

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM  
TRUNG TÂM CÔNG NGHỆ ĐIỆN**



**GIÁO TRÌNH MÔN HỌC**

# PLC

## MITSUBISHI PROGRAMMABLE CONTROLLERS



**BIÊN SOẠN : LÊ HOÀNG VINH \*ĐÀO DUY KHƯƠNG  
VÕ THỊ ÁNH TUYẾT \*TRẦN THỊ THU THỦY**

TÁI BẢN THÁNG 07 NĂM 2006  
LƯU HÀNH NỘI BỘ

# LỜI NÓI ĐẦU

Trong những năm gần đây, bộ điều khiển lập trình PLC được sử dụng ngày càng rộng rãi trong công nghiệp ở nước ta như là một giải pháp điều khiển lý tưởng cho việc tự động hoá các quá trình sản xuất. Cùng với sự phát triển của công nghệ máy tính, Hiện nay bộ điều khiển lập trình PLC hay CPU đạt được những ưu thế cơ bản trong ứng dụng điều khiển công nghiệp đáp ứng yêu cầu điều khiển và giám sát hệ thống sản xuất từ đơn giản đến phức tạp với độ tin cậy cao. PLC (Programmable logic controller) là một máy tính thu nhỏ nhưng với các tiêu chuẩn công nghiệp cao và khả năng lập trình logic mạnh PLC là đầu não quan trọng và linh hoạt trong điều khiển tự động hoá điện công nghiệp. Trong những nhà máy sản xuất tự động hoá hiện nay, ở nhiều cấp độ khác nhau, phương thức điều khiển giám sát (SCADA – supervisory control and data acquisition ) được áp dụng để dàng với các phần tử điều khiển chấp hành gồm toàn bộ các bộ điều khiển lập trình (PLC)

Môn PLC thuộc trung tâm tự động hoá điện công nghiệp của TRUNG TÂM CÔNG NGHỆ ĐIỆN – TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM là môn học khai thác sử dụng các bộ điều khiển lập trình (PLC), qua đó giới thiệu các ngôn ngữ lập trình logic phương pháp lập trình công nghệ và các ứng dụng mở rộng nhằm cung cấp các kiến thức cơ bản về PLC cho sinh viên để tiếp cận sự điều khiển tự động hoá trong các nhà máy sản xuất hiện đại, góp phần trong công cuộc công nghiệp hoá hiện đại hoá của đất nước .

Giáo trình này cung cấp một số khía cạnh của bộ điều khiển lập trình (PLC) dựa trên quá trình tìm hiểu về bộ PLC FX1N thuộc họ FX của hãng Mitsubishi electric. Qua đó giúp cho sinh viên, học sinh có những khái niệm cơ bản về điều khiển tự động hoá dùng PLC trong công nghiệp, trình bày một số lệnh cơ bản chủ yếu về tập lệnh của PLC Mitsubishi electric, cách cài đặt, sử dụng các lệnh để tạo ra chương trình điều khiển và cách sử dụng phần mềm FXGPWIN và phần mềm mô phỏng FX TRAINING BEG –E. Ngoài ra còn có nhiều bài tập đã được giải đáp và nhiều bài tập mở rộng giúp cho sinh viên học sinh phát triển kỹ năng cần thiết trong việc lập trình PLC.

Qua quá trình biên soạn giáo trình này chắc chắn còn có nhiều thiếu sót. Chúng tôi mong nhận được những ý kiến đóng góp xây dựng của quý Thầy Cô đồng nghiệp và sinh viên học sinh để giáo trình này được hoàn thiện hơn, xin chân trọng cảm ơn.

# MỤC LỤC

## LỜI NÓI ĐẦU

## MỤC LỤC

✓ <b>Chương 1: GIỚI THIỆU TỔNG QUAN VỀ PLC.....</b>	<b>2</b>
I. Lịch sử ra đời và phát triển của PLC.....	2
II. Khái quát về bộ lập trình PLC.....	3
III. Cấu Trúc phần cứng PLC của họ FX .....	5
IV. Giới thiệu các PLC họ FX Family.....	6
✓ <b>Chương 2: CÁC THIẾT BỊ VÀ LỆNH CƠ BẢN.....</b>	<b>11</b>
<b>A. khái niệm.....</b>	<b>11</b>
I. Khái niệm về một chương trình.....	11
II. Các khái niệm cơ bản dùng trong lập trình .....	11
III. Cách đọc Logic của chương trình Ladder.....	12
<b>B. Các tập lệnh cơ bản của lập trình ladder và instruction.....</b>	<b>13</b>
✓ <b>Chương 3: KỸ THUẬT LẬP TRÌNH STEP LADDER .....</b>	<b>28</b>
I. Khái niệm về lập trình STL.....	28
II. Khởi động và kết thúc chương trình STL.....	28
III. Lập bước giữa các trạng thái của chương trình STL.....	29
IV. Cấu trúc của một bước trong chương trình STL.....	30
✓ <b>Chương 4: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM FXTRN-BEG-E.....</b>	<b>31</b>
I. Giới thiệu .....	31
II. Giao diện phần mềm FXTRN-BEG - E .....	34
III. Giới thiệu Giao diện chương trình FXTRN-BEG - E .....	35
IV. Phương pháp lập trình FXTRN-BEG - E .....	37
V. Bài tập ứng dụng phần mềm FX-TRN-BEG E .....	42
✓ <b>Chương 5: CÁC BÀI TẬP ỨNG DỤNG TRONG PLC FXTRN-BEG-E .....</b>	<b>43</b>
I. Các dạng bài tập.....	43
II. Bài giải tham khảo các bài tập trong FXTRN-BEG - E .....	46
1. Bài tập A3.....	46
2. Bài tập B1,B2.....	48
3. Bài tập B3.....	50
4. Bài tập B4.....	52
5. Bài tập C1,2.....	54
6. Bài tập C4.....	56
7. Bài tập D1.....	57
8. Bài tập D2.....	59
9. Bài tập D3.....	61
10. Bài tập D4,5.....	64
11. Bài tập D6.....	67
12. Bài tập E1.....	70

13. Bài tập E2.....	72
14. Bài tập E3.....	74
15. Bài tập E4.....	75
16. Bài tập E5.....	77
17. Bài tập E6.....	79
18. Bài tập F1.....	81
19. Bài tập F2.....	84
20. Bài tập F3.....	88
21. Bài tập F4.....	91
22. Bài tập F5.....	92
23. Bài tập F6.....	95
24. Bài tập F7.....	98
<b>✓ Chương 6: SỬ DỤNG PHẦN MỀM FXGP-WIN E.....</b>	<b>101</b>
I. Giới thiệu .....	101
II. Phần mềm FXGP-WINE.....	102
III. Các bước lập một chương trình mới.....	109
IV. Phương pháp lập trình.....	109
V. Bài tập ứng dụng phần mềm FXGP-WINE.....	112
<b>✓ Chương 7: LỰA CHỌN, LẮP ĐẶT VÀ KIỂM TRA.....</b>	<b>121</b>
<b>BẢO TRÌ HỆ THỐNG PLC</b>	
I. Xem xét sự khả thi .....	121
II. Trình tự thiết kế hệ thống PLC .....	121
III. Tổ chức bố trí phần cứng hệ thống .....	123
IV. Chạy thử chương trình .....	124
V. Lập tài liệu cho hệ thống .....	125
VI. Bảo trì hệ thống PLC.....	125

## GIÁO TRÌNH MÔN HỌC

# PLC

## MITSUBISHI PROGRAMMABLE CONTROLLERS

High End Processing  
with Connector Type I/O

**FX<sub>2NC</sub>**  
The FX<sub>2NC</sub> Series PLC has some of the most spectacular characteristics while retaining the FX<sub>2N</sub> Series potential features set. The connector type I/O reduces wiring costs and maintenance time.

**FX<sub>2N</sub>**  
The FX<sub>2N</sub> Series PLC is the most advanced model of the FX Series. With unmatched speed, advanced functions, analog systems and positioning control, the FX<sub>2N</sub> is the choice for many applications from 16 to 128 I/O.

**FX<sub>1N</sub>**  
The FX<sub>1N</sub> Series PLC is a popular choice for control of up to 128 points. Because the FX<sub>1N</sub> Series PLC has an excellent I/O, analog control, and communication functions, it can be used in a wide range of general sequence control applications.

**Control Scale**

# GIỚI THIỆU TỔNG QUAN VỀ PLC

## I. LỊCH SỬ RA ĐỜI VÀ PHÁT TRIỂN CỦA PLC

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

Khái niệm bộ điều khiển lập trình PLC là ý tưởng của nhóm kỹ sư hãng General Motors vào năm 1968, và họ đã đề ra các chỉ tiêu kỹ thuật nhằm đáp ứng những yêu cầu điều khiển như sau :

- Dễ lập trình và thay đổi chương trình điều khiển, sử dụng thích hợp trong các nhà máy công nghiệp.
- Cấu trúc dạng Module dễ mở rộng, dễ bảo trì và sửa chữa.
- Đảm bảo độ tin cậy trong môi trường sản xuất của các nhà máy công nghiệp.
- Sử dụng các linh kiện bán dẫn nên phải có kích thước nhỏ gọn hơn mạch relay mà chức năng vẫn tương đương.
- Giá cả cạnh tranh.

Những chỉ tiêu này đã tạo được sự quan tâm của những kỹ sư thuộc nhiều ngành nghiên cứu khả năng ứng dụng PLC trong công nghiệp. Các kết quả nghiên cứu đã đưa ra thêm một số các chỉ tiêu cần phải có trong chức năng của PLC :

### a) Về phần mềm :

Từ các lệnh logic đơn giản được hỗ trợ thêm các lệnh về tác vụ định thì, tác vụ đếm. Sau đó là các lệnh về xử lý toán học, xử lý bảng dữ liệu, xử lý xung ở tốc độ cao, tính toán số liệu thực 32 bit, xử lý thời gian thực, đọc mã vạch...

### b) Về phần cứng :

- Bộ nhớ lớn hơn.
- Số lượng ngõ vào, ngõ ra nhiều hơn.
- Nhiều loại module chuyên dùng hơn.

Đến năm 1976 thì PLC có khả năng điều khiển các ngõ vào, ngõ ra từ xa bằng kỹ thuật truyền thông ( khoảng 200 mét ).

Sự gia tăng những ứng dụng của PLC trong công nghiệp đã thúc đẩy các nhà sản xuất hoàn chỉnh kỹ thuật của các họ PLC với mức độ khác nhau về khả năng tốc độ xử lý và hiệu suất.

Các họ PLC phát triển từ loại làm việc độc lập, chỉ với 20 ngõ vào / ra và dung lượng bộ nhớ chương trình khoảng 500 bước, đến các họ PLC có cấu trúc module nhằm làm dễ dàng hơn cho việc mở rộng thêm chức năng chuyên dùng như :

- Xử lý tín hiệu liên tục.

- Điều khiển động cơ Servo, động cơ bước.
- Truyền thông.
- Bộ nhớ mở rộng.

Với cấu trúc module cho phép mở rộng hay nâng cấp một hệ thống điều khiển PLC với chi phí và công sức thấp nhất.

Riêng nước ta, hàng rào thuế quan khu vực đang dần dần được loại bỏ, kinh tế mở cửa hợp tác với nước ngoài. Trước tình hình đó, nền công nghiệp sẽ gặp không ít khó khăn vì còn khá nhiều dây chuyền công nghệ lạc hậu. Nhà nước cần phải chú trọng đến những ứng dụng và phát triển của tự động trong sản xuất công nghiệp, nhằm nâng cao năng suất, chất lượng sản phẩm, cũng như giá thành sản phẩm hạ. Một trong những phương án tốt nhất và được sử dụng rộng rãi ngày nay là thay thế những công nghệ cũ bằng những hệ thống điều khiển tự động dùng PLC.

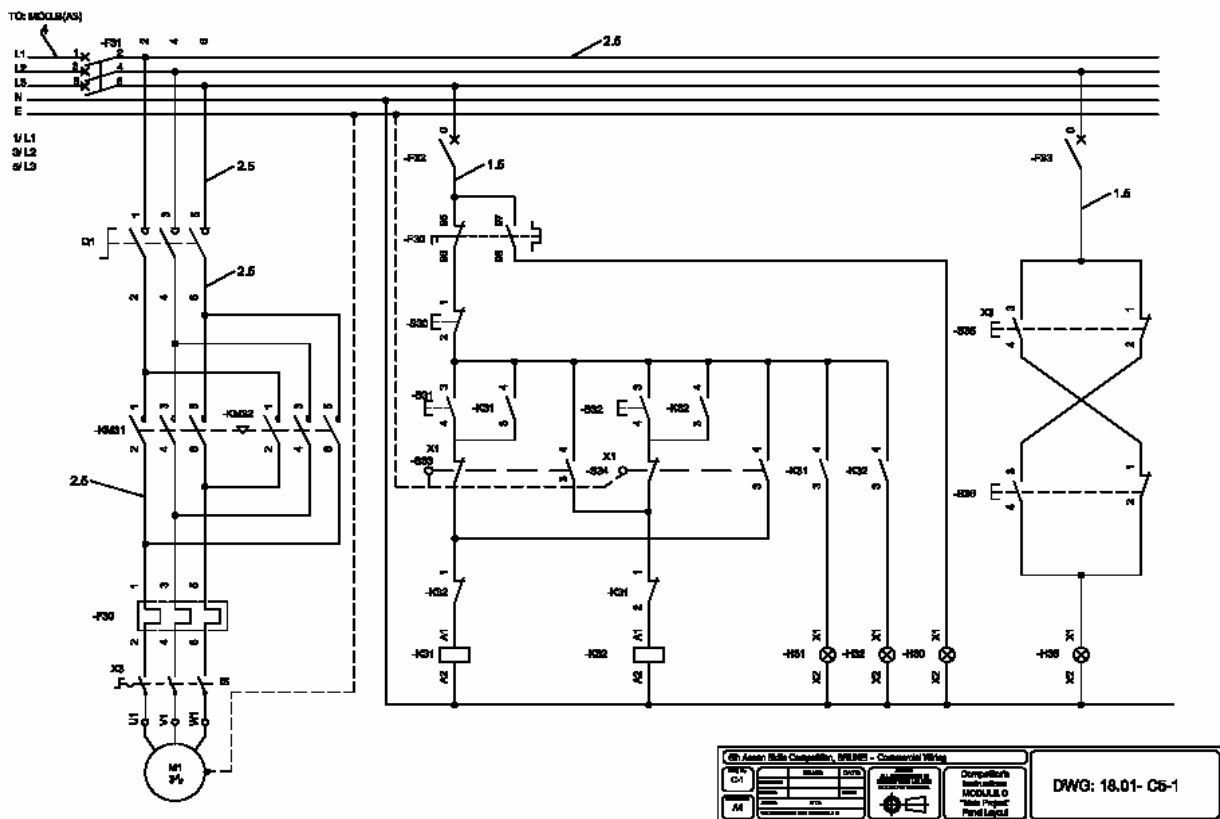
## II. KHÁI QUÁT VỀ BỘ LẬP TRÌNH PLC

Trong kỹ thuật tự động, điều khiển được chia làm hai loại : điều khiển bằng dây nối, và điều khiển bằng bộ lập trình PLC.

### 1. Điều khiển bằng dây nối :

Các bộ điều khiển được gọi là bộ điều khiển dây nối nếu các phần tử chuyển mạch của nó được nối với nhau bằng dây dẫn cố định một cách vĩnh viễn.

VD : mạch điều khiển sau



Trong ký hiệu trên, đó là bộ điều khiển bằng dây nối dùng để nối kết các nút nhấn và các phần tử chuyển mạch là các tiếp điểm với nhau bằng dây dẫn song song hoặc nối tiếp. Các công tắc và tiếp điểm được sắp xếp với nhau khi ta đã biết rõ chức năng mà bộ điều khiển cần thực hiện, tức phải biết rõ sơ đồ nguyên lý và vị trí khi tiến hành đấu dây. Mặt khác, khi muốn thay đổi chức năng của bộ điều khiển ta phải thay đổi lại cấu trúc cũng như sơ đồ đấu dây. Đối với những mạch điện lớn phức tạp thì sự đấu dây trở nên rất khó khăn và dễ bị sai sót. Ngoài ra trong cách dùng này cần tốn nhiều linh kiện như : role trung gian, role thời gian, bộ đếm...

## 2. Điều khiển dùng PLC :

PLC là từ viết tắt của Programmable Logical Controller ( chương trình điều khiển tự động có lập trình), chương trình này được lưu trữ trong bộ nhớ ROM và được nạp vào thông qua máy vi tính cá nhân.

Trong PLC chức năng bộ điều khiển cần thực hiện sẽ được xác định bởi một chương trình, chương trình này được nạp vào bộ nhớ PLC. Khi đó PLC sẽ thực hiện quá trình điều khiển dựa vào chương trình đã được nạp sẵn. Cấu trúc và sơ đồ đấu dây của bộ điều khiển không phụ thuộc vào chức năng hay quá trình hoạt động.

Tất cả các linh kiện cần thiết cho việc thiết kế mạch đều được lập trình sẵn trong bộ PLC như : sensor, công tắc, nút nhấn, tế bào quang điện, và tất cả các cơ cấu chấp hành như cuộn dây, đèn tín hiệu, bộ định thì, role trung gian, ... đều được nối vào PLC.

Nếu muốn thay đổi hay mở rộng chức năng của quy trình công nghệ ta chỉ cần thay đổi chương trình bên trong bộ PLC. Điều này rất tiện ích cho các kỹ sư thiết kế.

## 3. Những ưu điểm kỹ thuật của bộ điều khiển PLC :

Chỉ tiêu so sánh	Role	Mạch số	Máy tính	PLC
Giá thành	Khá thấp	Thấp	Cao	Thấp
Kích thước vật lý	Lớn	Rất gọn	Khá gọn	Rất gọn
Tốc độ điều khiển	Chậm	Rất nhanh	Khá nhanh	Nhanh
Khả năng chống nhiễu	Rất tốt	Tốt	Khá tốt	Tốt
Lắp đặt	Mất thời gian thiết kế và lắp đặt.	Mất thời gian để thiết kế.	Lập trình phức tạp và tốn thời gian.	Lập trình và lắp đặt đơn giản.
Khả năng điều khiển các tác vụ phức tạp.	Không có	Có	Có	Có
Thay đổi, nâng cấp và điều khiển.	Rất khó	Khó	Khá đơn giản	Rất đơn giản
Công tác bảo trì	Kém	Kém	Kém	Tốt

Theo bảng so sánh ta nhận thấy được bộ điều khiển lập trình PLC với những ưu điểm về phần cứng và phần mềm có thể đáp ứng được hầu hết các yêu cầu chỉ tiêu



trên. Mặt khác, PLC có khả năng kết nối mạng và kết nối các thiết bị ngoại vi rất cao giúp cho việc điều khiển được dễ dàng.

#### **4. Phạm vi ứng dụng PLC :**

Dùng để điều khiển Robot : ví dụ như gấp phôi từ băng tải bỏ qua bàn gia công của máy CNC, hay điều khiển Robot đưa vật liệu thiết bị vào băng tải, thực hiện các việc đóng hộp, dán tem nhãn...

Ngoài ra, PLC có thể ứng dụng để giám sát các quá trình trong các nhà máy mạ, dây chuyền lắp ráp linh kiện điện tử, dây chuyền kiểm tra sản phẩm... bằng các sensor, công tắc hành trình.

Ứng dụng PLC trong các lĩnh vực

### **III . CẤU TRÚC PHẦN CỨNG PLC HỌ FX CỦA HÃNG MITSUBISHI**

Cấu trúc của PLC được phân thành các phần như sau :

#### **1. Đơn vị điều khiển trung tâm ( CPU : Central Processing Unit )**

Là bộ vi xử lý thực hiện các lệnh trong bộ nhớ chương trình. Nhập dữ liệu ở ngõ vào, xử lý chương trình, nhớ chương trình, xử lý các kết quả trung gian và các kết

quả này được truyền trực tiếp đến cơ cấu chấp hành để thực hiện chương trình xuất dữ liệu ra các ngõ ra.

## 2. Bộ nhớ (Memory) :

Dùng để chứa chương trình số liệu, đơn vị nhỏ nhất là bit. Bộ nhớ là vùng nắm giữ hệ điều hành và vùng nhớ của người sử dụng (hệ điều hành là một phần mềm hệ thống mà nó kết nối PLC để PLC thực sự hoạt động được).

Có nhiều loại bộ nhớ khác nhau. Để PLC có thể hoạt động được, cần thiết phải có bộ nhớ để lưu trữ chương trình. Đôi khi cần mở rộng bộ nhớ để thực hiện các chức năng khác như

\_ Vùng đệm tạm thời lưu trữ trạng thái của các kênh xuất - nhập được gọi là RAM xuất -nhập.

\_ Lưu trữ tạm thời các trạng thái của các chức năng bên trong : các bộ định thì(Timer), các bộ đếm (Counter), các Role.

Bộ nhớ gồm có các loại sau đây :

### a) Bộ nhớ chỉ đọc (ROM : Read Only Memory)

ROM không phải là bộ nhớ khả biến, nó có thể lập trình chỉ được một lần. Do đó nó không thích hợp cho việc điều khiển “mềm” của PLC, và nó ít phổ biến so với các loại bộ nhớ khác.

### b) Bộ nhớ ghi đọc (RAM : Random Access Memory)

Bộ nhớ của PLC là CMOSRAM, tiêu tốn năng lượng khá ít, và được cấp pin dự phòng khi mất nguồn. Nhờ đó dữ liệu sẽ không bị mất.

### c) Bộ nhớ chỉ đọc chương trình xóa được (EPROM : Erasable Programmable Read Only Memory)

EPROM lưu trữ dữ liệu giống như ROM, tuy nhiên nội dung của nó có thể được xóa đi nếu bị ảnh hưởng của tia tử ngoại. Khi đó phải viết lại chương trình cho bộ nhớ.

### d) Bộ nhớ chỉ đọc chương trình xóa được bằng điện (EEPROM : Electric Erasable Programmable Read Only Memory)

Nội dung trên EEPROM có thể bị xóa và lập trình bằng điện, tuy nhiên chỉ giới hạn một số lần nhất định.

## 3. Các Module xuất-nhập ( Input – Output ) :

Khối xuất – nhập đóng vai trò là mạch giao tiếp giữa vi mạch điện tử bên trong PLC với mạch ngoài. Module nhập nhận tín hiệu từ sensor và đưa vào CPU, module xuất đưa tín hiệu điều khiển từ CPU ra cơ cấu chấp hành.

Mọi hoạt động xử lý tín hiệu từ bên trong PLC có mức điện áp từ 5 ÷ 15 VDC, trong khi tín hiệu bên ngoài có thể lớn hơn nhiều. Ta có nhiều loại ngõ ra như : ngõ ra dùng role, ngõ ra dùng transistor, ngõ ra dùng triac.

4. **Hệ thống BUS** : là hệ thống tập hợp một số dây dẫn kết nối các module trong PLC gọi là BUS, đây là tuyến dùng để truyền tín hiệu, hệ thống gồm nhiều tín hiệu song song.

## IV. GIỚI THIỆU CÁC PLC HỌ FX FAMILY

### 1. PLC loại FXO & FXOS

Đây là loại PLC có kích thước thật nhỏ gọn, phù hợp với các ứng dụng với số lượng I/O nhỏ hơn 30 cổng, với việc sử dụng bộ nhớ chương trình bằng EEPROM cho phép dữ liệu chương trình được lưu lại trong bộ nhớ khi mất nguồn đột xuất. Dòng FXO được tích hợp sẵn bên trong bộ đếm tốc độ cao và các bộ tạo ngắt (role trung gian), cho phép xử lý tốt một số ứng dụng phức tạp.

Nhược điểm của dòng FXO là không có khả năng mở rộng số lượng I/O được quản lý, không có khả năng nối mạng, không có khả năng kết nối với các module chuyên dùng, thời gian thực hiện chương trình lâu.

### 2. PLC loại FXON

**FXON** sử dụng cho các máy điều khiển độc lập hay các hệ thống nhỏ với số lượng I/O có thể quản lý nằm trong miền 10-128 I/O. FXON thực chất là bước đệm trung gian giữa FXOS với FXO. PLC FXON có đầy đủ các đặc trưng cơ bản của dòng FXOS, đồng thời còn có khả năng mở rộng tham gia nối mạng.

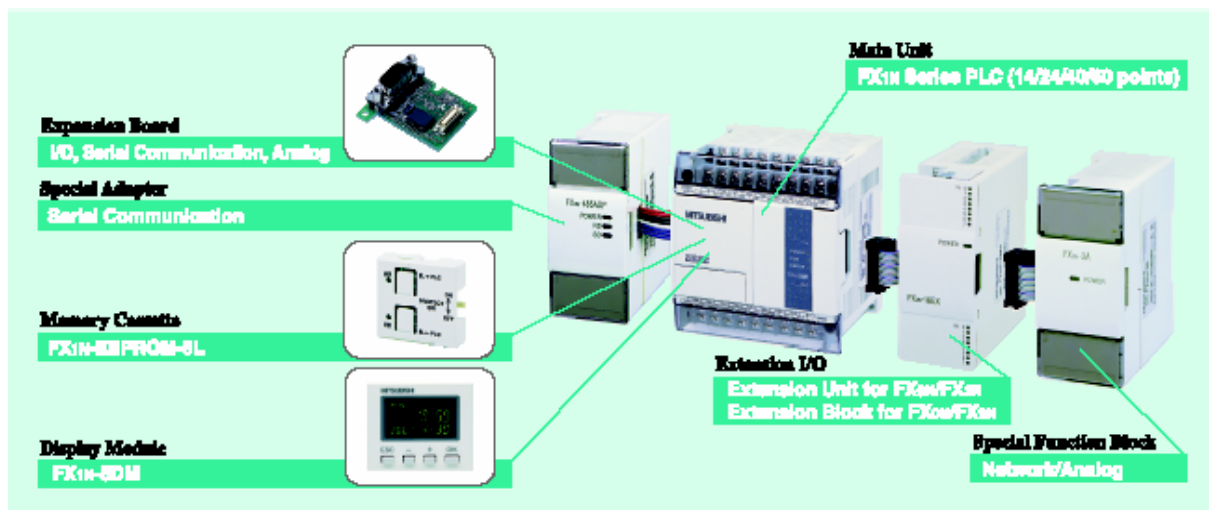
### 3. PLC loại FX1S

**FX1S** có khả năng quản lý số lượng I/O trong khoảng 10-34 I/O. Cũng giống như FXOS, FX1S không có khả năng mở rộng hệ thống. Tuy nhiên, FX1S được tăng cường thêm một số tính năng đặc biệt như: tăng cường hiệu năng tính toán, khả năng làm việc với các đầu vào ra tương tự thông qua các card chuyển đổi, cải thiện tính năng bộ đếm tốc độ cao, tăng cường 6 đầu vào xử lý, trang bị thêm các chức năng truyền thông trong mạng (giới hạn số lượng trạm tối đa là 8 trạm) hay giao tiếp với các bộ HMI đi kèm. FX1S thích hợp với các ứng dụng trong công nghiệp chế biến gỗ, đóng gói sản phẩm, điều khiển động cơ, máy móc, hay các hệ thống quản lý môi trường.

### 4. PLC loại FX1N

#### a. Giới thiệu:

**PLC FX1N** thích hợp với các bài toán điều khiển với số lượng đầu vào ra trong khoảng 14-60 I/O. Tuy nhiên khi sử dụng các module vào ra mở rộng, FX1N có thể tăng cường số lượng I/O lên tới 128 I/O. FX1N được tăng khả năng truyền thông, nối mạng, cho phép tham gia trong nhiều cấu trúc mạng khác nhau như Ethernet, Profibus, cc-Link, Canopen, Devicenet... FX1N có thể làm việc với các module analog, các bộ điều khiển nhiệt độ. Đặc biệt, FX1N được tăng cường chức năng điều khiển vị trí với 6 bộ đếm tốc độ cao, hai bộ phát xung đầu ra với tần số điều khiển tối đa là 100KHz. Điều này cho phép các bộ điều khiển lập trình thuộc dòng FX1N có thể cùng một lúc điều khiển một cách độc lập hai động cơ servo hay tham gia các bài toán điều khiển vị trí.

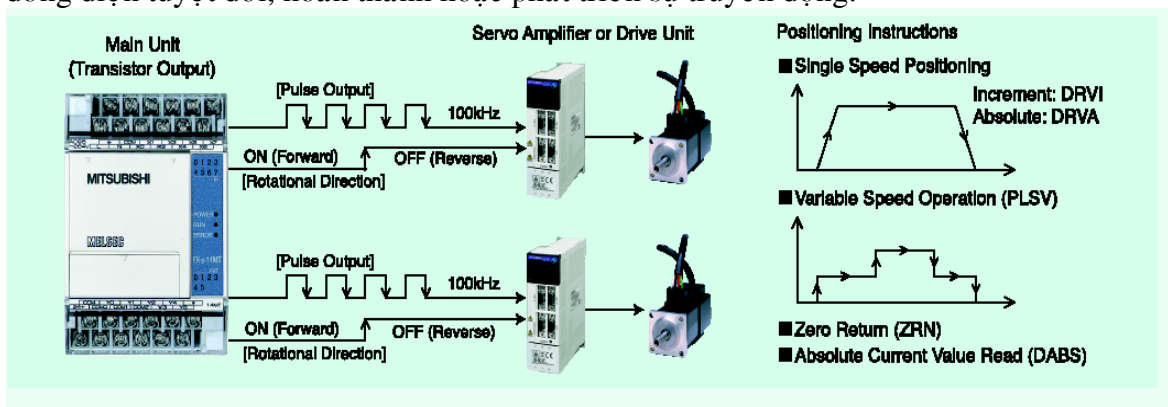


### b. Đặc điểm:

- Cơ cấu máy nhỏ gọn, chi phí thấp, module màn hình và khối mở rộng có hệ thống dễ dàng nâng cấp.
- Vận hành tốc độ cao đối với lệnh cơ bản tốc độ xử lý từ 0,55 đến 0,7 $\mu$ s/lệnh, đối với lệnh ứng dụng tốc độ xử lý từ 3,7 đến vài trăm  $\mu$ s/lệnh.
- Đặc tính kỹ thuật của bộ nhớ chất lượng và phong phú. Bộ nhớ EEPROM cho phép 8000 bước.
- Dây thiết bị dụng cụ đa năng như: role phụ trợ 1536 điểm, bộ đếm thì 256 điểm, bộ đếm 235 điểm, thanh ghi dữ liệu 8000 điểm.
- Những module chức năng đặc biệt: có đến hai dãy mở rộng của những module chức năng đặc biệt có thể được thêm vào cho những nhu cầu riêng.
- Dây mở rộng tự cung cấp điện: Độ biến thiên mở rộng của sự cung cấp điện AC có thể đáp ứng sự cung cấp điện áp từ bất kỳ nơi nào trên thế giới (100 đến 240V AC). Sự cung cấp dòng điện DC cũng được cho phép từ 12 đến 24 V DC.
- Quá trình điều khiển được tăng, sử dụng lệnh PID cho những hệ thống đòi hỏi sự điều khiển chính xác.
- Khả năng kết nối: Việc thực hiện hoàn chỉnh của những module kết nối sẽ làm cho thông tin và dữ liệu được cung cấp dễ dàng.
- Dễ dàng lắp đặt: sử dụng thanh DIN hoặc khoảng trống có sẵn.
- Đồng hồ thời gian thực tế: Sử dụng tiêu chuẩn đồng hồ thời gian thực tế cho những ứng dụng độc lập về thời gian.
- Phần mềm cơ bản: chương trình sẽ được chạy nhanh chóng và dễ dàng với phần mềm **GX Developer** hoặc **FX-PCS/WIN-E** Software.
- Tác vụ điểm kết nối: Tác vụ tại điểm kết nối riêng biệt khi kết nối một line, ta có thể liên kết với dữ liệu đã được cung cấp qua hệ thống.
- Bộ điện thế kế sử dụng tín hiệu Analog: dễ dàng thay đổi thiết bị định thời gian ở bộ điện thế kế ở màn hình phía trước.
- Vị trí và xung chức năng ngõ ra:

PLC có hai ngõ ra phát ra xung có tần số 100KHZ cùng một lúc.

PLC cung cấp 7 vị trí lệnh truyền kể cả quay trở về điểm zero, đọc giá trị dòng điện tuyệt đối, hoàn thành hoặc phát triển sự truyền động.

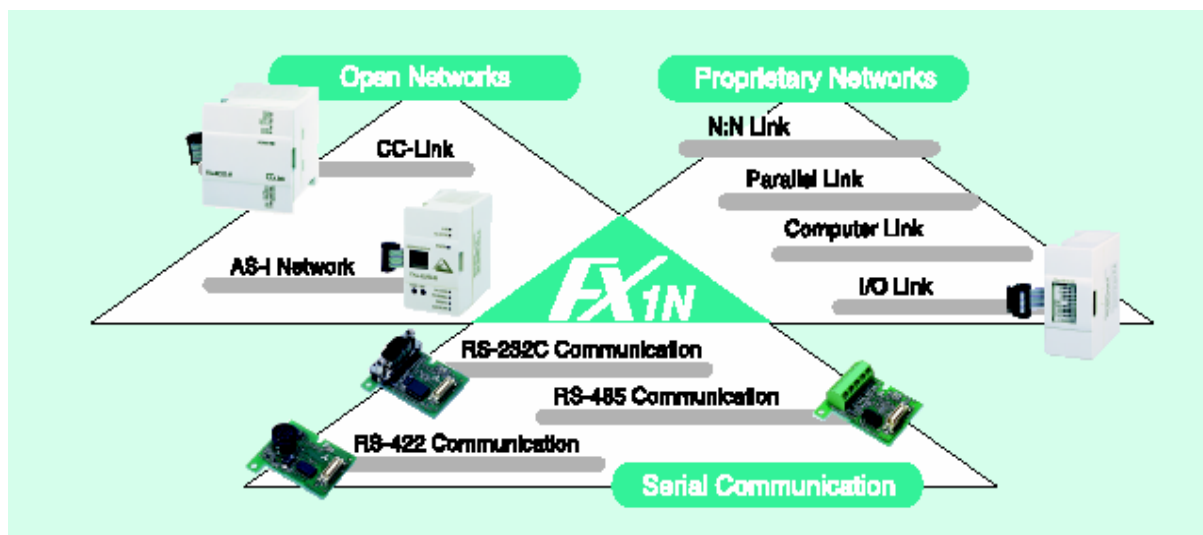


- Nâng cấp hệ thống bằng khối mở rộng hoặc kết nối module: Bảng mở rộng có thể được sử dụng để kết nối chức năng truyền thông bằng cách dùng bộ kết nối tương thích **RS-232C**, **RS-485** hoặc **RS-422** kết hợp với ngõ I/O bằng tín hiệu analog hoặc tín hiệu số.

- Kết nối module có thể quang sát qua cách sắp xếp các bộ định thì, bộ đếm, thanh ghi dữ liệu và có thể sử dụng để kết nối với các khối mở rộng.



- Mạng truyền thông: thông tin đa dạng và kết nối dữ liệu có thể được thực hiện bởi sự liên kết với các khối mở rộng hoặc các thiết bị tích hợp chuyên dùng được sử dụng cho FX2N.



## 5. PLC loại FX2N

Đây là một trong những dòng PLC có tính năng mạnh nhất trong dòng FX. FX2N được trang bị tất cả các tính năng của dòng FX1N, nhưng tốc độ xử lý được tăng cường, thời gian thi hành các lệnh cơ bản giảm xuống cỡ  $0.08\mu s$ . FX2N thích hợp với các bài toán điều khiển với số lượng đầu vào ra khoảng 16-128 I/O, trong trường hợp cần thiết FX2N có thể mở rộng đến 256 I/O. Tuy nhiên, trong trường hợp mở rộng số lượng I/O lên 256, FX2N sẽ làm mất lợi thế về giá cả và không gian lắp đặt của FX2N. Bộ nhớ của FX2N là 8kstep, bộ nhớ RAM có thể mở rộng đến 16kstep cho phép thực hiện các bài toán điều khiển phức tạp. Ngoài ra, FX2N còn được trang bị các hàm xử lý PID với tính năng tự chỉnh, các hàm xử lý số thực cùng đồng hồ thời gian thực tích hợp sẵn bên trong. Những tính năng vượt trội trên cùng với khả năng truyền thông, nối mạng nói chung của dòng FX1N đã đưa FX2N lên vị trí hàng đầu trong dòng FX, có thể đáp ứng tốt các đòi hỏi khắt khe nhất đối với các ứng dụng sử dụng trong các hệ thống điều khiển cấp nhỏ và trung bình. FX2N thích hợp với các bài toán điều khiển sử dụng trong các dây chuyền sản xuất, xử lý nước thải, các hệ thống xử lý môi trường, điều khiển các máy dệt, trong các dây chuyền đóng lắp ráp tàu biển.

## 6. PLC loại FX2NC

Bộ điều khiển lập trình với kích thước siêu gọn, thích hợp cho các ứng dụng đòi hỏi cao về yêu cầu tiết kiệm không gian lắp đặt. FX2NC có đầy đủ các tính năng của FX2N nhưng lại tiết kiệm đến 27% không gian sử dụng. Lĩnh vực ứng dụng chủ yếu của FX2NC là dùng trong xây dựng, trong các hệ thống bơm hay các bài toán điều khiển liên quan đến môi trường. 🇺🇸

# Chương II

## CÁC THIẾT BỊ VÀ LỆNH CƠ BẢN

### A. KHÁI NIỆM

#### I. KHÁI NIỆM MỘT CHƯƠNG TRÌNH

Chương trình là một chuỗi các lệnh nối tiếp nhau được viết theo một ngôn ngữ mà PC có thể hiểu được. Có 3 dạng chương trình : INSTRUCTION, LADDER, SFC/STL .

\_ *Instruction* : hệ thống gồm những dòng lệnh nhập liên tiếp nhau.

\_ *Ladder* : phương pháp xây dựng chương trình dạng đồ họa dùng các ký hiệu logic dạng role.

\_ *SFC / STL* : chương trình dạng lưu đồ.

Không phải tất cả các công cụ lập trình đều có thể làm việc được với cả 3 dạng trên. Nói chung bộ lập trình cầm tay chỉ làm việc được với dạng INSTRUCTION, trong khi hầu hết các công cụ lập trình đồ họa sẽ làm việc được với cả dạng INSTRUCTION và LADDER . Các phần mềm chuyên dùng sẽ cho phép làm việc với dạng SFC.

Dạng Instruction

Dạng Ladder

Dạng SFC / STL

#### II. CÁC THIẾT BỊ CƠ BẢN DÙNG TRONG LẬP TRÌNH

Có 6 thiết bị cơ bản dùng trong lập trình. Mỗi thiết bị có công dụng riêng. Để dễ dàng xác định thì mỗi thiết bị được gán cho một ký tự :

- X : dùng chỉ ngõ vào vật lý gắn trực tiếp vào PC.
- Y : dùng chỉ ngõ ra nối trực tiếp từ PC.
- T : dùng để xác định thiết bị định thì có trong PC.
- C : dùng để xác định thiết bị đếm có trong PC.
- M và S : dùng như là các cờ hoạt động bên trong PC.

Tất cả các thiết bị trên được gọi là “ thiết bị Bit “ , nghĩa là các thiết bị này có hai trạng thái ON hoặc OFF , mức 1 hoặc mức 0 ( mức 0 là ở trạng thái tĩnh mức 1 là ở trạng thái động).

### III . CÁCH ĐỌC LOGIC CỦA CHƯƠNG TRÌNH LADDER

Logic Ladder rất gần với logic role cơ bản. Các công tắc và cuộn dây được kết nối và điều khiển theo nhiều dạng mạch khác nhau, tuy vậy, nguyên tắc cơ bản không có gì khác.

Cuộn dây có thể được dùng để điều khiển trực tiếp ngõ ra từ PC ( ví dụ thiết bị Y ), hoặc nó có thể điều khiển bộ định thì, bộ đếm, hoặc cờ ( như là thiết bị T, C, M, S). Mỗi cuộn dây gắn với các công tắc hoặc tiếp điểm, các công tắc hoặc tiếp điểm này có thể là thường mở (NO – normal open ) hoặc thường đóng ( NC – normal close).

Thuật ngữ “thường” nói đến trạng thái các công tắc khi cuộn dây không có điện. Nếu role được dùng và cuộn dây giả sử là OFF, một công tắc NO sẽ không có điện, nghĩa là tải được nối với công tắc NO sẽ không hoạt động, còn công tắc NC thì có dòng điện đi qua, vì vậy tải được nối với công tắc sẽ hoạt động.

Khi cuộn dây được kích hoạt thì trạng thái của công tắc bị đảo ngược, nghĩa là có dòng trong công tắc NO và công tắc NC lại ngăn không cho dòng điện đi qua.

Các ngõ vào vật lý nối đến bộ điều khiển lập trình ( thiết bị X) không có cuộn dây có thể lập trình. Các thiết bị này chỉ có thể được dùng ở dạng công tắc (nhận tín hiệu) mà thôi loại tiếp điểmNO và NC

*Thí dụ :*

Vì có sự kết hợp với role, các chương trình ladder có thể được đọc như dòng điện đi từ trái sang phải. Dòng này phải qua một loạt các tiếp điểm như X0 và X1 để bật cuộn dây ngõ ra Y0 lên ON. Do đó, trong ví dụ sau, nếu X0 được bật ON thì ngõ ra Y0 cũng được bật lên ON. Tuy nhiên, nếu công tắc hành trình X1 được kích hoạt thì ngõ ra Y0 bị tắt OFF vì sự nối kết giữa cột trái và cột phải bị đứt, nghĩa là không có dòng điện đi qua.



## B. CÁC TẬP LỆNH CƠ BẢN CỦA LẬP TRÌNH INSTRUCTION VÀ LADDER

### 1. TẬP LỆNH LOAD và LOAD INVERSE

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>LD</b> (LoaD)	Có nhiệm vụ logic khởi tạo loại công tắc NO. Nối trực tiếp đầu bên trái của mạch.		X, Y, M, S, T, C
<b>LDI</b> (LoaD Inverse)	Có nhiệm vụ logic khởi tạo loại công tắc NC. Nối trực tiếp đầu bên trái của mạch.		X, Y, M, S, T, C

Chương trình ví dụ :

### 2. TẬP LỆNH OUT

lệnh	Chức năng	Dạng mẫu	Thiết bị
------	-----------	----------	----------

<b>OUT</b> (OUT)	-Điều khiển cuộn dây. -Nối trực tiếp vào đầu bên phải của mạch. -Nhiều lệnh OUT có thể được nối song song. -Không thể điều khiển thiết bị ngõ vào loại 'X'.		Y, M, S, T, C
---------------------	--	--	---------------------

### 3. TẬP LỆNH AND, AND INVERSE

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>AND</b> (AND)	Nối tiếp các công tắc NO (thường mở), có thể nối tiếp nhiều công tắc cùng một lúc.		X, Y, M, S, T, C.
<b>ANI</b> (ANd Inverse)	Nối tiếp các công tắc NC (thường đóng), có thể nối tiếp nhiều công tắc cùng một lúc.		X, Y, M, S, T, C.

Chương trình ví dụ :

*Lưu ý :*

6 Việc thêm một cuộn dây qua một công tắc thì lệnh OUT đầu tiên ( OUT Y4) được gọi là ngõ ra “follow-on”. Các ngõ ra Follow-on cho phép xử lý ngõ ra theo đúng trình tự đã ghi.

7 Mặc dù không có giới hạn số công tắc mắc song song, nhưng một số bảng điều khiển lập trình, màn hình và máy in không thể nào hiển thị hoặc in chương trình nếu nó vượt quá giới hạn của phần cứng. Mỗi dòng hay mỗi nhánh của chương trình Ladder nên chứa tối đa 10 công tắc và 1 cuộn dây. Số ngõ ra Follow-on nên giới hạn tối đa là 24.

### 4. TẬP LỆNH OR , OR INVERSE

<b>lệnh</b>	<b>Chức năng</b>	<b>Dạng mẫu</b>	<b>Thiết bị</b>
<b>OR</b> (OR)	Nối song song các công tắc NO (thường mở). Tối đa là 10 nhánh nối song song cho 1 cuộn dây.		X, Y, M, S, T, C.
<b>ORI</b> (OR Inverse)	Nối song song các công tắc NC (thường đóng). Tối đa là 10 nhánh nối song song cho 1 cuộn dây.		X, Y, M, S, T, C.

\**Lưu ý* : Một bên của lệnh OR/ORI luôn nối với đầu bên trái.

Chương trình ví dụ :

## 5. TẬP LỆNH OR BLOCK

<b>Tên lệnh</b>	<b>Chức năng</b>	<b>Dạng mẫu</b>	<b>Thiết bị</b>
<b>ORB</b> (OR Block)	Nối song song nhiều mạch công tắc.		Không có

### \* *Đặc điểm* :

\_ Lệnh ORB là lệnh độc lập và không kết hợp với bất kỳ thiết bị nào hay con số nào.

\_ Lệnh ORB được dùng để nối song song nhiều mạch công tắc (các khối nối tiếp) với khối phía trước. Các khối nối tiếp là các khối có nhiều công tắc nối tiếp nhau, hay dùng trong lệnh ANB.

\_ Để khai báo điểm bắt đầu của một khối, ta dùng lệnh LD hay LDI. Sau một khối nối tiếp, ta nối nó vào khối trước bằng lệnh ORB.

\* *Lưu ý* :

- 1 Dùng lệnh ORB theo lô không dùng quá 8 lệnh LD và LDI nhằm để tránh lỗi chương trình.
- 2 Dùng lệnh ORB trong mạch tuần tự không có giới hạn số mạch mắc song song.

Chương trình ví dụ :

lô	Lập trình tuần tự				Lập trình theo			
	0	LD	X	0	0	LD	X	0
	1	AND	X	1	1	AND	X	1
	2	LD	X	2	2	LD	X	2
	3	AND	X	3	3	AND	X	3
	4	<b>ORB</b>			4	LDI	X	4
	5	LDI	X	4	5	AND	X	5
	6	AND	X	5	6	<b>ORB</b>		
	7	<b>ORB</b>			7	<b>ORB</b>		
	8	OUT	Y	6	8	OUT	Y	6

## 6. TẬP LỆNH AND BLOCK :

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>ANB</b> (AND Block )	Nối tiếp các mạch song song.		Không có

Chương trình ví dụ :

Dùng phương pháp lập trình tuần tự

0	LD	X	0
1	OR	X	1
2	LD	X	2
3	AND	X	3
4	LDI	X	4
5	AND	X	5
6	<b>ORB</b>		
7	OR	X	6
8	<b>ANB</b>		
9	OR	X	3
10	OUT	Y	7

**\* Đặc điểm :**

- \_ Lệnh ANB là lệnh độc lập và không kết hợp với bất kỳ thiết bị nào hay con số nào.
- \_ Lệnh ANB được dùng để nối tiếp nhiều mạch công tắc (thường là các khối song song) với khối phía trước. Các khối song song là các khối có nhiều công tắc nối song song nhau, hay dùng trong lệnh ORB.
- \_ Để khai báo điểm bắt đầu của một khối ta dùng lệnh LD hay LDI. Sau một khối nối tiếp, ta nối nó vào khối trước bằng lệnh ANB.
- \_ Khi dùng ANB theo lô, không dùng quá 8 lệnh LD và LDI để tránh lỗi chương trình.
- \_ Lệnh ANB có thể dùng nhiều lần để nối tiếp các mạch song song với khối trước nó.

#### 7. TẬP LỆNH MPS, MRD, MPP :

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>MPS</b> (Point Store)	Lưu kết quả hiện hành của các tác vụ trong PC.	MPS	Không có
<b>MRD</b> (Read)	Đọc kết quả hiện hành của các tác vụ trong PC.	MRD	Không có
<b>MPP</b> (PoP)	Gọi kết quả đã lưu và loại bỏ nó.	MPP	Không có

#### \* Cách dùng các tập lệnh MPS, MRD, MPP :

Khi chương trình được viết dưới dạng Ladder thì tất cả các lệnh MPS, MRD, MPP sẽ tự động được thêm vào khi thực hiện chuyển chương trình (program conversion) sang dạng Instruction . Nếu xem chương trình dạng Instruction sau khi chuyển ta sẽ thấy có các lệnh MPS, MRD, MPP.

Khi chương trình được viết ở dạng Instruction, người sử dụng phải nhập vào toàn bộ tất cả các lệnh MPS, MRD, MPP.

**\* Đặc điểm tập lệnh MPS, MRD, MPP:**

- Được dùng để nối các ngõ vào bên trái của các cuộn dây với bất kỳ ngõ vào bên trái của công tắc nào. Nếu không có tập lệnh này thì ta chỉ có thể nối cuộn dây vào bên phải của công tắc cuối cùng.
- MPS lưu lại điểm nối của mạch ladder, do đó nó có thể được gọi lại nhiều lần để nối nhiều nhánh cuộn dây.
- MRD có chức năng gọi lại hoặc đọc dữ liệu của điểm nối đã lưu trước đó và nối nó với công tắc tiếp theo.
- MPP có chức năng gọi điểm nối đã lưu (từ vùng lưu tạm) ra và loại bỏ nó khi công tắc kế tiếp nối vào điểm này.
- Đối với mỗi lệnh MPS thì bắt buộc phải có một lệnh MPP tương ứng.
- Lệnh MPP được dùng để nối mạch công tắc hay cuộn dây cuối cùng.
- Ở bất kỳ bước lập trình nào, số cặp lệnh MPS – MPP hoạt động không được vượt quá con số 11.

**Các chương trình ví dụ :**

*2 Ví dụ 1 :*

0	LD	X	0
1	<b>MPS</b>		
2	LD X	1	
3	OR X	2	
4	ANB		
5	OUT Y	0	
6	<b>MRD</b>		
7	LD	X	3
8	AND X	4	
9	LD X	5	
10	AND X	6	
11	ORB		
12	ANB		
13	OUT	Y	1
14	<b>MPP</b>		
15	AND X	7	
16	OUT Y	2	
17	LD X	10	
18	OR X	11	
19	ANB		
20	OUT	Y	3

\* Ví dụ 2 :

0	LD	X	0
1	<b>MPS</b>		
2	AND	X	1
3	<b>MPS</b>		
4	AND	X	2
5	OUT	Y	0
6	<b>MPP</b>		
7	AND	X	3
8	OUT	Y	1
9	<b>MPP</b>		
10	AND	X	4
11	<b>MPS</b>		
12	AND	X	5
13	OUT	Y	2
14	<b>MPP</b>		
	15	AND	X
6			16
OUT	Y	3	

### TẬP LỆNH SET và RESET

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>SET</b>	Đặt một thiết bị (bit) lên chế độ ON vĩnh viễn.		Y, M, S
<b>RESET</b>	Đặt một thiết bị (bit) xuống chế độ OFF vĩnh viễn.		Y, M, S, D, V, Z

\* Đặc điểm :

\_ SET và RESET có thể dùng cho cùng một thiết bị bao nhiêu lần tùy ý. Tuy nhiên trạng thái cuối cùng mới là trạng thái tác động.

Ví dụ :

0	LD	X	0
1	SET	Y	0
	2	LD	X
1			3
RST	Y	0	
4	LD	X	2
	5	SET	M
0			6
LD	X	3	
7	RST	M	0
	8	LD	X
4			9
SET	S	0	
10	LD	X	5
	11	RST	S
0			12
LD	X	6	
13	SET	D	0

*Nhận xét :*

- Một khi X0 đã bật ON thì Y0 hoạt động và duy trì trạng thái ON ngay cả khi X1 đã tắt OFF.
- Khi X1 bật ON thì Y0 sẽ tắt OFF và duy trì trạng thái OFF ngay cả sau khi X1 tự nó chuyển thành OFF.
- Quá trình xảy ra tương tự cho các M0, S0, D0.

## 8. BỘ ĐỊNH THÌ và BỘ ĐẾM

*a) Dạng chung OUT và RESET của bộ định thì và bộ đếm :*

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
OUT	Điều khiển cuộn dây bộ định thì hoặc bộ đếm.		T, C
RST	Đặt lại giá trị tác động cho bộ định thì và bộ đếm.		Y, M, S, D, V, Z

*b) Bộ định thì (Timer) :*



*Ký hiệu* : T

*Công dụng* : dùng để định các khoảng thời gian.

*Cách đánh số* : đánh số bằng chữ số thập phân ( ví dụ : T0 ; T1 ; T2 ; .....)

*Ví dụ cách dùng bộ định thì :*

● **Hoạt động chung của bộ định thì :**

Các bộ định thì hoạt động bằng cách đếm các xung clock (xung 1 ; 10 ; 100 mili giây). Ngõ ra của bộ định thì được kích hoạt khi giá trị đếm được đạt đến giá trị hằng số K. Khoảng thời gian trôi qua được tính bằng cách lấy giá trị đếm được nhân với độ phân giải của bộ định thì.

*Ví dụ :*

Bộ định thì loại 10 mili giây đếm được đến giá trị là 567, khi đó khoảng thời gian trôi qua được tính như sau :

$$567 \times 10 \text{ mili giây} = 567 \times 0,01 \text{ giây} = 5,67 \text{ giây.}$$

Khoảng thời gian định thì được đặt trực tiếp thông qua hằng số K, hoặc gián tiếp thông qua thanh ghi dữ liệu (D). Thường dùng thanh ghi dữ liệu được chốt để đảm bảo không bị mất dữ liệu khi mất điện. Tuy nhiên nếu điện áp của nguồn pin giảm quá mức thì thời gian định thì có thể bị sai.

● **Độ chính xác của bộ định thì :**

Độ chính xác của bộ định thì có thể bị ảnh hưởng do cách lập trình. Có hai trường hợp có thể ảnh hưởng đến độ chính xác của bộ định thì :

\* **Trường hợp 1 :** Công tắc của bộ định thì đặt phía sau cuộn dây kích của bộ định thì.

Sai số định thì lớn nhất :

$$2 \times \text{chu kỳ quét} + \text{thời gian bộ lọc ngõ vào}$$

Sai số định thì nhỏ nhất :

$$\text{Thời gian bộ lọc ngõ vào} - \text{độ phân giải bộ định thì}$$

\* **Trường hợp 2 :** Công tắc bộ định thì được đặt trước cuộn dây kích của bộ định thì, khi đó sai số của bộ định thì sẽ lớn hơn.

Sai số định thì lớn nhất :

$$3 \times \text{chu kỳ quét} + \text{thời gian bộ lọc ngõ vào}$$

Sai số định thì nhỏ nhất :

*Thời gian bộ lọc ngõ vào – độ phân giải bộ định thì*

Sai số trung bình cho cả hai trường hợp :

$1,5 \times$  chu kỳ quét chương trình

**c) Bộ đếm (Counter) :**

- Ký hiệu : C
- Công dụng : đếm các sự kiện
- Cách đánh số : thập phân
- Phân loại : Bộ đếm 16 bit.  
Bộ đếm 32 bit.  
Bộ đếm tốc độ cao.
- Ví dụ cách dùng bộ đếm :

- Số lượng bộ đếm có thể dùng :

Xem cụ thể loại PLC là loại nào và tra vào bảng thông số kỹ thuật.

**d) Các thông số cho bộ định thì và bộ đếm :**

Khi lệnh OUT áp dụng cho bộ định thì hoặc bộ đếm thì phải nhập vào một hằng số. Hằng số đó bắt đầu bằng ký tự “K”.

Trong trường hợp bộ định thì, hằng số K chỉ giá trị khoảng thời gian định thì; nghĩa là nếu bộ định thì loại 100 mili giây có hằng số “K100” sẽ định thì một khoảng thời gian là :  $100 \times 100$  mili giây = 10 giây trước khi cuộn dây bộ định thì được kích hoạt.

Khi dùng bộ đếm, hằng số K xác định số cần đếm. ( Ví dụ : một bộ đếm với hằng số K8 sẽ phải được kích 8 lần trước khi cuộn dây bộ đếm có điện).

Bảng thông số của các loại bộ định thì và bộ đếm :

Bộ định thì / Bộ đếm		Giá trị đặt hằng số K cho phép	Giá trị thực sự của hằng số K
Bộ định thì	1 mili giây		$0.001 \div 32.767$ giây
	10 mili giây		$0.01 \div 327.67$ giây
	100 mili giây		$0.1 \div 3276.7$ giây
Bộ đếm	16 bit	$1 \div 32,767$	$1 \div 32,767$
	32 bit	$-2,147,483,648 \div 2,147,483,647$	$-2,147,483,648 \div 2,147,483,647$

**9. NGÕ VÀO / RA (Input / Output)**

**a) Ngõ vào :**

- *Kí hiệu* : X
- *Công dụng* : biểu diễn các ngõ vào vật lý trên bộ điều khiển.
- *Kí hiệu khác* : I/P  
Inp  
Ngõ vào (X)  
Công tắc ngõ vào
- *Hình thức sử dụng* : công tắc thường mở NO (Normal Open),  
và thường đóng NC (Normal Close)
- *Cách đánh số* : dùng số bát phân, nghĩa là X0 ÷ X7 ; X10 ÷ X17.
- *Ví dụ cách dùng ngõ vào* :

**b) Ngõ ra :**

- *Kí hiệu* : Y
- *Công dụng* : biểu diễn các ngõ ra vật lý trên bộ điều khiển.
- *Kí hiệu khác* : O/P  
Otp  
Ngõ ra (Y)  
Cuộn dây / role / công tắc ngõ ra.
- *Các hình thức sử dụng* : công tắc thường mở NO, thường đóng NC và các cuộn dây ngõ ra.
- *Cách đánh số* : dùng số bát phân, nghĩa là Y0 ÷ Y7 ; Y10 ÷ Y17.
- *Ví dụ cách dùng ngõ ra* :

**Số lượng các ngõ vào/ra :**

<b>Loại bộ điều khiển</b>	<b>Số ngõ vào tối đa</b>	<b>Số ngõ ra tối đa</b>	<b>Tổng số ngõ vào/ra</b>
<b>FX<sub>0(S)</sub></b>	Phụ thuộc vào việc chọn Base Unit		30
<b>FXon</b>	84	64	128
<b>FX</b>	128	128	256
<b>FX<sub>(2C)</sub></b>	256 ( khả năng định được trên phần mềm)	256 (khả năng định được trên phần mềm)	512 ( tổng số định được trên phần cứng hay trên phần mềm)

## 11. ROLE PHỤ TRỢ (auxiliary relays)

- *Kí hiệu* : M
- *Công dụng* : Cờ nhớ trạng thái của bộ điều khiển.
- *Kí hiệu khác* : Cuộn dây, role, công tắc, cờ phụ trợ.  
Cuộn dây, role, công tắc, cờ M.  
Thiết bị (bit) M.
- *Các hình thức sử dụng* : công tắc thường mở NO, thường đóng NC và các cuộn dây ngõ ra.
- *Cách đánh số* : dùng số thập phân, nghĩa là M0 ÷ M9 ; M10 ÷ M19.
- *Phân loại* : Role phụ trợ ổn định trạng thái.  
Role phụ trợ có nguồn pin nuôi / được chốt.  
Role phụ trợ chuẩn đoán chuyên dùng.
- *Ví dụ cách dùng role phụ trợ* :

### 1. Role phụ trợ ổn định trạng thái :

Các PC đều có role phụ trợ. Cuộn dây của các role này được điều khiển bởi các công tắc tương tự như cách điều khiển role ngõ ra.

Tất cả các role phụ trợ đều có một số công tắc thường mở và thường đóng nối với PC nếu cần. Lưu ý các công tắc này không thể điều khiển trực tiếp tải ngoài, chỉ có role ngõ ra mới có thể làm được điều này.

### 2. Role phụ trợ có nguồn pin nuôi / được chốt :

Nếu xảy ra sự cố về điện trong khi PC đang chạy thì tất cả trạng thái của các role ngõ ra và các role công dụng chung đều bị RESET (OFF). Khi PC hoạt động trở lại, nếu như ta không kích lại các role này thì chúng sẽ vẫn giữ trạng thái OFF .

Role phụ trợ có nguồn pin nuôi (được chốt) là loại role mà trạng thái của chúng không bị thay đổi khi bị mất điện. Trạng thái này được duy trì bởi nguồn pin bên trong hay dùng bộ nhớ EEPROM được gắn vào PC. Role được chốt bởi tập lệnh SET (hay RESET) dùng để duy trì trạng thái tạm thời của role.

**Dạng mạch tự nhớ**

**Dạng mạch được chốt**

Ở dạng mạch tự nhớ, role M507 được duy trì điện nhờ vào tiếp điểm tự giữ của nó. Nếu như X1 là ON thì M507 bị Reset.

Ở dạng mạch được chốt bởi lệnh SET, role M507 sẽ không bị Reset khi X1 là ON mà nó sẽ vẫn được duy trì trạng thái, nó chỉ bị Reset khi X2 là ON.

• *Lưu ý : các tải ngoài*

Các role phụ trợ được dùng kết hợp với các công tắc thường mở và thường đóng. Các công tắc này có thể dùng tùy ý khi lập trình. Các công tắc này không thể dùng để điều khiển trực tiếp các tải ngoài. Tất cả các tải ngoài nên được điều khiển thông qua việc dùng các ngõ ra trực tiếp.

### 3. Các role phụ trợ chuẩn đoán chuyên dùng :

Một PC có một số các role phụ trợ chuyên dùng. Các role này đều có chức năng chuyên biệt, về mặt sử dụng chia làm hai loại sau :

a) *Công tắc role phụ trợ chuyên dùng :*

Role này được điều khiển tự động bởi PC, người sử dụng không thể can thiệp. Các công tắc này có thể được sử dụng trong chương trình để chuẩn đoán một tình trạng nào đó.

*Ví dụ :*

M8000 : báo RUN ( ON khi PC đang trong trạng thái chạy chương trình).

M8002 : xung khởi động (ON khi PC khởi động, PC chuyển từ OFF → ON).

M8012 : xung clock 100 mili giây.

b) *Điều khiển những cuộn dây role chuyên dùng :*

Khi người sử dụng SET các cuộn dây này, PC sẽ thực thi một tác vụ chuyên biệt đã được xác định trước.

*Ví dụ :*

M8033 : tất cả các trạng thái ngõ ra được duy trì khi PC ngừng hoạt động.

M8034 : tất cả các ngõ ra đều bị vô hiệu.

M8039 : PC hoạt động ở chế độ quét thời hằng.

## 12. ROLE TRẠNG THÁI (state relays)

- *Kí hiệu* : S
- *Công dụng* : Cờ trạng thái trong bộ điều khiển.
- *Kí hiệu khác* : Cuộn dây, role, công tắc, cờ trạng thái.  
Cuộn dây, role, công tắc, cờ S.  
Cuộn dây, role, công tắc, cờ bước STL.  
Cờ hiệu.

- *Hình thức sử dụng* : công tắc thường mở NO, thường đóng NC, các cuộn dây ngõ ra
- *Cách đánh số* : dùng số thập phân, nghĩa là M0 ÷ M9 ; M10 ÷ M19.
- *Phân loại* : Role ổn định trạng thái chung – role trạng thái.  
Role trạng thái có nguồn pin nuôi ( được chốt )  
Role bước STL  
Cờ hiệu.
- *Ví dụ cách dùng role trạng thái* :

### 1. Role trạng thái ổn định :

Các role này được điều khiển bởi các công tắc trong PC giống như việc điều khiển các ngõ ra. Tất cả các role trạng thái đều có một số các công tắc vật lý thường mở và thường đóng nối với PC nếu cần. Các công tắc này không thể điều khiển trực tiếp tải ngoài, ngoại trừ các role ngõ ra.

#### 1. Role trạng thái có nguồn pin nuôi :

Nếu nguồn cấp điện bị hỏng khi PC đang hoạt động thì tất cả role ngõ ra và role công dụng chung đều bị RESET. Tất cả các role này sẽ vẫn ở trạng thái OFF nếu như không được kích hoạt lại trạng thái khi PC hoạt động lại, đây gọi là role trạng thái được chốt. Trạng thái chốt được duy trì bởi nguồn pin nuôi bên trong hay dùng bộ nhớ EEPROM gắn vào PC. Hoạt động của role trạng thái được chốt này tương tự như role phụ trợ được chốt đã đề cập phần trên.

• *Lưu ý* : Giống như role phụ trợ, role trạng thái cũng không thể điều khiển trực tiếp các tải ngoài mà phải điều khiển bằng role ngõ ra trực tiếp.

• *Số lượng role trạng thái* :

Loại role trạng thái	Số lượng có trên PC	Tổng số
Role trạng thái thường	Không có	
Role trạng thái được chốt	128 ( từ S0 ÷ S127 )	

### 2. Role trạng thái bước STL :

Các role trạng thái rất quan trọng khi lập trình điều khiển trình tự và được dùng kết hợp với lệnh STL.

Khi lập trình STL thì từng trạng thái có tác vụ tương ứng sẽ được xác định trước.

Bước trạng thái trước đó sau khi đã chuyển trạng thái vẫn duy trì trạng thái hoạt động cho đến khi hết chu kỳ quét hiện hành.

## 13 . XUNG CẠNH LÊN VÀ XUNG CẠNH XUỐNG

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>PLS</b> (PuLSe)	Kích xung khi có cạnh tác động lên		Y, M ( <i>Lưu ý</i> : không được dùng cuộn M chuyên dùng).
<b>PLF</b> (PuLse Falling)	Kích xung khi có cạnh tác động xuống		Y, M ( <i>Lưu ý</i> : không được dùng cuộn M chuyên dùng).

Chương trình ví dụ :

--

**\* Đặc điểm :**

\_ Khi lệnh PLS được thi hành, các thiết bị Y, M sẽ hoạt động trong khoảng thời gian một chu kỳ sau khi tín hiệu ngõ vào đã bật ON.

\_ Khi lệnh PLF được thi hành, các thiết bị Y, M sẽ hoạt động trong khoảng thời gian một chu kỳ sau khi tín hiệu ngõ vào đã tắt OFF.

## 14 . LỆNH END

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>END</b> (END)	Buộc chương trình kết thúc		Không có

--	--	--	--

Khi đặt lệnh END trong chương trình có tác dụng buộc kết thúc quá trình quét chương trình hiện hành và tiến hành cập nhật các ngõ vào / ra, các bộ định thì. Thực hiện cập nhật các ngõ vào ở đầu chu kỳ quét và cập nhật các ngõ ra ở cuối chu kỳ quét.

Việc quét chương trình là quá trình xử lý từng lệnh trong chương trình từ đầu đến cuối. Khoảng thời gian này gọi là thời gian quét, phụ thuộc vào độ dài và sự phức tạp của chương trình. Ngay khi dòng quét hiện hành được hoàn tất thì dòng quét tiếp theo sẽ bắt đầu ngay. Toàn bộ quá trình là một chu kỳ liên tục.

Chèn lệnh END vào giữa chương trình giúp tìm lỗi chương trình vì phần sau lệnh END bị vô hiệu và cách ly khỏi vùng kiểm lỗi.

## 15. LỆNH NO- OPERATION

Tên lệnh	Chức năng	Dạng mẫu	Thiết bị
<b>NOP</b> (No OPeration)	Không tác vụ, hay bước rỗng	Không có	Không có

### \* Đặc điểm :

Thêm lệnh NOP trong chương trình nhằm giảm thiểu sự thay đổi số bước chương trình khi ta soạn thảo hoặc thay đổi chương trình.

Có thể thay đổi sự hoạt động của một mạch bằng cách thay lệnh đã lập trình bằng lệnh NOP.

Thay lệnh LD, LDI, ANB, ORB bằng lệnh NOP có thể gây ra lỗi ở nhiều nơi trong chương trình.



# Chương III

## LẬP TRÌNH STL

### (CHƯƠNG TRÌNH TUẦN TỰ )

#### I. KHÁI NIỆM

Lập trình STL ( SStep Ladder) là một trong những kiểu lập trình cơ bản có trong tất cả các họ FX. Lập trình STL rất đơn giản và dễ hiểu, cung cấp cho người sử dụng một kỹ thuật lập trình mạnh nhất. Sự hiệu quả của lập trình STL là nó cho phép người lập trình tạo ra một chương trình thực thi gần giống như ý đồ của người lập trình. Chương trình STL không hoàn toàn thay thế chương trình Ladder chuẩn.

STL thể hiện mặt mạnh của nó thông qua việc tổ chức một chương trình lớn thành những phần tử nhỏ hơn. Mỗi phần tử này gọi là một trạng thái hoặc một bước. Để nhận biết các trạng thái, mỗi trạng thái được gán một ký hiệu xác định duy nhất.

Mỗi trạng thái hoàn toàn tách biệt với các trạng thái khác trong chương trình. Cách tốt nhất để hình dung điều này là mỗi trạng thái xem như là một chương trình tách biệt và người sử dụng đặt từng phần chương trình này lại với nhau theo thứ tự hoạt động củ nó. Ngoài ra các trạng thái này có thể được dùng lại nhiều lần và theo thứ tự khác nhau. Điều này tiết kiệm được thời gian lập trình và hạn chế các lỗi lập trình.

#### II. KHỞI ĐỘNG VÀ KẾT THÚC CHƯƠNG TRÌNH STL

##### 1. Khởi động chương trình tuần tự:

Để bắt đầu một lập trình tuần tự STL, ta phải bật trạng thái khởi tạo lên ON sao cho thích hợp. Có nhiều phương pháp điều khiển trạng thái khởi tạo.

Các cuộn dây trạng thái khởi tạo STL có thể được kích bằng xung, dùng lệnh SET hay với lệnh OUT. Thông thường ta chỉ dùng lệnh SET để chọn các trạng thái. Khi một trạng thái đã được kích hoạt thì có nghĩa là nó được đặt lên ON ( hình a).

Đối với một chương trình STL được kích hoạt khi vừa cấp điện cho bộ điều khiển, ta có thể dùng role phụ trợ M8002 để điều khiển việc xác lập trạng thái khởi tạo ( hình b). Khi chương trình STL đã được khởi tạo, CPU sẽ xử lý tất cả lệnh bên trong chương trình STL.

Hình a

Hình b

## 2. Kết thúc chương trình tuần tự:

Bằng cách đặt lệnh RET (tập lệnh RETurn) ở câu lệnh cuối cùng của trạng thái STL cuối cùng, khi đó lệnh này sẽ trả sự điều khiển về cho Ladder . Sau đó, mạch khởi tạo đã lập trình sẽ được xử lý đúng như phần chương trình Ladder và không còn nằm trong phạm vi hoạt động trạng thái STL cuối cùng.

## III . LẬP BƯỚC GIỮA CÁC TRẠNG THÁI

Để kích hoạt một trạng thái STL, trước tiên người sử dụng phải tác động cuộn dây STL. Việc tác động lên cuộn dây đó giống như cách khởi động một chương trình STL, hay gọi là kích trạng thái khởi tạo.

Tuy nhiên dùng lệnh OUT và lệnh SET để kích cuộn dây STL có sự khác biệt nhau

### 1. Dùng SET để kích cuộn dây STL :

SET được dùng để kích trạng thái STL làm cho nó hoạt động. Khi trạng thái STL1 hiện hành kích hoạt tiếp trạng thái STL2 thì cuộn dây của STL1 bị Reset. Như vậy, chỉ cần dùng SET kích hoạt một trạng thái, các trạng thái kế tiếp được kích lại một cách tự động.

Lệnh SET được dùng để kích một trạng thái nằm trong lưu trình STL, các trạng thái có liên kết với trạng thái STL hiện hành.

Lệnh SET được dùng để kích trạng thái STL thấp hơn trạng thái STL hiện hành ngay sau đó.

Để Reset một trạng thái STL, ta dùng lệnh RST hoặc ZRST ( ZRST : Zone ReSeT sẽ reset một chuỗi các ngõ ra nằm trong lệnh này). Tuy nhiên, lệnh ZRST chỉ Reset được trạng thái STL sau khi các trạng thái STL đã được thực hiện xong hết một chu kỳ của nó. Đây cũng chính là một nhược điểm của lệnh ZRST.

## 2. Dùng lệnh OUT để kích cuộn dây STL :

Lệnh OUT có tính năng giống như lệnh SET.

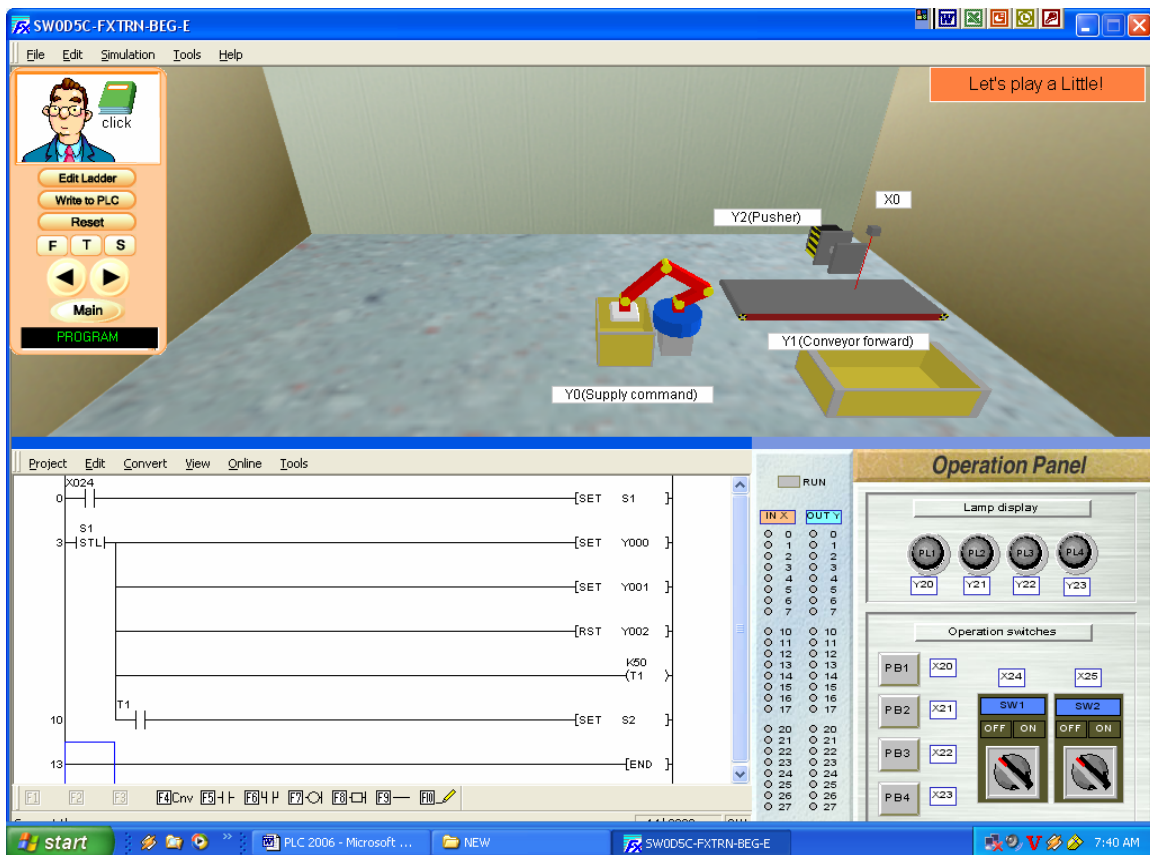
Tuy nhiên có một điều lệnh OUT khác lệnh SET là lệnh OUT có thể cho phép người lập trình được nhảy cách khoảng, bỏ qua nhiều trạng thái STL.

## IV . CẤU TRÚC CỦA MỘT BƯỚC TRONG STL

Trong chương trình tuần tự sẽ có nhiều bước, nhưng mỗi bước đưa ra cần theo đúng cấu trúc của một bước như sau:

1. Chỉ thị của bước thực hiện.
2. Trạng thái xảy ra trong bước.
3. Điều kiện để chuyển tiếp đến bước kế.
4. Bước mới được chuyển đến.

Ví dụ:



# CHƯƠNG IV

## HƯỚNG DẪN SỬ DỤNG

### PHẦN MỀM FXTRN-BEG-E

#### I. Giới thiệu:



Đây là phần mềm mô phỏng các cơ cấu có trong thực tế của hãng **MITSUBISHI ELECTRIC** Phần mềm **FXTRN-BEG-E** được ứng dụng hình ảnh các cơ cấu chấp hành một cách sinh động, nó giúp cho người sử dụng có thể có một cái nhìn khái quát hơn trong đời sống sản xuất thực tế.

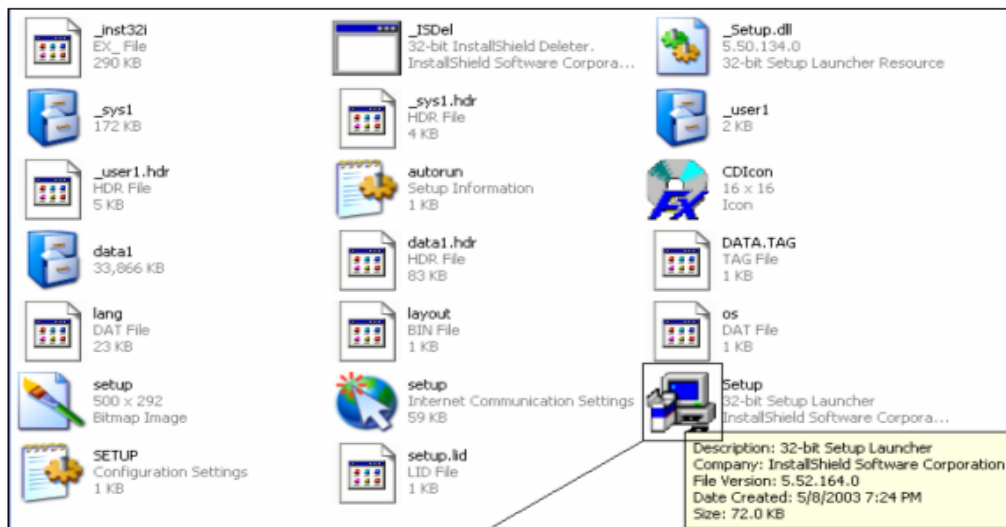
Để dùng phần mềm **FXTRN-BEG-E** yêu cầu về cấu hình phần cứng:

- Hệ điều hành **Microsoft Windows 98, Microsoft Windows 2000,...**
- **CPU Pentium** có tốc độ xử lý là **500MHZ** trở lên.
- Bộ nhớ 64 MB.
- Dung lượng còn trống của đĩa cứng là **150 MB** trở lên.
- **CD\_ROM** đọc cần có cho việc cài đặt.
- Display **XGA 1,024 x 768** hoặc cao hơn.
- Yêu cầu về Card màn hình **VRAM** từ **4MB** trở lên.

**\*Thực hiện cài đặt:**

a. Đặt đĩa chương trình **FX\_TRN\_BEG\_E** vào ổ **CD\_ROM**.

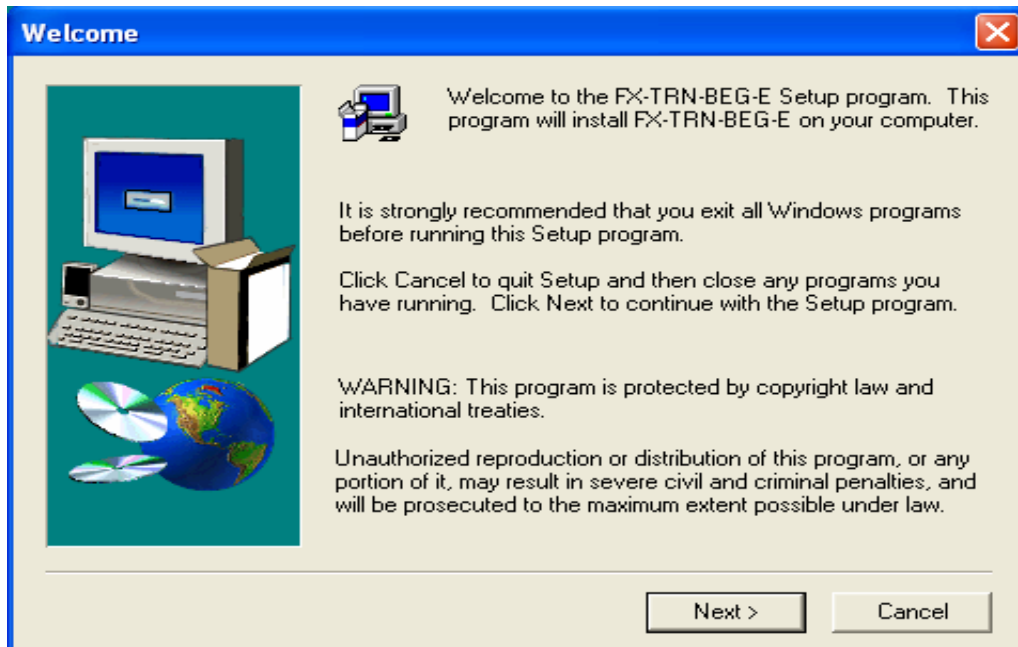
b. Tại **START** → **My Computer** → chọn ổ đĩa **CD\_ROM**.



Nhấp double trái chuột vào biểu tượng Setup

**Chú ý:** Có ba loại biểu tượng Setup và hãy chọn biểu tượng có hình dạng như trên.

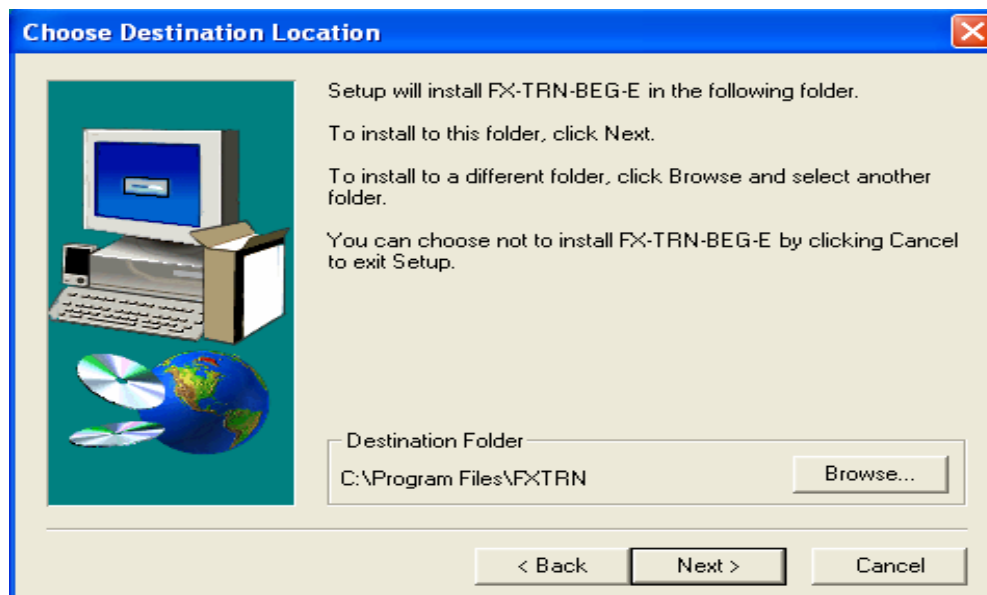
c. Sau khi nhấp double vào biểu tượng Setup: sẽ xuất hiện cửa sổ **Welcome**



- Kích Next để tiếp tục việc cài đặt, sau khi đọc và hiểu các phần hướng dẫn ở trên.

- Kích Cancel để hủy bỏ việc cài đặt.

*d. Khi kích Next ở phần trên màn hình sẽ xuất hiện cửa sổ Choose Destination Location:*



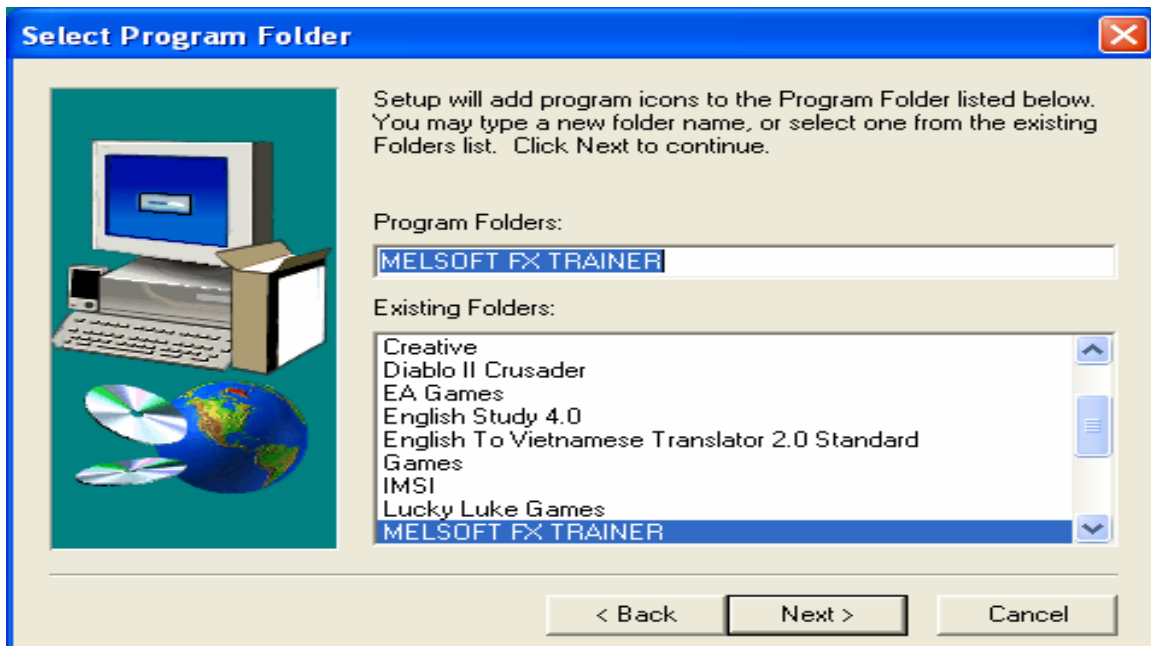
Tại đây cửa sổ này cho phép lựa chọn đường dẫn cho chương trình bằng cách click vào  Sau đó gõ vào đường dẫn mới.

- Click vào Next để tiếp tục việc cài đặt chương trình.

- Click vào Back để quay lại cửa sổ trước đó.
- Click vào Canel để hủy bỏ việc cài đặt.

*e. Khi click vào Next ở cửa sổ Chose Detination Location ,sau đó xuất hiện cửa sổ Select Program Folder*

Program Folders : Thư mục của chương trình có tên là **MELSOFT FX TRAINER.**

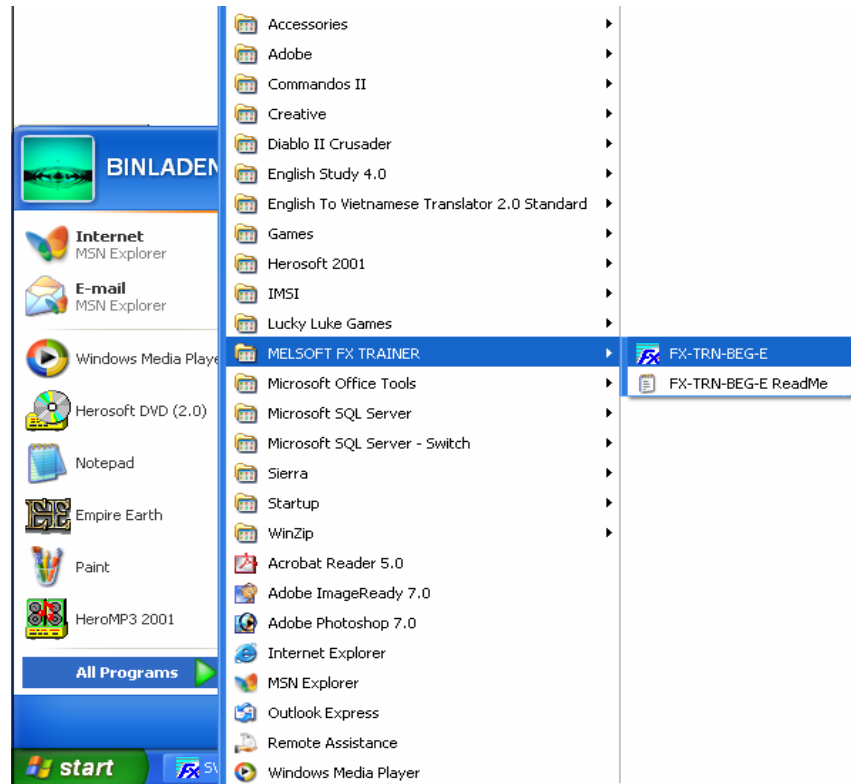


**Existing Folders :** Thư mục hiện có trong chương trình máy tính là **MELSOFT FX TRAINER.**

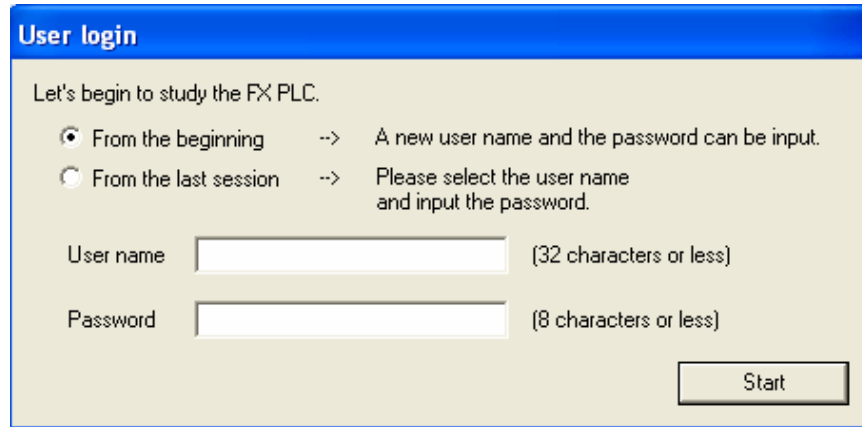
- Click vào Next để tiếp tục việc cài đặt chương trình.
- Click vào Back để quay lại cửa sổ trước đó.
- Click vào Canel để hủy bỏ việc cài đặt.

*f. Khi click Next tại cửa sổ Select Program Folder thì việc cài đặt bắt đầu.*

\*Khởi động phần mềm lập trình mô phỏng **FXTRN-BEG-E**  
Chọn Start→Program(hoặc All Program)→ **MELSOFT FX TRAINER**→**FXTRN-BEG-E** (double click).



## II. Giao diện:



Khi khởi tạo chương trình sẽ xuất hiện cửa sổ User login

- User name: gõ vào tên đăng ký sử dụng cho chương trình không quá 32 ký tự.
- Password : gõ vào khẩu lệnh cho chương trình đã đăng ký không quá 8 ký tự.

Sau đó click vào Start để xuất hiện cửa sổ giao diện.

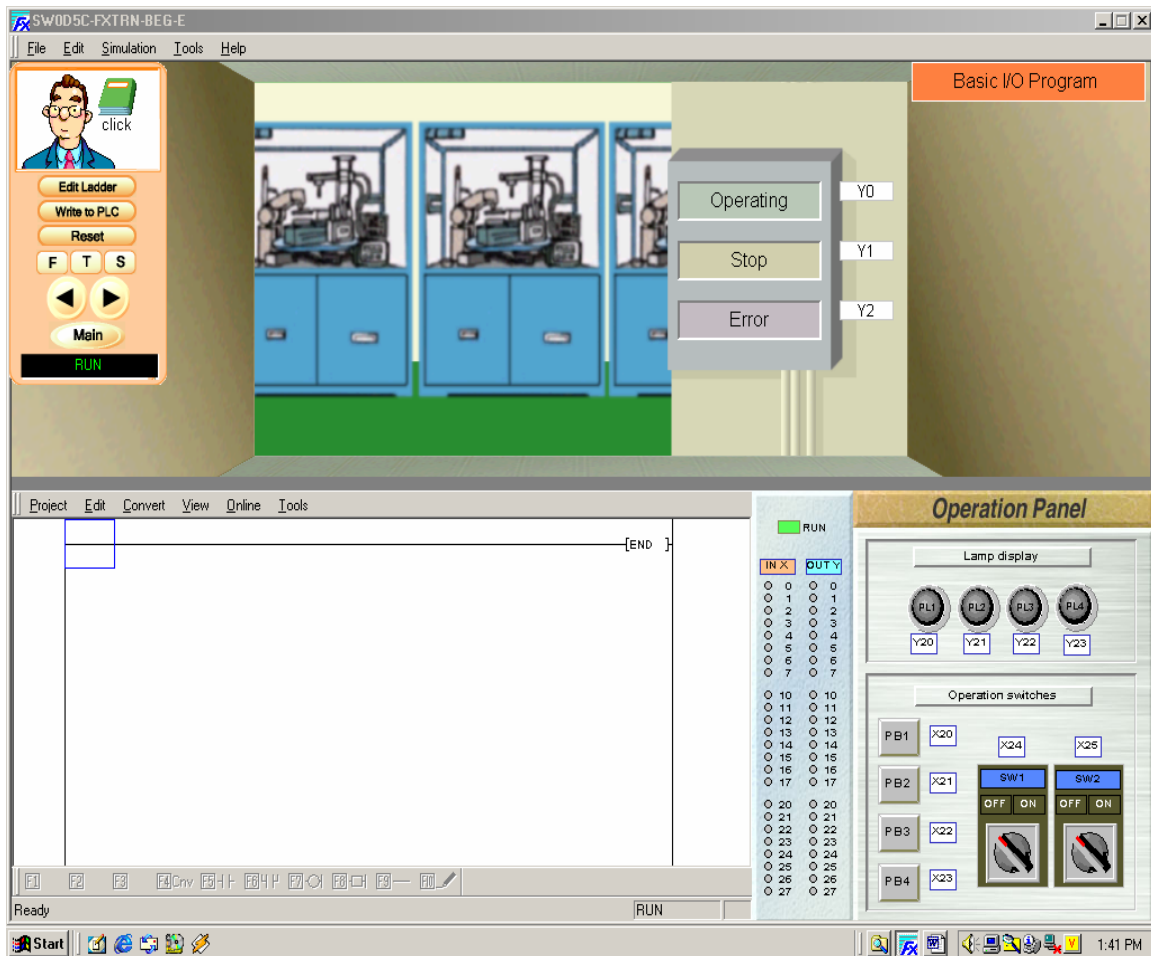




### **III. Giới thiệu cơ bản về giao diện một chương trình FXTRN-BEG-E:**

#### **1. Giao diện:**

Giao diện của chương trình mô phỏng FXTRN-BEG-E là giao diện có các chức năng linh hoạt trong cơ cấu chuyển cũng như việc vận hành điều khiển .



### Chú thích:

**Edit Ladder** : Chuyển sang chế độ lập trình ,lúc này bộ điều khiển ở chế độ **PROGRAM**.

**Reset** : Khởi tạo lại trạng thái ban đầu của chương trình.

**Write to PLC** : Nạp chương trình vào bộ PLC, lúc này bộ điều khiển ở chế độ **RUN**.

**F** (Front view) : Chức năng chiếu mặt của cơ cấu hình thể.

**S** ( Side view ) : Tổng quan về hình thể ở trạng thái 3D.

**T** (Top view ) : Chức năng chiếu bằng của cơ cấu hình thể.

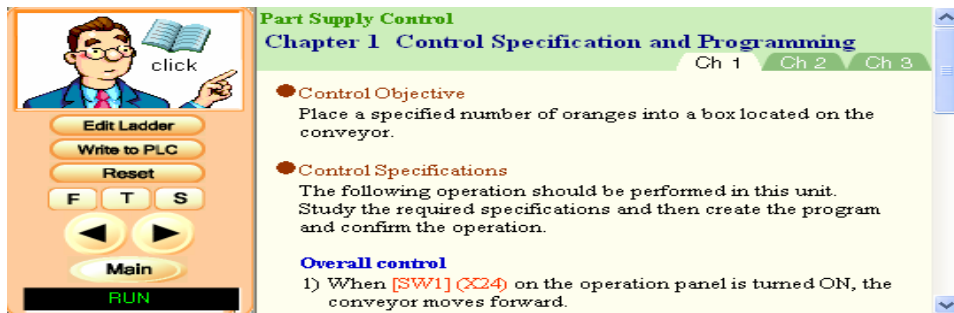
**Main** : Trở về cửa sổ chính của **FX\_TRN\_BEG\_E**.



: Khi Ckick vào hai nút này nó sẽ di chuyển tới hoặc lùi trong bảng hướng dẫn.

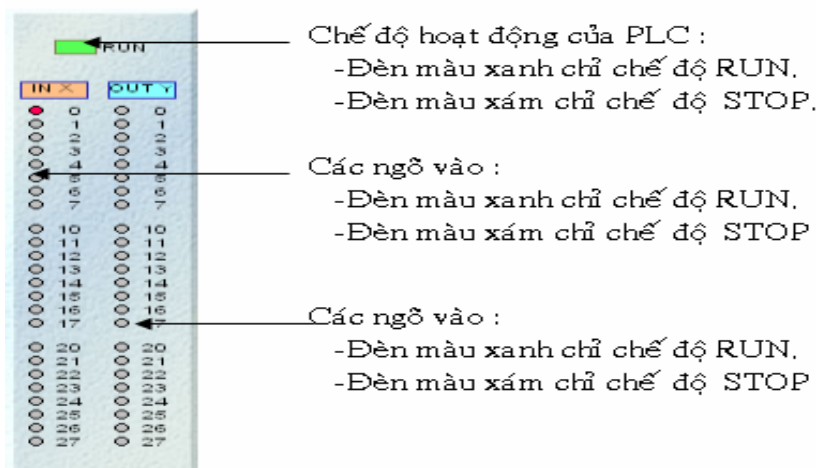


Khi click vào đây ,trên màn hình sẽ hiện trang hướng dẫn



**2. Bộ lập trình PLC mô phỏng :**

Các ký hiệu của PLC về các ngõ I/O được thể hiện qua cơ cấu sau :

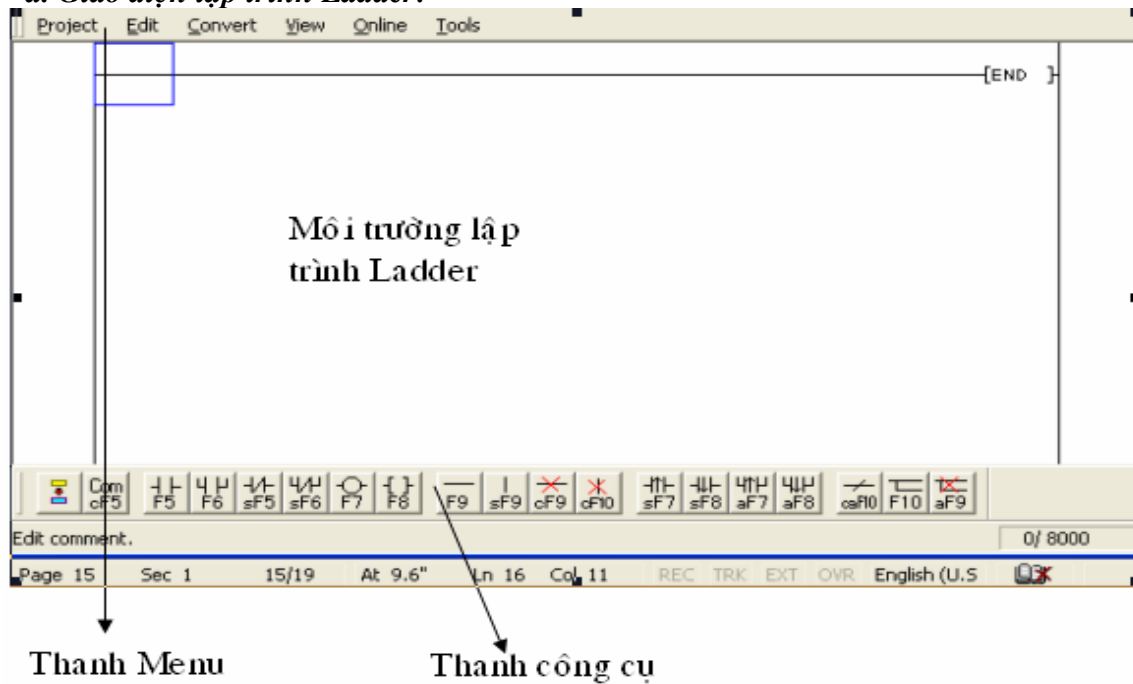


- Chế độ hoạt động của PLC :
  - Đèn màu xanh chỉ chế độ RUN.
  - Đèn màu xám chỉ chế độ STOP.
- Các ngõ vào :
  - Đèn màu xanh chỉ chế độ RUN.
  - Đèn màu xám chỉ chế độ STOP
- Các ngõ ra :
  - Đèn màu xanh chỉ chế độ RUN.
  - Đèn màu xám chỉ chế độ STOP

**IV. Phương pháp lập trình FXTRN-BEG-E:**

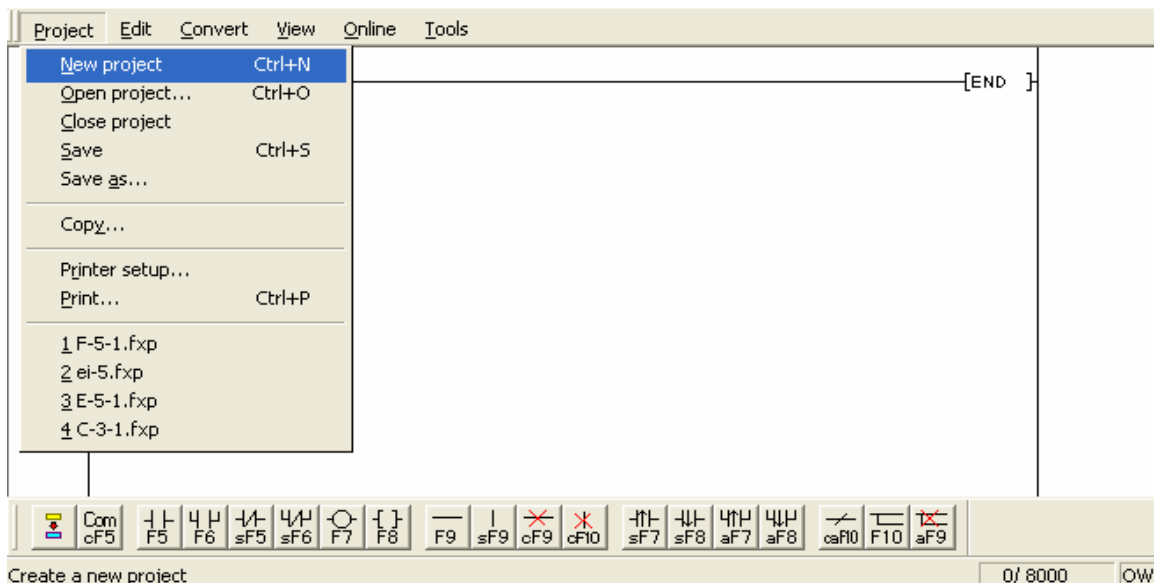
**1. Sử dụng phần mềm mô phỏng PLC:**

**a. Giao diện lập trình Ladder:**



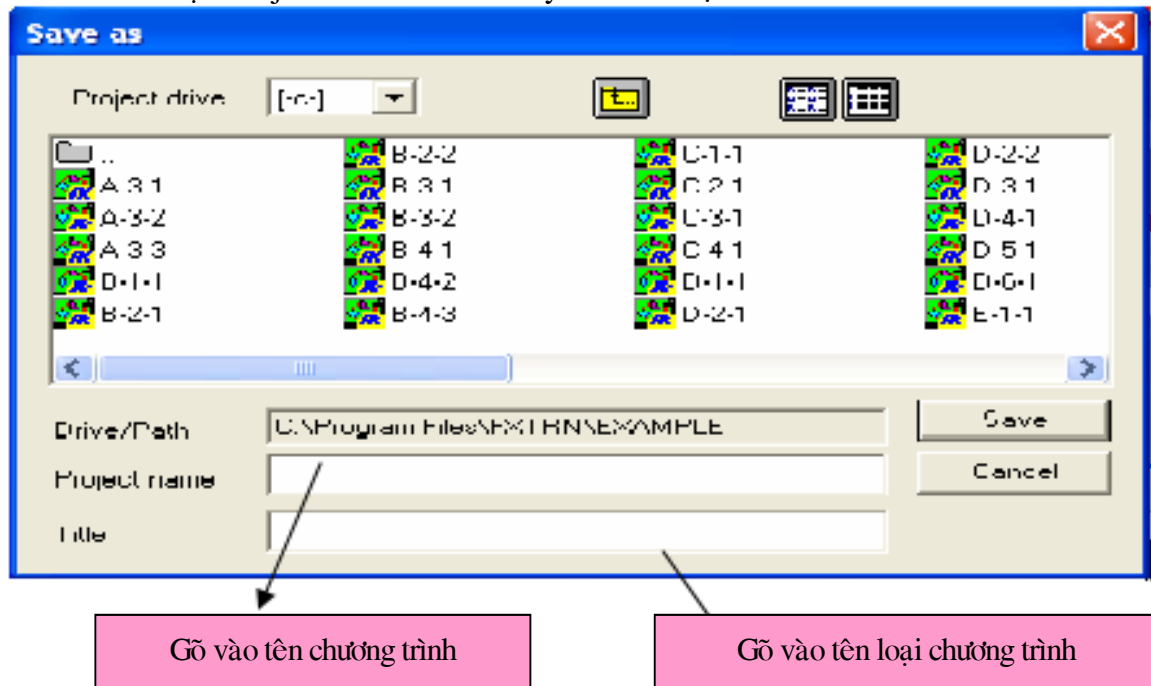
**b. Phương pháp sử dụng:**

1 Thiết lập một chương trình mới : Chọn **Project** → New project .Lúc một chương trình đã được thiết lập trên **FXTRN-BEG-E**. Màn hình lập trình cho phép người sử dụng viết chương trình.

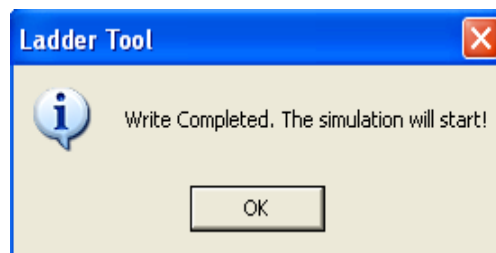
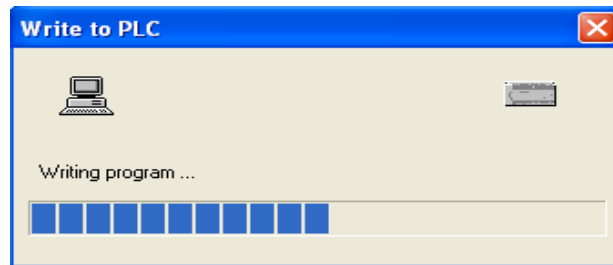


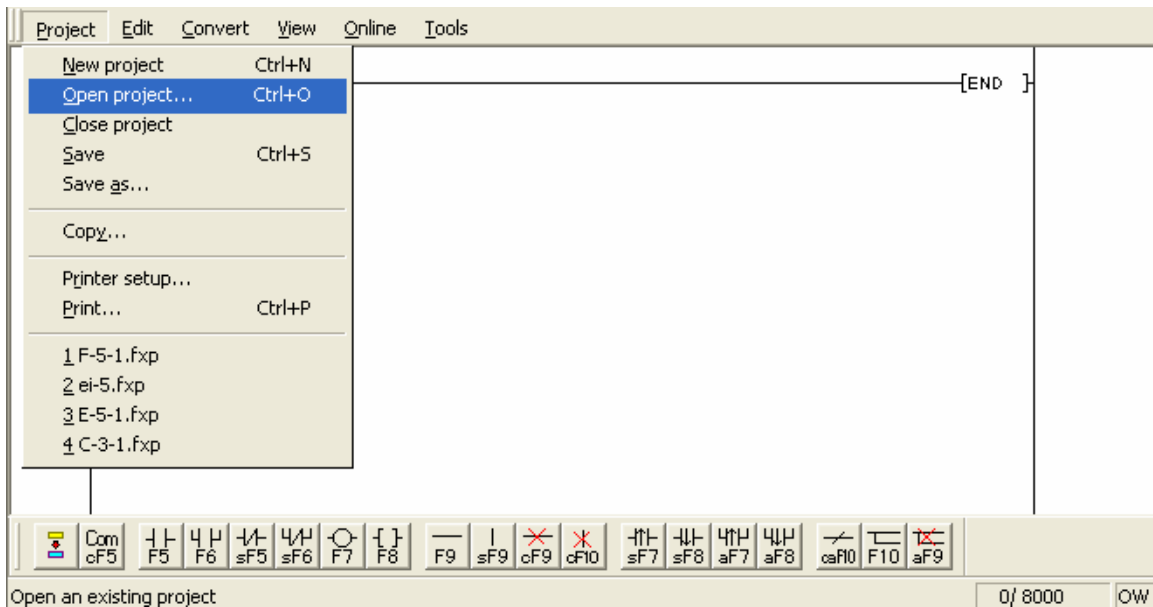
2 Lưu chương trình :

- Chọn Project → Save .Lúc này sẽ xuất hiện cửa sổ Save as



- Lưu chương trình với một tên khác : Chọn Project → Save as
- 1 Mỗi một chương trình : Chọn Project → Open . Sau đó xuất hiện cửa sổ Open Project.






Gõ vào tên tiếp điểm .Chọn OK

**c. Nạp chương trình PLC:**


Có thể click vào **Write to PLC** trên bảng điều khiển từ xa ( Người trợ lý hướng dẫn) .  
Hoặc có thể chọn **Online** → **Write to PLC**.

**2. Phương pháp lập trình:**

**a.Tiếp điểm thường mở:**

Kích vào biểu tượng  hoặc nhấn phím **F5** .Sau đó xuất hiện cửa sổ **Enter symbol**


**b.Tiếp điểm thường đóng:**

Kích vào biểu tượng  hoặc nhấn phím **Shift+F5** . Sau đó xuất hiện cửa sổ **Enter symbol**



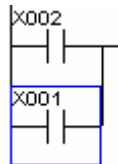
Gõ tên tiếp điểm .Chọn OK

**c.Tiếp điểm thường mở song song:**

Kích vào biểu tượng  hoặc nhấn phím **F6** .Sau đó xuất hiện cửa sổ




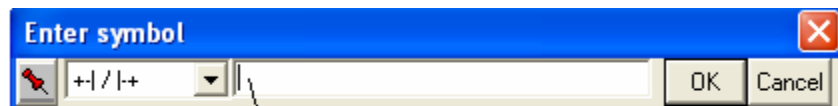
Gõ vào tên tiếp điểm.Chọn OK



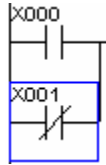
Dạng tiếp điểm song song :

**d.Tiếp điểm thường đóng dạng song song:**


Kích vào biểu tượng  hoặc nhấn phím Shift+F6 .Sau đó xuất hiện cửa sổ



Gõ vào tên tiếp điểm . Chọn OK



Dạng tiếp điểm thường đóng song song:

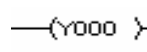
e. **Cuộn dây:** ( bao gồm **Y,T,C,M,S**) kích vào biểu tượng  hoặc nhấn phím **F7** .Sau đó xuất hiện cửa sổ



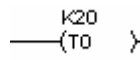
Gõ vào tên thiết bị .Chọn OK

**Các dạng cuộn dây:**

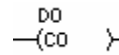
1 Cuộn dây **Y**:



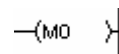
2 Cuộn dây **Timer (T)**:



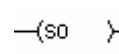
3 Cuộn dây **Counter (C)**:



4 Relay phụ trợ **M**:



5 Relay trạng thái **S**: (cờ hiệu)



**f. Cung cấp hàm truyền đạt:**



Kích vào biểu tượng  hoặc nhấn phím F8 .Sau đó xuất hiện cửa sổ **Enter symbol**



Gõ vào tên hàm cần truyền đạt

Các hàm truyền đã được giới thiệu ở chương III.

### 3. Xử lý trạng thái mạch:

#### **1 Mạch thẳng:**

Kích vào  hoặc nhấn phím **F9** và nhấn **ENTER** (↵).

#### **2 .Mạch rẽ nhánh :**

Kích vào  hoặc nhấn phím **Shift+F9** và nhấn **ENTER** (↵).

#### **3 Xoá mạch thẳng:**

Nhấn **Ctrl + F9**, đặt con trỏ tại vị trí muốn xoá.

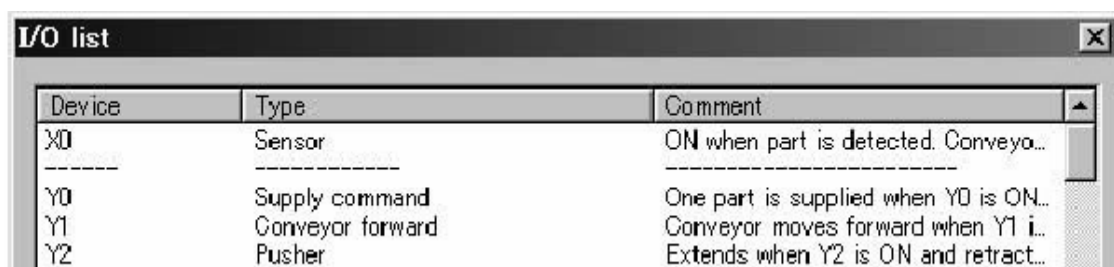
#### **4 Xoá mạch vuông góc:**

Nhấn **ctrl + F10**, đặt con trỏ bên phải vị trí muốn xoá.

### 4. Xử lý trong FXTRN-BEG-E:

#### **a. Khai báo các I/O:**

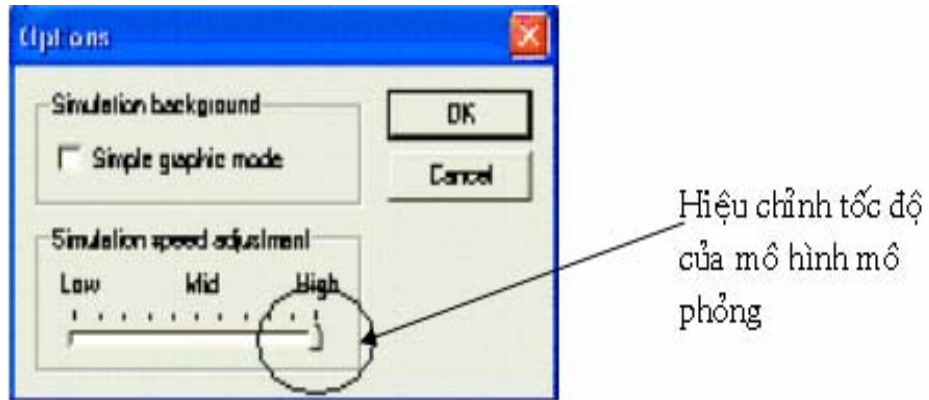
Chọn **Edit**→**I/O list**. Tại đây có thể xem các khai báo trong chương trình mô phỏng.



Device	Type	Comment
X0	Sensor	ON when part is detected. Conveyo...
Y0	Supply command	One part is supplied when Y0 is ON...
Y1	Conveyor forward	Conveyor moves forward when Y1 i...
Y2	Pusher	Extends when Y2 is ON and retract...

#### **b. Điều chỉnh tốc độ trong mô hình:**

Chọn **Tools**→**Option**.



\*Simulation speed adjustment : Điều chỉnh tốc độ hoạt động cho các cơ cấu trong mô hình mô phỏng. Chọn OK.

***c. Thoát chương trình:***

Chọn File → Exit.

CHƯƠNG V

# CÁC BÀI TẬP ỨNG DỤNG TRONG PLC FXTRN-BEG-E

## I. CÁC DẠNG BÀI TẬP

### A. Giới thiệu về họ lập trình FX (Learn the FX series PLC) :

Bài tập	Chủ đề	Mức độ khó	Nội dung điều khiển
A-1 Giới thiệu về các loại PLC FX	Các ứng dụng của phần mềm mô phỏng qua các ngõ I/O	1	Điều khiển cửa
A-2 Ứng dụng về chủng loại PLC FX	Giới thiệu các họ PLC FX	1	Điều khiển cửa
A-3 Cách thức thiết lập một chương trình	Ứng dụng cho người mới bắt đầu lập trình PLC	1	Điều khiển băng tải

### B. Sử dụng các ngõ điều khiển (Let`s study the Basics):

Bài tập	Chủ đề	Mức độ khó	Nội dung điều khiển
B-1 Các ngõ I/O cơ bản trong chương trình	Tìm hiểu về các ngõ I/O trong chương trình	1	Điều khiển đèn
B-2 Một chương trình cơ bản	Tìm hiểu về ngõ ra và hàm lệnh SET- RST	1	Điều khiển đèn
B-3 Chương trình điều khiển thiết lập bằng thời gian	Tìm hiểu về dạng chương trình nhánh	1	Điều khiển đèn giao thông với hai đèn đơn
B-4 Tìm hiểu về các kí hiệu ngõ vào	Tìm hiểu về các hệ thống cảm biến và cơ cấu hoạt động của các dây truyền đơn giản	2	Điều khiển băng tải

**C. Dạng lập trình cơ bản (Easy dose it !):**

Bài tập	Chủ đề	Mức độ khó	Nội dung điều khiển
<b>C-1</b> Vận hành cơ cấu bằng timer cơ bản	Tìm hiểu về timer – on delay	1	Điều khiển cửa
<b>C-2</b> Ứng dụng timer trong chương trình 1	Tìm hiểu về timer – off delay và cách dùng thời gian theo dõi	1	Điều khiển cửa
<b>C-3</b> Ứng dụng timer trong chương trình 2	Sự chuyển đổi thời gian trong timer	2	Điều khiển đèn giao thông với hai đèn đơn
<b>C-4</b> Chương trình couter cơ bản	Tìm hiểu về hoạt động đếm của couter	2	Điều khiển băng tải

**D. Bắt đầu với phân lập trình cơ bản( Begginer Challenge ):**

Bài tập	Chủ đề	Mức độ khó	Nội dung điều khiển
<b>D-1</b> Thành phần phục vụ ứng dụng trong lập trình	Tìm hiểu về sự điều khiển tự động trong nhà hàng	1	Điều khiển khu vực phục vụ trong nhà hàng
<b>D-2</b> Chế độ hoạt động của cảm biến quang	Khả năng nhận biết của cảm biến quang	1	Điều khiển đèn tia
<b>D-3</b> Điều khiển mạch đèn giao thông	Điều khiển mạch đèn giao thông bằng timer theo chu kỳ	1	Điều khiển đèn giao thông với ba đèn đơn
<b>D-4</b> Bộ phận nhận biết kích cỡ (I)	Khả năng nhận biết của sensor kích cỡ	1	Điều khiển băng tải và robot cung cấp sản phẩm
<b>D-5</b> Tắt và mở băng tải	tắt và mở băng tải trong một cơ cấu sản xuất	2	Điều khiển băng tải
<b>D-6</b> Hoạt động của băng tải	Điều khiển băng tải thông qua các sensor dữ liệu	2	Điều khiển băng tải

***E. Thực hiện chương trình với mức trung bình ( Intermediate Challenge ):***

Bài tập	Chủ đề	Mức độ khó	Nội dung điều khiển
E-1 Nút nhấn đơn	Điều khiển đèn giao thông bằng nút nhấn	2	Điều khiển đèn giao thông với ba đèn đơn
E-2 Sự nhận biết của sensor kích cỡ(II)	Khả năng nhận biết kích thước của các sensor trong mỗi bộ phận	2	Điều khiển băng truyền rẽ nhánh
E-3 Bộ phận tách rời	Sự chuyển động của cánh tay robot thông qua hàm truyền	2	Điều khiển băng tải và robot
E-4 Máy khoan	Điều khiển máy khoan và các thiết bị khác trong cơ cấu	2	Điều khiển băng tải, máy khoan và robot cung cấp sản phẩm
E-5 Điều khiển bộ phận cung cấp	Điều khiển sự hoạt động của các bộ phận cung cấp	3	Điều khiển băng tải và robot
E-6 Điều khiển băng tải	Điều khiển chuyển động băng tải tới hoặc lùi	3	Điều khiển băng tải và robot

***F. Thực hiện chương trình với mức độ khó (Advanced Challenge ):***

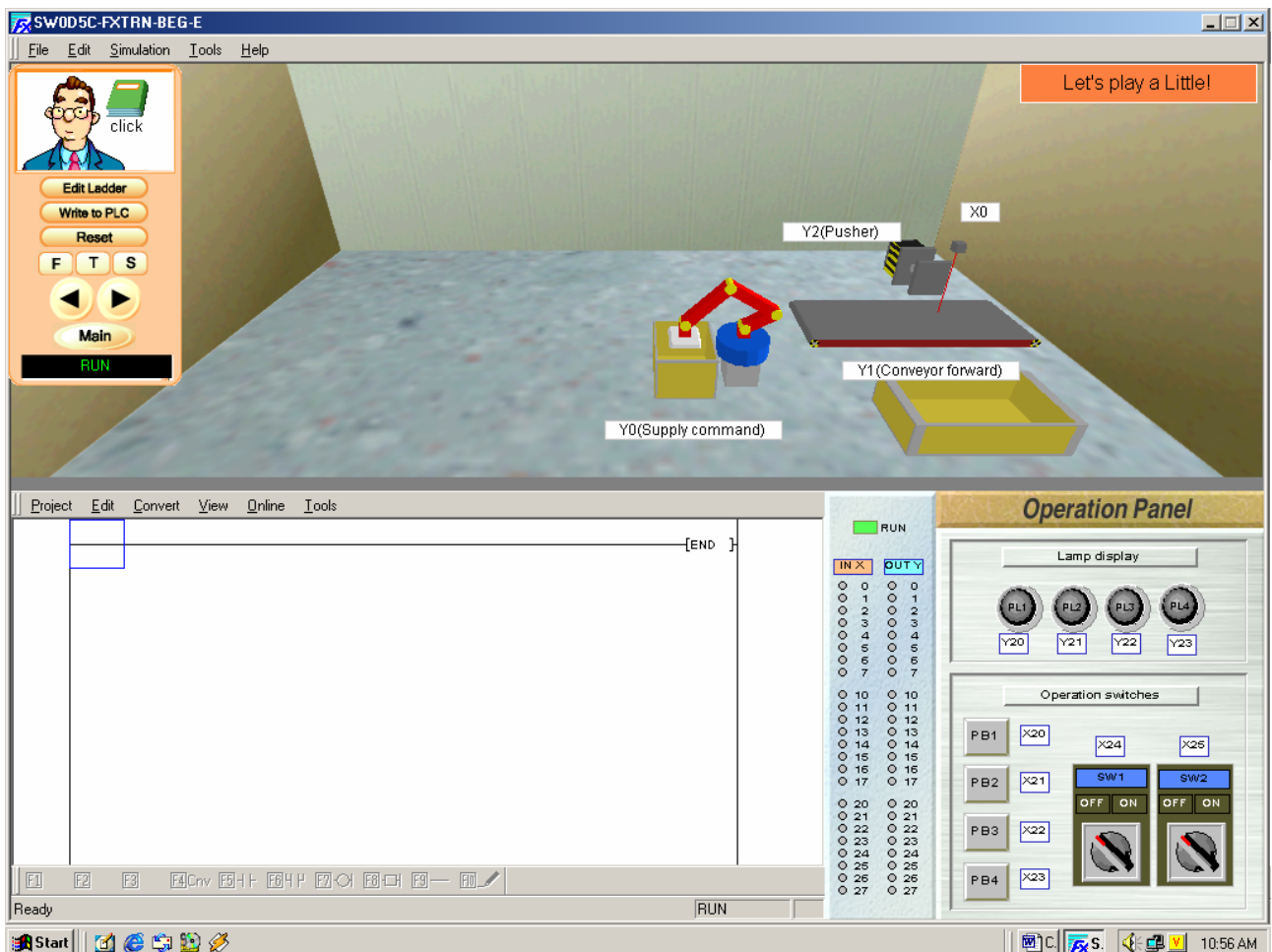
Bài tập	Chủ đề	Mức độ khó	Nội dung điều khiển
F-1 Điều khiển cửa tự động	Vận hành cửa tự động đóng/mở	3	Điều khiển cửa
F-2 Chuyển động vị trí trong cơ cấu	Điều khiển các sản phẩm đi qua các khu vực cho phép thông qua các sensor	3	Điều khiển các cơ cấu dừng
F-3 Bộ phận phân phối sản phẩm	Phân phối các dạng sản phẩm tùy theo kích thước	3	Điều khiển băng tải và pittông
F-4 Bộ phận nhận biết sự sai lệch trong dây truyền	Khả năng nhận biết sản phẩm tốt/xấu trong dây truyền thông qua cảm biến	3	Điều khiển băng tải, máy khoan và robot cung cấp sản phẩm
F-5 Điều khiển chuyển động quay thuận/ngược	Chuyển động của băng tải tới hoặc lùi	3	Điều khiển băng tải, máy khoan và robot cung cấp sản phẩm

F-6 Điều khiển thang máy	Sử dụng thang máy 3 cấp	3	Điều khiển băng tải và thang máy
F-7 Khả năng nhận biết và cung cấp sản phẩm của sensor	Nhận biết sản phẩm thông qua sensor	4	Điều khiển trạng thái hoạt động của cánh tay robot On/Off

## II. BÀI GIẢI THAM KHẢO CÁC BÀI TẬP MÔ HÌNH TRONG FXTRN-BEG-E

### 1. BÀI TẬP A3 : ĐIỀU KHIỂN BĂNG TẢI

#### a/. Giao diện mô hình



**b/. Bảng Thiết bị cần dùng (device Assignment)**

Ngõ thiết bị	Kí hiệu ngõ I/O	Tên thiết bị	Chức năng
Input	X024	Công tắc	Khởi động và dừng hệ thống
Input	X000	Cảm biến	Nhận hàng tại pitông
Output	Y000	Động cơ điều khiển cánh tay robot	Điều khiển bộ phận cung cấp hàng
Output	Y001	Động cơ kéo băng tải	ON khi hệ thống làm việc
Output	Y020	Đèn đỏ	Sáng khi hệ thống ngưng hoạt động
Output	Y023	Đèn xanh	Sáng khi hệ thống đang làm việc

**c/. Nguyên lý hoạt động:**

Khi hệ thống đang ngưng hoạt động thì đèn đỏ Y020 sáng. Khi bật công tắc X024 thì cuộn dây Y00 và Y001 có điện và tự duy trì vì lúc trước đó cảm biến X000 là thường đóng. Cánh tay robot bắt đầu cấp hàng lên băng tải và băng tải quay thuận đưa hàng về cuối hành trình. Lúc này cuộn dây công tắc tơ Y002 cũng chưa có điện bởi vì được mắc nối tiếp với tiếp điểm thường mở của cảm biến X000. Khi hàng tới trước cảm biến thì lập tức cảm biến này đổi trạng thái mở tiếp điểm thường đóng ngắt điện của hai cuộn dây Y000 và Y001 để động cơ kéo băng tải ngưng lại và cánh tay robot trở về trạng thái ban đầu. Đồng thời đóng tiếp điểm thường mở cấp điện cho cuộn dây Y002 điều khiển pitông đẩy hàng về nơi quy định. Khi hàng vừa được đẩy xuống thì cảm biến đổi trạng thái ngắt điện cuộn dây Y002 đồng thời đóng điện cho Y000 và Y001 để cánh tay robot tiếp tục cấp hàng và băng tải tiếp tục hoạt động theo hành trình trước. Trong lúc đang hoạt động đèn xanh Y023 sáng.

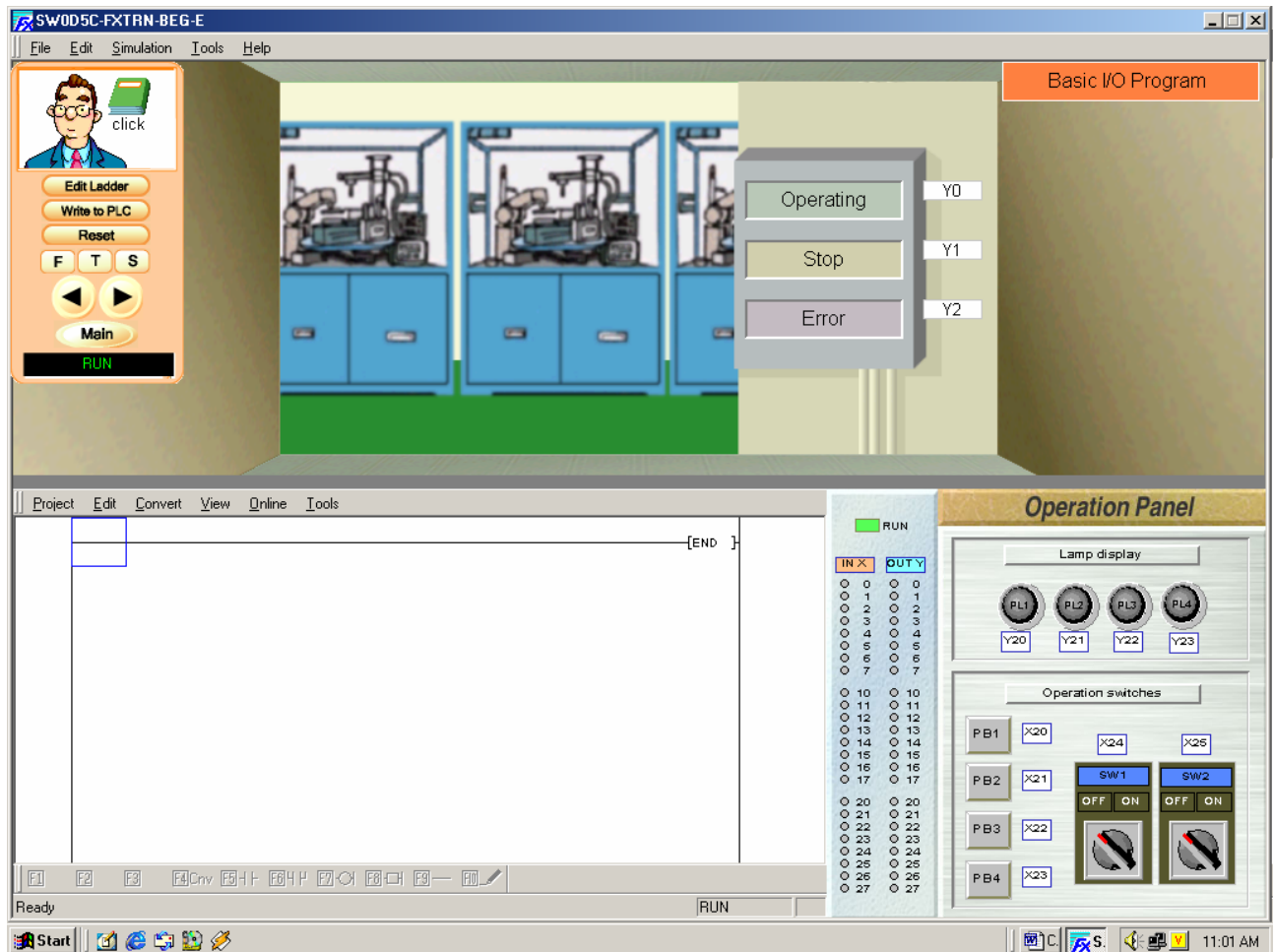
Khi muốn dừng ta chỉ việc bật công tắc X024 lập tức cánh tay robot trở về trạng thái ban đầu. Nếu còn hàng trên băng tải thì dây chuyền vẫn hoạt động để khi hàng về hết hành trình thì hệ thống mới ngưng hoạt động và đèn đỏ Y020 sáng báo dây chuyền đã ngưng hoạt động.

d/. Sơ đồ mạch điện



2. BÀI TẬP B1, B2 : ĐIỀU KHIỂN ĐÈN

a/. Giao diện mô hình



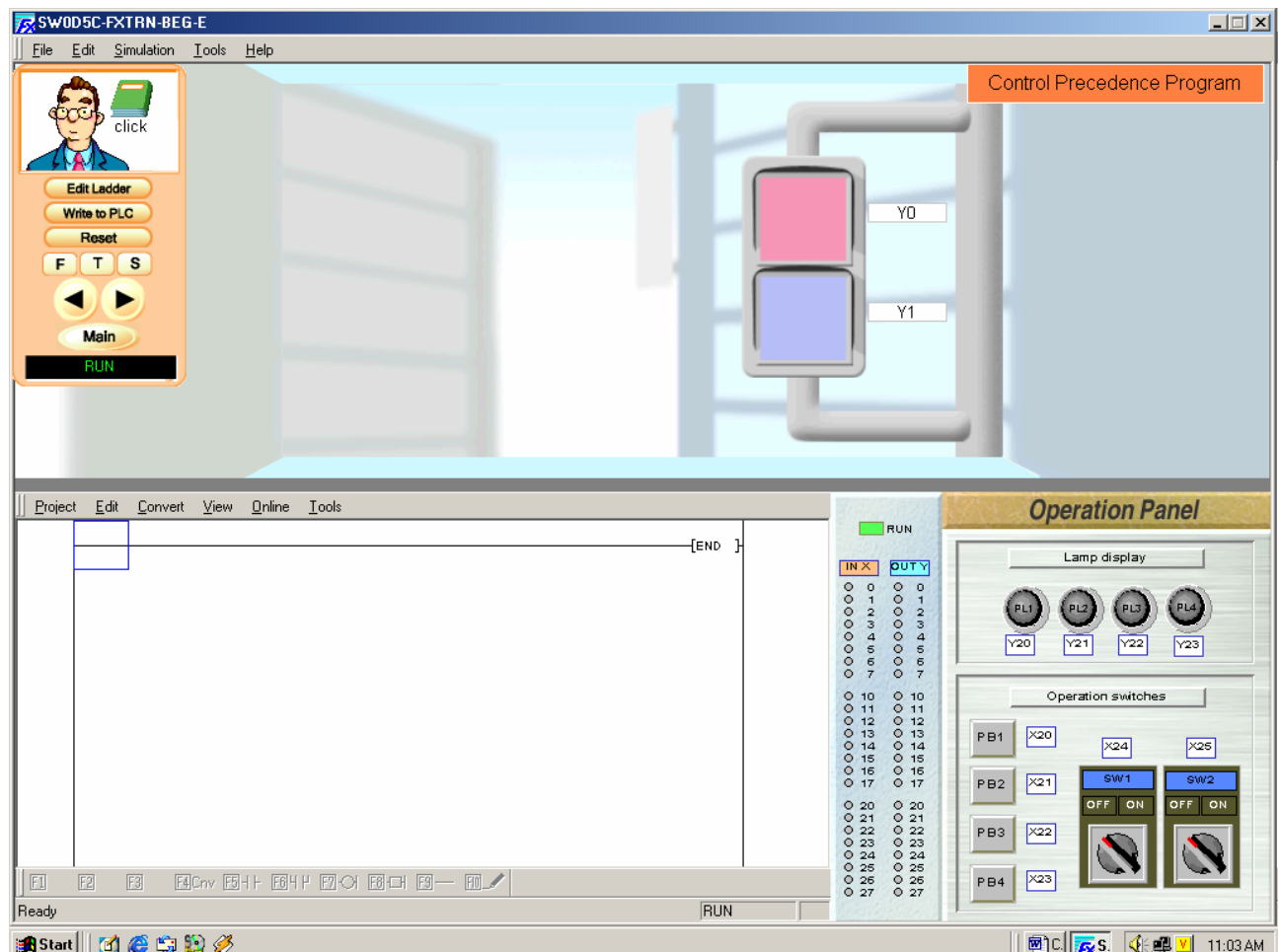


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X20	Nút điều khiển đèn xanh	
	X21	Nút điều khiển đèn đỏ	
	X22	Nút điều khiển đèn đỏ	
	X24	Nút điều khiển đèn đỏ	ON khi hàng chuyển đến quét ngang băng tải dừng.
NGÕ RA	Y0	Đèn vận hành	Đèn xanh sáng khi Y0 lên mức on
	Y1	Đèn báo dừng	Đèn vàng sáng khi Y1 lên mức on
	Y2	Đèn báo lỗi	Đèn đỏ sáng khi Y2 lên mức on

**BÀI TẬP B3 : ĐIỀU KHIỂN ĐÈN GIAO THÔNG VỚI HAI ĐÈN ĐƠN**

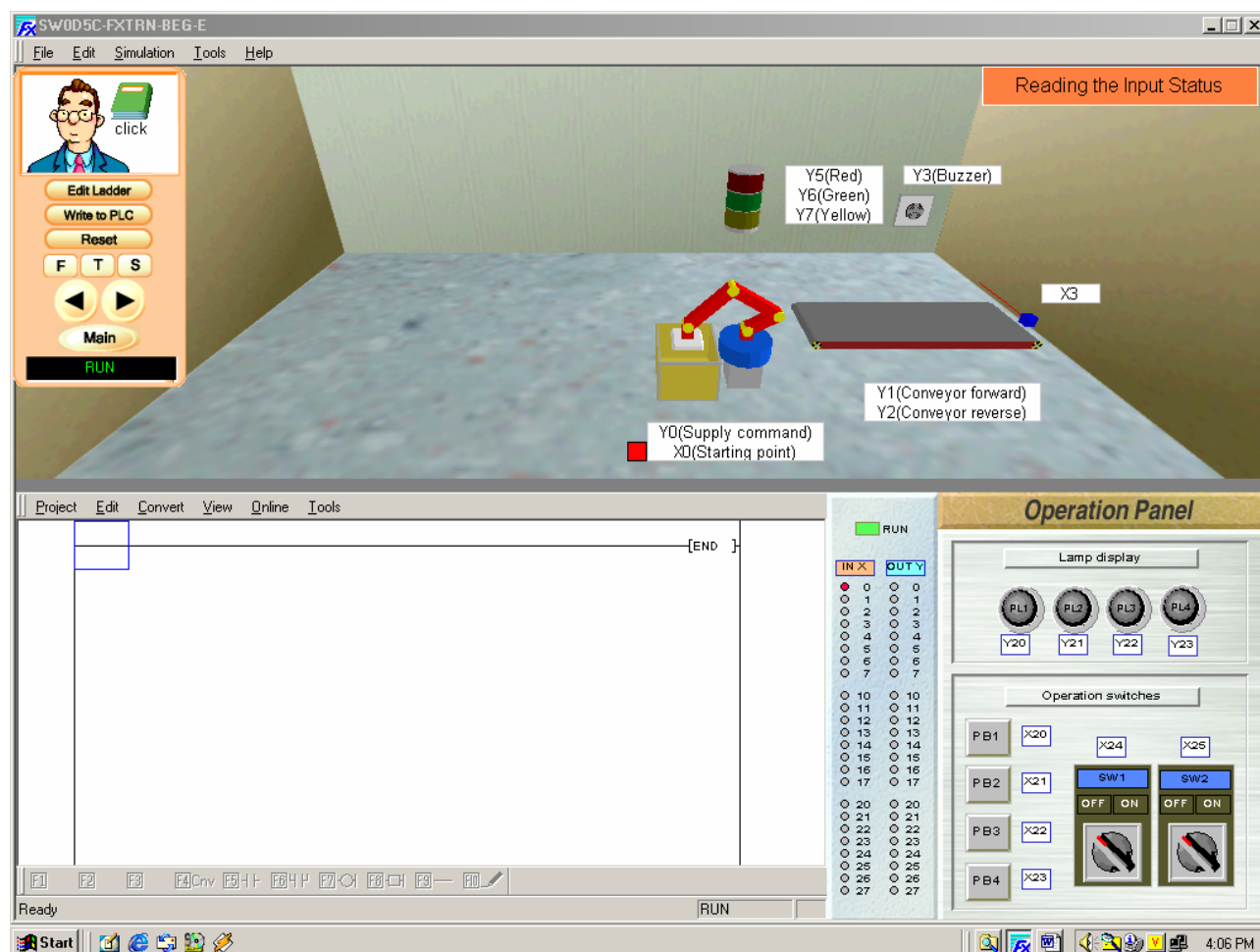


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X20	Nút nhấn ĐK đèn đỏ	
	X24	Công tắc ĐK đèn đỏ	
	X21	Nút nhấn ĐK đèn xanh	
	X25	Công tắc ĐK đèn xanh	
NGÕ RA	Y0	Đèn tín hiệu đỏ	Đèn đỏ sáng khi Y0 lên mức ON
	Y1	Đèn tín hiệu xanh	Đèn xanh sáng khi Y1 lên mức ON

### BÀI TẬP B4 : ĐIỀU KHIỂN BĂNG TẢI

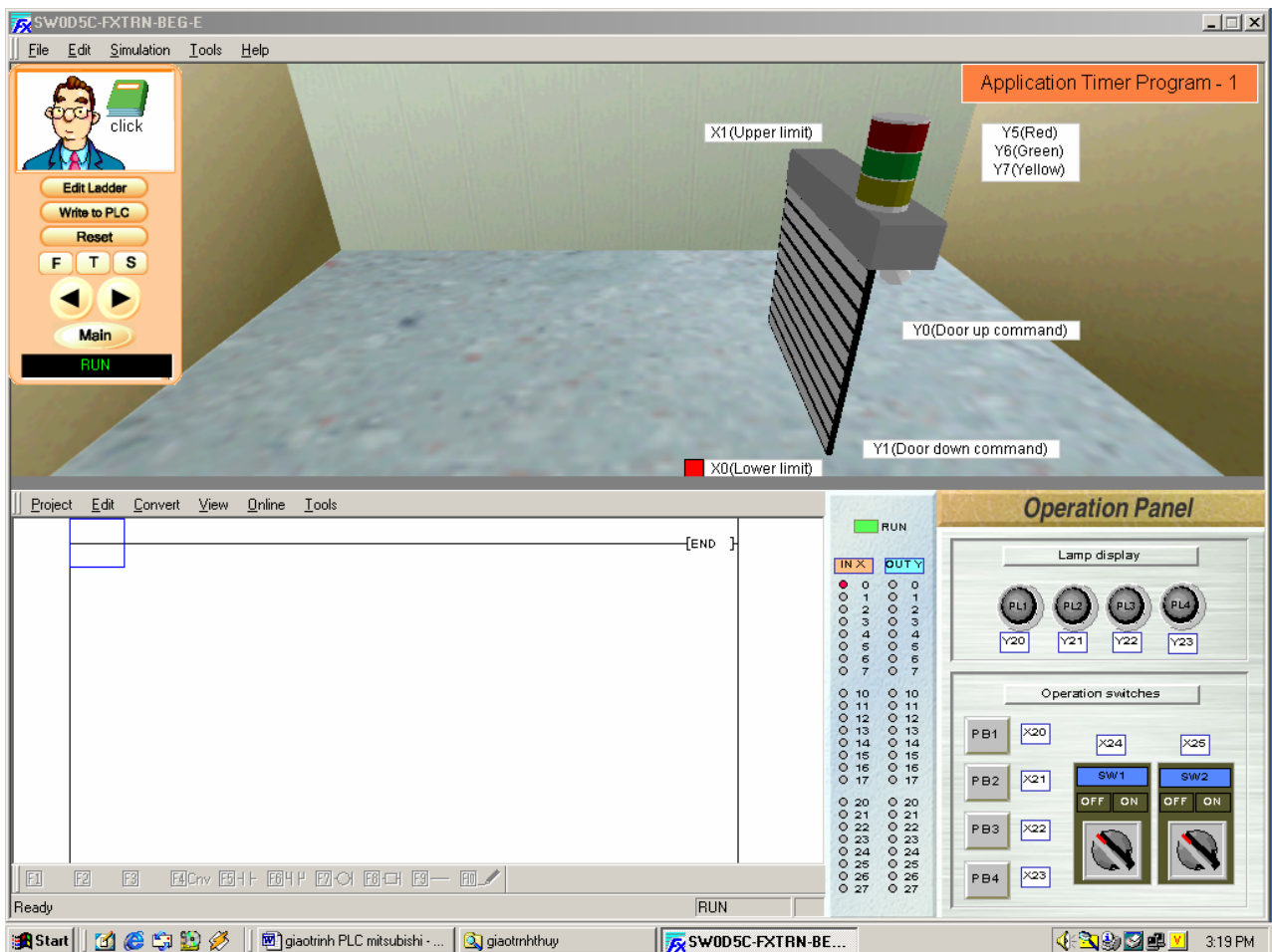


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X00		
	X03		
	X20		
	X25		
NGÕ RA	Y0		
	Y1		
	Y2		
	Y3		
	Y5		

**BÀI TẬP C1, C2 : ĐIỀU KHIỂN CỬA**

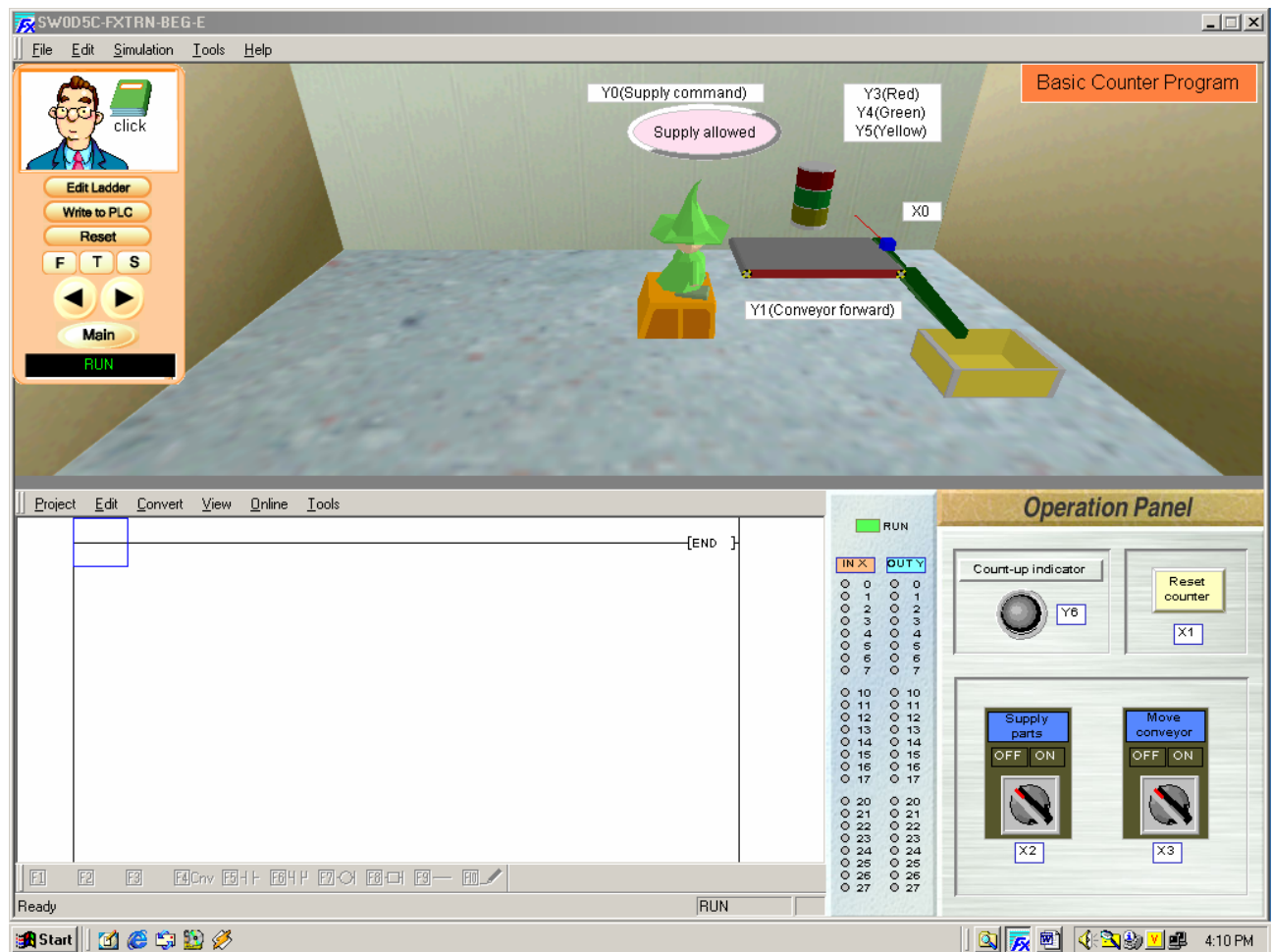


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X00		
	X01		
	X20		
	X24		
NGÕ RA	Y0		
	Y1		
	Y5		
	Y6		
	Y7		

**BÀI TẬP C4 : ĐIỀU KHIỂN BĂNG TẢI**

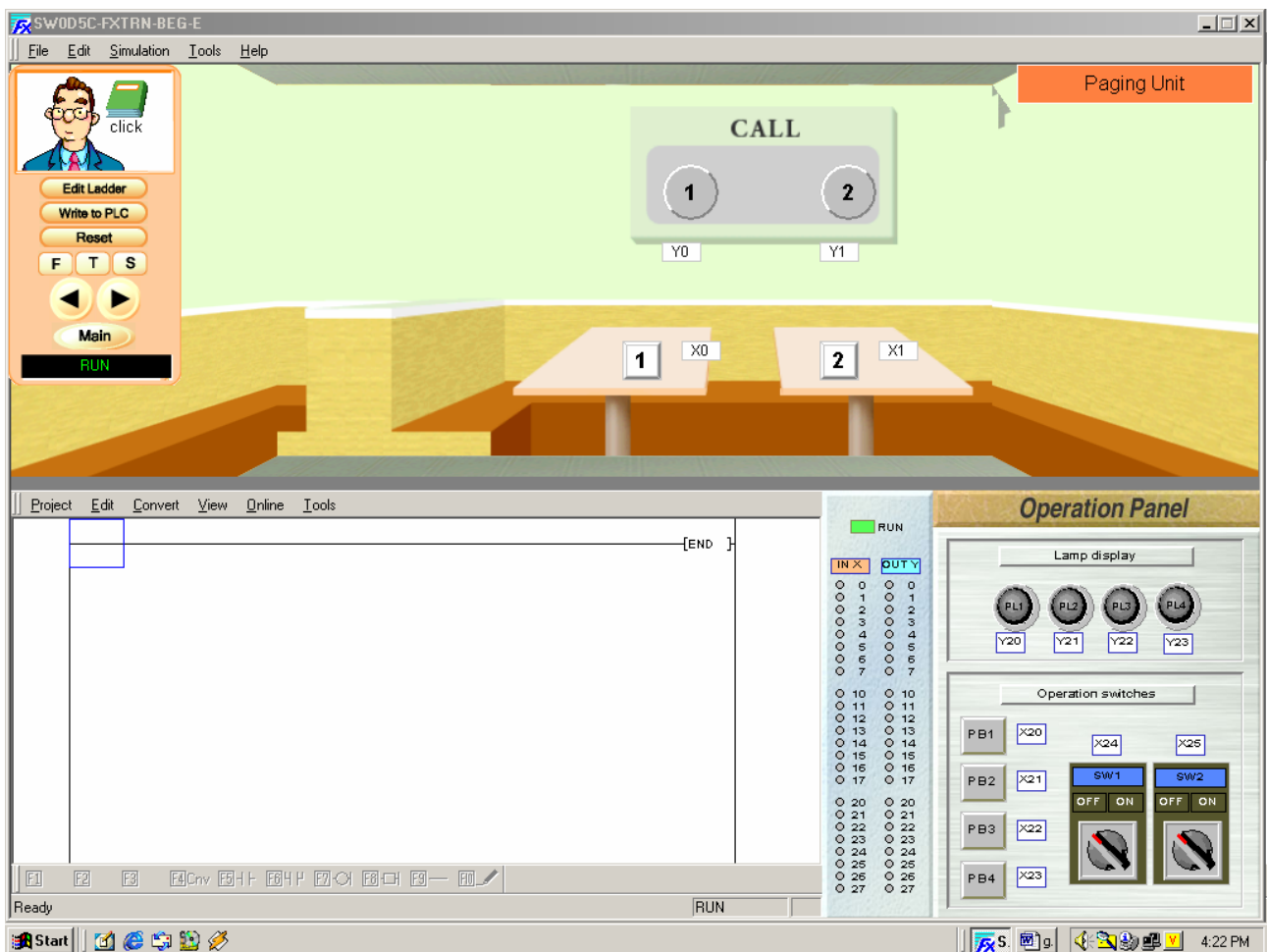


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X00		
	X20		
	X21		
	X24		
NGÕ RA	Y0		
	Y1		
	Y3		
	Y4		
	Y5		

**BÀI TẬP D1: ĐIỀU KHIỂN KHU VỰC PHỤC VỤ TRONG NHÀ HÀNG**

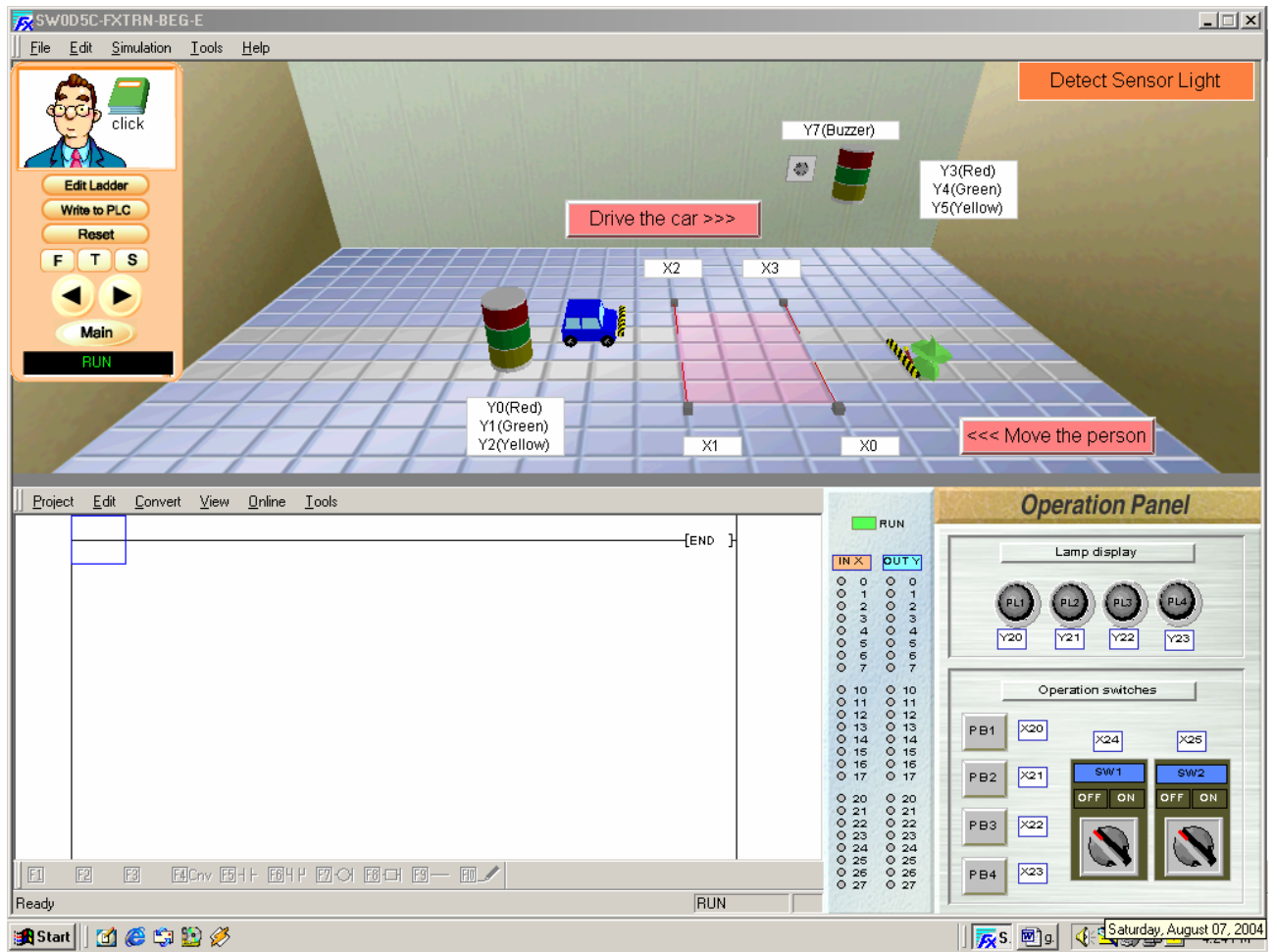


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X00		
	X01		
	X20		
	X21		
NGÕ RA	Y0		
	Y1		

### BÀI TẬP D2 : ĐIỀU KHIỂN ĐÈN TIA

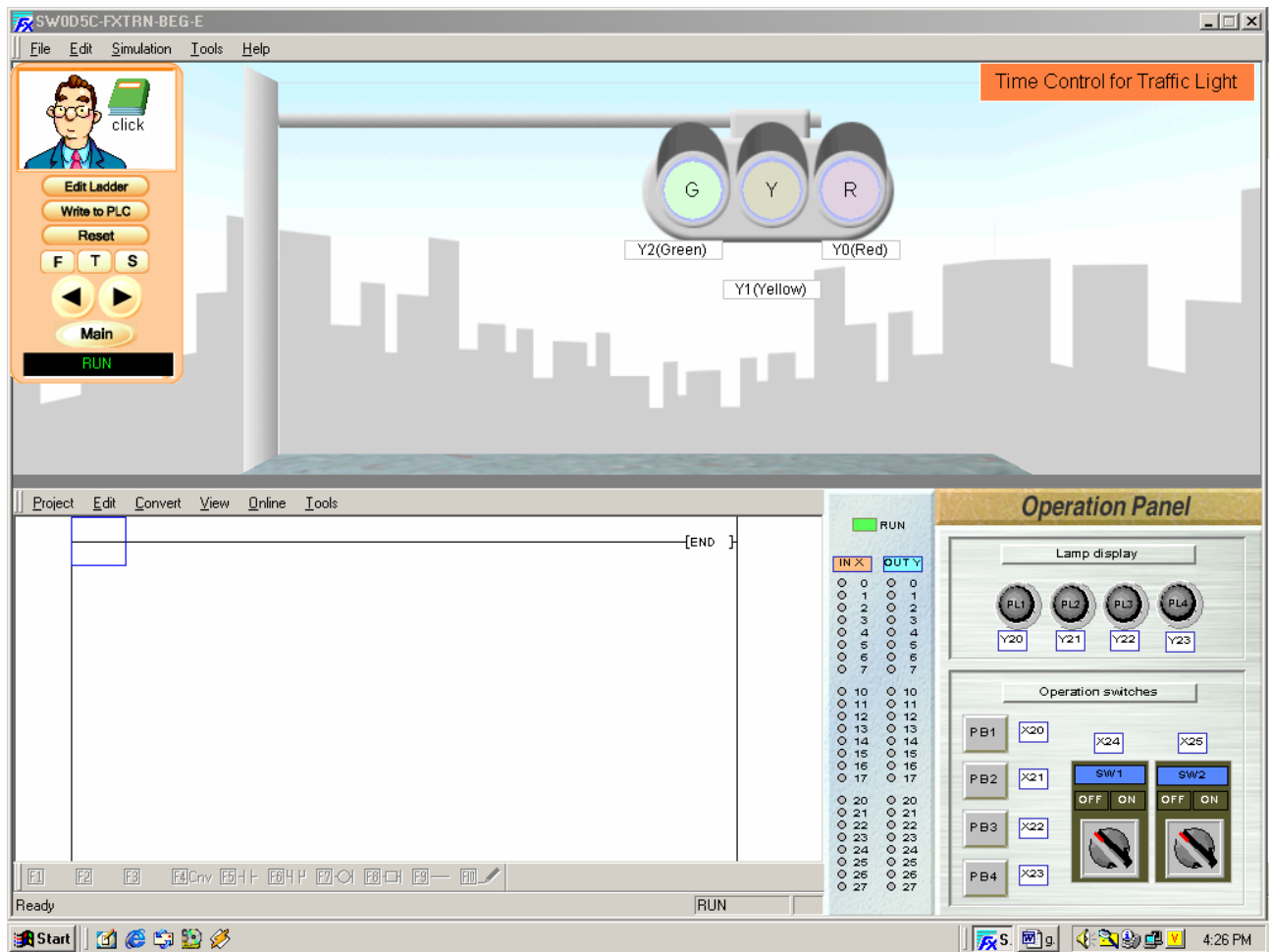


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X00		
	X01		
	X02		
	X03		
NGÕ RA	Y0		
	Y1		
	Y2		
	Y3		
	Y7		

**BÀI TẬP D3 : ĐIỀU KHIỂN ĐÈN GIAO THÔNG VỚI BA ĐÈN ĐƠN**

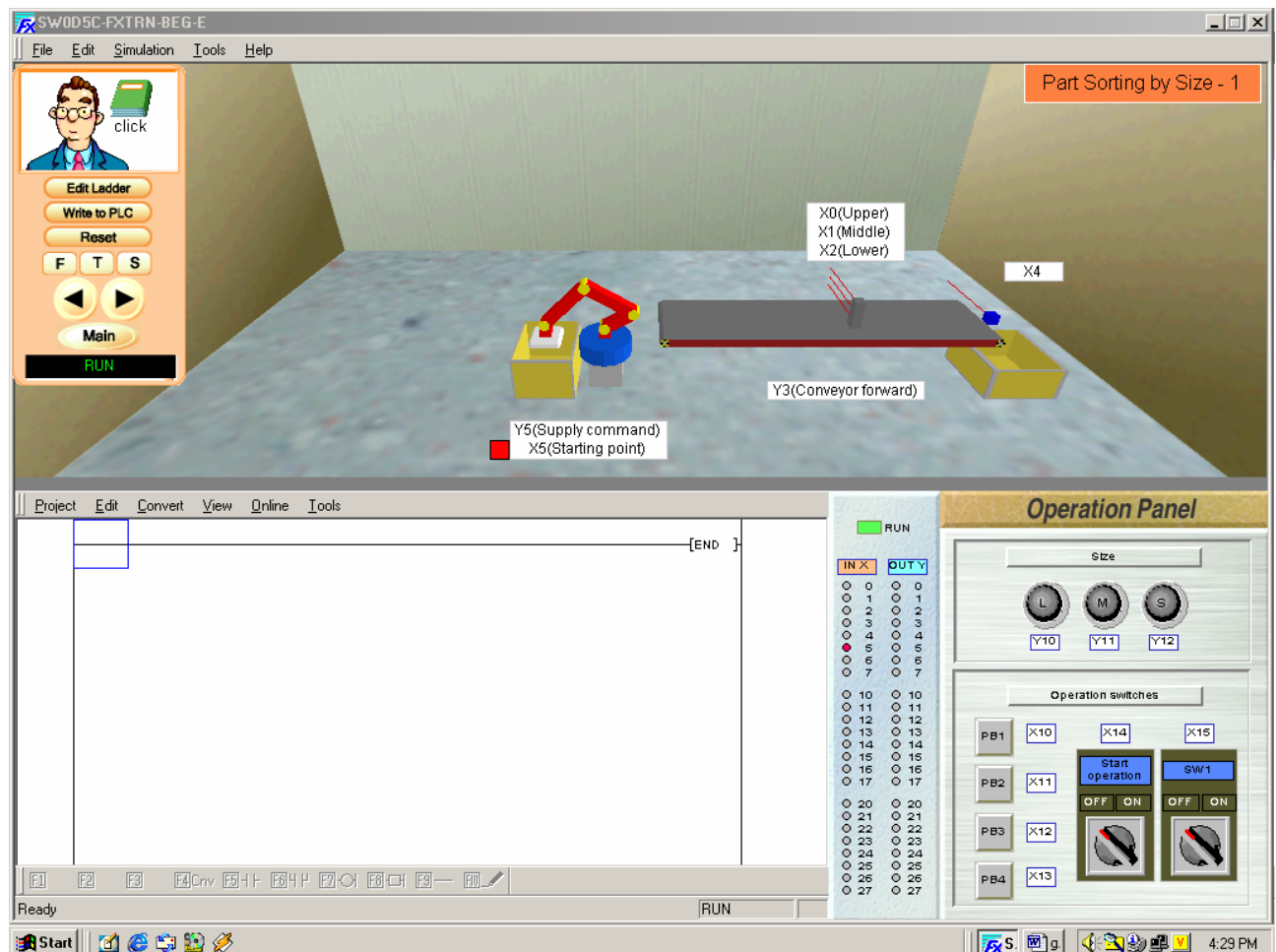


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X24		
	X25		
NGÕ RA	Y0		
	Y1		
	Y2		
	Y20		
	Y21		
	Y23		

**BÀI TẬP D4, D5 : ĐIỀU KHIỂN BĂNG TẢI VÀ ROBOT CUNG CẤP SẢN PHẨM**



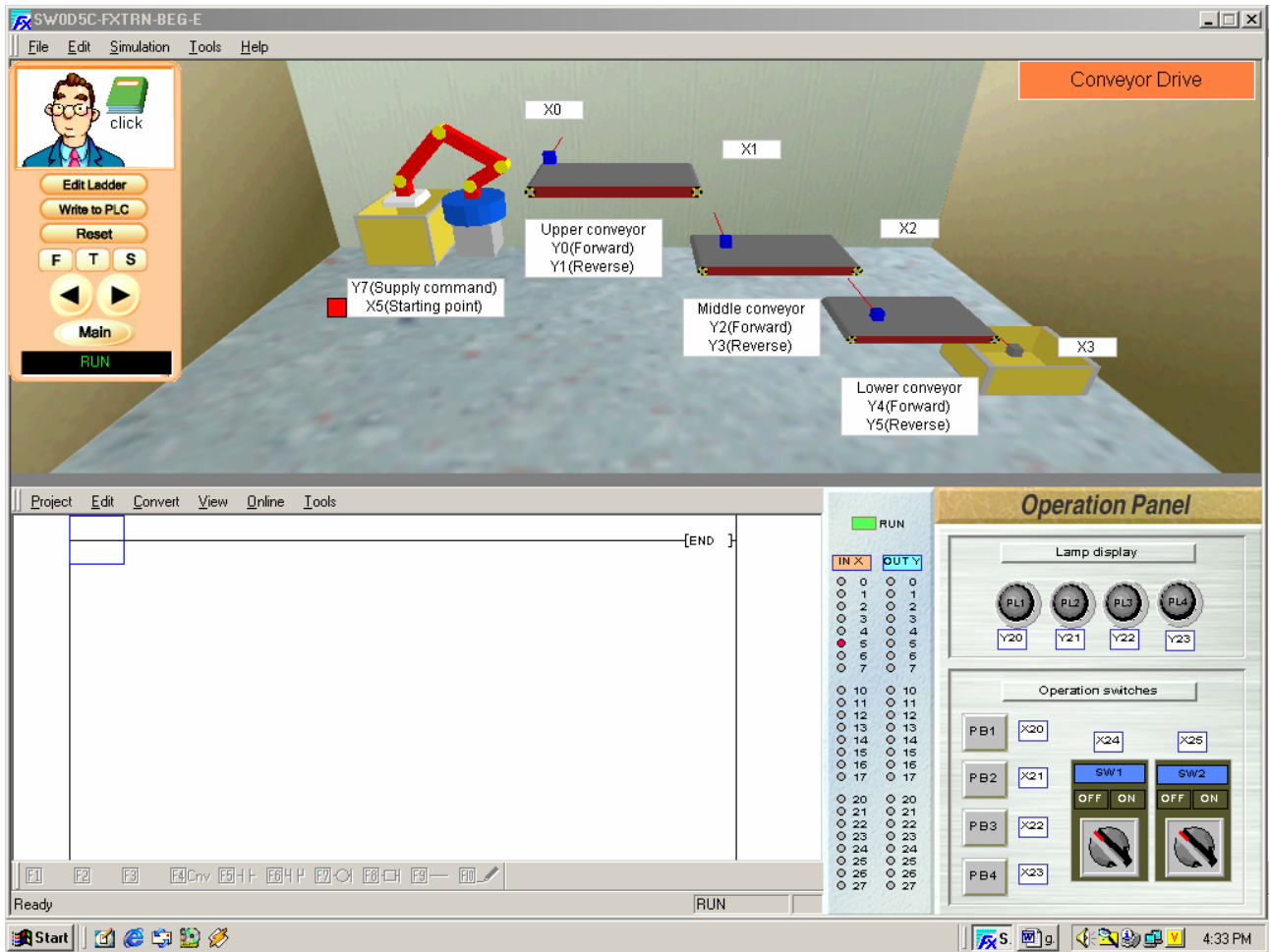


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO	X00		
	X01		
	X02		
	X04		
	X05		
	X20		
NGÕ RA	Y03		
	Y05		

**BÀI TẬP D6 : ĐIỀU KHIỂN KHỞI ĐỘNG VÀ DỪNG BĂNG TẢI**

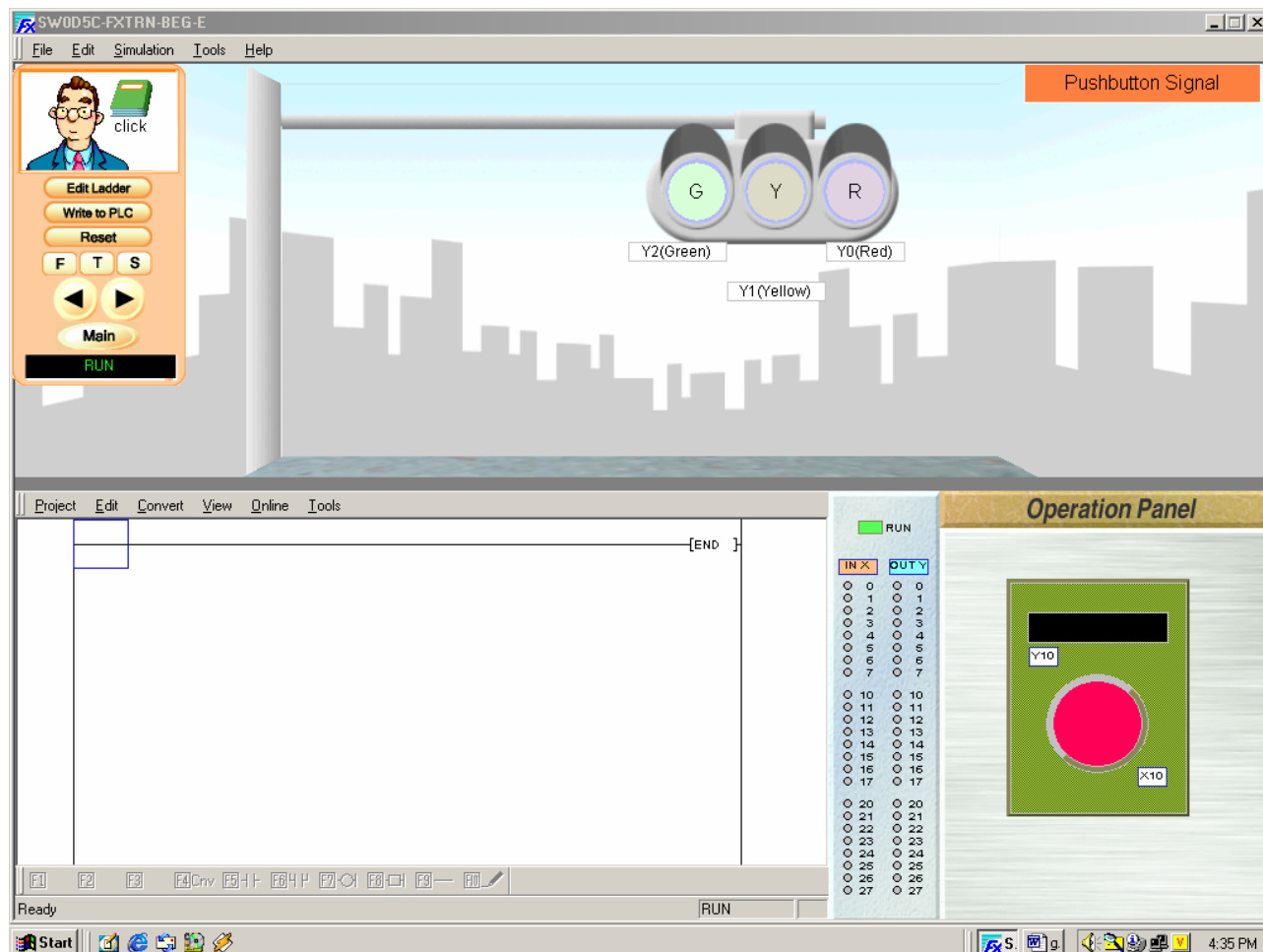


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

**BÀI TẬP E1 : ĐIỀU KHIỂN ĐÈN GIAO THÔNG VỚI BA ĐÈN ĐƠN DÙNG NÚT NHẤN**

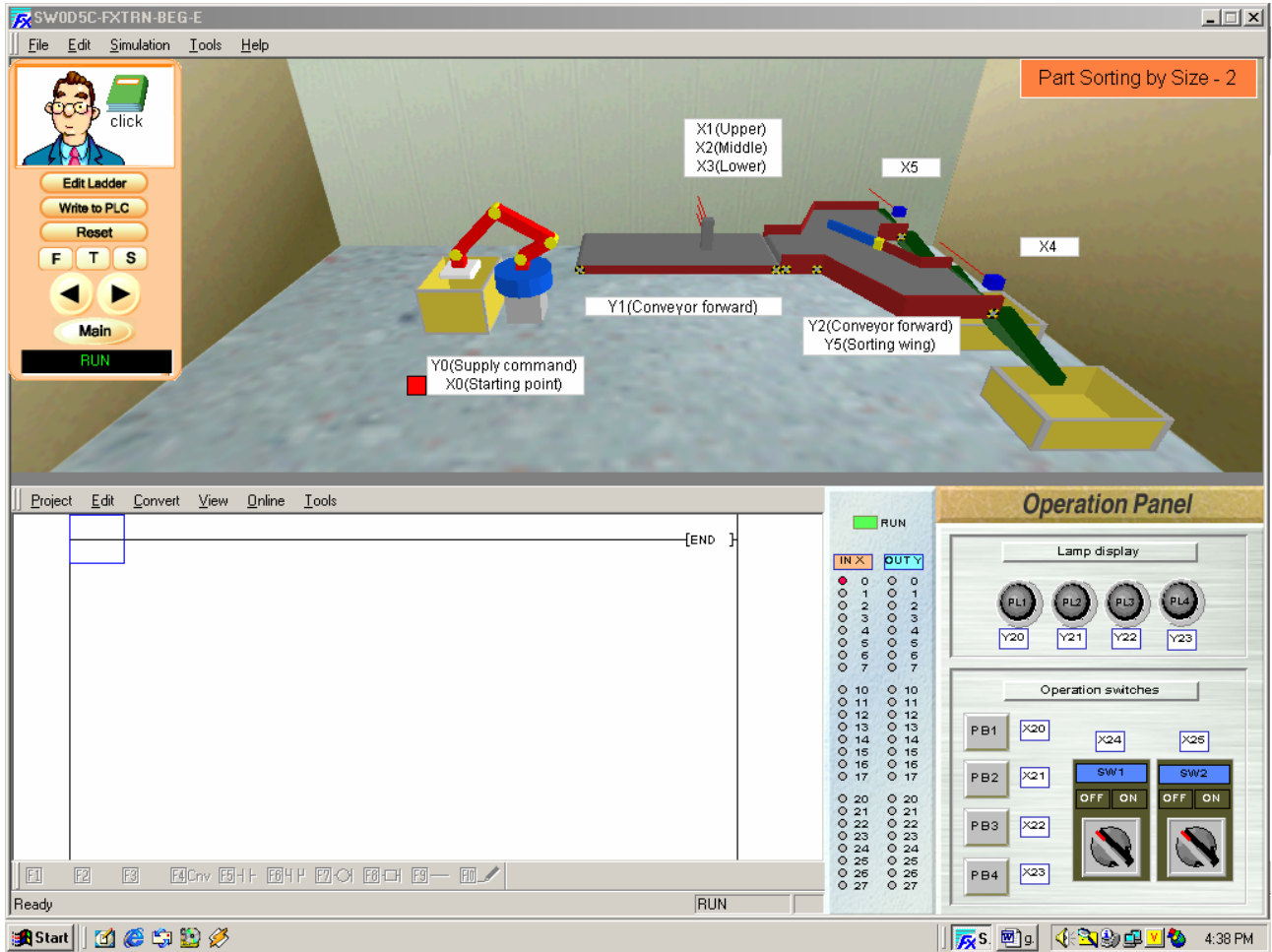


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

### Bảng Thiết bị cần dùng (device Assignment)

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

### BÀI TẬP E2 : ĐIỀU KHIỂN BĂNG TRUYỀN Rẽ NHÁNH



Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

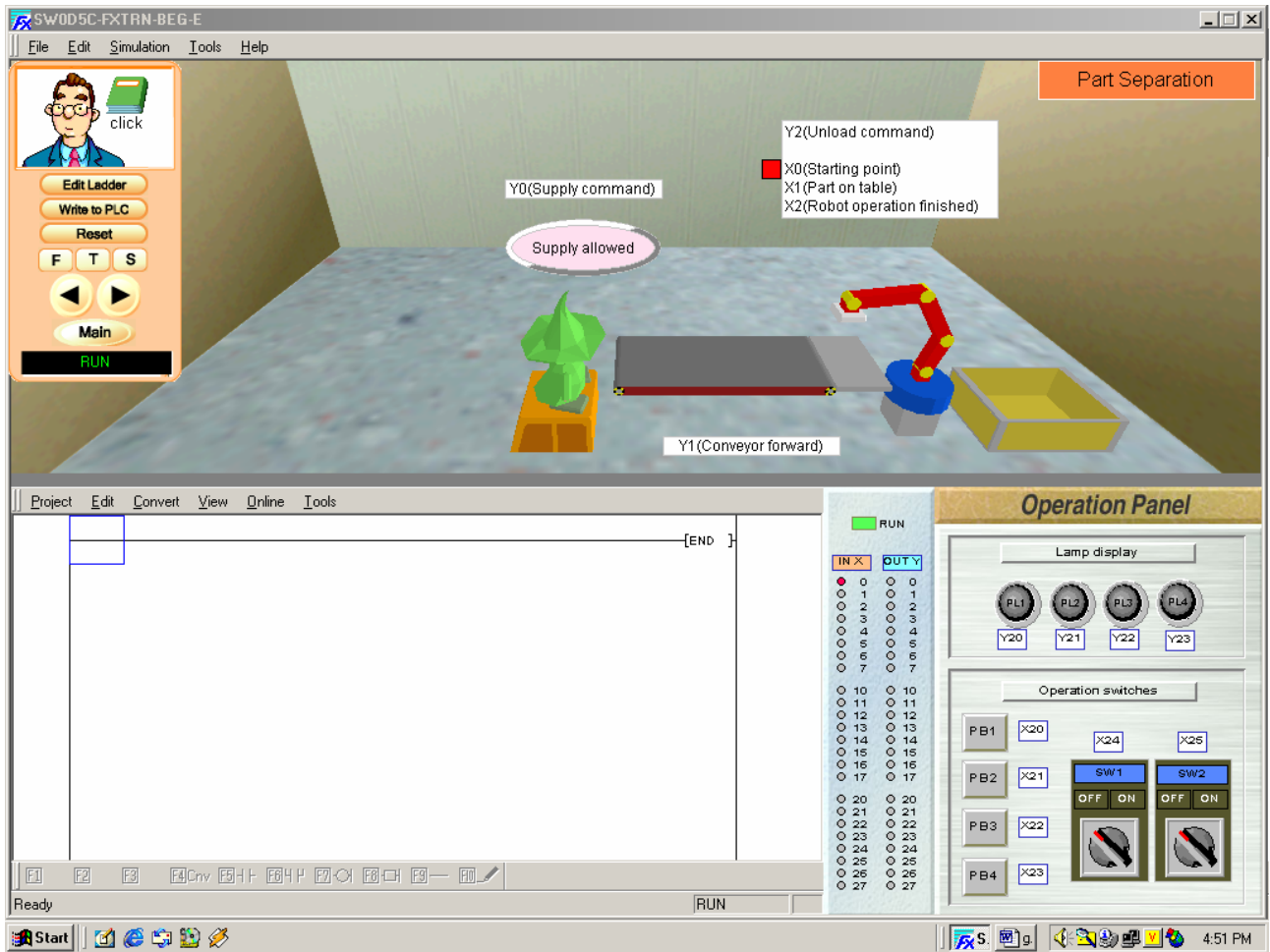
**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

Chương trình ladder bài tập E2



## BÀI TẬP E3 :ĐIỀU KHIỂN BĂNG TẢI VÀ ROBOT

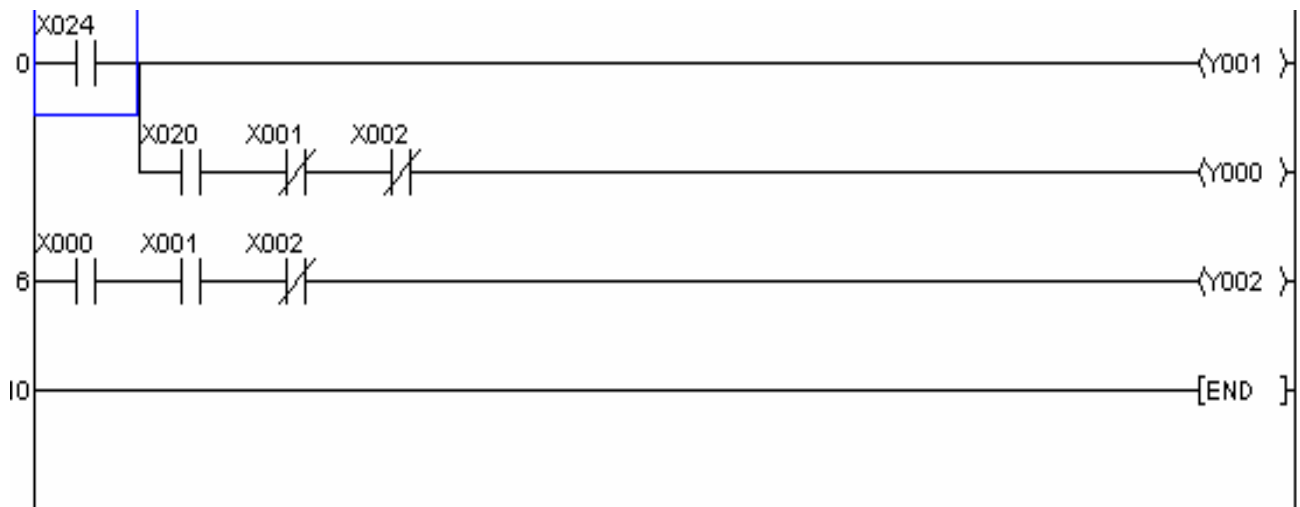


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

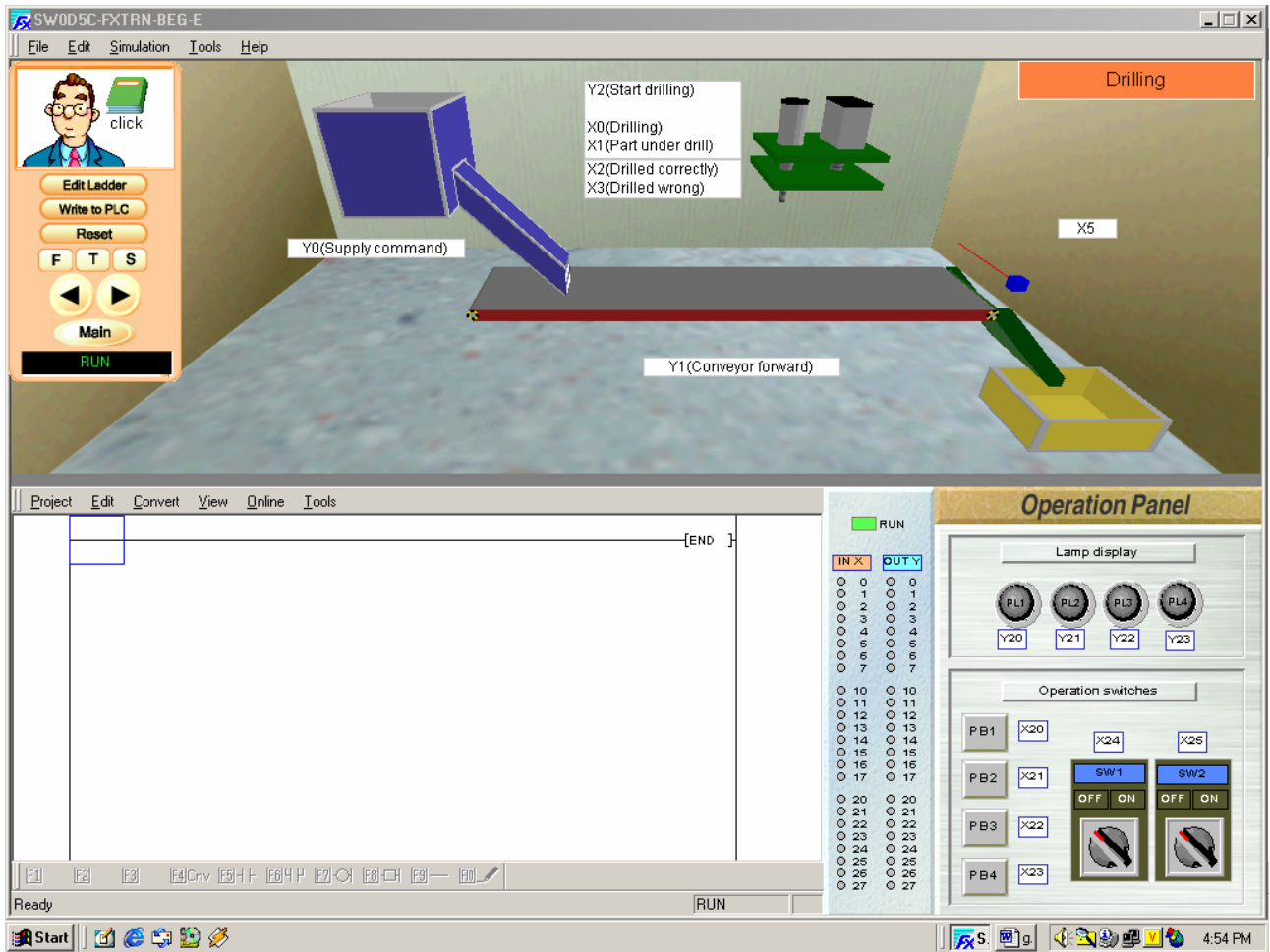
**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

Chương trình ladder bài tập E3



**BÀI TẬP E4 :ĐIỀU KHIỂN BĂNG TẢI MÁY KHOAN VÀ ROBOT**



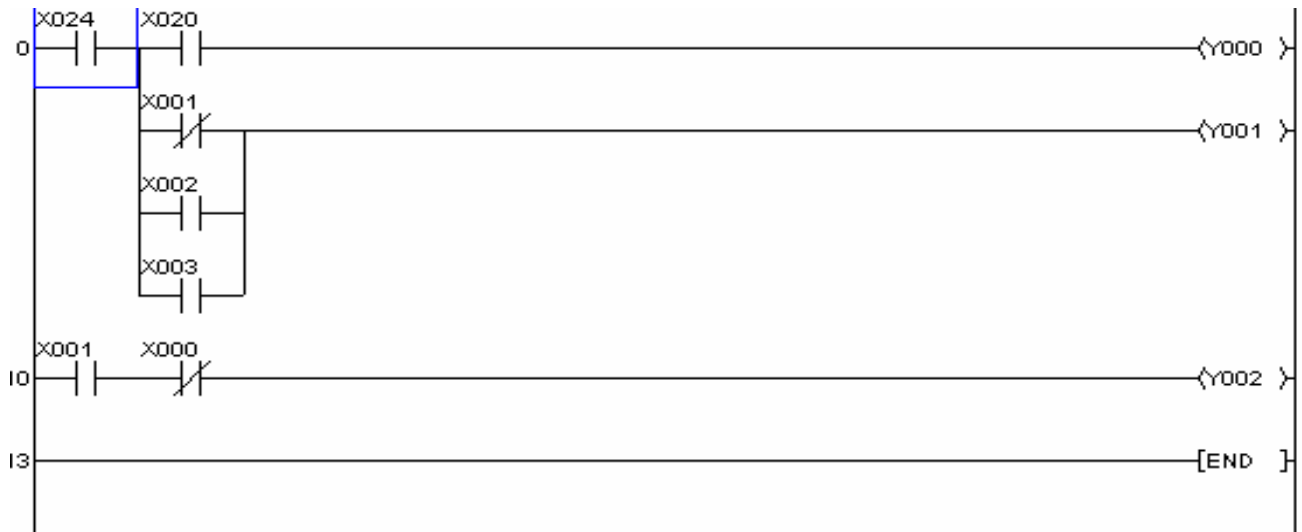
Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

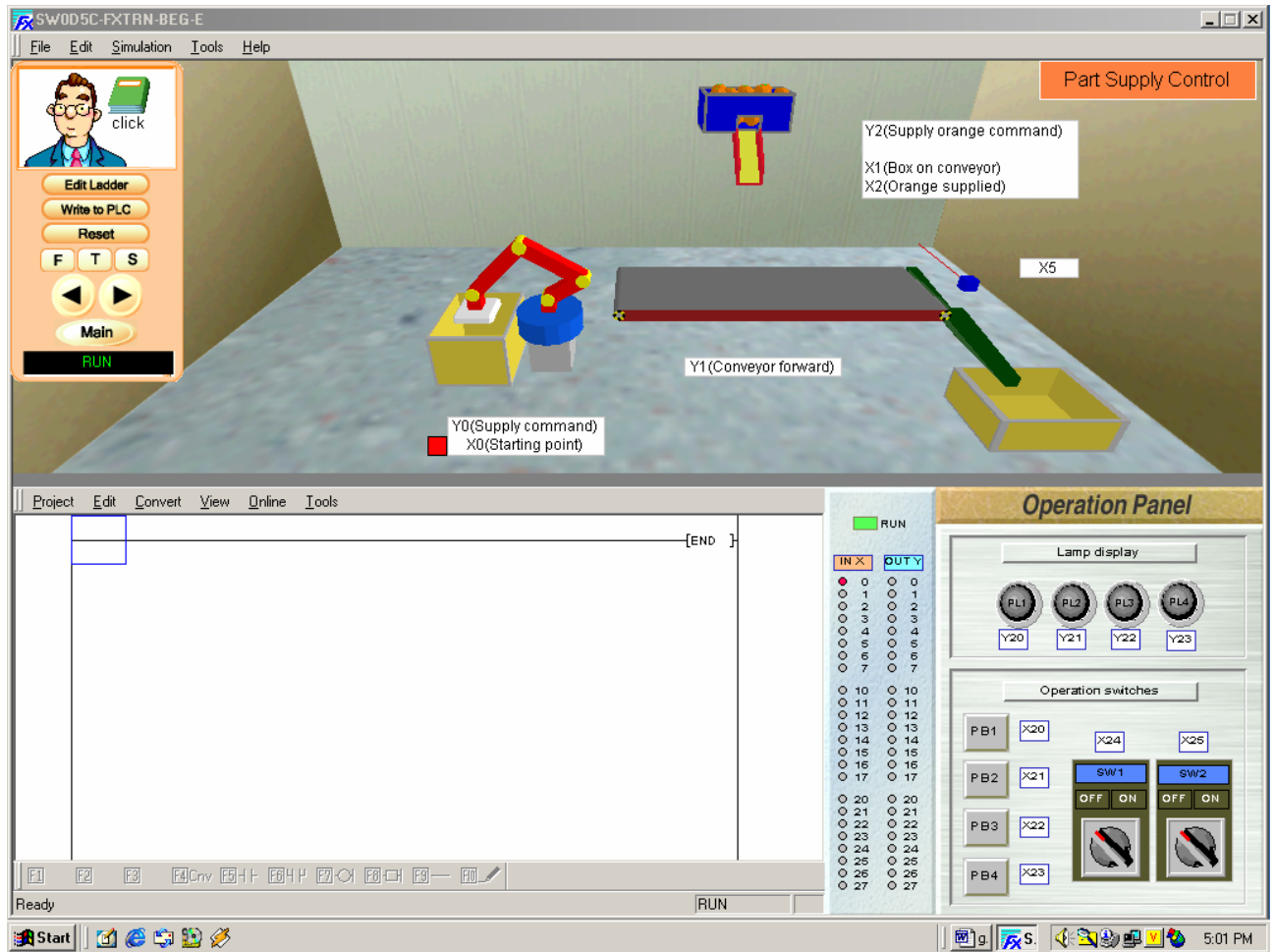
LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

Chương trình ladder bài tập E4





## BÀI TẬP E5 : ĐIỀU KHIỂN BĂNG TẢI VÀ ROBOT CUNG CẤP SẢN PHẨM

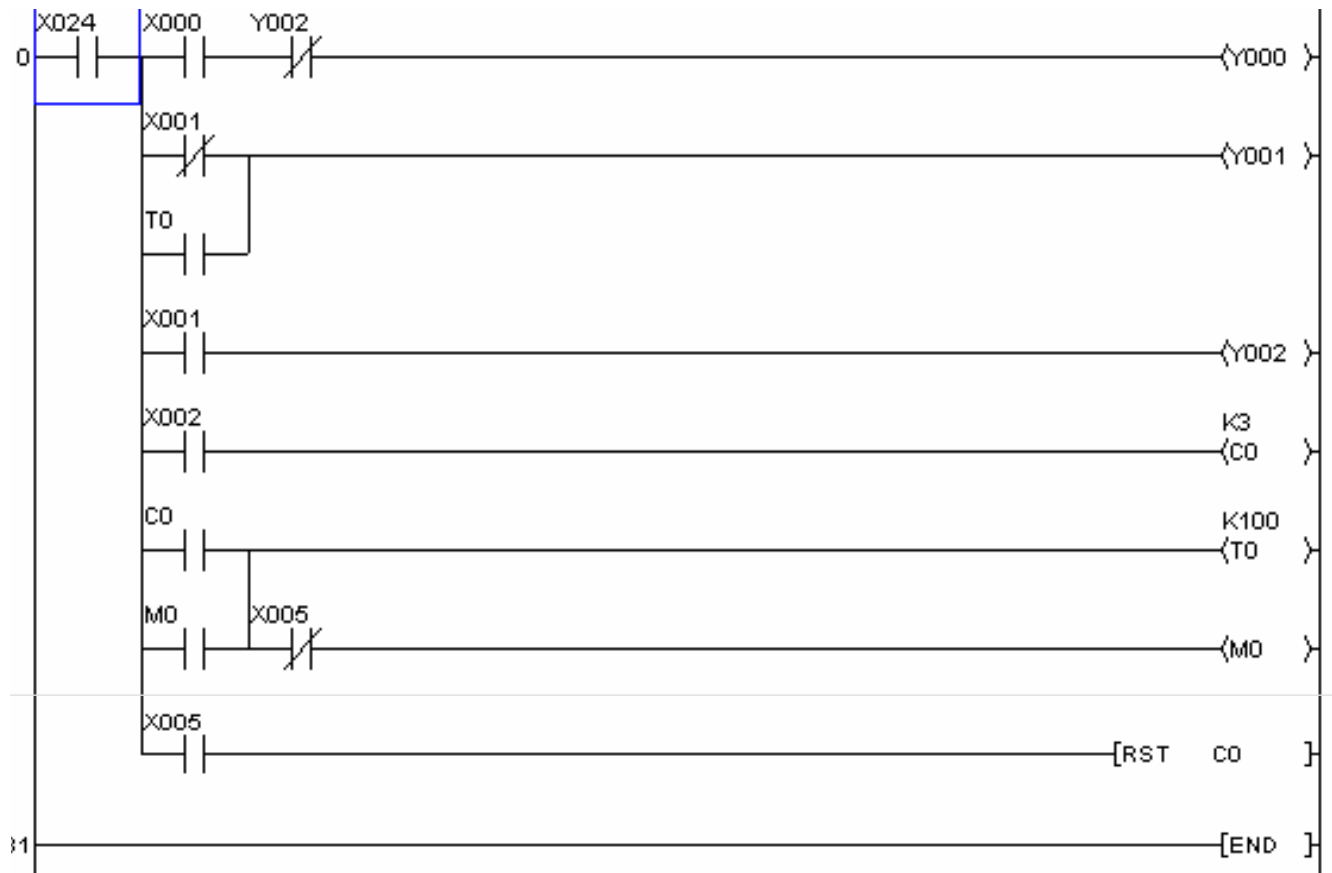


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

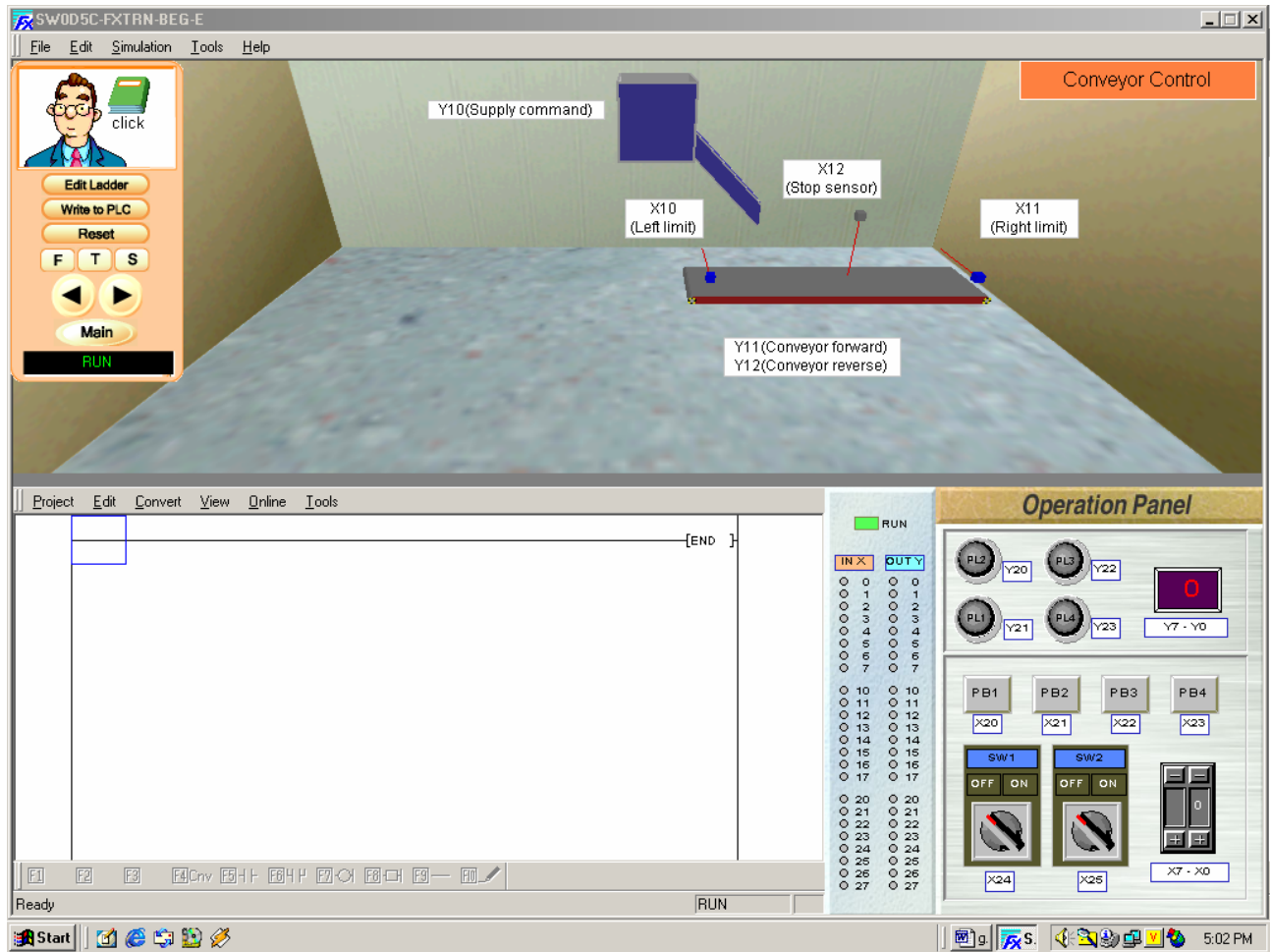
**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

Chương trình ladder bài tập E5



**BÀI TẬP E6 : ĐIỀU KHIỂN BĂNG TẢI VÀ ROBOT CUNG CẤP SẢN PHẨM**

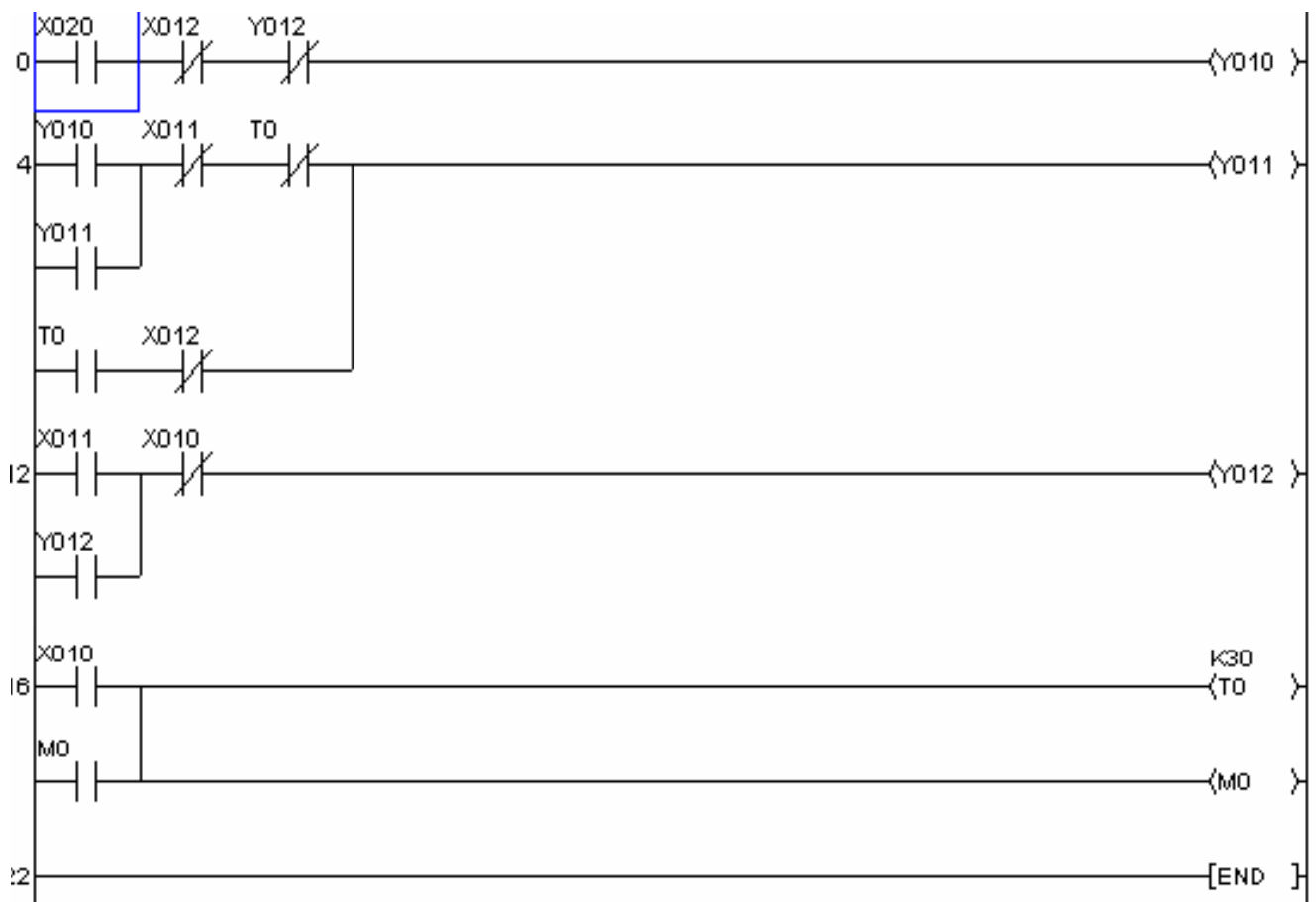


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

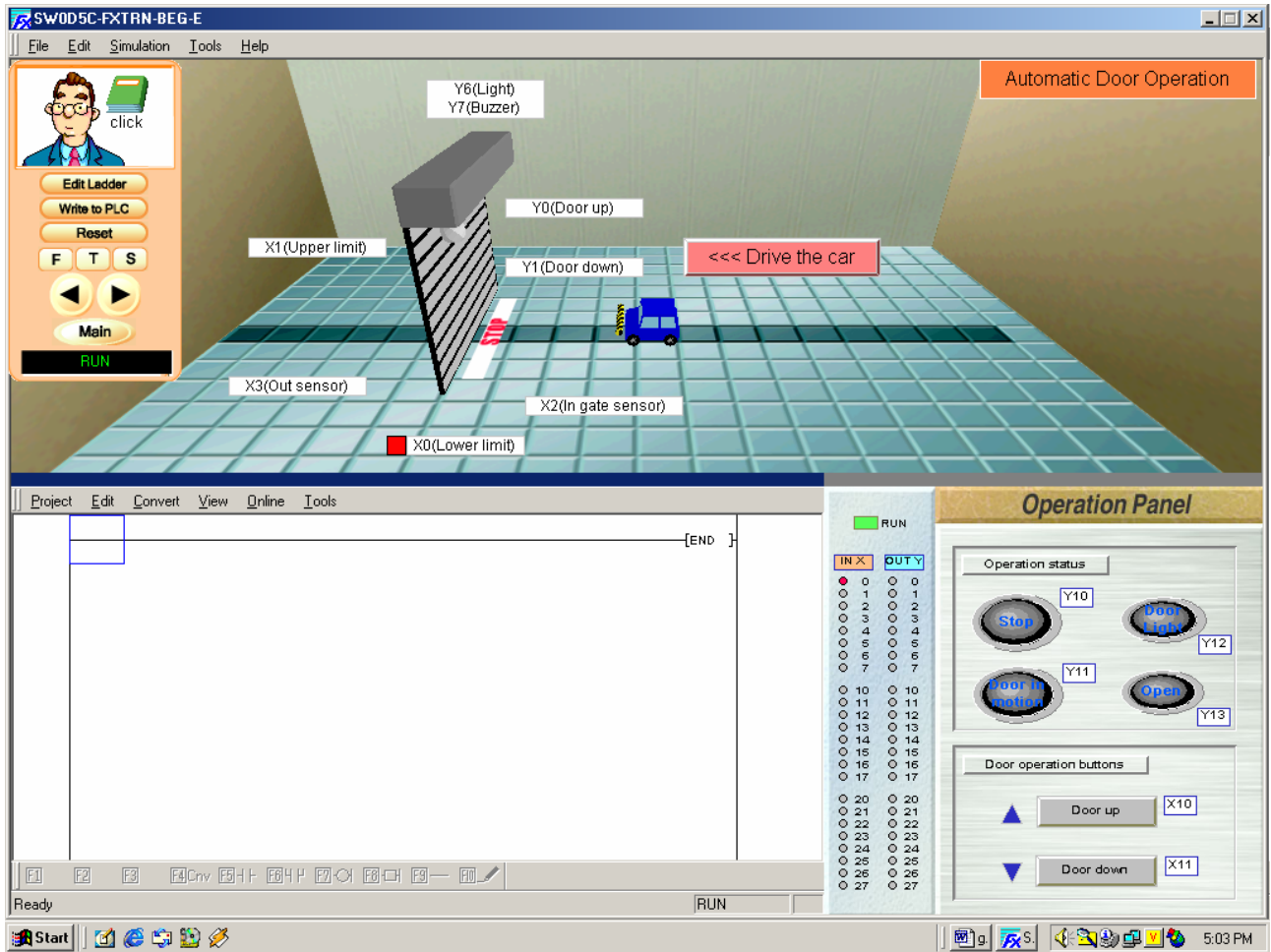
**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

Chương trình ladder bài tập E6



**BÀI TẬP F1 : ĐIỀU KHIỂN CỬA**

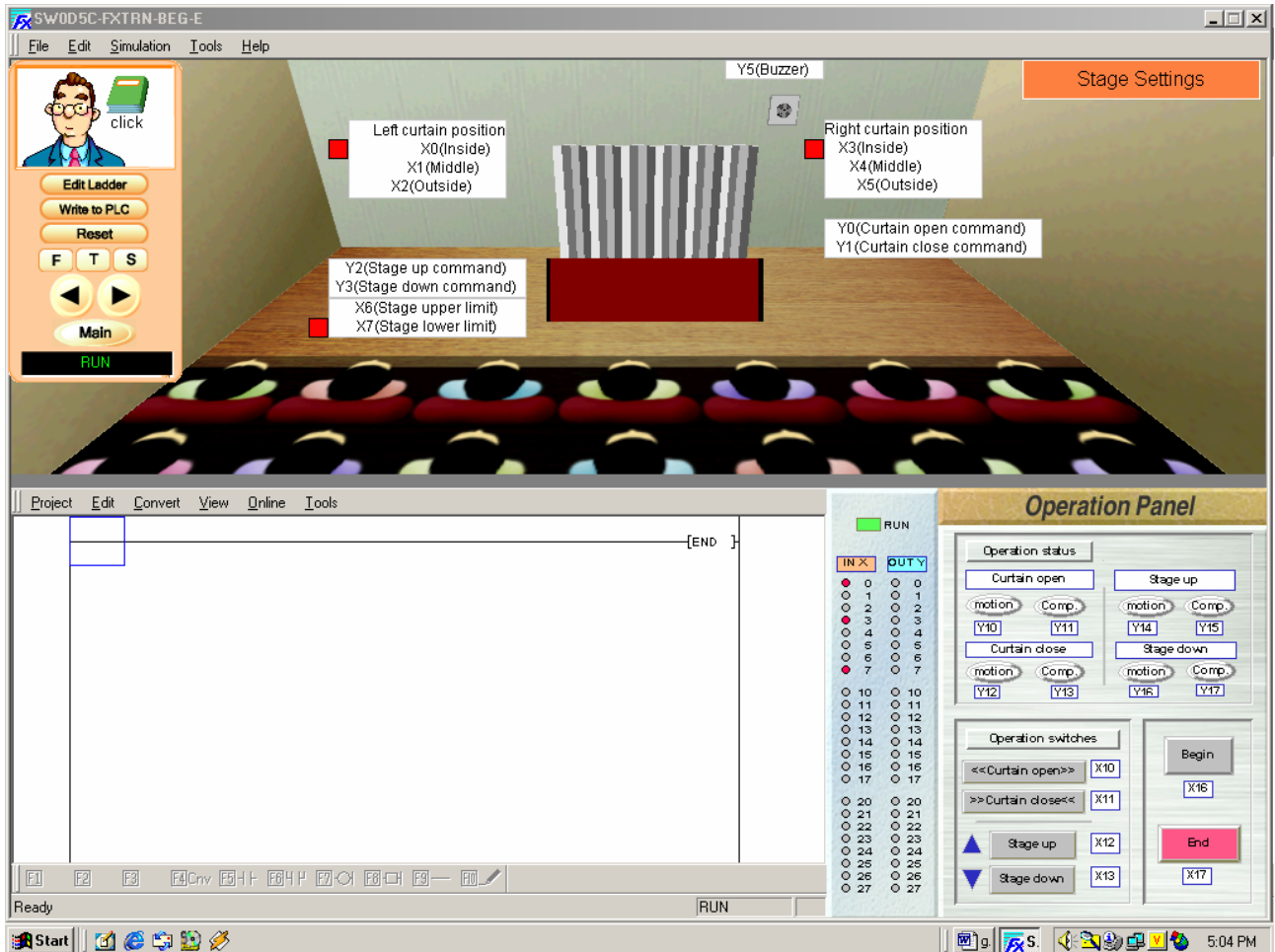


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

**BÀI TẬP F2 : ĐIỀU KHIỂN CÁC CƠ CẤU DỪNG**

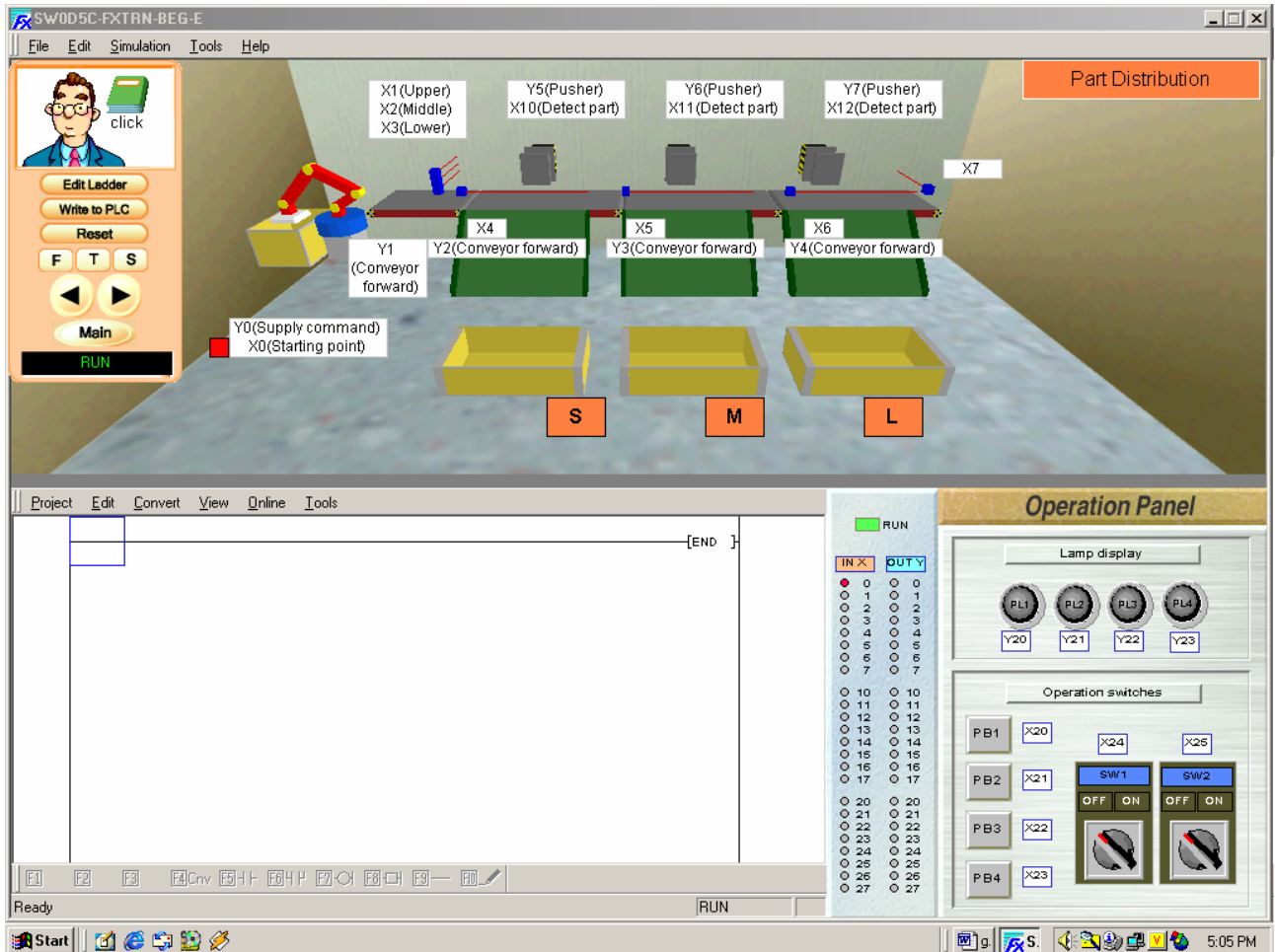


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

**BÀI TẬP F3 : ĐIỀU KHIỂN BĂNG TẢI VÀ PITTÔNG**



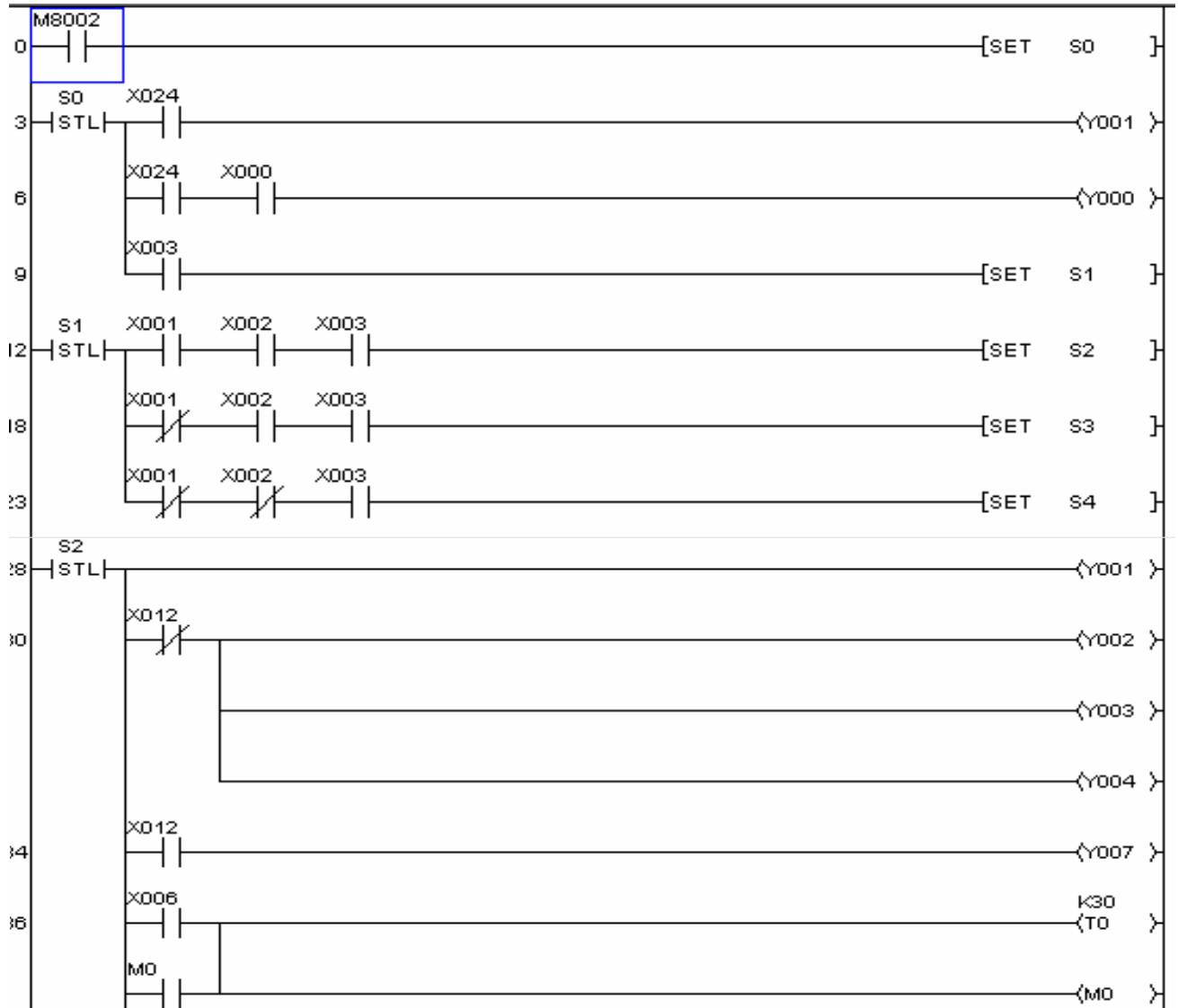
Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

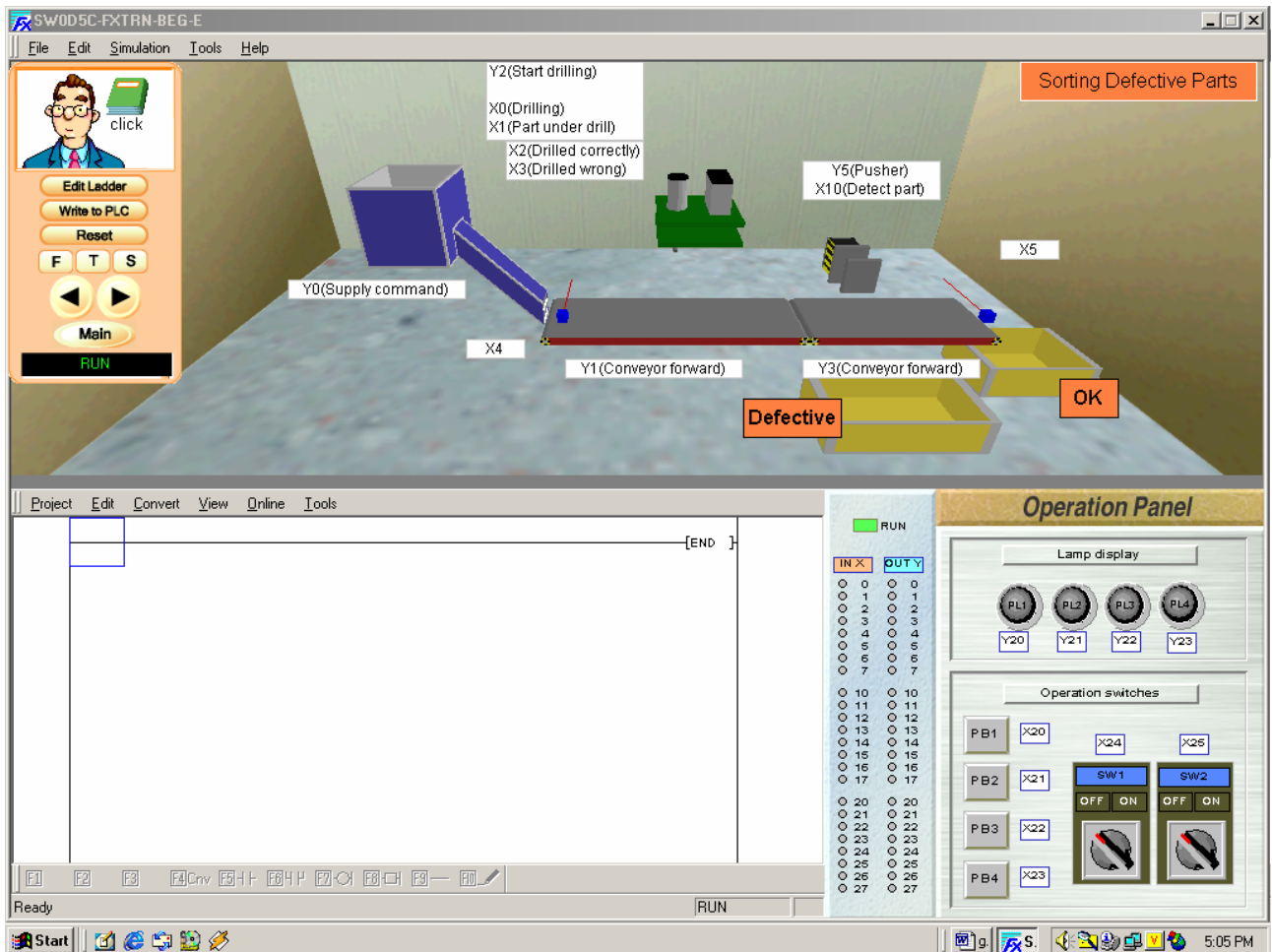


Chương trình ladder bài tập F3





**BÀI TẬP F4 : ĐIỀU KHIỂN BĂNG TẢI MÁY KHOAN VÀ ROBOT**

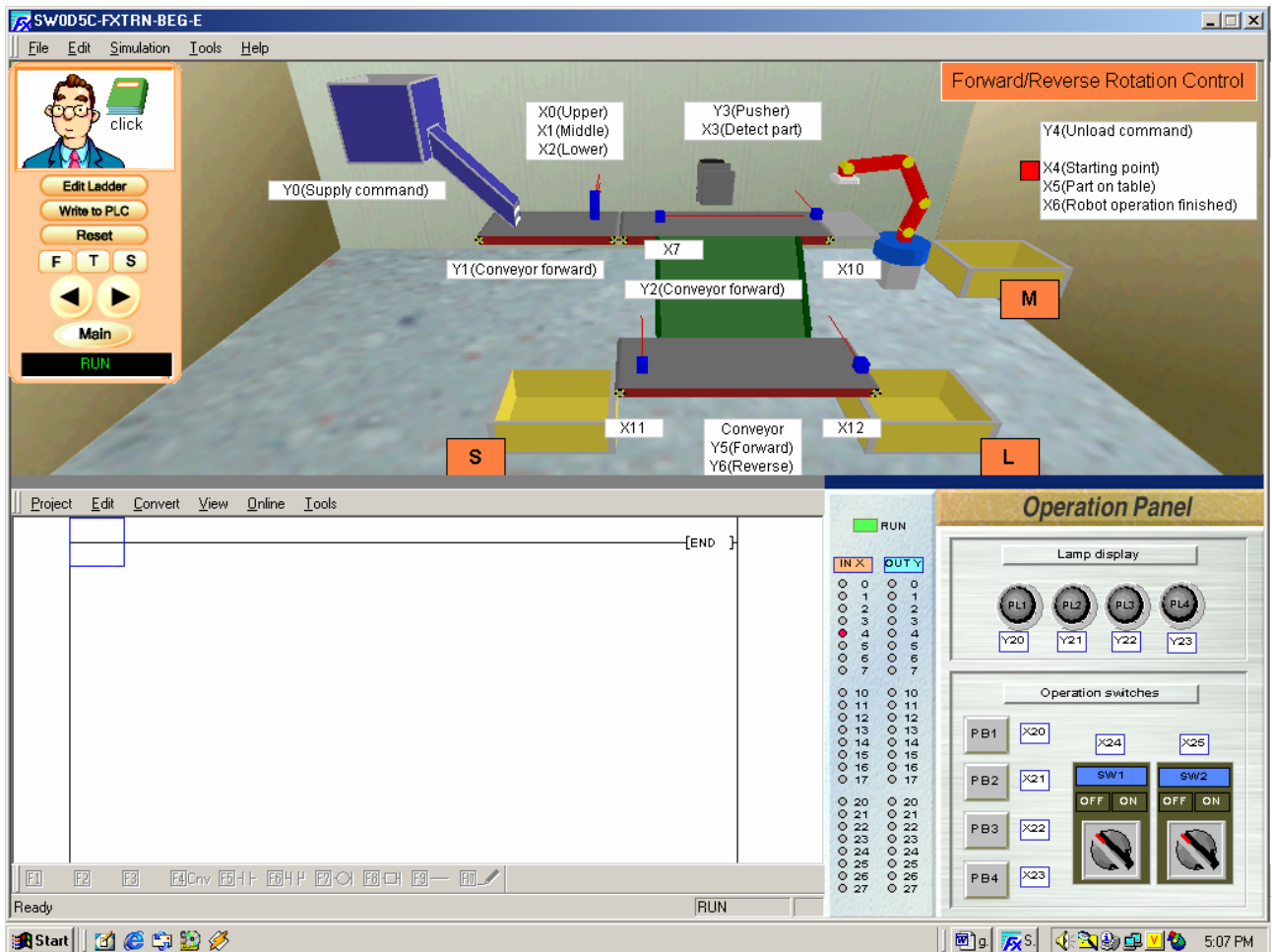


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

**BÀI TẬP F5 : ĐIỀU KHIỂN BĂNG TẢI MÁY KHOAN VÀ ROBOT**

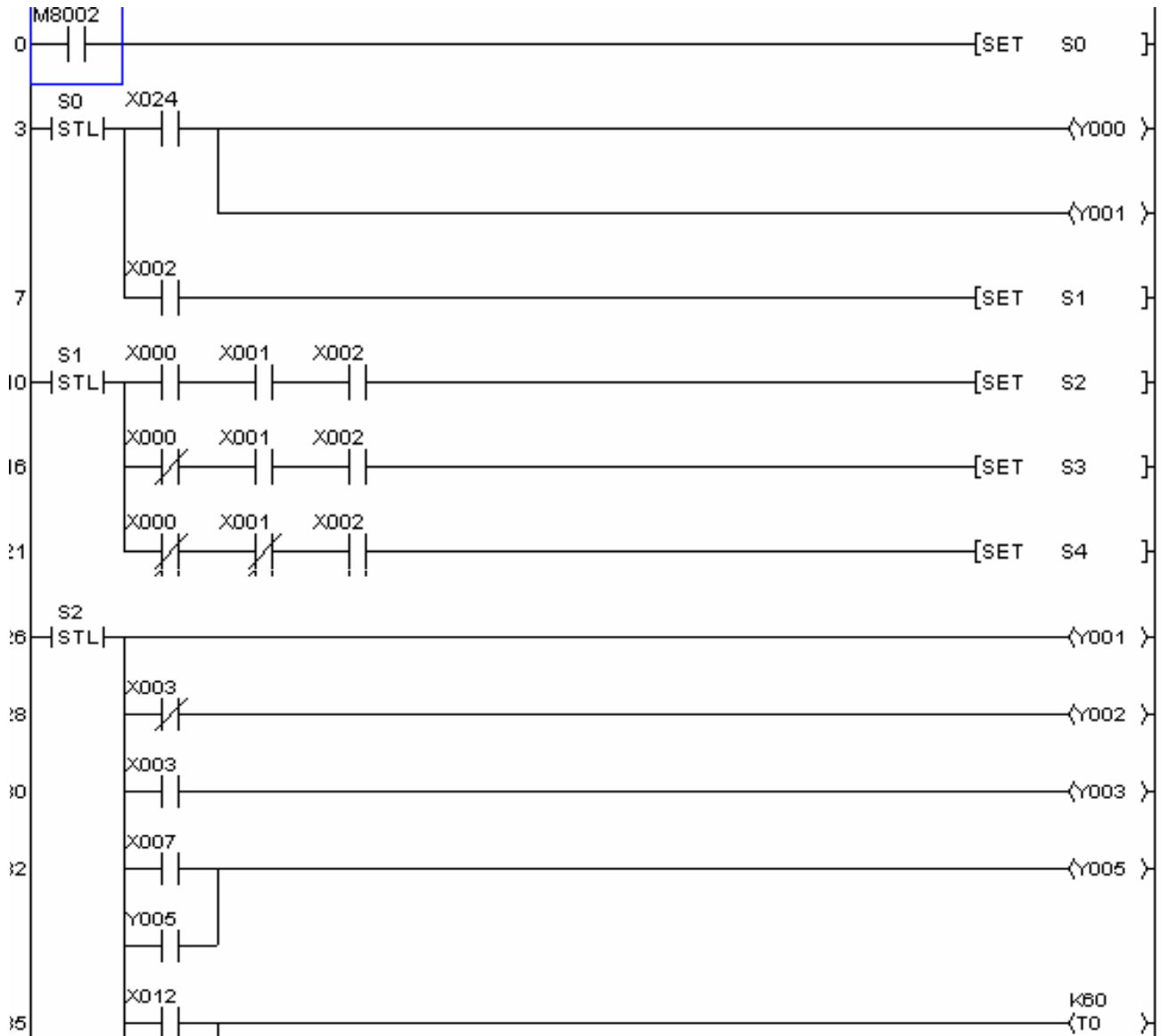


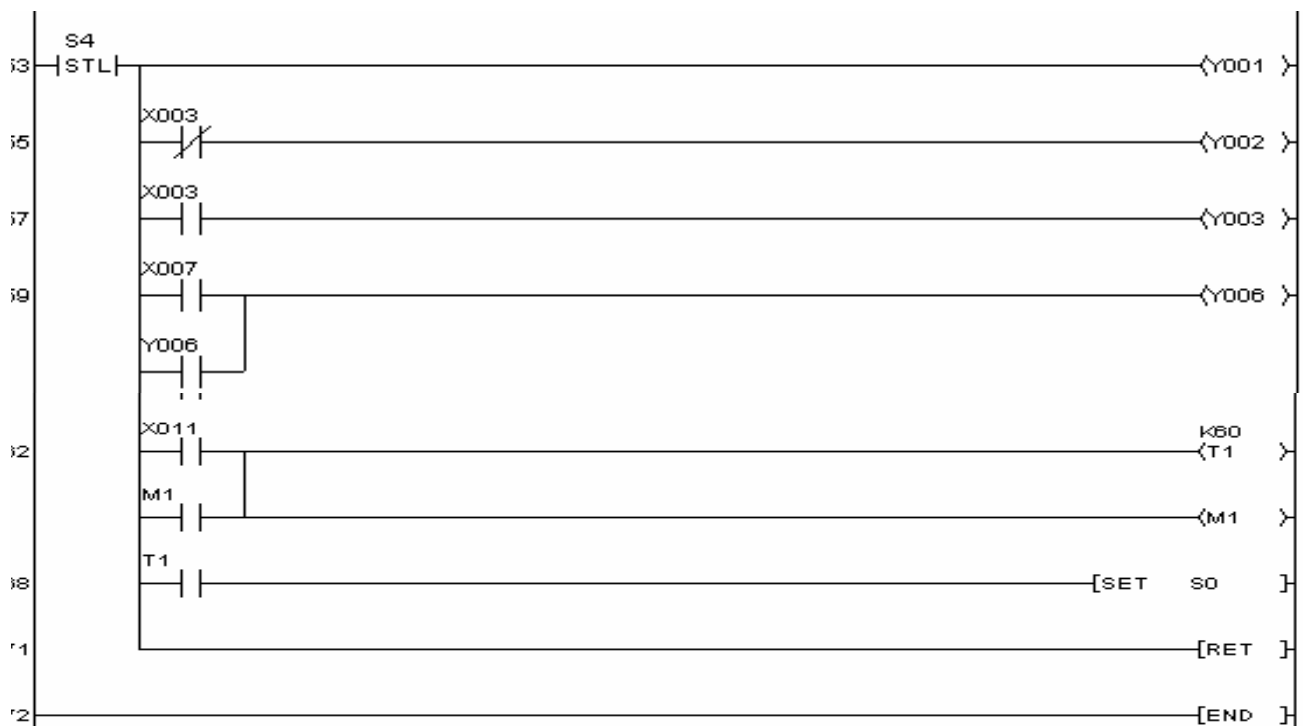
Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

**Bảng Thiết bị cần dùng (device Assignment)**

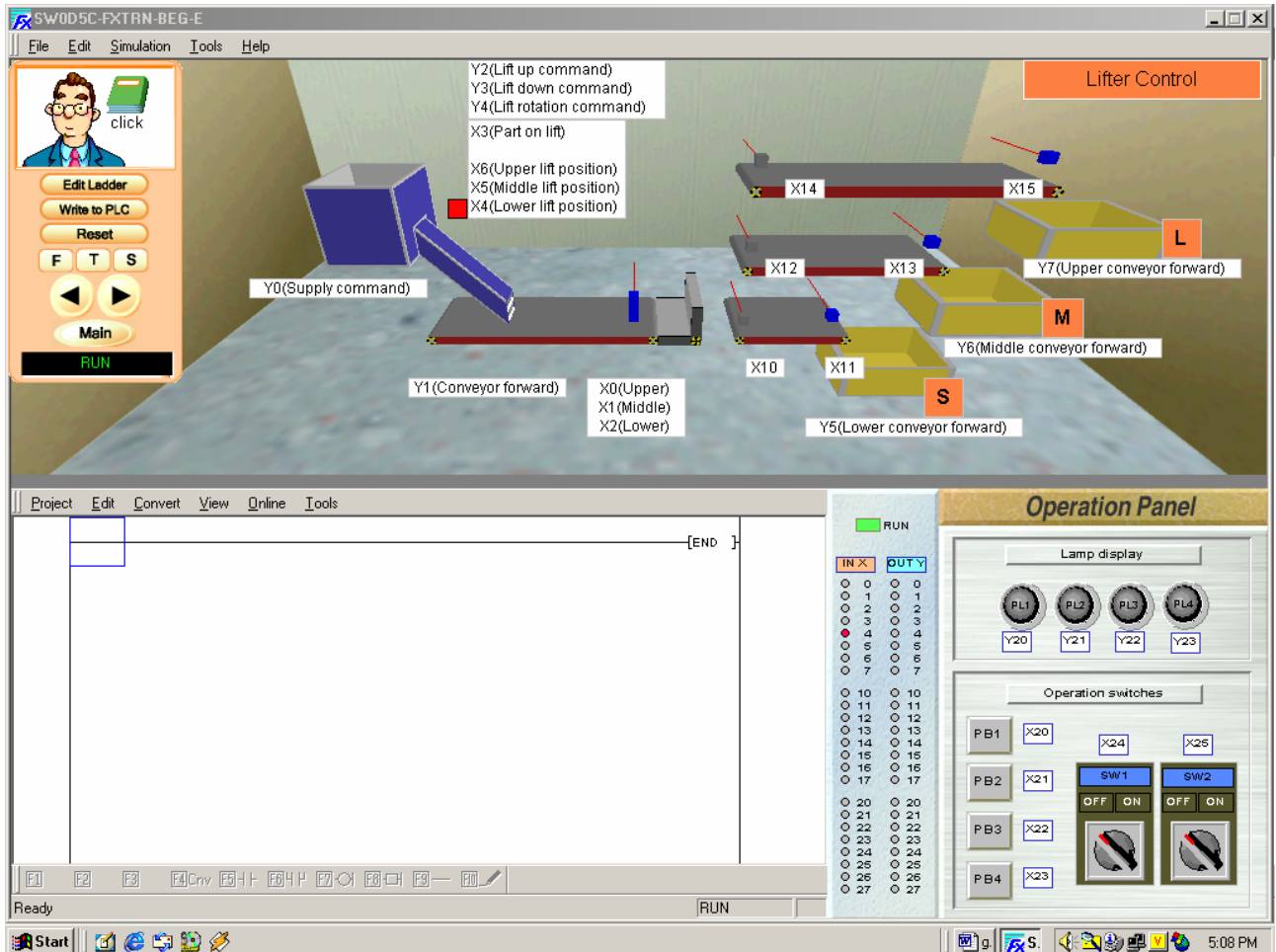
LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

Chương trình ladder bài tập F5





**BÀI TẬP F6 : ĐIỀU KHIỂN BĂNG TẢI VÀ THANG MÁY**

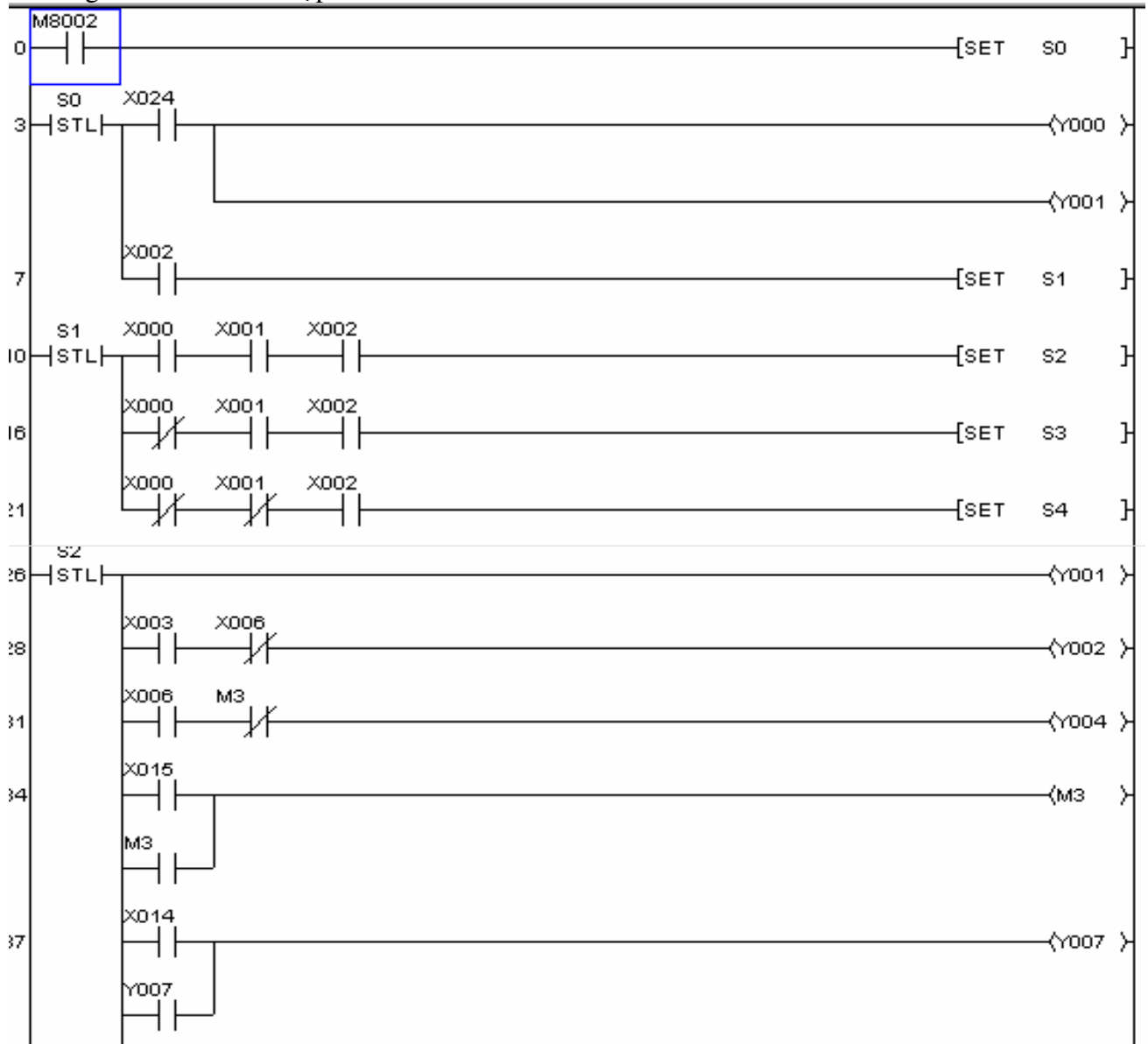


Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

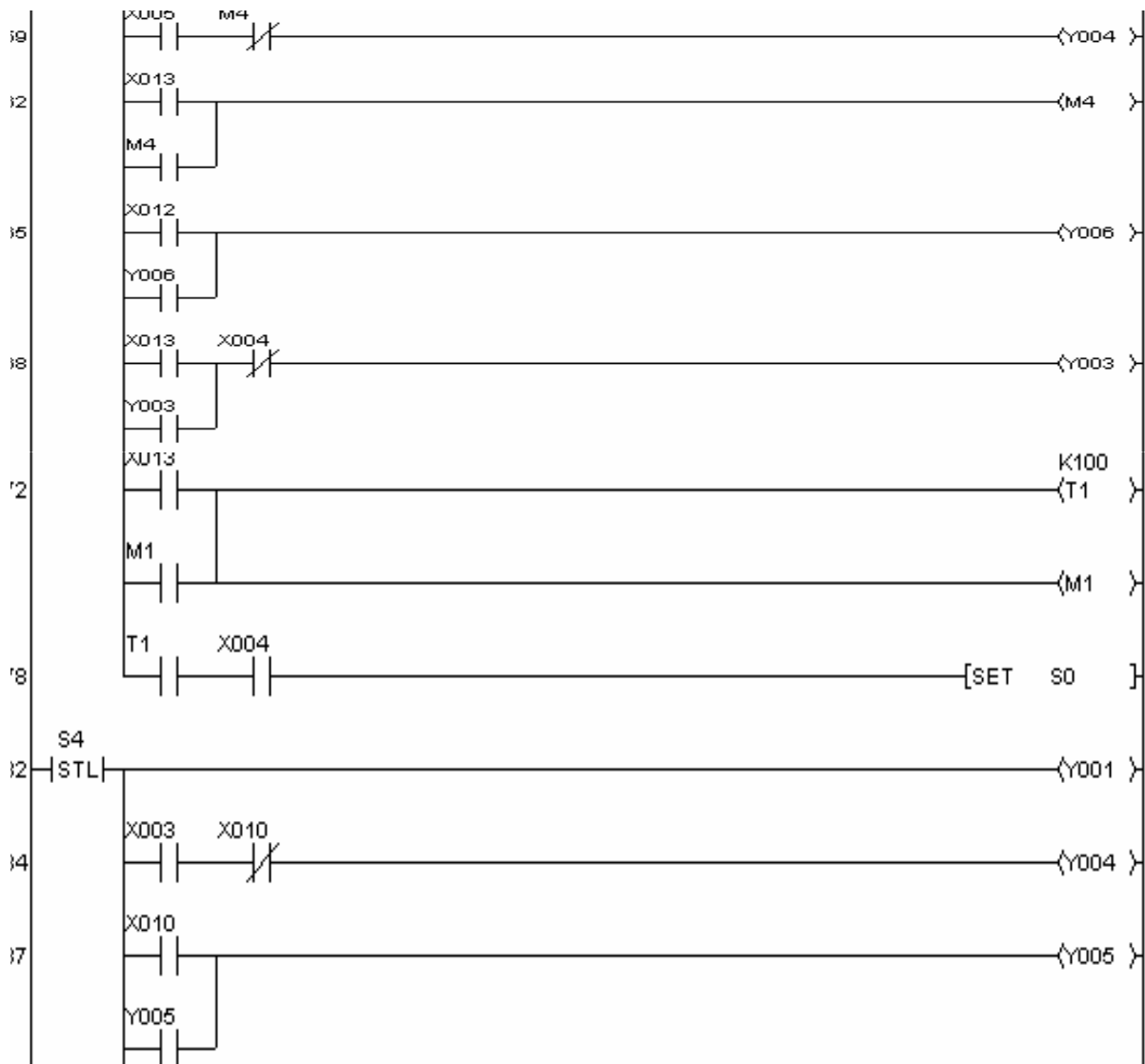
**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
NGÕ VÀO			
NGÕ RA			

Chương trình ladder bài tập F6







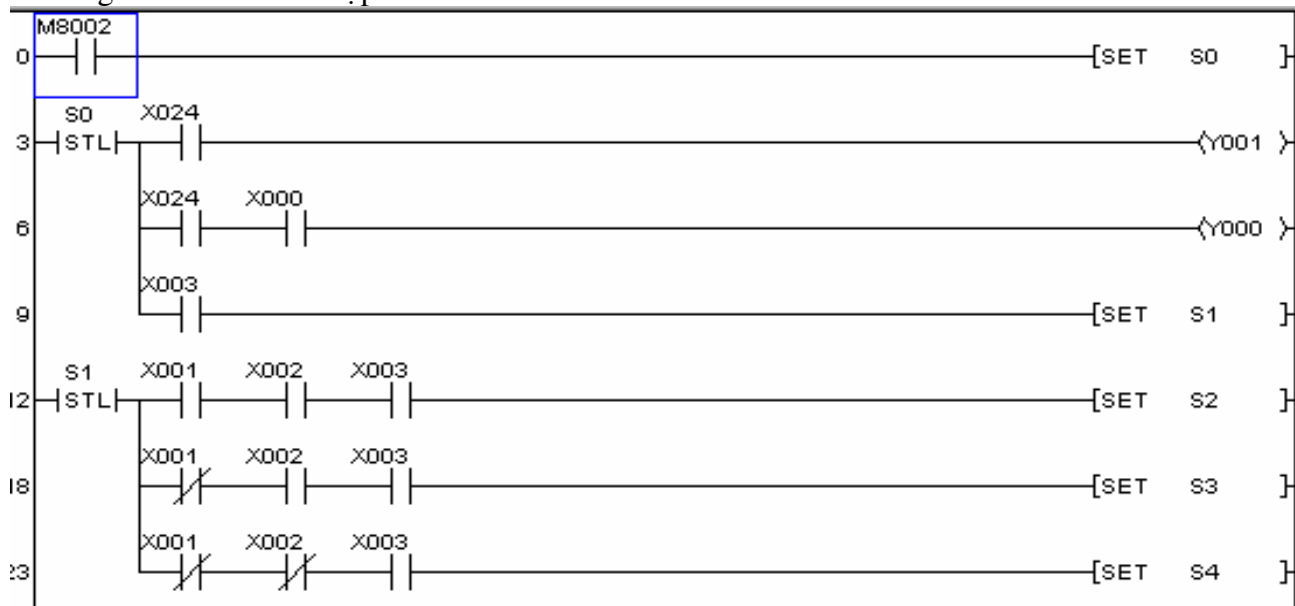


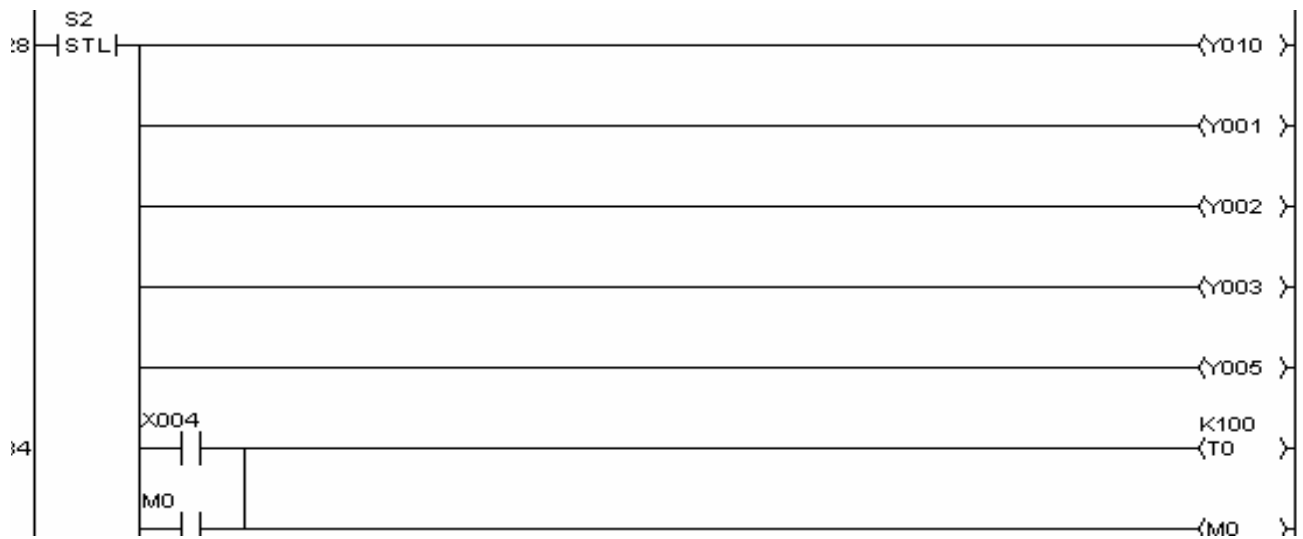
Hãy liệt kê và mô tả nguyên tắc hoạt động của các thiết bị , lập bảng trạng thái . Dùng ngôn ngữ ladder viết chương trình điều khiển , nạp chương trình vào PLC , cho mô hình vận hành.

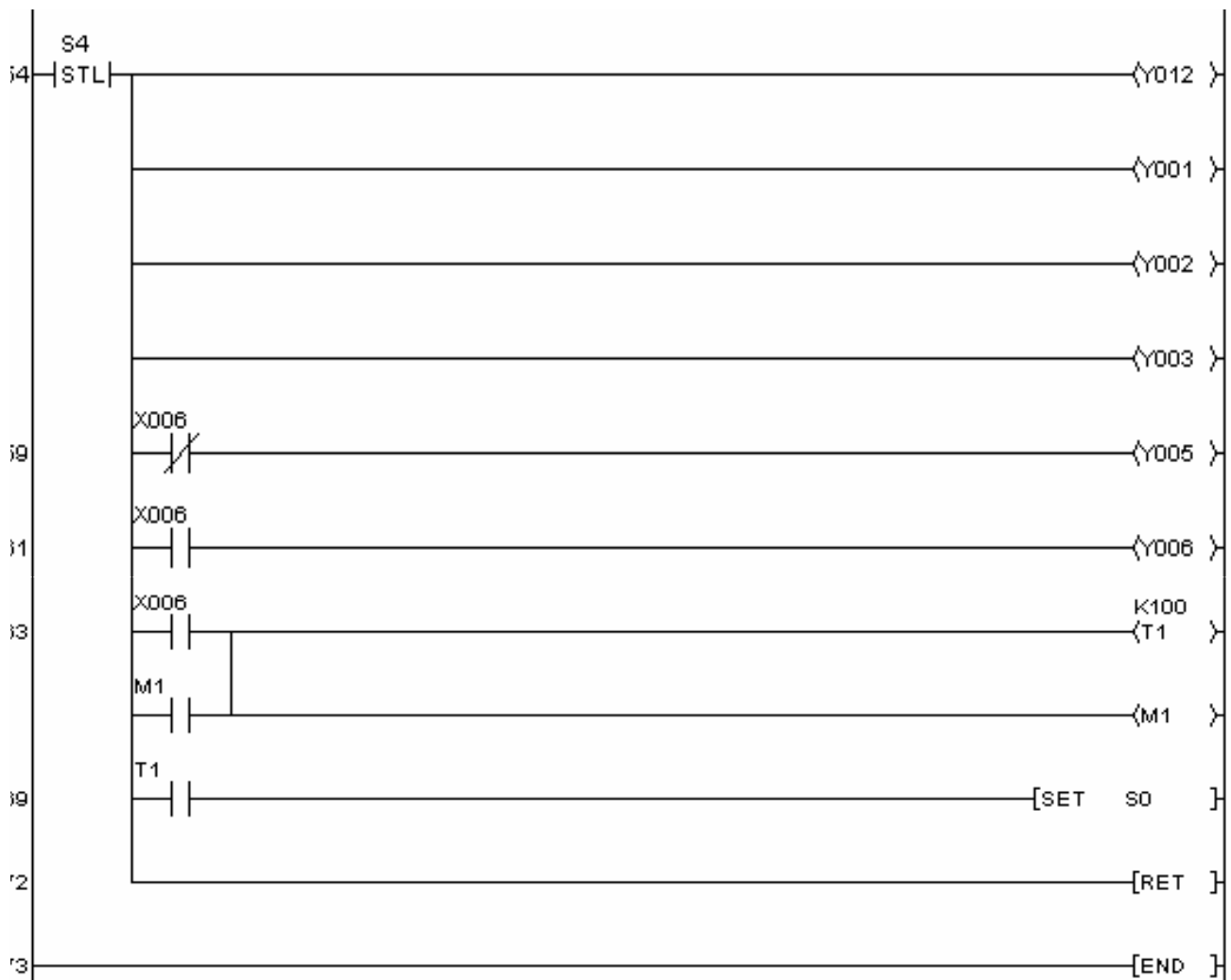
**Bảng Thiết bị cần dùng (device Assignment)**

LOẠI	ĐỊA CHỈ THIẾT BỊ	TÊN THIẾT BỊ	HOẠT ĐỘNG
<b>NGÕ VÀO</b>			
<b>NGÕ RA</b>			

Chương trình ladder bài tập F7

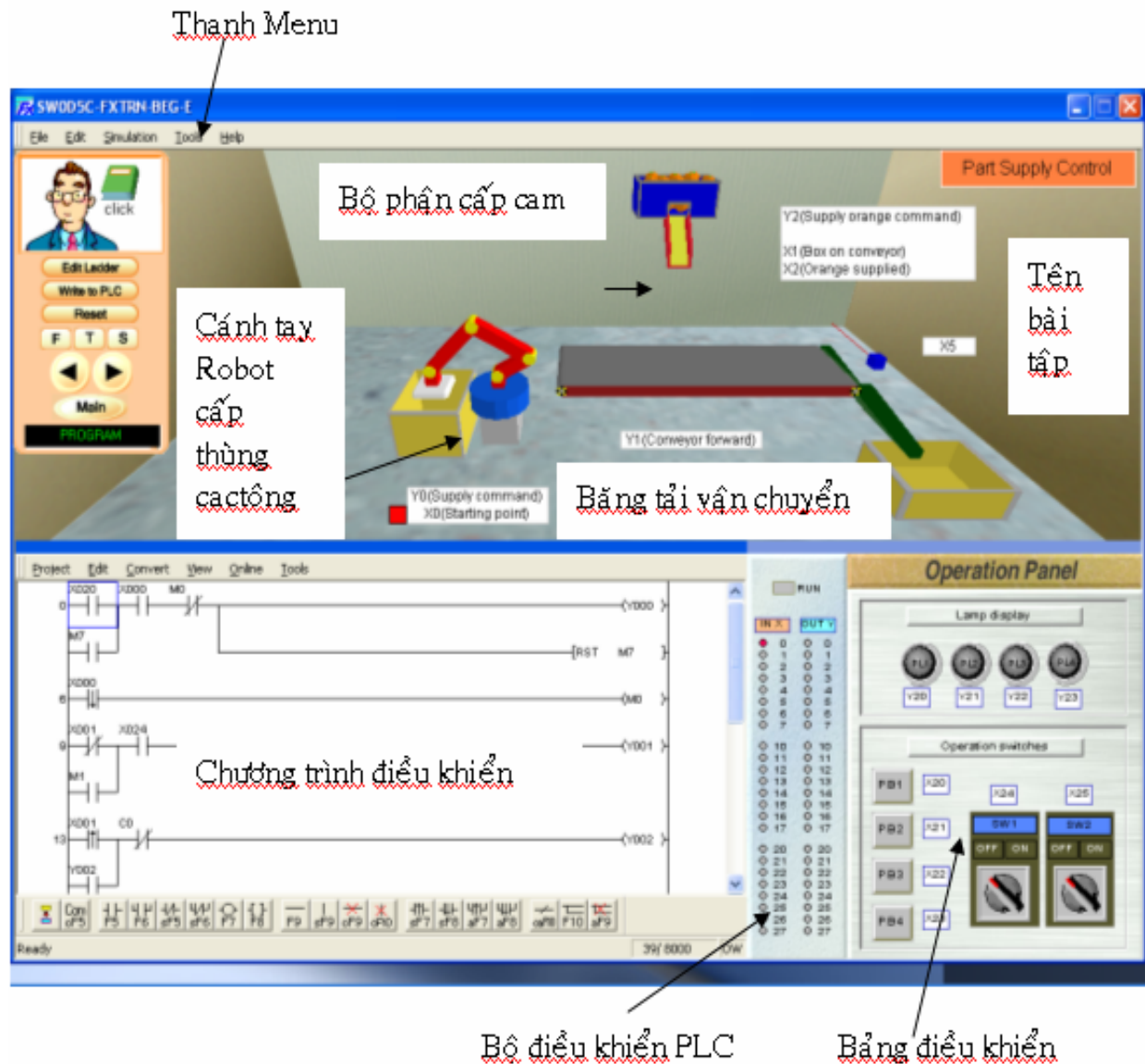






➤ Ví dụ 1: **CHƯƠNG TRÌNH MÔ PHỎNG E-5**

**I. Các thiết bị được sử dụng trong chương trình E-5:**



Cánh tay robot có nhiệm vụ cung cấp thùng cacton cho băng tải  
Băng tải có nhiệm vụ di chuyển theo chiều từ trái sang phải .  
Bộ phận cung cấp cam.

## II. Nguyên lý hoạt động :

Khi bật công tắc X24 thì băng tải hoạt động Y1 ở chế độ ON ( lúc này cánh tay robot chưa cung cấp thùng cacton bởi vì Y0 ở chế độ OFF ). Sau khi kích xung X20 , lúc này cánh tay Robot sẽ cung cấp thùng cacton cho băng tải . băng tải di chuyển mang theo thùng cacton , khi thùng cacton đến gần bộ phận cung cấp cam thì sensor X1 sẽ hoạt động để tác động Y1 chuyển từ trạng thái On sang Off . Tiếp theo bộ phận cung cấp cam sẽ hoạt động thông qua cảm biến

X2 . Khi thùng cacton đã có 5 quả cam ( do bộ đếm counter thiết lập ) thì bộ counter sẽ thông báo cho timer T0 hoạt động để cho băng tải mang thùng cacton chứa cam đến kho cất giữ .

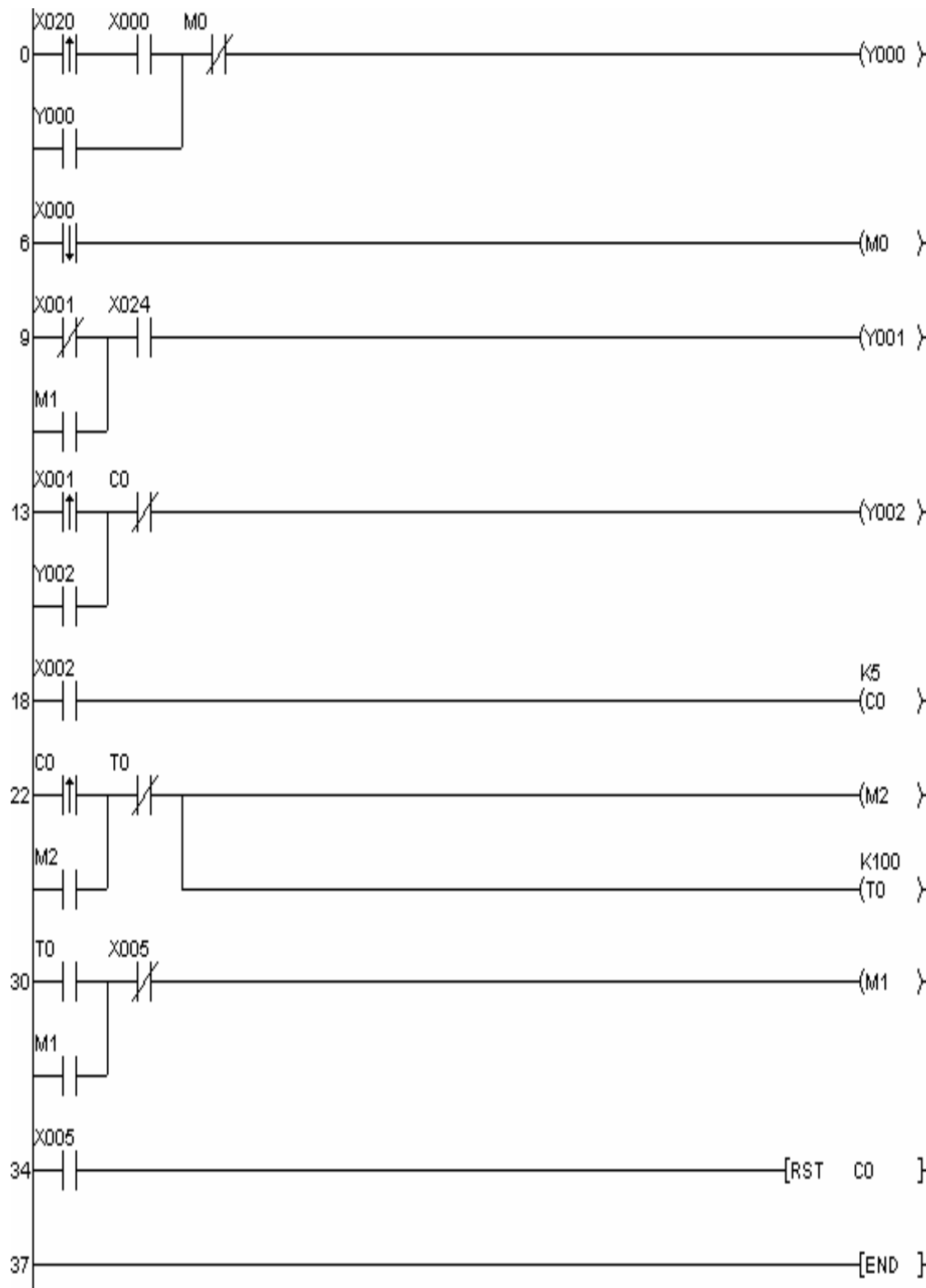
Khi thùng cam đi qua cuối băng ( Tại đây có lắp đặt sensor X5 ) thì lúc này cánh tay Robot sẽ được kích hoạt thông cảm biến X5 và relay phụ trợ M7 .

**Bảng thiết bị :**

Ngõ thiết bị	Tên thiết bị	Kí hiệu ngõ I/O	Chức năng
INPUT	Điểm bắt đầu của cánh tay robot	X000	ON khi robot ở tại vị trí bắt đầu
INPUT	Cảm biến nhận biết hộp cactông trên băng tải	X001	On khi thùng cactông ở phía dưới bộ phận cung cấp cam
INPUT	Cảm biến cung cấp cam	X002	On khi bộ counter đếm
INPUT	Cảm biến trạng thái	X005	On khi thùng cactông đi qua để trở về trạng thái khởi tạo ban đầu
OUTPUT	Bộ phận cung cấp thùng cactông thông qua cánh tay robot	Y000	Điều khiển bộ phận cung cấp thùng cactông cho băng tải
OUTPUT	Băng tải chuyển động theo chiều từ trái sang phải	Y001	Băng tải chuyển động khi Y1 ở chế độ ON
OUTPUT	Bộ phận cung cấp cam	Y002	Khi Y2 ở chế độ ON thì các quả cam được cung cấp cho thùng cactông

### III.Chương trình điều khiển :

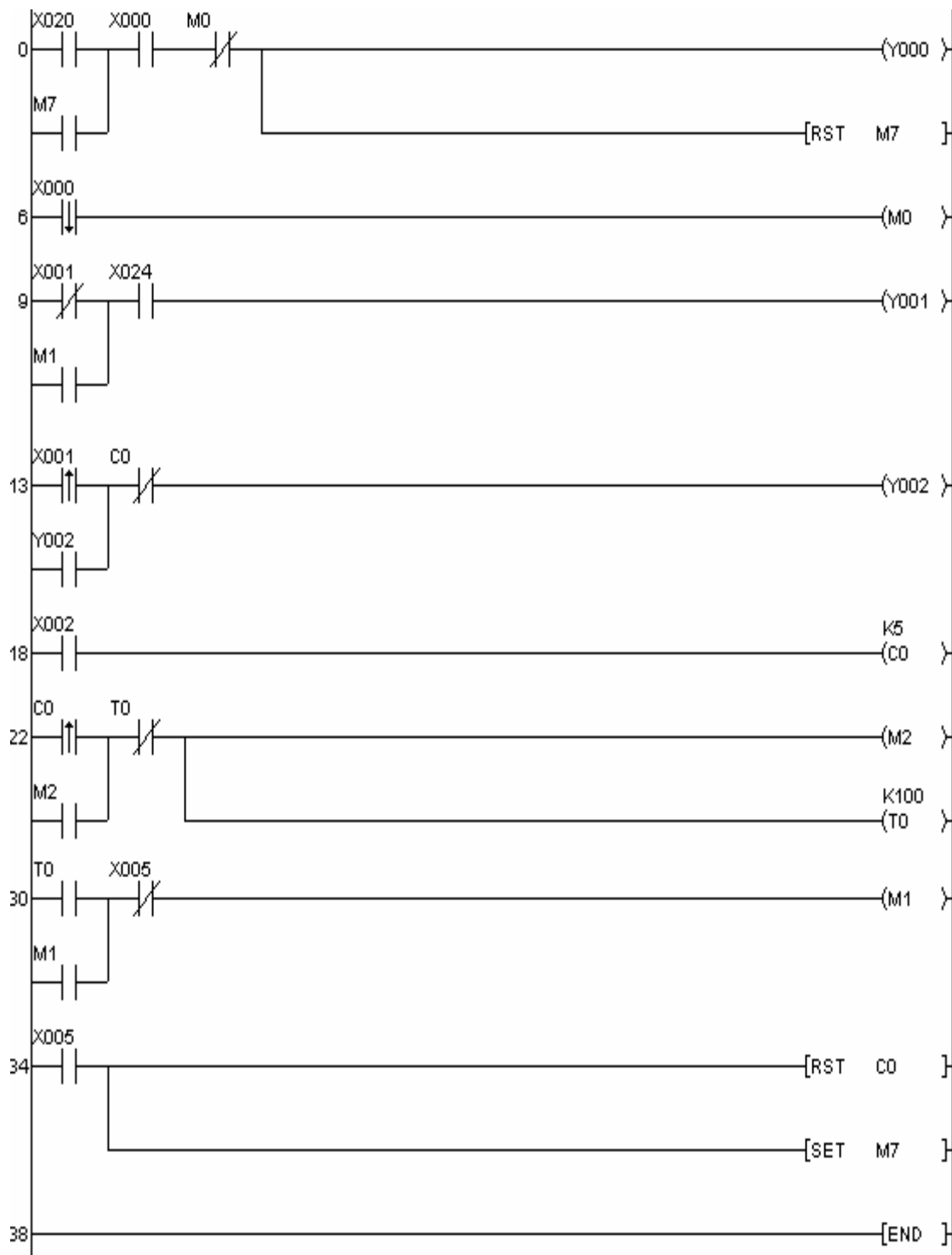
#### 1) Chương trình điều khiển cơ cấu hoạt động hở :



2) Chương trình điều khiển cơ cấu dạng xung :

a) Chương trình 1:





b) Chương trình 2:



Các chương trình điều khiển 1 và 2 có chế độ hoạt động luân phiên theo từ chu kỳ, thông qua việc sử dụng sensor X5 làm mốc khởi tạo trạng thái ban đầu. Việc sử dụng một chương trình tự động sẽ làm công việc trở nên đơn giản hơn và hiệu quả hơn trong công tác vận hành cơ cấu trong một dây chuyền sản xuất phức tạp.

#### IV. Các hàm lệnh được sử dụng trong chương trình điều khiển :

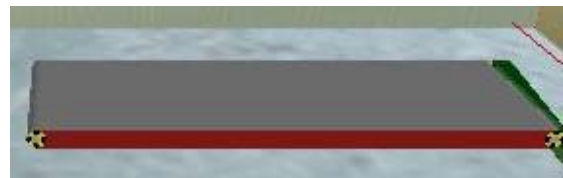
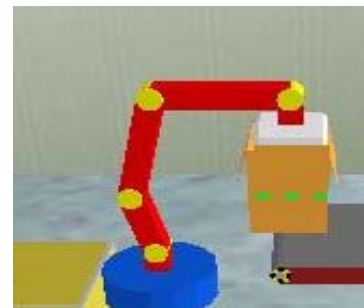
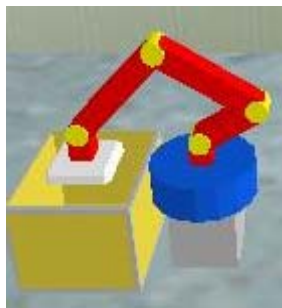
1. Hàm LD, LDP : dùng tạo trạng thái kích hoạt cho cơ cấu thông qua X20 .
2. Hàm AND ,ANI,: dùng làm các tiếp điểm trạng thái cho các sensor , băng tải , counter , cánh tay Robot, relay trạng thái M.
3. Hàm OR ,ORI : dùng làm các tiếp điểm trạng thái cho các sensor , băng tải , counter , cánh tay Robot, relay trạng thái M.
4. Hàm SET , RST : dùng làm trạng thái khởi tạo cho Counter và cơ cấu thông qua relay trạng thái M9.
5. Hàm PLS : dùng làm trạng thái khởi tạo chương trình thông qua Y006. (Chúng ta cũng có thể dùng kênh PLF để khởi tạo chương trình điều khiển ).
6. Hàm MRD : dùng tạo các nhánh cho mạch điện bậc thang .

#### Điều khiển cơ cấu :

- Sau khi nạp chương trình [Write to PLC ].
- Lúc này bộ plc sẽ chuyển từ trạng thái OFF sang RUN .
- Bật công tắc X24 , để cho băng tải hoạt động ( lúc này chỉ có duy nhất băng tải hoạt động trong cơ cấu ).
- Nhấn nút nhấn X20 để cho cơ cấu hoạt động . (Cánh tay Robot sẽ cung cấp thùng chứa cam ).

Các thiết bị :

- Cánh tay Robot :  
Nhiệm vụ chính của cánh tay Robot là cung cấp thùng cactong cho dây chuyền .Cánh tay robot hoạt động khi nó ở vị trí điểm bắt đầu như hình.
- Băng tải :  
Nhiệm vụ của băng tải là vận chuyển thùng cactong rộng đến bộ phận cung cấp cam và vận chuyển thùng chứa cam đến nơi lưu trữ .



- Bộ phận cung cấp cam :  
Nhiệm vụ chính của bộ phận này là cung cấp cam cho thùng cactong ở phía dưới thanh trượt cung cấp cam .



- Cảm biến nhận biết : ( cảm biến hồng ngoại )  
Sensor X5 có nhiệm vụ nhận biết thùng chứa cam đi qua , bằng cách nhận đường viền phía hông của thùng chứa cam .



Cảm biến nhận biết thùng cactong khi thùng đến gần bộ phận cung cấp cam là sensor X1.

Cảm biến điều khiển quá trình cung cấp cam là sensor X2.

### ➤ Ví dụ 2 : **MẠCH ĐÈN GIAO THÔNG HAI CHẾ ĐỘ**

Mạch đèn giao thông là yêu cầu thiết yếu cho phần thực hành điều khiển ứng dụng dùng bộ điều khiển lập trình PLC . Mạch đèn giao thông có thể được thiết lập bằng câu lệnh tuần tự hoặc ngẫu nhiên .

I.Nguyên lý hoạt động :

#### 1. bảng trạng thái hoạt động :

**Chế độ 1:**

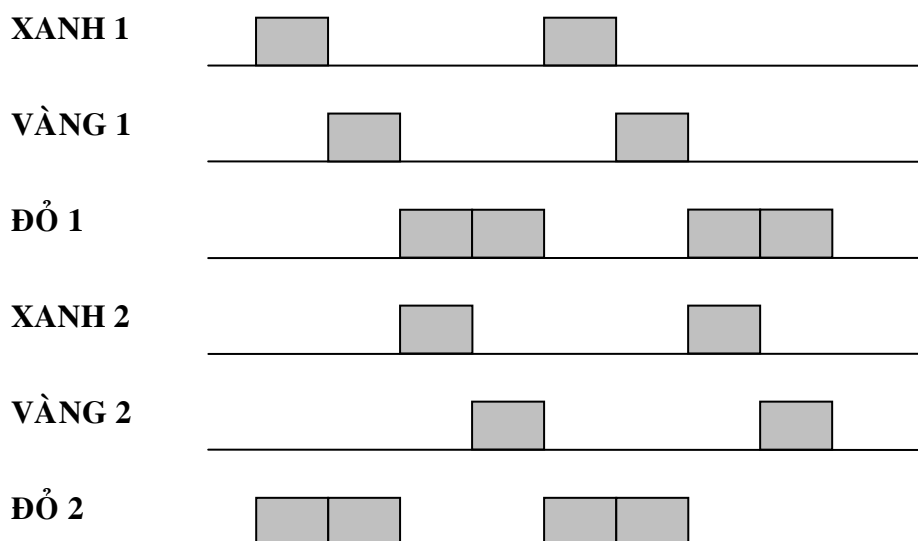
THỨ TỰ	XANH 1	VÀNG 1	ĐỎ 1	XANH 2	VÀNG 2	ĐỎ 2
1	1	0	0	0	0	1
2	0	1	0	0	0	1
3	0	0	1	1	0	0
4	0	0	1	0	1	0
5	1	0	0	0	0	1

**Chế độ 2:**

THỨ TỰ	XANH 1	VÀNG 1	ĐỎ 1	XANH 2	VÀNG 2	ĐỎ 2
1	0	1	0	0	1	0
2	0	0	0	0	0	0
3	0	1	0	0	1	0

2. Biểu đồ trạng thái :

**CHẾ ĐỘ 1:**

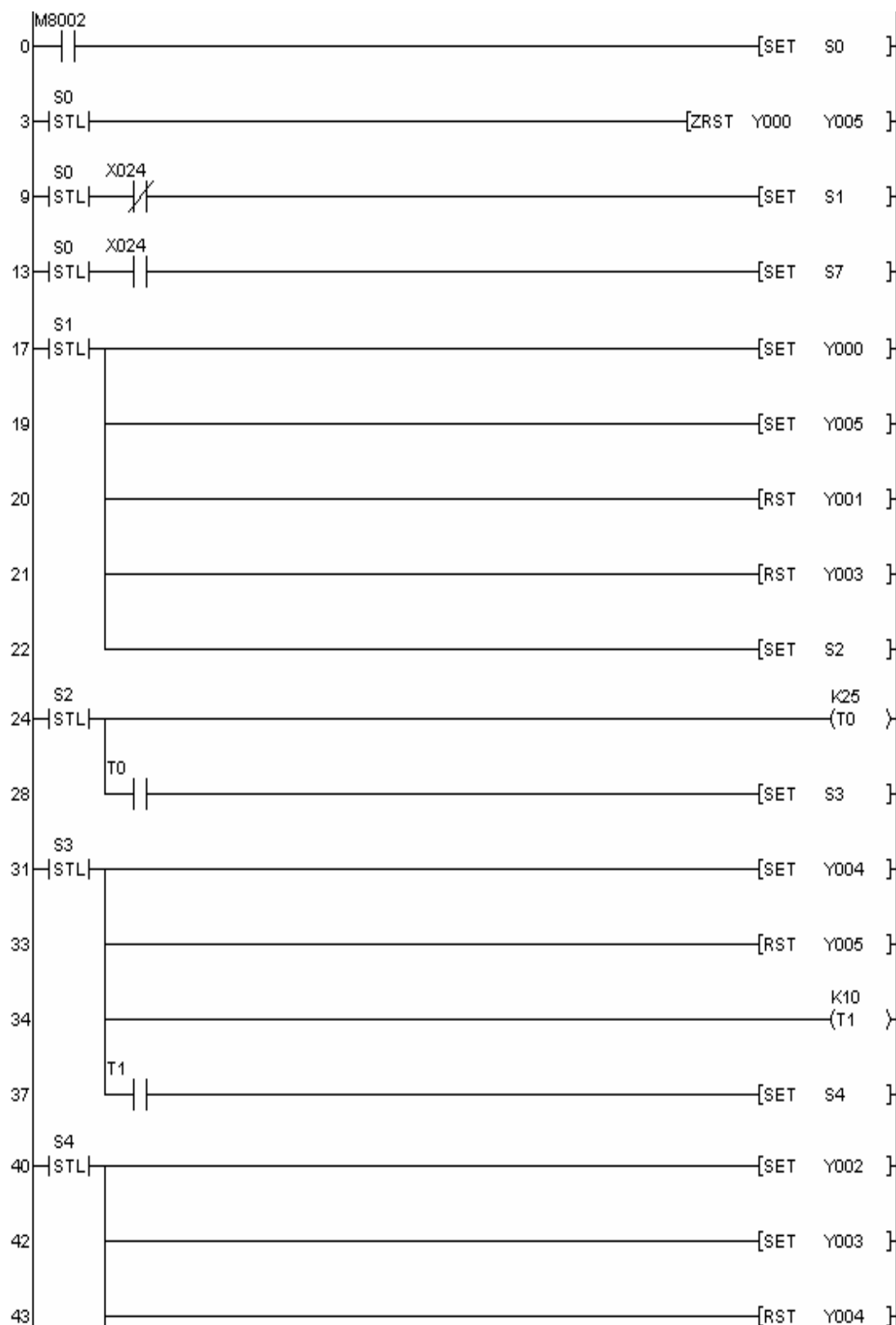


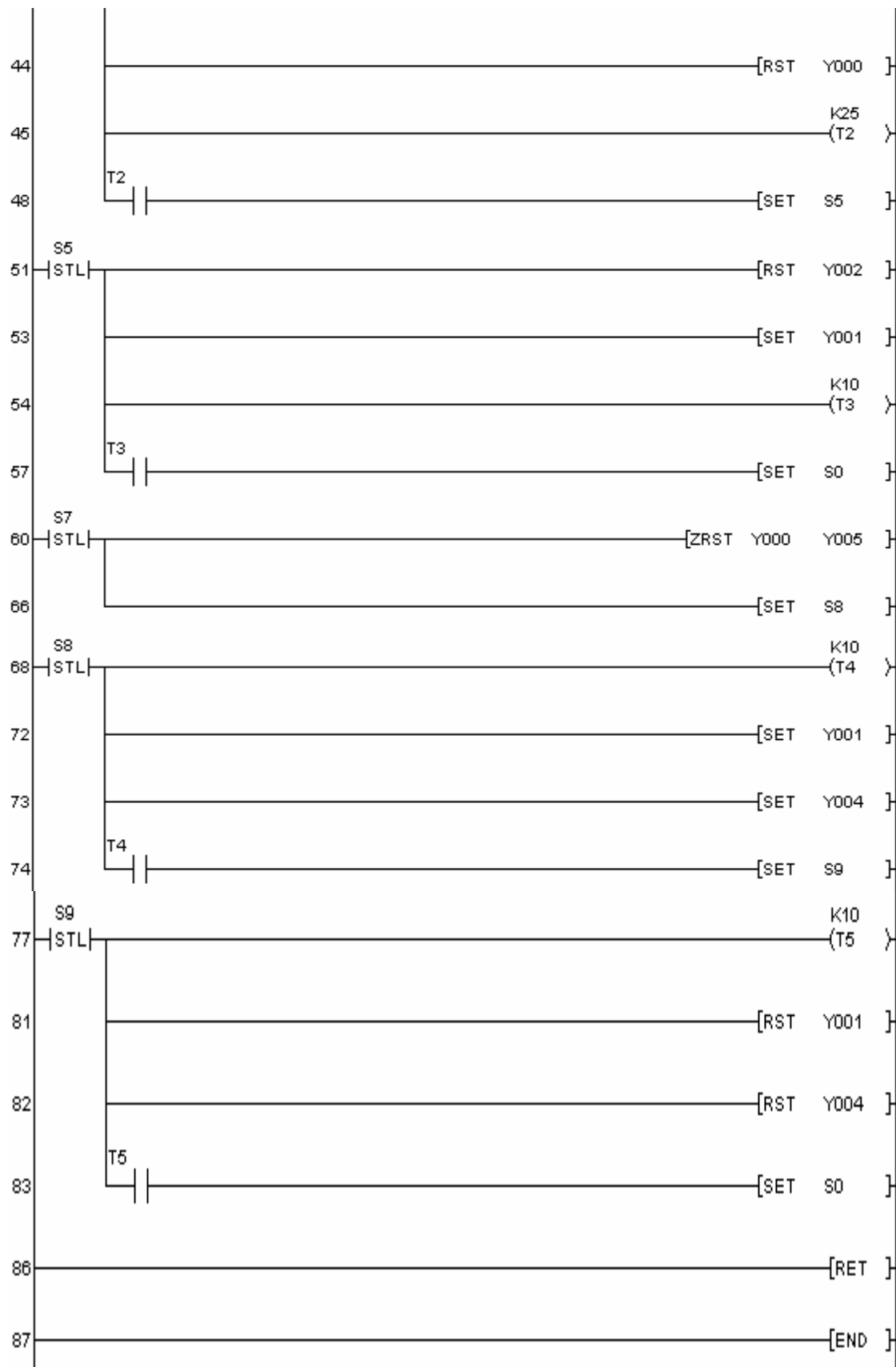
**CHẾ ĐỘ 2:**



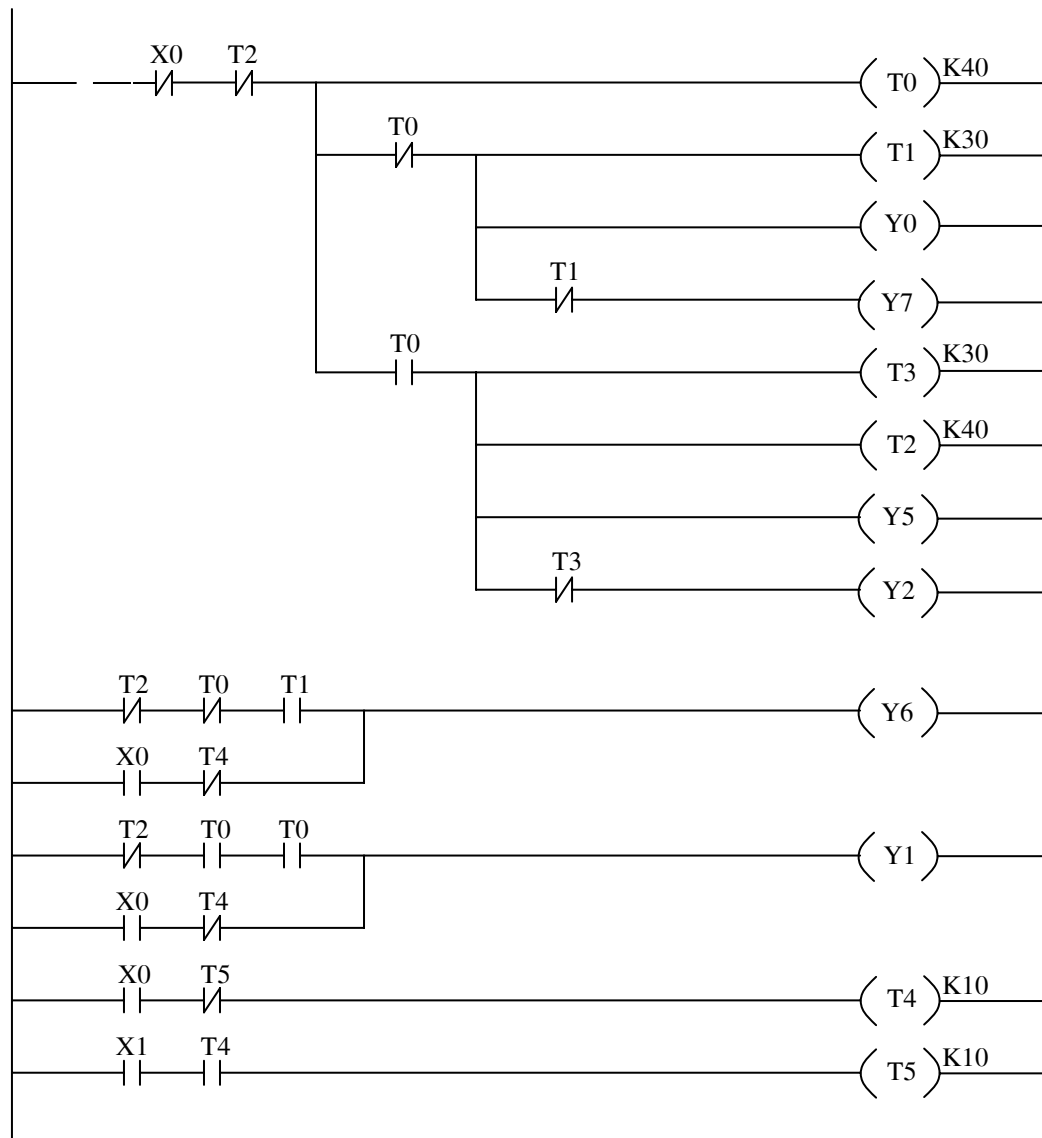
**II. Chương trình điều khiển :**

**1. Chương trình điều khiển tuần tự :**





2. Chương trình điều khiển ngẫu nhiên:





# LỰA CHỌN, LẮP ĐẶT, KIỂM TRA VÀ BẢO TRÌ HỆ THỐNG PLC

## I. Xem xét sự khả thi :

Trong trường hợp nào đó việc xem xét về sự khả thi là yêu cầu không thể thiếu trong một quyết định hay một giải pháp điều khiển. Phạm vi của lĩnh vực này có thể thay đổi rất nhiều, từ việc đơn giản là đặc tả tính khả thi của yêu cầu cho đến việc phân tích toàn diện mọi trường hợp. Việc nghiên cứu tính khả thi bao gồm những lĩnh vực sau:

- tính khả thi về kinh tế
- tính khả thi về kỹ thuật
- sự lựa chọn các phương án : nghiên cứu đánh giá các phương án để đưa ra phương án tối ưu.

## II. Trình tự thiết kế hệ thống PLC:

Vì PLC được thiết kế dạng modun nên gần như việc thiết kế và cài đặt phần cứng, phần mềm được thực hiện đồng thời nhưng độc lập với nhau. Việc thực hiện như vậy có ưu điểm là tiết kiệm thời gian và hệ thống linh hoạt bất chất chức năng của hệ thống.

### 1. Chọn PLC :

Hiện nay trên thị trường có rất nhiều loại PLC với các tính năng ngày càng được tăng cường nhằm cải thiện hiệu suất và chất lượng. Tuy nhiên, hầu hết PLC cùng cỡ thì có các chức năng điều khiển tương đương nhau. Điểm khác nhau quan trọng nhất là ở ngôn ngữ lập trình, cùng với các mức độ về sự hỗ trợ của nhà sản xuất. Sự hỗ trợ từ nhà sản xuất là yếu tố quan trọng khi thực hiện thiết kế một hệ thống điều khiển tự động.

Việc chọn PLC của hãng nào là do sự quen dùng đối với PLC đó và hệ thống điều khiển nói chung. Đối với những người có kinh nghiệm trong việc thiết kế và lắp đặt hệ thống điều khiển, vấn đề quan trọng là sự vượt trội về kỹ thuật và hiệu suất. Đối với những người chưa có kinh nghiệm nhiều về PLC và không nắm vững về thị trường PLC thì cần xem xét những vấn đề sau đây :

- người dùng có nhận được sự hỗ trợ trong công việc thiết kế?
- Tỷ lệ về thị trường và lĩnh vực ứng dụng của nhà sản xuất.
- Nhà sản xuất có tổ chức các khoá huấn luyện về hệ thống PLC đang sử dụng?
- Sổ tay và tài liệu hiện có với ngôn ngữ đọc được.
- Khả năng tương thích giữa các hệ thống tương đương hoặc loại PLC khác của cùng nhà sản xuất.
- Phương pháp lập trình có thích hợp với các điều khiển ứng dụng.

### 2. Loại và cỡ PLC:

Sự lựa chọn này có thể thực hiện cùng với việc lựa chọn nhà sản xuất PLC. Quy mô của hệ thống có một số điểm cần xem xét:

- Yêu cầu ngõ vào ra cần thiết

- Loại ngõ vào/ra.
- Dung lượng bộ nhớ.
- Tốc độ và khả năng của CPU và tập lệnh.

### 3. Số lượng ngõ vào ra:

Số lượng ngõ vào ra của PLC phải có khả năng đáp ứng đủ số đường tín hiệu từ cảm biến cũng như đường điều khiển phần công suất do cơ cấu tác động. Các tín hiệu không những tuân theo các chỉ tiêu kỹ thuật của hệ thống như là mức điện áp, dòng tải, tần số đáp ứng mà còn quan tâm đến các điểm sau:

- Số lượng và ngõ vào ra trên mỗi modul(hay trên PLC nếu các ngõ vào ra có sẵn trên PLC) .
- Sự cách ly giữa bộ điều khiển và phần công suất điều khiển cơ cấu tác động.
- Nhu cầu về ngõ vào ra xử lý tốc độ cao, điều khiển đầu ra từ xa, hay các chức năng chuyên dùng khác.
- Nhu cầu mở rộng thêm khả năng mở rộng và lắp đặt thêm ngõ vào ra.
- Nguồn cấp điện cho các ngõ vào ra, nghĩ là nhu cầu có modul PSU (Power Source Unit) cung cấp cho mạch chuyển đổi tín hiệu hay cơ cấu tác động

### 4. Dung lượng bộ nhớ:

Đối với PLC có khả năng mở rộng bộ nhớ thì dung lượng bộ nhớ có thể được mở rộng bằng cách gắn thêm hộp bộ nhớ (**memory cassette**). Dung lượng bộ nhớ phụ thuộc vào số lượng ngõ vào ra sử dụng trên hệ thống. Một chương trình điều khiển phức tạp, dùng nhiều logic khóa lẫn hay các chương trình trình tự hiển nhiên cần nhiều bộ nhớ hơn là chương trình đơn giản.

**BỘ NHỚ CẦN THIẾT = BỘ NHỚ NGÕ VÀO RA + BỘ NHỚ CHƯƠNG TRÌNH + BỘ NHỚ LƯU DỮ LIỆU CHO CÁC LỆNH CHUYÊN DÙNG + KHOẢNG DUNG LƯỢNG CẦN THIẾT CHO MỞ RỘNG VÀ HIỆU CHỈNH CHƯƠNG TRÌNH SAU NÀY.**

### 5. Tập lệnh CPU:

Mọi PLC đều phải có tập lệnh để phục vụ cho việc lập trình giải quyết các nhiệm vụ điều khiển. Tất cả các PLC đều có thể hiểu được lệnh điều khiển logic, điều khiển trình tự... sự khác nhau nổi bật là khả năng xử lý dữ liệu, những chức năng chuyên dùng và truyền thông. PLC loại lớn có tập lệnh mạnh hơn các PLC nhỏ hơn. Tuy nhiên ta nên xem xét kỹ khả năng của PLC loại nhỏ và trung bình thường có thể cũng có các chức năng chuyên dùng rất tốt như chức năng điều khiển PID.

Ở các PLC dạng modul có thể lựa chọn được modul CPU với các mức độ và tốc độ và khả năng xử lý. Khi số lượng và khả năng xử lý tăng, yêu cầu về tốc độ CPU cũng tăng, vì khi đó CPU phải xử lý nhiều lệnh hơn trong một chu kỳ quét của PLC. Điều đó có thể đòi hỏi phải dùng CPU mạnh hơn nếu thời gian quét không đáp ứng được nhu cầu.



Việc thiết kế nên dùng cáp có bọc che chắn để nối với các ngõ vào/ra của PLC. Khi nối cáp dài, các đường tín hiệu vào và ra nên tách riêng trên các dây cáp khác nhau. Các bọc che chắn nên được nối đất tại một điểm duy nhất để tránh nhiễu nối đất tín hiệu.

### 3. Lắp đặt:

Việc lắp đặt phần cứng bao gồm việc bố trí các tủ điện và các thanh gắn trên PLC, sau đó lắp đặt và nối dây.

Tủ gắn PLC và các khe cắm phụ phải được đáp ứng đầy đủ yêu cầu về môi trường, như vấn đề bảo mật, an toàn và bảo vệ như sau:

- Bảo mật: tủ đặt ở nơi chốt chốt và cửa có khoá.
- An toàn: trang bị mạch tự động ngắt hay báo động nếu tủ điện bị mở
- Bảo vệ: tránh độ ẩm hay môi trường ăn mòn do không khí bằng cách làm kín trên khung nắp. Việc ngăn tĩnh điện bằng cách nối đất cho thân khung lắp.
- Với mục đích bảo trì, các khe cắm được bố trí dễ dàng thực hiện kiểm tra, thay thế các modul. Các đèn chỉ thị on/off hay đèn chỉ thị trạng thái hoạt động nên gắn trên cửa tủ.

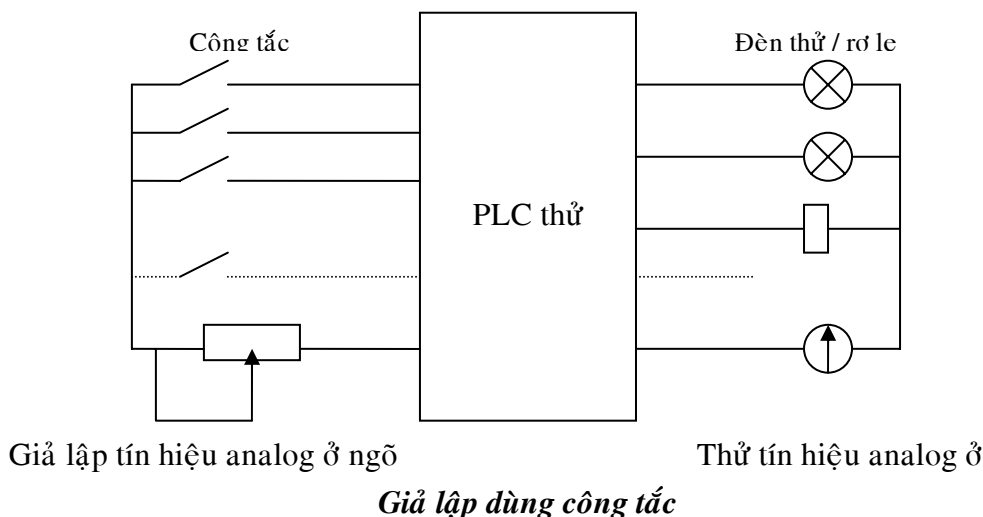
### IV. Chạy thử chương trình:

Khi hoàn tất việc lắp đặt hệ thống, bước kế tiếp là thử hệ thống. Việc chạy thử chương trình gồm hai giai đoạn cơ bản:

- Kiểm tra việc kết nối dây giữa PLC và các thiết bị được điều khiển.
- Cài chương trình điều khiển và chạy thử chương trình

Việc nối kết hệ thống phải được thực hiện cẩn thận để đảm bảo các ngõ vào/ra được nối đúng đến các ngõ vào/ra trên PLC. Trước khi phần cứng được kiểm tra, phải kiểm tra lại toàn bộ các nguồn cấp điện chính, việc nối đất.

*Chạy thử chương trình với ngõ vào giả lập:*



Lắp đặt và chạy chương trình trên hệ thống thật: khi chương trình đã chạy thử tốt, ta có thể chuyển sang bước thử trên thiết bị thực tế .

### ***Hiệu chỉnh chương trình:***

Một trong những ưu điểm lớn của PLC là dễ dàng thay đổi chương trình. Ở giai đoạn thiết kế và hoàn tất dự án, ưu điểm này cho phép thay đổi và thử đi thử lại nhiều lần chương trình đến khi thoả mãn thiết kế.

Lưu chương trình dự phòng: Khi hoàn tất giai đoạn thử và hiệu chỉnh chương trình, các bản sao dự phòng tương ứng với từng giai đoạn phải được lưu trữ kèm theo các mô tả về chúng. Ngay cả khi chương trình được viết xong nên lưu một bản dự phòng để nạp lại chương trình khi bộ nhớ trên PLC bị hư.

### ***V.Lập tài liệu cho hệ thống:***

Tập tài liệu đầy đủ cho việc lắp đặt PLC nên bao gồm các phần sau:

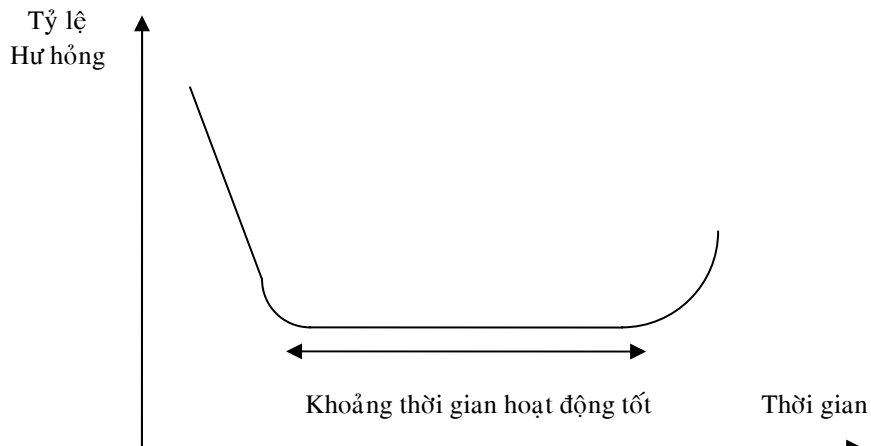
- Các đặc tả về kế hoạch hay tiến trình thực hiện.
- Các đặc tả và đặc điểm về yêu cầu điều khiển
- Các đặc tả về PLC gồm cả sổ tay về thiết bị cần điều khiển.
- Các đặc tả về các chức năng của chương trình điều khiển, gồm lưu đồ hay sơ đồ chức năng, bộ nhớ và sự cấp phát bộ nhớ.
- Toàn bộ chương trình cùng với các chú thích và giải thích.
- Tập chương trình dự phòng.
- Sơ đồ lắp đặt điện và mô tả
- Danh sách các thiết bị xuất nhập dữ liệu gồm cả màn hình hiển thị, máy in...
- Sổ tay sử dụng ghi chi tiết trình tự khởi động hệ thống, dừng hệ thống và các báo động của hệ thống.

### ***VI.Bảo trì hệ thống PLC:***

#### ***1. Lỗi do PLC:***

PLC được thiết kế để hoạt động tin cậy và bền vững trong môi trường công nghiệp. Đây là ưu điểm chính vượt trội so với điều khiển cơ điện tử truyền thống và điều khiển từ máy vi tính. Do cấu trúc mạch điện tử mạch và cảm ứng điện hiệu quả nên các mạch điện trong PLC sử dụng an toàn ở các nơi không thuận lợi về môi trường, dễ gây ra các hư hỏng cho phần cứng về điện và về vật lý.

PLC được bảo vệ tránh các khả năng hư hỏng trên các ngõ vào/ra bởi các mạch cách ly quang (**opto-isolated**). Việc dùng bộ nhớ RAM có nguồn pin nuôi hay EEPROM lưu giữ được chương trình bảo đảm được sản xuất vẫn được duy trì nếu chương trình bị mất hay sai do nguồn cấp điện bị hỏng hay trong các trường hợp tương tự. Tóm lại, các biện pháp khả thi trong thiết kế được áp dụng cho PLC nhằm đạt được độ tin cậy cao có thể được với giá thành hợp lý.



**Đồ thị biểu diễn tỉ lệ phần trăm hư hỏng theo thời gian**

Tuy nhiên, độ tin cậy cao không có nghĩa là hệ thống hoàn toàn không bị hỏng. Dù sử dụng các vi mạch điện tử có chất lượng cao và được lắp ráp theo tiêu chuẩn cao thì **PLC** vẫn có lỗi. **PLC** thường được thử rất kỹ trước khi xuất xưởng và chúng được cho chạy liên tục chương trình thử trong thời gian dài.

**Lỗi do PLC:** các lỗi nghiêm trọng làm cho **PLC** bị ngưng hoạt động, một số lỗi khác có thể cho phép **PLC** tiếp tục hoạt động, hiển thị mã lỗi trên các màn hình thông báo hay trên các đèn bẫy đoạn. Khi **Self-test** không thành công, **PLC** bị ngưng hoạt động và không khởi động được cho đến khi lỗi được khắc phục.

### **2. Lỗi từ phần cứng bên ngoài PLC:**

**PLC** là một phần trong phần điều khiển; trong hệ thống có cảm biến, cơ cấu tác động, dây kết nối, nguồn cấp điện và chương trình điều khiển. Mỗi phần đều có khả năng hư hỏng; tuy nhiên phần lớn tỷ lệ hư hỏng thuộc về các bộ phận, thiết bị nằm bên ngoài **PLC** như:

- Hư hỏng thiết bị vào/ra – mạch chuyển đổi tín hiệu hay cơ cấu tác động.
- Hư hỏng phần dây kết nối.
- Hư hỏng phần dây kết nối truyền thông.
- Nguồn cấp điện không ổn định – nhiễu hai mức nguồn.

### **3. Trình tự lập trình báo lỗi:**

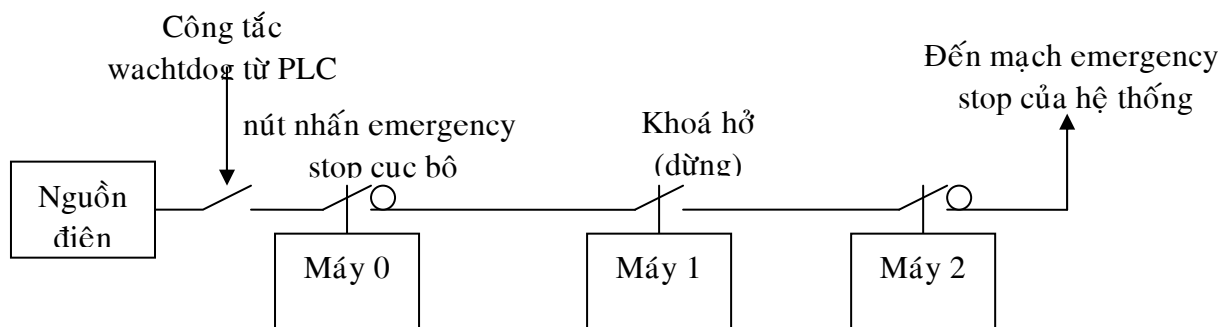
**Vấn đề bảo vệ:** Một dạng lỗi trong **PLC** về xử lý thương liên quan đến mạch bảo vệ (**watchdog**). Hầu hết **PLC** có trang bị rơle bảo vệ (**watchdog relay**) bên trong dùng để điều khiển nguồn cấp điện cho một hay nhiều thiết bị bên ngoài.

**Vấn đề an toàn:** Mặc dù chương trình có hoàn hảo và tinh vi đến đâu thì cũng có khi xử lý sai, chương trình sẽ làm việc không đúng ngay cả khi **PLC** đang hoạt động bình thường. Vì không có **PLC** nào có độ tin cậy là 100% nên là chỉ có cách là thiết kế một hệ thống an toàn thông qua mạch phần cứng hơn là thông qua phần mềm.

**4. Mạch an toàn bằng phần cứng:**

Các mạch này phải được thiết kế độc lập với PLC và nên phối hợp với các công tắc bảo vệ (**watchdog relay**) của PLC để đảm bảo mạch này thực hiện việc ngắt hệ thống nếu PLC bị lỗi.

Thường nút nhấn Start và Stop được thiết kế nối vào ngõ vào của PLC, nghĩa là Start và Stop được thực hiện bởi chương trình, trong khi đó công tắc và nút nhấn **EMERGENCY STOP** (dừng khẩn cấp) dùng để dừng sự chuyển động của máy và các thiết bị có nhiều khả năng gây nguy hiểm phải được thiết kế ở mạch phần cứng.



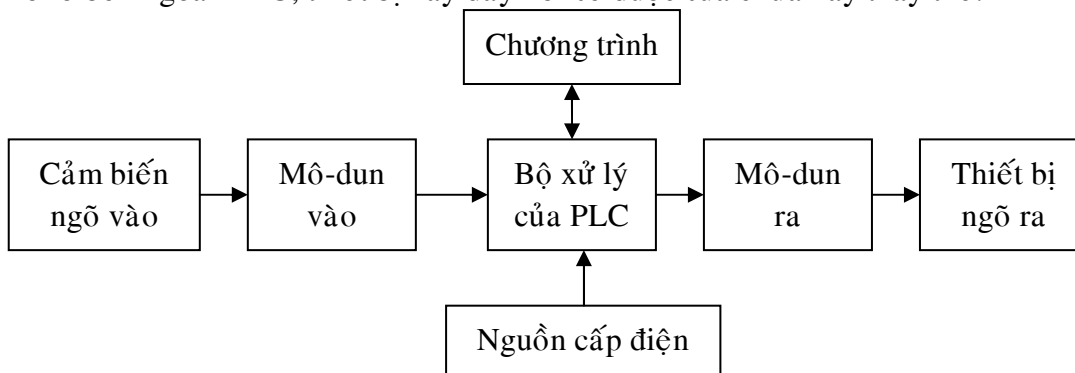
**Mạch phần cứng EMERGENCY STOP thông dụng**

**5. Sửa lỗi :**

Khi hệ thống điều khiển chưa làm việc hay đang chạy đúng, nguyên nhân các hư hỏng phải được xác định, phải được khắc phục theo trình tự:

- Ghi chú và nghiên cứu các triệu chứng.
- Xác định các vùng có nghi vấn.
- Cô lập lỗi.
- Sửa lỗi.

**Sửa lỗi :** nếu lỗi đã được xác định thì cần phải thay thế một modul nào đó; khi đó, tùy thuộc vào chức năng của modul đó mà có thể phải cấu hình lại cho hệ thống. Nếu lỗi xảy ra trong chương trình thì người thiết kế hệ thống nên xem xét kỹ trước khi thực hiện thay đổi nào đó của chương trình, vì còn có các hoạt động khác liên quan đến các dòng lệnh sẽ bị thay đổi. Khi lỗi ở bên ngoài PLC, thiết bị hay dây nối sẽ được sửa chữa hay thay thế.



**Các thành phần của hệ thống PLC**

Chương VI

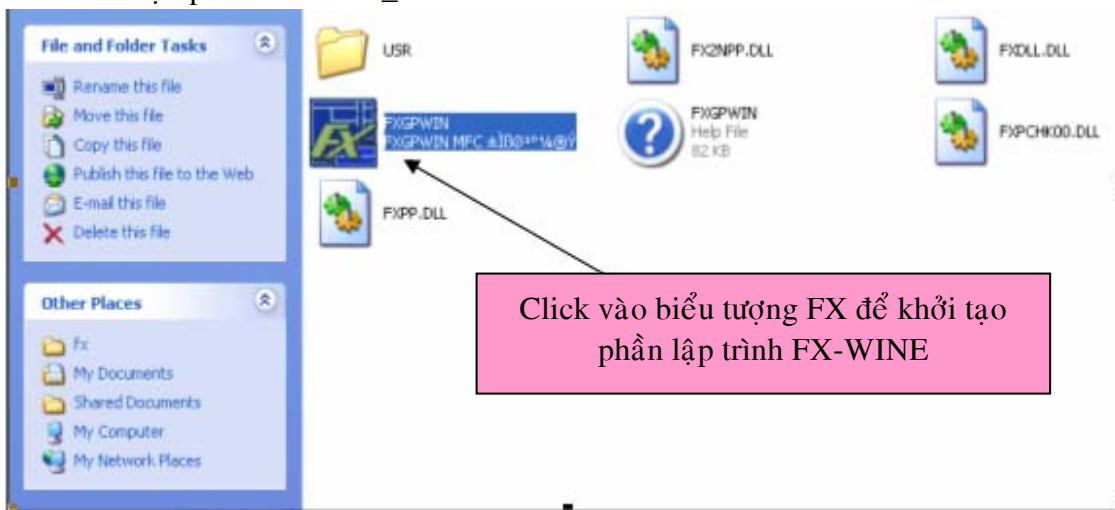
# SỬ DỤNG PHẦN MỀM FXGP-WINE

## I-Giới thiệu:

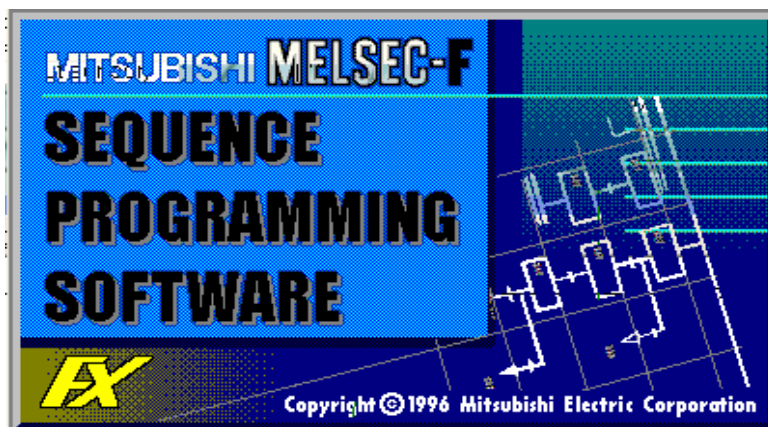
Phần mềm lập trình cho PLC của MITSUBISHI ELECTRIC là phần mềm **FXGP-WINE**. Phần mềm này cho phép lập trình trên các cơ cấu sản xuất thực tế. Đây là phần mềm ứng dụng dựa trên **WINDOWS**. Ngoài ra hãng MITSUBISHI còn có phần mềm mô phỏng dựa trên các cơ cấu sản xuất thực tế, nhằm giúp chúng ta có một cái nhìn khái quát hơn.

- Dung lượng của **FX\_WINE** 1,38 MB.
- Cài đặt trên tất cả các hệ điều hành WINDOWS
- Bộ nhớ 8MB hoặc lớn hơn.
- Display 640x480 hoặc cao hơn.
- Cổng kết nối từ PLC đến máy tính, cáp truyền RS-232C nên máy tính cần ít nhất một cổng COM1 hoặc nhiều hơn.

Khởi tạo phần mềm **FX\_WINE**:



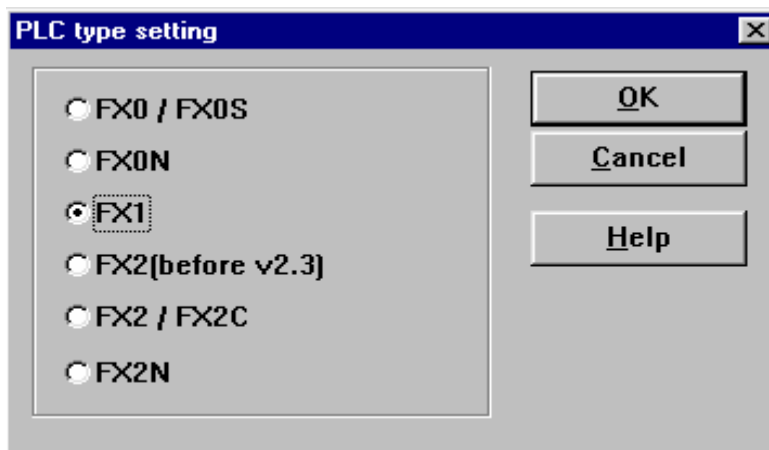
Việc khởi tạo phần mềm **FXGP\_WINE** tùy thuộc việc cài đặt phần mềm này tại đâu.





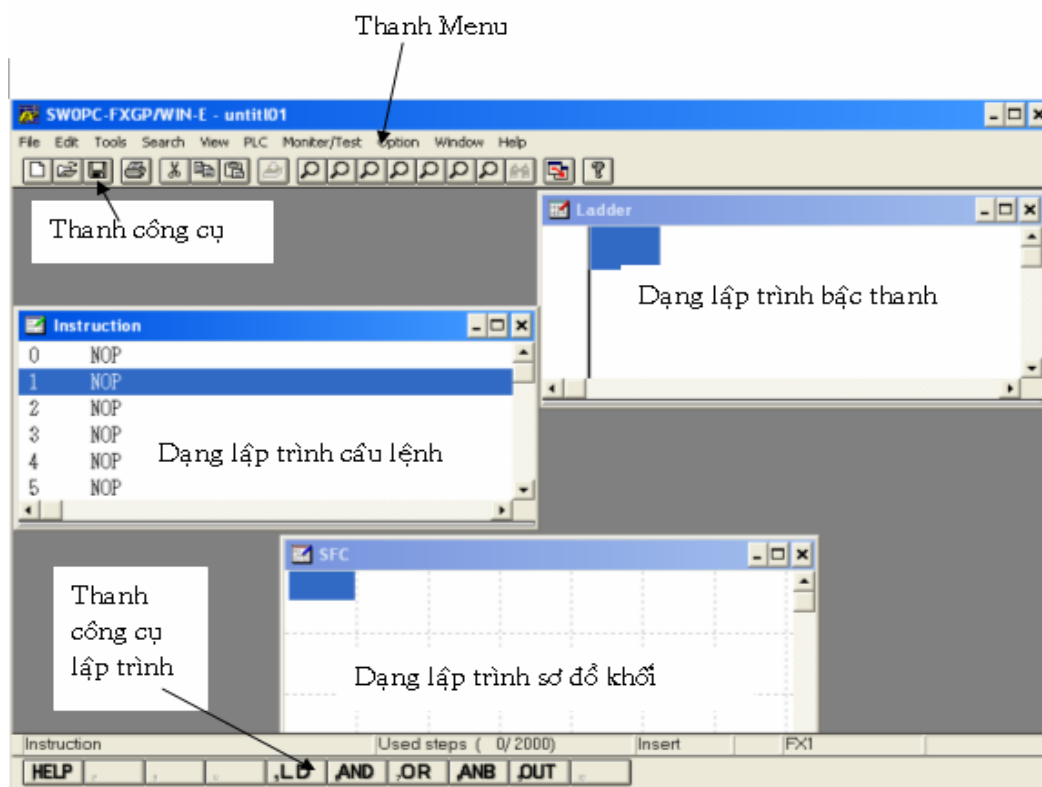
## II. Phần mềm FXGPWINE:

Sau khi khởi tạo chương trình lập trình **FXGPWINE** trên màn hình sẽ xuất hiện cửa sổ **PLC type setting**



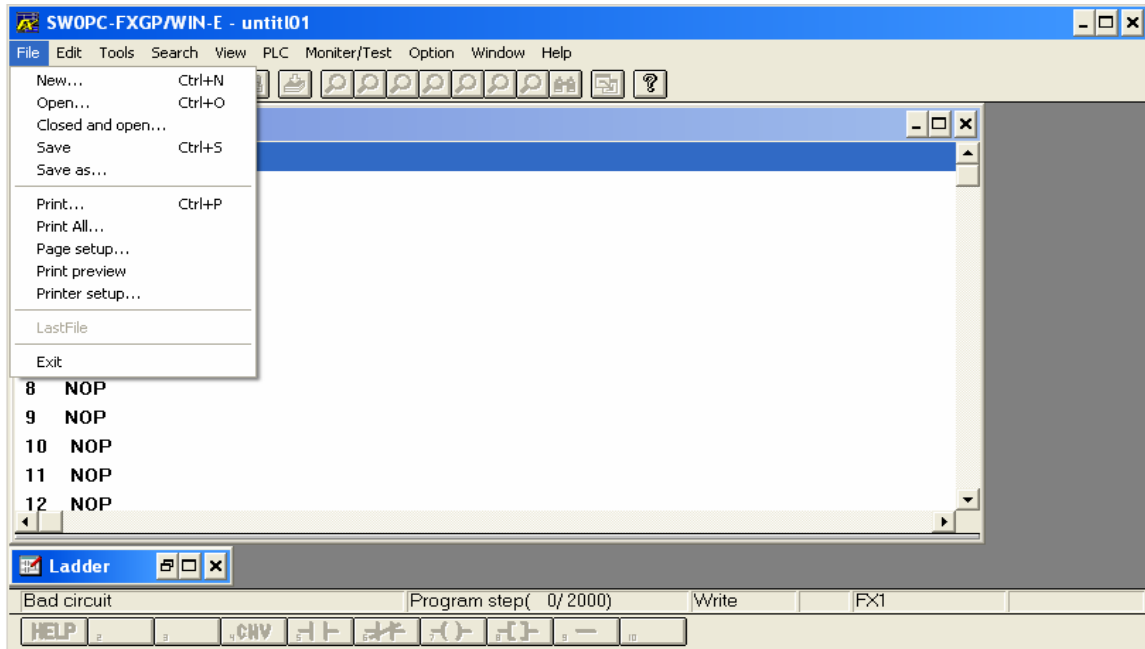
Chọn họ **FX** cần dùng cho việc lập trình , sau đó click OK.

### 1. Giao diện FXGPWINE:



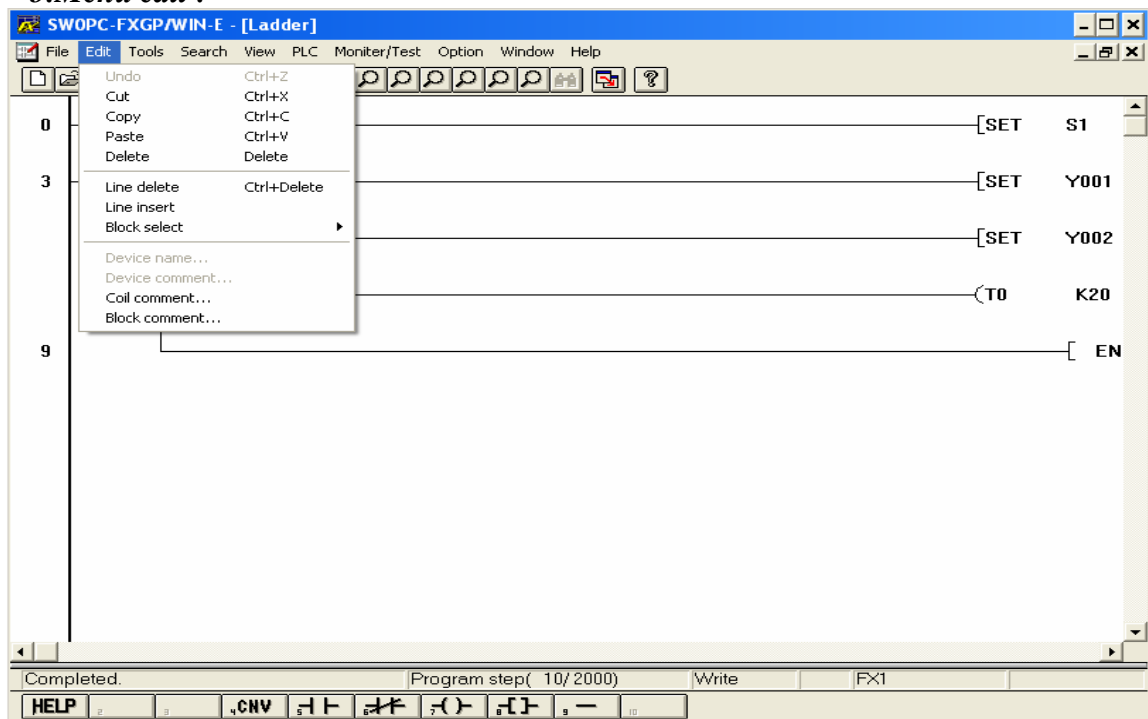
## 2. Giới thiệu về thanh Menu:

### a. Menu File:



- \*New..: Tạo một chương trình mới.
- \*Open..: Mở một chương trình mới.
- \*Save : Lưu một chương trình .
- \*Save as :Lưu chương trình với một tên khác.
- \*Exit : Thoát khỏi chương trình FXGPWINE.

### b. Menu edit :

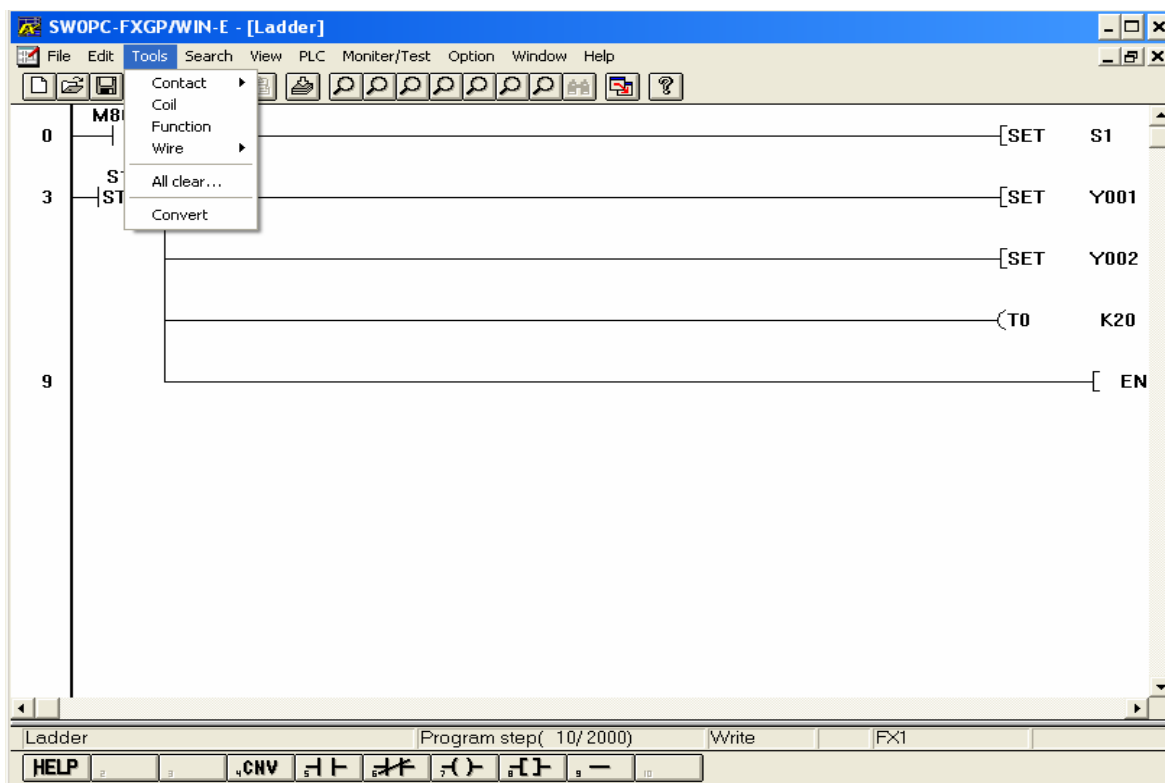


- \* Cut : Xóa tại vị trí con trỏ.
- \* Line delete :Xóa một dòng tại vị trí con trỏ.
- \* Line Insert :Chèn một dòng tại vị trí con trỏ.  
Trong thanh Menu→Edit→Line insert
- \* Block select : Xác định vị trí của khối tại up và down.
- \* Coil comment:Vị trí cuộn dây.
- \* Block comment:Vị trí khối.

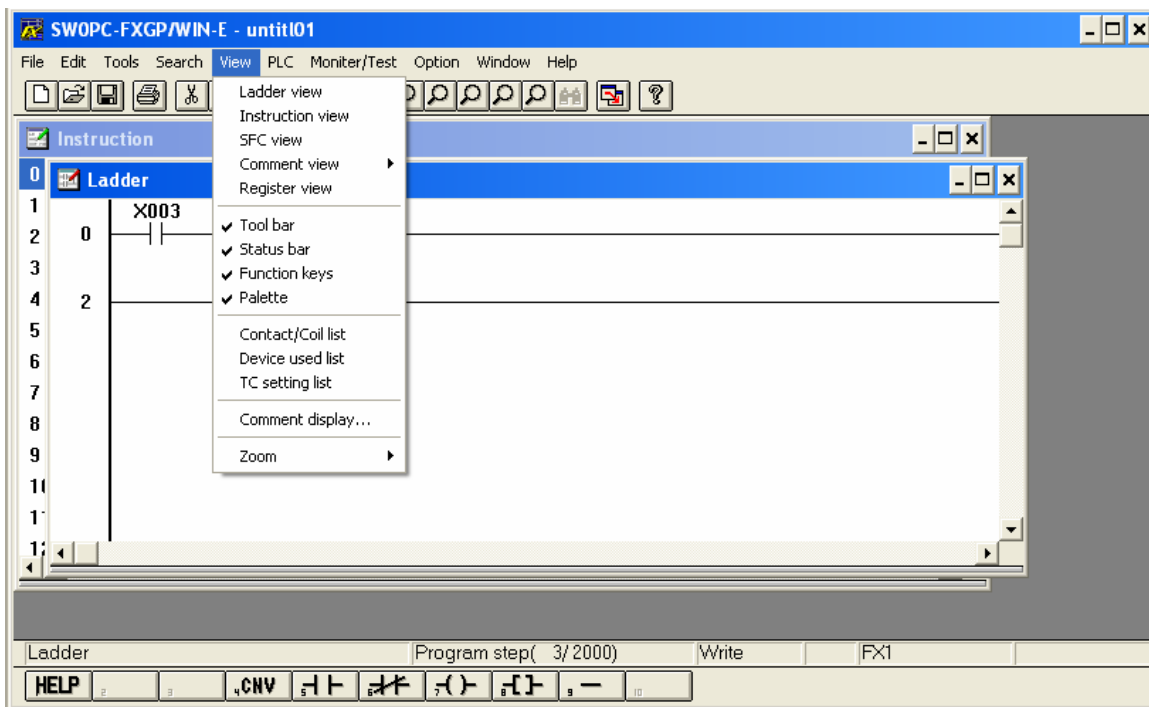
Undo	Ctrl+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Delete
Line delete	Ctrl+Delete
Line insert	
Block select	▶
Device name...	
Device comment...	
Coil comment...	
Block comment...	

### c.Menu Tools:

- \*Contact :Khai báo các tiếp điểm thường đóng và thường mở.
- \*Coil: Khai báo cuộn dây.
- \*Function: Khai báo về hàm truyền.
- \*Wire: Khai báo các line trong mạch.
- \*All clear:Xóa tất cả line.
- \*Convert: Xác nhận các line.



*d.MenuView:*



Ladder view :Chuyển từ ngôn ngữ Instruction sang Ladder.

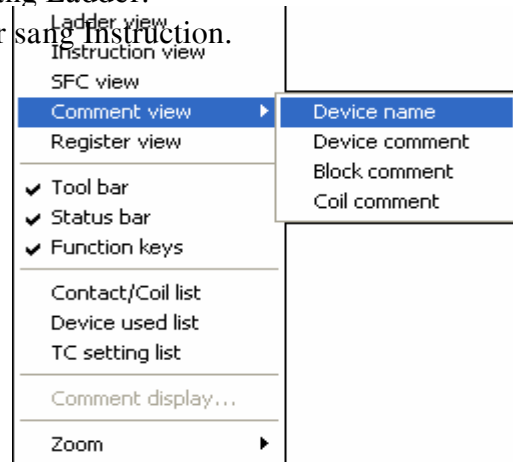
- Instruction view :Chuyển từ ngôn ngữ ladder sang Instruction.
- SFC view : Chuyển sang ngôn ngữ SFC.
- Comment view: khai báo về các thiết bị lập trình.

Khai báo tên các thiết bị I/O:

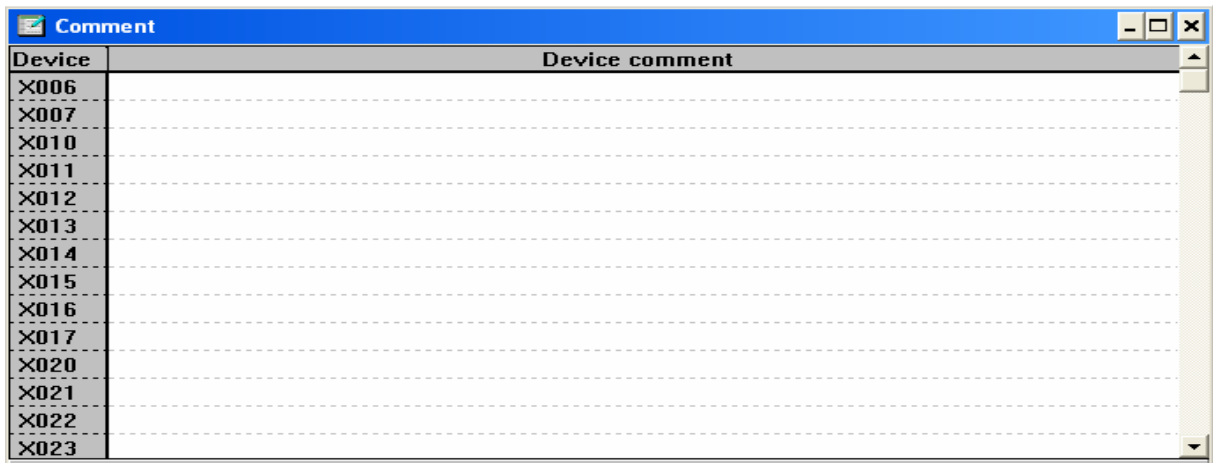
Tại thanh Menu →View→Comment View

- Device name: tên thiết bị.
- Device comment: Chú thích về thiết bị X, Y, T, C, M, S.
- Block comment: Chú thích về khối Ladder trong Step#.
- Coil comment : Chú thích về cuộn dây trong Step#.

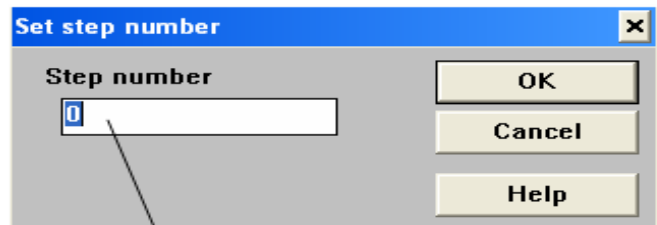
Tại cửa sổ Comment sau khi nhấp double vào Device comment, tùy theo việc chọn thiết bị .Sau đó khai báo các thiết bị tại cửa sổ Comment.



Gõ vào tên các thiết bị X,Y,T,C,M,S chọn OK.



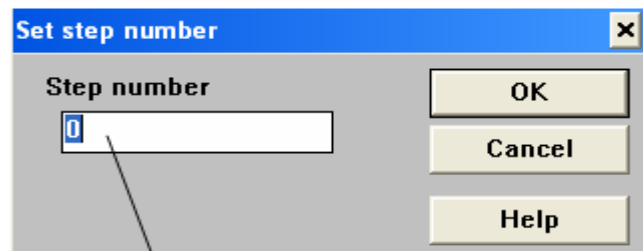
Nhấp double vào Block comment sẽ xuất hiện cửa sổ Set Step number. Sau đó xuất hiện cửa sổ Block comment. Tại đây khai báo các khối (Block).



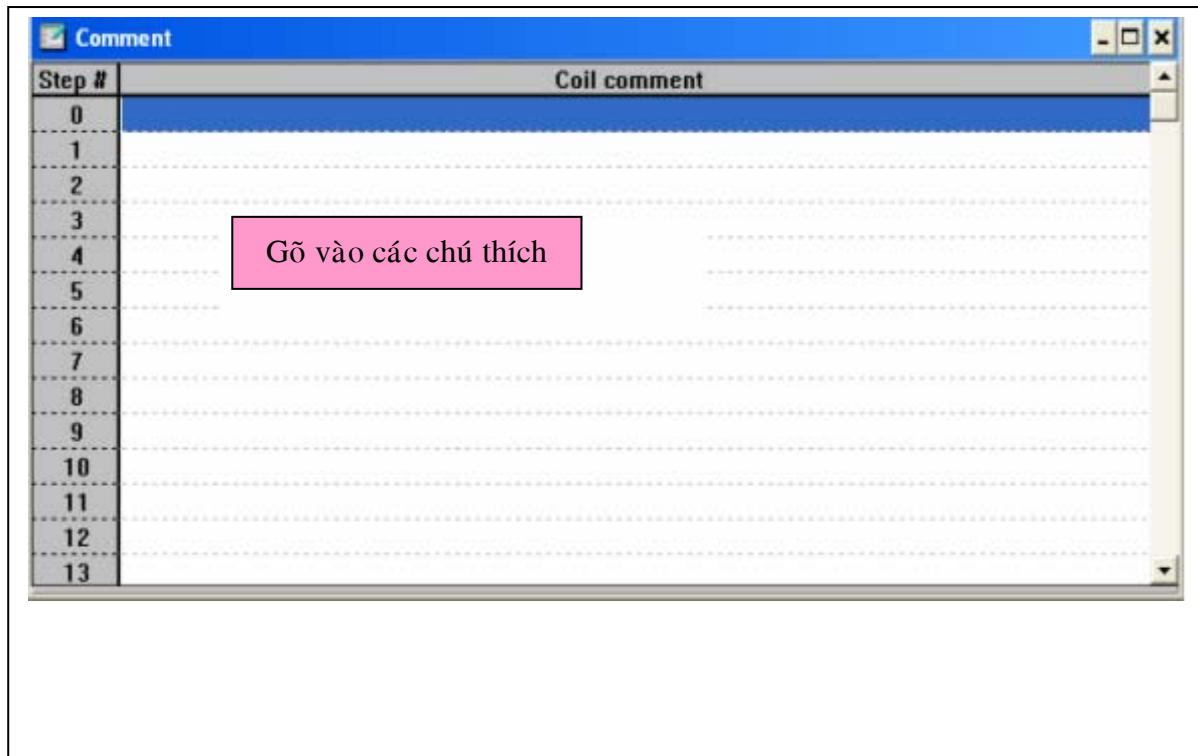
Gõ vào số bước cần chú thích. Chọn OK



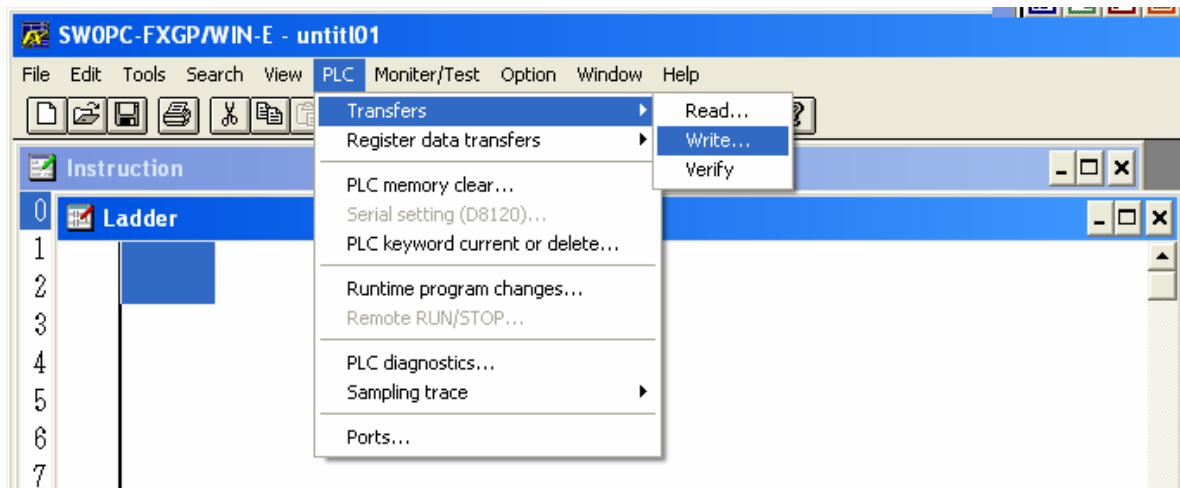
Nhấp double vào Coil comment xuất hiện cửa sổ Set Step number. Sau đó xuất hiện cửa sổ Coil comment. Tại đây khai báo các cuộn dây (Coil).



Gõ vào số bước cần chú thích. Chọn OK

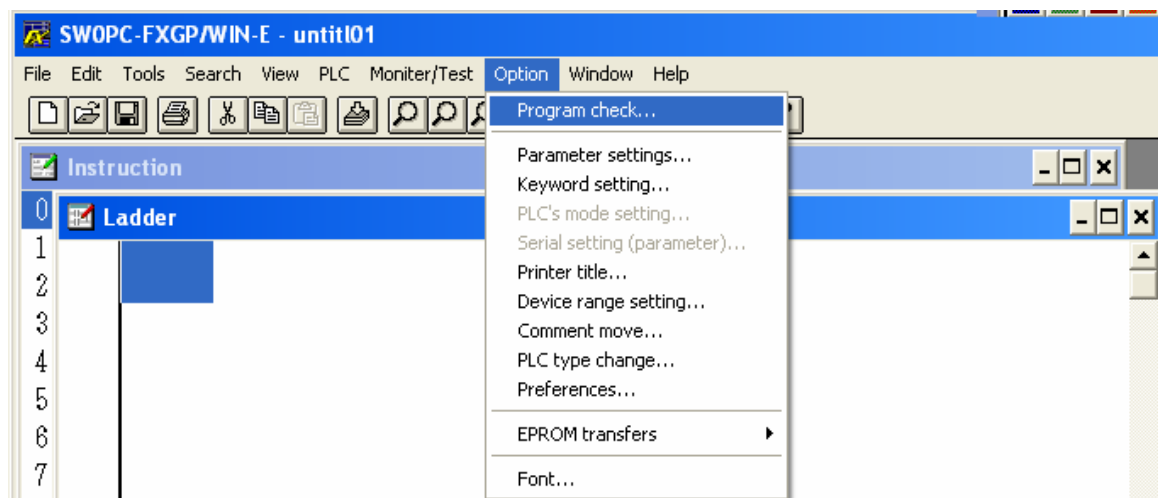


*e.Menu PLC :*

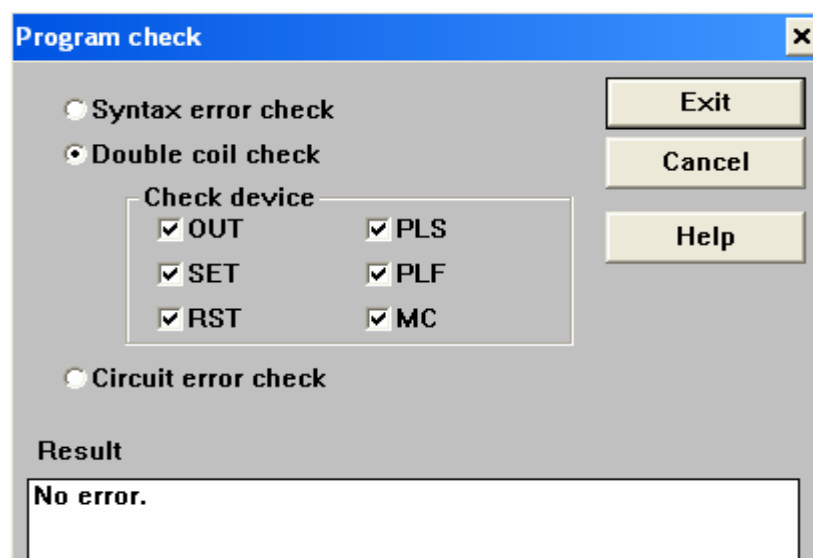


- Transfers : nạp chương trình PLC từ máy tính đến bộ điều khiển lập trình
  - Read : nạp chương trình từ bộ PLC về máy tính .
  - Write : nạp chương trình từ máy tính đến bộ PLC.
  - Verify : có chức năng kiểm tra lại quá trình nạp chương trình.
- Register data transfers : kiểm tra dữ liệu thanh ghi .
- PLC memory clear : có chức năng xóa bộ nhớ của PLC.

*f. Menu Option :*



Chức năng quan trọng nhất của thanh công cụ này là kiểm tra lỗi của chương trình .  
Chọn Option →Program check ... Sau đó sẽ xuất hiện cửa sổ Program check



- Syntax error check: kiểm tra lỗi về cú pháp trong ngôn ngữ lập trình **PLC**.
- Double coil check, kiểm tra lỗi về thiết bị ngõ ra của **PLC** , bao gồm các lệnh: OUT, SET, RST, PLS, PLF, MC.
- Circuit error check: kiểm tra về lỗi mạch điện .
- Chọn Exit để kiểm tra lỗi của chương trình .

***Chú ý:***

- No error : Không có lỗi, và có thể nạp chương trình từ máy tính vào **PLC**.

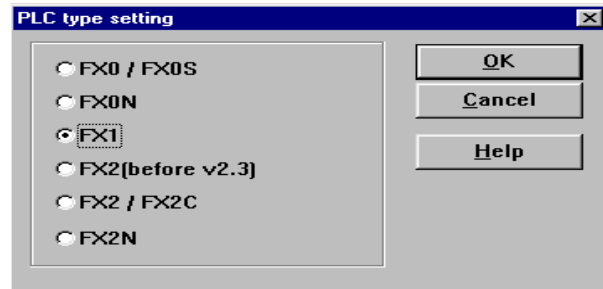
### III. Các bước lập một chương trình mới:

#### 1. Mở một tập tin mới:

Nhấn Ctrl+N, hoặc vào Menu FILE→NEW , hay có thể kích vào biểu tượng .

#### 2. Chọn kiểu PLC:

Sau khi tạo tập tin mới sẽ xuất hiện cửa sổ **PLC type setting** setting→FX1→OK.



#### 3. Mở một tập tin:

Kích vào biểu tượng , hay vào Menu FILE→OPEN, hoặc nhấn CTRL+O

#### 4. Lưu một tập tin:

Kích vào biểu tượng , hay vào Menu FILE→SAVE, hoặc nhấn CTRL+S .

### IV. Phương pháp lập trình:

#### 1. Ngôn ngữ lập trình LADDER:

Ngôn ngữ **ladder** thường được sử dụng để thiết kế hệ thống điều khiển mới .Khi viết chương trình ladder để điều khiển thiết bị có hoạt động đơn giản thì ít có đòi hỏi về hoạch định và thiết kế chương trình .Người viết chương trình có thể hiểu rõ các chức năng cũng như hoạt động của nó , nhưng chỉ trong một thời gian nhất định thì họ sẽ quên đi. Do đó cần phải có kế hoạch để thiết kế một chương trình điều khiển lập trình dạng **Ladder**.

Ngôn ngữ lập trình **ladder** được sử dụng rộng rãi, đây là ngôn ngữ lập trình dạng bậc thang. Ngôn ngữ này có dạng như một sơ đồ mạch điện logic và dùng các ký hiệu điện để biểu diễn các công tắc logic ngõ vào và các relay logic ngõ ra. Phần mềm lập trình có nhiệm vụ biên dịch các ký hiệu logic trên thành mã máy và được lưu vào bộ nhớ của PLC. Sau đó, PLC sẽ thực hiện các nhiệm vụ điều khiển theo logic thể hiện trong chương trình.

Về phương pháp lập trình ngôn ngữ **ladder** gồm các bước sau:

- ❖ Xác định yêu cầu và phân tích yêu cầu .
- ❖ Thiết kế chương trình.
- ❖ Lập tài liệu thiết kế chương trình.
- ❖ Chạy thử chương trình bằng các mô hình giả định.

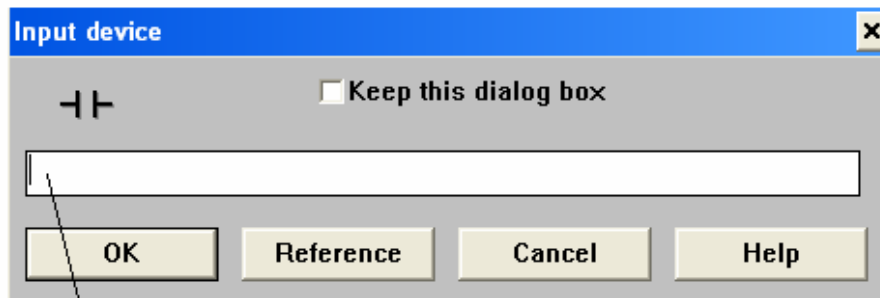
Đây là thanh công cụ sử dụng lập trình cho Ladder Bao gồm các tiếp điểm , các thiết bị và các hàm truyền đạt dùng để lập trình một chương trình PLC.






**a. Tiếp điểm thường mở:**

Kích biểu tượng  hoặc nhấn phím F5. Xuất hiện cửa sổ **Input device**.




Gõ vào tên thiết bị ngõ vào (Input device). Chọn OK hay


**b. Tiếp điểm thường mở song song :**

Kích vào biểu tượng  hoặc nhấn phím **Shift + F5** .Trong cửa sổ Input device, gõ vào tên thiết bị ngõ vào ( Input device). Chọn **OK** hay **Enter** (↵).


**c. Tiếp điểm thường đóng:**


Kích vào biểu tượng  hoặc nhấn phím F6 .Trong cửa sổ Input device , gõ vào tên thiết bị ngõ vào ( Input device). Chọn **OK** hay **Enter** (↵).

**d. Tiếp điểm thường đóng song song :**

Kích vào biểu tượng  hoặc nhấn phím F6 .Trong cửa sổ Input device , gõ vào tên thiết bị ngõ vào ( Input device). Chọn **OK** hay **Enter** (↵).

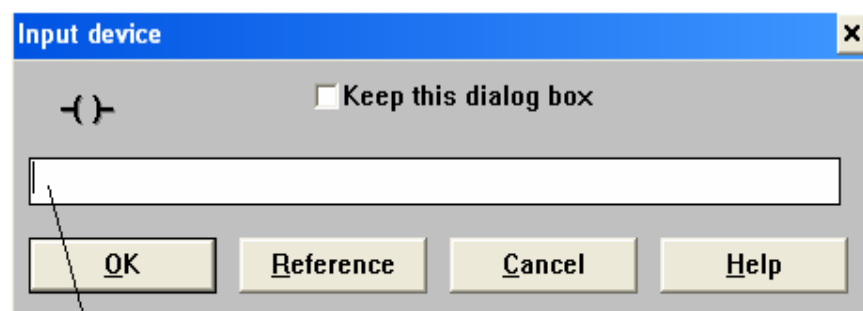
**e. Xử lý trạng thái mạch:**

\*Mạch thẳng: Kích vào  hoặc nhấn phím F9 và **Enter** (↵).

\*Mạch rẽ nhánh: Kích vào  hoặc nhấn phím Shift+F9 và **Enter** (↵).

**f. Cuộn dây:**

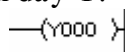
Kích vào  hoặc nhấn phím F7 .Xuất hiện cửa sổ Input device



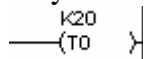
Gõ vào tên thiết bị ngõ ra (output device). Chọn OK hay enter

**Các dạng cuộn dây:**

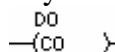
\*Cuộn dây Y:



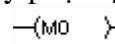
\*Cuộn dây Timer (T):



\*Cuộn dây Counter (C):



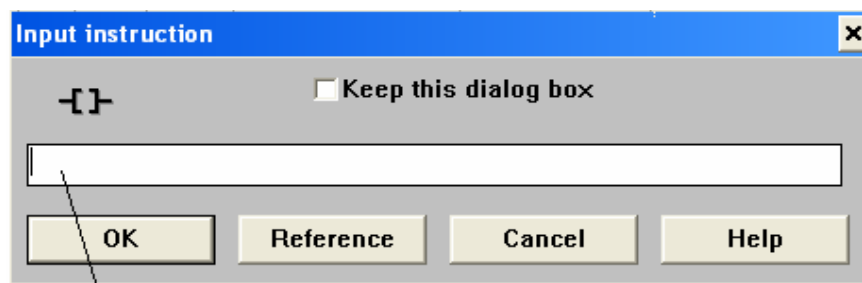
\*Relay phụ trợ M:



\*Relay trạng thái S: (cờ hiệu)

**g. Cung cấp hàm truyền đạt :**

Kích vào biểu tượng hoặc nhấn phím F8 .Xuất hiện khung cửa sổ Input instruction



Gõ vào tên hàm truyền. Chọn OK hay enter

Ví dụ:

SET S0  
→ [SET S0 ]

**2. Dạng INSTRUCTION:**

Đây là dạng lập trình PLC được sử dụng các lệnh cơ bản trong chương 3 đã trình bày ở trên.

**Giaodiện:**



Gõ vào tên hàm truyền. Hay lệnh

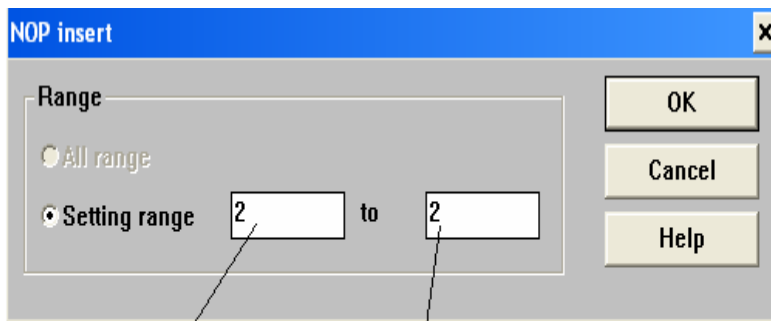
**b.Thanh công cụ:**



- \*Lệnh LD: nhấn phím F5
- \*Lệnh AND: nhấn phím F6
- \*Lệnh OR: nhấn phím F7
- \*Lệnh ANB: nhấn phím F8
- \*Lệnh OUT: nhấn phím F9

**Chú ý:**

Chèn một dòng trong vùng lập trình dạng **Instruction:**  
Trong thanh **Menu** → **Edit**→**NOP insert..**



Số cần chèn                      số đích tại vị trí vết sáng

Undo	Ctrl+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Delete
NOP overwrite...	
NOP insert...	
NOP remove...	

**V. BÀI TẬP ỨNG DỤNG PHẦN MỀM FX-WINE**

**A-Những hàm logic cơ bản dùng trong lập trình :**

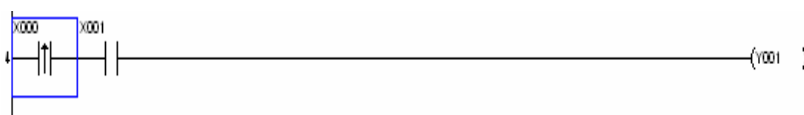
**a. Hàm LD :**

0    LD                      X001

**b. Hàm LDI :**

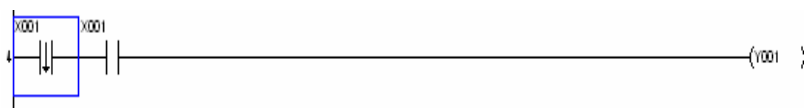
0    LDI                      X000

**c. Hàm LDP :**



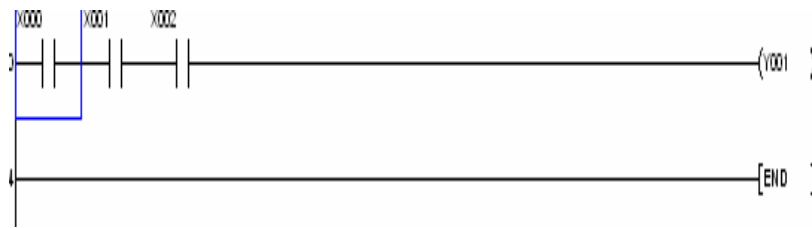
0	LDP	X000
2	AND	X001
3	OUT	Y001

**d. Hàm LDF :**



0	LDF	X000
2	AND	X001
3	OUT	Y001

**e. Hàm AND :**



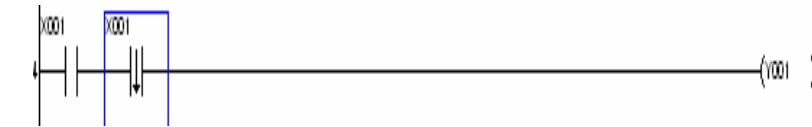
0	LD	X000
1	AND	X001
2	AND	X002
3	OUT	Y001

**f. Hàm ANDI :**



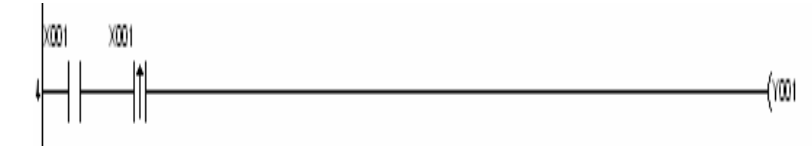
0	LDI	X000
1	ANI	X001
2	ANI	X002
3	OUT	Y001

**g. Hàm ANF :**



0	LD	X000
1	ANF	X001
3	OUT	Y001

**h. Hàm ANP :**



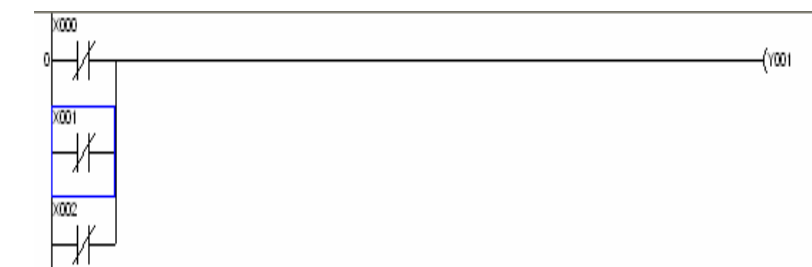
0	LD	X000
1	ANP	X001
3	OUT	Y001

**i. Hàm OR :**



0	LD	X000
1	OR	X001
2	OR	X002
3	OUT	Y001

**j. Hàm ORI :**



0	LDI	X000
1	ORI	X001
2	ORI	X002
3	OUT	Y001

**k. Hàm ORP :**



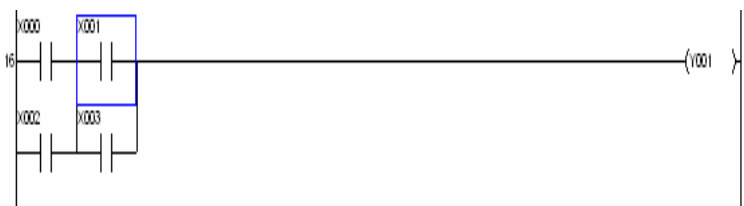
1	LD	X000
2	ORP	X000
4	ANB	
5	OUT	Y001

**l. Hàm ORF :**



1	LD	X000
2	ORF	X000
4	ANB	
5	OUT	Y001

**m. Hàm ANB :**



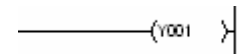
0	LD	X000
1	OR	X002
2	LD	X001
3	OR	X003
4	ANB	
5	OUT	Y001

**n. Hàm ORB :**



0	LD	X000
1	AND	X001
2	LD	X002
3	AND	X003
4	ORB	
5	OUT	Y001

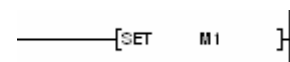
OUT Y001



**o. Hàm OUT :**

**p. Hàm SET :**

SET M1



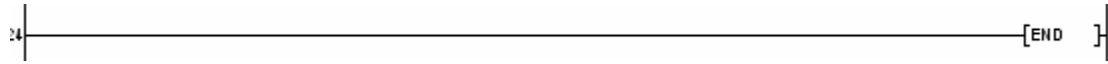
**q. Hàm RST :**

RST M1



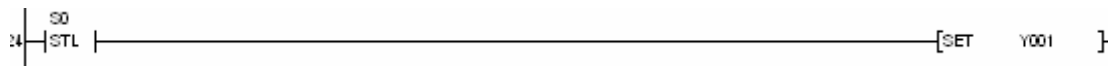
**r. Hàm END :**

Đây là hàm kết thúc một chương trình lập trình .



**s. Hàm tuần tự STL:**

Trong đó ta có hàm kết thúc **RET** dùng để khởi tạo lại trạng thái ban đầu của chương trình tuần tự. Thông qua các cờ S0 đến S9 cho phép chúng ta lựa chọn trạng thái khởi tạo thông qua các cờ chuyên dùng M8041 hoặc các cờ chức năng khác.



**B-BÀI TẬP ỨNG DỤNG :**

Ký hiệu các thiết bị trong lập trình:

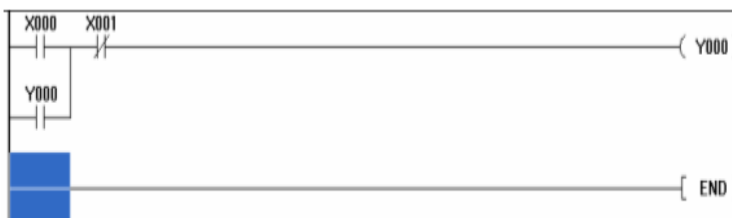
- 1.Thiết bị ngõ vào (Input ) là X.
- 2.Thiết bị ngõ ra (Output ) là Y.
- 3.Cờ trạng thái (Internal flag) là M hoặc S.
- 4.Bộ định thì (Timer) là T.
- 5.Bộ đếm (Counter) là C.
- 6.Hằng số (Constant) là K ,được dùng trong Timer và Counter.

**I.Dạng bài tập cơ bản:**

1.Viết một chương trình PLC điều khiển trực tiếp động cơ không đồng bộ ba pha sử dụng ngôn ngữ **Ladder** và **Instruction**.

Bài làm:

**a.Dạng Ladder:**



**b.Dạng Instruction:**

```
LD X000
OR Y000
ANI X001
OUT Y000
END
```

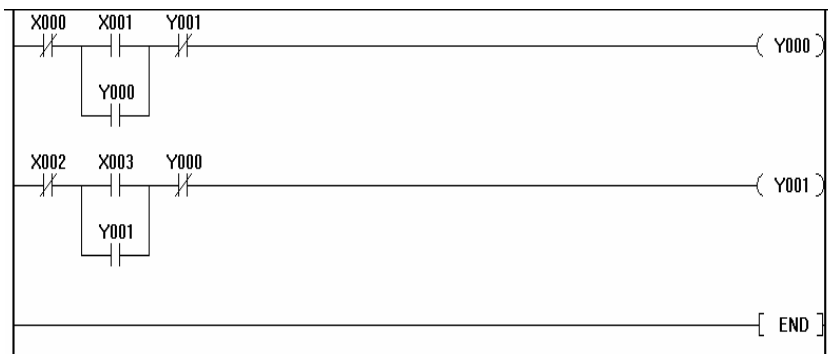
**c.Các ngõ I/O:**

- \*Nút nhấn OFF →X001.
- \*Nút nhấn ON →X000.
- \*Cuộn dây Contacter điều khiển động cơ ba pha. Y0

2. Viết một chương trình PLC điều khiển động cơ không đồng ba pha quay ở hai chế độ thuận và nghịch.

Bài làm:

*a. Dạng Ladder:*



*b. Dạng Instruction:*

```
LDI      X000
LD       X001
OR       Y000
ANB
ANI      Y001
OUT      Y000
LDI      X002
LD       X003
OR       Y001
ANB
ANI      Y000
OUT      Y001
END
```

*c. Các ngõ I/ O:*

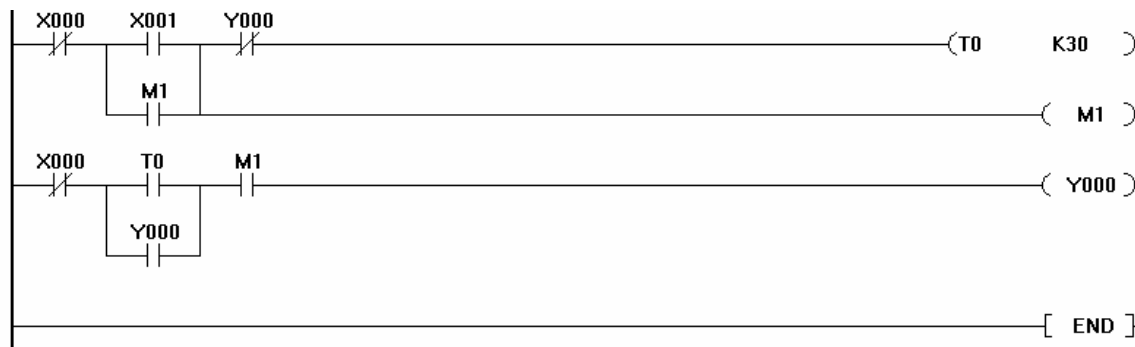
- \*Nút nhấn OFF → X000 và X002.
- \*Nút nhấn ON → X001 và X003.
- \*Cuộn dây Contacter quay thuận là Y000.
- \* Cuộn dây Contacter quay nghịch là Y001.

## II. Các bài tập về Timer:

1. Viết một chương trình khởi động động cơ với yêu cầu sau: khi tác động ON sau 3 giây động cơ hoạt động.

Bài làm:

*a. Dạng Ladder:*



***b.Dạng Instruction:***

LDI	X000
LD	X001
OR	M1
ANB	
MPS	
ANI	Y000
OUT	T0 K30
MPP	
OUT	M1
LDI	X000
LD	T3
OR	Y000
ANB	
AND	M1
OUT	Y000
END	

***Các ngõ I/O:***

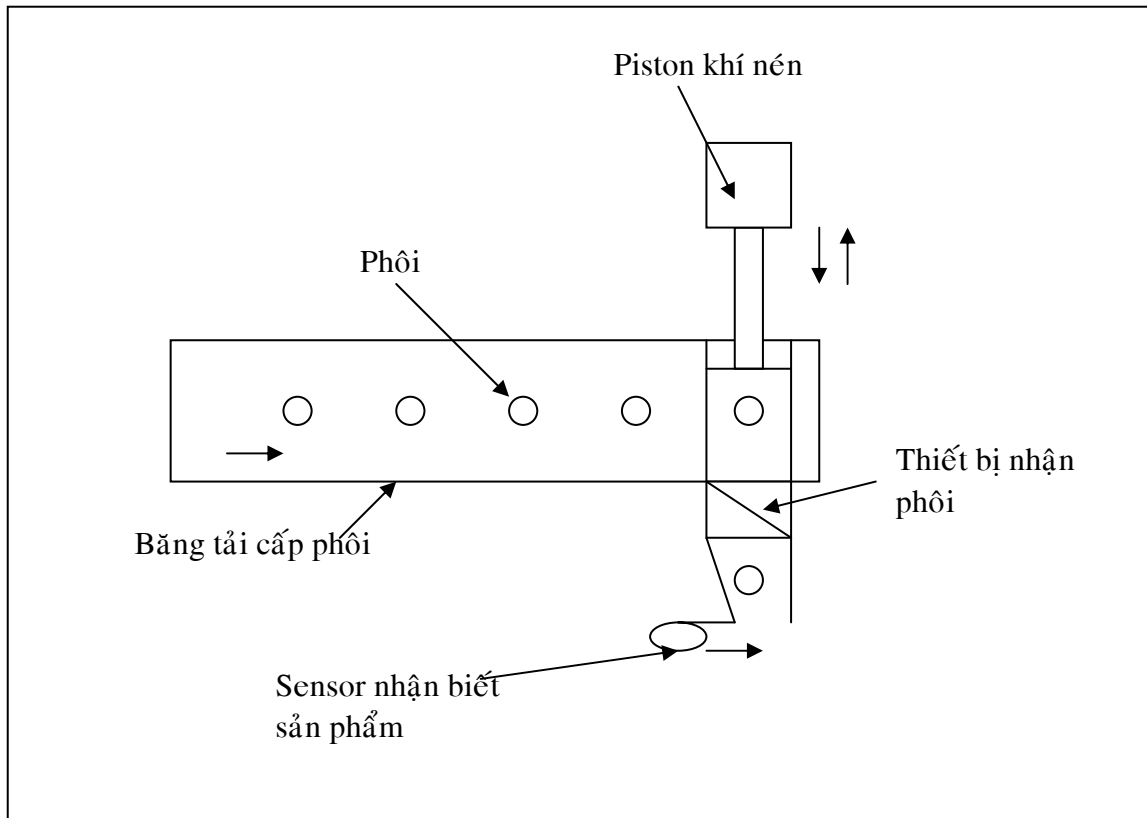
- \*Nút nhấn OFF là X000
- \*Nút nhấn ON là X001
- \*Rơle phụ M1
- \*Timer là T0 K30
- \*Cuộn dây Contacter là Y000.

**III. Một số bài tập nâng cao :**

- 1.Viết chương trình điều khiển một băng tải với những yêu cầu sau :băng tải chuyển động theo chiều từ trái sang phải sau thời gian 5 giây băng tải sẽ chuyển động ngược trở lại .
- 2.Viết chương trình điều khiển động cơ với yêu cầu sau : động cơ 1 hoạt động sau khi nhấn ON , sau 5 giây thì động cơ 2 hoạt động .
- 3.Viết chương trình điều khiển mạch đèn cầu thang hai chế độ .
4. Viết chương trình điều khiển mạch đèn cầu thang ba chế độ .
5. Viết chương trình điều khiển mạch đèn hầm lò .
- 6.Viết chương trình điều khiển động cơ với yêu cầu như sau : động cơ 1 hoạt động khi nhấn ON sau 5 giây động cơ 2 hoạt động và sau 3 giây động cơ 3 hoạt động .Khi tác động OFF thì động cơ 3 ngưng hoạt động , sau 2giây động cơ 1 và 2 ngưng hoạt động .



7. Viết chương trình điều khiển trình tự cơ cấu cấp phôi cho máy dập :

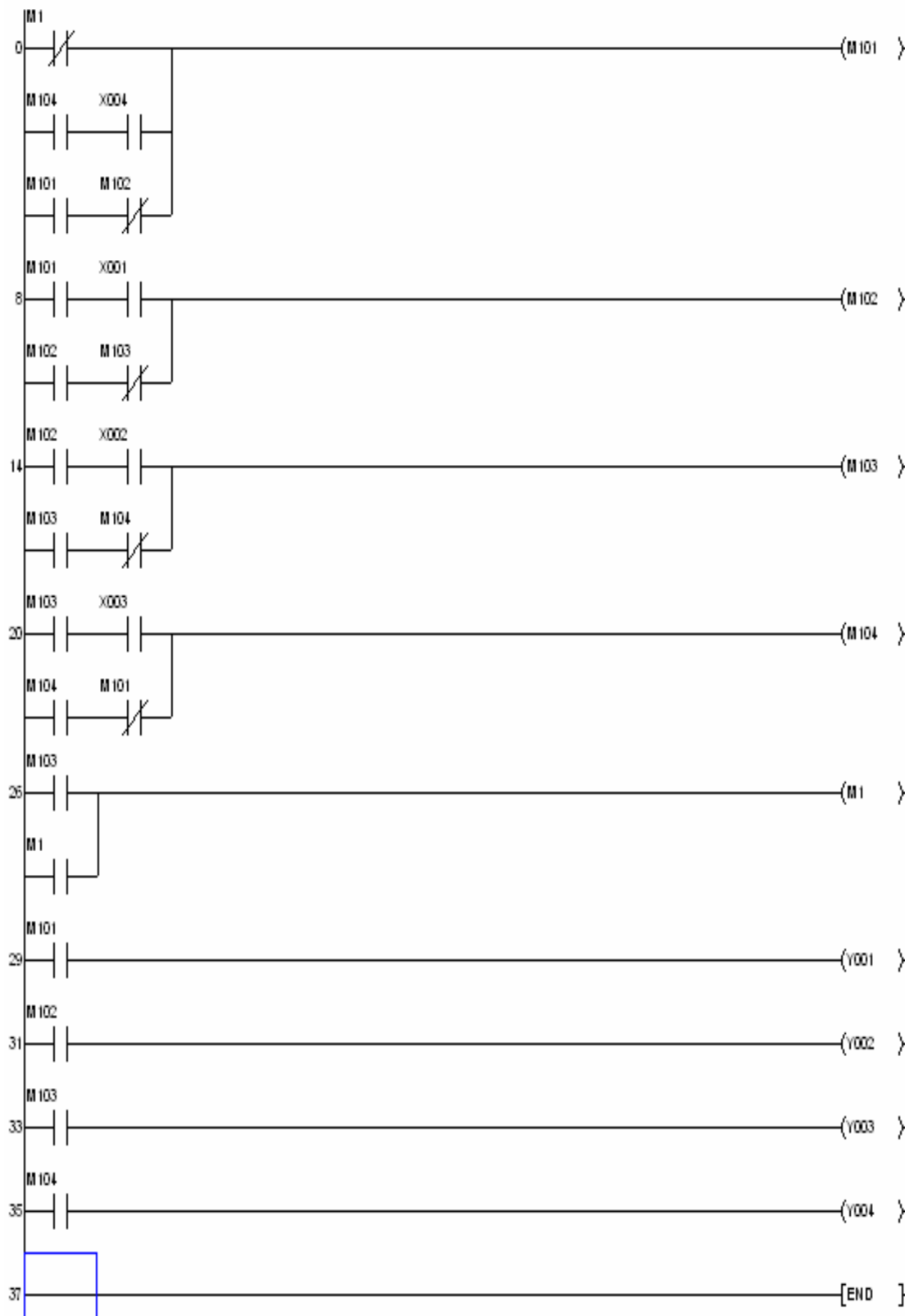


**Nguyên lý hoạt động của cơ cấu :**

- Trạng thái 1 :Băng tải hoạt động , xy- lanh co , cơ cấu dập không hoạt động .
- Trạng thái 2 :Xy-lanh duỗi ra , băng tải dừng , cơ cấu dập không hoạt động .
- Trạng thái 3 :Xy lanh co , băng tải dừng , cơ cấu kẹt không hoạt động .
- Trạng thái 4 :Cơ cấu dập hoạt động , băng tải dừng , xy lanh co.
- Sau đó quay về trạng thái 1 .

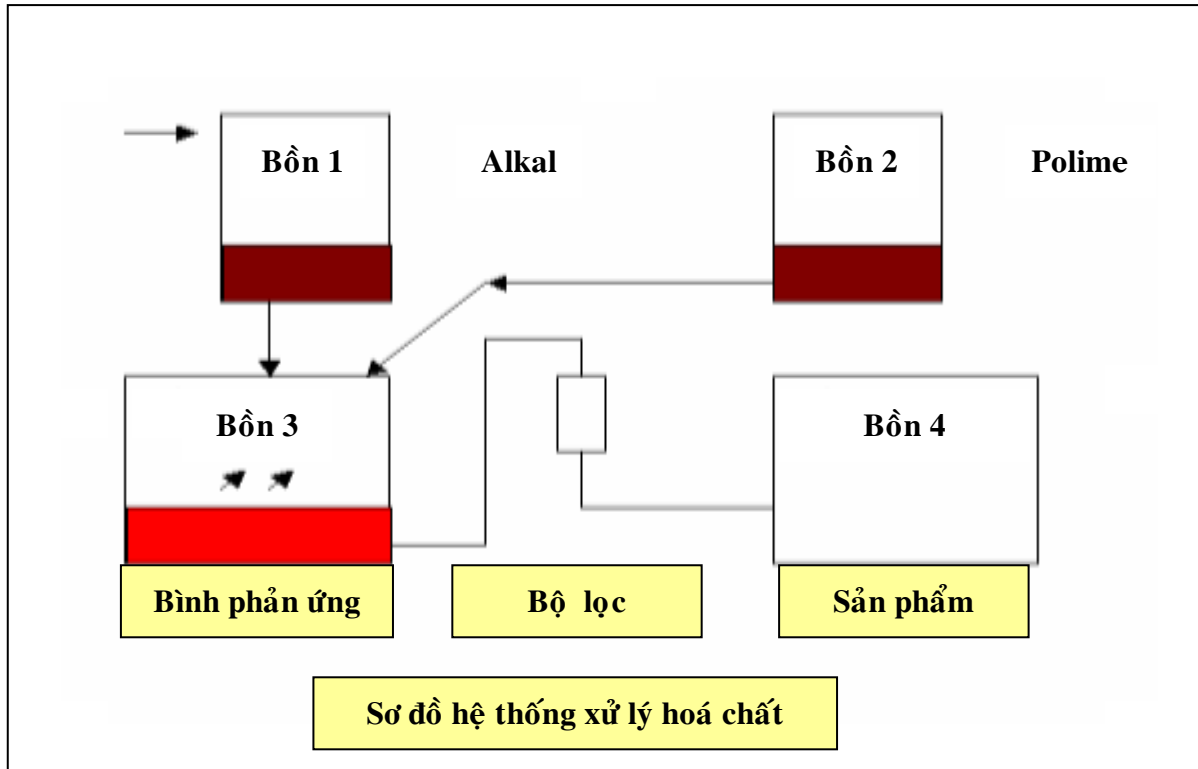
Thiết bị	I/O trên PLC	Thiết bị	I/O trên PLC
LS1	x1	Trạng thái 3	m103
LS2	x2	Trạng thái 4	m104
LS3	x3	Băng tải	y1
LS4	x4	Xy lanh duỗi	y2
Trạng thái 1	m101	Xy lanh co	y3
Trạng thái 2	m102	Dập	y4
		Khởi động	m1

Chương trình điều khiển :



### 8. Viết chương trình điều khiển máy xử lý hóa chất :

Hệ thống gồm 4 bồn chứa có các bơm để chuyển chất lỏng qua hệ thống , mỗi bồn được gắn một cảm biến để nhận biết khi nào bồn cạn hay đầy và bồn 2 phần tử phát nóng được nối với cảm biến nhiệt độ . Bồn 3 được gắn một cần khuấy để trộn hai thành phần tử lỏng khi chúng được vào bồn 1 và 2 . Các bồn phía dưới , bồn 3 và 4 , có dung tích gấp đôi bồn 1 và 2



#### Nguyên lý hoạt động :

Bồn 1 và 2 được đổ đầy các từ các bồn chứa chất kiềm và polime riêng biệt , thông qua bơm 1 và 2 . Bơm 1 và 2 ngưng hoạt động khi có tín hiệu từ cảm biến báo đầy bồn .Phần tử phát nóng trong bồn 2 được kích hoạt , nâng nhiệt độ polime lên 60°C . Khi cảm biến nhiệt độ đóng , tín hiệu này sẽ tắt bộ điều khiển rung và kích hoạt bơm 3 và 4 để chuyển dung dịch vào bồn phản ứng , bồn 3. cần khuấy cũng được kích hoạt khi bồn này có hỗn hợp dung dịch và trong khoảng thời gian tối thiểu là 60 giây , bơm 5 sẽ chuyển hỗn hợp đã trộn vào bồn 4 , bồn sản phẩm , thông qua bộ lọc . Bơm 5 dừng hoạt động khi bồn 4 đầy và bồn 3 cạn . Cuối cùng , sản phẩm dung dịch được đưa vào bồn chứa lưu trữ .Quá trình xử lý kết thúc một chu kỳ hoạt động.

Viết chương trình điều khiển với nguyên lý trên.

# LỰA CHỌN, LẮP ĐẶT, KIỂM TRA VÀ BẢO TRÌ HỆ THỐNG PLC

## I. Xem xét sự khả thi :

Trong trường hợp nào đó việc xem xét về sự khả thi là yêu cầu không thể thiếu trong một quyết định hay một giải pháp điều khiển. Phạm vi của lĩnh vực này có thể thay đổi rất nhiều, từ việc đơn giản là đặc tả tính khả thi của yêu cầu cho đến việc phân tích toàn diện mọi trường hợp. Việc nghiên cứu tính khả thi bao gồm những lĩnh vực sau:

- tính khả thi về kinh tế
- tính khả thi về kỹ thuật
- sự lựa chọn các phương án : nghiên cứu đánh giá các phương án để đưa ra phương án tối ưu.

## II. Trình tự thiết kế hệ thống PLC:

Vì PLC được thiết kế dạng modun nên gần như việc thiết kế và cài đặt phần cứng, phần mềm được thực hiện đồng thời nhưng độc lập với nhau. Việc thực hiện như vậy có ưu điểm là tiết kiệm thời gian và hệ thống linh hoạt bất chất chức năng của hệ thống.

### 1. Chọn PLC :

Hiện nay trên thị trường có rất nhiều loại PLC với các tính năng ngày càng được tăng cường nhằm cải thiện hiệu suất và chất lượng. Tuy nhiên, hầu hết PLC cùng cỡ thì có các chức năng điều khiển tương đương nhau. Điểm khác nhau quan trọng nhất là ở ngôn ngữ lập trình, cùng với các mức độ về sự hỗ trợ của nhà sản xuất. Sự hỗ trợ từ nhà sản xuất là yếu tố quan trọng khi thực hiện thiết kế một hệ thống điều khiển tự động.

Việc chọn PLC của hãng nào là do sự quen dùng đối với PLC đó và hệ thống điều khiển nói chung. Đối với những người có kinh nghiệm trong việc thiết kế và lắp đặt hệ thống điều khiển, vấn đề quan trọng là sự vượt trội về kỹ thuật và hiệu suất. Đối với những người chưa có kinh nghiệm nhiều về PLC và không nắm vững về thị trường PLC thì cần xem xét những vấn đề sau đây :

- người dùng có nhận được sự hỗ trợ trong công việc thiết kế?
- Tỷ lệ về thị trường và lĩnh vực ứng dụng của nhà sản xuất.
- Nhà sản xuất có tổ chức các khoá huấn luyện về hệ thống PLC đang sử dụng?
- Sổ tay và tài liệu hiện có với ngôn ngữ đọc được.
- Khả năng tương thích giữa các hệ thống tương đương hoặc loại PLC khác của cùng nhà sản xuất.
- Phương pháp lập trình có thích hợp với các điều khiển ứng dụng.

### 2. Loại và cỡ PLC:

Sự lựa chọn này có thể thực hiện cùng với việc lựa chọn nhà sản xuất PLC. Quy mô của hệ thống có một số điểm cần xem xét:

- Yêu cầu ngõ vào ra cần thiết

- Loại ngõ vào/ra.
- Dung lượng bộ nhớ.
- Tốc độ và khả năng của CPU và tập lệnh.

### 3. Số lượng ngõ vào ra:

Số lượng ngõ vào ra của PLC phải có khả năng đáp ứng đủ số đường tín hiệu từ cảm biến cũng như đường điều khiển phần công suất do cơ cấu tác động. Các tín hiệu không những tuân theo các chỉ tiêu kỹ thuật của hệ thống như là mức điện áp, dòng tải, tần số đáp ứng mà còn quan tâm đến các điểm sau:

- Số lượng và ngõ vào ra trên mỗi modul(hay trên PLC nếu các ngõ vào ra có sẵn trên PLC) .
- Sự cách ly giữa bộ điều khiển và phần công suất điều khiển cơ cấu tác động.
- Nhu cầu về ngõ vào ra xử lý tốc độ cao, điều khiển đầu ra từ xa, hay các chức năng chuyên dùng khác.
- Nhu cầu mở rộng thêm khả năng mở rộng và lắp đặt thêm ngõ vào ra.
- Nguồn cấp điện cho các ngõ vào ra, nghĩ là nhu cầu có modul PSU (Power Source Unit) cung cấp cho mạch chuyển đổi tín hiệu hay cơ cấu tác động

### 4. Dung lượng bộ nhớ:

Đối với PLC có khả năng mở rộng bộ nhớ thì dung lượng bộ nhớ có thể được mở rộng bằng cách gắn thêm hộp bộ nhớ (**memory cassette**). Dung lượng bộ nhớ phụ thuộc vào số lượng ngõ vào ra sử dụng trên hệ thống. Một chương trình điều khiển phức tạp, dùng nhiều logic khóa lẫn hay các chương trình trình tự hiển nhiên cần nhiều bộ nhớ hơn là chương trình đơn giản.

**BỘ NHỚ CẦN THIẾT = BỘ NHỚ NGÕ VÀO RA + BỘ NHỚ CHƯƠNG TRÌNH + BỘ NHỚ LƯU DỮ LIỆU CHO CÁC LỆNH CHUYÊN DÙNG + KHOẢNG DUNG LƯỢNG CẦN THIẾT CHO MỞ RỘNG VÀ HIỆU CHỈNH CHƯƠNG TRÌNH SAU NÀY.**

### 5. Tập lệnh CPU:

Mọi PLC đều phải có tập lệnh để phục vụ cho việc lập trình giải quyết các nhiệm vụ điều khiển. Tất cả các PLC đều có thể hiểu được lệnh điều khiển logic, điều khiển trình tự... sự khác nhau nổi bật là khả năng xử lý dữ liệu, những chức năng chuyên dùng và truyền thông. PLC loại lớn có tập lệnh mạnh hơn các PLC nhỏ hơn. Tuy nhiên ta nên xem xét kỹ khả năng của PLC loại nhỏ và trung bình thường có thể cũng có các chức năng chuyên dùng rất tốt như chức năng điều khiển PID.

Ở các PLC dạng modul có thể lựa chọn được modul CPU với các mức độ và tốc độ và khả năng xử lý. Khi số lượng và khả năng xử lý tăng, yêu cầu về tốc độ CPU cũng tăng, vì khi đó CPU phải xử lý nhiều lệnh hơn trong một chu kỳ quét của PLC. Điều đó có thể đòi hỏi phải dùng CPU mạnh hơn nếu thời gian quét không đáp ứng được nhu cầu.



### 3. Lắp đặt:

Việc lắp đặt phần cứng bao gồm việc bố trí các tủ điện và các thanh gắn trên PLC, sau đó lắp đặt và nối dây.

Tủ gắn PLC và các khe cắm phụ phải được đáp ứng đầy đủ yêu cầu về môi trường, như vấn đề bảo mật, an toàn và bảo vệ như sau:

- Bảo mật: tủ đặt ở nơi chất chắn và cửa có khoá.
- An toàn: trang bị mạch tự động ngắt hay báo động nếu tủ điện bị mở
- Bảo vệ: tránh độ ẩm hay môi trường ăn mòn do không khí bằng cách làm kín trên khung nắp. Việc ngăn tĩnh điện bằng cách nối đất cho thân khung lắp.
- Với mục đích bảo trì, các khe cắm được bố trí dễ dàng thực hiện kiểm tra, thay thế các modul. Các đèn chỉ thị on/off hay đèn chỉ thị trạng thái hoạt động nên gắn trên cửa tủ.

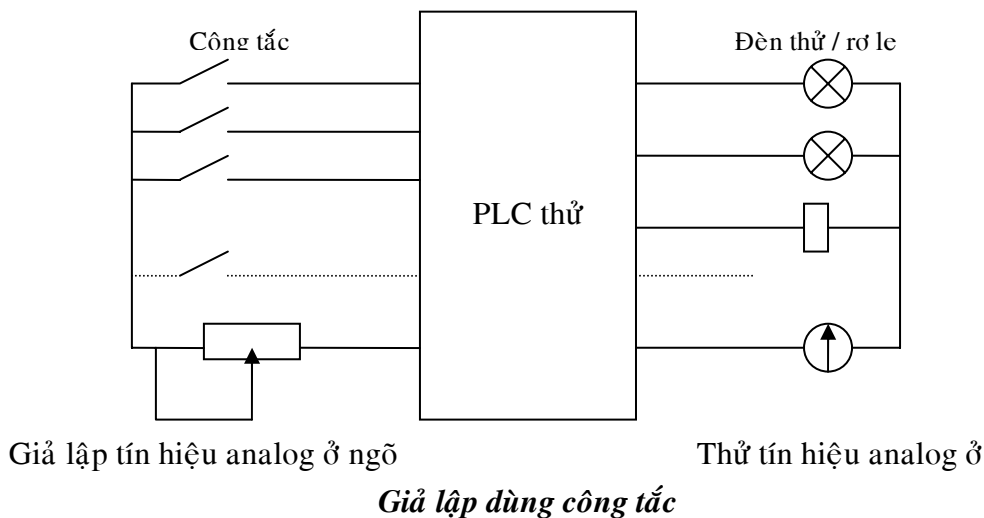
### IV. Chạy thử chương trình:

Khi hoàn tất việc lắp đặt hệ thống, bước kế tiếp là thử hệ thống. Việc chạy thử chương trình gồm hai giai đoạn cơ bản:

- Kiểm tra việc kết nối dây giữa PLC và các thiết bị được điều khiển.
- Cài chương trình điều khiển và chạy thử chương trình

Việc nối kết hệ thống phải được thực hiện cẩn thận để đảm bảo các ngõ vào/ra được nối đúng đến các ngõ vào/ra trên PLC. Trước khi phần cứng được kiểm tra, phải kiểm tra lại toàn bộ các nguồn cấp điện chính, việc nối đất.

*Chạy thử chương trình với ngõ vào giả lập:*



Lắp đặt và chạy chương trình trên hệ thống thật: khi chương trình đã chạy thử tốt, ta có thể chuyển sang bước thử trên thiết bị thực tế .

### ***Hiệu chỉnh chương trình:***

Một trong những ưu điểm lớn của PLC là dễ dàng thay đổi chương trình. Ở giai đoạn thiết kế và hoàn tất dự án, ưu điểm này cho phép thay đổi và thử đi thử lại nhiều lần chương trình đến khi thoả mãn thiết kế.

Lưu chương trình dự phòng: Khi hoàn tất giai đoạn thử và hiệu chỉnh chương trình, các bản sao dự phòng tương ứng với từng giai đoạn phải được lưu trữ kèm theo các mô tả về chúng. Ngay cả khi chương trình được viết xong nên lưu một bản dự phòng để nạp lại chương trình khi bộ nhớ trên PLC bị hư.

### ***V.Lập tài liệu cho hệ thống:***

Tập tài liệu đầy đủ cho việc lắp đặt PLC nên bao gồm các phần sau:

- Các đặc tả về kế hoạch hay tiến trình thực hiện.
- Các đặc tả và đặc điểm về yêu cầu điều khiển
- Các đặc tả về PLC gồm cả sổ tay về thiết bị cần điều khiển.
- Các đặc tả về các chức năng của chương trình điều khiển, gồm lưu đồ hay sơ đồ chức năng, bộ nhớ và sự cấp phát bộ nhớ.
- Toàn bộ chương trình cùng với các chú thích và giải thích.
- Tập chương trình dự phòng.
- Sơ đồ lắp đặt điện và mô tả
- Danh sách các thiết bị xuất nhập dữ liệu gồm cả màn hình hiển thị, máy in...
- Sổ tay sử dụng ghi chi tiết trình tự khởi động hệ thống, dừng hệ thống và các báo động của hệ thống.

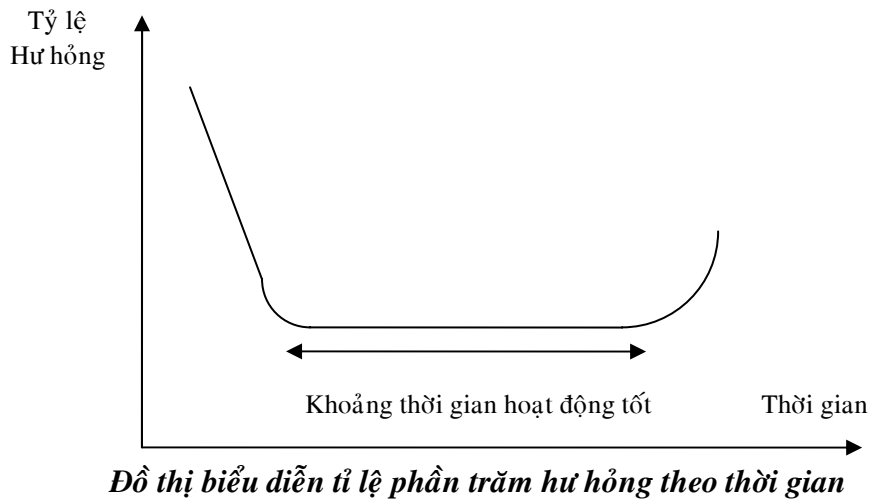
### ***VI.Bảo trì hệ thống PLC:***

#### ***1. Lỗi do PLC:***

PLC được thiết kế để hoạt động tinh cậy và bền vững trong môi trường công nghiệp. Đây là ưu điểm chính vượt trội so với điều khiển cơ điện tử truyền thống và điều khiển từ máy vi tính. Do cấu trúc mạch điện tử mạch và cảm tính điện hiệu quả nên các mạch điện trong PLC sử dụng an toàn ở các nơi không thuận lợi về môi trường, dễ gây ra các hư hỏng cho phần cứng về điện và về vật lý.

PLC được bảo vệ tránh các khả năng hư hỏng trên các ngõ vào/ra bởi các mạch cách ly quang (**opto-isolated**). Việc dùng bộ nhớ RAM có nguồn pin nuôi hay EEPROM lưu giữ được chương trình bảo đảm được sản xuất vẫn được duy trì nếu chương trình bị mất hay sai do nguồn cấp điện bị hỏng hay trong các trường hợp tương tự. Tóm lại, các biện pháp khả thi trong thiết kế được áp dụng cho PLC nhằm đạt được độ tin cậy cao có thể được với giá thành hợp lý.





Tuy nhiên, độ tin cậy cao không có nghĩa là hệ thống hoàn toàn không bị hỏng. Dù sử dụng các vi mạch điện tử có chất lượng cao và được lắp ráp theo tiêu chuẩn cao thì PLC vẫn có lỗi. PLC thường được thử rất kỹ trước khi xuất xưởng và chúng được cho chạy liên tục chương trình thử trong thời gian dài.

**Lỗi do PLC:** các lỗi nghiêm trọng làm cho PLC bị ngưng hoạt động, một số lỗi khác có thể cho phép PLC tiếp tục hoạt động, hiển thị mã lỗi trên các màn hình thông báo hay trên các đèn bẫy đoạn. Khi **Self-test** không thành công, PLC bị ngưng hoạt động và không khởi động được cho đến khi lỗi được khắc phục.

### 2. Lỗi từ phần cứng bên ngoài PLC:

PLC là một phần trong phần điều khiển; trong hệ thống có cảm biến, cơ cấu tác động, dây kết nối, nguồn cấp điện và chương trình điều khiển. Mỗi phần đều có khả năng hư hỏng; tuy nhiên phần lớn tỷ lệ hư hỏng thuộc về các bộ phận, thiết bị nằm bên ngoài PLC như:

- Hư hỏng thiết bị vào/ra – mạch chuyển đổi tín hiệu hay cơ cấu tác động.
- Hư hỏng phần dây kết nối.
- Hư hỏng phần dây kết nối truyền thông.
- Nguồn cấp điện không ổn định – nhiễu hai mức nguồn.

### 3. Trình tự lập trình báo lỗi:

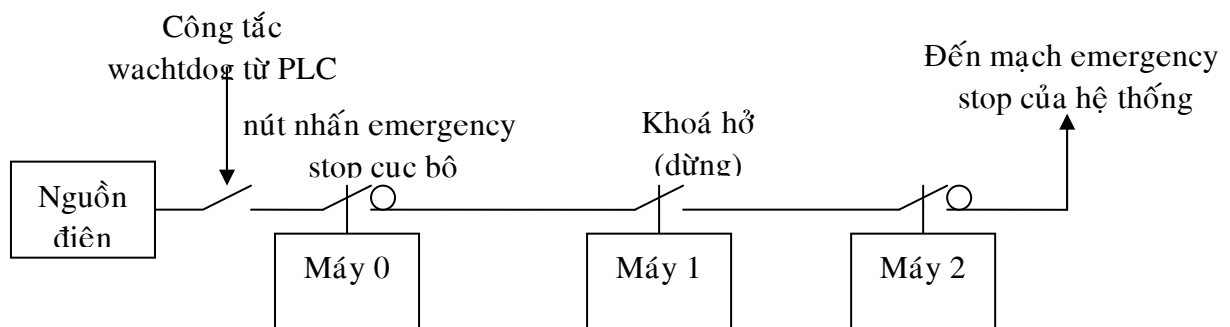
**Vấn đề bảo vệ:** Một dạng lỗi trong PLC về xử lý thương liên quan đến mạch bảo vệ (**watchdog**). Hầu hết PLC có trang bị rơle bảo vệ (**watchdog relay**) bên trong dùng để điều khiển nguồn cấp điện cho một hay nhiều thiết bị bên ngoài.

**Vấn đề an toàn:** Mặc dù chương trình có hoàn hảo và tinh vi đến đâu thì cũng có khi xử lý sai, chương trình sẽ làm việc không đúng ngay cả khi PLC đang hoạt động bình thường. Vì không có PLC nào có độ tin cậy là 100% nên là chỉ có cách là thiết kế một hệ thống an toàn thông qua mạch phần cứng hơn là thông qua phần mềm.

**4. Mạch an toàn bằng phần cứng:**

Các mạch này phải được thiết kế độc lập với PLC và nên phối hợp với các công tắc bảo vệ (**watchdog relay**) của PLC để đảm bảo mạch này thực hiện việc ngắt hệ thống nếu PLC bị lỗi.

Thường nút nhấn Start và Stop được thiết kế nối vào ngõ vào của PLC, nghĩa là Start và Stop được thực hiện bởi chương trình, trong khi đó công tắc và nút nhấn **EMERGENCY STOP** (dừng khẩn cấp) dùng để dừng sự chuyển động của máy và các thiết bị có nhiều khả năng gây nguy hiểm phải được thiết kế ở mạch phần cứng.



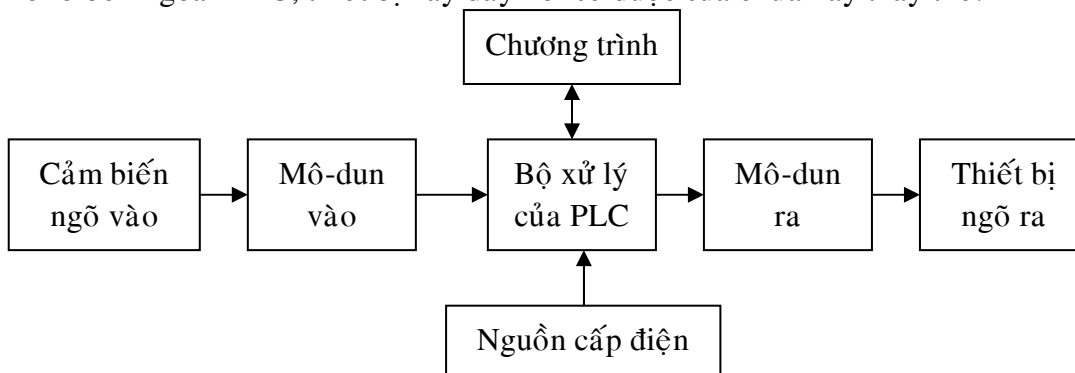
**Mạch phần cứng EMERGENCY STOP thông dụng**

**5. Sửa lỗi :**

Khi hệ thống điều khiển chưa làm việc hay đang chạy đúng, nguyên nhân các hư hỏng phải được xác định, phải được khắc phục theo trình tự:

- Ghi chú và nghiên cứu các triệu chứng.
- Xác định các vùng có nghi vấn.
- Cô lập lỗi.
- Sửa lỗi.

**Sửa lỗi :** nếu lỗi đã được xác định thì cần phải thay thế một modun nào đó; khi đó, tùy thuộc vào chức năng của modun đó mà có thể phải cấu hình lại cho hệ thống. Nếu lỗi xảy ra trong chương trình thì người thiết kế hệ thống nên xem xét kỹ trước khi thực hiện thay đổi nào đó của chương trình, vì còn có các hoạt động khác liên quan đến các dòng lệnh sẽ bị thay đổi. Khi lỗi ở bên ngoài PLC, thiết bị hay dây nối sẽ được sửa chữa hay thay thế.



**Các thành phần của hệ thống PLC**