

Mục lục

Chương 1 : Tổng quan PLC – Giới thiệu dòng PLC FX-3U	2
I. TÓNG QUAN VỀ PLC	2
1. Giới thiệu tổng quan	2
2. Một số dòng PLC của họ FX	2
II. Giới thiệu về PLC dòng FX-3U	2
1. Giới thiệu PLC FX-3U	2
2. Phân loại	3
3. Sơ đồ Nguồn	5
4. Sơ đồ Input Output	5
Chương 2 : Giới thiệu phần mềm - cách khai báo 1 Project	7
I. Phần mềm lập trình PLC GX-Works2	7
II. Sử dụng phần mềm và cách tạo một project	8
1. Sử dụng phần mềm	8
2. Cách tạo một project mới	8
III. Cách download chương trình xuống PLC	10
1. Download chương trình xuống CPU	10
Chương 3 : Các vùng nhớ và tập lệnh cơ bản	13
1. Các vùng nhớ	13
2. Các tập lệnh cơ bản	13
3. Một số lệnh toán học và lệnh di chuyển	17
Chương 4: Lệnh TIMER và lệnh COUNTER	19
1. Lệnh Timer	19
2. Lệnh COUNTER	20
Chương 5 : Lập trình HMI	24
1. Phần mềm GT Designer 3	24
2. Cách tạo một project mới trên GT Designer	24
3. Cách gán Tag IN, OUT cho một chương trình đơn giản	26
4. Chạy mô phỏng trên máy tính	29
Chương 6 : Tín hiệu tương tự	31
1. Xử lý tín hiệu Analog	31
Chương 7 : Lập trình và điều khiển động cơ Servo	34
1. Giới thiệu chung về động cơ Servo	34
2. Một số lệnh phát xung trên FX-3U	34
Chương 8: Bài Tập	37
BÀI 1: ĐIỀU KHIỂN CỬA CUỐN	37
BÀI 2: BÀI TOÁN ƯU TIÊN	38
BÀI 3: DÂY CHUYỀN ĐÓNG GÓI	39
BÀI 4: ĐIỀU KHIỂN TRỘN LIỆU	40

Chương 1 : Tổng quan PLC – Giới thiệu dòng PLC FX-3U

I. TỔNG QUAN VỀ PLC

1. Giới thiệu tổng quan

PLC là viết tắt của Programmable Logic Controller, là thiết bị điều khiển lập trình được (khả trình) cho phép thực hiện linh hoạt các giải pháp điều khiển logic thông qua một ngôn ngữ lập trình.

PLC có các ưu điểm:

- + Lập trình PLC khá đơn giản, ngôn ngữ lập trình dễ học.
- + Gọn nhẹ, dễ dàng bảo trì và sửa chữa.
- + Dung lượng bộ nhớ lớn để có thể chứa được những chương trình phức tạp.
- + Hoạt động tin cậy trong môi trường công nghiệp.
- + Giao tiếp được với các thiết bị thông minh khác như: máy tính, HMI, các mô đun mở rộng, mạng truyền thông
- + Giá cả cạnh tranh.

2. Một số dòng PLC của họ FX

- FX0-FX0S
- FX1S
- FX1N
- FX2N
- FX3G
- FX3U
- FX3UC
- FX5U

II. Giới thiệu về PLC dòng FX-3U

1. Giới thiệu PLC FX-3U

Dòng sản phẩm mới PLC FX3U là thế hệ thứ ba trong gia đình họ FX-PLC, là một PLC dạng nhỏ gọn và thành công của hãng Mitsubishi Electric.

Sản phẩm được thiết kế đáp ứng cho thị trường quốc tế, tính năng đặc biệt mới là hệ thống “adapter bus” được bổ sung cho hệ thống bus giúp cho việc mở rộng thêm những tính năng đặc biệt và khối truyền thông mạng. Khả năng tối đa có thể mở rộng lên đến 10 khối trên bus mới này.



Với tốc độ xử lý mạnh mẽ, thời gian chỉ $0.065\mu s$ trên một lệnh đơn logic, cùng với 209 tập lệnh được tích hợp sẵn và cải tiến liên tục đặc biệt cho nghiệp vụ điều khiển vị trí. Dòng PLC mới này còn cho phép mở rộng truyền thông qua cổng USB, hỗ trợ cổng Ethernet và Cổng lập trình RS-422 mini DIN.

Với tính năng mạng mở rộng làm cho PLC này nâng cao được khả năng kết nối tối đa về I/O lên đến 384 I/O, bao gồm cả các khối I/O qua mạng.

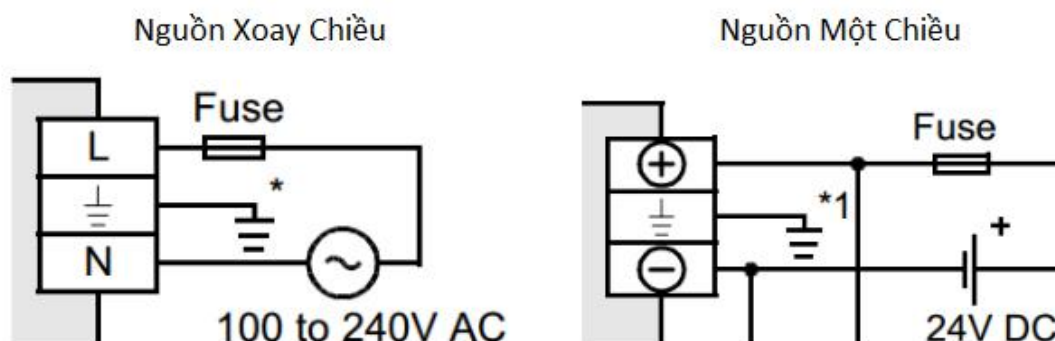
2. Phân loại

Mã	I/O tối đa	Input (số cổng)	Output (số cổng, loại)	Nguồn cấp	Công suất (W)	Trọng lượng (kg)	Kích thước (mm)
1. Main Units với 16 ngõ vào ra I/O							
FX3U-16MR/DS	16	8	8, Relay	24V DC	25	0.6	130x90x86
FX3U-16MR/ES-A	16	8	8, Relay	220V AC	30	0.6	130x90x86
Mã	I/O tối đa	Input (số cổng)	Output (số cổng, loại)	Nguồn cấp	Công suất (W)	Trọng lượng (kg)	Kích thước (mm)
FX3U-16MT/DS	16	8	8, Transistor(Sink)	24V DC	25	0.6	130x90x86
FX3U-16MT/ES-A	16	8	8, Transistor(Sink)	220V AC	30	0.6	130x90x86
FX3U-16MT/DSS	16	8	8, Transistor(Source)	24V DC	25	0.6	130x90x86
FX3U-16MT/ESS	16	8	8, Transistor(Source)	220V AC	30	0.6	130x90x86
2. Main Units với 32 ngõ vào ra I/O							
FX3U-32MR/DS	32	16	16, Relay	24V DC	30	0.65	150x90x86
FX3U-32MR/ES-A	32	16	16, Relay	220V AC	35	0.65	150x90x86
FX3U-32MT/DS	32	16	16, Transistor(Sink)	24V DC	30	0.65	150x90x86
FX3U-32MT/ES-A	32	16	16, Transistor(Sink)	220V AC	35	0.65	150x90x86

FX3U-32MT/DSS	32	16	16, Transistor(Source)	24V DC	30	0.65	150x90x86
FX3U-32MT/ESS	32	16	16, Transistor(Source)	220V AC	35	0.65	150x90x86
3.Main Units với 48 ngõ vào ra I/O							
FX3U-48MR/DS	48	24	24, Relay	24V DC	35	0.85	182x90x86
FX3U-48MR/ES-A	48	24	24, Relay	220V AC	40	0.85	182x90x86
FX3U-48MT/DS	48	24	24, Transistor(Sink)	24V DC	35	0.85	182x90x86
FX3U-48MT/ES-A	48	24	24, Transistor(Sink)	220V AC	40	0.85	182x90x86
Mã	I/O tối đa	Input (số cổng)	Output (số cổng,loại)	Nguồn cấp	Công suất (W)	Trọng lượng (kg)	Kích thước (mm)
FX3U-48MT/DSS	48	24	24, Transistor(Source)	24V DC	35	0.85	182x90x86
FX3U-48MT/ESS	48	24	24, Transistor(Source)	220V AC	40	0.85	182x90x86
4.Main Units với 64 ngõ vào ra I/O							
FX3U-64MR/DS	64	32	32, Relay	24V DC	40	1.00	220x90x86
FX3U-64MR/ES-A	64	32	32, Relay	220V AC	45	1.00	220x90x86
FX3U-64MT/DS	64	32	32, Transistor(Sink)	24V DC	40	1.00	220x90x86
FX3U-64MT/ES-A	64	32	32, Transistor(Sink)	220V AC	45	1.00	220x90x86
FX3U-64MT/DSS	64	32	32, Transistor(Source)	24V DC	40	1.00	220x90x86
FX3U-64MT/ESS	64	32	32, Transistor(Source)	220V AC	45	1.00	220x90x86
5.Main Units với 80 ngõ vào ra I/O							
FX3U-80MR/DS	80	40	40, Relay	24V DC	45	1.2	285x90x86

FX3U-80MR/ES-A	80	40	40, Relay	220V AC	50	1.2	285x90x86
FX3U-80MT/DS	80	40	40, Transistor(Sink)	24V DC	45	1.2	285x90x86
FX3U-80MT/ES-A	80	40	40, Transistor(Sink)	220V AC	50	1.2	285x90x86
FX3U-80MT/DSS	80	40	40, Transistor(Source)	24V DC	45	1.2	285x90x86
FX3U-80MT/ESS	80	40	40, Transistor(Source)	220V AC	50	1.2	285x90x86
Mã	I/O	Input	Output	Nguồn	Công	Trong	Kích
	tối đa	(số cổng)	(số cổng, loại)	cấp	suất (W)	lượng (kg)	thước (mm)
6.Main Units với 128 ngõ vào ra I/O							

3. Sơ đồ Nguồn



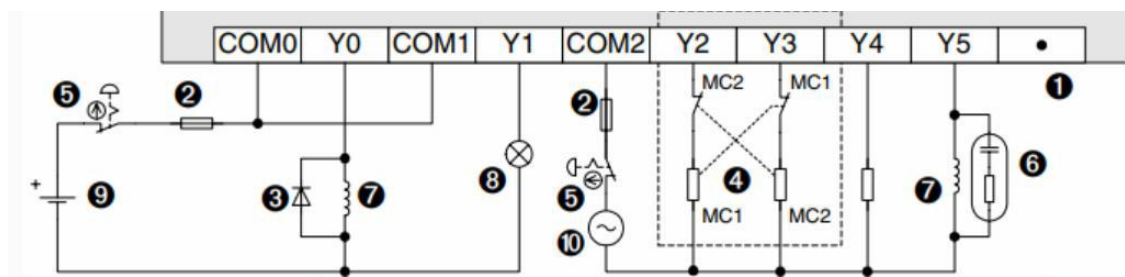
4. Sơ đồ Input Output

a. Ngõ ra Relay

- Đặc điểm của ngõ ra Relay là có thể sử dụng được cả điện áp 1 chiều ($\leq 30\text{VDC}$) và điện áp xoay chiều ($\leq 240\text{VAC}$) với dòng điện định mức lên tới 2A.

Tuy nhiên, nhược điểm của loại ngõ ra này là tần số đóng cắt nhỏ, cỡ 10ms.

- Sơ đồ đấu nối:



Trên ngõ ra PLC sẽ chia ra các cổng COM tương ứng với một hoặc một nhóm ngõ ra dùng chung mức điện áp.

Khi viết chương trình PLC, ví dụ chuyển Y0 lên mức tích cực thì tiếp điểm Rơ-le giữa COM0 và Y0 sẽ đóng, hoàn toàn tương tự với các cặp tiếp điểm khác như COM1-Y1, COM2-Y2, COM2-Y3, COM2-Y4, COM2-Y5... cũng tương tự. Khi đó với việc có nguồn nuôi bên ngoài, mạch điện sẽ trở thành một mạch điện khép kín và có dòng điện chạy qua tải.

Ví dụ đối với sơ đồ trên hình:

+ COM0 và COM1 đấu chung với chân dương của nguồn điện 1 chiều, đầu ra Y0 và Y1 đấu vào chân dương của tải.

+ COM 2 đấu với nguồn xoay chiều, các đầu ra Y2 Y3 Y4 Y5 đấu với tải xoay chiều.

b. Ngõ ra Transistor

- Đặc điểm của loại ngõ ra Transistor đóng ngắt bằng linh kiện bán dẫn nên có tốc độ đóng ngắt nhanh, có thể dùng để phát xung tốc độ cao lên tới 100Khz, 200Khz, 500Khz.

- Ưu điểm nữa của ngõ ra Transistor là nó chịu được số lần đóng ngắt lớn, tuổi thọ tính theo số lần đóng ngắt thì lớn hơn ngõ ra Rơ-le rất nhiều.

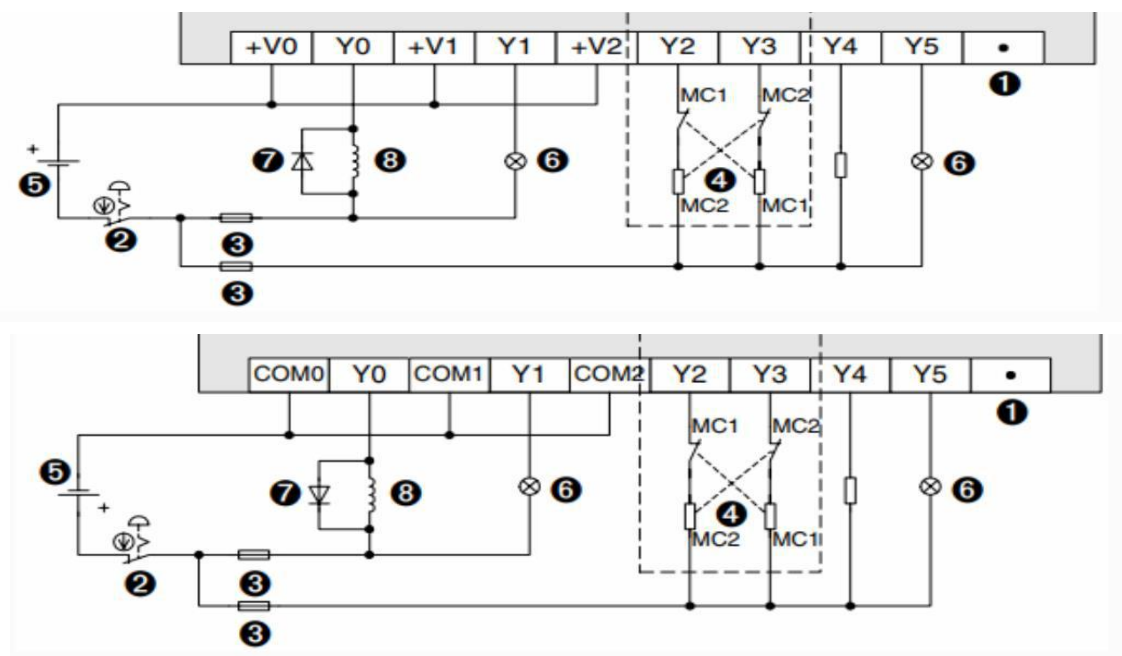
Tuy nhiên điểm hạn chế của PLC Mitsubishi ngõ ra transistor là không sử dụng được điện áp xoay chiều và dòng qua nó chịu giới hạn ở dưới 0.5A nên một số ứng dụng đóng ngắt thông thường khách hàng phải sử dụng thêm Rơ-le trung gian.

- PLC Mitsubishi ngõ ra Transistor còn có 2 loại là loại ngõ ra Sink và Source.

Đối với loại ngõ ra Sink, chân COM0 COM1 COM2... luôn đấu với chân 0V của nguồn 1 chiều.

Đối với loại ngõ ra Source, chân +V0 +V1 +V2... luôn đấu với chân dương của nguồn 1 chiều.

Khi các đầu ra lên mức tích cực, các chân đầu ra PLC sẽ nối với các chân COM hoặc +V tương ứng. Khi đó với việc có nguồn nuôi bên ngoài, mạch điện sẽ trở thành một mạch điện khép kín và có dòng điện chạy qua tải như hình bên dưới.



Chương 2 : Giới thiệu phần mềm - cách khai báo 1 Project

I. Phần mềm lập trình PLC GX-Works2

GX Works2 là phần mềm cấu hình và lập trình thế hệ kế tiếp cho điều khiển FX và Q Series.

GX Works2 cho phép nhà phát triển có thể “trộn” và kết hợp từ năm ngôn ngữ lập trình khác nhau, phù hợp với các phong cách lập trình khác nhau. Nhà phát triển thoải mái thoải mái lựa chọn ngôn ngữ để phát triển cho phù hợp với công việc.

Môi trường này tuân theo tiêu chuẩn IEC1131-3, cũng cho phép các bộ phận của dự án được lưu trong thư viện để sử dụng trong các ứng dụng trong tương lai. Điều đó có nghĩa mỗi một chức năng, chức năng chặn các chương trình, hoặc cấu trúc được viết, được thử nghiệm.

Cho phép lựa chọn công cụ và các phím tắt để tối ưu hóa khả năng trực giác của riêng người dùng.

Tích hợp mạng và các module chức năng đặc biệt giữ cho các tập tin dự án tổ chức và dễ dàng truy cập.

Được xây dựng với PLC ảo trên máy tính mô phỏng cho phép thực hiện chạy thử chương trình trên PLC ảo.

GX-Works2 hỗ trợ nhiều chế độ theo dõi, theo dõi chức năng, và khả năng gỡ lỗi trực tuyến cho phép kiểm soát được tình trạng của ứng dụng.



II. Sử dụng phần mềm và cách tạo một project

1. Sử dụng phần mềm

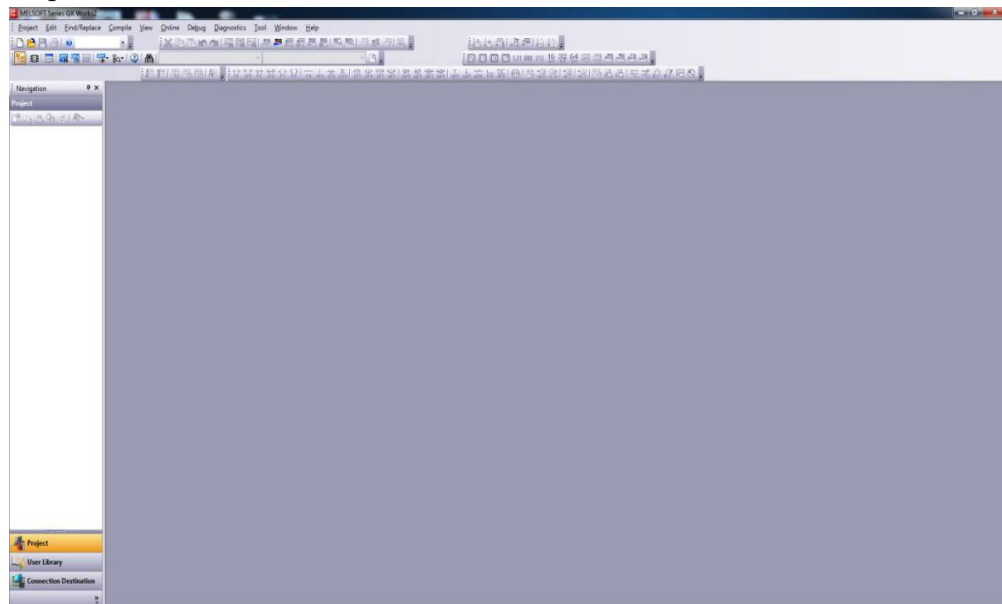
Để sử dụng phần mềm trước tiên ta click chuột vào biểu tượng GX Works2 trên màn hình desktop



Hoặc ta vào Start rồi chọn vào GX Works2 như hình vẽ để khởi động phần mềm



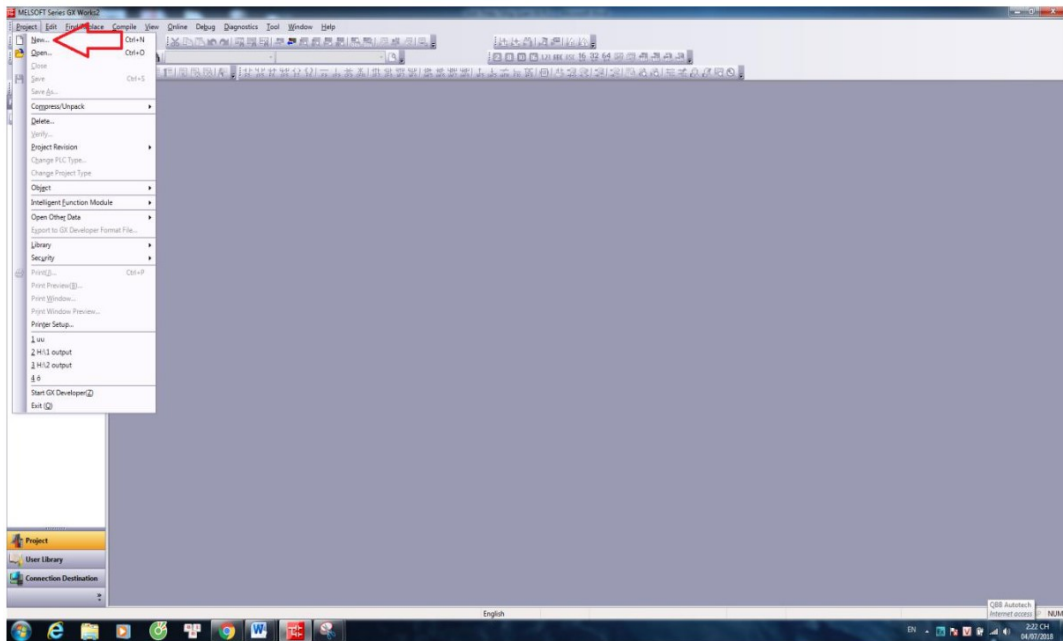
Sau khi khởi động phần mềm màn hình máy tính của bạn sẽ xuất hiện một giao diện làm việc của phần mềm như hình vẽ



Đây chính là giao diện làm việc của phần mềm GX Works2

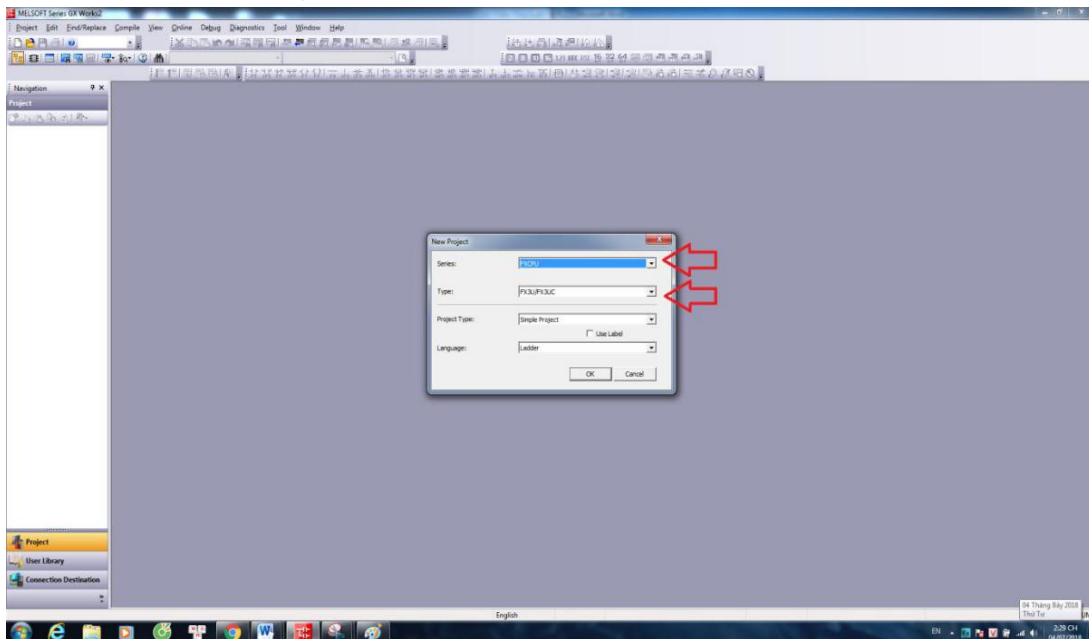
2. Cách tạo một project mới

Bước 1: Sau khi đã khởi động phần mềm ta click chuột vào Project sau đó chọn New hoặc click vào biểu tượng new để tạo một Project mới



Bước 2: Sau khi tạo xong một project mới trên màn hình sẽ xuất hiện một cửa sổ, ở đây là mục người dùng chọn đúng tên đúng dòng PLC đang lập trình, nếu trường hợp chọn sai với PLC thực tế bên ngoài thì lúc Download chương trình PLC sẽ báo lỗi.

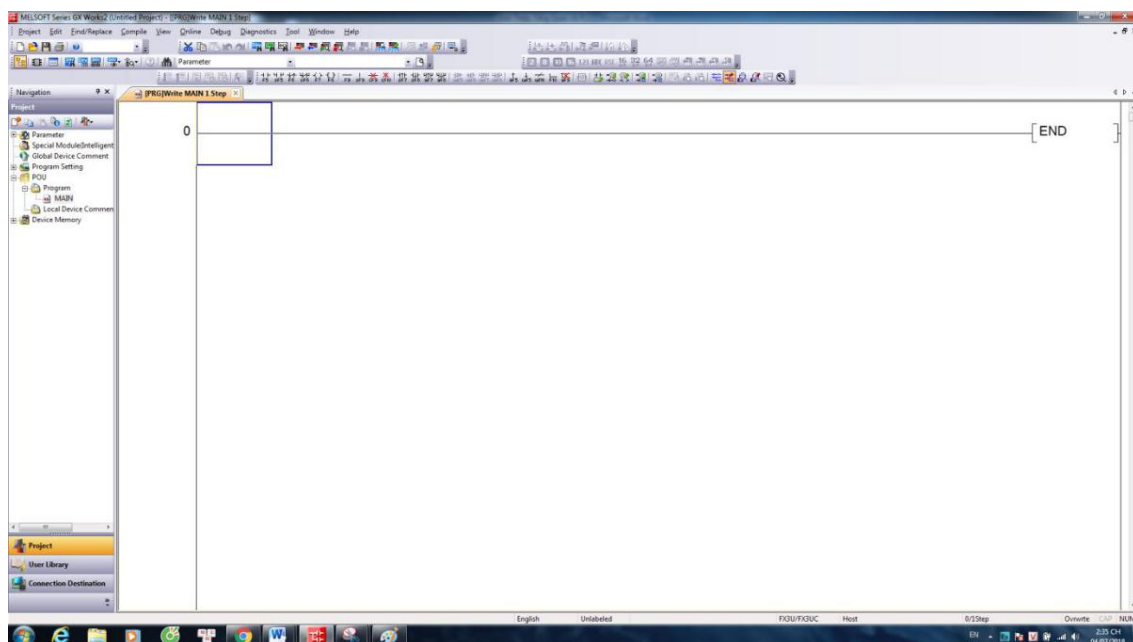
Ví dụ : ở đây ta chọn dòng CPU là FX và tên CPU là FX3U/FX3UC như hình ảnh



ở trường hợp này chương trình của bạn đều có thể Download xuống được cho FX3U và FX3UC

Bước 4: Sau khi đã chọn đúng tên và dòng CPU cần lập trình ta nhấn OK để hoàn tất quá trình tạo project mới

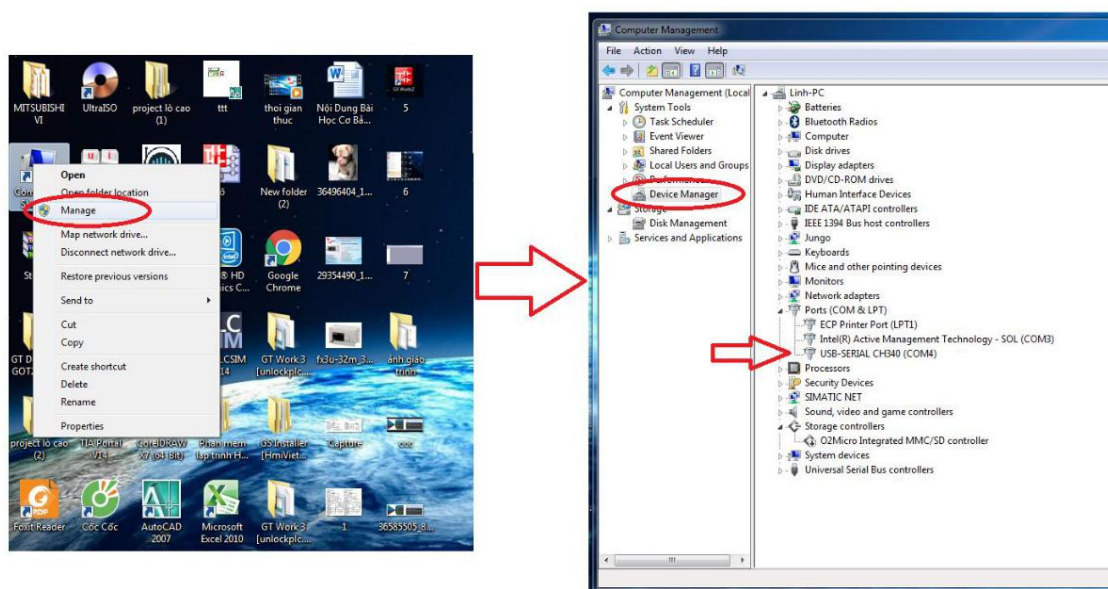
Lúc này trên màn hình sẽ xuất hiện một giao diện đây chính là giao diện viết chương trình cho PLC



III. Cách download chương trình xuống PLC

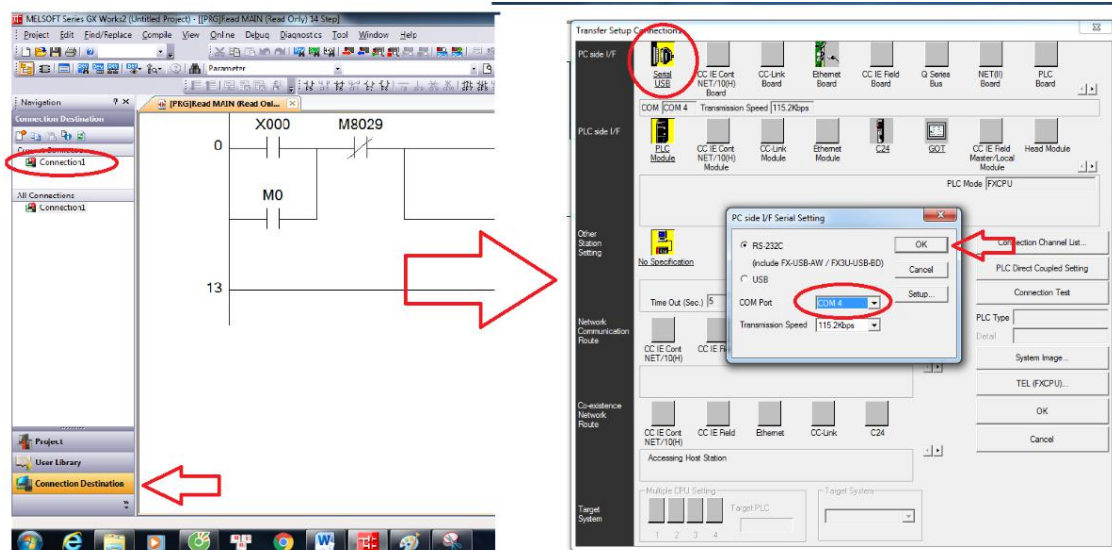
1. Download chương trình xuống CPU

Trước tiên để download được một chương trình người dùng phải cài Driver cho cáp nạp của mình và xem vị trí cáp đang nằm COM bao nhiêu bằng cách click chuột phải vào biểu tượng Manage như hình ảnh



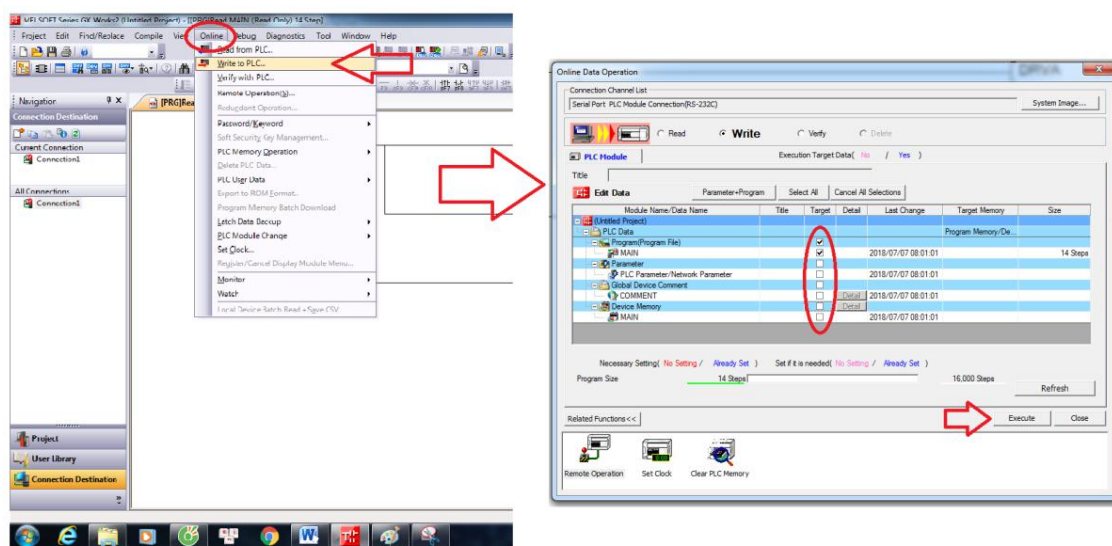
Ví dụ trường hợp ở đây cáp người dùng nằm ở vị trí COM4 ta lại tiến hành cài đặt cổng COM trên phần mềm GX Works2 như sau

Ta kích chuột vào **Connection Destination** rồi tiếp tục chọn **Connection1** sau đó chọn vào Serial USB để chọn vị trí cáp

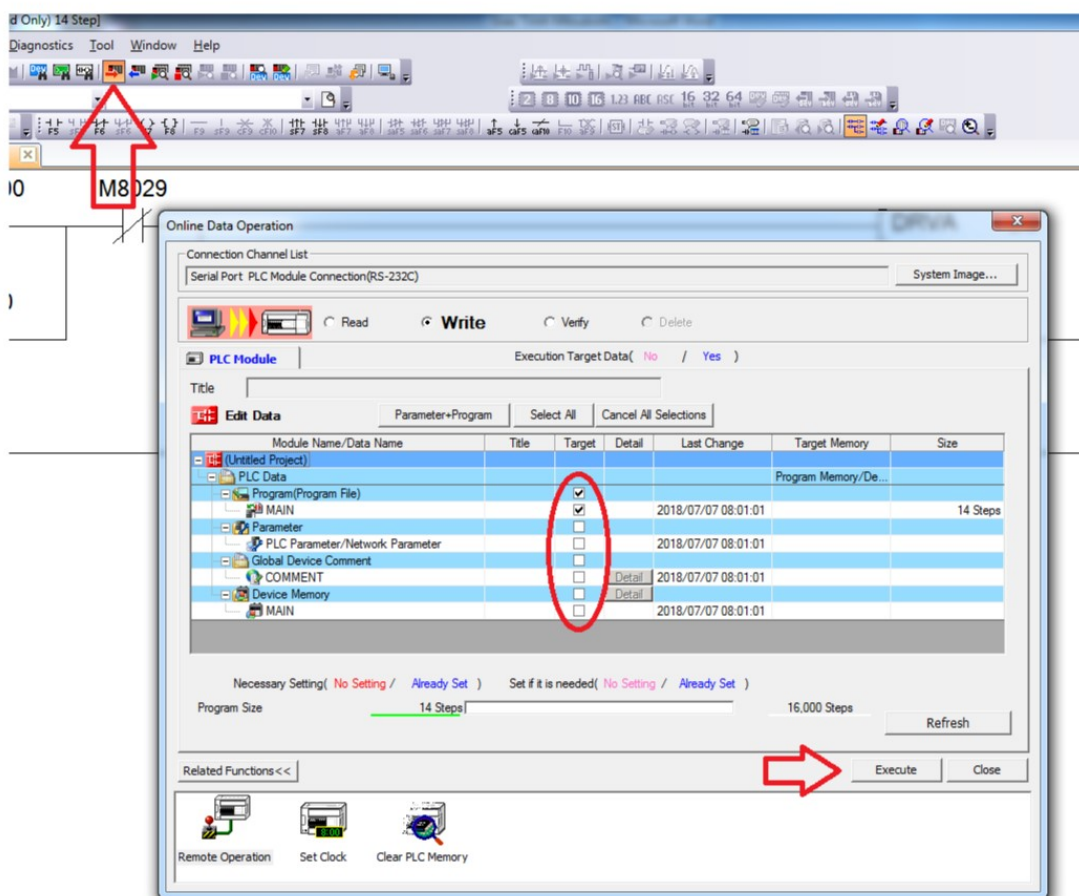


Sau khi chọn xong vị trí cáp ta tiến hành down chương trình xuống CPU bằng hai cách sau

Cách 1: Ta kích chuột vào biểu tượng Online chọn Write to PLC trên màn hình sẽ xuất hiện một bảng ở đây phần Target ta chọn những mục cần down xuống rồi nhấn Execute để down chương trình xuống CPU



Cách 2: Trên màn hình ta kích chuột vào biểu tượng Write to PLC trên thanh công cụ sau đó làm tương tự như cách trên



Chương 3 : Các vùng nhớ và tập lệnh cơ bản

1. Các vùng nhớ

PLC Mitsubishi có 3 loại vùng nhớ chính: vùng nhớ chương trình EEPROM, vùng nhớ Bit và vùng nhớ dữ liệu.

Vùng nhớ chương trình (EEPROM): Vùng nhớ chứa chương trình chính của PLC, tham số thiết lập, các chú thích và thanh ghi File

Vùng nhớ bit: Bao gồm vùng nhớ đầu vào số (X), Vùng nhớ đầu ra số Y, Vùng nhớ trung gian(M), Vùng nhớ trạng thái (S), Vùng nhớ tiếp điểm Timer (T), Vùng nhớ tiếp điểm Counter (C). Cụ thể:

+ Vùng nhớ đầu vào (ký hiệu là X) chứa các ô nhớ lưu dữ liệu mức logic ở đầu vào vật lý của PLC và có kiểu dữ liệu là kiểu Bit (0 hoặc 1)

+ Vùng nhớ đầu ra (ký hiệu là Y) chứa các ô nhớ lưu dữ liệu mức logic mà PLC đưa ra ở đầu ra vật lý và có kiểu dữ liệu là kiểu Bit (0 hoặc 1)

+ Vùng nhớ trung gian (ký hiệu là M) chứa các ô nhớ trung gian, các ô nhớ này có thể sử dụng ở các vị trí mà người lập trình cần sử dụng biến nhớ/biến trung gian trong chương trình.

+ Vùng nhớ trạng thái (ký hiệu là S) chứa các ô nhớ trạng thái, thường được sử dụng trong chương trình sử dụng ngôn ngữ instruction list hoặc SFC, nhằm thể hiện trạng thái chương trình.

+ Timer (T) có tác dụng định thời (đếm xung đồng hồ có chu kỳ 1ms, 10ms, 100ms,...). Các tín hiệu ở dạng tiếp điểm của Timer (Bit chờ báo Timer đã đếm đủ, Bit reset timer,...) sẽ được lưu trong vùng nhớ Timer, còn giá trị hiện thời của Timer sẽ được lưu trong vùng nhớ dữ liệu.

+ Counter (C) có tác dụng làm bộ đếm (đếm các sự kiện). Các tín hiệu ở dạng tiếp điểm của Counter (Bit chờ báo Counter đã đếm đủ, Bit reset counter,...) sẽ được lưu trong vùng nhớ Counter, còn giá trị hiện thời của Counter sẽ được lưu trong vùng nhớ dữ liệu.

Vùng nhớ dữ liệu: Chứa các thanh ghi dữ liệu (D), giá trị tức thời của Timer (T), giá trị tức thời của Counter (C) và các thanh ghi chỉ số (V, Z)

+ Thanh ghi được ký hiệu D và đánh số thập phân.


Ví dụ: D0, D9, D128.

Thanh ghi rất quan trọng khi xử lý dữ liệu số được thập phân bên ngoài.

Ví dụ: dữ liệu từ các công tắc chọn nhấn, bộ chuyển đổi A/D, D/A.

2. Các tập lệnh cơ bản

a. Lệnh LD (load)

Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
LD (load)	Có nhiệm vụ khởi tạo loại công tắc NO, nối trực tiếp đầu bên trái của mạch		X,Y,M,S,T,C

Ví dụ :

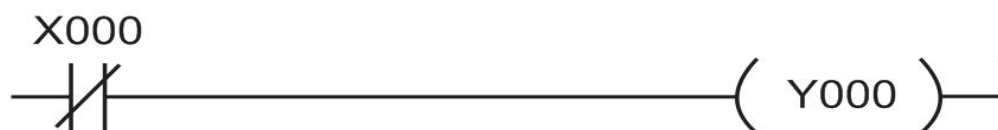


Ngõ ra Y000 có điện khi ngõ vào X000 đóng hay ngõ vào X000=1

b. Lệnh LDI (load Inverse)

Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
LDI (load Inverse)	Có nhiệm vụ khởi tạo loại công tắc NC, nối trực tiếp đầu bên trái của mạch		X,Y,M,S,T,C

Ví dụ :



Ngõ ra Y000 luôn có điện và sẽ mất điện khi ngõ vào X000 được tác động hay ngõ vào X000=1

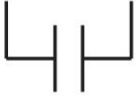
c. Lệnh out

Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
Out	<ul style="list-style-type: none"> - Điều khiển cuộn dây - Nhiều lệnh out có thể được nối song song với nhau - Không thể điều khiển thiết bị ngõ vào loại (X) 		Y,M,S,T,C

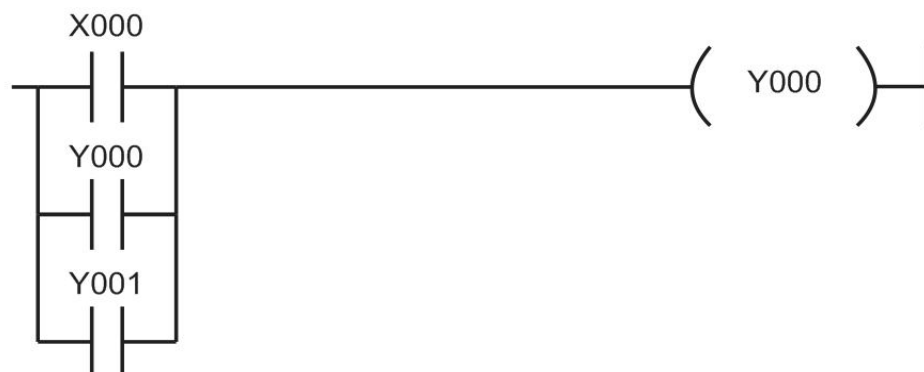
Ví dụ :



Ngõ ra Y000 = ON khi công tắc logic thường đóng X000 đóng (X000 = 0); Ngõ ra Y000 = OFF khi công tắc logic thường đóng X000 hở (X000 = ON) d. Lệnh AND


Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
OR	Nối song song các công tắc NO Tối đa là 10 nhánh nối song song cho một cuộn dây		X,Y,M,S,T,C

Ví dụ :



Khi ngõ vào X000 = ON thì ngõ ra Y000 sẽ ON và đồng thời tiếp điểm thường hở Y000 = 1 và sẽ lưu giá trị ON cho ngõ ra kể cả khi ngõ vào X000 = OFF

e. Lệnh PLS (sườn lên)

Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
PLS	Tiếp điểm phát hiện sườn lên. Khi đầu vào IN chuyển từ 0-1 thì bit trạng thái được set lên 1		X,Y,M,S,T,C

Ví dụ:



Khi X000 = 1 thì Lệnh sườn lên sẽ bắt cạnh xung sườn lên và đưa ra ngõ ra 1 xung duy nhất khi ngõ ra Y000 = 1 tiếp điểm thường hở Y000 sẽ =1 và ngõ ra được duy trì.

f. Lệnh PLF (sườn xuống)

Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
PLF	Tiếp điểm phát hiện sườn xuống. Khi đầu vào IN chuyển từ 0-1 thì bit trạng thái được set lên 1		X,Y,M,S,T,C

Ví dụ:

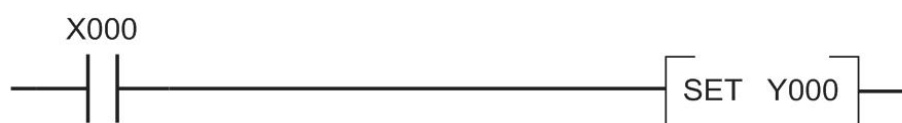


Khi X000 = 1 rồi X000 = 0 thì Lệnh sườn xuống sẽ bắt cạnh xung sườn xuống và đưa ra ngõ ra 1 xung duy nhất khi ngõ ra Y000 = 1 tiếp điểm thường hở Y000 sẽ = 1 và ngõ ra được duy trì

g. Lệnh sét (S)

Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
S	Ghi giá trị tại địa chỉ đầu ra = 1 (OUT = 1)		Y,M,S,T,C

Ví dụ :

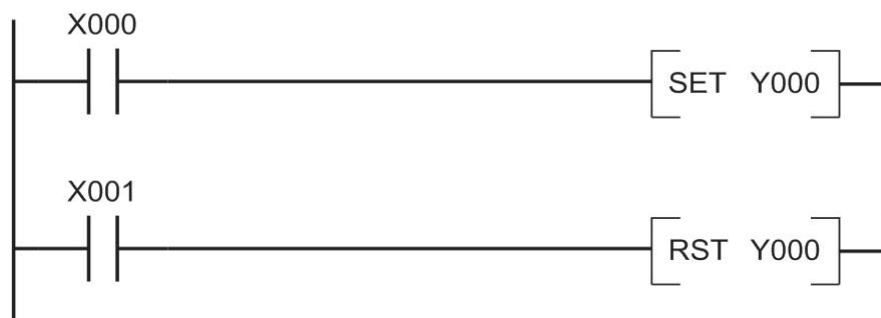


Khi ngõ vào = 1 thì ngõ ra Y000 sẽ luôn luôn = 1

h. Lệnh Reset (RST)

Tập Lệnh	Chức Năng	Dạng Mẫu	Thiết Bị
RST	Ghi giá trị tại địa chỉ đầu ra = 0 (OUT = 0)		Y,M,S,T,C

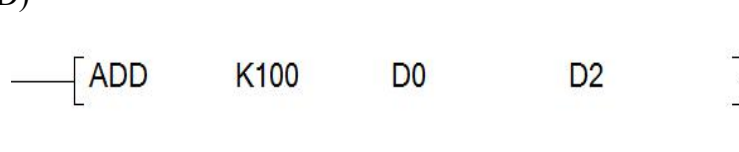
Ví dụ:



Khi ngõ vào X000=1 thì ngay lập tức ngõ ra Y000 sẽ luôn sét ON khi ngõ vào X001=1 thì Ngõ ra Y000 sẽ bị reset OFF

3. Một số lệnh toán học và lệnh di chuyển

a. Phép cộng (ADD)



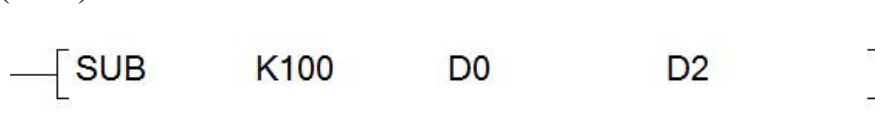
Lệnh ADD là lệnh thực hiện phép cộng với 2 giá trị cho trước và tổng sẽ được ghi vào một vùng nhớ.

Ở đây cấu trúc lệnh như sau : ADD là phép cộng K100 là hằng số đặt trước dạng hằng số.

D0 là hằng số cho trước trong vùng nhớ

D2 vùng nhớ để lưu giá trị tổng của phép cộng

b. Phép trừ (SUB)



Lệnh SUB là lệnh thực hiện phép trừ với 2 giá trị cho trước và hiệu sẽ được ghi vào một vùng nhớ của người dùng đặt trước.

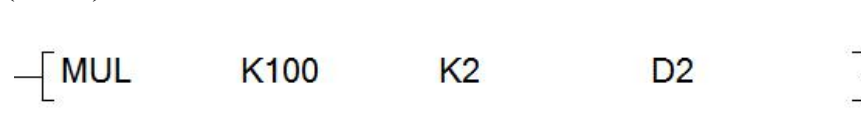
ở đây cấu trúc lệnh như sau : SUB là phép trừ

K100 là hằng số đặt trước dạng hằng số

D0 là hằng số cho trước trong vùng nhớ

D2 vùng nhớ để lưu giá trị hiệu của phép trừ

c. Phép nhân (MUL)



Lệnh MUL là lệnh thực hiện phép Nhân với 2 giá trị cho trước và hiệu sẽ được ghi vào một vùng nhớ của người dùng đặt trước.

ở đây cấu trúc lệnh như sau : MUL là phép Nhân

K100 là hằng số thứ nhất đặt trước dạng hằng số

K2 là hằng số thứ 2 đặt trước

D2 vùng nhớ để lưu giá trị thương của phép nhân

d. Phép chia (DIV)

```

      —[ DIV      K100      K2      D2      ]

```

Lệnh DIV là lệnh thực hiện phép Chia với 2 giá trị cho trước và hiệu sẽ được ghi vào một vùng nhớ của người dùng đặt trước.

ở đây cấu trúc lệnh như sau : MUL là phép Chia

K100 là hằng số thứ nhất đặt trước dạng hằng số

K2 là hằng số thứ 2 đặt trước

D2 vùng nhớ để lưu giá trị tích của phép chia

e.Lệnh di chuyển (MOV)

Lệnh MOV là lệnh di chuyển giá trị của 1 thanh ghi hay di chuyển 1 giá trị hằng số vào một vùng nhớ thanh ghi khác với cấu trúc lệnh như sau

```

      —————[ MOV      K100      D1      ]
      —————[ MOV      D0      D10     ]

```

Trong đó MOV là lệnh di chuyển

K100, D0 là giá trị đặt trước cần di chuyển

D1, D10 là thanh ghi lưu giữ giá trị di chuyển

Chương 4: Lệnh TIMER và lệnh COUNTER

1. Lệnh Timer

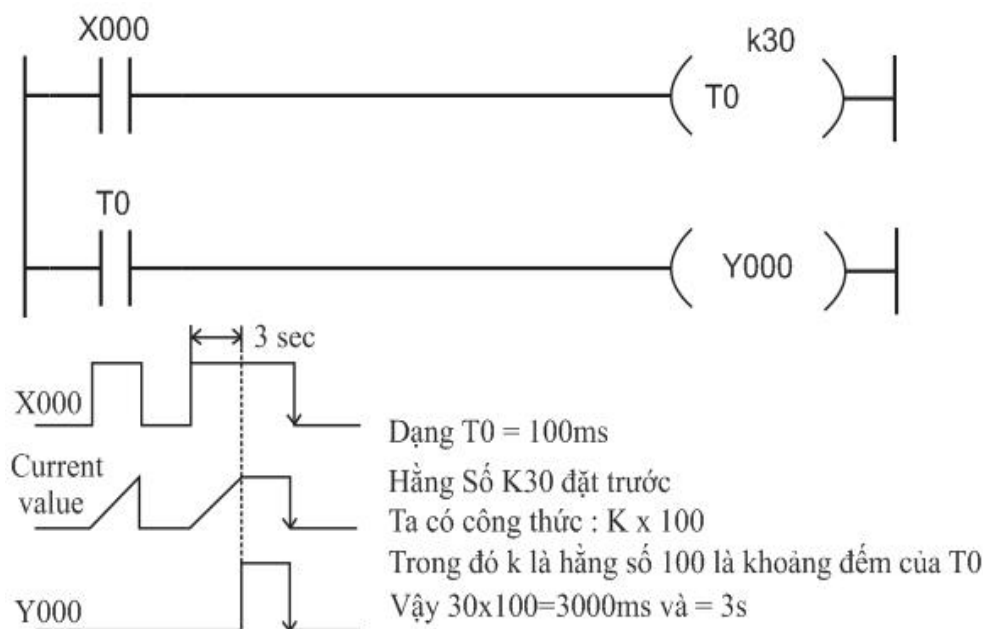
Các timer cộng và đếm các xung clock 1ms, 10ms, 100ms,... trong PLC.

Khi giá trị đếm được đạt được giá trị cài đặt, tiếp điểm ngõ ra của timer bật ON.

Giá trị cài đặt có thể được xác định trực tiếp bằng hằng số (K) trong bộ nhớ chương trình hoặc gián tiếp bởi nội dung của thanh ghi dữ liệu (D).

100ms 0.1 to 3276,7 giây	10 ms 0.01 to 327.67giây	Loại khả nhớ cho các xung 1ms*1 0.001 to 32.767 giây	Loại khả nhớ 100ms *1 0.1 to 3276.7 giây	Loại khả nhớ 100ms *1 0.1 to 3276.7 giây
T0 đến T199 (200 bộ)	T200 đến T245 (46 bộ)	T246 đến T249 (4 bộ) Loại được chốt	T250 đến T255 (6 bộ) Loại được chốt	T256 đến T511 (256 bộ)

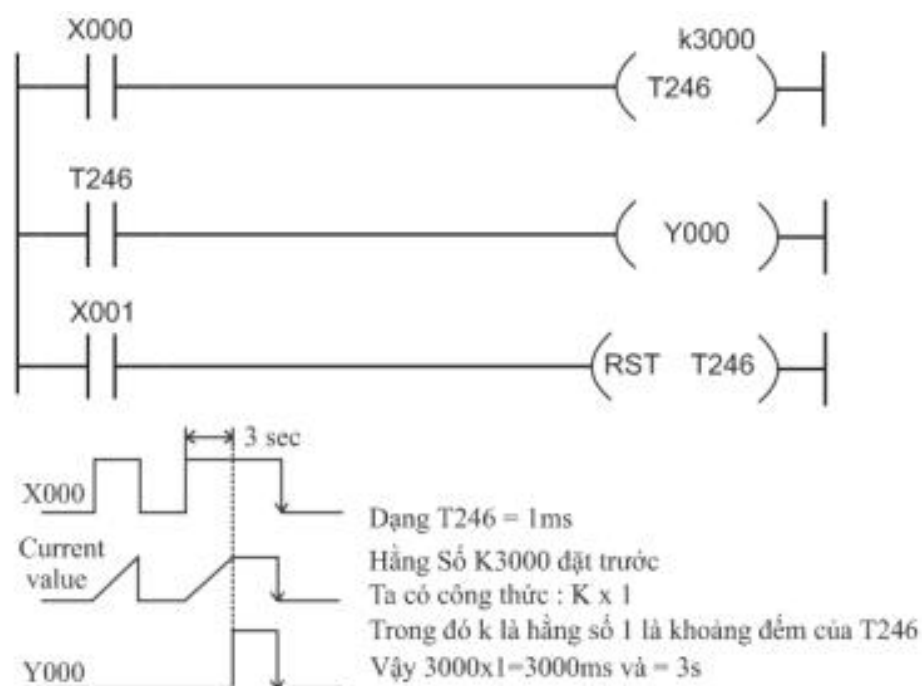
Ví dụ:



Khi ngõ vào điều khiển X000 của cuộn dây timer T0 bật ON, bộ đếm giá trị hiện hành của T0 cộng và đếm các xung clock 100ms. Khi giá trị đếm được bằng với giá trị cài đặt là K30, tiếp điểm ngõ ra của timer bật ON.

Nói một cách khác, tiếp điểm ngõ ra bật ON sau 3s khi cuộn dây được điều khiển. Khi ngõ vào điều khiển X000 là OFF hoặc khi nguồn điện OFF, timer được reset và các tiếp điểm ngõ ra trở về trạng thái ban đầu.

Ví dụ: Loại time có chốt



Khi ngõ vào điều khiển X000 của cuộn dây timer T246 bật ON, bộ đếm giá trị hiện hành của T246 cộng và đếm các xung clock 1ms. Khi giá trị đếm được bằng với giá trị cài đặt là K3000, tiếp điểm ngõ ra của timer bật ON.

Nói một cách khác, tiếp điểm ngõ ra bật ON sau 3s khi cuộn dây được điều khiển. Khi ngõ vào điều khiển X000 là OFF hoặc khi nguồn điện OFF, timer vẫn luôn được set ON khi ngõ vào X001 là ON thì lập tức lệnh Reset (RST) sẽ reset T246 về =0 lúc này Time sẽ về trạng thái ban đầu và ngõ ra Y000 OFF.

2. Lệnh COUNTER

Counter là bộ đếm thực hiện chức năng đếm sườn (cạnh) xung của tín hiệu đầu vào

	Bộ đếm 16 bit Đếm từ 0 đến 32767		Bộ đếm hai chiều 32 bit Đếm từ -2,147,483,648 đến +2,147,483,647	
	Kiểu chung	Kiểu chốt (nguồn pin)	Kiểu chung	Kiểu chốt (nguồn pin)
PLC FX3U/ FX3UC	C0 to C99 100 điểm* ¹	C100toC199 100 điểm* ²	C200 to C219 20 điểm* ¹	C220 to C234 15 điểm* ²

+ Đối với kiểu chung: Vùng này không được chốt, nó có thể thay đổi tới một vùng được chốt bởi việc điều chỉnh những thông số.

+ Đối với kiểu chốt: Vùng này được chốt, nó có thể thay đổi tới một vùng không được chốt bởi việc điều chỉnh những thông số.

Bảng ở dưới cho thấy những đặc tính của bộ đếm 16 bit và 32 bit.

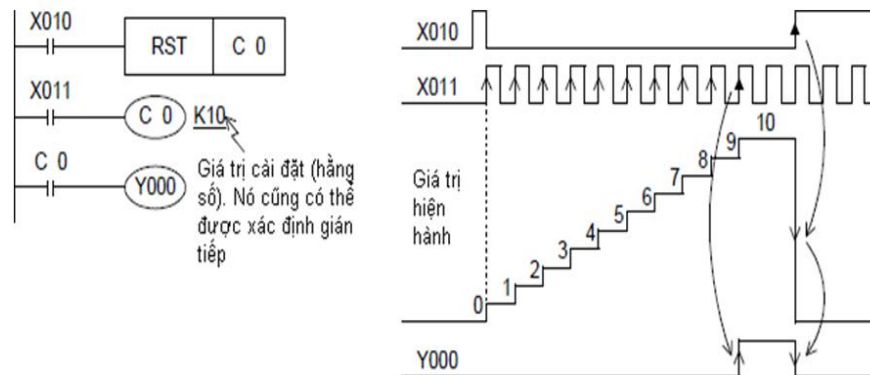
Mục	Bộ đếm 16-bit	Bộ đếm 32-bit
Hướng đếm (chiều đếm)	Đếm lên	Đếm lên và đếm xuống có thể được chuyển
Giá trị cài đặt	Từ 1 đến 32767	Từ -2,147,483,648 đến +2,147,483,647
Xác định giá trị cài đặt	Hằng số (K) hay thanh ghi dữ liệu	Hằng số (K) hay một cặp thanh ghi dữ liệu
Thay đổi giá trị hiện hành	Không thay đổi sau khi đếm hết	Thay đổi ngay cả sau khi đếm hết (bộ đếm vòng)
Tiếp điểm ngõ ra	Chốt sau khi đếm hết	Chốt (ở chiều đếm lên) hoặc reset (ở chiều đếm xuống)
Hoạt động reset	Khi lệnh RST được thực thi, giá trị hiện hành của bộ đếm được reset về 0 và tiếp điểm ngõ ra trở về trạng thái đầu	
Thanh ghi giá trị hiện hành	16 bit	32 bit

Khi một relay phụ trợ cho việc chuyển hướng đếm được đặt ON, bộ đếm thực hiện đếm xuống và khi đặt OFF, bộ đếm thực hiện đếm lên.

Bộ đếm	Relay chuyển hướng đếm	Bộ đếm	Relay chuyển hướng đếm	Bộ đếm	Relay chuyển hướng đếm	Bộ đếm	Relay chuyển hướng đếm
C200	M8200	C210	M8210	C220	M8220	C230	M8230
C201	M8201	C211	M8211	C221	M8221	C231	M8231
C202	M8202	C212	M8212	C222	M8222	C232	M8232
C203	M8203	C213	M8213	C223	M8223	C233	M8233
C204	M8204	C214	M8214	C224	M8224	C234	M8234
C205	M8205	C215	M8215	C225	M8225		
C206	M8206	C216	M8216	C226	M8226		
C207	M8207	C217	M8217	C227	M8227		
C208	M8208	C218	M8218	C228	M8228		
C209	M8209	C219	M8219	C229	M8229		

- Giá trị cài đặt hợp lệ của bộ đếm lên 16 bit là từ K1 đến K32767 (hằng số).
- Trong các bộ đếm kiểu chung, giá trị đếm được xóa khi PLC bật OFF. Tuy nhiên trong những bộ đếm được chốt, giá trị đếm ngay trước khi mất nguồn được lưu lại. Giá trị bộ đếm trong các hoạt động tiếp theo sau có thể được cộng với giá trị bộ đếm trước đó.
- Mỗi lần cuộn dây C0 được điều khiển bằng ngõ vào đếm X011, giá trị hiện hành của bộ đếm tăng lên. Khi lệnh cuộn dây được thực thi 10 lần, tiếp điểm ngõ ra bật ON. Sau đó, giá trị hiện hành của bộ đếm không thay đổi ngay cả khi ngõ vào

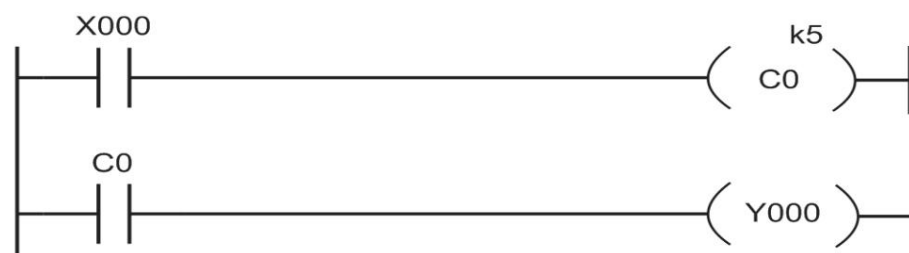
đếm X011 bật ON. Khi ngõ vào reset X010 bật ON và sau đó lệnh RST được thực thi, giá trị hiện hành của bộ đếm được reset về 0 và tiếp điểm ngõ ra quay trở về trạng thái đầu.



- Giá trị cài đặt bộ đếm có thể được thiết lập bởi 1 hằng số (K) như trên hay được thiết lập một cách gián tiếp bằng thanh ghi dữ liệu. Ví dụ, khi D10 được xác định và nội dung của D10 là 123, nó tương đương với K123.
- Nếu dữ liệu vượt quá giá trị cài đặt được ghi đến thanh ghi giá trị hiện hành bởi lệnh MOV,..., cuộn dây OUT bật ON và thanh ghi giá trị hiện hành trở thành giá trị cài đặt khi ngõ vào đếm kế tiếp được nhận.
- Trong một bộ đếm được chốt, giá trị hiện hành, hoạt động của tiếp điểm ngõ ra và trạng thái reset của bộ đếm được chốt bởi nguồn pin trong PLC.

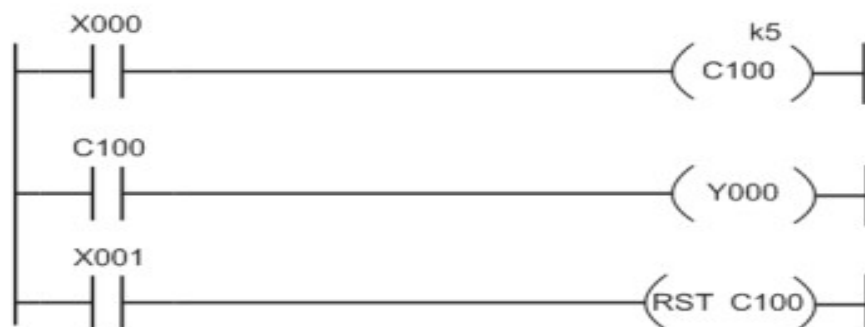
Các ví dụ

a. VÍ DỤ: kiểu chung từ C0 đến C99



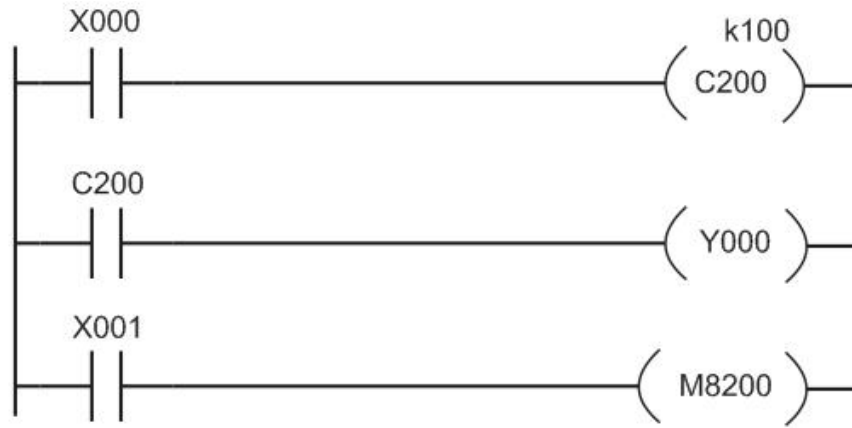
Khi mỗi lần ngõ vào X000 ON thì C0 sẽ nhận 1 xung và C0 sẽ cộng thêm một đơn vị cho đến khi C0 = 5 thì ngõ ra Y000 sẽ ON với trường hợp PLC bị nhát nguồn thì counter sẽ về trạng thái 0 ban đầu

b. VÍ DỤ: kiểu chốt từ C100 đến C19



Khi mỗi lần ngõ vào X000 ON thì C0 sẽ nhận 1 xung và C100 sẽ công thêm một đơn vị cho đến khi $C100 = 5$ thì ngõ ra Y000 sẽ ON với trường hợp PLC bị nhất nguồn thì counter sẽ luôn giữ giá trị đếm được cho đến khi ngõ vào X001 ON thì C100 sẽ bị reset về 0

c. VÍ DỤ: kiểu chung có thể đếm lên và đếm xuống bắt đầu từ C200 đến C219



Khi mỗi lần ngõ vào X000 ON thì C0 sẽ nhận 1 xung và C200 sẽ công thêm một đơn vị nếu đầu vào X001 ON cộng với ngõ vào X000 ON thì C200 sẽ trừ 1 đơn vị khi nhận được cạnh xung cũng tương tự khi $C200 = 10$ thì ngõ ra Y000 sẽ ON với trường hợp PLC bị nhất nguồn thì counter sẽ về giá trị ban đầu là 0.

Chương 5 : Lập trình HMI

1. Phần mềm GT Designer 3

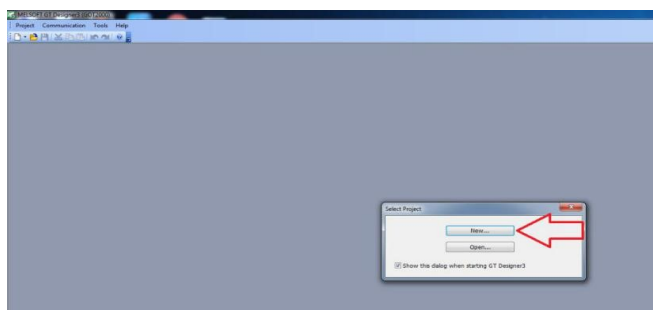
GT Designer là phần mềm hỗ trợ lập trình HMI với các dòng HMI của Mitsubishi ngoài việc Download chương trình xuống cho các màn hình thì GT Designer còn hỗ trợ cho việc mô phỏng trên máy tính thông qua phần mềm hỗ trợ Simunition để người dùng dễ dàng hơn trong việc tiếp cận các bài toán thực tế.

2. Cách tạo một project mới trên GT Designer

Bước 1: chúng ta khởi động phần mềm GT Designer lên như các phần mềm khác bằng cách click vào biểu tượng như sau

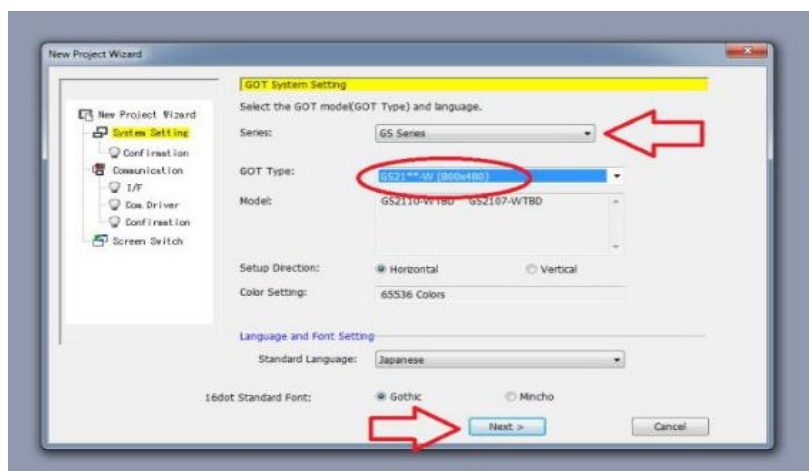


Bước 2: Sau khi khởi động phần mềm trên màn hình sẽ xuất hiện một cửa sổ ở đây ta chọn New để tiếp tục



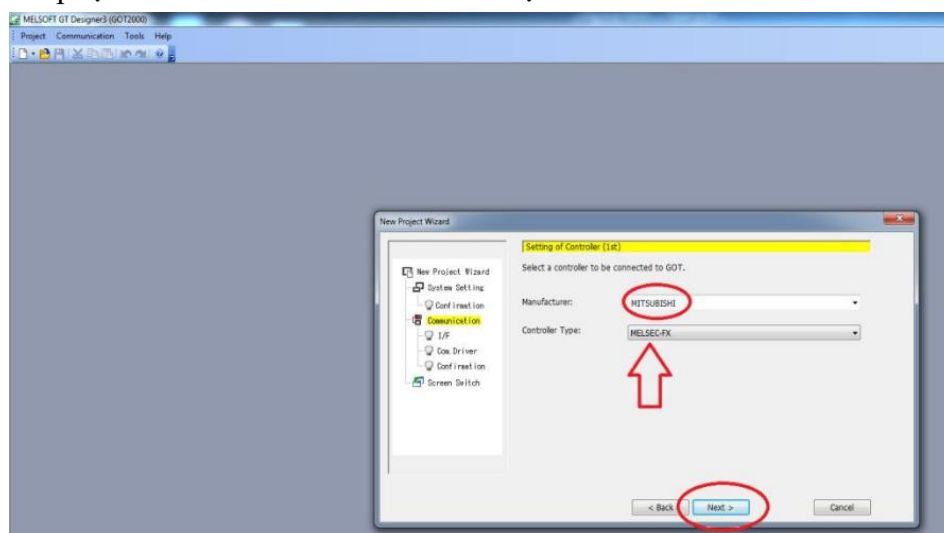
Sau khi chọn New ta tiếp tục click vào Next để tiếp tục

Bước 3 : Sau khi chọn Next trên màn hình xuất hiện 1 cửa sổ ở đây người dùng có thể chọn kiểu, mã màn hình mà mình muốn lập trình



Sau khi chọn xong ta nhấn Next để tiếp tục

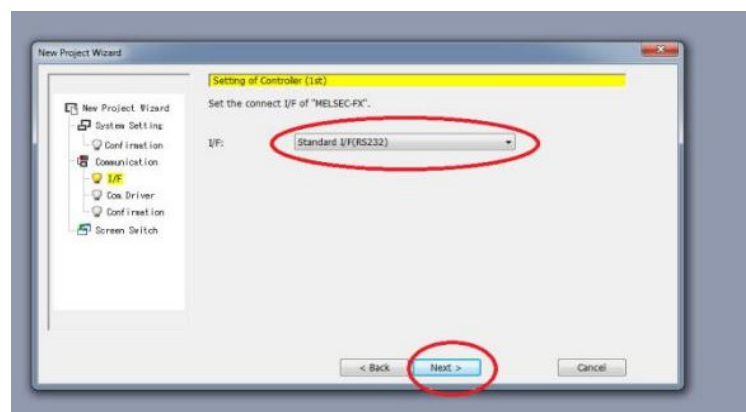
Bước 4 : Tiếp tục nhấn Next cho đến khi xuất hiện cửa sổ như hình vẽ



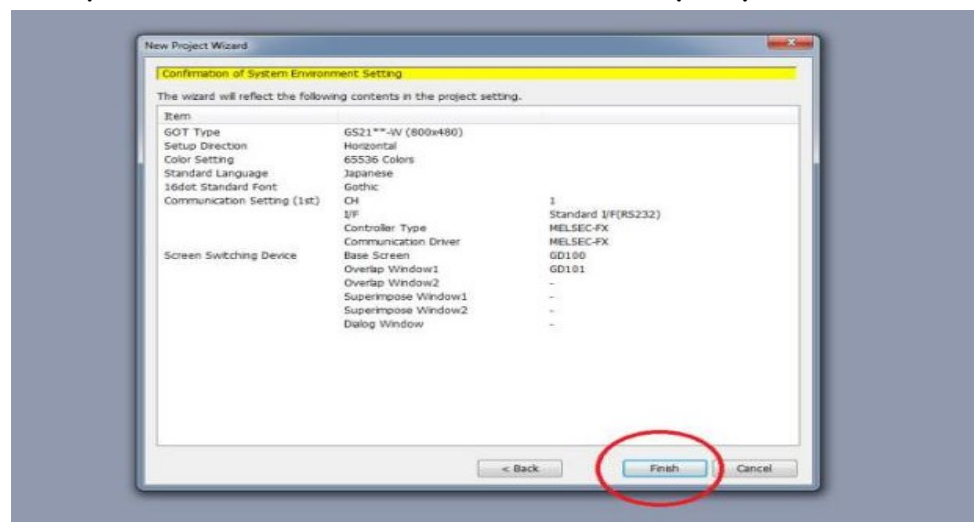
Ở đây người dùng chọn hãng PLC muốn kết nối và tên dòng PLC muốn kết nối ví dụ ở đây ta chọn dòng PLC là Mitsubishi và kiểu MELSEC-FX để kết nối sau đó ta tiếp tục chọn next để tiếp tục

Bước 5 : Sau khi nhấn Next trên màn hình lại tiếp tục xuất hiện 1 cửa sổ ở

đây người dùng chọn chế độ cáp kết nối phù hợp với HMI của mình với CPU



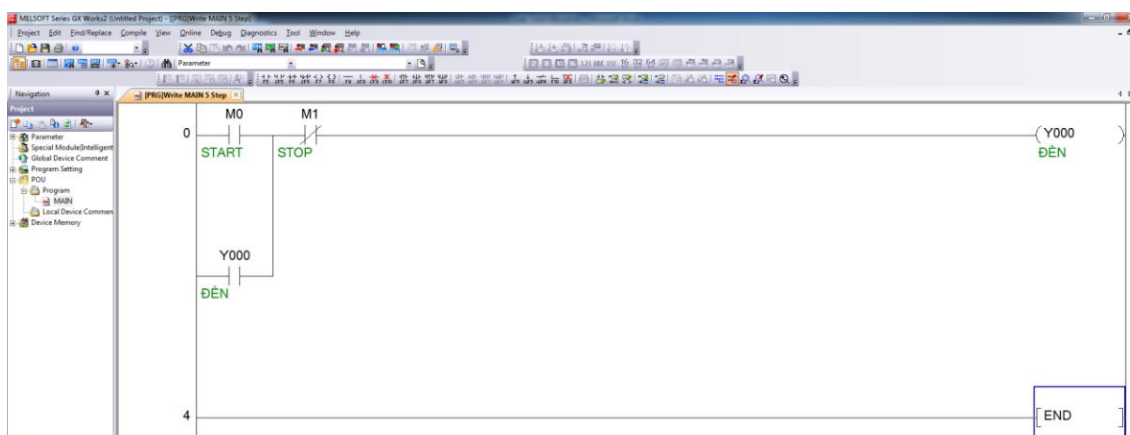
Sau đó ta chọn Next rồi nhấn Finish để hoàn thành bước tạo một màn hình mới



3. Cách gán Tag IN, OUT cho một chương trình đơn giản

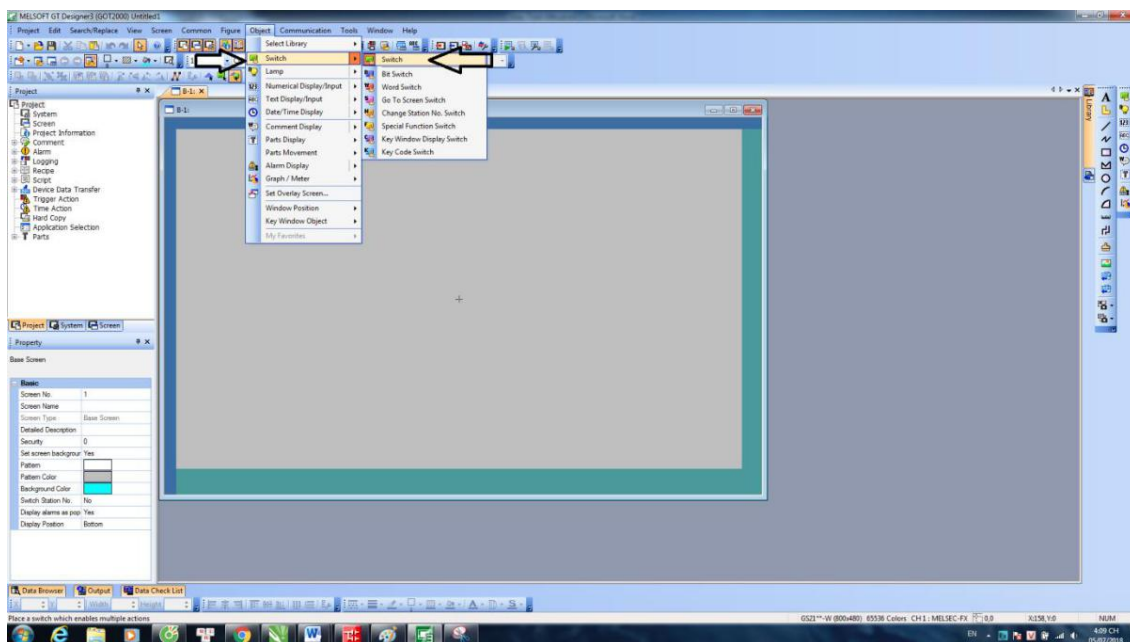
Trước tiên ta lập trình một bài toán ON/OFF đơn giản trên phần mềm GX Works2 như hình vẽ

Lưu ý: đối với lập trình HMI các biến đầu vào ta không nên sử dụng biến X mặc định thay đó ta sử dụng biến M để Tag cho HMI

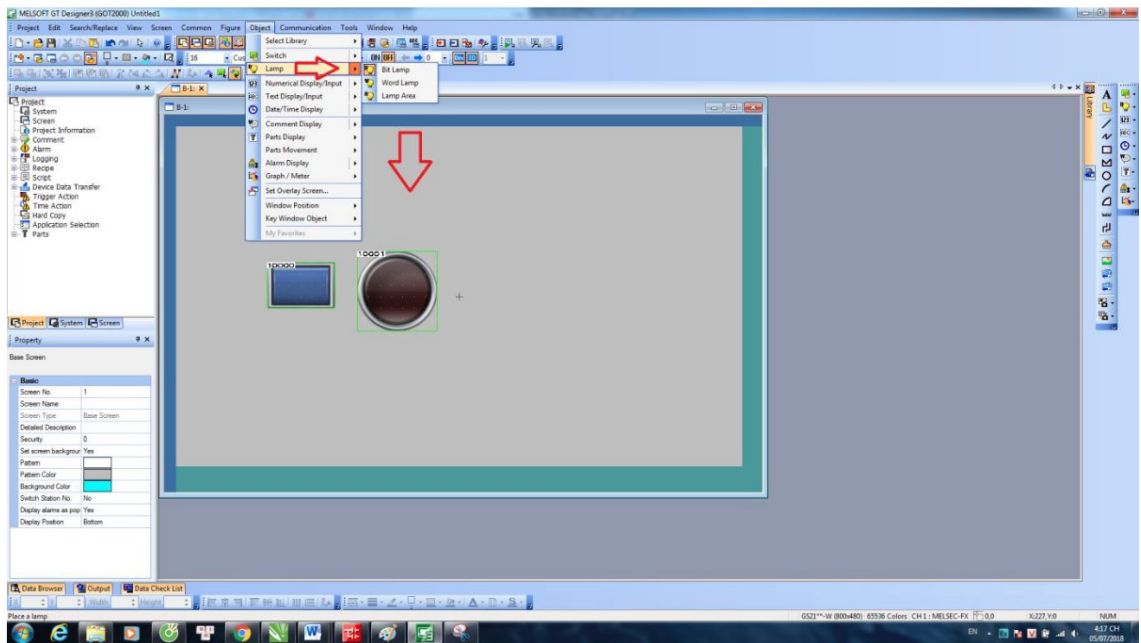


Sau khi lập trình một bài toán ON/OFF đơn giản ta tiếp tục trở lại với phần mềm GT Designer để tiến hành TAG địa chỉ IN/OUT cho bài toán.

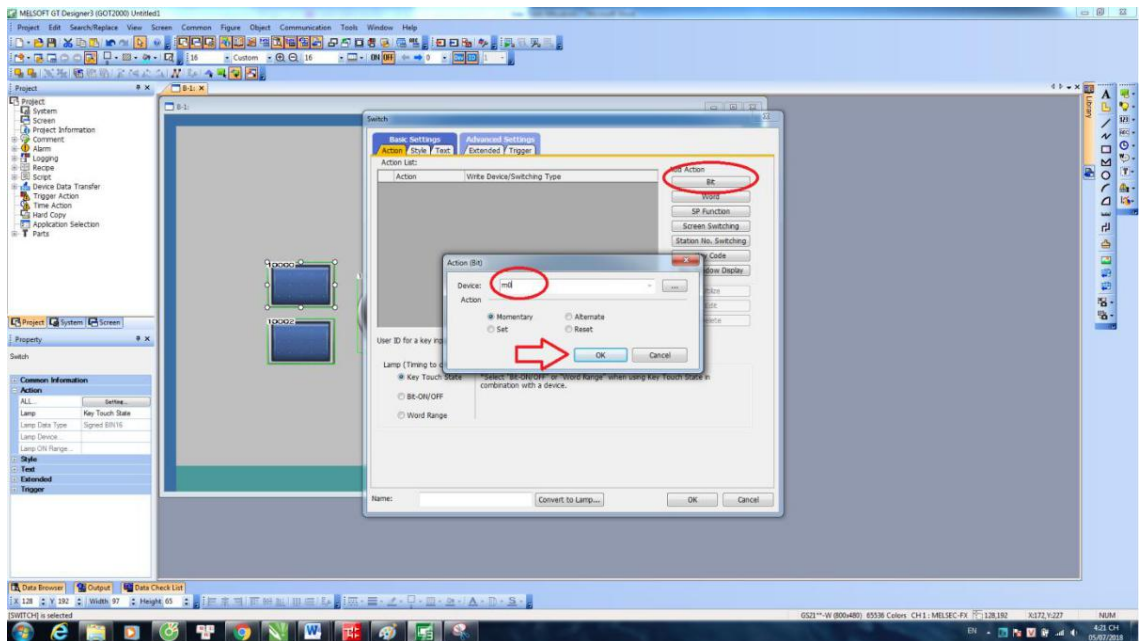
+ Sau khi vào GT Designer ta vào phần Object để đưa các nút nhấn và các hiển thị đèn lên giao diện của màn hình.



Ở đây để chọn nút nhất ta click vào Switch để chọn hiển thị đèn thì ta chọn Lamp sau đó kéo ra màn hình như hình vẽ.



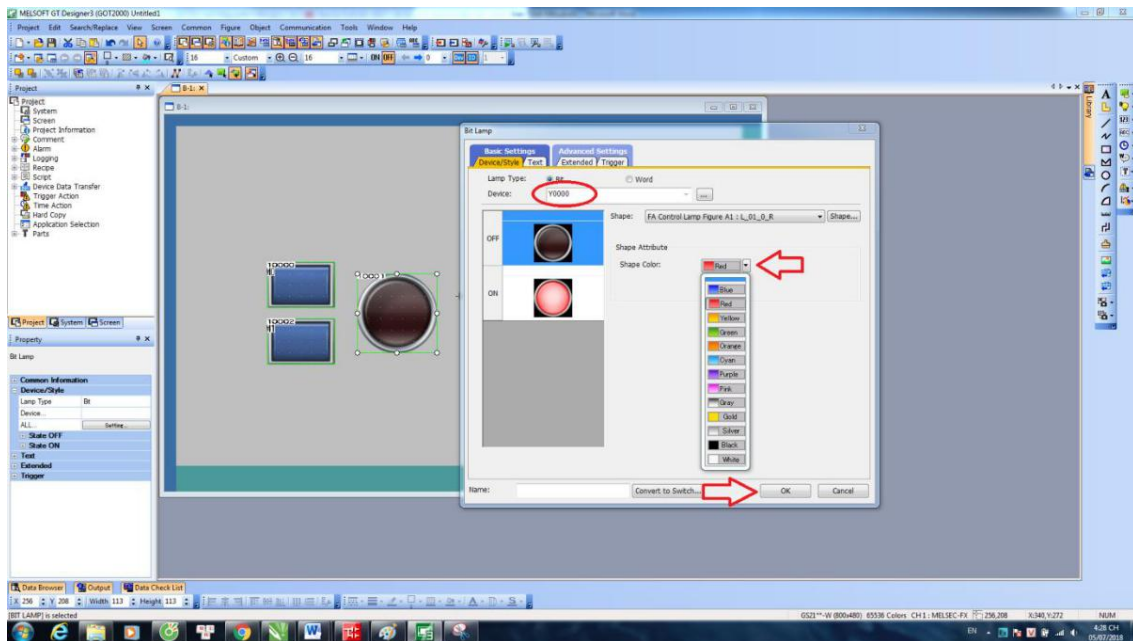
Sau khi kéo kéo các nút nhấn và các hiển thị đèn ra màn hình ta tiếp tục Tag địa chỉ kết nối giữa HMI và PLC bằng cách click đúp chuột vào nút nhấn để tiến hành Tag địa chỉ IN như sau:



Sau khi đúp chuột vào nút nhấn ngay lập tức sẽ hiện ra 1 cửa sổ ở đây người dùng sẽ Tag địa chỉ cho nút nhấn bằng cách click vào Bít và tag địa chỉ ở phần DEVICE ở chương trình phía trên ta đã dùng M0 là Start và M1 là Stop vậy ta lần lượt tag địa chỉ cho nút nhấn bằng biến M như trên.

Sau đó ta nhấn OK để hoàn thành quá trình Tag, tương tự ta Tag địa chỉ cho các nút nhấn còn lại như trên.

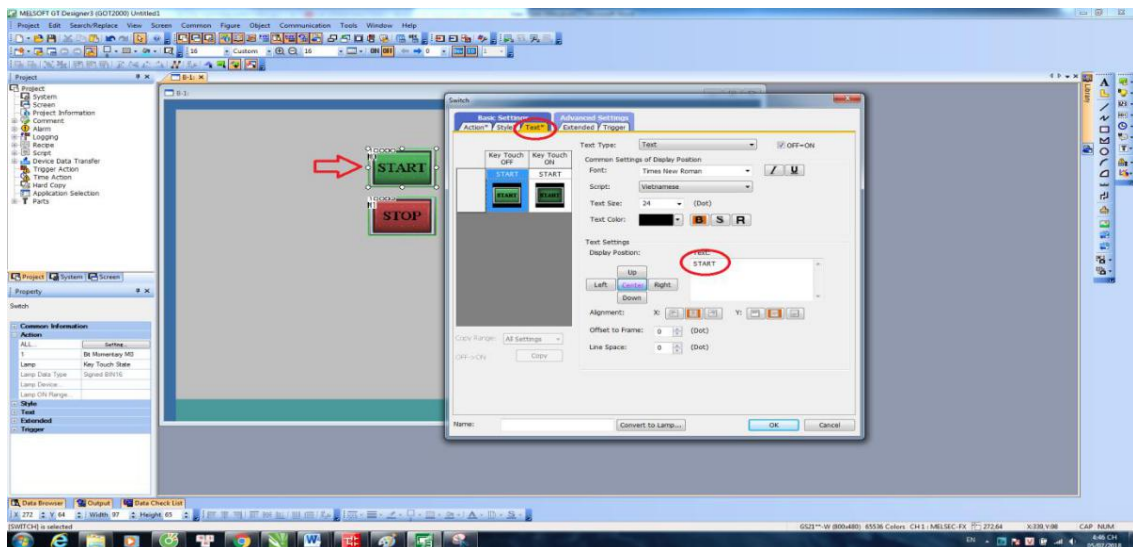
Đối với đèn hiển thị ta vẫn tiếp tục click chuột vào hình biểu thị đèn như trường hợp ta làm với nút nhấn.



ở phần DEVICE ta tag địa chỉ ngõ ra cho đèn ở đây ta đang sử dụng Y0 nên ta tag là Y0.

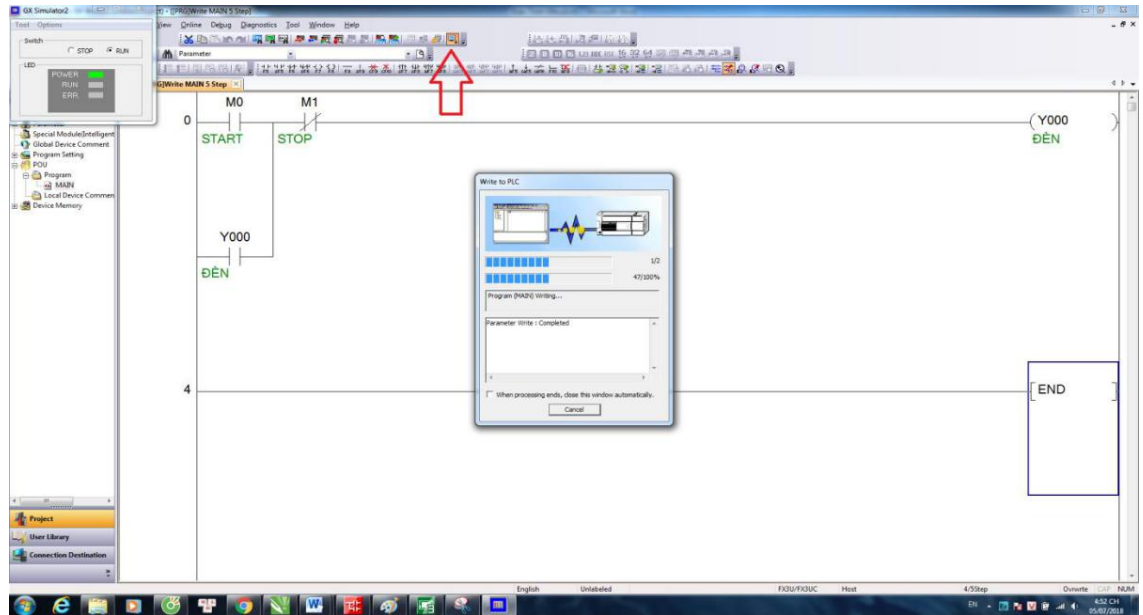
Tiếp theo ta lần lượt chọn đến biểu tượng hai đèn hiển thị hai trạng thái ON và OFF để chọn màu hiển thị cho đèn khi hoạt động rồi nhấn OK để kết thúc phần Tag địa chỉ cho đèn:

Ta có thể đặt tên cho nút nhấn và đèn bằng cách click chuột vào nút biểu tượng nút nhấn hoặc đèn rồi chọn Text để đặt tên cho nút nhấn và đèn như dưới:

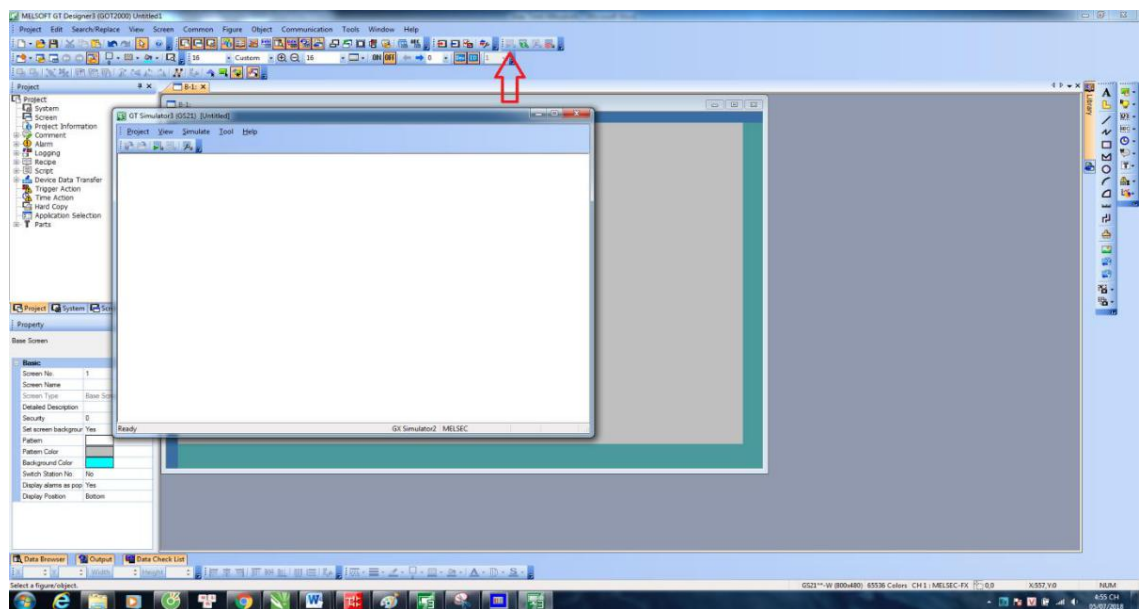


4. Chạy mô phỏng trên máy tính

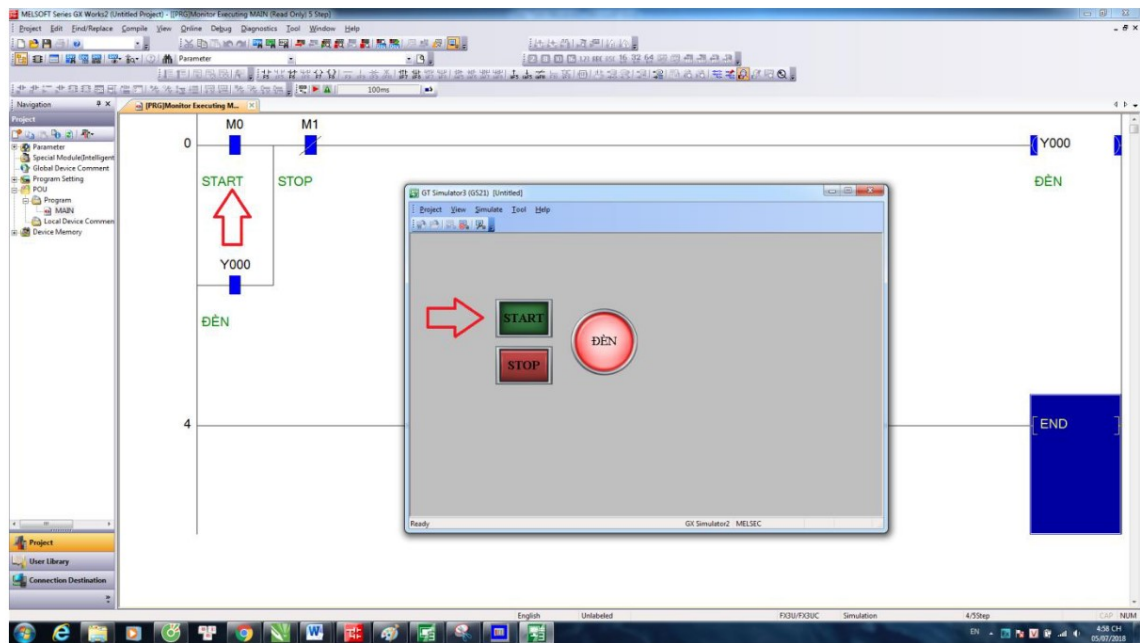
Trước tiên ta đến chương trình trên GX-Works2 sau đó bật mô phỏng lên bằng cách click chuột vào biểu tượng Start/Stop Simulation rồi tiến hành load chương trình xuống như thường lệ.



Sau khi load chương trình mô phỏng xong ta vào GT-Designer để mô phỏng màn hình lên



Sau khi cả chương trình và HMI đều bật mô phỏng lúc này ta thử kết nối giữa CPU và HMI bằng cách click vào nút nhấn để xem hoạt động của chương trình và HMI



Khi nhấn biểu tượng thấy hoạt động như hình ảnh thì chúng ta đã Tag thành công cho chương trình HMI

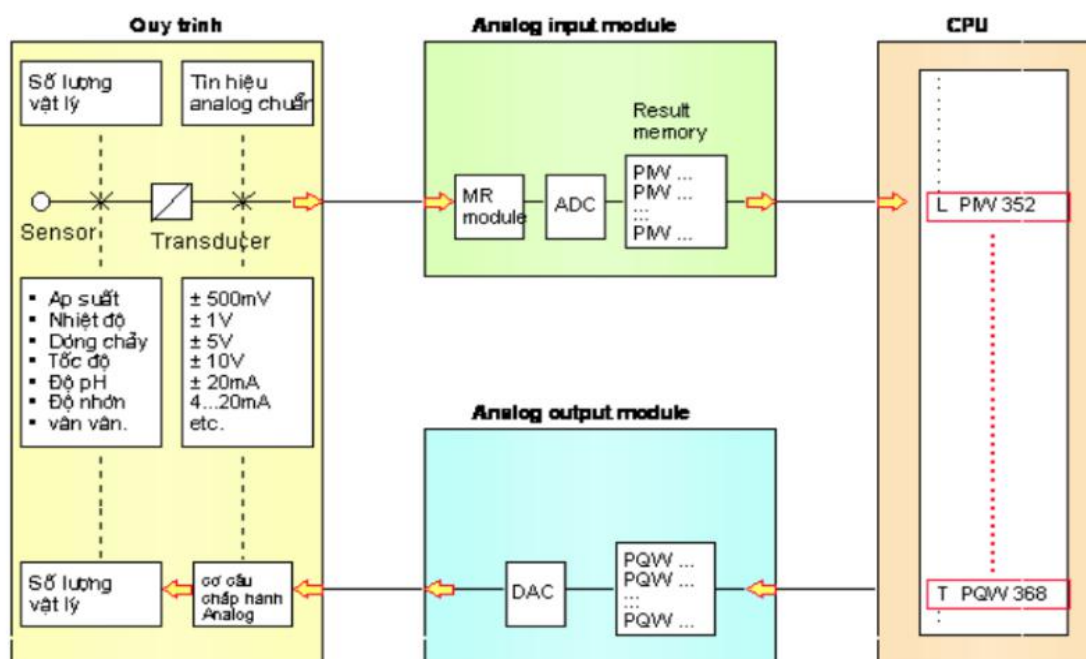
Chương 6 : Tín hiệu tương tự

1. Xử lý tín hiệu Analog

Trong quá trình điều khiển một hệ thống tự động hoá có thể có các yêu cầu điều khiển liên quan đến việc xử lý các tín hiệu Analog.

Các đại lượng vật lý như: nhiệt độ, áp suất, tốc độ, dòng chảy, độ PH... cần phải được các bộ Transducer chuẩn hoá tín hiệu trong phạm vi định mức cho phép trước khi nối tín hiệu vào ngõ vào Analog.

Chuẩn của tín hiệu điện áp là từ 0 đến 10 VDC hoặc chuẩn của tín hiệu Analog là dòng điện từ 4 đến 20 mA. Các Modul ngõ vào Analog (AI) bên trong có các bộ chuyển đổi ADC (Analog Digital Converter) để chuyển đổi các tín hiệu Analog nhận được thành các tín hiệu số đưa về CPU qua Bus dữ liệu. Các Modul ngõ ra Analog (AO) bên trong có bộ chuyển đổi DAC (Digital-Analog Converter) chuyển các tín hiệu số nhận được từ CPU ra các giá trị Analog có thể là áp hoặc dòng.



Output	Analog output range	Normal resolution mode	
		Digital input value	Maximum resolution
Voltage	0 to 5V	0 to 4000	1.25mV
	1 to 5V		1.0mV
	-10 to 10V	-4000 to 4000	2.5mV
Current	0 to 20mA	0 to 4000	5μA
	4 to 20mA		4μA

Trong bảng trên ta thấy giá trị điện áp từ 0 tới 5V, được biểu diễn bằng số từ 0 tới 4000

Ví dụ: giá trị đo được là 0.00125 V thì trong PLC được biểu diễn bằng số là

$$\frac{0.00125 \times 4000}{5}$$

5

Ví dụ: giá trị đo được là 0.0075 V thì trong PLC được biểu diễn bằng số là

$$\frac{0.0075 \times 4000}{5}$$

5

Từ những ví dụ trên ta thấy: ở phương pháp biểu diễn này PLC chỉ phát hiện mức điện áp chính xác tới 0.00125 V, nếu bé hơn thì không thể biểu diễn được, lúc đó ta phải dùng các biểu diễn khác để có thể thu được độ chính xác cao hơn.

Ta cũng làm tương tự với tín hiệu là dòng điện.

+ Cú pháp câu lệnh ghi xuống CPU



Trong đó TO là lệnh nạp giá trị xuống cho CPU

K0 vị trí thứ nhất là vị trí modul

K0 vị trí thứ hai là vị trí thanh ghi

H1033 là chế độ làm việc của các kênh

K1 số đơn vị nạp xuống

+ Cú pháp câu lệnh đọc lên CPU



Trong đó FROM là lệnh đọc giá trị lên

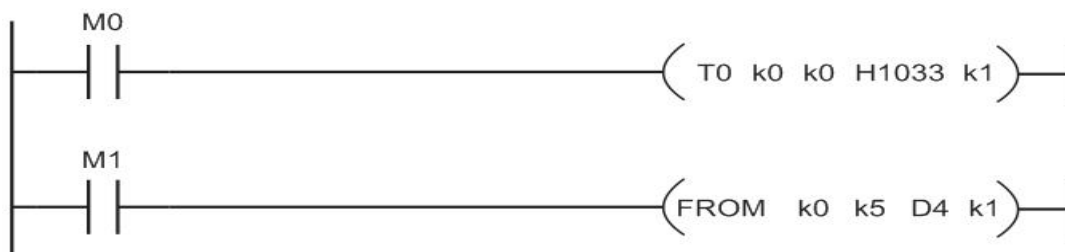
K0 là vị trí modul

K7 là vị trí kênh cần đọc ở đây K7 tức là ta đang muốn đọc kênh 3

D10 là thanh ghi lưu giữ giá trị đọc lên

K1 là đơn vị down xuống

Ví dụ:



=====

Khi ngõ vào M0 ON ngay lập tức các giá trị cài đặt sẽ được nạp vào CPU các chế độ làm việc các kênh làm việc như hình trên

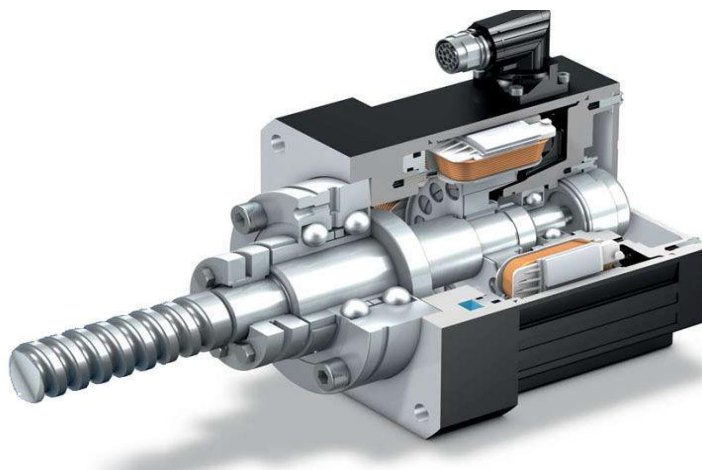
Khi ngõ vào M1 ON thì giá trị analog sẽ được đọc lên vidu ở trường hợp này giá trị down xuống bao gồm vị trí làm việc ở modun 0 ở thanh ghi 0 chế độ làm việc các kênh là kênh 1 làm việc ở chế độ (-20mA đến +20mA)

Chương 7 : Lập trình và điều khiển động cơ Servo

1. Giới thiệu chung về động cơ Servo

Động cơ servo được thiết kế cho những hệ thống hồi tiếp vòng kín.

Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bất kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.



2. Một số lệnh phát xung trên FX-3U

Các dòng PLC hỗ trợ phát xung tốc độ cao thì điều kiện cần thiết ngõ ra của PLC phải là dạng điện tử - Transistor, khi đó nếu CPU hỗ trợ lệnh, chúng ta sẽ có thể dùng một trong các lệnh như sau:

- PLSY : Phát xung vuông với tần số và số xung được đặt trong tham số lệnh.
- PLSR : Phát xung vuông tương tự lệnh PLSY nhưng có thêm tham số hiệu chỉnh việc tăng tần số phát xung và giảm tần số phát xung khi khởi động và kết thúc lệnh phát xung. Việc này tạo sườn dốc khi khởi động và dừng, giúp làm mềm chuyển động hơn lệnh PLSY ở những tốc độ cao.
- DRVA : Tương tự lệnh DRVI, nhưng vị trí ban đầu được xác định tuyệt đối. Số xung sẽ lưu lại trong thanh ghi và xác định tuyệt đối so với điểm ban đầu.
- DRVI : Phát xung kèm thêm phát lệnh đảo chiều theo giá trị +/- của xung. Lệnh này cũng cho phép cài đặt chỉ số hiệu chỉnh sườn dốc khi bắt đầu và chuẩn bị kết thúc lệnh. Mỗi lần phát xung, số xung được tính tương đối theo lệnh.

a. Điều khiển khoảng cách với lệnh PLSY và PLSR

+ Giả sử chúng ta sẽ lập trình lệnh PLSY phát xung ra cổng Y000 với cấu trúc chương trình như sau



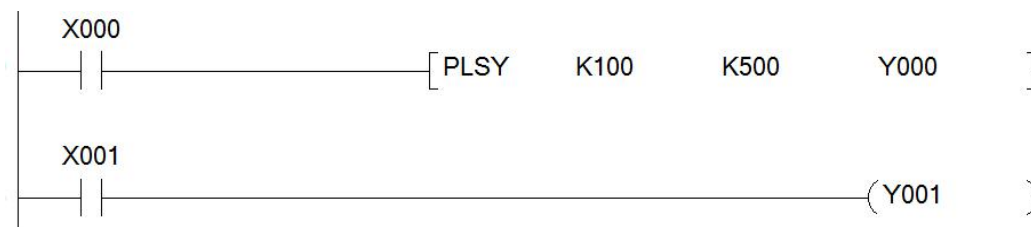
Ý nghĩa của lệnh PLSY như sau

D0 : Tần số xung sẽ xuất ra việc này quyết định tốc độ của Servo

D4 : Số xung sẽ phát ra khi lệnh PLSY được kích hoạt

Y000 : Là cổng phát xung được lựa chọn

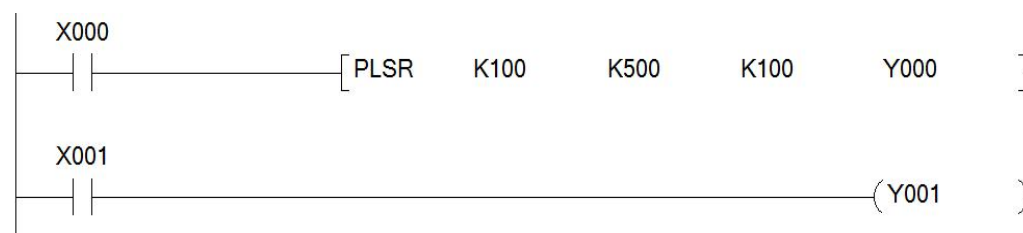
Ví dụ lệnh PLSY



Khi ngõ vào X000 ON thì động cơ sẽ quay thuận với độ 100 bằng hằng số K100 đặt trước và động cơ sẽ chạy được 500 xung với hằng số K500 đặt trước.

Để động cơ có thể quay ngược ta đồng thời ON ngõ vào X000 và X001

Ví dụ lệnh PLSR



Ý nghĩa của lệnh PLSR như sau

K100 : Tần số xung sẽ xuất ra việc này quyết định tốc độ của Servo

K500 : Số xung sẽ phát ra khi lệnh PLSR được kích hoạt

K100 : Thời gian tạo quá trình tăng tần số và giảm tần số khi thực hiện lệnh. Tính theo đơn vị ms.

Y000 : Là cổng phát xung được lựa chọn

Khi ngõ vào X000 ON thì động cơ sẽ quay thuận với độ 100 bằng hằng số K100 đặt trước và động cơ sẽ chạy được 500 xung với hằng số K500 đặt trước với thời gian tăng giảm tần số là 100ms tương ứng với hằng số đặt trước là K100

Để động cơ có thể quay ngược ta đồng thời ON ngõ vào X000 và X001

b. Điều khiển vị trí với các lệnh DRVA và DRVI

Cấu trúc lệnh lệnh DRVA như sau



Ý nghĩa của cấu trúc lệnh

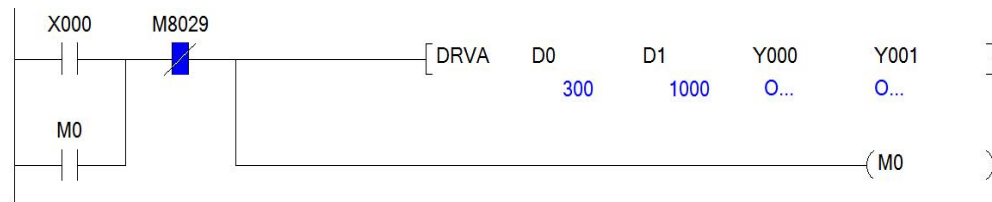
D0 : Tần số xung sẽ xuất ra việc này quyết định tốc độ của Servo

D4 : Số xung sẽ phát ra khi lệnh PLSY được kích hoạt

Y000 : Là cổng phát xung được lựa chọn cho quay thuận

Y001 : Là cổng phát xung được lựa chọn cho quay ngược

Ví dụ lệnh DRVA



Giải thích chương trình :

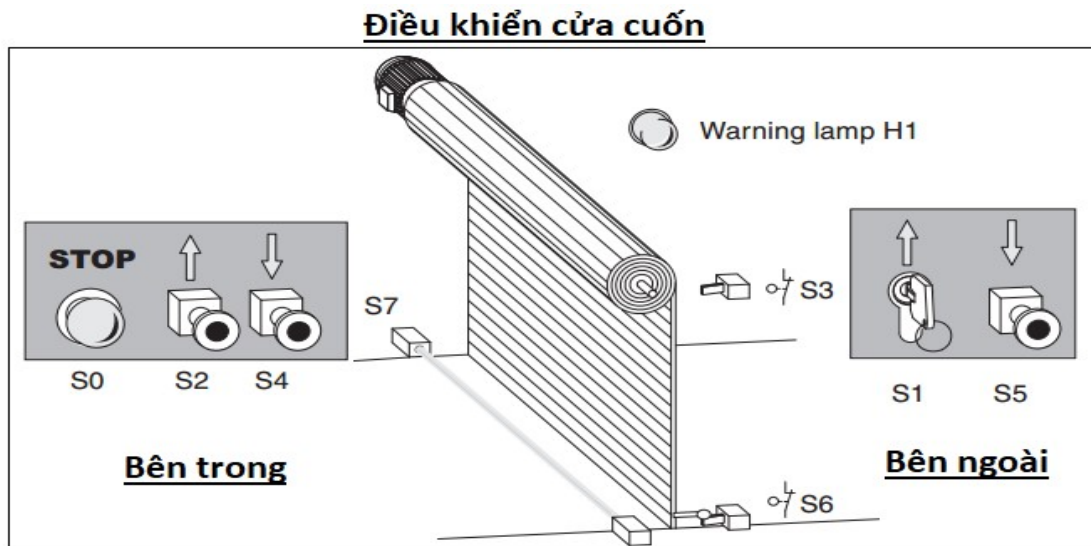
Đối với lệnh DRVA khi download chương trình xuống thì CPU sẽ tự xét vị trí đang đứng của servo là vị trí gốc khi ta gọi số âm xung hay số dương xung lệnh này sẽ tự điều khiển ngõ ra là Y000 và Y001 để hoạt động đối với trường hợp trên :

Khi ngõ vào X000 ON tiếp điểm duy trì M0 sẽ ON và duy trì trạng thái để lệnh DRVA có thể làm việc ở trên ta thấy tham số tốc độ của động cơ đặt trước là 300 (D300), số xung đặt trước của động cơ là 1000 (D1000), hai ngõ ra điều khiển là Y000 và Y001 khi servo chạy hoàn thành 1 chu trình được gọi thì tiếp điểm M8029 sẽ được ON và ngắt tiếp điểm duy trì M0

Đối với lệnh DRVA người dùng có thể đặt khoảng phạm vi hoạt động của servo vì thực tế người dùng có thể tự quy đổi số xung qua hệ mm hay cm.

Chương 8: Bài Tập

BÀI 1: ĐIỀU KHIỂN CỬA CUỐN



Nguyên lý vận hành:

Để điều khiển cửa cuốn lên và xuống, chúng ta thiết kế 2 bảng điều khiển ở hai phía bên trong, bên ngoài.

Bảng điều khiển này bố trí hai nút lên - xuống, riêng phía bên trong có thêm nút Stop.

- Phía bên trái thiết kế cảm biến S7: có tín hiệu này trong khi cửa đang đóng thì PLC sẽ điều khiển cửa tự động mở ra.
- Phía bên phải đặt 2 công tắc hành trình giới hạn trên (S3), giới hạn dưới (S6)

Để điều khiển cửa lên, xuống chúng ta dùng 2 đầu ra của PLC để điều khiển động cơ quay thuận và ngược. Ta quy ước quay thuận để đóng cửa, quay ngược để mở cửa.

Ngoài ra có thêm đèn cảnh báo trong trường hợp có đồng thời cả hai tín hiệu giới hạn S3 và S6.

a. Lập bảng địa chỉ vào/ra?

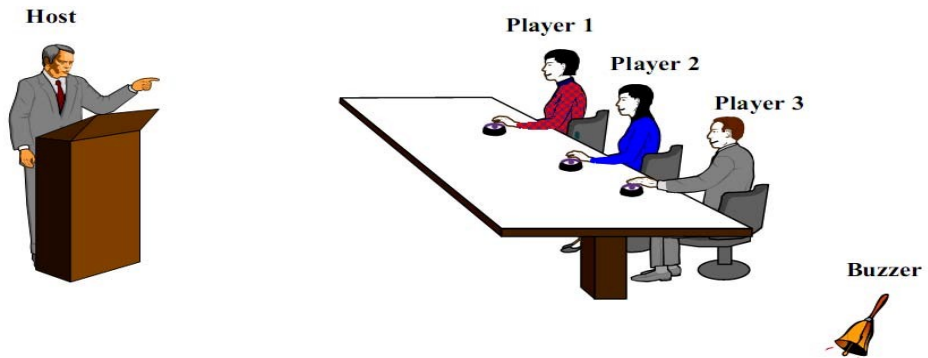
	Địa chỉ	Ghi chú
Input		
Output		

b. Vẽ sơ đồ kết nối.

.....

c. Viết lưu đồ thuật toán và chương trình PLC.

.....

Bài 2: BÀI TOÁN ƯU TIÊN**Yêu cầu:**

Có 3 người chơi: player1, player2, player3 và một người dẫn chương trình (host).

Người dẫn chương trình đặt câu hỏi (cả 3 đèn báo tắt). Sau khi kết thúc câu hỏi ba người nhấn nút để trả lời, chương trình sẽ ưu tiên người nhấn trước, 2 người còn lại nhấn không tác dụng.

Người chơi: Khi người nhấn đầu tiên nhấn thì chuông kêu trong 3 giây (1 chuông dùng chung cho cả 3 người chơi) và đèn của người nhấn đó sáng nhấp 3 lần chu kỳ 1s sau đó sáng duy trì.

Người dẫn chương trình có nút nhấn reset để bắt đầu câu hỏi mới, và nút start để bắt đầu 1 cuộc chơi (chỉ cần nhấn Start 1 lần cho 1 cuộc chơi)

Thực hiện: Làm theo các bước sau

d. Lập bảng địa chỉ vào/ra?

	Địa chỉ	Ghi chú
Input		
Output		

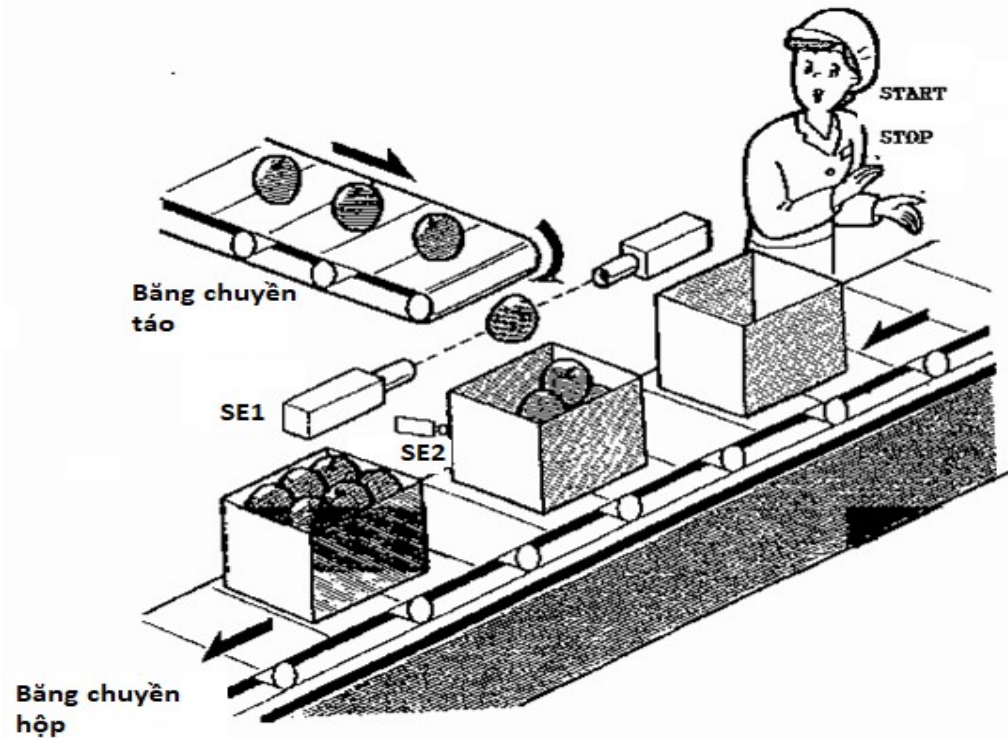
e. Vẽ sơ đồ kết nối.

.....

f. Viết lưu đồ thuật toán và chương trình PLC.

.....

BÀI 3: DÂY CHUYỀN ĐÓNG GÓI



Yêu cầu:

Khi nhấn nút START băng chuyền hộp hoạt động. Khi cảm biến thùng SE2 báo, thì băng chuyền hộp dừng lại và băng chuyền táo hoạt động.

Táo rơi vào thùng có cảm biến táo SE1 đếm. Khi đếm đủ 10 quả táo, băng tải táo dừng và băng tải hộp chạy trở lại.

Chu trình làm việc cứ lặp lại như vậy.

Khi nhấn nút Stop, cả 2 băng tải đều dừng lại.

Thực hiện:

a. Bảng địa chỉ vào/ra.

	Địa chỉ	Ghi chú
Input		

Output		

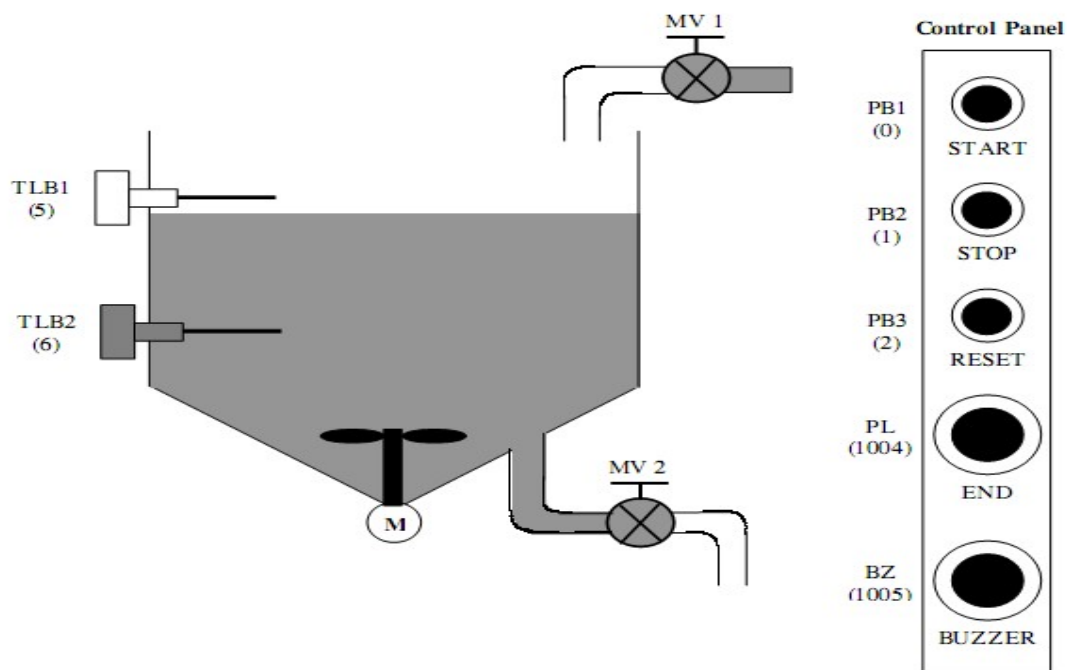
b. Vẽ sơ đồ kết nối.

.....

c. Viết lưu đồ thuật toán và chương trình điều khiển.

.....

BÀI 4: ĐIỀU KHIỂN TRỘN LIỆU



Yêu cầu: Thiết kế và điều khiển 1 chu trình trộn liệu làm việc như sau

Bước1:

Khi nhấn PB1, van MV1 mở và dung dịch bắt đầu chảy vào thùng chứa. Sau một thời gian thì động cơ M bắt đầu chạy (khuấy).

Bước2:

Khi dung dịch tràn qua cảm biến TLB2 tới cảm biến TLB1, van MV1 đóng lại và động cơ cũng ngừng quay.

Bước3:

Tiếp theo, van MV2 sẽ mở và bắt đầu xả dung dịch đã được khuấy. Khi dung dịch tụt xuống tới vị trí cảm biến TLB2 van MV2 đóng lại.

Chu trình làm việc từ bước 1 đến bước 3 lặp lại 4 lần, sau đó đèn END và Chuông báo Buzzer hoạt động.

Một chu trình làm việc mới chỉ được bắt đầu khi nhấn lại nút PB1.

Thực hiện:

a. Lập bảng địa chỉ

	Địa chỉ	Ghi chú
Input		
Output		

b. Vẽ sơ đồ đấu nối thiết bị.

.....

c. Viết lưu đồ thuật toán chương trình điều khiển.

.....
