

ỦY BAN NHÂN DÂN HUYỆN CỬ CHI
TRƯỜNG TRUNG CẤP NGHỀ CỬ CHI

GIÁO TRÌNH

MÔ ĐUN: LẬP TRÌNH VÀ ĐIỀU KHIỂN PLC
NGHỀ: KỸ THUẬT MÁY LẠNH VÀ ĐIỀU HÒA KHÔNG KHÍ
TRÌNH ĐỘ: TRUNG CẤP NGHỀ

*Ban hành kèm theo Quyết định số 89/QĐ-TCNCC ngày 15 tháng 8 năm 2024 của
Hiệu trưởng trường Trung Cấp Nghề Cử Chi*

Tp. Hồ Chí Minh, năm 2024

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 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 180 giờ gồm các bài:

Bài 01: Đại cương về điều khiển lập trình

Bài 02: Cấu trúc và phương thức hoạt động của một PLC

Bài 03: Kết nối giữa PLC và thiết bị ngoại vi

Bài 04: Các phép toán nhị phân của PLC

Bài 05: Các phép toán số của PLC

Bài 06: Bộ xử lý tín hiệu Analog

Bài 07: Các bài tập ứng dụng trong điều khiển động cơ

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 củng cố và áp dụng kiến thức phù hợp với kỹ năng.

Tuy nhiên, tùy theo điều kiện cơ sở vật chất và trang thiết bị, 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.

Củ Chi, ngày.....tháng..... năm 2024

Tham gia biên soạn

Mục lục

TUYÊN BỐ BẢN QUYỀN	i
LỜI GIỚI THIỆU	ii
Mục lục	iii
MÔ ĐUN: PLC CƠ BẢN	vi
Bài 1: ĐẠI CƯƠNG VỀ ĐIỀU KHIỂN LẬP TRÌNH.....	1
1.1 TỔNG QUAN VỀ ĐIỀU KHIỂN	1
1.2 ĐIỀU KHIỂN NÓI CỨNG VÀ ĐIỀU KHIỂN LẬP TRÌNH ĐƯỢC.....	3
1.2.1 Phương pháp điều khiển nói cứng (Hard-wired control).....	3
1.2.2 Phương pháp điều khiển lập trình được.....	4
1.3 SO SÁNH PLC VỚI CÁC HÌNH THỨC ĐIỀU KHIỂN KHÁC.....	6
1.3.1 Hệ thống điều khiển dùng Rơ-le	6
1.3.2 Hệ thống điều khiển dùng mạch điện tử (Transistor).....	6
1.3.3 Hệ thống điều khiển dùng IC số.....	7
1.3.4 Hệ thống điều khiển dùng máy tính	7
1.4 CÁC ỨNG DỤNG CỦA PLC TRONG THỰC TẾ.....	7
Bài 2:CẤU TRÚC VÀ PHƯƠNG THỨC HOẠT ĐỘNG CỦA MỘT PLC	9
2.1 CẤU TRÚC CỦA MỘT PLC	9
2.1.1 Đơn vị xử lý trung tâm (CPU Central Processing Unit)	10
2.1.2 Bộ nhớ.....	10
2.1.3 Khối ngõ vào / ra	10
2.2 THIẾT BỊ ĐIỀU KHIỂN LẬP TRÌNH S7-200	11
2.2.1 Khái quát chung	11
2.2.2 Các đèn báo trên S7-200 CPU 224	13
2.2.3 Chọn chế độ làm việc cho PLC	13
2.2.4 Cổng truyền thông trên S7-200	13
2.3 ĐỊA CHỈ CÁC NGÕ VÀO RA	14
2.4. CẤU TRÚC BỘ NHỚ CỦA S7-200.....	15
2.5 XỬ LÝ CHƯƠNG TRÌNH	16
2.5.1. Truy xuất dữ liệu	16
2.5.2 Xử lý chương trình	18
2.5.3 Cấu trúc chương trình	20
2.5.4 Phương pháp lập trình PLC S7-200	21
Bài 3: KẾT NỐI DÂY GIỮA PLC VÀ THIẾT BỊ NGOẠI VI	24
3.1 KẾT NỐI DÂY GIỮA PLC VÀ THIẾT BỊ NGOẠI VI.....	24
3.1.1 Giới thiệu CPU 224 và cách kết nối với thiết bị ngoại vi.....	24
3.1.2 Kết nối với máy tính	25
3.1.3 Nối nguồn cung cấp cho CPU.....	26
3.1.4 Kết nối vào/ra số với ngoại vi.....	27
3.2. CÀI ĐẶT VÀ SỬ DỤNG PHẦN MỀM STEP7-MICROWIN 4.0.....	31
3.2.1 Cài đặt phần mềm Step7-Microwin 4.0.....	31

3.2.2 Các phần tử cơ bản trong chương trình PLC S7-200	33
3.3.3 Ngôn ngữ lập trình.....	35
3.3.4 Soạn thảo chương trình với phần mềm STEP7-Micro/Win V4.0.....	36
3.3 KIỂM TRA VIỆC NỐI DÂY BẰNG PHẦN MỀM	48
3.4 THỰC HÀNH KẾT NỐI DÂY PLC VÀ THIẾT BỊ NGOẠI VI.....	50
3.5 CÀI ĐẶT VÀ SỬ DỤNG PHẦN MỀM STEP7- MICROWIN.....	53
3.6 Bài tập ứng dụng.....	53
Bài 4: CÁC PHÉP TOÁN NHỊ PHÂN CỦA PLC	55
4.1 CÁC LIÊN KẾT LOGIC	55
4.1.1 Các phép toán Logic cơ bản.....	55
4.1.2 Các lệnh ghi / xoá giá trị cho tiếp điểm	61
4.2 TIMER	64
4.2.1 Giới thiệu	64
4.2.2 Timer đóng chậm TON không có nhớ	65
4.2.3 Timer đóng mạch chậm có nhớ TONR	66
4.2.4 Timer mở chậm TOF	66
4.2.5 Thực hành với Timer	67
4.3 CUONTER	70
4.3.1 Bộ đếm lên CTU (Count Up)	70
4.3.2 Bộ đếm xuống CTD (Count Down)	71
4.3.3 Bộ đếm lên-xuống CTUD (Count Up/Down).....	72
Bài 5: CÁC PHÉP TOÁN SỐ CỦA PLC.....	78
5.1 CHỨC NĂNG TRUYỀN DẪN	78
5.1.1 Truyền Byte ,Word, Doubleword.....	78
5.1.2 Truyền một vùng nhớ dữ liệu	79
5.2 CHỨC NĂNG SO SÁNH.....	81
5.3 CHỨC NĂNG DỊCH CHUYỂN	83
5.3.1 Các lệnh sao chép, trao đổi nội dung.....	83
5.3.2 Các lệnh sao chép một mảng lớn dữ liệu.....	85
5.4 CHỨC NĂNG CHUYỂN ĐỔI	86
5.5 CHỨC NĂNG TOÁN HỌC.....	90
5.5.1 Cộng và trừ.....	90
5.5.2 Nhân và chia	90
5.5.3 Thực hành phép toán số học.....	90
Bài 6: XỬ LÝ TÍN HIỆU ANALOG	96
6.1 TÍN HIỆU ANALOG	96
6.1.1 Đọc tín hiệu analog từ Modul EM231	96
6.1.2 Xuất tín hiệu analog qua modul EM232.....	98
6.1.3 Modul EM235	98
6.1.4 I/O cục bộ và mở rộng	98
6.1.5 Lọc đầu vào tương tự	99

6.1.6 Điều chỉnh tương tự	100
6.2 Thực hành đo lường và giám sát nhiệt độ với module EM235 và nhận cảm biến nhiệt điện trở Pt100	100
<i>BÀI TẬP THỰC HÀNH</i>	<i>108</i>
Bài 7: CÁC BÀI TẬP ỨNG DỤNG TRONG ĐIỀU KHIỂN	109
7.1 ON/OFF	109
7.2 MẠCH KHỞI ĐỘNG ĐỘNG CƠ	109
7.3 ĐẢO CHIỀU ĐỘNG CƠ1	114
7.4 ĐẢO CHIỀU ĐỘNG CƠ 2	115
7.5 KHỞI ĐỘNG SAO TAM GIÁC	116
7.6 TỰ ĐỘNG ĐÓNG CÁC CẤP ĐIỆN TRỞ	117
7.7 BẢNG TẢI 1	118
7.8 BẢNG TẢI 2	120
7.9 ĐÈN GIAO THÔNG	121
TÀI LIỆU THAM KHẢO	124

MÔ ĐƠN: LẬP TRÌNH VÀ ĐIỀU KHIỂN PLC

Mã mô đun: MĐ20

I. Vị trí, tính chất, ý nghĩa và vai trò của môn học:

- Vị trí của môn học: Môđun được bố trí dạy cuối chương trình sau khi học xong các môn chuyên môn như điện tử công suất, Kỹ thuật xung – số, Vi xử lí, trang bị điện...
- Tính chất của môn học: Mô đun PLC cơ bản mang tính tích hợp.
- Ý nghĩa của mô đun: Là môn học bắt buộc
- Vai trò của mô đun: Sau khi học xong mô đun này, người học có thể kết nối dây giữa PC - PLC và thiết bị ngoại vi, viết chương trình, nạp trình để thực hiện được một số bài toán ứng dụng đơn giản trong công nghiệp, phân tích luận lý một số chương trình, phát hiện sai lỗi và sửa chữa khắc phục.

II. Mục tiêu của Mô đun:

Sau khi học xong mô đun này học viên có năng lực:

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

ĐẠI CƯƠNG VỀ ĐIỀU KHIỂN LẬP TRÌNH

GIỚI THIỆU

Như đã biết, nước ta hiện nay đang trong quá trình công nghiệp hóa, hiện đại hóa. Vì thế, tự động hóa sản xuất đóng vai trò quan trọng, tự động hóa giúp tăng năng suất, tăng độ chính xác và do đó tăng hiệu quả quá trình sản xuất. Để có thể thực hiện tự động hóa sản xuất, bên cạnh các máy móc cơ khí hay điện, các dây chuyền sản xuất...v.v, cũng cần thiết phải có các bộ điều khiển để điều khiển chúng. Trong đó, được yêu cầu đó điều khiển lập trình là một trong các bộ điều khiển đáp ứng.

MỤC TIÊU BÀI HỌC

Kiến thức:

- Phát biểu khái niệm về điều khiển lập trình.

Kỹ năng:

- So sánh ưu nhược điểm của điều khiển lập trình được với các hình thức điều khiển khác.
- Trình bày các ứng dụng của điều khiển lập trình PLC trong thực tế.

Thái độ:

- Rèn luyện tính tư duy, tác phong công nghiệp.

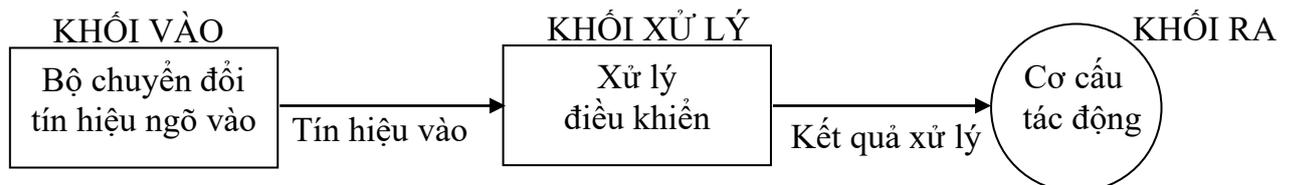
NỘI DUNG BÀI HỌC

1.1 TỔNG QUAN VỀ ĐIỀU KHIỂN

Trong mọi ngành sản xuất, mục tiêu tăng năng suất lao động được giải quyết bằng con đường gia tăng mức độ tự động hoá các quá trình và thiết bị sản xuất. Việc tự động hoá có thể nhằm mục đích tăng sản lượng hoặc cải thiện chất lượng và độ chính xác của sản phẩm.

Tự động hoá nhằm thay thế một phần hoặc toàn bộ các thao tác vật lý của công nhân hoặc thiết bị thông qua hệ thống điều khiển. Hệ thống này có thể điều khiển quá trình sản xuất với độ tin cậy cao, ổn định mà không cần hoặc cần rất ít sự can thiệp của con người. Điều này đòi hỏi hệ thống phải có khả năng khởi động, kiểm soát và dừng một quá trình theo yêu cầu sản phẩm. Một hệ thống có khả năng như vậy được gọi là **hệ thống điều khiển**.

Một hệ thống điều khiển bất kỳ được cấu tạo từ ba thành phần: khối vào, khối xử lý và khối ra.



Hình 1.1: Các khối trong hệ thống điều khiển

Khối vào là bộ chuyển đổi các đại lượng vật lý thành tín hiệu điện như: nút nhấn, công tắc, cảm biến.... Tùy theo loại chuyển đổi mà các tín hiệu tạo ra có dạng tương tự (analog) hay dạng số (binary).

Khối xử lý thay thế người vận hành thực hiện các thao tác đảm bảo các quá trình hoạt động “có sự điều khiển”. Nó nhận thông tin từ các tín hiệu ở khối vào và xuất tín hiệu đến khối ra để thực hiện tác động đến thiết bị.

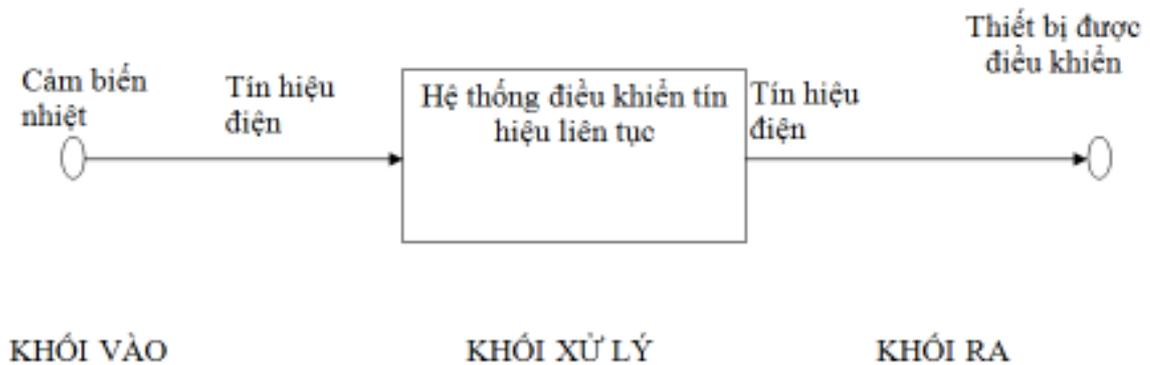
Khối ra Tín hiệu ra là kết quả của quá trình xử lý của hệ thống điều khiển. các tín hiệu này tạo ra những hoạt động cụ thể ở máy hoặc thiết bị nhằm đảm bảo thực hiện quá trình mục tiêu. Quá trình mục tiêu do những thiết bị ở ngõ ra như: động cơ, xy lanh khí nén, dầu ép, bơm, rơ le... cũng như các bộ chuyển đổi tín hiệu vào, các thiết bị ngõ ra có thể làm việc với tín hiệu dạng Analog hoặc Digital.

Hệ thống điều khiển được chia thành hai dạng:

Điều khiển liên tục điều khiển tín hiệu liên tục nhận trực tiếp từ các bộ cảm biến (biến đổi các đại lượng không điện thành tín hiệu điện) điều chế tín hiệu thông qua các mạch khuếch đại, mạch cộng, mạch tích phân... và xuất tín hiệu điều khiển đến các cơ cấu tác động như: van, bơm, đầu phát nhiệt... các cơ cấu tác động có thể là các thiết bị hoạt động liên tục hay hoạt động với hai trạng thái on/off.

Việc xử lý tín hiệu liên tục trong hệ thống phụ thuộc vào các quá trình xử lý tương ứng, nhưng thường phải qua mạch khuếch đại và một số phép tính toán để tạo ra sự thay đổi ở các ngõ ra theo mong muốn.

Bộ điều khiển xử lý tín hiệu liên tục gồm mạch điện tử tuyến tính, máy tính hoặc máy vi tính.



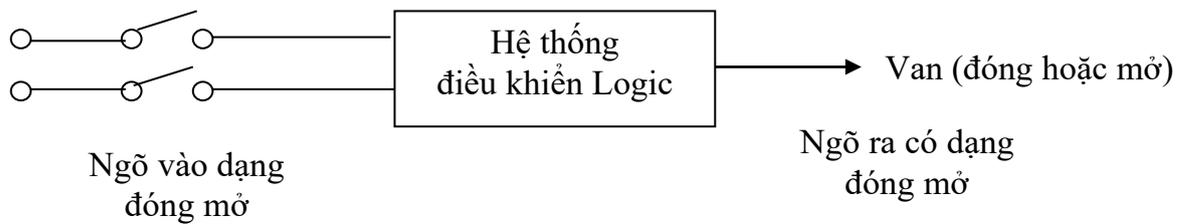
Hình 1.2: Loại điều khiển liên tục

Điều khiển nhị phân hay điều khiển on/off được dùng nhiều trong công nghiệp, vì máy móc và thiết bị là tập hợp nhiều bộ phận, thông thường những bộ phận này chỉ có hai trạng thái đóng hoặc mở được điều khiển bằng một số một số hoạt động đơn giản hay theo các bước trình tự.

Tín hiệu vào từ các cảm biến qua bộ chuyển đổi thành tín hiệu nhị phân và xuất ra tín hiệu điều khiển cơ cấu tác động hoạt động đóng mở. Có nhiều trường hợp mà các tín hiệu ngõ vào thường có dạng rời rạc, chẳng hạn như tín hiệu từ công tắc, chuỗi bit dữ liệu nhập từ bàn phím... trong trường hợp như vậy ta dùng kỹ thuật điều khiển nhị phân.

Các bộ điều khiển nhị phân gồm các mạch rơ-le, các hệ thống điều khiển điện, dầu, khí nén, máy tính, PLC.

Mỗi loại điều khiển có lĩnh vực ứng dụng riêng và phát huy hiệu quả trong phạm vi của nó. Trong công nghiệp ta sẽ gặp cả hai dạng này, nhưng *điều khiển nhị phân* về sự phức tạp thường bằng hoặc vượt trội so với *điều khiển tương tự*.



Hình 1.3: Loại điều khiển nhị phân

Phương pháp điều khiển

Điều khiển vòng hở thiết lập một hệ thống hoạt động để điều chỉnh trực tiếp hoạt động của ngõ ra. Hệ thống không có thông tin phản hồi đến bộ điều khiển để xác định hoặc hiệu chỉnh tín hiệu ra.

Điều khiển kích tiếp thiết lập hệ thống điều khiển trực tiếp hoạt động của ngõ ra có sự bù trừ từ hoạt động giám sát nhiều.

Điều khiển vòng kín là phương pháp điều khiển làm mất tác hại của nhiễu bằng cách đo ảnh hưởng của chúng trên tín hiệu hay trên sản phẩm để từ đó tính toán tác các tác động cần thiết để hiệu chỉnh làm mất tác dụng của nhiễu và duy trì tín hiệu hay sản phẩm đầu ra được ổn định. Tín hiệu phản hồi lấy từ ngõ ra được đưa về để so sánh với tín hiệu ngõ vào (so sánh giá trị mong muốn với giá trị thực tế). Sai biệt giữa chúng được đưa vào khối điều khiển để hiệu chỉnh tín hiệu ra đạt tới giá trị mong muốn.

1.2 ĐIỀU KHIỂN NÓI CỨNG VÀ ĐIỀU KHIỂN LẬP TRÌNH ĐƯỢC

Trong lĩnh vực điều khiển, người ta phân biệt hai phương pháp điều khiển là: phương pháp điều khiển nói cứng và phương pháp điều khiển lập trình được.

1.2.1 Phương pháp điều khiển nói cứng (Hard-wired control)

Điều khiển nói cứng được chia làm hai loại; Điều khiển nói cứng có tiếp điểm và điều khiển nói cứng không tiếp điểm

Điều khiển nói cứng có tiếp điểm dùng các khí cụ điện từ như rơ le, công tắc tơ, kết hợp với các bộ cảm biến, đèn, công tắc... Các khí cụ điện này được nối với nhau theo một mạch điện cụ thể để thực hiện một yêu cầu công nghệ nhất định.

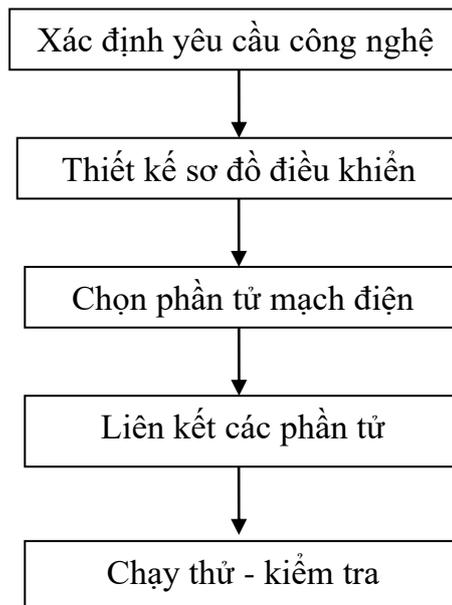
Điều khiển nói cứng không tiếp điểm dùng các cổng logic, mạch tuần tự (gọi chung là IC số) kết hợp với các bộ cảm biến, đèn, công tắc... Các IC số này cũng được nối lại với nhau theo một sơ đồ Logic cụ thể để thực hiện một yêu cầu công nghệ nhất định. Các mạch điều khiển nói cứng sử dụng các linh kiện bán dẫn công suất như: SCR, Triac để thay thế công tắc tơ trong các mạch động lực.

Trong các hệ thống điều khiển nói cứng, các khí cụ điện hay linh kiện được nối với nhau vĩnh viễn. Khi muốn thay đổi lại chức năng điều khiển thì phải nối dây lại toàn bộ mạch điện. Vậy với các hệ thống phức tạp thì không hiệu quả và rất tốn kém.

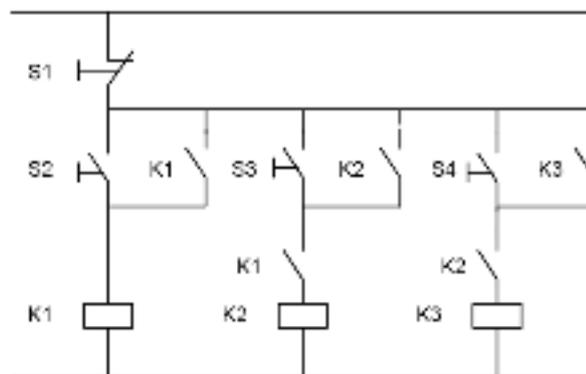
Phương pháp điều khiển nói cứng được thực hiện theo các bước hình 1.4.

Thí dụ: Thực hiện sơ đồ điều khiển ba động cơ chạy tuần tự. Hệ thống điều khiển dùng khí cụ điện có sơ đồ như hình 1.5.

Khi thay đổi mạch điều khiển cho các động cơ chạy độc lập thì phải nối lại mạch bỏ khoá K1 ở nhánh nối tiếp cuộn dây khởi động từ K2.



Hình 1.4: Các bước thiết lập hệ thống điều khiển nối cứng



Hình 1.5: Sơ đồ điều khiển 3 động cơ dùng khí cụ điện

1.2.2 Phương pháp điều khiển lập trình được

Trong hệ thống điều khiển lập trình được PLC (Programmable Logic Controller), cấu trúc của bộ điều khiển và cách nối dây độc lập với chương trình. Chương trình điều khiển được ghi trực tiếp vào bộ nhớ của bộ điều khiển nhờ sự trợ giúp của bộ lập trình cầm tay hay một máy vi tính.

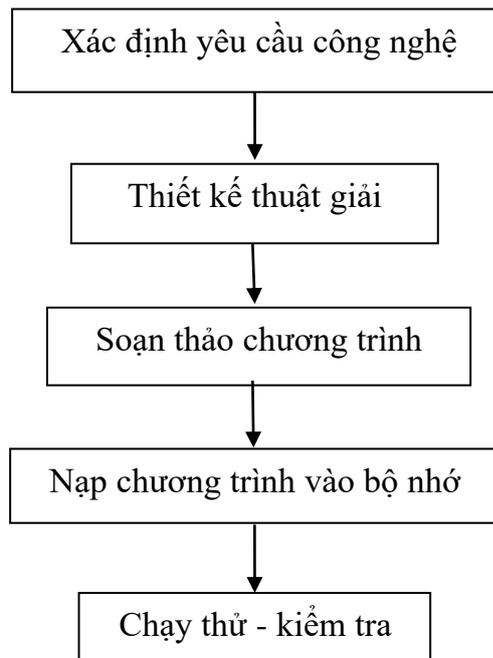
Để thay đổi chương trình điều khiển, chỉ cần thay đổi nội dung bộ nhớ của bộ điều khiển, phần nối dây bên ngoài không bị ảnh hưởng.

PLC tạo ra một khả năng điều khiển thiết bị dễ dàng và linh hoạt dựa trên việc lập trình bằng các lệnh logic cơ bản. Ngoài ra PLC còn có thể thực hiện các tác vụ khác như định thời, đếm, v.v... làm tăng khả năng điều khiển cho những hoạt động phức tạp.

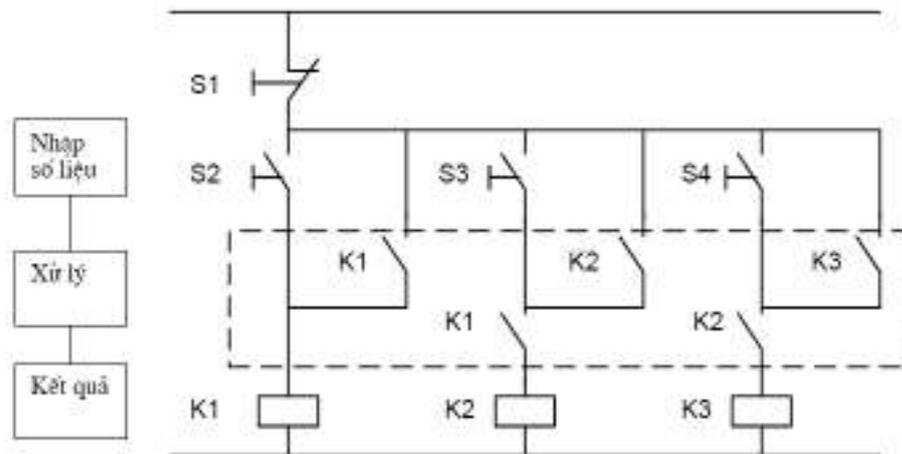
Phương pháp điều khiển lập trình được thực hiện theo các bước hình 1.6.

Thí dụ: Hình 1.7 thực hiện sơ đồ điều khiển ba động cơ chạy tuần tự. Hệ thống điều khiển dùng thiết bị điều khiển lập trình được có thể thay thế như sau:

- Tín hiệu vào: S1, S2, S3, S4 vẫn giữ nguyên.
- Tín hiệu ra: K1, K2, K3 là các khởi động từ vẫn giữ nguyên.
- Phần tử xử lý: được thay thế bằng PLC

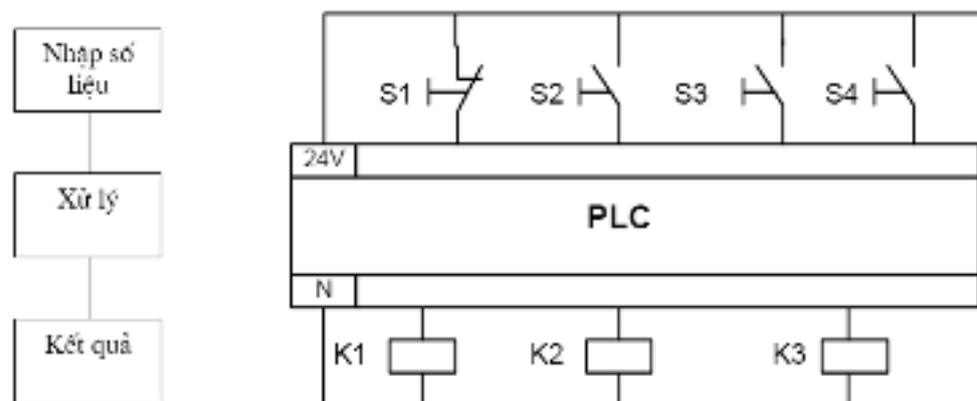


Hình 1.6: Các bước thiết lập hệ thống điều khiển lập trình



Hình 1.7: Sơ đồ thay đổi điều khiển dùng PLC

Khi thực hiện bằng chương trình điều khiển có nhớ PLC, ta chỉ cần thực hiện nối mạch theo sơ đồ sau:



Hình 1.8: Sơ đồ nối dây thực hiện bằng PLC

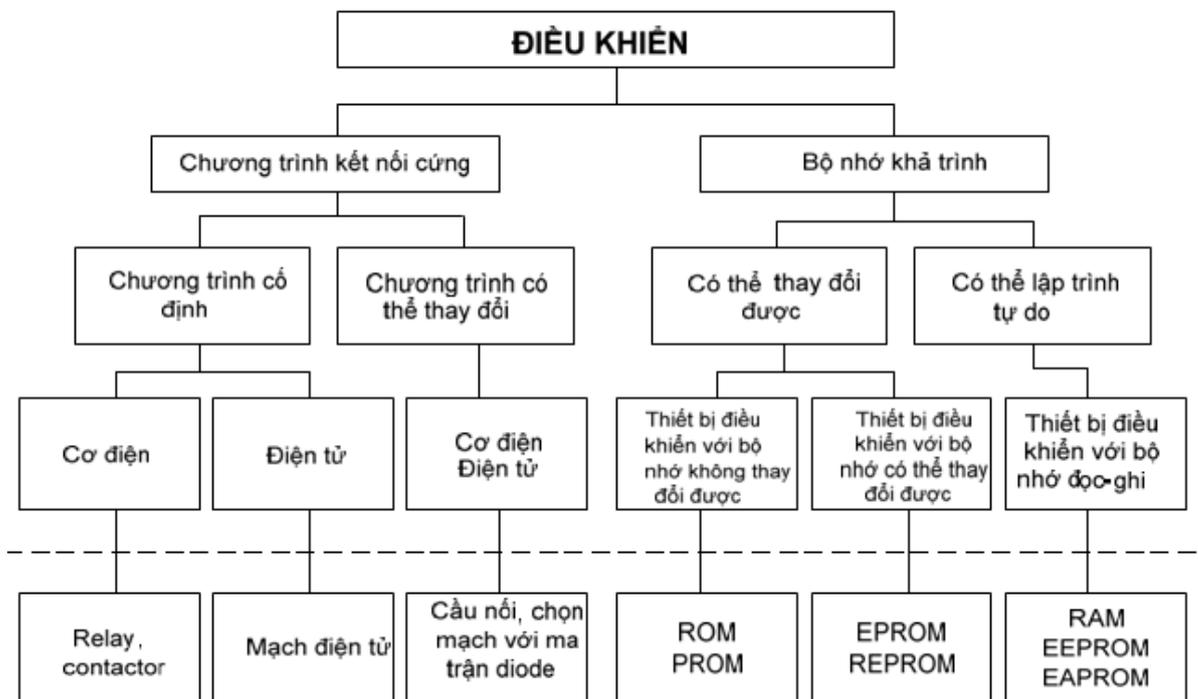
Với sơ đồ này ta chỉ việc viết lại chương trình khi cần thay đổi chế độ hoạt động độc lập của từng động cơ, rồi nạp chương trình vào lại PLC.

PLC tương tự như máy tính truyền thống và chúng có các đặc điểm thích hợp cho mục đích điều khiển trong công nghiệp:

- Khả năng kháng nhiễu tốt;
- Cấu trúc dạng mô-đun cho phép dễ dàng thay thế, tăng khả năng mở rộng chức năng điều khiển;
- Chuẩn hoá điện áp ngõ vào và ra;
- Ngôn ngữ lập trình dễ hiểu và dễ sử dụng;
- Thay đổi chương trình dễ dàng.

1.3 SO SÁNH PLC VỚI CÁC HÌNH THỨC ĐIỀU KHIỂN KHÁC

Trong hệ thống điều khiển có nhiều loại điều khiển khác nhau được trình bày trong hình 1.9:



Hình 1.9: Sơ đồ các loại điều khiển

1.3.1 Hệ thống điều khiển dùng Rơ-le

Rơ-le là một công tắc điện chịu được dòng điện cao, được tác động gián tiếp bởi dòng điện điều khiển có cường độ thấp. Rơ-le có cấu tạo về cơ để đóng / mở tiếp điểm nên bị hạn chế về tốc độ tác động, tuổi thọ và độ tin cậy. Kích thước rơ-le công kênh chiếm nhiều không gian trong tủ điện.

Đặc điểm chung của hệ thống này là dễ thiết kế và lắp đặt; toàn bộ các công việc điều khiển được thực hiện thông qua phối hợp đơn giản trình tự hoạt động của các rơ le.

1.3.2 Hệ thống điều khiển dùng mạch điện tử (Transistor)

Transistor được dùng như công tắc đóng mở. Với kích thước nhỏ và không có cấu tạo cơ khí nên tốc độ chuyển mạch nhanh, giá thành thấp nhưng khả năng điều khiển công suất không cao.

Vi mạch là mạch tích hợp nhiều linh kiện bán dẫn để thực hiện một số chức năng nào đó. Nhiều hệ thống phức tạp và tinh vi được thực hiện bằng cách kết nối các vi mạch thích hợp với nhau.

1.3.3 Hệ thống điều khiển dùng IC số

Khi thiết kế một hệ thống dùng vi mạch và cổng Logic, người thiết kế phải vẽ sơ đồ mạch với các ký hiệu Logic thể hiện các ngõ vào và ra nhị phân. Hệ thống xử lý thông qua các phép tính đại số Boole. Quá trình xử lý có thể là tuần tự hay tổ hợp những ngõ vào để cho kết quả ngõ ra. Để tăng khả năng xử lý hệ thống dùng thêm bộ nhớ.

Với các vi mạch số cho phép thiết kế và lắp đặt các hệ thống có khả năng điều khiển logic với tốc độ cao trong các trường hợp điều khiển trình tự và điều khiển liên tục. tuy nhiên, nếu có yêu cầu thay đổi chức năng điều khiển thì do bản chất không linh hoạt về kết nối phần cứng, nên phải mất nhiều thời gian thiết kế và lắp đặt lại.

1.3.4 Hệ thống điều khiển dùng máy tính

Máy tính là một bộ máy điện tử xử lý thông tin ở dạng nhị phân. Máy tính được ứng dụng trong điều khiển từ giữa thập niên 1950, nhưng đến cuối thập niên 1970, việc ứng dụng máy tính trong điều khiển mới có hiệu quả kinh tế và hấp dẫn các nhà thiết kế hệ thống điều khiển. Bên cạnh đó, ngôn ngữ lập trình cho máy tính ngày càng phong phú, thuận lợi cho các ứng dụng điều khiển.

Hiện nay máy tính cấu hình mạnh với giá rẻ cho phép ứng dụng vào những công việc điều khiển hệ thống theo trình tự và điều khiển liên tục với những thủ tục truyền thông đơn giản, dễ áp dụng.

Tóm lại: khi thiết kế một hệ thống điều khiển, người ta dựa vào một số chỉ tiêu so sánh để lựa chọn hệ thống thích hợp nhất.

Bảng 1.1 So sánh các hệ thống điều khiển

Chỉ tiêu so sánh	Rơ-le	Mạch số	Máy tính	PLC
Giá thành	Khá thấp	Thấp	Cao	Thấp
Kích thước	Lớn	Rất gọn	Khá gọn	Rất gọn
Tốc độ điều khiển	Chậm	Rất nhanh	Khá nhanh	Nhanh
Khả năng chống nhiễu	Rất tốt	Tốt	Khá tốt	Tốt
Lắp đặt	Mất thời gian thiết kế, lắp đặt	Mất thời gian thiết kế	Mất thời gian lập trình	Lập trình, lắp đặt đơn giản
Khả năng ĐK phức tạp	Không	Có	Có	Có
Khả năng thay đổi	Rất khó	Khó	Khá đơn giản	Rất đơn giản
Công tác bảo trì	Kém	Kém	Kém	Tốt

1.4 CÁC ỨNG DỤNG CỦA PLC TRONG THỰC TẾ

Về cơ bản chức năng của một bộ điều khiển lập trình được cũng giống như chức năng của bộ điều khiển thiết kế trên cơ sở các rơ-le hoặc các thành phần điện tử:

- Thu nhận các tín hiệu đầu vào và phản hồi
- Liên kết ghép nối lại và đóng mở mạch phù hợp với chương trình
- Tính toán và soạn thảo các lệnh điều khiển trên các thông tin thu nhận được.
- Phân phát các lệnh điều khiển đó đến các địa chỉ thích hợp.

PLC được ứng dụng nhiều trong công nghiệp, thương mại:

- Hệ thống điều khiển đèn giao thông.

- Điều khiển rèm cửa
- Điều khiển nhiệt độ và thông gió
- Hệ thống bơm nước tự động
- Điều khiển Gara ô tô
- Hệ thống cảnh báo và chuông báo động
- Hệ thống tưới nước tự động
- Chiếu sáng trong các phòng, cầu thang, cửa hàng

Đến các ứng dụng trong công nghiệp:

- Thang máy, máy nâng
- Hệ thống hoà đồng bộ
- Lò hơi
- Hệ thống trạm cân xe tải
- Hệ thống băng tải
- Hệ thống trộn nguyên liệu
- Hệ thống đo mức, áp suất
- Hệ thống quản lý năng lượng
- Hệ thống chuyển tải
- Máy hàn, máy cưa, máy uốn và máy cắt
- Hệ thống quan sát, điều khiển bãi giữ xe
- Hệ thống điều khiển công nhà máy
- Hệ thống điều hoà nhiệt độ, làm mát và điều hoà không khí...

CÂU HỎI ÔN TẬP

1. Trình bày các khối trong hệ thống điều khiển
2. Nêu các bước thiết kế một hệ thống điều khiển nối cứng, hệ thống điều khiển lập trình.
3. So sánh hệ thống điều khiển dùng PLC với các hệ thống điều khiển khác.

Bài 2:

CẤU TRÚC VÀ PHƯƠNG THỨC HOẠT ĐỘNG CỦA MỘT PLC

GIỚI THIỆU

PLC viết tắt của Programmable Logic Controller, là thiết bị điều khiển lập trình được (khả trình) cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua một ngôn ngữ lập trình. Người sử dụng có thể lập trình để thực hiện một loạt trình tự các sự kiện. Các sự kiện này được kích hoạt bởi tác nhân kích thích (ngõ vào) tác động vào PLC hoặc qua các hoạt động có trễ như thời gian định thì hay các sự kiện được đếm. PLC dùng để thay thế các mạch relay (rơ le) trong thực tế. PLC hoạt động theo phương thức quét các trạng thái trên đầu ra và đầu vào. Khi có sự thay đổi ở đầu vào thì đầu ra sẽ thay đổi theo. Ngôn ngữ lập trình của PLC có thể là Ladder hay State Logic. Hiện nay có nhiều hãng sản xuất ra PLC như Siemens, Allen-Bradley, Mitsubishi Electric, General Electric, Omron, Honeywell, LS, Schneider, Fatek...

MỤC TIÊU CỦA BÀI

+ Về kiến thức:

- Phát biểu được cấu trúc của một thiết bị điều khiển lập trình PLC;
- Trình bày được cấu trúc bộ nhớ PLC.

+ Về kỹ năng:

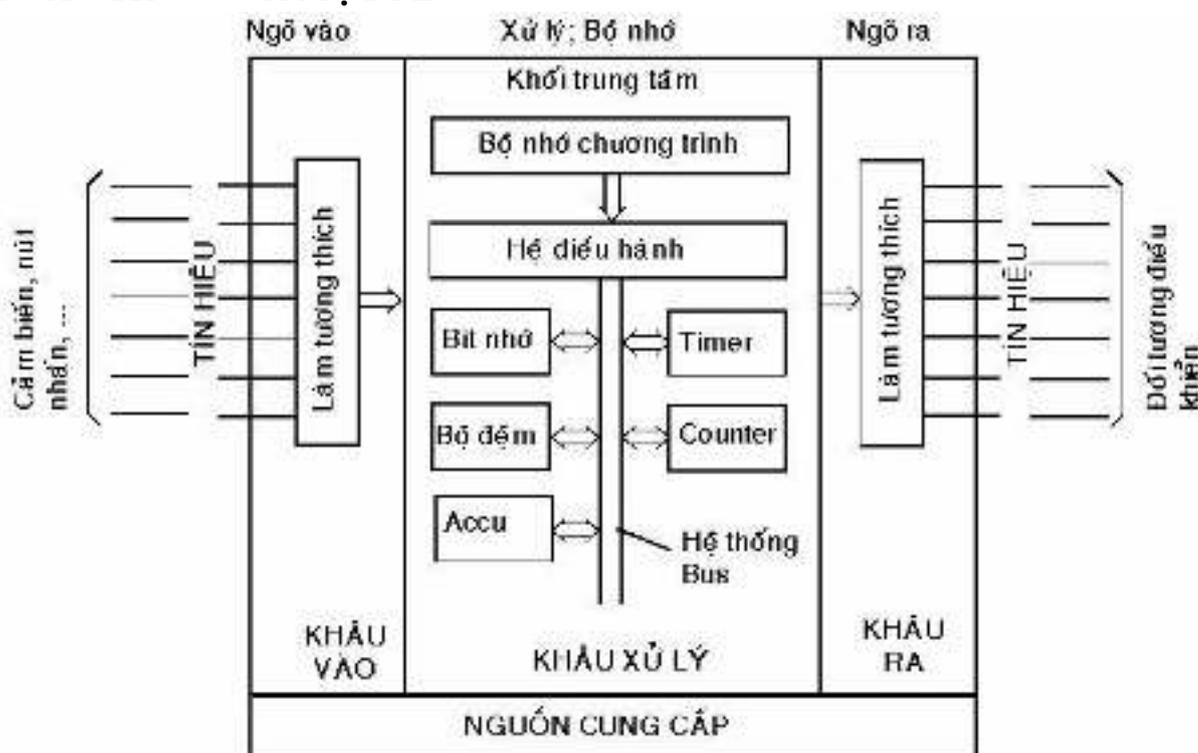
- Thực hiện xử lý chương trình đúng theo nội dung đã học.

+ Về thái độ:

- Đảm bảo an toàn và vệ sinh công nghiệp

NỘI DUNG CỦA BÀI

2.1 CẤU TRÚC CỦA MỘT PLC



Hình 2.1: 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.

2.1.1 Đơn vị 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 khống 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.

2.1.2 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.

2.1.3 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

2.2 THIẾT BỊ ĐIỀU KHIỂN LẬP TRÌNH S7-200

2.2.1 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. S7-200 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. Thành phần cơ bản của S7-200 là khối vi xử lý trung tâm bao gồm hai chủng loại CPU 21X và CPU 22X.

CPU 21X ngày nay không còn sử dụng nữa. Tuy nhiên thay thế cho nó bởi CPU 22X và vẫn còn sử dụng rất nhiều trong các trường học và trong sản xuất. Tiêu biểu cho loại này là CPU 224.

CPU 224 có các đặc tính như sau:

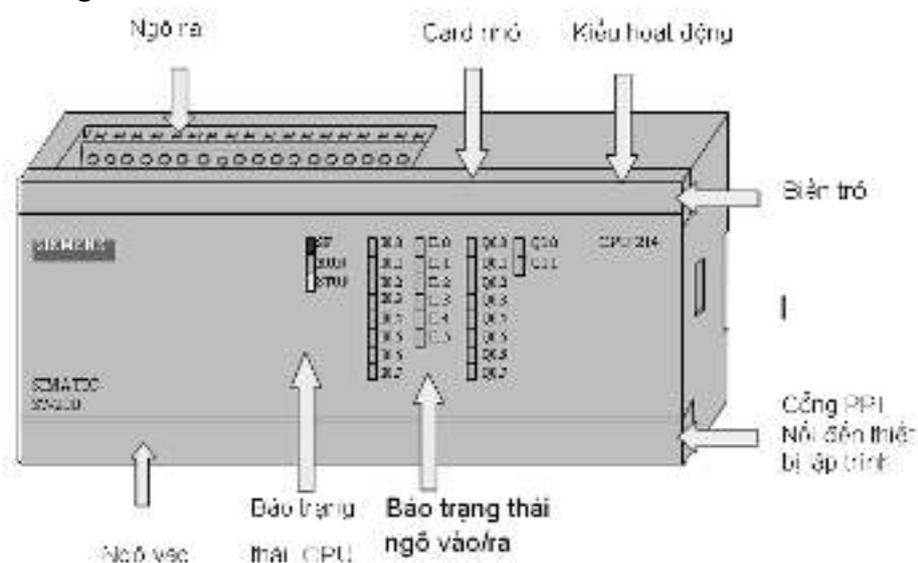
- Bộ nhớ chương trình (chứa trong EEPROM): 4096 Byte.
- Bộ nhớ dữ liệu: 4096 Byte (trong đó 512 Byte chứa trong EEPROM)
- Số lượng ngõ vào số: 14
- Số lượng ngõ ra số: 10

- Số module mở rộng: 7 gồm cả module analog
- Số lượng vào/ra số cực đại: 64
- Số lượng Timer: 256 Timer chia làm 3 loại theo độ phân giải khác nhau: 8 Timer 1ms, 32 Timer 10ms và 116 Timer có độ phân giải 100ms.
- Số lượng Counter: 256 bộ đếm chia làm hai loại: 192 Counter Up và 64 Counter Up/Down.
- Bit memory (Vùng nhớ M): 256 bit
- Special memory (SM – Bit nhớ đặc biệt): 688 bit dùng để thông báo trạng thái và đặt chế độ làm việc.
- Có phép tính số học.
- High - Speed Counters (HSC - Bộ đếm tốc độ cao): 4 Counter 2 KHz và 2 Counter 7 KHz.
- Ngõ vào analog tích hợp sẵn (biến trở): 2.
- Các chế độ ngắt và xử lý ngắt gồm: ngắt truyền thông, ngắt theo sườn lên hoặc xuống, ngắt thời gian, ngắt của bộ đếm tốc độ cao và ngắt truyền xung.
- Toàn bộ vùng nhớ không bị mất dữ liệu trong khoảng thời gian 190 giờ khi PLC bị mất nguồn nuôi.

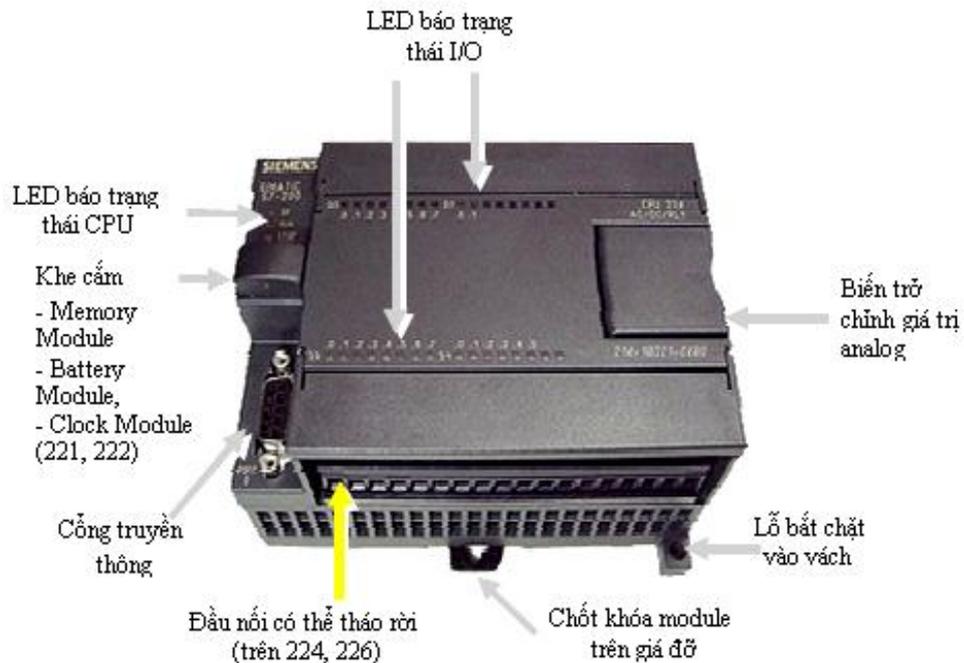
CPU 22X với nhiều tính năng vượt trội và hiện đang còn được sử dụng rất nhiều. Tiêu biểu cho loại này là các dòng CPU222, CPU224, CPU224XP, CPU226. Về hình thức bên ngoài chúng có sự khác nhau và nhận biết được nhờ số đầu vào/ra và nguồn cung cấp.

Hiện nay, PLC S7-200 đã ngừng sản xuất và được Siemens thay thế bởi PLC S7-1200 có thiết kế nhỏ gọn, chi phí thấp, và một tập lệnh mạnh giúp những giải pháp hoàn hảo hơn cho ứng dụng tự động hóa trong công nghiệp và dân dụng. Các dòng chính của PLC S7-1200: CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C và CPU 1217C.

Tuy nhiên, với khả năng tiếp nhận và kiến thức của học sinh trung cấp thì PLC S7-200 là lựa chọn để tiếp thu kiến thức cơ bản về điều khiển lập trình trong công nghiệp và dân dụng.



Hình 2.2: Bộ điều khiển lập trình S7-200 CPU214



Hình 2.3: Bộ điều khiển lập trình S7-200 CPU224

2.2.2 Các đèn báo trên S7-200 CPU 224

- 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)

2.2.3 Chọn chế độ làm việc cho PLC

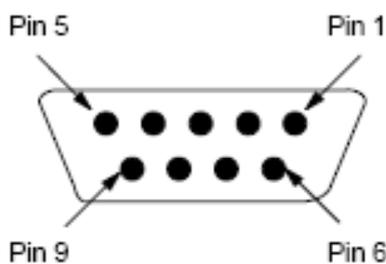
Công tắc chọn chế độ làm việc nằm ở phía trên, có ba vị trí cho phép chọn các chế độ làm việc khác nhau của PLC:

- RUN: Cho phép PLC thực hiện chương trình trong bộ nhớ. PLC S7-200 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: Cường 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.

2.2.4 Cổng truyền thông trên S7-200

S7-200 sử dụng cổng truyền thông nối tiếp RS485 với phích nối 9 chân để phục vụ cho việc ghép nối với thiết bị lập trình hoặc với các trạm PLC khác. Tốc độ truyền cho máy lập trình kiểu PPI là 9600 baud.

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



Hình 2.4: Cổng nối tiếp RS485

2.3 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}...

Bảng 2.1 Địa chỉ vào/ra và modul mở rộng

CPU224	Modul 0 4 vào/4 ra	Modul 1 8 vào	Modul 2 3vào/1ra Analog	Modul 3 8 ra	Modul 4 3 vào/1ra Analog
I0.0 Q0.0	I2.0	I3.0	AIW 0	Q3.0	AIW8
I0.1 Q0.1	I2.1	I3.1	AIW 2	Q3.1	AIW12
I0.2 Q0.2	I2.2	I3.2	AIW 4	Q3.2	AQW 4
I0.3 Q0.3	I2.3	I3.3		Q3.3	
I0.4 Q0.4	Q2.0	I3.4		Q3.4	
I0.5 Q0.5	Q2.1	I3.5	AQW 0	Q3.5	
I0.6 Q0.6	Q2.2	I3.6		Q3.6	
I0.7 Q0.7	Q2.3	I3.7		Q3.7	
I1.0 Q1.0					
I1.1 Q1.1					
I1.2					
I1.3					
I1.4					
I1.5					

- 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 S7-200 đặt theo thứ tự hệ bát phân. Trong đó CPU 222 có sẵn 8 vào và 6 ra, CPU224 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.

2.4. CẤU TRÚC BỘ NHỚ CỦA S7-200

Bộ nhớ của S7-200 được chia thành ba vùng nhớ chính như hình 2.5:

+ **Vùng chứa chương trình**: 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 cò): 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).

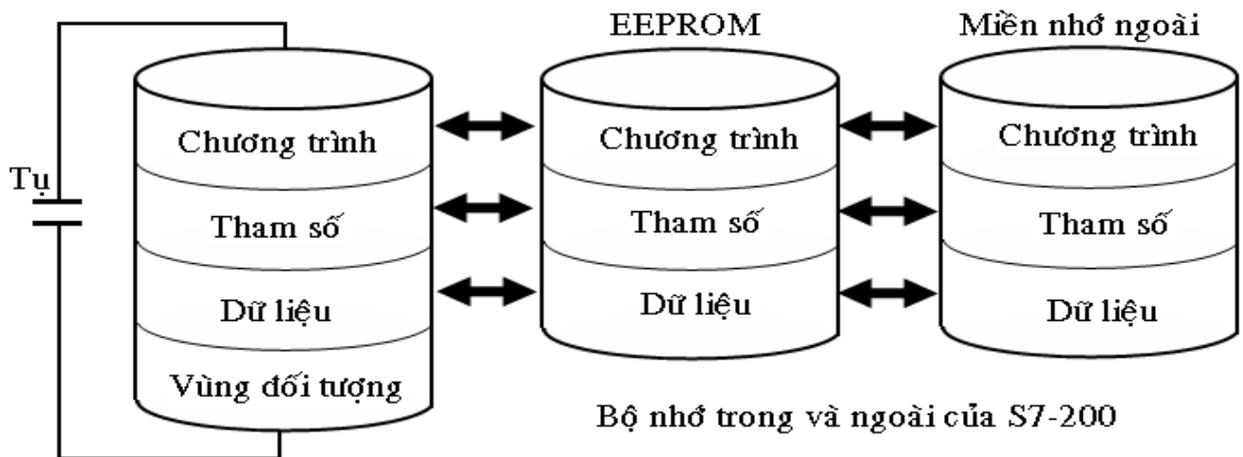
- T (Timer): Miền nhớ phục vụ bộ thời gian (Timer) bao gồm việc lưu trữ giá trị thời gian đặt trước (PV-Preset Value), giá trị đếm thời gian tức thời (CV –Current Value) cũng như giá trị Logic đầu ra của bộ thời gian.

- 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).

- 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 nhấp 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).

+ *Vùng chứa các khối dữ liệu:*



Hình 2.5: Vùng nhớ dữ liệu

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

Vùng nhớ		CPU 221	CPU 222	CPU 224	CPU 226
Vùng dữ liệu	V	V0.0÷V2047.7	V0.0÷V2047.7	V0.0÷V5119.7	V0.0÷V5119.7
	I	I0.0÷I15.7	I0.0÷I15.7	I0.0÷I15.7	I0.0÷I15.7
	Q	Q0.0÷Q15.7	Q0.0÷15.7	Q0.0÷Q15.7	Q0.0÷Q15.7
	M	M0.0÷M31.7	M0.0÷M31.7	M0.0÷M31.7	M0.0÷M31.7
	SM	SM0.0÷SM179.7	SM0.0÷SM179.7	SM0.0÷SM179.7	SM0.0÷SM179.7
	S	S0.0÷S31.7	S0.0÷S31.7	S0.0÷S31.7	S0.0÷S31.7
	L	L0.0÷L63.7	L0.0÷L63.7	L0.0÷63.7	L0.0÷L63.7
Vùng đối tượng	Timer	T0÷T255	T0÷T255	T0÷T255	T0÷T255
	Counter	C0÷C255	C0÷C255	C0÷C255	C0÷C255
	AI	none	AIW0÷AIW30	AIW0÷AIW62	AIW0÷AIW62
	AO	none	AQW0÷AQW30	AQW0÷AQW62	AQW0÷AQW62
	ACC	AC0÷AC3	AC0÷AC3	AC0÷AC3	AC0÷AC3
	HSC	HSC0,HSC3, HSC4, HSC5	HSC0,HSC3, HSC4, HSC5	HSC0÷HSC5	HSC0÷HSC5

2.5 XỬ LÝ CHƯƠNG TRÌNH

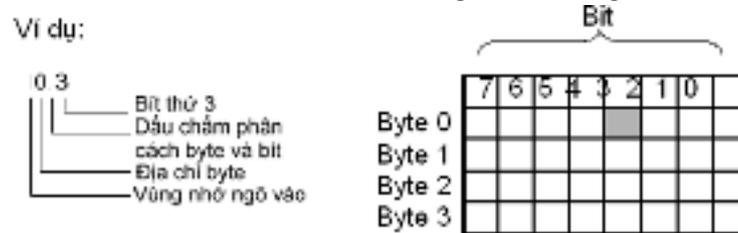
2.5.1. Truy xuất dữ liệu

Việc xử lý chương trình trong S7-200 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.

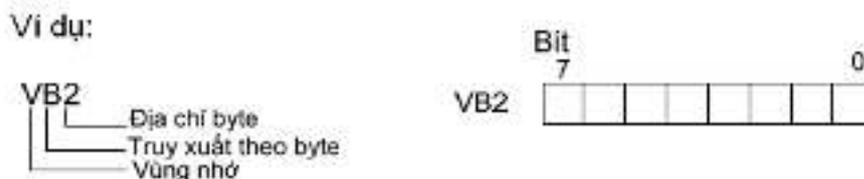
Trong hình 2.6 là bản đồ vùng nhớ của bộ đệm dữ liệu ngõ vào I(Process Image Input). Bản đồ của các vùng nhớ khác cũng có cấu trúc tương tự như vậy. Bit thấp nhất là bit 0 nằm bên phải và bit cao nhất là bit 7 nằm bên trái. Do đó chúng ta hoàn toàn có thể khai báo tương tự như ví dụ trên, chẳng hạn như: Q1.0, V5.2, M0.1...Dung lượng của các vùng nhớ phụ thuộc vào loại CPU mà chúng ta sử dụng.



Hình 2.6: 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.

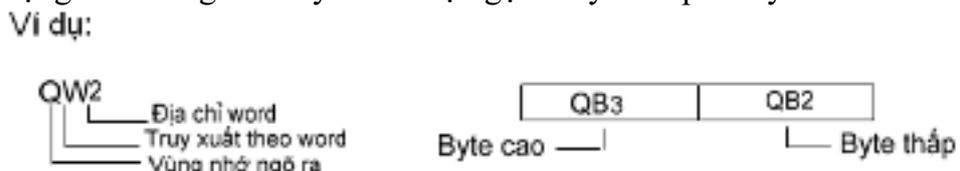


Hình 2.7: Truy xuất vùng nhớ byte

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.



Hình 2.8: Truy xuất vùng nhớ word

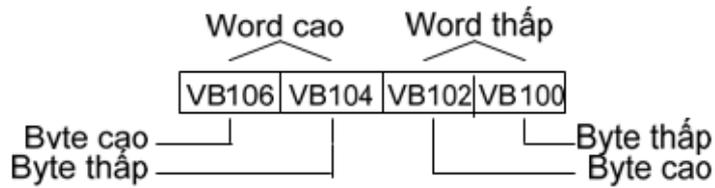
Đối với tín hiệu tương tự (Analog) thì chúng ta chỉ có một dạng truy xuất duy nhất là truy xuất theo word. Điều này là do mỗi tín hiệu tương tự sẽ ứng với một giá trị số nguyên 16 bit. Ví dụ: AIW0, AIW2, AQW0....

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 2.9: 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....

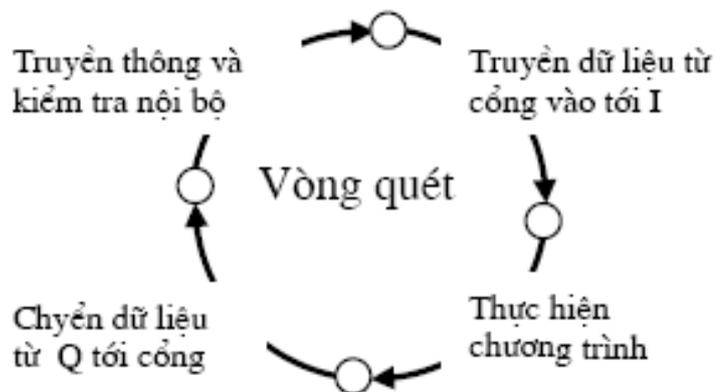
- Nhận tín hiệu từ các cảm biến tạo ra tín hiệu analog như cảm biến nhiệt độ, áp suất, độ ẩm ... thì sử dụng địa chỉ word, ví dụ: AIW0, AIW2, AIW4....

- 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...

2.5.2 Xử lý 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 2.10: 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.

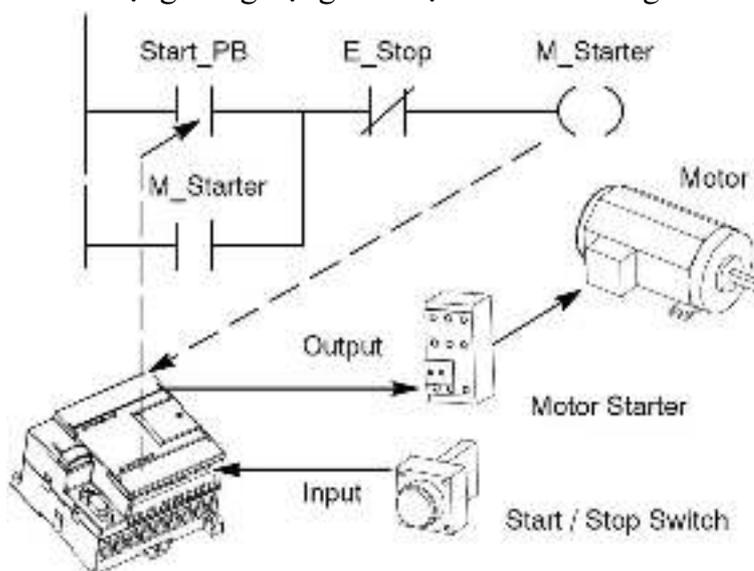
Thường việc thực hiện một vòng quét xảy ra với một thời gian rất ngắn, một vòng quét đơn có thời gian thực hiện từ 1ms tới 100ms. Việc thực hiện một chu kỳ quét dài hay ngắn còn phụ thuộc vào độ dài của chương trình và cả mức độ giao tiếp giữa PLC với các thiết bị ngoại vi. Vi xử lý có thể đọc được tín 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 S7-200 rất đơn giản và thực hiện theo trình tự sau:

- Đọc trạng thái các ngõ vào.
- S7-200 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 2.11 là một sơ đồ đơn giản chỉ mối quan hệ giữa sơ đồ điện và PLC S7-200. Các nút nhấn khởi động/dừng động cơ được kết nối với ngõ vào.



Hình 2.11: 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.

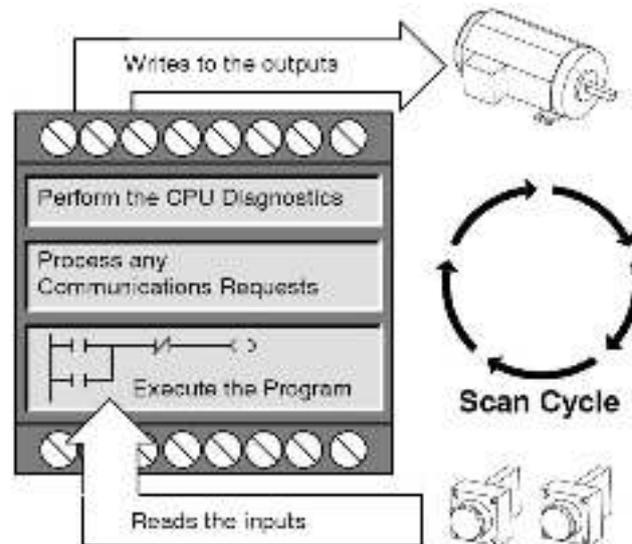
Hình 2.12 là ví dụ một chu kỳ quét. S7-200 thực hiện các nhiệm vụ sau trong một chu kỳ quét.

***Đọc ngõ vào:** S7-200 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:* S7-200 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, S7-200 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, S7-200 đọc giá trị ngõ vào analog từ module mở rộng mỗi lần chương trình truy xuất ngõ vào analog.



Hình 2.12: Chu kỳ quét S7-200

* **Thực hiện theo logic điều khiển trong chương trình:** S7-200 thực hiện các lệnh trong chương trình và lưu giá trị vào vùng nhớ.

Khi thực hiện chu kỳ quét, S7-200 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: S7-200 thi hành bất kỳ nhiệm vụ được yêu cầu cho truyền thông.

Trong giai đoạn xử lý thông tin của chu kỳ quét, S7-200 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: S7-200 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, S7-200 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, S7-200 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 S7-200 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.

2.5.3 Cấu trúc chương trình

Chương trình trong S7_200 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_200 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.

2.5.4 Phương pháp lập trình PLC S7-200

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

Cách lập trình cho PLC S7-200 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.

Đối với thiết bị điều khiển lập trình PLC S7 - 200, ta không thể lập trình trực tiếp ngay trên nó được mà phải lập trình gián tiếp bằng cách sử dụng phần mềm STEP 7 – Micro/WIN.

Phần mềm này đều có thể cài đặt được trên các máy lập trình họ PG7xx hoặc các máy tính cá nhân. Công việc lập trình là ta sử dụng máy tính để tiến hành lắp ghép các lệnh cơ bản lại với nhau nhằm thỏa mãn những yêu cầu đề ra của quy trình công nghệ rồi sau đó mới chuyển vào PLC để điều khiển. Các lệnh này thường ở 3 dạng LAD, STL và FBD.

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): $\text{---}(\text{---})\text{---}$ là biểu tượng mô tả rơle, được mắc theo chiều dòng điện cung cấp cho rơle.
- 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 (thường không được thể hiện khi dùng chương trình STEP 7 MICRO / DOS hoặc STEP 7 – MICRO/WIN). 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.

Ngăn xếp và tên của từng bit trong ngăn xếp được biểu diễn ở bảng bên.

S0	Bit đầu tiên
S1	Bit thứ hai
S2	Bit thứ ba
S3	Bit thứ tư
S4	Bit thứ năm
S5	Bit thứ sáu
S6	Bit thứ bảy
S7	Bit thứ tám
S8	Bit thứ chín

Hình 2.13: Ngăn xếp

- Nội dung thực hành: Khảo sát PLC S7-200 CPU 222 và PLC S7-200 CPU 224
 - Xác định CPU loại gì?
 - Xác định các chân nguồn cho PLC, nguồn ngõ vào, nguồn ngõ ra.
 - Nhận biết Switch Mode chọn chế độ làm việc và Led báo trạng thái.

- Nhận biết khe mở rộng.
- Nhận biết các biến trở hiệu chỉnh Analog.
- Nhận biết cổng giao tiếp máy tính.
- Nhận biết các đầu nối và led trạng thái ngõ vào ra.

CÂU HỎI ÔN TẬP

1. Trình bày cấu trúc của PLC.
2. Nêu trình tự thực hiện chương trình trong PLC.
3. Nêu các phương pháp lập trình cho PLC S7-200 của Siemens.

Bài 3:

KẾT NỐI DÂY GIỮA PLC VÀ THIẾT BỊ NGOẠI VI

GIỚI THIỆU

Việc kết nối giữa dây và PLC rất quan trọng. Nó quyết định đến việc PLC có thể giao tiếp 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

MỤC TIÊU CỦA BÀI:

+ Về kiến thức:

- Trình bày được các thiết bị ngoại vi, cơ cấu chấp hành.
- Trình bày cách kết nối giữa PLC và thiết bị ngoại vi đúng theo yêu cầu kỹ thuật của PLC S7-200 CPU 224

+ Về kỹ năng:

- Thực hiện cài đặt phần mềm đạt các yêu cầu kỹ thuật và sử dụng được phần mềm lập trình Step7Microwin V4.0 cho PLC S7-200;
- Kết nối giữa PLC và thiết bị ngoại vi đạt các yêu cầu kỹ thuật;
- Kiểm tra nối dây bằng phần mềm chính xác

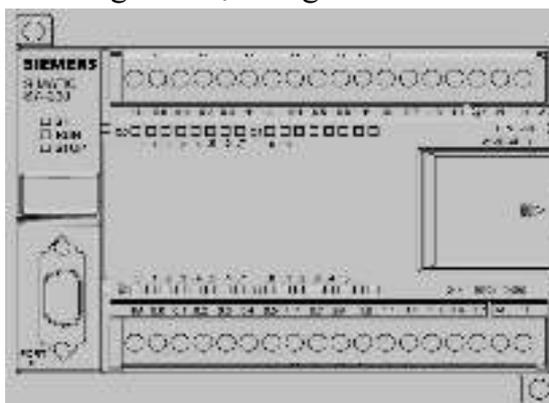
+ Về thái độ:

- Đảm bảo an toàn và vệ sinh công nghiệp.

NỘI DUNG BÀI

3.1 KẾT NỐI DÂY GIỮA PLC VÀ 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.



Hình 3.1: Bộ điều khiển lập trình S7-200 CPU 224

3.1.1 Giới thiệu CPU 224 và cách kết nối với thiết bị ngoại vi

Sơ đồ bề mặt của bộ điều khiển lập trình S7-200 CPU 224 được cho như hình 3.1. Để cho bộ điều khiển lập trình này hoạt động được thì người sử dụng phải kết nối PLC với nguồn cung cấp và các ngõ vào ra của nó với thiết bị ngoại vi. Muốn nạp chương trình vào CPU, người sử dụng phải soạn thảo chương trình bằng các thiết bị lập trình hoặc máy tính với phần mềm tương ứng cho loại PLC đang sử dụng và có thể nạp trực tiếp vào CPU hoặc copy chương trình vào card nhớ để cắm vào rãnh cắm card nhớ

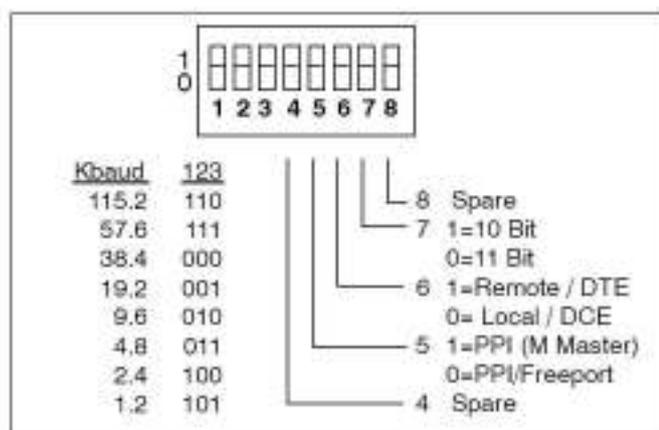
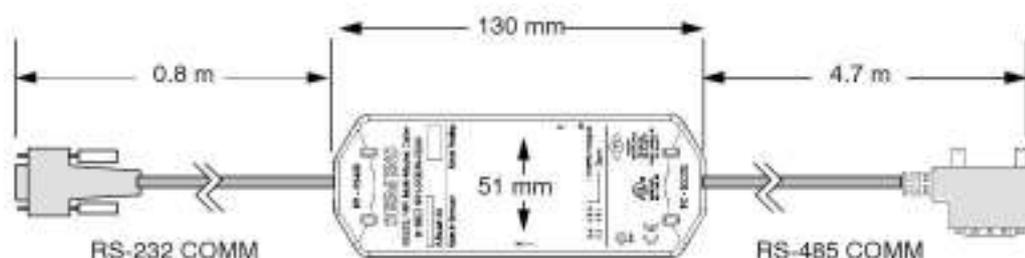
trên CPU của PLC. Thông thường khi lập trình cũng như khi kiểm tra hoạt động của PLC thì người lập trình thường kết nối trực tiếp thiết bị lập trình hoặc máy tính cá nhân với PLC. Như vậy, để hệ thống điều khiển khiển bằng PLC hoạt động cũng như lập trình cho nó, cần phải kết nối PLC với máy tính cũng như các ngõ vào ra với ngoại vi.

3.1.2 Kết nối với máy tính

Đối với các thiết bị lập trình của hãng Siemens có các cổng giao tiếp PPI thì có thể kết nối trực tiếp với PLC thông qua một sợi cáp. Tuy nhiên đối với máy tính cá nhân cần thiết phải có cáp chuyển đổi PC/PPI. Có 2 loại cáp chuyển đổi là cáp RS-232/PPI Multi-Master và cáp USB/PPI Multi-Master.

Cáp RS-232/PPI multi-master:

Hình dáng của cáp và công tắc chọn chế độ truyền được cho ở hình 3.2.



LED	Color	Description
Tx	Green	RS-232 transmit indicator
Rx	Green	RS-232 receive indicator
PPI	Green	RS-485 transmit indicator

Hình 3.2: Hình dáng cáp RS-232/PPI và các chuyển mạch trên cáp.

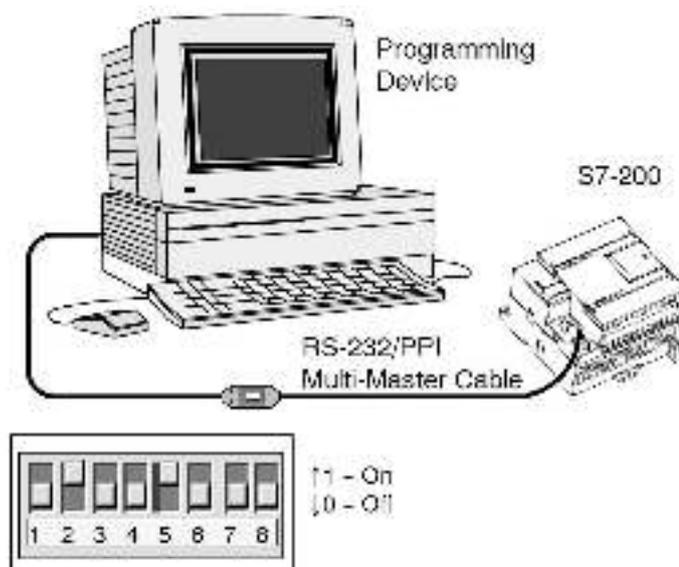
Tùy theo tốc độ truyền giữa máy tính và CPU mà các công tắc 1,2,3 được để ở vị trí thích hợp. Thông thường đối với CPU 22x thì tốc độ truyền thường đặt là 9,6 KBaud (tức công tắc 1,2,3 được đặt theo thứ tự là 010).

Tùy theo truyền thông là 10 Bit hay 11Bit mà công tắc 7 được đặt ở vị trí thích hợp. Khi kết nối bình thường với máy tính thì công tắc 7 chọn ở chế độ truyền thông 11 Bit (công tắc 7 đặt ở vị trí 0).

Công tắc 6 ở cáp RS-232/PPI Multi-Master được sử dụng để kết nối port truyền thông RS-232 của một modem với S7-200 CPU. Khi kết nối bình thường với máy tính

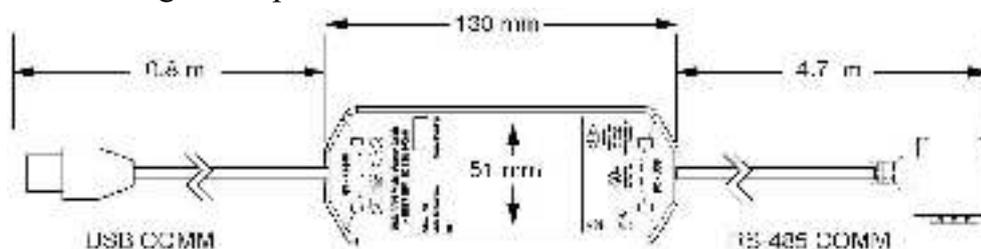
thì công tắc 6 được đặt ở vị trí data Communications Equipment (DCE) (công tắc 6 ở vị trí 0). Khi kết nối cáp PC/PPI với một modem thì port RS-232 của cáp PC/PPI được đặt ở vị trí Data Terminal Equipment (DTE) (công tắc 6 ở vị trí 1). Công tắc 5 được sử dụng để đặt cáp RS-232/PPI Multi-Master thay thế cáp PC/PPI hoặc hoạt động ở chế độ Freeport thì đặt ở chế độ PPI/Freeport (công tắc 5 ở vị trí 0). Nếu kết nối bình thường là PPI (master) với phần mềm STEP 7 Micro/Win 3.2 SP4 hoặc cao hơn thì đặt ở chế độ PPI (công tắc 5 ở vị trí 1).

Sơ đồ nối cáp RS-232/PPI Multi-Master giữa máy tính và CPU S7-200 với tốc độ truyền 9,6 Kbaud được cho như hình 3.3.



Hình 3.3: Kết nối máy tính với CPU S7-200 RS-232/PPI Multi-Master
Cáp USB/PPI multi-master:

Hình dáng của cáp được cho ở hình 3.4.



LED	Color	Description
Tx	Green	USB transmit indicator
Rx	Green	USB receive indicator
PPI	Green	RS-485 transmit indicator

Hình 3.4: Hình dáng cáp USB/PPI.

Cách thức kết nối cáp USB/PPI Multi-Master cũng tương tự như cáp RS-232/PPI Multi-Master. Để sử dụng cáp này, phần mềm cần phải là STEP 7- Micro/WIN 3.2 SP4 (hoặc cao hơn). Cáp chỉ có thể được sử dụng với loại CPU22x hoặc sau này. Cáp USB không được hỗ trợ truyền thông Freeport và download cấu hình màn hình TP070 từ phần mềm TP Designer.

3.1.3 Nối nguồn cung cấp cho CPU

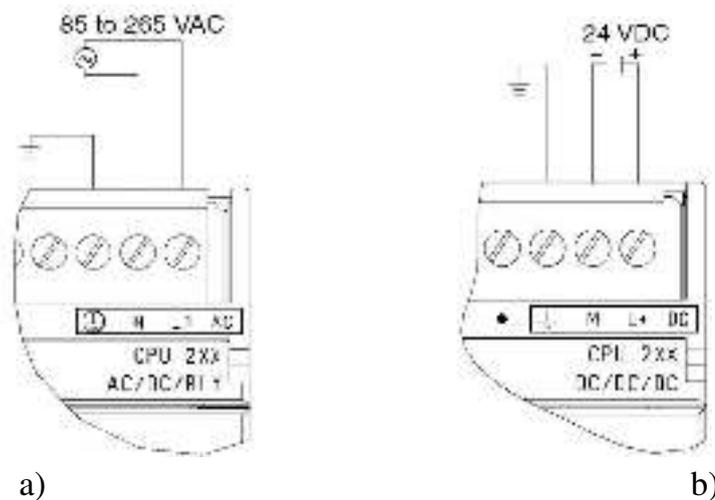
Tùy theo loại và họ PLC mà các CPU có thể là khối riêng hoặc có đặt sẵn các ngõ vào/ra cũng như một số chức năng đặc biệt. Hầu hết các PLC họ S7-200 được nhà sản xuất lắp đặt các khâu vào, khâu ra và CPU trong cùng một vỏ hộp. Nhưng nguồn cung cấp cho các khâu này hoàn toàn độc lập nhau. Nguồn cung cấp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

Xoay chiều: 20...29 V_{AC}, f = 47...63 Hz;

85...264 V_{AC}, f = 47...63 Hz

Một chiều: 20,4 ... 28,8 V_{DC}

Sơ đồ nối dây nguồn cung cấp cho CPU như hình 3.5



a) *Cấp nguồn cho CPU2xx loại DC/DC/DC*

b) *Cấp nguồn cho CPU 2xx loại AC/DC/RLY*

Hình 3.5: Nối nguồn cung cấp cho CPU

Để có thể nhận biết việc cấp nguồn cho CPU, khối vào, khối ra số ta căn cứ vào các chữ số đi kèm theo CPU. Các mã số kèm theo CPU 2xx có thể có như sau:

- CPU 2xx DC/DC/DC: Nguồn cấp cho CPU là DC, nguồn cho ngõ vào là DC, nguồn cấp cho ngõ ra là DC.

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

3.1.4 Kết nối vào/ra số với ngoại vi

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

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

Kết nối các ngõ vào số với ngoại vi

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

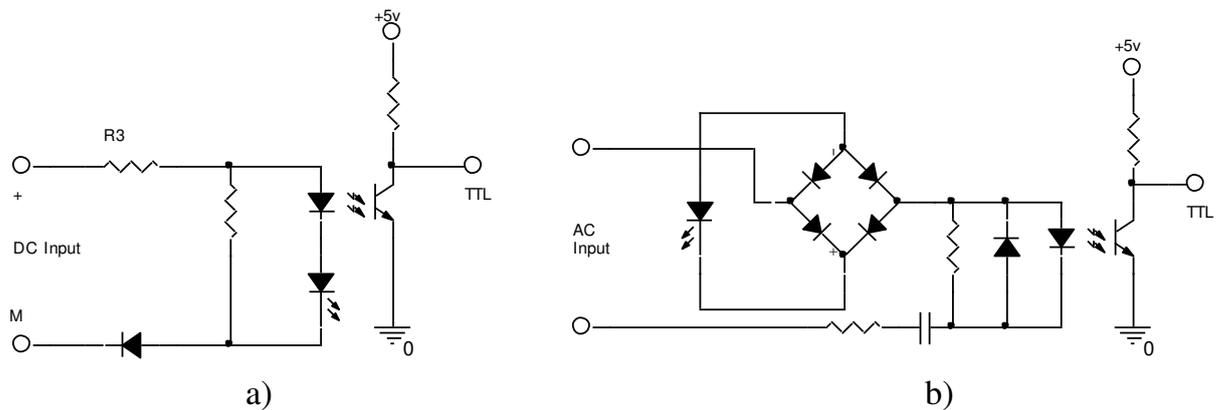
Trong trường hợp nào cũng vậy, các ngõ vào cũng phải được cung cấp nguồn riêng với cấp điện áp tùy thuộc vào loại ngõ vào. Cần lưu ý trong một khối ngõ vào cũng như các ngõ vào được tích hợp sẵn trên CPU có thể có các nhóm được cung cấp nguồn độc lập nhau. Vì vậy cần lưu ý khi cấp nguồn cho các nhóm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

Xoay chiều: $15...35V_{AC}, f = 47...63Hz$; dòng cần thiết nhỏ nhất $4mA$

79...135 $V_{AC}, f = 47...63 Hz$; dòng cần thiết nhỏ nhất $4mA$

Một chiều: $15...30 V_{DC}$; dòng cần thiết nhỏ nhất $4mA$

Sơ đồ mạch điện bên trong của một số ngõ vào được cho như hình 3.6.



a) Mạch điện của 1 ngõ vào số sử dụng nguồn cung cấp DC

b) Mạch điện của 1 ngõ vào số sử dụng nguồn cung cấp AC

Hình 3.6: Mạch điện ngõ vào PLC

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

+ Ngõ vào DC:

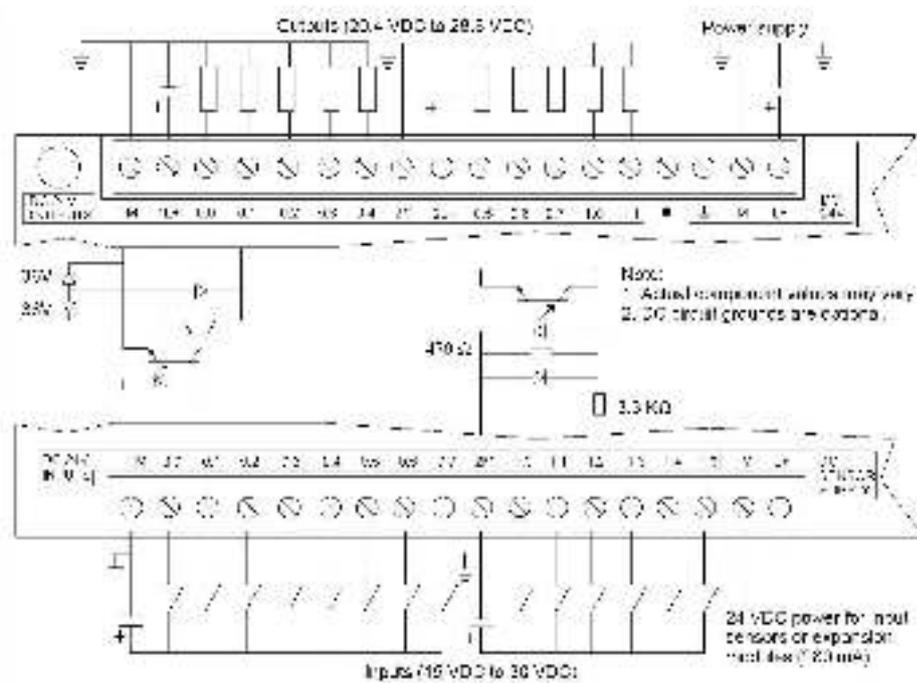
- Điện áp DC thường thấp do đó an toàn hơn.
- Đáp ứng ngõ vào DC rất nhanh.
- Điện áp DC có thể được kết nối với nhiều phần tử trong hệ thống điện.

+ 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....

Đối với các ngõ vào ra của CPU214 DC/DC/DC, CPU224 AC/DC/Relay theo số tay được kết nối như hình 3.7 và hình 3.9



Hình 3.7: Sơ đồ nối dây CPU 214 DC/DC/DC với nguồn và ngoại vi
Kết nối các ngõ ra số với ngoại vi

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

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

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

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

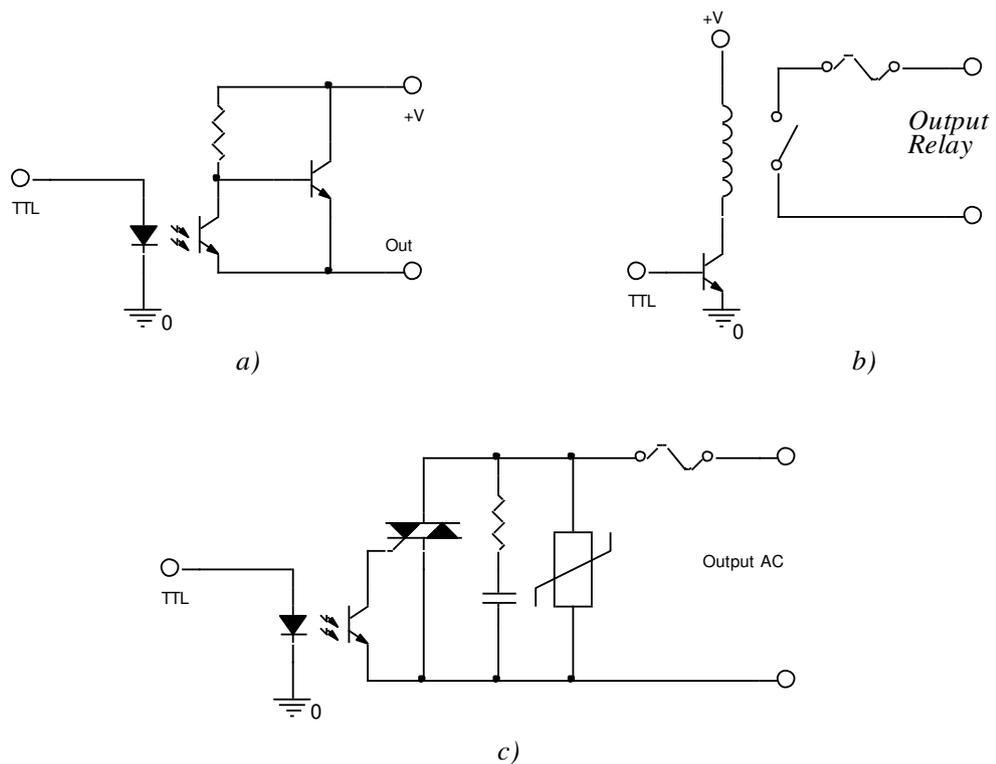
$20,4 \dots 28,8V_{DC}$ đối với ngõ ra Transistor

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. Rơ le là ngõ ra linh hoạt nhất. Chúng có thể là ngõ ra AC và DC. Tuy nhiên đáp ứng của ngõ ra rơ le chậm, giá thành cao và bị hư hỏng sau vài triệu lần đóng cắt. Còn ngõ ra transistor thì chỉ sử dụng với nguồn cung cấp là DC và ngõ ra triac thì chỉ sử dụng được với nguồn AC. Tuy nhiên đáp ứng của các ngõ ra này nhanh hơn.

Sơ đồ mạch điện bên trong của các ngõ ra được cho như hình 3.8.

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

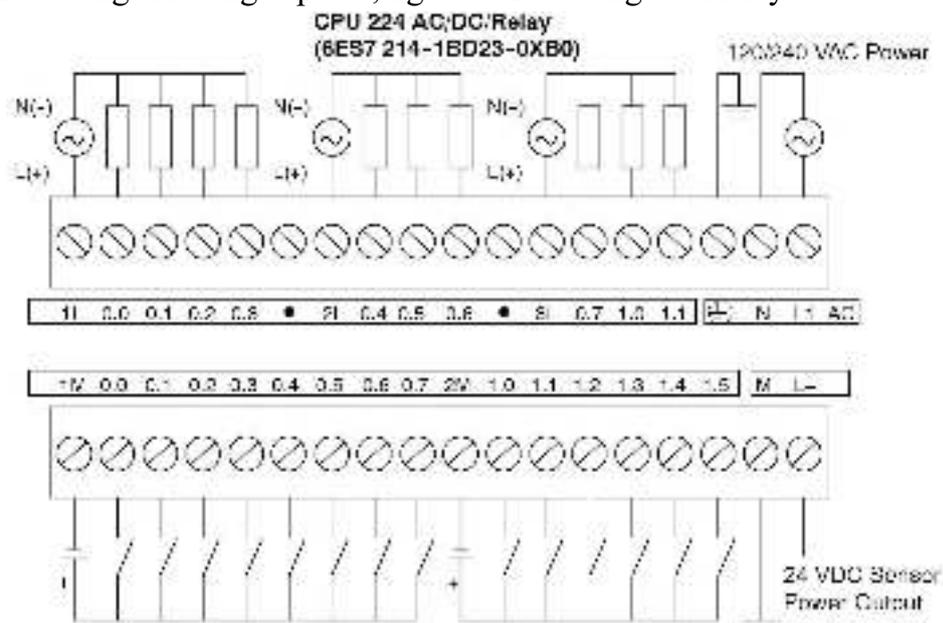
Đối với các ngõ ra số, khi kết nối với ngoại vi, ngoại trừ các trường hợp đặc biệt thì thông thường mỗi một ngõ ra được kết nối với một đối tượng điều khiển nhận tín hiệu nhị phân như: đèn báo, cuộn dây rơ le, chuông báo....



Hình 3.8: Các mạch điện khác nhau bên trong ngõ ra của PLC

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

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 3.7 là ví dụ của CPU 224 với nguồn cung cấp DC, ngõ vào DC và ngõ ra DC được nối dây với ngoại vi (trích từ sổ tay S7-200 Programmable Controller System Manual). Ta nhận thấy mỗi một nhóm ngõ vào cũng như một nhóm ngõ ra và CPU được cung cấp nguồn riêng là 24 Vdc. Ngoài ra trên khối CPU còn có nguồn phụ 24 Vdc (đến 280 mA) có thể được sử dụng để cung cấp cho các cảm biến hoặc khối mở rộng. Còn hình 3.9 là ví dụ của CPU 224 với nguồn cung cấp AC, ngõ vào DC và ngõ ra Relay.



Hình 3.9: Sơ đồ nối dây CPU 224 AC/DC/Relay với nguồn và ngoại vi

3.2. CÀI ĐẶT VÀ SỬ DỤNG PHẦN MỀM STEP7-MICROWIN 4.0

3.2.1 Cài đặt phần mềm Step7-Microwin 4.0

STEP 7-Micro/WIN là một phần mềm lập trình cho họ PLC S7-200. Hiện phiên bản đang được sử dụng là STEP 7-Micro/Win V4.0 Service Pack 9.

Yêu cầu hệ điều hành và phần cứng

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

- Microsoft Windows 2000 Service Pack 3 hoặc cao hơn, Windows XP Home, hoặc Windows XP;

- Có ít nhất 350 MB ổ đĩa cứng còn trống;

- Sử dụng chế độ cài đặt font chữ nhỏ độ phân giải màn hình tối thiểu là 1024x768 pixels;

Nếu chưa có cáp để kết nối máy tính với PLC S7-200 thì ta vẫn có thể soạn thảo chương trình ở chế độ offline và kiểm tra hoạt động của chương trình với một phần mềm mô phỏng.

Để truyền thông với S7-200, ta cần một trong các phần cứng sau:

- PC/PPI Cable kết nối CPU S7-200 với PC qua cổng USB;

- PC/PPI Cable kết nối CPU S7-200 với PC qua cổng RS232 (COM1 hoặc COM2);

- CP card (Communications processor) và cáp MPI (multipointinterface);

- EM241 modem ;

- CP243-1 hoặc CP243-1 IT Ethernet.

Cài đặt phần mềm

Thực hiện theo các bước sau:

1. Đóng tất cả các ứng dụng

2. Chèn đĩa CD STEP 7-Micro/Win vào ổ đĩa CD-Rom hoặc file. Chương trình sẽ được tự động cài đặt. Ta cũng có thể khởi động chương trình cài đặt bằng cách nhấp



đúp chuột vào file “Setup.exe” trên CD hoặc file.

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

4. Khi cài đặt xong, hộp thoại “set PG/PC Interface” tự động xuất hiện. Kích “Cancel” để kết thúc.

5. Ta cần khởi động lại máy để hoàn tất việc cài đặt. Sau khi đã cài đặt xong có thể bắt đầu soạn thảo chương trình nhờ phần mềm STEP 7-Micro/Win bằng cách nhấp đúp chuột vào biểu tượng STEP 7-MicroWin trên màn hình.

Chú ý: Khi cài đặt phiên bản STEP 7-Micro/Win V4.0 SP9 thì trước tiên ta cần phải uninstall phiên bản cũ và sau đó mới cài đặt được phiên bản này. Sau khi download ta nhấp đúp chuột vào file STEP7-MicroWin V.4.0.exe và thực hiện theo các bước sau:

Bước 1: Uninstall phiên bản STEP7-Micro/Win V4.0 bằng công cụ “control panel” trong Window (menu Start → setting → control panel → Add or remove program);

chương trình ngắt. Sử dụng chương trình con khi chúng ta muốn phân chia nhiệm vụ điều khiển. Mỗi một chương trình con viết cho một nhiệm vụ nhỏ hoặc khi có các yêu cầu điều khiển tương tự nhau (ví dụ: điều khiển băng tải 1, điều khiển băng tải 2...) thì chúng ta chỉ cần tạo ra chương trình con một lần và có thể gọi ra nhiều lần từ chương trình chính.

Sử dụng chương trình con có một số ưu điểm sau:

- Chương trình điều khiển được chia theo nhiệm vụ điều khiển nên có cấu trúc rõ ràng, rất thuận tiện cho việc chỉnh sửa hay kiểm tra chương trình.
- Giảm thời gian vòng quét của chương trình. CPU không phải liên tục xử lý tất cả các lệnh của chương trình mà chỉ xử lý chương trình con khi có lệnh gọi tương ứng.
- Chương trình con cho phép giảm công việc soạn thảo khi có các yêu cầu điều khiển tương tự nhau.

Chương trình ngắt INT(interrupt routine)

Chương trình ngắt được thiết kế để sử dụng cho một sự kiện ngắt được định nghĩa trước. Bất cứ khi nào sự kiện ngắt xác định xảy ra, thì S7-200 thực hiện chương trình ngắt.

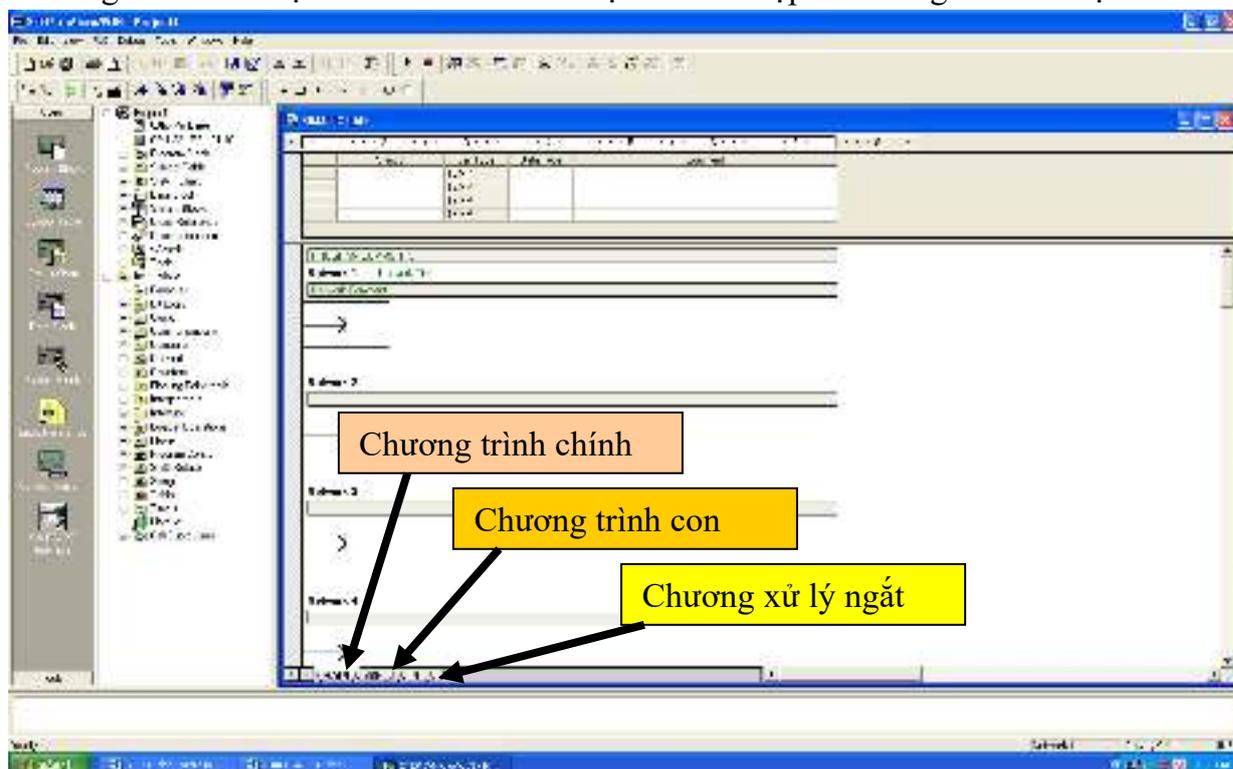
Chương trình ngắt không được gọi bởi chương trình chính mà theo sự kiện ngắt xảy ra. Chương trình ngắt sẽ chỉ được xử lý mỗi khi sự kiện ngắt xảy ra.

Khối hệ thống (system block)

System block cho phép ta cấu hình các tùy chọn phần cứng khác nhau cho S7-200.

Khối dữ liệu (data block)

Data block lưu trữ các giá trị biến khác nhau (vùng nhớ V) được sử dụng trong chương trình. Giá trị ban đầu của các dữ liệu có thể nhập vào trong khối dữ liệu.



Hình 3.15: Vị trí cấu chương trình trên giao diện Step7-MicroWin 4.0

3.3.3 Ngôn ngữ lập trình

Để có thể soạn thảo chương trình cho các PLC S7-200, chúng ta dùng phần mềm Step7-MicroWin. Và cũng giống như PLC của các hãng khác, chúng ta có 3 dạng soạn thảo thông dụng là dạng LAD, FBD và STL. Việc chọn dạng soạn thảo nào để viết chương trình điều khiển là do người dùng.

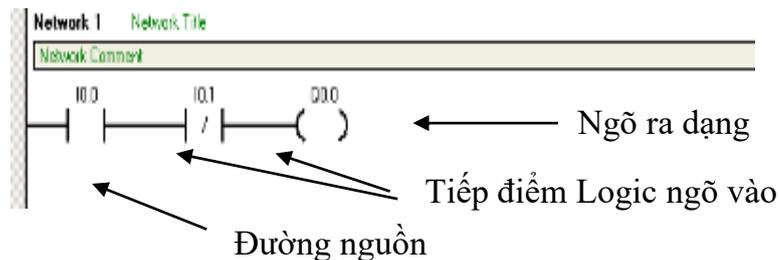
Dạng hình thang: LAD (Ladder logic)

Ở dạng soạn thảo này chương trình được hiển thị gần giống như sơ đồ nối dây một mạch trang bị điện dùng các relay và contactor. Chúng ta xem như có một dòng điện từ một nguồn điện chạy qua một chuỗi các tiếp điểm logic ngõ vào từ trái qua phải để tới ngõ ra. Chương trình điều khiển được chia ra làm nhiều Network, mỗi một Network thực hiện một nhiệm vụ nhỏ và cụ thể. Các Network được xử lý lần lượt từ trên xuống dưới và từ trái sang phải.

Các phần tử chủ yếu dùng trong dạng soạn thảo này là:

- Tiếp điểm không đảo: —|—
- Tiếp điểm đảo: —|/—
- Ngõ ra: $()$
- Các hộp chức năng (Box): các chức năng được biểu diễn ở dạng hộp như các phép toán số học, định thời, bộ đếm...

Ví dụ:



Hình 3.16: Cấu trúc chương trình vào/ra của PLC

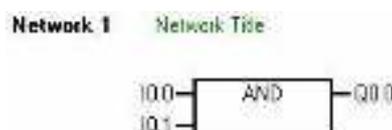
Dạng soạn thảo này có một số ưu điểm:

- Dễ dàng cho những người mới bắt đầu lập trình
- Biểu diễn dạng đồ họa dễ hiểu và thông dụng với sơ đồ trang bị điện
- Luôn luôn có thể chuyển từ dạng LAD sang STL

Dạng khối chức năng Logic: FBD (Function Block Diagram)

Dạng soạn thảo FBD hiển thị chương trình ở dạng đồ họa tương tự như sơ đồ các cổng logic. FBD không sử dụng khái niệm đường nguồn cung cấp trái và phải; do đó khái niệm “dòng điện” không được sử dụng. Thay vào đó là logic ”1”. Không có tiếp điểm và cuộn dây như ở dạng LAD, nhưng có các cổng logic và các hộp chức năng. Các cổng logic như AND, OR, XOR... sẽ tương ứng với các tiếp điểm logic nối tiếp hay song song...

Ví dụ:



Hình 3.17: Dạng FBD

Đầu ra của các công logic hay hộp chức năng có thể được sử dụng để nối tiếp với đầu vào của các công logic hay các hộp chức năng khác. Với dạng soạn thảo này có một số điểm chính sau:

- Biểu diễn ở dạng đồ họa các công chức năng Logic giúp chúng ta dễ đọc hiểu theo trình tự điều khiển.
- Luôn có thể chuyển từ hiển thị dạng FBD sang STL.

Dạng liệt kê lệnh: STL (StaTement List)

Đây là dạng soạn thảo chương trình dạng tập hợp các câu lệnh. Người dùng phải nhập các câu lệnh từ bàn phím, giữa lệnh và toán hạng (toán hạng có thể là địa chỉ, dữ liệu) có khoảng trắng và mỗi lệnh chiếm một hàng. Ở dạng soạn thảo này sẽ có một số chức năng mà ở dạng soạn thảo LAD hay FBD không có.

Ví dụ:

Network 1	Network Title	
LD	I0.0	// Đọc trạng thái I0.0
A	I0.1	// Thực hiện AND với I0.1
=	Q0.0	// Xuất kết quả ra Q0.0

Dạng soạn thảo này có một số điểm chính:

- Là dạng soạn thảo phù hợp cho những người có kinh nghiệm lập trình máy tính, lập trình điều khiển.
- STL cho phép giải quyết một số vấn đề mà đôi khi dùng LAD hoặc FBD không thực hiện được.
- Luôn luôn có thể chuyển từ dạng LAD hay FBD về dạng STL nhưng khi chuyển ngược lại từ STL sang LAD hay FBD sẽ có một số phần tử chương trình không chuyển được.

3.3.4 Soạn thảo chương trình với phần mềm STEP7-Micro/Win V4.0

Mở màn hình soạn thảo chương trình

Để mở STEP 7-MicroWin, nhấp đúp chuột vào biểu tượng STEP 7-Microwin



trên màn hình desktop, hoặc chọn **Start > Simatic > Step7-MicrowV4.0**. Giao diện màn hình có dạng (hình 3.9).

Vùng soạn thảo chương trình

Vùng soạn thảo chương trình chứa chương trình và bảng khai báo biến cục bộ của khối chương trình đang được mở. Chương trình con (viết tắt là SUB) và chương trình ngắt (viết tắt là INT) xuất hiện ở cuối cửa sổ soạn thảo chương trình. Tùy thuộc vào việc nhấp chuột ở mục nào mà cửa sổ màn hình soạn thảo chương trình tương ứng sẽ được mở.

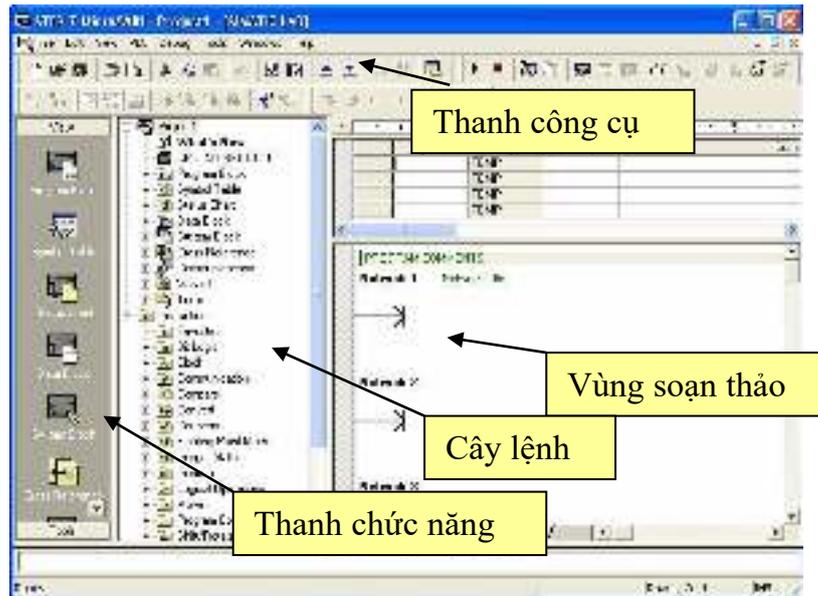
Cây lệnh

Cây lệnh hiển thị tất cả các đối tượng của dự án và các lệnh để viết chương trình điều khiển. Có thể sử dụng phương pháp “drag and drop” (kéo và thả) từng lệnh riêng từ cửa sổ cây lệnh vào chương trình, hay nhấp đúp chuột vào một lệnh mà muốn chèn nó vào vị trí con trỏ ở màn hình soạn thảo chương trình.

Thanh chức năng

Thanh chức năng chứa một hóm các biểu tượng để truy cập các đặc điểm chương

trình khác nhau của STEP 7-Micro/Win.



Hình 3.18: Màn hình soạn thảo chương trình STEP 7-MicroWin

Program Block:



Nhấp trái chuột vào biểu tượng này để mở ra cửa sổ soạn thảo các chương trình ứng dụng (OB1, SUB hoặc INT)

Symbol Table



Bảng ký hiệu (Symbol table) cho phép người dùng mô tả các địa chỉ sử dụng trong chương trình dưới dạng các tên gọi nhớ. Điều này giúp cho việc đọc hiểu chương trình dễ dàng và khi viết chương trình ít bị sai sót do sử dụng trùng địa chỉ.

Status Chart



Bảng trạng thái (Status chart) cho phép người dùng giám sát trạng thái các ngõ vào và thay đổi trạng thái từng ngõ ra. Sử dụng bảng trạng thái để kiểm tra nối dây phần cứng và xem nội dung các vùng nhớ.

Bảng 3.1 Status chart

	Address	Format	Current Value	New Value
1	I0.0	Bit		
2	Q0.0	Bit		2#1
3	VW2	Unsigned		1200
4		Signed		

- Trong đó:
- Cột **Address**: Cho phép nhập địa chỉ các biến hay vùng nhớ
 - Cột **Format**: Cho phép chọn dạng dữ liệu của địa chỉ
 - Cột **Current Value**: Hiện thị giá trị hiện hành của địa chỉ
 - Cột **New Value**: Cho phép thay đổi trạng thái ngõ ra.

Data Block



Sử dụng Data Block như một vùng nhớ để đặt trước dữ liệu cho các biến thuộc vùng nhớ V. Có thể tạo ra các Data block khác nhau và đặt tên theo dữ liệu chương trình.



System Block

Đây là khối chức năng hệ thống, khi mở System Block chúng ta có thể cài đặt các chức năng như:

- *Communication ports*: Chọn các thông số truyền thông với thiết bị khác như máy tính hay CPU khác.

- *Retentive Ranges*: Chọn các vùng nhớ và địa chỉ sẽ có thuộc tính Retentive

- *Output Tables*: Cho phép thiết lập cấu hình trạng thái ON và OFF của mỗi ngõ ra số khi CPU chuyển từ trạng thái Run sang Stop

- *Input filter*: Cho phép chọn thời gian trễ cho một vài ngõ vào hoặc tất cả ngõ vào số (từ 0.2ms đến 12.8 ms). Mục đích là giúp chống nhiễu ở việc nối dây ngõ vào.

- *Pulse Catch Bits*: Cho phép thiết lập một ngõ vào để bắt lấy sự chuyển đổi trạng thái tín hiệu rất nhanh. Ngay khi có chuyển đổi, giá trị ngõ vào sẽ được chốt cho đến khi được đọc bởi chu kỳ quét của PLC.

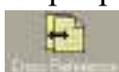
- *Background Time*: Cho phép thiết lập lượng thời gian PLC sẽ dành cho các hoạt động nền trong chế độ RUN. Đặc điểm này được sử dụng chủ yếu để điều khiển ảnh hưởng của chu kỳ quét khi xử lý trạng thái và trong hoạt động soạn thảo runtime

- *EM Configuration* Các module intelligent và địa chỉ cấu hình tương ứng được định nghĩa trong dự án. Thường thì STEP 7-Micro/WIN wizard đặt các địa chỉ này.

- *Configure LED*: LED SF/DIAG (System Fault/Diagnostic) có thể được chọn sáng khi thực hiện chức năng cưỡng bức (Force) hoặc xảy ra lỗi vào/ra (I/O).

- *Increase Memory*: Tăng bộ nhớ chương trình bằng cách không cho soạn thảo ở chế độ RUN. Đối với bộ nhớ Dữ liệu thì không thể.

- *Password*: Cho phép đặt mật khẩu để bảo vệ chương trình.



Cross Reference

Bảng tham chiếu cho biết những địa chỉ vùng nhớ nào (Byte, bit, word hay DWord, timer, counter...) đã sử dụng và vị trí (location) trong chương trình cũng như chức năng của chúng.

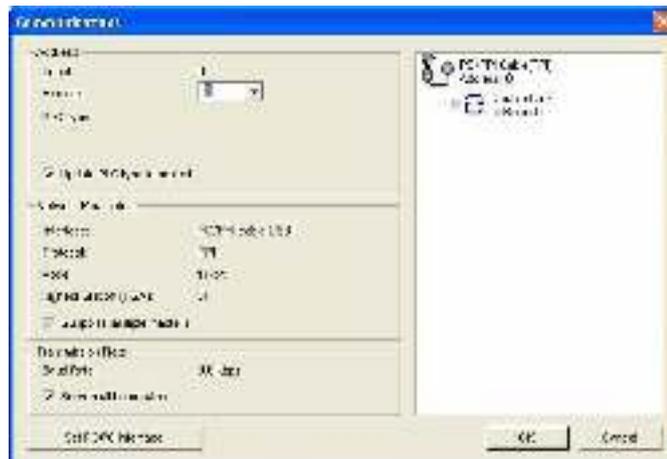


Communication

và Set PG/PC



Các biểu tượng này khi kích hoạt sẽ mở ra hộp thoại cho phép chúng ta cài đặt các giao tiếp với máy tính như: chọn cổng giao tiếp, địa chỉ CPU, tốc độ truyền. Đây là bước cần thực hiện khi bắt đầu giao tiếp giữa PLC với máy tính.



Hình 3.19: Cửa sổ Communications



Hình 3.20: Cửa sổ Set PG/PC Interface.

Thanh công cụ (Toolbar) trong STEP7-Micro/WIN

Trong phần mềm có đặt sẵn nhiều công cụ giúp người lập trình dễ dàng trong việc sử dụng. Các công cụ có ý nghĩa như sau:

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

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

Save Project (File menu)  : Lưu dự án

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

Print Preview (File menu)  : Xem trước khi in

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

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

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

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

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

Data Block).

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

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

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

Option (Tools menu) : Truy cập menu Options

RUN (PLC menu) : Đặt PLC ở chế độ RUN

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

Program Status (Debug menu) : ON/OFF trạng thái chương trình trong PLC, thể hiện trạng thái làm việc của các phần tử trong chương trình.

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

Chart Status (Debug menu) : ON/OFF hiển thị trạng thái dữ liệu trong bảng Status chart.

Trend View (View menu): ON/OFF xem trạng thái dữ liệu trong PLC ở dạng đồ thị

Pause Trend View: Dừng việc vẽ đồ thị dữ liệu

Single Read (Debug menu): Sử dụng Single Read để cập nhật một lần tất cả các giá trị trong bảng Status Chart.

Write All (Debug menu): Ghi tất cả các giá trị ở cột New Value trong bảng Status Chart vào PLC.

Force (Debug menu): Cường bức dữ liệu PLC

Unforce For (Debug menu): Gỡ bỏ cường bức dữ liệu PLC

Unforce All (Debug menu): Gỡ bỏ tất cả các cường bức trong bảng Status Chart.

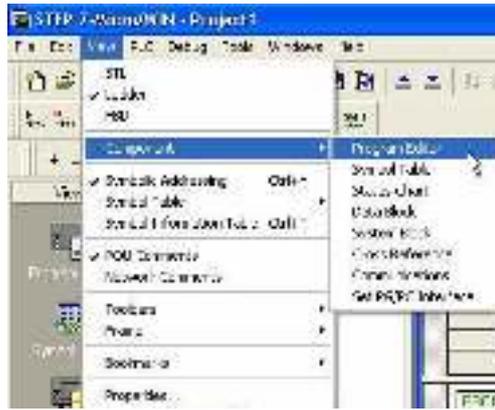
Read All Forced (Debug menu): Đọc tất cả các giá trị cường bức trong Status Chart.

Tạo một dự án STEP 7-Microwin

Tạo dự án mới

Để tạo một dự án mới trong STEP 7-Micro/Win, chọn menu **File > New** hoặc biểu tượng  trong toolbar để mở hộp thoại "New" cho phép tạo mới một dự án (project).

Trong thanh chức năng, bấm vào biểu tượng  hoặc vào menu **View > Component > Program Editor** để mở màn hình soạn thảo chương trình (hình 3.11).



Hình 3.21: Đường dẫn vào màn hình soạn thảo chương trình

Cũng trong menu View, ta có thể chọn ngôn ngữ lập trình là STL, Ladder hay FBD theo mong muốn.

Để soạn thảo bảng ký hiệu cho các địa chỉ ta bấm vào biểu tượng  trong thanh chức năng, hoặc vào menu **View > Component > symbol Table**. Sau đó có thể đặt ký hiệu cho các địa chỉ.

Lưu dự án

Để lưu dự án, nhấp chuột vào biểu tượng  hoặc vào menu **File > Save**. Cửa sổ màn hình xuất hiện như hình 3.22. Chọn thư mục cần chứa dự án, đặt tên dự án và nhấp chuột vào thẻ **Save** để lưu dự án.

Mở một dự án

Để mở một dự án đang có sẵn, nhấp chuột vào biểu tượng  vào menu **File > Open**. Cửa sổ màn hình **Open** xuất hiện. Chọn thư mục chứa chương trình cần mở, chọn tên dự án và sau đó nhấp chuột vào thẻ **Open**.

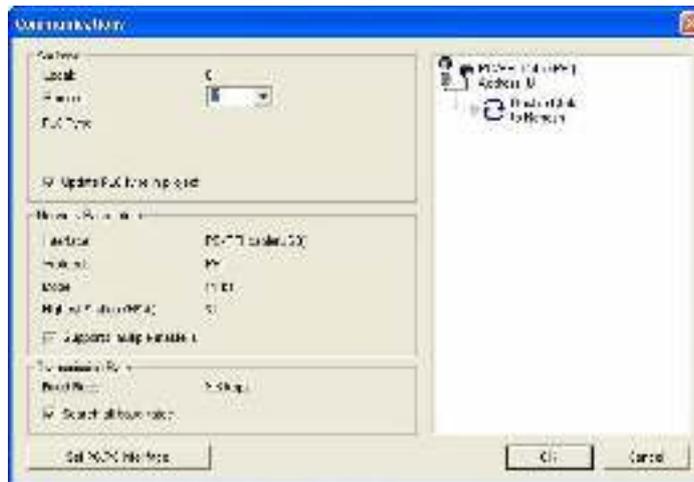
Kết nối truyền thông S7-200 với thiết bị lập trình

Để kết nối truyền thông S7-200 với thiết bị lập trình thì cần phải có cáp kết nối. Việc kết nối truyền thông thực hiện theo các bước sau:

Bước 1: Nhấp chuột vào biểu tượng communication  trong thanh chức năng hay vào **View > Component > Communications**.



Hình 3.22: Cửa sổ màn hình lưu dự án



Hình 3.23: Màn hình thiết lập truyền thông

Bước 2. Kiểm tra xem địa chỉ của cáp PC/PPI trong hộp thoại có được đặt là 0 chưa? Thường mặc định là 0.

Bước 3. Kiểm tra tham số mạng (Network Parameters) và tốc độ truyền (Transmission Rate) có đúng chưa. Nếu chưa đúng thì nhấp chuột vào thẻ



để thiết lập lại giao tiếp giữa PC và PLC

Bước 4. Nhấp đúp chuột vào biểu tượng  để tìm trạm S7-200 và một biểu tượng CPU cho trạm S7-200 được kết nối sẽ hiển thị.

Bước 5. Chọn S7-200 và nhấp OK. Nếu STEP 7-Microwin không tìm ra CPU S7-200, kiểm tra việc đặt chỉnh các tham số truyền thông và lặp lại bước này.

Bước 6. Sau khi đã thiết lập truyền thông với S7-200, ta có thể sẵn sàng tạo và download chương trình vào CPU.

Tải chương trình từ PLC vào máy tính

Có thể sử dụng biểu tượng trên toolbar hoặc menu File để tải (upload chương trình từ PLC về máy tính khi sử dụng phần mềm STEP 7-Microwin. Cần lưu ý là PLC được kết nối truyền thông với thiết bị lập trình, chương trình tải mới sẽ đè lên chương trình đang có.

Thủ tục tải chương trình được thực hiện qua các bước sau:

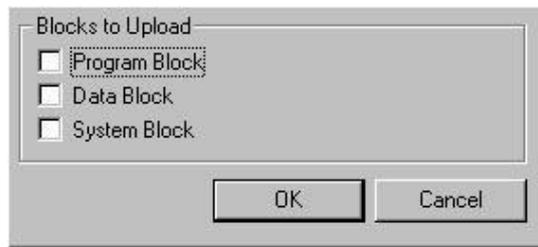
Bước 1: Trong STEP 7-Microwin mở một dự án để giữ các khối sẽ được upload từ PLC.

* Nếu Upload từ một chương trình rỗng, chọn File > new hoặc sử dụng biểu tượng New Project trên thanh toolbar.

* Nếu Upload từ một chương trình tồn tại, chọn File > Open hoặc sử dụng biểu tượng Open Project trên thanh toolbar.

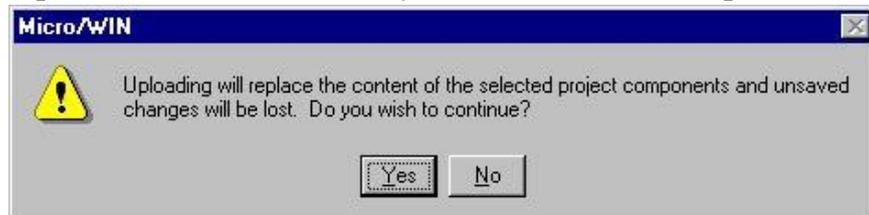
Bước 2: Chọn **File > Upload** hoặc sử dụng biểu tượng  Upload toolbar để khởi động quá trình upload.

Bước 3: Hộp thoại Upload xuất hiện để yêu cầu chọn các khối: program block, data block, and system block. Hãy chọn các khối muốn Upload, và sau đó nhấp OK.



Hình 3.24: Hộp thoại Upload

Bước 4: Step7 Microwin hiển thị chú ý sau: Nhấn **Yes** để chấp nhận việc upload.



Hình 3.25: Chú ý khi Upload từ PLC về thiết bị lập trình

STEP 7-Micro/WIN hiển thị một thông báo khi upload các khối thành công từ PLC về thiết bị lập trình hoặc máy tính PC.

Nạp chương trình từ thiết bị lập trình vào PLC

Khi cho phép kết nối truyền thông giữa PC và PLC, ta có thể download chương trình vào PLC. Cần lưu ý rằng khi download một program block, data block hay system block vào PLC thì nội dung của các khối được download vào sẽ viết đè lên các khối hiện hành trong PLC. Các bước thực hiện như sau:

Bước 1. Trước khi download vào PLC, cần phải kiểm tra xem PLC đã ở chế độ Stop chưa thông qua đèn báo STOP trên PLC. Nếu công tắc chọn chế độ trên PLC đặt ở vị trí TERM thì ta có thể chọn PLC ở chế độ RUN hoặc STOP từ máy lập trình. Nếu PLC không ở chế độ STOP, thì nhấp chuột vào biểu tượng STOP  trong thanh toolbar hoặc chọn PLC > STOP.

Trong trường hợp không dùng phần mềm thì chuyển công tắc chọn chế độ cho PLC về vị trí STOP.

Bước 2: Nhấp chuột vào biểu tượng  trong thanh toolbar hoặc chọn file > Download. Hộp download xuất hiện.

Bước 3: Chọn các khối cần download. Thông thường là chọn tất cả.

Bước 4: Nhấp OK để bắt đầu quá trình download.

Bước 5: Nếu download thành công, thì một hộp thoại hiển thị thông báo: *Download Successful*. Tiếp tục đến bước 12.

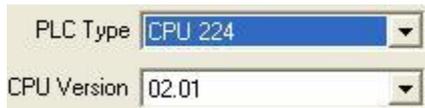
Bước 6: Nếu loại PLC được chọn cho chương trình trong STEP /Micro/WIN không phù hợp với PLC thực tế, thì một hộp thoại xuất hiện với thông báo:

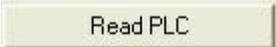
"The PLC type selected for the project does not match the remote PLC type. Continue Download?".

Bước 7: Đặt lại loại PLC cho phù hợp, chọn **No** để dừng tiến trình download.

Bước 8: Chọn **PLC > Type** để vào hộp thoại chọn loại PLC

Bước 9: Có thể chọn đúng loại PLC theo danh sách trong mục PLC type Cửa hộp



thoại. hoặc nhấp chuột vào thẻ  để Step7- Microwin tự động tìm đúng loại PLC đang kết nối.

Bước 10: Nhấp **OK** để chấp nhận loại PLC và đóng hộp thoại.

Bước 11: Khởi động lại quá trình download bằng cách nhấp chuột vào biểu tượng download  trong toolbar hay chọn **File > Download**

Bước 12: Ngay khi download thành công, ta phải chuyển PLC từ STOP sang RUN trước khi PLC có thể thực hiện chương trình. Nhấp chuột vào biểu tượng RUN  trong toolbar hay chọn **PLC > RUN** để chuyển PLC sang chế độ RUN khi công tắc chọn chế độ cho PLC để ở vị trí TERM.

Trường hợp sử dụng công tắc thì chuyển từ vị trí STOP sang RUN.

Soạn thảo chương trình

Trước khi soạn thảo chương trình, cần phải hoàn thành các bước sau đây:

- Kết nối giữa PLC và máy tính
- Kết nối dây đúng các ngõ vào và ra với ngoại vi

Trường hợp không có PLC, thì ta chỉ có thể soạn thảo chương trình và lưu trữ lại. Còn nếu muốn kiểm tra thì cần phải có phần mềm mô phỏng S7-200.

Các bước để soạn thảo một dự án mới:

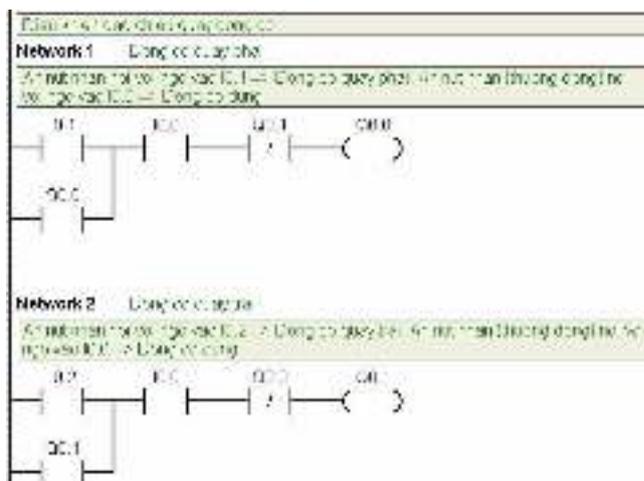
1. Mở màn hình soạn thảo chương trình
2. Nhập bảng ký hiệu
3. Nhập chương trình
4. Lưu chương trình
5. Download chương trình vào CPU.
6. Đặt CPU ở chế độ RUN.
7. Tìm lỗi và chỉnh sửa chương trình.

Thực hành

Để hiểu được phần mềm STEP 7-Micro/WIN dễ dàng, chúng ta nên viết một ví dụ đơn giản được cho ở hình 3.26 và bảng thiết lập vào/ra cho ở bảng 3.1. Do mới bắt đầu, ta nên viết chương trình ở dạng LAD, rồi sau đó có thể xem ở dạng FBD hay STL.

Bảng 3.1: Gán tên chức năng

	Symbol	Address	Comment
1	S_Stop	I0.0	Nút ấn dừng động cơ, thường đóng
2	S_Right	I0.1	Nút ấn động cơ quay phải, thường hở
3	S_Left	I0.2	Nút ấn động cơ quay trái, thường hở
4	K1	Q0.0	Contactơ cấp điện cho động cơ quay phải
5	K2	Q0.1	Contactơ cấp điện cho động cơ quay trái



Hình 3.26: Chương trình điều khiển đảo chiều quay động cơ

Các bước thực hiện:

Bước 1: Mở màn hình soạn thảo chương trình

Nhấp chuột vào biểu tượng Program Block để mở màn hình soạn thảo chương trình (hình 3.26). Chú ý cửa sổ cây lệnh (instruction tree) và vùng soạn thảo chương trình. Sử dụng cây lệnh để chèn các lệnh được biểu diễn ở dạng LAD vào các networks của màn hình soạn thảo chương trình bằng cách kéo và thả các lệnh từ cây lệnh vào các networks.

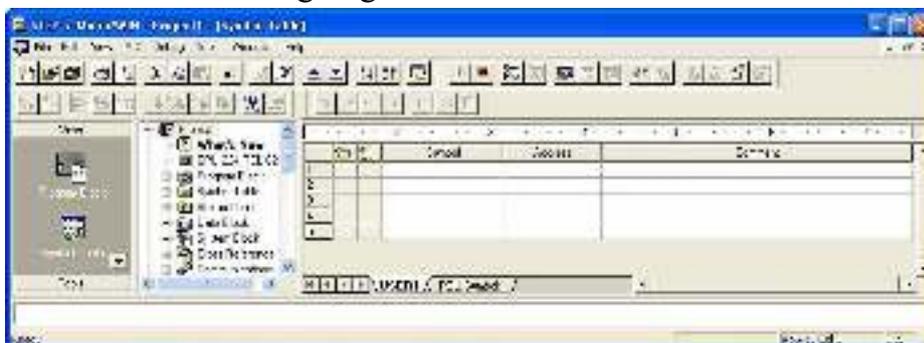
Để có thể nhập đầy đủ các chú thích (comment), thì cần hiển thị các chú thích trong màn hình soạn thảo chương trình. Vào **View > POU Comment** để hiển thị dòng chú thích tiêu đề chương trình và **View > Network comments** để hiển thị dòng chú thích của từng network.

Bước 2: Nhập bảng ký hiệu

Nhấp chuột vào biểu tượng Symbol Table  để mở màn hình soạn thảo bảng ký hiệu.

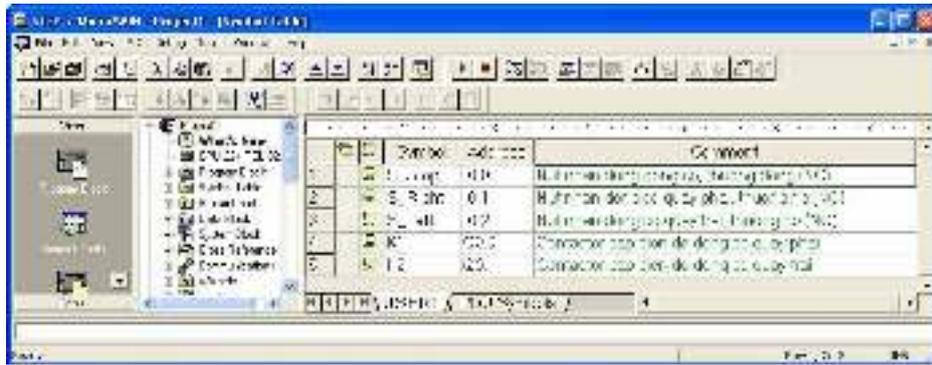
Nhập các thông tin (chữ không dấu) ở bảng 3.1 vào bảng Symbol Table. Với:

- *Cột ký hiệu* tương ứng với cột *Symbol*.
- *Cột địa chỉ* tương ứng với cột *Address*.
- *Cột chú thích* tương ứng với cột *comment*



Hình 3.27: Màn hình soạn thảo bảng ký hiệu

Sau khi nhập xong, ta có bảng ký hiệu như hình 3.28



Hình 3.28: Bảng ký hiệu các phần tử trong chương trình

Trong quá trình lập trình có thể phát sinh thêm các địa chỉ mới. Khi phát sinh thêm địa chỉ mới, ta nên bổ sung địa chỉ đó vào trong bảng ký hiệu để dễ dàng cho quá trình tìm và xử lý lỗi sau này.

Bước 3: Nhập chương trình

Nhấp chuột vào biểu tượng Program Block  để mở lại màn hình soạn thảo chương trình (hình 3.26).

Nhập Network 1: Động cơ quay phải

Khi ấn nút nhấn S_Right (I0.1), thì tiếp điểm I0.1 đóng, nút nhấn S_Stop là thường đóng nên ngõ vào I0.0 luôn luôn có điện hay tiếp điểm I0.0 cũng đóng, và bình thường ngõ ra Q0.1 cũng không có điện (0) nên tiếp điểm này cũng đóng. Kết hợp 3 tiếp điểm này sẽ có dòng điện cung cấp cho cuộn dây Q0.0 (nối với K1). Contactor K1 có điện đóng tiếp điểm động lực của nó để cấp nguồn cho động cơ quay phải. Tiếp điểm Q0.0 (song song I0.1) đóng duy trì dòng cung cấp cho Q0.0 khi nút nhấn S_Right hở ra.

Nhập các dòng chú thích như đã cho trong hình 3.26.

Nhập các tiếp điểm như sau:

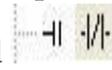
1. Nhấp đúp chuột vào hình tượng Bit Logic  hoặc nhấp chuột vào dấu cộng (+) ở cửa sổ cây lệnh để hiển thị các lệnh trong bit logic. (*nhấp F4*)

2. Chọn tiếp điểm Normally Open 

3. Giữ chuột trái và kéo tiếp điểm vào network đầu tiên.

4. Nhấp chuột vào “???” trên tiếp điểm và nhập vào địa chỉ: I0.1 và sau đó nhấn phím Enter.

5. Tương tự từ bước 2 đến bước 4 nhập địa chỉ I0.0

6. Chọn tiếp điểm Normally Closed  và sau đó nhập vào địa chỉ Q0.1

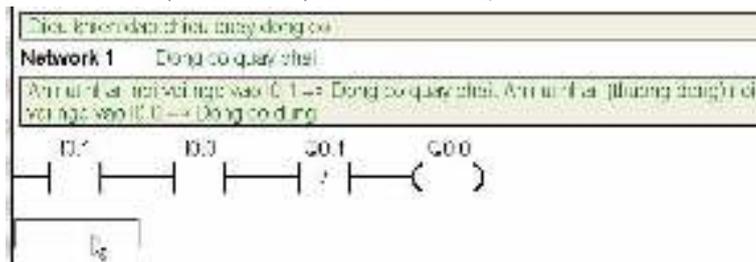
7. Chọn cuộn dây  hoặc (nhấp F6) và nhập vào ở “???” địa chỉ Q0.0

Chú ý: khi gõ các địa chỉ I0.0, I0.1, Q0.0, Q0.1 có thể ta sẽ nhận được kết quả là các địa chỉ theo ký hiệu. Để hiện lại các địa chỉ tuyệt đối ta bỏ kích hoạt **View > Symbolic Addressing**.

Rẽ nhánh Network 1

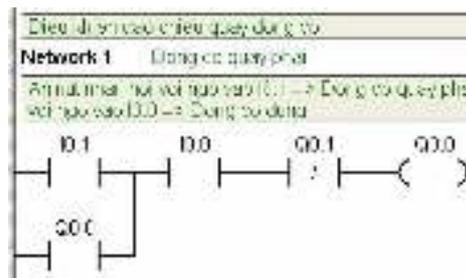
1. Tương chọn tiếp điểm Normally Open  giữ chuột trái và kéo tiếp

điểm vào vị trí con trỏ chuột (hình 3.29) và đặt tên Q0.0.



Hình 3.29: Rẽ nhánh network

2. Để trỏ chuột ở vị trí như hình 3.29 và nhấp chuột vào biểu tượng line up để kết thúc như hình 3.30



Hình 3.30: Rẽ nhánh network

Nhập network 2: Động cơ quay trái

Thực hiện tương tự như network 1.

Bước 4: Lưu chương trình

Sau khi nhập hai network lệnh, ta đã nhập xong chương trình. Khi lưu chương trình, ta tạo một dự án bao gồm loại CPU S7-200 và các tham số khác. Để lưu một dự án, thực hiện như sau:

1. Chọn **File > Save As**
2. Nhập vào tên của dự án trong hộp thoại Save As
3. Nhấp OK để lưu dự án.

Bước 5: Download chương trình vào CPU

Sau khi lưu dự án, ta có thể download chương trình vào S7-200.

Mỗi dự án được liên kết với một loại CPU (CPU221, CPU222, CPU224, CPU 224XP, hoặc CPU 226). Nếu kiểu dự án không phù hợp với CPU đang kết nối, thì STEP 7-Micro/WIN báo lỗi không tương thích và các đường dẫn để ta tiếp tục công việc. Nếu điều này xảy ra, chọn “Continue Download”.

Thực hiện download chương trình như sau:

1. Nhấp chuột vào biểu tượng Download  trên toolbar hoặc chọn **File> Download** để download chương trình.
2. Nhấp OK để download các phần tử chương trình vào S7-200. Nếu S7-200 ở chế độ RUN, một hộp thoại xuất hiện yêu cầu bạn đặt S7-200 ở chế độ STOP. Nhấp chuột vào Yes để đặt S7-200 ở chế độ STOP.

Bước 6: Đặt S7-200 ở chế độ RUN

Đối với phần mềm STEP 7-Micro/WIN để đặt CPU S7-200 vào chế độ RUN, thì công tắc chọn chế độ của S7-200 phải được đặt ở vị trí **TERM** hoặc **RUN**. Khi đặt S7-

200 ở chế độ RUN, thì S7-200 thực hiện chương trình:

1. Nhấp chuột vào biểu tượng RUN  trên toolbar hoặc chọn **PLC > RUN**.
2. Nhấp OK chuyển chế độ hoạt động của S7-200.
3. Khi S7-200 đi vào chế độ RUN thì đèn RUN trên PLC sáng.

Bước 7: Tìm lỗi và chỉnh sửa chương trình

Sau khi CPU đã ở chế độ RUN, ta có thể kiểm tra lại chương trình bằng cách ấn các nút nhấn S_Right, S_Stop, S_Left và quan sát các đèn LED Q0.0 và Q0.1.

Nếu ấn nút nhấn S_Right, thì đèn LED Q0.0 sáng.

Ấn nút S_Stop, thì đèn LED Q0.0 tắt.

Ấn nút S_Left, thì đèn Q0.1 sáng.

Ấn nút S_Stop, thì đèn LED Q0.1 tắt.

Nếu việc kiểm tra không đạt được kết quả như mô tả, thì có thể giám sát chương trình bằng cách chọn **Debug > Program Status**. Dựa vào trạng thái tiếp điểm và các cuộn dây trong chương trình mà tìm ra các lỗi và chỉnh sửa cho phù hợp với yêu cầu công nghệ.

Để dừng chương trình, đặt S7-200 về chế độ STOP bằng cách nhấp chuột vào biểu tượng STOP  hoặc chọn **PLC > STOP**.

3.3 KIỂM TRA VIỆC NỐI DÂY BẰNG PHẦN MỀM

Một công việc quan trọng cho người lắp đặt và vận hành là biết được các kết nối của các ngõ vào/ra với ngoại vi có đúng hay không trước khi nạp chương trình điều khiển vào CPU. Hoặc khi một hệ thống đang hoạt động bình thường nhưng một sự cố hư hỏng xảy ra thì các phần ngoại vi nào bị hư và phát hiện nó bằng cách nào. 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 S7-200) 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 Step7 Micro/WinV4.0 hoặc vào mục View → Component → Status Chart. Giám sát và thay đổi biến với Status Chart.

Hình 3.31 chỉ một ví dụ về cách sử dụng Status Chart. Để đọc hay ghi các biến chúng ta thực hiện theo các bước sau:

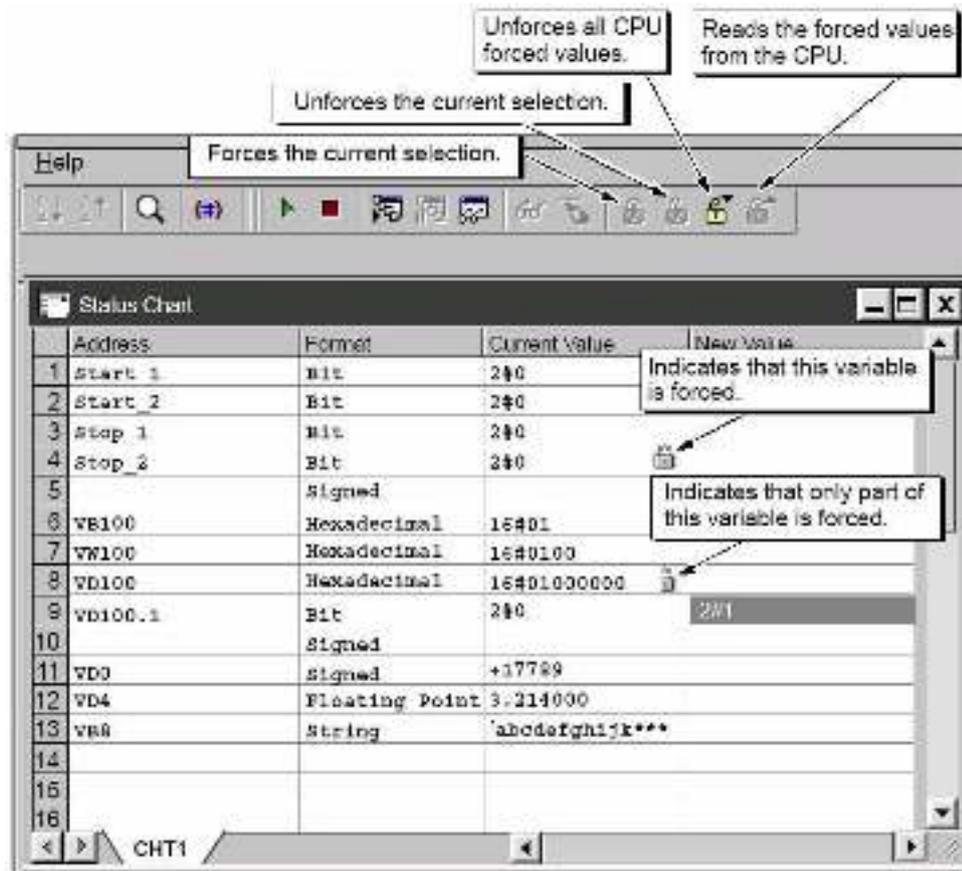
Bước 1: Ở ô đầu tiên trong cột Address ta nhập vào địa chỉ hay tên ký hiệu của một biến trong chương trình ứng dụng mà muốn giám sát hoặc điều khiển, sau đó ấn ENTER. Lặp lại bước này cho tất cả các biến được thêm vào biểu đồ.

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.



Hình 3.31: Ví dụ về Status Chart

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).

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

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

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

Bước 3: Để cường bức biến với giá trị hiện hành, trước tiên hãy đọc giá trị hiện hành trong PLC bằng cách nhấp chuột vào biểu tượng  hoặc chọn Debug → Chart Status. Nhấp hoặc cuộn ô chứa giá trị hiện hành muốn cường bức. Nhấp chuột vào biểu tượng  hoặc chọn Debug → Force ở trên vị trí giá trị hiện hành để cường bức biến giá trị đó.

Bước 4: Để cường bức một giá trị cho một biến, nhập giá trị vào cột “New Value” và nhấp chuột vào biểu tượng  hoặc chọn Debug → Force.

Bước 5: Để xem giá trị hiện hành của tất cả các biến bị cường bức, kích chuột vào biểu tượng Read All Forced  hoặc chọn Debug → Read All Forced.

Bước 6: Để cho tất cả các biến trở lại trạng thái bình thường, hãy kích chuột vào biểu tượng Unforce All  hoặc chọn Debug → Unforce All. Muốn gỡ bỏ cường bức một biến, hãy chọn biến mong muốn và nhấp chuột vào biểu tượng  hoặc chọn Debug → Unforce.

Ứng dụng Status Chart trong việc kiểm tra kết nối dây trong S7-200

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 bị sai và sửa chữa.

3.4 THỰC HÀNH KẾT NỐI DÂY PLC VÀ THIẾT BỊ NGOẠI VI

Thiết bị

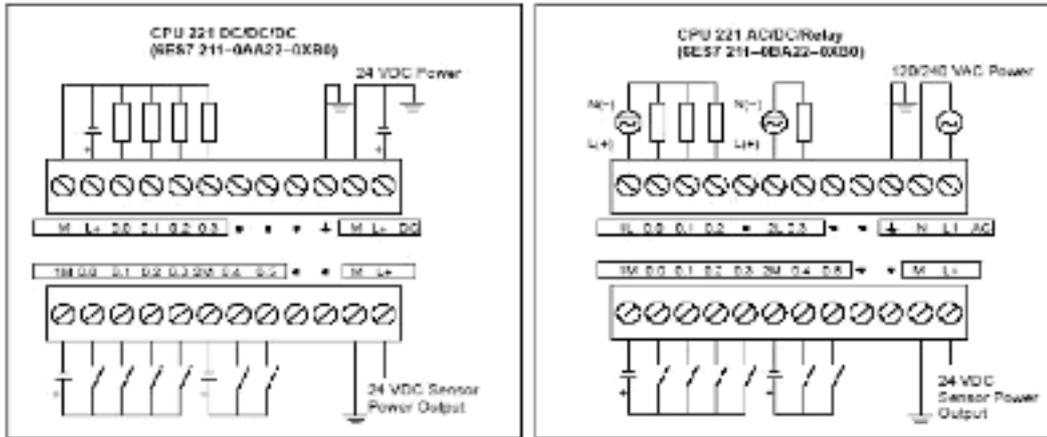
- Máy tính, Cáp PC/PPI loại cổng COM RS232
- Bộ thực hành PLC, Dây nối

- Đồng hồ đo VOM

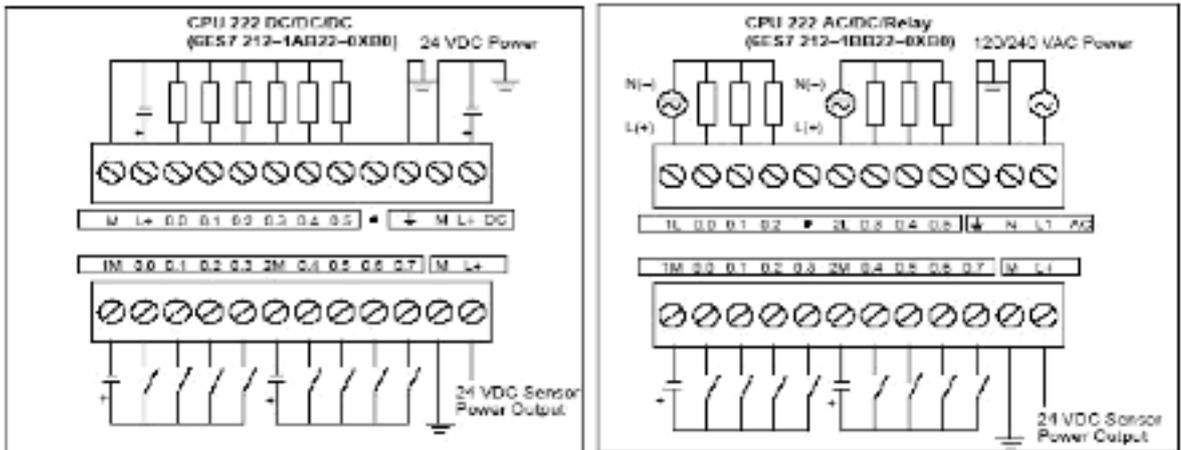
Các bước thực hiện

Bước 1. Xác định các thành phần cơ bản trên PLC

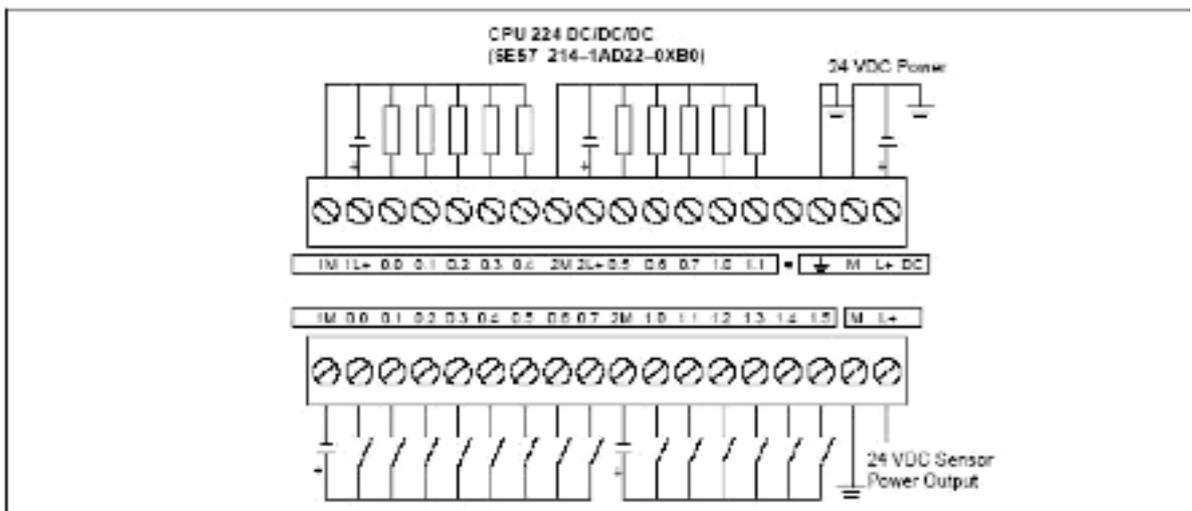
Bước 2. Xác định sơ đồ đấu dây cơ bản cho PLC



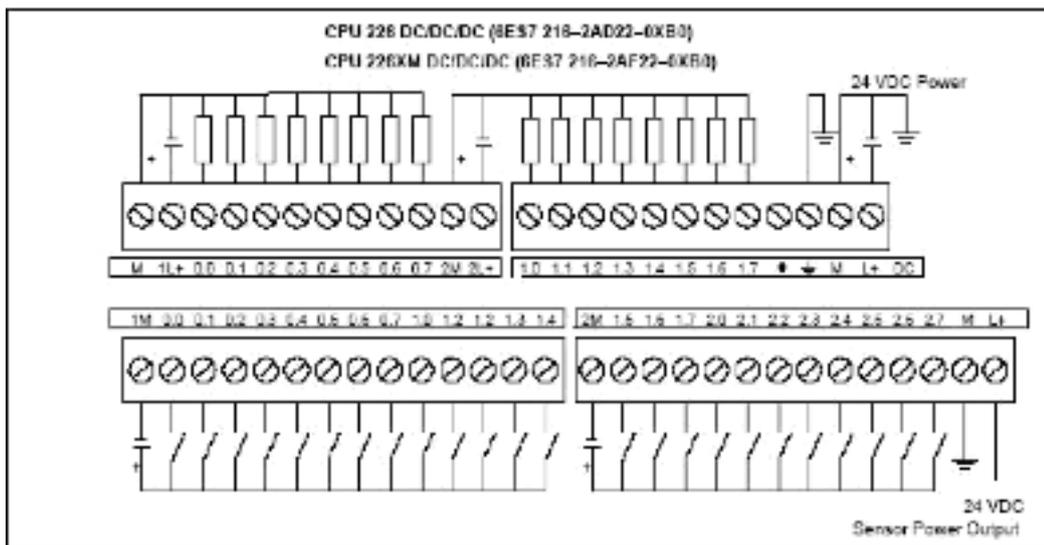
Hình 3.32: Sơ đồ đấu nối CPU 221



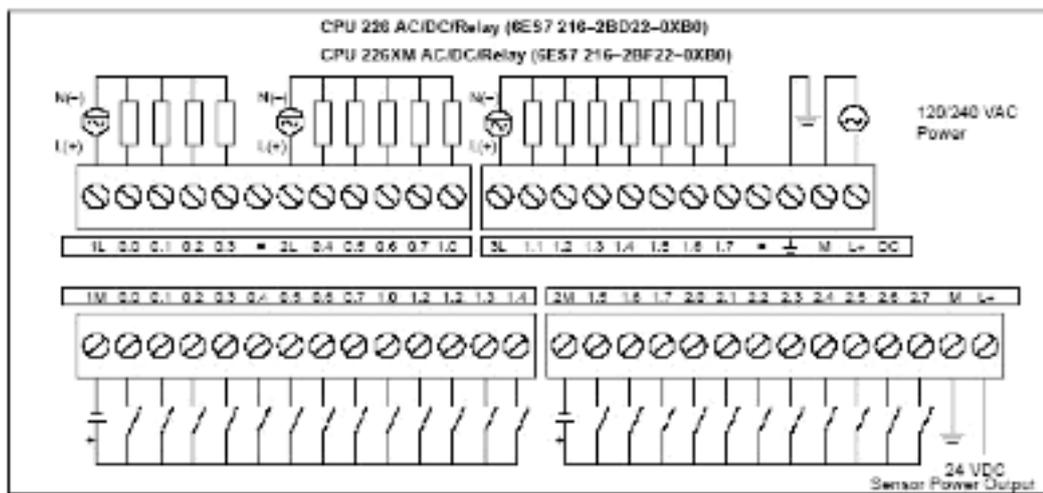
Hình 3.33: Sơ đồ đấu nối CPU 222



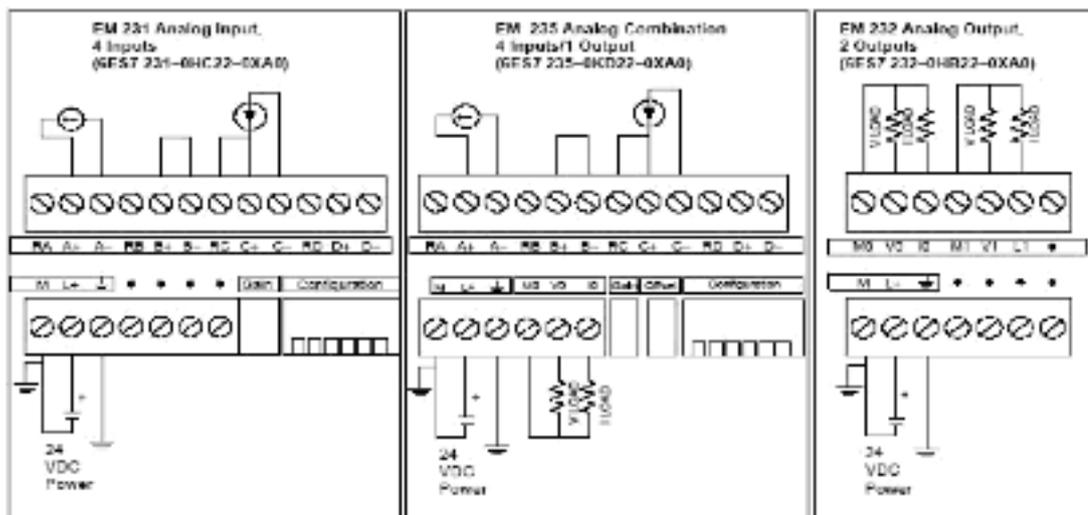
Hình 3.34: Sơ đồ đấu nối CPU 224 DC/DC/DC



Hình 3.35: Sơ đồ đấu nối CPU 226 DC/DC/DC



Hình 3.36: Sơ đồ đấu nối CPU 226 AC/DC/RLY



Hình 3.37: Sơ đồ đấu nối các module mở rộng Analog

Bước 3. Thực hiện đấu nối

- PLC với các module trong bộ thực hành PLC
- Dùng đồng hồ đo và vẽ lại sơ đồ nối dây của PLC với bộ thực tập
- Kết nối dây PLC và thiết bị lập trình.

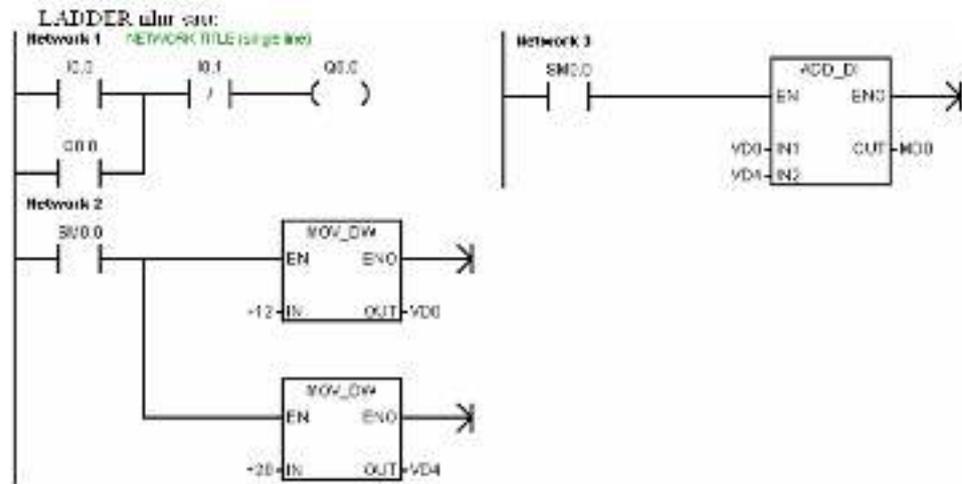
3.5 CÀI ĐẶT VÀ SỬ DỤNG PHẦN MỀM STEP7- MICROWIN

Thiết bị

- Kit PLC.
- PC có phần mềm STEP7-MicroWin32.
- Cáp PC/PPI.
- Dây nối.
- VOM kế.

Các bước thực hiện

Bước 1. Tạo một chương trình điều khiển mới cho S7-200 trên phần mềm Step7-Micro/Win32 có dạng LADDER như sau:



Hình 3.38: Chương trình mẫu

Bước 2. Tạo bảng Symbol như sau:

- I0.0: “PushStart”, I0.1: “PushStop”, Q0.0: “Lamp”
- VD0: “Numer1”, VD4: “Numer2”, MD0: “Result”

Bảng 3.2 Symbol

	Symbol	Address	Comment
1	PushStart	I0.0	nút nhấn start (thường hở)
2	PushStop	I0.1	nút nhấn stop (thường đóng)
3	Lamp	Q0.0	
4	Numer1	VD0	
5	Numer2	VD4	
6	Result	MD0	

Bước 3. So sánh sự thay đổi của chương trình trước và sau khi tạo bảng Symbol.

Nhận xét?

Bước 4. Biên dịch và nạp chương trình vào PLC.

Bước 5. Kiểm tra hoạt động chương trình.

3.6 Bài tập

1. Nêu cách nhận biết loại PLC S7- 200?
2. Trình bày công cụ kiểm tra dây nối bằng phần mềm.
3. Viết chương trình cho PLC điều khiển các đèn như sau:
 - Khi nhấn I0.0 thì đèn Q0.0 sáng.
 - Khi nhấn I0.1 thì đèn Q0.1 sáng.

- Khi nhấn I0.2 thì đèn Q0.2 sáng.
- Khi nhấn I0.3 thì đèn Q0.3 sáng.
- Khi nhấn I0.4 thì đèn Q0.4 sáng.
- Khi nhấn I0.5 thì đèn Q0.5 sáng.
- Khi nhấn I0.6 hoặc I0.7 thì các đèn tắt hết.

Bài 4:

CÁC PHÉP TOÁN NHỊ PHÂN CỦA PLC

GIỚI THIỆU

Hệ nhị phân (hay hệ đếm cơ số 2) là một hệ đếm dùng hai ký tự để biểu đạt một giá trị số, bằng tổng số các lũy thừa của 2. Hai ký tự đó thường là 0 và 1. Sử dụng hệ nhị phân trong lập trình PLC có ưu điểm là dễ dàng thực hiện, tính toán đơn giản.

MỤC TIÊU BÀI HỌC

+ Về kiến thức:

- Trình bày các liên kết Logic;
- Trình bày các lệnh ghi và xoá giá trị tiếp điểm;
- Trình bày được nguyên lý làm việc của Timer, Counter.
- Trình bày nguyên lý làm việc và ứng dụng của timer, Counter trong s7-200;

+ Về kỹ năng:

- Sử dụng thành thạo các lệnh logic, timer, counter;
- Thực hiện các phép toán nhị phân trên PLC đạt yêu cầu kỹ thuật.

+ Về thái độ:

- Đảm bảo an toàn và vệ sinh công nghiệp

NỘI DUNG BÀI HỌC

4.1 CÁC LIÊN KẾT LOGIC

4.1.1 Các phép toán Logic cơ bản

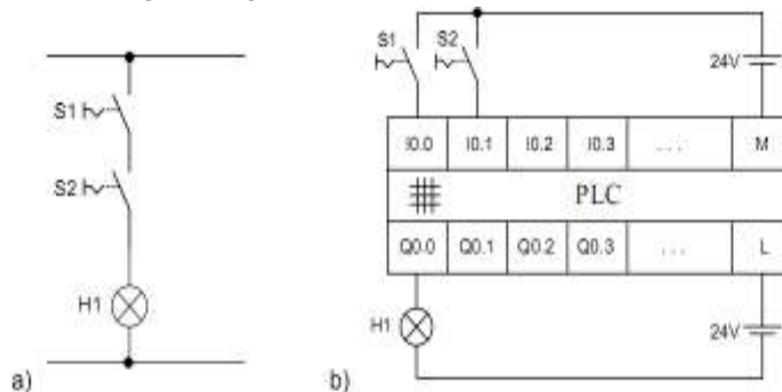
Trong phần này trình bày các phép toán đối với dữ liệu là bit. Trước tiên là phân lý thuyết sau đó tới ví dụ và chương trình. CPU sử dụng trong các ví dụ là loại DC/DC/DC (nguồn cung cấp cho ngõ vào, ra và CPU là 24Vdc).

Bài này chủ yếu trình bày về các phép toán liên quan đến bit hay còn gọi là phép toán nhị phân. Vì vậy khi viết chương trình, ta chỉ lấy các phần tử trong bit logic.

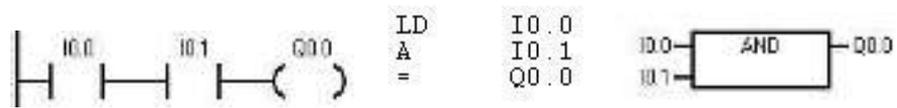
Phép toán AND

Phép toán AND được sử dụng khi có yêu cầu điều khiển là trạng thái của 2 hay nhiều tín hiệu *đồng thời* xảy ra thì sẽ thực hiện một nhiệm vụ điều khiển nào đó (*ngõ ra bằng 1 khi tất cả ngõ vào ở trạng thái 1*). Trong điều khiển có tiếp điểm, phép toán AND tương đương hai tiếp điểm NO mắc nối tiếp.

Ví dụ 1: Đèn H1 sẽ sáng nếu đồng thời cả 2 công tắc S1 và S2 ở trạng thái đóng mạch. Đèn tắt khi 1 trong 2 công tắc hở mạch.



Hình 4.1: a) Sơ đồ tiếp điểm, b) Sơ đồ kết nối PLC

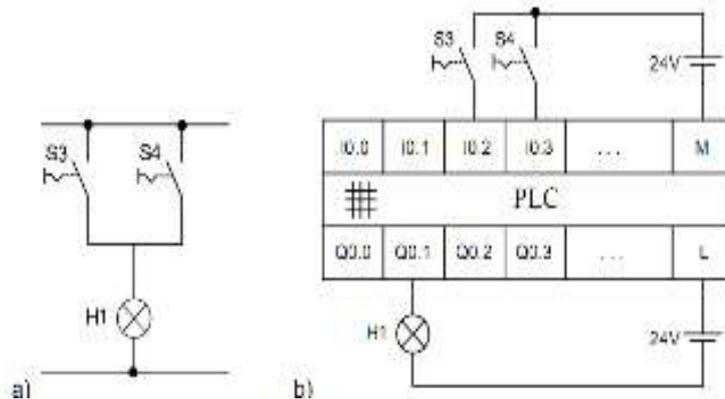


Hình 4.2: Chương trình biểu diễn liên kết And dạng LAD, STL và FBD

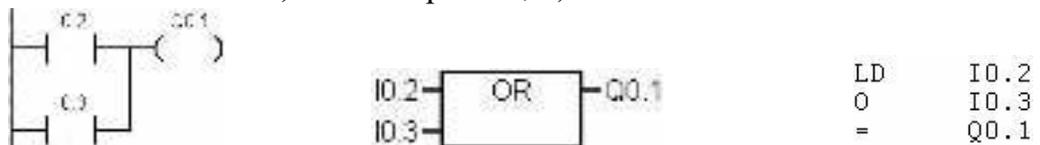
Phép toán OR

Phép toán OR được sử dụng khi có ít nhất một trong các ngõ vào thỏa mãn điều kiện của yêu cầu thì sẽ thực hiện một nhiệm vụ điều khiển nào đó (*ngõ ra bằng 1 khi có ít nhất một ngõ vào ở trạng thái 1*). Trong điều khiển có tiếp điểm, phép toán OR tương đương hai tiếp điểm NO mắc song song.

Ví dụ 2: Có 2 công tắc S3 và S4 đều là thường hở. Hãy viết chương trình sao cho nếu một trong 2 công tắc đóng lại thì đèn H2 sẽ sáng. Đèn tắt khi cả 2 công tắc đều mở.



Hình 4.3: a) Sơ đồ tiếp điểm, b) Sơ đồ kết nối PLC

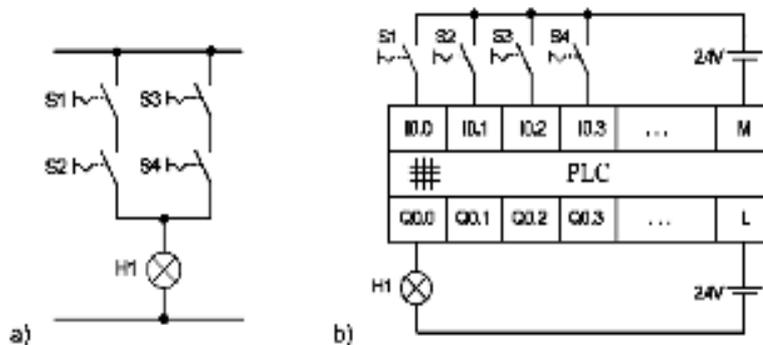


Hình 4.4: Liên kết OR dạng LAD, FBD và STL

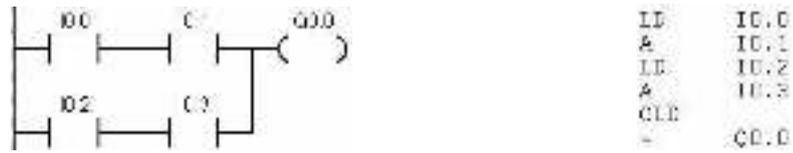
Tổ hợp liên kết And và Or

Trong thực tế, các đối tượng điều khiển phụ thuộc vào một tổ hợp các liên kết logic AND và OR. Tùy theo liên kết nào đứng trước mà sẽ có các lệnh ở STL khác nhau, ở dạng Lad các công tắc ghép song song và nối tiếp hay ngược lại.

Để thực hiện phép OR hai liên kết AND lại với nhau thì trong chương trình viết ở dạng STL phải sử dụng thêm lệnh **OLD**.

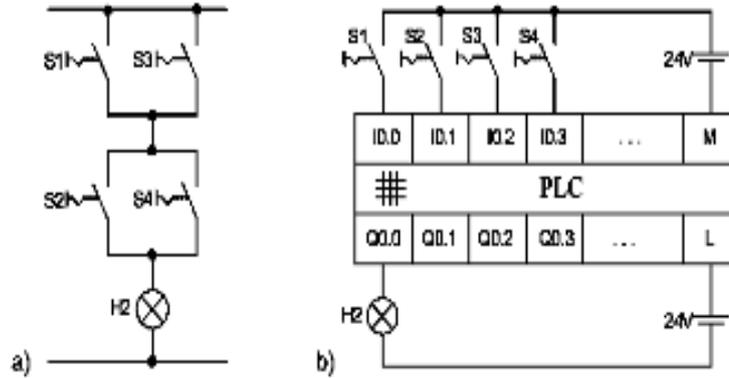


Hình 4.5: Liên kết And trước Or a) Sơ đồ tiếp điểm, b) Sơ đồ kết nối PLC

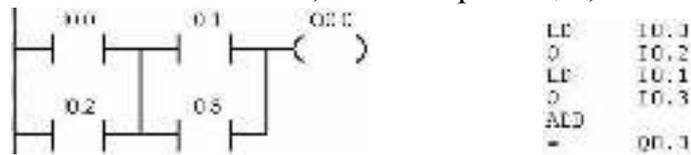


Hình 4.6: Chương trình liên kết And trước Or

Để thực hiện phép AND hai liên kết OR lại với nhau thì trong chương trình viết ở dạng STL phải sử dụng thêm lệnh **ALD**.



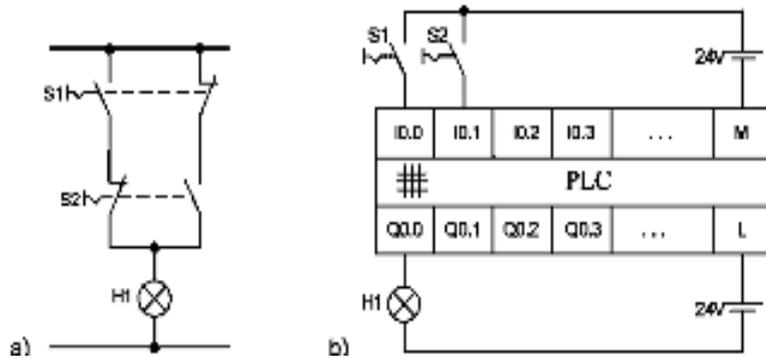
Hình 4.7: Liên kết Or trước And a) Sơ đồ tiếp điểm, b) Sơ đồ kết nối PLC



Hình 4.8: Liên kết Or trước And

Phép toán XOR

Phép toán XOR được sử dụng khi có 2 tín hiệu mà nếu chúng có cùng trạng thái thì ngõ ra sẽ xuống mức 0 còn nếu 2 tín hiệu này khác trạng thái thì ngõ ra sẽ lên mức 1. Đây là phép so sánh khác.



Hình 4.9: Liên kết And Or

a) Sơ đồ tiếp điểm

b) Sơ đồ kết nối PLC

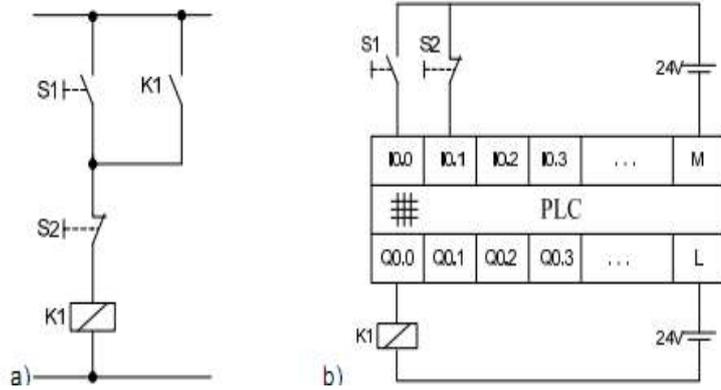


Hình 4.10: Liên kết xor dạng LAD, FBD và STL

BÀI TẬP Thực hành ứng dụng liên kết logic trong điều khiển

Phần này trình bày một số ví dụ ứng dụng nhỏ sử dụng các liên kết logic. Ở một số ví dụ có trình bày mạch điều khiển thông thường với kiểu nối dây khi không dùng PLC để chúng ta thấy sự giống nhau và khác nhau giữa 2 kiểu điều khiển.

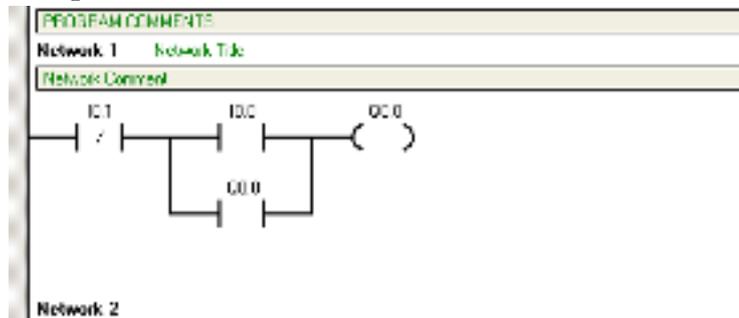
 Mạch duy trì



Hình 4.11: Mạch duy trì ưu tiên Stop a) Sơ đồ tiếp điểm, b) Sơ đồ kết nối PLC

Khi ấn S1 thì dòng điện qua S2 cấp cho cuộn dây K1, tiếp điểm K1 đóng lại, khi nhả S1 thì dòng điện cấp nguồn qua khoá K1 tự duy trì. Khi ấn S2 thì ngắt dòng qua K1 ngưng.

Chương trình nạp cho PLC



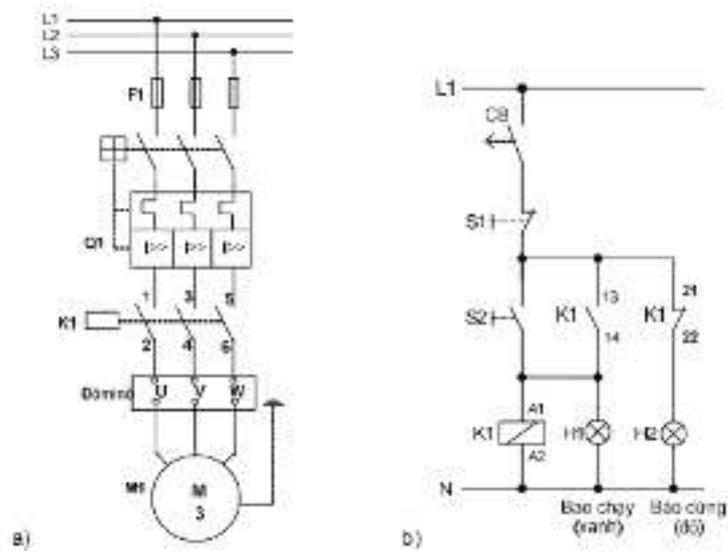
Hình 4.12: Liên kết Logic tạo mạch duy trì

Khi ấn mở I0.0 thì ngõ ra Q0.0 tác động, tiếp điểm của ngõ ra Q0.0 mắc song song I0.0 duy trì cấp điện cho chính nó mặc dù không còn tác động I0.0. Khi ấn dừng I0.1 thì ngõ ra mất điện. Mạch này còn có tên gọi là duy trì ưu tiên dừng máy.

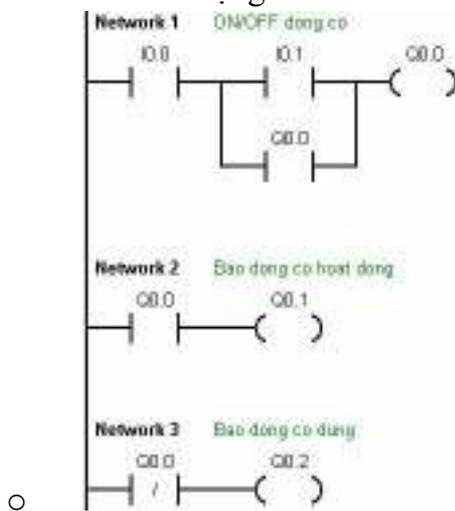
 Điều khiển ON/OFF có báo trạng thái

Khi S2 mở thì đèn đỏ sáng báo dừng, khi S2 đóng thì công tắc tơ K1 có điện mở động cơ, tiếp điểm thường kín K1 mở ra đèn đỏ ngưng, tiếp điểm thường hở K1 đóng lại làm đèn xanh sáng báo chạy hình 4.13.

Bình thường Q0.0 không có điện tiếp điểm thường đóng của nó cấp cho Q0.2 hoạt động báo dừng. Bật I0.0 để sẵn sàng mở máy. Khi ấn mở I0.1 thì ngõ ra Q0.0 tác động, tiếp điểm của ngõ ra Q0.0 mắc song song I0.1 duy trì cấp điện cho chính nó mặc dù không còn tác động I0.0. Tiếp điểm thường đóng của Q0.0 ngắt điện làm Q0.2 dừng.

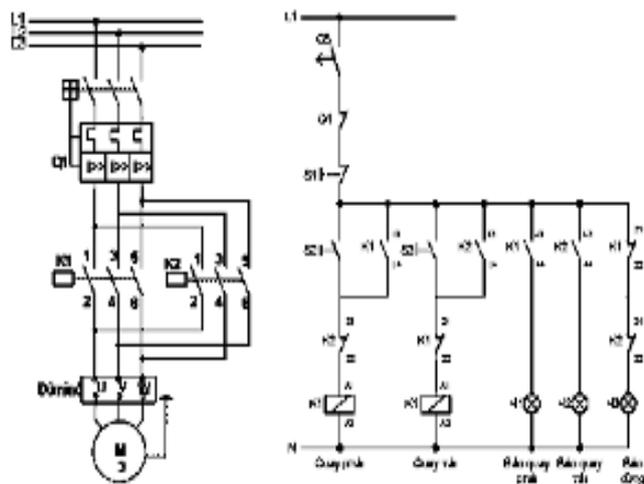


Hình 4.13: Mạch điều khiển động cơ on/off có đèn báo trạng thái

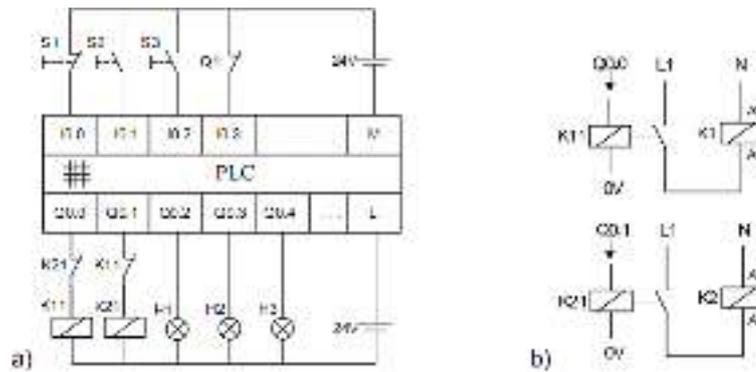


Hình 4.14: Điều khiển động cơ có báo trạng thái

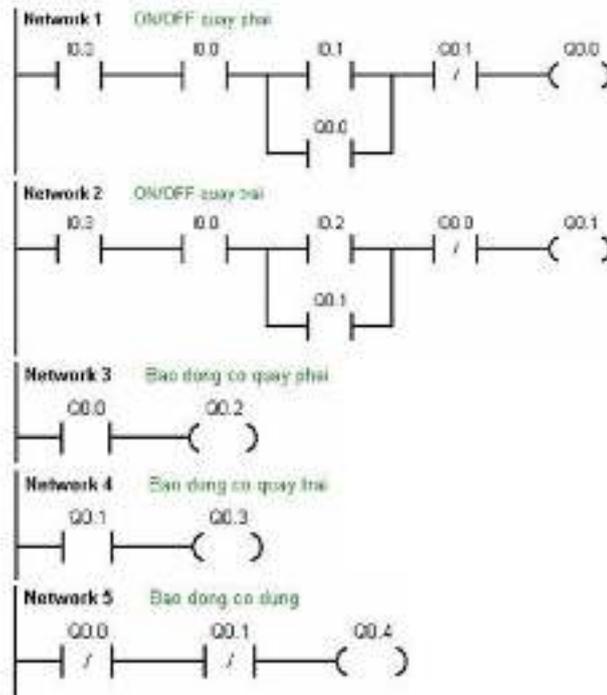
🔧 Điều khiển đảo chiều quay động cơ



Hình 4.15: Mạch đảo chiều quay dùng khí cụ điện



Hình 4.16: Sơ đồ kết nối PLC



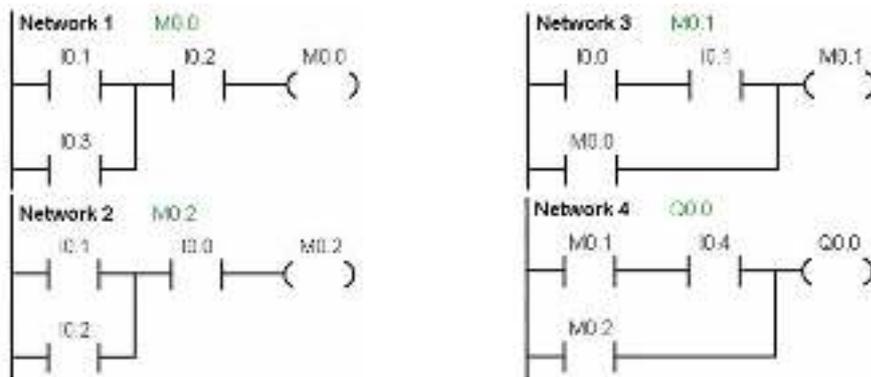
Hình 4.17: Chương trình đảo chiều quay động cơ

Mx.y là các ô nhớ trung gian để lưu kết quả của một phép toán nào đó. Trong S7-200 có 32 byte nhớ M (từ M0.0 đến M31.7). Chúng được xem như là các ngõ ra trung gian. Khi mất nguồn cấp thì nội dung được nhớ trong các bit nhớ M có thể bị mất hoặc vẫn còn giữ lại tùy thuộc vào việc đặt thuộc tính cho vùng nhớ này là retentive (nhớ lâu dài) hay non-retentive (không nhớ lâu dài).

Trong LAD thì bit nhớ này lấy ra và sử dụng như một ngõ ra (coil).

Ví dụ: Mạch logic sử dụng bit nhớ

Khi I0.1 và I0.2 đóng thì bit M0.0 có điện, lúc đó M0.1 cũng có điện, lúc này nếu đóng I0.4 thì ngõ ra Q0.0 có điện. hoặc khi I0.1 và I0.0 có điện thì Q0.0 có điện bởi M0.2 có điện.

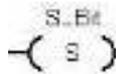


Hình 4.18: Sơ đồ điều khiển Logic sử dụng bit nhớ

4.1.2 Các lệnh ghi / xoá giá trị cho tiếp điểm

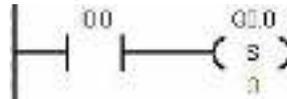
Lệnh ghi giá trị cho tiếp điểm (SET)

Lệnh SET (S) là lệnh thông dụng có trong hầu hết các PLC. Lệnh Set sẽ đặt trạng thái của một hoặc nhiều bit (thuộc vùng nhớ V, M, Q, T, C, SM, L) có địa chỉ liên tục lên mức và duy trì ở trạng thái này cho đến khi bị xóa bằng một lệnh khác. Ta có thể Set một lần tối đa tới 255 bit. Lệnh SET chỉ được thực hiện khi Stack 0 có giá trị logic “1”.



Cú pháp: dạng LAD $\begin{matrix} \text{S_Bit} \\ \text{---} \\ \text{(S)} \\ \text{---} \\ n \end{matrix}$. Với S_Bit là bit đầu tiên của vùng nhớ cần đặt lên mức logic “1” và n là số lượng bit bắt đầu từ S_Bit.

Ví dụ: Chương trình Lad có dạng như sau:



Hình 4.19: Sơ đồ điều khiển set 3 bit lên “1”

Khi tín hiệu tại IO.0 lên mức “1” thì sẽ set 3 bit bắt đầu từ Q0.0 đến Q0.2 lên giá trị logic “1”.

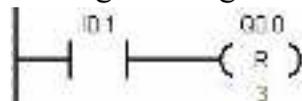
Lệnh xoá giá trị cho tiếp điểm (RESET)

Lệnh RESET (R) cũng là lệnh thông dụng có trong hầu hết các PLC. Lệnh Reset sẽ đặt trạng thái của một hoặc nhiều bit (thuộc vùng nhớ V, M, Q, T, C, SM, L) có địa chỉ liên tục về mức 0. tương tự như lệnh Set chúng ta có thể Reset một lần tối đa tới 255 bit.



Cú pháp: dạng LAD $\begin{matrix} \text{S_Bit} \\ \text{---} \\ \text{(R)} \\ \text{---} \\ n \end{matrix}$. Với R_Bit là bit đầu tiên của vùng nhớ cần đặt về mức logic “0” và n là số lượng bit bắt đầu từ R_Bit.

Ví dụ: Chương trình Lad có dạng như hình 4.20: Khi tín hiệu tại IO.1 lên mức 1 thì sẽ Reset 3 bit từ Q0.0 đến Q0.2 xuống mức logic “0”.



Hình 4.20: Sơ đồ điều khiển reset 3 bit về “0”

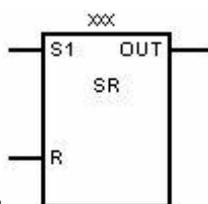
Mạch nhớ R-S

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*.

Bộ ưu tiên set



Ký hiệu

Với: xxx: địa chỉ cần điều khiển

S1: Ngõ vào Set, ký hiệu ưu tiên set

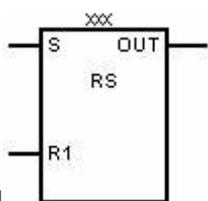
R: Ngõ vào Reset

OUT: Ngõ ra

SR: Ký hiệu khâu R-S

Nếu cả hai ngõ vào cùng lên mức “1” thì ngõ ra là “1”, ưu tiên Set

Bộ ưu tiên Reset



Ký hiệu

Với: xxx: địa chỉ cần điều khiển

S: Ngõ vào Set,

R1: Ngõ vào Reset, ký hiệu ưu tiên Reset

OUT: Ngõ ra

SR: Ký hiệu khâu R-S

Nếu cả hai ngõ vào cùng lên mức “1” thì ngõ ra là “0”, ưu tiên Reset

Các qui tắc khi sử dụng Set và Reset

Khi sử dụng các lệnh S và R trong chương trình PLC cần chú ý các qui tắc sau:

- Các điều kiện làm cho đối tượng điều khiển ở mức tích cực (logic “1”) được sử dụng với lệnh S.

- Các điều kiện làm cho đối tượng điều khiển ở mức không tích cực (logic “0”) được sử dụng với lệnh R.

- Khi viết lệnh S cho một đối tượng điều khiển thì nhất thiết (tùy theo yêu cầu công nghệ) phải có một lệnh R cho đối tượng điều khiển đó.

- Nếu lệnh S được viết trước lệnh R thì kết quả thu được sẽ là kết quả của lệnh R nếu cả hai điều kiện cho S và R cùng ở mức logic “1” nghĩa là đối tượng điều khiển ở mức logic “0”.

- Nếu lệnh R được viết trước lệnh S thì kết quả thu được sẽ là kết quả của lệnh S nếu cả hai điều kiện cho S và R cùng ở mức logic “1” nghĩa là đối tượng điều khiển ở mức logic “1”.

- Khi đã viết chương trình với lệnh S thì không được sử dụng tiếp điểm tự duy trì (loại bỏ tiếp điểm tự duy trì).

- Tùy theo công nghệ khi sử dụng các điều kiện cho lệnh R thì ở trạng thái bình thường các điều kiện này phải có mức logic "0".

➤ Thực hành lệnh Set và Reset

✚ Mạch đảo chiều quay động cơ

Để đơn giản, ví dụ này lấy lại yêu cầu công nghệ của mạch điều khiển đảo chiều quay ở mục trên hình 4.15. Tuy nhiên cần phải sử dụng mạch nhớ RS khi lập trình.

Phân tích: Theo yêu cầu công nghệ ta có các nhận xét sau:

1. Đối với contactor K1 (được đóng điện gián tiếp bởi K11).

$\overline{K2}$ - Điều kiện Set (làm cho K1 có điện): Nút nhấn S2 được ấn. Tuy nhiên vì lý do an toàn K2 mất điện mới được phép mở máy nên phải kết hợp thêm điều kiện K2 mất điện.

$$\text{Set K1} = S1 \wedge \overline{K2}$$

- Điều kiện Reset (làm cho K1 mất điện): Có 2 khả năng là hoặc nút nhấn dừng S1 được ấn hoặc tiếp điểm bảo vệ quá dòng Q1 tác động.

$$\text{Reset K1} = \overline{S1} \vee \overline{Q1}$$

- Vì lý do an toàn, K1 bị mất điện nếu điều kiện SET và RESET cho nó cùng ở logic "1" và sử dụng khâu SR.

2. Đối với contactor K2 (được đóng điện gián tiếp bởi K21)

- Điều kiện Set: Nút nhấn S3 được ấn. Tuy nhiên vì lý do an toàn K1 mất điện mới được phép mở máy nên phải kết hợp thêm điều kiện K1 mất điện.

$$\text{Set K2} = S3 \wedge \overline{K1}$$

- Điều kiện Reset: Có 2 khả năng là hoặc nút nhấn dừng S1 được ấn hoặc tiếp điểm bảo vệ quá dòng Q1 tác động.

$$\text{Reset K2} = \overline{S1} \vee \overline{Q1}$$

- Vì lý do an toàn, K2 bị mất điện nếu điều kiện SET và RESET cho nó cùng ở logic "1" và sử dụng khâu SR.

3. Đối với đèn báo H1.

- Đèn sáng khi K1 có điện và tắt khi K1 mất điện

$$H1 = K1$$

4. Đối với đèn báo H2

- Đèn sáng khi K2 có điện và tắt khi K2 mất điện.

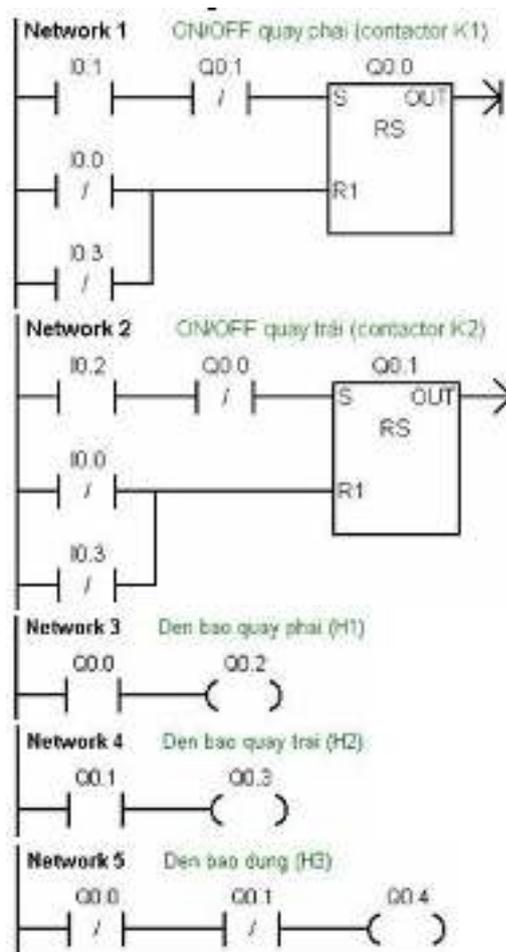
$$H2 = K2$$

5. Đối với đèn báo H3

- Đèn sáng khi cả K1 và K2 mất điện.

$$H3 = \overline{K1} \wedge \overline{K2}$$

Theo các phân tích ta viết được chương trình như sau:



Hình 4.21: Chương trình lad

4.2 TIMER

4.2.1 Giới thiệu

Bộ định thời được sử dụng trong các yêu cầu điều khiển cần trì hoãn về thời gian. Đây là phần tử chức năng cơ bản của các bộ PLC và rất thường được sử dụng trong các chương trình điều khiển. Chẳng hạn như một băng tải khi có tín hiệu hoạt động sẽ chạy trong 10s rồi dừng lại, một van khí nén cần có điện trong 5s, nguyên liệu cần trộn trong thời gian 10 phút.... PLC S7-200 loại CPU 224 có 256 Timer có địa chỉ từ T0 đến T255, chia làm 3 loại:

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

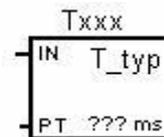
Khi sử dụng một timer chúng ta cần phải xác định các thông số sau:

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

Bảng 4.1 Timer trong PLC S7-200

Lệnh	Độ phân giải	Giá trị cực đại	CPU 221	CPU 222	CPU 224	CPU 226
TON	1ms	32,767s	T32, T96	T32, T96	T32, T96	T32, T96
	10ms	327,67s	T33÷T36 T97÷T100	T33÷T36 T97÷T100	T33÷T36 T97÷T100	T33÷T36 T97÷T100
	100ms	3276,7s	T37÷T63 T101÷T255	T37÷T63 T101÷T255	T37÷T63 T101÷T255	T37÷T63 T101÷T255
TONR	1ms	32,767s	T0, T64	T0, T64	T0, T64	T0, T64
	10ms	327,67s	T1÷T4 T65÷T68	T1÷T4 T65÷T68	T1÷T4 T65÷T68	T1÷T4 T65÷T68
	100ms	3276,7s	T5÷T31 T69÷T95	T5÷T31 T69÷T95	T5÷T31 T69÷T95	T5÷T31 T69÷T95

Ký hiệu chung của Timer trong S7-200 biểu diễn ở LAD như sau:



Txxx: Ký hiệu và số thứ tự của timer,

IN: Ngõ vào bit, cho phép timer hoạt động

PT: Ngõ vào số Integer, hằng số thời gian

T_typ: Cho biết loại Timer. Có thể là TON, TONR hay TOF

*???*ms: Báo độ phân giải của timer, tự động xuất hiện theo *Txxx*.

Thời gian trễ được tính theo công thức

$$\text{Thời gian trì hoãn} = [PT] \times [???ms].$$

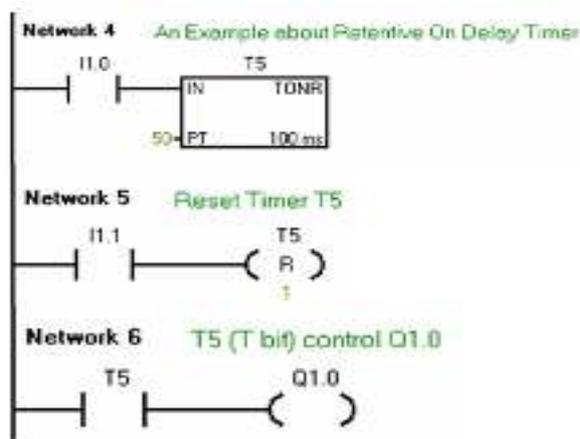
4.2.2 Timer có nhớ TONR

Các Timer này được sử dụng khi cần tích lũy một số khoảng thời gian rời rạc. Giá trị hiện hành TONR chỉ có thể bị xóa bằng lệnh Reset (R).

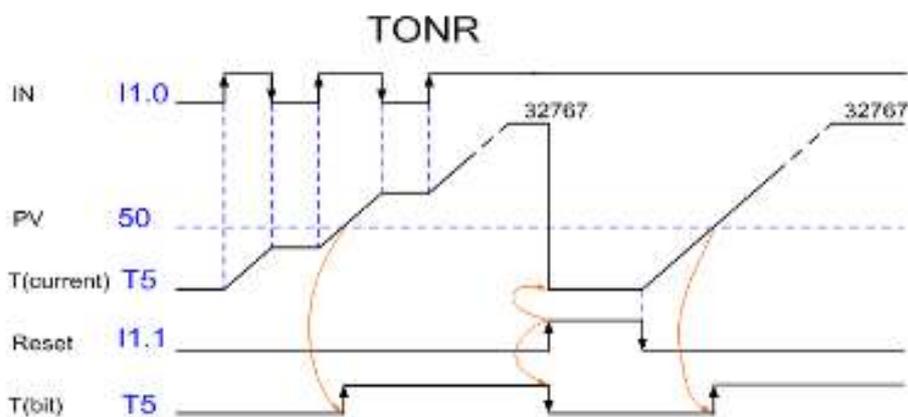
Timer đóng mạch chậm có nhớ TONR (Retentive On-Delay Timer) thực hiện đếm thời gian khi ngõ vào IN ở mức logic “1”. Khi giá trị hiện hành *Txxx* lớn hơn hoặc bằng thời gian đặt trước PT (preset time), thì Timer Bit ở logic “1”. Giá trị hiện hành của TONR được giữ lại khi ngõ vào IN ở logic “0”. TONR được sử dụng để tích lũy thời gian cho nhiều chu kỳ ngõ vào IN ở mức “1”. Timer này vẫn tiếp tục đếm sau khi đã đạt đến giá trị đặt trước và dừng lại ở giá trị max 32767.

Có 64 timer TONR trong S7-200 CPU 224 được phân chia theo độ phân giải như ở bảng 4.1.

Ví dụ sử dụng TONR như chương trình sau:



Hình 4.22: Chương trình sử dụng TONR
 Chương trình hoạt động như mô trong giản đồ thời gian hình 4.23



Hình 4.23: Hoạt động của TONR

Thời gian trễ $T = PT \cdot \text{độ phân giải của } T5 = 50 \cdot 100\text{ms} = 5000\text{ms} = 5\text{s}$

4.2.3 Timer không có nhớ TON (On-Delay Timer)

Timer này được sử dụng khi có yêu cầu trì hoãn một khoảng thời gian sau khi ngõ vào IN tác động lên “1”. Giá trị hiện hành của TON bị xóa khi ngõ vào IN ở logic “0”.

TON thực hiện đếm thời gian khi ngõ vào IN ở mức logic “1”. Khi giá trị hiện hành (Txxx) lớn hơn hoặc bằng thời gian đặt trước PT (pre-set time), thì Timer Bit ở logic “1”. Giá trị hiện hành của TON bị xóa khi ngõ vào IN ở logic “0”. Timer tiếp tục đếm dù đã đạt đến giá trị đặt PT và dừng lại khi đếm đến giá trị cực đại 32767.

Số thứ tự timer TON/TOF trong S7-200 được phân chia theo độ phân giải như ở bảng 4.1:

4.2.4 Timer không có nhớ TOF (Off-Delay Timer)

Sử dụng timer này khi cần trì hoãn thêm một khoảng thời gian rồi mới tắt ngõ ra kể từ khi tín hiệu ngõ vào IN xuống “0”. Timer TOF chỉ thực hiện đếm thời gian khi IN chuyển từ “1” xuống “0”.

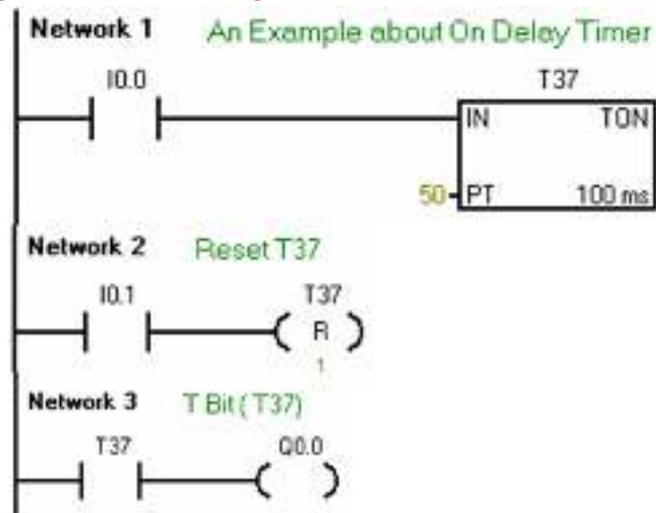
Khi ngõ vào IN của TOF ở logic “1”, thì Timer Bit ngay lập tức được đặt lên mức logic “1” và giá trị hiện hành được xóa về 0. Khi ngõ vào IN xuống “0”, thì timer đếm cho đến khi thời gian trôi qua đạt đến giá trị thời gian đặt trước. Khi đạt đến giá trị đặt trước, Timer Bit được đặt về “0” và giá trị hiện hành dừng đếm. Nếu ngõ vào IN ở

“0” trong khoảng thời gian ngắn hơn giá trị đặt trước, thì Timer Bit giữ ở “1”.

Để xóa timer, có thể sử dụng lệnh Reset (R) sẽ làm cho Timer Bit ở mức logic “0” và giá trị hiện hành của timer (Timer Current) =0.

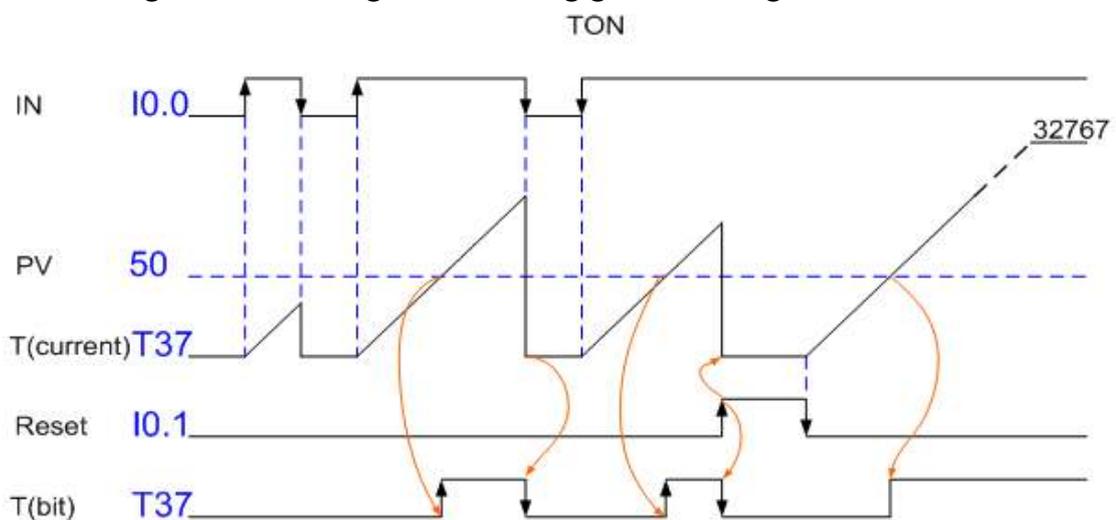
Chú ý: Vì TON và TOF sử dụng cùng loại timer, nên trong một chương trình điều khiển không thể đặt cho cả hai có cùng số Timer. Ví dụ đã đặt TON là T37 thì không được đặt TOF là T37.

Ví dụ sử dụng TON như chương trình sau:



Hình 4.24: Chương trình sử dụng TON

Chương trình hoạt động như mô trong giản đồ thời gian hình 4.25

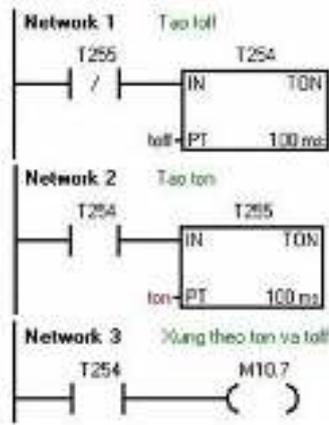


Hình 4.25: Hoạt động của TON

Thời gian trễ $T = PT * \text{độ phân giải của } T37 = 50 * 100ms = 5000ms = 5s$

4.2.5 Thực hành với Timer

✚ Tạo xung có tần số theo mong muốn

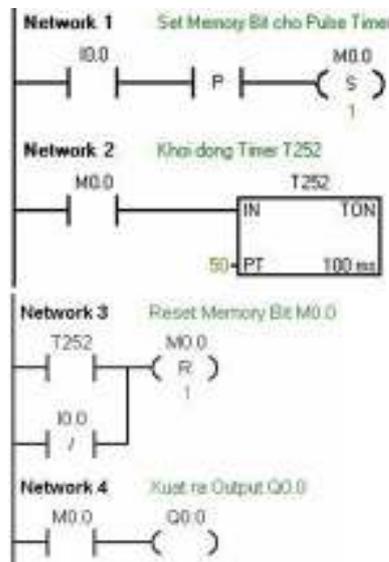


Hình 4.26: Chương trình tạo xung

Chương trình theo hình 4.26 sử dụng 2 timer TON khóa chéo nhau. Tùy thuộc vào xung cần lấy có thời gian t_{on} và t_{off} là bao nhiêu mà ta có thể chọn số timer TON phù hợp. Trong ứng dụng này, chọn T254 và T255 làm timer tạo xung và thời gian thì tùy theo người sử dụng mong muốn cho vào giá trị t_{on} và t_{off} ở ngõ PT của timer. Xung được lưu ở bit M10.7.

Timer xung (Pulse timer)

Timer xung sẽ cho ngõ ra là một xung khi tín hiệu vào ở mức logic “1” có thời gian lớn hơn hay bằng thời gian đặt ở timer xung. Để dễ hình dung xem giản đồ thời gian của chương trình tạo timer xung với ngõ ra timer là Q0.0, ngõ vào tín hiệu là I0.0, thời gian xung là 5s như sau:



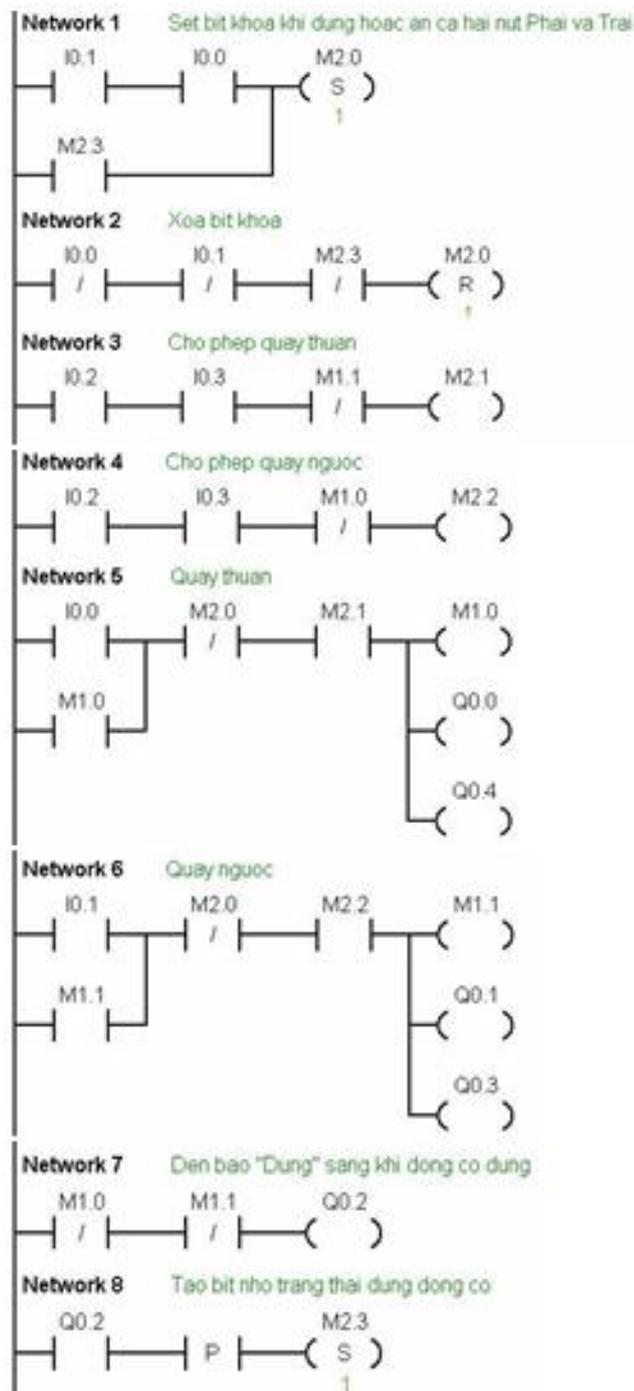
Hình 4.27 Chương trình Timer xung

Đảo chiều quay động cơ có không chế thời gian

Mô tả hoạt động

Một động cơ điện 3 pha có thể đảo chiều quay. Khi ấn nút quay phải “S1” (NO) thì động cơ quay phải, đèn “H1” sáng báo động cơ quay phải. Khi ấn nút nhấn quay trái “S2” (NO) thì động cơ quay trái, đèn “H2” sáng báo động cơ quay trái. Động cơ có thể dừng bất cứ lúc nào nếu ấn nút nhấn dừng “S3” (NC) hoặc xảy ra sự cố quá dòng làm cho tiếp điểm (NC) của thiết bị bảo vệ “Q1” (motor CB) tác động. Khi dừng thì đèn báo “H0” sáng.

Việc đảo chiều quay không thể thực hiện được sau khi nút dừng “S3” được ấn và chưa hết 5s chờ cho động cơ dừng hẳn. Đèn báo chờ đợi “H3” sẽ chớp tắt với tần số 1Hz trong thời gian chờ động cơ dừng hẳn.



Hình 4.28 Chương trình Lad

Phân tích:

- Trong các bài toán điều khiển động cơ, ta cần phải chú ý xem, nếu có sự cố xảy ra với các nút nhấn có làm cho động cơ hoạt động không theo mong muốn hay không. Để đề phòng trường hợp này xảy ra, người lập trình phải tạo ra một khóa.

Đối với mạch đảo chiều quay, có khống chế thời gian dừng (ở đây là 5s) thì khóa sẽ khống chế không cho động cơ khởi động không theo mong muốn cũng như sai chiều quay. Nếu khóa chưa được xóa về 0, thì không thể khởi động hay đảo chiều động cơ

được. Trong bài toán này, khóa xóa về 0 khi cả 2 nút nhấn “S1” và “S2” không được tác động (ở trạng thái bình thường), hoặc thời gian chờ dừng đã hết. Khóa được chọn là M2.0.

- Khi nút nhấn dừng “S3” được ấn, động cơ dừng và phải đợi trong thời gian 5s mới dừng hẳn, nên ta cần nhớ lại trạng thái này trong thời gian 5s để làm điều kiện SET cho khóa M2.0. Chọn memory bit M2.3.

- Để định thời 5s, sử dụng Timer TON. Chọn timer T33

Chương trình ở LAD: Hình 4.28

4.3 CUOUNTER

Trong nhiều trường hợp, việc kiểm tra một số lượng xác định phải thông qua tổng các xung. Có thể thực hiện đếm các xung này bằng các bộ đếm. Sử dụng bộ đếm có thể giải quyết được một số vấn đề sau:

- Đếm số lượng;
- So sánh với một giá trị đặt trước ở các trường hợp bằng nhau, nhỏ hơn, lớn hơn;
- Kiểm tra sự khác biệt về số lượng.

Bộ đếm cũng có thể sử dụng để thực hiện các nhiệm vụ như: Cộng các xung của bộ phát xung nhịp và dựa vào đó để gọi các giai đoạn điều khiển liên tiếp nhau. Hoặc các yêu cầu điều khiển theo chu kỳ lặp như điều khiển đèn giao thông.

Các PLC thường có 3 loại bộ đếm: bộ đếm lên, bộ đếm xuống, bộ đếm lên-xuống. Có 255 bộ đếm ở S7-200 CPU 224 có địa chỉ từ C0 đến C255. Chúng cũng có 3 loại bộ đếm là:

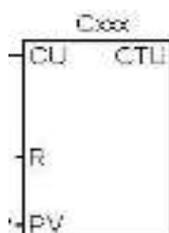
- Bộ đếm lên CTU (Up Counter).
- Bộ đếm xuống CTD (Down Counter).
- Bộ đếm lên-xuống CTUD (Up/Down Counter).

Khi sử dụng một counter chúng ta cần phải xác định các thông số sau:

- Loại counter (CTU, CTD hay CTUD)
- Số của counter sẽ sử dụng, không được gán cùng một số counter cho nhiều counter trong cùng một chương trình.
- Khai báo giá trị cần đếm cho counter.
- Tín hiệu xung cung cấp cho bộ đếm.
- Tín hiệu xóa bộ đếm.

4.3.1 Bộ đếm lên CTU (Count Up)

Bộ đếm CTU được biểu diễn ở LAD như hình:



Cxxx: Ký hiệu và số thứ tự của counter,

CTU: Ký hiệu nhận biết bộ đếm lên

CU: Đếm lên. Ngõ vào bit,

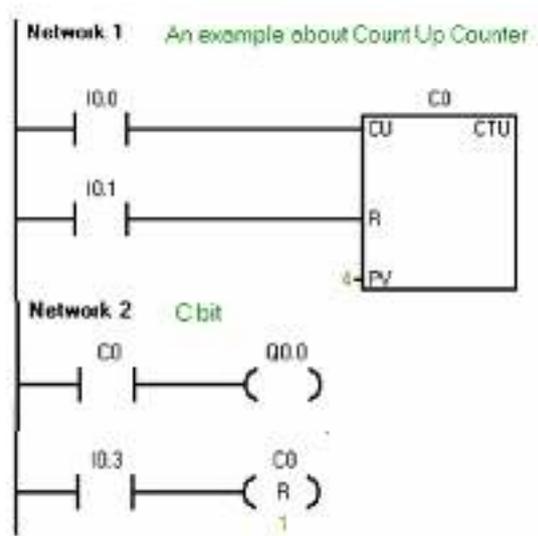
R: Xóa bộ đếm về 0. Ngõ vào bit

PV: Giá trị đặt trước cho bộ đếm. Biểu diễn ở số Integer.

Mỗi khi tín hiệu tại CU từ mức “0” lên “1” thì bộ đếm sẽ tăng giá trị hiện hành của nó lên 1 đơn vị. Khi giá trị hiện hành của bộ đếm (*Cxxx*) lớn hơn hoặc bằng giá trị đặt trước tại ngõ vào *PV* (Preset Value) thì ngõ ra bit của counter (counter bit) sẽ lên mức “1”. Giá trị đếm lên tối đa là 32.767. Phạm vi của bộ đếm là C0 đến C255. Bộ đếm sẽ bị xóa về 0 khi ngõ vào Reset (*R*) lên mức “1”, hoặc khi sử dụng lệnh Reset để xóa bộ đếm.

Ví dụ: Cứ mỗi xung từ “0” chuyển lên “1” tại ngõ vào I0.0, bộ đếm sẽ tăng 1 đơn vị. Từ xung thứ 5 trở đi ngõ ra Q0.0 sẽ lên “1”. Nếu có xung vào tại ngõ I0.1 thì ngõ ra Q0.0 xuống “0”.

Ví dụ về cách sử dụng bộ đếm CTU

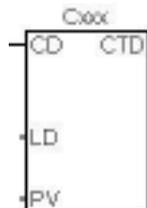


Hình 4.29: Chương trình sử dụng CTU

Chương trình hoạt động như mô trong giản đồ thời gian hình 4.30

4.3.2 Bộ đếm xuống CTD (Count Down)

Bộ đếm xuống CTD được biểu diễn ở LAD như sau:



Cxxx: Ký hiệu và số thứ tự của counter, ví dụ: C20.

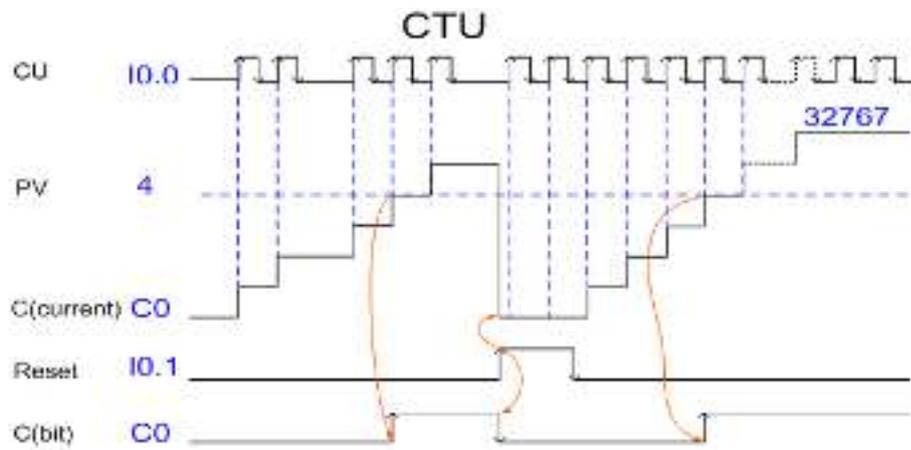
CTD: Ký hiệu nhận biết bộ đếm xuống

CD: Ngõ vào đếm xuống. Ngõ vào bit,

LD: Nạp giá trị đặt trước cho bộ đếm xuống. Ngõ vào bit

PV: Giá trị đặt trước cho bộ đếm. Biểu diễn ở số Integer

Mỗi khi tín hiệu tại CD từ mức “0” lên “1” thì bộ đếm sẽ giảm giá trị hiện hành của nó xuống 1 đơn vị. Khi giá trị hiện hành của bộ đếm (*Cxxx*) bằng 0, thì Counter Bit *Cxxx* lên “1”. Bộ đếm xóa Counter Bit *Cxxx* và nạp giá trị đặt trước ở *PV* khi ngõ vào *LD* (load) lên mức “1”.

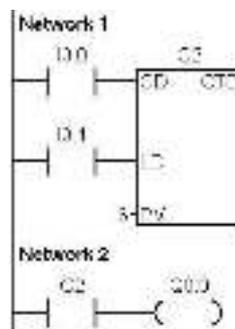


Hình 4.30: Giải đồ hoạt động của CTU

Bộ đếm sẽ dừng đếm khi giá trị hiện hành bằng 0 và counter bit Cxxx lên “1”.

Khi xóa bộ đếm bằng lệnh Reset, counter bit bị xóa và giá trị hiện hành được đặt về logic “0”.

Ví dụ:



Hình 4.31 Chương trình LAD

Sử dụng bộ đếm xuống C2, giá trị hiện hành giảm từ 3 trở về 0. Với I0.1 ở logic “0” và mỗi lần I0.0 chuyển từ “0” lên “1” thì bộ đếm C2 giảm đi một đơn vị. Khi giá trị hiện hành trong bộ đếm C2 bằng 0 thì ngõ ra Q0.0 lên “1”. Khi I0.1 ở “1” thì bộ đếm được đặt trước giá trị đếm là 3.

4.3.3 Bộ đếm lên-xuống CTUD (Count Up/Down)

Bộ đếm xuống CTUD được biểu diễn ở LAD như sau:

Cxxx: Ký hiệu và số thứ tự của counter,.

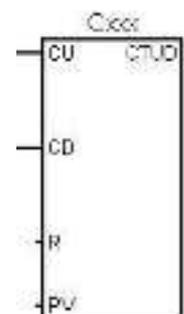
CTUD: Ký hiệu nhận biết bộ đếm xuống

CD: Ngõ vào đếm xuống. Ngõ vào bit,

CU: Ngõ vào đếm lên. Ngõ vào bit

R: Ngõ vào Reset về 0

PV: Giá trị đặt trước cho bộ đếm. Biểu diễn ở số Integer



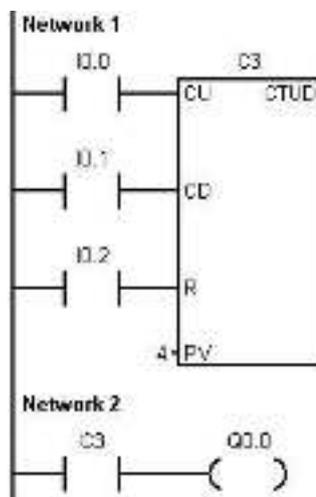
Lệnh đếm lên-xuống (CTUD) sẽ đếm lên mỗi khi ngõ vào đếm lên (CU) từ mức “0” lên “1”, và đếm xuống mỗi khi ngõ vào đếm xuống (CD) chuyển từ “0” lên “1”. Giá trị hiện hành Cxxx giữ giá trị hiện hành của bộ đếm. Giá trị đặt trước PV được so sánh với giá trị hiện hành mỗi khi thực hiện lệnh đếm. Phạm vi của bộ đếm là C0 đến C255.

Khi đạt đến giá trị max (32.767), thì ở cạnh lên kế tiếp tại ngõ vào đếm lên bộ đếm sẽ đặt về giá trị min (-32.768). Khi đạt đến giá trị min (-32.768), thì ở cạnh lên kế

tiếp tại ngõ vào đếm xuống bộ đếm sẽ đặt về giá trị max (32.767).

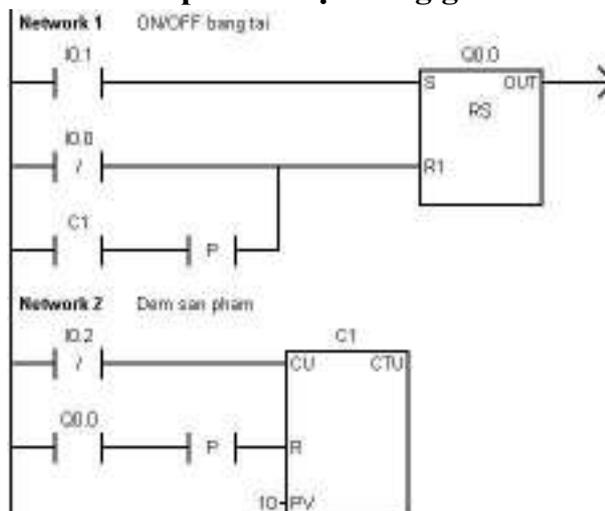
Khi giá trị hiện hành Cxxx lớn hơn hoặc bằng giá trị đặt trước PV, thì Counter Bit Cxxx lên “1”. Ngược lại Counter Bit Cxxx bằng “0”. Bộ đếm sẽ bị xóa về 0 khi ngõ vào Reset (R) lên mức “1”, hoặc khi sử dụng lệnh Reset để xóa bộ đếm.

Ví dụ: Sử dụng bộ đếm xuống C3. Ngõ vào đếm lên nối với I0.0. Ngõ vào đếm xuống nối với I0.1. Xóa bộ đếm bằng I0.2. Khi bộ đếm có giá trị hiện hành ≥ 4 thì ngõ ra Q0.0 lên “1”.



Hình 4.32 Chương trình LAD

➤ **Thực hành bộ đếm đếm sản phẩm được đóng gói**



Hình 4.33 Chương trình LAD

Sản phẩm đã đóng gói được đưa vào một thùng chứa bằng một băng tải (kéo bởi động cơ M). Mỗi thùng chứa được 10 sản phẩm. Khi sản phẩm đã được đếm đủ thì băng tải dừng lại để cho người vận hành đưa một thùng rỗng vào. Sau khi người vận hành ấn nút S1(NO) để tiếp tục thì băng tải hoạt động.

Quá trình cứ lặp đi lặp lại cho đến khi nào ấn nút dừng S0 (NC).

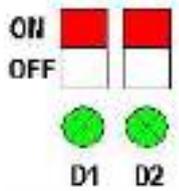
Sản phẩm trước khi đưa vào thùng sẽ đi qua cảm biến quang S2 (NC).

➤ **Bài tập THỰC HÀNH**

ĐIỀU KHIỂN HỆ THỐNG ĐÈN NHẬP NHÁY

Yêu cầu công nghệ:

- Điều khiển hệ thống đèn nhấp nháy gồm 2 đèn:
- Nhấn nút ON: đèn 1 (D1) sáng, sau thời gian 10 giây đèn 2 (D2) sáng (D1 tắt), sau thời gian 10 giây đèn 1 (D1) sáng (D2 tắt), lặp lại liên tục.
- Nhấn nút OFF 2 đèn ngừng hoạt động.

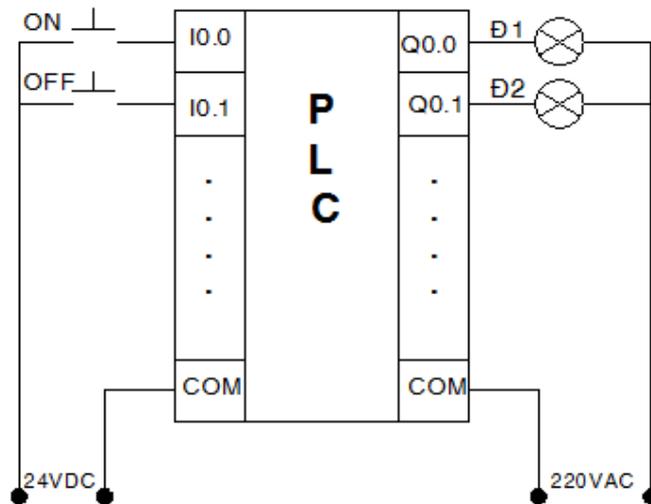


Trình tự thực hành:

- Tìm hiểu cách hoạt động của các đèn:
- Quy định địa chỉ ngõ vào/ra:

Ngõ vào		Ngõ ra	
Địa chỉ	Mô tả	Địa chỉ	Mô tả
I0.0	Nút nhấn ON	Q0.0	Đèn 1
I0.1	Nút nhấn OFF	Q0.1	Đèn 2

- Kết nối PLC với thiết bị ngoại vi:



Hình 4.34 Sơ đồ nối dây PLC

Kết nối thiết bị ngõ vào:

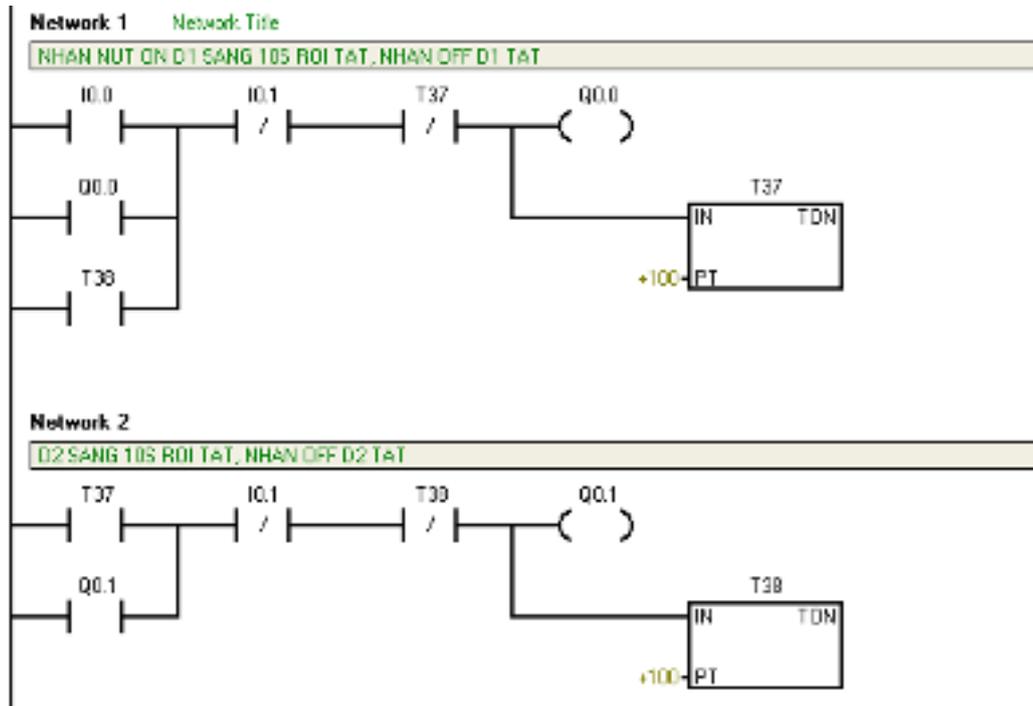
- Nối dây nút nhấn ON với ngõ vào I0.0
- Nối dây nút nhấn OFF với ngõ vào I0.1
- Nối dây đầu còn lại của nút nhấn ON, OFF, với nguồn +24 VDC

Kết nối thiết bị ngõ ra:

- Nối dây điểm A1 của Đ1 với ngõ ra Q0.0
- Nối dây điểm A1 của Đ2 với ngõ ra Q0.1
- Nối dây điểm A2 của Đ1, Đ2 với nguồn 220 VAC
- Nối dây COM của ngõ ra Q0.0 và Q0.1 với cực còn lại của nguồn 220

VAC

- Viết chương trình điều khiển:



Hình 4.35 Chương trình LAD

- Chạy mô phỏng chương trình:

BÀI TẬP NÂNG CAO

1. An toàn cho lò hơi

Một thiết bị lò hơi có hơi đi vào và ra khỏi lò được thực hiện tự động qua bộ điều chỉnh đặt ở bên ngoài. Lò hơi có đặt 3 bộ cảm biến áp suất P1, P2 và P3 ở các vị trí khác nhau để kiểm soát quá áp suất.

Mạch an toàn sẽ hoạt động khi có sự cố, trường hợp áp suất trong lò hơi tăng quá cao thì van an toàn từ tính Y1 sẽ hoạt động xả bớt hơi ra ngoài. Cần có ít nhất bất kỳ hai trong ba cảm biến tác động thì mạch an toàn mở van từ tính Y1. Hãy:

- Viết chương trình sao cho nếu có *bất kỳ 2 trong 3* cảm biến tác động thì van Y1 mở. Một thiết bị lò hơi có hơi đi vào và ra khỏi lò được thực hiện tự động qua bộ điều chỉnh đặt ở bên ngoài. Lò hơi có đặt 3 bộ cảm biến áp suất P1, P2 và P3 ở các vị trí khác nhau để kiểm soát quá áp suất.

Mạch an toàn sẽ hoạt động khi có sự cố, trường hợp áp suất trong lò hơi tăng quá cao thì van an toàn từ tính Y1 sẽ hoạt động xả bớt hơi ra ngoài. Cần có ít nhất bất kỳ hai trong ba cảm biến tác động thì mạch an toàn mở van từ tính Y1. Hãy:

- Viết chương trình sao cho có *bất kỳ 2 trong 3* cảm biến tác động thì van Y1 mở.

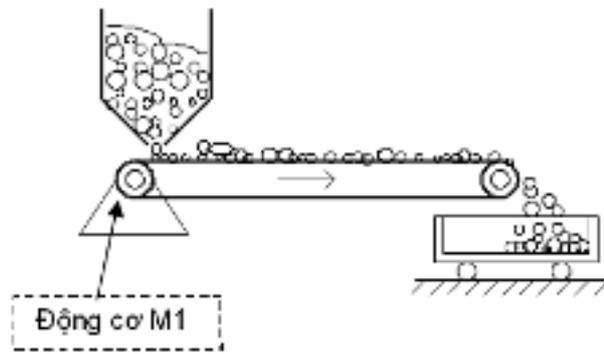
Bảng 4.5 Xác lập ngõ vào/ra

			Symbol	Address	Comment
1			Sensor_P1	I0.0	Tiếp điểm NC
2			Sensor_P2	I0.1	Tiếp điểm NC
3			Sensor_P3	I0.2	Tiếp điểm NC
4			Van_Y1	Q0.0	
5					



2. Bảng tải chuyển vật liệu

- Sơ đồ công nghệ:



Hình 4.36: Mô tả hệ thống

Một thiết bị băng tải dùng để chuyển vật liệu từ thùng chứa vào xe gòn. Hãy viết chương trình sao cho:

- Khi bật khởi động S0 (NO), thì đèn H0 sáng báo hệ thống sẵn sàng làm việc.
- Khi nhấn nút S1 (NO) động cơ M1 chạy kéo băng tải và nguyên liệu trong thùng chứa được vận chuyển theo băng tải.
- Khi nhấn nút dừng S2 (NC) thì băng tải dừng lại.
- Khi có sự cố quá dòng (tiếp điểm nhiệt F3 (NC) tác động) thì động cơ sẽ dừng lại.

Bảng 4.6 Xác lập ngõ vào/ra

		Symbol	Address	Comment
1		S0	I0.0	Cong tac khai dong
2		S1	I0.1	Start
3		S2	I0.2	Stop
4		F3	I0.3	Tiep diem nhiet NC
5		H1	Q0.0	Den bao san sang
6		M1	Q0.1	Contacto Motor M1

3. Điều khiển công ra vào

Một cổng ở công ty cần được điều khiển ở 2 chế độ tay và tự động nhờ một công tắc chọn S0 có 2 vị trí:

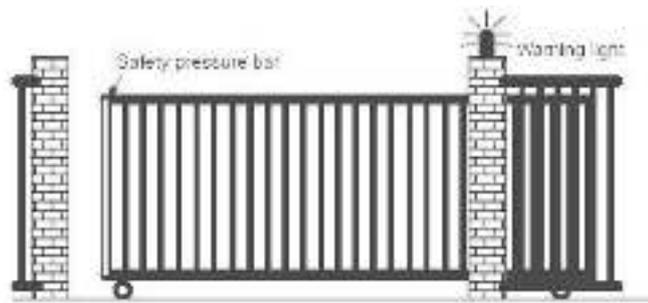
- Ở chế độ tay: Nhấn nút mở S1 (NO) thì động cơ M1 quay phải và cổng mở ra, nếu thả tay ra thì động cơ dừng lại. Tuy nhiên, nếu cổng mở ra đụng công tắc hành trình giới hạn mở S3 (NC) thì cũng dừng lại. Tương tự, nếu nhấn nút đóng S2 (NO) thì động cơ M1 quay trái và cổng đóng lại, nếu thả tay ra thì động cơ dừng lại. Nếu đụng công tắc hành trình giới hạn đóng S4 (NC) thì cổng cũng dừng lại.

- Ở chế độ tự động: Nhấn nút mở thì cửa sẽ mở cho tới khi đụng công tắc hành trình giới hạn mở S3 mới dừng lại. Khi nhấn nút đóng, cổng sẽ đóng lại cho tới khi đụng công tắc hành trình đóng S4 mới dừng lại.

- Có thể dừng quá trình đóng hoặc mở bất cứ lúc nào nếu nhấn nút dừng S5 (NC) hoặc động cơ bị quá tải (tiếp điểm nhiệt F3 (NC) tác động).

- Trong quá trình đóng hoặc mở một đèn báo H1 sẽ sáng lên báo cổng đang hoạt động. Hãy:

- Vẽ sơ đồ nối dây phần cứng
- Sơ đồ công nghệ:



Hình 4.37: Mô tả hệ thống

Bảng 4.7 Xác lập ngõ vào/ra

			Symbol	Address	Comment
1			S0	I0.0	= 0 là che do tay, = 1 là che do tu dong
2			S1	I0.1	Mo cua
3			S2	I0.2	Dong cua
4			S3	I0.3	Cong tac hanh trinh mo
5			S4	I0.4	Cong tac hanh trinh dong
6			S5	I0.5	Stop
7			F3	I0.6	Qua tai
8			H1	Q0.0	Bao hoat dong
9			K1	Q0.1	Contacto cho M1 quay phai (mo cua)
10			K2	Q0.2	Contacto cho M1 quay trai (dong cua)

Bài 5: CÁC PHÉP TOÁN SỐ CỦA PLC

GIỚI THIỆU

Để có thể thực hiện được một chương trình điều khiển PLC phải có tính năng như một máy tính, nghĩa là phải có một bộ vi xử lý (CPU), một hệ điều hành, bộ nhớ để lưu chương trình điều khiển dữ liệu và các cổng vào /ra để giao tiếp được với đối tượng điều khiển và để trao đổi thông tin với môi trường xung quanh. Bên cạnh đó nhằm phục vụ bài toán điều khiển số PLC còn phải có thêm các khối chức năng đặc biệt như bộ đếm (Counter), bộ thời gian (Time)... và những khối hàm chuyên dụng.

MỤC TIÊU CỦA BÀI

+ Về kiến thức:

- Trình bày được nguyên lý hoạt động các phép toán số của PLC theo nội dung đã học.

+ Về kỹ năng:

- Kiểm tra, xử lý chức năng toán số của PLC đạt yêu cầu kỹ thuật.

+ Về thái độ:

- Đảm bảo an toàn và vệ sinh công nghiệp

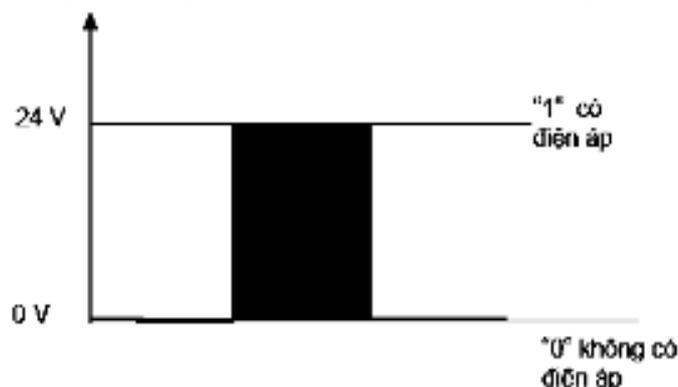
NỘI DUNG BÀI HỌC

5.1 CHỨC NĂNG TRUYỀN DẪN

5.1.1 Truyền Byte ,Word, Doubleword

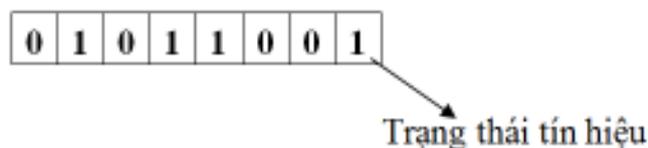
Trong PLC, hầu hết các khái niệm trong xử lý thông tin cũng như dữ liệu đều được sử dụng như Bit, Byte, Word và Doubleword.

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

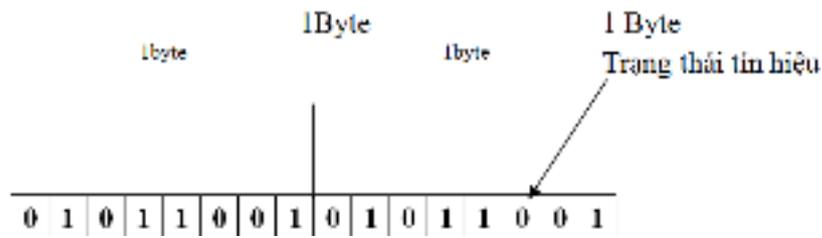


Hình 5.1: Biểu diễn số nhị phân

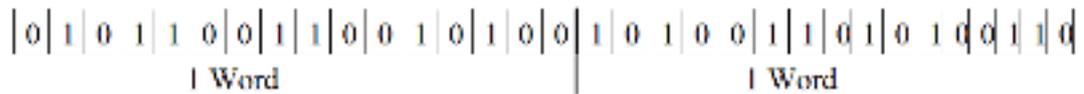
Byte: 1 Byte gồm có 8 Bit



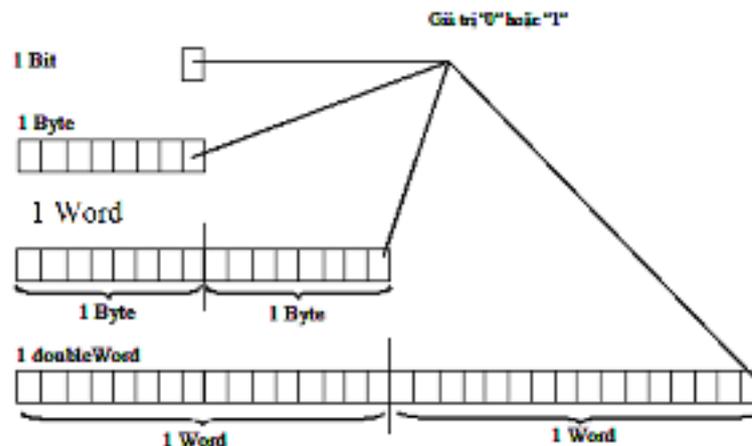
Word: 1 Word gồm có 2 Byte hay 16 Bit. Với Word có thể biểu diễn các dạng: số nhị phân, ký tự hay câu lệnh điều khiển.



DoubleWord: 1 DoubleWord gồm có 4 Byte hay 32 Bit. Với DoubleWord có thể biểu diễn



Tóm tắt



Hình 5.2: Tóm tắt các kiểu dữ liệu

5.1.2 Truyền một vùng nhớ dữ liệu

Kiểu dữ liệu Integer (INT)

Giá trị kiểu dữ liệu *Integer* hoàn toàn là giá trị số không có dấu chấm thập phân. S7-200 lưu trữ giá trị dữ liệu kiểu *Integer* có dấu ở mã 16 bit. Phạm vi của số integer là -32768 đến +32767. STEP 7 sử dụng dạng hiển thị *Decimal* (không phải BCD) để xác định các hằng số của kiểu dữ liệu *Integer*. Nó cũng được mô tả ở dạng có dấu và không dấu. Theo nguyên lý thì có thể sử dụng các giá trị integer hằng số biểu diễn ở dạng *Binary* và *Hexadecimal*, nhưng vì không rõ ràng, nên chúng không còn phù hợp nữa. Vì lý do này, cú pháp của STEP7 chỉ cung cấp giá trị của integer biểu diễn ở decimal.

Trong hệ thống máy tính số, tất cả các giá trị được lưu trữ ở dạng mã binary. Chỉ các số 0 và 1 được sử dụng trong hệ thống số nhị phân. Cơ số 2 của hệ thống số này là kết quả từ số của các số có giá trị. Giá trị của mỗi vị trí của số nhị phân là kết quả của lũy thừa của cơ số 2. Nó được biểu diễn ở dạng $2^{\#} \dots$. Giá trị số âm là sự biểu diễn các số nhị phân ở dạng bù hai. Trong dạng biểu diễn này, bit có trọng số lớn nhất (most significant bit) (bit số 15 cho kiểu dữ liệu Integer) có giá trị -2^{15} . Vì giá trị này lớn hơn tổng của tất cả các giá trị còn lại, nên bit này được làm bit thông tin dấu. Nếu bit = 0, thì giá trị dương; nếu bit = 1, thì giá trị là âm. Việc chuyển đổi giữa các số nhị phân thành số decimal được thực hiện bằng cách cộng các giá trị của các vị trí có bit = 1

Hệ thống số hexadecimal cung cấp 16 chữ số khác nhau (0 đến 9 và A đến F).

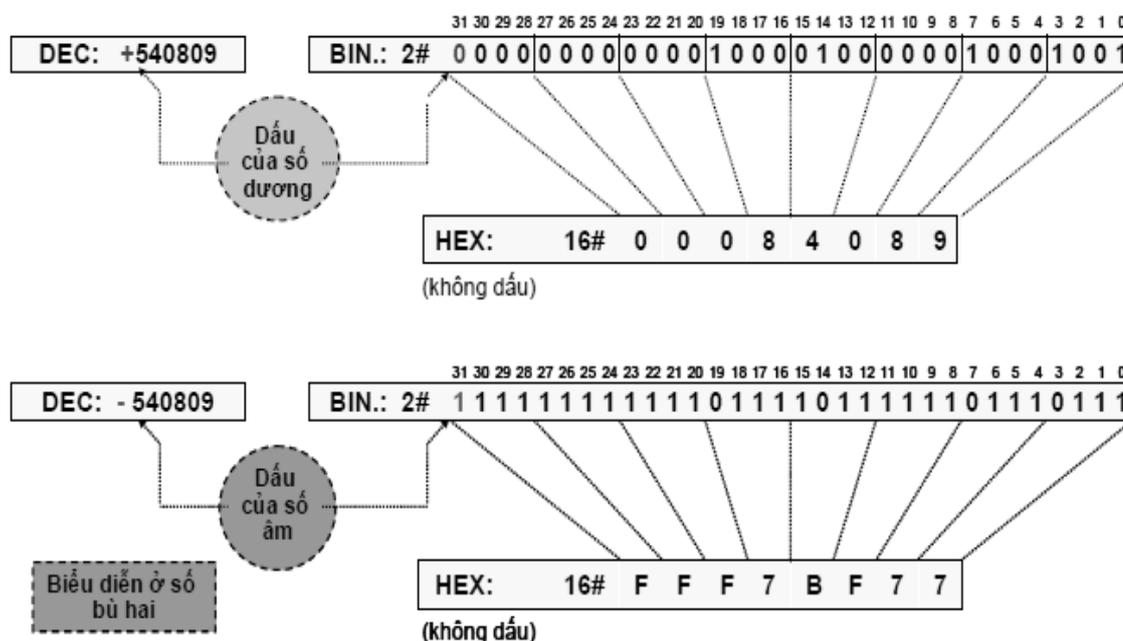
Đây là hệ thống số theo cơ số 16. Do đó, giá trị mỗi vị trí của số hexadecimal có kết quả từ lũy thừa của cơ số 16.

Các số Hexadecimal được xác định với dạng 16#. Các chữ số A đến F biểu diễn theo giá trị số decimal 10 đến 15. Giá trị 15 là giá trị cuối cùng có thể được mã hóa nhị phân của 4 bit không dấu. 4 bit nhị phân tạo thành một số của số hexadecimal. Hàng số trong dạng số Hexadecimal không được sử dụng cho các giá trị số integer.

Kiểu dữ liệu *Double Integer (DINT)*

S7-200 lưu giá trị kiểu dữ liệu *Double Integer* với mã 32 bit có dấu. Phạm vi giá trị kiểu double Integer từ -2147483648 đến +2147483647. S7-200 sử dụng số decimal (không phải BCD) để xác định một hàng số kiểu dữ liệu *Double Integer*.

Ví dụ: Biểu diễn số +540809 và -540809



Hình 5.3: Biểu diễn số double integer

Kiểu dữ liệu số thực (*REAL*)

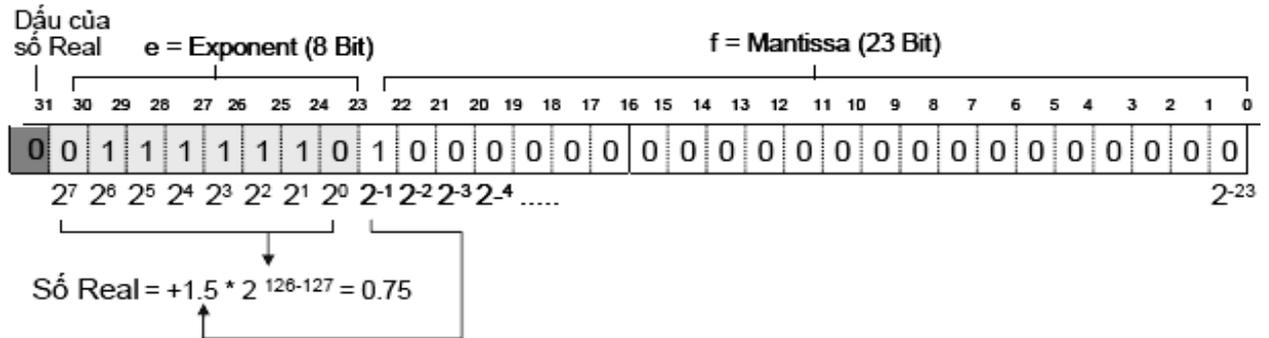
Các kiểu dữ liệu INT và DINT được mô tả trước được sử dụng để lưu toàn bộ các giá trị số có dấu. Do đó, chỉ có các phép toán được cung cấp các giá trị số nguyên mới có thể thực hiện được. Trong trường hợp các biến là analog như điện áp, dòng điện, và nhiệt độ thì các giá trị thực trở nên cần thiết. Để trình diễn các giá trị thập phân, các số nhị phân phải được định nghĩa là giá trị của nó nhỏ hơn 1 (lũy thừa của cơ số 2 với số mũ âm).

Để biểu diễn số thực S7-200 sử dụng double word (32 bit). Trong mã nhị phân của số thực, một phần của các chữ số nhị phân sử dụng cho phần thập phân, phần còn lại là để biểu diễn số mũ và dấu của số thực. Phạm vi biểu diễn của số thực từ $-1.175495 \cdot 10^{-38}$ đến $3.402823 \cdot 10^{+38}$. Khi sử dụng các giá trị của số thực, ta không cần phải xác định dạng của nó. Khi nhập vào một hàng số là số thực thì ta bắt buộc phải nhập có thành phần thập phân cho dù phần thập là số 0, ví dụ 20.0. Số thực được sử dụng để „xử lý giá trị analog. Ưu điểm lớn của số thực là các phép toán được sử dụng với nó. Các phép toán này bao gồm: cộng, trừ, nhân, chia cũng như các lệnh sin, cos,

exp, ln, v.v..., được sử dụng chính trong các thuật giải điều khiển vòng kín (closed-loop control algorithms).

Dạng tổng quát của số Real = (dấu) • (1.f) • (2^{e-127}), với f: phần thập phân.

Ví dụ: Biểu diễn số 0.75



Hình 5.4: Biểu diễn số real

Kiểu dữ liệu số BCD (Binary Coded Decimal)

Trước đây, để liệt kê và mô tả các số nguyên được thực hiện đơn giản với các nút nhấn số dạng xoay vòng và bộ chỉ thị số. Các nút nhấn số và hiển thị số này được kết nối với các module vào và ra số của PLC. Mỗi chữ số của số decimal được mã hóa ở bốn bit. Vì chữ số cao nhất của decimal là 9 nên bốn bit được sử dụng và có mã nhị phân tương ứng cho các chữ số decimal như sau:

Bảng 5.1 Mã thập phân và BCD

Số Decimal	BCD Code	Số Decimal	BCD Code
0	0000	6	0110
1	0001	7	0111
2	0010	8	1000
3	0011	9	1001
4	0100	10	Không có
5	0101	... 15	

Để các số âm cũng có thể được xác định bằng nút nhấn số xoay vòng mã BCD, thì S7-200 mã hóa dấu trong bit có trọng số cao nhất (most significant bit). Bit dấu = 0 để chỉ số dương. Bit dấu = 1 chỉ thị số âm. S7-200 chấp nhận các số BCD mã 16-bit (dấu + 3 digits) và mã 32-bit (dấu + 7 digits). Phạm vi biểu diễn của số BCD 16 bit từ -999 đến +999, phạm vi biểu diễn của số BCD 32 bit từ -9999999 đến +9999999. Không có định dạng dữ liệu cho việc xác định các giá trị theo mã BCD trong S7-200. Tuy nhiên ta có thể xác định số decimal với mã BCD được cho ở số HEX. Mã nhị phân của số HEX và số decimal mã BCD thì giống nhau.

5.2 CHỨC NĂNG SO SÁNH

Với chức năng so sánh, giá trị của hai toán hạng của cùng kiểu dữ liệu sẽ được so sánh với nhau. Kết quả của so sánh là một giá trị logic, nếu đúng theo chức năng so sánh thì kết quả logic là “1”, còn nếu sai kết quả logic là logic “0”. Tùy thuộc vào loại CPU của họ S7-200 mà có thể có ít hoặc nhiều chức năng so sánh. Các chức năng so sánh đối CPU 22x có thể là:

Bảng 5.2 Các thuật toán so sánh

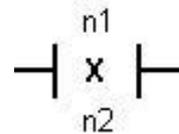
Toán hạng 1 (n_1)	Chức năng so sánh	Toán hạng 2 (n_2)
Dữ liệu có thể là: <i>Byte, Int, DInt, Real</i>	> : Lớn hơn >= : Lớn hơn hoặc bằng == : Bằng nhau <> : Không bằng nhau (khác nhau) <= : Bé hơn hoặc bằng < : Bé hơn	Dữ liệu có thể là: <i>Byte, Int, DInt, Real</i>

Khi so sánh giá trị Byte (B) thì không cần phải để ý đến dấu của toán hạng, ngược lại khi so sánh là các số Int (I), Dint (D), Real (R) thì phải chú ý đến dấu của toán hạng.

Cú pháp tổng quát cho phép toán so sánh ở LAD là:

Trong đó:

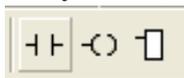
- X: là phép so sánh. Nó có thể là:
- + So sánh byte: >B, >=B, ==B, <>B, <B, <=B
- + So sánh số Int: >I, >=I, ==I, <>I, <I, <=I
- + So sánh số Dint: >D, >=D, ==D, <>D, <D, <=D
- + So sánh số Real: >R, >=R, ==R, <>R, <R, <=R
- n1: Giá trị cần được so sánh (giá trị chưa biết).
- n2: Giá trị so sánh (giá trị đã biết).



Đối với ngôn ngữ LAD và FBD, khi kết quả so sánh là đúng, thì lệnh so sánh sẽ đặt tiếp điểm (LAD) hoặc ngõ ra (FBD) ở trạng thái “ON”.

Đối với ngôn ngữ STL, khi kết quả so sánh là đúng thì lệnh so sánh Load, AND, hoặc OR giá trị 1 với giá trị ở đỉnh của ngăn xếp.

Để lấy các lệnh so sánh ở màn hình soạn thảo LAD, ta nhấp chuột vào biểu tượng



contact trên thanh công cụ hoặc nhấn “F4” trên bàn phím. Sau đó click chuột vào một trong các lệnh cần lấy. Nhập giá trị chưa biết theo lệnh cần so sánh (byte, word, double word) vào vị trí các dấu chấm hỏi nằm trên lệnh. Nhập giá trị đã biết (thường là các con số) hoặc giá trị được chứa trong các ô nhớ byte, word, double word vào vị trí các dấu chấm hỏi nằm dưới lệnh.

So sánh Byte: Lệnh so sánh Byte dùng để so sánh 02 giá trị dạng byte được định địa chỉ bởi hai toán hạng ở đầu vào của lệnh: [IN1] và [IN2]. Có tất cả 06 phép so sánh có thể được thực hiện: Các byte được đem so sánh là những giá trị không dấu. Trong LAD, lệnh này có dạng một công tắc và công tắc đó đóng (ON) khi điều kiện đem so sánh có giá trị đúng. Trong FBD, đầu ra sẽ có giá trị 1 nếu điều kiện đem so sánh là đúng. Trong STL, lệnh được thực hiện sẽ ghi giá trị 1 vào đỉnh ngăn xếp (với những lệnh Load) hoặc thực hiện phép toán lô gic AND hay OR (tùy theo lệnh cụ thể) giá trị 1 với đỉnh ngăn xếp nếu điều kiện so sánh đúng.

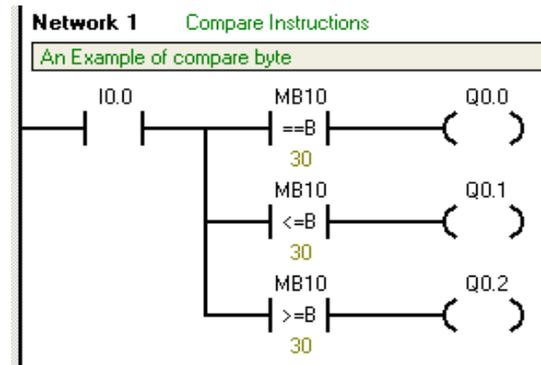
So sánh số nguyên (Integer): Lệnh so sánh số nguyên dùng để so sánh 02 giá trị dạng Integer được định địa chỉ bởi hai toán hạng ở đầu vào của lệnh: Các số nguyên

được đem so sánh là những giá trị có dấu: $16\#7FFF > 16\#8000$.

So sánh từ kép (Double Word): Lệnh so sánh từ kép dùng để so sánh 02 giá trị dạng Double Word được định địa chỉ bởi hai toán hạng ở đầu vào của lệnh: [IN1] và [IN2]. Có tất cả 06 phép so sánh có thể được thực hiện: Các giá trị từ kép được đem so sánh là những giá trị có dấu: $16\#7FFFFFFF > 16\#80000000$.

So sánh số thực (Real): Lệnh so sánh số thực dùng để so sánh 02 giá trị dạng Real được định địa chỉ bởi hai toán hạng ở đầu vào của lệnh: Các số thực được đem so sánh là những giá trị có dấu theo kiểu dấu phẩy động.

Ví dụ:



Hình 5.5: Lệnh so sánh sử dụng trong LAD

Hoạt động:

Ngõ ra Q0.0 có điện khi đạt đủ 2 điều kiện sau: Ngõ vào I0.0 ở mức 1 và giá trị trong MB10 bằng 30.

Ngõ ra Q0.1 có điện khi đạt đủ 2 điều kiện sau: Ngõ vào I0.0 ở mức 1 và giá trị trong MB10 nhỏ hơn hoặc bằng 30.

Ngõ ra Q0.2 có điện khi đạt đủ 2 điều kiện sau: Ngõ vào I0.0 ở mức 1 và giá trị trong MB10 lớn hơn hoặc bằng 30.

5.3 CHỨC NĂNG DỊCH CHUYỂN

Với chức năng dịch chuyển, nội dung của một vùng này sẽ được di chuyển hoặc sao chép đến một vùng khác trong bộ nhớ. Việc trao đổi hay sao chép nội dung có thể thực hiện với một byte, một word, một double word hay một giá trị số hoặc một mảng lớn dữ liệu từ vùng này sang vùng khác trong bộ nhớ.

5.3.1 Các lệnh sao chép, trao đổi nội dung

Để sao chép các dữ liệu kiểu byte, word, double word kể cả số thực (real) từ nơi này đến nơi khác ta sử dụng lệnh Move. Trong một số trường hợp cần trao đổi nội dung của một byte (byte thấp và byte cao) trong một word ta sử dụng lệnh Swap.

Cú pháp của các lệnh ở STL như sau:

- **Lệnh MOVB IN,OUT:** Lệnh Move Byte (MOVB) thực hiện sao chép nội dung của byte IN sang byte OUT

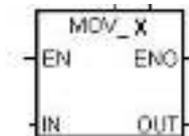
- **Lệnh MOVW IN,OUT:** Lệnh Move Word (MOVW) thực hiện sao chép nội dung của word IN sang word OUT

- **Lệnh MOVD IN,OUT:** Lệnh Move Double Word (MOVD) thực hiện sao chép nội dung của double word IN sang double word OUT.

- **Lệnh MOV R IN,OUT:** Lệnh Move Real (MOV R) thực hiện sao chép nội dung của một số thực IN sang số thực OUT.

- **Lệnh SWAP IN:** Lệnh Swap Byte (Swap) thực hiện trao đổi nội dung của byte thấp và byte cao trong word IN.

Cú pháp của các lệnh ở LAD như sau:



Với:

X: Có thể là B (Byte), W (Word), D (Double word) hoặc R(Real)

IN: Dữ liệu cần sao chép, có thể là byte, word, double word hoặc real tùy theo **X** là B, W, D hay R

OUT: Vị trí của nơi cần sao chép đến, có thể là byte, word, double word hoặc real tùy theo **X** là B, W, D hay R

EN: Là ngõ vào bit. Cho phép thực hiện lệnh được viết ở LAD hoặc FBD. Trường hợp không cần thiết có điều kiện ở ngõ vào EN thì phải sử dụng SM0.0

ENO: Ngõ ra bit. Cho phép kết nối song song hoặc nối tiếp với các hộp khác. Nếu phép toán xử lý không có lỗi thì EN=ENO

Để lấy lệnh MOV ở màn hình soạn thảo LAD, ta nhấp chuột vào dấu (+) ở biểu tượng  Move trong cây lệnh. Sau đó trở chuột vào một trong các lệnh cần lấy và giữ chuột trái, kéo và thả vào vị trí mong muốn. Nhập điều kiện cho ngõ vào EN, nếu lúc nào cũng thực hiện thì sử dụng bit nhớ SM0.0, Byte có nội dung cần sao chép đặt ở ngõ IN và byte chứa đựng thông tin sao chép chứa ở OUT. Hoặc nhấn phím **F9** trên bàn phím sau đó nhập tên *Move-X* rồi nhấn **Enter**.

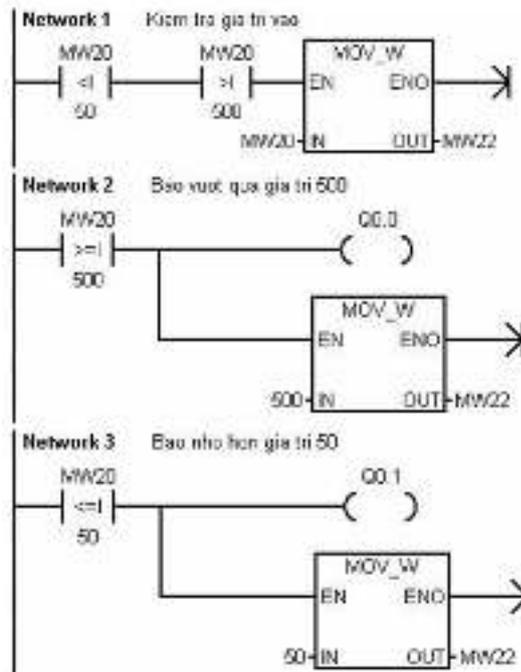
Ví dụ: Giới hạn giá trị.

Viết một chương trình thực hiện nhiệm vụ sau: Nếu giá trị ở MW20 nằm trong phạm vi (50;500) thì sẽ cho phép xuất giá trị ra ở ngõ ra MW22. Nếu giá trị ở MW20 lớn hơn giá trị 500 thì ngõ ra số MW22 là giá trị 500 và đèn báo giá trị max sáng. Nếu giá trị ở MW20 nhỏ hơn giá trị 50 thì ngõ ra số MW22 là giá trị 50 và đèn báo giá trị min sáng. Chú ý các ngõ vào ra số là Int

Bảng 5.3 Ký hiệu

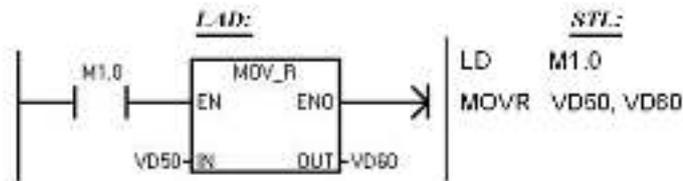
<i>Ký hiệu</i>	<i>Địa chỉ</i>	<i>Chú thích</i>
GT_sosanh	MW20	Giá trị số cần biết có vượt ngoài phạm vi (50;500)
GT_dung	MW22	Giá trị nằm trong phạm vi cho phép
Bao_max	Q0.0	Đèn báo giá trị lớn hơn 500
Bao_min	Q0.1	Đèn báo giá trị nhỏ hơn 50

Chương trình



Hình 5.6: Lệnh so sánh sử dụng trong LAD

Ví dụ: Copy ô nhớ số thực ở VD50 vào ô nhớ số thực VD60 khi M1.0 tích cực.
Chương trình được viết như sau:



Hình 5.7: Lệnh chuyển nội dung VD50 vào VD60

Cú pháp của lệnh SWAP được lấy như sau:

LAD	Toán hạng
	IN (Word): VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD

5.3.2 Các lệnh sao chép một mảng lớn dữ liệu

Để sao chép một mảng lớn dữ liệu từ nơi này đến nơi khác ta sử dụng lệnh Block Move. Lệnh sao chép một mảng lớn cho phép thực hiện với Byte, Word và Double Word.

Cú pháp của các lệnh ở STL như sau:

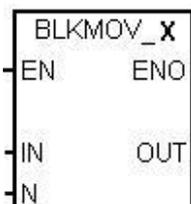
Lệnh BMB IN,OUT,N: Lệnh Block Move Byte (BMB) sao chép nội dung của một mảng Byte. Số lượng byte được sao chép xác định bởi N có kiểu byte. Do đó có thể sao chép tối đa là 255 byte. Byte đầu tiên của mảng được xác định ở ngõ IN (kiểu byte). Nơi đến được xác định với byte đầu tiên của mảng ở ngõ OUT.

Lệnh BMW IN,OUT,N: Tương tự như lệnh BMB, lệnh Block Move Word (BMW) sao chép nội dung của một mảng word. Số lượng word được sao chép xác định bởi N có kiểu byte. Do đó có thể sao chép tối đa là 255 word. Word đầu tiên của mảng

được xác định ở ngõ IN (kiểu word). Nơi đến được xác định với word đầu tiên của mảng ở ngõ OUT.

Lệnh BMD IN,OUT,N: Tương tự như lệnh BMB, lệnh Block Move Double Word (BMD) sao chép nội dung của một mảng Double Word. Số lượng Double word được sao chép xác định bởi N có kiểu byte. Do đó có thể sao chép tối đa là 255 Double word. Double Word đầu tiên của mảng được xác định ở ngõ IN (kiểu Double word). Nơi đến được xác định với Double word đầu tiên của mảng ở ngõ OUT.

Cú pháp của lệnh ở LAD và FBD có cấu trúc tổng quát như sau:



Trong đó: **X:** Có thể là B (Byte), W (Word), D (Double word).

IN: Vị trí đầu tiên của mảng dữ liệu cần sao chép, có thể là Byte, Word hoặc double Word tùy theo **X**.

OUT: Vị trí đầu tiên của mảng dữ liệu cần lưu trữ thông tin sao chép. có thể là Byte, Word hoặc double Word tùy theo **X**.

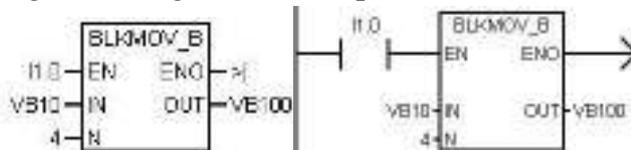
N: Số lượng Byte, Word, Double word sao chép, có giá trị từ 0 đến 255.

EN, ENO: tương tự như ở lệnh MOVE.

Để lấy lệnh BLKMOV ở màn hình soạn thảo LAD, ta nhấp chuột vào dấu (+) ở biểu tượng  Move trong cây lệnh. Sau đó trở chuột vào một trong các lệnh cần lấy và giữ chuột trái, kéo và thả vào vị trí mong muốn. Nhập điều kiện cho ngõ vào EN, nếu lúc nào cũng thực hiện thì sử dụng bit nhớ SM0.0, Byte có nội dung cần sao chép đặt ở ngõ IN và byte chứa đựng thông tin sao chép chứa ở OUT. Hoặc nhấn phím **F9** trên bàn phím sau đó nhập tên **Move-X** rồi nhấn **Enter**.

Ví dụ: Khi kích hoạt I1.0 thì nội dung của một mảng gồm 4 byte bắt đầu từ Byte VB10 sẽ được copy sang vùng nhớ gồm có 4 byte khác có byte đầu tiên là VB100. Chương trình được viết như dưới đây

Giả thiết nội dung của mảng cần sao chép là:



5.4 CHỨC NĂNG CHUYỂN ĐỔI

Các hàm chuyển đổi dữ liệu cho phép thực hiện việc biến đổi dữ liệu từ kiểu này sang kiểu khác. Trong S7-200 có các hàm chuyển đổi dữ liệu sau:

- DTR (DI_REAL): Hàm chuyển đổi số nguyên 32bit có dấu IN sang số thực 32bit và ghi kết quả vào OUT. Hàm này chỉ có ở CPU 214.

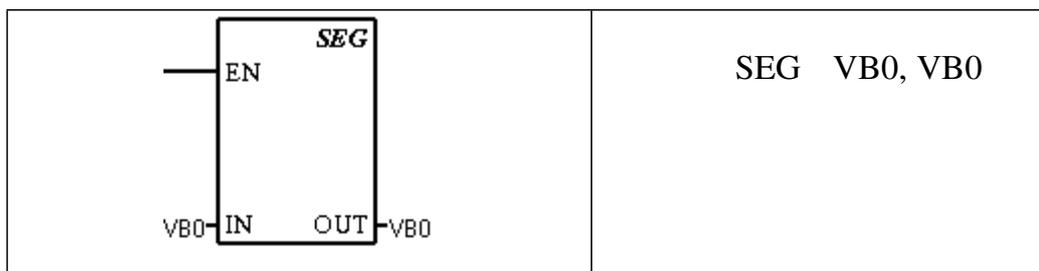
- ATH: chuyển đổi một chuỗi ký tự có độ dài chỉ thị trong toán hạng LEN, bắt đầu bằng ký tự chỉ định trong toán hạng IN, sang số nguyên hệ thập lục phân có cơ số

16 và ghi vào vùng nhớ kể từ byte được chỉ định bởi OUT. Độ dài cực đại của chuỗi ký tự là 255. Nếu mã hoá một ký tự sai, quá trình dừng lại và bit SM1.7 có giá trị 1.

🚦 Lệnh chuyển đổi số nguyên hệ thập lục phân sang led 7 đoạn:

Dạng LAD:

Dạng STL:



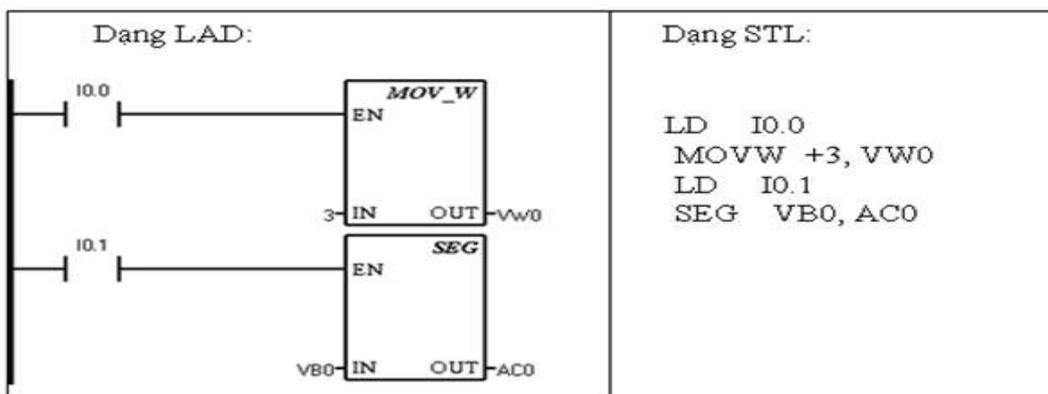
Ý nghĩa:

- Lệnh này có tác dụng chuyển đổi các số trong hệ thập lục phân từ 0 đến F chứa trong 4 Bit thấp của byte có địa chỉ ở ngõ vào IN thành giá trị BIT chứa trong 8 bit của byte có địa chỉ ở ngõ ra OUT tương ứng với thanh led 7 đoạn . Trong lệnh này byte có địa chỉ ở ngõ vào IN và byte có địa chỉ ở ngõ ra OUT có thể cùng địa chỉ và nằm trong những vùng sau:

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

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

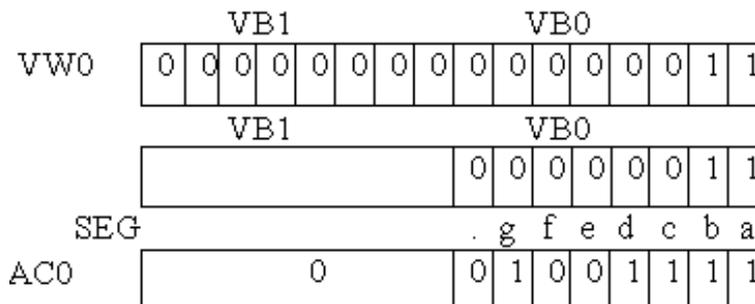
Ví dụ:



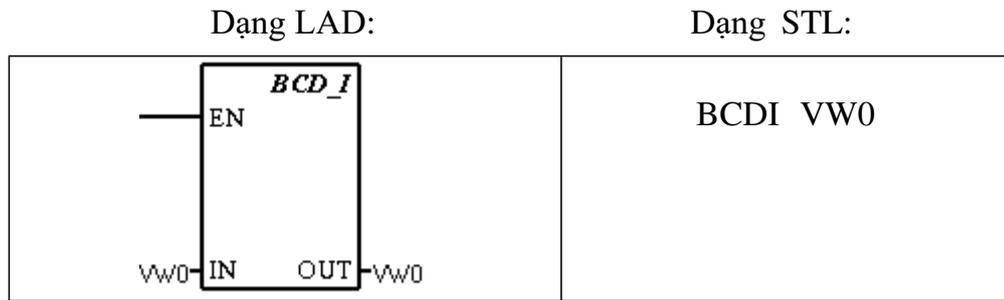
Hình 5.8: Sơ đồ LAD

Giải thích:

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



Lệnh chuyển đổi số mã BCD sang số nguyên:



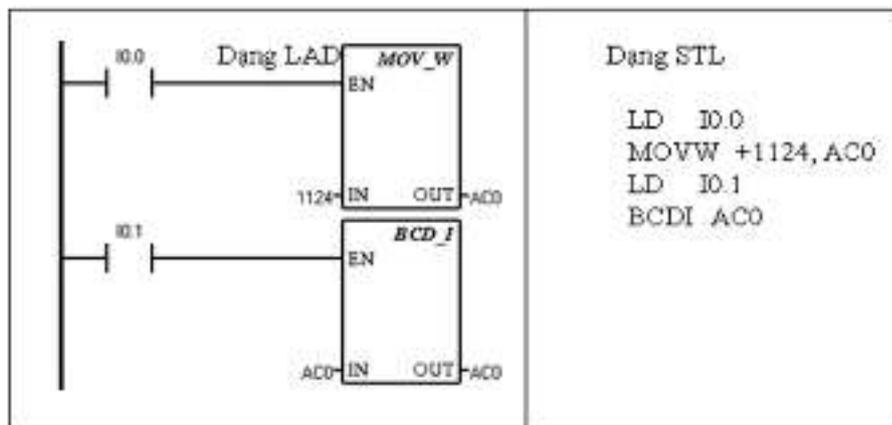
Ý nghĩa:

- Lệnh này thực hiện phép biến đổi một số nhị thập phân 16 bit chứa trong word có địa chỉ ở ngõ vào IN sang số nguyên 16 bit chứa trong word có địa chỉ ở ngõ ra OUT. Đặc biệt ở đây word có địa chỉ ở ngõ vào IN và word có địa chỉ ở ngõ ra OUT có thể cùng một địa chỉ. Địa chỉ này thường nằm trong các vùng sau:

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

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

Ví dụ:



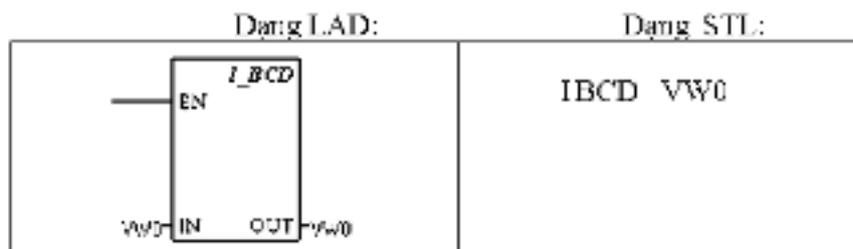
Hình 5.9: Sơ đồ LAD

Giải thích:

- Khi I0.0 đóng, giá trị 1124 theo mã BCD là 0001 0001 0010 0100 được ghi vào địa chỉ AC0. Tiếp điểm I0.1 đóng thì giá trị BCD đó được chuyển sang số nguyên và lưu vào AC0. Ta biểu diễn theo bit như sau:

MOV	AC0
MaBCD	Số nguyên
0 0 0 1	0 0 0 0
0 0 0 1	0 1 0 0
0 0 1 0	0 1 1 0
0 1 0 0	0 1 0 0
$AC0 = 1024 - 64 - 32 - 4 = 1124$	

Lệnh chuyển đổi số nguyên sang mã BCD



Ý nghĩa:

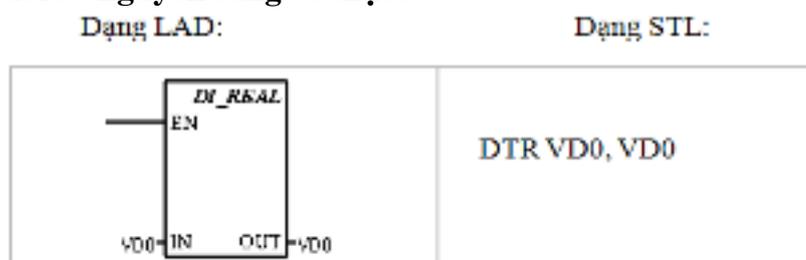
- Lệnh này thực hiện phép biến đổi một số nguyên 16 bit chứa trong word có địa chỉ ở ngõ vào IN sang số nhị thập phân 16 bit chứa trong word có địa chỉ ở ngõ ra OUT. Đặc biệt ở đây word có địa chỉ ở ngõ vào IN và word có địa chỉ ở ngõ ra OUT có thể cùng một địa chỉ.

- Địa chỉ này thường nằm trong các vùng sau:

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

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

Lệnh chuyển đổi số nguyên sang số thực:



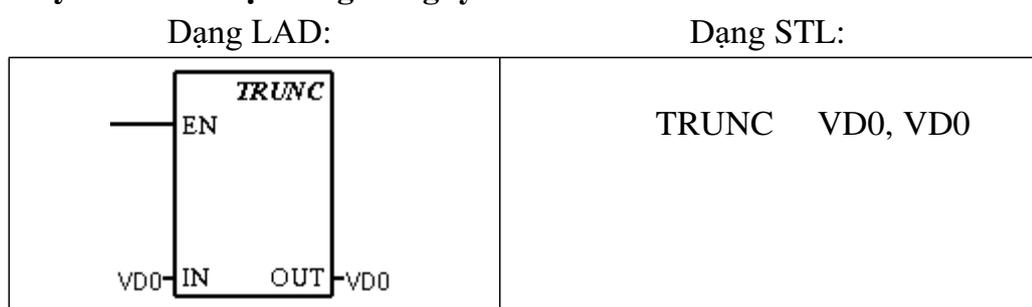
Ý nghĩa:

- Lệnh này thực hiện phép biến đổi một số nguyên 32 bit có địa chỉ ở ngõ vào IN thành số thực 32 bit rồi ghi vào Dword có địa chỉ ở ngõ ra OUT. Trong đó toán hạng IN và OUT có thể cùng địa chỉ và thuộc một trong các vùng sau:

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

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

Lệnh chuyển đổi số thực sang số nguyên:



Ý nghĩa:

- Lệnh này thực hiện phép biến đổi một số thực 32 bit chứa trong Dword có địa chỉ ở ngõ vào IN thành số nguyên 32 bit rồi ghi vào Dword có địa chỉ ở ngõ ra OUT. Trong đó toán hạng IN và OUT có thể cùng địa chỉ và thuộc một trong các vùng sau:

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

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

5.5 CHỨC NĂNG TOÁN HỌC

Ở nhiều nhiệm vụ đếm như đếm sản phẩm, đếm số vòng quay, đếm xung.v.v... thì kết quả đếm phải được giám sát. Bên cạnh các phép toán so sánh đã biết cần phải có thêm các phép toán số học như cộng, trừ, nhân, chia. Còn các phép toán khác như sin, cos, tan, PID sẽ được khảo sát ở *tập 2* của bộ sách *kỹ thuật điều khiển lập trình PLC SIMATIC S7-200*.

5.5.1 Cộng và trừ

Các phép toán cộng và trừ có thể thực hiện được đối với các số Integer (16 bit), Double integer (32 bit) và số thực (32 bit). Tùy thuộc vào phép toán là cộng hoặc trừ dạng số nào mà kết quả thu được sẽ ở dạng số đó.

Khi có lỗi do tràn hoặc giá trị không hợp lệ, bit SM1.1 được set lên mức logic 1.

Cú pháp lệnh biểu diễn cho phép toán cộng và trừ như sau:

Để lấy lệnh ***cộng hoặc trừ số nguyên*** ở màn hình soạn thảo LAD, ta nhấp chuột vào dấu (+) ở biểu tượng trong cây lệnh. Sau đó trở chuột vào một trong các lệnh cần lấy là (cộng số Integer), (cộng số DInt), (trừ số Integer), hoặc (trừ số DInt), giữ chuột trái, kéo và thả vào vị trí mong muốn. Nhập điều kiện cho ngõ vào EN, nếu lúc nào cũng thực hiện thì sử dụng bit nhớ SM0.0. Nhập các biến của phép toán tương ứng vào các ngõ IN1 và IN2. Nhập biến chứa kết quả ở ngõ OUT.

Để lấy lệnh ***cộng hoặc trừ số thực (real)*** ở màn hình soạn thảo LAD, ta nhấp chuột vào dấu (+) ở biểu tượng trong cây lệnh Sau đó trở chuột vào một trong các lệnh cần lấy là: (cộng số real) (trừ số real), giữ chuột trái, kéo và thả vào vị trí mong muốn. Nhập điều kiện cho ngõ vào EN, nếu lúc nào cũng thực hiện thì sử dụng bit nhớ SM0.0. Nhập các biến của phép toán tương ứng vào các ngõ IN1 và IN2.

Nhập biến chứa kết quả ở ngõ OUT.

5.5.2 Nhân và chia

Các phép toán nhân và chia có thể thực hiện được đối với các số Integer (16 bit), Double integer (32 bit) và số thực (32 bit). Tùy thuộc vào phép toán là nhân hoặc chia dạng số nào mà kết quả thu được sẽ ở dạng số đó.

Khi có lỗi do tràn hoặc giá trị không hợp lệ thì bit SM1.1 được set lên mức logic “1“. Nếu kết quả là zero thì SM1.0 =”1”, kết quả âm thì SM1.2 =”1”, và SM1.3 =”1” nếu chia cho 0.

Cú pháp lệnh biểu diễn cho phép toán nhân và chia như sau: số DInt), giữ chuột trái, kéo và thả vào vị trí mong muốn. Nhập điều kiện cho ngõ vào EN, nếu lúc nào cũng thực hiện thì sử dụng bit nhớ SM0.0. Nhập các biến của phép toán tương ứng vào các ngõ IN1 và IN2. Nhập biến chứa kết quả ở ngõ OUT.

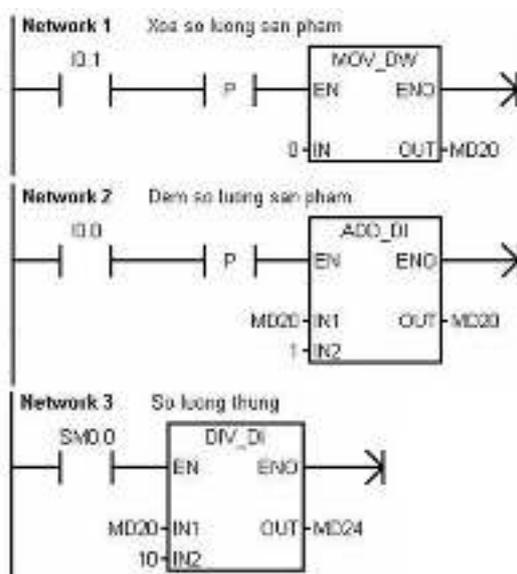
Để lấy lệnh ***nhân hoặc chia số thực (real)*** ở màn hình soạn thảo LAD, điều kiện cho ngõ vào EN, nếu lúc nào cũng thực hiện thì sử dụng bit nhớ SM0.0. Nhập các biến của phép toán tương ứng vào các ngõ IN1 và IN2. Nhập biến chứa kết quả ở ngõ OUT.

5.5.3 Thực hành phép toán số học

Ví dụ: Đếm sản phẩm

Sản phẩm trên một băng tải được nhận biết bởi cảm biến S1. Tổng số lượng sản

phẩm đếm được chứa trong MD20. Cứ 10 sản phẩm sẽ được đóng thành một thùng và số lượng thùng được chứa trong MD24. Số lượng sản phẩm có thể bị xóa bằng nút nhấn S2.



Hình 5.10: Chương trình đếm sản phẩm

CÁC BÀI TẬP THỰC HÀNH:

Bài 1:

1. Yêu cầu công nghệ:

- Trong ứng dụng này, hệ thống sẽ lựa sản phẩm thiếu chất lượng trộn lẫn trong sản phẩm đúng tiêu chuẩn. Ví dụ phân biệt bóng đen và bóng trắng và dựa vào hai contenơ khác nhau.

- Khi bắt đầu hoạt động, sensor (S1) sẽ cảm nhận sự hiện diện của bóng trong hộp. Cuộn selenoid ở trên (top selenoid) thả ra cho sensor (S2) cảm nhận màu trước khi thả vào contenơ. Nếu bóng đen, piston (1002) sẽ được kích hoạt và chuyển hướng bóng đen xuống contenơ khác.

2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình

Bài 2:

1. Yêu cầu công nghệ:

Hãy viết chương trình điều khiển hai động cơ hoạt động theo chế độ như sau:

- Động cơ 1 chạy 5 giây rồi ngừng sau đó đến động cơ 2 chạy 5 giây rồi ngừng 2 giây, lặp lại 5 lần như vậy, kể đến thì chu kỳ làm việc của hai động cơ lặp lại 10 lần rồi nghỉ. Muốn làm việc nữa thì khởi động lại.

2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi

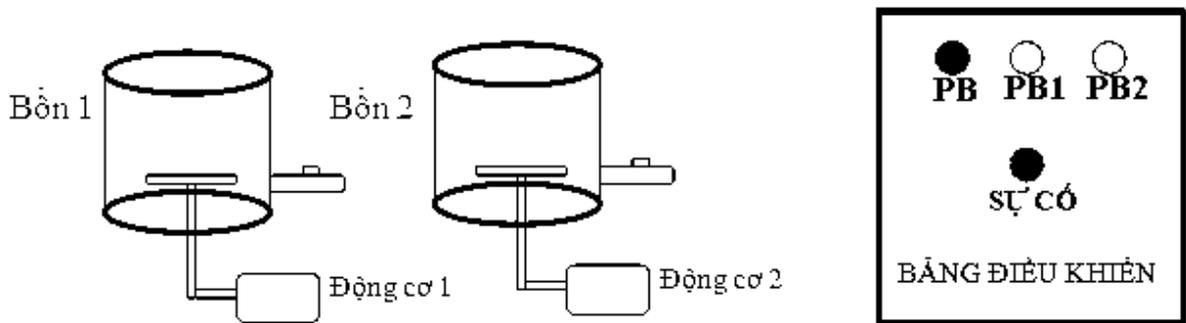
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình

Bài 3:

1. Yêu cầu công nghệ:

Công ty TECHNOPIA có hai bồn trộn hóa chất, mỗi bồn được kéo bởi một động cơ

- Bồn 1 trộn hóa chất a.
- Bồn 2 trộn hóa chất b.
- Trên bảng điều khiển có ba chọn lựa:
- Nếu nhấn nút PB thì cả hai bồn đều được chọn làm việc trong 30 giây.
- Nếu nhấn nút PB1 thì chỉ có bồn 1 làm việc trong 30 giây (bồn 2 nghỉ).
- Nếu nhấn nút PB2 thì chỉ có bồn 2 làm việc trong 30 giây (bồn 1 nghỉ).
- Khi đang trộn hóa chất nếu bồn hóa chất bị hỏng van thì phải báo động ngay lập tức và dừng quá trình trộn lại. (hình vẽ).



Hình 5.11: Trộn hóa chất

2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình

Bài 4:

1. Yêu cầu công nghệ:

Có một máy bán nước tự động, tùy thuộc số tiền ta đưa vào trong máy thì loại nước uống tương ứng sẽ được cho ra. Tiền đưa vào phải tương đương hoặc lớn hơn giá tiền qui định cho từng sản phẩm.

A, B, C, D : 8 đồng. E : 4 đồng. F : 2 đồng

2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi.
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình



Hình 5.12: Máy bán nước tự động

Bài 5:

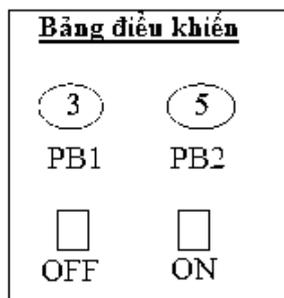
1. Yêu cầu công nghệ:

- Do yêu cầu của công nghệ nên có một động cơ vận hành theo chế độ như sau:

+ Nhấn nút ON (động cơ chuẩn bị làm việc), sau đó chọn chế độ làm việc .

+ Nếu nhấn nút PB1: thì động cơ chạy 50 giây dừng 10 giây rồi chạy ngược 50 giây, dừng 10 giây và chu kỳ lặp lại 3 lần như ban đầu.

+ Nếu nhấn nút PB2 thì động cơ chạy 50 giây ,dừng 10 giây sau đó chạy ngược 50 giây, dừng 10 giây và chu kỳ bắt đầu lặp lại như ban đầu 5 lần.



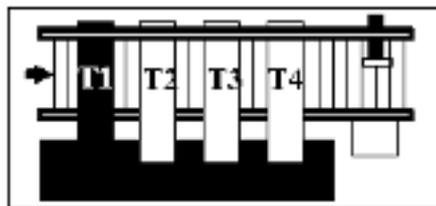
Hình 5.14: Điều khiển động cơ

2. Yêu cầu thực hành:.

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình

Bài 6:

1. Yêu cầu công nghệ:



Hình 5.15: Dây chuyền sản xuất

- Nhà máy sản xuất bia, sau khi đổ bia vào chai thì các chai bia này được đưa qua một băng tải. Dọc theo băng tải có 4 trạm kiểm tra:

- Trạm 1: Kiểm tra chai có bị mẻ hay không.

- Trạm 2: Kiểm tra nhãn chai .
- Trạm 3: Kiểm tra nút chai .
- Trạm 4: Kiểm tra bia đầy hay không .
- Nếu chai bia nào không đảm bảo bất kỳ tiêu chuẩn kiểm tra nào thì sẽ bị loại bỏ sau khi qua 4 trạm. Hãy viết chương trình kiểm tra sản phẩm theo yêu cầu trên.

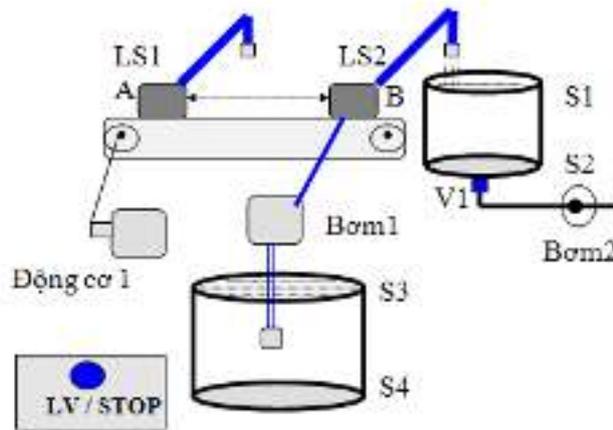
2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình

Bài 7:

1. Yêu cầu công nghệ:

- Có công nghệ đồ nguyên liệu như hình vẽ:



Hình 5.16: Dây chuyền sản xuất

- LS1, LS2: Công tắc hành trình
- S1, S2: Báo bồn 1 đầy và cạn
- S3, S4: Báo bồn nguyên liệu đầy và cạn

2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình

Bài 8:

1. Yêu cầu công nghệ:

- Viết 1 chương trình điều khiển đèn giao thông cho ngã tư, đèn hoạt động như sau:

- + Có thể thay đổi được thời gian cho đèn xanh tuyến 1 và tuyến 2.
- + Vàng tuyến 1 và tuyến 2 có giá trị mặc định là 5 giây.
- + Xanh tuyến 1, tuyến 2 có giá trị mặc định là 15 giây.
- + Vàng tuyến 1 và tuyến 2 là 5 giây.

2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển
- Chạy mô phỏng chương trình

Bài 9:

1. Yêu cầu công nghệ:

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

- Chủ Nhật chuông không kêu.
- Từ 7h00'00" đến 7h00'10" chuông kêu báo vào giờ học.
- Từ 9h00'00" đến 9h00'08" chuông kêu báo giờ giải lao.
- Từ 9h15'00" đến 9h15'10" chuông kêu báo vào giờ học.
- Từ 11h00'00" đến 11h00'20" chuông kêu báo giờ học kết thúc.

2. Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển

Bài 6

XỬ LÝ TÍN HIỆU ANALOG

GIỚI THIỆU

Tín hiệu analog là tín hiệu liên tục, đồ thị biểu diễn tín hiệu analog là một đường liên tục (ví dụ sin, cos, hoặc đường cong lên xuống bất kỳ), analog có nghĩa là tương tự, tương tự có nghĩa là tín hiệu lúc sau cũng có dạng tương tự như lúc trước đó, nhưng không có nghĩa là giống nhau hoàn toàn mà chỉ tương tự về bản chất tín hiệu, nhưng sẽ khác về cường độ tín hiệu lúc sau so với lúc trước. Trong thiết bị điện tín hiệu analog là dòng điện, trong cuộc sống analog có thể là tín hiệu âm thanh ta nghe, hình ảnh ta thấy, trong viễn thông là sóng điện từ (tức ánh sáng không nhìn thấy)...

Tín hiệu analog cho plc thì có nhiều loại về mặt vật lý như: 0-10Vdc, 4-20mA, RTD, T/C,... Để xử lý tín hiệu Analog khi đọc về PLC thì cần để ý đến độ phân giải đầu vào là 11bit + 1bit dấu, 12bit hay 16bit.

MỤC TIÊU BÀI HỌC

+ Về kiến thức:

- Trình bày được nguyên lý hoạt động, đặc tính và phạm vi ứng dụng các bộ Analog theo nội dung đã học.

- Trình bày cách đo tín hiệu tương tự bằng PLC.

+ Về kỹ năng:

- Kiểm tra, sửa chữa các kết nối hoặc chương trình xử lý đúng yêu cầu kỹ thuật.

- Thực hiện kết nối các ngõ vào/ra analog trong PLC.

- Lập trình hiệu chỉnh được tín hiệu tương tự với PLC S7-200 trong thực tế

+ Về thái độ:

- Rèn luyện tính tỷ mỉ, chính xác, an toàn và vệ sinh công nghiệp

NỘI DUNG BÀI HỌC

6.1 TÍN HIỆU ANALOG

Tín hiệu Analog là các tín hiệu tương tự (0 – 10VDC, hoặc 4-20mA.....). Tín hiệu analog có thể là tín hiệu từ các cảm biến đo khoảng cách, cảm biến áp suất, cảm biến đo trọng lượng, tốc độ, nhiệt độ,... Khi chuyển sang giá trị số thì thường là giá trị số 12 bit được lưu giữ trong thanh ghi hay tại các vị trí từ đơn. Thực nghiệm tính năng **Trend View** để xem dạng sóng vào chân tương tự PLC. Hầu hết các ứng dụng của chương trình PLC Siemens nói riêng hay các ứng dụng khác đều cần phải đọc các tín hiệu analog.

Các bước đọc tín hiệu Analog:

6.1.1 Đọc tín hiệu analog từ Modul EM231:

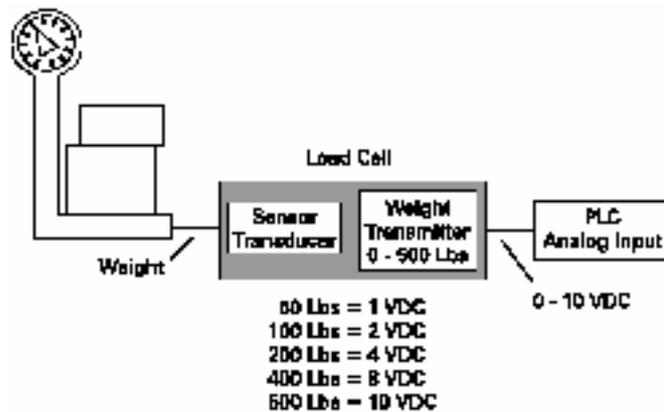
Các tín hiệu có thể đọc được từ Modul EM231 (tùy thuộc việc chọn các Switch trên modul):

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

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

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

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

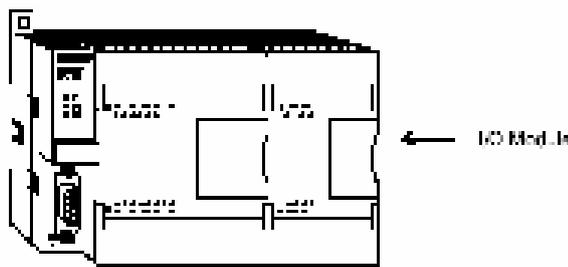


Hình 6.1: Đọc tín hiệu vào tương tự

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

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

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



Hình 6.2: Module mở rộng tương tự

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

Ví dụ:

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

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

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

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

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

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

- Bước 1: Ta cần xác định 2 điểm:

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

Từ đó ta có 2 điểm: Điểm 1 (x1,m1) , Điểm 2 (x2,m2). Phương trình đường thẳng đi qua 2 điểm 1,2 có dạng: $(X-X1/X2-X1) = (Y-Y1/Y2-Y1)$,

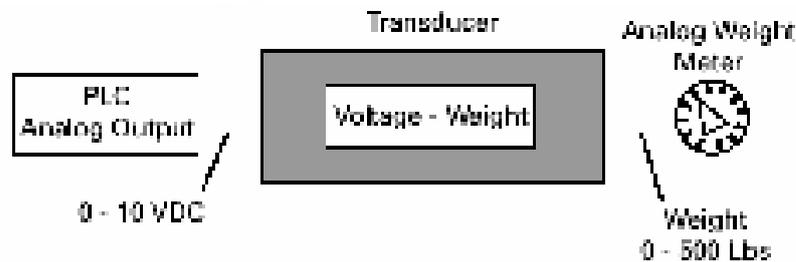
Từ đó rút Y theo X Đó chính là phương trình khối lượng theo AIW.

- Ví dụ cụ thể: Điểm 1 (0,0), điểm 2 (32000,1000) Phương trình lập:

$(X-0/32000-0) = (Y-0/1000-0)$ Từ đó suy ra: $Y= 1xX/ 32$

Vậy khối lượng = AIW / 32

6.1.2 Xuất tín hiệu analog qua modul EM232:



Hình 6.3: Xuất ra tín hiệu tương tự

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

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

- Tín hiệu lưỡng cực (tín hiệu điện áp): -10VDC – 10VDC Tín hiệu 0 -20mA tương ứng với giá trị 0 – 32000

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

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

6.1.3 Modul EM235:

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

Đơn cực: 0 – 50mV, 0 – 100mV, 0 – 500mV, 0 – 1V, 0 – 5VDC, 0 – 20mA , 0 – 10VDC. Lưỡng cực: ±25mV, ±50mV, ±100mV, ±250mV, ±500mV, ±1VDC , ±2.5VDC, ±5VDC , ±10VDC

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

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

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

6.1.4 I/O cục bộ và mở rộng

Cấu trúc MODULE của S7-200 tạo sự linh hoạt tối đa để giải quyết các bài toán, nó cho phép chúng ta chọn số đầu vào ra tối ưu về mặt kinh tế. Tăng số ngõ vào ra bằng các module mở rộng.

Các module mở rộng này được cắm nối tiếp nhau vào bên phải CPU. Địa chỉ

các đầu vào ra trên các module mở rộng được tính liên tiếp, riêng cho từng loại (vào, ra, vào tương tự, ra tương tự) không ảnh hưởng lẫn nhau. Các đầu vào ra rời rạc được định địa chỉ chẵn byte, nghĩa là trên một module phải bắt đầu bằng x.0, x.1,... còn các đầu vào ra tương tự được định địa chỉ theo từ đơn, cách hai, nghĩa là bằng các số chẵn: AIW0, AIW2, AIW4,... AQW0, AQW2, AQW4,...

Tín hiệu ngõ vào (Analog Input):

Đại lượng điện áp và dòng điện:

- Muốn đo tín hiệu điện áp hoặc dòng điện ta cần chọn module Analog mở rộng phù hợp như: EM 235 input/output

- Giá trị số các thang đo của các module Analog S7-200

Thang đo	Thang đo	Thang đo	Thang đo	Thang đo	Dữ liệu dạng số
± 25 mV	± 50 mV	± 100 mV	± 250 mV	± 500 mV	± 32000
± 1 V	± 2,5 mV	± 5 V	± 10 V		± 32000
0 → 50 mV	0 → 100mV	0 → 500mV			0 → 32000
0 → 1 V	0 → 5 V	0 → 10 V	0 → 20 mA		0 → 32000

Đại lượng điện trở, nhiệt độ:

- Cũng như ở trên ta cần chọn phần cứng Analog phù hợp cho S7-200:

+ EM 231 RTD

+ EM 231 Thermocouple

Thang đo	Dữ liệu dạng số
Cảm biến loại K -270 → 1372	± 32000
Cảm biến loại R -50 → 1768	
Cảm biến loại T -270 → 400	
Cảm biến loại J -210 → 750	
Cảm biến loại E -270 → 1000	
± 80 mV/> 1MΩ	

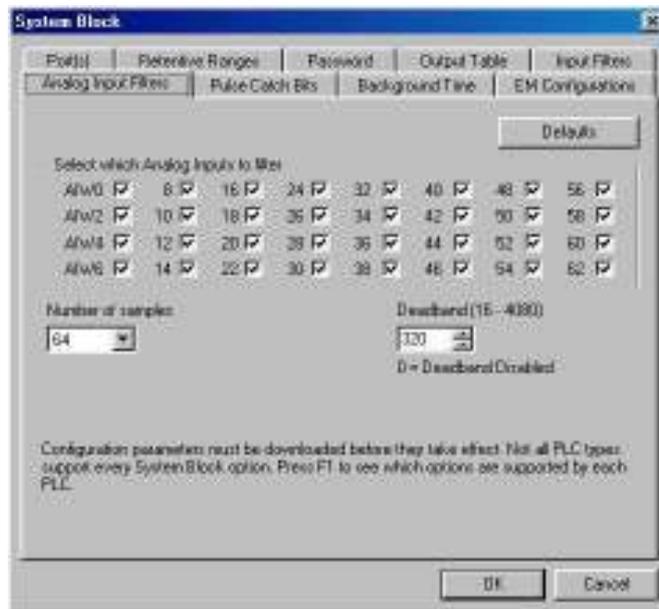
Tín hiệu ngõ ra (Output) Analog:

Thang đo	Dữ liệu dạng số
0 → 20 mA	0 → 32000
± 10 V	± 32000

6.1.5 Lọc đầu vào tương tự

Các đầu vào tương tự, cũng như các đầu vào rời rạc, có thể được lọc để chống hiện tượng nhiễu. Bản chất bộ lọc của một đầu vào tương tự là phép tính giá trị trung bình một số hữu hạn các giá trị lấy mẫu liên tiếp, nhằm giảm tác động của các giá trị ngoại lai. Tất nhiên tác động của bộ lọc bao giờ cũng làm chậm tín hiệu, trong trường hợp này có thể không thích hợp nếu đầu vào biến đổi nhanh. S7-200 xử lý vấn đề đó

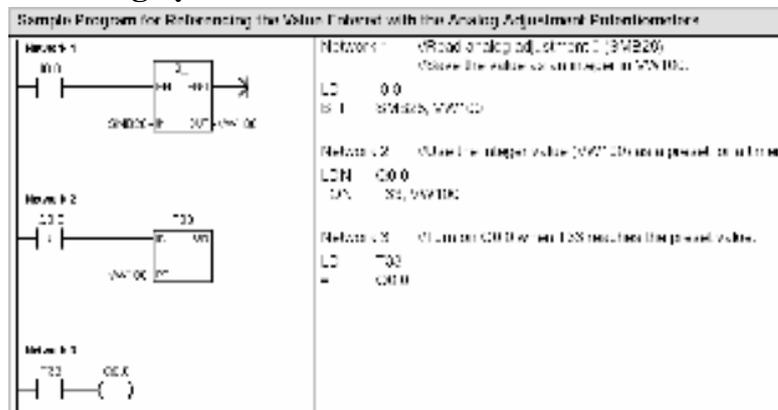
bằng khái niệm “deadband”: nếu giá trị lấy mẫu vượt ra ngoài qui định so với giá trị trung bình thì bộ lọc không tính giá trị trung bình nữa mà cập nhật luôn giá trị mới.



Hình 6.4: Lọc tín hiệu tương tự

Trong mọi trường hợp, người lập trình có thể bật hay tắt chức năng lọc cho từng đầu vào theo yêu cầu và cũng có thể đặt thông số chung cho các bộ lọc tương tự (số giá trị để tính trung bình, deadband) thông qua menu chính **View** → **Component** → **System Block** và chọn tag **Analog Input Filters**.

6.1.6 Định chỉnh tương tự



Hình 6.5: Chương trình LAD

S7-200 CPU có 1 hoặc 2 điều chỉnh tương tự phía trước. Ta có thể vận chúng theo chiều kim đồng hồ hay ngược lại trong khoảng 270 (để tăng hay giảm giá trị tương ứng với chúng là các byte trong SMB28 và SMB29. Như vậy những giá trị này có thể thay đổi trong khoảng từ 0 đến 255 và chương trình có thể sử dụng chúng như những giá trị chỉ đọc, thay đổi được theo sự can thiệp từ ngoài chương trình.

Ví dụ: hình 6.5

6.2 Thực hành đo lường và giám sát nhiệt độ với module EM235 và nhận cảm biến nhiệt điện trở Pt100

Yêu cầu phần cứng:

- S7-200 CPU, EM235 Analog Expansion Module

- Pt100 Temperature Sensor
- TD200 Operator Interface

Chọn dây điện áp trong giới hạn 0V÷10V cho EM235, bật các công tắc trên module theo các vị trí đã được ấn định tương ứng với từng dây điện áp đầu và độ phân dải của tín hiệu vào theo bảng dưới đây:

Bảng 6.1 SW lựa chọn giới hạn đo

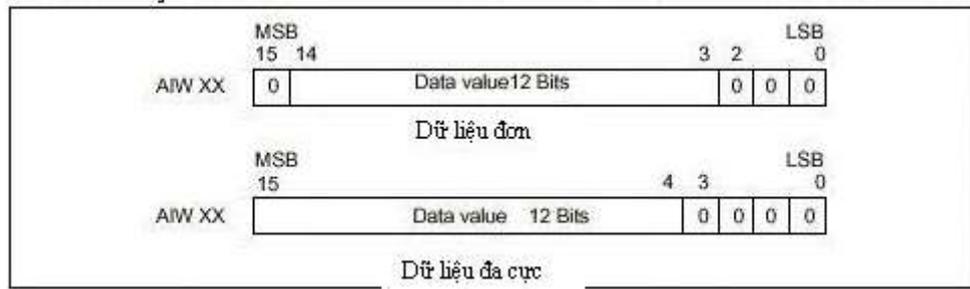
Không đảo dấu						Giới hạn dây điện áp đầu	Độ phân dải
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	ON	0 ÷ 50 mV	12.5 μV
OFF	ON	OFF	ON	OFF	ON	0 ÷ 100 mV	25 μV
ON	OFF	OFF	OFF	ON	ON	0 ÷ 500 mV	125 μV
OFF	ON	OFF	OFF	ON	ON	0 ÷ 1 V	250 μV
ON	OFF	OFF	OFF	OFF	ON	0 ÷ 5 V	12.5 mV
ON	OFF	OFF	OFF	OFF	ON	0 ÷ 20 mA	5 μA
OFF	ON	OFF	OFF	OFF	ON	0 ÷ 10 V	2.5 mV
Đảo dấu						Giới hạn dây điện áp đầu	Độ phân dải
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	OFF	±25 mV	12.5 μV
OFF	ON	OFF	ON	OFF	OFF	± 50mV	25 μV
OFF	OFF	ON	ON	OFF	OFF	± 100mV	50 μV
ON	OFF	OFF	OFF	ON	OFF	± 250 mV	125 μV
OFF	ON	OFF	OFF	ON	OFF	± 500 mV	250 μV
OFF	OFF	ON	OFF	ON	OFF	± 1V	500 μV
ON	OFF	OFF	OFF	OFF	OFF	± 2.5V	12.5 mV
OFF	ON	OFF	OFF	OFF	OFF	± 5V	25 mV
OFF	OFF	ON	OFF	OFF	OFF	± 10V	50 mV

SW6: Chọn điện áp và dòng vào có dấu hoặc không dấu; **SW4, SW5:** chọn hệ số khuếch đại; **SW3,2,1:** chọn hệ số suy giảm.

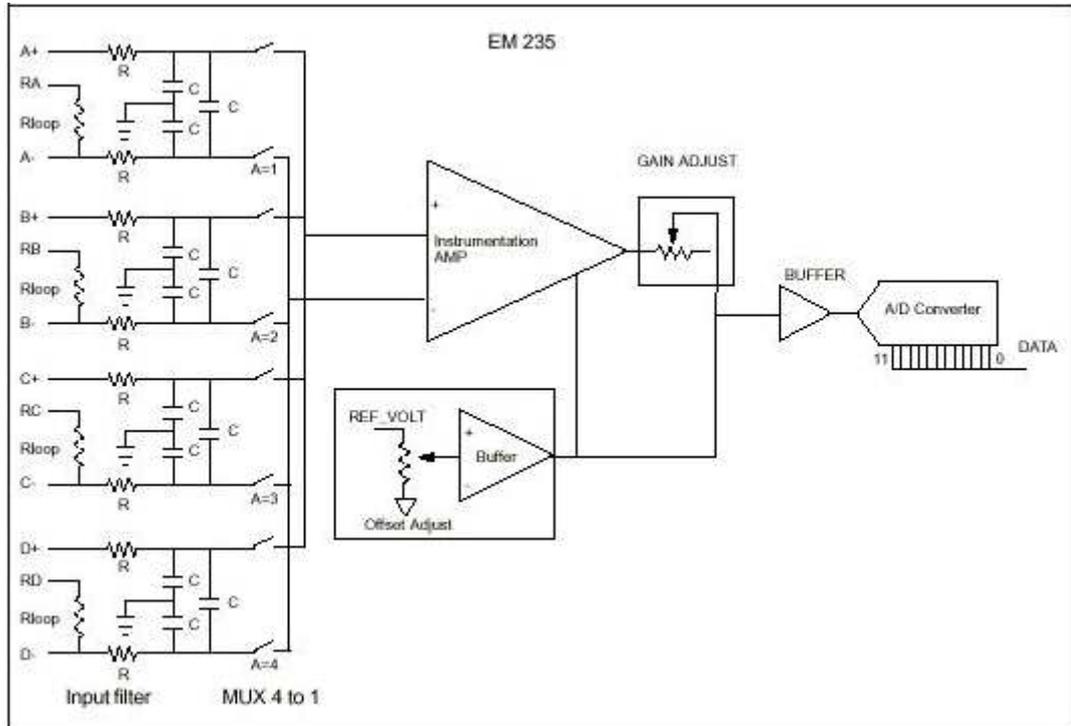
Bảng 6.2 Trạng thái ngõ vào tương tự

EM 235 Configuration Switches						Unipolar/Bipolar Select	Gain Select	Attenuation Select
SW1	SW2	SW3	SW4	SW5	SW6			
					ON	Unipolar		
					OFF	Bipolar		
			OFF	OFF			x1	
			OFF	ON			x10	
			ON	OFF			x100	
			ON	ON			invalid	
ON	OFF	OFF						0.8
OFF	ON	OFF						0.4
OFF	OFF	ON						0.2

Giá trị chuyển đổi ADC 12 bit của từ đơn đối với tín hiệu vào có hoặc không có dấu:



Hình 6.6: Thanh xử lý tín hiệu tương tự



Hình 6.7: Cấu trúc của module EM23

Tùy thuộc vào số kênh sử dụng trên module analog EM235 tương ứng với địa chỉ đầu vào (từ đơn) phải sử dụng trong quá trình lập trình: AWI0_cho channel 1, AWI2_cho channel 2, AWI4_cho channel 3.

Sau đây là chương trình gọi mở cho người sử dụng trong quá trình đo lường và giám sát nhiệt độ dựa trên hệ thống 1 module CPU, 1 module EM235, 1 cảm biến nhiệt điện Pt100 và 1 TD200(Text Display).

Module tiến hành đọc giá trị nhiệt điện trở được biến thành giá trị điện áp theo bậc. Đầu đầu ra analog được sử dụng như hằng số của nguồn dòng. Dòng cung cấp cho Pt100 là 12.5 mA nguồn dòng.

Với mạch này đầu vào là tuyến tính của 5mV/1°C. Giá trị analog của đầu vào được số hoá qua hệ thống biến đổi ADC và được đọc đều đặn theo chu kỳ. Từ giá trị này, chương trình sẽ thực hiện tính toán và chuyển đổi theo công thức sau:

$$T[°C] = (\text{Digital value} - 0°C \text{ offset}) / 1°C \text{ value}$$

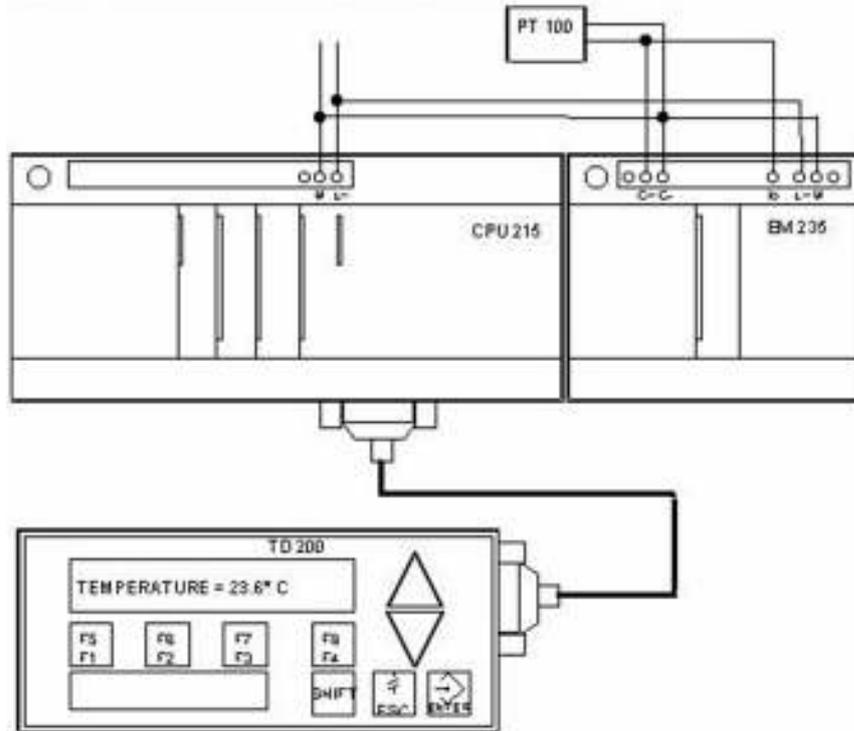
Digital value: giá trị đầu vào analog đã được chuyển đổi.

0°C offset: giá trị số, được đo ở 0°C; trong ví dụ này giá trị offset là 4000.

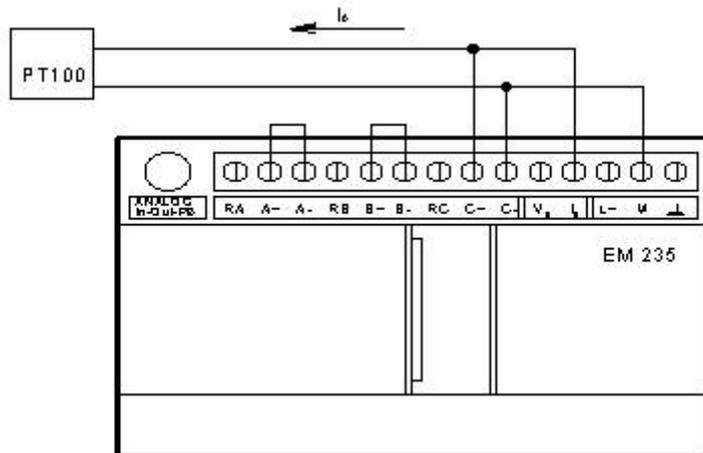
1°C value: giá trị tương ứng với 1°C, trong ví dụ này thì 1°C =16. Chương trình tính toán giá trị thập phân và ghi kết quả vào biến của message1:

"Temperature = xxx.x°C" giá trị này được hiển thị trên TD200. Trước khi khởi tạo chương trình này, phải xác định được giới hạn nhiệt độ thấp nhất và nhiệt cao nhất.

Nếu nhiệt độ vượt quá ngưỡng thì sẽ xuất hiện dòng cảnh báo trên TD200. Xuất hiện dòng thông báo Message 2: "Temperature > xxx.x°C" nếu nhiệt độ vượt quá ngưỡng. Message 3: "Temperature < xxx.x°C" nếu nhiệt độ dưới ngưỡng.



Hình 6.8: Cách lắp TD200 với CPU và module EM235.



Hình 6.9: Cách lắp ghép cảm biến với module EM235.

Chương trình viết trên Step 7 bằng ngôn ngữ STL:

Network 1: Set the High and Low Temperature Limits

```

LD First_Scan_On:SM0.1 // In the first scan cycle,
MOVD +0, VD196 // clear VW196 and VW198. MOVW +16, VW250
// Load 1° C = 16in VW250
MOVW +4000, VW252 // Set the 0° C offset = 4000. MOVW +300, VW260
// Set the high temperature , limit = 30° C.
MOVW +200, VW262 // Set the low temperature, limit = 20°C.

```

```

MOVW +20000, AQW0      // Initialize a 12.5 mA current
                        //at analog output word AQW0.

Network 2: Calculate the Value and Enable Message 1
LD  Always_On:SM0.0    // Every scan cycle,
MOVW AIW4, VW200       // move the value in analog,
                        // input word AIW4 to VW200.

-I  VW252, VW200       // Subtract the 0° C offset.
DIV  VW250, VD198      // Divide the result by the 1° C value.
MUL  +10, VD196        // Multiply the remainder by 10.
DIV  VW250, VD196      // Divide the value in variable double word
                        //VD196 (remainder x 10) by the 1° C value.

MOVW VW198, VW160      // Shift the quotient by 1 decimal point to the left.
MOVW +0, VW198         // Clear VW198.
MUL  +10, VD198        // Multiply the temperature value by0.
+I  VW160, VW200       // Add the result of temp value x10 with the value
                        // that is stored as the digit follo the decimal point.

MOVW VW200, VW116      // Transfer the result to VW116
                        // (embedded value on the TD 200) for display.

S   V12.7, 1          // Enable message 1 for display on the TD 200.

Network 3: If Temperature Exceeds High Limit, Enable Message 2 and Turn Off
Furnace
LDW >= VW200, VW260    // If the temperature value >= the high temperature
                        // limit stored in
VW260,=  V12.6        // enable message 2 on the TD 200.
R   Q0.0, 1           // Turn off the furnace.
MOVW VW260, VW136     // Move the high temperature limit value to VW136
                        //(embedded value on the TD 200) for display
                        // in message 2.

Network 4: If Temperature Drops Below Low Limit, Enable Message 3 and Turn
On Furnace
LDW <= VW200, VW262    // If the temperature value <= the low temperature
                        //limit stored in 262,
=   V12.5             // enable message 3 on the TD
200. S Q0.0, 1        // Turn on the furnace.
MOVW VW262, VW156     // Move the low temperature limit value to VW156
                        //(embedded value on the TD 200) for display in
                        // message
                        3.

Network 5: Main Program
End

```

BÀI TẬP THỰC HÀNH:

Mô hình

Giả lập một hệ thống dò mức nước lên tục, tín hiệu analog từ sensor dò mức nước gửi về ngõ vào PLC. Lập trình xác định mức nước đó.

Thiết bị

- PLC S7-200 CPU222/224/226.
- EM235.
- Màn hình HMI Siemens TP170A.
- Sensor dò mức Endress & Hauser.
- Mô hình.

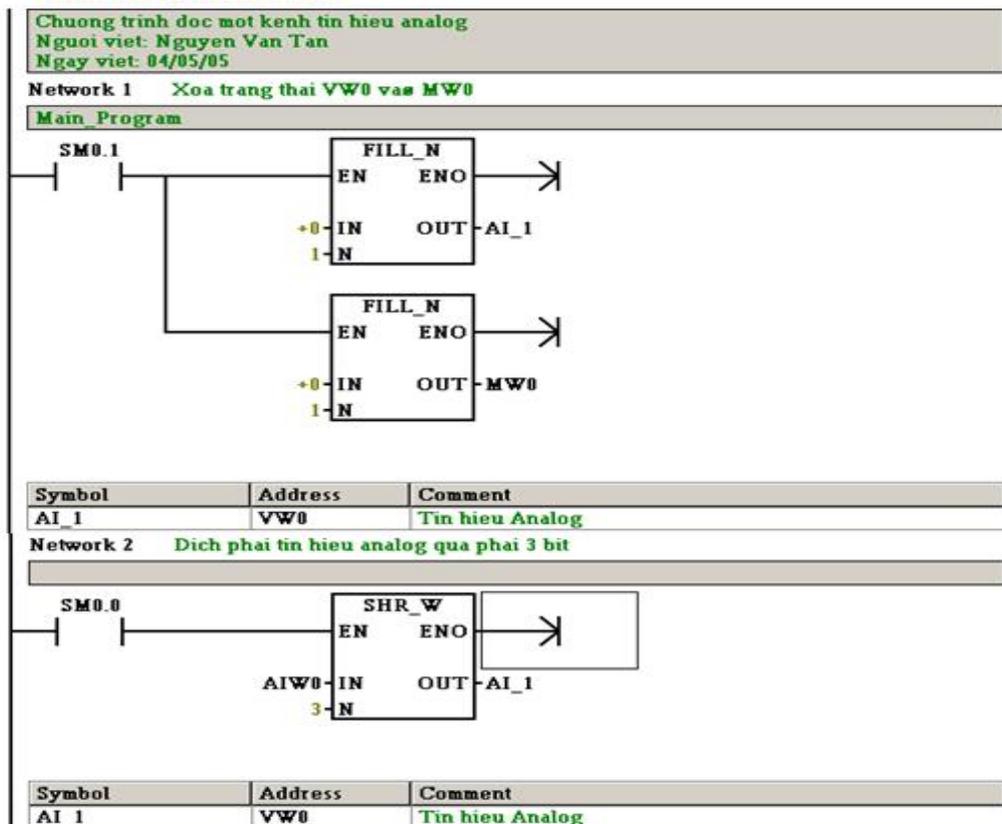
Kết nối hệ thống

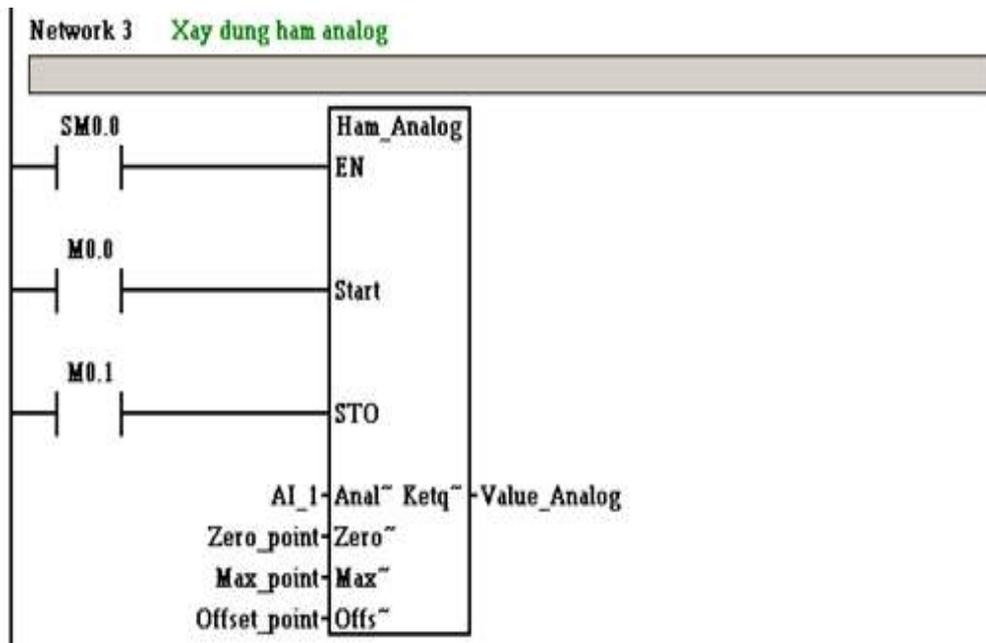
Thực hiện các bước:

- Kết nối module vào PLC.
- Kiểm tra nguồn, tín hiệu ngõ vào và ngõ ra.
- Sơ đồ nối dây analog.
- Lập trình PLC.
- Lập trình HMI.
- Kết nối HMI vào PLC.

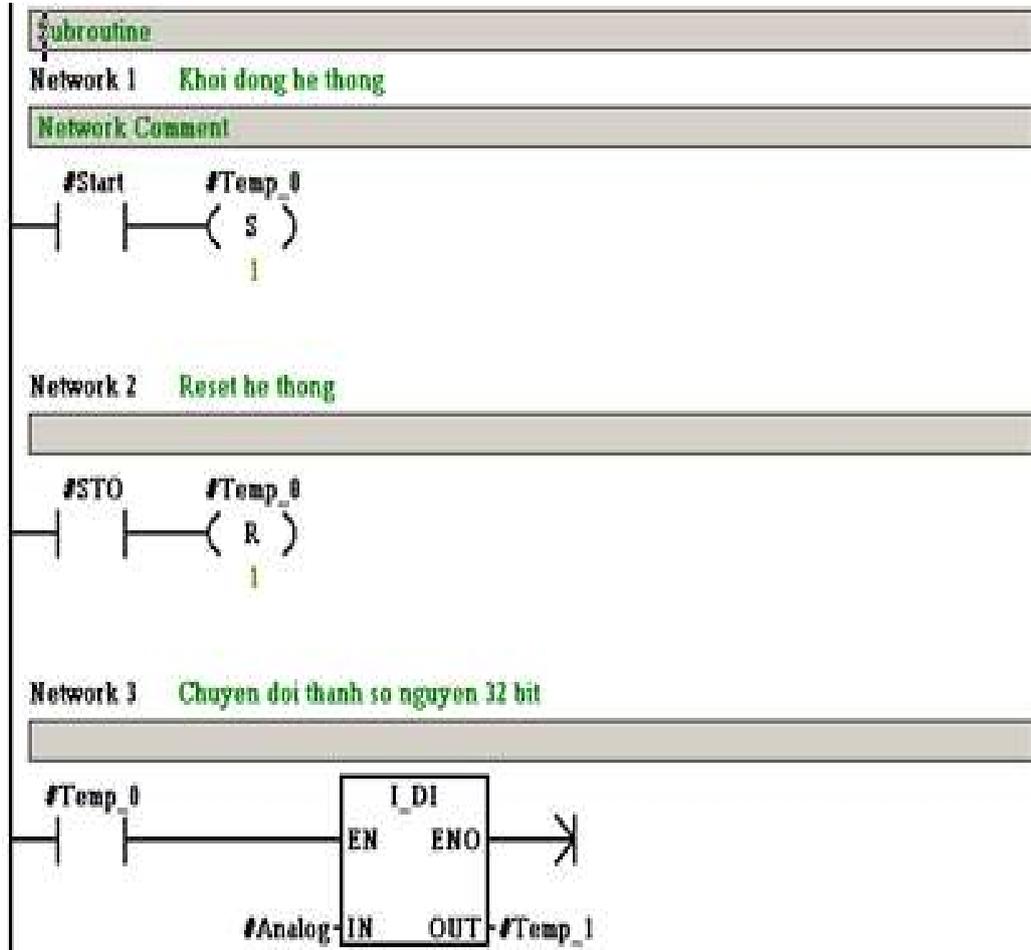
Chương trình PLC

Chương trình chính

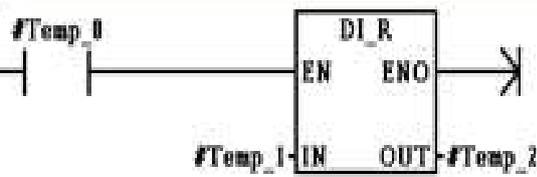




Chương trình con:



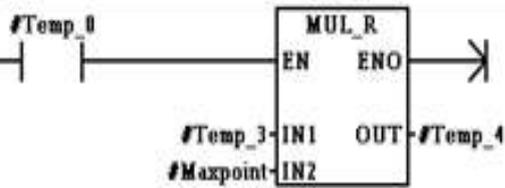
Network 4 Thanh số thực



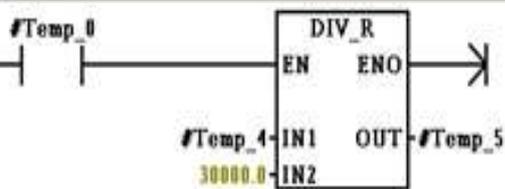
Network 5 Tru cho Zeropoint



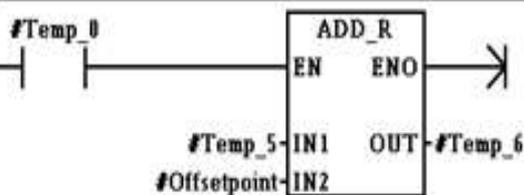
Network 6 Nhân với Maxpoint

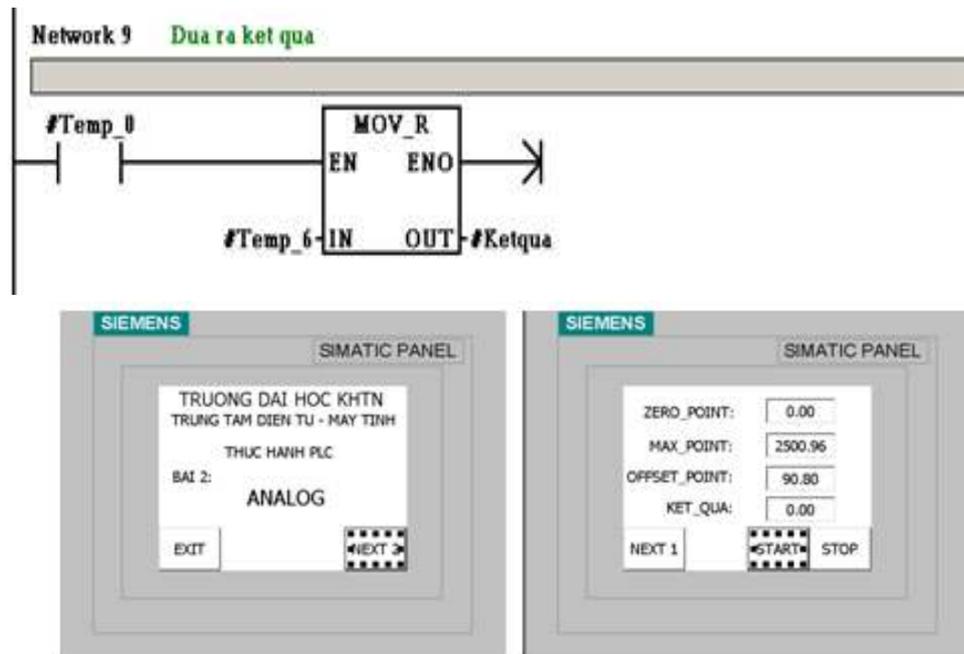


Network 7 Chia 30.000



Network 8 Cong Offset





Name	Type	Controller	Address	Acquisition cyc
START	BOOL	PLC_1	M 0.0	1.0
STOP	BOOL	PLC_1	M 0.1	1.0
OFFSET_POINT	REAL	PLC_1	VD 12	1.0
VALUE ANALOG	REAL	PLC_1	VD 16	1.0
ZERO_POINT	REAL	PLC_1	VD 4	1.0
MAX_POINT	REAL	PLC_1	VD 8	1.0

Hình 6.10: Giao diện màn hình HMI

Bài tập

1. Sử dụng modul EM235 và PLC CPU 224 viết chương trình đo nhiệt độ bằng cảm biến PT1000
2. Sử dụng modul EM235 và PLC CPU 224 viết chương trình đo điện áp từ 0Vdc đến 10Vdc.
3. Sử dụng modul EM235 và PLC CPU 224 viết chương trình đo dòng điện từ 0mA đến 20mA.

Bài 7:

CÁC BÀI TẬP ỨNG DỤNG TRONG ĐIỀU KHIỂN

GIỚI THIỆU

Nước ta hiện nay đang trong quá trình công nghiệp hóa, hiện đại hóa. Vì thế, tự động hóa sản xuất đóng vai trò quan trọng, tự động hóa giúp tăng năng suất, tăng độ chính xác và do đó tăng hiệu quả quá trình sản xuất. Để có thể thực hiện tự động hóa sản xuất, bên cạnh các máy móc cơ khí hay điện, các dây chuyền sản xuất...v.v, cũng cần thiết phải có các bộ điều khiển để điều khiển chúng. Trong đó, điều khiển lập trình là một trong các bộ điều khiển đáp ứng được yêu cầu đó.

MỤC TIÊU

+ Về kiến thức:

- Trình bày được cách kết nối giữa PLC và thiết bị ngoại vi, nắm được quy trình công nghệ của một số mô hình: mạch khởi động động cơ, mạch đảo chiều quay động cơ, điều khiển tốc độ và mạch mở máy sao/tam giác

+ Về kỹ năng:

- Thực hiện lập trình cho PLC đạt các yêu cầu về kỹ thuật, Xử lý các hư hỏng trên PLC đạt yêu cầu kỹ thuật, Thực hiện thay thế các hệ thống PLC đạt yêu cầu kỹ thuật.

+ Về thái độ:

- Rèn luyện tính tỉ mỉ, chính xác, đảm bảo an toàn và vệ sinh công nghiệp

NỘI DUNG CHÍNH

7.1 ON/OFF (Contact, coil, box)

Yêu cầu

Đóng cắt một tiếp điểm vào/ ra bằng nút ấn

Bảng đầu vào ra

1.1.1.1.1	Tên biến	Địa chỉ	Chí chú
	Bat	I0.0	Nút nhấn Bật
	Tat	I0.1	Nút nhấn Tắt
	IIT	Q0.0	Đầu ra bật/tắt hệ thống

Giản đồ thời gian



Hình 7.1: Giản đồ thời gian

7.2 MẠCH KHỞI ĐỘNG ĐỘNG CƠ

Nội dung

- Làm quen cách sử dụng phần mềm Step7 Microwin 40
- Phân tích các thành phần trong 1 đề án S7 200.
- Quản lý 1 đề án S7 200. Sọan thảo, lưu trữ , mở chương trình vv...

- Cách lập trình dạng LAD. Dạng LAD là dạng lập trình PLC cơ bản, gần giống với sơ đồ điều khiển role và tồn tại trong hầu hết các phần mềm lập trình của các hãng khác.
- Cách dùng địa chỉ tuyệt đối và địa chỉ thông qua ký hiệu.
- Thiết lập liên kết truyền thông PC/PPI (giữa PC và PLC). Download và Upload chương trình.
- Cách kết nối thiết bị ngoài vào inputs , outputs của PLC. Thực hành nội dung trên thông qua ví dụ mạch điều khiển khởi động động cơ.

Mục đích, yêu cầu

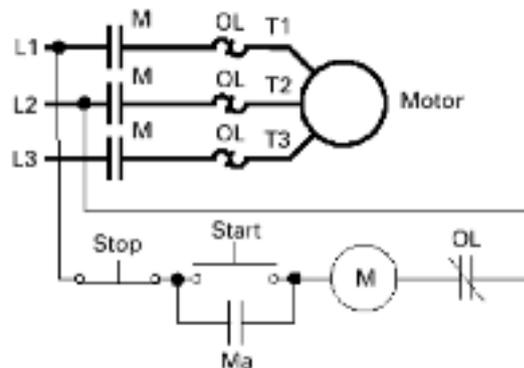
- Nắm những kỹ năng cơ bản nhất trong phần mềm Step7 Microwin 32 để chuẩn bị thực hành các bài sau.
- Hiểu rõ cách nối dây cho PLC.
- Nắm vững kiến thức và kỹ năng sử dụng một số lệnh logic căn bản và lập trình ở dạng LAD cho ứng dụng thực tế.

Thiết bị.

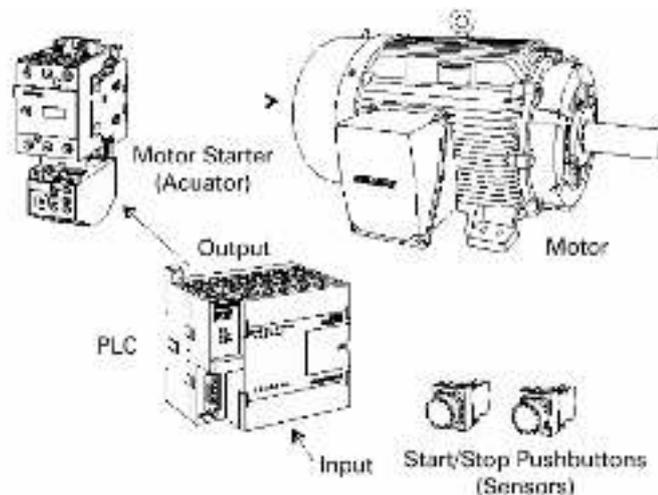
Máy tính + bộ thí nghiệm PLC + Cáp lập trình + Dây nối + Dụng Cụ.

Mạch điều khiển khởi động động cơ

Trong thí dụ này động cơ được khởi động (M) được mắc nối tiếp với một nút nhấn bình thường hở NO (nút Start), nút nhấn bình thường đóng NC (Stop) và các tiếp điểm bình thường đóng rơ-le quá tải (OL).



Hình 7.2: Mạch điều khiển



Hình 7.3: Kết nối PLC với động cơ thông qua relay trung gian

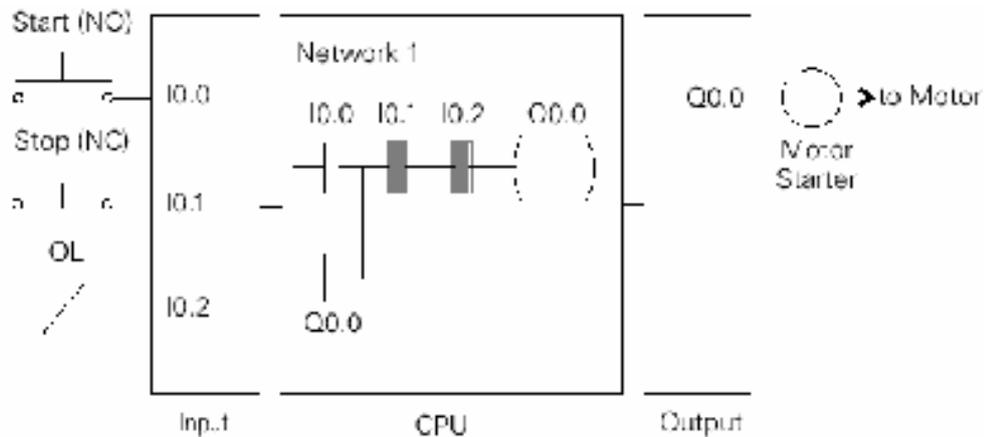
Khi nhấn Start thì có dòng điện đi qua mạch làm khởi động động cơ, nó làm đóng các tiếp điểm

M và Ma tương ứng của động cơ. Khi nhấn Start thì động cơ vẫn hoạt động do các tiếp điểm M, Ma đóng. Động cơ sẽ tiếp tục chạy cho đến khi nhấn nút Stop hay khi có quá tải làm mở các tiếp xúc OL.

Công việc điều khiển này được thực hiện qua PLC:

Chương trình PLC

Nút nhấn Start (NO) được nối vào ngõ vào thứ nhất I0.0, nút nhấn Stop (NC) nối vào ngõ vào thứ hai I0.1 và các tiếp điểm rò le quá tải OL được nối ngõ vào thứ ba I0.2. Một mạch AND 3 ngõ vào này tạo nên mạch điều khiển trong Network 1. Bit trạng thái I0.1 ở mức logic 1 vì nút Stop là loại NC; bit trạng thái I0.2 ở mức logic 1 vì các tiếp điểm OL đóng. Bộ điều khiển động cơ được nối vào ngõ ra Q0.0.



Hình 7.4: Kết nối và chương trình PLC

Các bước cần thực hiện

Bước 1: Đọc hiểu yêu cầu điều khiển và sơ đồ nối dây vào PLC.

Bước 2: Đọc hiểu sơ chương trình trên.

Bước 3: Nhập chương trình trên vào máy tính theo dạng LAD.

Bước 4: Biên dịch và download chương trình đã nhập xuống PLC.

Cho PLC chạy để thử điều khiển như yêu cầu đặt ra. (Chú ý là các ngõ xuất và nhập số được mô phỏng thông qua bộ digital simulator, do đó công tắc NC được mô phỏng qua việc đặt nó ở ON)

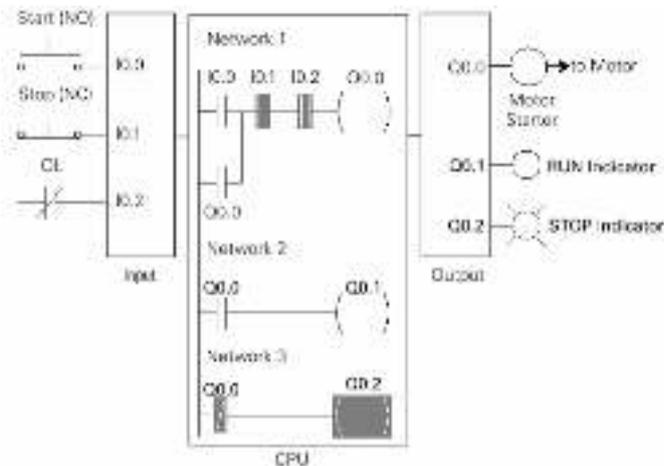
Bước 5: Học viên thử đổi các ngõ vào ra trong chương trình, ví dụ sử dụng Q1.0 thay vì Q0.0; I1.0 thay vì I0.0 rồi lặp lại bước 4.

Bước 6: Nhận xét.

Cải tiến mạch điều khiển động cơ có đèn báo

Ta thêm hai chỉ thị bên ngoài báo động cơ đang chạy hay dừng lại như sau:

Học viên hiệu đính chương trình cũ và thêm các ngõ ra điều khiển đèn. Với chương trình này, học viên cũng làm các bước như chương trình trên. Chương trình cải tiến được thực hiện như hình 7.5.



Hình 7.5: Kết nối và chương trình cải tiến của PLC

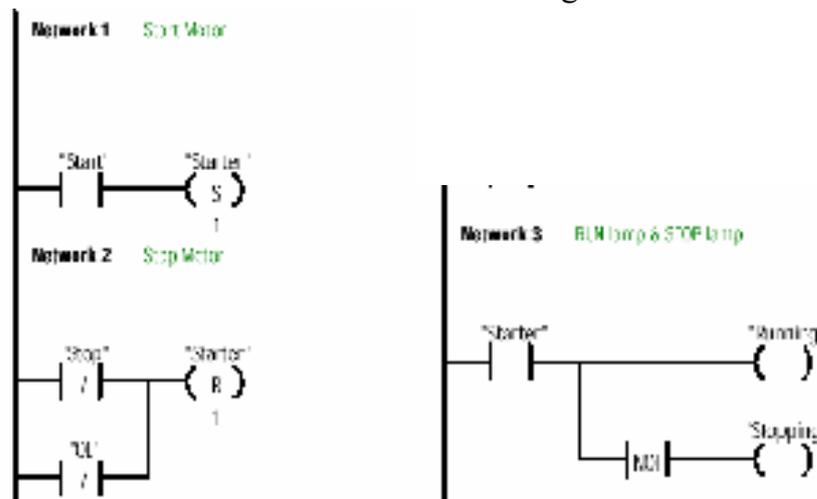
Viết lại chương trình sử dụng các lệnh S và R

Học viên viết lại chương trình bằng các lệnh S và R; và viết thêm các chú thích cho các network. Hãy đặt tên cho các ngõ xuất/nhập theo bảng ký hiệu sau:

Bảng 4.2 Xác lập ngõ vào/ra

Name	Address	Comment
Start	I0.0	NO contact
Starter	Q0.0	Contact (motor starter)
Stop	I0.1	NC contact
OL	I0.2	Thermal OverLoad (NC contact)
Running	Q0.1	RUN Lamp
Stopping	Q0.2	STOP lamp

Ta ghi chú thích vào các network và viết chương trình LAD như sau:



Hình 7.6 Chương trình dùng lệnh Set, Reset

Các bước cần thực hiện

Bước 1: Lập bảng ký hiệu và chọn chức năng cho phép hiển thị ký hiệu (Nhấn Ctr + Y, muốn tắt chức năng này thì nhấn như vậy 1 lần nữa).

Bước 2: Đọc hiểu sơ chương trình trên.

Bước 3: Nhập chương trình trên vào máy tính theo dạng LAD.

Bước 4: Biên dịch và download chương trình đã nhập xuống PLC.

Cho PLC chạy để thử điều khiển như yêu cầu đặt ra. (Chú ý là các ngõ xuất và nhập số được mô phỏng thông qua bộ digital simulator, do đó công tắc NC được mô phỏng qua việc đặt nó ở ON)

Bước 6: Nhận xét.

7.3 ĐẢO CHIỀU ĐỘNG CƠ

Nội dung

- Ôn lại cách sử dụng, thao tác trên phần mềm S7 Microwin 40.
- Các lệnh chốt: Set và Reset.
- Thực hành mạch điều khiển đảo chiều quay động cơ.

Mục đích, yêu cầu

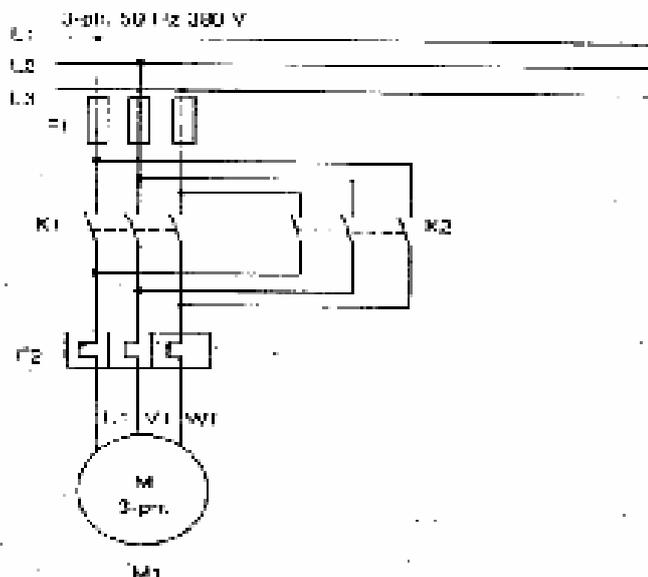
- Nắm vững một số lệnh logic và lập trình ở dạng LAD cho ứng dụng thực tế.
- Kết hợp các lệnh logic và Set , Reset để thực hiện 1 chương trình thực tế.

Thiết bị

Máy tính + bộ thí nghiệm PLC + Cáp lập trình + Dây nối + Dụng cụ điện.

Mạch contactor đảo chiều động cơ (đảo chiều trực tiếp)

Một động cơ AC 3 pha được khởi động cho chạy quay cùng chiều kim đồng hồ qua nút nhấn S1 (NO), để quay ngược chiều kim đồng hồ thì qua nút nhấn S2 (NO) và cho dừng bằng nút nhấn S0 (NC). Chiều quay của động cơ có thể được thay đổi không cần tắt động cơ.



Hình 7.7: Mạch động lực

Các đèn chỉ thị H1, H2 và H3 để chỉ trạng thái hoạt động của động cơ:

- H1 sáng để chỉ động cơ ngừng
- H2 sáng để chỉ động cơ quay cùng chiều kim đồng hồ
- H3 sáng để chỉ động cơ quay ngược chiều kim đồng hồ

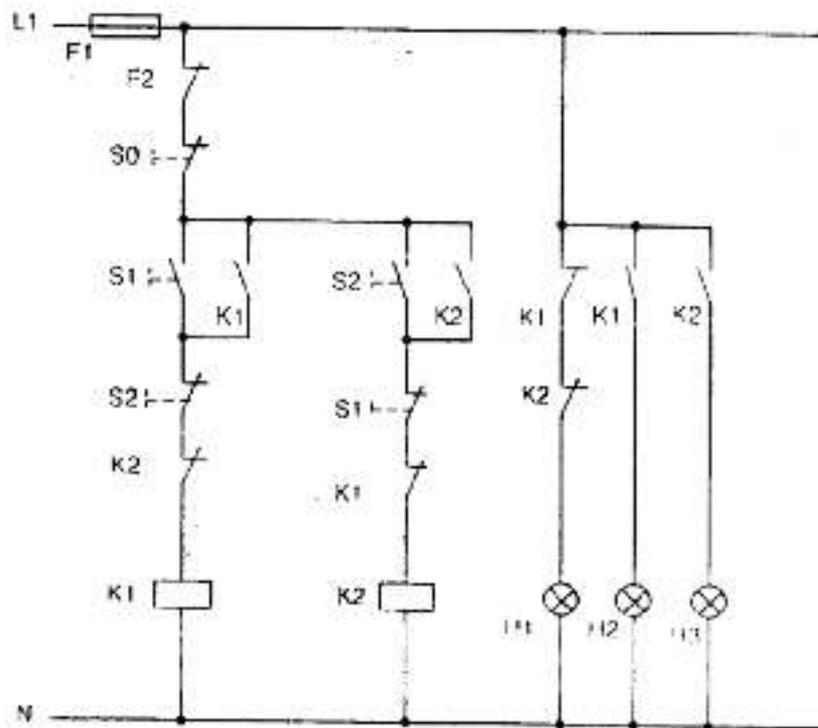
Động cơ được bảo vệ bằng các tiếp điểm quá tải nhiệt (F2), khi quá tải nhiệt thì hở mạch và làm ngưng động cơ.

Ta có sơ đồ mạch điện điều khiển như hình 4.34 và bảng 4.3 xác lập ngõ vào/ra, ta có thể gán các ký hiệu như sau:

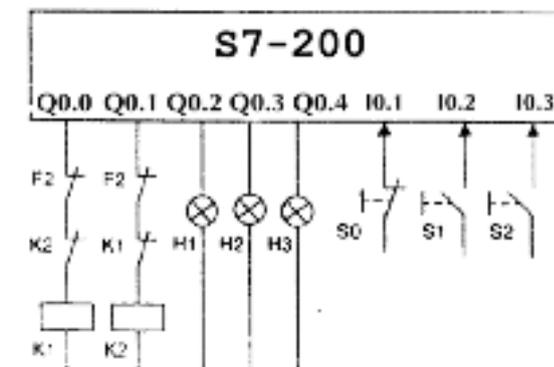
Bảng 4.3 Xác lập ngõ vào/ra

Name	Address	Comment
F2	I0.0	Thermal Overload
S0	I0.1	OFF (NC contact)
S1	I0.2	Quay cùng chiều kim đồng hồ (NO contact)
S2	I0.3	Quay ngược chiều kim đồng hồ (NO contact)
K1	Q0.0	Contactora của quay cùng chiều kim đồng hồ
K2	Q0.1	Contactora của quay ngược chiều kim đồng hồ
H1	Q0.2	OFF lamp (báo motor ngừng)
H2	Q0.3	Báo đang quay cùng chiều kim đồng hồ
H2	Q0.4	Báo đang quay ngược chiều kim đồng hồ

l-ph. 50 Hz 220 V



Hình 7.8: Sơ đồ điều khiển dùng khí cụ điện



Hình 7.9: Sơ đồ kết nối PLC

Các bước cần thực hiện

Bước 1: Đọc hiểu yêu cầu điều khiển và sơ đồ nối dây của PLC.

Bước 2: Viết chương trình PLC.

Bước 3: Nhập bảng ký hiệu và chọn chức năng hiển thị ký hiệu.

Bước 4: Nhập chương trình đã viết vào máy tính.

Bước 5: Biên dịch và download chương trình đã nhập xuống PLC.

Bước 6: Cho PLC chạy để thử điều khiển như yêu cầu đặt ra. (Chú ý là các ngõ vào và ra số được mô phỏng thông qua bộ digital simulator, do đó công tắc NC được mô phỏng qua việc đặt nó ở ON)

Bước 7: Nhận xét.

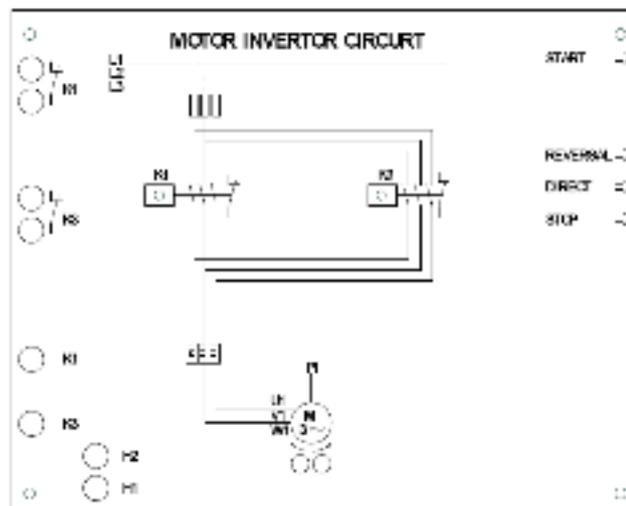
7.4 ĐẢO CHIỀU ĐỘNG CƠ (Set, reset)

Yêu cầu

Nút **Direct** = động cơ quay thuận

Nút **Reversal** = động cơ quay ngược

Nút **stop** = dừng động cơ

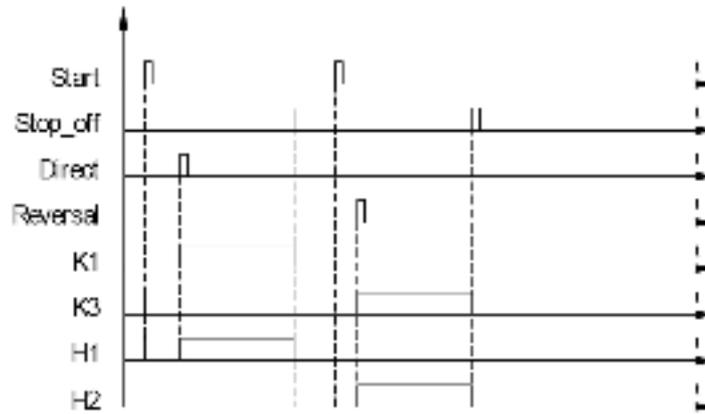


Hình 7.10: Mạch động lực

Bảng qui định ngõ vào/ra

2.1.1.1.1	Tên biến	Địa chỉ	Ghi chú
	Start	I0.0	Cáp nguồn (Bat)
	Stop_off	I0.1	Cáp nguồn (Tat)
	Direct	I0.2	Nút chạy thuận
	Reversal	I0.3	Nút chạy ngược
	K1	Q0.0	Contactơ K1
	K3	Q0.1	Contactơ K3
	H1	Q0.2	Chỉ thị DC đang chạy thuận
	H2	Q0.3	Chỉ thị DC đang chạy ngược
	Mean1	V0.0	Bit trung gian bat/tat
	Mean2	V0.1	Bit trung gian "Chạy thuận"
	Mean3	V0.2	Bit trung gian "Chạy ngược"

Giản đồ thời gian

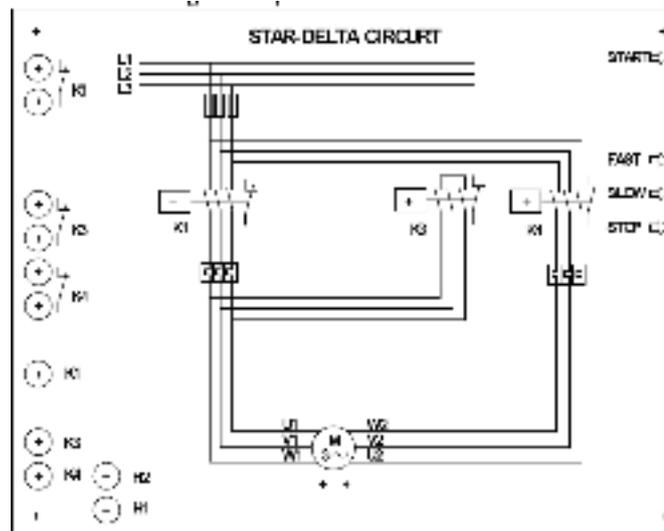


Hình 7.11: Giải đồ thời gian

7.5 KHỞI ĐỘNG SAO TAM GIÁC (Timer)

Mô hình và Yêu cầu

Mục đích: giảm điện áp khi khởi động động cơ để đảm bảo dòng khởi động không ảnh hưởng đến điện lưới.



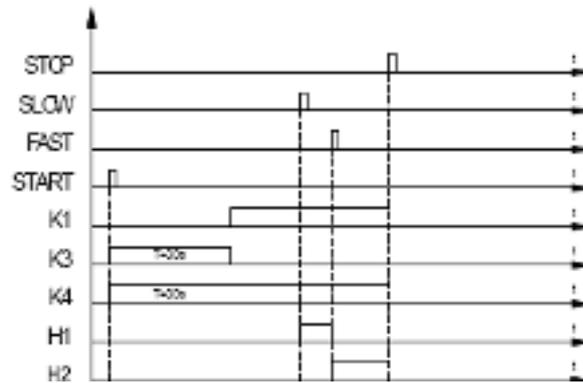
Hình 7.12: Mạch động lực

Nhấn nút START (S6) hệ thống đóng contactor K3 và K4: động cơ hoạt động chế độ đầu sao (điện áp làm việc 220VAC). Sau thời gian 30 giây hệ thống ngắt contactor K3 và chuyển sang đóng contactor K1, động cơ hoạt động ở chế độ đầu tam giác (điện áp làm việc 380VAC). Các role K1, K3, K4 cho phép đấu nối motor thực bên ngoài.

Bảng đầu vào ra

Symbol	Address	Comment
STOP	I0.0	Dừng
SLOW	I0.1	Chạy chậm
FAST	I0.2	Chạy nhanh
START	I0.3	Khởi động động cơ sao tam giác
K1	Q0.0	Contactơ K1
K3	Q0.1	Contactơ K3
K4	Q0.2	Contactơ K4
H1	Q0.4	Đèn báo chạy chậm
H2	Q0.5	Đèn báo chạy nhanh

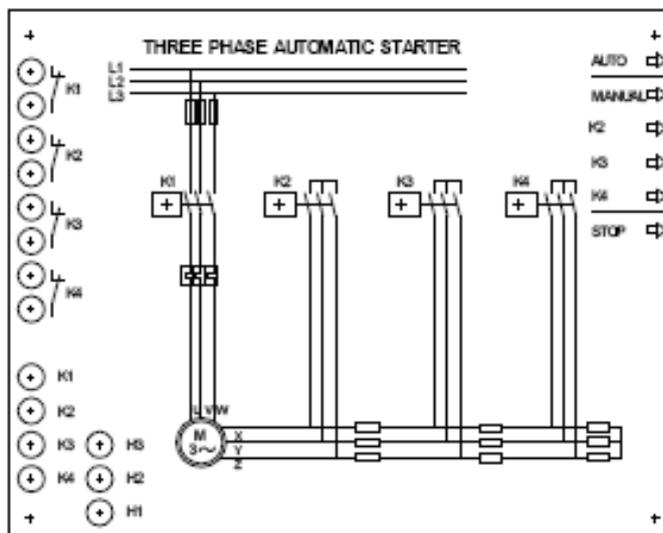
Giải đồ thời gian



Hình 7.13: Mạch động lực

7.6 TỰ ĐỘNG ĐÓNG CÁC CẤP ĐIỆN TRỞ (Timer)

Mô hình và Yêu cầu



Hình 7.14: Mạch động lực

Mục đích là giảm dòng khởi động hoặc thay đổi tốc độ động cơ.

Chế độ tự động: nhấn nút AUTO (S5) thì hệ thống đóng contactor K1, sau 10s đóng K4, và 10s tiếp theo đóng K3, 10s tiếp theo đóng K2. Nhấn STOP thì các contactor K1 đến K4 đều mở.

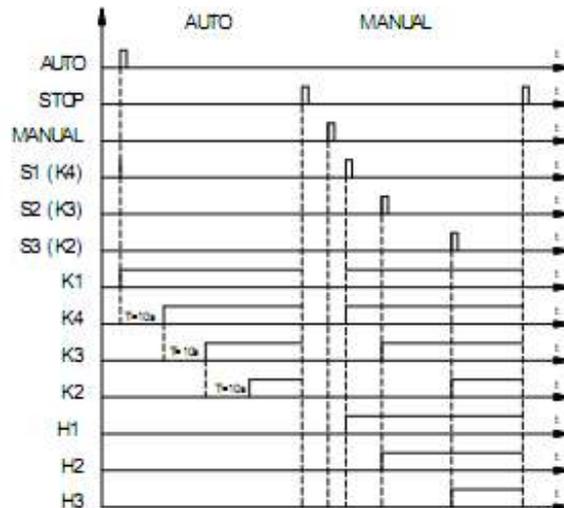
Chế độ tay: Nhấn MANUAL để chuyển sang chế độ bằng tay, muốn đóng cấp điện trở nào nhấn nút tương ứng K2, K3, K4 (S3, S2, S1) lúc này contactor K1 đóng kèm theo với contactor K2 hoặc K3 hoặc K4.

Bảng đầu vào ra

Symbol	Address	Comment
AUTO	I0.0	Chế độ tự động
STOP	I0.1	Chế độ dừng
MANUAL	I0.2	Chế độ bằng tay

S1(K4)	I0.3	Nút nhấn S1 tác động contactor K4
S2(K3)	I0.4	Nút nhấn S2 tác động contactor K3
S3(K2)	I0.5	Nút nhấn S3 tác động contactor K2
K1	Q0.0	Đóng contactor K1
K2	Q0.1	Đóng contactor K2
K3	Q0.2	Đóng contactor K3
K4	Q0.3	Đóng contactor K4
H1	Q0.4	Đèn báo đóng contactor K2 ở chế độ MANUAL
H2	Q0.5	Đèn báo đóng contactor K3 ở chế độ MANUAL
H3	Q0.6	Đèn báo đóng contactor K4 ở chế độ MANUAL

Giản đồ thời gian



Hình 7.15: Giản đồ thời gian

7.7 BẢNG TẢI 1

Nội dung

Phân tích và viết chương trình điều khiển băng tải.

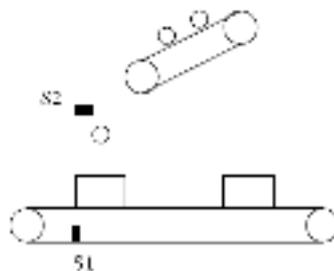
Mục đích, yêu cầu

Sử dụng các lệnh với Timer, Counter.

Thiết bị

Máy tính + bộ demo PLC + Cáp lập trình + Dây nối + Dụng cụ.

Nguyên tắc hoạt động



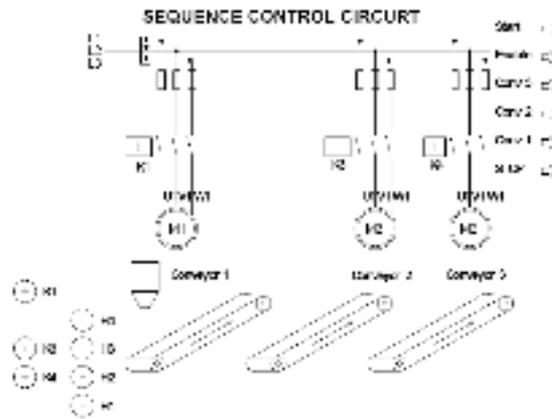
Hình 7.16: Mô hình

Ban đầu băng chuyền thùng và băng chuyền sản phẩm không hoạt động. Khi nhấn “Start”, băng chuyền thùng hoạt động. Khi thùng gặp cảm biến S1 thì băng chuyền thùng dừng lại, băng chuyền sản phẩm hoạt động. Khi 50 sản phẩm vào thùng, thì băng chuyền sản phẩm dừng lại, băng chuyền thùng hoạt động. Và quá trình cứ tiếp tục như vậy. Khi được 10 thùng thì cả băng chuyền thùng lẫn băng chuyền sản phẩm dừng lại,

Cho PLC chạy để quan sát các đèn. Kiểm tra xem chương trình chạy có đúng không. Nếu không đúng như yêu cầu thì kiểm tra lại chương trình.

7.8 BẢNG TẢI 2

Mô hình và yêu cầu



Hình 7.17: Mạch động lực

- Sử dụng trong các nhà máy chuyển cát, hệ thống trộn bê tông nhựa nóng hay định lượng phối liệu trong nhà máy xi măng ...

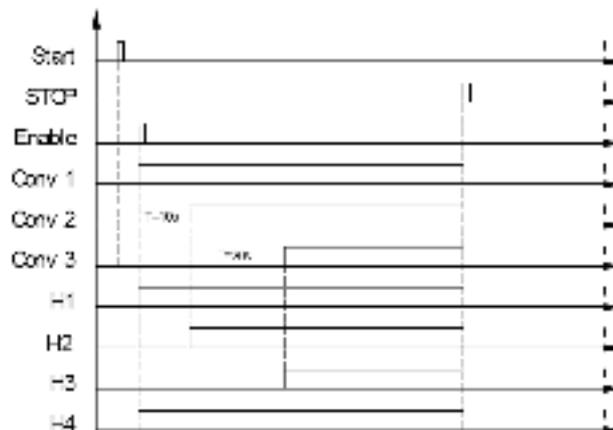
Chế độ tự động: Nút ENABLE: bảng tải thứ nhất hoạt động, 10s tiếp theo thì bảng tải thứ 2 hoạt động, 10s nữa bảng tải thứ 3.

Chế độ tay: Muốn bảng tải nào hoạt động thì ấn nút CONV của bảng tải đó. Để chuyển đổi giữa hai chế độ: nhấn STOP rồi nhấn START trở lại.

Bảng đầu vào ra

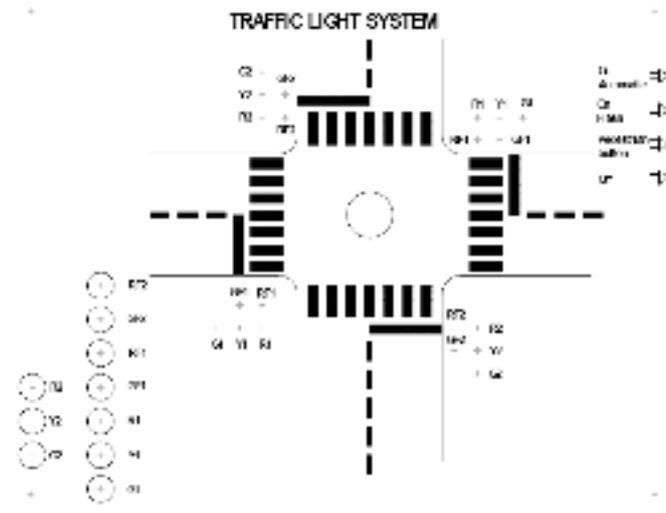
Symbol	Address	Comment
Start	I0.0	Nút nhấn start
STOP	I0.1	Nút nhấn stop
Enable	I0.2	Cho phép hoạt động tự động
Conv 1	Q0.0	Bảng tải 1 (K1)
Conv 2	Q0.1	Bảng tải 2 (K2)
Conv 3	Q0.2	Bảng tải 3 (K3)
H1	Q0.4	Đèn báo Conv 1
H2	Q0.5	Đèn báo Conv 2
H3	Q0.6	Đèn báo Conv 3
H4	Q0.7	Đèn báo ở chế độ tự động

Giải đồ thời gian



Hình 7.18: Giải đồ xung

7.9 ĐÈN GIAO THÔNG (sử dụng Timer, compare, SM,M) Mô hình và yêu cầu



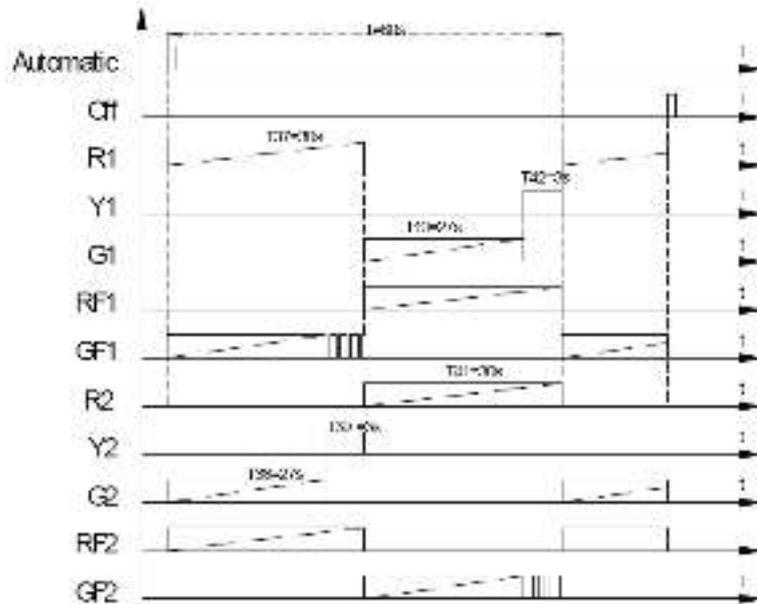
Hình 7.18: Mô hình

Điều khiển đèn giao thông ngã tư. Thời gian sáng đèn đỏ, đèn vàng và đèn xanh lần lượt là 30s, 3s và 27s. đèn xanh đi bộ sáng khi đèn đỏ cùng phía sáng và nhấp nháy với chu kỳ 1s khi đèn vàng cùng phía sáng.

Bảng đầu vào/ra

Symbol	Address	Comment
Automatic	I0.0	Nút nhấn start
Off	I0.1	Nút nhấn stop
R1	Q0.0	Đỏ 1
Y1	Q0.1	Vàng 1
G1	Q0.2	Xanh 1
R1F1	Q0.3	Xanh đi bộ 1
GF1	Q0.4	Đỏ đi bộ 1
R2	Q0.5	Đỏ 2
Y2	Q0.6	Vàng 2
G2	Q0.7	Xanh 2
RF2	Q1.0	Xanh đi bộ 2
GF2	Q1.1	Đỏ đi bộ 2

Giản đồ thời gian

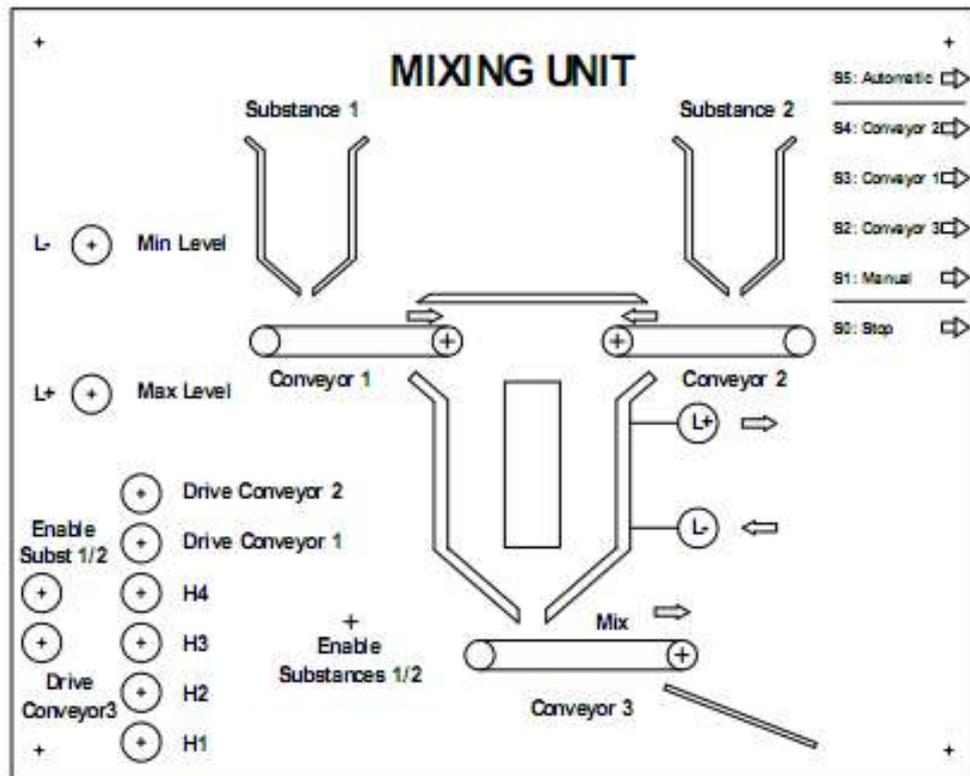


Hình 7.20: Tín hiệu điều khiển

7.10 Bài tập

1. Trộn phối liệu

Mô hình và Yêu cầu



Hình 7.21: Mô hình

Hệ thống định lượng phối liệu các loại phụ gia trong nhà máy xi măng, định lượng và trộn phối liệu trong máy trộn bê tông...

Chế độ tay: (Manual) Lúc này ta có thể điều khiển độc lập các băng tải (Conveyor) 1, 2 hoặc 3 bằng các nút S1, S2, S3 đèn báo tương ứng là H2, H3, H4.

Chế độ tự động: (Auto) tín hiệu điều khiển mở van *Enable Substances 1/2* tác

động để băng tải 1 và 2 hoạt động, đến khi liệu trong xilo đạt mức cao L+ thì dừng. Tiếp đó băng tải 3 hoạt động đưa phối liệu ra ngoài đến khi liệu trong xilo tới mức thấp L- thì dừng, van *Enable Substances 1/2* tác động trở lại.

2. Động cơ chạy tuần tự

Yêu cầu công nghệ:

Nhấn nút Start: động cơ 1 chạy, sau 3s động cơ 2 chạy, sau 5s động cơ 3 chạy

Nhấn nút Stop: động cơ 3 dừng, sau 2s động cơ 2 dừng, sau 4s động cơ 1 dừng

Yêu cầu thực hành:

- Vẽ giản đồ thời gian
- Vẽ mạch động lực và sơ đồ kết nối PLC với thiết bị ngoại vi
- Viết chương trình điều khiển

TÀI LIỆU THAM KHẢO

- [1]. Giáo trình PLC (Lưu hành nội bộ)
- [2]. Giáo trình PLC căn bản - Tổng cục dạy nghề
- [3]. Tự động hóa với SIMATIC S7-200 – NXB Hà Nội.
- [4]. Tự động hóa với simatic s7-200 - Nguyễn Doãn Phước - NXB Nông nghiệp.
- [5]. Kỹ thuật điều khiển lập trình - Trung tâm Việt Đức Trường ĐHSPKT.
- [6]. Automatisieren mit sps - Guenter, Wellenreuther, Dieter Zastrow. NXB Viweg.