TRƯỜNG CAO ĐẲNG NGHỀ BẮC GIANG
Khoa điện - điện tử

Thuyết minh đồ án tốt nghiệp

Thiết kế mô hình hệ thống điều khiển đèn giao thông tại ngã tư

Phu lục

Lời nói đầu

Hiện nay sự tiến bộ khoa học kĩ thuật trên thế giới diễn ra nhanh chóng, với sự ra đời của hàng loạt những sản phẩm mới ứng dụng những tiến bộ ở những nước phát triển.Đặc biệt trong những năm gần đây kĩ thuật điều khiển phát triển mạnh mẽ, có nhiêug công nghệ điều khiển mới được ra đời để thay thế cho những công nghệ đã lỗi thời.

Để bắt kịp với tiến bộ khoa học kĩ thuật trên thế giới cũng như đáp ứng yêu cầu CNH_HĐH đất nước thì ngành công nghiệp Việt Nam đang thay đổi nhanh chóng, công nghệ và thiết bị hiện đại đang dần dần được thay thế các công nghệ lạc hậu và thiết bị cũ. Các thiết bị công nghệ tiên tiến với hệ thống điều khiển lập trình PLC, Vi xử lý, điện khí nén, điện tử. Đang được úng dụng rộng rãi trong công nghiệp như các dây truyền xản xuất nước ngọt, chế biến thức ăn gia xúc, máy điều khiển theo chương trình CNC, các hệ thống đèn giao thông, các hệ thống báo động. Trong các trường đại học, cao đẳng và các trường trung học đã và đang đưa các thiết bị hiện đại có khả năng lập trình được vào giảng dạy. Một trong những loại thiết bị có ứng dụng mạnh mẽ và đảm bảo có độ tin cậy cao là hệ thống điều khiển tự động PLC.

Với đề tài "Thiết kế mô hình hệ thống điều khiển đèn giao thông tại ngã tư ". Chúng em đã vận dụng được những ưu điểm của hệ thông điều khiển này có hiệu quả cao. Điều đặc biệt là ý tưởng này được ứng dụng trong thực tế rất nhiều. Bởi vì hiện trạng giao thông Việt Nam còn rất thô sơ, lạc hậu, người tham gia giao thông không đi theo đúng nguyên tắc nào mới dẫn đến tắc đường, tai nạn..

Sau quá trình học tập rèn luyện và nghiên cứu tại trường chúng em đã tích luỹ được vốn kiến thức để thực hiện đề tài của mình. Cùng với sụ hướng dẫn tận tình của thầy giáo **nguyen dinh khanh**, cũng như các thầy cô giáo trong khoa và các bạn sinh viện cùng khoá đến nay chúng em đã hoàn thành đề tài này với nội dung sau:

1: Xác định nhiệm vụ điều khiển hệ thống.

2: Giới thiệu chung về PLC.

3: Thiết kế chế tạo mô hình mô phỏng.

4: Viết chương trình chạy cho hệ thống qua phần mềm ứng dụng.

5: Hoàn thành thuyết minh.

Do thời gian nghiên cứu có hạn nên không thể tránh khỏi nhưng sai sót, chúng em rất mong nhận đượoc sự góp ý, chỉ dẫn thêm của các thầy cô cũng như ý kiến đóng góp của các bạn sinh viên để đề tài của chúng em hoàn thiện hơn, đáp ứng đầy đủ những mục tiêu đã đặt ra.

Chúng em xin chân thành cảm ơn!

bac giang, ngày 11 tháng 6 năm 2010

Nhận xét của giáo viên hướng dẫn 1

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

Nhận xét của giáo viên hướng dẫn 2

....
....

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………

DẪN NHẬP

I. Đặt vấn đề

Tự động ngày càng đóng vai trò quan trọng trong đời sống và công nghiệp. Ngày nay ngành tự động đã phát triển tới trình độ cao nhờ những tiến bộ của lý thuyết điều khiển tự động, của những ngành khác như điện tử, tin học, … Nhiều hệ thống điều khiển đã ra đời, nhưng phát triển mạnh và có khả năng phục vụ rộng là bộ điều khiển PLC. Sở dĩ như thế, do bộ PLC có nhiều ưu điểm nổi bậc so những bộ điều khiển khác :

- Đơn giản, dể dàng thay đổi, lập trình .
- Tin cậy trong môi trường công nghiệp.
- Cạnh tranh được giá thành với các bộ diều khiển khác.

Cuối thập niên 60 xuất hiện khái niệm về PLC và đã được phát triển rất nhanh. Năm 1974 PLC đã sử dụng nhiều bộ xử lý như : mạch định thời, bộ đếm, dung lượng nhớ đến 12KB và có 1024 điểm nhập xuất. Năm 1976 đã giới thiệu hệ thống đưa tín hiệu vào ra từ xa. Năm 1977 PLC đã dùng đến vi xử lý. Năm1980 phát triển các khối nhập xuất thông minh nâng cao điều khiển thuận lợi qua viễn thông, nâng cao việc phát triển phần mềm, dùng máy tính cá nhân lập trình. Đến năm 1985 đã thành lập mạng PLC.

Riêng nước ta sắp tới đây hành rào thuế quan khu vực được loại bỏ, kinh tế mở cửa hợp tác với nước ngoài. Trước tình hình đó, nền công nghiệp sẽ gặp không ít khó khăn do còn nhiều dây chuyền có công nghệ lạc hậu. Để có chổ đứng và thế mạnh trên thương trường, nhà nước đã đặc biệt chú trọng đến ứng dụng và phát triển tự động trong sản xuất, nhằm nâng cao năng suất, chất lượng sản phẩm và hạ giá thành. Một trong những phương án tốt nhất và được sử dụng rộng hiện nay là thay thế những hệ thống đó bằng bộ điếu khiển PLC. Để phát triển mạnh hơn nữa, nhiệm vụ đặt ra hàng đầu là đào tạo những chuyên gia về tự động điều khiển nói chung và về PLC nói riêng.

Là một kỹ sư điện công nghiệp, công việc sẽ gắn liền với điều khiển, vận hành hệ thống sản xuất. Như vậy, những hiểu biết về PLC sẽ tạo nhiều thuận lợi để làm việc tốt hơn. Khi đang còn ngồi trên ghế nhà trường, việc tìm hiểu, nghiên cứu để

nắm vững phương pháp lập trình trên bộ PLC rất có ý nghĩa và là điều kiện tốt nhất học hỏi, tích lũy kinh nghiệm.

II. Giới hạn đề tài

-Do thi gian nghiên cu c hạn nên viƯc tìm hiỈu vỊ PLC và SIMATIC S7-300 cđa SIEMENS còn nhiỊu thiu st và không đầy đđ.

-Do hoàn cảnh hc tp không đượ tip xĩc nhiỊu với PLC nên trông qua trình khảo sát và thc hành với PLC còn c nhiỊu kh khăn.

-Do yêu cầu cđa đỊ tài xut phát t thc t nên trong khi xư lý các trưng hợp trong thc t còn c nhiỊu trưng hợp không xư lý đượ.

Chương i: khái quát chung về hệ thống điều khiển

Khái niệm và phân loại về hệ thống điều khiển.

I. Khái niệm về điều khiển.

Điều khiển là một quá trình của một hệ thống trong đó dưới tác động của hay nhiều đại lượng gọi là các đại lượng vào, những đại lượng khác gọi là đại lượng ra được thay đổi theo một quy luật nhất định của hệ thống đó.

II. Phân loại.

Hiện nay người ta chia công nghệ điều khiển ra làm hai loại chính là:

* Phương pháp điều khiển nối cứng (điều khiển lập tuyến).

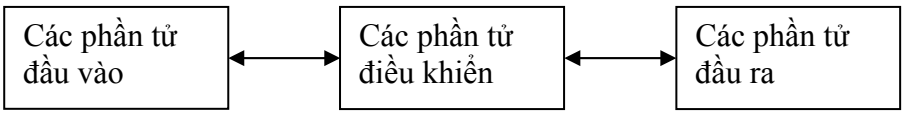
* Phương pháp điều khiển lập trình được.

II.1. Phương pháp điều khiển nối cứng (điều khiển lập tuyến).

Khái niệm: Phương pháp điều khiển nối cứng là hệ thống được thực hiện bởi các phần tử tự động nối với nhau bằng các đường dây.

Trong điều khiển nối cứng người ta chia làm hai loại: điều khiển nối cứng tiếp điểm và điều khiển nối cứng không tiếp điểm.

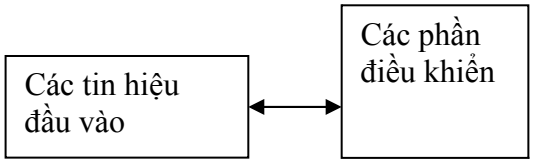
a. Phương pháp điều khiển nối cứng có tiếp điểm: Dùng các khí cụ điên tử như rơle, công tắc tơ với các bộ cảm biến, các đèn , các công tắc, các khí cụ này được nối lại với nhau theo một mạch điện cụ thể để thực hiện một yêu cầu công nghệ nhất định như mạch đổi chiều quay, mạch khởi động giới hạn dòng hay mạch điều khiển động cơ chạy tuần tự và dừng tuần tự.



Sơ đồ cấu trúc hệ thống điều khiển nối cứng có tiếp điểm.

b. Phương pháp điều khiển nối cứng không tiếp điểm: Dùng các cổng logic cơ bản đa năng hay các mạch tuần tự (Gọi chung là IC số) kết hợp với các bộ cảm biến, các đèn, công tắc - Các IC số này cũng được nối lại với nhau theo một sơ đồ logic cụ thể để thực hiện một yêu cầu công nghệ nhất định. Các mạch điều khiển

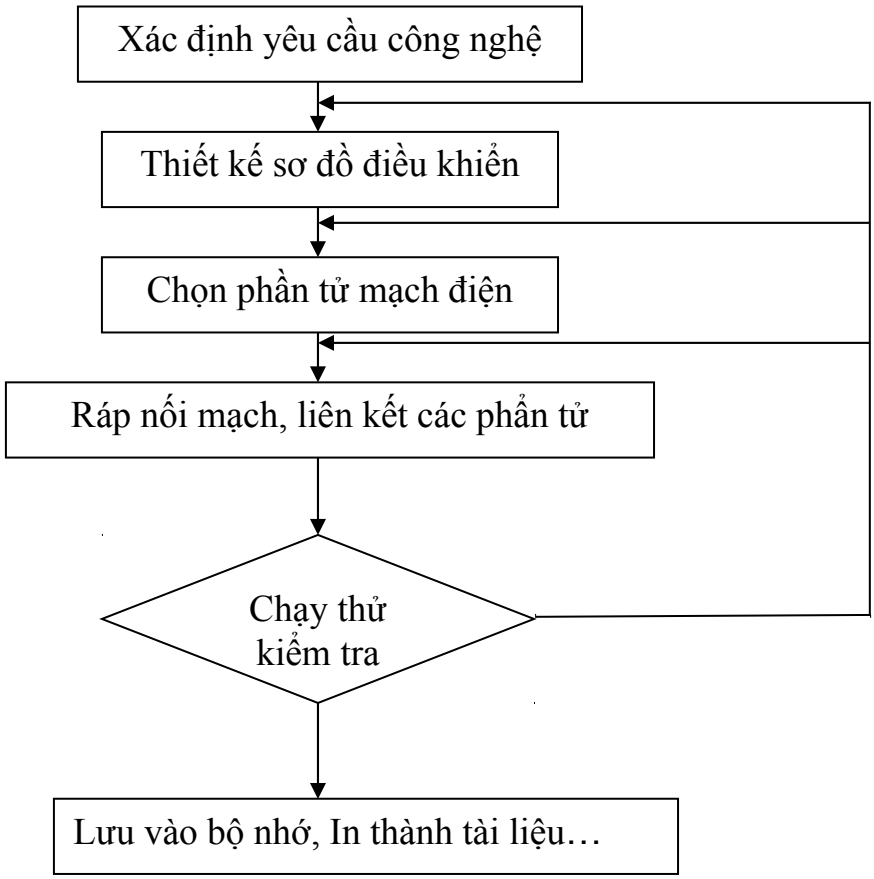
nối cứng sử sụng các linh kiện điện tử công suất, quang trở, triac, tranzitor để thay thế công tắc trong các mạch động lực.



Cấu trúc hệ thống điều khiển nối cứng không tiếp điểm.

Trong hệ thống điều khiển nối cứng, các linh kiện hay khí cụ điện được nối vĩnh viễn với nhau. Do đó khi muốn thay đổi lại nhiệm vụ điều khiển thì phải nối dây lại toàn bộ mạch điện. Với các mạch phức tạp thì không hiệu quả và rất tốn kém.

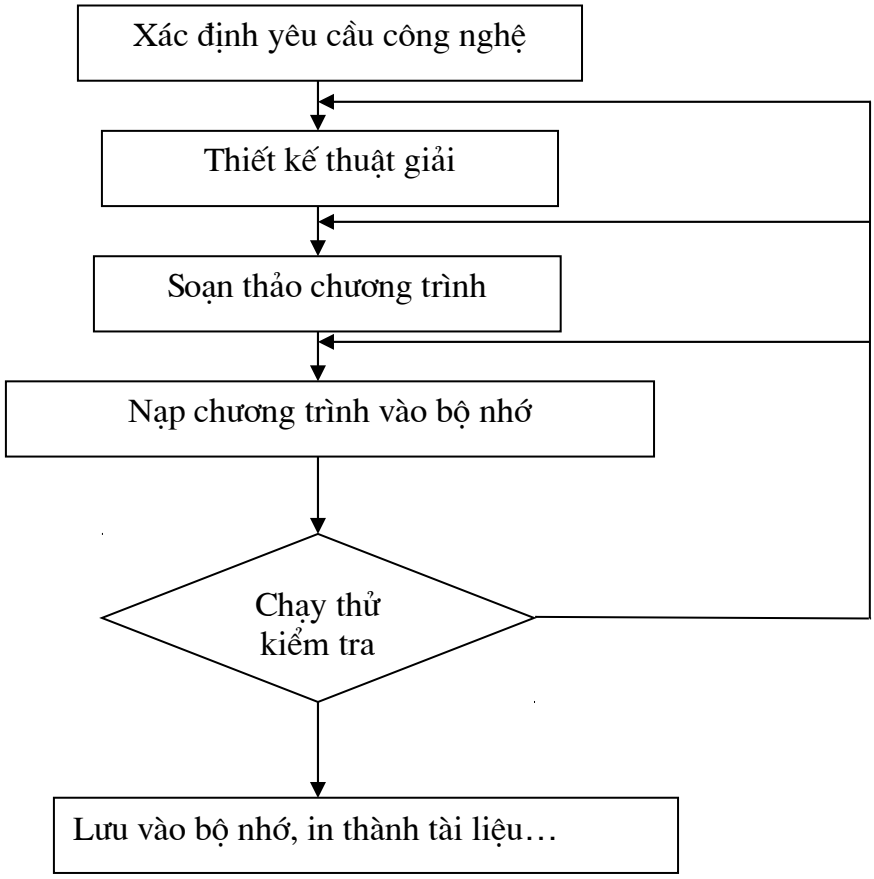
Phương pháp diều khiển nối cứng được thực hiện theo các bước sau.



II.2.Hệ thống điều khiển lập trình được (PLC)

Trong hệ thống điều khiển lập trình được cấu trúc của bộ điều khiển và cách nối dây độc lập với chương trình. Chương trình được định nghĩa hoạt động điều khiển được ghi trực tiếp vào bộ nhớ của bộ điều khiển nhờ sự trợ giúp của bộ lập trình hay máy vi tính. Để thay đổi chương trình điều khiển chỉ cần thay đổi nội dung bộ nhớ của bộ điều khiển, phần nối dây bên ngoài không bị ảnh hưởng. đây là ưu điểm của phương pháp điều khiển lập trình được.

Các bước thiết lập sơ đồ diều khiển lập trình:



III. Sự khác nhau giữa hệ thống điều khiển nối cứng và hệ thống điều khiển lập trình được có thể minh hoạ bằng ví dụ sau:

- Điều khiển hệ thống 3 máy bơm nước qua 3 khởi động từ K1, K2, K3. Trình tự điều khiển như sau: Các máy bơm hoạt động tuần tự nghĩa là K1 đóng trước, tiếp đến là K2 rồi cuối cùng là K3 đống.

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

Trong đó các nút ấn S1, S2, S3, S4 là các phần tử nhập tín hiệu.

Các tiếp điểmK1, K2, K3 và các mối liên kết là các phần xử lý.

Các khởi động từ K1, K2, K3 là kết quả xử lý.

Sơ đồ điều khiển theo kiểu nối cứng:

-Nếu ta thay bằng thiết bị điều khiển lập trình được có thể mô tả như sau:

Tín hiệu vào S1, S2, S3, S4 vẫn giữ nguyên

Tín hiệu ra K1, K2, K3 là các khởi động từ vẫn giữ nguyên

-Phần xử lý được thay thế bằng thiết bị điều khiển lập trình được.

IV.Bộ điều khiển lập trình được.

Bộ điều khiển lập trình được(Programble Logic Controler): gọi tắt là PLC bao gồm các module sau:

-Khối xử lý trung tâm CPU và bộ nhớ chương trình

-Module xuất nhập(Input / Output)

-Hệ thống Bus truyền tín hiệu

-Khối nguồn nuôi

-Module nhập (input module) được nối với các công tắc, nút ấn, các bộ cảm biến để điều khiển chương trình từ bên ngoài. Các ngõ vào được kí hiệu theo thứ tự I1, I2, I3……

-Module xuất (output module) được nối với tải ở ngõ ra như cuộn dây rơle, công tắc tơ, đèn tín hiệu, van điện từ…..

-Chương trình điều khiển được nạp vào bộ nhí nhờ sự trợ giúp của bộ lập trình hay bằng một máy vi tính.

Chương II giới thiệu về PLC - s7 300

I. Đại cương về thiết bị điều khiển logic lập trình PLC.

1. Khái niệm.

Thiết bị điều khiển logic lập trình (Programmable Logic Control , viết tắt là PLC) là loại thiết bị cho phép thực hiện linh hoạt các thuật toán điều khiển số thông qua một ngôn ngữ lập trình. Thay cho việc thực hiện thuật toán đó bằng mạch số như vậy với chương trình điều khiển PLC trở thành một bộ điều khiển số nhỏ gọn dễ dàng thay đổi thuật toán và đặc biệt dễ trao đổi thông tin với môi trường xung quanh (với các PLC khác hay máy tính). Toàn bộ chương trình điều khiển được lưu trong bộ nhớ của PLC dưới dạng các khối chương trình như khối OB, FC hoặc FB, và được thiết lập theo chu kỳ vòng quét.

Để có thể thực hiện được một chương trình điều khiển, tất nhiên PLC có tình năng như một máy tính. Nghĩa là phải có một bộ vi xử lý (PLC), một hệ điều hành, bộ nhớ để lưu chương trình điều khiển, dữ liệu và tất nhiên phải có cổng đầu vào/ra để giao tiếp được với đối tượng điều khiển và trao đổi thông tin với môi trường xung quanh. Bên cạnh đó PLC còn có thêm các khối chức năng đặc biệt khác như bộ đếm (Counter), bộ thời gian (Timer)… và các khối chuyên dụng khác.

2. Cấu trúc của PLC.

Thiết bị điều khiển logic lập trình PLC là thiết bị điều khiển đặc biệt dựa trên bộ vi xử lý, sử dụng bộ nhớ lập trình được để lưu trữ các lệnh và thực hiện các chức năng: phép logic, lập chuỗi, định giờ, đếm, thuật toán để điều khiển máy và các quá trình.

PLc được thiết kế cho phép những người không yêu cầu kiến thức cao về máy tính và ngôn ngữ máy tính có thể vận hành được. Khi cần điều khiển một bài toán ta chỉ cần viết chương trình theo ngôn ngữ PLC và nhập vào bộ nhớ PLC. Thiết bị điều khiển sẽ giám sát các tín hiệu vào /ra theo các chương trình này và thực hiện các quy tắc điều khiển đã lập trình.

PLC có 5 thành phần cơ bản: Đơn vị xử lý trung tâm, bộ nhớ, bộ nguồn nuôi, khối tín hiệu vào/ra và thiết bị lập trình.

Sơ đồ cấu trúc cơ bản của PLC

3. Cấu tạo PLC.

Một PLC điển hình có cấu tạo như hình vẽ:

Ta thấy cấu trúc cơ bản của PLC bao gồm một bộ vi xử lý trung tâm CPU, bộ nhơ (ROM, RAM), khối vào ra, khối phát xung nhịp, pin và hệ thống các BUS.

Toàn bộ hoạt động của PLC được điều khiển bởi CPU, nó được cung cấp bởi khối phát xung nhịp, do đó tốc độ của CPU sẽ phụ thuộc vào khối phát xung nhịp(thông thường khối phát xung nhịp có tần số vào khoảng $1 \div 8$ MHz), xung nhịp này sẽ cung cấp cho tất cả các khối trong PLC để đồng bộ hóa quá trình hoạt động của khối này với CPU.

Hệ thống BUS bao gồm BUS địa chỉ (xác định địa chỉ trên các vùng nhớ), BUS điều khiển (truyền tải các thông tin điều khiển), BUS dữ liệu (truyền tải dữ liệu)và các BUS vào/ra (mang thông tin từ các đầu vào ra).

Có bốn bộ nhớ trong PLC:

+Bộ nhớ ROM: là loại bộ nhớ không thể thay đổi được, bộ nhớ này chỉ nạp được một lần nên ít được sử dụng phổ biến như các loại bộ nhớ khác.

+Bộ nhớ RAM: là loại bộ nhớ có thể thay đổi được và dùng để chứa các chương trình ứng dụng cũng như dữ liệu, dữ liệu chứa trong RAM sẽ bị mất khí mất điện. Tuy nhiên, điều này có thể khắc phục bằng cách dung Pin.

+Bộ nhớ EPROM: giống như RAM, nhuồn nuôi cho EPROM không cần dung Pin, tuy nhiên nội dung chứa trong nó có thể xóa bằng cách chiếu tia cực tím vào một cửa sổ nhỏ trên EPROM và sau đó nạp lại nội dung bằng máy nạp.

+Bộ nhớ EEPROM: kết hợp hai ưu điểm của RAM và EPROM, loại này có thể xoa và nạp bằng tín hiệu điện. Tuy nhiên số lần nạp cũng có giới hạn.

4. Ưu nhược điểm của hệ thống:

Trong giai đoạn đầu của thời kỳ phát triển công nghiệp vào khoảng 1960 - 1980, yêu cầu tự động của hệ điều khiển được thực hiện bằng các Rơle điện tử nối với nhau bằng dây dẫn điện trong bảng điều khiển, trong nhiều trường hợp bảng điều khiển có kích thước quá lớn đến nỗi không thể gắn toàn bộ lên trên tường và các dây nối cũng không hoàn toàn tốt vì thế rất hay xảy ra trục trặc trong hệ thống. Một điểm quan trọng nữa là do thời gian làm việc của các Rơle có giới hạn nên khi cần thay thế phải ngừng toàn bộ hệ thống và dây nối cũng phảit hay mới cho phù hợp, bảng điều khiển chỉ dùng được một yêu cầu riêng biệt không thể thay đổi tức thời chức năng khác mà đòi hỏi thợ chuyên môn có tay nghề cao. Tóm lại hệ thống điều khiển Rơle hoàn toàn không linh hoạt.

**Tóm tắc nhược điểm của hệ thống điều khiển dùng Rơle:*

-Tổn kém rất nhiều dây dẫn.

-Thay thế rất phức tạp.

-Cần công nhân sửa chữa tay nghề cao.

-Công suất tiêu thụ lớn.

-Thời gian sửa chữa lâu.

-Khó cập nhật sơ đồ nên gây khó khăn cho công tác bảo trì cũng như thay thế.

**Ưu điểm của hệ điều khiển PLC:*

Sự ra đời của hệ điều khiển PLC đã làm thay đổi hẳn hệ thống điều khiển cũng như các quan niệm thiết kế về chúng, hệ điều khiển dùng PLC có nhiều ưu điểm sau:

-Giảm 80% số lượng dây dẫn.

-Công suất tiêu thụ của PLC rất thấp.

-Có chức năng tự chuẩn đoán do đó dễ dàng cho công tác sửa chữa được nhanh chóng và dễ dàng.

-Chức năng điều khiển thây đổi dễ dàng bằng thiết bị lập trình (máy tính, màn hình)

mà không cần thay đổi phần cứng nếu không có yêu cầu thêm bớt các thiết bị xuất nhập.

-Số lượng Rơle và Timer ít hơn nhiều so với hệ điều khiển cổ điển.

-Số lượng tiếp điểm trong chương trình sử dụng không hạn chế.

-Thời gian hoàn thành một chu chình điều khiển rất nhanh(vài mS) dẫn đến tăng cao tốc độ sản xuất.

-Chi phí lắp đặt thấp.

-Độ tin cậy cao.

-Chương trình điều khiển có thể in ra giấy chỉ trong vài phút thuận tiện cho vấn đề bảo trì và sửa chữa hệ thống.

5. Phân loại PLC.

Hiện nay trong lĩnh vực điều khiển nói chung và ngành tự động hóa nói riêng, các PLC mới được đưa vào sử dụng ngày càng nhiều với tính năng rất lớn như:

+ PLC S_5

+ PLC S7 - 200

+ PLC S7 - 300

+ PLC S7 - 400

+ PLC LOGO

II. Hệ thống điều khiển PLC S7 - 300.

II.1. Cấu trúc phần cứng của hệ thống PLC S7 - 300.

Thông thường, để tăng tính mềm dẻo trong ứng dụng thực tế mà ở đó phần lớn các đối tượng điều khiển có số tín hiệu đầu vào/ra cũng như chủng loại tín hiệu vào /ra khác nhau mà các bộ điều khiển PLC được thiết kế không bị cứng hóa về cấu hình. Chúng được chia nhỏ thành các modul. Số các modul được sử dụng nhiều hay ít tuỳ theo yêu cầu công nghệ, song tối thiểu bao giờ cũng phải có một modul chính là các modul CPU, các modul chức năng chuyên dụng như PID, điều khiển động cơ. Chúng được gọi chung là modul mở rộng. Tất cả các modul được gá trên những thanh ray (RACK).

*** Modul CPU.**

Là modul có chứa bộ vi xử lý, hệ điều hành, bộ nhớ, các bộ thời gian, bộ đếm, cổng truyền thông(chuẩn truyền ***RS485***) và có thể còn có một vài cổng vào /ra số (Digital). Các cổng vào ra có trên modul CPU được gọi là cổng vào ra ONBOART.

Modul CPU bao gồm các loại sau :

***CPU 312-IFM**

-6ES7-312-5AC00-OABO

-6ES7-312-5AC01-OABO

-6ES7-312-5AC02-OABO

-6ES7-312-5AC81-OABO

-6ES7-312-5AC82-OABO

+Các module này có:

-Vùng nhớ làm việc :6KB

-Thời gian xử lí 1 khối lệnh:0.6ms/KAW

-DI/DO trên module CPU:10/6

-Sử dụng trong nối mạng MPI

***CPU 313**

-6ES7 313-1AD00-0AB0

-6ES7 313-1AD01-0AB0

-6ES7 313-1AD02-0AB0

-6ES7 313-1AD03-0AB0

+Các module này có:

-Vùng nhớ làm việc :12KB

-Thời gian xử lí 1 khối lệnh:0.6ms/KAW

-Sử dụng trong nối mạng MPI

***CPU 314**

-6ES7 314-1AE01-0AB0

-6ES7 314-1AE02-0AB0

-6ES7 314-1AE03-0AB0

-6ES7 314-1AE04-0AB0

-6ES7 314-1AE83-0AB0

-6ES7 314-1AE84-0AB0

+Các module này có:

-Vùng nhớ làm việc :24KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-Sử dụng trong nối mạng MPI

***CPU 314 IFM**

-6ES7 314-5AE00-0AB0

-6ES7 314-5AE01-0AB0

-6ES7 314-5AE02-0AB0

-6ES7 314-5AE03-0AB0

-6ES7 314-5AE82-0AB0

-6ES7 314-5AE83-0AB0

+Các module này có:

-Vùng nhớ làm việc :Từ 24KB đến 32KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-DI/DO trên module CPU:20/16

-Truyền thông kiểu MPI

***CPU 315**

-6ES7 315-1AF00-0AB0

-6ES7 315-1AF01-0AB0

-6ES7 315-1AF02-0AB0

-6ES7 315-1AF03-0AB0

+Các module này có:

-Vùng nhớ làm việc :48KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-Sử dụng trong nối mạng MPI

***CPU 315-2DP**

-6ES7 315-2AF00-0AB0

-6ES7 315-2AF01-0AB0

-6ES7 315-2AF02-0AB0

-6ES7 315-2AF03-0AB0

-6ES7 315-2AF82-0AB0

-6ES7 315-2AF83-0AB0

+Các module này có:

-Vùng nhớ làm việc :48KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-Truyền thông kiểu MPI,Profilbus-DP

***CPU 316**

-6ES7 316-1ag00-0ab0

+Các module này có:

-Vùng nhớ làm việc :128KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-Sử dụng trong nối mạng MPI

***CPU 316-DP**

-6ES7 316-2AG00-0AB0

+Các module này có:

-Vùng nhớ làm việc :128KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-Truyền thông kiểu MPI,Profilbus-DP

***CPU 318-2**

-6ES7 318-2AJ00-0ab0

+Các module này có:

-Vùng nhớ làm việc :256KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-Sử dụng trong nối mạng MPI

***CPU 614**

-6ES7 614-1aH00-0ab3

-6ES7 614-1aH01-0ab3

-6ES7 614-1aH02-0ab3

-6ES7 614-1aH03-0ab3

+Các module này có:

-Vùng nhớ làm việc :128KB đến 192KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-Sử dụng trong nối mạng MPI

***CPU 614**

-6ES7 614-1AH00-0AB3

-6ES7 614-1AH01-0AB3

-6ES7 614-1AH02-0AB3

-6ES7 614-1AH03-0AB3

+Các module này có:

-Vùng nhớ làm việc :Từ 128KB đến 192KB

-Thời gian xử lí 1 khối lệnh:0.3ms/KAW

-DI/DO trên module CPU:512KB

-Truyền thông kiểu MPI

***CPU M7**

+CPU 388-4

-6ES7-388-4BN00-0AC0

*** Các modul mở rộng.**

Các modul mở rộng được chia làm 5 loại chính……..

1. PS (Power supply) module nguồn nuôi: có 3 loại 2A, 5A, 10A.

2. SM (Sigal module): Module mở rộng cổng tín hiệu vào/ra gồm:

- DI (Digital Input): module mở rộng cổng vào số có thể là 8, 16 hoặc 32 tuỳ thuộc vào từng loại module.

- DO (Digital Output): module mỏ rộng cổng ra số.

- DI/DO: module mỏ rộng cổng vào/ra số.

- AI (Analog Input):cổng vào tương tự, chúng là những bộ chuyển đổi tương tự số 12 bits.

- AO (Analog Output) Module cổng ra tương tự, là những bộ chuyển đổi tương tự(DA).

- AI/AO: Module mở rộng các cổng vào/ra tương tự.

3. IM (Interface Module) Module ghép nối:

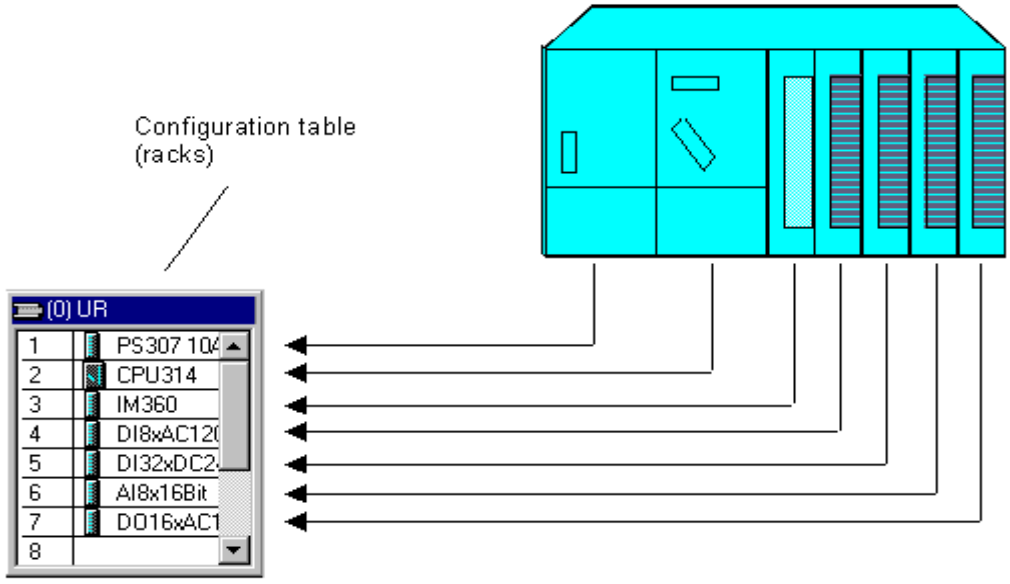
Là loại module chuyên dụng có nhiêm vụ nối từng nhóm các module mở rộng lại vơi nhau thành một khối và được quản ly chung bởi 1 module CPU.

Thông thường các module mở rộng được gá liền nhau trên một thanh đỡ gọi là Rack. Mỗi 1 Rack có thể gá được nhiều nhất 8 module mở rộng (không kể module

CPU, module nguồn nuôi). Một module CPU S7-300 có thể làm việc nhiều nhất với 4 Rack và các Rack này phải được nối với nhau bằng module IM.

4. FM (Function Module) :Module có chức năng điều khiển riêng: VD module động cơ bước, module PID….

5. CP (Commuication Module):Module phục vụ truyền thông trong mạng giữa các PLC với nhau hoặc giữa PLC với máy tính.



Sơ đồ kết nối trạm PLC S7 - 300

II.2 Xử lý các tín hiệu vào ra, cấu trúc bộ nhớ trong PLC.

Các tín hiệu vào ra từ đầu vào ra của PLC sẽ được lưu trữ trong các vùng nhớ. Để xử lý các tín hiệu này ta truy nhập vào vùng địa chỉ để lấy các giá trị của chúng. Sau đây sẽ trình bày cấu trúc bộ nhớ và các truy nhập cho PLC Siemens.

*** Phương pháp truy nhập.**

PLC lưu trữ thông tin trong bộ nhớ. Bộ nhớ của PLC được chia làm nhiều vùng (I, Q, M, T, C,….) mỗi vùng nhớ đều có địa chỉ xác định. Ta có thể truy nhập (ghi

hoặc đọc thông tin) vào các ô nhớ trong các vùng bằng địa chỉ của chúng. Có 2 cách truy nhập theo ting bit hoặc truy nhập theo byte.

+*Truy nhập theo từng bit:* Để truy nhập theo từng bit ta phải đánh địa chỉ bao gồm: Địa chỉ vùng nhớ, địa chỉ byte, địa chỉ bit (ngăn cách giữa địa chỉ byte và địa chỉ bit là dấu “.”

Như vậy thông tin của đầu vào I3.4 sẽ được lưu trữ trong ô nhớ có địa chỉ I3.4 . Truy nhập vào ô nhớ này sẽ biết được thông tin đầu vào I3.4.

+*Truy nhập theo byte:* Ta có thể truy nhập các vùng nhớ theo byte, Word (2 byte), Double Word (4 byte). để truy nhập theo các phương pháp này ta phải đánh địa chỉ bao gồm: Địa chỉ vùng nhớ (V, I, Q, M, SM, T, C, HC…)

II.3 Nguồn nuôi và ngõ ra của PLC S7-300.

- Nguồn nuôi: là đơn vị dùng để chuyển đổi nguồn AC thành nguồn DC (5V, 24V) để cung cấp cho CPU và các khối vào ra.

- Ngõ ra: Plac S7-300 có ngõ ra là các phần tử hoạt động tương thích với các loại tín hiệu vào như Role, các van điều khiển….

II.4 Các hệ đếm và các kiểu dữ liệu.

4.1. Các hệ đếm:

Chúng ta sử dụng rất nhiều hệ đếm, quen dùng nhất vẫn là hệ thập phân (hệ đếm cơ số 10). Tuy nhiên ngoài hệ thập phân còn có rất nhiều các hệ đếm khác:

- Hệ nhị phân: là hệ đếm cơ số 2, sử dụng 2 con số 0 và 1 để biểu diễn giá trị.

- Hệ bát phân: là hệ đếm cơ số 8, sử dụng 8 con số 0,1,2,3,4,5,6,7 để biểu diễn các giá trị.

- Hệ thập phân: là hệ đếm cơ số 10 dùng các con số 0,1,2,3,4,5,6,7,8,9 để biểu diễn các giá trị.

- Hệ thập lục phân: là hệ đếm cơ số 16 sử dụng 16 con số 0……..F để biểu diễn các giá trị.

4.2. Các kiểu dữ liệu.

PLC lưu trữ các dữ liệu trong các bộ nhớ, các dữ liệu này có thể được lưu trữ ở nhiều dạng khác nhau:

- BOOL: với dung lượng là 1bit và có giá trị là 0 hoặc 1, đây là kiểu dữ liệu biến có 2 giá trị.

- BYTE: gồm 8 bits, thường được dùng để biểu diễn 1 số nguyên dương trong khoảng 0….255 hoặc mã ASCII của 1 ký tự.

- WORD: gồm 2 bytes để biểu diễn số nguyên dương từ 0……65535.

- DWORD: là từ kép có giá trị là 0….2^{32}-1.

- INT: cũng có dung lượng 2 bytes, dùng để biểu diễn 1 số nguyên trong khoảng -32768…..+32767 (2^{-15}….2^{15}-1).

- REAL: có dung lượng là 4 bytes dùng để biểu diễn 1 số thực trong khoảng -3,4E^{38}…..3,4E^{38}.

III. Cấu trúc bộ nhớ của CPU của PLC S7 - 300.

Được chia ra làm 3 vùng chính:

1) Vùng chứa chương trình ứng dụng: vùng nhớ chương trình được chia làm 3 miền:

+ OB: Miền chứa chương trình tổ chức.

+ FC: (*Funktion*) 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.

+ FB: (*Funktion Block*) Miền chứa chương trình con,được tổ chức thành hàm và có khả năng trao đổi dữ liệu với bất cứ một khối chương trình nào khác. Các dữ liệ phải được xây dụng thành một khối dữ liệu riêng (gọi là DB - *Data block*).

2) Vùng chứa các tham số của hệ điều hành và chương trình ứng dụng, được chia thành 7 miền khác nhau, bao gồm:

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

Q (*Procees image output*): Miền bộ đếm các cổng ra số. Kết thúc giai đoạn thực hiện chương trínhẽ chuyển giá trị của bộ đếm tới cổng ra số. Thông thường không trực tiếp gán giá trị tới tận cổng ra mà chỉ chuyển chúng vào bộ nhớ Q.

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

T: Miền nhớ phụ vụ bộ thời gian(TIME) bao gồm việc lưu giữ giá trị thời gian đặt trước (PV - Preset Value), giá trị đếm thời gian tức thời (CV - Curren Value) cũng như các giá trị logic đầu ra của bộ thời gian.

C: Miền nhớ phục vụ bộ đếm (Counter) bao gồm việc lưu giữ giá trị đặt trước (PV), và giá trị đếm tức thời (CV) và giá trị logic đầu ra của bộ đếm.

PI: Miền địa chỉ cổng vào của các modul tương tự. Các giá trị tương tự tại cổng vào của modul tương tự sẽ được đọc và chuyển tự động theo những địa chỉ. Chương trình ứng dụng có thể truy nhập miền nhớ PI theo tong byte (PIB), từng từ (PIW) hoặc theo từ kép (PID).

PQ: Miền địa chỉ cổng ra cho các modul tương tự. Các giá trị theo những địa chỉ này được modul tương tự chuyển tới các cổng ra tương tự. Chương trình ứng dụng có thể truy cập miền PQ theo từng byte (PQB), từng từ (PQW) hoặc theo từ kép (PQD).

3) Vùng chứa các khối dữ liệu: được chia làm hai loại

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

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

IV. Vòng quét của chương trình.

SPS (PLC) thực hiện các công việc (bao gồm cả chương trình điều khiển) theo chu trình lặp. Mỗi vòng lặp được gọi là một vòng quét (scancycle). Mỗi vòng quét đều bắt đầu bằng việc 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 OB1. sau giai đoạn thực hiện chương trình là giai đoạn chuyển nội dung của bộ đệm ảo Q tới các cổng ra số. Vòng quét được kết thúc bằng giai đoạn xử lý các yêu cầu truyền thông (nếu có) và kiển tra trạng thái của CPU. Mỗi vòng quét có thể được mô tả như sau:

Quá trình hoạt động của một vòng quét

Chú ý: 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 một vòng quét được gọi là thời gian vòng quét (Scan time). Thời gian vòng quét không cố định, tức là không phải vòng quét nào cũng thực hiện trong một khoảng thời gian như nhau. Có vòng quét thực hiện

lâu, có vòng quét thực hiện nhanh tuỳ thuộc vào số câu lệnh trong chương trình được thực hiện, vào khối dữ liệu truyền thông trong vòng quét đó.

Như vậy giữa việc đọc dữ liệu từ đối tượng cần xử lý, tính toán và việc giử thông tinđiều khiển đến đối tượng có một khoảng thời gian bằng thời gian một vòng quét. Nói cách khác, thời gian vòng quét quyết định thời gian thực của chương trình điều khiển trong PLC. Thời gian vòng quét càng ngắn, tính thời gian thực của chương trình càng cao.

Nếu sử dụng các khối chương trình đặc biệt có chế độ ngắt, ví dụ khối OB40, OB80….Chương trình của các khối đó sẽ được thực hiện trong vòng quét khi xuất tín hiệu báo ngắt cùng chủng loại. Các khối chương trình này có thể thực hiện tại mọi vòng quét chứ không bị gò ép là phải ở trong giai đoạn chương trình. Chẳng hạn một tín hiệu báo ngắt xuất hiện khi PLC đang ở giai đoạn truyền thông và kiển tra nội bộ, PLC sẽ tạm dừng công việc truyền thông, kiển tra, để thực hiện ngắt như vậy, thời gian vòng quét sẽ càng lớn khi càng có nhiều tín hiệu ngắt xuất hiện trong vòng quét. Do đó để nâng cao tính thời gian thực cho chương trình điều khiển, tuyệt đối không nên viết chương trình xử lý ngắt quá nhiều hoặc sử dụng quá lạm dụng chế độ ngắt trong chương trình điều khiển.

Tại thời điểm thực hiện lệnh vào ra, thông thường lệnh không làm việc trực tiếp với cổng ra vào mà chỉ thông qua bộ nhớ đệm của cổng trong vùng nhớ tham số. Việc truyền thông giữa các bộ đệm ảo với ngoại vi trong giai đoạn 1 và 3 do hệ điều hành CPU quản lý, ở một số modul CPU, khi gặp lệnh vào /ra ngay lập tức hệ thông sẽ cho dừng mọi công việc khác, ngay cả chương trình xử lý ngắt để thực hiện với cổng vào /ra.

V. Những khối OB đặc biệt.

Khối OB1 có chức năng quản lý chính trong toàn bộ chương trình, có nghĩa là nó sẽ thực hiện một cách đều đặn ở từng vòng quét khi thực hiện chương trình. Ngoài ra Step7 còn có nhiều khối OB1 đặc biệt khác và mỗi khối OB đó có một nhiệm vụ khác nhau, ví dụ các khôi OB chứa các chương trình ngắt của chương trình báo lỗi….Tuỳ thuộc vào CPU khác nhau mà có các khối OB khác nhau. ví dụ khối OB đặc biệt.

OB10 (*Time of Day Interrupt*): Chương trình trong khối OB10 sẽ được thực hiện khi giá trị của đồng hồ thời gian thực nằm trong một khoảng thời gian đã quy định. OB10 có thể được gọi một lần, nhiều lần cách đều nhau từng phút, từng giờ, từng ngày….Việc quy định thời gian hay số lần gọi OB10 được thực hiện bằng chương trình hệ thống SFC28 hoặc trong bảng tham số modul CPU nhờ phần mềm Step7.

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

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

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

OB80 (*Cycle Time Fault*): Chương trình sẽ được thực hiệnkhi thời gian vòng quét (Scan time) vượt qua thời gian cực đại đã quy định hoặc khi có tín hiệu ngắt gọi một khối OB nào đó mà khối OB này chưa kết thúc ở lần gọi trước. Mặc định, Scan time cực đại là 150ms nhưng có thể thay đổi tham số nhờ phần mềm Step7.

OB81 (*Power Supply Fault*): Nếu có lỗi về phần nguồn cung cấp thì gọi chương trình trong khôi OB81.

OB82 (*Diagnostic Interrput*) : Chương trình trong khối này sẽ được gọi khi CPU phát hiện có lỗi ở các modul vào/ra mở rộng. Với điều kiện các modul vào /ra này phải có chức năng tự kiểm tra mình.

OB85 (*Not Load Fault*): CPU sẽ gọi khối OB85 khí phát hiện khối chương trình ứng dụng có sử dụng chế độ ngắt nhưng chương trình xử lý ngắt lại không có trong khối OB tương ứng.

OB87 (*Communication Fault*): Chương trình trong khối này sẽ được gọi khi CPU thấy có lỗi trong truyền thông.

OB100 (*Start Up Information*): Khối này sẽ được thực hiện một lần khi CPU chuyển trạng thái từ STOP sang trạng thái RUN.

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

OB122 (*Synchronouns error*): Khối này sẽ được thực hiện khi CPU phát hiện thấy lỗi truy cập modul trong chương trình, ví dụ trong chương trình có lệnh truy nhập modul mở rộng nhưng lại không có modul này.

VI. Thanh ghi trạng thái.

Khi thực hiện lệnh, CPU sẽ ghi lại trạng thái của phép tính trung gian cũng như kết quả vào 1 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 1 dãy các lệnh logic liên tiếp nhau gồm các lệnh ‘ và’, ‘hoặc’ và nghịch đảo, bits FC có giá trị bằng 1. Nói cách khác, FC = 0 khi dãy các lệnh logic liên tiếp vừa được kết thúc. Ví dụ:

```
A   I0.3      // FC = 1
AN  I0.3      // FC = 1
=   Q4.0      // FC = 0
```

- RLO (*Result of Logic Operation*): Kết quả tức thời của phép tính logic vừa được thực hiện. Ví dụ lệnh

```
A      I0.3
```

+ Nếu trước khi thực hiện bits FC = 0 thì có tác động chuyển đổi nội dung của cổng vào I0.3 vào bít trạng thái RLO.

+ Nếu trước khi thực hiện bits FC = 1 thì có tác dụng thực hiện phép tính AND giữa RLO và giá trị logic cổng vào I0.3. Kết quả của phép tính được ghi lại vào bits trạng thái RLO.

- STA (*Status bits*): Bits trạng thái luôn có giá trị logic của tiếp điểm chỉ định trong lệnh. Ví dụ cả hai lệnh

A I0.3

AN I0.3

Đều được gán cho bits STA cùng một giá trị là nội dung của cổng vào số I0.3

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

- OS (*Stored overflow bit*): Ghi lại giá trị bít phép tính tràn ra ngoài mảng ô nhớ.

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

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

* Khi thực hiện lệnh toán học như cộng, trừ, nhân, chia với số nguyên hoặc số thực.

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)

* Khi thực hiện lệnh toán học với số nguyên nhưng kết quả bị tràn ô nhớ.

CC1	CC0	ý nghĩa
0	0	Kết quả quá nhỏ khi thực hiện lệnh cộng (+I, +D)
0	1	Kết quả quá nhỏ khi thực hiện lệnh nhân (*I, *D) hoặc quá lớn khi thực hiện lệnh cộng trừ (+I, +D, -I, -D).
1	0	Kết quả quá lớn khi thực hiện lệnh nhân, chia (*I, *D, /I, /D)

		hoặc quá nhỏ khi thực hiện lệnh cộng trừ (+I, +D, -I, -D).
1	1	Kết quả bị tràn do thực hiện lệnh chia cho 0 (/I, /D).

* Khi thực hiện lệnh toán học với số thực nhưng kết quả bị tràn ô nhớ

CC1	CC0	ý nghĩa
0	0	Kết quả có số mũ *e* quá lớn
0	1	Kết quả có mantissa quá nhỏ
1	0	Kết quả có mantissa quá lớn
1	1	Phép tính sai quy chuẩn

* Khi thực hiện lệnh chuyển dịch:

CC1	CC0	ý nghĩa
0	0	Giá trị của bit bị đẩy ra bằng 0
1	0	Giá trị của bit bị đẩy ra bằng 1

* Khi thực hiện lệnh logic trong ACCU:

CC1	CC0	ý nghĩa
0	0	Kết quả bằng 0 (=0)
1	0	Kết quả khác 0 (#0)

- BR (*Binary result bit*): Bit tạng thái cho phép liên kết hai loại ngôn ngư lập trình STL và LAD. Chẳng hạn cho phép người sử dụng có thể viết một khối chương trình FB hoặc FC trên STL nhưng gọi và sử dụng chúng trong một chương trình khác trên LAD. Để tạo ra được mối liên kết đó, ta cần phải kết thúc chương trình trong FB, FC bằng bảng ghi

+ 1 vào BR, nếu chương trình chạy không có lỗi.

+ 0 vào BR, nếu chương trình chạy có lỗi.

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

+1, nếu SFC hay SFB thực hiện không có lỗi.

+0, nếu có lỗi khi thực hiện SFC hay SFB.

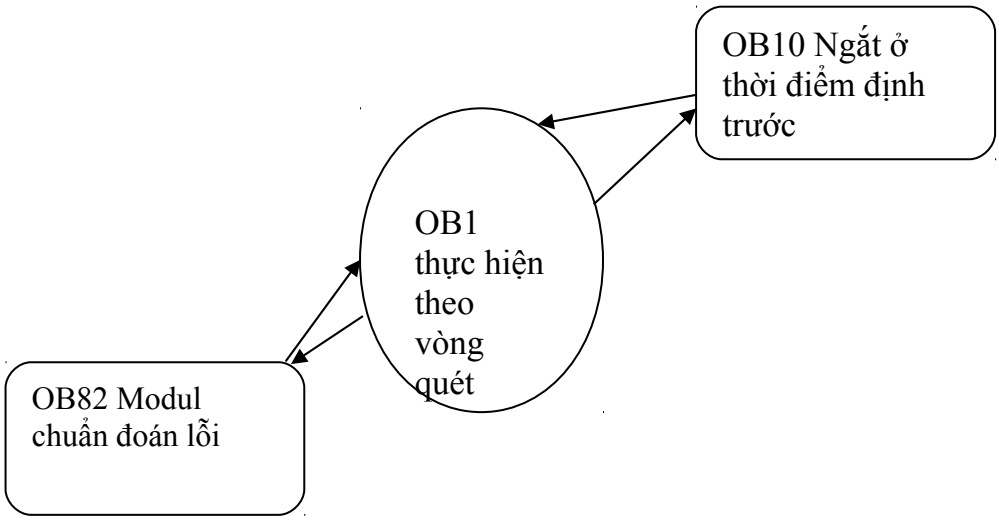
Chương III kỹ thuật lập trình PLC s7 - 300

I. Giới thiệu chung.

I.1. Lập trình tuyến tính và lập trình có cấu trúc.

Bộ nhớ của CPU dành cho chương trình ứng dụng có tên gọi là logic block. Như vậy logic block là tên chung để gọi tất cả các khối chương trình bao gồm: khối chương trình tổ chức OB, khối chương trình FC, khối hàm FB.trong các khối chương trình đó chỉ có duy ngất khối OB1 được thực hiện trực tiếp theo vòng quét. Nó được hệ điều hành gọi theo chu kỳ lặp với khoảng thời gian không cách đều nhau mà phụ thuộc vào độ dài của chương trình. Các loại khối chương trình khác không tham gia vào vòng quét.

Với hình thức tổ chức như vậy thì phần chương trình trong khối OB1 có đầy đủ điều kiện của một chương trình, điều kiện thời gian thực và toàn bộ chương trình ứng dụng có thể chỉ cần viết trong OB1 là đủ. Cách viết tổ chức chương trình với chỉ một khối OB1 duy nhất như vậy gọi là lập trình tuyến tính (Linear Programming)



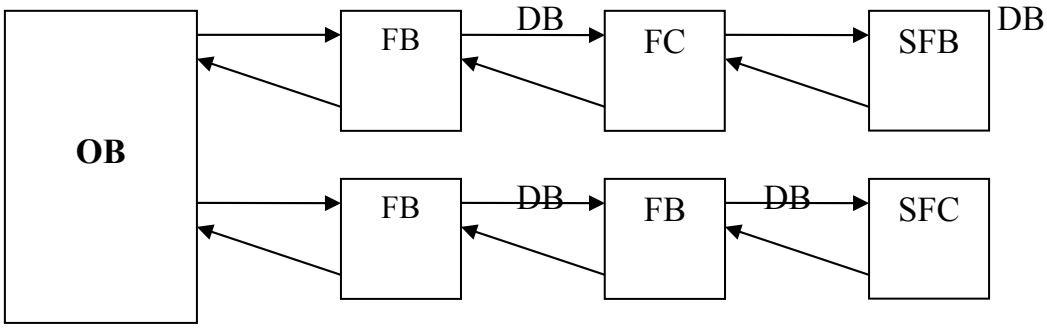
Sơ đồ khối kiểu lập trình tuyến tính

Khối OB1 được hệ thống gọi xoay liên tục theo vòng quét.

Các khối OB khác không tham gia vào vòng quét được gọi bằng các tín hiệu ngắt. S7 - 300 có nhiều tín hiệu báo ngắt như tín hiệu báo ngắt khi có sự cố nguồn nuôi, có sự cố chập mạch ở các modul mở rộng, tín hiệu báo ngắt theo chu kỳ thời gian, và mỗi tín hiệu ngắt như vậy cũng chỉ có khả năng gọi một khối OB nhất định. Ví dụ sự cố báo ngắt nguồn nuôi chỉ gọi khối OB81, tín hiệu báo ngắt truyền thông chỉ gọi khối OB87.

Mỗi tín hiệu báo ngắt hệ thống sẽ dừng công việc đang thực hiện lại, chẳng hạn tạm dừng công việc trên khối OB1 và thực hiện chuyển sang thực hiện chương trình xử lý ngắt trong các khối OB tương ứng. Ví dụ khi đang thực hiện chương trình trên khối OB1 mà xuất hiện báo sự cố truyền thông, hệ thống sẽ tạm dừng thực hiện trên OB1 lại để gọi chương trình truyền thông khối OB87. Chỉ khi nào thực hiện xong chương trình trên khối OB87 thì hệ thống quay trở lại thực hiện tiếp chương trình OB1.

Lập trình có cấu trúc: Chương trình được chia thành nhiều phần nhỏ với từng nhiệm vụ riêng và các phần này nằm trong những khối chương trình khác nhau. Loại hình cấu trúc này phù hợp với nhiều bài toán điều khiển nhiều nhiệm vụ và phức tạp, lại rất thuận lợi cho việc sửa chữa sau này.



Sơ đồ kiểu lập trình có cấu trúc.

I.2 Quy trình thiết kế hệ điều khiển PLC và các phần tử lôgic cơ bản.

1. Quy trình thiết kế hệ thống điều khiển dùng PLC bao gồm các bước sau:

a. Xác định quy trình điều khiển.

Điều đầu tiên cần biết là đối tượng điều khiển của hệ thống, mục đích chính của PLC là phải điều khiển được các thiết bị ngoại vi. Các chuyển động của đối tượng điều khiển được kiểm tra thường xuyên bởi các thiết bị vào, các thiết bị này gửi tín hiệu vào PLC và tiếp đó PLC sẽ đưa tín hiệu điều khiển đến các thiết bị để điều khiển chuyển động của đối tượng.

b. Xác định tín hiệu vào ra.

Bước thứ 2 là phải xác định vị trí kết nối giữa các thiết bị vào ra với PLC. Thiết bị vào có thể là tiếp điểm, cảm biến….Thiết bị ra có thể là rơle điện từ, môtơ, đèn… Mỗi vị trí kết nối được đánh số tương tự ứng với PLC sử dụng

c. Soạn thảo chương trình.

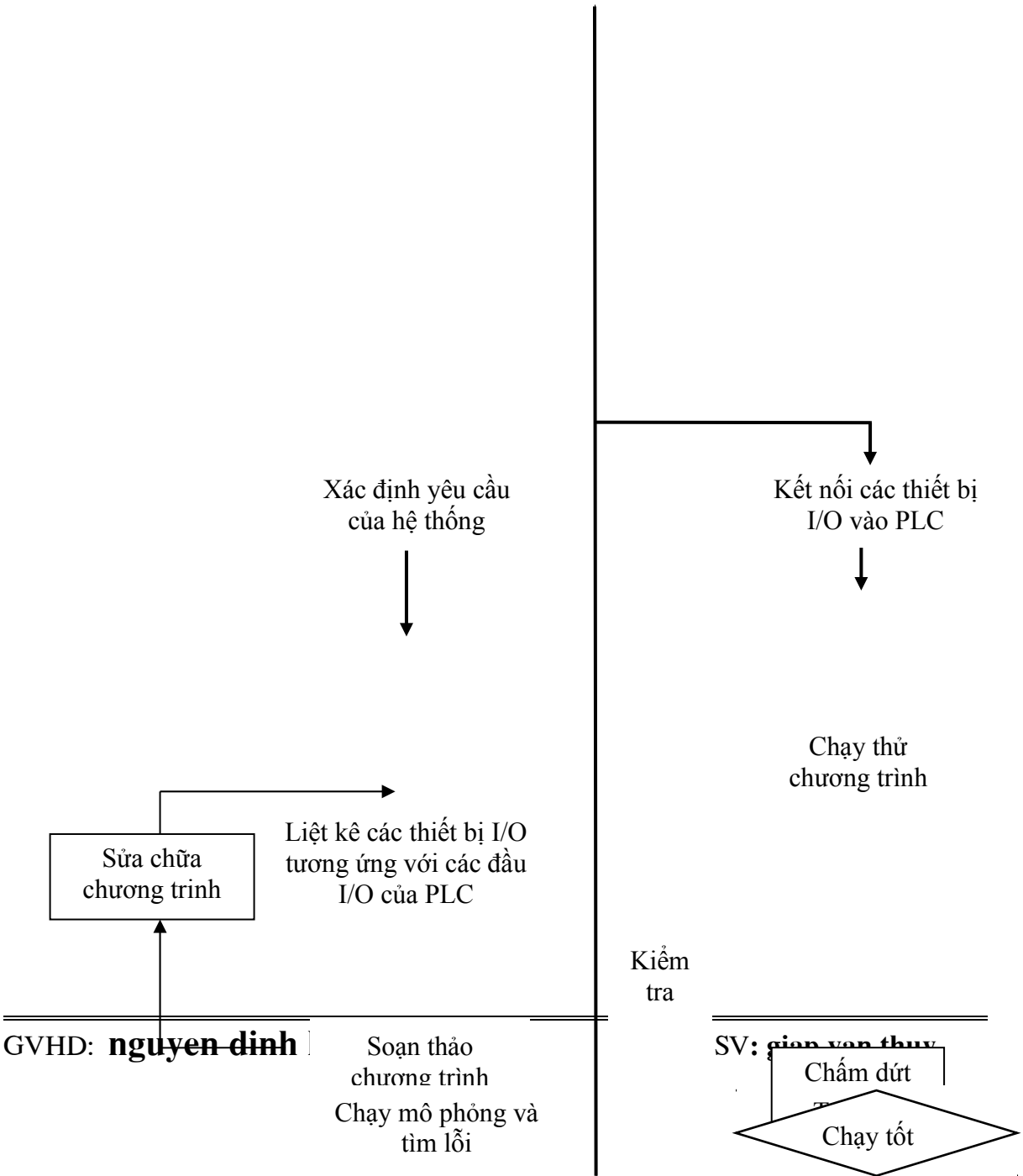
Chương trình điều khiển được soạn thảo dưới dạng lưu đồ hình thang.

d. Nạp chương trình vào bộ nhớ.

Cấp nguồn cho PLC, cài đặt cấu hình khối giao tiếp I/O nếu cần. Sau đó nạp chương trình soạn thảo trên màn hình vào bộ nhớ của PLC. Sau khi hoàn tất nên kiển tra lỗi bằng chức năng chuẩn đoán và nếu có thể thì chạy chương trình mô phỏng của hệ thống.

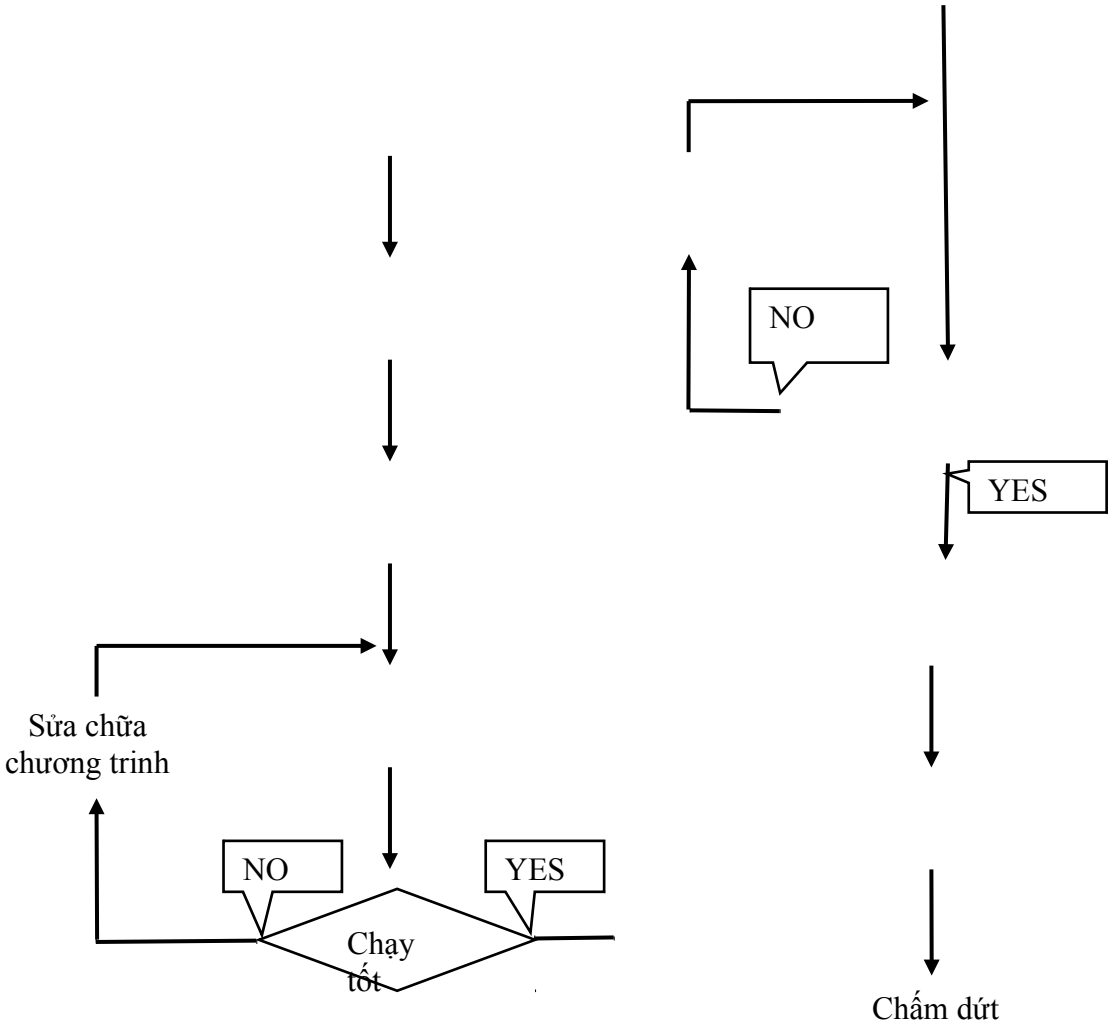
e. Chạy chương trình.

Trước khi khởi động hệ thống cần phải chắc chắn dây nối tử PLC đến các thiết bị ngoại vi là đúng. Trong quá trình chạy kiển tra có thể cần thiết thực hiện các bước chỉnh hệ thống nhằm đảm bảo an toàn khi đưa vào hoạt động thực tế.



Kiểm tra dây
nối

Vẽ lưu đồ điều
khiển



Quy trình thiết kế một hệ điều khiển tự động

II. Ngôn ngữ lập trình cho PLC S7 - 300.

Để viết chương trình điều khiển trên PLC có 3 phương pháp cơ bản là:

-Sơ đồ hình thang LAD (*Ladderr Diagram*).

-Lưu đồ hệ thống điều khiển FBD (*Function Block Diagram*).

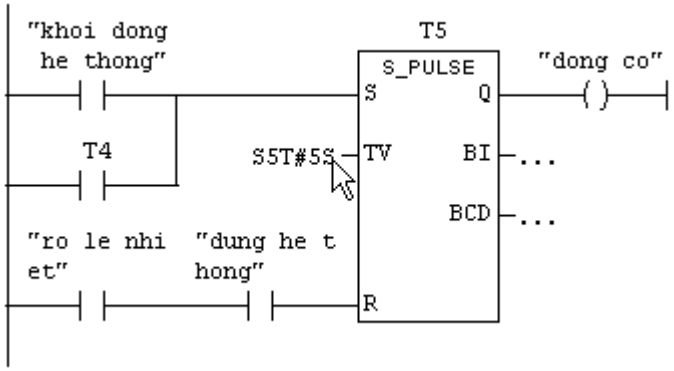
-Liệt kê lệnh STL (*Statement List*).

Một chương trình viết trên LAD hoặc FBD có thể chuyển sang STL, nhưng không xảy ra ngược lại vì trong STL có nhiều lệnh không có trong LAD hay FBD.

1. Phương pháp lập trình bằng LAD.

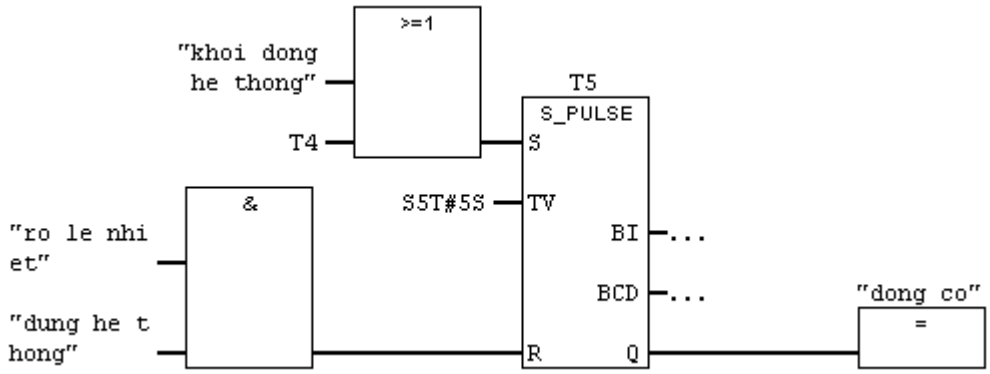
Phương pháp này có cách biểu diễn chương trình tương tự như sơ đồ tiếp điểm dùng Rơle trong sơ đồ điện công nghiệp.VD:

Sơ đồ điều khiển nối cứng dùng Rơle được biểu diễn bằng phương pháp LAD.:



2. Phương pháp lập trình bằng FBD.

Phương pháp này có cách biểu diễn dưới dạng liên kết của các hàm lôgic kỹ thuật số, loại ngôn ngữ này thích hợp cho những người quen sử dụng và thiết kế mạch điều khiển số. VD:



3. Phương pháp lập trình theo ngôn ngữ STL.

Phương pháp này là ngôn ngữ lập trình theo kiểu liệt kê các câu lệnh thành tập hợp lệnh, mỗi lệnh thực hiện một chức năng. Tương tự với ngôn ngữ Assembler ở máy tính, phương pl ượng làm việc trong lĩnh vực tin học. VD:

```
A(
O     "khoi dong he thong"
O     T      4
)
L     S5T#5S
SP    T      5
A     "ro le nhiet"
A     "dung he thong"
R     T      5
NOP   0
NOP   0
A     T      5
=     "dong co"
```

III. Lập trình và chọn chế độ làm việc cho PLC S7-300.

1.Giới thiệu chung.

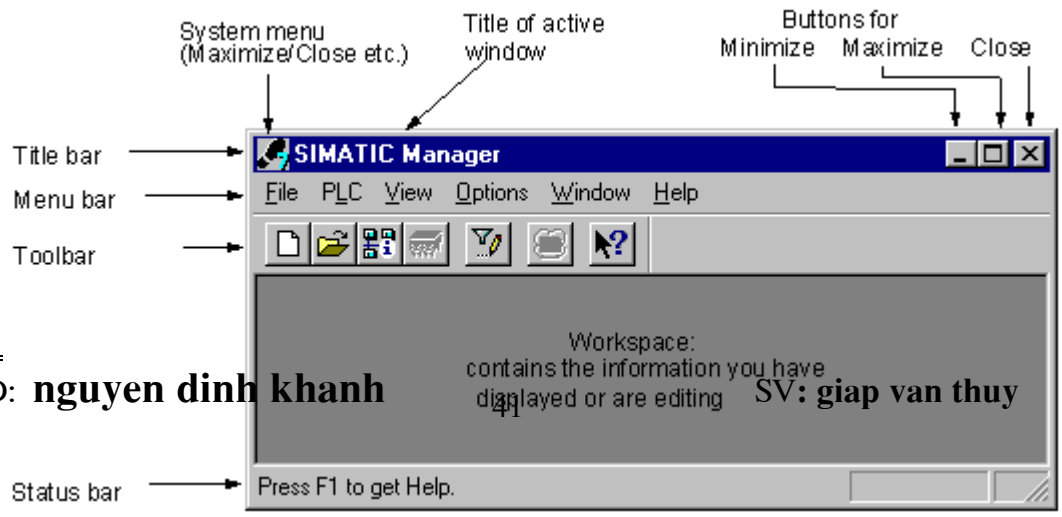
Lập trình có nghĩa là nhập một mạch vào trong phần mềm của PLC S7-300. Đây thực ra là cách biểu diễn khác của sơ đồ mạch. Chúng ta viết chương trình trên phần mềm soạn thảo Simentic S7 một cách ngắn gọn và phù hợp nhất.

Trên phần mềm soạn thảo này, sơ đồ mạch điều khiển có thể được viết theo các ngôn ngữ khác nhau như bằng ngôn ngữ LAD, FBD, STL…và một điểm cần lưu ý là với Simentic S7-300 ta thường soạn thảo chương trình trên khối OB1.

2. Lập trình trên Simentic S7-300.

2.1.Chọn giao diện cho PLC.

Muốn chọn giao diện nào, ta đánh dấu bộ giao diện đó ở phía trái rồi ấn phím Install… Bộ giao diện đã được chọn sẽ được ghi vào ô bên phải. Sau khi chọn xong bộ giao diện sử dụng, ta còn phải cài đặt tham số làm việc cho bộ giao diện bao gồm tốc độ truyền, cổng ghép nối máy tính..



2.2. Khai báo và mở một ProJect mới.

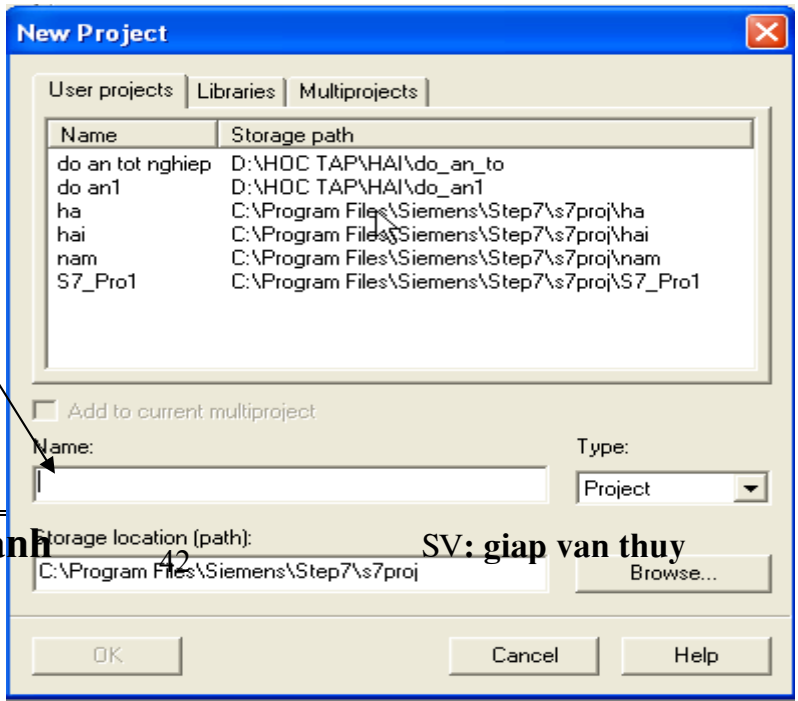
Từ giao diện của PLC chọn File -> New hoặc kích chuột vào biểu tượng "New Project/Library'

Khai báo một Projest mới

Nơi mở một Project đã có

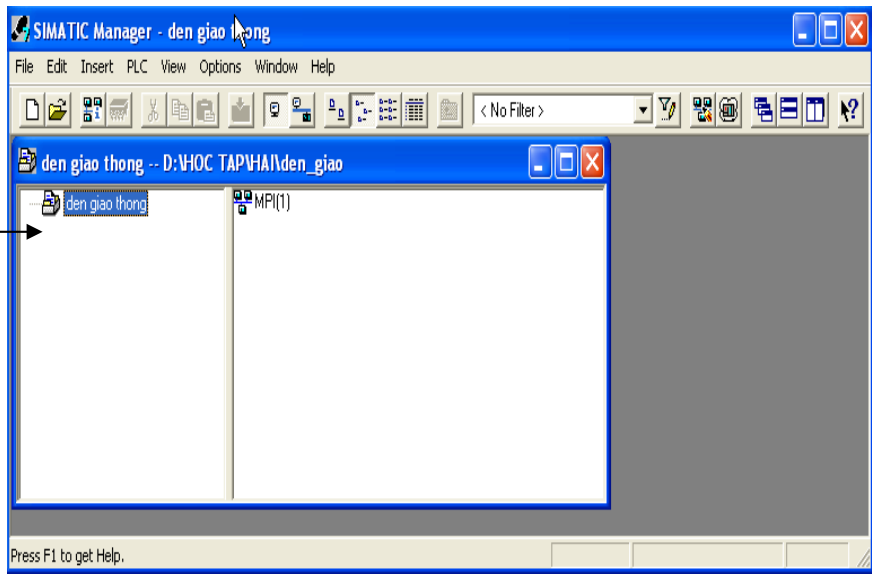
Khi đó trên màn hình sẽ xuất hiện hộp thoại, gõ tên Project rồi ấn phím OK và như vậy ta đã khai báo xong một Projeck mới. Ta cũng có thể chọn nơi cất Project mới, mặc định nơi cất là thư mục C:\siemens\step7\S7 Proj.

N ơi viết tên một Project



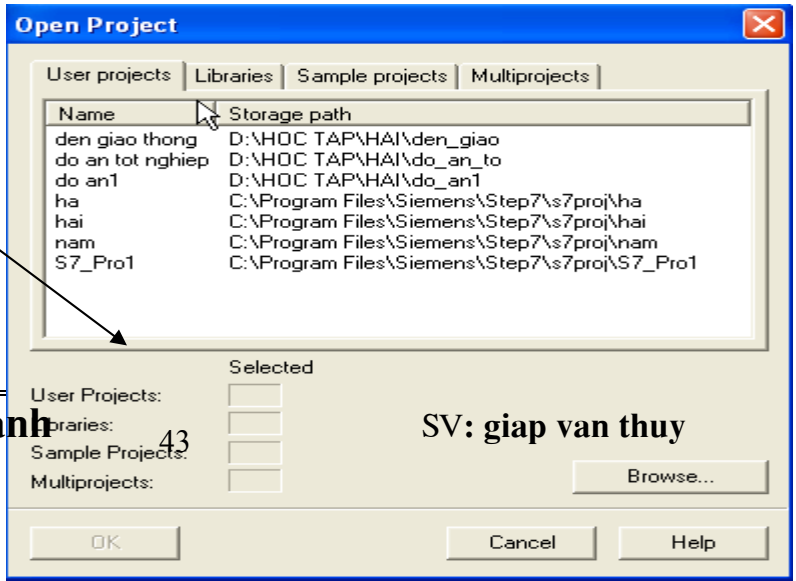
Sau khi khai báo xong một Project mới thì trên màn hình xuất hiện Project đó nhưng ở dạng rỗng (chưa có gì), như hình vẽ:

Biểu tượng một thư mục rỗng



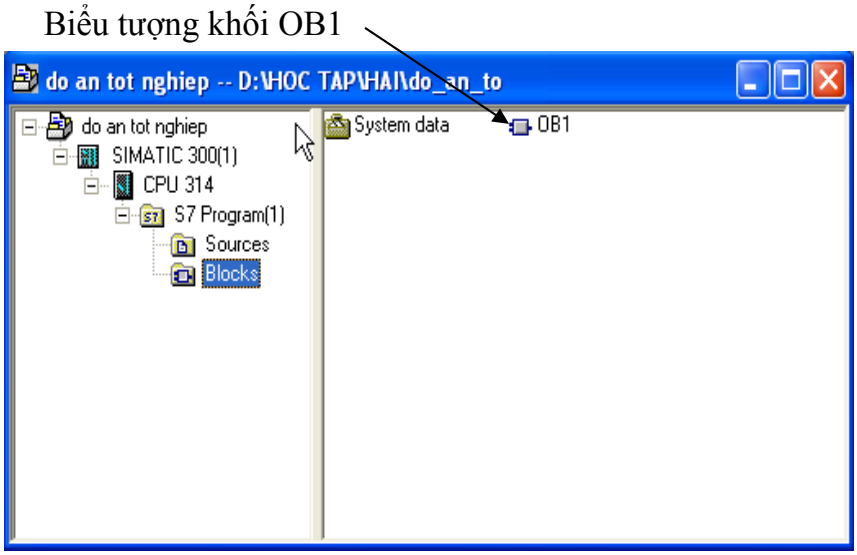
Trong trường hợp muốn mở một Project mới ta chọn File -> Open, hoặc kích chuột vào biểu tượng "Open Project/Library" rồi chọn tên muốn mở sau đó ấn phím OK.

Tên một Project đã có sẵn



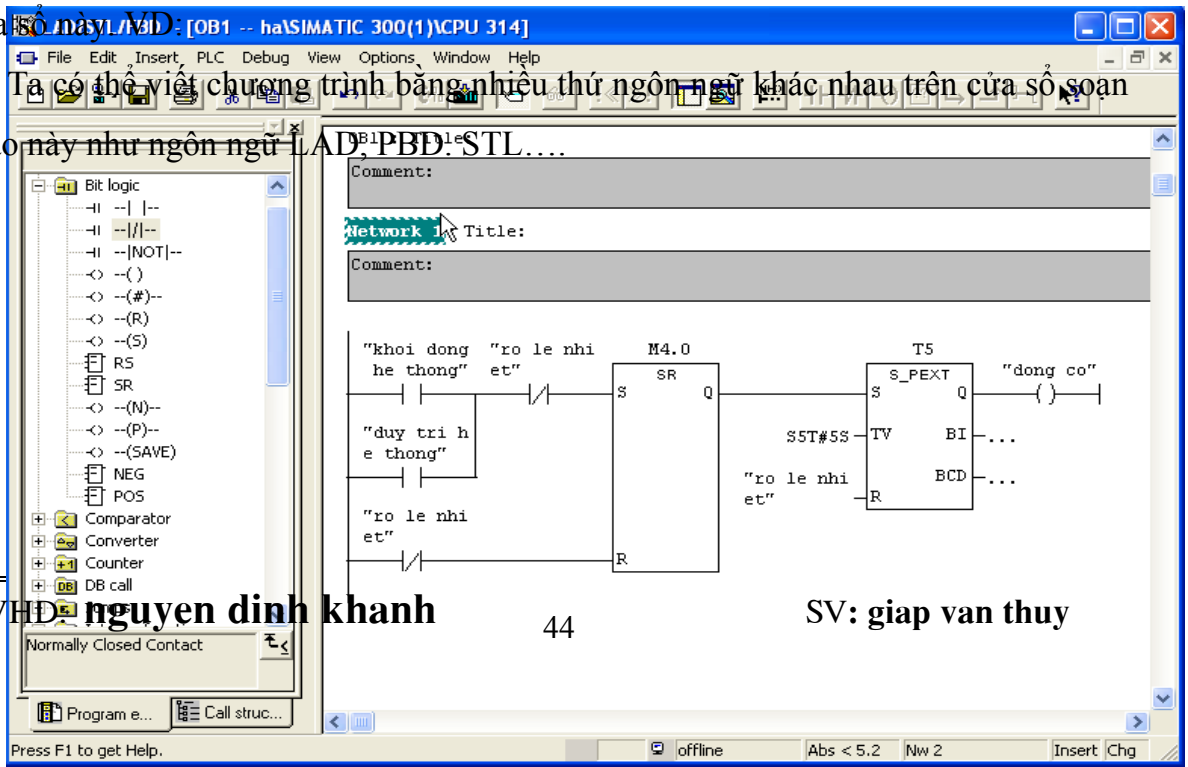
2.3. Soạn thảo chương trình trên khối OB1.

Ta nháy chuột vào biểu tượng của khối OB1 ở của sổ bên phải. như hình vẽ:



Khi ấy trên màn hình sẽ xuất hiện cửa sổ, ta viết chương trình điều khiển trên cửa sổ này. VD:

Ta có thể viết chương trình bằng nhiều thứ ngôn ngữ khác nhau trên cửa sổ soạn thảo này như ngôn ngữ LAD, PBD. STL….

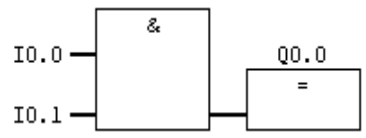
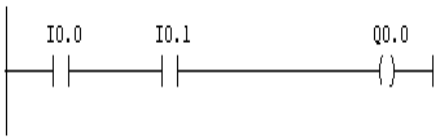


Khi lập trình xong có thể chạy thử chương trình bằng cách: vào biểu tượng **Simulation** -> Download -> chọn số lượng đầu vào và số lượng đầu ra -> Run_p -> quay lại màn hình soạn thảo kích vào biểu tượng Monitor.

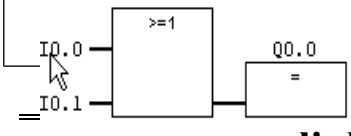
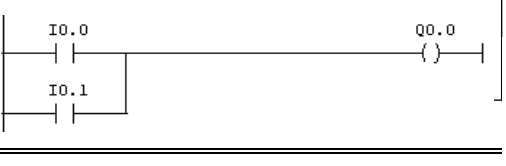
IV. Các khối, hàm và chức năng của nó trong PLC.

1. Các hàm lôgic tiếp điểm.

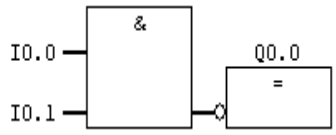
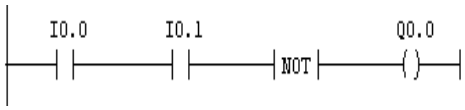
- Hàm AND: tín hiệu ra bằng 1 khi tất cả các tín hiệu vào bằng 1.

FBD	LAD	STL
		A I 0.0 A I 0.1 = Q 0.0

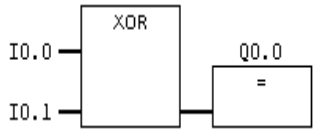
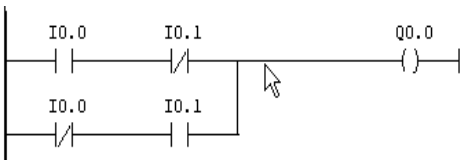
- Hàm OR: Tiệu ra bằng 1 khi một trong các tín hiệu vào bằng 1.

FBD	LAD	STL
		O I 0.0 O I 0.1 = Q 0.0

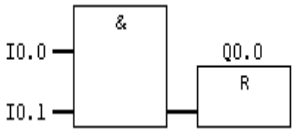
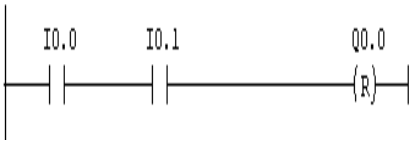
- Hàm NOT: Tín hiệu ra là đảo của tín hiệu vào:

FBD	LAD	STL
		A I 0.0 A I 0.1 NOT = Q 0.0

- Hàm XOR: tín hiệu ra bằng 1 khi hai tin hiệu vào khác nhau:

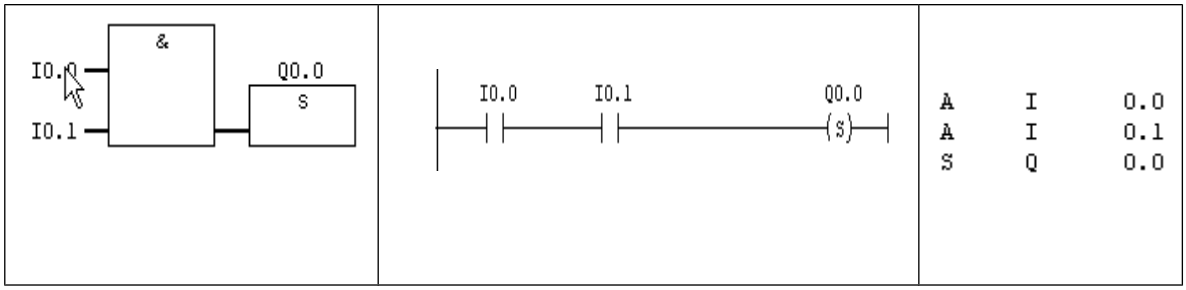
FBD	LAD	STL
		X I 0.0 X I 0.1 = Q 0.0

-Lệnh xoá RESET: Tín hiệu ra bị xoá khi co tín hiệu vào.

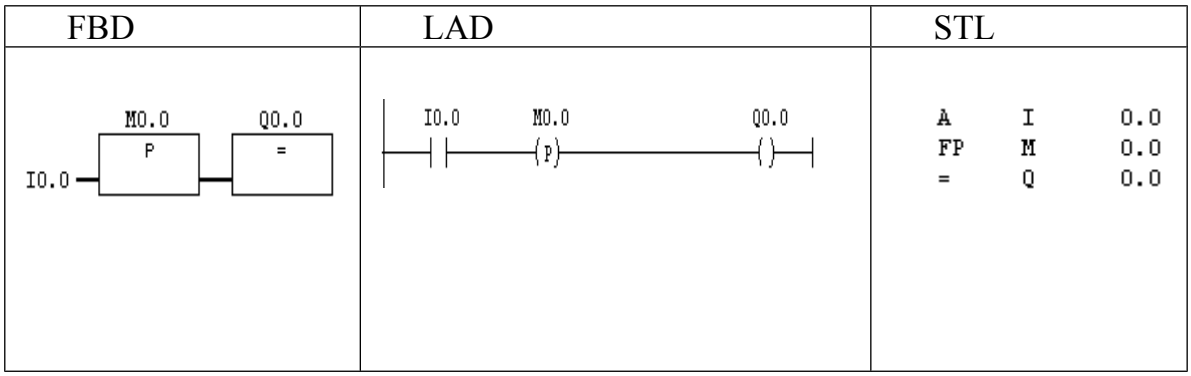
FBD	LAD	STL
		A I 0.0 A I 0.1 R Q 0.0

- Lệnh SET: Tín hiệu ra bằng 1 khi có tín hiệu vào (tín hiệu này được lưu giữ cả khi không có tín hiệu vào):

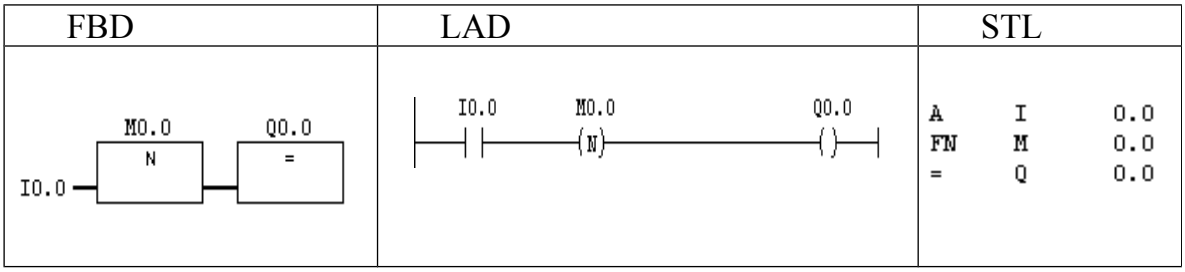
FBD	LAD	STL



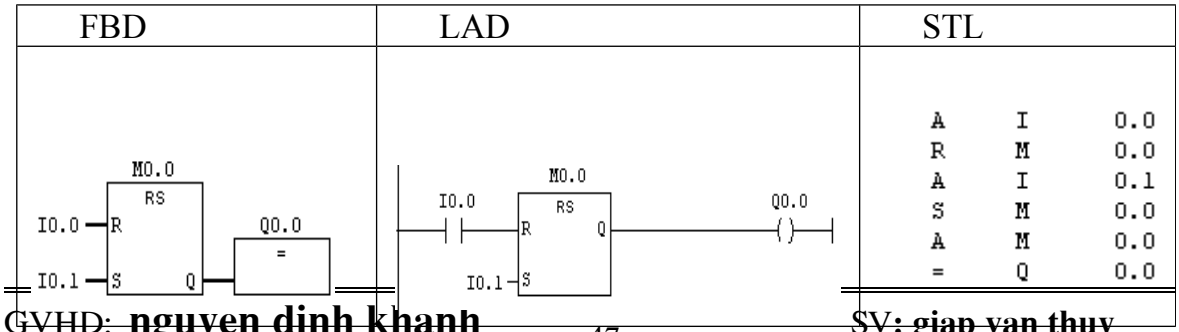
- Lệnh POSITIVE: Cho một xung có độ rộng bằng một vòng quét tại thời điểm có sườn lên của xung đầu vào:



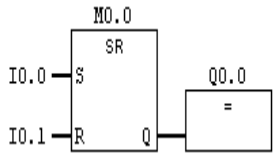
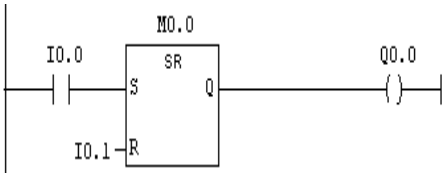
- Lệnh NEGAITIVE: Cho một xung có độ rộng bằng một vòng quét tại thời điểm có sườn xuống của xung đầu vào:



Bộ nhớ RS: Đầu ra bằng 1 khi đầu S bằng 1, đầu ra bằng 0 khi đầu R bằng 1, nếu R và S đều bằng 1 thì đầu ra bằng 1(ưu tiên SET):

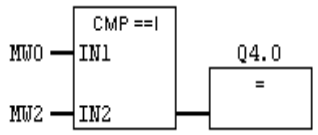
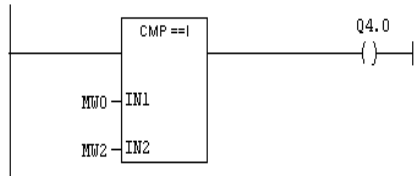


Bộ nhớ SR: đầu ra bằng 1 khi đầu vào S bằng 1, đầu ra bằng 0 khi đầu R bằng 1, nếu R và S bằng 1 thì đầu ra bằng 0 (ưu tiên R).

FBD	LAD	STL
		A I 0.0 S M 0.0 A I 0.1 R M 0.0 A M 0.0 = Q 0.0

2. Nhóm hàm so sánh.

* Nhóm hàm so sánh số nguyên 16 bít:

FBD	LAD	STL
		L MW 0 L MW 2 ==I = Q 4.0

Có các dạng so sánh hai số nguyên 16 bít như sau:

+ Hàm so sánh bằng nhau giữa hai số nguyên 16 bít: ==

+ Hàm so sánh khác nhau giữa hai số nguyên 16 bít: <>

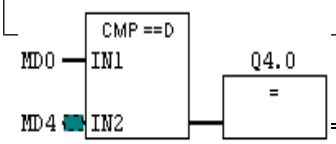
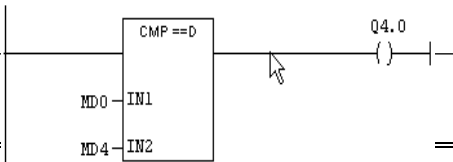
+ Hàm so sánh lớn hơn giữa hai số nguyên 16 bít: >

+ Hàm so sánh nhỏ hơn giữa hai số nguyên 16 bít: <

+ Hàm so sánh nhỏ hơn hoặc bằng nhau giữa hai số nguyên 16 bít: >=

+ Hàm so sánh lớn hơn hoặc bằng nhau giữa hai số nguyên 16 bít: <=

* Nhóm hàm so sánh số nguyên 32 bít:

FBD	LAD	STL
		L MD 0 L MD 4 ==D = Q 4.0

Có các dạng so sánh hai số nguyên 32 bít như sau:

+ Hàm so sánh bằng nhau giữa hai số nguyên 32 bít: ==

+ Hàm so sánh khác nhau giữa hai số nguyên 32 bít: <>

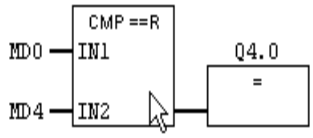
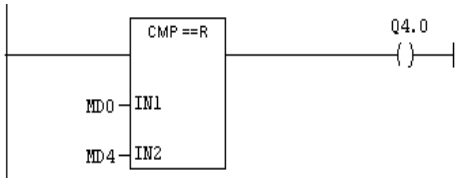
+ Hàm so sánh lớn hơn giữa hai số nguyên 32 bít: >

+ Hàm so sánh nhỏ hơn giữa hai số nguyên 32 bít: <

+ Hàm so sánh nhỏ hơn hoặc bằng nhau giữa hai số nguyên 32 bít: >=

+ Hàm so sánh lớn hơn hoặc bằng nhau giữa hai số nguyên 32 bít: <=

* Nhóm hàm so sánh số thực 32 bít:

FBD	LAD	STL
		L MD 0 L MD 4 ==R = Q 4.0

Có các dạng so sánh hai số thực 32 bít như sau:

+ Hàm so sánh bằng nhau giữa hai số thực 32 bít: ==

+ Hàm so sánh khác nhau giữa hai số thực 32 bít: <>

+ Hàm so sánh lớn hơn giữa hai số thực 32 bít: >

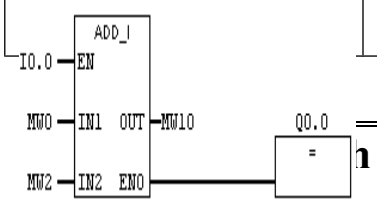
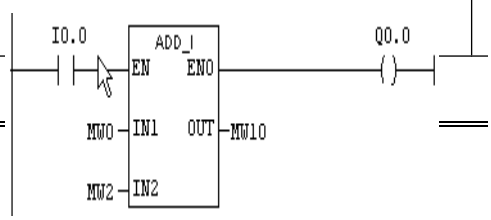
+ Hàm so sánh nhỏ hơn giữa hai số thực 32 bít: <

+ Hàm so sánh nhỏ hơn hoặc bằng nhau giữa hai số thực 32 bít: >=

+ Hàm so sánh lớn hơn hoặc bằng nhau giữa hai số thực 32 bít: <=

3. Các hàm toán học.

1. Cộng hai số nguyên 16 bít:

FBD	LAD	STL
		A I 0.0 JNB _001 L MW 0 L MW 2 +I T MW 10 AN OV SAVE CLR _001: A BR = Q 0.0

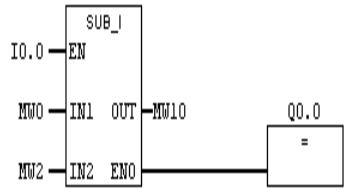
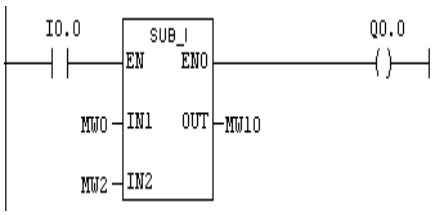
Dữ liệu vào và ra:

EN: BOOL IN1: INT

IN2: INT OUT: INT EN0: BOOL

Khi tín hiệu vào I0.0 = 1 đầu ra Q0.0 = 1 và hàm sẽ thực hiện cộng hai số nguyên 16 bít MW0 với MW2. Kết quả được cất vào MW10. Trường hợp tín hiệu vào I0.0 đầu ra Q0.0 và hàm sẽ không được thực hiện chức năng này.

2. Trừ hai số nguyên 16 bít:

FBD	LAD	STL
		A I 0.0 JNB _001 L MW 0 L MW 2 -I T MW 10 AN OV SAVE CLR _001: A BR = Q 0.0

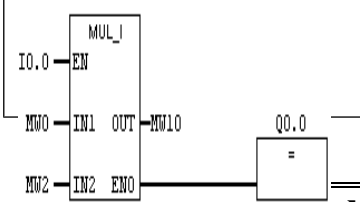
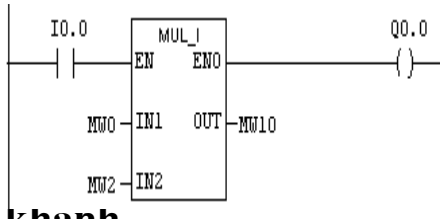
Dữ liệu vào và ra:

EN: BOOL IN1: INT

IN2: INT OUT: INT EN0: BOOL

Khi tín hiệu vào I0.0 = 1 đầu ra Q0.0 = 1 và hàm sẽ thực hiện trừ hai số nguyên 16 bít MW0 với MW2. Kết quả được cất vào MW10. Trường hợp tín hiệu vào I0.0 đầu ra Q0.0 và hàm sẽ không được thực hiện chức năng này.

3. Nhân hai số nguyên 16 bít:

FBD	LAD	STL
		A I 0.0 JNB _001 L MW 0 L MW 2 *I T MW 10 AN OV SAVE CLR _001: A BR = Q 0.0

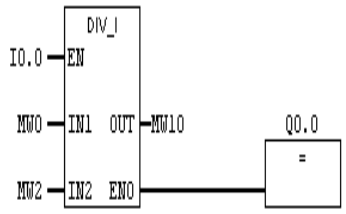
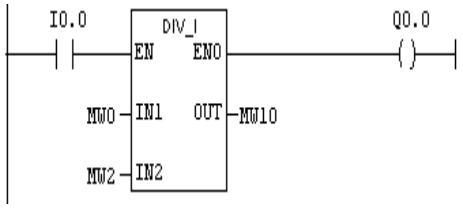
Dữ liệu vào và ra:

EN: BOOL IN1: INT

IN2: INT OUT: INT EN0: BOOL

Khi tín hiệu vào I0.0 = 1 đầu ra Q0.0 = 1 và hàm sẽ thực hiện nhân hai số nguyên 16 bít MW0 với MW2. Kết quả được cất vào MW10. Trường hợp tín hiệu vào I0.0 đầu ra Q0.0 và hàm sẽ không được thực hiện chức năng này.

4. Chia hai số nguyên 16 bít:

FBD	LAD	STL
		A I 0.0 JNB _001 L MW 0 L MW 2 /I T MW 10 AN OV SAVE CLR _001: A BR = Q 0.0

Dữ liệu vào và ra:

EN: BOOL IN1: INT

IN2: INT OUT: INT EN0: BOOL

Khi tín hiệu vào I0.0 = 1 đầu ra Q0.0 = 1 và hàm sẽ thực hiện chia hai số nguyên 16 bít MW0 với MW2. Kết quả được cất vào MW10. Trường hợp tín hiệu vào I0.0 đầu ra Q0.0 và hàm sẽ không được thực hiện chức năng này.

4. Nhóm hàm đổi kiểu dữ liệu.

Trong ngôn ngữ lập trình của S7- 300 có một kiểu dữ liệu khác nhau như:

-Số nguyên 16 bít (Interger)

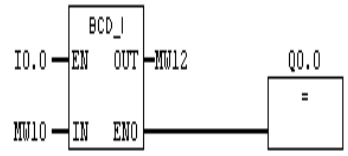
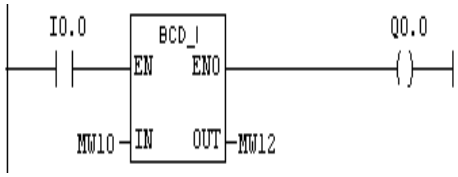
-Số nguyên 32 bít (DI)

-Số nguyên dạng BCD

- Và các dạng các dữ liệu khác.

Khi làm việc với nhiều dạng dữ liệu khác nhau cho ta cần phải chuyển đổi chúng. Ví dụ khi đọc tín hiệu từ cổng vào tương tự ta nhận được số liệu dạng số nguyên 16 bít mang giá trị tín hiệu tương tự chứ không phải giá trị bản thân đó, bởi vậy để phải xử lý tiếp thì cần phải chuyển đổi số nguyên đó thành đúng giá trị thực, dấu phảy động của tín hiệu ở cổng. Ta có một số hàm chuyển đổi các dạng dũ liệu như sau:

a. Hàm chuyển số BCD thành số nguyên 16 bít:

FBD	LAD	STL
		A I 0.0 JNB _001 L MW 10 BTI T MW 12 SET SAVE CLR _001: A BR = Q 0.0

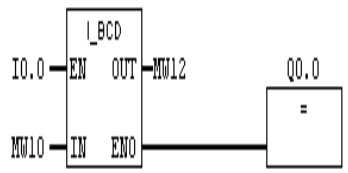
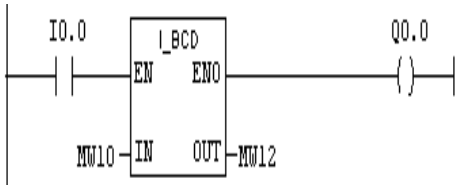
Dữ liệu vào và ra:

EN: BOOL IN: WORD

OUT: INT ENO: BOOL

Khi tín hiệu vào I0.0 = 1 đầu ra Q0.0 = 1 và hàm thực hiện chức năng chuyển số BCD (MW10) sang số nguyên rồi cất vào MW12. Khi tín hiệu vào I0.0 = 0 vào đầu ra Q0.0 = 0 và hàm không thực hiện chức năng chuyển đổi.

b. Hàm chuyển số nguyên 16 bít thành số BCD:

FBD	LAD	STL
		A I 0.0 JNB _001 L MW 10 ITB T MW 12 AN OV SAVE CLR _001: A BR = Q 0.0

Dữ liệu vào và ra:

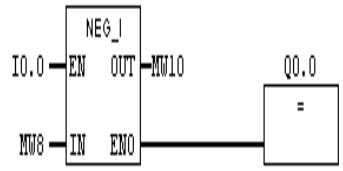
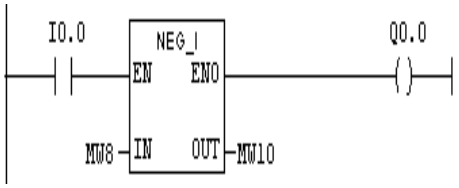
EN: BOOL IN: WORD

OUT: INT ENO: BOOL

Khi tín hiệu vào I0.0 = 1 đầu ra Q0.0 = 1 và hàm thực hiện chức năng chuyển số nguyên (MW10) sang số BCD rồi cất vào MW12. Khi tín hiệu vào I0.0 = 0 vào đầu ra Q0.0 = 0 và hàm không thực hiện chức năng chuyển đổi.

5. Các hàm đổi dấu.

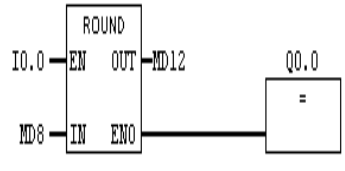
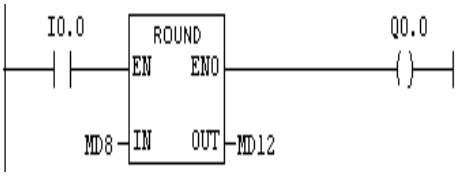
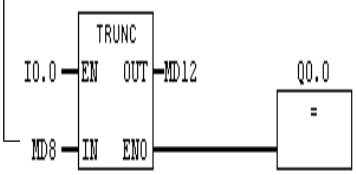
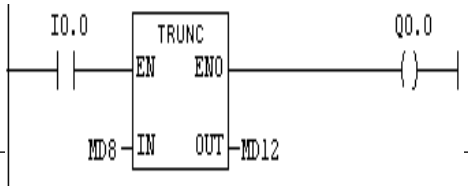
Hàm sẽ thực hiện chức năng đổi dấu dữ liệu vào. Các hàm đổi dấu như đổi dấu số thực 16 bít, 32 bít, hay số nguyên (R).

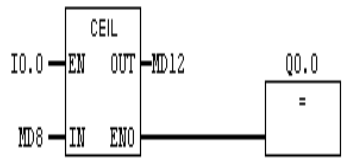
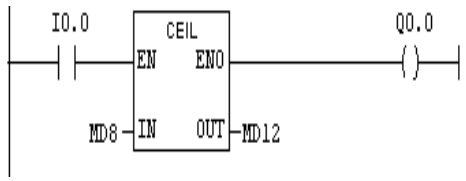
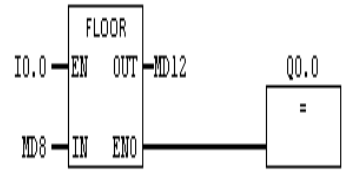
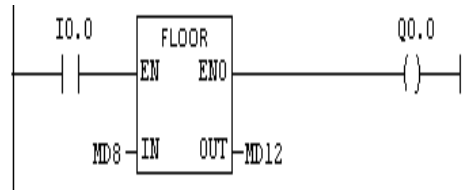
FBD	LAD	STL
		A I 0.0 JNB _001 L MW 8 NEGI T MW 10 AN OV SAVE CLR _001: A BR = Q 0.0

Dạng dữ liệu vào:

	NEG_I	NEG_DI	NEG_R
EN	BOOL	BOOL	BOOL
IN	INT	DI	REAL
OUT	INT	DI	REAL
ENO	BOOL	BOOL	BOOL

6. Các hàm thực hiện chức năng làm tròn (đổi liểu dữ liệu):

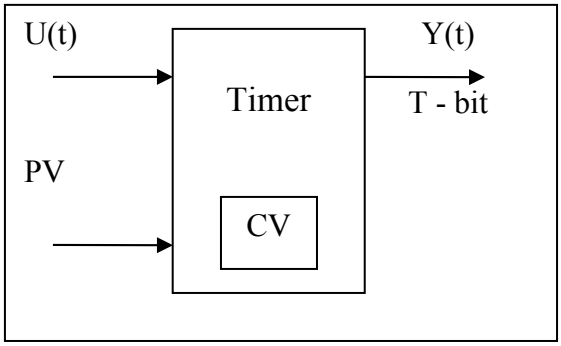
FBD	LAD	STL
		A I 0.0 JNB _001 L MD 8 RND T MD 12 AN OV SAVE CLR _001: A BR = Q 0.0
		A I 0.0 JNB _001 L MD 8 TRUNC T MD 12 AN OV SAVE CLR _001: A BR = Q 0.0

		A I 0.0 JNB _001 L MD 8 RND+ T MD 12 AN OV SAVE CLR _001: A BR = Q 0.0
		A I 0.0 JNB _001 L MD 8 RND- T MD 12 AN OV SAVE CLR _001: A BR = Q 0.0

V. bộ thời gian.

1.Nguyên tắc làm việc.

Bộ thời gian Timer là bộ tạo thời gian chễ T mong muốn giữa tín hiệu logic đầu vào U(t) và đầu ra Y(t).



S7 – 300 có 5 bộ thời gian timer khác nhau. Tất cả 5 loại này cùng bắt đầu tạo thời gian trễ tín hiệu kể từ thời điểm có sườn lên của tín hiệu đầu vào tức là khi có tín hiệu đầu vào U(t) chuyển trạng thái từ logic “0” lên logic “1”, được gọi là thời điểm Timer được kích.

Thời gian trễ T mong muốn được khai báo với Timer bằng giá trị 16 bít bao gồm hai thành phần:

- Độ phân giải với đơn vị là mS. Timer của S7 có 4 loại phân giải khác nhau là 10ms, 100ms, 1s, 10s.

- Một số nguyên BCD trong khoảng từ 0 đến 999 được gọi là PV (Preset Value – giá trị đặt trước).

Như vậy thời gian trễ T mong muốn sẽ được tính như sau:

T = Độ phân giải * PV

Cấu hình thời gian trễ đặt trước cần khai báo bộ Timer

Ngay tại thời điểm kích Timer, giá trị PV được chuyển vào thanh ghi 16 bít của Timer T- Word (gọi là thanh ghi CV- Curren value – giá trị tức thời). Timer sẽ ghi nhớ khoảng thời gian trôi qua kể từ khi kíchbằng cách giảm dần một cách tương ứng nội dung thanh ghi CV. Nếu nội dung thanh ghi CV trở về bằng 0 thì Timer đạt được thời gian mong muốn T và điều này được báo ra ngoài bằng cách thay đổi trạng thái tín hiệu đầu ra Y(t). Việc thông báo ra ngoài bằng cách như thế nào còn phụ thuộc vào loại Timer được sử dụng.

Bên cạnh sườn lên của tín hiệu đầu vào U(t). Timer còn kích bằng sườn lên của tín hiệu chủ động có tên là tín hiệu ENABLE nếu như tại thời điểm có tín hiệu sườn lên ENABLE, tín hiệu đầu vào U(t) có logic “1”.

Từng loại Timer được đánh số từ 0 đến 255 (tuỳ thuộc vào từng loại CPU). Một Timer được đặt tên Tx, trong đó x là số hiệu bộ Timer. Ký hiệu Tx cũng đồng thời là tín hiệu hình thức của thanh ghi CV (T- Word) và đầu ra T- bit của Timer đó. Tuy chúng có cùng địa chỉ hình thức nhưng T- Word và T-bit vẫn được phân biệt

với nhau nhờ lệnh sử dụng toán hạng Tx. Khi làm việc với từ Tx được hiểu là T-Word còn khi làm việc với điểm thì Tx được hiểu là T- bit.

Để xoá tức thời trạng thái của T- Word và T- bit người ta sử dụng một tín hiệu Reset Timer. Tại thời điểm sườn lên của tín hiệu này giá trị T- Word và T- bit đồng thời giá trị bằng 0 tức là thanh ghi tức thời CV được đặt về 0 và tín hiệu đầu ra cũng ở trạng thái logic 0. Trong thời gian tín hiệu Reset có giá tị logic là 1 Timer sẽ không làm việc.

2. Khai báo sử dụng.

Các tín hiệu điều khiển cho bộ Timer phải được khai báo theo các bước sau đây:

- Khai báo tín hiệu ENABLE nếu muốn tín hiệu chủ động kích.
- Khai báo tín hiệu đầu vào U(t)
- Khai báo thời gian trễ mong muốn TW.
- Khai báo loại Timer được sử dụng (SP,SE, SD, SS, SF)
- Khai báo tín hiệu xóa Timer nếu muốn sử dụng chế độ Reset chủ động.

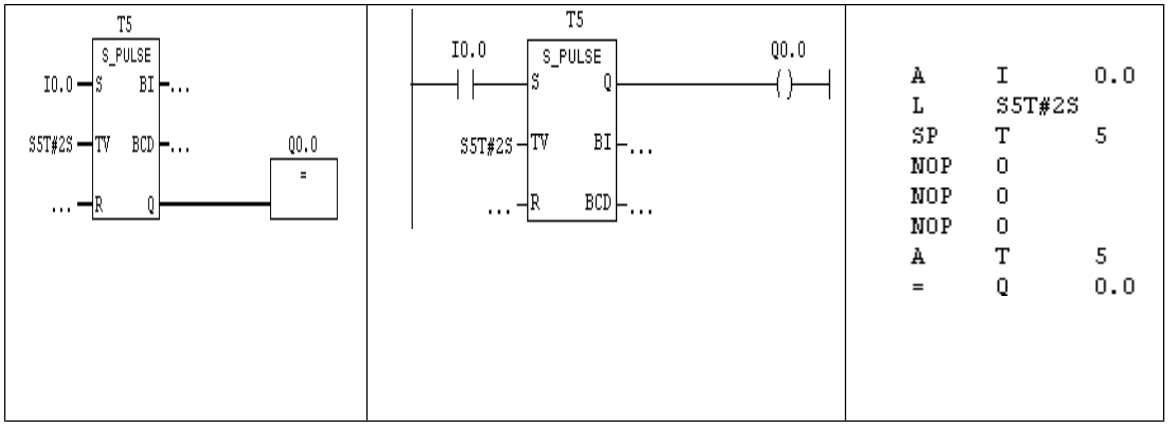
Trong các bước trên thì bước 1 và 5 có thể bỏ qua.

- Dạng dữ liệu vào ra của bộ Timer:

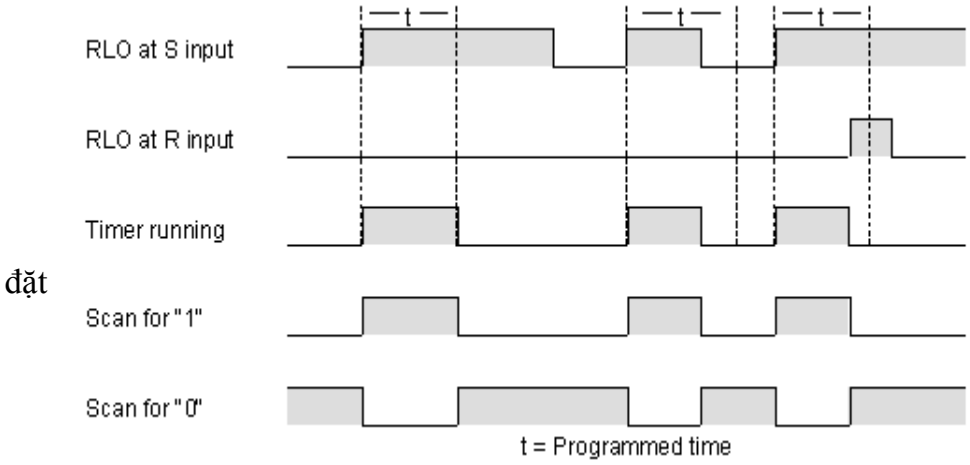
S: BOOL	BI (DUAL): WORD
TW: S5TIME	BCD(DEZ): WORD
R: BOOL	Q:BOOL

- Bộ thời gian SP: Tại thời điểm sườn lên của tín hiệu vào SET thời gian sẽ được tính, đồng thời giá trị lôgic ở đầu ra là 1. Khi thời gian đặt kết thúc giá trị đầu ra trở về 0:

FBD	LAD	STL



Giản đồ thời gian



Tín hiệu S

Tín hiệu R

Thời gian đặt

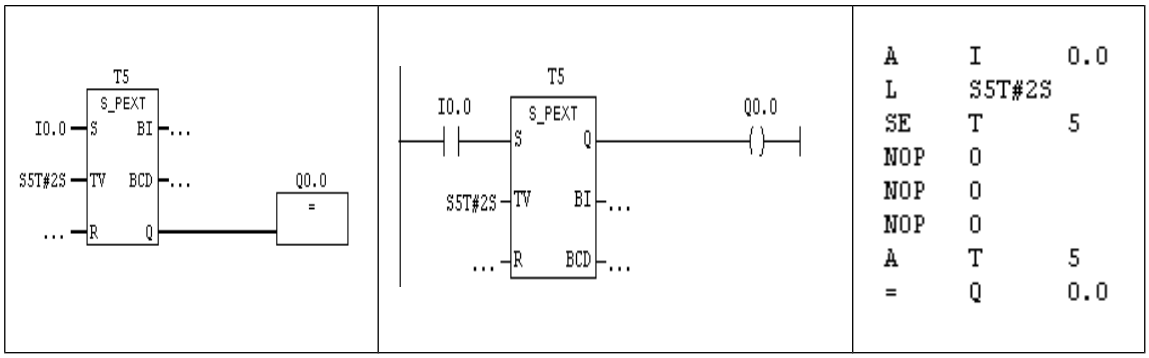
Đầu ra

Đầu ra đảo

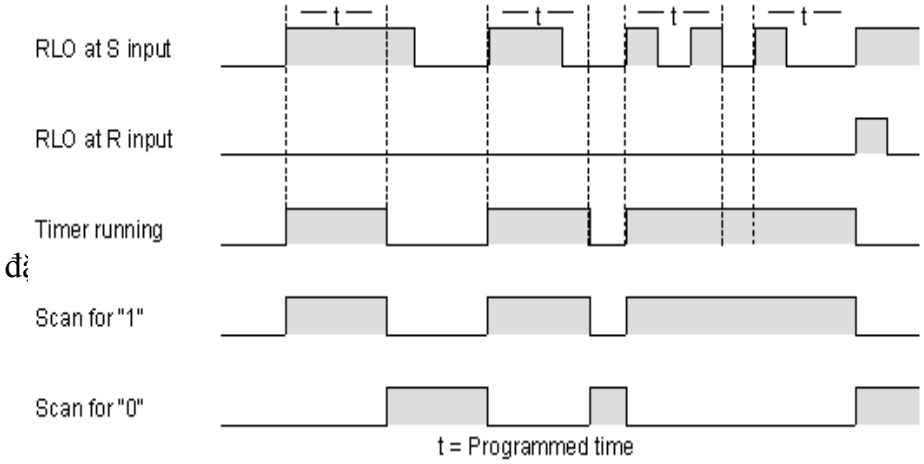
Khi có tín hiệu RESET (R) thời gian tính lập tức trở về 0 và tín hiệu đầu ra cũng bằng có giá trị "0".

- Bộ thời gian SE: Tại thời điểm sườn lên của tín hiệu vào SET cuối cùng bộ thời gian được thiết lập và thời gian sẽ được tính đồng thời giá trị đầu ra là 1. Kết thúc thời gian đặt thì đầu ra bằng 0. Khi có tín hiệu RESET (R) thời gian tính lập tức trở về 0 và tín hiệu đầu ra cũng bằng có giá trị "0".

FBD	LAD	STL



Giản đồ thời gian:



Tín hiệu S

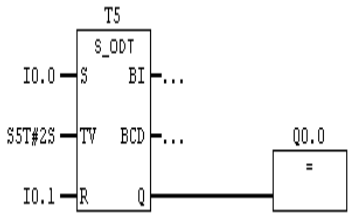
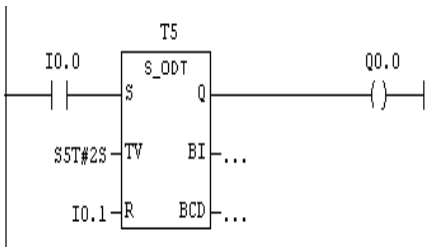
Tín hiệu R

Thời gian đặ

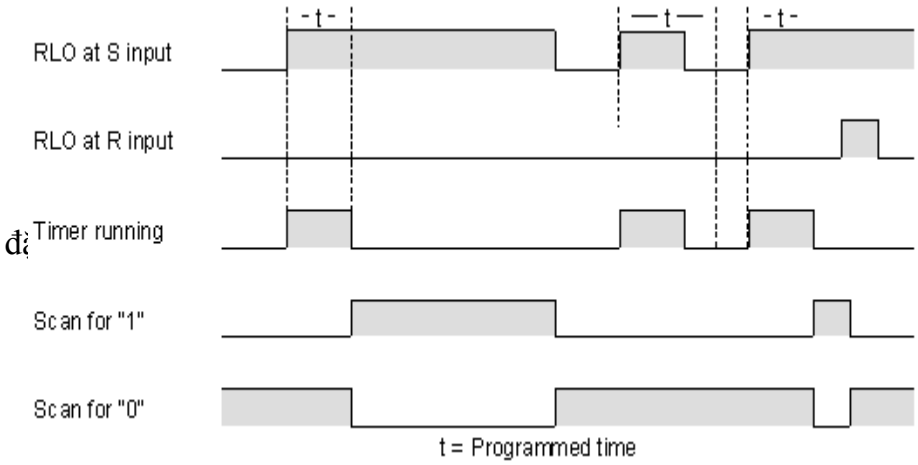
Đầu ra

Đầu ra đảo

- Bộ thời gian SD: Tại thời điểm sườn lên của tín hiệu vào SET bộ thời gian được thiết lập và thời gian sẽ được tính. Kết thúc thời gian đặt tín hiệu đầu ra sẽ có giá trị là 1. Khi S là 0 đầu ra cũng lập tưc trở về 0, nghĩa là tín hiệu đầu ra sẽ không được duy trì khi tín hiệu kích có giá trị là 0.

FBD	LAD	STL
		A I 0.0 L S5T#2S SD T 5 A I 0.1 R T 5 NOP 0 NOP 0 A T 5 = Q 0.0

Giản đồ thời gian:



Tín hiệu S

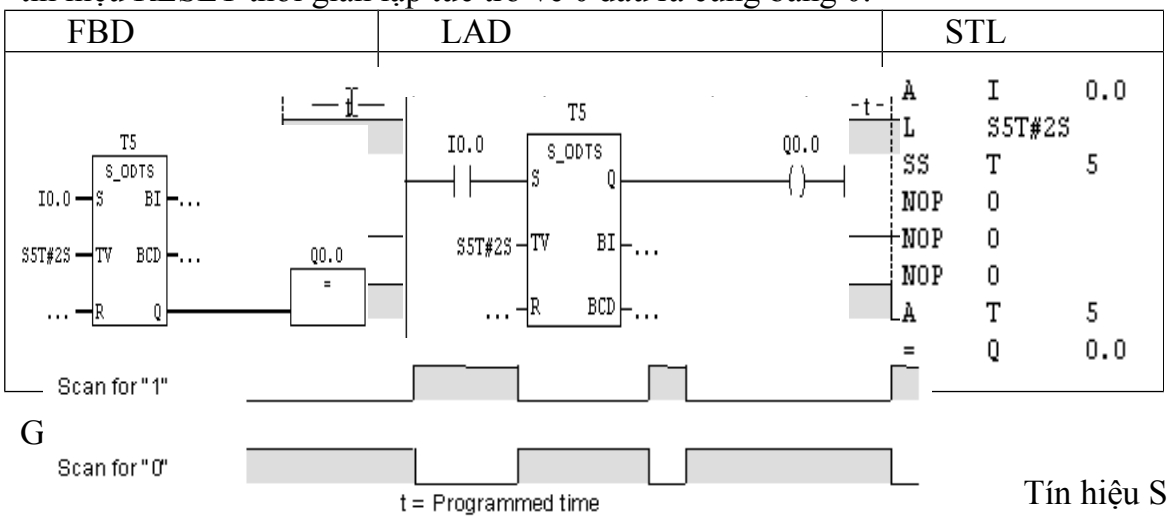
Tín hiệu R

Thời gian đặ

Đầu ra

Đầu ra đảo

- Bộ thời gian SS: Tại thời điểm sườn lên của tín hiệu vào SET bộ thời gian đực thiết lập và thời gian sẽ được tính. Kết thúc thời gian đặt tín hiệu đầu ra sẽ có giá trị 1, giá trị này vẫn được duy trì ngay cả khi tín hiệu vào kích S bằng 0. Khi có tín hiệu RESET thời gian lập tức trở về 0 đầu ra cũng bằng 0.

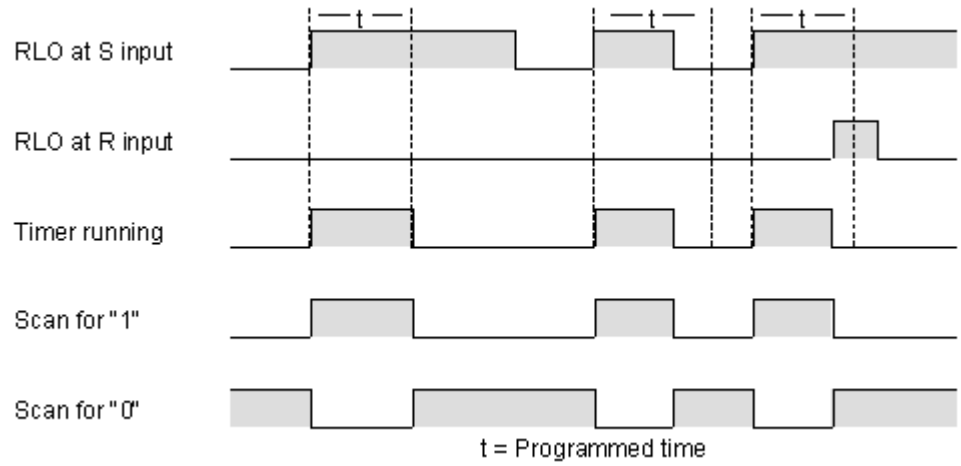


G

Tín hiệu S

Tín hiệu R

Thời gian đặt

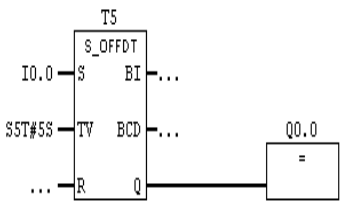
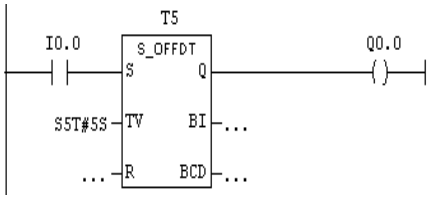


Đầu ra

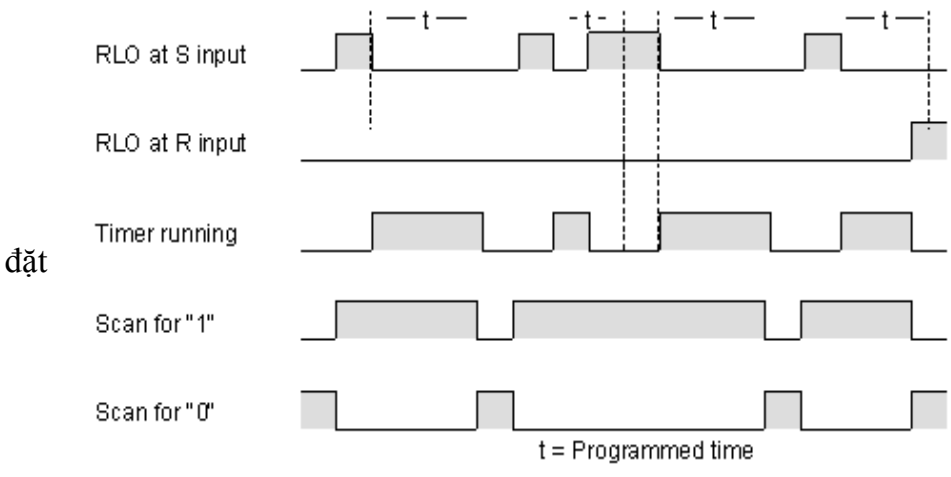
Đầu ra đảo

van thuy

- Bộ thời gian SF: Tại thời điểm sườn lên của tín hiệu vào SET bộ thời gian được thiết lập. Đầu ra có giá trị bằng1. Nhưng thời gian sẽ được tính ở thời điểm sườn xuống cuối cùng của tín hiệu đầu vào S. Kết thúc thời gian đặt thì đầu ra bằng 0:

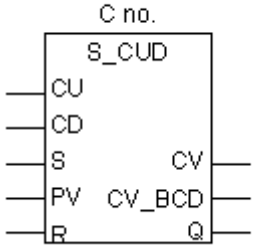
FBD	LAD	STL
		A I 0.0 L S5T#5S SF T 5 NOP 0 NOP 0 NOP 0 A T 5 = Q 0.0

Giản đồ thời gian:

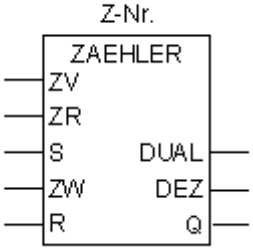


VII. Bộ đếm COUNTER.

English



German



Counter thực hiên chức năng đếm tại các sườn của các xung đầu vào. S7-300 có tối đa là 256 bộ đếm phụ thuộc vào từng loại CPU, ký hiệu Cx. Trong đó x là số nguyên trong khoảng từ 0 đến 255. Trong S7-300 có 3 loại bộ đếm thường sử dụng nhất đó là : Bộ đếm tiến(CU), Bộ đếm lùi(CD), Bộ đếm tiếm lùi(CUD).

CU: BOOL là tín hiệu đếm tiến

CD: BOOL là tín hiệu đếm lùi

S: BOOL là tín hiệu đặt

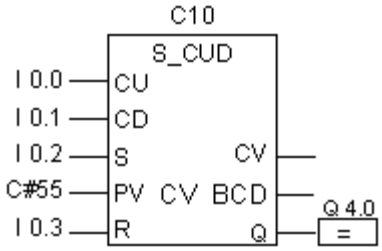
PV: WORD là giá trị đặt trước

R: BOOL là tín hiệu xoá

CV: WORD là giá trị đếm ở hệ đếm 16

CV_BCD: WORD là giá trị đếm ở hệ đếm BCD

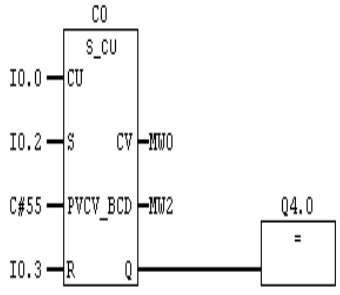
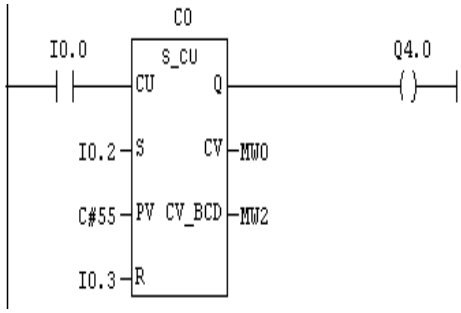
Q: BOOL là tín hiệu đầu ra



- Bộ đếm tiến CU:

Nguyên lý làm việc: Khi tín hiệu I0.2 chuyển từ 0 lên 1 bộ đếm được đặt giá trị là 55. Giá trị đầu ra Q4.0 =1.

Bộ đếm sẽ thực hiện đếm tiến tại các sườn lên của tín hiệu tại chân CU khi tín hiệu I0.0 chuyển giá trị từ 0 lên 1. Giá trị bộ đếm trở về 0 khi có tín hiệu tại sườn lên của chân R.

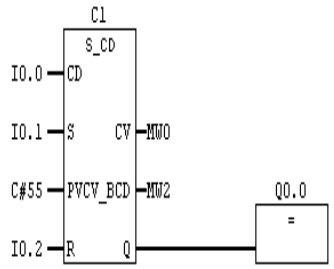
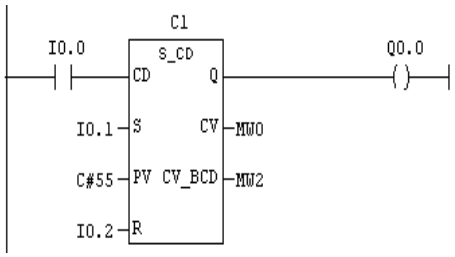
FBD	LAD	STL
		A I 0.0 CU C 0 BLD 101 A I 0.2 L C#55 S C 0 A I 0.3 R C 0 L C 0 T MW 0 LC C 0 T MW 2 A C 0 = Q 4.0

- Bộ đếm lùi CD:

Nguyên lý hoạt động: Khi tín hiệu I0.2chuyển từ 0 lên 1 bộ đếm được đặt giá trị là 55. Giá trị đầu ra Q4.0 = 1

Bộ đếm sẽ thực hiện đếm lùi tại các sườn lên của tín hiệu tại chân CD, khi tín hiệu I0.0 chuyển giá trị từ 0 lên 1. Giá trị của bộ đếm sẽ trở về 0 khi có tín hiệu tại sườn lên của chân R. Bộ đếm sẽ chỉ đếm đến giá trị >=0.

Sơ đồ khối:

FBD	LAD	STL
		A I 0.0 CD C 1 BLD 101 A I 0.1 L C#55 S C 1 A I 0.2 R C 1 L C 1 T MW 0 LC C 1 T MW 2 A C 1 = Q 0.0

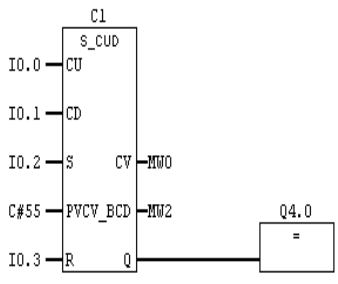
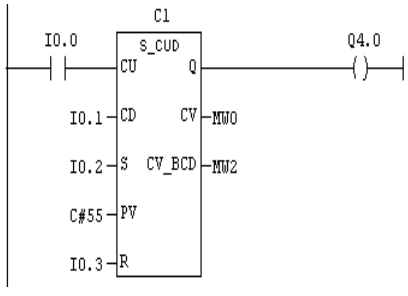
- Bộ đếm tiến lùi.

Nguyên lý hoạt động: Khi tín hiệu I0.2 chuyển từ 0 lên 1 bộ đếm được đặt giá trị là 55.Giá trị đầu ra Q4.0 = 1

Bộ đếm thực hiện đếm tiến tại các sườn lên của tín hiệu tại chân CU khi tín hiệu I0.0chuyển giá trị từ 0 lên 1.

Bộ đếm sẽ đếm lùi tại các sườn lên của tín hiệu tại chân I0.1 khi tín hiệu I0.1 chuyển từ 0 lên 1.

Giá trị của bộ đếm sẽ trở về 0 khi có tín hiệu tại sườn lên của chân R.

FBD	LAD	STL
		A I 0.0 CU C 1 A I 0.1 CD C 1 A I 0.2 L C#55 S C 1 A I 0.3 R C 1 L C 1 T MW 0 LC C 1 T MW 2 A C 1 = Q 4.0

Chương IV kết nối mạng trong PLC

IV.1-Mạng MPI *(Multi-point-Capable-Interface)*

1-Giới thiệu :

Mạng MPI là mạng giao diện nhiều điểm phục vụ cho việc nối máy lập trình với các thiết bị ngoại vi khác.Trong thiết bị điều khiển logic khả trình(PLC),chỉ tồn tại một đường lối duy nhất với máy lập trình.MPI tạo ra khả năng ghép nối với các module khả trình khác như FM. Giao diện MPI được nối bằng các bus ở phía sau như một bus truyền thông.

Ghép nối: Một vài thiết bị có khả năng ghép nối dữ liệu với CPU

Khả năng:Một máy lập trình có khả năng cùng làm việc song song với một panel điều hành,và nối thêm một PLC nữa.Có thể nối đồng thời đồng nhiều điểm trong một CPU.Bốn điểm /node có thể nối trong CPU 314.

Đặc điểm của MPI:Sự linh hoạt của mạng MPI được thể hiện bằng văn bản trên màn hình,bằng các panel điều hành,và bằng máy lập trình của Siemens, mạng MPI cung cấp các khả năng sau đây:

*Lập trình CPU và cổng vào/ra thông minh.

*Chức năng điều hành hệ thống và chức năng thông báo.

*Trao đổi dữ liệu giữa máy lập trình và PLC.

*Trao đổi chương trình giữa CPU và thiết bị lập trình.

Đặc tính: Những đặc tính quan trọng của mạng MPI bao gồm :

*Cổng RS 485 và tốc độ truyền thông 187.5Kbaud.

>Khoảng cách từ 50m cho đến 9100m cần có khuyếch đại trung gian(bộ phục hồi)

>Các thành phần tiêu chuẩn của mạng PROFIBUS(Cable,nối,và bộ phục hồi)

.2-Khai báo mạng MPI.

Để khai báo một mạng MPI ta cần làm các bước sau:

*Ta vào phần mềm STEP7.

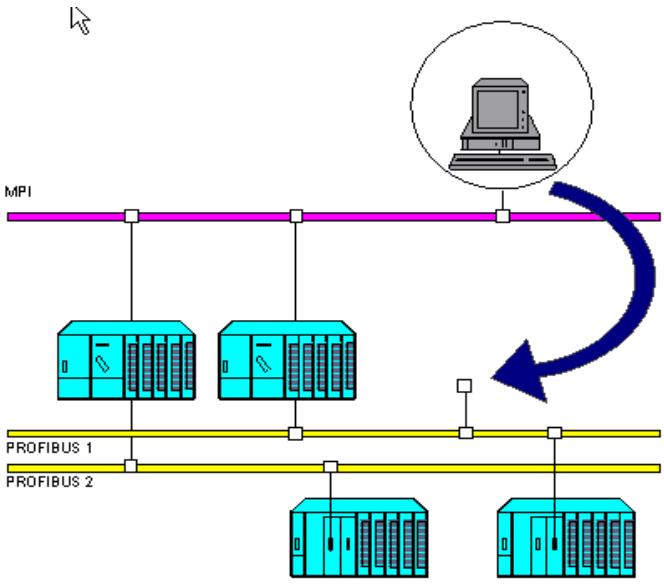
*Mở một Project mới băng cách chọn File→New.

*Sau đó xây dựng cấu hình cứng cho trạm PLC bằng cách ta vào

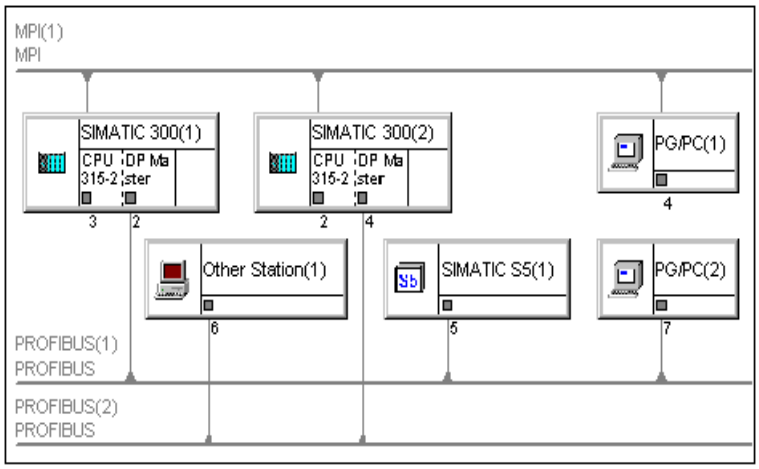
-Insert→Station→Simatic 300 Station

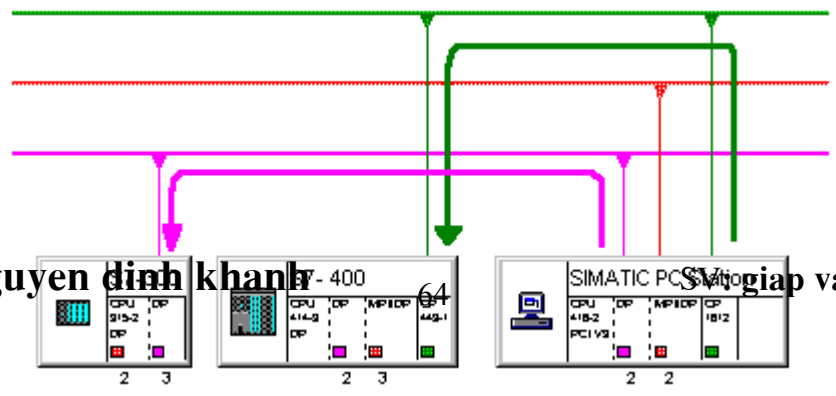
-Insert→Station→Simatic 400 Station

-Sau đó ta khai báo phần cứng cho từng Progam S7-300 và S7-400



The connection points in the network view look like this ("PG/PC(1)" and ("PG/PC(2)"):





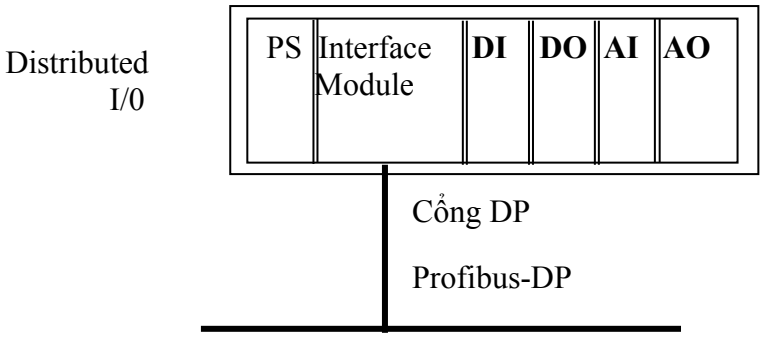
2.3-Mạng vào ra phân tán

Mạng vào ra phân tán là một mạng trong đó các thiết bị vào ra phân tán được ghép nối với nhau qua các Bus như :

+Profilbus-DP

+Profilbus-PA

Thiết bị vào ra phân tán khác với một PLC ở chỗ nó không có bộ xử lý trung tâm CPU.Thay vào đó,nó đợc tích hợp các vi mạch giao diện mạng cũng như phần mền xử lý giao thức. Tuỳ theo cấu trúc của thiết bị vào/ra phân tán là dạng module hay dạng gọn nhẹ mà phần giao diện mạng được thực hiện bằng một module riêng biệt hay không.



Hình trên minh họa cách nối mạng Profilbus-DP cho một thiết bị vào/ra phân tán có cấu trúc module.Về nguyên tắc,phương pháp này không khác so với cách ghép nối các bộ PLC

Chương V thiết kế chế tạo mô hình mô phỏng

I. Khảo sát hệ thống điều khiển hệ thống đèn giao thông tại ngã tư.

1. nguyên tắc điều khiển và hoạt động của hệ thống.

Tại một ngã tư giao nhau bởi hai đường lớn người ta có rất nhiều cách điều khiển bằng 3 pha. Nhưng một trong những cách điều khiển đang được sử dụng rất nhiều trong thực tế, nguyên tắc đó hoạt động như sau:

Mô hình các đèn và nguyên tắc điều khiển hướng đi

Giả sử ta đang tham gia giao thông tại ngã tư nay. Ta sẽ tham gia giao thông như sau: Khi có đèn xanh ở đường chính, ta có quyền đi theo chiều mũi tên:

Sau thời gian đặt cố định hệ thống điều khiển hoạt động. Hệ thông sẽ chuyển từ đèn xanh đến đèn đỏ của đường chính. Hệ thống đèn diều khiển ở trạng thái lần hai, người tham gia giao thông sẽ thực hiện theo chiều mũi tên:

Khi hệ thống chuyển dịch đèn đỏ của hệ thống đền phân nhánh rẽ. Người tham gia giao thông sẽ dừng lại và chuyển sang cho hệ thống đèn đường khác:

Sau vài giây chuyển sang đường khác và hoạt động theo nguyên tắc như đường trên:

2. Phương pháp điều khiển bằng Rơ le trung gian.

a) Đặc điểm

Các loại rơle trung gian tự động điều khiển có đặc điểm chung là:

- Tần số đóng ngắt lớn (1000ữ1200 lần/giờ) nên yêu cầu có tuổi thọ cao có thể từ (1ữ 10).10^6 lần đóng ngắt.

- Có rơle điện xoay chiều và rơle điện một chiều, nhiều khi vì công suất tiêu thụ của rơle (từ 0,1W ữ 2W) nên lõi thép nam châm điện của rơle điện xoay chiều được làm giống như của rơle điện một chiều (lõi hình trụ tròn bằng thép khối, thân và nắp hút bằng thép tấm dẹt) chỉ khác là trên mỗi cực từ lõi thép rơle xoay chiều có

vòng ngắn mạch để chống rung, còn lõi thép rơle một chiều thì có mũ lõi. Cuộn dây xoay chiều có số nhỏ hơn cuộn dây một chiều khi điện áp làm việc bằng nhau.

- Điện áp làm việc của cuộn dây có các loại: 6; 9; 12; 24; 48; 110; 220v.
- Dòng điện tải của tiếp điểm rơle: 1; 3; 5A.
- Số lượng tiếp điểm thường đóng và thường mở từ 2 ữ 8 tiếp điểm.
- Các đầu nối điện vào rơle được thực hiện ở dạng chân cắm hoặc chân hàn đã được tiêu chuẩn hoá. Về kích thước và vị trí, đảm bảo tiếp xúc tốt, rất thuận tiện trong chế tạo, lắp đặt và sửa chữa thay thế.
- Một số loại được đặt trong vỏ kín bằng kim loại (thường là nhôm) ở trong môi trường chân không, đảm bảo rơle làm việc tin cậy và bền vững.
- Phần lớn rơle có vỏ hộp bằng nhựa trong suốt, một số loại có lắp kèm đèn tín hiệu LED các màu đỏ xanh vàng để chỉ thị trạng thái làm việc của rơle.
- Trên thị trường có rất nhiều hãng sản xuất loại rơle trung gian này tuy hình dáng và kích thước cụ thể có khác nhau. Nhưng về nguyên lý cấu tạo và các thông số cơ bản đều như nhau.
- Cấu tạo của rơ le trung gian như hình vẽ :

+ Điện áp vào :12VDC
+ Dòng điện định mức tiếp điểm :10A
+ Số lượng tiếp điểm :có 2 tiếp điểm thường đóng và 2 tiếp điểm thường mở.
+ Đóng cắt Nguồn : 220 VAC,10A

28 VDC, 10A

+ Công suất tiêu thụ : 1,6 W (DC).

3. Cấu hình chung của thiết bị mô phỏng.

Mô hình gồm các thiết bị sau:

+Khung hộp làm sàn

+Hệ thống các cột và bóng đèn

+Hệ thống điều khiển

Trong đó:

Khung sàn: được làm bằng gỗ, có kích thước 85cm . 85cm . 10cm. Đây là nơi làm sàn để dung cột đèn đồng thời đặt các linh kiện và toàn bộ hệ thống đi dây ở trong lòng khung

Hệ thống các cột và bóng:

+Cột được làm bằng Inox và tổng số có 8 cột: 4 cột nhỏ và 4 cột lớn cho đường chính. Cột nhỏ có kích thước: đường kính 15cm, chiều cao 20cm. Cột lớn có chiều cao 28cm.

+Hệ thống bóng: bao gồm 44 bóng phân làm 3 loại, 12 bóng mầu vàng, 16 bóng mầu xanh, 16 bóng mầu đỏ. Là loại bóng điện tử có công suất 4,4 W.

4. Giản đồ thời gian.

Chú thích: theo nguyên tắc hoạt động của hệ thống đèn thì rất khó có thể xác định được giản đồ thời gian của cả hệ thống. Do vậy, chúng ta phải tìm các đèn hoạt

động cùng một khoảng thời gian và nhóm vào một rơ le. Các thời gian này được các bộ thời gian trong PLC điều khiển.

***Mạch điện điều khiển từng trạng thái của hệ thống đèn.**

Hệ thống hoạt động theo đúng 1 chu kỳ thì phải hoạt động qua 12 trạng thái.

Chu trình của hệ thông đèn lại được lặp lại theo chu kỳ ban đầu.

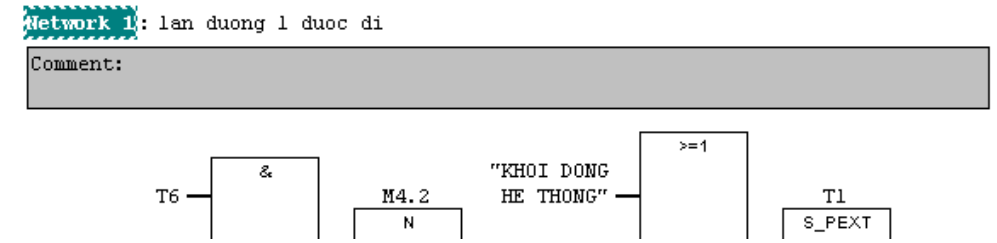
***Mạch điều khiển bằng PLC.**

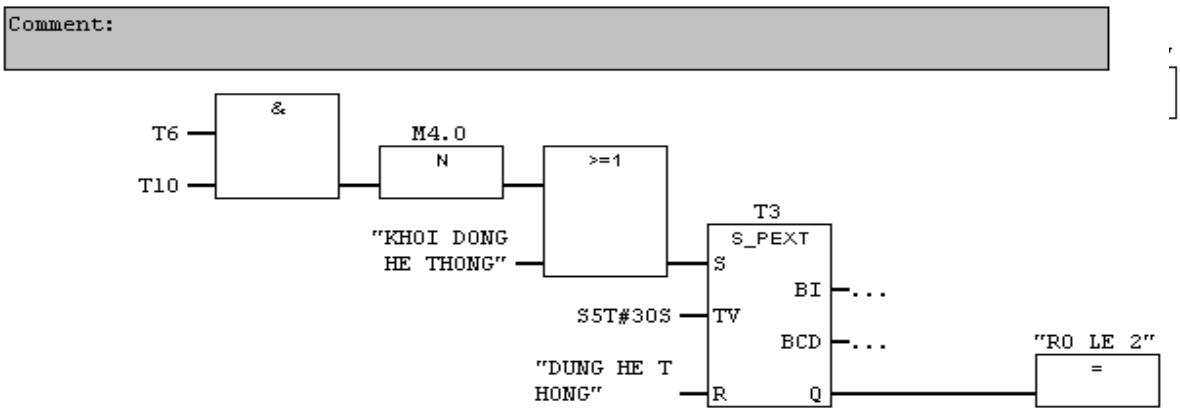
1. Bảng quy định đầu vào ra.

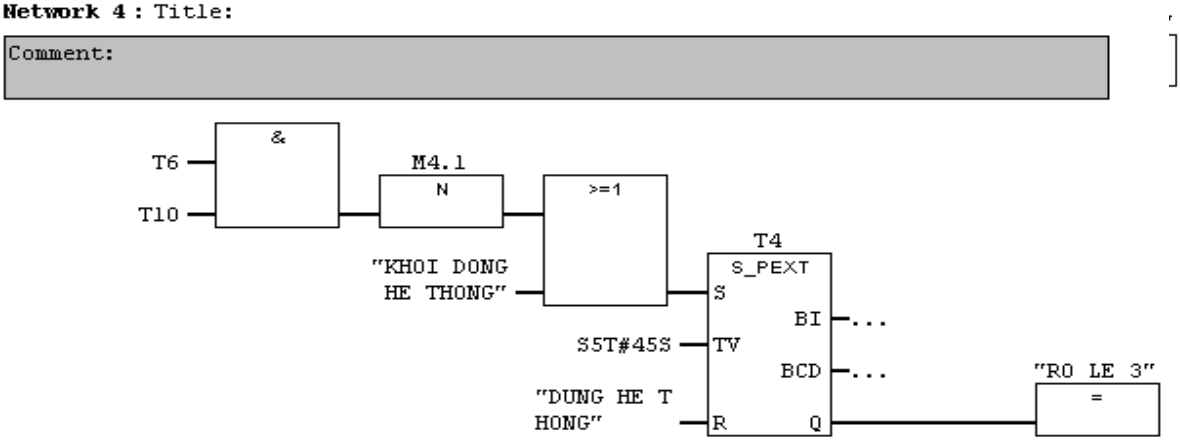
S7 Program(1) (Symbols) -- do an1\SIMATIC 300(1)\CPU 314

	Status	Symbol	Address	Data type	Comment
1					
2		DUNG HE THONG	I 0.1	BOOL	
3		KHOI DONG HE THONG	I 0.0	BOOL	
4		RO LE 0	Q 0.0	BOOL	
5		RO LE 00	Q 4.5	BOOL	
6		RO LE 02	Q 4.6	BOOL	
7		RO LE 03	Q 4.7	BOOL	
8		RO LE 1	Q 0.1	BOOL	
9		RO LE 10	Q 4.2	BOOL	
10		RO LE 11	Q 4.3	BOOL	
11		RO LE 12	Q 4.4	BOOL	
12		RO LE 2	Q 0.2	BOOL	
13		RO LE 3	Q 0.3	BOOL	
14		RO LE 4	Q 0.4	BOOL	
15		RO LE 5	Q 0.5	BOOL	
16		RO LE 6	Q 0.6	BOOL	
17		RO LE 7	Q 0.7	BOOL	
18		RO LE 8	Q 4.0	BOOL	
19		RO LE 9	Q 4.1	BOOL	
20					

***Chương trình của hệ thống.**

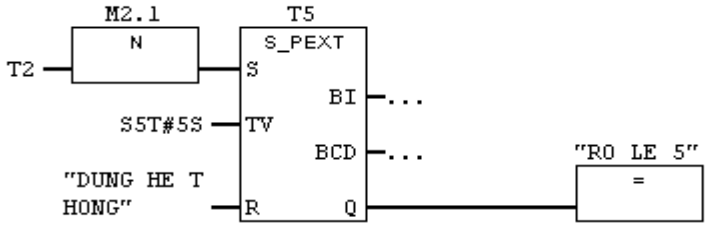






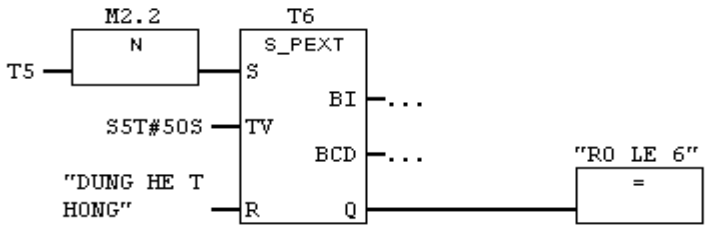
Network 5 : Title:

Comment:



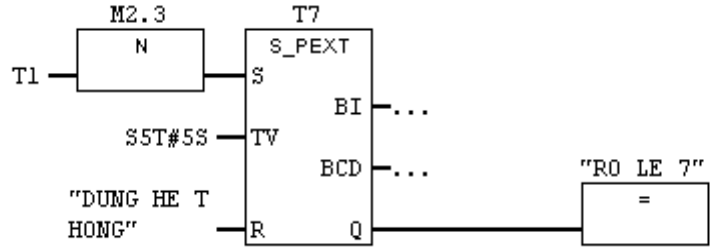
Network 6 : Title:

Comment:



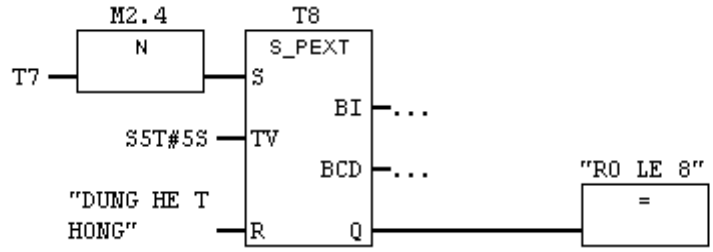
Network 7 : Title:

Comment:



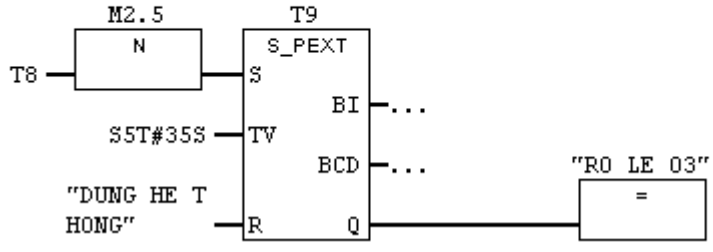
Network 8 : Title:

Comment:



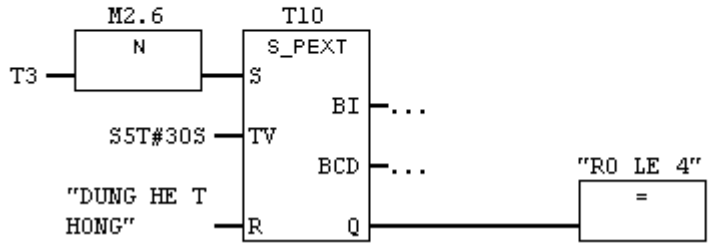
Network 9 : Title:

Comment:



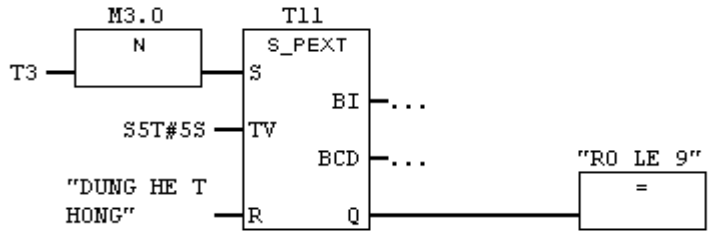
Network 10 : Title:

Comment:



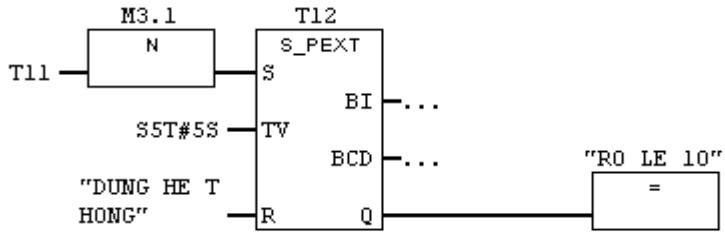
Network 11 : Title:

Comment:



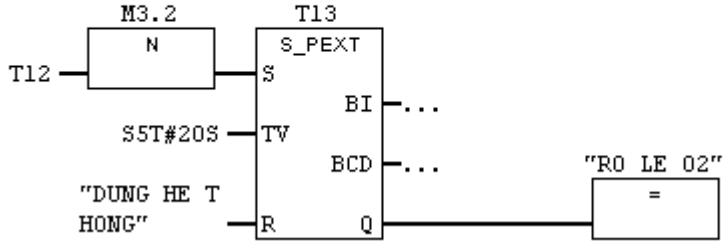
Network 12 : Title:

Comment:



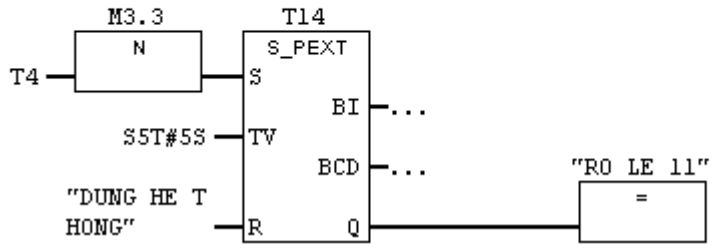
Network 13 : Title:

Comment:



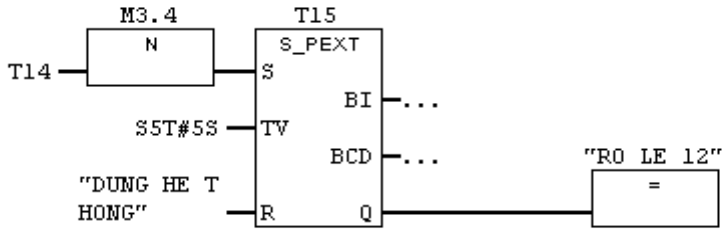
Network 14 : Title:

Comment:



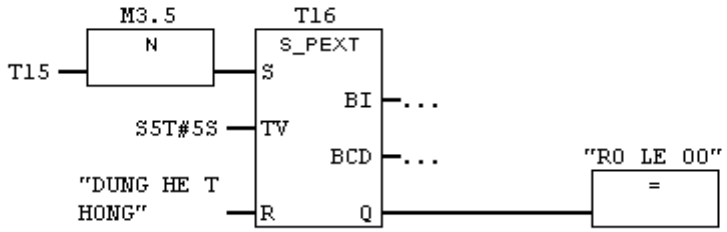
Network 15 : Title:

Comment:



Network 16 : Title:

Comment:



Chương VI: ứng dụng của PLC và hệ thống mô hình.

I. ứng dụng của PLC.

PLC được ứng dụng rất rộng rãi ở nhiều lĩnh vực khác nhau VD:

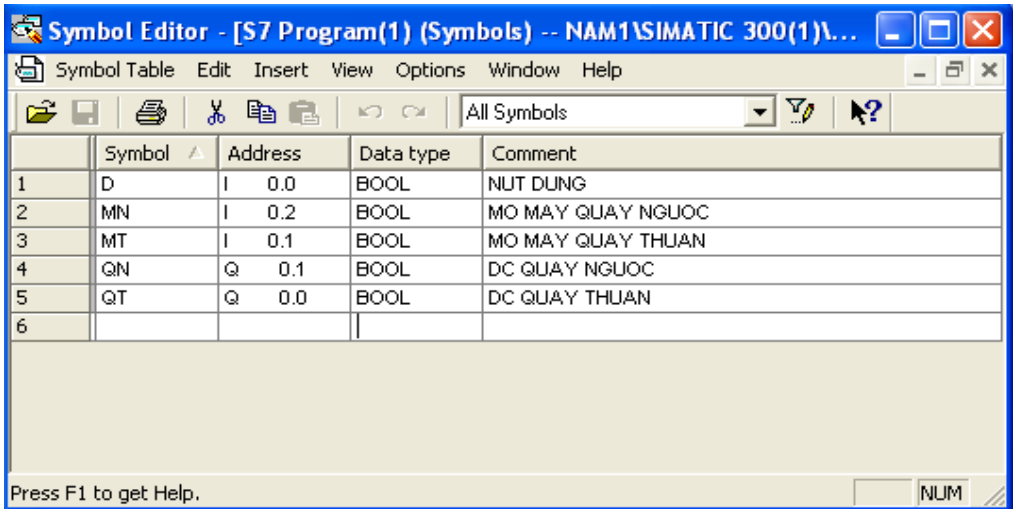
- Điều khiển hệ thống chiếu sáng trong các cửa hàng, siêu thị, nhà hàng,nhà xưởng.
- Điều khiển đóng mở máy các động cơ.
- Điều khiển hệ thống đèn giao thông.
- Hệ thống nâng, vận chuyển.
- Dây truyền đóng gói.
- Các ROBOT lắp ráp sản phẩm.
- Điều khiển bơm.
- Công nghệ sản xuất giấy.
- Công nghệ chế biến thực phẩm.
- Dây truyền chế tạo linh kiện bán dẫn.
- Dây truyền lắp giáp tivi
- Dây truyền sản xuất xi măng.
- Điều khiển bơm.
- Dây truyền xử lý hoá học.
- ứng dụng trong hệ thống bán nước tự động, máy in hoa văn……

Giới thiệu về hệ thống điều khiển mở máy động cơ đảo chiều gián tiếp:

Yêu cầu công nghệ:

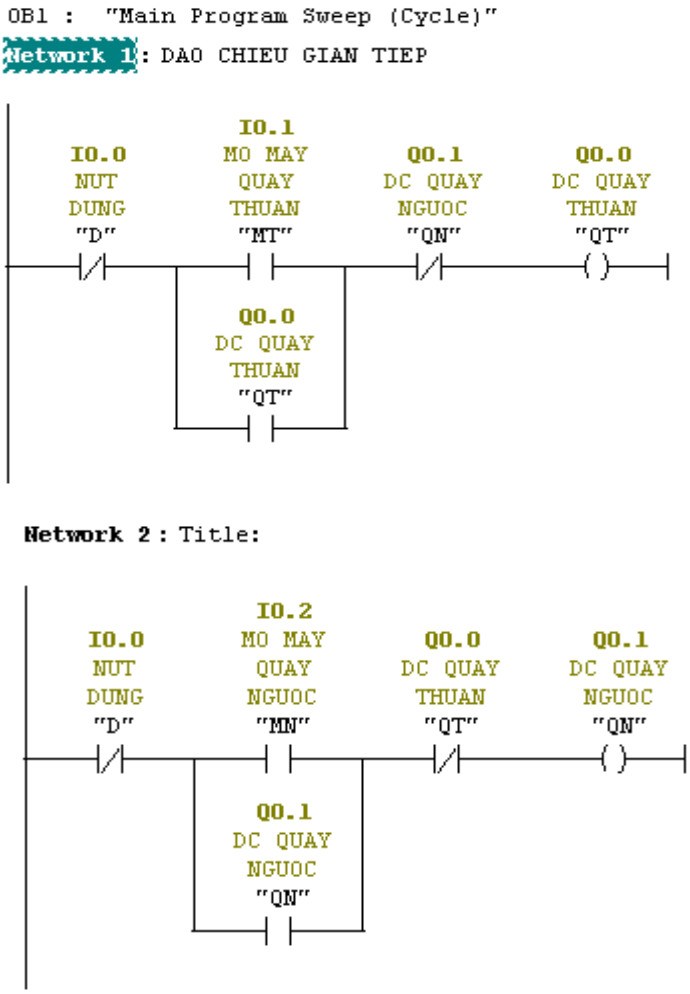
Khi nhấn nút khởi động động cơ quay thuận I0.1cấp điện động cơ quay thuận, khi ấy có nhấn nút khởi động động cơ quay ngược thì động cơ cũng không thể quay ngược được. Như vậy chỉ cho phép khởi động động cơ quay ngược khi đã nhấn nút dừng I0.0.

Bảng quy đinh đầu vào ra:



	Symbol	Address	Data type	Comment
1	D	I 0.0	BOOL	NUT DUNG
2	MN	I 0.2	BOOL	MO MAY QUAY NGUOC
3	MT	I 0.1	BOOL	MO MAY QUAY THUAN
4	QN	Q 0.1	BOOL	DC QUAY NGUOC
5	QT	Q 0.0	BOOL	DC QUAY THUAN
6				

Chương trình lập trình:



Lời KếT

Sau khi nhận đề tài chúng em xác định rõ nhiệm vụ và trách nhiệm của mình với đề tài được giao. Qua việc nghiên cứu đề tài tham khảo tài liệu cộng kiến thức đã học, cùng với sự giúp đỡ, hướng dẫn nhiệt tình của thầy giáo **nguyen dinh khanh** cùng giúp đỡ của bạn bè cùng với sự lỗ lực của bản thân đến nay đề tài của chúng em đã hoàn được thành. Trong quá trình hoàn thành đề tài chúng em còn nhiều sai sót và còn nhiều trường hợp trong thực tế không giải quyết được. Chúng em mong các thầy cô giáo và bạn bè trong ngành đóng góp ý kiến cho chúng em để những đề tài về sau tốt hơn. Với thời gian có hạn nên một số phần chúng em chưa được hoàn chỉnh, chúng mong các bạn ra sau sẽ phát huy khả năng và vốn kiến thức của mình để làm tốt hơn.

Trong quá trình hoàn thành đề tài, chúng em đã trình bày một cách ngắn gọn, dễ hiểu và có hệ thống giúp cho bạn được thuận lợi cho quá trình nghiên cứu và ứng dụng. Chúng em rất mong thầy cô giáo và bạn bè đóng góp ý kiến để chúng em hoàn thành tốt hơn những phần được giao sau.

Chúng em xin chân thành cảm ơn thầy giáo **nguyen dinh khanh** Giảng Viên khoa Điện - Điện Tử đã giúp đỡ chúng em hoàn thành đề tài này!

bac giang, Ngày 15 tháng 6 năm 2010!

Sinh viên thực hiện : giap van thuy

Tài liệu tham khảo

1. ứng dụng PLC SIEMENS và MOELLER trong tự động hoá.
   Nguyễn Tấn Phước. Nhà xuất bản TP Hồ chí minh .
2. Tự Động Hoá với Simentic S7-300 Nguyễn Doãn Phước, Phan Xuân Minh NXB KH-KT 2000.
3. Phần mền Simen – LOGO Version 4.0.
4. Lập trình PLC với S7 - 300 tác giả Nguyễn Xuân Công giảng viên đại học sư phạm kỹ thuật Hưng Yên
5. www.siemens.com/logo
6. Luận văn thạc sĩ về PLC của ĐH Bách Khoa Hà Nội