đệm ngõ ra đến các ngõ ra số (Các ngõ ra analog thì được cập nhật ngay lập tức, không phụ thuộc vào chu kỳ quét).

Việc thực hiện chương trình còn tùy thuộc vào S7-1200 đang ở chế độ STOP hay chế độ RUN. Ở chế độ RUN thì chương trình được thực hiện; còn ở chế độ STOP thì chương trình không được thực hiện.

3.2. Cấu trúc chương trình

Chương trình trong S7-1200 được lưu trong bộ nhớ của PLC ở vùng giành riêng cho chương trình và có thể được lập với 2 dạng cấu trúc khác nhau.

Lập trình tuyến tính: toàn bộ chương trình nằm trong một khối trong bộ nhớ. Loại hình cấu trúc tuyến tính này phù hợp với những bài toán tự động nhỏ, không phức tạp. Khối được chọn phải là khối OB1, là khối mà PLC luôn quét và thực hiện các lệnh trong đó thường xuyên, từ lệnh đầu tiên đến lệnh cuối cùng và quay lại lệnh đầu tiên. Chương trình này gọi là **chương trình chính**.

Lập trình có cấu trúc: Chương trình được chia thành những phần nhỏ và mỗi phần thực thi những nhiệm vụ chuyên biệt riêng của nó, từng phần này nằm trong những khối chương trình khác nhau. Loại hình cấu trúc này phù hợp với những bài toán điều khiển nhiều nhiệm vụ và phức tạp.

Với lập trình có cấu trúc PLC S7-1200 có 3 loại khối cơ bản sau:

- Loại khối OB1 (Organization Block): Khối tổ chức và quản lý chương trình điều khiển. Khối này luôn luôn được thực thi, và luôn được quét trong mỗi chu kỳ quét.

- Loại khối SBR (Khối chương trình con): Khối chương trình với những chức năng riêng. Chương trình này sẽ được thực thi khi có lệnh gọi từ chương trình chính.

- Loại khối INT (Khối chương trình ngắt): Là loại khối chương trình đặc biệt có khả năng trao đổi 1 lượng dữ liệu lớn với các khối chương trình khác. Chương trình này sẽ được thực thi mỗi khi có sự kiện ngắt xảy ra.

Một chương trình ứng dụng có thể có nhiều khối chương trình con và các khối chương trình con này được phân biệt với nhau bằng tên của chương trình con đó.

3.3. Phương pháp lập trình PLC S7-1200

Chương trình của PLC bao gồm một dãy các tập lệnh. PLC S7-1200 thực hiện chương trình bắt đầu từ lệnh lập trình đầu tiên và kết thúc ở lập trình cuối trong một vòng quét.

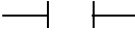
Cách lập trình cho PLC S7-1200 nói riêng và cho các PLC nói chung dựa trên hai phương pháp cơ bản. Phương pháp hình thang (Ladder, viết tắt là LAD) và phương pháp liệt kê lệnh (Statement List, viết tắt là STL).

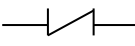
Nếu có một chương trình viết dưới dạng LAD, thiết bị lập trình sẽ tự động tạo ra một chương trình theo dạng STL tương ứng. Tuy nhiên không phải mọi chương trình viết dưới dạng STL đều có thể chuyển sang được dạng LAD.

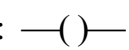
Phương pháp LAD:

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

➤ Tiếp điểm: là biểu tượng (Symbol) mô tả các tiếp điểm của rơle.

- Tiếp điểm thường hở 

- Tiếp điểm thường đóng 

➤ Cuộn dây (coil):  là biểu tượng mô tả rơle, được mắc theo chiều dòng điện cung cấp cho rơ le.

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

➤ Mạng LAD: Là đường nối các phần tử thành một mạch hoàn thiện, đi từ đường nguồn bên trái sang đường nguồn bên phải. Đường nguồn bên trái là dây pha, đường nguồn bên phải là dây trung tính và cũng là đường trở về nguồn cung cấp. Dòng điện chạy từ trái qua tiếp điểm đến đóng các cuộn dây hoặc các hộp trở về bên phải nguồn.

Phương pháp liệt kê lệnh (STL):

Phương pháp liệt kê lệnh (STL) là phương pháp thể hiện chương trình dưới dạng tập hợp các câu lệnh. Mỗi câu lệnh trong chương trình biểu diễn một chức năng của PLC.

Ngăn xếp logic:

Để tạo ra chương trình dạng STL, cần hiểu rõ cách sử dụng 9 bit ngăn xếp của S7-200. Ngăn xếp logic là 1 khối gồm 9 bit chồng lên nhau. Các lệnh liên quan đến ngăn xếp sẽ làm việc với 1 hoặc 2 bit đầu tiên của ngăn xếp. Giá trị logic mới đều có thể gởi vào ngăn xếp. Khi phối hợp 2 bit đầu tiên của ngăn xếp thì ngăn xếp sẽ được kéo lên 1 bit.

4. Kết nối dây giữa PLC và các thiết bị ngoại vi.

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

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

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

CPU S7-1200 có một cổng PROFINET được tích hợp, hỗ trợ cả tiêu chuẩn truyền thông Ethernet và dựa trên TCP/IP. Các giao thức ứng dụng sau đây được hỗ trợ bởi CPU S7-1200:

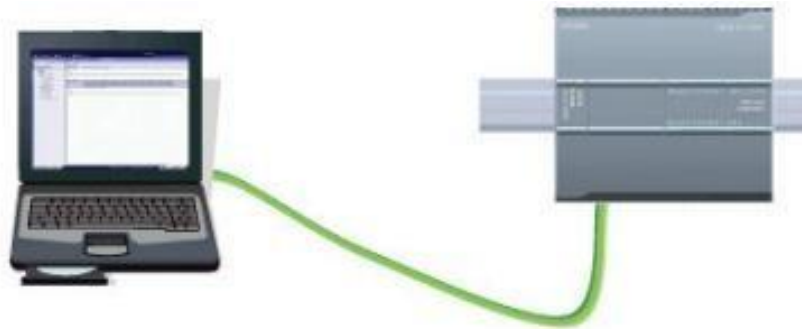
Giao thức điều khiển vận chuyển (Transport Control Protocol – TCP) ISO trên TCP (RFC 1006)

CPU S7-1200 có thể giao tiếp với các CPU S7-1200 khác.

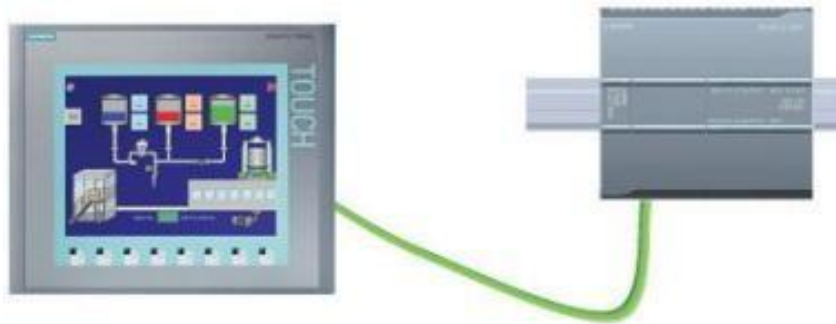
Basic, với các thiết bị HMI, và với các thiết bị không phải của Siemens bằng cách sử dụng các giao thức truyền thông TCP tiêu chuẩn. Có hai cách để giao tiếp sử dụng PROFINET:

Kết nối trực tiếp: sử dụng kết nối trực tiếp khi ta đang sử dụng một thiết bị lập trình, HMI hay một CPU khác được kết nối đến một CPU riêng lẻ.

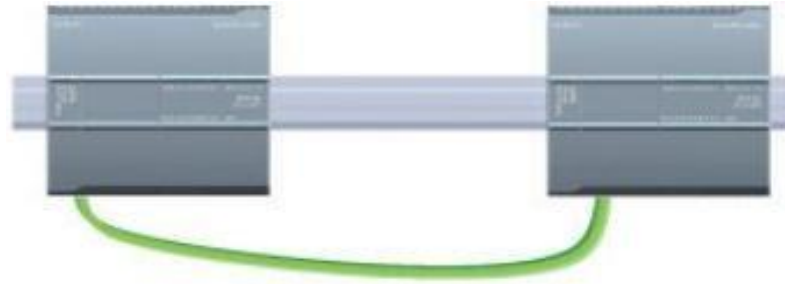
Kết nối mạng: sử dụng các truyền thông mạng khi ta đang kết nối với hơn hai thiết bị (ví dụ các CPU, HMI, các thiết bị lập trình, và các thiết bị không phải của Siemens).



Hình 1.15 Kết nối trực tiếp: thiết bị lập trình được kết nối đến CPU S7-1200



Hình 1.16 Kết nối trực tiếp: HMI được kết nối đến CPU S7-1200



Hình 1.17 Kết nối trực tiếp hai CPU S7-1200

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

Kết nối các ngõ ra số với ngoại vi

Các ngõ ra của PLC có thể được chế tạo là một khối riêng, hoặc kết hợp với các ngõ ra chung trong một khối hoặc được tích hợp trên khối CPU.

Trong trường hợp nào cũng vậy, các ngõ ra cũng phải được cung cấp nguồn riêng với cấp điện áp tùy thuộc vào loại ngõ ra. Cần lưu ý trong một khối ra cũng như các ngõ ra được tích hợp sẵn trên CPU có thể có các nhóm được cung cấp nguồn độc lập nhau. Vì vậy cần lưu ý khi cấp nguồn cho các nhóm này. Nguồn cung cấp cho các khối ra của họ S7-1200 có thể là:

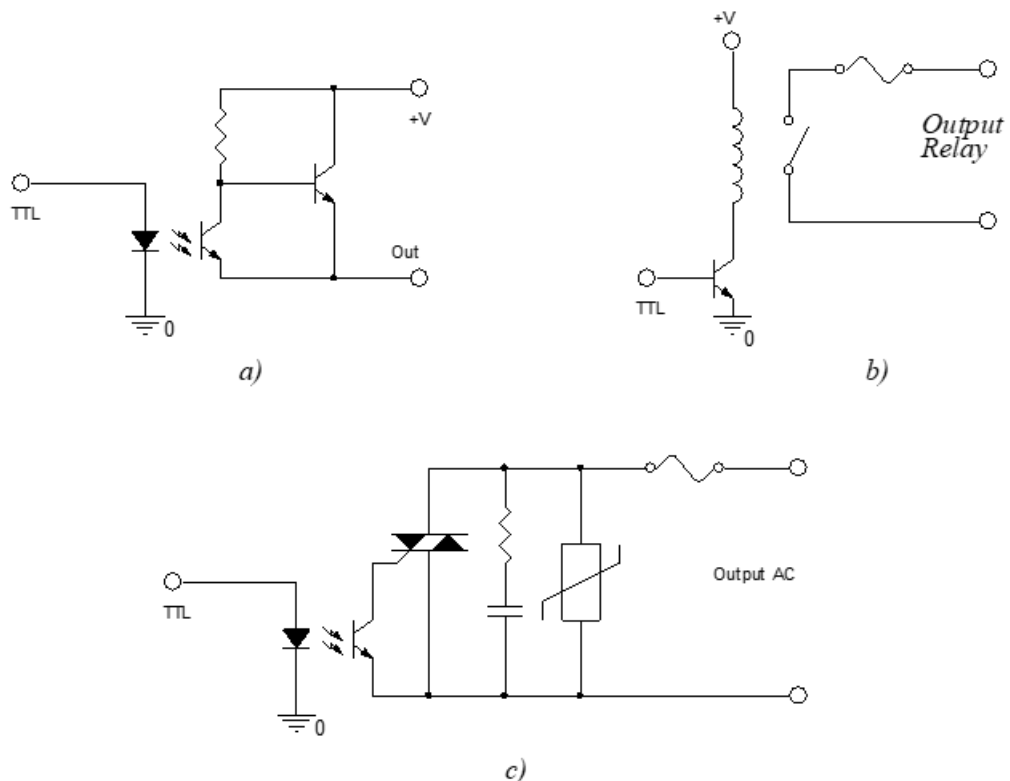
Xoay chiều: $20 \dots 264V_{AC}$, $f = 47 \dots 63Hz$

Một chiều : $5 \dots 30V_{DC}$ đối với ngõ ra Rơle

$20,4 \dots 28,8V_{DC}$ đối với ngõ ra Transistor

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

Sơ đồ mạch điện bên trong của các ngõ ra được cho như hình 1.18.



Hình 1.18 Các mạch điện khác nhau bên trong ngõ ra của PLC

a) Ngõ ra transistor b) Ngõ ra relay c) Ngõ ra triac

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

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

Một chú ý quan trọng khi kết nối các ngõ ra cần tra cứu sổ tay khối ngõ ra hiện có để có được thông tin chính xác tránh được những sự cố đáng tiếc xảy ra. Hình 1.15 là ví dụ của CPU. Ta nhận thấy mỗi một nhóm ngõ vào cũng như một nhóm ngõ ra và CPU được cung cấp nguồn riêng là 24 Vdc. Ngoài ra trên khối CPU còn có nguồn phụ 24 Vdc (đến 280 mA) có thể được sử dụng để cung cấp cho các cảm biến hoặc khối mở rộng.

Ngõ vào DC:

- Điện áp DC thường thấp do đó an toàn hơn.
- Đáp ứng ngõ vào DC rất nhanh.
- Điện áp DC có thể được kết nối với nhiều phần tử trong hệ thống điện.

Ngõ vào AC:

- Ngõ vào AC yêu cầu cần phải có thời gian. Ví dụ đối với điện áp có tần số 50 Hz phải yêu cầu thời gian đến 1/50 giây mới nhận biết được.
- Tín hiệu AC ít bị nhiễu hơn tín hiệu DC, vì vậy chúng thích hợp với khoảng cách lớn và môi trường nhiễu (từ).
- Nguồn AC kinh tế hơn.
- Tín hiệu AC thường được sử dụng trong các thiết bị tự động.

Đối với các ngõ vào số, khi kết nối với thiết bị ngoại vi, ngoại trừ các trường hợp đặc biệt thì thông thường mỗi một ngõ vào được kết nối với một bộ tạo tín hiệu nhị phân như: nút nhấn, công tắc, cảm biến tiếp cận....

5. KIỂM TRA VIỆC NỐI DÂY BẰNG PHẦN MỀM

5.1. Status Chart

Một công việc quan trọng cho người lắp đặt và vận hành là biết được các kết nối của các ngõ vào/ra với ngoại vi có đúng hay không trước khi nạp chương trình điều khiển vào CPU. Hoặc khi một hệ thống đang hoạt động bình thường nhưng một sự cố hư hỏng xảy ra thì các phần ngoại vi nào bị hư và phát hiện nó bằng cách nào. Các phần mềm cho các bộ điều khiển bằng PLC thường có trang bị thêm công cụ để kiểm tra việc kết nối dây ngõ vào/ra với ngoại vi. Trong phần mềm Step7 Micro/Win (phần mềm lập trình cho họ S7-1200) có trang bị thêm phần này đó là mục Status Chart.

Chúng ta có thể sử dụng Status Chart để đọc, ghi hoặc cưỡng bức các biến trong chương trình theo mong muốn. Để có thể mở Status Chart, ta nhấp đúp chuột vào biểu tượng trong cửa sổ Navigation Bar trên màn hình hoặc vào mục View → Component → Status Chart. Giám sát và thay đổi biến với Status Chart.

Để đọc hay ghi các biến chúng ta thực hiện theo các bước sau:

Bước 1: Ở ô đầu tiên trong cột Address ta nhập vào địa chỉ hay tên ký hiệu của một biến trong chương trình ứng dụng mà muốn giám sát hoặc điều khiển, sau đó ấn ENTER. Lặp lại bước này cho tất cả các biến được thêm vào biểu đồ.

Bước 2: Nếu biến là 1 Bit (ví dụ: I, Q, hoặc M), thì kiểu biến đặt ở cột Format là bit. Nếu biến là một byte, word, hay double word thì chọn ở cột Format và nhấp đúp chuột để tìm kiểu biến mong muốn.

Bước 3: Để xem giá trị hiện hành của các biến trong PLC trong biểu đồ, hãy nhấp chuột vào biểu tượng hoặc chọn Debug → Chart Status. Để chụp được một giá trị của các biến tại thời điểm nhấp chuột sử dụng Debug → Single Read hoặc nhấp chuột vào biểu tượng

Bước 4: Để dừng việc giám sát thì nhấp chuột vào biểu tượng hoặc chọn Debug → Chart Status.

Bước 5: Để thay đổi giá trị của một biến hoặc nhiều biến, hãy nhập giá trị mới vào cột “New Value” cho các biến mong muốn và nhấp chuột vào biểu tượng hoặc chọn Debug → Write All để ghi tất cả các giá trị này vào các biến tương ứng trong CPU.

Cưỡng bức biến với Status Chart

Trong một số trường hợp cần thiết phải ép buộc một ngõ vào hoặc một ngõ ra hoặc bất kỳ một biến nào đó trong chương trình theo một giá trị mong muốn cho phù hợp với hoàn cảnh hoạt động hiện tại của hệ thống hoặc để kiểm tra các lỗi xảy ra trong hệ thống điều khiển, ta có thể sử dụng công cụ cưỡng bức biến (Force).

5.2. Đọc và thay đổi biến với Status Chart.

Sau khi kết nối dây ngoại vi với các ngõ vào/ra của PLC, việc kế tiếp là kiểm tra lại kết nối dây này để phát hiện ra các lỗi kết nối. Một công cụ hữu hiệu là sử dụng Status Chart. Lưu ý khi kiểm tra kết nối dây:

Đối với ngõ vào:

- Các ngõ vào nào được nối với các tiếp điểm thường đóng hay tín hiệu có mức logic “1” thì các ngõ vào có điện áp và đèn báo trạng thái các ngõ vào sáng. Khi quan sát trong status chart, ta sẽ nhận thấy các giá trị này có mức logic “1”.

- Việc kiểm tra các ngõ vào nên thực hiện lần lượt cho từng ngõ vào theo bảng kết nối dây vào/ra với ngoại vi. Có nghĩa là mỗi lần ta chỉ thay đổi trạng thái của một bộ tạo tín hiệu (nút nhấn, cảm biến,...) và quan sát trạng thái của ngõ vào được kết nối với nó trong status chart.

- Ghi chép lại các kết nối bị sai và sửa chữa.

Đối với ngõ ra:

- Ở trạng thái bình thường khi chưa có chương trình thì tất cả các ngõ ra của PLC đều ở mức logic “0” (không có điện áp) và đèn báo trạng thái các ngõ ra đều tắt.

- Việc kiểm tra nối dây ngõ ra nên thực hiện lần lượt từng ngõ ra theo bảng kết nối dây bằng cách cho ngõ ra muốn kiểm tra lên mức logic “1” trong status chart và quan sát trạng thái của ngoại vi được kết nối tương ứng. Nếu ngoại vi tương ứng có điện chứng tỏ nó được kết nối đúng còn ngược lại kết nối sai.

- Ghi chép lại các kết nối sai và sửa chữa.

6. Cài đặt và sử dụng phần mềm

6.1. Những yêu cầu đối với máy tính PC

Để việc cài đặt được nhanh chóng chúng ta cần một máy tính có cấu hình cao hơn nhiều so với phần mềm S7-200. CPU máy tính I5 trở lên, Bộ nhớ 100 GB, RAM 4G trở lên

Phần mềm TIA Portal viết tắt của Totally Integrated Automation Portal là một phần mềm tổng hợp của nhiều phần mềm điều hành quản lý tự động hóa, vận hành điện của hệ thống. Có thể hiểu, TIA Portal là phần mềm tự động hóa đầu tiên, có sử dụng chung 1 môi trường/ nền tảng để thực hiện các tác vụ, điều khiển hệ thống.

TIA Portal được phát triển vào năm 1996 bởi các kỹ sư của Siemens, nó cho phép người dùng phát triển và viết các phần mềm quản lý riêng lẻ một cách nhanh chóng, trên 1 nền tảng thống nhất. Giải pháp giảm thiểu thời gian tích hợp các ứng dụng riêng biệt để thống nhất tạo hệ thống.

TIA Portal - Tích hợp tự động toàn diện là phần mềm cơ sở cho tất cả các phần mềm khác phát triển: Lập trình, tích hợp cấu hình thiết bị trong dải sản phẩm. Đặc điểm TIA Portal cho phép

các phần mềm chia sẻ cùng 1 cơ sở dữ liệu, tạo nên tính thống nhất, toàn vẹn cho hệ thống ứng dụng quản lý, vận hành.

6.2. Cài đặt phần mềm lập trình

TIA Portal tạo môi trường dễ dàng để lập trình thực hiện các thao tác:

- Thiết kế giao diện kéo thả thông tin dễ dàng, với ngôn ngữ hỗ trợ đa dạng.
- Quản lý phân quyền User, Code, Project tổng quát.

- Thực hiện go online và Diagnostic cho tất cả các thiết bị trong project để xác định bệnh, lỗi hệ thống.

- Tích hợp mô phỏng hệ thống.

- Dễ dàng thiết lập cấu hình và liên kết giữa các thiết bị Siemens.

- Hiện tại phần mềm TIA Portal có nhiều phiên bản như TIA Portal V14, TIA Portal V15, TIA Portal V16 và mới nhất là TIA Portal V17. Tùy theo nhu cầu sử dụng mà người dùng sẽ lựa chọn cài đặt TIA portal phiên bản tương ứng.



Hình 1.19 Giao diện phần mềm

Tải xuống các phần tử của chương trình.

Ta có thể tải xuống các phần tử của đề án từ các thiết bị lập trình đến CPU. Khi ta tải xuống một đề án, CPU sẽ lưu trữ chương trình người dùng (các OB, FC, FB và DB) trong bộ nhớ thường trực.

Ta có thể tải xuống đề án từ một thiết bị lập trình đến CPU từ bất kỳ trong các khu vực sau:

- “Project tree”: nhấp chuột phải lên phần tử chương trình, sau đó nhấp vào mục chọn “Download” phụ thuộc vào ngữ cảnh.
- Trình đơn “Online”: nhấp vào mục chọn “Download to device”.
- Thanh công cụ: nhấp vào biểu tượng “Download to device”.

Tải lên các phần tử của chương trình.

Ta có thể tải lên tất cả các khối chương trình và bảng thẻ ghi từ một CPU trực tuyến (online) đến một đề án ngoại tuyến (offline), nhưng không thể tải lên cấu hình thiết bị hay các bảng quan sát. Ta không thể tải lên vào trong một đề án trống, mà phải có một CPU ngoại tuyến có thể tải lên. Ta không thể tải lên một khối đơn lẻ, mà chỉ có thể tải lên toàn bộ chương trình. Nếu một việc tải lên được thực hiện, CPU ngoại tuyến sẽ “bị xóa” (tất cả các khối và bảng thẻ ghi sẽ bị xóa) trước việc tải lên sau một câu hỏi kiểm tra. Ta không thể chỉnh sửa một khối trong vùng trực tuyến, mà trước tiên phải tải lên nó đến một vùng ngoại tuyến, sau đó chỉnh sửa tại đó và cuối cùng tải lại trở về PLC.

Người dùng tạo ra cấu hình cho thiết bị bằng cách chèn một CPU vào đề án. Việc lựa chọn CPU từ hộp thoại “Add new device” sẽ tạo ra thanh đỡ (rack) và CPU.

Hộp thoại “Add new device”



Hình 1.20 Giao diện phần mềm **Kéo và thả vào cây đề án**

1. Tạo ra một đề án mới.
2. Thêm một thiết

bị CPU phù hợp với CPU mà ta đang tải lên từ đó.

3. Mở rộng nút của CPU một lần để thư mục “Program blocks” nhìn thấy được.
4. Từ cây Project, mở rộng nút “Online access”, mở rộng nút cho mạng muốn có, và nhấp đôi chuột lên “Update accessible devices”.
5. Sau khi các CPU có sẵn được liệt kê, mở rộng nút của CPU mà ta quan tâm.
6. Nhấp chuột trái và giữ thư mục “Program blocks” từ vùng “Online access” và kéo nó đến thư mục “Program blocks” từ vùng ngoại tuyến, sau đó thả chuột. Con trỏ của chuột phải chuyển sang “+” khi ta đi qua vùng hiệu chỉnh.
7. Nên xem hộp thoại “Upload preview” mở. Nhấp vào hộp “Continue” và sau đó nhấp vào “Upload from device”.
8. Cho phép việc tải lên hoàn thành. Lúc này nên xem tất cả các khối chương trình, các khối công nghệ và các thẻ ghi trong vùng ngoại tuyến.
9. Do cấu hình thiết bị không thể được tải lên, sử dụng “Device configuration” để thiết lập các thuộc tính của CPU một cách thủ công, bao gồm cả địa chỉ IP mong muốn, và để thêm vào các thiết bị khác đến đề án ngoại tuyến.

Bài 2: Tính toán và các phép toán nhị phân của PLC S7-1200

Thời gian: 16 giờ

Mục tiêu của bài:

- Trình bày được các chức năng của RS, Timer, counter (bộ định thời, bộ đếm).
- Ứng dụng linh hoạt các chức năng của RS, Timer, counter trong các bài toán thực tế: Lập trình, kết nối, chạy thử...
- Rèn luyện tính tỉ mỉ, sáng tạo trong công việc

1. Lập trình về các liên kết logic

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

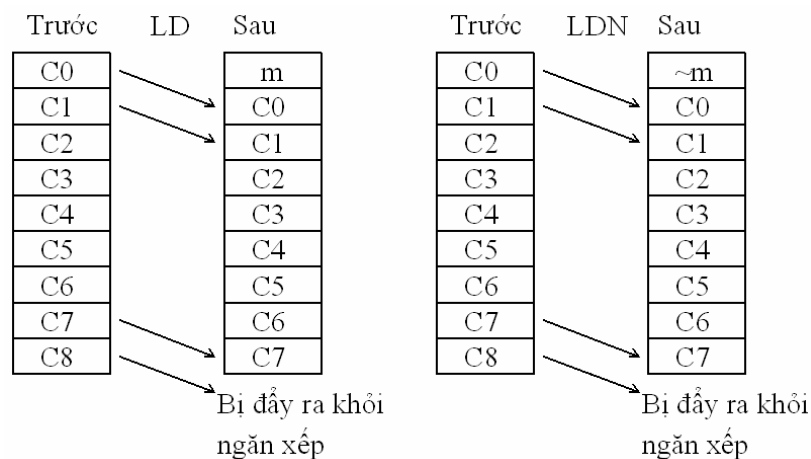
Các lệnh thay đổi ngăn xếp

S0	Stack 0 - Bit thứ 1 của ngăn xếp
S1	Stack 1 - Bit thứ 2 của ngăn xếp
S2	Stack 2 - Bit thứ 3 của ngăn xếp
S3	Stack 3 - Bit thứ 4 của ngăn xếp
S4	Stack 4 - Bit thứ 5 của ngăn xếp
S5	Stack 5 - Bit thứ 6 của ngăn xếp
S6	Stack 6 - Bit thứ 7 của ngăn xếp
S7	Stack 7 - Bit thứ 8 của ngăn xếp
S8	Stack 8 - Bit thứ 9 của ngăn xếp

Hình 2.1 Ngăn xếp trong S7-1200.

Load (LD): lệnh LD nạp giá trị của một tiếp điểm vào trong bit đầu tiên của ngăn xếp, các giá trị cũ còn lại trong ngăn xếp bị đẩy xuống 1 bit.

Load Not (LDN): lệnh LDN nạp giá trị nghịch đảo của 1 tiếp điểm vào trong bit đầu tiên của ngăn xếp, các giá trị cũ còn lại bị đẩy xuống 1 bit.



Hình 2.2 Trạng thái ngăn xếp trước và sau lệnh LD; LDN.

Các lệnh trong đại số Boolean

Trong LAD các lệnh này được biểu diễn qua cấu trúc mạch, mắc nối tiếp hay song song các tiếp điểm thường đóng và các tiếp điểm thường mở. STL có thể sử dụng các lệnh A (And) hay o (Or) cho các hàm hở hoặc lệnh AN (And Not), ON (Or Not) cho các hàm kín. Giá trị ngăn xếp thay đổi phụ thuộc vào từng lệnh (*Bảng 1*).

Ngoài những lệnh làm việc trực tiếp với tiếp điểm, s7-1200 còn 5 lệnh đặc biệt biểu diễn các phép tính của đại số Boolean cho các bit trong ngăn xếp, được gọi là các lệnh *Stack Logic*. Đó là các lệnh ALD (And load), OLD (Or Load), LPS (Logic push), LRD (Logic read) và LPP (logic Pop). Lệnh *Stack Logic* được dùng để tổ hợp, sao chụp hoặc xóa các mệnh đề logic. LAD không có bộ đếm dành cho lệnh *Stack Logic*. STL sử dụng các lệnh *Stack Logic* để thực hiện phương trình tổng thể có nhiều biểu thức con (*Bảng 2.3*).

Bảng 1: Các lệnh đại số trong STL.

Lệnh	Mô tả lệnh	Toán hạng
------	------------	-----------

O _n A _n	Lệnh thực hiện toán tử A và O giữa giá trị logic của tiếp điểm n và giá trị bit đầu tiên trong ngăn xếp. Kết quả được ghi lại vào bit đầu tiên của ngăn xếp	n: I, Q, M, SM, T, C, V (bit)
ON _n AN _n	Lệnh thực hiện toán tử A và O giữa giá trị logic nghịch đảo của tiếp điểm n và giá trị bit đầu tiên trong ngăn xếp. Kết quả được ghi lại vào bit đầu tiên của ngăn xếp	
OI _n AI _n	Lệnh thực hiện tức thời toán tử A và O giữa giá trị logic của tiếp điểm n và giá trị bit đầu tiên trong ngăn xếp. Kết quả được ghi lại vào bit đầu tiên của ngăn xếp	n: I (bit)
ONI _n ABI _n	Lệnh thực hiện tức thời toán tử A và O giữa giá trị logic nghịch đảo của tiếp điểm n và giá trị bit đầu tiên trong ngăn xếp. Kết quả được ghi lại vào bit đầu tiên của ngăn xếp	

Bảng 2: Một số lệnh thường gặp.

Lệnh	Mô tả lệnh	Toán hạng
ALD	Thực hiện phép A giữa bit thứ 1 và bit thứ 2 của ngăn xếp. Kết quả được ghi vào bit thứ 1. Giá trị còn lại của ngăn xếp được kéo lên 1 bit.	không có
OLD	Thực hiện phép O giữa bit thứ 1 và bit thứ 2 của ngăn xếp. Kết quả được ghi vào bit thứ 1. Giá trị còn lại của ngăn xếp được kéo lên 1 bit.	không có
LPS	Lệnh logic Push, sao chép bit đầu tiên của ngăn xếp	không có
LRD	Sao chép giá trị thứ hai của ngăn xếp lên giá trị thứ 1.	không có
LPP	Lệnh kéo ngăn xếp lên 1 bit. Giá trị bit sau được chuyển cho bit trước.	không có

Trên hình 3 là kết quả khi thực hiện hai lệnh ALD và OLD.

Trước	ALD	Sau	Trước	OLD	Sau
C0		$m=C0 \wedge C1$	C0		$m=C0 \vee C1$
C1		C2	C1		C2
C2		C3	C2		C3
C3		C4	C3		C4
C4		C5	C4		C5
C5		C6	C5		C6
C6		C7	C6		C7
C7		C8	C7		C8
C8			C8		

Hình 2.3 Thực hiện lệnh ALD và OLD.

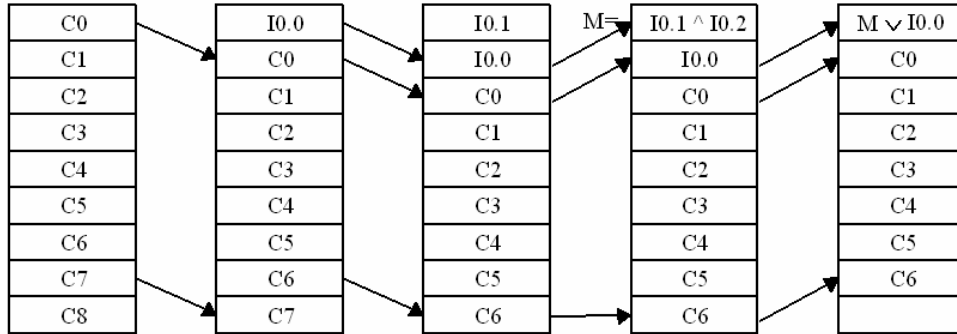
Ví dụ 1: Phân tích sự thay đổi nội dung ngăn xếp cho đoạn mã lệnh sau:

LD I0.0

LD I0.1

A I0.2

OLD = Q0.0



Hình 2.4 Nội dung ngăn xếp khi thực hiện đoạn mã lệnh.

1.2. Các lệnh liên kết logic và các cổng logic cơ bản:

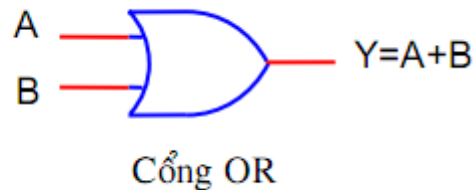
a, *Phép toán OR và cổng OR*

Gọi A và B là 2 biến logic độc lập. Khi A và B kết hợp qua phép toán OR, kết quả x được mô tả như sau:

$$X = A + B$$

Trong biểu thức này, dấu “+” không có nghĩa là phép cộng thuần túy. Nó là phép toán OR, kết quả của phép toán OR được cho trong bảng sự thật sau:

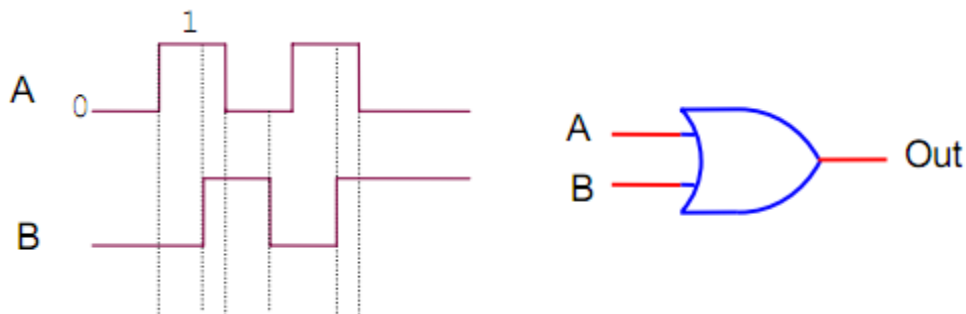
A	B	$X=A+B$
0	0	0
0	1	1
1	0	1
1	1	1



Hình 2.5 Bảng sự thật của phép toán OR.

Ví dụ 2: Xác định dạng sóng ngo ra cổng OR khi ngo vào A, B thay đổi theo giản đồ sau:

Hình 2.6 Giản đồ xung của phép toán .



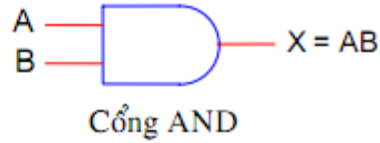
b, *Phép toán AND và cổng AND*

Nếu hai biến logic A và B được kết hợp qua phép AND, kết quả là:

$$X = A.B$$

Bảng sự thật của phép nhân 2 biến A và B như sau:

A	B	X=A.B
0	0	0
0	1	0
1	0	0
1	1	1



Hình 2.7 Bảng sự thật của phép toán AND.

Ví dụ 3: Xác định dạng sóng ngõ ra của cổng AND ứng với các ngõ vào như sau:



Hình 2.8 Giải đồ xung của phép toán AND .

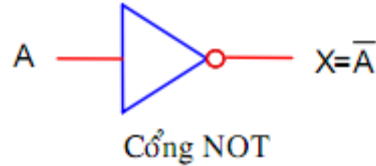
Trong ví dụ này thấy rằng, ngõ ra sẽ bằng với ngõ vào A khi B ở mức logic 1. Vì vậy ta có thể xem ngõ vào B như ngõ vào điều khiển, nó cho phép dạng sóng ở ngõ vào A xuất hiện ở ngõ ra hay không.

c, Phép toán NOT và cổng NOT

Nếu biến A được đưa qua phép toán NOT, kết quả x sẽ là:

$$X = \bar{A}$$

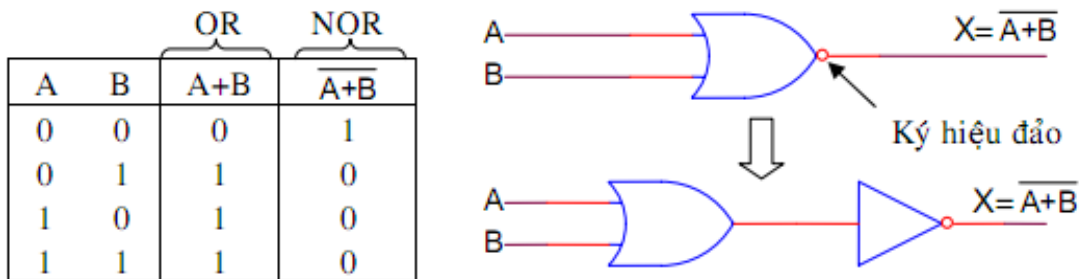
Cổng NOT chỉ có một ngõ vào và một ngõ ra. Trên hình 9 là bảng sự thật và kí hiệu của phần tử NOT



Hình 2.9 Bảng sự thật của phép toán NOT.

d, Phần tử NOR và cổng NOR

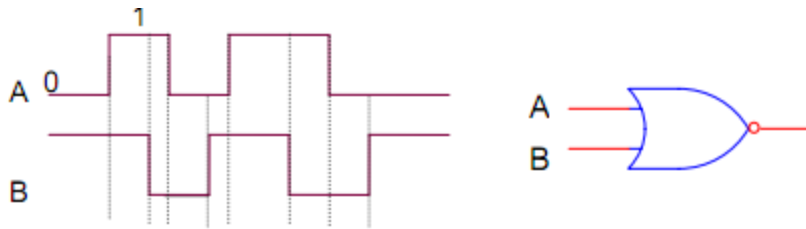
Cổng NOR hoạt động giống như hai cổng OR và NOT mắc nối tiếp như sau:



Hình 2.10 Bảng sự thật của phép toán NOR

Trên sơ đồ mạch điện cổng NOR có kí hiệu giống như cổng OR nhưng có thêm vòng tròn ở phải đầu ra đại diện cho tín hiệu ra đảo so với cổng OR. Phần tử OR có thể có hai hoặc nhiều đầu vào. Nếu các đầu vào của cổng OR được nối chung thì cổng OR có chức năng như phần tử NOT.

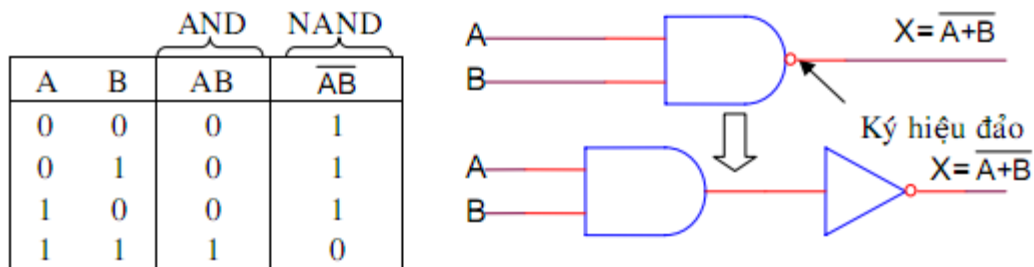
Ví dụ 4: Xác định dạng sóng ngõ ra của cổng NOR ứng với các ngõ vào như sau:



Hình 2.11 Giải đồ xung của phép toán NOR .

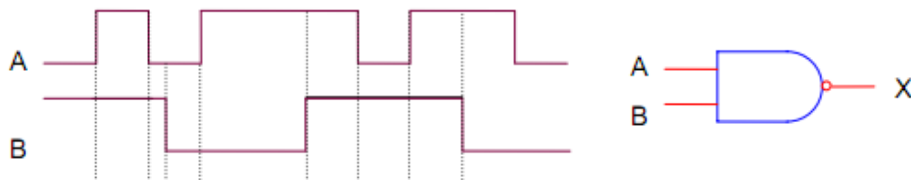
e, Phần tử NAND và cổng NAND

Cổng NAND tương ứng với cổng AND và NOT:



Hình 2.12 Bảng sự thật của phép toán NAND.

Ví dụ 5: Xác định sóng ngõ ra của phần tử NAND khi biến sóng ngõ vào như sau:



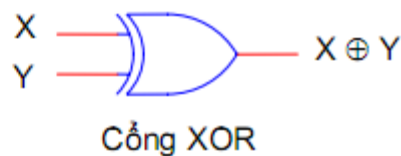
Hình 2.13 Giải đồ xung của phép toán NAND .

Đầu vào của phần tử NAND có 2 xung với xung cao tương ứng với “1” logic, xung ở mức thấp tương ứng với mức “0” logic. Dựa vào bảng chân lý ứng với phần tử NAND chúng ta có thể xác định được dạng sóng đầu ra.

f, Phép toán XOR và cổng XOR

Phép toán XOR có bảng sự thật như sau:

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0



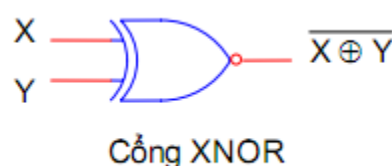
Hình 2.14 Bảng sự thật của phép toán XOR.

Biểu thức toán: $X \oplus Y = \overline{X}Y + X\overline{Y}$

g, Phép toán tương đương và cổng XNOR

Bảng sự thật:

X	Y	$X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1



Hình 2.15 Bảng sự thật của phép toán XNOR.

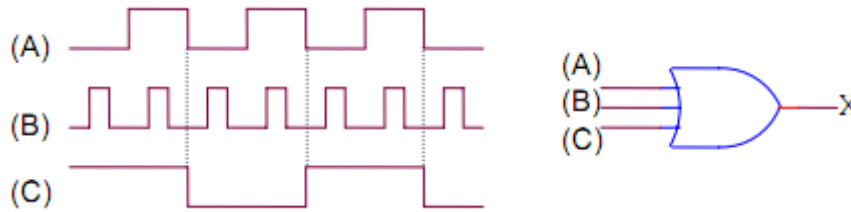
Từ bảng sự thật thấy rằng: $X \equiv Y = 0$ khi $X \neq Y$, và $X \equiv Y = 1$ khi $X = Y$

Biểu thức toán: $X \equiv Y = \overline{X \oplus Y} = XY + \overline{X} \cdot \overline{Y}$

1.3. Bài tập ứng dụng

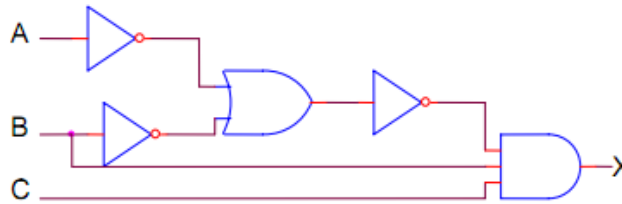
1, Vẽ lại sóng ngõ ra cho mạch hình sau:

a, Giả sử ngõ vào A=0, vẽ dạng sóng ngõ ra.



b, Giả sử ngõ ra A=1, vẽ dạng sóng ngõ ra.

c, Thay cổng OR thành cổng AND rồi vẽ sóng ngõ ra



2, Có bao nhiêu tổ hợp ngõ vào của cổng OR 5 ngõ vào làm cho ngõ ra ở mức cao?

3, Viết biểu thức Boolean cho ngõ ra X. Xác định giá trị của X Ứng với các điều kiện ngõ vào có thể.

2. Các lệnh ghi/xoá giá trị cho tiếp điểm

Mục tiêu: Trình bày lệnh ghi/xoá giá trị cho tiếp điểm.

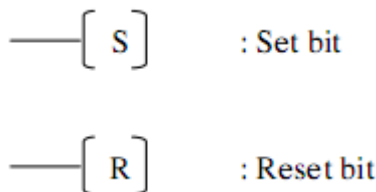
2.1. Mạch nhớ R_S:

Mạch nhớ là mạch có hai trạng thái ổn định và thông qua tín hiệu ngõ vào mà trạng thái của nó thay đổi. Đối với mạch điều khiển dùng relay và contactor ta có mạch tự duy trì. Còn trong PLC có khâu R-S (viết tắt của Reset và Set). Mạch nhớ R-S là rất cần thiết trong kỹ thuật điều khiển. Nó được xem là một chức năng cơ bản trong hầu hết các loại PLC và được chia thành hai loại là: Ưu tiên SET và ưu tiên RESET.

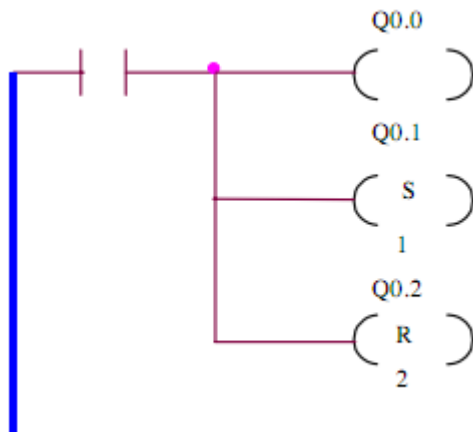
2.2. Lệnh Set (S) và Reset (R) trong PLC S7-200

Mạch nhớ R-S được thể hiện qua hai lệnh set và reset với các ví dụ ứng dụng dùng bộ nhớ với cú pháp như sau:

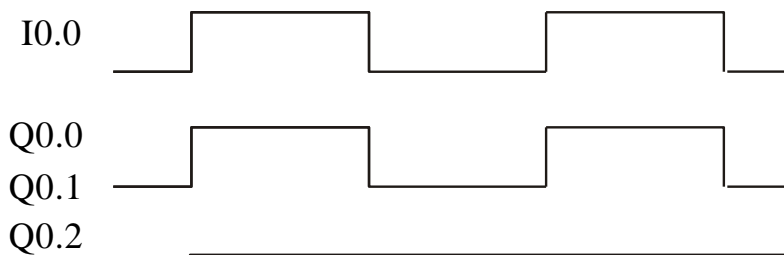
Hình 2.16: Lệnh SET và RESET trong S7-200.



Ví dụ 6: Sử dụng lệnh set và reset:



a)



b)

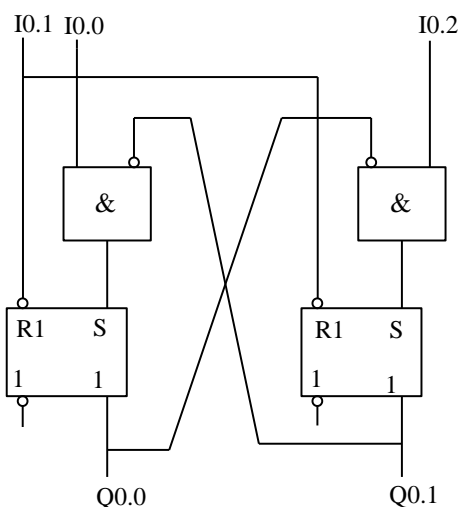
Hình 2.17: a) Đoạn lệnh sử dụng ngôn ngữ LAD.
b) Giải đồ xung lối ra.

Trong chương trình I0.0 là đầu vào. Q0.0, Q0.1, Q0.2 là đầu ra.

2.3. Các ví dụ ứng dụng dùng bộ nhớ

a, Mạch chốt lẫn nhau của 2 van từ

Sơ đồ logic và bảng xác lập vào ra.

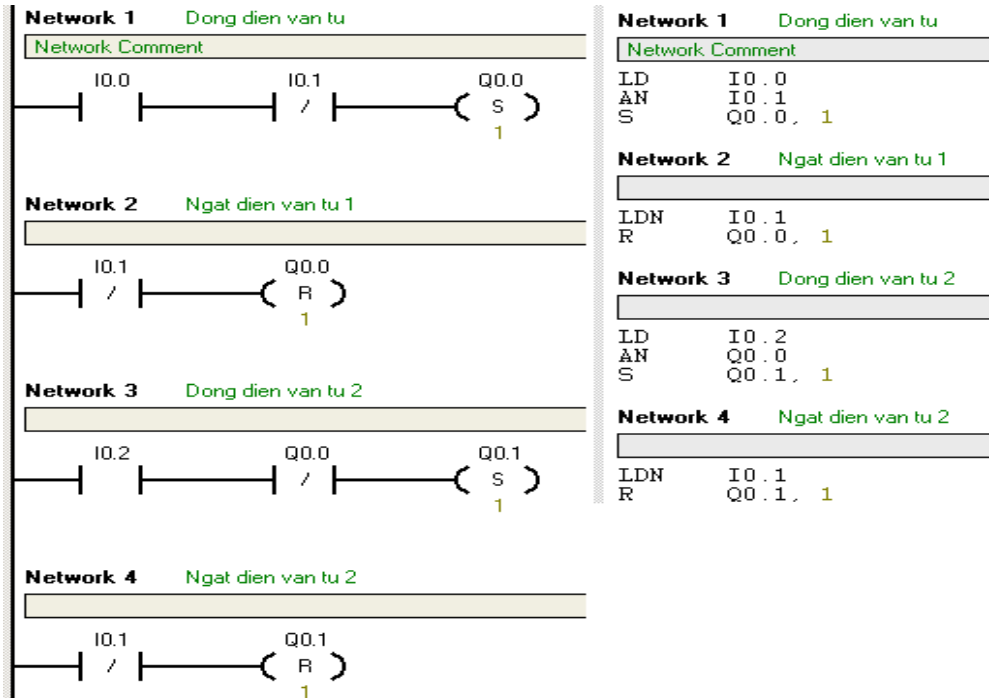


Xác lập vào/ra		
Kí hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút ấn thường mở
S2	I0.1	Nút ấn thường đóng
S3	I0.2	Nút ấn thường mở
Y1	Q0.0	Van từ 1
Y2	Q0.1	Van từ 2

Mô tả hoạt động:

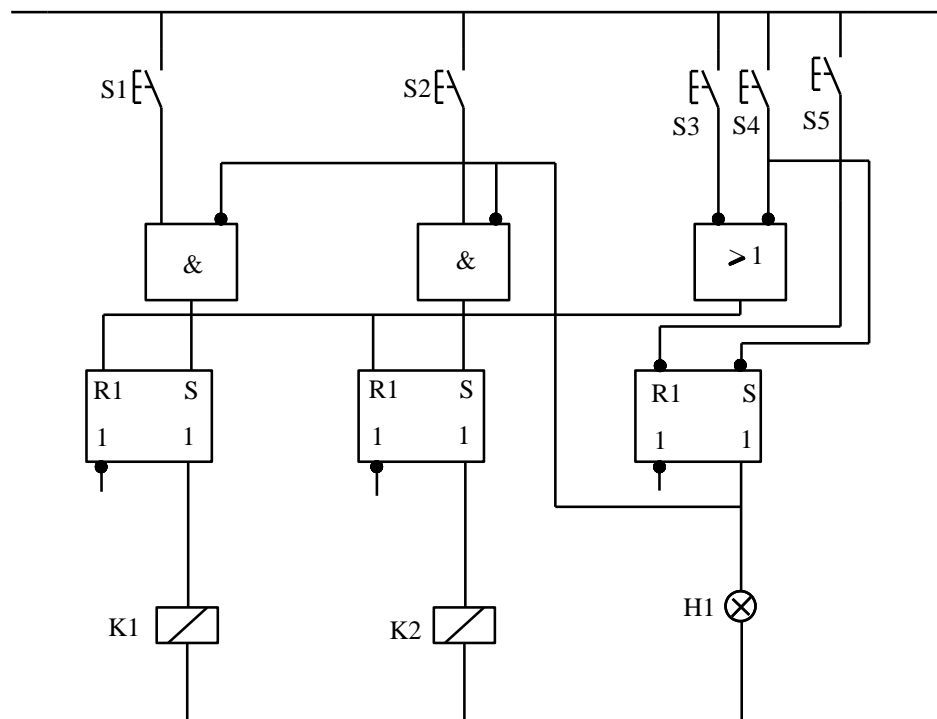
Qua việc khởi động S1 hoặc S3 các bộ phận nhớ một (van từ 1) hoặc bộ nhớ hai (van từ 2) sẽ được đặt. Nút ấn S2 làm nhiệm vụ cắt mạch.

Chương trình được viết ở LAD như sau:



b, Mạch tuần tự cưỡng bức có báo lỗi

Sơ đồ logic:



Mô tả hoạt động:

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

Nhiệm vụ

- Lập bảng xác định vào/ra
- Vẽ sơ đồ kết nối với PLC

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

c, Bộ chọn theo bước

Mô tả hoạt động

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

Nhiệm vụ

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

3. Timer

Mục tiêu: Trình bày chức năng, cấu trúc, nguyên lý hoạt động của bộ định thời trong PLC.

Timer là bộ tạo thời gian trễ giữa tín hiệu vào và tín hiệu ra. Nếu ký hiệu tín hiệu (logic) và là $x(t)$ và thời gian trễ được tạo ra bằng timer là τ thì tín hiệu đầu ra của timer đó là $x(t-\tau)$. S7-1200 có timer được chia làm 2 loại khác nhau, đó là:

- Timer tạo thời gian trễ không có nhớ (On-delay timer), ký hiệu TON
- Timer tạo thời gian trễ có nhớ (Retentive On-delay timer), ký hiệu là TONR.

Hai kiểu timer của S7-1200 (TON và TONR) cùng bắt đầu tạo thời gian trễ tín hiệu kể từ thời điểm có sườn lên ở tín hiệu đầu vào, tức là khi tín hiệu đầu vào chuyển trạng thái logic từ 0 lên 1, được gọi là *thời điểm timer được kích*, và không tính khoảng thời gian khi đầu vào có giá trị logic 0 vào thời gian trễ tín hiệu được đặt trước.

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

Timer TON và TONR bao gồm 3 loại với độ phân giải khác nhau, độ phân giải 1ms, 10ms và 100ms. Thời gian trễ τ được tạo ra chính là tích của độ phân giải của bộ timer và giá trị đặt trước cho timer.

Ví dụ như một bộ timer có độ phân giải bằng 10ms và giá trị đặt là 50 thì thời gian trễ $\tau=10 \times 50=500\text{ms}$.

Timer của S7-1200 có những tính chất cơ bản sau:

- Các bộ timer được điều khiển bởi một cổng vào và giá trị đếm tức thời. Giá trị đếm tức thời của timer được nhớ trong thanh ghi 2 byte (T-word) của timer, xác định khoảng thời gian trễ kể từ khi timer được kích. Giá trị đặt trước của các bộ timer được ký hiệu trong LAD và STL là PT. Giá trị đếm tức thời của thanh ghi T-word thường xuyên được so sánh với giá trị đặt trước của timer.
- Mỗi timer, ngoài thanh ghi 2 byte T-word lưu giá trị đếm tức thời còn có một bit, ký hiệu bằng T-bit, chỉ trạng thái logic đầu ra. Giá trị logic này phụ thuộc vào kết quả so sánh giữa giá trị đếm tức thời với giá trị đặt trước.

- Trong khoảng thời gian tín hiệu x(t) có giá trị logic 1, giá trị đếm tức thời trong T-word luôn được cập nhật và thay đổi tăng dần cho đến khi đạt giá trị cực đại. Khi giá trị đếm tức thời lớn hơn hay bằng giá trị đặt trước, T-bit có giá trị logic 1.

Bảng 4: Các loại Timer của CPU 212 và CPU 214.

Lệnh	Độ phân giải	Giá trị cực đại	CPU 12..	
TON	1ms	32,767s	T32	T32, T96
	10ms	327,67s	T33-T36	T33-T36, T97-T100
	100ms	3276,7s	T37-T63	T37-T63, T101-T127
TONR	1ms	32,767s	T0	T0, T64
	10ms	327,67s	T1-T4	T1-T4, 65-T68
	100ms	3276,7s	T5-T31	T5-T31, T69-T95

Cú pháp khai báo sử dụng timer trong LAD, bao gồm 2 loại như sau:

3.1. On - delay Timer (TON)

- Cú pháp:

Hình 2.18: Khai báo sử dụng TON.



Khai báo timer số hiệu Txx kiểu TON để tạo thời gian trễ tính từ khi đầu vào IN được kích. Nếu như giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước PT thì T-bit có giá trị logic bằng 1. Có thể reset timer kiểu TON bằng lệnh R hoặc bằng giá trị logic 0 tại đầu vào IN.

3.2. Retentive On-Delay Timer(TONR)

- Cú pháp:



Hình 2.19: Khai báo sử dụng TONR.

Cú pháp khai báo sử dụng timer trong STL như sau: Khai báo timer của S7-200 là lệnh có điều kiện. Tại thời điểm khai báo tín hiệu đầu vào có giá trị logic bằng giá trị logic của bit đầu tiên trong ngăn xếp.

Bảng 5: Các bước khai báo sử dụng Timer.

STT	Mô tả
TON Txx n	Khai báo timer số hiệu xx kiểu TON để tạo thời gian trễ tính từ khi bit đầu trong ngăn xếp có giá trị logic 1. Nếu như giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước n thì T-bit có giá trị 1. Có thể reset timer

	kiểu TON bằng lệnh R hoặc bằng giá trị logic 0 tại đầu vào.
TONR Txx n	Khai báo timer số hiệu xx kiểu TONR để tạo thời gian trễ tính từ khi bit đầu tiên trong ngăn xếp có giá trị logic 1. Nếu như giá trị đếm tức thời lớn hơn hoặc bằng giá trị đặt trước n thì T-bit có giá trị logic bằng 1. Chỉ có thể reset timer kiểu TONR bằng lệnh R cho T-bit.

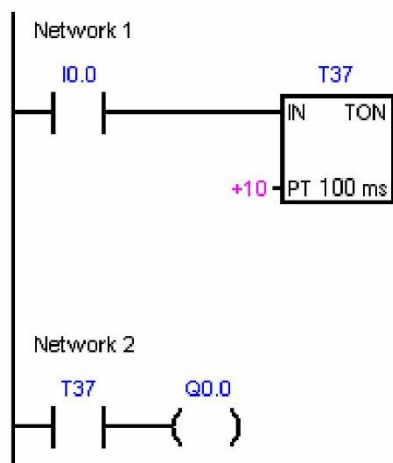
Khi sử dụng timer kiểu TONR, giá trị đếm tức thời được lưu lại và không bị thay đổi trong khoảng thời gian tín hiệu đầu vào logic 0. Giá trị của T-bit không được nhớ mà hoàn toàn phụ thuộc vào kết quả so sánh giữa giá trị đếm tức thời và giá trị đặt trước.

Các timer được đặt tên là Txx, trong đó xx là số hiệu của timer. Txx đồng thời cũng là đại chỉ hình thức của T-word và T-bit của timer đó. Tuy chúng có cùng địa chỉ hình thức, song T-word và T-bit vẫn được phân biệt với nhau nhờ kiểu sử dụng lệnh với Txx. Khi dùng lệnh làm việc với từ, Txx được hiểu là T-word, ngược lại khi sử dụng lệnh làm việc với tiếp điểm, Txx được hiểu là T-bit.

Khi timer được reset, T-word và T-bit của nó đồng thời được xóa và có giá trị 0. Đối với timer TON có hai phương pháp reset là xóa tín hiệu đầu vào hoặc dùng lệnh R. Còn đối với timer TONR chỉ có một phương pháp reset duy nhất là dùng lệnh R (R Txx K1).

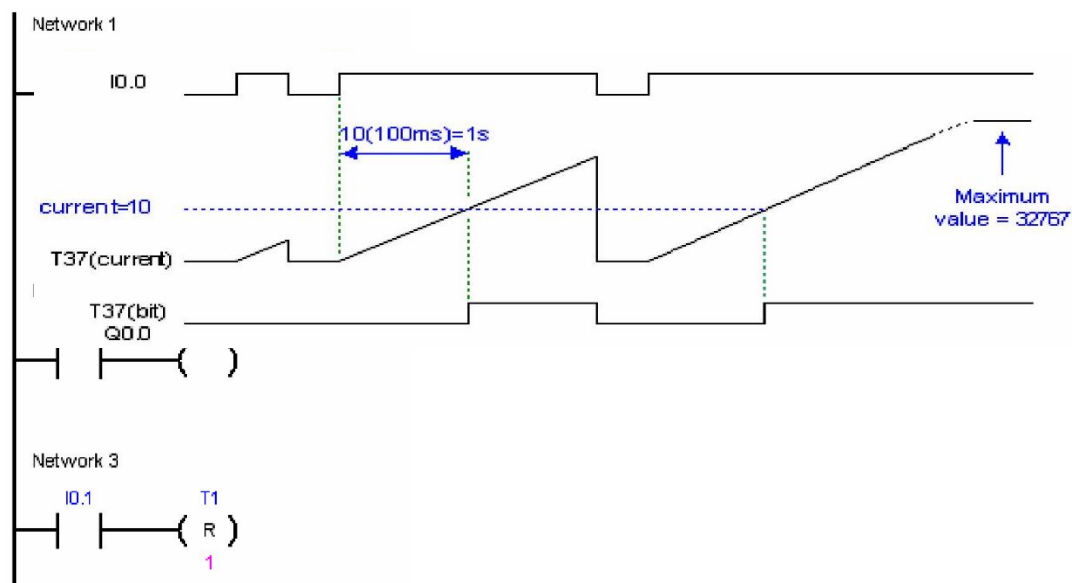
Ví dụ 7: Cách khai báo và sử dụng sử dụng timer kiểu TON.

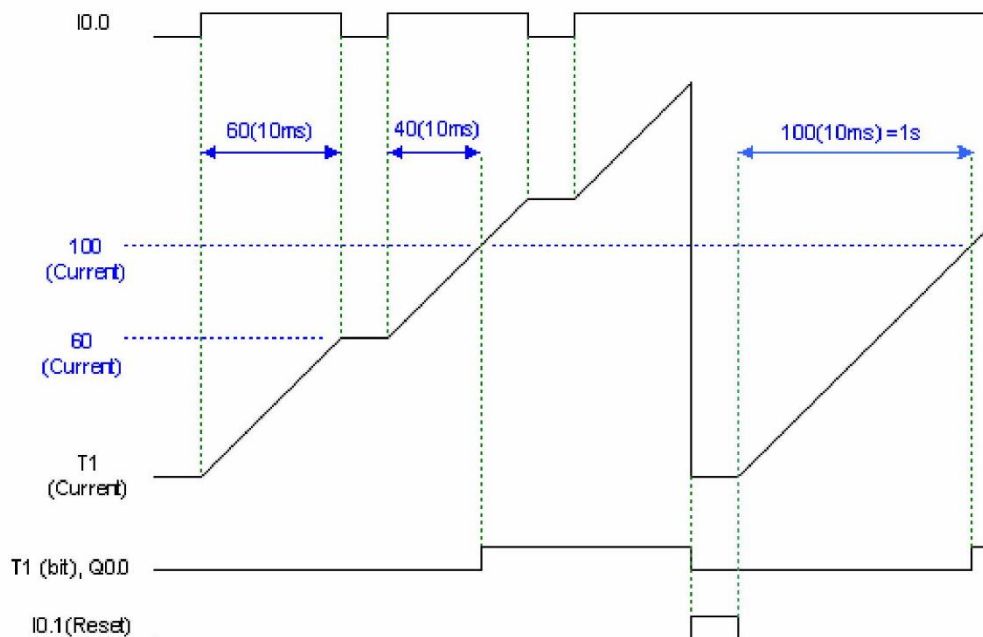
Khi ngõ vào I0.0 =1 Timer T37 được kích , Nếu sau 10x100ms =1s I0.0 vẫn giữ



trạng thái thì bit T37 sẽ lên 1 (Khi đó Q0.0 lên 1). Nếu I0.0 =1 không đủ thời gian 1S thì bit T37 sẽ không lên 1.

Ví dụ 8: Cách khai báo và sử dụng sử dụng timer kiểu TONR





Ngõ vào IO.0 có tác dụng kích thời gian cho Timer, khi ngõ IO.0 =1 thời gian Timer được tính, khi IO.0=0 thời gian không bị Reset về 0. Khi đủ thời gian thì Bit T1 sẽ lên 1. Thời gian Timer chỉ bị Reset khi có tín hiệu Reset Timer (tín hiệu từ ngõ IO.1)

- **Cập nhật timer có độ phân giải 1ms**

CPU của S7-1200 có các bộ timer có độ phân giải 1ms cho phép PLC cập nhật và thay thay đổi giá trị đếm tức thời trong T-word mỗi 1ms một lần. Các bộ timer với độ phân giải thấp này có khả năng điều khiển chính xác các thao tác.

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

Do việc cập nhật T-word của timer với độ phân giải 1ms hoàn toàn tự động nên thời gian trễ có thể bị trôi trong khoảng thời gian 1ms. Bởi vậy, ví dụ để có thể có được thời gian trễ không dưới 56ms ta nên đặt giá trị ban đầu cho PT là 57.

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

- **Cập nhật timer có độ phân giải 10ms**

CPU của S7-1200 có các bộ timer với độ phân giải 10ms. Sau khi được kích, việc cập nhật T-word và T-bit để thay đổi giá trị đếm tức thời và trạng thái logic đầu ra của các bộ timer này không phụ thuộc vào chương trình và được tiến hành *hoàn toàn tự động* mỗi vòng quét một lần vào thời điểm đầu vòng quét.

Thực hiện lệnh R đối với timer có độ phân giải 10ms đang ở trạng thái làm việc là đưa timer về trạng thái ban đầu và đưa T-word và T-bit giá trị 0.

Do việc cập nhật T-word của timer chỉ được thực hiện tự động mỗi vòng quét một lần nên thời điểm thay đổi giá trị đếm tức thời và giá trị logic của T-bit của timer có thể bị

trôi trong khoảng 10ms. Bởi vậy, ví dụ để tạo được một khoảng thời gian trễ ít nhất là 140ms, nên chọn giá trị đặt trước cho timer có độ phân giải 10ms là PT=15.

- **Cập nhật timer có độ phân giải 100ms**

Hầu hết các bộ timer của S7-1200 đều là timer có độ phân giải 100ms. Giá trị để lưu trữ trong bộ timer 100ms được tính tại đầu mỗi vòng quét và thời gian để tính sẽ là khoảng thời gian từ đầu vòng quét trước đó.

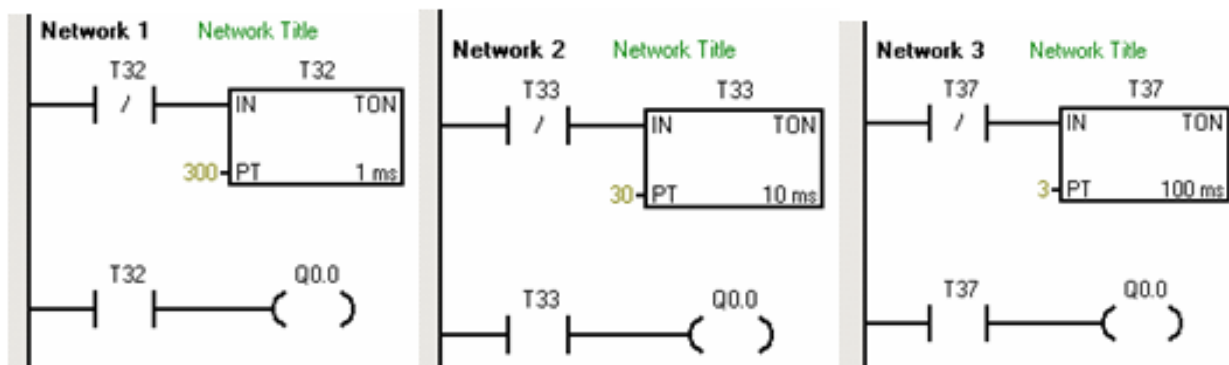
Việc cập nhật để thay đổi giá trị đếm tức thời của timer chỉ được tiến hành ngay tại thời điểm có lệnh khai báo cho timer trong chương trình. Bởi vậy quá trình cập nhật giá trị đếm tức thời không phải là một quá trình tự động và không nhất thiết phải được thực hiện một lần trong mỗi thời gian vòng quét ngay cả khi timer đã được kích. Đối với trường hợp một lệnh timer 100ms được khai báo nhiều lần trong một vòng quét thì có thể xảy ra trường hợp giá trị lưu trữ bị cộng nhiều lần với giá trị đếm tức thời, vì vậy nên sử dụng lệnh khai báo timer 100ms chính xác một lần trong một vòng quét.

- **Hiệu quả của việc cập nhật giá trị đếm tức thời của timer 1ms, 10ms, 100ms.**

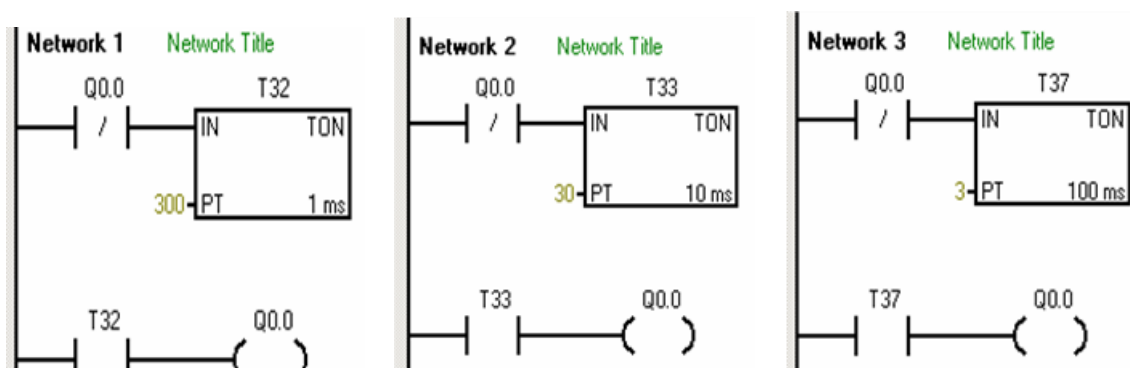
Việc cập nhật giá trị đếm tức thời của các timer với độ phân giải khác nhau được thực hiện tại các thời điểm khác nhau phụ thuộc vào các sử dụng timer đó. Ví dụ sau mô phỏng sự khác nhau đó với 3 loại timer cùng đặt thời gian 300ms.

- Trong trường hợp sử dụng timer 1ms, Q0.0 sẽ có giá trị logic bằng 1 trong khoảng thời gian 1 vòng quét nếu thời điểm cập nhật giá trị tức thời xảy ra trước khi tiếp điểm thường mở T32 và tiếp điểm thường đóng T32 chuyển trạng thái.
- Trong trường hợp sử dụng timer có độ phân giải 10ms, Q0.0 sẽ luôn có giá trị logic 0 vì khi bit T33 có giá trị logic 1 ở đầu vòng quét thì ngay sau đó sẽ bị chuyển về trạng thái 0.
- Trong trường hợp sử dụng timer 100ms, Q0.0 sẽ luôn có giá trị logic 1 trong khoảng thời gian 1 vòng quét mỗi khi giá trị đếm tức thời bằng giá trị đặt trước.

Hình 2.20: Ảnh hưởng của độ phân giải đến đầu ra của timer.



Việc sử dụng tiếp điểm thường đóng Q0.0 làm tín hiệu đầu vào cho timer đảm bảo Q0.0 sẽ có giá trị logic bằng 1 trong một vòng quét ở mỗi thời điểm mà giá trị đếm của bộ timer đạt được giá trị đặt trước PT:



Hình 2.21: Khắc phục ảnh hưởng của độ phân giải đến đầu ra của timer.

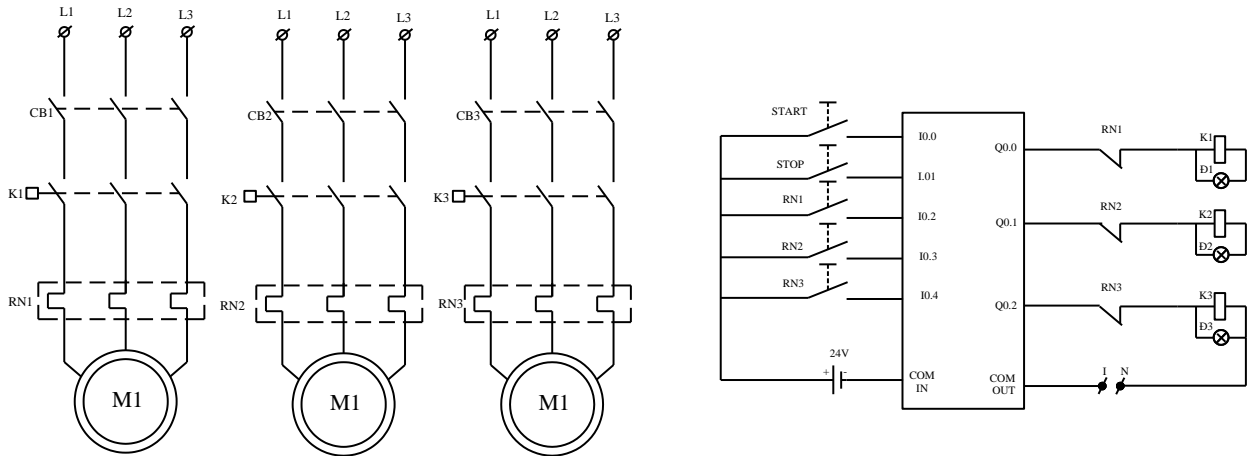
3.3. Bài tập ứng dụng timer

Khởi động hệ thống băng tải gồm 3 động cơ:

Khi khởi động START thì động cơ 1 chạy, sau 3s thì tự động động cơ 2 chạy, tiếp theo 3s kể từ động cơ chạy. Tương ứng mỗi động cơ chạy thì có đèn sáng.

Khi ấn STOP thì động cơ thứ 1 dừng trước, sau 5s thì tự động động cơ thứ 2 dừng và sau 5s thì tự động động cơ thứ 3 dừng hẳn.

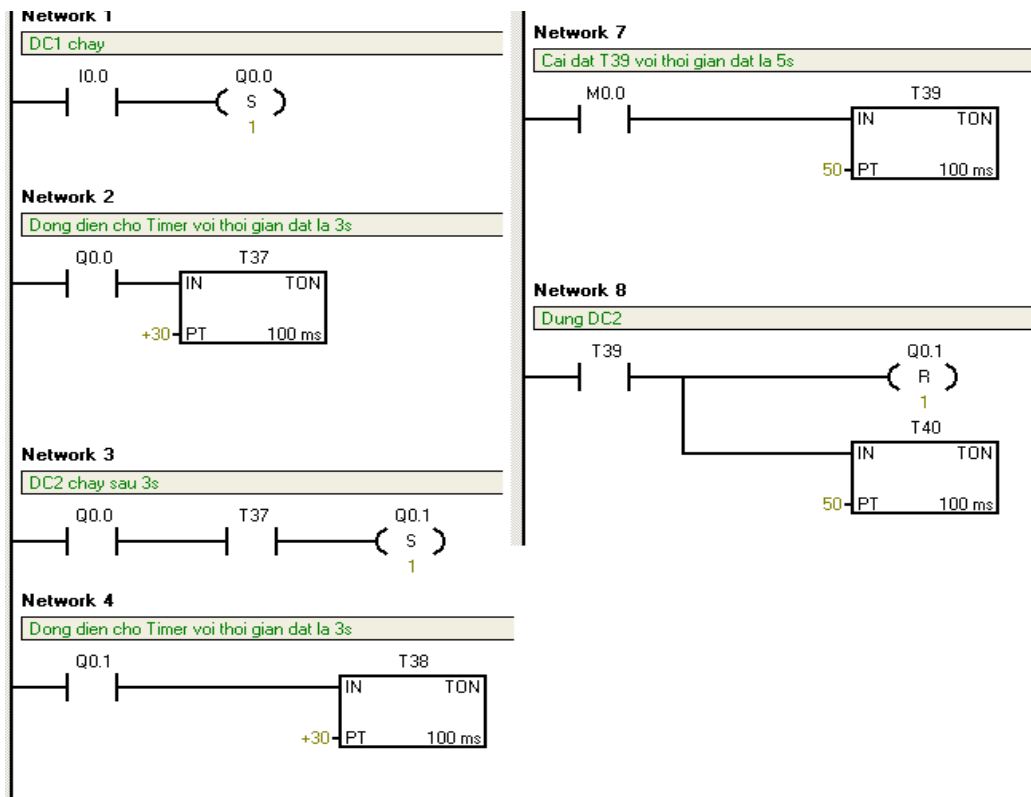
Mạch động lực:



Bảng đầu vào/ra

Kí hiệu	Địa chỉ	Giải thích
START	I0.0	Nút ấn khởi động
STOP	I0.1	Nút ấn dừng
K1	Q0.0	Điều khiển động cơ 1
K2	Q0.1	Điều khiển động cơ 2
K3	Q0.2	Điều khiển động cơ 3

Chương trình viết bằng ngôn ngữ LAD



Chương trình viết bằng ngôn ngữ STL

Network 1

DC1 chạy

```
LD I0.0
S Q0.0, 1
```

Network 2

Động điện cho Timer với thời gian đặt là 3s

```
LD Q0.0
TON T37, +30
```

Network 3

DC2 chạy sau 3s

```
LD Q0.0
A T37
S Q0.1, 1
```

Network 4

Động điện cho Timer với thời gian đặt là 3s

```
LD Q0.1
TON T38, +30
```

Network 5

DC 3 chạy

```
LD Q0.0
A T38
S Q0.2, 1
```

Network 6

Dừng DC1

```
LD I0.1
R Q0.0, 1
S M0.0, 1
```

Network 7

Cài đặt T39 với thời gian đặt là 5s

```
LD M0.0
TON T39, 50
```

Network 8

Dừng DC2

```
LD T39
R Q0.1, 1
TON T40, 50
```

Network 9

Dừng DC3

```
LD T40
R Q0.2, 1
R M0.0, 1
```

Nguyên lý hoạt động của hệ thống:

Khi nhấn nút I0.0 (START) ở Network 1 Q0.0 có điện → công tắc tơ K1 có điện, đóng tiếp điểm trên mạch động lực → động cơ 1 (M1) kéo băng tải 1 chạy. Khi Q0.0 có điện sẽ đóng tiếp điểm thường mở Q0.0 ở network 2, sau khoảng thời gian là 3s, T1 có điện → Q0.1 có điện → công tắc tơ K2 có điện, đóng tiếp điểm K2 trên mạch động lực → động cơ 2 chạy và tiếp điểm thường mở của T1 ở network 3 đóng lại, sau khoảng thời gian 3s, T2 có điện → Q0.2 có điện → công tắc tơ K3 có điện, đóng tiếp điểm K3 trên mạch động lực →

động 3 có điện kéo băng tải 3 chạy.

Khi ấn nút I0.1 ở network 1, Q0.0 mất điện, công tắc tơ K1 mất điện, mở tiếp điểm K1 trên mạch động lực → M1 dừng, đồng thời đóng điện cho T39. Sau 5s, ngắt điện động cơ 2 và đóng điện cho T40. Sau 5s dừng động cơ 3.

4. Counter

Mục tiêu: Trình bày chức năng, cấu trúc, nguyên lý hoạt động của bộ đếm trong PLC.

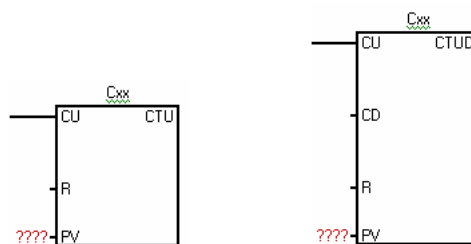
Counter là bộ đếm thực hiện chức năng đếm sườn xung trong S7-1200. các bộ đếm của S7-200 được chia ra làm hai loại: bộ đếm tiến (CTU) và bộ đếm tiến/lùi (CTUD).

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

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

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

- Cú pháp hai bộ đếm CTU và CTUD của s7-1200



Hình 2.22: Khai báo và sử dụng Counter.

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

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

+ Khi đầu vào của chân xóa, ký hiệu bằng R trong LAD hoặc bit thứ nhất của ngăn xếp trong STL có giá trị logic bằng 1.

+ Bằng lệnh R với C-bit.

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

Bộ đếm tiến CTU có miền giá trị đếm tức thời từ 0 đến 32.767. Bộ đếm tiến/lùi CTUD có miền giá trị đếm tức thời là -32.768 đến 32.768.

Về nguyên lý hoạt động, có thể mô tả như sau:

4.1. Counter up (CTU)

Khai báo bộ đếm tiến theo sườn lên CU. Khi giá trị đếm tức thời C-word của Cxx lớn hơn hoặc bằng giá trị đặt trước PV, C-bit có giá trị logic bằng 1. Bộ đếm được reset khi đầu vào R có giá trị logic bằng 1. Bộ đếm ngừng khi C-word đạt giá trị cực đại bằng 32.767.

Các toán hạng

Cxx CPU 212: 0-47

(word) CPU214: 0-47, 80-127.

PV VW, T, C, IW,

(word) QW, MW, SMW, AC, AIW, hằng số, *VD, *AC

4.2. Counter up – down (CTUD)

Khai báo bộ đếm tiến/lùi, đếm tiến theo sườn lên của CU và đếm lùi theo sườn lên của CD. Khi giá trị đếm tức thời C-word lớn hơn hoặc bằng giá trị đặt trước PV, C-bit có giá trị logic bằng 1. Bộ đếm ngừng đếm tiến khi C-word đạt giá trị 32,767 và ngừng đếm lùi khi đạt giá trị cực tiểu -32,767, CTUD reset khi đầu vào R có giá trị logic bằng 1.

Các toán hạng : Cxx CPU 212: 48-63

(word) CPU 214: 48-79

PV VW, T, C, IW,

(word) QW, MW, SMW, AC, AIW, hằng số, *VD, *AC

Các bộ đếm được đánh số từ 0-63 (với CPU 212) hoặc từ 0-127 (với CPU 214) và ký hiệu bằng Cxx, trong đó xx là số thứ tự của bộ đếm. Ký hiệu Cxx đồng thời cũng là địa chỉ hình thức của C-word và C-bit.

Ví dụ 9:

LD C48 //lệnh làm việc với C-bit của C48

LDW C48 K2 //lệnh làm việc với C-word của C48

Ví dụ 10: Về bộ đếm CTUD trong LAD và STL

-Viết bằng STL

NETWORK 1 // I0.0 counts up - I0.1 counts down - I0.2 resets current value to 0

LD I0.0

LD I0.1

LD I0.2

CTUD C48 +4

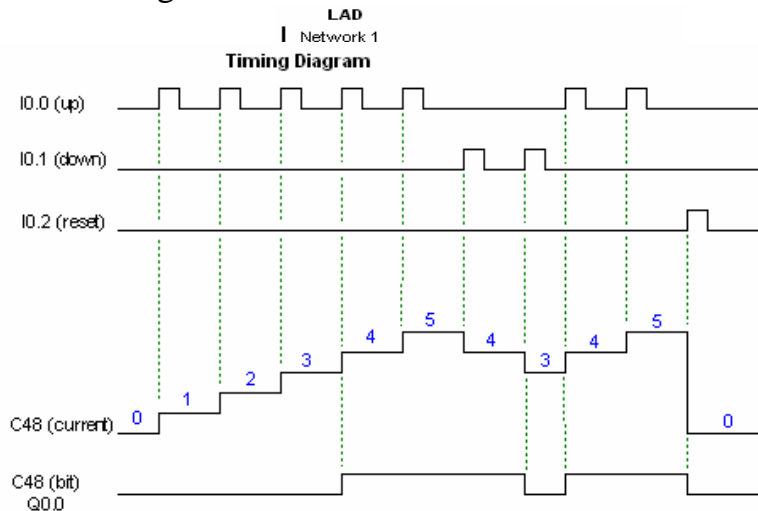
NETWORK 2 // Count Up/Down counter C48 turns on C48 bit when current value >= 4

LD C48

= Q0.0

Viết bằng LAD

- Giải đồ thời gian



* Sử dụng bộ đếm tốc độ cao HSC

Bộ đếm tốc độ cao được theo dõi và điều khiển các quá trình có tốc độ cao mà PLC không thể khống chế được do hạn chế của thời gian vòng quét. Trong CPU 212 có một bộ đếm tốc độ cao (HSC0) và CPU 214 có 3 bộ (HSC0, HSC1, HSC2). Nguyên tắc hoạt động của bộ đếm tốc độ cao tương tự như các bộ đếm khác của PLC, đầu vào đếm theo sườn lên của tín hiệu đầu vào. Số đếm được lưu vào trong một ô nhớ đặc biệt kiểu từ kép và được gọi là giá trị đếm tức thời của bộ đếm, ký hiệu CV (Current Value). Khi giá trị đếm tức thời bằng giá trị đặt trước thì bộ đếm phát ra một ngắt. Giá trị đặt trước là một số nguyên 32 bit cũng được lưu trong một ô nhớ kiểu từ kép và ký hiệu PV (Preset Value).

Nếu chế độ ngắt vào/ra với bộ đếm tốc độ cao được khai báo sử dụng, các tín hiệu sau đây sẽ được phát:

- Ngắt khi $PV=CV$ (đối với HSC0, HSC1, HSC2)
- Ngắt khi có tín hiệu báo thay đổi hướng đếm từ cổng vào (với HSC1, HSC2).
- Ngắt khi có tín hiệu báo xóa (reset) từ cổng vào (với HSC1, HSC2).

Mỗi bộ đếm được chọn chế độ làm việc bằng lệnh HDEF. Từng chế độ làm việc lại có các kiểu hoạt động khác nhau. Kiểu hoạt động của mỗi bộ đếm được xác định bằng nội dung của một byte điều khiển trong vùng nhớ đặt biệt sau đó được khai báo với bộ đếm bằng lệnh HSC.

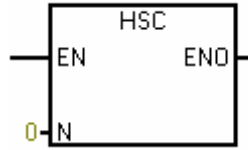
Bộ đếm tốc độ cao HSC0

Bộ đếm HSC0 có một cổng vào là IO.0. Nó chỉ chạy một chế độ làm việc duy nhất là đếm tiền hoặc đếm lùi theo sườn lên của IO.0. Chiều đếm được quy định bởi trạng thái logic

của bit SM37.3 như sau:

- SM37.7 =0 Đếm lùi theo sườn lên của I0.0
 =1 Đếm tiến theo sườn lên của I0.0

Tần số đếm của HSC0 là 2kHz. HSC0 sử dụng từ kép SMD38 để lưu giá trị đếm tức thời CV. Giá trị đặt trước PV được ghi vào từ kép SMD42. Cả giá trị PV và CV đều là số nguyên 32 bit có dấu.



Hình 2.23: Khai báo sử dụng bộ đếm HSC.

HSC0 sử dụng byte SMB37 để xác định kiểu hoạt động như: đếm tiến hay lùi; cho phép hay không cho phép thay đổi giá trị đếm tức thời CV, PV và cho phép/không cho phép bộ đếm. Kiểu hoạt động của HSC0 phải được định nghĩa trong SMB37 trước khi thực hiện lệnh HDEF.

Cấu trúc SMB37, còn được gọi là byte điều khiển HSC0, như sau:

Bảng 5: Chức năng của byte điều khiển HSC0.

Bit	Chức năng
SM37.0	không sử dụng
SM37.1	không sử dụng
SM37.2	không sử dụng
SM37.3	chiều đếm: 0 - đếm tiến, 1 - đếm lùi
SM37.4	Cho phép sửa đổi giá trị đặt trước: 0 – không cho phép, 1 – cho phép
SM37.5	Cho phép sửa đổi giá trị đặt trước: 0 – không cho phép, 1 – cho phép
SM37.6	Cho phép sửa giá trị đếm tức thời: 0 – không cho phép, 1 – cho phép
SM37.7	1 – cho phép kích HSC0, 0 – không cho phép kích HSC0

Các bước khai báo sử dụng HSC0 (nên thực hiện tại vòng quét đầu tiên):

- Nạp giá trị điều khiển phù hợp cho SMB37
- Xác định chế độ làm việc cho bộ đếm bằng lệnh HDEF. Do HSC0 chỉ có một chế độ làm việc nên lệnh xác định kiểu sẽ là: HDEF K0 K0.
- Nạp giá trị đếm tức thời ban đầu và giá trị đặt trước vào SMD38 và SMD42.
- Khai báo sử dụng chế độ ngắt vào ra và kích tín hiệu báo ngắt cho HSC0 bằng lệnh ATCH.
- Kích hoạt bộ đếm bằng lệnh HSC K0.

Sau khi được kích, bộ đếm HSC0 bắt đầu làm việc và sử dụng byte SMB36 để thông báo trạng thái hoạt động của nó như sau:

Khi sử dụng bộ đếm HSC0 cùng với chế độ ngắt vào/ra, tín hiệu báo ngắt HSC0 sẽ xuất hiện khi CV=PV nếu tín hiệu báo ngắt đã được khai báo.

Bảng 6: Chức năng của byte thông báo trạng thái.

Bit	Chức năng
SM36.0	không sử dụng
SM36.1	không sử dụng
SM36.2	không sử dụng
SM36.3	không sử dụng
SM36.4	không sử dụng
SM36.5	chiều đang đếm: 0 - đếm lùi, 1 - đếm tiến
SM36.6	So sánh kết quả tức thời: 0 - $CV \neq PV$, 1 - $CV = PV$
SM36.7	So sánh kết quả tức thời: 0 - $VC < PV$, 1 - $CV > PV$

Thủ tục khai báo sử dụng bộ đếm tốc độ cao

Khai báo sử dụng các bộ đếm nên được thực hiện tại vòng quét đầu tiên khi bit SM0.1=1.

Thủ tục khai báo sử dụng bộ đếm tốc độ cao bao gồm:

- Nạp giá trị về kiểu hoạt động phù hợp cho byte điều khiển. Ví dụ như khai báo kiểu hoạt động cho HSC1 với:

o Tín hiệu xoá ngoài tích cực khi có logic 1 thì phải ghi 0 vào SM47.0

o Tín hiệu kích start tích cực khi có giá trị logic 1 thì ghi 0 vào SM47.1

o Cho phép kích HSC1 thì ghi 1 vào SM47.7

- Xác định chế độ làm việc cho bộ đếm bằng lệnh HDEF. Ví dụ muốn xác định chế độ làm việc số 3 cho HSC1 thì thực hiện. (HSC0 chỉ có một chế độ đếm là mode0, còn HSC1 và HSC2 có 12 chế độ đếm từ mode0 → mode11)

HDEF K1 K3

- Nạp giá trị đếm tức thời ban đầu và giá trị đặt trước. Ví dụ nạp giá trị tức thời ban đầu là 0 và giá trị đặt trước là 3 cho HSC1

MOVD K0 SMD48

MOVD K3 SMD52

- Khai báo sử dụng chế độ ngắt vào ra và kích tín hiệu báo ngắt. Ví dụ như sử dụng HSC1 làm tín hiệu báo ngắt vào ra mã 13 (khi $PV=CV$) và mã tín hiệu 14 (khi đổi chiều bộ đếm) với các chương trình xử lý ngắt tương ứng có nhãn là 0 và 1 thì thực hiện trong STL như sau

ATCH K0 K13

ATCH K0 K14

- Kích bộ đếm với kiểu làm việc đã ghi trong byte điều khiển bằng lệnh HSC. Ví dụ kích bộ đếm HSC1 theo SMB47 bằng cách thực hiện lệnh trong STL

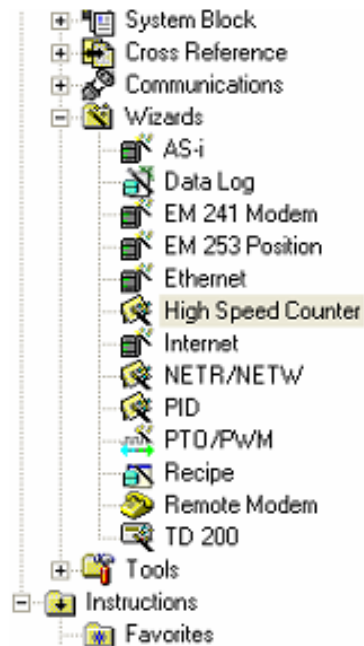
HSC K1

HDEF: lệnh xác định chế độ làm việc cho bộ đếm tốc độ cao. Tên bộ đếm được chỉ định bằng toán hạng HSC. Chế độ làm việc được chọn là nội dung của toán hạng trong MODE.

HSC: lệnh đặt kiểu làm việc cho bộ đếm tốc độ cao. Tên của bộ đếm được chỉ định bằng toán hạng N. Kiểu làm việc được đặt là nội dung của byte điều khiển bộ đếm.

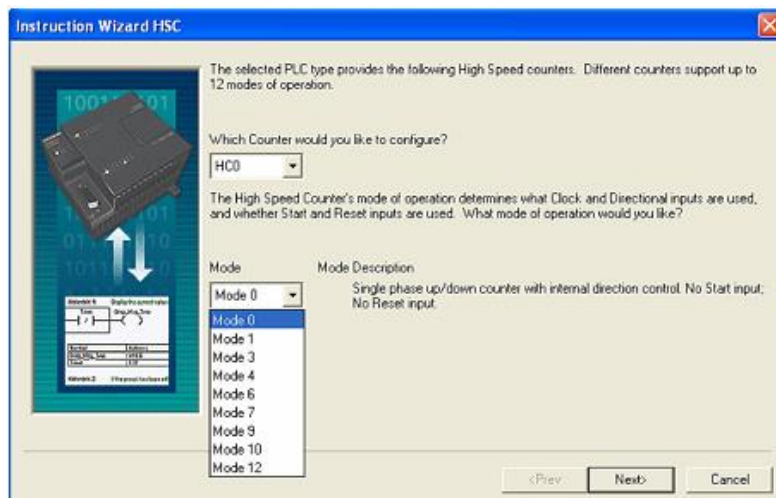
Để khai báo bộ đếm ta có thể dùng chương trình Winzard của PLC như sau:

- Vào Winzard chọn High Speed Counter



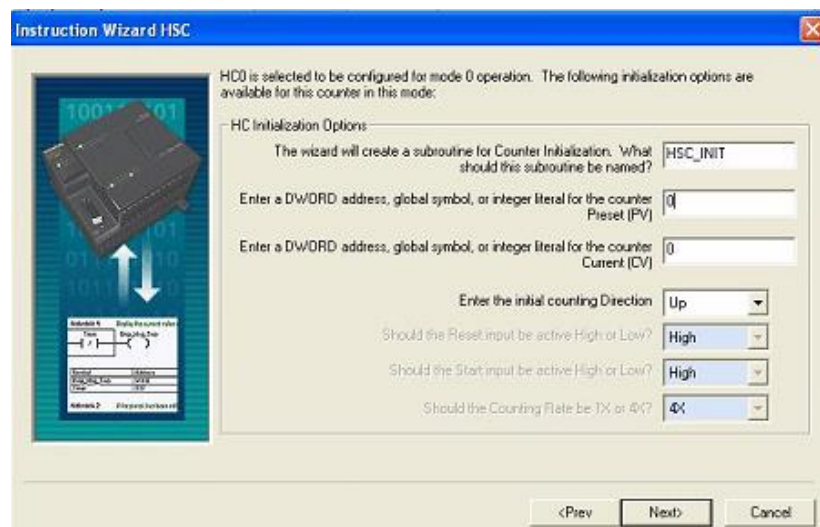
Hình 2.24: Lựa chọn bộ đếm tốc độ cao.

Chọn loại bộ đếm tốc độ cao và chế độ làm việc của bộ đếm



Hình 2.25. Chọn loại bộ đếm.

Chọn ngắt cho bộ đếm, đặt giá trị cho giá trị đặt trước PV và giá trị đếm tức thời VC.



Hình 2.26: Cài đặt tham số cho bộ đếm.

4.3. Bài tập ứng dụng bộ đếm

Chương trình điều khiển máy trộn.

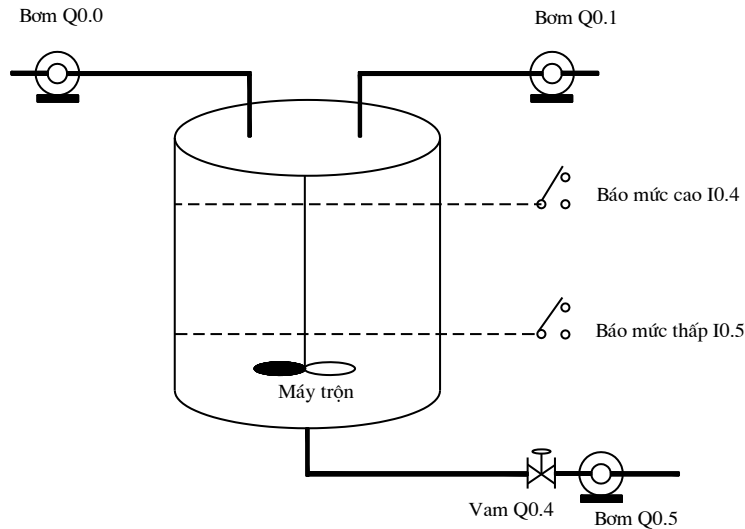
Trong hình dưới là sơ đồ bình trộn để tạo ra các màu khác nhau có 2 cảm biến:

Báo mức cao : I0.4

Báo mức thấp :I0.5

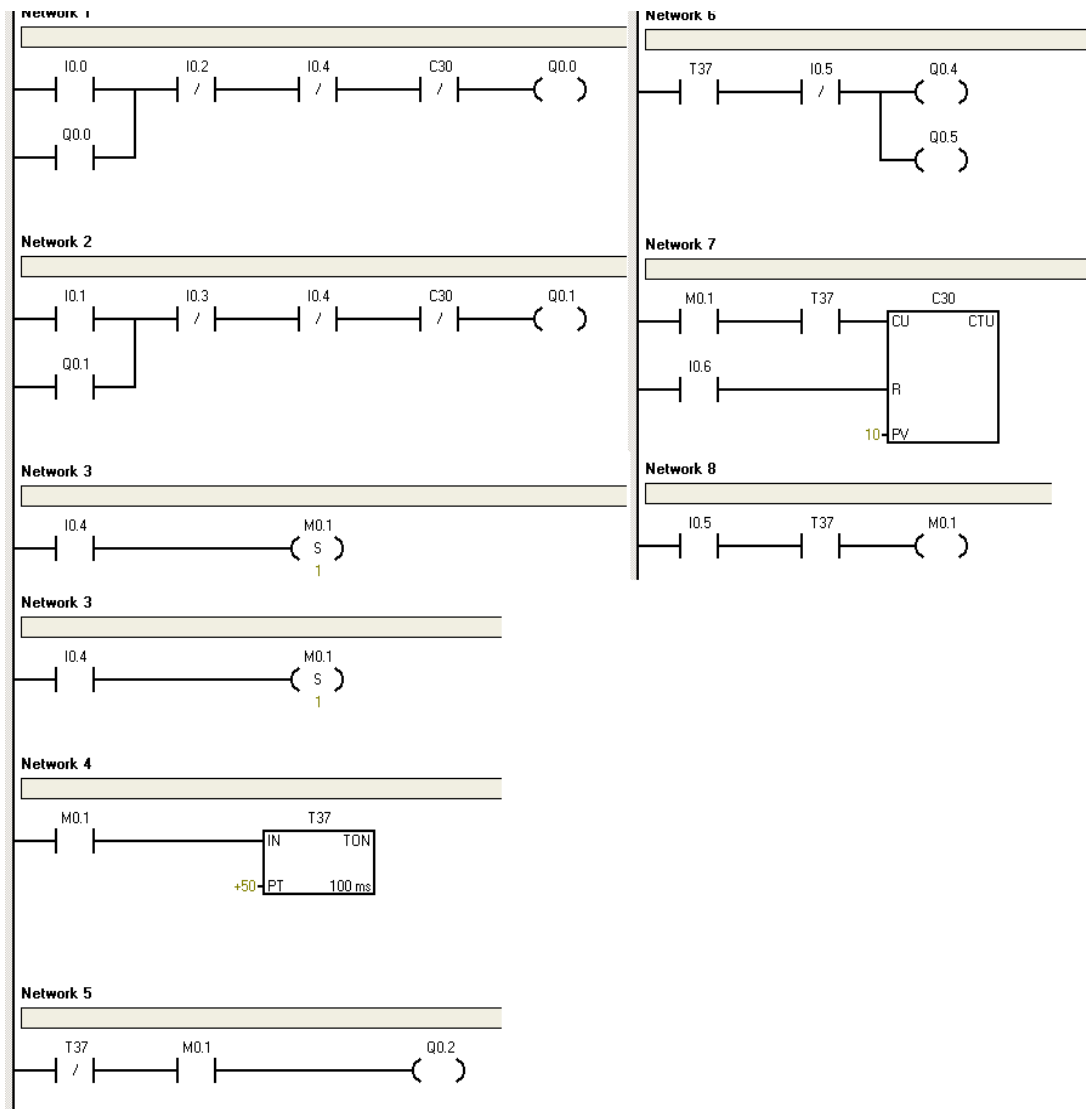
Động cơ trộn điều khiển bởi Q0.2

Quá trình được thực hiện như sau: trước tiên bơm 2 loại sơn khác nhau vào bình. Loại sơn thứ nhất được đưa vào bình máy bơm được điều khiển qua Q0.0. Loại sơn



thứ hai được đưa vào bình nhờ máy bơm thứ 2 được điều khiển qua Q0.1. Sau khi dung dịch trong bình đạt mức cực đại ($I0.4=1$) thì dừng hai máy bơm và bắt đầu quá trình trộn, quá trình này được điều khiển bởi động cơ trộn (Q0.2) và thời gian trộn cần thiết là 5s. Sau khi trộn xong, sản phẩm được đưa ra để rót vào các hộp đựng sơn qua van (Q0.4) và máy bơm (Q0.5). Có thể tóm tắt quá trình trộn như sau:

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



Chương trình được viết trong PLC ở dạng LAD:

5. Bài tập ứng dụng:

Mục tiêu: Nêu một số bài toán thường gặp khi sử dụng PLC.

1, Sử dụng phương pháp mạch tự giữ để khởi động động cơ theo phương pháp sao/tam giác.

2, Sử dụng các tập lệnh về bit để thực hiện khởi động tuần tự động cơ theo thứ tự sau:

- Khi ấn start1: động cơ 1 khởi động, ấn stop1, động cơ 1 tắt.
- Khi động cơ 1 không đủ tải, nhấn start2, động cơ 2 sẽ hoạt động, nhấn stop2, động cơ 2 tắt (khi đã dư tải).
- Tương tự cho động cơ 3 và 4 (sẽ được khởi động khi tải tương ứng không đủ)

Trong quá trình hoạt động gặp sự cố ta có thể nhấn nút dừng khẩn cấp để dừng toàn bộ hệ thống.

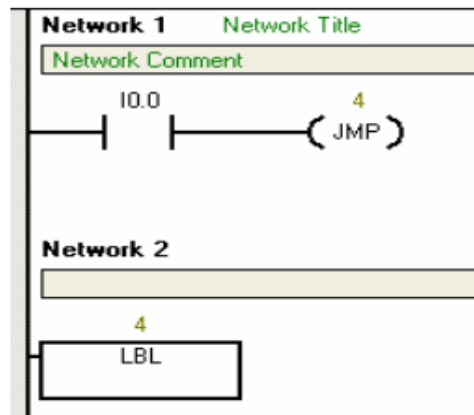
3, Phát hiện chiều di chuyển của vật: Để phát hiện chiều di chuyển của vật, ta phải sử dụng 2 sensor1,2 kế tiếp nhau.

- Trường hợp vật di chuyển theo chiều thuận: sensor 1 tác động rồi đến sensor 2.
- Chiều ngược thì sensor tác động theo thứ tự ngược lại.

4, Điều khiển cho tín hiệu đèn tại các ngã tư giao thông với 2 chế độ ngày và đêm.

6. Lệnh nhảy và lệnh gọi chương trình con

Mục tiêu: Trình bày lệnh nhảy và gọi chương trình con.



Hình 2.27: Mô tả về lệnh nhảy và gọi chương trình con.

Khi I0.0 lên 1, chương trình sẽ thực hiện lệnh nhảy: nhảy tới nhãn tương ứng, khi đó đoạn chương trình ở giữa lệnh nhảy và nhãn sẽ được bỏ qua ở chu kỳ đó.

Kí hiệu của nhãn nhảy phải là một số nguyên n. Việc đặt nhãn cho lệnh nhảy phải nằm trong chương trình. Nhãn của chương trình con hoặc chương trình xử lý ngắt được khai báo ở đầu chương trình. Không thể dùng lệnh nhảy JMP để điều khiển chương trình chính vào một nhãn bất kỳ trong chương trình con hoặc trong chương trình xử lý ngắt.

Bài 3: Lắp đặt mô hình điều khiển bằng PLC

Thời gian: 20 giờ

Mục tiêu của bài:

Học xong bài này học sinh có khả năng:

- Phân tích qui trình công nghệ của một số mạch máy sản xuất.
- Lập trình được một số mạch ứng dụng thường gặp trong thực tế.
- Nạp trình, vận hành và kiểm tra mạch hoạt động theo yêu cầu kỹ thuật.
- Rèn luyện tính cẩn thận, tỉ mỉ, tinh thần làm việc nhóm cho sinh viên

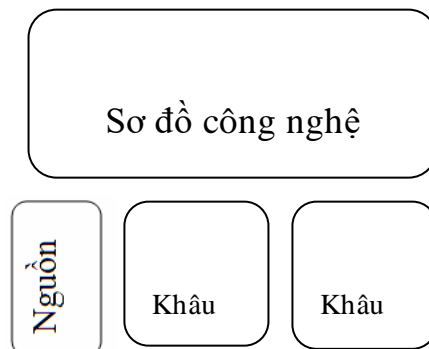
1. Lập trình điều khiển

Mục tiêu: Giới thiệu sơ lược về mô hình bố trí và sử dụng thiết bị thực hành.

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

Từ lý do đó, việc có được những mô hình đáp ứng được những yêu cầu trên là vô cùng cần thiết, có thể dùng cho môn học PLC từ cơ bản đến nâng cao để mô phỏng các quy trình công nghệ trong thực tế sau này.

Nhìn chung, mô hình được bố trí như sau:



Hình 3.1: Cấu trúc mô hình điều khiển.

Mục đích của việc phân thành từng /cụm riêng để giúp học sinh tránh nhầm lẫn đáng tiếc trong quá trình thực hành, đồng thời tiếp thu thêm cách thức tổ chức thực hành.

Các mô hình thực tập gồm có:

- Mô hình thang máy xây dựng
- Mô hình điều khiển động cơ sao – tam giác
- Mô hình xe chuyển nguyên liệu
- Mô hình đo chiều dài và sắp xếp vật liệu
- Mô hình thiết bị nâng hàng hóa
- Mô hình thiết bị vô nước chai
- Mô hình thiết bị trộn hóa chất

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

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

Các mô hình cũng có thể được ứng dụng cho các bộ lập trình cơ nhỏ như LOGO của hãng Siemens, EASY của hãng Moeller, ZEN của omron...

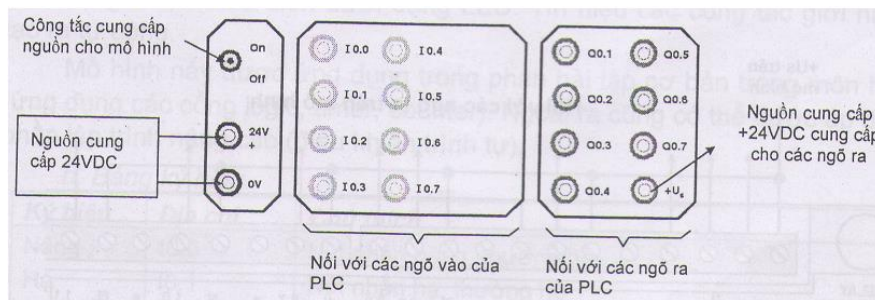
Tùy theo nội dung bài học mà có thể chọn mô hình thích hợp cho bài tập ứng dụng. Một mô hình có thể sử dụng với nhiều bài tập ứng dụng khác nhau.

VD: Mô hình thang máy xây dựng có thể được sử dụng trong các bài học như điều khiển theo tổ hợp logic, điều khiển với các lệnh ghi/xóa tiếp điểm, sử dụng timer, counter, và ứng dụng trong điều khiển trình tự.

2. Cách kết nối dây

Mục tiêu: Trình bày cách kết nối dây của PLC với các thiết bị ngoại vi trong mô hình thực hành.

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

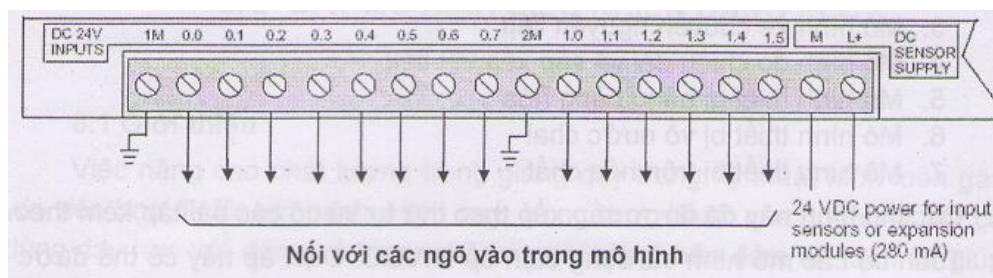


Hình 3.2: Cách kết nối với mô hình.

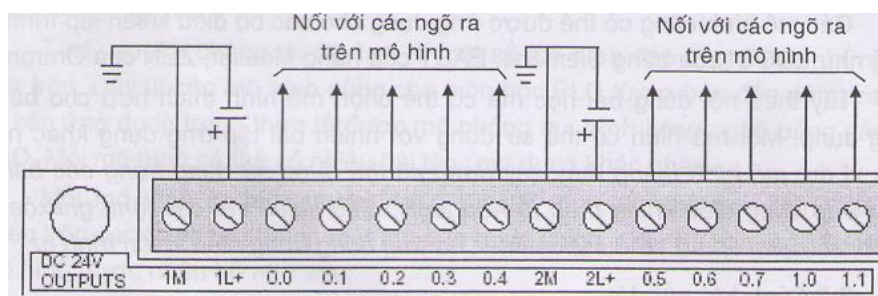
Để kết nối được vi PLC, yêu cầu các modul vào/ra của PLC như sau:

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

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



Hình 3.3: Cách kết nối với các ngõ vào trong mô hình.



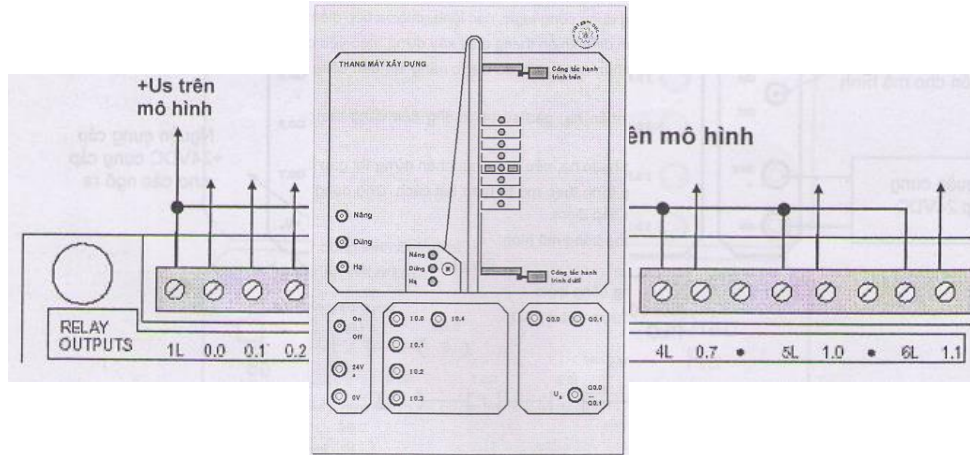
Hình 3.4: Cách kết nối ngõ ra 24VDC của PLC với các ngõ ra trong mô hình.

Hình 3.5: Cách kết nối các ngõ ra của PLC với các ngõ ra trong mô hình.

3. Lập trình trên mô hình và bài tập ứng dụng.

Mục tiêu: Trình bày một số mô hình cụ thể thường sử dụng PLC và các bài toán thông dụng.

3.1. Mô hình thang máy xây dựng.



Hình 3.6: Mô hình thang máy xây dựng.

a, Mô tả

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

Mô hình này được ứng dụng trong phần bìa tập cơ bản trong môn học PLC (ứng dụng các công logic, timer, counter). Ngoài ra cũng có thể được áp dụng cho phần mềm lập trình nâng cao (điều khiển trình tự).

b, Bảng ký hiệu:

Ký hiệu	Địa chỉ	Ghi chú
Nâng	I0.0	Nút nhấn nâng, thường mở
Hạ	I0.1	Nút nhấn hạ, thường mở
Dừng	I0.2	Nút nhấn dừng, thường đóng
GH trên	I0.3	Công tắc hành trình trên, thường đóng
GH dưới	I0.4	Công tắc hành trình dưới, thường đóng
K1	Q0.0	Cuộn dây khởi động từ K1, nâng gầu
K2	Q0.1	Cuộn dây khởi động từ K2, hạ gầu

c, Bài tập mẫu:

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

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

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

- Khi nhấn nút nâng hàng, gầu sẽ được nâng lên đến công tắc giới hạn trên thì dừng lại.
- Khi nhấn nút hạ, gầu sẽ hạ xuống đến công tắc giới hạn dưới thì dừng lại.
- Khi đang nâng hoặc hạ, nếu nhấn nút dừng thì gầu dừng lại.
- Hãy viết chương trình theo mô tả với hai cách: ứng dụng công logic và sử dụng các lệnh ghi xóa tiếp điểm.

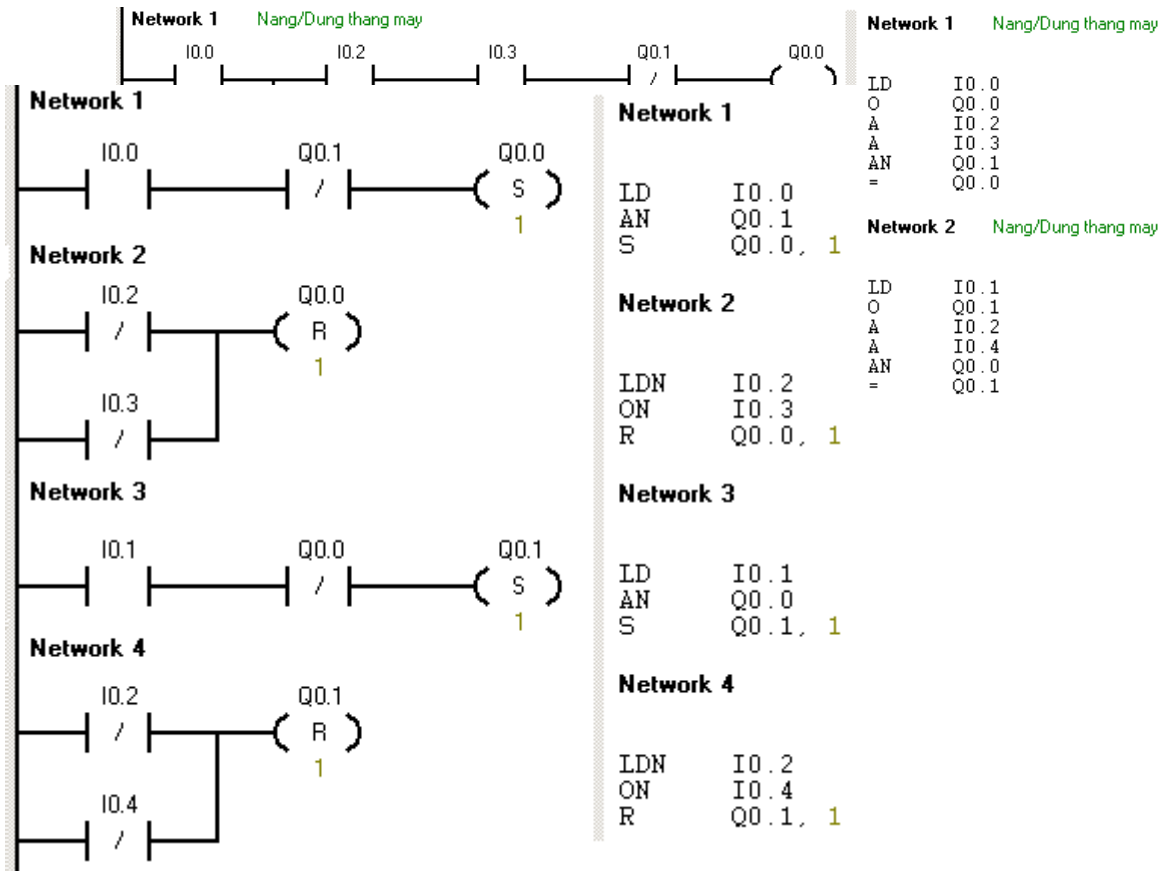
Kiểm tra hoạt động bằng mô hình.

Bài giải

Cách 1: Ứng dụng cổng logic

- Chương trình được viết ở LAD và STL:

Cách 2: Sử dụng lệnh set/reset:



- Chương trình viết ở LAD và STL:

Bài tập 2: Sử dụng timer.

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

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

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

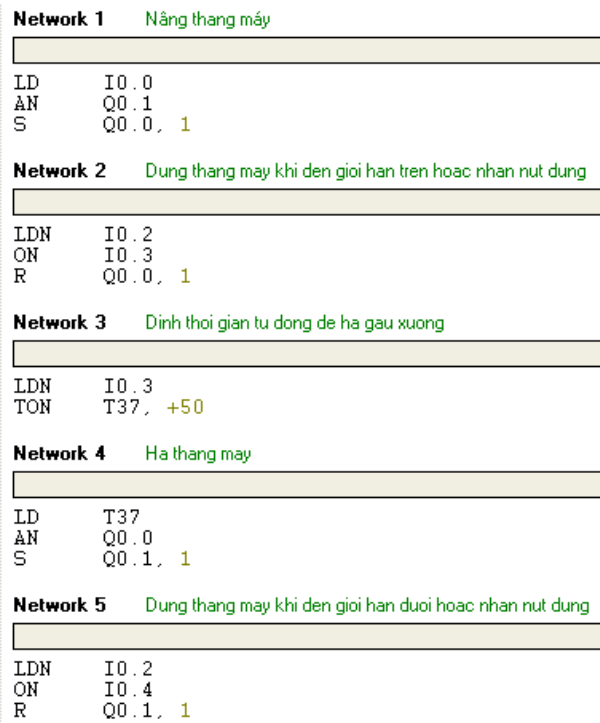
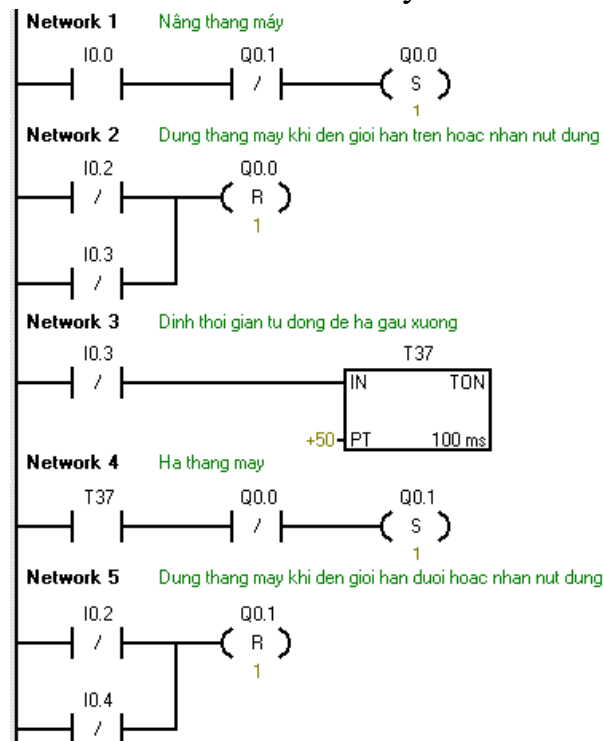
b/. Trong khi ấn nút nâng thì gàu được nâng lên, đến giới hạn trên thì dừng lại 5s, sau đó tự động hạ xuống đến giới hạn dưới thì dừng lại 10s, sau đó tự động nâng lên.

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

Bài giải:

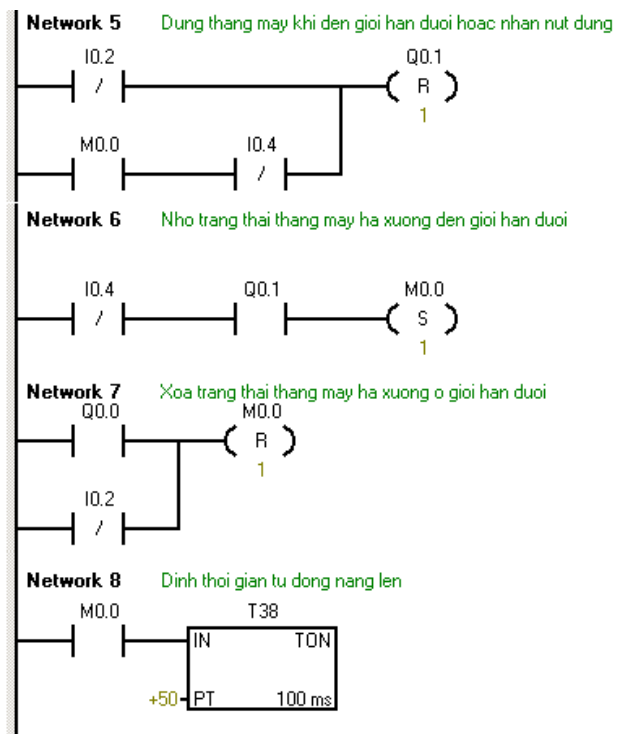
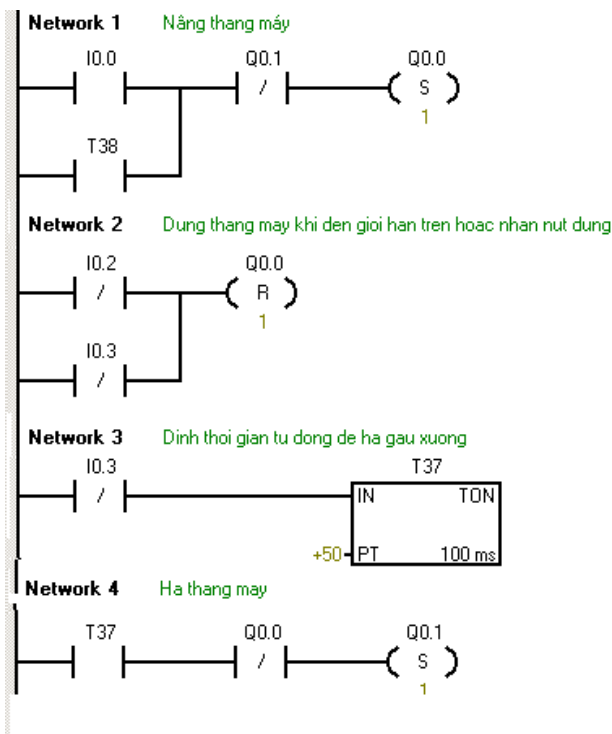
Câu a: Chương trình được viết ở LAD và STL:

Ghi chú: Nếu bài toán có yêu cầu khi nhấn nút hạ thì gâu cũng hạ, lúc này chèn thêm một



tiếp điểm “NO” của I0.1 song song với tiếp điểm “NO” của T37 ở Network 4.

Câu b: Chương trình được viết ở LAD



Chương trình viết ở STL:

Network 1 Nâng thang máy

```
LD I0.0
O T38
AN Q0.1
S Q0.0, 1
```

Network 2 Dừng thang máy khi đến giới hạn trên hoặc nhấn nút dừng

```
LDN I0.2
ON I0.3
R Q0.0, 1
```

Network 3 Định thời gian tự động để hạ gàu xuống

```
LDN I0.3
TON T37, +50
```

Network 4 Hạ thang máy

```
LD T37
AN Q0.0
S Q0.1, 1
```

Network 5 Dừng thang máy khi đến giới hạn dưới hoặc nhấn nút dừng

```
LDN I0.2
LD M0.0
AN I0.4
OLD
R Q0.1, 1
```

Network 6 Nho trạng thái thang máy hạ xuống đến giới hạn dưới

```
LDN I0.4
A Q0.1
S M0.0, 1
```

Network 7 Xóa trạng thái thang máy hạ xuống ở giới hạn dưới

```
LD Q0.0
ON I0.2
R M0.0, 1
```

Network 8 Định thời gian tự động nâng lên

```
LD M0.0
TON T38, +50
```

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

Bài tập 3: Sử dụng bộ đếm

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

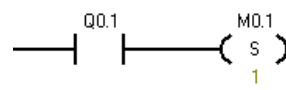
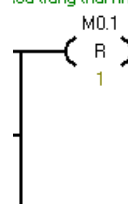
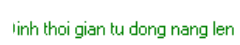
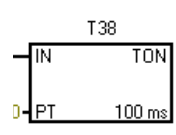
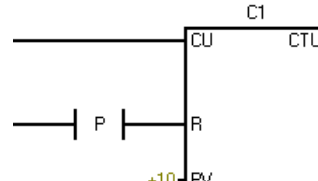
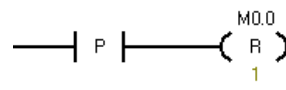
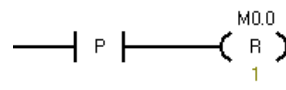
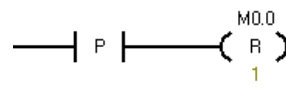
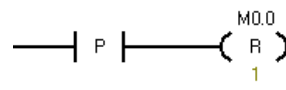
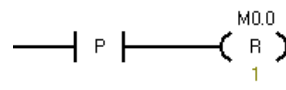
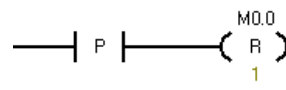
Khi ấn nút nâng thì gàu được nâng lên, đến giới hạn trên thì dừng lại 5s, sau đó tự động hạ gàu xuống đến giới hạn dưới thì dừng lại 10s, sau đó tự động nâng lên. Khi gàu nâng lên được 10 lần thì không nâng lên nữa và sau đó hạ xuống trở về vị trí cơ bản và quá trình lặp lại.

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

Giải:

Chương trình viết ở LAD:

Chương trình viết ở STL:

<p>Network 1 Nho trạng thái nút an năng</p> <pre>LD I0.0 S M0.0, 1</pre>	<p>Nho trạng thái nút an năng</p> 
<p>Network 2 Nang thang may</p> <pre>LD I0.0 ON T38 A M0.0 AN Q0.1 R Q0.0, 1</pre>	<p>Nho trạng thái thang hạ xuống khi đến giới hạn dưới</p> 
<p>Network 3 Dừng thang máy khi đến giới hạn trên hoặc an nút dừng</p> <pre>LDN I0.2 ON I0.3 R Q0.0, 1</pre>	<p>Định thời gian tự động nâng lên</p> 
<p>Network 4 Định thời gian tự động để hạ gầu xuống</p> <pre>LDN I0.3 TON T37, +50</pre>	<p>Định thời gian tự động để hạ gầu xuống</p> 
<p>Network 5 Hạ thang máy</p> <pre>LD T37 AN Q0.0 S Q0.1, 1</pre>	<p>Đếm số lần nâng gầu</p> 
<p>Network 6 Dừng thang máy khi đến giới hạn dưới hoặc an nút dừng</p> <pre>LDN I0.2 LD M0.1 AN I0.4 OLD R Q0.1, 1</pre>	<p>Đồng hồ cho phép gầu tự động nâng lên khi đã nâng đủ 10 lần</p> 
<p>Network 7 Nho trạng thái thang hạ xuống khi đến giới hạn dưới</p> <pre>LDN I0.4 A Q0.1 S M0.1, 1</pre>	<p>Đồng hồ cho phép gầu tự động nâng lên khi đã nâng đủ 10 lần</p> 
<p>Network 8 Xóa trạng thái nhớ thang máy ở dưới hạn dưới</p> <pre>LD Q0.0 ON I0.2 O C1 R M0.1, 1</pre>	<p>Đồng hồ cho phép gầu tự động nâng lên khi đã nâng đủ 10 lần</p> 
<p>Network 9 Định thời gian tự động nâng lên</p> <pre>LD M0.1 TON T38, +50</pre>	<p>Đồng hồ cho phép gầu tự động nâng lên khi đã nâng đủ 10 lần</p> 
<p>Network 10 Đếm số lần nâng gầu</p> <pre>LD M0.1 LD M0.0 EU CTU C1, +10</pre>	<p>Đồng hồ cho phép gầu tự động nâng lên khi đã nâng đủ 10 lần</p> 
<p>Network 11 Không cho phép gầu tự động nâng lên khi đã nâng đủ 10 lần</p> <pre>LD C1 EU R M0.0, 1</pre>	<p>Đồng hồ cho phép gầu tự động nâng lên khi đã nâng đủ 10 lần</p> 

3.2. Mô hình điều khiển động cơ Y-Δ



Hình 6.7: Điều khiển khởi động động cơ.

a, Mô tả:

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

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

Có thể dựa vào mô hình băng tải để lập thành các bài tập khác nhau như: mở máy động cơ Y/ Δ , đảo chiều quay động cơ, điều khiển động cơ ở các tốc độ khác nhau, bồn sậy.

Cách thức nối dây tương tự như trên.

Sau khi đã nối dây với mô hình PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào/ra bằng phần mềm, ví dụ như s7-200 dùng bảng status chart) và sau đó thực hiện mô phỏng với mô hình.

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

** Cách vận hành mô hình:*

Sau khi đã nối dây với mô hình PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào/ra bằng phần mềm, ví dụ như s7-200 dùng bảng status chart) và sau đó thực hiện mô phỏng với mô hình.

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

b. Bảng ký hiệu:

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

c. Bài tập mẫu:

Các bài tập về điều khiển động cơ có thể sử dụng được với mô hình này. Trong phần này đưa ra bốn bài tập mẫu như sau:

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

Động cơ được điều khiển theo yêu cầu sau:

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

Động cơ được cung cấp điện bởi CB 3 pha, và được bảo vệ bởi relay nhiệt và nút dừng stop.

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

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

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

Chú ý: không được phép đảo chiều trực tiếp, cũng như chuyển đổi tốc độ có thể từ thấp sang cao.

Bài 3: (Sử dụng timer) Viết chương trình điều khiển mở máy động cơ Y/ Δ .

Viết chương trình điều khiển mở máy động cơ Y/ Δ theo yêu cầu sau:

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

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

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

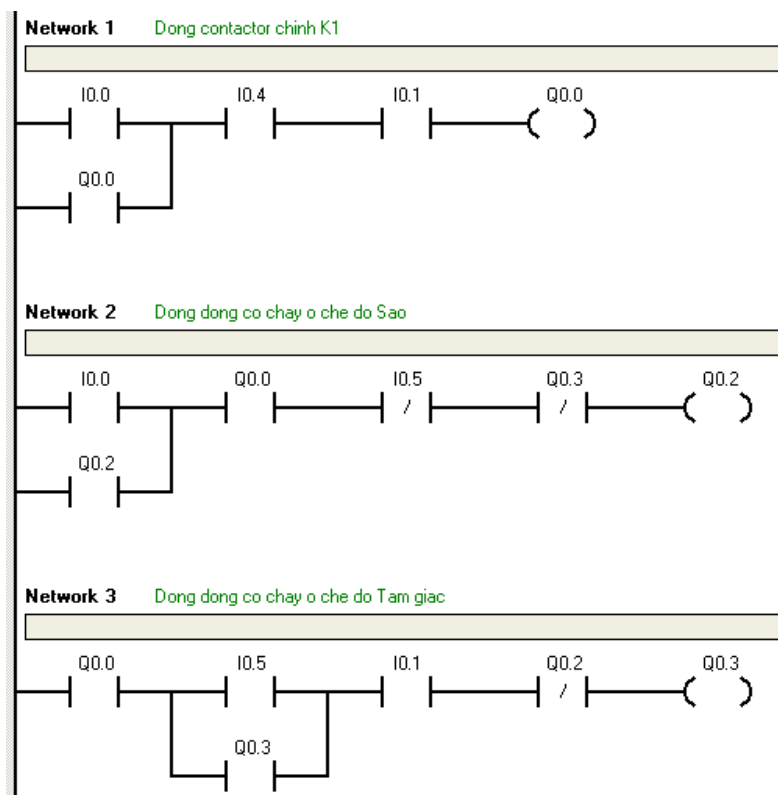
Bài 4: (Điều khiển trình tự) Viết chương trình điều khiển bồn trộn.

Viết chương trình điều khiển bồn trộn theo yêu cầu sau: Khi ấn nút “start”, thì động cơ quay phải ở tốc độ thấp trong thời gian 10s, sau đó dừng 5s, sau đó quay trái 10s, tiếp đó dừng 5s. Quá trình cứ thế lặp lại. Sau khoảng 20 lần thì động cơ dừng 10s và sau đó quay phải ở tốc độ thấp được 5s thì chuyển sang quay ở tốc độ cao khoảng 30s thì dừng hẳn.

Bồn trộn được bảo vệ bởi nút dừng “stop”.

Giải mẫu: (bài tập 1)

Chương trình viết ở LAD:



Chương trình viết ở STL:

Network 1 Dong contactor chinh K1

LD	I0.0
O	Q0.0
A	I0.4
A	I0.1
=	Q0.0

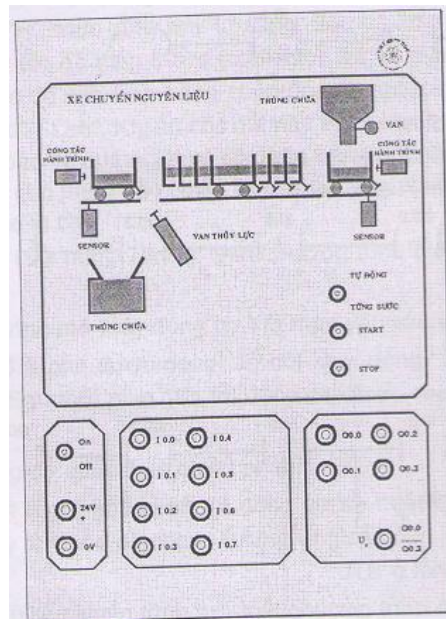
Network 2 Dong dong co chay o che do Sao

LD	I0.0
O	Q0.2
A	Q0.0
AN	I0.5
AN	Q0.3
=	Q0.2

Network 3 Dong dong co chay o che do Tam giac

LD	Q0.0
LD	I0.5
O	Q0.3
ALD	
A	I0.1
AN	Q0.2
=	Q0.3

6.3. Mô hình xe chuyển nguyên liệu:



Hình 6.8: mô hình xe chuyển nguyên liệu

a. Mô tả:

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

Ứng dụng trong PLC cơ bản: điều khiển tổ hợp logic

Ứng dụng trong PLC nâng cao: điều khiển trình tự

Cách thức nối dây tương tự như trên.

b. Cách vận hành mô hình:

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đã đưa ra (có thể tự kiểm tra các ngõ vào/ra bằng phần mềm sau đó thực hiện mô phỏng chương trình).

Nguyên liệu trong bồn chứa khi đang được xả hoặc được rót vào biểu thị bởi những dòng LED chạy, các dòng LED chạy đuổi tượng trưng cho xe chạy ở chế độ tự động, chế độ tay, hoặc hoạt động ở cả hai chế độ.

d. Bảng ký hiệu:

Ký hiệu	Địa chỉ	Chú thích
Start	I0.0	Khởi động hệ thống, thường hở
End 1	I0.1	Công tắc hành trình ở trạm xả, thường đóng
Fill 1	I0.2	Cảm biến báo xe rỗng, thường đóng
End 2	I0.3	Công tắc hành trình trạm nạp, thường đóng
Fill 2	I0.4	Cảm biến báo đầy, thường hở
Stop	I0.5	Dừng, thường đóng
Step	I0.6	Chế độ bước, thường hở
Auto	I0.7	Chế độ tự động, thường hở
Dir_A	Q0.0	Xe chạy về hướng A
Dir_B	Q0.1	Xe chạy về hướng B

Y1	Q0.2	Van xả nguyên liệu
Y2	Q0.3	Van thủy lực

d. Bài tập mẫu:

Xe vận chuyển nguyên liệu hoạt động như sau:

Xe có thể thực hiện thông qua công tắc chuyển chế độ:

- Chế độ tự động I0.6
- Chế độ bước I0.7

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

- Chế độ tự động:

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

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

- Chế độ bước:

Ở mỗi bước thực hiện phải thông qua nút ấn “Start”

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

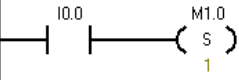
Viết chương trình, kết nối và kiểm tra hoạt động theo hai cách:

- Điều khiển dùng tổ hợp logic
- Điều khiển trình tự

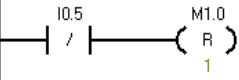
Giải bài tập mẫu: Chương trình được viết theo kiểu trình tự

Chương trình được viết ở LAD:

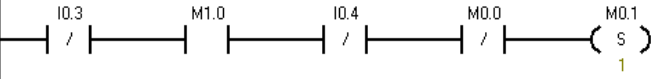
Network 1 Khởi động o nhỏ tự động



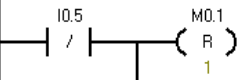
Network 2 Reset o nhỏ tự động



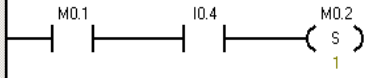
Network 3 Bước 1: lấy hàng



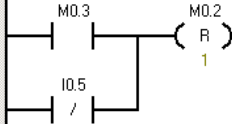
Network 4 Reset bước 1



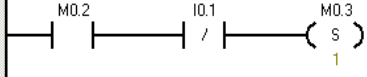
Network 5 Bước 2: Chuyển hàng



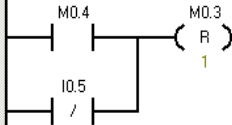
Network 6 Reset bước 2



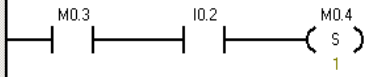
Network 7 Bước 3: Xả hàng



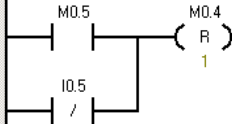
Network 8 Reset bước 3



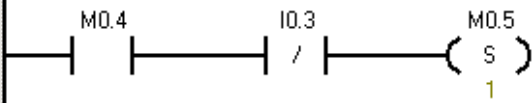
Network 9 Bước 4: trở về



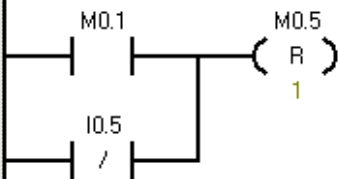
Network 10 Reset bước 4



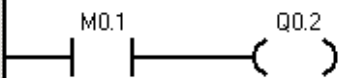
Network 11 Bước 5: Kết thúc



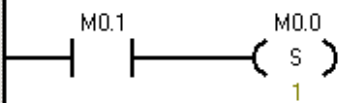
Network 12 Reset bước 5



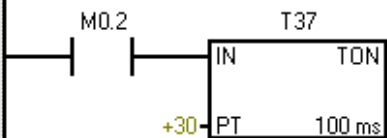
Network 13 lấy hàng



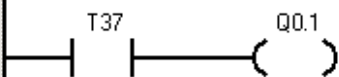
Network 14 Set o nhỏ khởi động



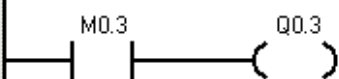
Network 15 Định thời gian on định sau khi hàng đã do dây xe



Network 16 Xe chuyển về trạm xe



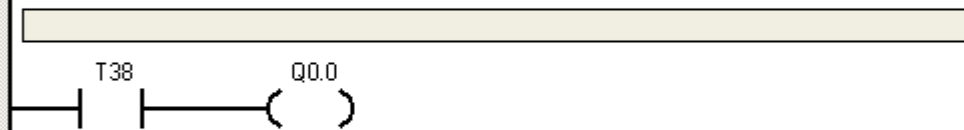
Network 17 Xả hàng



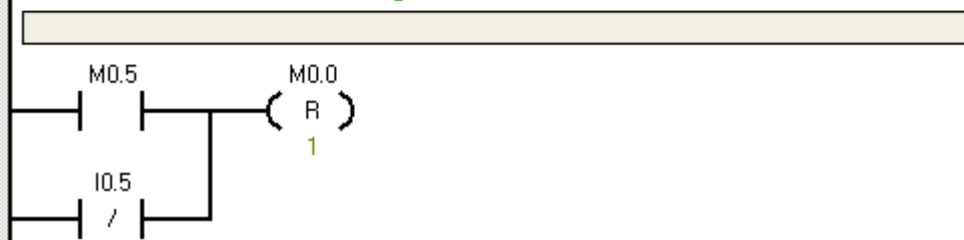
Network 18 Dình thời gian on định sau khi xa hết hàng



Network 19 Di chuyển xe về trạm nhận hàng



Network 20 Reset o nhỏ khởi động



Chương trình được viết ở STL:

Network 1 Khởi động o nhỏ tự động

```
LD I0.0
S M1.0, 1
```

Network 2 Reset o nhỏ tự động

```
LDN I0.5
R M1.0, 1
```

Network 3 Bước 1: lấy hàng

```
LDN I0.3
A M1.0
AN I0.4
AN M0.0
S M0.1, 1
```

Network 4 Reset bước 1

```
LDN I0.5
O M0.2
R M0.1, 1
```

Network 5 Bước 2: Chuyển hàng

```
LD M0.1
A I0.4
S M0.2, 1
```

Network 5 Buoc 2: Chuyen hang

```
LD M0.1
A I0.4
S M0.2, 1
```

Network 6 Reset buoc 2

```
LD M0.3
ON I0.5
R M0.2, 1
```

Network 7 Buoc 3: Xa hang

```
LD M0.2
AN I0.1
S M0.3, 1
```

Network 8 Reset buoc 3

```
LD M0.4
ON I0.5
R M0.3, 1
```

Network 9 Buoc 4: tro ve

```
LD M0.3
A I0.2
S M0.4, 1
```

Network 10 Reset buoc 4

```
LD M0.5
ON I0.5
R M0.4, 1
```

Network 11 Buoc 5: Ket thuc

```
LD M0.4
AN I0.3
S M0.5, 1
```

Network 12 Reset buoc 5

```
LD M0.1
ON I0.5
R M0.5, 1
```

Network 13 lay hang

```
LD M0.1
= Q0.2
```

Network 14 Set o nho khoi dong

```
LD M0.1
S M0.0, 1
```

Network 15 Dình thời gian on dình sau khi hàng đã do đầy xe

LD M0.2
TON T37, +30

Network 16 Xe chuyển về trạm xe

LD T37
= Q0.1

Network 17 Xả hàng

LD M0.3
= Q0.3

Network 18 Dình thời gian on dình sau khi xả hết hàng

LD M0.4
TON T38, +30

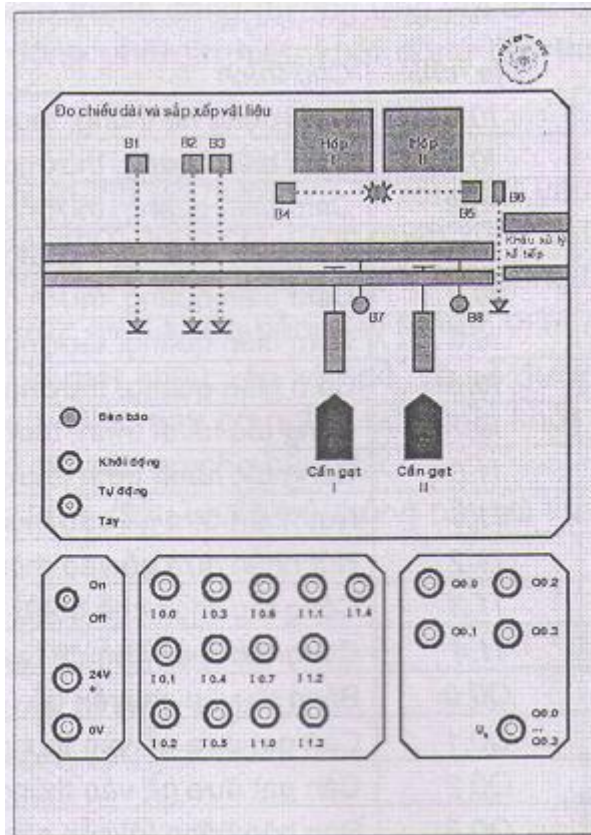
Network 19 Di chuyển xe về trạm nhận hàng

LD T38
= Q0.0

Network 20 Reset o nho khởi động

LD M0.5
ON I0.5
R M0.0, 1

6.4. Đo chiều dài và sắp xếp vật liệu:



Hình 6.9: Mô hình đo chiều dài và sắp xếp vật liệu

a. Mô tả:

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

Ứng dụng trong PLC cơ bản: điều khiển tổ hợp logic

Ứng dụng trong PLC nâng cao: điều khiển trình tự

b. Vận hành mô hình:

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào/ra bằng phần mềm và thực hiện mô phỏng chương trình)

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

c. Bảng ký hiệu:

Ký hiệu	Địa chỉ	Chú thích
Khởi động	I0.0	Khởi động hệ thống, thường hở
B1	I0.1	Cảm biến quang, thường đóng
B2	I0.2	Cảm biến quang, thường đóng

B3	I0.3	Cảm biến quang, thường đóng
B4	I0.4	Cảm biến quang, thường đóng
B5	I0.5	Cảm biến quang, thường đóng
B6	I0.6	Cảm biến quang, thường đóng
B7	I0.7	Công tắc hành trình, thường đóng
B8	I1.0	Công tắc hành trình, thường đóng
SI	I1.1	Nút ấn đưa gỗ vào thùng I
SII	I1.2	Nút ấn đưa gỗ vào thùng II
Tự động	I1.3	Công tắc chọn chế độ tự động
Tay	I1.4	Công tắc chọn chế độ tay
Băng tải	Q0.0	Băng tải vận chuyển gỗ
Cần gạt I	Q0.1	Cần gạt đưa gỗ vào thùng I
Cần gạt II	Q0.2	Cần gạt đưa gỗ vào thùng II
Đèn báo	Q0.3	Đèn báo băng tải sẵn sàng nhận gỗ

d. Bài tập mẫu:

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

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

- Chế độ tự động:

Khi đèn báo sáng báo hệ thống sẵn sàng làm việc. Nhấn nút “khởi động”, đèn báo tắt, tín hiệu khởi động được đưa ra. Các thanh gỗ đơn được đặt lên băng tải và băng tải chuyển động.

Chiều dài thanh gỗ được xác nhận bởi các cảm biến:

Cảm biến B1 tác động tương ứng gỗ ngắn

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

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

Khi gỗ ngắn đến cảm biến B7 thì “Tay gạt 1” sẽ đẩy thanh gỗ này vào thùng 1. Khi gỗ trung bình đến cảm biến B8 thì “Tay gạt 2” sẽ đẩy thanh gỗ này vào thùng 2. Gỗ dài thì được di chuyển tiếp tục đến khâu xử lý kế tiếp. Tay gạt 1 và 2 được sử dụng bằng khí nén, điều khiển khoảng 1s và sau đó trở về vị trí cơ bản của nó.

Sau khi sắp xếp thành công thì thiết bị tự động phát tín hiệu khởi động tiếp theo và băng tải tiếp tục vận chuyển gỗ.

- Chế độ tay:

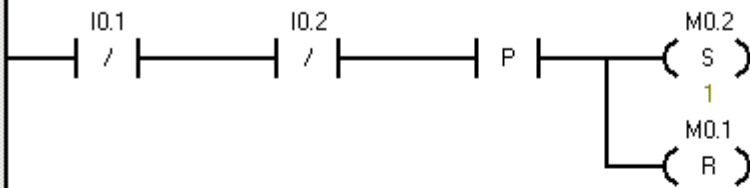
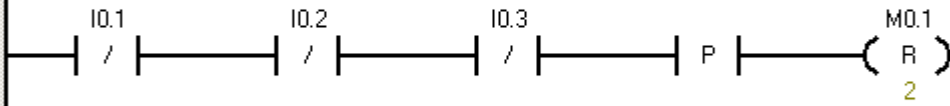
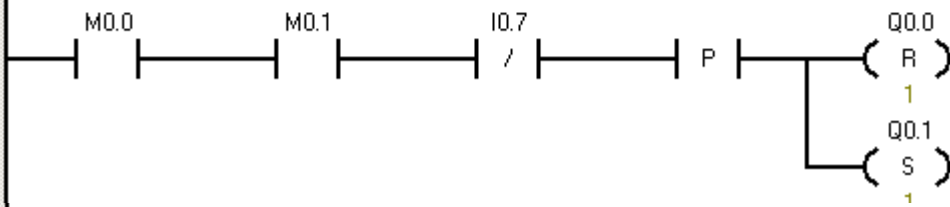
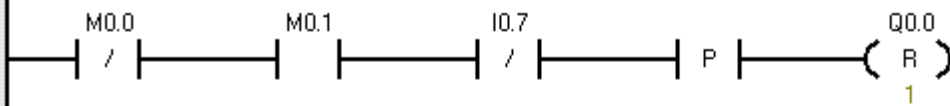
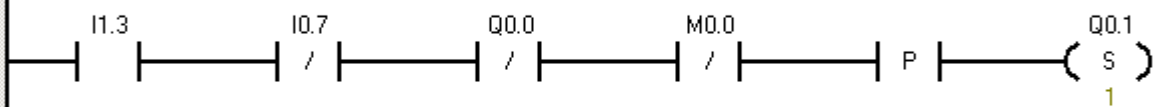
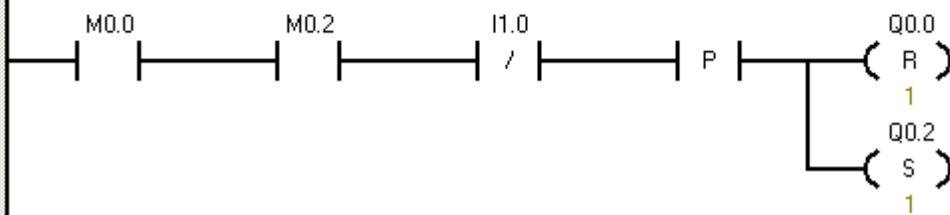
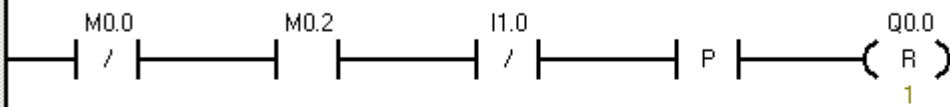
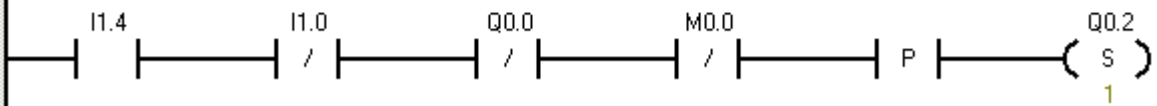
Ở chế độ này, mỗi thanh gỗ được xử lý xong thì yêu cầu khởi động lại hệ thống bằng tay. Tín hiệu khởi động chỉ được phép xử lý nếu việc điều khiển trước đây được báo bằng đèn. Ngay sau khi sắp xếp thành công thì đèn báo lại sáng.

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

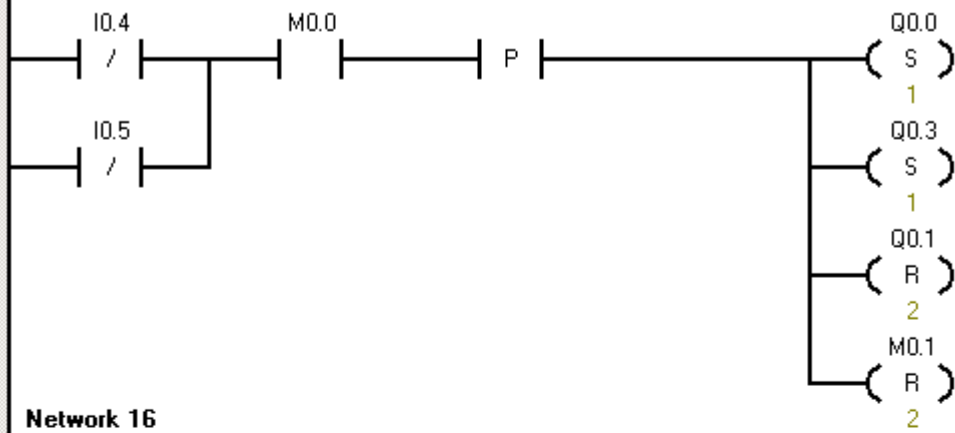
Ghi chú: Đây chỉ là một khâu sắp xếp gỗ, và gỗ được đặt vào băng tải nhờ vào nút ấn khởi động. Điều này có nghĩa là nút ấn khởi động vừa đóng vai trò khởi động, vừa là nơi cung cấp gỗ cho băng tải.

Viết chương trình, kết nối và kiểm tra hoạt động theo hai cách:

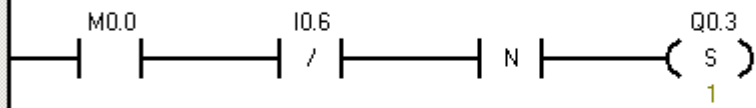
- Điều khiển dùng tổ hợp logic

Network 7**Network 8****Network 9****Network 10****Network 11****Network 12****Network 13****Network 14**

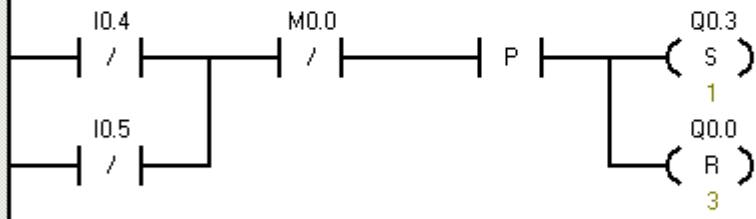
Network 15



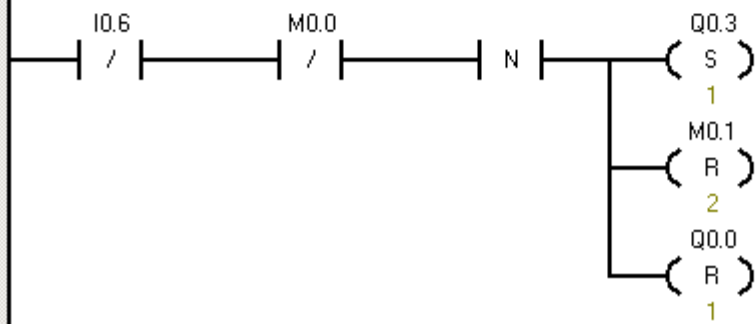
Network 16



Network 17



Network 18



Chương trình viết ở STL:

<p>Network 1 <i>Network Title</i></p> <p>LD SM0.1 S Q0.3, 1 R Q0.0, 3 R M0.0, 3</p>	<p>Network 9</p> <p>LD M0.0 A M0.1 AN I0.7 EU R Q0.0, 1 S Q0.1, 1</p>	<p>Network 15</p> <p>LDN I0.4 ON I0.5 A M0.0 EU S Q0.0, 1 S Q0.3, 1 R Q0.1, 2 R M0.1, 2</p>
<p>Network 2</p> <p>LD I1.1 S M0.0, 1</p>	<p>Network 10</p> <p>LDN M0.0 A M0.1 AN I0.7 EU R Q0.0, 1</p>	<p>Network 16</p> <p>LD M0.0 AN I0.6 ED S Q0.3, 1</p>
<p>Network 3</p> <p>LD I1.2 R M0.0, 1</p>	<p>Network 11</p> <p>LD I1.3 AN I0.7 AN Q0.0 AN M0.0 EU S Q0.1, 1</p>	<p>Network 17</p> <p>LDN I0.4 ON I0.5 AN M0.0 EU S Q0.3, 1 R Q0.0, 3</p>
<p>Network 4</p> <p>LD I1.2 EU R Q0.0, 1</p>	<p>Network 12</p> <p>LD M0.0 A M0.2 AN I1.0 EU R Q0.0, 1 S Q0.2, 1</p>	<p>Network 18</p> <p>LDN I0.6 AN M0.0 ED S Q0.3, 1 R M0.1, 2 R Q0.0, 1</p>
<p>Network 5</p> <p>LD I0.0 EU S Q0.0, 1 R Q0.3, 1</p>	<p>Network 13</p> <p>LDN M0.0 A M0.2 AN I1.0 EU R Q0.0, 1</p>	
<p>Network 6</p> <p>LDN I0.1 EU S M0.1, 1</p>	<p>Network 14</p> <p>LD I1.4 AN I1.0 AN Q0.0 AN M0.0 EU S Q0.2, 1</p>	
<p>Network 7</p> <p>LDN I0.1 AN I0.2 EU S M0.2, 1 R M0.1, 1</p>		
<p>Network 8</p> <p>LDN I0.1 AN I0.2 AN I0.3 EU R M0.1, 2</p>		

6.5. Thiết bị nâng hàng:



Hình 6.10: Mô hình thiết bị nâng hàng

a. Mô tả:

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

Ứng dụng:

Ứng dụng trong PLC cơ bản: điều khiển tổ hợp logic

Ứng dụng trong PLC nâng cao: điều khiển trình tự

b. Vận hành mô hình:

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào/ra bằng phần mềm và thực hiện mô phỏng chương trình).

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

c. Bảng ký hiệu

Ký hiệu	Địa chỉ	Chú thích
ON	I0.0	Khởi động hệ thống, thường hở
OFF	I0.1	Dừng hệ thống, thường đóng
S2	I0.2	Báo hàng ở vị trí cuối bàn nâng, thường đóng
S3	I0.3	Giới hạn dưới bàn nâng, thường đóng
S4	I0.4	Cảm trên bàn nâng, thường đóng
S5	I0.5	Báo hàng ở cuối bàn lăn 2
Thanh chắn	Q0.0	Chặn hàng hóa ở bàn nâng 1

Băng tải	Q0.1	Băng tải chuyển hàng
K1	Q0.2	Nâng hàng hóa lên
K2	Q0.3	Hạ bàn nâng xuống

d. Bài tập mẫu:

Thiết bị nâng hàng hoạt động như sau:

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

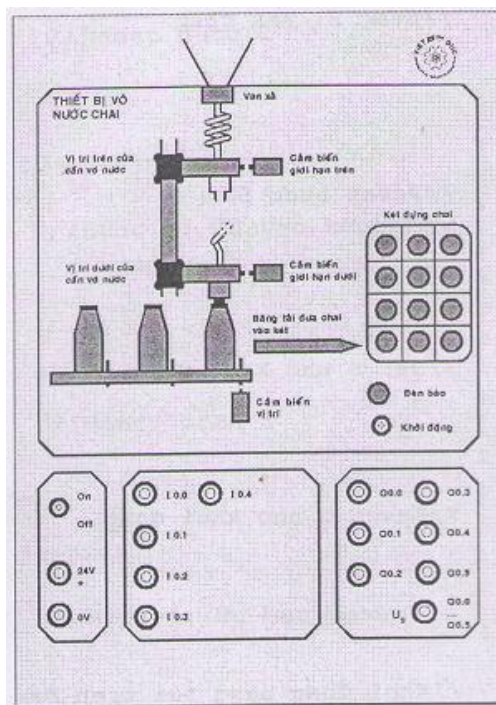
Khi hàng hóa đến vị trí cuối bàn nâng (S2), thì băng tải dừng. Khởi động từ K1 của động cơ M1 có điện kéo bàn nâng lên. Khi đến giới hạn trên thì bàn nâng dừng lại. Băng tải bắt đầu chuyển động đưa hàng sang bàn lăn 2. Khi hàng đến công tắc hành trình S5 thì băng tải dừng. Khởi động từ K2 của động cơ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

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

Viết chương trình, kết nối và kiểm tra hoạt động theo hai cách:

- Điều khiển dùng tổ hợp logic
- Điều khiển trình tự

6.6. Thiết bị vô nước chai:



Hình 6.11: Mô hình thiết bị vô nước chai

a. Mô tả:

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

Ứng dụng trong PLC cơ bản: điều khiển tổ hợp logic

Ứng dụng trong PLC nâng cao: điều khiển trình tự

b. Vận hành mô hình:

Sau khi đã nối dây mô hình với PLC xong, thực hiện viết chương trình theo bài tập đưa ra (có thể tự kiểm tra các ngõ vào/ra bằng phần mềm và thực hiện mô phỏng chương trình).

Hai chai bia trên phải được xem là chai rỗng. Chai ở vị trí thứ 3 được xem như chai đã được đưa đến đúng vị trí. Nước trong chai dâng lên được mô phỏng bằng đèn LED sáng dần. Tùy theo sự sáng dần này mà có thể định thời gian làm đầy chai. Khi chai đã được đổ đầy nước và nếu băng tải vận chuyển chai hoạt động thì chai đầy tự động được chuyển sang băng tải đưa chai vào két. Một tín hiệu sẽ phát ra nếu chai đã đúng vị trí trong két. Khi két đạt đến 12 thì nó không thể tự reset được. Để có thể xóa các LED trong két này phải ấn nút “khởi động”. Để hoạt động thực tế thì khi cần vô nước đến miệng chai phải dừng lại 1s để ổn định.

c. Bảng ký hiệu

Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Giới hạn trên của cần vô nước, thường đóng
S2	I0.1	Giới hạn dưới của cần vô nước, thường đóng
S3	I0.2	Cảm biến vị trí chai, thường hở
S4	I0.3	Khởi động hệ thống, thường hở
S5	I0.4	Chai đúng vị trí trong két, thường hở
K1	Q0.0	Van xả nước
K2	Q0.1	Hạ cần vô nước xuống
K3	Q0.2	Nâng cần vô nước lên
K4	Q0.3	Băng tải vận chuyển chai rỗng
K5	Q0.4	Đèn báo két đầy

d. Bài tập mẫu:

Thiết bị vô nước chai hoạt động như sau:

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

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

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

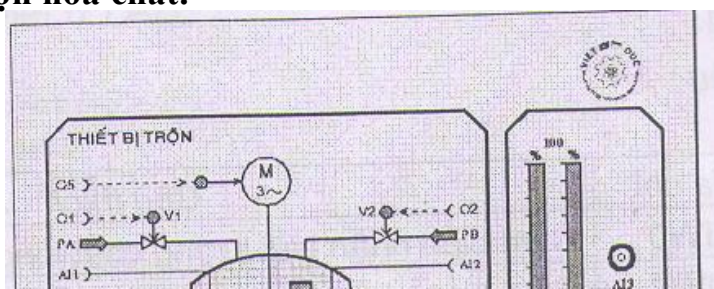
Chai đã đổ đầy nước được đưa sang băng tải đưa chai vào két khi băng tải chai rỗng hoạt động, khi chai đúng vị trí trong két thì có một tín hiệu phát ra (I0.4).

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

Viết chương trình, kết nối và kiểm tra hoạt động theo hai cách:

- Điều khiển dùng tổ hợp logic
- Điều khiển trình tự

6.7. Thiết bị trộn hóa chất.



Hình 6.12: Mô hình thiết bị trộn hóa chất

a. Mô tả:

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

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

b. Vận hành mô hình:

- Vài nét về mô hình

Đây là mô hình thiết bị trộn dùng các cảm biến quang analog. Mô hình này được thiết kế cho PLC gắn các modul ngõ vào analog có điện áp (0.10V). tùy theo loại PLC và cũng tùy theo các modul ngõ vào analog hiện có mà chúng ta có thể đưa ra một số bài tập phù hợp.

- Ghi chú:

Cảm biến đo nhiệt độ ở mô hình này được thiết kế phụ thuộc vào V3. Có nghĩa là nếu V3 được cung cấp điện thì nhiệt độ tăng dần từ khoảng 5⁰C lên và đến cực đại là 100⁰C. Nếu V3 mất điện thì nhiệt độ tự động giảm xuống từ từ.

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

c. Bảng ký hiệu

Ký hiệu	Địa chỉ	Chú thích
A11	AIW0	Cảm biến nhận biết lượng chất lỏng trong bình
A12	AIW2	Cảm biến nhận biết nhiệt độ trong bình
A13	AIW4	Đặt chỉnh trước lượng chất lỏng cần thiết
A14	AIW6	Đặt chỉnh trước nhiệt độ cần thiết
ON	I0.0	Khởi động hệ thống, thường hở

OFF	I0.1	Dừng hệ thống, thường đóng
V1	Q0.0	Bơm chất lỏng A
V2	Q0.1	Bơm chất lỏng B
V3	Q0.2	Cung cấp nhiệt độ cho bình trộn
V4	Q0.3	Van xả chất lỏng
M1	Q0.4	Quạt trộn hóa chất

d. Bài tập mẫu:

Thiết bị trộn hóa chất hoạt động như sau:

Có hai loại chất lỏng A và B cần được trộn với nhau theo tỷ lệ 1/3. Nhiệt độ cần thiết để trộn hai chất này được đặt chính ở AI1. Lượng chất lỏng muốn trộn đặt chính ở AI3.

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

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

Viết chương trình điều khiển cho bồn trộn này.

Yêu cầu về đánh giá kết quả học tập:

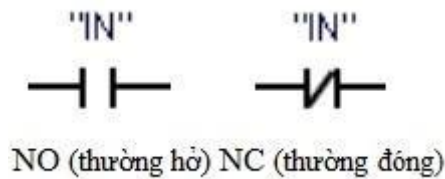
Áp dụng hình thức kiểm tra tích hợp giữa lý thuyết với thực hành. Các nội dung trọng tâm cần kiểm tra là:

- Giải thuật phù hợp đơn giản, ngắn gọn.
- Nạp trình thành thạo, kiểm tra sửa chữa lỗi khi nạp trình.
- Sử dụng đúng các khối chức năng, các lệnh cơ bản (các phép toán nhị phân các phép toán số của PLC, xử lý tín hiệu analog).
- Sử dụng, khai thác thành thạo phần mềm mô phỏng. Thực hiện kết nối tốt với PC.
- Lắp ráp thành thạo mạch động lực đảm bảo kỹ thuật và an toàn.

Mục tiêu của bài:

- Sử dụng, khai thác đúng chức năng các hàm của PLC S7 1200
- 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.

Các tiếp điểm ladder (LAD)



Ta có thể kết nối các tiếp điểm với nhau và tạo ra mạch *logic* kết nối. Nếu *bit* ngõ vào mà ta chỉ rõ sử dụng bộ định danh I (ngõ vào) hay Q (ngõ ra), giá trị *bit* sẽ được đọc từ một thanh ghi ảnh tiến trình. Các tín hiệu tiếp điểm vật lý trong tiến trình điều khiển được nối đến các đầu cực I trên PLC. CPU quét các tín hiệu ngõ vào được nối và cập nhật liên tục các giá trị tương ứng trong thanh ghi ngõ vào ảnh tiến trình.

Ta có thể ghi rõ một kết quả tức thời của một ngõ vào vật lý bằng cách sử dụng “:P” theo sau sự dịch chỉnh I (ví dụ: “%I3.4:P”). Đối với một kết quả tức thời, các giá trị dữ liệu *bit* được đọc một cách trực tiếp từ ngõ vào vật lý thay vì từ ảnh tiến trình. Một kết quả tức thời thì không cập nhật ảnh tiến trình.

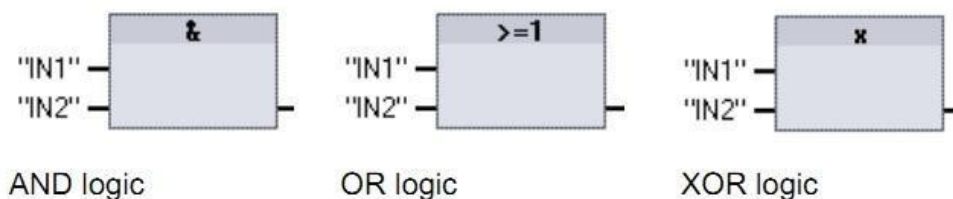
Thông số	Kiểu dữ liệu	Miêu tả
IN	Bool	Bit được gán giá trị

- Tiếp điểm thường hở NO (*Normally Open*) được đóng lại (*ON*) khi giá trị *bit* được gán bằng 1.
- Tiếp điểm thường đóng NC (*Normally Closed*) được đóng lại (*ON*) khi giá trị *bit* được gán bằng 0.
- Các tiếp điểm được nối nối tiếp sẽ tạo ra mạch *logic* AND.
- Các tiếp điểm được nối song song sẽ tạo ra mạch *logic* OR.

Các hộp FBD: AND, OR và XOR

Trong lập trình FBD, các mạng tiếp điểm LAD được chuyển đổi thành các mạng dùng các khối logic AND (&), OR (≥ 1) và OR loại trừ (XOR) mà ta có thể chỉ rõ các giá trị bit cho các ngõ vào và ngõ ra của hộp. Ta còn có thể kết nối đến các hộp logic khác và tạo ra một tổ hợp liên hợp logic riêng. Sau khi hộp được đặt trong mạng, ta có thể kéo công cụ “Insert binary input” từ thanh công cụ “Favorites” hay từ cây lệnh và sau đó thả nó lên trên phía đầu vào của hộp để thêm nhiều ngõ vào. Ta còn có thể nhấp chuột phải lên bộ kết nối ngõ vào của hộp và chọn “Insert input”.

Các ngõ vào và ngõ ra của hộp có thể được kết nối đến một hộp *logic* khác, hay ta có thể nhập vào một địa chỉ *bit* hay tên ký hiệu *bit* đối với một ngõ vào chưa được kết nối. Khi lệnh trong hộp được thực thi, trạng thái ngõ vào hiện tại được áp dụng cho mạch *logic* hộp nhị phân và nếu đúng thì ngõ ra của hộp sẽ là đúng.



Thông số	Kiểu dữ liệu	Miêu tả
IN1, IN2	Bool	Bit ngõ vào

- Tất cả các ngõ vào của hộp AND phải là “TRUE” để ngõ ra là “TRUE”.
- Bất kỳ ngõ vào nào của hộp OR phải là “TRUE” để ngõ ra là “TRUE”.
- Một số lẻ các ngõ vào của hộp XOR phải là “TRUE” để ngõ ra là “TRUE”.

Bộ đảo logic NOT

Đối với lập trình FBD, ta có thể kéo công cụ “Negate binary input” từ thanh công cụ “Favorites” hay từ cây lệnh và sau đó thả nó lên một ngõ vào hay ngõ ra để tạo ra một bộ đảo logic trên bộ kết nối của hộp đó.

┌ NOT ─┐

LAD: bộ đảo tiếp điểm NOT



FBD: hộp AND với 1 ngõ vào logic đảo

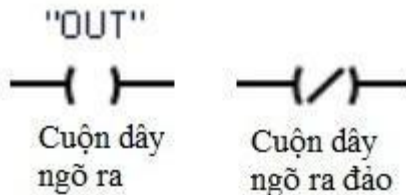


FBD: hộp AND với ngõ vào và ngõ ra logic đảo

Tiếp điểm NOT (LAD) chuyển đổi trạng thái *logic* của đầu vào dòng tín hiệu.

- Nếu không có dòng tín hiệu vào trong tiếp điểm NOT, sẽ có dòng tín hiệu đi ra.
- Nếu có dòng tín hiệu vào trong tiếp điểm NOT, sẽ không có dòng tín hiệu đi ra.

Cuộn dây ngõ ra (LAD)



Lệnh xuất cuộn dây sẽ ghi một giá trị cho một bit ngõ ra. Nếu bit ngõ ra ta chỉ ra sử dụng định danh bộ nhớ Q, thì sau đó CPU sẽ chuyển *bit* ngõ ra trong thanh ghi ảnh tiến trình về *on* hoặc *off*, thiết lập giá trị bit được gán bằng với trạng thái luồng tín hiệu. Các tín hiệu ngõ ra cho cơ cấu điều khiển được nối đến các đầu cực Q của S7 – 1200. Trong chế độ RUN, hệ thống CPU quét một cách liên tục các tín hiệu ngõ vào, xử lý các trạng thái ngõ vào theo chương trình logic, và sau đó tác động trở lại bằng cách thiết lập các giá trị trạng thái ngõ ra mới trong thanh ghi ngõ ra ảnh tiến trình. Sau mỗi chu trình thực thi chương trình, hệ thống CPU chuyển phản ứng trạng thái ngõ ra mới được lưu trữ trong thanh ghi ảnh tiến trình đến các đầu cực nối dây ngõ ra.

Ta có thể xác định một kết quả ghi tức thời của một ngõ ra vật lý bằng cách sử dụng “:P” theo sau độ dịch chuyển Q (ví dụ “%Q3.4:P”). Đối với một kết quả ghi tức

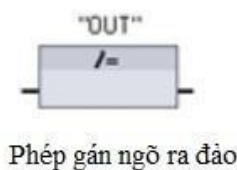
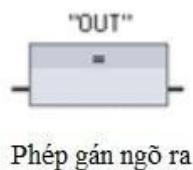
thời, các giá trị dữ liệu bit được ghi đến ngõ ra ảnh tiến trình và trực tiếp đến ngõ ra vật lý.

Thông số	Kiểu dữ liệu	Miêu tả
OUT	Bool	Bit được gán giá trị

- Nếu có luồng tín hiệu chạy qua một cuộn dây ngõ ra, *bit* ngõ ra được đặt lên 1.
- Nếu không có luồng tín hiệu chạy qua một cuộn dây ngõ ra, *bit* ngõ ra được đặt về 0.
- Nếu có luồng tín hiệu chạy qua một cuộn dây ngõ ra đảo, *bit* ngõ ra được đặt về 0.
- Nếu không có luồng tín hiệu chạy qua một cuộn dây ngõ ra đảo, *bit* ngõ ra được đặt lên 1.

Hộp gán ngõ ra (FBD)

Trong lập trình FBD, các cuộn dây LAD được chuyển đổi thành các hộp gán (= và /=) mà ta có thể định rõ một địa chỉ *bit* cho hộp ngõ ra. Các ngõ vào và ngõ ra của hộp có thể được kết nối với khối *logic* khác hay ta có thể nhập vào một địa chỉ *bit*.



Thông số	Kiểu dữ liệu	Miêu tả
OUT	Bool	Bit được gán giá trị

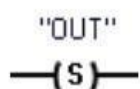
- Nếu tín hiệu vào của hộp ngõ ra là 1, *bit* OUT được đặt lên 1.
- Nếu tín hiệu vào của hộp ngõ ra là 0, *bit* OUT được đặt về 0.
- Nếu tín hiệu vào của hộp ngõ ra đảo là 1, *bit* OUT được đặt về 0.
- Nếu tín hiệu vào của hộp ngõ ra đảo là 0, *bit* OUT được đặt lên 1.

5.1.1.1. Các lệnh Set (đặt) và Reset (đặt lại).

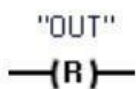
S và R: Set và Reset 1 bit

- Khi lệnh S (Set) được kích hoạt, giá trị dữ liệu ở địa chỉ OUT được đặt lên 1. Khi lệnh S không được kích hoạt, ngõ ra OUT không bị thay đổi.
- Khi lệnh R (Reset) được kích hoạt, giá trị dữ liệu ở địa chỉ OUT được đặt về 0. Khi lệnh R không được kích hoạt, ngõ ra OUT không bị thay đổi.
- Những lệnh này có thể được đặt tại bất cứ vị trí nào trong mạch.

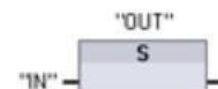
LAD: Set



LAD: Reset



FBD: Set



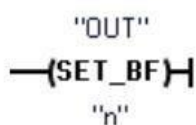
FBD: Reset



Thông số	Kiểu dữ liệu	Miêu tả
IN	Bool	Vị trí <i>bit</i> được giám sát
OUT	Bool	Vị trí <i>bit</i> được đặt hoặc đặt lại

SET_BF và RESET_BF: Set và Reset một trường bit

LAD: SET_BF



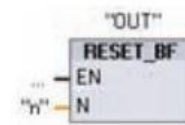
LAD: RESET_BF



FBD: SET_BF



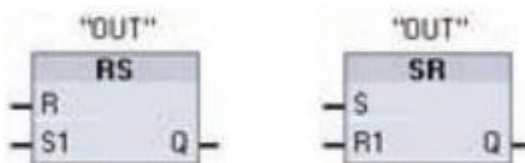
FBD: RESET_BF



Thông số	Kiểu dữ liệu	Miêu tả
n	Constant	Số lượng các <i>bit</i> để ghi
OUT	Phần tử của một mảng Boolean	Phần tử bắt đầu của một trường bit được đặt hay đặt lại. Ví dụ #MyArray[3]

- Khi SET_BF được kích hoạt, một giá trị dữ liệu bằng 1 được gán cho “n” bit bắt đầu tại địa chỉ OUT. Khi SET_BF không được kích hoạt, địa chỉ OUT không bị thay đổi.
- RESET_BF ghi một giá trị dữ liệu bằng 0 đến “n” bit bắt đầu tại địa chỉ OUT. Khi RESET_BF không được kích hoạt, địa chỉ OUT không bị thay đổi.
- Những lệnh này phải là lệnh nằm về bên phải trong một nhánh.

RS và SR: các mạch chốt của bit set trội và reset trội



RS là một mạch chốt *set* trội mà *set* chiếm ưu thế. Nếu tín hiệu *set* (S1) và *reset* (R) đều là đúng, địa chỉ ngõ ra OUT sẽ bằng 1.

SR là một mạch chốt *reset* trội mà *reset* chiếm ưu thế. Nếu tín hiệu *set* (S) và *reset* (R1) đều là đúng thì địa chỉ ngõ ra OUT sẽ là 0.

Thông số OUT định rõ địa chỉ *bit* được *set* hay *reset*. Ngõ ra OUT tùy chọn (Q) phản ánh trạng thái tín hiệu của địa chỉ OUT.

Thông số	Kiểu dữ liệu	Miêu tả
S, S1	Bool	Ngõ vào <i>set</i> ; số “1” biểu thị sự ưu thế
R, R1	Bool	Ngõ vào <i>reset</i> ; số “1” biểu thị sự ưu thế
OUT	Bool	Ngõ ra của <i>bit</i> được gán “OUT”
Q	Bool	Trạng thái kèm theo của <i>bit</i> “OUT”

Lệnh	S1	R	Bit “OUT”
RS	0	0	Trạng thái kế trước
	0	1	0
	1	0	1
	1	1	1
	S	R1	Bit “OUT”
SR	0	0	Trạng thái kế trước
	0	1	0
	1	0	1
	1	1	0

