

TRƯỜNG CAO ĐẲNG NGHỀ ĐÀ LẠT

KHOA ĐIỆN



Giáo trình

ĐIỀU KHIỂN LẬP TRÌNH PLC

Nâng cao

*Dùng cho hệ Cao đẳng
và Trung cấp nghề*



LƯU HÀNH NỘI BỘ



	Nội dung	A
<i>Giáo trình</i>	Giới thiệu chung về PLC S7-300	1
ĐIỀU KHIỂN LẬP TRÌNH PLC Nâng cao	Tập lệnh của PLC S7-300	2
	Phần mềm Step 7/ MicroWin	3
	Bài tập thực hành PLC S7-200; S7-300	5
	Phụ lục	B

Nội dung

A	Nội dung	i
1	Giới thiệu PLC S7 - 300	1
1.1.	Giới thiệu chung về PLC S7_300	2
1.1.1.	Tổng quan	2
1.1.2.	Cấu trúc, chức năng PLC S7-300	3
1.1.3.	Module CPU	3
1.1.4.	Module mở rộng.....	4
1.1.5.	Ngôn ngữ lập trình	8
1.2.	Giới thiệu PLC S7_300 CPU312C	9
1.2.1.	Cấu trúc bộ nhớ	9
1.2.2.	Đơn vị chính CPU 312C	11
1.2.3.	Các ngõ vào ra	13
2	Tập lệnh của PLC S7-300 (dạng LAD)	14
2.1.	Các lệnh logic tiếp điểm	15
2.1.1.	Tập lệnh	15
2.2.	Nhóm lệnh so sánh với số nguyên và số thực	18
2.2.1.	Số nguyên	18
2.2.2.	Số thực.....	20
2.2.3.	Lệnh so sánh số DI.....	22
2.3.	Các lệnh số học	24
2.3.1.	Số nguyên	24
2.3.2.	Số thực.....	27
2.4.	Lệnh đổi kiểu dữ liệu và di chuyển	32
2.4.1.	Các lệnh đổi kiểu dữ liệu	32
2.4.2.	Các lệnh di chuyển	37
2.5.	Timer	41
2.5.1.	Lệnh S_PULSE	41
2.5.2.	Lệnh S_PEXT	42
2.5.3.	Lệnh S_ODT	43
2.5.4.	Lệnh S_OFFDT	43
2.5.5.	Cài đặt Timer.....	44
2.6.	Counter.....	50
2.6.1.	Lệnh đếm lên xuống S_CUD	52
2.6.2.	Lệnh đếm lên S_CU	52
2.6.3.	Lệnh đếm xuống S_CD.....	53
2.6.4.	Set Counter	54
2.7.	Một số lệnh khác và bài tập.....	56
3	Phần mềm Simatic S7 V5.3	58
3.1.	Cài đặt Simatic S7 V5.3	59
3.2.	Cấu hình, tạo mới chương trình điều khiển	63
3.3.	Các vùng nhớ của PLC S7-300	82

3.4.	Kết nối mạng.....	93
4	Bài tập thực hành	103
4.1.	Điều khiển động cơ	
4.1.1.	Điều khiển đảo chiều động cơ 3 pha	104
4.1.2.	Điều khiển khởi động động cơ xoay chiều 3 pha rotor lồng sóc qua điện trở	106
4.1.3.	Điều khiển 2 động cơ xoay chiều 3 pha	109
4.1.4.	Điều khiển khởi động động cơ xoay chiều 3 pha dạng sao tam giác	112
4.1.5.	Điều khiển khởi động động cơ xoay chiều 3 pha dạng sao tam giác có báo lỗi khởi động	114
4.1.6.	Điều khiển động cơ xoay chiều 3 pha theo chu kỳ làm việc	115
4.1.7.	Điều khiển khởi động động cơ KĐB 3 pha qua 4 cấp điện trở.....	118
4.1.8.	Điều khiển 08 động cơ 3 pha chạy tuần tự.....	119
4.1.9.	Điều khiển 03 động cơ 3 pha	120
4.1.10.	Điều khiển động cơ AC 3 pha theo chu kỳ làm việc	121
4.2.	Điều khiển dây chuyền	122
4.2.1.	Điều khiển dây chuyền đóng gói	124
4.2.2.	Điều khiển dây chuyền dây chuyền sản xuất bia.....	127
4.2.3.	Điều khiển dây chuyền sấy	132
4.2.4.	Điều khiển động cơ băng tải	133
4.2.5.	Điều khiển dây chuyền chiết Yaghurt vào hũ	134
4.2.6.	Điều khiển dây chuyền chiết n- ớc vào chai	135
4.3.	Các dạng điều khiển khác	136
4.3.1.	Điều khiển bãi đỗ xe	136
4.3.2.	Điều khiển trộn sơn theo thời gian	137
4.3.3.	Điều khiển trộn sơn theo mức	141
4.3.4.	Điều khiển trộn hóa chất.....	142
4.3.5.	Điều khiển máy bán hàng tự động	143
4.3.6.	Điều khiển đóng mở cửa garage ô tô	144
4.3.7.	Điều khiển bộ đếm sản phẩm	145
4.3.8.	Điều khiển chuông báo tiết học.....	146
4.3.9.	ĐKCT trò chơi dạng “Đường lên đỉnh Olympia”	148
4.3.10.	Điều khiển thao tác máy khoan.....	149
4.3.11.	Điều khiển cơ cấu máy lựa sản phẩm	150
4.3.12.	Điều khiển dẫn n- ớc ra hay đổ n- ớc vào	151
4.3.13.	Điều khiển tín hiệu đèn giao thông.....	152
4.3.14.	Điều khiển báo giờ làm việc tan tầm	153
4.3.15.	Điều khiển cửa cuốn	154
4.3.16.	Ch- ơng trình đếm từ 0 đến 255.....	155
4.3.17.	Điều khiển đèn cầu thang – hành lang.....	156
4.3.18.	Đk kiểm soát độ sáng của bóng đèn tròn 24V/1W.....	157
B	Phụ lục	ii

Giới thiệu PLC S7-300

1

Chương này giới thiệu tổng quan về thiết bị PLC S7-300, đồng thời giới thiệu về thiết bị PLC S7-300 (CPU 312C) đang sử dụng tại xưởng thực hành PLC.

1.1 Giới thiệu chung về PLC S7_300

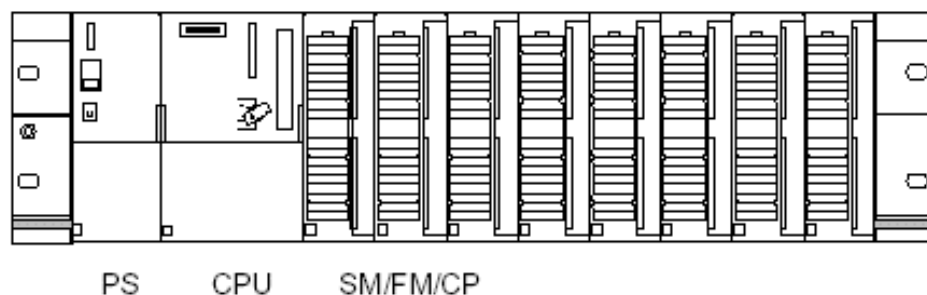
1.2 Giới thiệu PLC S7_300 (CPU312C)

1.1 Giới thiệu chung về PLC S7-300

Tổng quan PLC S7-300 là thiết bị có thể lập trình được của hãng Siemen (Đức) ra đời sau S7-200, có nhiều chức năng và mạnh hơn rất nhiều so với PLC S7-200. PLC S7-300 được dùng trong những ứng dụng lớn, cần nhiều ngõ vào/ra, thời gian đáp ứng nhanh, yêu cầu kết nối mạng và có khả năng mở rộng sau này. PLC S7-300 thuộc dạng đa khối, cũng có cấu trúc dạng module (các module mở rộng về phía bên phải) và gồm các thành phần sau:

- CPU các loại khác nhau: 312IFM, 312C, 313, 313C, 314, 314IFM, 314C, 315, 315-2 DP, 316-2 DP, 318-2,
- Module tín hiệu SM xuất nhập tín hiệu tương tự/số: SM321, SM322, SM323, SM331, SM332, SM334, SM338, SM374
- Module chức năng FM
- Module truyền thông CP
- Module nguồn PS307 cấp nguồn 24VDC cho các module khác, dòng 2A, 5A, 10A
- Module ghép nối IM: IM360, IM361, IM365

Các module được gắn trên thanh ray như hình dưới, tối đa 8 module SM/FM/CP ở bên phải CPU, tạo thành một rack, kết nối với nhau qua bus connector gắn ở mặt sau của module. Mỗi module được gắn một số slot tính từ trái sang phải, module nguồn là slot 1, module CPU slot 2, module kế mang số 4...



Nếu có nhiều module thì bố trí thành nhiều rack (trừ CPU312IFM và CPU313 chỉ có một rack), CPU ở rack 0, slot 2, kế đó là module phát IM360, slot 3, có nhiệm vụ kết nối rack 0 với các rack 1, 2, 3, trên mỗi rack này có module kết nối thu IM361, bên phải mỗi module IM là các module SM/FM/CP. Cấp nối hai module IM dài tối đa 10m. Các module được đánh số theo slot và dùng làm cơ sở để đặt địa chỉ đầu cho các module ngõ vào ra tín hiệu. Đối với CPU

315-2DP, 316-2DP, 318-2 có thể gán địa chỉ tùy ý cho các module.

Cấu trúc, chức năng PLC S7_300

Các khối chức năng :

- Khối tín hiệu (SM:signal module)
 - Khối ngõ vào digital: 24VDC, 120/230VAC
 - Khối ngõ ra digital: 24VDC
 - Khối ngõ vào analog: Áp, dòng, điện trở, thermocouple.

- Khối giao tiếp (IM):

Khối IM360/IM361 dùng để nối nhiều cấu hình. Chúng điều khiển nhiều thanh ghi của hệ thống.

- Khối giả lập (DM): Khối giả lập DM370 dự phòng các khối tín hiệu chưa được chỉ định.
- Khối chức năng (FM): thể hiện những chức năng đặc biệt sau:
 - Đếm
 - Định vị
 - Điều khiển hồi tiếp
- Xử lý liên lạc (CP):
 - Nối điểm-điểm
 - Mạng PROFIBUS
 - Ethernet công nghiệp

Module CPU

Module CPU là loại module có chứa bộ vi xử lý, hệ điều hành, bộ nhớ, các bộ thời gian, bộ đếm, cổng truyền thông (RS485)... và có thể có 1 vài cổng vào ra số. Các cổng vào ra số có trên module CPU được gọi là cổng vào/ra onboard.

PLC S7_300 có nhiều loại module CPU khác nhau. Chúng được đặt tên theo bộ vi xử lý có trong nó như module CPU312, module CPU314, module CPU315...

Những module cùng sử dụng 1 loại bộ vi xử lý, nhưng khác nhau về cổng vào/ra onboard cũng như các khối hàm đặc biệt được tích hợp sẵn trong thư viện của hệ điều hành phục vụ việc sử dụng các cổng vào/ra onboard này sẽ được phân biệt với nhau trong tên gọi bằng thêm cụm chữ IFM(Intergrated Function Module). Ví dụ như Module CPU312 IFM, Module CPU314 IFM...

Ngoài ra còn có các loại module CPU với 2 cổng truyền thông, trong đó cổng truyền thông thứ hai có chức năng chính là phục vụ việc nối mạng phân tán. Các loại module này phân biệt với các loại module khác bằng cụm từ DP (Distributed Port) như là module CPU315-DP.

Module mở rộng *Các module mở rộng được chia thành 5 loại chính:*

- PS (Power supply): Module nguồn nuôi. Có 3 loại: 2A, 5A, 10A.
- SM (Signal module): Module mở rộng cổng tín hiệu vào/ra, bao gồm:
 - DI (Digital input): Module mở rộng các cổng vào số. Số các cổng vào số mở rộng có thể là 8, 16, 32 tùy từng loại module.
 - DO (Digital output): Module mở rộng các cổng ra số. Số các cổng ra số mở rộng có thể là 8, 16, 32 tùy từng loại module.
 - DI/DO (Digital input/ Digital output): Module mở rộng các cổng vào/ra số. Số các cổng vào/ra số mở rộng có thể là 8 vào/8ra hoặc 16 vào/ 16 ra tùy từng loại module.
 - AI (Analog input): Module mở rộng các cổng vào tương tự. Số các cổng vào tương tự có thể là 2, 4, 8 tùy từng loại module.
 - AO (Analog output): Module mở rộng các cổng ra tương tự. Số các cổng ra tương tự có thể là 2, 4 tùy từng loại module.
 - AI/AO (Analog input/ Analog output): Module mở rộng các cổng vào/ra tương tự. Số các cổng vào/ra tương tự có thể là 4 vào/2 ra hay 4 vào/4 ra tùy từng loại module.

Module vào số có các loại sau:

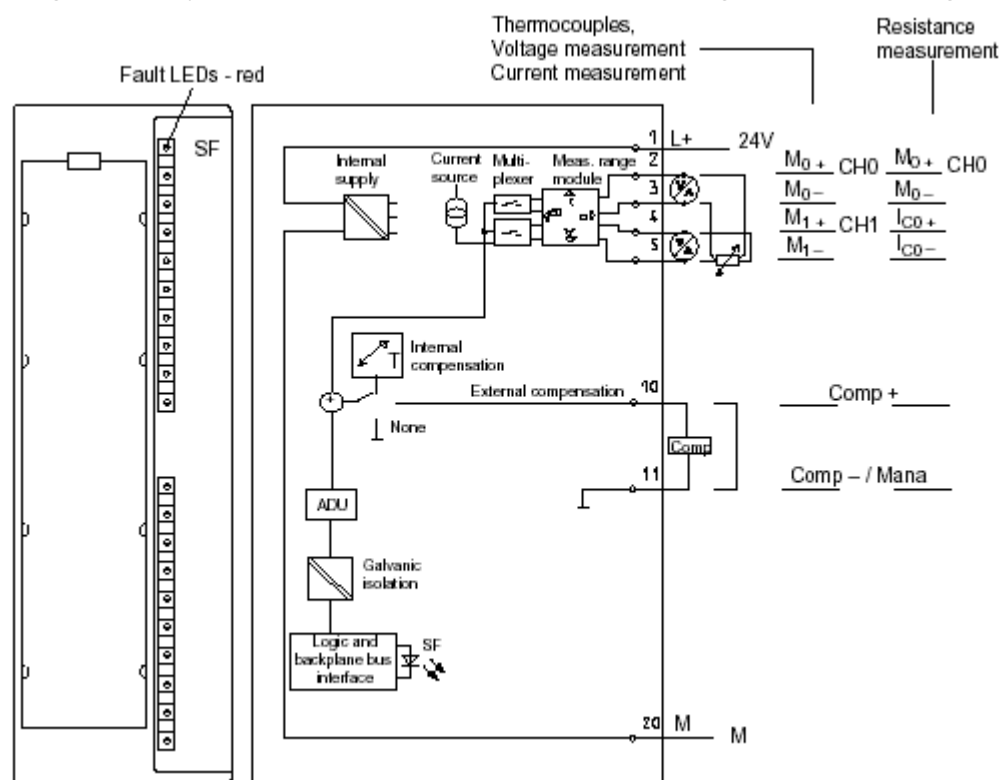
- SM 321; DI 32 _ 24 VDC
- SM 321; DI 16 _ 24 VDC
- SM 321; DI 16 _ 120 VAC, 4*4 nhóm
- SM 321; DI 8 _ 120/230 VAC, 2*4 nhóm
- SM 321; DI 32 _ 120 VAC 8*4 nhóm

Module ra số:

- SM 322; DO 32 _ 24 VDC/0.5 A, 8*4 nhóm
- SM 322; DO 16 _ 24 VDC/0.5 A, 8*2 nhóm
- SM 322; DO 8 _ 24 VDC/2 A, 4*2 nhóm
- SM 322; DO 16 _ 120 VAC/1 A, 8*2 nhóm
- SM 322; DO 8 _ 120/230 VAC/2 A, 4*2 nhóm
- SM 322; DO 32_ 120 VAC/1.0 A, 8*4 nhóm
-
- SM 322; DO 16 _ 120 VAC ReLay, 8*2 nhóm
- SM 322; DO 8 _ 230 VAC Relay, 4*2 nhóm
- SM 322; DO 8 _ 230 VAC/5A Relay, 1*8 nhóm
- Module vào/ ra
- SM 323; DI 16/DO 16 _ 24 VDC/0.5 A
- SM 323; DI 8/DO 8 _ 24 VDC/0.5 A

Module Analog in

Module analog in có nhiều ngõ vào, dùng để đo điện áp, dòng điện, điện trở ba dây, bốn dây, nhiệt độ. Có nhiều tầm đo, độ phân giải, thời gian chuyển đổi khác nhau. Cài đặt thông số hoạt động cho



module bằng phần mềm S7- Simatic 300 Station – Hardware và/hoặc chương trình người dùng sử dụng hàm SFC 55, 56, 57 phù hợp (xem mục) và/hoặc cài đặt nhờ module tầm đo (measuring range module) gắn trên module SM. Kết quả chuyển đổi là số nhị phân phụ hai với bit MSB là bit dấu.

- *SM331 AI 2*12* : module chuyển đổi hai kênh vi sai áp hoặc dòng, hoặc một kênh điện trở 2/3/4 dây, dùng phương pháp tích phân, thời gian chuyển đổi từ 5ms đến 100ms, độ phân giải 9, 12, 14 bit + dấu, các tầm đo như sau: ± 80 mV; ± 250 mV; ± 500 mV; ± 1000 mV; ± 2.5 V; ± 5 V; $1 \dots 5$ V; ± 10 V; ± 3.2 mA; ± 10 mA; ± 20 mA; $0 \dots 20$ mA; $4 \dots 20$ mA. Điện trở 150 Ω ; 300 Ω ; 600 Ω ; Đo nhiệt độ dùng cặp nhiệt E, N, J, K, L, nhiệt kế điện trở Pt 100, Ni 100. Các thông số mặc định đã được cài sẵn trên module, kết hợp với đặt vị trí của module tầm đo (bốn vị trí A, B, C, D) nếu không cần thay đổi thì có thể sử dụng ngay.

- *SM331, AI 8*12 bit* , 8 kênh vi sai chia làm hai nhóm, độ phân giải 9 (12, 14) bit + dấu

- *SM331, AI 8*16 bit* , 8 kênh vi sai chia làm 2 nhóm , độ phân giải 15 bit + dấu

Module Analog Out:

Cung cấp áp hay dòng phụ thuộc số nhị phân phụ hai

- *SM332 AO 4*12 bit*: 4 ngõ ra dòng hay áp độ phân giải 12 bit, thời gian chuyển đổi 0.8 ms .

- *SM332 AO 2*12 bit*

- *SM332 AO 4*16 bit*

Module Analog In/Out

- *SM 334; AI 4/AO 2 * 8 Bit*

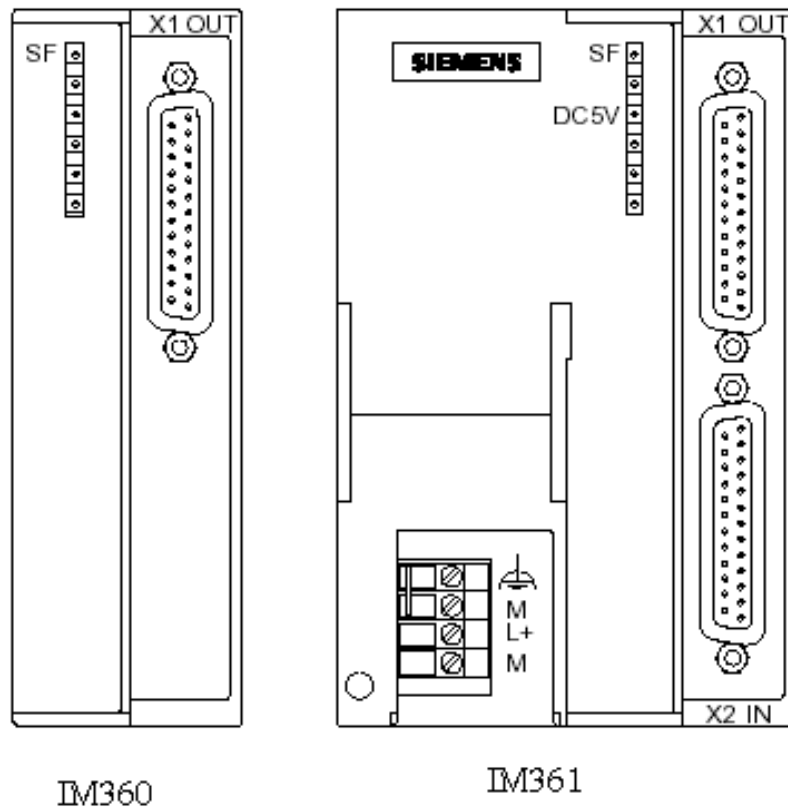
- *SM334; AI 4/AO 2* 12 Bit*

- IM (Interface module): Modul ghép nối.

Đây là loại Modul chuyên dụng có nhiệm vụ nối từng nhóm các Modul mở rộng lại với nhau thành từng một khối và được quản lí chung bởi một module CPU. Thông thường các Modul mở rộng được gá liền với nhau trên một thanh đỡ gọi là Rack.

Trên mỗi một Rack chỉ có thể gá được nhiều nhất 8 module mở rộng (không kể module CPU, module nguồn nuôi). Một module CPU có

thể làm việc trực tiếp với nhiều nhất 4 Rack, và các Rack này phải được nối với nhau bằng Module IM (IM360 :truyền; IM361:nhận).



Module IM360 gắn ở rack 0 kế CPU dùng để ghép nối với module IM361 đặt ở các rack 1, 2, 3 giúp kết nối các module mở rộng với CPU khi số module lớn hơn 1. Cáp nối giữa hai rack là loại 368.

Trong trường hợp chỉ có hai rack, ta dùng loại IM365.

- FM (Function module): Module có chức năng điều khiển riêng. Ví dụ như module PID, module điều khiển động cơ bước...
 - FM350-1* : đếm xung một kênh
 - FM350-2* : đếm xung tám kênh
 - FM351, 353, 354, 357-2* : điều khiển định vị
 - FM352*: bộ điều khiển cam điện tử
 - FM355*: bộ điều khiển hệ kín
- CP (Communication module): Module phục vụ truyền thông trong mạng giữa các PLC với nhau hoặc giữa PLC với máy tính.

Ngôn ngữ lập trình *Tương tự như PLC S7-200 thì PLC S7-300 cũng có 3 ngôn ngữ lập trình cơ bản sau:*

- Ngôn ngữ “liệt kê lệnh”, ký hiệu là STL (Statement List). Đây là dạng ngôn ngữ lập trình thông thường của máy tính. Một chương trình được ghép bởi nhiều câu lệnh theo 1 thuật toán nhất định, mỗi lệnh chiếm 1 hàng và đều có cấu trúc chung là “tên lệnh”+”toán hạng”.

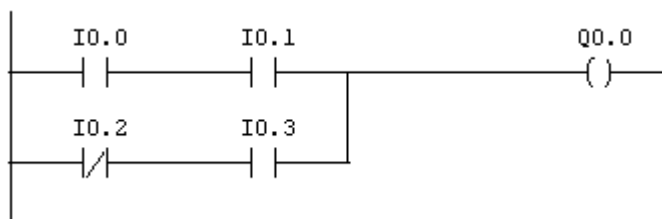
```

A      I      0.0
A      I      0.1
O
AN     I      0.2
A      I      0.3
=      Q      0.0
    
```

Ví dụ:

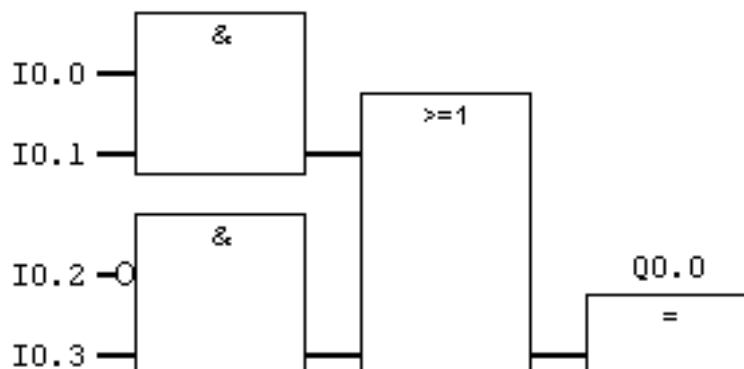
- Ngôn ngữ “hình thang”, ký hiệu là LAD (Ladder Logic). Đây là dạng ngôn ngữ đồ họa thích hợp với những người quen thiết kế mạch điều khiển logic.

Ví dụ:



- Ngôn ngữ “hình khối”, ký hiệu là FBD (Function Block Diagram). Đây là dạng ngôn ngữ đồ họa thích hợp với những người quen thiết kế mạch điều khiển số.

Ví dụ:



1.2 Giới thiệu PLC S7_300 CPU312C

Cấu
trúc bộ
nhớ



Bộ nhớ làm việc 16KB, chu kỳ lệnh 0.1us, tích hợp sẵn 10DI/6DO, 2 xung tốc độ cao 2.5KHz, 2 kênh đọc xung tốc độ cao 10Khz.

Vùng chứa chương trình ứng dụng:

- OB (Organisation block): Miền chứa chương trình tổ chức, trong đó:

Khối OB1: Khối tổ chức chính, mặc định, thực thi lặp vòng. Nó được bắt đầu khi quá trình khởi động hoàn thành và bắt đầu trở lại khi nó kết thúc.

Khối OB10 (Time of day interrupt): được thực hiện khi có tín hiệu ngắt thời gian.

Khối OB20 (Time delay interrupt): được thực hiện sau 1 khoảng thời gian đặt trước.

Khối OB35 (Cyclic Interrupt): khối ngắt theo chu kỳ định trước

Khối OB40 (Hardware Interrupt): được thực hiện khi tín hiệu ngắt cứng xuất hiện ở ngõ vào I124.0...I124.3

- FC (Function): Miền chứa chương trình con được tổ chức thành hàm có biến hình thức để trao đổi dữ liệu với chương trình đã gọi nó, được phân biệt bởi các số nguyên. Ví dụ: FC1, FC7, FC30...ngoài ra còn có các hàm SFC là các hàm đã được tích hợp sẵn trong hệ điều

hành.

- FB (Function Block): tương tự như FC, FB còn phải xây dựng 1 khối dữ liệu riêng gọi là DB (Data Block) và cũng có các hàm SFB là các hàm tích hợp sẵn trong hệ điều hành.

Vùng chứa các tham số hệ điều hành và chương trình ứng dụng:

- I (Process image input): Miền bộ đệm dữ liệu các ngõ vào số. Trước khi bắt đầu thực hiện chương trình, PLC sẽ đọc tất cả giá trị logic của các cổng vào rồi cất giữ chúng trong vùng I. khi thực hiện chương trình CPU sẽ sử dụng các giá trị trong vùng I mà không đọc trực tiếp từ ngõ vào số.
- Q (Process image output): tương tự vùng I, miền Q là bộ đệm dữ liệu cổng ra số. Khi kết thúc chương trình, PLC sẽ chuyển giá trị logic của bộ đệm Q tới các cổng ra số.
- M (Memory): Miền các biến cờ. Do vùng nhớ này không mất sau mỗi chu kì quét nên chương trình ứng dụng sẽ sử dụng vùng nhớ này để lưu giữ các tham số cần thiết. Có thể truy nhập nó theo bit (M), byte (MB), theo từ (MW) hay từ kép (MD).
- T (Timer): Miền nhớ phục vụ bộ thời gian bao gồm việc lưu trữ các giá trị đặt trước (PV-Preset Value), các giá trị tức thời (CV-Current Value) cũng như các giá trị logic đầu ra của Timer.
- C (Counter): Miền nhớ phục vụ bộ đếm bao gồm việc lưu giữ các giá trị đặt trước (PV-Preset Value), các giá trị tức thời (CV-Current Value) cũng như các giá trị logic đầu ra của Counter.
- PI: Miền địa chỉ cổng vào của các module tương tự (I/O External input). Các giá trị tương tự tại cổng vào của module tương tự sẽ được module đọc và chuyển tự động theo những địa chỉ. Chương trình ứng dụng có thể truy cập miền nhớ PI theo từng byte (PIB), từng từ (PIW) hoặc theo từng từ kép (PID).
- PQ: Miền địa chỉ cổng ra của các module tương tự (I/O External output). Các giá trị tương tự tại cổng vào của module tương tự sẽ được module đọc và chuyển tự động theo những địa chỉ. Chương trình ứng dụng có thể truy cập miền nhớ PI theo từng byte (PQB), từng từ (PQW) hoặc theo từng từ kép (PQD).

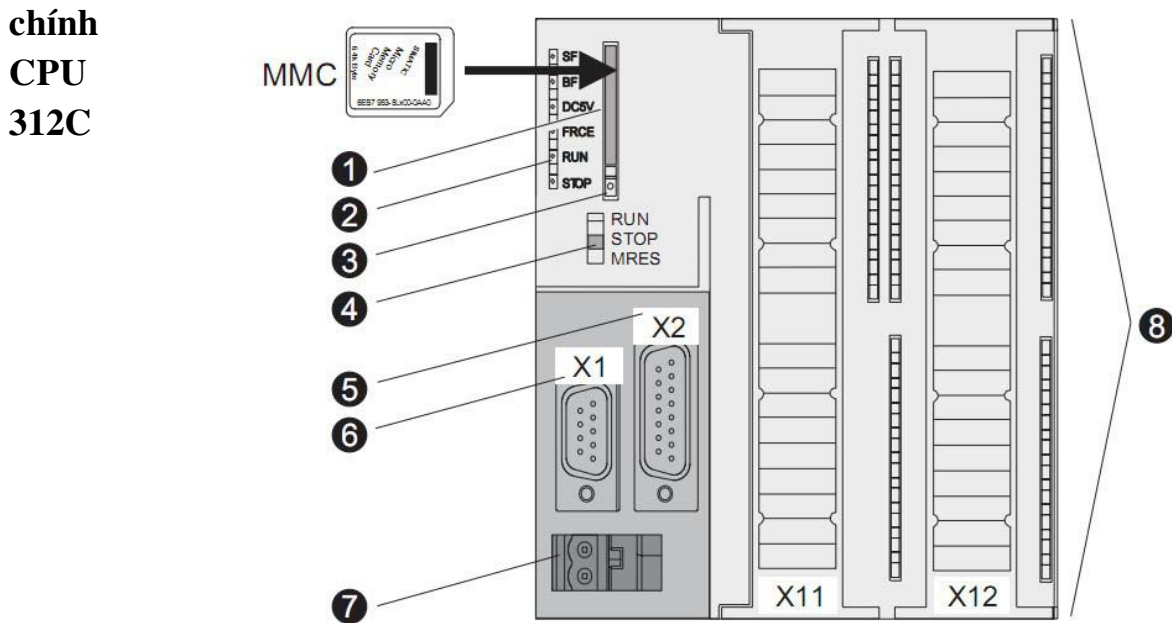
Vùng chứa các khối dữ liệu, được chia thành 2 loại:

- DB (Data block): Miền chứa các dữ liệu được tổ chức thành

khối. Kích thước hay số lượng khối do người sử dụng qui định. Có thể truy nhập miền này theo từng bit (DBX), byte(DBB), từng từ (DBW), từ kép (DBD).

- L (Local data block): Miền dữ liệu địa phương, được các khối chương trình OB, FC, FB tổ chức và sử dụng cho các biến nhấp tức thời và trao đổi dữ liệu của biến hình thức với những khối đã gọi nó. Toàn bộ vùng nhớ sẽ bị xoá sau khi khối thực hiện xong. Có thể truy nhập theo từng bit (L), byte (LB), từ (LW), hoặc từ kép (LD).

Đơn vị chính CPU 312C CPU 312C như hình dưới:



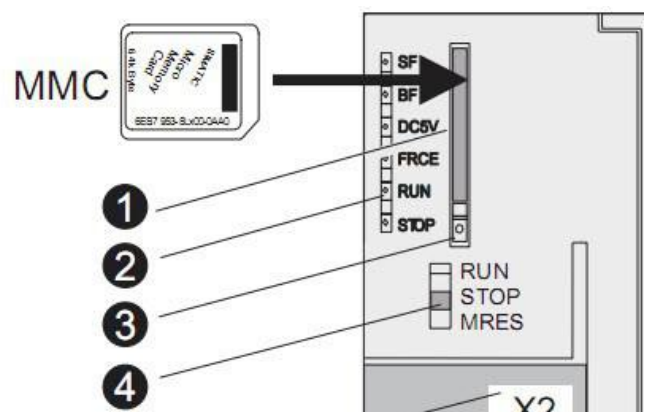
1. Chỗ cắm thẻ nhớ.
2. Đèn báo trạng thái và báo lỗi.
3. Chốt tháo thẻ nhớ.
4. Công tắc chọn trạng thái.
5. Cổng Truyền thông 2X2 Profibus PtP hoặc DP
6. Cổng Truyền thông MPI

Đèn:

- SF (màu đỏ): lỗi phần cứng hay mềm, lỗi CPU (Lỗi nhóm, chương trình sai hay lỗi từ khối chuẩn đoán);
- BATF (màu đỏ): báo nguồn nuôi bị lỗi (lỗi pin hay không có pin);
- DC5V (màu xanh lá cây): báo nguồn 5V bình thường;
- FRCE (màu vàng): force request tích cực (sáng lên khi biến cưỡng bức tác động);
- RUN (màu xanh lá cây) : CPU ở chế độ đang làm việc với chương trình đã được nạp vào CPU (mode RUN), LED chớp lúc PLC khởi động và dừng khi đã ổn định;
- STOP (màu vàng): CPU đang ở chế độ dừng (có thể sửa chữa, upload hay download chương trình),
Ổn định ở chế độ STOP,
Chớp chậm khi có yêu cầu RESET bộ nhớ,
Chớp nhanh khi đang RESET bộ nhớ;
- BUSF (màu đỏ): lỗi phần cứng hay phần mềm ở giao diện PROFIBUS

Công tắc:

- RUN: chế độ chạy chương trình
- STOP: ngừng chạy chương trình
- MRES: reset bộ nhớ (reset khối)



**Các ngõ CPU 312C như hình dưới:
vào ra**



10 ngõ vào số được định địa chỉ từ I0.0 đến I1.1 trong đó:

6 ngõ ra số từ Q0.0 đến Q0.5 có mức điện áp là 24VDC và dòng tối đa là 0.5A.

Tập lệnh của PLC S7-300 (dạng LAD)

2

Chương này giới thiệu tập lệnh của thiết bị PLC S7-300 ở dạng ngôn ngữ LAD.

2.1 Các lệnh logic tiếp điểm

2.2 Nhóm lệnh so sánh với số nguyên và số thực

2.3 Các lệnh toán học

2.4 Lệnh đổi kiểu dữ liệu và di chuyển

2.5 Bộ thời gian (Timer)

2.6 Bộ đếm (Counter)

2.7 Một số lệnh khác + Bài tập

2.1 Các lệnh logic tiếp điểm

Tập lệnh *Thanh Ghi Trạng Thái:*

Khi thực hiện lệnh ,CPU sẽ ghi nhận lại trạng thái của phép tính trung gian cũng như của kết quả vào một thanh ghi đặc biệt 16 Bits,được gọi là thanh ghi trạng thái (Status Word) >Mặc dù thanh ghi trạng thái này có độ dài 16 Bits nhưng chỉ sử dụng 9 Bits với cấu trúc như sau:

BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
----	-----	-----	----	----	----	-----	-----	----

FC (First check) : Khi phải thực hiện một dãy các lệnh logic liên tiếp nhau gồm các phép tính giao ,hợp và nghịch đảo,bit FC có giá trị bằng 1,hay nói cách khác ,FC=0 khi dãy lệnh Logic tiếp điểm vừa được kết thúc.

RLO (Result of logic operation) : Kết quả tức thời của phép tính logic vừa được thực hiện.

STA (Status bit) : Bit trạng thái này luôn có giá trị logic của tiếp điểm được chỉ định trong lệnh.

OR :Ghi lại giá trị của phép tính logic giao cuối cùng được thực hiện để phụ giúp cho việc thực hiện phép toán hợp sau đó.Điều này là cần thiết vì trong một biểu thức hàm 2 trị ,phép tính giao bao giờ cũng phải được thực hiện trước các phép tính hợp.

OS (Stored overflow bit) : Ghi lại giá trị Bit bị tràn ra ngoài mảng ô nhớ.

OV(Overflow Bit): Bit báo cáo kết quả phép tính bị tràn ra ngoài mảng ô nhớ.

CC0 và CC1 (Condition code) : Hai bit báo trạng thái của kết quả phép tính với số nguyên,số thực phép dịch chuyển hoặc phép tính logic trong ACCU.

CC1	CC0	Ý nghĩa
0	0	Kết quả bằng 0 (=0)
0	1	Kết quả nhỏ hơn 0 (<0)
1	0	Kết quả lớn hơn 0 (>0)

BR (Binary result bit) : Bit trạng thái cho phép liên kết hai loại ngôn ngữ lập trình STL và LAD .Chẳng hạn cho phép người sử dụng

có thể viết một khối chương trình FB hoặc FC trên ngôn ngữ STL nhưng gọi và sử dụng chúng trong một chương trình khác viết trên LAD. Để tạo ra được mối liên kết đó, ta cần phải kết thúc chương trình trong FB, FC bằng lệnh ghi.

BR = 1, nếu chương trình chạy không có lỗi

BR = 0, nếu chương trình chạy có lỗi

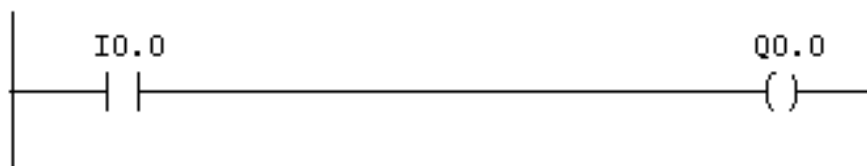
Khi sử dụng các khối hàm đặc biệt của hệ thống (SFC hoặc SFB) ,trạng thái làm việc của chương trình cũng được thông báo ra ngoài qua bit trạng thái BR như sau:

BR=1 nếu SFC hay SFB thực hiện không có lỗi

BR=0 nếu có lỗi khi thực hiện SFC hay SFB

Lệnh GÁN

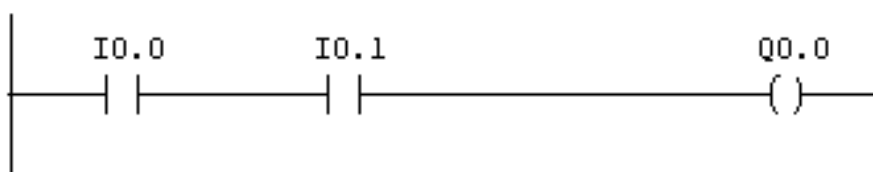
Ví dụ:



Khi ngõ vào I0.0 lên mức 1 thì ngõ ra Q0.0 ON

Lệnh AND

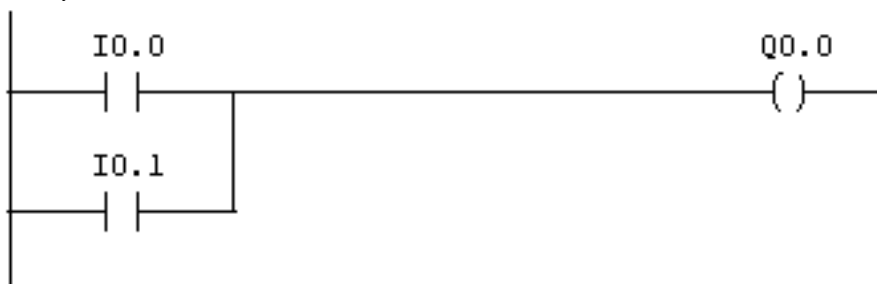
Ví dụ:



Khi I0.0 và I0.1 đồng thời lên mức 1 thì ngõ ra Q0.0 ON

Lệnh OR

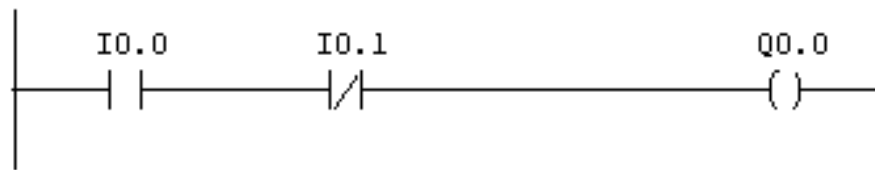
Ví dụ:



Khi 1 trong 2 ngõ vào I0.0 hoặc I0.1 lên mức 1 thì ngõ ra Q0.0 ON

Lệnh AND NOT

Ví dụ:



Khi I0.0 lên mức 1 và I0.1 ở mức 0 thì Q0.0 ON

Lệnh OR NOT

Ví dụ:

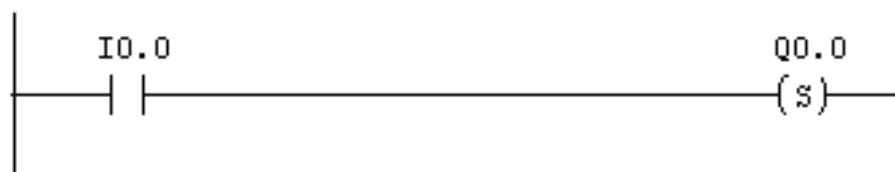


Khi I0.0 mức 1 hay I0.1 mức 0 thì Q0.0 ON

Lệnh GÁN có điều kiện

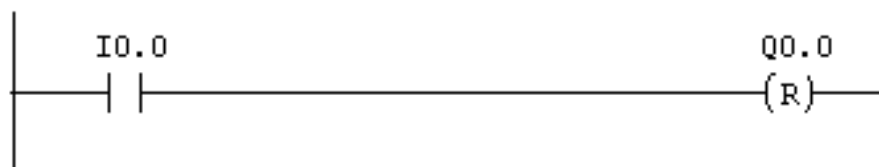
- Lệnh gán giá trị 1

Ví dụ:



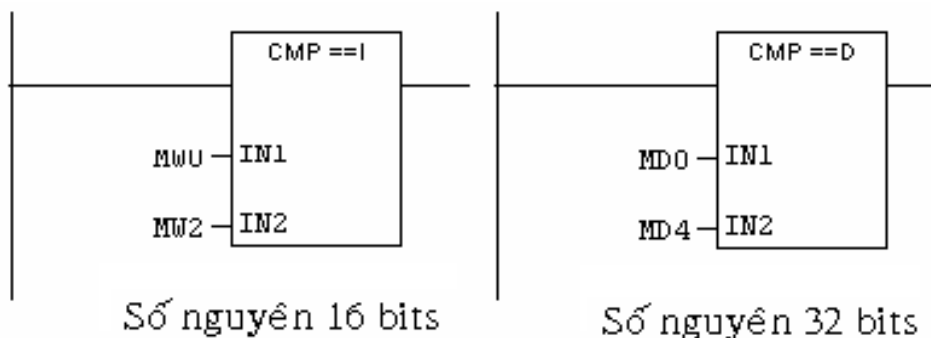
- Lệnh gán giá trị 0

Ví dụ:



2.2 Nhóm lệnh so sánh với số nguyên và số thực

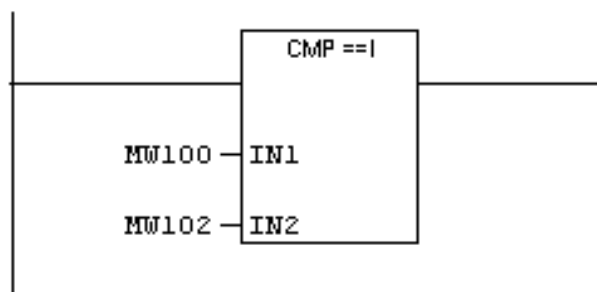
Với số nguyên So sánh bằng
Ví dụ:



(KQ là kết quả thu được sau phép tính

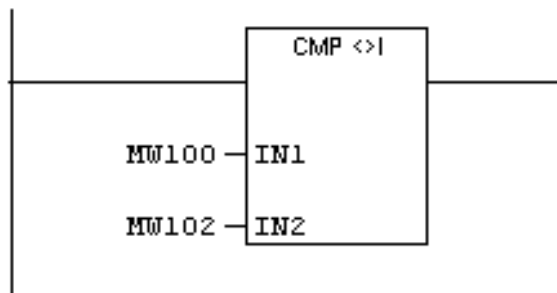
KT là kết quả trước phép tính)

Lệnh *EQ_I* (Equal Integer): So sánh MW100 và MW102, nếu 2 số nguyên này bằng nhau thì KQ=KT.



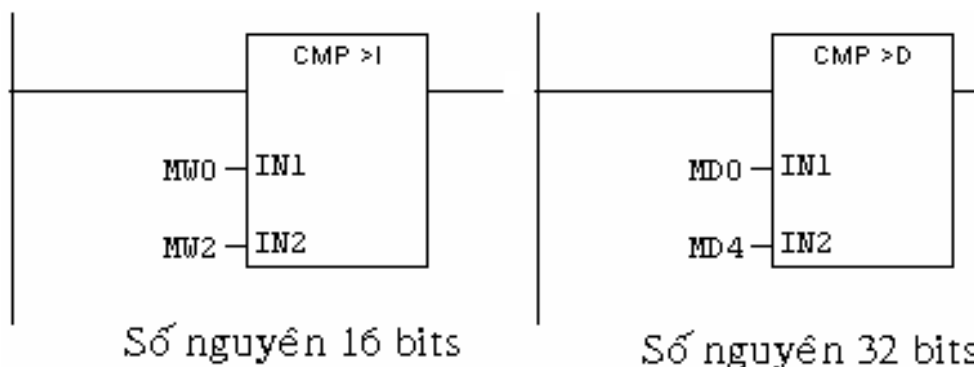
So sánh khác

Lệnh *NE_I* (Not Equal Integer) : So sánh MW100 và MW102, nếu 2 số này khác nhau thì KQ=KT.

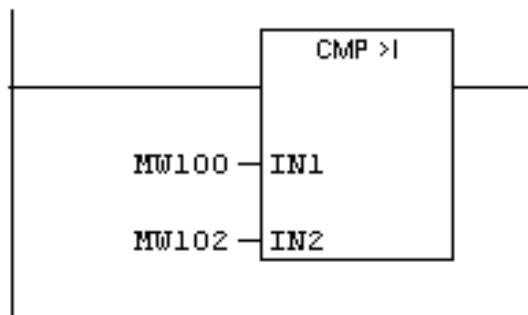


So sánh lớn hơn

Ví dụ:



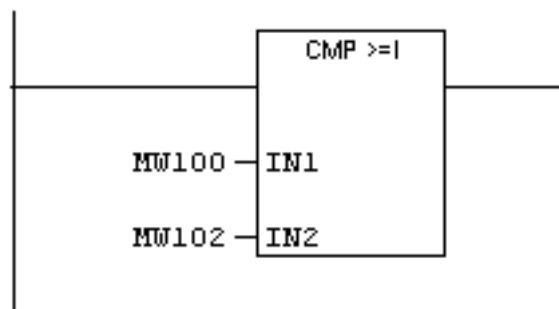
Lệnh *GT_I* (Greater than Integer) : So sánh 2 số MW100 và MW102 nếu MW100 lớn hơn MW102 thì KQ=KT.



So sánh lớn hơn hoặc bằng

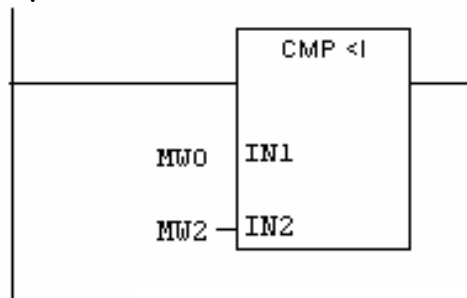
Ví dụ:

Lệnh *GE_I* (Greater than or equal Integer) : So sánh 2 số MW100 và MW102, Nếu MW100 lớn hơn hoặc bằng MW102 thì KQ=KT.

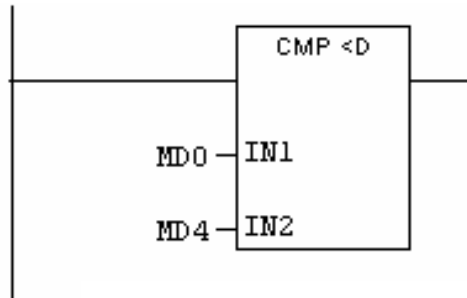


So sánh bé hơn

Ví dụ:



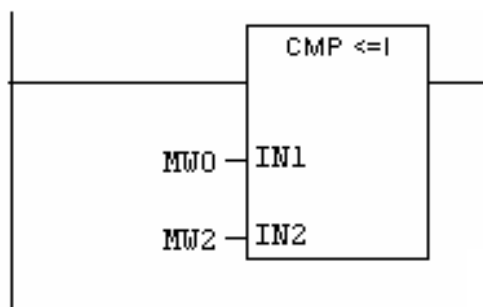
Số nguyên 16 bits



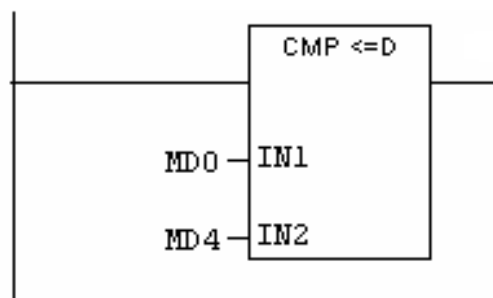
Số nguyên 32 bits

So sánh bé hơn hoặc bằng

Ví dụ:

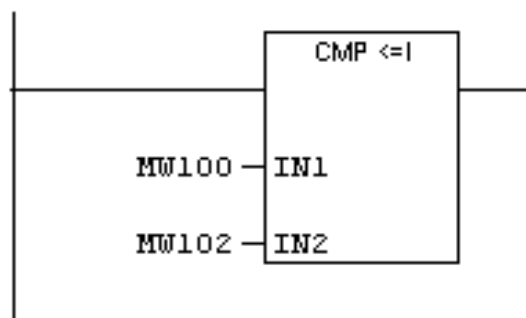


Số nguyên 16 bits



Số nguyên 32 bits

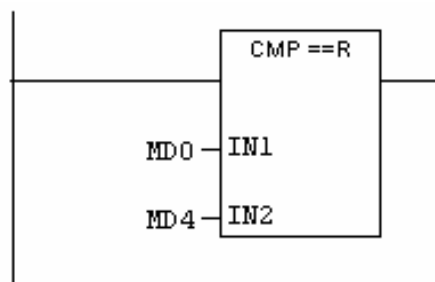
Lệnh *LE_I* (Less than or equal Integer) : So sánh 2 số MW100 và MW102, Nếu MW100 bé hơn hoặc bằng MW102 thì KQ=KT.



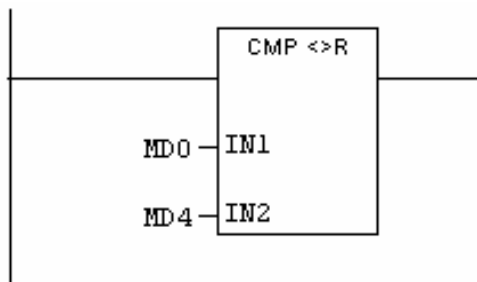
Với số thực

So sánh bằng

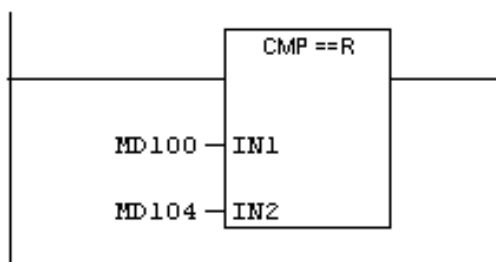
Ví dụ:



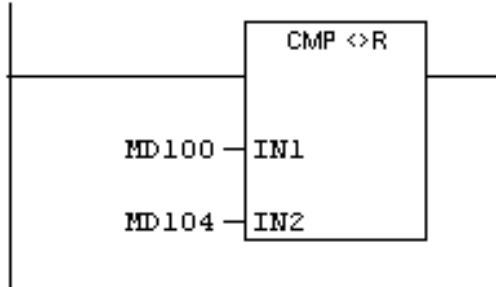
So sánh khác



Lệnh *EQ_R* (Equal Real): So sánh MD100 và MD104, nếu 2 số nguyên này bằng nhau thì KQ=KT.



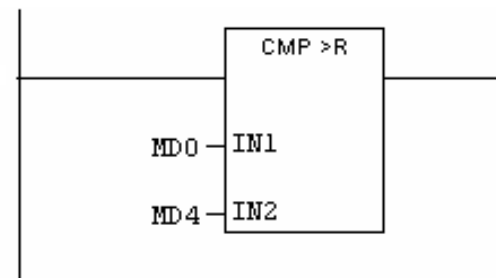
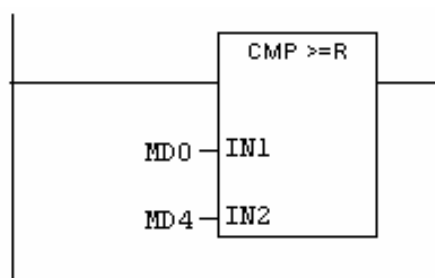
Lệnh *NE_R* (Not Equal Real) : So sánh MD100 và MD104, nếu 2 số này khác nhau thì KQ=KT.



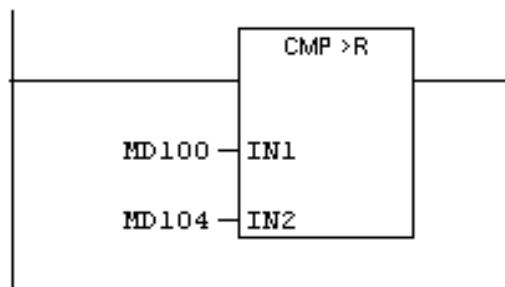
So sánh lớn hơn

So sánh lớn hơn hoặc bằng

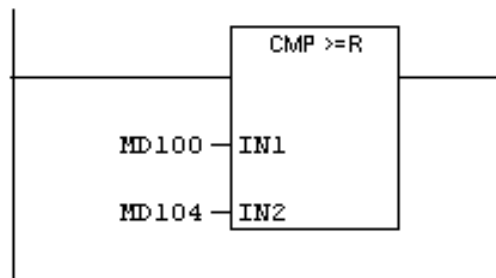
Ví dụ:



Lệnh GT_R (Greater than Real)
: So sánh 2 số MD100 và MD104, nếu MD100 lớn hơn MD104 thì KQ=KT.

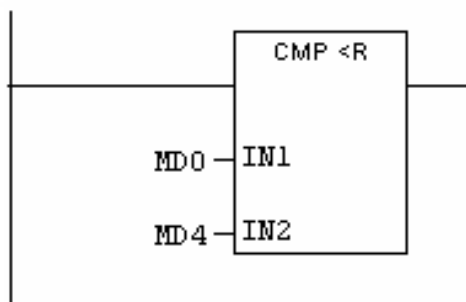


Lệnh GE_R (Greater than or equal Real)
: So sánh 2 số MD100 và MD104, Nếu MD100 lớn hơn hoặc bằng MD104 thì KQ=KT.

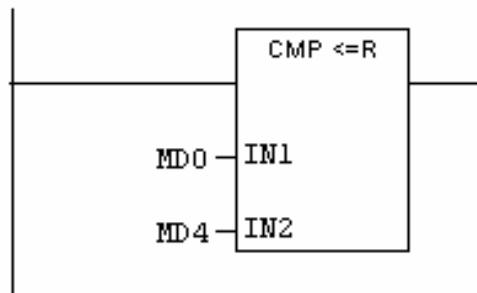


So sánh bé hơn

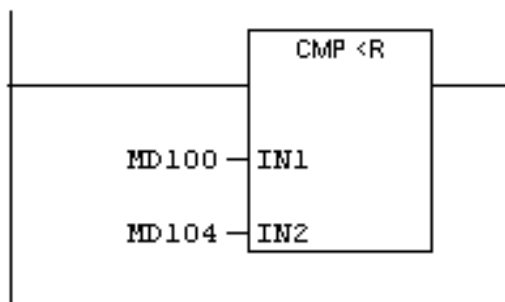
Ví dụ:



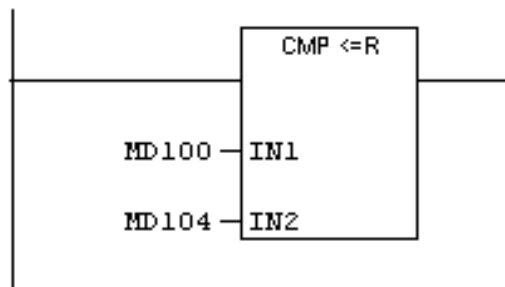
So sánh bé hơn hoặc bằng



Lệnh LT_R (Less than Real)
: So sánh 2 số MD100 và MD104, Nếu MD100 bé hơn MD104 thì KQ=KT.

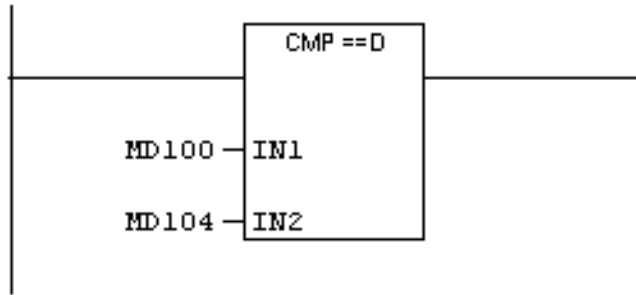


Lệnh LE_R (Less than or equal Real)
: So sánh 2 số MD100 và MD104, Nếu MD100 bé hơn hoặc bằng MD104 thì KQ=KT.

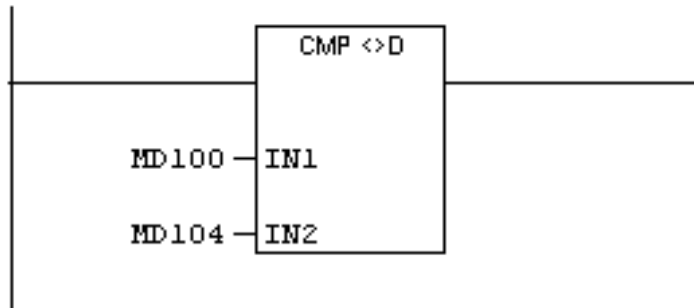


Lệnh so sánh số Double Integer *Lệnh EQ_D* (Equal Double Integer): So sánh MD100 và MD104, nếu 2 số nguyên này bằng nhau thì KQ=KT.

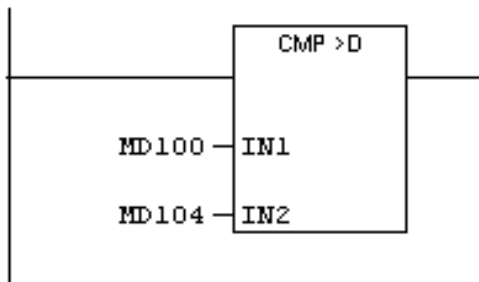
Double Integer



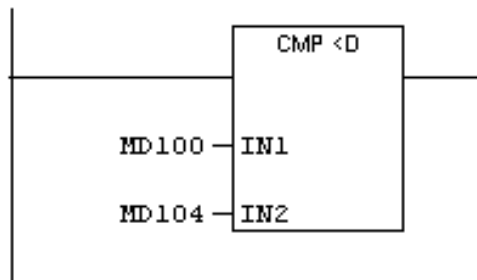
Lệnh NE_D (Not Equal Double Integer) : So sánh MD100 và MD104 ,nếu 2 số này khác nhau thì KQ=KT.



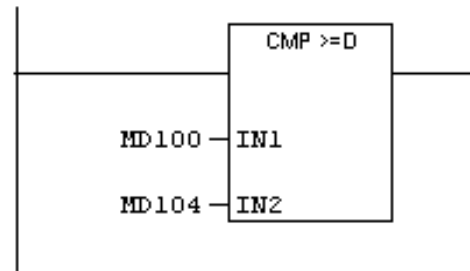
Lệnh GT_D (Greater than DoubleInteger) : So sánh 2 số MD100 và MD104, nếu MD100 lớn hơn MD104 thì KQ=KT.



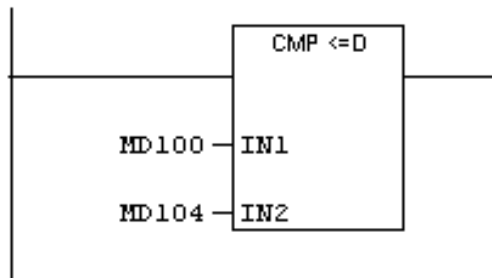
Lệnh LT_D (Less than DoubleInteger) : So sánh 2 số MD100 và MD104, Nếu MD100 bé hơn MD104 thì KQ=KT.



Lệnh *GE_D* (Greater than or equal DoubleInteger) : So sánh 2 số MD100 và MD104, Nếu MD100 lớn hơn hoặc bằng MD104 thì KQ=KT.

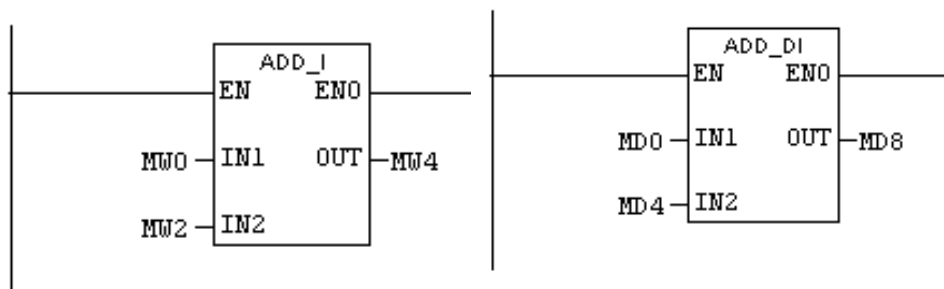


Lệnh *LE_D* (Less than or equal DoubleInteger) : So sánh 2 số MD100 và MD104, Nếu MD100 bé hơn hoặc bằng MD104 thì KQ=KT.



2.3 Các lệnh toán học

Với số nguyên *Lệnh cộng số nguyên*
 nguyên Ví dụ:

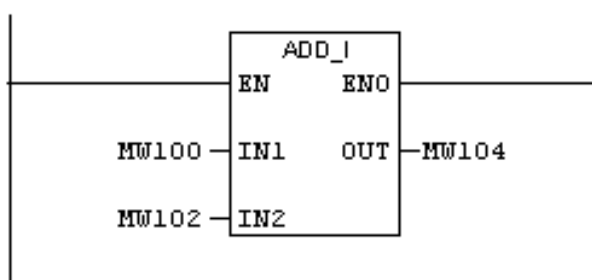


Số nguyên 16 bits

Số nguyên 32 bits

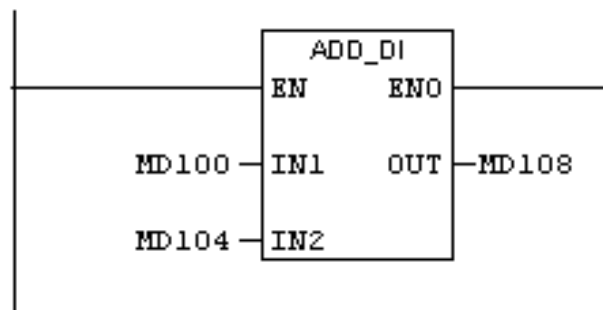
Lệnh ADD_I : Lệnh thực hiện việc cộng 2 số nguyên 16 Bit, kết quả cất vào số nguyên 16 Bit, nếu kết quả vượt quá 16 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MW104 = MW100 + MW102$$



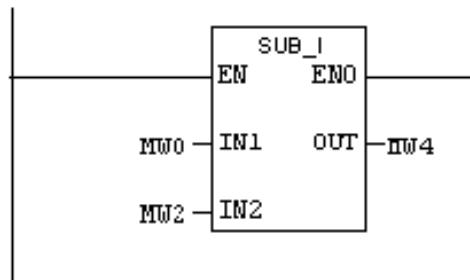
Lệnh ADD_DI : Lệnh thực hiện việc cộng 2 số nguyên 32 Bit, kết quả cất vào số nguyên 32 Bit, nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MD108 = MD100 + MD104$$

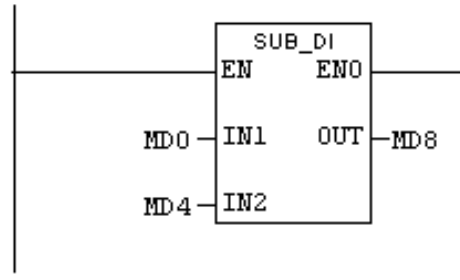


Lệnh trừ số nguyên

Ví dụ:



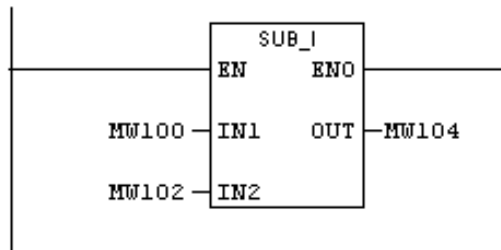
Số nguyên 16 bits



Số nguyên 32 bits

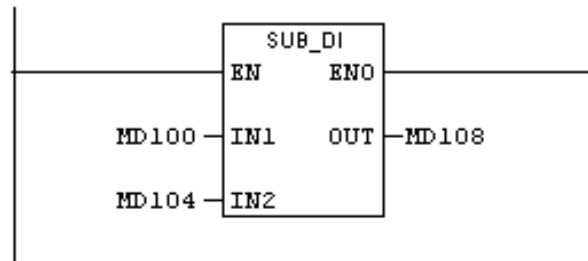
Lệnh SUB_I: Lệnh thực hiện việc trừ 2 số nguyên 16 Bit, kết quả cất vào số nguyên 16 Bit, nếu kết quả vượt quá 16 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MW104 = MW100 - MW102$$



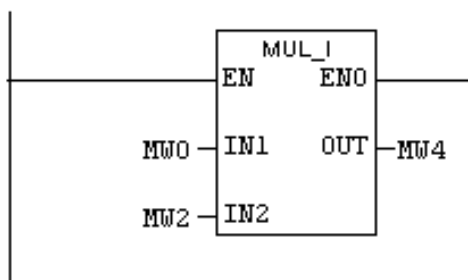
Lệnh SUB_DI: Lệnh thực hiện việc trừ 2 số nguyên 32 Bit, kết quả cất vào số nguyên 32 Bit, nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MD108 = MD100 - MD104$$

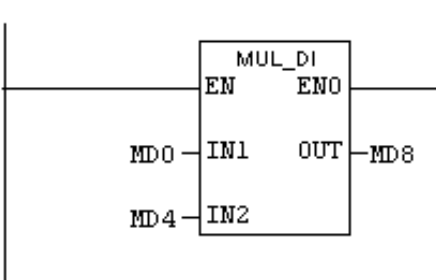


Lệnh nhân số nguyên

Ví dụ:



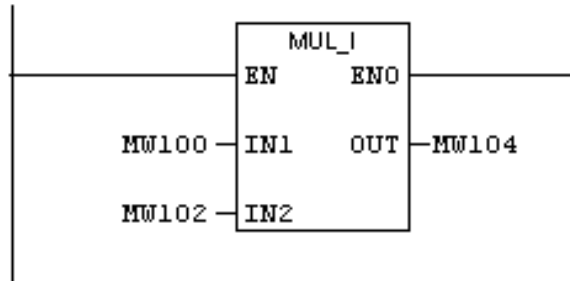
Số nguyên 16 bits



Số nguyên 32 bits

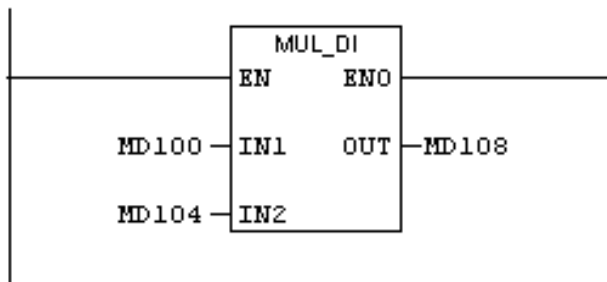
Lệnh MUL_I : : Lệnh thực hiện việc nhân 2 số nguyên 16 Bit, kết quả cất vào số nguyên 16 Bit, nếu kết quả vượt quá 16 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MW104 = MW100 * MW102$$



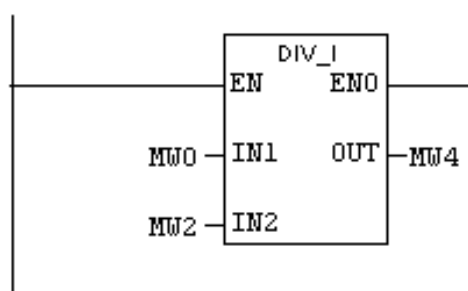
Lệnh MUL_DI : : Lệnh thực hiện việc nhân 2 số nguyên 32 Bit, kết quả cất vào số nguyên 32 Bit, nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MD108 = MD100 * MD104$$

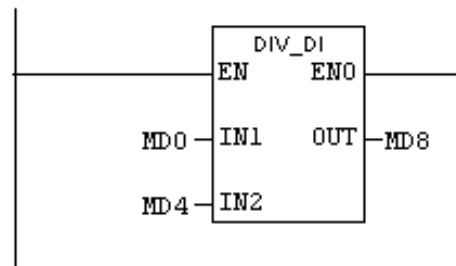


Lệnh chia số nguyên

Ví dụ:



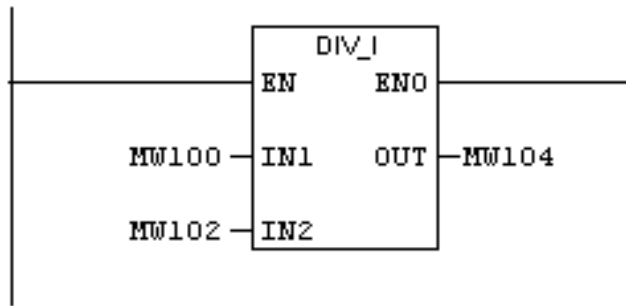
Số nguyên 16 bits



Số nguyên 32 bits

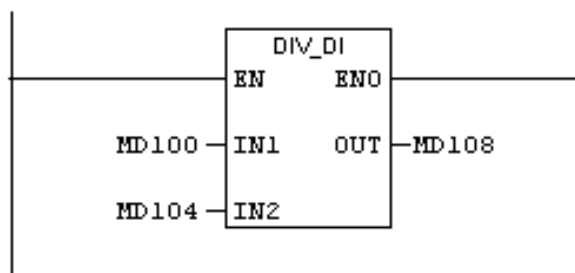
Lệnh DIV_I : Lệnh thực hiện việc chia 2 số nguyên 16 Bit, kết quả cất vào số nguyên 16 Bit, nếu kết quả vượt quá 16 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MW104 = MW100 : MW102$$



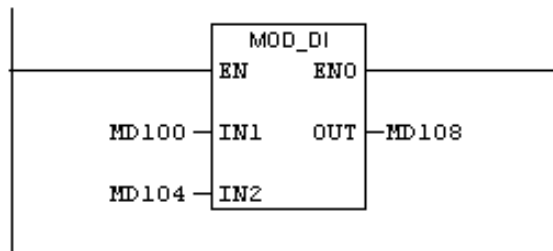
Lệnh DIV_DI : Lệnh thực hiện việc chia 2 số nguyên 32 Bit, kết quả cất vào số nguyên 32 Bit , nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

$$MD108 = MD100 : MD104$$



Lệnh MOD_DI : Lệnh xác định phần dư của phép chia 2 số nguyên 32 Bit, kết quả cất vào số nguyên 32 Bit, nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

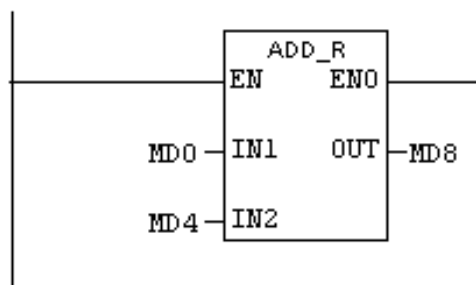
$$MD108 = MD100 \text{ mod } MD104$$



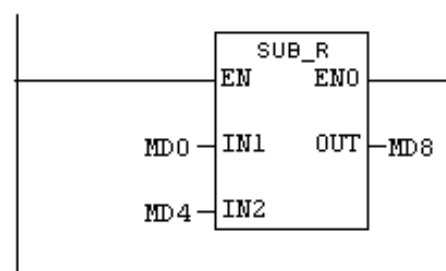
Với số thực

Lệnh cộng số thực

Ví dụ:



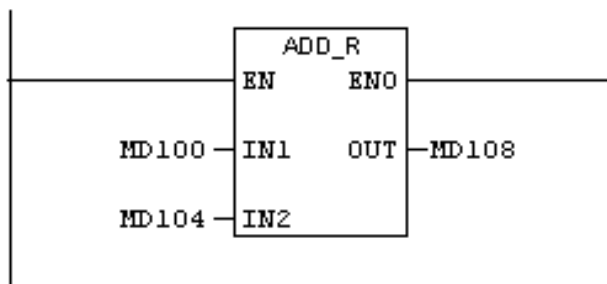
Lệnh trừ số thực



Lệnh ADD_R : Lệnh thực hiện việc cộng 2 số thực, kết quả cất vào số thực, nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ

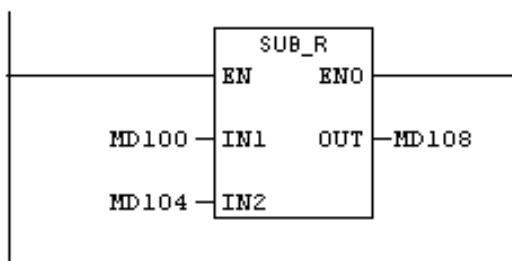
lưu Bit bị tràn đó.

$$MD108 = MD100 + MD104$$



Lệnh SUB_R : Lệnh thực hiện việc trừ 2 số thực, kết quả cất vào số thực , nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

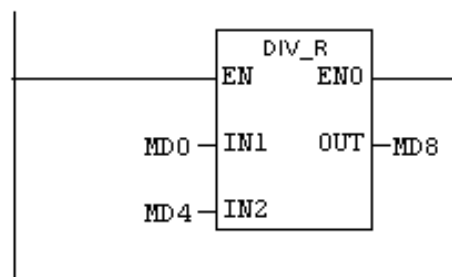
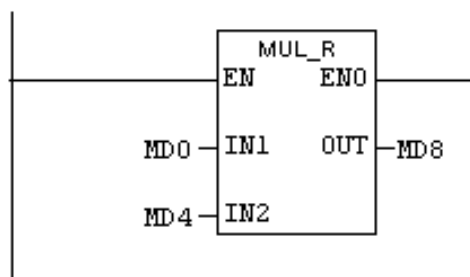
$$MD108 = MD100 - MD104$$



Lệnh nhân số thực

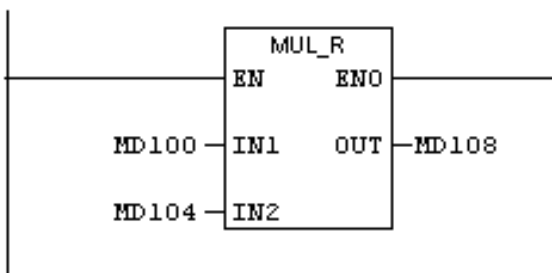
Lệnh chia số thực

Ví dụ:



Lệnh MUL_R : Lệnh thực hiện việc nhân 2 số thực, kết quả cất vào số thực, nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

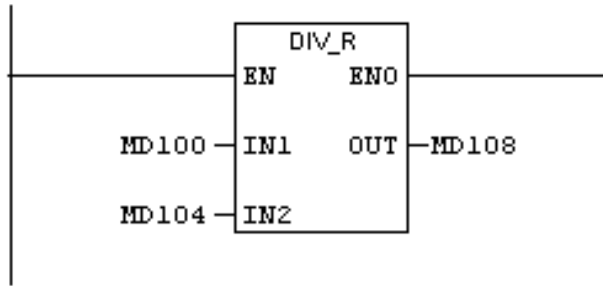
$$MD108 = MD100 * MD104$$



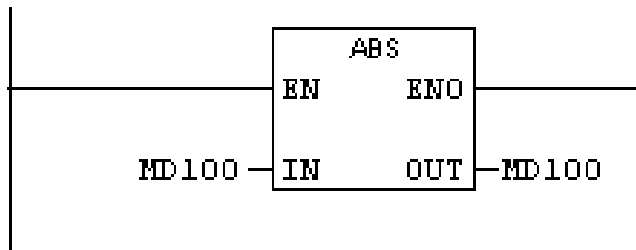
Lệnh DIV_R : Lệnh thực hiện việc chia 2 số thực, kết quả cất vào số

thực, nếu kết quả vượt quá 32 Bit thì cờ OV sẽ bật lên 1, cờ OS sẽ lưu Bit bị tràn đó.

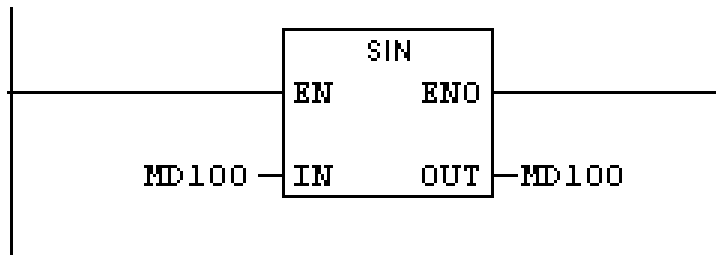
$$MD108 = MD100 : MD104$$



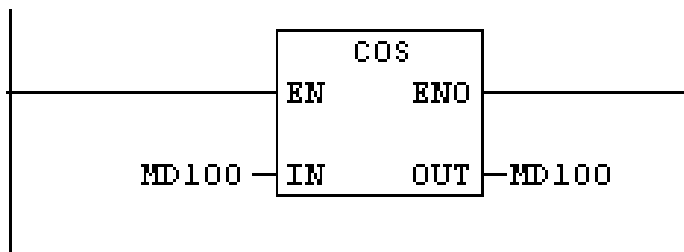
Lệnh ABS: Lệnh xác định giá trị tuyệt đối của số thực, kết quả cất vào số thực.



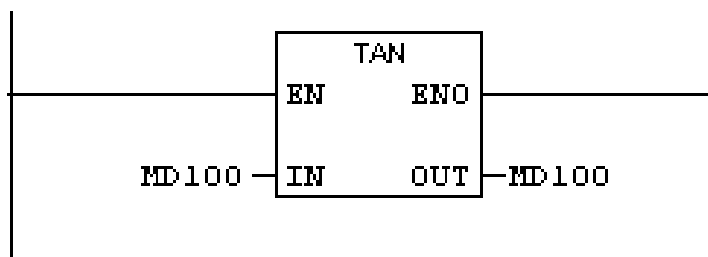
Lệnh SIN: Lệnh tính SIN của số thực, kết quả cất vào số thực. Nếu kết quả nằm ngoài khoảng $[-1,1]$ thì cờ OV bật lên 1



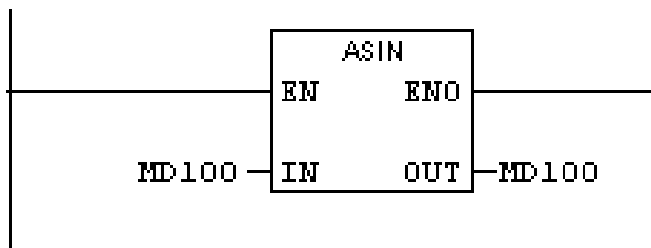
Lệnh COS: Lệnh tính COS của số thực, kết quả cất vào số thực. Nếu kết quả nằm ngoài khoảng $[-1,1]$ thì cờ OV bật lên 1



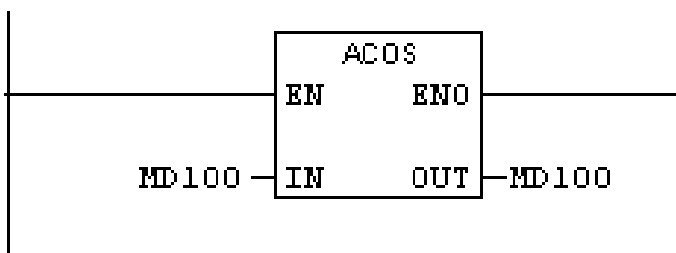
Lệnh TAN: Lệnh tính TAN của số thực, kết quả cất vào số thực. Nếu kết quả nằm ngoài khoảng 16Bit thì cờ OV bật lên 1



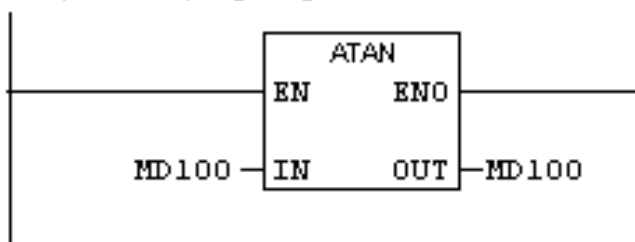
Lệnh ASIN: Lệnh tính Arcsin của số thực, số thực phải nằm trong khoảng $[-1,1]$ kết quả là 1 số thực trong khoảng $[-\pi/2, \pi/2]$ và được cất vào số thực.



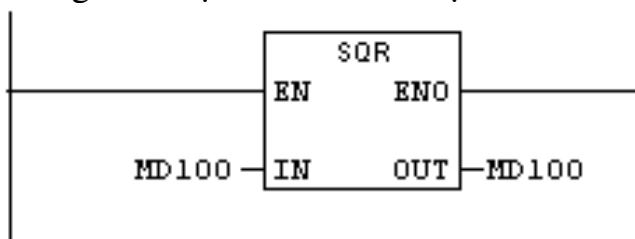
Lệnh ACOS: Lệnh tính Arccos của số thực, số thực phải nằm trong khoảng $[-1,1]$ kết quả là 1 số thực trong khoảng $[-\pi, 0]$ và được cất vào số thực.



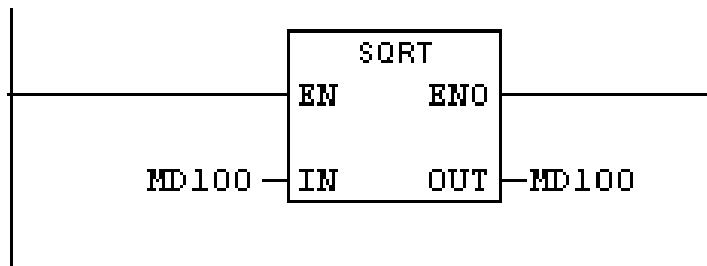
Lệnh ATAN: Lệnh tính Arctang của số thực, kết quả là 1 số thực trong khoảng $[-\pi/2, \pi/2]$ và được cất vào số thực.



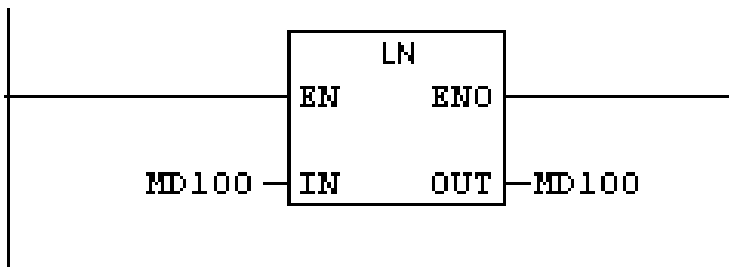
Lệnh SQR: Lệnh tính bình phương của số thực, kết quả là 1 số thực không âm được cất vào số thực.



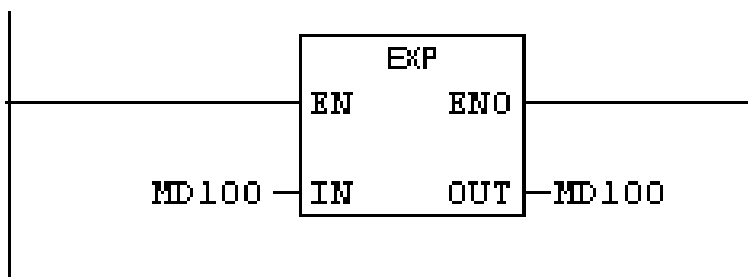
Lệnh SQRT: Lệnh tính căn bậc hai của số thực, số thực này phải là 1 số thực không âm, kết quả là 1 số thực không âm được cất vào số thực.



Lệnh Ln: Lệnh tính $\ln(x)$ của số thực, số thực này phải là 1 số thực không âm, kết quả là 1 số thực được cất vào số thực.



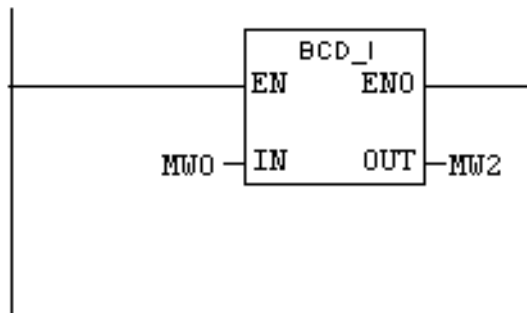
Lệnh EXP: Lệnh tính e^x của số thực, kết quả là 1 số thực không âm được cất vào số thực.



2.4 Lệnh đổi kiểu dữ liệu và di chuyển

Các lệnh đổi kiểu dữ liệu: Số BCD → số nguyên 16 bits

Ví dụ:



Lệnh *BCD_I*: Chuyển đổi từ số định dạng dưới dạng BCD (chứa 3 Digit) sang số nguyên 16 Bit.

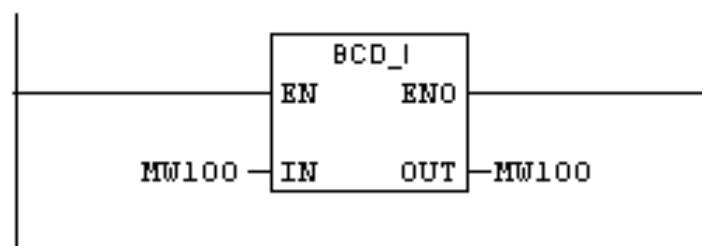
Số BCD có tầm (+/- 999) chứa trong 12Bit.

Vd: MW100 = 22 được định dạng dưới dạng BCD như sau:

0000 0010 0010
 2 2

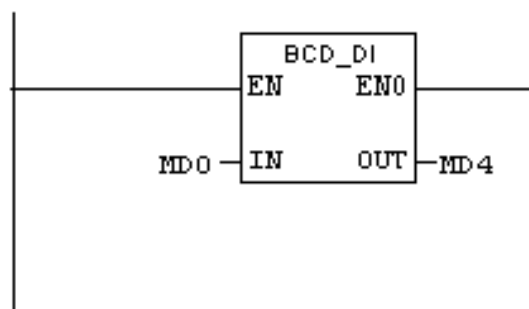
Sau khi thực hiện lệnh chuyển đổi thành số Integer 16 Bit được định dạng:

0000 0000 0001 0110 := $1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 = 22$



Số BCD → số nguyên 32 bits

Ví dụ:



Lệnh BCD_DI : Chuyển đổi từ số định dạng dưới dạng BCD (chứa 7 Digit) sang số nguyên 32 Bit.

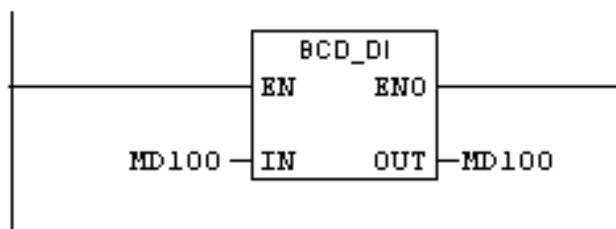
Số BCD có tầm (+/- 9999999) chứa trong 28Bit.

Vd: MD100 =22 được định dạng dưới dạng BCD như sau:

0000 0000 0000 0000 0000 0010 0010
2 2

Sau khi thực hiện lệnh chuyển đổi thành số DoubleInteger 32 Bit được định dạng:

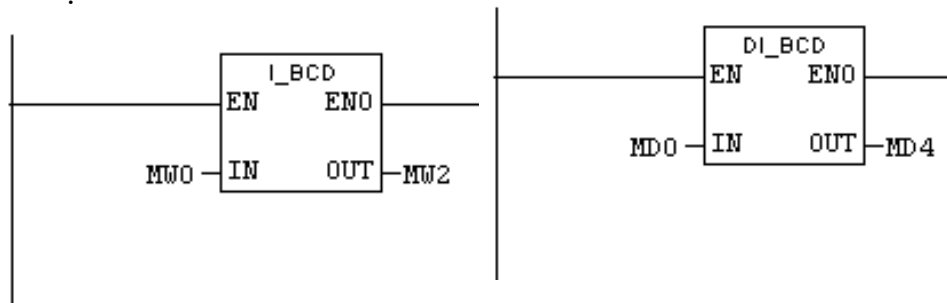
0000 0000 0000 0000 0000 0000 0001 0110 =
 $1*2^4 + 1*2^2 + 1*2^1 = 22$



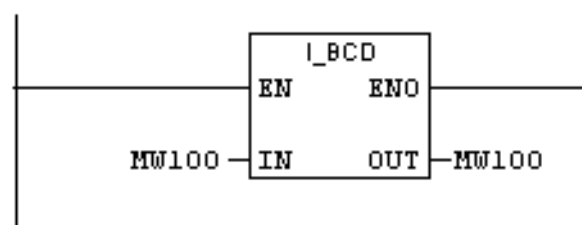
Số nguyên 16 bits → BCD

Số nguyên 32 bits → BCD

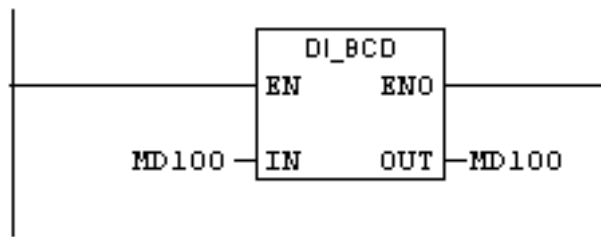
Ví dụ:



Lệnh I_BCD: Chuyển đổi từ số nguyên sang số được định dạng dưới dạng BCD (chứa 3 Digit), do số BCD tối đa 999 nên số nguyên phải tối đa 999.

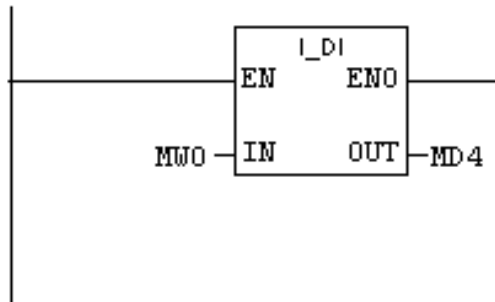


Lệnh DI_BCD: : Chuyển đổi từ số nguyên 32 Bit sang số được định dạng dưới dạng BCD (chứa 7 Digit), do số BCD tối đa 9999999 nên số nguyên phải tối đa 9999999.

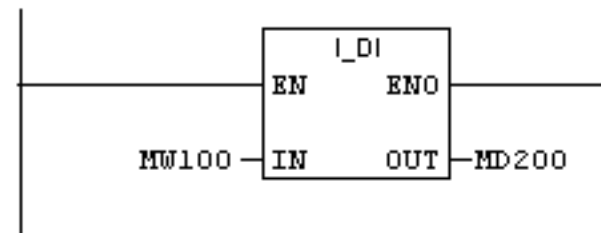


Số nguyên 16 bits → số nguyên 32 bits

Ví dụ:

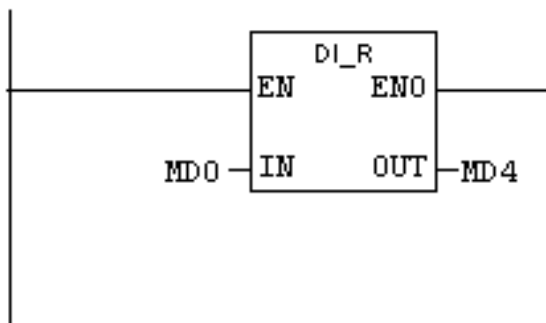


Lệnh I_DI : Chuyển đổi số nguyên từ 16Bit sang số nguyên 32 Bit để thực hiện cho các phép toán trên số 32 Bit.



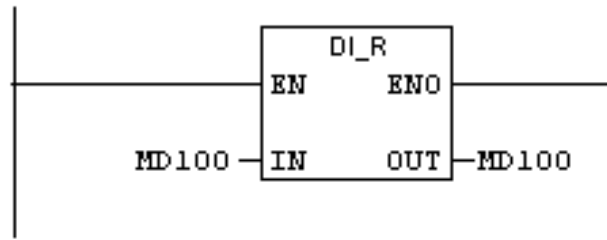
Số nguyên 32 bits → số thực

Ví dụ:



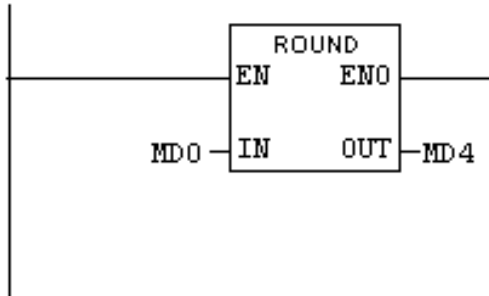
Lệnh DI_R : Chuyển đổi từ số nguyên 32 Bit sang số thực để phục vụ cho các phép toán trên số thực.

Vd: MD100= 457 ; Sau phép toán giá trị mới là MD100=457.0



Lệnh làm tròn số (số thực → số nguyên 32 bits)

Ví dụ:

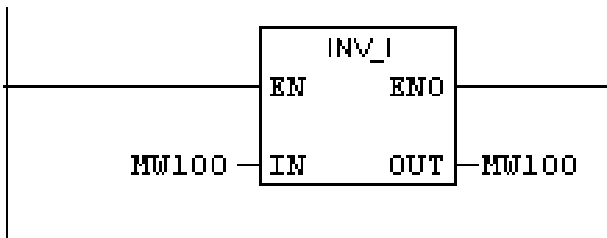


Lệnh Đảo

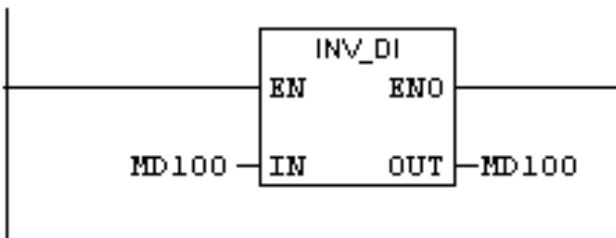
Lệnh INV_I : Đảo tất cả các Bit của số nguyên 16 Bit.

Vd : MW100 ban đầu là 0101 0111 1000 0101 ; sau lệnh chuyển đổi.

MW100 lúc sau 1010 1000 0111 1010



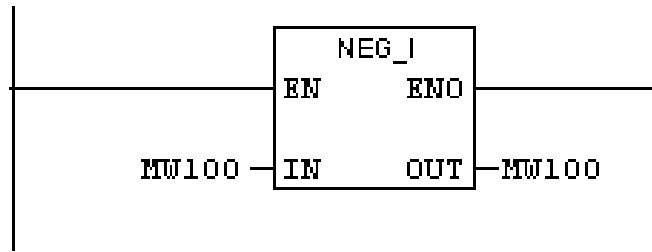
Lệnh INV_DI : Đảo tất cả các Bit của số nguyên 32 Bit.



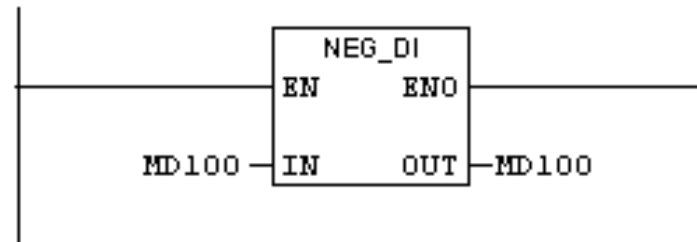
Lệnh Đảo dấu số nguyên:

Lệnh NEG_I : Đảo dấu số nguyên 16 Bit.

Vd: MW100 = 8 sau lệnh MW100 = -8.

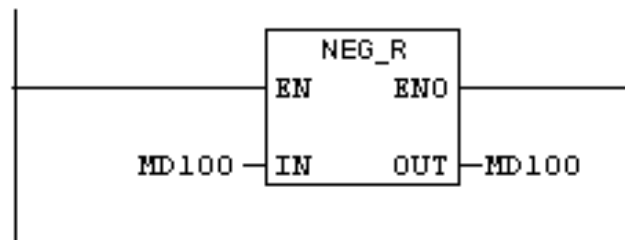


Lệnh *NEG_DI* : Đổi dấu số nguyên 32 Bit.



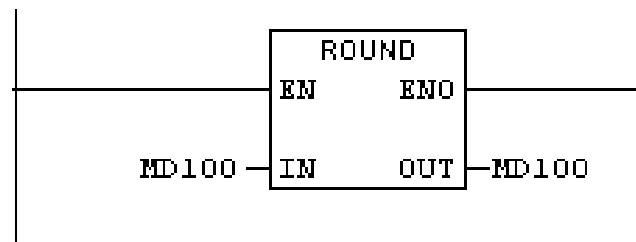
Lệnh Đổi dấu số thực:

Lệnh *NEG_R* : Đổi dấu số thực.

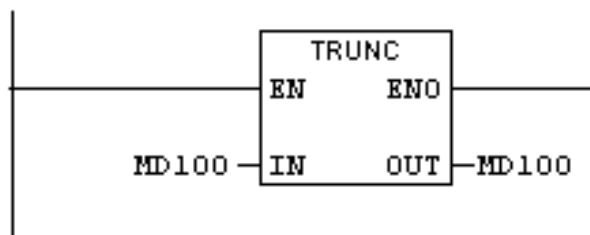


Lệnh *Round* : Lệnh chuyển đổi số thực thành số nguyên 32 Bit bằng cách làm tròn.

Vd: MD100 = 20.35 làm tròn thành 20.



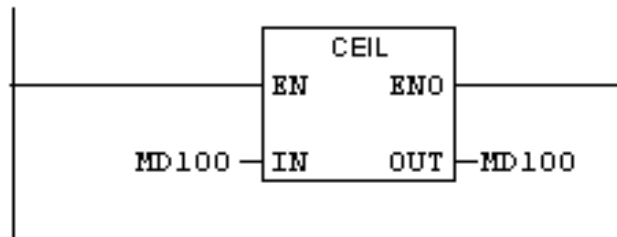
Lệnh *Trunc*: Lệnh chuyển đổi số thực thành số nguyên 32 Bit bằng cách cắt phần nguyên.



Vd: MD100 = 20.56 chuyển thành 20.

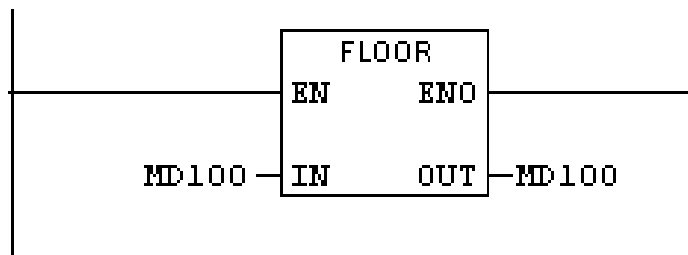
Lệnh Ceil: Lệnh chuyển đổi số thực thành số nguyên 32 Bit bằng cách làm tròn lên.

Vd: MD100 = 20.04 làm tròn lên thành 21.



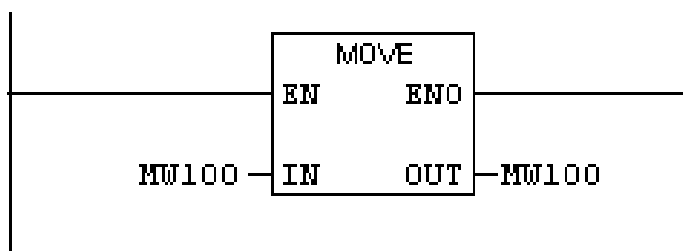
Lệnh Floor: Lệnh chuyển đổi số thực thành số nguyên 32 Bit bằng cách làm tròn xuống

Vd: MD100 = 23.45 làm tròn xuống còn 23

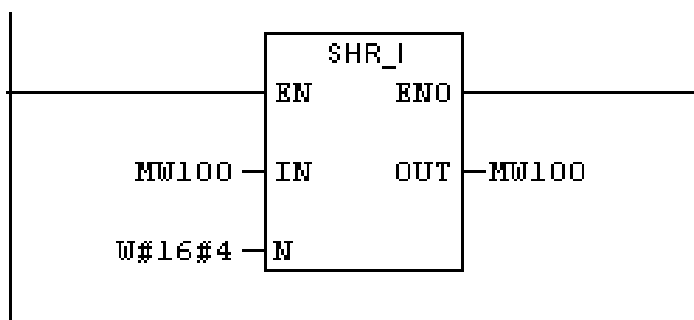


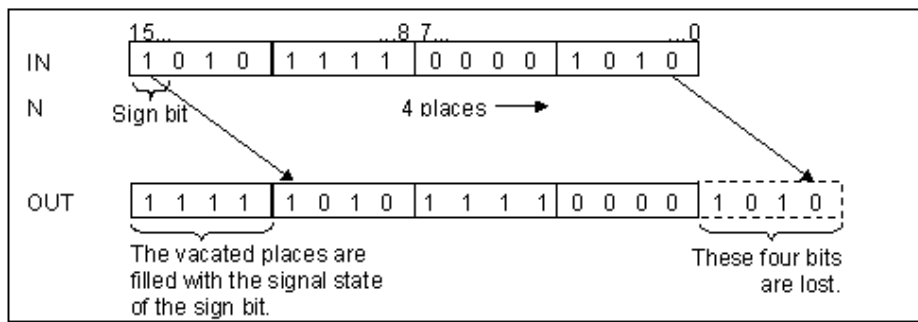
Các lệnh di chuyển

Lệnh MOV : Lệnh đưa giá trị một ô nhớ sang 1 ô nhớ khác, lệnh này có thể áp dụng cho mọi kiểu số khác nhau. (Int, Dint, Real, Byte....)

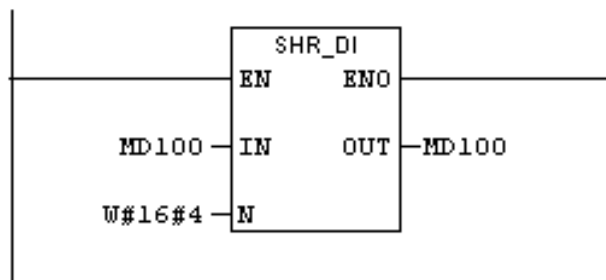


Lệnh SHR_I: Lệnh thực hiện việc dịch phải ô nhớ 16Bit, kết quả cất vào ô nhớ 16 Bit, N là số Bit dịch.



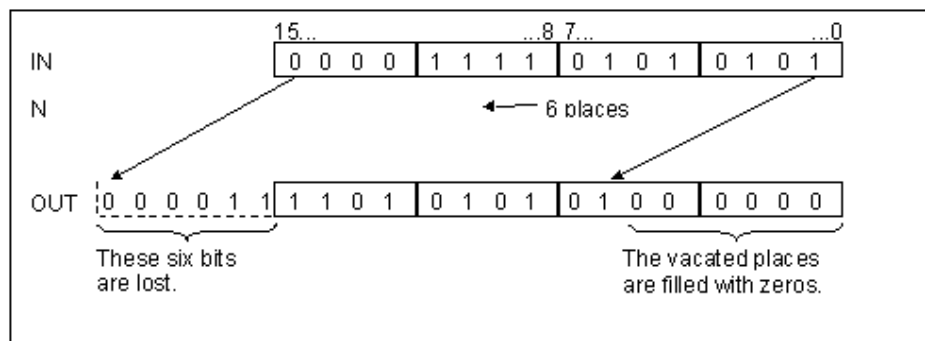
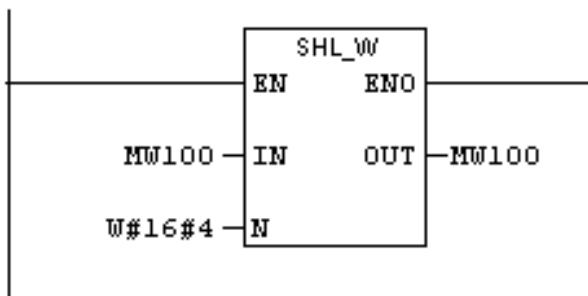


Lệnh SHR_DI: Lệnh thực hiện việc dịch phải ô nhớ 32Bit, kết quả cất vào ô nhớ 32 Bit, N là số Bit dịch.

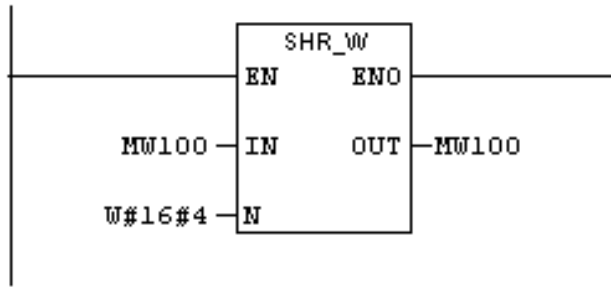


Lệnh SHL_W: Lệnh thực hiện việc dịch trái ô nhớ 16Bit, kết quả cất vào ô nhớ 16 Bit, N là số Bit dịch. Ô nhớ này được định dạng theo kiểu Word.

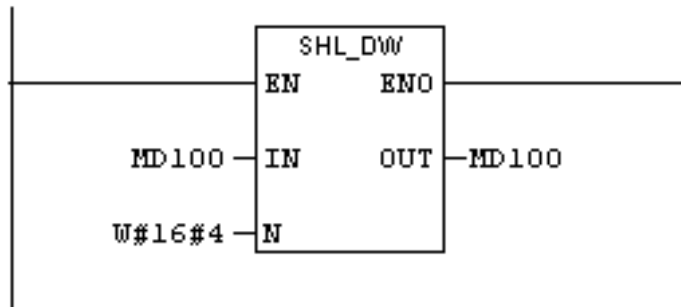
Nếu N lớn hơn 16 thì MW100 =0 và cờ CC0, OV trong thanh ghi trạng thái đều bằng 0.



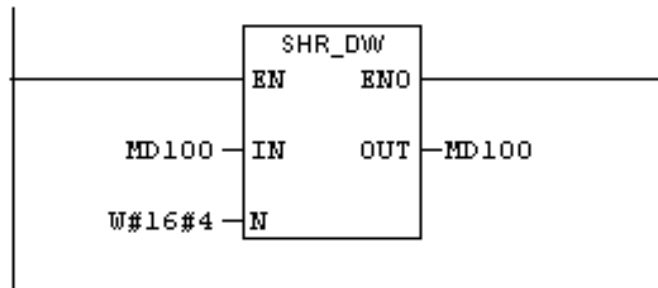
Lệnh SHR_W: Lệnh thực hiện việc dịch phải ô nhớ 16Bit, kết quả cất vào ô nhớ 16 Bit, N là số Bit dịch. Ô nhớ này được định dạng theo kiểu Word.



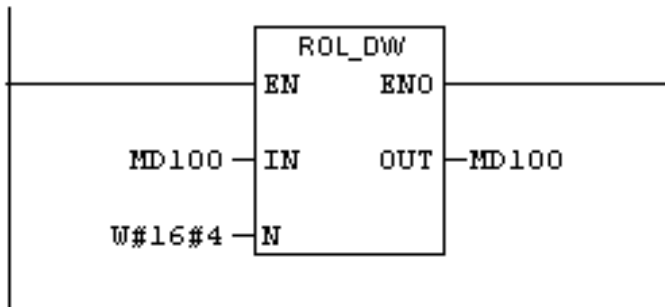
Lệnh SHL_DW: Lệnh thực hiện việc dịch trái ô nhớ 32Bit, kết quả cất vào ô nhớ 32 Bit, N là số Bit dịch. Ô nhớ này được định dạng theo kiểu Word.

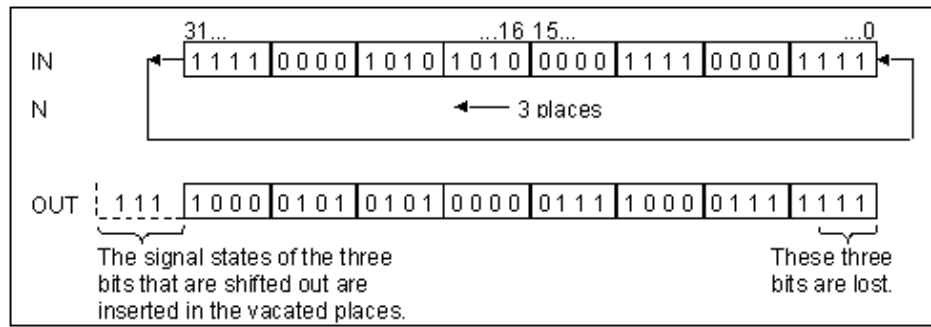


Lệnh SHR_DW: Lệnh thực hiện việc dịch phải ô nhớ 32Bit, kết quả cất vào ô nhớ 32 Bit, N là số Bit dịch. Ô nhớ này được định dạng theo kiểu Word.

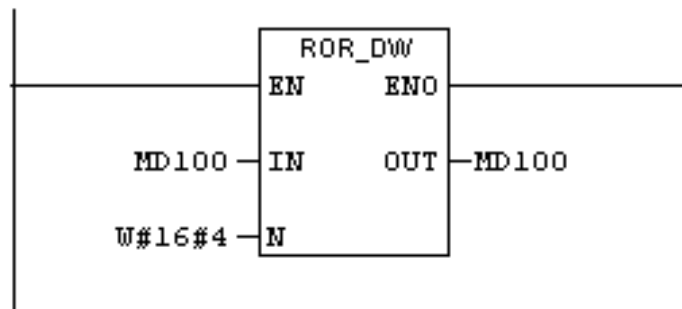


Lệnh ROL_DW: Lệnh thực hiện việc dịch trái xoay tròn ô nhớ 32Bit, N là số Bit dịch. Ô nhớ này được định dạng theo kiểu Word.





Lệnh ROR_DW: Lệnh thực hiện việc dịch phải xoay tròn ô nhớ 32Bit, N là số Bit dịch. Ô nhớ này được định dạng theo kiểu Word.



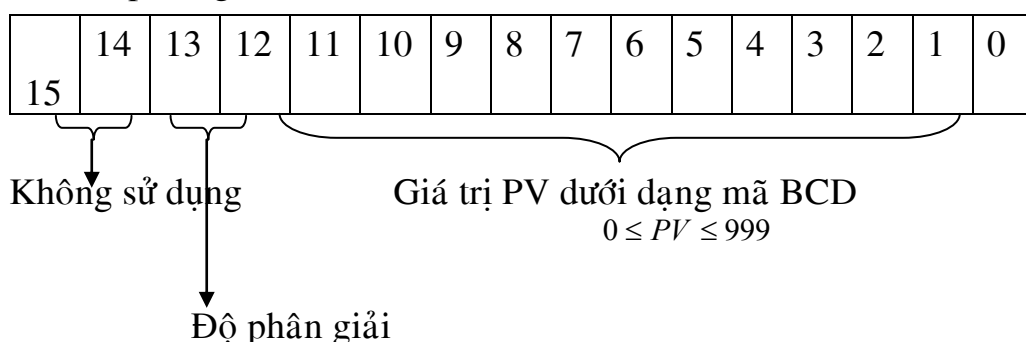
2.5 Bộ thời gian (Timer)

Bộ thời gian là bộ tạo thời gian trễ t mong muốn giữa tín hiệu logic đầu vào $u(t)$ và tín hiệu logic đầu ra $y(t)$.

S7_300 có 5 loại Timer khác nhau. Thời gian trễ t mong muốn được khai báo với Timer bằng 1 giá trị 16 bits trong đó 2 bits cao nhất không sử dụng, 2bits cao kế tiếp là độ phân giải của Timer, 12 bits thấp là 1 số nguyên BCD trong khoảng $0 \div 999$ được gọi là PV(Preset Value).

Thời gian trễ t chính là tích:

$$T = \text{Độ phân giải} * PV$$



0	0	10ms
0	1	100ms
1	0	1s
1	1	10s

Thời gian có thể được khai báo dưới dạng bằng kiểu S5T

Ví dụ: S5T#3s



Khi ngõ vào I0.0 lên mức 1 thì sau 2s T1 sẽ ON

Lệnh S_PULSE Nếu I0.0=1 Timer được kích chạy, khi I0.0=0 hoặc chạy đủ thời gian đặt 2s thì Timer dừng

Hoặc có tín hiệu I0.1 thì Timer cũng dừng

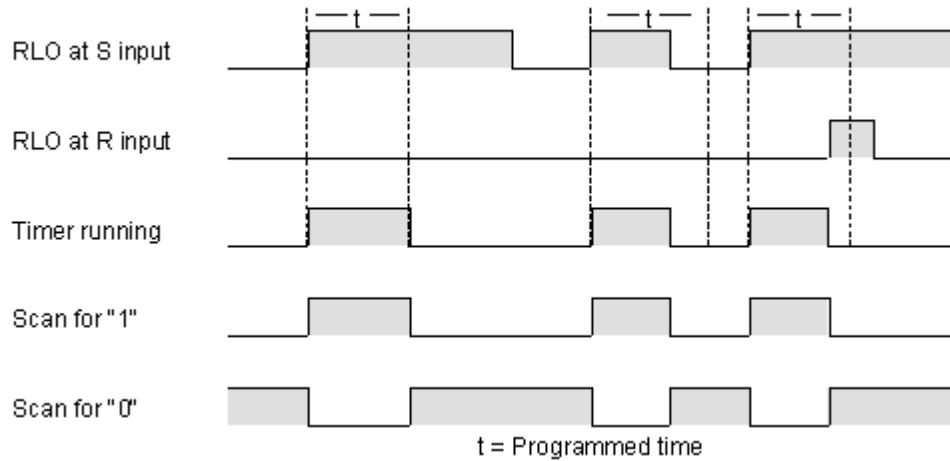
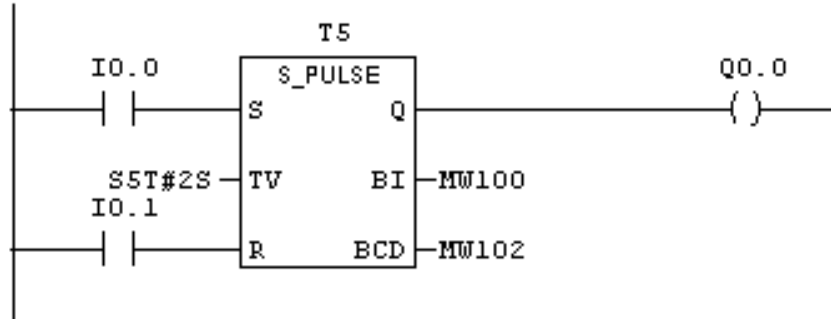
Timer chỉ chạy lại khi có tín hiệu mới từ I0.0 (tức là I0.0 chuyển trạng thái từ 0 lên 1)

Q0.0=1 khi Timer đang chạy.

MW100 lưu giá trị đếm của Timer theo dạng Integer

MW102 lưu giá trị của Timer theo dạng BCD

Chức năng của Timer này là tạo xung có thời gian được đặt sẵn.

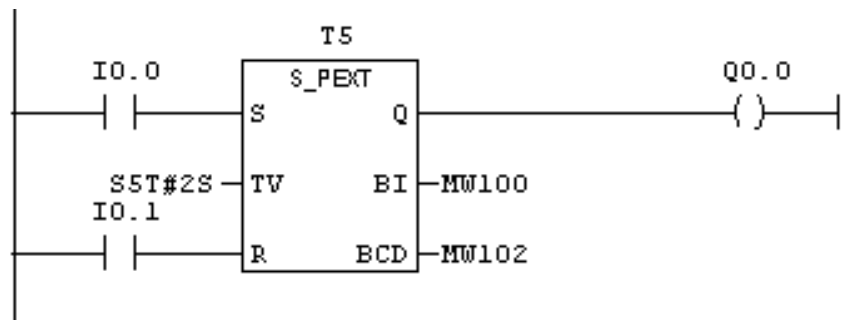


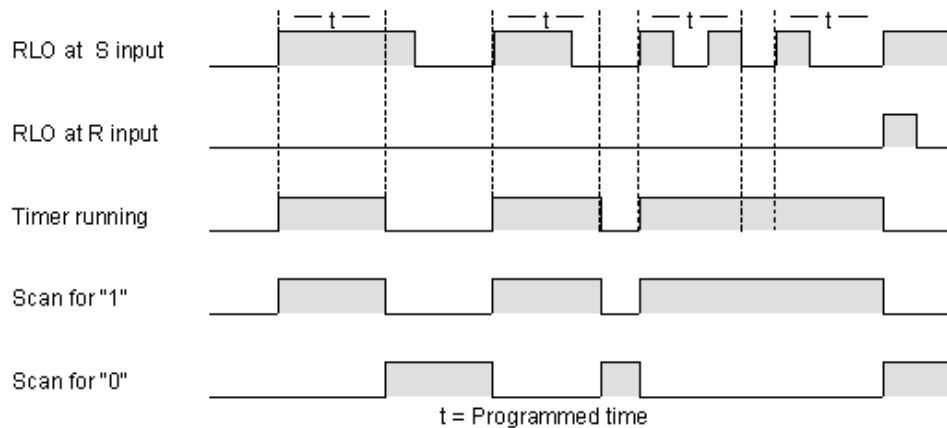
Lệnh S_PEXT Timer kích có nhớ, Khi có tín hiệu cạnh lên ở IO.0 Timer T5 chạy, nếu đủ thời gian đặt Timer dừng.

Trong quá trình chạy nếu có tín hiệu mới từ chân IO.0 thì thời gian Timer lại được tính lại từ đầu.

Trong quá trình chạy nếu có tín hiệu IO.1 thì Timer dừng. Q0.0 = 1 khi Timer đang chạy.

Các ô nhớ MW100 và MW102 lưu giá trị hiện thời của Timer theo dạng Integer và dạng BCD

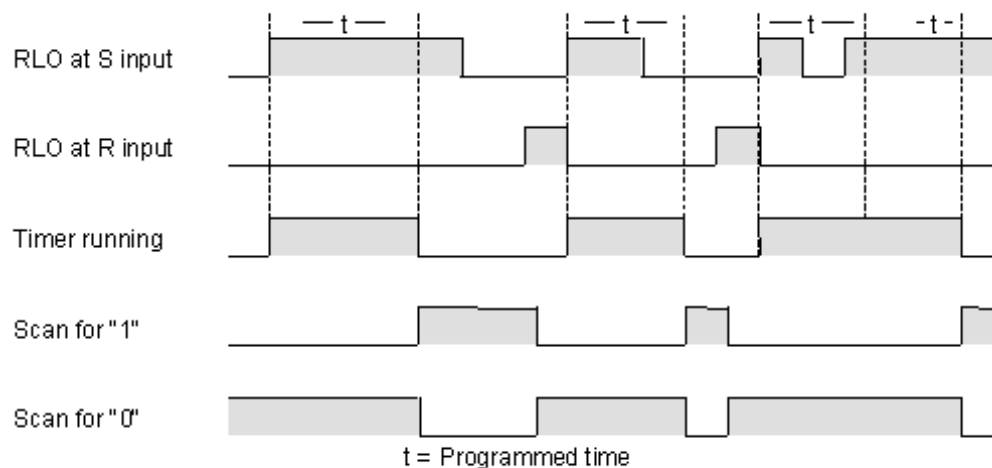
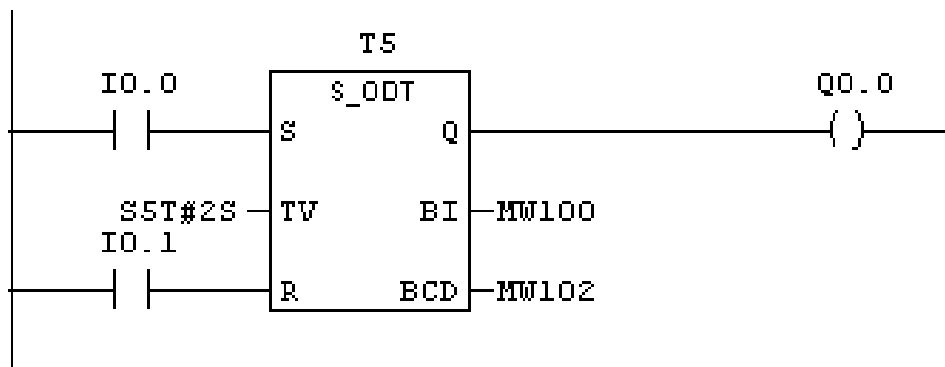




Lệnh S_ODT

Nếu I0.0=1 Timer bắt đầu chạy khi đủ thời gian thì ngưng khi đó ngõ Q0.0 sẽ lên 1 nếu I0.0 vẫn còn giữ trạng thái 1, khi có tín hiệu I0.1 thì tất cả phải được Reset về 0

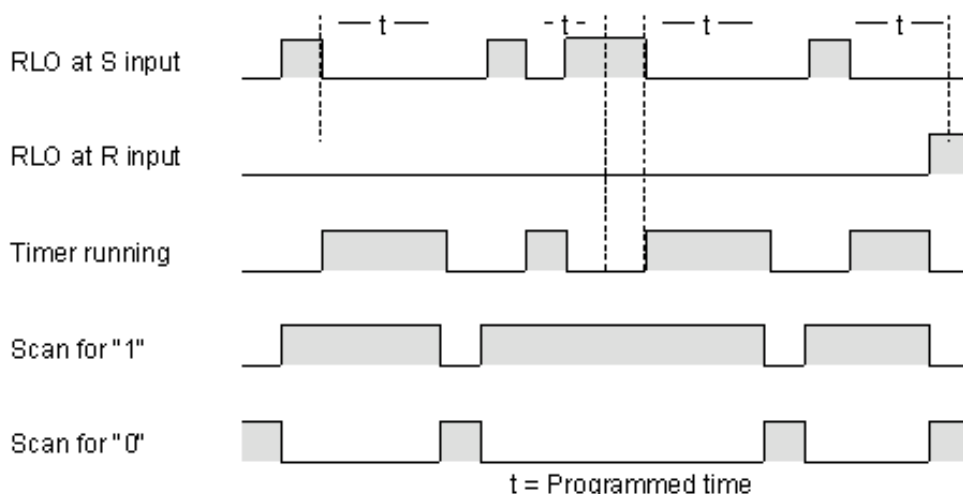
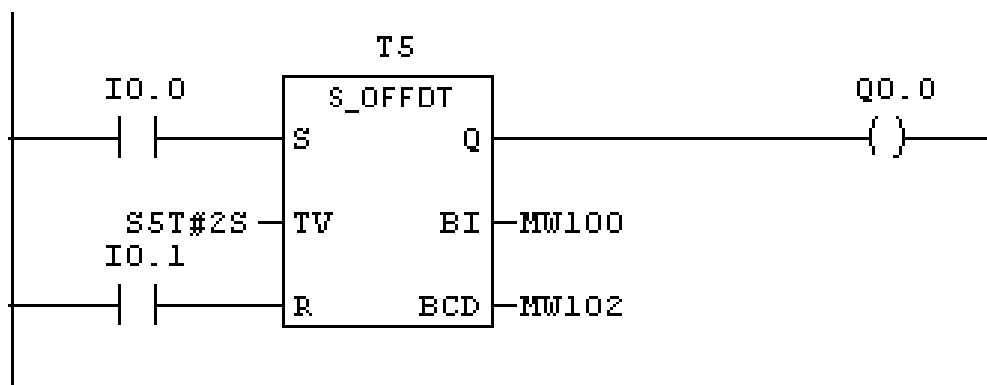
Các ô nhớ MW100 và MW102 lưu giá trị hiện thời của Timer theo dạng Integer và dạng BCD



Lệnh S_OFFDT

Khi I0.0 ON , Q0.0 =1 ,khi I0.0 OFF Timer bắt đầu chạy và Q0.0 chỉ tắt khi đủ thời gian và I0.0 vẫn OFF

Khi có tín hiệu Reset I0.1 thì tất cả tín hiệu đều OFF



Cài đặt Timer

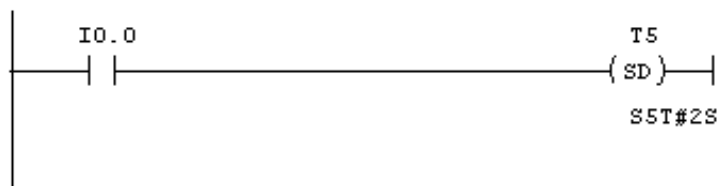
Số Timer trong S7_300 phụ thuộc vào loại CPU.

CPU 312: có 128 Timer

CPU 313 trở lên: có 256 Timer.

Có 2 cách cài đặt giá trị cho Timer:

Lệnh TON:



Parameter	Data Type	Memory Area	
<T no.>	TIMER	T	Số hiệu timer
<time value>	S5TIME	I, Q, M, L, D	Giá trị đặt cho timer

Cài thông số thời gian trực tiếp:

Để cài giá trị trực tiếp cho Timer ta phải thêm kí tự S5T# trước giá

trị đặt. Các kí tự kế tiếp là thông số thời gian muốn cài đặt cho Timer.

Tổng quát như sau: S5T#aH_bM_cS_dMS. Trong đó:

H: giờ

M: phút

S: giây

MS: mili giây

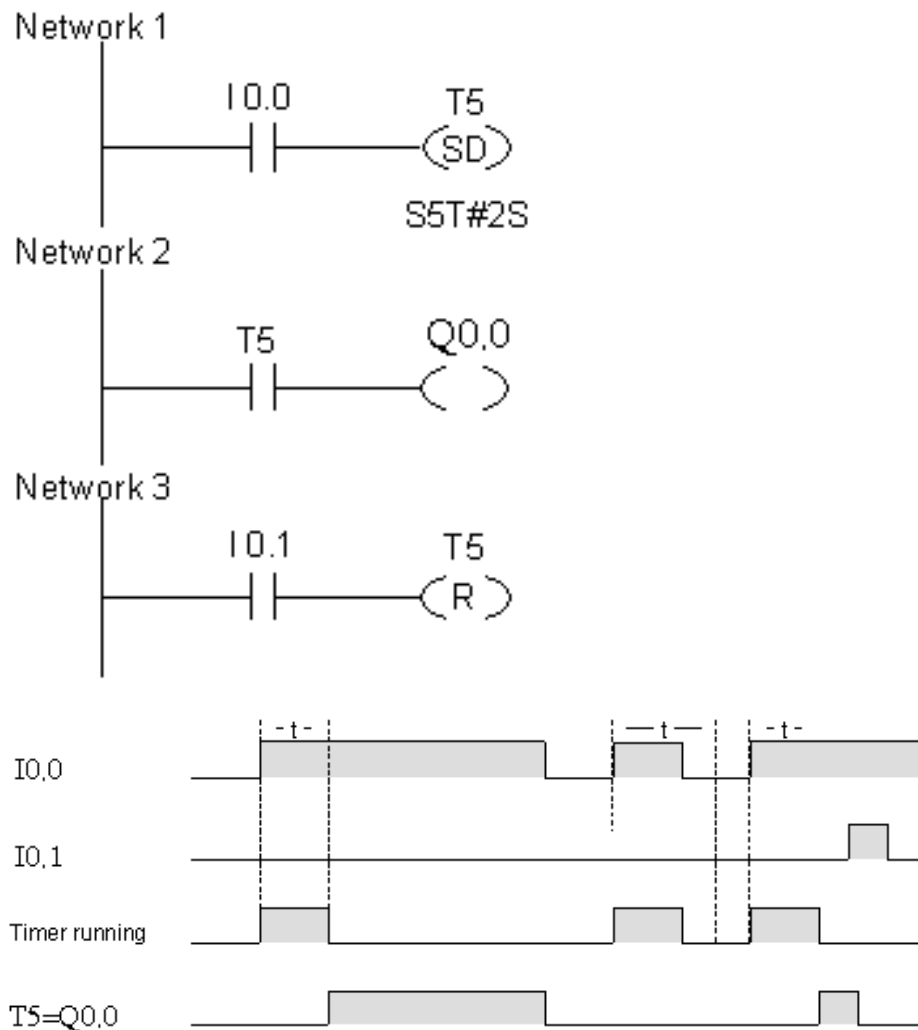
a,b,c,d: các thông số cài đặt.

VD: S5T#3S: thời gian cài đặt là 3s

S5T#7S500MS: thời gian cài đặt là 7,5s

Trong VD trên, khi I0.0 ON, Timer T5 sẽ được kích hoạt. Đủ thời gian cài đặt là 2 s thì bit T5 tác động làm cho Q0.0 ON.

Khi ngõ vào I0.1 tác động thì Timer được reset. Giá trị hiện tại của Timer cũng như Bit T5 được Reset về 0.



<u>Time Base</u>	<u>Binary Code for the Time Base</u>
10 ms	00
100 ms	01
1 s	10
10 s	11

12 bit kế tiếp là giá trị cài đặt thời gian cho Timer dưới dạng số BCD (giá trị từ 0-999). Như trong VD trên thì giá trị cài đặt cho Timer sẽ là 127s.

Như vậy để có thể cài đặt giá trị cho Timer thay đổi theo ô nhớ:

Ta phải thực hiện các bước:

Giá trị Timer phải bé hơn hoặc bằng 999

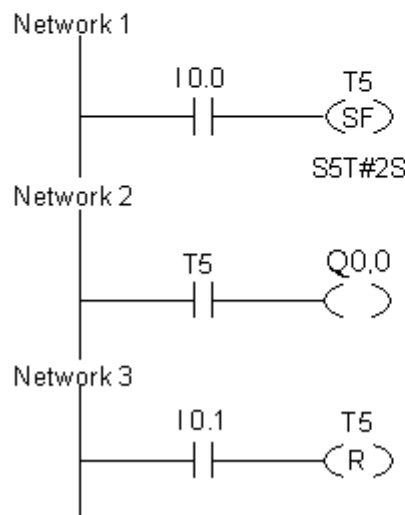
Chuyển giá trị đó sang dạng BCD dùng lệnh I_BCD

Sau đó chọn Time Base theo mong muốn như bảng trên bằng cách chọn 4 Bit đầu.

TOFF:

<T no.>
 ---(SF)
 <time value>

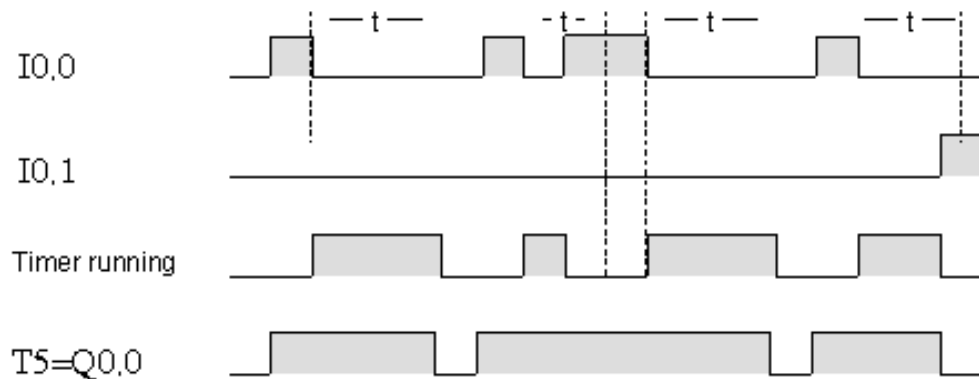
<u>Parameter</u>	<u>Data Type</u>	<u>Memory Area</u>	
<T no.>	TIMER	T	Số hiệu timer
<time value>	S5TIME	I, Q, M, L, D	Giá trị đặt cho timer



Trong VD trên, khi I0.0 ON, Bit T5 sẽ ON ngay khi I0.0 ON. Khi I0.0 chuyển từ ON sang OFF, Timer T5 sẽ được kích hoạt. Đủ thời gian cài đặt là 2 s thì Timer T5 tác động, bit T5 OFF làm cho Q0.0 OFF.

Khi ngõ vào I0.1 tác động thì Timer được reset. Giá trị hiện tại của Timer cũng như Bit T5 được Reset về 0.

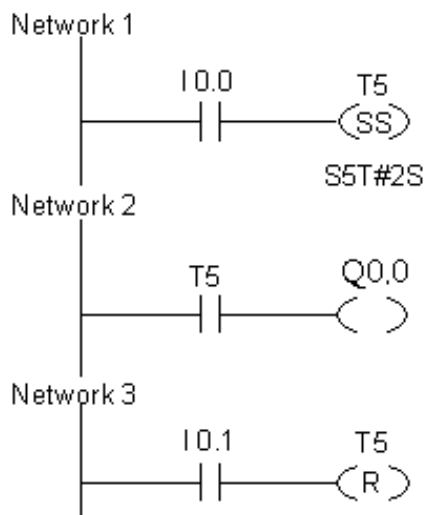
Cách cài đặt thông số thời gian của Timer OFF tương tự như Timer ON.



TON có nhớ:

<T no.>
 --(SS)
 <time value>

Parameter	Data Type	Memory Area	
<T no.>	TIMER	T	Số hiệu timer
<time value>	S5TIME	I, Q, M, L, D	Giá trị đặt cho timer



Trong VD trên, khi I0.0 ON, Timer T5 sẽ được kích hoạt. Đủ thời gian cài đặt là 2 s thì bit T5 tác động làm cho Q0.0 ON. Trong trường hợp thời gian chưa đủ 2S mà I0.0 chuyển OFF sang ON một lần nữa, giá trị đếm của Timer sẽ được khởi động lại.

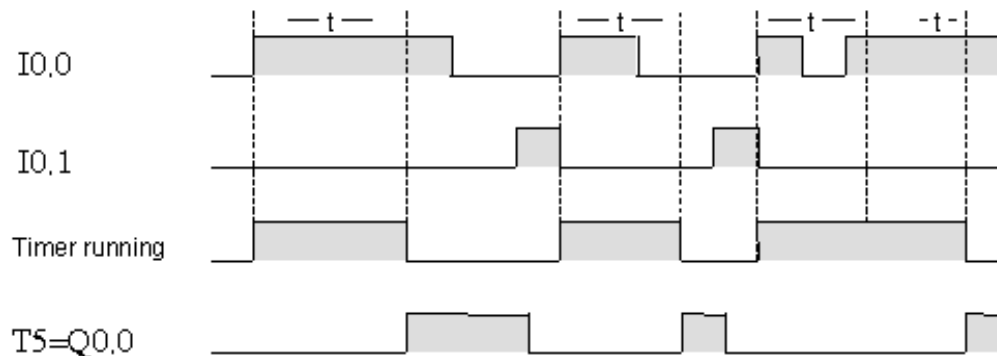
Giữa Timer ON và Timer ON có nhớ còn khác nhau một điểm nữa như sau:

Timer ON: sau khi Timer tác động, Bit của Timer được bật ON,

nếu tín hiệu kích Timer mất đi thì Timer sẽ được Reset, Bit timer sẽ OFF.

Timer ON có nhớ: sau khi Timer tác động, Bit của Timer được bật ON, nếu tín hiệu kích Timer mất đi thì Timer vẫn không Reset, Bit timer sẽ vẫn ON.

Khi ngõ vào I0.1 tác động thì Timer được reset. Giá trị hiện tại của Timer cũng như Bit T5 được Reset về 0.



Timer xung:

<T no.>

--(SP)

<time value>

Parameter

Data Type

Memory Area

<T no.>

TIMER

T

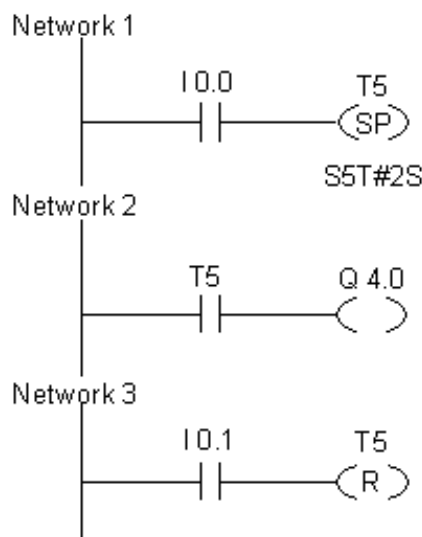
Số hiệu timer

<time value>

S5TIME

I, Q, M, L, D

Giá trị đặt cho timer



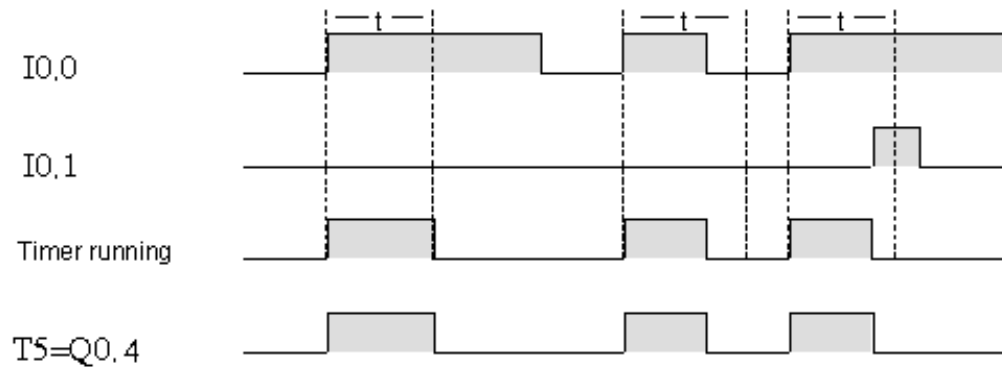
Mô tả:

Khi I0.0 chuyển từ 0 lên 1, Timer T5 sẽ được khởi động, ngõ ra bit T5 sẽ ON ngay lập tức.

Khi hết thời gian cài đặt là 2s thì bit T5 OFF (nếu ngõ vào I0.0 vẫn

còn ON). Trong trường hợp chưa đủ 2s mà ngõ vào I0.0 đã OFF, Timer sẽ được reset và ngõ ra bit T5 OFF.

Trong khi Timer chạy mà chưa đủ 2s, nếu I0.1 chuyển từ 0 lên 1. Ngõ ra bit T5 sẽ OFF và thời gian được reset.



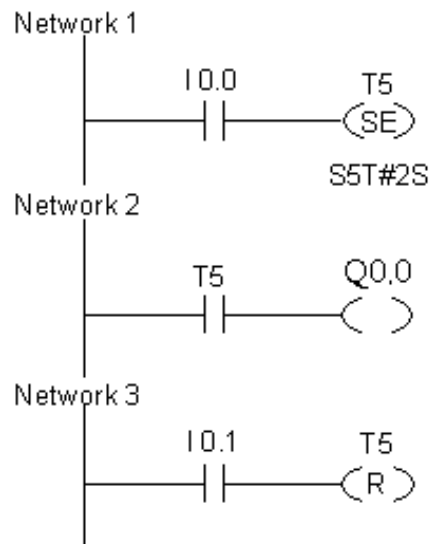
Timer xung mở rộng:

<T no.>

---(SE)

<time value>

Parameter	Data Type	Memory Area	
<T no.>	TIMER	T	Số hiệu timer
<time value>	S5TIME	I, Q, M, L, D	Giá trị đặt cho timer

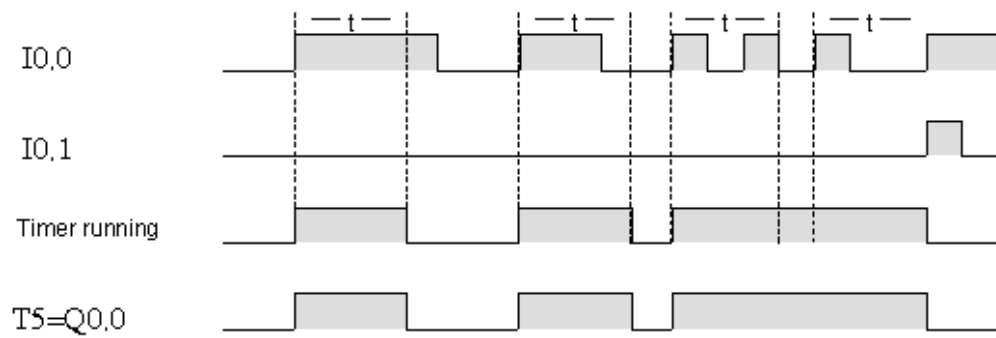


Mô tả:

Khi I0.0 chuyển từ 0 lên 1, Timer T5 sẽ được khởi động, ngõ ra bit T5 sẽ ON ngay lập tức.

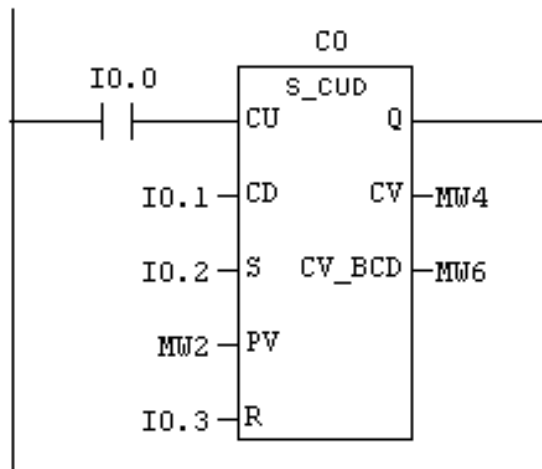
Khi hết thời gian cài đặt là 2s thì bit T5 OFF (bất kể ngõ vào I0.0 vẫn còn ON hay đã OFF). Trong trường hợp chưa đủ 2s mà ngõ vào I0.0 chuyển từ OFF lên ON một lần nữa, Timer sẽ được khởi động lại.

Khi I0.1 chuyển từ 0 lên 1. Ngõ ra bit T5 sẽ OFF và thời gian được reset.



2.6 Bộ đếm (Counter)

Counter là bộ đếm có chức năng đếm sườn xung của tín hiệu đầu vào. Có tối đa 256 Counter được kí hiệu từ C0 ÷ C255.



Ví dụ: Loại Counter đếm lên và đếm xuống

CU: tín hiệu đếm lên (BOOL)

CD: tín hiệu đếm (BOOL)

S: tín hiệu đặt (BOOL), khi có sườn lên thì giá trị đặt được nạp cho CV

P: giá trị đặt (WORD)

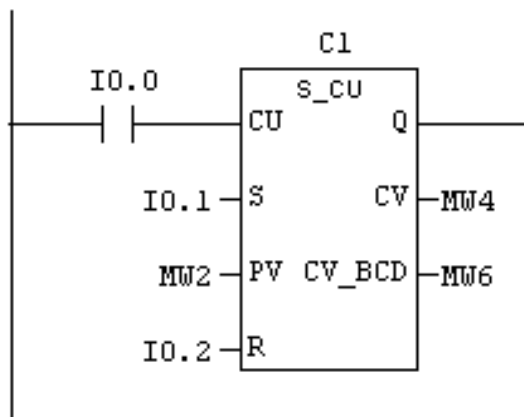
R: tín hiệu xoá (BOOL), khi có sườn lên thì giá trị CV được xoá về 0.

Q : ngõ ra

CV: giá trị hiện tại của bộ đếm dạng Integer

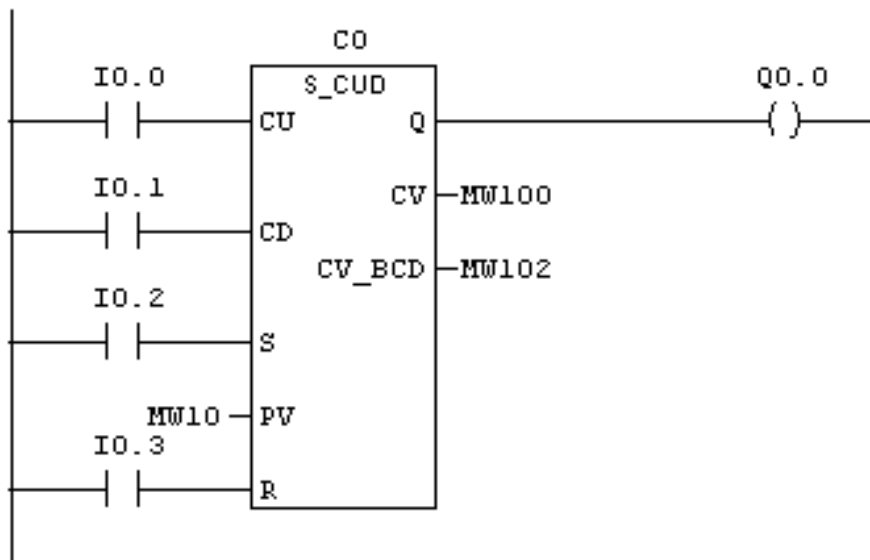
CV_BCD: giá trị hiện tại của bộ đếm dạng BCD

Loại Counter đếm lên



Giống loại Counter trên nhưng không có chân kích đếm xuống.

**Lệnh
đếm lên
xuống
S_CUD**



Ngõ vào I0.2=1 : đưa giá trị đếm vào PV.

Khi I0.0 chuyển trạng thái từ 0 lên 1 ,C0 đếm tăng lên 1.

Khi I0.1 chuyển trạng thái từ 0 lên 1 ,C0 đếm giảm xuống 1.

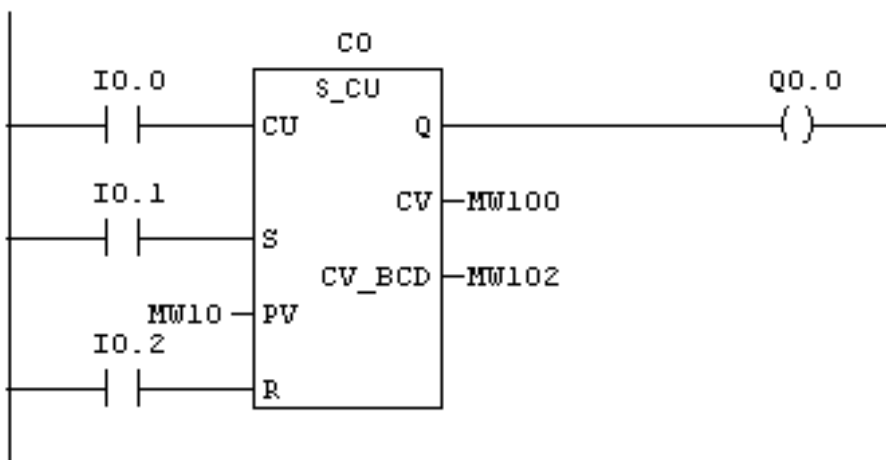
Khi cả I0.0 và I0.1 đều chuyển trạng thái thì C0 không thay đổi.

Khi I0.3=1 thì C0 bị Reset về 0.

Giá trị bộ đếm hiện thời nằm trong 2 ô nhớ MW100 và MW102 dưới dạng Integer và dạng BCD ,giá trị này có tầm từ 0 – 999.

Ngõ ra Q0.0=1 khi giá trị đếm lớn hơn 0.

**Lệnh
đếm lên
S_CU**



Ngõ vào I0.1=1 : đưa giá trị đếm vào PV.

Khi I0.0 chuyển trạng thái từ 0 sang 1 , C0 đếm tăng lên 1.

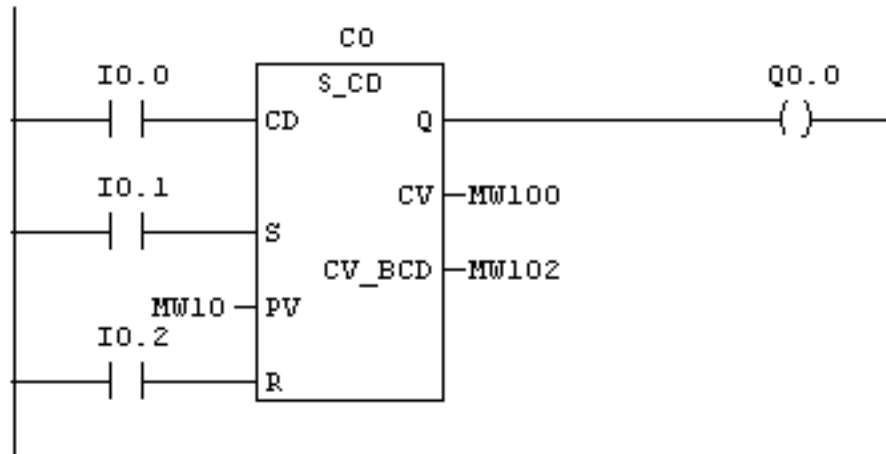
Khi I0.2 = 1 Counter bị Reset.

Ngõ ra Q0.0=1 khi giá trị đếm lớn hơn 0.

Giá trị bộ đếm hiện thời nằm trong 2 ô nhớ MW100 và MW102 dưới dạng Integer và dạng BCD , giá trị này có tầm từ 0 – 999.

Ngõ ra Q0.0=1 khi giá trị đếm lớn hơn 0.

**Lệnh
đếm
xuống
S_CD**



Ngõ vào I0.1=1 : đưa giá trị đếm vào PV.

Khi I0.0 chuyển trạng thái từ 1 sang 0 , CO giảm đi 1.

Khi I0.2 = 1 Counter bị Reset.

Ngõ ra Q0.0=1 khi giá trị đếm lớn hơn 0.

Giá trị bộ đếm hiện thời nằm trong 2 ô nhớ MW100 và MW102 dưới dạng Integer và dạng BCD , giá trị này có tầm từ 0 – 999.

Ngõ ra Q0.0=1 khi giá trị đếm lớn hơn 0.

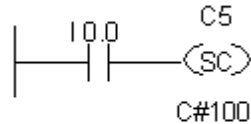
Set Counter

Lệnh Set Counter: (SC)

<C no.>
 ---(SC)
 <preset value>

<u>Parameter English</u>	<u>Parameter German</u>	<u>Data Type</u>	<u>Memory Area</u>	
<C no.>	<Z no.>	COUNTER	C	Số hiệu Counter
<preset value>	<preset value>	WORD	I, Q, M, L, D or constant	Giá trị đặt cho Counter

S7_300 có 1000 counter (từ C0 đến C999)



Mô tả:

Khi I0.0 ON, giá trị 100 sẽ được nạp cho Counter C5.

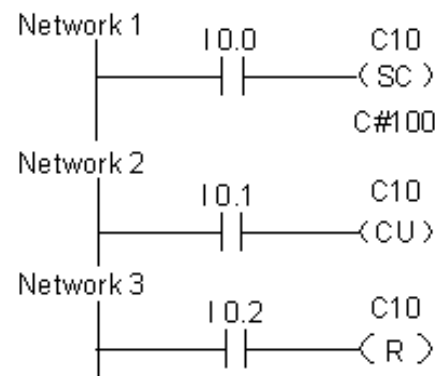
Khi I0.0 OFF, giá trị Counter sẽ phụ thuộc vào các tín hiệu kích đếm.

Đếm lên

Lệnh đếm lên: (CU)

<C no.>
 ---(CU)

<u>Parameter English</u>	<u>Parameter German</u>	<u>Data Type</u>	<u>Memory Area</u>	
<C no.>	<Z no.>	COUNTER	C	Số hiệu Counter



Mô tả:

Khi I0.0 chuyển từ 0 lên 1, giá trị 100 sẽ được nạp vào cho Counter C10.

Cứ mỗi xung cạnh lên ở ngõ vào I0.1, bộ đếm C10 sẽ tăng 1 đơn vị. Khi giá trị tăng đến 999 thì tín hiệu kích tăng không còn tác dụng.

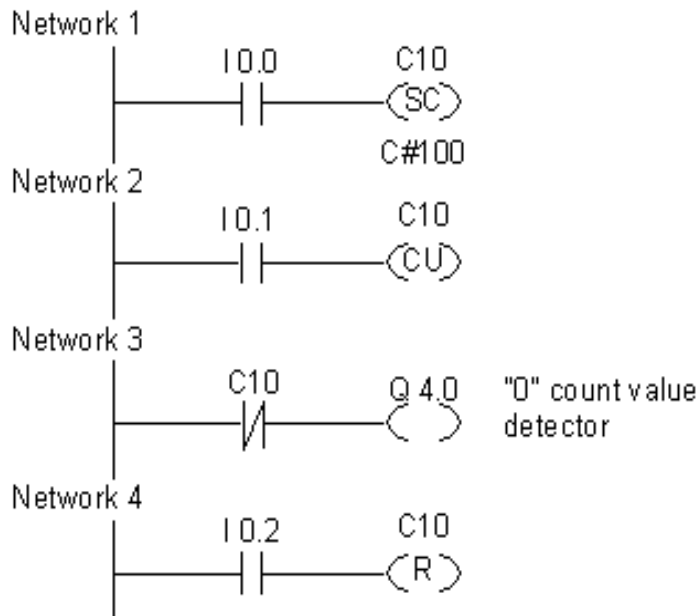
Khi I0.2 chuyển từ 0 lên 1, giá trị bộ đếm và bit C10 sẽ được reset.

Đếm xuống

Lệnh đếm xuống: (CD)

<C no.>
 ---(CD)

<u>Parameter English</u>	<u>Parameter German</u>	<u>Data Type</u>	<u>Memory Area</u>	
<C no.>	<Z no.>	COUNTER	C	Số hiệu Counter



Mô tả:

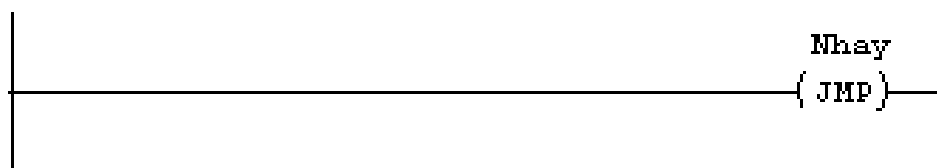
Khi I0.0 chuyển từ 0 lên 1, giá trị 100 sẽ được nạp vào cho Counter C10.

Cứ mỗi xung cạnh lên ở ngõ vào I0.1, bộ đếm C10 sẽ giảm 1 đơn vị. Khi giá trị giảm đến 0 thì tín hiệu kích giảm không còn tác dụng, đồng thời lúc đó C10 sẽ OFF. Nếu bộ đếm khác 0, C10 sẽ ON.

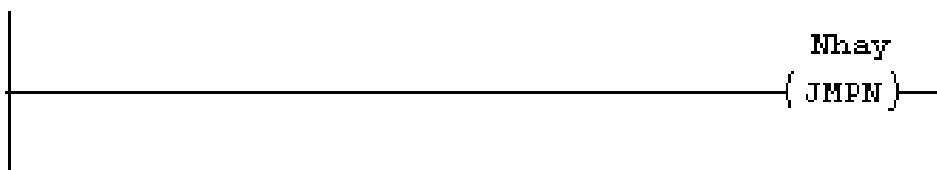
Khi I0.2 chuyển từ 0 lên 1, giá trị bộ đếm và bit C10 sẽ được reset.

2.7 Một số lệnh khác và Bài tập

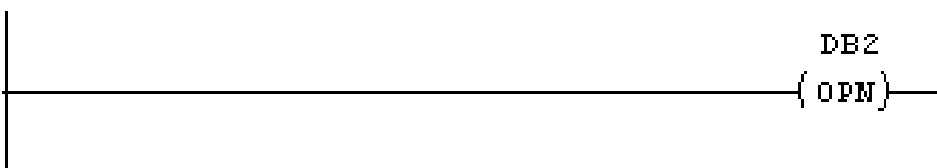
Lệnh JMP: Nhảy nếu RLO=1, Nếu RLO=1 chương trình sẽ nhảy đến nhãn “nhảy”.



Lệnh JMPN: Nhảy nếu RLO=0, Nếu RLO=0 chương trình sẽ nhảy đến nhãn “nhảy”.



Lệnh OPN : Lệnh mở khối DB để có thể truy cập trực tiếp tới khối này.



Bài tập 1

- Tạo chương trình có tên là Baitap_1
- Tạo chương trình con có tên là DKMT
- Tạo khối dữ liệu DB1: TGChay_thuan : INT
TGChaynguoc : INT
- Nhấn Start : I20.0 Motor chạy thuận sau 10s, Motor chạy ngược, sau 15s Motor dừng, khi đang chạy, nhấn Stop dừng
- Cách 1 : Sử dụng thời gian theo cách trực tiếp
- Cách 2 : Sử dụng thời gian theo biến, với cách 2 thử thay đổi thời gian chạy thuận và chạy ngược
- Chương trình con DKMT luôn được gọi trong chương trình chính .

Bài tập 2

- Tạo chương trình có tên là Baitap_1
- Tạo chương trình con có tên là DKDGT
- Tạo khối dữ liệu có tên là DB2

TGXA :INT

TGVA :INT

TGXB :INT

TGVB :INT

- Nhấn Start hệ thống hoạt động tuần tự XA—VA—XB—VB rồi lặp vòng.
- Khi nhấn Stop ,hệ thống tự động dừng.
- Timer cũng sử dụng theo 2 cách, trực tiếp, gián tiếp.
Chương trình DKDGT cũng luôn được gọi trong chương trình chính.

Bài tập
3

- Tạo chương trình có tên là Baitap_1
- Tạo chương trình con có tên là DKDEN
- Tạo khối DB3 : TGDICH : INT
- Khi nhấn Start, hệ thống đèn sẽ được dịch chuyển qua trái,khi nhấn nút đảo, hệ thống đèn sẽ được điều chỉnh dịch ngược trở lại qua phải,khi nhấn Stop, hệ thống đèn sẽ tự động dừng.
- Thời gian dịch cũng được dùng theo 2 cách.
- Với cách dùng ô nhớ ,thì giá trị thay đổi theo ô nhớ TGDICH.
- Chương trình con DKDEN luôn được gọi trong chương trình chính.

3

Phần mềm Simatic S7 V5.3

Chương này giới thiệu phương pháp cài đặt phần mềm Simatic S7 V5.3, cách thức tạo mới một chương trình điều khiển trên Step 7, phân vùng nhớ cho PLC S7-300, kết nối mạng.

3.1 Cài đặt Simatic S7 V5.3

3.2 Cấu hình, tạo mới chương trình điều khiển

3.3 Các vùng nhớ của PLC S7 - 300

3.4 Kết nối mạng

3.1 Cài đặt Simatic S7 V5.3



Cấu hình phần cứng

Để cài đặt STEP 7 yêu cầu tối thiểu cấu hình như sau:

- 80486 hay cao hơn, đề nghị Pentium
- Đĩa cứng trống: Tối thiểu 300MB
- RAM: > 32MB, đề nghị 64MB
- Giao tiếp: CP5611, MPI card hay tiếp hợp PC để lập trình với mạch nhớ
- Hệ điều hành: Windows 95/ 98/ NT/XP/WIN7

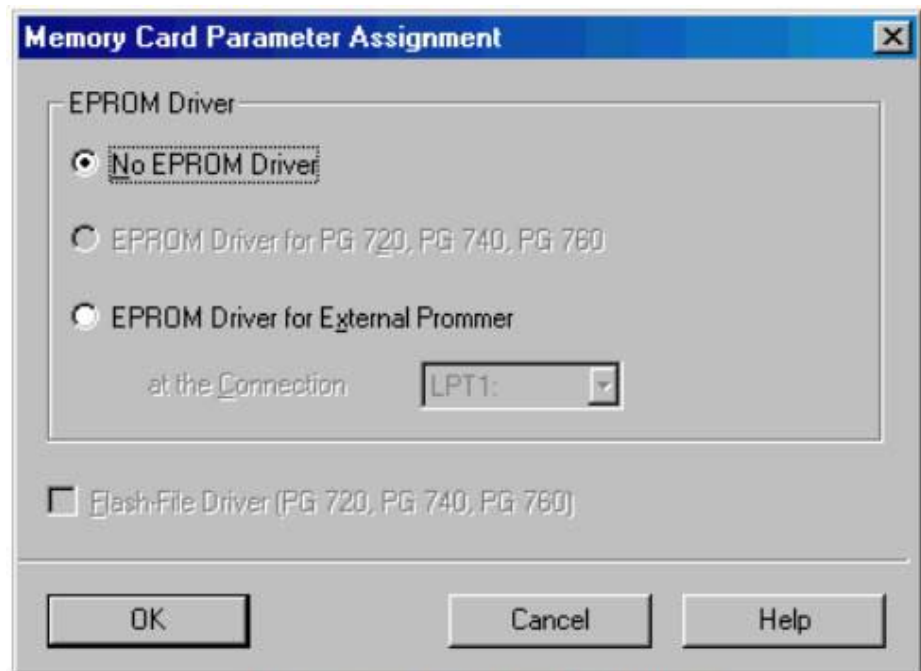
Có nhiều phiên bản của bộ phần mềm gốc của STEP 7 hiện có tại Việt Nam. Đang được sử dụng nhiều nhất là phiên bản 4.x và 5.x. Trong khi phiên bản 4.x khá phù hợp với những PC có cấu hình trung bình nhưng lại đòi hỏi phải tuyệt đối có bản quyền thì phiên bản 5.x, đòi hỏi cấu hình PC phải mạnh tốc độ cao, có thể chạy ở chế độ không cài bản quyền (ở mức hạn chế).

- Chạy file Step7 V5.3\Automation License Manager\disk1\Setup
- Để cài đặt File quản lí bản quyền
- Chạy File Step7 V5.3\STEP7\Disk1\Setup. Để cài đặt phần mềm STEP7
- Chọn ngôn ngữ cho cài đặt
- Chọn việc di chuyển bản quyền cho cài đặt

Phần lớn các đĩa gốc của STEP 7 đều có khả năng tự thực hiện chương trình cài đặt (autorun). Bởi vậy ta chỉ cần bỏ đĩa vào và thực hiện theo những chỉ dẫn. Ta cũng có thể chủ động thực hiện cài đặt bằng cách gọi chương trình setup.exe có trên đĩa. Công việc cài đặt STEP 7 nói chung không khác gì nhiều so với việc cài đặt các phần mềm ứng dụng khác như Windows, Office....

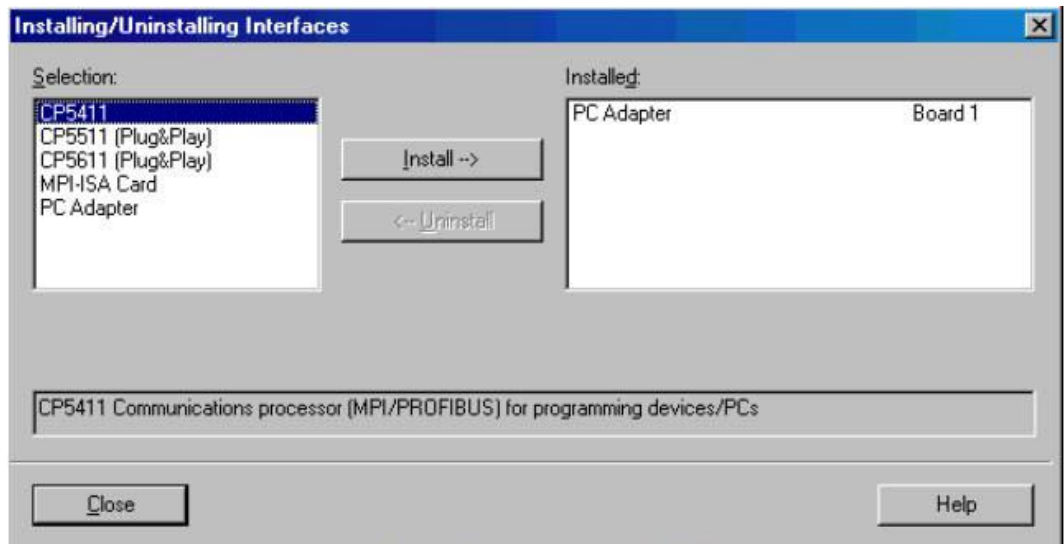
Tuy nhiên, so với các phần mềm khác thì việc cài đặt STEP 7 sẽ có vài điểm khác biệt cần được giải thích rõ thêm:

- Khai báo mã hiệu sản phẩm: Mã hiệu sản phẩm luôn đi kèm theo phần mềm STEP 7 và in ngay trên đĩa chứa bộ cài STEP 7. Khi trên màn hình hiện ra cửa sổ yêu cầu cho biết mã hiệu sản phẩm, ta điền đầy đủ vào tất cả các mục trong ô cửa sổ đó thì mới có thể tiếp tục cài đặt phần mềm.
- Đăng ký bản quyền: Bản quyền của STEP 7 nằm trên một đĩa mềm riêng (thường có màu vàng hoặc đỏ). Ta có thể cài đặt bản quyền trong quá trình cài đặt hay sau khi cài đặt phần mềm xong thì chạy chương trình đăng ký AuthorsW.exe có trên đĩa CD cài đặt.
- Khai báo thiết bị đốt EPROM: Chương trình STEP 7 có khả năng đốt chương trình ứng dụng lên thẻ EPROM cho PLC. Nếu máy tính của ta có thiết bị đốt EPROM thì cần thông báo cho STEP 7 biết khi trên màn hình xuất hiện cửa sổ (hình dưới):



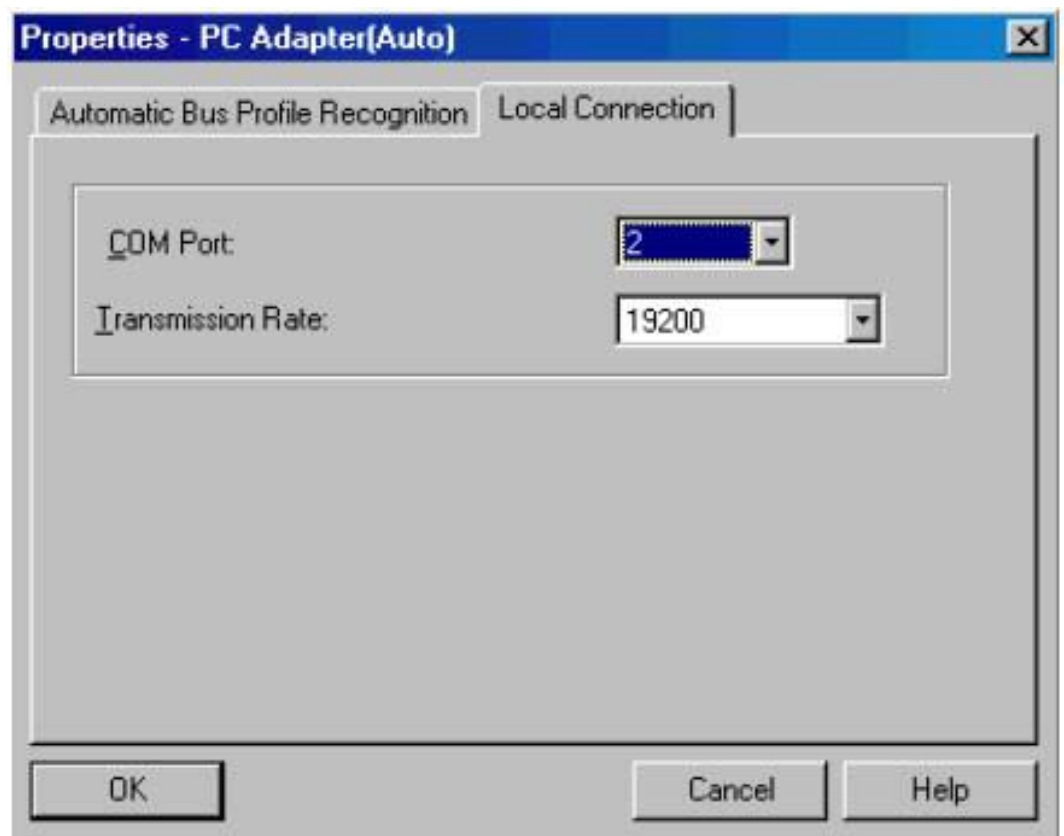
Hình 5.1 Cài đặt thiết bị đốt EPROM

Chọn giao diện PC/PLC: Chương trình được cài đặt trên PG/PC để hỗ trợ việc soạn thảo cấu hình phần cứng cũng như chương trình cho PLC. Ngoài ra, STEP 7 còn có khả năng quan sát việc thực hiện chương trình của PLC. Muốn như vậy ta cần tạo bộ giao diện ghép nối giữa PC và PLC để truyền thông tin, dữ liệu. STEP 7 có thể được ghép nối giữa PC và PLC qua nhiều bộ giao diện khác nhau và ta có thể chọn giao diện sẽ được sử dụng trong cửa sổ sau:



Hình 5.2 Các bộ giao diện có thể chọn

Sau khi chọn bộ giao diện ta phải cài đặt tham số làm việc cho nó thông qua cửa sổ màn hình dưới đây khi chọn mục “Set PG/PC Interface...”.



Hình 5.3 Cài đặt thông số cho bộ giao diện

Đặt tham số làm việc:

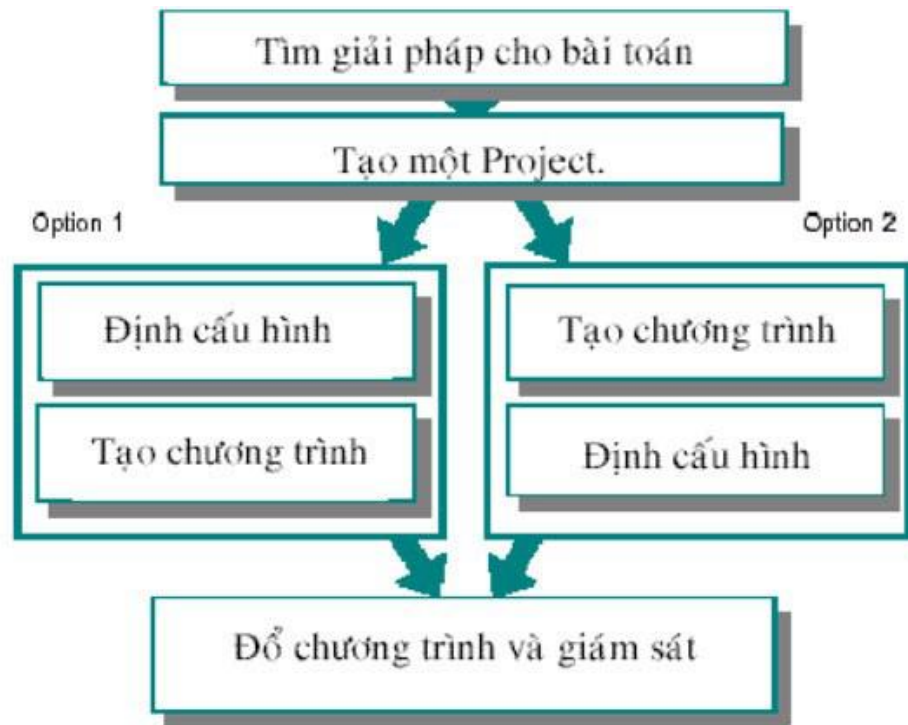
Sau khi cài đặt xong STEP 7, trên màn hình desktop sẽ xuất hiện biểu tượng của phần mềm STEP 7.



Hình 5.4 Biểu tượng của STEP 7

Đồng thời trong menu Start của Windows cũng có thư mục Simatic với tất cả các tên của những thành phần liên quan, từ các phần mềm trợ giúp đến các phần mềm cài đặt cấu hình, chế độ làm việc của STEP 7

3.2 Cấu hình, tạo mới chương trình điều khiển



**Khởi
động
chương
trình
tạo
project**

Chương trình quản lý SIMATIC là giao diện đồ họa với người dùng bằng chương trình soạn thảo trực tuyến/ngoại tuyến đối tượng S7 (đề án, tập tin người dùng, khối, các trạm phần cứng và công cụ).

Với chương trình quản lý SIMATIC có thể:

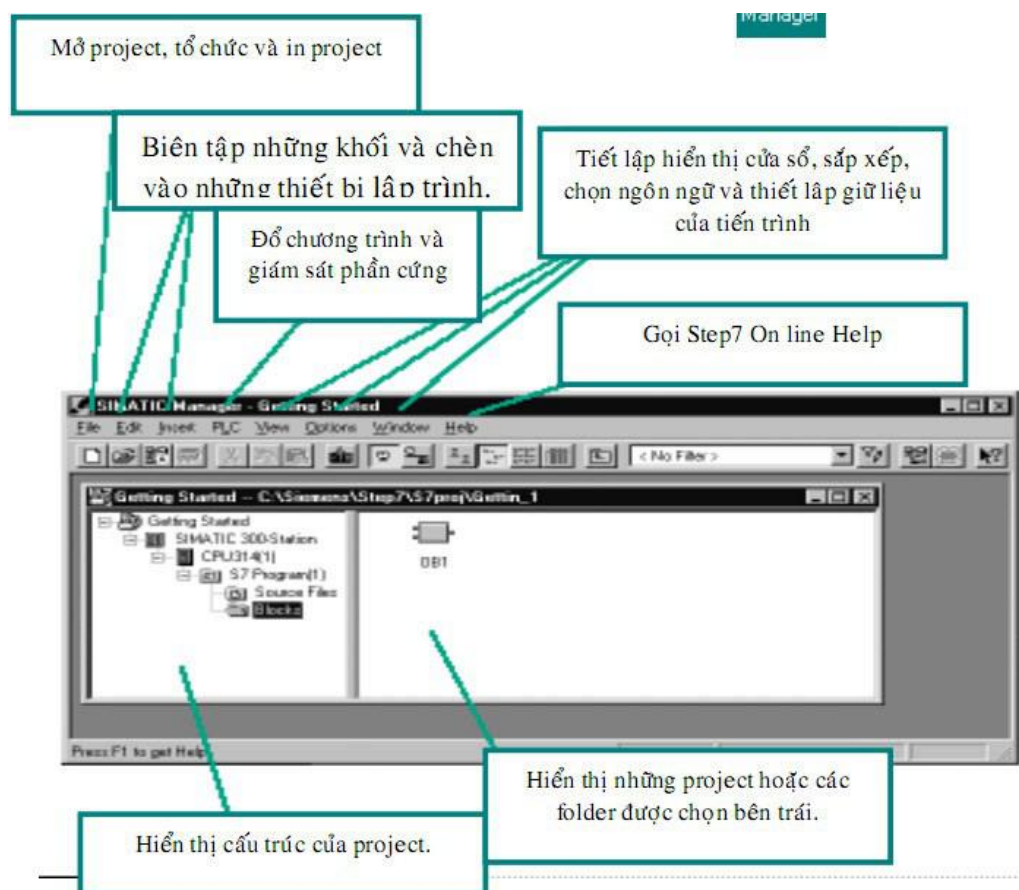
- Quản lý đề án và thư viện.
- Tác động công cụ của STEP 7.
- Truy cập trực tuyến PLC.
- Soạn thảo thẻ nhớ.

Các công cụ của STEP 7 có ở trong SIMATIC Maneger. Để khởi động có thể làm theo hai cách:

- Bằng Task bar → Start → SIMATIC → STEP7 → SIMATIC Maneger.

- Nhấn kép vào biểu tượng SIMATIC Manager,





Hình 5.5 Các thành phần cửa sổ Manager

Thanh tiêu đề:

Thanh tiêu đề gồm cửa sổ và các nút để điều khiển cửa sổ

Thanh thực đơn:

Gồm các thực đơn cho các cửa sổ đang mở.

Thanh công cụ:

Gồm các thao tác thường dùng nhất dưới dạng ký hiệu. Những ký hiệu này có thể tự giải thích.

Thanh trạng thái:

Hiện ra trạng thái hiện tại và nhiều thông tin khác.

Thanh công tác:

Chứa các ứng dụng đang mở và cửa sổ dưới dạng các nút. Thanh công tác có thể đặt 2 bên màn hình bằng cách nhấn chuột phải.

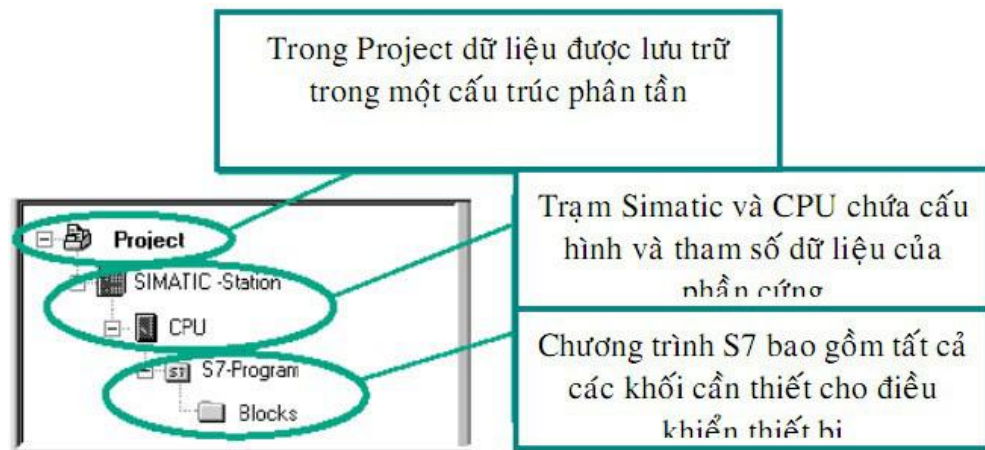
Thanh công cụ chương trình quản lý SIMATIC bao gồm:

New (File Menu)Tạo mới

Open (File Menu)Mở file

Display Accesible Nodes (PLC Menu)Hiển thị các nút
S7 Memory Card (File Menu)Thẻ nhớ S7
Cut (Edit menu)Cắt
Paste (Edit Menu)Dán
Copy (Edit Menu)Sao chép
Download (PLC Menu) Tải xuống
Online (View Menu)Trực tuyến
Offline (View Menu)Ngoại tuyến
Large Icons (View Menu)Biểu tượng lớn
Small Icons (View Menu)Biểu tượng nhỏ
List (View Menu)Liệt kê
Details (View Menu)Chi tiết
Up on level (View Menu)Lên một cấp
Simulate Modules (OptionMenu) Khối mô phỏng
Help SymbolBiểu tượng trợ giúp

Cấu trúc project step7



Hình 5.6 Cấu trúc project step7

Viết chương trình điều khiển

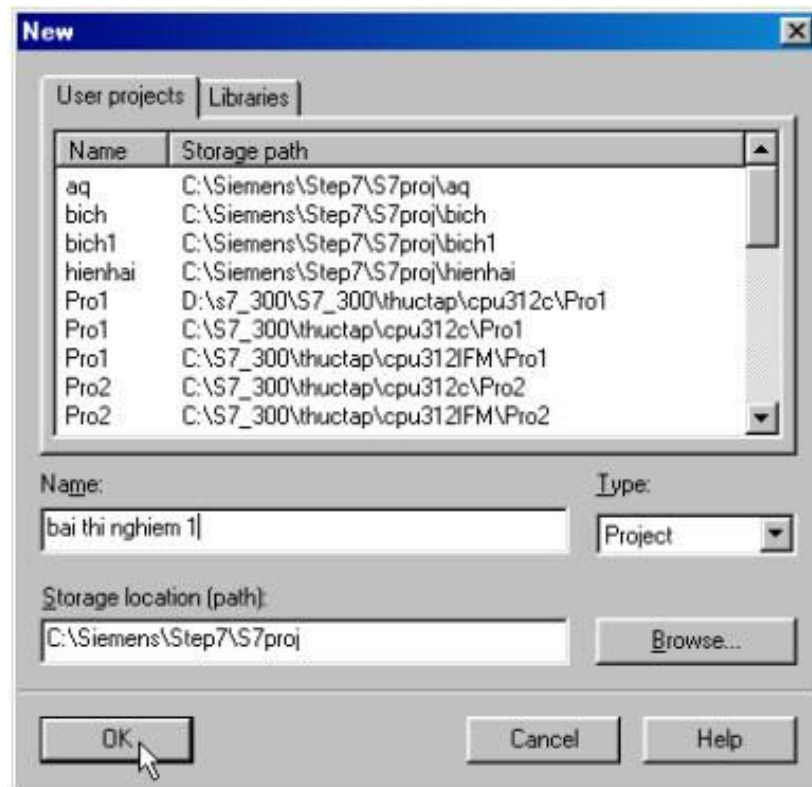
Khai báo phần cứng

Ta phải xây dựng cấu hình phần cứng khi tạo một project. Dữ liệu về cấu hình sẽ được truyền đến PLC sau đó.

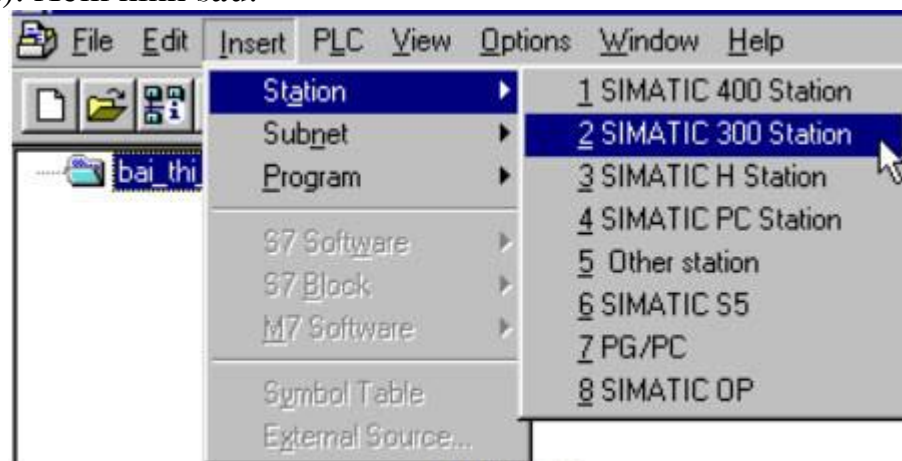
Ta sẽ thử khai báo phần cứng cho các Module sau:

CPU 312C-5BD01-0AB0, DI 321-1BH02-0AA0, DO 322-1HF01-0AA0, AI 331-7KB02-0AB0, AO 332-5H501-0AB0

- ◆ Click vào biểu tượng New để mở chương trình mới. Khi cửa sổ New hiện ra, ta nhập tên của chương trình vào và Click OK như hình sau:



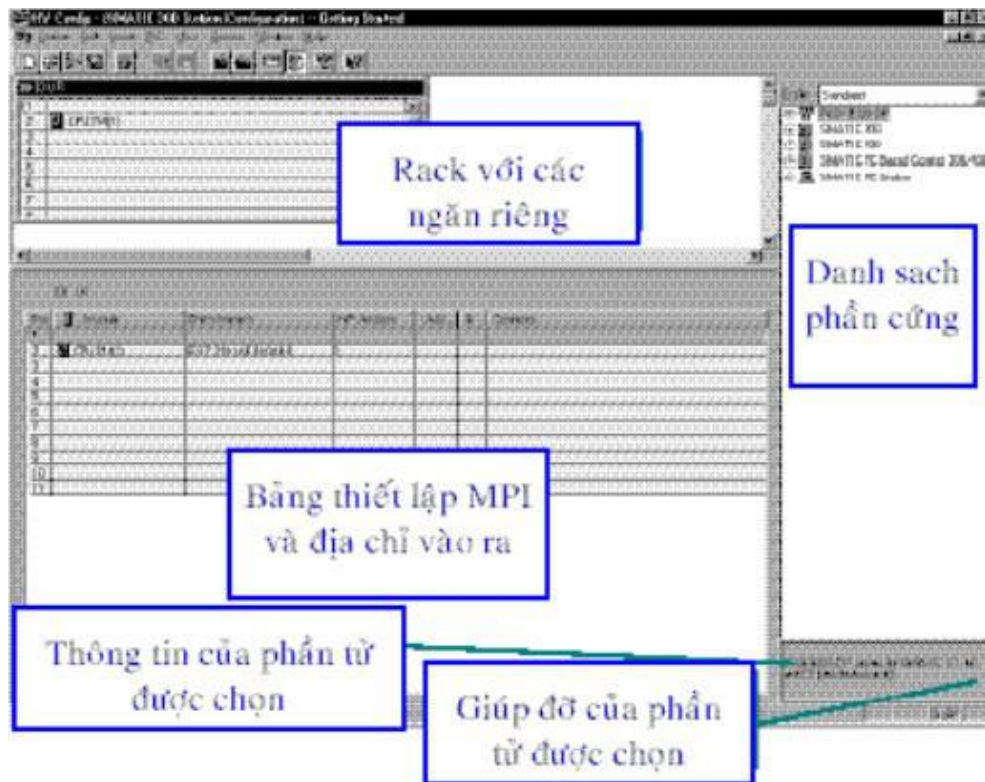
Trở vào màn hình chính ta vào Insert -> Station -> SIMATIC 300 Station để chèn cấu hình cho chương trình (module CPU, module IM,...). Xem hình sau:



Hình 5.8

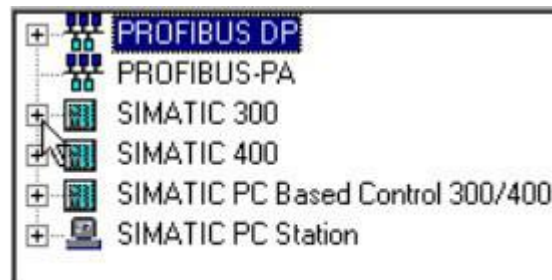
Khi Click vào biểu tượng SIMATIC 300 bên phải màn hình xuất hiện biểu tượng Hardware. Ta Click vào biểu tượng Hardware để khai báo cấu hình cho chương trình.

Cửa sổ HW Config được mở (xem hình dưới), ta phải chèn rack cho project:



Hình 5.9 Cửa sổ khai báo Hardware

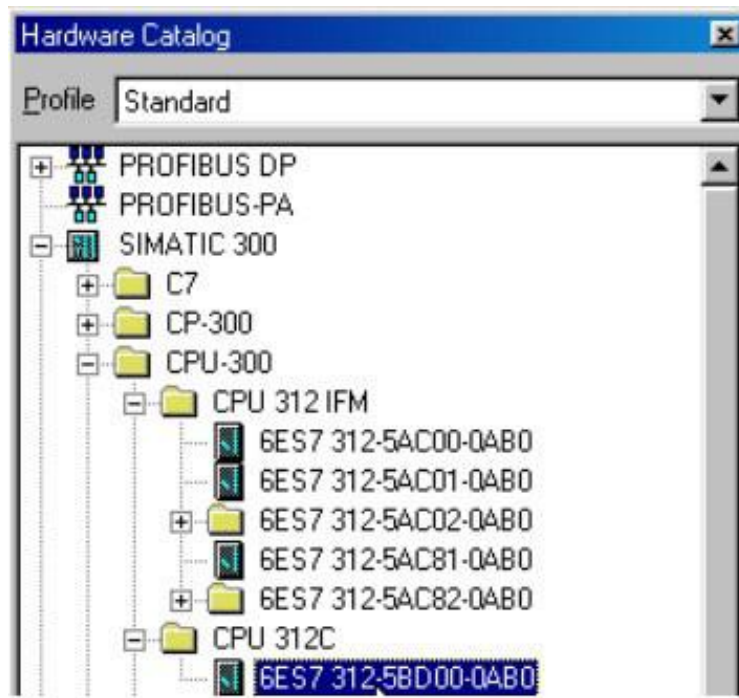
Trong cửa sổ HW_config ta Click vào biểu tượng để mở thư viện. Trong thư viện, ta Click vào SIMATIC 300 (hình 15) để lấy các thành phần cần thiết.



Hình 5.10

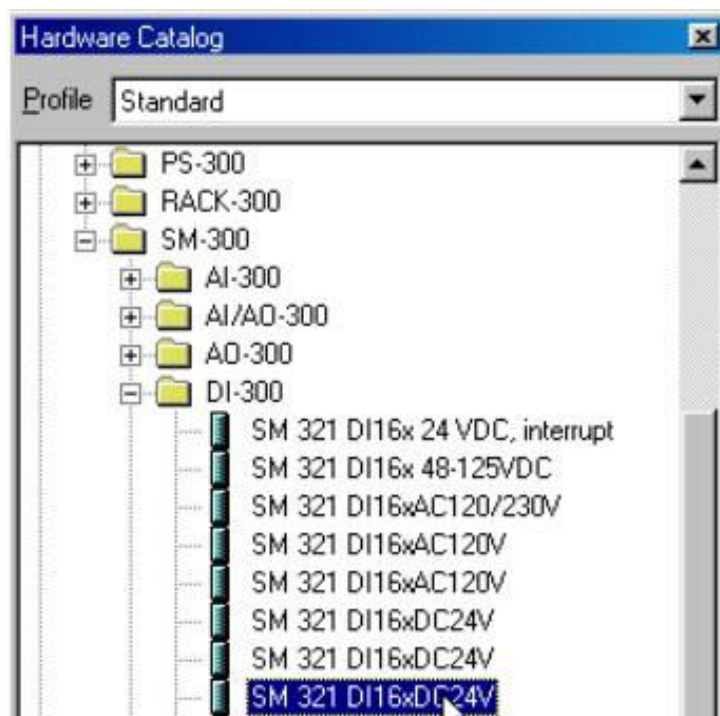
Tiếp theo, ta tìm thư mục RACK 300 và Click vào biểu tượng để tạo Rail chứa các Module.

Một Rail sẽ hiện ra gồm 11 Slot (xem hình). Ta Click vào Slot 2 (tô đậm Slot 2), sau đó Click vào CPU_300 -> CPU_312C -> 6ES7 312-5BD00-0AB0.



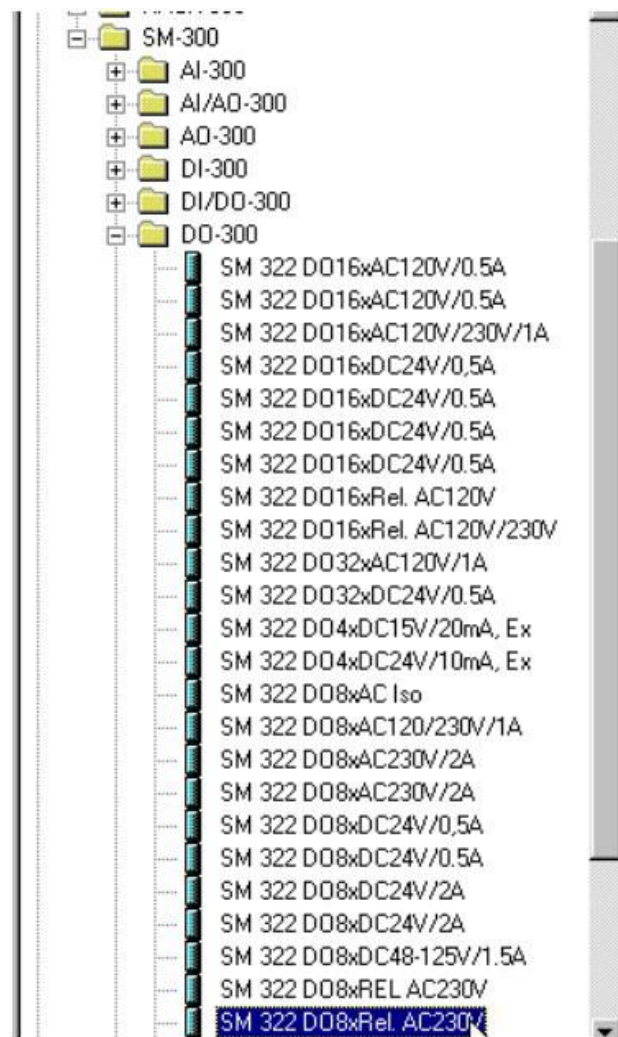
Hình 5.11

Tiếp theo, Click vào Slot 4 và Click vào SM_300 -> DI_300 -> D_Click vào SM 321 DI16×DC24V (hình 5.12). Số hiệu này tùy thuộc loại Module DI mà ta có.



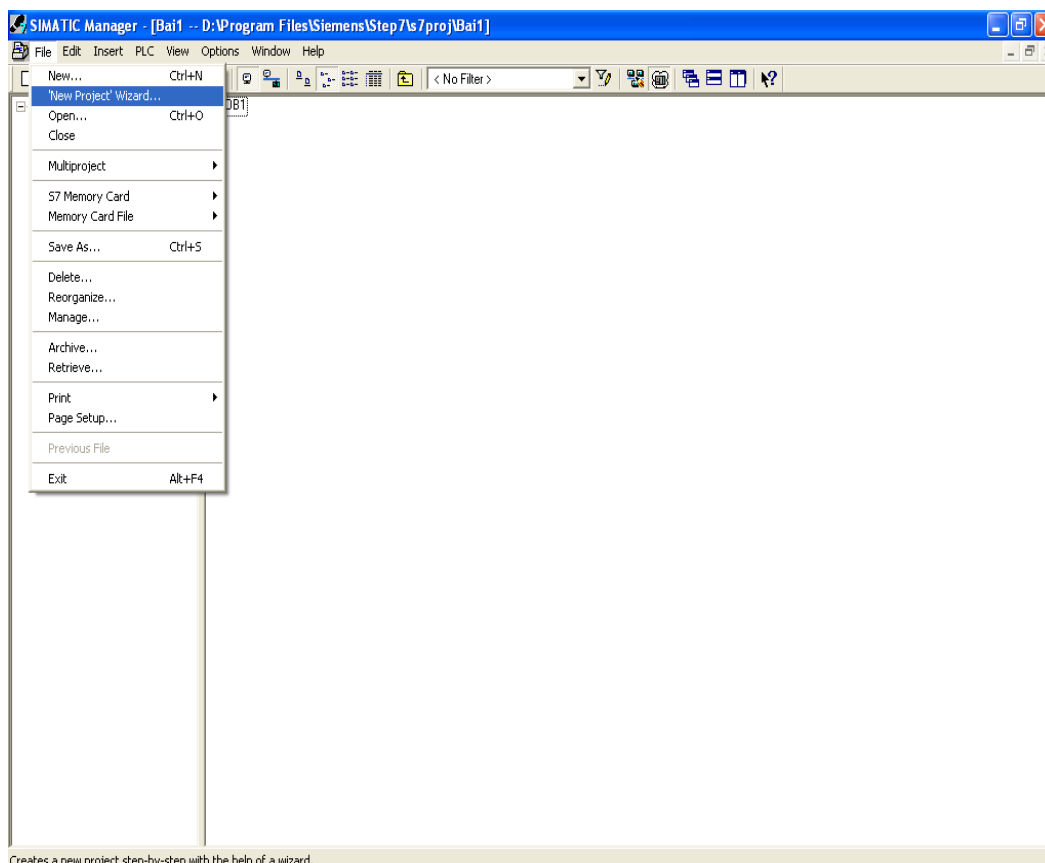
Hình 5.12

Tương tự, ta Click vào Slot 5 và Click vào DO_300 -> D_Click vào SM 322 DO16×DC24V/0.5A. (Hình 5.13):



Hình 5.13

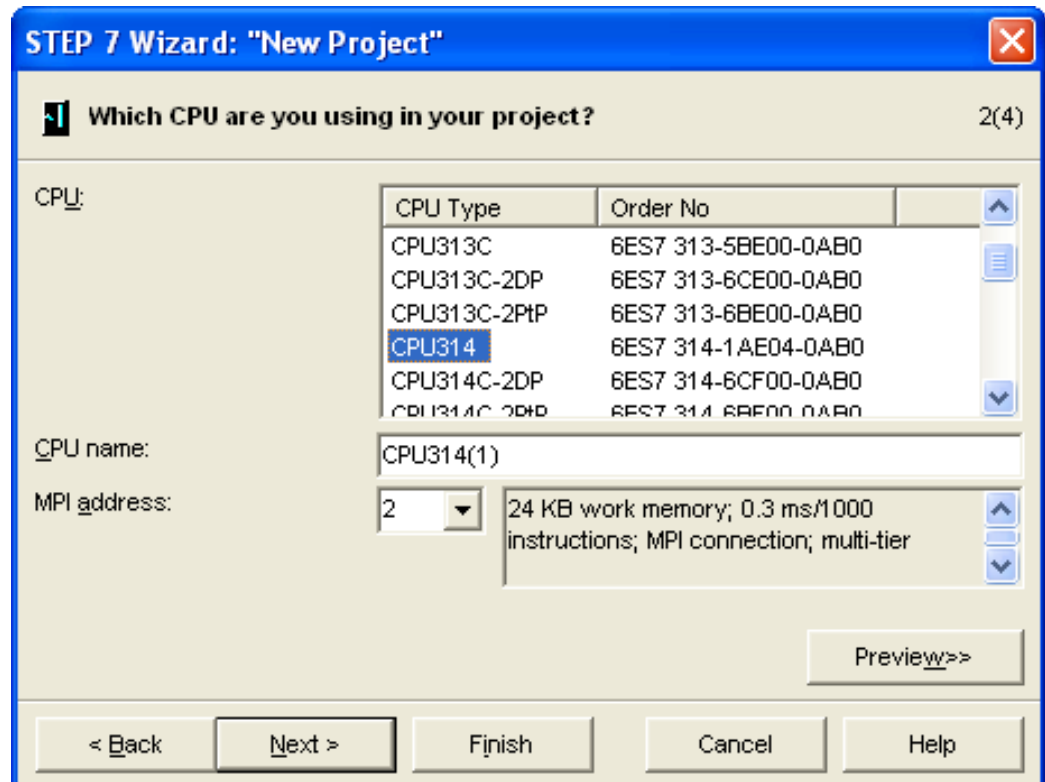
Ví dụ Để tạo Project mới chúng ta có 2 cách :
Dùng “New Project” Wizard và sử dụng “New”
Tạo Project mới bằng “New Project” Wizard
Bạn chọn File->”New Project” Wizard..



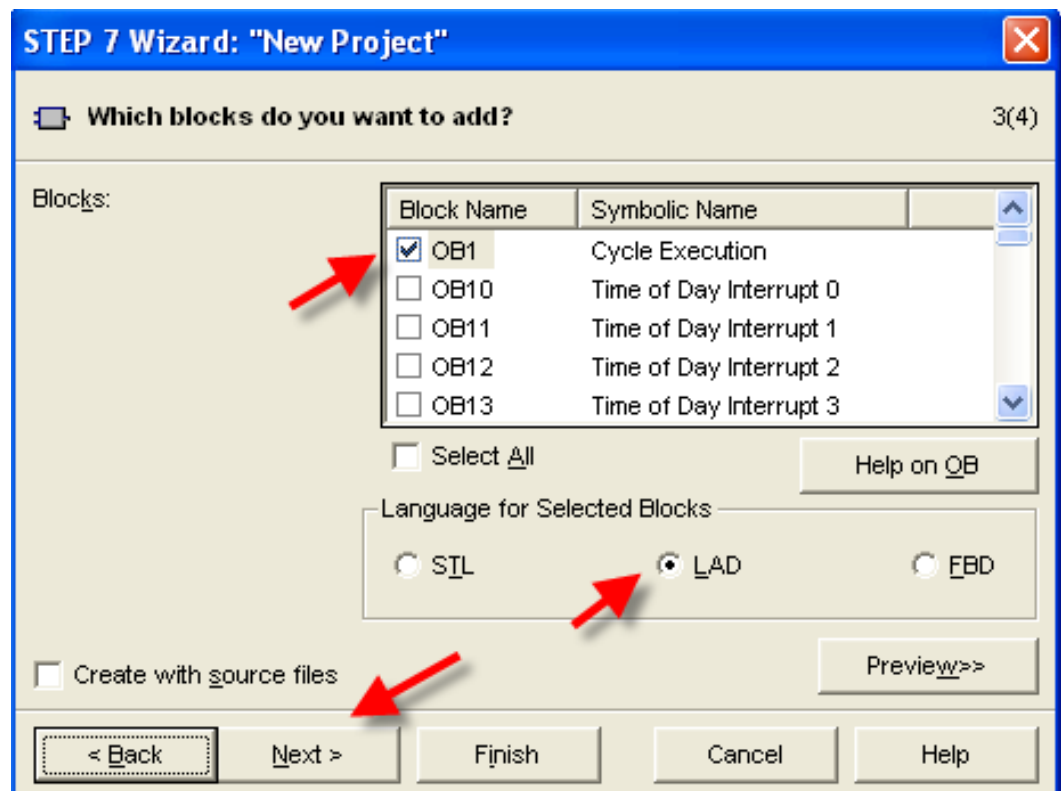
Cửa sổ New Project hiện ra. Click Next để tiếp tục



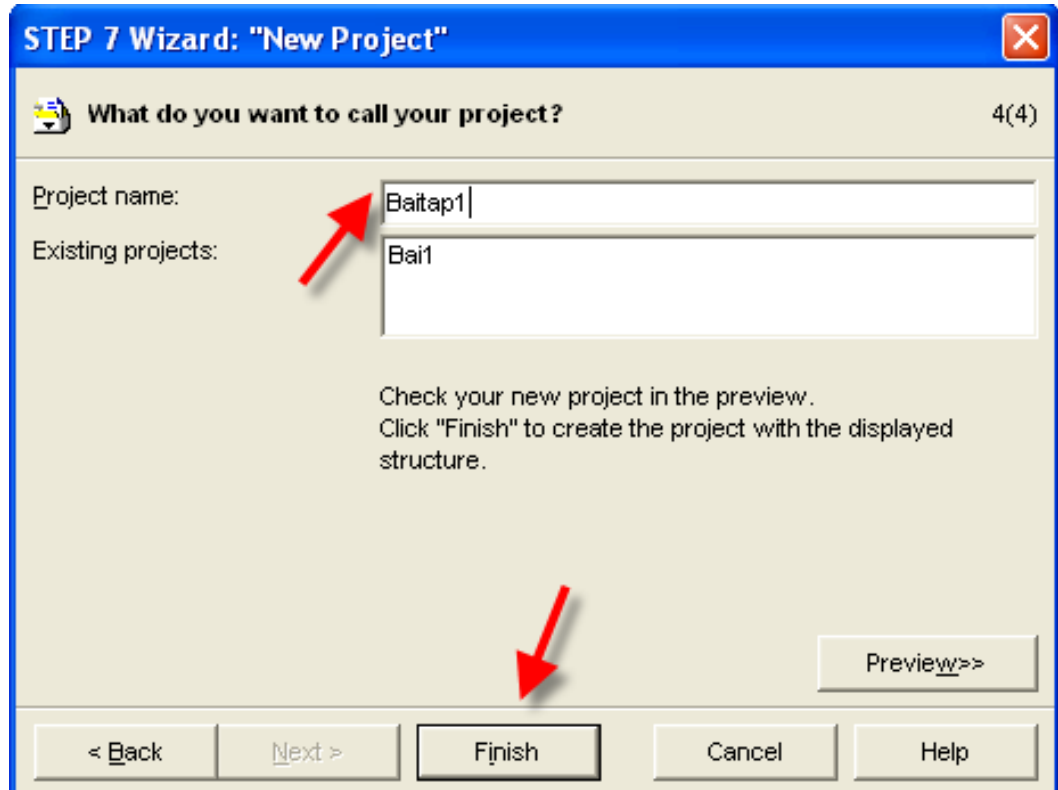
Ở cửa sổ kế bạn chọn loại CPU rồi click Next



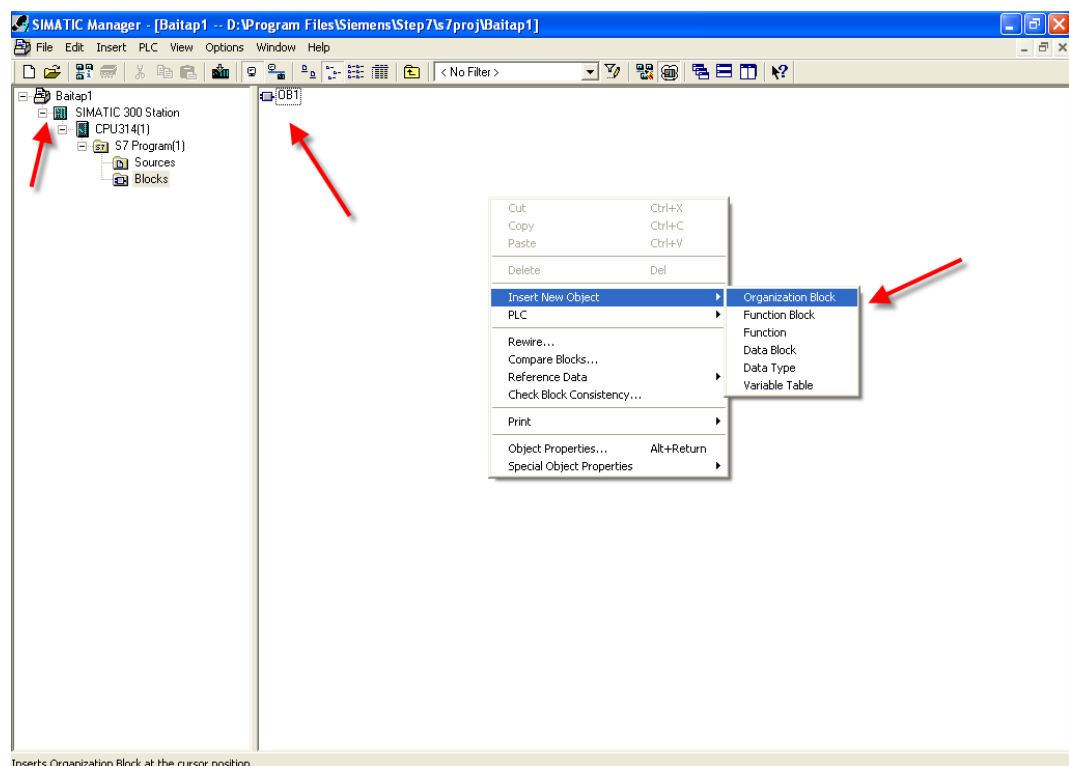
Ở cửa sổ kế tiếp, Chọn các khối OB và ngôn ngữ lập trình. OB1 là khối OB chính giống như hàm Main



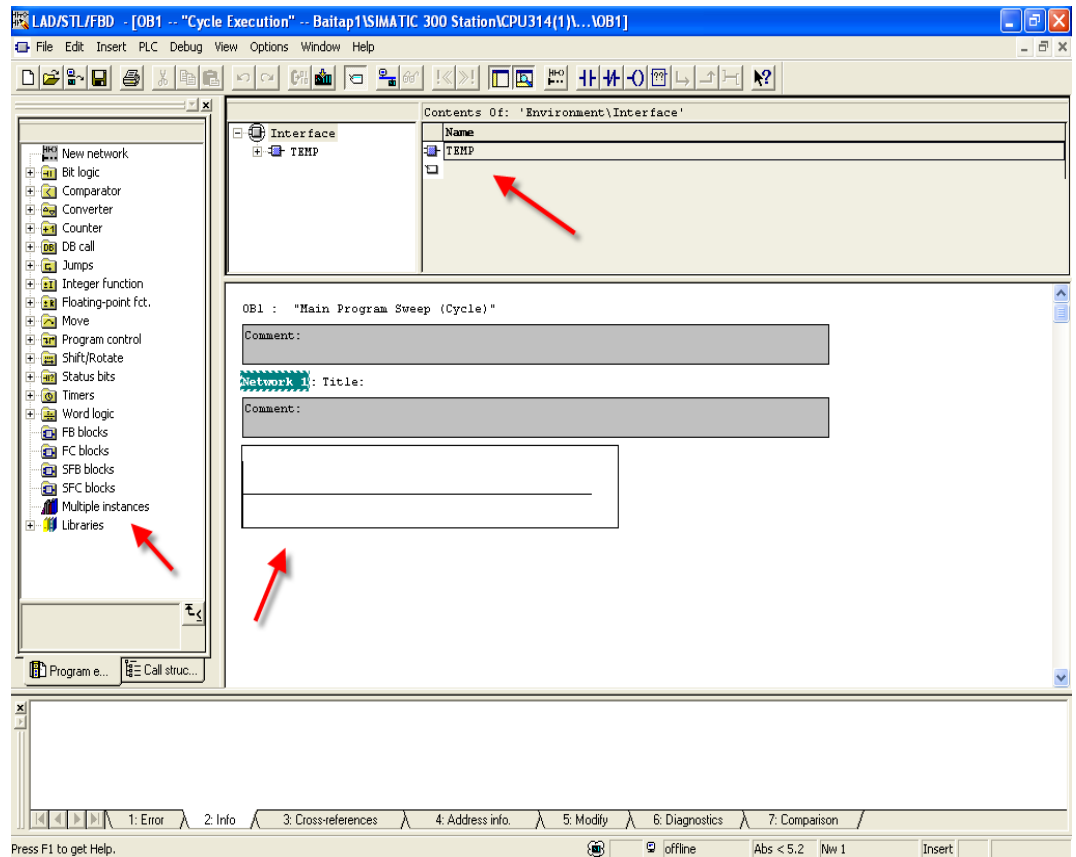
Đặt tên cho New Project cuối cùng chọn Finish



Ở cửa sổ Simatic Manager chia ra 2 phần chính : Phần bên trái là cấu trúc phần cứng, Phần bên phải là hiển thị các khối. Ở phần bên phải chúng ta có thể tạo các khối OB ngắt, Khối FC, FB, DB, VAT. Để lập trình cho khối nào ta double click mở khối đó ra. Ở đây ta lập trình cho khối OB1

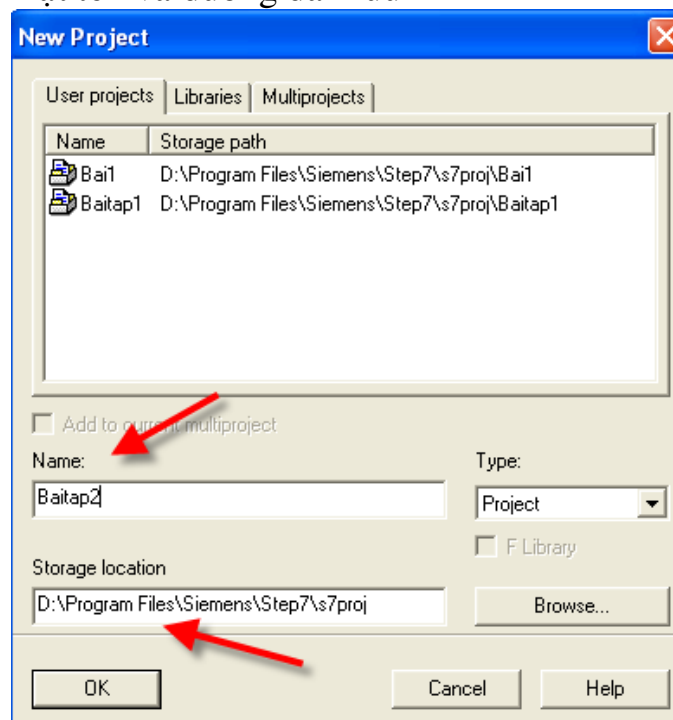


Cửa sổ lập trình cho khối OB1 chia ra 4 phần chính. Chúng ta có thể bắt đầu lập trình cho PLC. Lấy ví dụ 1 chương trình đơn giản là nhấn I0.0 đèn sáng 10s rồi tắt. (Chỉ là chương trình đơn giản để biết cách mô phỏng).

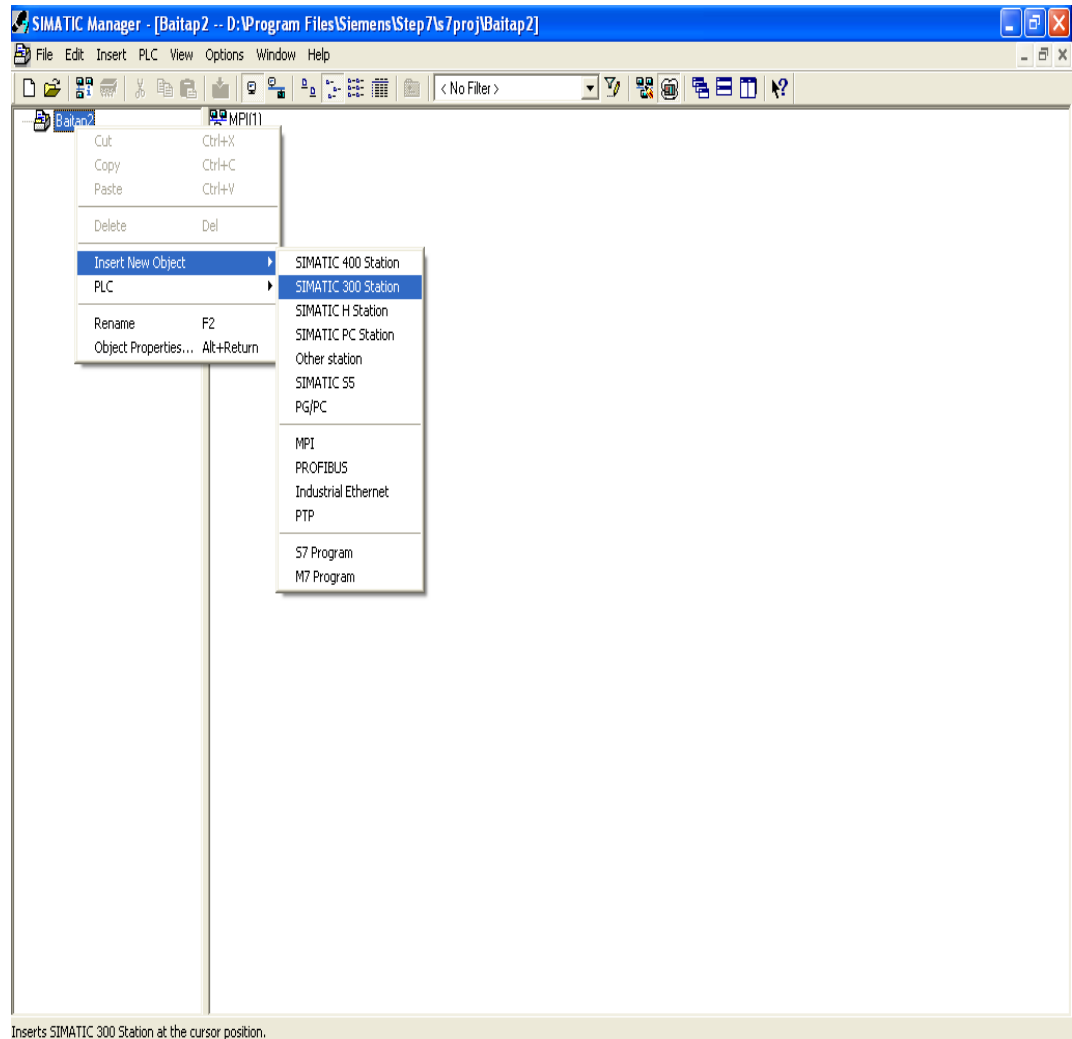


Cách tạo New Project thủ công

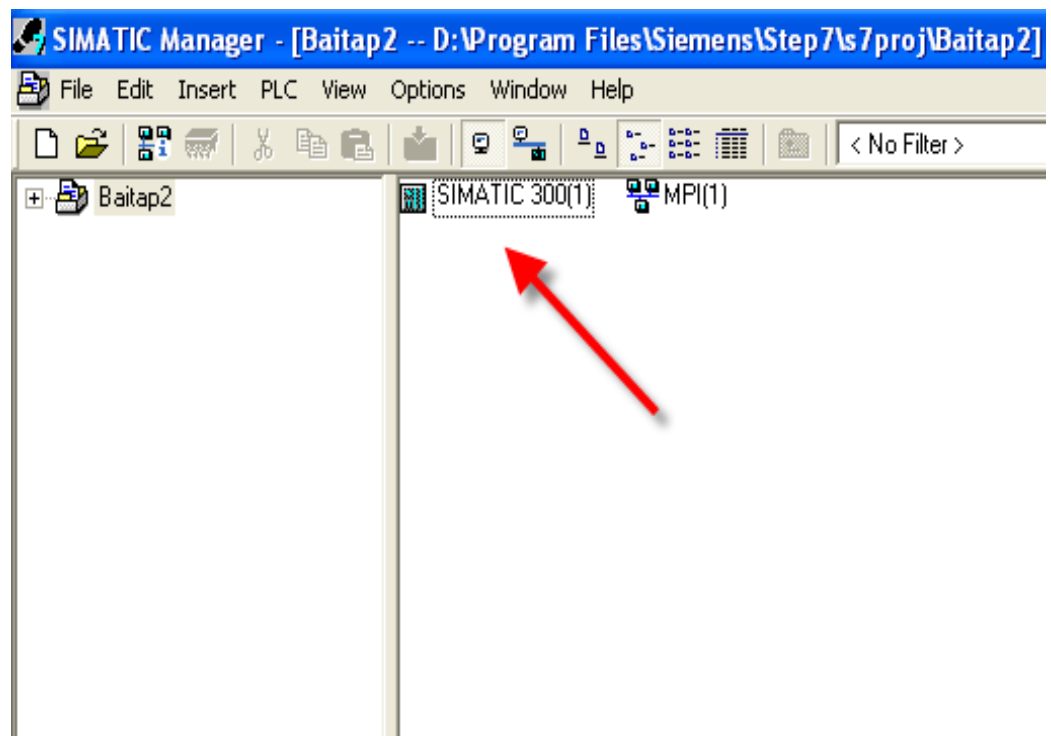
Đặt tên và đường dẫn lưu



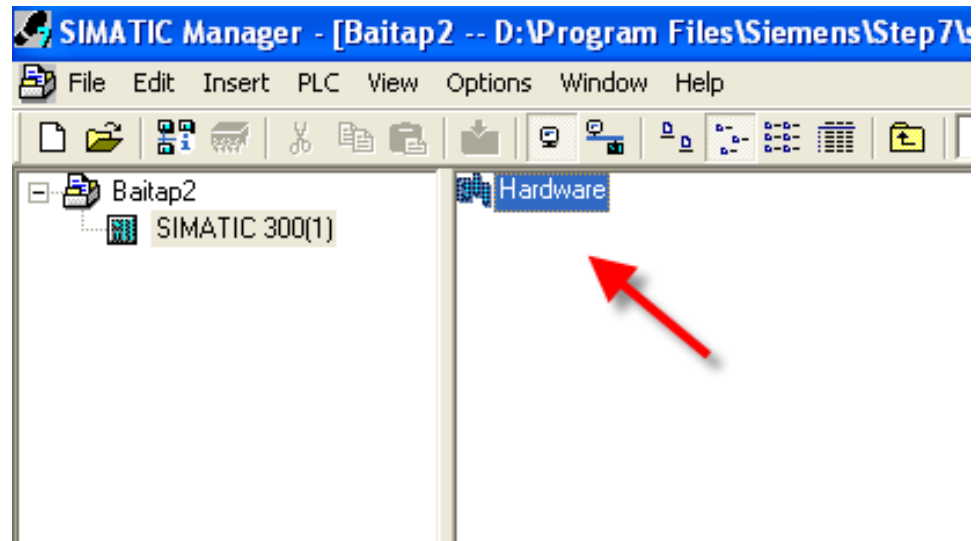
Click chuột phải : Tạo Trạm 300



Double Click vào Simatic300



Double vào Hardware để chọn phần cứng cho PLC



Cửa sổ bên phải để chọn thiết bị :

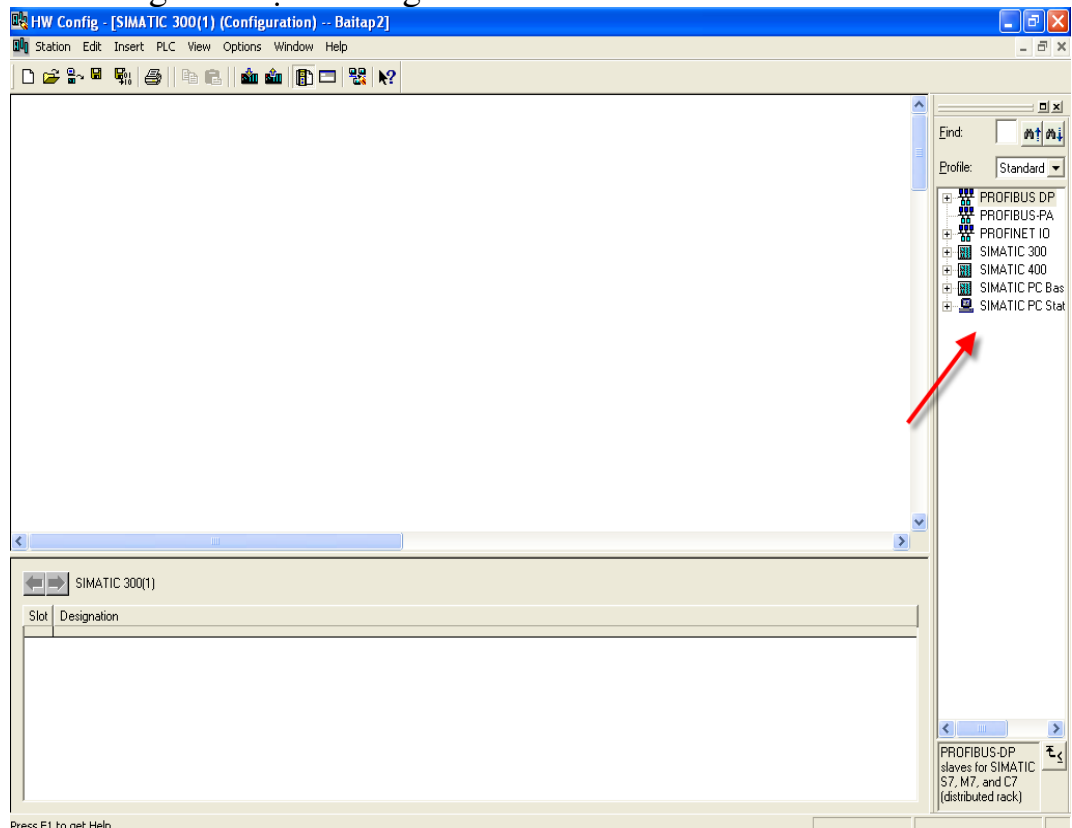
Bước 1 : Chọn Rack

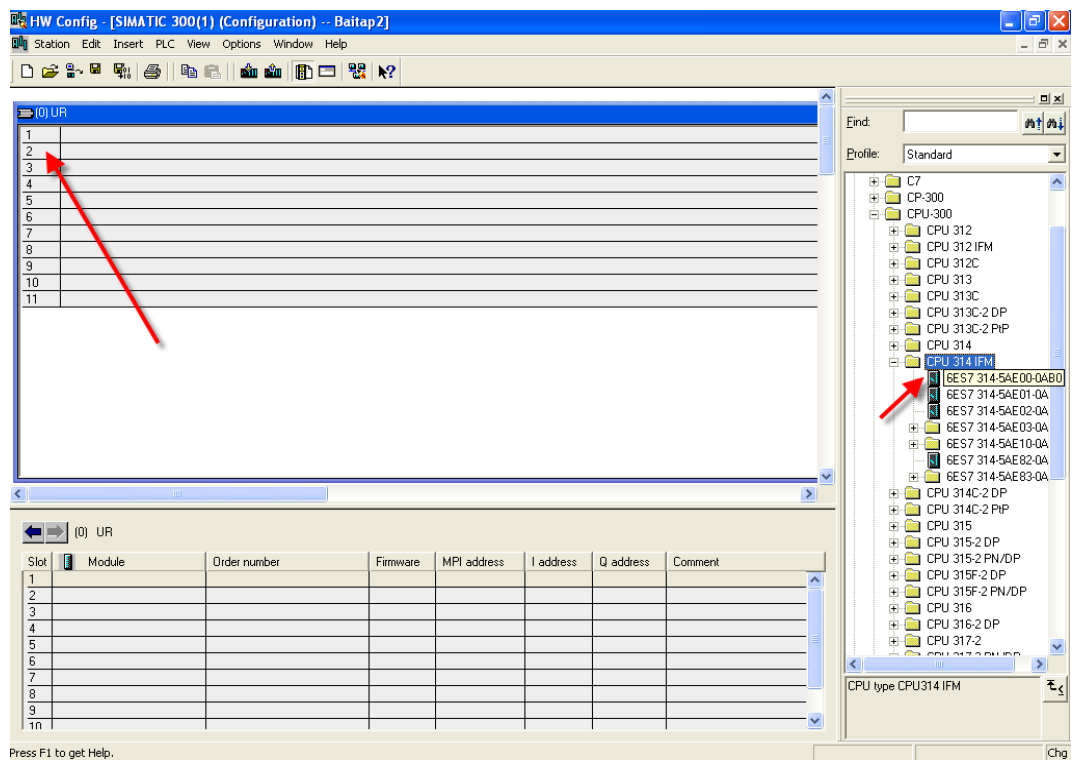
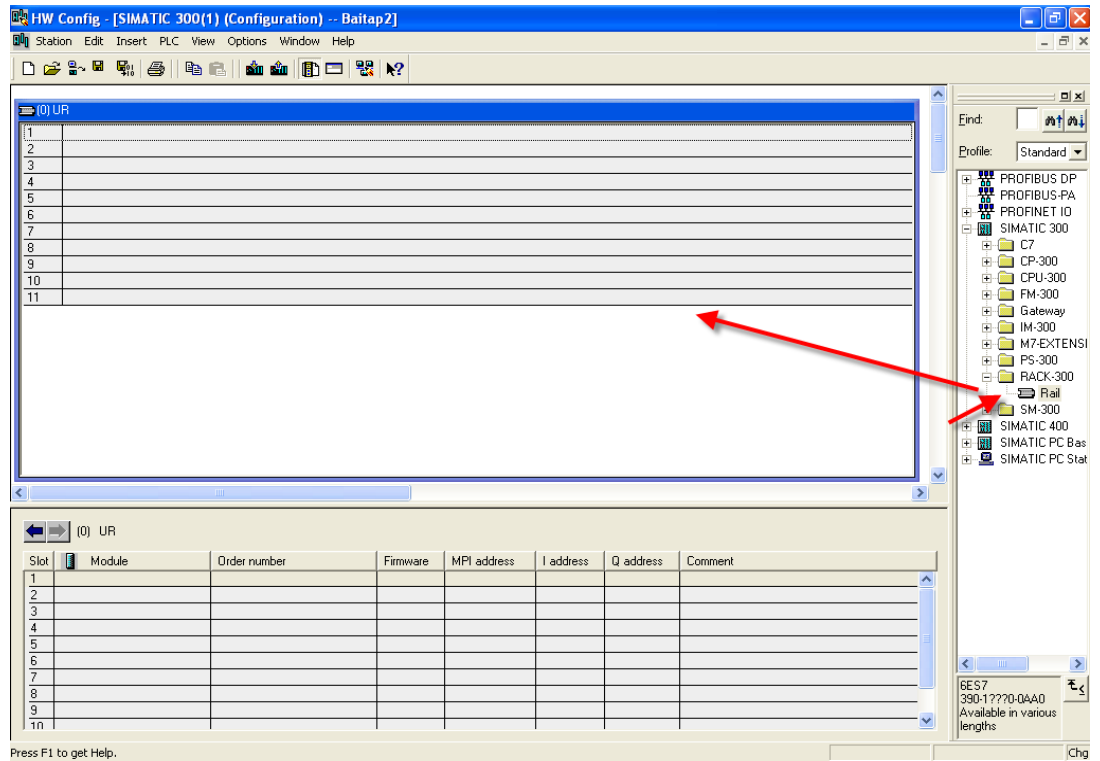
Bước 2 : Chọn CPU (Lưu ý , CPU chỉ để vào Plot 2 của Rack. Mặc định Plot 1 để lắp nguồn).

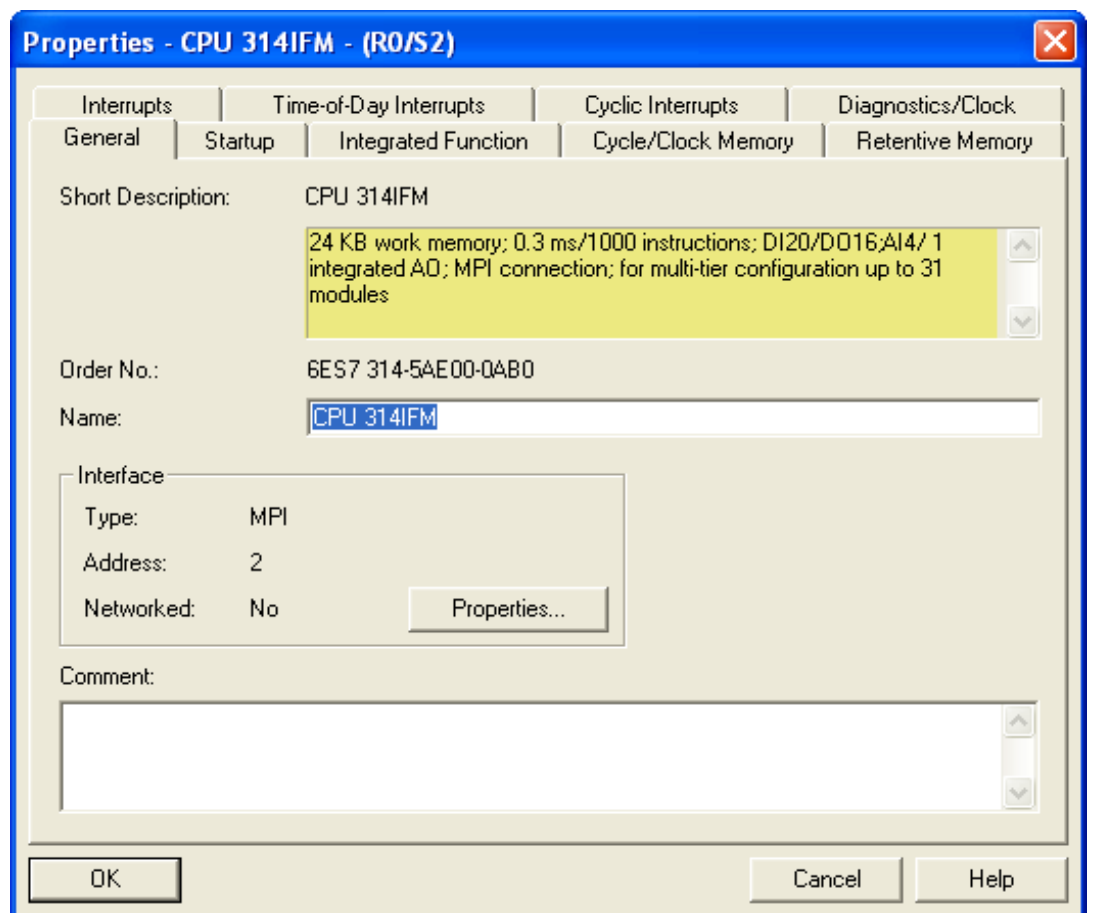
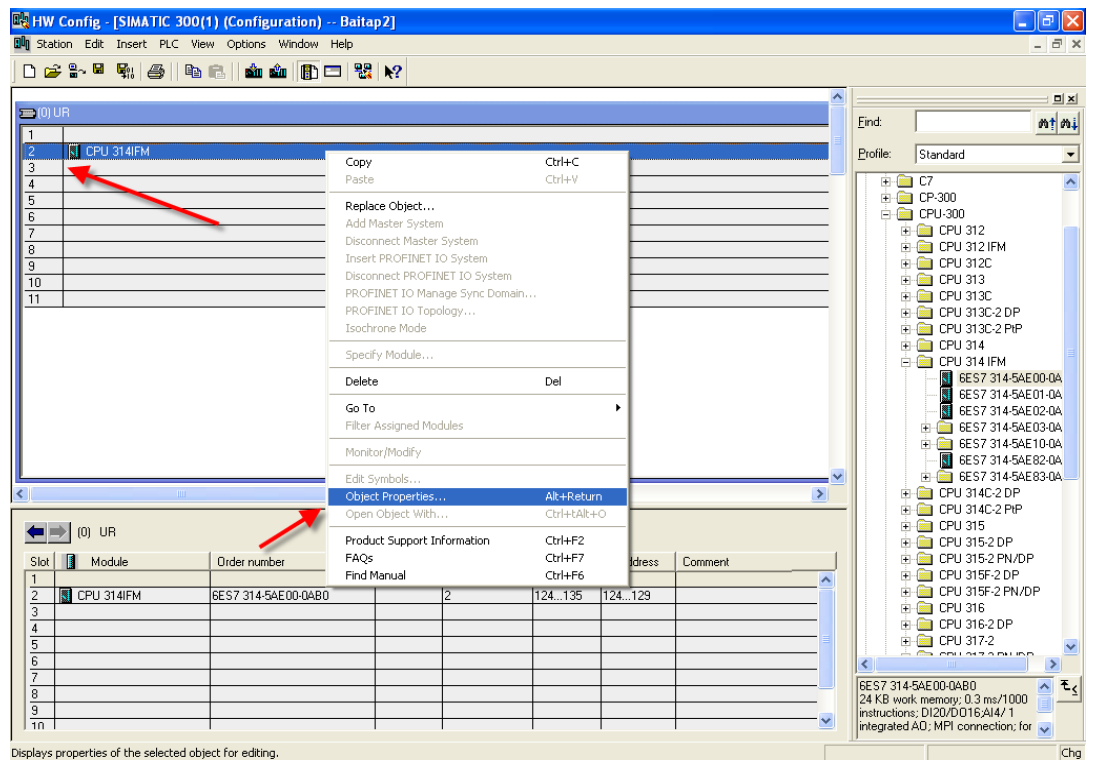
Bước 3: Chọn Modul mở rộng (từ Plot 3 trở đi).

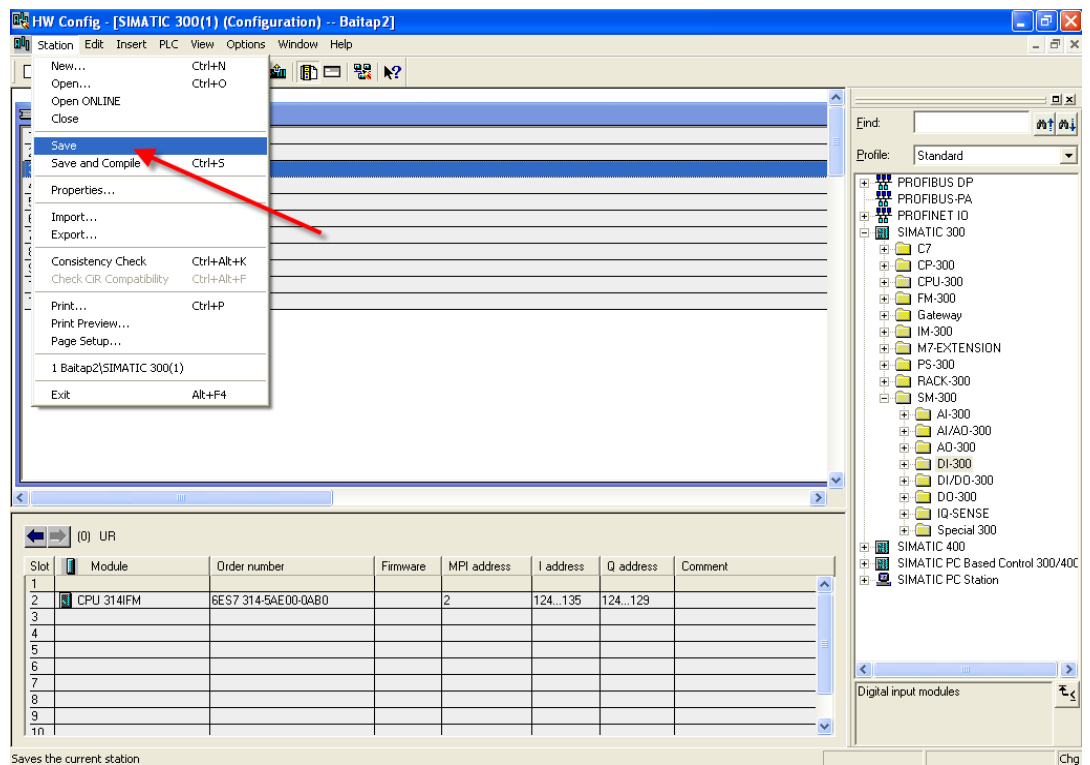
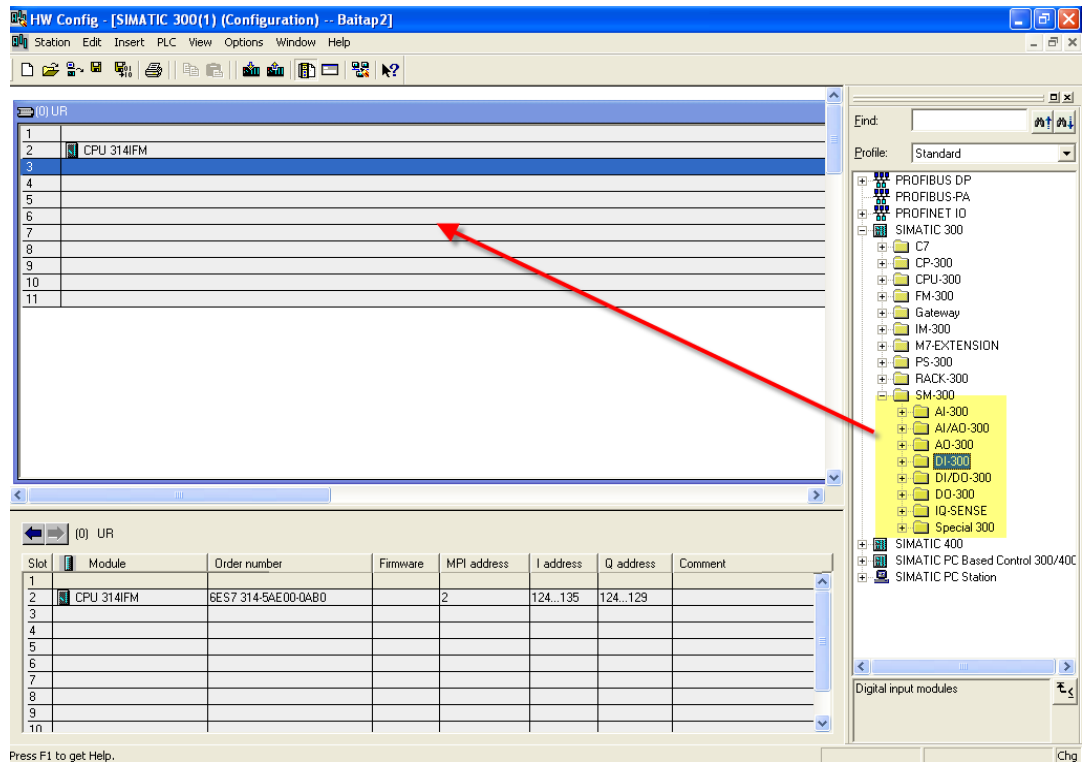
Để biết CPU bạn chọn có những chức năng gì click chuột phải vào CPU và chọn Properties.

Cuối cùng save lại là xong.



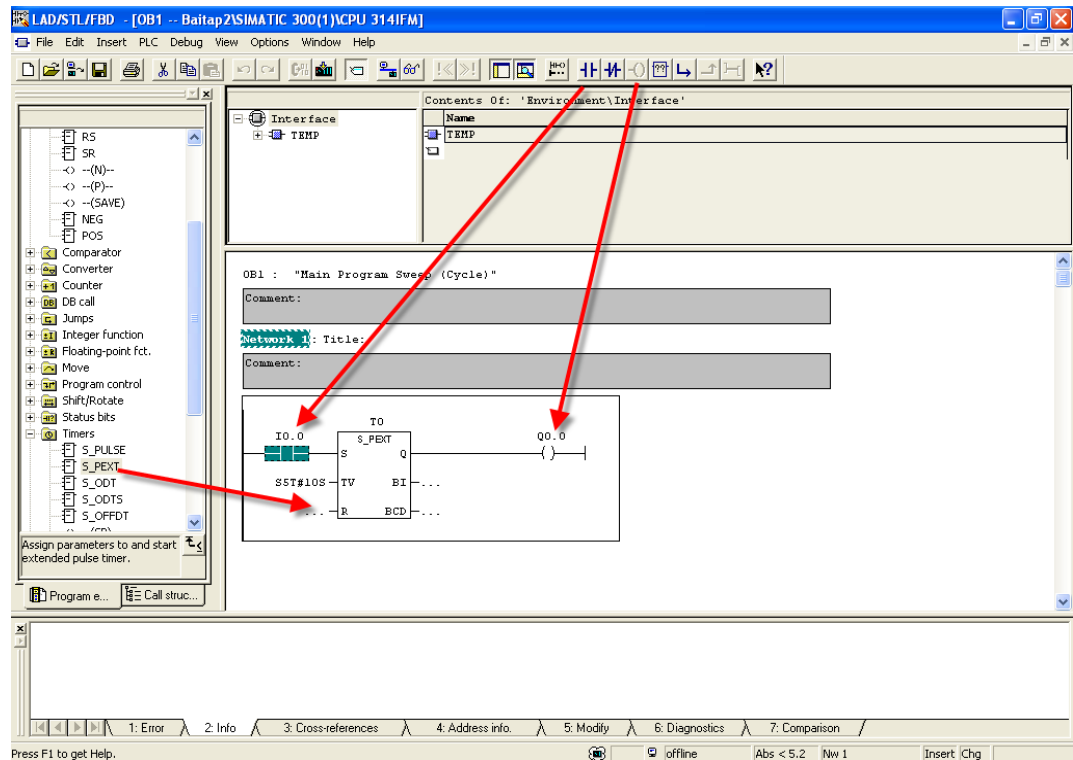




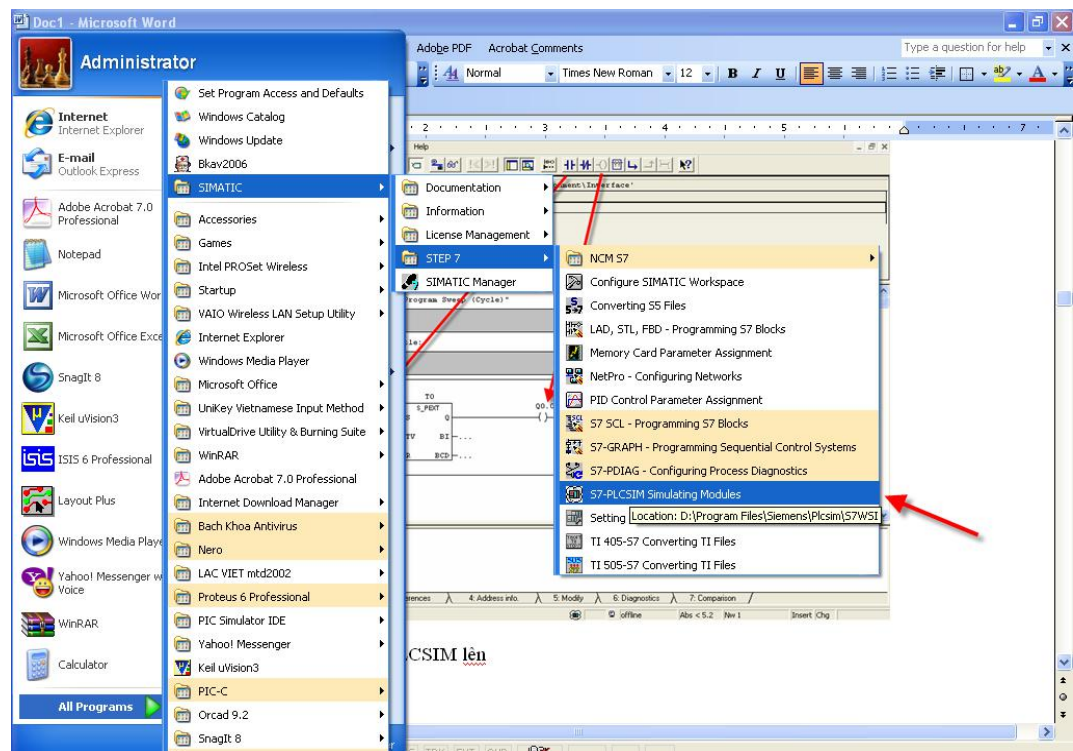


Bắt đầu lập trình:

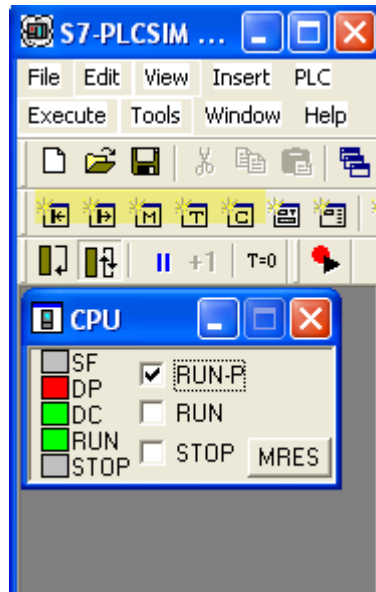
Tiếp điểm thường mở nhấn F2, thường đóng nhấn F3, Cuộn dây nhấn F6 còn các thiết bị khác nhấn Atl+ F9 (Hay có thể dùng chuột click chọn từng lệnh).



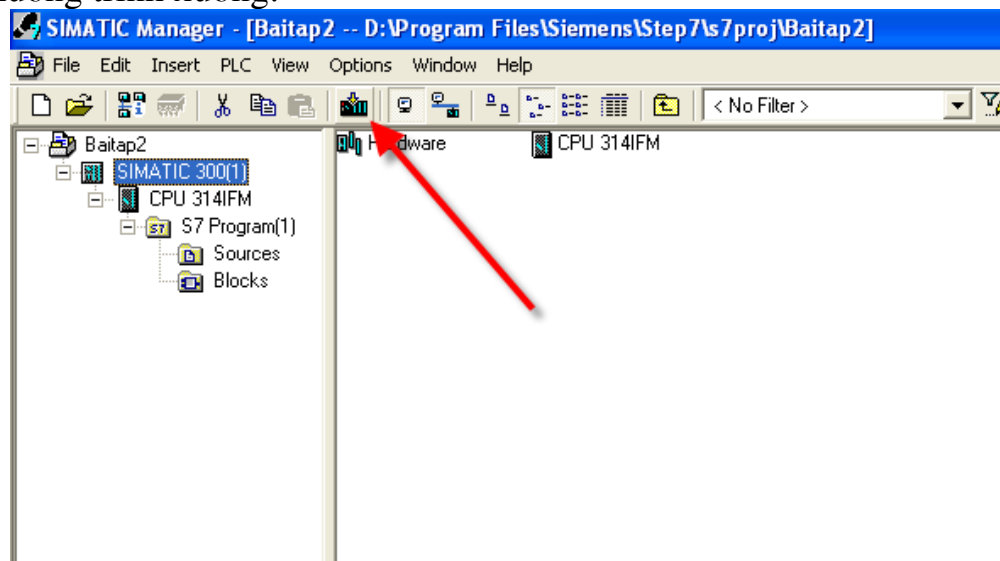
Khởi động chương trình PLCSIM lên



Mở các Input, OutPut, Memory, Timer, Con.....

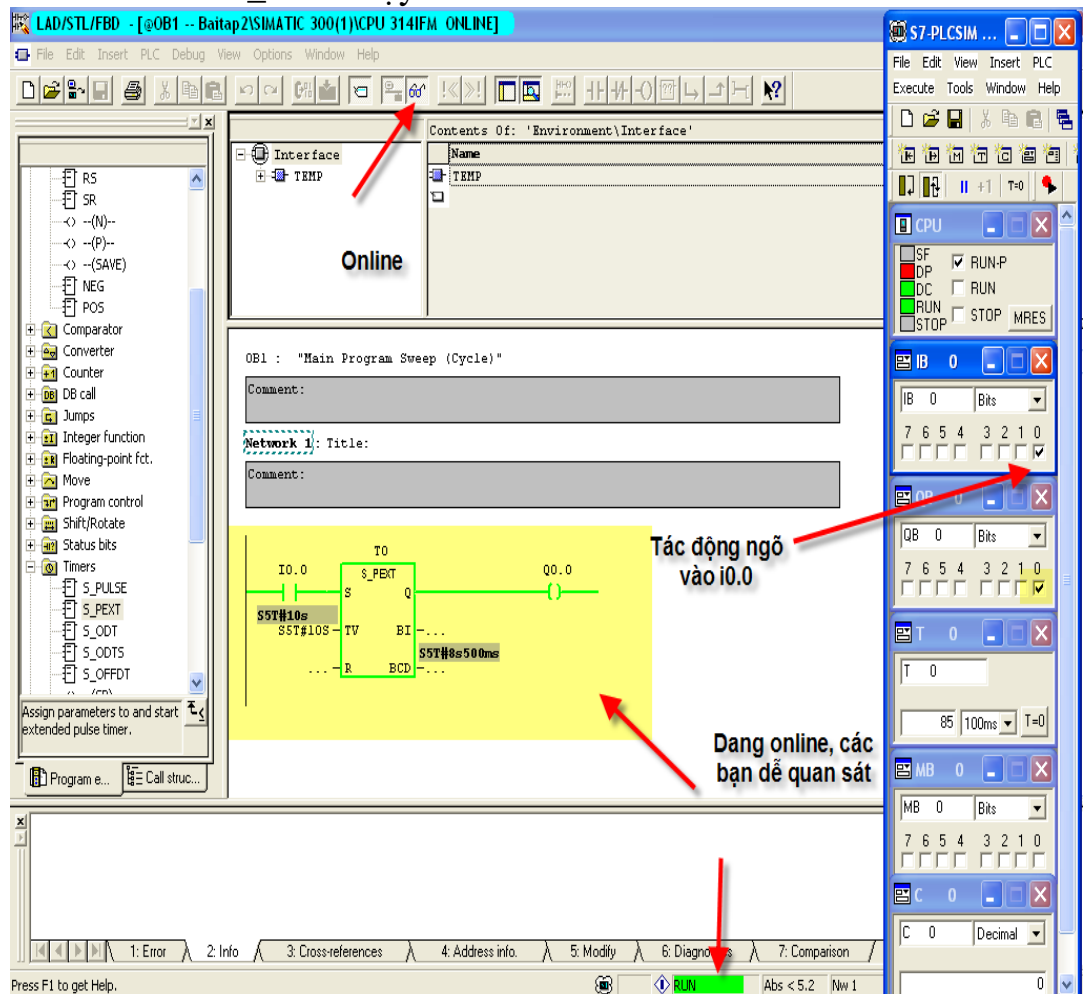


Sau đó về cửa sổ Simatic Manager chọn Station và download chương trình xuống.



Về cửa sổ lập trình cho khối OB1 để có thể online quan sát chương trình.

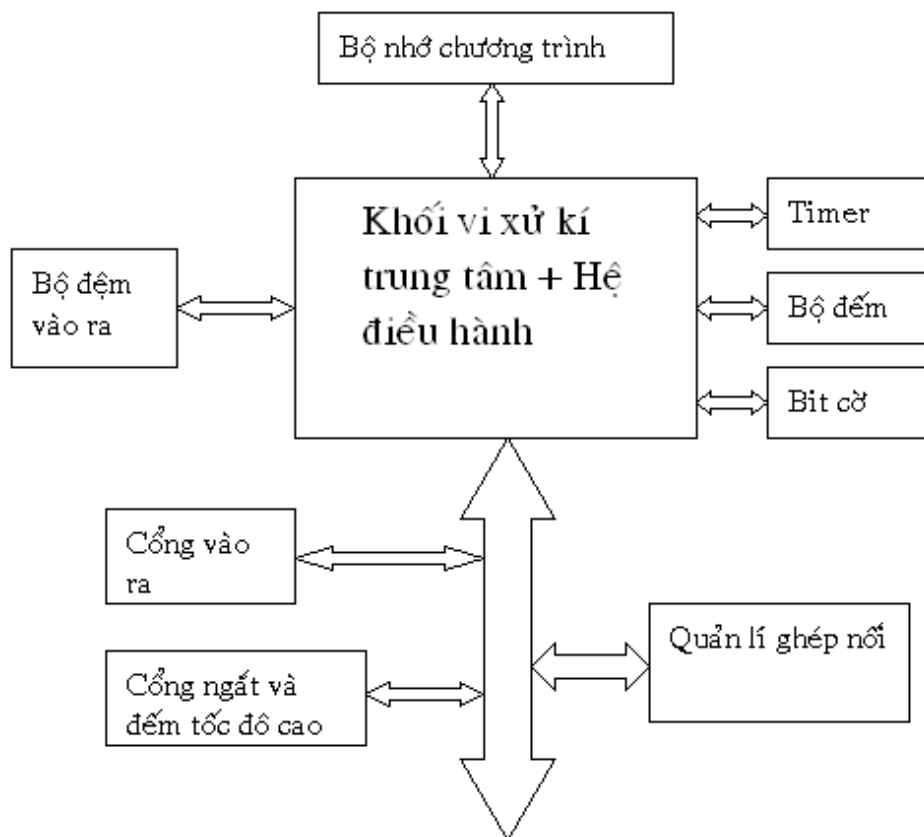
Click vào RUN_P để chạy PLCSIM.



3.3 Các vùng nhớ của PLC S7 - 300

Là Toàn bộ chương trình được lưu trong bộ nhớ dưới dạng các khối chương trình (OB,FC,FB..) và được thực hiện với chu kì quét.

Để có thể thực hiện một chương trình điều khiển. Tất nhiên PLC phải có tính năng như một máy tính. Nghĩa là phải có một bộ vi xử lý trung tâm (CPU), một hệ điều hành, một bộ nhớ chương trình để lưu chương trình cũng như dữ liệu và tất nhiên phải có các cổng vào ra để giao tiếp với các thiết bị bên ngoài..Bên cạnh đó, nhằm phục vụ bài toán điều khiển số, PLC phải có các khối hàm chức năng như Timer, Counter, và các hàm chức năng đặc biệt khác.



Nhắc lại

Các tín hiệu kết nối PLC :

+ Tín hiệu số : Là tín hiệu dạng Boolean, có giá trị 0 hoặc 1.

Vd: Tín hiệu từ nút nhấn, công tắc hành trình...

+ Tín hiệu tương tự : Là tín hiệu liên tục từ 0-10VDC hoặc từ 4-20mA.

Vd : Tín hiệu từ Loadcell, Sensor đo mức....

+ Các tín hiệu khác: Bao gồm các tín hiệu giao tiếp máy tính, giao tiếp với các thiết bị bên ngoài bằng các chuẩn giao tiếp khác nhau như RS232,RS485...

Các Module trong S7_300:

- + Module nguồn (PS : Power Supply)
- + Module CPU
- + Module tín hiệu vào ra (SM :Signal Module): Bao gồm tín hiệu số ,tín hiệu tương tự....
- + Module truyền thông (IM :Interface Module) : Module ghép nối ,là loại Module có chứa năng ghép nối từng loại Module lại với nhau.

Ví dụ : IM360 : Module truyền

 IM361 : Module nhận

- + Module chức năng (FM : Function Module): Module có chức năng riêng biệt như điều khiển Servo,điều khiển vị trí.
- + Module truyền thông (CP : Communication Module).

Bộ nhớ PLC : Có 3 vùng chính

Vùng chứa chương trình ứng dụng: Chia làm 3 miền chính.

OB (Organization Block): Chứa chương trình chính.

FC (Function): Miền chứa chương trình con ,được tổ chức thành hàm và có biến hình thức để trao đổi dữ liệu.

FB (Function Block): Chứa chương trình chính được tổ chức thành hàm và có khả năng trao đổi dữ liệu với bất kì 1 khối chương trình nào khác. Các dữ liệu này phải được xây dựng thành một khối dữ liệu riêng.

Vùng chứa tham số hệ điều hành: Chia làm 7 miền:

I (Process image input) : Miền dữ liệu các cổng vào số, trước khi bắt đầu thực hiện chương trình, PLC sẽ đọc giá trị logic của tất cả các cổng đầu vào và cất giữ chúng trong vùng nhớ I. Thông thường chương trình ứng dụng không đọc trực tiếp trạng thái logic của cổng vào số mà chỉ lấy dữ liệu của cổng vào từ bộ đệm I.

Q (Process Image Output): Miền bộ đệm các dữ liệu cổng ra số. Kết thúc giai đoạn thực hiện chương trình, PLC sẽ chuyển giá trị

logic của bộ đệm Q tới các cổng ra số. Thông thường chương trình không trực tiếp gán giá trị tới tận cổng ra mà chỉ chuyển chúng tới bộ đệm Q.

M (Miền các biến cờ): Chương trình ứng dụng sử dụng những biến này để lưu giữ các tham số cần thiết và có thể truy nhập nó theo Bit (M), byte (MB), từ (MW) hay từ kép (MD).

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

C (Counter): Miền nhớ phục vụ bộ đếm bao gồm việc lưu trữ giá trị đặt trước (PV- Preset Value), giá trị đếm tức thời (CV _ Current Value) và giá trị logic đầu ra của bộ đệm.

PI : Miền địa chỉ cổng vào của các Modul tương tự (I/O External input). Các giá trị tương tự tại cổng vào của modul tương tự sẽ được module đọc và chuyển tự động theo những địa chỉ.Chương trình ứng dụng có thể truy cập miền nhớ PI theo từng Byte (PIB), từng từ PIW hoặc từng từ kép PID.

PQ: Miền địa chỉ cổng ra cho các module tương tự (I/O External Output). Các giá trị theo những địa chỉ này sẽ được module tương tự chuyển tới các cổng ra tương tự. Chương trình ứng dụng có thể truy nhập miền nhớ PQ theo từng Byte (PQB), từng từ (PQW) hoặc theo từng từ kép (PQD).

Vùng chứa khối dữ liệu: Chia làm 2 miền:

DB (Data Block): Miền chứa dữ liệu được tổ chức thành khối. Kích thước cũng như số lượng khối do người sử dụng quy định, phù hợp với từng bài toán điều khiển. Chương trình có thể truy nhập miền này theo từng bit (DBX), byte (DBB), từ (DBW) hoặc từ kép (DBD).

L (Local data block) : Miền dữ liệu địa phương, được các khối chương trình OB, FC, FB tổ chức và sử dụng cho các biến nháp tức thời và trao đổi dữ liệu của biến hình thức với những khối chương trình gọi nó. Nội dung của một khối dữ liệu trong miền nhớ này sẽ bị xoá khi kết thúc chương trình tương ứng trong OB, FC, FB. Miền này có thể được truy nhập từ chương trình theo bit (L), byte(LB) từ (LW) hoặc từ kép (LD).

Vòng quét chương trình :

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

Chú ý rằng bộ đệm I và Q không liên quan tới các cổng vào ra tương tự nên các lệnh truy nhập cổng tương tự được thực hiện trực tiếp với cổng vật lý chứ không thông qua bộ đệm.

Thời gian cần thiết để PLC thực hiện 1 vòng quét gọi là thời gian vòng quét (Scan Time). Thời gian vòng quét không cố định, tức là không phải vòng quét nào cũng được thực hiện trong một khoảng thời gian như nhau. Có vòng quét được thực hiện lâu, có vòng quét được thực hiện nhanh tùy thuộc vào số lệnh trong chương trình được thực hiện và khối dữ liệu truyền thông trong vòng quét đó.

Chương trình trong S7_300 được lưu trong bộ nhớ của PLC ở vùng giành riêng cho chương trình và có thể được lập với 2 dạng cấu trúc khác nhau.

Lập trình tuyến tính: toàn bộ chương trình nằm trong một khối trong bộ nhớ. Loại hình cấu trúc tuyến tính này phù hợp với những bài toán tự động nhỏ, không phức tạp. Khối được chọn phải là khối OB1, là khối mà PLC luôn quét và thực hiện các lệnh trong đó thường xuyên, từ lệnh đầu tiên đến lệnh cuối cùng và quay lại lệnh đầu tiên.

Lập trình có cấu trúc: Chương trình được chia thành những phần nhỏ và mỗi phần thực thi những nhiệm vụ chuyên biệt riêng của nó, từng phần này nằm trong những khối chương trình khác nhau. Loại hình cấu trúc này phù hợp với những bài toán điều khiển nhiều nhiệm vụ và phức tạp.

Các khối OB đặc biệt:

OB10 (Time of Day Interrupt): Chương trình trong khối sẽ được thực hiện khi giá trị của đồng hồ thời gian thực nằm trong một khoảng thời gian đã được quy định. OB10 có thể gọi một lần, nhiều lần cách đều nhau từng phút, từng giờ, từng ngày ...Việc quy định khoảng thời gian hay số lần gọi OB10 được thực hiện nhờ chương trình hệ thống

SFC28 hoặc trong bảng tham số của module CPU nhờ phần mềm Step 7.

OB20 (Time Day Interrupt): Chương trình trong khối sẽ được thực hiện sau một khoảng thời gian trễ đặt trước kể từ khi gọi chương trình hệ thống SFC32 để đặt thời gian trễ.

OB35 (Cyclic Interrupt): Chương trình trong OB35 sẽ được thực hiện cách đều nhau 1 khoảng thời gian cố định. Mặc định khoảng thời gian này sẽ là 100ms, xong ta có thể thay đổi nó trong bảng tham số của module CPU, nhờ phần mềm Step7.

OB40 (Hardware Interrupt): Chương trình trong OB sẽ được thực hiện khi xuất hiện 1 tín hiệu báo ngắt từ ngoại vi đưa vào module CPU thông qua các cổng vào ra số onboard đặc biệt, hoặc thông qua các module SM, CP, FM.

OB80 (Cycle Time Fault): Chương trình trong khối OB80 sẽ được thực hiện khi thời gian vòng quét (Scan time) vượt quá khoảng thời gian cực đại đã được quy định hoặc khi có một tín hiệu ngắt gọi một khối OB nào đó mà khối OB này chưa kết thúc ở lần gọi trước. Mặc định thời gian Scan time cực đại là 150ms, nhưng có thể thay đổi nó thông qua bảng tham số của module CPU nhờ phần mềm Step 7.

OB81 (Power Supply fault): CPU sẽ gọi chương trình trong khối OB81 khi phát hiện thấy có lỗi về nguồn nuôi.

OB82 (Diagnostic Interrupt): Chương trình trong OB82 được gọi khi CPU phát hiện sự cố từ các Modul vào ra.

OB85 (Not Load fault): Chương trình trong OB82 được gọi khi CPU phát hiện thấy chương trình ứng dụng có sử dụng chế độ ngắt nhưng chương trình xử lý tín hiệu ngắt lại không có trong khối OB tương ứng.

OB87 (Communication fault): Khối OB87 sẽ được gọi khi CPU phát hiện thấy lỗi trong truyền thông ví dụ như không có tín hiệu trả lời từ các đối tác.

OB100 (Start Up Information): Khối OB100 sẽ được thực hiện 1 lần khi CPU chuyển trạng thái Stop sang Run.

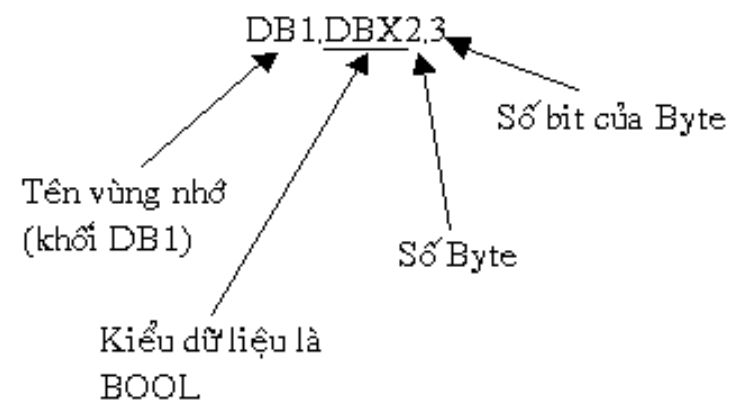
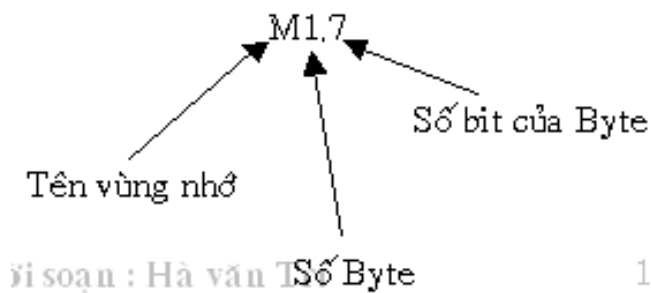
OB121 (Synchronous error): Khối OB121 sẽ được gọi khi CPU phát hiện thấy lỗi logic trong chương trình như đổi sai kiểu dữ liệu hoặc lỗi truy nhập khối DB, FC, FB không có trong bộ nhớ CPU.

OB122 (Synchronous error): Khối OB122 sẽ được gọi khi CPU phát hiện thấy lỗi truy cập module trong chương trình, ví dụ chương trình có lệnh truy nhập module vào ra mở rộng nhưng lại không tìm thấy module này.

Các vùng nhớ của S7300

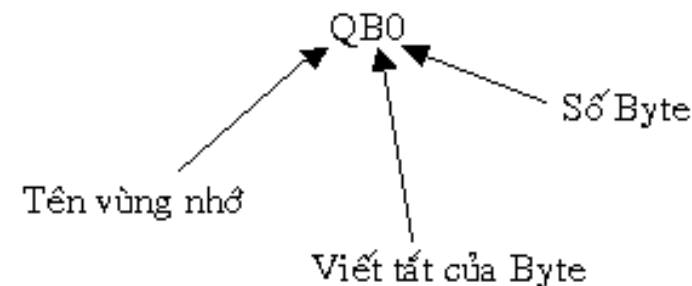
Kiểu Bool:

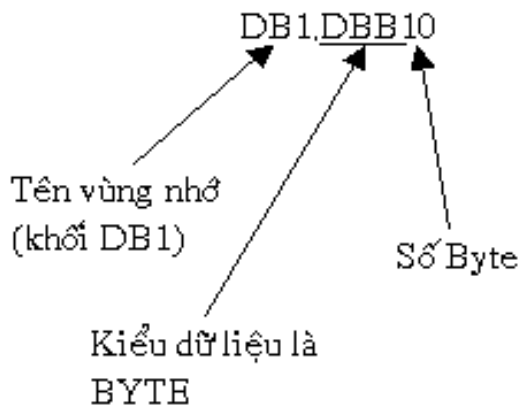
VD: Q0.0, I0.0, DB1.DBX2.3, M1.7,....



Kiểu Byte:

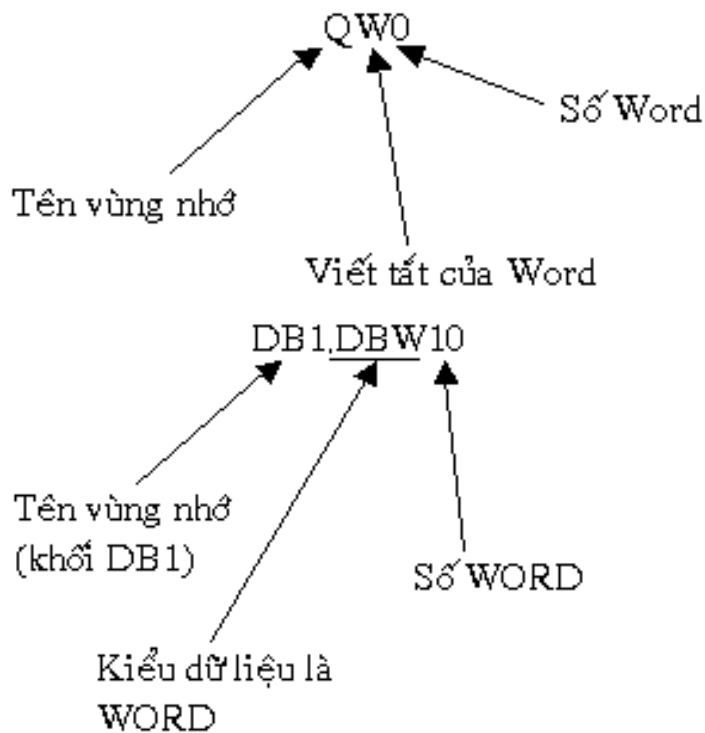
VD: QB0, MB3, VB10, SMB2, DB1, DBB10, ...





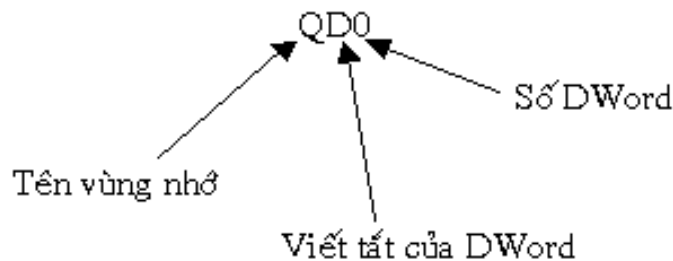
Kiểu Word:

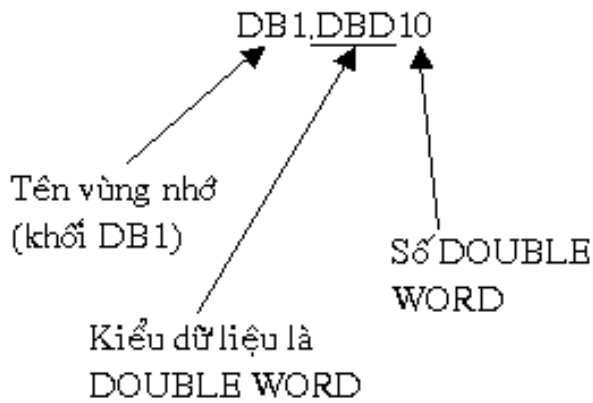
VD: IW0, QW0, MW3, DB1.DBW10, ...



Kiểu Double Word:

VD: ID0, QD0, MD3, DB1.DBD10, ...





Khối dữ liệu Trong SIMATIC thông thường vùng nhớ được sử dụng là khối Data Block.

Dung lượng vùng nhớ không phụ thuộc vào số DB mà chỉ phụ thuộc vào dung lượng của tổng tất cả các DB.

Chính vì vậy tùy ứng dụng ta có thể đặt tên cho khối DB phù hợp, các DB có thể phân biệt bởi tên và chỉ số.

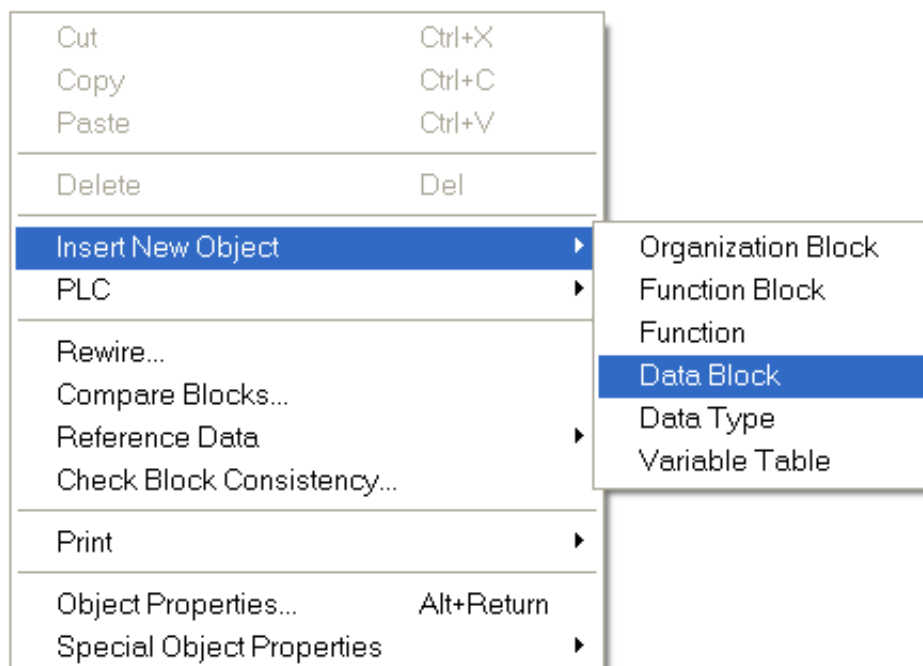
Ví dụ :

DB1, DB2, DB3....

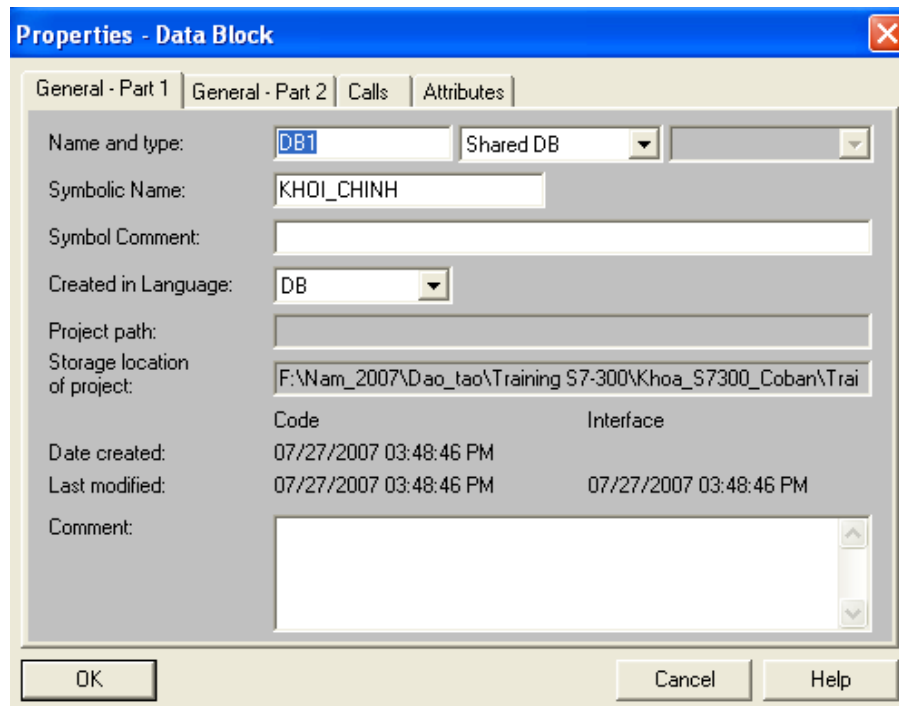
Khối DB được sử dụng như vùng nhớ trung gian để lưu trữ dữ liệu khi cần thiết, dữ liệu này được lưu ngay cả trong trường hợp mất điện.

Các bước xây dựng khối Data Block:

+ Click chuột phải, chọn Insert Data Block



- + Đặt tên cho khối DB và loại DB cho chương trình
- + Click OK để tạo ra khối DB1 có tên là khối chính



Tạo biến trong khối DB:

- + Đặt tên biến cho khối DB, loại dữ liệu cho biến
- + Đặt giá trị ban đầu cho biến

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	BIEN_1	DWORD	DW#16#0
+4.0	BIEN_2	REAL	0.000000e+000
+8.0	BIEN_3	INT	0
=10.0		END_STRUCT	

- + Để truy cập được khối DB ta cần phải xác định được biến cần truy.

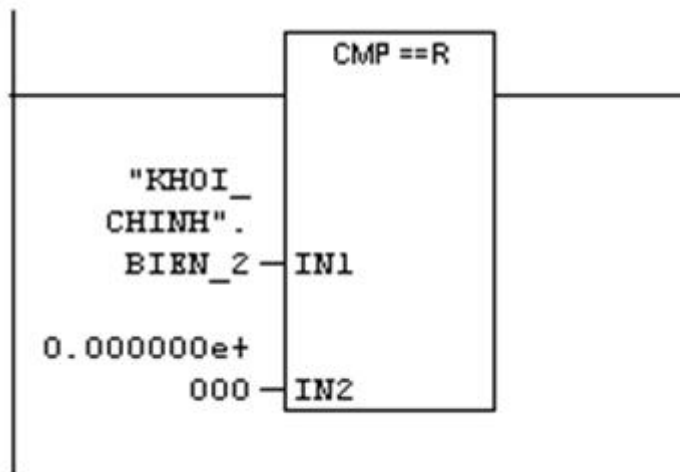
Có nhiều cách truy cập :

Truy cập theo vị trí

vdụ: DB1.DBDO

Truy cập theo tên

vdụ: DB1.BIEN_1

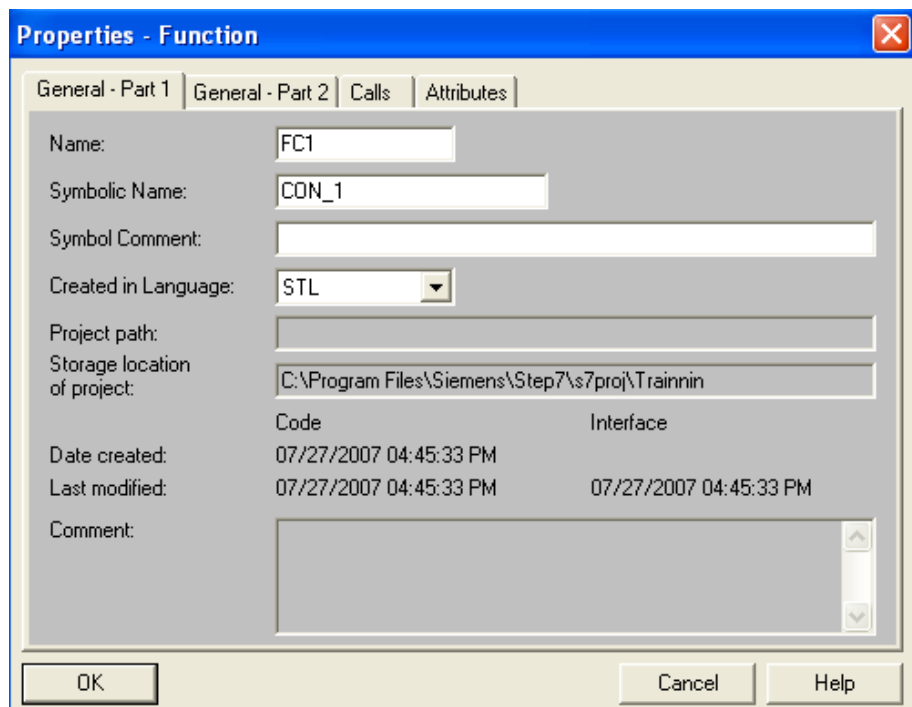


Cách tạo chương trình con:

+ Click chuột phải ,chọn Insert Function

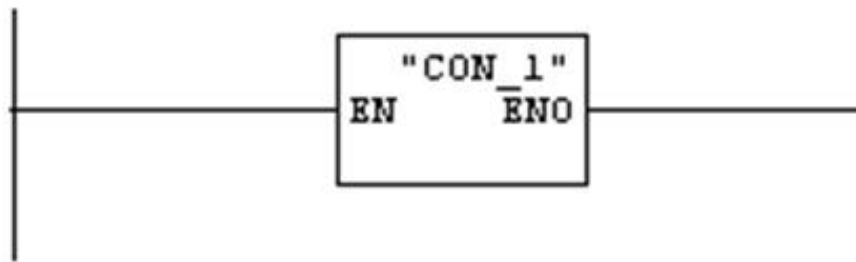


+ Đặt tên cho chương trình con, loại ngôn ngữ để viết chương trình.



Cách gọi chương trình:

+ Chương trình con sẽ được gọi trong chương trình chính



Bài tập Tạo chương trình con có tên FC1

Tạo khối dữ liệu DB1 có 3 biến

1. BIEN_1 : DINT
2. BIEN_2 : REAL
3. BIEN_3 : Bool

Chương trình FC1 được gọi trong chương trình chính

Chương trình chính lưu 3 biến các giá trị sau

BIEN_1: = 20

BIEN_2: = 25.0

BIEN_3: = true

3.4 Kết nối mạng

Truyền *Kết nối MPI (Multi-point Interface):*

thông
trên
MPI

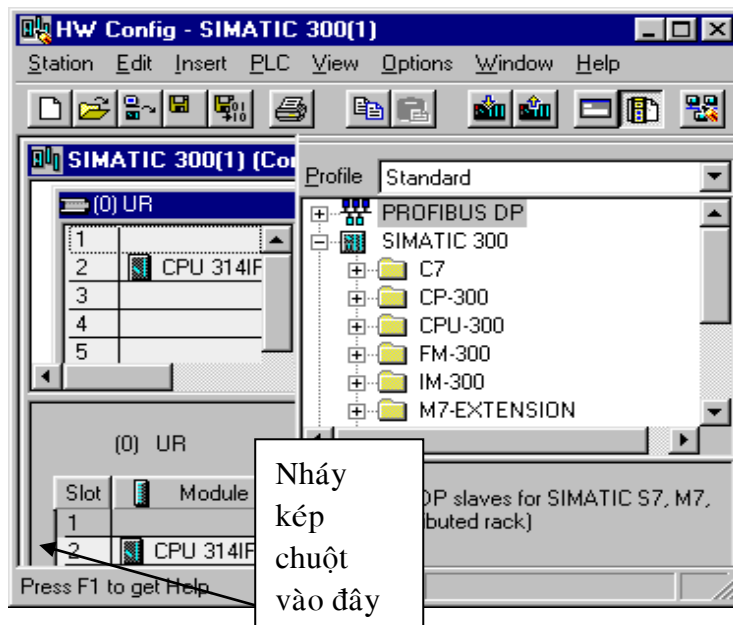
Đây là một kết nối trực tiếp đa điểm, tạo thành 1 mạng con nhằm trao đổi dữ liệu giữa các PLC (CPU, FM, CP) và thiết bị lập trình PC/PG.

Các thiết bị tham gia kết nối và các dữ liệu cần truyền đi và nhận về được xác lập khi tạo mạng.

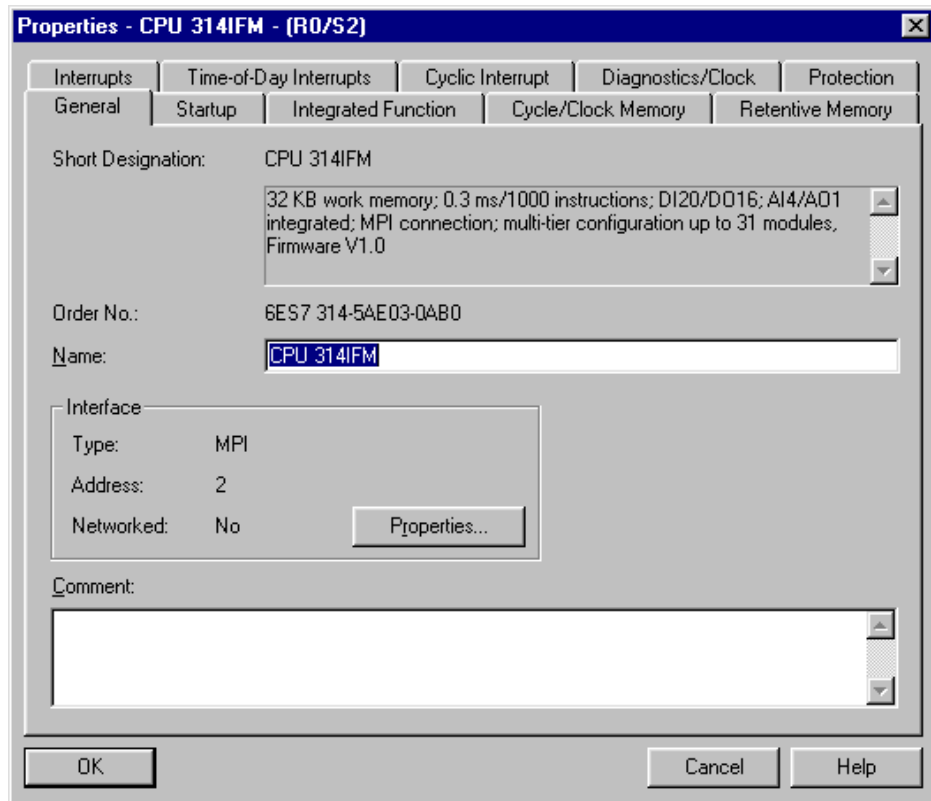
Các trạm (NODE) có 1 định danh riêng (số MPI) xác lập địa chỉ của trạm.

Cách qui định địa chỉ MPI cho module CPU

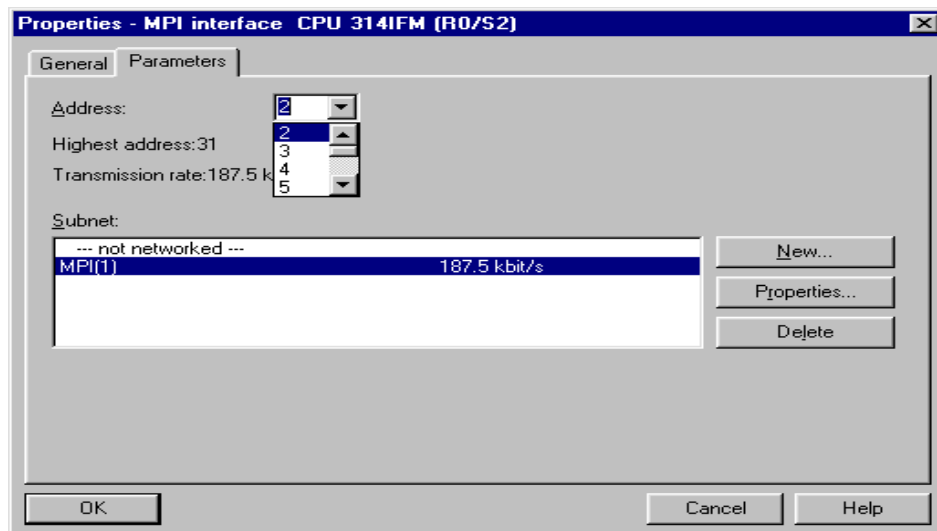
Vì máy tính có thể làm việc cùng lúc được với nhiều trạm PLC, nên sau khi ghép nối máy tính với các module CPU ta cần xác định địa chỉ truyền thông cho trạm PLC. Mặc định các module CPU đều có địa chỉ là 2 (địa chỉ MPI). Muốn thay đổi địa chỉ module CPU ta nháy kép phím chuột trái tại tên của module trong bảng khai báo cấu hình cứng để vào chế độ đặt lại tham số làm việc như sau:



Sau khi nháy kép chuột vào sẽ xuất hiện bảng sau:

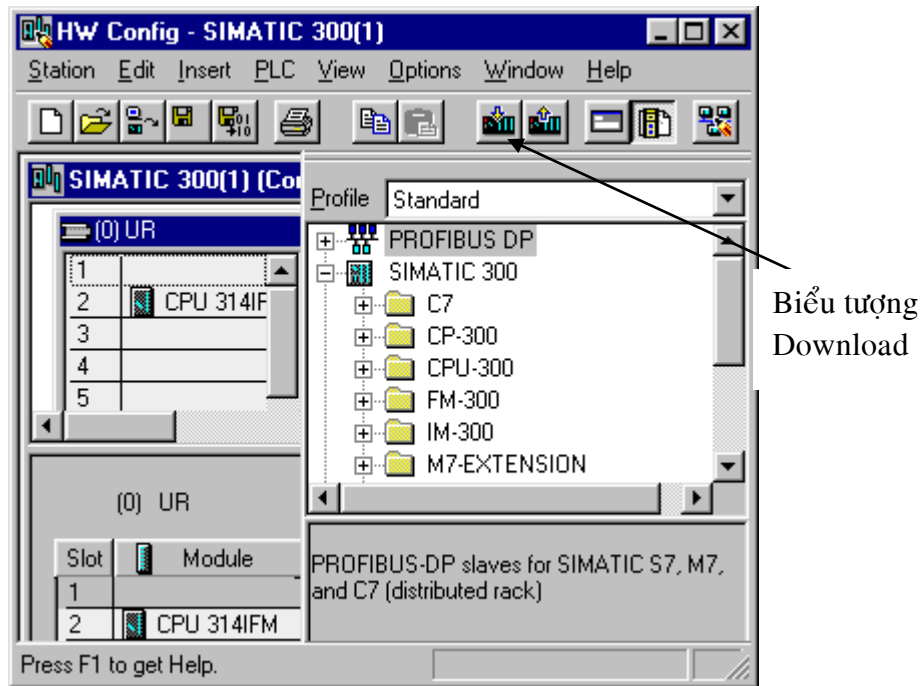


Kích chuột vào *Properties*, sau đó chọn tiếp *General* → *MPI* và sửa lại địa chỉ *MPI* như mong muốn:



Thay đổi địa chỉ ở phần *Address*:

Sau khi đã định nghĩa lại địa chỉ MPI cho trạm PLC, ta phải ghi lại địa chỉ đó lên module CPU và chỉ khi đó module CPU mới thực sự làm việc theo địa chỉ mới này. Sau đó ta Download xuống PLC bằng cách kích vào biểu tượng Download trên thanh công cụ.



Tính năng

- ✓ Các trạm được kết nối qua cổng MPI và cáp MPI. Giá trị MPI của các trạm được chọn trong khoảng 0÷126. Số trạm tham gia có thể đến 32.
- ✓ Chuẩn kết nối là RS485. Từ máy tính có đầu ra theo chuẩn RS232 ta phải dùng bộ nối chuyển đổi RS232 ↔ RS485.
- ✓ Tốc độ truyền có thể là 19.2Kbps, 187.5Kbps, 1.5Mp. Thường chọn tốc độ truyền là 187.5Kbps.
- ✓ Khoảng cách giữa các PLC bị giới hạn đến 50m.

Phương thức kết nối Vòng GD

Vòng GD (Global Data) là phương thức giao tiếp của các trạm trong mạng MPI.

Trong mỗi vòng GD xác định rõ số CPU tham gia. Số lượng cực đại là 15.

Trong vòng có nhiều hơn 2 CPU, chỉ có 1 CPU gửi tin, các CPU khác chỉ nhận tin.

Trong vòng chỉ có 2CPU, mỗi CPU có thể gửi 1 gói tin và nhận 1 gói tin.

Trong mỗi vòng có nhiều gói tin. Mỗi gói tin do 1 CPU gửi và có thể có nhiều CPU nhận. Trong 1 vòng mỗi CPU chỉ có 1 gói tin gửi và có 1 gói tin nhận.

Trong mỗi gói tin GD xác định rõ địa chỉ tin gửi và địa chỉ tin nhận của từng trạm. Các địa chỉ này có thể là:

- Vùng nhớ bit M
- Vùng nhớ ảnh ngõ vào I, ảnh ngõ ra Q
- Vùng nhớ khối dữ liệu DB
- Vùng nhớ T, C không nên sử dụng vì thời gian gửi không thích hợp với thời gian tồn tại dữ liệu, và chỉ được cấu hình trong phần gửi.
- Vùng nhớ PI, PQ và L không sử dụng trong truyền thông MPI

Tính năng

Mỗi CPU có thể tham gia đến 4 vòng GD.

Mỗi gói GD chỉ chứa tối đa 22 byte (nếu cùng vùng địa chỉ). Nếu sử dụng nhiều vùng địa chỉ thì phải giảm 2 byte cho mỗi vùng địa chỉ thêm vào

Mỗi dữ liệu Bit chiếm 1 byte trong gói GD.

Tạo dữ liệu toàn cục GD

Mở bảng GD từ mạng MPI (trong Project đã chọn), chọn trong Menu Option → Define Global Data

Điền nội dung vào bảng:

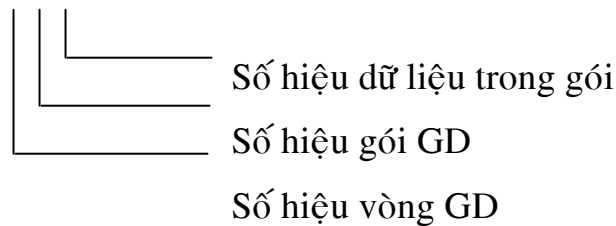
- + Xác định các CPU tham gia liên kết trong mạng MPI.
- + Ghi địa chỉ gửi và nhận cho mỗi gói GD.
- + Độ rộng (số byte) chứa trong vùng phát và thu phải hoàn toàn giống nhau.

Ví dụ:

GD Identifier	CPU 300(1)	CPU 300(2)
GD 1.1.1	>MW0	IW0
GD 2.1.1	>MW100:4	IW30:4

Ý nghĩa của GD Identifier:

GD 1.1.1



Ta xác định CPU gửi và thu cho mỗi dòng dữ liệu. Dấu > chỉ CPU gửi (phát).

Định danh GD do chương trình dịch tạo thành, ta không phải điền.

Biên dịch lần đầu

Sau khi điền các nội dung và xác định các CPU phát, thu cho bảng GD, ta tiến hành biên dịch lần đầu theo lệnh Menu: GD Table → Compile. Lúc này số hiệu định danh GD hiện lên trong coat tương ứng của bảng GD.

Lựa chọn tốc độ truyền thông (Scan rate):

+ Tốc độ truyền thông là số chu kỳ quét của OB1 của CPU mà đường truyền thực hiện 1 lần phát, thu tín hiệu GD.

+ Giá trị mặc định là 8, ta có thể viết đè lên giá trị ta mong muốn. Chu kỳ truyền thông T không thể quá nhỏ:

+ $T = \text{scan rate} \times \text{Scan cycle time} \geq 60\text{ms}$.

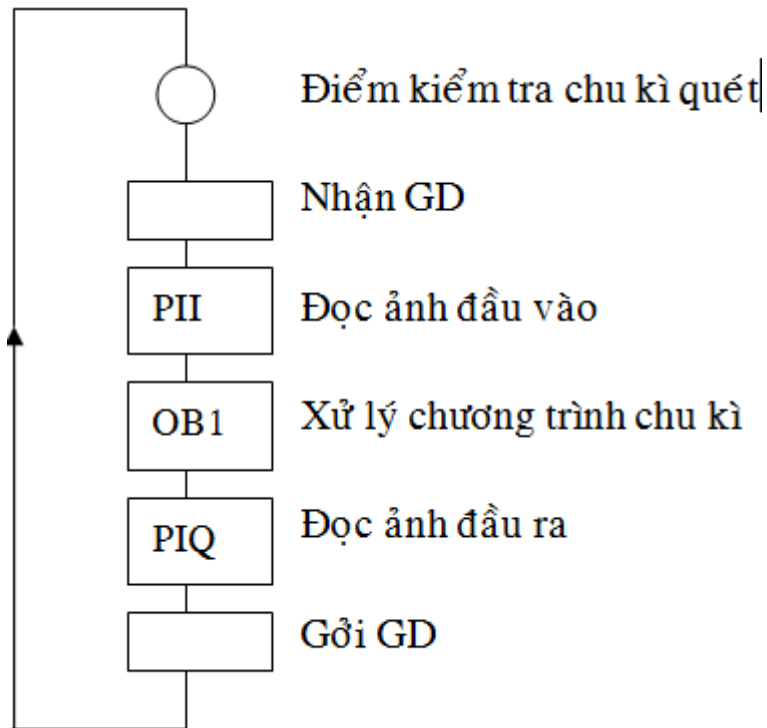
+ Việc lựa chọn tốc độ và ô nhớ trạng thái có thể không thực hiện (option).

Biên dịch lần hai

Sau khi lựa chọn tốc độ truyền và ô nhớ trạng thái, ta tiến hành biên dịch lần 2.

Sau khi biên dịch lần 2, ta có được bảng GD hoàn chỉnh, có thể nạp cho các CPU.

Trao đổi dữ liệu GD



CPU phát GD mỗi cuối chu kỳ, sau khi đã ghi kết quả ra ảnh PIQ.

CPU nhận GD vào đầu mỗi chu kỳ, trước khi ghi vào ảnh PII.

Kết nối Tạo liên kết mạng

Trong Simatic Manager ta Insert các trạm của mạng và chọn cấu hình phần cứng cho các mạng. Nên chọn đúng cấu hình phần cứng của trạm để khi Download khỏi bị lỗi.

Gán địa chỉ MPI cho từng trạm, các địa chỉ này cần phải khác nhau. Việc gán địa chỉ này cần được tiến hành riêng lẻ cho từng PLC.

Nạp chương trình cho các trạm

Ta dùng lệnh Download để nạp chương trình và dữ liệu hệ thống cho từng trạm.

Nối mạng MPI vật lý

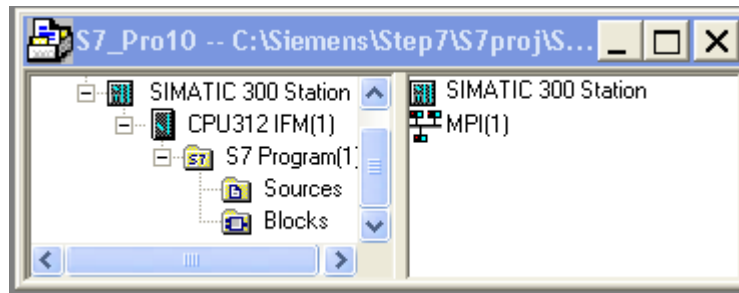
Dùng cáp MPI nối các CPU trong mạng, sau đó cho các PLC trong mạng hoạt động.

Ví dụ Các bước lập trình mạng

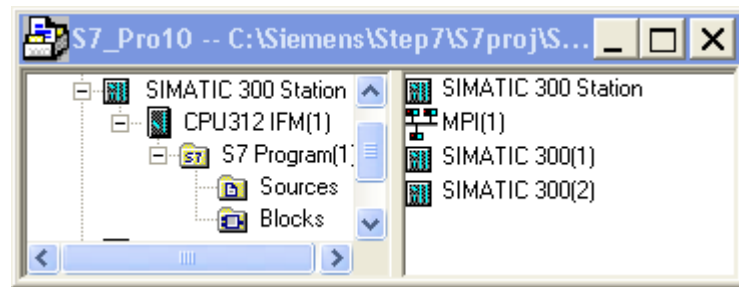
Giả sử tạo mạng MPL gồm ba PLC S7300 .

+ Vào Simatic Manager-File- New Project Wizard ta tạo một Project

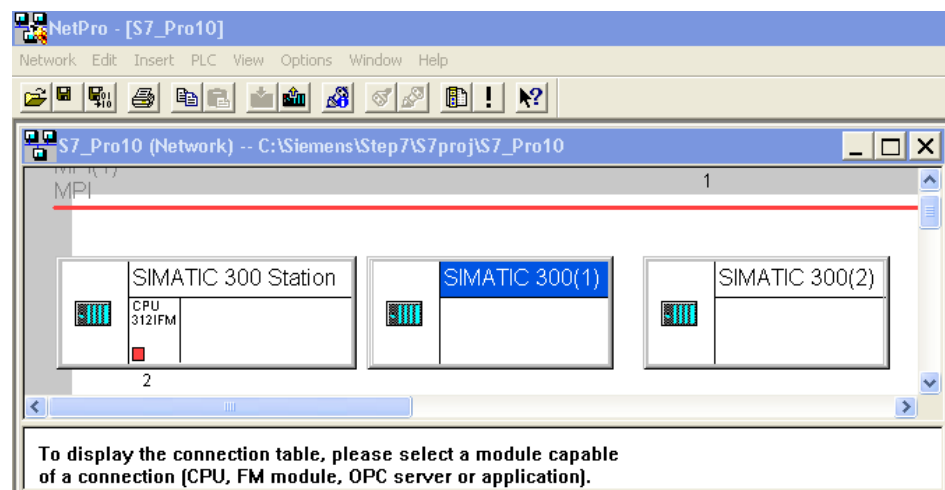
với S7-300 Station và Subnet MPI.



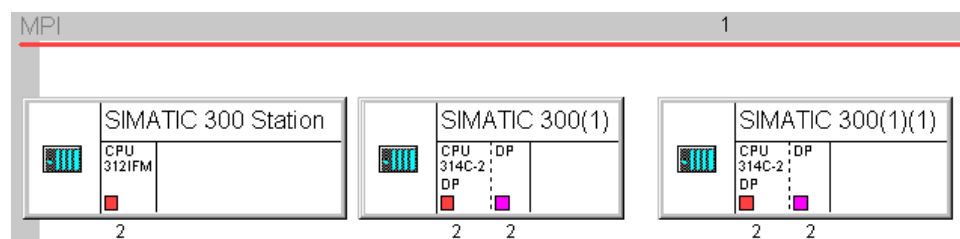
+ Vào menu Insert- Station ta thêm hai S7-300 station.



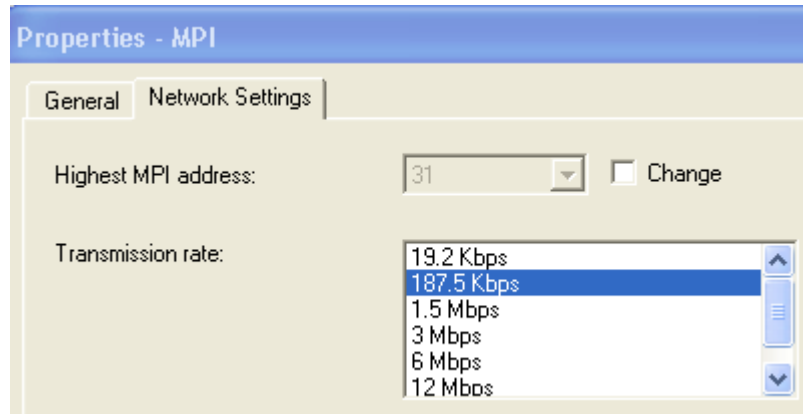
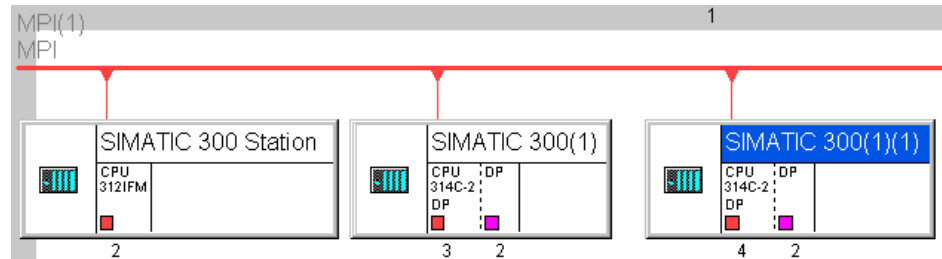
+ Bấm chuột kép vào biểu tượng MPI(1) xuất hiện cửa sổ NetPro.



Bấm chuột kép vào các SIMATIC 300 Station để đặt cấu hình cho các trạm (loại CPU , các module).



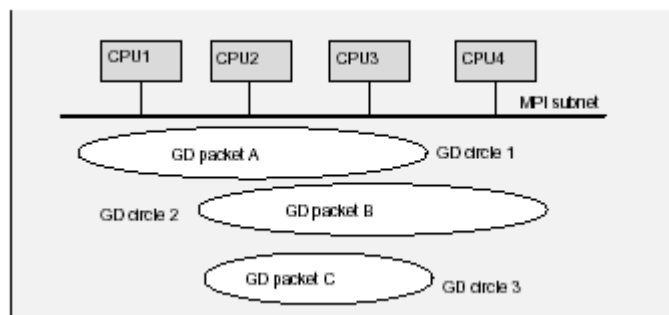
+ Bấm giữ chuột vào ô vuông dưới chữ CPU rồi kéo lên đường ngang MPI, sau đó bấm chuột kép vào ô CPU để đặt địa chỉ, các PLC trong mạng phải có địa chỉ riêng từ 2 đến 31, máy tính có địa chỉ 0, vận tốc truyền trên mạng có thể chọn từ 19.2kbps đến 12Mbps (tùy loại CPU).



Truyền thông theo chu kỳ

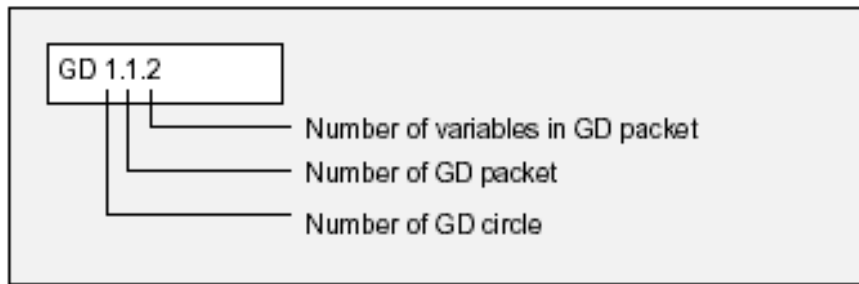
Theo chu kỳ quét của OB1, các dữ liệu vùng nhớ M, T,C, I/O được truyền từ một CPU đến mọi CPU khác theo cách truyền global data communication (GD). Dữ liệu được truyền đi ở cuối chu kỳ quét và nhận ở đầu chu kỳ quét.

Dữ liệu được truyền theo từng gói GD (GD packet), các CPU tham gia vào việc truyền tạo nên vòng GD (GD circle). Số lượng dữ liệu tối đa được truyền phụ thuộc loại CPU. Ví dụ có ba vòng GD, vòng 1 CPU1 truyền MB100 đến địa chỉ MB100 của CPU 2 và 3, đến EB100 của CPU 1 và EB102 của CPU 3; Vòng 2 CPU2 truyền MW120 đến AW40 của CPU3 và EW40 của CPU4...



Bảng GD gồm nhiều hàng ứng với các vòng, mỗi hàng có cột GD Identifier là tên của gói GD, các CPU tham gia vào truyền thông

(tối đa 15) và địa chỉ các dữ liệu

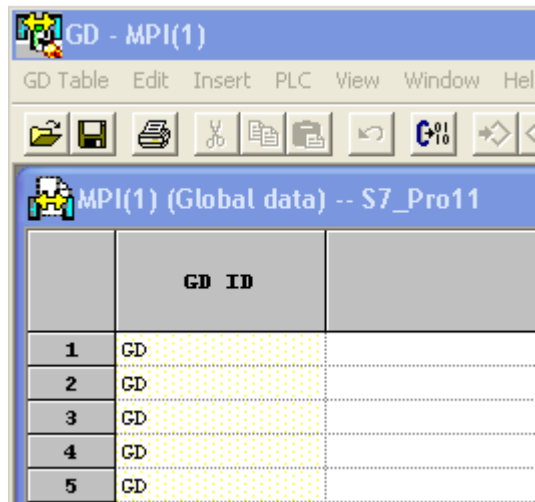


Global Data for Subnet "Example/MPI-Net1"				
GD Identifier	CPU 1	CPU 2	CPU 3	CPU 4
GD 1.1.1	>>MB100	MB100	MB100	
GD 1.1.2	>>MB100	EB100	EB102	
GD 2.1.1		>>MW120	AW40	EW40
GD 3.1.1		MW30:8	>>MW30:8	

GD Resources	CPU 312	CPU 412	CPU 416
	CPU 313	CPU 413	
	CPU 314	CPU 414	
	CPU 315		
Max. No. of GD circles per CPU	4	8	16
Max. No. of receive GD packets for all GD circles	4	16	32
Max. No. of send GD packets per GD circle	1	1	1
Max. No. of receive GD packets per GD circle	1	2	2
Max. No. of send GD packets for all GD circles	4	8	16
Max. length of GD packet ¹⁾	22 bytes	54 bytes	54 bytes
Reduction factor	1 to 225	1 to 255	1 to 255
Event-driven data transfer	No	Yes	Yes

¹⁾ The values in the table are based on one variable. The maximum size of useful data per GD packet is reduced by 2 bytes for each additional variable.

+ Bảng GD được tạo ra bằng cách bấm chuột phải vào biểu tượng MPI, chọn Define Global Data



+ Bổ sung tên các CPU tham gia vào gói truyền, các địa chỉ gửi và nhận, sau đó vào menu GD Table- Save rồi Compile

+ Vào menu View- Scan Rates, GD Status để nhìn chi tiết hơn

	GD ID	CPU 1\ CPU312 IEM(1)	CPU 2\ CPU 314IEM	CPU 3\ CPU 316
1	GD 1.1.1	>M01	M01	M01
2	GD 2.1.1	M02	>I00	M02
3	GD			

4

Bài tập thực hành PLC S7-200; S7-300

Chương này trình bày một số bài tập thực hành ứng dụng trong thực tế sử dụng PLC S7-200.

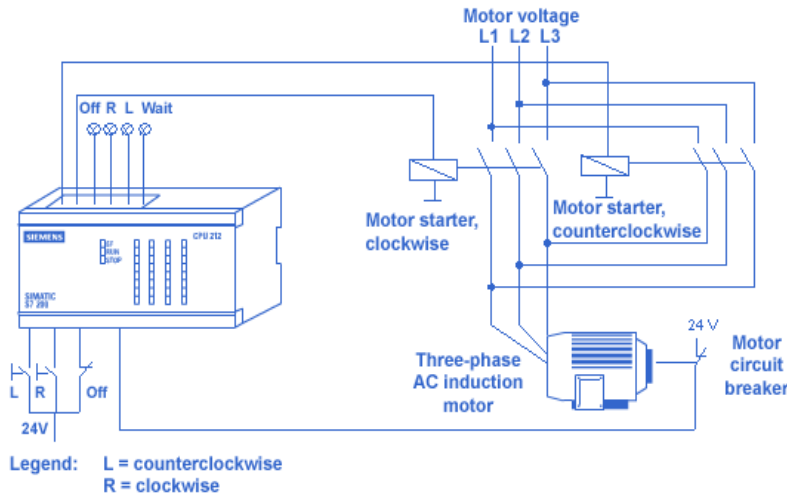
4.1 Điều khiển động cơ

4.2 Điều khiển dây chuyền

4.3 Các dạng điều khiển khác

4.1 Điều khiển động cơ

4.1.1 Điều khiển đảo chiều động cơ 3 pha



Yêu cầu công nghệ

Khi nhấn nút nhấn L, động cơ khởi động quay nghịch (ngược chiều kim đồng hồ). Khi nhấn nút nhấn R, động cơ quay thuận (thuận chiều kim đồng hồ). Tuy nhiên để động cơ hoạt động được điều trước tiên là đóng CB bảo vệ động cơ (hay Overload) và công tắc OFF. Sự đảo chiều không thể thay đổi nếu công tắc OFF chưa được bật lại và thời gian chờ là 5s. (Điều này có tác dụng chờ cho động cơ dừng hẳn để bắt đầu quay theo chiều khác). Các chế độ làm việc đều được thông báo bằng đèn báo (Off, R, L, Wait).

Nhiệm vụ:

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.

Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, lệnh các tiếp điểm đặc biệt và bit nhớ.
- Hiểu và biết cách sử dụng lệnh điều khiển Timer, lệnh các tiếp điểm đặc biệt và bit nhớ trong quá trình soạn thảo.

Các kiến thức cần thiết:

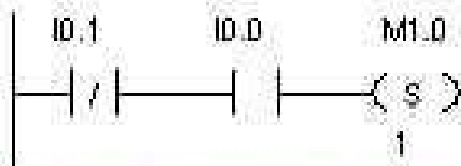
- Xem lại lệnh điều khiển Timer, lệnh các tiếp điểm đặc biệt và bit nhớ.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

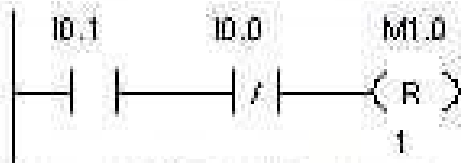
Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn – công tắc, các cầu dao, khởi động từ và động cơ.

Ví dụ tham khảo:

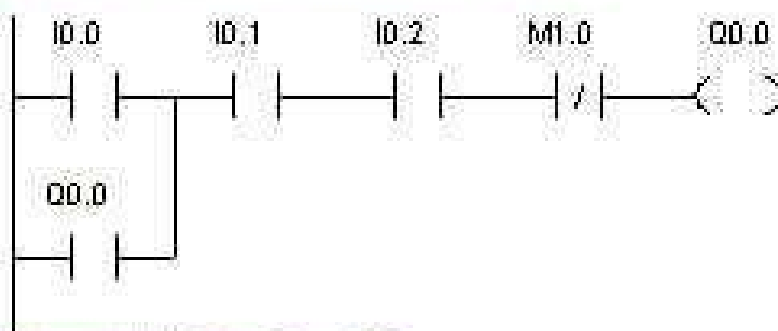
Network 1 // Set the Interlock



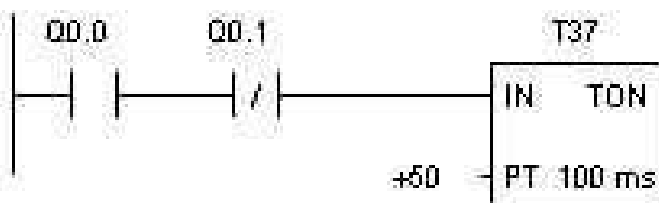
Network 2 // Clear the Interlock



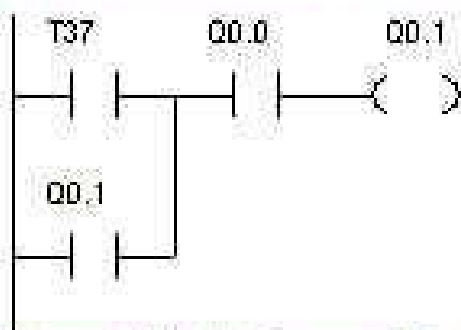
Network 3 // Switch On the Motor Starter Bit



Network 4 // Start Timer T37

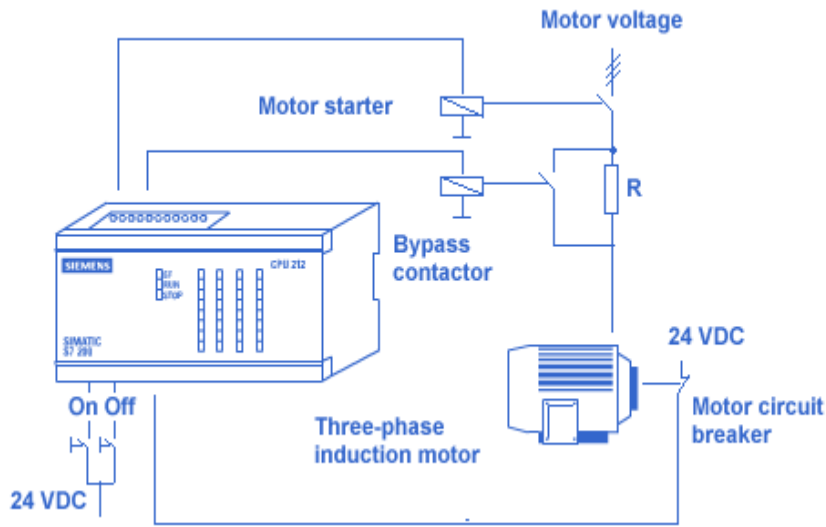


Network 5 // Close the Bypass Contactor



Network 6 // Main Program End

4.1.2 Điều khiển khởi động động cơ xoay chiều 3 pha rotor lồng sóc qua điện trở



Yêu cầu công nghệ

Khi nhấn nút nhấn On, động cơ khởi động thông qua điện trở R (Dạng khởi động mềm làm giảm dòng khởi động). Sau một khoảng thời gian chỉnh định trước, động cơ đạt tới tốc độ định mức thì R được loại khỏi mạch điện. Động cơ làm việc không thông qua điện trở R. Mạch có mắc CB bảo vệ động cơ (hay Overload).

Nhấn nút Off, động cơ dừng hoạt động.

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

Nhiệm vụ:

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.

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:

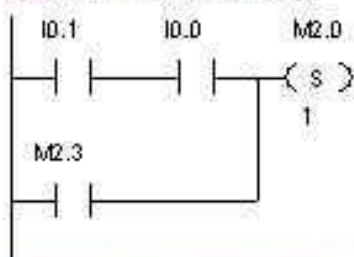
- Xem lại lệnh điều khiển Timer và bit nhớ.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

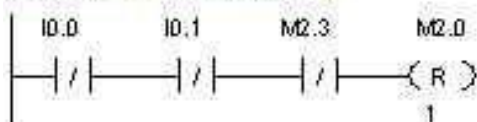
Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn – công tắc, các cầu dao, khởi động từ và động cơ.

Ví dụ tham khảo:

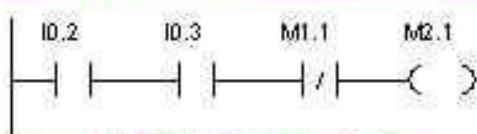
Network 1 // Set the Interlock



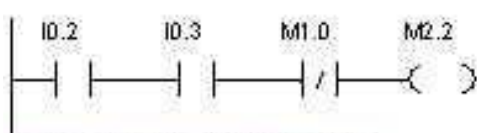
Network 2 // Clear the Interlock



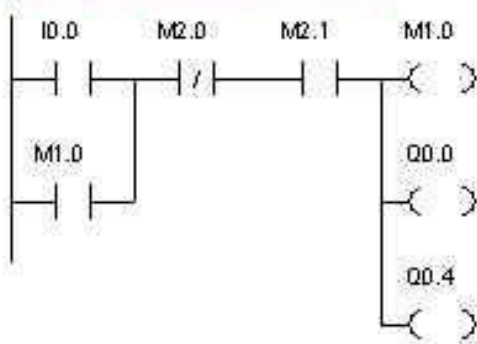
Network 3 // Enable Counterclockwise Rotation



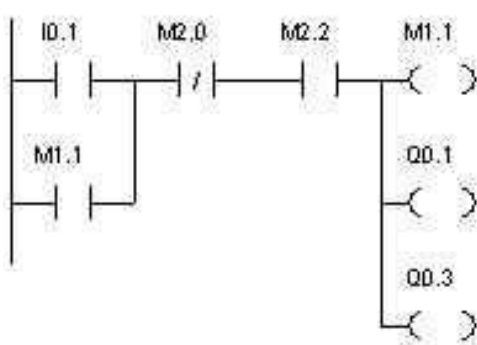
Network 4 // Enable Clockwise Rotation



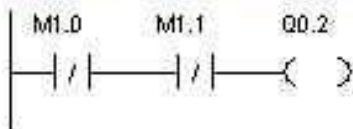
Network 5 // Run Counterclockwise



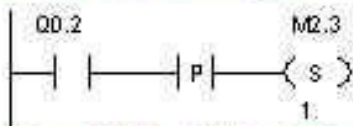
Network 6 // Run Clockwise



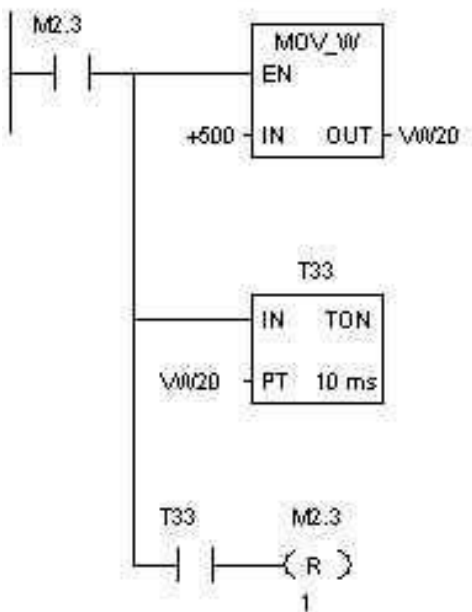
Network 7 // Turn On "Off" Lamp When Status Bits Not Set



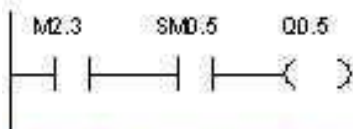
Network 8 // Set Waiting Period Timer Bit



Network 9 // Set Waiting Period and Start Timer

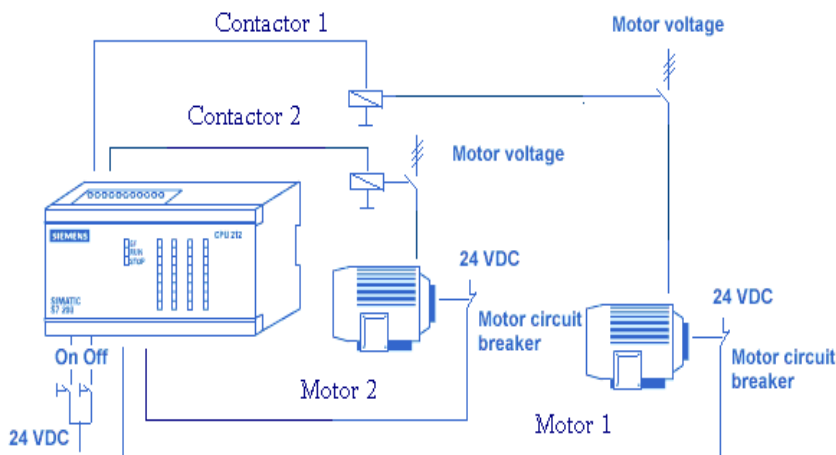


Network 10 // Flash Signal Lamp during Waiting Period



Network 11 // Main Program End

4.1.3 Điều khiển hai động cơ xoay chiều 3 pha



Yêu cầu công nghệ

Khi nhấn nút nhấn On, động cơ 1 khởi động chạy 5s rồi ngừng, đồng thời động cơ 2 chạy 5s rồi ngừng 5s, động cơ 2 làm việc lặp lại 5 lần như vậy rồi ngừng, động cơ 1 chạy lại. Chu kỳ làm việc của 2 động cơ lặp lại 10 lần thì dừng hẳn. Muốn làm việc lại thì nhấn nút On. Mạch có mắc CB bảo vệ động cơ (hay Overload).

Nhấn nút Off, động cơ 1 và 2 dừng khẩn cấp.

Hãy viết chương trình điều khiển 2 động cơ 3 pha trên.

Nhiệm vụ:

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.

Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, Counter và bit nhớ.
- Hiểu và biết cách sử dụng lệnh điều khiển Timer, Counter và bit nhớ trong quá trình soạn thảo.

Các kiến thức cần thiết:

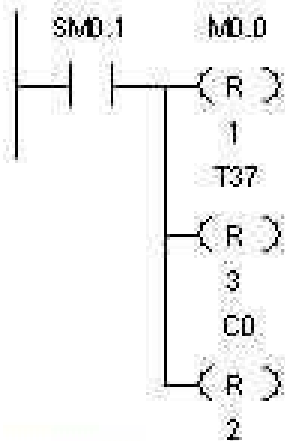
- Xem lại lệnh điều khiển Timer, Counter và bit nhớ.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

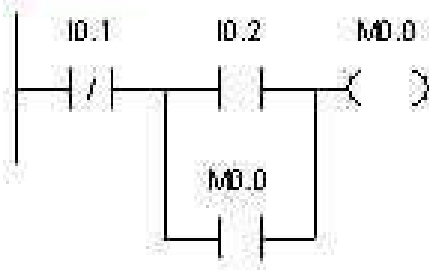
Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn – công tắc, các cầu dao, khởi động từ và động cơ.

Ví dụ tham khảo:

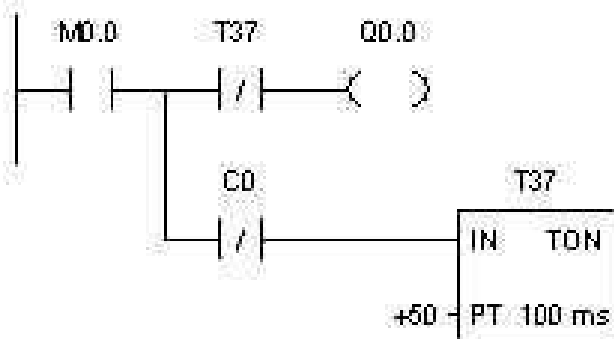
Network 1 // Network Title



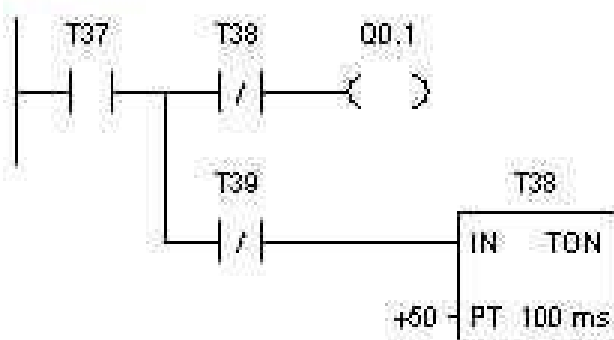
Network 2



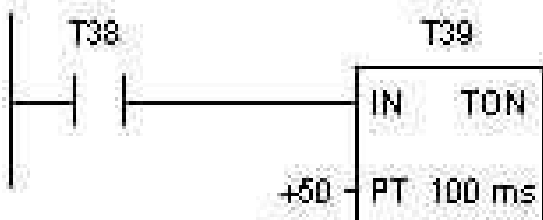
Network 3



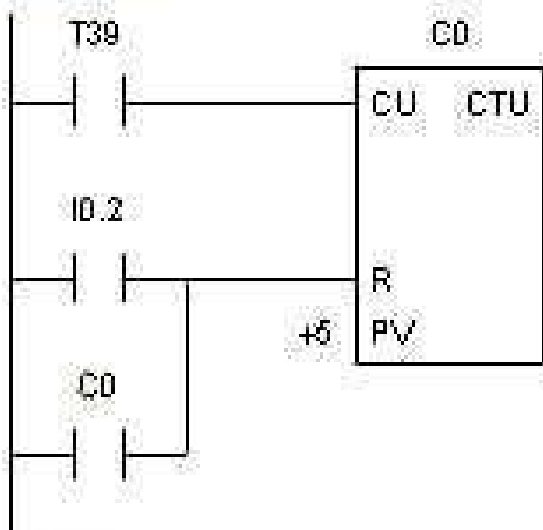
Network 4



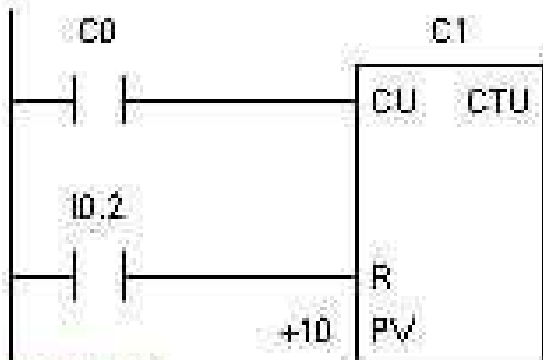
Network 5



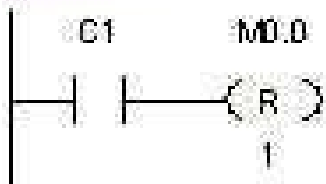
Network 6



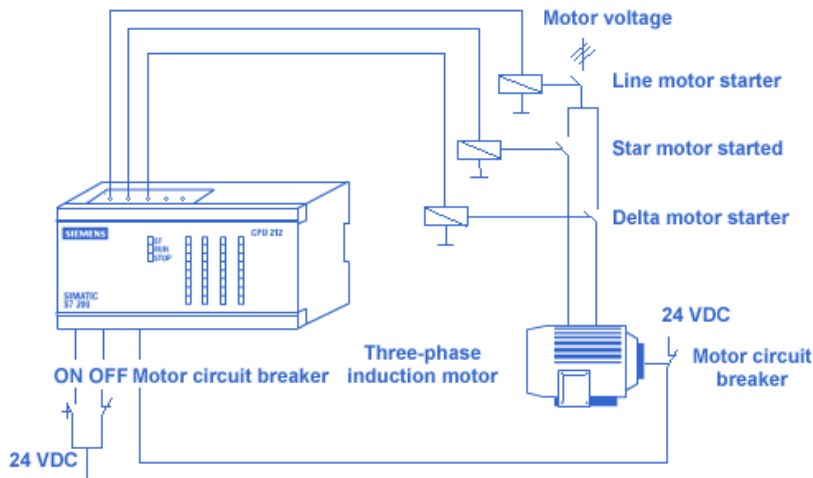
Network 7



Network 8



4.1.4 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.

Nhiệm vụ:

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.

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:

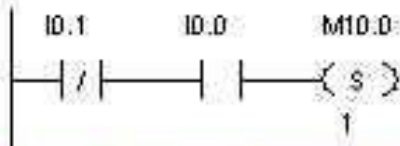
- Xem lại lệnh điều khiển Timer và bit nhớ.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

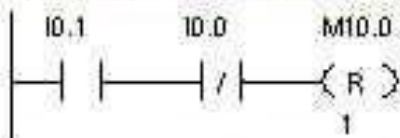
Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn – công tắc, các cầu dao, khởi động từ và động cơ.

Ví dụ tham khảo:

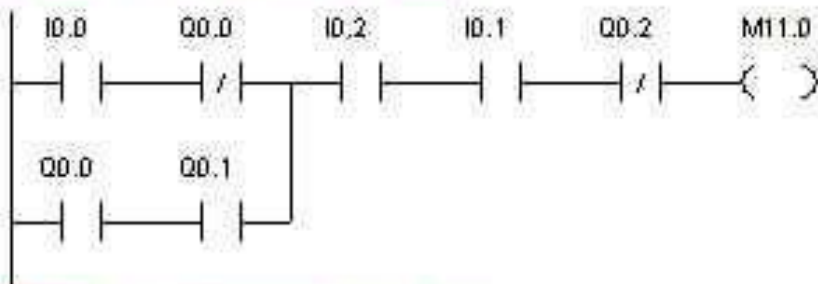
Network 1 // Set the Interlock



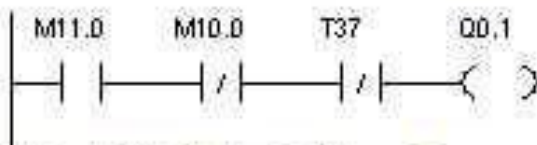
Network 2 // Clear the interlock



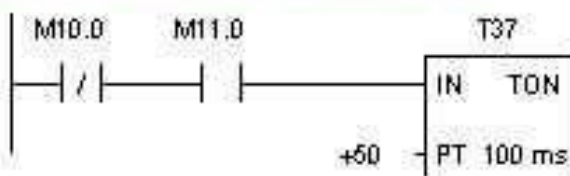
Network 3 // Enable the Motor Startup



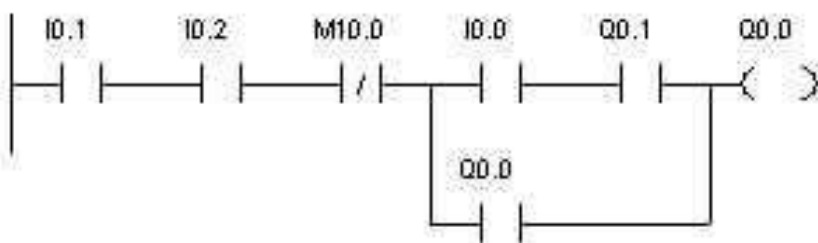
Network 4 // Switch On Star Motor Starter



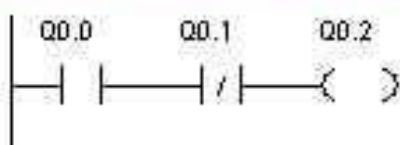
Network 5 // Start the Switchover Time



Network 6 // switch On Line Motor Starter

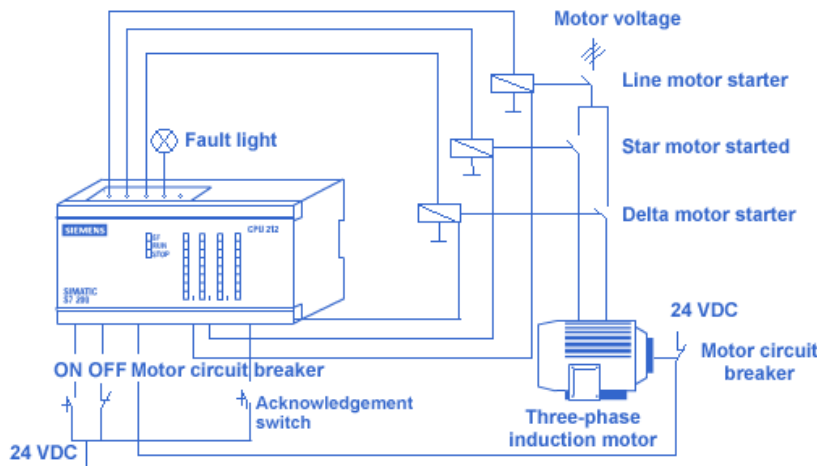


Network 7 // switch Delta Motor Starter



Network 8 // Main Program End

4.1.5 Điều khiển khởi động động cơ 3 pha dạng sao – tam giác có báo lỗi khởi động



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).

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.

Nhiệm vụ:

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.

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.

4.1.6 Điều khiển động cơ xoay chiều 3 pha theo chu kỳ làm việc 1

Yêu cầu công nghệ

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

Nhấn nút On (động cơ chuẩn bị làm việc), sau đó động cơ được chọn chế độ làm việc như sau:

- Nếu nhấn nút nhấn PB1 thì động cơ chạy 50s, dừng 10s rồi chạy ngược 50s, dừng 10s và chu kỳ lặp lại 3 lần như ban đầu.
- Nếu nhấn nút nhấn PB2 thì động cơ chạy 50s, dừng 10s rồi chạy ngược 50s, dừng 10s và chu kỳ lặp lại 5 lần như ban đầu.

Nút nhấn Off để dừng khẩn cấp.

Hãy viết chương trình điều khiển động cơ 3 pha theo yêu cầu công nghệ 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.

Ký hiệu	Vào/Ra	Mô tả
D	I0.0	Nút nhấn dừng động cơ
On	I0.1	Nút nhấn khởi động động cơ
PB1	I0.2	Nút nhấn làm việc 3 lần
PB2	I0.3	Nút nhấn làm việc 5 lần
T	Q0.0	Ngõ ra chạy thuận
N	Q0.1	Ngõ ra chạy nghịch

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:

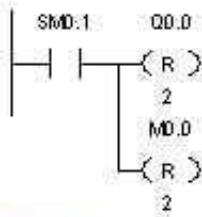
- Xem lại lệnh điều khiển Timer, chức năng di chuyển, lệnh các tiếp điểm đặc biệt và bit nhớ.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

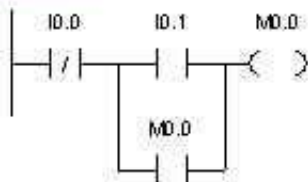
Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn, công tắc, các cầu dao, khởi động từ và động cơ.

Ví dụ tham khảo:

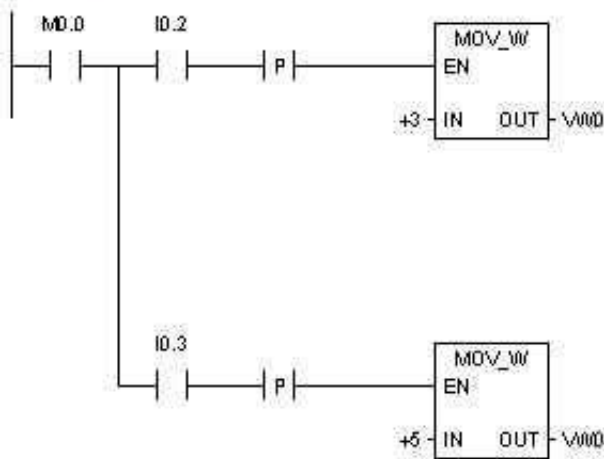
Network 1 // Network Title



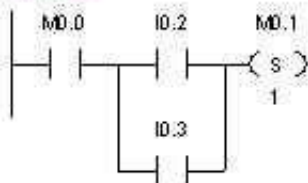
Network 2



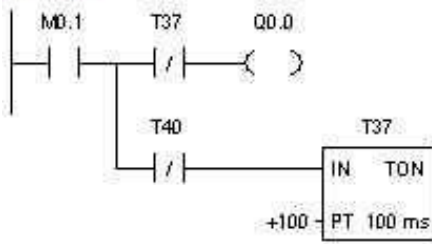
Network 3



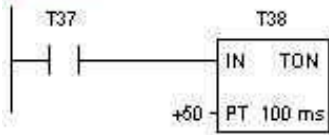
Network 4



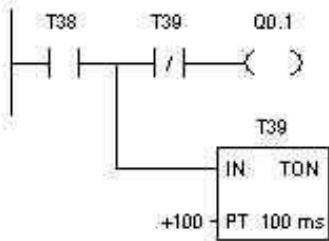
Network 5



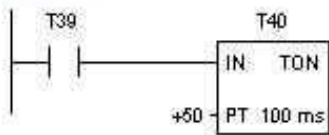
Network 6



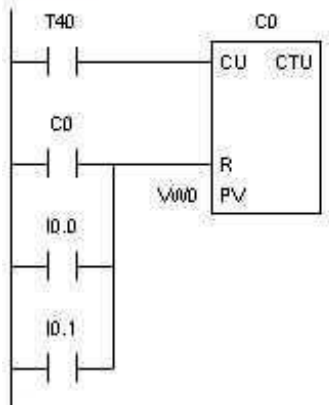
Network 7



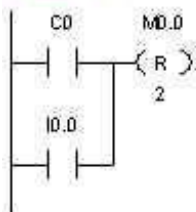
Network 8



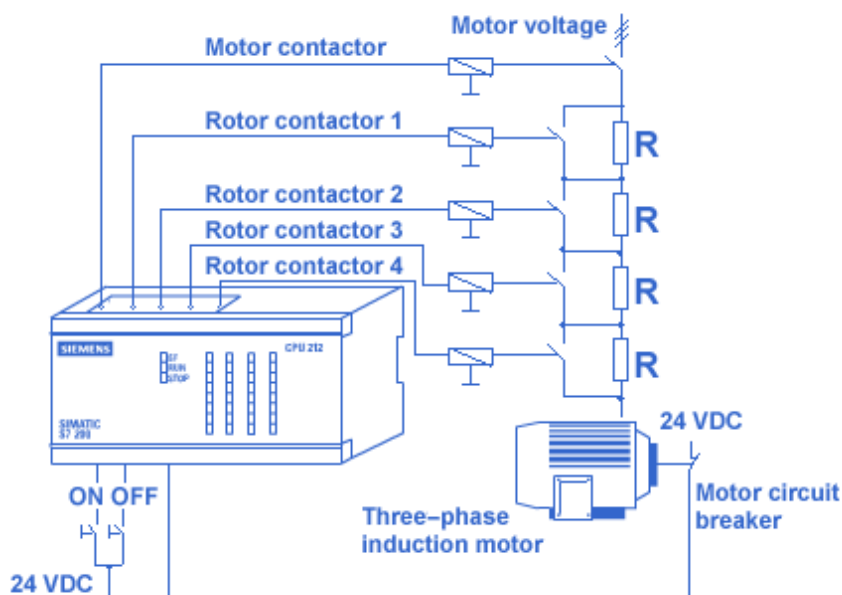
Network 9



Network 10



4.1.7 Điều khiển khởi động động cơ KĐB 3 pha qua 4 cấp điện trở



Yêu cầu công nghệ

Nhấn nút On động cơ khởi động qua 4 cấp điện trở. Sau thời gian đã chỉnh định trước, Contactor 1 đóng loại bỏ điện trở thứ nhất ra khỏi mạch khởi động, lần lượt sau đó tùy thuộc thời gian chỉnh định các contactor đóng lại loại bỏ lần lượt các điện trở còn lại.

Nhấn nút Off động cơ dừng.

Hãy viết chương trình điều khiển khởi động động cơ KĐB 3 pha theo yêu cầu công nghệ trên.

Nhiệm vụ:

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.

Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer và bit nhớ.
- Hiểu và biết cách sử dụng lệnh điều khiển Timer, 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.

4.1.8 Điều khiển 8 động cơ 3 pha chạy tuần tự

Yêu cầu công nghệ

Dây truyền công nghệ sản xuất sản phẩm gồm 8 động cơ hoạt động như sau:

Mỗi lần hoạt động chỉ có 1 động cơ trong số 8 động cơ vận hành, bắt đầu khởi động thì động cơ 1 chạy trước.

- Nếu nhấn nút nhấn PBR thì động cơ đang vận hành ngừng hoạt động và động cơ ở phía bên phải của nó sẽ vận hành.

- Nếu nhấn nút nhấn PBL thì động cơ đang vận hành ngừng hoạt động và động cơ ở phía bên trái của nó sẽ vận hành.

Nút nhấn Off để dừng khẩn cấp.

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.

Ký hiệu	Vào/Ra	Mô tả
Off	I0.0	Nút nhấn dừng động cơ
On	I0.1	Nút nhấn khởi động động cơ
PBR	I0.2	Nút nhấn điều khiển phải
PBL	I0.3	Nút nhấn điều khiển trái
K1	Q0.0	Ngõ ra điều khiển motor 0
K2	Q0.1	Ngõ ra điều khiển motor 1
K3	Q0.2	Ngõ ra điều khiển motor 2
K4	Q0.3	Ngõ ra điều khiển motor 3
K5	Q0.4	Ngõ ra điều khiển motor 4
K6	Q0.5	Ngõ ra điều khiển motor 5
K7	Q0.6	Ngõ ra điều khiển motor 6
K8	Q0.7	Ngõ ra điều khiển motor 7

Mục đích – Yêu cầu:

- Làm quen với chức năng di chuyển, chức năng quay phải và quay trái, lệnh các tiếp điểm đặc biệt và bit nhớ.

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

Các kiến thức cần thiết:

- Xem lại chức năng di chuyển, chức năng quay phải và quay trái, 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, các nút nhấn – công tắc, các cầu dao, khởi động từ và động cơ.

4.1.9 Điều khiển 3 động cơ 3 pha

Yêu cầu công nghệ

Có 3 động cơ A, B, C với chế độ vận hành như sau :

Khi nhấn nút On, động cơ A chạy 3 phút, sau đó động cơ B tự động chạy, 5 phút sau động cơ C chạy.

Khi nhấn nút Off, quá trình dừng xảy ra ngược lại.

Hãy viết chương trình điều khiển 3 động cơ 3 pha theo yêu cầu công nghệ 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.

Ký hiệu	Vào/Ra	Mô tả
Off	I0.0	Nút nhấn dừng động cơ
On	I0.1	Nút nhấn khởi động động cơ
K1	Q0.0	Ngõ ra điều khiển motor A
K2	Q0.1	Ngõ ra điều khiển motor B
K3	Q0.1	Ngõ ra điều khiển motor C

Mục đích – Yêu cầu:

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

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

4.1.10 Điều khiển động cơ xoay chiều 3 pha chọn chu kỳ làm việc 2

Yêu cầu công nghệ

Viết chương trình điều khiển động cơ 3 pha theo yêu cầu hoạt động như sau :

- Chạy thuận 50s
- Ngừng 10s
- Chạy nghịch 50s

Khi cấp điện phải nhấn Start và sau đó nhấn số chu kỳ làm việc của động cơ.

Số chu kỳ có thể chọn là 5, 10, 15, 20, 25, 30. Khi động cơ hoạt động đúng với chu kỳ đặt trước phải nhấn nút Reset rồi mới đặt thông số khác. Nhấn Stop để dừng khẩn cấp.

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.

Ký hiệu	Vào/Ra	Mô tả
Off	I0.0	Nút nhấn dừng khẩn cấp
On	I0.1	Nút nhấn khởi động
5	I1.0	Nút nhấn điều khiển phải
10	I1.1	Nút nhấn điều khiển trái
15	I1.2	Ngõ ra điều khiển motor 0
20	I1.3	Ngõ ra điều khiển motor 1
25	I1.4	Ngõ ra điều khiển motor 2
30	I1.5	Ngõ ra điều khiển motor 3
T	Q0.0	Ngõ ra điều khiển motor 4
N	Q0.1	Ngõ ra điều khiển motor 5

Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, Counter, chức năng dịch chuyển và bit nhớ.

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

Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer, Counter, chức năng dịch chuyển và bit nhớ.

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

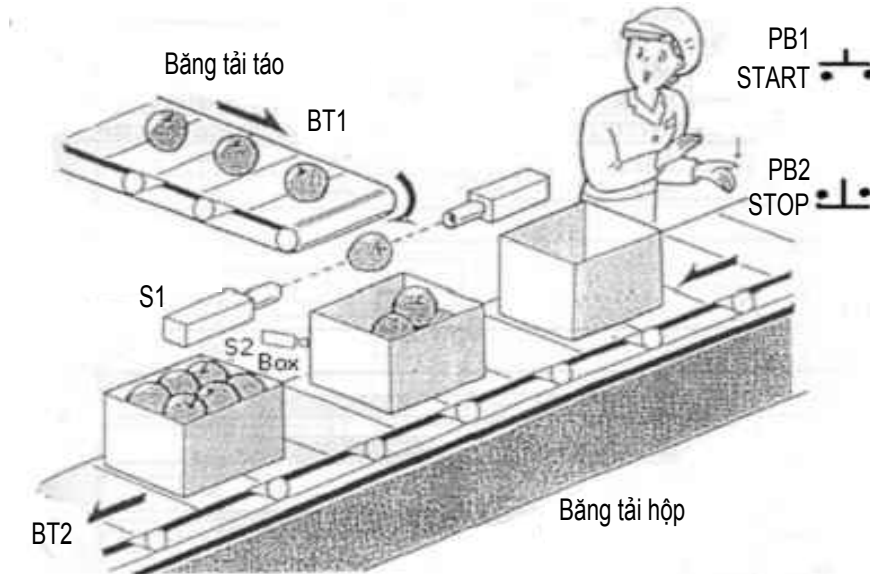
- Cách kết nối tính hiệu vào/ ra của PLC.

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.

4.2 Điều khiển dây chuyền

4.2.1 Điều khiển dây chuyền đóng gói



Yêu cầu công nghệ

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.

Giáo trình
Lập trình điều khiển PLC Nâng cao

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:

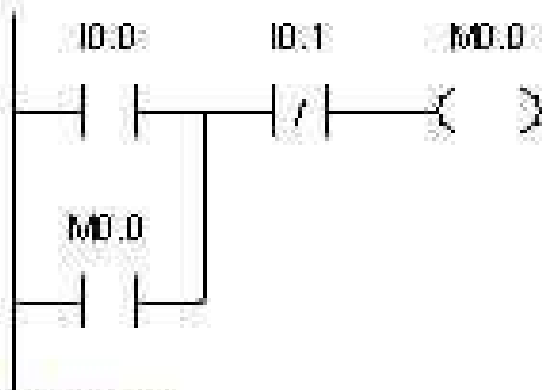
- Xem lại lệnh điều khiển Counter, lệnh điều khiển lưu trình.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

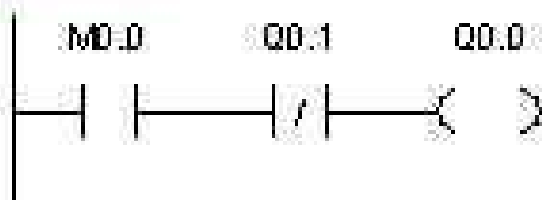
Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn, các công tắc, 2 động cơ, cảm biến.

Ví dụ tham khảo:

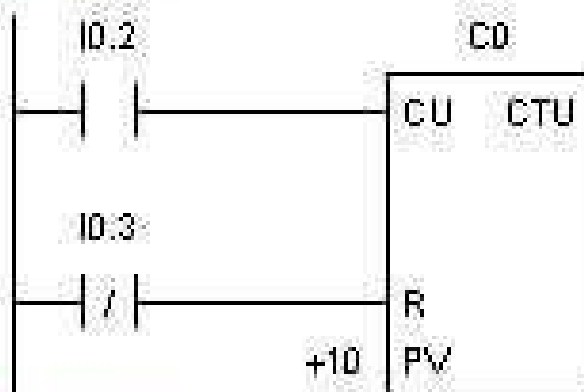
Network 1 // Network Title



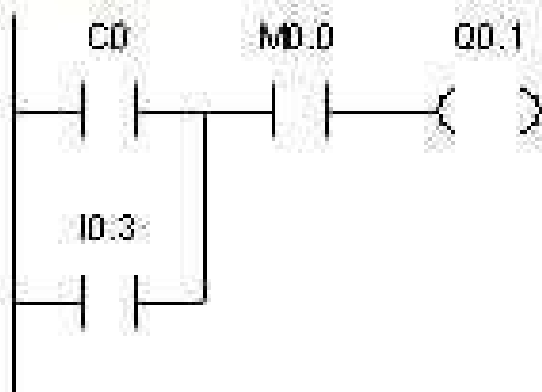
Network 2



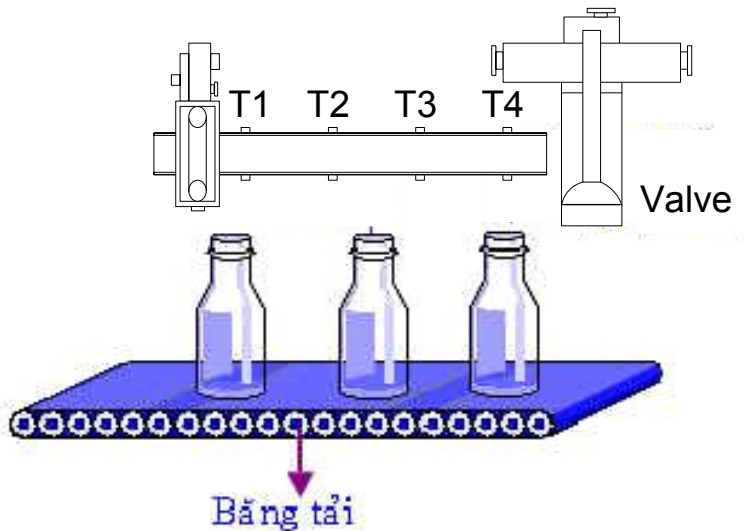
Network 3



Network 4



4.2.2 Điều khiển dây chuyền sản xuất bia



Yêu cầu công nghệ

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.

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.

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:

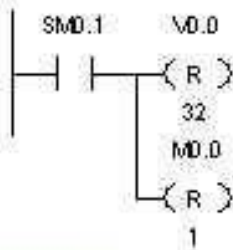
- Xem lại lệnh điều khiển Counter, lệnh điều khiển lưu trình.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tín hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

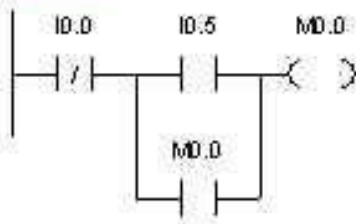
Máy tính, bộ điều khiển lập trình, dây nối, các nút nhấn, 1 Van, 1 động cơ, 4 cảm biến.

Ví dụ tham khảo:

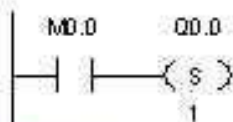
Network 1 // Network Title



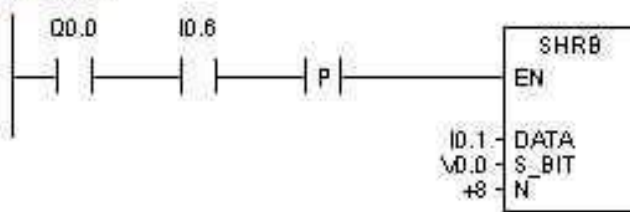
Network 2



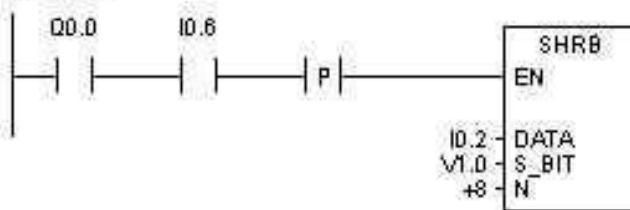
Network 3



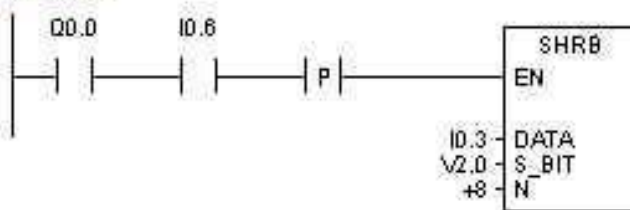
Network 4



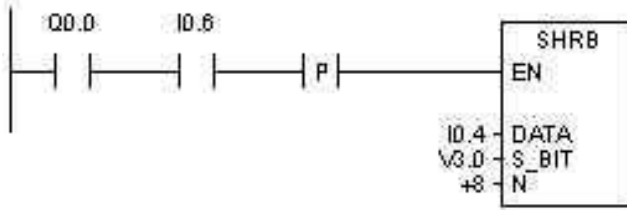
Network 5



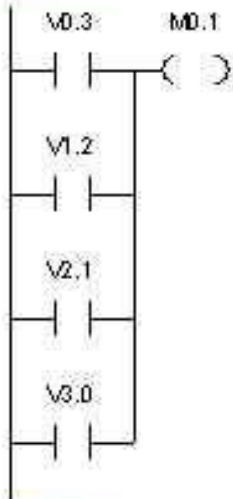
Network 6



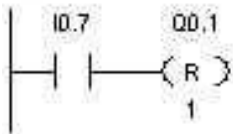
Network 7



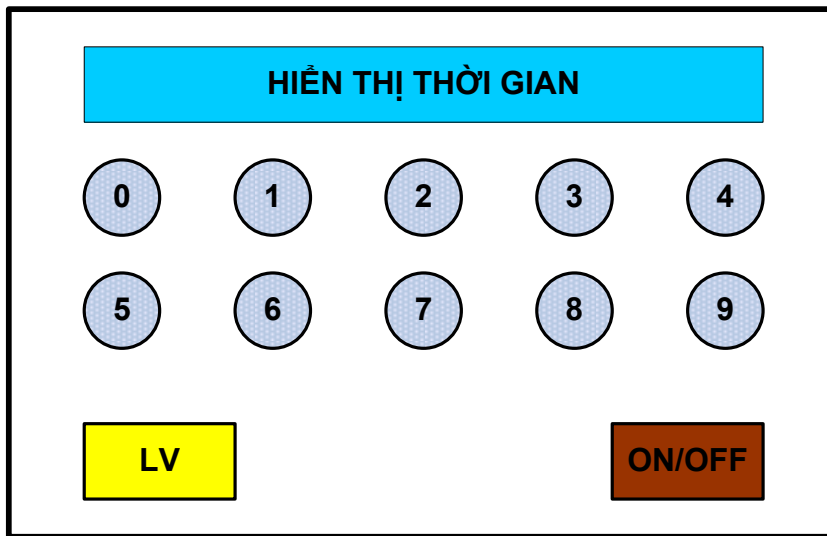
Network 8



Network 9



4.2.3 Điều khiển dây chuyền sấy



Yêu cầu công nghệ

Có một dây chuyền sấy động cơ như sau:

Khi động cơ đã được sơn xong thì được đưa vào buồng sấy, thời gian sấy được đặt trước tùy người sử dụng. Sau khi sấy xong thì đèn báo sáng và ngừng quá trình sấy.

Hãy viết chương trình kích hoạt buồng sấy khi sản phẩm đã được đưa vào buồng và thời gian sấy có thể đặt trước nhờ 10 phím số thập phân từ bên ngoài, giá trị thời gian được hiển thị ở dạng BCD tại ngõ ra (chỉ hiển thị số phút).

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.

Mục đích – Yêu cầu:

- Làm quen với lệnh Timer, lệnh gọi chương trình con, các tiếp điểm đặc biệt, lệnh dịch chuyển và bit nhớ.
- Hiểu và biết cách sử dụng lệnh Timer, lệnh gọi chương trình con, các tiếp điểm đặc biệt, lệnh dịch chuyển và bit nhớ trong quá trình soạn thảo.

Các kiến thức cần thiết:

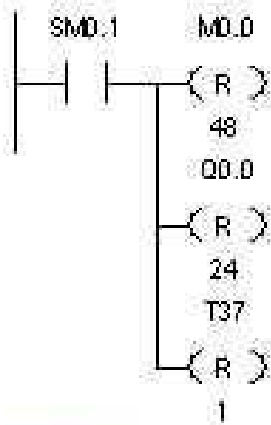
- Xem lại lệnh các tiếp điểm đặc biệt và bit nhớ.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

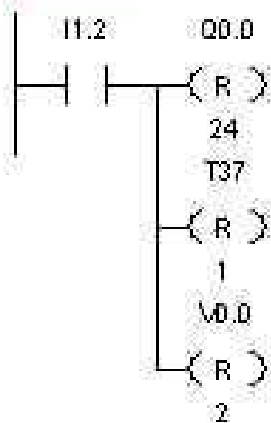
Máy tính, bộ điều khiển lập trình, dây nối, nút nhấn, cảm biến, bóng đèn, động cơ.

Ví dụ tham khảo:

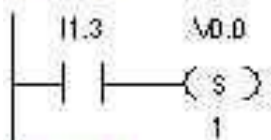
Network 1 // Network Title



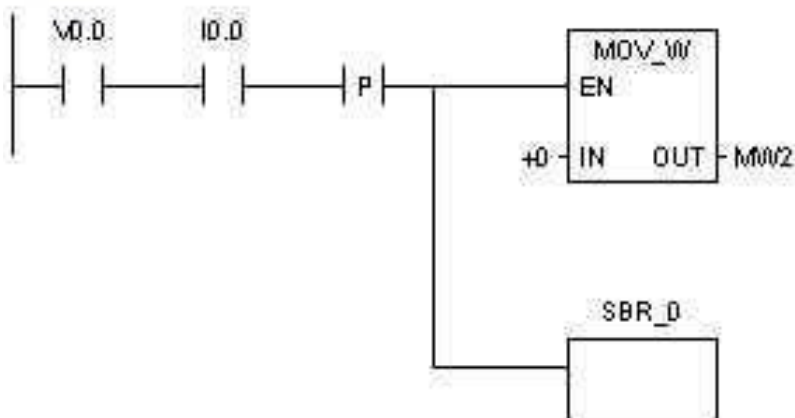
Network 2



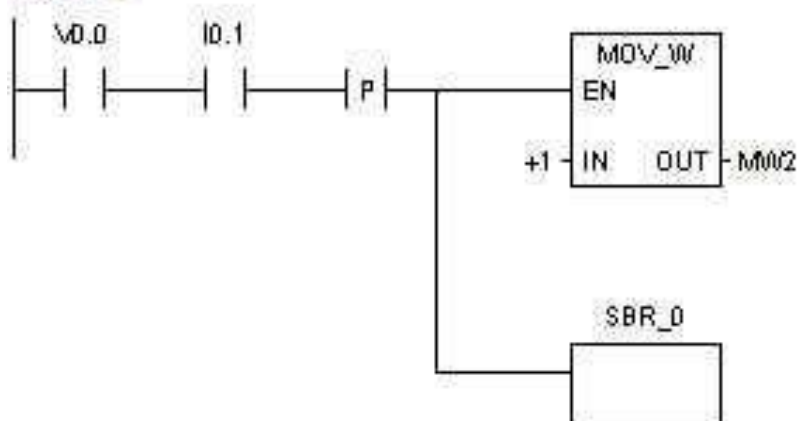
Network 3



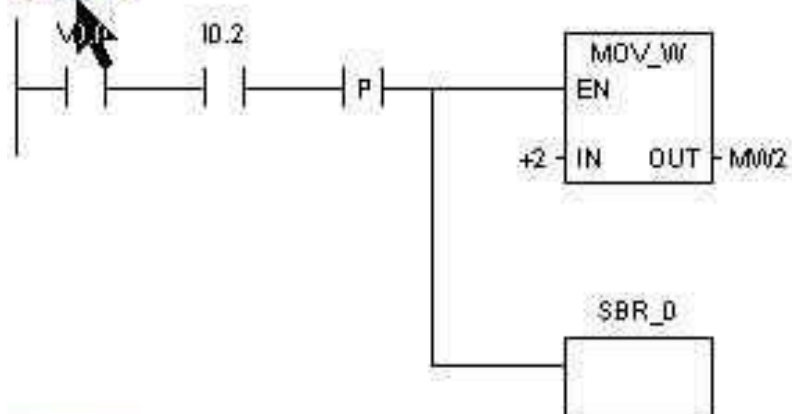
Network 4



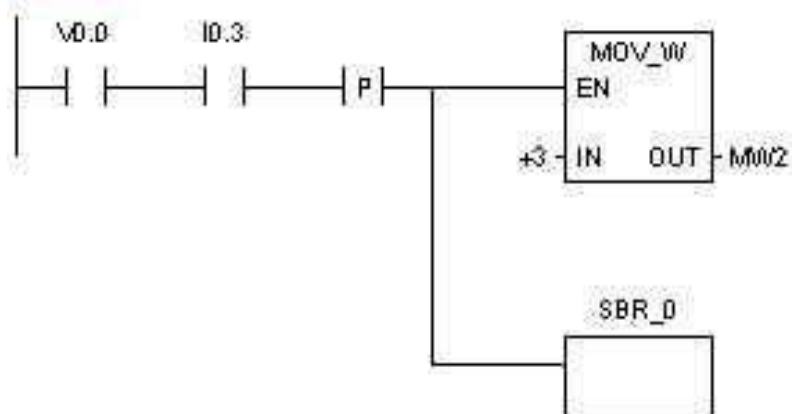
Network 5



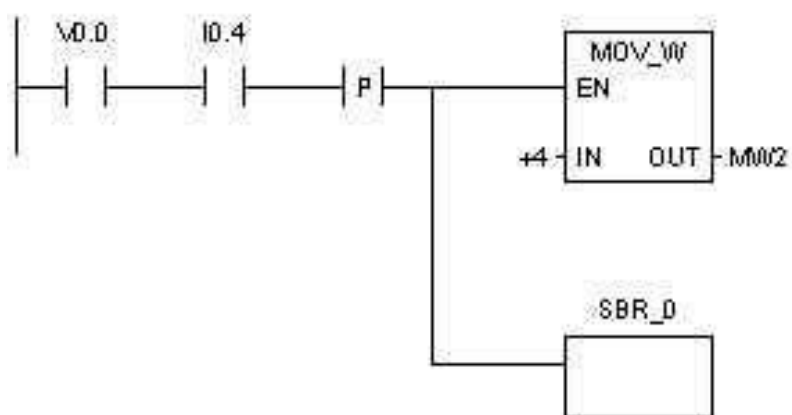
Network 6



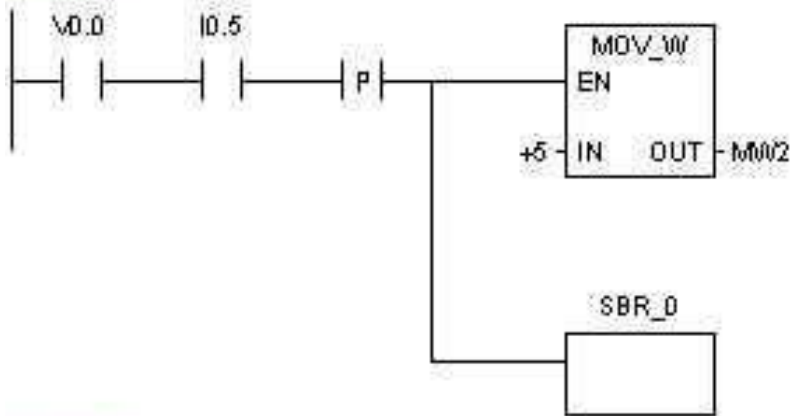
Network 7



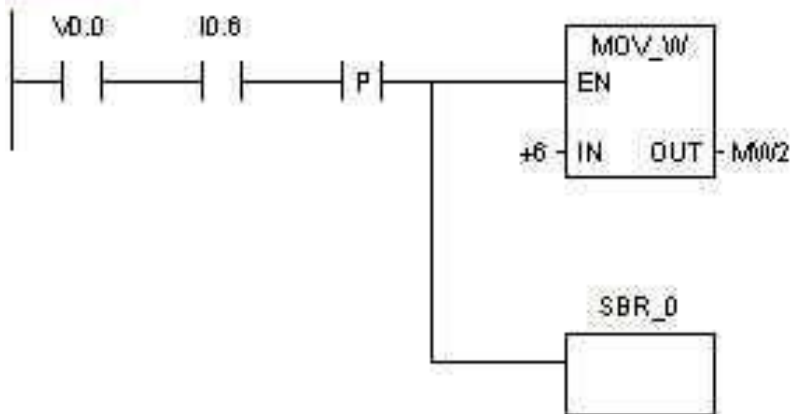
Network 8



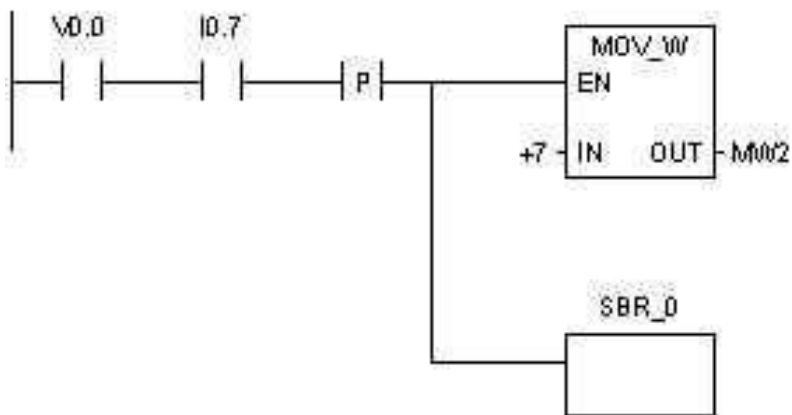
Network 9



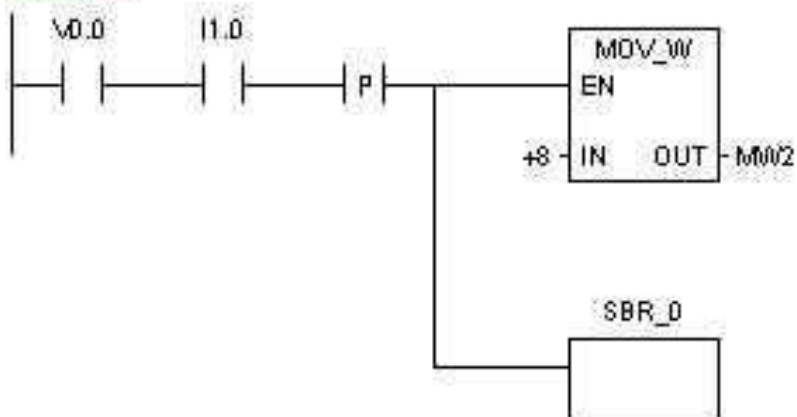
Network 10

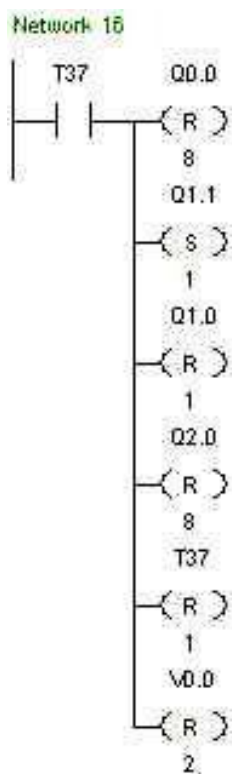
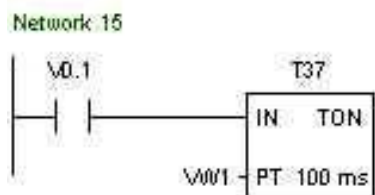
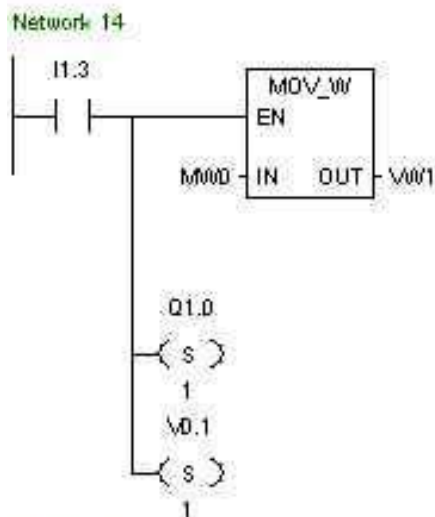
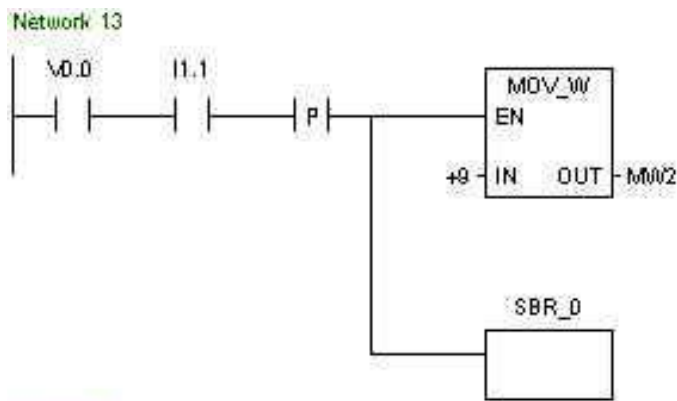


Network 11

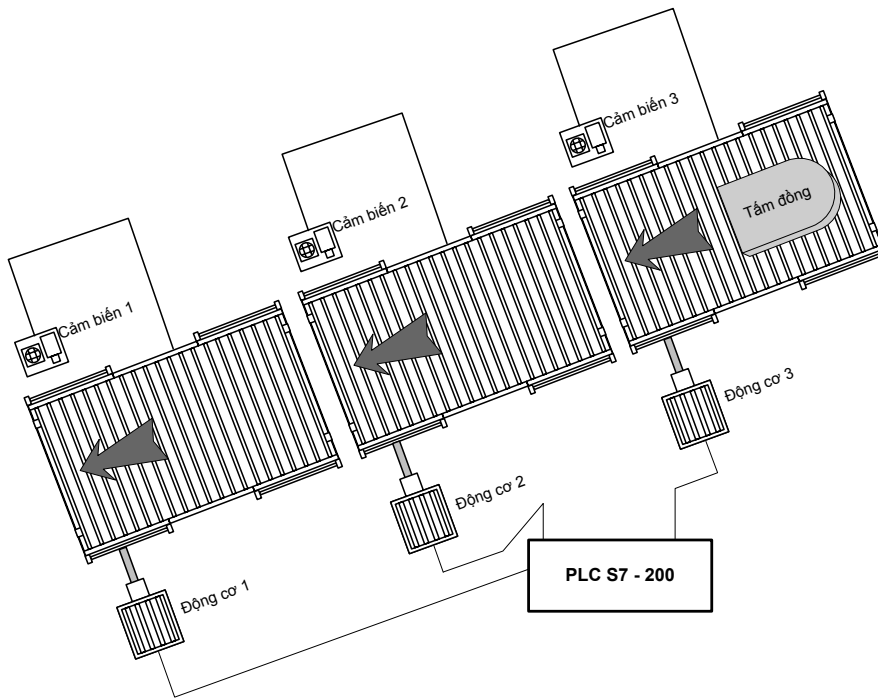


Network 12





4.2.4 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.

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.

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.

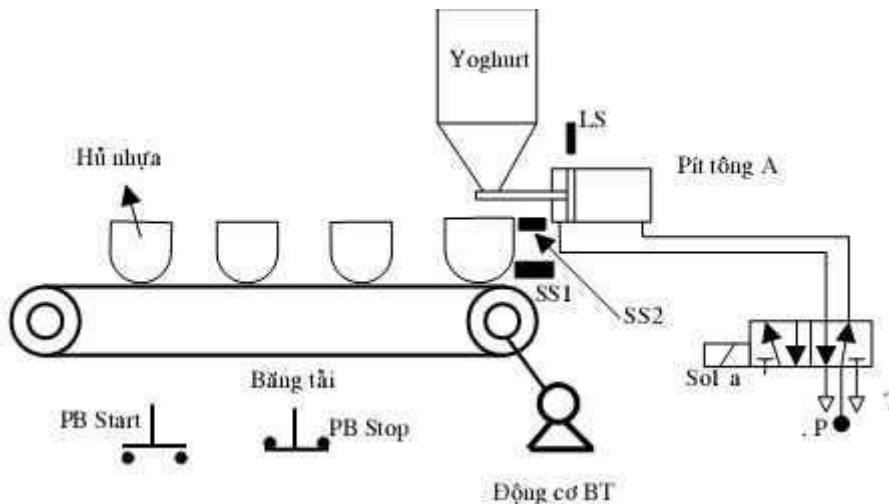
Các kiến thức cần thiết:

- 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.

4.2.5 Điều khiển dây chuyền chiết Yoghurt vào hũ



Yêu cầu công nghệ

Nhấn nút PB Start thì băng tải mang các hũ nhựa dịch chuyển, khi gặp cảm biến nhận được hũ (SS1) thì băng tải dừng lại. Pít tông A dịch chuyển qua phải và Yoghurt được chiết vào hũ, thể tích được xác định bằng cảm biến điện dung (SS2). Sau khi chiết đủ pít tông I dịch chuyển qua trái để ngưng chiết. Khi gặp LS (giới hạn từ trường) thì băng tải lại chuyển động và một hũ mới sẽ được chiết.

- Nhấn nút PB Stop thì hệ thống ngừng hoạt động.

Chú ý: Van khí là loại 5/2/ 1 side. Bình thường có khí nén mà không có tín hiệu điện thì pít tông nằm phía trái đóng miệng phễu chứa Yoghurt. Khi có tín hiệu điện tác động vào cuộn coil thì pít tông sẽ dịch chuyển qua phải làm mở miệng phễu.

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.

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.

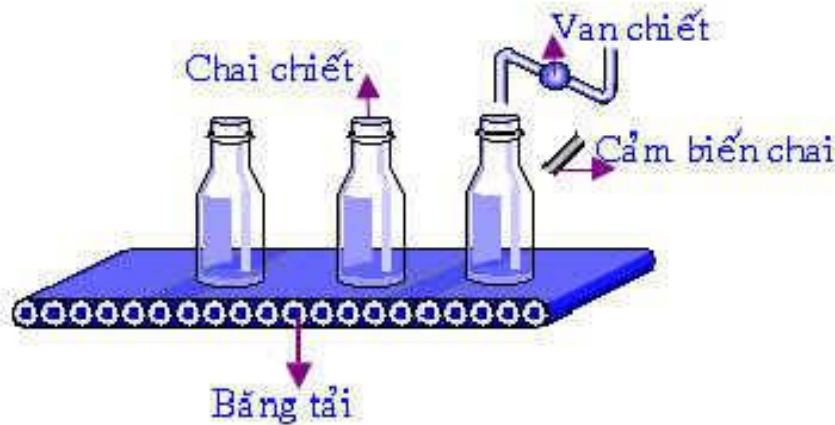
Các kiến thức cần thiết:

- 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 tín hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, 01 động cơ, các công tắc ht, cảm biến, pít tông.

4.2.6 Điều khiển dây chuyền chiết nước vào chai



Yêu cầu công nghệ

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.

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.

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 tín 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.

4.3 Các dạng điều khiển khác

4.3.1 Điều khiển bãi đỗ xe



Yêu cầu công nghệ

Hệ thống điều khiển bãi đậu xe chứa tối đa là 12 chiếc mô tả ở hình trên. Mỗi lần xe vào, PLC tự động tăng thêm 1 bởi cảm biến phát hiện xe S1. Bất kỳ một chiếc xe nào đi ra khỏi bãi, PLC sẽ tự động giảm đi 1 bởi cảm biến phát hiện S2. Khi 12 chiếc xe được đăng ký, bảng hiệu đầy xe sẽ được sáng lên thông báo đến các xe không được vào nữa.

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.

Mục đích – Yêu cầu:

- Làm quen với lệnh Counter, lệnh các tiếp điểm đặc biệt và bit nhớ.
- Hiểu và biết cách sử dụng lệnh lệnh Counter, lệnh các tiếp điểm đặc biệt và bit nhớ trong quá trình soạn thảo.

Các kiến thức cần thiết:

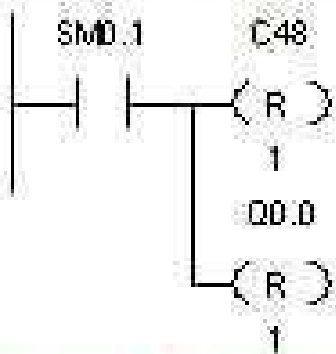
- Xem lại lệnh Counter, lệnh các tiếp điểm đặc biệt và bit nhớ.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

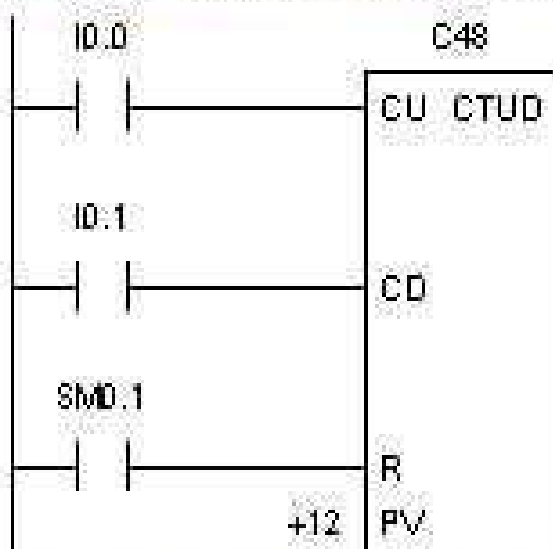
Máy tính, bộ điều khiển lập trình, dây nối, cảm biến, bóng đèn.

Ví dụ tham khảo:

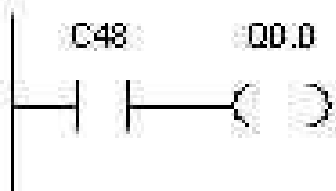
Network 1 // Xóa Counter



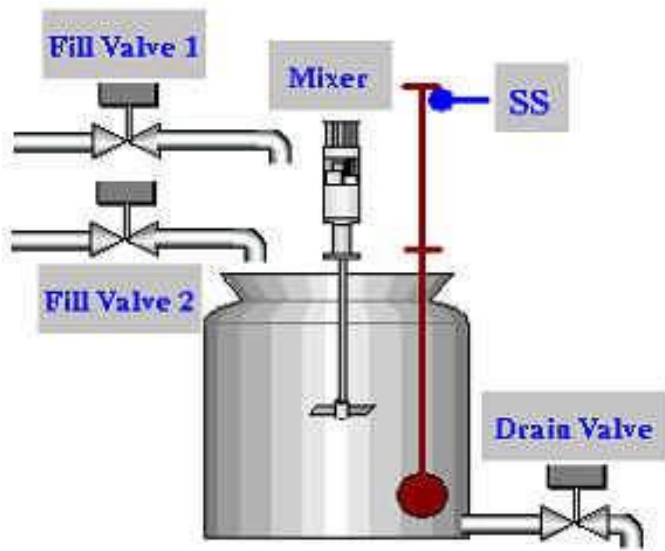
Network 2 // Đếm số lượng xe cơ trong bãi đỗ xe



Network 3 // Đèn báo hiệu bãi đỗ đầy xe



4.3.2 Điều khiển trộn sơn theo thời gian



Yêu cầu công nghệ

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

Nhấn nút On hai Fill Valve 1 và 2 mở ra ch 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. Quy trình lặp lại 05 lần rồi dừng.

Nhấn nút Off, hết sơn mới được dừng.

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.

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.

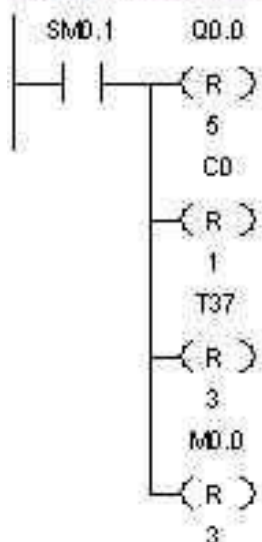
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

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

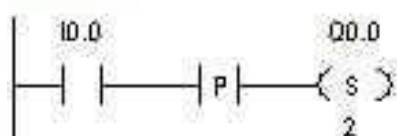
Dụng cụ và thiết bị:

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

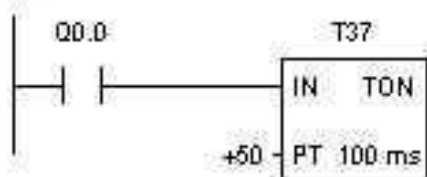
Network 1 // Network Title



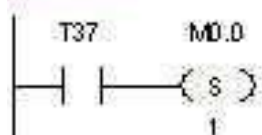
Network 2



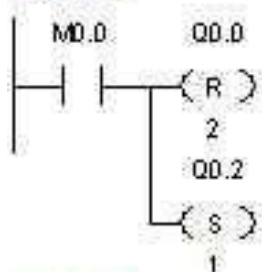
Network 3



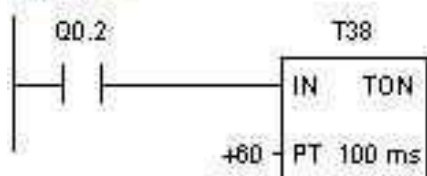
Network 4



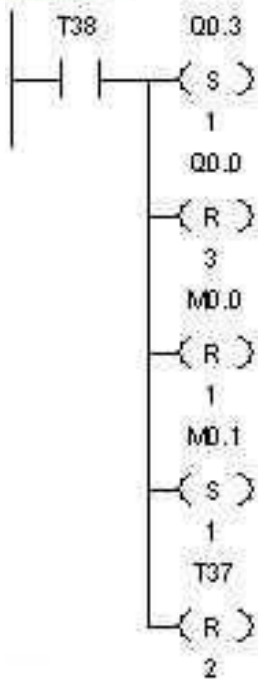
Network 5



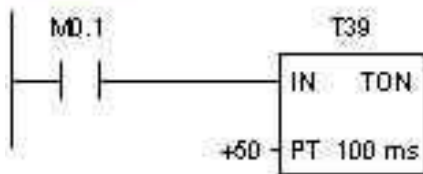
Network 6



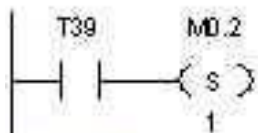
Network 7



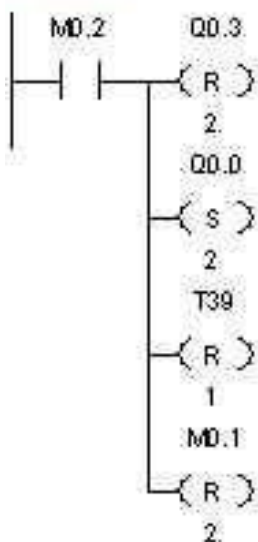
Network 8

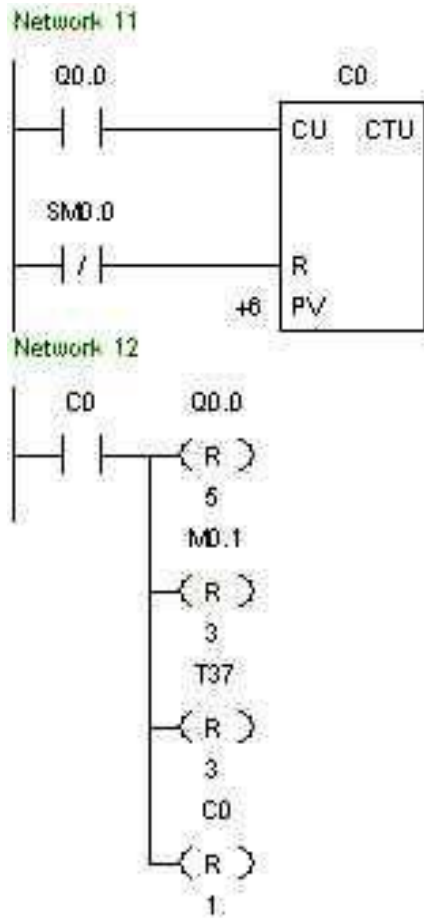


Network 9

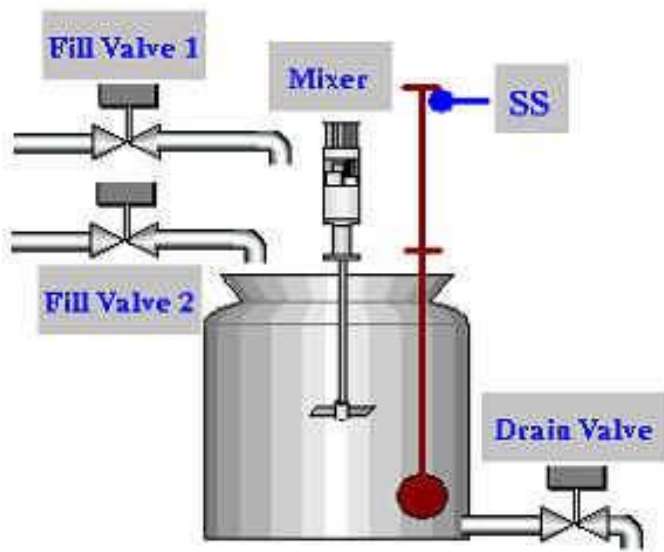


Network 10





4.3.3 Điều khiển trộn sơn theo mức



Yêu cầu công nghệ

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.

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

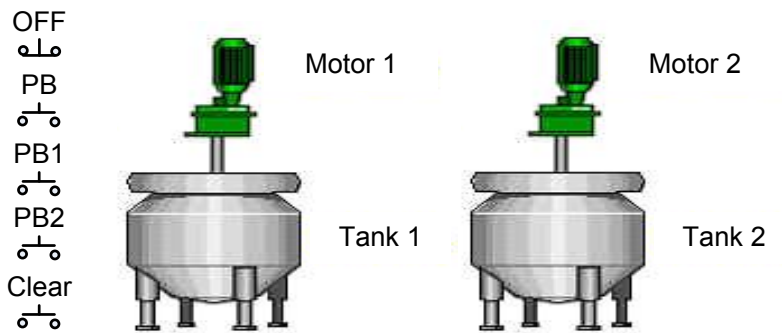
Các kiến thức cần thiết:

- 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.

4.3.4 Điều khiển trộn hóa chất



Yêu cầu công nghệ

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 hỏng 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.

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, 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.

4.3.5 Điều khiển máy bán hàng tự động



Yêu cầu công nghệ

Có một máy bán nước tự động đơn giản, tùy thuộc số tiền ta đưa vào trong máy thì loại nước uống tương ứng sẽ được đưa ra. Tiền đưa vào phải tương đương hoặc lớn hơn giá tiền qui định cho từng sản phẩm. A, B, C, D : 8 đồng; E : 4 đồng; F : 2 đồng.

Máy bán hàng tự động này làm việc với tải nhẹ nên chủ yếu sử dụng các động cơ khí nén.

Hãy viết chương trình điều khiển máy bán nước tự động 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.

Mục đích – Yêu cầu:

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

- Hiểu và biết cách sử dụng lệnh điều khiển Timer, Counter chức năng di chuyển, lệnh các tiếp điểm đặc biệt, lệnh cộng trừ 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.

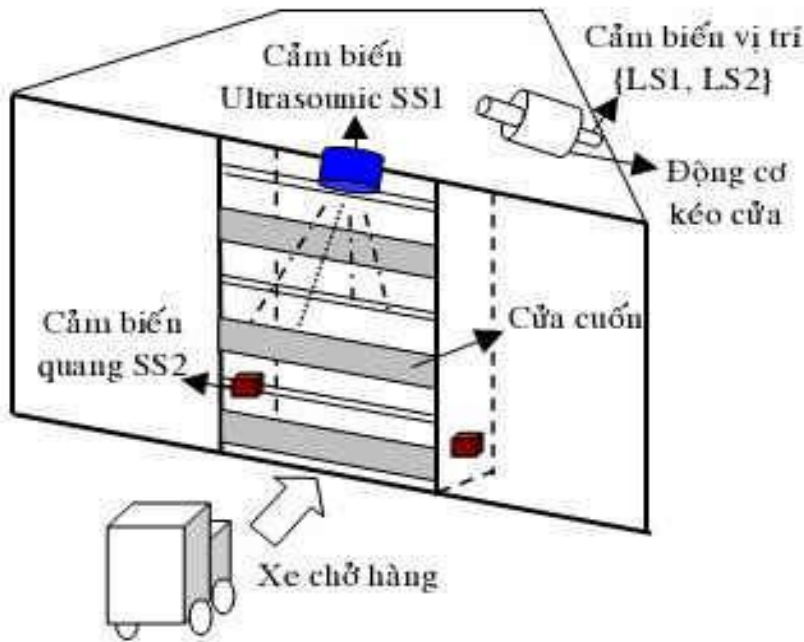
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.

- 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.

4.3.6 Điều khiển đóng mở cửa Garage ô tô



Yêu cầu công nghệ

Khi xe đang tiến về gần cửa kho, cảm biến Ultrasonic SS1 nhận dạng được xe và cửa sẽ được mở ra đến gần giới hạn hành trình trên LS2 thì cửa dừng lại rồi xe chạy vào. Khi cảm biến quang SS2 đặt phía trong cổng cửa nhận dạng được xe đã đi qua khỏi cửa thì cửa sẽ được đóng lại, chạm vào giới hạn hành trình dưới LS1 thì cửa dừng lại. Chú ý xe chỉ đi một chiều.

Hãy viết chương trình điều khiển đóng mở cửa Garage ô tô 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.

Mục đích – Yêu cầu:

- Làm quen với các lệnh các tiếp điểm đặc biệt và bit nhớ.
- Hiểu và biết cách sử dụng lệnh các tiếp điểm đặc biệt và bit nhớ trong quá trình soạn thảo.

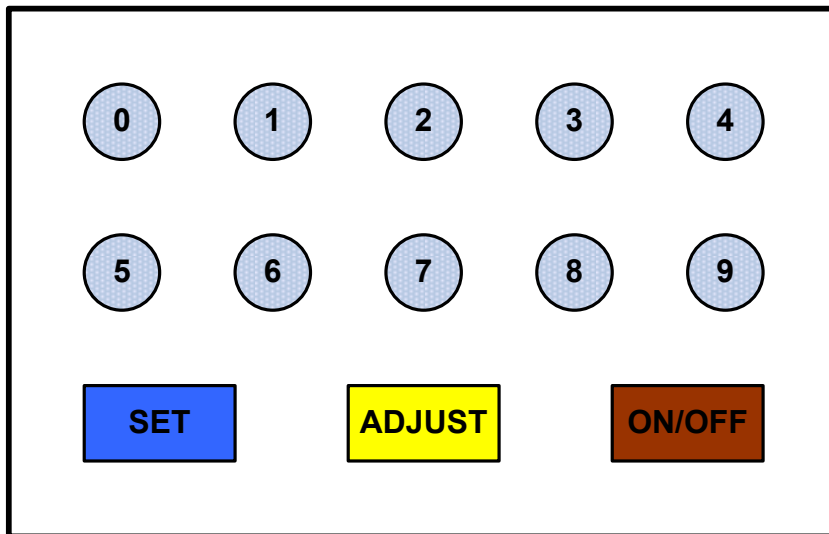
Các kiến thức cần thiết:

- Xem lại 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, công tắc hành trình.

4.3.7 Điều khiển bộ đếm sản phẩm



Yêu cầu công nghệ

Viết chương trình cài đặt trước giá trị cho một bộ đếm sản phẩm mà gói từ 0 đến 9999 bằng các nút nhấn của các đầu vào từ I0.0 đến I1.1 (tương ứng các số từ 0 đến 9), giá trị đặt này được hiển thị trên các ngõ ra (Led 7 đoạn). Khi đặt xong ta nhấn nút Set thì việc đặt sẽ hoàn thành. Khi đang đếm muốn sửa lại giá trị đặt thì ta nhấn nút chỉnh định Adjust.

Hãy viết chương trình điều khiển cho bộ đếm sản phẩm 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.

Mục đích – Yêu cầu:

- Làm quen với các lệnh các tiếp điểm đặc biệt, lệnh dịch chuyển, hàm đổi dữ liệu tương ứng thanh ghi 7 đoạn và bit nhớ.
- Hiểu và biết cách sử dụng lệnh các tiếp điểm đặc biệt, lệnh dịch chuyển, hàm đổi dữ liệu tương ứng thanh ghi 7 đoạn và bit nhớ trong quá trình soạn thảo.

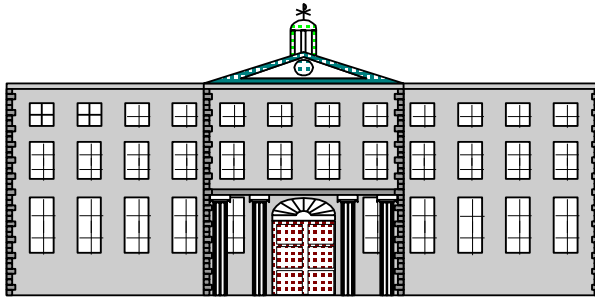
Các kiến thức cần thiết:

- Xem lại 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.

4.3.8 Điều khiển chuông báo tiết học



Yêu cầu công nghệ

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

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

Hãy viết chương trình điều khiển chuông báo tiết học 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.

Mục đích – Yêu cầu:

- Làm quen với lệnh truy xuất thời gian thực, lệnh so sánh.
- Hiểu và biết cách sử dụng lệnh truy xuất thời gian thực, lệnh so sánh trong quá trình soạn thảo.

Các kiến thức cần thiết:

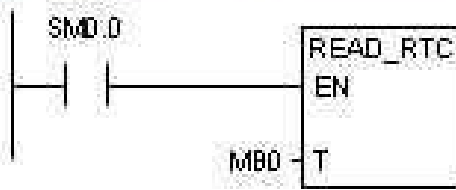
- Xem lại lệnh truy xuất thời gian thực, lệnh so sánh.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tính hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

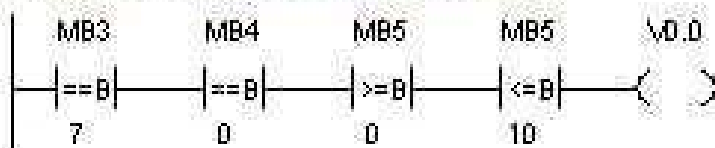
Máy tính, bộ điều khiển lập trình, dây nối, chuông, bóng đèn.

Ví dụ tham khảo:

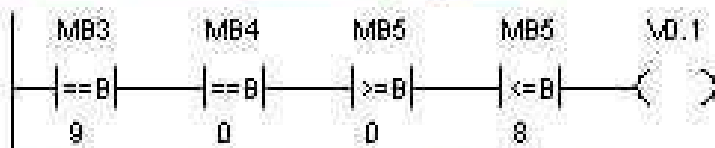
Network 1 // Đóng hồ thời gian thực



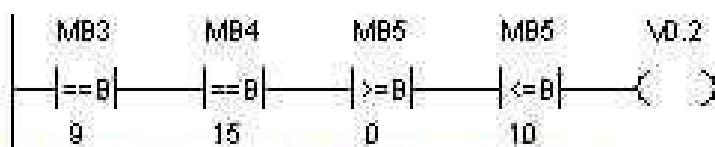
Network 2 // Chương kêu báo vào tiết học 7h



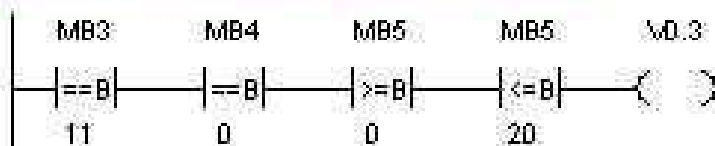
Network 3 // Chương kêu báo giờ giải lao 9h



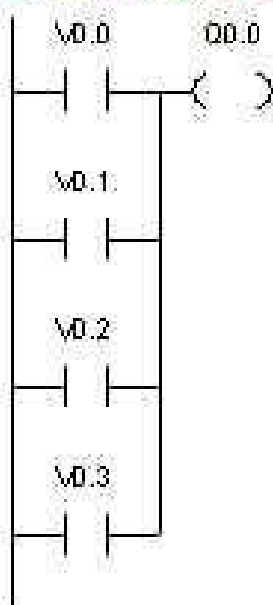
Network 4 // Chương kêu báo vào tiết học 9h15



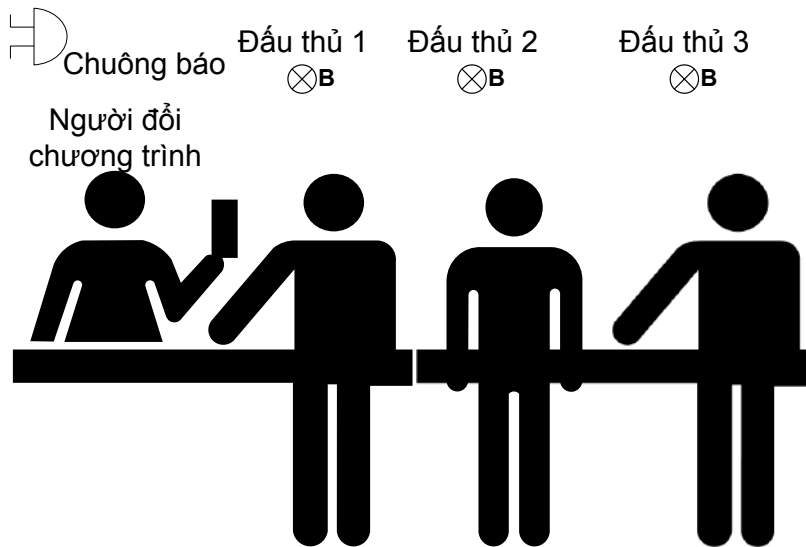
Network 5 // Chương kêu Báo buổi học kết thúc



Network 8 // Ngo ra điều khiển chương



4.3.9 Điều khiển chương trình trò chơi dạng Đường lên đỉnh Olympia”



Yêu cầu công nghệ

Sau khi người dẫn chương trình đã nêu xong các câu hỏi, các đấu thủ sẽ bấm nút phía trước mặt để trả lời câu hỏi. Ai bấm trước trả lời trước. Chuông kêu trong 10 s sau khi đấu thủ bất kỳ nào bấm nút, cùng lúc đó đèn trước mặt đấu thủ đó sẽ sáng và nó sẽ chỉ được tắt bởi người dẫn chương trình.

Hãy viết chương trình điều khiển trò chơi 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.

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.

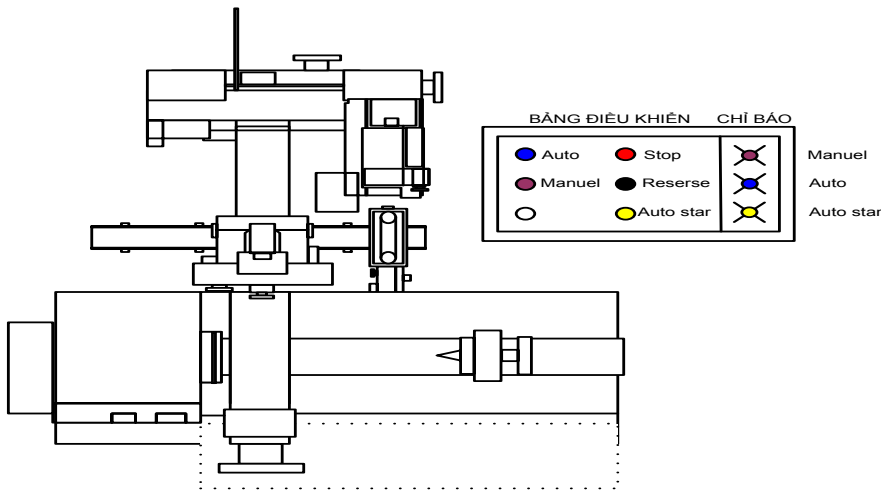
Các kiến thức cần thiết:

- 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 đèn, 01 chuông báo.

4.3.10 Điều khiển thao tác máy khoan



Yêu cầu công nghệ

1. Quy trình hoạt động của nhân công (bằng tay):

Khi nhấn nút quay thuận SW (Forward Switch), động cơ quay theo chiều thuận. Nó có thể được ngưng lại bằng cách nhấn nút dừng S2 (Stop Switch). Khi máy chạm công tắc hành trình LS2, động cơ sẽ dừng lại.

Khi nhấn nút quay nghịch RW (Reverse Switch), động cơ quay nghịch. Nó có thể được ngưng lại bằng cách nhấn nút dừng S2 (Stop Switch). Khi máy chạm công tắc hành trình LS1, động cơ sẽ dừng lại.

2. Chu kỳ tự động:

Khi nhấn nút PB (Auto Star) và công tắc hành trình LS1 ở trạng thái On, động cơ chạy theo chiều thuận cho đến khi chạm vào công tắc hành trình LS2. Bộ định thời (Timer) bắt đầu đếm thời gian. Động cơ sẽ chuyển động ngược lại khi bộ định thời đếm đến 2s. Khi thay đổi trạng thái của công tắc hành trình LS1 chu kỳ lại tiếp tục.

Nhiệm vụ:

- 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.

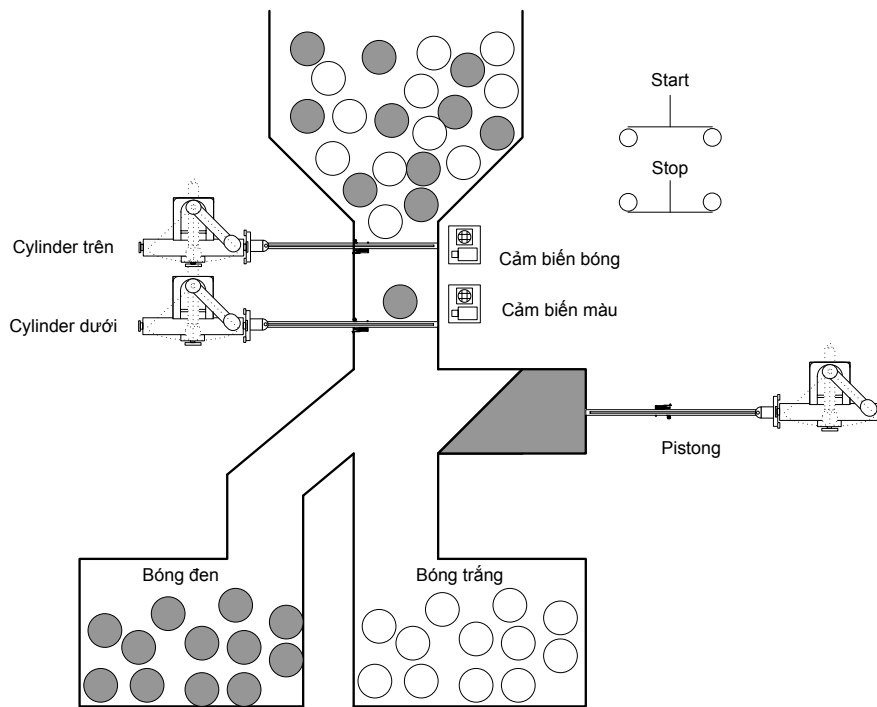
Các kiến thức cần thiết:

- 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, động cơ, nút nhấn, công tắc hành trình.

4.3.11 Điều khiển cơ cấu máy lựa sản phẩm



Yêu cầu công nghệ

Trong ứng dụng này, hệ thống sẽ lựa ra sản phẩm thiếu chất lượng trộn lẫn trong sản phẩm tiêu chuẩn. Ví dụ phân biệt bóng màu đen và bóng màu trắng đưa vào 2 container khác nhau.

Khi bắt đầu hoạt động, sensor S1 sẽ cảm nhận sự hiện diện của bóng ở trong hộp. Cuộn Solenoid ở trên thả ra cho sensor S2 phân biệt màu trước khi thả vào container. Nếu là bóng trắng thì Solenoid dưới sẽ thả trực tiếp xuống container. Nếu là bóng màu đen, piston sẽ được kích hoạt trước khi Solenoid dưới mở ra và chuyển hướng bóng xuống container khác.

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.

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.

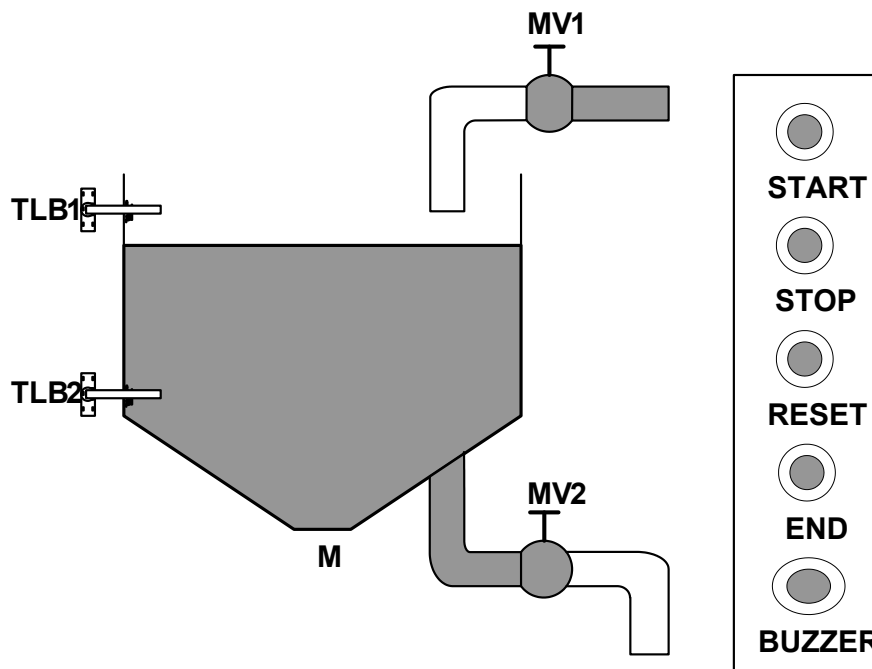
Các kiến thức cần thiết:

- 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, pistong, cảm biến.

4.3.12 Điều khiển điều dẫn nước ra hay đổ nước vào



Yêu cầu công nghệ

Khi nút nhấn bắt đầu PB1 (Start button) được nhấn, van cung cấp nước MV1 (Water valve) mở, nước bắt đầu được đổ vào thùng. Trong thời gian này động cơ khuấy nước M bắt đầu hoạt động.

Khi mức nước qua cảm biến báo mức nước thấp TBL2 và tiến đến cảm biến báo mức nước cao TBL1 van cung cấp nước MV1 đóng và động cơ khuấy nước M đứng lại.

Sau đó, van dẫn nước ra MV2 (Drain valve) mở và bắt đầu dẫn nước ra ngoài khi mức nước xuống dưới cảm biến báo mức nước thấp TBL2, van dẫn nước ra MV2 đóng lại.

Chu kỳ hoạt động 4 lần quá trình hoạt động kết thúc.

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.

Mục đích – Yêu cầu:

- Làm quen với lệnh điều khiển Timer, Counter lệnh tích phân.
- Hiểu và biết cách sử dụng lệnh điều khiển Timer, Counter lệnh tích phân trong quá trình soạn thảo.

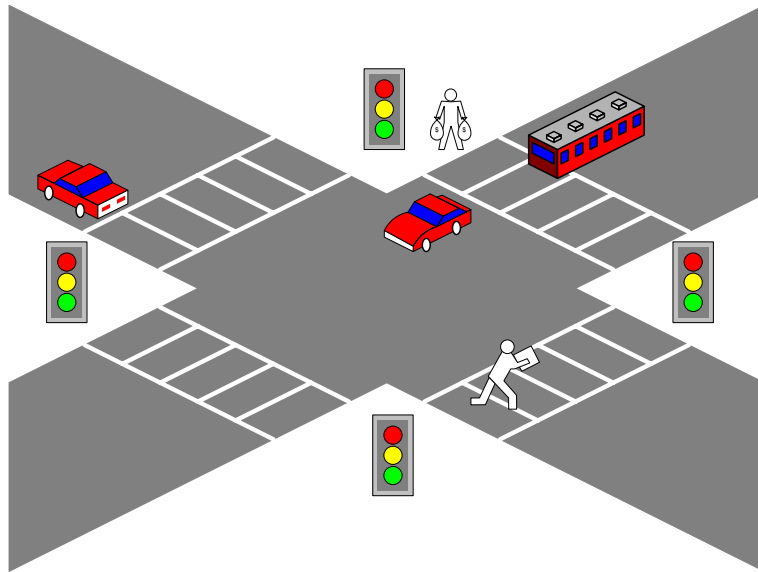
Các kiến thức cần thiết:

- Xem lại lệnh điều khiển Timer, Counter lệnh tích phân về điều khiển lưu trình.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tín hiệu vào/ ra của PLC.

Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, 1 động cơ, cảm biến, đèn báo, van.

4.3.13 Điều khiển tín hiệu đèn giao thông



Yêu cầu công nghệ

Viết chương trình điều khiển đèn giao thông ở ngã tư với các yêu cầu sau:

Từ 5h đến 22h các đèn hoạt động theo thời gian sau:

- Đèn đỏ sáng trong khoảng thời gian 30s
- Đèn xanh sáng trong khoảng thời gian 25s
- Đèn vàng sáng trong khoảng thời gian 5s

Từ 22h đến 5h chỉ có đèn vàng hoạt động với chu kỳ chớp tắt là một giây.

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.

Mục đích – Yêu cầu:

- Làm quen với đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh.
- Hiểu và biết cách sử dụng đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh trong quá trình soạn thảo.

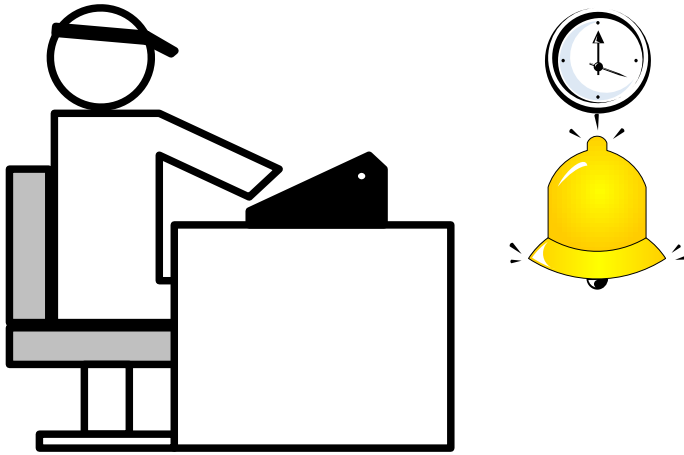
Các kiến thức cần thiết:

- Xem lại đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh.
- Xem lại cách sử dụng các lệnh vào/ra, lệnh ghi/xóa giá trị tiếp điểm.
- Cách kết nối tín hiệu vào/ra của PLC.

Dụng cụ và thiết bị:

Máy tính, bộ điều khiển lập trình, dây nối, 06 đèn báo.

4.3.14 Điều khiển báo giờ làm việc tan tầm



Yêu cầu công nghệ

Một xí nghiệp được trang bị một hệ thống tự động báo giờ làm việc và tan tầm như sau:

- **Giờ làm việc:** Đúng 7h00 sáng thì một hồi chuông vang lên kéo dài trong 1 phút báo hiệu giờ làm việc bắt đầu.
- **Giờ tan tầm:** Đúng 17h00 thì một hồi chuông vang lên kéo dài trong 1 phút báo hiệu đã hết giờ làm việc.

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.

Mục đích – Yêu cầu:

- Làm quen với đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh.
- Hiểu và biết cách sử dụng đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh trong quá trình soạn thảo.

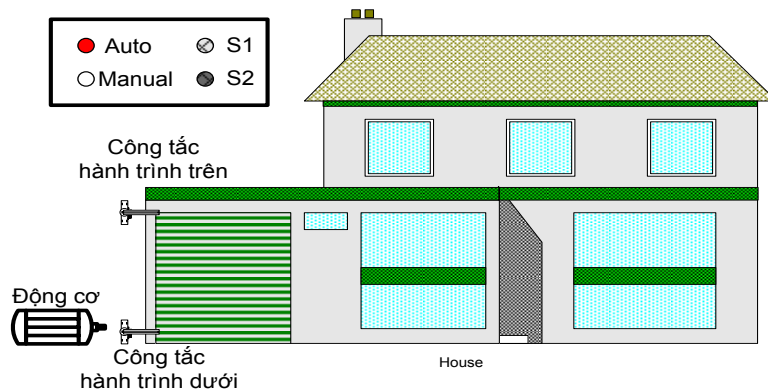
Các kiến thức cần thiết:

- Xem lại đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh.
- 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, chuông báo.

4.3.15 Điều khiển cửa cuốn



Yêu cầu công nghệ

Một cửa cuốn được kéo bởi động cơ M1. Cửa cuốn này có thể hoạt động ở 2 chế độ tay và tự động.

Ở chế độ tay:

- Khi nhấn nút nâng S1 thì động cơ sẽ kéo cửa cuốn lên, khi gặp công tắc hành trình trên S3 thì cửa cuốn dừng lại.
- Khi nhấn nút hạ S2 thì động cơ sẽ kéo cửa cuốn xuống, khi gặp công tắc hành trình dưới S4 thì cửa cuốn dừng lại.

Trong quá trình nâng hay hạ cửa cuốn sẽ được dừng khi nút nhấn dừng S0 được ấn.

Ở chế độ tự động: - Sáng: đúng 7h00 cửa cuốn tự động kéo lên - Chiều: đúng 17h00 cửa cuốn tự động đóng.

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.

Mục đích – Yêu cầu:

- Làm quen với đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh.
- Hiểu và biết cách sử dụng đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh. trong quá trình soạn thảo.

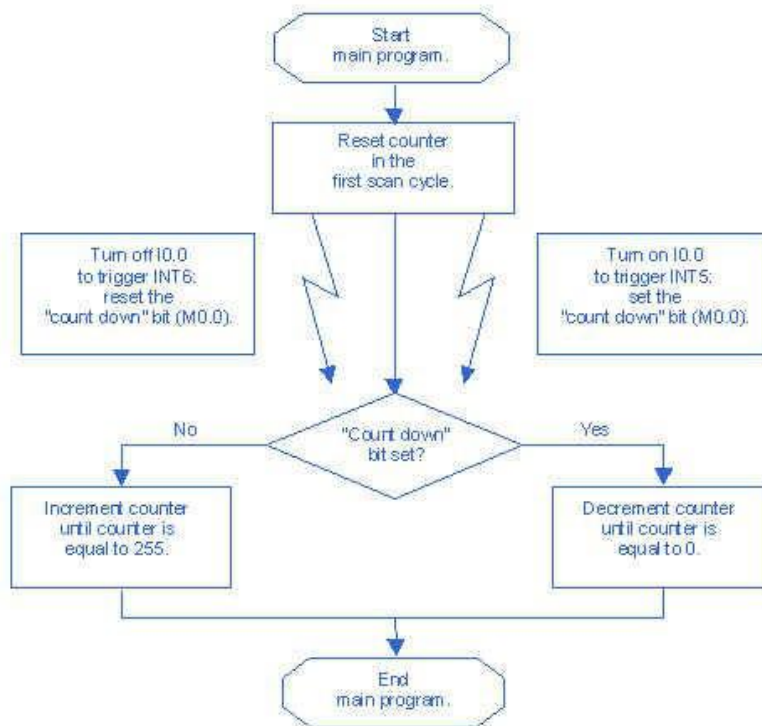
Các kiến thức cần thiết:

- Xem lại đồng hồ thời gian thực, lệnh Counter, lệnh gọi chương trình con, lệnh so sánh.
- 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, động cơ, nút nhấn, công tắc hành trình.

4.3.16 Chương trình đếm từ 0 đến 255



Yêu cầu công nghệ

Dựa vào lưu đồ trên hãy viết chương trình đếm từ 0 đến 255, phụ thuộc vào ngõ vào I0.0 (công tắc). Nếu ngõ vào I0.0 ở trạng thái logic 0, chương trình đếm lên. Nếu ngõ vào I0.0 ở trạng thái logic 1, chương trình đếm xuống. Ngay khi ngõ vào chuyển trạng thái, một thủ tục ngắt các ngõ vào/ra được thúc đẩy, chương trình đếm lên/xuống tương ứng được khởi động.

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.

Mục đích – Yêu cầu:

- Làm quen các tiếp điểm đặc biệt, chức năng dịch chuyển, chức năng toán học và chức năng so sánh, thủ tục ngắt.
- Hiểu và biết cách sử dụng các tiếp điểm đặc biệt, chức năng dịch chuyển, chức năng toán học và chức năng so sánh, thủ tục ngắt trong quá trình soạn thảo.

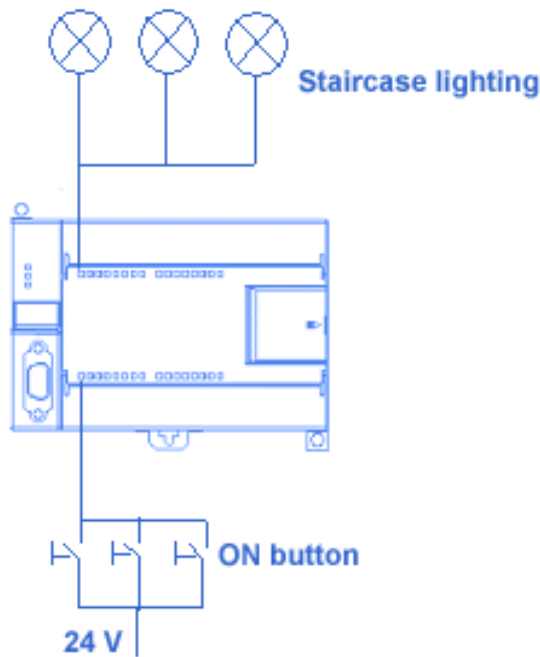
Các kiến thức cần thiết:

- Xem lại các tiếp điểm đặc biệt, chức năng dịch chuyển, chức năng toán học và chức năng so sánh, thủ tục ngắt.
- 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, bóng đèn.

4.3.17 Điều khiển đèn cầu thang – hành lang



Yêu cầu công nghệ

Viết chương trình điều khiển đèn cầu thang – hành lang với yêu cầu sau.

Khi nhấn nút On, đèn sáng trong khoảng thời gian 30s.

Lưu ý: đèn sẽ sáng trong khoảng thời gian 30s kể từ khi nút On được nhấn (bất kỳ lúc nào, kể cả khi nó đã được nhấn lần trước).

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.

Mục đích – Yêu cầu:

- Làm quen các tiếp điểm đặc biệt, lệnh Timer.
- Hiểu và biết cách sử dụng các tiếp điểm đặc biệt, lệnh Timer trong quá trình soạn thảo.

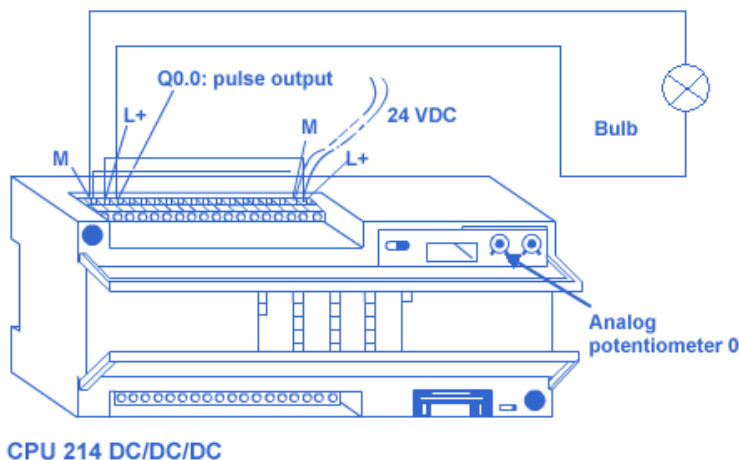
Các kiến thức cần thiết:

- Xem lại các tiếp điểm đặc biệt, lệnh Timer.
- 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, động cơ, nút nhấn, bóng đèn.

4.3.17 Điều khiển kiểm soát độ sáng của bóng đèn tròn 24V/1W



Yêu cầu công nghệ

Viết chương trình điều khiển kiểm soát độ sáng của bóng đèn tròn 24V/1W với thao tác đầu ra tích hợp tốc độ cao. Sự cài đặt thông qua bộ điều chỉnh tương tự (Analog potentiometer: bộ chiết áp 0, hay POT0) tác động tới cả độ rộng xung của tín hiệu xung vuông ở ngõ ra Q0.0 và độ sáng của bóng đèn.

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.

Mục đích – Yêu cầu:

- Làm quen với các lệnh dịch chuyển, lệnh so sánh, lệnh tạo xung vuông.
- Hiểu và biết cách sử dụng các lệnh dịch chuyển, lệnh so sánh, lệnh tạo xung vuông trong quá trình soạn thảo.

Các kiến thức cần thiết:

- Xem lại các các lệnh dịch chuyển, lệnh so sánh, lệnh tạo xung vuông.
- 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, động cơ, nút nhấn, bóng đèn.

Phụ lục

Tài liệu tham khảo

- [1]. Kỹ thuật điều khiển lập trình với Simatic S7 – 200 Phần I
Biên soạn: Châu Chí Đức (Trung tâm Việt - Đức) Trường Đại học Sư phạm Kỹ thuật TP.HCM
- [3]. WWW.OMRON.COM
- [4]. WWW.SIEMENS.COM
- [5]. TS. Nguyễn Thị Phương Hà, “Điều khiển tự động” Nhà xuất bản khoa học kỹ thuật
- [6]. “Automation with Micro PLC SIMATIC S7-200” Siemens, Germany.
- [7]. “Success_e.pdf” Siemens, Germany.
- [8]. “ A beginner’s guide to PLC” OMRON, Japan.
- [9]. Robert N.Bateson, “Introduction To Control System Technology” Maxwell Macmillan International Editions.
- [10] “Statement List for S7-300 and S7-400 Programming” Siemens, Germany.
- [11]. “Automation with Micro PLC SIMATIC S7-200” Siemens, Germany.
- [12]. “S7-300 Programmable Controller Installation and Hardware” Siemens, Germany.
- [13]. “ A beginner’s guide to PLC” OMRON, Japan.
- [14]. Peter Rohner, “Automation With Programmable Logic Controllers” UNSW PRESS
- [15]. “Textbook for Programmable guide” Mitsubishi Electric
- [16]. “Programmable Controller Programming” NAIS, Matsushita Electric
- [17]. Lê Văn Tiến Dũng, “Hướng dẫn thực hành PLC & mạng PLC” Đại Học Kỹ thuật Công nghệ Tp.HCM, năm 2004.