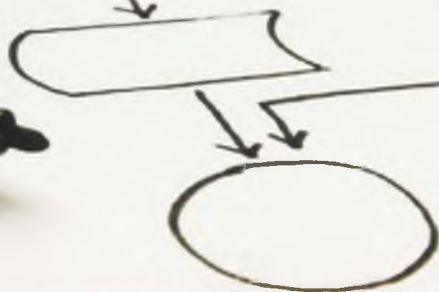


NGUYỄN VĂN KHANG



BỘ ĐIỀU KHIỂN LOGIC KHẢ TRÌNH **PLC** VÀ ỨNG DỤNG

(Xuất bản lần thứ 2, có sửa chữa)



NHÀ XUẤT BẢN BÁCH KHOA HÀ NỘI

NGUYỄN VĂN KHANG

BỘ ĐIỀU KHIỂN LOGIC KHẢ TRÌNH

PLC

VÀ ỨNG DỤNG

(Xuất bản lần thứ hai, có sửa chữa)

NHÀ XUẤT BẢN BÁCH KHOA HÀ NỘI

Bản quyền thuộc về trường Đại học Bách Khoa Hà Nội.

Mọi hình thức xuất bản, sao chép mà không có sự cho phép bằng văn bản của trường là vi phạm pháp luật.

Mã số: 205 – 2015/CXBIPH/14 – 02/BKHN

Biên mục trên xuất bản phẩm của Thư viện Quốc gia Việt Nam

Nguyễn Văn Khang

Bộ điều khiển logic khả trình PLC và ứng dụng /-Nguyễn Văn Khang. -

H. : Bách khoa Hà Nội, 2015. - 208tr. : hình vẽ, bảng ; 24cm

Thư mục: tr. 203

ISBN 978-604-938-208-6

1. Điều khiển logic 2. Lập trình PLC

629.89 - dc23

BKH0038p-CIP

LỜI NÓI ĐẦU

Bộ điều khiển logic khả trình PLC được sử dụng rộng rãi trong các hệ thống tự động hóa công nghiệp. PLC trở thành một thành phần cơ bản để tích hợp hệ thống điều khiển các dây chuyền sản xuất. PLC được sử dụng không những trong các bộ điều khiển logic mà còn trong các bộ điều khiển quá trình, tích hợp trong các hệ điều khiển phân tán (DCS) và các hệ thống giám sát, thu thập dữ liệu (SCADA).

Chủng loại của PLC rất phong phú từ nhiều nhà sản xuất, nhiều hãng cung cấp thiết bị. Một số họ PLC do hãng sản xuất chuyên dùng để tích hợp trong các hệ thống của chính họ. Một số họ PLC có tính thương mại, phục vụ các nhà thiết kế và tích hợp hệ thống. Thậm chí, một nhà sản xuất có thể đưa ra nhiều chủng loại PLC rất khác nhau đáp ứng nhu cầu thực tế. Có một đặc điểm chung là, nhà cung cấp PLC đồng thời là nhà cung cấp phần mềm, các công cụ lập trình, các sản phẩm liên quan và các giải pháp ứng dụng. Các hãng sản xuất PLC nổi tiếng và có sản phẩm được sử dụng rộng rãi có thể kể đến: Siemens (Đức), Allen-Bradley (Mỹ), Omron (Nhật)...

Vai trò to lớn, tính ứng dụng rộng rãi và mềm dẻo của PLC đã nhận được sự quan tâm sâu sắc của các nhà thiết kế, tích hợp hệ thống, các kỹ sư, cán bộ kỹ thuật... thuộc nhiều lĩnh vực khác nhau và được giảng dạy ở các ngành tự động hóa, điều khiển tự động, điện tử... trong các trường đại học, cao đẳng công nghệ. Tuy nhiên, việc tiếp cận một cách có hệ thống còn gặp nhiều khó khăn do sự phong phú về chủng loại PLC, tính riêng biệt của từng loại sản phẩm, phần mềm và các công cụ của từng nhà cung cấp.

Nội dung cuốn sách trình bày các kiến thức chung cơ bản nhất về PLC với quan điểm hệ thống hóa và tính ứng dụng, với hy vọng người đọc có thể sử dụng cho các chủng loại PLC được lựa chọn. Một số ví dụ ứng dụng được trình bày trong phần Phụ lục trên cơ sở PLC của hãng Omron.

Tác giả xin chân thành cảm ơn sự đóng góp quý báu của tập thể cán bộ Bộ môn Tự động hóa công nghiệp – Viện Điện và Bộ môn Điện tử và Kỹ thuật máy tính – Viện Điện tử Viễn thông, Đại học Bách Khoa Hà Nội về nội dung cuốn sách. Tác giả cũng xin cảm ơn sự giúp đỡ của Văn phòng đại diện hãng Omron tại Hà Nội.

Cuốn sách chắc chắn không tránh khỏi những thiếu sót về nội dung và hình thức. Tác giả xin cảm ơn và trân trọng tiếp thu những ý kiến đóng góp, phản biện của bạn đọc.

Tác giả

MỤC LỤC

LỜI NÓI ĐẦU	3
CHƯƠNG 1. TỔNG QUAN	9
1.1. Bộ điều khiển logic khả trình PLC (Programmable Logic Controller).....	9
1.2. Tính ưu việt của việc sử dụng PLC	10
1.3. Các thành phần và hoạt động của PLC	12
1.4. Quét vòng – nguyên tắc hoạt động cơ bản của PLC	17
1.5. PLC và PC.....	19
1.6. Phân loại PLC và ứng dụng.....	19
1.7. Các họ PLC thông dụng.....	22
1.7.1. Họ SIMATIC của SIEMENS (CHLB Đức)	22
1.7.2. Họ SYSMAC của OMRON (Nhật Bản).....	22
1.7.3. PLC của ALLEN BRADLEY (Mỹ)	22
1.8. Phần mềm và công cụ lập trình	25
1.9. Kết luận	25
CHƯƠNG 2. CÁC THÀNH PHẦN CỦA PLC.....	26
2.1. Sơ đồ khối chức năng hệ điều khiển PLC.....	26
2.1.1. Khối nguồn cung cấp	27
2.1.2. Khối xử lý trung tâm (CPU)	27
2.1.3. Các thiết bị vào/ra.....	32
2.1.4. Các thiết bị ngoại vi (<i>Peripheral Devices</i>)	34
2.2. Các module vào/ra	36
2.2.1. Module vào/ra rời rạc (<i>Discrete I/O module</i>)	36
2.2.2. Các module vào/ra tương tự (<i>Analog I/O modules</i>).....	44
2.2.3. Các module vào/ra đặc biệt.....	45
2.2.4. Các tham số đặc trưng của module vào/ra	47

2.3. Địa chỉ hóa vào/ra.....	48
2.3.1. Địa chỉ hóa module vào/ra rời rạc.....	49
2.3.2. Địa chỉ hóa module vào/ra đặc biệt.....	51
2.4. Cấu hình hệ điều khiển logic PLC.....	52
2.4.1. Cấu hình cơ bản.....	52
2.4.2. Cấu hình đặc biệt.....	54
 CHƯƠNG 3. TỔ CHỨC BỘ NHỚ VÀ CẤU TRÚC DỮ LIỆU	55
3.1. Tổ chức bộ nhớ	55
3.2. Cấu trúc dữ liệu	56
3.3. Tổ chức bộ nhớ và cấu trúc dữ liệu của một số họ PLC	58
3.3.1. Họ SIMATIC S7-200 của SIEMENS [6].....	58
3.3.2. Họ SYSMAC của OMRON [4].....	70
3.3.3. Họ SLC500 của Allen-Bradley [3]	73
 CHƯƠNG 4. THIẾT KẾ HỆ ĐIỀU KHIỂN DÙNG PLC.....	84
4.1. Trình tự các bước thiết kế.....	84
4.1.1. Bước 1: Mô tả bài toán	84
4.1.2. Bước 2: Thống kê thiết bị vào ra	84
4.1.3. Bước 3: Chọn thiết bị.....	84
4.1.4. Bước 4: Thiết lập cấu hình của hệ điều khiển logic.....	85
4.1.5. Bước 5: Địa chỉ hóa vào/ra.....	85
4.1.6. Bước 6: Xây dựng phần mềm.....	85
4.1.7. Bước 7: Chạy mô phỏng.....	86
4.1.8. Bước 8: Chạy công nghệ	86
4.1.9. Bước 9: Lưu giữ phần mềm.....	87
4.1.10. Bước 10: Xây dựng tài liệu.....	87
4.2. Lập trình cho hệ điều khiển logic dùng PLC	87
4.2.1. Chương trình và ngôn ngữ lập trình.....	87
4.2.2. Chế độ lập trình	88
4.2.3. Các phương pháp biểu diễn chương trình.....	88

4.3. Giản đồ thang và phương pháp biểu diễn chương trình bằng giản đồ	
thang (LAD)	90
4.3.1. Giản đồ hình thang.....	90
4.3.2. Phương pháp biểu diễn chương trình điều khiển bằng giản đồ	
hình thang	91
4.4. Một số khái niệm	93
4.4.1. Logic tiếp điểm nhị phân	93
4.4.2. Quan hệ trạng thái của thiết bị vào với Bit dữ liệu	93
4.4.3. Trạng thái thiết bị đầu ra.....	95
4.4.4. Các Bit toán hạng.....	95
4.5. Biểu diễn chương trình bằng dãy lệnh (STL).....	95
4.5.1. Các lệnh cơ bản dạng STL.....	96
4.5.2. Các lệnh khối: ANDLD và ORLD	96
4.5.3. Các lệnh dạng khối chức năng (<i>Function Block</i>).....	98
4.6. Lập trình các hàm logic cơ bản trên PLC	98
4.6.1. Hàm YES.....	98
4.6.2. Hàm NOT	99
4.6.3. Hàm AND.....	99
4.6.4. Hàm OR.....	99
4.6.5. Hàm INHIBITION.....	100
4.6.6. Hàm EXCLUSIVE OR	100
4.6.7. Hàm NAND.....	101
4.6.8. Hàm NOR.....	101
4.6.9. Hàm OR INHIBIT	102
4.6.10. Hàm AND INHIBIT	102
4.7. Các bit đầu ra hai trạng thái ổn định và một trạng thái ổn định	
(bistable và monstable)	104
4.8. Bộ định thời (TIMER).....	108
4.9 Bộ đếm (COUNTER).....	116
4.10. Lập trình các khối phức hợp	123
4.11. Lập trình các hàm số học.....	133

PHỤ LỤC

PHỤ LỤC 1. MỘT SỐ VÍ DỤ ỨNG DỤNG [5]	143
1. Điều khiển đèn giao thông	143
2. Mạch trễ ON/OFF	145
3. Đếm tới 20000	145
4. Bộ TIMER với thời gian đặt tới 1000 giờ	146
5. Điều khiển dây chuyền đóng gói sản phẩm	147
6. Mạch điều khiển cho máy đóng bao	148
7. Điều khiển máy khoan	150
8. Điều khiển quá trình rót/tháo	152
9. Điều khiển băng tải	154
10. Phát hiện và loại bỏ sản phẩm lỗi	155
11. Điều khiển chuyển động của robot	156
12. Điều khiển xe tự hành	159
13. Điều khiển tự động đóng mở cửa kho hàng	160
14. Điều khiển tự động cấp dầu bôi trơn	161
15. Điều khiển động cơ quay băng tải	163
16. Hiện thị mã lỗi	165
17. Đo tuổi thọ dao cắt	167
18. Điều khiển bãi đỗ xe	168
PHỤ LỤC 2. TẬP LỆNH	170
1. Tập lệnh của PLC dòng C (OMRON) [4]	170
2. Tập lệnh của PLC SLC 500 (ALLEN BRADLEY) [3]	175
MỘT SỐ THUẬT NGỮ	199
TÀI LIỆU THAM KHẢO	203
CHỈ MỤC	204

Chương 1

TỔNG QUAN

1.1. BỘ ĐIỀU KHIỂN LOGIC KHẢ TRÌNH PLC (Programmable Logic Controller)

Bộ điều khiển logic khả trình PLC là thiết bị điện tử bán dẫn thực hiện các hàm điều khiển logic bằng chương trình thay thế cho các mạch logic kiểu rơ le (tiếp điểm và phi tiếp điểm). Hoạt động của PLC dựa trên nguyên tắc quét vòng. PLC đọc các tín hiệu logic từ các cổng vào, ví dụ: phím bấm, tiếp điểm, tín hiệu ra của các cảm biến..., thực hiện hàm điều khiển logic bằng chương trình và gửi kết quả đến cổng ra để điều khiển các cơ cấu chấp hành, ví dụ: cuộn dây rơ le, đèn, van điện từ...

Về bản chất, PLC là hệ vi xử lý được thiết kế tương tự máy tính số, với ngôn ngữ lập trình riêng gắn gũi với người sử dụng, được ứng dụng trong các bài toán điều khiển logic. Hạt nhân của hệ là bộ vi xử lý thực hiện các phép tính số học và logic cùng với các thành phần cấu thành hệ như bộ nhớ, các cổng vào/ra,...

Về phạm vi ứng dụng, PLC là thiết bị đặt tại dây chuyền sản xuất, tích hợp với các thành phần của hệ thống điều khiển để thực hiện điều khiển trực tiếp công nghệ một quá trình kỹ thuật. PLC thường làm việc trong môi trường rất khắc nghiệt (nhiệt độ cao, độ ẩm lớn, thời gian hoạt động liên tục) và gắn liền với người vận hành trực tiếp thiết bị. Vì vậy, PLC được thiết kế và chế tạo với các tiêu chuẩn đặc biệt về độ bền, tính module hóa cao, ngôn ngữ lập trình phù hợp và thân thiện với trình độ người sử dụng.

Về chức năng, PLC là thiết bị điều khiển ở mức trường. Ban đầu, chức năng cơ bản của PLC là bộ điều khiển các đại lượng logic. Tuy nhiên, do sự phát triển, hoàn thiện của các bộ vi xử lý và tích hợp hệ thống nên hiện nay PLC còn đóng vai trò là các thiết bị tính toán, được sử dụng như một bộ điều khiển quá trình rời rạc và liên tục. Sự khác nhau về nguyên tắc hoạt động và chức năng của PLC với các máy tính công nghiệp đang ngày càng được thu hẹp. Các máy tính công nghiệp với ưu điểm nổi bật về tốc độ xử lý, bộ nhớ lớn có thể thực hiện nhiều nhiệm vụ và chức năng điều khiển. Các họ PLC hiện đại được tích hợp các tính năng xử lý thông tin, quản lý dữ liệu và mở rộng các chức năng xử lý ngắt. Ngoài chức năng điều khiển, PLC còn đóng vai trò là khâu thu thập và xử lý dữ liệu trong các hệ SCADA và là một nút trong các hệ

điều khiển phân tán (DCS). Vì vậy, với quan điểm hệ thống, PLC là thành phần cơ bản cấu thành hệ điều khiển.

Nghiên cứu ứng dụng PLC trong các hệ điều khiển bao gồm hai vấn đề không thể tách rời nhau: phần cứng và phần mềm. Phần cứng là các thiết bị vật lý cấu thành hệ thống bao gồm: nguồn cung cấp, CPU, module vào/ra và các thiết bị phụ trợ... Các thiết bị vật lý được lắp ghép với nhau tạo thành một cấu hình vật lý của hệ thống. Phần mềm bao gồm hệ điều hành và chương trình ứng dụng. Hệ điều hành do nhà sản xuất cung cấp được cài đặt sẵn trong bộ nhớ của PLC. Chương trình ứng dụng do người sử dụng lập bằng ngôn ngữ lập trình của PLC để thực hiện một thuật toán (*algorithm*) điều khiển xác định. Giữa phần cứng và phần mềm có mối liên hệ chặt chẽ với nhau. Một chương trình ứng dụng chỉ được thiết lập trên cơ sở một cấu hình vật lý cụ thể. Ngược lại, một hệ thống chỉ có thể thực hiện được đúng thuật toán điều khiển nếu chương trình đó được thiết kế phù hợp với cấu hình của nó.

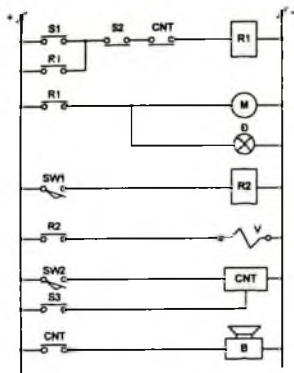
1.2. TÍNH ƯU VIỆT CỦA VIỆC SỬ DỤNG PLC

Việc sử dụng PLC thay thế các bộ điều khiển logic nối dây đem lại các lợi ích căn bản.

Các bộ điều khiển logic nối dây bao gồm các bộ điều khiển logic tiếp điểm và phi tiếp điểm. Các bộ điều khiển logic tiếp điểm sử dụng các phần tử logic kiểu tiếp điểm như: tiếp điểm rơ le điện từ, công tắc hành trình... Các bộ điều khiển phi tiếp điểm sử dụng các phần tử logic bán dẫn như: diode bán dẫn, transistor, IC số... Đặc điểm chung của các bộ điều khiển logic nối dây là các phần tử logic là các phần tử vật lý. Bộ điều khiển logic nối dây thực hiện hàm điều khiển bằng sơ đồ nối các phần tử logic bằng dây dẫn vật lý (dây dẫn điện, mạch in). Sơ đồ thực hiện một hệ điều khiển logic nối dây được trình bày trên hình 1.1.

Trong sơ đồ hình 1.1, các phần tử logic được nối bằng dây dẫn vật lý và sử dụng các rơ le trung gian để thực hiện điều khiển các thiết bị một cách đồng thời.

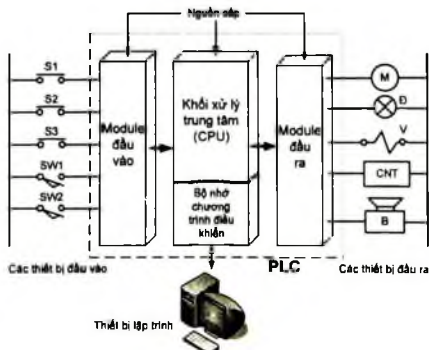
Như vậy, với một cấu trúc vật lý (bao gồm thiết bị vào, thiết bị ra và sơ đồ nối dây), hệ chỉ thực hiện một hàm điều khiển duy nhất. Muốn thay đổi hàm điều khiển, cần phải thay đổi cấu trúc của hệ. Đó là tính không mềm dẻo của bộ điều khiển logic nối dây. Đối với các hệ phức tạp, nhiều phần tử thì tính không mềm dẻo là một nhược điểm lớn. Tuy nhiên, ưu điểm của bộ điều khiển logic nối dây là tốc độ xử lý nhanh, phù hợp với các hệ đơn giản, ít phần tử và công suất lớn.



Hình 1.1. Sơ đồ hệ điều khiển logic nối dây.

S1: *Phím nhấn khởi động (Start);* V: *Van điện từ;*
S2: *Phím nhấn dừng (Stop);* SW1, SW2: *Công tắc hành trình;*
R1, R2: *Rơ le trung gian;;* CNT: *Bộ đếm;*
M: *Động cơ;* S3: *Phím nhấn RESET;*
Đ: *Đèn báo RUN;* B: *Còi báo.*

Đặc điểm của PLC là các phần tử logic được định nghĩa bằng chương trình và thực hiện hàm điều khiển bằng chương trình. Sơ đồ hệ điều khiển logic dùng PLC được trình bày trên hình 1.2.



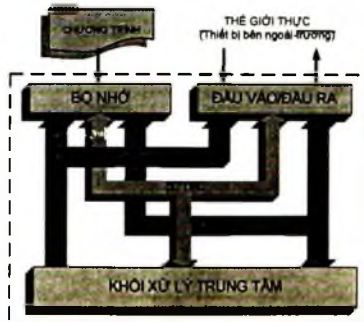
Hình 1.2. Sơ đồ hệ điều khiển logic dùng PLC.

Trong sơ đồ hình 1.2, các module vào và module ra là các thiết bị kết nối với các phần tử logic vật lý bên ngoài. Chương trình điều khiển được lưu giữ trong bộ nhớ. PLC thực hiện tuần tự các lệnh của chương trình để điều khiển các thiết bị tương tự như sơ đồ hình 1.1.

PLC đã thực hiện thay thế các mạch logic nối dây bằng các “mạch logic lập trình được”. Trong các mạch logic này, có thể cắt bỏ, chèn, thêm vào các phần tử một cách dễ dàng và đơn giản. Trong thực tế, việc thay đổi tham số điều khiển của chương trình, thậm chí thay đổi chương trình điều khiển thường xuyên xảy ra khi thay đổi sản phẩm, thay đổi công nghệ. Đối với hệ điều khiển logic dùng PLC, cùng một cấu trúc vật lý có thể thực hiện các hàm điều khiển khác nhau, tùy thuộc vào chương trình. Nghĩa là, có thể thay đổi hàm điều khiển mà không cần thay đổi cấu trúc của hệ. Đó là tính mềm dẻo của PLC. Tính mềm dẻo này đảm bảo PLC được sử dụng có hiệu quả cao trong các hệ phức tạp, có nhiều phần tử. Ngoài ra, ưu điểm của PLC là hoạt động tin cậy, tiêu thụ năng lượng ít, dễ dàng mở rộng hệ thống, việc chuyển giao công nghệ được nhanh và hiệu quả hơn so với các hệ logic nối dây. Hạn chế của PLC là tính tác động nhanh không cao và chỉ sử dụng để tạo ra các tín hiệu điều khiển công suất nhỏ. Một ưu điểm cần nhấn mạnh khi mở rộng phạm vi ứng dụng của PLC là có thể tiến hành mô phỏng khi khảo sát và thiết kế hệ thống. PLC với các chức năng truyền thông có thể kết nối mạng với các bộ điều khiển khác, với các hệ thống máy tính và điều khiển để thực hiện các chức năng điều khiển quá trình, điều khiển phân tán, thu nhập dữ liệu và giao diện người – máy.

1.3. CÁC THÀNH PHẦN VÀ HOẠT ĐỘNG CỦA PLC

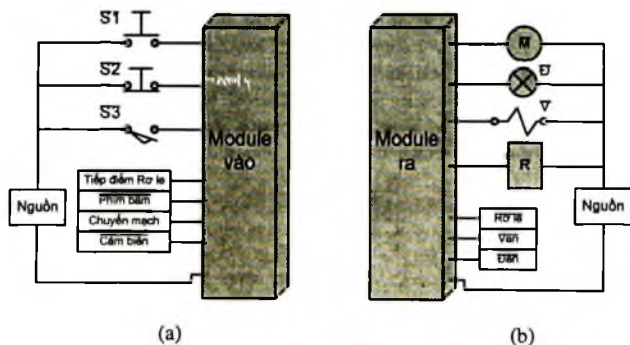
Thành phần cơ bản của PLC gồm có: khối xử lý trung tâm (CPU – Central Processing Unit), các module vào/ra, nguồn cung cấp (Power Supply Unit) và thiết bị lập trình (Programming Device). Sơ đồ cấu trúc bao gồm các thành phần của PLC được trình bày trên hình 1.3.



Hình 1.3. Sơ đồ cấu trúc của PLC.

Chương trình được soạn thảo trong thiết bị lập trình và được nạp vào bộ nhớ của PLC. Các module vào/ra là các cổng ghép nối PLC với thiết bị bên ngoài (gọi là thiết bị trường – *Field Device*). Các cổng vào/ra có nhiệm vụ chuyển đổi thích ứng giữa các nguồn tín hiệu và PLC. Các module vào là các thiết bị nhận tín hiệu từ thiết bị vào, chuyển đổi thành dữ liệu, ví dụ: phím bấm, công tắc hành trình, cảm biến, chuyển mạch... Các module ra là các thiết bị ghép nối PLC với các thiết bị ra, chuyển đổi dữ liệu thành tín hiệu điều khiển các cơ cấu chấp hành, ví dụ: rơ le, van, đèn... Sơ đồ nối các thiết bị vào/ra (I/O) với các module vào/ra được trình bày trên hình 1.4.

Trong thực tế, các cổng vào/ra có hai loại: loại cố định (*Fixed*) và loại dạng module hóa (*Modular*). Loại cố định được sử dụng cho các PLC cỡ nhỏ, các cổng vào/ra gắn cố định vào khối CPU, không thay đổi được vị trí. Ưu điểm của loại này là giá thành thấp. Tuy nhiên, nếu muốn mở rộng cổng vào/ra thì cần phải trang bị thêm khối mở rộng tương ứng. Loại module hóa được sử dụng trong đa số các trường hợp và là cấu trúc tiêu chuẩn của PLC. Các module vào/ra có thể tháo lắp, thay đổi vị trí dễ dàng trên các khe cắm (*Slot*) và các rãnh (*Rack*). Cấu trúc kiểu này (bao gồm cả các đầu nối) tạo thành bảng mạch Bus (*Backplane*), trên đó có thể lắp các khối nguồn, CPU, module vào/ra, module mở rộng... và thực hiện trao đổi thông tin với nhau.



Hình 1.4. Sơ đồ module vào (a) và module ra (b).

Khối nguồn cung cấp nguồn một chiều cho các khối được lắp đặt vào bảng mạch Bus. Công suất của khối nguồn được chọn tùy thuộc vào cấu hình của hệ. Trong đa số các trường hợp, nguồn cung cấp này không phù hợp với các thiết bị trường. Vì vậy, các thiết bị trường thường được cung cấp bằng nguồn ngoài riêng.

Khối CPU là bộ não của PLC, hạt nhân là bộ vi xử lý (8 Bit, 16 Bit,...), quyết định tính chất và khả năng của PLC: tốc độ xử lý, khả năng quản lý vào/ra... CPU thực hiện chương trình trong bộ nhớ chương trình, đưa ra các quyết định và trao đổi thông tin với bên ngoài thông qua các cổng vào/ra.

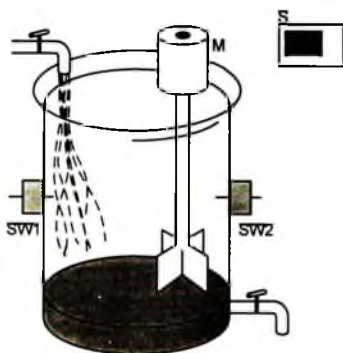
Thiết bị lập trình được sử dụng để soạn thảo chương trình, nạp vào bộ nhớ của PLC. Ngoài ra, thiết bị lập trình còn được sử dụng để theo dõi, gỡ rối, thay đổi lệnh, lưu giữ chương trình và thực hiện các thao tác điều khiển PLC. Thiết bị lập trình có các loại sau: bộ lập trình cầm tay và bộ lập trình chuyên dụng dạng máy tính laptop hoặc máy tính PC có cài đặt phần mềm lập trình. Các thiết bị lập trình được trình bày trên hình 1.5.



Hình 1.5. Các loại thiết bị lập trình cho PLC.

Hoạt động của PLC có thể được mô tả thông qua một ví dụ điều khiển quá trình đơn giản được trình bày trên hình 1.6.

Động cơ M được sử dụng để trộn chất lỏng trong thùng chứa. Cảm biến áp suất SW1 và cảm biến nhiệt độ SW2 ở dạng tiếp điểm. Khi nhiệt độ và áp suất trong thùng chứa đạt giá trị cho trước thì cảm biến áp suất và cảm biến nhiệt độ tác động. Các tiếp điểm SW1 và SW2 chuyển sang trạng thái ON (*Closed*). Ngoài ra, trong sơ đồ sử dụng phím bấm S để khởi động M bằng tay. Sơ đồ mạch logic dạng hình thang (*LAD – Ladder Diagram*) được trình bày trên hình 1.7.



Trong đó:

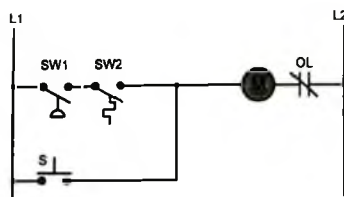
SW1: Cảm biến áp suất;

M: Động cơ trộn;

SW2: Cảm biến nhiệt độ;

S: Nút ấn khởi động M bằng tay.

Hình 1.6. Mô hình hệ điều khiển trộn chất lỏng.

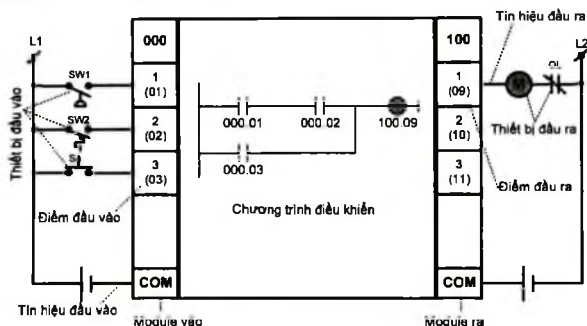


Hình 1.7. Sơ đồ mạch điều khiển logic.

Trong đó: L1, L2 là hai dây nguồn cung cấp; OL là tiếp điểm bảo vệ quá tải; M là cuộn dây khởi động động cơ.

Sơ đồ hình 1.7 chỉ ra rằng: Động cơ M được cấp điện khi cả hai tiếp điểm SW1 và SW2 ở trạng thái ON hoặc phím S được nhấn.

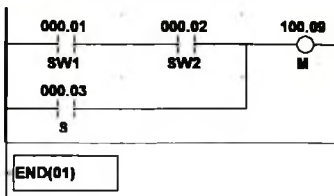
Khi sử dụng PLC để thực hiện mạch điều khiển trên, các phần tử vào là các tiếp điểm SW1, SW2 và S, phần tử ra là động cơ M được nối với các module vào và module ra theo sơ đồ trên hình 1.8.



Hình 1.8. Sơ đồ nối các phần tử vào và phần tử ra.

Mỗi phần tử vào và mỗi phần tử ra được nối với một điểm (*Terminal, Point*) trên module vào hoặc module ra. Mỗi đầu vào/ra sẽ có một địa chỉ xác định. PLC quản lý vào/ra theo các địa chỉ này. Vấn đề này sẽ được xét kỹ trong chương 3. Trên sơ đồ hình 1.8, phía module vào, SW1 nối với đầu nối 1 có địa chỉ 000.01, SW2 nối với đầu nối 2 có địa chỉ 000.02, SW3 nối với đầu nối 3 có địa chỉ 000.03 và ở phía module ra, cuộn dây khởi động M nối với đầu nối 1 có địa chỉ 100.09. Chú ý là vị trí nối các phần tử vào và ra là tùy ý. Nhưng mỗi vị trí trên module vào và ra có địa chỉ xác định. Chương trình điều khiển biểu diễn ở dạng giản đồ hình thang (*LAD*) được trình bày trên hình 1.9 và được nạp vào bộ nhớ của PLC.

Giản đồ thang (Ladder)



Mã ngữ (Mnemonic)

LD	000.01
AND	000.02
OR	000.03
OUT	100.09
END (01)	

Hình 1.9. Ví dụ chương trình điều khiển dạng LAD.

Định dạng của chương trình tương tự mạch logic nối dây hình 1.7. Các biểu trưng (Symbol) biểu diễn lệnh, các số ở phía trên biểu diễn toán hạng (địa chỉ Bit). Ví dụ:



Biểu trưng mô tả tiếp điểm thường hở (NO – *Normaly Opened Contact*).



Biểu trưng mô tả cuộn dây rơ le (Coil).

Vì vậy, trên sơ đồ hình 1.9, “cuộn dây” 100.09 được “cấp điện” khi các tiếp điểm 000.01 và 000.02 ở trạng thái ON hoặc tiếp điểm 000.03 ở trạng thái ON. Về hình thức, “cuộn dây” 100.09 được “cấp điện” khi các điều kiện logic của dây thang (Rung) là liên tục từ trái sang phải. Nói cách khác, “mạch điện” phải được khép kín. Để mô tả hoạt động của PLC, trước hết cần làm rõ các thuật ngữ được sử dụng.

Thiết bị (Device) hay phần tử (Component) logic nối với PLC và gửi tín hiệu đến module vào gọi là thiết bị hay phần tử vào (Input Device, Input Component).

Tín hiệu mà thiết bị vào gửi đến PLC gọi là tín hiệu vào (Input). Trong trường hợp tín hiệu vào logic thì có hai trường hợp. Trường hợp một chiều, tín hiệu mức thấp là 0 V và mức cao là 24 VDC. Trường hợp xoay chiều, tín hiệu mức thấp là 0 V, mức cao là (110 ÷ 220) VAC.

PLC nhận tín hiệu vào qua các đầu nối (Terminals) có vị trí xác định trên module vào gọi là các điểm đầu vào (Input Points).

Trạng thái tín hiệu ở mỗi đầu nối được PLC đọc và lưu giữ trong một vùng nhớ dữ liệu gọi là vùng ảnh đầu vào (bao gồm các thanh ghi ảnh đầu vào). Mỗi đầu nối tương ứng với một Bit trong thanh ghi ảnh đầu vào. Địa chỉ của Bit là địa chỉ của đầu nối tương ứng. Khi tín hiệu vào ở mức cao thì Bit tương ứng có giá trị bằng 1. Khi tín hiệu vào ở mức thấp thì Bit tương ứng có giá trị bằng 0.

PLC thực hiện lần lượt từng lệnh của chương trình trong bộ nhớ. Kết quả thực hiện chương trình là các dữ liệu hoặc các quyết định gửi ra bên ngoài. Các dữ liệu được cất trong bộ nhớ để lưu giữ hoặc để làm điều kiện thực hiện tiếp theo. Các quyết định gửi ra bên ngoài ở dạng dữ liệu ra được gửi đến một vùng nhớ dữ liệu gọi là vùng ảnh đầu ra (bao gồm các thanh ghi ảnh đầu ra).

Thiết bị nối với PLC và nhận tín hiệu điều khiển từ module ra gọi là thiết bị ra (*Output Device*) hoặc cơ cấu chấp hành (*Actuator*).

PLC gửi tín hiệu điều khiển thiết bị ra thông qua các đầu nối (*Terminals*) có vị trí xác định trên module ra gọi là điểm đầu ra (*Output Point*).

Mỗi điểm đầu ra tương ứng với một Bit trong thanh ghi ảnh đầu ra. Địa chỉ của Bit là địa chỉ của điểm đầu ra tương ứng. Giá trị của Bit ảnh đầu ra quyết định trạng thái của thiết bị ra. Thiết bị ra logic có hai trạng thái: tích cực (*Active*) và không tích cực (*Inactive*). Trạng thái tích cực hay không tích cực của thiết bị do tính chất của thiết bị và do người sử dụng quy định.

Ở trạng thái hoạt động (*Run Mode*), PLC thực hiện đọc trạng thái tín hiệu vào, ghi vào vùng ảnh đầu vào, thiết lập trạng thái ảnh đầu ra trên cơ sở thực hiện chương trình. Quá trình được lặp lại liên tục gọi là quét vòng. Đó là nguyên tắc hoạt động của PLC.

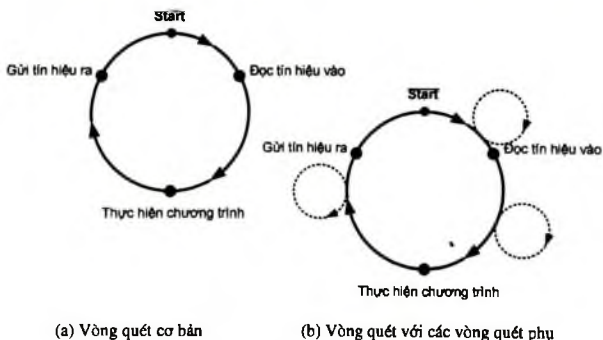
Trở lại ví dụ trên hình 1.9, khi SW1 và SW2 ở trạng thái ON, thì các Bit 000.01 và Bit 000.02 có giá trị 1. Kết quả Bit 100.09 có giá trị 1 và "cuộn dây" M ở trạng thái tích cực. Kết quả tương tự nếu S ở trạng thái ON, Bit 000.03 có giá trị 1. Biểu thức logic được thực hiện bằng chương trình như sau:

$$(100.09) = (000.01).(000.02) + (000.03)$$

Trong đó, ký hiệu (n) là giá trị logic của Bit có địa chỉ n.

1.4. QUÉT VÒNG – NGUYÊN TẮC HOẠT ĐỘNG CƠ BẢN CỦA PLC

Như trên đã trình bày, PLC hoạt động theo nguyên tắc quét vòng (*Scan*). Mỗi vòng quét (*Scan Cycle*) bao gồm ba giai đoạn cơ bản được trình bày trên hình 1.10a.



Hình 1.10. Sơ đồ vòng quét thực hiện chương trình của PLC.

Ở giai đoạn thứ nhất, PLC đọc trạng thái tín hiệu ở các module vào, gửi vào vùng ảnh đầu vào để làm dữ liệu thực hiện chương trình. Giai đoạn thứ hai là thực hiện chương trình trong bộ nhớ. Kết quả thực hiện chương trình là dữ liệu và các quyết định được lưu giữ trong bộ nhớ để phục vụ vòng quét sau hoặc gửi đến module ra. Ở giai đoạn thứ ba, PLC gửi dữ liệu đến vùng ảnh đầu ra và biến đổi thành tín hiệu điều khiển cơ cấu chấp hành nối với module ra. Khi đó, một vòng quét được hoàn thành, vòng quét tiếp theo bắt đầu và quá trình được thực hiện liên tục.

Quá trình đọc tín hiệu vào và gửi tín hiệu ra gọi là quá trình quét vào/ra. Quá trình thực hiện chương trình gọi là quét chương trình.

Thời gian để thực hiện một vòng quét gọi là chu kỳ quét. Chu kỳ quét có ảnh hưởng đến tốc độ xử lý của PLC và ảnh hưởng đến khả năng xử lý thời gian thực của PLC. Nói cách khác, việc sử dụng PLC trong các bài toán điều khiển chỉ được chấp nhận khi chu kỳ quét của PLC đủ nhỏ so với hằng số thời gian của hệ điều khiển. Khi đó, có thể chấp nhận xử lý đồng thời (thời gian thực) được thay thế bằng xử lý tuần tự. Ví dụ, tín hiệu vào thay đổi trạng thái hai lần trong một vòng quét thì PLC không thể phát hiện được. Khi đó, đáp ứng của PLC đối với sự thay đổi của đầu vào không chính xác nữa. Liên quan đến chu kỳ quét còn một số vấn đề cần thảo luận sau đây.

Chu kỳ quét phụ thuộc vào các nhân tố sau: tốc độ của bộ vi xử lý của CPU, độ dài chương trình, số lượng các đầu vào/ra. Ngoài ra, chu kỳ quét còn phụ thuộc vào một số các chu kỳ quét phụ (hình 1.10b) như: thời gian chuyển đổi song song – nối tiếp của hệ thống vào ra phân tán (*Remote I/O*), thời gian xử lý truyền thông nối tiếp, thời gian xử lý ngắt, thời gian đọc/ghi đầu vào/ra tương tự, thời gian thực hiện các chương trình kiểm tra, cảnh báo hệ thống... Tuy nhiên, đối với một hệ cụ thể thì các nhân tố, trừ tốc độ của bộ vi xử lý, đều là cố định. Vì vậy, để giảm chu kỳ quét thì phải chọn CPU có tốc độ xử lý cao. Đây là vấn đề cần lưu ý khi chọn thiết bị để xây dựng hệ điều khiển. Nguyên tắc hoạt động quét vòng của PLC hạn chế khả năng xử lý tức thời của PLC. Vì vậy, PLC chủ yếu được sử dụng trong các hệ điều khiển quá trình biến thiên chậm. Tuy nhiên, các PLC hiện đại đã được trang bị và tăng cường các tính năng xử lý ngắt ngày càng hoàn thiện để xử lý nhanh và kịp thời. Vì vậy, ranh giới giữa PLC và PC ngày càng thu hẹp.

Vấn đề xử lý vòng quét đầu tiên cần phải được quan tâm khi nghiên cứu nguyên tắc hoạt động quét vòng của PLC. Thật vậy, ở vòng quét đầu tiên, các dữ liệu đều chưa sẵn sàng, hệ đang ở quá trình khởi tạo. Đối với các hệ mà quá trình khởi tạo không ảnh hưởng đến quá trình điều khiển thì có thể bỏ qua. Ngược lại, các hệ thống khác cần lưu ý vòng quét này. Vì vậy, PLC đều cung cấp cờ trạng thái có giá trị bằng 1 ở vòng quét đầu tiên và bằng 0 ở các vòng quét khác, gọi là *First Scan Flag*. Người sử dụng có thể dùng cờ trạng thái này để tiến hành khởi tạo và thiết lập các điều kiện ban đầu cho hệ thống.

1.5. PLC VÀ PC

Về cấu trúc, PLC tương tự như máy tính số. Tuy nhiên, giữa PLC và máy tính số có sự khác nhau rất căn bản.

Thứ nhất, PLC được thiết kế để hoạt động trong môi trường công nghiệp rất khắc nghiệt với sự thay đổi lớn về độ ẩm, nhiệt độ và nhiễu mạnh.

Thứ hai, phần cứng và phần mềm của PLC được thiết kế để sử dụng và phù hợp với trình độ của người vận hành trực tiếp tại dây chuyền sản xuất. Phần cứng được chế tạo ở dạng các module tiêu chuẩn dễ lắp ráp, bảo dưỡng. Chương trình của PLC được biểu diễn một cách tiêu chuẩn không chính thức ở dạng giản đồ thang (LAD) rất trực quan và dễ sử dụng. Mỗi họ PLC có hệ điều hành riêng và chỉ sử dụng một ngôn ngữ lập trình do nhà sản xuất cung cấp. Vì vậy, không thể chạy chương trình của PLC họ này trên PLC họ khác. Trong khi đó, máy tính có thể cài đặt nhiều hệ điều hành, có thể sử dụng nhiều ngôn ngữ lập trình. Có thể sử dụng máy tính vai trò như PLC, thậm chí có thể chạy chương trình mô phỏng PLC trên máy tính. Máy tính được cài đặt phần mềm lập trình trở thành thiết bị lập trình cho PLC và điều khiển PLC. Máy tính có thể được sử dụng như thiết bị giao diện người – máy trong các hệ điều khiển mà PLC vừa là các bộ điều khiển, vừa là thiết bị thu nhập dữ liệu.

Thứ ba, máy tính là thiết bị tính toán phức tạp, có chức năng đa nhiệm (Multitask). Bộ nhớ của máy tính có thể chứa đồng thời nhiều chương trình. Trong khi đó, PLC chỉ thực hiện một chương trình được lưu giữ trong bộ nhớ RAM.

Thứ tư, PLC hoạt động theo nguyên tắc quét vòng, máy tính hoạt động theo nguyên tắc xử lý ngắt.

Về chức năng, PLC còn rất nhiều hạn chế so với máy tính. Các PLC hiện đại ngày được hoàn thiện về tốc độ tính toán, mở rộng bộ nhớ, tăng cường trang bị các chức năng tính toán và xử lý ngắt để thu hẹp khoảng cách so với máy tính. Vì vậy, các PLC hiện đại ngoài chức năng cơ bản là điều khiển logic, nó còn là một trạm tính toán đóng vai trò như các bộ điều khiển quá trình, điều khiển vị trí và xử lý thông tin.

1.6. PHÂN LOẠI PLC VÀ ỨNG DỤNG

PLC có rất nhiều chủng loại và do rất nhiều nhà sản xuất cung cấp. Một số nhà sản xuất và tích hợp hệ thống sử dụng PLC do chính họ chế tạo. Nó là một thành phần cấu thành hệ thống và được sử dụng trong phạm vi hẹp. Một số nhà sản xuất cung cấp PLC như là sản phẩm đa dụng cho người thiết kế và tích hợp hệ thống. Nhà sản xuất cung cấp thiết bị, phần mềm, hỗ trợ kỹ thuật và đào tạo để người sử dụng có điều kiện ứng dụng các sản phẩm này vào các hệ thống của mình. Có thể liệt kê một số hãng sản xuất điển hình là: SIEMENS (Đức), ALLEN- BRADLEY, GE-FUNUC (Mỹ), OMRON, MITSUBISHI, TOSHIBA (Nhật Bản)...

Do PLC được sử dụng rất rộng rãi từ các bài toán đơn giản đến các bài toán phức tạp, nên PLC được chế tạo thành nhiều loại khác nhau phù hợp với yêu cầu của

thực tế. Việc phân loại PLC dựa trên cơ sở khả năng (tốc độ xử lý, dung lượng bộ nhớ, số lượng đầu vào/ra) được chia thành các loại chính sau: loại nhỏ, loại vừa và loại lớn.

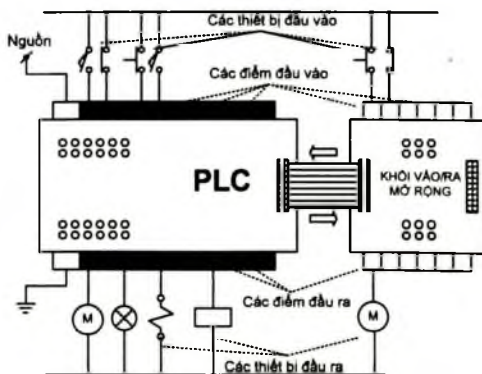
– PLC loại nhỏ có nhiều tên gọi khác nhau tùy thuộc các hãng chế tạo (*Small, Micro*), có dung lượng bộ nhớ dưới 2 KB, quản lý số điểm vào/ra dưới 128 và được sử dụng trong các ứng dụng đơn giản, yêu cầu ít điểm vào/ra.

– PLC cỡ vừa (*Medium*) có bộ nhớ đến 32 KB và quản lý số điểm vào/ra đến 2048. Cấu hình của hệ có thể sử dụng các module vào/ra đặc biệt để thực hiện các chức năng điều khiển quá trình và xử lý thông tin.

– PLC cỡ lớn (*Large*) là thiết bị phức tạp nhất, có thể quản lý đến 2 MB bộ nhớ và 16.000 điểm vào/ra. PLC loại này có ứng dụng không hạn chế từ điều khiển một quá trình công nghệ đến điều khiển một phân xưởng, một nhà máy.

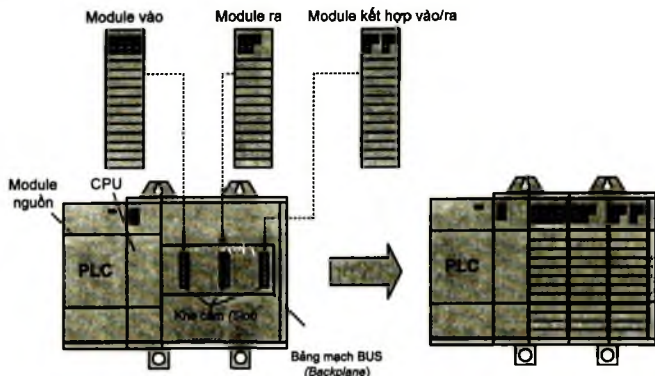
Phương pháp phân loại PLC ở trên kết hợp với kiểu dáng chế tạo sẽ đưa ra các chủng loại PLC sau đây.

Các PLC cỡ nhỏ thường được chế tạo ở dạng cố định (*Compact, Fixed*). Với loại này, nguồn cung cấp, CPU và một số điểm vào/ra được chế tạo trên cùng một khối không thể tách rời nhau (*Onboard*). Ưu điểm cơ bản của PLC loại này là giá thành thấp, nhỏ, gọn và thích hợp các ứng dụng nhỏ. Số các điểm vào/ra trên PLC theo tỷ lệ 3 : 2, ví dụ, loại 10 điểm (6 vào, 4 ra), loại 20 điểm (12 vào, 8 ra), loại 30 điểm (12 vào, 18 ra) và loại 48 điểm, 60 điểm... Khi cần thiết có thể sử dụng các module vào/ra mở rộng. Tuy nhiên, với PLC loại này ít khi sử dụng cách mở rộng như vậy. Nhược điểm chính là tính mềm dẻo không cao, tốc độ xử lý chậm, bộ nhớ nhỏ, hạn chế số điểm vào/ra. Sơ đồ tổ chức PLC loại nhỏ, dạng cố định được trình bày trên hình 1.11.



Hình 1.11. PLC loại nhỏ, dạng cố định.

Các PLC loại vừa và loại lớn được chế tạo ở dạng các module riêng biệt, có thể tháo, lắp dễ dàng (*Modular*). Các module cơ bản là: nguồn, CPU, vào/ra... Đây là cấu trúc tiêu chuẩn của PLC, đảm bảo cho PLC được sử dụng một cách mềm dẻo và người sử dụng có nhiều lựa chọn cho cấu hình của mình. Các module được lắp vào các khe cắm (*Slot*) trên bảng mạch Bus (*Bus Module, Backplane*). Sơ đồ tổ chức PLC dạng module được trình bày trên hình 1.12. Để mở rộng các điểm vào/ra, có thể sử dụng các bảng mạch Bus mở rộng.



Hình 1.12. PLC dạng Modular.

Ứng dụng của PLC được chia làm ba nhóm chính là: đơn nhiệm (*Single*), đa nhiệm (*Multitask*) và quản lý điều khiển (*Control Managment*). Ứng dụng đơn nhiệm là chỉ sử dụng một PLC duy nhất để điều khiển một quá trình kỹ thuật. Đó là một khối điều khiển độc lập, không có trao đổi thông tin với máy tính hoặc các PLC khác. Cấu hình của hệ có thể dùng PLC các loại nhỏ, vừa hoặc lớn. Ứng dụng đa nhiệm thường sử dụng PLC cỡ vừa để điều khiển một công đoạn của dây chuyền sản xuất hoặc để điều khiển một vài quá trình kỹ thuật với số lượng điểm vào/ra thích hợp.

Mỗi PLC có thể trở thành một nút trong hệ điều khiển phức tạp (ví dụ: hệ điều khiển DCS). Khi đó, yêu cầu có sự trao đổi dữ liệu, thông tin giữa các PLC với nhau, hoặc giữa PLC với các thiết bị khác (ví dụ: máy tính, trạm kỹ thuật...). Việc trao đổi dữ liệu, thông tin nhờ mạng LAN hoặc ETHERNET theo chuẩn công nghiệp. Thông thường, kết nối giữa các PLC với nhau theo kiểu LINK:LINK. Ứng dụng quản lý điều khiển sử dụng PLC để điều khiển một vài quá trình kỹ thuật khác nhau. Thông thường sử dụng các PLC cỡ lớn, với cấu hình của hệ là một mạng LAN điều khiển thống nhất, có sự trao đổi dữ liệu và thông tin giữa các thành phần của hệ. Trong đó, PLC đóng vai trò là bộ điều khiển, đồng thời quản lý hoạt động toàn bộ hệ là trạm chủ (*Master*).

Các PLC khác là các bộ điều khiển và đồng thời là thiết bị thu nhập dữ liệu phục vụ cho công tác quản lý và theo dõi hệ thống gọi trạm tớ (*Slave*). Kết nối thực hiện theo kiểu HOST:LINK.

1.7. CÁC HỌ PLC THÔNG DỤNG

1.7.1. Họ SIMATIC của SIEMENS (CHLB Đức)

Hãng SIEMENS đưa ra nhiều họ PLC khác nhau. Tuy nhiên, PLC họ SIMATIC được sử dụng phổ biến và có hai thế hệ S5 (cũ) và S7 (mới).

Các loại PLC họ S5 như sau:

- Loại nhỏ có S5 90U, S5 95U.
- Loại vừa có S5 100U với các CPU 100, 101, 102, 103, 104.
- Loại lớn có S5 115, S5 135, S5 155 với các CPU 941, 942,...

Họ S5 dùng phần mềm lập trình là STEP5.

Các loại PLC họ S7 như sau:

- Loại nhỏ có S7 200, với các CPU 212, 222, 226,...
- Loại vừa có S7 300, với các CPU 312, 315,...
- Loại lớn có S7 400 với các CPU 412, 415,...

Họ S7 dùng phần mềm lập trình gọi chung là STEP 7.

1.7.2. Họ SYSMAC của OMRON (Nhật Bản)

Hãng OMRON chế tạo các PLC gồm nhiều chủng loại, ví dụ, PLC series C, CV, CJ... Phần mềm lập trình có thể dùng Syswin hoặc CX-Programmer. Ở đây giới thiệu PLC series C gồm các loại sau:

- Loại nhỏ có CPM1, CPM1A, CPM2A;
- Loại vừa có CQM1, CQM1H,...
- Loại lớn có C200H, C1000H, C2000H, C2000HS...

1.7.3. PLC của ALLEN BRADLEY (Mỹ)

- Loại nhỏ có MicroLogix 500, MicroLogix 1000.
- Loại vừa có SLC 500 với CPU 01, 02, 03, 04.

PLC loại nhỏ và loại vừa dùng phần mềm lập trình RSLOGIX 500.

- Loại lớn có PLC5 với CPU 25, 41, 45, 55,... và dùng phần mềm lập trình RSLOGIX 5.

Trong thực tế, các loại PLC thế hệ mới đã và đang được đưa ra thị trường và ngày càng được sử dụng rộng rãi, nhanh chóng. Tài liệu về PLC thế hệ mới nhất có thể được cập nhật thông qua các Website của các hãng sản xuất.

Bên cạnh các họ PLC được giới thiệu ở trên, còn có rất nhiều họ PLC của các nhà sản xuất khác, như họ 9300 của GE- FUNUC, họ FX, MELSEC của MITSUBISHI, họ T1, T2, T3 của TOSHIBA, họ Master-K của LG...

Trên bảng 1.1 trình bày hình ảnh một số họ PLC thông dụng.

Bảng 1.1. Một số họ PLC thông dụng





C200H



CP1H

(b) PLC hãng OMRON



MicroLogix 500



MicroLogix 1000



MicroLogix 1500



SCL 500

(c) PLC hãng ALLEN BRADLEY

Việc lựa chọn họ PLC xuất phát từ yêu cầu thực tế của người sử dụng. Các căn cứ lựa chọn như sau:

– Thứ nhất là yêu cầu công nghệ và yêu cầu điều khiển. Điều này liên quan đến các tính năng kỹ thuật của PLC có đáp ứng được yêu cầu công nghệ và điều khiển đặt ra hay không, như tốc độ xử lý, dung lượng bộ nhớ, quản lý vào/ra và khả năng mở rộng hệ thống...

– Thứ hai là tính kinh tế và khả năng đáp ứng về thời gian cung cấp thiết bị của nhà sản xuất.

– Thứ ba là sự hỗ trợ và giúp đỡ của nhà cung cấp về các vấn đề giải pháp kỹ thuật, phần mềm, công cụ và đào tạo.

1.8. PHẦN MỀM VÀ CÔNG CỤ LẬP TRÌNH

Công cụ lập trình cho PLC (*Programmer*) có thể là các bộ lập trình chuyên dụng hoặc máy tính PC. Bộ lập trình chuyên dụng cũng có thể là bộ lập trình cầm tay hoặc bộ lập trình dạng máy laptop. Máy tính PC có cài đặt phần mềm có thể dùng làm bộ lập trình.

Phần mềm lập trình cho PLC là môi trường giao tiếp giữa người và máy. Nhờ đó, người sử dụng có thể soạn thảo chương trình cho PLC, điều khiển và theo dõi hoạt động của PLC thông qua thiết bị lập trình. Phần mềm lập trình cho PLC do nhà sản xuất cung cấp và chạy trên hệ điều hành của PC. Thông thường, mỗi họ PLC có phần mềm lập trình riêng, mặc dù cách biểu diễn chương trình tương tự nhau. Vì vậy, việc sử dụng PLC có liên quan chặt chẽ đến phần mềm của nó.

PLC họ S5 của hãng SIEMENS sử dụng phần mềm lập trình gọi là STEP5 với các phiên bản gốc chạy trên hệ điều hành DOS và hiện nay có các phiên bản chạy trên WINDOWS. PLC họ S7 của hãng SIEMENS sử dụng phần mềm lập trình có tên chung là STEP7 (STEP7-200, STEP7-300, STEP7-400).

PLC của hãng OMRON sử dụng phần mềm lập trình gọi là SYSWIN và hiện nay là CX-PROGRAMMER...

PLC của hãng ALLEN BRADLEY sử dụng phần mềm RSLOGIX 500 cho PLC loại nhỏ và loại vừa (Micrologix và SLC 500) và RSLOGIX5 cho PLC loại lớn (PLC 5).

Một số họ PLC và phần mềm lập trình tương ứng được tổng kết trong bảng 1.2.

Bảng 1.2. Một số họ PLC và phần mềm lập trình tương ứng

STT	Họ PLC	Phần mềm
1	PLC hãng OMRON	SYSWIN
		CX-PROGRAMMER
2	S5 (SIEMENS)	STEP5
3	S7 (SIEMENS)	STEP7
4	MicroLogix 1000 SLC 500	RSLogix 500
	(ALLEN BRADLEY)	

1.9. KẾT LUẬN

PLC là thiết bị được sử dụng rộng rãi trong các hệ tự động hóa quá trình sản xuất. Về chức năng, PLC có thể thực hiện các bài toán điều khiển một công đoạn hoặc toàn bộ một dây chuyền sản xuất. PLC có thể thực hiện điều khiển các đại lượng logic, điều khiển quá trình và quản lý điều khiển.

Vì vậy, việc nghiên cứu và sử dụng PLC là đòi hỏi và là yêu cầu cần thiết đối với kỹ sư, cán bộ kỹ thuật các ngành điện, điện tử, tự động hóa, thủy khí... Những kiến thức cơ bản cần thiết khi nghiên cứu PLC là: kỹ thuật vi xử lý, kỹ thuật số, điều khiển logic, kỹ thuật lập trình và tự động hóa quá trình công nghệ.

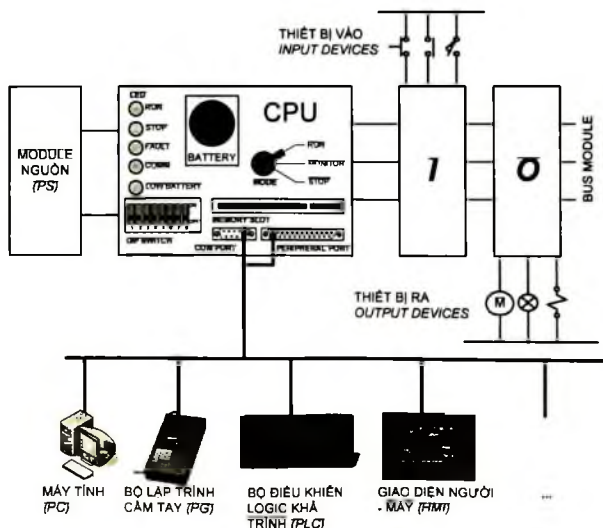
Chương 2

CÁC THÀNH PHẦN CỦA PLC

Cấu hình vật lý của hệ điều khiển dùng PLC được xây dựng trên cơ sở lắp ghép các khối chức năng của PLC. Các khối chức năng này được chế tạo ở dạng các module. Ví dụ, module nguồn cung cấp, module xử lý trung tâm (CPU), các module vào/ra số, các module vào/ra tương tự... Trong chương này sẽ trình bày chi tiết các thiết bị trên và các phương pháp cấu hình hệ điều khiển PLC.

2.1. SƠ ĐỒ KHỐI CHỨC NĂNG HỆ ĐIỀU KHIỂN PLC

Về tổng quát, sơ đồ khối chức năng của hệ điều khiển PLC được trình bày trên hình 2.1.



Hình 2.1. Sơ đồ khối chức năng hệ điều khiển PLC.

2.1.1. Khối nguồn cung cấp

Module nguồn có chức năng cung cấp nguồn một chiều cho các module khác trong hệ thống được lắp vào bảng mạch Bus. Thông thường, điện áp đầu ra của module nguồn là 24VDC. Một số PLC cỡ lớn có thể yêu cầu các cấp điện áp khác nhau như: $\pm 5VDC$, $\pm 15VDC$, 24VDC. Các cấp điện áp này có thể do module nguồn tạo ra hoặc được tạo ra trong khối CPU. Module nguồn là bộ nguồn ổn áp xung, dải rộng. Công suất đầu ra của module nguồn được lựa chọn tùy thuộc vào yêu cầu. Nhà sản xuất cung cấp nhiều loại module nguồn với công suất khác nhau. Lưu ý rằng, module nguồn chỉ cung cấp cho các module chức năng của hệ mà không cung cấp cho các thiết bị bên ngoài. Các thiết bị này được cung cấp từ nguồn ngoài. Vì vậy, khi lựa chọn công suất của module nguồn phải xuất phát từ số lượng và chủng loại các module chức năng của hệ.

2.1.2. Khối xử lý trung tâm (CPU)

Khối xử lý trung tâm CPU là bộ não của PLC, có chức năng nhận dữ liệu từ các module vào, thực hiện chương trình, đưa ra quyết định và điều khiển các cơ cấu chấp hành được nối vào module ra. CPU được chế tạo trên bảng mạch in gọi là bảng mạch chính (*Mainboard*) và được đóng gói thành module CPU. Các thành phần chính của CPU gồm: bộ vi xử lý, bộ nhớ, các mạch truyền thông, các mạch số phụ trợ và mạch biến đổi nguồn... Việc tìm hiểu khối CPU bao gồm: các thành phần chính bên trong và các thành phần bên ngoài.

Các thành phần bên trong khối CPU gồm có:

2.1.2.1. Bộ vi xử lý

Bộ vi xử lý là hạt nhân của CPU, quyết định tính năng của PLC: tốc độ xử lý, khả năng quản lý bộ nhớ và vào/ra. Các bộ vi xử lý được sử dụng có các loại 8 bit, 16 bit, 32 bit... Một số họ PLC sử dụng bộ vi xử lý tương tự như ở máy tính PC. CPU có thể có một hoặc nhiều bộ vi xử lý. Mỗi bộ vi xử lý thực hiện một nhiệm vụ khác nhau. Lợi ích của việc sử dụng nhiều bộ vi xử lý là tăng tốc độ xử lý của PLC. Ví dụ, bộ vi xử lý thực hiện điều khiển, bộ vi xử lý thực hiện truyền thông, bộ vi xử lý thực hiện các nhiệm vụ tính toán, xử lý tín hiệu phức tạp, bộ vi xử lý thực hiện các phép toán logic, các chức năng định thời gian, các chức năng đếm và theo dõi các trạng thái của hệ thống khi hoạt động.

2.1.2.2. Bộ nhớ

Bộ nhớ là thiết bị để lưu giữ thông tin: chương trình, dữ liệu, tham số hệ thống, cấu hình hệ thống. Việc tổ chức và quản lý bộ nhớ do hệ điều hành đảm nhiệm và sẽ được trình bày kỹ trong chương 3.

Trong phần này sẽ nêu các tính chất và đặc điểm của bộ nhớ được sử dụng trong PLC. Theo tính chất, có thể chia bộ nhớ thành hai loại: bộ nhớ duy trì (*Non-Volatile*) và bộ nhớ không duy trì (*Volatile*). Nội dung của bộ nhớ không duy trì sẽ bị mất khi ngắt nguồn cung cấp. Ưu điểm của bộ nhớ không duy trì là khả năng ghi/đọc

tốc độ cao. Trong trường hợp cần thiết, có thể dùng nguồn cung cấp ổn định là pin (*Battery*) làm nguồn nuôi cho bộ nhớ này. Bộ nhớ không duy trì được sử dụng làm bộ nhớ chính cho PLC để lưu giữ chương trình và dữ liệu đang thực hiện. Việc truy nhập (đọc/ghi) bộ nhớ không duy trì có thể được thực hiện thông qua chương trình ứng dụng. Bộ nhớ duy trì có khả năng lưu giữ thông tin khi mất nguồn cung cấp. Tuy nhiên, tốc độ ghi thông tin vào bộ nhớ duy trì không cao và phải sử dụng thiết bị hoặc chương trình đặc biệt. Bộ nhớ duy trì được sử dụng để lưu giữ các thông tin ít thay đổi, dữ liệu cần được bảo vệ khi mất nguồn. Ví dụ: chương trình đã chạy ổn định, tham số đặt, các trạng thái đặc biệt của hệ thống, cấu hình hệ thống, hệ điều hành...

Việc ghi thông tin vào bộ nhớ gọi là viết (*Write*) và nhận thông tin từ bộ nhớ gọi là đọc (*Read*). Việc đọc/ghi bộ nhớ gọi là truy nhập bộ nhớ. Thông tin lưu giữ trong bộ nhớ ở dạng số nhị phân (0, 1). Mỗi số 0 hoặc 1 gọi là Bit thông tin (*Binary Digit*). Về tổ chức, bộ nhớ gồm các ô nhớ được sắp xếp liên tiếp nhau về mặt logic. Mỗi ô nhớ có hai tham số là: nội dung của ô nhớ và địa chỉ của ô nhớ trong bộ nhớ. Nội dung của ô nhớ, viết là (M), chứa một từ nhị phân N Bit. Trong đó, N gọi là kích thước của ô nhớ. Thông thường, kích thước của ô nhớ vật lý là 8 Bit hoặc 16 Bit. Cần phân biệt kích thước của ô nhớ và kích thước của một từ nhị phân (*Binary Word*). Một từ nhị phân có thể có kích thước bằng 1, 2 hoặc 4 lần kích thước ô nhớ vật lý tùy thuộc kiểu dữ liệu, nghĩa là, nó được lưu giữ trong 1, 2 hoặc 4 ô nhớ. Về nguyên tắc, hệ thống quản lý dữ liệu ở dạng từ (*Word*). Địa chỉ ô nhớ, viết là M, là vị trí của ô nhớ trong bộ nhớ. Hệ thống quản lý bộ nhớ thông qua địa chỉ. Để xác định địa chỉ của ô nhớ, có thể dùng địa chỉ vật lý hoặc địa chỉ quy ước. Hai ô nhớ liên tiếp nhau là hai ô nhớ có địa chỉ hơn kém nhau 1. Địa chỉ vật lý là vị trí của ô nhớ so với địa chỉ gốc của bộ nhớ. Thông thường, quy ước địa chỉ gốc là 0. Địa chỉ vật lý thường được biểu diễn ở dạng số HEXA. Địa chỉ quy ước do hệ điều hành quy định. Người sử dụng truy nhập đến bộ nhớ bằng địa chỉ quy ước thông qua hệ điều hành. Địa chỉ của ô nhớ được xác định trên Bus địa chỉ (*Address Bus*). Bus địa chỉ kích thước N Bit thì có thể địa chỉ hóa 2^N ô nhớ, gọi là không gian nhớ. Vì vậy, Bus địa chỉ quy định dung lượng của bộ nhớ và số điểm vào, ra của PLC. Việc đọc/ghi nội dung ô nhớ thông qua Bus dữ liệu (*Data Bus*). Thông thường, kích thước của Bus dữ liệu bằng kích thước của ô nhớ. Việc điều khiển quá trình truy nhập bộ nhớ được thực hiện thông qua các tín hiệu điều khiển trên Bus điều khiển (*Control Bus*). Về tổ chức, bộ nhớ được chia thành các vùng (*Areas*), mỗi vùng lưu giữ một kiểu thông tin nhất định. Ví dụ: vùng nhớ mã chương trình gọi là bộ nhớ chương trình (*Program Memory*), vùng nhớ dữ liệu gọi là bộ nhớ dữ liệu (*Data Memory*). Hình ảnh của bộ nhớ với các vùng nhớ gọi là bản đồ bộ nhớ (*Memory Map*). Đánh giá bộ nhớ trên cơ sở các tham số sau: dung lượng (*Capacity*) và thời gian truy nhập (*Access Time*). Dung lượng bộ nhớ là số lượng thông tin mà bộ nhớ có khả năng lưu giữ được và được tính bằng Bit, Byte, Kbit, Kbyte. Tuy nhiên, đánh giá dung lượng bộ nhớ của PLC được chia thành hai thành phần: dung lượng bộ nhớ chương trình tính bằng K lệnh (*K Instructions*) và dung lượng bộ nhớ dữ liệu tính bằng K từ (*K Words*).

1 K lệnh = 1024 lệnh cơ bản của PLC

1 K từ = 1024 từ dữ liệu

Tốc độ truy nhập bộ nhớ được đánh giá qua thời gian truy nhập. Nếu dung lượng bộ nhớ đặc trưng cho khả năng của PLC giải các bài toán phức tạp với chương trình lớn và dữ liệu nhiều thì tốc độ truy nhập đóng vai trò quan trọng đối với tính tác động nhanh của PLC. Trong thực tế, bộ nhớ trong (*Built in Memory*) của CPU chỉ có dung lượng nhất định, nhỏ hơn không gian nhớ để đảm bảo tính kinh tế và tính thực tiễn. Trong trường hợp yêu cầu dung lượng bộ nhớ lớn hơn dung lượng bộ nhớ trong, người sử dụng có thể trang bị thêm thẻ nhớ mở rộng (*Memory Card*) với dung lượng tùy chọn. Bộ nhớ của PLC được xây dựng trên cơ sở hai loại bộ nhớ: bộ nhớ cố định và bộ nhớ đọc/ghi. Bộ nhớ cố định có các loại: ROM, EPROM, EEPROM và FLASH ROM. Bộ nhớ cố định là bộ nhớ duy trì, nội dung của bộ nhớ không bị mất khi ngắt nguồn cung cấp. Vì vậy, bộ nhớ cố định được sử dụng để lưu giữ chương trình, dữ liệu không thay đổi. Do đặc điểm của bộ nhớ loại EEPROM và FLASH ROM có thể xóa bằng điện, nên nó còn được sử dụng làm một phần của bộ nhớ trong của CPU để lưu giữ các dữ liệu, thông tin, trạng thái của hệ thống đang hoạt động khi mất nguồn cung cấp. Ví dụ, giá trị của bộ đếm, cờ lỗi, dữ liệu duy trì (*Retentive Data*). Bộ nhớ đọc/ghi (*RAM*) là bộ nhớ không duy trì. Nội dung của bộ nhớ bị mất khi ngắt nguồn cung cấp và có thể đọc/ghi dễ dàng với tốc độ cao. Bộ nhớ RAM được sử dụng như bộ nhớ hoạt động của PLC để lưu giữ chương trình và dữ liệu đang thực hiện. Bộ nhớ RAM có các loại sau: SRAM (*Static RAM*) và DRAM (*Dynamic RAM*). Để duy trì được thông tin trong RAM khi mất nguồn cung cấp thì phải sử dụng nguồn nuôi cố định là pin (*Battery*). Một số họ PLC dùng tụ điện dung tích lớn để làm nguồn nuôi cho bộ nhớ CMOS RAM.

2.1.2.3. Các mạch phụ trợ

Trên bảng mạch CPU còn có các mạch phụ trợ như: mạch tạo các nguồn cung cấp với các mức điện áp khác nhau: +5V, -5V, +12V, -12V; mạch ghép nối các cổng song song, nối tiếp...

Các thành phần bên ngoài nằm trên khối CPU gồm có:

2.1.2.4. Hệ thống các đèn LED chỉ thị trạng thái (*Status Led*)

Đây là các cờ (*Flag*) trạng thái của hệ thống mà PLC lập trong quá trình hoạt động. Ví dụ một số các LED quan trọng là: RUN, STOP, FAULT, COMM, LOW BATTERY...

RUN LED báo PLC ở chế độ đang thực hiện chương trình trong bộ nhớ của PLC.

STOP LED báo PLC ở trạng thái dừng, không thực hiện chương trình. Các trạng thái ở đầu ra đều bằng 0.

FAULT LED báo hệ thống có lỗi. Việc xác định lỗi cụ thể phải dùng chương trình. Nếu lỗi nhẹ (*non-Fatal*) thì chương trình vẫn tiếp tục thực hiện, RUN LED vẫn sáng. Nếu lỗi nặng (*Fatal*) thì chương trình dừng thực hiện, STOP LED sáng.

COMM LED báo việc kết nối giữa PLC và các thiết bị ngoại vi đang được thực hiện.

LOW BATTERY LED báo nguồn cung cấp BATTERY đã yếu, cần phải thay thế. Việc thay BATTERY phải được thực hiện trong khi nguồn cung cấp cho PLC vẫn được duy trì.

2.1.2.5. Các chuyển mạch đặt chế độ hoạt động của PLC

Các chuyển mạch này ở dạng DIP SWITCH có N chuyển mạch. Mỗi chuyển mạch có hai trạng thái. Vì vậy, có thể thiết lập 2^N chế độ hoạt động. Mỗi lần khởi động, CPU đọc trạng thái của DIP SWITCH và thiết lập chế độ hoạt động cho PLC. Việc thiết lập chế độ hoạt động cho PLC có thể được thực hiện bằng phần mềm. Một số các chế độ hoạt động của PLC được thiết lập bằng DIP SWITCH như sau:

- Mỗi lần khởi động, chương trình được nạp từ bộ nhớ ngoài vào bộ nhớ trong của PLC.
- Thiết lập chế độ sử dụng bộ nhớ ngoài.
- Đặt điều kiện cho phép/không cho phép hoạt động (*Enable/Disable*) của một số ngắt.

2.1.2.6. Chuyển mạch chọn chế độ làm việc của PLC

Chuyển mạch chọn chế độ làm việc của PLC – gọi là MODE SWITCH – có ba vị trí tương ứng với ba chế độ làm việc: RUN, MONITOR và STOP.

Vị trí RUN có tác dụng:

- + Đặt PLC vào chế độ chạy, thực hiện chương trình trong bộ nhớ.
- + Không cho phép sửa, thay đổi chương trình và dữ liệu trong bộ nhớ.
- + Không cho phép thay đổi chế độ hoạt động của PLC bằng thiết bị lập trình hoặc từ màn hình giao diện.

Vị trí MONITOR (còn gọi là chế độ REM – Remote) có tác dụng:

- + Đặt PLC vào chế độ điều khiển từ xa. Các chế độ có thể chọn từ xa là: REMOTE RUN, REMOTE PROGRAM, REMOTE TEST.

- + Cho phép theo dõi, thay đổi chương trình và dữ liệu trong bộ nhớ.
- + Cho phép thay đổi chế độ hoạt động của PLC bằng thiết bị lập trình, màn hình giao diện.

Vị trí STOP (còn gọi là chế độ PROGRAM) có tác dụng:

- + Đặt PLC vào chế độ dừng, không thực hiện chương trình.
- + Cho phép nạp chương trình vào PLC (*Download*) từ thiết bị lập trình và cho phép soạn thảo trực tiếp chương trình trong PLC (*Online Edit*).

+ Không cho phép thay đổi chế độ hoạt động của PLC bằng thiết bị lập trình hoặc từ màn hình giao diện.

Ở một số họ PLC, chuyển mạch chọn chế độ hoạt động chỉ có hai vị trí: RUN và STOP. Trong trường hợp này, vị trí STOP có cả hai chức năng REM và PROGRAM.

2.1.2.7. Ổ cắm pin (Battery)

Pin (3,6 V, 5 V...) được sử dụng làm nguồn cung cấp cố định cho bộ nhớ duy trì. Vai trò của pin là duy trì nội dung một phần bộ nhớ của PLC khi ngắt nguồn cung cấp. Đó là:

+ Chương trình trong bộ nhớ RAM.

+ Dữ liệu của chương trình như: giá trị đặt, giá trị các bộ đếm, các kết quả tính toán...

+ Các trạng thái đặc biệt khi thực hiện chương trình. Các trạng thái này cần phải được duy trì khi nguồn cung cấp mất đột ngột, là điều kiện để thực hiện chương trình khi nguồn cung cấp có trở lại. Ví dụ, các trạng thái báo lỗi, trạng thái chỉ hướng, chiều chuyển động của thiết bị, các vị trí của cơ cấu chấp hành...

+ Các tham số thiết lập cấu hình của hệ thống. Ví dụ, chọn chế độ hoạt động cho các cổng truyền thông nối tiếp, đặt chế độ làm việc đặc biệt của PLC theo yêu cầu của người sử dụng...

+ Thực hiện thời gian thực của hệ thống (*Real Time*).

Vì vai trò quan trọng của pin và thời gian sống của pin có hạn, nên khi hết pin thì hệ thống lập cờ Low Battery để báo hiệu cần phải thay pin mới. Việc thay pin phải được tiến hành khi có nguồn cung cấp.

Một số loại PLC cỡ nhỏ sử dụng tụ dung lượng lớn (*Super Capacity*) thay thế cho pin. Trong điều kiện tiêu chuẩn, thời gian duy trì nội dung bộ nhớ có thể đạt được vài chục ngày sau khi ngắt nguồn.

Để kéo dài tuổi thọ của pin, một số loại PLC mới còn dùng bộ nhớ EEPROM để làm bộ nhớ duy trì. Khi ngắt nguồn thì nội dung của bộ nhớ EEPROM không bị mất. Tuy nhiên, việc ghi và xóa bộ nhớ EEPROM phải được thực hiện bằng chương trình và thiết bị riêng được tích hợp trong PLC.

2.1.2.8. Khe cắm bộ nhớ mở rộng (Memory Slot)

Khe cắm bộ nhớ mở rộng dùng để lắp các thẻ nhớ (*Memory Card*) bên ngoài. Đó là:

+ Thẻ nhớ RAM hoặc EEPROM dùng để mở rộng bộ nhớ hoạt động của PLC.

+ Thẻ nhớ ngoài EPROM hoặc EEPROM lưu giữ chương trình điều khiển đã hoàn thiện. Chương trình sẽ được nạp vào bộ nhớ trong một cách tự động khi khởi động máy hoặc bằng một vài thao tác đơn giản. Đây là phương pháp lưu giữ chương trình an toàn và hiệu quả.

Thẻ nhớ ngoài có các loại 8 KB, 16 KB, 32 KB...

2.1.2.9. Cổng giao tiếp song song (Parallel Communication Port)

CPU thực hiện trao đổi dữ liệu với các module vào/ra bằng phương pháp song song thông qua hệ thống Bus. Cổng giao tiếp song song được thực hiện bằng các mạch ghép nối giữa CPU và hệ thống Bus.

2.1.2.10. Cổng nối tiếp (Serial Communication Port)

CPU thực hiện trao đổi thông tin với các thiết bị ngoại vi thông qua cổng nối tiếp bằng phương pháp không đồng bộ. Các thiết bị ngoại vi gồm có: bộ lập trình PG (*Programmer*), máy tính PC (*Personal Computer*), giao diện người máy HMI (*Human Machine Interface*), bảng vận hành OP (*Operator*) và các thiết bị ngoại vi khác. Trên khối CPU có một hoặc nhiều cổng nối tiếp. Tuy nhiên, mỗi họ PLC đều chọn một tiêu chuẩn cổng nối tiếp cụ thể làm cổng chuẩn của thiết bị. Ví dụ, hãng SIEMENS chọn tiêu chuẩn RS485, hãng OMRON chọn tiêu chuẩn RS422 và hãng AB chọn cổng DH 485. Vì vậy, nếu trên khối CPU chỉ có một cổng nối tiếp thì chính là cổng chuẩn của hãng. Nếu trên khối CPU có hai cổng thì cổng thứ hai thường là cổng chuẩn RS232... Các PLC thế hệ mới có sử dụng cổng chuẩn USB. Thực hiện ghép nối giữa các cổng nối tiếp có chuẩn khác nhau nhờ các bộ chuyển đổi (*Converter, Adapter*). Ví dụ, bộ chuyển đổi RS485/RS232, bộ chuyển đổi RS422/RS232...

2.1.3. Các thiết bị vào/ra

Các thiết bị vào/ra thực hiện ghép nối giữa CPU và thế giới bên ngoài. Các thiết bị vào/ra bao gồm Bus module và các module vào/ra riêng biệt.

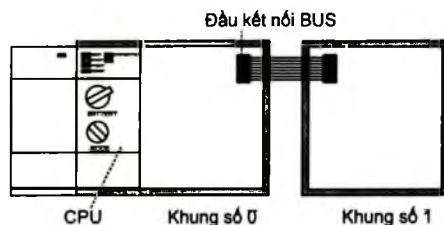
2.1.3.1. Bus module và hệ thống Bus (Bus System)

Bus module là thiết bị để ghép nối giữa CPU và các module vào/ra. Về vật lý, Bus module gồm khung kim loại vững chắc (*Chassis*) đỡ bằng mạch Bus (*Backplane*), trên đó là Bus hệ thống và các khe cắm (*Slot*) cho các module chức năng. Bus hệ thống gồm có: Bus dữ liệu (*Data Bus*), Bus địa chỉ (*Address Bus*), Bus điều khiển (*Control Bus*) và Bus nguồn (*Power Supply Bus*). Bus module có nhiều kích cỡ tùy thuộc vào số khe cắm. Ví dụ, loại 4, 7, 10, 16 khe cắm. Trên Bus module còn có các đầu nối (*Terminals*) để nối với các thiết bị bên ngoài. Cấu hình cụ thể của hệ điều khiển PLC có thể dùng một hoặc nhiều Bus module tùy thuộc vào số các module chức năng. Khi đó, các Bus module được nối với nhau bằng đầu nối (*Bus Connector*), được trình bày trên hình 2.2, gọi là hệ thống Bus (*Bus System*). Bus module chứa CPU gọi là Bus module chính, có địa chỉ 0 và các Bus module khác gọi là các Bus module mở rộng, có địa chỉ 1, 2, 3,...

Khi thiết lập cấu hình của hệ điều khiển, người lập trình phải khai báo thành phần của hệ thống Bus gồm: số lượng, loại Bus module, địa chỉ...

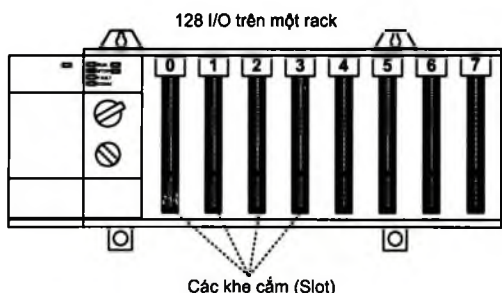
Về logic, Bus module được tổ chức ở dạng đơn vị logic (gọi là *Logic Rack*) để hệ thống quản lý các module vào/ra được gắn trên Bus module. Mỗi đơn vị logic có 128 đầu vào và 128 đầu ra logic.

Tùy thuộc vào CPU, mỗi đơn vị logic gồm một số từ (Word) dữ liệu trong vùng ảnh đầu vào và vùng ảnh đầu ra. Ví dụ, với CPU 8 Bit, mỗi đơn vị logic gồm 16 Byte ảnh đầu vào và 16 Byte ảnh đầu ra.



Hình 2.2. Sơ đồ hệ thống Bus.

Với CPU 16 Bit, mỗi đơn vị logic có 8 Word (16 Bit) ảnh đầu vào và 8 Word (16 Bit) ảnh đầu ra. Trên một Bus module có thể chứa một hoặc nhiều đơn vị logic và ngược lại, một đơn vị logic có thể bao gồm một vài Bus module tùy thuộc vào loại module vào/ra được sử dụng. Trên hình 2.3 trình bày sơ đồ tổ chức của hệ thống Bus.



Hình 2.3. Sơ đồ tổ chức của hệ thống Bus.

Bus module và hệ thống Bus của các họ PLC hiện đại và các loại PLC cỡ nhỏ có thể được tích hợp trong các module chức năng. Khi đó, việc ghép nối các module chức năng được thực hiện thông qua các đầu (Jack) nối trên module và thanh đỡ (Rail).

2.1.3.2. Các module vào/ra (Input Module, Output Module)

Các module vào/ra là các thiết bị mà qua đó CPU trao đổi dữ liệu với thế giới bên ngoài. Các module vào nhận tín hiệu từ các thiết bị vào, biến đổi thành dữ liệu gửi đến CPU. Các module ra nhận dữ liệu từ CPU, biến đổi thành tín hiệu điều khiển các thiết bị ra. Do nguồn tín hiệu vào và các thiết bị ra rất đa dạng về chủng loại, nên các

module vào/ra có rất nhiều loại. Ví dụ, module vào/ra rời rạc, module vào/ra tương tự, module vào/ra đặc biệt... Các module vào/ra được chế tạo theo chuẩn và ghép nối với CPU qua các khe cắm trên Bus module.

Việc trao đổi dữ liệu giữa CPU và các module vào/ra bằng phương pháp song song nhờ thao tác đọc/ghi. Mỗi lần trao đổi (đọc hoặc ghi) một từ dữ liệu (8 Bit hoặc 16 Bit).

Hệ thống quản lý các đầu vào/ra theo địa chỉ. Địa chỉ này được xác định trên cơ sở vị trí của khe cắm, kiểu module vào/ra. Vấn đề này sẽ được trình bày ở phần sau.

2.1.4. Các thiết bị ngoại vi (*Peripheral Devices*)

Các thiết bị ngoại vi trao đổi thông tin với CPU qua các cổng nối tiếp. Các thiết bị ngoại vi điển hình được giới thiệu sau đây.

2.1.4.1. Bộ lập trình chuyên dụng PG (*Programmer*)

Bộ lập trình chuyên dụng là thiết bị dùng để lập chương trình cho PLC, trao đổi dữ liệu với PLC và điều khiển hoạt động của PLC. Ví dụ, soạn thảo chương trình, nạp chương trình vào PLC, theo dõi hoạt động của chương trình, phát hiện và sửa lỗi... Bộ lập trình là một máy tính chuyên dụng được cài đặt phần mềm lập trình và trang bị các công cụ để giao tiếp với PLC. Bộ lập trình có các loại sau:

a) Bộ lập trình cầm tay (*Hand Held Programmer, Console*) là bộ lập trình gọn, nhẹ, tiện lợi cho người sử dụng khi cài đặt, thay đổi tham số, lệnh của chương trình của PLC đang hoạt động tại dây chuyền. Một số bộ lập trình cầm tay được trình bày trên hình 2.4.



CQM1-PRO01-E của OMRON

1761-HHP-B30 của ALLEN BRADLEY

Hình 2.4. Các bộ lập trình cầm tay.

Sử dụng bộ lập trình cầm tay có nhược điểm là màn hình hiển thị LCD nhỏ, giao tiếp người – máy không thuận tiện, thông tin hiển thị ở dạng ký tự và thao tác phức tạp.

b) Bộ lập trình chuyên dụng có màn hình CRT tương tự máy tính laptop. Thông tin hiển thị trên màn hình lớn, ở dạng ký tự hoặc đồ họa tăng tính trực quan và dễ giao tiếp. Do ưu điểm có bộ nhớ lớn, tốc độ xử lý nhanh nên bộ lập trình này có nhiều chức năng điều khiển, giao tiếp và xử lý thông tin từ PLC tương tự như máy tính cá nhân. Một số bộ lập trình được trình bày trên hình 2.5.



Sản phẩm của ALLEN BRADLEY



PG685 của SIEMENS

Hình 2.5. Các bộ lập trình chuyên dụng.

2.1.4.2. Máy tính cá nhân PC (Personal Computer)

Máy tính PC là thiết bị ngoại vi được sử dụng rất rộng rãi do ưu điểm về tốc độ xử lý, dung lượng bộ nhớ lớn và giá thành rẻ. Máy tính PC được sử dụng với các chức năng sau:

- Máy tính PC được cài đặt phần mềm lập trình trở thành bộ lập trình cho PLC rất thuận tiện vì sử dụng hệ điều hành và các công cụ phần mềm thông dụng.
- Máy tính PC đóng vai trò như trạm thu nhập dữ liệu từ PLC, xử lý thông tin, tham gia vào các hệ thống điều khiển, vận hành, quản lý hệ thống.
- Máy tính PC được cài đặt chương trình mô phỏng PLC đóng vai trò như PLC ảo phục vụ cho việc đào tạo và thiết kế hệ thống với chi phí đầu tư rẻ.

2.1.4.3. Các thiết bị giao diện người – máy HMI (Human Machine Interface)

Các thiết bị giao diện người – máy nối với PLC gọi là các trạm vận hành hoặc giao diện vận hành hệ thống (*Work Station, Operation Interface*). Các thiết bị này là một máy tính chuyên dụng, màn hình LCD đen trắng hoặc màu. Người sử dụng có thể tạo lập các giao diện để vận hành và theo dõi hoạt động của hệ thống như: Start, Stop, thay đổi tham số chương trình, hiển thị giá trị, trạng thái các biến, quản lý lỗi, theo dõi và thực hiện các thao tác cần thiết tác động lên hệ thống.

Các nhà sản xuất PLC thường cung cấp các thiết bị giao diện phù hợp với chuẩn của hãng. Ví dụ, hãng SIEMENS có các giao diện OP, hãng AB có giao diện PanelView và OMRON có giao diện NT. Tuy nhiên, một số hãng chuyên sản xuất các giao diện và phần mềm có thể kết nối với hầu hết các họ PLC, ví dụ sản phẩm PROFACE của hãng DIGITAL.

Việc kết nối các PLC với nhau tạo thành mạng PLC để thực hiện điều khiển một quá trình công nghệ cũng được thực hiện thông qua cổng nối tiếp. Các PLC trong mạng thực hiện trao đổi dữ liệu trong quá trình thực hiện chương trình thông qua một vùng nhớ có chung địa chỉ.

2.1.4.4. Các thiết bị ngoại vi khác

Các thiết bị ngoại vi khác có thể kết nối với PLC thông qua cổng nối tiếp để thực hiện các ứng dụng cụ thể như: bộ đọc mã vạch (*Barcode Reader*), bộ xử lý video, các thiết bị kiểm soát (*Access Control*)...

Quá trình trao đổi thông tin qua cổng nối tiếp được thực hiện theo hai phương thức: chủ – tớ (*HOST-LINK*) và *LINK-LINK*.

2.2. CÁC MODULE VÀO/RA

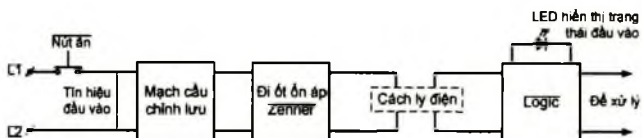
2.2.1. Module vào/ra rời rạc (*Discrete I/O module*)

Module vào/ra rời rạc, còn gọi là module vào/ra số (*Digital I/O module*), là loại module phổ biến nhất, là thành phần cấu hình cơ bản và phong phú nhất của PLC. Module vào/ra rời rạc là cổng giao tiếp với các thiết bị vào và thiết bị ra kiểu ON/OFF.

2.2.1.1. Module vào rời rạc

Module vào rời rạc thực hiện các nhiệm vụ nhận tín hiệu từ các thiết bị vào, biến đổi thành tín hiệu số gửi đến CPU. Module rời rạc là thiết bị giao tiếp giữa PLC và thiết bị vào. Sơ đồ nối thiết bị vào với module vào rời rạc được trình bày trên hình 2.6.

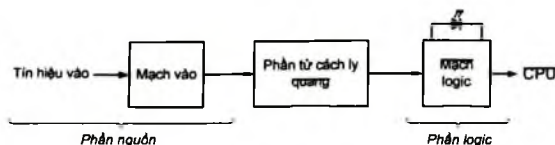
Các thiết bị vào rời rạc là các thiết bị logic như: chuyển mạch (*Selector Switch*), phím nhấn (*Push Button*), công tắc hành trình (*Limit Switch*), các tiếp điểm đầu ra của các bộ điều khiển (*Contact*), các loại cảm biến tiệm cận (*Proximity Sensor*), cảm biến quang (*Photo Sensor*)...



Hình 2.6. Sơ đồ nối thiết bị vào với module vào rời rạc.

Mỗi thiết bị vào nối với module vào tại một điểm có vị trí xác định gọi là điểm đầu vào (*Input Point*). Mỗi điểm đầu vào tương ứng với một địa chỉ của Bit dữ liệu trong vùng ảnh đầu vào. Giá trị của Bit dữ liệu phản ánh trạng thái của tín hiệu vào. Nếu tín hiệu vào ở mức cao (ví dụ, đối với tín hiệu một chiều là 24VDC, đối với tín hiệu xoay chiều là 110VAC hoặc 220VAC) thì Bit tương ứng có giá trị bằng 1. Nếu tín hiệu vào ở mức thấp (0V) thì Bit tương ứng có giá trị bằng 0.

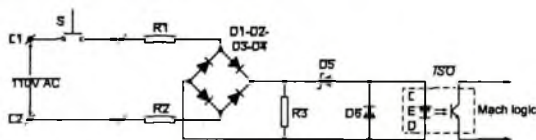
Trên các module vào đều có LED chỉ thị trạng thái của tín hiệu. Mạch điện của khối CPU và mạch ngoài được cách ly bằng phần tử quang (*Optocoupler*). Sơ đồ khối chức năng của module vào rời rạc được trình bày trên hình 2.7.



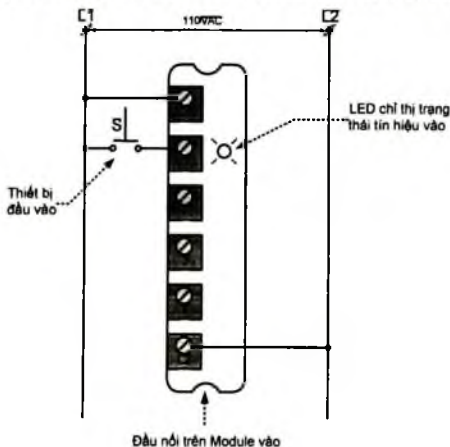
Hình 2.7. Sơ đồ khối chức năng của module vào rời rạc.

Sơ đồ chia làm hai phần: phần nguồn cung cấp cho các thiết bị vào và phần logic tạo ra tín hiệu phù hợp với CPU. Vì vậy, tín hiệu vào có thể là xoay chiều, một chiều với các mức điện áp khác nhau. Mạch vào là các mạch biến đổi tín hiệu, mạch lọc nhiễu. Phần tử cách ly thường dùng là phần tử cách ly quang hoặc biến áp xung. Mạch logic tạo ra tín hiệu logic phù hợp với CPU và LED chỉ thị trạng thái của tín hiệu vào.

Sơ đồ nguyên lý một đầu vào của module vào xoay chiều với điện áp 110 VAC – 220 VAC được trình bày trên hình 2.8.



(a) Sơ đồ nguyên lý một đầu vào của module vào xoay chiều

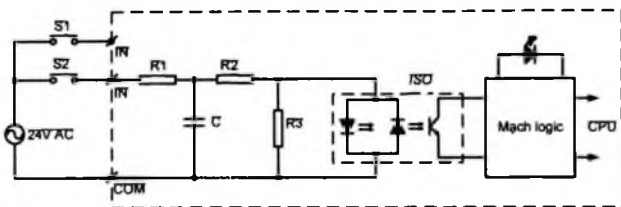


(b) Sơ đồ nối dây một đầu vào của module vào xoay chiều

Hình 2.8. Sơ đồ nguyên lý và sơ đồ nối dây một đầu vào của module vào xoay chiều.

Tín hiệu vào nhận được từ nguồn xoay chiều L1–L2 qua phím S. Tín hiệu xoay chiều biến đổi thành tín hiệu một chiều nhờ bộ chỉnh lưu (D1– D4). Điện trở R1, R2 hạn chế dòng vào, R3 là điện trở tải của bộ chỉnh lưu. Diode ổn áp D5 xác định ngưỡng nhỏ nhất của tín hiệu vào. Phần tử cách ly quang ISO chuyển đổi tín hiệu từ nguồn thành tín hiệu logic gửi đến CPU.

Hình 2.9 trình bày sơ đồ nguyên lý một đầu vào của module vào xoay chiều, điện áp 24VAC.



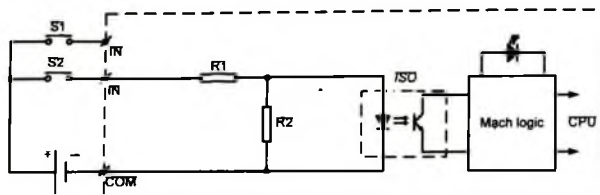
Hình 2.9. Sơ đồ nguyên lý và sơ đồ nối một đầu vào của module vào xoay chiều, điện áp 24VAC.

Sơ đồ này không sử dụng mạch chỉnh lưu mà sử dụng phần tử cách ly quang gồm hai LED mắc song song ngược. Đầu vào sử dụng mạch lọc thông thấp R1C và mạch phân áp R2R3.

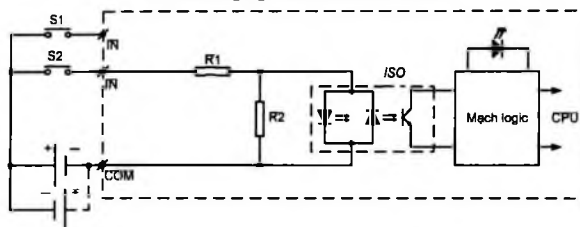
Hình 2.10 trình bày sơ đồ nguyên lý một đầu vào của module vào một chiều, điện áp 24VDC.

Sơ đồ hình 2.10a chỉ sử dụng nguồn dương. Sơ đồ hình 2.10b có thể sử dụng nguồn một chiều dương hoặc âm.

Việc phân loại module vào rời rạc dựa trên số lượng các điểm vào trên module và kiểu tín hiệu vào. Trên cơ sở số lượng các đầu vào trên module có các loại 8 điểm, 16 điểm, 32 điểm... Trên cơ sở kiểu tín hiệu vào có các loại 5VDC, 24V AC/DC, 48V AC/DC, 110VAC, 220VAC.



(a) Sử dụng nguồn một chiều cố định

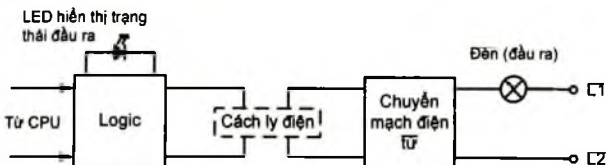


(b) Sử dụng nguồn một chiều tùy ý

Hình 2.9. Sơ đồ nguyên lý một đầu vào của module vào một chiều, điện áp 24VDC.

2.2.1.2. Các module ra rời rạc (Discrete Output Module)

Module ra rời rạc thực hiện các nhiệm vụ nhận dữ liệu từ CPU, biến đổi thành tín hiệu phù hợp điều khiển các thiết bị ra (cơ cấu chấp hành). Sơ đồ nối thiết bị ra với module ra rời rạc được trình bày trên hình 2.11.

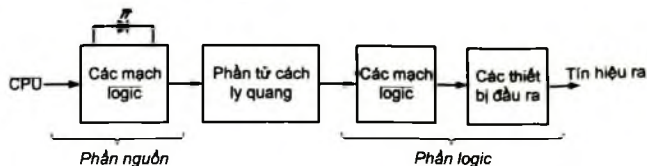


Hình 2.11. Sơ đồ nối thiết bị ra với module ra rời rạc.

Các thiết bị ra là các thiết bị logic như: đèn, rơ le, contactor, van... Đó là các thiết bị có hai trạng thái ON/OFF.

Mỗi thiết bị ra logic nối với module ra tại một điểm có vị trí xác định gọi là điểm đầu ra (*Output Point*). Mỗi điểm đầu ra tương ứng với một địa chỉ của Bit dữ liệu trong vùng ảnh đầu ra. Giá trị của Bit dữ liệu quyết định trạng thái của thiết bị ra. Nếu giá trị của Bit bằng 1 thì trạng thái của thiết bị ra là tích cực (*Active*). Nếu giá trị của Bit bằng 0, thì trạng thái của thiết bị ra là không tích cực (*Inactive*).

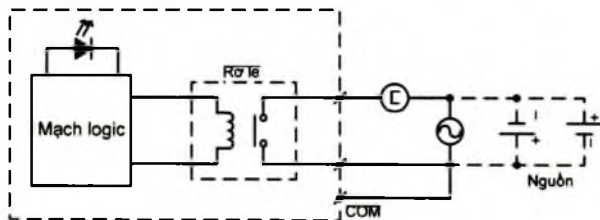
Trên các module ra đều có LED chỉ thị tín hiệu ra. Mạch điện của khối CPU và mạch ngoài được cách ly với nhau. Sơ đồ khối chức năng của module ra rời rạc được trình bày trên hình 2.12.



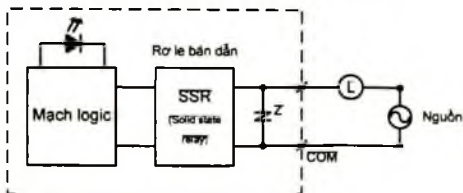
Hình 2.12. Sơ đồ khối chức năng của module ra rời rạc.

Sơ đồ chia làm hai phần: phần logic và phần nguồn. Các mạch logic xác định trạng thái đầu ra phụ thuộc tín hiệu nhận từ CPU. Trạng thái tín hiệu ra được chỉ thị bằng LED. Phần tử đầu ra có hai trạng thái ON/OFF tương ứng với tín hiệu từ mạch logic.

Hình 2.13 trình bày sơ đồ một đầu ra kiểu tiếp điểm. Phần tử L là tải. Đầu ra kiểu tiếp điểm rơ le điện từ (hình 2.13a) sử dụng nguồn cung cấp một chiều hoặc xoay chiều. Đầu ra kiểu rơ le bán dẫn (hình 2.13b) sử dụng nguồn xoay chiều.



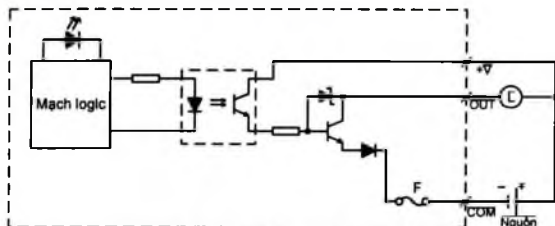
(a) Đầu ra kiểu tiếp điểm rơ le điện từ



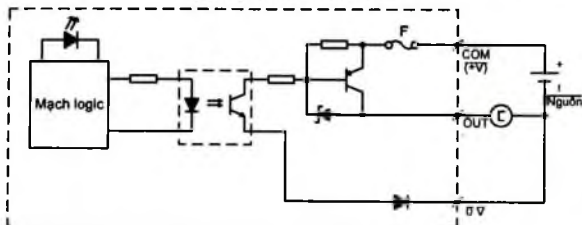
(b) Đầu ra kiểu rơ le bán dẫn (SSR – Solid State Relay)

Hình 2.13. Sơ đồ một đầu ra kiểu tiếp điểm.

Hình 2.14 trình bày sơ đồ đầu ra kiểu tín hiệu điện áp. Mạch đầu ra có hai kiểu: NPN và PNP. Mạch đầu ra kiểu NPN (hình 2.14a) có đặc điểm: điểm COM là 0V, tải L được nối giữa đầu ra (Out) và cực dương của nguồn (+V). Mạch đầu ra kiểu PNP (hình 2.14b) có đặc điểm: điểm COM là dương nguồn +V, tải L được nối giữa đầu ra (Out) và 0V. Các đầu ra kiểu tín hiệu điện áp đều sử dụng mạch collector hở, cầu chì bảo vệ quá dòng F và chỉ sử dụng nguồn cung cấp một chiều.



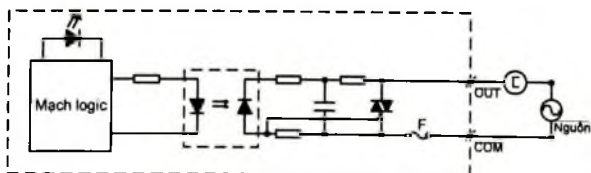
(a) Mạch đầu ra kiểu NPN



(b) Mạch đầu ra kiểu PNP

Hình 2.14. Sơ đồ đầu ra kiểu tín hiệu điện áp.

Hình 2.15 trình bày sơ đồ đầu ra kiểu xoay chiều. Phần tử đóng/cắt là Triac. Đầu ra xoay chiều này cho phép dòng qua tải lớn và điện áp nguồn 110VAC hoặc 220VAC.



Hình 2.15. Sơ đồ đầu ra kiểu xoay chiều.

Việc phân loại các module ra dựa trên cơ sở số lượng các đầu ra và kiểu đầu ra. Các kiểu đầu ra đã trình bày ở trên. Số lượng các đầu ra trên mỗi module ra có thể là 8, 16, 32.

Trong thực tế, để giảm kích thước, còn có các module vào/ra hỗn hợp. Ví dụ, module 8 đầu vào và 8 đầu ra, module 8 đầu vào và 16 đầu ra, module 16 đầu vào và 16 đầu ra, module 16 đầu vào và 8 đầu ra.

Về hình thức, các loại module vào/ra được mã hóa theo màu sắc để dễ nhận biết, trình bày trên bảng 2.1.

Bảng 2.1. Màu sắc các module vào/ra

<i>TT</i>	<i>Màu sắc</i>	<i>Kiểu I/O</i>
1	Đỏ (Red)	Xoay chiều (AC)
2	Xanh lam (Blue)	Một chiều (DC)
3	Da cam (Orange)	Đầu ra kiểu tiếp điểm rơ le điện từ
4	Xanh lục (Green)	Các module đặc biệt

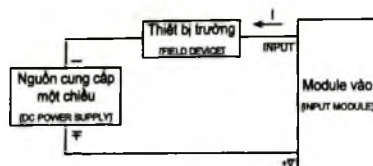
2.2.1.3. Các thiết bị trường (Field Devices)

Các thiết bị trường là các thiết bị vào và các thiết bị ra nối với các module vào/ra. Nguồn cung cấp cho các thiết bị trường là nguồn ngoài, có thể là một chiều hoặc xoay chiều. Trong trường hợp nguồn cung cấp là một chiều thì cần phải phân biệt khái niệm sourcing (cấp nguồn) và sinking (được cấp nguồn). Khái niệm sourcing và sinking được dùng để biểu diễn quan hệ về chiều dòng điện giữa thiết bị trường và nguồn cung cấp cho nó. Thiết bị trường nối với cực dương (+V) của nguồn cung cấp gọi là thiết bị trường kiểu sourcing. Thiết bị trường nối với cực âm của nguồn cung cấp (0V) gọi là thiết bị trường kiểu sinking. Như vậy, thiết bị vào và thiết bị ra đều có hai loại: sourcing và sinking. Thiết bị vào nối với module vào, thiết bị ra nối với module ra thông qua nguồn cung cấp. Vì vậy, các module vào/ra cũng có hai loại: sourcing và sinking. Tính tương thích giữa các thiết bị trường với các module vào/ra rời rạc một chiều được thể hiện như sau:

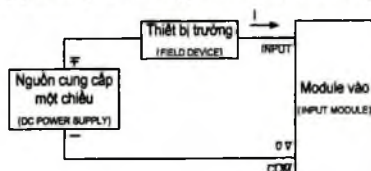
- Module vào/ra kiểu sourcing sẽ cung cấp dòng điện cho thiết bị trường kiểu sinking. Chiều dòng điện đi ra từ module vào/ra.

- Module vào/ra kiểu sinking sẽ nhận dòng điện từ thiết bị trường kiểu sourcing. Chiều dòng điện đi vào module vào/ra.

Hình 2.16a trình bày sơ đồ thiết bị vào kiểu sinking nối với module vào kiểu sourcing. Khi thiết bị vào ở trạng thái tích cực, nó nhận dòng điện từ module vào. Hình 2.16b trình bày sơ đồ thiết bị vào kiểu sourcing nối với module vào kiểu sinking. Khi thiết bị vào ở trạng thái tích cực, nó cấp dòng điện cho module vào.



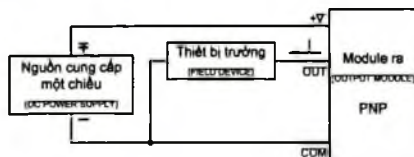
(a) Thiết bị vào sinking – module vào sourcing



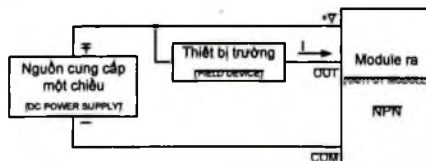
(b) Thiết bị vào sourcing – module vào sinking

Hình 2.16. Sơ đồ thiết bị vào nối với module vào.

Hình 2.17a trình bày sơ đồ thiết bị ra kiểu sinking nối với module ra kiểu sourcing. Khi đầu ra trạng thái tích cực, thiết bị ra được cấp dòng điện từ đầu ra kiểu PNP. Hình 2.17b trình bày sơ đồ thiết bị ra kiểu sourcing nối với module vào kiểu sinking. Khi đầu ra ở trạng thái tích cực, module ra nhận dòng điện từ thiết bị ra qua đầu ra kiểu NPN.



(a) Thiết bị ra sinking – module ra sourcing



(b) Thiết bị ra sourcing – module ra sinking

Hình 2.17. Sơ đồ thiết bị ra nối với module ra.

2.2.2. Các module vào/ra tương tự (Analog I/O modules)

Trong thực tế, PLC được ứng dụng trong các bộ điều khiển quá trình với nguồn tín hiệu và đối tượng điều khiển là các thiết bị tương tự. Các module vào/ra tương tự là các mạch ghép nối PLC với các thiết bị này.

2.2.2.1. Các module vào tương tự

Các module vào tương tự nhận tín hiệu tương tự (dòng điện, điện áp) từ thiết bị trường, từ các bộ chuyển đổi (*Transducer*), từ các bộ truyền tín hiệu (*Transmitter*) biến đổi thành tín hiệu số nhờ bộ biến đổi ADC.

Tín hiệu vào tương tự có hai dạng được chuẩn hóa là tín hiệu điện áp và tín hiệu dòng điện.

Tín hiệu điện áp có thể là đơn cực hoặc lưỡng cực. Tín hiệu đơn cực có các dải điện áp sau: (0V, +5V), (0V, +10V), (1V, +5V). Tín hiệu lưỡng cực có các dải điện áp sau: (-5V, +5V), (-10V, +10V).

Tín hiệu dòng điện có các dải (0 mA, 20 mA) và (4 mA, 20 mA).

Trên module vào tương tự có thể có 2, 4, 8 đầu vào gọi là các kênh. Kiểu và dải tín hiệu (V/I) ở mỗi kênh được chọn nhờ các chuyển mạch chọn trên module. Nguồn cung cấp cho các module vào tương tự thông qua Bus nguồn của hệ thống. Tuy nhiên, một số họ PLC yêu cầu nguồn cung cấp riêng từ ngoài cho các module vào tương tự.

CPU nhận tín hiệu số từ các kênh của module vào tương tự nhờ lệnh đọc riêng và cất vào một vùng nhớ riêng do hệ thống quy định. Mỗi họ PLC có cách tổ chức bộ nhớ riêng.

Các tham số đặc trưng cho module vào tương tự là:

- Số kênh;
- Kiểu và dải tín hiệu vào;
- Trở kháng vào;
- Độ phân dải: 8 Bit, 10 Bit, 12 Bit,...;
- Tốc độ biến đổi;
- Hệ số nén tín hiệu đồng pha.

Ngoài các module vào tương tự với tín hiệu điện áp và dòng điện chuẩn như trên, còn có các module vào tương tự, mà tín hiệu vào nhận trực tiếp từ các sensor. Ví dụ, các sensor nhiệt độ (cặp nhiệt ngẫu, điện trở nhiệt Pt 100...), sensor áp suất... Mạch vào của các module này là các bộ khuếch đại tín hiệu nhỏ. Vì vậy, dây nối sensor với các đầu vào phải được bọc kim để chống nhiễu.

2.2.2.2. Các module ra tương tự

Các module ra tương tự nhận tín hiệu số từ CPU, biến đổi thành tín hiệu điện áp và dòng điện để điều khiển các thiết bị trường. Thành phần cơ bản của module ra tương tự là bộ DAC. Tín hiệu ra tương tự được chuẩn hóa theo các thiết bị trường. Ví dụ,

tín hiệu điện áp có các dải: (0V, +5V), (0V, +10V), (-5V, +5V), (-10V, +10V), tín hiệu dòng điện có các dải: (0 mA, 20 mA), (4 mA, 20 mA).

Trên module ra tương tự có thể có 2, 4, 8 đầu ra gọi là các kênh. Kiểu và dải tín hiệu (V/I) ở mỗi kênh được chọn nhờ các chuyển mạch chọn trên module.

Các tham số đặc trưng cho module ra tương tự là:

- Số kênh;
- Kiểu và dải tín hiệu ra;
- Trở kháng ra;
- Độ phân dải: 8 Bit, 10 Bit, 12 Bit...;
- Tốc độ biến đổi.

Trong thực tế, ngoài các module vào/ra tương tự riêng biệt, nhà sản xuất còn cung cấp các module vào/ra tương tự kiểu hỗn hợp. Ví dụ, module 2 kênh vào – 2 kênh ra tương tự, module 4 kênh vào – 1 kênh ra tương tự.

2.2.3. Các module vào/ra đặc biệt

PLC ngày càng được sử dụng rộng rãi trong các hệ điều khiển và nhiều ứng dụng khác nhau. Các module vào/ra đặc biệt là các thiết bị chuyên dụng mà nhà sản xuất cung cấp để tích hợp hệ thống. Một số các module vào/ra đặc biệt thường sử dụng được trình bày dưới đây.

2.2.3.1. Module đếm tốc độ cao (High Speed COUNTER)

Module đếm tốc độ cao được sử dụng để đếm xung có tần số cao từ sensor, encoder hoặc các thiết bị phát xung. Tần số giới hạn có thể đến 500 KHz.

Ví dụ điển hình là module ghép nối với encoder trong các bộ điều khiển vị trí. Encoder là thiết bị biến đổi tốc độ quay hoặc góc quay của trục thành tín hiệu xung. Encoder có hai loại: đếm tăng (*Increment*) và tuyệt đối (*Absolute*). Loại đếm tăng có ba đầu ra: A, B, Z. Trong đó, ở đầu ra A và B là dãy xung tỷ lệ với vòng quay của trục, ở đầu ra Z là tín hiệu xác định điểm bắt đầu vòng quay (Zero). Sử dụng encoder loại đếm tăng có thể xác định tốc độ quay, số vòng quay và chiều quay của trục. Các module ghép với encoder kiểu này có ba đầu vào A, B, Z nối với các đầu ra tương ứng của encoder. CPU sử dụng lệnh đọc encoder để xác định số xung và chiều quay của trục. Encoder loại tuyệt đối dùng để đo góc quay tuyệt đối của trục (0 – 360°). Đầu ra của encoder là mã nhị phân, BCD hoặc mã GRAY 8 Bit, 10 Bit, 12 Bit... biểu diễn góc quay của trục. Các module ghép với encoder kiểu này có số đầu vào bằng với số Bit và CPU sử dụng lệnh đọc encoder để xác định góc quay của trục.

2.2.3.2. Module điều khiển động cơ bước (Stepper Motor Module)

Module điều khiển động cơ bước nhận lệnh từ CPU, biến đổi thành dãy xung để điều khiển trực tiếp các bộ công suất cho động cơ bước.

2.2.3.3. Module truyền thông (Communication Processor Module)

PLC là một nút tích hợp một hệ điều khiển. Dữ liệu trong hệ thống phải được trao đổi với nhau và thậm chí các thành phần của hệ thống phải được chia sẻ tài nguyên cho nhau. Các module truyền thông cho phép kết nối PLC với các thành phần khác của hệ thống thông qua mạng truyền thông số để thực hiện nhiệm vụ trên. Khác với các cổng truyền thông nối tiếp trên khối CPU, các cổng trên module truyền thông có nhiều chế độ hoạt động. Một số chế độ sau:

- Chế độ chủ tớ (*Host Link*).
- Chế độ không giao thức (*No Protocol*).
- Chế độ có giao thức (*Protocol Macro*).
- Chế độ kết nối dữ liệu điểm – điểm (*Point to Point*).
- Chế độ kết nối điểm – nhiều điểm (*Point to Multipoint*).

Trên module truyền thông có một hoặc nhiều cổng với các chuẩn khác nhau: RS232, RS422, RS485.

Các thiết bị ngoại vi cũng có thể kết nối với CPU thông qua module truyền thông. Trên hình 2.18 là một ví dụ về việc kết nối giữa module truyền thông và các thiết bị ngoại vi.

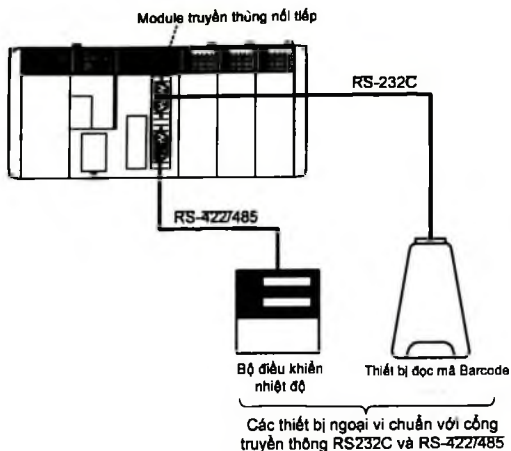
Các module truyền thông đóng vai trò rất quan trọng và được sử dụng rộng rãi trong các hệ điều khiển quá trình, các hệ DCS và SCADA. Các mạng truyền thông số công nghiệp theo chuẩn ISO điển hình: Profile Bus của SIEMENS, ControlNet của AB và Compro Bus của OMRON... Các phương thức truyền thông cũng rất khác nhau: cáp xoắn đôi, dây đồng trục, cáp quang, radio... Các module truyền thông của PLC cũng bao gồm cả các module ghép nối với các mạng truyền thông công nghiệp kể trên.

2.2.3.4. Module điều khiển PID

Các module điều khiển PID thực hiện bộ điều chỉnh PID trong các ứng dụng điều khiển quá trình. Đây là một khối tính toán bằng số theo phương pháp điều khiển PID bên ngoài CPU để giảm nhẹ công việc cho CPU. Module PID nhận giá trị đặt và các tham số điều chỉnh từ CPU, so sánh với giá trị thực và đưa ra tín hiệu điều khiển quá trình. Module PID điển hình là các bộ điều khiển nhiệt độ với đầu vào là sensor nhiệt độ, đầu ra là tín hiệu ON/OFF (ON/OFF PID), tín hiệu tương tự ($0 \div 10$ V), ($4 \div 20$ mA) điều khiển mạch công suất hoặc tín hiệu điều khiển góc mở van ($0 \div 100\%$).

2.2.3.5. Module vào/ra phân tán (Remote I/O)

Trong một số trường hợp, việc trao đổi dữ liệu giữa CPU với các module vào ra phân tán (ở xa) được thực hiện bằng phương pháp nối tiếp. Module vào/ra phân tán có nhiệm vụ chuyển đổi dữ liệu từ song song – nối tiếp và nối tiếp – song song.



Hình 2.18. Ví dụ kết nối giữa module truyền thông và một số thiết bị ngoại vi với PLC CQM1H của OMRON.

2.2.3.6. Các module vào/ra đặc biệt khác

Ngoài các module cơ bản nói trên thì mỗi nhà cung cấp PLC đều cho ra một số loại module vào/ra đặc biệt khác được liệt kê dưới đây:

- Module đặt số cơ khí (*Thumb-Wheel Module*): cho phép người sử dụng đặt thông số cho PLC ở dạng các chuyển mạch số cơ khí. Các thông số này chỉ được thay đổi ở bên ngoài PLC chứ không thay đổi được bằng chương trình trong PLC.
- Module đầu ra BCD: cho phép PLC gửi thông tin đến các thiết bị yêu cầu mã BCD. Ví dụ, module điều khiển LED 7 thanh (*7-segments*).
- Module Basic và ASCII: cho phép PLC nhận và gửi các File dữ liệu ASCII. Chương trình được viết bằng Basic. Loại module này được ứng dụng trong điều khiển máy in, hiển thị các thông tin điều khiển.
- Module điều khiển hệ servo: cho phép thực hiện hệ điều khiển vòng kín. Ví dụ, module điều khiển vị trí 1 trục, 2 trục...

2.2.4. Các tham số đặc trưng của module vào/ra

Các module vào/ra là thiết bị ghép nối PLC với các thiết bị trường. Việc ghép nối phải đảm bảo các thiết bị làm việc an toàn, tin cậy. Các tham số đặc trưng của module vào/ra bao gồm các tham số giới hạn mà người sử dụng cần biết khi chọn thiết bị cho ứng dụng.

– **Điện áp vào định mức:** là giá trị điện áp một chiều hoặc xoay chiều của tín hiệu cho phép đặt lên module vào.

– **Dải điện áp vào ở trạng thái ON:** là dải điện áp mà tín hiệu vào chắc chắn được PLC nhận ở trạng thái ON.

– **Dòng điện định mức đầu vào:** là giá trị dòng điện cực tiểu của một đầu vào cho phép hoạt động bình thường.

– **Nhiệt độ môi trường:** là giá trị nhiệt độ lớn nhất cho phép của môi trường mà các module vào/ra làm việc ở trạng thái tốt.

– **Thời gian trễ tín hiệu vào:** là thời gian kéo dài nhỏ nhất mà PLC có thể ghi nhận chắc chắn trạng thái tín hiệu vào kể từ thời điểm bắt đầu có tín hiệu tích cực. Thời gian trễ khoảng từ 9 ÷ 75 ms. Thời gian trễ này do mạch lọc thông thấp đầu vào quy định, mạch lọc này có chức năng khử nhiễu tần số cao.

– **Điện áp đầu ra định mức:** là giá trị điện áp một chiều hoặc xoay chiều của nguồn điều khiển các thiết bị bằng đầu ra.

– **Dải điện áp ra:** là điện áp trong khoảng cực tiểu và cực đại mà đầu ra chảy trên tải.

– **Dòng điện đầu ra cực đại/đầu ra/module:** là dòng điện cho phép cực đại cho một đầu ra để điều khiển tải. Đặc trưng cho chế độ làm việc dài hạn.

– **Dòng điện đầu ra xung cực đại:** là giá trị lớn nhất của dòng điện tồn tại trong khoảng thời gian ngắn. Đặc trưng cho chế độ làm việc ngắn hạn.

– **Dòng điện rò của đầu ra ở trạng thái OFF:** là giá trị lớn nhất cho phép của dòng rò khi đầu ra ở trạng thái OFF.

– **Độ cách điện:** là giá trị cực đại của điện áp đảm bảo cách ly giữa PLC và các thiết bị vào/ra.

– **Số lượng đầu vào/ra trên module:** xác định số lượng các thiết bị trường có thể được nối với module.

– **Dòng điện cho phép của bảng mạch Bus:** là giá trị cực đại mà tất cả các module tiêu thụ từ nguồn cung cấp thông qua Bus.

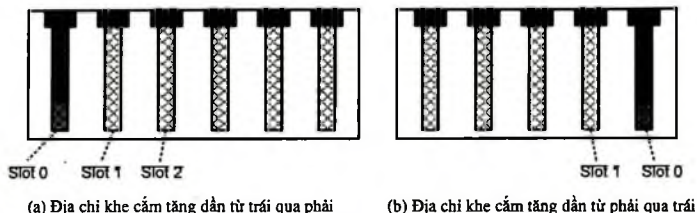
2.3. ĐỊA CHỈ HÓA VÀO/RA

Các thiết bị vào/ra là các thiết bị vật lý được nối với PLC qua các module vào/ra. CPU quản lý các thiết bị vào/ra bằng địa chỉ.

Trạng thái của tín hiệu vào/ra được lưu giữ trong vùng nhớ vào/ra của PLC. CPU trao đổi dữ liệu với các module vào/ra bằng phương pháp song song qua hệ thống Bus. Mỗi lần CPU đọc/ghi dữ liệu có độ dài bằng 1 kích thước logic của ô nhớ (8 Bit,

16 Bit, 32 Bit) gọi là một từ vào/ra (*I/O Word*). Tuy nhiên, trên mỗi module vào/ra có thể có số lượng các đầu vào/ra khác nhau (8, 16, 32, 64...), cũng có thể là module vào/ra hỗn hợp và các module vào/ra đặc biệt cũng rất phong phú. Do vậy, việc địa chỉ hóa vào/ra phải tuân theo nguyên tắc nhất định và do hệ điều hành quy định.

Các module vào/ra được lắp vào các khe cắm vật lý (*Slot*) trên bảng mạch Bus và chúng có thể được bố trí về phía bên phải hoặc bên trái của khối CPU. Thông thường khối CPU được sử dụng làm gốc địa chỉ (địa chỉ 0 hoặc 1). Trên hình 2.19 là một số cách bố trí khe cắm dành cho module vào/ra.



Hình 2.19. Một số cách bố trí khe cắm dành cho module vào/ra trên bảng mạch Bus.

Hệ thống quản lý địa chỉ các điểm vào ra dựa trên cơ sở khái niệm địa chỉ kênh logic (*Rack, Element*). Địa chỉ kênh logic là địa chỉ của vùng nhớ vào/ra. Mỗi kênh logic có kích thước bằng kích thước của một từ vào/ra. Như vậy, hệ thống có thể quản lý 1 kênh, 2 kênh,... trên một khe cắm vật lý tùy thuộc vào kiểu module vào/ra. Ví dụ, đối với PLC 8 Bit, trên module vào rời rạc có 8 đầu vào chiếm địa chỉ 1 kênh vào; trên module vào có 16 Bit đầu vào sẽ chiếm địa chỉ 2 kênh vào; trên module có 8 đầu vào, 8 đầu ra sẽ chiếm địa chỉ 1 kênh vào và 1 kênh ra.

2.3.1. Địa chỉ hóa module vào/ra rời rạc

Một cách tổng quát, mỗi điểm trên module vào/ra rời rạc có địa chỉ được xác định như sau:

$$\text{Địa chỉ vào/ra} = [\text{tên}]x.y$$

Trong đó: [tên] là các chữ cái hoặc tập các ký tự chỉ ra kiểu dữ liệu (vào hoặc ra);

x: địa chỉ kênh logic phụ thuộc vào vị trí của khe cắm vật lý (địa chỉ kênh gốc tính từ CPU);

y: địa chỉ Bit trong kênh.

Ví dụ:

Hãng SIEMENS, ALLEN BRADLEY:

[Tên] I: chỉ kiểu dữ liệu vào
Q hoặc O: chỉ kiểu dữ liệu ra

Hãng OMRON:

[Tên] IR000: chỉ kiểu dữ liệu vào
IR100: chỉ kiểu dữ liệu ra

Ví dụ:

PLC SIEMENS: I0.0: Địa chỉ Bit vào đầu tiên của kênh vào đầu tiên.

Q0.0: Địa chỉ Bit ra đầu tiên của kênh ra đầu tiên.

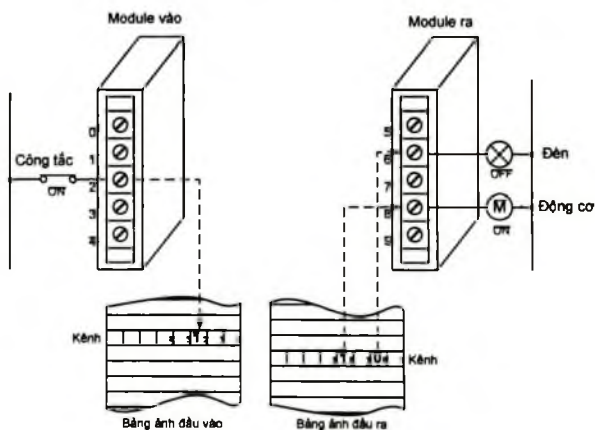
PLC ALLEN BRADLEY: I:000/0: Địa chỉ Bit vào đầu tiên của kênh vào đầu tiên.

O:000/0: Địa chỉ Bit ra đầu tiên của kênh ra đầu tiên.

PLC OMRON: IR000.00: Địa chỉ Bit vào đầu tiên của kênh vào đầu tiên.

IR100.00: Địa chỉ Bit ra đầu tiên của kênh ra đầu tiên.

Trên hình 2.20 minh họa cách địa chỉ hóa vào/ra rời rạc.



Hình 2.20. Địa chỉ hóa vào/ra rời rạc.

CPU có thể truy nhập đến module vào/ra theo địa chỉ kênh (Word) hoặc theo địa chỉ Bit. Giữa địa chỉ kênh và địa chỉ Bit được ngăn cách bởi dấu "." hoặc dấu "/". Ví dụ:

Địa chỉ kênh:	IW1	QW1	I:100	O:010	IR000	IR100
Địa chỉ Bit:	I1.0	Q1.0	I:100/1	Q:010/2	IR000.00	IR100.00

Trường hợp trên module vào/ra số các điểm vào/ra nhỏ hơn số Bit cùng một kênh logic thì các Bit còn lại không được sử dụng để địa chỉ hóa các điểm vào/ra của kênh khác, mà chỉ được sử dụng như các Bit trung gian. Nghĩa là: CPU quản lý vào/ra là theo từng kênh nguyên.

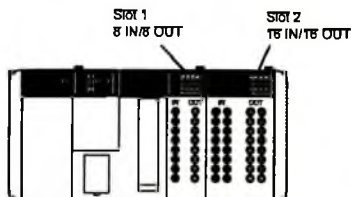
Xét ví dụ hình 2.21 với một PLC có mỗi từ vào/ra là 16 Bit, trên khe cắm số 1 là module vào/ra gồm 8 đầu vào và 8 đầu ra, trên khe cắm số 2 là module vào/ra gồm 16 đầu vào và 16 đầu ra. Khi đó địa chỉ của các đầu vào/ra được xác định trên bảng 2.2.

Bảng 2.2. Địa chỉ các đầu vào/ra

	SIEMENS		ALLEN BRADLEY		OMRON	
	Địa chỉ vào	Địa chỉ ra	Địa chỉ vào	Địa chỉ ra	Địa chỉ vào(IR)	Địa chỉ ra(IR)
Slot 1	I1.0 ÷ I1.7	Q1.0 ÷ Q1.7	I:1/0 ÷ I:1/7	Q:1/0 ÷ Q:1/7	000.00 ÷ 000.07	100.00 ÷ 100.07
Slot 2	I2.0 ÷ I2.15	Q2.0 ÷ Q2.15	I1.0 ÷ I1.7	Q2/0 ÷ Q2/15	001.00 ÷ 001.15	101.00 ÷ 101.15

Khi trạng thái vào là ON, tín hiệu vào ở mức cao (*High*) thì Bit vào tương ứng có giá trị 1. Khi trạng thái vào là OFF, tín hiệu vào ở mức thấp (*Low*) thì Bit vào tương ứng có giá trị 0.

Tương tự, khi Bit ra có giá trị 1 thì tín hiệu ra sẽ tích cực (*Active*) và khi Bit ra có giá trị 0 thì tín hiệu ra không tích cực (*Inactive*).



Hình 2.21. Ví dụ địa chỉ hóa vào/ra cho một PLC với 2 module vào/ra.

2.3.2. Địa chỉ hóa module vào/ra đặc biệt

Đối với module vào/ra phân tán (*Remote I/O module*) thì CPU quản lý tương tự như module vào/ra rời rạc tại chỗ (*Local*). Đối với các module vào/ra tương tự, CPU quản lý theo địa chỉ của từ (*Word*) biểu diễn số tín hiệu tương tự. Mỗi từ có thể có kích thước 1, 2, 4 kênh logic. CPU dành một vùng nhớ đặc biệt để lưu giữ các từ vào/ra tương tự. Địa chỉ của mỗi đầu vào/ra tương tự trên module vào/ra tương tự phụ thuộc vào vị trí của khe cắm, thứ tự trên module vào/ra và địa chỉ được xác định trên module vào/ra bằng chuyển mạch.

Có hai cách CPU địa chỉ hóa các đầu vào/ra:

- Dành một phần trên vùng nhớ vào/ra.
- Dành một vùng nhớ riêng.

Tương tự, đối với các module vào/ra đặc biệt khác như module truyền thông, module điều khiển PID..., các từ dữ liệu được xác định theo địa chỉ (*Address*) được chỉ ra trong phần khai báo toán hạng (*Operand*) của lệnh (*Instruction*) tương ứng.

2.4. CẤU HÌNH HỆ ĐIỀU KHIỂN LOGIC PLC

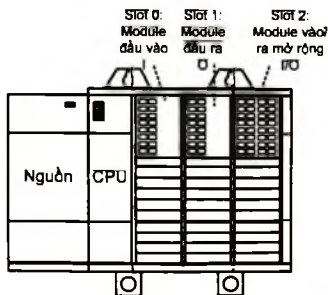
Cấu hình hệ điều khiển logic PLC bao gồm các thiết bị và cách tổ chức, ghép nối các thiết bị đó với nhau tạo thành một hệ thống đáp ứng yêu cầu đặt ra. Như vậy, bước đầu tiên của việc thiết lập cấu hình của hệ là chọn lựa thiết bị dựa trên yêu cầu điều khiển, chọn lựa các module chức năng. Bước tiếp theo là sắp xếp, tổ chức ghép nối các module chức năng với nhau và ghép nối với thiết bị vào/ra, thiết bị ngoại vi để tạo nên một cấu hình phần cứng của hệ thống. Ở đây cần phân biệt cấu hình phần cứng với cấu hình hoạt động của hệ. Cấu hình hoạt động của hệ bao gồm cấu hình phần cứng và thiết lập các tham số cấu hình hệ thống bằng phần mềm.

Một cách hình thức, cấu hình của hệ điều khiển PLC được chia thành hai loại: cấu hình cơ bản và cấu hình mở rộng (đặc biệt).

2.4.1. Cấu hình cơ bản

Cấu hình cơ bản của hệ điều khiển PLC bao gồm các khối chức năng cơ bản là: khối nguồn, khối CPU và các module vào/ra rời rạc, hệ thống truyền thông và giao diện người – máy... Cấu hình cơ bản của hệ điều khiển PLC được trình bày trên hình 2.22, gồm có:

Module 16 đầu vào bố trí ở khe cắm số 0, module 16 đầu ra bố trí ở khe cắm số 1 và module 32 đầu vào/ra (16 đầu vào và 16 đầu ra) bố trí ở khe cắm số 2.



Hình 2.22. Cấu hình cơ bản của một hệ điều khiển PLC.

Trong cấu hình trên, địa chỉ các đầu vào ra được trình bày trong bảng 2.3.

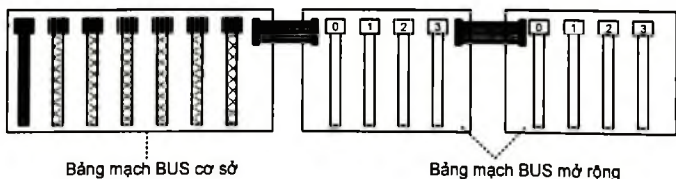
Bảng 2.3. Địa chỉ vào/ra trên các module vào/ra

STT	Module vào/ra trên Slot	SIEMENS		OMRON	AB
		S7-200	S7-300	CQM1	SLC500
1	Module vào trên Slot 0	I0.0 ÷ I0.7 I1.0 ÷ I1.7	I0.00 ÷ I0.15	IR00.00 ÷ IR00.15	I:0/00 ÷ I:0/15
2	Module ra trên Slot 1	Q0.0 ÷ Q0.7 Q1.0 ÷ Q1.7	Q0.00 ÷ Q0.15	IR100.00 ÷ IR100.15	O:1/00 ÷ O:1/15
3	Module vào/ra trên Slot 2	I2.0 ÷ I0.7 I3.0 ÷ I1.7 Q2.0 ÷ Q2.7 Q3.0 ÷ Q3.7	I1.00 ÷ I1.15 Q1.00 ÷ Q1.15	IR01.00 ÷ IR01.15 IR101.00 ÷ IR101.15	I:2/00 ÷ I:2/15 O:3/00 ÷ :3/15

Ghi chú: S7-200 là họ PLC 8Bit.

Bảng mạch Bus chỉ chứa một số khe cắm nhất định. Ví dụ, bảng mạch Bus chứa 4, 7, 10... khe cắm. Trong trường hợp số khe cắm không đủ cho các module vào/ra thì phải sử dụng các bảng mạch Bus mở rộng. Trên hình 2.23 mô tả bảng mạch Bus mở rộng.

Mỗi họ PLC (và mỗi kiểu PLC) quy định số cực đại các bảng mạch Bus mở rộng và trên mỗi bảng mạch Bus chỉ quản lý một số địa chỉ kênh xác định. Số các bảng mạch Bus phụ thuộc vào dung lượng bộ nhớ của PLC dành cho vào/ra. Vì vậy, khi số đầu vào/ra trên một bảng mạch Bus chiếm đủ số địa chỉ theo quy định thì phải sử dụng thêm các bảng mạch Bus mở rộng. Mặc dù số khe cắm vẫn còn.



Hình 2.23. Bảng mạch Bus mở rộng.

2.4.2. Cấu hình đặc biệt

Cấu hình đặc biệt gồm cấu hình cơ bản nói trên và các module vào/ra đặc biệt như: module vào ra tương tự, module PID... Các module vào/ra đặc biệt được bố trí trên các khe cắm của bảng mạch Bus theo thứ tự nhất định và chiếm các địa chỉ trong bộ nhớ tùy thuộc vào vị trí của chúng trên bảng mạch Bus. Mỗi kiểu module vào/ra đặc biệt khi sử dụng phải được đặt chế độ (*Setting*) và khai báo cho hệ thống.

Một số cấu hình đặc biệt khác:

- Cấu hình vào/ra từ xa (*Remote I/O*): Trên một khe cắm của bảng mạch Bus sử dụng khối vào/ra từ xa có nhiệm vụ chuyển đổi dữ liệu từ song song sang nối tiếp và ngược lại. Mỗi khối này sẽ quản lý một số nhất định các điểm vào/ra rời rạc ở xa. Địa chỉ các đầu vào/ra từ xa tương tự như các đầu vào/ra tại chỗ.

- Cấu hình dự phòng nóng: Các hệ thống yêu cầu độ an toàn, tin cậy cao khi hoạt động thường sử dụng cấu hình dự phòng nóng. Hệ thống sử dụng hai khối CPU giống nhau. Khi hoạt động, chỉ có một CPU chính hoạt động, CPU còn lại làm nhiệm vụ dự phòng nóng. Trong trường hợp CPU chính bị lỗi hay hỏng thì CPU dự phòng sẽ được đưa vào hoạt động thay thế. Cấu hình này thường sử dụng với các hệ thống lớn và CPU phải có tính năng làm việc song song.

Chương 3

TỔ CHỨC BỘ NHỚ VÀ CẤU TRÚC DỮ LIỆU

3.1. TỔ CHỨC BỘ NHỚ

Bộ nhớ của PLC là thiết bị lưu giữ thông tin của hệ thống. Bộ nhớ được xây dựng trên cơ sở các bộ nhớ bán dẫn: bộ nhớ không duy trì (RAM tĩnh – SRAM, RAM động – DRAM) và bộ nhớ duy trì (bộ nhớ ROM, EPROM, EEPROM, Flash ROM).

Các thông tin lưu giữ trong bộ nhớ là chương trình ứng dụng (*User Program*), dữ liệu (*Data*) và các tham số của hệ thống (*System Parameters*). Việc quản lý bộ nhớ do hệ điều hành đảm nhiệm và người sử dụng có thể truy nhập đến các vùng bộ nhớ do hệ điều hành quy định.

Về tổ chức, bộ nhớ được chia thành các vùng nhớ (*Memory Location, Memory Space*). Mỗi vùng nhớ chỉ lưu giữ một kiểu thông tin nhất định. Hình ảnh của bộ nhớ với các vùng nhớ như vậy gọi là bản đồ bộ nhớ (*Memory Map*). Vùng nhớ để lưu giữ chương trình gọi là vùng nhớ chương trình (*Program Memory*). Chương trình là chuỗi các lệnh mà PLC phải xử lý để thực hiện một thuật giải điều khiển (*Algorithm*). Chương trình có thể được biểu diễn ở dạng giản đồ thang (**LAD**), dòng lệnh (**STL**) hoặc sơ đồ khối chức năng điều khiển (**CSF**). Tuy nhiên, nội dung của bộ nhớ chương trình chỉ là các lệnh ở dạng mã máy. Tùy thuộc vào kiểu lệnh mà mỗi lệnh có độ dài 1, 2, 3... từ nhị phân (*Binary Word*) gọi là mã máy. CPU chỉ hiểu lệnh ở dạng mã máy.

Kích thước của bộ nhớ chương trình tùy thuộc vào họ PLC và kiểu CPU. Tham số để đánh giá bộ nhớ chương trình là dung lượng tính bằng *K lệnh* – 1024 lệnh. Khác với bộ nhớ chương trình của máy tính, bộ nhớ chương trình của PLC chỉ lưu giữ một chương trình tại một thời điểm. Khi nạp vào bộ nhớ chương trình một chương trình mới khác thì chương trình cũ ở trong bộ nhớ sẽ mất.

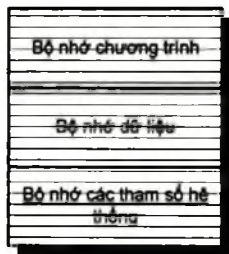
Người sử dụng không thể truy nhập trực tiếp được vào từng ô nhớ của bộ nhớ chương trình. Nghĩa là không thể ghi/đọc được từng mã lệnh từ bộ nhớ chương trình. Việc đọc/ghi chương trình từ/vào bộ nhớ chương trình hoàn toàn do hệ điều hành đảm nhiệm. Người sử dụng giao tiếp với PLC thông qua ngôn ngữ lập trình.

Vùng nhớ dùng để lưu giữ dữ liệu gọi là vùng nhớ dữ liệu (*Data Memory*). Bộ nhớ dữ liệu lưu giữ các kiểu dữ liệu khác nhau để thực hiện chương trình hoặc các kết quả thực hiện chương trình. Ví dụ, vùng nhớ ảnh vào/ra, vùng nhớ kết quả trung gian, vùng nhớ các hằng số, TIMER, COUNTER... Bộ nhớ dữ liệu được chia thành

các vùng, mỗi vùng chỉ lưu giữ một kiểu dữ liệu nhất định gọi là cấu trúc dữ liệu (*Data Structure*). Mỗi họ PLC có một cấu trúc dữ liệu riêng. Người sử dụng có thể truy nhập đến các thành phần (Word hoặc Bit) của từng vùng dữ liệu từ chương trình hoặc thiết bị lập trình.

Các phương pháp truy nhập đến các vùng dữ liệu từ mức chương trình gọi là các mode địa chỉ hóa. Việc truy nhập đến các vùng nhớ dữ liệu từ thiết bị lập trình để soạn thảo, lưu giữ, giám sát các File dữ liệu của chương trình. Tham số đánh giá bộ nhớ dữ liệu là dung lượng của nó.

Vùng nhớ để lưu giữ các tham số của hệ thống dùng để thiết lập cấu hình của hệ, các lỗi hệ thống... Người sử dụng có thể truy nhập đến vùng nhớ này. Tuy nhiên, một số vùng chỉ cho phép đọc nội dung, một số vùng khác có thể đọc/ghi nội dung của các ô nhớ được. Người ta có thể xếp vùng nhớ các tham số hệ thống vào bộ nhớ dữ liệu. Hình 3.1 trình bày cách tổ chức bộ nhớ của PLC.



Hình 3.1. Cách tổ chức bộ nhớ của PLC.

3.2. CẤU TRÚC DỮ LIỆU

Bộ nhớ dữ liệu là một phần của bộ nhớ của PLC được dùng để lưu giữ các kiểu dữ liệu khác nhau. Hệ điều hành quản lý bộ nhớ dữ liệu theo các vùng dữ liệu tạo thành một cấu trúc dữ liệu.

Mỗi vùng dữ liệu có kích thước nhất định bằng số các ô nhớ mà vùng dữ liệu chiếm và phụ thuộc vào kiểu CPU được chọn. Mỗi vùng dữ liệu có một tên riêng do hệ điều hành quy định.

Ví dụ: I: Vùng ảnh đầu vào.

Q: Vùng ảnh đầu ra.

Kích thước mỗi ô nhớ dữ liệu bằng kích thước một kênh logic có độ dài: 8, 16, 32 Bit.

Người sử dụng chỉ có thể truy nhập đến các vùng nhớ dữ liệu qua tên của vùng dữ liệu. Dữ liệu có thể truy nhập ở dạng từ (*Word*) hoặc Bit tùy thuộc vào lệnh. Vị trí

của từ và Bit trong vùng dữ liệu gọi là địa chỉ của nó. Phương pháp địa chỉ hóa dữ liệu gọi là mode địa chỉ hóa.

Mỗi họ PLC có cấu trúc dữ liệu riêng. Tuy nhiên, các vùng dữ liệu cơ bản của PLC sẽ được liệt kê dưới đây.

– **Vùng ảnh đầu vào (Input Image):** là vùng dữ liệu lưu giữ trạng thái của tín hiệu vào do CPU đọc từ các module vào trong quá trình quét và đọc dữ liệu vào để thực hiện chương trình.

– **Vùng ảnh đầu ra (Output Image):** là vùng dữ liệu lưu giữ trạng thái sẽ gửi ra module ra để điều khiển các cơ cấu chấp hành. Đó là kết quả của việc thực hiện chương trình.

Kích thước của vùng ảnh vào/ra tùy thuộc vào CPU được chọn và tương ứng với số đầu vào/ra vật lý mà PLC có thể quản lý. Vùng ảnh vào/ra có liên quan chặt chẽ đến phương pháp địa chỉ hóa vào/ra.

Mỗi vị trí của module vào/ra trên bảng mạch Bus đều tương ứng với một vùng xác định trong vùng ảnh vào/ra. Người sử dụng có thể đọc/ghi vùng ảnh vào/ra ở dạng từ hoặc Bit.

– **Vùng Bit trung gian (Work Bit):** là vùng dữ liệu lưu giữ các trạng thái Bit trung gian gọi là rơ le bên trong (*Internal Relay*). Các kết quả trung gian trong quá trình thực hiện chương trình có thể được lưu giữ trong vùng nhớ này. Trong vùng nhớ này cũng lưu giữ các Bit duy trì trạng thái (*Retentive Bit, Hold Bit*) khi mất nguồn. Người sử dụng có thể đọc/ghi theo từ hoặc Bit.

– **Vùng các biến (Variable Memory):** là vùng dữ liệu lưu giữ các giá trị của các biến dùng trong chương trình. Ví dụ, các hằng số, các giá trị đặt, các kết quả tính toán. Người sử dụng có thể đọc/ghi theo từ hoặc Bit.

– **Vùng lưu giữ các Bit trạng thái đặc biệt (Special Bit):** là vùng lưu giữ các cờ (*Flag*) trạng thái hệ thống, các tham số cấu hình của hệ thống... Người sử dụng có thể truy nhập ở dạng từ hoặc Bit. Tuy nhiên, một số dữ liệu chỉ có thể đọc mà không ghi được (*Read Only*).

– **Vùng số liệu (Data Memory):** được chia làm hai vùng: *vùng chỉ đọc* dành cho hệ thống và *vùng đọc/ghi* dành cho người sử dụng. Vùng dữ liệu chỉ có thể truy nhập ở dạng từ.

– Vùng dành cho COUNTER và TIMER:

Đây là vùng nhớ để lưu giữ các tham số của bộ đếm (*COUNTER*) và bộ định thời (*TIMER*) như giá trị đặt (*SV- Set Value*), giá trị hiện thời (*PV- Present Value*) và các cờ trạng thái của COUNTER, TIMER.

– Các vùng nhớ dữ liệu khác:

Tùy thuộc vào họ PLC mà các vùng nhớ dữ liệu sau đây có thể được sử dụng:

+ Vùng nhớ dữ liệu cho các module vào/ra tương tự (*Analog I/O Module*).

- + Vùng nhớ dành cho tổ chức ngắt.
- + Vùng nhớ dành cho kết nối giữa các PLC với nhau (Link – Link) và với các thiết bị khác.
- + Vùng nhớ dành cho bộ đếm tốc độ cao (*High Speed Counter*).
- + Vùng nhớ dành cho các module vào/ra đặc biệt (*Remote I/O, PID Module...*).
- + Vùng nhớ tạm thời (*Template Bit*)...

Khác với các họ PLC cũ, xu hướng tổ chức bộ nhớ dữ liệu của các họ PLC hiện đại là không chiếm vùng ảnh vào/ra. Nghĩa là, mỗi thiết bị, module đặc biệt... nối vào hệ đều được hệ thống dành cho một vùng nhớ dữ liệu riêng biệt, không chiếm vùng ảnh vào/ra chỉ dành riêng cho các module vào/ra rời rạc. Vì vậy, việc sử dụng các thiết bị ngoại vi không ảnh hưởng đến số các module vào/ra rời rạc được sử dụng.

3.3. TỔ CHỨC BỘ NHỚ VÀ CẤU TRÚC DỮ LIỆU CỦA MỘT SỐ HỌ PLC

3.3.1. Họ simatic S7–200 của Siemens [6]

3.3.1.1. Tổ chức bộ nhớ

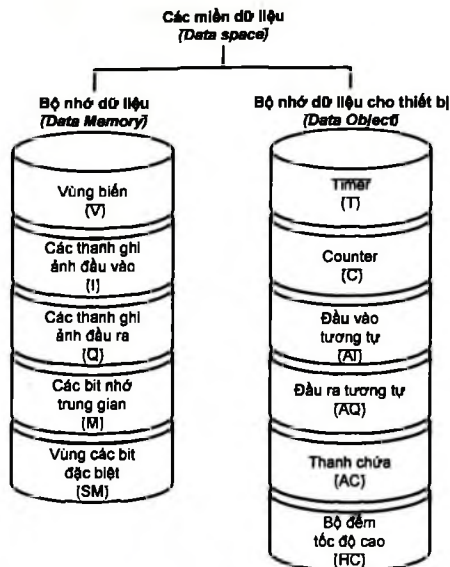
Bộ nhớ của PLC S7–200 chia thành ba miền:

- Miền chương trình (*Program Space*): lưu giữ các lệnh của chương trình sau khi đã được biên dịch sang mã máy.
- Miền dữ liệu (*Data Space*): bao gồm các vùng nhớ để tính toán, lưu giữ tạm thời các kết quả tính toán, lưu giữ các hằng được sử dụng trong chương trình, các tham số điều khiển cố định (giá trị đặt). Miền dữ liệu cũng bao gồm các vùng nhớ dành cho các thiết bị như TIMER, COUNTER và bộ đếm tốc độ cao (*High Speed COUNTER*), các đầu vào/ra tương tự.
- Miền tham số cấu hình (*Configurable Parameter Space*): lưu giữ các tham số cấu hình hệ thống mặc định (*Default*) hoặc tham số cấu hình do người sử dụng thiết lập.

Trong đó nội dung của miền nhớ chương trình, miền tham số và một phần miền dữ liệu được sao lưu (*Backup*) trong bộ nhớ EEPROM trong CPU. Một số vùng nhớ của miền dữ liệu được nuôi bằng nguồn cung cấp từ tụ điện có dung lượng lớn thay cho pin (*Super Capacity*).

3.3.1.2. Cấu trúc dữ liệu

Cấu trúc dữ liệu của PLC họ S7–200 được thể hiện qua cách tổ chức miền dữ liệu. Miền dữ liệu gồm hai thành phần: bộ nhớ dữ liệu (*Data Memory*) và bộ nhớ dành cho các thiết bị (*Data Object*) được trình bày trên hình 3.2.



Hình 3.2. Cấu trúc dữ liệu của SIMATIC S7-200.

Mỗi vùng nhớ có một tên riêng và kích thước các vùng nhớ tùy thuộc vào kiểu CPU. Trên hình 3.2 trình bày cấu trúc dữ liệu của họ S7-200. Bảng 3.1 trình bày cấu trúc dữ liệu của S7-200 CPU 212 và CPU 214.

Truy nhập đến các vùng nhớ có thể ở dạng Bit, Byte, Word (từ), DWord (Double Word – từ kép). Họ S7-200 định nghĩa 1 Word = 2 Byte và 1 DWord = 4 Byte.

Trong bộ nhớ dữ liệu, vùng biến (V) có hai dạng bộ nhớ là RAM và EEPROM. Bộ nhớ EEPROM là bộ nhớ duy trì, bộ nhớ RAM là bộ nhớ không duy trì. CPU 212 có 1024 Byte (1KB) ở vùng biến, trong đó 200 Byte đầu tiên lưu giữ cả ở RAM và EEPROM dùng để duy trì trạng thái khi ngắt nguồn. Các thanh ghi ảnh đầu vào (I) lưu giữ trạng thái của tín hiệu vào ở đầu chu kỳ quét và thanh ghi ảnh đầu ra (Q) ghi trạng thái của các đầu ra ở cuối chu kỳ quét. Tuy nhiên, một số lệnh vào/ra tức thì thì cho phép truy nhập trực tiếp đến các thanh ghi ảnh vào/ra như một ngắt. CPU 212 có 8 Byte dành cho vùng I và 8 Byte dành cho vùng Q.

Bảng 3.1. Cấu trúc dữ liệu của CPU212 và CPU214

Vùng dữ liệu	CPU 212			CPU 214		
	Byte	Word	DWord	Byte	Word	DWord
Đầu vào	IB0 đến IB7	IW0 đến IW6	ID0 đến ID4	IB0 đến IB7	IW0 đến IW6	ID0 đến ID4
Đầu ra	QB0 đến QB7	QW0 đến QW6	QD0 đến QD4	QB0 đến QB7	QW0 đến QW6	QD0 đến QD4
Bit nhớ trong (trung gian)	MB0 đến MB15	MW0 đến MW14	MD0 đến MD12	MB0 đến MB31	MW0 đến MW30	MD0 đến MD28
Bit nhớ đặc biệt	SM0 đến SM45	SM0 đến SM44	SM0 đến SM42	SM0 đến SM85	SM0 đến SM84	SM0 đến SM82
Vùng nhớ động	VB0 đến VB1023	VW0 đến VW1022	VD0 đến VD1020	VB0 đến VB4095	VW0 đến VW4094	VD0 đến VD4092

Vùng các Bit trung gian (M) còn gọi là vùng các Bit nhớ bên trong hoặc rơ le bên trong (*Internal Relay*) lưu giữ các kết quả trung gian khi thực hiện chương trình hoặc các thông tin điều khiển. CPU 212 có 16 Byte (MB0 ÷ MB15) và CPU 214 có 32 Byte (MB0 ÷ MB31) dành cho vùng các Bit trung gian.

Vùng các Bit đặc biệt (SM) lưu giữ các trạng thái hoạt động của PLC (cờ), các chức năng điều khiển, cấu hình hệ thống... Một số đoạn trong vùng các Bit đặc biệt chỉ đọc (*Read Only*) dành cho hệ thống và một số đoạn có thể ghi/đọc, xóa được. CPU 212 có 46 Byte và CPU 214 có 85 Byte dành cho vùng nhớ SM.

Bộ nhớ (dữ liệu) dành cho các thiết bị gồm có các vùng nhớ dành cho **TIMER** (T), **COUNTER** (C), đầu vào tương tự (AIW), đầu ra tương tự (QIW), các thanh chứa (AC) và bộ đếm tốc độ cao (HC). Dữ liệu của mỗi **TIMER** (T) hoặc **COUNTER** (C) gồm có một từ 16 Bit lưu giữ giá trị hiện thời (PV) và một Bit trạng thái (cờ) kết thúc chu trình hoạt động.

CPU 212 có 64 **TIMER** và 64 **COUNTER**.

CPU 214 có 128 **TIMER** và 128 **COUNTER**.

Dữ liệu của mỗi đầu vào và đầu ra tương tự gồm hai từ 16 Bit.

CPU 212 và CPU 214 có 15 đầu vào tương tự: AIW0 ÷ AIW30.

CPU 212 và CPU 214 có 15 đầu ra tương tự: AQW0 ÷ AQW30.

Các thanh chứa là các thanh ghi 32 Bit. CPU 212 và CPU 214 có bốn thanh chứa AC3 ÷ AC0.

Vùng dữ liệu dành cho bộ đếm tốc độ cao là các thanh ghi 32 Bit.

CPU 212 có một bộ đếm tốc độ cao HC0, CPU 214 có ba bộ đếm tốc độ cao HC2 ÷ HC0.

Cấu trúc dữ liệu chi tiết có thể tìm trong tài liệu [6].

3.3.1.3. Truy nhập vùng dữ liệu

Truy nhập đến một vùng dữ liệu phải chỉ ra địa chỉ của nó trong ô nhớ. Bộ nhớ dữ liệu của S7-200 chia làm năm vùng: I, Q, M, SM và V. Người sử dụng có thể truy nhập đến các vùng nhớ này ở dạng Bit, Byte, Word và DWord. Dữ liệu kiểu Bit mô tả trạng thái của một rơ le bên trong có hai giá trị 0, 1. Dữ liệu kiểu Byte có độ dài 8 Bit. Dữ liệu kiểu Word có độ dài 16 Bit và dữ liệu kiểu DWord có độ dài 32 Bit.

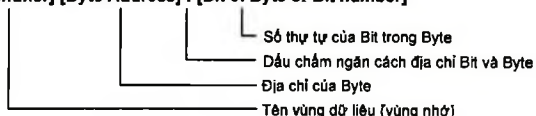
Bảng 3.2 chỉ ra dài giá trị mà các kiểu dữ liệu có thể biểu diễn được.

Bảng 3.2. Dài giá trị mà các kiểu dữ liệu có thể biểu diễn

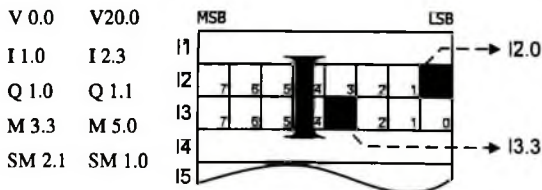
Kiểu dữ liệu	Số nguyên không dấu		Số nguyên có dấu	
	Thập phân	HEXA	Thập phân	HEXA
B (Byte – giá trị 8 Bit)	0 đến 255	0 đến FF	-128 đến 127	80 đến 7F
W (Word – Từ, giá trị 16 Bit)	0 đến 65.535	0 đến FFFF	-32.768 đến 32.767	8000 đến 7FFF
D (Double Word – Từ kép, giá trị 32 Bit, còn gọi là DWord)	0 đến 4.294.967.295	0 đến FFFF FFFF	-2.147.483.648 đến 2.147.483.647	8000 000 đến 7FFF FFFF

– Địa chỉ Bit có dạng sau:

[Area identifier][Byte Address] . [Bit of Byte or Bit number]

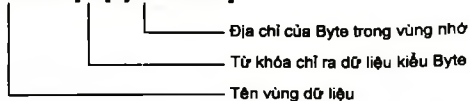


Ví dụ:



– Địa chỉ Byte có dạng sau:

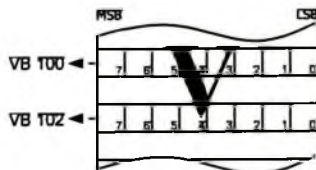
[Area Identifier] B [Byte Address]



Ví dụ:

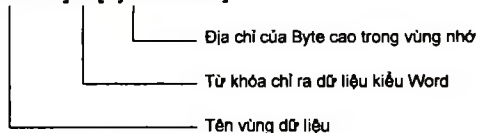
VB 100

VB 102

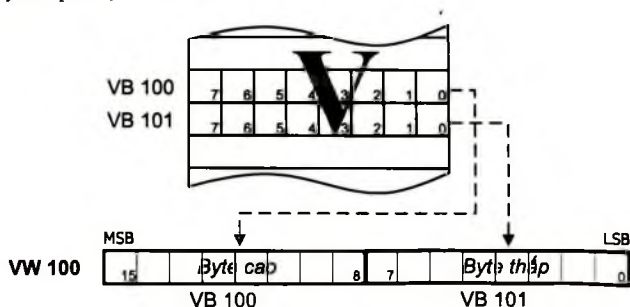


– Dữ liệu kiểu Word gồm 2 Byte có địa chỉ liền nhau. Quy ước Byte cao có địa chỉ thấp, Byte thấp có địa chỉ cao. Địa chỉ Word có dạng sau:

[Area Identifier] W [Byte Address]

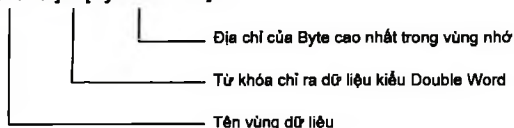


Ví dụ: VW 100 là địa chỉ của 1 Word gồm 2 Byte: Byte cao có địa chỉ VB 100 và Byte thấp có địa chỉ VB 101.

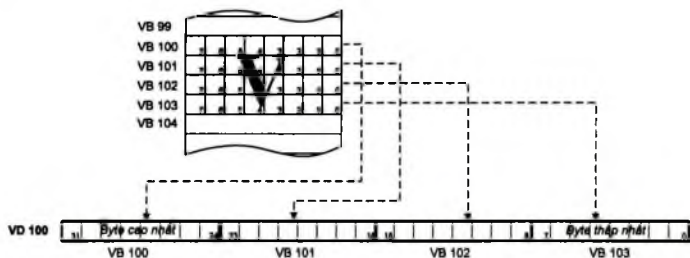


Dữ liệu kiểu DWord gồm 4 Byte có địa chỉ liên nhau. Quy ước Byte cao nhất có địa chỉ thấp nhất, Byte thấp nhất có địa chỉ cao nhất. Địa chỉ của DWord có dạng sau:

[Area Identifier] D [Byte Address]

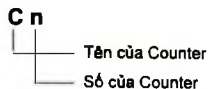
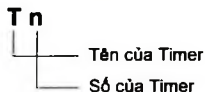


Ví dụ: VD 100 là địa chỉ của DWord gồm 4 Byte. Byte cao nhất có địa chỉ VB 100, Byte thấp nhất có địa chỉ VB 103.



Bộ nhớ dành cho các thiết bị gồm có 6 vùng: TIMER (T), COUNTER (C), đầu vào tương tự (AI), đầu ra tương tự (AO), thanh chứa (AC) và bộ đếm tốc độ cao (HC). Mỗi loại thiết bị gồm một tập các kiểu dữ liệu cấu thành. Việc truy nhập đến kiểu dữ liệu của thiết bị do lệnh quy định.

– Địa chỉ của TIMER và COUNTER có dạng:



Tn và Cn có thể là dạng Bit hoặc dạng Word (16 Bit) tùy thuộc câu lệnh. Dữ liệu dạng Bit là cờ hoàn thành (*Complement Flag*) của TIMER, COUNTER. Dữ liệu dạng Word là từ 16 Bit chứa giá trị hiện thời (*PV-Present Value*) của TIMER, COUNTER. Khi giá trị hiện thời bằng hoặc lớn hơn giá trị đặt (*SV-Set Value*) thì cờ hoàn thành của TIMER, COUNTER sẽ được thiết lập 1 (ON).

Ví dụ: T0, C1,...

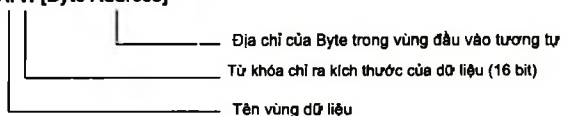
15	0	Bit
T0		T0
Giá trị hiện thời <i>Present Value (PV)</i>		Cờ

15	0	Bit
C1		C1
Giá trị hiện thời <i>Present Value (PV)</i>		Cờ

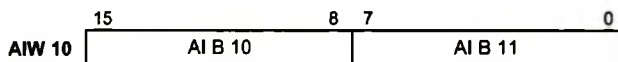
Vùng dữ liệu dành cho mỗi đầu vào hoặc mỗi đầu ra tương tự có kích thước 16 Bit. Như vậy, việc truy nhập đến các vùng nhớ này là theo dạng từ (*Word*), Byte cao có địa chỉ thấp và Byte thấp có địa chỉ cao.

– Địa chỉ của đầu vào tương tự (*Analog Input*) có dạng:

AI W [Byte Address]

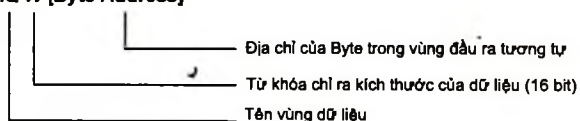


Ví dụ: AIW 10.

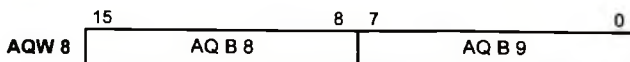


– Địa chỉ của đầu ra tương tự (*Analog Output*) có dạng:

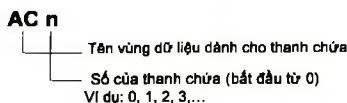
AQ W [Byte Address]



Ví dụ: AQW 8

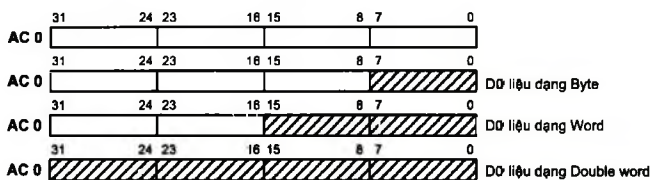


– Thanh chứa (*ACcumulator*) có độ dài 32 Bit. Địa chỉ của thanh chứa là:

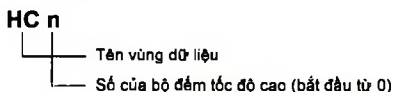


Kiểu dữ liệu có thể truy nhập đến thanh chứa ở dạng Byte, Word, Double Word tùy thuộc vào câu lệnh sử dụng. Nếu truy nhập dạng Byte thì dữ liệu được chứa trong Byte trẻ nhất của thanh chứa. Nếu truy nhập dạng Word thì dữ liệu được chứa trong 2 Byte thấp nhất của thanh chứa. Nếu truy nhập dạng Double Word thì dữ liệu chứa trong cả 4 Byte của thanh chứa.

Ví dụ: AC 0 là địa chỉ của thanh chứa đầu tiên.



Bộ đếm tốc độ cao (*High speed COUNTER*) là thiết bị đặc biệt để đếm sự kiện xảy ra nhanh hơn chu kỳ quét. Vùng nhớ để lưu giữ giá trị hiện tại của bộ đếm tốc độ cao có độ dài 32 Bit. Địa chỉ của vùng nhớ này như sau:



Ta chỉ có thể truy nhập đến vùng nhớ này ở dạng Double Word (32 Bit).

Ví dụ: HC 0



Tóm tắt về các vùng dữ liệu và các phương pháp truy nhập các vùng dữ liệu được trình bày trong bảng 3.3.

Bảng 3.3. Các vùng dữ liệu và phương pháp truy nhập

Vùng dữ liệu	Mô tả	Truy nhập theo Bit	Truy nhập theo Byte	Truy nhập theo Word	Truy nhập theo DWord	Có thể duy trì (Retentive)	Có thể áp đặt (Force)
I	Các đầu vào rời rạc và thanh ghi ảnh đầu vào	đọc/ghi	đọc/ghi	đọc/ghi	đọc/ghi	Không	Có
Q	Các đầu ra rời rạc và thanh ghi ảnh đầu ra	đọc/ghi	đọc/ghi	đọc/ghi	đọc/ghi	Không	Có
M	Các Bit làm việc ở bộ nhớ trong	đọc/ghi	đọc/ghi	đọc/ghi	đọc/ghi	Có	Có
SM	Các Bit ở vùng nhớ đặc biệt (SM0 ÷ SM29 chỉ đọc)	đọc/ghi	đọc/ghi	đọc/ghi	đọc/ghi	Không	Không
V	Vùng nhớ biến	đọc/ghi	đọc/ghi	đọc/ghi	đọc/ghi	Có	Có
T	Các TIMER định thời và TIMER Bit	T-Bit đọc/ghi	Không	T- định thời đọc/ghi	Không	T- định thời – có, T-Bit – không	Không
C	Các COUNTER định thời và COUNTER Bit	C-Bit đọc/ghi	Không	C- định thời đọc/ghi	Không	C- định thời – có, C-Bit – không	Không
HC	Các bộ đếm tốc độ cao	Không	Không	Không	chỉ cho đọc	Không	Không
AI	Các đầu vào tương tự	Không	Không	chỉ cho đọc	Không	Không	Có
AQ	Các đầu ra tương tự	Không	Không	chỉ cho đọc	Không	Không	Có
AC	Các thanh chứa	Không	đọc/ghi	đọc/ghi	đọc/ghi	Không	Không
L	Vùng nhớ biến cục bộ	đọc/ghi	đọc/ghi	đọc/ghi	đọc/ghi	Không	Không
S	SCR	đọc/ghi	đọc/ghi	đọc/ghi	đọc/ghi	Không	Không

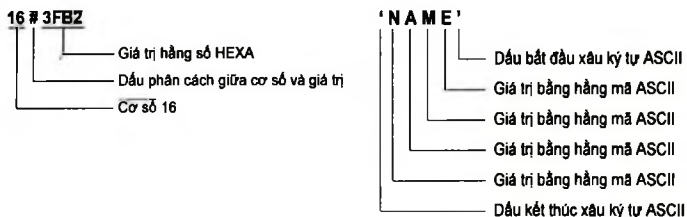
3.3.1.4. Hằng và kiểu dữ liệu

Người sử dụng có thể sử dụng các hằng ở dạng Byte, Word hoặc DWord tùy thuộc vào lệnh. Các hằng có thể được biểu diễn ở dạng thập phân (*Decimal*), mã Hexa (*Hexadecimal*) hoặc mã ASCII. Tuy nhiên, các giá trị được lưu giữ trong bộ nhớ (Byte, Word hoặc Double Word) chỉ là các số nhị phân biểu diễn số thập phân, mã Hexa, mã ASCII hoặc chính mã nhị phân. Trong PLC, kiểu dữ liệu không cần phải khai báo trước khi sử dụng mà nó hoàn toàn được xác định trong câu lệnh.

Ví dụ: lệnh ADD sử dụng nội dung của VW100 là số nguyên có dấu, trong khi lệnh OR sử dụng nội dung của VW100 là số nhị phân không dấu. Bởi vì lệnh ADD là lệnh cộng số học trong khi lệnh OR là lệnh cộng logic (phép Hoặc).

Các hằng số trong câu lệnh thông thường địa chỉ biểu diễn ở dạng thập phân. Để chỉ ra các số không là thập phân thì người sử dụng phải chỉ ra kiểu của hằng.

Ví dụ:



3.3.1.5. Vùng dữ liệu đặc biệt (SM—Special Memory)

Vùng SM lưu giữ các từ hoặc Bit trạng thái (*Status Bit*). Các từ hoặc Bit này có chức năng điều khiển một số hoạt động đặc biệt của PLC. Việc truy nhập đến vùng nhớ SM có thể thực hiện ở dạng Bit, Byte, Word hoặc DWord. Ngoài ra, vùng SM được chia làm hai phần: vùng chỉ đọc (*Read Only*) và vùng có thể đọc/viết (*Read/Write*). Vùng chỉ đọc bao gồm các Byte SM0 ÷ SM29. Dữ liệu ở vùng này là các trạng thái hoạt động của PLC do hệ thống cập nhật trong quá trình hoạt động. Vùng có thể đọc/viết bao gồm các Byte SM30 ÷ SM85, được sử dụng để chọn và điều khiển các chức năng đặc biệt của PLC.

Một số vùng nhớ của SM được trình bày dưới đây (chi tiết xem thêm tài liệu hướng dẫn [6]).

3.3.1.6. Byte trạng thái SMB0

Các Bit trạng thái của Byte SMB0 (SM0.0 ÷ SM0.7) được cập nhật cuối mỗi chu kì quét. Mô tả các Bit trạng thái của Byte SMB0 được trình bày trên bảng 3.4.

Bảng 3.4. Mô tả các Bit trạng thái của Byte SMB0

Các Bit SM	Mô tả (Read Only)
SM0.0	Bit luôn ON.
SM0.1	Bit này ON trong vòng quét đầu tiên.
SM0.2	Bit này ON trong vòng quét đầu tiên nếu vùng nhớ duy trì bị mất.
SM0.3	Bit này ON trong một vòng quét khi chế độ RUN được thiết lập.
SM0.4	Bit này cho một xung đồng hồ chu kì 1 phút: thời gian ON là 30 giây, thời gian OFF là 30 giây.
SM0.5	Bit này cho một xung đồng hồ chu kì 1 giây: thời gian ON là 0,5 giây, thời gian OFF là 0,5 giây.
SM0.6	Bit này cho một xung có chu kì bằng hai chu kì quét, ON ở một chu kì quét và OFF ở chu kì quét tiếp theo.
SM0.7	Bit này phản ánh trạng thái của chuyển mạch chọn Mode trên CPU.

3.3.1.7. Byte trạng thái SMB1

Các Bit trạng thái của Byte SMB1 (SM1.0+SM1.7) là các cờ trạng thái khi CPU thực hiện lệnh được trình bày trên bảng 3.5.

Bảng 3.5. Mô tả các Bit trạng thái của Byte SMB1

Các Bit SM	Mô tả (Read Only)
SM1.0	Cờ zero. Bit này ON khi kết quả thực hiện lệnh bằng 0.
SM1.1	Cờ tràn. Bit này ON khi kết quả thực hiện lệnh tràn ô nhớ.
SM1.2	Cờ dấu. Bit này ON khi kết quả thực hiện lệnh số học là âm.
SM1.3	Cờ chia cho 0. Bit này ON khi thực hiện phép chia cho số 0.
SM1.4	Cờ trạng thái khi thực hiện lệnh cộng (ADD). Bit ON khi bảng được điền đầy.
SM1.5	Cờ trạng thái khi thực hiện lệnh LIFO (Last In–First Out) và FIFO (First In–First Out). Bit này ON khi đọc bảng rỗng.
SM1.6	Cờ trạng thái lật ON khi thực hiện lệnh biến đổi từ số không phải BCD sang số nhị phân.
SM1.7	Cờ trạng thái lật ON khi giá trị ASCII không biến đổi được thành số HEXA.

3.3.1.8. Một số các Byte trạng thái khác

SMB2: Freeport Receive Character

Chứa mã ký tự nhận được ở chế độ trao đổi thông tin.

SMB3: Freeport Parity Error

Chứa Bit báo lỗi chẵn lẻ (*Parity*). Cụ thể là Bit SM3.0: OFF khi không có lỗi, ON khi phát hiện lỗi.

SMB4: Queue Overflow

Chứa các Bit trạng thái khi thực hiện lệnh ngắt kiểu hàng chờ (*Queue*).

SMB5: I/O Status

Chứa các Bit trạng thái khi báo lỗi vào/ra.

SMB6: CPU ID Register

Thanh ghi mã nhận dạng của CPU.

SMB7: Reserved

Byte dành trước cho các ứng dụng tương lai.

SMB8 to SMB21: I/O Module ID and Error Registers

Các thanh ghi mã nhận dạng và mã lỗi của các module vào/ra mở rộng. Các Byte này được tổ chức thành từng cặp, mỗi cặp chứa mã nhận dạng module vào/ra và 1 Byte chứa mã lỗi.

SMW22 to SMW26: Scan Times

Các từ cung cấp các thông tin về thời gian quét của PLC.

SMB28 and SMB29: Analog Adjustment

Các Byte lưu giữ giá trị bằng số của các biến trở trên CPU.

SMB30 and SMB130: Freeport Control Registers

Các thanh ghi điều khiển các cổng truyền thông nối tiếp ở chế độ hoạt động Freeport. SMB30 điều khiển cổng 0, SMB130 điều khiển cổng 1.

SMB31 and SMW32: Permanent Memory (EEPROM) Write Control

Byte và từ điều khiển quá trình ghi vào bộ nhớ duy trì.

SMB34 and SMB35: Time Interval Registers for Timed Interrupts

Các thanh ghi điều khiển ngắt định thời.

SMB36 to SMB65: HSC0, HSC1, and HSC2 Register

Các thanh ghi được dùng để điều khiển và theo dõi các hoạt động của bộ đếm tốc độ cao.

SMB66 to SMB85: PTO/PWM Registers

Các thanh ghi được dùng để điều khiển và theo dõi các đầu ra xung và điều chế độ rộng xung.

Chi tiết về cấu trúc dữ liệu của S7-200, người đọc có thể tham khảo các tài liệu hướng dẫn lập trình hoặc tài liệu kèm theo thiết bị hoặc website của hãng SIEMENS [6].

3.3.2. Họ SYSMAC của OMRON [4]

PLC của hãng OMRON có nhiều dòng (Series) từ thấp đến cao. Tuy nhiên, tổ chức bộ nhớ và cấu trúc dữ liệu của PLC họ SYSMAC là thống nhất. Sự khác nhau chủ yếu là ở kích thước các vùng dữ liệu và các vùng nhớ dữ liệu mở rộng ở các PLC cấp cao. Đa số các PLC họ SYSMAC đều quy ước kích thước một từ dữ liệu là 16 Bit. Cấu trúc dữ liệu điển hình của PLC họ SYSMAC gồm các vùng nhớ cơ bản được trình bày trong bảng 3.6.

Bảng 3.6. Cấu trúc dữ liệu của PLC họ SYSMAC (OMRON)

STT	Vùng dữ liệu	Mô tả	Ghi chú
1	IR Vùng nhớ bên trong (Internal Relay)	Vùng ảnh đầu vào (INPUT area)	Chiếm vùng nhớ từ địa chỉ IR000
		Vùng ảnh đầu ra (OUTPUT area)	Chiếm vùng nhớ từ địa chỉ IR100
		Vùng Bit trung gian (Work area)	Lưu giữ các kết quả trung gian được sử dụng trong chương trình
		Vùng dành cho MACRO	Dùng khi sử dụng các lệnh MACRO để nhận dữ liệu từ I/O
		Vùng dành cho bộ đếm tốc độ cao (High-speed COUNTER)	Lưu giữ giá trị hiện thời của bộ đếm tốc độ cao (Preset Value)
2	SR Vùng nhớ đặc biệt (Special Relay)	Lưu giữ các cờ trạng thái và các Bit điều khiển. Được sử dụng cho các chức năng đặc biệt.	Có thể truy nhập vùng nhớ này ở dạng Word hoặc dạng Bit
3	TR Vùng nhớ tạm thời (Temporary area)	Gồm các Bit phục vụ việc rẽ nhánh chương trình	Chỉ truy nhập được ở dạng Bit
4	HR Vùng nhớ duy trì (Hold Relay area)	Lưu giữ các Bit duy trì kể cả khi ngắt nguồn	Có thể truy nhập vùng nhớ này ở dạng Word hoặc dạng Bit

Bảng 3.6. Cấu trúc dữ liệu của PLC họ SYSMAC (OMRON) (tiếp)

5	AR Vùng nhớ phụ (Auxiliary area)	Lưu giữ cờ và các Bit điều khiển được sử dụng cho các chức năng đặc biệt như thiết lập cổng truyền thông, đặt tốc độ trao đổi truyền thông nối tiếp	Có thể truy nhập vùng nhớ này ở dạng Word hoặc dạng Bit
6	LR Vùng nhớ liên kết (Link Relay area)	Được sử dụng làm vùng dữ liệu chung phục vụ việc trao đổi dữ liệu giữa các PLC	Có thể truy nhập vùng nhớ này ở dạng Word hoặc dạng Bit
7	TC (Vùng nhớ dành cho TIMER /COUNTER)	Lưu giữ các kiểu dữ liệu của TIMER/COUNTER	
8	DM Vùng nhớ dữ liệu (Data Memory area)	Lưu giữ số liệu, duy trì khi mất nguồn. Vùng này còn được sử dụng để cài đặt cấu hình và chế độ hoạt động của PLC.	Chỉ có thể truy nhập ở dạng từ (Word). Bao gồm một số vùng chỉ đọc (Read Only) cho hệ thống và một số vùng cho phép đọc/viết (Read/Write)

Một số lưu ý khi sử dụng vùng nhớ của PLC họ SYSMAC:

– Đối với các vùng nhớ như AR, LR: Nếu không sử dụng vào mục đích đặc biệt do hệ thống quy định thì có thể sử dụng chúng như các Bit trung gian. Một số PLC họ SYSMAC thế hệ mới như CJ, CH... sử dụng dạng cấu trúc dữ liệu mới. Tuy nhiên, về tổ chức thì căn bản không có sự khác biệt so với các PLC cũ. Việc mở rộng các vùng nhớ mới phục vụ cho việc thiết lập truyền thông trên mạng và quản lý dữ liệu.

– Có thể truy nhập các vùng IR, SR, LR, HR, AR ở dạng Bit hoặc Word. Vùng dữ liệu DM chỉ có thể truy nhập ở dạng Word.

– Vùng dữ liệu TC dành cho TIMER/COUNTER có thể truy nhập ở dạng Word (giá trị hiện tại PV của TIMER/COUNTER) hoặc dạng Bit (cờ hoàn thành của TIMER/COUNTER) tùy thuộc vào câu lệnh được sử dụng. Địa chỉ của vùng TC là số thứ tự của TIMER/COUNTER và chỉ được dùng một lần trong chương trình.

Ví dụ: Cấu trúc dữ liệu của PLC CQM1H-CPU21 của OMRON trình bày trong bảng 3.7.

Bảng 3.7. Cấu trúc dữ liệu của CQM1H

Vùng dữ liệu		Kích thước	Địa chỉ từ	Địa chỉ Bit
Vùng IR	Vùng đầu vào	256 Bit	IR 000 ÷ IR 015	IR 00000 ÷ IR 01515
	Vùng đầu ra	256 Bit	IR 100 ÷ IR 115	IR 10000 ÷ IR 11515
	Các vùng làm việc	2,528 Bit	IR 016 ÷ IR 089	IR 01600 ÷ IR 08915
			IR 116 ÷ IR 189	IR 11600 ÷ IR 18915
			IR 216 ÷ IR 219	IR 21600 ÷ IR 21915
			IR 224 ÷ IR 229	IR 22400 ÷ IR 22915
Vùng trạng thái liên kết điều khiển		96 Bit	IR 090 ÷ IR 095	IR 09000 ÷ IR 09515
			IR 190 ÷ IR 195	IR 19000 ÷ IR 19515
Vùng toán hạng MACRO	Vùng đầu vào	64 Bit	IR 096 ÷ IR 099	IR 09600 ÷ IR 09915
	Vùng đầu ra	64 Bit	IR 196 ÷ IR 199	IR 19600 ÷ IR 19915
Vùng nhớ khe cắm slot 1		256 Bit	IR 200 ÷ IR 215	IR 20000 ÷ IR 21515
Vùng cấu hình tương tự		64 Bit	IR 220 ÷ IR 223	IR 22000 ÷ IR 22315
Giá trị hiện thời bộ đếm tốc độ cao COUNTER0		32 Bit	IR 230 ÷ IR 231	IR 23000 ÷ IR 23115
Inner Board slot 2 area		192 Bit	IR 232 ÷ IR 243	IR 23200 ÷ IR 24315
Vùng SR		184 Bit	SR 244 ÷ SR 255	SR 24400 ÷ SR 25507
Vùng HR		1,600 Bit	HR 00 ÷ HR 99	HR 0000 ÷ HR 9915
Vùng AR		448 Bit	AR 00 ÷ AR 27	AR 0000 ÷ AR 2715
Vùng TR		8Bit	—	TR 0 ÷ TR7
Vùng LR		1,024 Bit	LR 00 ÷ LR 63	LR 0000 ÷ LR 6315
Vùng TIMER/COUNTER		512 Bit	TIM/CNT 000 ÷ TIM/CNT 511	

Lưu ý xem thêm hệ lệnh MACRO trong hệ lệnh.

Để minh họa, một số vùng dữ liệu điển hình sẽ được trình bày sau đây:

– Vùng các Bit đặc biệt SR lưu giữ các cờ trạng thái hoạt động và các Bit điều khiển. Vùng này gồm 12 Word từ SR244 ÷ SR255. Trong đó 8 Word từ SR244 ÷ SR251 dành cho điều khiển ngắt. Các Bit SR253.00 ÷ SR255.15 là các cờ trạng thái.

Một số Bit của vùng Bit nhớ đặc biệt của PLC họ SYSMAC được trình bày trên bảng 3.8.

Bảng 3.8. Các Bit đặc biệt

SR 253	SR 253.13	Cờ luôn ON
	SR 253.14	Cờ luôn OFF
	SR 253.15	Cờ quét lần đầu, chỉ được thiết lập trong một chu kì quét đầu tiên
SR 254	SR 254.00	Xung 1 phút (30 giây ON; 30 giây OFF)
	SR 254.01	Xung 0,02 giây (0,01 giây ON; 0,01 giây OFF)
SR 255	SR 255.00	Xung 0,1 giây (0,05 giây ON; 0,05 giây OFF)
	SR 255.01	Xung 0,2 giây (0,1 giây ON; 0,1 giây OFF)
	SR 255.02	Xung 1 giây (0,5 giây ON; 0,5 giây OFF)
	SR 255.04	Cờ nhớ (CY)
	SR 255.05	Cờ lớn hơn (GR)
	SR 255.06	Cờ bằng nhau (EQ)
	SR 255.07	Cờ nhỏ hơn (LE)

- Vùng AR bao gồm các Bit trạng thái và điều khiển PLC liên quan đến cổng truyền thông nối tiếp.
- Vùng DM6600 ÷ DM6655 được sử dụng để thiết lập chế độ hoạt động cho PLC theo yêu cầu của người sử dụng.

3.3.3. Họ SLC500 của Allen–Bradley [3]

Cấu trúc dữ liệu của họ SLC500 (bao gồm cả Micrologix) được tổ chức thành các File dữ liệu (*Data File*). Các File dữ liệu chứa thông tin trạng thái liên quan đến I/O, dữ liệu các lệnh của chương trình, cấu hình hệ thống, trạng thái hoạt động...

Để địa chỉ hóa, mỗi kiểu File dữ liệu được xác định bởi tên File bằng chữ cái và số File. Bảng 3.9 trình bày các kiểu File dữ liệu của họ SLC500.

Mỗi kiểu File dữ liệu có một tên riêng bằng một chữ cái. Ví dụ: File dữ liệu của vùng ảnh đầu vào có tên là I, File dữ liệu của vùng ảnh đầu ra có tên là O, File dữ liệu của vùng các Bit trung gian là B... Tổng số có 256 File dữ liệu, từ File số 0 đến File số 255. Trong đó 9 File đầu tiên từ File số 0 đến File số 8 do hệ thống quy định. Ví dụ

File ảnh đầu ra có số File là 0, File ảnh đầu vào có số File là 1, File dữ liệu dành cho TIMER là File số 4... Các File dữ liệu từ số 9 đến số 255 do người sử dụng định nghĩa và hoàn toàn linh hoạt. Ví dụ File dữ liệu của vùng Bit trung gian có thể là File số 9, File số 10... File dữ liệu dành cho TIMER là File số 11... Cần chú ý là hai File dữ liệu dành cho ảnh đầu ra (*Output*) và ảnh đầu vào (*Input*) không được định nghĩa trong vùng số File từ 9 ÷ 255.

Bảng 3.9. Các File dữ liệu của họ SLC500

Các File dữ liệu của hệ thống

<i>Kiểu File</i>	<i>Tên File</i>	<i>Số File</i>
Output (đầu ra)	O	0
Input (đầu vào)	I	1
Status (trạng thái)	S	2
Bit	B	3
TIMER (bộ định thời)	T	4
COUNTER (bộ đếm)	C	5
Control (điều khiển)	R	6
Integer (số nguyên)	N	7
Float (số thực)	F	8

Các File dữ liệu do người sử dụng định nghĩa

<i>Kiểu File</i>	<i>Tên File</i>	<i>Số File</i>
Bit	B	9 ÷ 255
TIMER (bộ định thời)	T	
COUNTER (bộ đếm)	C	
Control (điều khiển)	R	
Integer (số nguyên)	N	
Float (số thực)	F	
String (chuỗi kí tự)	St	
ASCII	A	

Kích thước của từng kiểu File dữ liệu (trừ File số 0 đến File số 2) hoàn toàn tùy thuộc vào người sử dụng. Kích thước của từng File dữ liệu tùy thuộc vào kiểu File.

Tổ chức File dữ liệu gồm có các phần tử (*Element*). Mỗi phần tử chứa từ 1 đến 3 Word dữ liệu (1 Word = 16 Bit). Địa chỉ hóa File dữ liệu là địa chỉ hóa phần tử, địa chỉ hóa Word và địa chỉ hóa Bit của nó.

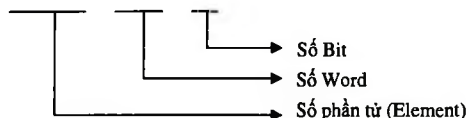
Ví dụ: File dữ liệu ảnh đầu ra và File dữ liệu ảnh đầu vào gồm các phần tử kích thước 1 Word. Mỗi phần tử được xác định bởi số thứ tự của khe cắm (*Slot*) và số thứ tự

của Word trong vùng dữ liệu. File dữ liệu của TIMER, COUNTER, CONTROL, ASCII, có phần tử kích thước 3 Word. File dữ liệu STATUS, BIT, INTEGER có phần tử kích thước 1 Word. File dữ liệu STRING có phần tử kích thước 42 Word.

Định dạng địa chỉ của các File dữ liệu rất khác nhau, phụ thuộc vào kiểu File dữ liệu. Điều đó được trình bày ở các mục dưới đây.

Định dạng địa chỉ chung là:

<Tên File><Số File>:<Element>.<Word>/<Bit>

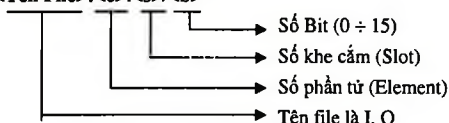


3.3.3.1. File dữ liệu số 0 và số 1

File dữ liệu số 0 và số 1 là File dữ liệu ảnh đầu ra và ảnh đầu vào. Mỗi File dữ liệu này có 256 Word. Về lý thuyết, trong trường hợp tổng quát, mỗi Word tương ứng với 1 vị trí của khe cắm module vào/ra trên bảng mạch Bus và số Bit tương ứng với số các đầu vào/ra trên module vào/ra. Các Bit không sử dụng trong một Word sẽ không được sử dụng tiếp theo. Tuy nhiên, trong thực tế, trên một module vào/ra vật lý có thể chứa số điểm đầu vào, số điểm đầu ra khác nhau, ví dụ 8, 16, 32... Thậm chí có thể tích hợp cả đầu vào và đầu ra trên một module. Vì vậy, địa chỉ hóa vào/ra cần phải tính đến cả vị trí của khe cắm trên bảng mạch Bus. Vấn đề này đã được xét trong mục III chương 2.

Định dạng của địa chỉ vào/ra được biểu diễn như sau:

<Tên File>:<e>.<s>/



Định dạng trên không yêu cầu số File vì File số 0 và File số 1 chỉ có trong vùng do hệ thống quản lý.

Ví dụ: I:0.1/7 : Đầu vào Bit 7, Word 1, Slot 0.

O:0/4 : Đầu ra Bit 4, Slot 0.

O:1 : Đầu vào Word 0, Slot 1.

3.3.3.2. File dữ liệu số 2 (STATUS)

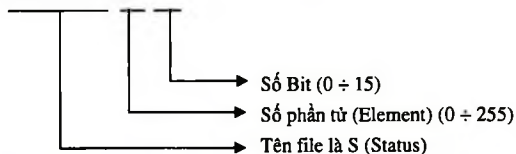
File dữ liệu số 2 là File lưu giữ các trạng thái của CPU, gọi là các cờ trạng thái. Kích thước của File số 2 tùy thuộc vào kiểu CPU, được trình bày ở bảng sau:

Kiểu CPU	Kích thước File số 2 (Word)
SLC 5/01	16
SLC 5/02	32
SLC 5/03	83
SLC 5/04	97

File số 2 chỉ có trong vùng do hệ thống định nghĩa và nội dung của nó không thể ghi hoặc xóa được từ chương trình.

Định dạng địa chỉ của File số 2 được biểu diễn như sau:

<Tên File>: <e>/



Mỗi phần tử của File số 2 chứa 1 Word (16 Bit).

Có thể truy cập dữ liệu của File số 2 ở dạng Word hoặc Bit.

Ví dụ: S: 1/7 : Element 1, Bit 7

S: 3 : Element 3

Một số các trạng thái đặc biệt của hệ thống được lưu giữ trong File số 2 như sau:

S: 0 : Chứa các cờ số học, các cờ này được thiết lập khi CPU thực hiện các phép toán số học, logic hoặc các lệnh chuyển dữ liệu. Cụ thể các cờ sau:

S: 0/0 : Cờ nhớ (Carry)

S: 0/1 : Cờ tràn (Over flow)

S: 0/2 : Cờ zero

S: 0/3 : Cờ dấu (Sign)

S: 0/15 : Cờ First scan

Chi tiết và ảnh hưởng của các cờ có thể xem thêm trong tài liệu hướng dẫn lập trình [3].

3.3.3.3. File dữ liệu số 3 (Bit)

File dữ liệu số 3 lưu giữ các Bit trung gian (*Work Bit*) sử dụng cho các lệnh Bit. Kích thước của File số 3 là 256 Word. Mỗi Element chứa 1 Word. Như vậy tổng số có 4096 Bit. Địa chỉ Bit có thể được xác định bằng 2 cách sau:

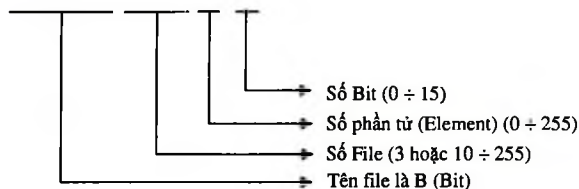
Cách thứ nhất: Xác định bằng số Element ($0 \div 255$) và số Bit ($0 \div 15$) trong Element đó.

Cách thứ hai: Đánh số Bit từ $0 \div 4095$.

Định dạng địa chỉ Bit được biểu diễn theo hai cách như sau:

Cách 1:

<Tên File><Số File>:<e>/



Cách 2:

<Tên File><Số File>/



Sơ đồ biểu diễn File số 3 được trình bày trên bảng sau:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Element
																B3:0
																B3:1
																B3:2
	X															B3:3

																B3:252
																B3:253
																B3:254
																B3:255

3.3.3.4. File dữ liệu số 4 (TIMER)

File dữ liệu số 4 dành cho TIMER do hệ thống định nghĩa. Kích thước của File dữ liệu số 4 là 256 Element. Mỗi Element chứa 3 Word và được biểu diễn như sau:

TIMER ELEMENT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Word
EN	TT	DN														0
Giá trị đặt (<i>Present</i>) – PRE																1
Giá trị tích lũy (<i>Accumulated</i>) – ACC																2

Trong đó, Word 0 là từ điều khiển, Word 1 chứa giá trị đặt (*Present Value*) và Word 2 chứa giá trị tích lũy (*Accumulated Value*).

Các Bit (0 ÷ 7) của Word 0 không địa chỉ hóa được.

Các Bit (8 ÷ 15) của Word 0 có thể địa chỉ hóa được.

Đó là:

Bit 15 có tên là EN (Enable);

Bit 14 có tên là TT (TIMER Timming);

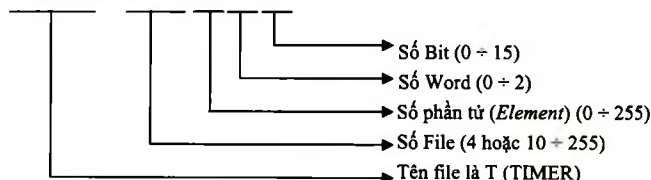
Bit 13 có tên là DN (Done).

Word 1 địa chỉ hóa PRE và Word 2 địa chỉ hóa là ACC.

Người sử dụng có thể truy nhập đến các Bit địa chỉ hóa được của Word 0 và có thể truy nhập được dữ liệu dạng Word hoặc Bit đối với các Word 1 và Word 2.

Định dạng địa chỉ của File dữ liệu dành cho TIMER được biểu diễn như sau:

<Tên File><Số File>.<e>.<w>/



Ví dụ: T4:0 : File số 4 dành cho TIMER, Element 0
T10:0 : File số 10 dành cho TIMER, Element 0
T4:0/15 = T4:0/EN : Enable Bit
T4:0/14 = T4:0/TT : TIMER Timming Bit
T4:0/13 = T4:0/DN : Done Bit
T4:0.1 = T4:0.PRE : Giá trị đặt (Word 1)
T4:0.2 = T4:0.ACC : Giá trị tích lũy (Word 2)
T4:0.1/0 : Bit 0 của Word 1, Element 0
T4:0.2/1 : Bit 1 của Word 2, Element 0

File dữ liệu dành cho TIMER có số File là 4 nếu nằm trong vùng do hệ thống định nghĩa hoặc có số File từ 10 ÷ 255 nếu nằm trong vùng do người sử dụng định nghĩa. Trong trường hợp số File là 4 thì trong địa chỉ có thể bỏ qua số File.

Ví dụ: T4:0/14 = T:0/14

3.3.3.5. File dữ liệu số 5 (COUNTER)

File dữ liệu số 5 dành cho COUNTER do hệ thống định nghĩa. Kích thước File dữ liệu số 5 là 256 Element. Mỗi Element chứa 3 Word được biểu diễn như sau:

COUNTER ELEMENT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Word
CU	CD	DN	OV	UN												0
Giá trị đặt (Present) – PRE																1
Giá trị tích lũy (Accumulated) – ACC																2

Trong đó Word 0 là từ điều khiển, Word 1 chứa giá trị đặt (Present Value) và Word 2 chứa giá trị tích lũy (Accumulated Value).

Các Bit (0 ÷ 7) của Word 0 không địa chỉ hóa được.

Các Bit (8 ÷ 15) của Word 0 có thể địa chỉ hóa được.

Đó là: Bit 15 có tên là CU (Count Up Enable)

Bit 14 có tên là CD (Count Down Enable)

Bit 13 có tên là DN (Done)

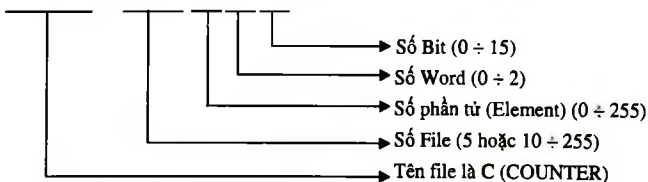
Bit 12 có tên là OV (Over flow)

Bit 11 có tên là UN (Under flow)

Word 1 địa chỉ hóa là PRE và Word 2 địa chỉ hóa là ACC.

Người sử dụng có thể truy nhập đến các Bit địa chỉ hóa được của Word 0 và có thể truy nhập được dữ liệu dạng Word hoặc Bit đối với các Word 1 và Word 2. Định dạng địa chỉ hóa của File dữ liệu dành cho COUNTER được biểu diễn như sau:

<Tên File><Số File>:<e>.<w>/



Ví dụ: C5:0 : File số 5 dành cho COUNTER, Element 0

C10:1 : File số 10 dành cho COUNTER, Element 1

C5:0/15 = C5:0/CU : Count Up Enable Bit

C5:0/14 = C5:0/CD : Count Down Enable Bit

C5:0/13 = C5:0/DN : Done Bit

C5:0.1 = C5:0.PRE : Giá trị đặt (Word 1)

C5:0.2 = C5:0.ACC : Giá trị tích lũy (Word 2)

File dữ liệu dành cho COUNTER có số File là 4 nếu nằm trong vùng do hệ thống định nghĩa hoặc có số File từ 10 ÷ 255 nếu nằm trong vùng do người sử dụng định nghĩa. Trong trường hợp có số File là 5 thì trong địa chỉ có thể bỏ qua số File.

Ví dụ: C5:0/14 = C:0/14

3.3.3.6. File dữ liệu số 6 (CONTROL)

File dữ liệu số 6 dùng để điều khiển hoạt động một số cách lệnh như Bit Shift, FIFO, LIFO, Sequencer,... Kích thước File dữ liệu số 6 là 256 Element. Mỗi Element chứa 3 Word được biểu diễn như sau:

CONTROL ELEMENT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Word
EN	EU	DN	EM	ER	UL	IN	FD	Error Code								0
Chiều dài của mảng Bit hoặc File: LEN																1
Con trỏ Bit hoặc vị trí Bit: POS																2

Trong đó, Word 0 là từ trạng thái, Word 1 chứa kích thước (chiều dài) của mảng dữ liệu và Word 2 xác định vị trí của mảng dữ liệu.

Các Bit (0 ÷ 7) của Word 0 tạo thành Byte mã lỗi.

Các Bit (0 ÷ 15) của Word 0 có thể địa chỉ hóa được là các Bit trạng thái.

Bit 15: EN (Enable)

Bit 14: EU (Unload Enable)

Bit 13: DN (Done)

Bit 12: EM (Stack Empty)

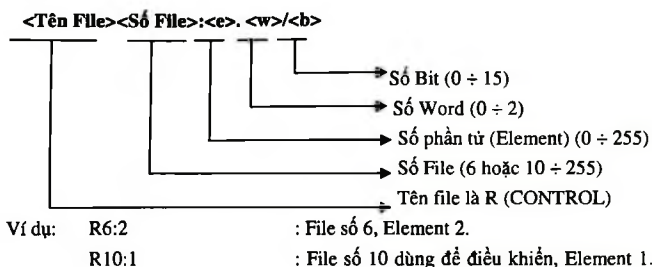
Bit 11: ER (Error)

Bit 10: UL (Unload)

Bit 9: IN (InhiBit)

Bit 8: FD (Found)

Word 1 địa chỉ hóa là LEN (*Length*) và Word 2 địa chỉ hóa là POS (*Position*). Người sử dụng có thể truy nhập đến các Bit địa chỉ hóa được của Word 0 và dữ liệu dạng Word hoặc Bit đối với các Word 1 và Word 2. Định dạng địa chỉ của File dữ liệu cho điều khiển được biểu diễn như sau:



R6:2/15 = R6:2/EN : Enable Bit
 R6:2/13 = R6:2/DN : Done Bit
 R6:2.1 = R6:2.LEN : Kích thước mảng dữ liệu.
 R6:2.2 = R6:2.POS : Vị trí mảng dữ liệu.
 R6:2.1/0 : Bit 0 của Word 1, Element 2.
 R6:2.2/0 : Bit 0 của Word 2, Element 2.

File dữ liệu dùng để điều khiển có số File là 6 nếu nằm trong vùng do hệ thống định nghĩa hoặc có số File từ 10 ÷ 255 nếu nằm trong vùng do người sử dụng định nghĩa. Trong trường hợp số File là 6 thì trong địa chỉ có thể bỏ qua số File.

Ví dụ: R6:2/EN = R:2/EN

3.3.3.7. File dữ liệu số 7 (INTEGER)

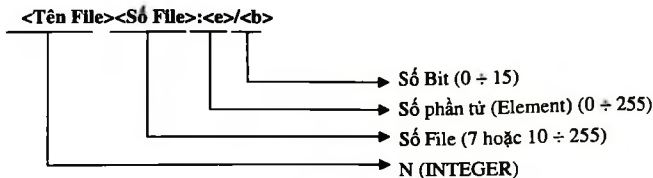
Element	0	1	2	3	4	5	6	7	8	9
N7:0	0	0	0	0	0	0	0	0	0	0
N7:10	0	0	0	1098	0	0	0	0	0	0
N7:20	0	0	0	0	0	0	195	0	0	0
...										
N7:240	0	0	0	0	0	0	0	0	0	66
N7:250	0	0	0	0	0	0				

Bảng trên biểu diễn File dữ liệu số 7 dùng để lưu giữ các số nguyên. Kích thước File dữ liệu số 7 là 256 Element. Mỗi Element chứa 1 Word.

Trong bảng trên, Element N7:13 chứa giá trị 1098, N7:26 chứa giá trị 195 và Element N7:249 chứa giá trị 66.

Người sử dụng có thể truy nhập đến dữ liệu ở dạng Word hoặc Bit.

Định dạng địa chỉ của File dữ liệu lưu giữ số nguyên được biểu diễn như sau:



Ví dụ: N7:2 : Element 2, File số 7
 N10:36 : Element 36, File số 10 lưu giữ số nguyên
 N7:2/8 : Bit 8 trong Element 2, File số 7

File dữ liệu lưu giữ số nguyên có số File là 7 nếu nằm trong vùng do hệ thống định nghĩa hoặc có số File từ 10 ÷ 255 nếu nằm trong vùng do người sử dụng định nghĩa. Trong trường hợp số File là 7 thì trong địa chỉ có thể bỏ qua số File.

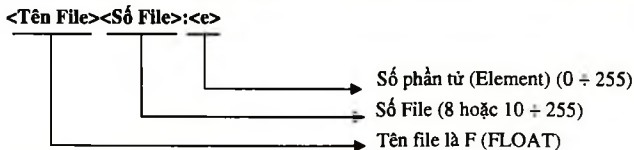
Ví dụ: R7:2/8 = R:2/8

3.3.3.8. File dữ liệu số 8 (FLOATING Point)

File dữ liệu số 8 dùng để lưu giữ các số dấu phẩy động. Kích thước của File này là 256 Element. Mỗi Element chứa 2 Word và được biểu diễn như sau:

Address	0	1	2	3	4
F8:0	0.000000	0.000000	0.000000	0.000000	
F8:5					
F8:10					
F8:15					
F8:20					
F8:25					
F8:30					

Người sử dụng chỉ truy nhập được dữ liệu ở dạng Element. Định dạng như sau:



Cần chú ý là dữ liệu ở dạng Element có 32 Bit và không mở rộng được.

Ví dụ: F8:2 : Element 2, File số 8
 F10:36 : Element 36, File số 10 dành cho số dấu phẩy động

3.3.3.9. Một số File dữ liệu khác

Các File dữ liệu do người sử dụng định nghĩa có số File từ $(9 \div 255)$. File dữ liệu ASCII có kích thước 256 Element. Mỗi Element chứa 1 Word. Định dạng địa chỉ của File dữ liệu ASCII được biểu diễn như sau:

<Tên File><Số File>:<e>/

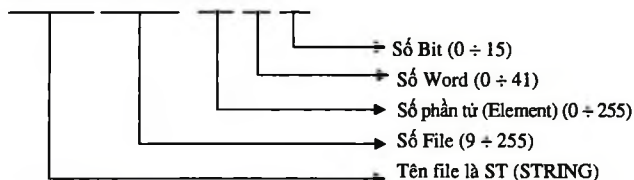


Ví dụ: A9:2 : Element 2, File dạng ASCII số 9.

A10:0/7 : Bit 7, Element 0, File dạng ASCII số 10.

File dữ liệu dạng chuỗi ký tự (STRING) có kích thước 256 Element. Mỗi Element chứa 42 Word. Người sử dụng có thể truy nhập đến dữ liệu ở dạng Element (42 Word), dạng Word trong Element và dạng Bit trong Word. Định dạng địa chỉ của File dữ liệu dạng String được biểu diễn như sau:

<Tên File><Số File>:<e>. <w>/

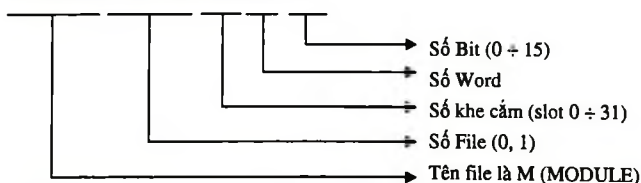


Ví dụ: ST9:2 : Element 2, File kiểu String số 9.

ST10:2.3/6 : Bit 6, Word 3, Element 2, File kiểu string số 10.

File dữ liệu M0 và M1 là File dữ liệu dành cho các module vào/ra đặc biệt. Định dạng địa chỉ của File này như sau:

<Tên File><Số File>:<e>. <w>/



File dữ liệu kiểu G (*ConfiGuration*) dành cho các module vào/ra đặc biệt để cấu hình hệ thống.

Chương 4

THIẾT KẾ HỆ ĐIỀU KHIỂN DÙNG PLC

4.1. TRÌNH TỰ CÁC BƯỚC THIẾT KẾ

Thiết kế hệ điều khiển logic dùng PLC bao gồm hai công việc là chọn thiết bị và xây dựng phần mềm. Việc chọn thiết bị dựa trên yêu cầu thực tế đặt ra trên cơ sở đã trình bày ở chương 1 và chương 2. Xây dựng phần mềm điều khiển được tiến hành chủ yếu thông qua việc lập chương trình cho PLC.

Khi tiến hành thiết kế hệ điều khiển logic dùng PLC cần phải tuân theo trình tự các bước sau đây. Đối với người bắt đầu thiết kế nên thực hiện đầy đủ các bước. Đối với người đã làm quen với công việc thiết kế thì có thể bỏ qua một số bước.

4.1.1. Bước 1: Mô tả bài toán

Mô tả bài toán có thể ở dạng một hàm, biểu thức logic, có thể ở dạng đồ thị thời gian hoặc trong trường hợp tổng quát là một đoạn văn bản. Cách mô tả bằng đoạn văn bản là phổ biến nhất vì nó tương ứng với yêu cầu của người sử dụng. Người thiết kế phải chuyển được từ ngôn ngữ văn bản thành ngôn ngữ điều khiển (hàm logic, bảng, đồ thị thời gian). Yêu cầu cơ bản của bước này là phải mô tả một cách đầy đủ và chính xác các yêu cầu của bài toán.

4.1.2. Bước 2: Thống kê thiết bị vào ra

Trên cơ sở các yêu cầu đã có ở bước 1, người thiết kế phải thống kê đầy đủ về chủng loại và kiểu của các thiết bị vào/ra. Các thiết bị vào/ra thông thường là thiết bị tiêu chuẩn được mô tả trong các tài liệu kỹ thuật.

Ví dụ: Thiết bị vào/ra kiểu một chiều, điện áp 24VDC, 48VDC.

Thiết bị vào kiểu xoay chiều, điện áp 110VAC,...

Thiết bị ra một chiều, xoay chiều.

4.1.3. Bước 3: Chọn thiết bị

Công việc chọn thiết bị gồm các phần việc sau:

– Chọn họ PLC sẽ được sử dụng khi thiết kế. Việc chọn họ PLC xuất phát từ yêu cầu kỹ thuật (khả năng thích ứng với công nghệ, tính tác động nhanh, phần mềm,...), yêu cầu kinh tế (giá thành, thời gian đáp ứng) và khả năng hỗ trợ về giải pháp

của nhà cung cấp thiết bị. Các yêu cầu trên được phân tích và cân nhắc cụ thể đối với mỗi bài toán thực tiễn.

– Chọn các thành phần cấu trúc của hệ bao gồm các thiết bị: bảng mạch Bus, nguồn, CPU, các module vào/ra và các thiết bị phụ trợ. Trong phần này cần lưu ý các vấn đề sau: chọn CPU phải đảm bảo về tốc độ xử lý, đáp ứng yêu cầu công nghệ, dung lượng bộ nhớ chương trình và dữ liệu.

Các module vào/ra phải có phải có tổng số đầu vào/ra và kiểu phù hợp với số đầu vào/ra đã được thống kê ở bước 2. Bảng mạch Bus được chọn có số khe cắm (Slot) đủ và module nguồn có công suất thỏa mãn. Các thiết bị phụ trợ như: đầu nối (Connector), cáp truyền thông, các bộ chuyển đổi (Adapter)...

4.1.4. Bước 4: Thiết lập cấu hình của hệ điều khiển logic

Cấu hình của hệ được thiết lập bằng cách lắp ghép các module thành phần trên bảng mạch Bus. Ngoài các module nguồn và module CPU có vị trí xác định, các module còn lại có thể lắp đặt ở vị trí bất kỳ trên bảng mạch Bus sao cho thuận lợi nối ghép với các thiết bị vào/ra.

4.1.5. Bước 5: Địa chỉ hóa vào/ra

Trên cơ sở bước 2, 3 và 4, địa chỉ của các đầu vào/ra được xác định bằng cách địa chỉ hóa. Kết quả của công việc này là thiết lập được hai bảng địa chỉ vào/ra có dạng như sau:

Tên thiết bị vào	Địa chỉ	Tên thiết bị ra	Địa chỉ
Start (Nút khởi động)	000.00	M (Động cơ)	100.00
Stop (Nút dừng)	000.01	Đ (Đèn)	100.01
***	***	***	***

4.1.6. Bước 6: Xây dựng phần mềm

Phần mềm cho hệ điều khiển logic dùng PLC gọi là một thiết kế (Project hoặc Processor File) được lưu giữ dưới một tên duy nhất. Nó bao gồm hai File: File chương trình (Program File) và File dữ liệu (Data File).

File chương trình chứa các lệnh được tạo ra khi soạn thảo chương trình bằng phần mềm lập trình của họ PLC đã chọn.

File dữ liệu chứa dữ liệu thực hiện lệnh và cấu hình hệ thống. File dữ liệu được tạo ra bằng hai cách. Cách thứ nhất, File dữ liệu được tự động tạo ra khi tiến hành xây dựng một thiết kế mới và soạn thảo File chương trình. Cách thứ hai, File dữ liệu được tạo ra bằng cách soạn thảo trực tiếp từ phần mềm lập trình. File dữ liệu được tạo thành theo cách thứ hai phải phù hợp với File chương trình.

Trong đa số trường hợp, khi xây dựng phần mềm, người thiết kế tiến hành soạn thảo File chương trình. Tiếp theo, từ File dữ liệu được hình thành, người thiết kế tiến hành sửa, thêm bớt, ... để tạo thành File dữ liệu cho thiết kế phù hợp với yêu cầu của bài toán. Ví dụ: thiết lập tham số, thiết lập cấu hình riêng của người sử dụng.

Vì vậy, công việc chủ yếu của xây dựng phần mềm là tiến hành xây dựng chương trình.

4.1.7. Bước 7: Chạy mô phỏng

Phần mềm sau khi hoàn thành được nạp vào bộ nhớ của PLC. Thao tác này được thực hiện trên thiết bị lập trình. Để đánh giá chương trình, người thiết kế phải tiến hành chạy thử. Bước chạy thử đầu tiên là chạy mô phỏng, còn gọi là chạy khô (Dry Run) gồm các thủ tục sau:

- Dùng các thiết bị mô phỏng thiết bị vào/ra thực tế và nối thiết bị mô phỏng với các đầu nối trên module vào/ra. Ví dụ các công tắc, chuyển mạch thay thế cho thiết bị vào; đèn, rơ le thay thế cho thiết bị ra.

- Chạy từng bước của chương trình theo cách mô tả bài toán công nghệ ở bước 1.
- Dùng các công cụ phần mềm trên thiết bị lập trình để quan sát, đánh giá.
- Sửa chương trình, kết thúc.

Kết quả của việc chạy mô phỏng phải đạt được như sau:

- Chương trình không có lỗi cú pháp.
- Các bước thực hiện tuần tự đúng với yêu cầu công nghệ.

4.1.8. Bước 8: Chạy công nghệ

Việc hoàn thành bước 7 là điều kiện để thực hiện chạy công nghệ, còn gọi là chạy ướt (*Wet Run*). Chạy công nghệ bao gồm các bước sau:

- Lắp đặt các thiết bị của hệ điều khiển logic PLC vào dây chuyền công nghệ.
- Kết nối các thiết bị vào/ra với module vào/ra.
- Nạp phần mềm vào PLC và tiến hành chạy thử nghiệm toàn bộ hệ thống theo yêu cầu công nghệ đã mô tả ở bước 1.
- Dùng các công cụ phần mềm trên thiết bị lập trình để quan sát, đánh giá.
- Thêm vào chương trình các khâu thời gian trễ, thay đổi dữ liệu trong File dữ liệu để phối hợp hoạt động một cách đồng bộ các khâu, các cơ cấu chấp hành của dây chuyền.

Kết quả của việc chạy công nghệ phải đạt được như sau:

- Toàn bộ hệ thống chạy đúng theo yêu cầu công nghệ.
- Hệ thống điều khiển chạy ổn định.

4.1.9. Bước 9: Lưu giữ phần mềm

Phần mềm sau khi chạy công nghệ là phần mềm gốc, cần phải được cất giữ để phục vụ cho việc bảo trì, bảo dưỡng, thay thế thiết bị sau này. Cần nhấn mạnh ở đây là phải cất giữ cả File chương trình và File dữ liệu. Có thể sử dụng các phương pháp cất giữ như sau: cất giữ trong các thẻ nhớ cố định (EPROM, EEPROM), cất giữ bằng File chương trình, File dữ liệu, cất giữ bằng tài liệu in từ File chương trình, File dữ liệu.

Vấn đề bản quyền phần mềm cần được lưu ý. Để bảo vệ phần mềm, có thể sử dụng công cụ lập trình theo các bước sau:

- Bảo vệ không cho phép mở File chương trình.
- Bảo vệ không cho phép truy nhập vào bộ nhớ chương trình của PLC.

4.1.10. Bước 10: Xây dựng tài liệu

Một sản phẩm chỉ được hoàn chỉnh khi có đủ tài liệu đi kèm. Tài liệu gồm hai loại sau:

- Tài liệu hướng dẫn lắp đặt, vận hành hệ thống dành cho người sử dụng.
- Tài liệu kỹ thuật dành cho cán bộ kỹ thuật bảo trì, bảo dưỡng, thay thế thiết bị gồm có: sơ đồ lắp ráp, cấu hình hệ thống, bảng địa chỉ hóa vào/ra, bảng liệt kê cò, biến trung gian (nếu có), File chương trình, File dữ liệu.

4.2. LẬP TRÌNH CHO HỆ ĐIỀU KHIỂN LOGIC DÙNG PLC

4.2.1. Chương trình và ngôn ngữ lập trình

Chương trình là dãy các lệnh liên tiếp mà PLC phải thực hiện. Chương trình được lưu giữ trong bộ nhớ chương trình ở dạng mã máy. Chương trình do người sử dụng soạn thảo bằng ngôn ngữ lập trình do nhà sản xuất cung cấp. Ngôn ngữ lập trình là môi trường giao tiếp giữa người và máy. Mỗi họ PLC có một hệ lệnh và phần mềm lập trình riêng. Vì vậy, sử dụng họ PLC nào thì phải trang bị phần mềm phù hợp. Ví dụ, PLC họ S7-200 của SIEMENS phải có phần mềm STEP7-200, PLC họ S7-300 của SIEMENS phải có phần mềm STEP7-300, PLC họ SYSMAC của OMRON phải có phần mềm SYSWIN, PLC họ SLC500 của Allen-Bradley phải có phần mềm RSLogix500...

Phần mềm lập trình cho PLC do nhà sản xuất cung cấp. Ngoài chức năng cơ bản là công cụ soạn thảo chương trình cho PLC, phần mềm lập trình còn bao gồm nhiều công cụ tiện ích để người sử dụng giao tiếp, điều khiển PLC và thực hiện các chức năng khác. Ví dụ kết nối với PLC, thiết lập cấu hình hệ thống, nạp chương trình từ thiết bị lập trình vào PLC (*Download*) hoặc đọc chương trình từ PLC (*Upload*), quản lý bộ nhớ, thực hiện các chức năng Force, Force Set, Force Reset, Test, chạy thử chương trình, thay đổi các biểu diễn chương trình, làm tài liệu (*Document*)...

Phần mềm lập trình được cài đặt trong thiết bị lập trình chuyên dụng hoặc trong máy tính PC. Các phần mềm lập trình cho PLC thông thường chạy trên nền hệ điều hành thông dụng như WINDOWS, NT...

4.2.2. Chế độ lập trình

Lập chương trình cho PLC bằng thiết bị lập trình và có thể thực hiện ở các chế độ sau:

– Chế độ OFFLINE: Việc soạn thảo chương trình hoàn toàn tiến hành trên thiết bị lập trình, không kết nối với PLC. Sau khi kết thúc, chương trình được lưu giữ ở một File (*Project File*). Việc nạp chương trình vào PLC được thực hiện ở thời điểm bất kỳ. Khi đó, PLC được kết nối với thiết bị lập trình, File chương trình được mở và thực hiện nạp chương trình vào PLC. Chế độ lập trình OFFLINE chỉ được thực hiện trên các thiết bị lập trình chuyên dụng kiểu máy tính Laptop hoặc máy tính PC có cài đặt phần mềm lập trình.

– Chế độ ONLINE: Đây là chế độ nạp từng lệnh trực tiếp vào bộ nhớ chương trình. Khi đó PLC được kết nối với thiết bị lập trình. Mỗi lệnh soạn thảo trên thiết bị lập trình sẽ được ghi trực tiếp vào bộ nhớ chương trình của PLC. Sử dụng phương pháp này trong hai trường hợp sau:

+ Soạn thảo trực tiếp một chương trình mới trong PLC.

+ Thay đổi một số lệnh của chương trình đã có sẵn trong PLC khi chương trình đang chạy.

Đây là chế độ khá đặc biệt giúp cho người sử dụng dễ dàng thay đổi (xóa, thêm, bớt) các lệnh của chương trình một cách nhanh chóng mà không cần phải lập lại các thao tác nạp chương trình mới. Chế độ này gọi là ONLINE EDIT.

4.2.3. Các phương pháp biểu diễn chương trình

Chương trình cho PLC có thể được biểu diễn bằng một trong ba cách phổ biến sau: phương pháp biểu diễn bằng dãy lệnh (*STL – Statement List*), phương pháp biểu diễn bằng giản đồ thang (*LAD – Ladder*) và phương pháp biểu diễn bằng sơ đồ khối chức năng điều khiển (*CSF – Control System Flow Chart*).

Trong phương pháp biểu diễn bằng dãy lệnh, mỗi lệnh được biểu diễn bằng một cụm từ viết tắt mô tả lệnh tương tự như hợp ngữ. Các lệnh gồm có lệnh vào (*Input Instruction*) và lệnh ra (*Output Instruction*). Ví dụ, lệnh LD (*Load*), lệnh NOT, lệnh AND, lệnh OR là các lệnh vào, lệnh OUT là lệnh ra. Ngoài ra còn có các lệnh khác như lệnh FLIP-FLOP, lệnh TIMER, lệnh COUNTER, lệnh số học (ADD, SUB), lệnh chuyển dữ liệu (MOVE),...

Cấu trúc của câu lệnh có dạng sau:

<Tên lệnh>	Toán hạng 1, toán hạng 2
(<Operation>	Operand 1, Operand 2)

Tên lệnh (*Operation*) là cụm từ viết tắt chỉ ra chức năng của lệnh, gọi là mã nhớ (*Mnemonic Code*).

Toán hạng (*Operand*) là địa chỉ của dữ liệu thực hiện lệnh. Tùy thuộc vào kiểu lệnh, dữ liệu có thể ở dạng Bit, Byte, Word, Double Word,... Phương pháp địa chỉ hóa dữ liệu gọi là các mode địa chỉ hóa. Các mode địa chỉ hóa thông dụng là: mode địa chỉ hóa trực tiếp (*Direct Addressing*), mode địa chỉ hóa gián tiếp (*Indirect Addressing*), mode địa chỉ hóa tức thời (*Immediated Addressing*)...

Trong mode địa chỉ hóa trực tiếp, toán hạng là địa chỉ của dữ liệu.

Trong mode địa chỉ hóa gián tiếp, toán hạng là địa chỉ chứa địa chỉ của dữ liệu.

Trong mode địa chỉ hóa tức thời, toán hạng là dữ liệu.

Một câu lệnh có thể chỉ có tên lệnh (đối với câu lệnh không có toán hạng). Ví dụ: lệnh CLR(41): xóa cờ Carry. Một câu lệnh cũng có thể chứa một hoặc nhiều toán hạng. Các toán hạng cách nhau bởi dấu phẩy. Ví dụ: lệnh LD 000.00; lệnh ADD(30), CP1, CP2, SUM.

Biểu diễn lệnh ở dạng STL có các ưu điểm sau:

- Dễ dàng soạn thảo chương trình và có thể biểu diễn tất cả các lệnh ở dạng STL.

- File chương trình là File văn bản (*Text*), nên kích thước nhỏ, vì vậy thích hợp để biểu diễn các chương trình cỡ lớn.

- Thuận lợi cho người lập trình chuyên nghiệp và ngôn ngữ thể hiện tương tự như hợp ngữ.

Nhược điểm của phương pháp biểu diễn lệnh ở dạng STL là:

- Việc thiết kế mạch điều khiển dùng PLC trực tiếp từ cách biểu diễn bằng STL gặp rất nhiều khó khăn, thậm chí không thể được. Bởi vì, người thiết kế hệ điều khiển PLC là các kỹ sư điện, điện tử nên kỹ năng lập trình chuyên nghiệp không cao và thực tế họ thiết kế mạch điều khiển ở dạng giản đồ thang hoặc sơ đồ các khối điều khiển chức năng.

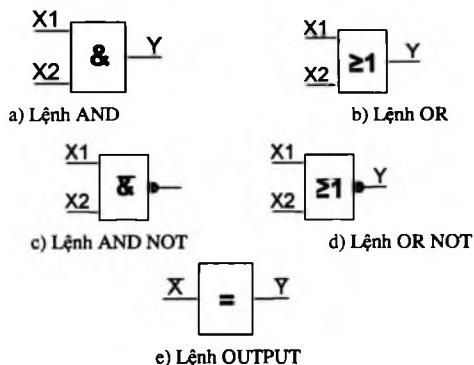
- Biểu diễn dưới dạng STL không trực quan. Đặc biệt khi thực hiện việc kiểm soát, theo dõi hoạt động của chương trình.

Vì vậy, biểu diễn chương trình cho PLC ở dạng STL ít được sử dụng trong khi thiết kế.

Trong phương pháp biểu diễn bằng giản đồ thang, các lệnh được biểu diễn ở dạng biểu tượng đặc trưng (*Symbol*). Chương trình được biểu diễn ở dạng đồ họa (*Graphic*). Đây là cách biểu diễn thông dụng nhất và trở thành cách biểu diễn chuẩn của chương trình cho PLC.

Trong các biểu diễn bằng sơ đồ khối chức năng điều khiển (CSF), các lệnh được biểu diễn bằng các khối chức năng logic, như sơ đồ các phần tử logic cơ bản. Ví dụ,

các lệnh AND, OR, AND NOT, OR NOT và OUTPUT được biểu diễn trên hình 4.1 như sau:



Hình 4.1. Biểu diễn lệnh dạng CSF.

Cách biểu diễn lệnh CSF thích hợp đối với nhà thiết kế chuyên ngành điện tử. Trong thực tế, cách biểu diễn CSF được sử dụng không nhiều. Người sử dụng có thể tùy chọn một trong ba cách biểu diễn chương trình như ở trên. Việc chuyển đổi cách biểu diễn có thể thực hiện được nhờ công cụ trong phần mềm lập trình.

4.3. GIẢN ĐỒ HÌNH THANG VÀ PHƯƠNG PHÁP BIỂU DIỄN CHƯƠNG TRÌNH BẰNG GIẢN ĐỒ THANG (LAD)

4.3.1. Giản đồ hình thang

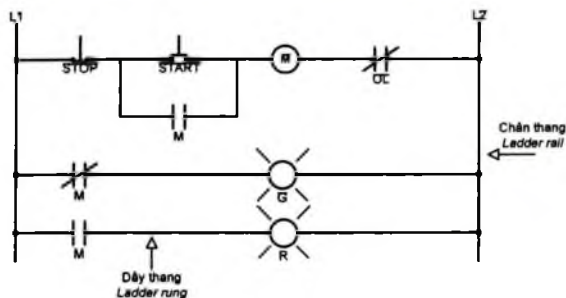
Trong các mạch điều khiển logic, các phần tử logic kiểu rơ le được nối với nhau theo sơ đồ thiết kế để thực hiện hàm điều khiển xác định – gọi là mạch logic nối dây cứng (Hard – wired Logic). Biểu diễn mạch logic nối dây cứng nhờ giản đồ thang với các phần tử rơ le.

Ví dụ trên hình 4.2 trình bày giản đồ thang của mạch điều khiển logic nối dây cứng.

Đây là giản đồ thang biểu diễn mạch điều khiển START/STOP. Thiết bị M (Motor) với các đèn báo dừng G và báo chạy R. Start, Stop là các tiếp điểm thường mở (NO – Normally Opened Contact) và thường kín (NC – Normally Closed Contact), OL là tiếp điểm bảo vệ quá nhiệt (Over Load), các tiếp điểm M là tiếp điểm phụ của thiết bị ... L1 và L2 là hai dây nguồn. Như vậy, tất cả các phần tử đều được bố trí giữa hai

dây nguồn L1, L2. Mạch điện được khép kín khi các tiếp điểm trên dây thang (Ladder Rung) ở trạng thái ON. Hai dây dây nguồn L1, L2 gọi là chân thang (Rail).

Ưu điểm cơ bản của phương pháp biểu diễn giản đồ thang là tính trực quan và đơn giản. Đây là phương pháp biểu diễn cơ bản trong các sơ đồ điều khiển logic.



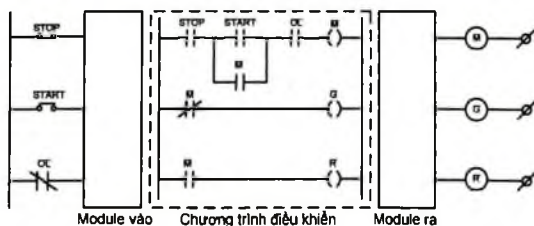
Hình 4.2. Giản đồ hình thang rơ le.

4.3.2. Phương pháp biểu diễn chương trình điều khiển bằng giản đồ hình thang

Về hình thức, chương trình biểu diễn ở dạng giản đồ hình thang tương tự như sơ đồ mạch điều khiển logic hình thang.

Trong đó, các phần tử logic (rơ le) được biểu diễn bằng các biểu tượng (Symbol). Ví dụ, các tiếp điểm thường hở (NO) được biểu diễn bằng biểu tượng $-|-$, các tiếp điểm thường kín (NC) được biểu diễn bằng biểu tượng $-|/$, các phần tử mô tả cuộn dây rơ le được biểu diễn bằng biểu tượng $-()$, một số các phần tử, các hàm logic cơ bản được biểu diễn ở dạng khối chức năng (Function Block).

Sơ đồ và chương trình dạng LAD của mạch điều khiển logic hình 4.2 ở trên được trình bày trên hình 4.3.



Hình 4.3. Chương trình điều khiển dạng LAD.

Nội dung của phương pháp biểu diễn chương trình bằng giản đồ thang như sau:

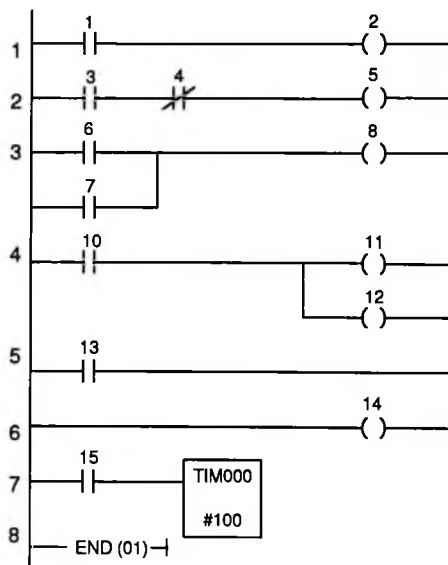
- Hai đường thẳng đứng tương ứng cho hai dây nguồn.
- Nối hai dây nguồn là các đường logic (*Logic Line*) tương ứng cho dây thang (*Rung*).

- Trên các đường logic bố trí các phần tử logic vào (*Input*) đặt ở bên trái, các phần tử logic ra (*Output*) đặt ở bên phải.

- Mạch logic (*Logic Network*) bao gồm một hoặc nhiều đường logic. Trong đó, các phần tử vào có thể mắc nối tiếp, song song, có thể có một hoặc nhiều phần tử ra. Phần tử ra có thể ở dạng khối chức năng. Mạch logic biểu diễn đầy đủ một biểu thức logic.

- Chương trình thực hiện lần lượt từ trái sang phải và từ trên xuống dưới.

Trong ví dụ hình 4.4, các chữ số phía trên các biểu trưng là toán hạng (địa chỉ Bit).



Hình 4.4. Ví dụ chương trình LAD.

Mạch logic 1 gồm một đường logic, thực hiện biểu thức logic: $(2) = (1)$.

Mạch logic 2 gồm một đường logic, thực hiện biểu thức logic: $(5) = (3) \cdot (\overline{4})$.

Mạch logic 3 gồm hai đường logic, thực hiện biểu thức logic: $(8) = (6) + (7)$.

Mạch logic 4 gồm hai đường logic, thực hiện biểu thức logic: $\begin{cases} (11) = (10) \\ (12) = (10) \end{cases}$

Mạch logic 5 gồm một đường logic, chỉ có một phần tử vào. *Vì vậy sai về cú pháp.*

Mạch logic 6 gồm một đường logic, chỉ có một phần tử ra. *Vì vậy sai về cú pháp.*

Mạch logic 7 gồm một đường logic, trong đó phần tử ra là 1 TIMER (dạng khối chức năng).

Mạch logic 8 chỉ gồm một lệnh kết thúc chương trình.

Cách biểu diễn chương trình dạng LAD cho phép người sử dụng có thể chuyển đổi dễ dàng từ sơ đồ mạch điều khiển logic cứng sang dạng chương trình LAD. Tuy nhiên, khi chuyển đổi cần phải lưu ý việc chuyển đổi phải tương đương về chức năng.

Soạn thảo chương trình LAD được tiến hành bằng phần mềm lập trình và các công cụ lập trình do nhà sản xuất cung cấp.

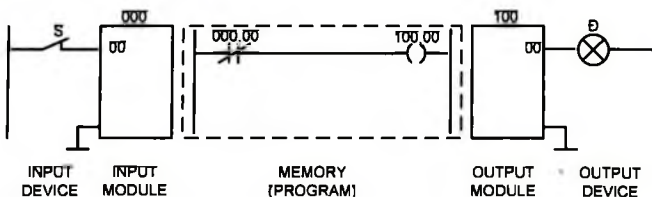
4.4. MỘT SỐ KHÁI NIỆM

4.4.1. Logic tiếp điểm nhị phân

Trong cách biểu diễn chương trình LAD, các lệnh vào thường là lệnh “tiếp điểm” NO hoặc NC. Trạng thái của nó có thể là ON hoặc OFF phụ thuộc vào giá trị của Bit dữ liệu của toán hạng. Đối với lệnh “tiếp điểm” NO, trạng thái của nó là OFF nếu Bit dữ liệu bằng 0. Khi Bit dữ liệu bằng 1 thì trạng thái là ON, là trạng thái tích cực (Active). Ngược lại đối với lệnh “tiếp điểm” NC, trạng thái của nó là ON nếu Bit dữ liệu bằng 0. Khi Bit dữ liệu bằng 1 thì trạng thái là OFF, là trạng thái tích cực.

4.2. Quan hệ trạng thái của thiết bị vào với Bit dữ liệu

Các thiết bị vào được nối vào các tiếp điểm trên module vào của PLC. Hình 4.5a dưới đây mô tả quan hệ logic và kết quả logic từ các trạng thái của các thiết bị vào và các lệnh LAD.



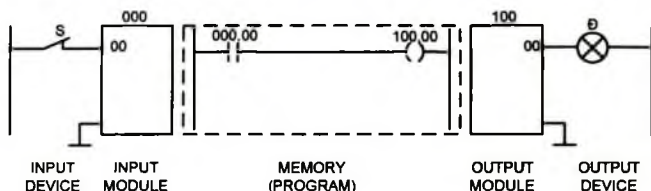
Hình 4.5a. Ví dụ 1 về quan hệ trạng thái của thiết bị vào với Bit dữ liệu.

Trường hợp hình 4.5a: tiếp điểm vào S là tiếp điểm NC, nối vào điểm 000.00 trên module vào. Bit dữ liệu 000.00 trong vùng ánh đầu vào có giá trị 1 khi thiết bị S ở

trạng thái ON (*Inactive*) và có giá trị 0 khi thiết bị vào S ở trạng thái OFF (*Active*). Chương trình trong bộ nhớ thực hiện biểu thức logic:

$$(100.00) = \overline{(000.00)}$$

Nghĩa là thiết bị ra Đ chỉ ON khi thiết bị vào S ở trạng thái OFF (*Active*). Đường nối giữa nguồn và đầu vào PLC bị ngắt.



Hình 4.5b. Ví dụ 2 về quan hệ trạng thái của thiết bị vào với Bit dữ liệu.

Trường hợp hình 4.5b, thiết bị vào S giống như trường hợp hình 4.5a. Tuy nhiên, chương trình trong bộ nhớ thực hiện biểu thức logic:

$$(100.00) = (000.00)$$

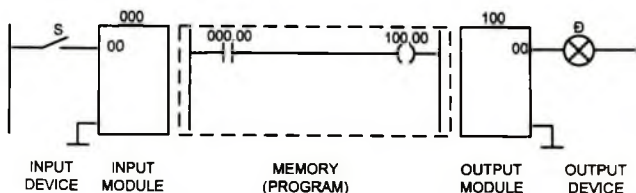
Nghĩa là, thiết bị ra Đ ở trạng thái ON khi thiết bị vào ở trạng thái ON (*Inactive*), đường nối giữa nguồn và đầu vào PLC thông mạch.

Trường hợp hình 4.5c: thiết bị vào S là tiếp điểm NO, nối vào điểm 000.00 trên module vào. Bit dữ liệu 00.00 trong vùng ảnh đầu vào có giá trị 1 khi thiết bị vào S ở trạng thái ON (*Active*) và có giá trị 0 khi thiết bị vào S ở trạng thái OFF (*Inactive*).

Chương trình trong bộ nhớ thực hiện biểu thức logic:

$$(100.00) = (000.00)$$

Nghĩa là, thiết bị ra Đ ở trạng thái ON khi thiết bị vào S ở trạng thái ON (*Active*), đường nối giữa nguồn và đầu vào PLC thông mạch.

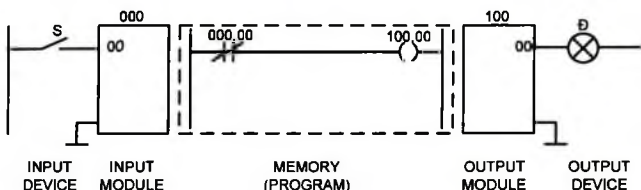


Hình 4.5c. Ví dụ 3 về quan hệ trạng thái của thiết bị vào với Bit dữ liệu.

Trường hợp hình 4.5d: thiết bị vào S giống như trường hợp hình 4.5a. Tuy nhiên, chương trình trong bộ nhớ thực hiện biểu thức logic:

$$(100.00) = \overline{(000.00)}$$

Nghĩa là, thiết bị ra Đ ở trạng thái ON khi thiết bị vào S ở trạng thái OFF (*Inactive*), đường nối giữa nguồn và đầu vào PLC hở mạch.



Hình 4.5d. Ví dụ 4 về quan hệ trạng thái của thiết bị vào với Bit dữ liệu.

4.4.3. Trạng thái thiết bị đầu ra

Các thiết bị ra được nối với các điểm trên module đầu ra. Tùy thuộc vào tín hiệu (có hoặc không) mà thiết bị ra ở trạng thái tích cực (*Active*) hay không tích cực (*Inactive*). Tính tích cực (*Active/Inactive*) của thiết bị ra phụ thuộc vào kiểu thiết bị và quy ước của người sử dụng. Ví dụ, thiết bị ra là đèn báo, tính *Active/Inactive* có thể là trạng thái sáng/tối hoặc ngược lại. Đặc biệt, khi thiết bị ra là các van khí nén hoặc thủy lực điều khiển các cơ cấu hành trình dạng piston thì vị trí ban đầu (*Normal*) của cơ cấu tùy thuộc vào kiểu van. Việc chọn kiểu van phải được thực hiện cẩn thận để tránh xảy ra nguy hiểm khi hệ thống mất nguồn khí hoặc thủy lực.

4.4.4. Các Bit toán hạng

Cấu trúc của lệnh gồm hai phần: tên lệnh (*Operation*) và toán hạng (*Operand*).

Toán hạng là địa chỉ của dữ liệu. Đối với các lệnh kiểu tiếp điểm, kiểu dữ liệu là Bit. Vì vậy, toán hạng là địa chỉ Bit. Toán hạng Bit này có thể được sử dụng nhiều lần, trong tất cả các lệnh của hệ lệnh PLC có dữ liệu kiểu Bit.

Các vùng dữ liệu có thể truy nhập ở dạng Bit đều có thể dùng làm toán hạng Bit trong các câu lệnh. Các vùng dữ liệu Bit của PLC họ S7-200 (SIEMENS) là V, I, Q, M, SM, T, C. Các vùng dữ liệu Bit của PLC họ SYMAC (OMRON) là IR, HR, LR, AR, T, C, ... Các vùng dữ liệu Bit của PLC họ Controllogix (AB) là I, O, B, N, T, C, ...

4.5. BIỂU DIỄN CHƯƠNG TRÌNH BẰNG DÂY LỆNH (STL)

Chương trình có thể được soạn thảo ở dạng STL (*STatement List*). Khi đó, File chương trình là dạng TEXT. Thực chất, soạn thảo chương trình dạng STL là

chuyển đổi chương trình từ dạng LAD sang dạng STL. Việc chuyển đổi này có thể được thực hiện bằng tay hoặc tự động nhờ công cụ trong phần mềm lập trình.

4.5.1. Các lệnh cơ bản dạng STL

- Lệnh LD (*Load*): Lệnh nạp một phần tử mới.
Bắt đầu một mạch logic mới hoặc một khối mới.
- Lệnh NOT: Lệnh phủ định.
- Lệnh AND: Lệnh mắc nối tiếp hai phần tử.
- Lệnh OR: Lệnh mắc song song hai phần tử.
- Lệnh OUT: Lệnh đầu ra.

Ví dụ: Chuyển chương trình dạng LAD hình 4.4 sang dạng STL từ NETWORK

1 đến NETWORK 4:

Network 1:	LD	1
	OUT	2
Network 2:	LD	3
	AND NOT	4
	OUT	5
Network 3:	LD	6
	OR	7
	OUT	8
Network 4:	LD	10
	OUT	11
	OUT	12

4.5.2. Các lệnh khối: ANDLD và ORLD

Trong trường hợp, mạch logic bao gồm tổ hợp mắc song song và nối tiếp các phần tử thì phải sử dụng các lệnh khối ANDLD và ORLD.

Lệnh ANDLD là lệnh mắc nối tiếp hai khối.

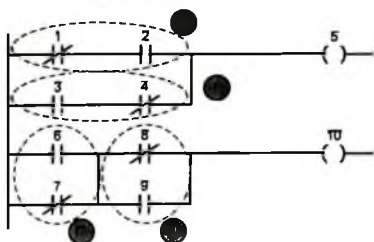
Lệnh ORLD là lệnh mắc song song hai khối.

Khối (*Block*) là một đoạn mạch logic chỉ gồm các phần tử mắc nối tiếp hoặc song song. Như vậy, một phần tử có thể coi là một khối. Việc soạn thảo chương trình dạng STL cho một khối tương tự như soạn thảo chương trình cho một Logic NETWORK mới. Các khối được soạn thảo độc lập. Các khối được mắc nối tiếp bằng lệnh ANDLD và được mắc song song bằng lệnh ORLD.

Thực hiện theo thứ tự từ trái sang phải, từ trên xuống dưới.

Ví dụ: Chuyển chương trình dạng LAD sau đây ra dạng STL.

Giản đồ thang:



Chương trình dạng STL:

Network 1:	LD NOT	1
	AND	2
	LD	3
	AND NOT	4
	ORLD	
	OUT	5
Network 2:	LD	6
	OR NOT	7
	LD NOT	8
	OR	9
	ANDLD	
	OUT	10

Network 1 gồm 2 khối mắc song song: I và II.

Network 2 gồm 2 khối mắc nối tiếp: III và IV.

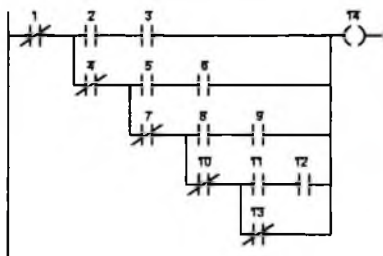
Ví dụ: Chuyển chương trình dạng LAD sau đây sang dạng STL.

Chương trình dạng STL:

```

LD      NOT 1
LD      2
AND     3
LD      NOT 4
LD      5
AND     6
LD      NOT 7
LD      8
AND     9
LD      NOT 10
LD      11
AND     12
LD      NOT 13
ORLD
ANDLD
ORLD
ANDLD
ORLD
ANDLD
ORLD
ANDLD
OUT
    
```

Giản đồ thang:



4.5.3. Các lệnh dạng khối chức năng (Function Block)

Khi soạn thảo chương trình dạng STL, các lệnh gọi trực tiếp như đã trình bày ở trên. Tuy nhiên với các lệnh phức tạp có nhiều toán hạng thì việc gọi trực tiếp gặp nhiều khó khăn. Vì vậy, đa số các lệnh còn lại được gọi gián tiếp thông qua mã hàm (Function Code). Ví dụ, PLC họ SYSMAC của OMRON có các mã hàm như lệnh END (01), CLC (41), MOVE(21), SFT(10)...

Trong đó, số nằm trong ngoặc đơn là mã hàm. Một số lệnh có nhiều toán hạng, hoặc cần phải khai báo phức tạp hơn, được biểu diễn ở dạng khối chức năng (Function Block). Ví dụ: các lệnh MOV(21), lệnh TIMER, lệnh COUNTER...[4].

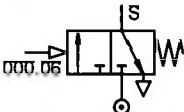
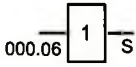
4.6. LẬP TRÌNH CÁC HÀM LOGIC CƠ BẢN TRÊN PLC

Để có thể lập trình cho PLC một cách có hiệu quả, cần nắm được các khái niệm logic cơ bản được sử dụng trong khi lập trình điều khiển.

Các khái niệm này bao gồm:

Hàm logic cơ bản	Hàm logic phức hợp
Hàm "YES" (Đồng nhất)	Hàm "EXCLUSIVE OR" (Cộng module 2)
Hàm "NOT" (Phủ định)	Hàm "NAND" (Và đảo)
Hàm "AND" (Và)	Hàm "NOR" (Hoặc đảo)
Hàm "OR" (Hoặc)	Hàm "AND INHIBIT" (Cấm và)
Hàm "INHIBITION" (Cấm)	Hàm "OR INHIBIT" (Cấm hoặc)

4.6.1. Hàm YES

	<pre>LD 000.06 OUT 100.04</pre>
Chương trình dạng LAD	Chương trình dạng STL
	
Phần tử logic khí nén	Phần tử khối chức năng
Biểu thức logic: $(100.04) = (000.06)$	

4.6.2. Hàm NOT

	LD NOT 000.13 OUT 100.11
Chương trình dạng LAD	Chương trình dạng STL
Phần tử logic khí nén	Phần tử khối chức năng
Biểu thức logic: $(100.11) = (\overline{000.13})$	

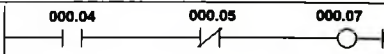
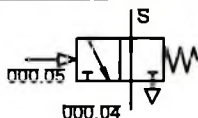
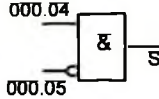
4.6.3. Hàm AND

	LD 000.07 AND 000.15 OUT 100.03
Chương trình dạng LAD	Chương trình dạng STL
Phần tử logic khí nén	Phần tử khối chức năng
Biểu thức logic: $(100.03) = (000.07).(000.15)$	

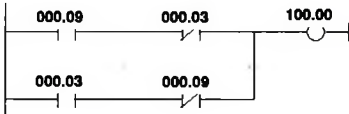
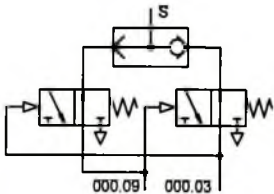
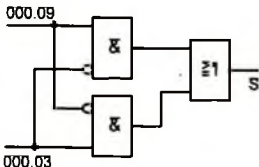
4.6.4. Hàm OR

	LD 000.00 OR 000.08 OUT 100.01
Chương trình dạng LAD	Chương trình dạng STL
Phần tử logic khí nén	Phần tử khối chức năng
Biểu thức logic: $(100.01) = (000.00) + (000.08)$	

4.6.5. Hàm INHIBITION

	LD 000.04 AND NOT 000.05 OUT 100.07
Chương trình dạng LAD	Chương trình dạng STL
	
Phản tử logic khí nén	Phản tử khối chức năng
Biểu thức logic: $(100.07) = (000.04).(000.05)$	

4.6.6. Hàm EXCLUSIVE OR

	LD 000.09 AND NOT 000.03 LD 000.03 AND NOT 000.09 OR LD OUT 100.00
Chương trình dạng LAD	Chương trình dạng STL
	
Phản tử logic khí nén	Phản tử khối chức năng
Biểu thức logic: $(100.00) = (000.09).(000.03) + (000.03).(000.09)$	

4.6.7. Hàm NAND

	<table> <tr><td>LD</td><td>000.07</td></tr> <tr><td>AND</td><td>000.12</td></tr> <tr><td>OUT</td><td>200.00</td></tr> <tr><td>LD NOT</td><td>200.00</td></tr> <tr><td>OUT</td><td>100.09</td></tr> </table>	LD	000.07	AND	000.12	OUT	200.00	LD NOT	200.00	OUT	100.09
LD	000.07										
AND	000.12										
OUT	200.00										
LD NOT	200.00										
OUT	100.09										
Chương trình dạng LAD	Chương trình dạng STL										
Phần tử logic khí nén	Phần tử khối chức năng										
Biểu thức logic: $(100.09) = \overline{(000.07) \cdot (000.12)}$											

4.6.8. Hàm NOR

	<table> <tr><td>LD</td><td>000.13</td></tr> <tr><td>OR</td><td>000.00</td></tr> <tr><td>OUT</td><td>200.03</td></tr> <tr><td>LD NOT</td><td>200.03</td></tr> <tr><td>OUT</td><td>100.08</td></tr> </table>	LD	000.13	OR	000.00	OUT	200.03	LD NOT	200.03	OUT	100.08
LD	000.13										
OR	000.00										
OUT	200.03										
LD NOT	200.03										
OUT	100.08										
Chương trình dạng LAD	Chương trình dạng STL										
Phần tử logic khí nén	Phần tử khối chức năng										
Biểu thức logic: $(100.08) = \overline{(000.00) + (000.13)}$											

4.6.9. Hàm OR INHIBIT

	<pre> LD 000.01 OR 000.02 OR 000.03 AND NOT 000.04 OUT 100.00 </pre>
Chương trình dạng LAD	Chương trình dạng STL
Phần tử logic khí nén	Phần tử khối chức năng
Biểu thức logic: $(100.00) = [(000.01) + (000.02) + (000.03)] \overline{(000.04)}$	

4.6.10. Hàm AND INHIBIT

	<pre> LD 000.01 AND 000.06 AND 000.04 AND NOT 000.03 OUT 100.00 </pre>
Chương trình dạng LAD	Chương trình dạng STL
Phần tử logic khí nén	Phần tử khối chức năng
Biểu thức logic: $(100.00) = [(000.01) + (000.02) + (000.03)] \overline{(000.04)}$	

Bảng 4.1. Bảng tóm tắt các hàm logic căn bản cho PLC

YES	<table><tr><td>A</td><td>S</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	S	0	0	1	1			<table><tr><td>000.01</td><td>100.00</td></tr><tr><td>A</td><td>S</td></tr></table>	000.01	100.00	A	S	$A = S$																	
A	S																															
0	0																															
1	1																															
000.01	100.00																															
A	S																															
NOT	<table><tr><td>A</td><td>S</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	S	0	1	1	0			<table><tr><td>000.01</td><td>100.00</td></tr><tr><td>A</td><td>S</td></tr></table>	000.01	100.00	A	S	$\bar{A} = S$																	
A	S																															
0	1																															
1	0																															
000.01	100.00																															
A	S																															
AND	<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1			<table><tr><td>000.01</td><td>000.02</td><td>100.00</td></tr><tr><td>A</td><td>B</td><td>S</td></tr></table>	000.01	000.02	100.00	A	B	S	$A \cdot B = S$						
A	B	S																														
0	0	0																														
0	1	0																														
1	0	0																														
1	1	1																														
000.01	000.02	100.00																														
A	B	S																														
OR	<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1			<table><tr><td>000.01</td><td>100.00</td></tr><tr><td>000.02</td><td></td></tr><tr><td>A</td><td>B</td></tr></table>	000.01	100.00	000.02		A	B	$A + B = S$						
A	B	S																														
0	0	0																														
0	1	1																														
1	0	1																														
1	1	1																														
000.01	100.00																															
000.02																																
A	B																															
INHIBITION	<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	0	0	1	0	1	0	1	1	1	0			<table><tr><td>000.01</td><td>000.02</td><td>100.00</td></tr><tr><td>A</td><td>B</td><td>S</td></tr></table>	000.01	000.02	100.00	A	B	S	$A \cdot \bar{B} = S$						
A	B	S																														
0	0	0																														
0	1	0																														
1	0	1																														
1	1	0																														
000.01	000.02	100.00																														
A	B	S																														
NAND	<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0			<table><tr><td>000.01</td><td>000.02</td><td>200.00</td></tr><tr><td>200.00</td><td>100.00</td><td></td></tr><tr><td>A</td><td>B</td><td>S</td></tr></table>	000.01	000.02	200.00	200.00	100.00		A	B	S	$\overline{A \cdot B} = S$			
A	B	S																														
0	0	1																														
0	1	1																														
1	0	1																														
1	1	0																														
000.01	000.02	200.00																														
200.00	100.00																															
A	B	S																														
NOR	<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0			<table><tr><td>000.01</td><td>200.00</td></tr><tr><td>000.02</td><td></td></tr><tr><td>200.00</td><td>100.00</td></tr><tr><td>A</td><td>B</td></tr></table>	000.01	200.00	000.02		200.00	100.00	A	B	$\overline{A + B} = S$				
A	B	S																														
0	0	1																														
0	1	0																														
1	0	0																														
1	1	0																														
000.01	200.00																															
000.02																																
200.00	100.00																															
A	B																															
EXCLUSIVE OR	<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0			<table><tr><td>000.01</td><td>000.02</td><td>100.00</td></tr><tr><td>000.01</td><td>000.02</td><td></td></tr><tr><td>A</td><td>B</td><td>S</td></tr></table>	000.01	000.02	100.00	000.01	000.02		A	B	S	$(A \cdot \bar{B}) + (\bar{A} \cdot B) = S$			
A	B	S																														
0	0	0																														
0	1	1																														
1	0	1																														
1	1	0																														
000.01	000.02	100.00																														
000.01	000.02																															
A	B	S																														
MEMORY (INTERNAL)	<table><tr><td>A</td><td>S</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	S	Q	0	0	0	0	1	1	1	0	0	1	1	1			<table><tr><td>000.01</td><td>KEEP(11)</td></tr><tr><td>A</td><td></td></tr><tr><td>000.02</td><td>R001.00</td></tr><tr><td>B</td><td>Q</td></tr></table>	000.01	KEEP(11)	A		000.02	R001.00	B	Q	$A = Q$ $B = \bar{Q}$				
A	S	Q																														
0	0	0																														
0	1	1																														
1	0	0																														
1	1	1																														
000.01	KEEP(11)																															
A																																
000.02	R001.00																															
B	Q																															
TIMER DELAY	<table><tr><td>A</td><td>T</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	T	Q	0	0	0	0	1	1	1	0	0	1	1	1			<table><tr><td>000.01</td><td>TMR</td></tr><tr><td>A</td><td></td></tr><tr><td>003</td><td></td></tr><tr><td>Q</td><td></td></tr><tr><td>00000</td><td></td></tr><tr><td>0000 bcd</td><td></td></tr></table>	000.01	TMR	A		003		Q		00000		0000 bcd		
A	T	Q																														
0	0	0																														
0	1	1																														
1	0	0																														
1	1	1																														
000.01	TMR																															
A																																
003																																
Q																																
00000																																
0000 bcd																																
COUNTER DELAY	<table><tr><td>CP</td><td>CNT</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	CP	CNT	Q	0	0	0	0	1	1	1	0	0	1	1	1			<table><tr><td>000.01</td><td>CNT</td></tr><tr><td>A</td><td></td></tr><tr><td>000.02</td><td>104</td></tr><tr><td>Q</td><td></td></tr><tr><td>0128</td><td></td></tr><tr><td>0128 bcd</td><td></td></tr></table>	000.01	CNT	A		000.02	104	Q		0128		0128 bcd		
CP	CNT	Q																														
0	0	0																														
0	1	1																														
1	0	0																														
1	1	1																														
000.01	CNT																															
A																																
000.02	104																															
Q																																
0128																																
0128 bcd																																

4.7. CÁC BIT ĐẦU RA HAI TRẠNG THÁI ỔN ĐỊNH VÀ MỘT TRẠNG THÁI ỔN ĐỊNH (BISTABLE VÀ MONSTABLE)

Trong PLC, các Bit ra (*Output Bit, Output Relay*) là kết quả thực hiện một hàm logic. Các Bit ra này có thể được sử dụng để điều khiển các thiết bị ngoài nối vào PLC, gọi là rơ le ngoài (*External Output Relay*). Ví dụ các thiết bị ngoài: đèn, van điện khí, cuộn dây bộ khởi động, đóng ngắt thanh đốt... Các Bit ra này có thể được sử dụng như các Bit trung gian gọi là rơ le trong (*Internal Output Relay*). Thậm chí, các Bit ra này có thể sử dụng cho cả hai mục đích trên.

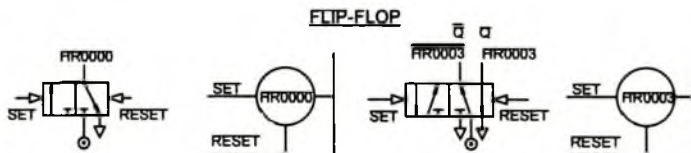
Xét trên quan điểm đặc tính, các Bit ra này (gồm cả Internal và External Relay) có hai loại: các Bit hai trạng thái ổn định và các Bit một trạng thái ổn định.

Các Bit ra hai trạng thái ổn định là một bộ nhớ Bit. Phần tử khối chức năng bộ nhớ 1 Bit là các Flip-Flop (RS, JK, D, T...).)

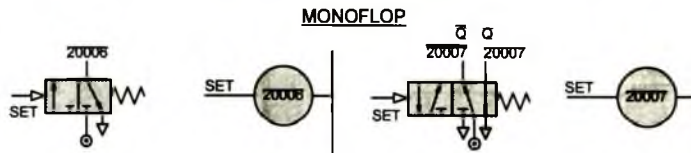
Lệnh Bit ra hai trạng thái ổn định có hai đầu vào SET và RESET. Trạng thái của Bit ra ở ON khi kích thích đầu vào SET và được duy trì cả khi tín hiệu vào SET mất. Trạng thái Bit ra ở OFF khi kích thích đầu vào RESET.

Các Bit ra một trạng thái ổn định chỉ có một đầu vào SET, gọi là Monoflop. Trạng thái của Bit ra chỉ ở ON khi có tín hiệu ở đầu vào SET.

Sơ đồ các Bit ra hai trạng thái ổn định (Flip-Flop) và các Bit ra một trạng thái ổn định (Monoflop) được trình bày trên hình 4.6 và 4.7.



Hình 4.6. Sơ đồ các Bit ra hai trạng thái ổn định (Flip-Flop).



Hình 4.7. Sơ đồ các Bit ra một trạng thái ổn định (Monoflop).

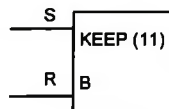
Để xây dựng các Flip-Flop (RS, JK, D, T,...), trong PLC thường sử dụng các lệnh Flip-Flop với các tên gọi khác nhau. Ví dụ, lệnh KEEP(11) trong PLC của OMRON là lệnh duy trì Bit ra hai trạng thái ổn định. Cú pháp lệnh được trình bày trên hình 4.8.

RS Flip-Flop với đầu vào ưu tiên RESET được trình bày trên hình 4.9a. Và RS Flip-Flop với đầu vào ưu tiên SET được trình bày trên hình 4.9b.

T Flip-Flop chỉ có một đầu vào và chỉ thay đổi trạng thái tại thời điểm đầu vào xuất hiện tín hiệu. T Flip-Flop còn được gọi là bộ chia 2 và được trình bày trên hình 4.11.

Bit đầu ra chốt (*Latching*) còn được gọi là Bit tự duy trì và được trình bày trên hình 4.10.

Cú pháp lệnh:



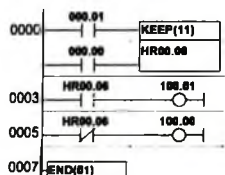
Trong đó:

S là đầu vào SET

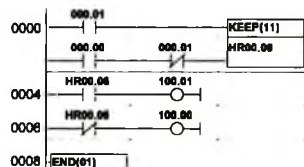
R là đầu vào RESET

B là địa chỉ Bit ra

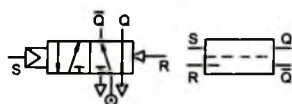
Hình 4.8. Lệnh KEEP(11).



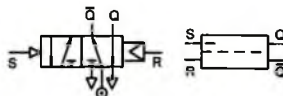
ADDRESS	MNEMONIC	OPERAND
0000	LD	000.01
0001	LD	000.00
0002	KEEP	HR00.06
0003	LD	HR00.06
0004	OUT	100.01
0005	LD NOT	HR00.06
0006	OUT	100.00
0007	END	



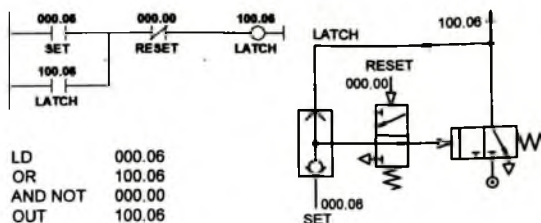
ADDRESS	MNEMONIC	OPERAND
0000	LD	000.01
0001	LD	000.00
0002	AND NOT	000.01
0003	KEEP	HR00.06
0004	LD	HR00.06
0005	OUT	100.01
0006	LD NOT	HR00.06
0007	OUT	100.00
0008	END	



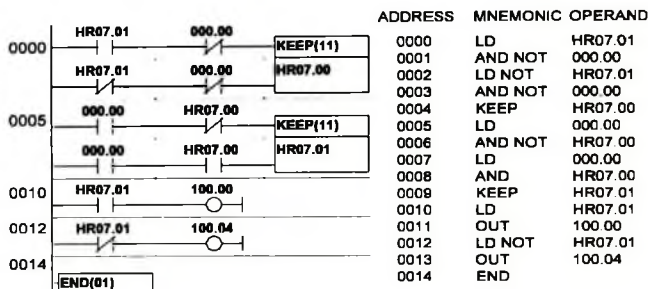
Hình 4.9a. RS Flip-Flop với đầu vào ưu tiên RESET.



Hình 4.9b. RS Flip-Flop với đầu vào ưu tiên SET.



Hình 4.10. Bit chốt (LATCHING).



Hình 4.11. T Flip-Flop xây dựng với 2 rơ le duy trì.

Ứng dụng của Flip-Flop được trình bày trong ví dụ về máy rửa khoai tây sau đây.

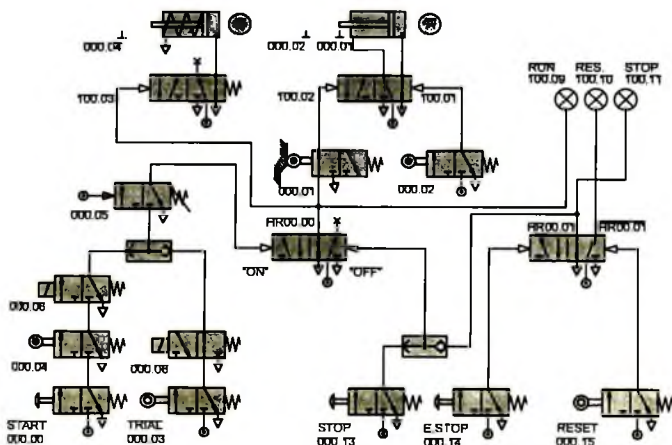
Ví dụ ứng dụng: Máy rửa khoai tây

<Yêu cầu công nghệ>

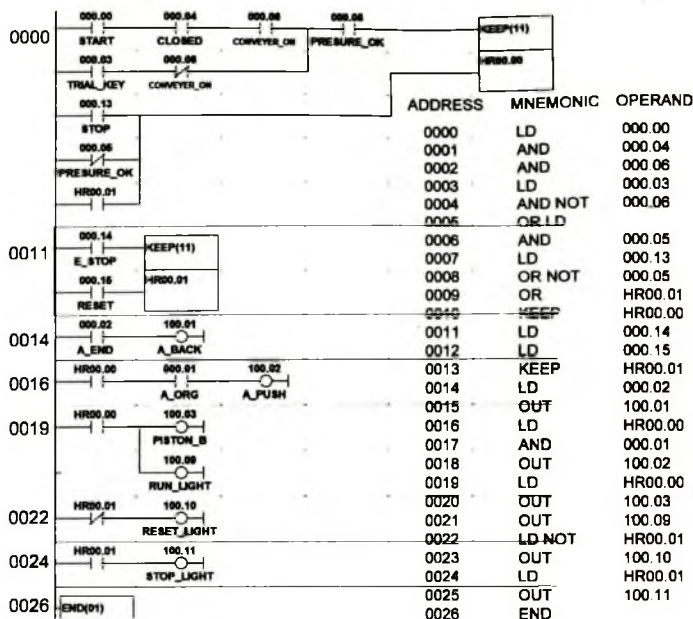
Một thùng khoai tây được làm sạch nhờ hai cơ cấu piston. Piston A rung khoai trong suốt quá trình rửa, piston B đẩy nắp thùng rửa đến vị trí khóa để tránh của mở trong quá trình làm việc. Áp suất yêu cầu ≥ 700 kPa. Để khởi động máy ấn nút START, cửa thùng rửa phải đang đóng, băng tải khoai phải đang chuyển động. Để điều chỉnh máy và chạy thử, thùng rửa được phép hoạt động khi nắp thùng không đóng, bằng cách ấn một nút khởi động đặc biệt TRIAL, để an toàn thì chế độ này bằng tải khoai phải dừng hẳn. Dừng máy bình thường bằng nút STOP và dừng khẩn cấp nhờ nút E.STOP, sau khi dừng khẩn cấp thì máy chỉ hoạt động lại được khi ấn nút RESET. Các chế độ chạy, dừng khẩn cấp, phục hồi của máy được hiển thị bằng các đèn chỉ thị tương ứng.

< Địa chỉ hóa các đầu/vào ra >

Bảng chỉ số công suất			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Nút START	100.01	Cơ cấu A tác động
000.01	Chuyển mạch báo A đã lùi hết	100.02	Cơ cấu A phục hồi
000.02	Chuyển mạch báo A đã tiến	100.03	Cơ cấu B
000.03	Khóa khởi động đặc biệt TRIAL	100.09	Đèn RUN
000.04	Chuyển mạch báo cửa đóng	100.10	Đèn RES (Reset)
000.05	Chuyển mạch áp suất	100.11	Đèn STOP
000.06	Chuyển mạch báo băng tải HĐ		
000.13	Nút STOP		
000.14	Nút E.STOP		
000.15	Nút RESET		



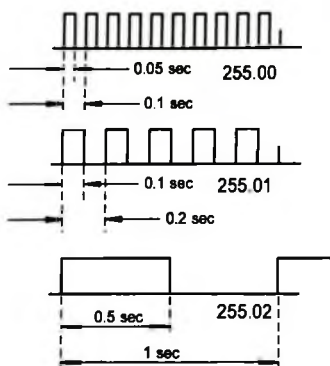
Hình 4.12. Sơ đồ máy rửa khoai tây.



Hình 4.13. Chương trình điều khiển dạng LAD và STL cho máy rửa khoai tây.

4.8. BỘ ĐỊNH THỜI (TIMER)

TIMER là bộ định thời gian có rất nhiều ứng dụng trong điều khiển. TIMER thực chất là một bộ đếm xung thời gian. Hệ thống cung cấp cho người sử dụng các cờ định thời trong vùng dữ liệu (SR) với các chu kỳ cố định: 0,01s, 0,1s, 1s,...



Trong đó:

SR255.00 là cờ định thời có chu kỳ là 0,1s.

SR255.01 là cờ định thời có chu kỳ là 0,2s.

SR255.02 là cờ định thời có chu kỳ là 1s.

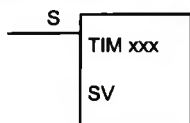
Hình 4.14. Ví dụ về các cờ định thời hệ thống của PLC OMRON.

TIMER có hai loại chính: On-Delay và OFF-Delay. Tuy nhiên từ một loại (ON-Delay) có thể xây dựng các kiểu TIMER khác như OFF-Delay, ON-OFF Delay,...

Các họ PLC khác nhau cung cấp các lệnh TIMER kiểu ON-Delay, OFF-Delay, hoặc ON-OFF Delay khác nhau. Dữ liệu dành cho TIMER được xác định trong vùng nhớ dữ liệu dành cho TIMER.

Cú pháp lệnh TIMER ON-Delay được trình bày trên hình 4.15.

Cú pháp lệnh:



Hình 4.15. Cú pháp lệnh TIMER.

Trong đó:

S (Start) là đầu vào điều kiện cho phép TIMER hoạt động;

xxx là số TC trong vùng nhớ dữ liệu dành cho TIMER/COUNTER;

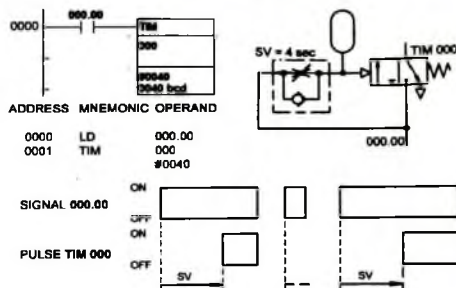
TIM xxx là toán hạng chỉ ra địa chỉ của giá trị hiện tại của TIMER (Present Value) hoặc cờ hoàn thành (Complement Flag);

SV là địa chỉ chứa giá trị đặt của TIMER.

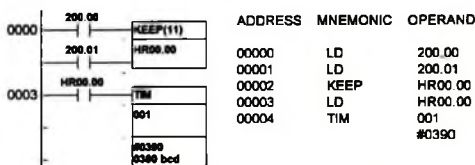
Hoạt động của TIMER (ON-Delay) được trình bày trên hình 4.16.

Cần chú ý khi đầu vào S không cho phép (S = 0) thì TIMER bị RESET. Khi đó giá trị hiện tại (TIM xxx) = (SV) và cờ hoàn thành TIM xxx vẫn không được lập. Trong trường hợp này có thể sử dụng chương trình để tạo ra Bit đầu ra trễ so với đầu vào một giá trị tùy ý như trên hình 4.17.

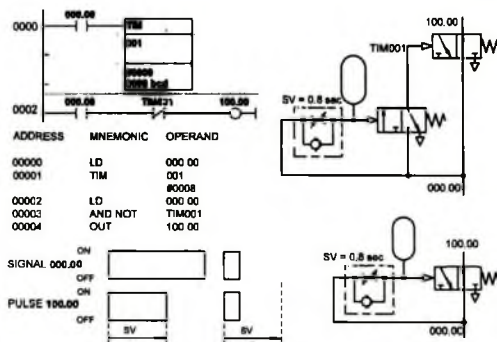
Hoạt động của TIMER (OFF-Delay) được trình bày trên hình 4.18.



Hình 4.16. Hoạt động của TIMER kiểu ON-Delay.

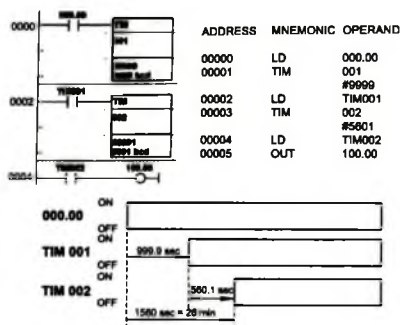


Hình 4.17. Chương trình tạo Bit ra trễ khi thời gian tồn tại của điều kiện vào là nhỏ.



Hình 4.18. Hoạt động của TIMER kiểu OFF-Delay.

Trong trường hợp thời gian trễ T yêu cầu quá lớn, vượt quá khả năng của một TIMER, có thể sử dụng phương pháp mắc nối tiếp TIMER. Chương trình mắc nối tiếp 2 TIMER được trình bày trên hình 4.19.

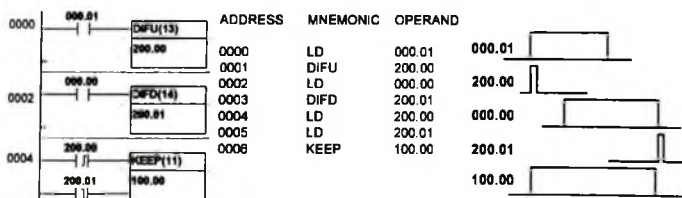


Hình 4.19. Mắc nối tiếp TIMER.

Các lệnh TIMER thông thường có độ phân giải là 0,1s, nghĩa là thời gian trễ T được xác định bằng công thức: $T = (SV) \times 0,1 [s]$. Một số lệnh TIMER có thể xác định độ phân giải thời gian 1s, 0,1s, 0,01s, ... Trường hợp cần độ phân giải cao (0,01s), có thể sử dụng các lệnh TIMER mở rộng, còn được gọi là High Speed TIMER (TIMH).

Một số họ PLC có lệnh TIMER mà giá trị hiện tại (PV) của TIMER có thể được duy trì khi mất nguồn hoặc điều kiện cho phép mất. Tính chất này có thể được thực hiện nếu sử dụng bộ đếm (COUNTER) để xây dựng TIMER.

Một trong các ứng dụng quan trọng của TIMER là xây dựng các lệnh DIFU(13) và DIFD(14), gọi là lệnh EDGE-PULSE TIMER. DIFU(13) là lệnh tạo xung ngắn khi điều kiện vào chuyển từ mức thấp lên cao (*Positive Edge*). DIFD(14) là lệnh tạo xung ngắn khi điều kiện vào chuyển từ mức cao xuống thấp (*Negative Edge*). Hình 4.20 mô tả hoạt động của lệnh DIFU(13) và DIFD(14).



Hình 4.20. Hoạt động của lệnh DIFU(13) và DIFD(14).

Một số ví dụ sau đây minh họa việc sử dụng TIMER trong các bài toán điều khiển.

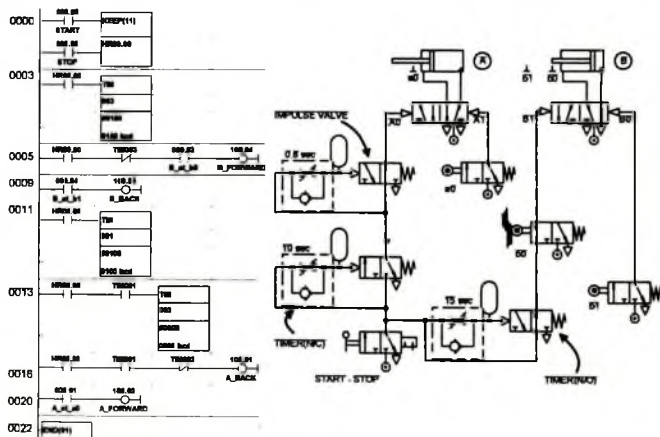
Ví dụ ứng dụng: Điều khiển các piston A, B

<Yêu cầu công nghệ>

Tại trạng thái bình thường, piston A kéo dài, piston B thụt vào. Khi có tín hiệu khởi động, piston B bắt đầu chuyển động qua lại giữa các chuyển mạch giới hạn b0 và b1 trong vòng 15s, piston A chưa tác động luôn mà đợi 10s sau khi ấn khởi động mới thụt vào, piston A về đến chuyển mạch giới hạn a0 thì lại đẩy ra và chu trình kết thúc.

<Địa chỉ hóa vào/ra>

Bảng phân công vào/ra cho PLC (b)			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.01	Chuyển mạch giới hạn a0	100.01	Piston A thụt vào (A0)
000.03	Chuyển mạch giới hạn b0	100.02	Piston A đẩy ra (A1)
000.04	Chuyển mạch giới hạn b1	100.04	Piston B đẩy ra (B1)
000.08	Nút START	100.05	Piston B thụt vào (B0)
000.09	Nút STOP		



Hình 4.21. Sơ đồ và chương trình ứng dụng TIMER điều khiển hoạt động 2 piston.

Trong ứng dụng trên thì do tác dụng của TIM002 với giá trị đặt là 0,8s, tín hiệu điều khiển thụt vào của piston A là dạng xung, tương tự sự dụng DIFU(13).

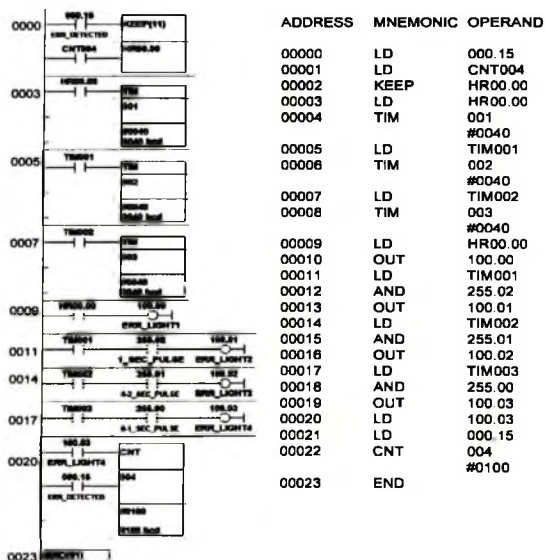
Ví dụ ứng dụng: Hệ thống cảnh báo sự cố máy bằng đèn nhấp nháy

<Yêu cầu công nghệ>

Hệ thống cảnh báo sự cố máy nhằm thông báo sự cố khi tín hiệu tức thời lỗi được ghi lại. Đèn cảnh báo 1 sáng và duy trì cho đến khi tắt hệ thống. Sau 4s, đèn nhấp 2 phát sáng với tần số 1 lần/s (0,5s sáng, 0,5s tối), 4s tiếp sau đó đèn nhấp 3 phát sáng với tần số 5 lần/s và 4s tiếp sau đó đèn nhấp 4 phát sáng với tần số 10 lần/s. Lúc này cả 4 đèn đều sáng, sau 100 lần nhấp sáng của đèn 4 thì hệ thống tự động bị ngừng hoạt động.

<Địa chỉ hóa vào/ra>

Chương trình công vào/ra cho thiết bị			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.15	Tín hiệu lỗi tức thời	100.00	Đèn báo lỗi 1
		100.01	Đèn nhấp 2 (1 lần/s)
		100.02	Đèn nhấp 3 (5 lần/s)
		100.03	Đèn nhấp 4 (10 lần/s)



Hình 4.22. Chương trình ứng dụng TIMER cho hệ thống báo lỗi máy bằng đèn nhấp.

Ví dụ ứng dụng: Máy tự động rửa rau

<Yêu cầu công nghệ>

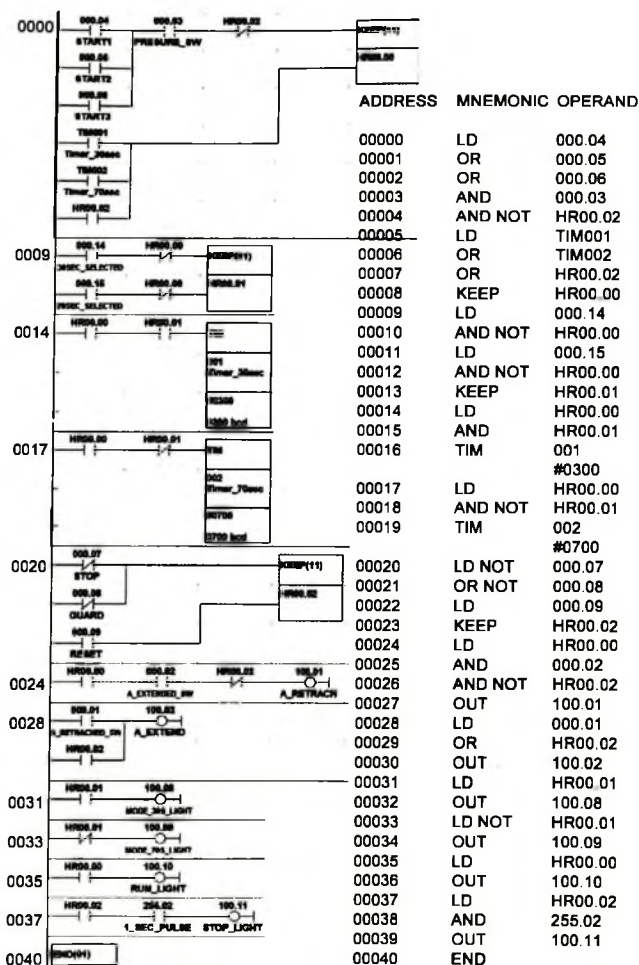
Rau được rửa nhờ một cơ cấu chuyển động qua lại với tần số có thể lựa chọn: 30s hoặc 70s. Máy tự động dừng khi ấn nút dừng khẩn cấp hoặc khi chuyển mạch giới hạn bị tác động trong khi cơ cấu chưa chạy hết hành trình. Ngay lúc này đèn cảnh báo nháy sáng với tần số 1 lần/s, đèn chỉ tắt khi ấn nút RESET. Trong suốt quá trình chạy đèn RUN sáng liên tục.



Hình 4.23. Sơ đồ máy rửa rau.

<Địa chỉ hóa vào/ra>

Bảng phân công thiết bị vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.01	Chuyển mạch báo A đã thụt vào	100.01	Cơ cấu A thụt vào
000.02	Chuyển mạch báo A đã đẩy ra	100.02	Cơ cấu A đẩy ra
000.03	Chuyển mạch áp suất	100.10	Đèn RUN
000.04,05,06	Nút Start1, Start2, Start3	100.11	Đèn STOP
000.07	Nút Stop		
000.08	Chuyển mạch giới hạn		
000.09	Nút Reset		
000.14	Chọn chế độ 30s		
000.15	Chọn chế độ 70s		

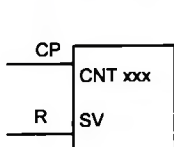


Hình 4.24. Chương trình ứng dụng TIMER điều khiển máy rửa rau.

4.9. BỘ ĐẾM (COUNTER)

Bộ đếm dùng để đếm sự kiện xảy ra. Bộ đếm có các loại bộ đếm tăng (COUNTER-Up), Bộ đếm giảm (COUNTER-Down) và bộ đếm thuận nghịch (COUNTER Up/Down). Các họ PLC khác nhau cung cấp các lệnh COUNTER khác nhau. Dữ liệu dành cho COUNTER được xác định trong vùng nhớ dữ liệu dành cho COUNTER.

Cú pháp lệnh COUNTER được trình bày trên hình 4.25.



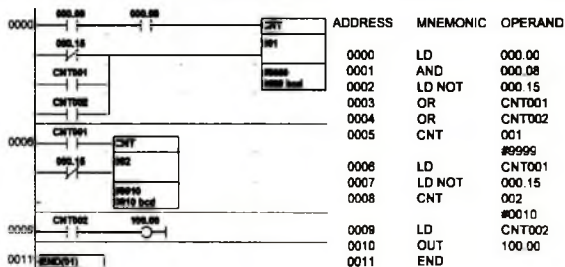
Hình 4.25. Cú pháp lệnh COUNTER.

Trong đó:

- CP là đầu vào xung đếm. Khi ở đầu vào CP chuyển từ thấp lên cao thì giá trị của bộ đếm sẽ tăng (giảm) 1 đơn vị. Vì vậy, các tín hiệu trước khi vào bộ đếm thường được qua lệnh DIFU(13) hoặc DIFD (14).
- xxx là số TC trong vùng dữ liệu dành cho TIMER/COUNTER.
- CNTxxx là toán hạng chỉ ra địa chỉ của giá trị hiện tại PV của COUNTER (Present Value) hoặc cờ hoàn thành.
- SV là địa chỉ chứa giá trị đặt của COUNTER.
- R là đầu vào RESET.

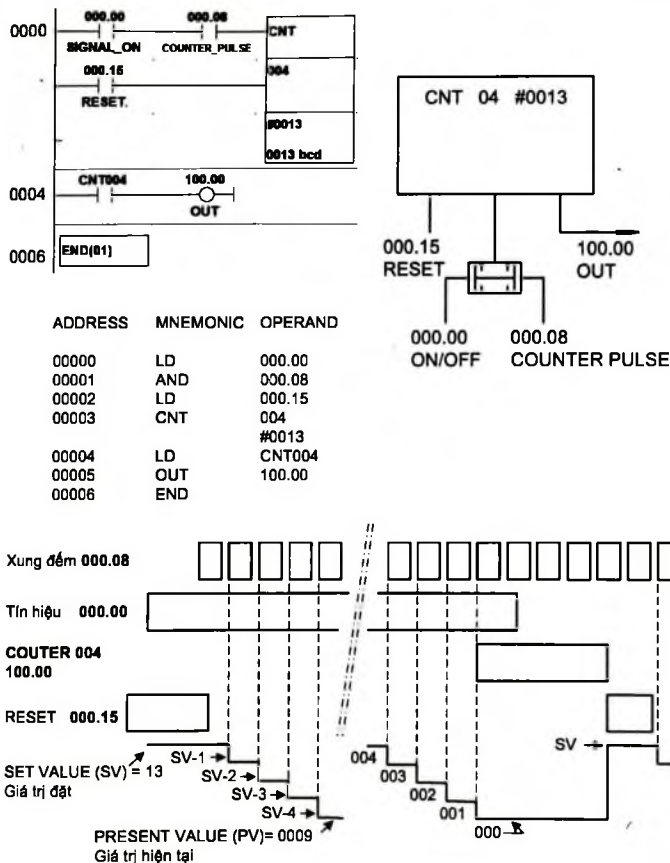
Hoạt động của lệnh COUNTER được trình bày trên hình 4.27. Trong trường hợp này, PLC cung cấp lệnh COUNTER-Down.

Có một số điểm cần chú ý: Khi có tín hiệu tích cực (ON) ở đầu vào RESET thì giá trị hiện tại PV bằng giá trị đặt SV và cờ hoàn thành bị xóa (CNTxxx = 0). Khi ở đầu vào RESET ở trạng thái OFF, mỗi khi xuất hiện xung ở đầu vào CP thì giá trị PV của bộ đếm giảm 1 đơn vị. Khi giá trị PV = 0, thì cờ hoàn thành CNTxxx được lập (CNTxxx = 1). Khi mất nguồn cung cấp, giá trị PV của bộ đếm vẫn được duy trì.



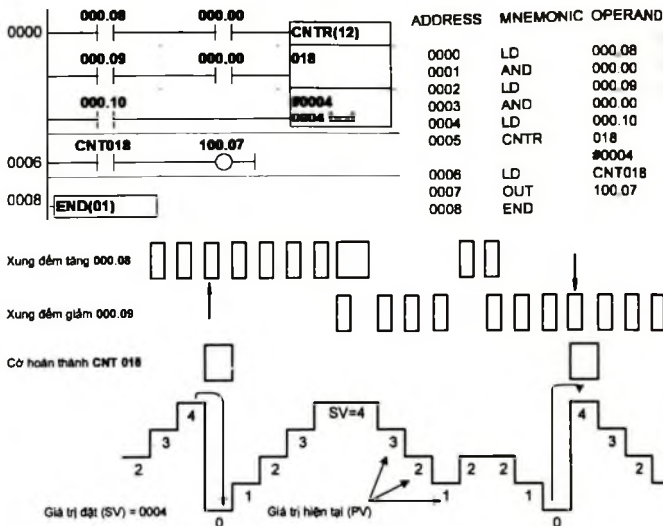
Hình 4.26. Mắc nối tiếp hai bộ đếm.

Trong trường hợp muốn mở rộng khả năng của bộ đếm, có thể sử dụng mắc nối tiếp các bộ đếm như trên hình 4.26.



Hình 4.27. Hoạt động của COUNTER.

Các lệnh bộ đếm thuận nghịch (COUNTER Up/Down) có thể được sử dụng trong hệ lệnh mở rộng (CNTR). Lệnh bộ đếm thuận nghịch được mô tả trên hình 4.28.



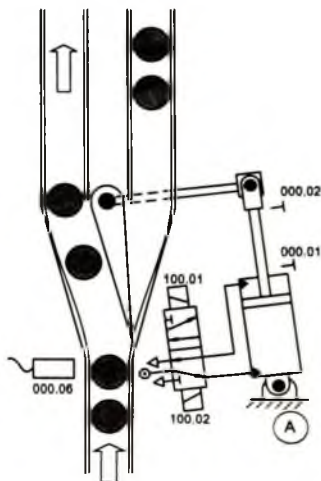
Hình 4.28. Bộ đếm thuận nghịch.

Một số ứng dụng của COUNTER được trình bày sau đây.

Ví dụ ứng dụng: Bộ điều khiển băng tải phân loại

<Yêu cầu công nghệ>

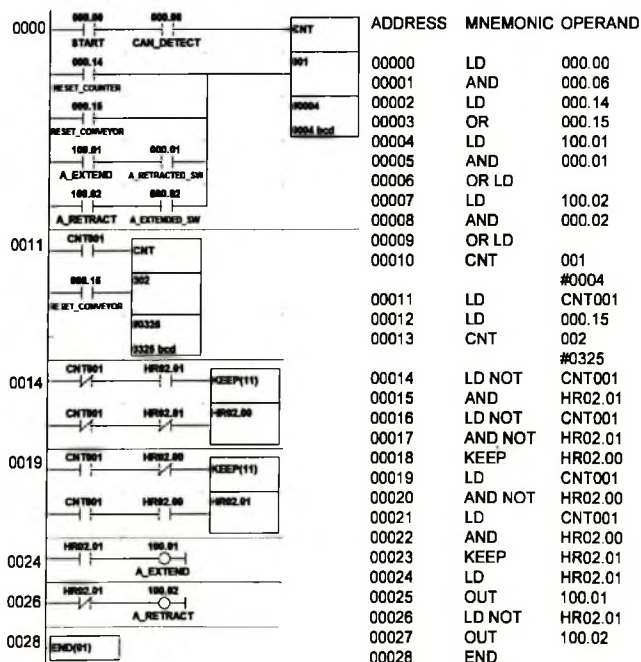
Hệ thống dẫn hướng phân chia hộp ra hai đường song song được điều khiển nhờ một tay gạt gắn piston. Hộp chạy trên băng tải được phát hiện khi đến vị trí có tay gạt nhờ một cảm biến. Tay gạt ban đầu cho 4 hộp đi qua, sau đó gạt 4 hộp tiếp sang đầu ra khác song song, sau khi đếm được 1300 hộp, hệ thống tự động dừng lại. Hệ thống khởi động khi được ấn START, một nút ấn RESET bộ đếm của tay gạt và một nút khác RESET cả hệ thống băng tải.



Hình 4.29. Sơ đồ băng tải phân loại.

<Địa chỉ hóa vào/ra>

Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Nút START	100.01	Tay gạt sang trái
000.01	Chuyển mạch báo tay gạt sang trái	100.02	Tay gạt sang phải
000.02	Chuyển mạch báo tay gạt sang phải		
000.06	Cảm biến phát hiện hộp		
000.14	Nút RESET bộ đếm của tay gạt		
000.15	Nút RESET bằng tải		



Hình 4.30. Chương trình ứng dụng COUNTER điều khiển băng tải phân loại.

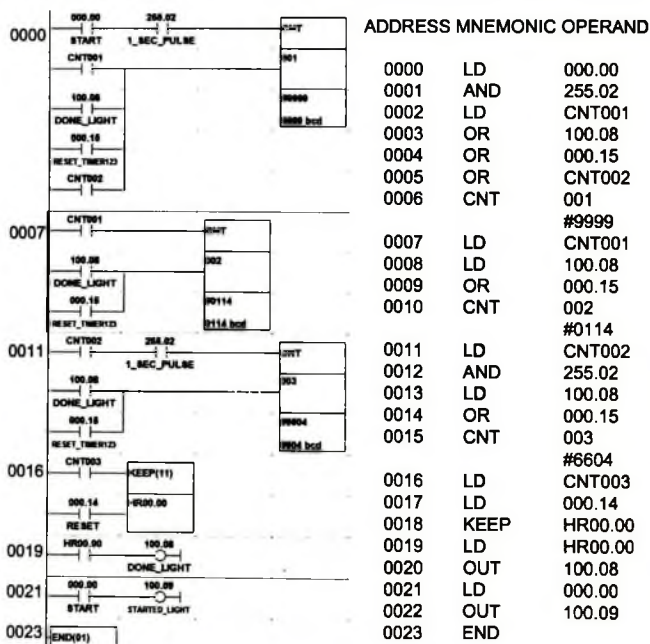
Ví dụ ứng dụng: Hệ thống thông báo hoàn thành phản ứng hóa học

<Yêu cầu công nghệ>

Phản ứng hóa học của một loại keo sử dụng trong khi sản xuất xe máy sẽ hoàn thành sau chính xác 13 ngày, 6 giờ, 28 phút, 10 giây. Cần thiết lập hệ thống để thông báo thời điểm trên.

<Địa chỉ hóa vào/ra >

Hàng phân công thiết bị vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Start	100.08	Thông báo phản ứng đã hoàn thành
000.15	Nút RESET TIMER 1,2,3	100.09	Đèn báo phản ứng bắt đầu
000.14	Nút RESET hệ thống hoàn toàn		

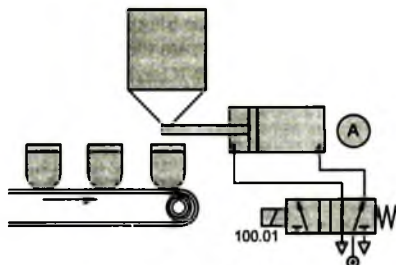


Hình 4.31. Chương trình ứng dụng COUNTER thông báo hoàn thành phản ứng hóa học.

Ví dụ ứng dụng: Bộ điều khiển bơm rót

<Yêu cầu công nghệ>

Các chai rỗng được vận chuyển trên băng tải đến vị trí rót và được phát hiện bởi 1 sensor, khi đó piston A trượt vào để mở miệng rót, thời gian rót cho mỗi chai là 2,0s, hết thời gian này piston A tự động đóng lại để hòa thành một chu trình rót. Lưu ý, để tránh việc mất nguồn của PLC ảnh hưởng đến lượng rót thì bộ TIMER điều khiển rót phải được duy trì (sử dụng COUNTER).



Hình 4.32. Sơ đồ hệ thống bơm rót.

<Địa chỉ hóa vào/ra >

Bảng phân công thiết bị			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến phát hiện chai đến vị trí rót	100.01	Piston A trượt vào (mở miệng rót)

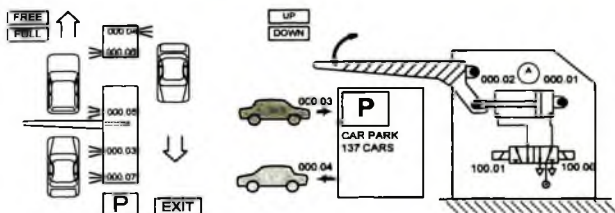
		ADDRESS	MNEMONIC	OPERAND
0000	000.00 DIFUK(12)	0000	LD	000.00
	000.00	0001	DIFU	200.00
0002	000.00	0002	LD	200.00
	CNT006	0003	LD	CNT006
	HR08.00	0004	KEEP	HR08.00
0005	HR08.00	0005	LD	HR08.00
	255.02	0006	AND	255.02
	1_SEC_PULSE	0007	LD	CNT006
	CNT006	0008	CNT	006
				#0002
0009	HR08.00	0009	LD	HR08.00
	100.01	0010	OUT	100.01
0011	END(W1)	0011	END	

Hình 4.33. Chương trình ứng dụng COUNTER điều khiển hệ thống bơm rót.

Ví dụ ứng dụng: Bộ điều khiển bãi đỗ xe

<Yêu cầu công nghệ>

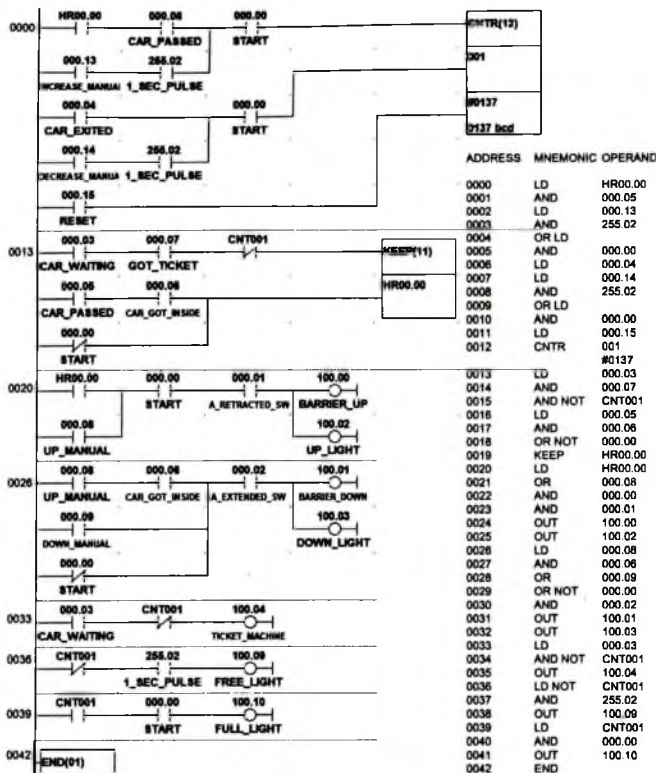
Bãi xe tự động sử dụng một bộ đếm hai chiều (thuận và ngược) để đếm số xe trong bãi. Việc vào ra của các xe được điều khiển nhờ một rào chắn tự động và máy bán vé. Một cảm biến phát hiện ô tô đến vị trí máy phát vé, tài xế ngắt vé in từ máy bán vé và rào chắn nâng cho xe qua. Để đảm bảo xe đã qua rào chắn và ngăn chặn xe thứ hai qua rào mà không trả tiền hoặc lấy vé thì rào chắn chỉ hạ sau khi cả hai cảm biến phát hiện xe qua rào cùng ON. Bãi xe chứa tối đa 137 xe. Rào chắn có thể nâng hoặc hạ bằng tay, bộ đếm xe trong bãi có thể tăng hoặc giảm bằng các nút ấn tương ứng. Nút RESET đưa bộ đếm về 0. Đèn FULL báo bãi xe đầy, đèn FREE báo bãi xe còn chỗ trống, đèn xanh UP báo hàng rào đang nâng, đèn đỏ DOWN báo hàng rào đang hạ xuống.



Hình 4.34. Mô hình bãi đỗ xe sử dụng PLC điều khiển.

<Địa chỉ hóa vào/ra >

Bảng phân công thiết bị vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Bật hệ thống rào và máy bán vé	100.00	Nâng rào chắn
000.01	Piston A thụt vào (rào hạ)	100.01	Hạ rào chắn
000.02	Piston A kéo ra (rào nâng)	100.02	Đèn xanh UP
000.03	Có xe chờ vào	100.03	Đèn đỏ DOWN
000.04	Xe ra khỏi bãi	100.04	Tín hiệu khởi động máy bán vé
000.05	Xe vừa qua rào	100.09	Đèn FREE báo bãi xe đã đầy
000.06	Xe đang tiến vào bãi	100.10	Đèn FULL báo bãi xe còn trống
000.07	Vé đã được tài xế lấy		
000.08	Nâng rào bằng tay		
000.09	Hạ rào bằng tay		
000.13	Tăng COUNTER		
000.14	Giảm COUNTER		
000.15	RESET COUNTER		

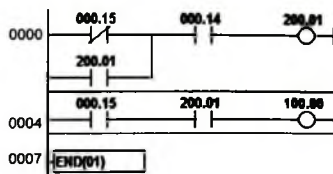


Hình 4.35. Chương trình ứng dụng COUNTER điều khiển bãi đỗ xe.

4.10. LẬP TRÌNH CÁC KHỐI PHỨC HỢP

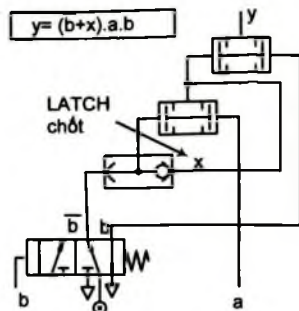
Chương trình thực hiện các hàm điều khiển logic, được xây dựng trên cơ sở các khối chức năng phức hợp. Các khối chức năng phức hợp được tạo thành trên cơ sở các lệnh của PLC. Trong phần này sẽ trình bày một số các khối chức năng phức hợp.

Khối chức năng thực hiện mạch logic A trước B (A Before B) được trình bày trên hình 4.36.



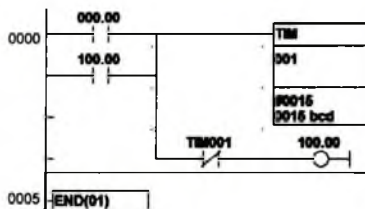
ADDRESS MNEMONIC OPERAND

0000	LD NOT	000.15
0001	OR	200.00
0002	AND	000.14
0003	OUT	200.00
0004	LD	000.15
0005	AND	200.00
0006	OUT	100.00
0007	END	



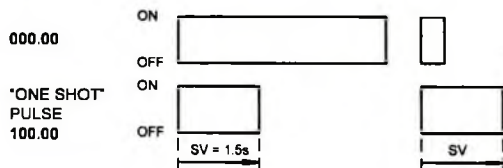
Hình 4.36. Mạch điều khiển trình tự A trước B.

Khởi chức năng phát một xung ngắn (one-shot) có độ rộng cố định không phụ thuộc độ dài tín hiệu vào được trình bày trên hình 4.37.



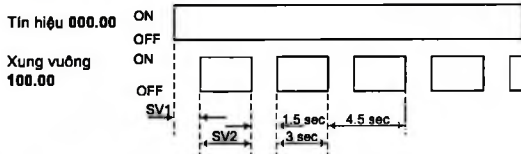
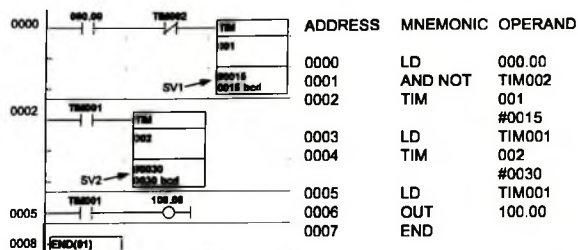
ADDRESS MNEMONIC OPERAND

0000	LD	000.00
0001	OR	100.00
0002	TIM	001
0003	AND NOT	#0015
0004	OUT	100.00
0005	END	



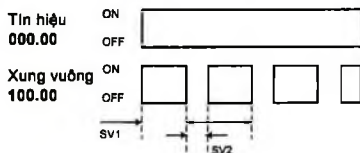
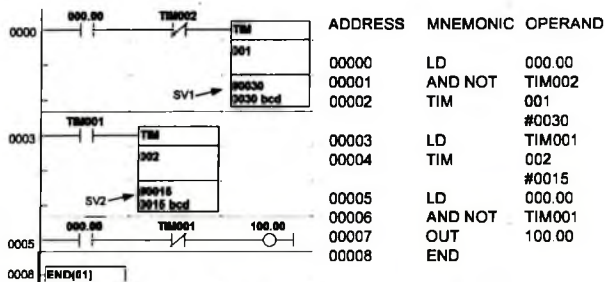
Hình 4.37. Khối tạo xung "one-shot".

Khởi chức năng phát dãy xung vuông có chu kỳ ($T = SV1 + SV2$) được trình bày trên hình 4.38 và 4.39.



Trong đó: SV1: Thời gian trễ; SV2: Độ rộng xung.

Hình 4.38. Khởi tạo xung vuông.



Trong đó: SV1: Độ rộng xung; SV2: Thời gian trễ.

Hình 4.39. Khởi tạo xung vuông.

(Start) và từ dữ liệu cuối (End) tạo ra thanh ghi dịch. Nếu hai địa chỉ Start và End trùng nhau thì thanh ghi dịch có kích thước bằng một từ dữ liệu 16 Bit.

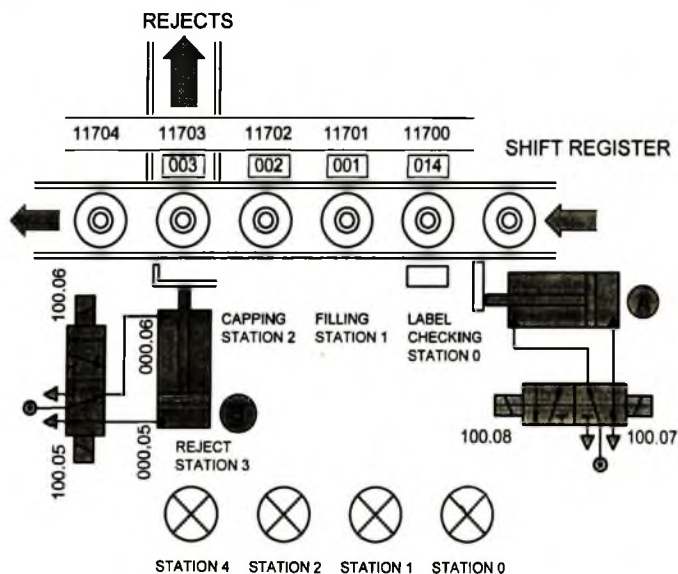
Khi có một xung ở đầu vào, thanh ghi dịch được dịch trái một Bit. Dữ liệu vào được nạp vào Bit trẻ nhất (LSB), còn Bit già nhất (MSB) bị mất. Khi đầu RESET có tín hiệu tích cực (ON) thì nội dung thanh ghi dịch bị xóa. Một số ứng dụng của thanh ghi dịch được trình bày trong phụ lục.

Một số ứng dụng của các khối chức năng được trình bày trong các ví dụ sau:

Ví dụ ứng dụng: Máy nạp và đóng nắp hộp

<Yêu cầu công nghệ>

Hộp được vận chuyển trên một băng tải qua bốn trạm thao tác: kiểm nhãn, nạp (rót), đóng nắp, loại bỏ (nếu không đạt chất lượng). Băng tải được điều khiển tự động nhờ piston A theo từng bước. Qua trạm kiểm nhãn, nếu hộp được phát hiện có nhãn thì sẽ được rót và đóng nắp tự động ở các trạm tương ứng, nếu không thì sẽ bị đẩy ra tại vị trí 3 nhờ piston B, các hộp đã được rót và đóng nhãn thì tiếp tục theo băng tải sang công đoạn đóng thùng. Sơ đồ công nghệ được trình bày trên hình 4.42.



Hình 4.42. Sơ đồ máy nạp và đóng hộp.

<Địa chỉ hóa vào/ra>

Bảng phân công thiết bị và địa chỉ			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến hộp ở Station0 (kiểm nhân)	100.00	Đèn nháy
000.01	Cảm biến hộp ở Station1 (rót)	100.01	Khởi động rót
000.02	Cảm biến hộp ở Station2 (đóng nắp)	100.02	Khởi động đóng nắp hộp
000.03	Cảm biến hộp ở Station3 (loại bỏ)	100.03	Đèn RUN
000.05	Piston B thụt vào	100.04	Đèn STOP
000.06	Piston B đẩy ra	100.05	Piston B thụt vào
000.13	Nút START khởi động máy	100.06	Piston B đẩy ra
000.14	Cảm biến phát hiện hộp không nhân	100.07	Piston A thụt vào
000.15	Nút STOP-RESET	100.08	Piston A đẩy ra
		100.09	Đèn loại bỏ hộp ở Station0
		100.10	Đèn loại bỏ hộp ở Station1
		100.11	Đèn loại bỏ hộp ở Station2

Chương trình điều khiển máy nạp và đóng hộp được trình bày trên hình 4.43.

Ví dụ ứng dụng: Chương trình khóa số (4 digit) (Cho kết bực, máy đặc biệt)

<Yêu cầu công nghệ>

Để an toàn thì khi khởi động hoặc dừng máy, người sử dụng bắt buộc phải nhập một mật khẩu gồm 4 chữ số theo một trình tự nhất định từ một bàn phím. Khi một trong các chữ số trên bị sai, lập tức đèn cảnh báo sáng.

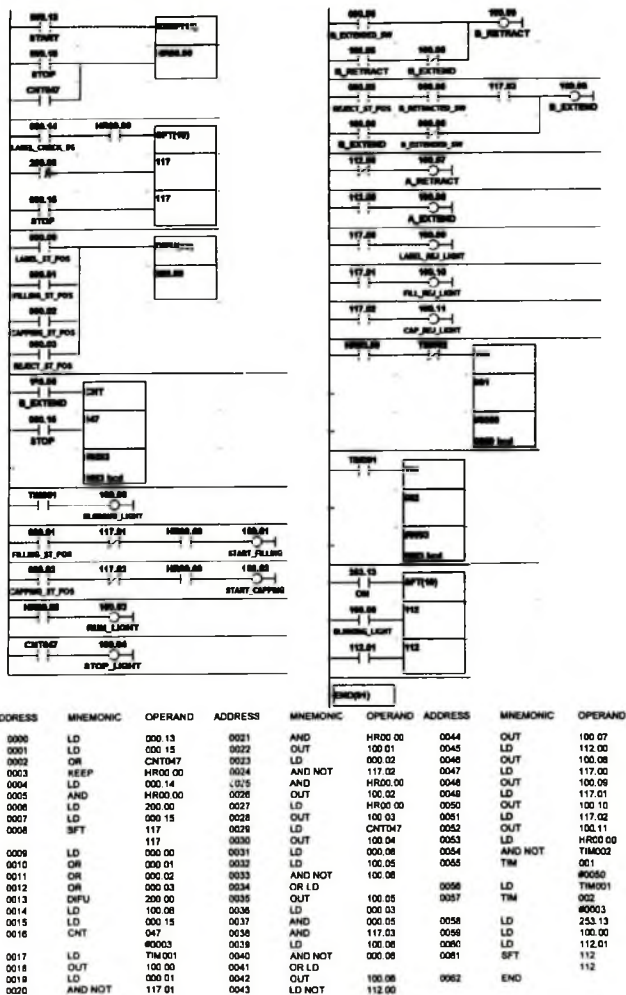
Mật khẩu như sau:

2	8	3	9
---	---	---	---

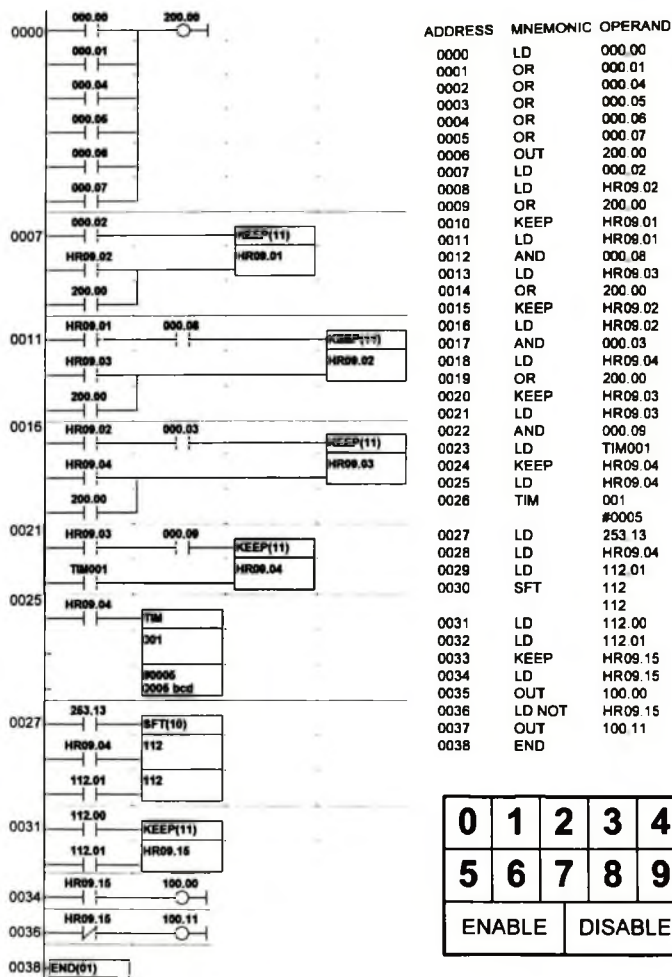
<Địa chỉ hóa vào/ra>

Bảng phân công thiết bị và địa chỉ			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00 ÷ 000.09	Phím bấm 0 đến 9	100.00	Đèn MACHINE ENABLE
		100.01	Đèn MACHINE DISABLE

Chương trình điều khiển mã khóa 4 số được trình bày trên hình 4.44.



Hình 4.43. Chương trình điều khiển máy nạp và đóng hộp.



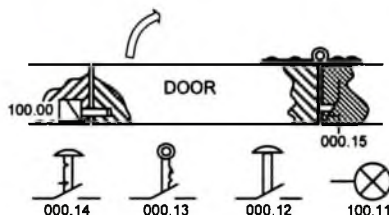
Hình 4.44. Chương trình mã khóa số 4 digits.

Ví dụ ứng dụng: Khóa số đóng mở cửa

<Yêu cầu công nghệ>

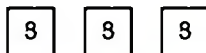
Nhằm tránh việc ra vào mà không được sự cho phép, cửa được điều khiển tự động đóng mở dựa trên các điều kiện sau đây:

- + Cửa luôn mở được bằng một chìa khóa riêng.
- + Cửa cũng có thể mở được bằng cách gõ đúng mật khẩu gồm ba chữ số 3 8 5 trong vòng 30s, quá thời gian này cửa tự động hủy lệnh mở.
- + Việc ấn sai và giả mạo đều bị ghi lại, nếu ấn sai 12 lần thì đèn nháy cảnh báo với tần số 1 lần/s, đèn này chỉ tắt được từ nút RESET bên trong phòng.
- + Sau 4 lần ấn sai, cửa chỉ có thể mở được bằng chìa khóa hoặc bằng cách nhập mật khẩu chính xác trong vòng 5s.



Hình 4.45. Mô hình khóa cửa số.

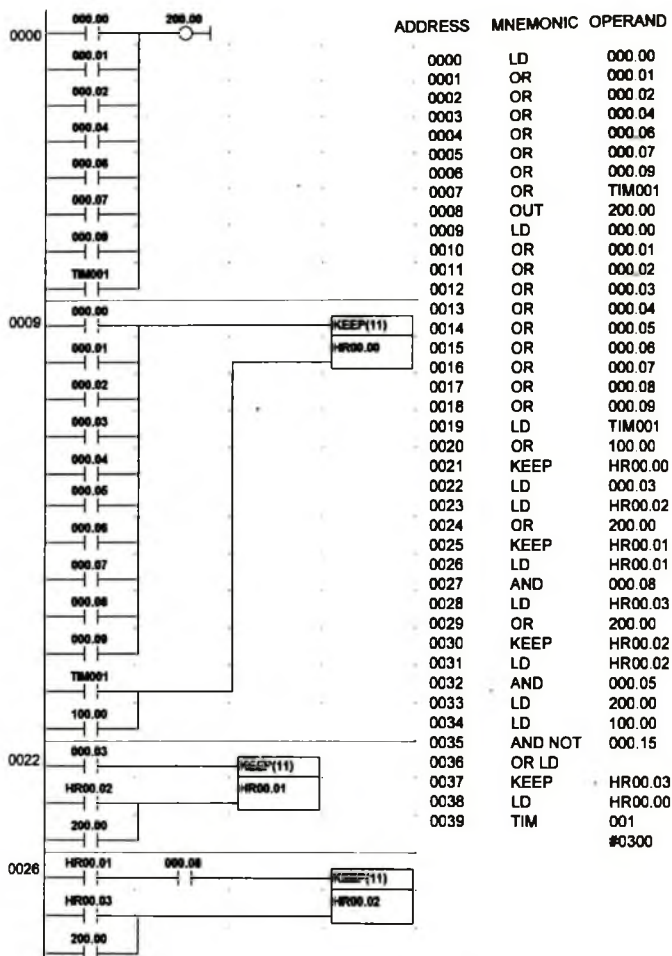
Mật khẩu như sau:



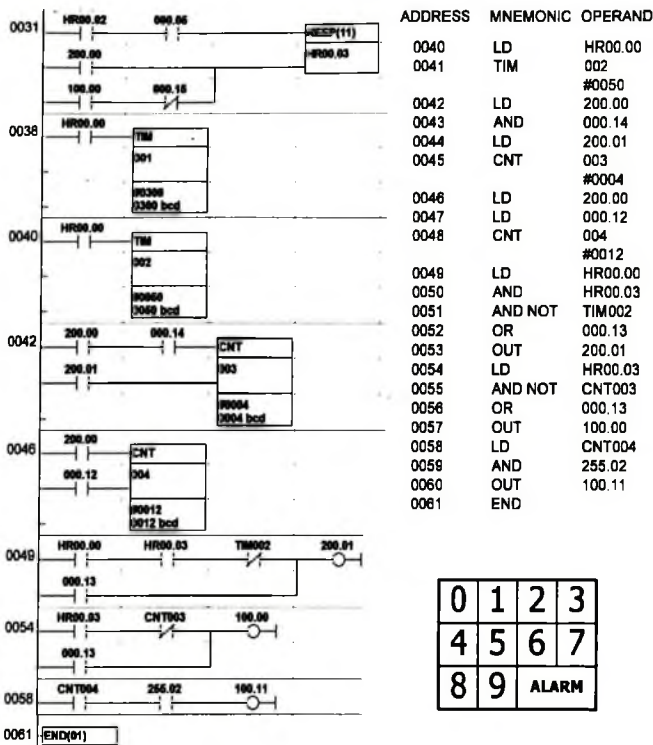
<Địa chỉ hóa vào/ra>

Bảng phân công thiết bị vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00 ÷ 000.09	Phím bấm 0 đến 9	100.00	Mở khóa cửa
000.12	Nút RESET đèn cảnh báo	100.01	Đèn cảnh báo
000.13	Chìa khóa cửa		
000.14	Công tắc chống gõ mật khẩu		
000.15	Nút RESET mạch mã hóa		

Chương trình điều khiển đóng mở cửa tự động được trình bày trên hình 4.46a và 4.46b.



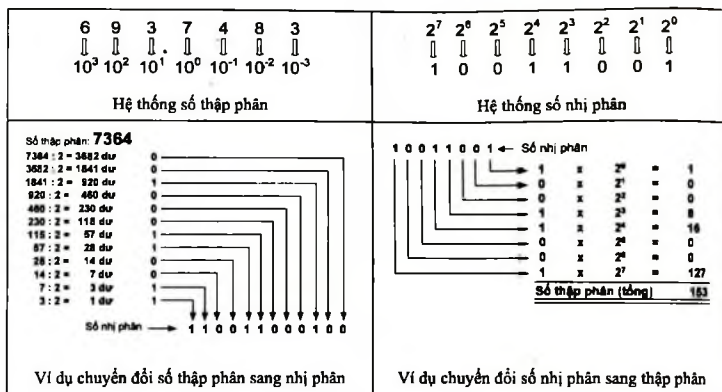
Hình 4.46a. Chương trình điều khiển cửa.



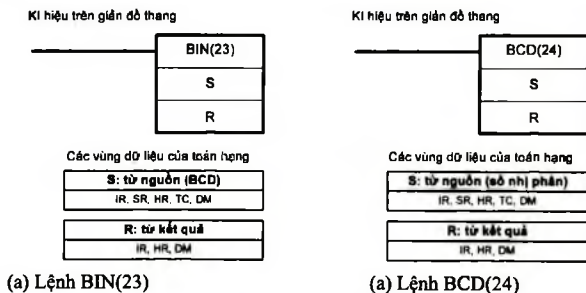
Hình 4.46b. Chương trình điều khiển cửa (tiếp).

4.11. LẬP TRÌNH CÁC HÀM SỐ HỌC

Trong PLC có thể biểu diễn số ở dạng nhị phân, BCD, thập phân, Hexa. Trên hình 4.47 trình bày hai hệ thống số nhị phân (*Binary*), thập phân (*Decimal*) và phương pháp chuyển đổi bằng tay giữa hai hệ thống này. Trong PLC, việc chuyển đổi cơ số có thể được thực hiện bằng các lệnh đổi cơ số. Ví dụ, lệnh BIN (23) chuyển đổi từ mã BCD sang nhị phân, lệnh BCD (24) chuyển đổi từ mã nhị phân sang BCD. Dữ liệu được lưu giữ trong các từ dữ liệu. Cú pháp của lệnh BIN và BCD được trình bày trên hình 4.48.



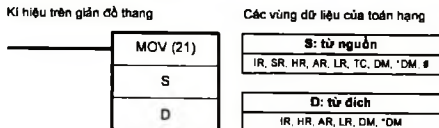
Hình 4.47. Hệ thống số nhị phân, thập phân và phương pháp chuyển đổi bằng tay.



Hình 4.48. Lệnh chuyển đổi cơ số BIN(23), BCD(24).

PLC có thể thực hiện các lệnh chuyển dữ liệu, tính toán số học, lệnh dịch từ dữ liệu.

Lệnh chuyển dữ liệu MOV (21)



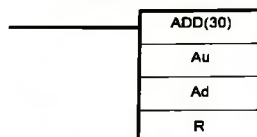
Hình 4.49. Lệnh chuyển dữ liệu MOVE(21).

Trong đó: S: địa chỉ của từ nguồn (Source Word)

D: địa chỉ của từ đích (Destination Word)

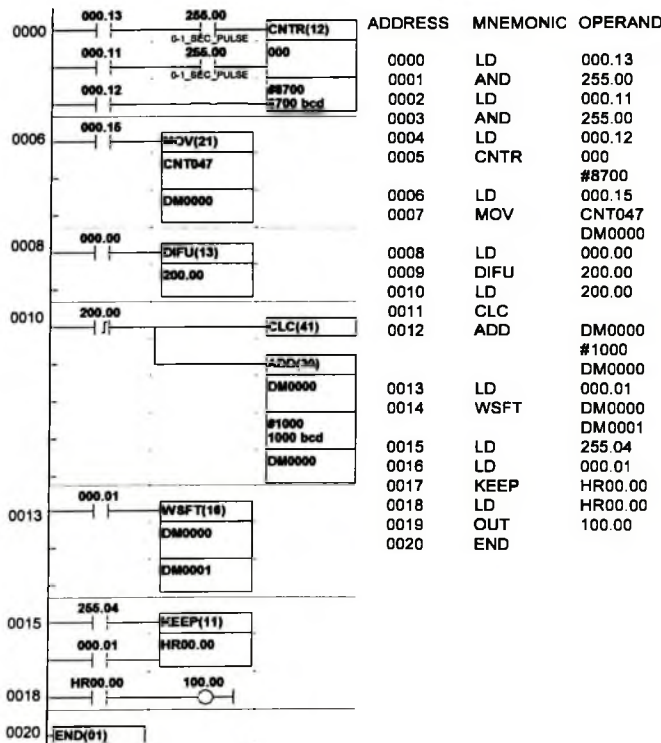
Lệnh cộng hai số BCD: ADD(30)

Kí hiệu trên giản đồ thang



Các vùng dữ liệu của toán hạng

Au: Số cộng (BCD)
IR, SR, HR, TC, DM, #
Ad: Số cộng (BCD)
IR, SR, HR, TC, DM, #
R: Từ kết quả
IR, HR, DM



Hình 4.50. Lệnh cộng hai số BCD ADD(30) và ứng dụng.

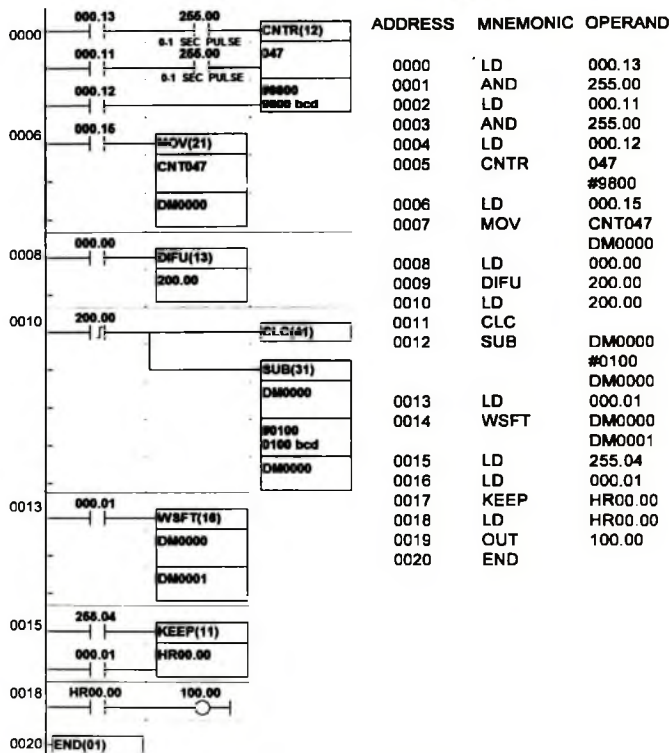
Lệnh trừ hai số BCD: SUB(31)

Kí hiệu trên giản đồ thang

SUB(31)
Mi
Su
R

Các vùng dữ liệu của toán hạng

Mi: Số bị trừ (BCD)
IR, SR, HR, TC, DM, #
Su: Số trừ (BCD)
IR, SR, HR, TC, DM, #
R: Từ kết quả
IR, HR, DM



Hình 4.51. Lệnh trừ hai số BCD SUB(31) và ứng dụng.

Lệnh nhân hai số BCD: MUL(32)

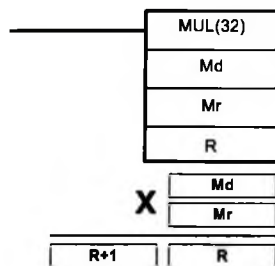
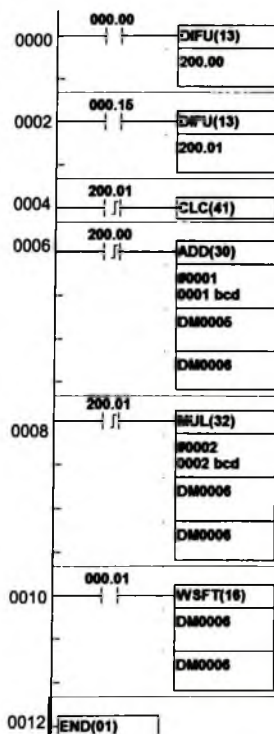
Kí hiệu trên giản đồ thang

Trong đó:

Md: Số nhân

Mr: Thừa số

R: Địa chỉ bắt đầu lưu kết quả



ADDRESS MNEMONIC OPERAND

0000	LD	000.00
0001	DIFU	200.00
0002	LD	000.15
0003	DIFU	200.01
0004	LD	200.01
0005	CLC	
0006	LD	200.00
0007	ADD	#0001 DM0005 DM0006
0008	LD	200.01
0009	MUL	#0002 DM0005 DM0006
0010	LD	000.01
0011	WSFT	DM0005 DM0006
0012	END	

Hình 4.52. Lệnh nhân hai số BCD MUL(32) và ứng dụng.

Lệnh chia hai số BCD: DIV(33)

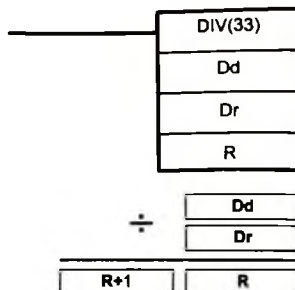
Kí hiệu trên giản đồ thang

Trong đó:

Ms: Số bị chia

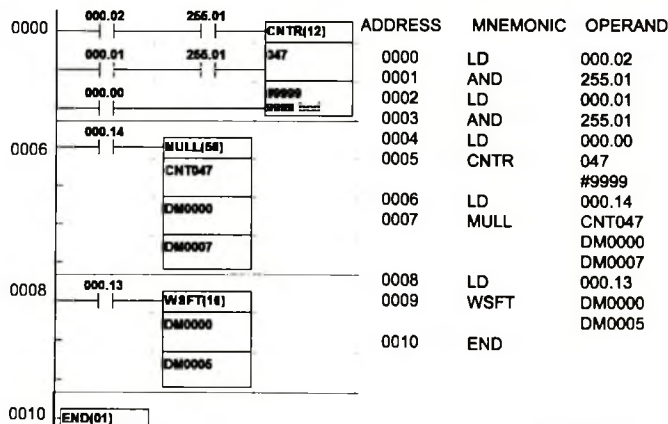
Mr: Số chia

R: Địa chỉ bắt đầu lưu kết quả



Hình 4.53. Lệnh chia hai số BCD DIV(33).

Ngoài các lệnh nhân, chia thông thường, PLC còn cung cấp các lệnh nhân, chia với độ chính xác kép tương ứng: MULL(56) và DIVL(57), một ứng dụng tương ứng được trình bày trên hình 4.54.



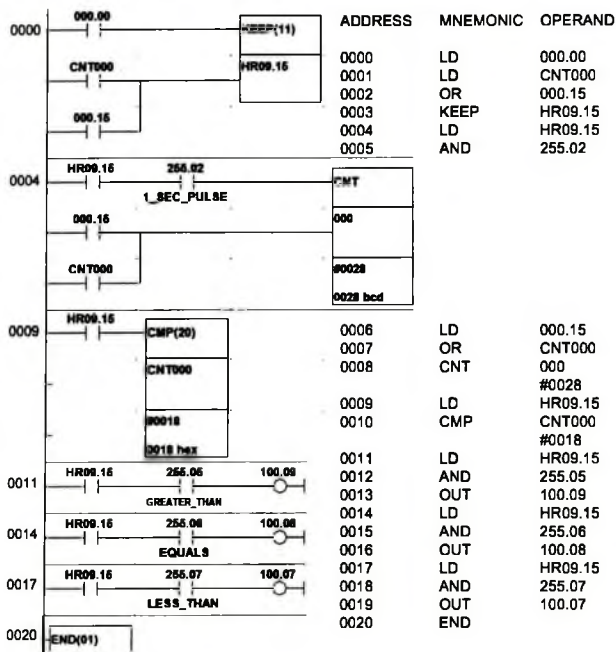
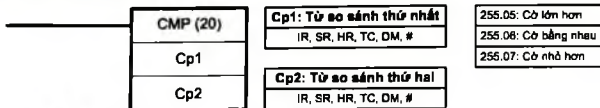
Hình 4.54. Ví dụ về ứng dụng dùng lệnh nhân với độ chính xác kép MULL(56).

Lệnh so sánh hai số BCD: CMP(20)

Kí hiệu trên giản đồ thang

Các vùng dữ liệu của toán hạng

Các cờ so sánh



Hình 4.55. Lệnh so sánh hai số BCD CMP(20).

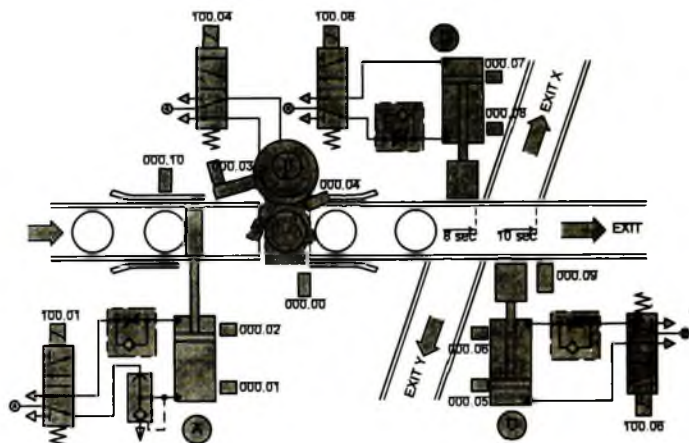
Một số các ví dụ ứng dụng sử dụng lệnh số học, so sánh... chuyển dữ liệu được trình bày trong phần phụ lục.

Ví dụ ứng dụng: Máy nạp và đóng hộp (chức năng cân kiểm tra và loại hộp không đạt khối lượng)

<Yêu cầu công nghệ>

Máy có chức năng kiểm tra khối lượng hộp sau khi rót, các hộp thiếu hoặc thừa khối lượng đều bị loại ra. Sơ đồ máy của trạm cân – tách trình bày trên hình 4.56.

Piston A dùng hộp trên băng chuyền và thả hộp sang trạm cân khi trạm cân trống. Cơ cấu B quét các hộp đã cân sang băng tải đầu ra. Piston C đẩy các hộp thiếu khối lượng sang băng tải X sau khi hộp ra khỏi trạm cân 10 giây, các hộp này sẽ được rót thêm cho đến khi đủ khối lượng. Piston D đẩy các hộp thừa khối lượng sang băng tải Y sau khi hộp qua trạm cân 8 giây, các hộp này sẽ được đổ dung dịch trở lại bể. Khối lượng tiêu chuẩn cho mỗi hộp là 1000 g với dung sai cho phép 10 g.



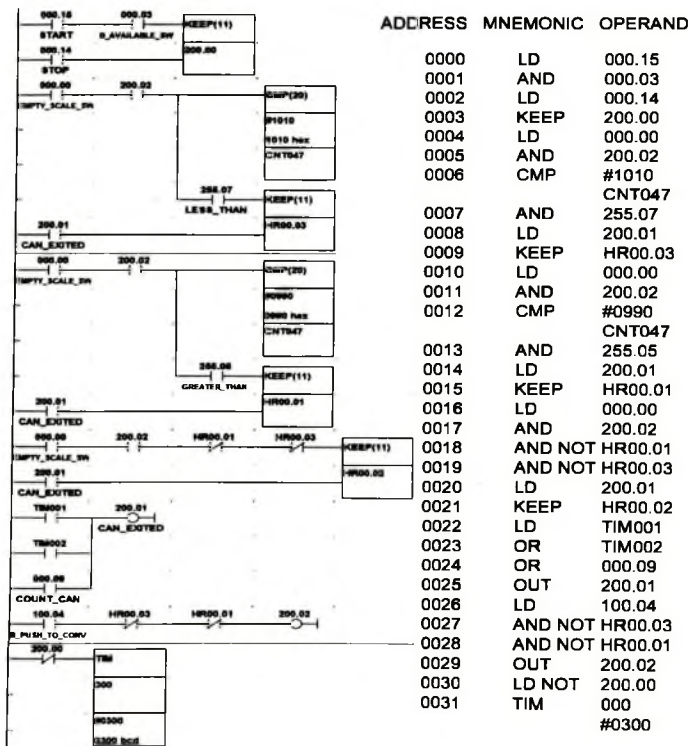
Hình 4.56. Sơ đồ máy nạp và đóng hộp phân trạm cân – tách.

<Địa chỉ hóa vào/ra>

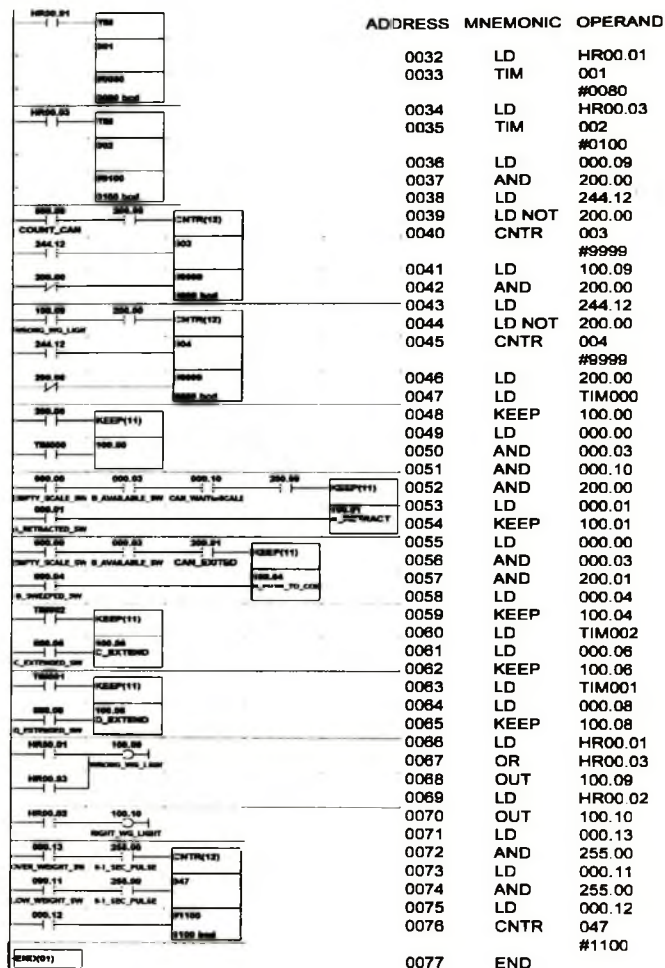
Bảng phân công thiết bị nạp/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Trạm cân trọng (không có hộp)	100.01	Piston A thả hộp sang trạm cân
000.01	Piston A thực vào	100.04	Cơ cấu B quét hộp sang băng tải
000.02	Piston A đẩy ra	100.06	Piston C đẩy hộp sang băng tải X
000.03	Cơ cấu B rời	100.08	Piston D đẩy hộp sang băng tải Y
000.04	Cơ cấu B quét hộp ra	100.09	Đèn báo có hộp sai khối lượng
000.05	Piston C thực vào	100.10	Đèn báo hộp đúng khối lượng
000.06	Piston C đẩy ra		
000.07	Piston D thực vào		

000.08	Piston D đẩy ra		
000.09	Đếm hộp ra (đúng khối lượng)		
000.10	Có hộp đang chờ được cân		
000.11	Có hộp thiếu khối lượng		
000.13	Có hộp quá khối lượng		
000.14	Nút STOP		

Chương trình điều khiển chức năng cân – tách được trình bày trên hình 4.57a và 4.57b.



Hình 4.57a. Chương trình điều khiển chức năng cân – tách máy nạp và đóng hộp.



Hình 4.57b. Chương trình điều khiển chức năng cân – tách mảy nẹp và đóng hộp.

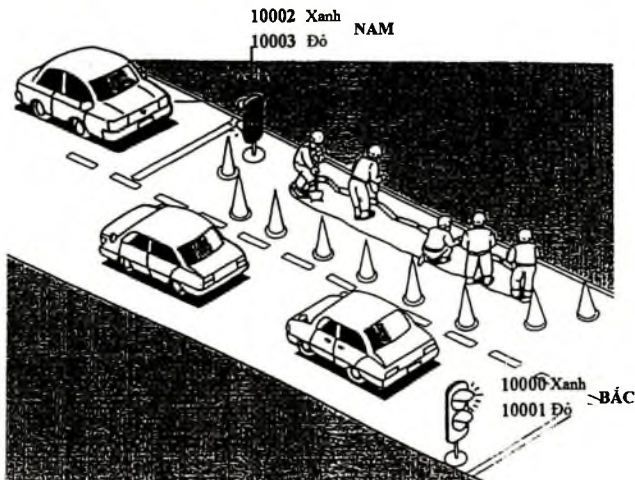
PHỤ LỤC

PHỤ LỤC 1. MỘT SỐ VÍ DỤ ỨNG DỤNG [5]

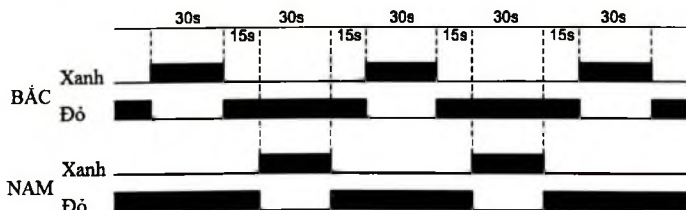
1. Điều khiển đèn giao thông

<Yêu cầu công nghệ>

Do sửa đường, yêu cầu các phương tiện đi ngược chiều theo trình tự luân phiên, với gián đoạn thời gian như sau:

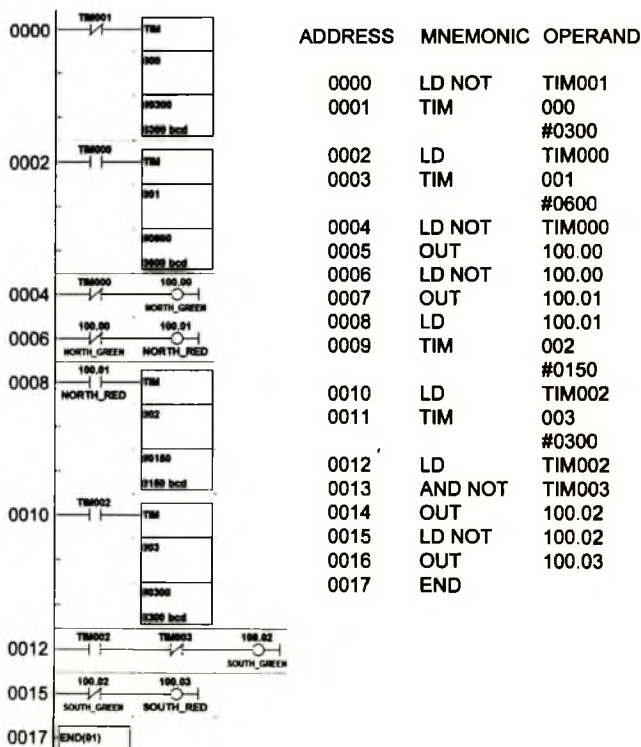


<Gián đoạn thời gian>



Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
		100.00	Đèn xanh của dòng xe phía Bắc
		100.01	Đèn đỏ của dòng xe phía Bắc
		100.02	Đèn xanh của dòng xe phía Nam
		100.03	Đèn đỏ của dòng xe phía Nam

<Chương trình điều khiển dạng LAD và STL>

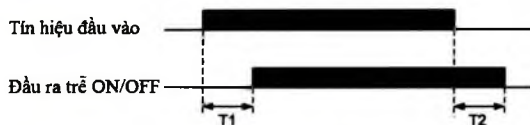


2. Mạch trễ ON/OFF

<Yêu cầu công nghệ>

Lập trình cho động cơ khởi động và dừng theo gián đồ thời gian yêu cầu sau:

<Giản đồ thời gian>

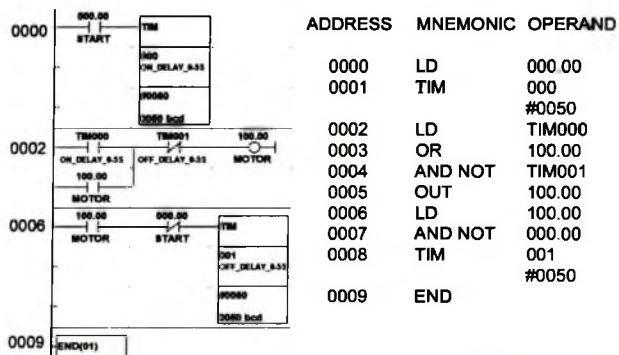


T1: Thời gian trễ ON

T2: Thời gian trễ OFF

Địa chỉ hóa các đầu vào			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Công tắc	100.00	Động cơ

<Chương trình điều khiển dạng LAD và STL>



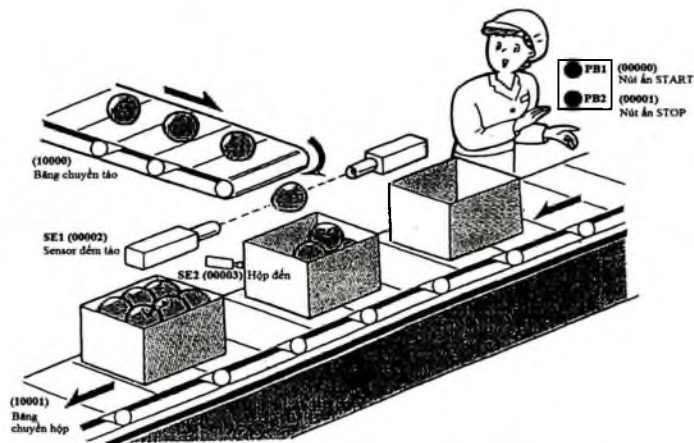
3. Đếm tới 20000

Địa chỉ hóa các đầu vào			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Bắt đầu đếm	100.00	Đếm hoàn thành

5. Điều khiển dây chuyền đóng gói sản phẩm

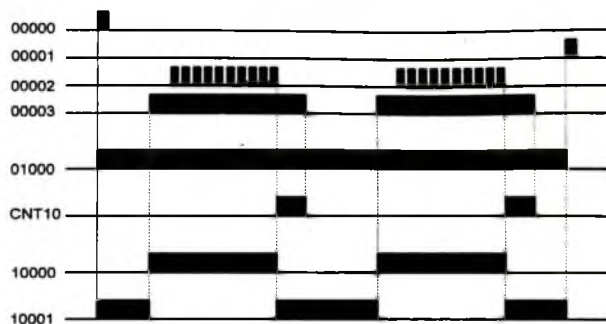
<Yêu cầu công nghệ>

Nhấn nút START (PB1), băng tải sản phẩm chạy đến vị trí SE2 thì dừng lại, băng tải sản phẩm chạy để đưa sản phẩm vào hộp. Sensor SE1 đếm số sản phẩm vào hộp. Khi số sản phẩm trong hộp là 10 thì băng tải sản phẩm dừng, băng tải hộp chạy tiếp đưa hộp đủ sản phẩm ra ngoài và đưa hộp rỗng vào. Nhấn nút STOP (PB2) thì cả hệ thống dừng.

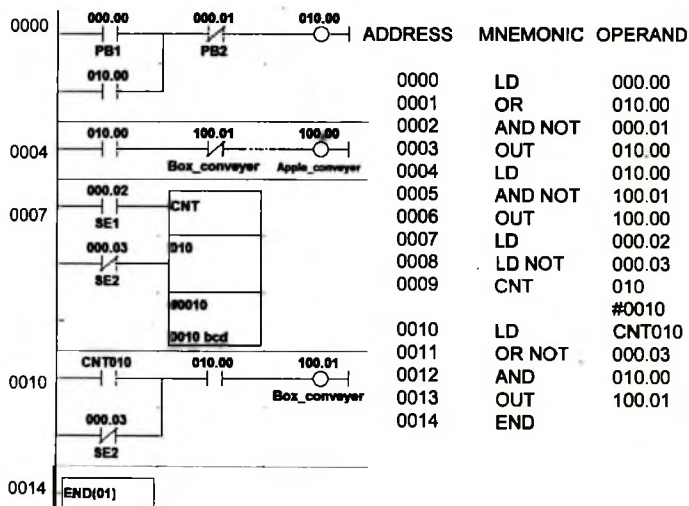


Địa chỉ hóa các đầu vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Nút ấn START PB1	100.00	Băng chuyền sản phẩm
000.01	Nút ấn STOP PB2	100.01	Băng chuyền hộp
000.02	Cảm biến đếm sản phẩm SE1		
000.03	Cảm biến vị trí hộp SE2		

<Giản đồ thời gian>



<Chương trình điều khiển dạng LAD và STL>

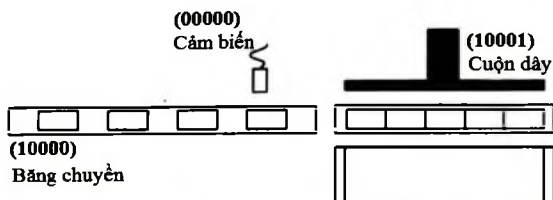


6. Mạch điều khiển cho máy đóng bao

<Yêu cầu công nghệ>

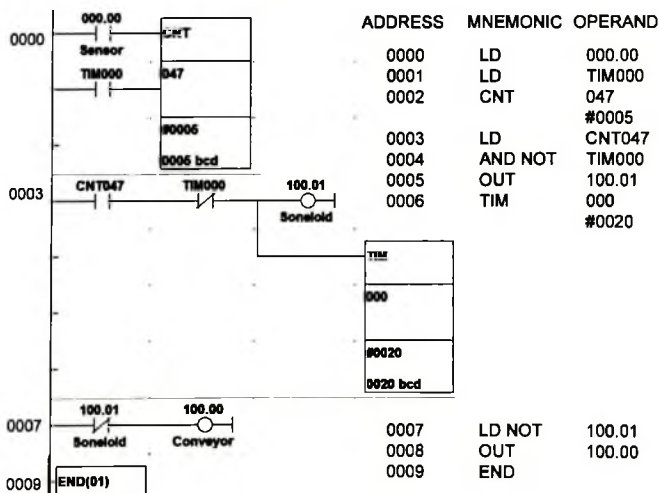
Mạch điều khiển dùng để phát hiện và đếm số sản phẩm đang ở trên dây chuyền lắp ráp. Khi đếm đủ 5 sản phẩm thì băng tải dừng, mạch kích điện cho van đóng bao

trong vòng 2 giây. Sau khi đóng xong, cơ cấu đóng trở về vị trí ban đầu và băng tải tiếp tục hoạt động.



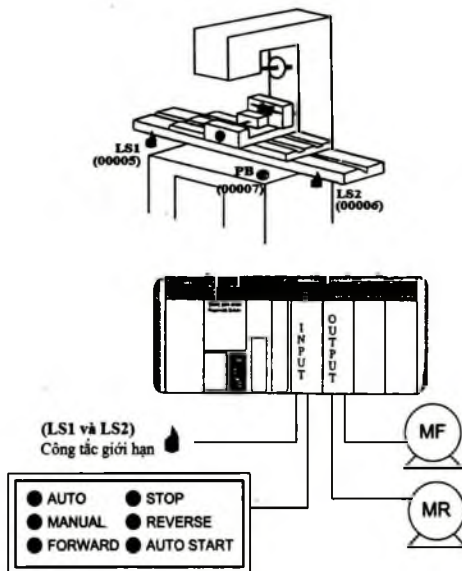
Định nghĩa các đầu vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến báo sản phẩm vào bao	100.00	Băng tải
		100.01	Van đóng bao

<Chương trình điều khiển dạng LAD và STL>



7. Điều khiển máy khoan

<Yêu cầu công nghệ>



Máy có hai chế độ vận hành:

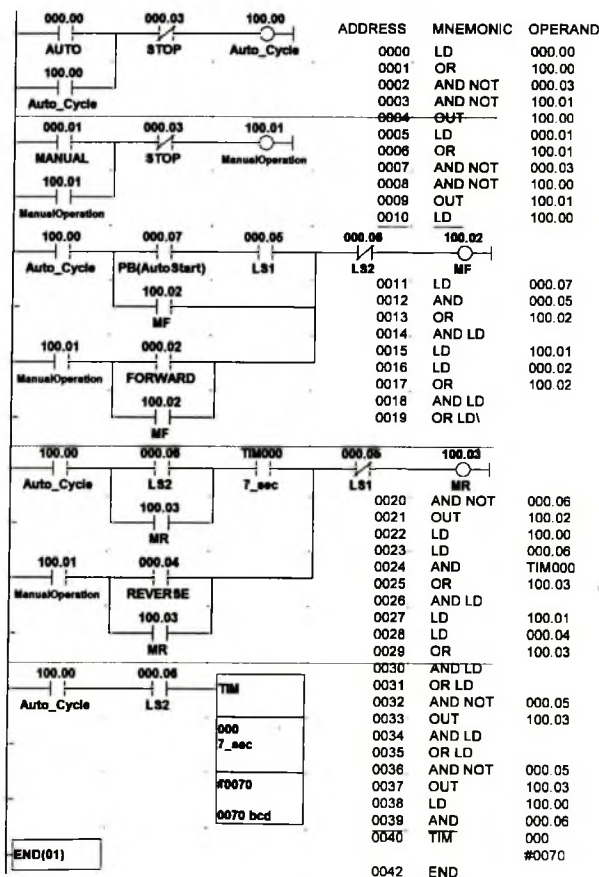
a) Vận hành tay (Manual): Khi bật SW1, động cơ chạy thuận đến khi chạm LS2. Khi bật SW3, động cơ chạy ngược đến khi chạm LS1. Dừng động cơ bằng nút SW2.

b) Chu trình chạy tự động (Auto): Ấn nút PB, nếu LS1 đóng thì động cơ chạy thuận cho đến khi chạm LS2, sau 7 giây, động cơ đảo chiều và quay ngược về đến khi chạm LS1 thì dừng hẳn.

Địa chỉ hóa các đầu vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Nút ấn AUTO	100.00	Đèn Auto Cycle
000.01	Nút ấn MANUAL	100.01	Đèn Manual Operation
000.02	Nút ấn FORWARD SW1	100.02	Động cơ tiến MF
000.03	Nút ấn STOP SW2	100.03	Động cơ lùi MR

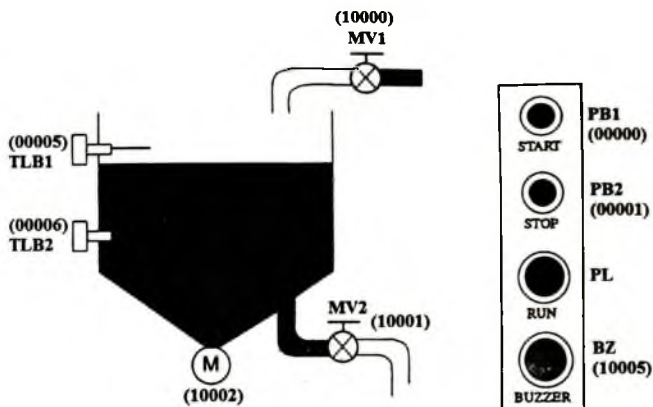
000.04	Nút ấn REVERSE SW3		
000.05	Công tắc giới hạn LS1		
000.06	Công tắc giới hạn LS2		
000.07	Nút ấn tự động khởi động PB		

<Chương trình điều khiển dạng LAD và STL>



8. Điều khiển quá trình rót/tháo

<Yêu cầu công nghệ>



a) Khi ấn nút **START** (PB1), van **MV1** mở ra cấp nước vào thùng và động cơ khuấy **M** bắt đầu chạy.

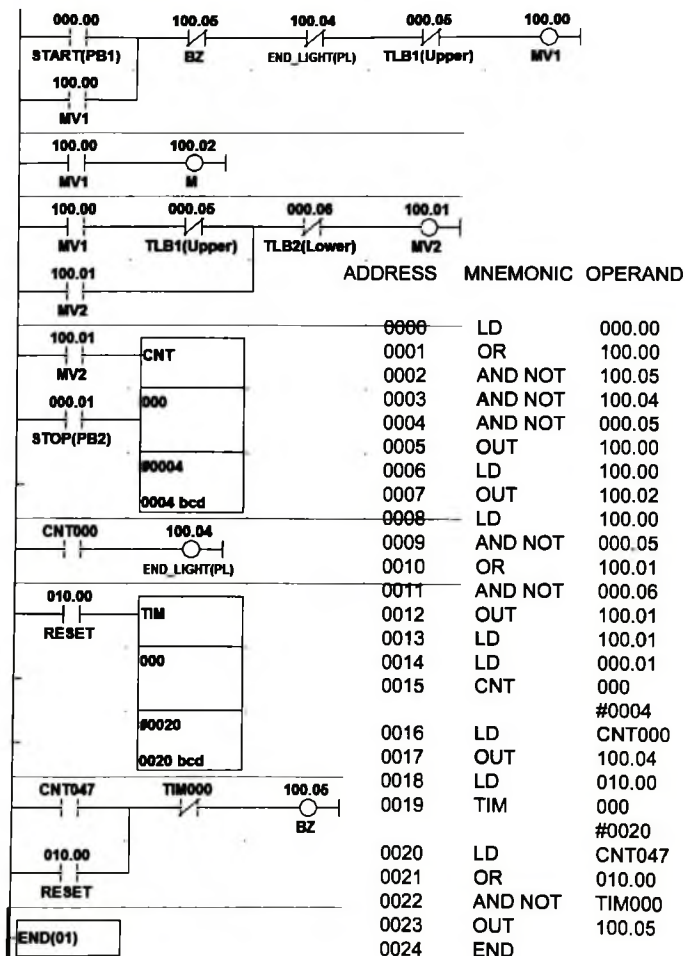
b) Khi mức nước vượt quá sensor báo mức nước thấp **TLB2** và đạt sensor báo mức nước cao **TLB1** thì van **MV1** đóng và động cơ **M** ngừng khuấy.

c) Khi ấn **START** một lần nữa, van **MV2** mở ra và tháo nước, khi nước rút xuống mức nước thấp **TLB2**, van **MV2** đóng lại.

d) Khi cả chu trình trên lặp lại đủ bốn lần thì đèn **END** sáng, còi **BZ** kêu báo quá trình rót tháo kết thúc. Hệ thống không khởi động lại ngay cả khi ấn **PB1**.

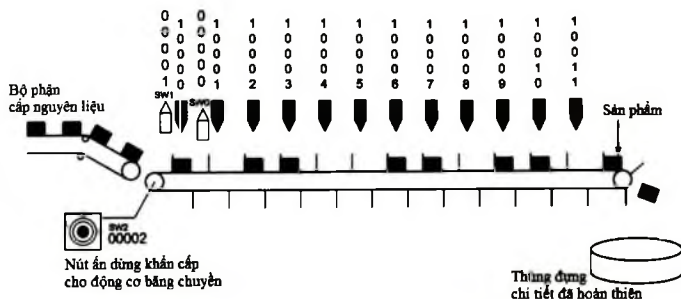
Bảng chỉ định các đầu vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Nút ấn START PB1	100.00	Van cấp MV1
000.01	Nút ấn STOP PB2	100.01	Van xả MV2
000.02	Công tắc mức cao TLB1	100.02	Động cơ khuấy M
000.03	Công tắc mức thấp TLB2	100.04	Đèn báo kết thúc
		100.05	Còi báo hoàn thành bốn chu trình

<Chương trình điều khiển dạng LAD và STL>



9. Điều khiển băng tải

<Yêu cầu công nghệ>



Trên hình vẽ là một băng chuyền gia công các chi tiết qua 12 trạm, các chi tiết được cấp lên băng chuyền và dừng lại trước các trạm gia công một cách đều đặn. Đầu băng chuyền có một cảm biến quang điện SW0, chỉ có chi tiết được phát hiện bởi SW0 thì mới được gia công lần lượt qua 12 trạm. Ngoài ra có một cảm biến khác SW1 xác định vị trí của các chi tiết trên các trạm. Nút STOP dùng để dừng băng chuyền khẩn cấp.

Địa chỉ nhớ các đầu vào			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến phát hiện chi tiết SW0	100.00	Các trạm gia công
000.01	Cảm biến sẵn sàng SW1	...	
000.02	Nút ấn dừng khẩn cấp SW2	100.11	

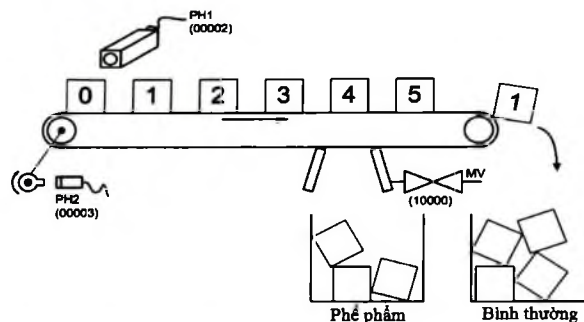
<Chương trình điều khiển dạng LAD và STL>

ADDRESS	MNEMONIC	OPERAND
0000	LD	000.00
0001	LD	000.01
0002	LD	000.02
0003	SFT	100
		100
0004	END	

0000	000.00	SFT(10)
	SW0	
	000.01	100
	SW1	STATIONS
	000.02	100
	SW2	STATIONS
0004	END(01)	

10. Phát hiện và loại bỏ sản phẩm lỗi

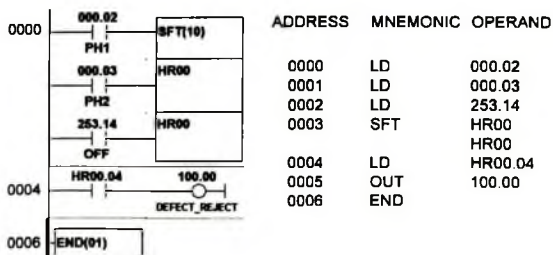
<Yêu cầu công nghệ>



- Cảm biến quang điện PH1 phát hiện sản phẩm lỗi tại đầu vào của băng chuyền.
- Cảm biến quang điện PH2 dùng để phát xung đồng bộ để xác định vị trí của sản phẩm.
- Khi sản phẩm lỗi (đã được PH1 ghi lại) qua vị trí của tay gạt MV thì lập tức bị gạt xuống thùng phế phẩm.

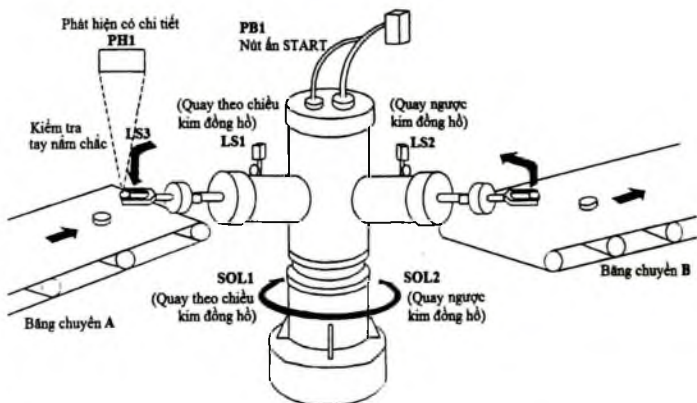
Địa chỉ hóa các đầu vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.02	Cảm biến quang điện PH1	100.00	Tay gạt phế phẩm
000.03	Cảm biến quang điện PH2		

<Chương trình điều khiển dạng LAD và STL>



11. Điều khiển chuyển động của robot

<Yêu cầu công nghệ>



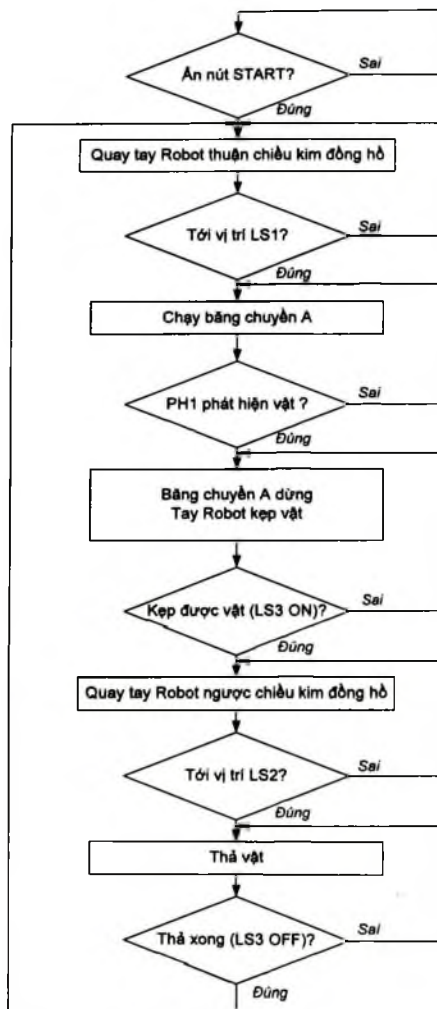
Trên hình vẽ là một dạng Robot điển hình trong công nghiệp, có chức năng di chuyển chi tiết từ băng chuyền A sang băng chuyền B.

a) Khi ấn nút khởi động PB1, tay Robot quay theo chiều kim đồng hồ đến vị trí LS1 có vật trên băng chuyền A và cặp lấy vật.

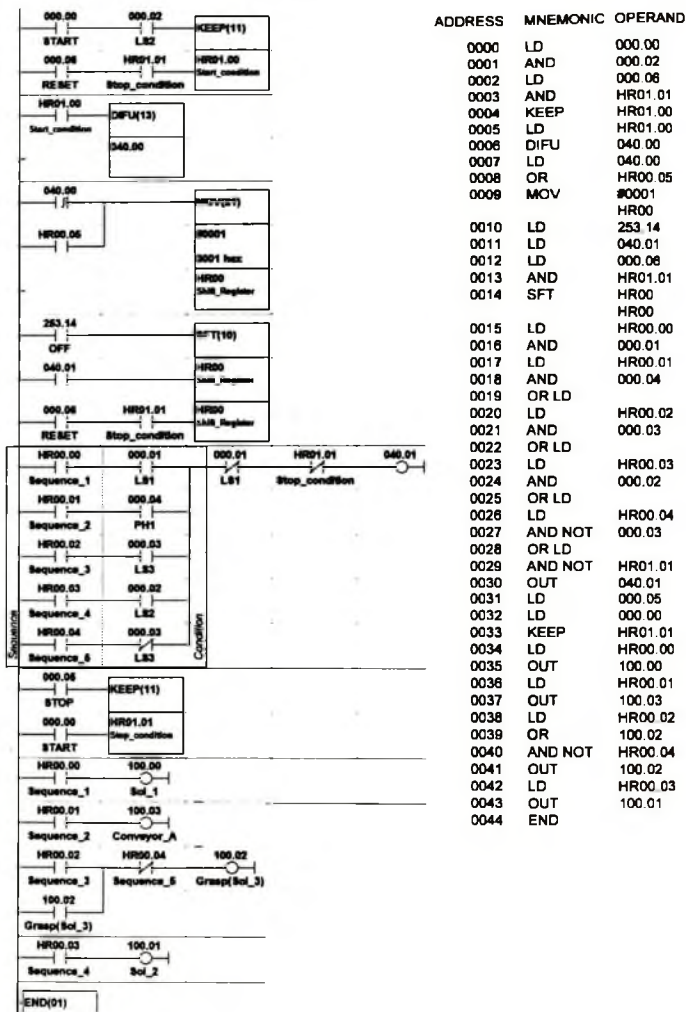
b) Khi kẹp được vật, nó quay tay sang băng chuyền B đến vị trí LS2 và đặt vật lên đó.

Địa chỉ hóa các đầu vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Nút khởi động PB1	100.00	Robot quay thuận SQL1
000.01	Chuyển mạch giới hạn LS1	100.01	Robot quay thuận SQL2
000.02	Chuyển mạch giới hạn LS2	100.02	Kẹp vật
000.03	Chuyển mạch LS3 (tay robot)	100.03	Băng chuyền A
000.04	Cảm biến quang điện PH1		
000.05	Nút STOP		
000.06	Nút RESET		

<Chu trình hoạt động>

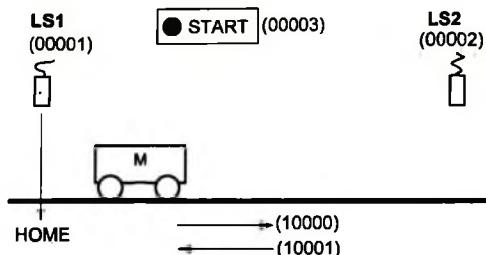


<Chương trình điều khiển dạng LAD và STL>



12. Điều khiển xe tự hành đơn giản

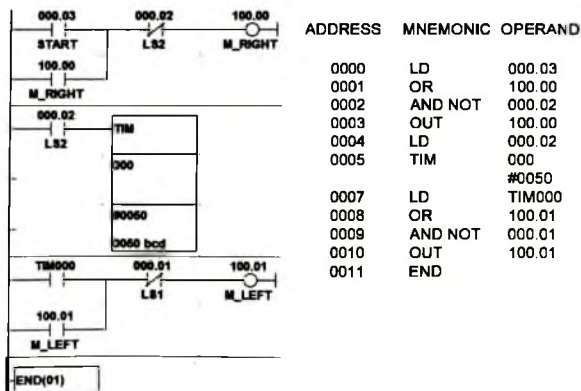
<Yêu cầu công nghệ>



Khi ấn nút START, động cơ M chạy, xe chạy từ trái sang phải cho đến khi tới vị trí LS2 thì dừng lại, tại đây xe dừng lại 5s, sau đó M chạy từ phải sang trái, khi xe về đến LS1 thì chu trình hoàn thành.

Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến LS1(HOME)	100.00	Động cơ M (phải)
000.01	Cảm biến LS2	100.01	Động cơ M (Trái)
000.02	Nút START		

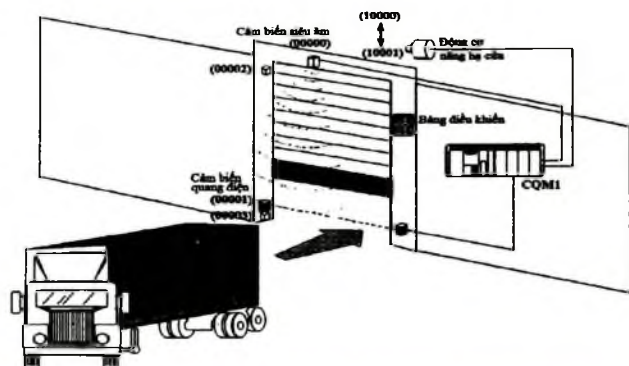
<Chương trình điều khiển dạng LAD và STL>



13. Điều khiển tự động đóng mở cửa kho hàng

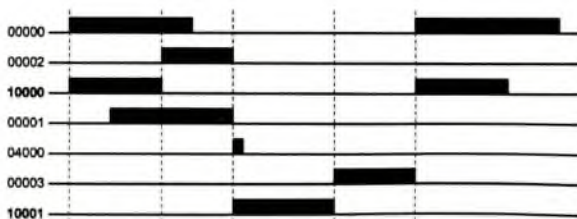
<Yêu cầu công nghệ>

Trên cửa có gắn một cảm biến siêu âm để phát hiện xe chờ hàng đang đến, gắn một cảm biến quang điện khác để phát hiện xe đi qua cửa. Hai chuyển mạch giới hạn để phát hiện giới hạn trên và giới hạn dưới của cửa. Khi có xe đến, cửa tự động mở và tự động đóng sau khi xe đi qua cửa.

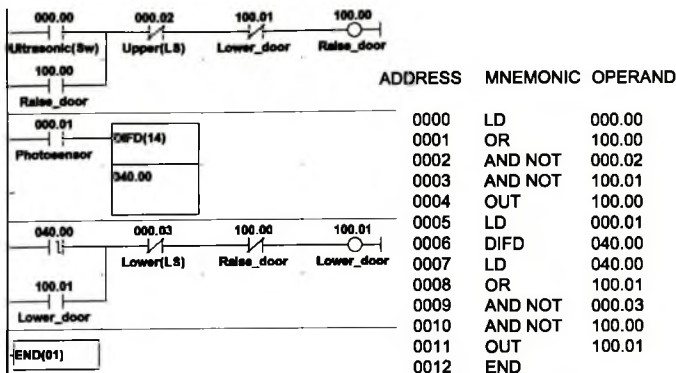


Địa chỉ hóa các đầu vào/ra			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến siêu âm	100.00	Động cơ kéo cửa lên
000.01	Cảm biến quang	100.01	Động cơ hạ cửa xuống
000.02	Công tắc giới hạn trên của cửa		
000.03	Công tắc giới hạn dưới của cửa		

<Giản đồ thời gian>

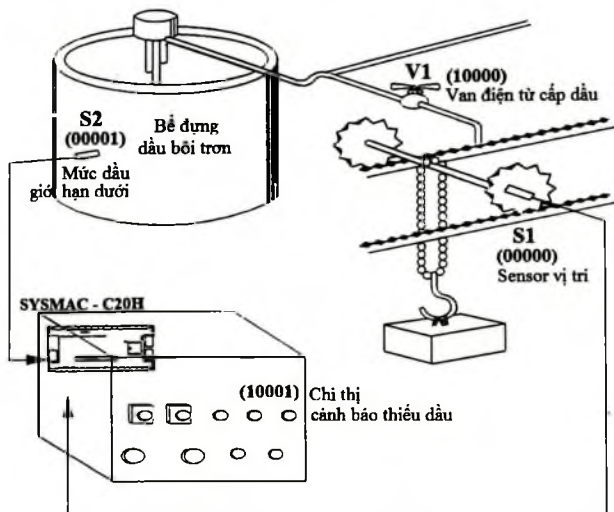


<Chương trình điều khiển dạng LAD và STL>



14. Điều khiển tự động cấp dầu bôi trơn

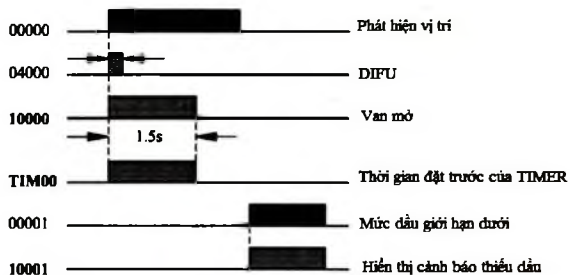
<Yêu cầu công nghệ>



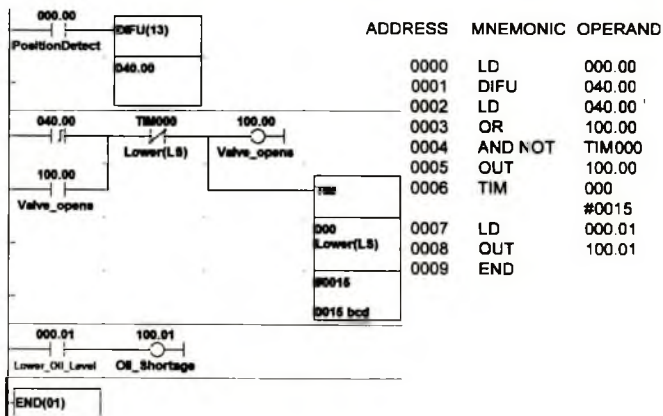
- a) Khi bánh răng chạy qua vị trí của cảm biến vị trí S1, van điện từ V1 sẽ mở và cấp dầu trong khoảng thời gian 1,5s, sau đó đóng lại.
- b) Khi mức dầu trong bể xuống thấp tới vị trí S2, đèn chỉ thị thiếu dầu sáng.

Đèn chỉ thị thiếu dầu			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến vị trí (bánh răng) S1	100.00	Van điện từ cấp dầu V1
000.01	Cảm biến báo mức dầu thấp (S2)	100.01	Đèn chỉ thị thiếu dầu

<Giản đồ thời gian>

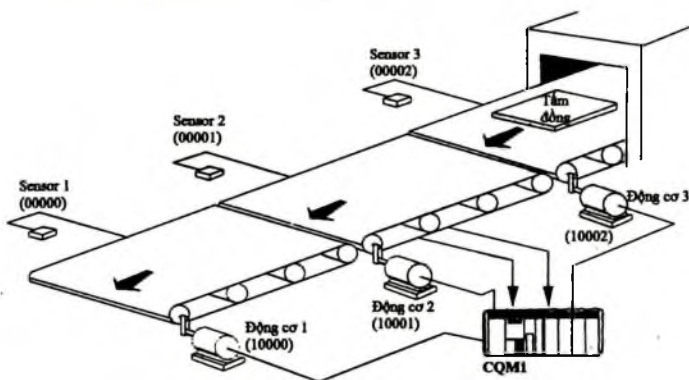


<Chương trình điều khiển dạng LAD và STL>



15. Điều khiển động cơ quay băng tải

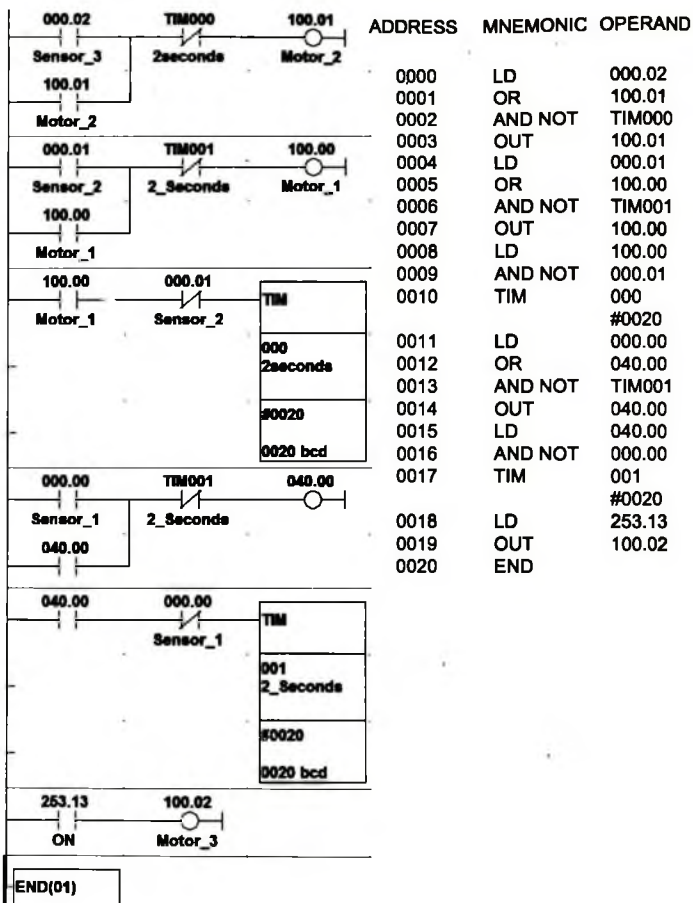
<Yêu cầu công nghệ>



Băng tải gồm nhiều đoạn dùng để vận chuyển tấm kim loại theo hướng như hình vẽ. Đoạn băng tải chỉ được khởi động khi mang tấm kim loại. Các sensor gắn bên cạnh các đoạn băng tải sẽ phát hiện tấm kim loại trên đoạn băng tải. Để đảm bảo chắc chắn tấm kim loại đã đi qua hết mỗi đoạn băng tải thì động cơ chỉ dừng lại sau 2s kể từ khi tấm kim loại ra khỏi đoạn băng tải.

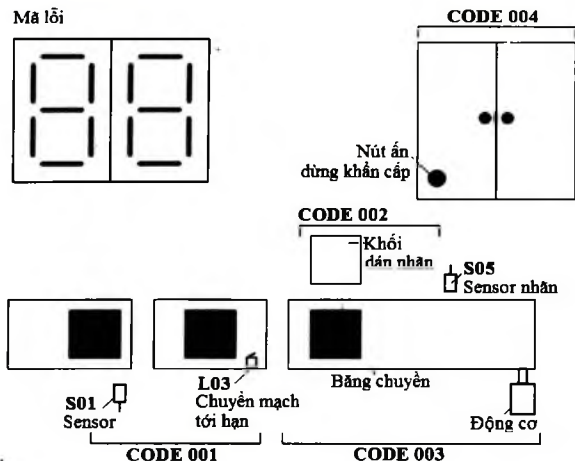
Địa chỉ các đầu vào			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến của BC1	100.00	Động cơ BC1
000.01	Cảm biến của BC2	100.01	Động cơ BC2
000.02	Cảm biến của BC3	100.02	Động cơ BC3

<Chương trình điều khiển dạng LAD và STL>



16. Hiện thị mã lỗi

<Yêu cầu công nghệ>

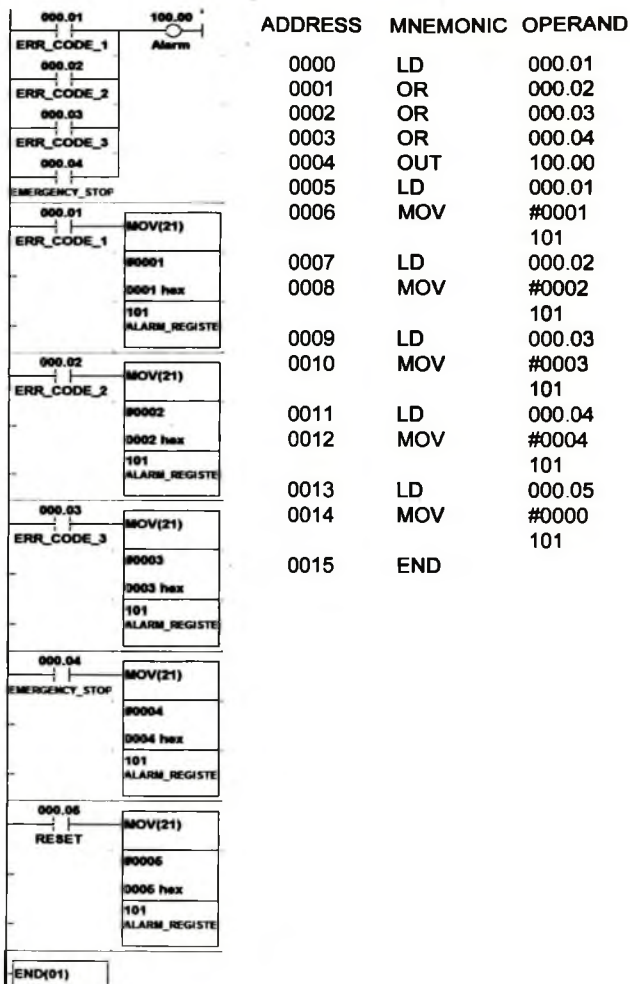


a) Các cảm biến sẽ phát hiện lỗi và đưa tín hiệu về 4 đầu vào, lỗi được hiển thị dưới dạng mã lỗi bằng LED 7 thanh.

b) Nút RESET dùng để xóa hiển thị báo lỗi sau khi khắc phục lỗi.

Danh sách báo lỗi của máy bơm			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.01	Lỗi vùng cấp Lỗi cảm biến S01 Lỗi công tắc giới hạn LS3	100.00	Báo ALARM
000.02	Lỗi ở bộ phận nhân Lỗi công tắc tơ	101	Thanh ghi mã báo động
000.03	Kẹt băng chuyển Kiểm tra cảm biến nhân		
000.04	Dừng khẩn cấp Kiểm tra nút dừng khẩn cấp		
000.05	RESET		

<Chương trình điều khiển dạng LAD và STL>



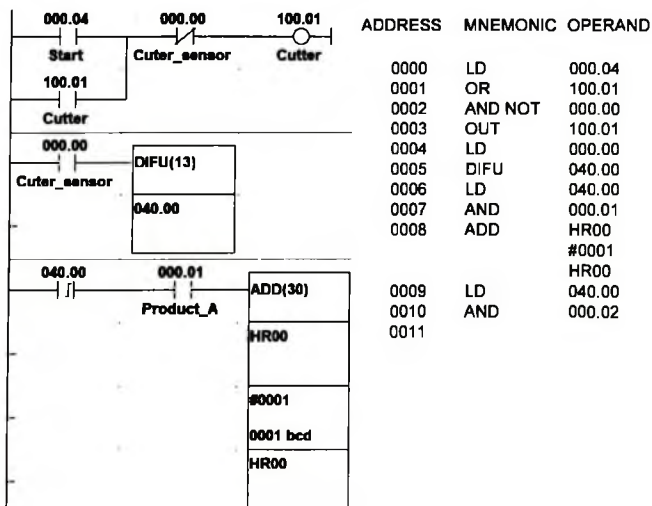
17. Đo tuổi thọ dao cắt

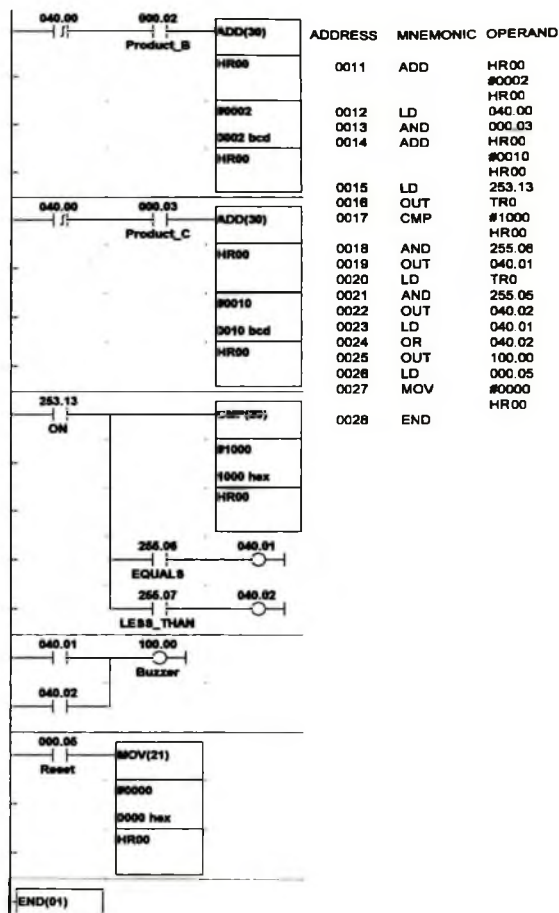
<Yêu cầu công nghệ>

Dao cắt dùng để cắt ba loại sản phẩm, phải thay dao sau 1000 lần cắt sản phẩm A hoặc 500 lần cắt sản phẩm B hoặc 100 lần cắt sản phẩm C. Các sản phẩm này được đưa đến dao cắt một cách ngẫu nhiên và được phát hiện nhờ ba cảm biến khác nhau. Khi phải thay dao thì còi báo. Khi thay dao mới xong thì nhấn RESET.

Điều khiển dao cắt dựa vào PLC			
Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến cắt	100.00	Còi báo thay dao
000.01	Cảm biến phát hiện sản phẩm A	100.01	Dao cắt
000.02	Cảm biến phát hiện sản phẩm B		
000.03	Cảm biến phát hiện sản phẩm C		
000.04	Nút START		
000.05	Nút RESET		

<Chương trình điều khiển dạng LAD và STL>

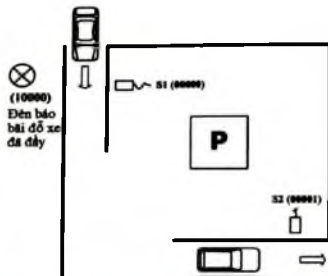




18. Điều khiển bãi đỗ xe

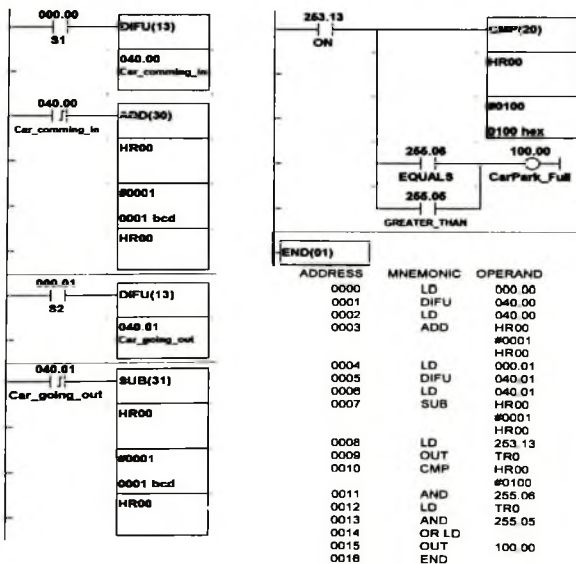
<Yêu cầu công nghệ>

Hệ thống điều khiển bãi đỗ xe cho phép đỗ tối đa 100 xe. Cảm biến S1 đếm xe vào bãi, S2 đếm xe ra. Khi số xe trong bãi vượt quá 100 xe, đèn báo sẽ sáng.



Địa chỉ	Tên thiết bị đầu vào	Địa chỉ	Tên thiết bị đầu ra
000.00	Cảm biến S1	100.00	Đèn báo bãi xe đầy
000.01	Cảm biến S2		

<Chương trình điều khiển dạng LAD và STL>



PHỤ LỤC 2. TẬP LỆNH

1. Tập lệnh của PLC dòng C (OMRON) [4]

Ladder Instructions

AND	And
AND LD	And Load
AND NOT	And Not
LD	Load
LD NOT	Load Not
OR	Or
OR LD	Or Load
OR NOT	Or Not
OUT	Output
OUT NOT	Output Not

Program Control Instructions

END	(01)	End
IL	(02)	Interlock
ILC	(03)	Interlock Clear
JME	(05)	Jump End
JMP	(04)	Jump
NOP	(00)	No Operation
STEP	(08)	Step Define
SNXT	(09)	Step Start
STOP	(99)	Run Stop

Bit Control Instructions

DIFU	(13)	Differentiate Up
DIFD	(14)	Differentiate Down
KEEP	(11)	Keep
RSET		Reset
SET		Set

Timer/Counter Instructions

CNT		Counter
CNTR	(12)	Reversible Counter
CTBL	(--)	Register Comparison Table
HDM	(61)	High-Speed Counter
INI	(--)	Mode Control
PRV	(--)	High-Speed Counter PV Read
RDM	(60)	Reversible Drum Counter
STIM	(--)	Interval Timer
TIM		Timer
TIML	(--)	Long Timer
TIMH	(15)	High-Speed Timer
TMHH	(--)	Very High-Speed Timer
TTIM	(87)	Totalizing Timer

Subroutine Instructions

CMCR	(--)	PCMCIA Card Macro
MCRO	(99)	Macro
PMCR	(--)	Protocol Macro
RET	(93)	Return
SBN	(92)	Subroutine Define
SBS	(91)	Subroutine Entry

Data Shifting Instructions

ASFT	(--)	Asynchronous Shift Register
ASL	(25)	Arithmetic Shift Left
ASR	(26)	Arithmetic Shift Right

ROL	(27)	Rotate Left
ROR	(28)	Rotate Right
SFT	(10)	Shift Register
SFTR	(84)	Reversible Shift Register
SLD	(74)	One Digit Shift Left
SRD	(75)	One Digit Shift Right
WSFT	(16)	Word Shift
RWS	(17)	Reversible Word Shift

Data Movement Instructions

BSET	(71)	Block Set
BOXF2	(--)	Block Transfer to Other EM Bank
COLL	(81)	Data Collect
DIST	(80)	Single Word Distribute
IEMS	(--)	Set EM Indirect Number
MOV	(21)	Move
MOVB	(82)	Move Bit
MOVD	(83)	Move Digit
MVN	(22)	Move Not
XCHG	(73)	Data Exchange
XDMR	(--)	Expansion DM Read
XFER	(70)	Block Transfer
XFRB	(--)	Transfer Bits
XFR2	(--)	Block Transfer by Constant Value

Data Comparison Instructions

BCMP	(68)	Block Compare
CMP	(20)	Compare
CMPL	(60)	Double Compare
CPS	(--)	Signed Binary Compare
CPSL	(--)	Double Signed Binary Compare
MCMP	(19)	Multi-Word Compare
SRCH	(--)	Data Search
TCMP	(85)	Table Compare

ZCP	(--)	Area Range Compare
ZCPL	(--)	Double Area Range Compare

Data Conversion Instructions

ASC	(86)	ASCII Convert
BCD	(24)	Binary to BCD
BCDL	(59)	Double Binary to Double BCD
BCNT	(67)	Bit Counter
BIN	(23)	BCD to Binary
BINL	(58)	Double BCD to Double Binary
COLM	(--)	Line to Column
CTW	(63)	Column to Word
DMPX	(77)	16-to-4 Encoder
HEX	(69)	ASCII to Hexadecimal
HEX	(--)	ASCII to Hexadecimal
LINE	(--)	Line
MLPX	(76)	4-to-16 Decoder
NEG	(--)	2's Complement
NEGL	(--)	Double 2's Complement
SCL	(--)	Scaling
SCL2	(--)	Signed Binary to BCD Scaling
SCL3	(--)	BCD to Signed Binary Scaling
SDEC	(78)	7-Segment Decoder
WTC	(64)	Word to Column

BCD Calculation Instructions

ADD	(30)	BCD Add
ADDL	(54)	Double BCD Add
DEC	(39)	Decrement
DIV	(33)	BCD Divide
DIVL	(57)	Double BCD Divide
INC	(38)	Increment

MUL	(32)	BCD Multiply
MULL	(56)	Double BCD Multiply
SUB	(31)	BCD Subtract
SUBL	(55)	Double BCD Subtract

Binary Calculation Instructions

ADB	(50)	Binary Add
ADBL	(--)	Double Binary Add
DBS	(--)	Signed Binary Divide
DBSL	(--)	Double Signed Binary Divide
DVB	(53)	Binary Divide
MBS	(--)	Signed Binary Multiply
MBSL	(--)	Double Signed Binary Multiply
MLB	(52)	Binary Multiply
SBB	(51)	Binary Subtract
SBBL	(--)	Double Binary Subtract

Special Maths Instructions

APR	(--)	Arithmetic Process
AVG	(--)	Average Value
FDIV	(79)	Floating Point Divide
MAX	(--)	Find Maximum
MIN	(--)	Find Minimum
ROOT	(72)	Square Root
SUM	(--)	Sum
VCAL	(69)	Value Calculate

Floating Point Maths Instructions

+F	(454)	Floating-Point Add
-F	(455)	Floating-Point Subtract
*F	(456)	Floating-Point Multiply
/F	(457)	Floating-Point Divide
ACOS	(464)	Cosine to Angle
ASIN	(463)	Sine to Angle
ATAN	(465)	Tangent to Angle

COS	(461)	Cosine
DEG	(459)	Radians to Degrees
EXP	(467)	Exponent
FIX	(450)	Floating to 16-Bit
FIXL	(451)	Floating to 32-Bit
FLT	(452)	16-Bit to Floating
FLTL	(453)	32-Bit to Floating
LOG	(468)	Logarithm
RAD	(458)	Degrees to Radians
SIN	(460)	Sine
SQRT	(466)	Square Root
TAN	(462)	Tangent

Symbol Maths Instructions

+(400)	Signed Binary Add Without Carry
+L (401)	Double Signed Binary Add Without Carry
+C (402)	Signed Binary Add With Carry
+CL (403)	Double Signed Binary Add With Carry
+B (404)	BCD Add Without Carry
+BL (405)	Double BCD Add Without Carry
+BC (406)	BCD Add With Carry
+BCL (407)	Double BCD Add With Carry
- (410)	Signed Binary Subtract Without Carry
-L (411)	Double Signed Binary Subtract Without Carry
-C (412)	Signed Binary Subtract With Carry
-CL (413)	Double Signed Binary Subtract With Carry
-B (414)	BCD Subtract Without Carry

-BL	(415)	Double BCD Subtract Without Carry
-BC	(416)	BCD Subtract With Carry
-BCL	(417)	Double BCD Subtract With Carry
*	(420)	Signed Binary Multiply
*L	(421)	Double Signed Binary Multiply
*U	(422)	Unsigned Binary Multiply
*UL	(423)	Double Unsigned Binary Multiply
*B	(424)	BCD Multiply
*BL	(425)	Double BCD Multiply
/	(430)	Signed Binary Divide
/L	(431)	Double Signed Binary Divide
/U	(432)	Unsigned Binary Divide
/UL	(433)	Double Unsigned Binary Divide
/B	(434)	BCD Divide
/BL	(435)	Double BCD Divide

Time Instructions

HMS	(--)	Seconds to Hours
SEC	(--)	Hours to Seconds
HTS	(65)	Hours to Seconds
STH	(66)	Seconds to Hours

Logic Instructions

ANDW	(34)	Logical AND
COM	(29)	Complement
ORW	(35)	Logical OR
XNRW	(37)	Exclusive NOR
XORW	(36)	Exclusive OR

Flag/Register Instructions

CLC	(41)	Clear Carry
STC	(40)	Set Carry

Advanced I/O Instructions

7SEG	(--)	7-Segment Display Output
DSW	(--)	Digital Switch
HKY	(--)	Hexadecimal Key Input
MTR	(--)	Matrix Input
TKY	(--)	Ten Key Input

Pulse Instructions

ACC	(--)	Acceleration Control
PLS2	(--)	Pulse Output
PULS	(--)	Set Pulses
PWM	(--)	Pulse With Variable Duty Ratio
SPED	(--)	Speed Output
SYNC	(--)	Synchronised Pulse Control

PID Instructions

PID	(--)	PID Control
-----	------	-------------

Serial Communications Instructions

FCS	(--)	FCS Calculate
KEY	(62)	Key Input
LMSG	(47)	Long Message
MSG	(46)	Message Display
PIN	(64)	RS-232C Port Input
POUT	(63)	RS-232C Port Output
RXD	(--)	Receive
STUP	(--)	Setup
TERM	(48)	Terminal Mode
TXD	(--)	Transmit

Network Instructions

CMND	(194)	Deliver Command
FILP	(44)	External Program Read
FILR	(42)	File Memory Read
FILW	(43)	File Memory Write
RECV	(98)	Network Receive
SEND	(90)	Network Send

ID Communications

IDAR	(63)	DC Autoread
IDAW	(64)	DC Autowrite
IDCA	(65)	DC Clear
IDMD	(66)	DC Manage Data
IDRD	(61)	DC Read
IDWT	(62)	DC Write

Interrupt Control Instructions

INT	(89)	Interrupt Control
-----	------	-------------------

I/O Instructions

IORD	(-)	I/O Read
IORF	(97)	I/O Refresh
IOWR	(-)	I/O Write
MPRF	(-)	Group-2 High-Density

I/O Refresh

READ	(88)	Intelligent I/O Read
WRIT	(87)	Intelligent I/O Write

Error and Diagnostics Instructions

FAL	(06)	Failure Alarm
FALS	(07)	Severe Failure Alarm
FPD	(-)	Failure Point Detect
TRSM	(45)	Data Tracing

System Control Instructions

ENDW	(62)	End Wait
SCAN	(18)	Scan Time
SYS	(49)	Set System
WDT	(94)	Watchdog Timer
Refresh		

Block Programming Instructions

BEND	<01>	Block Program End
BPPS	<11>	Block Program Pause
BPRG	(96)	Block Program Begin
BPRS	<12>	Block Program Restart
CNTW	<14>	Counter Wait
ELSE	<03>	Else
EXIT	<06>	Conditional Block Exit
IEND	<04>	Block Branching End
IF	<02>	Block Branching If
LEND	<10>	Block Loop Control End
LOOP	<09>	Block Loop Control
RSET	<08>	Reset
SET	<07>	Set
TIMW	<13>	Timer Wait
TMHW	<15>	High-Speed Timer Wait
WAIT	<05>	One Scan and Wait

Comment Instructions

NETW	(63)	Notation Insert
------	------	-----------------

2. Tập lệnh của PLC SLC 500 (ALLEN BRADLEY) [3]

Instruction	Description	Instruction Parameter	Valid Addressing Mode(s)	Valid File Types
ABL(1)	ASCII Test Buffer for Line	channel		
		control	direct	R
		characters		
ABS(2)	Absolute Value	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
ACB(1)	ASCII Number of Characters in Buffer	channel		
		control	direct	R
		characters		
ACI(1)	ASCII String to Integer	source	direct, indirect	ST
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
ACL(1)	ASCII Clear Buffer	channel		
		transmit buffer		
		receive buffer		
ACN(1)	ASCII String Concatenate	source A	direct, indirect	ST
		source B	direct, indirect	ST
		destination	direct	ST

ACS(2)	Arc Cosine	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
ADD	Add	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
AEX(1)	ASCII String Extract	source	direct, indirect	ST
		index	direct, indexed direct	O, I, S, B, T, C, R, N,
			indirect, indexed indirect	A, ST, M
		number	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct	ST
AHL(1)	ASCII Set/Reset Handshake Lines	channel		
		AND mask	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		OR mask	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		control	direct	R
		channel status		
AIC(1)	ASCII Integer to String	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct	ST

AND	Logical AND	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
ARD(1)	ASCII Read Characters	channel		
		destination	direct	ST
		control	direct	R
		string length		
		characters read		
ARL(1)	ASCII Read Line	channel		
		destination	direct	ST
		control	direct	R
		string length		
		characters read		
ASC(1)	ASCII String Search	source	direct, indirect	ST
		index	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		search	direct, indirect	ST
		result	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
ASN(2)	Arc Sine	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M

		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
ASR(1)	ASCII String Compare	source A	direct, indirect	ST
		source B	direct, indirect	ST
ATN(2)	Arc Tangent	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
AWA(1)	ASCII Write with Append	channel		
		source	direct	ST
		control	direct	R
		string length		
		characters sent		
AWT(1)	ASCII Write	channel		
		source	direct	ST
		control	direct	R
		string length		
		characters sent		
BSL	Bit Shift Left	File	indexed direct indexed indirect	O, I, S, B, N, A, ST
		control	direct	R
		Bit address	direct, indirect	O, I, S, B, T, C, R, N, A, ST, M
		length		
BSR	Bit Shift Right	File	indexed direct indexed indirect	O, I, S, B, N, A, ST

		control	direct	R
		Bit address	direct, indirect	O, I, S, B, T, C, R, N, A, ST, M
		length		
BTR	Block Transfer Read	control	direct	N
		source "Data"	direct	I, O, S, B, N, A, F
		buffer	direct M1 File address	M
BTW	Block Transfer Write	control	direct	N
		source "Data"	direct	I, O, S, B, N, A, F
		buffer	direct M0 File address	M
CEM	ControlNet Explicit Message	control block	direct	N
		control block length		
		1747-SCNR slot		
		size in Words (receive data)		
		size in Words (send data)		
		data table address (receive data)	direct	N
		data table address (send data)	direct	N
		message timeout (x1 ms)		
		ControlNet address		

		service		
		service code (hex)		
		class (hex)		
		instance (hex)		
		attribute (hex)		
		member (hex)		
CLR	Clear	destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
COP	Copy File	source	indexed direct indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	indexed direct indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		length		
COS(2)	Cosine	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct	O, I, S, B, T, C, R, N,
			indirect, indexed indirect	F, A, ST, M
CPT(2)	Compute	destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		expression		
CTD	Count Down	counter	direct	C
		preset		
		accum		
CTU	Count Up	counter	direct	C
		preset		
		accum		
DCD	Decode 4 to 1 of 16	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M

		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
DDV	Double Divide	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
DDT	Diagnostic Detect	control	direct	R
		source "File"	index direct	I, O, S, B, N, A
		reference "File"	index direct	I, O, S, B, N, A
		result "File"	index direct	I, O, S, B, N, A
DEG(2)	Radians to Degrees	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
DEM	DeviceNet Explicit Message	control block	direct	N
		control block length		
		1747-SDN slot		
		size of send data (bytes)		
		message timeout (x1 sec)		
		DeviceNet		
		address		

		service		
		service code (hex)		
		class (hex)		
		instance (hex)		
		attribute (hex)		
DIV	Divide	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
EEM	EtherNet/IP Explicit Message	control block	direct	N
		control block length		
		size in Words (receive data)		
		size in Words (send data)		
		data table address (receive data)	direct	N
		data table address (send data)	direct	N
		service		
		service code (hex)		

		class (hex)		
		instance (hex)		
		attribute (hex)		
ENC	Encode 1 of 16 to 4	source	direct, indirect, index, indirect index	I, O, S, B, N, A, T
		destination	direct, indirect, index, indirect index	I, O, S, B, N, A, C, R
EQU	Equal	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
FBC	File Bit Comparison	control	direct	R
		source "File"	index direct	I, O, S, B, N, A
		reference "File"	index direct	I, O, S, B, N, A
		result "File"	index direct	I, O, S, B, N, A
FFL(3)	FIFO Load	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M(4)
		FIFO array	indexed direct indexed indirect	O, I, S, B, N, A
		FIFO control	direct	R
		length		
		position		
FFU(3)	FIFO Unload	FIFO array	indexed direct indexed indirect	O, I, S, B, N, A
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M(4)
		FIFO control	direct	R

		length		
		position		
FLL	Fill File	source	direct, indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	indexed direct indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		length		
FRD	From BCD to Binary	source (SLC 5/01)	direct	O, I, S, B, T, C, R, N, A, ST, M
		source (SLC 5/02, 5/03, 5/04, 5/05)	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
GEQ	Greater Than or Equal	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
GRT	Greater Than	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
HSC(5)	High-speed Counter (SLC 5/01)	counter		
		preset		
		counter	direct	C
		preset		
		accum		

		source	direct	B and N
		length		
IID(6)	I/O Interrupt Disable	slot		
IIE(6)	I/O Interrupt Enable	slot		
IIM	Immediate Input with Mask	slot	direct	I
		mask	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		length (SLC 5/03, 5/04, and 5/05)		
INT(3)	I/O Interrupt			
IOM	Immediate Output with Mask	slot	direct	O
		mask	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		length (SLC 5/03, 5/04, and 5/05)		
IMP	Jump	label number		
JSR	Jump to Subroutine	subroutine File number		
LBL	Label Declaration	label number		
LEQ	Less Than or Equal To	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M

LES	Less Than	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
LFL(3)	LIFO Load	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M(4)
		LIFO array	indexed direct indexed indirect	O, I, S, B, N, A
		LIFO control	direct	R
		length		
		position		
LFU(3)	LIFO Unload	LIFO array	indexed direct indexed indirect	O, I, S, B, N, A
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M(4)
		LIFO control	direct	R
		length		
		position		
LIM(3)	Limit Test (circ)	low limit	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		test	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		high limit	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
LN(2)	Natural Log	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M

		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
LOG(2)	Log to the Base 10	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
MCR	Master Control Relay			
MEQ	Mask Compare Equal To	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		source mask	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		compare	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
MOV	Move	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
MSG (5/02 only)	Message	read/write		
		target device		
		control block	direct	N
		control block length		
		local address	direct	O, I, S, B, T, C, R, N, A
		target node		

		target address	direct	O, I, S, B, T, C, R, N, A
		message length		T, C, R
				I, O, S, B, N
MSG (5/03, 5/04, and 5/05(7))	Message	type		
		read/write		
		target device		
		local/remote		
		control block	direct	N
		control block length		
		channel number		
		target node		
		remote bridge link ID		
		remote bridge node address		
		local bridge node address		
		local File address	direct	O, I, S, B, T, C, R, N, F, A, ST, M(9)
		target File address	direct	O, I, S, B, T, C, R, N, F, A, ST, M(9)
		message length		O, I, B, N, A(9)
				S(9)

				F(9)
				T
				C, R
				ST(9)
		message timeout		
MSG (5/05 Ethernet)	Message	type		
		read/write		
		target device		
		local		
		control block	direct	N
		control block length		
		channel number		
		target node, remote bridge link ID, local and remote bridge node address	not applicable	
		IP address (ww.xx.yy.zz)		
		local File address	direct	O, I, S, B, T, C, R, N, F, A, ST, M

		target File address	direct	O, I, S, B, T, C, R, N, F, A, ST, M
		message timeout		
MUL	Multiply	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
MVM	Masked Move	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		source mask	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
NEG	Negate	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
NEQ	Not Equal To	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
NOT	Logical NOT	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M

		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
OR	Logical OR	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
OSR	One-shot Rising	Bit address	direct, indirect	O, I, S, B, T, C, R, N, A, ST
OTE	Output Energize	Bit address	direct, indirect	O, I, S, B, T, C, R, N, A, ST, M
OTL	Output Latch	Bit address	direct, indirect	O, I, S, B, T, C, R, N, A, ST, M
OTU	Output Unlatch	Bit address	direct, indirect	O, I, S, B, T, C, R, N, A, ST, M
PID(1)	PID	control block	direct	N
		process variable	direct, indirect	O, I, B, T, C, R, N, A
		control variable	direct, indirect	O, I, B, T, C, R, N, A
		control block length		
RAD(2)	Degrees to Radians	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct	O, I, S, B, T, C, R, N,
			indirect, indexed indirect	F, A, ST, M

REF(1)	I/O Refresh	channel 0		
		channel 1		
RES	Timer/Counter Reset	structure	direct	T, C, R
RET	Return			
RHC	Read High Speed Clock	destination	direct	N, F
RMP	Ramp	control	direct	N
		destination	direct	O, I, S, B, N
RPC	Read Program Checksum	Proc/Mmod		
		destination	direct	N
RPI(1)	Reset Pending Interrupt	slots		
RTO	Retentive Timer On	timer	direct	T
		time base (SLC 5/01)		
		time base (SLC 5/02, 5/03, 5/04, 5/05)		
		preset		
		accum		
SBR	Subroutine			
SCL(3)	Scale	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		rate	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M

		offset	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
SCP(2)	Scale with Parameters	input	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		input min.	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		input max.	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		scaled min.	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		scaled max.	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		scaled output	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
SIN(2)	Sine	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
SQC	Sequencer Compare	File	indexed direct indexed indirect	O, I, S, B, N, A, ST
		mask	direct, indexed direct(4) indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M

		source	direct, indexed direct(4) indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		control	direct	R
		length		
		position		
SQL(3)	Sequencer Load	File	indexed direct indexed indirect	O, I, S, B, N, A, ST
		source	direct, indexed direct(4) indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		control	direct	R
		length		
		position		
SQO	Sequencer Output	File	indexed direct indexed indirect	O, I, S, B, N, A, ST
		mask	direct, indexed direct(4) indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct, indexed direct(4) indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		control	direct	R
		length		
		position		
SQR(3)	Square Root	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct	O, I, S, B, T, C, R, N,
			indirect, indexed indirect	F, A, ST, M
STD(3)	Selectable Timed Interrupt Disable			

STE(3)	Selectable Timed Interrupt Enable			
STS(3)	Selectable Timed Interrupt Start	File	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		time	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
SUB	Subtract	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
SUS	Suspend	suspend ID		
SVC(1)	Service Communications	channel 0 (SLC 5/03, 5/04, 5/05)		
		channel 1 (SLC 5/03, 5/04, 5/05)		
SWP(2)	Swap	source	indexed direct indexed indirect	B, N, A, ST
		length		
TAN(2)	Tangent	source	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M

		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
TDF	Compute Time Difference	start	direct	N, F
		stop	direct	N, F
		destination	direct	N, F
TND	Temporary End			
TOD	Convert to BCD	source (SLC 5/01)	direct	O, I, S, B, T, C, R, N
		source (SLC 5/02,	direct, indexed direct	O, I, S, B, T, C, R, N,
		5/03, 5/04, 5/05)	indirect, indexed indirect	A, ST, M
		destination	direct	O, I, S, B, T, C, R, N, A, ST, M
TOF	Timer Off Delay	timer	direct	T
		time base (SLC 5/01)		
		time base (SLC 5/02, 5/03, 5/04, 5/05)		
		preset		
		accum		
TON	Timer on Delay time base (SLC 5/01) time base	timer	direct	T

	(SLC 5/02, 5/03, 5/04, 5/05) preset accum			
XIC	Examine On (Examine if Closed Contact)	source Bit	direct, indirect	O, I, S, B, T, C, R, N, A, ST, M
XIO	Examine Off (Examine if Open Contact)	source Bit	direct, indirect	O, I, S, B, T, C, R, N, A, ST, M
XOR	Logical Exclusive OR	address A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		address B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, A, ST, M
XPY(2)	X to the Power of Y	source A	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		source B	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M
		destination	direct, indexed direct indirect, indexed indirect	O, I, S, B, T, C, R, N, F, A, ST, M

Function Codes

The function codes for the instructions are listed below in alphabetical order.

Instruction	AND Flag	PLS Code	Notes
ADD		00	Add
AND		13	And Block
AND (Bit Input)	—1—	05	And
AND (Reset Input)		100	And
AND		03	And Inverted
BSL		100	BS Shift Left
BSR		101	BS Shift Right
CLR		00	Clear
COP		104	File Copy
CTD		4	Count Down
CTU		8	Count Up
DDD		102	Divide 1 by 1 or 16
DDV		04	Double Divide
DN		00	Done
ENC		103	Encode 1 of 16 to 4
AND EQU —1 EQU —		01	
LD EQU —1 EQU —		02	Equal
OR EQU —1 EQU —		03	
FR		110	FRQ Load
FRU		114	FRQ Unload
FL		105	FR File
FRD		101	Convert from BCD
AND GED —1 GED —		06	
LD GED —1 GED —		05	Greater Than or Equal
OR GED —1 GED —		07	
AND GRT —1 GRT —		08	
LD GRT —1 GRT —		09	Greater Than
OR GRT —1 GRT —		04	
HBC		119	High-Speed Counter
HBD		114	High-Speed Counter Internal Divide
HBE		115	High-Speed Counter Internal Enable
HBL		111	High-Speed Counter Load
IM		120	Interrupt Input with Mask

Instruction	AND Flag	PLS Code	Notes
INT —1 LD INT —		100	Interrupt Subroutine
ION		110	Interrupt Output with Mask
JMP		100	Jump to Label
JBR		100	Jump to Subroutine
LAL LD LAL —1 LAL —		101	Label
LD		—1—	Load
LDI		—1—	Load Inverted
LDI		—1 LD I —	Load True
AND LEO —1 LEO —		00	
LD LEO —1 LEO —		00	Less Than or Equal
OR LEO —1 LEO —		01	
AND LES —1 LES —		07	
LD LES —1 LES —		06	Less Than
OR LES —1 LES —		08	
AND LIM —1 LIM —		70	
LD LIM —1 LIM —		71	Limit Test
OR LIM —1 LIM —		72	
LPL		110	LPL Load
LPU		110	LPL Unload
MCA		100	Master Control Reset
AND MEO —1 MEO —		00	
LD MEO —1 MEO —		00	Modified Comparison or Equal
OR MEO —1 MEO —		70	
MOV		100	Move
MPP		10	Memory Pop
MPS		00	Memory Push
MRO		11	Memory Read
MUL		00	Multiply
MVM		107	Modified Move
NEG		110	Negate
AND NEG —1 NEG —		04	
LD NEG —1 NEG —		03	Not Equal
OR NEG —1 NEG —		05	

Instruction	AND Flag	PLS Code	Notes
NOT		111	Not
OR (Bit Input)		00	Or
OR (Reset Input)		100	Or
OR		14	Or Mask
OR		00	Or Inverted
ORT		00	Or True
AND ORR —1 ORR —		00	Or-And Rising
LD ORR —1 ORR —		00	
OUT (Reset)	—1—	00	Output
OUT (High-Speed Counter)		00	Update High-Speed Counter Increment
RAC		112	High-Speed Counter Reset Accumulator
RBC (Reset/Counter)		1	Reset
RBC (High-Speed Counter)		7	High-Speed Counter Reset
RET		120	Return from Subroutine
RST	—0—	02	Reset
RTO		0	Retention Time
BSR LD BSR —1 BSR —		100	Retain
RCL		00	Reset Clear
RET	—0—	01	Set
ROC		100	Resequencer Compare
RCL		100	Resequencer Load
ROD		100	Resequencer Output
ROR		00	Rotate Right
STD		100	Subroutine True Interrupt Enable
STE		100	Subroutine True Interrupt Enable
STB		100	Subroutine True Interrupt Enable
SLB		01	Subtract
SLB		100	Subtract
TND		100	Temporary End
TOD		100	Time of Day
TDF		1	Time On-Off
TDR		0	Time On-Off
XOR		110	Exclusive Or



Allen-Bradley

1791-4.2.1 - October 1994

40072-015-01(A)

Copyright 1994 Allen-Bradley Company, Inc. Printed in USA

MỘT SỐ THUẬT NGỮ

Thuật ngữ tiếng Anh

Dip Switch	: Chuyển mạch DIP
Absolute	: Tuyệt đối
Accumulator	: Thanh chứa
Active	: Tích cực
Address	: Địa chỉ
Address Bus	: Bus địa chỉ
Algorithm	: Thuật toán
Analog	: Tương tự
AND function	: Hàm và
AND INHIBIT function	: Hàm cấm và
ASCII	: Mã ASCII
Assembly	: Lắp ráp
Automatic	: Tự động
Backplane	: Bảng mạch BUS
Battery	: Pin
Binary Data	: Dữ liệu nhị phân
Bit	: Bit
Block	: Khối
Buzzer	: Chuông
Capacity	: Dung lượng
Chassis	: Khung đỡ
Check	: Kiểm tra
Clock	: Đồng hồ
Communication	: Truyền thông
Communication Port	: Cổng truyền thông
Component	: Thành phần

Configurable Parameter Space	: Miền tham số cấu hình được
Configuration	: Cấu hình
Connecting	: Kết nối
Contact	: Tiếp điểm, công tắc
Control Bus	: Bus điều khiển
Converter, Adapter	: Bộ chuyển đổi
Converting	: Chuyển đổi
Counter	: Bộ đếm
CPU (Central Processing Unit)	: Khối xử lý trung tâm
CSF (Control System Flow Chart)	: Sơ đồ khối chức năng điều khiển
Current	: Dòng tiêu thụ
Consumption	
Cycle Time	: Chu kỳ (quét)
Data	: Dữ liệu
Data Bus	: Bus dữ liệu
Data File	: File dữ liệu
Data Memory	: Bộ nhớ dữ liệu
Data Structure	: Cấu trúc dữ liệu
Decimal Data	: Dữ liệu thập phân
Default	: Mặc định
Device	: Thiết bị
Digital	: Số
Direct	: Trực tiếp
Disable	: Không cho phép
Discrete	: Rời rạc
Display	: Hiện thị

Down Load	: <i>Nạp chương trình</i>	INHIBITION	: <i>Hàm cấm</i>
Dry Run	: <i>Chạy mô phỏng</i>	function	
Dword (Double Word)	: <i>Từ dữ liệu kép</i>	Input	: <i>Đầu vào</i>
Dynamic RAM	: <i>RAM động</i>	Installation	: <i>Cài đặt</i>
Element	: <i>Phần tử</i>	Instruction	: <i>Lệnh</i>
Enable	: <i>Cho phép</i>	Interface	: <i>Giao diện</i>
EXCLUSIVE OR function	: <i>Hàm cộng module 2</i>	Internal Relay	: <i>Rơ le trong</i>
Expansion Instructions	: <i>Lệnh mở rộng</i>	Interrupt	: <i>Ngắt</i>
Fatal	: <i>Nghiêm trọng (lỗi)</i>	LAD (Ladder Diagram)	: <i>Giản đồ thang</i>
Field Device	: <i>Thiết bị trường</i>	Ladder	: <i>Hình thang</i>
Fixed	: <i>Cố định</i>	Latching	: <i>Chốt</i>
Flag	: <i>Cờ</i>	Leakage Current	: <i>Dòng rò</i>
Flip-Flop	: <i>Bit đầu ra hai trạng thái ổn định</i>	Limit Switch	: <i>Công tắc hành trình</i>
Function Block	: <i>Khối chức năng</i>	Load	: <i>Tải</i>
Function Code	: <i>Mã hàm</i>	Local	: <i>Tại chỗ</i>
Ground	: <i>Nối đất</i>	Logic Network	: <i>Mạch logic</i>
Hexadecimal	: <i>Hệ đếm HEXA</i>	Master	: <i>Trạm chủ</i>
Hexadecimal Data	: <i>Dữ liệu HEXA</i>	Memory	: <i>Bộ nhớ</i>
High Speed	: <i>Tốc độ cao</i>	Memory Area	: <i>Vùng nhớ</i>
HMI (Human Machine Interface)	: <i>Giao diện người - máy</i>	Memory Card	: <i>Thẻ nhớ</i>
I/O	: <i>Vào/ra</i>	Memory Map	: <i>Bản đồ bộ nhớ</i>
I/O Module	: <i>Khối vào/ra</i>	Memory Space	: <i>Không gian bộ nhớ</i>
Image	: <i>Vùng ảnh</i>	Mnemonic	: <i>Mã ngữ</i>
Immediate	: <i>Tức thời</i>	Mode	: <i>Chế độ (hoạt động)</i>
Inactive	: <i>Không tích cực</i>	Modular	: <i>Mô đun hóa</i>
Increment	: <i>Tăng</i>	Monitoring	: <i>Theo dõi, giám sát</i>
Indicator	: <i>Chỉ thị</i>	MONSTABLE	: <i>Bit đầu ra một trạng thái ổn định</i>
Indirect	: <i>Gián tiếp</i>	Multitask	: <i>Đa nhiệm</i>
		NAND function	: <i>Hàm phủ định và</i>
		NC (Normally Closed)	: <i>Thường kín</i>

NO (Normally Openned)	: Thường mở	Programmer	: Công cụ lập trình
NOT function	: Hàm phủ định	Programming	: Lập trình
Non- Volatile	: Duy trì	Programming Console	: Bộ lập trình cầm tay
Non-Fatal	: Không nghiêm trọng (lỗi)	Programming Device	: Thiết bị lập trình
NOR function	: Hàm phủ định hoặc	Project	: Thiết kế
Online Edit	: Soạn thảo trực tiếp khi đang chạy	Protocol	: Giao thức
Operand	: Toán hạng	Proximity Sensor	: Cảm biến tiệm cận
Operation	: Toán tử lệnh	Pulse	: Xung
OR function	: Hàm hoặc	PV (Present Value)	: Giá trị hiện tại
OR INHIBIT function	: Hàm cấm hoặc	Read Only	: Chỉ đọc
Output	: Đầu ra	Real Time	: Thời gian thực
Over load	: Quá tải	Register	: Thanh ghi
PC (Personal Computer)	: Máy tính cá nhân	Remote	: Từ xa
Peripheral Device	: Thiết bị ngoại vi	Retentive Data	: Dữ liệu duy trì
Peripheral Port	: Cổng ngoại vi	Rung	: Dây thang
Photo Sensor	: Cảm biến quang	Scan	: Quét vòng
Point	: Điểm, đầu	Selector Switch	: Chuyển mạch
Point to Multipoint	: Kết nối điểm - nhiều điểm	Series	: Dòng (sản phẩm)
Point to Point	: Kết nối điểm - điểm	Service Life	: Tuổi thọ
Power Cable	: Cáp nguồn	Shift Register	: Thanh ghi dịch
Power Supply Bus	: Bus nguồn cung cấp	Single	: Đơn nhiệm
Power Supply Unit	: Nguồn cung cấp	Sinking	: Được cấp nguồn
Press Button	: Nút nhấn	Slave	: Trạm tớ
Program	: Chương trình	Slot	: Khe cắm
Program File	: File chương trình	SM (Special Memory)	: Vùng nhớ đặc biệt
Program Memory	: Bộ nhớ chương trình	Sourcing	: Cấp nguồn
PLC (Programmable Logic Controller)	: Bộ điều khiển logic khả trình	Specification	: Các đặc tính kỹ thuật
		Static RAM	: Ram tĩnh

Status	: <i>Trạng thái</i>
Status LED	: <i>LED chỉ thị trạng thái</i>
Stepper Motor	: <i>Động cơ bước</i>
STL (Statement List)	: <i>Dãy lệnh</i>
SV- Set Value	: <i>Giá trị đặt</i>
Switch	: <i>Chuyển mạch</i>
Syntax	: <i>Cú pháp</i>
Terminal	: <i>Điểm, đầu nối</i>
Time	: <i>Thời gian</i>
Timer	: <i>Bộ định thời</i>
Transducer	: <i>Bộ chuyển đổi</i>
Transmitter	: <i>Bộ truyền tín hiệu</i>
Upload	: <i>Đọc chương trình từ PLC</i>
Variable	: <i>Biến</i>
Volatile	: <i>Không duy trì</i>
Wet Run	: <i>Chạy công nghệ</i>
Wiring	: <i>Nối dây</i>
Word	: <i>Từ dữ liệu</i>
Work Bit	: <i>Bit trung gian</i>
YES function	: <i>Hàm đồng nhất</i>

TÀI LIỆU THAM KHẢO

1. Frank D. Petruzella, *Programmable Logic Controller*, Glencoe/McGraw – Hill 1998.
2. Peter Rohner, *Automation with Programmable Logic Controllers*, Southwood Press, Marrickville, NSW, 1996.
3. Allen – Bradley PLC, *Advanced Programming Software – User Manual & Reference Manual*, (Catalog No. 1747.PA2E).
4. OMRON PLC, SYSMAC CQM1/CPM1/CPM1A/SRM1, *Programmable Controllers, Programming Manual*.
5. OMRON PLC, SYSMAC CQM1 *Training Manual*.
6. SIEMENS PLC, *S7–200 and S7–300 Statement List and Ladder Logic Programming Reference Manual*.

CHỈ MỤC

B

- Bản đồ bộ nhớ 28, 55
Bảng mạch BUS 13, 21, 27, 32, 48-49, 53-54, 57, 75, 85
Biến 9, 13-14, 18, 22, 27, 33, 35, 36-39, 44-45, 57, 59, 66, 68-69, 84, 87-88, 118-119, 121-122, 128, 147, 149, 154-156, 159, 160, 162-163, 165, 167-169
Bit đầu ra hai trạng thái ổn định 104
Bit trung gian 51, 57, 60, 70-71, 73-74, 76, 104
Bit 7, 13, 16-17, 27-28, 33-34, 36, 39, 44-45, 48-51, 56-57, 59-61, 63-69, 71-77, 79-82, 94, 98, 104-105, 127, 178, 183, 191
Bộ chuyển đổi 32, 44, 85
Bộ đếm 11, 29, 57-58, 60, 63, 65-66, 69-70, 72, 74, 108, 111, 116-119, 122
Bộ điều khiển logic khả trình 9
Bộ định thời 57, 74, 108
Bộ lập trình cảm tay 14, 25, 34
Bộ nhớ chương trình 13, 28, 55, 85, 87-88
Bộ nhớ dữ liệu 28, 55-56, 58-59, 61
Bộ nhớ 9-10, 12-16, 18-20, 24, 27-30, 35, 54-56, 58-60, 66, 87-88, 95
Bộ truyền tín hiệu 44
Bus dữ liệu 28, 32
Bus địa chỉ 28, 32
Bus điều khiển 28, 32
Bus nguồn cung cấp 32, 44

C

- Cài đặt 10, 14, 19, 25, 34-35, 71, 88
Cảm biến quang 36, 154-156, 160
Cảm biến tiệm cận 36
Cấu hình 10, 13, 20-21, 26-28, 31-32, 36, 52-54, 56, 60, 71-73, 83, 85-87
Cấu trúc dữ liệu 56-60, 70-72
Còi 11, 152, 167
Cổ định 13, 18, 20, 29, 31, 39, 58, 87, 108, 124
Công cụ lập trình 25, 87, 93
Công tắc hành trình 10-11, 13, 36
Công tắc 10-11, 13, 36, 86, 131, 145, 151-152, 160, 165
Cổng truyền thông 31, 46, 69, 71, 73
Cờ 18, 29, 31, 57, 60, 63, 68, 70-73, 75-76, 89, 108-109, 116
Cú pháp 86, 93, 104-105, 109, 116, 133
Chạy công nghệ 86, 87
Chạy mô phỏng 86
Chế độ hoạt động 30-31, 69, 71, 73
Chỉ đọc 57, 60, 66-67, 71
Cho phép 30-31, 41, 46-48, 56, 59, 71, 87, 93, 109, 111, 131, 140, 168
Chốt 105-106
Chu kỳ quét 18, 59, 65
Chuyển mạch DIP 30

Chuyển mạch 13, 30-31, 36, 44-45,
47, 51, 68, 86, 107, 112, 114,
119, 156, 160

Chương trình 9-13, 15-19, 21,
25, 27-31, 34, 47, 55-58, 60,
70-71, 76, 85-90, 94-97, 99,
109-110, 115, 119-120, 128-130,
142, 144-146, 148-149, 154-155,
159, 161, 162, 164, 166-167,
169

D

Dãy lệnh 88

Dây thang 16, 91-92

Dòng điện rò 48

Dung lượng 20, 24, 28-29, 35,
53, 55-56, 58, 85

Duy trì 27-29, 31, 57, 59, 66,
69-71, 104-106, 111, 113, 116,
121

Dữ liệu duy trì 29

Dữ liệu 9, 12-13, 16, 18-19, 21-22,
27-30, 32-36, 39, 46-50, 52,
54-62, 64-66, 70-77, 80, 85-86,
94-95, 106, 116, 126-127, 134

Đ

Đa nhiệm 19, 21

Đầu nối 13, 15-17, 32, 85-86

Đầu ra 16-18, 27, 33, 36, 39-43,
46-48, 51, 56, 59-60, 64, 66,
70, 72-75, 95, 105, 107, 109,
112-113, 118-122, 128, 131,
140, 144-147, 149-150, 152,
154-156, 159, 160, 162-163,
165, 167, 169

Đầu vào 15-18, 32-34, 36-39, 44-46,
48-49, 51-54, 56-57, 59-60,
63-64, 66, 70, 72-75, 85, 93-95,

104-105, 107, 109, 113, 116,
121-122, 127-128, 131, 140,
144-147, 149-150, 152, 154-156
159-160, 162-163, 165, 167, 169

Địa chỉ 15-17, 28, 32, 34-35,
49-54, 57, 63-65, 70-72, 74-81,
83, 85, 89, 95, 107, 109,
112-114, 119-122, 126-128, 131,
134, 140, 144-147, 149-150,
152, 154-156, 159-160, 162-163,
165, 167, 169

Đọc chương trình 87

Động cơ bước 45

Đồng hồ 68, 156

Đơn nhiệm 21

Được cấp nguồn 42

F

File chương trình 85-87, 95

File dữ liệu 47, 56, 73-76, 78-83,
85, 87

G

Giá trị đặt 31, 46, 57-58, 63, 78-80,
85, 87

Giá trị hiện tại 65, 71, 109, 111, 116

Giản đồ thang 16, 19, 55, 88-92, 97

Gián tiếp 89, 98

Giao diện 12, 19, 30-32, 52

Giao thức 46

H

Hàm cấm hoặc 98, 102

Hàm cấm 98, 100

Hàm cộng module2 98, 100

Hàm đồng nhất 98

Hàm hoặc 98-99

Hàm phủ định hoặc 98, 101
Hàm phủ định và 98, 101
Hàm phủ định 98-99
Hàm và 98-99
Hình thang 14-15, 90-91

K

Kết nối điểm – điểm 46
Kết nối điểm – nhiều điểm 46
Kết nối 12, 21, 22, 30, 35-36,
46-47, 58, 86-88
Khe cắm 13, 21, 31-32, 34, 49,
51-54, 72, 74-75, 83, 85
Khối chức năng 5, 7, 26, 36-37,
40, 52, 55, 88-89, 91-93, 98-102,
104, 123-124, 126-127
Khối vào/ra 54
Khối xử lý trung tâm 5, 12, 27
Không cho phép 30-31, 87, 109
Không duy trì 27-29, 55, 59
Không gian nhớ 17, 39
Không tích cực 17, 39, 51, 95

L

Lắp ráp 16, 76, 135
Lập trình 3, 5-7, 9-10, 12-14,
19, 22, 25, 30-32, 34-35, 55-56,
70, 76, 85-90, 93, 96, 98,
145-146
LED chỉ thị trạng thái 29, 36-37
Lệnh mở rộng 117
Lệnh 7-8, 12, 14, 16, 28-29,
34, 44-45, 52, 55-56, 58-59, 63,
65, 67-73, 76, 80, 85, 87-93,
95-96, 98, 104-105, 109, 111,
116-117, 123, 126, 131, 133-139,
175

Lỗi nhẹ 30
Lỗi nặng 30

M

Mã ASCII 67
Mã hàm 98
Mã HEXA 28, 61, 67-68, 133
Mã ngữ 16
Mạch logic 9, 12, 14, 16, 37,
40, 90, 92-93, 96, 123
Máy tính cá nhân 34-35
Miền tham số cấu hình 58

N

Nạp chương trình 30, 34, 87-88
Nối dây 10-12, 16, 37, 90
Ngắt 9, 19-19, 27, 29-31, 59,
69-70, 72, 94, 104, 122
Nguồn cung cấp 5, 10, 12-13, 15,
20, 26-30, 37, 40-42, 44, 48,
58, 116
Nhị phân 7, 28, 45, 55, 67-68,
93, 133-134

P

Pin 25, 27-28, 52, 159
Phản từ 10-12, 15-16, 36-38,
40-41, 74-79, 81-83, 89-93, 96,
98-102, 104

Q

Quá tải 15
Quét vòng 5, 9, 17-19

R

RAM động 55
RAM tĩnh 55
Rơ le trong 104

Rời rạc 5, 6, 9, 34, 36-40, 24,
49-52, 54, 58, 66

S

Soạn thảo 13-14, 25, 30, 34, 56,
85-89, 93, 95-96, 98

T

Tại chỗ 51, 54

Tích cực 17, 39, 42-43, 48, 51,
95, 116, 127

Tiếp điểm 7, 9-10, 14-16, 36,
40, 42, 90-91, 93-95

Toán hạng 7, 16, 52, 72, 88-89,
92-93, 95, 98, 109, 116, 126

Tốc độ cao 28-29, 45, 58, 60,
63, 65-66, 69-70, 72

Tuổi thọ 8, 31, 167

Từ dữ liệu kép 59, 61

Từ dữ liệu 29, 34, 52, 70,
126-127, 133-134

Từ xa 30, 54

Tức thời 18, 89, 113

Tương tự 5, 9, 12, 16-19, 25-27,
34, 44-46, 51-52, 54, 57-58, 60,
63-64, 66, 72, 88-89, 91, 96,
112

Thanh chứa 60, 63, 65-66

Thanh ghi dịch 126-127

Thanh ghi 16-17, 59-66, 69,
126-127, 165

Thành phần 3, 5, 9-10, 12, 19,
21, 27-29, 30, 36, 44, 46, 56,
58, 85

Thập phân 61, 67, 113, 134

Thẻ nhớ 29, 31, 87

Thiết bị lập trình 12-14, 19, 25,
30-31, 56, 86-88

Thiết bị ngoại vi 5, 30, 32, 34-36,
47, 52, 58

Thiết bị trường 13, 42, 44, 47-48

Thiết kế 3, 6, 9-10, 12, 19, 25,
84-86, 89-90

Thời gian thực 18, 31

Thuật toán 10

Thường kín 90-91

Thường mở 90

Trạm chủ 21

Trạm tớ 22

Trực tiếp 9, 19, 30, 44-45, 55,
59, 85, 88-89, 98

V

Vào/ra 5, 6, 9-10, 12-13, 15-18,
20-21, 24, 26-27, 32-34, 36, 42,
44-55, 57-59, 69, 75, 83-86,
107, 113-114, 119, 122, 128,
131, 140, 144-145, 147, 149-150,
152, 154-156, 159-163, 165,
167, 169

Vùng ảnh 16-18, 33, 36, 39,
56-58, 70, 73, 93-94

Vùng nhớ đặc biệt 51, 66, 70

Vùng nhớ 16, 28, 35, 44, 48-49,
51-52, 55-56, 65-68, 70-73, 116

X

Xung 27, 37, 45, 48, 68-69, 73,
108, 111-112, 116, 124-127,
155

BỘ ĐIỀU KHIỂN LOGIC KHẢ TRÌNH PLC VÀ ỨNG DỤNG

NHÀ XUẤT BẢN BÁCH KHOA HÀ NỘI

Ngõ 17 Tạ Quang Bửu – Hai Bà Trưng – Hà Nội

ĐT: 04. 38684569;

Fax: 04. 38684570

<http://nxbbk.hust.edu.vn>

Chịu trách nhiệm xuất bản:

Giám đốc – Tổng biên tập: TS. PHÙNG LAN HƯƠNG

Biên tập: ĐỖ THANH THÙY

Sửa bản in: VŨ THỊ HẰNG

Trình bày: NGUYỄN KIM CHI

In 500 cuốn khổ 16 × 24 cm tại Công ty TNHH in – thương mại và dịch vụ Nguyễn Lâm, số 352 đường Giải Phóng, Thanh Xuân, Hà Nội.

Số xuất bản: 205 – 2015/CXBIPH/14 – 02/BKHN; ISBN: 978–604–938–208–6.

Số QĐXB: 34/QĐ – ĐHBK – BKHN ngày 22/04/2015.

In xong và nộp lưu chiểu quý II năm 2015.