

**TÓM TẮT MÔN HỌC TỰ ĐỘNG ĐIỀU KHIỂN - PLC**

1. Tên môn học:                   - **Tự động điều khiển - PLC**
2. Vị trí môn học:               - **Môn học có tính chất tiên quyết, chuyên sâu của ngành**
3. Quan hệ với các môn học khác  
  - **Môn PLC của ngành Điện công nghiệp – dân dụng, cần được bố trí giảng dạy sau môn học Điện tử cơ bản, thực hành truyền động điện, Tin học cơ bản**
4. Mục đích:
  - Có tầm nhìn tổng quát về hệ thống điều khiển.
  - Hiểu rõ các phương pháp lập trình
  - Từng bước lập trình điều khiển với PLC
  - Luyện tập kỹ năng kết nối từ PLC đến cơ cấu chấp hành.
  - Luyện tập kỹ năng tư duy logic, nhạy bén phán đoán & xử lý các tình huống vận hành thuộc máy điện & các phần liên quan.
5. Số đơn vị học trình:               **03 (45 tiết)**
6. Đánh giá, tính điểm:               **ĐTBMH = [Điểm trung bình kiểm tra + Điểm Thi] : 2**
7. Giáo trình, tài liệu:

**Tài liệu tham khảo chính :**

- Tự động hoá với Simatic S7-200 Nguyễn Doãn Phước – NXB Nông nghiệp – 1997

**Tài liệu tham khảo :**

- Hệ thống Simatic – Trung Tâm Việt Đức – Đại học Sư Phạm Kỹ Thuật – 2003
  - Điều khiển Logic lập trình – Tăng Văn Mùi – NXB Thống kê – 2003
  - Giáo trình đo lường các đại lượng không điện – Vụ THCN – 2003
-

**MỤC LỤC**

	Trang
Chương 1 : Tổng quan về hệ thống điều khiển .....	4
I. Tự động hóa và điều khiển.....	4
II. Tín hiệu tương tự (analog) và tín hiệu số (digital) .....	11
III. Phương pháp điều khiển .....	12
IV. Các phương pháp ổn định.....	13
V. Sơ lược quá trình phát triển của kỹ thuật điều khiển .....	14
VI. Các chủng loại PLC.....	15
VII. Các ứng dụng của PLC.....	15
Chương 2 : Cấu hình hệ thống.....	16
I. Đặt điểm bộ điều khiển lập trình .....	16
II. Cấu trúc phần cứng.....	16
III. Một số khái niệm xử lý thông tin .....	20
IV. Cấu trúc bộ nhớ.....	21
V. Hoạt động của PLC.....	24
Chương 3 : Tập lệnh và các ví dụ ứng dụng .....	25
I. Phương pháp lập trình .....	25
II. Tập lệnh .....	26
1. Lệnh kết thúc chương trình .....	26
2. Lệnh vào tiếp điểm.....	26
3. Lệnh ra tiếp điểm.....	26
4. Các lệnh đặc biệt về tiếp điểm .....	27
5. Bộ Timer .....	31
6. Counter .....	34
7. Nhóm lệnh so sánh : (Compare).....	38
8. Nhóm lệnh di chuyển dữ liệu .....	41

---

9. Nhóm lệnh toán học.....	41
10. Lệnh tăng giảm một đơn vị và lệnh đảo giá trị thanh ghi .....	43
11. Lệnh chuyển đổi các dạng số.....	44
12. Lệnh SHRB .....	44
13. Analog Adjustments .....	45
14. Chương trình con .....	45
15. Đồng hồ thời gian thực. ....	47
16. Hàm Phát xung tốc độ cao.....	48
17. Khai báo và sử dụng bộ đếm tốc độ cao.....	50
18. ANALOG .....	52
Phụ lục : Các từ viết tắt .....	54

---

## CHƯƠNG I : TỔNG QUAN VỀ HỆ THỐNG ĐIỀU KHIỂN

### A. Mục tiêu :

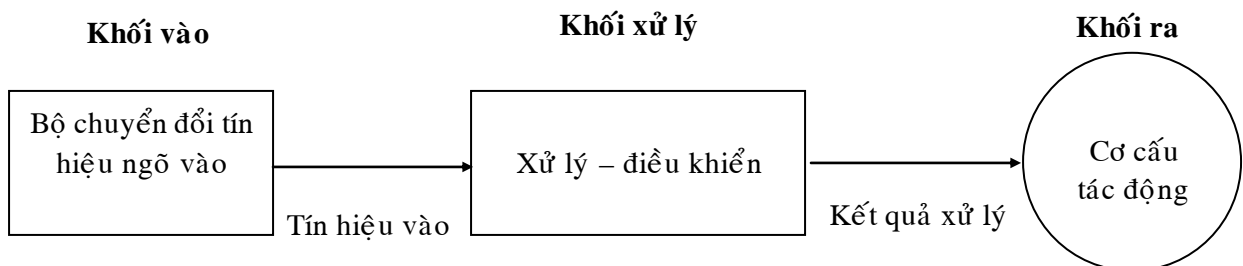
- Trình bày nhiệm vụ, chức năng của các khối trong hệ thống điều khiển.
- Tóm lược được nguyên lý làm việc một số loại cảm biến và cách dùng.
- Tóm tắt một số phương pháp điều khiển
- Trình bày ưu nhược điểm của hệ thống điều khiển dùng rơle, số và PLC

### B. Nội dung :

#### I. Tự động hóa và điều khiển

Tự động hóa nhằm thay thế một phần hoặc toàn bộ thao tác vật lý của người công nhân. Những hệ thống có thể điều khiển quá trình sản xuất với độ tin cậy cao, ổn định mà không cần hoặc cần rất ít sự can thiệp của con người gọi là **hệ thống điều khiển**.

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



Hình 1.1. Sơ đồ khối một hệ thống điều khiển

#### 1. Khối vào :

Là bộ chuyển đổi các đại lượng vật lý thành tín hiệu điện. Các bộ chuyển đổi có thể là nút nhấn, công tắc, cảm biến nhiệt, cảm biến áp suất. ... Tùy theo bộ chuyển đổi mà tín hiệu ra khỏi bộ chuyển đổi có dạng tương tự (analog) hay số ON-OFF (digital)

BỘ CHUYỂN ĐỔI	ĐẠI LƯỢNG ĐO	ĐẠI LƯỢNG RA
Công tắc (Switch)	Sự dịch chuyển/ vị trí	Điện áp ON-OFF
Công tắc hành trình (limit-Switch)	Sự dịch chuyển/ vị trí	Điện áp ON-OFF
Bộ điều chỉnh nhiệt (thermostat)	Nhiệt độ	Điện áp ON-OFF

Cặp nhiệt điện (Thermocouple)	Nhiệt độ	Điện áp thay đổi
Nhiệt trở (Thermistor)	Nhiệt độ	Trở kháng thay đổi
Điện trở đo sức căng (Strain gage)	Áp suất/ sự dịch chuyển	Trở kháng thay đổi
Tế bào quang điện (photocell)	Cường độ sáng	Điện áp thay đổi
Tế bào tiệm cận (Proximity cell)	Sự hiện diện của đối tượng	Trở kháng thay đổi

**Công tắc cơ và nút nhấn cơ :**

Công tắc tạo ra tín hiệu đóng ngắt ( ON-OFF) dưới tác động của lực cơ học. Có một số loại công tắc như : Công tắc đơn 2 vị trí, công tắc 3 vị trí, công tắc đôi 4 vị trí, công tắc đôi 6 vị trí.



Hình 1.2. Một số loại công tắc

Nút nhấn khi còn tác động thì đóng, hết tác động lò xo sẽ đẩy về hoặc ngược lại. Nút nhấn có các loại là thường hở NO, thường đóng NC và nút nhấn đôi bao gồm thường hở và thường đóng.



Hình 1.3. Một số loại nút nhấn

## Công tắc giới hạn

Công tắc giới hạn thường được dùng cho các thiết bị chuyên dùng để phát hiện sự có mặt của chi tiết chuyển động.



Hình 1.4 Công tắc hành trình. (a) tác động 2 phía, (b) tác động 1 phía, (c) nguyên lý

## Bộ cảm biến từ tính

**Công tắc từ tính kiểu dòng điện eddy** (dòng điện foucault) có cuộn dây được cấp nguồn bằng dòng điện xoay chiều và sinh ra từ trường xoay chiều. Khi vật thể bằng kim loại đến gần công tắc dòng điện eddy được sinh ra trong vật thể đó. Từ trường do dòng eddy tác động trở lại lên cuộn dây. Kết quả là biên độ điện áp thay đổi để duy trì dòng điện trong cuộn dây ổn định. Điện áp này có thể được dùng để kích hoạt mạch công tắc điện tử. Khoảng cách phát hiện thường vào khoảng 0.5-20 mm.

**Công tắc từ tính kiểu cảm ứng** : gồm cuộn dây quấn quanh lõi thép. Khi một đầu của lõi thép được đặt gần một vật thể bằng kim loại có chứa sắt, sẽ có sự thay đổi về lượng của lõi thép. Do đó, làm thay đổi về độ cảm ứng của lõi thép. Sự thay đổi này có thể được giám sát bằng mạch cộng hưởng, sự hiện diện của vật thể bằng kim loại sẽ làm thay đổi dòng điện trong mạch. Dòng điện này có thể được dùng để kích hoạt mạch công tắc điện tử tạo thành thiết bị đóng ngắt. Vật thể kim loại có thể bị phát hiện ở khoảng cách 2-15mm.

**Công tắc từ tính kiểu lưỡi gà** : Gồm 2 dải sắt từ đàn hồi, xếp chồng nhưng không tiếp xúc nhau, được gắn vào vỏ thủy tinh. Khi nam châm hay dòng điện đến gần, các lõi sắt sẽ bị từ hóa và hút lẫn nhau, làm các tiếp điểm đóng. Nam châm làm đóng các tiếp điểm khi khoảng cách là 1mm



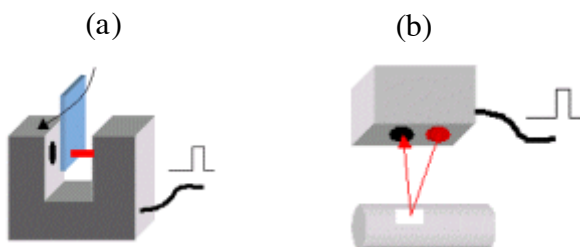
Công tắc thường được dùng nhiều trong thiết bị chống trộm để phát hiện khi cửa bị mở, nam châm gắn trên cửa khi cửa đóng tiếp điểm đóng, khi cửa mở tiếp điểm sẽ mở theo.

**Bộ cảm biến điện dung**

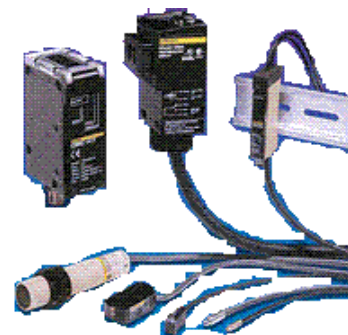
Công tắc điện dung : thường được sử dụng để phát hiện vật thể phi kim loại và kim loại. Điện dung của tụ được xác định bằng khoảng cách của 2 bản cực, khoảng cách càng nhỏ điện dung càng cao. Với vật bằng kim loại thì đầu dò của cảm biến là một bản cực, vật thể bằng kim loại là bản cực còn lại. Với vật thể phi kim loại thì 2 bản cực là đầu cảm biến và dây nối đất, vật phi kim loại là chất điện môi. Khoảng cách phát hiện của công tắc điện dung là 4-60mm.

**Bộ cảm biến quang điện**

Thông thường dùng diode phát quang và diode thu quang hoặc transistor quang. Tia phát được truyền thẳng hoặc phản xạ.



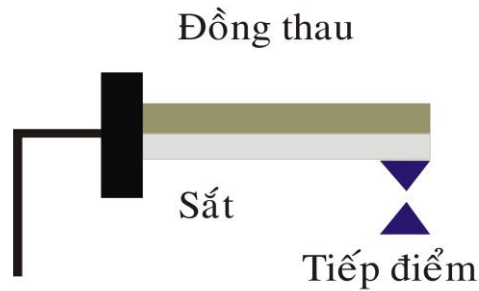
Hình 1.6 Nguyên tắc truyền thẳng (a) và phản xạ (b)



Hình 1.7 Một số cảm biến quang điện

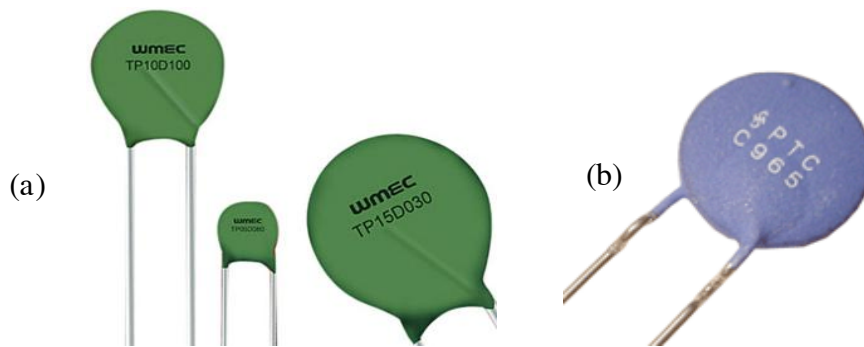
**Bộ cảm biến nhiệt độ**

Thường được dùng và dễ chế tạo là lá lưỡng kim ThermoSwitch. Khi nhiệt độ cao 2 thanh kim loại có hệ số giãn nở nhiệt khác nhau nên lá lưỡng kim cong và làm hở tiếp điểm.



Hình 1.8 Lá lưỡng kim

Thiết bị thứ 2 là điện trở nhiệt RTD gồm có nhiệt trở dương PTC thermistor và nhiệt trở âm NTC thermistor



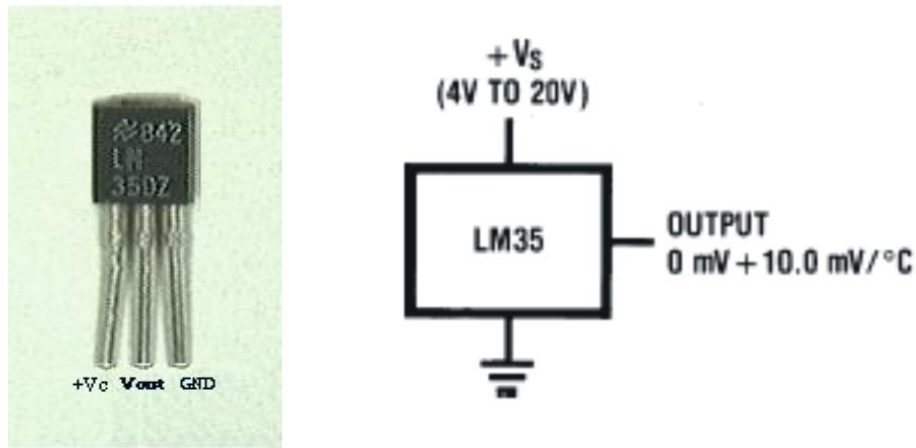
Hình 1.9 NTC thermistor (a), PTC thermistor (b)

Thiết bị thứ 3 là Transistor nhiệt được tích hợp trong IC LM35 điện áp ra thay đổi  $10\text{mV}/^\circ\text{C}$  khi điện áp nguồn là +5V. Hoặc dùng IC tích hợp cho ra mức cao và mức thấp như IC LM3911N

Thiết bị thứ 4 là cặp nhiệt điện :

Cặp nhiệt gồm 2 dây điện khác nhau A và B tạo thành mối nối. Khi mối nối được nung nóng đến nhiệt độ cao hơn các mối nối khác trong mạch (nhiệt độ của mạch không đổi) lực điện động sẽ xuất hiện có quan hệ với nhiệt độ của mối nối nóng. Điện áp do cặp nhiệt điện tạo ra thấp, cần được khuếch đại trước khi cấp cho ngõ vào PLC. Ngoài ra cần có mạch điện bù nhiệt độ cho mối nối nguội, vì nhiệt độ của mối nối nguội ảnh hưởng đến giá trị lực điện động của mối nối nóng tạo ra.





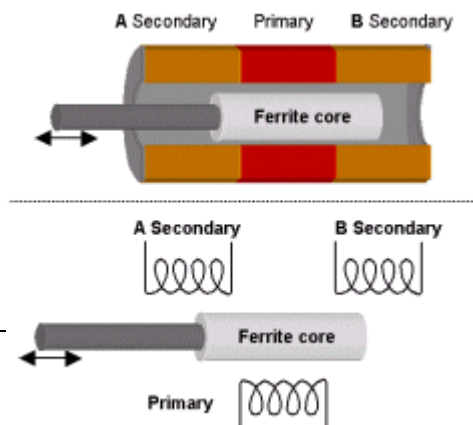
Hình 1.10 IC nhiệt LM 35



Hình 1.11 một số cảm biến cặp nhiệt điện

**Bộ cảm biến khoảng dịch chuyển**

Bộ cảm biến khoảng dịch chuyển thường được dùng là biến áp vi sai biến thiên tuyến tính. Khi lõi sắt ở chính giữa, điện áp trên 2 cuộn dây thứ cấp bằng nhau. Khi lõi sắt dịch chuyển ra khỏi điểm giữa điện áp trên 2 cuộn thứ cấp sẽ khác nhau. Sự chênh lệch điện áp này phụ thuộc vào khoảng cách dịch chuyển của lõi thép.



Hình 1.12 Cảm biến khoảng cách

### **Biến dạng kế**

Khi dây hoặc dải vật liệu bán dẫn bị co hoặc giãn thì điện trở của chúng thay đổi. Sự thay đổi điện trở này phụ thuộc vào độ biến dạng. Biến dạng kế thường được dùng để đo lực tác động hoặc đo áp suất.

### **2. Khối ra :**

Là kết quả của quá trình xử lý của hệ thống điều khiển. Các tín hiệu này được sử dụng để tác động những yêu cầu mong muốn.

THIẾT BỊ NGỒ RA	ĐẠI LƯỢNG RA	ĐẠI LƯỢNG TÁC ĐỘNG
Động cơ điện	Chuyển động quay	Điện
Xylanh-piston	Chuyển động thẳng/ áp lực	Dầu ép / khí nén
Solenoid	Chuyển động thẳng/ áp lực	Điện
Lò sấy / lò cấp nhiệt	Nhiệt	Điện
Rơle/ contactor	Tiếp điểm điện	Điện

### **3. Khối xử lý :**

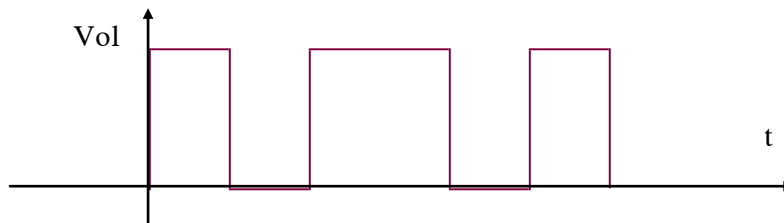
Thân thế người vận hành thực hiện các thao tác. Nó nhận các thông tin ở khối vào và tác động vào khối ra theo một yêu cầu đặt trước. Khối xử lý có thể là một mạch điều khiển đơn giản dùng contactor, mạch điện tử được kết nối “cứng” hoặc bằng chương trình điều khiển như PLC, hệ thống điều khiển dùng máy tính ...

## **II. Tín hiệu tương tự (analog) và tín hiệu số (digital)**

Các thiết bị công nghiệp thường làm việc với tín hiệu số. Với 2 mức logic là mức cao (còn gọi là mức 1) và mức thấp (còn gọi là mức 0).

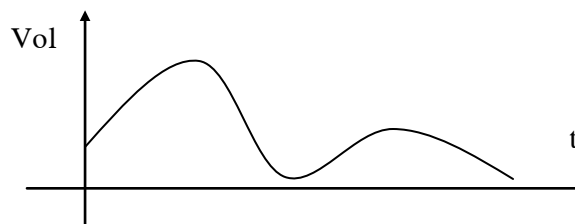
Ví dụ

Mức cao	Mức thấp
Bật	Tắt
Có	Không
Đúng	Sai



Hình 1.13 Tín hiệu số (digital)

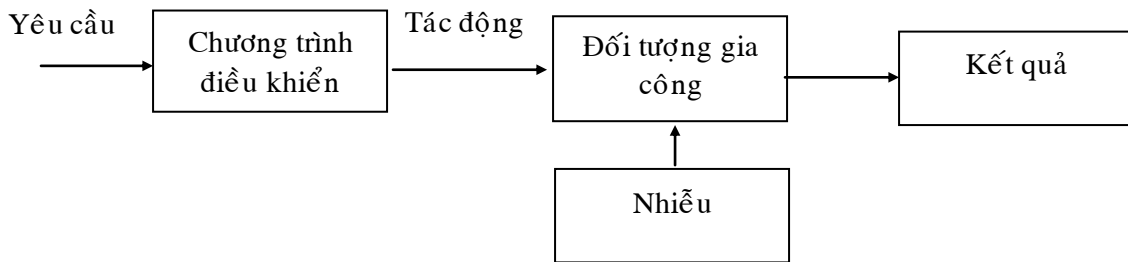
Các đại lượng như : nhiệt độ, cường độ sáng thay đổi từ từ và liên tục qua một dãy vô số các giá trị. Để xử lý tín hiệu số người ta dùng mạch so sánh hoặc dùng bộ chuyển đổi tín hiệu tương tự sang số.



Hình 1.14 Tín hiệu tương tự (analog)

### III. Phương pháp điều khiển

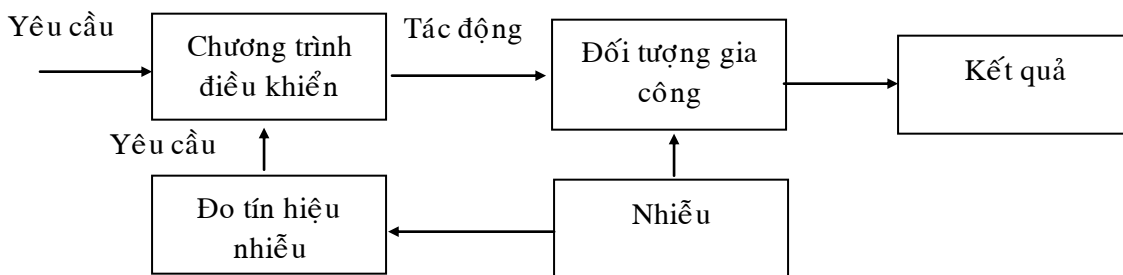
Điều khiển vòng hở :



Hình 1.15 Điều khiển vòng hở

Ở hệ thống này, chương trình tác động vào đối tượng cần gia công mà không quan tâm đến nhiễu. Khi có nhiễu xảy ra nhiễu sẽ tác động làm sai lệch kết quả. Nhiễu trong hệ thống điều khiển có nhiều lý do, có thể do nguồn điện không ổn định, do tải thay đổi, do quá trình khởi động hoặc quá trình dừng ...

Điều khiển kích tiếp :

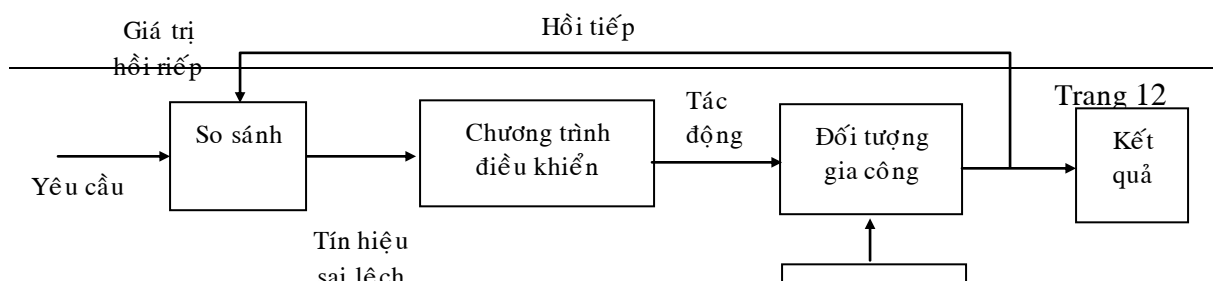


Hình 1.16 Điều khiển kích tiếp

Phương pháp này dùng tín hiệu nhiễu để bù trừ vào tín hiệu điều khiển. Thường được dùng trong các trường hợp “nhiễu” có thể đo được chính xác.

Điều khiển vòng kín :

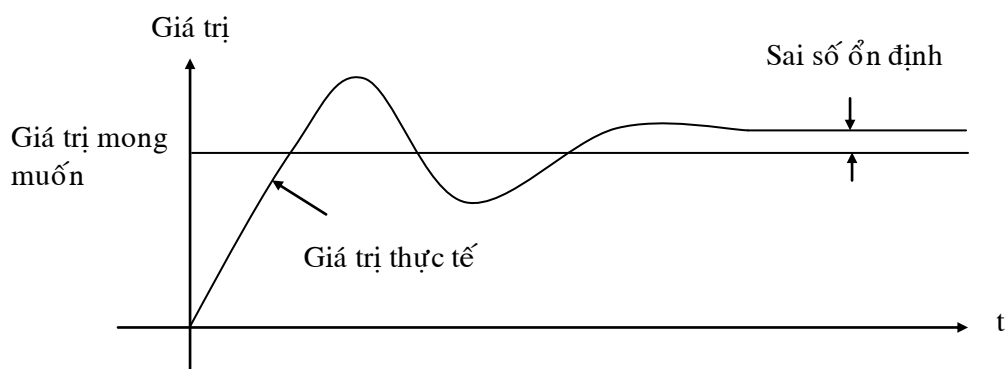
Phương pháp này làm mất tác hại của nhiễu bằng cách đo ảnh hưởng của nhiễu trên tín hiệu ra hay trên sản phẩm.



#### IV. Các phương pháp ổn định

##### Điều khiển tỷ lệ (Proportional action):

Tín hiệu sai lệch hay sai số được dùng làm tín hiệu điều khiển sẽ được nhân với một hệ số tỷ lệ  $K_p$ . Dạng điều khiển này không đủ để hiệu chỉnh hoàn hảo vì Khi ngõ ra gần tới giá trị đặt trước thì sai số sẽ giảm tương ứng tỷ lệ và làm giảm tín hiệu điều khiển. Và dẫn đến sai số ổn định giữa giá trị đặt trước (mong muốn) và giá trị thực tế của hệ thống.



Hình 1.18 Tín hiệu thực tế khác với tín hiệu mong muốn

##### Điều khiển vi phân (Derivative action):

Khâu vi phân cung cấp tín hiệu ngõ ra tỷ lệ với mức thay đổi của sai số. Vì vậy khi sai số tăng nhanh, tác động này tạo ra một tín hiệu điều chỉnh lớn và ngược lại tín hiệu sai số giảm, tác động này lại tạo tín hiệu sai số giảm. Tác động này giúp hệ thống mau trở về trạng thái xác lập nhưng không cải thiện được sai số ổn định của hệ thống.

##### Điều khiển tích (Integral action):

Khâu tích phân ngay khi tác động sẽ tạo nhận ở ngõ vào một tín hiệu và tín hiệu này sẽ duy trì cho đến khi sai số chưa bị triệt tiêu.

**Điều khiển PID :** PID là bộ tích hợp 3 khâu tỷ lệ vi phân và tích phân để kết hợp ưu điểm của 3 khâu điều khiển trên. Ngày nay, bộ điều khiển PID được dùng nhiều trong công nghiệp.

## V. Sơ lược quá trình phát triển của kỹ thuật điều khiển.

### 1. Hệ thống điều khiển dùng rơle.

Hệ thống điều khiển này dùng : contactor, rơle điện từ, bộ định thời, bộ đếm. Hệ thống này được liên kết với nhau để trở thành một hệ thống hoàn chỉnh. Hệ thống dùng rơle rất phức tạp: nhiều dây kết nối, thiết bị công kênh rất khó sửa chữa bảo trì khi hư hỏng, không thể thực hiện được những công việc mang tính phức tạp cao, hơn nữa khi có yêu cầu thay đổi về điều khiển thì bắt buộc phải thiết kế lại và nối dây lại từ đầu.

Tuy nhiên, vẫn được dùng cho những yêu cầu không phức tạp vì giá thành thấp.

### 2. Hệ thống điều khiển dùng vi xử lý :

Hệ thống điều khiển dùng vi xử lý ra đời đã khắc phục được một số nhược điểm của hệ thống dùng rơle như sơ đồ nối dây và một số ưu điểm khác như khả năng nhớ và thực hiện được những chức năng phức tạp mà hệ thống điều khiển bằng rơle không thực hiện được. Thường được dùng trong tủ vi, máy giặt, tủ lạnh, quang báo ...

Tuy nhiên, hệ thống điều khiển bằng vi xử lý vẫn tồn tại một số nhược điểm là : khó lập trình, giao tiếp công suất và vấn đề xử lý nhiễu.

### 3. Hệ thống điều khiển dùng PLC

Với những khó khăn và phức tạp khi thiết kế hệ thống dùng rơle điện, những năm 80, người ta chế tạo ra các bộ điều khiển có lập trình nhằm nâng cao độ tin cậy, ổn định, đáp ứng hệ thống làm việc trong môi trường công nghiệp khắc nghiệt như nhiệt độ cao, độ ẩm lớn ... đem lại hiệu quả kinh tế cao. Đó là bộ lập trình được PLC, được chuẩn hoá theo ngôn ngữ Anh Quốc là Programmable Logic Controller (viết tắt là PLC). **PLC là sự kết hợp của hệ thống điều khiển dùng vi xử lý và rơle.**

**VI. Các chủng loại PLC :**

Hiện nay, có một số nhãn hiệu PLC đang sử dụng trên thị trường Việt Nam:

- Allen Braley (Mỹ).
- Siemens, Boost, Festo (Đức)
- LG (Hàn Quốc)
- Mitsubishi, Omron, Fanuc, Masusita , Fuzi (Nhật).

Tài liệu này chỉ giới thiệu về PLC của Siemens.

**VII. Các ứng dụng của PLC :**

Hiện nay PLC đã được ứng dụng thành công trong nhiều lĩnh vực sản xuất cả trong công nghiệp và dân dụng. Từ những ứng dụng để điều khiển các hệ thống đơn giản, chỉ có chức năng đóng/mở (ON/OFF) thông thường đến các ứng dụng cho các lĩnh vực phức tạp, đòi hỏi tính chính xác cao, ứng dụng các thuật toán trong quá trình sản xuất.

Các lĩnh vực tiêu biểu ứng dụng PLC hiện nay bao gồm :

- *Hóa học và dầu khí:*
- *Chế tạo máy và sản xuất:*
- *Bột giấy, giấy, xử lý giấy*
- *Thủy tinh và phim ảnh:*
- *Thực phẩm, rượu bia, thuốc lá:*
- *Kim loại:*
- *Năng lượng:*

❧

## CHƯƠNG II : CẤU HÌNH HỆ THỐNG

### A. Mục tiêu :

- Liệt kê các đời, các thành phần của PLC S7-200
- Vẽ lại được và giải thích được sơ đồ kết nối
- Liệt kê được tên và ý nghĩa các khái niệm xử lý thông tin
- Liệt kê các miền nhớ và dung lượng mỗi miền
- Liệt kê các phương pháp truy nhập bộ nhớ
- Phân tích cách mở rộng bộ nhớ

### B. Nội dung :

#### I. Đặt điểm bộ điều khiển lập trình.

PLC được viết tắt của Programmable Logic Controller tạm dịch là thiết bị điều khiển lập trình được. Nó cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua một ngôn ngữ lập trình.

#### **PLC có những ưu điểm sau:**

- Độ ổn định, độ tin cậy cao.
- Lắp đặt đơn giản.
- Rất dễ lập trình.
- Đáp ứng yêu cầu đòi hỏi mà không cần thay đổi phần cứng.
- Kích thước nhỏ, gọn.
- Có thể nối mạng vi tính để giám sát hệ thống.
- Điều khiển linh hoạt đa dạng.

#### II. Cấu trúc phần cứng :

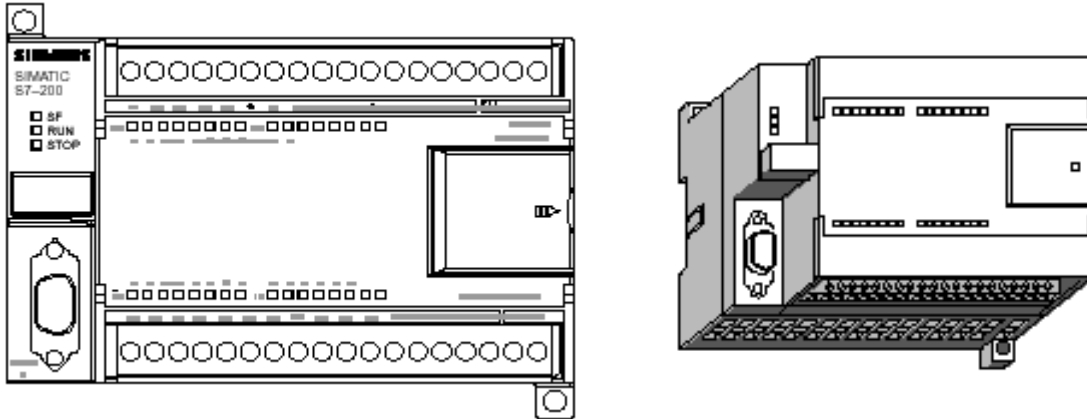
##### 1. Bộ điều khiển lập trình PLC :

PLC của hãng Siemens hiện có các loại sau : S7-200, S7-300, S7 - 400. Do phù hợp với thiết bị thực hành hiện có tài liệu này chỉ đi sâu vào PLC S7 – 200.



**PLC S7-200** có các đời CPU sau : CPU 212, CPU 214, CPU 216, CPU 221, CPU 222, CPU224 ...

Trong tài liệu này trình bày cấu trúc chung **họ S7-200 có CPU 224**.



Hình 2.1 Hình dáng PLC S7-200 CPU 214

## 2. Các thành phần CPU :

### a. CPU 224 bao gồm :

- Dung lượng bộ nhớ chương trình 2K.
- Dung lượng bộ nhớ dữ liệu 2K.
- Có 14 cổng vào 10 cổng ra.
- Có thể thêm vào 7 Modul mở rộng kể cả Modul Analog.
- Có 128 timer, 180 counter, 688 bits nhớ đặc biệt.
- Có các chế độ ngắt và xử lý ngắt.
- Có 3 bộ đếm tốc độ cao.
- Có 2 bộ điều chỉnh tương tự.
- Toàn bộ dung lượng nhớ không bị mất dữ liệu trong thời gian 190 giờ kể từ khi PLC mất điện.

### b. Các đèn báo trên CPU :

- SF : đèn báo hiệu hệ thống bị hỏng ( đèn đỏ ).
- RUN : PLC đang ở chế độ làm việc ( đèn xanh ).

- STOP : PLC đang ở chế độ dừng (đèn vàng).
- Ixx, Qxx: chỉ định trạng thái tức thời ngõ vào và ngõ ra (đèn xanh).

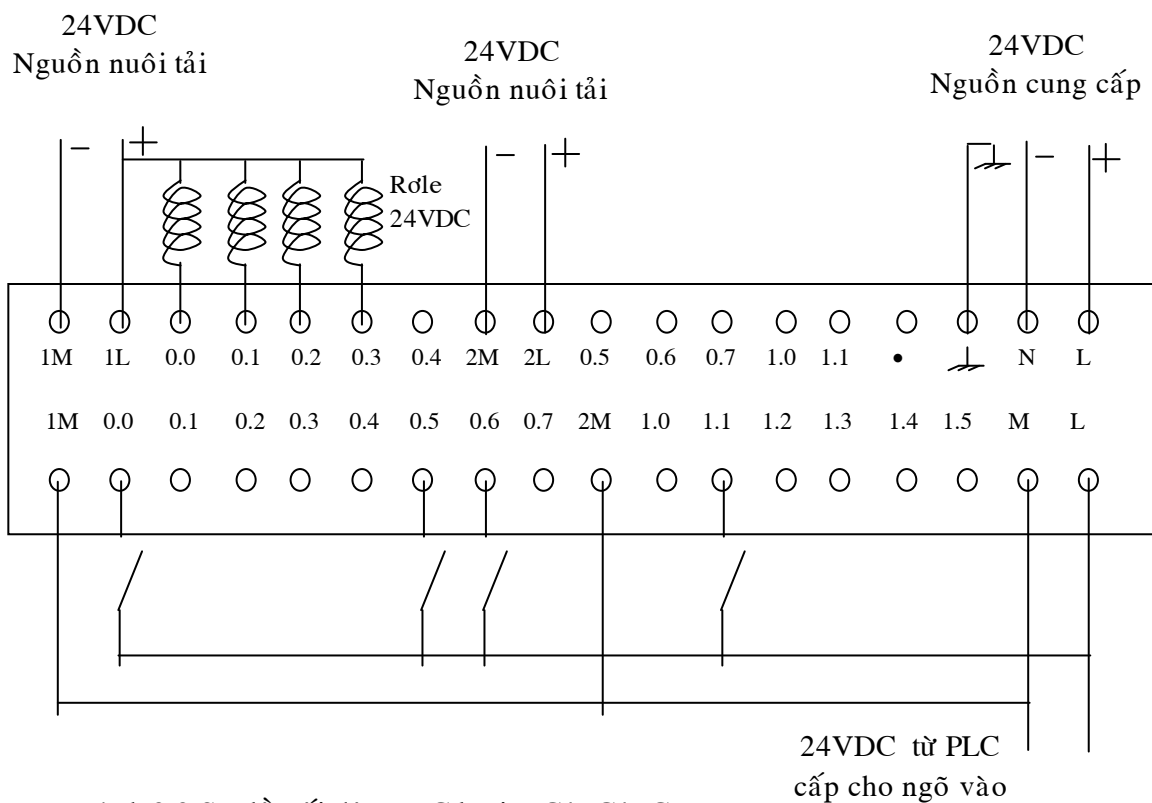
**c. Công tắc chọn chế độ làm việc :**

- RUN : cho phép PLC thực hiện chương trình trong bộ nhớ, PLC sẽ chuyển từ RUN qua STOP nếu gặp sự cố .
- STOP : PLC dừng công việc thực hiện chương trình ngay lập tức.
- TERM : cho phép máy lập trình quyết định chế độ làm việc của PLC. Dùng phần mềm điều chỉnh RUN, STOP (trong quá trình thực hành nên chọn chế độ này)

**3. Kết nối điều khiển :**

**a. Loại DC /DC / DC :**

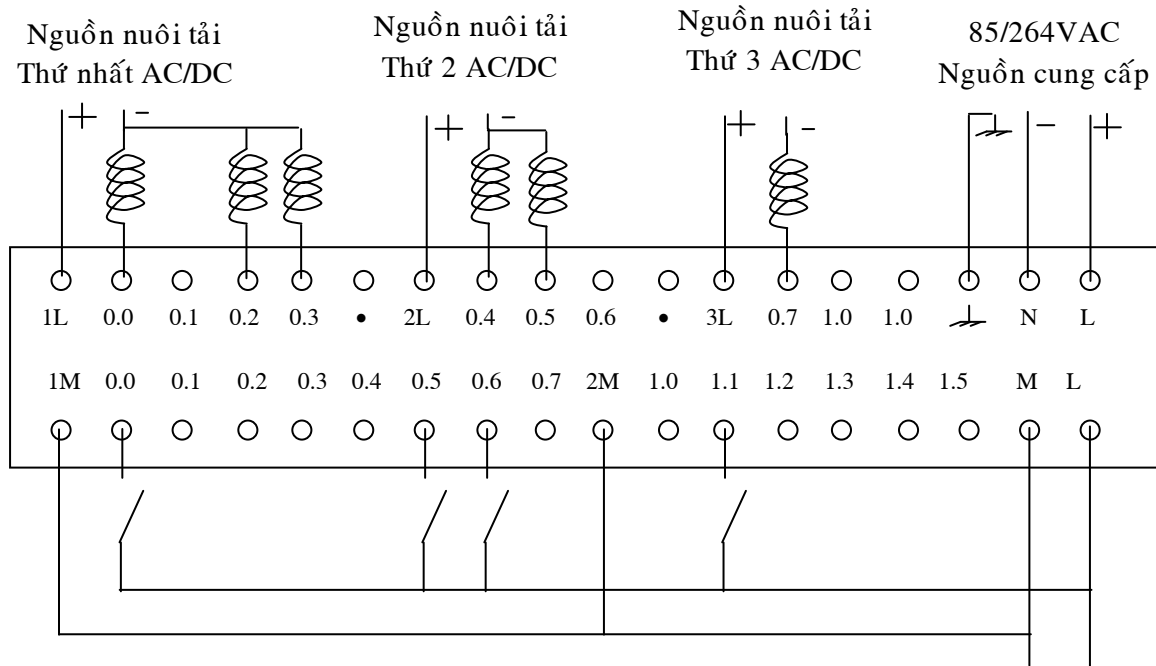
- Nguồn cung cấp 24 VDC.
- Đầu vào số 24 VDC.
- Đầu ra số 24 VDC,  $I_{max} = 0,75A$ .



Hình 2.2 Sơ đồ nối dây PLC loại DC/DC/DC

**b. Loại AC /DC / Rôle:**

- Nguồn cung cấp : 85-264 VAC .
- Đầu vào số : 24 VAC.
- Đầu ra số : 5-30 VDC hoặc 5-250 VAC,  $I_{max} = 2A$ .



Hình 2.3 Sơ đồ nối dây PLC loại AC/DC/rôle

24VDC  
 Nguồn từ PLC cấp  
 cho ngõ vào

**4. Pin và nguồn nuôi bộ nhớ :**

Nguồn nuôi dùng để ghi chương trình hoặc nạp một chương trình mới.

Nguồn pin được sử dụng để tăng thời gian lưu giữ cho các dữ liệu có trong bộ nhớ. Nguồn pin tự động chuyển sang trạng thái tích cực nếu dung lượng tụ nhớ bị cạn kiệt và nó phải thay thế vào vị trí đó để dữ liệu trong bộ nhớ không bị mất đi.

Khi thay pin ta phải cấp nguồn nuôi cho PLC nếu không chương trình trong PLC sẽ mất đi.

**III. Một số khái niệm xử lý thông tin :**

**1. Hệ thống số :**

**a. Số thập phân :**

Hệ thập phân được biểu diễn bằng 10 con số từ 0 đến 9. Trong cuộc sống sinh hoạt hằng ngày ta dùng hệ thống số thập phân.

**b. Số nhị phân :**

Trong xử lý các phần tử nhớ, các cổng vào ra ... PLC sử dụng hệ thống nhị phân .

Hệ nhị phân được biểu diễn bằng hai con số 0 và 1. Số 0 còn được định nghĩa là mức logic thấp, công tắt hay nút nhấn đang hở, trạng thái không làm việc của động cơ, đèn... Số 1 được định nghĩa là mức logic cao, công tắt nút nhấn đang đóng, trạng thái làm việc của động cơ, đèn...

Số nhị phân có thể chuyển thành số thập phân bằng cách :

$$\text{Giá trị} = \sum \text{mức logic} \cdot \text{số mũ của cơ số 2.}$$

Ví dụ : số nhị phân 101 có giá trị là :  $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$  (thập phân)

**c. Số thập lục phân :**

Ngoài sử dụng số nhị phân, PLC còn sử dụng hệ thống số Hex hệ thập lục phân.

Hệ thập lục phân sử dụng 16 con số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Chuyển qua hệ thập phân :

$$\text{Ví dụ : } 2AB = 2 \cdot 16^2 + A \cdot 16^1 + B \cdot 16^0 = 683.$$

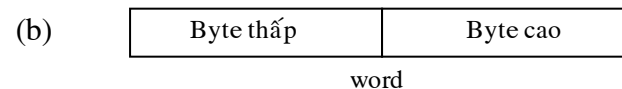
**2. Các khái niệm xử lý thông tin :**

Bit : là giá trị ô nhớ có giá trị logic là 0 hoặc 1.

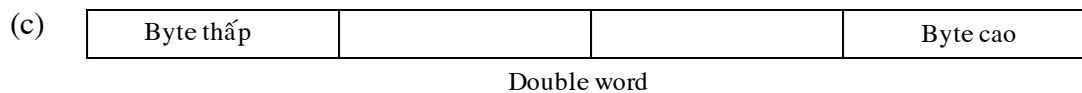
Byte : gồm 8 bit.



Word : là từ nhớ gồm 2 byte.



Double word : gồm có 4 byte.



1Kbyte =  $2^{10}$  byte.

1M byte =  $2^{20}$  byte.

#### IV. Cấu trúc bộ nhớ :

##### 1. Phân chia bộ nhớ :

Bộ nhớ của S7-200 được chia làm 3 vùng :

- ❖ **Vùng nhớ chương trình** : Dùng để lưu trữ các lệnh chương trình.
- ❖ **Vùng nhớ dữ liệu** : lưu trữ dữ liệu chương trình, kết quả phép toán, hằng số được định nghĩa trước. Nó có thể truy nhập theo từng byte, bit hoặc word.
- ❖ **Vùng nhớ thông số** : Lưu trữ các từ khoá, địa chỉ tạm thời.
- ❖ **Vùng dữ liệu** : Lưu giữ dữ liệu chương trình : kết quả phép tính, hằng số được định nghĩa trong chương trình. Là vùng nhớ động nó có thể truy nhập theo từng Bit, Byte, Word và Double Word
  - Miền V (variable) : V0 → V4095
  - Vùng đệm cổng vào I : I0.x → I7.x (x có giá trị từ 0 →7)
  - Vùng đệm cổng ra Q : Q0.x → Q7.x (x có giá trị từ 0 →7)
  - Vùng nhớ nội M ( Internal Memory) : M0.x → M31.x (x = 0 →7 )
  - Vùng nhớ đặc biệt : Special memory bits SM0.x –SM85.x (x =0 →7)
- ❖ **Vùng đối tượng** :
  - Timer : T0 → T255
  - Counter : C0 → C255
  - Vùng đệm cổng vào tương tự AIW0 – AIW62
  - Vùng đệm cổng ra tương tự AQW0 – AQW62
  - Thanh ghi : Ac 0, AC 1 ,AC 2, AC 3.

- Bộ đếm tốc độ cao : high counter HSC0 → HSC4

## 2. Các phương pháp truy nhập :

### a. Truy nhập theo bit :

Tên miền + địa chỉ byte ++ chỉ số bit.

VD : V150.4 : chỉ bit 4 của byte 150 thuộc miền V.

### b. Truy nhập theo byte :

Tên miền + B + địa chỉ byte

VD : VB5 : chỉ byte 5 thuộc miền V.

### c. Truy nhập theo từ :

Tên miền + W + địa chỉ byte thấp của từ trong miền

VD : VW100 : chỉ từ đơn 2 byte 100 và 101 thuộc miền V. Trong đó byte 100 có vai trò byte thấp.

### d. Truy nhập theo từ ghép :

Tên miền + D + địa chỉ byte thấp của từ trong miền

VD : VD150 : chỉ từ ghép gồm 4 byte 150, 151, 152, 153 thuộc miền V, trong đó byte 150 có vai trò là byte thấp.

## 3. Mở rộng vào / ra :

CPU 224 cho phép mở rộng nhiều nhất 7 modul. Các modul tương tự và số đều có trong S7-200.

Có thể mở rộng cổng vào ra của PLC bằng cách ghép nối thêm vào nó các modul mở rộng về phía bên phải của CPU thành một móc xích. Địa chỉ của các modul được xác định bằng kiểu vào ra và vị trí của các modul trong móc xích.

Ví dụ cách đặt địa chỉ cho các modul mở rộng CPU 224.

<b>CPU 224</b>	<b>Modul 0</b> <b>4/4</b>	<b>Modul 1</b> <b>8 vào</b>	<b>Modul 3</b> <b>8 ra</b>	<b>Modul 4</b> <b>3/1 analog</b>
I0.0 Q0.0	I2.0	I3.0	Q4.0	AIW 8
I0.1 Q0.1	I2.1	I3.1	Q4.1	AIW 10
I0.2 Q0.2	I2.2	I3.2	Q4.2	AIW 12
I0.3 Q0.3	I2.3	I3.3	Q4.3	
I0.4 Q0.4	Q2.0	I3.4	Q4.4	AIW 4
I0.5 Q0.5	Q2.1	I3.5	Q4.5	
I0.6 Q0.6	Q2.2	I3.6	Q4.6	
I0.7 Q0.7	Q2.3	I3.7	Q4.7	
I1.0 Q1.0				
I1.1 Q1.1				
I1.2				
I1.3				
I1.4				
I1.5				

## V. Hoạt động của PLC

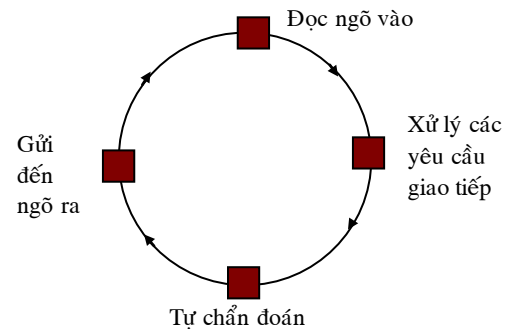
### 1. Cấu trúc chương trình :

Các chương trình cho S7-200 phải có cấu trúc bao gồm : chương trình chính (main program) sau đó đến các chương trình con và các xử lý ngắt.

### 2. Thực hiện chương trình:

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 :

- Đọc các ngõ vào.
- Xử lý yêu cầu.
- CPU tự kiểm tra và chẩn đoán lỗi.
- Chuyển dữ liệu từ bộ đếm đến các ngõ ra.



Hình 2.5 Chu trình làm việc của PLC

### 3. Quy trình thiết kế hệ thống điều khiển dùng PLC :

Để chương trình gọn gàng, dễ quan sát và không nhầm lẫn địa chỉ trong quá trình soạn thảo chương trình cần thực hiện các yêu cầu sau :

- Hiểu rõ nguyên lý hoạt động của hệ thống.
- Xác định có bao nhiêu tín hiệu vào ra.
- Lập bảng phân phối nhiệm vụ I / O.
- Xây dựng giải thuật.
- Lập chương trình.
- Chạy thử (test)



## CHƯƠNG IV : TẬP LỆNH VÀ CÁC VÍ DỤ ỨNG DỤNG

### A. Mục tiêu :

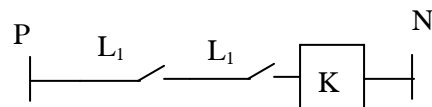
- Phân biệt sự khác nhau giữa ba phương pháp lập trình
- Hiểu ý nghĩa câu lệnh, biết cách lập trình một số ví dụ nhỏ
- Biết đọc một sơ đồ điện dùng PLC

### B. Nội dung :

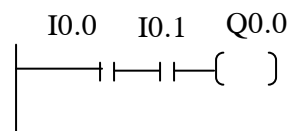
#### I. Phương pháp lập trình :

Cách lập trình cho S7- 200 nói riêng và cho các PLC của Siemen nói chung dựa trên hai phương pháp cơ bản : Phương pháp hình thang (Ladder Logic viết tắt là LAD) và phương pháp liệt kê lệnh (Statement List viết tắt là STL). Nếu chương trình được viết trong LAD thì thiết bị lập trình sẽ tạo một chương trình theo kiểu STL tương ứng. Ngược lại không phải mọi chương trình viết trong STL đều chuyển sang LAD được.

Ví dụ ta có mạch điện như sau :



#### ❖ Lập trình theo LAD



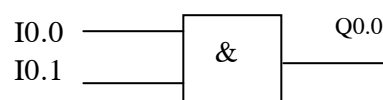
#### ❖ Lập trình theo STL

LD I0.0

A I0.1

= Q0.0

#### ❖ Lập trình theo FBD.



Phần lập trình theo phương pháp LAD để sử dụng và theo dõi được hoạt động của mạch. Vì vậy tài liệu này chỉ đề cập sâu đến lập trình theo phương pháp LAD

**II. Tập lệnh :**

**1. Lệnh kết thúc chương trình :**

♣ Ký hiệu :  $\text{---}\{ \text{End} \}$

**2. Lệnh vào tiếp điểm :**

b. Tiếp điểm thường hở (NO) :

♣ Ký hiệu :  $\text{---}| \text{---}$ <sup>n</sup>

Tiếp điểm thường hở sẽ đóng khi n có giá trị Logic bằng 1

Toán hạng I, Q, M, SM, T, C, V.

c. Tiếp điểm thường đóng (NC)

♣ Ký hiệu :  $\text{---}+| \text{---}$ <sup>n</sup>

Tiếp điểm thường đóng sẽ mở khi n có giá trị logic bằng 1.

Toán hạng I, Q, M, SM, T, C, V.

**3. Lệnh ra tiếp điểm :**

a. Lệnh OUT  $\text{---}\{ \text{ } \}$ <sup>n</sup>

♣ Ký hiệu :

Tiếp điểm sẽ đóng khi có dòng điện chạy qua.

Toán hạng I, Q, M, SM, T, C, V.

b. Lệnh SET và RESET

♣ Ký hiệu :  $\text{---}\left( \begin{matrix} n \\ S \\ i \end{matrix} \right)$   $\text{---}\left( \begin{matrix} n \\ R \\ i \end{matrix} \right)$

Lệnh Set sẽ đặt i bit kể từ bit thứ n lên mức logic 1.

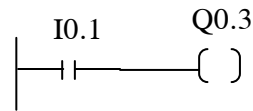
Lệnh reset sẽ đặt i bit kể từ bit thứ n xuống mức logic 0.

Toán hạng I, Q, M, SM, T, C, V. i = 1 ÷ 255

**Ví dụ 1:**

Tác động bit I0.1 lên mức cao thì ngõ ra Q0.3 lên mức cao và ngược lại.

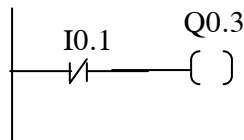
Chương trình:



**Ví dụ 2 :**

Tác động bit I0.1 lên mức cao thì ngõ ra Q0.3 xuống mức thấp và ngược lại.

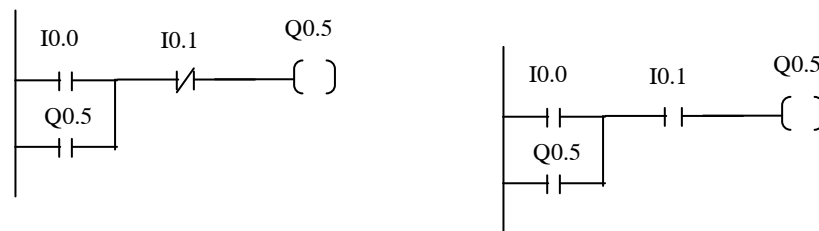
Chương trình:



**Ví dụ 3 :**

Viết chương trình điều khiển động cơ có yêu cầu như sau : Nhấn nút ON (I0.0) thì động cơ làm việc, nhấn nút OFF (I0.1) thì động cơ dừng.

Chương trình:



Lưu ý : Nút nhấn ON/OFF sử dụng trong chương trình là nút nhấn thường hở NO. Nếu quen dùng nút nhấn OFF là nút nhấn thường đóng NC thì chương trình phải viết lại như sau :

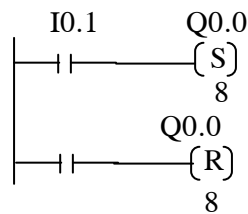
Do vậy, khi viết chương trình với PLC ta cần chú ý 2 khái niệm : thiết bị và tiếp điểm.  
 Ví dụ : nút nhấn thường hở NC là thiết bị, thiết bị này có hai trạng thái làm việc là tác động (mức 1) hoặc không tác động (mức 0). Với hệ thống điều khiển bằng điện ta thường cho nó có 1 tiếp điểm là thường hở :  $\text{---| |---}$  khi tác động sẽ đóng lại. Với PLC thì ta phải quan niệm rằng nó là một cặp tiếp điểm bao gồm một tiếp điểm thường hở  $\text{---| |---}$  và một tiếp điểm thường đóng  $\text{---|/|---}$  tiếp điểm thường hở sẽ đóng khi được tác động, tiếp điểm thường đóng sẽ mở khi được tác động và ngược lại.

Do vậy, khi sử dụng PLC người ta thay nút nhấn thường hở bằng nút nhấn thường đóng và viết chương trình như với hệ thống điện.

**Ví dụ 4 :** Nhấn nút I0.5 thì 8 bit từ Q0.0 đến Q0.7 lên mức cao.

Nhấn nút I0.6 thì 8 bit từ Q0.0 đến Q0.7 xuống mức thấp.

Chương trình :



**4. Các lệnh đặc biệt về tiếp điểm :**

a. SM0.0

Ký hiệu :  $\text{SM0.0} \text{ ---|---}$

Tiếp điểm luôn luôn đóng.

b. SM0.1

Ký hiệu :  $\text{SM0.1} \text{ ---|---}$

Tiếp điểm cho ra một xung khi PLC chuyển từ Stop sang Run.

c. SM0.4

Ký hiệu :  $\text{SM0.4} \text{ ---|---}$

Tiếp điểm phát xung nhịp chu kỳ 60s

d. SM0.5

Ký hiệu :  $\text{SM0.5} \text{ ---|---}$

Tiếp điểm phát xung chu kỳ 1s

e. Tiếp điểm phát hiện cạnh lên.

Ký hiệu :  $\text{---| P |---}$

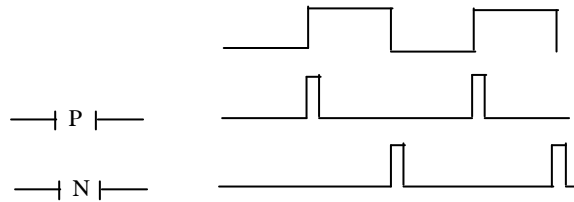
Khi đầu vào lên mức cao thì tiếp điểm cho ra một xung.

f. Tiếp điểm phát hiện cạnh xuống

Ký hiệu :  $\text{---| N |---}$

Khi đầu vào xuống mức thấp thì tiếp điểm sẽ cho ra một xung. Độ rộng xung bằng một chu kỳ quét.

Giản đồ thời gian :



**Ví dụ 5 :** Mạch đảo chiều quay động cơ xoay chiều 3 pha.

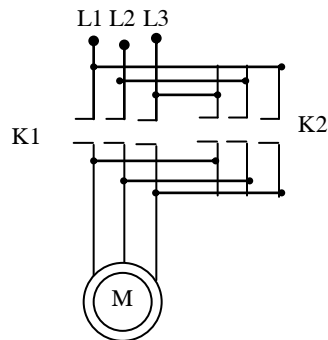
Mô tả : Nhấn nút Start động cơ quay cùng chiều kim đồng hồ. Nhấn nút Reverse động cơ quay ngược chiều kim đồng hồ. Nhấn nút Stop động cơ dừng hoạt động. Khi động cơ đang quay thuận thì không được phép quay nghịch mà chỉ khi dừng mới được phép quay nghịch.

Yêu cầu : Lập bản phân phối nhiệm vụ.

Viết chương trình điều khiển.

**Bài giải mẫu :**

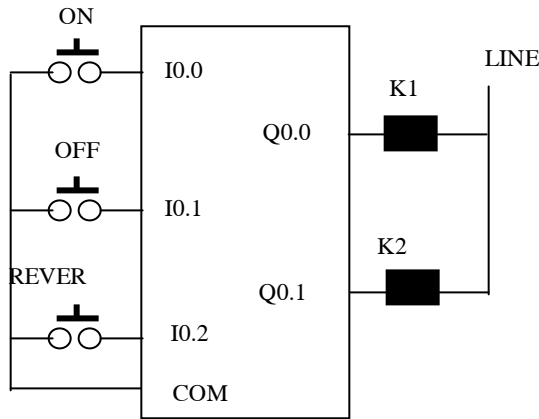
Mạch động lực :



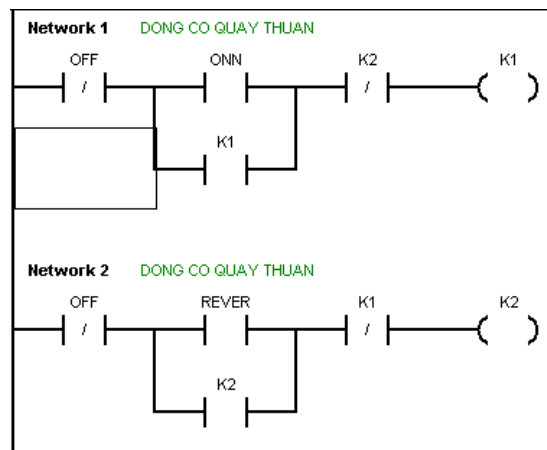
Bảng phân phối nhiệm vụ ( Symbol Table)

Symbol Table				
	Symbol	Address	Com	
1	ONN	I0.0	NUT NHAN ON	
2	OFF	I0.2	NUT NHAN OFF	
3	REVER	I0.2	NUT NHAN REVERSE	
4	K1	Q0.0	KHOI DONG TU QUAY THUAN	
5	K2		KHOI DONG TU QUAY NGHICH	

Sơ đồ nối dây :



Chương trình :



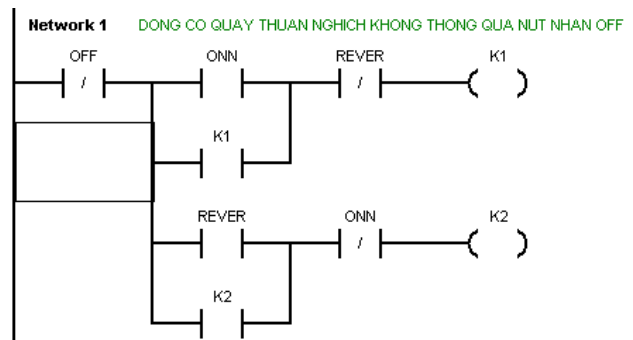
**Ví dụ 6 :** Mạch đảo chiều quay động cơ xoay chiều 3 pha.

Mô tả : Nhấn nút Start động cơ quay cùng chiều kim đồng hồ. Nhấn nút Reverse động cơ quay ngược chiều kim đồng hồ. Nhấn nút Stop động cơ dừng hoạt động. Khi động cơ đang quay thuận mà nhấn nút quay nghịch thì phải quay nghịch.

Yêu cầu : Lập bảng phân phối nhiệm vụ.

Viết chương trình điều khiển.

Chương trình :



Lưu ý : Với hệ thống điều khiển bằng điện, khi dùng nút nhấn có cùng chức năng nhưng ở 2 vị trí khác nhau ta phải dùng nút nhấn đôi. Với hệ thống dùng PLC ta chỉ dùng nút nhấn đơn là đủ rồi. Trong chương trình cho phép ta sử dụng tới 255 tiếp điểm của 1 thiết bị.

**Ví dụ 7 :** Mạch điều khiển trò chơi.

Mô tả : Có 3 đấu thủ và một người điều khiển chương trình. Trước mặt mỗi đối thủ có một nút nhấn. Khi có câu hỏi, đấu thủ sẽ bấm nút trước mặt và được quyền ưu tiên trả lời. Lúc này đèn trước mặt đối thủ đó sẽ sáng và các đối thủ còn lại bấm nút đèn không sáng. Đèn của đối thủ sẽ được tắt bởi người dẫn chương trình.

Yêu cầu : Lập bảng phân phối nhiệm vụ.

Viết chương trình điều khiển.

**Ví dụ 8 :** Mạch điều khiển thang máy xây dựng.

Mô tả : Thang máy xây dựng được điều khiển để chuyển vật tư, thiết bị từ thấp lên cao và ngược lại. Nhấn nút Up thì motor nâng gầu lên, gặp công tắc hành trình motor dừng lại. Nhấn nút Down thì motor hạ gầu xuống, gặp công tắc hành trình motor dừng. Khi nâng gầu lên hoặc xuống nhấn nút Stop thì gầu dừng, lúc này có thể nâng gầu lên hoặc xuống tùy ý. Trong khi gầu đang chuyển động thì đèn báo luôn chớp.

Yêu cầu : Lập bản phân phối nhiệm vụ.

Viết chương trình điều khiển.

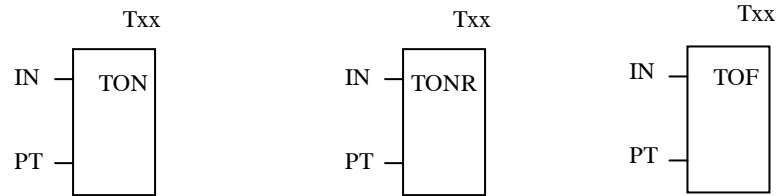
### 5. Bộ Timer :

Timer là bộ tạo thời gian trễ giữa tín hiệu đầu ra so với đầu vào

Trong S7 –200 CPU 224 có 128 bộ timer chia làm thành hai loại khác nhau.

- Timer tạo thời gian trễ không nhớ TON (On Delay Timer)
- Timer tạo thời gian trễ có nhớ TONR (Retentive On Delay Timer)
- Timer tạo thời gian trễ không nhớ TOF (Off Delay Timer)

Ký hiệu :



Trong đó :

PT : Preset timer (0 - 32767)

Txx: T0 – T255

Toán hạng : VW, T, C, IW, MW ...

Bộ TON và bộ TONR được chia thành 3 vùng với độ phân giải khác nhau : 1ms, 10ms, 100ms.

Độ phân giải	TON	TOF	TONR
1ms	T32 , T96	T32 , T96	T0 , T64
10ms	T33 → T36	T33 → T36	T1 → T4
100ms	T97 → T100	T97 → T100	T65 → T68
	T37 → T63	T37 → T63	T69 → T95
	T101 → T225	T101 → T225	T5 → T31

Hoạt động :

Cả hai loại timer TON và TONR tạo thời gian trễ khi tín hiệu đầu vào lên mức cao. Nếu giá trị tức thời lớn hơn hoặc bằng giá trị đặt trước thì ngõ ra tương ứng của bộ định thời được kích hoạt.

Tín hiệu đầu vào xuống mức thấp thì TON tự động Reset còn TONR thì không tự động Reset mà cần đến một tín hiệu tác động.

Lệnh Reset là phương pháp duy nhất để đưa bộ TONR về mức thấp.

TOF tạo thời gian trễ khi tín hiệu đầu vào xuống mức thấp. Tín hiệu đầu vào lên mức cao thì TOF tự động Reset



**Ví dụ 9 :** Mạch khởi động động cơ sao – tam giác.

Nhấn nút Start để khởi động động cơ, để đảm bảo an toàn chuông sẽ reo trong 10s và đèn chớp đúng 10 lần. Sau đó động cơ hoạt động ở chế độ đấu sao. Sau một khoảng thời gian định trước đủ để động cơ đạt tốc độ nhất định, đng65 cơ chuyển sang hoạt động ở chế độ đấu tam giác.

Khi nhấn nút Stop động cơ dừng hoạt động.

**Ví dụ 10 :** Mạch đảo chiều quay động cơ.

Động cơ xoay chiều 3 pha sẽ đảo chiều quay khi chúng ta thực hiện đảo chiều 2 trong 3 pha của chúng lại với nhau. Khi nhấn nút Start động cơ quay theo chiều thuận được 10s động cơ dừng, sau 2s động cơ tự chạy ngược sau 30s động cơ dừng.

Nếu động cơ đang chạy nhấn nút dừng (Stop) động cơ dừng và chu kỳ mới được lập lại.

**Ví dụ 11 :** Mạch điều khiển tuần tự 3 máy bơm.

Hệ thống 3 máy bơm được thiết lập theo chế độ sau, mỗi máy bơm hoạt động trong 50 giây.

Nhấn nút Start động cơ hoạt động theo trình tự. Nhấn nút Stop 3 động cơ dừng.

Máy 1	Máy 2	Máy 3
1	0	0
0	1	0
0	0	1

**Ví dụ 12 :** Mạch khởi động tuần tự các động cơ.

Khi nhấn nút On thì M1 chạy để bơm nước. Sau 5s thì M2 tự động chạy để bơm dầu. Sau 8s thì M3 tự động chạy ở chế độ không tải. Động cơ M3 chạy không tải 3s và nếu áp suất khí trong bình chưa đạt mức giới hạn thì van điện từ Y có điện để nạp khí và o bình chứa, M3 chạy ở chế độ có tải.

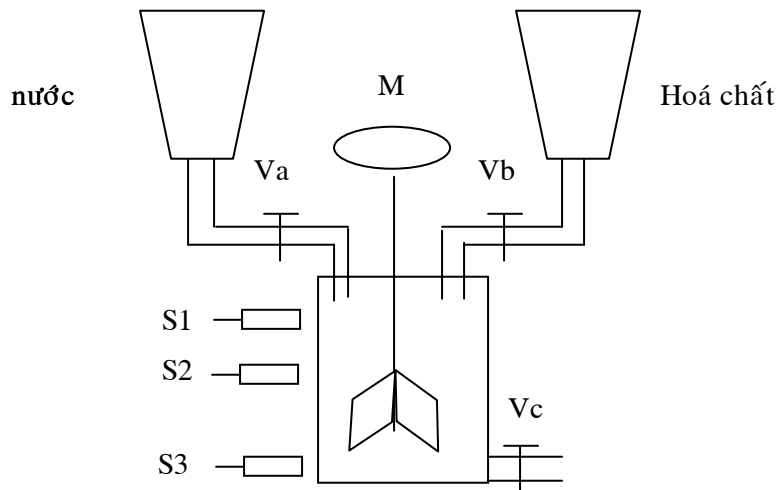
Khi quá áp suất trong bình chứa, M3 trở lại chế độ không tải.

Khi nhấn nút Off thì cả 3 động cơ đều dừng

**Ví dụ 13 :** Viết chương trình điều khiển mô hình máy pha trộn chất lỏng có yêu cầu như sau :

- Nhấn nút On hệ thống hoạt động.

- Khi hoạt động Va bơm nước vào bồn, khi nước đến S2 thì Va dừng Vb hoạt động, khi nước đến S3 thì Vb dừng, động cơ M quay 10s thì dừng Vc hoạt động đổ nước ra ngoài, khi nước xuống đến S1 thì Vc đóng Va mở để chu trình mới lập lại.



Hình 3.1 Mô hình bộ trộn chất lỏng

- Nhấn nút Off hệ thống dừng.
- Nhấn nút On trở lại thì chương trình đang hoạt động ở đâu thì làm việc tại đó.

**Ví dụ 14 :** Viết chương trình điều khiển hệ thống sau : Nhấn nút On động cơ 1 làm việc, 5s sau động cơ 2 làm việc, 5s sau động cơ 3 làm việc. Nhấn nút OFF động cơ 1 dừng, 5s sau động cơ 2 dừng, 5s sau động cơ 3 dừng.

**Ví dụ 15 :** Viết chương trình điều khiển hệ thống sau : Nhấn nút On động cơ 1 làm việc, 5s sau động cơ 2 làm việc, 5s sau động cơ 3 làm việc. Nhấn nút OFF động cơ 3 dừng, 5s sau động cơ 2 dừng, 5s sau động cơ 1 dừng.

## 6. Counter :

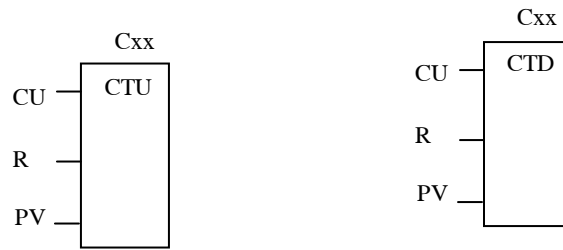
Khái niệm :

Counter là bộ đếm thực hiện chức năng đếm sườn lên của xung trong S7 – 200. Các bộ đếm trong CPU 224 chia làm 2 loại :

- Bộ đếm lên CTU (counter up)
- Bộ đếm xuống CTD (counter down)

**a. Bộ đếm lên và bộ đếm xuống.**

Ký hiệu :



Trong đó : Cxx : C0 ↔ C47 hoặc C80 ↔ C127

CU : đầu vào đếm lên

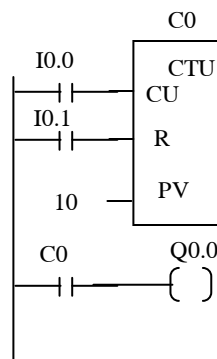
R : đầu vào reset

PV : Preset value (0 ↔ 32767)

Hoạt động :

- CTU thực hiện đếm lên theo số sườn lên của tín hiệu logic đầu vào CTU
- Số sườn xung đếm được lưu vào thanh ghi của bộ đếm có độ dài 2 byte và so sánh với giá trị đặt trước PV.
- Khi giá trị đếm được lớn hơn hoặc bằng giá trị đặt trước thì bộ đếm xuất tín hiệu ứng với mức cao.
- Khi đầu vào R lên mức cao thì bộ đếm được Reset

Chương trình đếm xung đầu vào :

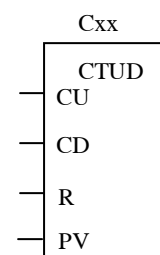


**b. Bộ đếm lên / xuống**

Ký hiệu :

Đặc điểm : Cxx : C48 ↔ C79

CU : Đầu vào đếm lên

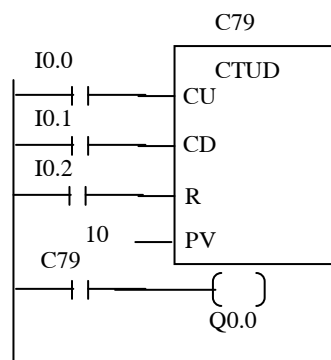


- CD : Đầu vào đếm xuống
- R : Đầu vào Reset
- PV : reset Value ( - 32767 ↔ 32767 )

Hoạt động :

- CTU thực hiện đếm lên theo số sườn lên của tín hiệu logic đầu vào CU
- CTU thực hiện đếm xuống theo số sườn lên của tín hiệu logic đầu vào CD
- Số sườn xung đếm được lưu vào thanh ghi của bộ đếm có độ dài 2 byte và so sánh với giá trị đặt trước PV
- Khi giá trị đếm được lớn hơn hoặc bằng giá trị đặt trước thì bộ đếm xuất hiện tín hiệu ứng với mức cao.
- Khi đầu vào R lên mức cao thì bộ đếm bị Reset.

Chương trình đếm lên và đếm xuống :



**Ví dụ 16 :** Kết hợp nhiều bộ đếm để tăng giá trị đếm.

Ví dụ muốn có số đếm 1000.000 thì một bộ đếm không cho phép đặt đủ số đếm quy định. Do đó, ta dùng 2 bộ đếm với số đếm là  $100.000 = 50 \cdot 2.000$

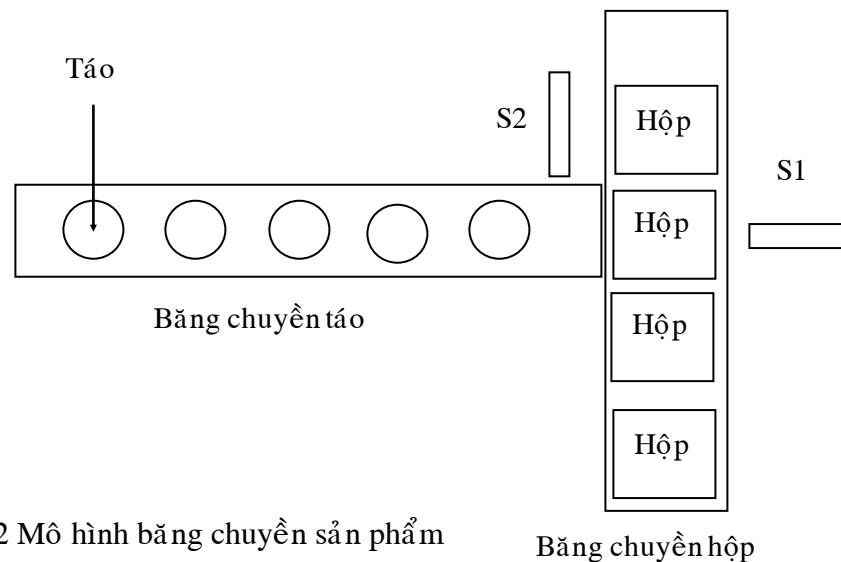
**Ví dụ 17 :** Mạch nhấp nháy có số lần quy định trước.

Khi nhấn nút ON thì đèn báo nhấp nháy 12 lần rồi dừng. Nếu đang nháy nhấn nút OFF thì dừng, nhấn nút ON thì chu kỳ mới lập lại

**Ví dụ 18 :** Lập bảng phân phối nhiệm vụ, vẽ kết nối phần cứng và viết chương trình điều khiển hệ thống bằng chuyên có yêu cầu như sau :

- Khi nhấn nút ON thì băng chuyền hoạt động để kéo hộp.

- Khi hộp vào vị trí (Cảm biến S1 lên mức cao) thì băng chuyền hộp dừng. Sau 2 giây băng chuyền tảo hoạt động để chuyển tảo vào hộp.
- Cảm biến 2 (S2) sẽ đếm số lượng tảo. Nếu số tảo đủ 10 thì băng chuyền tảo dừng. Sau 2 giây băng chuyền hộp hoạt động.
- Hệ thống liên tục hoạt động cho đến khi nhấn nút OFF thì hệ thống dừng.
- Nếu vì sự cố như mất điện (không nhấn nút OFF) hệ thống dừng thì khi nhấn nút ON hệ thống hoạt động lại ngay tại vị trí mất điện.



Hình 3.2 Mô hình băng chuyền sản phẩm

Băng chuyền hộp

### Ví dụ 19 : Kiểm soát chỗ cho Garaga ngầm.

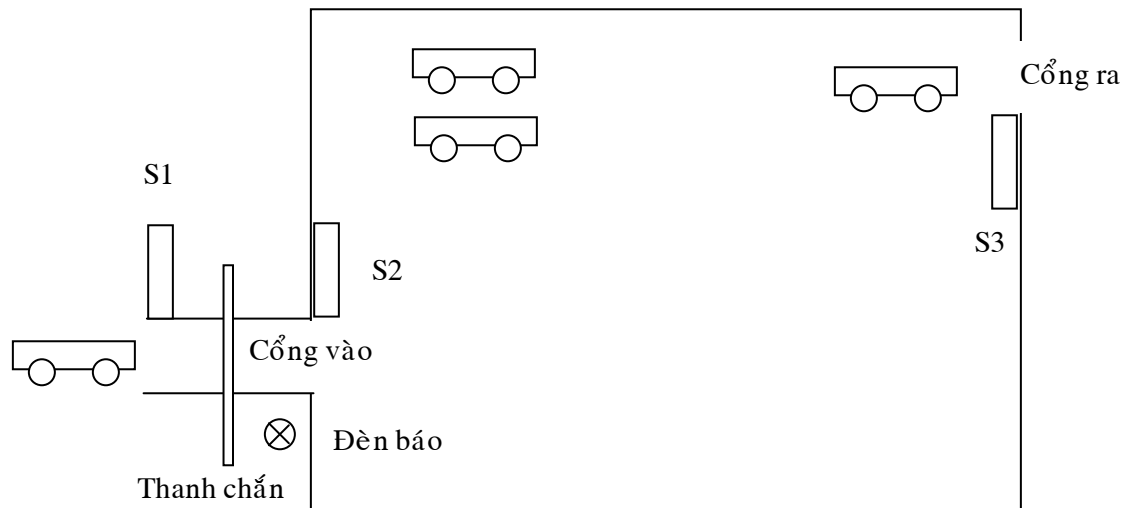
Garaga ngầm chỉ cho phép tối đa 100 xe đỗ tại một thời điểm. Mỗi khi có một xe mới vào sensor 1 phát hiện và đếm lên. Khi xe ra thì sensor 2 phát hiện và bộ đếm giảm. Bình thường đèn xanh sáng, và khi đầy xe đèn đỏ sáng các xe không được vào bãi.

### Ví dụ 20 :

Lập bảng phân phối nhiệm vụ, vẽ kết nối phần cứng và viết chương trình điều khiển hệ thống kiểm tra số lượng xe trong nhà xe có yêu cầu như sau :

- Khi có xe vào cổng thì cảm biến 1 (S1) sẽ nhận biết và mở thanh chắn với điều kiện số lượng xe trong nhà xe dưới 100.
- Khi xe vào đến cảm biến 2 (S2) thì sau 20 giây thanh chắn đóng lại.
- Có một xe vào cảm biến 2 (S2) sẽ đếm số lên.

- Có một xe ra cảm biến 3 (S3) sẽ đếm xuống.
- Nếu số lượng xe trong nhà xe vừa đủ 100 thì thanh chắn không mở, không cho xe vào nhà xe, đồng thời đèn báo sẽ nhấp nháy.
- Thanh chắn mở ra gặp tiếp điểm hành trình S4 thì dừng. Thanh chắn đóng lại gặp tiếp điểm hành trình S5 thì dừng.



Hình 3.4 Mô hình kiểm tra chỗ trong nhà xe

**7. Nhóm lệnh so sánh : (Compare)**

Khi lập trình nếu có các quyết định được thực hiện dựa trên kết quả của việc so sánh thì có thể sử dụng các lệnh so sánh theo byte, word, double word, real trong S7- 200.

Các phép so sánh thường gặp trong toán học là : so sánh bằng, so sánh lớn hơn hoặc bằng, so sánh nhỏ hơn hoặc bằng.

**a. Phép so sánh bằng :**

Ký hiệu :

$$\begin{array}{c} n_1 \\ \text{---} \text{==} \text{D} \text{---} \\ n_2 \end{array} \quad \begin{array}{c} n_1 \\ \text{---} \text{==} \text{I} \text{---} \\ n_2 \end{array} \quad \begin{array}{c} n_1 \\ \text{---} \text{==} \text{R} \text{---} \\ n_2 \end{array} \quad \begin{array}{c} n_1 \\ \text{---} \text{==} \text{B} \text{---} \\ n_2 \end{array}$$

Tiếp điểm đóng khi  $n_1 = n_2$

$n_1, n_2$  là số thực, counter, timer ...

D, I, R, B, lần lượt là Double word, số tự nhiên, số thực, Byte.

**b. Phép so sánh lớn hơn hoặc bằng :**

Ký hiệu :

$$\begin{array}{cccc} \text{---} | \begin{array}{c} n_1 \\ \geq D \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ \geq I \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ \geq R \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ \geq B \\ n_2 \end{array} | \text{---} \end{array}$$

Tiếp điểm đóng khi  $n_1 \geq n_2$

c. Phép so sánh nhỏ hơn hoặc bằng :

Ký hiệu

$$\begin{array}{cccc} \text{---} | \begin{array}{c} n_1 \\ \leq D \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ \leq I \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ \leq R \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ \leq B \\ n_2 \end{array} | \text{---} \end{array}$$

Tiếp điểm đóng khi  $n_1 \leq n_2$

d. Phép so sánh lớn hơn, nhỏ hơn.

$$\begin{array}{cccc} \text{---} | \begin{array}{c} n_1 \\ < D \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ < I \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ < R \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ < B \\ n_2 \end{array} | \text{---} \\ \text{---} | \begin{array}{c} n_1 \\ > D \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ > I \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ > R \\ n_2 \end{array} | \text{---} & \text{---} | \begin{array}{c} n_1 \\ > B \\ n_2 \end{array} | \text{---} \end{array}$$

**Ví dụ 21 :** Mạch điều khiển đèn giao thông. Khi PLC chuyển từ Stop sang Run, đèn đỏ tuyến 1

và 2 sáng. Sau 3s đèn xanh 2 chuyển sang vàng sau 1s vàng chuyển sang đỏ và đồng thời đỏ 1 chuyển sang xanh 1. sau 3s xanh 1 chuyển sang vàng 1. sau 1s thì chu kỳ mới lặp lại

Vẽ giản đồ thời gian và viết chương trình điều khiển.

Khi tăng thời gian làm việc của các đèn lên là : xanh 28, vàng 4, đỏ 32 thì làm thế nào ?

**Ví dụ 22 :** Điều khiển cánh tay máy

Cánh tay máy được sử dụng để gấp sản phẩm hoặc làm việc ở nơi môi trường nguy hiểm, khắc nghiệt.

Tay máy có các chức năng cơ bản sau. Xoay, đưa ra, đưa xuống, gấp, đưa lên, đưa vào

Viết chương trình điều khiển cánh tay máy theo các động tác sau :

Khi nhấn nút ON tay máy tiến hành :

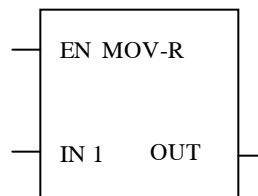
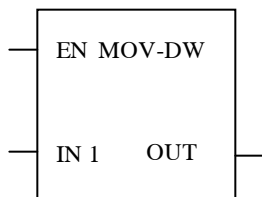
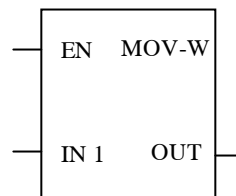
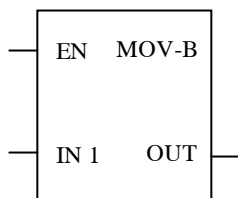




**Ví dụ 24 :** Viết chương trình điều khiển 9 đèn làm việc theo yêu cầu sau :

- a. Sáng nhấp nháy (chu kỳ 1s)
- b. Sáng luân phiên (chu kỳ 1s)
- c. Sáng dần. (chu kỳ 1s)
- d. Tắt dần. (chu kỳ 1s)
- e. Sáng dần từ giữa ra (chu kỳ 1s)

**8. Nhóm lệnh di chuyển dữ liệu :**

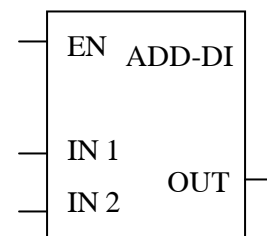
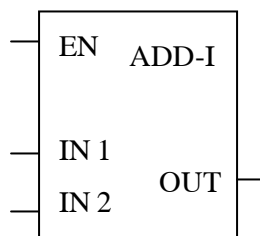
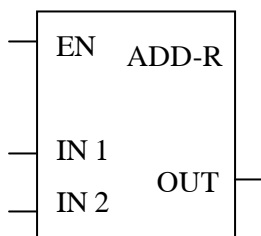


Người ta dùng lệnh Mov để :

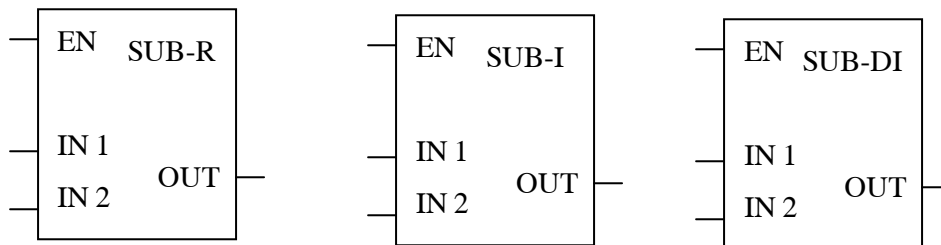
- Nạp giá trị từ ngoài vào bộ nhớ.
- Duyệt chuyển dữ liệu từ vùng nhớ này sang vùng nhớ khác, từ miền nhớ này sang miền nhớ kia.
- Xuất dữ liệu từ miền nhớ ra ngoài.

**9. Nhóm lệnh toán học :**

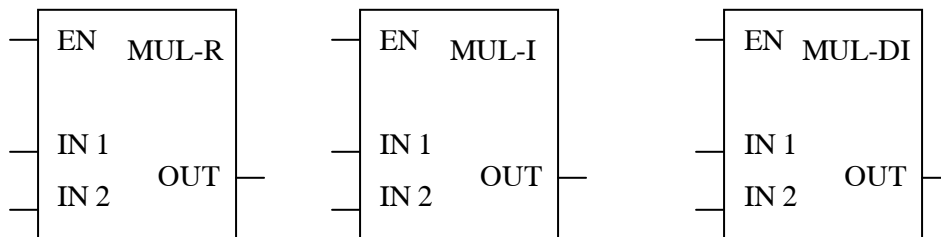
Lệnh ADD (cộng )



Lệnh SUB (trừ )

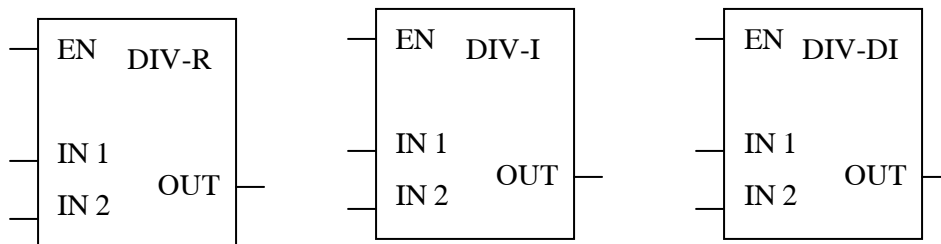


Lệnh MUL (nhân)



Khi nhân 2 số thì dữ liệu sẽ tăng lên 4 byte.

Lệnh DIV chia



Khi chia 2 số phần thương số sẽ được ghi ở 2 byte thấp, dư số được ghi ở 2 byte cao.

- SQRT lệnh thực hiện việc lấy căn bậc 2 của số thực 32 bit.

Lệnh	SM1.0 Kết quả 0	SM1.1 Báo tràn	SM1.2 Kết quả âm	SM1.3 Chia cho 0
ADD-I-D-R-DI	Có	Có 1	Có	Không
SUB-I-D-R-DI	Có	Có 1	Có	Không
MUL-I-D-DI	Có	Không		không
MUL-R	Có	Có 2		Không

DIV-I-D-DI	Có	Có 1		Có
DIV-R	Có	Có 2		Có
SQRT	Có	Có 2	Có	Không
Có 1 : kết quả bị tràn ô nhớ.				
Có 2 : tràn hoặc toán hạn không hợp kiểu.				

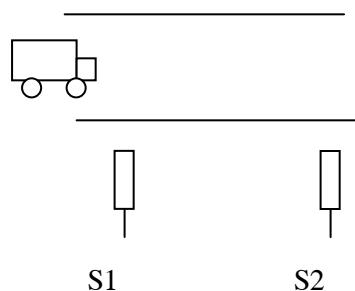
**Ví dụ 25 :** Kiểm soát tốc độ xe.

Dùng 2 cảm biến S1 và S2 để đo khoảng thời gian xe chạy.

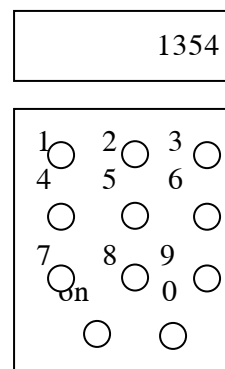
Khoảng cách giữa S1 và S2 là 10m.

Tính vận tốc của xe.

Nếu vận tốc của xe lớn hơn 5m/s thì đèn chớp



Hình 3.6 Kiểm tra tốc độ xe



Hình 3.7 Bộ nhập giá trị và hiển thị

**10. Lệnh tăng giảm một đơn vị và lệnh đảo giá trị thanh ghi.**

Những lệnh này làm đơn giản hoá các vòng điều khiển bên trong chương trình.

**INC-W, INC-B, INC-DW :** lệnh cộng số nguyên 1 vào nội dung một byte, word hoặc 1 Double word.

**DEC - W, INC-B, INC-DW :** Lệnh bớt nội dung từ đơn, byte, DW đi một đơn vị.

**INV- W, INC-B, INC-DW :** Lệnh đảo giá trị từng bit của từ đơn, byte, DW.

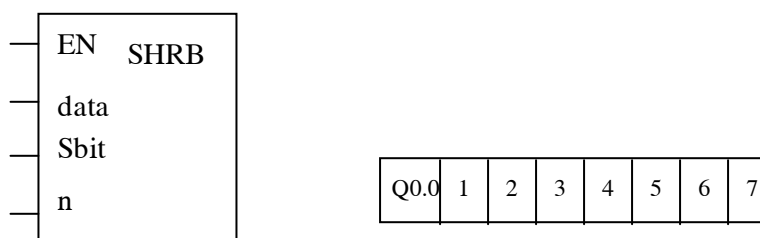
Lệnh	SM1.0 Kết quả 0	SM1.1 Bảo tràn	SM1.2 Kết quả âm	SM1.3 Chia cho 0
INV	Có	Không	Không	Không
DEC	Có	Có	Có	Không
INC	Có	Có	Có	Không

**11. Lệnh chuyển đổi các dạng số :**

- Đổi số dạng byte (B) sang số tự nhiên (I): B-I và ngược lại I-B
- Có các dạng số sau : I = Word, DI = Double Word, R = Real, BCD (AC0,1)
- TRUNC
- ATH
- SEG
- HTA

**12. Lệnh SHRB**

Lệnh dịch chuyển các bit của thanh ghi đi một vị trí trong 1 vòng quét. Thanh ghi được xác định lệnh bằng các toán hạng: **S-BIT** chỉ địa chỉ bit thấp trong thanh ghi và **n** chỉ độ dài thanh ghi. Dấu của toán hạng n xác định chiều dịch chuyển từ bit thấp sang bit cao (n>0) hay ngược lại từ bit cao sang bit thấp (n<0). Toán hạng **Data** là địa chỉ bit mà giá trị logic của nó sẽ được đẩy vào thanh ghi. Giá trị bị đẩy ra khỏi thanh ghi sẽ được ghi vào bit báo tràn SM1.1



- Data : dữ liệu nhập.
- Sbit : địa chỉ bit đầu tiên.
- N số bit.

### 13. Analog Adjustments

CPU 224 có hai bộ điều chỉnh tương tự (Analog Adjustments ) Port 0 và Port 1 giá trị của bộ điều chỉnh tương tự từ 0 đến 255.

- Giá trị của Port 0 được ghi ở ô nhớ SMB28 .
- Giá trị của Port 1 được ghi ở ô nhớ SMB29.

Cần sử dụng một vít dẹp để xoay biến trở, khi đó giá trị của 2 bộ điều chỉnh tương tự sẽ thay đổi theo.

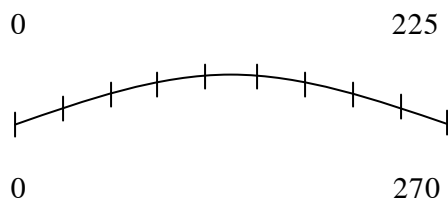
Góc xoay của bộ điều chỉnh tương tự là  $0 \rightarrow 270$  độ.

Để xem được dữ liệu trong SMB28 và SMB29 ta dùng hàm B\_I để chuyển thành số thực.

Ta thấy góc quay và giá trị ngõ ra thay đổi tuyến tính.

Như vậy nếu muốn chia thang không mịn ta dùng hàm so sánh . ví dụ như sau :

- Chia góc xoay biến trở ra thành 10 đoạn.
- Khi biến trở ở thang 1 thì PLC hiển thị là 0s
- Khi biến trở ở thang 2 thì PLC hiển thị là 1s
- Khi biến trở ở thang 3 thì PLC hiển thị là 2 s...



Hình 1.8 Thang chuyển đổi độ quay và giá trị bộ chỉnh tương tự

**Ví dụ 26 :** Lặp lại ví dụ 20 với số quả táo có thể điều chỉnh được bằng cách điều chỉnh biến trở của bộ Analog Adjustments là 5 sản phẩm, 6 sản phẩm ... 10 sản phẩm.

### 14. Chương trình con

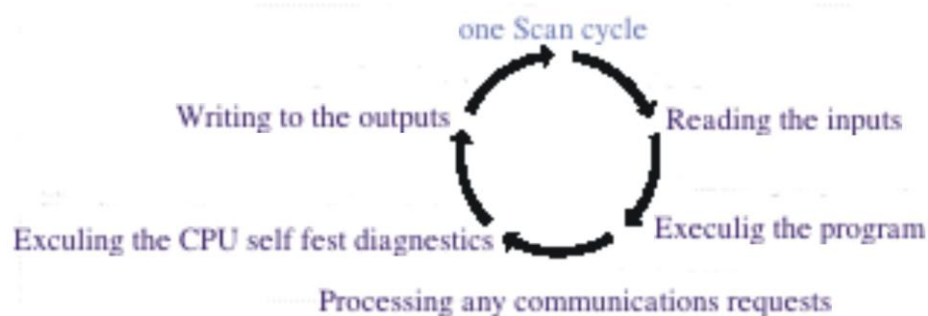
#### a. Cấu trúc chương trình khi sử dụng chương trình con.

Một chương trình bao gồm một chương trình chính và nhiều chương trình con.

Số chương trình con tối đa là 225.

### b. Cách thực hiện một chương trình.

Chương trình được thực hiện theo từng vòng quét Scan. Bắt đầu ở chương trình chính, thực hiện việc quét điều kiện của chương trình con, nếu thỏa điều kiện sẽ thực hiện theo chương trình con đó. Kết thúc chương trình con chương trình quay về chương trình chính thực hiện một vòng quét mới.



### c. Mục đích sử dụng chương trình con :

- Khai thác triệt để nguồn tài nguyên của PLC.
- Dễ lập trình.

### d. Các lưu ý khi sử dụng chương trình con :

- Lệnh gọi chương trình con phải đặt ở chương trình chính.
- Chương trình con chỉ hoạt động khi được chương trình chính gọi.
- Giá trị Logic các Bit, thanh ghi, bộ đếm, bộ định thời nên thực hiện ở chương trình chính.
- Cần lưu ý đến biến toàn cục và biến nội bộ.

### Ví dụ 27 :

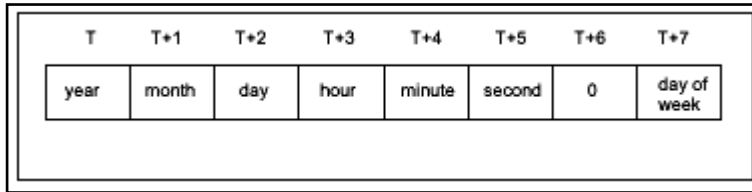
Viết chương trình điều khiển hệ thống đèn quảng cáo có yêu cầu như sau :

- Nhấn nút On các đèn từ Q0.0 đến Q0.7 nhấp nháy 5 lần thì chuyển sang chế độ luân phiên 2 lần thì chuyển lại trạng thái ban đầu.
- Nhấn nút Off thì hệ thống dừng (tất cả các đèn đều không hoạt động).
- Khi nhấn nút On trở lại thì hệ thống hoạt động từ đầu.

**15. Đồng hồ thời gian thực.**

a. Khai báo thời gian ( nạp ngày giờ khai báo vào đồng hồ ):

- Ghi các giá trị ngày tháng vào ô nhớ T.



- Đọc giá trị từ vùng nhớ T vào đồng hồ. Dùng lệnh SET-RTC . Nhớ rằng chỉ được khai báo một lần duy nhất.

**b. Xuất dữ liệu từ đồng hồ ra bộ nhớ.**

Dùng lệnh READ-RTC để xuất dữ liệu từ đồng hồ ra ô nhớ T.

Year/Month	yy mm	yy - 0 to 99	mm - 1 to 12
Day/Hour	dd hh	dd - 1 to 31	hh - 0 to 23
Minute/Second	mm ss	mm - 0 to 59	ss - 0 to 59
Day of week	d	d - 0 to 7	1 = Sunday 0 = disables day of week (remains 0)

c. Chuyển số nhị phân thành số thập phân

**Ví dụ 28 :** Khai báo đồng hồ thời gian thực

**Ví dụ 29 :** Viết chương trình điều khiển hệ thống đèn giao thông có yêu cầu như sau :

Đèn giao thông có 2 tuyến : tuyến 1 và tuyến 2.

Hệ thống hoạt động ở 2 chế độ : Auto và Handy

Ở chế độ Auto :

- Từ 6h00 -19h00 các đèn hoạt động theo trình tự đèn Xanh sáng 4s đèn Vàng sáng 1s Đèn đỏ sáng 5s.
- Từ 19h00 – 6h00 hôm sau chỉ còn 2 đèn vàng nhấp nháy.

Ở chế độ Handy :

- Công tắc S1 đóng thì đèn vàng tuyến 1 và đèn xanh tuyến 2 hoạt động.
- Công tắc S1 mở thì đèn vàng tuyến 2 và đèn xanh tuyến 1 hoạt động.

**Ví dụ 30 :** Viết chương trình điều khiển chuông lớp học.

Hoạt động ở 2 chế độ Auto và handy

Ở chế độ Auto thì :

- 6h40 chuông reng 3 lần mỗi lần 1s nghỉ 1s, lúc 8h30 chuông reng 2 lần mỗi lần 1s nghỉ 1s.

Ở chế độ Handy thì :

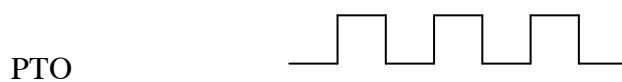
- Nhấn nút On chuông reng không nhấn nút thì chuông dừng.

## 16. Hàm Phát xung tốc độ cao.

### a. Giới thiệu

CPU 224 phát 2 loại xung vuông là : PTO và PWM.

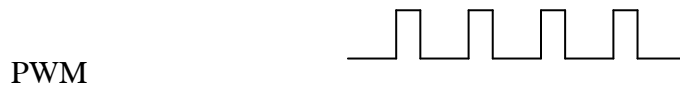
Xung PTO là xung có thời gian ở mức cao bằng thời gian ở mức thấp.



Thời gian T từ :  $250\mu\text{s} \rightarrow 65.535 \mu\text{s}$  hoặc :  $2\text{ms} \rightarrow 65.353 \text{ms}$

Số xung :  $1 \rightarrow 4.294.965.297$  xung

Xung PWM là xung có thời gian ở mức cao khác với thời gian ở mức thấp.



Thời gian T từ :  $250\mu\text{s} \rightarrow 65.535 \mu\text{s}$  hoặc :  $2\text{ms} \rightarrow 65.353 \text{ms}$

Số xung :  $1 \rightarrow 4.294.965.297$  xung

Độ rộng xung :  $0\mu\text{s} \rightarrow 65.535 \mu\text{s}$  hoặc :  $0\text{ms} \rightarrow 65.353 \text{ms}$

### b. Trình tự khai báo :

- Nạp giá trị cho thanh ghi điều khiển.
- Chọn chu kỳ, độ rộng xung, số xung của dãy.
- Khai báo ngắt
- Kích hàm phát xung.



### c. Các địa chỉ cần khai báo :



PLS	THANH GHI ĐIỀU KHIỂN	CHU KỲ	ĐỘ RỘNG XUNG	SỐ XUNG CỦA DÂY
Q0.0	SMB67	SMW68	SMW70	SMD72
Q0.1	SMB77	SMW78	SMW80	SMD82

#### d. Cấu trúc thanh ghi điều khiển

Cấu trúc thanh ghi điều khiển:

SMB67.0	SMB67.0	SMB67.0	SMB67.0	SMB67.0	SMB67.0	SMB67.0	SMB67.0
---------	---------	---------	---------	---------	---------	---------	---------

SMB77.0	SMB77.1	SMB77.2	SMB77.3	SMB77.4	SMB77.5	SMB77.6	SMB77.7
---------	---------	---------	---------	---------	---------	---------	---------

Q0.0	Q0.1	Mục Đích
SM67.0	SM77.0	Đổi chu kỳ. 1- cho phép 0- không cho phép
SM67.1	SM77.1	Đổi độ rộng xung. 1- cho phép 0- không cho phép
SM67.2	SM77.2	Đổi số đếm xung. 1- cho phép 0- không cho phép
SM67.3	SM77.3	Đơn vị thời gian. 1- ms, 0- $\mu$ s
SM67.4	SM77.4	Không sử dụng
SM67.5	SM77.5	Không sử dụng
SM67.6	SM77.6	Chọn kiểu xung. 1- PWM, 0- PTO
SM67.7	SM77.7	Khai báo 0 - huỷ, 1- kích

#### e. Các ngắt khi khai báo phát xung :

- Ngắt PLS0 : ngắt số 19.
- Ngắt PLS1 : ngắt số 20.

**Ví dụ 31 :** Khi PLC chuyển từ STOP sang RUN thì phát xung PTO ra cổng Q0.0, Xung có chu kỳ là 1s, sau khi phát được 10 xung thì dừng sau 10s thì đèn H sáng. Khi nhấn tiếp nút On thì chu kỳ mới lập lại.

**Ví dụ 32 :** Khi nhấn nút On thì phát xung PTO ra cổng Q0.1 chu kỳ 0.5s, số xung là 15s. Đủ số xung thì ngừng phát, sau 10s thì tự động phát lại.

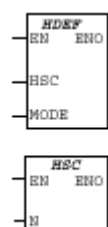
Khi nhấn nút OFF thì vẫn tiếp tục phát. Nhưng sẽ không tự động phát lại cho đến khi nhấn nút ON.

**Ví dụ 33 :** Phát xung PWM ra cổng Q0.0, chu kỳ 1s khi nhấn I0.1 thì chuyển sang phát với chu kỳ 2 s khi nhấn nút OFF thì dừng.

**Ví dụ 44 :** Phát xung PTO với 10 xung đầu chu kỳ 1s, sau đó phát tiếp 20 xung với chu kỳ là 0.5s.

### 17. Khai báo và sử dụng bộ đếm tốc độ cao.

Khi cần đếm xung tần số cao, người ta cần khai báo và lập trình để đếm được tần số cao đó.



#### a. Các địa chỉ cần khai báo :

Bộ đếm	Thanh ghi điều khiển	Giá trị ban đầu	Giá trị đặt trước
HSC0	SMB37	SMD38	SMD42
HSC1	SMB47	SMD48	SMD52
HSC2	SMB57	SMD58	SMD62
HSC3	SMB137	SMD138	SMD142
HSC4	SMB147	SMD148	SMD152
HSC5	SMB157	SMD158	SMD162

#### b. Cấu trúc thanh ghi điều khiển

SMB37	SMB47	SMB57	M Ụ C Đ Ị C H
SM37.0	SM47.0	SM57.0	Không sử dụng (HSC0)
SM37.1	SM47.1	SM57.1	Không sử dụng (HSC0)
SM37.2	SM47.2	SM57.2	Không sử dụng (HSC0)
SM37.3	SM47.3	SM57.3	Chiều đếm. 1 lên, 0 xuống
SM37.4	SM47.4	SM57.4	Đảo chiều đếm. 1 cho phép
SM37.5	SM47.5	SM57.5	Thay đổi giá trị đặt trước. 1 cho phép
SM37.6	SM47.6	SM57.6	Thay đổi giá trị đếm tức thời. 1 cho phép
SM37.7	SM47.7	SM57.7	Cho phép bộ đếm hoạt động. 1 cho phép.

#### c. Đặt điểm các bit không sử dụng đối với HSC0

- SM47.0, SM57.0 : Bit điều khiển tác động chân Reset .

0 reset tác động ở mức cao.

1 reset tác động ở mức thấp.

- SM47.1, SM57.1 : Bit điều khiển mức tác động chân Start.

0 Start ở mức cao.

1 Start ở mức thấp.

- SM47.2, SM57.2 : chọn tốc độ khi đếm lệch pha 90 độ của 2 xung.

0 số đếm được bằng  $4 * \text{số lần lệch pha}$ .

1 số đếm được bằng  $1 * \text{số lần lệch pha}$ .

#### d. Các ngắt liên quan đến bộ đếm.

Event number	Mô tả
0	Cạnh lên I0.0
1	Cạnh xuống I0.0
2	Cạnh lên I0.1
3	Cạnh xuống I0.1
4	Cạnh lên I0.2
5	Cạnh xuống I0.2
6	Cạnh lên I0.3
7	Cạnh xuống I0.3
8	Nhận dữ liệu
9	Xuất dữ liệu
10	Theo thời gian 0, SMB34
11	Theo thời gian 1, SMB35
12	HSC0 _ CV=PV
13	HSC1 _ CV=PV
14	HSC1 thay đổi hướng đếm.
15	HSC1 reset từ bên ngoài.
16	HSC2 CV=PV
17	HSC2 thay đổi hướng đếm.
18	HSC2 reset từ bên ngoài.
19	PLS0 khi phát xung hoàn tất.
20	PLS1 khi phát xung hoàn tất.

#### e. Các chế độ làm việc của bộ đếm HSC1

Event number	Mô tả
0	Đếm tiến hoặc lùi theo I0.6
1	Đếm tiến hoặc lùi theo I0.6 nhận tín hiệu I1.0 làm tín hiệu reset từ bên ngoài.
2	Đếm tiến hoặc lùi theo I0.6, nhận I1.0 làm tín hiệu reset từ bên ngoài, Nhận I1.1 làm tín hiệu khởi phát.
3	Đếm tiến hoặc lùi theo I0.6, chiều đếm quy định bởi I0.7.
4	Đếm tiến hoặc lùi theo I0.6, chiều đếm quy định bởi I0.7. Nhận I1.0 làm tín hiệu reset từ bên ngoài.
5	Đếm tiến hoặc lùi theo I0.6, chiều đếm quy định bởi I0.7. Nhận I1.0 làm tín hiệu reset từ bên ngoài. Nhận I1.1 làm tín hiệu khởi phát.
6	Đếm tiến theo I0.6, lùi theo I0.7.
7	Đếm tiến theo I0.6, Lùi theo I0.7 Nhận I1.0 nhận I1.0 làm tín hiệu Reset từ bên ngoài.
8	Đếm tiến theo I0.6, Lùi theo I0.7 Nhận I1.0 nhận I1.0 làm tín hiệu Reset từ bên ngoài. Nhận I1.1 làm tín hiệu khởi phát.
9	Đếm số lần lệch pha giữa I0.6 và I0.7.
10	Đếm số lần lệch pha giữa I0.6 và I0.7. Nhận I1.0 làm tín hiệu reset từ bên ngoài.
11	Đếm số lần lệch pha giữa I0.6 và I0.7. Nhận I1.0 làm tín hiệu reset từ bên ngoài. Nhận I1.1 làm tín hiệu khởi phát.

Các HSC khác tương tự.

**Ví dụ 35:** Viết chương trình đếm số xung của I0.0. Nếu số xung bằng 20 thì đèn Q0.0 sáng.

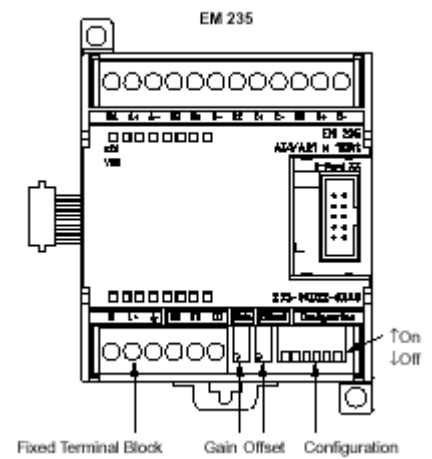
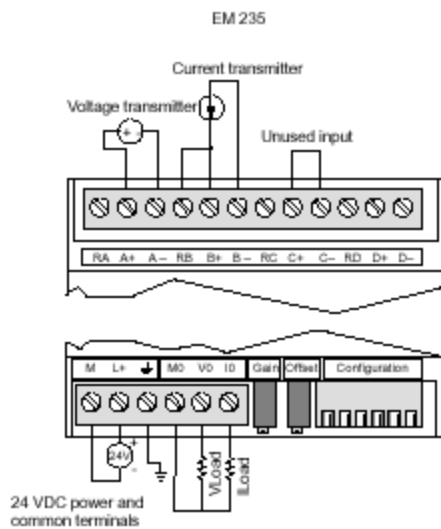
## 18. ANALOG

### a. Các Modul :

EM231 – 4 vào , không có ngõ ra.

EM232 - 0 vào , 2 ra.

EM235 – 4 vào 1 ra.



Đặt điểm EM235.

Modul Analog :

- Địa chỉ cho các ngõ vào :
- AIW0 ngõ vào A.
- AIW2 ngõ vào B.
- AIW4 ngõ vào C.
- AIW6 ngõ vào D.

A : analog, I : input, W : word.

- Các địa chỉ ngõ là Q.
- AQW0, AQW2, ...

Gain : Điều chỉnh độ lợi :

Offset : Điều chỉnh điểm lệch 0.

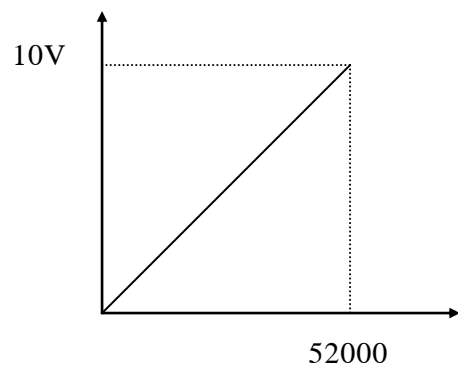
b. Các ví dụ :

**Ví dụ 36 :** Thu nhận tín hiệu analog.

**Ví dụ 37 :** Hoạt động ở 50 độ C

**Ví dụ 38 :** Hoạt động trong khoảng 40 đến 50 độ C.

**Ví dụ 39 :** Hiển thị nội dung của thanh ghi AIW0 ra Led hiển thị số.



## PHỤ LỤC

### CÁC TỪ VIẾT TẮT :

PLC :        **P**rogrammable **L**ogic **C**ontroller

STL:        **S**tatement **L**ist

LAD :       **L**adder **L**ogic

NTC :

PTC :