

### TUYÊN BỐ BẢN QUYỀN

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

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

## **LỜI GIỚI THIỆU**

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

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

Nội dung giáo trình được biên soạn với dung lượng thời gian đào tạo 120 giờ gồm có:

Bài MĐ22-01: Đại cương về điều khiển lập trình

Bài MĐ22-02: Cấu trúc và phương thức hoạt động của một PLC

Bài MĐ22-03: Kết nối giữa PLC và thiết bị ngoại vi

Bài MĐ22-04: Các phép toán nhị phân của PLC

Bài MĐ22-05: Các phép toán số của PLC

Bài MĐ22-06: Bộ xử lý tín hiệu Analog

Bài MĐ22-07: Các bài tập ứng dụng trong điều khiển động cơ

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

Tham gia biên soạn Chủ biên ThS Nguyễn Mạnh Cường

# MỤC LỤC

A	Lời g	yi giới thiệu					
1	<b>Đại (</b>	<b>cương về điều khiển lập trình</b>	1				
	1.1	Tổng quan về điều khiển	2				
	1.2	Điều khiển nối cứng và điều khiển lập trình	4				
	1.3	So sánh PLC với các hình thức điều khiển khác	8				
	1.4	Các ứng dụng của PLC trong thực tế	14				
2	Cấu	trúc và phương thức hoạt động của một PLC	20				
	2.1.	Cấu trúc của một PLC	21				
	2.2.	Thiết bị điều khiển lập trình S7-200	24				
	2.3.	Địa chỉ các ngõ vào/ ra	31				
	2.4.	Cấu trúc bộ nhớ của S7- 200	33				
	2.5.	Xử lý chương trình.	36				
3	Kết 1	<b>nối dây giữa PLC và thiết bị ngoại vi</b>	37				
	3.1.	Kết nối dây giữa PLC và thiết bị ngoại vi	38				
	3.2.	Cài đặt và sử dụng phần mềm Step7-Micro/win 3.2	51				
	3.3.	Kiểm tra việc kết nối bằng phần mềm Step7-Micro/win 3.2	69				
4	Các 4.1. 4.2. 4.3. 4.4. 4.5.	<ul> <li>phép toán nhị phân của plc</li></ul>	71 72 72 75 77 81 87 93 93 93 93 93 93 93 93 102 102 103 103 104 105 106				
5	Các 5.1. 5.2.	<b>phép toán số của PLC</b> Chức năng so sánh Chức năng dịch chuyển 5.2.1. Các lệnh dịch chuyển nội dung ô nhớ	107 108 113 113				

	5.2.2.	Các lệnh dịch chuyển thanh ghi11	5
5.3.	Chức r	năng chuyển đổi12	1
5.4.	Chức r	năng toán học122	2
	5.4.1.	Lệnh cộng 122	2
	5.4.2.	Lệnh trừ 12	2
	5.4.3.	Lệnh nhân 124	4
	5.4.4.	Lệnh chia12	5
	5.4.5.	Lệnh lấy căn bậc hai 12	6
	5.4.6.	Lệnh cộng trừ số nguyên12	6
5.5.	Các ch	ức năng khác	0
	5.5.1.	Lệnh nhảy chương trình con13	0
	5.5.2.	Đồng hồ thời gian thực13	3
	5.5.3.	Các lệnh can thiệp vào thời gian vòng quét13	5
Các	bài tập t	ứng dụng trong điều khiển động cơ130	6
6.1.	Giới th	13'	7
6.2.	Bài tập	) ứng dụng13	8
	6.2.1.	Mạch khởi động động cơ13	8
	6.2.2.	Mạch đổi chiều quay14	0
	6.2.3.	Mạch điều khiển tốc độ14	3
	6.2.4.	Mạch mở máy sao/ tam giác14	5
	6.2.5.	Mạch khởi động động cơ trường hợp có tảI nặng 14	7
Phụ	lục		ii

6

B

# GIÁO TRÌNH MÔ ĐUN

#### Tên mô đun: PLC Cơ bản

#### Mã môn học/mô đun: MĐ22

#### Vị trí, tính chất, ý nghĩa và vai trò của mô đun:

- Vị trí: Trước khi học mô đun này cần hoàn thành các môn học: An toàn điện; Điện tử công suất; Điều khiển điện khí nén; Trang bị điện.

- Tính chất: Là mô đun đào tạo nghề bắt buộc và mang tính tích hợp.

- Ý nghĩa và vai trò của mô đun:

#### Mục tiêu của môn học/mô đun:

\* Về kiến thức:

- Trình bày được nguyên lý hệ điều khiển lập trình PLC; So sánh các ưu nhược điểm với bộ điều khiển có tiếp điểm và các bộ lập trình cỡ nhỏ khác.

- Phân tích được cấu tạo phần cứng và nguyên tắc hoạt động của phần mềm trong hệ điều khiển lập trình PLC.

- Phương pháp kết nối dây giữa PC - CPU và thiết bị ngoại vi.

\* Về kỹ năng:

- Thực hiện được một số bài toán ứng dụng đơn giản trong công nghiệp.

- Kết nối thành thạo phần cứng của PLC - PC với thiết bị ngoại vi.

- Viết được chương trình, nạp trình để thực hiện được một số bài toán ứng dụng đơn giản trong công nghiệp.

- Phân tích được một số chương trình đơn giản, phát hiện sai lỗi và sửa chữa khắc phục.

\* Về năng lực tự chủ và trách nhiệm:

- Rèn luyện đức tính cẩn thận, tỉ mỉ, tư duy sáng tạo và khoa học, đảm bảo an toàn, tiết kiệm.

- Phát huy tính tích cực, chủ động, sáng tạo, tác phong công nghiệp.

#### Nội dung của môn học/mô đun:

#### Bài 1: Đại cương về điều khiển lập trình Mã chương/Bài: MĐ22-01

**Giới thiệu:** Bài này giới thiệu khái niệm về điều khiển lập trình, so sánh ưu nhược điểm của điều khiển lập trình với các hình thức điều khiển khác và trình bày một số các ứng dụng cơ bản của PLC trong thực tế.

#### Mục tiêu:

- Phát biểu khái niệm về điều khiển lập trình theo nội dung đã học.

- Trình bày được tổng quan về hệ thống điều khiển và các dạng điều khiển cơ bản.

- Trình bày được các ưu, nhược điểm của điều khiển lập trình so với các loại điều khiển khác và các ứng dụng của chúng trong thực tế.

- Trình bày được khái niệm và đặc điểm của PLC.

- Phân tích được các dạng bài toán điều khiển và giải bài toán điều khiển.

#### Nội dung chính:

1.	Tổng quan về hệ thống điều khiển
2.	Điều khiển nối cứng và điều khiển lập trình
3.	Bài toán điều khiển và giải quyết bài toán điều khiển
4.	Giới thiệu chung về PLC
	Các ứng dụng của PLC trong thực tế

# 1. Tổng quan về hệ thống điều khiển

1.1. Hệ thống điều khiển là sự tập trung của các máy móc và thiết bị cơ – điện – điện tử. Nó điều khiển hay vận hành thông suốt một quá trình hoặc một hoạt động chế tạo, đảm bảo sự linh hoạt, ổn định và có tính chính xác cao. Hệ thống điều khiển thi hành bất cứ hình thức nào: từ đơn giản cho đến phức tạp tuỳ thuộc vào hệ thống được thiết lập hoặc chương trình được cài đặt sẵn nơi thiết bị điều khiển trung tâm.

Đơn giản có thể hiểu Hệ thống điều khiển sẽ tự động thực thi một quá trình hoặc một hoạt động chế tạo mà không cần có sự can thiệp của con người tuy nhiên vẫn bảo đảm sự linh hoạt, ổn định và có tính chính xác cao. Để giải quyết được vấn đề này thì cơ bản Hệ thống điều khiển phải có ba thành phần cơ bản đó là thiết bị điều khiển trung tâm, thiết bị đầu vào và cơ cấu chấp hành. Ba thành phần này liên kết với nhau thành một hệ thống hoàn chỉnh đảm bảo có thể thực thi một qui trình hay một hoạt động chế tạo cụ thể.

1.2. Còn gọi là Khối cảm biến ngõ vào hay các thiết bị trường nhập. Là các thiết bị ngoại vi trong HTĐK. Nó cung cấp các tín hiệu bị tác động bởi các tác nhân bên ngoài hoặc các tín hiệu kiểm tra, kiểm soát cho bộ xử lý trung tâm. Nút nhấn, cầu dao, các hình thức giao diện cơ bản, các thiết bị cảm biến, cảm ứng ... là dạng của các thiết bị trường nhập.

1.3. Cơ Còn gọi là các thiết bị đầu vào bên ngoài hay các thiết bị trường xuất. Trong cấu chấp HTĐK nó cũng là các thiết bị bên ngoài. Cơ cấu chấp hành bị tác động hay điều khiển bởi bộ xử lý trung tâm. Mọi hoạt động, hành vi của các thiết bị trường xuất đều chấp hành theo sự điều khiển của bộ xử lý trung tâm. Đèn, chuông, động cơ ... là dạng của các thiết bị trường xuất.

1.4. Thiết bị Điều khiển trung tâm (CPU: Central Processing Unit).

Thiết bị
Còn gọi là Bộ xử lý trung tâm hay thiết bị điều khiển chính; máy chủ. Nơi
Điều
liên kết, tổng hợp và quyết định mọi hoạt động của hệ thống. Trong suốt một
quá trình hay một hoạt động chế tạo thiết bị này liên tục kiểm tra trạng thái
trung
của hệ thống, thu nhận tín hiệu phản hồi từ các thiết bị trường nhập ( khối

tâm



cảm biến ngõ vào) dựa vào chương trình logic (đã được thiết lập và cài đặt) để xử lý sau đó quyết định chu kỳ hoạt động cho cơ cấu chấp hành. PLC, OP3, Logo, hoặc các thiết bị lập trình được khác là dạng của bộ xử lý trung tâm.

# 2. Điều khiển nối cứng và điều khiển lập trình

#### Trong kỹ thuật điều khiển cũng như tự động hoá người ta chia ra làm 2 loại điều khiển: điều khiển nối cứng và điều khiển lập trình (điều khiển khả trình).

2.1. Điều biều khiển nối cứng là dạng điều khiển sử dụng các tiếp điểm. Trong hệ khiển nối thống điều khiển nối cứng chúng ta sử dụng các khí cụ điện như công tắc, nút nhấn, role, khởi động từ, các bộ cảm biến,... kết hợp với các thiết bị điện như đèn, chuông, động cơ (1 pha, 3 pha),... Các thiết bị, khí cụ điện này được nối lại với nhau theo một mạch điện cụ thể để thực hiện một yêu cầu công nghệ nhất định (điều khiển hay vận hành một quá trình hoặc một hoạt động chế tạo cụ thể).





*Ví dụ* Lắp mạch điện điều khiển khởi động động cơ không đồng bộ 3 pha dạng đơn giản. Yêu cầu mạch điều khiển sử dụng nguồn điện 24VAC.

Hình dưới là sơ đồ mạch điện dạng điều khiển nối cứng sử dụng tiếp điểm, trong đó:

- □ M làcông tắc tơ sử dụng nguồn điện 460VAC.
- CR là role trung gian sử dụng nguồn điện 24VAC.
- Nút nhấn thường mở Start (NO: Normal open), thường đóng Stop (NC: Normal close).
- Ngoài ra còn có cầu chì, máy biến áp 460/24VAC và Role quá tải (OL: Overload).



Nhận xét	Điều khiển nối cứng:
----------	----------------------

- Chức năng được đặt cố định (nối dây, mạch điện tử).
- Nếu muốn thay đổi chức năng có nghĩa là phải thay đổi lại kết nối dây hay thay đổi mạch điện tử.
- Diều khiển nối cứng có thể thực hiện với các tiếp điểm (role, công tắc tơ,...) hay mạch điện tử.

2.2. Điều khiển
PLC viết tắt của Progammable Logic Control. Là thiết bị điều khiển Logic lập trình được hay khả trình, cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua ngôn ngữ lập trình. Nó bao gồm một khối vi xử lý trung tâm chứa chương trình ứng dụng (liên kết các hoạt động của hệ thống PLC : thi hành chương trình; xử lý tín hiệu nhập xuất và chuyển giao với các thiết bị ngoài). Bộ nhớ (memory) nắm giữ hệ điều hành và vùng nhớ chương trình người dùng và cũng là nơi lưu trữ chương trình điều khiển và các chương trình nhớ trung gian.

Các loại bộ nhớ ROM: Read only Memory RAM: Random Access Memory.

EFROM: Electrically Erasable Programmable Read only Memory

Và các loại module giao diện nhập xuất.

PLC chỉ có thể hoạt động được sau khi đã được lập trình. Có nghĩa rằng chương trình ứng dụng được người sử dụng viết (thiết lập) bằng bộ bàn phím lập trình cầm tay (Programming Consle) hoặc bằng máy tính (PC) dựa trên các phần mềm ứng dụng được cài đặt trong máy (như SYSWIN đối với PLC của hãng OMRON; STEP – Micro đối với PLC của hãng SIEMEN) sau đó nạp vào bộ nhớ chương trình của PLC.





*Nhận xét* Điều khiển lập trình được

- Chức năng được đặt cố định (thông qua một chương trình còn gọi là bộ nhớ chương trình).
- Các phần tử nhập tín hiệu được nối ở ngõ vào (Input) của bộ điều khiển.
- Các cuộn dây ngõ ra được khởi động bằng các phần tử ở ngõ vào và chương trình trong bộ nhớ.
- Quá trình điều khiển ở đây được thực hiện bằng một chương trình soạn thảo và đưa vào bộ nhớ chương trình
- Nếu muốn thay đổi chức năng thì chỉ cần thay đổi chương trình trong bộ điều khiển.

#### 3. So sánh PLC với các hình thức điều khiển khác

Trong công nghiệp, yêu cầu tự động hóa ngày càng tăng, đòi hỏi kỹ thuật điều khiển phải đáp ứng được các yêu cầu đó. Trong những năm gần đây, bên cạnh việc điều khiển bằng role và khởi động từ thì việc điều khiển có thể lập trình được càng phát triển với hệ thống đóng mạch điện tử và thực hiện lập trình bằng máy tính.

Trong nhiều lĩnh vực, các loại điều khiển cũ đã được thay đổi bởi điều khiển có thể lập trình được, có thể gọi là điều khiển logic khả trình. Viết tắt trong tiếng Anh là **PLC** (Programmable Logic Controller)

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

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

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



So sánh

Khi thay đổi nhiệm vụ điều khiển thì người ta thay đổi mạch điều khiển : Lắp lại mạch, thay đổi các phần tử mới ở hệ điều khiển bằng role điện.

Trong khi đó thay đổi nhiệm vụ điều khiển ở hệ điều khiển logic khả trình (PLC) thì người ta chỉ thay đổi chương trình soạn thảo.

Sự khác nhau giữa hệ điều khiển bằng rơle điện và hệ điều khiển logic khả trình có thể minh họa một cách cụ thể như sau

Để thay đổi 1 mạch điện từ dạng mạch "Mở máy ĐC KĐB 3pha bằng phương pháp đổi nối Sao/Tam giác dạng tự động" sang dạng mạch "Mở máy ĐC KĐB 3pha bằng phương pháp khởi động qua biến trở dạng tự động".





3. Viết lại chương trình điều khiển.



- So sánh Khi thay đổi nhiệm vụ điều khiển. So sánh giữa 2 phương pháp trên ta thấy:
   Phương pháp sử dụng role và công tắc tơ:
  - □ Mạch điều khiển sẽ thay đổi rất nhiều.
  - Chi phí cho nhiệm vụ mới cao hơn.
  - Dehức tạp và tốn thời gian.



Phương pháp sử dụng thiết bị lập trình có nhớ (PLC):

- □ Mạch điều khiển không thay đổi.
- Nhanh và đơn giản hơn bằng cách thay đổi lại chương trình.



#### *Kết luận* Hệ điều khiển lập trình có nhớ PLC có những ưu điểm sau :

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

#### 1.4 Các ứng dụng của PLC trong thực tế

Hệ thống điều khiển theo lập trình có nhớ được sử dụng rộng rãi trong các nghành khác nhau và đặc biệt được sử dụng chủ yếu trong điều khiển công nghiệp như : điều khiển thang máy, điều khiển các quá trình sản xuất khác nhau, hệ thống rửa ôtô tự động, các dây chuyền đóng gói sản phẩm, điều khiển máy tự động (máy khoan, máy sấy,...), điều khiển đèn giao thông,...



Máy khai thác mỏ



Điều khiển dây chuyền lắp ráp máy tính



Máy dãn nhãn thuốc lá



Hệ thống khử mùi



Điều khiển xe nghiền rác



Khởi động mềm động cơ



Điều khiển thang máy





Hệ thống rửa xe tự động





Hệ thống giao tiền của nhà băng



Điều khiển hệ băng tải hàng hóa



Điều khiển thời gian cửa, đèn



Máy hàng góc khung cửa PVC



Hệ thống điều khiển an toàn cần trục



Hệ thống xử lý mốt nường





Máy đóng sách, tập vỡ





Điều khiển quá tải hệ thống điện

Máy băm và nghiền vật liệu



Diễu khiển máy ép nhựa



Máy lắp ghép bao bì kim loại



Diễu khiến động cơ bước





Máy làm nguội kim loại



Điều khiển tàu điện



Điều khiển máy CNC



Diều khiển robot



Hệ thống băng tải tự hành

#### Bài 2: Cấu trúc và phương thức hoạt động của PLC Mã chương/Bài: MĐ22-02

**Giới thiệu:** Chương này giới thiệu cấu trúc của một PLC, thiết bị điều khiển lập trình S7 – 200, cấu trúc bộ nhớ S7 – 200 và thực hiện xử lý chương trình. **Mục tiêu:** 

- Phát biểu được cấu trúc của một PLC theo nội dung đã học.

- Trình bày được cấu trúc và nhiệm vụ các khối chức năng của PLC.

- Trình bày được cấu hình cứng của PLC cũng như các ng vo/ra, cấu trúc bộ nhớ, mô tả đèn báo, công tắc chọn chế độ và cổng truyền thông.

- Thực hiện được sự kết nối giữa PLC và các thiết bị ngoại vi.

- Lắp đặt được các thiết bị bảo vệ cho PLC theo yêu cầu kỹ thuật.
- Trình bày được phương pháp lập trình cho PLC S7-200.
- Cài đặt và sử dụng được phần mềm SIMATIC S7 MicroWIN.
- Rèn luyện tính tỉ mỉ, cẩn thận trong cơng việc.

#### Nội dung chính:

1.	Cấu trúc của một PLC
2.	Thiết bị điều khiển lập trình S7 - 200
3.	Kết nối dây giữa PLC và thiết bị ngoại vi
4.	Cài đặt và sử dụng phần mềm STEP 7 – MicroWin
5.	Xử lý chương trình
6.	Phương pháp lập trình

#### 1. Cấu trúc của một PLC

1.1. Thiết bị logic khả trình được lắp đặt sẵn thành bộ. Trước tiên chúng chưa có một nhiệm vụ nào cả. Tất cả các cổng logic cơ bản, chức năng nhớ, timer, counter vv... được nhà chế tạo tích hợp trong chúng và được kết nối với nhau bằng chương trình cho một nhiệm vụ điều khiển cụ thể nào đó. Có nhiều thiết bị điều khiển và được phân biệt với nhau qua các chức năng sau:

- □ Các ngõ vào và ra
- □ Dung lượng nhớ
- □ Bộ đếm (Counter)
- □ Bộ định thời (Timer)
- □ Bit nhớ
- □ Các chức năng đặc biệt

□ Tốc độ xử lý

□ Loại xử lý chương trình

Các thiết bị điều khiển lớn thì được lắp thành modul riêng. Đối với các nhiệm vụ điều khiển nhỏ, chúng được lắp đặc chung trong một bộ. Các bộ điều khiển này có số lượng ngõ vào/ra cho trước cố định.



Thiết bị điều khiển được cung cấp tín hiệu bởi các tín hiệu từ các cảm biến ở bộ phận ngõ vào của thiết bị tự động. Tín hiệu này được xử lý tiếp tục thông qua chương trình điều khiển đặt trong bộ nhớ chương trình. Kết quả xử lý được đưa ra bộ phận ngõ ra của thiết bị tự động để đến đối tượng điều khiển hay khâu điều khiển ở dạng tín hiệu. Cấu trúc của một PLC có thể được mô tả như hình vẽ trên:

1.2. Bộ nhớ chương trình trong PLC là một bộ nhớ điện tử đặc biệt có thể đọc được. Nếu sử dụng bộ nhớ đọc – ghi được (RAM), thì nội dung của nó luôn luôn được thay đổi, ví dụ như trong trừnh
trình
trừnh
trừnh
thì nội dung trong Ram có thể vẫn được giữ lại nếu như có sử dụng Pin dự phòng. Nếu sự điều khiển làm việc ổn định, hợp lý, chương trình có thể không thay đổi nếu nạp vào một bộ nhớ cố định, ví dụ như trong một EPROM. Nội dung chương trình ở EPROM có thể xóa bằng tia cực tím.

1.3. Hệ Sau khi bật nguồn, hệ điều hành sẽ đặt các counter, timer, và bit nhớ với thuộc tính nonretentive (không được nhớ bởi Pin dự phòng) cũng như accu về 0.

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

1.4. Bit nhớ (Memory Bit)	Các memory bit là các phần tử nhớ, mà hệ điều hành ghi nhớ trạng thái tín hiệu
1.5. Bộ đệm (ProcessI mage)	Bộ đệm là một vùng nhớ, mà hệ điều hành ghi nhớ các trạng thái tín hiệu ở các ngõ vào/ra nhị phân.
1.6. Accumul ator	Accumulator là một bộ nhớ trung gian mà qua nó timer hay counter được nạp vào hay thực hiện các phép toán số học.

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

Timer

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

#### 2. Thiết bị điều khiển lập trình S7 - 200

Tổng
S7 – 200 là thiết bị điều khiển lôgic khả trình loại nhỏ của hãng Siemen (Đức) có cấu trúc theo kiểu Modul và có các modul mở rộng. Các modul này được sử dụng cho nhiều những ứng dụng lập trình khác nhau. Thành phần cơ bản của S7 - 200 là khối vi xử lý, ví dụ CPU 212 (222) hoặc CPU 214 (224). Về hình thức bên ngoài sự khác nhau của 2 loại CPU này nhận biết được nhờ số đầu vào/ra và nguồn cung cấp. Ví dụ:

- CPU 212 (222) có 8 cổng vào và 6 cổng ra và có khả năng mở rộng thêm bằng 2 Modul mở rộng.
- CPU 214 (224) có 14 cổng vào và 10 cổng ra và có khả năng mở rộng thêm bằng 7 Modul mở rộng.
- S7 có nhiều loại Modul mở rộng khác nhau.

#### Cấu hình Modul nguồn điện cung cấp (PS : Power supply)

*cứng* Thông thường là nguồn điện 24 VDC

#### Modul xử lý trung tâm ( CPU : Control Proccessing Unit)

SIMATIC S7 - 200 có nhiều loại Modul xử lý trung tâm, một trong số đó là CPU 212 (222) và CPU 214 (224)

*CPU 212(222) bao gồm :* 

512 từ đơn (word) hay 1K byte để lưu chương trình thuộc miền bộ nhớ đọc/ghi được vàkhông bị mất dữ liệu nhờ có giao diện với EFPROM (vùng nhớ có tính chất như vậy được gọi là vùng nhớ Non – volatile)

- 512 từ đơn để lưu dữ liệu, trong đó có 100 từ nhớ đọc/ghi được thuộc miền non – volatile.
- □ 8 cổng vào logic và 6 cổng ra logic.
- □ Có thể ghép nối thêm 2 modul mở rộng cổng vào/ra bao gồm cả modul tương tự (Analog).
- □ Tổng số cổng logic vào/ra cực đại là 64 cổng vào và 64 cổng ra.
- 64 Timer trong đó 2 Timer có độ phân giải 1ms, 8 timer có độ phân giải 10ms, 54 timer có độ phân giải 100 ms.
- 64 counter được chia làm 2 loại. Bộ đếm chỉ đếm tiến và bộ đếm vừa đếm tiến vừa đếm lùi.
- 368 bit nhớ đặc biệt, sử dụng làm các bit trạng thái hoặc các bit đặt chế độ.

Bộ nhớ không bị mất dữ liệu trong khoảng thời gian 50 giờ khi PLC bị mất nguồn nuôi.

*CPU 214(224) bao gồm :* 

- □ 2048 Word (= 4 K byte) thuộc miền nhớ đọc/ghi non volatile
- 2048 Word (= 4 K byte)kiểu đọc/ghi để lưu dữ liệu, trong đó có 512 từ đầu thuộc miền nhớ non volatile
- □ 14 cổng vào logic và 10 cổng ra logic.
- Có 7 modul để mở rộng thêm cổng vào/ra bao gồm cả modul Analog.
- □ Tổng số cổng vào ra cực đại là 64 cổng vào và 64 cổng ra
- 128 Timer trong đó 4 Timer có độ phân giải 1ms,16 timer có độ phân giải 10ms, 108 timer có độ phân giải 100 ms.
- 688 bit nhớ đặc biệt, sử dụng thông báo trạng thái trạng thái hoặc các bit đặt chế độ.
- □ 3 bộ đếm tốc độ cao với nhịp 2 Khz và 7 Khz.
- 2 bộ phát xung nhanh cho dây xung kiểu PTO hoặc PWM.
- □ 2 bộ điều chỉnh tương tự.

Toàn bộ vùng nhớ không bị mất dữ liệu trong khoảng thời gian 100 giờ khi PLC bị mất nguồn.



Mô tả SF (đèn đỏ) : đèn SF sáng lên khi PLC có hỏng hóc

đèn báo

**RUN (đèn xanh)** : chỉ định PLC đang ở chế độ làm việc và thực hiên chương trình được nạp vào trong máy.

*STOP (đèn vàng)* : chỉ định PLC đang ở chế độ dừng.

*Ix.x (đèn xanh)* : chỉ định trạng thái tức thời của cổng Ix.x (x.x = 0.0 - 1.5). Báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.

**Qy.y (đèn xanh)** : chỉ định trạng thái tức thời của cổng Qy.y (y.y = 0.0 - 1.1). Báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.



Công tắc Công tắc chọn chế độ làm việc nằm phía trên, bên cạnh các cổng ra của S7 chọn chế 200 có 3 vị trí cho phép chọn các chế độ làm việc khác nhau cho PLC :

độ làm việc
 RUN cho phép PLC thực hiện chương trình trong bộ nhớ. PLC S7 - 200 sẽ rời khỏi chế độ RUN và chuyển sang chế độ STOP nếu trong máy có sự cố, hoặc trong chương trình gặp lệnh STOP, thậm chí ngay cả khi công tắc ở chế độ RUN. Nên quan sát trạng thái thực tại của PLC theo đèn báo.

*STOP* cưỡng bức PLC dừng công việc thực hiện chương trình đang chạy và chuyển sang chế độ STOP. O chế độ STOP PLC cho phép hiệu chỉnh lại chương trình hoặc nạp 1 chương trình mới.

**TERM** cho phép máy lập trình tự quyết định một trong chế độ làm việc cho PLC hoặc ở RUN hoặc ở STOP.

- Chỉnh định tương tự.
- CPU222 AC/DC/RLY . 5 1.0 E - STAND Mode Switch RUN TERM لع 2 5 STOP 2 ID. Analog Adjustment .5.6.7 212-1BB20-0XB0
- Pin và nguồn nuôi bộ nhớ.

Cổng truyền thông S7-200 sử dụng cổng truyền thông nối tiếp RS485 với phích nối 9 chân để phục vụ cho việc ghép nối với thiết bị lập trình hoặc với các trạm PLC khác. Tốc độ truyền cho máy lập trình kiểu PPI là 9600 baud. Tốc độ truyền cung cấp của PLC theo kiểu tự do là 300 ÷38.400 baud.

Để ghép nối S7-200 với máy lập trình PG702 hoặc các loại máy lập trình thuộc họ PG7xx có thể dùng một cáp nối thẳng MPI. Cáp đó đi kèm với máy lập trình.

Ghép nối S7-200 với máy tính PC qua cổng RS232 cần có cáp nối PC / PPI với bộ chuyển đổi RS232 / RS485.

Cổng truyền thông cho phép ghép nối với các trạm PLC khác hoặc là để kết nối với thiết bị lập trình (Touch panel, PC) sử dụng cáp nối đi kèm.

PC : cổng RS232 – cáp PC/PPI bộ chuyển đổi RS232/RS485

PLC cáp nối thẳng MPI (Online)





Cổng truyền	Chân	Giải thích $(5)$ $(1)$
thông	1	Đất Canada Canad
	2	24 VDC
	3	Truyền và nhận dữ liệu 9 6
	4	Không sử dụng
	5	Đất
	6	5 VDC (điện trở trong 100 $\Sigma$ )
	7	24 VDC (120mA tối đa)
	8	Truyền và nhận dữ liệu
	9	Không sử dụng

2.1. Địa CPU 212(222) có 8 cổng vào và 6 cổng ra và có khả năng mở rộng thêm bằng 2 Modul mở rộng. Các ngõ vào có địa chỉ từ I0.0 đến I0.7 và các ngõ ra có địa chỉ từ Q0.0 đến Q0.5
vào/ra

CPU 214(224) có 14 cổng vào và 10 cổng ra và có khả năng mở rộng thêm bằng 7 Modul mở rộng. Các ngõ vào có địa chỉ từ I0.0 đến I0.7 + I1.0 đến I1.5 và các ngõ ra có địa chỉ từ Q0.0 đến Q0.7 + Q1.0 đến Q1.1.

CPU 212(222) cho phép mở rộng nhiều nhất 2 Modul và CPU 214 (224) cho 2.2. phép nhiều nhất 7 Modul. Các Modul mở rộng tương tự và số đều có trong Phần S7 - 200.chữ chỉ vị trí và Có thể mở rộng cổng vào/ra của PLC bằng cách ghép nối thêm vào nó các kích Modul mở rộng về phía bên phải của CPU, làm thành một móc xích. Địa chỉ thước của các vị trí của Modul có cùng kiểu. Ví dụ như một Modul cổng ra không của ô thể gán địa chỉ của một Modul cổng vào. Cũng như một Modul tương tự nhớ không thể có đia chỉ như một Modul số và ngược lại. (Mở Các Modul mở rộng số hay rời rạc đều chiếm chỗ trong bộ đệm, ứng với số rông

rộng Các Modul mở rộng số này rồi rặc đều chiếm chỗ trong bộ đệm, t cổng đầu vào/ra của Modul.

vào/ra)

Sau đây là một ví dụ về cách đặt địa chỉ cho các Modul mở rộng trên CPU 214 (224):

		Мо	dul 0	Modul 1	Мо	dul 2	Modul 3	Мос	dul 4
CPU ( CPU	J 214 Dr U215	4 ng 4 ng	õ vào/ gõ ra	8 ngõ vào	3 ng Ana 1 ng An	õ vào alog/ gõ ra alog	8 ngõ ra	3 ngô Ana 1 ng Ana	ŏ vào 1log/ jõ ra alog
				1					
I0.0	Q0.0	I2.0	Q2.0	13.0	AIW0	AQW0	Q3.0	AIW8	AQW4
I0.1	Q0.1	I2.1	Q2.1	I3.1	AIW2		Q3.1	AIW10	
I0.2	Q0.2	I2.2	Q2.2	13.2	AIW4		Q3.2	AIW41	2
I0.3	Q0.3	I2.3	Q2.3	I3.3			Q3.3		
I0.4	Q0.4			I3.4			Q3.4		
I0.5	Q0.5			13.5			Q3.5		
I0.6	Q0.6			13.6			Q3.6		
I0.7	Q0.7			13.7			Q3.7		
I1.0	Q1.0								
I1.1	Q1.1								
I1.2									
I1.3									
I1.4									
I1.5									

CPU 224		4 In / 4 Out	8 In	4 Analog In 1 Analog Out	8 Out	4 Analog In 1 Analog Out	
10.0 10.1 10.2 10.3 10.4 10.5 10.6 10.7 11.0	Q0.0 Q0.1 Q0.2 Q0.3 Q0.4 Q0.5 Q0.6 Q0.7 Q1.0 Q1.1	Module 0 12.0 Q2.0 12.1 Q2.1 12.2 Q2.2 12.3 Q2.3 12.4 Q2.4 12.5 Q2.5 12.6 Q2.6 12.7 Q2.7	Module 1 13.0 13.1 13.2 13.3 13.4 13.5 13.6 13.7	Module 2 AIWO AQWO AIW2 AQW2 AIW4 AIW6	Module 3 Q3.0 Q3.1 Q3.2 Q3.3 Q3.4 Q3.5 Q3.6 Q3.7	Module 4 AIW8 AQW4 AIW10 AQW5 AIW12 AIW14	
11.2 11.3 11.4 11.5 11.6 11.7	Q1.2 Q1.3 Q1.4 Q1.5 Q1.6 Q1.7	Expansion I/O					
.ocal I/C	2						

2.4 Cấu trúc bộ nhớ của S7 - 200

Bộ nhớ của S7 – 200 được chia làm 3 vùng: Vùng nhớ chương trình, vùng nhớ dữ liệu và vùng nhớ thông số. Vùng nhớ chương trình, vùng nhớ thông số và một phần vùng nhớ dữ liệu được chứa trong Rom điện EFPROM. Đối với CPU 214 cho phép cắm thêm khối nhớ mở rộng, chương trình mới có thể nạp vào card nhớ này mà không cần đến thiết bị lập trình. Phần sau đây mô tả chi tiết về các vùng nhớ.

#### Vùng nhớ chương trình:

Là vùng nhớ được sử dụng để lưu giữ các lệnh chương trình. Vùng này thuộc kiểu đọc/ghi được và không bị mất đi khi mất nguồn cung cấp cho CPU (do chương trình được lưu giữ trong EFPROM).

### Vùng nhớ thông số:

Gồm các ô chứa các thông số cài đặt, mật khẩu, địa chỉ thiết bị điều khiển và các thông tin về các vùng trống có thể sử dụng. Nội dung của vùng nhớ này không bị mất đi khi mất điện giống như trong vùng nhớ chương trình.

#### Vùng nhớ dữ liệu:

Vùng nhớ dữ liệu là nơi làm việc, vùng này gồm các địa chỉ để lưu trữ các phép tính, lưu trữ tạm thời các kết quả trung gian và chứa các hằng số được sử dụng trong các chỉ dẫn hoặc các thông số điều chỉnh khác. Ngoài ra trong vùng này còn có các phần tử và đối tượng như: Bộ định thời, bộ đếm, các bộ đếm tốc độ cao và các ngõ vào/ra analog.

Một phần tử của vùng nhớ dữ liệu được đặt trong bộ nhớ đọc ghi thuộc kiểu non – volatie, vì vậy các hằng số cũng như các thông tin khác vẫn được duy trì khi mất điện giống như trong vùng nhớ chương trình. Một phần khác được chứa trong RAM, nội dung trong RAM cũng được duy trì trong một khoảng thời gian nhất định khi mất điện bằng một điện dung có độ rỉ thấp.

Vùng nhớ dữ liệu của S7 – 200 không chỉ chứa dữ liệu mà còn bao gồm các đối tượng dữ liệu.

Vùng dữ liệu là một miền nhớ động. Nó có thể được truy nhập theo từng bit,

từng byte, từng word hoặc theo từng double word và được sử dụng làm miền lưu trữ dữ liệu cho các thuật toán, các thuật toán, các hàm truyền thông, lập bảng, các hàm dịch chuyển, xoay vòng thanh ghi, con trỏ địa chỉ...

Vùng dữ liệu được chia thành những miền nhớ nhỏ có các công dụng khác nhau. Chúng được ký hiệu bằng các chữ cái đầu của tên tiếng Anh, đặc trưng cho công dụng riêng của chúng như sau:

- V Variable memory (Bộ nhớ biến)
- I, E Input image register (Bộ đệm ngõ vào)
- Q, A Output image register (Bộ đệm ngõ ra)
- M Internal memory bits
- SM Special memory bits

Tất cả các miền này, ngoại trừ một vài ô nhớ đặc biệt chỉ cho phép đọc, đều có thể truy nhập được theo từng bit, byte, word hoặc double word.

Vùng nhớ dữ liệu được cho như ở hình dưới.



Vùng đối tượng được sử dụng để lưu trữ dữ liệu cho các đối tượng lập trình như các giá trị tức thời, giá trị đặt trước của bộ đếm hay Timer. Đối tượng dữ liệu bao gồm các thanh ghi của timer, bộ đếm, bộ đếm tốc độ cao, bộ đệm vào ra tương tự và các thanh ghi Accumulator (AC).

#### 3. Kết nối dây giữa PLC S7 - 200 và thiết bị ngoại vi

PLC S7 – 200 được kết nối với các thiết bị trường nhập (Inputs) và các thiết bị trường xuất như hình sau:

Khối cảm biến ngõ vào : (Input Devices các thiết bị trường nhập): Là các thiết bị ngoại vi trong HTĐK. Nó cung cấp các tín hiệu bị tác động bởi các tác nhân bên ngoài hoặc các tín hiệu kiểm tra, kiểm soát cho bộ xử lý trung tâm. Nút nhấn, cầu dao, các hình thức giao diện cơ bản, các thiết bị cảm biến, cảm ứng ... là dạng của các thiết bị trường nhập.

Cơ cấu chấp hành : (Output Devices các thiết bị đầu vào bên ngoài hay các thiết bị trường xuất): Trong HTĐK nó cũng là các thiết bị bên ngoài. Cơ cấu chấp hành bị tác động hay điều khiển bởi bộ xử lý trung tâm. Mọi hoạt động, hành vi của các thiết bị trường xuất đều chấp hành theo sự điều khiển của bộ xử lý trung tâm. Đèn, chuông, động cơ ... là dạng của các thiết bị trường xuất.







# Ghép nối thiết bị vào







Ví dụ:





Giao diện kiểu Transistor và Relay:


## Ghép nối thiết bị ra







Ví dụ:

















Connector	Pin Number	PROFIBUS Signal	Port 0/Port 1
	1	Shield	Chassis ground
$\cap$	2	24 V Return	Logic common
Pin 1	3	RS-485 Signal B	RS-485 Signal B
•	P 4	Request-to-Send	RTS (TTL)
•	5	5 V Return	Logic common
•	6	+5 V	+5 V, 100 Ω series resistor
Pin 5	7	+24 V	+24 V
	8	RS-485 Signal A	RS-485 Signal A
	9	Not applicable	10-bit protocol select (input)
	Connector shell	Shield	Chassis ground













Order Number	Expansion Model	EM Inputs	EM Outputs	Removable Connector
6ES7 231-0HC22-0XA0	EM 231 Analog Input, 4 Inputs	4	120	No
6ES7 232-0HB22-0XA0	EM 232 Analog Output, 2 Outputs	(1 <u>4</u> 2)	2	No
6ES7 235-0KD22-0XA0	EM 235 Analog Combination 4 Inputs/1 Output	4	11	No



## 4. Cài đặt và sử dụng phần mềm Step7-Micro/win 3.2

## PHẦN MỀM V3.2 STEP 7 MicroWIN

V3.2 STEP 7 MicroWIN là một phần mềm lập trình cho PLC SIEMENS S7 -200 dưới dạng LAD (Ladder Diagram), STL (Statement List Programming) và FBD (Fuction Block Diagram). Để cài đặt và chạy chương trình phần mềm này cần đảm bảo máy tính có cấu hình tối thiểu như sau :

☐ Microsoft Windows 95, Windows 98, Windows Me, Windows NT or Windows 2000 – WIN XP.

- □ Máy tính Pentium II 400 MHz (hoặc tương đương) trở lên
- $\square > 8 \text{ MB RAM}$
- $\square$  > 20 MB đĩa cứng trống

## CÀI ĐẶT PHẦN MỀM V3.2 STEP 7 MicroWIN

Dóng tất cả các chương trình ứng dụng đang chạy trên máy

☐ Đưa đĩa CD STEP 7-Micro/WIN 32 vào ổ CD – ROM. Chương trình sẽ tự động chạy nếu như ổ CD đặt ở chế độ chạy tự động, nếu không ta vào My Computer trên màn hình (Desktop) bằng cách nhắp đúp chuột trái (double click) vào biểu tượng

Bạn sẽ nhìn thấy hộp thoại My Computer



hiện ra.



Trong hộp thoại My Computer bạn chọn biểu tượng ổ CD Drive (J) và Double Click (hoặc Click chuột phải và chọn Open), một hộp thoại mới mở ra như sau :

🥶 PLC (J:)	
File Edit View Favorites Tools Help	1
🔇 Back 🝷 🕥 🖓 🏂 Search 🎼 Folders 🕼 🎲	× »
Address 💽 J:\	~
PLC plc ciemen	

Trong cửa sổ của hộp thoại PLC trên bạn Double click vào Folder plc siemen để mở ra một hộp thoại nữa :



Tiếp theo chọn Folder Step7MicroWin.32.

Đây chính là chương trình mà bạn cần cài đặt.

Hộp thoại Setup Step7MicroWin.32 hiện ra như bên

🗀 Step 7 Micro Win. 32	
File Edit View Favorites Tools Help	
🕜 Back 🔹 🕥 - 🎓 🔎 Search 🕼	Folders
Address 🗁 J:\plc ciemen\Step7MicroWin.32	~
0ata1 874 KB	HDR. File
data2 40,083 KB	ikernel EX_File 333 KB
BIN File 1 KB	setup INX File 185 KB
Setup 512 x 381 ACDSee BMP Image	Setup InstallShield (R) Setup Launcher InstallShield Software Corpora

Để cài đặt bạn chạy file Setup.exe (double click vào biểu tượng file Setup.exe), một hộp thoại lựa chọn ngôn ngữ cài đặt sẽ hiện ra. Trong cửa sổ bạn chọn English như bên dưới và nhấn OK.



Một hộp thoại kế tiếp hiện ra, bạn đánh dấu chọn Modify và click vào biểu tượng Next >



Hộp thoại hiện ra cho phép bạn lựa chọn ngôn ngữ cho chương trình sử dụng, click chọn bỏ các ngôn ngữ khác chỉ để lại English và nhấn Next

InstallShield Wizard	×
Select Components Choose the components Setup will install.	
Select the components you want to install, and install. German English French Spanish Italian	clear the components you do not want to Description Install German language support in application strings, on-line help, and sample projects.
Space Required on C: Space Available on C: 33 InstallShield	0 K 15936 K
	< Back Next > Cancel

Trong quá trình Setup một vài hộp thoại sẽ hiện ra ví dụ như hình bên dưới, yêu cầu xác lập, bạn chỉ cần chọn Yes hoặc OK.

Set PG/PC Interface							
Access Path							
Access Point of the Application: Micro/WIN> PC/PPI cable(PPI)							
(Standard for Micro/WIN)							
Interface Parameter Assignment Used: PC/PPI cable(PPI)	Properties						
IIII <none> IIIII PC/PPI cable(PPI)</none>	Copy Delete						
(Assigning Parameters to an PC/PPI cable for an PPI Network)							
Interfaces							
Add/Remove:	Select						
OK	Cancel Help						

Hộp thoại cuối cùng hiện ra như bên dưới, chọn Finish, kết thúc quá trình cài đặt. Bây giờ chương trình phần mềm V3.2 Step 7 MicroWIN đã được cài đặt trong máy của bạn.

InstallShield Wizard	
	InstallShield Wizard Complete The InstallShield Wizard has successfully installed STEP 7-Micro/WIN 32. Click Finish to exit the wizard. Yes, I want to view the Read Me file now. Yes, I want to launch STEP 7-Micro/WIN 32 now.
	< Back Finish Cancel

## GIỚI THIỆU VỀ MÔI TRƯỜNG SIMATIC STEP7 MicroWIN

## Bắt đầu với Step 7 MicroWIN

Khởi động Step 7 MicroWIN cũng giống như khởi động bất kì ứng dụng Windown nào.

Bước 1 Khởi động Step 7 MicroWIN từ Start menu bằng cách chọn Programs > Simatic > Step 7 - MicroWIN 32 V3.2.0.105 > Step 7 - MicroWIN 32

Môi trường Step 7 MicroWIN được xây dựng bằng các cửa sổ. Cửa sổ xuất hiện trên màn hình phụ thuộc vào cách thiết lập môi trường ban đầu của mỗi người.

Men	u chính	Navigation	Bar	<u>Toolbar</u>	<u>Cửa số</u>	viết chươn	ıg trình
	PLC Debug	Proje 11 Jos Windows	Held K IDX A				
tea lina		336 3	200 (#)	+±+-	+ + + 0	Co 110 000	
View Program Block Symbol Table Status Charl	Project Projec	ett(CPU 221 REL  rogram Block MAIN (0B1) SBR_0 (SBR0) INT_0 (INT0) ymbol Table tatus Chart ata Block ystem Block ross Reference ommunications ctions avorites t Logic		TTC EAD Symbol	Var Type TEMP TEMP TEMP TEMP TEMP	Data Type	
Data Block System Block		oork ommunications ompare onvert ounters oating-Point Math tener Math	Ni Ni			ß	
Ready						Row 1, Col 4	INS //

#### Thanh công cụ (ToolBar)

Các chức năng thanh công cụ của Step 7 MicroWIN giống như thanh công cụ của bất cứ ứng dụng Microsoft nào khác. Nó cung cấp những lệnh thực hiện các thao tác thông thường. Ngoài ra nó còn hiển thị đường dẫn các lệnh đang hiện hành.

Standard ToolBar



#### **Navigation Bar**

Navigation Bar là nơi định hướng những truy nhập tới công cụ lập trình, bạn có thể lựa chọn bằng cách nhấp chuột vào các nút ấn trong cửa sổ View.

Các công cụ lập trình bạn có thể truy xuất bằng biểu tượng trong cửa số View:



Nếu trên hộp thoại chính không hiện Navigation Bar ta chọn View > Frame > Navigation Bar

Instruction (thanh công cụ chọn lệnh)

Instru	ction					
⊋	_	←	<b>→</b>	⊣⊦	-0	1

<u>+</u>

-0

Box

Trong thanh công cụ chọn lệnh bạn có thể sử dụng các đường chèn kết nối lệnh (thẳng đứng hoặc nằm ngang); chèn các lệnh đầu vào (Contacts : các tiếp điểm), các lệnh ngõ ra (Coil : cuộn dây) hoặc các Box lệnh ứng dụng khác (Timer : bộ

định thời, Counter : bộ đếm ...)

Line Down : đường chèn kết nối lệnh - xuống dưới

- Line Up : đường chèn kết nối lệnh lên trên
- Line Left : đường chèn kết nối lệnh sang trái
- Line Right : đường chèn kết nối lệnh sang phải
- Contact : chèn các lệnh đầu vào (các tiếp điểm)
- Coil : chèn các lệnh ngõ ra (các cuộn dây)
  - : chèn các Box lệnh ứng dụng khác (Timer, Counter)

**Chú ý :** Trước khi ta chèn các lệnh thì phải chọn vị trí đặt lệnh sau đó di chuyển con chuột tới nút nhấn chọn lệnh để lựa chọn bằng cách nhấn Enter hoặc Double Click chuột vào biểu trợng lệnh.

 $N \acute{e}u$  thanh Instruction không hiển thị trên màn hình ta chọn View > Toolbars > Instruction như hình bên dưới :



# Tạo chương trình ứng dụng sử ngôn ngữ LAD

Ngôn ngữ LAD (Ladder Logic Programming) là ngôn ngữ lập trình bằng đồ họa. Trong chương trình LAD các phần tử cơ bản dùng để biểu diễn lệnh logic như sau :

- Tiếp điểm (Contact) :là biểu tượng mô tả các lệnh đầu vào hoặc các tiếp điểm sử dụng trong chương trình. Tiếp điểm đó có thể là thường mở, thường đóng hoặc các tiếp điểm khác.

- Cuộn dây (Coil) : là biểu tượng mô tả các lệnh ngõ ra hoặc các cuộn dây rơle được mắc theo chiều dòng điện cung cấp.

- Hộp (Box) : là biểu tượng mô tả các hàm khác nhau, nó làm việc khi có dòng điện chạy đến hộp. Những hàm thường được biểu diễn bằng hộp là các bộ định thời (Timer), bộ đếm (Counter) và các hàm toán học.

Chú ý : Cuộn dây và các hộp phải được mắc đúng chiều dòng điện.

## Có hai bước để tạo một chương trình ứng dụng :

- 1. Chèn các lệnh bằng cách kéo thả các biểu tượng lệnh trên thanh Instruction.
- 2. Thiết lập các giá trị hiện hành, địa chỉ và tên ban đầu cho các lệnh.

Trong phần thực hành sau, bạn sẽ tạo chương trình ứng dụng bằng ngôn ngữ LAD như sau :



**Bước 2** Chọn ngôn ngữ LAD để thiết lập chương trình (sau khi bạn khởi động chương trình Step 7 MicroWIN ở Bước 1). Di chuyển con trỏ đến Menu chính và chọn View > Ladder.

		🔚 STEP 7-1	Micro/WIN	32 - Proje	ct1 - [5	IMATIC L	AD]	
		📆 File Ec	lit View Pl	.C Debug	Tools	Windows	Help	
		∭ੴ ⊯ (	Ĵ STL ✓ Ladde	r			3	
		<b>1</b> 60 <b>1</b> 60	FBD				t (#)	
	Network	1 Network	Title					
Cửa sổ để tạo chương trình sẽ hiện ra như								
bên :	Network	2						
	Network	3						
Nếu như bạn không thấy		, SBR_0 🖌 INT,	_0/	•				<b>1</b> 11
cửa số trên thì b	ạn di chuyể:	n con trỏ tơ	ới Menu o	chính ch	ọn Vi	ew>		Program Plack

Component > Program Editor hoặc chọn biểu tượng Program Block ở Navigation Bar

Chèn lệnh. Trước tiên để chèn lệnh thì bạn phải di chuyển con trỏ xác định Bước 3 vị trí cần chèn lệnh sau đó di chuyển con trỏ lên thanh Instruction và nhắp vào biểu tượng hoặc biểu tương lênh cần chèn và nhấn Enter. Chon  $\dashv$  F

Hình bên dưới biểu diễn các bước bạn

chèn một lệnh.

 $- \bigcirc$ 

SIMATIC LAD **SIMATIC LAD** SIMATIC LAD ana tha ana 16 1/11 2 2 Network 1 Network Title Network 1 Network Title 27.7 Network 1 Tit ≫ -111-11-=R Network 2

Với các lệnh Coil hay lệnh Box ta cũng làm tương tự.

Thiết lập địa chỉ cho lệnh. Di chuyển con trỏ tới lệnh cần viết địa chỉ và Bước 4 Click hoặc nhấn Enter. (Bạn cũng có thể sử dụng các mũi tên lên, xuống, sang phải hoặc sang trái để di chuyển vị trí cần trọn lệnh).



**Chú ý** : Nếu bạn thiết lập địa chỉ cho lệnh sai thì sau khi hoàn thành dòng địa chỉ sẽ hiện màu đỏ. Một thiết lập đúng thì màu của dòng địa chỉ phải là màu đen.



## Kiểm tra lỗi

Sau khi bạn tạo hoàn thành một chương trình ứng dụng thì công việc tiếp theo là bạn phải kiểm tra xem chương trình bạn tạo có lỗi hay không. Phần mềm này cho phép bạn kiểm tra nhanh bằng cách :

Bước 5 Click lên nút Compile All trên thanh công cụ. Bạn cũng có thể kiểm tra lỗi bằng cách vào Menu chính và chọn PLC > Compile All.



Nếu như có lỗi thì phần màn hình hiển thị lỗi sẽ thông báo tổng số lỗi mà bạn tạo ra sai. Để kiểm tra chính xác, nhanh các lỗi mà bạn tạo sai thì bạn có thể vào thanh Menu chính và chọn View > STL các lỗi sai sẽ hiển thị màu đỏ. Bạn sẽ sửa các lỗi đó và kiểm tra lại đến khi nào không còn lỗi nữa thì thôi.



Màn hình hiển thị thông báo tổng

## Chạy chương trình ứng dụng

Để chạy được chương trình mà bạn tạo ra thì bạn cần phải có một thiết bị logic khả trình PLC cùng cáp chuyển đổi RS 232/RS 485 để kết nối với máy vi tính. Còn nếu không, các bước sau bạn không thể thực hiện được.

Sau khi bạn kết nối cáp giữa PLC và máy vi tính bạn phải cài đặt truyền thông (khi kết nối bạn chú ý địa chỉ cổng Com của máy vi tính để khai báo truyền thông).

Các bước cài đặt truyền thông như sau :

#### Chọn biểu tượng Communications trong cửa sổ Bước 6 Navigation bar hoặc vào thanh Menu chính chọn View > Component > Communications môt hôp thoai sẽ xuất hiện như sau :



Communications			X
Address Local: Remote: PLC Type:	0 2	Address: 0	
Network Parameters			
Interface:	PC/PPI cable(COM 1)		
Protocol:	PPI		
Mode:	11-bit		
Highest Station (HSA):			
🔲 Interface supports r	nultiple masters		
Transmission Rate			
Baud Rate:	9.6 kbps		
Search all baud rate	25	OK Cancel	

Phần Remote ban chon 2, đánh dấu kiểm vào ô Save setting with project. Double vào biểu

PC/PPI cable(PPI) Address: 0

tương truyền thông . Hộp thoại mới hiện ra bạn chọn Properties. Khi hôp thoai Properties hiên ra ở phần PPI ban chon Address : 0, Timer out : 1s, Transmission Rate : 9.6 kbps, Highest node Address : 31 và nhấn OK, ở phần Local Connection tùy vào cổng Com máy vi tính bạn kết nối cáp mà bạn chọn địa chỉ 1 hoặc 2 vv ... rồi bạn nhấn OK > OK.

et PG/PC Interface 🛛 🔀	
Access Path	
Access Point of the Application:	
Micro/WIN -> PC/PFY cable(PPI)	Properties - PC/DDI cable(DDI)
(Standard for Micro/WIN) Interface Parameter Assignment Used:	PPI Local Connection
PC/PPI cable(PPI) Properties	Station Parameters
We (None) PC/PPI cable(PPI) Copy	Address:
(Assigning Parameters to an PC/PPI cable for an PPI Network)	Network Parameters Advanced PPI Multiple Master Network
Add/Remove: Select	Transmission Rate: 9.6 kbps . Highest Node Address: 31 .
DK Cancel Help	DK Default Cancel Help

Trở về hộp thoại Communications bạn Double Click vào biểu tượng Double-Click to Refresh . Lúc này một biểu tượng cùng thông tin của loại PLC được kết nối sẽ hiện ra. Ví dụ

CPU 224 REL100B3 Address: 2

bạn chỉ việc nhắp chuột và OK.

Nếu những khai báo truyền thông kết nối đúng thì máy tính sẽ chấp nhận còn nếu không thì máy sẽ thông báo lỗi và bạn phải khai báo lại.

**Bước 7** Download chương trình đã viết trên máy vi tính xuống CPU của PLC. Click lên nút Download trên thanh công cụ.



Downio

Một hộp thoại sẽ hiện ra như sau :

Download	
Remote Address Remote PLC Type	2 CPU 224 REL 01.22
Blocks to Download Program Block Data Block System Block	
	OK Cancel

Bạn nhấn OK

Nếu công việc Download thành công thì màn hình sẽ xuất hiện thông báo :



Yes

No

Nếu không thì bạn phải kiểm tra lại chương trình hoặc kết nối truyền thông. Có thể bạn viết chương trình không đúng hoặc sự truyền thông giữa máy tính và PLC chưa được kết nối.

Chú ý khi bạn Download thì đảm bảo rằng PLC đang ở chế độ dừng (Stop).

**Bước 8** Chạy chương trình ứng dụng trên PLC và máy vi tính. Click lên nút RUN (hình nút Play) trên thanh công cụ.



Một hộp thoại xác nhận chế độ hiện hành của PLC hiện ra và bạn chọn Yes.

Chú ý lúc này PLC ở chế độ RUN hoặc TERM.

Lưu chương trình ứng dụng

Bước cuối cùng trong phần này là lưu ứng dụng khi chương trình bạn đã chạy thử và hài lòng với nó.

Bước 9 Click lên nút Save trên thanh công cụ. Một hộp thoại hiện ra :





**Bước 10** Thay đổi tên của Project1 là Bài1\*.mwp và click Save. Phần mở rộng "mwp" cho biết đây là một project file.

## Bài tập

## Bài 1 :

1. Tạo một chương trình ứng dụng như dưới đây.



- 2. Tên Network 1. Bạn nên viết tên không có dấu sau khi xóa tên mặc định của chương trình.
- 3. Khi thiết lập tên, thuộc tính và địa chỉ cho các lệnh bạn Click chuột phải vào lệnh và chọn Define Symbol. Trong hộp thoại Define Symbol hiện ra bạn điền tên (Name), địa chỉ (Address) và phần mô tả cho lệnh (Comment) và chọn OK.
- 4. Kiểm tra lỗi và chạy thử chương trình.
- 5. Lưu project với tên Bai 2.
- 6. Đóng Step 7 MicroWIN V3.2

*Bài 2 :* 

- 1. Khởi động Step 7 MicroWIN V3.2
- 2. Mở Bài đã lưu trong bài 1
- 3. Thay đổi Network thứ nhất và thêm 1 Network thứ hai như dưới đây



- 4. Kiểm tra lỗi và chạy thử chương trình
- 5. Lưu project sử dụng tên file cũ

Chú ý : Nếu bạn tạo 2 đường dẫn trong 1 Network như ở dưới là sai.



### 5. Xử lý chương trình

5.1. PLC thực hiện chương trình theo chu trình lặp, mỗi vòng lặp được gọi là vòng quét (Scan). Mỗi vòng quét bao gồm các bước sau:

**quét** I Nhập dữ liệu từ ngoại vi (các ngõ vào) vào bộ đệm ảo.

trình 🗆 Thực hiện chương trình.

- □ Truyền thông nội bộ và tự kiểm tra lỗi.
- □ Chuyển dữ liệu từ bộ đệm ảo ra ngoại vi (các ngõ ra).

Một chu kỳ quét (Scan cycle) được bắt đầu bằng việc đọc trạng thái của đầu vào và sau đó thực hiện chương trình. Scan cycle kết thúc bằng việc thay đổi trạng thái đầu ra. Trước khi bắt đầu 1 vòng quét tiếp theo PLC thực hiện nhiệm vụ bên trong và nhiệm vụ truyền thông. Chu trình thực hiện chương trình là chu trình lặp lại.



6. Phương pháp lập trình

### Bài 3: Các phép toán nhị phân của PLC Mã chương/Bài: MĐ22-03

Giới thiệu: Bài này trình bày các liên kết logic, các lệnh ghi /xóa giá trị tiếp điểm, nguyên lý làm việc của Timer - Counter và các dạng bài tập ứng dụng trong PLC S7-200.

#### Mục tiêu:

- Trình bày được các chức năng của RS, Timer, counter (bộ định thời, bộ đếm).

- Ứng dụng linh hoạt các chức năng của RS, Timer, counter trong các bài toán thực tế: Lập trình, kết nối, chạy thử...

- Rèn luyện đức tính tích cực, chủ động và sáng tạo

#### Nội dung chính:

1.	Các liên kết logic
2.	Các lệnh ghi / xóa giá trị cho tiếp điểm
3.	Timer
4.	Counter
4. 5.	Counter Bài tập ứng dụng

## 1. Các liên kết logic

1.1. Các lệnh vào/ra cơ bản	
Lệnh vào (Load)	<i>Load (LD):</i> Lệnh LD nạp giá trị logic của một tiếp điểm vào trong bít đầu tiên của ngăn xếp, các giá trị cũ còn lại trong ngăn xếp bị đẩy lùi xuống một bít.
Lệnh vào đảo (Load Not)	<i>Load Not (LDN):</i> Lệnh LDN nạp giá trị logic nghịch đảo của một tiếp điểm vào trong bít đầu tiên của ngăn xếp, các giá trị còn lại trong ngằn xếp bị đẩy lùi xuống một bít.
Lệnh ra (Output)	Output (=): Lệnh sao chép nội dung của bít đầu tiên trong ngăn xếp vào bít được chỉ định trong lệnh. Nội dung của ngăn xếp không bị thay đổi.
	Mô tả các lệnh vào/ra cơ bản bằng ngôn ngữ STL và ngôn ngữ LAD như sau:

ST	L	LAD	Giải trình		
LD	n	<b>n</b>	Lệnh LD nạp giá trị Logic của 1 tiếp điểm vào trong bit đầu tiên của ngăn xếp.		
LDN	n	n   /	Lệnh LDN nạp giá trị Logic nghịch đảo của 1 tiếp điểm vào trong bit đầu tiến của ngăn xếp.		
(=)	n		Cuộn dây ở đầu ra ở trạng thái kích thích khi có dòng điều khiển đi qua.		

Ví Mô tả việc thực hiện lệnh LD, LDN và OUTPUT bằng ngôn ngữ LAD và ngôn ngữ dụ STL

Sơ đồ mạch	Ngôn ngữ LAD	Giải trình	Ngôn ngữ STL
s н 🚫		Ở trạng thái bình thường cuộn dây Q0.0 được kích thích. Khi I0.0 = 1 thì Q0.0 bị kích thích.	LD I0,0 = Q0,0
s н 🚫		Ö trạng thái bình thườn,         thì cuộn dây Q0,0 đ         được kích thích, Khi I0,         = 1 thì Q0,0 ngừng b         kích thích,	g à LDN 10,0 = Q0,0

Các dạng khác nhau của lệnh LD, LDN cho LAD như sau:

Lệ	nh	Mô tả	Toán hạng		
LD	n	Tiếp điểm thường mở sẽ được đóng nếu n = 1.	n: I, Q, M, SM, T, C, V(bit)		
LDN	n	Tiếp điểm thường đóng sẽ mở khin = 1.			
LDI	n	Tiếp điểm thường mở sẽ đóng tức thời khin=1	n: I		
LDNI	n	Tiếp điểm thường đóng sẽ mở tức thời khin = 1			

Các dạng khác nhau của lệnh LD, LDN cho STL như sau:

Lệnh	Mô tả	Toán hạng
LD n	Lệnh nạp giá trị logic của điểm n vào bít đầu tiên trong ngãn xếp.	n (bít): I, Q, M, SM, T, C, V
LDN n	Lệnh nạp giá trị logic nghịch đảo của điểm n vào bít đầu tiên trong ngăn xếp.	
LDI n	Lệnh nạp tức thời giá trị logic của điểm n vào bít đầu tiên trong ngăn xếp.	n: I
LDNI n	Lệnh nạp tức thời giá trị logic nghịch đảo của điểm n vào bít đầu tiên trong ngăn xếp.	

Mô tả lệnh ra bằng ngôn ngữ LAD như sau:

LAD	Mô tả	Toán hạng
,,,,,,,,	Cuộn dây đầu ra ở trạng thái kích thích khi có dòng điều khiển đi qua.	n: I, Q, M, SM, T, C, V (bít)
	Cuộn dây đầu ra được kích thích tức thời khi có dòng điều khiển đi qua.	n: Q (bít)

## Mô tả lệnh ra bằng ngôn ngữ STL như sau:

STL	Mô tả	Toán hạng
= <b>n</b>	Lệnh = sao chép giá trị của đỉnh ngăn xếp tới tiếp điểm n được chỉ dẫn trong lệnh.	n: I, Q, M, SM, T, C, V (bít)
= I n	Lệnh = I (immediate) sao chép tức thời giá trị của đỉnh stack tới tiếp điểm n được chỉ dẫn trong lệnh.	n: Q (bít)

Các lệnh tiếp điểm đặc biệt

Các bit nhớ đặc	SM0.0	Luôn luôn có giá trị bằng 1
	SM0.1	Có giá trị logic bằng 1 ở vòng quét đầu tiên
biệt	SM0.4	Phát nhịp 60s (0 cho 30s đầu và 1 cho 30s sau).
	SM0.5	Phát nhịp 1s (0 cho 0,5s đầu và 1 cho 0,5s sau)
Các lệnh tiếp điểm đặc biệt	Có thể dùng xung (sườn x điểm đặc biệ có toán hạng hoặc hộp đầu	các tiếp điểm đặc biệt để phát hiện sự chuyển tiếp trạng thái của kung) và đảo trạng thái của dòng cung cấp. LAD dùng các tiếp t để tác động vào dòng cung cấp. Các tiếp điểm đặc biệt không riêng vì thế phải đặt chúng vào vị trí phía trước của cuộn dây ra.

Các lệnh tiếp điểm đặc biệt được biểu diễn trong LAD và ngôn ngữ STL:

LAD	Mô tả	STL
— I NOT — ( )	Lệnh đảo trạng thái dòng cung cấp. Nếu dòng cung cấp có tiếp điểm đảo thì nó ngất mạch, nếu không có tiếp điểm đảo thì nó thông mạch.	NOT
	Tiếp điểm chuyển đổi dương cho phép dòng cung cấp thông mạch trong l vòng quét khi sườn xung điểu khiển chuyển từ 0 lên 1.	EU
— м — < >	Tiếp điểm chuyển đổi âm cho phép dòng cung cấp thông mạch trong 1 vòng quét khi sườn xung điều khiển chuyển từ 1 xuống 0.	ED

Ví dụ:

Sử dụng lệnh tiếp điểm đặc biệt trong ngôn ngữ LAD và STL



#### Giản đổ thời gian tương ứng



#### 1.2. Các lệnh liên kết logic cơ bản (Các lệnh logic đại số Boolean)

Các lệnh tiếp điểm đại số Boolean cho phép tạo lập được các mạch logic (không có nhớ). Trong LAD các lệnh này được biểu diễn thông qua cấu trúc mạch, mắc nối tiếp hay song song các tiếp điểm thường đóng và các tiếp điểm thường mở. STL có thể sử dụng các lệnh A (And) và O (Or) cho các hàm hở hoặc các lệnh AN (And Not), ON (Or Not) cho các hàm kín.

Giá trị của ngăn xếp thay đổi phụ thuộc vào từng lệnh.

Lệnh	ST	ΓL	LAD	Giải t	trình
OR	0	n	Biểu diễn thống qua cấu trúc mạch:	Tiếp	điểm
			- Mắc song song các tiếp điểm thường hở	thường	hở
			với các tiếp điểm khác ta được lệnh OR:	được	mác
			I0,0	song	song
			VD:	vđi 1	tiếp
			IO.1	điểm k	hác,
			- Mắc nối tiếp các tiếp điểm thường hở	Tiếp	điểm
AND	А	n	với các tiếp điểm khác ta được lệnh AND:	thường	hở
			VD:	được m	nắc nối
				tiếp 🔹	vđi 1
				tiếp	điểm
				khác,	

ORNOT	ON	n	Biểu diễn thông qua cấu trúc mạch:	Tiếp	điểm
			- Mắc song song các tiếp điểm thường	thường	đống
			đóng với các tiếp điểm khác ta được lệnh	duợc	mắc
			OR: 10,0	song	song
			VD:	vđi 1	tiếp
				điểm kh	nác,
ANDNOT	AN	n	- Mắc nối tiếp các tiếp điểm thường đóng	Tiếp	điểm
			với các tiếp điểm khác ta được lệnh OR:	thường	hở
			VD:	duoc	mắc
				nối tiếp	vđi 1
				t' điểm l	khác,

1.3. Liên kết các cổng logic cơ bản.

Ví dụ Thực hiện lệnh Đại số Boolean A; AN; O nhớ và ON theo LAD và STL bằng các tiếp điểm :





Các lệnh logic đại số Boolean OR LOAD và AND LOAD

Lệnh	STL	LAD			
ORLOAD	OLD	Nối song song hai khối lệnh thành một nhóm ta lệnh			
		OLD I0.0 I0.1 Q0.0			
		I0.2 I0.3			
ANDLOAD	ALD	Nối nối tiếp hai khối lệnh thành một nhóm ta lệnh ALD			
		VD: 10.0 I0.1 Q0.0			
		IO.2 IO.3			



## LUYỆN TẬP VỚI CÁC MẠCH HỖN HỢP

## 1. Mạch hỗn hợp : AND trước OR







Xác lập vào/ra					
Ký hiệu	Địa chỉ				
S	10.1				
S1	10.2				
S2	10.3				
S3	10.4				
Н	Q0.0				

*Mô tả:* Ngõ ra Q0.0 chỉ có tín hiệu bằng 1 nếu các ngõ vào I0.1 và I0.2 của cổng AND có tín hiệu bằng "1" hoặc các ngõ vào I0.3 hoặc I0.4 có tín hiệu bằng "1".







Xác lập vào/ra				
Địa chỉ				
10.0				
10.1				
10.2				
10.3				
10.4				
10.5				
10.6				
10.7				
Q0.0				

*Mô tả:* Ngõ ra Q0.0 chỉ có tín hiệu bằng 1 nếu hai cổng AND trước OR có tín hiệu cùng bằng 1.







Xác lập vào/ra				
Ký hiệu	Địa chỉ			
S	I0.1			
S1	10.2			
S2	10.3			
S3	10.4			
S4	10.5			
Н	Q0.0			

*Mô tả:* Ngõ ra Q0.0 chỉ có tín hiệu bằng 1 nếu có ít nhất một ngõ vào của các cổng AND hoặc ngõ vào I0.5 có tín hiệu bằng 1.



Network 1 // Mach hon hop And truce Gr 2



#### 2. Mạch hỗn hợp : OR trước AND



Sơ đồ LAD




Xác lập vào/ra		
Ký hiệu	Địa chỉ	
S 1	I0.1	
S2	10.2	
S3	10.3	
S4	10.4	
Н	Q0.0	

*Mô tả:* Ngõ ra Q0.0 chỉ có tín hiệu bằng 1 nếu có ít nhất một ngõ vào của các cổng AND hoặc ngõ vào I0.5 có tín hiệu bằng 1.

Sơ đồ LAD



1.4. Bài tập ứng dụng:

1. Sử dụng các lệnh VÀO/RA; AND và OR để viết chương trình điều khiển sau :

a) Điều khiển động cơ : Khởi động (Start) và dừng (Stop).

**b**) Điều khiển động cơ đảo c`hiều thuận nghịch với yêu cầu: quay thuận, quay nghịch, dừng.

2. Sử dụng các lệnh VÀO/RA; AND, OR, SET, RESET để viết chương trình điều khiển sau :

a) Điều khiển động cơ : Khởi động (Start ) bằng lệnh Set và dừng (Stop ) bằng lệnh Reset.

b) Điều khiển động cơ đảo chiều thuận nghịch với yêu cầu như bài trên bằng lệnh Set và Reset.

## Các bước thiết lập chương trình :

- Xác định qui trình của hệ thống.

Bố trí vào/ra.

- Viết chương trình.

## 2. Các lệnh ghi/xóa giá trị cho tiếp điểm

2.1. Trong các chương trình được thiết kế theo cấu trúc liên kết, có thể có một số lượng lớn các cổng logic liên kết với nhau để thực hiện một chức năng nào đó. Với một số lượng các liên kết logic lớn như vậy thì việc kiểm tra tình trạng tín hiệu trong khi tìm lỗi sẽ gặp nhiều khó khăn. Để chương trình điều khiển có thể đơn giản và dễ nhìn hơn người ta sẽ chứa vào các kết quả trung gian sau khi thực hiện một số liên kết. Kết quả trung gian này được gọi là Memory bit. Một memory bit được xác định trạng thái tín hiệu tương tự như một ngõ ra. Trạng thái tín hiệu này được lưu giữ trong thiết bị tự động. Ký hiệu toán hạng cho Memory bit là M.

*Ví dụ :* Thiết kế một công tắc xung dòng điện bằng PLC. Sơ đồ mạch điện của công tắc như hình vẽ sau:



Nếu ta sử dụng các ngõ ra để nối với K1, K2, K3 thì ta cần phải có 4 ngõ ra. Bây giờ có thể chỉ sử dụng một ngõ ra duy nhất để nối với đèn H0 mà mạch vẫn không thay đổi bằng cách gán K1, K2, K3 tương ứng với M0.0, M0.1 và M0.2. Chương trình được viết trong PLC như sau:



2.2. Lệnh dùng để đóng và ngắt các điểm gián đoạn đã được thiết kế. Trong Lệnh ghi
(Set) LAD, logic điều khiển dòng điện đóng hoặc ngắt các cuộc dây đầu ra. Khi dòng điều khiển đến các cuộc dây thì các cuộn dây đóng hoặc mở các tiếp điểm (hoặc một dãy các tiếp điểm).

xóa
(Reset)
Trong STL, lệnh truyền trạng thái bít đầu của ngăn xếp đến các điểm thiết kế. Nếu bít này có giá trị =1, các lệnh S và R sẽ đóng ngắt tiếp điểm hoặc một dãy các tiếp điểm (giới hạn từ 1 đến 255). Nội dung của ngăn xếp không bị thay đổi bởi các lệnh này.

Mô tả các lệnh ghi/xóa bằng ngôn ngữ STL và ngôn ngữ LAD như sau:

Lệnh	STL	LAD	Giải trình
SET	S S-Bit n	S-Bit (S) n	Đóng một mảng gồm n các tiếp điểm liên tiếp kể từ S- Bit.
RESET	R S-Bit n	S-Bit ( R ) n	Ngất một mảng gồm n các tiếp điểm liên tiếp kể từ S- Bit.

2.3. Mô tả việc thực hiện lệnh SET, RESET trong ngôn ngữ LAD và ngôn ngữ Ví dụ STL

Sơ đồ mạch	Ngôn ngữ LAD	Ngôn ngữ STL
S H $\otimes$ H <sub>1</sub> $\otimes$ H <sub>2</sub> $\otimes$ H <sub>3</sub> $\otimes$	I0,0 Q0,0 Q0,1 Q0,1 (S) 4 Q0,5 (R) 4	LD I0.0 = Q0.0 S Q0.14 R Q0.54



Mô tả các lệnh ghi/xóa bằng ngôn ngữ LAD:

LAD	Mô tả	Toán hạng
S-Bit (S) n	Đóng một mảng gồm n các tiếp điểm kể từ S BIT	S BIT: I, Q, M, SM, T, C, V n(byte): IB, QB, MB, SMB, VB,AC,
S-Bit (R) n	Đóng một mảng gồm n các tiếp điểm kể từ S BIT. Nếu S BIT lại chỉ vào Timer hoặc Counter thì lệnh sẽ xóa bít đầu ra của Timer/Counter đó.	Hằng số, *VD, *AC
S-Bit (SI) n	Đóng tức thời một mảng gồm n các tiếp điểm kể từ S BIT	S BIT: Q N(byte): IB, QB, MB, SMB, VB,AC, Hằng
S-Bit C RI > n	Ngất tức thời một mảng gồm n các tiếp điểm kể từ địa chỉ SBIT	số, *VD, *AC

Mô tả các lệnh ghi/xóa bằng ngôn ngữ STL:

STL	Mô tả	Toán hạng
S SBIT n	Ghi giá trị logic vào một mảng gồm n bít kể từ địa chỉ S BIT	S BIT: I, Q, M, SM, T, C, V (bit)
R SBIT n	Xóa một mảng gồm n bít kể từ địa chỉ S BIT. Nếu S BIT lại chỉ vào Timer hoặc Counter thì lệnh sẽ xóa bít đầu ra của Timer / Counter.	n: IB, QB, MB, SMB, VB (byte) AC, Hằng số, *VD, *AC
SI SBIT	Ghi tức thời giá trị logic 1 vào một mảng gồm n bít kể từ địa chỉ S BIT	S BIT: Q (bit)
RI SBIT	Xóa tức thời một mảng gồm n bít kể từ địa chỉ S BIT	n: IB, QB, MB, SMB, VB (byte) (byte) AC, Hằng số, *VD, *AC

#### 3. Timer

3.1. Bộ Timer là bộ tạo thời gian trễ giữa tín hiệu vào và tín hiệu ra nên trong điều định thời khiển vẫn gọi là khâu trễ. S7 - 200 có 64 Timer (với CPU 212) hoặc 128 (Timer) Timer (với CPU 214) được chia làm 2 loại khác nhau, đó là :

Timer tạo thời gian trễ không có nhớ (ON Delay Timer) ký hiệu TON.

Timer tạo thời gian trễ có nhớ (Retentive ON Delay Timer) ký hiệu TONR.

Cả hai loại Timer này đều có 3 loại với 3 độ phân giải khác nhau : 1ms; 10ms và 100ms.

Thời gian trễ  $\tau$ : được tạo ra chính là tích của độ phân giải của Timer được chọn và giá trị đặt trước cho Timer, ví dụ 1 bộ Timer có độ phân giải 10ms, giá trị đặt trước cho Timer là 30 thì thời gian trễ là :  $\tau = 10$ ms\*30 = 300ms.

Cả hai loại Timer TON và TONR cùng bắt đầu tạo thời gian trễ kể từ thời điểm có sườn lên ở tín hiệu đầu vào. Tức là khi tín hiệu đầu vào chuyển trạng thái logic từ từ logic 0 lên logic 1, được gọi là thời điểm Timer được kích thì Timer bắt đầu tạo thời gian trễ. Khi giá trị đếm tức thời của Timer > = thời gian trễ (giá trị đặt trước) thì đầu ra của Timer từ logic 0 chuyển lên logic 1. Lúc này các tiếp điểm thường đóng mở chậm của Timer mở ra còn các tiếp điểm thường mở đóng chậm của Timer đóng lại.

Các loại Timer của S7 - 200 (đối với CPU 212 và CPU 214) chia theo TON, TONR và độ phân giải bao gồm :

Lệnh	Độ phân giải	Giá trị cực đại	CPU 212	CPU 214
	lms	32,767	T 32	T32, T96
TON	10ms	327,67	T32 4T36	T 32 4T 36, T 97 4T 100
i	100ms	3276,7	T37 4T63	T37 4T63, T101 4T127
	1ms	32,767	TO	T0, T64
TONR	10ms	327,67	T14T4	T14T4, T654T68
	100ms	3276,7	T5 4 T 3 1	T5 4 T31, T69 4 T95

Cú pháp khai báo sử dụng Timer trong LAD và STL:

Ngôn ngữ LAD	Mô tả	Ngôn ngữ STL
TON — Txx IN PT	Khai báo Timer số hiệu xx kiểu TON để tạo thời gian trễ tính từ khi đầu vào IN được kích. Nếu như giá trị đếm tức thời > = giá trị đặt trước thì T-bit có giá trị logic = 1. Reset bằng lệnh Reset hoặc bằng giá trị logic 0 tại đầu vào IN	TON Txx n
TONR – Txx IN PT	Khai báo Timer số hiệu xx kiểu TONR để tạo thời gian trể tính từ khi đâu vào IN được kích, Nếu như giả trị đếm tức thời > = giá trị đặt trước thì T-bit có giá trị logic = 1, Reset bằng lệnh Reset cho T- bit,	TONR Txx n

**Chú ý** : *Khi sử dụng Timer kiểu TONR, giá trị đếm tức thời được lưu lại và không bị thay* đổi trong khoảng thời gian khi tín hiệu đầu vào có logic 0.

Ví dụ Về cách sử dụng Timer kiểu TON

Ngôn ngữ LAD	Ngôn ngữ STL	Giải trình
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	LD I0. TON T37 100 LD T37 = Q0.0	Khi ngồ vào I0,0 đóng mạch, Timer T37 bắt đầu đếm, Nếu giá trị đếm tức thời >= giá trị đặt trước PT (cụ thể PT = 100*100ms = 10s) thì bit thời gian hoạt động(T37 có giá trị logic 1)

Giản đồ thời gian tương ứng





Ngôn ngữ LAD	Ngôn ngữ STL	Giải trình
$ \begin{array}{c ccccc} I0,0 & TONR - TS \\ \hline I0,0 & PT \\ TS & Q0,0 \\ \hline I0,1 & TS \\ \hline I0,1 & TS \\ \hline I0,1 & I \\ I \\ \end{array} $	LD I0,0 TONR T37 100 LD T5 = Q0,0 LD I0,1 S T5 1	Khi ngồ vào IO,O đóng mạch, Timer T5 bắt đầu đếm. Nếu giá trị đếm tức thời >= giá trị đặt trước PT (cụ thể PT = 100*100ms = 10s) thì bit thời gian hoạt động(T5 có giá trị logic 1). Nếu IO,O ngắt mạch mà giá trị đếm tức thời chưa >= giá trị đặt trước, sau đó IO,O đóng mạch lại thì giá trị lấn đếm lần 1 + giá trị đếm lần 2 + giá trị đếm lần 3 + >= giá trị giá trị PT thì T5 có tín hiệu 1, T5 được reset khi IO,1 đóng mạch(Lúc này giá trị đếm tức thời trở về trạng thái ban đầu kể cả các tiếp điểm cũng trở về trạng thái ban đầu)





## 3.3. Bài tập ứng dụng Timer

**Bài 1** : sử dụng lệnh Vào/Ra, Set, Reset và TON viết chương trình điều khiển động cơ với yêu cầu sau:

a) Nhấn nút Khởi động (Start) sau thời gian 5s động cơ chạy. Muốn dừng nhấn nút Dừng (Stop).

b) Nhấn nút Khởi động (Start) sau thời gian 5s động cơ chạy. Nhấn nút Dừng (Stop) sau thời gian 5s động cơ mới ngừng chạy.

Bài 2 : Dùng các lệnh đã học viết chương trình điều khiển động cơ với yêu cầu:

a) Nhấn nút Khởi động (Start) động cơ 1 chạy. Sau thời gian 5s động cơ 1 ngừng chạy
- đồng thời động cơ 2 chạy. Muốn dừng nhấn nút Stop.

Nhấn nút Khởi động (Start) động cơ 1 chạy. Sau thời gian 5s động cơ 2 chạy. Nhấn nút Stop động cơ 2 dừng trước, sau thời gian 5s động cơ 1 mới ngừng chạy.

## 4. Counter

4.1. Bộ điều khiển lập trình loại SIMATIC S7 - 200 có 2 loại chức năng đếm là đếm lên (CTU) và vừa đếm lên vừa đếm xuống (STUD).

Bộ đếm lên (hay đếm tiến : Counter Up) : CTU đếm số sườn lên của tín hiệu logic đầu vào, tức là đếm số lần thay đổi trạng thái logic từ 0 lên 1 của tín hiệu đầu vào. Cứ mỗi lần tín hiệu đầu vào chuyển trạng thái từ 0 lên 1 thì giá trị đếm tức thời của bộ đếm tăng lên 1. Nếu như giá trị đếm tức thời >= giá trị đặt trước (PV) thì bộ đếm sẽ báo ra ngoài bằng cách cho cổng ra (bit đặc biệt) lên 1. (Khi giá trị đếm tức thời < hơn giá trị đặt trước thì cổng ra bằng 0).

Miền giá trị của miền đếm tiến CTU : bộ đếm tiến có thể đếm từ 0 đến giá trị cao nhất là 32767.

Reset Counter : bộ đếm tiến CTU có giá trị tức thời trở về 0 (giá trị ban đầu) nếu chân Reset được kích (có giá trị logic = 1) hoặc Reset bằng lệnh Reset S-bit.

Ngôn ngữ LAD	Mô tả	Ngôn ngữ STL
CTU — Cxx — CU PV R	CU : chân nhận xung đếm, PV : giá trị đặt trước, R : chân Reset, Cxx : vùng địa chỉ của bộ đếm, xx : từ 0 4 47; 80 4127	CTU Cxx n

Lệnh khai báo sử dụng bộ đếm trong ngôn ngữ LAD và ngôn ngữ STL.

#### STL. Giản đồ thời gian tương ứng I.0 (CU) Q0.0 (Output) IO.0 (Reset) Ngôn ngữ LAD Ngôn ngữ STL CTU - C47 LD I0.0 I0,0 LD IO.1 CU CTU C47 5 PV 5 -IO.1 LD C47 R = Q0,0 Q0.0 C47 )

Minh họa về sử dụng bộ đếm CTU trong ngôn ngữ LAD và ngôn ngữ

4.2. Bộ Bộ đếm lên/xuống - *Counter up - Down* (CTUD) đếm tiến khi gặp sườn lên của xung vào cổng đếm lên ký hiệu là CU và đếm xuống khi gặp sườn lên của xung vào cổng đếm xuống ký hiệu là CD.

Cũng giống như bộ đếm tiến CTU, bộ đếm lên/xuống cũng có cổng ra (bit đặc biệt) lên 1 khi giá trị đếm tức thời >= giá trị đặt trước.

Miền giá trị của bộ đếm lên/xuống (CTUD) là – 32767 4 +32767.

Reset Counter : - khi đầu vào logic của chân R có giá trị logic 1.

- Reset bằng lệnh Reset cho S-bit.

Lệnh khai báo sử dụng bộ đếm CTUD trong ngôn ngữ LAD và ngôn ngữ STL.

Ngôn ngữ LAD	Mô tẩ	Ngôn ngữ STL
CTUD _ Cxx _ CU PV CD R	CU : cổng vào đếm tiến CD : cổng vào đếm lùi PV : giá trị đặt trước. R : chân Reset Cxx : vùng địa chỉ bộ đếm Cxx : từ 48 4 79	CTUD Cxx n

Minh họa về sử dụng bộ đếm CTU trong ngôn ngữ LAD và ngôn ngữ STL.

Ngôn ngữ LAD	Ngôn ngữ STL
I0.3 CTUD _ C48	LD 10,3 LD 10,2
10.2 5 - PV	LD IO.1
	CTUD C48 5
IO.1	LD C48
	= Q0,0
C48 Q0.0	

## 4.3. Bài tập ứng dụng bộ đếm

#### Bài tập thực hành (CTU)

**Bài 1** : Sử dụng lệnh vào/ra ; set, reset; TON và Counter viết chương trình điều khiển sau : Nhấn nút Khởi động (Start) đèn sáng nhấp nháy cứ 2s sáng và 2s tắt. Sau chu kỳ 10 lần sáng/tắt thì dừng lại. Muốn dừng khẩn cấp thì nhấn nút Dừng(Stop).

Bài 2 : Viết chương trình điều khiển hoạt động đóng gói. Qui trình hoạt động như sau :

Nhấn nút Khởi động (Start) bộ truyền tải táo bắt đầu vận chuyển táo vào hộp. Phần cảm biến sẽ đếm đến 10 trái táo thì bộ truyền tải táo sẽ dừng lại và bộ truyền tải hộp đựng táo sẽ bắt đầu di chuyển. Bộ đếm sẽ được reset lại qua bộ cảm biến. Quá trình sẽ tiếp tục mãi cho đến khi nhấn nút nhấn Dừng(Stop).

## Bài tập thực hành (CTUD)

Viết chương trình điều khiển như sau :

Một Garage xe có 100 chỗ đậu xe. Ở ngõ vào có 2 đèn báo: đèn đỏ báo hiệu Garage đã đầy và đèn xanh báo Garage vẫn còn trống. Cửa vào Garage và cửa ra của Garage khác nhau và chỉ cho phép chỉ 1 xe lọt qua.

## 5. Các bài tập ứng dụng

# 5.1 Mạch điều khiển tuần tự cưỡng bức

#### Mô tả hoạt động:

Khi nhấn S2 cuộn dây K1 có điện và nó tự duy trì qua các tiếp điểm thường mở, đèn H1 sáng. Khi nhấn S3 cuộn dây K2 có điện và tự duy trì, đèn H2 sáng. Nhấn S1 thì mạch bị cắt hoàn toàn khỏi nguồn điện.

K2 chỉ có thể đóng mạch được nếu trước đó K1 đã được đóng.



## 5.2 Mạch lựa chọn 1 trong 3 khả năng

## Mô tả hoạt động:

Van Y1 có tín hiệu bằng 1, nếu chỉ cần 1 ngõ vào có tín hiệu bằng 1



## 5.3 Mạch lựa chọn 2 trong 3 khả năng

## Mô tả hoạt động:

Van Y1 có tín hiệu bằng 1, nếu chỉ cần 1 ngõ vào có tín hiệu bằng 1





5.5 Điều khiển
băng tải
Công tắc S0 dùng để khởi động cho thiết bị và đèn H0 chỉ báo chế độ làm việc. Khi nhấn S1 động cơ M1 khởi động kéo băng tải và than đá trong thùng chứa được vận chuyển theo băng tải. Nhấn S2 thì băng tải ngừng lại. Khi động cơ kéo băng tải bị quá tải, nó sẽ được cắt khỏi nguồn qua bộ bảo vệ quá dòng F2.





Xác định ngõ vào/ra			
Ký hiệu	Toán hạng	Mô tả	
S0	I0.0	Công tắc thường mở	
S1	IO.1	Nút nhấn thường mở	
S2	I0.2	Nút nhấn thường mở	
F2	I0.3	Tiếp điểm thường đóng	
K1	Q0.0	Khởi động từ	
H0	Q0.2	Đèn bảo	

#### Million vul

## Hãy thực hiện mạch theo các yêu cầu sau:

- 1/ Vẽ sơ đồ LAD.
- 2/ Viết bảng mô tả câu lệnh.
- 3/ Viết và thứ chương trình.

#### 5.6 Xe rót vật liệu – bể chứa

Công tắc S0 dùng để khởi động cho thiết bị và đèn H0 chỉ báo chế độ làm việc. Khi động cơ M1 có sự cố thì đèn H0 chớp tắt với tần số 1Kz. Khi nhấn S1, động cơ M1 có điện và xe di chuyển lên phía trên – đèn H1 chớp tắt với tần số 1 Hz. Khi xe lên đến vị trí trên cùng đụng phải công tắc hành trình S4 thì động cơ bị ngắt mạch. Nhấn S2 động cơ có điện trở lại và xe di chuyển xuống phía dưới – đèn H2 chớp tắt với tần số 1 Kz. Khi xe đến vị trí cuối cùng đụng phải công tắc hành trình S3 thì động cơ bị ngắt mạch. Quá trình được khởi động lập lại như ban đầu.



#### Kết nối với PLC



Hãy thực hiện mạch theo các yêu

- câu sau:
- 1/ Vẽ sơ đồ LAD.
- 2/ Viết bảng mô tả câu lệnh.
- 3/ Viết và thứ chương trình.

Xác định ngõ vào/ra		
Ký hiệu	Toản hạng	Mô tả
<b>S</b> 0	10.0	Công tắc thưởng mở
<b>S1</b>	10.1	Nút nhấn thường mở
S2	10.2	Nút nhấn thường mở
<b>S</b> 3	10.3	Công tắc hành trình
<b>S4</b>	10.4	Công tắc hành trình
Q1	10.5	Bảo vệ động cơ
K1	Q0.1	Khởi động từ chạy phải
K2	Q0.3	Khởi động từ chạy trái
H0	Q0.0	Đẻn báo
H1	Q0.2	Đẻn báo
H2	Q0.4	Đèn báo

## 5.7 Thang máy xây dựng

Khi nhấn nút nhấn nâng gầu sẽ chạy lên đến công tắc giới hạn trên thì gàu dừng lại. Khi nhấn nút nhấn hạ thì gàu sẽ hạ xuống đến công tắc giới hạn dưới thì gàu dừng lại. Trong khi đang di chuyển nếu nhấn nút dừng thì gàu dừng lại và sau đó có thể nâng gàu lên hay hạ gàu xuống theo mong muốn.

Các trạng thái nâng lên, hạ xuống hoặc dừng đều được thông báo bằng đèn báo.



	Х	ác định ngõ vào/ra
Ký hiệu	Toán hạng	Mô tả
<b>S</b> 0	10.0	Nâng, thường mở
S1	I0.1	Hạ, thường mở
S2	10.2	Dừng, thường đóng
<b>S</b> 3	10.3	Giới hạn trên, thường đóng
S4	10.4	Giới hạn dưới, thường đóng
K1	Q0.0	Gàu chạy lên
K2	Q0.1	Gàu chạy xuống
H0	Q0.2	Đèn báo nâng
H1	Q0.3	Đèn bảo hạ
H2	Q0.4	Đèn báo dùng

## Schiem yu:

Hãy thực hiện mạch theo các yêu cầu sau:

1/ Vẽ sơ đồ LAD.

2/ Viết bảng mô tả câu lệnh.

3/ Viết và thứ chương trình.

6. Lệnh nhảy và lệnh gọi chương trình con.

Lệnh nhảy chương trình con thực hiện theo thứ tự từ trên xuống dưới trong một vòng quét. Lệnh điều khiển chương trình cho phép thay đổi thứ tự thực hiện lệnh. Chúng cho phép chuyển thứ tự thực hiện, đáng lẽ ra là lệnh tiếp theo, tới một lệnh bất cứ nào khác của chương trình, trong đó nơi điều khiển chuyển đến phải được đánh dấu trước bằng một nhãn, chỉ, đích. Thuộc nhóm lệnh điều khiển chương trình gồm: lệnh nhảy, lệnh gọi chương trình con, nhãn chỉ đích, hay gọi đơn giản là nhãn, phải được đánh dấu trước khi thực hiện lệnh nhảy hay lệnh gọi chương trình con.

Việc đặt nhãn cho lệnh nhảy phải nằm trong chương trình. Nhãn của chương trình con, hoặc của chương trình xử lý ngắt được khai báo ở đầu chương trình. Không thể dùng lệnh nhảy JMP để chuyển điều khiển từ chương trình chính vào một nhãn bất kỳ trong chương trình con hoặc trong chương trình xử lý ngắt. Tương tự như vậy cũng không thể từ một chương trình con hay chương trình xử lý ngắt nhảy vào bất cứ một nhãn nào nằm ngoài các chương trình đó.

Lệnh gọi chương trình con là lệnh chuyển điều khiển đến chương trình con. Khi chương trình con thực hiện xong các phép tính của mình thì việc điều khiển lại được chuyển trở về lệnh tiếp theo trong chương trình chính nằm ngay sau lệnh gọi chương trình con. Từ một chương trình con có thể gọi được một chương trình con khác trong nó, có thể gọi như vậy nhiều nhất là 8 lần trong S7-200. Nói chung (trong một chương trình con có lệnh gọi đến chính nó) về nguyên tắc không bị cấm song phải để ý đến giới hạn trên.

Nếu lệnh nhảy hay lệnh gọi chương trình con được thực hiện thì đỉnh ngăn xếp luôn có giá trị logic 1. Bởi vậy trong chương trình con các lệnh có điều kiện được thực hiện như các lệnh không điều kiện. Sau các lệnh LBL (đặt nhãn) và SBR, lệnh LD trong STL sẽ bị vô hiệu hóa.

Khi một chương trình con được gọi, toàn bộ nội dung của ngăn xếp sẽ được cất đi,

đỉnh của ngăn xếp nhận giá trị logic mới là 1, các bít khác của ngăn xếp nhận giá trị logic 0 và điều khiển được chuyển đến chương trình con đã được gọi. Khi thực hiện xong chương trình con và trước khi điều khiển được chuyển trở lại chương trình đã gọi nó, nội dung ngăn xếp đã được cất giữ trước đó sẽ được chuyển trở lại ngăn xếp.

Nội dung của thanh ghi AC không được cất giữ khi gọi chương trình con, nhưng khi một chương trình xử lý ngắt được gọi, nội dung của thanh ghi AC sẽ được cất giữ trước khi thực hiện chương trình xử lý ngắt và nạp lại khi chương trình xử lý ngắt đã được thực hiện xong. Bởi vậy chương trình xử lý ngắt có thể tự do sử dụng bốn thanh ghi AC của S7-200.

JMP. Lệnh nhảy JMP và lệnh gọi chương trình con SBR cho phép chuyển điều khiển từ vị trí này đến vị trí khác trong chương trình. Cú pháp của lệnh nhảy và lệnh gọi chương trình con trong LAD và STL đều có toán hạng là nhãn chỉ đích (nơi nhảy đến, nơi chứa chương trình con).

Lệnh nhảy, lệnh gọi chương trình con, lệnh khai báo nhãn và lệnh thoát khỏi chương trình con được biểu diễn trong LAD và trong STL như sau:

LAD	STL	Mô tả	Toán hạng	
n (JMP)	JMP Kn	Lệnh nhẵy thực hiện việc chuyển điều khiển đến nhần n trong một chương trình.	n: 0 ÷ 255	
LBL:n	LBL Kn	Lệnh khai báo nhần n trong một chường trình		
n (CALL)	CALLKn	Lệnh gọi chương trình con, thực hiện phép chuyển điều khiển đến chương trình con có nhàn là n.	n: 0 ÷ 255	
SBR: n	SBR Kn	Lệnh gần nhần n cho một chương trình con.		
n ——(CRET)	CRET	Lệnh trở về chương trình đà gọi chương trình con có điều kiện (bít đầu tiền của ngăn xếp có giá trị logic bằng 1).	Không có	
n (RET)	RET	Lệnh trở về chương trình đã gọi chương trình con không điều kiện.		

## Bài 4: Các phép toán số của PLC Mã chương/Bài: MĐ22-04

**Giới thiệu:** Bài này trình bày nguyên lý hoạt động các phép toán số của PLC S7-200 theo nội dung đã học. Cách kiểm tra, xử lý chức năng toán số của PLC S7-200. **Mục tiêu:** 

- Trình bày được các phép toán so sánh, các phép toán số.
- Vận dụng các bài toán vào thực tế: Lập trình, kết nối, chạy thử...
- Rèn luyện đức tính tích cực, chủ động và sáng tạo

Nội dung chính:

1.	Chức năng truyền dẫn
2.	Chức năng so sánh
3.	Chức năng dịch chuyển
4.	Chức năng chuyển đổi
5.	Chức năng toán học
6.	Đồng hồ thời gian thực

## 1. Chức năng truyền dẫn

1.1. Các
lệnh dịch chuyển nội dung ô nhớ thực hiện việc di chuyển hoặc sao chép số liệu từ vùng này sang vùng khác trong bộ nhớ.
Trong LAD hay trong STL lệnh dịch chuyển thực hiện việc di chuyển hay sao chép nội dung của một byte, một từ đơn, một từ kép hoặc một giá trị thực từ vùng này sang vùng khác trong bộ nhớ.

Lệnh Là lệnh sao chép nội dung của byte IN sang byte OUT.

## MOV\_B Cú pháp dùng lệnh MOV\_B trong LAD hay MOV\_B trong STL như sau:

LAD		STL	
MOV B EN IN OUT	MOVB	IN	OUT

**MOV\_W** *Cú pháp dùng lệnh MOV\_B trong LAD hay MOV\_B trong STL như sau:* 

LAD		STL	
MOV W EN IN OUT	MOVW	IN	OUT

Lệnh Là lệnh sao chép nội dung của từ kép IN sang từ kép OUT.

MOV\_ DW Cú pháp dùng lệnh MOV\_B trong LAD hay MOV\_B trong STL như sau:

LAD	STL
MOV DW EN IN OUT	MOVD IN OUT

1.2.

Truyền một vùng dữ liệu

Lệnh Là lệnh sao chép một số thực từ IN (4 byte) sang OUT (4 byte).

MOV\_R

Cú pháp dùng lệnh MOV\_B trong LAD hay MOV\_B trong STL như sau:

LAD		STL	
MOV R EN IN OUT	MOVR	IN	OUT

LệnhLà lệnh trao đổi nội dung của Byte thấp và Byte cao trong nội dung từ đơn INSWAPCú pháp dùng lệnh SWAP trong LAD hay trong STL như sau:

LAD	STL
SWAP EN IN OUT	SWAP IN

#### 2. Chức năng so sánh

Khi lập trình, nếu có các quyết định về điều khiển được thực hiện dựa trên kết quả của việc so sánh thì có thể sử dụng lệnh so sánh cho byte, từ hay từ kép của S7-200.

LAD sử dụng lệnh so sánh để so sánh các giá trị của byte, từ và từ kép (giá trị thực hoặc nguyên). Những lệnh so sánh thường là so sánh nhỏ hơn hoặc bằng (< =); so sánh bằng (==) và so sánh lớn hơn hoặc bằng (> =).

Khi so sánh giá trị của byte thì không cần phải để ý đến dấu của toán hạng. Ngược lại khi so sánh các từ hoặc từ kép với nhau thì phải để ý đến dấu của toán hạng, ngược lại khi so sánh các từ hoặc từ kép với nhau thì phải để ý đến dấu của toán hạng là bít cao nhất trong từ hoặc từ kép.

Biểu diễn các lệnh so sánh trong LAD:

LAD	Mô tả	Toán hạng
$\begin{array}{c c} \mathbf{n1} & \mathbf{n2} \\ \mathbf{n1} & = \mathbf{B} &   \\ \mathbf{n1} & \mathbf{n2} \\ \mathbf{n1} & = \mathbf{I} &   \\ \mathbf{n1} & = \mathbf{D} &   \\ \mathbf{n1} & = \mathbf{D} &   \\ \mathbf{n1} & = \mathbf{R} &   \\ \mathbf{n1} & = \mathbf{n1} &   \\ \mathbf{n1} & =$	Tiếp điểm đóng khi n1 = n2 B = Byte I = Integer D = Double Integer R = Real	n1,n2 (byte) : VB , IB , QB, MB, SMB , AC , Const , *VD*, AC
$\begin{array}{c c} n1 & n2 \\ \hline & 1 & B \\ \hline & n1 & n2 \\ \hline & - & 1 & P \\ \hline & - & 1 & P \\ \hline & 1 & - & 1 \\ \hline & - & 1 & P \\ \hline$	Tiếp điểm đóng khi N1 > n2 B = Byte I = Integer D = Double Integer R = Real	n1,n2 (từ): VW, T, C, IW, QW, MW, SMW,AC, AIW, Hầng số, *VD, *AC

$\begin{array}{c c c c c c c c c c c c c c c c c c c $	n1, n2 (từ kép): VD, ID, QD, MD, SMD, AC, HC, Hằng số, *VD, *AC
--	---

Trong STL, những lệnh so sánh thực hiện phép so sánh byte, từ và từ kép. Căn cứ vào kiểu so sánh (<=, ==, >=), kết quả của phép so sánh có giá trị bằng 0 (nếu đúng) hoặc 1 (nếu sai) nên có thề sử dụng kết hợp cùng với các lệnh gogic LD, A, O. Để tạo ra được các phép so sánh mà S7-200 không có lệnh so sánh tương ứng như: so sánh không bằng nhau (< >), so sánh nhỏ hơn (>), có thể tạo ra được nhờ dùng kết hợp lệnh NOT với các lệnh đã có (==, >=, <=). Ví dụ sau mô tả việc thực hiện pháp so sánh không bằng nhau (< >) giữa nội dung của từ V>W100 và hằng số 50 bằng cách sử dụng kết hợp phép so sánh bằng nhau LDW = và lệnh đảo NOT.

LDB =, LDW =

LDD =, LDR =

Lệnh kiểm tra tính bằng nhau của nội dung 2 byte, từ, từ kép, hoặc số thực. Trong trường hợp phép so sánh cho kết quả đúng, bít đầu tiên trong ngăn xếp sẽ có giá trị logic bằng 1.

LDB <= , LDW <= LDD <= , LDR <=

Lệnh so sánh nội dung của byte, từ, từ kép hoặc số thực thứ nhất có nhỏ hơn hoặc bằng nội dung của byte, từ, từ kép hoặc số thực thứ hai hay không. Trong trường hợp phép so sánh cho kết quả đúng, bít đầu tiên trong ngăn xếp có giá trị logic bằng 1.

LDB > =, LDW > =

```
LDD > =, LDR > =
```

Lệnh so sánh nội dung của byte, từ, từ kép hoặc số thực thứ nhất có lớn hơn hoặc bằng nội dung của byte, từ, từ kép hoặc số thực thứ hai hay không. Trong trường hợp phép so sánh cho kết quả đúng, bít đầu tiên trong ngăn xếp có giá trị logic bằng 1.

AB =, AW =

AD =, AR =

Lệnh kiểm tra tính bằng nhau của nội dung 2 byte, từ, từ kép, hoặc số thực. Trong trường hợp phép so sánh cho kết quả đúng, sẽ thực hiện phép tính logic And giữa bít đầu tiên trong ngăn xếp với giá trị logic.

AB < =, AW < = AD < =, AR < =

Lệnh so sánh nội dung của byte, từ, từ kép hoặc số thực thứ nhất có nhỏ hơn hoặc bằng nội dung của byte, từ, từ kép hoặc số thực thứ hai hay không. Trong trường hợp phép so sánh cho kết quả đúng, sẽ thực hiện phép tính logic AND giữa bít đầu tiên trong ngăn xếp với giá trị logic 1

STL	Mô tả	Toán hạng
$LDB = n_1 n_2$	Lệnh thực hiện phép tính logic Load And hoặc Or giữa	
$AB = n_1 n_2$	giá trị logic 1 với nội dung đỉnh ngãn xếp khi nội dung 2 byte n <sub>1</sub> và n <sub>2</sub> thỏa mãn n <sub>1</sub> = n <sub>2</sub>	$n_1$ , $n_2$ (byte): VB, 1B, QB, MB, SMB, AC, hằng số, *VD, *AC
$OB = n_1 n_2$		
$LDB > = n_1 n_2$	Lệnh thực hiện phép tính logic Load, And hoặc Or giữa	
AB >= $n_1 n_2$	giá trị logic 1 với nội dung đỉnh ngăn xếp khi nội dung 2 byte n <sub>1</sub> và n <sub>2</sub> thỏa mãn n <sub>1</sub> > =	
OB >= $n_1 n_2$	n <sub>2</sub>	

Biểu diển lệnh so sánh trong STL:

$LDB \le n_1  n_2$ $AB \ \le n_1  n_2$ $OB \ \le n_1  n_2$	Lệnh thực hiện phép tính logic Load, And hoặc Or giữa giá trị logic l với nội dung đỉnh ngăn xếp khi nội dung 2 byte n <sub>1</sub> và n <sub>2</sub> thỏa mẫn n <sub>1</sub> < = n <sub>2</sub>	
$LDW = n_1  n_2$ $AW = n_1  n_2$ $OW = n_1  n_2$	Lệnh thực hiện phép tính logic Load, And hoặc Or giữa giả trị logic l với nội dung đỉnh ngăn xếp khi nội dung 2 từ n <sub>1</sub> và n <sub>2</sub> thỏa mẫn n <sub>1</sub> = n <sub>2</sub>	n <sub>1</sub> ,n <sub>2</sub> (từ):VW, T, C, QW, MW, SMW, AC, AIW, hằng số, *VD, *AC
$LDW \ge n_1  n_2$ $AW \ge n_1  n_2$ $OW \ge n_1  n_2$	Lệnh thực hiện phép tính logic Load, And hoặc Or giữa giá trị logic l với nội dung đỉnh ngăn xếp khi nội dung 2 từ n <sub>l</sub> văn <sub>2</sub> thỏa mãn n <sub>l</sub> > = n <sub>2</sub>	
$LDW \le n_1  n_2$ $AW \ \le n_1  n_2$ $OW \ \le n_1  n_2$	Lệnh thực hiện phép tính logic Load, And hoặc Or giữa giá trị logic l với nội dung đỉnh ngăn xếp khi nội dung 2 từ n <sub>1</sub> và n <sub>2</sub> thỏa mãn n <sub>1</sub> < = n <sub>2</sub>	

10		
$LDD = n_1  n_2$	Lệnh thực hiện phép tính logic Load , And hoặc Or giữa	п <sub>1</sub> , п <sub>2</sub> (từ kếp) :VD,
$AD = n_1 n_2$	giá trị logic l với nội dung đỉnh ngăn xếp khi nội dung 2 từ kép nị và nº thỏa mắn nị =	ID, QD, MD, SMD, AC, HC, hằng số, *VD, *AC
$OD = n_1 n_2$	n <sub>2</sub>	
$LDD > = n_1 n_2$	Lệnh thực hiện phép tính logic Load , And hoặc Or giữa	
$AD > = n_1 n_2$	giá trị logic l với nội dung đỉnh ngãn xếp khi nội dung 2	
$OD >= n_1 n_2$	$= n_2$	
$LDD <= n_1 n_2$	Lệnh thực hiện phép tinh logic Load, And hoặc Or giữa	
$AD \leq n_1 n_2$	giá trị logic l với nội dung đỉnh ngăn xếp khi nội dung 2	
OD <= n1 n2	tu kep $n_1$ va $n_2$ thoa man $n_1 <= n_2$	
$LDR = n_1  n_2$	Lênh thực hiện phép tính logic Load , And hoặc Or giữa	π, ma(từ kén):VD ID
$AR = n_1 n_2$	giá trị logic l với nội dung đỉnh ngăn xếp nếu hai số thực	QD, MD, SMD, AC, HC, hằng số, *VD, *AC
$OR = n_1 n_2$	$n_1$ va $n_2$ (4 byte) thoa man $n_1$ = $n_2$	
$LDR > = n_1 n_2$	Lênh thực hiện phép tính logic Load , And hoặc Or giữa	
$AR > = n_1 n_2$	giá trị logic l với nội dung đỉnh ngăn xếp nếu hai số thực	
$OR > = n_1 n_2$	$n_1$ và $n_2$ (4 byte ) thỏa mấn $n_1$ > = $n_2$	
$LDR \leq n_1 n_2$	Lênh thực hiện phép tính	
AR <= n <sub>1</sub> n <sub>2</sub>	logic Load , And hoặc Or giữa giả trị logic l với nội dung đỉnh ngăn xếp nếu hai số thực	
$OR < = n_1 n_2$	$n_1$ va $n_2$ (4 byte) thóa mãn $n_1$ <= $n_2$	

#### 3. Chức năng dịch chuyển

dich

chuyển

Các lênh Các lênh dich chuyển thanh ghi được chia làm hai nhóm:

- □ Nhóm các lệnh làm việc với thanh ghi có độ dài bằng một từ đơn (16-bít) hay môt từ kép (32-bít). thanh ghi
  - □ Nhóm các lệnh làm việc với thanh ghi có độ dài tùy ý mà được định nghĩa trong lệnh.

Nhóm lênh với thanh ghi có đô dài 16 hoặc 32 bít.

Lệnh dịch chuyển thuộc nhóm này cho phép dịch chuyển và quay các bít trong các từ đơn và trong các từ kép.

Số lần dịch chuyển các bít của từ đơn hay từ kép được chỉ thị bằng một toán hạng trong được gọi là số lần đếm đẩy.

Số lần quay các bít của từ đơn hay từ kép cũng được chỉ thị bằng một toán hạng trong lênh, được gọi là số lần đếm quay.

#### Khi sử dụng các lệnh dịch chuyển các bít của từ đơn hay từ kép cần chú ý:

Sẽ không thực hiện việc dịch chuyển nếu như số đếm lần đẩy bằng 0.

Nếu số lần đẩy có giá trị lớn hơn 0, bít nhớ tràn SM1.1 có giá trị logic của bít cuối cùng được đấy ra.

Nếu số đếm lần đẩy lớn hơn hoặc bằng 16 (từ đơn), lớn hơn hoặc bằng 32 (từ kép) khi dịch chuyển thì lệnh sẽ chỉ thực hiện với số đếm lần đây lớn nhất là 16 hoặc 32.

Lệnh SRW (đẩy các bít từ đơn sang phải) và SDR (đẩy các bít từ kép sang phải) sẽ chuyển giá tri 0 vào bít cao nhất của từ hoặc từ kép tại mỗi lần đẩy. Sau khi thực hiện lệnh, bít SM1.1 sẽ có giá trị ủa bít thứ N-1 của từ đơn hoặc từ kép với N là số lần đẩy.

Lệnh SLW (đẩy các bít từ đơn sang trái) và SRD (đẩy các bít từ kép sang trái) sẽ chuyển giá trị logic 0 vào bít thấp nhất của từ hoặc từ kép tại mỗi lần đẩy. Sau khi thực hiện lệnh, bít SM1.1 sẽ có giá trị của bít thứ 16-N đối với từ đơn hoặc 32-N đối với từ kép, trong đó N là số lần đẩy.

Bít báo kết quả 0 (bít SM1.0) sẽ có giá trị logic bằng 1 nếu như sau khi thực hiện lênh đẩy nôi dung của từ đơn hay từ kép bằng 0.

#### Khi sử dụng lệnh quay các bít của từ đơn hay từ kép cần chú ý:

Lệnh quay thực hiện phép đẩy vòng tròn sang trái hay phải các bít của một từ đơn hoặc của một từ kép. Tại mỗi lần quay, giá trị logic của bít bị đẩy ra khỏi đầu này cũng là giá trị logic được đưa vào đầu kia của từ hay của từ kép.

Lênh quay sẽ không thực hiện nếu như số đếm lần quay có giá tri là 0 hay bằng bội số của 16 (với từ đơn) hoặc 32 (với từ kép).

Đối với các giá trị khác của số đếm lần quay lớn hơn 16 (đối với từ đơn) hoặc 32 (đối với từ kép), lênh sẽ thực hiện với số đếm lần quay mới bằng phần dự của số đếm lần quay cũ chia cho 16 hoặc chia cho 32.

Khi thực hiên lênh quay sang phải RRW (với từ đơn) hay RRD (với từ kép), tai mỗi lần quay giá trị thấp nhất trong từ hoặc từ kép được ghi vào bít báo tràn SM1.1. Sau khi lênh được thực hiên xong bít SM1.1 sẽ có giá tri logic bít 16-N của từ đơn hoặc 32-N của từ kép, trong đó N là số đếm lần quay.

Khi thực hiện lệnh quay sang trái RLW (với từ đơn) hay RLD (với từ kép) tại mỗi lần quay, giá trị logic của bít cao nhất trong từ hoặc từ kép được ghi vào bít báo tràn SM1.1. Sau khi lệnh được thực hiện xon bít SM1.1 sẽ có giá trị logic bít thứ N-1 trong từ đơn hoặc từ kép, trong đó N là số đếm lần quay (mới).

Bít báo kết quả 0 (bít SM1.0) sẽ có giá trị logic 1 nếu từ hay từ kép được quay có giá trị bằng 0.

Lệnh
Là lệnh dịch chuyển các bít của từ đơn IN sang phải N vị trí, trong đó N được gọi
SHR\_R
là số đếm lần dịch chuyển. Tại mỗi lần dịch chuyển, giá trị logic 0 được đưa vào bít cao (bít thứ 15) và giá trị logic của bít thấp (bít 0) được chuyển vào bít báo tràn SM1.1.

Trong LAD kết quả được ghi vào OUT, còn trong STL kết quả vẫn nằm trong IN.

LAD	STL
- SHR W EN	SRW IN N
IN OUT N	

## Cú pháp của lệnh như sau:

Lệnh
 Là lệnh dịch chuyển các bít của từ đơn IN sang trái n vị trí, trong đó N được gọi là số đếm lần dịch chuyển. Tại mỗi lần dịch chuyển, giá trị logic 0 được đưa vào bít thấp (bít 0) và giá trị logic của bít cao (bít thứ 15) được chuyển vào bít báo tràn SM1.1. Trong LAD kết quả được ghi vào từ OUT, còn trong STL kết quả vẫn nằm trong IN.

Cú pháp của lệnh như sau:

LAD	STL
SHL W EN	SLW IN N
IN OUTN	

Lệnh
Là lệnh dịch chuyển các bít của từ kép IN sang phải N vị trí với N là số đếm lần
SHR\_
dịch chuyển. Tại mỗi lần dịch chuyển, giá trị logic 0 được đưa vào bít cao (bít thứ
DW
31) và giá trị của bít thấp (bít 0) được chuyển vào bít báo tràn SM1.1. Trong LAD kết quả được ghi vào từ kép OUT, còn trong STL kết quả vẫn nằm trong IN.

#### Cú pháp của lệnh như sau:

LAD	STL
SHR DW EN	SRD IN N

Lệnh
 Là lệnh dịch chuyển các bít của từ kép IN sang trái N vị trí, trong N được gọi là số
 SHL\_
 đếm lần dịch chuyển. Tại mỗi lần dịch chuyển, giá trị logic 0 được đưa vào bít thấp
 DW
 (bit 0) và giá trị logic của bít cao (bít 31) được chuyển vào bít báo tràn SM1.1

Trong LAD kết quả được ghi vào từ kép OUT.

Trong STL kết quả vẫn nằm trong IN.

#### Cú pháp của lệnh như sau:

LAD	STL
SHLDW EN INOUT	SLD IN N

Lệnh
 Là lệnh quay các bít của từ đơn IN sang phải N lần, với N được gọi là số đếm lần quay. Tại mỗi lần quay, giá trị logic của bít thấp (bít 0) được chuyển vào bít báo tràn SM1.1 vừa được ghi lại vào bít cao (bít 15) của từ IN.

Trong LAD kết quả được ghi vào từ OUT.

Trong STL kết quả vẫn nằm trong IN.

Cú pháp của lệnh như sau:

LAD	STL
ROR W EN IN OUT N	RRW IN N

Lệnh
Là lệnh quay các bít của từ đơn IN sang phải N lần, với N được gọi là số đếm lần quay. Tại mỗi lần quay, giá trị logic của bít thấp (bít 0) được chuyển vào bít báo DW
tràn SM1.1 vừa được ghi lại vào bít cao (bít 15) của từ IN.

Trong LAD kết quả được ghi vào từ OUT.

Trong STL kết quả vẫn nằm trong IN.

## Cú pháp của lệnh như sau:

LAD	STL
ROR DW EN IN OUT N	RRD IN N

Lệnh
 Là lệnh quay các bít của từ đơn IN sang trái N lần với N là số đếm lần quay. Tại mỗi lần quay, giá trị logic của bít cao (bít 15) vừa được chuyển vào bít báo tràn SM1.1 vừa được ghi lại vào bít thấp của từ IN.

Trong LAD kết quả được ghi vào từ OUT.

Trong STL kết quả vẫn nằm trong IN

#### Cú pháp của lệnh như sau:

LAD	STL
ROL W EN IN OUT	RLW IN N

Lệnh Là lệnh quay các bít của từ kép IN sang trái N lần, trong đó N được gọi là số đếm
ROL lần quay. Tại mỗi lần quay, giá trị logic của bít cao (bít 31) vừa được chuyển vào bít báo tràn SM1.1 vừa được ghi lại vào bít thấp (bít 0) của từ kép IN.

Trong LAD kết quả được ghi vào từ OUT.

Trong STL kết quả vẫn nằm trong IN

#### Cú pháp của lệnh như sau:

LAD	STL
ROL DW EN IN OUT	RLD IN N

## 4. Chức năng chuyển đổi

Hàm đổi<br/>dữ liệu<br/>tương<br/>ứng<br/>thanh ghiHàm SEG chuyển đổi số nguyên hệ cơ số Hexa trong khoảng 0 ÷ F sang thành giá<br/>trị bit tương ứng của thanh ghi 7 nét .Hàm SEG lập giá trị các bit của thanh ghi 7 nét tương ứng với nội dung của 4 bit<br/>thấp của byte đấu vào IN. Kết quả được ghi cào byte đầu ra OUT7 nét

<b>G</b> 2		11255	Tha	nh ş	ghi 7	nét		
so nguyên		g	f	e	d	Ċ	b	a
0	0	0	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0
2	0	1	0	1	1	0	1	1
3	0	1	0	0	1	1	1	1
4	0	1	1	0	0	1	1	0
5	0	1	1	0	1	1	0	1
6	0	1	1	1	1	1	1	
7	0	0	0	0	0	1	1	1
8	0	1	1	1	1	1	1	1
9	0	1	1	0	0	1	1	1
A	0	1	1	1	0	1	1	1
В	0	1	1	1	1	1	0	0
С	0	0	1	1	1	0	0	1
D	0	1	0	1	1	1	1	0
E	0	1	1	1	1	0	0	1
F	0	1	1	1	0	0	0	1

Sơ đồ các bit của thanh ghi 7 nét:

5. Chức năng toán học

## (ADD)

Lệnh ADD_I Lệnh ADD_DI	Là lệnh thực hiện phép cộng các số nguyên 16-bít IN1 và IN2. Trong LAD kết quả là một số nguyên 16-bít được ghi vào OUT, tức là: IN1 + IN2 = OUT. Còn trong STL, kết quả cũng là một giá trị 16-bít nhưng được ghi vào IN2, tức là IN1 + IN2 = IN2. Là lệnh thực hiện phép cộng các số nguyên 32-bít IN1 và IN2 Trong LAD, kết quả là một số nguyên 32-bít được ghi vào OUT, tức là: IN1 + IN2 = OUT.
	Còn trong STL, kết quả cũng là một giá trị 32-bít nhưng được ghi vào IN2, tức là IN1 + IN2 = IN2.
	Là lênh thực hiện phép công các số thực 32-bít IN1 và IN2
Lệnh ADD_R	Trong LAD, kết quả là một số thực 32-bít được ghi vào OUT, tức là: IN1 + IN2 = OUT.
	Còn trong STL, kết quả cũng là một giá trị thực 32-bít nhưng được ghi vào IN2, tức là $IN1 + IN2 = IN2$ .
5.2 Lệnh trừ (SUB)	
Lệnh SUB_I	
	Là lệnh thực hiện phép trừ các số nguyên 16-bít IN1 và IN2
	Trong LAD kết quả là một số nguyên 16-bít và được ghi vào OUT, tức là: IN1 - IN2 = OUT.
Lệnh SUB-DI	Còn trong STL, kết quả là một giá trị 16-bít nhưng được ghi lại vào IN2, tức là IN1- IN2 = IN2.
	Là lênh thực hiên phép trừ các số nguyên 32-bít IN1 và IN2
	Trong LAD kết quả là một số nguyên 32-bít được ghi vào IN2, tức là: IN1 - IN2 = IN2.
Lệnh SUB_R	Còn trong STL, kết quả là một giá trị 32-bít nhưng được ghi lại vào IN2, tức là IN1- IN2 = IN2.
	Là lênh thực hiện phép trừ các số thực 32-bít IN1 và IN2
	Trong LAD kết quả là một số thực 32-bít được ghi vào OUT, tức là: IN1 - IN2 = OUT.

Trong STL, kết quả là một giá trị 32-bít nhưng được ghi lại vào IN2, tức là IN1- IN2 = IN2.

Cú pháp dùng lệnh cộng và trừ trong LAD và STL như sau:

LAD	STL
ADD I EN IN1 IN2 OUT	+1 IN1 IN2
SUB I EN IN1 IN2 OUT	- I IN I IN 2
SUB R EN N1 N2 OUT	- R IN1 IN2
ADD DI EN IN1 IN2 OUT	+ D INI IN2
SUB DI EN N1 N2 OUT	- D IN1 IN2
ADD R EN IN1 IN2 OUT	+ R INLIN2

## 5.3 Lệnh nhân (MUL):

Lệnh MUL Trong LAD: Lệnh thực hiện phép nhân 2 số nguyên 16-bít IN1 và IN2 và cho ra kết quả 32-bít chứa trong từ kép OUT (4 byte).

Trong STL: Lệnh thực hiện phép nhân giữa 2 số nguyên 16-bít n1 và số nguyên chứa trong từ thấp (từ 0 đến bít 15) của toán hạng 32-bít n2 (4 byte). Kết quả 32-bít được ghi vào n2.

Lệnh MUL RTrong LAD: lệnh thực hiện phép nhân hai số thực 32-bít IN1 và IN2 và<br/>cho ra kết quả 32-bít chứa trong từ kép OUT (4 byte).

Trong STL: Lệnh thực hiện phép nhân giữa số thực 32-bít được ghi vào IN2.

Cú pháp dùng lệnh trong LAD và STL như sau:

LAD	STL
MUL EN IN1 IN2 OUT	MUL n1 n2
MUL R EN IN1 IN2 OUT	*R IN 1 IN2

5.4 Lệnh chia
(DIV)
Trong LAD: Lệnh thực hiện phép chia số nguyên 16-bít IN1 cho số nguyên 16-bít IN2. Kết quả 32-bít chứa trong từ kép OUT gồm thương số ghi trong mảng 16-bít từ bít 0 đến bít15 (từ thấp) và phần dư cũng 16-bít ghi trong mảng từ bít-16 đến bít-31 (từ cao).

Trong STL: Lệnh thực hiện phép chia số nguyên 16-bít n1 cho số nguyên, số nguyên 16-bít nằm trong từ thấp từ bít 0 đến bít 15 của toán hạng 32-bít n2. Kết quả 32-bít được ghi lại vào n2 bao gồm thương số ghi trong mảng 16-bít từ bít 0 đến bít 15 (từ thấp) và phần dư ghi trong mảng 16-bít từ bít-16 đến bít-31 (từ cao).

Trong LAD: lệnh thực hiện phép chia số thực 32-bít IN1 cho số thực 32bít IN2 và cho ra kết quả 32-bít chứa trong từ kép OUT.

Lệnh DIV\_R Trong STL, lệnh thực hiện phép chia số thực 32-bít IN1 cho số thực 32-bít IN2, kết quả 32-bít được ghi lại vào IN2.

Cú pháp dùng lệnh trong LAD và STL như sau:

LAD	STL
DIV EN IN1 IN2 OUT	DIV n1 n2
DIV R EN	/R n1 n2
IN1 N2 OUT	

5.5 Lệnh lấy căn Là một lệnh thực hiện lấy căn bậc hai của số thực 32-bít IN. Kết quả cũng là một số 32-bít được ghi vào từ kép OUT.

## Cú pháp dùng lệnh trong LAD và STL như sau:

LAD	s	TL
SQRT EN IN OUT	SQRT IN	OUT

**Lệnh INC\_B** Là lệnh cộng số nguyên 1 vào nội dung của byte đầu vào.

Trong LAD: Kết quả được ghi vào OUT, tức là: IN1 + 1 = OUT.

Trong STL: Kết quả được ghi vào IN.

Cú pháp dùng lệnh INCB trong LAD và trong STL như sau:

LAD	STL
INC B EN IN OUT	INCW IN

## **Lệnh INC\_W** Lệnh cộng số nguyên 1 vào nội dung từ đơn In.

Trong LAD: Kết quả được ghi vào OUT.

Trong STL: Kết quả được ghi lại vào IN.

Cú pháp dùng lệnh INCW trong LAD và trong STL như sau:

LAD	STL
INC W EN IN OUT	INCW IN

Lệnh INC\_DWLà lệnh cộng số nguyên 1 vào nội dung từ kép IN(DOUBLE<br/>WORD)Trong LAD: Kết quả được ghi vào OUT, tức là: IN + 1 = OUT<br/>Trong STL: Kết quả được ghi vào IN, tức là: IN + 1 = IN

Cú pháp dùng lệnh INC\_DW trong LAD và trong STL như sau:

LAD	STL
INC DW EN IN OUT	INCD IN

**Lệnh DEC\_B** Là lệnh bớt nội dung của byte đầu vào đi 1 đơn vị.

Trong LAD: Kết quả được ghi vào OUT, tức là: IN - 1 = OUT

Trong STL: Kết quả được ghi vào IN, tức là: IN - 1 = IN

Cú pháp dùng lệnh DECW trong STL và DEC\_W trong LAD như sau:

LAD	STL
DEC B EN IN OUT	DECB IN

**Lệnh DEC\_W** Là lệnh bớt nội dung IN đi 1 đơn vị.

Trong LAD: Kết quả được ghi vào OUT, tức là: IN - 1 = OUTTrong STL: Kết quả được ghi vào IN, tức là: IN - 1 = IN
LAD	STL	
INC W EN IN OUT	DECW IN	

Cú pháp dùng lệnh DECW trong STL và DEC\_W trong LAD như sau:

**Lệnh DEC\_DW** Là lệnh giảm nội dung từ kép IN đi 1 đơn vị.

Trong LAD: Kết quả được ghi vào OUT, tức là: IN - 1 = OUT

Trong STL: Kết quả được ghi vào IN, tức là: IN - 1 = I

# Cú pháp dùng lệnh DECDW trong STL hay DEC\_DW trong LAD như sau:

LAD	STL	
INC DW EN IN OUT	DECD IN	

# 6. Đồng hồ thời gian thực

Đồng hồ tời gian thực chỉ có ở CPU 214. Để có thể làm việc với đồng hồ thời gian thực CPU 214 cung cấp hai lệnh đọc và ghi giá trị cho đồng hồ . Những giá trị đọc được hoặc ghi được với đồng hồ thời gian thực là các giá trị về ngày, thánh , năm , và các giá trị giờ , phúc , giây .Các dữ liệu đọc , ghi với đồng hồ thời gian thực trong LAD và trong STL có độ dài một byte và phải được mã hóa theo kiểu số nhị phân BCD .

Byte 0	Nãm (0 ÷99)			
Byte 1	Tháng(0 ÷12)			
Byte 2	Ngây (0 ÷≈31)			
Byte 3	Giờ (0 ÷2 3)			
Byte 4	Phúc (0 ÷ 59)			
Byte 5	Giây (0 ÷59)			
Byte 6	0			
Byte 7	0	ngày trong tuần		

Năm	Tháng	Ngày	Giờ	Phút	Giây
(yy)	(mm)	(dd)	(hh)	(mm)	(ss)
0 ÷99	1 ÷ 12	1÷31	0÷23	0 ÷59	0÷59

Riêng giá trị về ngày trong tuần là một số tương ứng với nội dung của nibble(4 bit) thấp trong byte theo kiểu :

Chủ nhật	Thứ hai	Thứ máy biến áp	Thứ tư	Thứ năm	Thứ sáu	Thứ bảy
1	2	3	4	5	6	7

**READ\_RTC**Lệnh đọc nội dung của đồng hồ thời gian thực với bộ đệm 8 byte được chỉ<br/>thị trong lệnh bằng toán hạng T.

TODR (STL)

SET\_RTCLệnh ghi nội của bộ đệm 8 byte được chỉ thị trong lệnh bằng toán hạng(LAD)T vào đồng hồ thời gian thực .

**TODW (STL)**Cú pháp sử dụng lệnh đọc, ghi dữ liệu với đồng hồ thời gian thực trong<br/>LAD, STL:

LAD	STL	Toán hạng
READ RTC EN T	TODR T	T(byte) : VB , IB , QB , MB , SMB , *VD , *AC
SET RTC T	TODW T	

chương trình chính, vừa trong chương trình xử lý ngắt .Khi một lệnh TODR hay TODW đã được thực hiện , thì khi gọi chương trình xử lý ngắt , các lệnh làn việc với đồng hồ thời gian thực trong chương trình xử lý ngắt sẽ không được thục hiện nữa. Bit SM4.5 sẽ có logic 1 trong nhuững trường hợp như vậy.

#### Bài 5: Tín hiệu Analog Mã chương/Bài: MĐ22-05

Giới thiệu:

Mục tiêu:

- Trình bày được các bộ chuyển đổi đo.

- Vận dụng các bài toán vào thực tế: Lập trình, kết nối, chạy thử...

- Rèn luyện đức tính tích cực, chủ động và sáng tạo

#### Nội dung chính:

1.	Tín hiệu Anlog
2.	Biểu diễn các giá trị Analog
3.	Kết nối vào/ra Analog
4.	Hiệu chỉnh tín hiệu Analog
5.	Giới thiệu về Module analog PLC S7 – 200

#### 1. Tín hiệu Anlog

Trong kỹ thuật điện, tín hiệu là quan hệ giữa thời gian (hoặc cũng có thể là không gian, hoặc cả hai nhưng ít gặp) với một đặc trưng điện nào đó (điện áp, dòng điện, điện tích, mật độ dòng điện, ...). Lấy thí dụ, chúng ta thường gặp các tín hiệu như điện áp giữa hai cực của một mạng v(t) [volt] với t là thời gian - đây là trường hợp tín hiệu là hàm của thời gian (thường gặp). Chúng ta cũng có thể gặp phân bố điện tích của một vật tích tĩnh điện q(x,y,z) [coulomb] với x, y, z là tọa độ Descartes - đây là trường hợp tín hiệu là hàm của không gian (ít gặp hơn).

Định nghĩa bản tin và thông tin thì mang tính triết học quá, nhưng có thể hiểu đơn giản: bản tin là tập hợp các thông tin có quan hệ chặt chẽ để tạo nên một ý nghĩa tương đối hoàn chỉnh. Thí dụ, bạn có một số điện thoại +84915\*\*\*\*\* thì từng chữ số một là thông tin, nhưng bản tin phải là toàn bộ chuỗi ký hiệu này.

1.1 Tín hiệu liên tục theo biên độ và liên tục theo thời gian



Đây là dạng tín hiệu thường gặp, thí dụ như tín hiệu từ một cảm biến. 1.2. Tín hiệu liên tục theo biên độ và rời rạc theo thời gian



Thí dụ như tín hiệu xung PAM. 1.3. Tín hiệu rời rạc theo biên độ và liên tục theo thời gian



Trong thí dụ này, tín hiệu chỉ nhận các giá trị {-2, -1, 0, 1, 2, 3}. Bạn có thể thấy các tín hiệu thuộc loại này trong điều chế băng gốc (base band modulation). Thí dụ như tín hiệu nhị phân (2 mức) trong <u>NRZ</u>, trong đó tín hiệu chỉ nhận hai giá trị {-1, 1}; hay tam phân (3 mức) trong <u>RZ</u>, HDB3, ..., trong đó tín hiệu chỉ nhận ba giá trị {-1, 0, 1}. Tổng quát lên là các tín hiệu M-phân (M mức).

1.4. Tín hiệu rời rạc theo biên độ và rời rạc theo thời gian



Thí dụ như tín hiệu PCM.

Tiêu chí phân loại tín hiệu tương tự - số: thì (1) và (2) là tín hiệu tương tự; còn (3) và (4) là tín hiệu số.

Tiêu chí phân loại tín hiệu liên tục - rời rạc: ở đây hiểu ngầm rằng liên tục theo thời gian và rời rạc theo thời gian, do vậy (1) và (3) là tín hiệu liên tục; còn (2) và (4) là tín hiệu rời rạc.

Về so sánh xử lý tương tự và xử lý số, chắc có lẽ chỉ nói sơ sơ được thôi. Đặc trưng của xử lý số là thao tác trên linh kiện có chức năng nhớ, cụ thể là trên thanh ghi (register); dẫn đến cho phép thời gian xử lý có thể kéo rất dài, quá trình xử lý có thể rất phức tạp, chức năng xử lý rất phong phú nhưng bù lại là hệ thống có thể không có tính chất thời gian thực (real time - đáp ứng của hệ thống có rất nhanh, khi có tín hiệu ngõ vào thì gần như có liền tín hiệu ngõ ra). Trong khi đó, xử lý tương tự lại không có linh kiện nhớ (chỉ có thể làm delay trong thời gian ngắn) dẫn đến quá trình xử lý phải thật nhanh, đơn giản và do đó chức năng xử lý cũng đơn giản hơn so với xử lý số nhưng bù lại hệ thống có tính chất real time.

Thiết kế vi mạch có thể chia thành 2 mảng: thiết kế vi mạch tương tự (các vi mạch tương tự phổ biến như opamp, <u>ADC/DAC</u>, ổn áp) và thiết kế vi mạch số (vi xử lý chẳng hạn).

Công nghệ 22 nm (90 nm, 65 nm, 45 nm, 32 nm, ...): theo tôi hiểu chỉ là tên gọi của một công nghệ, con số 22 nm (và các con số khác) không có ý nghĩa rõ ràng và không ám chỉ một kích thước vật lý cụ thể nào. Tuy nhiên, nó có thể gợi ý về kích thước của một chi tiết nhỏ nhất (xấp xỉ) của transistor mà công nghệ này có thể thực hiện, đó thường là bề rộng của cực cổng (gate) của một transistor CMOS. Đây là quảng đường mà dòng điện phải chảy qua trong một transistor và do đó quyết định đến tốc độ đóng-ngắt của transistor; và ngoài ra nó cũng ảnh hưởng đến diện tích của một transistor. Con số này càng nhỏ thì tốc độ và mật độ tích hợp càng lớn. Mặc dù vậy, nó không phải là độ phân giải của công nghệ vì quá trình quang khắc sử dụng bước sóng ánh sáng 193 nm vốn lớn hơn 22 nm nhiều và người chế tạo được chi tiết 22 nm; dẫn đến là chúng ta có thể tạo được chi tiết 22 nm; dẫn đến là chúng ta có thể tạo được chi tiết 22 nm nhưng lại không thể tạo được chi tiết có kích thước y hệt vậy kế bên cạnh, tức là 22 nm không phải là độ phân giải.

#### 2.Biểu diễn các giá trị Analog trong S7-200

Ví dụ đọc tín hiệu đo mức nước có dãi đo từ 0m-40m. Tín hiệu truyền về là 4-20mA, nghĩa là 0m = 4mA, 40m=20mA. Nhiệm vụ là phải hiển thị được con số mực nước lên HMI.

Lưu ý, S7200 có module đọc analog có độ phân giải từ 4000 đến 32000, nghĩa là 4mA đọc vào thì trong PLC hiểu là 4000 đơn vị và 20mA đọc về là 32000 đơn vị.

Như vậy ta có thể lặp được 1 đường thẳng qua 2 điểm.

Giả sử nếu tín hiệu thực tế là 9mA, hỏi mức nước cần tính toán và hiển thị trên HMI là bao nhiêu? Công thức để tính cho tín hiệu Analog sẽ là:

Os=[(Osvmax -Osvmin)(Isv-Isvmin)/(Isvmax-Isvmin)]+Osvmin

O: outputs

s: sensor

v :value

#### 3. Kết nối vào/ra Analog

Order Number	Expansion Model	EM Inputs	EM Outputs	Removable Connector
6ES7 231-0HC22-0XA0	EM 231 Analog Input, 4 Inputs	4	1 <b>3</b> 7	No
6ES7 232-0HB22-0XA0	EM 232 Analog Output, 2 Outputs	121	2	No
6ES7 235-0KD22-0XA0	EM 235 Analog Combination 4 Inputs/1 Output	4	1	No



#### 4. Hiệu chỉnh tín hiệu Analog

Module tương tự khác module số là nó dùng để xử lý giá trị chứ không phải xử lý trạng thái. Do đó kiểu biến là dạng byte, word, double word thay vì dạng bit như module số. Mỗi module có 1 độ phân giải riêng (resolution), ví dụ 8 bit, 12 bit, 32 bit, v.v và 1 giải giá trị (range, ví dụ 0=10V, 4-20mA, 0-20mA). Khi nhận 1 giá trị từ cảm biến, nó sẽ số hóa giá trị đó dựa trên range và resolution của nó (vì PLC dùng thông tin số cho mọi quá trình xử lý bên trong).

Value = Input\* (2 mũ n-1)/(max range-min range); n: độ phân giải tính theo bit. Trong đó (max range-min range)/(2 mũ n-1) là giá trị tương ứng với 1 bit.

Giá trị sẽ được làm tròn dưới dạng số nguyên. Sai số do làm tròn sẽ nói lên độ phân giải là cao hay thấp.

Ví dụ chúng ta đo 1 đại lượng nào đó, cảm biến biến đổi thành giá trị điện áp là 3V. Lúc này bạn phải dùng đầu vào tương tự có dải 0-10V để nhận tín hiệu. Giả sử độ phân giải của module tương tự là 8 bit. Giá trị mà PLC tính ra sẽ là:

Value = 3\*(2 mũ 8-1)/(10-0)=3\*255/10=76.5, làm tròn lên sẽ là 77. Giá trị digital mà PLC sẽ sử dụng cho các quá trình tính toán bên trong sẽ là 01001101. Sai số là 0.5\*=0.5\*10/(2 mũ 8-1)

Nếu độ phân giải của bạn là 11 bit thì Value sẽ là 614.1, làm tròn là 614. Sai số là 0.1\*10/(2 mũ 11-1).

Rõ ràng là độ phân giải càng cao thì sai số càng ít. (Vì thế nên chọn module vào ra phải rất chú ý đến độ phân giải, và nhớ kết nối đúng với giải giá trị của đầu vào ra, đừng cắm tín hiệu điện áp vào IO có dải 0-20mA).

5. Giới thiệu về Module analog S7-200









# Bài 6: PLC của các hãng khác Mã chương/Bài: MĐ22-06

Giới thiệu:

Mục tiêu:

- Trình bày nguyên lý, cấu tạo của các họ PLC Omron, Mitsubishi...

- Thực hiện lập trình của các họ PLC nói trên.
- Rèn luyện đức tính tích cực, chủ động và sáng tạo

# Nội dung chính:

1.	PLC của hãng OMRON
2.	PLC của hãng Siemens (trung bình và lớn).
3.	PLC của hãng Allenbradley.
4.	PLC của hãng Telemecanique.
5.	PLC của hãng Mitsubishi

# Bài 7: Lắp đặt mô hình điều khiển bằng PLC Mã chương/Bài: MĐ22-04

**Giới thiệu:** Bài này trình bày các mô hình điều khiển bằng PLC, quy trình công nghệ, mạch động lực, mạch điều khiển và quy trình lấp đặt.

# Mục tiêu:

- Phân tích qui trình công nghệ của một số mạch máy sản xuất.
- Lập trình được một số mạch ứng dụng thường gặp trong thực tế.
- Nạp trình, vận hành và kiểm tra mạch hoạt động theo yêu cầu kỹ thuật.
- Rèn luyện đức tính tích cực, chủ động và sáng tạo

Nội dung chính:

1.	Giới thiệu chung
2.	Cách kết nối dây (các mô hình trong xưởng)
3.	Các bài tập ứng dụng

## 1. Giới thiệu chung

Trong tất cả các lĩnh vực đặc biệt là trong công nghiệp, điều khiển động cơ không đồng bộ 3 pha là một phần quan trọng vì tính phổ biến của chúng. Các mạch khởi động động cơ, đổi chiều quay hay các mạch điều khiển tốc độ động cơ là những mạch cơ bản nhất. Ở chương này giới thiệu một số mạch ứng dụng thực hành trong các mạch điều khiển động cơ không đồng bộ 3 pha.

# 2. Cách kết nối dây

Giới thiệu thực tế các mô hình trong xưởng PLC

#### 3. Các bài tập ứng dụng

#### 3.1. Mạch tự khởi động sao – tam giác cho động cơ 3 pha có vành trượt



Mô tả hoạt động:

Ở loại động cơ xoay chiều 3 pha có vành trượt dùng nếu dùng phương pháp khởi động qua việc cắt dần các bậc điện trở nới với phần ứng, điều đó tạo ra mô men khởi động lớn và dòng khởi động nhỏ. Qua việc giảm dần các bậc điện trở, tốc độ của động cơ được tăng dần đến tốc độ định mức. Khi nhấn S2, K1 có điện và tự duy trì, đồng thời rơ le thời gian K7 có điện, sau thời gian 3s, K4 có điện và tự duy trì, K7 mất điện, đồng thời K6 có điện. Nhóm điện trở R3 bị ngắn mạch qua K4. Sau 3s, K3 có điện và tự duy trì, K4 và K6 mất điện, K5 có điện. Nhóm điện trở R2 bị ngắn mạch qua K3. Sau một thời gian 3s thì K2 có điện, K3 và K5 mất điện, nhóm điện trở R1 bị ngắn mạch. Quá trình khởi động kết thúc. Nhấn S1 động cơ bị cắt khỏi nguồn điện.

# <u>Nhiệm vụ:</u>

Hãy thực hiện mạch theo các yêu cầu sau:

1/ Xác định ngõ vào/ra.

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

# F3 Nguồn cung cấp 51 10.0 10.1 00.0 52 10.2 10.3 0

3.1. Điều khiển khởi động động cơ xoay chiều 3 pha dạng sao – tam giác



# Yêu cầu công nghệ

Khi nhấn nút nhấn On, động cơ khởi động chạy ở chế độ nối sao. Sau 5s động cơ chạy ở chế độ nối tam giác. Mạch có mắc CB bảo vệ động cơ (hay Overload).

Nhấn nút Off, động cơ ngừng.

Hãy viết chương trình điều khiển khởi động động cơ 3 pha trên.

#### <u>Nhiệm vụ:</u>

Hãy thực hiện theo các yêu cầu sau:

# Mục đích – Yêu cầu:

 Làm quen với lệnh điều khiển Timer và bit nhớ.

- Hiểu và biết cách sử dụng lệnh điều khiển Timer và bit nhớ trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer và bit nhớ.

- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

# Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn – công tắc, các cầu dao, khởi động từ và động cơ.

# Kết nối với PLC

1/ Xác định ngõ vào/ra. Kết nối với PLC

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

# 3.1. Điều khiển khởi động động cơ 3 pha dạng sao – tam giác có báo lỗi khởi động



# <u>Yêu cầu công nghệ</u>

Khi nhấn nút nhấn On, động cơ khởi động chạy ở chế độ nối sao. Sau 5s động cơ chạy ở chế độ nối tam giác. Mạch có mắc CB bảo vệ động cơ (hay Overload).

Nếu khởi động ở chế độ sao có lỗi, ví dụ cảm biến báo lỗi nhận biết được lỗi này, sau 5s Simatic S7 – 200 không chuyển sang chế độ nối tam giác và như vậy tránh hư hỏng động cơ hay bất cứ thiệt hại nào có thể xảy ra. (Lưu ý: khi động cơ đã hoạt động bình thường mà có lỗi mạch ngừng lại tức thì). Khi có lỗi đèn báo lỗi (Fault light) sáng.

Nhấn nút Off, động cơ ngừng.

Hãy viết chương trình điều khiển khởi động động cơ 3 pha trên.

#### <u>Nhiệm vụ:</u>

Hãy thực hiện theo các yêu cầu sau:

1/ Xác định ngõ vào/ra. Kết nối với PLC

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

# 3.3. Mô hình trạm phân loại

#### Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer và bit nhớ.

- Hiểu và biết cách sử dụng lệnh điều khiển Timer và bit nhớ trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer và bit nhớ.

- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

# Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn – công tắc, các cầu dao, khởi động từ và động cơ.

# Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Counter, lệnh điều khiển lưu trình.



Có một dây chuyền sản xuất bia có một bộ phận kiểm tra hoạt động như sau:

Khi đổ bia vào chai thì các chai này được đưa qua một băng tải, dọc theo băng tải có 4 trạm kiểm tra.

- Trạm 1: kiểm tra chai có bị mẻ hay không?
- Trạm 2: kiểm tra chai có dán nhãn hay chưa?
- Trạm 3: kiểm tra chai đã được đóng nắp hay chưa?
- Trạm 4: kiểm tra bia đã được rót đầy hay chưa?

Nếu chai nào không đảm bảo bất kỳ 1 trong các tiêu chuẩn kiểm tra trên thì sẽ bị loại bỏ sau khi qua 4 trạm.

#### <u>Nhiệm vụ:</u>

Hãy thực hiện theo các yêu cầu sau:

- 1/ Xác định ngõ vào/ra. Vẽ sơ đồ kết nối với PLC
- 2/ Vẽ sơ đồ LAD.
- 3/ Viết và thử chương trình.

- Hiểu và biết cách sử dụng lệnh điều khiển Counter, lệnh điều khiển lưu trình trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Counter, lệnh điều khiển lưu trình.

- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

#### Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn, 1 Van, 1 động cơ, 4 cảm biến.

#### Điều khiển động cơ băng tải



## Yêu cầu công nghệ

Băng tải gồm 3 phân đoạn, cần điều khiển sao cho động cơ của mỗi phân đoạn chỉ chạy khi có đối tượng (tấm đồng) đang nằm trên phân đoạn tương ứng.

Vị trí của tấm kim loại được xác định bởi các cảm biến tiệm cận đặt gần nó (sensor 1, 2, 3). Khi tấm kim loại nằm trong khoảng cách phát hiện của 1 sensor, động cơ tương ứng sẽ vẫn làm việc. Khi tấm kim loại nằm ngoài khoảng cách phát hiện của sensor, một timer trễ sẽ được kích hoạt và khi thời gian đặt của quá trình hết thì động cơ tương ứng sẽ ngừng.

#### <u>Nhiệm vụ:</u>

Hãy thực hiện theo các yêu cầu sau:

1/ Xác định ngõ vào/ra. Vẽ sơ đồ kết nối với PLC

#### 2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

#### Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, lệnh về điều khiển lưu trình.

- Hiểu và biết cách sử dụng lệnh điều khiển Timer, lệnh về điều khiển lưu trình trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer, lệnh về điều khiển lưu trình.

 Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

#### Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, 03 động cơ, các công tắc, 03 cảm biến.

# Băng tải dây chuyền đóng gói



#### <u>Yêu cầu công nghệ</u>

Khi nhấn nút nhấn khởi động PB1 (Start) băng tải BT2 mang hộp đựng sản phẩm di chuyển. Cảm biến S2 phát hiện sự có mặt của hộp đựng sản phẩm, băng tải BT2 dừng lại. Băng tải táo BT1 dịch chuyển, táo di chuyển vào hộp đựng, số lượng quả táo được đếm bởi một cảm biến quang hồng ngoại S1, khi đếm được 10 sản phẩm (mỗi hộp chứa 10 sản phẩm) thì băng tải BT1 dừng, tiếp tục BT2 dịch chuyển để đóng gói hộp mới. Bộ đếm được Reset và hoạt động lập lại cho đến khi nút nhấn dừng PB2 (Stop) được nhấn.

Hãy viết chương trình điều khiển dây chuyền đóng gói táo trên.

#### Nhiệm vụ:

Hãy thực hiện theo các yêu cầu sau:

1/ Xác định ngõ vào/ra. Vẽ sơ đồ kết nối với PLC

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

#### 3.5. Điều khiển dây chuyền chiết nước vào chai

#### Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Counter, lệnh điều khiển lưu trình.

- Hiểu và biết cách sử dụng lệnh điều khiển Counter, lệnh điều khiển lưu trình trong quá trình soạn thảo.

Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Counter, lệnh điều khiển lưu trình.

 Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

#### Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn, các công tắc, 2 động cơ, cảm biến.

#### Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, lệnh về điều khiển lưu trình.



Viết chương trình cho công đoạn chiết nước vào chai theo nguyên tắc thời gian, mô tả hình trên. Nhấn nút Start PB, băng tải chạy và mang theo chai, cảm biến chai phát hiện, băng tải dừng, bắt đầu mở van chiết để chiết nước vào chai. Thời gian chiết đầy là 30s, sau khi chiết đầy băng tải lại chạy và tiếp tục chiết chai kế tiếp

Hãy viết chương trình điều khiển hoạt động của dây chuyền trên.

#### <u>Nhiệm vụ:</u>

Hãy thực hiện theo các yêu cầu sau:

1/ Xác định ngõ vào/ra. Vẽ sơ đồ kết nối với PLC

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

- Hiểu và biết cách sử dụng lệnh điều khiển Timer, lệnh về điều khiển lưu trình trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer, lệnh về điều khiển lưu trình.

- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

#### Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, 01 động cơ, các cảm biến, van.

#### 3.6. Mô hình trộn hóa chất

Điều khiển trộn sơn theo thời gian



# <u>Yêu cầu công nghệ</u>

Qui trình trộn hai loại sơn màu khác nhau diễn ra như sau:

Nhấn nút On hai Fill Valve 1 và 2 mở ra cho sơn vào bình, sau 05s hai Valve đóng lại. Mixer khởi động để trộn sơn trong vòng 06s rồi dừng lại, đồng thời lúc này Drain Valve mở để xả sơn. Qui trình lập lại 05 lần rồi dừng.

Nhấn nút Off, hết sơn mới được dừng.

Hãy viết chương trình điều khiển trộn sơn trên.

#### <u>Nhiệm vụ:</u>

Hãy thực hiện theo các yêu cầu sau:

1/ Xác định ngõ vào/ra. Vẽ sơ đồ kết nối với PLC

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

#### Điều khiển trộn sơn theo mức

## Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, Counter, chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ.

- Hiểu và biết cách sử dụng lệnh điều khiển Timer, chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer, Counter chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ.

- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

#### Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, cảm biến, bóng đèn.

#### Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, Counter, chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ.

- Hiểu và biết cách sử



Khi ấn nút khởi động, nó tác động lên Fill valve 1 và Fill valve 2, cho phép 2 chất lỏng bắt đầu đổ vào bình chứa. Khi bình chứa được đổ đầy, công tắc dò mức SS đi lên làm ngắt 2 valve fill 1 và 2, và khởi động động cơ Mixer để trộn 2 chất lỏng. Động cơ được cho chạy trong vòng 2 phút, sau đó tự động ngừng động cơ lại và cho mở Drain valve để xả chất lỏng ra. Khi bình chứa đã xả hết thì thì SS ngắt Drain valve. Người ta có thể dùng nút STOP để dừng quá trình xử lý bất kỳ lúc nào. Số lần trộn là 3 mẻ trộn và sẽ kết thúc. Nếu thực hiện lại ta phải nhấn nút Reset. Nếu động cơ quá tải thì toàn bộ mạch sẽ ngưng hoạt động.

Hãy viết chương trình điều khiển trộn sơn trên.

#### Nhiệm vụ:

Hãy thực hiện theo các yêu cầu sau:

1/ Xác định ngõ vào/ra. Vẽ sơ đồ kết nối với PLC

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

#### Điều khiển trộn hóa chất

dụng lệnh điều khiển Timer, Counter chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer, Counter chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ.

- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

#### Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, cảm biến, bóng đèn.

#### Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, Counter, chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ.

- Hiểu và biết cách sử dụng lệnh điều khiển Timer,



Có 2 bồn trộn hóa chất, mỗi bồn sử dụng 1 động cơ trộn. Tank 1 trộn hóa chất A, tank 2 trộn hóa chất B. Trên bảng điều khiển có 3 lựa chọn:

- Nếu nhấn nút PB thì cả 2 Tank đều được chọn làm việc trong 30s.

- Nếu nhấn nút PB1 thì chỉ có Tank 1 được chọn làm việc trong 30s. (Tank 2 nghỉ).

- Nếu nhấn nút PB2 thì chỉ có Tank 2 được chọn làm việc trong 30s. (Tank 1 nghỉ).

Khi đang trộn hóa chất, nếu bồn hóa chất bị hở van thì phải báo động ngay và lập tức dừng quá trình trộn lại.

Hãy viết chương trình điều khiển trộn hóa chất trên.

#### Nhiệm vụ:

Hãy thực hiện theo các yêu cầu sau:

1/ Xác định ngõ vào/ra. Vẽ sơ đồ kết nối với PLC

2/ Vẽ sơ đồ LAD.

3/ Viết và thử chương trình.

Counter chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ trong quá trình soạn thảo.

# Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer, Counter chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ.

- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- Cách kết nối tính hiệu vào/ ra của PLC.

# Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, nút nhấn, van, bóng đèn.