

**BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**



ISO 9001:2008

**NGHIÊN CỨU ỨNG DỤNG ĐIỀU KHIỂN MỜ CHO
HỆ THỐNG ĐIỀU KHIỂN MỨC NƯỚC VÀ KIỂM
CHỨNG TRÊN MATLAB SIMULINK**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP**

HẢI PHÒNG-2015

**BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**



ISO 9001:2008

**NGHIÊN CỨU ỨNG DỤNG ĐIỀU KHIỂN MỜ CHO
HỆ THỐNG ĐIỀU KHIỂN MỨC NƯỚC VÀ KIỂM
CHỨNG TRÊN MATLAB SIMULINK**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP**

Sinh viên: Trần Hồng Nam

Người hướng dẫn: T.S Nguyễn Trọng Thắng

HẢI PHÒNG-2015

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
ĐỘC LẬP TỰ DO HẠNH PHÚC
-----o0o-----
BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Trần Hồng Nam – mã SV: 1012102005
Lớp : ĐC1401- Ngành Điện Tự Động Công Nghiệp.
Tên đề tài: Nghiên cứu ứng dụng điều khiển mờ cho hệ thống điều khiển mức nước và kiểm chứng trên matlab-Simulink.

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp:.....

.....

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ 1.

Họ và tên : Đỗ Thị Hồng Lý
Học hàm, học vị : Thạc sĩ
Cơ quan công tác : Trường Đại học dân lập Hải Phòng
Nội dung hướng dẫn : Toàn bộ đề án

Người hướng dẫn thứ 2.

Họ và tên :
Học hàm, học vị :
Cơ quan công tác :
Nội dung hướng dẫn :

Đề tài tốt nghiệp được giao ngày.....tháng.....năm 2015.

Yêu cầu phải hoàn thành xong trước ngày.....tháng.....năm 2015.

Đã nhận nhiệm vụ Đ.T.T.N.
Sinh viên

Đã giao nhiệm vụ Đ.T.T.N
Cán bộ hướng dẫn Đ.T.T.N

Trần Hồng Nam

T.S Nguyễn Trọng Thắng

Hải Phòng, ngày.....tháng.....năm 2015

HIỆU TRƯỞNG

GS.TS.NGUYỄN TRẦN HỮU NGHỊ

LỜI MỞ ĐẦU

Cùng với sự phát triển của các ngành kỹ thuật điện tử, công nghệ thông tin, ngành kỹ thuật điện tử, ngành tự động hóa đã và đang đạt được nhiều tiến bộ mới. Tự động hóa không những làm giảm nhẹ sức lao động cho con người mà còn góp phần rất lớn trong việc nâng cao năng suất lao động, cải thiện chất lượng sản phẩm. Chính vì thế tự động hóa ngày càng khẳng định được vị trí cũng như vai trò của mình trong các ngành công nghiệp và đang được phổ biến rộng rãi trong các hệ thống công nghiệp trên thế giới nói chung và Việt Nam nói riêng.

Không chỉ dừng lại ở đó, sự phát triển của tự động hóa còn đem lại nhiều tiện ích phục vụ đời sống hàng ngày cho con người. Một minh chứng rõ nét chính là sự ra đời của chuông báo tự động và nhiều tiện ích hơn, đa năng hơn. Để phục vụ tốt hơn nữa đời sống con người trong thời điểm xã hội ngày càng hiện đại và phát triển hiện nay, vẫn luôn đòi hỏi cải tiến hơn nữa công nghệ cùng những tính năng tiện ích cho chuông tự động báo. Việc ứng dụng thành công các thành tựu của lý thuyết điều khiển tối ưu, công nghệ thông tin, công nghệ máy tính, công nghệ điện tử và các lĩnh vực khoa học kỹ thuật khác trong những năm gần đây đã dẫn đến sự ra đời và phát triển thiết bị điều khiển logic có khả năng lập trình (PLC). Cũng từ đây đã tạo ra một cuộc cách mạng trong lĩnh vực kỹ thuật điều khiển.

Ngày nay ai cũng biết rõ rằng công nghệ PLC đóng vai trò quan trọng trong năng lượng cơ và làm bộ não cho các bộ phận cần tự động hóa và cơ giới hóa. Đó đó điều khiển logic lập trình (PLC) rất cần thiết với các kỹ sư cơ khí và các kỹ sư điện, điện tử, từ đó giúp họ nắm được phạm vi ứng dụng rộng rãi và kiến thức về PLC cũng như cách sử dụng thông thường.

Trong thời gian làm đồ án tốt nghiệp, em được giao nhiệm vụ và nghiên cứu với đề tài: “Ứng dụng”

Đây là một đề tài không hoàn toàn là mới nhưng nó rất phù hợp với thực tế ở các công ty, nhà máy có thời gian làm việc chia theo ca kíp, càng đi sâu nghiên cứu càng thấy nó hấp dẫn và thấy được vai trò của nó trong việc điều khiển tự động.

Xác định được nhiệm vụ của mình em đã cố gắng hết sức, tập trung tìm hiểu Kết quả thu được chưa nhiều do còn hạn chế về kiến thức, thời gian và kinh nghiệm nhưng nó giúp e có thêm kiến thức để sau khi ra trường có nền tảng tiếp cận được với công nghệ mới.

Trong quá trình làm đồ án do trình độ hiểu biết của em có hạn, nên nội dung đồ án không tránh khỏi những sai sót. Vì vậy em rất mong được sự chỉ bảo góp ý của các thầy cô cũng như mọi người quan tâm đến vấn đề này.

Qua đồ án này cho em xin được bày tỏ lời cảm ơn chân thành tới thầy giáo () người đã trực tiếp hướng dẫn tận tình, giúp đỡ chỉ bảo cho em, cùng toàn thể các thầy cô giáo trong khoa và nhà trường đã giúp đỡ và tạo điều kiện thuận lợi cho em để hôm nay e hoàn thành đồ án một cách đầy đủ.

CHƯƠNG 1.

CHƯƠNG TỰ ĐỘNG TRONG CÔNG TY TNHH FEDERAL VIỆT NAM KCN TRÀNG DUỆ, AN DƯƠNG, HP

1.1. CHƯƠNG TỰ ĐỘNG TRONG CÔNG TY

Trước đây tại các nhà máy, phân xưởng để thuận tiện cho việc thông báo giờ nghỉ giữa các ca cho công nhân người ta sử dụng tiếng keng báo hiệu. Tại các nhà máy trước đây quy mô sản xuất nhỏ ít được phân khu số lượng công nhân ít. Vì vậy việc sử dụng keng báo ca làm việc khá thích hợp. Nhưng ngày nay tại các nhà máy có quy mô lớn và được chia theo nhiều khu sản xuất thì việc sử dụng tiếng keng để báo ca không còn hợp lý là do cá nguyên nhân:

- Điện tích của nhà máy lớn(thường hàng nghìn m²)
- Số lượng công nhân rất lớn.
- Cách bố trí dây truyền của nhà máy theo từng khu.

Từ những nguyên do trên mà ta không thể sử dụng keng để báo ca làm việc. Thay vào đó là hệ thống chuông bấm.

Hệ thống chuông điện giải quyết được các vấn đề sau:

- Lắp đặt dễ dàng, hệ thống bao gồm nhiều chuông được bố trí ở nhiều địa điểm cần thiết.
- Việc điều khiển rất đơn giản, chỉ cần một người bảo vệ ngồi trong phòng ấn nút điều khiển.
- Độ tin cậy cao.

Nhưng nhược điểm lớn nhất của hệ thống chuông bấm này chính là con người. phải mất một người thường xuyên phải trực ở đó để bấm chuông báo giờ. Đôi khi người trực ngủ quên hoặc xem nhầm giờ và rất nhiều nguyên nhân khác quan khác ảnh hưởng đến sự sai lệch thời gian cam làm việc khó phân biệt tiếng chuông và đầu ca làm và kết thúc ca làm việc. Đứng trước vấn đề này cần phải thiết kế hệ thống chuông báo tự động tại nhà máy.

Hệ thống chuông tự động có ưu điểm:

- Thuật toán lập trình đơn giản.
- Độ chính xác, độ tin cậy rất cao.
- Không cần có người trực điều khiển, chỉ cần ấn nút khởi động một lần hệ thống sẽ chạy tự động hoàn toàn và liên tục trong nhiều năm liên tiếp.
- Phân biệt rõ tiếng chuông và ca làm việc, nghỉ giữa ca và kết thúc ca.

1.2. PHÂN TÍCH MỤC ĐÍCH, YÊU CẦU ĐỒ ÁN

1.2.1. Mục đích:

- Hệ thống báo chuông tại các thời điểm vào, ra của các ca làm việc.
- Hệ thống có khả năng tính lại giờ.
- Thời gian kéo dài chuông vào ca làm việc và nghỉ giải lao là khác nhau.
- Hệ thống chuông được dùng đi dây điện đồng bộ 220V.

1.2.2. Yêu cầu:

- Hệ thống làm việc ổn định.
- Có khả năng đưa mô hình vào ứng dụng trong thực tế.

1.2.3. Thời gian các ca làm việc.

Trong quá trình đi thăm quan công ty TNHH Federal Việt Nam KCN Trảng Duyệt, An Dương, HP thời gian giữa các ca làm việc như sau:

Có 3 ca làm việc mỗi ngày, mỗi ca làm việc 8 tiếng đồng hồ, giữa một ca công nhân được nghỉ 30 phút. Riêng thời gian của các ca là 5 phút.

Ta chia làm 3 ca:

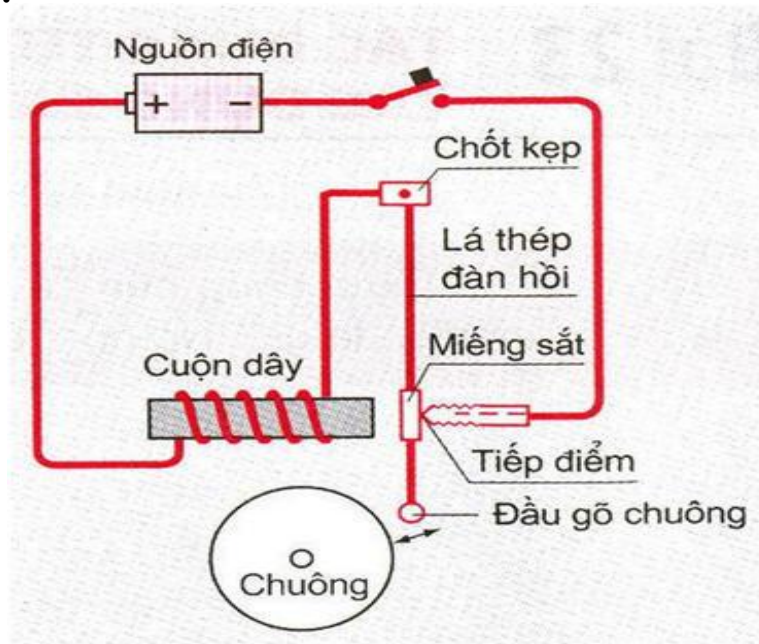
Ca 1 : 6h5 đến 14h

Ca 2 : 14h5 đến 22h

Ca 3 : 22h5 đến 6h sáng hôm sau

1.3. CẤU TẠO VÀ NGUYÊN LÝ HOẠT ĐỘNG CHUÔNG ĐIỆN

1.3.1. Cấu tạo



Chuông điện có cấu tạo gồm các phần chính:

- 1 Cuộn dây (nam châm điện)
- 2 Búa gõ
- 3 Chuông
- 4 Miếng sắt (tác dụng để nam châm điện hút và kéo búa gõ gõ vào chuông)
- 5 Chốt kéo

1.3.2. Nguyên lý

Bộ phận chính trong mọi chuông điện chính là một nam châm điện. Nam châm điện có cấu tạo chính là một cuộn dây điện quấn quanh một lõi kim loại từ tính như sắt hay thép.

Chúng hoạt động trên nguyên lý rất đơn giản như sau: Khi có dòng điện đi qua cuộn dây chúng sẽ tạo ra một từ trường trong lõi kim loại. Cuộn dây sẽ khuếch đại từ trường này và khi đó nam châm điện có thể hút các vật chất bằng sắt thép xung quanh nó giống như một nam châm vĩnh cửu thông thường.

Khi chúng ta nhấn công tắc thì dòng điện 220V sẽ được khép kín. Đầu tiên dòng điện này sẽ đi qua một máy biến áp đơn giản để giảm điện áp xuống khoảng vài vôn để vận hành chuông điện. Tiếp đó dòng điện đã được giảm áp này sẽ đi vào trong hệ thống mạch của chuông điện.

Mạch chuông điện là mạch tự gián đoạn. Mạch chuông đơn giản nhất bao gồm các chi tiết cơ bản (theo sơ đồ) sau: mạch điện mắc nối tiếp với một lá sắt qua một điểm. Một đầu lá sắt gắn với đầu gõ chuông, đầu kia đầu với một lá thép đàn hồi được cố định bởi chốt kẹp. Nam châm điện được gắn vào hai đầu dây dẫn sao cho vị trí của nó có thể hút được lá sắt. Tất cả tạo thành một mạch khép kín.

Khi ta ấn vào nút chuông điện, dòng điện đi vào mạch điện sẽ tạo thành một mạch kín, khi đó nam châm điện hoạt động và từ đó gây ra từ tính, hút lá sắt về phía nó đồng thời gây ra tiếng kêu do một đầu lá sắt gõ vào chuông. Tuy nhiên khi đó, lá sắt sẽ hở ngay tiếp điểm làm mạch điện bị ngắt khiến nam châm điện mất tác dụng và thả lá sắt ra. Lá sắt lại chạm vào tiếp điểm, mạch lại được đóng kín và quy trình này cứ lặp đi lặp lại miễn là chúng ta vẫn ấn vào nút chuông điện. Bằng cách này, các nam châm điện tự tắt mở, gây ra âm thanh không ngừng.

Cũng với nguyên tắc này người ta có thể thiết kế ra nhiều loại chuông điện có âm thanh khác nhau như tiếng chuông rè báo hiệu giờ học, tiếng còi cứu hỏa hay tiếng “kính coong” quen thuộc trong gia đình.

CHƯƠNG 2. TỔNG QUAN VỀ PLC S7-1200

2.1. TỔNG QUÁT VỀ PLC S7 - 1200

PLC viết tắt của Programmable Logic Controller là thiết bị điều khiển lập trình cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua một ngôn ngữ lập trình, người sử dụng có thể lập trình để thực hiện một loạt trình tự các sự kiện. Các sự kiện này được kích hoạt bởi các tác nhân kích thích tác động vào plc hoặc qua các hoạt động có trễ như thời gian định kỳ hay thời gian được đếm. Một khi sự kiện được kích hoạt sự thật, nó bật ON hay OFF các thiết bị bên ngoài gọi là thiết bị vật lý. Một bộ điều khiển lập trình sẽ liên tục lặp trong chương trình do người sử dụng lập ra chờ tín hiệu ở ngõ vào và xuất tín hiệu ở ngõ ra tại các thời điểm đã lập trình.

Để khắc phục những nhược điểm của bộ điều khiển dung dây nối, người ta đã chế tạo bộ điều khiển plc nhằm thỏa mãn các yêu cầu sau:

- Lập trình dễ dàng, ngôn ngữ lập trình dễ học
- Gọn nhẹ, dễ bảo quản, dễ sửa chữa
- Dung lượng bộ nhớ lớn để có thể chứa được những chương trình phức tạp
- Hoàn toàn tin cậy trong môi trường công nghiệp
- Giao tiếp được với các thiết bị thông minh khác như máy tính, nối mạng, các module mở rộng

Các thiết kế đầu tiên là nhằm thay các phần cứng Relay dây nối và các logic thời gian. Tuy nhiên bên cạnh đó việc đòi hỏi tăng cường dung lượng bộ nhớ và tính dễ dàng cho plc mà vẫn đảm bảo tốc độ xử lý cũng như giá cả...

Chính điều này đã tạo ra sự quan tâm sâu sắc đến việc sử dụng PLC trong công nghiệp, các tập lệnh nhanh chóng đi từ các lệnh logic đơn giản

đến lệnh đếm, định thời, thanh ghi dịch... Sự phát triển các máy tính dẫn đến các bộ PLC có dung lượng lớn, số lượng I/O nhiều hơn.

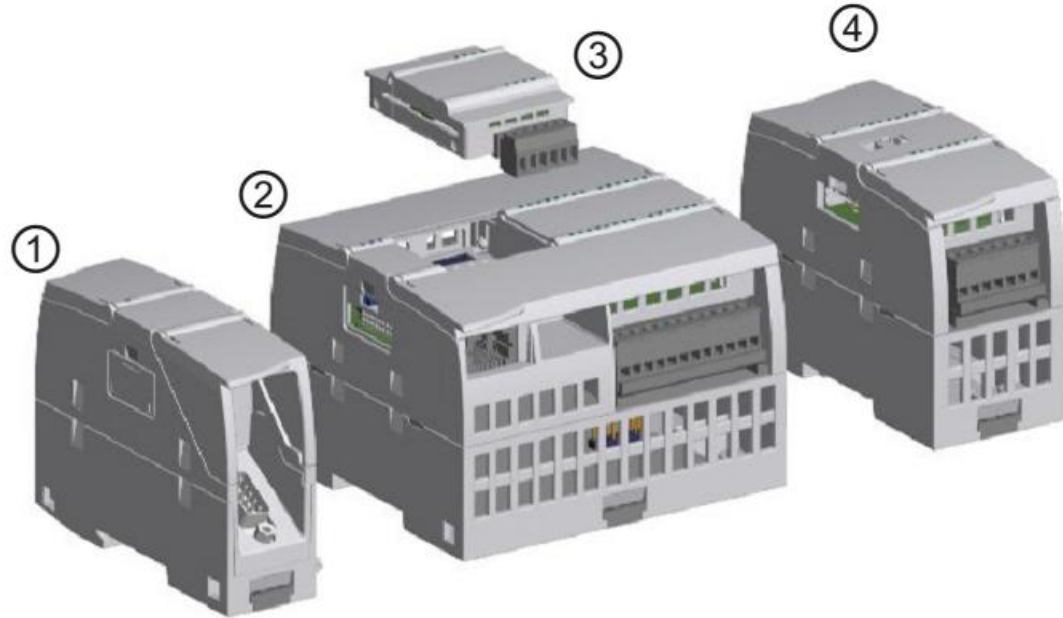
Trong PLC phần cứng CPU và chương trình là đơn vị cơ bản cho quá trình điều khiển và xử lý hệ thống, chức năng mà bộ điều khiển cần thực hiện sẽ được xác định bằng một chương trình. Chương trình này sẽ nạp sẵn vào bộ nhớ của PLC, PLC sẽ thực hiện việc điều khiển dựa vào chương trình này. Như vậy, nếu muốn thay đổi hay mở rộng chức năng của quy trình công nghệ ta chỉ cần thay đổi chương trình bên trong bộ nhớ PLC. Việc thay đổi hay mở rộng chức năng sẽ được thực hiện một cách dễ dàng mà không cần sự can thiệp vật lý nào so với các bộ dây nối hay Relay.

2.2. NHỮNG ĐIỂM NỔI BẬT CỦA SIMATIC S7 – 1200

2.2.1. Thiết kế dạng module.

- Tích hợp cổng truyền thông Profinet (Ethernet) tạo sự dễ dàng trong kết nối.
- Simatic S7 – 1200 với Simatic HMI Basic được lập trình chung trên một nền phần mềm là TIA Portal V10.5 hoặc version cao hơn. Các thao tác lập trình thực hiện theo cách kéo – thả, do đó tạo sự dễ dàng cho người sử dụng, lập trình nhanh chóng, đơn giản, chính xác trong sự truyền thông kết nối theo tags.
- Tích hợp sẵn các đầu vào ra, cùng với các board tín hiệu, khi cần mở rộng ứng dụng với số lượng đầu vào ra ít sẽ tiết kiệm được chi phí, không gian và phần cứng.
- Dễ dàng cho người sử dụng sản phẩm trong việc mua gói thiết bị.

2.2.2. Hình dạng bên ngoài



① Communication module (CM)

② CPU

③ Signal board (SB)

④ Signal module (SM)

Hình 2.1. hình dạng bên ngoài của PLC S7 – 1200 và các module mở rộng

2.2.3. Phân loại

Việc phân loại S7-1200 dựa vào loại CPU mà nó trang bị:

Các loại PLC thông dụng: CPU 1211C, CPU 1212C, CPU 1214C

Thông thường S7-200 được phân ra làm 2 loại chính:

- Loại cấp điện 220VAC:

Ngõ vào: Kích hoạt mức 1 ở cấp điện áp +24VDC(từ 15VDC – 30VDC).

Ngõ ra: Relay.

Ưu điểm của loại này là dùng ngõ ra Relay. Do đó có thể sử dụng ngõ ra ở nhiều cấp điện áp khác nhau(có thể sử dụng ngõ ra 0V, 24V, 220V...)

Tuy nhiên, nhược điểm của nó là do ngõ ra Relay nên thời gian đáp ứng không nhanh cho ứng dụng biến điệu độ rộng xung, hoặc Output

tốc độ cao...

- Loại cấp điện áp 24VDC:

Ngõ vào: Kích hoạt mức 1 ở cấp điện áp +24VDC(từ 15VDC –30VDC).

Ngõ ra: transistor

Ưu điểm của loại này là dùng ngõ ra transistor. Do đó có thể sử dụng ngõ ra này để biến điệu độ rộng xung, Output tốc độ cao...

Tuy nhiên, nhược điểm của loại này là do ngõ ra transistor nên chỉ có thể sử dụng một cấp điện áp duy nhất là 24VDC, do vậy sẽ gặp rắc rối trong những ứng dụng có cấp điện áp khác nhau. Trong trường hợp này, phải thông qua một Relay 24VDC đệm.

Hình 2.2. Các đặc điểm cơ bản của PLC S7- 1200

Đặc trưng	CPU 1211C	CPU 1212C	CPU 1214C
Kích thước (mm)	90*100*75		110*100*75
Bộ nhớ người dùng <ul style="list-style-type: none"> • Bộ nhớ làm việc • Bộ nhớ tải • Bộ nhớ sự kiện 	<ul style="list-style-type: none"> • 25 Kbytes • 1 Mbytes • 2 Kbytes 		<ul style="list-style-type: none"> • 50 Kbytes • 2 Mbytes • 2 Kbytes
Phân vùng I/O <ul style="list-style-type: none"> • Digital I/O • Analog I 	<ul style="list-style-type: none"> • 6 inputs/ 4 outputs • 2 inputs 	<ul style="list-style-type: none"> • 8 inputs/ 6 outputs • 2 input 	<ul style="list-style-type: none"> • 14 inputs/ 10 outputs • 2 inputs
Tốc độ xử lý ảnh	1024 bytes (inputs) and 1024 bytes (outputs)		
Module mở rộng	0	2	8
Mạch tín hiệu	1		
Module giao tiếp	3		
Bộ đếm tốc độ cao <ul style="list-style-type: none"> • Trạng thái đơn • Trạng thái đôi 	3	4	6
	<ul style="list-style-type: none"> • 3-100 KHz • 3-80 KHz 	<ul style="list-style-type: none"> • 3-100 KHz • 1-30 KHz • 3-80 KHz • 1-20 KHz 	<ul style="list-style-type: none"> • 3-100 Khz • 3-30 KHz • 3-80 KHz • 3-20 KHz
Mạch ngõ ra	2		
Thẻ nhớ	Thẻ Simatic tùy chọn		
Thời gian lưu chữ khi mất điện	240h		

PROFINET	1 cổng giao tiếp E thernet
Tốc độ thực thi phép toán số thực	18 us
Tốc độ thi hành	0.1 us

2.2.4. Phạm vi ứng dụng của Simatic S7 1200

- S7 – 1200 bao gồm các họ CPU 1211C, 1212C, 1214C. mỗi loại có các chức năng khác nhau thích hợp cho từng loại ứng dụng.
- Các kiểu cấp nguồn và đầu vào ra có thể là DC/DC/DC/DC hay DC/DC/Rly
- Đều có khe cắm thẻ nhớ, dung khi mở rộng bộ nhớ cho CPU, copy chương trình ứng dụng hay các cập nhật firmware
- Chuẩn đoán lỗi online/offline
- Một đồng hồ thời gian thực cho các ứng dụng thời gian thực

2.2.5. Modules mở rộng tín hiệu vào/ra

Các module mở rộng tín hiệu vào/ra được gắn trực tiếp vào phía bên phải của CPU. Với dải rộng các loại module tín hiệu vào/ra số và analog, giúp linh hoạt trong sử dụng S7 1200. tính đa dạng của các module tín hiệu vào/ra sẽ được tiếp tục phát triển.

2.2.6. Module Analog

SM tín hiệu module cho các đầu vào ra Analog (cho CPU 1212C tối đa của 2 SM có thể sử dụng, cho 1214C là 8).

2.2.7. Module truyền thông

Giao tiếp với RS 232/RS 485

2.2.8. Thẻ nhớ

Simatic thẻ nhớ 2MB hoặc 24MB cho các chương trình dữ liệu và thay thế CPU đơn giản để bảo trì

2.2.9. Module nguồn

Sử dụng module nguồn PM 1207 có các thông số

- Input : 120/230V AC, 50/60Hz, 1.2/0.7A
- Output : 24V DC/2.5A.

2.2.10. Switch

Module CSM 1277 có 4 cổng cắm RJ45, tốc độ 10/100Mb/s

2.3. CẤU TRÚC VÀ NGUYÊN LÝ HOẠT ĐỘNG

2.3.1. Cấu trúc

- Tất cả PLC đều có thành phần chính là bộ nhớ chương trình RAM bên trong, một bộ vi xử lý có cổng giao tiếp dung cho việc ghép nối vs PLC, các module I/O.
- Bên cạnh đó, một số PLC hoàn chỉnh còn đi kèm theo một đơn vị lập trình bằng tay hay bằng máy tính. Hầu hết cá đơn vị lập trình đơn giản đều có đủ RAM để chứa đựng chương trình dưới dạng hoàn thiện hay bổ sung. Nếu đơn vị lập trình là đơn vị xách tay, RAM thường là loại CMOS có pin dự phòng, chỉ khi nào chương trình đã được kiểm tra và sẵn sàng sử dụng thì nó mới chuyển sang bộ nhớ PLC. Đối với các PLC lớn thường lập trình trên máy tính nhằm hỗ trợ cho việc đọc, viết và kiểm tra chương trình. Các đơn vị lập trình nối với PLC qua cổng RS 232, RS422,RS458...

2.3.2. Nguyên lý hoạt động của PLC

CPU điều khiển các hoạt động bên trong PLC. Bộ xử lý sẽ đọc và kiểm tra chương trình được chứa trong bộ nhớ, sau đó thực hiện từng lệnh trong chương trình, sẽ đóng ngắt các đầu ra. Các trạng thái ngõ ra ấy được phát tới các thiết bị liên kết để thực thi điều đó phụ thuộc vào chương trình điều khiển được giữ trong bộ nhớ

Hệ thống bus là tuyến dung để truyền tín hiệu, hệ thống gồm nhiều đường tín hiệu song song :

- Address bus: bus địa chỉ dùng để truyền địa chỉ tới các module khác nhau
- Data bus: bus dùng để truyền dữ liệu
- Control bus: bus điều khiển dùng để truyền các tín hiệu định thời và điều khiển đồng bộ các hoạt động trong PLC.

Hệ thống Bus sẽ làm nhiệm vụ trao đổi thông tin giữa CPU, bộ nhớ và I/O. Bên cạnh đó CPU được cung cấp 1 xungclock có tần số từ 1.8 Mhz. Xung này quyết định tốc độ hoạt động của PLC và cung cấp các yếu tố về định thời, đồng hồ của hệ thống.

2.3.3. Đèn tín hiệu PLC

Có 3 loại đèn hoạt động:

- Run/stop : đèn xanh/ đèn vàng báo hiệu PLC đang hoạt động/ dừng hoạt động.
- Error : đèn báo lỗi.
- Maint : đèn báo khi ta buộc địa chỉ vào nó lên 1

Có 2 loại đèn hiển thị :

- Ix.x chỉ trạng thái logic ngõ vào
- Qx.x chỉ trạng thái logic ngõ ra

2.3.4. Bộ nhớ PLC

PLC thường yêu cầu bộ nhớ trong các trường hợp : làm bộ định thời các kênh trạng thái I/O. Làm bộ đệm trạng thái các chắc năng trong PLC như định thời, đếm, gọi các Relay.

Mỗi lệnh của chương trình có một vị trí riêng trong bộ nhớ, tất cả các vị trí trong bộ nhớ đều được đánh số, những số này chính là địa chỉ trong bộ nhớ. Địa chỉ của từng ô nhớ sẽ được trỏ đến bởi một bộ đếm địa chỉ nằm bên trong bộ vi xử lý. Bộ vi xử lý sẽ có giá trị trong bộ đếm này thêm một trước

khi xử lý lệnh tiếp theo. Với một địa chỉ mới, nội dung của ô nhớ tương ứng sẽ xuất hiện ở đầu ra, quá trình này gọi là quá trình đọc.

Bộ nhớ bên trong của PLC được tạo bởi vi mạch bán dẫn, mỗi vi mạch này có khả năng chứa 2000-16000 dòng lệnh tùy theo loại vi mạch trong PLC các bộ nhớ như RAM và EPROM đều được sử dụng.

Ram có thể nạp chương trình, thay đổi xóa bỏ nội dung bất kỳ lúc nào, nội dung RAM sẽ bị mất nếu nguồn điện nuôi bị mất. Để tránh tình trạng này các PLC đều được trang bị pin khô có khả năng cung cấp năng lượng dự trữ cho RAM từ vài tháng cho đến vài năm. Xu hướng hiện nay sử dụng CMOSRAM do khả năng tiêu thụ thấp và tuổi thọ cao.

EPROM là bộ nhớ mà người sử dụng bình thường có thể đọc chứ không ghi được nội dung vào, nội dung của EPROM không bị mất khi mất nguồn, nó được già sẵn trong máy, đã được nhà sản xuất nạp và chứa sẵn hệ điều hành. Nếu người dùng không muốn sử dụng bộ nhớ thì chỉ dung EPROM gắn bên trong PLC.

EEPROM liên kết với những truy xuất linh động của RAM và có tính ổn định. Nội dung của nó có thể xóa và lập chương trình bằng điện tùy nhiên số lần là có giới hạn.

2.4. GIỚI THIỆU PHẦN MỀM TIA PORTAL V12

2.4.1 Làm việc với một trạm PLC

2.4.1.1. Giới thiệu SIMATIC STEP 7 Basic – tích hợp lập trình PLC và HMI

Step 7 basic hệ thống kỹ thuật đồng bộ đảm bảo hoạt động liên tục hoàn hảo.

Một hệ thống kỹ thuật mới

Thông minh và trực quan cấu hình phần cứng kỹ thuật và cấu hình mạng, lập trình, chuẩn đoán và nhiều hơn nữa.

Lợi ích với người dùng:

- Trực quan: dễ dàng để tìm hiểu và dễ dàng để hoạt động
- Hiệu quả: tốc độ về kỹ thuật
- Chức năng bảo vệ: Kiến trúc phần mềm tạo thành cơ sở ổn định cho sự đổi mới trong tương lai.

2.4.1.2. kết nối qua giao thức TCP/IP

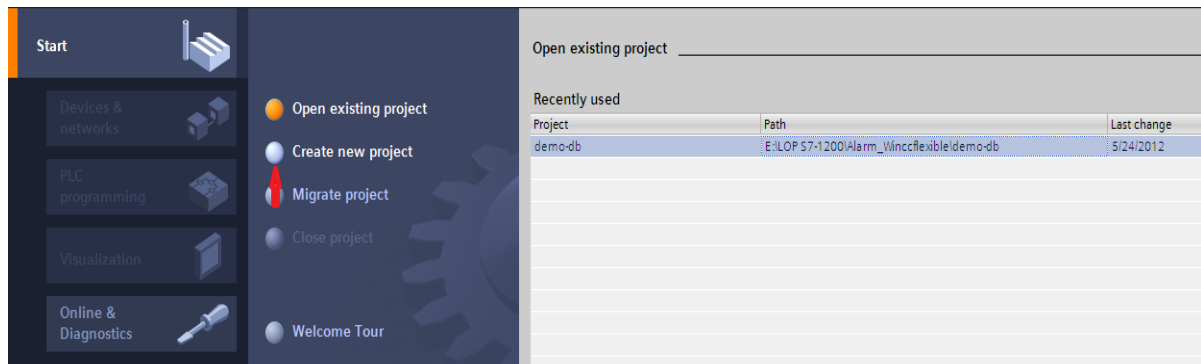
- Để lập trình SIMATIC S7 – 1200 từ PC hay Laptop cần một kết nối TCP/IP
- Để PC và SIMATIC S7 – 1200 có thể giao tiếp với nhau, điều quan trọng là các địa chỉ IP của cả hai thiết bị phải phù hợp với nhau

2.4.2. Cách tạo một Project

Bước 1: từ màn hình desktop nhấp đúp chọn biểu tượng Tia Portal V11



Bước 2: Click chuột vào Create new project để tạo dự án.



Bước 3: Nhập tên dự án vào Project name sau đó nhấn create

Create new project _____

Project name:

Path:

Author:

Comment:

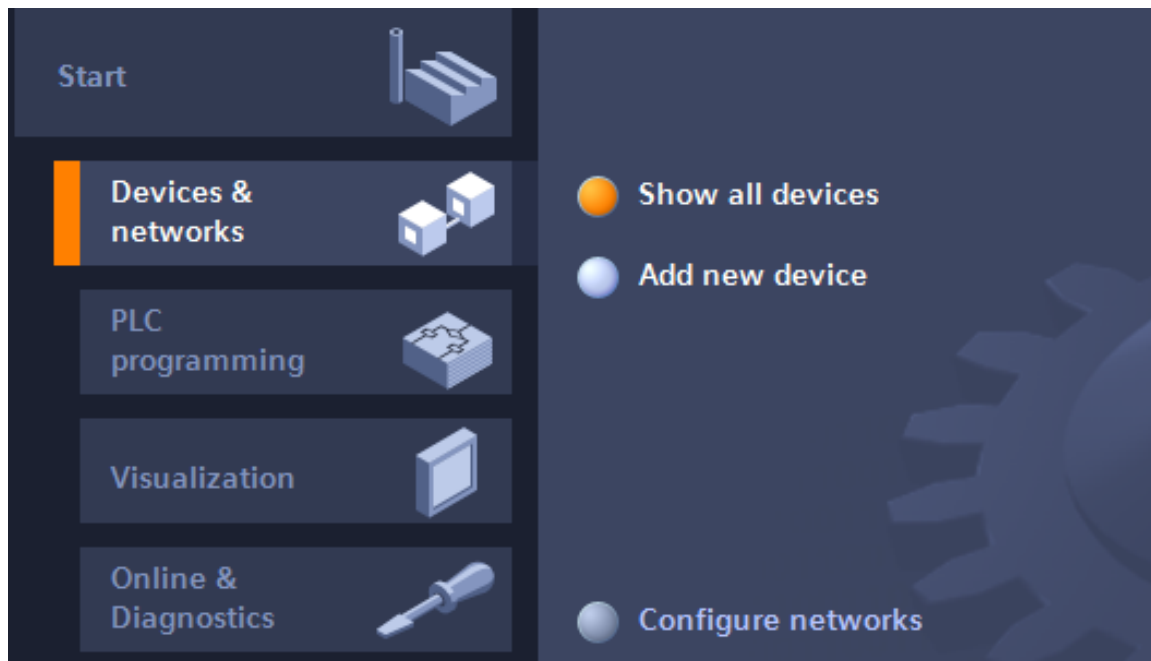
Bước 4: Chọn configure a device

First steps _____

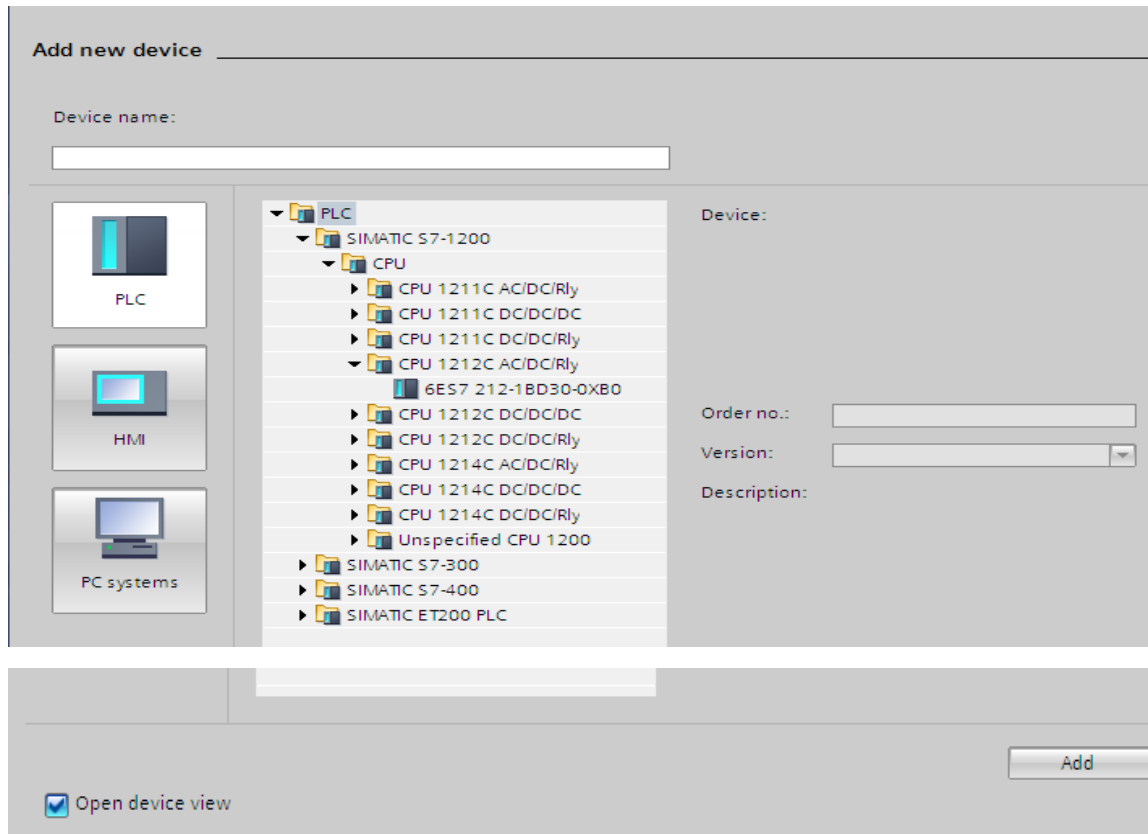
Project: "project3" was opened successfully. Please select the next step:

- Start
- Devices & networks Configure a device
- PLC programming Write PLC program
- Visualization Configure an HMI screen

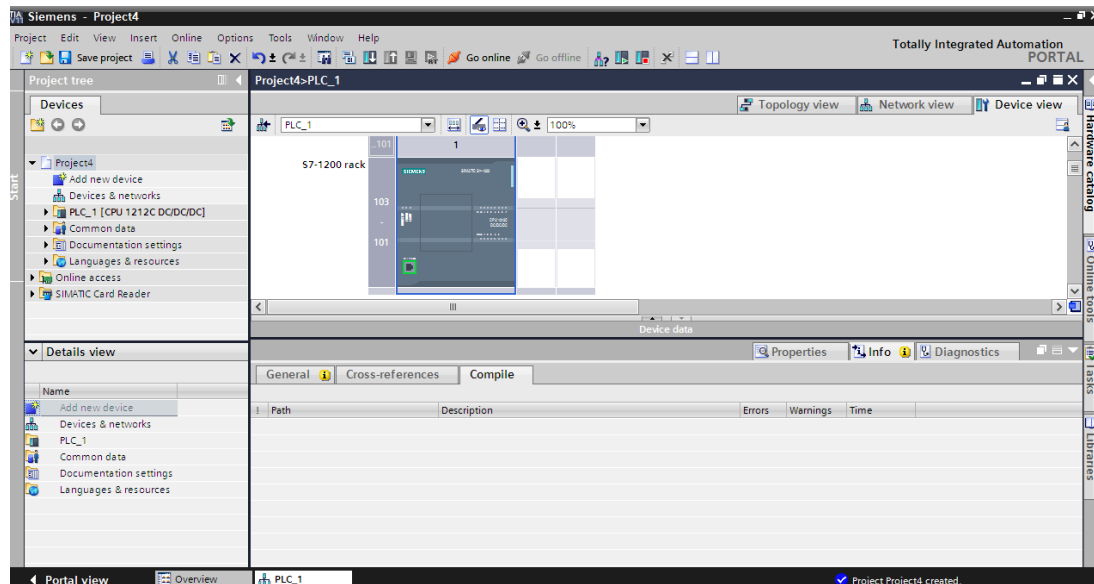
Bước 5: Chọn add new device



Bước 6: Chọn loại CPU PLC sau đó chọn add



Bước 7: Project mới được hiện ra



2.4.3. TAG của PLC / TAG local

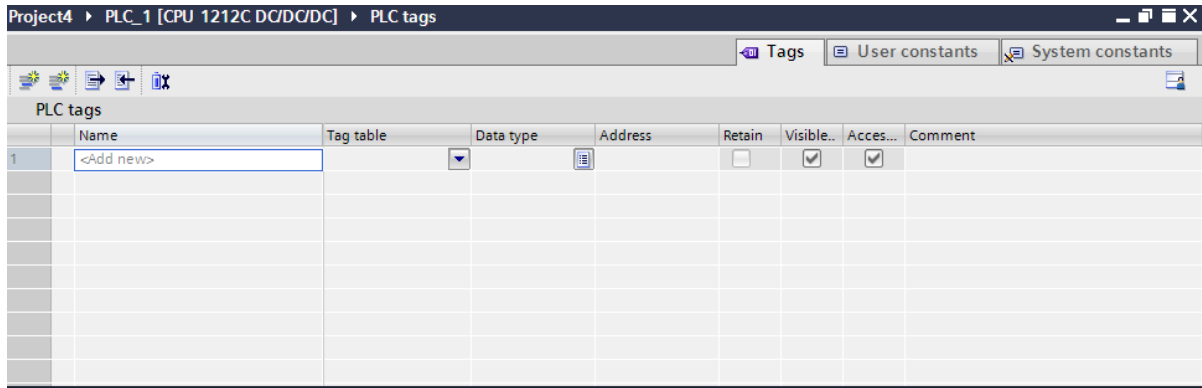
2.4.3.1. Tag của PLC

- Phạm vi ứng dụng: giá trị Tag có thể được sử dụng mọi khối chức năng trong PLC
- Ứng dụng: binary I/O, Bits of memory
- Định nghĩa vùng: Bảng tag của PLC
- Miêu tả: Tag PLC được đại diện bằng dấu ngoặc kép

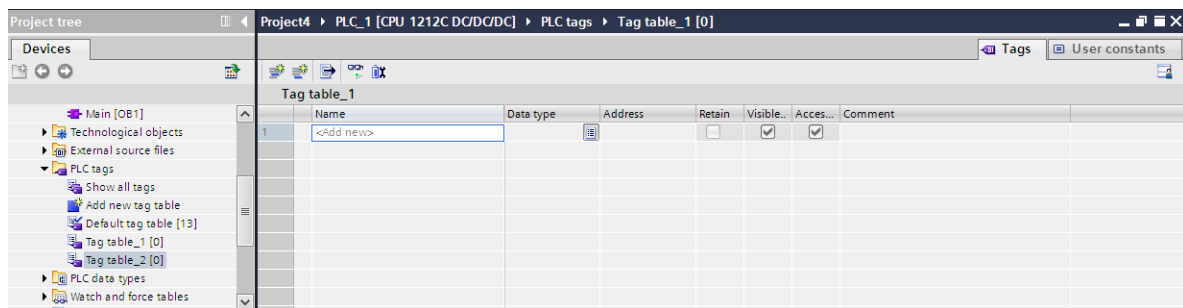
2.4.3.2. Tag Local

- Phạm vi ứng dụng: giá trị chỉ được ứng dụng trong khối được khai báo, mô tả tương tự có thể được sử dụng trong các khối khác nhau cho các mục đích khác nhau.
- Ứng dụng: tham số của khối, dữ liệu static của khối, dữ liệu tạm thời
- Định nghĩa vùng: khối giao diện
- Miêu tả: Tag được đại diện bằng dấu #

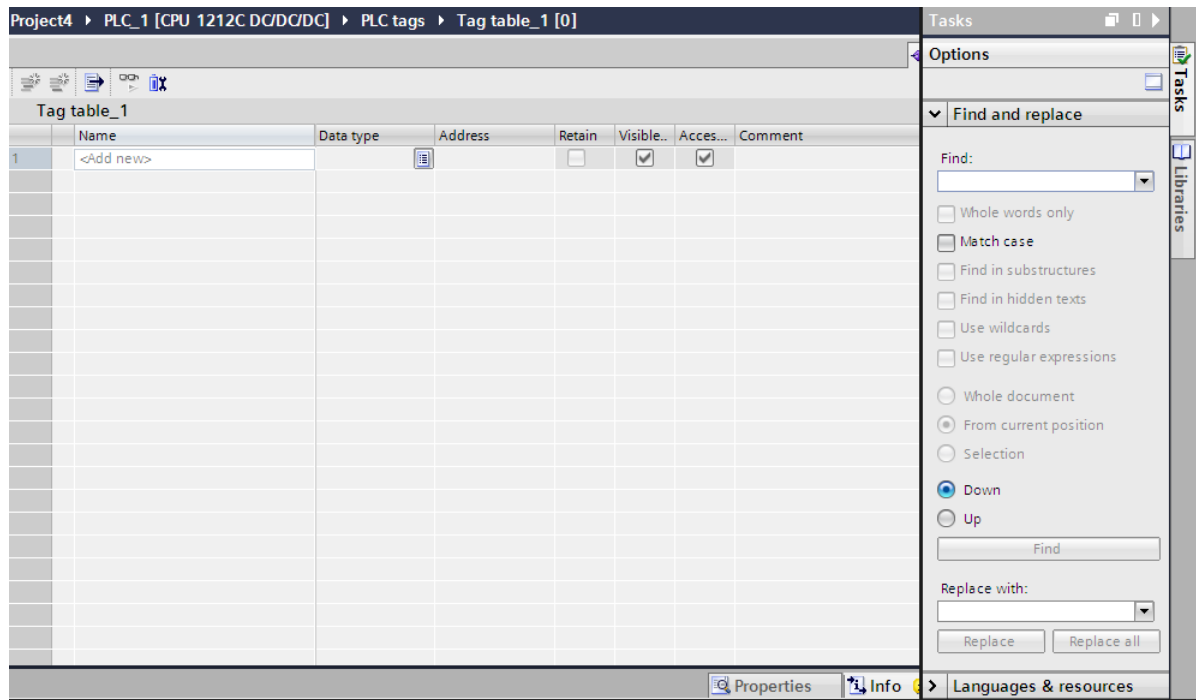
Sử dụng Tag trong hoạt động



- Layout: bảng tag PLC chứa các định nghĩa của các Tag và các hằng số có giá trị trong CPU. Một bảng tag của PLC được tự động tạo ra cho mỗi CPU được sử dụng trong project.
- Colum: mô tả biểu tượng có thể nhấp vào để di chuyển vào hệ thống hoặc có thể kéo thả như một lệnh chương trình
- Name: chỉ được khai báo và sử dụng một lần trên CPU
- Data type: kiểu dữ liệu chỉ định cho các tag
- Address: địa chỉ của tag
- Retain: khai báo của tag sẽ được lưu trữ lại
- Comment : comment miêu tả của tag
- Nhóm tag: tạo nhóm tag bằng cách chọn add new tag table



- Tìm và thay thế tag PLC



- Ngoài ra còn có một số chức năng sau:
- Lỗi tag
- Giám sát tag của plc
- Hiện / ẩn biểu tượng
- Đổi tên tag: Rename tag
- Đổi tên địa chỉ tag: Rewire tag
- Copy tag từ thư viện Global

2.4.4. Làm việc với một trạm PLC

2.4.4.1. Quy định địa chỉ IP cho module CPU

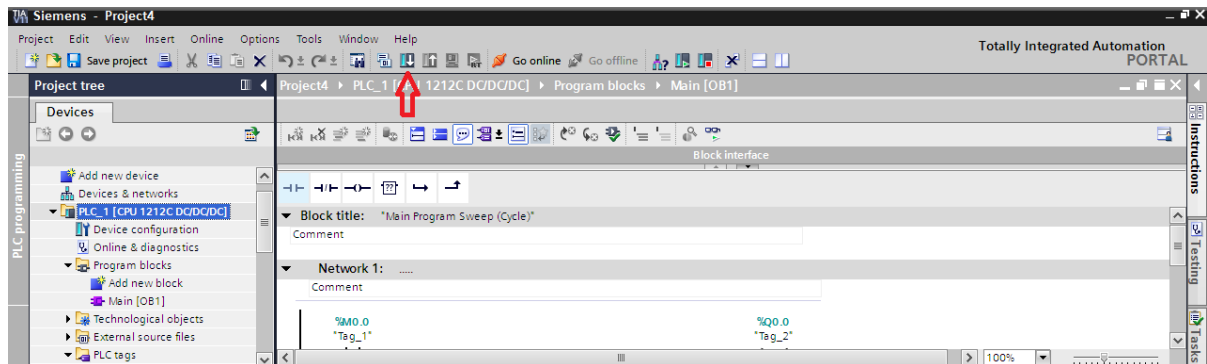
IP TOOL có thể thay đổi IP address của PLC S7-1200 bằng 1 trong 2 cách. Phương pháp thích hợp được tự động xác định bởi trạng thái của địa chỉ IP đó:

Gán một địa chỉ IP ban đầu: Nếu PLC S7-1200 không có địa chỉ IP, IP TOOL sử dụng các chức năng thiết lập chính để cấp phát một địa chỉ IP ban đầu cho PLC S7-1200.

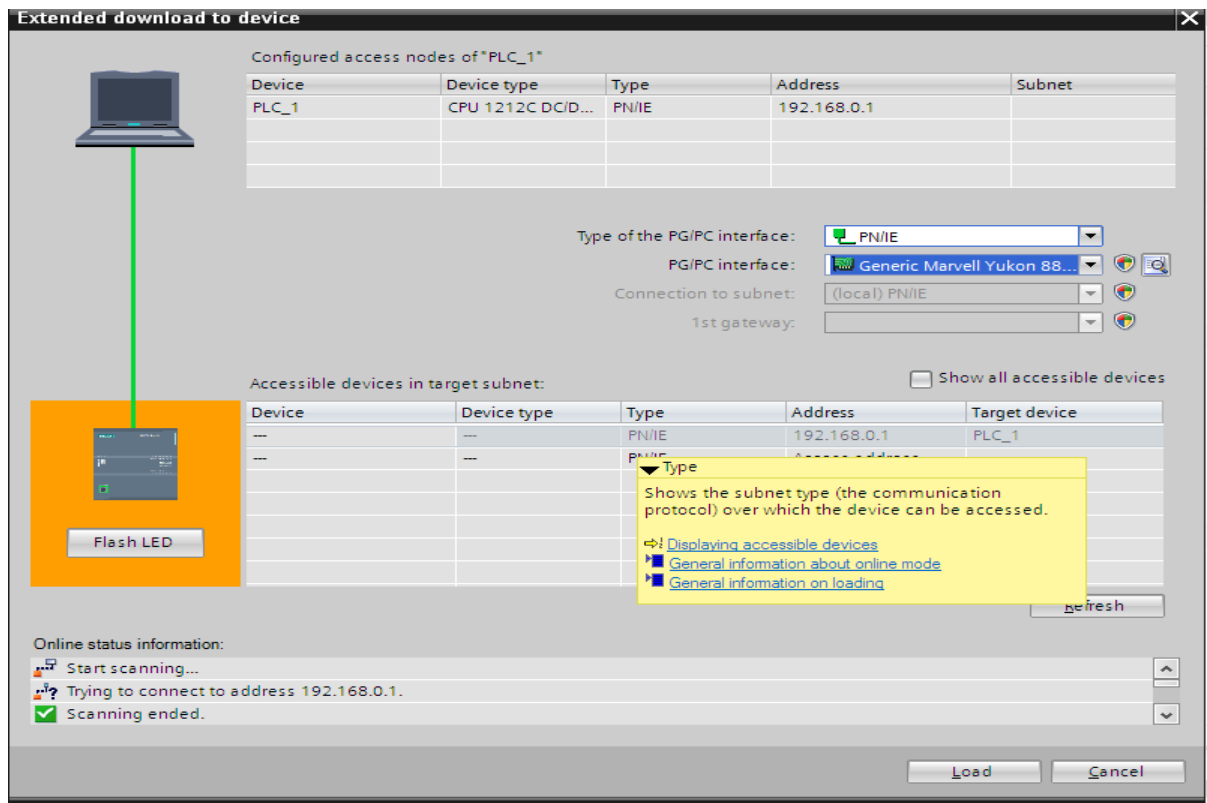
Thay đổi địa chỉ IP: nếu địa chỉ IP đã tồn tại, công cụ IP TOOL sẽ sửa đổi cấu hình phần cứng (HW config) của PLC S7-1200.

2.4.4.2. Đổ chương trình xuống CPU

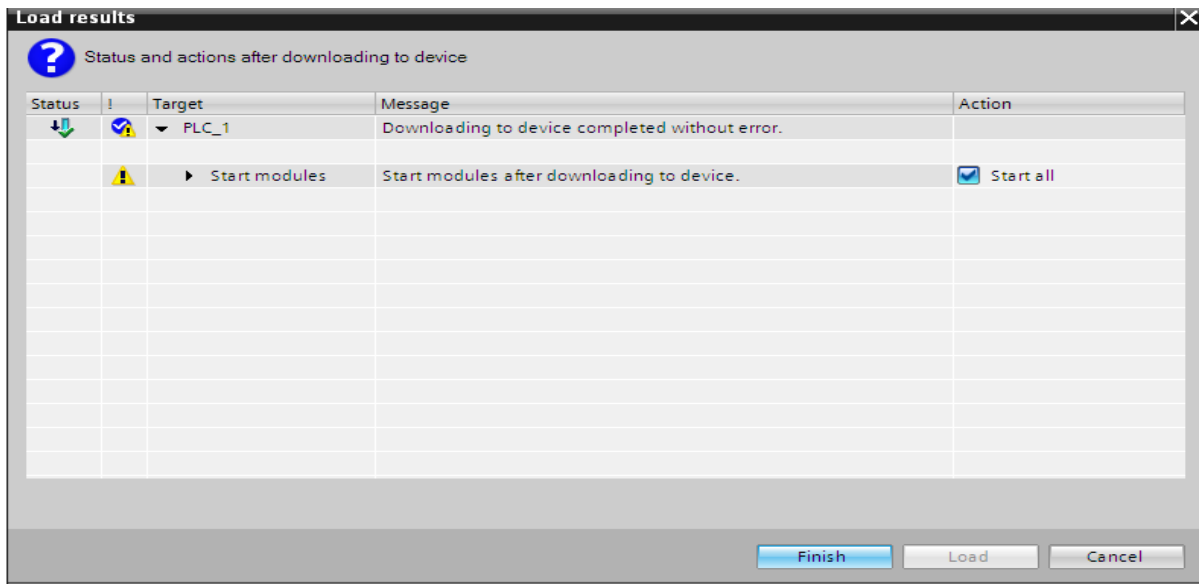
Đổ từ màn hình soạn thảo chương trình bằng cách kích vào biểu tượng download trên thanh công cụ của màn hình



Chọn cấu hình Type of the PG/PC interface và PG/PC interface như hình dưới sau đó nhấn chọn load

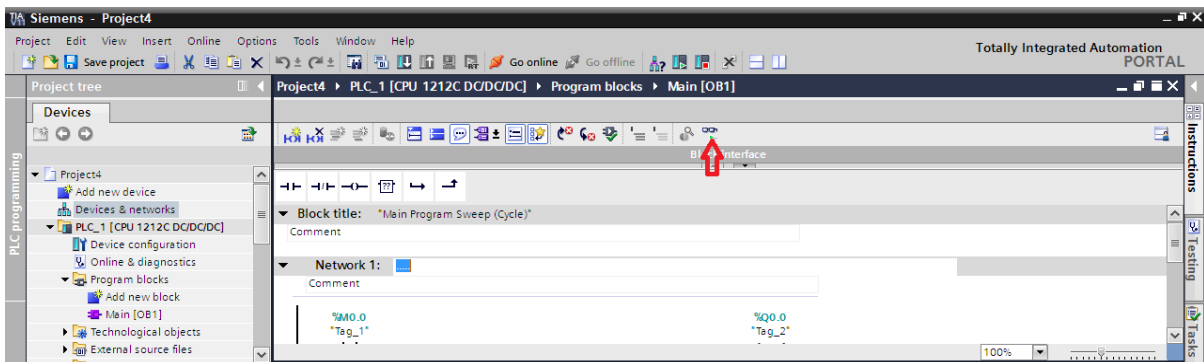


Chọn start all như hình vẽ và nhấn finish

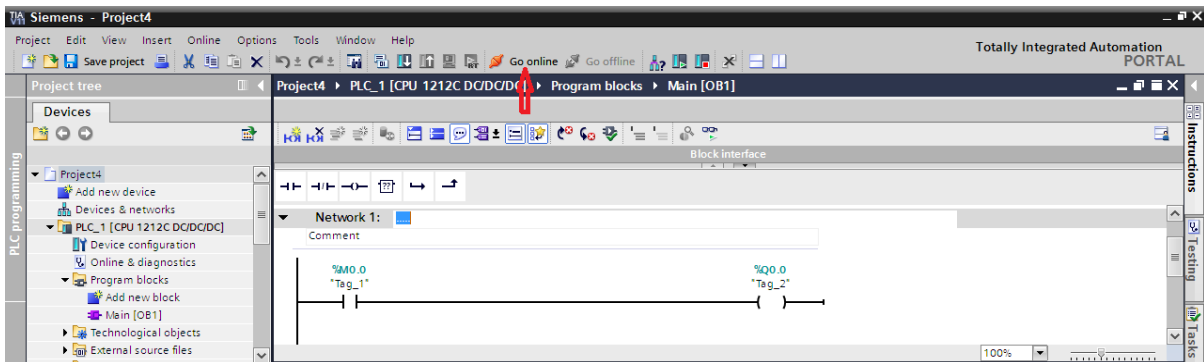


2.4.4.3. Giám sát và thực hiện chương trình

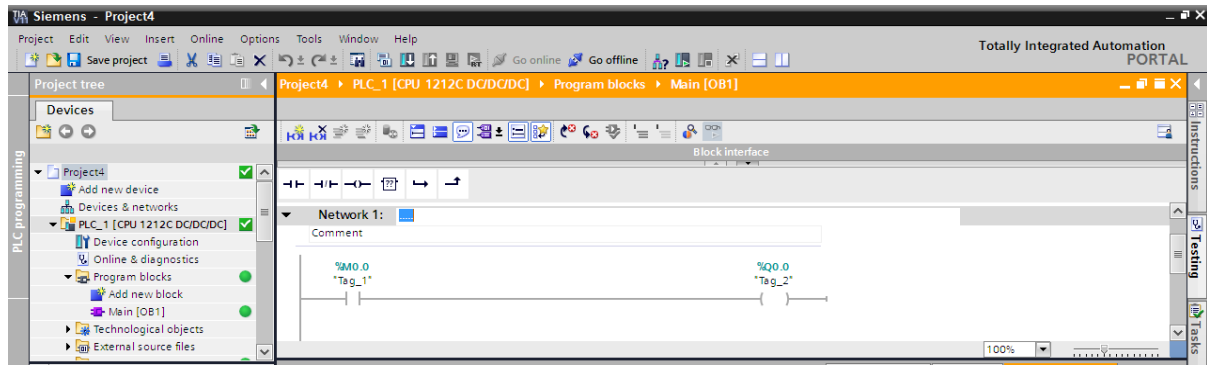
Để giám sát chương trình trên màn hình soạn thảo kích chọn Monitor trên thanh công cụ.



Hoặc cách 2 làm như hình dưới



Sau khi chọn monitor chương trình soạn thảo xuất hiện như sau:



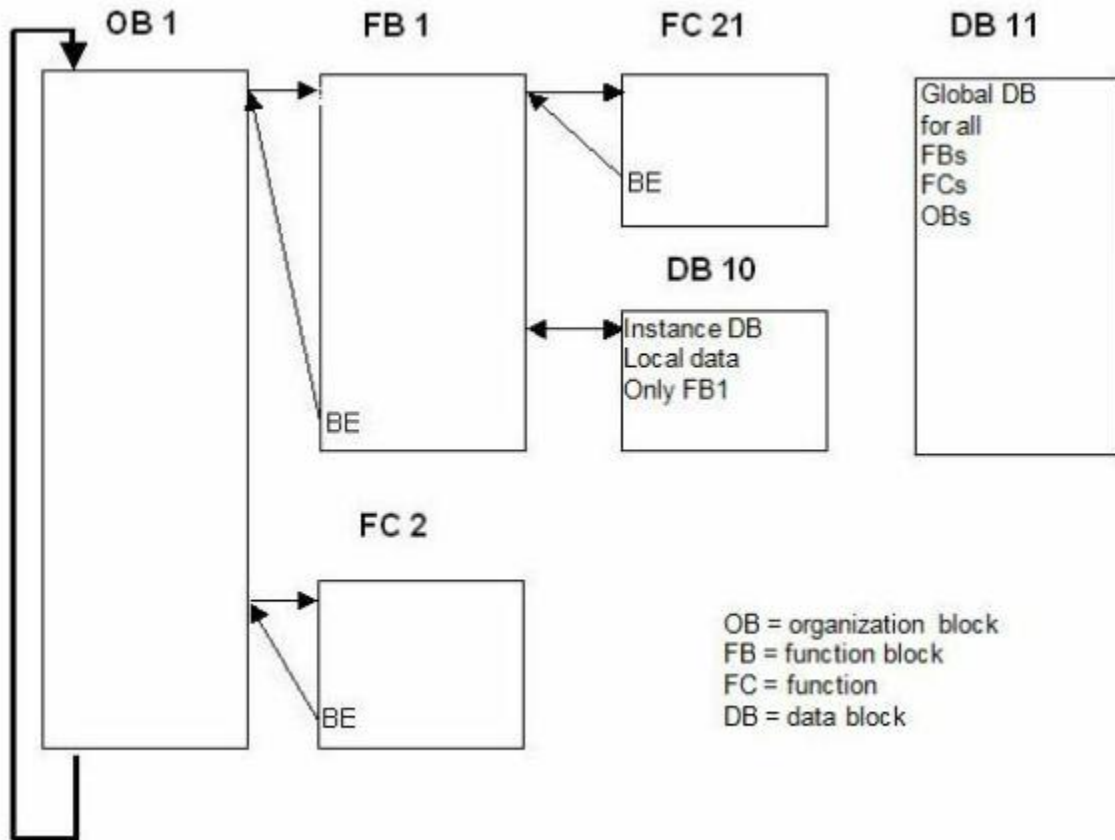
2.4.5. Kỹ thuật lập trình

2.4.5.1. Vòng quét chương trình

PLC thực hiện chương trình theo chu trình lặp. Mỗi vòng lặp được gọi là vòng quét. Mỗi vòng quét được bắt đầu bằng giai đoạn chuyển dữ liệu từ các cổng vào số tới vùng bộ đệm ảo I, tiếp theo là giai đoạn thực hiện chương trình. Trong từng vòng quét chương trình được thực hiện từ lệnh đầu tiên đến lệnh kết thúc của khối OB1. Sau giai đoạn thực hiện chương trình là giai đoạn chuyển các nội dung của bộ đệm ảo Q tới các cổng ra số. Vòng quét kết thúc bằng giai đoạn truyền thông nội bộ và kiểm tra lỗi.

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

2.4.5.2. Cấu trúc lập trình



Hình 2.3: Cấu trúc lập trình

2.4.5.3. Khối tổ chức OB – ORGANIZATION BLOCKS

Organization blocks (OBs) : là giao diện giữa hoạt động hệ thống và chương trình người dùng. Chúng được gọi ra bởi hệ thống hoạt động, và điều khiển theo quá trình:

- +Xử lý chương trình theo quá trình
- +Bảo động – kiểm soát xử lý chương trình
- +Xử lý lỗi

Startup OB, Cycle OB, Timing Error OB và Diagnosis OB : có thể chèn và lập trình các khối này trong các project. Không cần phải gán các thông số cho chúng và cũng không cần gọi chúng trong chương trình chính.

Process Alarm OB và Time Interrupt OB : Các khối OB này phải được tham số hóa khi đưa vào chương trình. Ngoài ra, quá trình báo động OB có thể được gán cho một sự kiện tại thời gian thực hiện bằng cách sử dụng các lệnh ATTACH, hoặc tách biệt với lệnh DETACH.

Time Delay Interrupt OB : OB ngắt thời gian trễ có thể được đưa vào dự án và lập trình. Ngoài ra, chúng phải được gọi trong chương trình với lệnh SRT_DINT, tham số là không cần thiết.

Start Information : Khi một số OB được bắt đầu, hệ điều hành đọc ra thông tin được thẩm định trong chương trình người dùng, điều này rất hữu ích cho việc chẩn đoán lỗi, cho dù thông tin được đọc ra được cung cấp trong các mô tả của các khối OB.

2.4.5.4. Hàm chức năng – FUNCTION

Functions (FCs) là các khối mã không cần bộ nhớ. Dữ liệu của các biến tạm thời bị mất sau khi FC được xử lý. Các khối dữ liệu toàn cầu có thể được sử dụng để lưu trữ dữ liệu FC.

Functions có thể được sử dụng với mục đích

+Trả lại giá trị cho hàm chức năng được gọi

+Thực hiện công nghệ chức năng, ví dụ : điều khiển riêng với các hoạt động nhị phân.

+Ngoài ra, FC có thể được gọi nhiều lần tại các thời điểm khác nhau trong một chương trình. Điều này tạo điều kiện cho lập trình chức năng lặp đi lặp lại phức tạp.

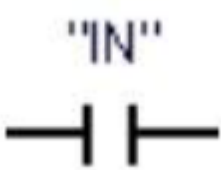
FB (function block) : đối với mỗi lần gọi, FB cần một khu vực nhớ. Khi một FB được gọi, một Data Block (DB) được gán với instance DB. Dữ liệu trong Instance DB sau đó truy cập vào các biến của FB. Các khu vực bộ nhớ khác nhau đã được gán cho một FB nếu nó được gọi ra nhiều lần.

DB (data block) : DB thường để cung cấp bộ nhớ cho các biến dữ liệu .
 Có hai loại của khối dữ liệu DB : Global DBs nơi mà tất cả các OB, FB và FC có thể đọc được dữ liệu lưu trữ, hoặc có thể tự mình ghi dữ liệu vào DB, và instance DB được gán cho một FB nhất định.


2.4.6. Giới thiệu các tập lệnh

2.4.6.1. Bit logic (tập lệnh tiếp điểm)

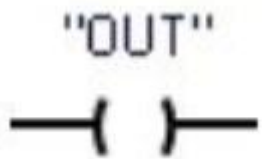
1) Tiếp điểm thường mở

L		Tiếp điểm thường mở sẽ đóng khi giá trị của bit có địa chỉ n là bằng 1. Toán hạng : I, Q, M, L, D.
A		
D		

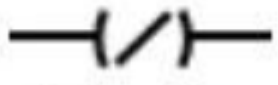
2) Tiếp điểm thường đóng

L		Tiếp điểm thường đóng sẽ đóng khi giá trị của bit có địa chỉ n là 0 Toán hạng n: I, Q, M, L, D
A		
D		


3) Lệnh OUT

L	 <p>Cuộn dây ngõ ra</p>	Giá trị của bit có địa chỉ là n sẽ bằng 1 khi đầu vào của lệnh này bằng 1 và ngược lại Toán hạng n : Q, M, L, D Chỉ sử dụng một lệnh out cho 1 địa chỉ
A		
D		

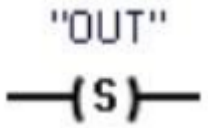
4) Lệnh OUT đảo

L A D		<p>Giá trị của bit có địa chỉ là n sẽ bằng 1 khi đầu vào của lệnh này bằng 0 và ngược lại</p> <p>Toán hạng n : Q, M, L, D</p> <p>Chỉ sử dụng một lệnh out not cho 1 địa chỉ</p>
-------------	---	---

5) Lệnh logic NOT

L A D		<p>Lệnh đảo trạng thái ngõ vào / ra</p>
-------------	---	---

6) Lệnh SET

L A D		<p>Giá trị của các bit có địa chỉ là n sẽ bằng 1 khi đầu vào của lệnh này bằng 1. Khi đầu vào của lệnh bằng 0 thì bit này vẫn giữ nguyên trạng thái.</p> <p>Toán hạng n: Q, M, L, D</p>
-------------	---	---

7) Lệnh Reset

L A D	<p>The symbol consists of the word "OUT" above a horizontal line. In the center of the line is a circle containing the letter "R".</p>	<p>Giá trị của các bit có địa chỉ là n sẽ bằng 0 khi đầu vào của lệnh này bằng 1. Khi đầu vào của lệnh bằng 0 thì các bit này vẫn giữ nguyên trạng thái.</p> <p>Toán hạng n: Q, M, L, D</p>
-------------	--	---


8) Lệnh set nhiều bit

L A D	<p>The symbol consists of the word "OUT" above a horizontal line. In the center of the line is a rectangle containing the text "SET_BF". Below the line is the letter "n".</p>	<p>Giá trị của các bit có địa chỉ đầu tiên là OUT sẽ bằng 1 khi đầu vào của lệnh này bằng 1 Khi đầu vào của lệnh bằng 0 thì các bit này vẫn giữ nguyên trạng thái. Trong đó số bit là giá trị của n</p> <p>Toán hạng OUT: Q, M, L, D n : là hằng số</p>
-------------	--	---

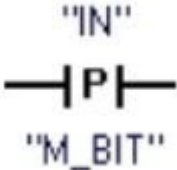
9) Lệnh reset nhiều bit

L A D	<p>The symbol consists of the word "OUT" above a horizontal line. In the center of the line is a rectangle containing the text "RESET_BF". Below the line is the letter "n".</p>	<p>Giá trị của các bit có địa chỉ đầu tiên là OUT sẽ bằng 0 khi đầu vào của lệnh này bằng 1 Khi đầu vào của lệnh bằng 0 thì các bit này vẫn giữ nguyên trạng thái. Trong đó số bit là giá trị của n</p> <p>Toán hạng OUT: Q, M, L, D n : là hằng số</p>
-------------	--	---


10) Tiếp điểm phát hiện xung cạnh lên dạng 1

<p>L A D</p>		<p>Tiếp điểm phát hiện cạnh lên sẽ phát ra một xung khi đầu vào tiếp điểm P có sự chuyển đổi từ mức thấp lên mức cao</p> <p>Trạng thái của tín hiệu được lưu lại vào "M_BIT"</p> <p>Độ rộng của xung này bằng thời gian của một chu kì quét.</p>
----------------------	---	--

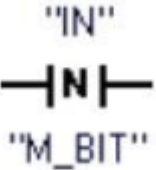
11) Tiếp điểm phát hiện xung cạnh lên dạng 2

<p>L A D</p>		<p>Thay đổi trạng thái tín hiệu phía trước không ảnh hưởng đến "IN"</p> <p>Phát hiện sự thay đổi trạng thái của 1 tín hiệu "IN" từ 0 lên 1</p> <p>Trạng thái của tín hiệu IN được lưu lại vào "M_BIT"</p> <p>Độ rộng của xung này bằng thời gian của một chu kì quét.</p>
----------------------	---	---


12) Tiếp điểm phát hiện xung cạnh xuống dạng 1

<p>L A D</p>		<p>Tiếp điểm phát hiện cạnh xuống sẽ phát ra một xung khi đầu vào tiếp điểm này có sự chuyển đổi từ mức cao xuống mức thấp</p> <p>Trạng thái của tín hiệu được lưu lại vào “M_BIT”</p> <p>Độ rộng của xung này bằng thời gian của một chu kì quét.</p>
----------------------	---	--

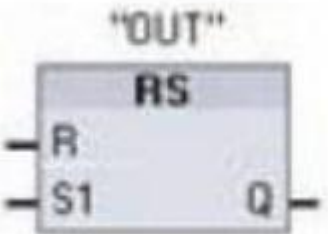
13) Tiếp điểm phát hiện xung cạnh xuống dạng 2

<p>L A D</p>		<p>Thay đổi trạng thái tín hiệu phía trước không ảnh hưởng đến “IN”</p> <p>Phát hiện sự thay đổi trạng thái của 1 tín hiệu “IN” từ 1 xuống 0</p> <p>Trạng thái của tín hiệu IN được lưu lại vào “M_BIT”</p> <p>Độ rộng của xung này bằng thời gian của một chu kì quét.</p>
----------------------	---	---

14) Lệnh SR fliplop

<p>L A D</p>		<p>Mạch chốt RS ưu tiên Reset</p>
----------------------	---	-----------------------------------

15) Lệnh RS fliplop


L A D		Mạch chốt RS ưu tiên Set
-------------	---	--------------------------

2.4.6.2. Sử dụng bộ Timer

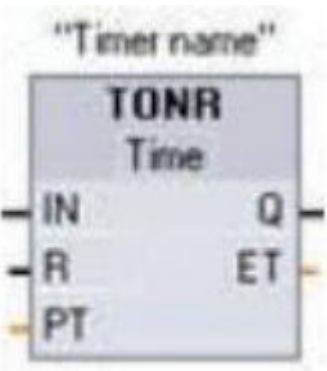
Sử dụng lệnh Timer để tạo một chương trình trễ định thời. Số lượng của Timer phụ thuộc vào người sử dụng và số lượng vùng nhớ của CPU. Mỗi timer sử dụng 16 byte IEC_Timer dữ liệu kiểu cấu trúc DB. Step 7 tự động tạo khối DB khi lấy khối Timer

Kích thước và tầm của kiểu dữ liệu Time là 32 bit, lưu trữ như là dữ liệu Dint : T#-14d_20h_31m_23s_648ms đến T#24d_20h_31m_23s_647ms hay là -2.147.483.648 ms đến 2.147.483.647 ms.

1) Timer tạo xung – TP

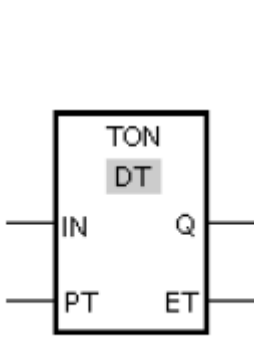
L A D		<p>Timer TP tạo một chuỗi xung với độ rộng xung đặt trước. Thay đổi PT, IN không ảnh hưởng khi Timer đang chạy.</p> <p>Khi đầu vào IN được tác động vào timer sẽ tạo ra một xung có độ rộng bằng thời gian đặt PT</p>
-------------	---	---

2) Timer trễ sườn lên có nhớ TONR

L A D		<p>Khi tổng thể tác động của ngõ vào lớn hơn hay bằng thời gian đặt PT thì Timer sẽ tác động và tiếp điểm thường mở của Timer sẽ chuyển lên mức 1. Và khi trạng thái Reset của Timer bị tác động thì Timer ngừng hoạt động và sẽ bị Reset lại.</p>
-------------	---	--

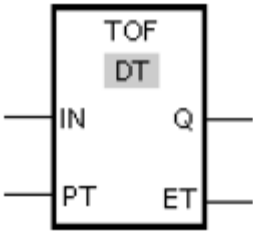
Tham số	Kiểu dữ liệu	Miêu tả
IN	Bool	Ngõ vào cho phép timer hoạt động
R	Bool	Reset timer
PT	Time	Thời gian đặt trước
Q	Bool	Ngõ ra
ET	Time	Thời gian thực hiện

3) Timer trễ không nhớ - TON

L A D		<p>Khi ngõ vào IN được tác động và duy trì trạng thái liên tục với thời gian hơn thời gian đặt thì ngõ ra Q sẽ chuyển lên mức 1. Khi ngõ vào ngừng tác động thì reset và dừng hoạt động Timer.</p> <p>Thay đổi PT khi Timer đang chạy không ảnh hưởng tới Timer</p>
-------------	---	---

Tham số	Khai báo	Kiểu dữ liệu	Vùng nhớ	Mô tả
IN	Input	BOOL	I, Q, M, D, L	Ngõ vào
PT	Input	TIME	I, Q, M, D, L or constant	Giá trị tham số PT phải là tích cực
Q	Output	BOOL	I, Q, M, D, L	Đầu ra được thiết lập khi thời gian PT hết
ET	Output	TIME	I, Q, M, D, L	Giá trị thời gian hiện tại

4) Timer trễ sườn xuống – TOF

L A D		<p>Khi ngõ vào tác động thì timer sẽ tác động và tiếp điểm thường mở của timer sẽ chuyển trạng thái lên 1. Khi ngõ vào ngừng tác động thì sau khoảng thời gian PT thì timer sẽ ngừng tác động.</p>
-------------	---	--

2.4.6.3. Sử dụng bộ Counter

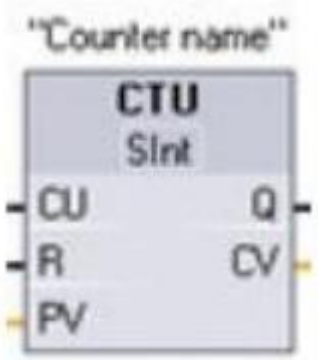
Lệnh Counter được dùng để đếm các sự kiện ở ngoài hay các sự kiện quá trình ở trong PLC. Mỗi Counter sử dụng cấu trúc lưu trữ của khối dữ liệu DB để làm dữ liệu của Counter. Step 7 tự động tạo khối DB khi lấy lệnh.

Tầm giá trị đếm phụ thuộc vào kiểu dữ liệu mà bạn chọn lựa. Nếu giá trị đếm là một số Integer không dấu, có thể đếm xuống tới 0 hoặc đếm lên

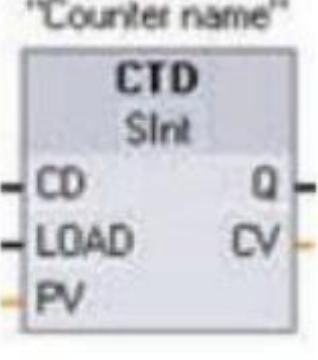
tới tầm giới hạn. Nếu giá trị đếm là một số interder có dấu, có thể đếm tới giá trị âm giới hạn hoặc đếm lên tới một số dương giới hạn.

Tham số	Kiểu dữ liệu	Miêu tả
CU, CD	Bool	Đếm lên hay đếm xuống
R	Bool	Reset giá trị đếm về 0
LOAD (CTD, CTUD)	Bool	Load giá trị đặt trước
PV	SInt, Int, Dint, UInt, UDIInt.	Giá trị đếm đặt trước
Q, QU	Bool	Mức 1 nếu $CV \geq PV$
QD	Bool	Mức 1 nếu $CV \leq 0$
CV	SInt, Int, Dint, UInt, UDIInt	Giá trị đếm hiện hành

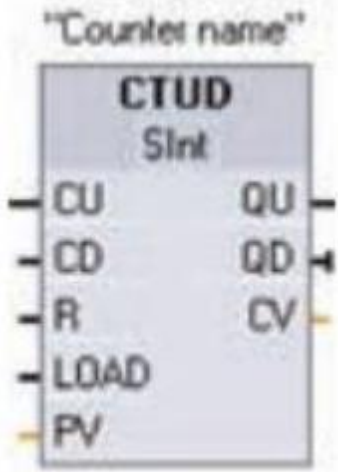
1) Counter đếm lên - CTU

L A D		<p>Giá trị bộ đếm CV được tăng lên 1 khi tín hiệu ngõ vào CU chuyển từ 0 lên 1. Ngõ ra Q được tác động lên 1 khi $CV \geq PV$. Nếu trạng thái R = Reset được tác động thì bộ đếm $CV = 0$.</p>
-------------	---	--

2) Counter đếm xuống – CTD

L A D		<p>Giá trị bộ đếm được giảm 1 khi tín hiệu ngõ vào CD chuyển từ 0 lên 1. Ngõ ra Q được tác động lên 1 khi $CV \leq 0$. Nếu trạng thái LOAD được tác động thì $CV = PV$.</p>
-------------	--	---

3) Counter đếm lên xuống – CTUD

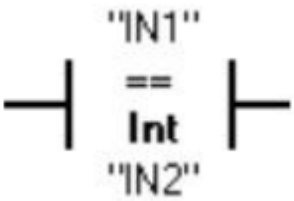
L A D		<p>Giá trị bộ đếm CV được tăng lên 1 khi tín hiệu ngõ vào CU chuyển từ 0 lên 1. Ngõ ra QU được tác động lên 1 khi $CV \geq PV$. Nếu trạng thái R = Reset được tác động thì bộ đếm $CV = 0$.</p> <p>Giá trị bộ đếm CV được giảm 1 khi tín hiệu ngõ vào CD chuyển từ 0 lên 1. Ngõ ra QD được tác động lên 1 khi $CV \leq 0$. Nếu trạng thái Load được tác động thì $CV = PV$.</p>
-------------	---	---

2.4.6.4. So sánh

1) Lệnh so sánh


So sánh 2 kiểu dữ liệu giống nhau, nếu lệnh so sánh thỏa thì ngõ ra sẽ là mức 1 = TRUE

Kiểu dữ liệu so sánh là : SInt, Int, Dint, USInt, UDInt, Real, LReal, String, Char, Time, DTL, Constant.


L A D		<p>Lệnh so sánh dùng để so sánh hai giá trị IN1 và IN2 bao gồm $IN1 = IN2$, $IN1 \geq IN2$, $IN1 \leq IN2$, $IN1 < IN2$, $IN1 > IN2$ hoặc $IN1 \triangleq IN2$</p> <p>So sánh 2 kiểu dữ liệu giống nhau, nếu lệnh so sánh thỏa thì ngõ ra sẽ là mức 1 = TRUE (tác động mức cao) và ngược lại</p>
-------------	---	--

		Kiểu dữ liệu so sánh là : SInt, Int, Dint, USInt, UDInt, Real, LReal, String, Char, Time, DTL, Constant.
--	--	--


2) Lệnh trong khoảng In – range

L A D		<p>Tham số : MIN, VAL, MAX</p> <p>Kiểu dữ liệu so sánh : SInt, Int, Dint, USInt, UInt, UDInt, Real, LReal, Constant</p> <p>So sánh 2 kiểu dữ liệu giống nhau, nếu so sánh $MIN \leq VAL \leq MAX$ thỏa thì tác động mức cao và ngược lại</p>
-------------	--	---


3) Lệnh ngoài khoảng out-of-range

L A D		<p>Tham số : MIN, VAL, MAX</p> <p>Kiểu dữ liệu so sánh : SInt, Int, Dint, USInt, UInt, UDInt, Real, LReal, Constant</p> <p>So sánh 2 kiểu dữ liệu giống nhau, nếu so sánh $MIN > VAL$ hoặc $MAX < VAL$ thỏa thì tác động mức cao và ngược lại</p>
-------------	---	---

4) Lệnh OK

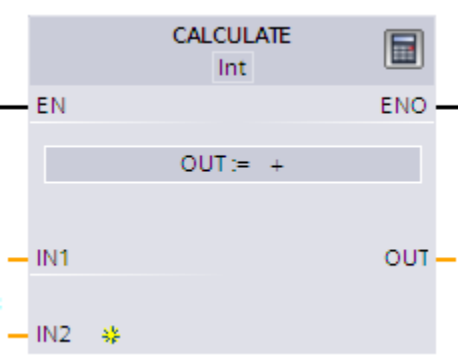
L A D		<p>Tham số : IN</p> <p>Kiểu dữ liệu : Real, LReal</p> <p>Lệnh OK kiểm tra tính hợp lệ của toán tử</p>
-------------	---	---

5) Lệnh NOT OK

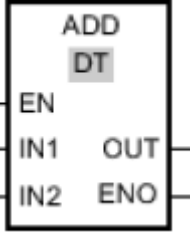
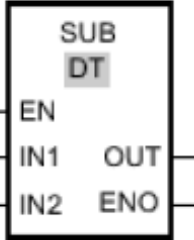
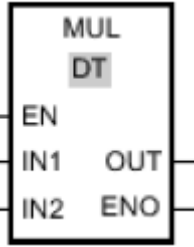
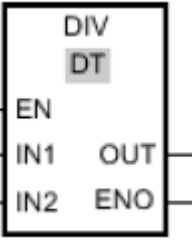
L A D		<p>Tham số : IN</p> <p>Kiểu dữ liệu : Real, LReal</p> <p>Lệnh NOT_OK kiểm tra tính không hợp lệ của toán tử</p>
-------------	--	---

2.4.6.5. Toán học

1) Lệnh tính toán

L A D		<p>Công dụng : thực hiện phép toán từ các giá trị ngõ vào IN1, IN2, IN(n) theo công thức $OUT = \dots (+, -, *, /)$ rồi xuất kết quả ra ngõ ra OUT.</p> <p>Các thông số ngõ vào dùng trong khối phải chung định dạng</p>
-------------	---	---

2) Lệnh cộng, trừ, nhân, chia

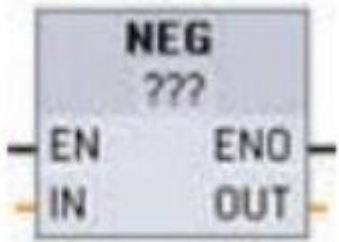
L A D	   	<p>Lệnh cộng ADD : $OUT = IN1 + IN2$</p> <p>Lệnh trừ SUB : $OUT = IN1 - IN2$</p> <p>Lệnh nhân MUL : $OUT = IN1 * IN2$</p> <p>Lệnh chia DIV : $OUT = IN1 / IN2$</p> <p>Tham số IN1, IN2 phải cùng kiểu dữ liệu : SInt, Int, Dint, USInt, UInt, UDIInt, Real, LReal, Constant</p> <p>Tham số OUT có kiểu dữ liệu : SInt, Int, Dint, USInt, UInt, UDIInt, Real, LReal</p> <p>Tham số ENO = 1 nếu không có lỗi xảy ra trong quá trình thực thi. Ngược lại ENO = 0 khi có lỗi, một số lỗi xảy ra khi thực hiện lệnh này :</p> <ul style="list-style-type: none"> -Kết quả toán học nằm ngoài phạm vi của kiểu dữ liệu. -Chia cho 0 ($IN2 = 0$) -Real/LReal : Nếu một trong những giá trị đầu vào là NaN sau đó được trả về NaN. -ADD Real/LReal : Nếu cả hai giá trị IN là INF có dấu khác nhau, đây là một khai báo không hợp lệ và được trả về NaN -SUB Real/LReal : Nếu cả hai giá trị IN là
-------------	---	--

		<p>INF cùng dấu, đây là một khai báo không hợp lệ và được trả về NaN</p> <p>-MUL Real/LReal : Nếu một trong 2 giá trị là 0 hoặc là INF, đây là khai báo không hợp lệ và được trả về NaN.</p> <p>-DIV Real/LReal : Nếu cả hai giá trị IN bằng không hoặc INF, đây là khai báo không hợp lệ và được trả về NaN.</p>
--	--	---


3) Lệnh lấy phần dư

L A D		<p>Lệnh Modulo sẽ lấy phần dư của phép toán. Giá trị ngõ vào IN1 chia cho IN2 và giá trị phần dư sẽ được lưu vào OUT</p> <p>Tham số:</p> <p>EN : Bool</p> <p>ENO : Bool</p> <p>IN1 : SINT, INT, DINT, USINT, UINT, UDINT</p> <p>IN2 : SINT, INT, DINT, USINT, UINT, UDINT</p> <p>OUT : SINT, INT, DINT, USINT, UINT, UDINT</p>
-------------	--	--

4) Lệnh phủ định

<p>L A D</p>		<p>Lệnh NEG đảo ngược dấu hiệu số học của giá trị ở trong tham số và lưu trữ các kết quả trong tham số OUT</p> <p>Tham số :</p> <p>EN : Bool – cho phép ngõ vào</p> <p>ENO: Bool – cho phép ngõ ra</p> <p>-ENO = 1 : không có lỗi</p> <p>-ENO = 0: kết quả giá trị nằm ngoài tầm giá trị của kiểu dữ liệu</p> <p>IN : toán tử đầu vào SInt, INT, Dint, Real, LReal, Constant</p> <p>OUT : toán tử đầu ra Sint, Int, Dint, Real, Lreal</p>
----------------------	---	---

5) Lệnh tăng, giảm

<p>L A D</p>		<p>Tăng / giảm giá trị kiểu số Integer lên / xuống một đơn vị</p> <p>Tham số :</p> <p>EN : cho phép ngõ vào</p> <p>IN/OUT : toán tử ngõ vào và ra</p> <p>ENO : cho phép ngõ ra</p> <p>-ENO = 1 : không có lỗi</p> <p>-ENO = 0: kết quả nằm ngoài tầm giá trị của kiểu dữ liệu</p>
----------------------	---	---


6) Lệnh giá trị tuyệt đối

L A D		<p>Tính giá trị tuyệt đối của một số nguyên hoặc số thực của tham số IN và lưu trữ kết quả vào tham số OUT</p> <p>Tham số :</p> <p>EN : cho phép ngõ vào</p> <p>IN : Toán tử ngõ vào</p> <p>OUT : Toán tử ngõ ra</p> <p>ENO : Cho phép ngõ ra</p>
-------------	--	---

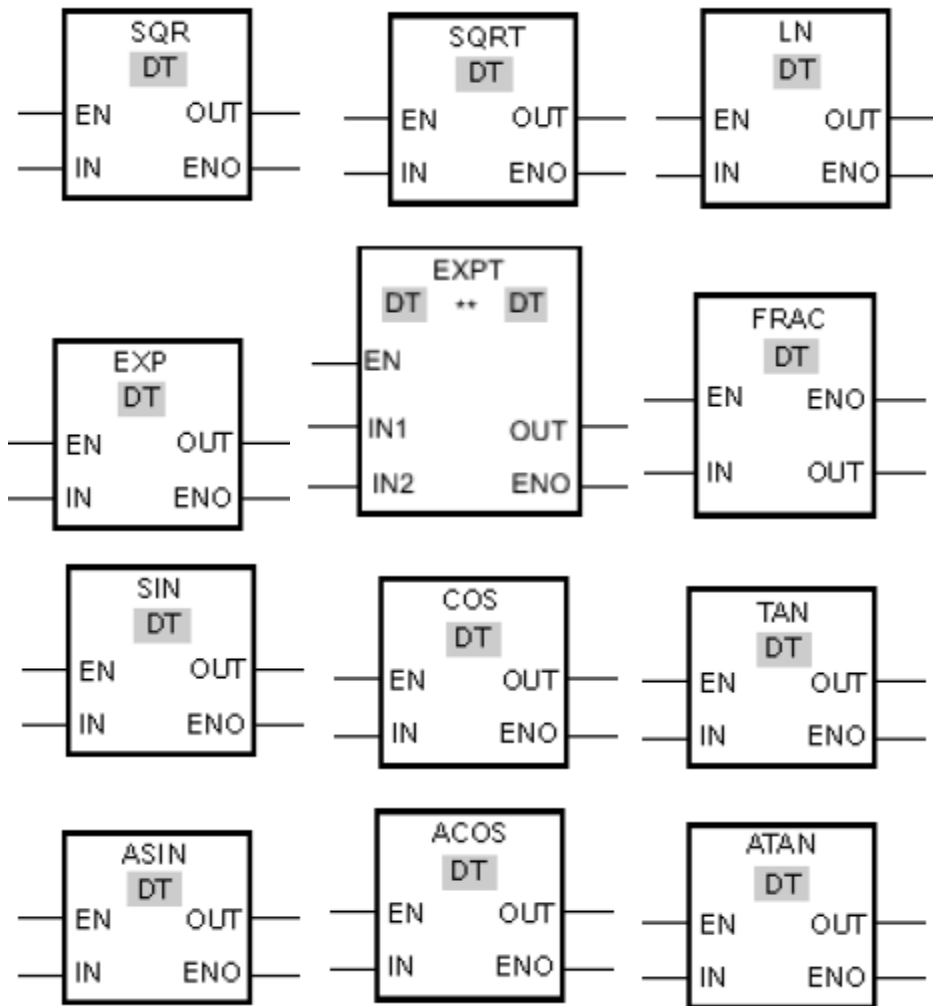
7) Lệnh giá trị nhỏ nhất và lớn nhất

L A D		<p>Lệnh MIN/MAX so sánh các giá trị đầu vào và trả lại giá trị nhỏ nhất/ lớn nhất ở đầu ra</p> <p>Tham số :</p> <p>EN : cho phép ngõ vào</p> <p>IN : Toán tử đầu vào, có thể lên tới 32 đầu vào</p> <p>OUT : Toán tử ngõ ra</p> <p>ENO : cho phép ngõ ra</p>
-------------	--	--

8) Lệnh giới hạn

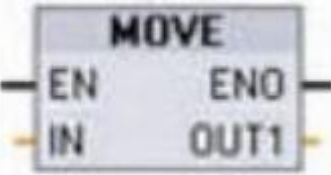
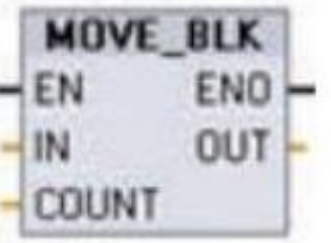
L A D		<p>Công dụng : Giới hạn giá trị của ngõ vào IN trong khoảng của ngõ vào MIN và MAX. Nếu giá trị của IN đáp ứng $MIN < IN < MAX$ thì giá trị của IN được copy vào giá trị của OUT. Còn nếu giá trị của $IN < MIN$ thì giá trị của MIN được copy vào OUT, và nếu giá trị của $IN > MAX$ thì giá trị của MAX được copy vào OUT</p> <p>Lệnh chỉ được thực hiện khi tín hiệu ngõ vào là 1 tại ngõ vào EN, Nếu lệnh được thực hiện mà không có lỗi xảy ra thì tại ngõ ra ENO cũng có giá trị bằng 1.</p> <p>Ngõ ra ENO có trạng thái 0 nếu 1 trong số các điều kiện sau đây không thỏa mãn :</p> <ul style="list-style-type: none">-Ngõ vào EN có tín hiệu “0”-Các thông số nhập vào không đúng định dạng-Các toán hạng không đúng giá trị-Giá trị Min lớn hơn giá trị Max
-------------	--	--

9) Lệnh toán học số thực dấu chấm động

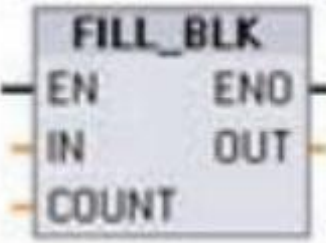


2.4.6.6. Di chuyển MOVE

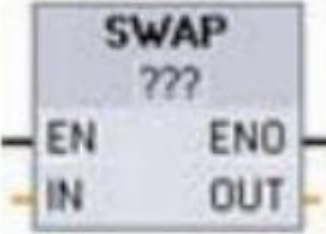
1) Lệnh MOVE

L A D	 	<p>Lệnh Move di chuyển nội dung ngõ vào IN đến ngõ ra OUT mà không làm thay đổi giá trị ngõ IN</p> <p>Tham số:</p> <p>EN : cho phép ngõ vào</p> <p>ENO : cho phép ngõ ra</p> <p>IN : nguồn giá trị đến</p> <p>OUT1: Nơi chuyển đến</p> <p>Lệnh Move_BLK sao chép các nội dung của một vùng nhớ IN đến một bộ nhớ xác định khác. Số lượng các giá trị được sao chép được quy định trong COUNT. Hoạt động sao chép theo hướng tăng dần các địa chỉ</p> <p>Tham số:</p> <p>EN : cho phép ngõ vào</p> <p>ENO : cho phép ngõ ra</p> <p>IN : nguồn giá trị đến</p> <p>COUNT : số giá trị sao chép</p> <p>OUT1: Nơi chuyển đến</p>
-------------	--	---

2) Lệnh làm đầy FILL

L A D		<p>Công dụng : dùng để lấp đầy một vùng nhớ với nội dung tại một vùng nhớ khác. Lệnh Fill block di chuyển nội dung của một vùng nhớ tới một vùng nhớ xác định. Hành động vận chuyển các biến sao chép theo hướng tăng dần</p>
-------------	---	---

3) Lệnh đảo Swap

L A D		<p>Công dụng : Đổi thứ tự của 2 byte hay 4 byte thành phần của một Word hay một Dword. Nó không làm đổi thứ tự của các bit trong mỗi byte</p>
-------------	---	---

2.4.6.7. Chuyển đổi

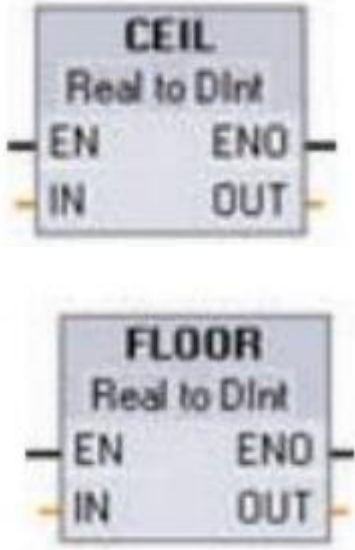
1) Lệnh CONV

L A D		<p>Công dụng : chuyển đổi từ kiểu dữ liệu này sang kiểu dữ liệu khác</p> <p>Tham số :</p> <p>IN : giá trị ngõ vào</p> <p>OUT : giá trị sau khi chuyển đổi</p>
-------------	--	---

2) Lệnh làm tròn ROUND và cắt bỏ TRUNCATE

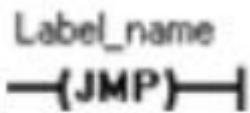
L A D		<p>Lệnh ROUND : Chuyển đổi số thực thành số Interger. Các phần phân số của số thực được làm tròn đến số nguyên gần nhất. Nếu số thực nằm ở giữa 2 số nguyên thì số thực này được làm tròn thành số nguyên chẵn. Ví dụ $ROUND(10.5) = 10$, $ROUND(11.5) = 12$.</p> <p>Lệnh TRUNC : chuyển đổi số thực thành số interger. Phần phân số của số thực bị cắt bỏ</p>
-------------	--	--

3) Lệnh CEILING và FLOOR


L A D		<p>Lệnh CEIL : chuyển đổi số thực thành số nguyên nhỏ nhất lớn hơn hay bằng số thực đó</p> <p>Lệnh FLOOR : Chuyển đổi số thực thành số nguyên nhỏ hơn hya bằng số thực đó.</p>
-------------	---	--

2.4.6.8. Lệnh điều khiển chương trình


1) Lệnh nhảy JUMP và nhãn LABEL

L A D		<p>Công dụng : Dừng chương trình đang chạy và tiếp tục trên một network khác, network này được xác định bởi 1 jump label.</p>
-------------	---	---

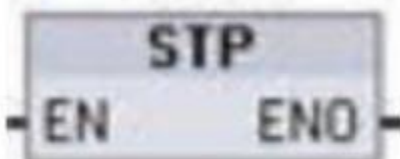
2) Lệnh điều khiển thực thi RET

L A D		Công dụng : Để dừng việc thực thi trong một khối hàm và chỉ được tiếp tục sau khi có lệnh gọi khối hàm đó.
-------------	---	--

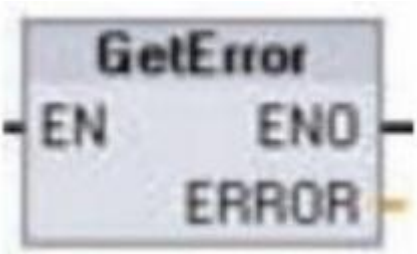
3) Lệnh Re – trigger giám sát quét chu kỳ

L A D		Công dụng : Khởi động lại việc giám sát chu kỳ của CPU. Thời gian giám sát được cấu hình trong phần cứng. Việc khởi động lại thời gian giám sát chu kỳ để ngăn chặn lỗi.
-------------	---	--

4) Lệnh ngừng quét chu kỳ

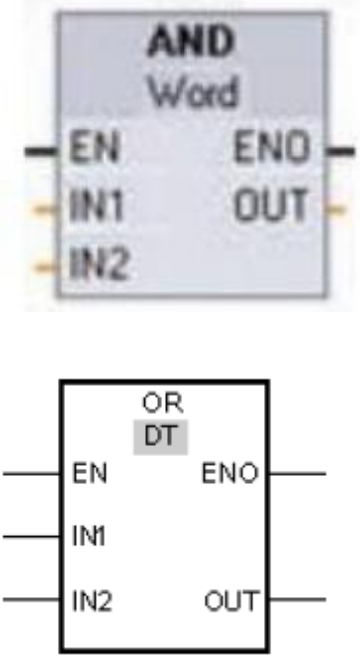
L A D		Công dụng : Đặt PLC về chế độ STOP, do vậy ngừng việc thực hiện chương trình
-------------	---	--

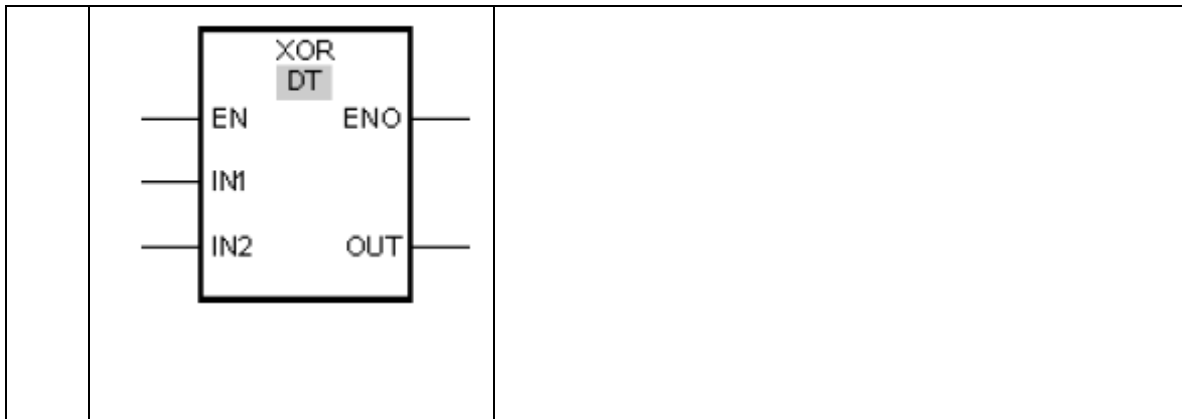
5) Lệnh lấy lỗi GET ERROR

L A D		Công dụng : Truy vấn các lỗi xảy ra trong một khối
-------------	---	--

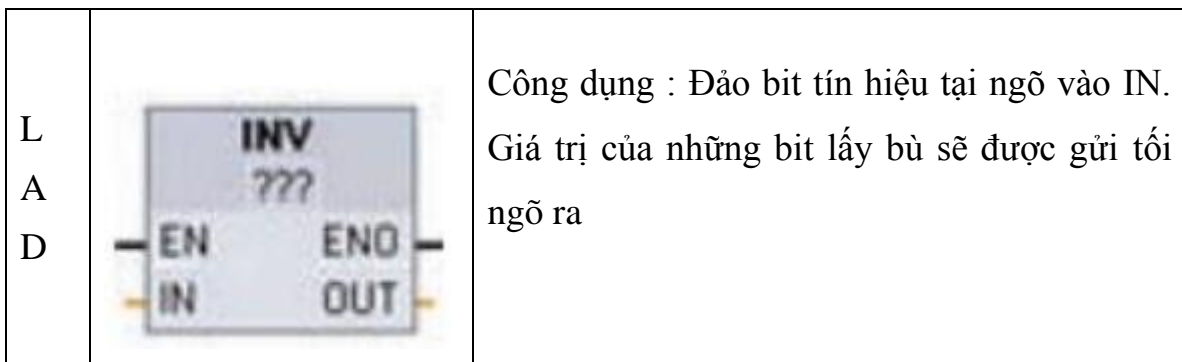
2.4.6.9. Toán tử word logic

1) Lệnh AND, OR và XOR

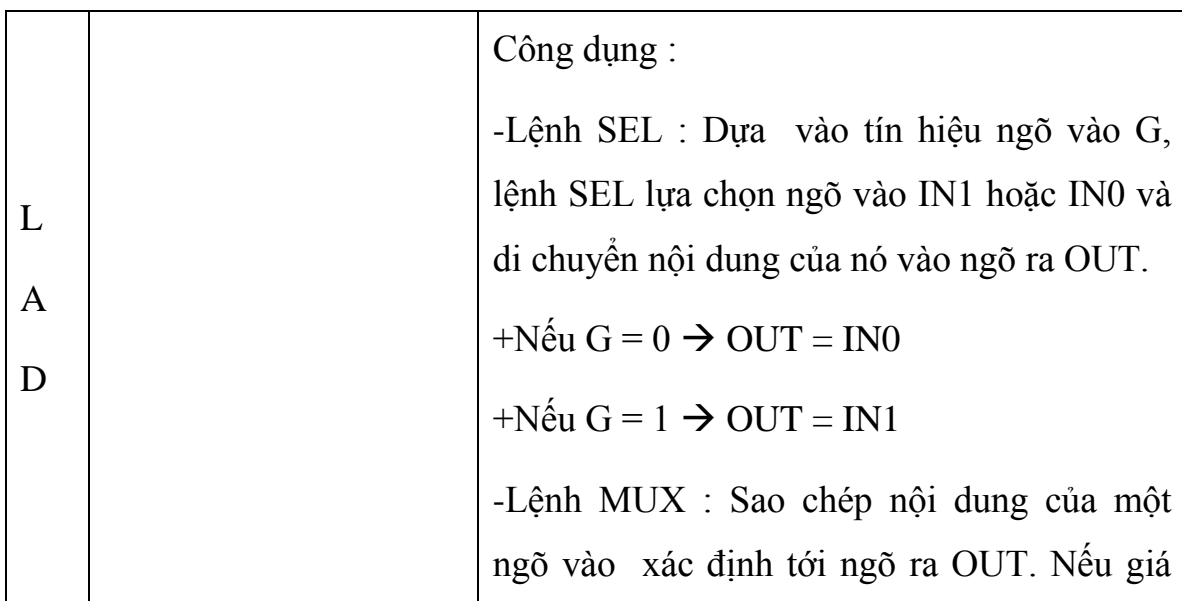
L A D		Công dụng : -Lệnh AND kết hợp các giá trị ngõ vào IN1 và IN2 theo các bit tương ứng theo phép AND logic, xuất kết quả tại OUT - Lệnh OR kết hợp các giá trị ngõ vào IN1 và IN2 theo các bit tương ứng theo phép OR logic, xuất kết quả tại OUT - Lệnh XOR kết hợp các giá trị ngõ vào IN1 và IN2 theo các bit tương ứng theo phép XOR logic, xuất kết quả tại OUT
-------------	---	--

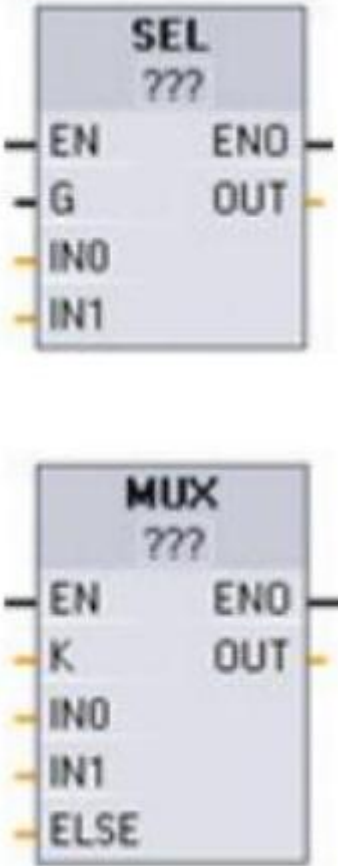


2) Lệnh đảo INVERT




3) Lệnh SELECT, MULTIPLEX và DEMULTIPLEX



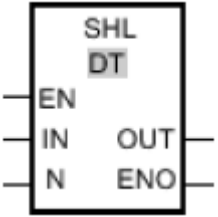
		<p>trị của tham số K lớn hơn số ngõ vào hiện hữu thì nội dung của tham số ELSE sẽ được sao chép tới ngõ ra OUT</p>
--	--	--

2.4.6.10. Dịch chuyển và xoay vòng

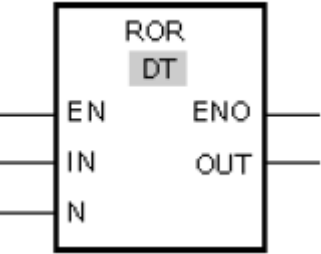
1) Lệnh dịch phải

L A D		<p>Công dụng : Dịch chuyển nội dung của toán hạng tại ngõ vào IN theo từng bit sang bên phải và truy xuất kết quả ra ngõ ra OUT.</p> <p>Thông số N để xác định số bit dịch chuyển</p>
-------------	---	---


2) Lệnh dịch trái

L A D		<p>Công dụng : Dịch chuyển nội dung của toán hạng tại ngõ vào IN theo từng bit sang bên trái và truy xuất kết quả ra ngõ ra OUT.</p> <p>Thông số N để xác định số bit dịch chuyển</p>
-------------	---	---

3) Lệnh quay phải

L A D		<p>Công dụng : Xoay nội dung của một toán hạng tại ngõ vào IN theo từng bit về hướng bên phải và truy xuất tại ngõ ra OUT</p> <p>Thông số N xác định số bit dịch chuyển</p>
-------------	--	---

4) Lệnh quay trái

L A D		<p>Công dụng : Xoay nội dung của một toán hạng tại ngõ vào IN theo từng bit về hướng bên trái và truy xuất tại ngõ ra OUT</p> <p>Thông số N xác định số bit dịch chuyển</p>
-------------	---	---

CHƯƠNG 3: XÂY DỰNG HỆ THỐNG CHUÔNG BÁO CA LÀM VIỆC TẠI NHÀ MÁY BẰNG PLC S7-1200

3.1. YÊU CẦU CÔNG NGHỆ CỦA CHUÔNG ĐIỆN

- Chuông thiết kế phải báo chuông vào/ra ca làm việc chuẩn xác, đúng giờ.
- Chuông thiết kế thông minh với việc ngừng hoạt động vào những ngày lễ, tết (dương lịch) ngày chủ nhật.
- Việc chuyển đổi thời gian đơn giản, để có thể ứng dụng cho nhiều nhà máy khác nhau.
- Chuông báo ca làm việc phải có hai chế độ auto và manual.
- Ở chế độ manual phải giới hạn được thời gian tối đa chuông reo.

3.2. CẤU TẠO VÀ NGUYÊN LÝ HOẠT ĐỘNG CỦA CHƯƠNG TRÌNH GIÁM SÁT

Tạo ra một mô hình chuông báo ca làm việc ở một nhà máy có thể hoạt động tốt, từ đó có thể thiết kế được hệ thống chuông báo ca làm việc tự động hoàn chỉnh cho các nhà máy.

Việc chế tạo mô hình hoạt động tốt sẽ tạo điều kiện cho sinh viên có cơ hội học tập nghiên cứu môn học một cách thực tế, là một cơ hội tốt cho sinh viên khỏi bỡ ngỡ khi làm việc thực tế.

Nghiên cứu chế tạo mô hình này sinh viên cũng được tham khảo thực tế nhiều lĩnh vực và tham khảo bằng nhiều tài liệu khác nhau. Điều đó mang lại sự hiểu biết sâu sắc hơn cho sinh viên không chỉ trong một lĩnh vực tự động hóa mà còn nhiều lĩnh vực khác, ngành nghề khác như điện, điện tử, cơ khí...

3.3. LỰA CHỌN THIẾT BỊ CHO MÔ HÌNH

Các thiết bị sử dụng trong hệ thống gồm có:

- PLC S7- 1200
- Chuông điện
- Nút nhấn
- Role 24VDC/280VAC
- Đèn báo
- Nguồn 24VDC

3.3.1. PLC S& - 1200



Hình 3.1

PLC S7 -1200: Thiết bị điều khiển chính của toàn bộ mô hình thiết bị này dùng nguồn xoay chiều 220v. Chức năng điều khiển theo chương trình lập trình sẵn theo chương trình cho trước.

3.3.2. Chuông điện



Hình 3.2 : Chuông điện

Chuông điện sử dụng nguồn điện xoay chiều 220VAC. Chế tạo dựa trên nguyên lý điện từ trường. Dùng lại búa gỗ. Chuông điện kêu Reng – Reng phù hợp lắp đặt tại các nhà máy.

3.3.3. Đèn báo

Đèn báo dùng cho các tủ điện, có các màu đỏ, vàng, xanh lá cây, trắng, xanh dương. Loại đèn này sử dụng công nghệ LED, đường kính 22mm

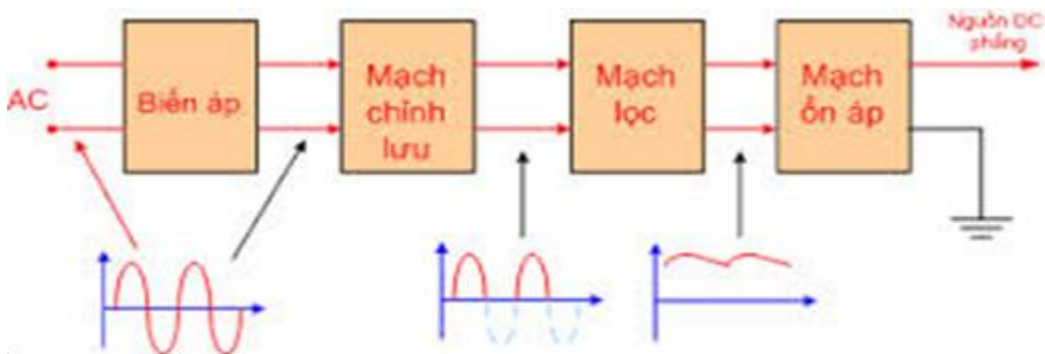


Hình 3.3 : các đèn báo

3.3.4. Bộ nguồn

Trong các mạch điện tử của các thiết bị như Radio – Cessette, âmly, tivi, đầu DVD... chúng ta sử dụng nguồn một chiều DC ở các mức điện áp khác nhau, nhưng ở ngoài zase cắm trực tiếp của các thiết bị này lại là vào nguồn điện AC 220V 50Hz, như vậy các thiết bị điện tử cần có một bộ phận để chuyển đổi nguồn xoay chiều thành ra điện áp một chiều, cung cấp cho các mạch trên, bộ chuyển đổi bao gồm:

- Biến áp nguồn: Hạ thế từ điện áp 220V xuống các mức điện áp thấp hơn như 6V, 9V, 12V...
- Mạch chỉnh lưu: Đổi điện áp AC thành DC
- Mạch lọc: Lọc gợn xoay chiều sau chỉnh lưu cho nguồn DC phẳng hơn
- Mạch ổn áp: Giữ một điện áp cố định cung cấp cho tải tiêu thụ.

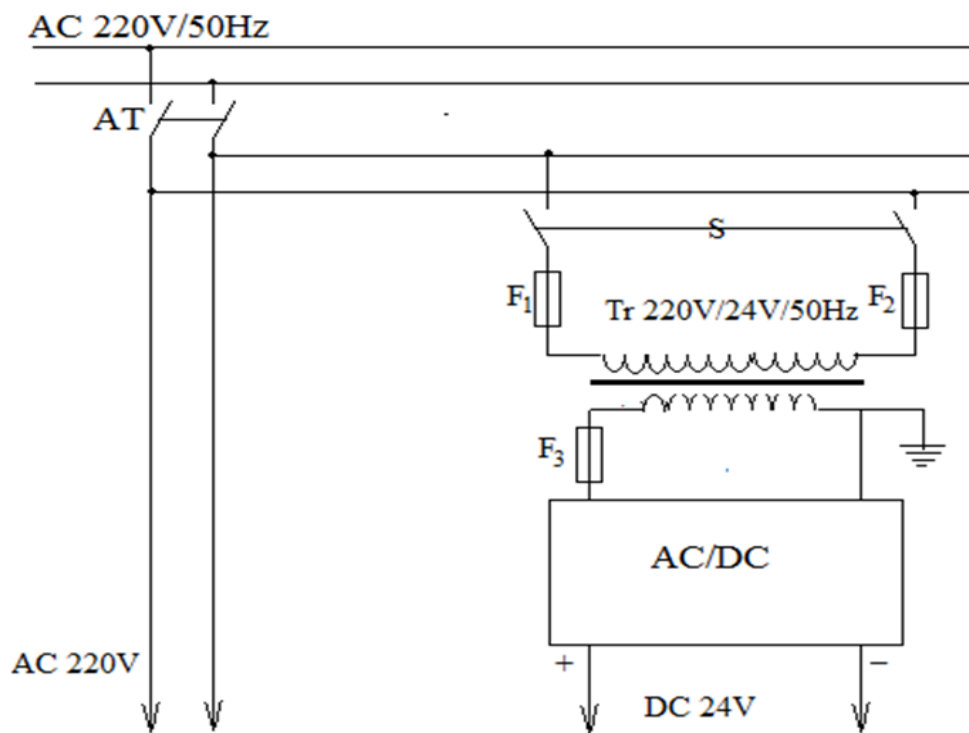


Hình 3.4 : Sơ đồ tổng quát của mạch cấp nguồn

3.4. LẬP TRÌNH CHƯƠNG TRÌNH ĐIỀU KHIỂN CHUÔNG BÁO CA LÀM VIỆC CỦA NHÀ MÁY.

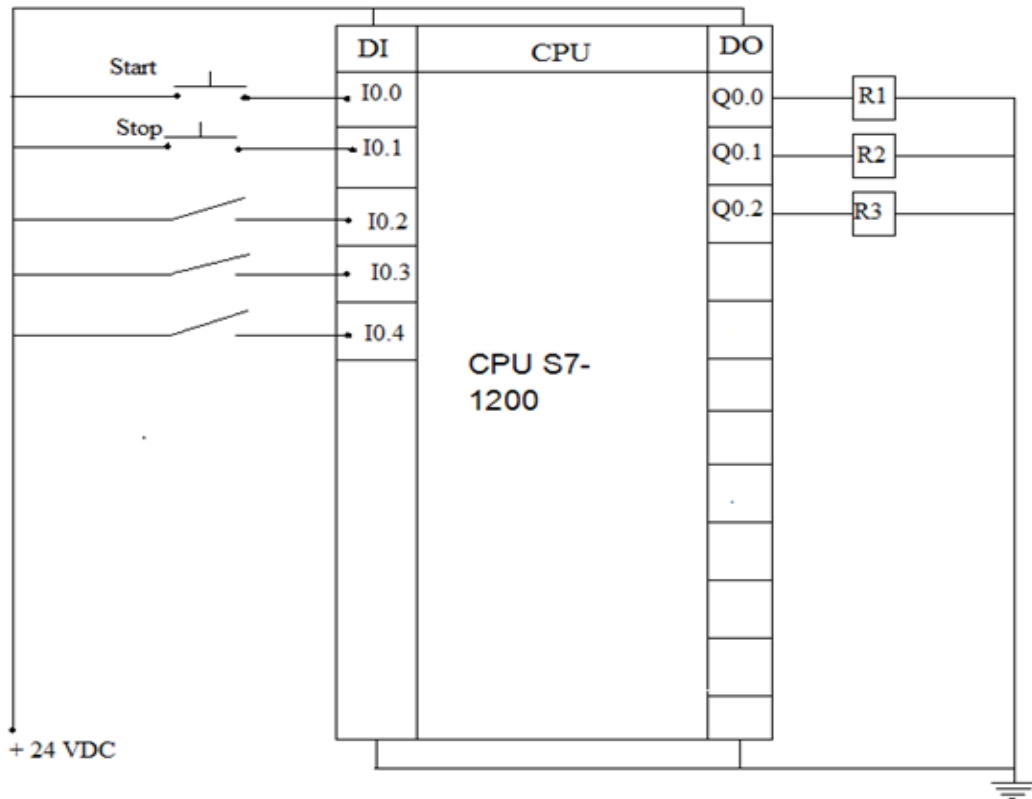
3.4.1. Sơ đồ nối điện cho PLC

Sơ đồ mạch cấp nguồn:



Hình 3.5 : Mạch cấp nguồn cho PLC

3.4.2. Sơ đồ nguyên lý đấu dây qua PLC



3.4.3. Các đầu vào/ra PLC

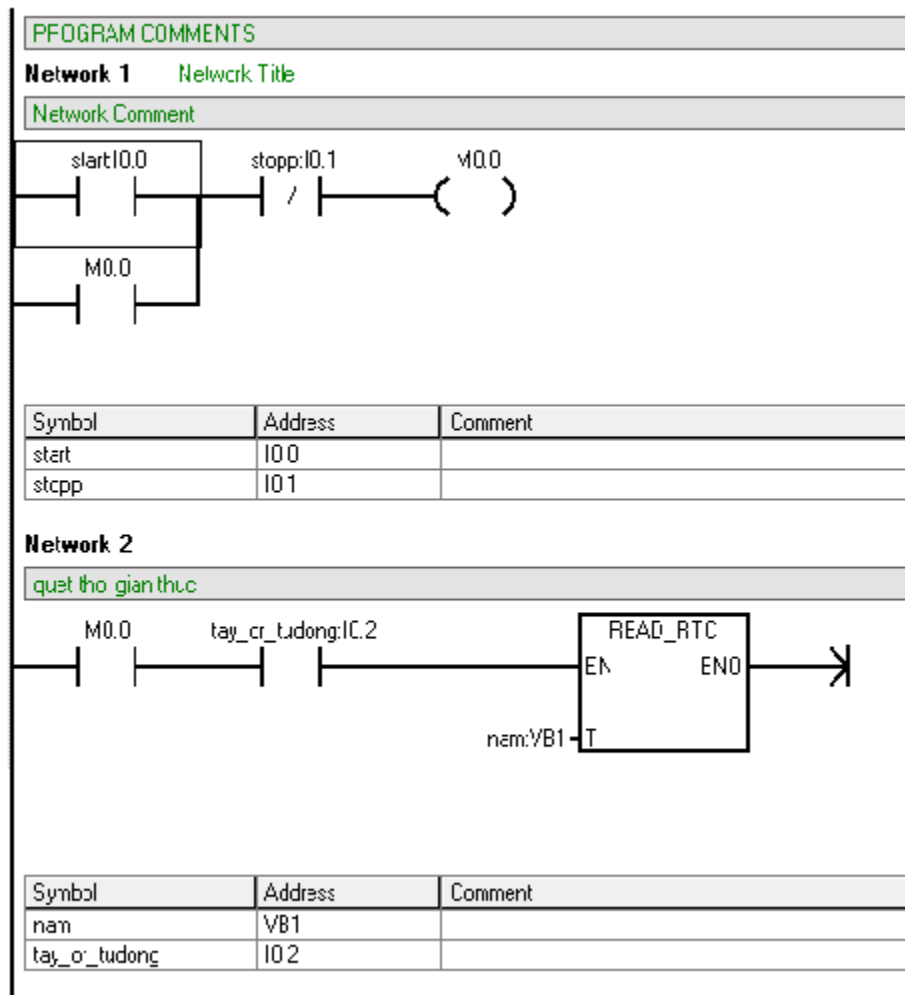
3.4.3.1. Đầu vào PLC

STT	Ký hiệu	Địa chỉ	Ghi chú
1	Start	I0.0	Khởi động hệ thống
2	Stop	I0.1	Dừng hệ thống
3	Auto/Manual	I0.2	Nút lựa chọn chế độ điều khiển tự động hoặc bằng tay

3.4.3.2. Đầu ra PLC

STT	Ký hiệu	Địa chỉ	Ghi chú
1	chuông	Q0.0	
2	Đèn báo tay	Q0.1	
3	Đèn báo tự động	Q0.2	

3.4.5. Chương trình PLC



Network 3

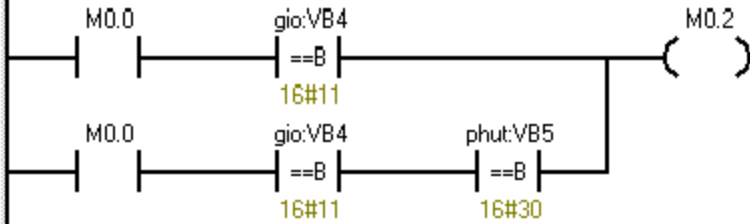
vao ca 1



Symbol	Address	Comment
gio	VB4	
phut	VB5	

Network 4

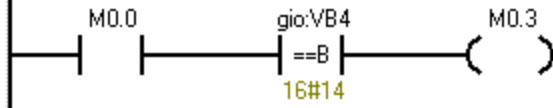
nghi giua ca 1



Symbol	Address	Comment
gio	VB4	
phut	VB5	

Network 5

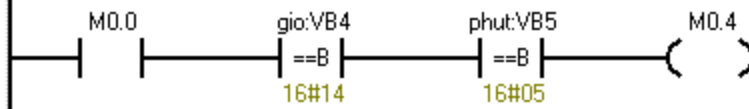
het ca 1



Symbol	Address	Comment
gio	VB4	

Network 6

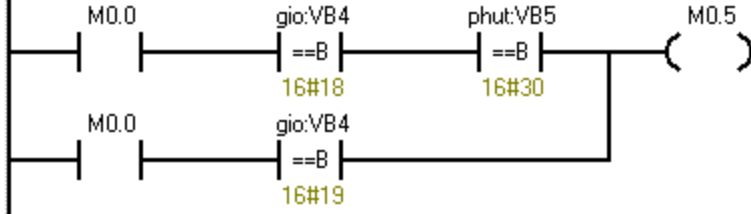
vao ca 2



Symbol	Address	Comment
gio	VB4	
phut	VB5	

Network 7

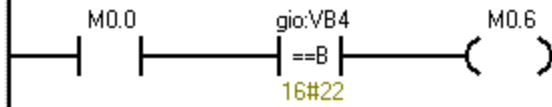
nghi giua ca 2



Symbol	Address	Comment
gio	VB4	
phut	VB5	

Network 8

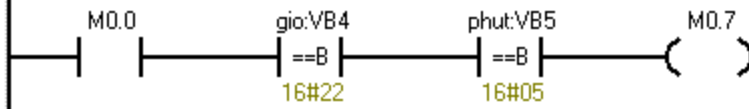
het ca 2



Symbol	Address	Comment
gio	VB4	

Network 9

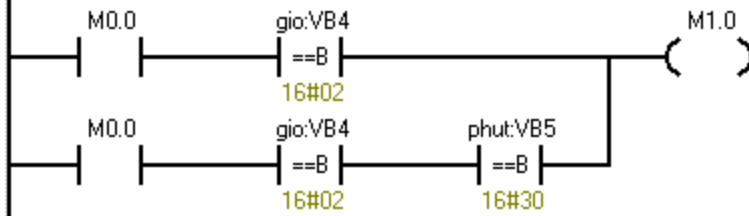
vao ca 3



Symbol	Address	Comment
gio	VB4	
phut	VB5	

Network 10

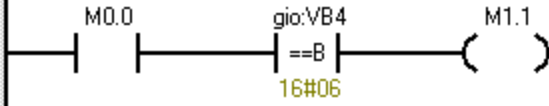
nghi giua ca 3



Symbol	Address	Comment
gio	VB4	
phut	VB5	

Network 11

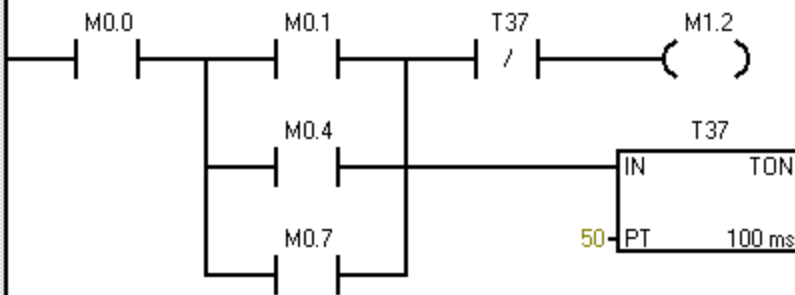
het ca 3



Symbol	Address	Comment
gio	VB4	

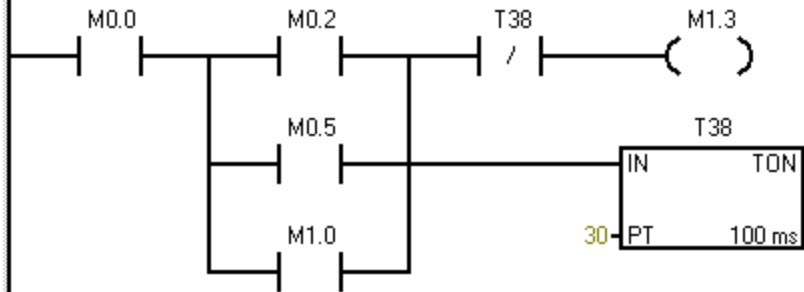
Network 12

chuong bao vao ca lam viec (5s)



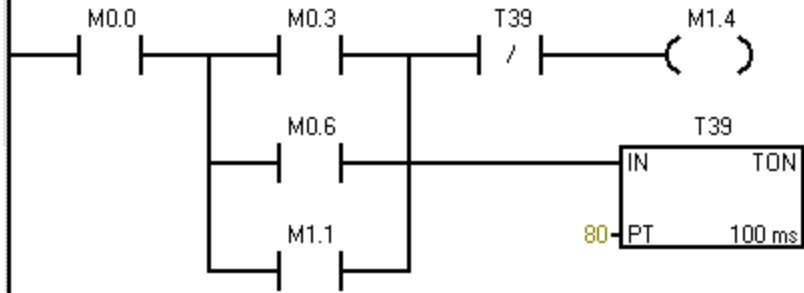
Network 13

chuong bao nghi giua ca (3s)



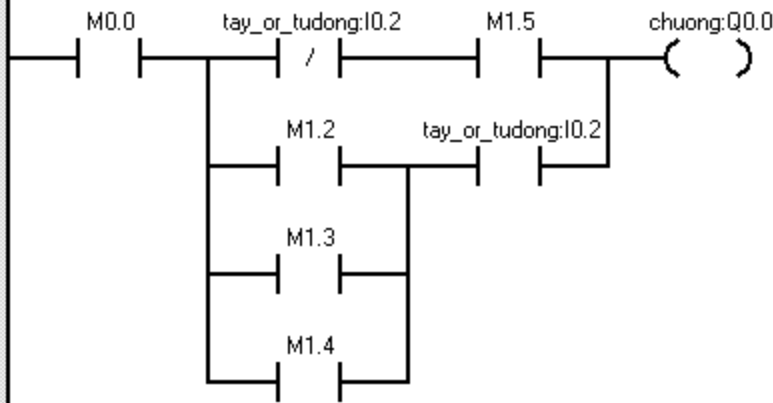
Network 14

chuong bao het ca (8s)



Network 16

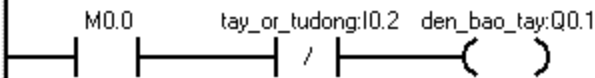
chuong bao



Symbol	Address	Comment
chuong	Q0.0	
tay_or_tudong	I0.2	

Network 17

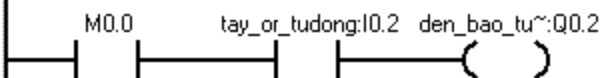
den bao dk tay



Symbol	Address	Comment
den_bao_tay	Q0.1	
tay_or_tudong	I0.2	

Network 18

den bao dk tu dong



Symbol	Address	Comment
den_bao_tudong	Q0.2	
tay_or_tudong	I0.2	

KẾT LUẬN

Qua thời gian thực hiện đồ án với tên đề tài "Ứng dụng PLC S7-1200 vào việc điều khiển chuông báo ca làm việc tại nhà máy" em đã đạt được những nội dung sau:

- Tìm hiểu được cấu trúc và hoạt động của chuông điện.
- Tìm hiểu và làm chủ được hoạt động của PLC S7 -1200 trong việc điều khiển chuông báo giờ trong nhà máy sản xuất.
- Thiết kế và xây dựng mô hình điều khiển chuông điện.

Đồ án này của em thực hiện dựa trên nghiên cứu tìm hiểu điều khiển chuông điện tại các nhà máy. Thông qua đề tài "**Ứng dụng PLC S7-1200 vào việc điều khiển chuông báo ca làm việc tại nhà máy**" thực sự giúp em hiểu rõ hơn về những gì em đã học được trong suốt thời gian qua. Qua đây em cũng được dịp mở rộng tầm hiểu biết của mình về mảng kiến thức PLC mà em đã được học, một ứng dụng tối ưu của ngành tự động hóa.

Do trình độ cũng như khả năng nhận thức còn có hạn, cộng với sự thiếu thốn về tài liệu tham khảo và thời gian nghiên cứu, tìm hiểu đề tài còn hạn chế nên dù đã rất cố gắng nhưng không thể tránh khỏi thiếu sót. Em mong nhận được sự chỉ bảo tận tình của các thầy cô để có thể hiểu hơn và tiếp cận gần hơn với công nghệ mới.

Em xin chân thành cảm ơn thầy giáo Nguyễn Trọng Thắng đã hướng dẫn và giúp đỡ em hoàn thành bản đồ án này. Đồng thời em cũng xin cảm ơn các thầy cô đã dạy dỗ em trong những năm học vừa qua, nhờ các thầy cô mà em mới có được kiến thức như ngày hôm nay. Đó chính là những kiến thức cơ bản giúp em thực hiện tốt nhiệm vụ tốt nghiệp và làm nền tảng cho công việc của em sau này.

Em xin chân thành cảm ơn!