

Th.S Châu Chí Đức

Kỹ thuật điều khiển

lập trình PLC
SIMATIC S7-200



Thành phố Hồ Chí Minh
10-2008

LỜI NÓI ĐẦU

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

Hiện nay tài liệu để giảng dạy và tham khảo về kỹ thuật điều khiển lập trình còn khá hạn chế. Tài liệu "kỹ thuật điều khiển lập trình PLC Simatic S7-200", là quyển sách đầu tiên trong bộ sách về kỹ thuật điều khiển lập trình PLC họ SIMATIC S7, được biên soạn với mong muốn góp một phần nhỏ vào việc giảng dạy và tự học về kỹ thuật điều khiển lập trình của giáo viên, học sinh, sinh viên và đọc giả quan tâm về PLC họ SIMATIC S7-200 của công ty Siemens.

Tài liệu được chia thành 2 tập. Tập 1 bao gồm các phần cơ bản phù hợp với các bạn mới bắt đầu làm quen với PLC, tuy nhiên nó cũng có thể là tài liệu tham khảo cho các bạn đã có kiến thức cơ bản về PLC. Tập 2 là phần nâng cao tập trung về các vấn đề điều khiển số, truyền thông và màn hình điều khiển. Cấu trúc chung của các tập sách là ở mỗi chương trong các phần đều có ví dụ minh họa cho các mục, ngoài ra cuối mỗi chương có thêm một số câu hỏi và bài tập để đọc giả rèn luyện thêm.

Dù có một thời gian dài làm việc và giảng dạy về kỹ thuật điều khiển lập trình PLC họ SIMATIC, mạng truyền thông công nghiệp và truyền động của hãng Siemens cho rất nhiều đối tượng khác nhau cũng như đã rất cố gắng trong quá trình biên soạn nhưng tài liệu không tránh khỏi thiếu sót. Rất mong được sự góp ý chân thành của quý đọc giả để giúp tài liệu được hoàn thiện hơn. **Thư từ góp ý xin gửi về địa chỉ: ccduc2006@gmail.com.** Xin cảm ơn.

LỜI TÂM SỰ

Tập 1 "kỹ thuật điều khiển lập trình PLC Simatic S7-200" đã được viết xong từ rất lâu. Nhưng vì nghĩ đến việc in ấn và phát hành quá nhiều khê, giá thành lại cao và phải chờ đợi thời gian rất lâu tập sách này mới đến tay bạn đọc, nên tác giả đã hoãn lại. Nghĩ rằng cung cấp cho đọc giả, các bạn học sinh, sinh viên và giáo viên thêm một tài liệu tham khảo để làm phong phú thêm kiến thức về tự động hóa là việc nên làm. Vì vậy tác giả chọn phương án phát hành qua mạng và truyền tay dưới dạng tập tin với phương châm " sách hữu ích thì mới có nhuận bút".

Các bạn thân mến!

Việc biên soạn tài liệu về kỹ thuật, nhất là kỹ thuật mới, đòi hỏi người biên soạn ngoài kinh nghiệm chuyên môn còn bỏ rất nhiều thời gian và công sức. Do đó sẽ là một niềm động viên vô cùng to lớn cho tác giả để tiếp tục hoàn thành tập 2, bộ sách về kỹ thuật điều khiển lập trình PLC SIMATIC S7-300/400, các tài liệu khác liên quan đến PLC họ SIMATIC, truyền thông công nghiệp, truyền động của hãng Siemens nếu được sự động viên từ tinh thần đến vật chất. Nếu thấy sách này giúp ích cho các bạn thì khi các bạn sở hữu nó (có được từ bất kỳ phương tiện nào) ở dạng tập tin hoặc được in ra ở dạng sách, xin vui lòng động viên tác giả bằng cách chuyển tiền vào **tài khoản số 49809449 cho CHÂU CHÍ ĐỨC, ngân hàng Thương mại Á Châu (ACB) chi nhánh Châu Văn Liêm** với số tiền tùy theo ý của các bạn.

Nếu các bạn có những ý động viên khác xin gửi thông tin cho tác giả qua địa chỉ mail ccduc2006@gmail.com.

Cám ơn sự động viên của đọc giả.

Mục lục

1	Tổng quan về điều khiển	1
1.1	Khái niệm chung về điều khiển	1
1.2	Cấu trúc một qui trình điều khiển	2
1.3	Các loại điều khiển	3
1.4	Hệ thống số	4
1.5	Các khái niệm xử lý thông tin	5
1.5.1	Bit	5
1.5.2	Byte	5
1.5.3	Word	6
1.5.4	DoubleWord	6
2	Bộ điều khiển lập trình PLC – Cấu trúc và phương thức hoạt động	7
2.1	Giới thiệu	7
2.2	Sự khác nhau giữa hệ điều khiển bằng relay và hệ điều khiển bằng PLC	8
2.3	Cấu trúc của một PLC	11
2.4	Các khối của PLC	13
2.4.1	Khối nguồn cung cấp	13
2.4.2	Bộ nhớ chương trình	14
2.4.3	Khối trung tâm (CPU)	15
2.4.4	Khối vào	15
2.4.5	Khối ra	16
2.4.6	Các khối đặc biệt	16
2.5	Phương thức thực hiện chương trình trong PLC	16
3	Cảm biến và cơ cấu chấp hành trong điều khiển logic	19
3.1	Cảm biến	19
3.1.1	Giới thiệu	19
3.1.2	Nối dây cho cảm biến	19
3.1.2.1	Switch	20
3.1.2.2	Ngõ ra TTL	20
3.1.2.3	Ngõ ra Sinking/Sourcing	20
3.1.2.4	Ngõ ra Solid state relay	23
3.1.3	Phát hiện đối tượng	23
3.1.3.1	Chuyển mạch tiếp xúc	23
3.1.3.2	Reed Switches	23
3.1.3.3	Cảm biến quang (Optical Sensor)	23
3.1.3.4	Cảm biến điện dung (Capacitive Sensor)	25
3.1.3.5	Cảm biến điện cảm (Inductive Sensor)	26
3.1.3.6	Cảm biến siêu âm (Ultrasonic sensor)	28
3.1.3.7	Hiệu ứng Hall (Hall Effect)	28
3.1.3.8	Lưu lượng (Fluid Flow)	28
3.1.4	Tóm tắt	29
3.2	Cơ cấu chấp hành	29
3.2.1	Giới thiệu	29

3.2.2 Solenoid	29
3.2.3 Van điều khiển (VALVE)	30
3.2.4 Xy lanh (CYLINDER)	32
3.2.5 Động cơ	33
3.2.6 Các cơ cấu chấp hành khác.....	34
4 Bộ điều khiển lập trình PLC Simatic S7-200	35
4.1 Cấu hình cứng	35
4.1.1 Khối xử lý trung tâm	35
4.1.2 Khối mở rộng	39
4.1.2.1 Digital module	39
4.1.2.2 Analog module	40
4.1.2.3 Intelligent module	41
4.1.2.4 Function module	41
4.2 Màn hình điều khiển	42
4.3 Các vùng nhớ	43
4.4 Qui ước địa chỉ trong PLC S7-200	46
4.4.1 Truy xuất theo bit	46
4.4.2 Truy xuất theo byte (8 bit)	46
4.4.3 Truy xuất theo word (16 bit)	46
4.4.4 Truy xuất theo 2 word (Double word = 32 bit)	47
4.5 Xử lý chương trình	48
5 Kết nối dây giữa PLC và thiết bị ngoại vi	51
5.1 Kết nối dây giữa PLC và các thiết bị ngoại vi	51
5.1.1 Giới thiệu CPU 224 và cách kết nối với thiết bị ngoại vi	51
5.1.2 Kết nối với máy tính	52
5.1.3 Nối nguồn cung cấp cho CPU	54
5.1.4 Kết nối vào/ra số với ngoại vi	54
5.1.4.1 Kết nối các ngõ vào số với ngoại vi	55
5.1.4.2 Kết nối các ngõ ra số với ngoại vi	57
5.2 Kiểm tra việc kết nối dây bằng phần mềm	60
5.2.1 Status Chart	60
5.2.2 Giám sát và thay đổi biến với Status Chart	60
5.2.3 Cường bức biến với Status Chart	62
5.2.4 Ứng dụng Status Chart trong việc kiểm tra kết nối dây trong S7-200	63
5.3 Câu hỏi và bài tập	64
6 Phần mềm Micro/Win và ngôn ngữ lập trình	65
6.1 Cài đặt phần mềm STEP 7-Micro/WIN	65
6.1.1 Yêu cầu hệ điều hành và phần cứng	65
6.1.2 Cài đặt phần mềm	65
6.2 Các phần tử cơ bản trong chương trình PLC S7-200	66
6.2.1 Chương trình chính OB1 (main program)	66
6.2.2 Chương trình con SUB (subroutine)	66
6.2.3 Chương trình ngắt INT(interrupt routine)	67
6.2.4 Khối hệ thống (system block)	67

6.2.5	Khối dữ liệu (data block)	67
6.3	Ngôn ngữ lập trình	67
6.3.1	Dạng hình thang: LAD (Ladder logic)	68
6.3.2	Dạng khối chức năng: FBD (Function Block Diagram)	68
6.3.3	Dạng liệt kê lệnh: STL (StaTement List)	69
6.4	Soạn thảo chương trình với phần mềm STEP7-Micro/Win V4.0 SP6	69
6.4.1	Mở màn hình soạn thảo chương trình	69
6.4.1.1	Vùng soạn thảo chương trình	70
6.4.1.2	Cây lệnh	70
6.4.1.3	Thanh chức năng	70
6.4.2	Thanh công cụ (Toolbar) trong STEP7-Micro/WIN	75
6.4.3	Tạo một dự án STEP 7-Micro/WIN	77
6.4.3.1	Tạo dự án mới	77
6.4.3.2	Lưu dự án	77
6.4.3.3	Mở một dự án	78
6.4.4	Thư viện	78
6.4.5	Hệ thống trợ giúp trong STEP 7-Micro/WIN	79
6.4.6	Xóa bộ nhớ CPU	80
6.4.7	Mở một dự án đang tồn tại sẵn	80
6.4.8	Kết nối truyền thông S7-200 với thiết bị lập trình	81
6.4.9	Tải dự án từ PLC	82
6.4.9.1	Tải một khối hoặc ba khối	82
6.4.9.2	Tải vào một dự án mới hoặc dự án rỗng	82
6.4.9.3	Tải vào một dự án tồn tại	82
6.4.9.4	Thủ tục tải dự án từ PLC về thiết bị lập trình	82
6.4.10	Nạp (download) một dự án vào PLC	83
6.4.11	Thiết lập cấu hình chung cho phần mềm (menu option và customize)	85
6.4.11.1	Menu Option	85
6.4.11.2	Menu Custommize	86
6.4.12	Soạn thảo chương trình	88
7	Các phép toán logic	95
7.1	Ngăn xếp (logic stack) trong S7-200	95
7.2	Các phép toán logic cơ bản	96
7.2.1	Phép toán AND	96
7.2.2	Phép toán OR	97
7.2.3	Tổ hợp các cổng AND và OR	98
7.2.3.1	AND trước OR	98
7.2.3.2	OR trước AND	98
7.2.4	Phép toán XOR	99
7.3	Xử lý các tiếp điểm, cảm biến được nối với ngõ vào PLC	100
7.4	Ví dụ ứng dụng các liên kết logic	102
7.4.1	Mạch tự duy trì ưu tiên mở máy	102
7.4.2	Mạch tự duy trì ưu tiên dừng máy	103
7.4.3	Điều khiển ON/OFF động cơ có chỉ báo	104
7.4.4	Điều khiển đảo chiều quay động cơ	106
7.5	Bit nhớ M (bit memory)	109

7.6	Các lệnh SET, RESET và mạch nhớ RS	111
7.6.1	Lệnh SET	111
7.6.2	Lệnh RESET (R)	112
7.6.3	Mạch nhớ R-S	112
7.6.3.1	Ưu tiên SET (khâu SR)	112
7.6.3.2	Ưu tiên RESET (khâu RS)	113
7.6.4	Các qui tắc khi sử dụng Set và Reset	114
7.6.5	Ví dụ ứng dụng mạch nhớ R-S	114
7.7	Các lệnh nhận biết cạnh tín hiệu và lệnh NOT	118
7.7.1	Lệnh NOT	118
7.7.1	Các lệnh nhận biết cạnh tín hiệu	118
7.8	Các Bit nhớ đặc biệt (Special Memory bits)	120
7.9	Câu hỏi và bài tập	121
8	Thiết kế theo logic Bool & biểu đồ Karnaugh	125
8.1	Giới thiệu	125
8.2	Đại số BOOL	125
8.3	Thiết kế Logic	127
8.3.1	Các kỹ thuật đại số Bool	131
8.4	Các dạng logic chung	132
8.4.1	Dạng cổng phức	132
8.4.2	Multiplexers	132
8.5	Một số ví dụ thiết kế đơn giản với đại số bool	133
8.5.1	Các chức năng logic cơ bản	133
8.5.2	Hệ thống an toàn xe hơi	134
8.5.3	Quay phải/trái động cơ	134
8.5.4	Cảnh báo trộm	135
8.6	Biểu đồ Karnaugh	136
8.6.1	Giới thiệu	136
8.7	Câu hỏi và bài tập	139
9	Bộ định thời (Timer)	147
9.1	Giới thiệu	147
9.2	Timer đóng mạch chậm TON	148
9.3	Timer đóng mạch chậm có nhớ TONR	149
9.4	Timer mở mạch chậm TOF	150
9.5	Ứng dụng Timer	152
9.5.1	Tạo xung có tần số theo mong muốn	152
9.5.2	Tạo Timer xung và timer xung có nhớ	152
9.5.2.1	Timer xung (Pulse timer)	152
9.5.2.2	Timer xung có nhớ (Extended Pulse timer)	153
9.5.3	Đảo chiều quay động cơ có khống chế thời gian	154
9.5.4	Chiếu sáng Garage	155
9.5.5	Thiết bị rót chất lỏng vào thùng chứa	156
9.6	Câu hỏi và bài tập	161
10	Bộ đếm (Counter)	170
10.1	Giới thiệu	170
10.2	Bộ đếm lên CTU (Count Up)	171

10.3	Bộ đếm xuống CTD (Count Down)	172
10.4	Bộ đếm lên-xuống CTUD (Count Up/Down)	173
10.5	Ứng dụng bộ đếm	174
10.5.1	Đếm sản phẩm được đóng gói	174
10.5.2	Kiểm soát chỗ cho Garage ngầm	175
10.6	Câu hỏi và bài tập	178
11	Điều khiển trình tự	181
11.1	Cấu trúc chung của một chương trình điều khiển	181
11.2	Điều khiển trình tự	182
11.2.1	Giới thiệu	182
11.2.2	Phương pháp lập trình điều khiển trình tự	184
11.3	Các thủ tục tổng quát để thiết kế bài toán trình tự	186
11.4	Cấu trúc của bài toán điều khiển trình tự	188
11.4.1	Chuỗi trình tự	188
11.4.2	Kiểu hoạt động	188
11.4.3	Các thông báo	190
11.4.4	Kích hoạt ngõ ra	190
11.5	Các ký hiệu	190
11.6	Bước trình tự	191
11.7	Các lệnh biểu diễn trong sơ đồ chức năng	193
11.8	Các chế độ hoạt động, cảnh báo và xuất lệnh	197
11.8.1	Bảng điều khiển	198
11.8.2	Các khâu chế độ hoạt động có cảnh báo	199
11.8.3	Hiện thị bước trình tự	201
11.8.4	Xuất lệnh	201
11.9	Các ví dụ ứng dụng	201
11.9.1	Máy phay đơn giản	201
11.9.2	Bảng chuyển đếm táo	205
11.10	Câu hỏi và bài tập	210
12	An toàn trong PLC	218
12.1	Khái niệm và mục đích	218
12.2	Hư hỏng ở PLC	218
12.3	Các quan điểm về kỹ thuật an toàn ở PLC	219
12.3.1	Các lỗi nguy hiểm và không nguy hiểm	219
12.3.2	Các cách giải quyết cho hoạt động an toàn của thiết bị điều khiển PLC	220
12.4	Bảo vệ các ngõ ra PLC	223
12.4.1	Bảo vệ ngõ ra dùng Transistor	224
12.4.2	Bảo vệ ngõ ra Role có nguồn điều khiển DC	224
12.4.3	Bảo vệ ngõ ra Role và ngõ ra AC có nguồn điều khiển AC	224
12.5	Câu hỏi và bài tập	225
13	Chuyển điều khiển kết nối cứng sang điều khiển bằng PLC	226
13.1	Kết nối ngõ vào/ ra của PLC từ một sơ đồ điều khiển có tiếp điểm	226
13.2	Chuyển đổi điều khiển từ contactor thành PLC	228

13.2.1	Điều khiển thiết bị bù công suất phản kháng	230
13.2.2	Thiết bị nghiền	237
13.3	Điều khiển khí nén	241
13.3.1	Máy uốn thanh kim loại	242
13.3.2	Máy doa miệng ống kim loại	246
13.4	Câu hỏi và bài tập	253
14	Các phép toán cơ bản trong điều khiển số	257
14.1	Các dạng số trong PLC	257
14.1.1	Kiểu dữ liệu Integer (INT)	257
14.1.2	Kiểu dữ liệu Double Integer (DINT)	258
14.1.3	Kiểu dữ liệu số thực (REAL)	259
14.1.4	Kiểu dữ liệu số BCD (Binary Coded Decimal)	260
14.2	Chức năng sao chép	261
14.2.1	Các lệnh sao chép, trao đổi nội dung	261
14.2.2	Các lệnh sao chép một mảng lớn dữ liệu	263
14.3	Phép toán so sánh	264
14.4	Phép toán số học	266
14.4.1	Cộng và trừ	266
14.4.2	Nhân và chia	267
14.4.3	Ví dụ phép toán số học	268
14.5	Tăng và giảm thanh ghi	269
14.6	Các phép toán logic số	271
14.6.1	Các logic số trong S7-200	271
14.6.2	Ứng dụng	272
14.6.2.1	Che vị trí các bit	272
14.6.2.2	Chèn thêm bit	273
14.7	Chức năng dịch/quay thanh ghi	273
14.7.1	Chức năng dịch chuyển thanh ghi	273
14.7.1.1	Dịch trái	273
14.7.1.2	Dịch phải	274
14.7.2	Chức năng quay thanh ghi	275
14.7.2.1	Quay trái	276
14.7.2.2	Quay phải	277

1 Tổng quan về điều khiển

1.1 Khái niệm chung về điều khiển

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

Sự điều khiển được phân biệt theo các đặc điểm khác nhau:

* Theo loại biểu diễn thông tin

- *Điều khiển nhị phân*: Xử lý tín hiệu đầu vào nhị phân (tín hiệu 1-0) thành các tín hiệu ra nhị phân.

- *Điều khiển số*: Xử lý các thông tin số, có nghĩa các thông tin được biểu diễn dưới dạng số.

* Theo loại xử lý tín hiệu

- *Điều khiển liên kết*: Các trạng thái tín hiệu xác định của ngõ ra được điều khiển bởi các trạng thái tín hiệu của ngõ vào tùy thuộc vào các chức năng liên kết (AND, OR, NOT).

- *Điều khiển trình tự*: Điều khiển với trình tự theo từng bước, sự đóng mạch của một bước sau xảy ra phụ thuộc vào điều kiện đóng mạch tiếp theo. Điều kiện đóng mạch tiếp theo có thể phụ thuộc vào qui trình hay thời gian.

- *Điều khiển không đồng bộ*: Việc điều khiển được xử lý ở sự thay đổi trực tiếp của tín hiệu ngõ vào không cần tín hiệu xung phụ (điều khiển chậm).

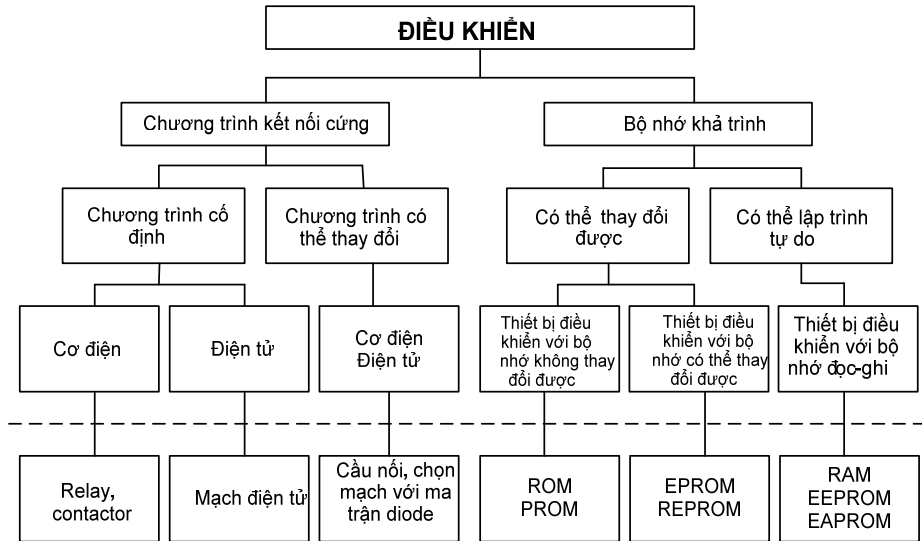
- *Điều khiển đồng bộ xung*: Việc điều khiển được xử lý ở các tín hiệu chỉ đồng bộ với một tín hiệu xung (điều khiển nhanh).

* Theo loại thực hiện chương trình

- *Điều khiển theo chương trình kết nối cứng*: Loại điều khiển này có thể được lập trình cố định, có nghĩa không thể thay đổi được ví dụ như lắp đặt dây nối cố định hay có thể thay đổi chương trình thông qua các đầu nối (ma trận diode).

- *Điều khiển khả trình*: Chức năng điều khiển được lưu giữ trong một bộ nhớ chương trình. Nếu sử dụng bộ nhớ đọc/ghi (RAM), thì có thể thay đổi chương trình mà không cần can thiệp đến phần cơ khí (điều khiển có thể lập trình tự do). Nếu ngược lại là một bộ nhớ chỉ đọc (ROM), thì chương trình có thể

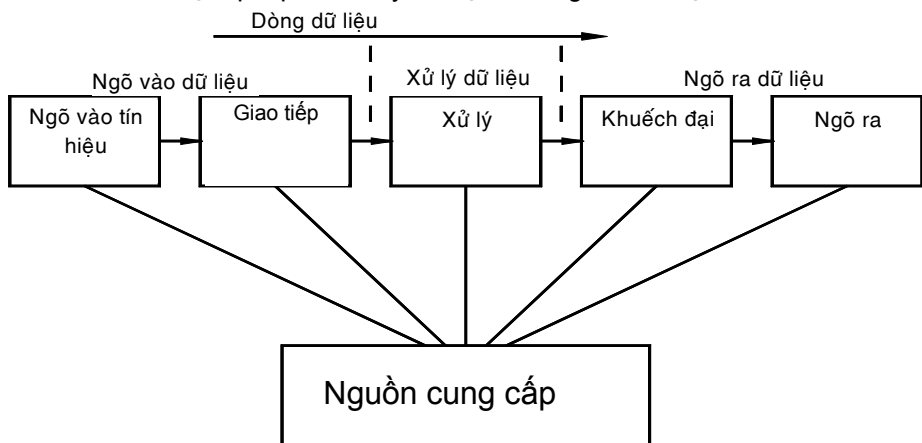
được thay đổi bằng cách thay đổi bộ nhớ (điều khiển có thể thay đổi chương trình).



Hình 1.1: Sơ đồ các loại điều khiển

1.2 Cấu trúc một qui trình điều khiển

Mỗi sự điều khiển được chia ra làm 3 bộ phận hợp thành: Ngõ vào dữ liệu (ngõ vào tín hiệu), Xử lý dữ liệu (xử lý tín hiệu cũng như các liên kết) và ngõ ra dữ liệu (ngõ ra tín hiệu). Dòng dữ liệu trong một sự điều khiển xảy ra từ đầu vào dữ liệu qua phần xử lý dữ liệu đến ngõ ra dữ liệu.



Hình 1.2: Cấu trúc chung của một qui trình điều khiển

+ **Ngõ vào tín hiệu:** Bao gồm các loại tín hiệu của các bộ phát tín hiệu như nút nhấn, công tắc hành trình, cảm biến điện dung, cảm biến điện cảm .v.v..

- Tuỳ thuộc vào loại điều khiển, các tín hiệu có thể là nhị phân, số hay tín hiệu tương tự.
- + Giao tiếp:** Phần này cần thiết, nếu tín hiệu của một hệ thống lạ cần phải được xử lý. Một bộ phận chuyển đổi từ tín hiệu ngõ vào thành tín hiệu phù hợp với mức của tín hiệu xử lý được đặt ở phần giao tiếp.
- + Xử lý:** Toàn bộ các liên kết, trình tự thời gian, các chức năng nhớ, đếm .v.v.. được thực hiện trong phần này.
- Phần xử lý là phần chính của tất cả các hệ thống điều khiển. Các kỹ thuật điều khiển có tiếp điểm như khởi động từ phụ, relay thời gian, kỹ thuật điều khiển bằng mạch điện tử (như AND, OR, NOT ...) được PLC hay máy tính điều khiển quá trình tổng hợp tại đây.
- + Khuếch đại:** Các tín hiệu từ phần xử lý có mức độ công suất bé được khuếch đại lớn lên nhiều lần ở đây để có thể điều khiển các khởi động từ, van từ hay các đối tượng điều khiển khác và các đèn báo.
- + Ngõ ra:** Phần này được kết nối với đối tượng điều khiển mà có ảnh hưởng trực tiếp đến quá trình điều khiển (ví dụ: Khởi động từ, van từ, thyristor, v.v..)

1.3 Các loại điều khiển

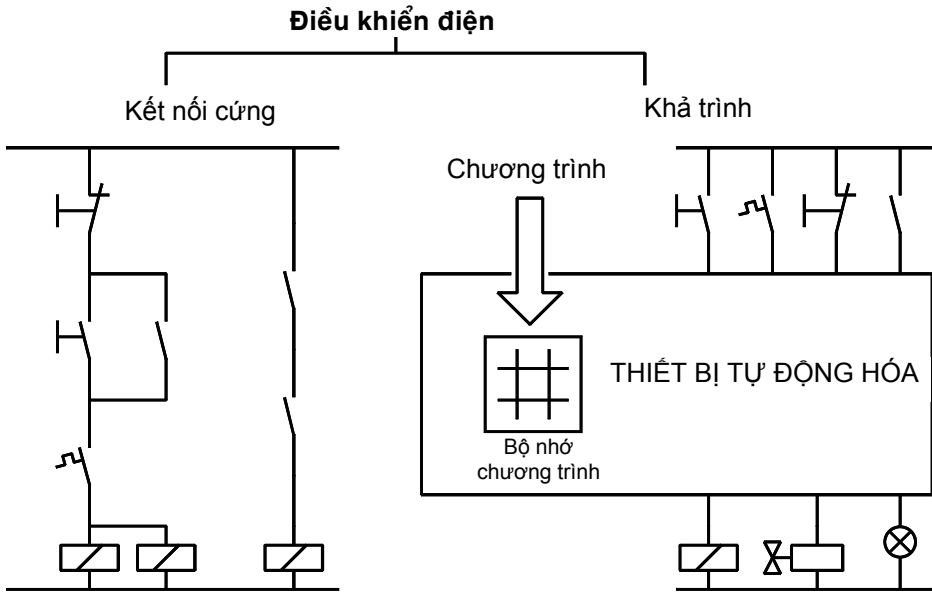
Trong kỹ thuật điều khiển cũng như tự động hóa, người ta chia ra làm hai loại điều khiển: điều khiển kết nối cứng và điều khiển khả trình.

* Điều khiển kết nối cứng

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

* Điều khiển khả trình (PLC)

Điều khiển khả trình là loại điều khiển mà chức năng của nó được đặt cố định thông qua một chương trình còn gọi là bộ nhớ chương trình. Sự điều khiển bao gồm một thiết bị điều khiển mà ở đó tất cả các bộ phát tín hiệu cần thiết và đối tượng điều khiển được kết nối cho một chức năng cụ thể. Nếu chức năng điều khiển cần được thay đổi, thì chỉ phải thay đổi chương trình bằng thiết bị lập trình ở đối tượng điều khiển tương ứng hay cắm một bộ nhớ chương trình đã lập trình khác vào trong điều khiển.



Hình 1.3: Hai loại điều khiển trong sản xuất

1.4 Hệ thống số

Trong xử lý các phần tử nhớ, các ngõ vào, các ngõ ra, thời gian, các ô nhớ v.v... bằng PLC thì hệ thập phân không được sử dụng mà là hệ thống số nhị phân (hệ hai trị).

*** Hệ nhị phân**

Hệ nhị phân chỉ có các số 0 và 1, có thể được đọc và biểu diễn giá trị dễ dàng trong kỹ thuật. Giá trị định vị của một số nhị phân là số mũ của hai. Độ lớn của số thông thường được biểu diễn ở dạng mã BCD (Binary-Code-Decimal). Đối với mỗi số Decimal được viết với số nhị phân 4 vị trí.

*** Số thập lục phân (Hexadecimal)**

Hệ thập lục phân có 16 ký hiệu khác nhau từ 0-9 và A-F. Giá trị định vị của một số thập lục phân số mũ của 16.

- Hệ nhị phân: Chữ số: 0,1

Giá trị định vị = Số mũ của cơ số 2

2^3	2^2	2^1	2^0
8	4	2	1

Ví dụ:

$$\begin{array}{cccc}
 & 1 & 1 & 0 & 1 \\
 & \swarrow & \swarrow & \swarrow & \swarrow \\
 1 \cdot 2^3 & + & 1 \cdot 2^2 & + & 0 \cdot 2^1 & + & 1 \cdot 2^0 \\
 8 & + & 4 & + & 0 & + & 1 & = & 13_D
 \end{array}$$

- Hệ thập lục phân: chữ số: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E;F

Giá trị định vị = Số mũ của cơ số 16

16^3	16^2	16^1	16^0
4096	256	16	1

Ví dụ:

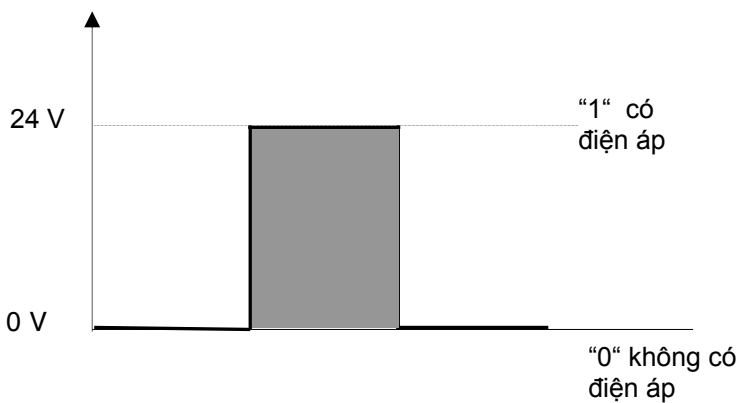
$$\begin{array}{ccc}
 & 2 & A & B \\
 & \swarrow & \swarrow & \swarrow \\
 2 \cdot 16^2 & + & A \cdot 16^1 & + & B \cdot 16^0 \\
 512 & + & 160 & + & 11 & = & 683_D
 \end{array}$$

1.5 Các khái niệm xử lý thông tin

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.

1.5.1 Bit

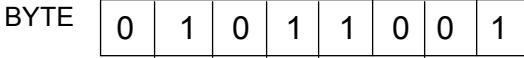
Bit là đơn vị thông tin nhị phân nhỏ nhất, có thể có giá trị 0 hoặc 1.



Hình 1.4: Một bit có thể có trạng thái tín hiệu “1” hoặc “0”

1.5.2 Byte

1 Byte gồm có 8 Bit

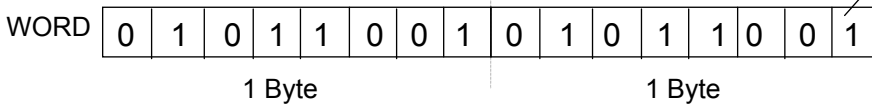


Trạng thái tín hiệu

1.5.3 Word

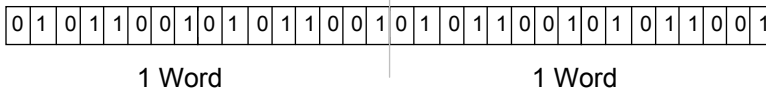
1 Word gồm có 2 Byte hay 16 Bit. Với Word có thể biểu diễn ở các dạng: số nhị phân, ký tự hay câu lệnh điều khiển.

Trạng thái tín hiệu

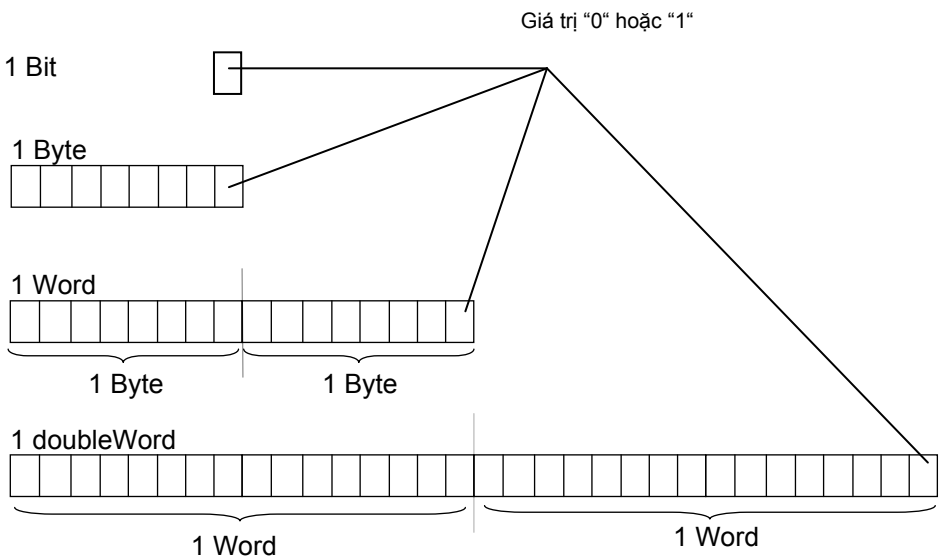


1.5.4 DoubleWord

1 DoubleWord gồm có 4 Byte hay 32 Bit. Với DoubleWord có thể biểu diễn ở các dạng: số nhị phân, ký tự hay câu lệnh điều khiển.



Tóm tắt:



2 Bộ điều khiển lập trình PLC – Cấu trúc và phương thức hoạt động

2.1 Giới thiệu

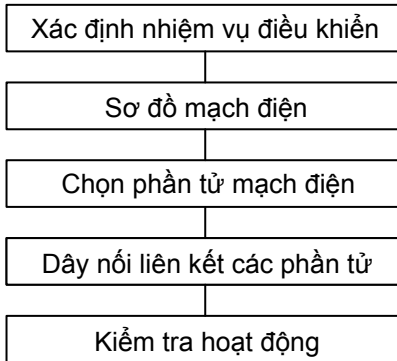
Các thành phần của kỹ thuật điều khiển điện và điện tử ngày càng đóng một vai trò vô cùng to lớn trong lĩnh vực tự động hóa ngày càng cao. Trong những năm gần đây, bên cạnh việc điều khiển bằng Relay và khởi động từ thì việc điều khiển có thể lập trình được càng phát triển với hệ thống đóng mạch điện tử và thực hiện lập trình bằng máy tính. Trong nhiều lĩnh vực, các loại điều khiển cũ đã được thay đổi bởi các bộ điều khiển có thể lập trình được, có thể gọi là các bộ điều khiển logic khả trình, viết tắt trong tiếng Anh là **PLC** (**P**rogrammable **L**ogic **C**ontroller).

Sự khác biệt cơ bản giữa điều khiển logic khả trình (thay đổi được qui trình hoạt động) và điều khiển theo kết nối cứng (không thay đổi được qui trình hoạt động) là: *Sự kết nối dây không còn nữa, thay vào đó là chương trình.*

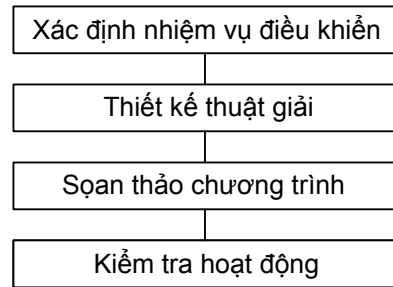
Có thể lập trình cho PLC nhờ vào các ngôn ngữ lập trình đơn giản. Đặc biệt đối với người sử dụng không cần nhờ vào các ngôn ngữ lập trình khó khăn, cũng có thể lập trình PLC được nhờ vào các liên kết logic cơ bản.

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

Các bước thiết lập hệ điều khiển bằng relay điện



Các bước thiết lập hệ điều khiển bằng PLC



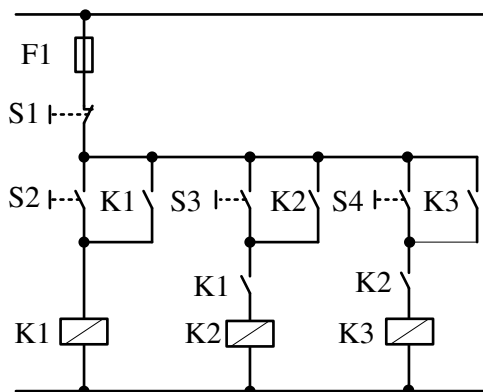
Khi thay đổi nhiệm vụ điều khiển thì người ta thay đổi mạch điều khiển: Lắp lại mạch, thay đổi các phần tử mới ở hệ điều khiển bằng relay điện. Trong khi đó khi thay đổi nhiệm vụ điều khiển ở hệ điều khiển logic khả trình (PLC) thì người ta chỉ thay đổi chương trình soạn thảo.

2.2 Sự khác nhau giữa hệ điều khiển bằng relay và hệ điều khiển bằng PLC

Sự khác nhau giữa hệ điều khiển bằng relay và hệ điều khiển bằng PLC có thể minh họa một cách cụ thể như sau:

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

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



Hình 2.1: Mạch điều khiển trình tự 3 máy bơm

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

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

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

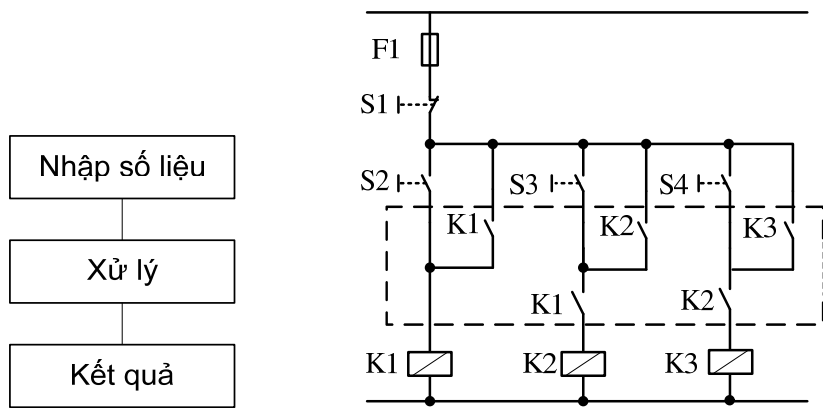
- *Phần tử vào:* Các nút nhấn S1, S2, S3, S4 vẫn giữ nguyên.
- *Phần tử ra:* Ba khởi động từ K1, K2, K3, để đóng và mở ba máy bơm vẫn giữ nguyên.
- *Phần tử xử lý:* Được thay thế bằng PLC.

Sơ đồ kết nối với PLC được cho như ở hình 2.3. Trình tự đóng mở theo yêu cầu đề ra sẽ được lập trình, chương trình sẽ được nạp vào bộ nhớ.

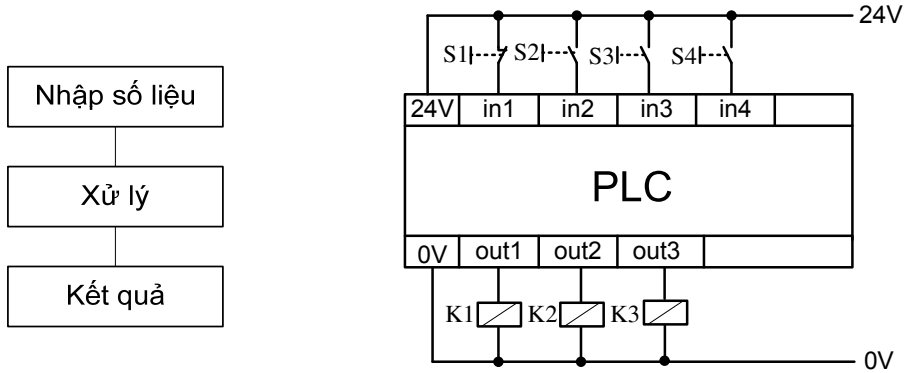
Bây giờ giả thiết rằng nhiệm vụ điều khiển sẽ thay đổi. Hệ thống ba máy bơm vẫn giữ nguyên, nhưng trình tự được thực hiện như sau: chỉ đóng được hai trong ba máy bơm hoặc mỗi máy bơm có thể hoạt động một cách độc lập. Như vậy theo yêu cầu mới đối với hệ thống điều khiển bằng relay điện phải thiết kế lại mạch điều khiển, sơ đồ lắp ráp phải thực hiện lại hoàn toàn mới. Sơ đồ mạch điều khiển biểu diễn như hình 2.4.

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

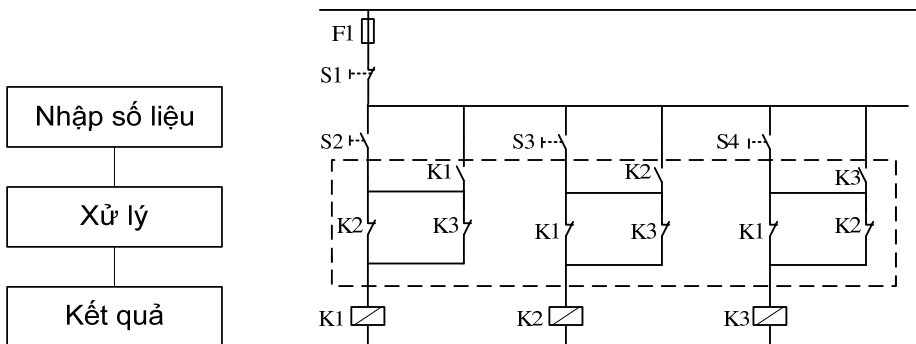
Nếu ta thay đổi hệ điều khiển trên bằng hệ điều khiển lập trình PLC, khi nhiệm vụ điều khiển thay đổi thì thực hiện sẽ nhanh hơn và đơn giản hơn bằng cách thay đổi lại chương trình.



Hình 2.2: Sơ đồ mạch được chuyển thành chương trình trong PLC



Hình 2.3: Sơ đồ kết nối với PLC



Hình 2.4: Sơ đồ mạch điều khiển 3 động cơ đã được thay đổi

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

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

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

- Điều khiển thang máy.
- Điều khiển các quá trình sản xuất khác nhau: sản xuất bia, sản xuất xi măng v.v

- Hệ thống rửa ô tô tự động.
- Thiết bị khai thác .
- Thiết bị đóng gói bao bì, tự động mạ và tráng kẽm v.v ...
- Thiết bị sấy.
- ...

2.3 Cấu trúc của một PLC

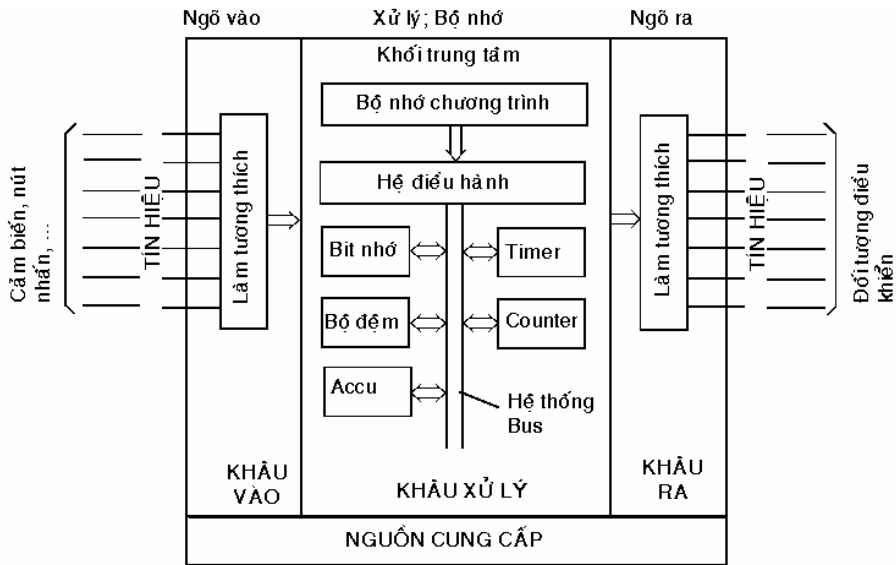
Các bộ điều khiển PLC được sản xuất theo dòng sản phẩm. Khi mới xuất xưởng, chúng chưa có một chương trình cho một ứng dụng nào cả. Tất cả các cổng logic cơ bản, chức năng nhớ, timer, counter .v.v... được nhà chế tạo tích hợp trong chúng và được kết nối với nhau bằng chương trình được viết bởi người dùng cho một nhiệm vụ điều khiển cụ thể nào đó. Bộ điều khiển PLC có nhiều loại khác nhau và được phân biệt với nhau qua các thành phần sau:

- Các ngõ vào và ra
- Dung lượng nhớ
- Bộ đếm (counter)
- Bộ định thời (timer)
- Bit nhớ
- Các chức năng đặc biệt
- Tốc độ xử lý
- Loại xử lý chương trình.
- Khả năng truyền thông.

Các bộ điều khiển lớn thì các thành phần trên được lắp thành các modul riêng. Đối với các bộ điều khiển nhỏ, chúng được tích hợp trong bộ điều khiển. Các bộ điều khiển nhỏ này có số lượng ngõ vào/ra cho trước cố định.

Bộ điều khiển được cung cấp tín hiệu bởi các 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 ra ngõ ra để đến đối tượng điều khiển hay khâu điều khiển ở dạng tín hiệu.

Cấu trúc của một PLC có thể được mô tả như hình vẽ sau:



Hình 2.5: Cấu trúc chung của bộ điều khiển lập trình PLC

* Bộ nhớ chương trình

Bộ nhớ chương trình trong PLC là một bộ nhớ điện tử đặc biệt có thể đọc được. Nếu sử dụng bộ nhớ đọc-ghi được (RAM), thì nội dung của nó luôn luôn được thay đổi ví dụ 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 nếu như có sử dụng Pin dự phòng.

Nếu chương trình điều khiển làm việc ổn định, hợp lý, nó có thể được nạp vào một bộ nhớ cố định, ví dụ như EPROM, EEPROM. Nội dung chương trình ở EPROM có thể bị xóa bằng tia cực tím.

* Hệ điều hành

Sau khi bật nguồn cung cấp cho bộ điều khiển, hệ điều hành của nó sẽ đặt các counter, timer, dữ liệu và bit nhớ với thuộc tính non-retentive (không được nhớ bởi Pin dự phòng) cũng như ACCU về 0.

Để xử lý chương trình, hệ điều hành đọc từng dòng chương trình từ đầu đến cuối. Tương ứng hệ điều hành thực hiện chương trình theo các câu lệnh.

* Bit nhớ (Bit memory)

Các bit memory là các phần tử nhớ, mà hệ điều hành ghi nhớ trạng thái tín hiệu.

* Bộ đếm (Process Image)

Bộ đếm là một vùng nhớ, mà hệ điều hành ghi nhớ các trạng thái tín hiệu ở các ngõ vào ra nhị phân.

*** Accumulator**

Accumulator là một bộ nhớ trung gian mà qua nó timer hay counter được nạp vào hay thực hiện các phép toán số học.

*** Counter, Timer**

Timer và counter cũng là các vùng nhớ, hệ điều hành ghi nhớ các giá trị đếm trong nó.

*** Hệ thống Bus**

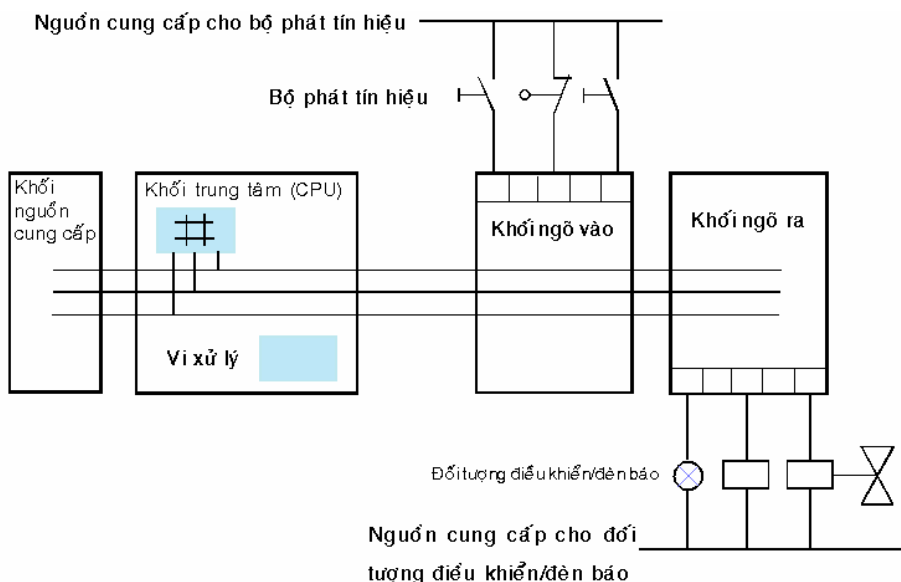
Bộ nhớ chương trình, hệ điều hành và các modul ngoại vi (các ngõ vào và ngõ ra) được kết nối với PLC thông qua Bus nối. Một Bus bao gồm các dây dẫn mà các dữ liệu được trao đổi. Hệ điều hành tổ chức việc truyền dữ liệu trên các dây dẫn này.

2.4 Các khối của PLC

Các khối khác nhau của một PLC được cho như hình 2.6.

2.4.1 Khối nguồn cung cấp

Khối nguồn có nhiệm vụ biến đổi điện áp lưới (110V hay 220V) thành điện áp thấp hơn cung cấp cho các khối của thiết bị tự động. Điện áp này là 24VDC. Các điện áp cho cảm biến, thiết bị điều chỉnh và các đèn báo nằm trong khoảng (24...220V) có thể được cung cấp thêm từ các nguồn phụ ví dụ như biến áp.



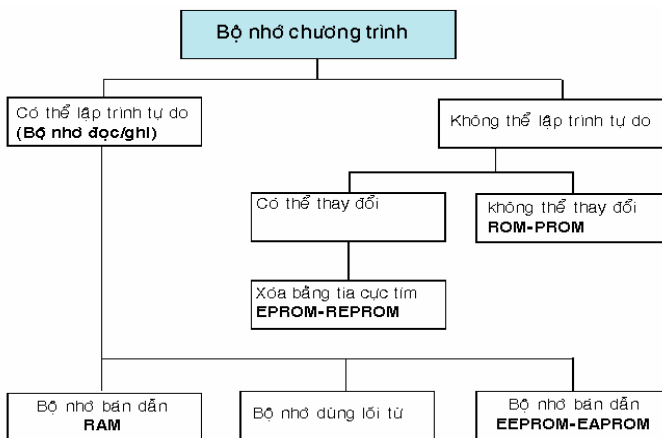
Hình 2.6: Các khối trong một PLC

2.4.2 Bộ nhớ chương trình

Các phần tử nhớ là các linh kiện mà thông tin có thể được lưu trữ (được nhớ) trong nó ở dạng tín hiệu nhị phân. Trong PLC các bộ nhớ bán dẫn được sử dụng làm bộ nhớ chương trình. Một bộ nhớ bao gồm 512, 1024, 2048 . . . phần tử nhớ, các phần tử nhớ này sắp đặt theo các địa chỉ từ 0 tới 511, 1023 hoặc 2047 . . . Thông thường số lượng của các phần tử nhớ trong một bộ nhớ cho biết dung lượng của nó là bao nhiêu kilobyte (1kB = 1024 byte). Trong mỗi ô nhớ có thể mô tả một câu lệnh điều khiển nhờ thiết bị lập trình. Mỗi phần tử nhị phân của một ô nhớ có thể có trạng thái tín hiệu "0" hoặc "1". Sơ đồ của một bộ nhớ chương trình được cho như hình 2.7.

* Bộ nhớ đọc-ghi RAM (random-access memory)

Bộ nhớ ghi-đọc có 1 số lượng các ô nhớ xác định. Mỗi ô nhớ có 1 dung lượng nhớ cố định và nó chỉ tiếp nhận 1 lượng thông tin nhất định. Các ô nhớ được ký hiệu bằng các địa chỉ riêng của nó. Bộ nhớ này chứa các chương trình còn sửa đổi hoặc các dữ liệu, kết quả tạm thời quá trình tính toán, lập trình. Đặc điểm của loại này là dữ liệu sẽ mất đi khi hệ thống mất điện. RAM được hình dung như một tủ chứa có nhiều ngăn kéo. Mỗi ngăn kéo được đánh số một địa chỉ và người ta có thể cất vào hoặc lấy các dữ liệu ra.



Hình 2.7: Sơ đồ một bộ nhớ chương trình

* Bộ nhớ cố định ROM (read-only memory)

Bộ nhớ cố định (ROM) chứa các thông tin không có khả năng xóa được và không thể thay đổi được. Các thông tin này do các nhà sản xuất viết ra và không thể thay đổi được. Chương trình trong bộ nhớ ROM có nhiệm vụ sau:

- Điều khiển và kiểm tra các chức năng hoạt động của CPU. Được gọi là hệ điều hành.
- Dịch ngôn ngữ lập trình thành ngôn ngữ máy.

Một ROM có thể so sánh với một quyển sách. Trong đó nó chứa các thông tin cố định, không thể thay đổi được và ta chỉ đọc các thông tin đó mà thôi. Đặc điểm của loại này là dữ liệu vẫn tồn tại khi mất điện.

*** EPROM (erasable read-only memory)**

EPROM là một bộ nhớ cố định có thể lập trình và xóa được. Nội dung của EPROM có thể xóa bằng tia cực tím và có thể lập trình lại.

*** EEPROM (electrically erasable read-only memory)**

EEPROM là bộ nhớ cố định có thể lập trình và xóa bằng điện. Mỗi ô nhớ trong EEPROM cho phép lập trình và xóa bằng điện.

2.4.3 Khối trung tâm (CPU)

Khối CPU là loại khối có chứa bộ vi xử lý, hệ điều hành, bộ nhớ, các bộ thời gian, bộ đếm, cổng truyền thông ... và có thể còn có một vài cổng vào ra số. Các cổng vào ra số có trên CPU được gọi là cổng vào/ra onboard.

2.4.4 Khối vào

Các ngõ vào của khối này sẽ được kết nối với các bộ chuyển đổi tín hiệu và biến đổi các tín hiệu này thành tín hiệu phù hợp với tín hiệu xử lý của CPU. Dựa vào loại tín hiệu vào sẽ có các khối ngõ vào tương ứng. Gồm có hai loại khối vào cơ bản sau:

- **Khối vào số (DI: Digital Input):**

Các ngõ vào của khối này được kết nối với các bộ chuyển đổi tạo ra tín hiệu nhị phân như nút nhấn, công tắc, cảm biến tạo tín hiệu nhị phân .v.v... Do tín hiệu tại ngõ vào có thể có mức logic tương ứng với các điện áp khác nhau, do đó khi sử dụng cần phải chú ý đến điện áp cần thiết cung cấp cho khối vào phải phù hợp với điện áp tương ứng mà bộ chuyển đổi tín hiệu nhị phân tạo ra.

Ví dụ: Các nút nhấn, công tắc được nối với nguồn 24VDC thì yêu cầu phải sử dụng khối vào có nguồn cung cấp cho nó là 24VDC.

- **Khối vào tương tự (AI: Analog Input):**

Khối này có nhiệm vụ biến đổi tín hiệu tương tự (hay còn gọi là tín hiệu analog) thành tín hiệu số. Các ngõ vào của khối này được kết nối với các bộ chuyển đổi tạo ra tín hiệu analog như cảm biến nhiệt độ (Thermocouple), cảm biến lưu lượng, ngõ ra analog của biến tần .v.v... Khi sử dụng các khối vào analog cần phải chú ý đến loại tín hiệu analog được tạo ra từ các bộ chuyển đổi (cảm biến)

Ví dụ: Các cảm biến tạo ra tín hiệu analog là dòng điện (4..20 mA) thì phải sử dụng ngõ vào analog là loại nhận tín hiệu dòng điện (4..20 mA). Nếu cảm biến tạo ra tín hiệu analog là điện áp (0..5V) thì phải sử dụng ngõ vào analog nhận tín hiệu là điện áp (0..5V).

2.4.5 Khối ra

Khối này có nhiệm vụ khuếch đại các tín hiệu sau xử lý của CPU (được gửi đến vùng đệm ra) cung cấp cho đối tượng điều khiển là cuộn dây, đèn báo, van từ .v.v.. Tùy thuộc vào đối tượng điều khiển nhận tín hiệu dạng nào mà sẽ có các khối ra tương ứng. Gồm có hai loại khối ra tiêu biểu:

- **Khối ra số (DO: Digital Output):**

Các ngõ ra của khối này được kết nối với các đối tượng điều khiển nhận tín hiệu nhị phân như đèn báo, cuộn dây relay .v.v... Vì đối tượng điều khiển nhận tín hiệu nhị phân sử dụng nhiều cấp điện áp khác nhau nên khi sử dụng các khối ra số cần phải chú ý đến điện áp cung cấp cho nó có phù hợp với điện áp cung cấp cho đối tượng điều khiển hay không. Theo loại điện áp sử dụng, ngõ ra số được phân thành hai loại:

- *Điện áp một chiều (DC: Direct Current):* Gồm có hai loại ngõ ra là Transistor và relay. Thông thường trong công nghiệp điện áp một chiều được sử dụng là 24V.
- *Điện áp xoay chiều (AC: Alternative Current):* Gồm có hai loại ngõ ra là relay và TRIAC.

- **Khối ra tương tự (AO: Analog Output):**

Khối này có nhiệm vụ biến đổi tín hiệu số được gửi từ CPU đến đối tượng điều khiển thành tín hiệu tương tự. Các ngõ ra của khối này được kết nối với các đối tượng điều khiển nhận tín hiệu tương tự như ngõ vào analog của biến tần, van tỷ lệ, .v.v... Khi sử dụng các ngõ ra tương tự cần chú ý đến loại tín hiệu tương tự cung cấp cho đối tượng điều khiển có phù hợp với tín hiệu tương tự mà đối tượng điều khiển cần nhận hay không.

Ví dụ: Ngõ vào analog của biến tần nhận tín hiệu là điện áp (0..10V) thì nhất thiết phải sử dụng ngõ ra tương tự tạo ra tín hiệu analog là điện áp (0..10V).

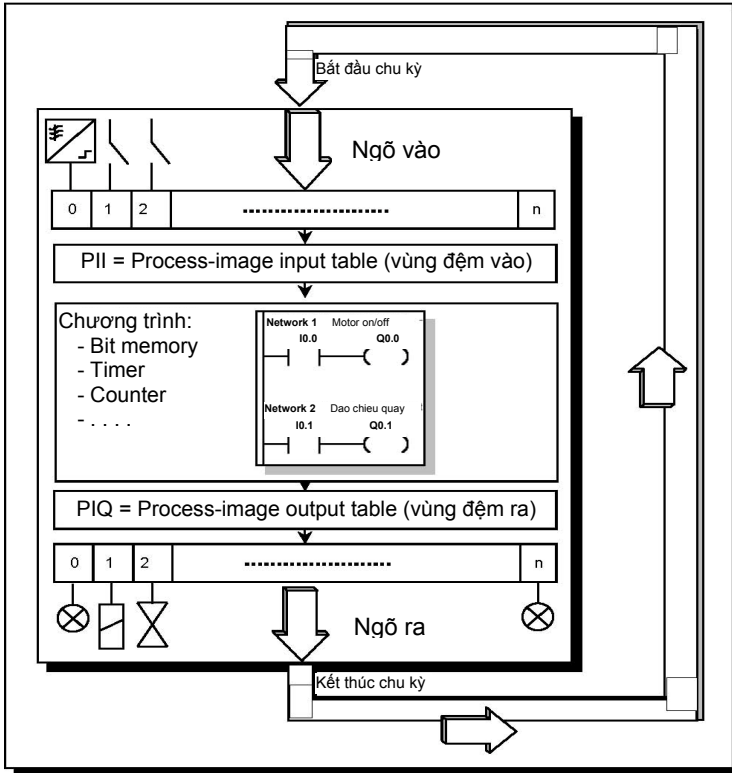
2.4.6 Các khối đặc biệt

Ngoài ra còn có một số khối khác đảm nhận các chức năng đặc biệt như xử lý truyền thông, thực hiện các chức năng đặc biệt như: điều khiển vị trí, điều khiển vòng kín, đếm tốc độ cao .v.v...

Tùy thuộc vào từng loại PLC mà các khối trên có thể ở các dạng module riêng hoặc được tích hợp chung trong khối xử lý trung tâm (CPU).

2.5 Phương thức thực hiện chương trình trong PLC

Hình vẽ minh họa việc xử lý chương trình trong CPU được cho như hình 2.8



Hình 2.8: Chu kỳ quét trong PLC

PLC thực hiện chương trình theo chu trình lặp. Mỗi vòng lặp được gọi là vòng quét (scan). Mỗi vòng quét được bắt đầu bằng giai đoạn chuyển dữ liệu từ các cổng vào số tới vùng bộ đệm ảo ngõ vào (I), tiếp theo là giai đoạn thực hiện chương trình. Trong từng dòng quét, chương trình được thực hiện từ lệnh đầu tiên đến lệnh kết thúc. Sau giai đoạn thực hiện chương trình là giai đoạn chuyển các nội dung của bộ đệm ảo ngõ ra (Q) tới các cổng ra số. Vòng quét được kết thúc bằng giai đoạn truyền thông nội bộ và kiểm tra lỗi.

Thời gian cần thiết để PLC thực hiện được một vòng quét gọi là thời gian vòng quét (Scan time). Thời gian vòng quét không cố định, tức là không phải vòng quét nào cũng được thực hiện trong một khoảng thời gian như nhau. Có vòng quét thực hiện lâu, có vòng quét thực hiện nhanh tùy thuộc vào số lệnh trong chương trình được thực hiện, vào khối lượng dữ liệu truyền thông ... trong vòng quét đó.

Như vậy giữa việc đọc dữ liệu từ đối tượng để xử lý, tính toán và việc gửi tín hiệu điều khiển tới đối tượng có một khoảng thời gian trễ đúng bằng thời gian vòng quét. Nói cách khác, thời gian vòng quét quyết định tính thời gian thực của chương trình điều khiển trong PLC. Thời gian quét càng ngắn, tính thời gian thực của chương trình càng cao.

Tại thời điểm thực hiện lệnh vào/ra, thông thường lệnh không làm việc trực tiếp với cổng vào/ra mà chỉ thông qua bộ đệm ảo của cổng trong vùng

nhớ tham số. Việc truyền thông giữa bộ đệm ảo với ngoại vi do hệ điều hành CPU quản lý. Ở một số module CPU, khi gặp lệnh vào/ra ngay lập tức, hệ thống sẽ cho dừng mọi công việc khác, ngay cả chương trình xử lý ngắt, để thực hiện lệnh trực tiếp với cổng vào/ra.

3 Cảm biến và cơ cấu chấp hành trong điều khiển logic.



Chương này nhằm giúp cho bạn đọc tìm hiểu sơ lược về một số các thiết bị ngoại vi sẽ được kết nối với các ngõ vào ra số của PLC và một số ký hiệu về các thiết bị ngoại vi.

3.1 Cảm biến

3.1.1 Giới thiệu

Cảm biến (sensor) cho phép PLC phát hiện trạng thái của một quá trình. Các cảm biến logic chỉ có thể phát hiện trạng thái đúng hoặc sai. Các hiện tượng vật lý tiêu biểu cần được phát hiện là:

- Tiếp cận cảm: cho biết một đối tượng là kim loại có đến gần vị trí cần nhận biết chưa?
- Tiếp cận dung: cho biết một đối tượng là không kim loại có đến gần vị trí cần nhận biết chưa?
- Sự xuất hiện ánh sáng: Cho biết một đối tượng có làm ngắt chùm tia sáng hay ánh sáng phản xạ?
- Tiếp xúc cơ học: Đối tượng có chạm vào công tắc?

Giá thành của cảm biến ngày càng giảm thấp và trở nên thông dụng. Chúng có nhiều hình dáng khác nhau được sản xuất bởi nhiều công ty khác nhau như Siemens, Omron, Pepperl+Fuch,... Trong các ứng dụng, các cảm biến được kết nối với PLC của nhiều hãng khác nhau, nhưng mỗi cảm biến sẽ có các yêu cầu giao tiếp riêng. Phần này sẽ trình bày cách thức nối dây cho các cảm biến và một số tính chất cơ bản của nó.

3.1.2 Nối dây cho cảm biến

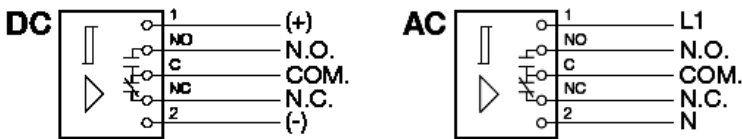
Khi một cảm biến phát hiện một sự thay đổi trạng thái logic thì nó phải truyền trạng thái thay đổi này đến PLC. Tiêu biểu là việc đóng hoặc ngắt dòng điện hay điện áp. Trong một vài trường hợp, ngõ ra của cảm biến sử dụng để đóng mạch trực tiếp cho tải mà không thông qua PLC. Các ngõ ra tiêu biểu của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

- *Sinking/Sourcing*: Đóng hoặc ngắt dòng điện
- *Switches*: Đóng hoặc ngắt điện áp

- **Solid State Relays:** Chuyển mạch AC
- **TTL (Transistor Transistor Logic):** Sử dụng điện áp 0V và 5V để chỉ thị mức logic.

3.1.2.1 Switch

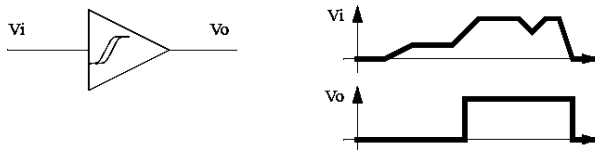
Một ví dụ đơn giản nhất của các ngõ ra cảm biến switch và relay được cho như hình 3.1.



Hình 3.1: Cảm biến có ngõ ra là relay sử dụng nguồn DC và AC .

3.1.2.2 Ngõ ra TTL

Ngõ ra TTL có hai mức điện áp: 0V tương ứng là mức thấp, 5V tương ứng mức cao. Điện áp thực tế có thể lớn hơn 0V hoặc nhỏ hơn 5V một chút vẫn có thể phát hiện đúng. Phương pháp này rất dễ bị nhiễu trong môi trường nhà máy cho nên nó chỉ được sử dụng khi cần thiết. Các ngõ ra TTL thường dùng trong các thiết bị điện tử và máy tính. Khi kết nối với các thiết bị khác thì một mạch Schmitt trigger thường được sử dụng để cải thiện tín hiệu (hình 3.2).



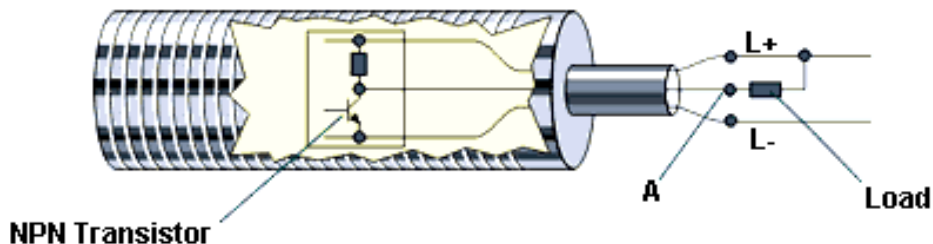
Hình 3.2: Mạch Schmitt trigger

Mạch Schmitt trigger sẽ nhận điện áp ngõ vào giữa 0-5V và chuyển đổi nó thành 0V hoặc 5V. Nếu điện áp nằm trong khoảng 1.5-3.5V thì không chấp nhận. Nếu một cảm biến có ngõ ra TTL thì PLC phải sử dụng các ngõ vào là TTL để đọc các giá trị này. Nếu các cảm biến TTL được sử dụng cho các ứng dụng khác thì nên chú ý dòng ngõ ra cực đại của cảm biến (thường khoảng 20mA).

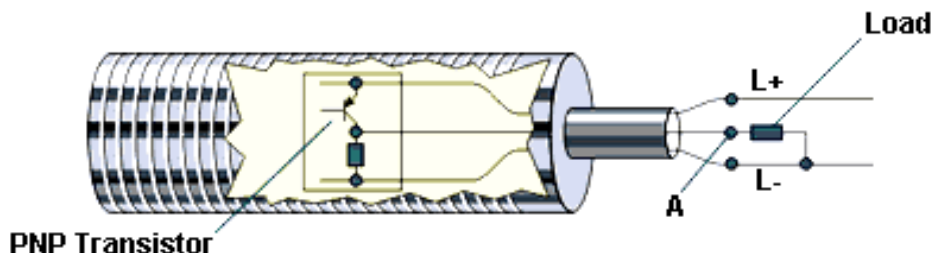
3.1.2.3 Ngõ ra Sinking/Sourcing

Các cảm biến có ngõ ra Sinking (rút dòng) cho phép dòng điện chạy vào cảm biến. Còn các cảm biến có ngõ ra sourcing (nguồn dòng) cho phép dòng điện chảy từ cảm biến ra đối tượng được kết nối. Ở hai ngõ ra này cần chú ý là dòng điện chứ không phải điện áp. Bằng cách sử dụng dòng điện thì nhiễu được loại trừ bớt.

Khi giải thích về vấn đề sinking hay sourcing thì ta nên quy các ngõ ra của cảm biến tác động như công tắc. Trong thực tế, các ngõ ra của cảm biến thường là một transistor chuyển mạch. Transistor PNP được sử dụng cho ngõ ra sourcing, và transistor NPN được sử dụng cho ngõ vào sinking. Khi giải thích các cảm biến này thì khái niệm “nguồn dòng” thường được dùng cho PNP, và “rút dòng” với NPN. Ví dụ cảm biến ngõ ra sinking được cho ở hình 3.3.



Hình 3.3: Cảm biến NPN (cảm biến “rút dòng”).



Hình 3.4: Cảm biến PNP (cảm biến “sourcing”)

Để cảm biến hoạt động cần phải có nguồn cung cấp (chân L+ và L-). Khi cảm biến phát hiện đối tượng thì có điện áp tại cực B của transistor NPN, transistor chuyển sang trạng thái dẫn và cho phép dòng chảy vào cảm biến xuống mass (chân L-).

Khi không phát hiện đối tượng thì điện áp tại cực B của transistor ở mức thấp (0V), transistor không dẫn. Điều này có nghĩa ngõ ra NPN sẽ không có dòng vào/ra.

Các cảm biến “sourcing” thì ngược với các cảm biến “sinking”. Nó sử dụng transistor PNP (hình 3.4). Khi cảm biến không được kích hoạt thì cực B của transistor ở giá trị L+, và transistor ở trạng thái ngưng dẫn. Khi cảm biến được kích hoạt thì cực B transistor sẽ được đặt ở 0V, và transistor cho phép dòng điện chảy từ cảm biến ra ngoài thiết bị được kết nối.

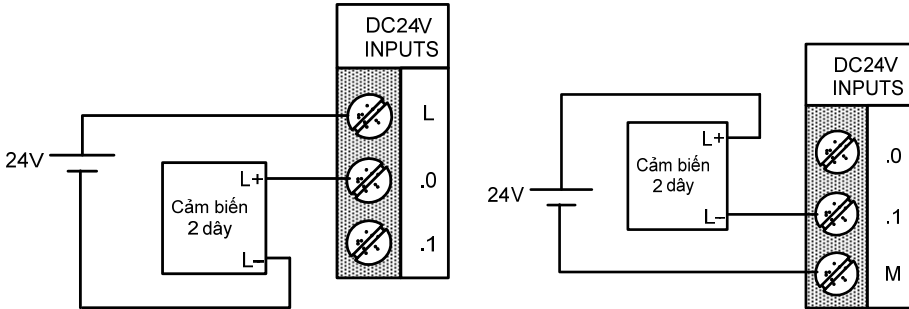
Hầu hết các cảm biến NPN/PNP có khả năng dòng đến vài ampere, và chúng có thể được sử dụng để nối trực tiếp với tải (luôn luôn kiểm tra sổ tay để biết chính xác dòng điện và điện áp định mức).

Chú ý: Cần phải nhớ kiểm tra dòng điện và điện áp định mức đối với các cảm biến. Khi nối dây các cảm biến cần chú ý đến các chân nguồn. Thường các

chân nguồn có ký hiệu là L+ và COM(chân chung), nhưng đôi khi không có chân COM mà có chân L-. Trong trường hợp này L- là chân chung.

Khi kết nối các cảm biến “sourcing” với các ngõ PLC, thì cần chú ý phải sử dụng các modul ngõ vào loại “sinking”. Thông thường các ngõ vào PLC thường là loại “sinking”.

Trong ứng dụng với PLC, để giảm lượng dây nối, thì các cảm biến hai dây thường được sử dụng. Ví dụ về sơ đồ nối dây các cảm biến sử dụng nguồn 24VDC với PLC được chỉ như hình 3.5. Cảm biến hai dây có thể được sử dụng cho cả hai loại ngõ vào sourcing hoặc ngõ vào sinking của PLC.



a. Ngõ vào PLC loại sourcing

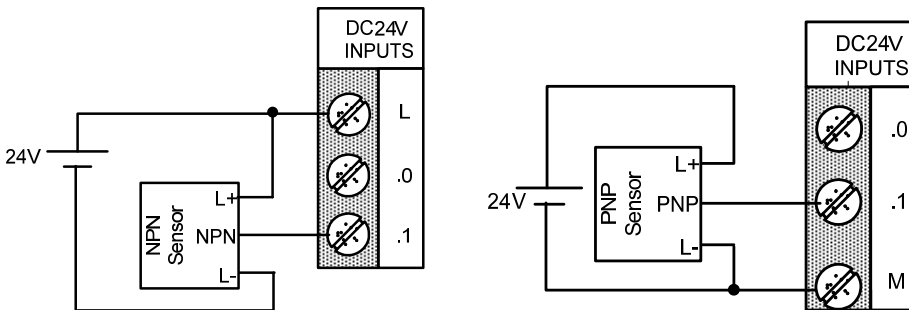
b. Ngõ vào PLC loại sinking

Hình 3.5: Kết nối cảm biến 2 dây với ngõ vào PLC.

Hầu hết các cảm biến hiện đại có cả hai ngõ ra PNP và NPN. Thông thường cảm biến loại PNP thường được sử dụng cho các ngõ vào PLC.

Trong các bản vẽ thì các chân của các cảm biến NPN và PNP có ký hiệu về màu sắc như sau: dây màu nâu là L+, dây màu xanh dương là L- và ngõ ra thì màu trắng đối với sinking và màu đen đối với sourcing.

Cần lưu ý là khi tiếp điểm trong cảm biến “sinking” đóng thì ngõ ra được nối với COM hoặc L-, tiếp điểm trong sourcing đóng thì ngõ ra nối với L+.



a. Ngõ vào PLC loại sourcing

b. Ngõ vào PLC loại sinking

Hình 3.6: Kết nối cảm biến NPN và PNP dây với ngõ vào PLC.

3.1.2.4 Ngõ ra Solid state relay

Các ngõ ra Solid state relays đóng mạch dòng điện AC. Các cảm biến này được sử dụng với tải lớn.

3.1.3 Phát hiện đối tượng

Có hai cách cơ bản để phát hiện đối tượng: tiếp xúc và tiếp cận (proximity).

Tiếp xúc có nghĩa là tiếp điểm cơ khí cần một lực tác động giữa cảm biến và đối tượng.

Tiếp cận để chỉ báo rằng một đối tượng đang ở gần nhưng không yêu cầu tiếp xúc.

Các phần sau đây sẽ minh họa các kiểu khác nhau của các cảm biến để phát hiện sự hiện diện của các đối tượng. Phần này không đi sâu vào các cảm biến mà chỉ mô tả các nguyên lý trong lĩnh vực ứng dụng.

3.1.3.1 Chuyển mạch tiếp xúc

Chuyển mạch tiếp xúc (contact switch) thường có hai dạng là thường hở (normally open) và thường đóng (normally closed). Vỏ của chúng được gia cố để có thể chịu được lực cơ tác động nhiều lần.

3.1.3.2 Reed Switches

Reed switches thì rất giống relay, ngoại trừ một nam châm vĩnh cửu được sử dụng thay thế cuộn dây. Khi nam châm ở xa thì tiếp điểm mở, nhưng khi nam châm đến gần thì tiếp điểm đóng lại (hình 3.7). Các cảm biến này rẻ tiền và chúng thường được sử dụng cho các màn chắn và cửa an toàn.



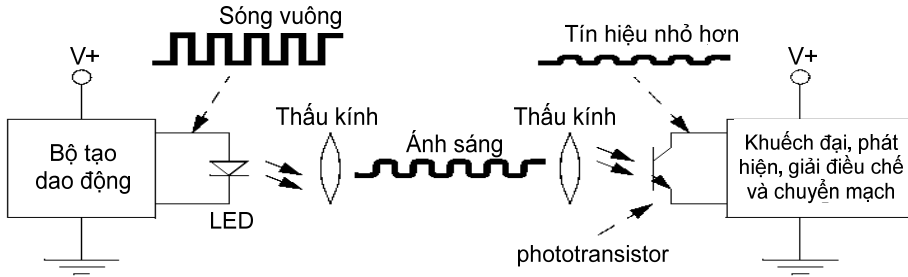
Hình 3.7: Reed switch

3.1.3.3 Cảm biến quang (Optical Sensor)

Cảm biến ánh sáng được sử dụng gần một thế kỷ qua. Nguyên thủy là tế bào quang được sử dụng cho các ứng dụng như đọc các track âm thanh trên các hình ảnh chuyển động. Nhưng các cảm biến quang hiện đại thì phức tạp hơn nhiều.

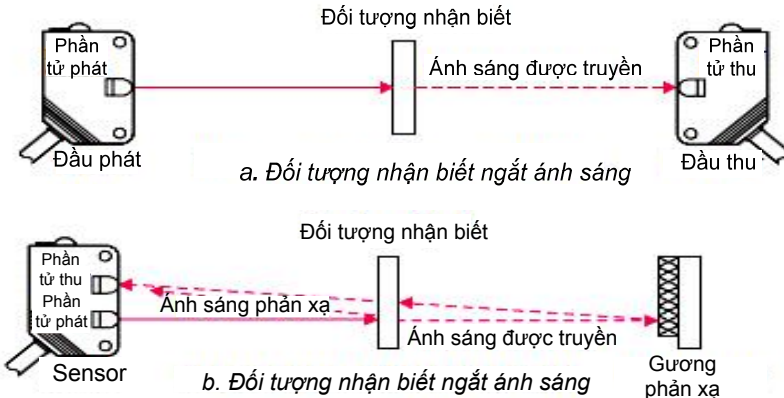
Các cảm biến quang yêu cầu có cả hai bộ phận là nguồn sáng (phát) và đầu thu (detector). Các đầu phát (emitter) sẽ phát ra các tia sáng trong vùng phổ nhìn thấy và không nhìn thấy được sử dụng LED và diode laser. Đầu thu có cấu tạo là các diode quang (photodiode) hoặc transistor quang (phototransistor). Đầu phát và đầu thu được đặt vào vị trí để đối tượng khi

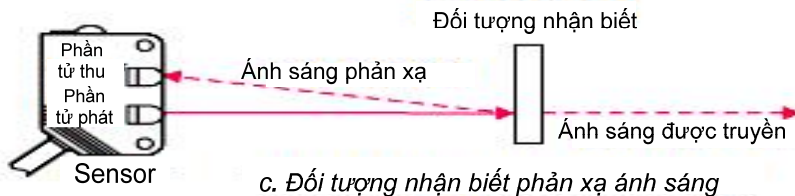
xuất hiện sẽ cắt ngang hoặc phản xạ lại tia sáng. Cảm biến quang đơn giản cho ở hình 3.8.



Hình 3.8: Cảm biến quang cơ bản

Trong hình, chùm sáng được tạo ra nằm ở bên trái, được hội tụ qua một thấu kính. Đối diện là đầu thu, chùm tia được hội tụ bằng một thấu kính thứ hai. Nếu chùm tia bị ngắt, thì đầu thu sẽ chỉ báo một đối tượng xuất hiện. Ánh sáng được tạo ra dưới dạng xung để cảm biến có thể lọc được ánh sáng bình thường trong phòng. Ánh sáng từ đầu phát được tắt và mở tại một tần số đặt. Khi đầu thu nhận ánh sáng, nó kiểm tra để đảm bảo chắc chắn rằng nó có cùng tần số. Nếu ánh sáng đang nhận được tại tần số đúng thì chùm tia không bị ngắt. Tần số dao động nằm trong phạm vi KHz. Ngoài ra với phương pháp tần số thì các cảm biến có thể được sử dụng với công suất thấp hơn và khoảng cách dài hơn. Đầu phát có thể bắt đầu từ một điểm trực tiếp tại đầu thu, đây còn gọi là chế độ tự phản xạ. Khi tia sáng bị ngắt, thì đối tượng được phát hiện. Cảm biến này cần hai bộ phận riêng (hình 3.9a). Sự xếp đặt này làm việc tốt với các đối tượng chắn sáng và phản xạ với đầu phát và đầu thu được tách riêng với khoảng cách lên đến cả trăm mét.





Hình 13.9: Các loại cảm biến quang khác nhau

Đầu thu và đầu phát tách riêng làm tăng vấn đề về bảo trì và yêu cầu về sự thẳng hàng. Một giải pháp khác là đầu phát và đầu thu được đặt chung trên một vỏ. Nhưng điều này yêu cầu ánh sáng tự phản xạ trở về (hình 3.9b,c). Các cảm biến này chỉ tốt cho các đối tượng lớn với khoảng cách một vài mét.

Trong hình, đầu phát phát một chùm tia sáng. Nếu ánh sáng bị dội trở về từ gương phản xạ thì hầu hết sẽ trở về đầu thu. Khi một đối tượng ngắt chùm tia giữa đầu phát và gương phản xạ thì chùm tia sẽ không tự phản xạ trở về đầu thu và cảm biến được tác động. Một vấn đề rủi ro cho các cảm biến này là các đối tượng tự phản xạ lại chùm tia sáng tốt. Để giải quyết thì sử dụng biện pháp phân cực ánh sáng tại đầu phát (bằng bộ lọc), và sau đó sử dụng một bộ lọc phân cực tại đầu thu.

3.1.3.4 Cảm biến điện dung (Capacitive Sensor)

Các cảm biến điện dung có thể phát hiện hầu hết các vật liệu với khoảng cách vài cm.

Công thức biểu diễn mối quan hệ điện dung:

$$C = \frac{\epsilon \cdot A}{d} \quad \text{với} \quad C: \text{Điện dung (Farads)}$$

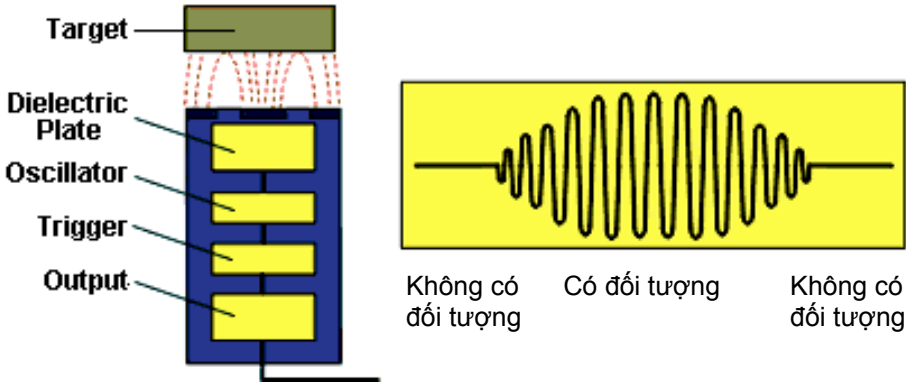
ϵ : Hằng số điện môi

A: Diện tích bản cực

D: Khoảng cách giữa các bản cực.

Trong cảm biến, diện tích các bản cực và khoảng cách giữa chúng là cố định. Nhưng hằng số điện môi của không gian xung quanh chúng sẽ thay đổi khi các vật liệu được mang đến gần cảm biến. Minh họa ở hình 3.10.

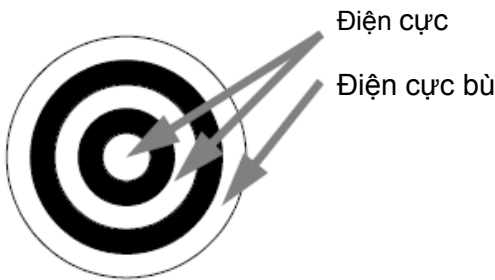
Bề mặt của cảm biến điện dung được hình thành bởi hai điện cực kim loại đồng tâm của một tụ điện. Khi một đối tượng đến gần bề mặt nhận biết nó đi vào vùng điện trường của các điện cực và thay đổi điện dung trong mạch dao động. Kết quả là bộ tạo dao động bắt đầu dao động. Mạch trigger đọc biên độ của bộ dao động và khi đạt đến mức xác định thì trạng thái ngõ ra sẽ thay đổi. Khi đối tượng rời khỏi cảm biến thì biên độ của bộ dao động giảm, cảm biến chuyển về trạng thái bình thường.



Hình 3.10: Cảm biến điện dung

Các cảm biến này làm việc tốt đối với chất cách điện (như chất dẻo) có hằng số điện môi cao (làm tăng điện dung). Hằng số điện môi càng lớn thì khoảng cách hoạt động càng cao. Ví dụ khi hiệu chỉnh đúng thì chất lỏng trong thùng chứa có thể được phát hiện được dễ dàng. Tuy nhiên, chúng cũng làm việc tốt đối với kim loại.

Các cảm biến thường được chế tạo với các vòng (không phải bản cực) theo hình 3.11. Trong hình, hai vòng kim loại nằm bên trong là các điện cực của tụ điện, nhưng vòng ngoài thứ ba được thêm vào để bù sự thay đổi. Nếu không có vòng bù này thì cảm biến sẽ rất nhạy cảm với bụi bặm, dầu và các chất khác dính trên cảm biến.



Hình 3.11: Bề mặt nhận biết của cảm biến điện dung

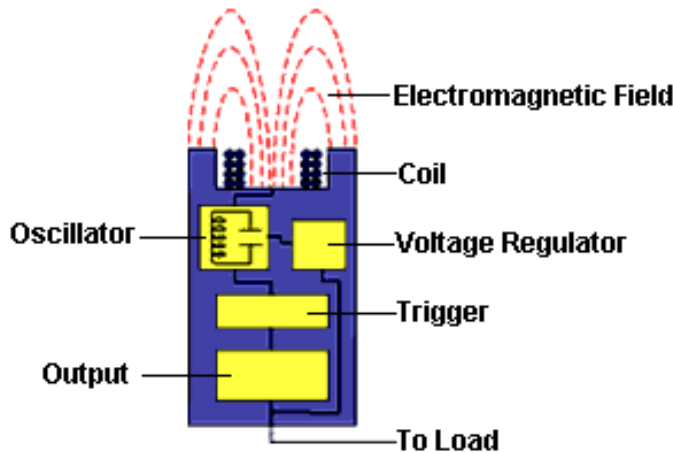
Phạm vi và độ chính xác của các cảm biến được xác định bởi kích thước của chúng. Các cảm biến lớn có thể có đường kính vài centimeter. Cái nhỏ có đường kính nhỏ hơn một centimeter và có phạm vi nhỏ hơn nhưng chính xác hơn.

3.1.3.5 Cảm biến điện cảm (Inductive Sensor)

Các cảm biến điện cảm sử dụng dòng điện cảm ứng để phát hiện đối tượng là kim loại. Cảm biến điện cảm sử dụng một cuộn dây để tạo một từ trường tần số cao được cho ở hình 3.12. Nếu có một đối tượng là kim loại đến gần làm thay đổi từ trường, thì sẽ có dòng chảy vào đối tượng. Dòng chảy này tạo ra một từ trường mới ngược với từ trường ban đầu. Kết quả là nó làm thay

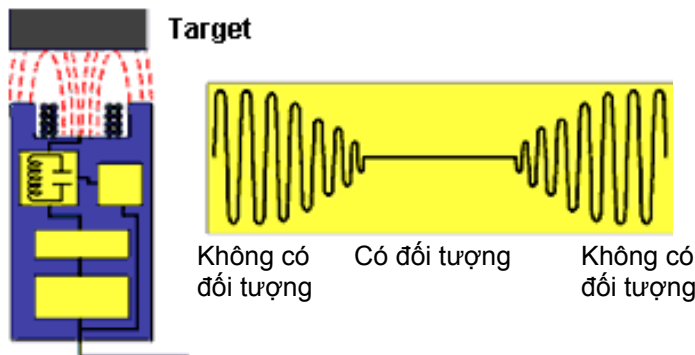
đổi độ tự cảm của cuộn dây trong cảm biến. Bằng cách đo độ tự cảm, cảm biến có thể xác định một đối tượng kim loại đến gần.

Các cảm biến này sẽ phát hiện bất kỳ kim loại nào, khi cần phát hiện các loại kim loại thì các cảm biến đa kim loại thường được sử dụng.



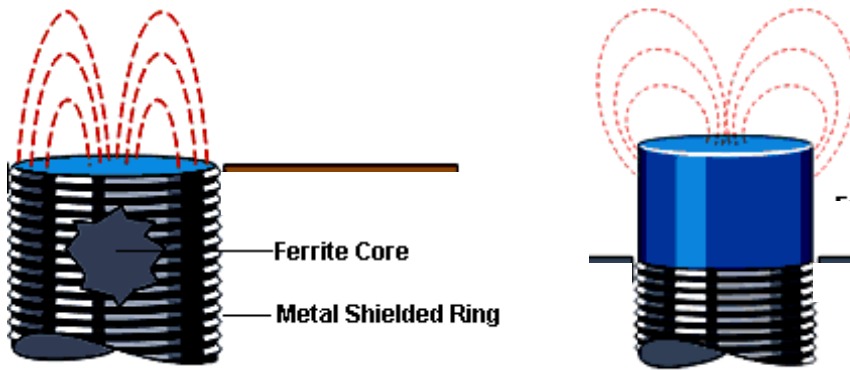
Hình 3.12: Cảm biến tiếp cận điện cảm

Khi đối tượng kim loại đi vào vùng điện từ trường, thì dòng điện xoáy truyền vào đối tượng. Điều này làm tăng tải trong cảm biến, làm giảm biên độ của điện từ trường. Mạch trigger giám sát biên độ dao động khi đạt đến mức định trước thì nó chuyển đổi trạng thái ngõ ra của cảm biến. Khi đối tượng di chuyển khỏi cảm biến, thì biên độ dao động tăng lên. Khi đến giá trị định trước thì mạch trigger chuyển đổi trạng thái ngõ ra trở về điều kiện bình thường.



Hình 3.13: Cảm biến tiếp cận điện cảm

Các cảm biến có thể phát hiện các đối tượng cách xa vài centimeter. Nhưng hướng của đối tượng có thể là bất kỳ như hình 3.14. Từ trường của các cảm biến không bọc bao phủ xung quanh đầu của cuộn dây lớn hơn. Bằng cách lắp thêm vỏ bọc kim loại thì từ trường sẽ nhỏ hơn, nhưng hướng của đối tượng nhận biết được cải thiện hơn.



Hình 3.14: Cảm biến bọc và không bọc vỏ kim loại

3.1.3.6 Cảm biến siêu âm (Ultrasonic sensor)

Cảm biến siêu âm phát ra âm thanh trên ngưỡng nghe bình thường 16kHz. Thời gian được yêu cầu để âm thanh di chuyển đến mục tiêu và phản hồi trở về tỷ lệ với khoảng cách mục tiêu. Có hai loại cảm biến là:

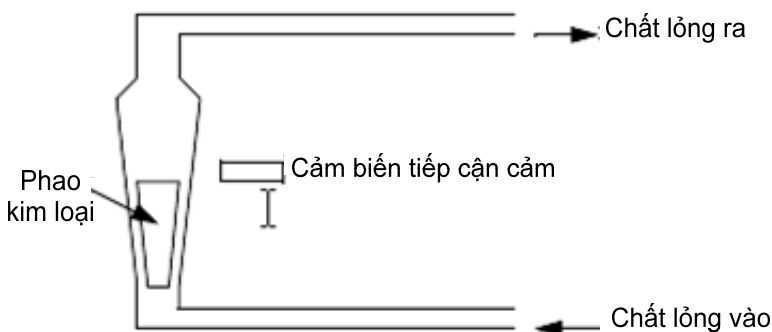
- Tĩnh điện (electrostatic): Sử dụng hiệu ứng điện dung. Phạm vi lớn và băng thông rộng hơn nhưng độ nhạy cao hơn với đối tượng ẩm ướt.
- Áp điện (piezoelectric): Dựa vào phần tử áp điện thạch anh.

Các cảm biến này có thể rất hiệu quả cho các ứng dụng như đo mức chất lỏng trong thùng chứa.

3.1.3.7 Hiệu ứng Hall (Hall Effect)

Các công tắc hiệu ứng Hall cơ bản là các transistor có thể chuyển mạch bởi từ trường. Các ứng dụng của chúng thì rất giống với reed switch, nhưng vì chúng chỉ là chất bán dẫn nên chúng phù hợp với các chuyển động. Các máy móc tự động hóa thường sử dụng chúng để thực hiện khởi động và phát hiện vị trí dừng.

3.1.3.8 Lưu lượng (Fluid Flow)



Hình 3.15: xác định lưu lượng dòng chảy với cảm biến tiếp cận cảm

Chúng ta có thể thay thế các cảm biến phức tạp bằng các cảm biến đơn giản. Hình 3.15 cho thấy một phao kim loại trong một kênh hình nón. Tốc độ dòng chảy tăng áp lực đẩy phao lên trên. Dạng hình nón của phao đảm bảo vị trí của chất lỏng tỷ lệ với tốc độ dòng chảy. Một cảm biến tiếp cận điện cảm có thể được định vị để nó phát hiện khi phao đạt đến độ cao nào đó, và hệ thống đạt đến tốc độ dòng chảy đã định.

3.1.4 Tóm tắt

- Cảm biến Sourcing cho phép dòng điện chảy từ cực L+ của nguồn.
- Cảm biến Sinking cho phép dòng điện chảy từ cực L- của nguồn..
- Cảm biến quang có thể sử dụng chùm tia phản xạ, đầu phát và đầu thu và ánh sáng phản xạ để phát hiện đối tượng.
- Cảm biến điện dung có thể phát hiện kim loại và các vật liệu khác.
- Cảm biến điện cảm phát hiện được kim loại.
- Cảm biến hiệu ứng Hall và reed switch có thể phát hiện được nam châm.
- Cảm biến siêu âm sử dụng sóng âm để phát hiện các phần tử cách xa nhiều meter.

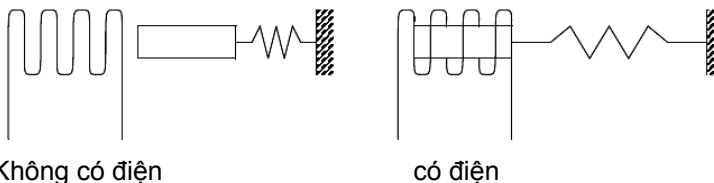
3.2 Cơ cấu chấp hành

3.2.1 Giới thiệu

Cơ cấu chấp hành được sử dụng để biến đổi năng lượng điện thành chuyển động cơ học.

3.2.2 Solenoid

Solenoid là cơ cấu chấp hành thông dụng nhất. Nguyên lý hoạt động cơ bản là sự di chuyển lõi sắt (piston) trong cuộn dây (hình 3.16). Bình thường piston được giữ bên ngoài cuộn dây. Khi cuộn dây được cấp điện, cuộn dây sinh ra từ trường hút piston và kéo nó vào trung tâm của cuộn dây. Ứng dụng quan trọng nhất của solenoid là điều khiển các van khí nén, thủy lực và khóa cửa xe.

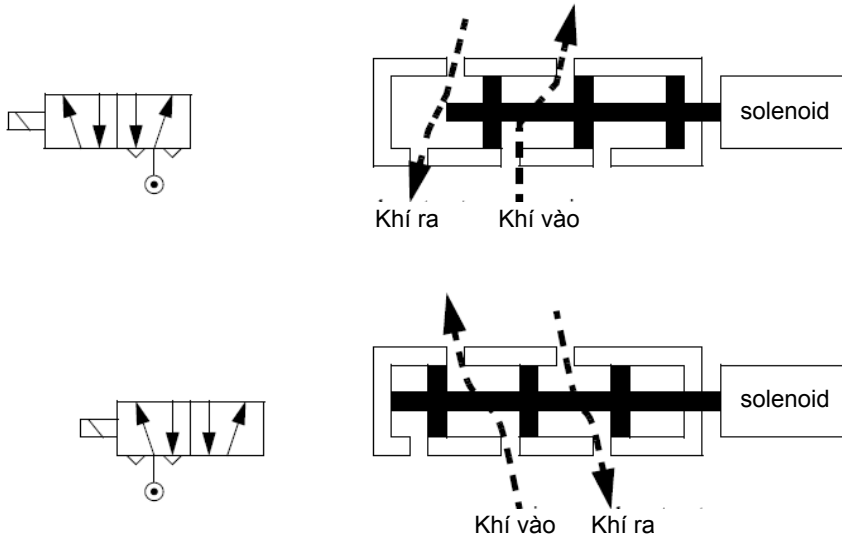


Hình 3.16: Solenoid

Cần chú ý là các cuộn cảm có thể tạo ra điện áp gai nhọn và có thể cần các bộ giảm sóc. Mặc dù vậy hầu hết trong các ứng dụng công nghiệp có điện áp thấp và dòng điện định mức, chúng có thể được kết nối trực tiếp với các ngõ ra của PLC. Hầu hết các solenoid công nghiệp sử dụng nguồn cung cấp 24Vdc và dòng định mức một vài trăm mA.

3.2.3 Van điều khiển (VALVE)

Dòng chất lỏng và khí có thể được điều khiển bằng các van điều khiển solenoid. Ví dụ van điều khiển solenoid được cho ở hình 3.17.



Hình 3.17: Một solenoid điều khiển van 5 cửa 2 vị trí

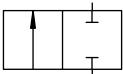
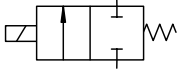
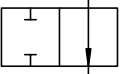
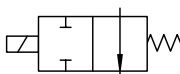
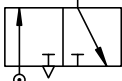
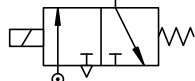
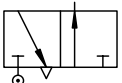
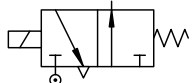
Các loại van được liệt kê dưới đây. Theo tiêu chuẩn, thuật ngữ ‘n-cửa’ (n-cửa) để chỉ định số lượng kết nối các ngõ vào và ra của van. Trong một vài trường hợp có cửa để xả khí ra. Việc thiết kế thường đóng/thường mở cho biết điều kiện van khi mất nguồn cấp.

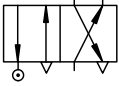
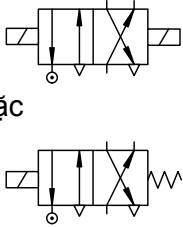
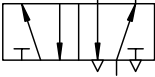
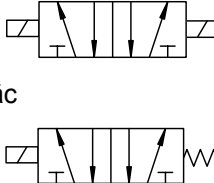
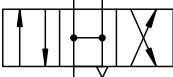
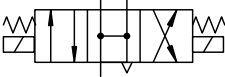
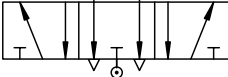
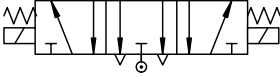
- *Van 2 cửa, 2 vị trí thường đóng (van 2/2):* Các van này có 1 cửa vào và một cửa ra. Khi mất nguồn cung cấp thì ở vị trí thường đóng. Khi có nguồn cung cấp, thì van mở cho phép dòng khí hay chất lỏng chảy qua. Các van này được sử dụng để cho phép dòng chảy.
- *Van 2 cửa, 2 vị trí thường mở (van 2/2):* Các van này có một cửa vào và một cửa ra. Khi mất nguồn thì mở cho phép dòng chảy. Khi có nguồn, van đóng. Các van này được sử dụng để ngắt dòng chảy.
- *Van 3 cửa, 2 vị trí thường đóng (van 3/2):* Các van này có cửa vào, cửa ra và cửa xả khí. Khi mất nguồn thì cửa ra được nối với cửa xả khí. Khi có nguồn thì cửa vào được nối với cửa ra. Các van này được sử dụng cho các cylinder tác động đơn.
- *Van 3 cửa, 2 vị trí thường mở (van 3/2):* Các van này có cửa vào, cửa ra và cửa xả khí. Khi mất nguồn thì cửa vào được nối với cửa ra. Khi có nguồn thì van nối cửa ra với cửa xả khí. Các van này được sử dụng cho các cylinder tác động đơn.
- *Van 3 cửa, 2 vị trí đa năng (van 3/2):* Các van này có 3 cửa. Một trong các cửa hoạt động như là cửa vào hoặc cửa ra, và được nối đến một trong hai cửa khác khi mất nguồn hoặc có nguồn. Các van này có thể

được sử dụng để làm chuyển hướng dòng chảy, hoặc chọn nguồn qua lại.

- *Van 4 cửa, 2 vị trí (van 4/2)*: Các van này có 4 cửa, 1 vào, 2 ra và 1 cửa xả khí. Khi có nguồn van nối các cửa vào với các cửa ra và ngược lại. Các van này được sử dụng với các cylinder tác động kép.
- *Van 5 cửa, 2 vị trí (van 5/2)*: Các van này có 5 cửa, 1 vào, 2 ra và 2 cửa xả khí.
- *Van 4 cửa, 3 vị trí (van 4/3)*: Các van này có 4 cửa, 1 vào, 2 ra và 1 xả. Ở trạng thái bình thường (không có nguồn năng lượng) thì các cửa vào/ra đều bị chặn. Van này được sử dụng để điều khiển vị trí các cylinder.
- *Van 5 cửa, 3 vị trí (van 5/3)*: Van này có 5 cửa, 1 vào, 2 ra và 2 cửa xả. Tương tự như van 4/3, van này được sử dụng để điều khiển vị trí các cylinder.

Ký hiệu của các van được cho ở hình 3.18. Khi sử dụng trong các bản vẽ thì vẽ ở trạng thái không được cấp nguồn năng lượng. Mũi tên chỉ đường dẫn dòng chảy đến các vị trí khác. Biểu tượng tam giác nhỏ để chỉ cửa xả khí.

Loại van	Ký hiệu	
	Điều khiển bằng khí nén	Điều khiển bằng solenoid
Van 2 cửa, 2 vị trí	 <p>Thường đóng</p>	 <p>Thường đóng</p>
	 <p>Thường mở</p>	 <p>Thường mở</p>
Van 3 cửa, 2 vị trí	 <p>Thường đóng</p>	 <p>Thường đóng</p>
	 <p>Thường mở</p>	 <p>Thường mở</p>

<p>Van 4 cửa, 2 vị trí</p>		<p>Hoặc</p> 
<p>Van 5 cửa, 2 vị trí</p>		<p>Hoặc</p> 
<p>Van 4 cửa, 3 vị trí</p>		
<p>Van 5 cửa, 3 vị trí</p>		

Hình 3.18 Ký hiệu các van điều khiển bằng khí và solenoid

Khi chọn lựa van, cần chú ý một số chi tiết sau:

- Kích thước ống: Cửa vào và ra theo tiêu chuẩn NPT (national pipe thread).

- Tốc độ dòng chảy: Tốc độ dòng chảy cực đại thường được cung cấp cho các van thủy lực.
- Áp suất hoạt động: Áp suất hoạt động cực đại phải được chỉ báo. Một vài van có yêu cầu áp suất tối thiểu để hoạt động.
- Nguồn điện: Các cuộn dây solenoid yêu cầu được cung cấp một điện áp và dòng điện cố định (AC hoặc DC).
- Thời gian đáp ứng: Đây là thời gian để van đóng/mở hoàn toàn. Thời gian tiêu biểu cho các van nằm trong phạm vi từ 5ms đến 150ms.
- Vỏ bọc: Vỏ bọc cho các van được xếp theo loại:

Loại 1 hoặc 2: Sử dụng trong nhà, yêu cầu bảo vệ chống nước.

Loại 3: Sử dụng ngoài trời, chống bụi bặm và mưa gió.

Loại 3R hoặc 3S hoặc 4: Chống nước và bụi.

Loại 4X: Chống nước, bụi và sự ăn mòn.

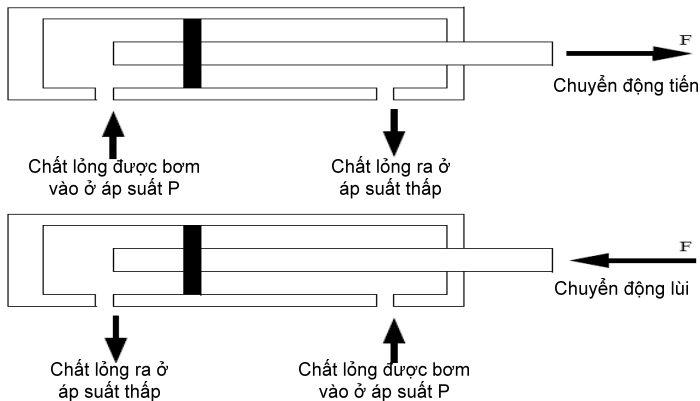
3.2.4 Xy lanh (CYLINDER)

Cylinder sử dụng áp lực khí hoặc chất lỏng để tạo lực/chuyển động tuyến tính (hình 3.19). Trong hình, dòng chất lỏng được bơm vào một phía của cylinder làm dịch chuyển piston về phía còn lại. Chất lỏng ở phía này được thoát tự do. Lực tác dụng lên cylinder tỷ lệ với diện tích bề mặt của piston.

Công thức tính lực:

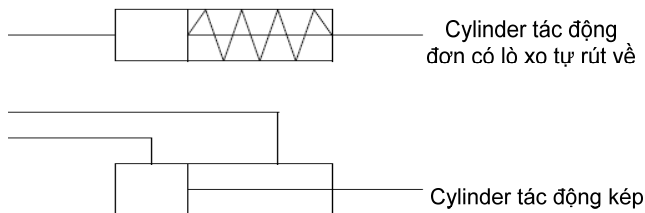
$$F = P \cdot A \qquad P = \frac{F}{A} \qquad \text{Với } P: \text{Áp suất thủy lực}$$

F: Lực đẩy piston
A: Diện tích piston



Hình 3.19 Mặt cắt của một cylinder thủy lực

Cylinder tác động đơn yêu cầu cung cấp lực khi duỗi ra và sử dụng lò xo để co về. Còn cylinder tác động kép thì cung cấp lực ở cả hai phía.



Hình 3.20 cylinder tác động đơn và cylinder tác động kép

Các cylinder từ thường được sử dụng trong điều khiển khí nén. Trên đầu của piston có một mảnh nam châm. Khi nó di chuyển đến vị trí giới hạn thì các công tắc reed switch sẽ phát hiện ra.

3.2.5 Động cơ

Động cơ là cơ cấu chấp hành thông thường, nhưng đối với ứng dụng cho điều khiển nhị phân thì đặc điểm của nó không quan trọng. Điều khiển logic tiêu biểu của các động cơ là đóng cắt điện cho nó. Các động cơ có dòng

điện nhỏ có thể đấu trực tiếp vào các ngõ ra của PLC, còn đối với các động cơ công suất lớn thì sử dụng relay hay contactor hoặc bộ khởi động động cơ. Các động cơ sẽ được khảo sát chi tiết hơn ở *chương các cảm biến và cơ cấu chấp hành analog (tập 2)*.

3.2.6 Các cơ cấu chấp hành khác

Ngoài các cơ cấu chấp hành kể trên còn có nhiều loại cơ cấu chấp hành khác nhau trong điều khiển logic. Một số cơ cấu chấp hành thường được sử dụng relay và contactor.

Ngoài ra có một số cơ cấu chấp hành khác:

- *Lò nhiệt*: Thường được điều khiển bằng relay, đóng và cắt điện để giữ nhiệt độ nằm trong một phạm vi nào đó.
- *Đèn báo*: Đèn báo được sử dụng cho hầu hết các máy móc để chỉ báo trạng thái máy và cung cấp thông tin cho người vận hành. Hầu hết các đèn báo có dòng điện thấp và được kết nối trực tiếp đến PLC.
- *Còi/chuông báo*: Còi hay chuông báo có thể được sử dụng cho các máy móc không được giám sát hoặc đang bị nguy hiểm. Chúng thường được nối trực tiếp với các ngõ ra của PLC.

4 Bộ điều khiển lập trình PLC Simatic S7-200

4.1 Cấu hình cứng

4.1.1 Khối xử lý trung tâm

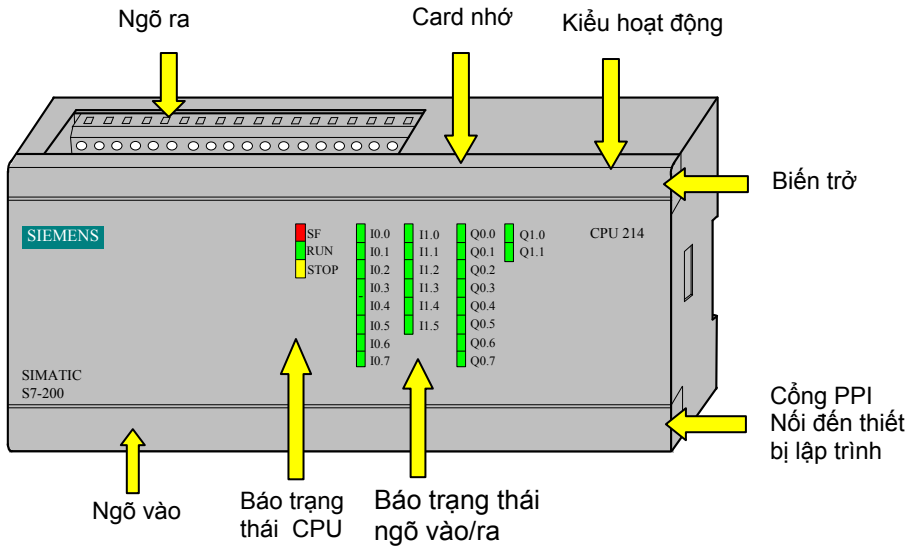
PLC S7-200 là thiết bị điều khiển lập trình loại nhỏ (micro PLC) của hãng Siemens (CHLB Đức) có cấu trúc theo kiểu *modul* và có các modul mở rộng. Thành phần cơ bản của S7 - 200 là khối xử lý trung tâm (CPU: Central Processing Unit) bao gồm hai chủng loại: CPU 21x và CPU 22x. Mỗi chủng loại có nhiều CPU. Loại CPU 21x ngày nay không còn sản xuất nữa, tuy nhiên hiện vẫn còn sử dụng rất nhiều trong các trường học và trong sản xuất. Tiêu biểu cho loại này là CPU 214. CPU 214 có các đặc tính như sau:

- Bộ nhớ chương trình (chứa trong EEPROM): 4096 Byte (4 kByte)
- Bộ nhớ dữ liệu (Vùng nhớ V): 4096 Byte (trong đó 512 Byte chứa trong EEPROM)
- Số lượng ngõ vào: 14 , và
- Số lượng ngõ ra: 10 ngõ ra digital tích hợp trong CPU
- Số module mở rộng: 7 gồm cả module analog
- Số lượng vào/ra số cực đại: 64
- Số lượng Timer :128 Timer chia làm 3 loại theo độ phân giải khác nhau: 4 Timer 1ms, 16 Timer 10 ms và 108 Timer có độ phân giải 100ms.
- Số lượng Counter: 128 bộ đếm chia làm hai loại: 96 Counter Up và 32 Counter Up/Down.
- Bit memory (Vùng nhớ M): 256 bit
- Special memory (SM) : 688 bit dùng để thông báo trạng thái và đặt chế độ làm việc.
- Có phép tính số học

- Bộ đếm tốc độ cao (High-speed counters): 2 counter 2 KHz và 1 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.

Sơ đồ bề mặt của bộ điều khiển logic khả trình S7-200 CPU 214 được cho như hình 4.1.



Hình 4.1: Bộ điều khiển lập trình S7-200 CPU 214

*** Mô tả các đèn báo trên CPU 214:**

- **SF (Đèn đỏ):** Đèn đỏ SF báo hiệu hệ thống bị lỗi. Đèn SF sáng lên khi PLC có lỗi.
- **RUN (Đèn xanh):** cho biết PLC đang ở chế độ làm việc và thực hiện chương trình được nạp vào trong bộ nhớ chương trình của PLC.
- **STOP (Đèn vàng):** Đèn vàng STOP chỉ định PLC đang ở chế độ dừng. Dừng chương trình đang thực hiện lại.
- **I x.x (Đèn xanh):** Đèn xanh ở cổng vào chỉ định trạng thái tức thời của cổng (x.x = 0.0 - 1.5). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.

- **Qy.y (Đèn xanh):** Đèn xanh ở cổng ra chỉ định trạng thái tức thời của cổng (y.y = 0.0 - 1.1). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.

Hiện nay, CPU 22x với nhiều tính năng vượt trội đã thay thế loại CPU 21x và hiện đang được sử dụng rất nhiều. Tiêu biểu cho loại này là CPU 224. Thông tin về CPU 22x được cho như bảng 4.1 và hình dáng CPU 224 ở hình 4.2.

Đặc điểm	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
I/O trên CPU Digital Analog	6DI/4DO -	8DI/6DO -	14DI/10DO -	14DI/10DO 2AI/1AO	24DI/16DO -
Số module mở rộng max.	0	2	7	7	7
Bộ nhớ chương trình	4KB	4KB	8KB	12KB	16KB
Bộ nhớ dữ liệu	2KB	2KB	8KB	10KB	10KB
Thời gian xử lý	0,37 μs	0,37 μs	0,37 μs	0,37 μs	0,37 μs
Memory bits/counters/timers	256/256/256	256/256/256	256/256/256	256/256/256	256/256/256
High-speed counters	4 x 30 kHz	4 x 30 kHz	6 x 30 kHz	4 x 30 kHz 2x 200 kHz	6 x 30 kHz
Real-time clock	card	card	Tích hợp	Tích hợp	Tích hợp
Ngõ ra xung	2 x 20 kHz	2 x 20 kHz	2 x 20 kHz	2 x 100 kHz	2 x 20 kHz
Cổng giao tiếp	1x RS-485	1x RS-485	1x RS-485	2x RS-485	2x RS-485
Biến trở analog trên CPU	1	1	2	2	2

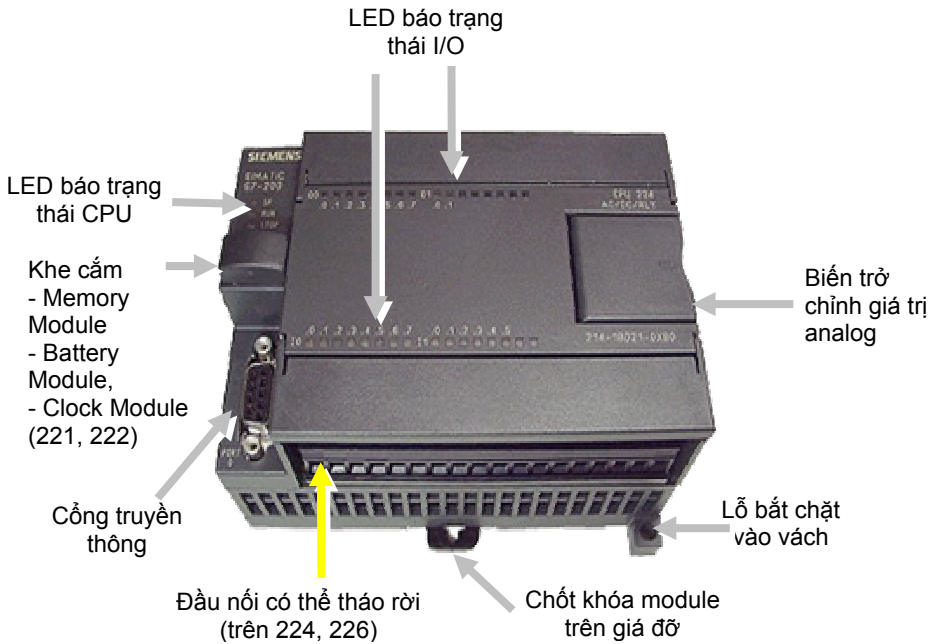
Bảng 4.1: Bảng dữ liệu về CPU họ 22x

*** 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.



Hình 4.2: Bộ điều khiển lập trình CPU 224

* Cổng truyền thông

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

Để ghép nối S7-200 với máy lập trình PG720 (hãng Siemens) hoặc với các loại máy lập trình thuộc họ PG7xx có thể sử dụng một cáp nối thẳng qua MPI. Cáp đi kèm theo máy lập trình.

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.

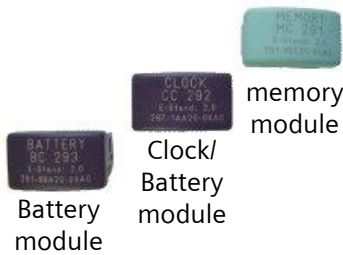
* Card nhớ, pin, clock (CPU 221, CPU222)

S7-200 cung cấp nhiều biện pháp đảm bảo cho chương trình người dùng, dữ liệu chương trình và cấu hình dữ liệu được duy trì sau:

Một tụ điện với điện dung lớn cho phép nuôi bộ nhớ RAM sau khi bị mất nguồn điện cung cấp. Tùy theo loại CPU mà thời gian lưu trữ có thể kéo dài nhiều ngày. Chẳng hạn ở CPU 224 là khoảng 100 giờ

Vùng nhớ EEPROM cho phép lưu chương trình, các vùng nhớ được người dùng chọn chứa vào EEPROM và cấu hình dữ liệu.

Cho phép gắn thêm Pin để nuôi RAM và cho phép kéo dài thêm thời gian lưu trữ dữ liệu, có thể lên đến 200 ngày kể từ khi mất nguồn điện. Nguồn của Pin sẽ được lấy sau khi tụ điện đã xả hết.



Hình 4.3: Hình dáng các module

- *Card nhớ*: Được sử dụng để lưu trữ chương trình. Chương trình chứa trong card nhớ bao gồm: program block, data block, system block, công thức (recipes), dữ liệu đo (data logs), và các giá trị cưỡng bức (force values).
- *Card pin*: Dùng để mở rộng thời gian lưu trữ các dữ liệu có trong bộ nhớ. Nguồn pin được tự động chuyển sang khi tụ trong PLC cạn. Pin có thể sử dụng đến 200 ngày.

- *Card Clock / Battery module*: đồng hồ thời gian thực (Real-time clock) cho CPU 221, 222 và nguồn pin để nuôi đồng hồ và lưu dữ liệu. Thời gian sử dụng đến 200 ngày.

* **Biến trở chỉnh giá trị analog:**

Hai biến trở này được sử dụng như hai ngõ vào analog cho phép điều chỉnh các biến cần phải thay đổi và sử dụng trong chương trình.

4.1.2 **Khởi mở rộng**

Trên các CPU đã tích hợp sẵn một số các ngõ vào và ngõ ra số, chẳng hạn như CPU 224 DC/DC/DC có sẵn 16 ngõ vào và 14 ngõ ra. Tuy nhiên trong thực tế, xuất phát từ yêu cầu điều khiển như: cần nhiều hơn số ngõ vào/ra có sẵn, có sử dụng tín hiệu analog hay có các yêu cầu về truyền thông, nối mạng các PLC...mà ta phải gắn thêm vào CPU các khối mở rộng (Expansion module) có các chức năng khác nhau (bảng 4.2).

4.1.2.1 **Digital module**

Các module số gắn thêm vào khối CPU để mở rộng số lượng các ngõ vào/ra số.

- *Khối ngõ vào số DI (Digital Input)*: Siemens sản xuất các khối ngõ vào số như: DI8 x 24VDC, DI8 x AC120/230V, DI16 x 24VDC.
- *Khối ngõ ra số (Digital Output)*: Các ngõ ra này được chia ra làm 3 loại là ngõ ra DC, ngõ ra AC và ngõ ra relay. Điện áp ngõ ra có thể là 24Vdc hoặc 230Vac tùy loại, với số lượng ngõ ra có thể là 4 hoặc 8.

Ngoài ra còn có sự kết hợp các ngõ vào và ra số trên cùng một module.

4.1.2.2 Analog module

Ngoại trừ CPU 224XP có tích hợp sẵn 2 ngõ vào và 1 ngõ ra analog (2AI/1AO) để kết nối với ngoại vi nhận và phát tín hiệu analog, thì hầu hết các CPU khác của họ S7-200 đều không có tích hợp sẵn. Vì vậy khi điều khiển với tín hiệu analog thì yêu cầu người sử dụng phải gắn thêm các khối analog.

- **Khối ngõ vào tương tự AI (Analog Input):** Tín hiệu analog ngõ vào có thể là tín hiệu điện áp hoặc dòng điện. Tùy thuộc vào tín hiệu analog cần đọc là loại nào mà người sử dụng có thể cài đặt cho phù hợp bằng các công tắc được gắn trên module (*Chi tiết xem chương xử lý tín hiệu analog*).

Hiện có các khối ngõ vào: 4AI, 8AI. Đối với tín hiệu analog được tạo ra bởi thermocoupe (cặp nhiệt) và RTD thì sử dụng các module đo nhiệt tương ứng (bảng 4.2).

- **Khối ngõ ra tương tự AO (Analog Output):** Tín hiệu tương tự này có thể là điện áp hoặc dòng điện tùy theo người dùng cài đặt. Tín hiệu ra là điện áp nằm trong khoảng $\pm 10Vdc$ tương ứng với giá trị số từ -32000 tới +32000 và tín hiệu dòng điện nằm trong khoảng từ 0 - 20mA tương ứng với giá trị số từ 0 tới +32000.

Ngoài các khối trên còn có các khối có sự kết hợp cả 2 loại tín hiệu vào và ra analog trên cùng một khối.

Các khối mở rộng	Loại			
Digital module				
Input	8 x DC In	8 x AC In	16 x DC In	
Output	4 x DC Out	4 x Relay	8 x Relay	
	8 x DC Out	8 x AC Out		
Tổ hợp	4 x DC In/	8 x DC In/	16 x DC In/	32 x DC In/
	4 x DC Out	8 x DC Out	16x DC Out	32x DC Out
	4 x DC In/	8 x DC In/	16 x DC In/	32 x DC In/
	4 x Relay	8 x Relay	16x DC Out	32x Relay
Analog module				
Input	4 x Analog In	8 x Analog In	4xThermocouple In	
	2 x RTD In	2 x RTD In		
Output	2 x Analog Out	4 x Analog Out		
Tổ hợp	4 x Analog In			
	4 x Analog Out			
Intelligent module				
	Position	Modem	PROFIBUS-DP	
	Ethernet	Ethernet IT		
Các module khác				
	AS-Interface	SIWAREX MS		

Bảng 4.2: Các loại khối mở rộng

4.1.2.3 Intelligent module

Các PLC S7-200 có thể nối vào các loại mạng khác nhau để tăng cường khả năng mở rộng, truyền thông với các thiết bị khác trong hệ thống tự động hóa.

- *Master trong mạng AS-Interface:* Giao tiếp AS-i (*Actuator Sensor Interface*) hay giao tiếp actuator/sensor là hệ thống kết nối cho cấp quá trình thấp nhất trong hệ thống tự động hóa nhằm tối ưu hóa việc kết nối cảm biến và cơ cấu chấp hành với thiết bị tự động hóa. Với module CP243-2 cho phép kết nối mạng AS-Interface vào PLC S7-200 và đóng vai trò là master.
- *Kết nối vào mạng PROFIBUS-DP:* Các PLC S7-200 có thể kết nối vào mạng Profibus hoạt động như một DP Slave nhờ vào khối mở rộng EM277. Việc sử dụng EM277 cho phép PLC S7-200 có thể kết nối truyền thông với các thiết bị trong mạng Profibus như: PLC S7-300, S7-400, màn hình điều khiển...
- *Kết nối vào mạng Ethernet:* Để có thể kết nối S7-200 vào mạng Industrial Ethernet thì cần có khối CP 243-1. Đây là khối truyền thông cho phép các PLC S7-200 có thể được cấu hình, lập trình, chẩn đoán từ xa qua Ethernet nhờ phần mềm STEP 7 Micro/win. Giúp cho các CPU S7-200 có thể giao tiếp với các S7-200 khác, S7-300 hay S7-400 qua Ethernet. Các CPU có thể sử dụng là họ CPU 22X. Có thể thực hiện cấu hình cho các CPU vào mạng Ethernet nhờ vào *Wizard (Menu Tools → Ethernet wizard)*.
- *Internet Technology:* Khối mở rộng CP 243-1 IT cho phép các CPU S7-200 có thể thực hiện các giám sát hay thay đổi qua trình duyệt Web từ một PC có nối mạng. Các thông báo chẩn đoán có thể gửi qua email từ một hệ thống. Sử dụng các chức năng IT cho phép trao đổi các tập tin dữ liệu với các máy tính hay các hệ thống điều khiển khác. Mỗi một khối CP 243-1IT chỉ nên kết nối cho 2 CPU S7-200.
- *Modem module:* Cho phép kết nối trực tiếp S7-200 vào đường dây điện thoại, và cung cấp truyền thông giữa S7-200 và Step 7- micro/Win.

Với công cụ *Modem Expansion wizard* cho phép thiết lập một modem ở xa hoặc kết nối S7-200 với một thiết bị ở xa qua modem.

Khả năng truyền thông của S7-200 được cho như hình 4.4.

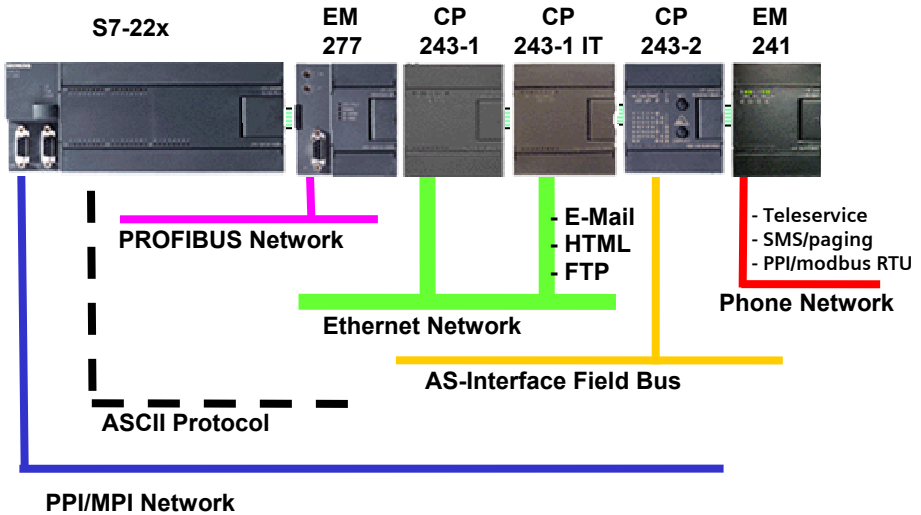
4.1.2.4 Function module

Là các khối chức năng thực hiện các chức năng đặc biệt như điều khiển vị trí (position module), cân (SIWREX MS).

- *Position module:* Module vị trí được sử dụng để điều khiển tốc độ và vị trí của động cơ bước (stepper motor) hoặc động cơ servo (servo motor). Với công cụ Position Control wizard trong phần mềm STEP 7--Micro/WIN

để thiết lập cấu hình cho module điều khiển vị trí. Module điều khiển vị trí thường được sử dụng là EM253.

SIWAREX MS: Là module cân đa năng và linh hoạt, nó được sử dụng với các hệ thống cân hoặc đo lực sử dụng PLC S7-200.

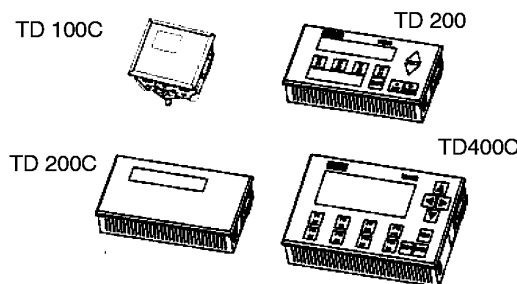


Hình 4.4: Khả năng truyền thông của PLC S7-200

4.2 Màn hình điều khiển

Trong các yêu cầu điều khiển có giám sát thì đối với các PLC S7-200 chúng ta có thể gắn thêm các màn hình để điều khiển và giám sát. Hiện có các loại là: màn hình hiển thị dòng văn bản (Text Display), màn hình điều khiển bằng bàn phím (Operator panel) và màn hình cảm ứng (Touch Panel).

* *Bảng điều khiển hiển thị dòng văn bản (Text Display):* Các màn hình này có giá thành thấp, cho phép người vận hành máy có thể xem, giám sát bằng các dòng văn bản và thay đổi các thông số hay chế độ hoạt động của hệ thống điều khiển bằng các phím trên bảng điều khiển. Gồm có các loại là TD100C, TD200C, TD 200, TD400C (hình 4.5).



Hình 4.5: Bảng điều khiển hiển thị dòng văn bản

Các bảng điều khiển này có thể được thiết lập các thông báo và nút nhấn điều khiển dễ dàng bằng công cụ **Text Display wizard** (menu lệnh *Tools > Text Display Wizard*) trong STEP 7--Micro/WIN.

* **Operator Panel và Touch Panel:** Các màn hình được ứng dụng điều khiển và giám sát các máy móc, thiết bị nhỏ. Thời gian thiết lập cấu hình và vận hành nhanh với phần mềm WinCC flexible. Gồm có các loại: OP 73micro, TP 177micro (màn hình này thay thế các màn hình trước TP 070/TP 170micro) (hình 4.6).



Hình 4.6: Màn hình OP 73micro và TP 177micro.

4.3 Các vùng nhớ

Bộ nhớ của các PLC S7-200 được chia ra làm các vùng nhớ như bảng 4.3.

* **Vùng nhớ đệm ngõ vào số I:**

CPU sẽ đọc trạng thái tín hiệu của tất cả các ngõ vào số ở đầu mỗi chu kỳ quét, sau đó sẽ chứa các giá trị này vào vùng nhớ đệm ngõ vào. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

* **Vùng nhớ đệm ngõ ra số Q:**

Trong quá trình xử lý chương trình CPU sẽ lưu các giá trị xử lý thuộc vùng nhớ ngõ ra vào đây. Tại cuối mỗi vòng quét CPU sẽ sao chép nội dung vùng nhớ đệm này và chuyển ra các ngõ ra vật lý. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

* **Vùng nhớ biến V:**

Sử dụng vùng nhớ V để lưu trữ các kết quả phép toán trung gian có được do các xử lý logic của chương trình. Cũng có thể sử dụng vùng nhớ để lưu trữ các dữ liệu khác liên quan đến chương trình hay nhiệm vụ điều khiển. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

* **Vùng nhớ M:**

Có thể coi vùng nhớ M như là các relay điều khiển trong chương trình để lưu trữ trạng thái trung gian của một phép toán hay các thông tin điều khiển khác. Có thể truy cập vùng nhớ này theo bit, Byte, Word hay Doubleword.

* **Vùng nhớ bộ định thời T:**

S7-200 cung cấp vùng nhớ riêng cho các bộ định thời, các bộ định thời được sử dụng cho các yêu cầu điều khiển cần trì hoãn thời gian. Giá trị thời gian sẽ được đếm tăng dần theo 3 độ phân giải là 1ms, 10ms và 100ms.

Mô tả	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU226
Kích thước chương trình người dùng	4 KB	4 KB	8 KB	12 KB	16 KB
Kích thước dữ liệu	2 KB	2 KB	8 KB	10 KB	10 KB
Vùng đếm vào số	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ... I15.7
Vùng đếm ra số	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ... Q15.7
Ngõ vào analog	AIW0 ... AIW30	AIW0 ... AIW30	AIW0 ... AIW62	AIW0 ... AIW62	AIW0 ... AIW62
Ngõ ra analog	AQW0 ... AQW30	AQW0 ... AQW30	AQW0 ... AQW62	AQW0 ... AQW62	AQW0 ... AQW62
Vùng nhớ biến (V)	VB0 ... VB2047	VB0 ... VB2047	VB0 ... VB8191	VB0 ... VB10239	VB0 ... VB10239
Vùng nhớ cục bộ (L)	LB0 ... LB63	LB0 ... LB63	LB0 ... LB63	LB0 ... LB63	LB0 ... LB63
Vùng nhớ bit (M)	M0.0 ... M31.7	M0.0 ... M31.7	M0.0 ... M31.7	M0.0 ... M31.7	M0.0 ... M31.7
Vùng nhớ đặc biệt Chỉ đọc (SM)	SM0.0 ... SM179.7 SM0.0 ... SM29.7	SM0.0 ... SM299.7 SM0.0 ... SM29.7	SM0.0 ... SM549.7 SM0.0 ... SM29.7	SM0.0 ... SM549.7 SM0.0 ... SM29.7	SM0.0 ... SM549.7 SM0.0 ... SM29.7
Timer Retentive on-delay	256 (T0...T255)	256 (T0...T255)	256 (T0...T255)	256 (T0...T255)	256 (T0...T255)
1ms	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
10ms	T1...T4, và T65...T68	T1...T4, và T65...T68	T1...T4, và T65...T68	T1...T4, và T65...T68	T1...T4, và T65...T68
100ms	T5...T31, và T69...T95	T5...T31, và T69...T95	T5...T31, và T69...T95	T5...T31, và T69...T95	T5...T31, và T69...T95
On/Off delay 1ms	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
10ms	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100	T33 ... T36, và T97 ... T100
100ms	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255	T37 ... T63, và T101 ... T255
Counter	C0 ... C255	C0 ... C255	C0 ... C255	C0 ... C255	C0 ... C255
Bộ đếm tốc độ cao	HC0 ... HC5	HC0 ... HC5	HC0 ... HC5	HC0 ... HC5	HC0 ... HC5
Bit điều khiển trình tự (S)	S0.0 ... S31.7	S0.0 ... S31.7	S0.0 ... S31.7	S0.0 ... S31.7	S0.0 ... S31.7
Thanh ghi Accu	AC0 ... AC3	AC0 ... AC3	AC0 ... AC3	AC0 ... AC3	AC0 ... AC3
Jumps/Labels	0 ... 255	0 ... 255	0 ... 255	0 ... 255	0 ... 255
Call/Subroutine	0 ... 63	0 ... 63	0 ... 63	0 ... 63	0 ... 127
Interrupt routines	0 ... 127	0 ... 127	0 ... 127	0 ... 127	0 ... 127
Ô nhớ sườn xung (positive/negative)	256	256	256	256	256
PID loops	0 ... 7	0 ... 7	0 ... 7	0 ... 7	0 ... 7
Port	Port 0	Port 0	Port 0	Port 0, Port 1	Port 0, Port 1

Bảng 4.3: Các vùng nhớ và đặc điểm của CPU S7-200.

* Vùng nhớ bộ đếm C:

Có 3 loại bộ đếm là bộ đếm lên, bộ đếm xuống và bộ đếm lên-xuống. Các bộ đếm sẽ tăng hoặc giảm giá trị hiện hành khi tín hiệu tại ngõ vào thay đổi trạng thái từ mức thấp lên mức cao.

** Vùng nhớ bộ đếm tốc độ cao HC (High speed Counter):*

Các bộ đếm tốc độ cao được sử dụng để đếm các sự kiện tốc độ cao độc lập với vòng quét của CPU. Giá trị đếm là số nguyên 32 bit có dấu. Để truy xuất giá trị đếm của các bộ đếm tốc độ cao cần xác định địa chỉ của bộ đếm tốc độ cao, sử dụng vùng nhớ HC và số của bộ đếm, ví dụ HC0. Giá trị đếm hiện hành của các bộ đếm tốc độ cao là các giá trị chỉ đọc và truy xuất theo double word.

** Các thanh ghi AC (Accumulators):*

Các thanh ghi AC là các phần tử đọc/ghi mà có thể được dùng để truy xuất giống như bộ nhớ. Chẳng hạn, có thể sử dụng các thanh ghi để truy xuất các thông số từ các chương trình con (Subroutine) và lưu trữ các giá trị trung gian để sử dụng cho tính toán. Các CPU S7-200 có 4 thanh ghi là AC0, AC1, AC2 và AC3. Chúng ta có thể truy xuất dữ liệu trong các thanh ghi này theo Byte, Word, và Doubleword.

** Vùng nhớ đặc biệt SM (Special Memory):*

Các bit SM là các phần tử cho phép truyền thông tin giữa CPU và chương trình người dùng. Có thể sử dụng các bit này để chọn lựa và điều khiển một số chức năng đặc biệt của CPU, chẳng hạn như bit lên mức 1 trong vòng quét đầu tiên, các bit phát ra các xung có tần số 1Hz...Chúng ta truy xuất vùng nhớ SM theo bit, byte, word, doubleword.

** Vùng nhớ cục bộ L (Local Memory Area):*

Vùng nhớ này có độ lớn 64 Byte, trong đó 60 byte có thể được dùng như vùng nhớ cục bộ hay chuyển các thông số tới các chương trình con, 4 byte cuối dùng cho hệ thống. Vùng nhớ này tương tự như vùng nhớ biến V chỉ khác ở chỗ các biến vùng nhớ V cho phép sử dụng ở tất cả các khối chương trình còn vùng nhớ L chỉ có tác dụng trong phạm vi soạn thảo của một khối chương trình mà thôi. Vị trí biến thuộc vùng nhớ L trong chương trình chính thì không thể sử dụng ở chương trình con và ngược lại.

** Vùng nhớ ngõ vào tương tự AI (Analog Inputs):*

Các PLC S7-200 chuyển giá trị một tương tự (chẳng hạn điện áp hay nhiệt độ) thành giá trị số và chứa vào một vùng nhớ 16 bit. Bởi vì các giá trị tương tự chiếm một vùng nhớ word nên chúng luôn luôn có các giá trị word chẵn, chẳng hạn như AIW0, AIW2, AIW4..và là các giá trị chỉ đọc.

** Vùng nhớ ngõ ra tương tự AQ (Analog Outputs):*

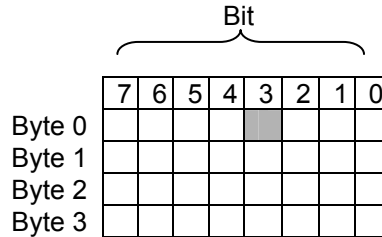
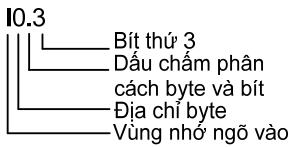
Các PLC S7-200 chuyển một giá trị số 16 bit sang giá trị điện áp hoặc dòng điện, tương ứng với giá trị số (digital). Giống như các ngõ vào tương tự chúng ta chỉ có thể truy xuất các ngõ ra tương tự theo word. Và là các giá trị word chẵn, chẳng hạn như AQW0, AQW2, AQW4.

4.4 Qui ước địa chỉ trong PLC S7-200

4.4.1 Truy xuất theo bit

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

Ví dụ:



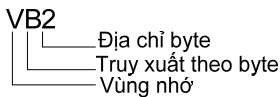
Hình 4.7: Vùng nhớ ngõ vào I

Trong hình 4.7 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.

4.4.2 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.

Ví dụ:

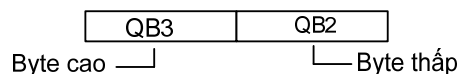
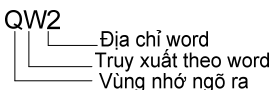


Tương tự như ví dụ ta khai báo cho các vùng nhớ khác, chẳng hạn như IB3, MB2, QB5..

4.4.3 Truy xuất theo word (16 bit)

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

Ví dụ:



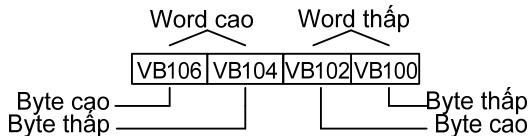
Chú ý:

- 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.

4.4.4 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



Bảng tóm tắt việc truy xuất các vùng nhớ theo bit, byte, word và double word được cho ở bảng 4.4.

Cách truy xuất	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226	
Truy xuất Bit (byte.bit)	I 0.0 ... 15.7 Q 0.0 ... 15.7 V 0.0 ... 2047.7 M 0.0 ... 31.7 SM 0.0 ... 165.7 S 0.0 ... 31.7 T 0 ... 255 C 0 ... 255 L 0.0 ... 63.7	0.0 ... 15.7 0.0 ... 15.7 0.0 ... 2047.7 0.0 ... 31.7 0.0 ... 299.7 0.0 ... 31.7 0 ... 255 0 ... 255 0.0 ... 63.7	0.0 ... 15.7 0.0 ... 15.7 0.0 ... 2047.7 0.0 ... 31.7 0.0 ... 549.7 0.0 ... 31.7 0 ... 255 0 ... 255 0.0 ... 63.7	0.0 ... 15.7 0.0 ... 15.7 0.0 ... 8191.7 0.0 ... 31.7 0.0 ... 549.7 0.0 ... 31.7 0 ... 255 0 ... 255 0.0 ... 63.7	0.0 ... 15.7 0.0 ... 15.7 0.0 ... 10239.7 0.0 ... 31.7 0.0 ... 549.7 0.0 ... 31.7 0 ... 255 0 ... 255 0.0 ... 63.7	0.0 ... 15.7 0.0 ... 15.7 0.0 ... 10239.7 0.0 ... 31.7 0.0 ... 549.7 0.0 ... 31.7 0 ... 255 0 ... 255 0.0 ... 63.7
Truy xuất Byte	IB 0 ... 15 QB 0 ... 15 VB 0 ... 2047 MB 0 ... 31 SMB 0 ... 165 SB 0 ... 31 LB 0 ... 63 AC 0 ... 3 KB (Constant)	0 ... 15 0 ... 15 0 ... 2047 0 ... 31 0 ... 299 0 ... 31 0 ... 63 0 ... 3 KB (Constant)	0 ... 15 0 ... 15 0 ... 8191 0 ... 31 0 ... 549 0 ... 31 0 ... 63 0 ... 3 KB (Constant)	0 ... 15 0 ... 15 0 ... 10239 0 ... 31 0 ... 549 0 ... 31 0 ... 63 0 ... 255 KB (Constant)	0 ... 15 0 ... 15 0 ... 10239 0 ... 31 0 ... 549 0 ... 31 0 ... 63 0 ... 255 KB (Constant)	
Truy xuất Word	IW 0 ... 14 QW 0 ... 14 VW 0 ... 2046 MW 0 ... 30 SMW 0 ... 164 SW 0 ... 30 T 0 ... 255 C 0 ... 255 LW 0 ... 62 AC 0 ... 3 AIW 0 ... 30 AQW 0 ... 30 KW (Constant)	0 ... 14 0 ... 14 0 ... 2046 0 ... 30 0 ... 298 0 ... 30 0 ... 255 0 ... 255 0 ... 62 0 ... 3 0 ... 30 0 ... 30 KW (Constant)	0 ... 14 0 ... 14 0 ... 8190 0 ... 30 0 ... 548 0 ... 30 0 ... 255 0 ... 255 0 ... 62 0 ... 3 0 ... 62 0 ... 62 KW (Constant)	0 ... 14 0 ... 14 0 ... 10238 0 ... 30 0 ... 548 0 ... 30 0 ... 255 0 ... 255 0 ... 62 0 ... 3 0 ... 62 0 ... 62 KW (Constant)	0 ... 14 0 ... 14 0 ... 10238 0 ... 30 0 ... 548 0 ... 30 0 ... 255 0 ... 255 0 ... 62 0 ... 3 0 ... 62 0 ... 62 KW (Constant)	
T. xuất Double word	ID 0 ... 12 QD 0 ... 12 VD 0 ... 2044	0 ... 12 0 ... 12 0 ... 2044	0 ... 12 0 ... 12 0 ... 8188	0 ... 12 0 ... 12 0 ... 10236	0 ... 12 0 ... 12 0 ... 10236	

MD	0 ... 28	0 ... 28	0 ... 28	0 ... 28	0 ... 28
SMD	0 ... 162	0 ... 296	0 ... 546	0 ... 546	0 ... 546
SD	0 ... 28	0 ... 28	0 ... 28	0 ... 28	0 ... 28
LD	0 ... 60	0 ... 60	0 ... 60	0 ... 60	0 ... 60
AC	0 ... 3	0 ... 3	0 ... 3	0 ... 3	0 ... 3
HC	0 ... 5	0 ... 5	0 ... 5	0 ... 5	0 ... 5
KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)

Bảng 4.4: Truy xuất các vùng nhớ theo địa chỉ bit, byte, word, 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...

4.5 Xử lý chương trình

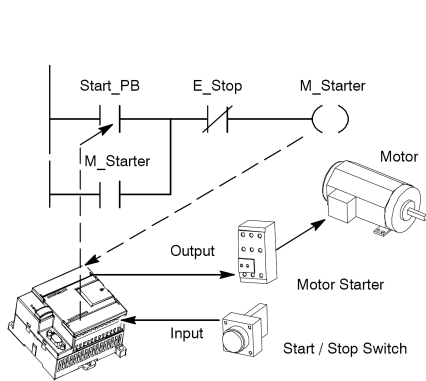
S7-200 thực hiện đọc và ghi dữ liệu theo logic điều khiển trong chương trình liên tục theo chu kỳ.

Hoạt động của S7-200 rất đơn giản:

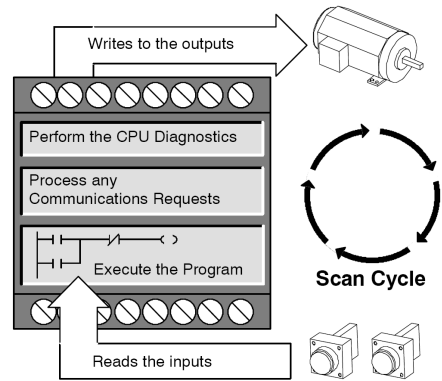
- Đọc trạng thái các ngõ vào
- S7-200 sử dụng các ngõ vào này để thực hiện logic điều khiển theo chương trình được lưu trữ trong nó. Dữ liệu luôn được cập nhật khi chương trình được thực hiện.
- Xuất dữ liệu ra ngõ ra.

Hình 4.8 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. 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 4.8: Điều khiển ngõ vào và ra



Hình 4.9: Chu kỳ quét S7-200

* Chu kỳ quét trong S7-200

S7-200 thực hiện một loạt các nhiệm vụ theo chu kỳ. Việc thực hiện các nhiệm vụ theo chu kỳ được gọi là chu kỳ quét (scan cycle). Hình 4.9 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 vào 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.

- **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.

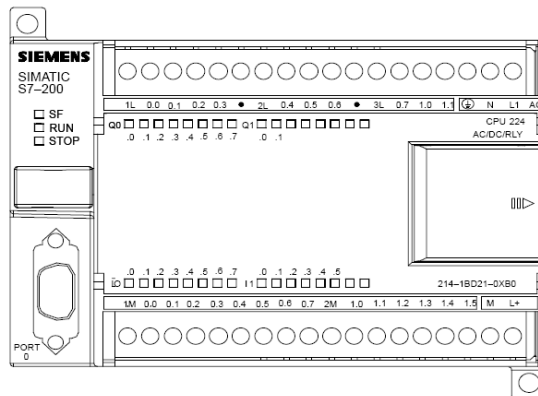
5 KẾT NỐI DÂY GIỮA PLC VÀ THIẾT BỊ NGOẠI VI

5.1 Kết nối dây giữa PLC và các 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.

5.1.1 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 5.1.



Hình 5.1: Bộ điều khiển lập trình S7-200 CPU 224

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

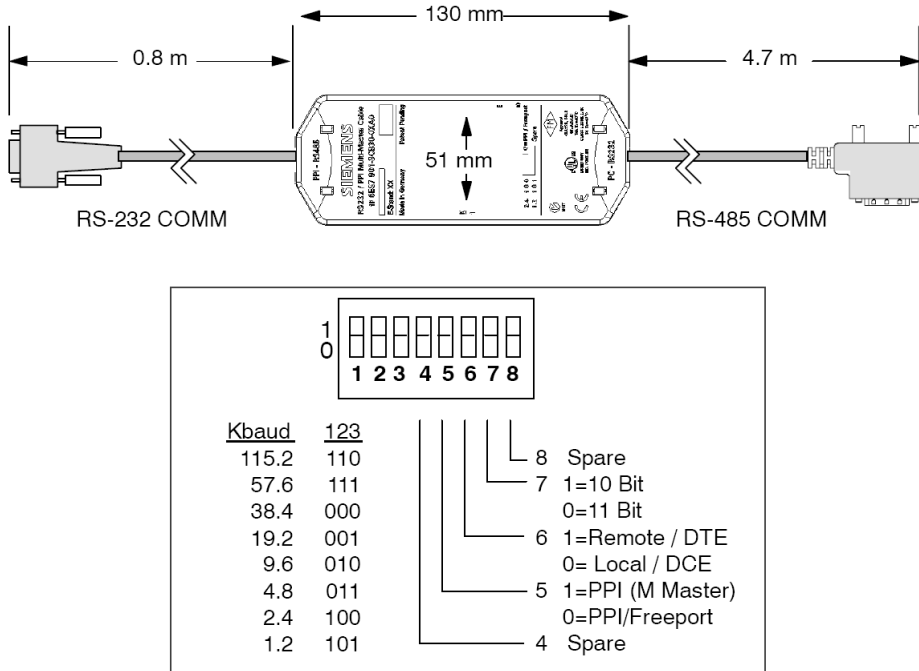
với PLC. Như vậy, để hệ thống điều khiển khiển bằng PLC hoạt động cũng như lập trình cho nó, cần phải kết nối PLC với máy tính cũng như các ngõ vào ra với ngoại vi.

5.1.2 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 5.2.



Hình 5.2: 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 123 được đặt theo thứ tự là 010).

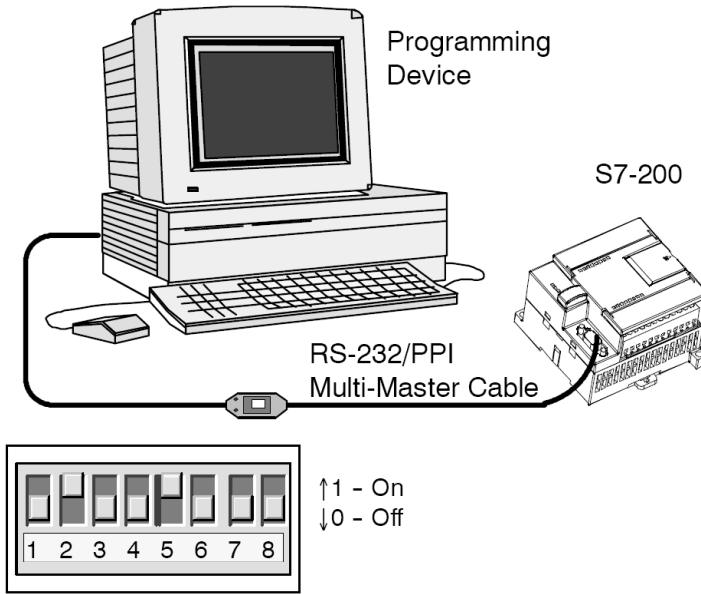
Tùy theo truyền thông là 10 Bit hay 11 Bit mà công tắc 7 được đặt ở vị trí thích hợp. Khi kết nối bình thường với máy tính thì công tắc 7 chọn ở chế độ truyền thông 11 Bit (công tắc 7 đặt ở vị trí 0).

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).

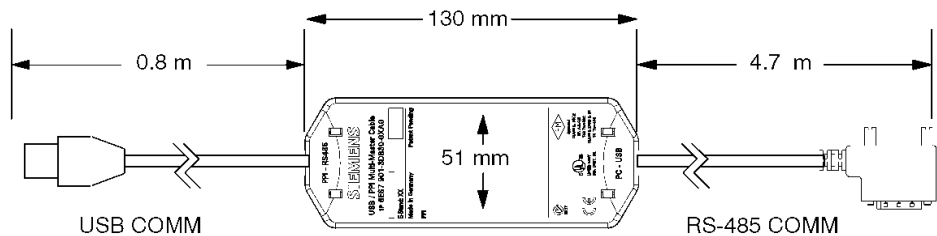
Sơ đồ nối cáp RS-232/PPI Multi-Master giữa máy tính và CPU S7-200 với tốc độ truyền 9,6 Kbaud được cho như hình 5.3.



Hình 5.3: 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 5.4.



Hình 5.4: 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 Service Pack 4 (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 TP070 từ phần mềm TP Designer.

5.1.3 Nối nguồn cung cấp cho CPU

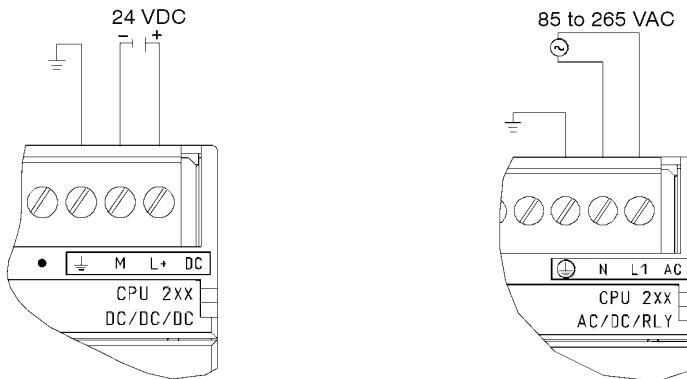
Tùy theo loại và họ PLC mà các CPU có thể là khối riêng hoặc có đặt sẵn các ngõ vào và ra cũng như một số chức năng đặc biệt khác. Hầu hết các PLC họ S7-200 được nhà sản xuất lắp đặt các khâu vào, khâu ra và CPU trong cùng một vỏ hộp. Nhưng nguồn cung cấp cho các khâu này hoàn toàn độc lập nhau. Nguồn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

Xoay chiều: 20...29 VAC , f = 47...63 Hz;

85...264 VAC, f = 47...63 Hz

Một chiều: 20,4 ... 28,8 VDC

Hình 5.5 a,b là sơ đồ nối dây nguồn cung cấp cho CPU



a. Cấp nguồn cho CPU 2xx loại DC/DC/DC;

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

Hình 5.5: 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.

5.1.4 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.

5.1.4.1 Kết nối các ngõ vào số với ngoại vi

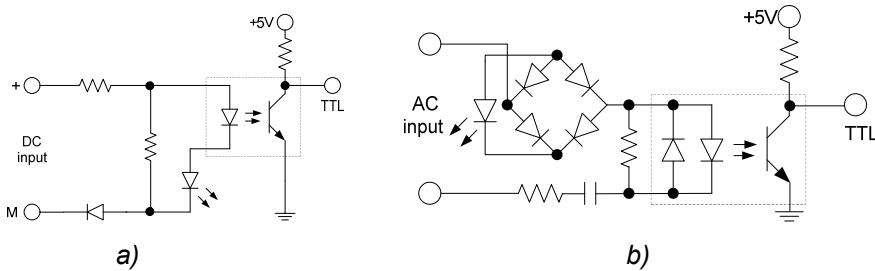
Các ngõ vào số của PLC có thể được chế tạo là một khối riêng, hoặc kết hợp với các ngõ ra chung trong một khối hoặc được tích hợp trên khối CPU. Trong trường hợp nào cũng vậy, các ngõ vào cũng phải được cung cấp nguồn riêng với cấp điện áp tùy thuộc vào loại ngõ vào. Cần lưu ý trong một khối ngõ vào 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...35 VAC, f = 47...63 Hz; dòng cần thiết nhỏ nhất 4mA

 79...135 VAC, f = 47...63 Hz; dòng cần thiết nhỏ nhất 4mA

 Một chiều: 15 ... 30 VDC; dòng cần thiết nhỏ nhất 4mA

Sơ đồ mạch điện bên trong của một số ngõ vào được cho như hình 5.6a,b.



Hình 5.6: 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

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

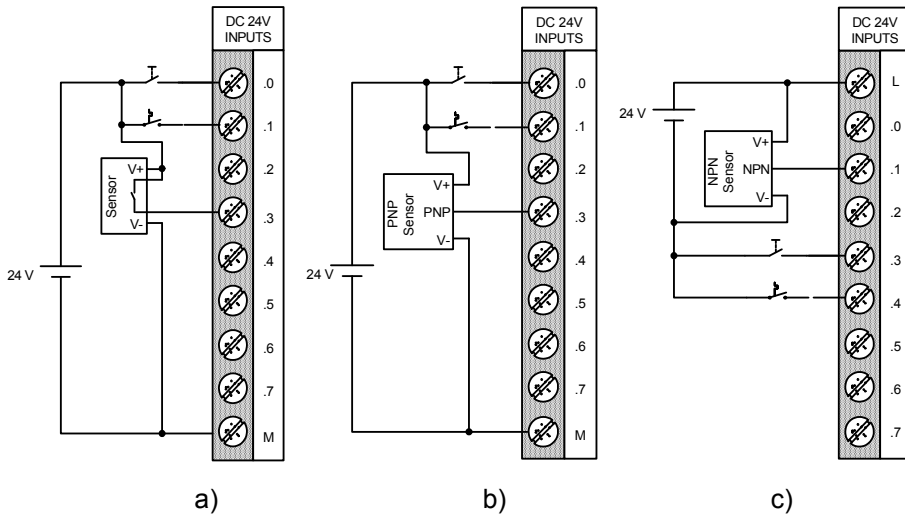
- + Ngõ vào DC:
 - Điện áp DC thường thấp do đó an toàn hơn.
 - Đáp ứng ngõ vào DC rất nhanh.
 - Điện áp DC có thể được kết nối với nhiều phần tử trong hệ thống điện.

- + 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 hiện hữu.

Đối với các ngõ vào số, khi kết nối với 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 Hình 5.7a,b,c minh họa cách kết nối dây các ngõ vào PLC với các bộ tạo tín hiệu nhị phân khác nhau.

Cần lưu ý đến các loại cảm biến khi kết nối với các ngõ vào PLC (*xem lại chương 3: cảm biến và cơ cấu chấp hành trong điều khiển logic*).

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



Hình 5.7: Kết nối ngõ vào với ngoại vi.

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

Đối với các ngõ vào ra của CPU 214 DC/DC/DC, CPU 224 AC/DC/Relay theo sơ tay được kết nối như hình 5.10 và hình 5.11.

5.1.4.2 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:

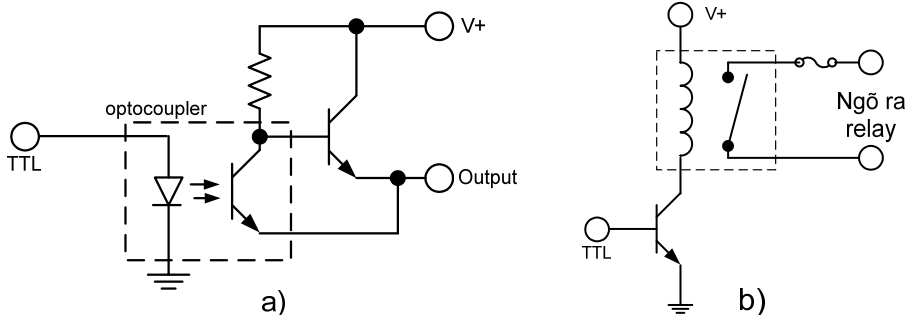
Xoay chiều: 20...264 VAC , f = 47...63 Hz;

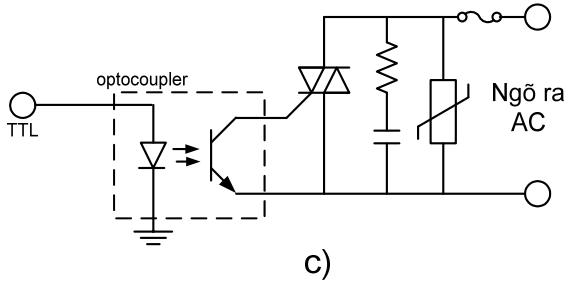
Một chiều: 5...30 VDC đối với ngõ ra rơ le; 20.4 ... 28.8 VDC đối với ngõ ra transistor;

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 5.8.

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.





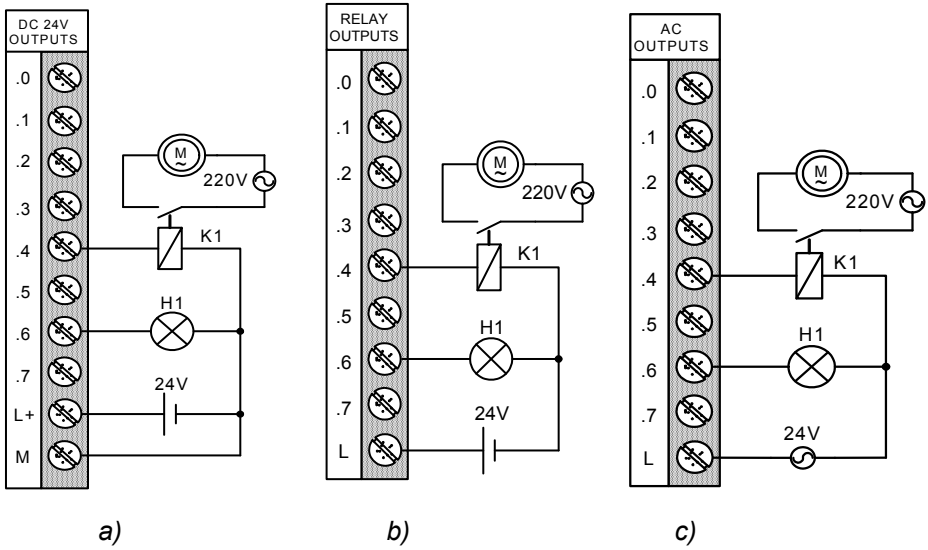
Hình 5.8: Mạch điện bên trong của các loại ngõ ra khác nhau.

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

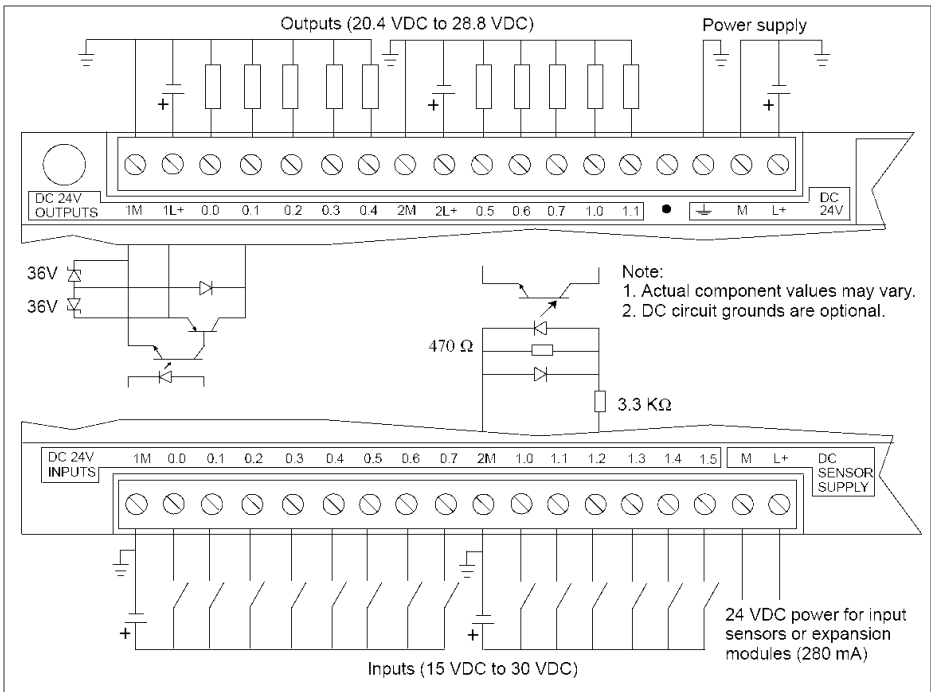
Đối với các ngõ ra số, khi kết nối với ngoại vi, ngoại trừ các trường hợp đặc biệt thì thông thường mỗi một ngõ ra được kết nối với một đối tượng điều khiển nhận tín hiệu nhị phân như: đèn báo, cuộn dây rơ le, chuông báo Hình 5.9 minh họa cách kết nối dây các ngõ ra PLC với các cơ cấu chấp hành. Hình 5.9a là một ví dụ cho các khối ra sử dụng 24Vdc với mass chung. Tiêu biểu cho loại này là ngõ ra transistor. Trong ví dụ này các ngõ ra được kết nối với tải công suất nhỏ là đèn báo và cuộn dây relay. Quan sát mạch kết nối này, đèn báo sử dụng nguồn cung cấp là 24Vdc. Nếu ngõ ra .6 ở mức logic "1" (24Vdc) thì dòng sẽ chảy từ ngõ ra .6 qua đèn H1 và xuống Mass (M), đèn sáng. Nếu ngõ ra ở mức logic "0" (0V), thì đèn H1 tắt. Nếu ngõ ra .4 ở mức logic "1" thì cuộn dây rơ le có điện, làm tiếp điểm của nó đóng lại cung cấp điện 220 Vac cho động cơ.

Hình 5.9b là một ví dụ ngõ ra relay sử dụng nguồn cấp là 24 Vdc, và hình 5.9c là ví dụ ngõ ra triac sử dụng nguồn xoay chiều 24 Vac.

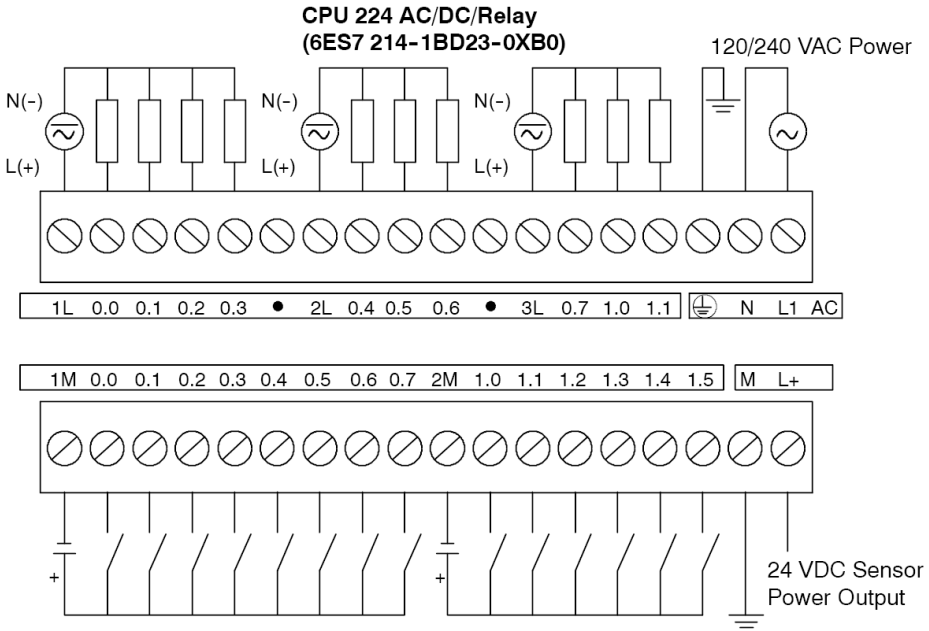
Một chú ý quan trọng khi kết nối các ngõ ra cần tra cứu sổ tay khối ngõ ra hiện có để có được thông tin chính xác tránh được những sự cố đáng tiếc xảy ra. Hình 5.10 là ví dụ của CPU 214 với nguồn cung cấp DC, ngõ vào DC và ngõ ra DC được nối dây với ngoại vi (trích từ sổ tay S7-200 Programmable Controller 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.



Hình 5.9: Kết nối dây ngõ ra PLC với cơ cấu chấp hành



Hình 5.10: Sơ đồ nối dây CPU 214 DC/DC/DC với nguồn và ngoại vi



Hình 5.11: Sơ đồ nối dây CPU 224 AC/DC/Relay với nguồn và ngoại vi

5.2 Kiểm tra việc kết 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 Step 7 Micro/Win (phần mềm lập trình cho họ S7-200) có trang bị thêm phần này đó là mục **Status Chart**.

Để sử dụng phần mềm tốt hơn hãy xem thêm chương “Phần mềm STEP 7-Micro/Win và ngôn ngữ lập trình”.

5.2.1 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

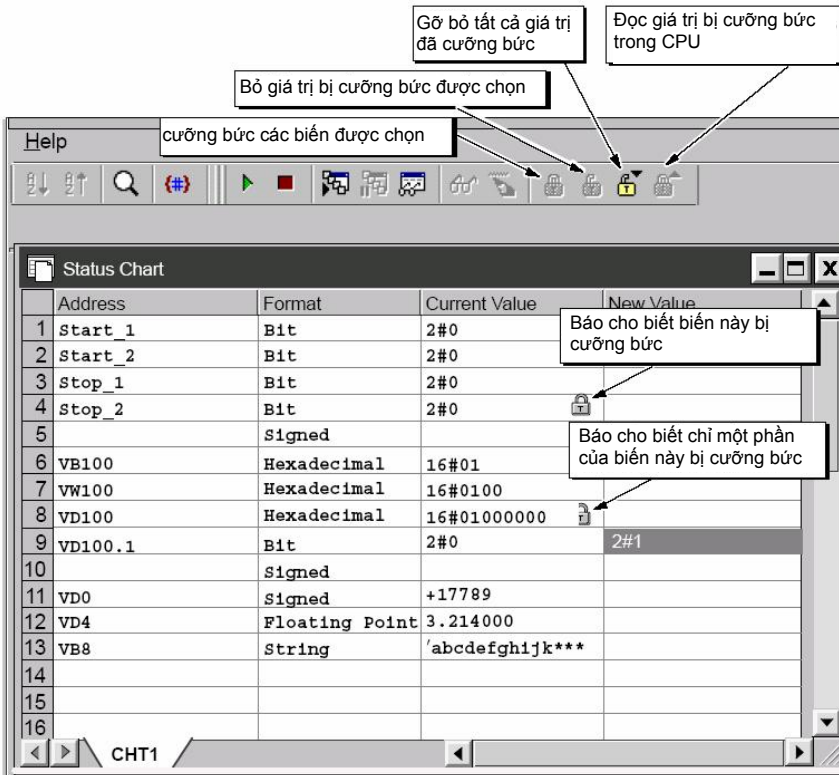


đúp chuột vào biểu tượng Status Chart trong cửa sổ Navigation Bar trên màn hình Step 7-Micro/Win32 hoặc vào mục View → Component → Status Chart.

5.2.2 Giám sát và thay đổi biến với Status Chart

Hình 5.9 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*.
- 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.



Hình 5.13: Ví dụ về status chart


5.2.3 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 Status Chart với một giá trị xác định, thực hiện các bước sau:


Bước 1: Chọn một ô trong cột Address, vào địa chỉ hay 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 cưỡng 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 cưỡng bức. Nhấp

chuột vào biểu tượng  hoặc chọn *Debug* → *Force* ở trên vị trí giá trị hiện hành để cưỡng bức biến giá trị đó.

Bước 4: Để cưỡng bức một giá trị mới 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 cưỡng 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 cưỡng bức một biến, hãy chọn biến mong muốn và nhấp chuột vào biểu tượng  hoặc chọn *Debug* → *Unforce*.

5.2.4 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 sai và sửa chữa.

5.3 Câu hỏi và bài tập

BT 5.1: Ngõ vào của PLC có thể đóng điện cho cuộn dây rơ le để điều khiển một động cơ được không? Các khối vào và khối ra đóng vai trò gì trong việc giao tiếp giữa PLC và thiết bị ngoại vi?

BT 5.2: Các khối mở rộng ngõ vào/ra có lợi ích gì?

BT 5.3: Điều gì xảy ra nếu một ngõ ra AC được cấp nguồn DC?

BT 5.4: Một khối vào/ra mở rộng của PLC họ S7-200 loại EM223 gồm có 8 ngõ vào DC/8 ngõ ra rơle. Các ngõ vào được nối với 4 nút nhấn, 2 ngõ ra được nối với một rơle trung gian sử dụng nguồn 24VDC dùng để đóng mạch cho một contactor 220VAC để điều khiển động cơ 3 pha 220V/380V. 2 ngõ ra được nối với 2 đèn báo 220VAC để báo chiều quay của động cơ. 2 ngõ ra được sử dụng cho các van khí nén 24 VDC. Hãy vẽ sơ đồ nối dây các ngõ vào và ra này với ngoại vi theo yêu cầu.

BT 5.5: Hãy thiết kế một dự án được điều khiển bằng PLC. Trước khi đặt hàng, cần phải phác thảo việc nối dây cơ bản và chọn lựa các loại PLC hoặc khối vào/ra có các ngõ vào/ra tương ứng. Các thiết bị được sử dụng để nối với các ngõ vào gồm có: 2 công tắc hành trình, 1 nút nhấn thường hở, 1 nút nhấn thường đóng và một tiếp điểm nhiệt. Ngõ ra sẽ điều khiển một van solenoid 24VDC, một đèn báo 110VAC và một động cơ 220VAC/50HP. Hãy lựa chọn loại PLC hoặc một khối vào/ra phù hợp và kết nối dây theo yêu cầu đặt ra.

BT 5.6: Hãy phác thảo sơ đồ nối dây cho các ngõ ra PLC theo yêu cầu được liệt kê dưới đây:

- Một van khí nén có 2 cuộn dây solenoid
- Một đèn báo 24VDC
- Một đèn báo 120 VAC
- Một động cơ công suất thấp 12 VDC.

6 Phần mềm Micro/Win và ngôn ngữ lập trình

6.1 Cài đặt phần mềm STEP 7-Micro/WIN

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 6.

6.1.1 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 Professional.
- 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 (multipoint interface).
- EM241 modem
- CP243-1 hoặc CP243-1 IT Ethernet

6.1.2 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. 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.

3. Sau đó sẽ nhận được dần dần 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 Service Pack 6 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_V40_SP6.exe và thực hiện theo các bước sau:

Bước 1: Uninstall phiên bản STEP 7-Micro/WIN V4.0 bằng công cụ “control panel” trong Window (menu Start → settings → control panel → add or remove program).

Bước 2: Khởi động lại máy tính

Bước 3: Cài đặt STEP 7-Micro/WIN V4.0 Service Pack (SP6) bằng cách nhấp đúp chuột vào file STEP7-MicroWIN_V40_SP6.exe.

6.2 Các phần tử cơ bản trong chương trình PLC S7-200

Các phần tử cơ bản trong một chương trình PLC S7-200 là:

1. Chương trình chính (main program)
2. Chương trình con (subroutine)
3. Chương trình ngắt (interrupt routine)
4. Khối hệ thống (system block)
5. Khối dữ liệu (data block)

6.2.1 Chương trình chính OB1 (main program)

Đây là phần khung của chương trình, chứa các lệnh điều khiển chương trình ứng dụng. Với một số chương trình điều khiển nhỏ, đơn giản chúng ta có thể viết tất cả các lệnh trong khối này. Chương trình ứng dụng được xử lý bắt đầu từ chương trình chính, các lệnh được xử lý lần lượt từ trên xuống dưới và chỉ một lần ở mỗi vòng quét. Trong S7-200 chương trình được chứa trong khối OB1.

6.2.2 Chương trình con SUB (subroutine)

Các lệnh viết trong chương trình con chỉ có thể được xử lý khi chương trình con được gọi (Call) từ chương trình chính, từ một chương trình con khác hoặc từ một 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.

(Bạn đọc xem phần ví dụ và cách sử dụng chương trình con ở chương “phép toán nhị phân”).

6.2.3 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.

(Phần chương trình ngắt sẽ được trình bày chi tiết ở tập 2).

6.2.4 Khối hệ thống (system block)

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

6.2.5 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.

(Phần khối dữ liệu sẽ được trình bày chi tiết ở tập 2).

6.3 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 tùy chọn.

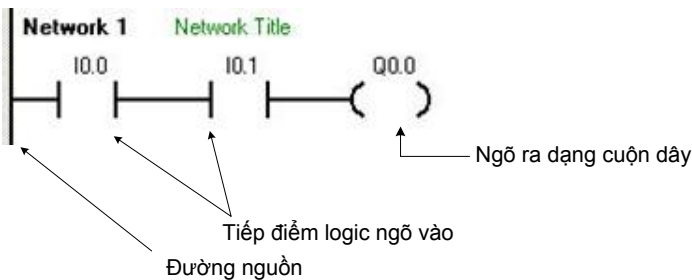
6.3.1 Dạng hình thang : LAD (Ladder logic)

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 (hoặc trạng thái nội của biến): ()
- 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:



Dạng soạn thảo này có một số ưu điểm:

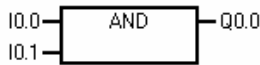
- Dễ dàng cho những người mới bắt đầu lập trình
- Biểu diễn dạng đồ họa dễ hiểu và thông dụng
- Luôn luôn có thể chuyển từ dạng STL sang LAD

6.3.2 Dạng khối chức năng : 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:

Network 1 Network Title



Đầu ra của các cổng logic hay hộp chức năng có thể được sử dụng để nối tiếp với đầu vào của các cổng 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 cổng chức năng giúp chúng ta dễ đọc hiểu theo trình tự điều khiển.
- Luôn có thể chuyển từ hiển thị dạng FBD sang STL.

6.3.3 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 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
À	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 PLC.
- STL cho phép giải quyết một số vấn đề mà đôi khi khó khăn khi dùng LAD hoặc FBD.
- 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.

6.4 Soạn thảo chương trình với phần mềm STEP7-Micro/Win V4.0 SP6

6.4.1 Mở màn hình soạn thảo chương trình

Để mở STEP 7--Micro/WIN, nhấp đúp chuột vào biểu tượng STEP 7-



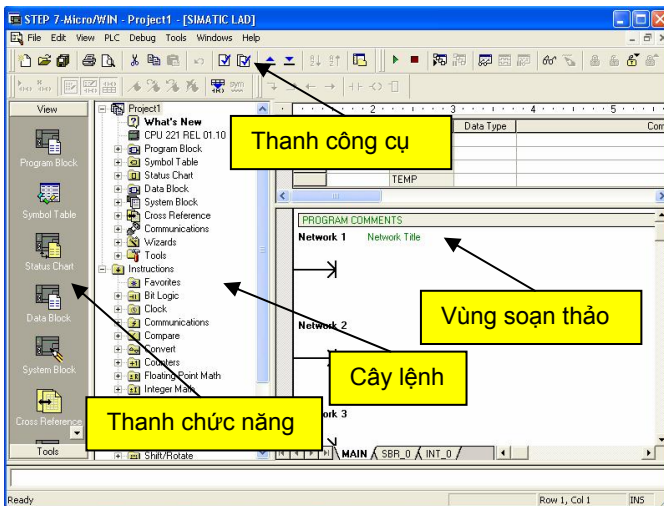
Micro/WIN trên màn hình desktop, hoặc chọn **Start > SIMATIC > STEP 7 MicroWIN V4.0**. Giao diện màn hình có dạng (hình 6.1).

6.4.1.1 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.

6.4.1.2 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.



Hình 6.1: Màn hình soạn thảo chương trình STEP 7-Micro/Win

6.4.1.3 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.

* **Program Block:**  Program Block

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

* **Symbol Table:**  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 gợi 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.

		Symbol	Address	Comment
1		START	I0.0	nut nhan khoi dong
2		M1	Q0.0	Dong co 1
3				

Tên gọi nhớ → Địa chỉ tuyệt đối → Chú thích



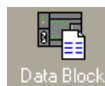
*** Status Chart:**

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.

	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*: Hiển thị giá trị hiện hành của địa chỉ
- + *Cột New Value*: Cho phép thay đổi trạng thái ngõ ra hay nội dung vùng nhớ



*** 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. Ví dụ:



Cửa sổ soạn thảo dữ liệu:

```
//Du lieu bon chua 1 (Tank 1)
//
VD100  500.0          // So lit maximum
VD104          // so lit hien hanh
VW106          // chua ma loi
```

Tank 1 / Tank 2



*** System Block :** 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 tắt 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. Có 4 cấp để người dùng tùy chọn theo bảng sau:

Mô tả chức năng	Level 1	Level 2	Level 3	Level 4
Đọc và ghi dữ liệu	Cho phép truy cập	Cho phép truy cập	Cho phép truy cập	Cho phép truy cập
Start, Stop, khởi động CPU				
Đọc và ghi đồng hồ thời gian (time-of-day)				

Clock)				
Upload chương trình, dữ liệu, cấu hình CPU				Không bao giờ cho phép
Download chương trình, data block hoặc system block				Yêu cầu password (không bao giờ cho phép với system Block)
Soạn thảo ở Runtime				Không bao giờ cho phép
Xóa chương trình, data block hoặc system block			Yêu cầu password	Yêu cầu password (không bao giờ cho phép với system Block)
Copy chương trình, data block hoặc system block vào card nhớ		Yêu cầu password		
Cưỡng bức dữ liệu trong status chart				Yêu cầu password
Ghi ngõ ra ở trạng thái stop				
Xóa tốc độ quét trong PLC information				
So sánh dự án				Không bao giờ cho phép



*** 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.

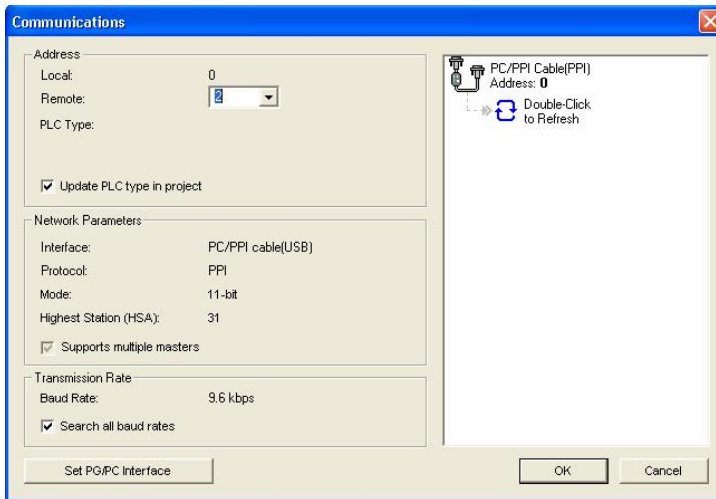
Một ví dụ bảng cross reference được cho ở hình 6.2. Tại cột Element, nhấp đúp vào địa chỉ nào thì trình soạn thảo sẽ mở cho chúng ta cửa sổ chương trình có chứa địa chỉ tương ứng. Việc này giúp cho chúng ta dễ dàng kiểm tra hay thay đổi địa chỉ khi có nhu cầu.

	Element	Block	Location	Context
1	I0.0	MAIN (OB1)	Network 1	- -
2	I0.0	MAIN (OB1)	Network 2	- -
3	I0.2	MAIN (OB1)	Network 1	- -
4	Q0.0	MAIN (OB1)	Network 2	-{-}
5	Q0.1	SBR_0 (SBR0)	Network 1	-{-}
6	C1	MAIN (OB1)	Network 1	CTU
7	T40	SBR_0 (SBR0)	Network 1	- -

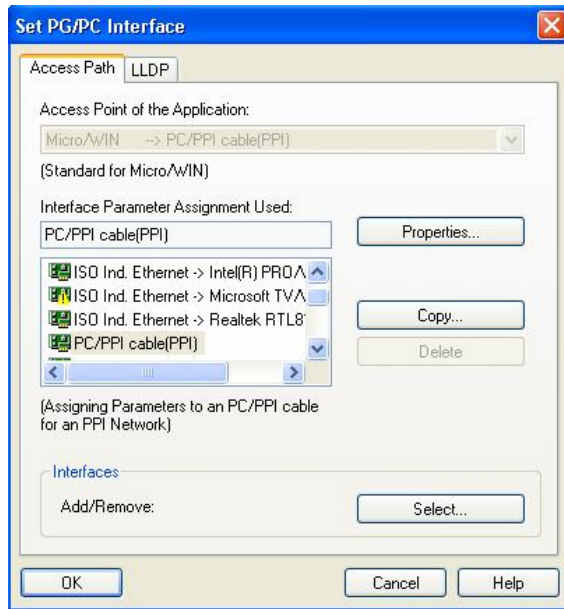
Hình 6.2: Ví dụ bảng cross reference.



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 6.3: Cửa sổ Communications



Hình 6.4: Cửa sổ Set PG/PC Interface.

6.4.2 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.



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

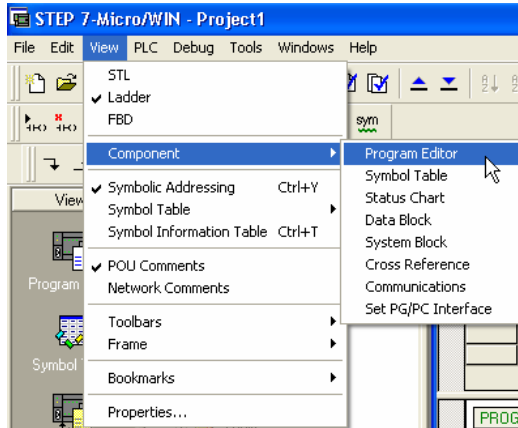


Unforce All (Debug menu): Gỡ bỏ tất cả các cưỡng bức trong bảng Status Chart.



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

6.4.3 Tạo một dự án STEP 7-Micro/WIN




Hình 6.5: Đường dẫn vào màn hình soạn thảo chương trình.

chương trình (hình 6.5).

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


6.4.3.1 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

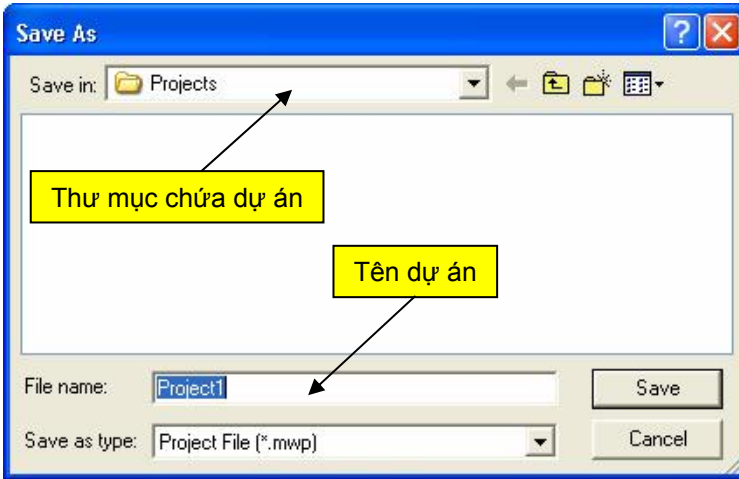


vào biểu tượng , hoặc vào menu **View > Component > Program Editor** để mở màn hình soạn thảo

Để soạn thảo bằng 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 như trình bày ở mục 6.4.1.3. Phần chi tiết sẽ được trình bày trong chương phép toán nhị phân.


6.4.3.2 Lưu dự án

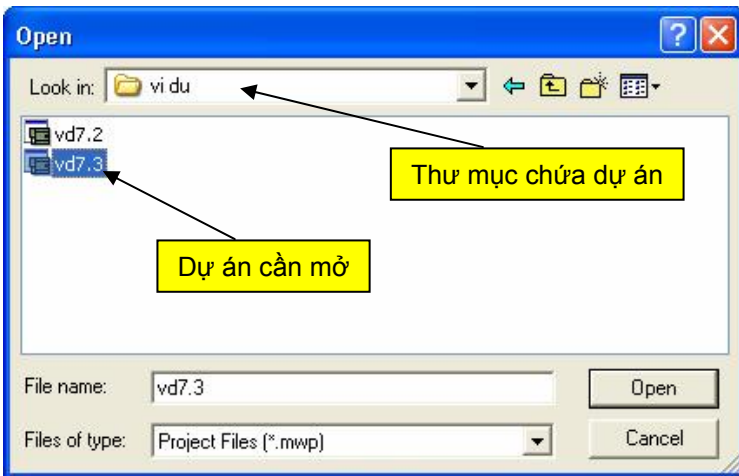
Để lưu dự án, nhấp chuột vào biểu tượng , hoặc vào menu **File > Save**. Cửa sổ màn hình xuất hiện như hình 6.6. Chọn thư mục cần chứa dự án, đặt tên dự án và nhấp chuột vào thẻ **Save** để lưu dự án



Hình 6.6: Cửa sổ màn hình lưu dự án

6.4.3.3 Mở một dự án

Để mở một dự án đang có sẵn, nhấp chuột vào biểu tượng , hoặc vào menu **File > Open**. Cửa sổ màn hình xuất hiện như hình 6.7. Chọn thư mục chứa chương trình cần mở, chọn tên dự án và sau đó nhấp chuột vào thẻ **Open**.



Hình 6.7: Cửa sổ màn hình chứa dự án cần mở

6.4.4 Thư viện

Thư viện (Libraries) được sử dụng để lưu trữ các khối chương trình con có truyền tham số được sử dụng để lập trình. Các khối có thể copy vào trong

một thư viện từ một dự án có sẵn hoặc chúng có thể được tạo ra trực tiếp trong thư viện độc lập với các dự án.

Khi cài đặt STEP 7-Micro/WIN thì các khối chưa được cài đặt vào trong thư viện. Để cài đặt thư viện chuẩn có thể download thư viện S7-200 từ trang www.siemens.com hoặc sử dụng đĩa phần mềm **STEP 7--Micro/WIN Add-on: STEP 7--Micro/WIN 32 Instruction Library, V1.1 (CD-ROM)**.

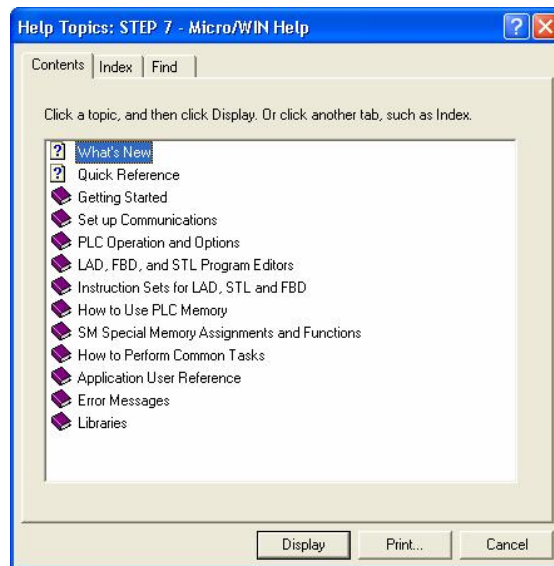
Có thể chèn thêm hoặc xóa bỏ bớt các khối chương trình trong thư viện sử dụng **File > Add/Remove Libraries** và sau đó chọn thẻ Add để chọn khối chương trình thư viện mong muốn đưa vào thư viện.

Để mở thư viện, vào **Cây Lệnh** chọn mục **Libraries**, chọn các khối chương trình cần sử dụng. Việc tạo thêm các khối chương trình con truyền tham số được sử dụng để làm thư viện có thể được tạo ra từ **File > Create Library** và chọn chương trình con cần làm thư viện.

6.4.5 Hệ thống trợ giúp trong STEP 7-Micro/WIN

Trường hợp gặp khó khăn trong lập trình cũng như cần tìm hiểu rõ hơn về một thông tin nào đó trong phần mềm ta có thể sử dụng công cụ trợ giúp. Có nhiều cách khác nhau để mở trợ giúp:

1. Sử dụng menu **Help > Contents and Index** để kích hoạt trợ giúp chung.
2. Sử dụng phím F1 để trợ giúp theo ngữ cảnh với đối tượng được chọn.



Hình 6.8: Màn hình trợ giúp

- **Thẻ Content:** Hiển thị danh sách các chủ đề trợ giúp

- **Thẻ Index:** Cho phép truy cập thông tin trợ giúp bằng việc hiển thị danh sách các thuật ngữ theo thứ tự alphabe.
- **Thẻ Find:** Cho phép tìm kiếm các từ cụ thể và thuật ngữ trong chủ đề trợ giúp.

Khi nhấp chuột vào các từ được nổi lên có màu xanh và gạch chân (hotwords) sẽ xuất hiện các trợ giúp chi tiết hơn.


6.4.6 Xóa bộ nhớ CPU

Khi xóa PLC thì PLC phải đặt ở chế độ STOP và reset PLC theo chuẩn nhà máy, ngoại trừ địa chỉ PLC, tốc độ truyền, và đồng hồ thời gian (time-of-date clock). Để xóa chương trình trong PLC thực hiện như sau:


1. Chọn **PLC > Clear...** thì hộp thoại Clear xuất hiện
2. Chọn tất cả các mục chấp nhận bằng cách nhấp OK.
3. Nếu đã có password trong bộ nhớ PLC thì hộp thoại yêu cầu password xuất hiện. Để xóa password thì nhập **CLEARPLC** vào hộp thoại và tiếp tục hoạt động xóa tất cả.

6.4.7 Mở một dự án đang tồn tại sẵn

Mở một dự án tồn tại (tập tin có phần mở rộng .mwp) hay thành phần của dự án và bắt đầu một phần soạn thảo mới bằng cách sử dụng các phương pháp sau:

1. Nhấp chuột vào biểu tượng Open Project .
2. Chọn menu lệnh **File > Open**.
3. Ấn tổ hợp phím Ctrl+O
4. Mở Windows Explorer và nhấp đúp chuột vào tập tin có phần mở rộng .mwp.
5. Mở một thành phần dự án bằng cách nhấp chuột phải vào các ghi chú trong cây lệnh (Instruction Tree). Chọn Open để mở.

Để mở các dự án được tạo với các phiên bản trước của STEP 7-

Micro/WIN hay STEP 7-Micro/DOS thì nhấp chuột vào Open  hay chọn File>Open và chọn tập tin mong muốn.

Chú ý:

- Dự án đã tạo bằng các phiên bản trước của STEP 7-Micro/WIN hay STEP 7-Micro/DOS có thể chứa một hay nhiều cấu trúc logic mà STEP 7-Micro/WIN, Version 3.0 và cao hơn không hỗ trợ. Để mở được dự án, ta phải sử dụng phiên bản cũ đã tạo dự án và lưu lại dự án theo thủ tục sau:

1. Chuyển màn hình soạn thảo sang STL.

2. Tắt địa chỉ theo ký hiệu.
3. Lưu tập tin dự án.

- Chương trình đã tạo với STEP 7-Micro/WIN V3.1 SP1 sử dụng lệnh AND có ngõ vào đơn ở FBD, và được lưu để xem ở FBD, thì không thể mở được với STEP 7-Micro/WIN V3.1. Để mở các dự án này với STEP 7-Micro/WIN V3.1, dự án trước tiên nên được chuyển sang để xem ở STL và lưu lại ở dạng này.

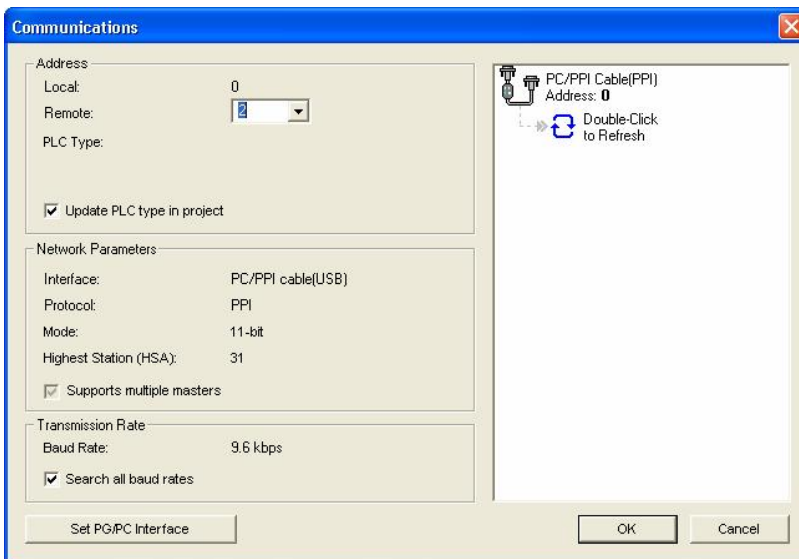
- Không thể sử dụng lệnh Open để mở một dự án trong PLC; Các tập tin dự án chỉ có thể mở được nếu nó được lưu trữ trên PC hoặc PG (thiết bị lập trình)

- Với phần mềm STEP-7 Micro/WIN mỗi lần mở chỉ được một dự án. Vì vậy muốn mở 2 dự án tại cùng một thời điểm thì phải chạy hai lần STEP-7 Micro/WIN. Khi mở hai dự án, ta có thể copy các phần tử chương trình lẫn nhau.

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




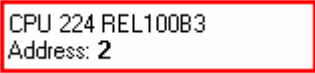
Để kết nối truyền thông S7-200 với thiết bị lập trình thì cần phải có cáp kết nối (xem chương 4). Việc kết nối truyền thông thực hiện theo các bước sau:

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



Hình 6.9: Màn hình thiết lập truyền thông

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

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.
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 được hiển thị (ví dụ biểu tượng  CPU 224 REL100B3 Address: 2). 
5. Chọn S7-200 và nhấp OK. Nếu STEP 7--Micro/WIN không tìm ra CPU S7-200, kiểm tra việc đặt chỉnh các tham số truyền thông và lặp lại bước này.
6. Sau khi đã thiết lập truyền thông với S7-200, ta có thể sẵn sàng tạo và download chương trình vào CPU.

6.4.9 Tải dự án từ PLC

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-Micro/WIN. Cần lưu ý là PLC đã được kết nối truyền thông với thiết bị lập trình.

6.4.9.1 Tải một khối hoặc ba khối

Có thể tải khối chương trình (OB1, chương trình con, chương trình ngắt), System Block, và Data Block hay chọn lựa một trong ba khối này từ PLC về máy tính. Chương trình trong PLC không chứa các địa chỉ ký hiệu hay thông tin status chart. Do đó, ta không thể tải một bảng Symbol Table hay Status Chart.

6.4.9.2 Tải vào một dự án mới hoặc dự án rỗng




Để tải chương trình về máy tính thì một cách không làm ảnh hưởng đến các chương trình đang mở là đóng nó lại và tạo một dự án mới, vì dự án mới là rỗng nên không thể vô tình phá hủy dữ liệu. Đây là cách thức an toàn để lấy khối chương trình, system block hoặc thông tin data block. Nếu muốn lấy sử dụng bảng ký hiệu (symbol table) hoặc status chart đã được tạo cho dự án này, thì có thể mở dự án cũ ở màn hình STEP 7-Micro/WIN khác và copy các thông tin này vào dự án được upload về.

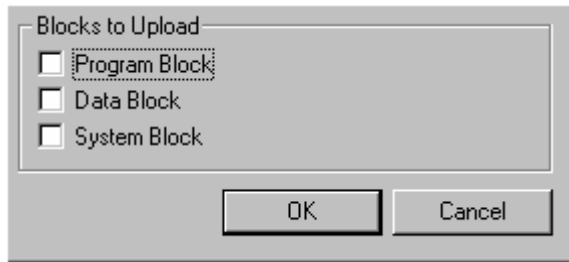
6.4.9.3 Tải vào một dự án tồn tại

Đây là một cách để viết đè tất cả các phần của chương trình hiện hành bằng chương trình đã được nạp vào PLC trước đó.

6.4.9.4 Thủ tục tải dự án từ PLC về thiết bị lập trình

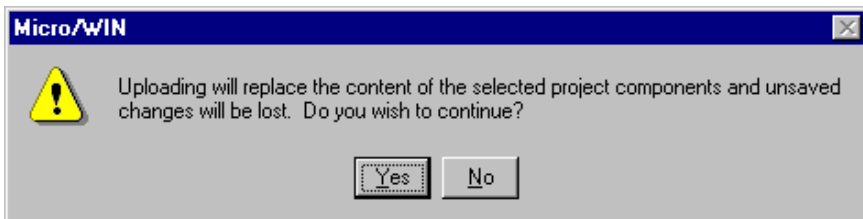
Để thực hiện tải, thực hiện các bước sau:

- Trong STEP 7-Micro/WIN mở một dự án để giữ các khối sẽ được upload từ PLC.
 - Nếu muốn upload vào một dự án rỗng, chọn **File > New** hoặc sử dụng biểu tượng New Project  trên toolbar.
 - Nếu muốn upload vào một dự án tồn tại, chọn **File > Open** hoặc sử dụng biểu tượng Open Project  trên toolbar.
- Chọn **File > Upload** hoặc sử dụng biểu tượng Upload  trên toolbar để khởi động quá trình upload.
- 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 nhấn OK.



Hình 6.10: Hộp thoại Upload

- STEP 7-Micro/WIN hiển thị chú ý sau:




Hình 6.11: Chú ý khi upload từ PLC về thiết bị lập trình

Nhấn **Yes** để chấp nhận việc upload.


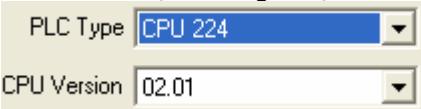



STEP 7-Micro/WIN hiển thị một thông báo khi upload các khối thành công từ PLC về thiết bị lập trình hoặc máy tính PC.

6.4.10 Nạp (download) một dự án 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:

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 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.

2. Nhấp chuột vào biểu tượng download  trong toolbar hoặc chọn **File > Download**. Hộp Download xuất hiện.
3. Chọn các khối cần download. Thông thường là chọn tất cả.
4. Nhấp OK để bắt đầu quá trình download.
5. Nếu download thành công, thì một hộp thoại hiển thị thông báo: *Download Successful*. Tiếp tục đến bước 12.
6. Nếu loại PLC được chọn cho chương trình trong STEP 7/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?".
7. Đặt lại loại PLC cho phù hợp, chọn **No** để dừng tiến trình download.
8. Chọn **PLC > Type...** để vào hộp thoại chọn loại PLC.
9. Có thể chọn đúng loại PLC theo danh sách trong mục
 của hộp thoại. Hoặc nhấp chuột vào thẻ  để STEP 7-Micro/WIN tự động tìm đúng loại PLC đang kết nối.
10. Nhấp **OK** để chấp nhận loại PLC và đóng hộp thoại.
11. Khởi động lại quá trình download bằng cách nhấp chuột vào biểu tượng download  trong toolbar hay chọn **File > Download**.
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.

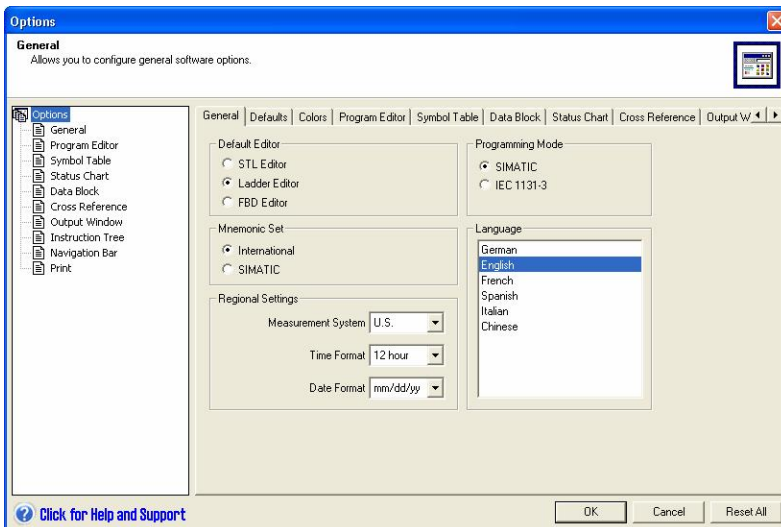
Trường hợp sử dụng công tắc thì chuyển từ vị trí STOP sang RUN.

6.4.11 Thiết lập cấu hình chung cho phần mềm (menu option và customize)

6.4.11.1 Menu Option

Có thể định nghĩa một đường dẫn mặc định đến một thư mục tập tin xác định để mở và lưu các dự án STEP 7-Micro/WIN. Ta sử dụng menu lệnh **Tools > Options**.

Ngoài ra, để truy cập trực tiếp Option cho từng thành phần trong cây lệnh (Instruction tree) thì trở chuột vào thành phần mong muốn và nhấp chuột phải, sau đó chọn mục **option**.



Hình 6.12: Cửa sổ Options

* General Options

- **Thẻ General:** Chọn thẻ này để lựa chọn Program Editor, Mnemonic Set, Programming Mode, Language, và Regional Settings(Measurement System, Time Format, and Date Format) mặc định.

- **Thẻ Defaults:** Chọn thẻ này để đặt vị trí tập tin và loại PLC mặc định cho các dự án mới. Ta cũng có thể chọn để thêm System Symbol Table cho tất cả các dự án mới.

- **Thẻ Colors:** Chọn thẻ này để gán Font và Color cho các cửa sổ khác nhau.

* Program Editor Options

- *Thẻ Program Editor*: Chọn thẻ này để định kích thước, hiển thị và font của cửa sổ soạn thảo chương trình. Chọn trạng thái hiển thị bên trong hay bên ngoài lệnh. Cấu hình địa chỉ theo ký hiệu. Ta cũng có thể chọn để cho phép soạn thảo toán tử sau khi đặt một lệnh và định dạng tự động bất kỳ mã lệnh STL được nhập vào.

- *Thẻ STL Status*: Chọn thẻ này để tùy biến cách thức mà Program Status được trình diễn ở STL. Ta có thể thay đổi các đặt chỉnh sau: Watch Values, Operands, Logic Stack, Instruction Status Bits.

* **Other Options**

- *Thẻ Symbol Table*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước của bảng ký hiệu (symbol table). Ta có thể chọn để hiển thị các ký hiệu trùng nhau, không được sử dụng.

- *Thẻ Status Chart*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước của status chart. Cũng có thể thiết lập việc định địa chỉ theo ký hiệu.

- *Thẻ Data Block*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước và độ rộng của data block.

- *Thẻ Cross Reference*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước của bảng cross reference. Cũng có thể thiết lập việc định địa chỉ theo ký hiệu.

- *Thẻ Output Window*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước của output window.

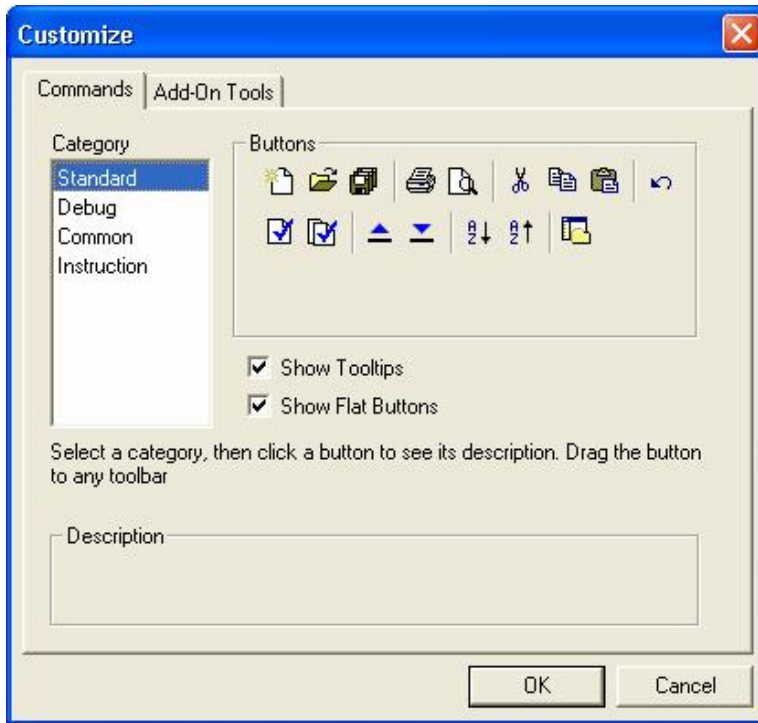
- *Thẻ Instruction Tree*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước của Instruction Tree (cây lệnh). Ta cũng có thể chọn để cho phép tự động xếp lại của instruction tree.

- *Thẻ Navigation Bar*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước của navigation bar.

- *Thẻ Print*: Chọn thẻ này để thiết lập kiểu font, kiểu dáng và kích thước của các dự án muốn in.

6.4.11.2 Menu Custommize

Menu custommize cho phép ta thay đổi sự xuất hiện nội dung trong toolbar và thêm vào các công cụ được sử dụng thường xuyên vào menu Tools.



Hình 6.13: Cửa sổ customize.

Chọn menu lệnh **Tools > Customize** để thiết lập các lựa chọn sau:

- Thẻ *Commands*: Cho phép thay đổi sự xuất hiện các nội dung của toolbars.
- Thẻ *Add-On Tools*: Cho phép thêm vào các công cụ được sử dụng thường xuyên vào menu Tools.

*** Thay đổi sự xuất hiện:**

- Chọn **Show Tooltips** nếu muốn các nút nhấn hiển thị các thông tin về nó khi con trỏ chuột dừng trên nút nhấn.
- Chọn **Show Flat Buttons** nếu muốn các nút nhấn xuất hiện ở dạng phẳng thay vì xuất hiện ở dạng 3-D.

*** Di chuyển một nút nhấn:**

- Chọn một toolbar từ hộp danh sách Category để hiển thị các nút nhấn của toolbar đó. Để di chuyển một nút nhấn từ toolbar mặc định sang toolbar khác, thì chọn tên của toolbar chứa nút nhấn cần di chuyển từ hộp danh sách Category. Kéo nút nhấn mong muốn trong vùng nút nhấn ra vùng toolbar để thêm nó vào toolbar.

- Để loại bỏ một nút nhấn trên toolbar, kéo nút nhấn trên toolbar và bỏ vào vùng nút nhấn của hộp thoại Customize.


* *Thẻ Add-On Tools*: Thêm một công cụ vào menu Tools.

Đặc điểm này được dự định để tiết kiệm thời gian đối với các công cụ được sử dụng thường xuyên. Để thêm một công cụ, nhấp vào thẻ Add-On Tools,

nhấp vào nút , và điền vào các vùng ở dưới:

Bất kỳ lệnh được yêu cầu được bắt đầu và kết thúc bởi dấu ngoặc kép khi nhập vào vùng command (ví dụ: "xxx xxx").

- *Menu Text*: Chọn một tên để nhận dạng công cụ trên menu Tools.
- *Command*: Cung cấp tên tập tin của chương trình công cụ hay bat. file.
- *Arguments*: Cung cấp các chủ đề dòng lệnh đã sử dụng bởi tập tin *.exe.
- *Initial Directory*: Cung cấp đường dẫn thư mục đang mở cho công cụ.

Sử dụng nút  để tìm các tập tin và thư mục.

Khi thêm vào một công cụ thành công, trong menu Tools xuất hiện công cụ đã thêm.

6.4.12 Soạn thảo chương trình

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

- 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.

Để hiểu được phần mềm STEP 7-Micro/WIN dễ dàng, chúng ta nên viết một ví dụ đơn giản được cho ở hình 6.14 và bảng thiết lập vào/ra cho ở bảng 6.1. Do mới bắt đầu, ta nên viết chương trình ở dạng LAD, rồi sau đó có thể xem ở dạng FBD hay STL.

LAD	STL
<p>Điều khiển đảo chiều quay động cơ</p> <p>Network 1 Động cơ quay phải</p> <p>An nút nhấn nối với ngo vào I0.1 --> Động cơ quay phải. An nút nhấn (thường đóng) nối với ngo vào I0.0 --> Động cơ dừng</p> <p>Network 2 Động cơ quay trái</p> <p>An nút nhấn nối với ngo vào I0.2 --> Động cơ quay trái. An nút nhấn (thường đóng) nối với ngo vào I0.0 --> Động cơ dừng</p>	<p>Điều khiển đảo chiều quay động cơ</p> <p>Network 1 Động cơ quay phải</p> <pre>LD I0.1 O Q0.0 A I0.0 AN Q0.1 = Q0.0</pre> <p>Network 2 Động cơ quay trái</p> <pre>LD I0.2 O Q0.1 A I0.0 AN Q0.0 = Q0.1</pre>

Hình 6.14: Ví dụ để soạn thảo một chương trình mới

Ký hiệu	Địa chỉ	Chú thích
S_Stop	I0.0	Nút nhấn dừng động cơ, thường đóng (NC)
S_Right	I0.1	Nút nhấn động cơ quay phải, thường hở (NO)
S_Left	I0.2	Nút nhấn động cơ quay trái, thường hở (NO)
K1	Q0.0	Contactơ cấp điện để động cơ quay phải
K2	Q0.1	Contactơ cấp điện để động cơ quay trái

Bảng 6.1: Bảng xác định kết nối dây vào/ra với ngoại vi

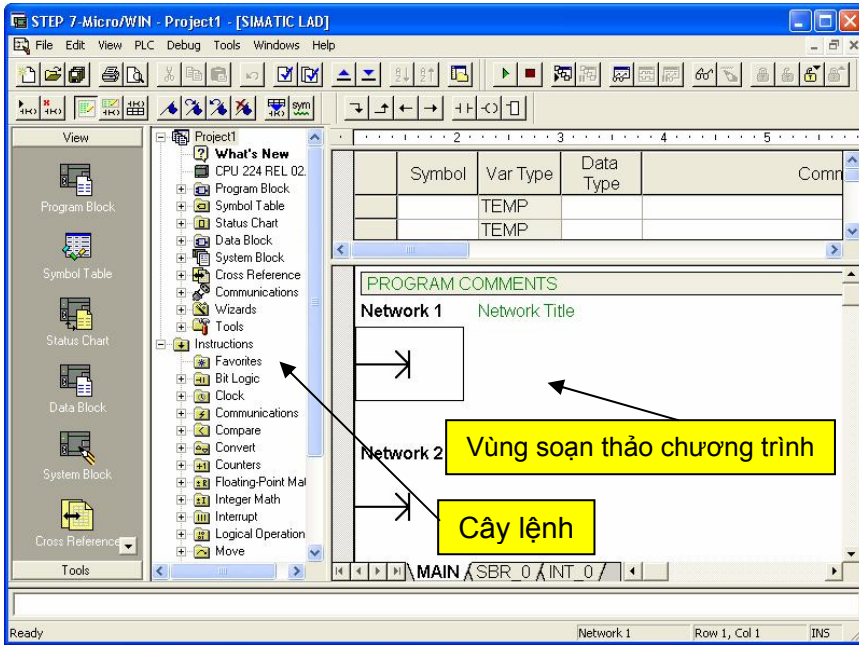
Các bước thực hiện:

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 6.15). 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.



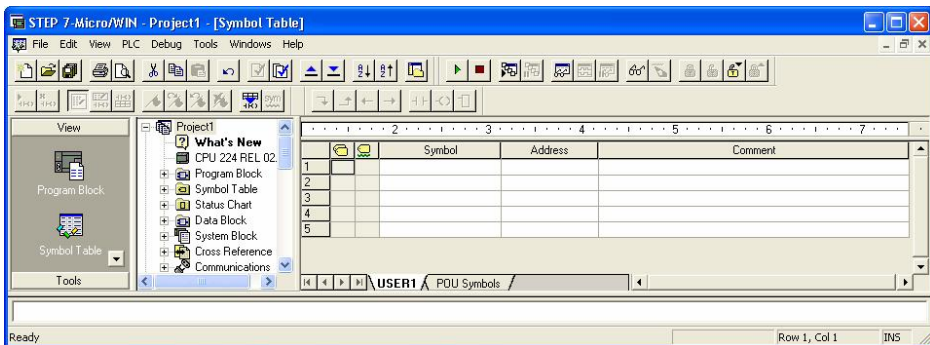
Hình 6.16: Màn hình soạn thảo chương trình

Bước 2: Nhập bảng ký hiệu

Nhấp chuột vào biểu tượng Symbol Table  để mở màn hình soạn thảo bảng ký hiệu (hình 6.17).

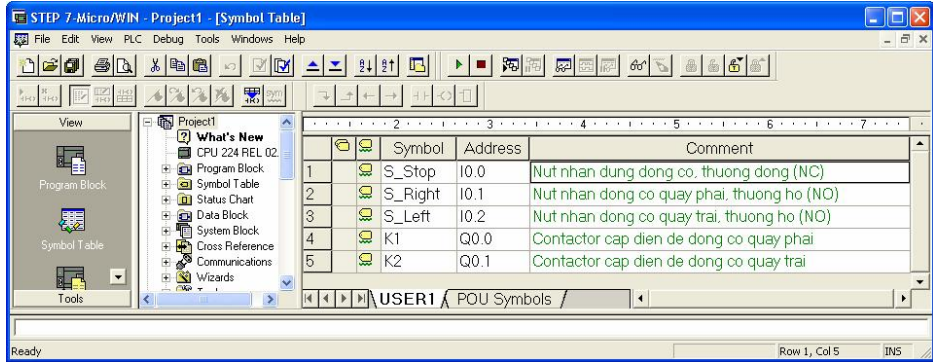
Nhập các thông tin (chữ không dấu) ở bảng 6.1 vào bảng Symbol Table. Với:

- *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 6.17: Màn hình soạn thảo bảng ký hiệu


Sau khi nhập xong, ta có bảng ký hiệu như hình 6.18.



Hình 6.18: 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 6.15).


• **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). Contactơ 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 6.14.


Nhập các tiếp điểm như sau:

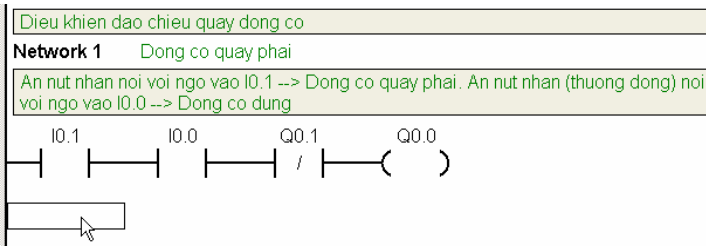
1. Nhấp đúp chuột vào hình tượng Bit Logic  hoặc nhấp chuột vào dấu cộng (+) ở cửa sổ cây lệnh để hiển thị các lệnh trong bit logic.
2. Chọn tiếp điểm Normally Open .
3. Giữ chuột trái và kéo tiếp điểm vào network đầu tiên.
4. Nhấp chuột vào “???” trên tiếp điểm và nhập vào địa chỉ: I0.1 và sau đó nhấn 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 Output  và nhập vào ở “???” địa chỉ Q0.0


Chú ý: khi gỡ 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**.

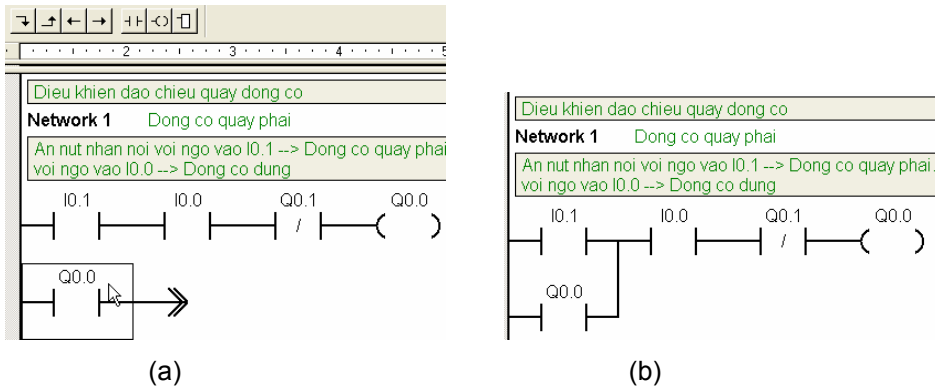
Rẽ nhánh Network 1.

1. Tương chọn tiếp điểm Normally Open  giữ chuột trái và kéo tiếp điểm vào vị trí con trỏ chuột (hình 6.18) và đặt tên Q0.0.



Hình 6.18: Rẽ nhánh network

2. Để con trỏ chuột ở vị trí như hình 6.19a và nhấp chuột vào biểu tượng line up  để kết thúc (hình 6.19b).



Hình 6.19: Rẽ nhánh network

- Nhập network 2: Dong co quay trai
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 (CPU 221, CPU 222, CPU 224, 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** hoặc nhấp chuột vào

biểu tượng . Dựa vào trạng thái của các tiếp điểm và các cuộn dây trong chương trình mà có thể 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**.

7 Các phép toán logic

7.1 Ngăn xếp (logic stack) trong S7-200

Trong các CPU S7-200 có một ngăn xếp gồm 9 bit, chúng được sử dụng cho các câu lệnh mà dữ liệu là dạng bit. Khi viết chương trình dạng STL thì người lập trình cần hiểu rõ về phương thức hoạt động của các bit trong ngăn xếp. Ngăn xếp logic là một khối gồm 9 bit chồng lên nhau. Tất cả các thuật toán liên quan đến ngăn xếp đều chỉ làm việc với bit đầu tiên hoặc với bit đầu và bit thứ hai của ngăn xếp. Giá trị logic mới đều có thể được gửi (hoặc được nối thêm) vào ngăn xếp. Khi phối hợp hai bit đầu tiên của ngăn xếp, thì ngăn xếp sẽ được kéo lên một bit. Ngăn xếp và tên của từng bit trong ngăn xếp được biểu diễn dưới đây:

S0	Stack 0 – bit đầu tiên hay bit trên cùng của ngăn xếp.
S1	Stack 1 – bit thứ hai của ngăn xếp.
S2	Stack 2 – bit thứ ba của ngăn xếp.
S3	Stack 3 – bit thứ tư của ngăn xếp.
S4	Stack 4 – bit thứ năm của ngăn xếp.
S5	Stack 5 – bit thứ sáu của ngăn xếp.
S6	Stack 6 – bit thứ bảy của ngăn xếp.
S7	Stack 7 – bit thứ tám của ngăn xếp.
S8	Stack 8 – bit thứ chín của ngăn xếp.


Trong 9 Stack, thì Stack 0 là ngăn xếp quan trọng nhất. Giá trị logic của nó sẽ là kết quả của phép toán logic. Hay nói khác đi, sau một phép toán logic nhị phân thì kết quả của phép toán sẽ được lưu ở Stack 0. Nếu giá trị logic ở Stack 0 có giá trị là “0” thì kết quả thu được là “0”, tương tự nếu có giá trị là “1” thì kết quả thu được là “1”.

Ngoài ra giá trị logic “1” của Stack 0 còn là điều kiện bắt buộc cho việc thi hành đối với một số lệnh.

7.2 Các phép toán logic cơ bản

Trong phần này trình bày các phép toán đối với dữ liệu là bit. Trước tiên là phần 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).

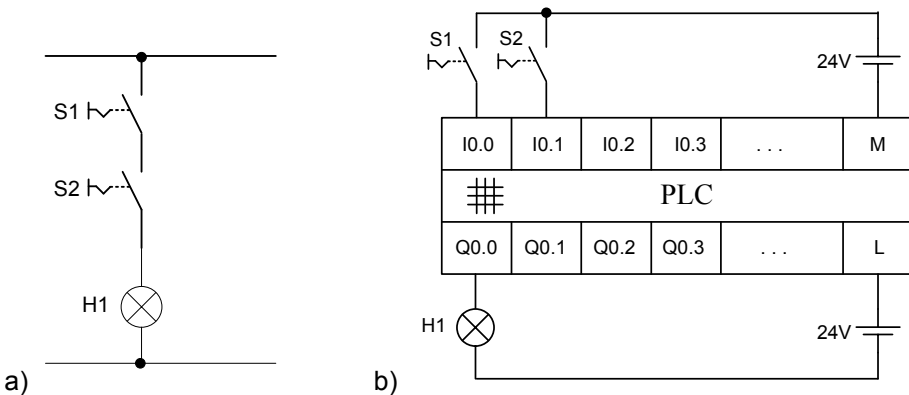
Vì phần soạn thảo chương trình đã được trình bày ở *chương 6*, nên trong phần này không trình bày lại. Bạn đọc có thể xem *mục 6.4.12 của chương 6* để thực hiện cho các ví dụ ở chương này và các chương tiếp theo.

Chương 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 ( Bit Logic) của cây lệnh.

7.2.1 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.

Ví dụ 7.1: 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 7.1 Liên kết AND: a) Sơ đồ mạch điện, b) Nối dây với ngõ vào/ra PLC

+ Lập bảng ký hiệu mô tả tên và địa chỉ của biến (soạn thảo bằng cách mở mục *Symbol Table* trong phần mềm soạn thảo):

		Symbol	Address	Comment
1		S1	I0.0	Cong tac
2		S2	I0.1	Cong tac
3		H1	Q0.0	Den bao

Hình 7.2 Bảng ký hiệu

+ Chương trình:

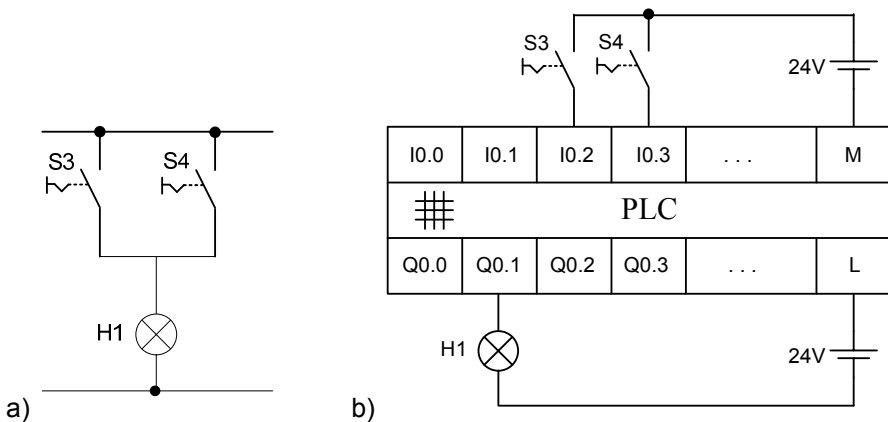
LAD	FBD	STL
		<pre>LD I0.0 A I0.1 = Q0.0</pre>

Hình 7.3 Chương trình được biểu diễn ở 3 dạng LAD, FBD và STL.

7.2.2 Phép toán OR

Phép toán OR sẽ được sử dụng khi trạng thái của một trong hai (hoặc nhiều) tín hiệu thỏa mãn điều kiện của yêu cầu điều khiển thì sẽ thực hiện một nhiệm vụ điều khiển nào đó.

Ví dụ 7.2: 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 7.4 Liên kết OR: a) Sơ đồ mạch điện, b) Nối dây với ngõ vào/ra PLC,

	Symbol	Address	Comment
1	S3	I0.2	Cong tac
2	S4	I0.3	Cong tac
3	H2	Q0.1	Den bao

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

Hình 7.5 Bảng ký hiệu và chương trình liên kết OR

7.2.3 Tổ hợp các cổng AND và OR

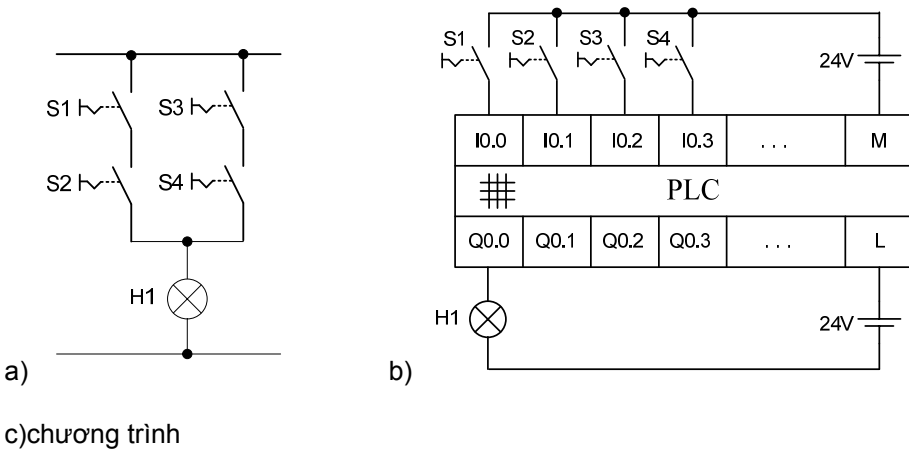
Trong thực tế, các đối tượng điều khiển phụ thuộc vào một tổ hợp các liên kết logic AND và OR. Tùy theo liên kết nào đứng trước mà sẽ có các lệnh ở STL khác nhau.

7.2.3.1 AND trước OR

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**.

Ví dụ 7.3:

		Symbol	Address	Comment
1		S1	I0.0	Cong tac
2		S2	I0.1	Cong tac
3		S3	I0.2	Cong tac
4		S4	I0.3	Cong tac
5		H1	Q0.0	Den bao



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

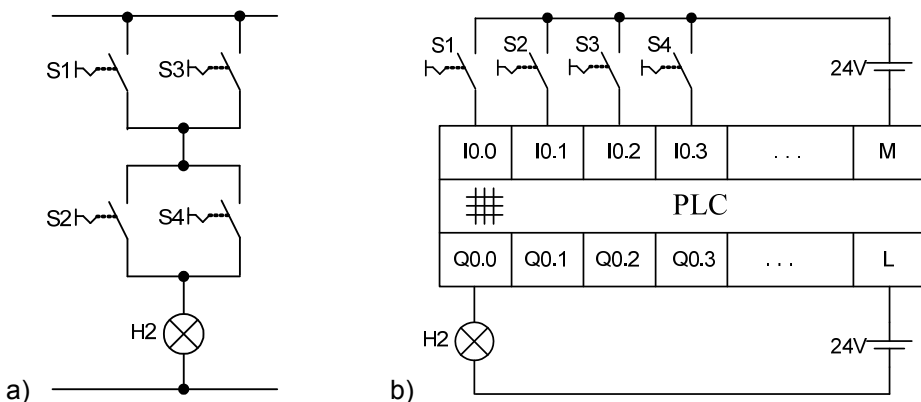
Hình 7.6 AND trước OR: a) Mạch điện, b) Nối dây với PLC, c) Chương trình

7.2.3.2 OR trước AND

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**.

Ví dụ 7.4:

		Symbol	Address	Comment
1		S1	I0.0	Cong tac
2		S2	I0.1	Cong tac
3		S3	I0.2	Cong tac
4		S4	I0.3	Cong tac
5		H2	Q0.0	Den bao



c) Chương trình

LAD	FBD	STL
		<pre> LD I0.0 OR I0.2 = OR1 LD I0.1 OR I0.3 = OR2 AND OR1 AND OR2 = Q0.0 </pre>

Hình 7.7 OR trước AND: a) Mạch điện, b) Nối dây với PLC, c) Chương trình

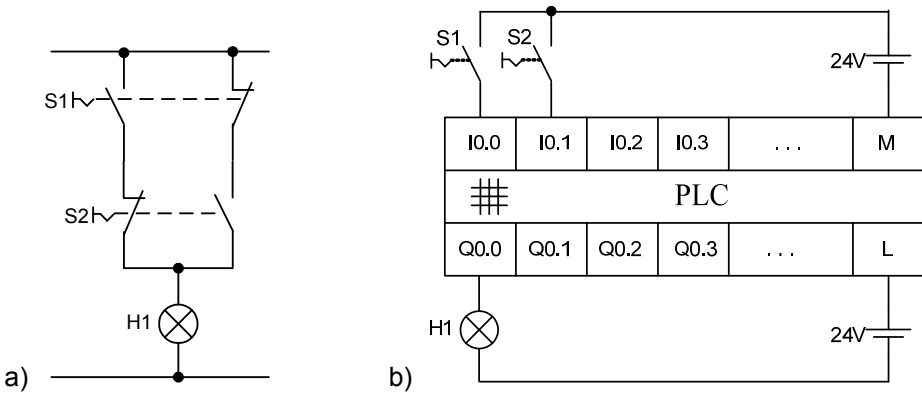
7.2.4 Phép toán XOR

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.

Ví dụ 7.5: Ở sơ đồ hình 7.8a, mỗi một nút nhấn được gắn 2 tiếp điểm (1NO và 1NC), khi tác động nút nhấn thì cả 2 tiếp điểm này tác động theo. Đèn sáng nếu tác động chỉ một trong hai công tắc S1 hoặc S2.

Bảng ký hiệu

		Symbol	Address	Comment
1		S1	I0.0	Cong tac
2		S2	I0.1	Cong tac
3		H1	Q0.0	Den bao



Hình 7.8 Liên kết XOR a) Sơ đồ mạch điện, b) Kết nối với PLC

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

Hình 7.9 Chương trình liên kết XOR

7.3 Xử lý các tiếp điểm, cảm biến được nối với ngõ vào PLC

Một vấn đề quan trọng đối với người mới làm quen với chương trình PLC là việc xác định đúng trạng thái các loại tiếp điểm được viết ở LAD. Đặc biệt là các tiếp điểm ngõ vào.

Các cảm biến, công tắc hoặc nút nhấn thường có hai dạng là *thường đóng (NC)*, hoặc *thường hở (NO)*. Vì các ngõ vào số được nối với các đối tượng này nên các tiếp điểm trong chương trình, tùy theo trường hợp, cũng sẽ có dạng tương ứng. Tuy nhiên, để dễ dàng phân biệt ta không nên gọi các tiếp điểm trong chương trình là thường đóng hoặc thường mở. Qui ước đặt tên cho các tiếp điểm trong chương trình như sau:











- Tiếp điểm : Được gọi là *tiếp điểm không đảo trạng thái tín hiệu*
- Tiếp điểm : Được gọi là *tiếp điểm đảo trạng thái tín hiệu*.

Để rõ hơn trạng thái các tiếp điểm được nối với ngõ vào số và kết quả xử lý chương trình trong PLC, ta xem bảng 7.1.

Từ bảng này, ta có một số nhận xét như sau:

1. Ngõ vào có logic "1" khi ngõ vào có điện áp.

- Nếu ngõ vào được nối với tiếp điểm thường đóng (NC), thì ngõ vào ở trạng thái bình thường luôn có điện (đèn LED báo ngõ vào tương ứng sáng). Nó chỉ bị mất điện nếu tiếp điểm NC bị tác động.
- Nếu ngõ vào được nối với tiếp điểm thường hở (NO), thì ngõ vào ở trạng thái bình thường không có điện (đèn LED báo ngõ vào tương ứng tắt). Nó chỉ có điện khi tác động tiếp điểm NO.
- Nếu sử dụng tiếp điểm không đảo trạng thái tín hiệu $\begin{matrix} | \\ | \\ | \end{matrix}$, thì kết quả xử lý trong chương trình có cùng trạng thái logic với ngõ vào.
- Nếu sử dụng tiếp điểm đảo trạng thái tín hiệu $\begin{matrix} | \\ | \\ | \end{matrix}$, thì kết quả xử lý trong chương trình có trạng thái logic ngược với ngõ vào.
- Không được thay tùy tiện tiếp điểm thường hở (NO) bằng tiếp điểm $\begin{matrix} | \\ | \\ | \end{matrix}$ trong chương trình, cũng như tiếp điểm thường đóng (NC) bằng tiếp điểm $\begin{matrix} | \\ | \\ | \end{matrix}$. Mà phải chú ý đến yêu cầu công nghệ đặt ra.

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

Bảng 7.1 Trạng thái các tiếp điểm và xử lý trong chương trình PLC

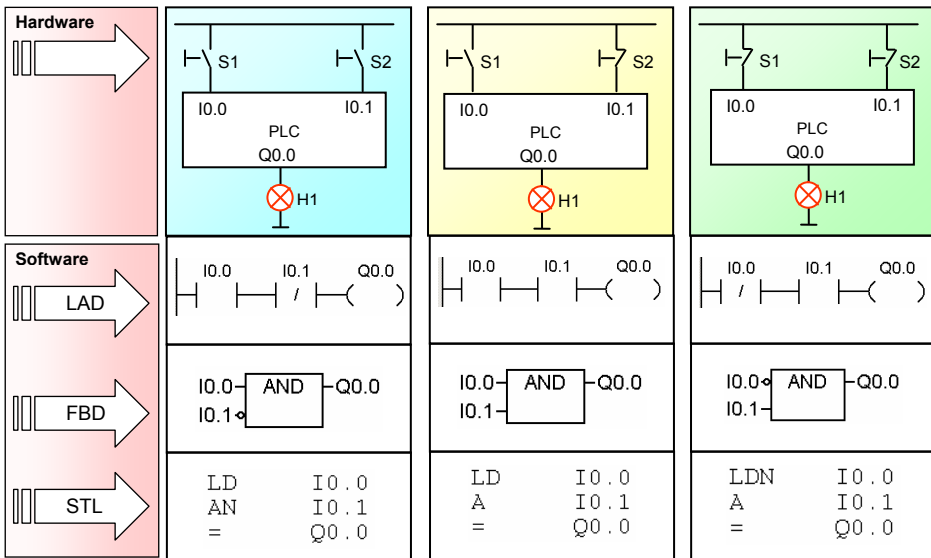
Ví dụ sau đây sẽ làm sáng tỏ hơn về việc xử lý các tiếp điểm nối với ngõ vào.

Ví dụ 7.6: Trong 3 mạch dưới đây (hình 7.10), đèn H1 sẽ sáng khi ấn nút nhấn S1 và không ấn nút nhấn S2.

Từ ví dụ ta nhận thấy dù ngõ vào được nối với loại nút nhấn nào cũng vẫn có thể lập chương trình để thỏa mãn được yêu cầu đặt ra. Tuy nhiên việc sử dụng các tiếp điểm thường mở hoặc thường đóng trong quá trình điều khiển phụ thuộc vào các qui tắc an toàn.

Các tiếp điểm thường đóng luôn luôn được sử dụng cho công tắc hành trình và công tắc an toàn, để không chế sự nguy hiểm nếu dây điện bị đứt trong mạch điện cảm biến.

Các tiếp điểm thường đóng cũng được dùng để tắt máy vì lý do tương tự như trên.



Hình 7.10: Ví dụ xử lý các loại tiếp điểm.

7.4 Ví dụ ứng dụng các liên kết logic

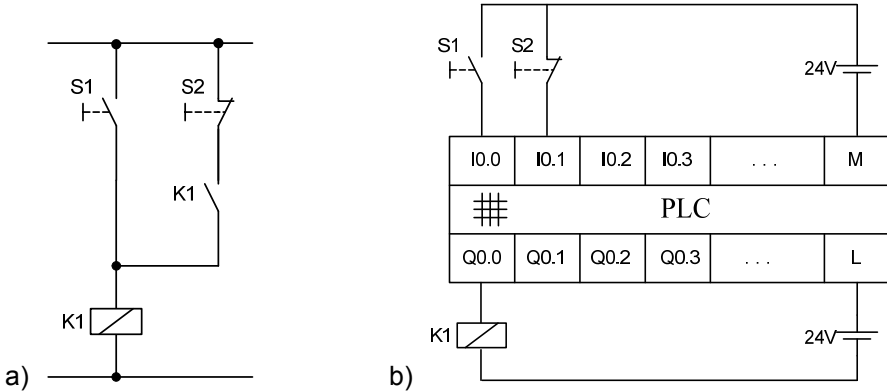
Phần này trình bày một số ví dụ ứng dụng nhỏ sử dụng các liên kết logic. Ở một số ví dụ có trình bày mạch điều khiển thông thường với kiểu nối dây khi không dùng PLC để chúng ta thấy sự giống nhau và khác nhau giữa 2 kiểu điều khiển.

7.4.1 Mạch tự duy trì ưu tiên mở máy

Mạch điều khiển dùng contactor có chức năng nhớ là mạch tự duy trì.

Trong trường hợp nếu cả hai nút nhấn mở máy S1 và dừng S2 cùng tác động mà contactor có điện thì là mạch tự duy trì ưu tiên mở máy.

Bảng ký hiệu		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn mở máy, thường hở (NO)
S2	I0.1	Nút nhấn dừng máy, thường đóng (NC)
K1	Q0.0	Contactor



Hình 7.11 Mạch ưu tiên mở máy: a) mạch điều khiển, b) nối dây PLC

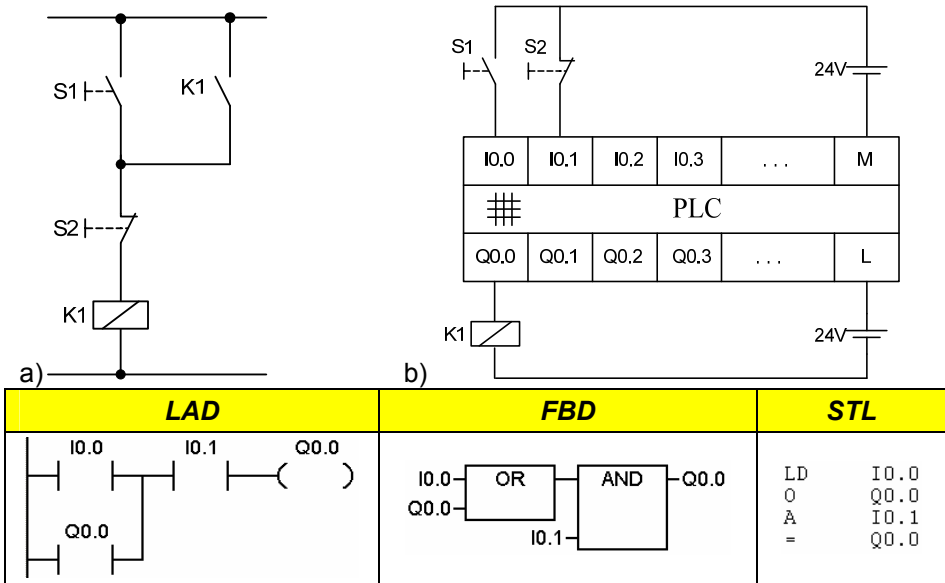
LAD	FBD	STL
		<pre>LD I0.0 LD I0.1 A Q0.0 O Q0.0</pre>

Hình 7.12 Chương trình mạch tự duy trì ưu tiên mở máy:

7.4.2 Mạch tự duy trì ưu tiên dừng máy

Trong trường hợp nếu cả hai nút nhấn mở máy S1 và dừng S2 cùng tác động mà contactor không có điện thì là mạch tự duy trì ưu tiên dừng máy.

Bảng ký hiệu		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn mở máy, thường hở (NO)
S2	I0.1	Nút nhấn dừng máy, thường đóng (NC)
K1	Q0.0	Contactor



Hình 7.13 Mạch ưu tiên dừng máy:
 a) mạch điều khiển, b) nối dây PLC và chương trình

7.4.3 Điều khiển ON/OFF động cơ có chỉ báo

Một động cơ điện 3 pha được điều khiển bằng một PLC S7-200. Khi nhấn nút S2 (thường hở) thì động cơ sẽ chạy. Khi nhấn nút S1 (thường đóng) thì động cơ sẽ dừng lại. Các chế độ hoạt động chạy và dừng được báo bằng 2 đèn báo H1 và H2.

Các thiết bị động lực gồm có:

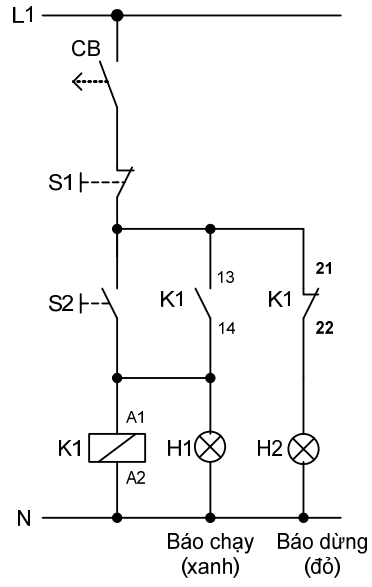
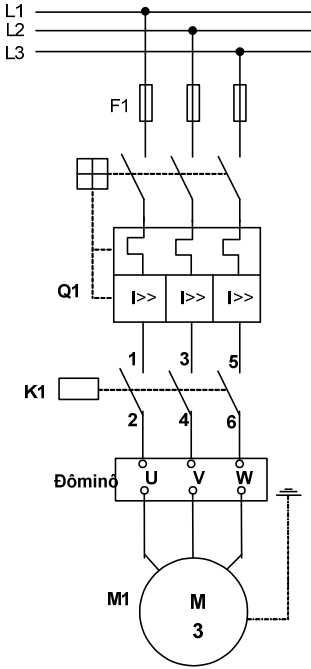
- Cầu chì 3 pha F1
- CB bảo vệ động cơ (Motor CB) Q1
- Contactor K1

Khi điều khiển dừng PLC thì mạch động lực vẫn giữ nguyên. Phần mạch điều khiển được biến đổi thành chương trình. Cần chú ý rằng các thiết bị điện như nút nhấn, CB, đèn báo đều giữ nguyên không thay đổi.

Nếu ta sử dụng PLC S7-200 loại DC/DC/DC thì ngõ ra của PLC cần phải kết nối với một relay trung gian K11 sử dụng nguồn 24Vdc. Relay này được dùng để đóng điện cho cuộn dây contactor K1 (hình 7.15). Riêng các đèn báo ta có thể thay thế bằng loại 24Vdc nhằm tiết kiệm relay trung gian.

Chú ý: Cũng có thể sử dụng loại CPU DC/DC/RLY, thì ngõ ra của nó có thể kết nối trực tiếp với cuộn dây K1. (xem thêm chương 5 về nối dây PLC với ngoại vi).

Bảng ký hiệu		
Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn dừng máy, thường đóng (NC)
S2	I0.1	Nút nhấn mở máy, thường hở (NO)
K11	Q0.0	Relay trung gian
H1	Q0.1	Đèn báo động cơ hoạt động
H2	Q0.2	Đèn báo động cơ dừng

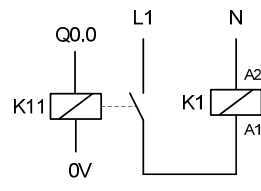
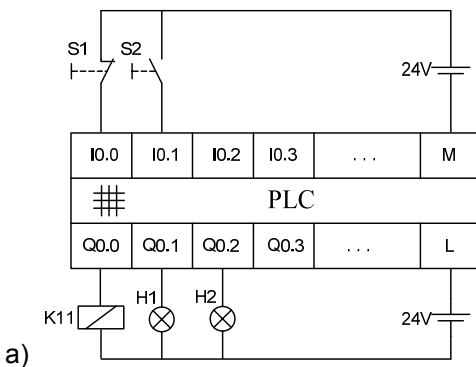


a)

b)

Hình 7.14 Mạch ON/OFF động cơ dùng contactor.

a) Mạch động lực; b) Mạch điều khiển



a)

b)

Hình 7.15: a) Sơ đồ nối dây PLC

b) Nối relay trung gian với contactor

+ Chương trình:

Biểu diễn ở STL:

Network 1 ON/OFF động cơ

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

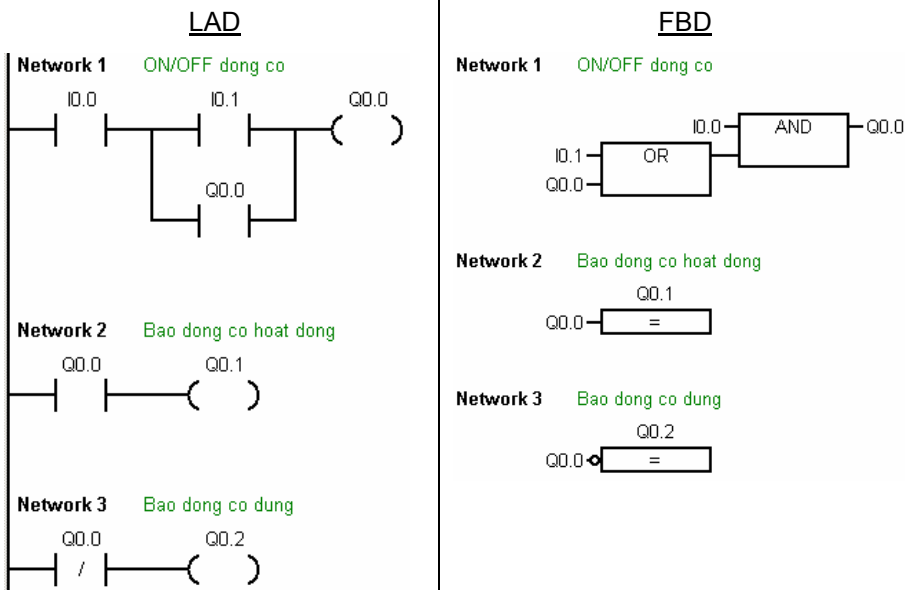
Network 2 Bảo động cơ hoạt động

```
LD Q0.0
= Q0.1
```

Network 3 Bảo động cơ dừng

```
LDN Q0.0
= Q0.2
```

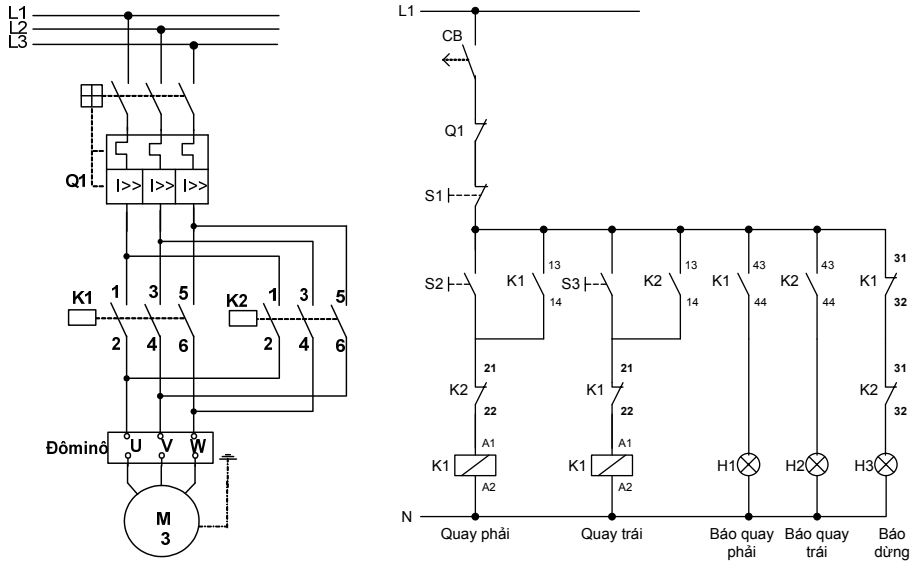
Biểu diễn ở LAD và FBD:



7.4.4 Điều khiển đảo chiều quay động cơ

Một động cơ điện 3 pha cần được điều khiển đảo chiều. Khi ấn S1 (thường hở) thì động cơ sẽ quay phải và đèn H1 sáng báo động cơ đang quay phải. Khi nhấn nút S2 (thường hở) thì động cơ quay trái và đèn H2 sáng báo động cơ đang quay trái. Động cơ có thể dừng bất cứ lúc nào nếu ấn nút dừng S3 (thường đóng) hoặc động cơ xảy ra sự cố quá dòng làm cho tiếp điểm của thiết bị bảo vệ Q1 tác động (tiếp điểm 13, 14 của Motor CB). Khi động cơ dừng đèn báo H3 sáng.

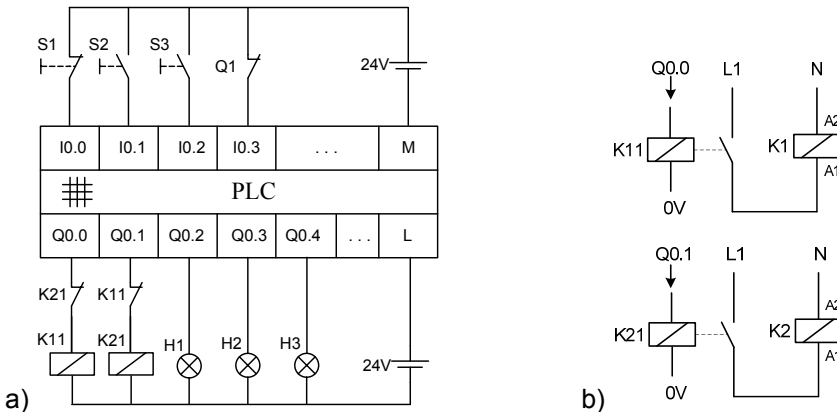
Tương tự như mục 7.4.3, ta sử dụng PLC S7-200 loại DC/DC/DC, ngõ ra của PLC điều khiển quay phải kết nối với relay trung gian K11, ngõ ra của PLC điều khiển quay trái kết nối với relay trung gian K21 sử dụng nguồn 24Vdc. Các relay này được dùng để đóng điện cho cuộn dây contactor K1 và K2 (hình 7.17). Riêng các đèn báo ta có thể thay thế bằng loại 24Vdc nhằm tiết kiệm relay trung gian.



Hình 7.16 Mạch động lực và điều khiển đảo chiều quay động cơ dùng contactor

Bảng ký hiệu

Ký hiệu	Địa chỉ	Chú thích
S1	I0.0	Nút nhấn dừng máy, thường đóng (NC)
S2	I0.1	Nút nhấn quay phải, thường hở (NO)
S3	I0.2	Nút nhấn quay trái, thường hở (NO)
Q1	I0.3	Tiếp điểm bảo quá dòng, thường đóng (NC)
K11	Q0.0	Relay trung gian điều khiển quay phải
K21	Q0.1	Relay trung gian điều khiển quay trái
H1	Q0.2	Đèn báo động cơ quay phải
H2	Q0.3	Đèn báo động cơ quay trái
H3	Q0.4	Đèn báo động cơ dừng

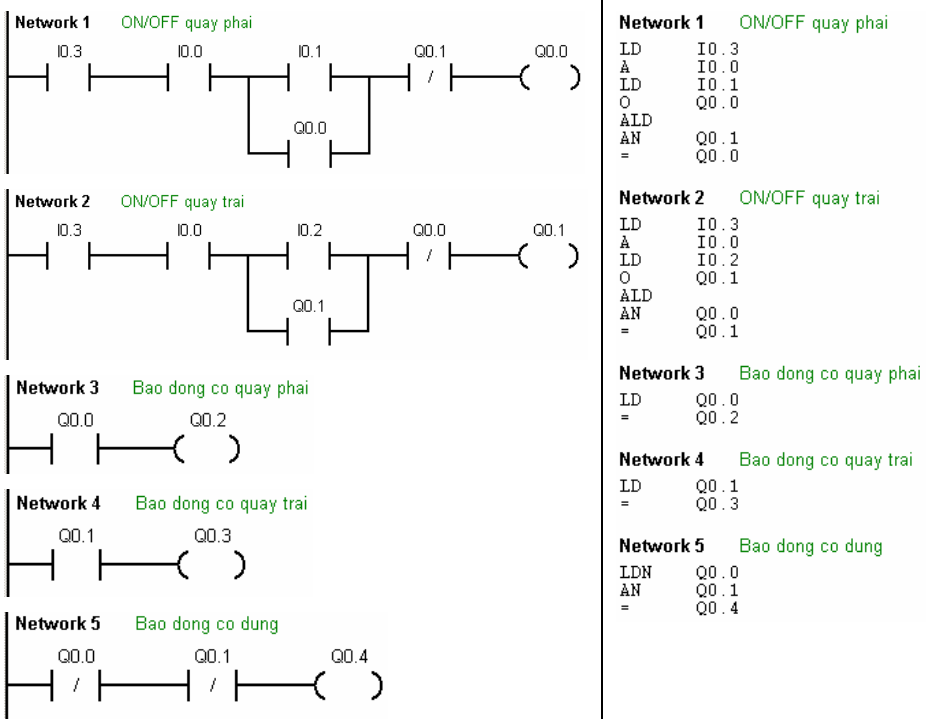


Hình 7.17 a) Sơ đồ nối dây PLC; b) Nối relay với contactor

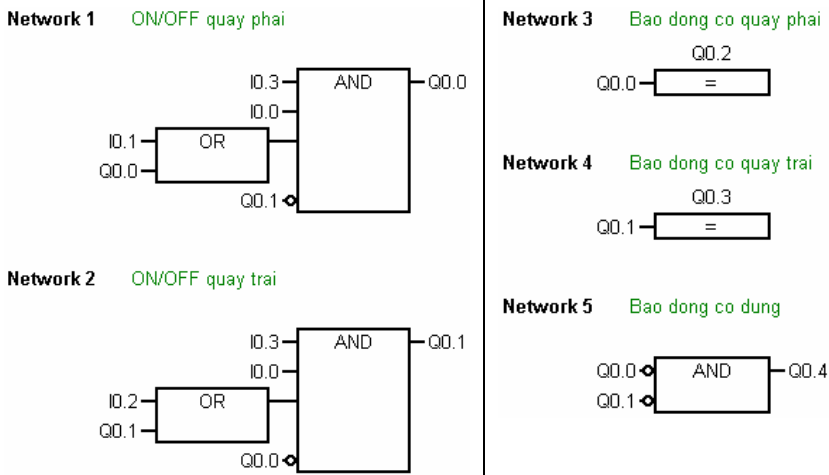
Chú ý: Trong các điều khiển có đảo chiều quay thì tại các ngõ ra PLC điều khiển 2 chiều quay của động cơ ta cần phải nối thêm 2 tiếp điểm thường đóng khóa chéo nhau của 2 contactor (hoặc relay) để đảm bảo an toàn.

Chương trình PLC:

Biểu diễn ở LAD và STL:

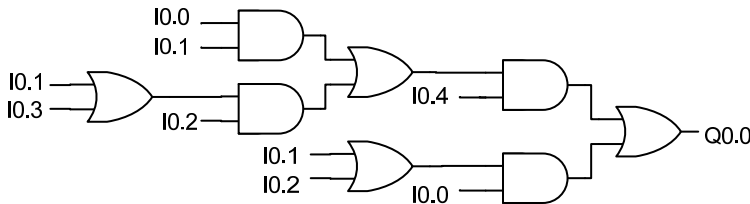


Biểu diễn ở FBD:

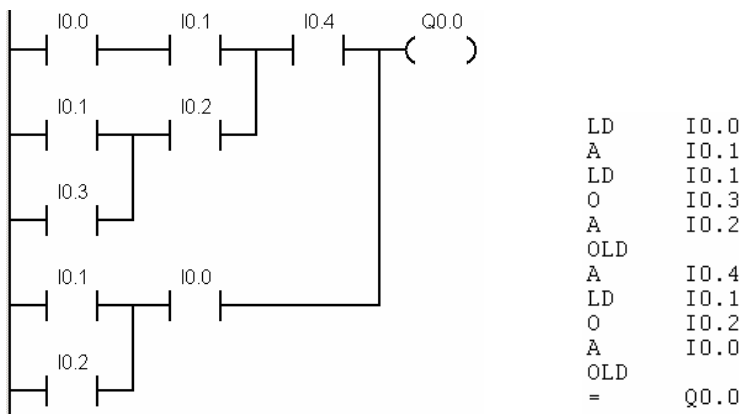


7.5 Bit nhớ M (bit memory)

Trong thiết kế các chương trình điều khiển, ta có thể có một số lượng lớn các logic được liên kết với nhau. Ví dụ như mạch sau:




Chương trình được viết ở LAD và STL:



Với các liên kết logic như thế này thì việc tìm lỗi rất khó khăn. Để dễ dàng hơn trong lập trình và tìm lỗi, thì các kết quả trung gian sẽ được lưu vào một ô nhớ. Trong S7-200 thì các ô nhớ này là bit memory (M).

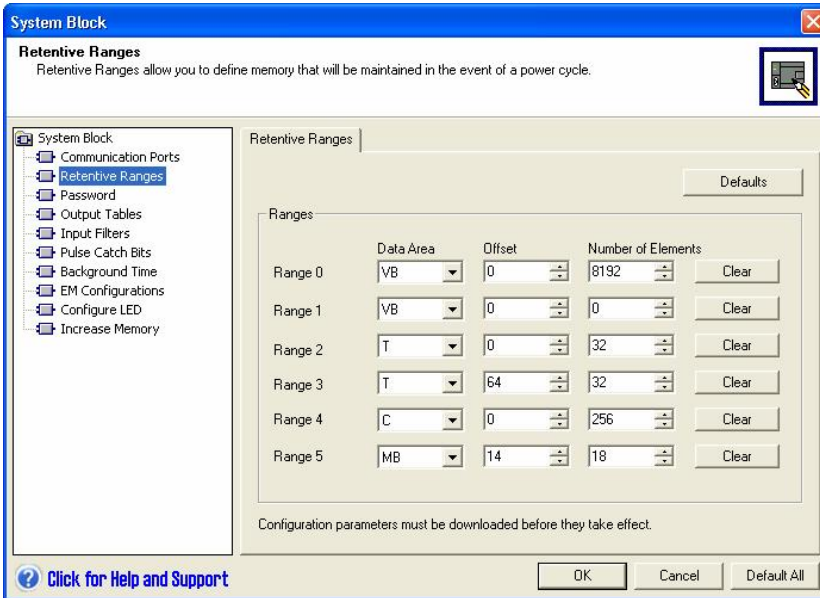
Trong S7-200 có 32 byte nhớ M (từ M0.0 đến M31.7). Chúng được xem như là các ngõ ra trung gian. Khi mất nguồn cấp thì nội dung được nhớ trong các bit nhớ M có thể bị mất hoặc vẫn còn giữ lại tùy thuộc vào việc đặt thuộc tính cho vùng nhớ này là retentive (nhớ lâu dài) hay non-retentive (không nhớ lâu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

** Bit memory có thuộc tính Retentive:* Các bit có thuộc tính này đều giữ lại giá trị của nó khi nguồn cung cấp bị mất. Nghĩa là nếu trước khi bị mất điện, ô nhớ M có giá trị nào thì nó vẫn giữ nguyên giá trị đó khi PLC bị mất điện. Các ô nhớ được ứng dụng để nhớ các trạng thái hoạt động của máy móc hay thiết bị trước khi bị mất điện. Ở lần khởi động kế tiếp thì các máy móc hay thiết bị có thể tiếp tục làm việc tại vị trí trước lúc mất điện. Vùng retentive được thiết

lập bằng cách nhấp chuột vào biểu tượng system Block  hoặc vào menu **View > Component > System Block**. Chọn mục Retentive Ranges. Nếu chọn thẻ defaults thì tất cả các vùng nhớ có thuộc tính retentive đều theo

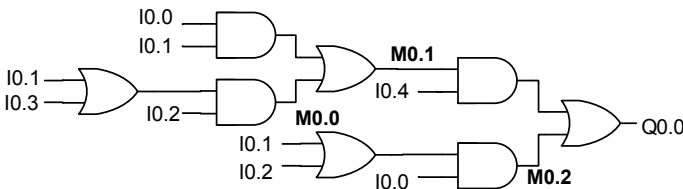
chuẩn của nhà sản xuất. Đối với vùng nhớ M thì bắt đầu từ byte MB14 đến MB31. Tuy nhiên chúng ta vẫn có thể đặt lại theo ý muốn (hình 7.18).

* *Bit memory có thuộc tính non-retentive*: Giá trị các bit này bị xóa khi PLC mất nguồn cung cấp. Theo chuẩn nhà sản xuất thì ta có MB0 đến MB13 ở thuộc tính non-retentive.



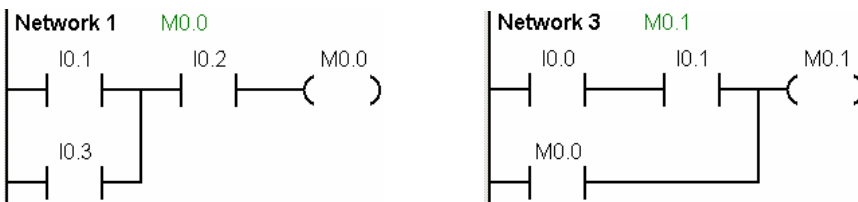
Hình 7.18: Màn hình thiết lập retentive memory.

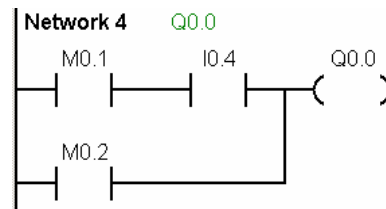
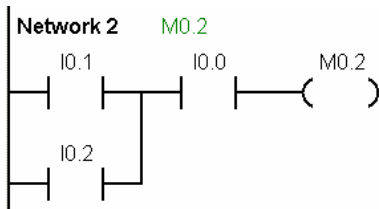
Khi sử dụng bit memory (M), ta có thể làm cho chương trình dễ đọc hơn. Sơ đồ mạch như hình 7.19.



Hình 7.19: Mạch logic được làm cho dễ đọc hơn với bit memory.

Chương trình ở LAD và STL như sau:





STL:

```

Network 1   M0.0
LD   I0.1
O    I0.3
A    I0.2
=    M0.0
    
```

```

Network 3   M0.1
LD   I0.0
A    I0.1
O    M0.0
=    M0.1
    
```

```

Network 2   M0.2
LD   I0.1
O    I0.2
A    I0.0
=    M0.2
    
```


```

Network 4   Q0.0
LD   M0.1
A    I0.4
O    M0.2
=    Q0.0
    
```

7.6 Các lệnh SET, RESET và mạch nhớ RS

7.6.1 Lệnh SET

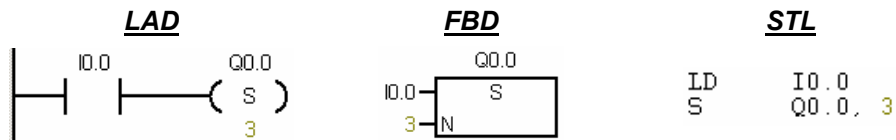
Lệnh SET (S) là lệnh thông dụng rất thường được sử dụng và lệnh này đều có trong hầu hết các PLC. Lệnh Set sẽ đặt trạng thái của một hoặc nhiều bit (thuộc vùng nhớ V, M, Q, T, C, SM, L) có địa chỉ liên tục lên mức 1 và duy trì ở trạng thái này cho đến khi bị xóa bằng một lệnh khác. Chúng ta có thể Set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

Cú pháp ở STL: $S \quad S_Bit, n$ và ở LAD: 

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: Khi tín hiệu tại I0.0 lên mức 1 thì sẽ set 3 bit từ Q0.0 đến Q0.2. Chương trình ở 3 dạng như sau:



Khi tín hiệu tại I0.0 xuống mức 0 thì 3 ngõ ra Q0.0, Q0.1, Q0.2 vẫn duy trì ở mức 1.

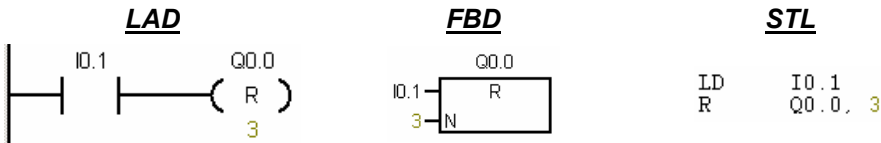
7.6.2 Lệnh RESET (R)

Lệnh Reset (R) đặt trạng thái của một hoặc nhiều bit có địa chỉ liên tục xuống mức 0. Tương tự như lệnh Set chúng ta có thể Reset tới 255 bit nhớ thuộc các vùng nhớ V, M, Q, T, C, SM, L. Lệnh RESET chỉ được thực hiện khi Stack 0 có giá trị logic “1”.

Cú pháp ở STL: $R \ S_Bit, n$ và ở LAD: $\left(\begin{matrix} S_Bit \\ R \\ n \end{matrix} \right)$

Với S_Bit là bit đầu tiên của vùng nhớ cần đặt xuống mức logic “0”.
và n là số lượng bit bắt đầu từ S_Bit .

Ví dụ: Khi tín hiệu tại I0.1 lên mức 1 thì sẽ reset 3 bit từ Q0.0 đến Q0.2 về logic “0”. Chương trình ở 3 dạng như sau:



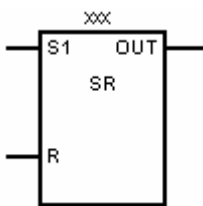
7.6.3 Mạch nhớ R-S

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*.

7.6.3.1 Ưu tiên SET (khâu SR)

Biểu diễn ở LAD:



Với:

xxx : Địa chỉ cần điều khiển

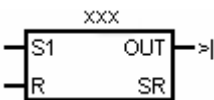
$S1$: Ngõ vào Set. Ký hiệu ưu tiên Set.

R : Ngõ vào Reset.

OUT : Ngõ ra, có thể nối với một địa chỉ dạng bit

SR : Ký hiệu gọi nhớ khâu SR


và FBD:



Nếu cả hai điều kiện cho S và R lên mức logic “1” thì ngõ ra OUT là “1”.

Bảng sự thật

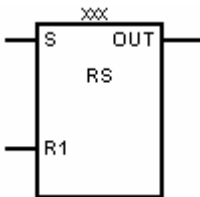
S1	R	OUT
0	0	Trạng thái trước
0	1	0
1	0	1
1	1	1

Để lấy khâu SR, ta nhấp chuột vào dấu cộng của  Bit Logic trong cây lệnh, chọn phần tử SR và kéo thả vào network mong muốn.

Khâu SR tương đương với mạch tự duy trì ưu tiên mở máy trong điều khiển dùng contactor.

7.6.3.2 Ưu tiên RESET (khâu RS)

Biểu diễn ở LAD:



Với:

xxx: Địa chỉ cần điều khiển

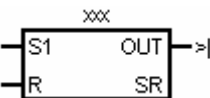
S: Ngõ vào Set.

R1: Ngõ vào Reset. Ký hiệu ưu tiên ReSet.

OUT: Ngõ ra, có thể nối với một địa chỉ dạng bit

RS: Ký hiệu gọi nhớ khâu RS


và FBD:



Nếu cả hai điều kiện cho S và R lên mức logic “1” thì ngõ ra OUT là “0”.

Bảng sự thật

S1	R	OUT
0	0	Trạng thái trước
0	1	0
1	0	1
1	1	0

Để lấy khâu RS, ta nhấp chuột vào dấu cộng của  Bit Logic trong cây lệnh, chọn phần tử RS và kéo thả vào network mong muốn.

Khâu RS tương đương với mạch tự duy trì ưu tiên dừng máy trong điều khiển dùng contactor.

7.6.4 Các qui tắc khi sử dụng Set và Reset

Khi sử dụng với 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".



7.6.5 Ví dụ ứng dụng mạch nhớ R-S

Ví dụ 7.7 : Mạch ưu tiên mở máy.

Yêu cầu của mạch ưu tiên mở máy như ở mục 7.4.1, tuy nhiên cần phải sử dụng mạch nhớ R-S khi lập trình.

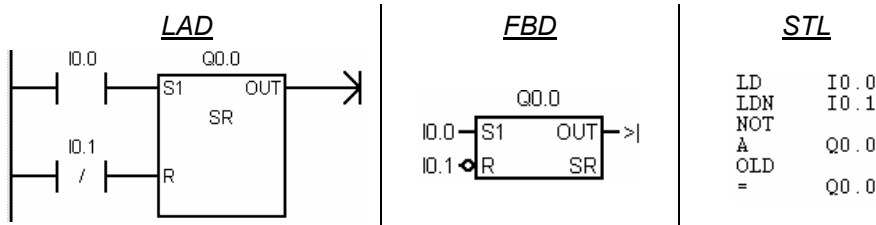
Để tránh lặp lại ta sử dụng lại bảng ký hiệu và sơ đồ nối dây PLC ở mục 7.4.1

Phân tích: Theo yêu cầu của mạch ta có các nhận xét sau:

1. Điều kiện để cho contactor K1 có điện là nút nhấn S1 được ấn → nút nhấn S1 được sử dụng với lệnh S.
2. Điều kiện để cho contactor K1 mất điện là nút nhấn S2 được ấn → nút nhấn S2 được sử dụng với lệnh R.
3. Khi cả hai nút nhấn S1 và S2 cùng ấn thì contactor có điện → sử dụng mạch nhớ **ưu tiên SET (khâu SR)**.
4. Trạng thái bình thường của nút nhấn S1 là thường hở (logic "0" tại ngõ vào I0.0) nên khi lập trình sử dụng tiếp điểm không đảo trạng thái tín hiệu (tiếp điểm  ). Còn S2 là thường đóng (logic "1"

tại ngõ vào I0.1) nên khi lập trình sử dụng tiếp điểm đảo trạng thái tín hiệu (tiếp điểm $\overline{1}$ / 1).

Chương trình được viết như sau:



Ví dụ 7.8 : Mạch ưu tiên dừng máy.

Yêu cầu của mạch ưu tiên dừng máy như ở mục 7.4.2, tuy nhiên cần phải sử dụng mạch nhớ R-S khi lập trình.

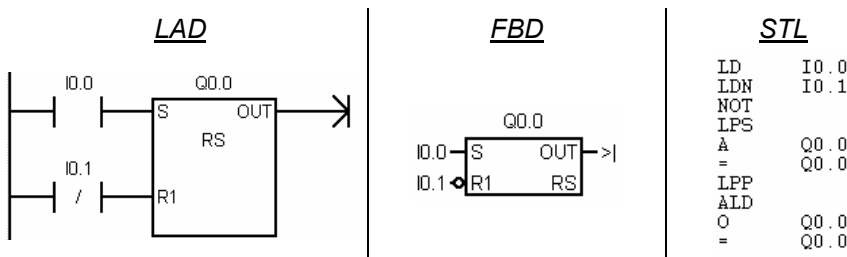
Để tránh lặp lại ta sử dụng lại bảng ký hiệu và sơ đồ nối dây PLC ở mục 7.4.2

Phân tích: Theo yêu cầu của mạch ta có các nhận xét sau:

1. Điều kiện để cho contactor K1 có điện là nút nhấn S1 được ấn → nút nhấn S1 được sử dụng với lệnh S.
2. Điều kiện để cho contactor K1 mất điện là nút nhấn S2 được ấn → nút nhấn S2 được sử dụng với lệnh R.
3. Khi cả hai nút nhấn S1 và S2 cùng ấn thì contactor mất điện → sử dụng mạch nhớ ưu tiên **RESET (khâu RS)**.
4. Trạng thái bình thường của nút nhấn S1 là thường hở (logic “0” tại ngõ vào I0.0) nên khi lập trình sử dụng tiếp điểm không đảo trạng thái

tín hiệu (tiếp điểm 1 / 1). Còn S2 là thường đóng (logic “1” tại ngõ vào I0.1) nên khi lập trình sử dụng tiếp điểm đảo trạng thái tín hiệu (tiếp điểm $\overline{1}$ / 1).

Chương trình được viết như sau:



Ví dụ 7.9 : Mạch đảo chiều quay động cơ.

Để đơn giản và dễ hiểu, ví dụ này lấy lại yêu cầu công nghệ của mạch điều khiển đảo chiều quay ở mục 7.4.4. Tuy nhiên cần phải sử dụng mạch nhớ R-S khi lập trình.

Để tránh lặp lại ta sử dụng lại bảng ký hiệu và sơ đồ nối dây PLC ở mục 7.4.4.

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).

- 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.

$$\text{Set } K1 = S2 \wedge \overline{K2}$$

- 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" → 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" → 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

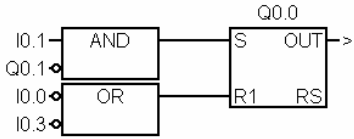
- Đèn sáng khi cả K1 và K2 mất điện.

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

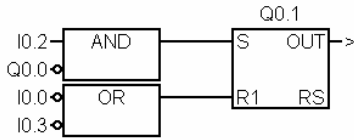
Theo các phân tích ta viết được chương trình như sau:

FBD

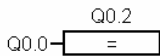
Network 1 ON/OFF quay phải (contactor K1)



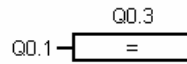
Network 2 ON/OFF quay trái (contactor K2)



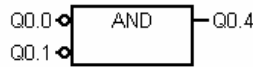
Network 3 Đèn báo quay phải (H1)



Network 4 Đèn báo quay trái (H2)

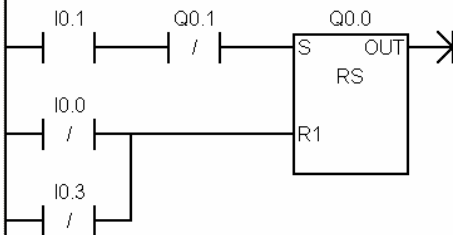


Network 5 Đèn báo dừng (H3)

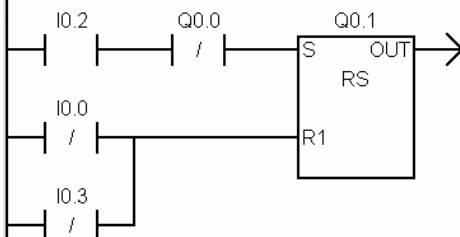


LAD

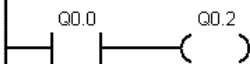
Network 1 ON/OFF quay phải (contactor K1)



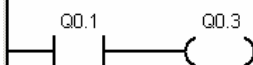
Network 2 ON/OFF quay trái (contactor K2)



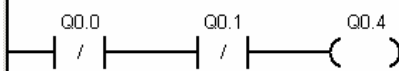
Network 3 Đèn báo quay phải (H1)



Network 4 Đèn báo quay trái (H2)



Network 5 Đèn báo dừng (H3)



STL

Network 1 ON/OFF quay phải (contactor K1)

```
LD I0.1
AN Q0.1
LDN I0.0
ON I0.3
NOT
LPS
A Q0.0
= Q0.0
LPP
ALD
O Q0.0
= Q0.0
```

Network 2 ON/OFF quay trái (contactor K2)

```
LD I0.2
AN Q0.0
LDN I0.0
ON I0.3
NOT
LPS
A Q0.1
= Q0.1
LPP
ALD
O Q0.1
= Q0.1
```

Network 3 Đèn báo quay phải (H1)

```
LD Q0.0
= Q0.2
```

Network 4 Đèn báo quay trái (H2)

```
LD Q0.1
= Q0.3
```

Network 5 Đèn báo dừng (H3)

```
LDN Q0.0
AN Q0.1
= Q0.4
```

7.7 Các lệnh nhận biết cạnh tín hiệu và lệnh NOT

Các lệnh nhận biết cạnh tín hiệu và lệnh NOT thực hiện các thuật toán đặc biệt trên bit đầu tiên của ngăn xếp (Stack 0).

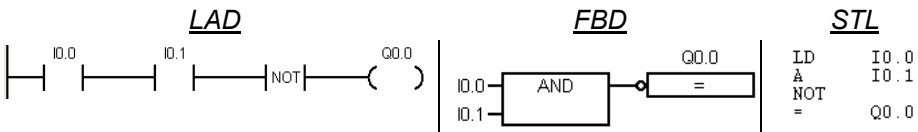
7.7.1 Lệnh NOT

Lệnh NOT đảo giá trị của bit đầu tiên trong ngăn xếp (Stack 0). Nếu sau một phép toán nhị phân mà sử dụng lệnh NOT thì kết quả sẽ bị đảo lại. Nghĩa là nếu kết quả phép toán nhị phân làm cho Stack 0 có giá trị logic "1" thì lệnh NOT sẽ cho kết quả là "0", và ngược lại.

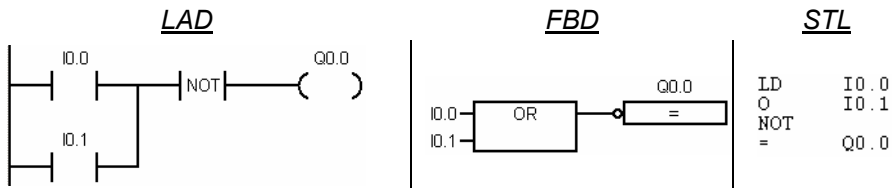
- Kết hợp lệnh NOT sau các cổng logic như OR, AND, XOR ta thu được các cổng NOR, NAND, XNOR.

Ví dụ:

- Cổng NAND với 2 ngõ vào I0.0 và I0.1 và ngõ ra Q0.0 là:



- Cổng NOR với 2 ngõ vào I0.0 và I0.1 và ngõ ra Q0.0 là:



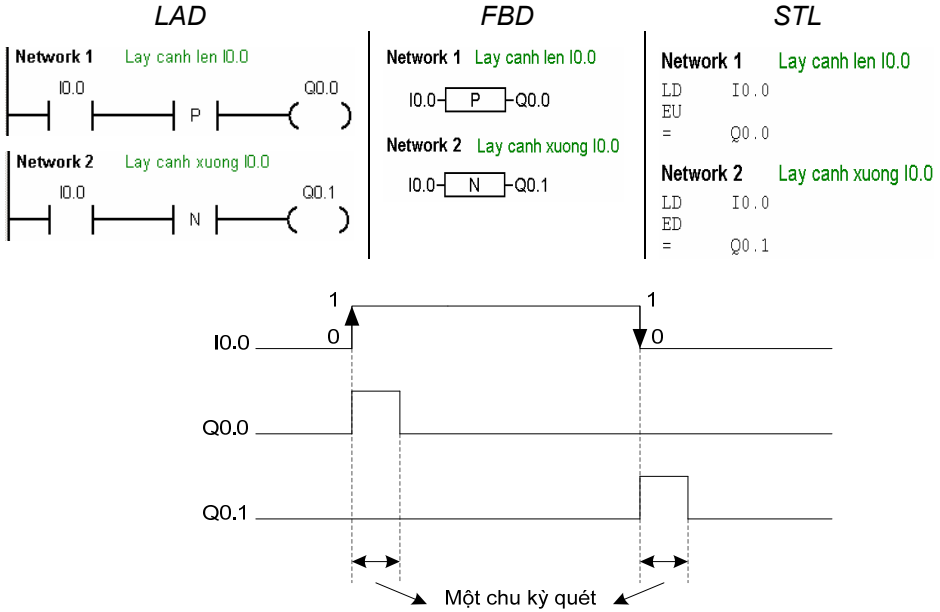
7.7.2 Các lệnh nhận biết cạnh tín hiệu

Hai lệnh nhận biết cạnh tín hiệu là lệnh nhận biết cạnh lên (EU) và nhận biết cạnh xuống (ED).

Lệnh nhận biết cạnh lên (EU) sẽ đặt giá trị logic "1" vào bit đầu tiên của Stack 0 trong một chu kỳ quét chương trình khi phát hiện sự chuyển trạng thái từ 0 lên 1 trong Stack 0. Còn các trường hợp khác nó sẽ đặt Stack 0 về "0".

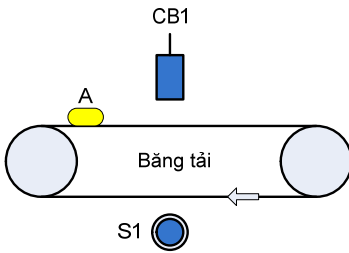
Lệnh nhận biết cạnh xuống (ED) sẽ đặt giá trị logic "1" vào bit đầu tiên của Stack 0 trong một chu kỳ quét chương trình khi phát hiện sự chuyển trạng thái từ 1 xuống 0 trong Stack 0. Còn các trường hợp khác nó sẽ đặt Stack 0 về "0".

Ví dụ: Lấy cạnh lên của I0.0 xuất ra Q0.0, còn cạnh xuống xuất ra Q0.1.



Hình 7.20: Giải đồ thời gian của ví dụ lấy cạnh lên và xuống của tín hiệu.

Ví dụ 7.10: Viết chương trình điều khiển đơn giản cho băng tải sản phẩm (hình 7.21). Khi sản phẩm A được vận chuyển đến vị trí cần thao tác thì băng tải dừng lại (được phát hiện bởi cảm biến CB1). Ấn nút S1 thì băng tải tiếp tục hoạt động cho đến khi nào một sản phẩm đến đúng vị trí thì dừng lại. Quá trình cứ lặp lại như trên.



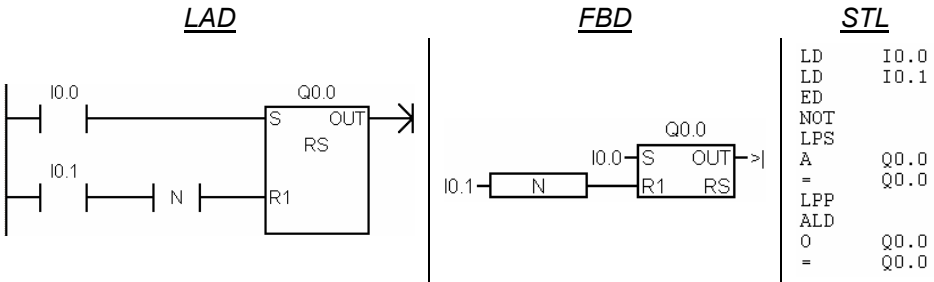
Hình 7.21: ví dụ 7.10

Phân tích:

- Điều kiện Set băng tải: Nút nhấn S1
- Điều kiện Reset băng tải: Cảm biến CB1.
- Sản phẩm đến cảm biến CB1 thì băng tải dừng lại, như vậy cảm biến luôn bị tác động. Nếu ta dùng ưu tiên Reset thì không thể nào khởi động lại băng tải. Còn nếu dùng ưu tiên Set thì khi nào sản phẩm qua khỏi cảm biến mới có thể buông tay thả nút nhấn S1 → Dùng lệnh nhận biết cạnh tín hiệu để khống chế. Và để chắc chắn sản phẩm đã qua cảm biến thì sử dụng lệnh nhận biết cạnh xuống.

Chương trình như sau:

		Symbol	Address	Comment
1		S1	I0.0	Nút nhấn khởi động băng tải
2		CB1	I0.1	Cảm biến sản phẩm dừng vị trí
3		K1	Q0.0	Contactơ dòng điện cho động cơ kéo băng tải



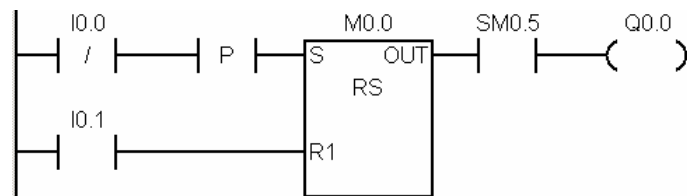
7.8 Các Bit nhớ đặc biệt (Special Memory bits)

Các bit nhớ SM (Special memory bits) cung cấp nhiều chức năng trạng thái và điều khiển, cũng như cung cấp thông tin truyền thông giữa S7-200 và chương trình. Các bit nhớ đặc biệt có thể được sử dụng ở dạng bits, bytes, words và double words. Trong phần này chỉ trình bày các bit trạng thái của SMB0. Còn các bit nhớ SM khác sẽ được trình bày ở mỗi chương tương ứng trong quyển sách này và ở quyển tiếp theo (tập 2).

SMB0 chứa tám bit trạng thái và được cập nhật ở mỗi chu kỳ quét của S7-200. Đây là các bit nhớ chỉ đọc.

Bit	Chức năng
SM0.0	Bit luôn luôn có trạng thái 1
SM0.1	Bit có trạng thái 1 ở vòng quét đầu tiên của chương trình
SM0.2	Bit báo dữ liệu bị thất lạc (0: dữ liệu còn đủ, 1: dữ liệu bị thất lạc).
SM0.3	Bit báo PLC được đóng nguồn. (1: ở vòng quét đầu tiên, 0: ở các vòng quét còn lại).
SM0.4	Bit tạo ra xung có chu kỳ 1 phút (0: trong 30s đầu, 1 trong 30s sau).
SM0.5	Bit tạo xung có chu kỳ 1s (tần số 1 Hz) (0: trong 0,5s đầu ; 1 trong 0,5 s sau).
SM0.6	Bit lên 1 ở một vòng quét và xuống 0 ở vòng quét tiếp theo. Nó được sử dụng để làm ngõ vào của bộ đếm vòng quét.
SM0.7	Bit báo vị trí của công tắc chọn chế độ làm việc của PLC (0: TERM, 1: RUN).

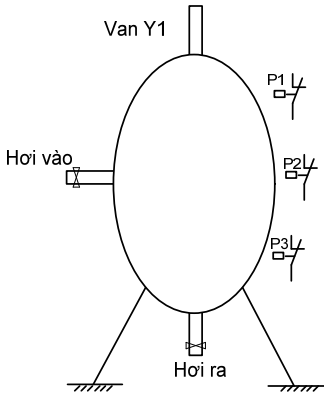
Ví dụ: Khi có tín hiệu sự cố (ngõ vào I0.0 (NC) xuống mức 0) thì đèn báo sự cố (Q0.0) sẽ nhấp nháy 1 Hz. Nhấn nút I0.1 để Reset.



7.9 Câu hỏi và bài tập

Các bài tập ứng dụng giả sử dùng CPU 224 DC/DC/DC để điều khiển.

BT7.1 An toàn cho lò hơi



Hình 7.22 Mô hình lò hơi

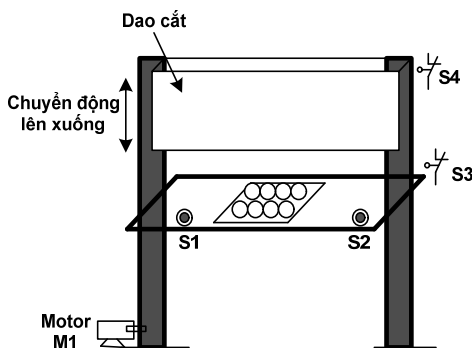
Một thiết bị lò hơi có hơi đi vào và ra khỏi lò được thực hiện tự động qua bộ điều chỉnh đặt ở bên ngoài. Lò hơi có đặt 3 bộ cảm biến áp suất P1, P2 và P3 ở các vị trí khác nhau để kiểm soát quá áp suất. Mạch an toàn sẽ hoạt động khi có sự cố, trường hợp áp suất trong lò hơi tăng quá cao thì van an toàn 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 mạch an toàn mở van từ tính Y1.
- Vẽ sơ đồ nối dây tín hiệu phần cứng

* Bảng ký hiệu:

		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				

BT7.2 Điều khiển cơ cấu máy dập










Hình 7.23 Mô hình máy dập nhỏ

Một cơ cấu dập trong một máy dập nguyên liệu (ví dụ dập ra các vỏ hộp) có thể chuyển động nâng lên hay hạ xuống nhờ một động cơ điện M1 quay 2 chiều. Để đảm bảo an toàn cho tay người vận hành thì chỉ khi nào người vận hành dùng cả 2 tay nhấn đồng thời 2 nút nhấn S1 (NO) và S2 (NO) thì bàn dập mới hạ xuống. Khi hạ xuống động cơ công tắc hành trình giới hạn dưới S3 (NC) thì tự chạy nâng lên cho tới khi động cơ công tắc hành trình giới hạn trên S4 (NC) thì dừng lại. Chu kỳ lặp lại

khi nào người vận hành lại nhấn 2 nút nhấn S1 và S2.

* Bảng ký hiệu:

		Symbol	Address	Comment
1		S1	I0.0	Nut nhấn NO
2		S2	I0.1	Nut nhấn NO
3		S3	I0.2	Cong tắc hành trình NC
4		S4	I0.3	Cong tắc hành trình NC
5		K1	Q0.0	Contactơ Motor M1 quay phải
6		K2	Q0.1	Contactơ Motor M1 quay trái

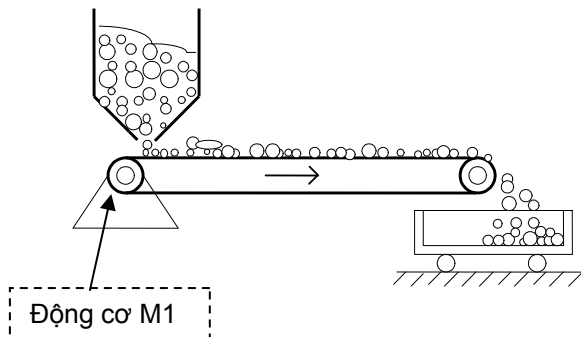
Hãy :

- Viết chương trình điều khiển
- Vẽ sơ đồ nối dây phần cứng

BT7.3 Bảng tải chuyển vật liệu








Một thiết bị băng tải dùng để chuyển vật liệu từ thùng chứa vào xe gòong. Hãy viết chương trình sao cho: Khi bật công tắc 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 xảy ra sự cố quá dòng (tiếp điểm nhiệt F3 (NC) tác động) thì động cơ sẽ dừng lại.

❖ Sơ đồ công nghệ:



Hình 7.24 Băng tải chuyển vật liệu

* Bảng ký hiệu:

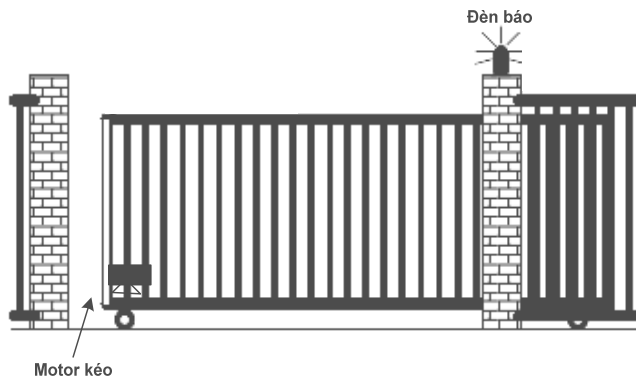
		Symbol	Address	Comment
1		S0	I0.0	Cong tắc khởi động
2		S1	I0.1	Start
3		S2	I0.2	Stop
4		F3	I0.3	Tiếp điểm nhiệt NC
5		H1	Q0.0	Đèn báo sáng
6		M1	Q0.1	Contactơ Motor M1

BT7.4 Điều khiển cổng 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 :
 - Viết 2 chương trình con: Sub0 cho chế độ tay và Sub1 cho chế độ tự động.
 - Vẽ sơ đồ nối dây phần cứng

❖ Sơ đồ công nghệ:



Hình 7.15 Điều khiển cổng

❖ Bảng ký hiệu:

		Symbol	Address	Comment
1		S0	I0.0	= 0 là che do tay, = 1 là che do tu động
2		S1	I0.1	Mở cửa
3		S2	I0.2	Đóng cửa
4		S3	I0.3	Công tắc hành trình mở
5		S4	I0.4	Công tắc hành trình đóng
6		S5	I0.5	Stop
7		F3	I0.6	Quá tải
8		H1	Q0.0	Báo hoạt động
9		K1	Q0.1	Contactơ cho M1 quay phải (mở cửa)
10		K2	Q0.2	Contactơ cho M1 quay trái (đóng cửa)

BT7.5 Điều khiển xe rót vật liệu vào bồn chứa

Một xe kéo dùng để rót vật liệu vào bồn chứa. Khi bật công tắc khởi động S0 (NO) thì đèn H0 sáng báo hệ thống sẵn sàng làm việc. Khi nhấn nút S1 (NO), động cơ M1 có điện kéo xe di chuyển lên, đồng thời đèn H1 chớp sáng với tần số 1Hz. Khi xe lên tới vị trí trên cùng dụng phải công tắc hành trình S4 (NC) thì dừng lại. Nhấn nút S2 (NO) động cơ M1 đảo chiều và kéo xe di chuyển xe xuống, đồng thời đèn báo H2 chớp với tần số 1Hz. Khi xe đến vị trí cuối cùng dụng phải công tắc hành trình S3 (NC) thì dừng lại. Khi động cơ M1 có sự cố quá dòng (tiếp điểm nhiệt F3 (NC) tác động) thì động cơ sẽ dừng lại và đèn H0 sẽ chớp sáng với tần số 1Hz.. Quá trình mới được khởi động khi bật lại công tắc S0. Hãy:

- Viết chương trình điều khiển
- Vẽ sơ đồ nối dây phần cứng với PLC

❖ Bảng ký hiệu:

		Symbol	Address	Comment
1		S0	I0.0	Công tắc khởi động
2		S1	I0.1	Xe chạy lên
3		S2	I0.2	Xe chạy xuống
4		S3	I0.3	Công tắc hành trình dưới
5		S4	I0.4	Công tắc hành trình trên
6		F3	I0.5	Quá tải
7		H0	Q0.0	Báo hoạt động + Báo sự cố
8		H1	Q0.1	Báo xe chạy lên
9		H2	Q0.2	Báo xe chạy xuống
10		K1	Q0.3	Contactơ cho M1 quay phải (Xe lên)
11		K2	Q0.4	Contactơ cho M1 quay trái (Xe xuống)