

# HƯỚNG DẪN LẬP TRÌNH CĂN BẢN PLC MASTER-K

## Chương I: BỘ ĐIỀU KHIỂN LOGIC THEO CHƯƠNG TRÌNH ( PLC)



Bộ điều khiển Logic theo chương trình (**Programmable Logic Controller**) là sản phẩm của ứng dụng kỹ thuật vi xử lý để điều khiển các thiết bị, máy móc, dây chuyền tự động hóa ...

Các vấn đề trước đây được giải quyết bằng các mạch logic relay, các cơ cấu cơ khí kém linh hoạt, khó thay đổi thì nay được mềm hóa bằng các chương trình điều khiển bằng vi xử lý.

Xét về mặt phân tích, thiết kế về qui trình công nghệ thì công sức bỏ ra như nhau. Nếu như tủ điều khiển thiết kế bằng các linh kiện phụ rời càng phức tạp thì chương trình viết tương ứng cho bộ điều khiển lập trình (PLC) cũng phức tạp tương tự.

Tuy nhiên nếu dùng bộ PLC thì còn có khả năng lập trình nâng cao để tăng chất lượng điều khiển. Nếu dùng phụ kiện rời, người ta có khuynh hướng thiết kế vừa đủ đạt yêu cầu ví thêm tính năng có nghĩa là thêm phụ kiện rời gây ảnh hưởng trực tiếp đến giá thành.

Ngoài ra với sản xuất số lượng nhiều nếu dùng PLC chỉ tốn công sức cho một lần lập trình mà thôi. Có thể viết chương trình ở dạng Module tổng quát thực hiện được nhiều chức năng có thể dùng chung cho nhiều ứng dụng. Khi đó công việc lập trình giảm được nhiều công sức.

Một hệ PLC yêu cầu cần phải có những đặc tính sau: dễ sử dụng, đa năng và rẻ tiền. Do đó chúng thường được thiết kế trên cơ sở phần cứng gồm: bộ vi xử lý, bộ nhớ, bộ xuất nhập.

PLC có những chức năng bên trong như: bộ định thì (Timer), bộ đếm (Counter) và các thanh ghi dịch (Shift Registers). Nó hoạt động dựa trên nguyên tắc: **Các tín hiệu vào được gửi tới PLC và được lưu trữ trong bộ nhớ, PLC xử lý các tín hiệu vào dựa theo những lệnh logic đã được lập trình, sau đó kết quả xử lý được gửi đến cơ cấu chấp hành.**

Trong bộ điều khiển logic theo chương trình (PLC), một công việc được thực hiện hoàn toàn phụ thuộc vào chương trình điều khiển cài đặt trong bộ nhớ.

Các chuẩn giao tiếp được xây dựng ở PLC cho phép chúng liên lạc trực tiếp với cơ cấu đọc/ghi quá trình và những bộ phận chuyển đổi mà không cần phải có các relay hay mạch trung gian.

Việc sử dụng những bộ PLC có thể giúp cho người sử dụng không cần phải có những thao tác tháo lắp, nối dây mất nhiều thời gian khi cần thay đổi chương trình điều khiển, người sử dụng chỉ cần ngồi ở nơi và điều khiển bàn phím cùng với một ít thời gian lập trình là có thể thay đổi hoàn toàn điều khiển cũ thành điều khiển mới.

Mặc dù PLC tương tự máy tính thông thường nhưng do yêu cầu làm việc trong môi trường công nghiệp nên phần cứng của nó được thiết kế với những đặc điểm sau:

- Thiết bị có cấu tạo chắc chắn, có khả năng chống nhiễu cao

- Được thiết kế theo kiểu Module nên cho phép dễ dàng thay thế, sửa chữa và ghép thêm vào những Module khác.
- Ngôn ngữ lập trình dễ hiểu, dễ học.
- Dễ lập trình và lập trình lại cho các ứng dụng trong những nhà máy công nghiệp.

Thỏa mãn những đặc điểm trên, hãng **LS Industrial Systems** (LSIS) đã cho ra đời PLC với 02 họ MASTER-K và GLOFA ngoài ra còn có các màn hình giao diện kết hợp với PLC giúp cho các bạn thiết kế các máy móc thể hệ mới cũng như cải tiến công nghệ sản xuất trong quá trình tự động hóa hiện đại hóa đất nước.

PLC của LSIS được chia làm nhiều loại tùy theo mục đích nhu cầu: loại nhỏ (Micro), loại trung bình (Compact), loại lớn (Module). Về hình thức bố trí bên ngoài thường không giống nhau, về tập lệnh cũng khác nhau. Tuy nhiên chúng có chung một đích là được thiết kế dễ dàng thực hiện, thay thế cho các logic điều khiển máy móc, thiết bị bằng phương pháp truyền thống.

Tuy nhiên cũng cần phân biệt rõ sự khác biệt giữa logic điều khiển thực hiện bằng relay, phụ kiện rời hỗ trợ khác (như bộ định thì, bộ đếm, bộ cam phân chia thời gian, ...) với logic điều khiển được lập trình bằng PLC.

Logic điều khiển relay hoạt động trong chế độ thời gian thực. Các sự kiện, biến cố xảy ra thực sự và nó phản ứng ngay khi có tín hiệu tác động. Còn logic thực hiện theo chương trình bằng vi xử lý thì hoạt động có khác biệt. Đó là hoạt động xử lý theo từng bước trong chương trình, lập lại tuần hoàn. Thường thì có thể phân làm 3 giai đoạn hoạt động như sau:

**Giai đoạn 1:** Đọc dữ liệu ở ngõ nhập (dạng Binary hoặc Analog) vào PLC và lưu trữ trong bộ nhớ (lưu trữ ở dạng Binary)

**Giai đoạn 2:** Xử lý dữ liệu theo trật tự logic đã sắp đặt trong chương trình. Kết quả logic sẽ được lưu trữ trong bộ nhớ.

**Giai đoạn 3:** Xuất kết quả ra ngoài PLC, tín hiệu ở ngõ ra PLC sẽ đưa đến cơ cấu chấp hành.

**Trong thời gian thực hiện từ giai đoạn 1 đến giai đoạn 3, PLC chỉ phản ứng có một lần đối với tín hiệu vào và chỉ có một lần khống chế, kiểm soát cơ cấu chấp hành.**

Thực tế với tốc độ nhanh của vi xử lý hiện nay, thì sự chậm trễ đáp ứng của PLC không ra gây ra tác động đáng kể. Với kinh nghiệm trong khi lập trình cũng như sự hỗ trợ về phần cứng của các MODULE trong cùng một họ PLC, cùng với khả năng xử lý song song, nối mạng của PLC người ta có thể giải quyết được nhiều bài toán thực tế.

Bảng so sánh sau đây sẽ trình bày những ưu điểm của bộ điều khiển PLC so với các hệ điều khiển khác:

<b>Đặc tính</b>	<b>Hệ relay</b>	<b>Hệ mạch số</b>	<b>Máy tính</b>	<b>PLC</b>
Phí tổn trên một đơn vị chức năng	Hơi thấp	Thấp	Cao	Thấp
Kích thước	Thô	Khít khao, chắc chắn gọn nhẹ	Hơi gọn	Rất gọn
Tốc độ hoạt động	Chậm	Rất nhanh	Hơi nhanh	Nhanh
Tính chống nhiễu về điện	Rất tốt	Tốt	Hơi tốt	Tốt
Lắp đặt	Tốn nhiều thời gian thiết kế và lắp đặt	Tốn nhiều thời gian thiết kế	Tốn nhiều thời gian cho lập trình	Lập trình đơn giản và lắp đặt nhanh
Khả năng thực hiện các chức năng phức tạp	Không	Có	Có	Có
Khả năng thay đổi các chức năng	Rất khó	Khó	Hơi dễ	Rất dễ
Bảo quản	Khó khăn do nhiều tiếp điểm	Khó nếu IC hàn mạch	Khó	Dễ, có Card chuẩn

Qua bảng so sánh trên chúng ta thấy rõ hơn về những lợi điểm của PLC so với các hệ điều khiển khác.



# HƯỚNG DẪN LẬP TRÌNH CĂN BẢN PLC MASTER-K

## Chương II: GIỚI THIỆU PLC MASTER-K CỦA LSIS



Hiện nay tùy thuộc vào công nghệ của từng máy mà chúng ta sẽ lựa chọn số lượng I/O (Input/Output) của PLC cho phù hợp, số lượng I/O các lớn thì giá thành càng cao. Trong họ Master-K được chia làm nhiều loại: K14P1(8 Input/6 Output), K80S (Max: 80 I/O), K120S, K200, K300 (Max:384 I/O)...

### MASTER-K14P1



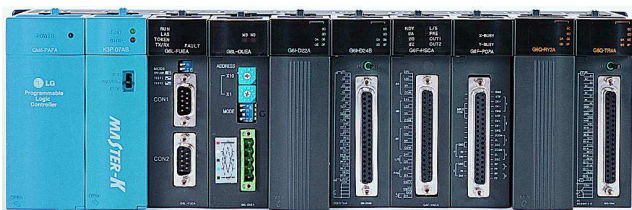
- Smallest Micro PLC
- Program Capacity: 800 Step
- RS-232C Interface
- Built-in High Speed Counter
- I/O Points: 8/6 Points

### MASTER-K80S/120S



- Compact Size
- Program Capacity: 7K Step
- RS-232C Interface
- High Speed Performance: 0.5  $\mu$ s/Step
- I/O Points: 10 +80 Points

### MASTER-K200S

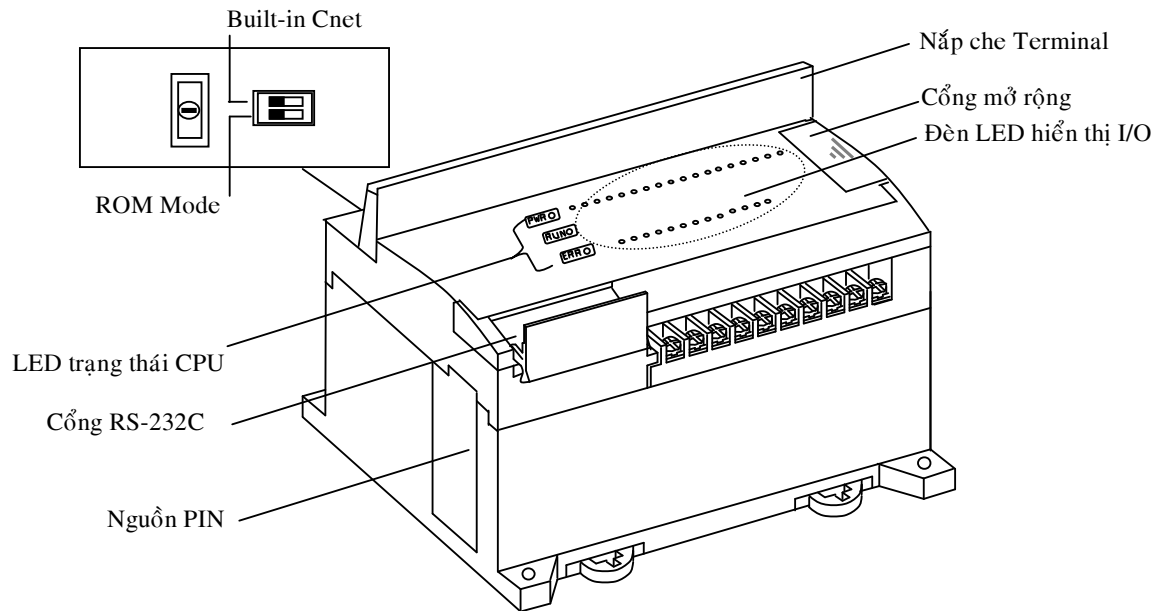


- High Speed Performance: 0.5  $\mu$ s/Step
- Program Capacity: 7K Step
- RS-232C Interface
- Communication Module: RS232C, RS-422/485, Modbus, Fieldbus(Fnet).
- I/O Points: Up to 384 Points

Các màn hình giao diện với PLC bao gồm: XP10BKB/DC, PMU-330BT, PMU-330TT (TFT colour LCD, 16 colour), PMU-530TT, PMU-830TT...



Đối với các PLC cỡ nhỏ như loại Master-K14P1, Master-K80S hay Master-K120S các bộ phận được kết hợp với nhau thành một khối. Với những loại lớn như Master-K200S, K300S hay K1000S được thiết kế theo dạng Module để người sử dụng có thể lựa chọn được một cấu hình PLC phù hợp nhất mà ít tốn kém nhất, đồng thời vẫn đáp ứng được những ứng dụng của mình.



Một bộ PLC thông thường bao gồm các thành phần sau:

➤ **BỘ VI XỬ LÝ (CPU):**

CPU là bộ não của PLC, nó điều khiển và kiểm soát tất cả mọi hoạt động bên trong của PLC, nó thực hiện các lệnh đã được chương trình hóa lưu trữ bên trong bộ nhớ. Một hệ thống BUS mang thông tin đến và đi từ CPU, bộ nhớ và bộ xuất nhập cũng chịu điều khiển của CPU.

CPU được cung cấp bởi một tần số đồng hồ do tinh thể thạch anh bên ngoài hay một mạch dao động RC, mạch dao động này có nhiệm vụ tạo ra tần số dao động từ 1 ÷ 8MHz tùy thuộc bộ vi xử lý đã được sử dụng và phạm vi sử dụng.

Xung đồng hồ nay quyết định đến tốc độ hoạt động của PLC và cung cấp sự đồng bộ hóa cho mọi phần tử trong hệ thống.

Một CPU bao gồm 3 thành phần riêng biệt như sau:

- Bộ điều khiển (CU\_Control Unit): gồm khối kiểm soát lệnh và ngăn xếp cá nhiệm vụ lấy lệnh ra từ bộ nhớ và xác định điều khiển.
- Bộ luận lý và số học (ALU): Để thực hiện các phép toán số học và logic như: AND, OR, NOT, ...
- Bộ nhớ tốc độ cao, kích thước nhỏ để lưu các kết quả tạm thời và các thông tin điều khiển.

Để thi hành một lệnh nào đó CPU phải làm một chuỗi tuần tự như sau:

1. Lấy lệnh kể từ bộ nhớ vào thanh ghi
2. Thay đổi bộ đếm chương trình để chỉ đến lệnh kế tiếp
3. Xác định kiểu lệnh vừa lấy ra
4. Xác định dữ liệu mà lệnh yêu cầu và xác định vị trí dữ liệu trong bộ nhớ.
5. Nếu lệnh cần dữ liệu trong bộ nhớ, thì nạp nó vào thanh ghi của CPU
6. Thực hiện lệnh
7. Lưu trữ kết quả ở nơi thích hợp
8. Trở về bước 1 để thực hiện lệnh kế tiếp.

➤ **BỘ NHỚ:**

Bao gồm bộ nhớ chứa chương trình, bộ nhớ dữ liệu,... Đơn vị nhỏ nhất của bộ nhớ là BIT có giá trị 1 (hoặc 0). Nhiều BIT hợp thành theo hàng và cột tạo thành một khối bộ nhớ. Nội dung bộ nhớ có thể được đọc ra hoặc ghi vào. Mỗi BIT có một địa chỉ riêng để bộ nhớ quản lý. Bộ nhớ có dung lượng đo bằng số lượng Byte (Kbyte, Mbyte ...)

Có 2 loại bộ nhớ như sau:

- Bộ nhớ RAM (Random Access Memory):

RAM là bộ nhớ chính trong mọi máy tính, kể cả PLC. Bộ nhớ RAM có lợi điểm là dung lượng lớn nhưng giá rẻ. RAM là loại bộ nhớ có thể đọc /ghi chương trình một cách dễ dàng. Tuy nhiên dữ liệu trong RAM sẽ bị xóa sạch khi có sự cố về điện. Vì vậy muốn lưu trữ chương trình trong bộ nhớ RAM thì người ta thường dùng phương pháp nuôi bộ nhớ RAM bằng một nguồn pin. Nếu cần lưu trữ trong thời gian dài thì ta dùng loại pin có tuổi thọ dài và tốt nhất nên thay pin vào những thời gian thích hợp.

- Bộ nhớ ROM (Read Only Memory):

ROM là bộ nhớ chỉ đọc. Bộ nhớ có đặc tính trái ngược với bộ nhớ RAM là rất khó xóa, nên khi có sự cố về điện thì nội dung chương trình vẫn còn trong bộ nhớ. Nhưng hiện nay đối với bộ ROM người ta có thể thay đổi nội dung của nó. Tùy thuộc vào các tạo nội dung, cách xóa nội dung, cách nạp nội dung mới vào nó mà người ta có các loại bộ nhớ ROM khác nhau như : PROM, EPROM, RROM và EEPROM.

Điển hình ở đây ta xét hai loại bộ nhớ ROM được sử dụng rộng rãi trong các PLC là: EPROM và EEPROM.

**EPROM** (Erasable Programmable Read-Only Memory): Bộ nhớ ROM có thể xóa nội dung chương trình. Bộ nhớ này có thể xóa bằng tia cực tím, sau khi nội dung cũ bị xóa người ta dùng một thiết bị đặc biệt để ghi nội dung chương trình mới vào trong ROM. Loại này rất phức tạp vì phải dùng các thiết bị đắt tiền.

**EEPROM** (Electrically Erasable Programmable Read-Only Memory ): Loại này cũng giống như bộ nhớ EPROM nhưng phương thức xóa nội dung đơn giản hơn, tức là xóa bằng điện và nạp nội dung mới cho nó cũng đơn giản. Ngoài 2 loại trên trong các PLC người ta còn dùng FLASH EROM.

Đối với những bộ điều khiển logic theo chương trình thuộc loại lớn có thể có nhiều CPU nhằm tăng tốc độ xử lý.

### ➤ TÍN HIỆU NGÕ VÀO (INPUT):

Các cảm biến (Sensor), nút nhấn (Push Button), Công tắc hành trình (Limits Switch)... được nối với ngõ vào của PLC. Thông thường đầu vào của PLC Master-K có 6 ngõ vào hoặc 8 ngõ vào hoặc 16 ngõ vào hoặc hơn nữa tùy thuộc vào yêu cầu đòi hỏi của người sử dụng mà chọn cho phù hợp. Đối với những ứng dụng nhỏ thì cần khoảng  $\leq 16$  ngõ vào, ứng dụng trung bình thì cần khoảng 80 ngõ vào, ứng dụng cỡ lớn thì cần dùng khoảng 256 ngõ vào.

Đối với PLC người ta dùng các cuộn dây relay cho ngõ vào. Điện áp hoạt động đưa vào cuộn dây này thường vào khoảng 24VDC với dòng vào vài mA (6mA), rất bé so với dòng tiêu thụ qua cuộn dây trong relay thực tế. Cũng có PLC hoạt động với điện áp 220 VAC. Mặc dù điện áp cao như vậy nhưng rất an toàn cho các mạch điện tử của PLC vì sử dụng phương pháp cách ly bằng các linh kiện Optocoupler.

Theo tiêu chuẩn công nghiệp với điện áp 24 VDC, người ta qui định như sau:

- Điện áp từ **0 ÷ 5 VDC** thể hiện **logic 0** ở ngõ vào
- Điện áp từ **11 + 30 VDC** thể hiện **logic 1** ở ngõ vào

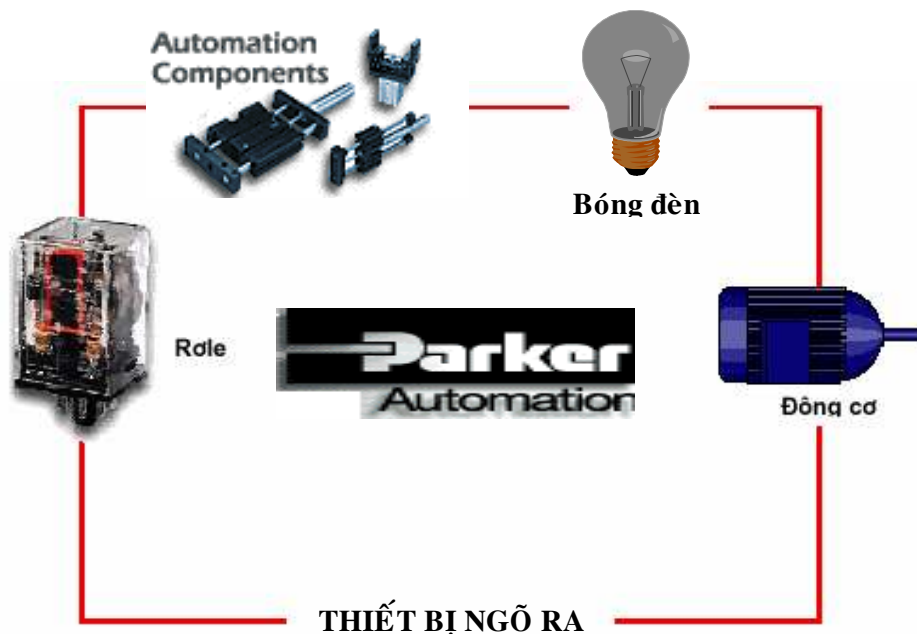


### ➤ TÍN HIỆU NGÕ RA (OUTPUT):

Trong PLC tín hiệu ngõ ra dùng để điều khiển các cơ cấu chấp hành. Đối với những ứng dụng nhỏ thì chỉ cần sử dụng  $\leq 16$  ngõ ra, những ứng dụng lớn hơn có thể dùng tới 64 hoặc tới 256 ngõ ra.

Cũng giống như tín hiệu ngõ vào thì các ngõ ra là các tiếp điểm của relay, khả năng chịu tải lớn 220V/1A. Nếu muốn khống chế tải công suất lớn thì thông qua các Contactor, Solid State Relay (SSR) ...

Ngoài ra còn có PLC với ngõ ra là tín hiệu điện: Logic 0 ứng với điện áp từ 0 + 0.8V và Logic 1 ứng với điện áp từ 12 + 28 V với dòng ra có khi lên tới 300mA. Dãi điện áp cấp nguồn từ 12V +28V.



### ➤ THỜI GIAN ĐÁP ỨNG CỦA NGÕ VÀO/RA:

Ngõ vào của PLC tiếp nhận các tín hiệu từ các sensor, từ sự đóng ngắt các tiếp điểm của nút nhấn Start, Stop, Reset hay từ công tắc hành trình, ...Để chống rung ở ngõ vào, PLC có một mạch lọc ở ngõ vào, do đó làm chậm thời gian đáp ứng của nó (từ  $100\mu\text{S}$  + 25.5ms). Nếu cần PLC cũng có những ngõ vào chuyên dùng với tốc độ đáp ứng nhanh.

Đáp ứng ở ngõ ra đủ nhanh (cỡ ms), đạt yêu cầu đối với ứng dụng điều khiển các cơ cấu chấp hành trong thiết bị tự động hóa công nghiệp.

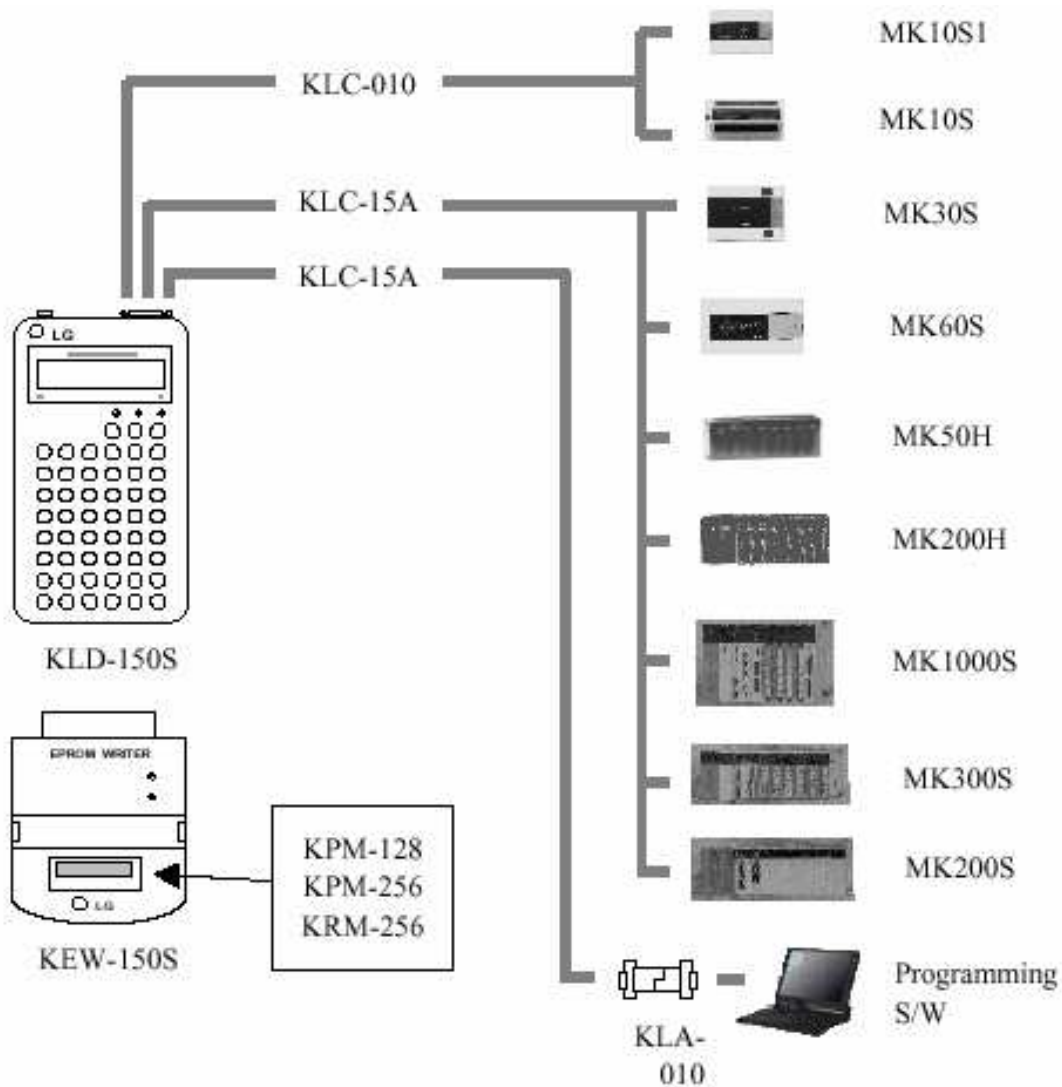


## HƯỚNG DẪN LẬP TRÌNH CĂN BẢN PLC MASTER-K

Chương III: CÁCH THẢO CHƯƠNG TRÌNH VỚI PHẦN MỀM  
KGL-WIN Ver3.65

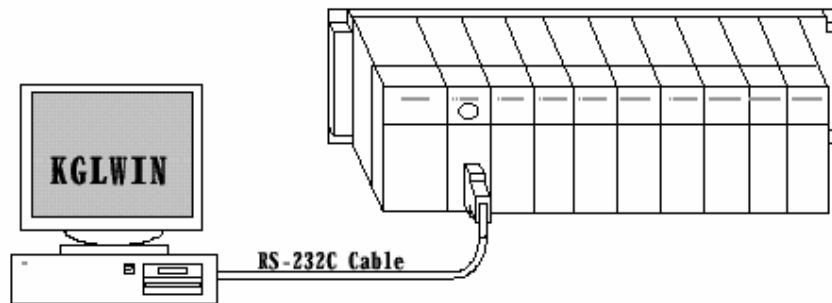


Thông thường để lập trình cho PLC người ta sử dụng một thiết bị chuyên dùng gọi là **Hand-held Program Loader** gồm có một bàn phím và màn hình chỉ thị để hiển thị các số liệu cần thiết như các lệnh, các thông số. Sau khi chương trình đã được lưu trữ trong bộ nhớ của PLC thì có thể thực hiện được.



Song song với Hand-held Program Loader chúng ta có thể sử dụng phần mềm **KGL-WIN V3.2** được cài đặt trong máy tính cá nhân PC chạy trên hệ điều hành Windows XP để có thể sửa chữa, viết lại hoặc lưu trữ các chương trình gốc.

Thông thường việc truyền các chương trình điều khiển ở dạng ngôn ngữ máy sang các bộ PLC thông qua cổng truyền nối tiếp **RS-232C**.



Thông thường chương trình được viết , sửa đổi , thử nghiệm nhiều lần trên PLC thì mới thể hiện được ý đồ logic điều khiển mà ta mong đợi.

Chúng tôi xin giới thiệu cách soạn thảo chương trình trên phần mềm **KGL-WIN Ver3.2** của **LSIS**.

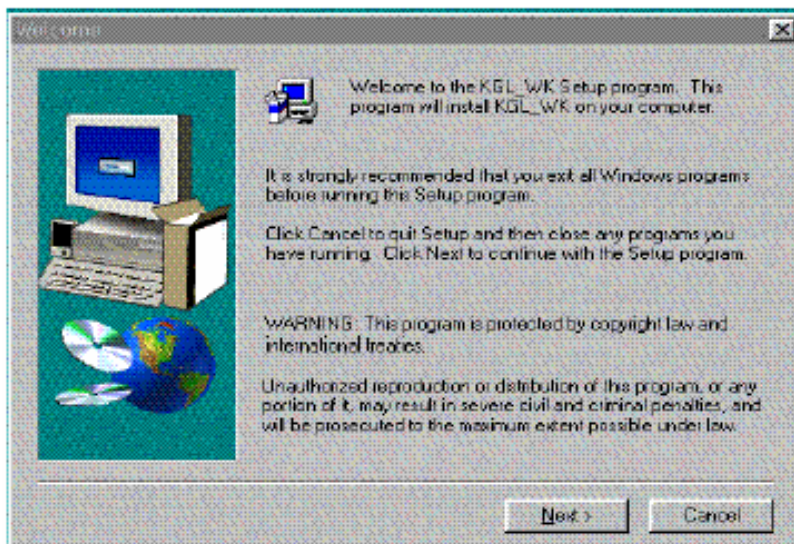
- **YÊU CẦU CẤU HÌNH MÁY TÍNH CHO PHẦN MỀM:**

- Máy tính tối thiểu phải từ 80486DX trở lên, RAM phải từ 8Mb RAM trở lên.
- Phải có cổng nối tiếp (serial port): một hay hai cổng
- Dung lượng ổ đĩa cứng: Phải còn trống 20Mb
- Phải có ổ đĩa mềm
- Chuột và máy in
- Chạy trên hệ điều hành Windows 9.x hay Windows XP

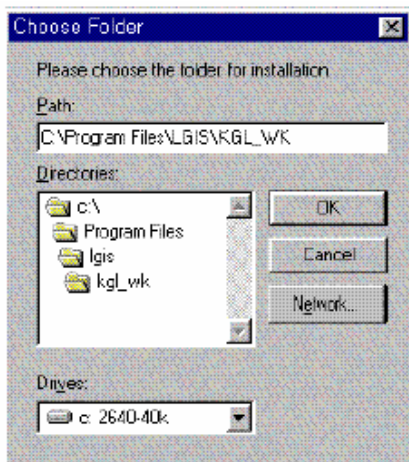
- **CÀI ĐẶT PHẦN MỀM:**

Đưa đĩa CDRom có chứa chương trình vào trong ổ đĩa:

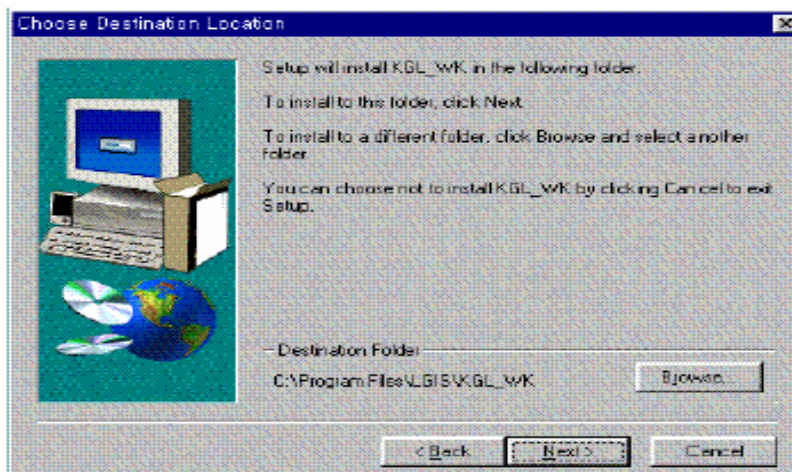
- Double-click vào tập tin **SETUP.EXE** trong đĩa.
- Chương trình Setup Wizard của chương trình sẽ hướng dẫn bạn cài đặt các bước tiếp theo.
- Nhấn nút NEXT để bắt đầu cài đặt. Bạn có thể nhấn nút Cancel nếu bạn không muốn cài đặt KGL for Windows tại thời điểm này.

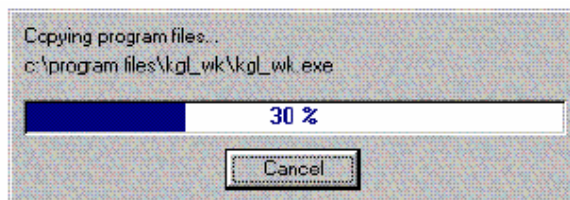



- Chọn thư mục mà phần mềm sẽ được cài đặt vào. Bạn có thể thay đổi bằng cách nhấn nút Browse. Nếu bạn không thay đổi thì đường dẫn mặc định sẽ là: **C:\Program Files\LSIS\KGL\_WE**.
- Nếu bạn nhấn nút Browse sẽ xuất hiện hộp thoại sau:



- Khi chương trình cài đặt sẽ được thể hiện như sau:

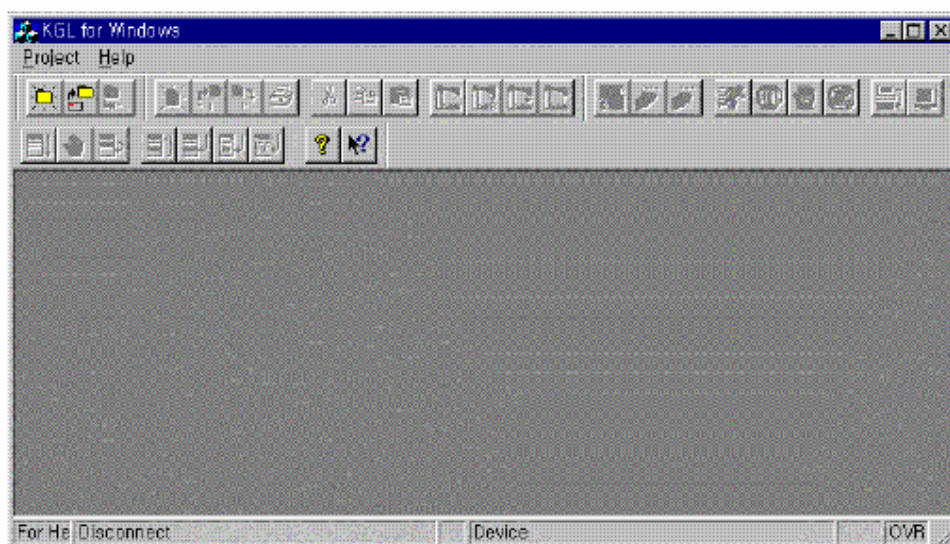




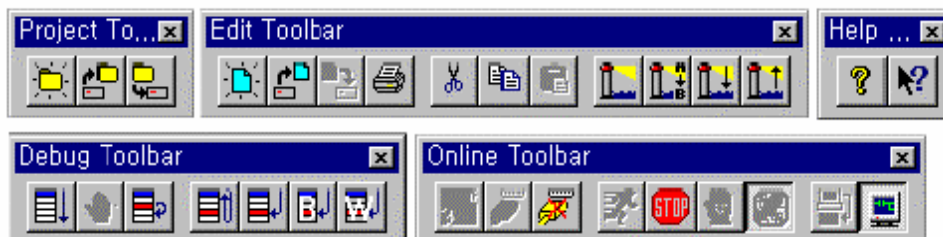
- Sau khi cài đặt xong. Muốn chạy chương trình double-click vào biểu tượng: (  ).


**• TẠO RA MỘT PROJECT MỚI:**

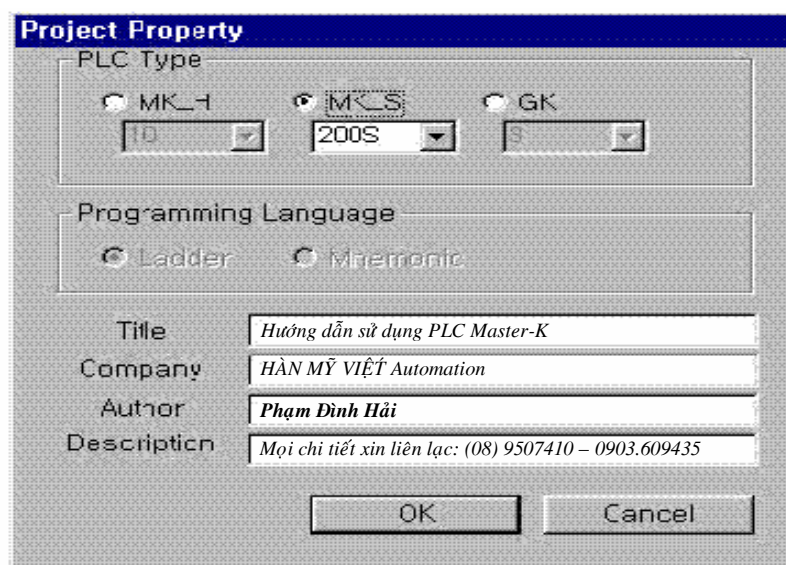
Khi khởi động phần mềm xong, một cửa sổ mới sẽ xuất hiện :



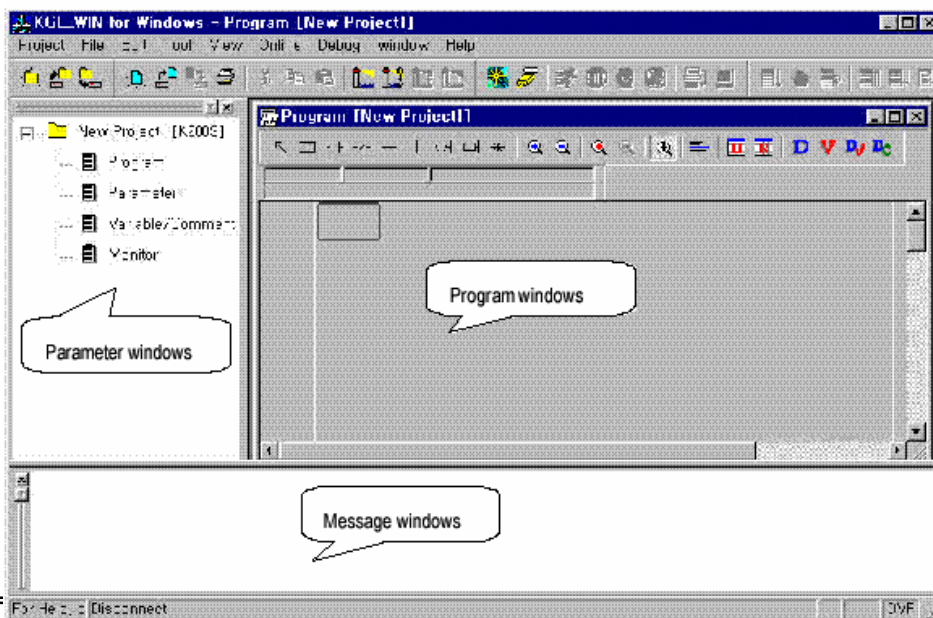
- Các thanh công cụ hỗ trợ của phần mềm KGL for Windows Ver 3.2



- Tạo một Project mới: Chọn **Project-New Project ...**
- (hay nhấn vào biểu tượng  )
- Chọn **Blank Project** trong hộp thoại rồi tiếp tục nhấn nút OK.
- Trong hộp thoại tiếp theo bạn phải khai báo các thông tin sau: Loại PLC, kiểu viết chương trình Ladder hay Mnemonic, tựa đề của chương trình, tên Công ty, tên Tác giả ...

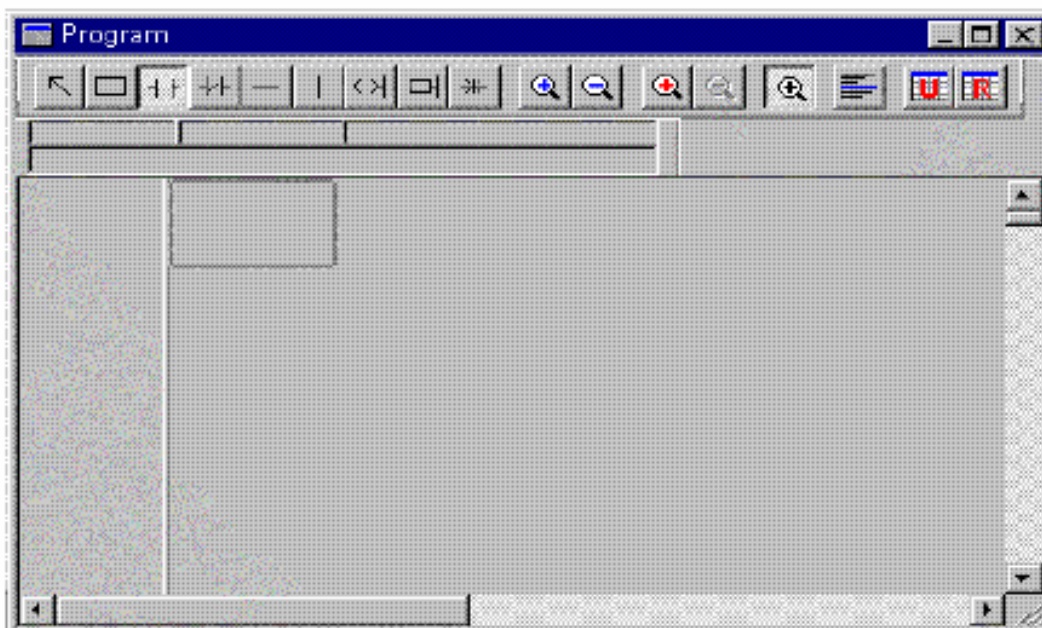


- Sau khi báo xong nhấn vào nút OK. Khi đó các cửa sổ **Parameter**, **Message**, **Program** sẽ được tạo ra.

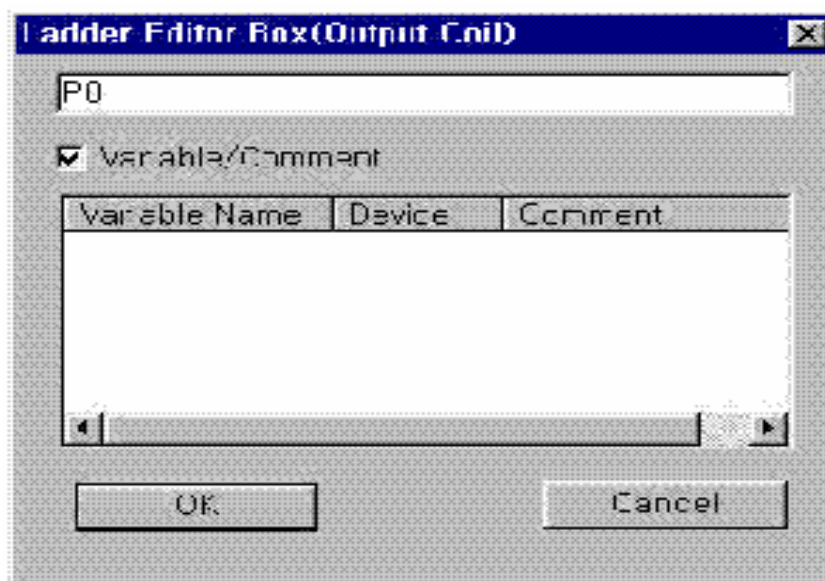


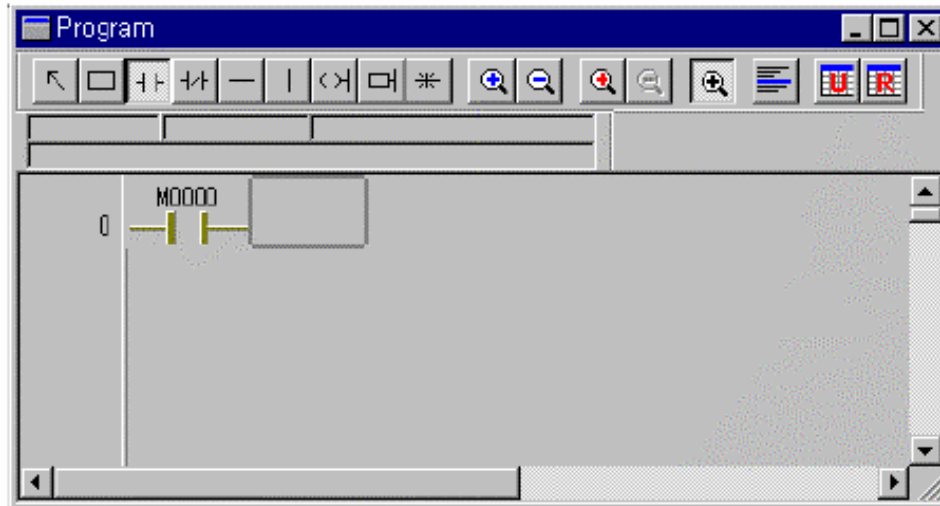
### CÁCH SOẠN THẢO CHƯƠNG TRÌNH:


Trong cửa sổ **Program** có thanh công cụ (Tool Bar) bao gồm các nút nhấn công tắc thường mở (Normally Open), công tắc thường đóng (Normally Close), thanh dọc, thanh ngang, output coil, các hàm chức năng ... giúp ta soạn thảo một cách dễ dàng .

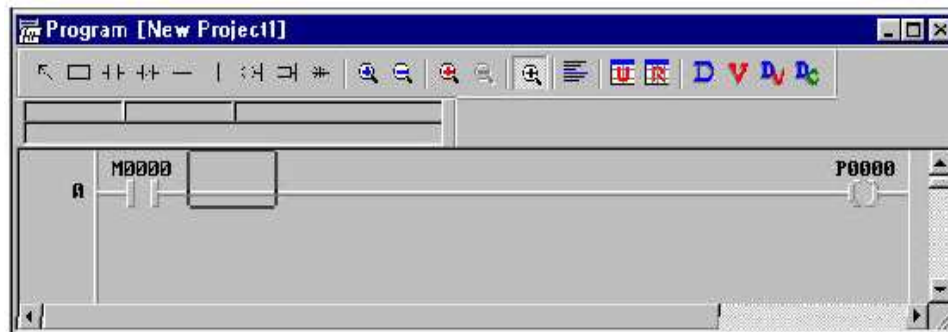



- Sau khi chọn công tắc thường mở (  $\text{---|---}$  ), di chuyển con trỏ đến vị trí cần chèn công tắc.
- Click chuột trái hay nhấn phím Enter, khi đó sẽ xuất hiện một hộp thoại cho bạn nhập địa chỉ input vào.

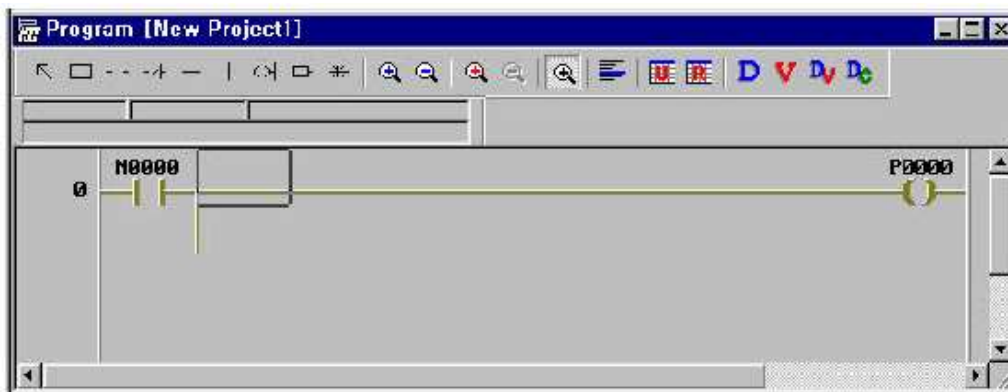






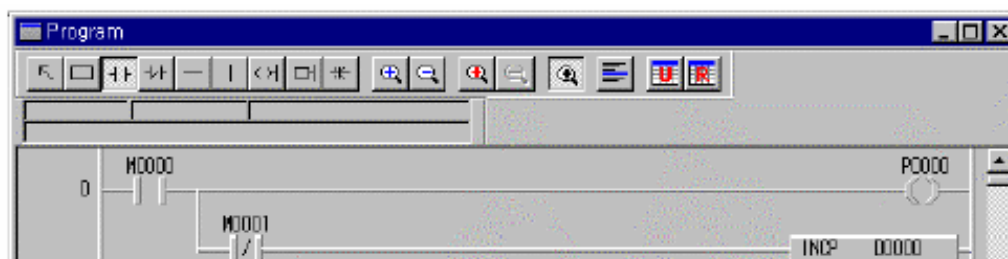
- Chọn tiếp Output Coil (  ) để điều khiển cơ cấu chấp hành, lấy ví dụ như ta sẽ điều khiển ngõ ra tại địa chỉ P0040. Sau khi chọn xong nhấn phím chuột hay đánh Enter một hộp thoại khác sẽ xuất hiện yêu cầu bạn đánh tên địa chỉ Output (P0040).




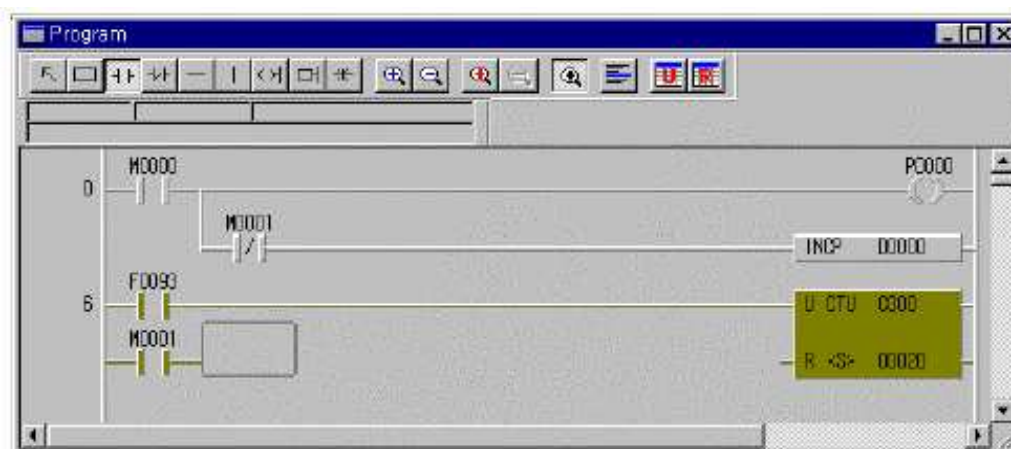
- Bây giờ chúng ta thêm một nhánh khác cho ngõ ra. Bạn chọn (  ) sau đó click chuột tại vị trí cần rẽ nhánh.



- Tại nhánh rẽ ở đầu vào ta thêm một công tắc thường đóng (  ). Click chuột vào biểu tượng rên thanh công cụ sau đó click vào vị trí cần đặt và đánh tên địa chỉ vào.
- Chọn vào biểu tượng (  ) để chọn một lệnh nào đó trong PLC. Ví dụ chúng ta sẽ thực hiện lệnh INCP.

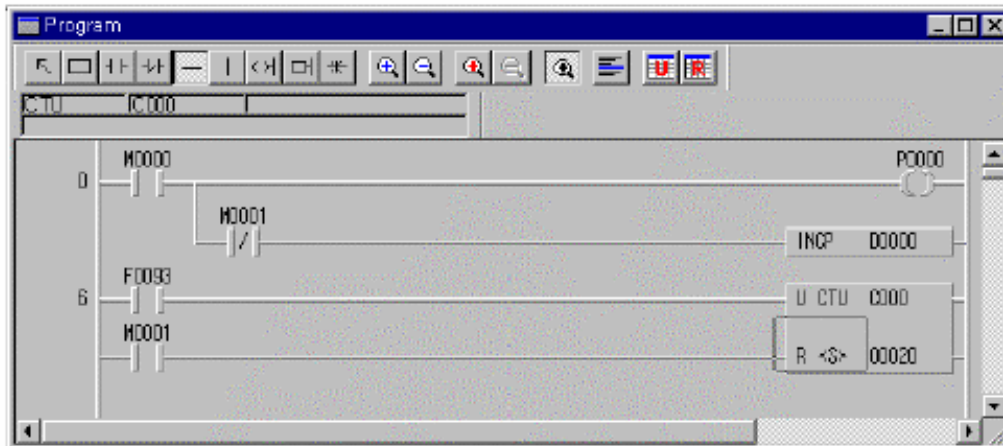



- Bây giờ chúng ta thực hiện viết một đoạn chương trình thực hiện lệnh đếm (Counter) trong PLC.
- Chọn công tắc thường mở trên thanh công cụ và di chuyển con trỏ đến vị trí đầu của nhánh tiếp theo, click hay đánh Enter, chúng ta đánh vào F0093 ( Cờ đếm xung clock – 1giây).
- Chọn (  ) trên thanh công cụ và đánh CTU C000 20 ( thực hiện lệnh đếm lên đến giá trị 20 ). Để Reset cho Counter ta thêm một công tắc thường mở.

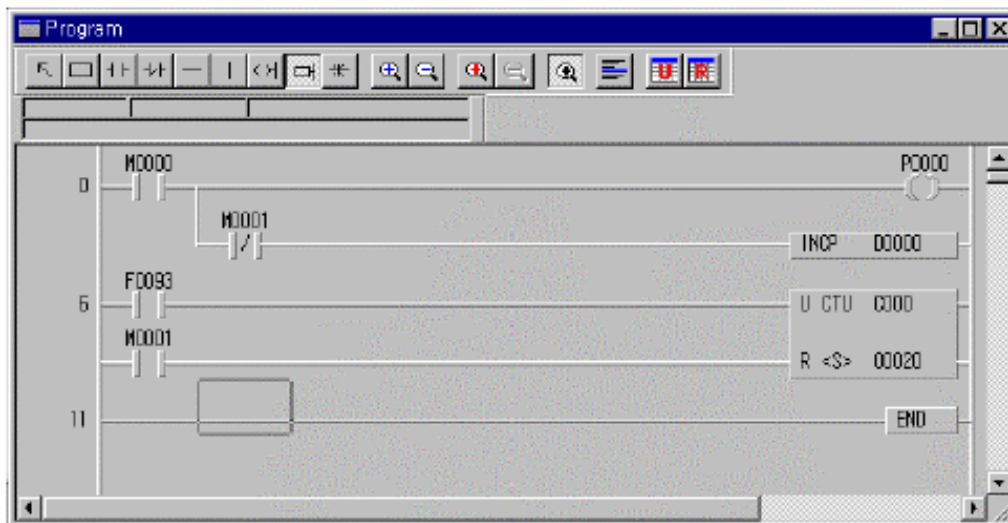


- Để hoàn thành câu lệnh Counter ta chọn biểu tượng đường ngang ( — ) để nối tín hiệu Reset vào Counter.






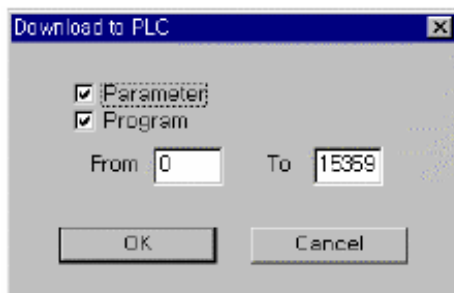
- Kết thúc một chương trình phải có lệnh END. Chọn biểu tượng (  ) sau đó đưa con trỏ đến dòng cuối cùng của chương trình và click chuột hay đánh Enter khi đó một hộp thoại sẽ xuất hiện và đánh vào chữ END vào.



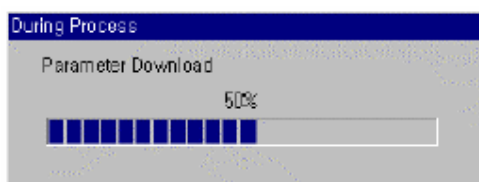
#### • KẾT NỐI PC ĐẾN PLC:


Sau khi viết chương trình xong, để đưa chương trình đã soạn thảo từ máy tính (PC) đến PLC sẽ được thông qua sợi Cable RS-232C.

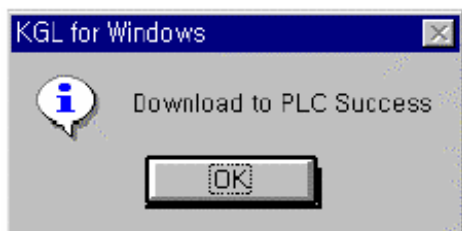
Trên menu ta chọn **Online-Connect** để kết nối PC và PLC. Sau khi việc kết nối đã hoàn thành ta chọn **Online- Download** (  ) để tải chương trình xuống. Khi đó trên màn hình sẽ xuất hiện hộp thoại sau:




Click vào nút OK để tiếp tục. Khi đó chương trình và các thông số sẽ được tải xuống đến PLC.

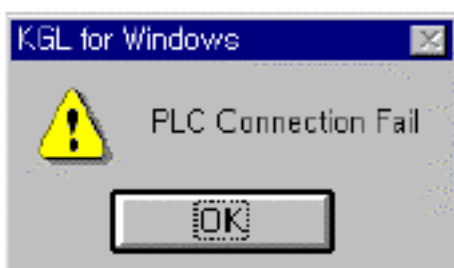


Khi đang Download thì PLC phải ở chế độ Stop (  ). Và khi Download chương trình sẽ thông báo:

































Để chạy chương trình ta nhấn vào nút (  )

Trong trường hợp Cable PLC bị lỗi hay lựa chọn PLC không đúng thì chương trình sẽ thông báo lỗi như sau:



**Bảng chú thích các nút nhấn của thanh công cụ (Tool Bar):**

Tools	Commands	Tools	Commands
	New Project		Connect
	Open Project		Disconnect
	Save Project		Download
	New File		Monitoring Mode
	Open File		Run
	Save File		Stop
	Print		Pause
	Cut		Debug
	Copy		Go
	Paste		Debug Stop
	Find		Trace
	Replace		Break Scan
	Forward		Break Step
	Backward		Break Bit
	Connect+Download+Run+Monitor Start		Break Word

# HƯỚNG DẪN LẬP TRÌNH CĂN BẢN PLC MASTER-K

## Chương IV: LẬP TRÌNH BẰNG SƠ ĐỒ BẬC THANG LADDER DIAGRAM

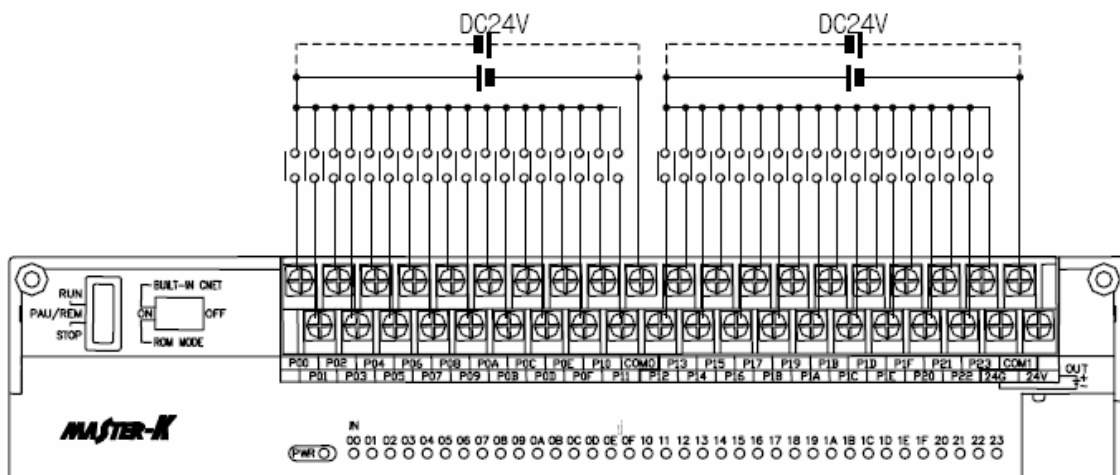


### I. SƠ ĐỒ ĐẦU DÂY ĐẦU VÀO (INPUT) VÀ ĐẦU RA (OUTPUT):

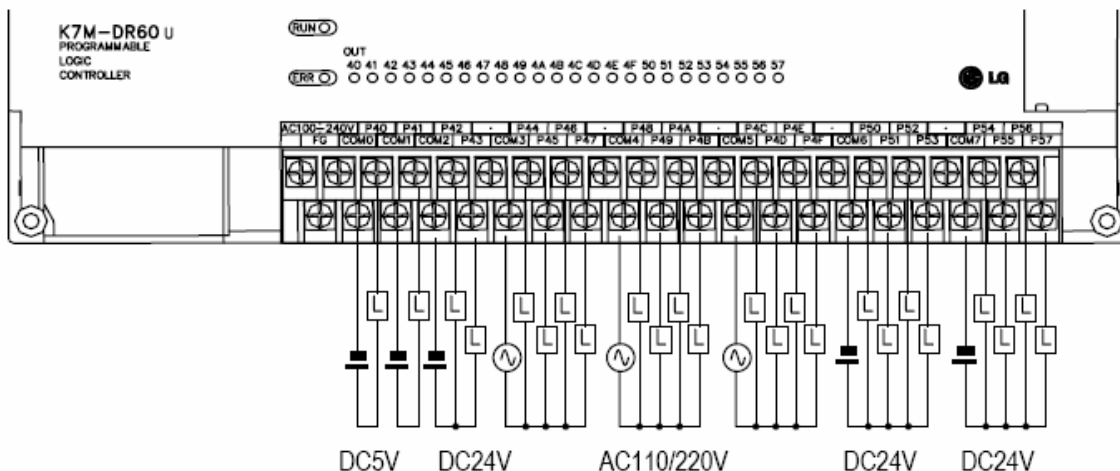
Các nối đầu vào có thể có 3 dạng :

- Đầu vào là tiếp điểm
- Đầu vào là transistor kiểu NPN
- Đầu vào là transistor kiểu PNP

Sơ đồ đấu dây đầu vào:



Sơ đồ đấu dây đầu ra:



## II. CÁC ĐỊA CHỈ VÙNG NHỚ TRONG PLC MASTER-K:

Tất cả các đầu vào ra cũng như các bộ nhớ lưu trữ khác trên PLC khi sử dụng trong chương trình đều thông qua các địa chỉ bộ nhớ tương ứng. Các địa chỉ bộ nhớ được tổ chức thành các nhóm gồm 16 bit gọi là Word. Mỗi bit có giá trị 0 hoặc 1 và được đánh số từ 00 đến 0F (15) từ phải sang trái.

- **Vùng nhớ P (Input/Output ):**
  - Biểu diễn các ngõ vào ra vật lý trên bộ điều khiển
  - Độ lớn vùng nhớ: P000 ÷ P63F
- **Vùng nhớ M (Auxiliary Relay ):**
  - Các tiếp điểm phụ trợ của bộ điều khiển
  - Độ lớn vùng nhớ: M000 ÷ M191F
- **Vùng nhớ K (Keep Relay ):**
  - Các tiếp điểm giữ
  - Độ lớn vùng nhớ: K000 ÷ K31F
- **Vùng nhớ L (Link Relay ):**
  - Các tiếp điểm khi kết nối mạng
  - Độ lớn vùng nhớ: L000 ÷ L63F
- **Vùng nhớ F (Special Relay ):**
  - Các cờ đặc biệt (Xem thêm chi tiết bảng phụ lục )
  - Độ lớn vùng nhớ: F000 ÷ F63F
  - Các cờ đặc biệt thường dùng:
    - ❖ **F000:** ON khi PLC đang hoạt động
    - ❖ **F010:** Cờ tiếp điểm luôn luôn ON
    - ❖ **F011:** Cờ tiếp điểm luôn luôn OFF
    - ❖ **F012:** ON cho lần quét đầu tiên, sau đó giữ ở trạng thái OFF
    - ❖ **F013 :** OFF cho lần quét đầu tiên, sau đó giữ ở trạng thái ON
    - ❖ **F030:** ON khi thực hiện các lệnh CALL và JMP bị lỗi.
    - ❖ **F031:** Báo lỗi I/O, ON khi các cờ báo của các I/O module là ON
    - ❖ **F033:** ON khi pin có lỗi.
    - ❖ **F040:** Báo lỗi I/O cho module số 0
    - ❖ **F041:** Báo lỗi I/O cho module số 1
    - ❖ **F042:** Báo lỗi I/O cho module số 2
    - ❖ **F043:** Báo lỗi I/O cho module số 3

- ❖ **F044:** Báo lỗi I/O cho module số 4
- ❖ **F045:** Báo lỗi I/O cho module số 5
- ❖ **F090:** Cờ xung nhịp 0.02 giây
- ❖ **F091:** Cờ xung nhịp 0.1 giây
- ❖ **F092:** Cờ xung nhịp 0.2 giây
- ❖ **F093:** Cờ xung nhịp 1 giây
- ❖ **F094:** Cờ xung nhịp 2 giây
- ❖ **F095:** Cờ xung nhịp 10 giây
- ❖ **F096:** Cờ xung nhịp 20 giây
- ❖ **F097:** Cờ xung nhịp 1 phút
- ❖ **F120 :** ON khi so sánh là nhỏ hơn (<)
- ❖ **F121:** ON khi so sánh là nhỏ hơn hoặc bằng ( $\leq$ )
- ❖ **F122:** ON khi so sánh là bằng (=)
- ❖ **F123:** ON khi so sánh là (>)
- ❖ **F124:** ON khi so sánh là ( $\geq$ )
- **Vùng nhớ S ( Step Control ):**
  - Các tiếp điểm cho chế độ chạy tuần tự
  - Độ lớn vùng nhớ: S00.00 ÷ S99.99
- **Vùng nhớ D (Data Register):**
  - Vùng nhớ dữ liệu cho bộ đếm, bộ định thời, các module analog...
  - Độ lớn vùng nhớ: D0000 ÷ D4999
  - Các vùng nhớ D đặc biệt:

Vùng nhớ	Module mở rộng	G7F-ADHA	G7F-ADHB	G7F-AD2A	G7F-DA2I	G7F-DA2V	G7F-AT2A	G7F-RD2A
<b>D4980</b>	#1	CH0 A/D Value	CH0 A/D Value	CH0 A/D Value	CH0 D/A Value	CH0 D/A Value	CH0 A/TValue	CH0 Temperature
<b>D4981</b>		CH1 A/D Value	CH1 A/D Value	CH1 A/D Value	CH1 D/A Value	CH1 D/A Value	CH1 A/TValue	CH1 Temperature
<b>D4982</b>		CH0 D/A Value	CH0 D/A Value	CH2 A/D Value	CH2 D/A Value	CH2 D/A Value	CH2 A/TValue	CH2 Temperature
<b>D4983</b>		-	CH1 D/A Value	CH3 A/D Value	CH3 D/A Value	CH3 D/A Value	CH3 A/TValue	CH3 Temperature

<b>D4984</b>	#2	CH0 A/D Value	CH0 A/D Value	CH0 A/D Value	CH0 D/A Value	CH0 D/A Value	CH0 A/TValue	CH0 Temperature
<b>D4985</b>		CH1 A/D Value	CH1 A/D Value	CH1 A/D Value	CH1 D/A Value	CH1 D/A Value	CH1 A/TValue	CH1 Temperature
<b>D4986</b>		CH0 D/A Value	CH0 D/A Value	CH2 A/D Value	CH2 D/A Value	CH2 D/A Value	CH2 A/TValue	CH2 Temperature
<b>D4987</b>		-	CH1 D/A Value	CH3 A/D Value	CH3 D/A Value	CH3 D/A Value	CH3 A/TValue	CH3 Temperature
<b>D4988</b>	#3	CH0 A/D Value	CH0 A/D Value	CH0 A/D Value	CH0 D/A Value	CH0 D/A Value	CH0 A/TValue	CH0 Temperature
<b>D4989</b>		CH1 A/D Value	CH1 A/D Value	CH1 A/D Value	CH1 D/A Value	CH1 D/A Value	CH1 A/TValue	CH1 Temperature
<b>D4990</b>		CH0 D/A Value	CH0 D/A Value	CH2 A/D Value	CH2 D/A Value	CH2 D/A Value	CH2 A/TValue	CH2 Temperature
<b>D4991</b>		-	CH1 D/A Value	CH3 A/D Value	CH3 D/A Value	CH3 D/A Value	CH3 A/TValue	CH3 Temperature

○ **Vùng nhớ T (Timer):**

- Vùng nhớ cho các bộ định thì và được chia làm các loại: 100ms, 10ms, 1ms và có thể điều chỉnh lại vùng nhớ trong parameter.
- Độ lớn vùng nhớ 100ms: T000 ÷ T191
- Độ lớn vùng nhớ 10ms : T192 ÷ T250
- Độ lớn vùng nhớ 1ms : T251 ÷ T255

○ **Vùng nhớ C (Counter):**

- Vùng nhớ cho các bộ đếm
- Độ lớn vùng nhớ : C000 ÷ C255

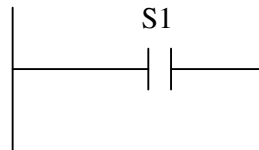


## CÁC LỆNH, PHÉP TOÁN CƠ BẢN :

### 1. Lệnh LOAD:

▫ Cú pháp:

- Mnemonic Code: LOAD <địa chỉ>
- Ladder:



- Các vùng địa chỉ có thể dùng với lệnh LOAD:

S1: M, P, K, L, F, T, C, S

▫ Chức năng: Lệnh LOAD dùng để đặt điều kiện ngõ vào thường mở cho một mạch Logic .

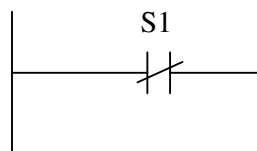
▫ Ví dụ:



### 2. Lệnh LOAD NOT:

▫ Cú pháp:

- Mnemonic Code: LOAD NOT <địa chỉ>
- Ladder:



- Các vùng địa chỉ có thể dùng với lệnh LOAD NOT:

S1: M, P, K, L, F, T, C, S

▫ Chức năng: Lệnh LOAD NOT để đặt điều kiện ngõ vào thường đóng cho một mạch Logic.

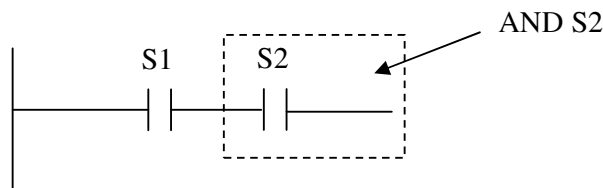
▫ Ví dụ:



### 3. Lệnh AND:

▫ Cú pháp:

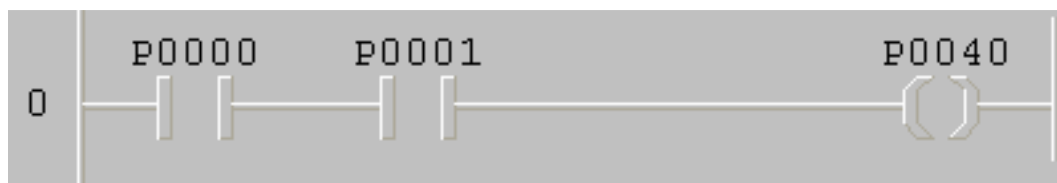
- Mnemonic Code: AND <địa chỉ>
- Ladder:



- Các vùng địa chỉ có thể dùng với lệnh AND :  
S2: M, P, K, L, F, T, C, S

▫ Chức năng: Lệnh AND dùng để nối tiếp một hoặc nhiều điều kiện ngõ vào thường mở cho một mạch Logic .

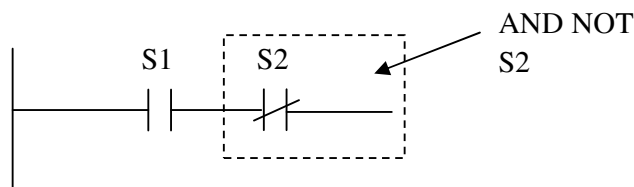
▫ Ví dụ:



### 4. Lệnh AND NOT:

▫ Cú pháp:

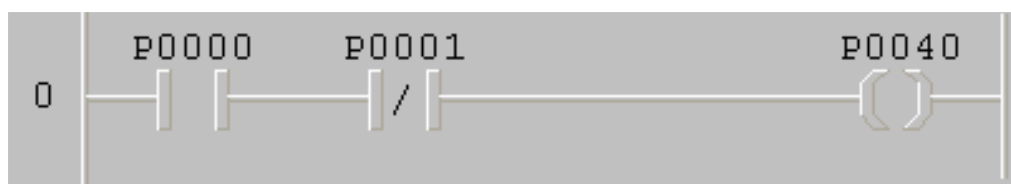
- Mnemonic Code: AND NOT <địa chỉ>
- Ladder:



- Các vùng địa chỉ có thể dùng với lệnh AND NOT:  
S2: M, P, K, L, F, T, C, S

▫ Chức năng: Lệnh AND NOT dùng để nối tiếp một hoặc nhiều điều kiện ngõ vào thường mở cho một mạch Logic .

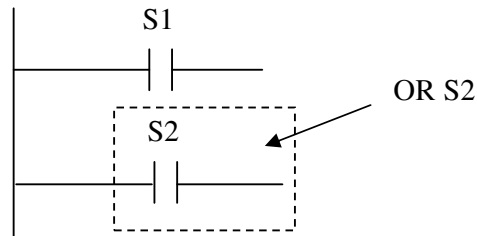
▫ Ví dụ:



### 5. Lệnh OR:

▫ Cú pháp:

- Mnemonic Code: OR <địa chỉ>
- Ladder:

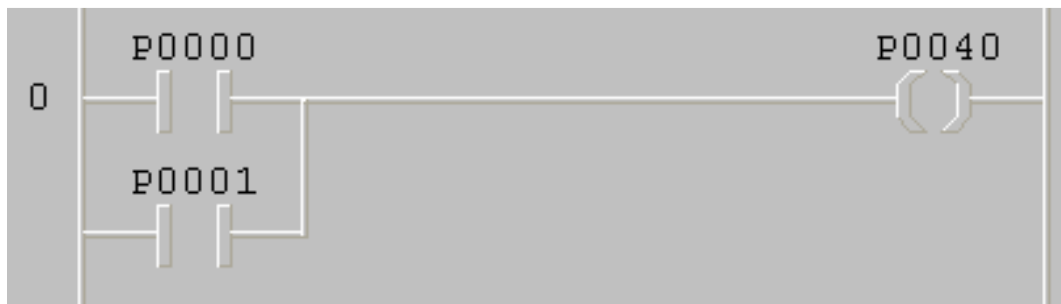


- Các vùng địa chỉ có thể dùng với lệnh OR :

S2: M, P, K, L, F, T, C, S

▫ Chức năng: Lệnh OR dùng để mắc song song một hoặc nhiều điều kiện ngõ vào thường mở cho một mạch Logic .

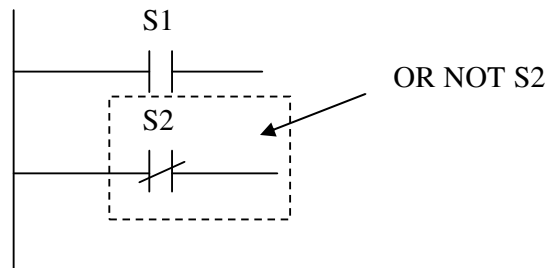
▫ Ví dụ:



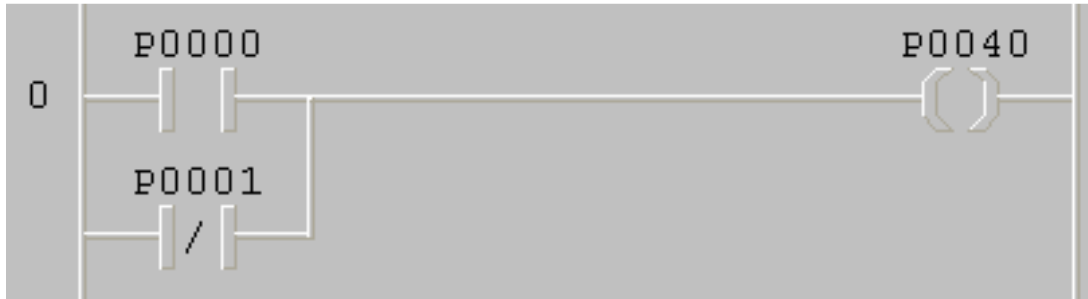
### 6. Lệnh OR NOT:

▫ Cú pháp:

- Mnemonic Code: OR NOT <địa chỉ>
- Ladder:

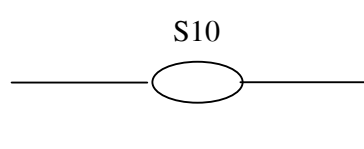


- Các vùng địa chỉ có thể dùng với lệnh OR NOT:  
S2: M, P, K, L, F, T, C, S
- Chức năng: Lệnh OR NOT dùng để mắc song song một hoặc nhiều điều kiện ngõ vào thường mở cho một mạch Logic .
- Ví dụ:

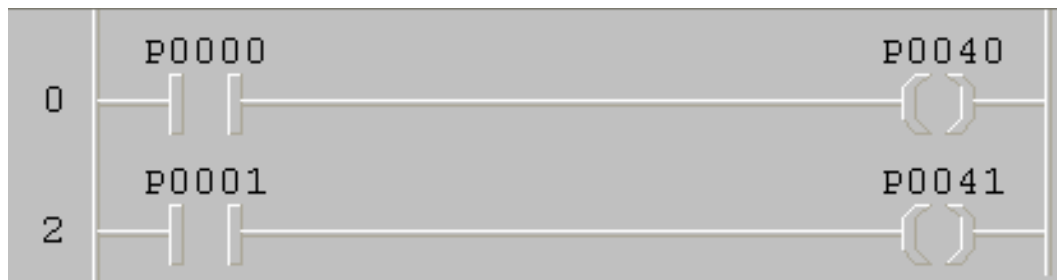


### 7. Lệnh OUT:

- Cú pháp:
  - Mnemonic Code: OUT <địa chỉ>
  - Ladder:

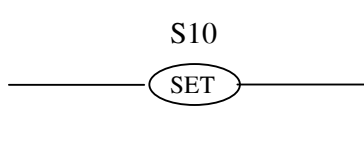


- Các vùng địa chỉ có thể dùng với lệnh OUT:  
S10: M, P, K, L, S
- Chức năng: Lệnh OUT dùng để xuất tín hiệu đến Bit địa chỉ được chỉ định trong lệnh .
- Ví dụ:



### 8. Lệnh SET:

- Cú pháp:
  - Mnemonic Code: SET <địa chỉ>
  - Ladder:



- Các vùng địa chỉ có thể dùng với lệnh SET:  
S10: M, P, K, L, S

▫ Chức năng:

- Lệnh SET dùng để đặt trạng thái của một địa chỉ (bit) lên ON khi điều kiện ngõ vào là ON.
- Khi điều kiện ngõ vào là OFF thì địa chỉ của SET vẫn là ON

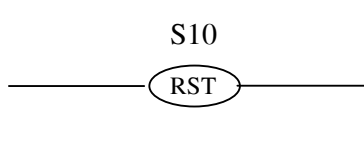
▫ Ví dụ:



### 8. Lệnh RESET:

▫ Cú pháp:

- Mnemonic Code: RST <địa chỉ>
- Ladder:



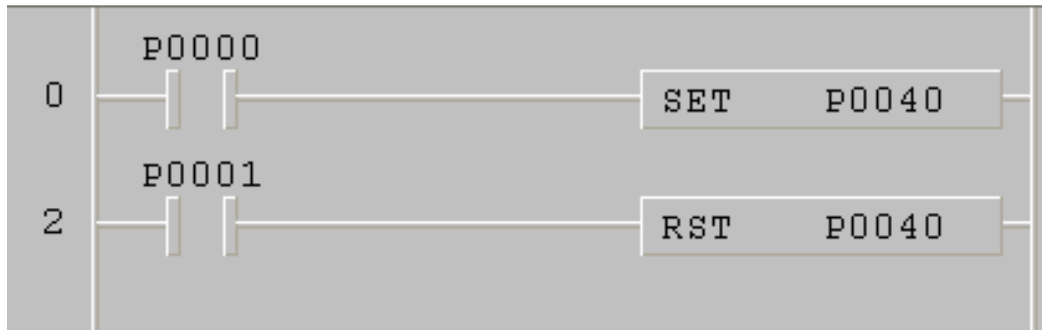
- Các vùng địa chỉ có thể dùng với lệnh RESET:  
S10: M, P, K, L, S

▫ Chức năng:

- Lệnh RESET dùng để đưa trạng thái của một địa chỉ (bit) về OFF (khi đã được SET lên ON), khi điều kiện ngõ vào là ON.
- Khi điều kiện ngõ vào là OFF thì địa chỉ của RESET vẫn là OFF.

▫ Ví dụ:

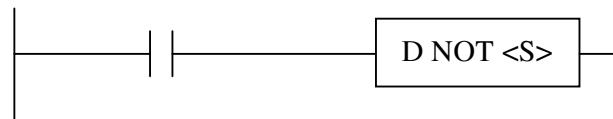
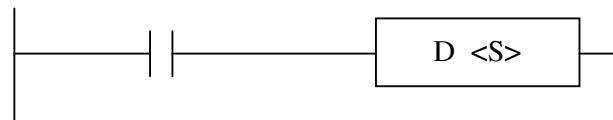




### 8. Lệnh kích xung cạnh lên (D) và xung cạnh xuống (D NOT):

▫ Cú pháp:

- Mnemonic Code: D <địa chỉ>  
D NOT <địa chỉ>
- Ladder:



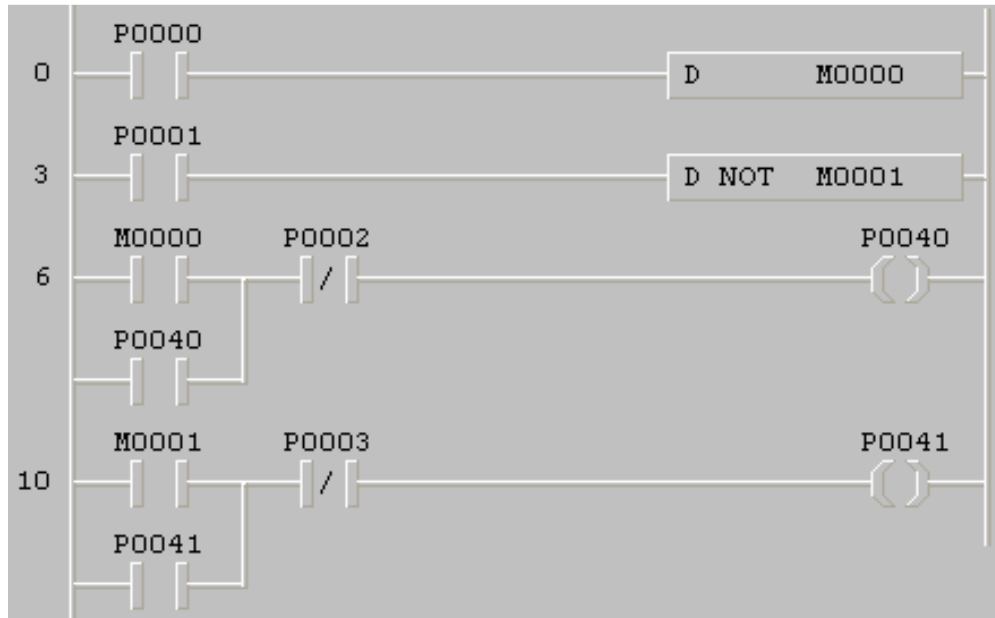
- Các vùng địa chỉ có thể dùng với lệnh D và D NOT:  
S: M, K, L

▫ Chức năng:

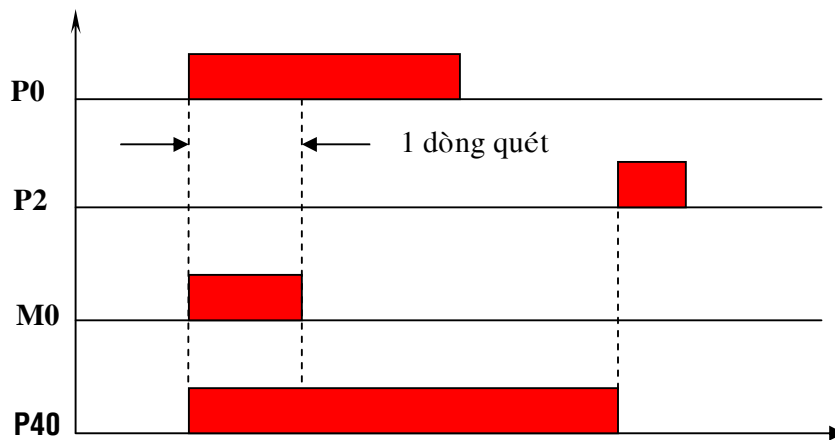
Lệnh D dùng để lấy xung cạnh lên trong thời gian rất ngắn (1 Scan) khi tín hiệu đầu vào chuyển từ OFF sang ON .

Lệnh D NOT dùng để lấy xung cạnh xuống trong thời gian rất ngắn (1 Scan) khi tín hiệu đầu vào chuyển từ OFF sang ON .

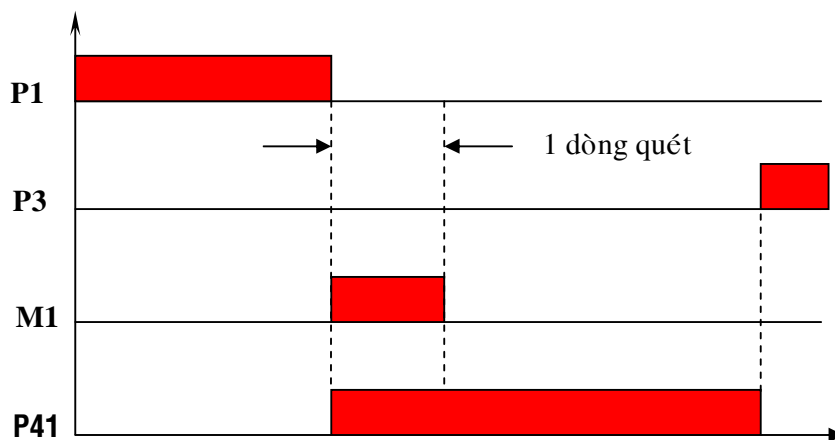
▫ Ví dụ:



Giải đồ lệnh D:



Giải đồ lệnh D NOT:



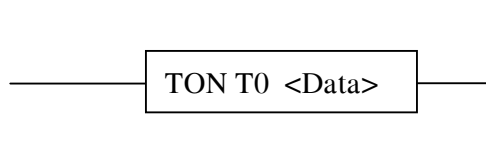


## II. BỘ ĐỊNH THỜI TIMER:

### 1. ON-Delay Timer

▫ Cú pháp:

- Mnemonic Code: TON <địa chỉ> <giá trị cài đặt>
- Ladder:

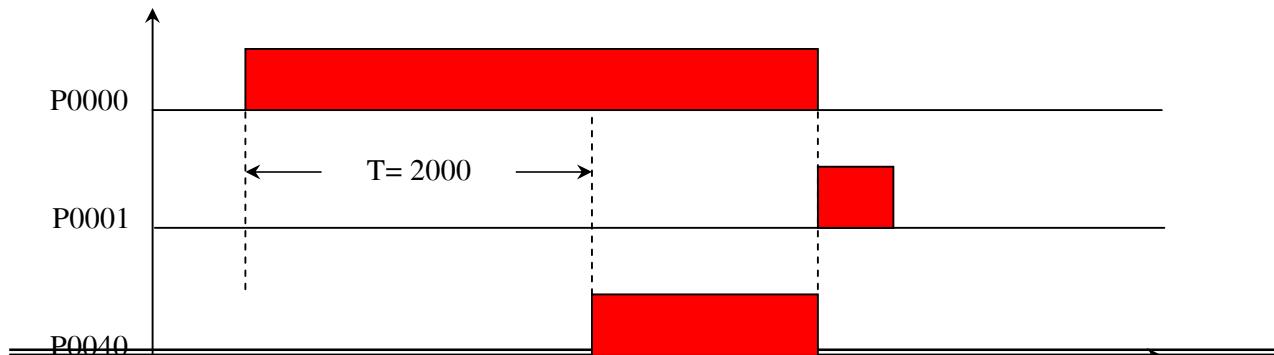
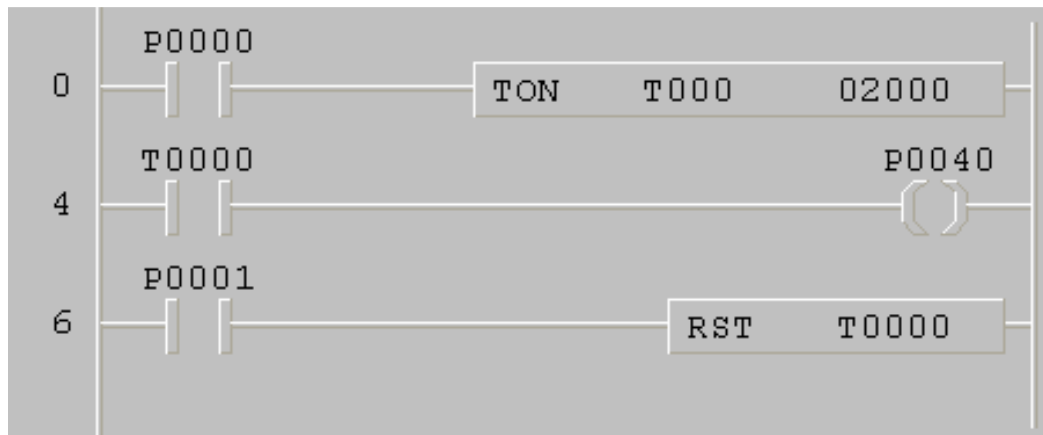


- Các vùng địa chỉ có thể dùng với lệnh TON: T, D, Hằng số.

▫ Chức năng:

- Khi giá hiện hành tăng đến bằng giá trị cài đặt trong khi điều kiện ngõ vào là ON thì tiếp điểm của TON là ON.
- Khi điều kiện ngõ vào là OFF hay khi TON gặp lệnh Reset thì Out Timer sẽ Off và giá trị hiện hành cũng trở về 0.

▫ Ví dụ:



## 2. OFF-Delay Timer

### ▫ Cú pháp:

- Mnemonic Code: TOFF <địa chỉ> <giá trị cài đặt>
- Ladder:

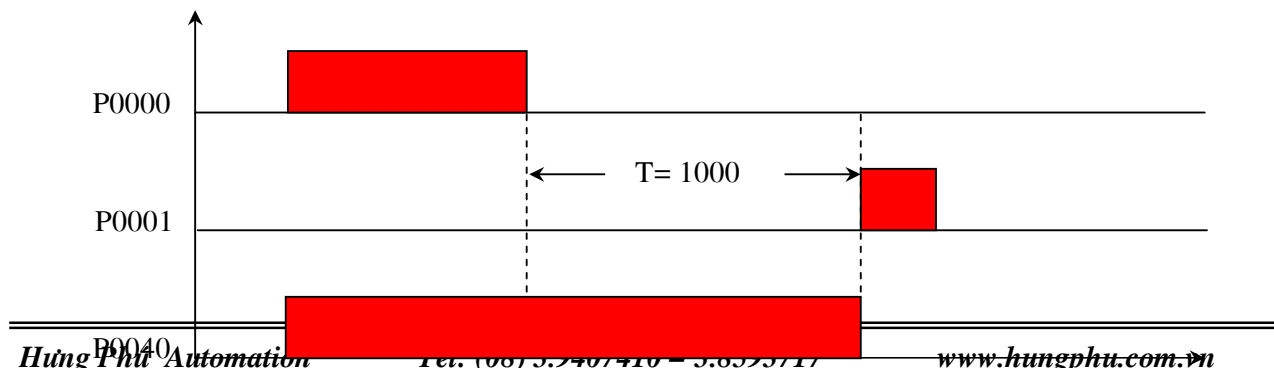
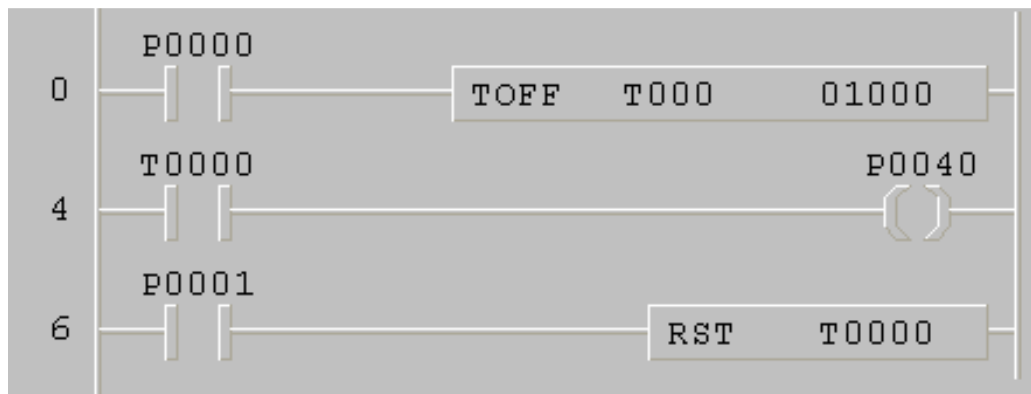


- Các vùng địa chỉ có thể dùng với lệnh TOFF: T, D, Hằng số.

### ▫ Chức năng:

- Khi tín hiệu đầu vào là ON thì Output cũng ON, và khi tín hiệu đầu vào chuyển từ ON sang OFF thì Timer sẽ được kích hoạt khi đó giá trị của Timer hiện hành đến bằng giá trị cài đặt và Output sẽ là OFF.
- TOFF gặp lệnh Reset thì Out Timer sẽ Off và giá trị hiện hành cũng trở về 0.

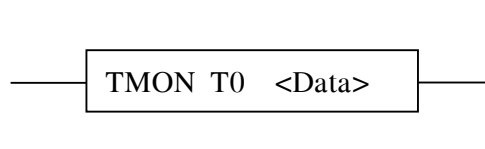
### ▫ Ví dụ:



### 3. Monostable Timer

▫ Cú pháp:

- Mnemonic Code: TMON <địa chỉ> <giá trị cài đặt>
- Ladder:

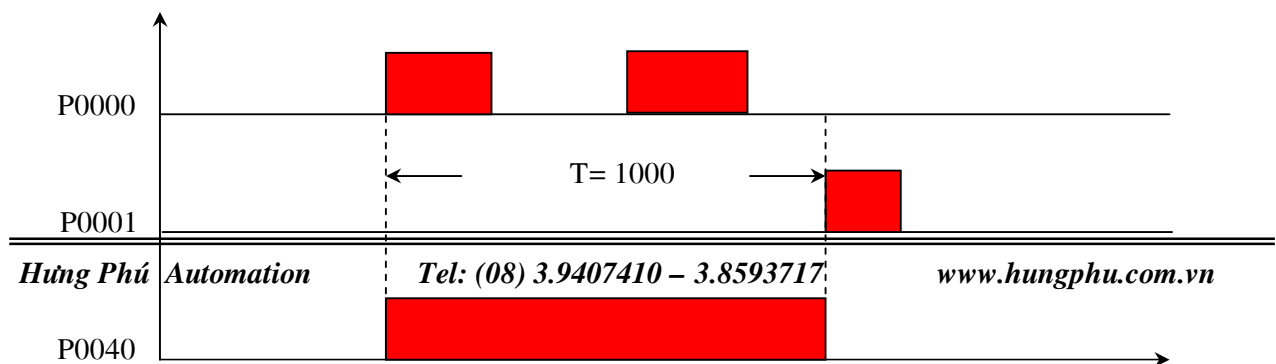
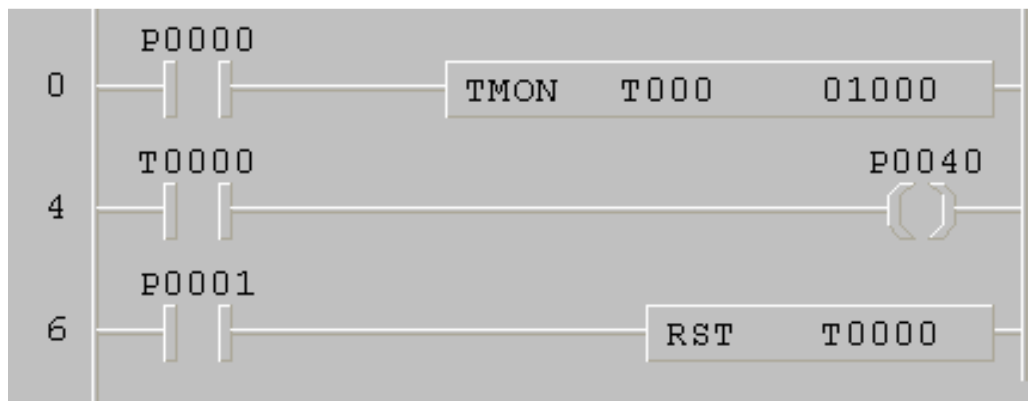


- Các vùng địa chỉ có thể dùng với lệnh TMON: T, D, Hằng số.

▫ Chức năng:

- Out Timer sẽ ON khi điều kiện input là ON và Out Timer sẽ OFF khi giá trị hiện hành của Timer giảm dần về giá trị 0 .
- Nếu Điều kiện input thay đổi từ ON sang OFF và sau đó ON lại trong khi Out timer đã ON sẽ không ảnh hưởng đến Out timer.
- TMON gặp lệnh Reset thì Out Timer sẽ Off và giá trị hiện hành cũng trở về 0.

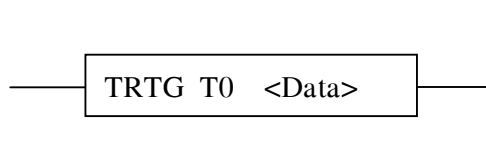
▫ Ví dụ:



#### 4. Retriggerble Monostable Timer

▫ Cú pháp:

- Mnemonic Code: TRTG <địa chỉ> <giá trị cài đặt>
- Ladder:

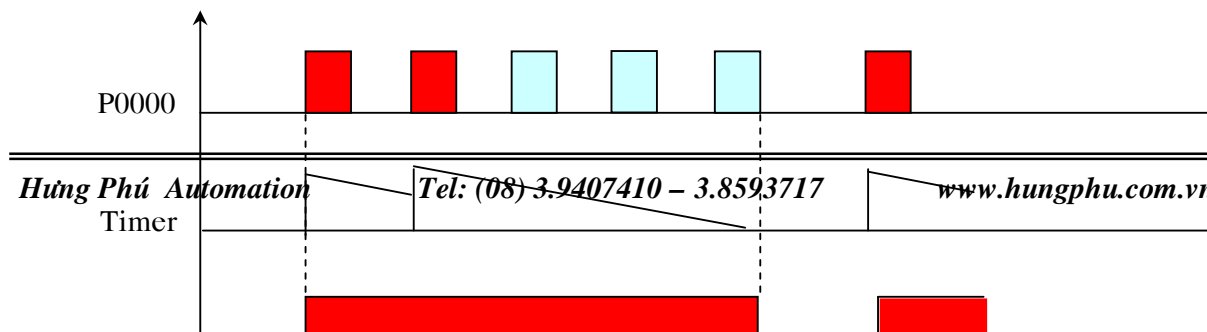
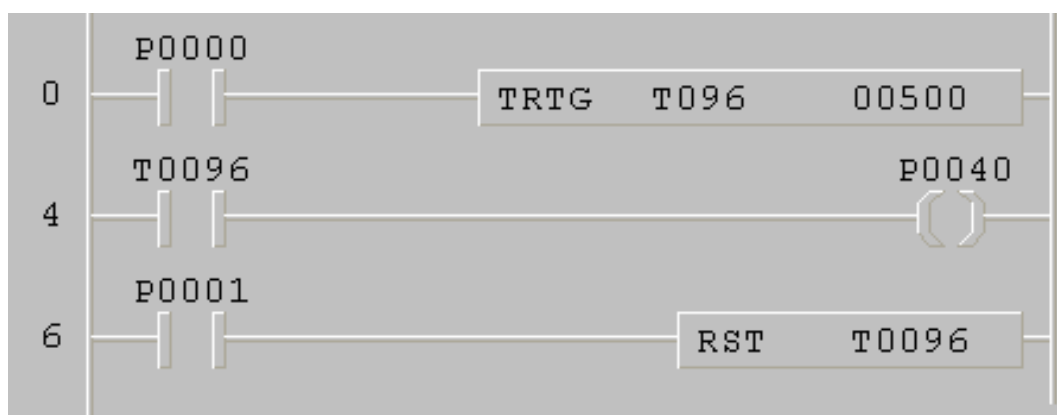


- Các vùng địa chỉ có thể dùng với lệnh TRTG: T, D, Hằng số.

▫ Chức năng:

- Out Timer sẽ ON khi điều kiện input là ON và Out Timer sẽ OFF khi giá trị hiện hành của Timer giảm dần về giá trị 0 .
- Nếu điều kiện input thay đổi từ ON sang OFF và sau đó ON lại trong khi Out timer đã ON thì giá trị hiện hành sẽ quay về lại giá trị ban đầu (giá trị cài đặt).
- TRTG gặp lệnh Reset thì Out Timer sẽ Off và giá trị hiện hành cũng trở về 0.

▫ Ví dụ:

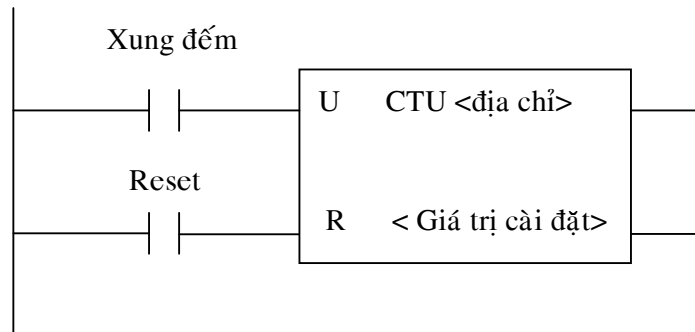


### III. BỘ ĐẾM COUNTER :

#### 1. Bộ đếm đếm lên CTU:

▫ Cú pháp:

- Mnemonic Code: CTU <địa chỉ> <giá trị cài đặt>
- Ladder:

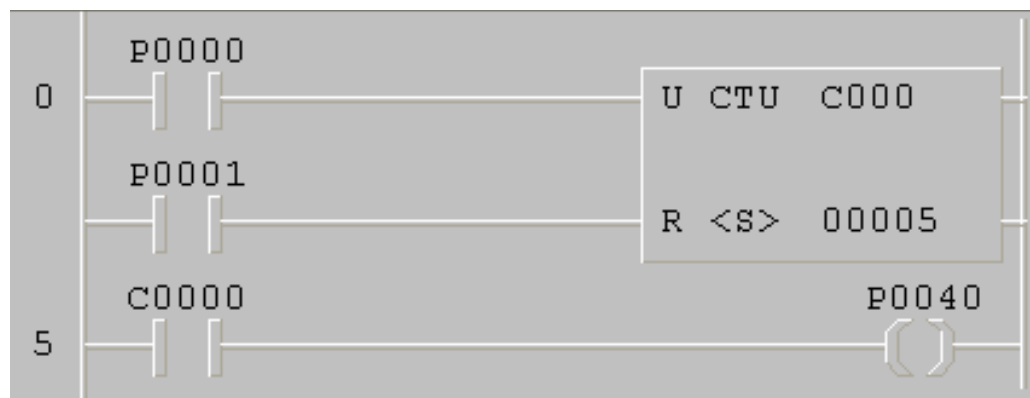


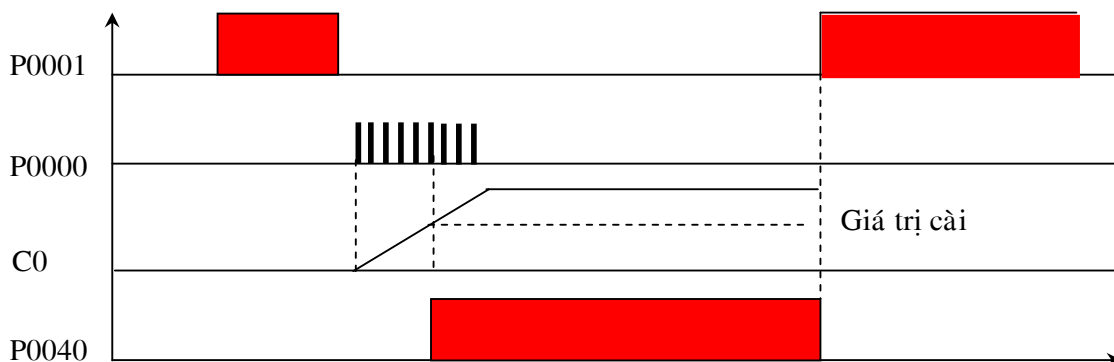
- Các vùng địa chỉ có thể dùng với lệnh CTU: C, D, Hằng số.

▫ Chức năng:

- Khi có tín hiệu xung đếm giá trị hiện hành của bộ đếm sẽ được thêm 1 và khi giá trị này bằng hoặc lớn hơn giá trị cài đặt thì Counter Out sẽ bật ON.
- Nếu Reset là ON thì Counter Out sẽ OFF và giá trị hiện hành sẽ trở về 0.
- Giá trị đếm max. của bộ đếm có thể lên đến (65535).

▫ Ví dụ:

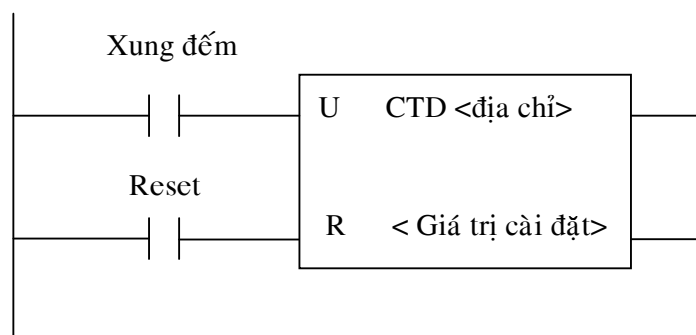




## 2. Bộ đếm đếm xuống CTD:

### ▫ Cú pháp:

- Mnemonic Code: CTD <địa chỉ> <giá trị cài đặt>
- Ladder:

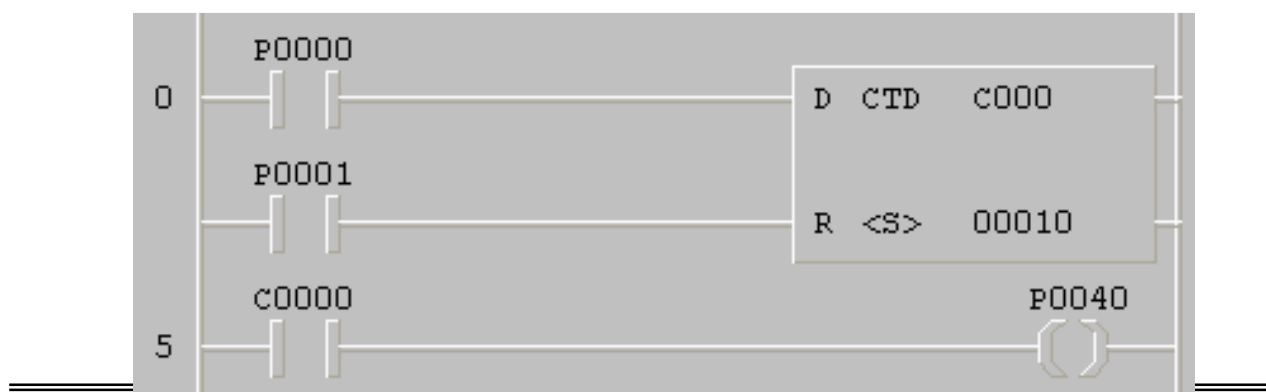


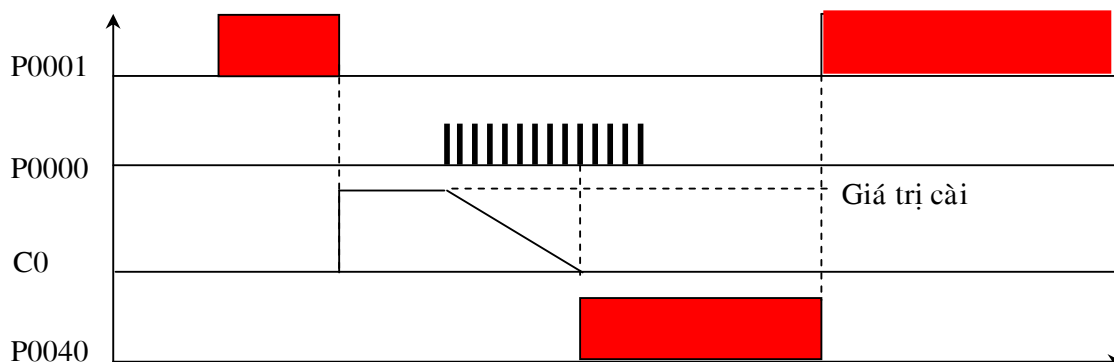
- Các vùng địa chỉ có thể dùng với lệnh CTD: C, D, Hằng số.

### ▫ Chức năng:

- Khi có tín hiệu xung đếm giá trị hiện hành của bộ đếm sẽ được trừ đi 1 và khi giá trị này bằng 0 thì Counter Out sẽ bật ON.
- Nếu Reset là ON thì Counter Out sẽ OFF và giá trị hiện hành sẽ trở về cài đặt.
- Giá trị đếm max. của bộ đếm có thể lên đến (65535).

### ▫ Ví dụ:

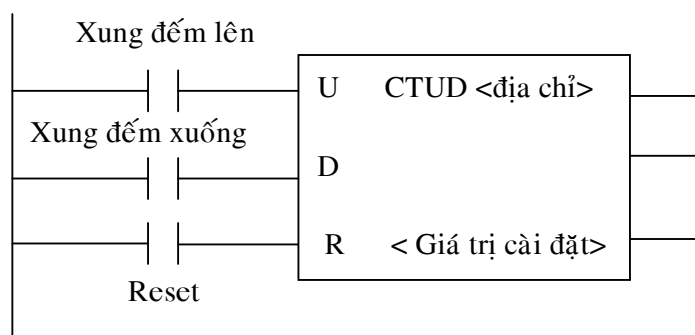




### 3. Bộ đếm đếm lên/xuống CTUD:

▫ Cú pháp:

- Mnemonic Code: CTUD <địa chỉ> <giá trị cài đặt>
- Ladder:

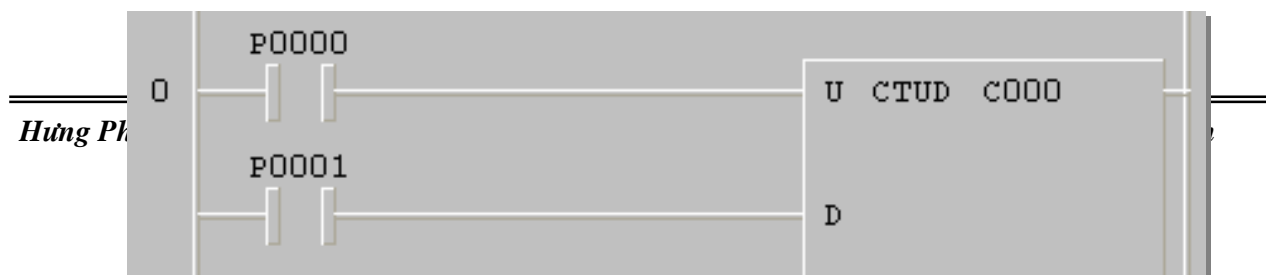


- Các vùng địa chỉ có thể dùng với lệnh CTUD: C, D, Hằng số.

▫ Chức năng:

- Khi có tín hiệu xung đếm lên giá trị hiện hành của bộ đếm sẽ được tăng thêm 1 và khi giá trị này bằng hoặc lớn hơn giá trị cài đặt thì Counter Out sẽ bật ON.
- Khi có tín hiệu xung đếm xuống giá trị hiện hành của bộ đếm sẽ được trừ đi 1 ON.
- Nếu Reset là ON thì Counter Out sẽ OFF và giá trị hiện hành sẽ trở về giá trị 0.
- Nếu tín hiệu xung đếm lên và xung đếm xuống tác động cùng một lúc thì giá trị hiện hành không thay đổi.
- Giá trị đếm max. của bộ đếm có thể lên đến (65535).

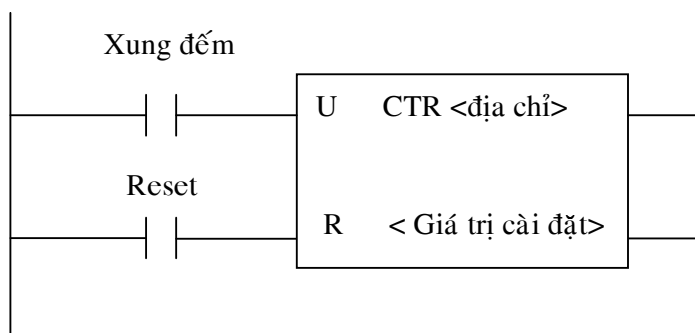
▫ Ví dụ:



#### 4. Bộ đếm Ring Counter CTR:

▫ Cú pháp:

- Mnemonic Code: CTR <địa chỉ> <giá trị cài đặt>
- Ladder:

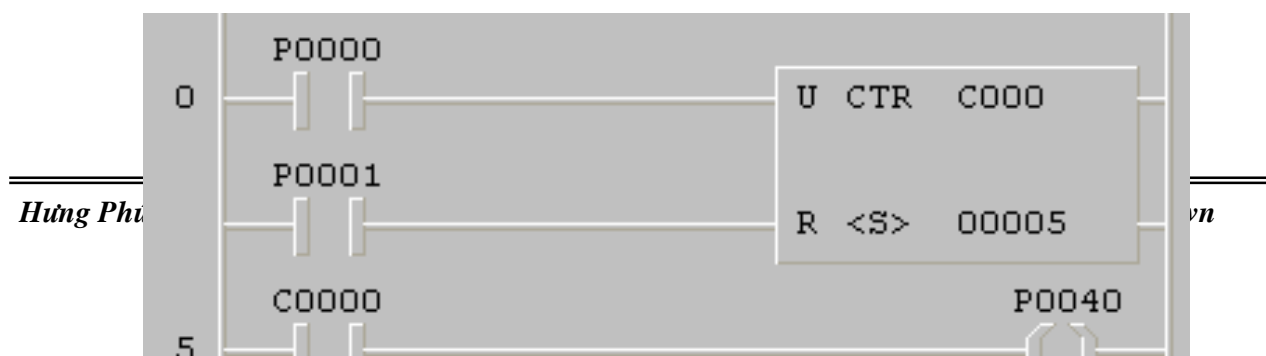


- Các vùng địa chỉ có thể dùng với lệnh CTR: C, D, Hằng số.

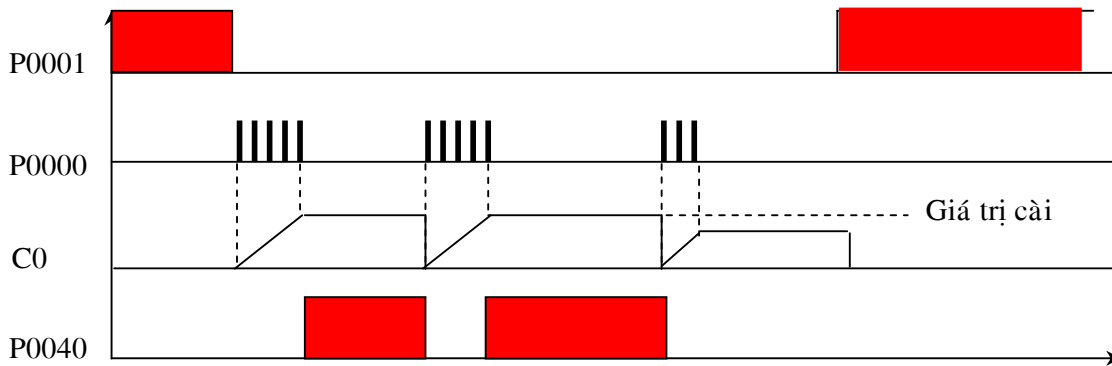
▫ Chức năng:

- Khi có tín hiệu xung đếm lên giá trị hiện hành của bộ đếm sẽ được tăng thêm 1 và khi giá trị này bằng hoặc lớn hơn giá trị cài đặt thì Counter Out sẽ bật ON đồng thời giá trị hiện hành sẽ trở về giá 0.
- Khi giá trị hiện hành nhỏ hơn giá trị cài đặt hay tín hiệu Reset là ON thì Counter Out sẽ là OFF.
- Giá trị đếm max. của bộ đếm có thể lên đến (65535).

▫ Ví dụ:







## HƯỚNG DẪN LẬP TRÌNH CĂN BẢN PLC MASTER-K

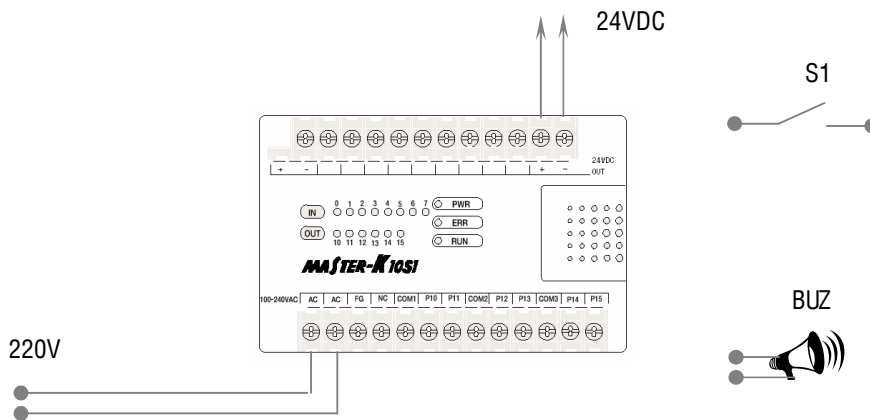
### Chương V: BÀI TẬP

**LS** Industrial Systems

#### BÀI TẬP I:

Một hệ thống báo cháy bao gồm một cảm biến mật độ khói (S1). Khi có cháy thì cảm biến S1 bị tác động làm cho còi (BUZ) phát tín hiệu.

□ **Vẽ sơ đồ đấu dây:**



□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Cảm biến S1	S1		
Còi BUZ	BUZ		

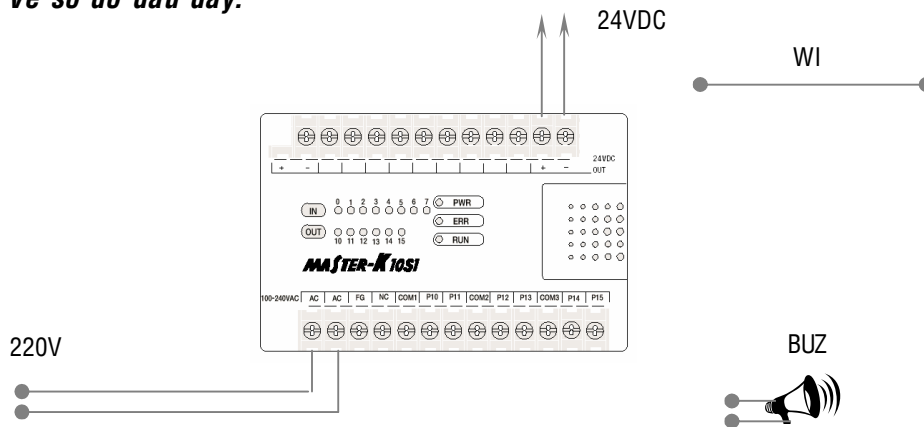
□ **Chương trình:**

**Ladder diagram:**

**BÀI TẬP 2:**

Một mạch điện bảo vệ được đặt trong một ngôi nhà. Mạch điện gồm một sợi dây kim loại mỏng (WI) được căng ngang qua cửa sổ. Nếu có người nào trèo qua cửa sổ sẽ làm đứt sợi dây, mạch bị ngắt và còi BUZ sẽ hú lên.

□ **Vẽ sơ đồ đấu dây:**

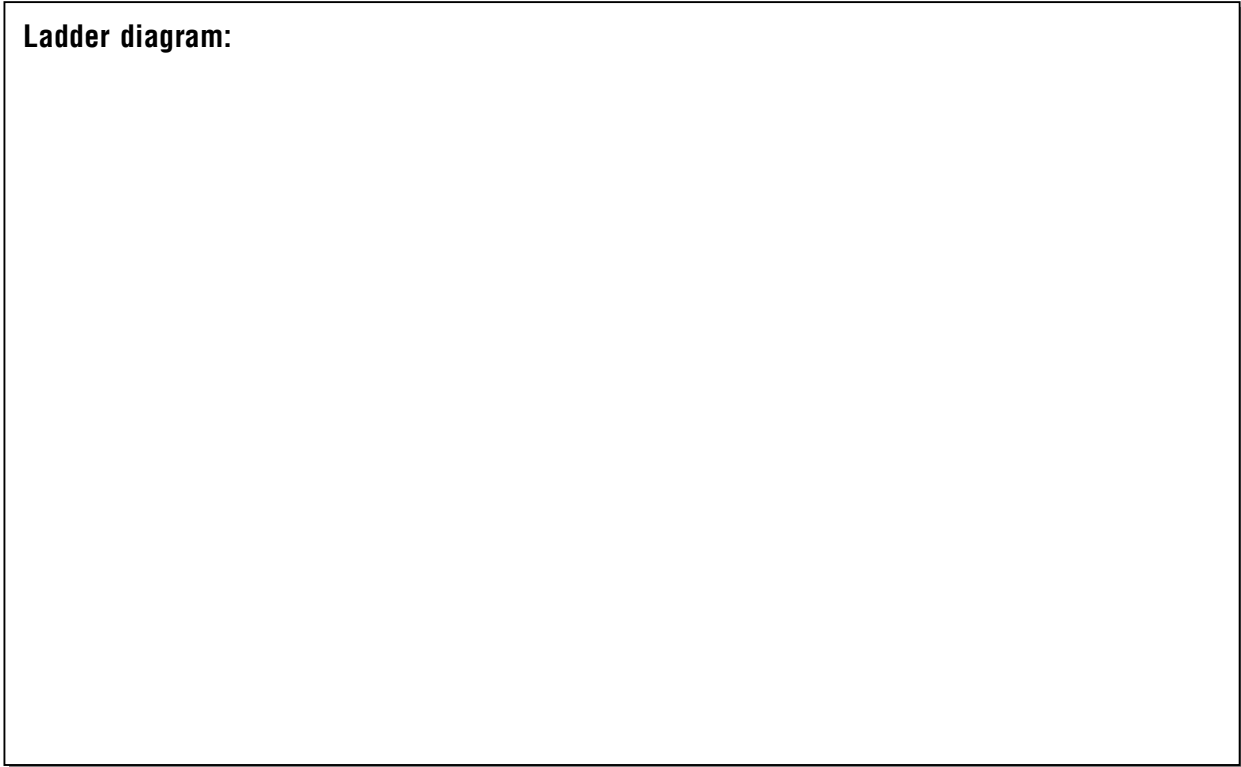


□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Sợi dây WI	WI		
Còi BUZ	BUZ		

□ **Chương trình:**

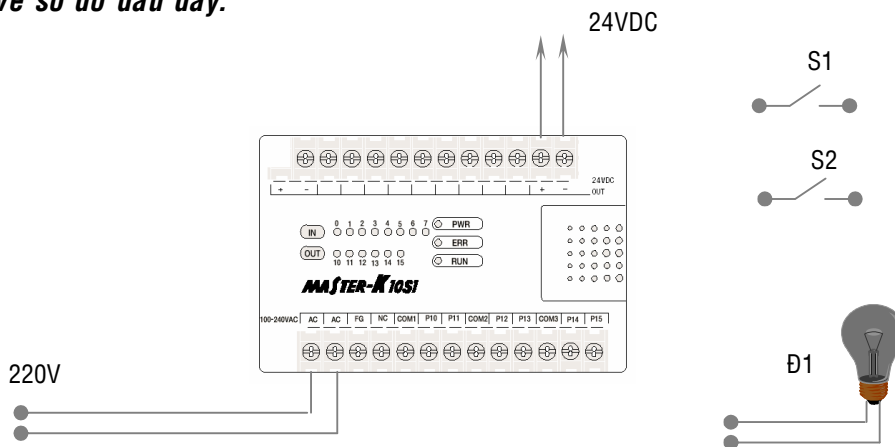
**Ladder diagram:**



**BÀI TẬP 3:**

Điều khiển một bóng đèn cầu thang, ta dùng 2 công tắc S1, S2. Công tắc S1 được gắn ở nhà dưới và công tắc S2 được gắn trên lầu. Khi bật công tắc S1 hoặc công tắc S2 thì đèn Đ1 sẽ sáng.

□ **Vẽ sơ đồ đấu dây:**

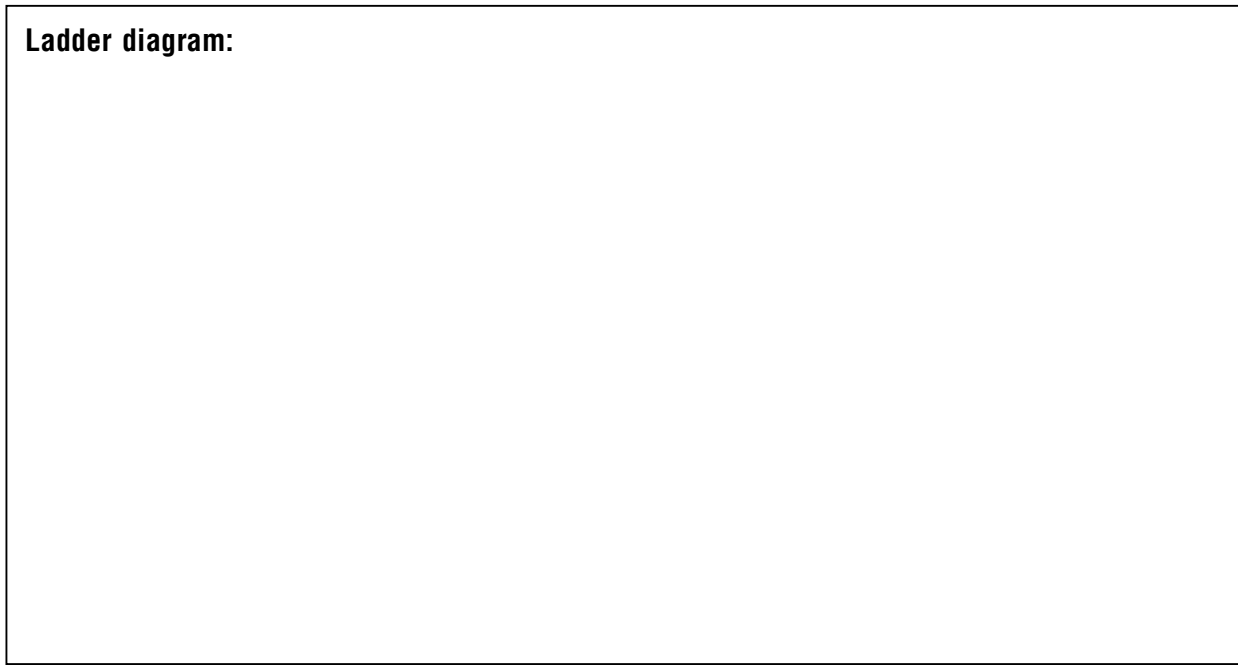


□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
----------	---------	-------------	---------

Công tắc S1	S1		
Công tắc S2	S2		
Đèn Đ1	Đ1		

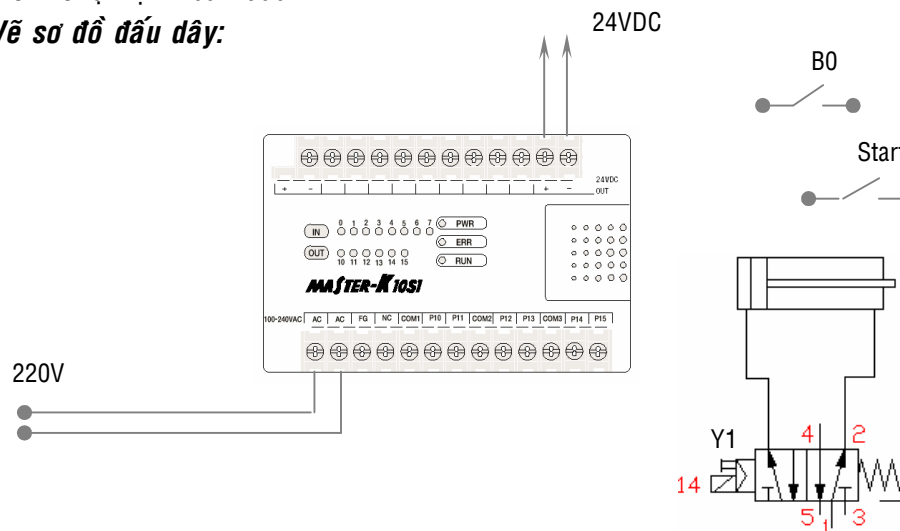
□ **Chương trình:**



**BÀI TẬP 4:**

Một máy dập bằng Cylinder khí được điều khiển bởi van điện từ và có lưới bảo vệ. Công tắc an toàn B0 (đặt dưới lưới bảo vệ) bị tác động khi lưới bảo vệ được kéo xuống, lúc này khi nhấn công tắc START thì máy sẽ hoạt động. Nếu lưới bảo vệ không được kéo xuống thì dù có bấm công tắc START máy cũng không hoạt động. Khi dập xuống máy sẽ giữ yên vị trí dập xuống khoảng 5 giây sau đó mới trở lại vị trí ban đầu.

□ **Vẽ sơ đồ đấu dây:**



□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Công tắc an toàn B0	S1		
Nút nhấn START	S2		
Van điện từ	Y1		
Bộ định thì	T0		

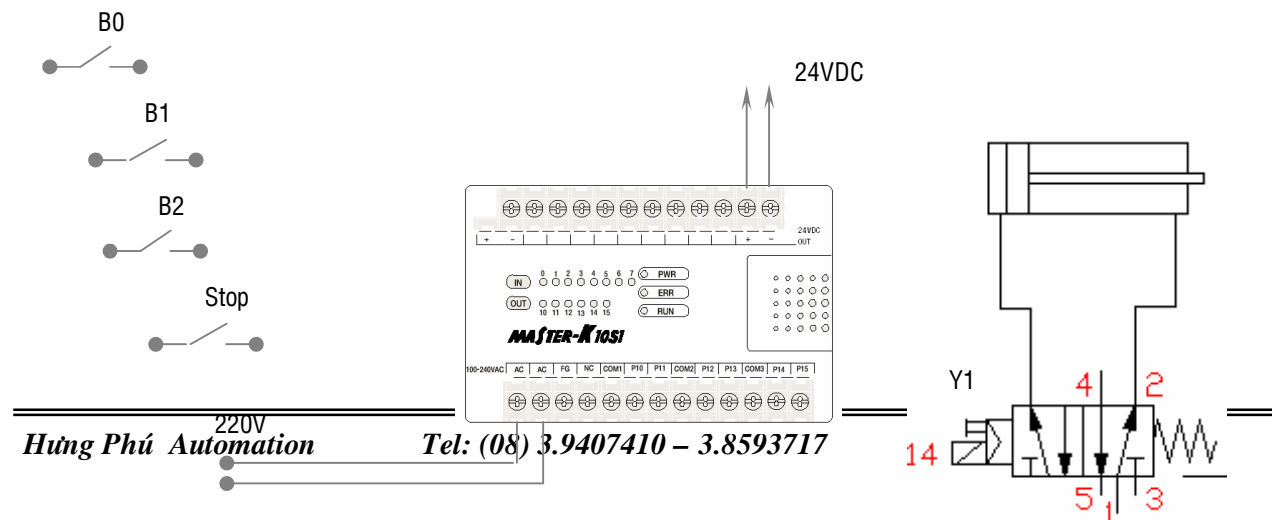
□ **Chương trình:**



**BÀI TẬP 5:**

Một máy dập lỗ có 3 công tắc B0, B1, B2 và nút nhấn STOP. Khi vật cần gia công vào thì chỉ cần chạm vào 2 trong 3 công tắc trên thì máy sẽ hoạt động. Máy hoạt động thì van điện từ Y1 được điều khiển và Cylinder khí dẫn ra, đục một lỗ thủng trên vật gia công và sẽ rút lên khi nào nhấn nút STOP.

□ **Vẽ sơ đồ đấu dây:**



□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Công tắc B0	B0		
Công tắc B1	B1		
Công tắc B2	B2		
Nút nhấn STOP	S1		
Van điện từ	Y1		

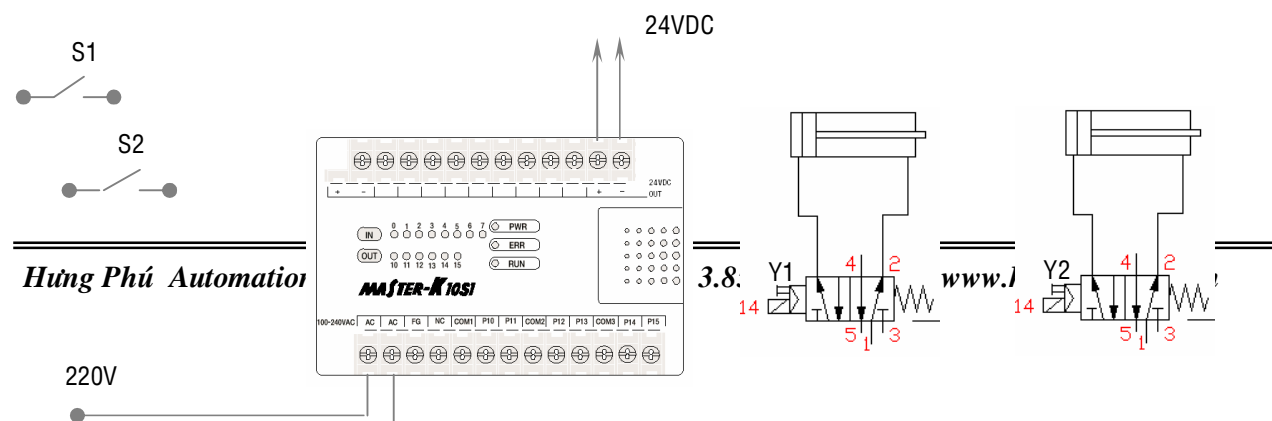
□ **Chương trình:**

**Ladder diagram:**

**BÀI TẬP 6:**

Trong một xưởng pha chế có 2 nguyên liệu A và B, công việc lựa chọn bởi hệ thống điều khiển hai Cylinder A và B được điều khiển bởi van điện từ Y1 và Y2 gắn vào trong hai thùng chứa. Công việc lựa chọn bắt đầu khi ta dùng công tắc chuyển S2, nếu công tắc S2 ở vị trí có logic '0' thì nguyên liệu A sẽ được chọn, còn nếu công tắc S2 ở vị trí có logic '1' thì nguyên liệu B được chọn. Khi bấm công tắc S1 thì nguyên liệu được chọn đi vào bồn chứa.

**Vẽ sơ đồ đấu dây:**



□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Nút nhấn S1	S1		
Công tắc S2	S2		
Van điện từ 1	Y1		
Van điện từ 2	Y2		

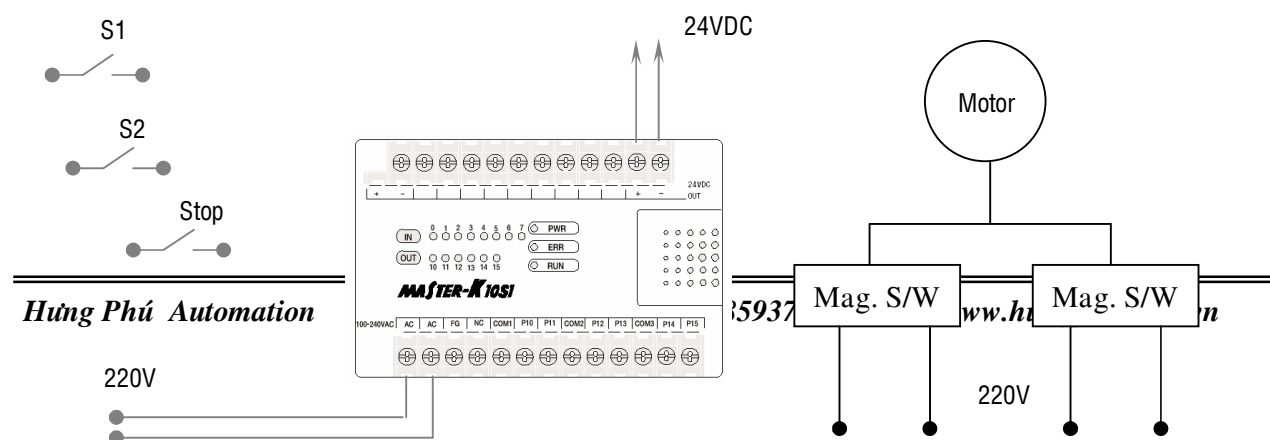
□ **Chương trình:**

**Ladder diagram:**

**BÀI TẬP 7:**

Mạch điều khiển Motor. Có 3 công tắc S1 là công tắc cho motor quay thuận, S2 là cho motor nghịch và S3 là nút nhấn STOP. Mạch điều khiển này có thể đổi chiều quay nhưng không cần nút stop. Hai khởi động từ Y1 và Y2 điều khiển hai chiều quay của motor.

□ **Vẽ sơ đồ đấu dây:**



□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Nút nhấn quay thuận	S1		
Nút nhấn quay nghịch	S2		
Nút nhấn Stop	Stop		
Cotactor Y1	Y1		
Contactơ Y2	Y2		

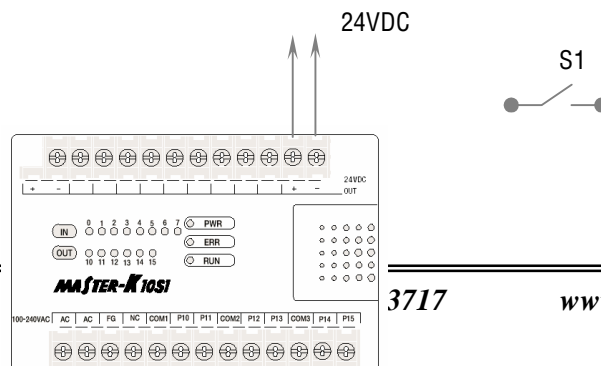
□ **Chương trình:**

**Ladder diagram:**

**BÀI TẬP 8:**

Tạo Flicker Timer được sử dụng bởi 2 bộ định thì T1 và T2. Bộ định thì hoạt động bởi công tắc S1 và khi đó sẽ có đèn Đ1 chớp tắt.

□ **Vẽ sơ đồ đấu dây:**







□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Công tắc S1	S1		
Đèn báo	Đ1		
Bộ định thì 1	T1		
Bộ định thì 2	T2		

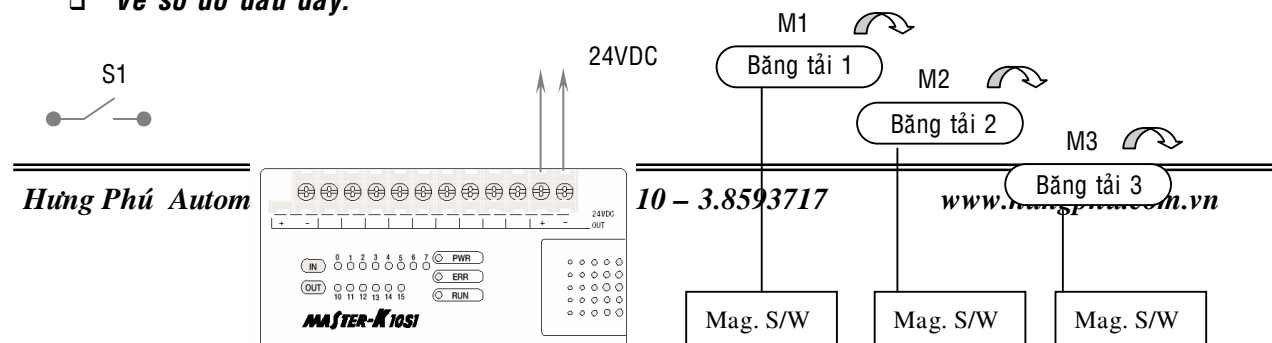
□ **Chương trình:**

**Ladder diagram:**

**BÀI TẬP 9:**

Có 3 băng tải M1, M2, M3 được điều khiển bởi 3 khởi động từ Y1, Y2, Y3. Khi công tắc S1 được đóng các băng tải sẽ hoạt động theo thứ tự M1 ⇒ M2 ⇒ M3 cách nhau 5 giây. Khi công tắc S1 mở thì các băng tải sẽ dừng theo thứ tự M3 ⇒ M2 ⇒ M1 cách nhau 5 giây.

□ **Vẽ sơ đồ đấu dây:**



□ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Công tắc S1	S1		
Contactor 1	Y1		
Contactor 2	Y2		
Contactor 3	Y3		

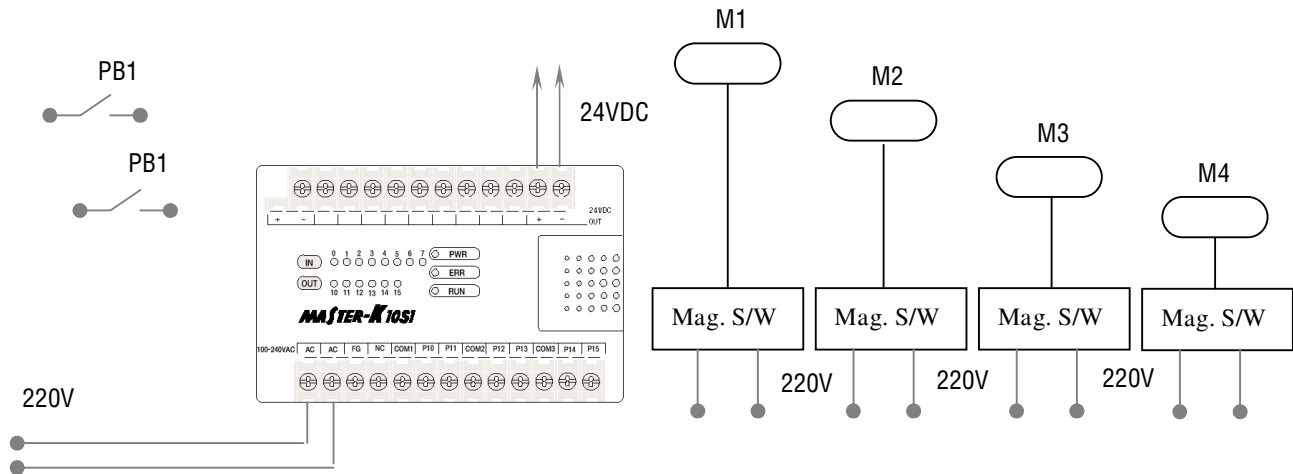
□ **Chương trình:**

**BÀI TẬP 10:**

Điều khiển số lượng motor hoạt động. Trong một hệ thống gồm có 4 motor được điều khiển bởi 4 contactor Y1, Y2, Y3, Y4. Có hai nút nhấn PB1 và PB2, PB1 để tăng số lượng motor hoạt động và PB2 dùng để giảm số lượng motor hoạt động. Các motor sẽ dừng lại khi nút PB1 nhấn trong khi 4

motor đang hoạt động và sẽ không còn motor hoạt động khi PB2 nhấn trong khi còn một motor đang hoạt động.

▣ **Vẽ sơ đồ đấu dây:**

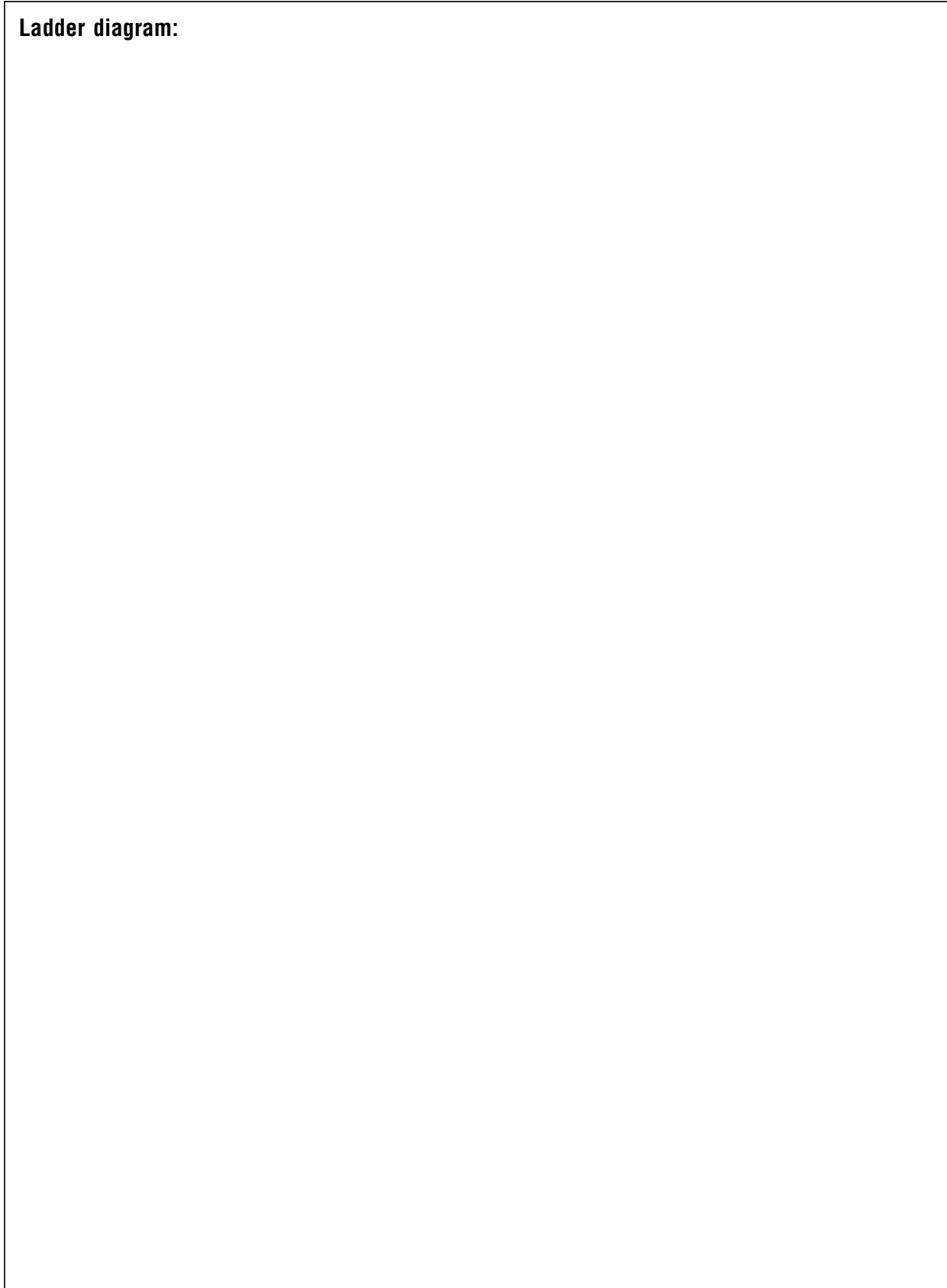


▣ **Bảng địa chỉ:**

Thiết kế	Ký hiệu	Địa chỉ PLC	Ý nghĩa
Nút nhấn tăng Motor	PB1		
Nút nhấn giảm Motor	PB2		
Contactơ 1	Y1		
Contactơ 2	Y2		
Contactơ 3	Y3		
Contactơ 4	Y4		

- Chương trình:

**Ladder diagram:**





---

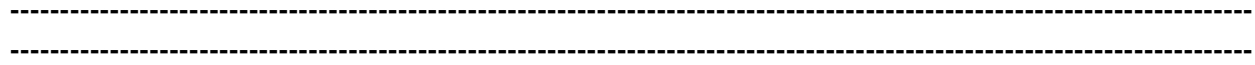
---

---

---

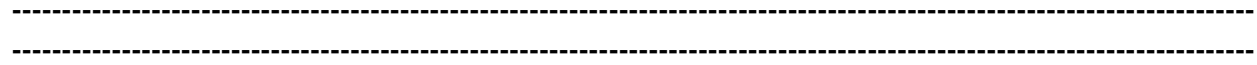
---



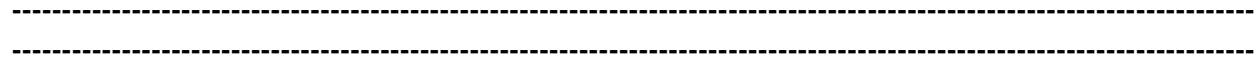












Dotted lines for writing.

