

Bùi Hồng Huế – Đinh Xuân Thủy

HƯỚNG DẪN THỰC HÀNH PLC MITSUBISHI FX



HÀ NỘI - 2020

MỤC LỤC

LỜI NÓI ĐẦU	3
HƯỚNG DẪN SỬ DỤNG TÀI LIỆU.....	4
PHẦN I: LÀM QUEN VỚI BỘ LẬP TRÌNH PLC MITSUBISHI FX3U	5
1. GIỚI THIỆU TỔNG QUAN VỀ PLC	5
1.1. GIỚI THIỆU CHUNG VỀ THIẾT BỊ PLC.....	5
1.1.1. PLC là gì?	5
1.1.2. Khả năng ứng dụng của PLC	5
1.1.3. Đặc điểm quá trình tự động hoá dùng PLC	6
1.2. CẤU TRÚC CƠ BẢN VÀ HOẠT ĐỘNG CỦA BỘ ĐIỀU KHIỂN PLC ..	6
1.2.1. Cấu trúc cơ bản của bộ điều khiển PLC.....	6
1.2.2. Hoạt động của PLC	9
1.3. BIỂU DIỄN CÁC ĐẠI LƯỢNG TRONG PLC.....	10
1.3.1. Các hệ đếm sử dụng trong PLC.....	10
1.3.2. Các phương pháp biểu diễn đại lượng trong PLC	11
1.4. CÁC BIT ĐẦU VÀO/RA TRONG PLC VÀ CÁC THIẾT BỊ ĐIỆN BÊN NGOÀI.....	13
1.4.1. Các bit đầu vào trong PLC với các thiết bị điện bên ngoài	13
1.4.2. Các bit đầu ra trong PLC với các thiết bị điện bên ngoài	14
1.5. CÁC ĐẶC TÍNH KỸ THUẬT CHỦ YẾU CỦA PLC	14
1.6. THỦ TỤC THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN SỬ DỤNG BỘ ĐIỀU KHIỂN PLC	15
2. GIỚI THIỆU PLC MITSUBISHI dòng FX-3U.....	16
2.1. CÁC DÒNG PLC PHỔ BIẾN CỦA MITSUBISHI	16
2.2. PHÂN LOẠI VÀ CÁCH NHẬN BIẾT KÝ MÃ HIỆU.....	16
2.3. ĐẶC TÍNH KỸ THUẬT CỦA PLC MITSUBISHI FX3U:	17
2.4. SƠ ĐỒ KẾT NỐI I/O PLC MITSUBISHI FX3U	19
2.4.1. Nối INPUT	19
2.4.2. Nối OUTPUT	20
2.5. PHẦN MỀM LẬP TRÌNH CHO PLC Mitsubishi dòng FX.....	22
2.6. CẤU TRÚC BỘ NHỚ, CÁC VÙNG NHỚ VÀ ĐỊA CHỈ BỘ NHỚ TRONG MITSUBISHI FX3U:.....	23
2.6.1. Cấu trúc bộ nhớ:	23
2.6.2. Địa chỉ bộ nhớ trong PLC MITSUBISHI FX3U:	24
2.7. NỐI GHÉP GIỮA PLC VÀ THIẾT BỊ NGOÀI VI.....	25
3. GIỚI THIỆU BỘ THỰC HÀNH PLC MITSUBISHI	25
PHẦN II: HƯỚNG DẪN THỰC HÀNH.....	30
Bài thực hành số 1	30
ĐẦU NỐI PHÂN CỨNG PLC MITSUBISHI FX3U	30
Bài thực hành số 2.....	36
HƯỚNG DẪN SỬ DỤNG PHẦN MỀM GX-WORKS 2	36
Bài thực hành số 3.....	44

THỰC HÀNH VỚI CÁC LỆNH CƠ BẢN	44
Bài thực hành số 4.....	52
THỰC HÀNH VỚI CÁC LỆNH TOÁN HỌC	52
Bài thực hành số 5	60
LẬP TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ XOAY CHIỀU 3 PHA.....	60
Bài thực hành số 6.....	66
LẬP TRÌNH HMI	66
Bài thực hành số 7.....	88
LẬP TRÌNH TIMER.....	88
Bài thực hành số 8.....	93
LẬP TRÌNH COUNTER	93
Bài thực hành số 9.....	99
LẬP TRÌNH ĐẾM SẢN PHẨM VÀ HIỂN THỊ LED 7 THANH	99
Bài thực hành số 10.....	105
LẬP TRÌNH BỘ ĐẾM TỐC ĐỘ CAO	105
ĐỌC XUNG ENCODER TƯƠNG ĐỐI	105
Bài thực hành số 11	114
ĐỌC CẢM BIẾN MÀU KEYENCE CZ-V1	114
Bài thực hành số 12.....	118
XỬ LÝ TÍN HIỆU ANALOG SỬ DỤNG MODULE FX2N-5A.....	118
Bài thực hành số 13.....	129
THỰC HÀNH CÀI ĐẶT VÀ ĐIỀU KHIỂN	129
BIẾN TẦN MITSUBISHI FR-E700.....	129
Bài thực hành số 14.....	143
LẬP TRÌNH ĐIỀU KHIỂN BIẾN TẦN MISUBISHI FR-E700	143
SỬ DỤNG MODULE ANALOG FX2N-2DA.....	143
Bài thực hành số 15.....	149
LẬP TRÌNH ĐO VÀ HIỂN THỊ TỐC ĐỘ ĐỘNG CƠ	149
Bài thực hành số 16.....	156
LẬP TRÌNH TRUYỀN THÔNG MODBUS RTU GIỮA PLC FX3U VỚI BIẾN TẦN MISUBISHI FR-E700 SỬ DỤNG MODULE FX3U-485BD..	156
Bài thực hành số 17.....	162
ĐIỀU KHIỂN ĐỘNG CƠ SERVO MITSUBISHI	162
Bài thực hành số 18.....	173
LẬP TRÌNH ĐIỀU KHIỂN VỊ TRÍ SỬ DỤNG PLC MITSUBISHI VÀ SERVO MR-J3-10A	173
Bài thực hành số 19.....	182
LẬP TRÌNH NGUYÊN ĐIỂM.....	182
SỬ DỤNG PLC MITSUBISHI VÀ SERVO MR-J3-10A	182

LỜI NÓI ĐẦU

Sự tiến bộ trong công nghệ điện tử – tin học ngày nay thực sự là một cuộc cách mạng công nghệ trên toàn thế giới. Ở nước ta, kỹ thuật điện tử – tin học đã được ứng dụng vào lĩnh vực điều khiển tự động, đặc biệt là kỹ thuật vi xử lý. Hiện nay người ta đã sản xuất những thiết bị có kết cấu rất nhỏ gọn dạng máy tính mà bên trong có chứa bộ vi xử lý có thể lập trình được. Đó chính là thiết bị điều khiển lập trình (**Programmable Logic Controller**) viết tắt là PLC. So với quá trình điều khiển bằng mạch điện tử thông thường thì PLC có nhiều ưu điểm hơn hẳn, chẳng hạn như: Kết nối mạch điện đơn giản, rút ngắn được thời gian lắp đặt công trình, dễ dàng thay đổi công nghệ nhờ việc thay đổi nội dung chương trình điều khiển, ứng dụng điều khiển trong phạm vi rộng, độ tin cậy cao...

Chính vì những ưu điểm trên, bộ điều khiển lập trình đã được ứng dụng rộng rãi vào lĩnh vực điều khiển tự động như: Tự động hoá quá trình cung cấp vật liệu cho quá trình sản xuất, tự động hoá các máy gia công cơ khí, điều khiển hệ thống trạm bơm, điều khiển các thiết bị thuỷ lực và khí nén, tự động hoá quá trình lắp ráp các linh kiện điện - điện tử, điều khiển thang máy, hệ thống đèn giao thông...

Nhằm nhanh chóng tiếp cận với kỹ thuật tiên tiến này đồng thời đưa vào việc giảng dạy ngành học tự động hoá và lập trình PLC cho các trường kỹ thuật, chúng tôi đã thiết kế xây dựng một số bài thực hành cơ bản và nâng cao ứng dụng nhiều trong sản xuất công nghiệp chạy trên họ PLC MITSUBISHI FX3U. Trên cơ sở các bài thực hành cơ bản các bạn có thể phát triển với các bài toán phức tạp hơn, đáp ứng nhiều các yêu cầu công nghệ khác nhau.

Do bộ tài liệu đề cập đến nhiều vấn đề mới, viết cho nhiều đối tượng ở các trình độ khác nhau nên không tránh khỏi những thiếu sót. Kính mong các bạn đọc gần xa, các chuyên gia kỹ thuật tham gia đóng góp ý kiến để lần tái bản sau được hoàn thiện hơn.

Chúng tôi xin chân thành cảm ơn các thầy, cô giáo, cũng như các chuyên gia về PLC đã quan tâm đóng góp ý kiến trước khi bộ tài liệu được hoàn thiện và phát hành.

Xin chân thành cảm ơn!

Hà Nội, tháng 5, năm 2020

Thay mặt nhóm tác giả

TS. Bùi Hồng Huế

HƯỚNG DẪN SỬ DỤNG TÀI LIỆU

1. Đối với giáo viên hướng dẫn

Giáo viên hướng dẫn cần phải trang bị những kiến thức thực tế và dựa vào các trang thiết bị sẵn có để mở rộng thêm các bài thực hành cho các học viên.

Trước mỗi bài hướng dẫn, giáo viên hướng dẫn cần phải làm trước ít nhất một lần, dự kiến trước các tình huống có thể xảy ra, sẵn sàng giải quyết bất cứ câu hỏi nào từ phía học viên, tránh những tình huống sư phạm đáng tiếc xảy ra.

Tài liệu này cũng hàm ý xây dựng một ngân hàng các bài tập thực hành về lập trình PLC, tùy theo mục tiêu, nội dung cũng như cấp bậc đào tạo học viên mà chúng ta lựa chọn những bài học phù hợp.

Trong bộ tài liệu này, các tác giả chủ yếu trang bị các kiến thức và kỹ năng thực hành. Các đồng nghiệp và các học viên muốn tìm hiểu sâu hơn về lĩnh vực PLC để ứng dụng điều khiển thiết bị điện công nghiệp nên tìm đọc cuốn “*Giáo trình điện Công nghiệp*” của cùng tác giả- in tại nhà xuất bản Xây dựng năm 2003.

2. Đối với học viên

Tài liệu “**HƯỚNG DẪN THỰC HÀNH PLC**” này chủ yếu đề cập đến kỹ năng lập trình PLC, do đó phần lý thuyết chỉ tóm tắt trang bị điện và nguyên lý hoạt động cơ bản của mạch điện. Vì vậy trước khi thực hành, học viên cần phải được trang bị kiến thức lý thuyết tương ứng.

Phải đọc kỹ từng bài học, từ việc xác định mục đích của bài học, phần tóm tắt lý thuyết, chuẩn bị vật liệu dụng cụ đến trình tự thực hành và phải tự viết báo cáo cũng như trả lời các câu hỏi mà tài liệu đặt ra.

Mỗi bài thực hành đều có sơ đồ nguyên lý cụ thể để mạch hoạt động đúng chức năng. Những bài mẫu mà chúng tôi đưa ra nhằm giúp các học viên rèn luyện các kỹ năng cơ bản để các bạn tham khảo. Các bạn có thể tự sáng tác theo ý tưởng riêng. Có như vậy bạn mới phát huy được khả năng sáng tạo của mình.

Tuyệt đối tuân thủ sự hướng dẫn của giáo viên để tránh những tai nạn đáng tiếc xảy ra.

PHẦN I: LÀM QUEN VỚI BỘ LẬP TRÌNH PLC MITSUBISHI FX3U

1. GIỚI THIỆU TỔNG QUAN VỀ PLC

1.1. GIỚI THIỆU CHUNG VỀ THIẾT BỊ PLC

1.1.1. PLC là gì?

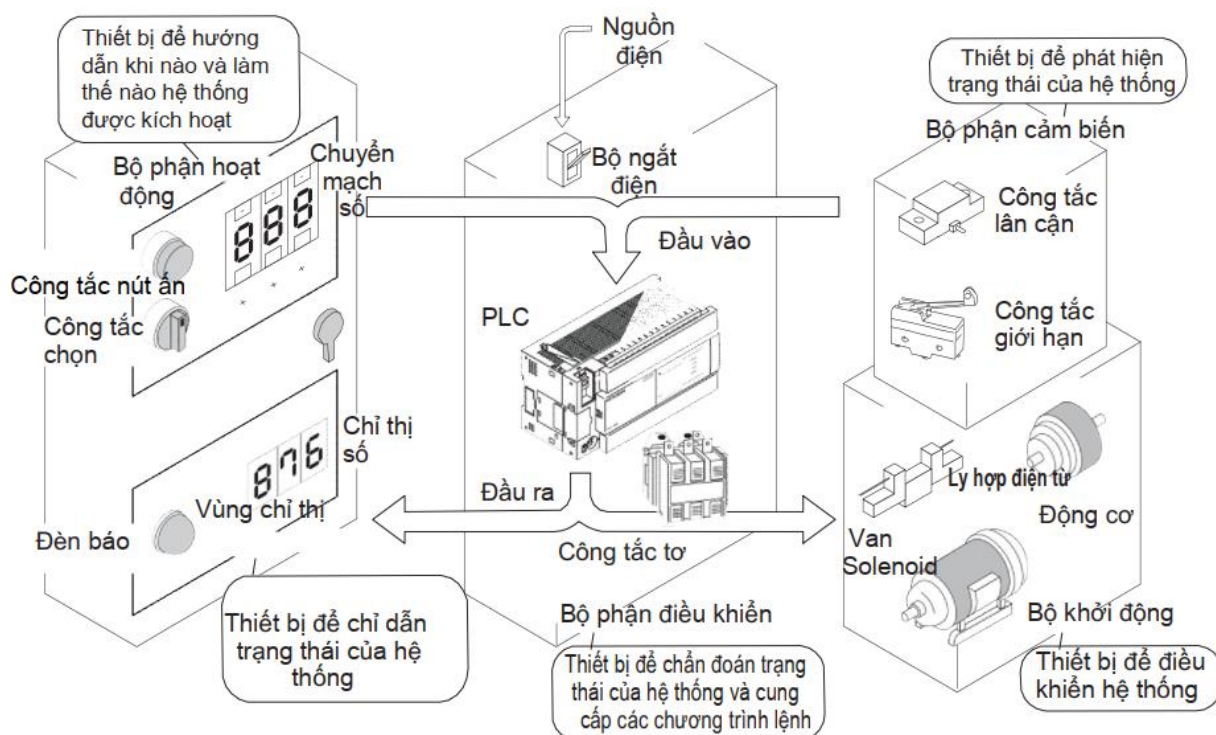
PLC viết tắt của cụm từ “**P**rogrammable **L**ogic **C**ontroller” được hiểu là bộ điều khiển có khả năng lập trình được. Nó chính là một máy tính công nghiệp để thực hiện một dãy quá trình sản xuất và thường được gắn ngay tại dây chuyền sản xuất. Một cách hiểu khác thì PLC là một thiết bị điều khiển được trang bị các chức năng logic, tạo xung, đếm thời gian, đếm xung và thực hiện nhiều phép tính kỹ thuật ứng dụng trong kỹ thuật điều khiển tự động hoá.

Ngoài thuật ngữ **PLC**, còn có một số thuật ngữ khác nữa cũng để chỉ bộ điều khiển lập trình như: **PC** - Programmable Controller (Anh) hay **PBS** - Programmable Binary System (Thụy điển)...

Hiện nay, trên thế giới PLC được sản xuất rất đa dạng về chủng loại, do các hãng khác nhau sản xuất như Mitsubishi, OMRON, Siemens...

1.1.2. Khả năng ứng dụng của PLC

Bộ điều khiển lập trình PLC được ứng dụng rất rộng rãi trong các ngành công nghiệp đặc biệt là lĩnh vực điều khiển tự động.



Hình 1.1. Sơ đồ ứng dụng PLC trong công nghiệp

Ví dụ:

- Tự động hoá quá trình cung cấp vật liệu cho quá trình sản xuất.
- Tự động hoá các máy gia công cơ khí như : khoan, tiện, phay...
- Điều khiển hệ thống trạm bơm
- Điều khiển các thiết bị thuỷ lực và khí nén.
- Tự động hoá quá trình lắp ráp các linh kiện điện - điện tử.
- Điều khiển các thiết bị nâng chuyển như băng tải, cầu thang máy, cần cầu...
- Tự động hoá quá trình phân loại sản phẩm
- Điều khiển rô-bốt...

1.1.3. Đặc điểm quá trình tự động hoá dùng PLC

Sử dụng PLC trong tự động hoá các quá trình sản xuất có những ưu điểm sau:

- Đấu nối các thiết bị với PLC đơn giản, rút ngắn được thời gian lắp đặt công trình.
- Dễ dàng thay đổi công nghệ cũng như nội dung chương trình điều khiển.
- Kết cấu mạch điện sử dụng PLC nhỏ gọn, giảm được kích thước định hình.
- Dễ dàng thay đổi thiết kế nhờ phần mềm.
- ứng dụng điều khiển trong phạm vi rộng.
- Xử lý sự cố dễ hơn và nhanh hơn.
- Độ tin cậy cao.
- Chuẩn hoá được phần cứng điều khiển.
- Thích ứng trong môi trường khắc nghiệt: Nhiệt độ, độ ẩm, điện áp dao động, tiếng ồn.

Tuy nhiên nó cũng có một số nhược điểm sau:

- Việc thiết kế, sửa chữa chương trình cho PLC đòi hỏi phải có đội ngũ cán bộ hiểu biết về lĩnh vực tin học – cần phải có quá trình đào tạo.
- Giá thành tương đối cao.

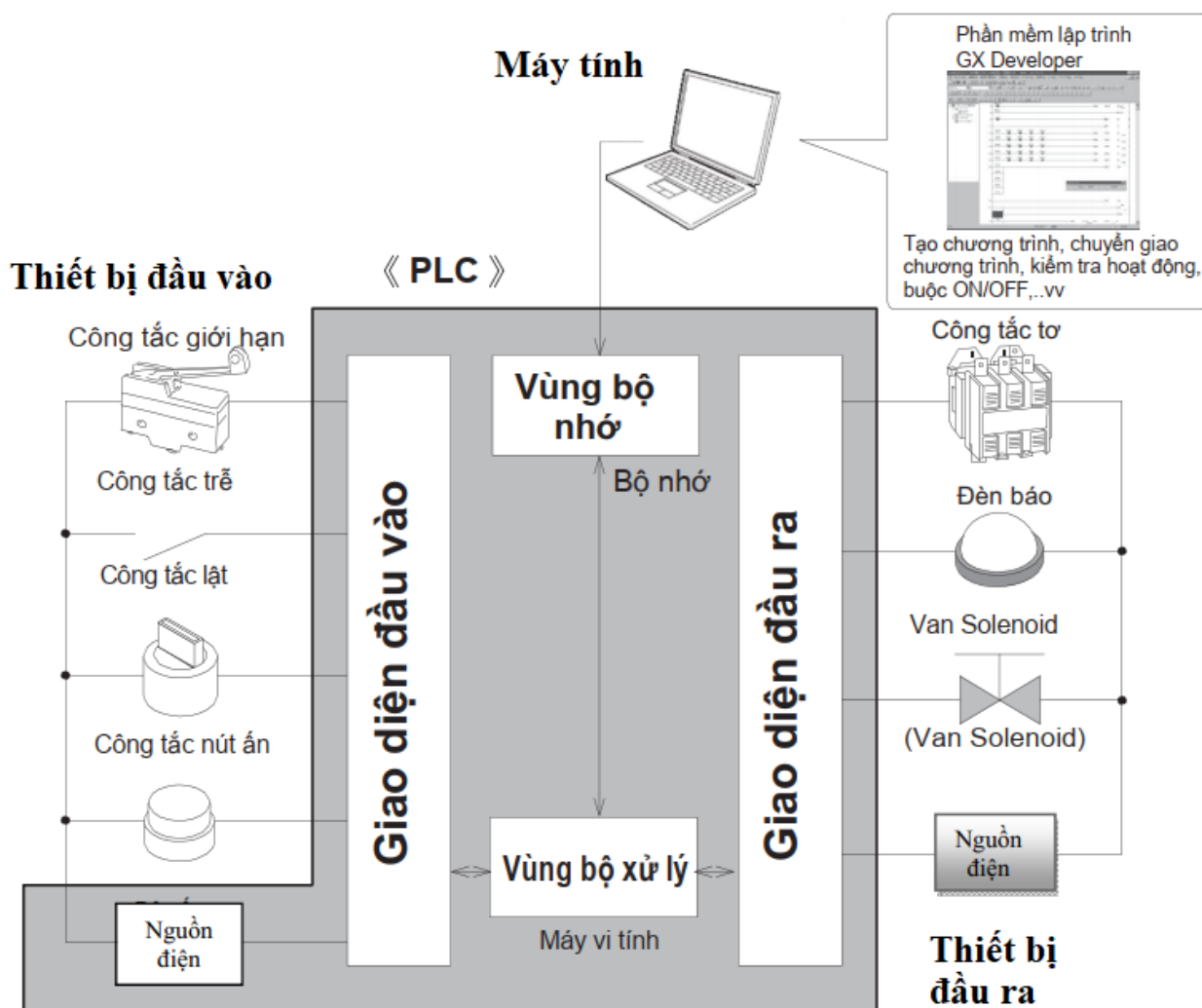
1.2. CẤU TRÚC CƠ BẢN VÀ HOẠT ĐỘNG CỦA BỘ ĐIỀU KHIỂN PLC

1.2.1. Cấu trúc cơ bản của bộ điều khiển PLC

PLC được coi như một máy vi tính dành cho các mục đích công nghiệp. Vì vậy PLC gồm ba khối chức năng cơ bản, đó là:

- Bộ xử lý.
- Bộ nhớ.
- Khối vào/ra.

Trạng thái ngõ vào của PLC được phát hiện và lưu vào bộ nhớ đệm. PLC thực hiện các lệnh logic trên các trạng thái của chúng và thông qua trạng thái ngõ ra được cập nhật và lưu vào bộ nhớ đệm; sau đó, trạng thái ngõ ra trong bộ nhớ đệm được dùng để đóng mở các tiếp điểm kích hoạt các thiết bị công tác. Như vậy, sự hoạt động các thiết bị được điều khiển hoàn toàn tự động theo chương trình trong bộ nhớ. Chương trình được nạp vào PLC qua thiết bị lập trình chuyên dùng. Xem hình 1.2



Hình 1.2. Sơ đồ kết nối vào ra PLC

- Đơn vị xử lý trung tâm

Đơn vị xử lý trung tâm (CPU- Central processing Unit) điều khiển và quản lý tất cả hoạt động bên trong PLC. Việc trao đổi thông tin giữa CPU, bộ nhớ và khối vào/ra được thực hiện thông qua hệ thống bus dưới sự điều khiển của CPU.

- Hệ thống bus

Hệ thống bus là tuyến dùng để truyền tín hiệu ở bên trong PLC. Nói cách khác, nó làm nhiệm vụ trao đổi thông tin giữa CPU, bộ nhớ và các cổng vào ra. Hệ thống gồm nhiều đường tín hiệu song song:

+ **Bus địa chỉ (Address bus):** để PLC truy cập đến các bộ phận khác nhau của PLC: bộ nhớ, khối vào ra, bộ đếm...

+ **Bus dữ liệu (Data bus):** Bus dùng để truyền dữ liệu.

+ **Bus điều khiển (Control Bus):** Dùng để truyền các tín hiệu: điều khiển, đồng bộ các hoạt động bên trong PLC.

Bên cạnh đó, CPU được cung cấp một xung Clock có tần số từ 1÷8MHz. Xung này quyết định tốc độ hoạt động của PLC và cung cấp các yếu tố về thời gian, đồng hồ của hệ thống.

- **Bộ nhớ**

Bộ nhớ (Memory) có nhiệm vụ lưu chương trình điều khiển được lập bởi người dùng và các dữ liệu khác như cờ, thanh ghi tạm, trạng thái đầu vào, lệnh điều khiển đầu ra, ... Nội dung của bộ nhớ được mã hoá dưới dạng mã nhị phân.

Mỗi lệnh trong chương trình có một vị trí riêng trong bộ nhớ, tất cả mọi vị trí trong bộ nhớ đều được “đánh số”. Những số này chính là địa chỉ ô nhớ trong bộ nhớ.

Địa chỉ của từng ô nhớ sẽ được trở đến bởi một bộ đếm địa chỉ ở bên trong bộ vi xử lý. Bộ xử lý sẽ tự động tăng giá trị trong bộ đếm này trước khi thực hiện lệnh tiếp theo. Với một địa chỉ mới, nội dung của ô nhớ tương ứng sẽ xuất hiện ở đầu ra. Quá trình này được gọi là quá trình đọc.

Tất cả PLC đều thường dùng các loại bộ nhớ sau:

+ ROM (Read Only Memory).

+ RAM (Random Access Memory).

+ EEPROM (Electronic Programmable Read Only Memory).

Với sự tiến bộ của công nghệ chế tạo bộ nhớ, nên hầu như các PLC đều dùng bộ nhớ EEPROM. Trường hợp ứng dụng cần bộ nhớ lớn có thể chọn lựa giữa bộ nhớ RAM có nguồn pin nuôi và bộ nhớ EEPROM. Ngoài ra, PLC cần thêm bộ nhớ RAM cho các chức năng khác như:

- Bộ đếm để lưu trạng thái các ngõ vào và ngõ ra

- Bộ nhớ tạm cho tác vụ định thì, tác vụ đếm, truy xuất cờ.

Dung lượng bộ nhớ:

Đối với PLC loại nhỏ, thường bộ nhớ có dung lượng cố định khoảng 2K. Dung lượng này là đủ đáp ứng cho 80% hoạt động điều khiển trong công nghiệp. Do giá thành bộ nhớ liên tục giảm, các nhà sản xuất PLC trang bị bộ nhớ ngày càng lớn cho các sản phẩm của họ.

Khối vào ra:

Mọi hoạt động xử lý tín hiệu bên trong PLC có mức điện áp 5V-DC và 15V-DC (điện áp cho TTL và CMOS), trong khi tín hiệu bên ngoài có thể lớn hơn nhiều thường 24v-DC ÷ 240v-DC với dòng lớn.

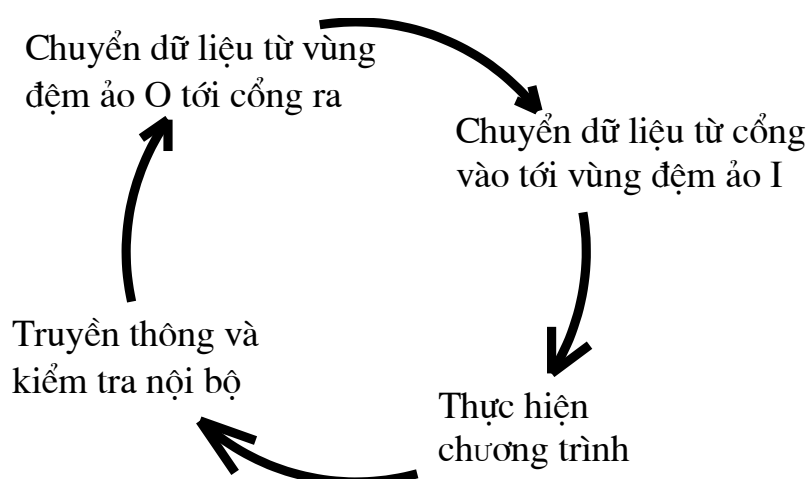
Mỗi đơn vị I/O có một địa chỉ duy nhất, trạng thái của các kênh vào/ra được chỉ thị bởi các đèn LED trên PLC, điều này làm cho việc kiểm tra hoạt động nhập xuất trở nên đơn giản và dễ dàng.

Khối vào/ra có vai trò mạch giao tiếp giữa vi mạch điện tử của PLC với các mạch công suất bên ngoài kích hoạt các cơ cấu tác động. Nó thực hiện sự chuyển đổi các mức điện áp tín hiệu và cách ly, tuy nhiên khối vào ra cho phép PLC kết nối trực tiếp với các cơ cấu tác động có công suất nhỏ cỡ 2A trở xuống, không cần các mạch công suất trung gian hay role trung gian.

1.2.2. Hoạt động của PLC

Trong quá trình làm việc, PLC vừa thực hiện chương trình vừa cập nhật các đầu vào/ra. Quá trình này được thực hiện liên tục không ngừng theo một vòng kín gọi là scan hay cycle hoặc sweep. Phần thực hiện chương trình gọi là program scan chỉ bị bỏ qua khi PLC chuyển sang chế độ Program.

Mỗi vòng quét được bắt đầu bằng giai đoạn chuyển dữ liệu từ các cổng vào số tới vùng bộ đệm ảo (giai đoạn 1). Tiếp theo là giai đoạn thực hiện chương trình (giai đoạn 2). Trong từng vòng quét, chương trình được thực hiện từ lệnh đầu tiên đến lệnh kết thúc của khối. Sau giai đoạn thực hiện chương trình là giai đoạn chuyển đổi các nội dung của bộ đệm ảo tới các cổng ra số (giai đoạn 3). Vòng quét được kết thúc giai đoạn truyền thông nội bộ và kiểm lỗi (giai đoạn 4) - Hình 1.3



Hình 1.3. Chu trình hoạt động của PLC

Thời gian cần thiết để PLC thực hiện được một vòng quét gọi là thời gian vòng quét (scan time). Thời gian vòng quét không cố định, tức là không phải vòng quét

nào cũng được thực hiện trong một thời gian như nhau. Có vòng quét được thực hiện lâu, có vòng quét được thực hiện nhanh tùy thuộc vào số lệnh trong chương trình được thực hiện, vào khối dữ liệu được truyền thông trong vòng quét đó.

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

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

1.3. BIỂU DIỄN CÁC ĐẠI LƯỢNG TRONG PLC

1.3.1. Các hệ đếm sử dụng trong PLC

Bộ xử lý trung tâm (CPU - Central Processing Unit) bên trong PLC chỉ làm việc với 2 trạng thái 0 hoặc 1 (dữ liệu số) hay ON/OFF, do đó cần thiết phải có một số cách biểu diễn các đại lượng liên tục thường gặp hàng ngày dưới dạng các dãy số 0 và 1. Trong PLC thường sử dụng các hệ đếm sau:

- Hệ nhị phân (Binary)

Là hệ đếm trong đó chỉ sử dụng 2 con số là 0 và 1 để biểu diễn tất cả các con số và đại lượng. Dãy số nhị phân được đánh số từ phải sang trái, bắt đầu từ bit 0, kể đến bit 1, rồi đến bit 2...cứ như vậy cho đến bit ngoài cùng bên trái là bit n.

Bit nhị phân thứ n có trọng số là $2^n \times 0$ hoặc 1, trong đó n = thứ tự của bit trong dãy số nhị phân, 0 hoặc 1 là giá trị của bit n đó. Giá trị của dãy số nhị phân bằng tổng trọng số của từng bit trong dãy.

Ví dụ: Dãy số nhị phân 1011 sẽ có giá trị như sau:

$$1011 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$$

- Hệ thập phân (Decimal)

Là hệ đếm sử dụng 10 chữ số là 0 1 2 3 4 5 6 7 8 9 để biểu diễn các con số. Đây là hệ đếm thường dùng hàng ngày. Hệ thập phân còn kết hợp với hệ nhị phân để có cách biểu diễn gọi là BCD (Binary - Code- Decimal).

- Hệ thập lục (Hexadecimal)

Là hệ đếm có cơ số là 16, tức là sử dụng 16 chữ số là: 0 1 2 3 4 5 6 7 8 9 A B C D E F

Ví dụ: Số 7F trong hệ 16 sẽ biểu diễn giá trị $7.16^1 + 15.16^0 = 137$

Bảng 1-1 sau đây sẽ thể hiện mối liên hệ giữa các hệ đếm.

Bảng 1-1

HEX	BCD	Số nhị phân 4 bit tương đương			
		Bit 3	Bit 2	Bit 1	Bit 0
		$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
A	-	1	0	1	0
B	-	1	0	1	1
C	-	1	1	0	0
D	-	1	1	0	1
E	-	1	1	1	0
F	-	1	1	1	1

1.3.2. Các phương pháp biểu diễn đại lượng trong PLC

- Các đơn vị dữ liệu:

Dữ liệu trong bộ PLC MITSUBISHI FX3U được mã hoá dưới dạng mã nhị phân. Mỗi chữ số được gọi là 1 bit, 8 bit liên tiếp được gọi là 1 byte, 16 bit liên tiếp được gọi là 1 Word, 32 bit liên tiếp được gọi là double word (DW).

Các đại lượng liên tục (Analog) như dòng điện, điện áp, ... khi ở trong PLC đều được đổi sang mã nhị phân.

Ngoài ra để biểu diễn những số lượng lớn hơn, người ta có thể sử dụng thêm các đơn vị sau:

+ Kilobit: Trong kỹ thuật số 1 Kilobit (viết tắt là Kb) = $2^{10} = 1024$ bit.

Tuy nhiên để tiện tính toán người ta thường dùng là 1 Kb = 1000 bit.

+ Megabit: 1 MB = 1024 KB. Người ta cũng thường tính gần đúng là:

$$1 \text{ MB} = 1000 \text{ Kb} = 1.000.000 \text{ bit.}$$

+ Kilobyte và megabyte: Tương tự như số đếm với bit nhưng các cách viết với byte là KB và MB.

+ Kiloword: 1 kWord = 1000 word

- *Biểu diễn số thập phân bằng số nhị phân.*

Trong PLC MITSUBISHI FX3U các giá trị được biểu diễn dưới dạng bit, byte (B), word (W), double word (DW). Để biểu diễn như vậy, người ta thường làm như sau: lấy số thập phân chia liên tiếp cho 2 sau đó lấy phần dư và viết theo thứ tự ngược lại. Nếu dãy này chưa đủ 16 bit thì ta chèn thêm số 0 vào.

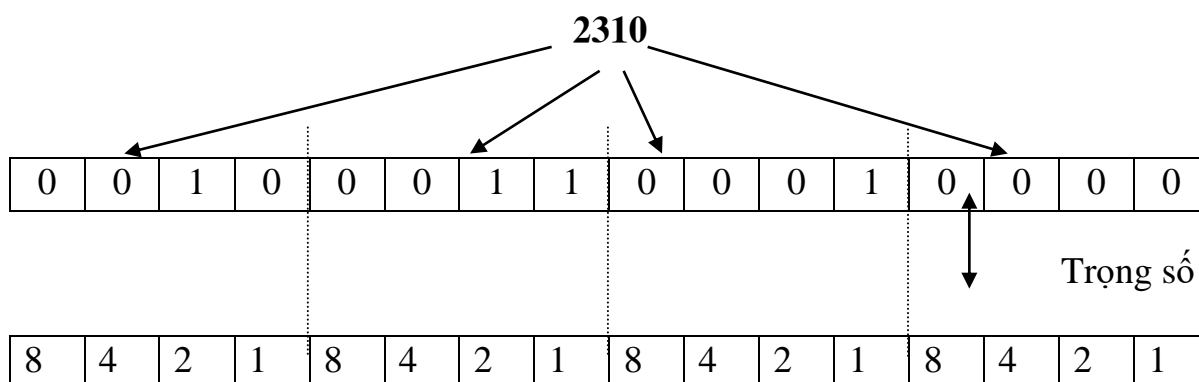
Ví dụ: số : $(17)_{10} = (0000\ 0000\ 0001\ 0001)_2$

Như trên ta thấy, việc biểu diễn một số thập phân bằng một dãy số nhị phân là rất dài và mất thời gian. Vì vậy người ta đã có một cách biểu diễn số thập phân dưới dạng đơn giản hơn. Đó là dạng BCD và được dùng phổ biến trong các loại PLC.

- *Biểu diễn số thập phân dưới dạng BCD*

Khi biểu diễn bằng mã BCD, mỗi số thập phân được biểu diễn riêng biệt bằng nhóm 4 bit nhị phân.

Ví dụ: Ta có một số hệ thập phân là 2.310 và cần chuyển nó sang dạng mã BCD 16 bit.

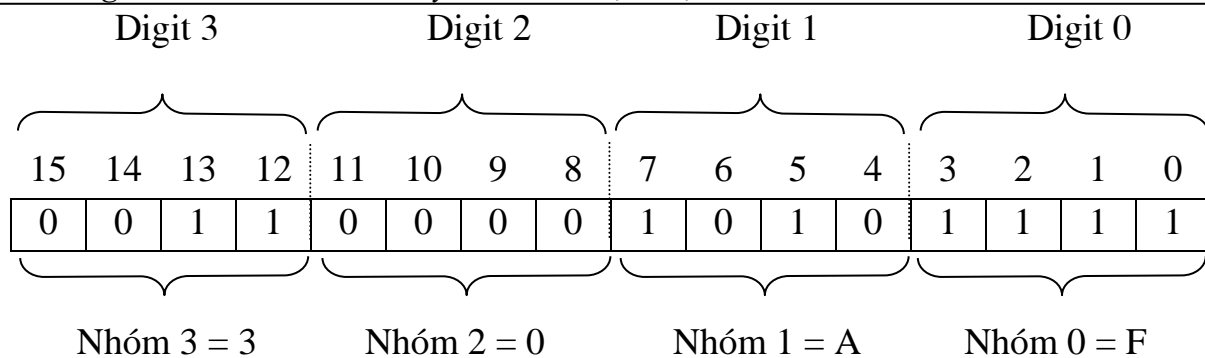


Số thập phân dưới dạng BCD:

$$(2310)_{10} = (0010\ 0011\ 0001\ 0000)_{BCD}$$

- *Biểu diễn số nhị phân dưới dạng HEXA*

Số nhị phân được biểu diễn dưới dạng HEXA bằng cách nhóm 4 bit một, bắt đầu từ phải qua trái và biểu diễn mỗi nhóm bit này bằng một chữ số (còn gọi là Digit) HEXA.



Như vậy: $0011\ 0000\ 1010\ 1111_2 = (30AF)_{16}$

Tuy nhiên, biểu diễn số thập phân dưới dạng HEXA và BCD là không hoàn toàn tương đương nhau (cho kết quả bằng dãy số nhị phân khác nhau).

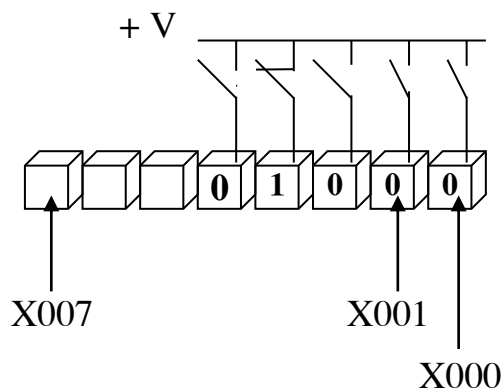
Mã BCD được dùng chủ yếu khi đổi số thập phân ra mã nhị phân dạng BCD, mã HEXA được dùng phổ biến khi biểu diễn dãy số nhị phân dưới dạng ngắn gọn hơn.

	MSB							LSB								
VW0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	VB1								VB0							

1.4. CÁC BIT ĐẦU VÀO/RA TRONG PLC VÀ CÁC THIẾT BỊ ĐIỆN BÊN NGOÀI

1.4.1. Các bit đầu vào trong PLC với các thiết bị điện bên ngoài

Các bit trong PLC phản ánh trạng thái đóng mở của công tắc điện bên ngoài như trên hình 1.4. Khi trạng thái công tắc đầu vào thay đổi (đóng/mở), trạng thái các bit tương ứng cũng thay đổi theo (tương ứng 1 hoặc 0). Trong ví dụ trên hình 1.4, các công tắc đầu vào được nối với đầu các vào Xi.

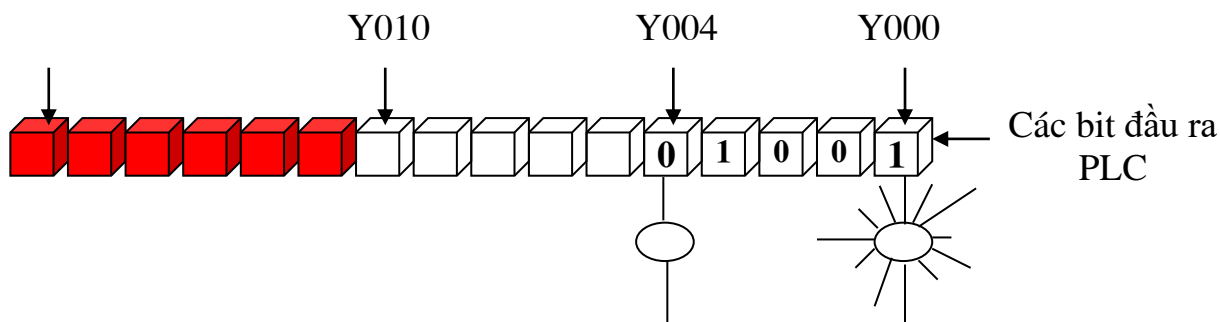


Hình 1.4. Sơ đồ đầu đầu vào

1.4.2. Các bit đầu ra trong PLC với các thiết bị điện bên ngoài

Tùy theo chủng loại PLC mà số lượng các bit ra (các bit có thể liên hệ điện với các thiết bị điện bên ngoài) trong PLC cũng khác nhau.

Trên hình 1.5 là ví dụ về các bit điều khiển đầu ra của bộ PLC MITSUBISHI FX3U. Các đầu ra Y0, Y1, ... Y7 có thể nối với các thiết bị bên ngoài và điều khiển chúng thông qua các trạng thái tương ứng với ("1" hoặc "0") của nó.



0V

Hình 1.5. Sơ đồ đầu đầu ra

1.5. CÁC ĐẶC TÍNH KỸ THUẬT CHỦ YẾU CỦA PLC

- Điện áp nguồn cung cấp:

Tùy thuộc vào từng loại PLC mà nguồn cấp có thể là 220VAC hoặc 24VDC.

- Điện áp nguồn cho đầu vào:

Thông thường là nguồn DC 24V

- Số lượng đầu vào/ra:

Phụ thuộc và chủng loại và kích cỡ của PLC. Nếu số đầu vào và ra càng nhiều thì khả năng kết nối của PLC với các thiết bị bên ngoài càng lớn.

- Công suất hay dòng điện đầu ra:

Đại lượng này đặc trưng cho khả năng chịu tải của PLC. Tuy nhiên để giảm thiểu kích thước của PLC thì hầu hết các PLC đều có dòng ra rất nhỏ, thường từ 0,1A đến 5A. Với giá trị này thì PLC không thể kết nối trực tiếp với các phụ tải công suất lớn được. Tuy nhiên các phụ tải công suất lớn vẫn có thể kết nối với PLC thông qua các thiết bị điện trung gian như rơ le, công tắc tơ...

- Dạng dữ liệu đầu ra:

Thông thường có 2 dạng chính là đầu ra kiểu rơ le (còn gọi là đầu ra tiếp điểm) và đầu ra Transistor (đầu ra phi tiếp điểm). Thường thì đầu ra dạng rơ le cho khả năng cung cấp dòng điện lớn hơn.

- Dung lượng bộ nhớ:

Đặc trưng cho khả năng lưu giữ dữ liệu và chương trình của PLC. Ví dụ bộ PLC MITSUBISHI FX3U có 4096 word chương trình và 2560 word dữ liệu.

- Tốc độ CPU:

Đặc trưng cho khả năng xử lý dữ liệu của PLC là nhanh hay chậm. Tốc độ này ảnh hưởng đến thời gian thực hiện lệnh. Ví dụ: PLC MITSUBISHI FX3U

FX3U thì thời gian thực hiện 1 lệnh logic là $0,37\mu s$, thời gian thực hiện 1 lệnh di chuyển từ là: $34\mu s$, thời gian thực hiện 1 lệnh Timer/Counter là $50 - 64 \mu s$...

1.6. THỦ TỤC THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN SỬ DỤNG BỘ ĐIỀU KHIỂN PLC

Để thiết kế hệ thống điều khiển tự động sử dụng bộ điều khiển PLC cần tuân theo các thủ tục sau:

- Phân tích quá trình công nghệ

Đây là bước quan trọng rất cần thiết cho các nhà lập trình. Đầu tiên, người lập trình phải đi tìm hiểu, phân tích hoạt động của hệ thống điều khiển (quá trình công nghệ), thứ tự hoạt động cũng như vai trò của từng thiết bị trong hệ thống.

Để đơn giản hoá, chúng ta cần mô phỏng quá trình công nghệ qua lưu đồ.

- Xác định và lựa chọn thiết bị vào và ra

Ở bước này người lập trình phải xác định và lựa chọn thiết bị vào ra. Tuy nhiên, một số thiết bị đã có sẵn mà chỉ yêu cầu người lập trình viết chương trình để điều khiển sự hoạt động của nó.

- Phân định những đầu vào và đầu ra

Tất cả các thiết bị đầu vào và đầu ra cần được lập trình hoá phải được xác định thông qua một địa chỉ tương ứng trong PLC. Việc này do người lập trình phân định. Những thiết bị đầu vào phản ánh nội dung dữ liệu cần xử lý như: Các công tắc, thiết bị cảm biến..., những thiết bị đầu ra là những thiết bị thừa hành như các van điện từ, động cơ điện, đèn chỉ báo ...

- Viết chương trình

Tiếp theo, việc viết chương trình dưới dạng sơ đồ LAD (Ladder) hoặc STL (Statement list) thông qua thứ tự thao tác của hệ thống điều khiển như đã xác định, theo từng bước một.

- *Mô phỏng chương trình*: Sử dụng công cụ mô phỏng để kiểm tra chương trình lập trình.

- Nạp chương trình vào trong bộ nhớ

Từ chương trình đã viết và các đầu vào/ra đã xác định. Chúng ta truy nhập chương trình vào trong bộ nhớ bằng bàn phím lập trình hoặc bằng máy vi tính với sự trợ giúp của công cụ phần mềm. Sau khi hoàn chỉnh phần lập trình, ta kiểm tra lỗi mã hoá bằng công cụ có chức năng chuẩn đoán và nếu có thể được thì mô phỏng toàn bộ thao tác để thấy rằng nó đã hoạt động được như mong muốn.

- Kết nối thiết bị.

Việc đấu nối thiết bị sẽ phải tuân thủ bảng phân định vào/ra. Tuy nhiên, ở bước này người ta chỉ kết nối các thiết bị vào để lấy dữ liệu cho bước hoạt động thử. Chỉ khi nào hoàn chỉnh bước lập trình người ta mới kết nối thiết bị một cách hoàn chỉnh.

- Chạy thử

Để đảm bảo cấu trúc chương trình và các tham số đã cài đặt là chính xác trước khi đưa vào hệ điều khiển. Ta cần phải chạy thử chương trình điều khiển.

Nếu có lỗi hoặc chưa hợp lý thì sửa. Khi chạy thử chương trình điều khiển tốt, ta ghép nối với đối tượng và hoàn chỉnh chương trình theo hoạt động của hệ thống.

- *Viết tài liệu chuyển giao công nghệ.*

2. GIỚI THIỆU PLC MISUBISHI dòng FX-3U

2.1. CÁC DÒNG PLC PHỔ BIẾN CỦA MISUBISHI

- Một số dòng phổ biến:

+ FX0 – FX0S

+ FX1s , FX1N, FX2N

+ FX3G, FX3u, FX3uc

+ FX5u

- Dòng FX3u là thế hệ thứ ba, tốc độ xử lý nhanh (0.065ms/1 lệnh) logic, kèm theo 209 tập lệnh được tích hợp sẵn và cải tiến liên tục thuận lợi cho việc điều khiển vị trí.

- Khả năng kết nối tối đa lên đến 384 I/O gồm cả các khối I/O qua mạng cho phép mở rộng qua cổng truyền thông



Hình 1.6. PLC Mitsubishi dòng FX

2.2. PHÂN LOẠI VÀ CÁCH NHẬN BIẾT KÝ MÃ HIỆU

- Theo tiếp điểm đầu ra, chia làm 02 loại:

+ MR (đầu ra role)

+ MT (đầu ra Transistor)

- Theo số ngõ vào ra 16/32/48/80/124 IO

- Theo nguồn cấp có loại 220 VAC và 24 VDC

+ 220 VAC kí hiệu là E

+ 24 VDC kí hiệu là D

- Theo kiểu Transistor đầu ra có 02 loại

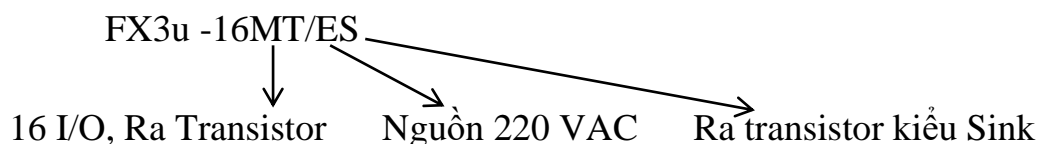
+ Transistor Sink – S

+ Transistor Source - SS

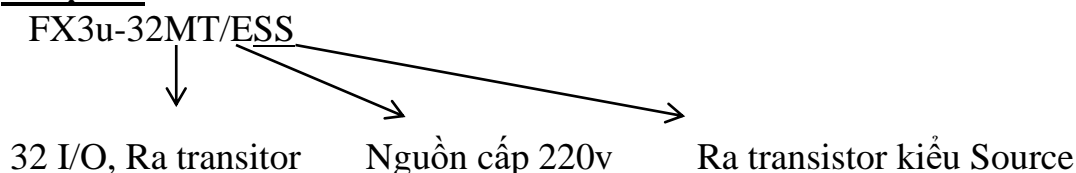
Ví dụ 01:



Ví dụ 02:

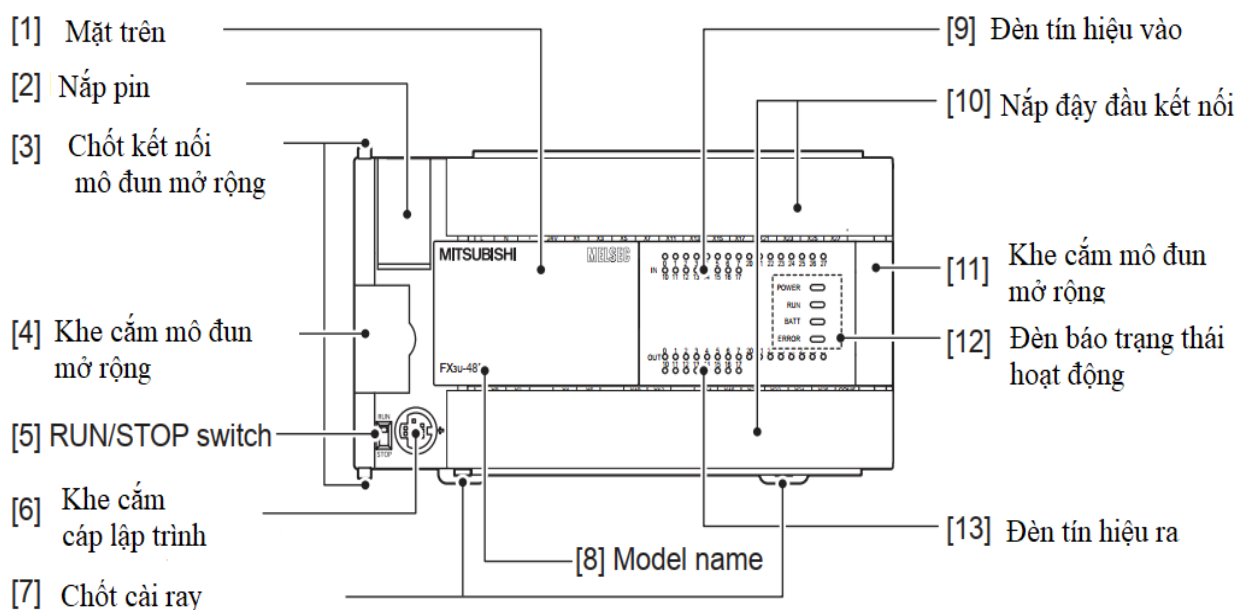


Ví dụ 03:



2.3. ĐẶC TÍNH KỸ THUẬT CỦA PLC MITSUBISHI FX3U:

Hiện nay, MITSUBISHI được coi là một trong những hãng điện tử hàng đầu thế giới về công nghệ tự động hoá. Các thiết bị tự động của MITSUBISHI có chất lượng cao, được sản xuất với công nghệ mới nhất và rất đa dạng: Từ công tắc, rơ le các loại, bộ định giờ, bộ đếm, cảm biến, kiểm soát nhiệt độ,... cho tới các thiết bị điều khiển logic lập trình - PLC. MITSUBISHI sản xuất nhiều loại bộ điều khiển lập trình (PLC) khác nhau, nhưng thông dụng nhất là loại MITSUBISHI FX3u. Đặc tính kỹ thuật của PLC MITSUBISHI FX3U như sau:



Hình 1.7. Mặt trước và các bộ phận chính của PLC FX3U

- Điện áp nguồn cung cấp: AC 85 ÷ 264V hoặc DC 20,4÷28,8V
- Điện áp nguồn cho đầu vào: DC 24V
- Số lượng đầu vào/ra: tùy theo loại PLC và có khả năng thêm 10 module mở rộng. Có thể mở rộng lên tới 256 I/Os thông qua module hoặc 384 I/O thông qua mạng CC-Link.
- Dòng điện đầu ra: 0,7A với loại ra transisto, 2A với loại ra Relays.
- Dung lượng bộ nhớ: 64000 bước
- Các chế độ làm việc: Có 2 chế độ làm việc:
 - + RUN: Chế độ PLC thực hiện chương trình trong bộ nhớ.
 - + STOP: Cưỡng bức PLC dừng chương trình đang chạy và chuyển sang chế độ STOP. PLC sẽ tự động chuyển từ RUN sang STOP nếu có sự cố hoặc trong chương trình gặp lệnh STOP.
- Kết nối truyền thông: hỗ trợ kết nối RS232, RS485, USB, Ethernet, profibus, CAN, CClink.
- Cấp lập trình: là loại FX-USB-AW (USB-RS-422 converter)
- Số lượng timer: 512 chia làm 3 loại với độ phân giải khác nhau: 1ms, 10ms và 100ms.
- Số lượng bộ đếm (counter): 256 chia làm 2 loại: thông thường (235) và bộ đếm tốc độ cao (11).
- Phát xung tốc độ cao: max 100kHz, lên tới 200kHz hoặc 1Mhz với module chức năng.
- Lập trình dạng ngôn ngữ bậc thang (LAD - Ladder) hoặc sơ đồ chức năng tuần tự (SFC - Sequential Function Chart) bằng phần mềm Gx-Works 2.
- Có đầu nối đất bảo vệ (Protective Earth Terminal) để tránh điện giật.
- Đầu nối nguồn cấp DC ra từ PLC (DC Power Supply Output Terminal).
- Điện áp ra chuẩn là DC 24V với dòng định mức là 0.3A có thể được cung cấp cho các đầu vào số DC.
- Các đèn LED chỉ thị trạng thái của PLC (PC Status Indicators)- Bảng 2-1

Bảng 2-1

Đèn	Trạng thái	Chức năng
POWER	Màu xanh	Có nguồn vào PLC
RUN	Màu đỏ	PLC đang STOP
	Màu xanh	PLC đang RUN
BATT.V	Tắt	Pin bình thường
	Sáng	Hết pin
ERROR	Nháy đỏ	Lỗi chương trình
	Sáng đỏ	Lỗi CPU

- Các đèn chỉ thị trạng thái đầu vào IN (Input Indicator). Đèn LED trong nhóm này sẽ sáng khi đầu vào tương ứng lên ON.
- Các đèn chỉ thị trạng thái đầu ra OUT (Output Indicator). Các đèn LED này sẽ sáng khi đầu ra tương ứng được bật.

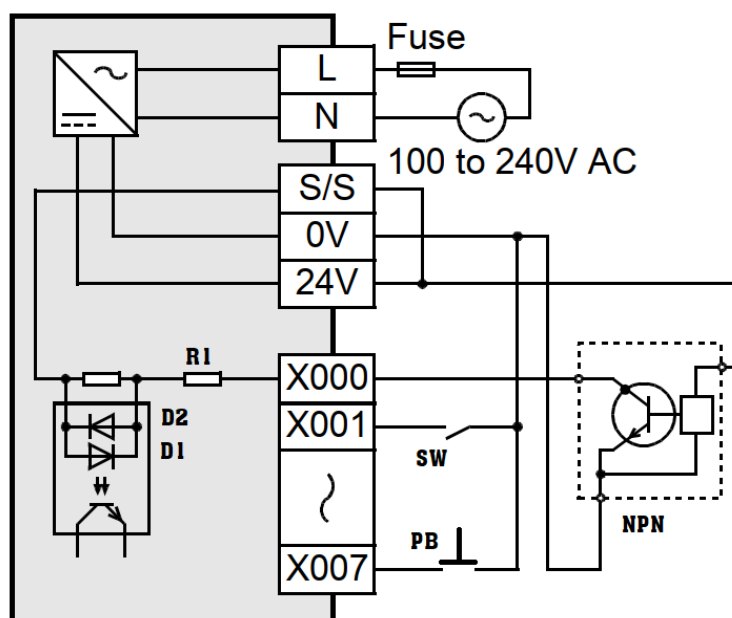
2.4. SƠ ĐỒ KẾT NỐI I/O PLC MISUBISHI FX3U

2.4.1. Nối INPUT

a. Nối INPUT kiểu SINK

- Nguyên tắc nối đầu vào điều khiển SINK:

- + Chân S/S nối 24 VDC
- + Các tín hiệu vào như nút nhấn, công tắc, transistor NPN hở sẽ nối chung 0V



Hình 1.8. Sơ đồ nối Input kiểu SINK

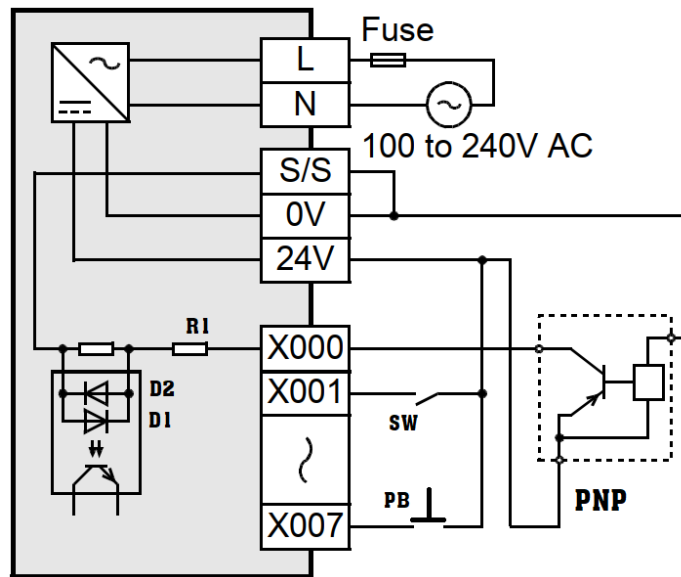
- Giải thích:

- + Cấp nguồn 220 VAC vào L và N, hoặc loại 24 VDC vào (+) và (-) tùy thuộc vào PLC loại E hay D. Với loại 220v thì PLC sẽ tự xuất ra 24 VDC.
- + Nối chân S/S với 24V, đi ốt D1 làm việc và các đầu vào X0 (tương tự với X1, X2,...) sẽ “chờ sẵn” ở mức cao (24V). Khi đó các đầu vào có thể nối qua công tắc (SW), nút ấn (PB), Transistor NPN như **hình 1.8**
- + Khi nhấn SW thì sẽ có dòng điện chạy từ chân S/S (24 VDC) qua D1 → R1 → X1 → SW → 0V, PLC hiểu rằng có tín hiệu ở đầu vào X1, đèn báo tương ứng sẽ sáng lên.
- + Tương tự khi sử dụng các tín hiệu đầu vào là nút nhấn PB hoặc transistor NPN hở.

b. Nối INPUT kiểu SOURCE

- Nguyên tắc đấu nối:
- + Chân S/S nối 0V.

+ Các tín hiệu vào như nút nhấn, công tắc, transistor PNP hở sẽ nối chung 24VDC.



Hình 1.9. Đấu nối đầu vào kiểu Source

- Giải thích:

+ Nối chân S/S với 0V, Diốt D2 làm việc đầu vào X0 (tương tự với X1, X2...) “chờ sẵn” ở mức thấp (0 VDC). Các đầu vào nối với công tắc, nút ấn, Transistor PNP như hình 1.

+ Khi nhấn SW thì sẽ có dòng điện chạy từ chân 24 VDC → SW → X1 → R1 → D2 → S/S (0V), PLC hiểu rằng có tín hiệu ở đầu vào X1, đèn báo tương ứng sẽ sáng lên.

+ Tương tự khi sử dụng các tín hiệu đầu vào là nút nhấn PB hoặc transistor PNP hở.

2.4.2. Nối OUTPUT

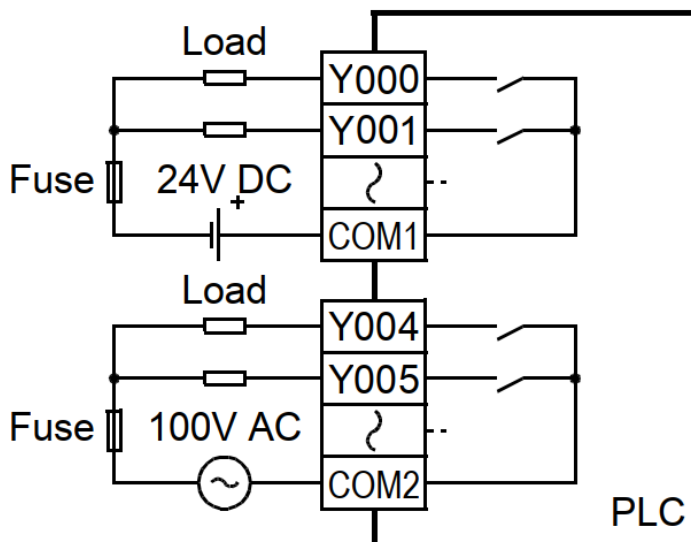
Chỉ phân biệt cách nối OUTPUT kiểu SINK hay SOURCE đối với loại PLC có đầu ra dạng transistor.

a. Nối OUTPUT cho PLC có đầu ra dạng Rơ le

- Mỗi chân COM sẽ là chân chung cho 4 đầu ra, ví dụ COM1 là chân chung của Y0, Y1, Y2, Y3.

- Có thể cấp các nguồn khác nhau vào các chân COM. Khi cấp nguồn 24VDC vào chân COM thì nên dùng nguồn ngoài, không nên dùng nguồn 24VDC xuất ra của PLC vì nguồn này có dòng nhỏ.

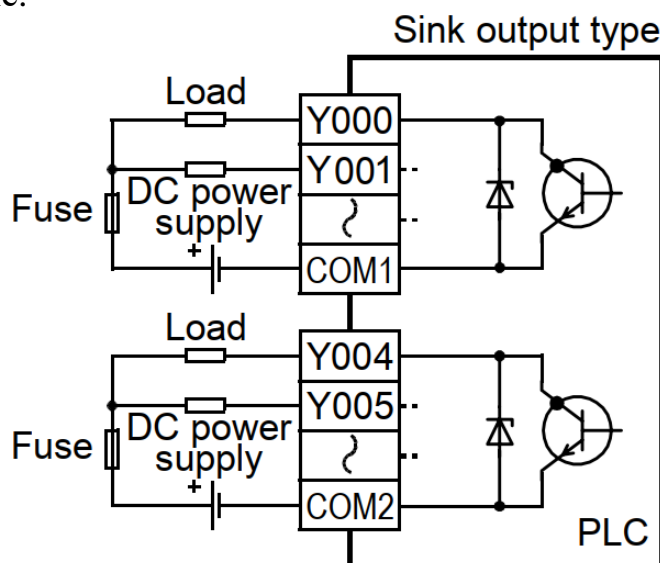
- Khi chân đầu ra Yx đóng sẽ đóng mạch điện cho tải.



Hình 1.10. Đầu nối đầu ra cho PLC có đầu ra Rơ le

b. Nối OUTPUT kiểu SINK (PLC ra Transisto)

- Đầu ra của PLC là Transistor kiểu SINK (NPN) nên khi dùng ta phải đấu như hình 1.11 cụ thể:

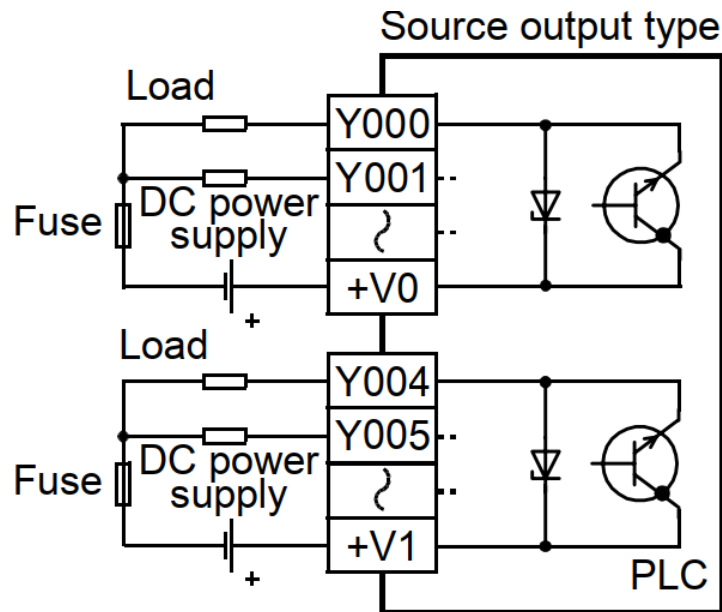


Hình 1.11. Đầu nối đầu ra kiểu SINK

- + Nguồn (+) được nối sẵn với tải, khi đó điện áp ở các điểm đầu ra Y0, Y1, Y2.... bằng điện áp (+) nguồn (đối với COM1)
- + Chân COM1 được nối vào “0v”
- + Khi đầu ra Y0 ON, sẽ có dòng điện chạy qua tải
- + Nguồn U+ độc lập với nguồn 24v DC xuất ra từ PLC nhưng cũng có thể mượn nguồn 24v của PLC (lưu ý nguồn 24v của PLC có dòng nhỏ...)
- + Không cần nối chân COM vào “0v” của nguồn 24v PLC khi ta dùng nguồn độc lập từ bên ngoài (còn dùng khi 24v PLC thì phải nối COM1 vào “0v”)

c. Nối OUTPUT kiểu SOURCE

- Đầu ra của PLC là Transistor kiểu PNP nên khi dùng ta phải đấu như hình 1.12 cụ thể:



Hình 1.12. Đầu nối đầu ra kiểu SOURCE

+ Nguồn (-) được nối sẵn với tải, khi đó điện áp ở các điểm đầu ra Y0, Y1, Y2.... bằng điện áp (-) nguồn

+ Chân +V0 được nối với (+)

+ Khi đầu ra Y0 ON, sẽ có dòng điện chạy qua tải

- **Chú ý:**

+ Với mỗi PLC có thể chọn đầu nối đầu vào kiểu SINK (S/S nối 24v) hoặc Source (S/S nối 0v) mà không ảnh hưởng tới câu lệnh lập trình.

+ Nhưng đầu ra Transistor sẽ mặc định sẵn là SINK hay SOURCE do nhà sản xuất và được ghi bởi các ký hiệu trên PLC.

2.5. PHẦN MỀM LẬP TRÌNH CHO PLC Mitsubishi dòng FX

- Phần mềm sử dụng để lập trình là Gx-Works 2.

- Chương trình lập trình: kiểu bậc thang (LAD) hoặc kiểu khối chức năng (SFC).

- Trình tự thực hiện lập trình (chi tiết tại bài thực hành số 2):

+ Khởi động phần mềm Gx-Works 2

+ Tạo Project

+ Soạn thảo chương trình: Sử dụng các phím tắt hoặc chuột để chọn các lệnh, hàm chức năng

+ Biên dịch chương trình sau khi soạn: nhấn Shift + Alt + F4 để biên dịch hoặc vào Compile/Rebuild All hoặc nhấn F4 sau khi sửa chương trình:

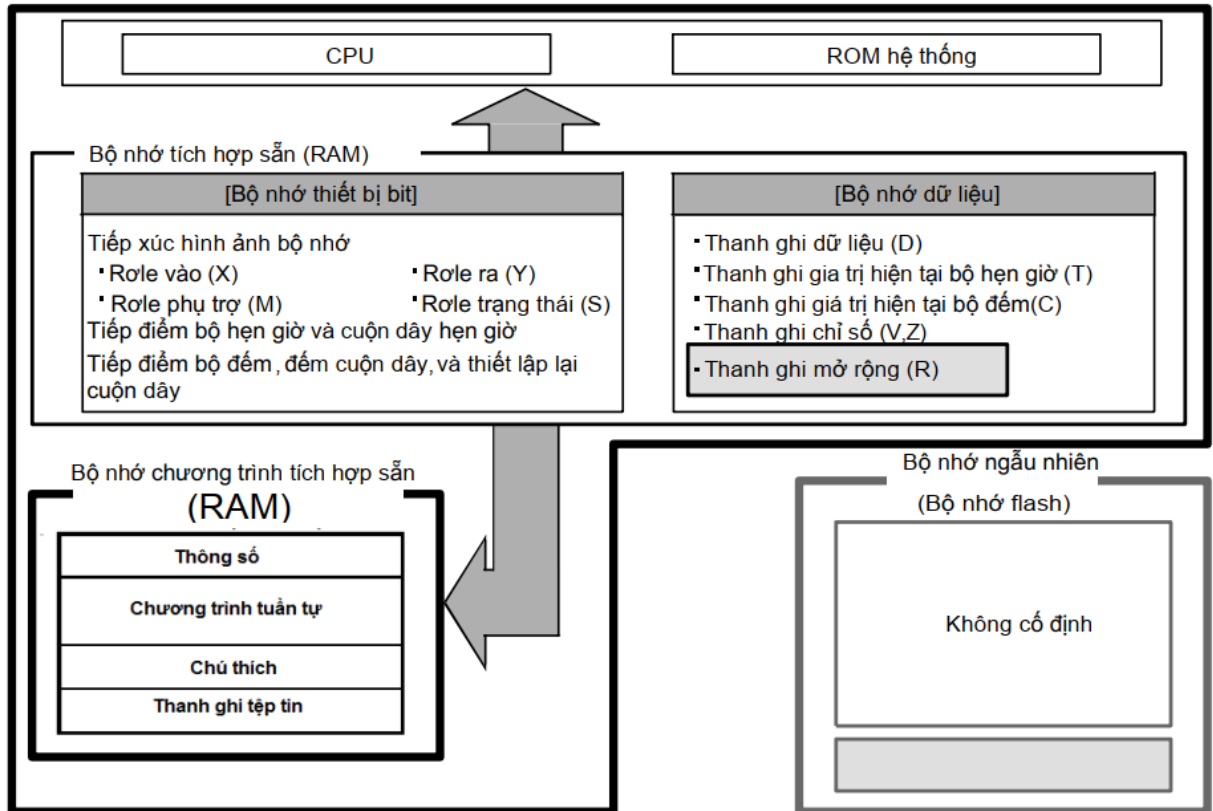
+ Kiểm tra kết nối PLC với máy tính

+ Dowload chương trình PLC

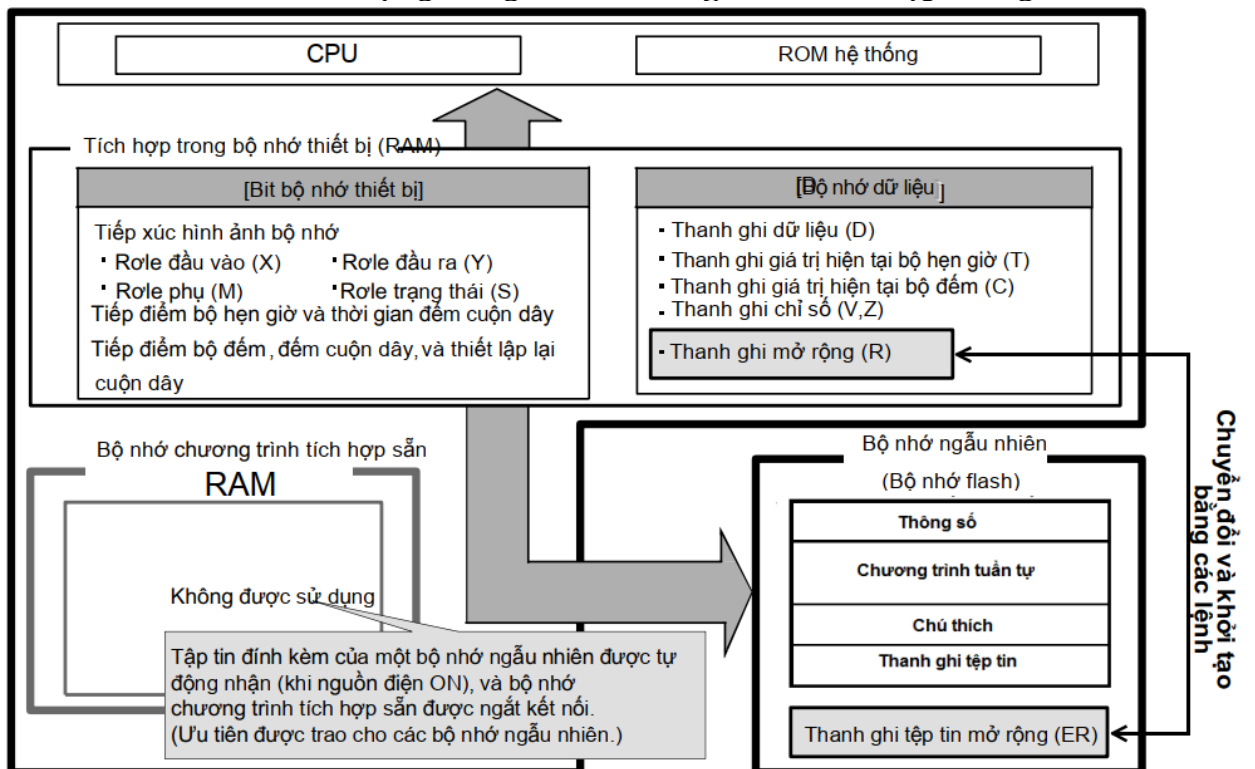
+ Giám sát, kiểm tra chương trình online

2.6. CẤU TRÚC BỘ NHỚ, CÁC VÙNG NHỚ VÀ ĐỊA CHỈ BỘ NHỚ TRONG MITSUBISHI FX3U:

2.6.1. Cấu trúc bộ nhớ:



Hình 1.13: Sử dụng vùng nhớ chương trình tích hợp trong PLC



Hình 1.14. Sử dụng vùng nhớ Flash bên ngoài

Bộ nhớ trong MITSUBISHI FX3U được chia thành 3 vùng với 1 tụ có nhiệm vụ duy trì dữ liệu trong một khoảng thời gian nhất định khi mất nguồn. Bộ nhớ của MITSUBISHI FX3U có tính linh hoạt cao. Có thể đọc, ghi trong toàn vùng, trừ các bit nhớ đặc biệt SM (Special Memory) chỉ có thể truy cập để đọc.

Các vùng nhớ trong PLC MITSUBISHI FX3U bao gồm:

- **Vùng chương trình:** là nơi lưu giữ các thông số, chương trình tuần tự, các ghi chú, thanh ghi tập tin của chương trình. Vùng này có thể đọc/ghi được.

- **Vùng nhớ thiết bị bit:** vùng này có thể đọc/ghi được, bao gồm:

+ Đầu vào số (X)

+ Đầu ra số (Y)

+ Rơ le trung gian (M)

+ Rơ le trạng thái (S)

+ Tiếp điểm bộ hẹn giờ và cuộn dây bộ hẹn giờ (T)

+ Tiếp điểm bộ đếm, đếm cuộn dây, và thiết lập lại cuộn dây (C)

- **Vùng dữ liệu:** là miền nhớ động, dùng để cất dữ liệu của chương trình. Nó có thể được truy cập theo từng bit, byte, từ đơn (W-Word) hoặc từ kép (DW-Double Word). Vùng dữ liệu được chia thành từng miền nhớ nhỏ với công dụng khác nhau. Chúng được ký hiệu bằng chữ cái đầu theo tên tiếng Anh, đặc trưng cho công dụng riêng của chúng như sau:

+ Thanh ghi dữ liệu (D – 16 bit)

+ Thanh ghi giá trị hiện tại bộ định thời (T - 16 bit)

+ Thanh ghi giá trị hiện tại bộ đếm (C - 16 bit)

+ Thanh ghi chỉ số (V, Z)

+ Thanh ghi mở rộng (R -16 bit)

- **Hàng số (K)(H)(E):**

+ “K”: là giá trị nguyên

+ “H”: là giá trị trong hệ Hexa (hệ cơ số 16)

+ “E”: là giá trị thực (dữ liệu dấu phẩy động)

2.6.2. Địa chỉ bộ nhớ trong PLC MITSUBISHI FX3U:

a. Địa chỉ bit

- Định địa chỉ I/O:

+ Đầu vào: X000, X001,..., X007; X010, X011,..., X017;...

+ Đầu ra: Y000, Y001,..., Y007; Y010, Y011,..., Y017;...

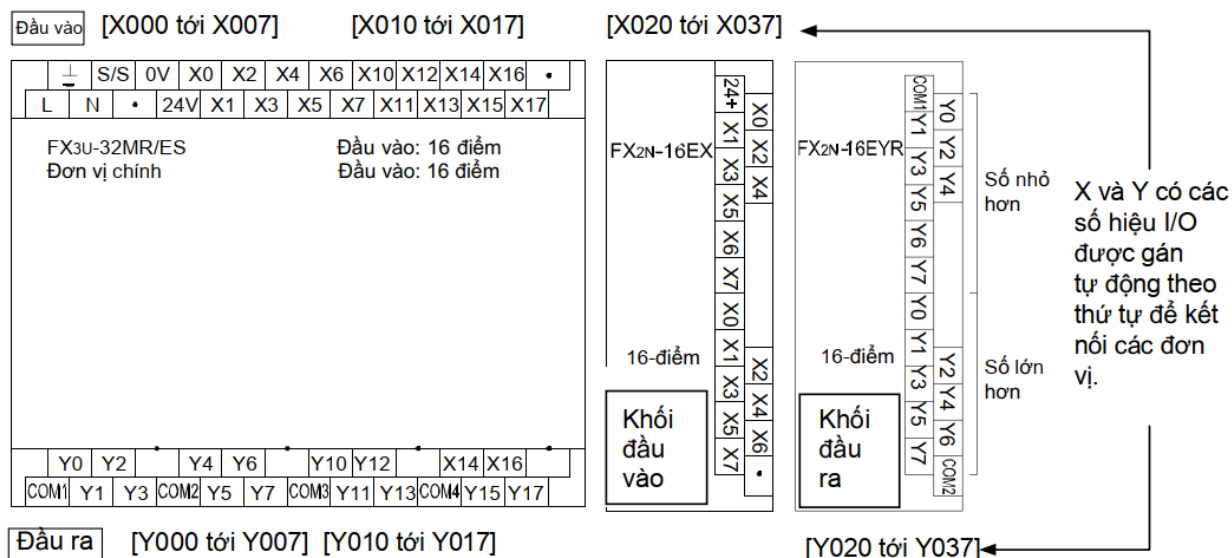
- Khi mở rộng nhiều modul I/O:

+ Số hiệu của các I/O sẽ được tự động gán tuần tự theo thứ tự tăng dần như trên hình vẽ.

- Trong trường hợp PLC có số I/O không phải là bội của 8 Ví dụ 14/10:

+ Đầu vào: X000 tới X007, X010 tới X015;

+ Đầu ra: Y000 tới Y007, Y010 tới Y011
 Các số hiệu còn lại: X016, X017 và Y012 đến Y017 cũng coi như đã được chiếm giữ. Số hiệu các I/O của các mô đun mở rộng sẽ bắt đầu từ X020 và Y020.



Hình 1.15. Địa chỉ bit vào ra

- Cách địa chỉ hóa bit cho thanh ghi:
- + Cú pháp: **[Tên vùng nhớ] [chỉ số] . [Thứ tự của bit]**
- + Ví dụ: D0.1 sẽ truy cập bit thứ 2 trong thanh ghi D0

b. Định địa chỉ thanh ghi

- Cú pháp: **[Tên vùng nhớ][chỉ số]**
- Ví dụ: D0, D1, D1000, R2000....

2.7. NỐI GHÉP GIỮA PLC VÀ THIẾT BỊ NGOẠI VI

PLC dòng FX3u hỗ trợ kết nối với các thiết bị ngoại vi qua các chuẩn kết nối RS232, RS485, USB, Ethernet, CAN, CClink, bằng cách sử dụng cổng kết nối sẵn có trên PLC hoặc các mô đun mở rộng ghép với PLC qua adapter bus.

Để PLC có thể giao tiếp được với các thiết bị ngoại vi qua cổng Peripheral Port, cần có một bộ chuyển đổi RS-232/RS-485 bên ngoài.

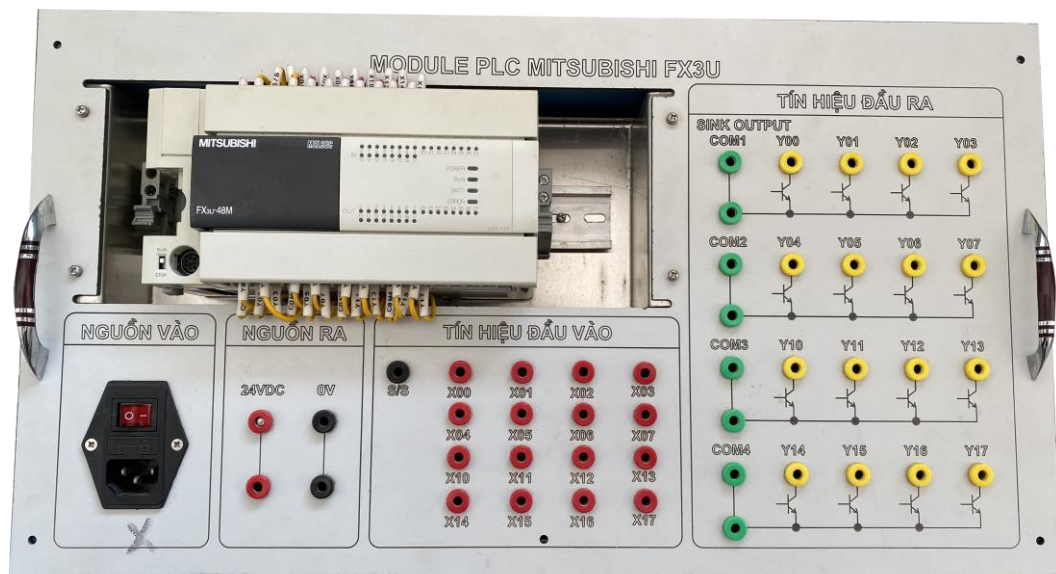
Các thiết bị ngoại vi có thể kết nối với PLC qua các chuẩn truyền thông như: HMI, bộ điều khiển nhiệt độ, biến tần,...

3. GIỚI THIỆU BỘ THỰC HÀNH PLC MITSUBISHI

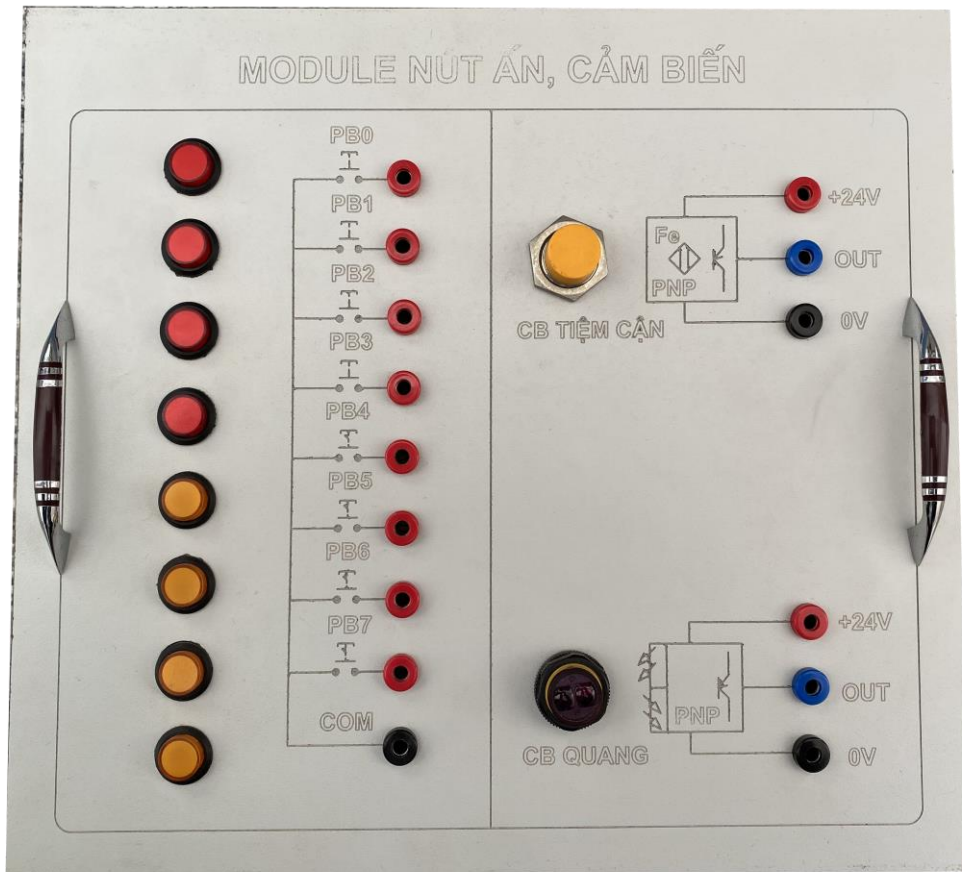


Hình 1.16. Bộ thực hành PLC Mitsubishi

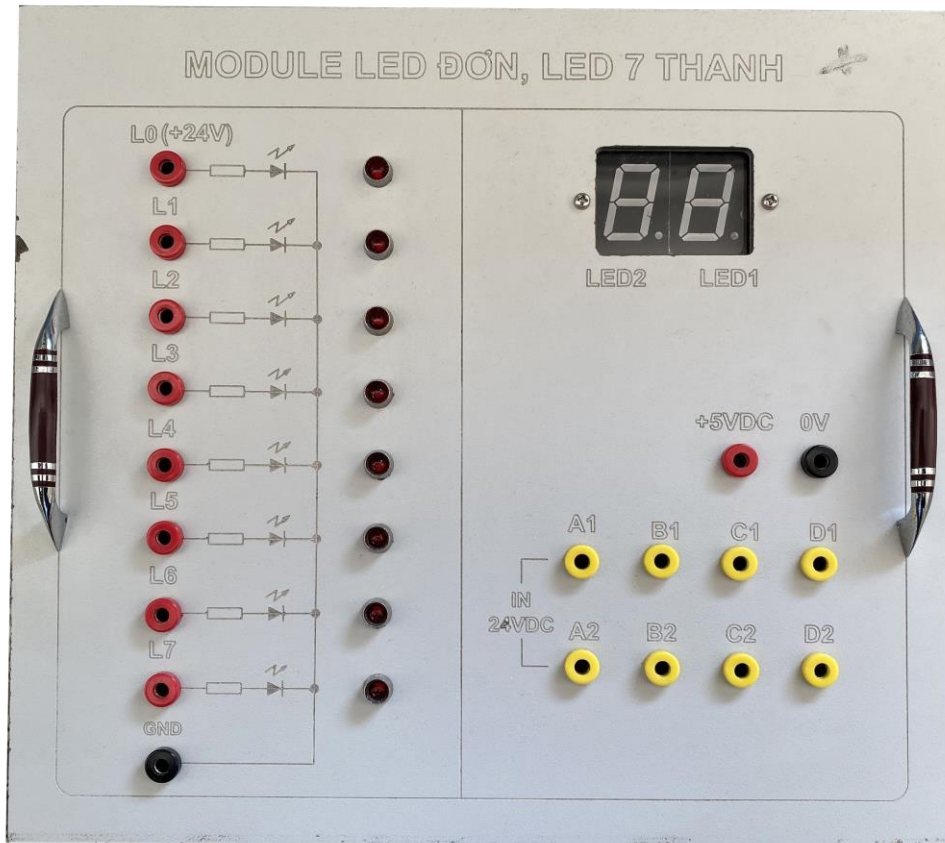
Bàn thực hành bao gồm các mô đun:



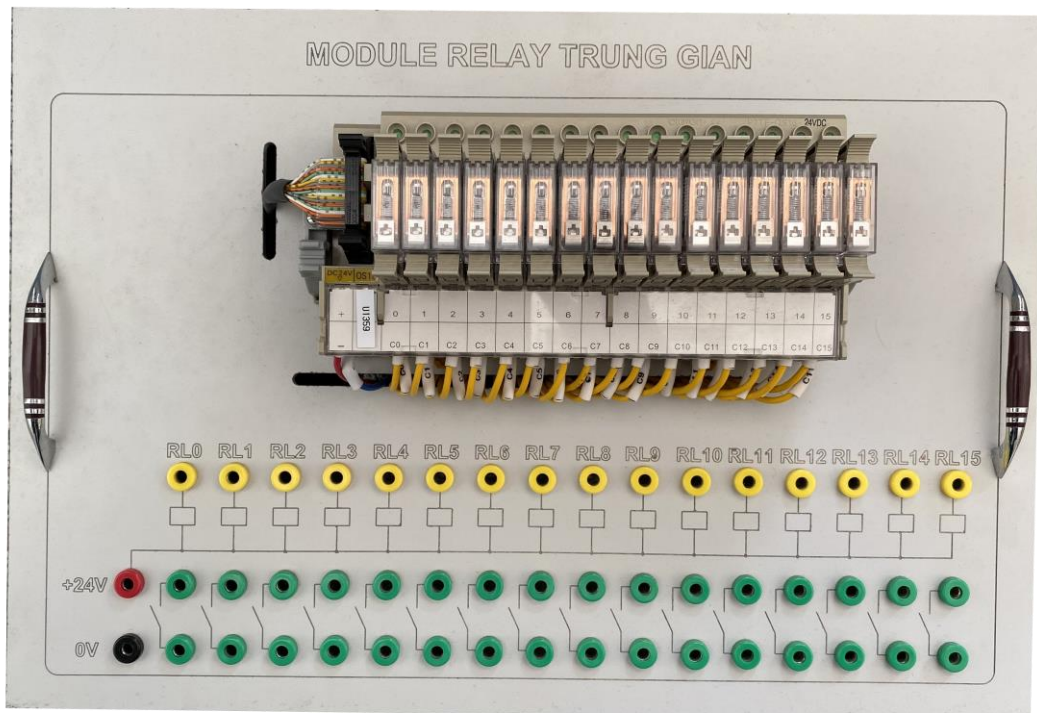
Hình 1.17-a. Module PLC



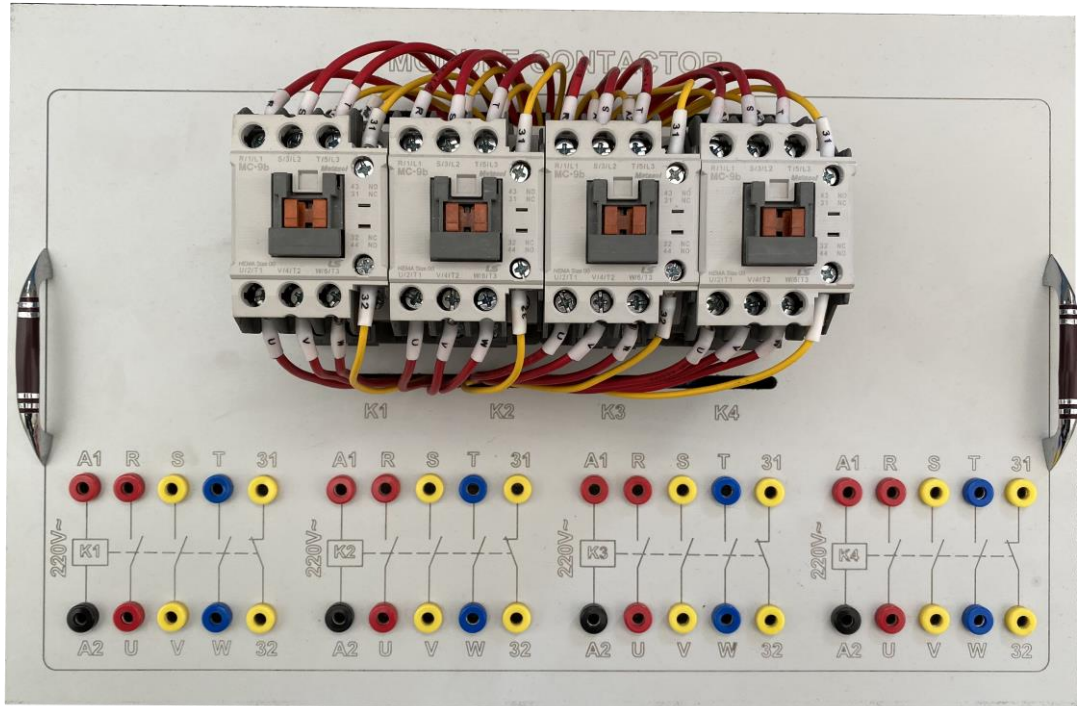
Hình 1.17-b. Module nút nhấn, cảm biến



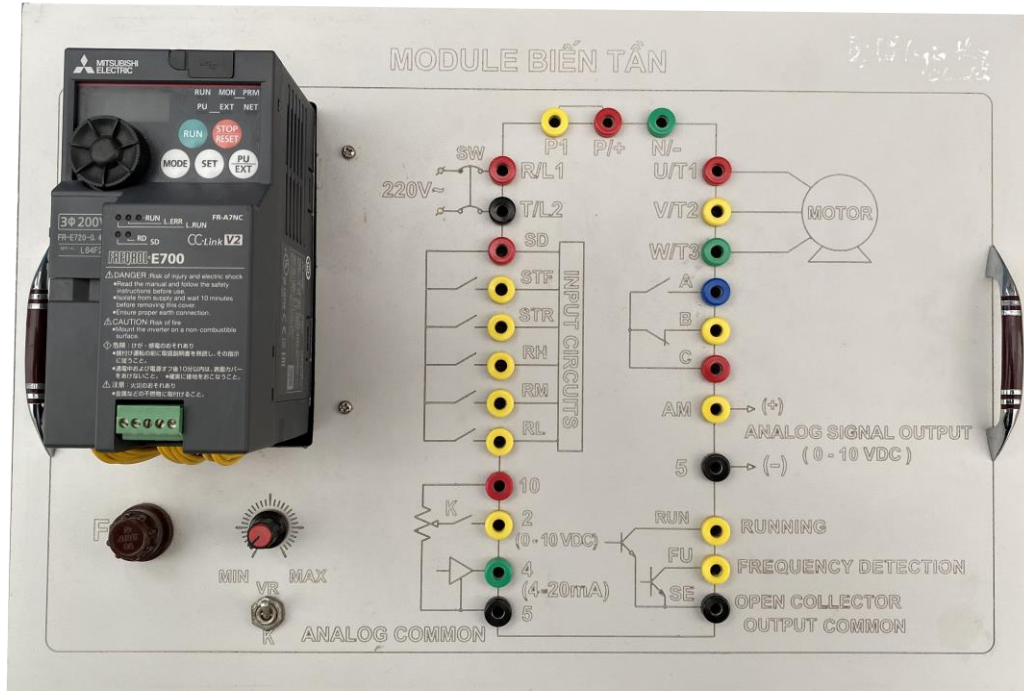
Hình 1.17-c. Module Led đơn, Led 7 thanh



Hình 1.17-d. Module Rơ le trung gian



Hình 1.17-e. Module Contactor



Hình 1.17-f. Module biến tần

PHẦN II: HƯỚNG DẪN THỰC HÀNH

Bài thực hành số 1

ĐẤU NỐI PHẦN CỨNG PLC MITSUBISHI FX3U

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Nhận biết được ký mã hiệu và phân loại được các dòng PLC Mitsubishi FX3u, để có phương án đấu nối đúng.
- Trình bày được trình tự đấu nối đầu vào, đầu ra cho PLC theo các cách đấu nối Sink/Source.

Về kỹ năng:

- Vẽ được sơ đồ đấu nối cho PLC.
- Đấu nối được đúng nguồn cung cấp, đầu vào đầu ra cho PLC.
- Kiểm tra và khắc phục được một số sai hỏng thường gặp khi đấu nối phần cứng cho PLC.

Về năng lực tự chủ và trách nhiệm:

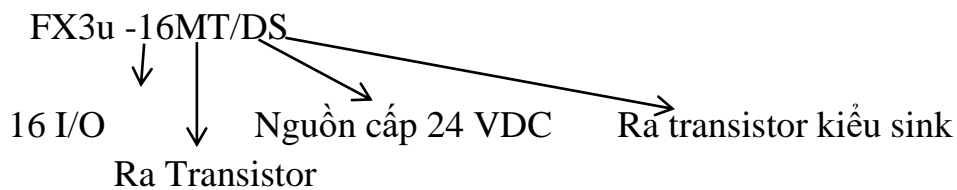
- Có tác phong công nghiệp, thao động tác chính xác, chuẩn chỉ khi đấu nối.
- Có khả năng làm việc độc lập và, tự đánh giá được kết quả thực hành của bản thân.

II. TÓM TẮT LÝ THUYẾT

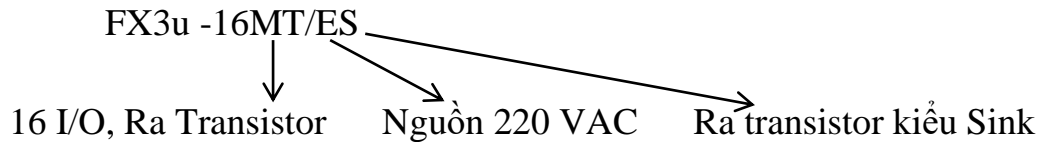
1. Nhận biết ký mã hiệu của PLC

- Phân loại theo tiếp điểm đầu ra, chia làm 02 loại:
 - + MR (đầu ra role)
 - + MT (đầu ra Transistor)
- Phân loại theo số ngõ vào ra 16/32/48/80/124 IO
- Phân loại theo nguồn cấp có loại 220 VAC và 24 VDC
 - + 220 VAC kí hiệu là E
 - + 24 VDC kí hiệu là D
- Phân loại theo kiểu Transistor đầu ra có 02 loại
 - + Transistor Sink – S
 - + Transistor Source - SS

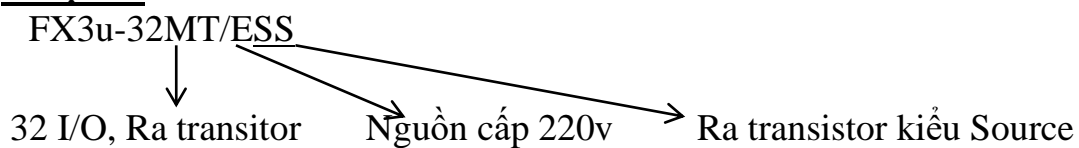
Ví dụ 01:



Ví dụ 02:



Ví dụ 03:



2. Đấu nối phần cứng (chi tiết tại Phần I, mục 2.4)

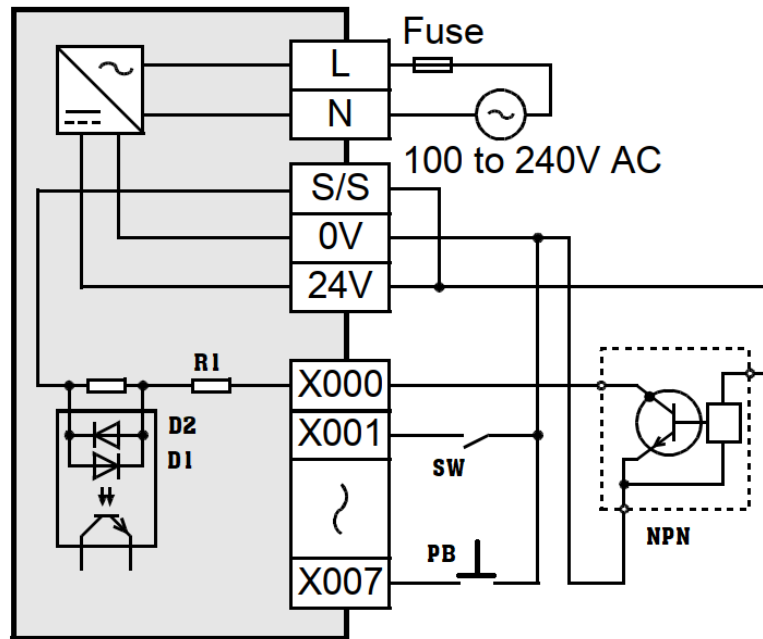
III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

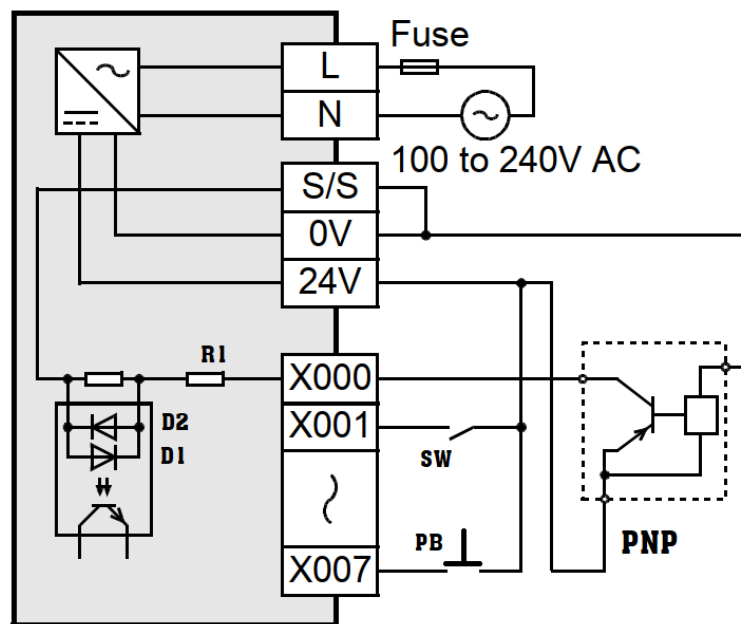
- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	PLC Mitsubishi dòng FX3u	1 bộ
2	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
3	Dây điện (1.0mm)	5m
4	Bộ đồ nghề dụng cụ	1 bộ
5	Máy tính cài đặt sẵn phần mềm	1 bộ

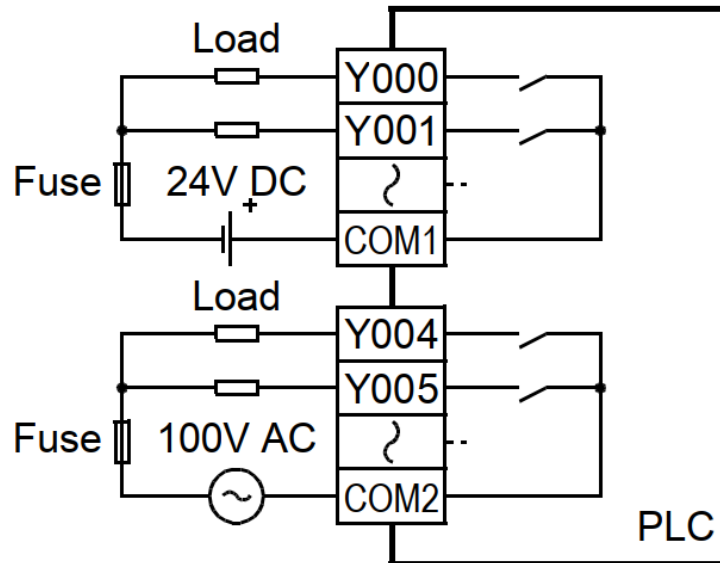
2. Sơ đồ thực hành



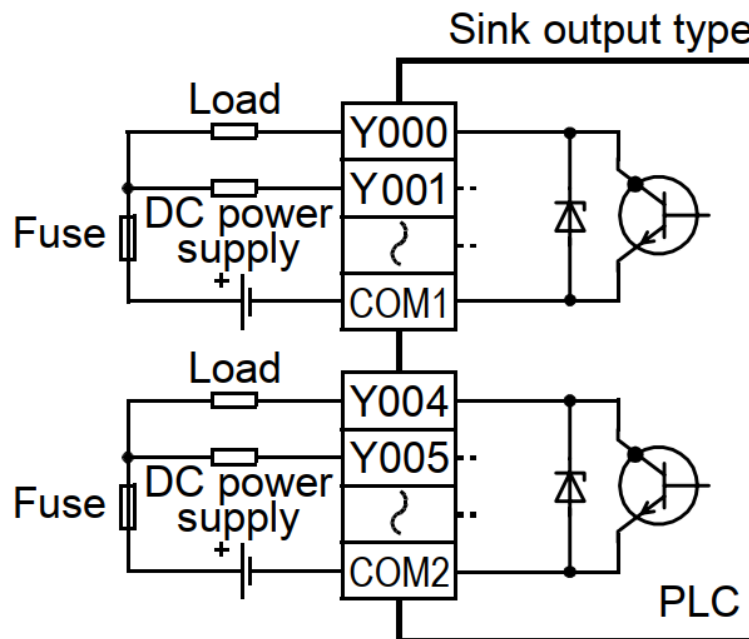
Hình: 2.1-1. Nối đầu vào kiểu SINK



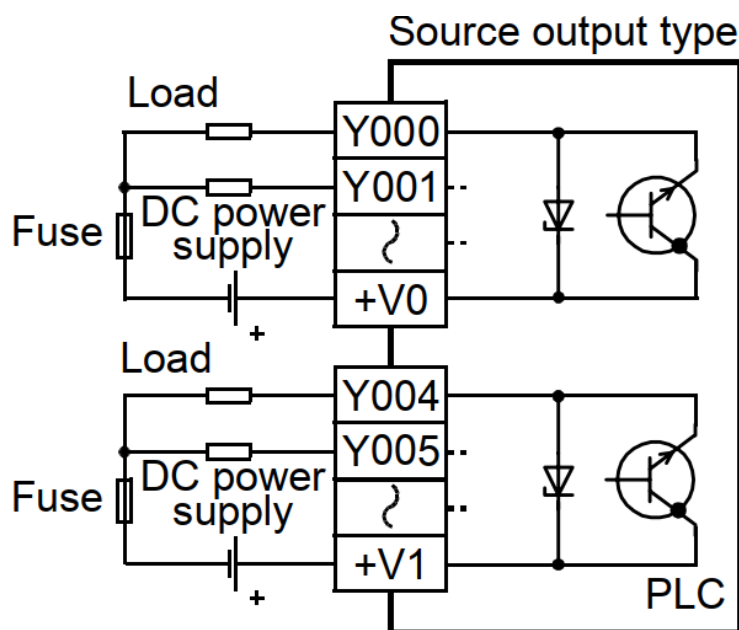
Hình 2.1-2. Nối đầu vào kiểu SOURCE



Hình 2.1-3. Đầu nối đầu ra cho PLC loại ra Rơ le



Hình 2.1-4. Nối đầu ra kiểu SINK



Hình 2.1-5. Nối đầu ra kiểu SOURCE

3. Các bước thực hiện

Bước 1: Đấu nối nguồn cung cấp:

Cấp nguồn 220VAC hoặc 24VDC theo đúng chủng loại PLC ghi trên vỏ tránh chập cháy thiết bị.

Bước 2: Đấu nối đầu vào:

- Đấu nối đầu vào nối theo kiểu SINK (sơ đồ hình....)
- Đấu nối đầu vào nối theo kiểu SOURCE (sơ đồ hình....)
- Sử dụng thiết bị đầu vào khác nhau để đấu nối: nút nhấn, cảm biến, encoder.
- Trong thực tế việc đấu nối đầu vào theo kiểu Sink hay Source là tùy thuộc vào loại thiết bị đầu vào (cảm biến, encoder...)

Bước 3: Đấu nối đầu ra:

- Loại đầu ra của PLC là Rowle, Transitor NPN hay Transito PNP là do nhà sản xuất đã quy định sẵn. Do đó cần đọc kỹ ký mã hiệu của PLC để đấu nối đầu ra cho chính xác.

- **Chú ý:** chỉ được phép cấp nguồn 24 VDC vào chân COM đối với loại PLC có đầu ra transistor.

Bước 5: Dùng đồng hồ VAO đo kiểm tra:

Do thông mạch, kiểm tra đấu nối đã đúng yêu cầu hay chưa?

Bước 6: Cấp nguồn và chạy thử:

- Cấp nguồn cho PLC
- Tác động đầu vào, quan sát các đèn tín hiệu IN
- Giáo viên nạp thử chương trình để test đầu ra, quan sát các đèn tín hiệu OUT.

4. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Đèn tín hiệu IN không sáng khi tác động đầu vào	Do đầu nối không đúng kiểu Sink/Source	Kiểm tra và đầu nối lại dây
Đèn tín hiệu OUT sáng nhưng tải đầu ra (Rơ le, đèn báo) không có điện	- Nguồn cung cấp cho tải không đúng - Đầu nối không đúng kiểu Sink/Source	- đầu dùng nguồn cho tải - Đầu nối lại dây

IV. TIÊU CHÍ ĐÁNH GIÁ

1. Đi dây và đầu nối

- Phân loại màu dây cho từng loại chức năng. Ví dụ: dây nguồn màu đỏ đen; dây tín hiệu vào ra màu vàng.

- Các mối nối đảm bảo bắt vít chắc chắn, tiếp xúc tốt.

2. Đảm bảo đúng chức năng

- Các chức năng vào ra (đèn tín hiệu IN/OUT, đóng/ngắt tải) hoạt động đúng.

V. BÁO CÁO THỰC HÀNH

1. Trình tự thực hiện đầu nối vào ra.

2. Vẽ sơ đồ đầu nối vào ra theo các phương án khác nhau.

3. Ghi lại trạng thái các đèn tín hiệu khi hoạt động.

4. Các dạng sai hỏng gặp phải trong quá trình thực hành.

VI. CÂU HỎI KIỂM TRA

1. Phân biệt các loại PLC?

2. Trình bày trình tự đầu nối phần cứng cho PLC?

3. Vẽ sơ đồ đầu nối vào ra theo các phương án khác nhau?

4. Có thể cấp nguồn 220VAC vào chân COM của PLC FX3u loại đầu ra Role hay không?

Bài thực hành số 2

HƯỚNG DẪN SỬ DỤNG PHẦN MỀM GX-WORKS 2**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

Trình bày được trình tự sử dụng phần mềm Gx-Works 2 để lập trình điều khiển cho PLC

Về kỹ năng:

- Sử dụng được phần mềm Gx-Works 2 để lập chương trình điều khiển cho PLC;
- Kết nối được phần mềm với PLC và tải chương trình cho PLC làm việc;
- Khắc phục được một số sai hỏng thường gặp trong quá trình viết chương trình và kết nối thiết bị.

Về năng lực tự chủ và trách nhiệm:

- Tỉ mỉ, cẩn thận và chuẩn chỉ khi lập trình cho PLC.
- Tự đánh giá được kết quả thực hành của bản thân và hướng dẫn được người khác sử dụng phần mềm.

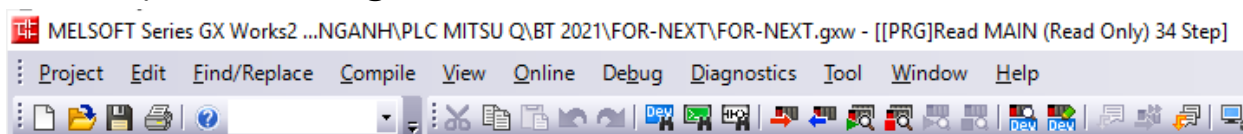
II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu phần mềm GX – Works 2**

Gx-Works 2 là phần mềm cấu hình và lập trình thế hệ kế tiếp cho điều khiển FX và Q Series. Gx-Works 2 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. Trong tài liệu này tác giả sẽ lập trình kiểu LAD.

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, cấu trúc được viết, được thử nghiệm, và chứng minh, có thể được tích hợp vào các hệ thống mới.

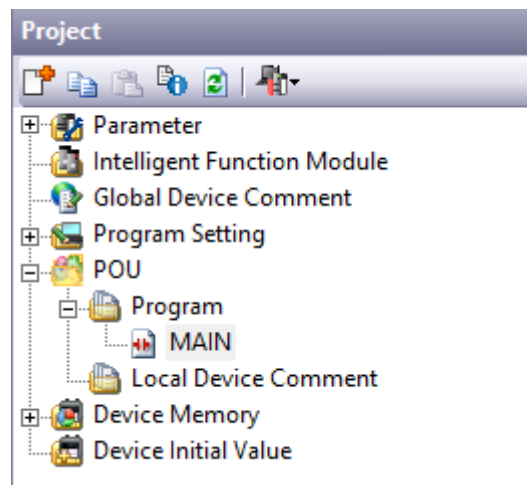
Có thể mô phỏng PLC ảo để kiểm chứng kết quả lập trình.

Gx-Works 2 bao gồm các chế độ theo dõi, giám sát, chỉnh sửa và khả năng gỡ lỗi trực tuyến cho phép kiểm soát được trạng thái hoạt động của PLC.

2. Một số chức năng cơ bản

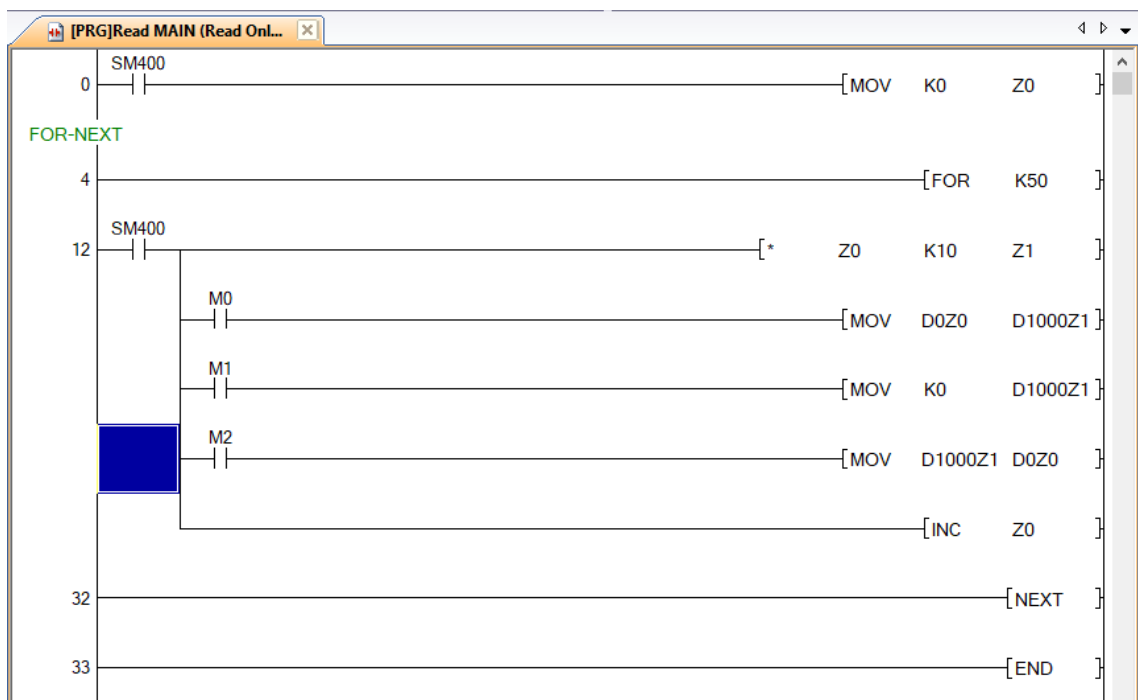
Hình 2.2-1. Thanh công cụ chính

Thanh công cụ chính bao gồm các chức năng chính như: thêm mới, mở, lưu project; các chức năng đọc, ghi chương trình; chức năng giám sát, mô phỏng online....



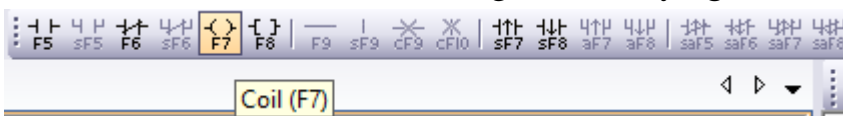
Hình 2.2-2. Cửa sổ project

Đây là cửa sổ giúp người lập trình nhanh chóng tìm được các file chức năng của dự án như: cài đặt parameter, thiết lập các module kết nối, mở cửa sổ viết lệnh, bộ nhớ RAM.



Hình 2.2-3. Cửa sổ viết câu lệnh

Đây là khu vị trí giúp người dùng viết các câu lệnh lập trình. Các dòng lệnh sẽ được đánh số từ 1 cho đến hết. Trong cửa sổ này người dùng sẽ kết hợp với sử

dụng:  để thiết lập các câu lệnh chức năng; thêm, sửa, xóa các dây nối.

Watch 1		
Device/Label	Current Value	Data Type
M0	--	Bit
D0	--	Word[Signed]
D1	--	Word[Signed]
D49	--	Word[Signed]
D1000	--	Word[Signed]
D1010	--	Word[Signed]
D1490	--	Word[Signed]
Z0	--	Word[Signed]
M1	--	Bit
M2	--	Bit

Hình 2.2-4. Cửa sổ theo dõi, thiết lập giá trị các bit và thanh ghi

III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

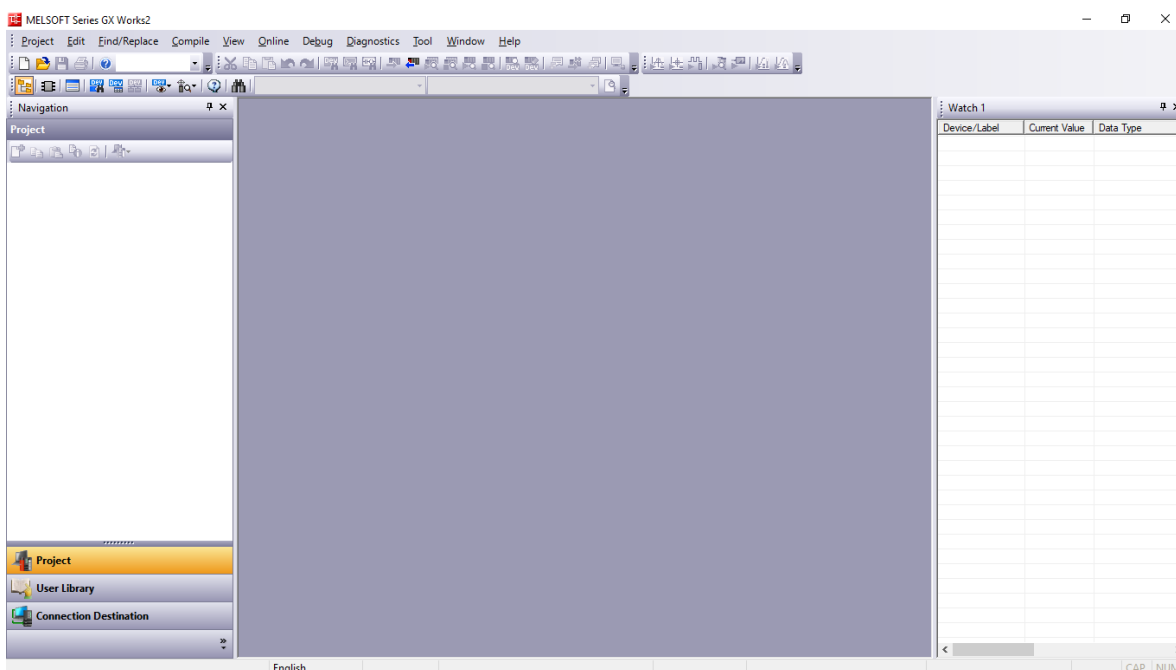
STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
2	Máy tính đã cài đặt sẵn phần mềm Gx – Works 2	1 bộ
3	Cáp kết nối máy tính với PLC	1 bộ
4	Bộ đồ nghề dụng cụ	1 bộ

2. Các bước thực hiện

Bước 1: Khởi động phần mềm Gx – Works 2:




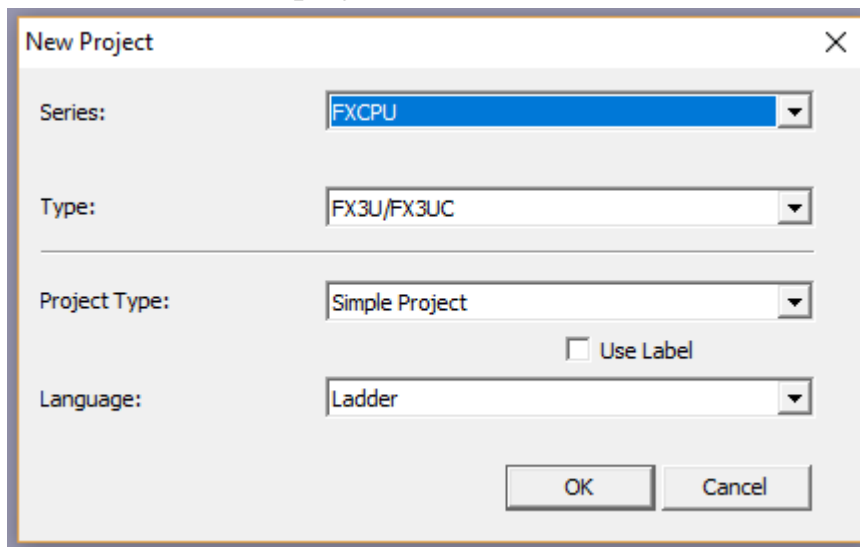
Chọn vào biểu tượng trên màn hình, giao diện phần mềm xuất hiện như hình vẽ:



Hình 2.2-5. Giao diện chính của phần mềm Gx – Works 2

Bước 2: Tạo dự án mới:

- Click chuột vào Project → New hoặc click vào biểu tượng  để tạo 1 dự án mới. Sẽ xuất hiện cửa sổ tạo project:

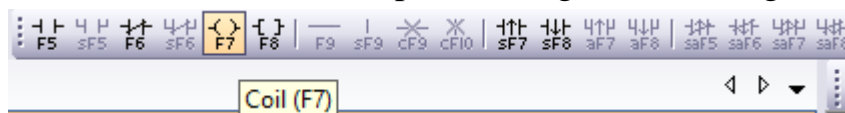


Hình 2.2-6. Cửa sổ tạo dự án mới








- Chọn các thông số Series, Type, Project Type language như trên hình. Sau đó nhấn OK. Cửa sổ viết chương trình sẽ xuất hiện như trên hình 2.2-3.




Bước 3: Viết chương trình:

Tiến hành viết câu lệnh lập trình bằng cách sử dụng khối chức năng:

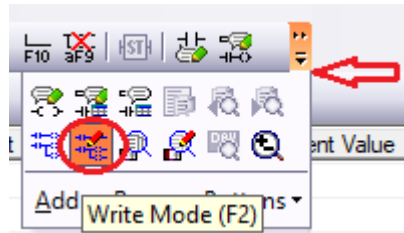


Hoặc sử dụng các phím tắt: F5, Shift + F5, F6, Shift + F6:

	Tiếp điểm thường mở (nhấn F5)
	Tiếp điểm thường mở (nhấn F6)
	Cuộn dây M _i , Y _i (Nhấn F7)
	Các lệnh chức năng (nhấn F8)
	Nối dây ngang (nhấn F9)
	Nối dây dọc (nhấn Shift + F9)
	Xóa dây ngang (Ctrl + F9)

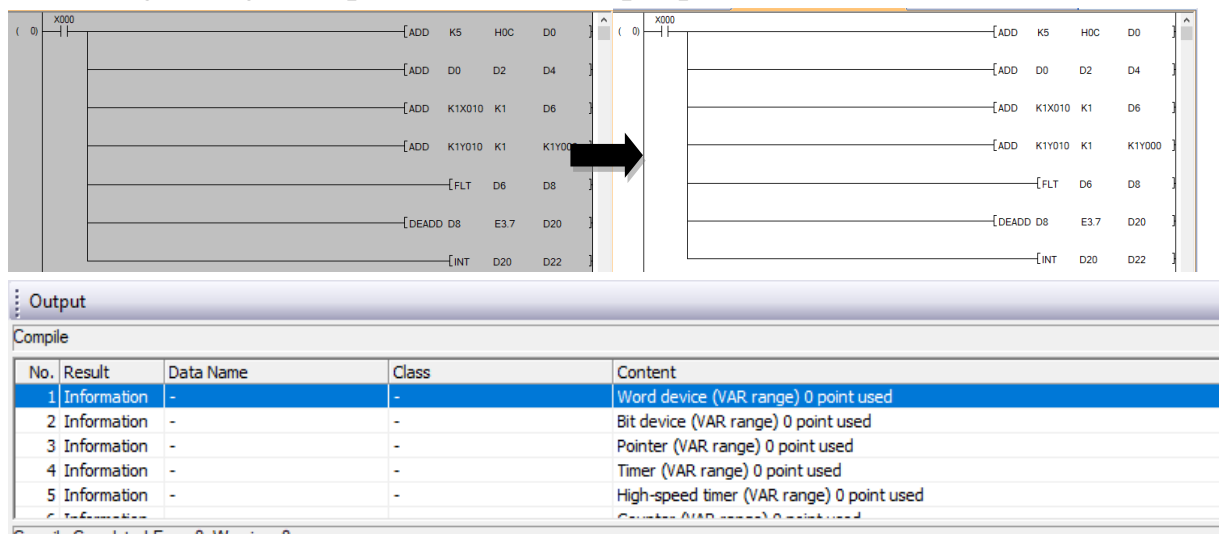
	Xóa dây dọc (Ctrl + F9)
	Xung sườn lên (Shift + F7)
	Xung sườn xuống (Shift + F8)

Khi viết chương trình cần chú ý chọn chế độ Write mode hoặc nhấn F2:



Bước 4: Biên dịch chương trình

Tại cửa sổ soạn thảo lệnh nhấn phím F4 để tiến hành biên dịch chương trình. Nếu chương trình ko có lỗi, màu nền của sổ soạn thảo sẽ chuyển từ màu xám sang màu trắng. Đồng thời quan sát cửa sổ Output phía dưới để kiểm soát lỗi:

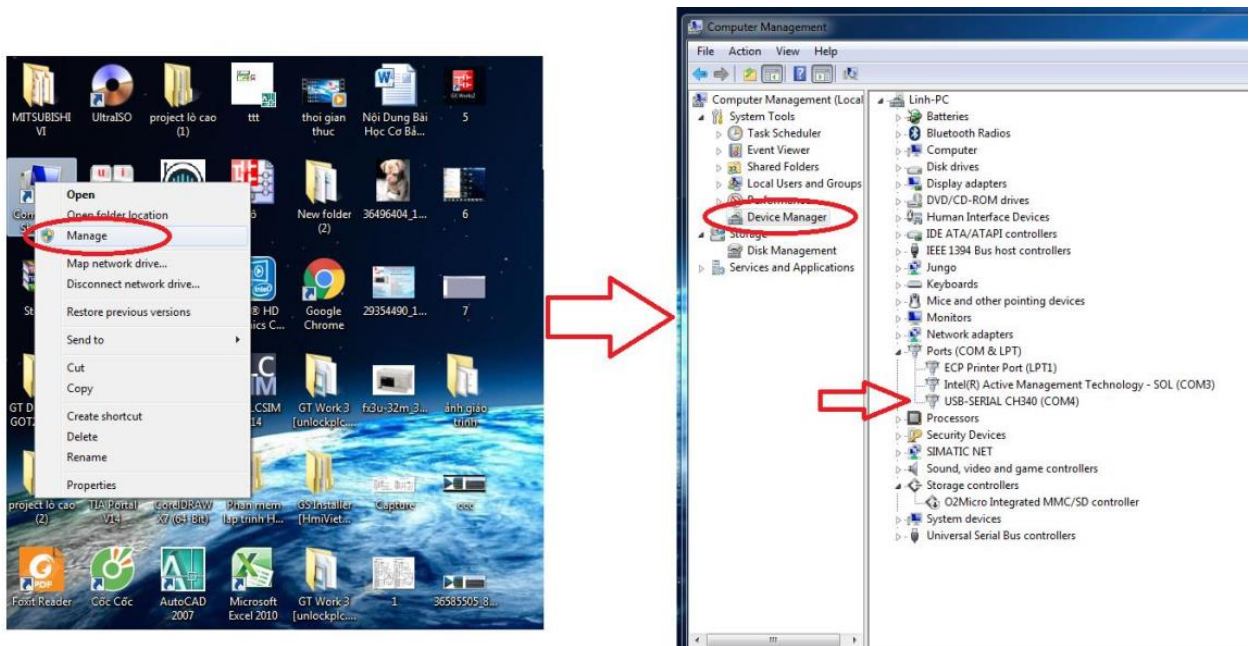


Hình 2.2-7. Biên dịch chương trình

Nếu chương trình có lỗi sẽ xuất hiện các cửa sổ thông báo hoặc hiển thị trong cửa sổ Output phía dưới.

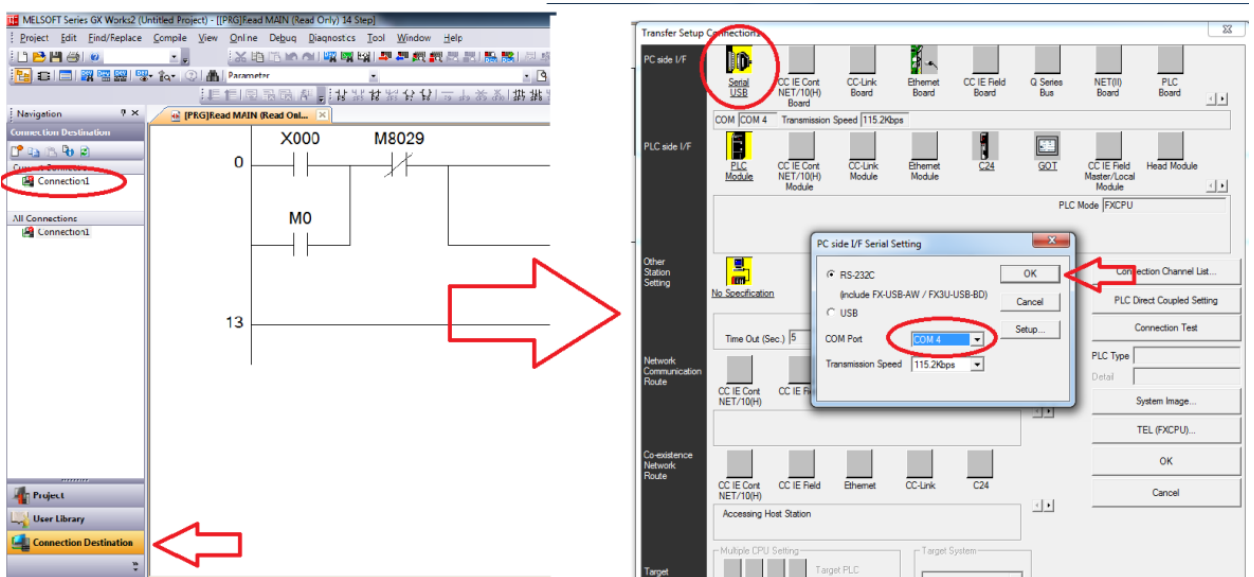
Bước 5: Tải chương trình vào PLC

- Cài đặt driver cho cáp nạp (phần mềm đi kèm theo cáp khi mua).
- Kiểm tra kết nối PLC với máy tính: Click chuột phải vào biểu tượng My Computer → Manage → kiểm tra kết nối của cáp với máy tính là COM nào? (ví dụ là COM 4)



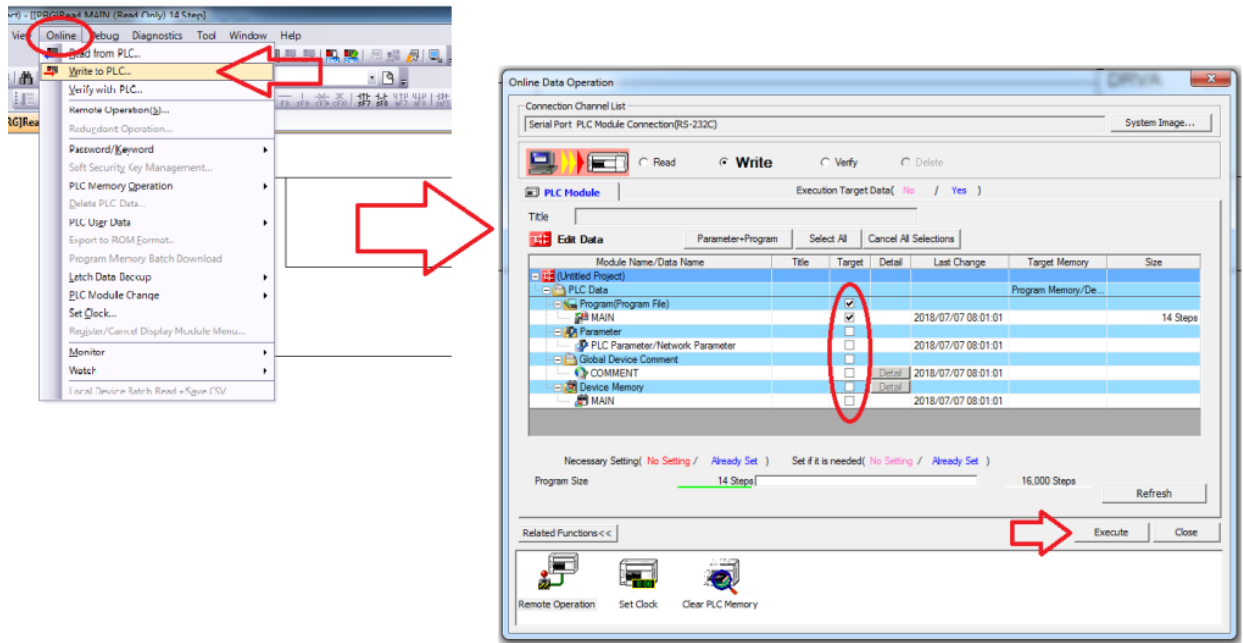
Hình 2.2-8. Kiểm tra kết nối cáp nạp

- Cài đặt cổng COM trên phần mềm Gx-Works 2: chọn Connection Destination → Connection 1 → Serial USB → chọn COM 4 → OK → OK.




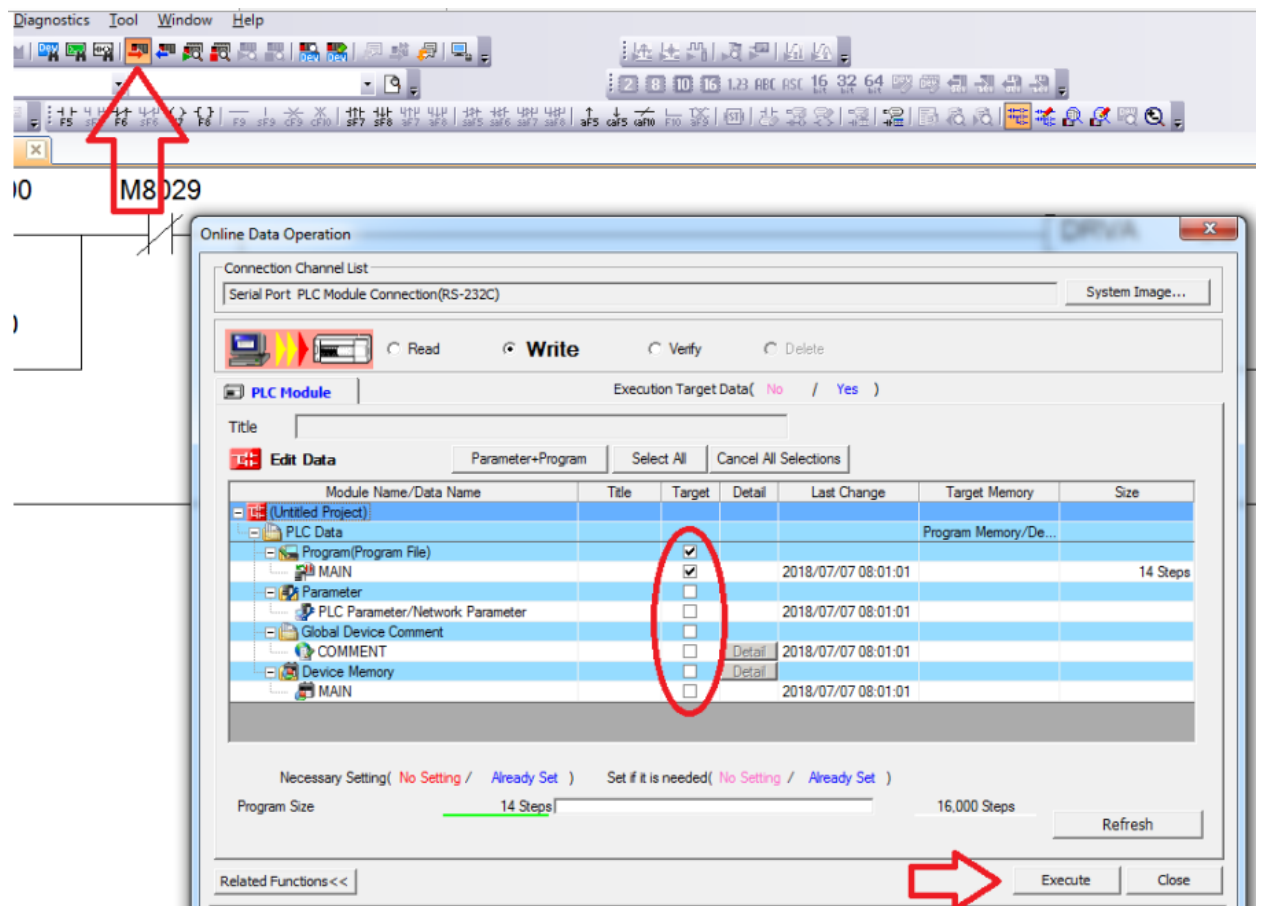
Hình 2.2-9. Chọn cổng COM kết nối

- Tiến hành tải chương trình cho PLC:
 + Cách 1: Chọn Online (trên thanh công cụ) → Write to PLC → Tick vào tagert (chọn các mục cần tải: Parameter, Main, comment) → Execute để tải chương trình xuống



Hình 2.2-10a. Tải chương trình vào PLC cách 1

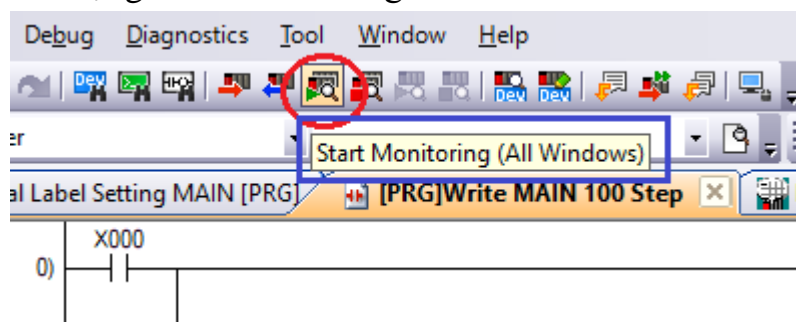
+ Cách 2: Click vào biểu tượng Write to PLC  trên thanh công cụ phía trên màn hình soạn thảo. Xuất hiện cửa sổ tải chương trình vào PLC giống như cách 1.



Hình 2.2-10a. Tải chương trình vào PLC cách 2

Bước 6: Giám sát online

- Chọn vào biểu tượng Start Monitoring như hình vẽ:



Hình 2.2-11. Giám sát hoạt động online

- Có thể quan sát trực tiếp giá trị của các bit, các thanh ghi trực tiếp trên màn hình soạn thảo hoặc nhập các bit, các thanh ghi trong cửa sổ Watch như hình 2.2-4.

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không cho soạn thảo, sửa, xóa chương trình	Chưa chọn vào chế độ soạn thảo	Chọn chế độ soạn thảo lệnh Write mode hoặc nhấn phím F2. (chi tiết theo bước 3)
Không tải được chương trình vào PLC	- Kết nối máy tính với cáp không đúng - Lỗi cáp kết nối	- Chọn lại cổng COM - Kiểm tra, thay cáp mới

IV. TIÊU CHÍ ĐÁNH GIÁ

1. Thao tác thành thạo các chức năng cơ bản trên phần mềm Gx-Works 2.
2. Soạn thảo được chương trình đơn giản dạng LAD
3. Tải được chương trình vào PLC

V. BÁO CÁO THỰC HÀNH

1. Trình bày trình tự các bước thực hành viết chương trình đơn giản cho PLC.

2. Các chú ý, các lỗi gặp phải trong quá trình thực hành

VI. CÂU HỎI KIỂM TRA

Viết chương trình điều khiển bật tắt bóng đèn theo bảng chức năng sau:

Nút nhấn	Nút nhấn	Nút nhấn
Pb1	X000	Nút nhấn ON
Pb2	X001	Nút nhấn OFF
H	Y000	Bóng đèn

Bài thực hành số 3

THỰC HÀNH VỚI CÁC LỆNH CƠ BẢN

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

Nhận biết được ký hiệu, cú pháp lệnh và chức năng của các tập lệnh cơ bản.

Về kỹ năng:


- Sử dụng được thành thạo các lệnh cơ bản đúng chức năng;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình sử dụng các câu lệnh lập trình.

Về năng lực tự chủ và trách nhiệm:

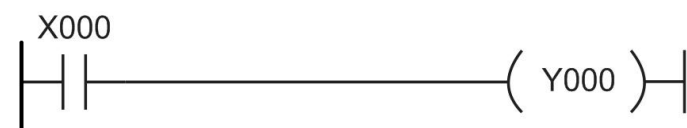
- Tự đánh giá được kết quả thực hành của bản thân.
- Hướng dẫn được người khác sử dụng đúng các câu lệnh lập trình cơ bản.

II. TÓM TẮT LÝ THUYẾT

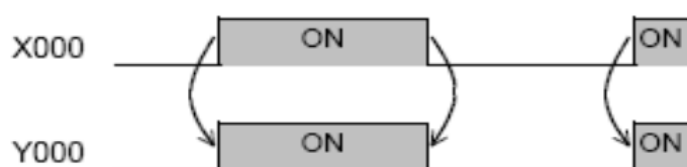
1. Tiếp điểm thường mở

Ký hiệu	Chức Năng	Thiết Bị
	Có nhiệm vụ khởi tạo trạng thái logic, loại tiếp điểm thường mở NO, Được nối trực tiếp với đường Bus bên trái của mạch	X, Y, M, S, T, C và Di.j (bit j của thanh ghi Di)

- Ví dụ:



Biểu đồ thời gian



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

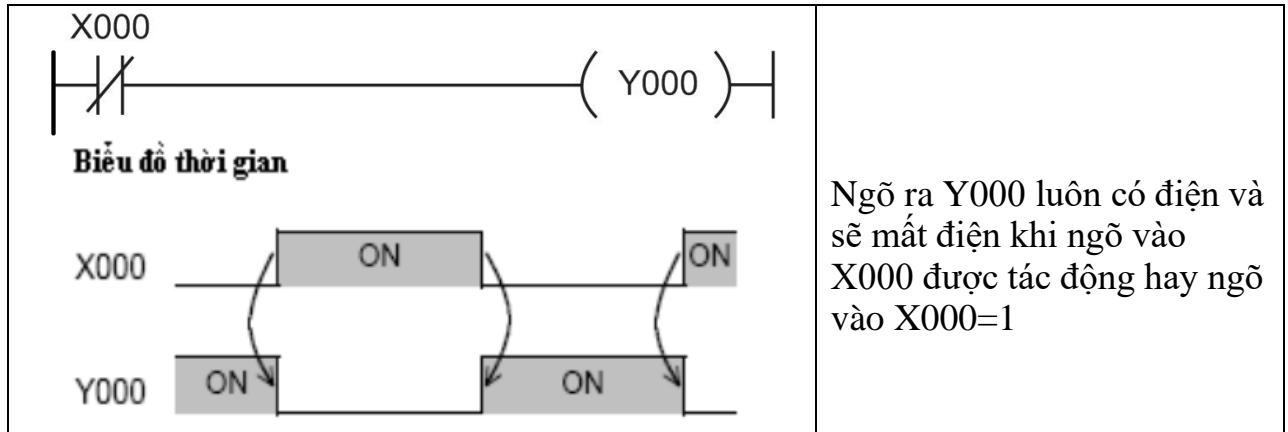
Hình 2.3-1. Minh họa sử dụng tiếp điểm NO

- Cách gọi lệnh: nhấn phím F5 hoặc chọn biểu tượng tiếp điểm NO trên

thanh công cụ: 

2. Tiếp điểm thường đóng

Ký hiệu	Chức Năng	Thiết Bị
	Có nhiệm vụ khởi tạo trạng thái logic, là loại tiếp điểm thường đóng NC, Được nối trực tiếp với đường Bus bên trái của mạch	X, Y, M, S, T, C và Di.j (bit j của thanh ghi Di)



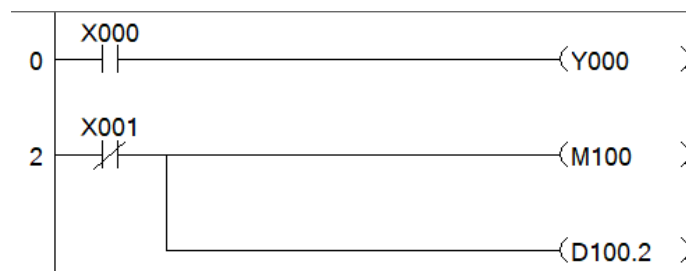
Hình 2.3-2. Minh họa sử dụng tiếp điểm NC

- Cách gọi lệnh: nhấn phím F6 hoặc chọn biểu tượng tiếp điểm NO trên thanh công cụ:

3. Lệnh out (xuất)

Ký hiệu	Chức Năng	Thiết Bị
	- Điều khiển cuộn dây - Có thể nối song song nhiều ngõ ra - Không dùng để điều khiển ngõ vào	Y, M, S, T, C và Di.j (bit j của thanh ghi Di)

Ví dụ:



Hình 2.3-3. Minh họa sử dụng tiếp điểm đầu ra

- Cách gọi lệnh: nhấn phím F7 hoặc chọn biểu tượng tiếp điểm NO trên thanh công cụ:

4. Tiếp điểm sườn lên (PLS)

Ký hiệu	Chức Năng	Thiết Bị
	- Phát hiện sườn lên, - Khi đầu vào chuyển trạng thái từ 0 lên 1 thì tiếp điểm này sẽ phát 1 xung kim tới đầu ra	Y, M, S, T, C và Di.j (bit j của thanh ghi Di)



Hình 2.3-4. Minh họa lệnh PLS

Khi X000 chuyển trạng thái từ 0 lên 1 (ví dụ khi nhấn nút nhấn) thì Lệnh sườn lên sẽ bắt cạnh xung sườn lên và đưa ra ngõ ra 1 xung kim duy nhất để bật ngõ ra Y000 =1, sau đó tiếp điểm thường hở Y000 sẽ =1 và tự ra duy trì.

- Cách lấy lệnh: Nhấn Shift + F7 hoặc chọn vào biểu tượng trên thanh công cụ:



5. Lệnh sườn xuống (PLF)

Ký hiệu	Chức Năng	Thiết Bị
	- Phát hiện sườn xuống - Khi đầu vào chuyển trạng thái từ 1 về 0 thì tiếp điểm này sẽ phát 1 xung kim tới đầu ra	Y, M, S, T, C và Di.j (bit j của thanh ghi Di)

- Ví dụ:



Hình 2.3-5. Minh họa lệnh PLF

Khi X000 chuyển trạng thái từ 1 về 0 thì Lệnh sườn xuống sẽ bắt cạnh xung sườn xuống và đưa ra ngõ ra 1 xung kim duy nhất để bật ngõ ra Y000 =1, sau đó tiếp điểm thường hở Y000 sẽ =1 và tự ra duy trì.

- Cách lấy lệnh: Nhấn Shift + F8 hoặc chọn vào biểu tượng trên thanh công cụ:



6. Lệnh SET và RESET

Cú pháp lệnh	Chức Năng	Thiết Bị
SET -[SET Y000]	Gán giá trị tại địa chỉ đầu ra bằng 1	Y, M, S, T, C và Di.j (bit j của thanh ghi Di)
RST -[RST Y000]	Gán giá trị tại địa chỉ đầu ra bằng 0	

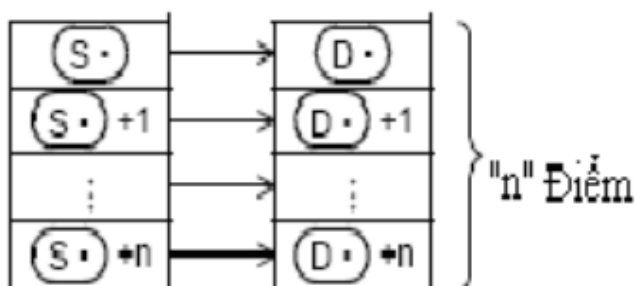
Cách lấy lệnh: Gõ SET hoặc RST → dấu cách tên địa chỉ đầu ra → Enter

7. Lệnh di chuyển dữ liệu MOV

Cú pháp lệnh	Chức Năng	Thiết Bị
MOV -[MOV S D]-	Di chuyển dữ liệu dạng số nguyên 16 bit từ nguồn (S) đến đích (D)	- S: Hằng số (K, H), D, T, C, R, V, Z, KnX, KnY, KnM - D: D, T, C, R, V, Z, KnY, KnM
DMOV -[DMOV S D]-	Di chuyển dữ liệu dạng số nguyên 32 bit từ nguồn (S) đến đích (D)	- S: Hằng số (K, H), D, T, C, R, V, Z, KnX, KnY, KnM - D: D, T, C, R, V, Z, KnY, KnM
DEMOV -[DEMOV S D]-	Di chuyển dữ liệu dạng số thực 32 bit từ nguồn (S) đến đích (D)	- S: Hằng số (E), D, T, C, R, V, Z, KnX, KnY, KnM - D: D, T, C, R, V, Z, KnY, KnM
MOVP DMOVP DEMOVP	Thực hiện lệnh tại thời điểm sườn lên của tiếp điểm đầu vào	

8. Lệnh di chuyển khối dữ liệu BMOV

Cú pháp lệnh	Chức Năng	Thiết Bị
BMOV -[BMOV S D n]-	Di chuyển vùng dữ liệu bắt đầu từ S, S+1, ... vào D, D+1...	- S: D, T, C, R, V, Z, KnX, KnY, KnM - D: D, T, C, R, V, Z, KnY, KnM - n: Hằng số (K, H), D, R
BMOV P -[BMOV P S D n]-	Thực hiện lệnh tại thời điểm sườn lên của tiếp điểm đầu vào	



9. Lệnh di chuyển dữ liệu với con trỏ V, Z

III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
2	Giắc cắm	
3	Bộ đồ nghề dụng cụ	1 bộ
4	Máy tính cài đặt sẵn phần mềm	1 bộ

2. Bảng tín hiệu vào ra

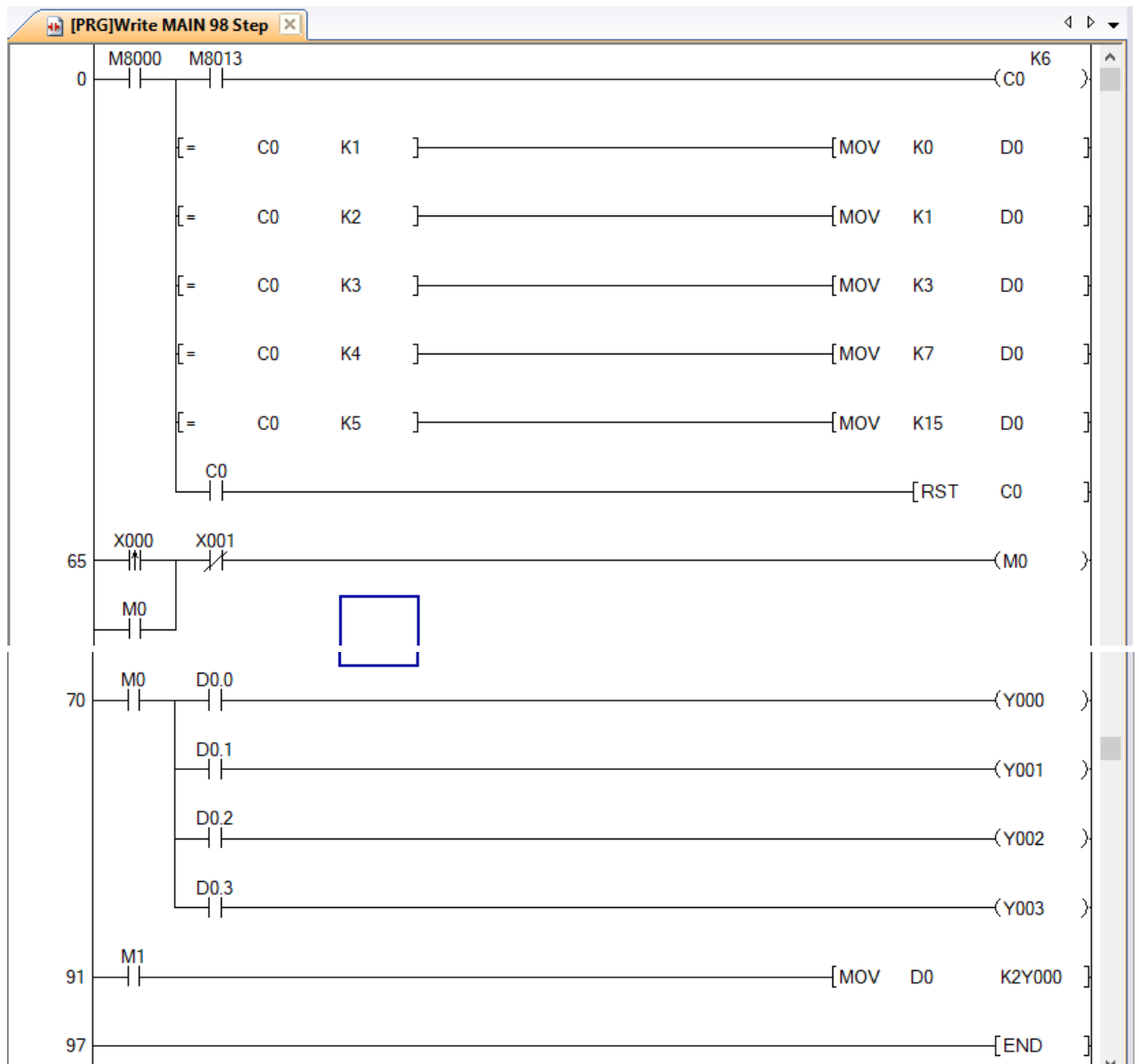
INPUT		OUTPUT	
PB1	X000	LED 1	Y000
Pb2	X001	LED 1	Y001
		LED 1	Y002
		LED 1	Y003

3. Các bước thực hiện

- **Bước 1:** Tạo project

Tạo 1 dự án mới để thực hành các câu lệnh cơ bản

- **Bước 2:** Viết chương trình sử dụng các lệnh cơ bản



Hình 2.3-6. Chương trình mẫu sử dụng các lệnh cơ bản

- Bước 3: Biên dịch và tải chương trình vào PLC

- + Nhấn F4 để biên dịch chương trình sau khi lập trình
- + Kiểm tra kết nối và tải chương trình vào PLC

- Bước 4: Chạy online và kiểm tra kết quả của các lệnh

- + Chọn biểu tượng Start monitoring:

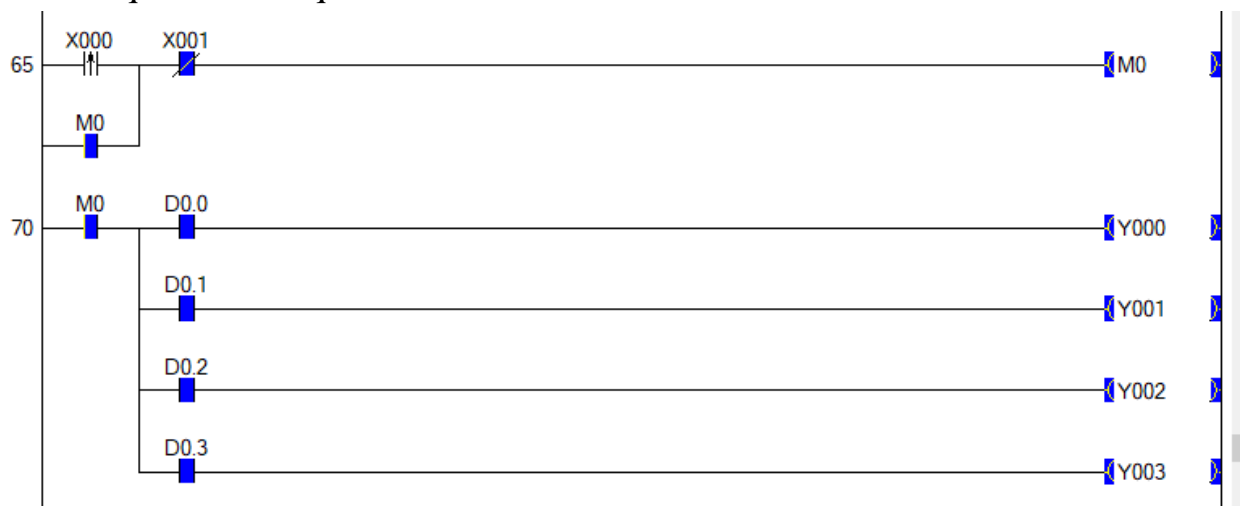


- + Chọn Online trên thanh công cụ → Watch → Register to watch:

Watch 1(Monitor Executing)		
Device/Label	Current Value	Data Type
X0	0	Bit
X1	0	Bit
M1	0	Bit
D0	7	Word[Signed]
Y0	1	Bit
Y1	1	Bit
Y2	1	Bit
Y3	0	Bit

Hình 2.3-7. Cửa sổ giám sát kết quả

- + Tại cửa sổ Watch 1: nhập các thanh ghi, đầu vào ở ô Devive/Label → chọn kiểu dữ liệu phù hợp ở ô Data type
- + ON/OFF các bit X000, X001 bằng cách nhập giá trị 1/0 vào ô Current Value (hoặc nhấn nút nhấn) để PLC thực hiện các lệnh.
- + quan sát kết quả:



Hình 2.3-8. Trạng thái dữ liệu khi chạy chương trình

4. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không hiển thị giá trị trên cửa sổ watch	- Chưa bật chế độ xem - Chưa chọn kiểu dữ liệu	- Chuột phải vào cửa sổ watch 1 → Start watch - Chọn đúng kiểu dữ liệu ở ô Data type.

IV. TIÊU CHÍ ĐÁNH GIÁ

- Phân biệt được chức năng các lệnh cơ bản.
- Viết được chương trình lập trình sử dụng các lệnh cơ bản.
- Kiểm tra được kết quả thực hiện lệnh.

V. BÁO CÁO THỰC HÀNH

- Liệt kê và trình bày ký hiệu, chức năng các tập lệnh cơ bản .

- Viết chương trình minh họa các tập lệnh.

VI. CÂU HỎI KIỂM TRA

Nhập dãy 10 số nguyên: 1, 2, 3, ..., 9, 10 vào các thanh ghi bắt đầu từ D10?

Sao chép dãy số này vào các thanh ghi bắt đầu từ D100.

Bài thực hành số 4

THỰC HÀNH VỚI CÁC LỆNH TOÁN HỌC**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

Nhận biết được ký hiệu, cú pháp lệnh và chức năng của các tập lệnh toán học;

Về kỹ năng:

- Sử dụng được thành thạo các lệnh toán học đúng chức năng;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình sử dụng các câu lệnh lập trình.

Về năng lực tự chủ và trách nhiệm:

- Tự đánh giá được kết quả thực hành của bản thân và người khác sử dụng các câu lệnh toán học.

II. TÓM TẮT LÝ THUYẾT**1. Các lệnh toán học***a. Phép cộng (ADD)*

Ký hiệu	Chức Năng	Thiết Bị
ADD -[ADD S1 S2 D]-	Cộng giá trị dạng số nguyên 16 bit S1 với S2, lưu kết quả vào D	- S1, S2: Hằng số (K, H), D, T, C, R, V, Z - D: D, T, C, R, V, Z
DADD -[DADD S1 S2 D]-	Cộng giá trị dạng số nguyên 32 bit S1 với S2, lưu kết quả vào D	- S1, S2: Hằng số (K, H), D, T, C, R, V, Z - D: D, T, C, R, V, Z
DEADD -[DEADD S1 S2 D]-	Cộng giá trị dạng số thực 32 bit S1 với S2, lưu kết quả vào D	- S1, S2: Hằng số (E), D, T, C, R, V, Z - D: D, T, C, R, V, Z
ADDP DADDP DEADDP	Thực hiện lệnh tại thời điểm sườn lên của tiếp điểm đầu vào	

Chú ý: Khi kết quả vượt quá khả năng lưu trữ của D (16 bit hoặc 32 bit) thì giá trị lưu trong D sẽ không đúng (bị tràn dữ liệu).

b. Phép trừ (SUB)

Ký hiệu	Chức Năng	Thiết Bị
SUB -[SUB S1 S2 D]-	- Lấy S1 trừ S2, lưu kết quả vào D - (S1, S2, D là dạng số nguyên 16 bit)	- S1, S2: Hằng số (K, H), D, T, C, R, V, Z - D: D, T, C, R, V, Z
DSUB -[DSUB S1 S2 D]-	- Lấy S1 trừ S2, lưu kết quả vào D - (S1, S2, D là dạng số nguyên 32 bit)	- S1, S2: Hằng số (K, H), D, T, C, R, V, Z - D: D, T, C, R, V, Z
DESUB -[DESUB S1 S2 D]-	- Lấy S1 trừ S2, lưu kết quả vào D - (S1, S2, D là dạng số thực 32 bit)	- S1, S2: Hằng số (E), D, T, C, R, V, Z - D: D, T, C, R, V, Z
SUBP DSUBP DESUBP	Thực hiện lệnh tại thời điểm sườn lên của tiếp điểm đầu vào	

Chú ý: Khi kết quả vượt quá khả năng lưu trữ của D (16 bit hoặc 32 bit) thì giá trị lưu trong D sẽ không đúng (bị tràn dữ liệu).

c. Phép nhân (MUL)

Ký hiệu	Chức Năng	Thiết Bị
MUL -[MUL S1 S2 D]-	- Lấy S1 nhân S2, lưu kết quả vào D - S1, S2 là dạng số nguyên 16 bit - D là dạng số nguyên 32 bit $\begin{matrix} \text{BIN} & \text{BIN} & \text{BIN} \\ ((S1)) \times ((S2)) \longrightarrow ((D)) +1, ((D)) \\ 16\text{bits} & 16\text{bits} & 32\text{bits} \end{matrix}$	- S1, S2: Hằng số (K, H), D, T, C, R, V, Z - D: D, T, C, R, V, Z
DMUL -[DMUL S1 S2 D]-	- Lấy S1 nhân S2, lưu kết quả vào D - S1, S2 là dạng số nguyên 32 bit - D là dạng số nguyên 64 bit $\begin{matrix} \text{BIN} & \text{BIN} & \text{BIN} \\ ((S1)) \times ((S2)) \longrightarrow ((D)) +1, ((D)) \\ 32\text{bits} & 32\text{bits} & 64\text{bits} \end{matrix}$	- S1, S2: Hằng số (K, H), D, T, C, R, V, Z - D: D, T, C, R, V, Z
DEMUL -[DEMUL S1 S2 D]-	- Lấy S1 nhân S2, lưu kết quả vào D - S1, S2 là dạng số thực 32 bit - D là dạng số thực 32 bit $[(S1) +1, (S1)] \times [(S2) +1, (S2)] \longrightarrow [(D) +1, (D)]$ Dấu chấm động nhị phân Dấu chấm động nhị phân Dấu chấm động nhị phân	- S1, S2: Hằng số (E), D, R - D: D, R

MULP DMULP DEMULP	Thực hiện lệnh tại thời điểm sườn lên của tiếp điểm đầu vào	
----------------------------------	---	--

d. Phép chia

Ký hiệu	Chức Năng	Thiết Bị
DIV -[DIV S1 S2 D]-	- S1 (16bit) là số bị chia - S2 (16bit) là số chia - D (16bit) lưu thương số - D+1 (16bit) lưu số dư - S1, S2, D là các số nguyên $\begin{matrix} \text{Số bị chia} & \text{Số chia} & \text{Thương} & \text{Số dư} \\ \text{BIN} & \text{BIN} & \text{BIN} & \text{BIN} \\ ((S1) \div (S2)) \rightarrow ((D) \dots (D+1)) \\ 16\text{bits} & 16\text{bits} & 16\text{bits} & 16\text{bits} \end{matrix}$	- S1, S2: Hàng số (K, H), D, T, C, R, V, Z - D: D, R
DDIV -[DDIV S1 S2 D]-	- [S1+1,S1] (32bit) là số bị chia - [S2+1,S2] (32bit) là số chia - [D+1,D] (32bit) lưu thương số - [D+3,D+2] (32bit) lưu số dư - S1, S2, D là các số nguyên $\begin{matrix} \text{Số bị chia} & \text{Số chia} & \text{Thương số} & \text{Số dư} \\ \text{BIN} & \text{BIN} & \text{BIN} & \text{BIN} \\ [(S1)+1, (S1)] \div [(S2)+1, (S2)] \rightarrow [(D)+1, (D)] \dots [(D)+3, (D)+2] \\ 32\text{ bits} & 32\text{ bits} & 32\text{ bits} & 32\text{ bits} \end{matrix}$	- S1, S2: Hàng số (K, H), D, T, C, R, V, Z - D: D, R
DEDIV -[DEDIV S1 S2 D]-	- [S1+1,S1] (32bit) là số bị chia - [S2+1,S2] (32bit) là số chia - [D+1,D] (32bit) lưu thương số $\begin{matrix} \text{Số bị chia} & \text{Số chia} & \\ [(S1)+1, (S1)] \div [(S2)+1, (S2)] \rightarrow [(D)+1, (D)] \\ \text{Dấu chấm động} & \text{Dấu chấm động} & \text{Dấu chấm động} \\ \text{nhị phân} & \text{nhị phân} & \text{nhị phân} \end{matrix}$	- S1, S2: Hàng số (K, H, E), D, R - D: D, R
DIVP DDIVP DEDIVP	Thực hiện lệnh tại thời điểm sườn lên của tiếp điểm đầu vào	

2. Lệnh chuyển đổi kiểu dữ liệu

Ký hiệu	Chức Năng	Thiết Bị
FLT -[FLT S D]-	Chuyển đổi kiểu dữ liệu của S từ số nguyên sang số thực và lưu vào D $(S) \rightarrow ((D)+1, (D))$ Số nguyên nhị phân → Dấu chấm động nhị phân (số thực)	S: D, R (16 bit) D: D, R (32 bit)

<p>DFLT -[DFLT S D]-</p>	$(\overline{S}) + 1, (\overline{S}) \rightarrow (\overline{D}) + 1, (\overline{D})$ <p>Số nguyên nhị phân Dấu chấm động nhị phân (số thực)</p>	<p>S: D, R (32 bit) D: D, R (32 bit)</p>
<p>INT -[INT S D]-</p>	<p>Chuyển đổi kiểu dữ liệu của S từ số thực sang số nguyên và lưu kết quả vào D</p> $\overline{S} + 1, \overline{S} \rightarrow \overline{D}$ <p>Dấu chấm động nhị phân Số nguyên nhị phân 16 bit Phân thập phân bỏ</p>	<p>S: D, R (32 bit) D: D, R (16 bit)</p>
<p>DINT -[DINT S D]-</p>	<p>Chuyển đổi kiểu dữ liệu của S từ số thực sang số nguyên và lưu kết quả vào D</p> $(\overline{S}) + 1, (\overline{S}) \rightarrow (\overline{D}) + 1, (\overline{D})$ <p>Dấu chấm động nhị phân Số nguyên nhị phân 32 bit Phân thập phân bỏ</p>	<p>S: D, R (32 bit) D: D, R (32 bit)</p>

III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

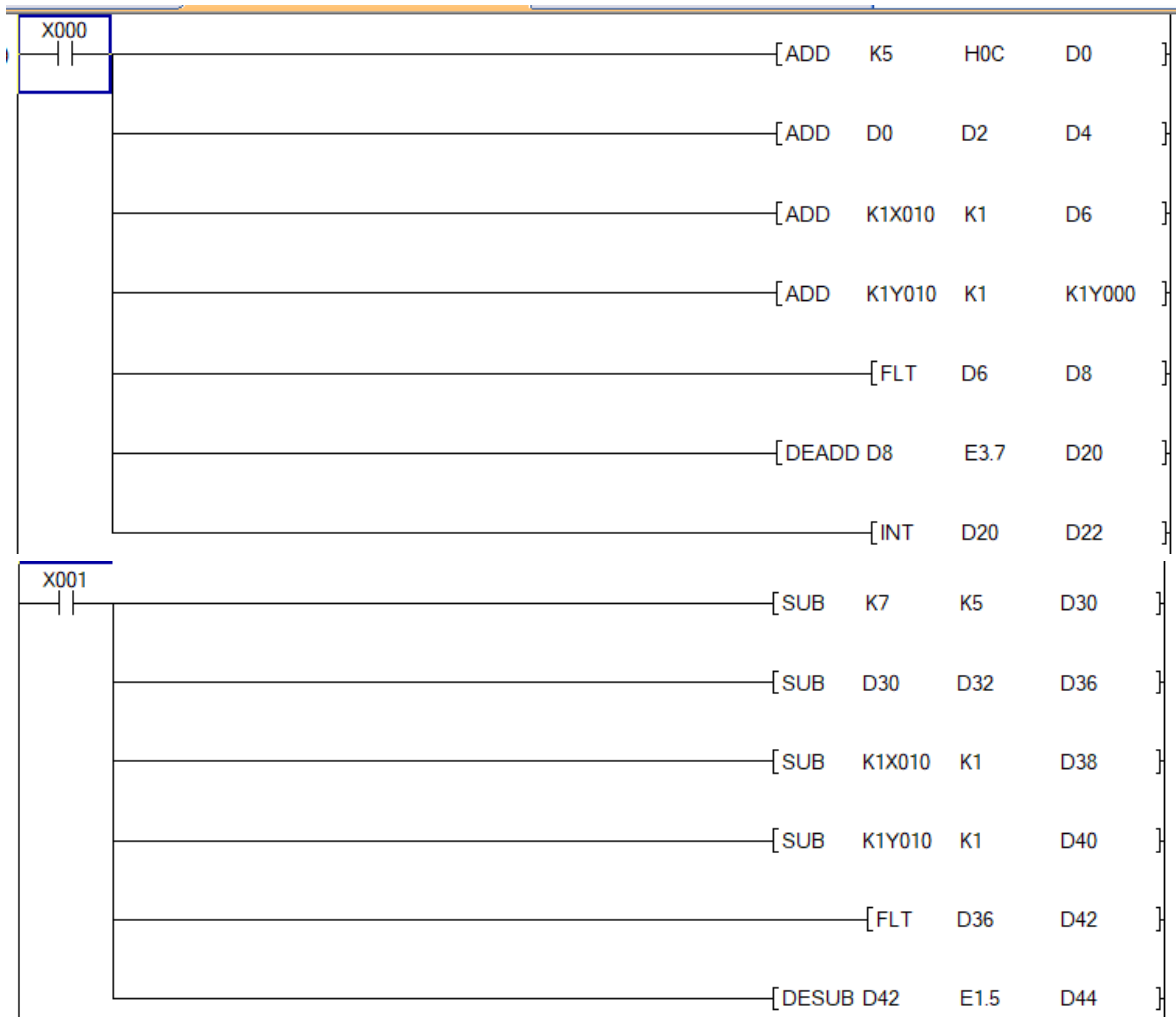
STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
2	Giắc cắm	
3	Bộ đồ nghề dụng cụ	1 bộ
4	Máy tính cài đặt sẵn phần mềm	1 bộ

2. Các bước thực hiện

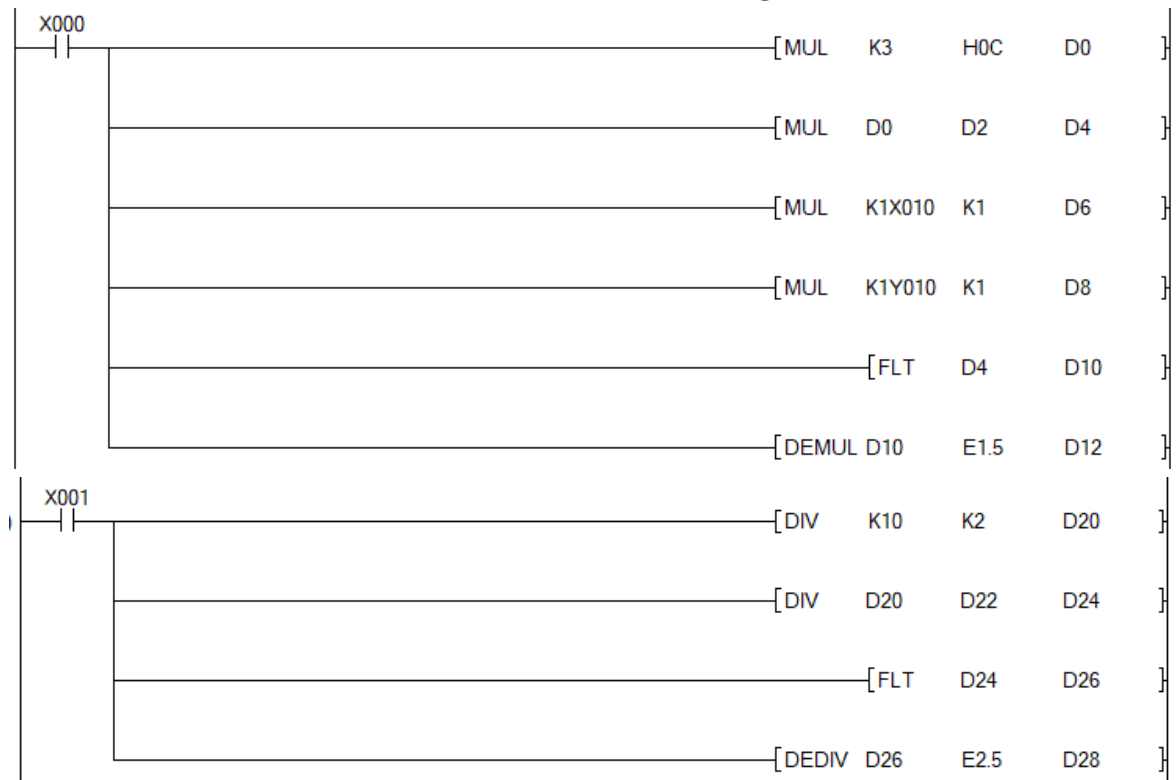
- Bước 1: Tạo project

Tạo 1 dự án mới để thực hành các câu lệnh cơ bản

- Bước 2: Viết chương trình sử dụng các lệnh cơ bản và toán học



Hình 2.4-1a. Các lệnh cộng trừ



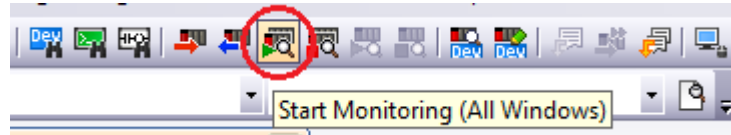
Hình 2.4-1b. Các lệnh nhân chia

- Bước 3: Biên dịch và tải chương trình vào PLC

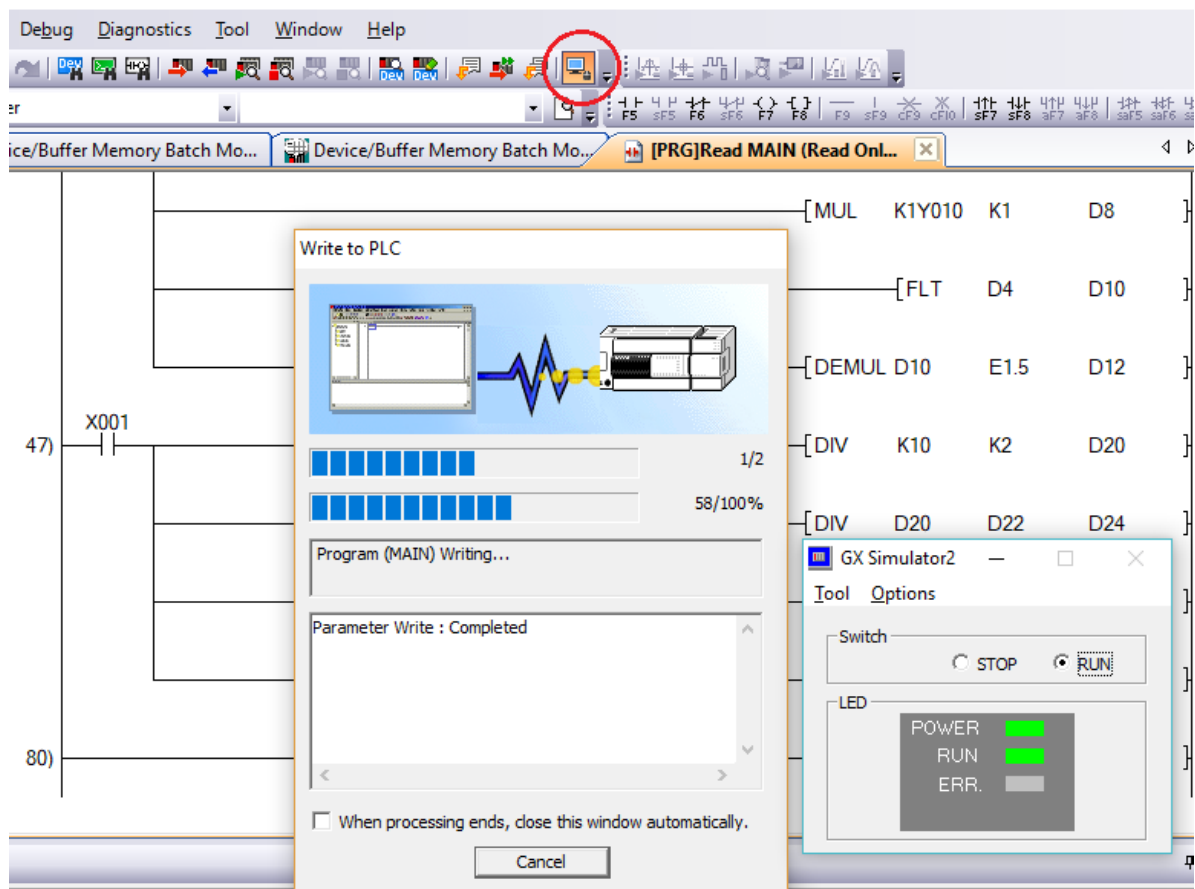
- + Nhấn F4 để biên dịch chương trình sau khi lập trình
- + Kiểm tra kết nối và tải chương trình vào PLC

- Bước 4: Chạy online và kiểm tra kết quả của các lệnh

- + Chọn biểu tượng Start monitoring:

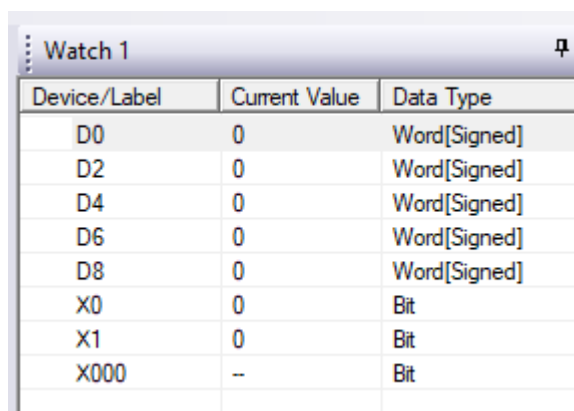


+ Hoặc chạy mô phỏng trên phần mềm: chọn biểu tượng Start/Stop Simulation



Hình 2.4-2. Chạy mô phỏng

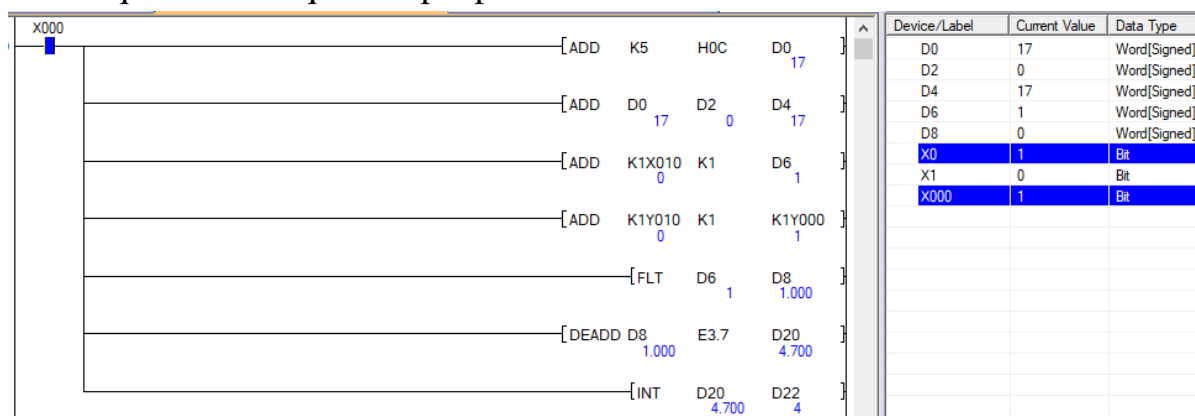
- + Hoặc cảm giác trên mô hình để kết nối nút nhấn với X0, X1
- + Chọn Online trên thanh công cụ → Watch → Register to watch:



Device/Label	Current Value	Data Type
D0	0	Word[Signed]
D2	0	Word[Signed]
D4	0	Word[Signed]
D6	0	Word[Signed]
D8	0	Word[Signed]
X0	0	Bit
X1	0	Bit
X000	--	Bit

Hình 2.4-3. Cửa sổ giám sát kết quả

- + Tại cửa sổ Watch 1: nhập các thanh ghi, đầu và ra ở ô Devive/Label → chọn kiểu dữ liệu phù hợp ở ô Data type
- + ON/OFF các bit X000, X001 bằng cách nhập giá trị 1/0 vào ô Current Value (hoặc nhấn nút nhấn) để PLC thực hiện các lệnh.
- + quan sát kết quả các phép tính:



Device/Label	Current Value	Data Type
D0	17	Word[Signed]
D2	0	Word[Signed]
D4	17	Word[Signed]
D6	1	Word[Signed]
D8	0	Word[Signed]
X0	1	Bit
X1	0	Bit
X000	1	Bit

Hình 2.4-4. Thực hiện các phép tính

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Kết quả tính toán hiển thị không đúng	Các toán tử không cùng kiểu dữ liệu (số nguyên hoặc số thực)	Dùng lệnh FLT, INT để chuyển đổi các toán hạng về cùng kiểu
Không hiển thị giá trị trên cửa sổ watch	- Chưa bật chế độ xem - Chưa chọn kiểu dữ liệu	- Chuột phải vào cửa sổ watch 1 → Start watch - Chọn đúng kiểu dữ liệu ở ô Data type.

IV. TIÊU CHÍ ĐÁNH GIÁ

- Phân biệt được chức năng các lệnh cơ bản và lệnh toán học.
- Viết được chương trình lập trình sử dụng các lệnh cơ bản và lệnh toán học.
- Kiểm tra được kết quả tính toán.

V. BÁO CÁO THỰC HÀNH

- Liệt kê và trình bày ký hiệu, chức năng các tập lệnh cơ bản và lệnh toán học.
- Viết chương trình minh họa các tập lệnh

VI. CÂU HỎI KIỂM TRA

Cho dãy 5 số: 5, 10.5, 15.5, 20.1, 25

- Lưu 5 số này vào các thanh ghi bắt đầu từ D10?
- Tính tổng, tích của 5 số?

Bài thực hành số 5

LẬP TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ XOAY CHIỀU 3 PHA**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Liệt kê được các trang thiết bị điện sử dụng trong mạch điều khiển động cơ xoay chiều 3 pha;

- Trình bày được trình tự lập trình để điều khiển động cơ xoay chiều 3 pha;

Về kỹ năng:

- Kết nối được các thiết bị với đầu vào - ra PLC;

- Lập trình cho PLC để chạy, dừng động cơ;

- Vận hành được hệ thống đảm bảo an toàn cho người và máy;

- Phát hiện và khắc phục được một số lỗi thường gặp trong quá trình thực hành.

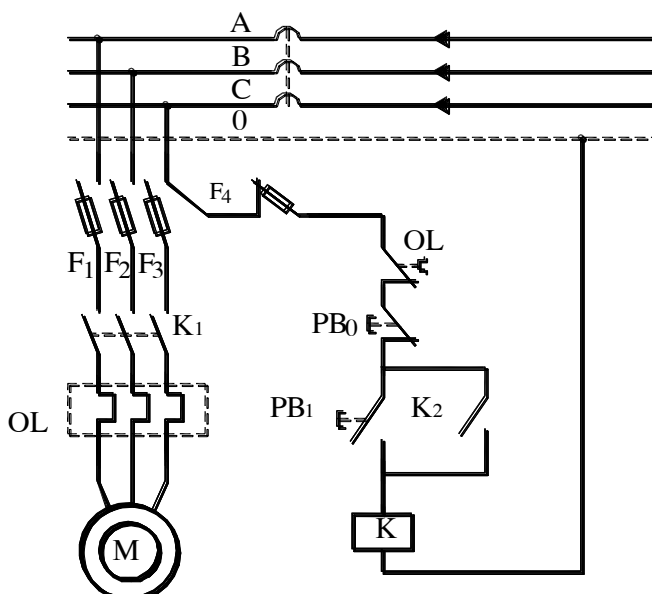
Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình thực hành;

- Có khả năng tự đấu nối, lập trình hoặc làm việc nhóm để điều khiển động cơ xoay chiều 3 pha;

- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

- Hướng dẫn được các thành viên khác trong nhóm thực hành.

II. TÓM TẮT LÝ THUYẾT**1. Sơ đồ nguyên lý mạch điện điều khiển động cơ xoay chiều 3 pha**

Hình 2.5-1. Sơ đồ nguyên lý mạch điện điều khiển động cơ xoay chiều 3 pha

2. nguyên lý hoạt động.

▪ Mở máy:

- Đóng áp tô mát nguồn.
- Ấn nút Pb1, cuộn hút công tắc tơ K có điện sẽ đóng điện cho động cơ hoạt động qua các tiếp điểm động lực K1 và duy trì hoạt động của mạch qua tiếp điểm K2.

▪ Tắt máy:

- Ấn nút Pb0, cuộn hút công tắc tơ K mất điện sẽ nhả các tiếp điểm K1 và K2, động cơ bị ngắt điện - ngừng hoạt động.
- Cắt áp tô mát.

▪ Bảo vệ quá tải:

Khi động cơ có sự cố (quá tải, mất pha ...) làm cho dòng điện qua phần tử đốt nóng của rơ le nhiệt tăng cao, tác động (nhả) tiếp điểm ol làm mạch điều khiển mất điện, bảo vệ an toàn cho động cơ.

Trong một hệ thống điều khiển phức tạp, ta có thể thay thế mạch điện sử dụng các khí cụ như trên bằng một PLC Mitsubishi. Khi đó việc đấu nối giữa các thiết bị sẽ gọn hơn, ít dây nối hơn rất nhiều, dễ ràng điều khiển, giám sát và sửa chữa.

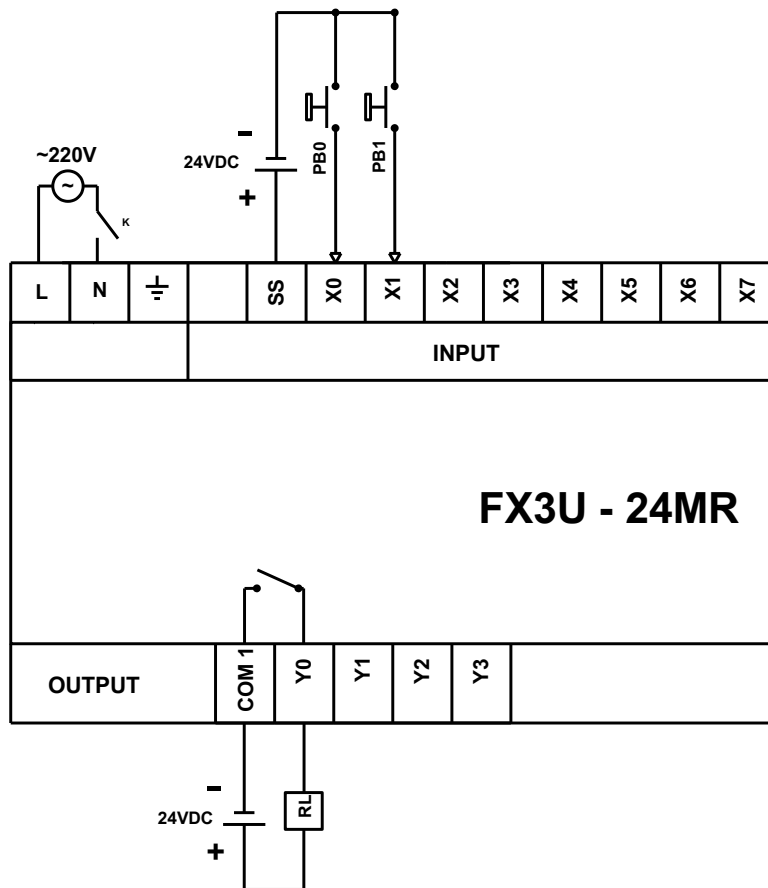
III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

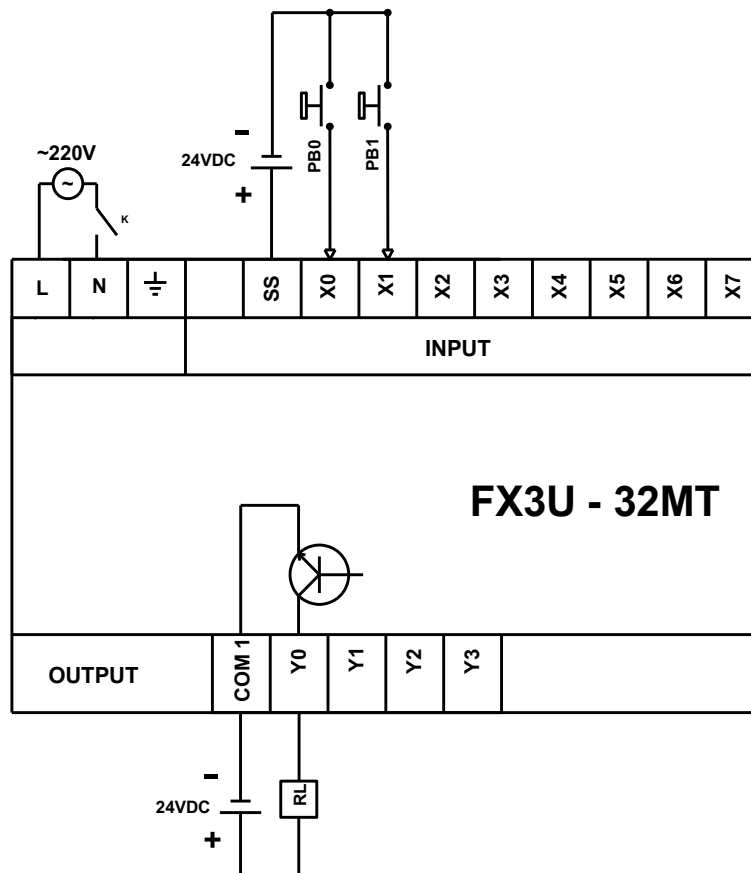
- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
2	Động cơ xoay chiều 3 pha	1 cái
3	Bộ dây cắm trên mô hình	1 bộ
4	Máy tính cài đặt sẵn phần mềm	1 bộ
5	Bộ đồ nghề dụng cụ	1 bộ

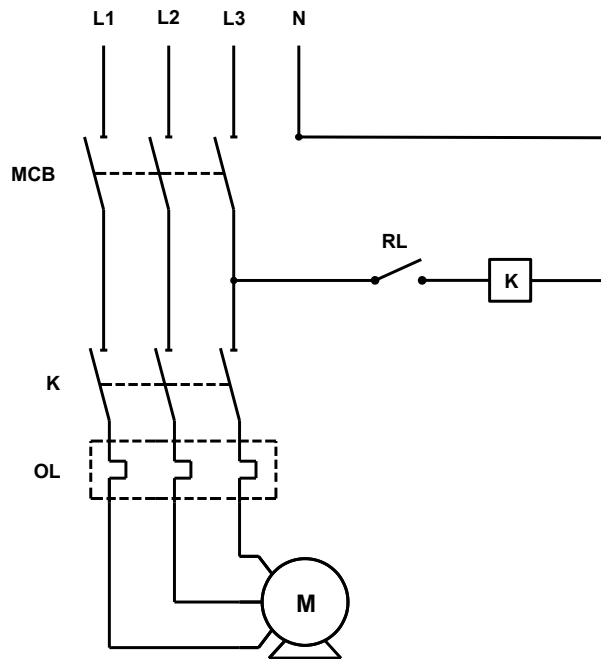
2. Sơ đồ kết nối thực hành



Hình 2.5-2. Sơ đồ kết nối vào ra cho PLC loại đầu ra Rơ le



Hình 2.5-3. Sơ đồ kết nối vào ra cho PLC loại đầu ra Transisto



Hình 2.5-4. Sơ đồ mạch động lực

3. Trình tự thực hiện

- Bước 1: Kết nối phần cứng

Cắm dây nối mạch điều khiển và mạch động lực trên bàn thực hành theo sơ đồ hình 2.5-2, 2.5-3, 2.5-4.

- Bước 2: Tạo project mới

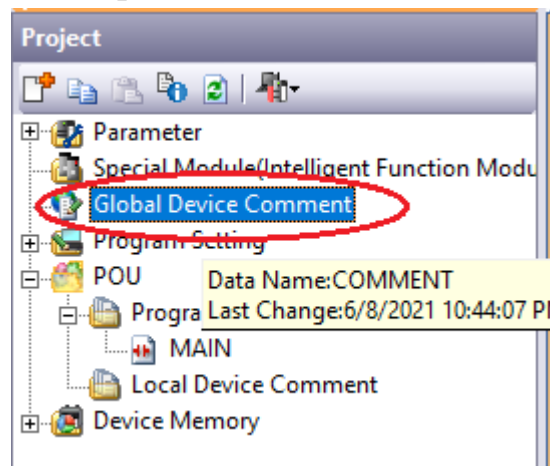
Khởi động phần mềm Gx-Works 2 và tạo project

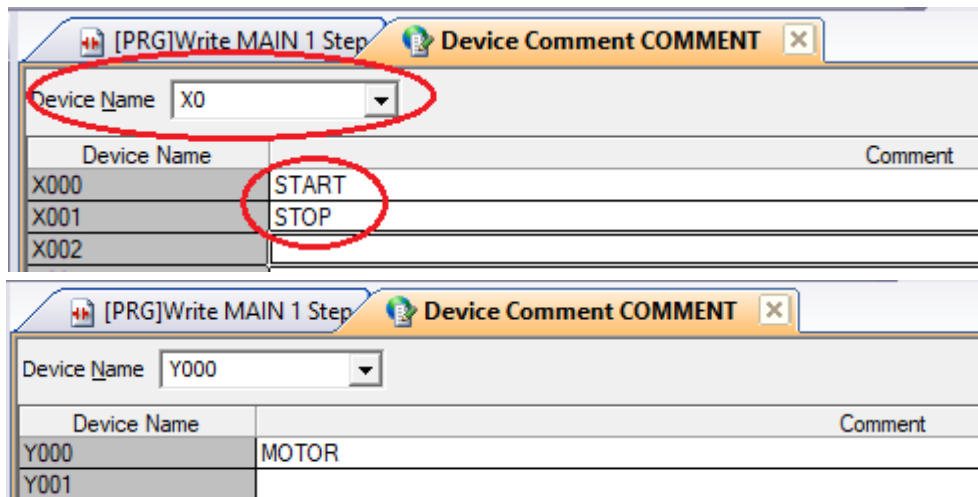
- Bước 3: Viết chương trình điều khiển

+ Lập bảng và đặt tên gọi nhớ cho các đầu vào/ra:

TT	Tên	Địa chỉ	Ý nghĩa
1	START	X000	Chạy động cơ
2	STOP	X001	Dừng động cơ
3	MOTOR	Y000	Động cơ

Tại cửa sổ Project click đúp chuột vào Global Device Comment:

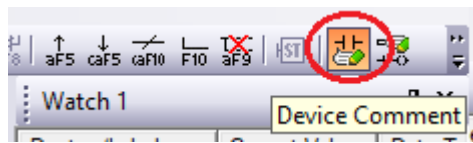




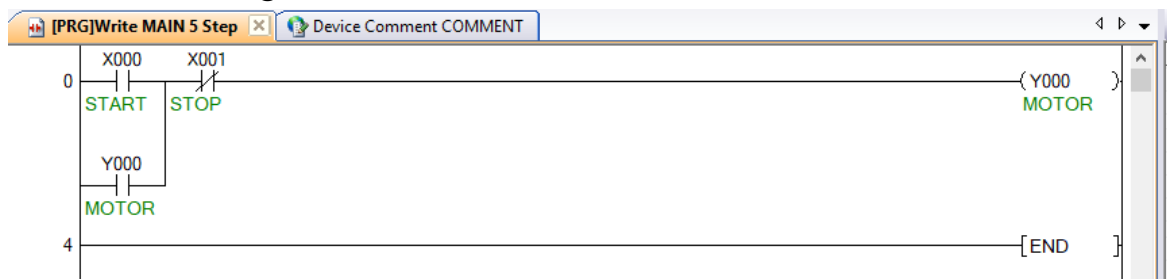
Hình 2.5-5. Device Comment

Tại cửa sổ Device Comment, lần lượt nhập địa chỉ X0, X1, Y0 vào ô Device Name, sau đó nhập tên thiết bị vào ô Comment bên cạnh (Chú ý: không nên gõ tiếng việt)

+ Chuyển sang cửa sổ MAIN để viết chương trình lập trình, chọn vào biểu tượng Device Comment để bật chế độ hiển thị tên thiết bị:



+ Viết chương trình điều khiển:



Hình 2.5-6. Chương trình điều khiển

- Bước 4: Biên dịch và tải chương trình vào PLC

- + Nhấn F4 để biên dịch chương trình;
- + Cắt aptomat mạch động lực để đảm bảo an toàn;
- + Click vào Write to PLC để tải chương trình vào PLC.

- Bước 5: Vận hành mạch:

- + Đo kiểm tra các kết nối tránh chạm chập;
- + Đóng aptomat 3 pha cấp nguồn cho mạch động lực;
- + Chọn biểu tượng Start Monitoring để bật chế độ giám sát online



+ Nhấn Pb0 để chạy động cơ;

+ Nhấn Pb1 để dừng động cơ.

+ Chú ý quan sát các đèn báo IN, OUT trên mặt PLC và tín hiệu của các tiếp điểm trên phần mềm.

4. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không có tín hiệu đèn báo IN khi nhấn Pb0, Pb1	Đầu vào nối không đúng kiểu Sink hoặc Source	Đấu lại theo sơ đồ hình 2.5-2
Nhấn Pb0 động cơ chạy, nhả ra thì động cơ dừng	Mất duy trì	Bổ sung tiếp điểm NO của Y0 như hình Hình 2.5-6

IV. TIÊU CHÍ ĐÁNH GIÁ

- Nguyên lý làm việc của mạch điều khiển, bảo vệ động cơ KĐB 3 pha bằng bộ khởi động từ đơn []
- Kiểm tra kết nối thiết bị với đầu vào - ra []
- Trình tự khởi tạo và lập trình điều khiển máy bơm []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Sơ đồ nguyên lý
3. Sơ đồ đấu nối
4. Trình tự thực hiện
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

1. So sánh sơ đồ điều khiển giữa phương pháp dùng khí cụ điện và dùng PLC?
2. Viết chương trình điều khiển động cơ không đồng bộ ba pha điều khiển tại 2 vị trí?
3. Viết chương trình điều khiển động cơ không đồng bộ ba pha với các chức năng: chạy thuận, chạy nghịch, dừng động cơ?

Bài thực hành số 6
LẬP TRÌNH HMI

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

Trình bày được trình tự khởi tạo, thiết kế giao diện HMI và kết nối với PLC.

Về kỹ năng:

- Thiết kế được giao diện chức năng cơ bản cho HMI;
- Lập chương trình điều khiển;
- Kết nối được HMI với PLC;
- Nhận biết và khắc phục được một số lỗi thường gặp trong quá trình lập

trình HMI.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình đấu nối, lập trình HMI;

- Tự đánh giá được kết quả thực hành của bản thân và hướng dẫn, đánh giá cho người khác thực hành.

II. TÓM TẮT LÝ THUYẾT

1. Giới thiệu về HMI

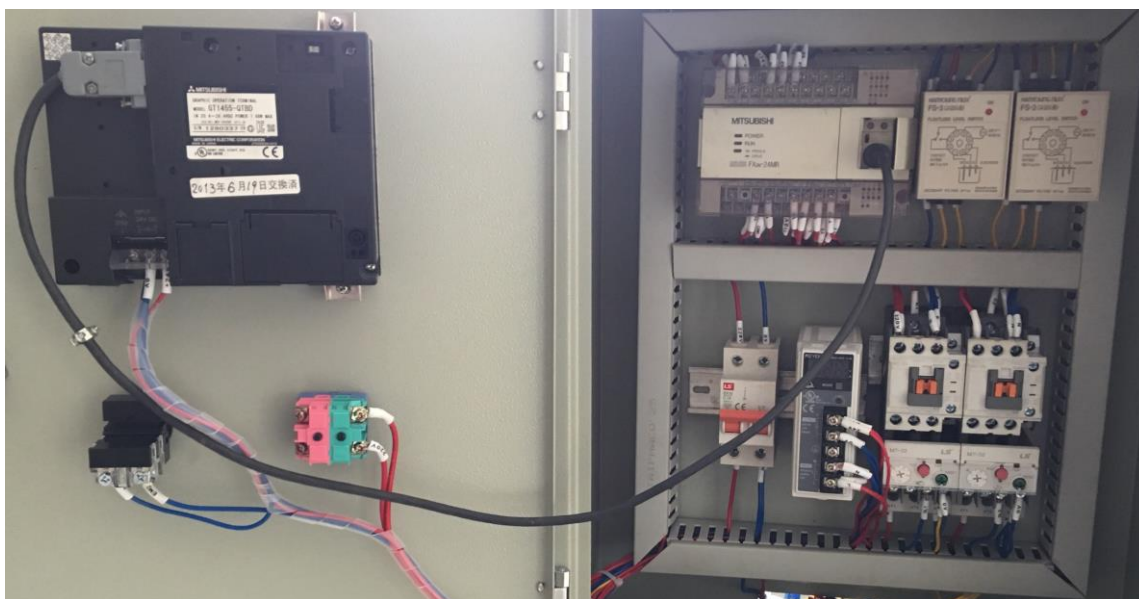
1.1. Giới thiệu tổng quan

Màn hình hay còn được gọi là HMI (Human Machine Interface) được ứng dụng rất nhiều trong công nghiệp. Màn hình gồm nhiều chủng loại khác nhau của các hãng như Mitsubishi, Siemen, Omron, Delta,... Mỗi hãng sản xuất đều có một số tính năng như bộ lập trình bằng tay, giám sát quá trình sản xuất, truy cập các thông số, dữ liệu cài đặt.....

Ngoài các tính năng trên màn hình HMI còn có thể cho nhiều tính năng khác như đồ họa để mô phỏng các thiết bị trong quá trình sản xuất giúp người vận hành các thiết bị có cái nhìn trực quan hơn về hệ thống sản xuất, giúp họ dễ thao tác kiểm tra hệ thống nhanh và hiệu quả hơn.

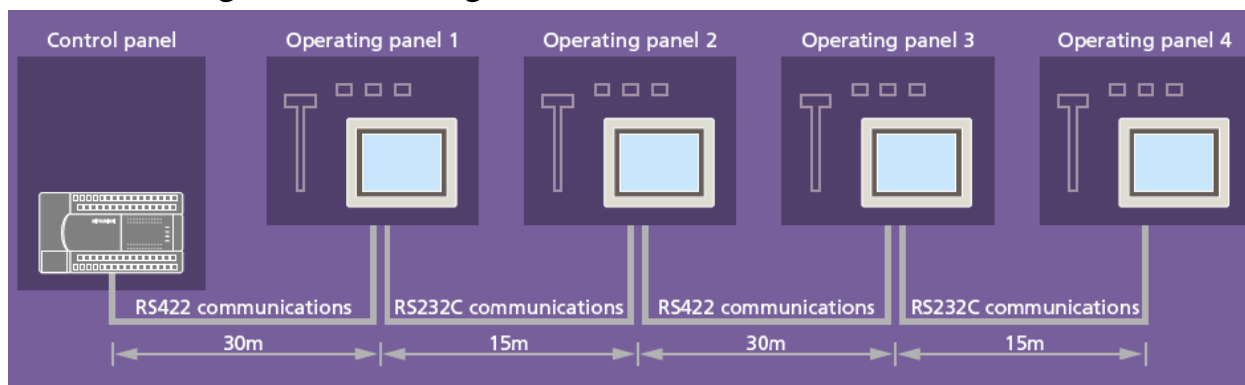
Một số loại màn hình HMI của hãng Mitsubishi:

- Loại FX-10DU đến FX-50DU.
- Loại GOT-F900 series (handy và Touch Screen).
- Loại F940GOT-LWD-E.
- Loại A800 Series.
- Loại GOT-1000 Series.



Hình 2.6-1. HMI kết nối PLC để điều khiển, giám sát trạm xử lý nước

HMI kết nối với PLC qua các chuẩn truyền thông RS232, RS422, RS485 để điều khiển và giám sát hệ thống.



Hình 2.6-2. Kết nối nhiều trạm HMI với PLC

Loại màn hình được dùng để thực hành trong bài học này là loại Loại GOT-1000 Series.

1.2. Giới thiệu HMI GOT-1000 Series

Loại HMI GOT-1000 bao gồm nhiều model khác nhau: GT16, GT15, GT14, GT12, GT11, GT10. Tuy theo model mà HMI sẽ có các thông số về: kích thước màn hình, dung lượng bộ nhớ, hỗ trợ kết nối khác nhau.



Hình 2.6-3. HMI Mitsubishi GOT-1000

HMI sẽ được kết nối với máy tính thông qua cáp lập trình để Upload /Download chương trình vào HMI.

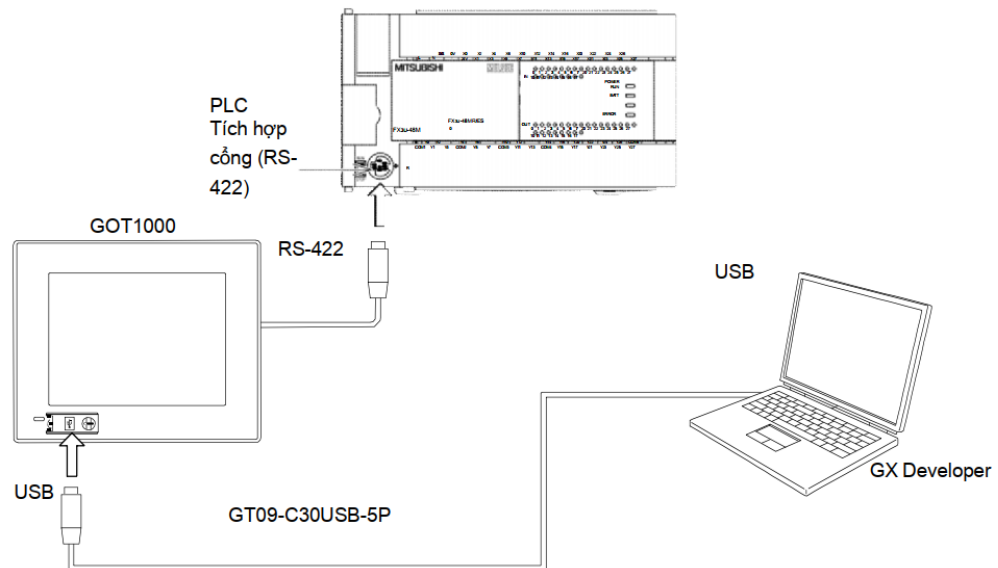
HMI cũng được kết nối với PLC bằng cáp chuyên dụng, sử dụng chuẩn truyền thông RS422.



Hình 2.6-4a. Cáp kết nối HMI với PLC



Hình 2.6-4b. Cáp lập trình HMI



Hình 2.6-5. Sơ đồ kết nối HMI Mitsubishi

1.3. Giới thiệu HMI Weintek

- HMI Weintek có một số mã như sau:

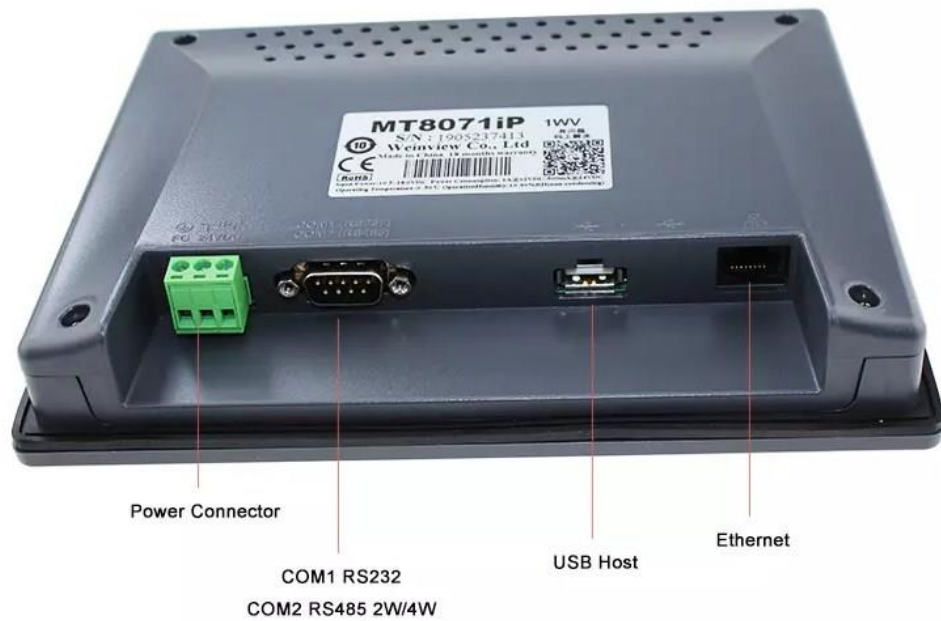
+ iP Series: MT8051iP (480 x 272), MT8071iP (800 x 480), MT8102iP (1024 x 600);

+ iE Series: MT8051iE (480 x 272), MT8071iE (800 x 480), MT8102iE (1024 x 600);

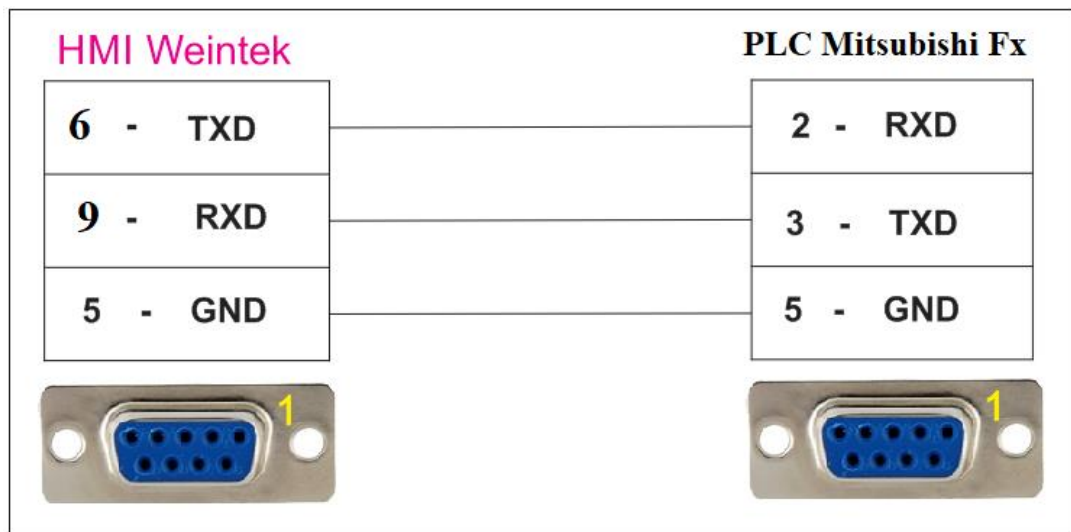
+ Ngoài ra còn nhiều Series khác.



Hình 2.6-6a. HMI Weintek mặt trước



Hình 2.6-6b. HMI Weintek mặt sau



Hình 2.6-7. Cáp kết nối RS232 HMI Weintek MT8071iP–PLC mitsubishi

- Thông số kỹ thuật HMI Weintek MT8071iP:

+ *Hiển thị:*

LCD hiển thị: 7 inch TFT

Độ phân giải (WxH dots): 800×480

Tuổi thọ LCD: >30,000 hr.

Màn hình LCD: 16.000 màu

+ *Cấu hình:*

Bộ nhớ(MB): 128

RAM (MB): 64
 USB Host: USB 2.0 x 1
 USB Client: Mico USB
 Khe cắm thẻ SD: Không
 Ethernet: có
 COM1 RS-232, COM2 RS-485

+ Nguồn: 24 ± 20% VDC

- HMI Weintek MT8071iP được kết nối với máy tính bằng cáp Ethernet để Download và Upload chương trình.

2. Giới thiệu phần mềm thiết kế giao diện HMI

Các phần mềm dùng để thiết kế giao diện:

- Phần mềm GT Designer sử dụng cho HMI Mitsubishi.
- Phần mềm EasyBuilder Pro sử dụng cho HMI Weintek.

Đây là các phần mềm hỗ trợ thiết kế giao diện và lập trình cho các dòng HMI tương ứng với các hãng khác nhau. Phần mềm cho phép người dùng thiết kế giao diện với các chức năng điều khiển, giám sát từ cơ bản đến nâng cao như: nút nhấn, công tắc, đèn báo, hiển thị thông số giám sát, biểu đồ, hình ảnh động 3D mô tả hoạt động của hệ thống...

Ngoài ra các phần mềm còn hỗ trợ cho việc mô phỏng trên máy tính thông qua phần mềm hỗ trợ simulation để người dùng dễ dàng mô phỏng, kiểm chứng kết quả trước đưa vào sử dụng trong hệ thống thực tế.

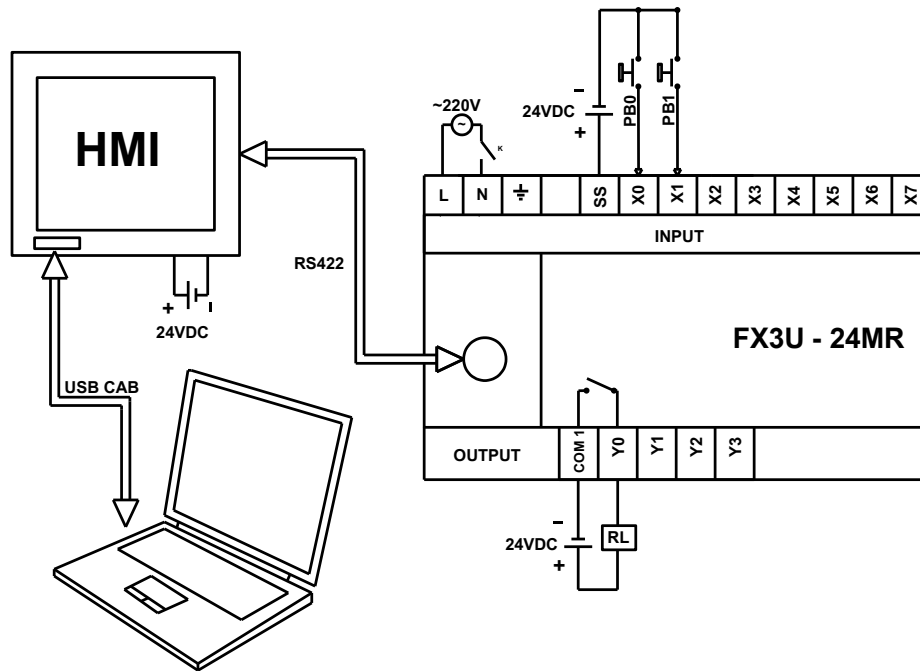
III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
2	HMI GT1455-QTBD	1 cái
3	HMI Weintek MT8071iP	1 cái
4	Bộ dây cắm	1 bộ
5	Máy tính	1 bộ
6	Bộ đồ nghề dụng cụ	1 bộ

2. Sơ đồ kết nối



Hình 2.6-8. Sơ đồ kết nối HMI hãng Mitsubishi

3. Các bước thực hiện

3.1. Thiết kế giao diện cho HMI mitsubishi GOT1000

- **Bước 1:** Kết nối thiết bị phần cứng

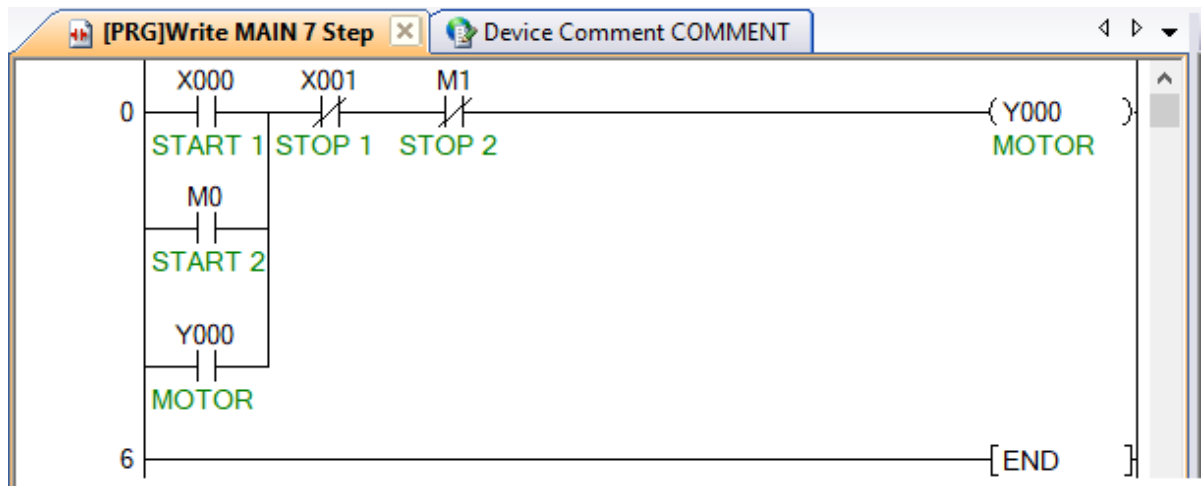
- + Sử dụng dây cắm, cáp kết nối phần cứng theo sơ đồ hình 2.6-7
- + Kết nối mạch động lực động cơ xoay chiều 3 pha như Hình 2.5-4 ở Bài thực hành số 5

- **Bước 2:** Lập chương trình điều khiển:

- + Mở phần mềm Gx-Works 2 và tạo project mới
- + Đặt tên gọi nhớ cho các đầu vào/ra:

TT	Tên	Địa chỉ	Ý nghĩa
1	START 1	X000	Chạy động cơ
2	STOP 1	X001	Dừng động cơ
3	START 2	M0	Chạy động cơ từ HMI
4	STOP 2	M1	Dừng động cơ từ HMI
5	MOTOR	Y000	Động cơ

- + Lập chương trình điều khiển: điều khiển động cơ từ 2 nơi, từ nút nhấn và từ HMI.



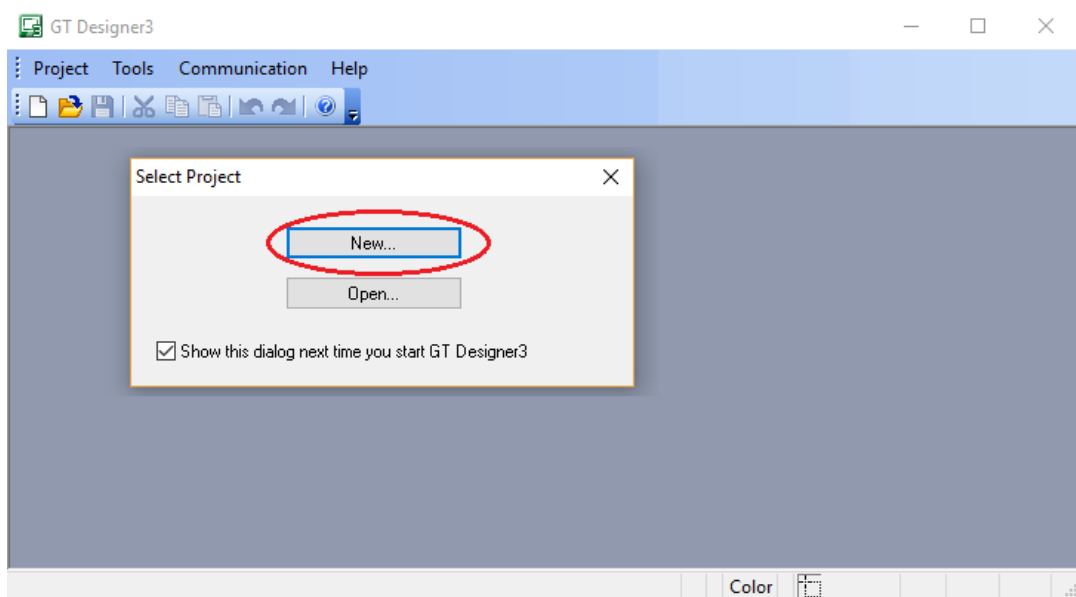
Hình 2.6-9. Chương trình điều khiển PLC

+ Biên dịch chương trình.

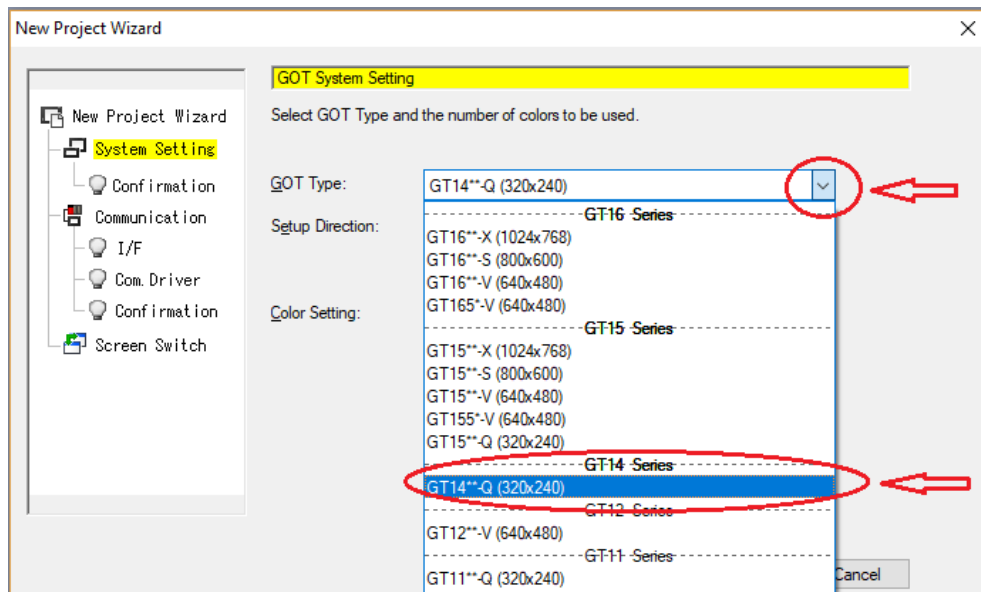
+ Nạp chương trình vào PLC

- **Bước 3: Cài đặt tham số cho HMI:**

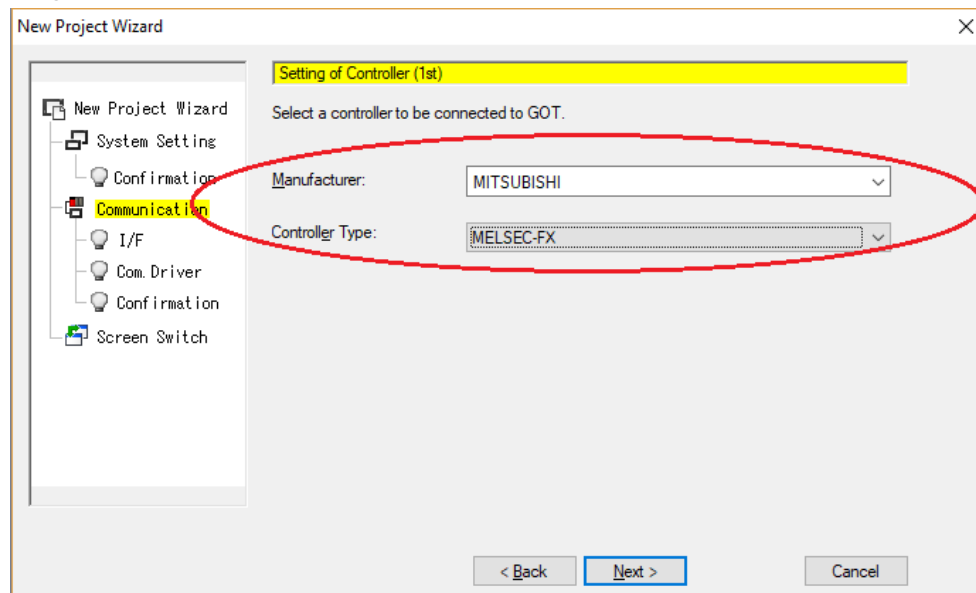
+ Khởi động phần mềm GT Designer  xuất hiện cửa sổ sau:



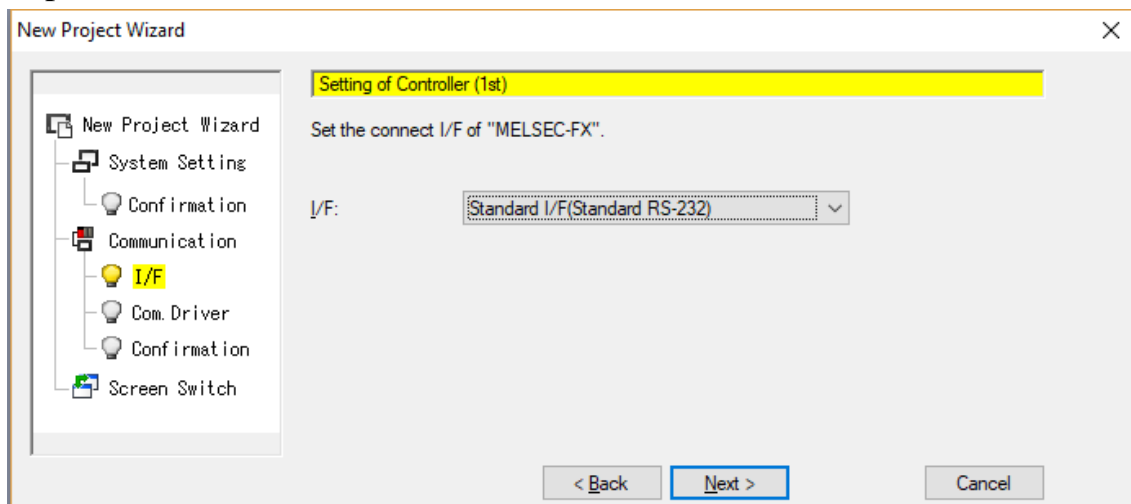
+ Click chọn New... → Next → chọn model HMI tại ô GOT type → Next



+ Chọn hãng PLC Mitsubishi → chọn loại PLC MELSEC – FX → Next:



+ Chọn kiểu kết nối giữa HMI và PLC (RS232 hoặc RS422/485) tùy thuộc loại cáp kết nối → Next:

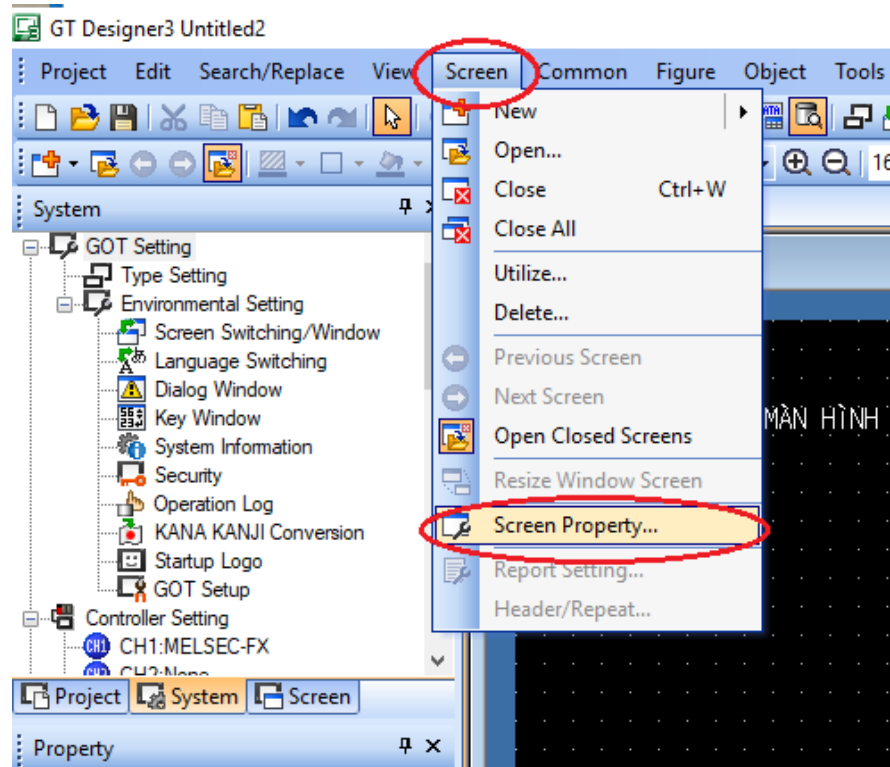


+ Tiếp tục nhấn Next tại các cửa sổ tiếp theo → nhấn Finish để bắt đầu thiết kế giao diện.

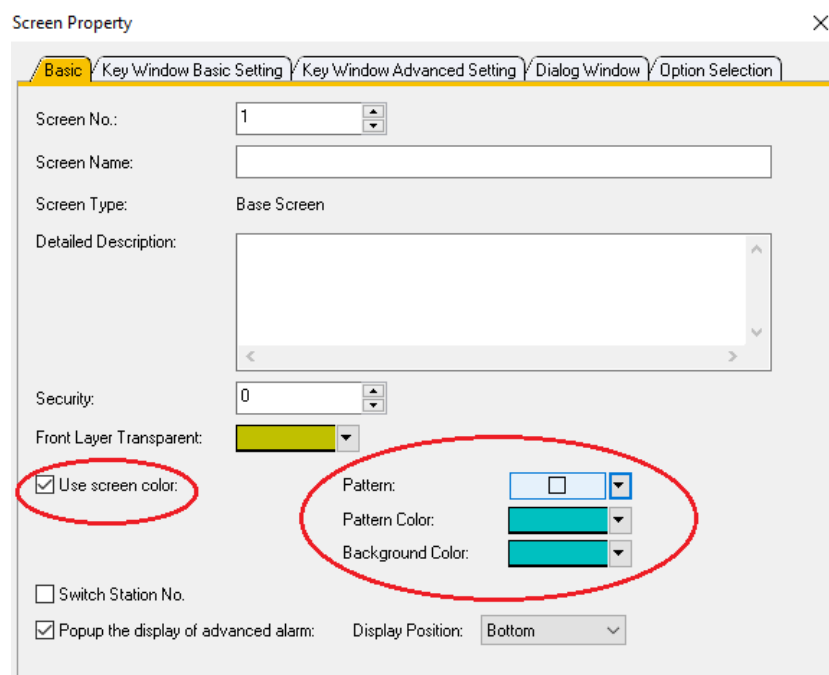
- Bước 4: Thiết kế giao diện HMI:

*** Thiết lập màu nền giao diện:**

→ Chọn Screen trên thanh công cụ → chọn Screen Property (Hoặc click chuột phải vào nền soạn thảo → chọn Screen Property).

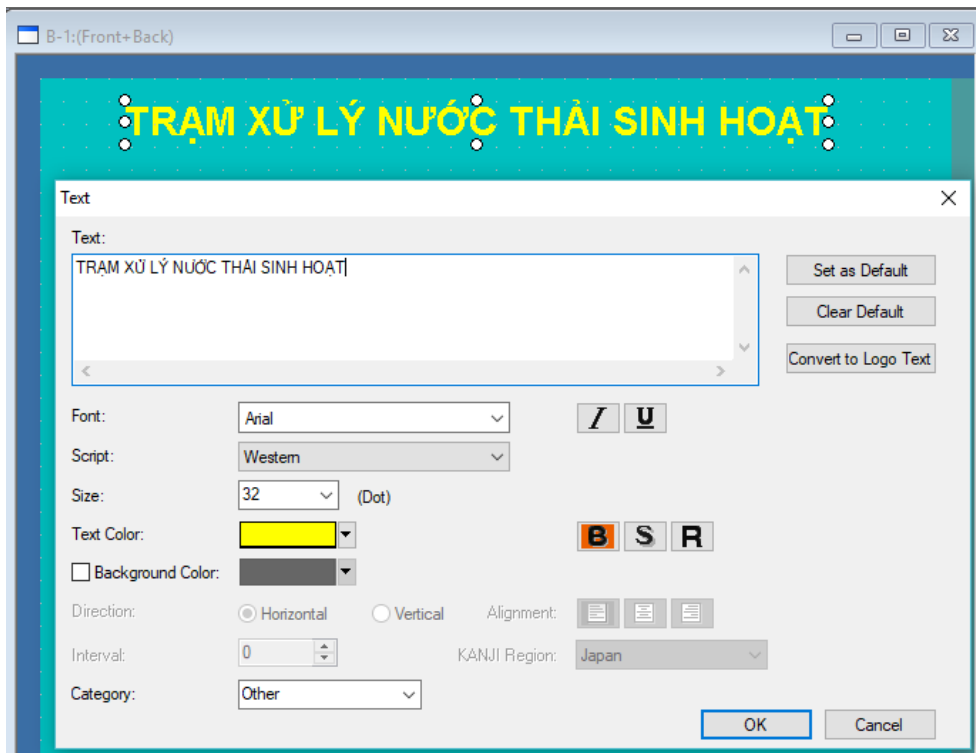


→ Tại cửa sổ mới xuất hiện tick chọn Use Screen Color → chọn màu Background và Pattern color mong muốn → OK:



*** Cách đặt dòng text trong giao diện:**

→ Chọn vào Figure trên thanh công cụ → chọn Text → click chuột vào nền soạn thảo.

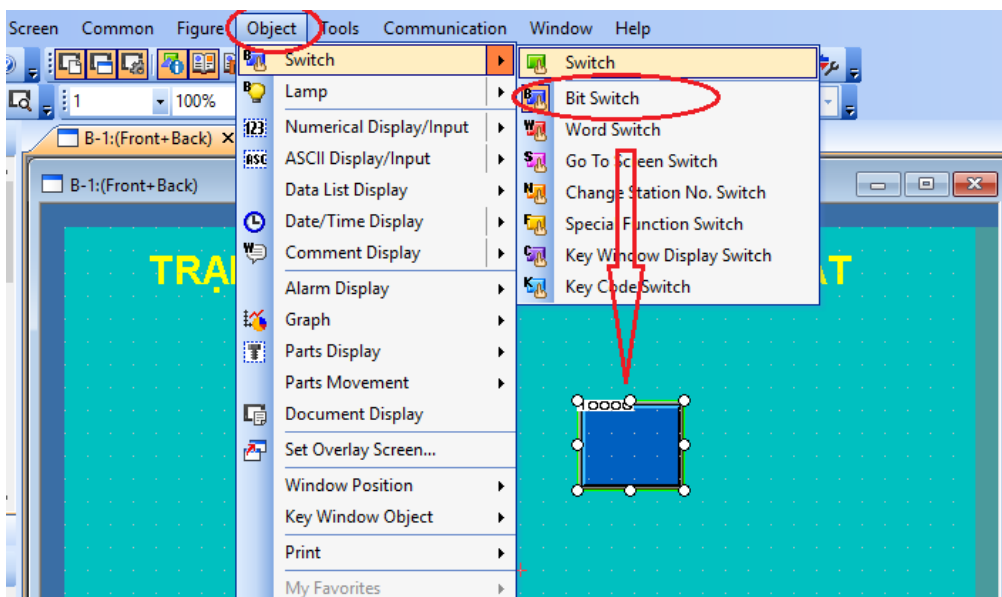


→ Tại cửa sổ Text: nhập chữ vào ô Text → chọn font chữ, cỡ chữ, màu chữ, màu nền → nhấn OK.

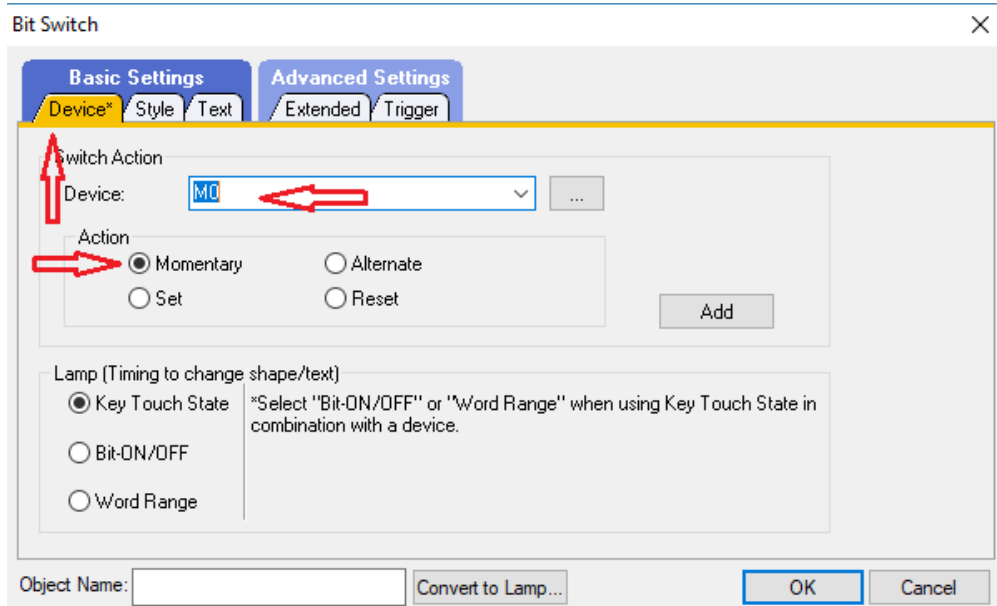
→ Có thể thay đổi vị trí của text bằng cách click, giữ chuột và kéo đến vị trí mong muốn.

*** Cách tạo nút nhấn:**

→ Chọn Object trên thanh công cụ → Switch → Bit Switch → nhấn giữ và kéo chuột để đặt vị trí và kích thước của nút nhấn:

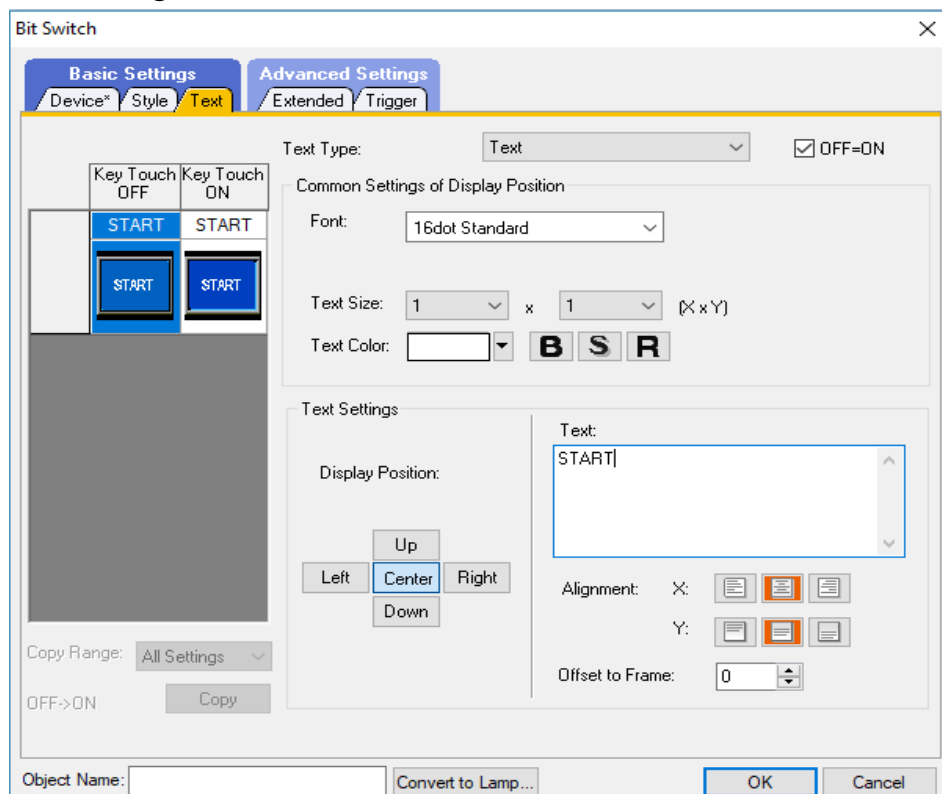


→ Click đúp chuột vào nút nhấn để cài đặt thông số. Tại thẻ Device → nhập địa chỉ bit M0 vào ô Device kết nối giữa nút nhấn và bit M0 trên PLC → tick chọn vào Momentary (loại nút nhấn tự nhả)



→ Tại tag Style: chọn màu cho trạng thái của nút nhấn

→ Tại thẻ Text: gõ tên nút nhấn START vào ô Text → chọn OK



→ Tiếp tục tạo nút nhấn STOP tương tự như trên với các tham số:

Device: M1

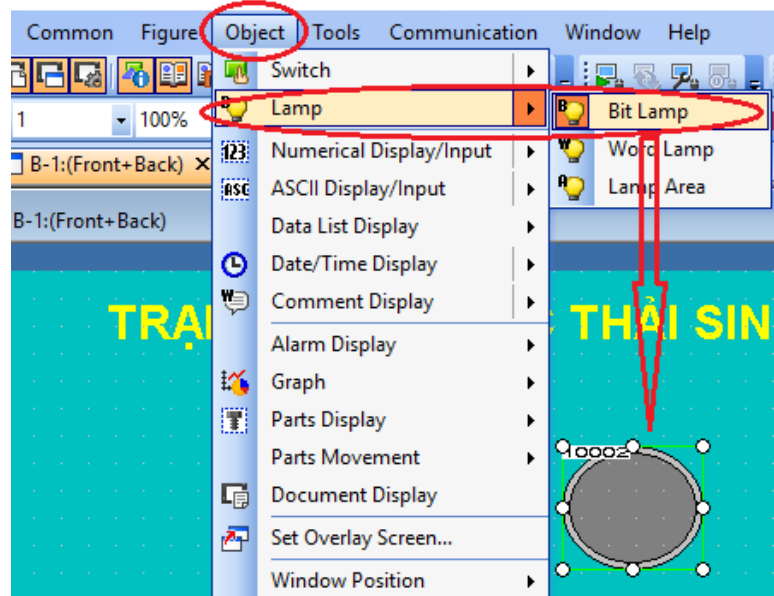
Kiểu nút: Momentary

Màu nút: vàng

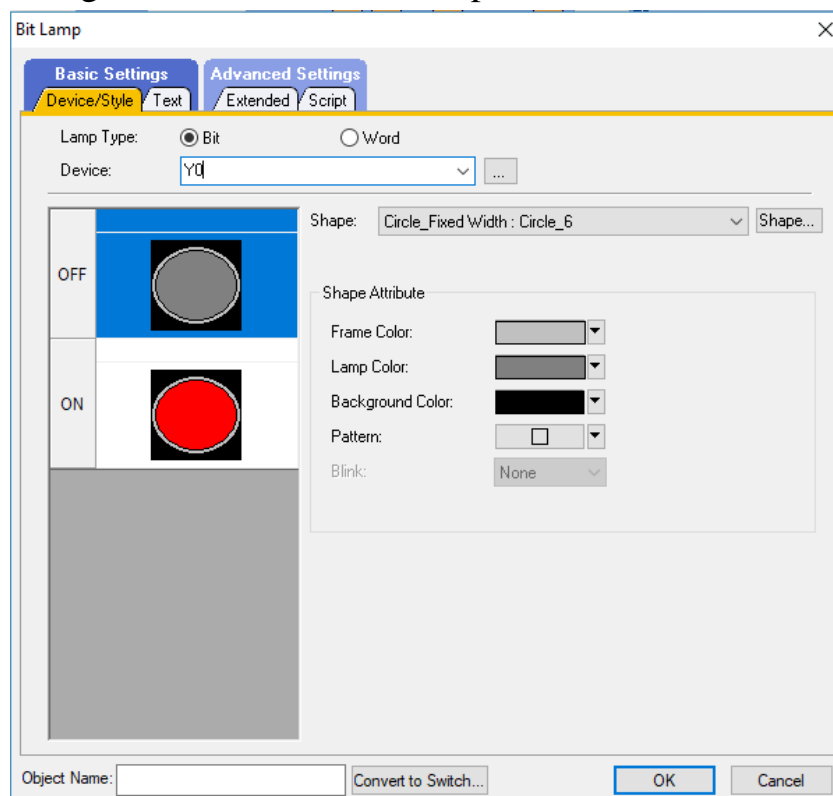
Text: STOP

*** Cách tạo đèn báo**

→ Trên thanh công cụ chọn Object/ Lamp/ Bit Lamp → nhấn giữ và kéo chuột để đặt vị trí và kích thước cho đèn báo:



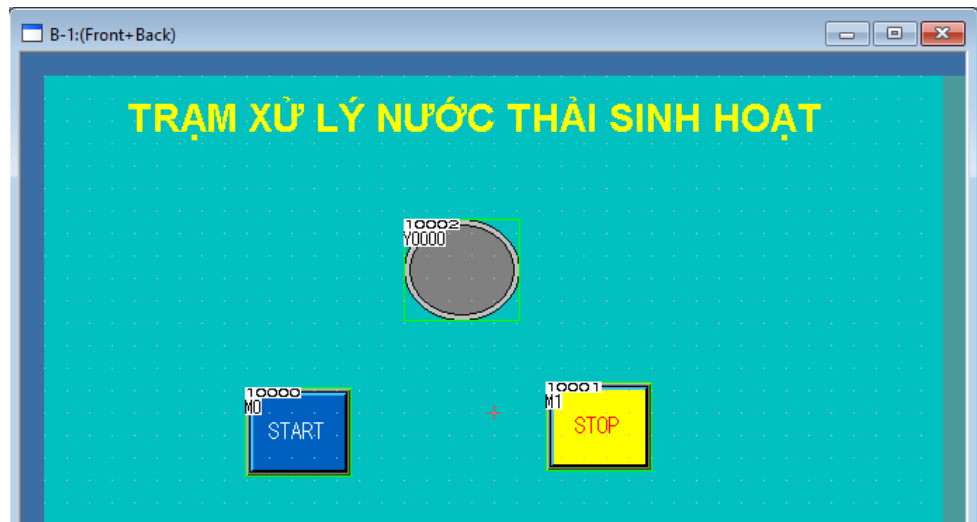
→ Cài đặt thông số cho đèn báo: click đúp chuột vào đèn báo:



→ Nhập Y0 vào ô Device để tạo kết nối giữa đèn báo và trạng thái của Y000 trên PLC

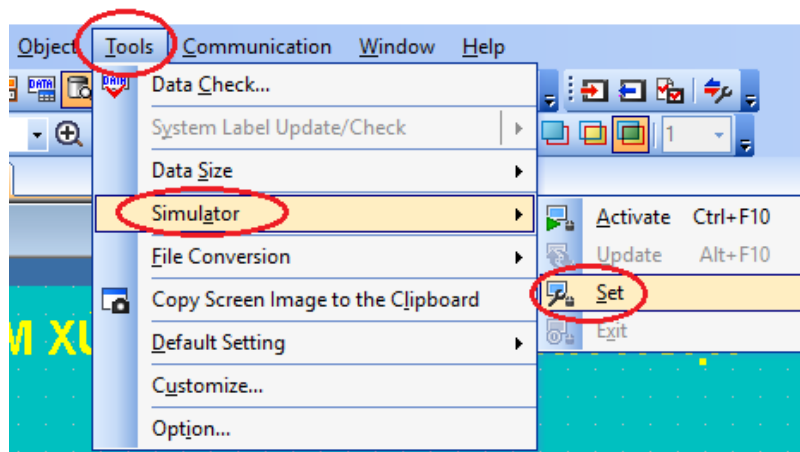
→ Chọn màu cho đèn báo ứng với các trạng thái ON/OFF

* Kết quả thực hiện sau bước 4:

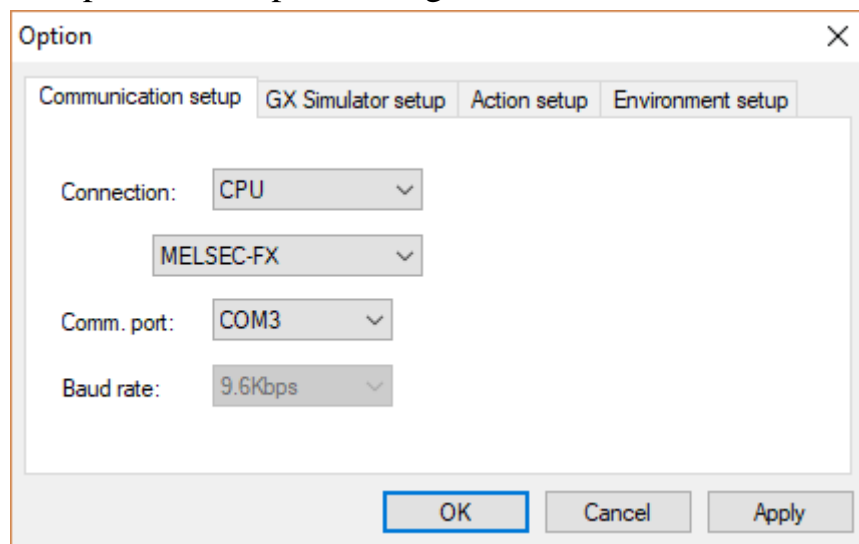


- Bước 5: Chạy mô phỏng:

* Thiết lập kết nối: trên thanh công cụ chọn Tool / Simulator / Set



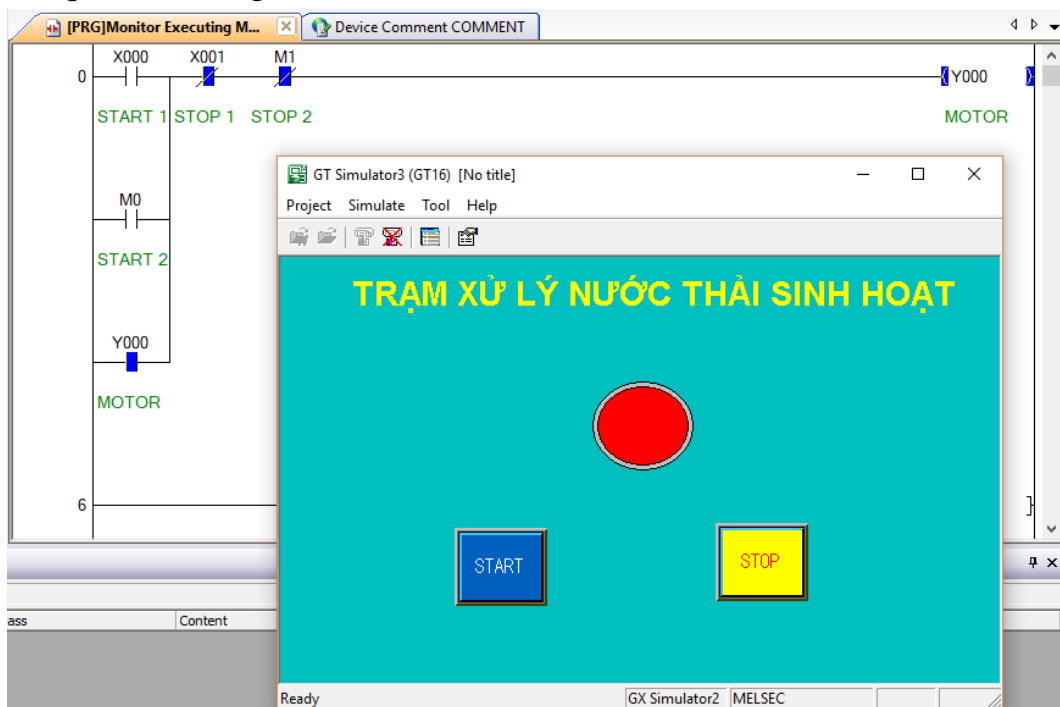
* Tại cửa sổ Option thiết lập các thông số như sau:



→ Connection: chọn CPU nếu chạy PLC thật, chọn GX Simulator 2 nếu chạy PLC ảo trên Gx-Works 2.

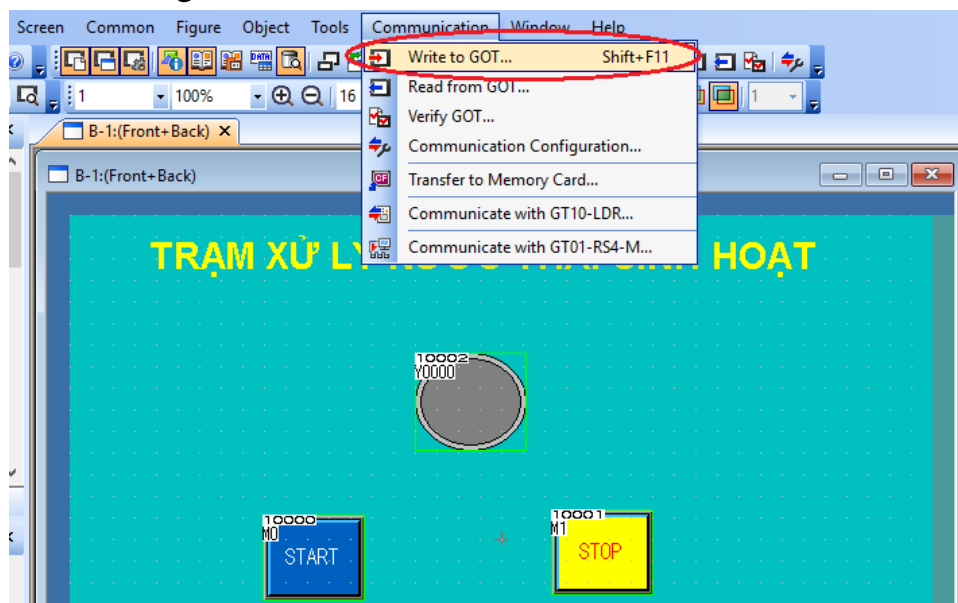
→ Chú ý chọn đúng cổng COM mà PLC đang kết nối với máy tính.

* Chạy mô phỏng: trên thanh công cụ chọn Tool / Simulator / Active (Ctrl + F10). Màn hình ảo sẽ xuất hiện → dùng chuột để nhấn các nút nhấn START, STOP và quan sát trạng thái đèn báo:

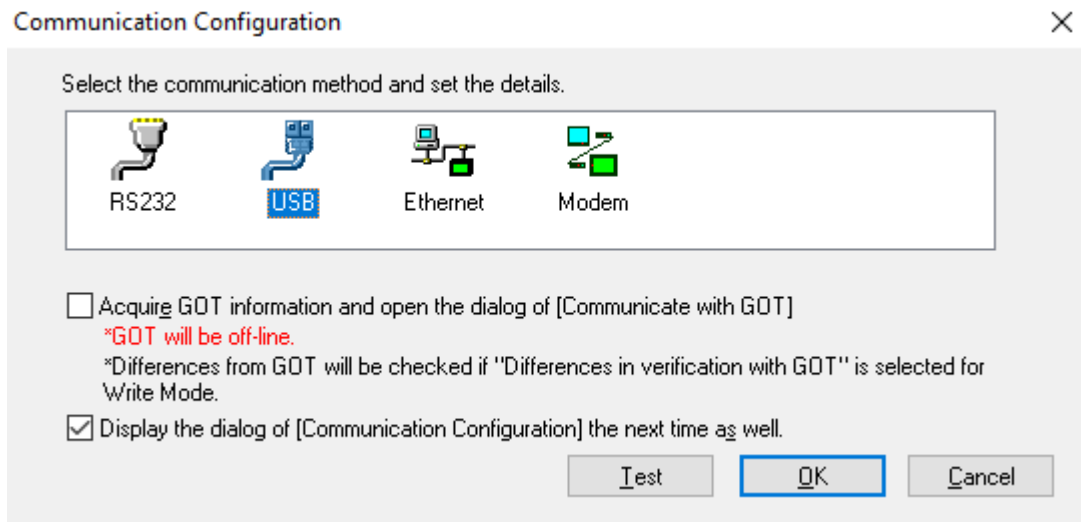


- Bước 6: Download chương trình vào HMI

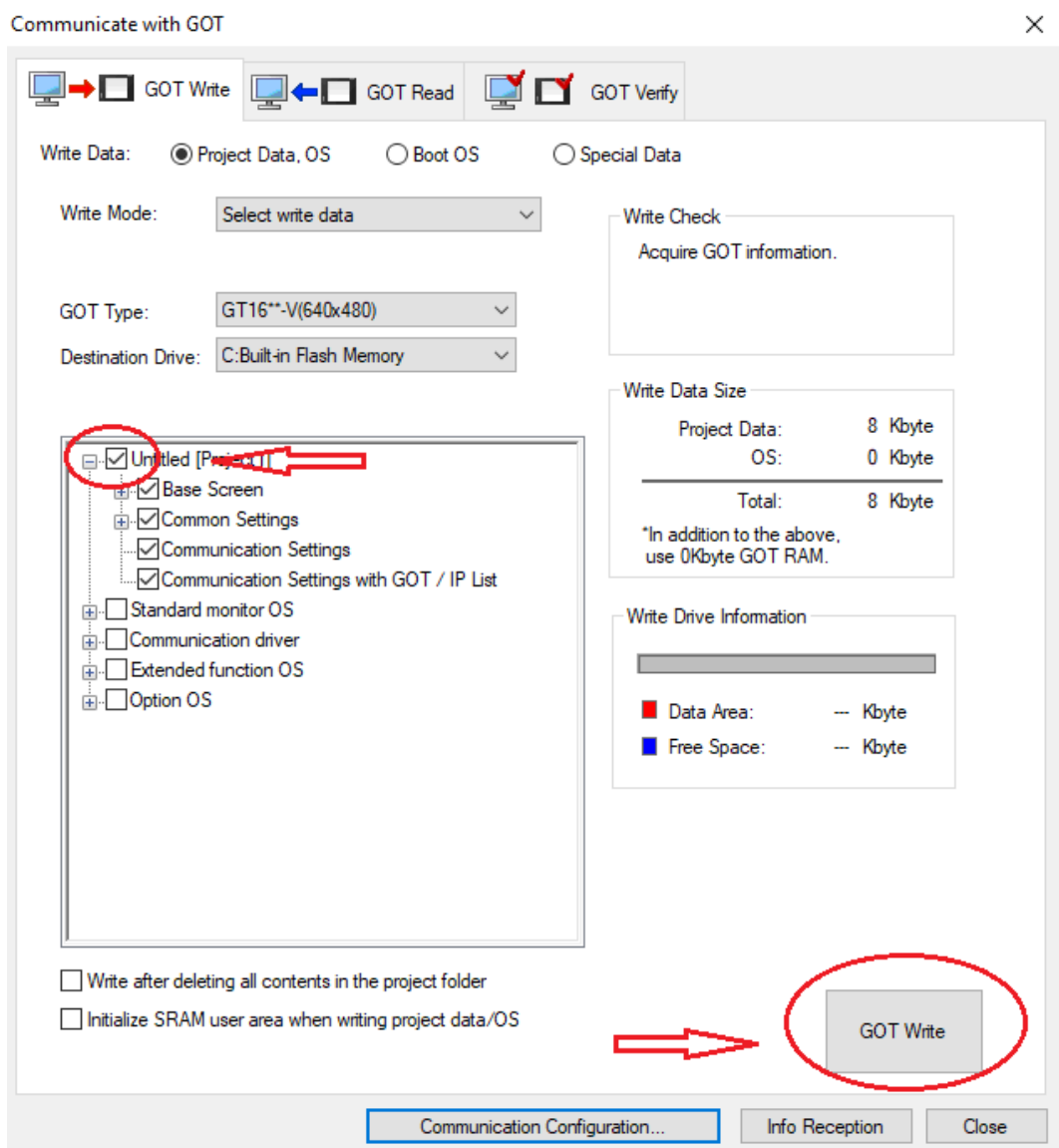
- + Kiểm tra cáp kết nối HMI với máy tính
- + Trên thanh công cụ chọn Communication / Write to GOT



- + Chọn kiểu kết nối USB → Test để kiểm tra kết nối → OK



+ Tại cửa sổ mới xuất hiện, Tick chọn Untitled → chọn GOT Write. Sau khi tải chương trình xong nhấn OK và đóng cửa sổ Communicate lại.



- Bước 7: Vận hành mạch:

+ Kiểm tra các điều kiện an toàn tránh chập cháy

- + Cấp nguồn cho mạch động lực
- + Điều khiển động cơ từ nút nhấn Pb0, Pb1 và từ HMI.

3.2. Thiết kế giao diện cho HMI Weintek MT8071iP

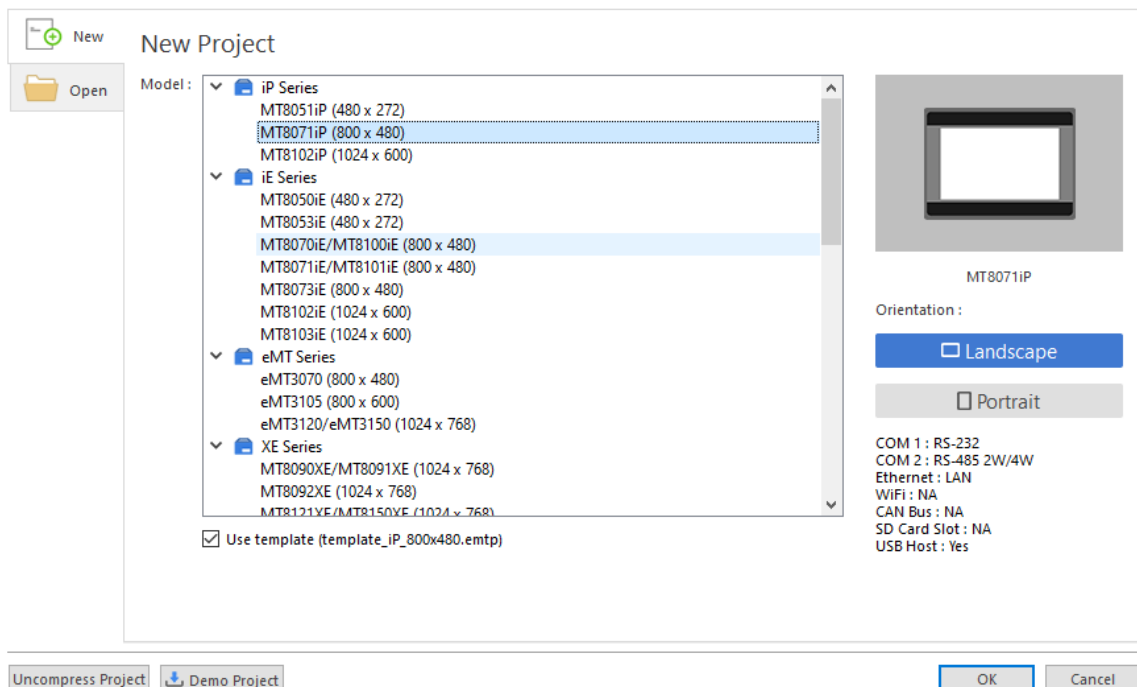
3.2.1. Sử dụng các bước 1, 2 như trên mục 3.1.

3.2.2. Cài đặt địa chỉ IP cho HMI

Nhấn vào biểu tượng cài đặt trên HMI để kiểm tra địa chỉ IP của HMI và chỉnh sửa lại nếu cần thiết

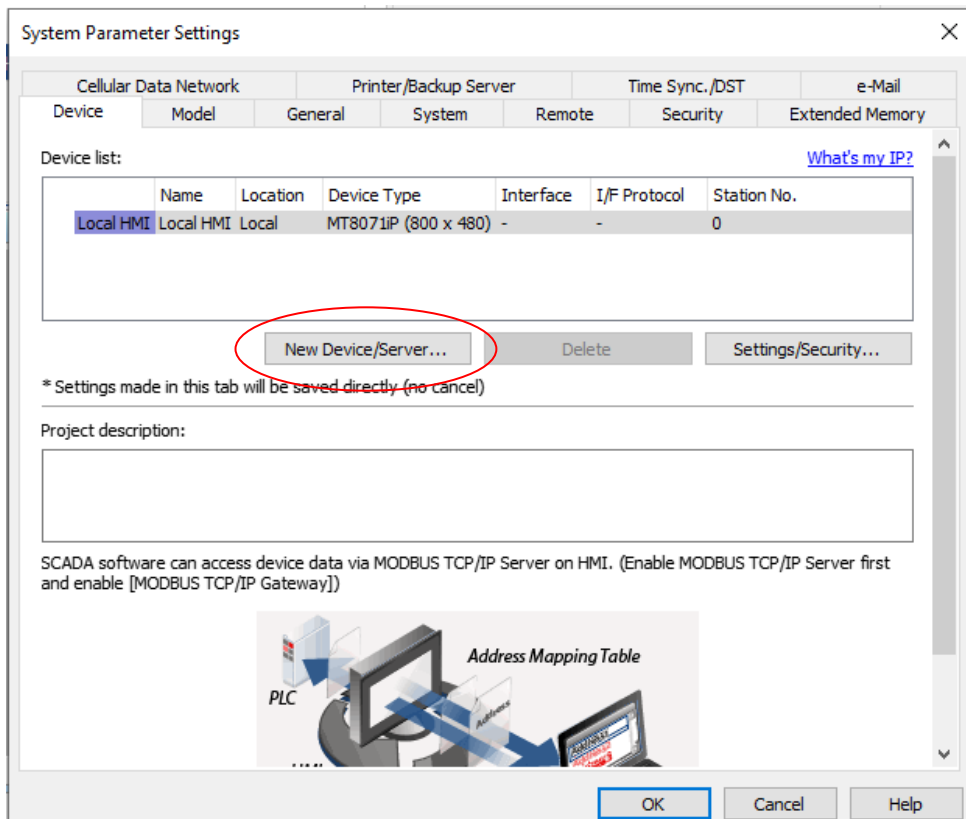
3.2.3. Thiết kế giao diện:

- *Bước 1:* Mở phần mềm Easy Builder Pro
- *Bước 2:* Nhấp chuột vào NEW, xuất hiện giao diện dưới đây, lưu ý lựa chọn đúng Model của màn hình HMI trong ô Model (*Xem hình dưới đây*)

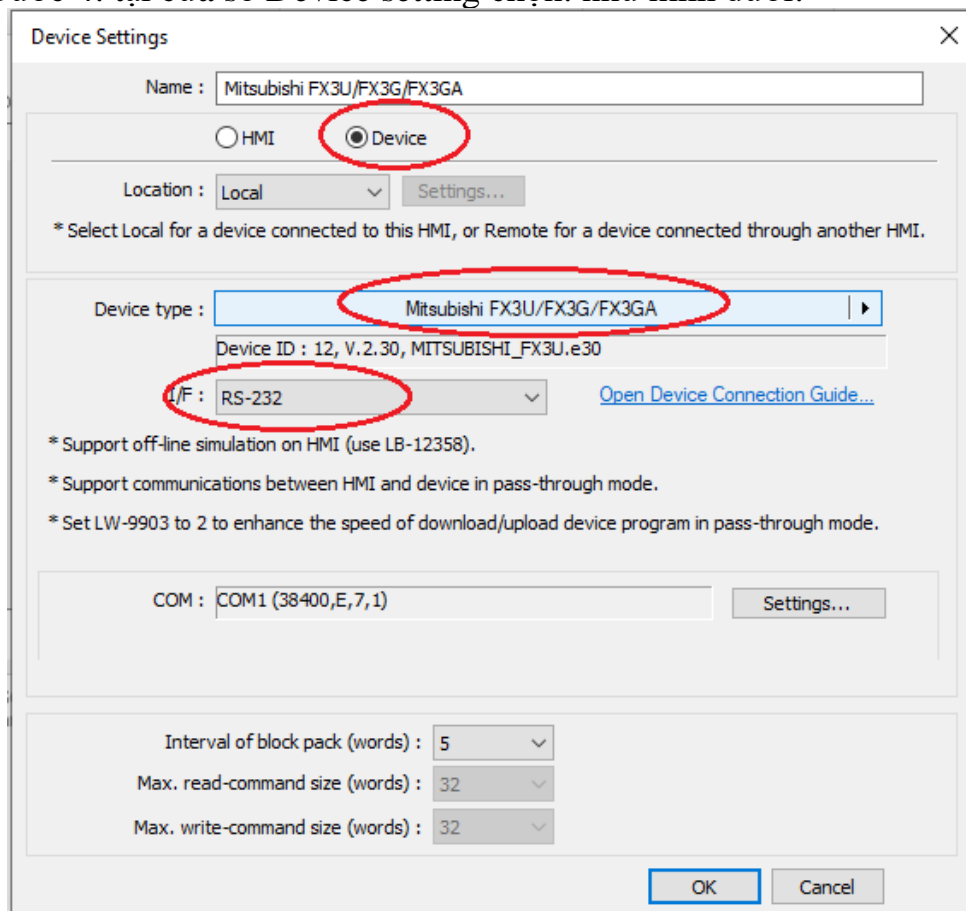


Nhấp chuột vào nút OK để chuyển sang bước 3

- *Bước 3:* Tại cửa sổ System Parameter Setting → chọn New Devive/Server



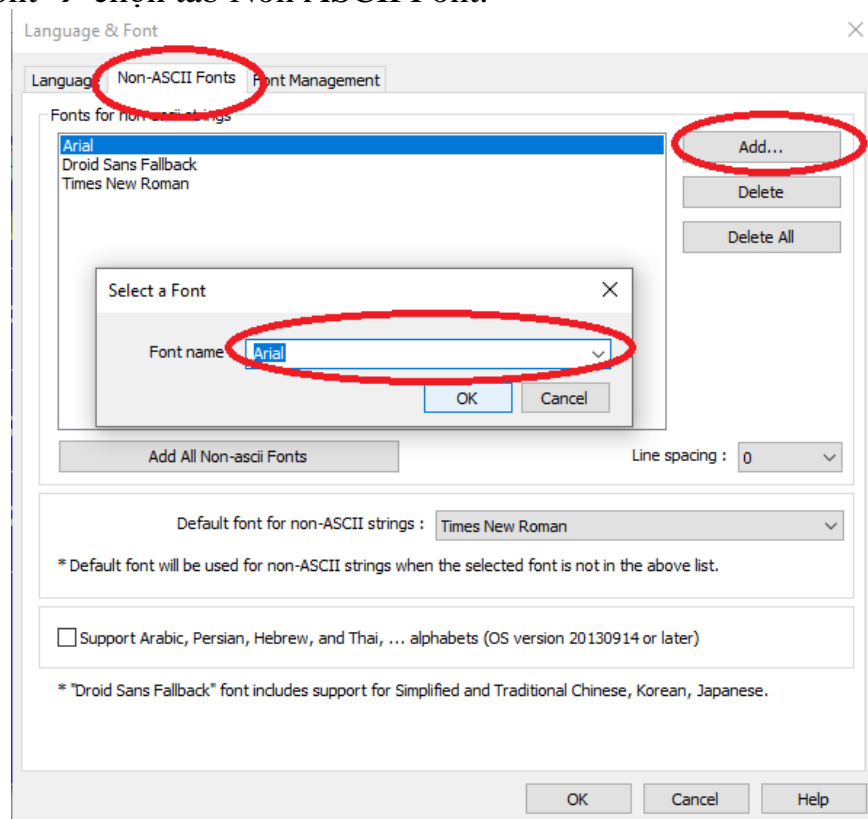
- Bước 4: tại cửa sổ Device setting chọn: như hình dưới:



Tiếp tục nhấn OK để bắt đầu thiết kế giao diện

- Bước 5: Cài đặt font Tiếng Việt:

→ Trên thanh công cụ → Project → chọn biểu tượng  → tại cửa sổ Language & Font → chọn tab Non ASCII Font:



→ Chọn Add → chọn Font chữ (arial, TimeNewroman...) → OK → OK.

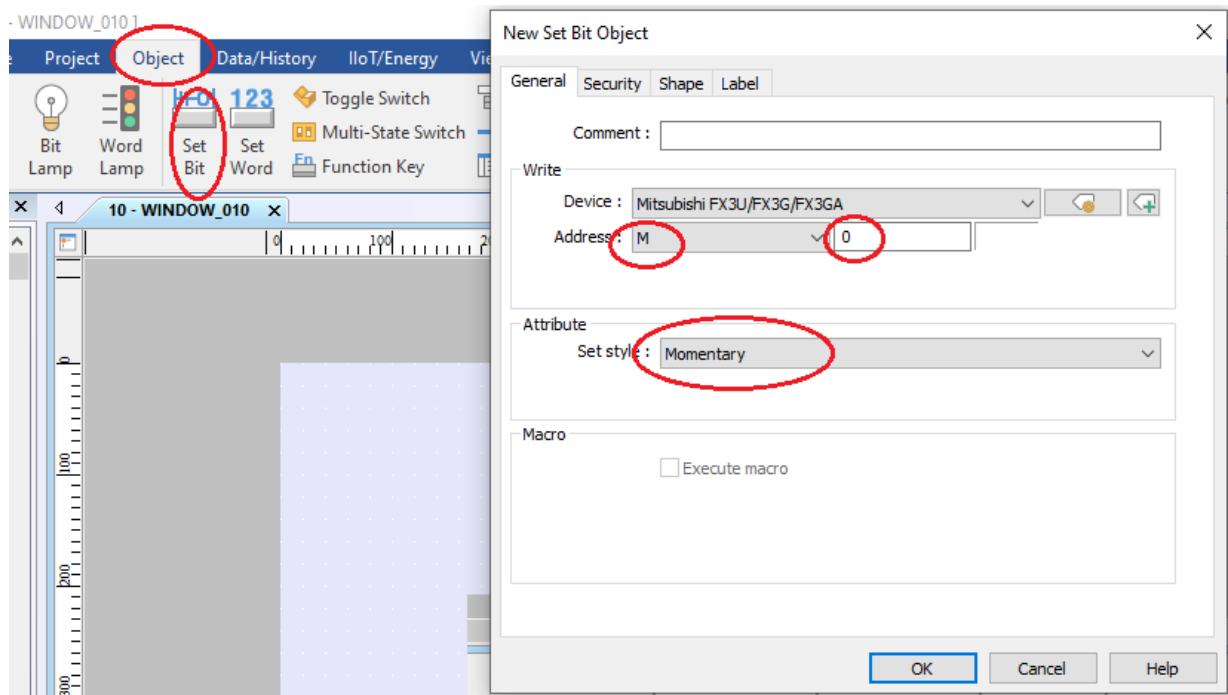
→ Lưu project → khởi động lại phần mềm EasyBuild.

- Bước 6: Tạo nút nhấn:

Trên thanh công cụ → Object → Set Bit → chọn địa chỉ bit cần điều khiển (nên dùng các bit trung gian M_i) → chọn kiểu nút nhấn (ON, OFF, Toggle, Momentary...)

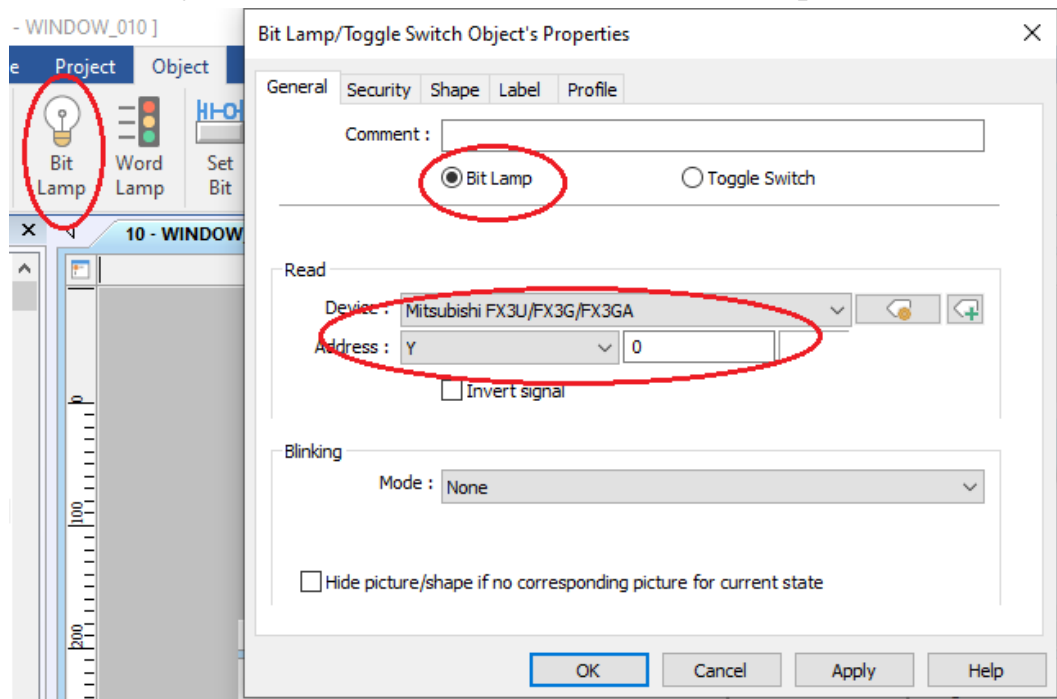
Cũng trong cửa sổ New Set Bit có thể đặt tên, thay đổi màu sắc cho nút nhấn ở các tab Shape và Label.

→ Nhấn OK để hoàn tất



- *Bước 7: tạo các đèn báo:* Trên thanh công cụ → Object → Bit Lamp → đặt địa chỉ bit điều khiển;

Đặt tên, thay đổi màu sắc cho nút nhấn ở các tab Shape và Label



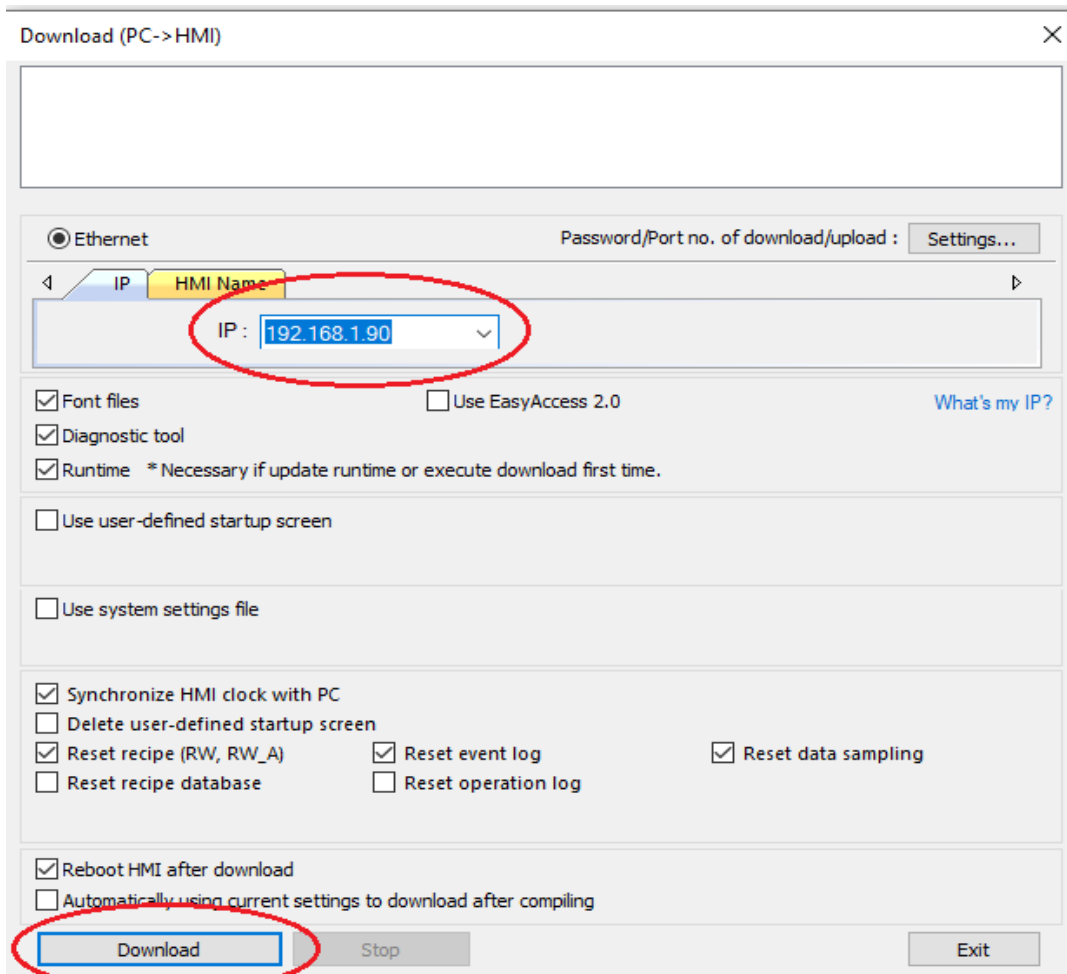
3.2.4. Tải giao diện vào HMI

- Kết nối HMI với máy tính bằng cáp Ethernet
- Chú ý đặt IP của máy tính cùng dải với HMI (hoặc để tự động)
- Tải chương trình:

Trên thanh công cụ → chọn Project → Download (PC to HMI)

→ Tại cửa sổ Download nhập IP của HMI

→ Nhấn Download



4. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không hiển thị đèn báo trạng thái trên HMI	Lỗi kết nối giữa HMI và PLC	- Kiểm tra lại cài đặt kết nối cho HMI trong phần mềm GT Design - Kiểm tra cáp kết nối
Không điều khiển được động cơ từ HMI	- Do chương trình lập trình thiếu các tiếp điểm M0, M1 - Gán sai địa chỉ cho các nút nhấn trên HMI	- Viết lại chương trình ở Bước 2 - Sửa lại địa chỉ các nút nhấn M0, M1.

IV. TIÊU CHÍ ĐÁNH GIÁ

- Kiểm tra kết nối thiết bị với đầu vào - ra []
- Trình tự khởi tạo và lập trình điều khiển máy bơm []
- Trình tự thiết kế giao diện HMI []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Phân loại và ứng dụng của HMI Mitsubishi
3. Sơ đồ đấu nối phân cứng
4. Trình tự thực hiện
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

1. Thiết kế giao diện và viết chương trình điều khiển động cơ xoay chiều ba pha với các chức năng: chạy, dừng, đảo chiều?

Bài thực hành số 7

LẬP TRÌNH TIMER**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Nhận biết được ký hiệu, cú pháp lệnh của các bộ Timer;
- Trình bày được nguyên lý hoạt động, chức năng của các bộ Timer.

Về kỹ năng:

- Sử dụng được thành thạo các lệnh Timer;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình sử dụng các câu lệnh lập trình Timer.

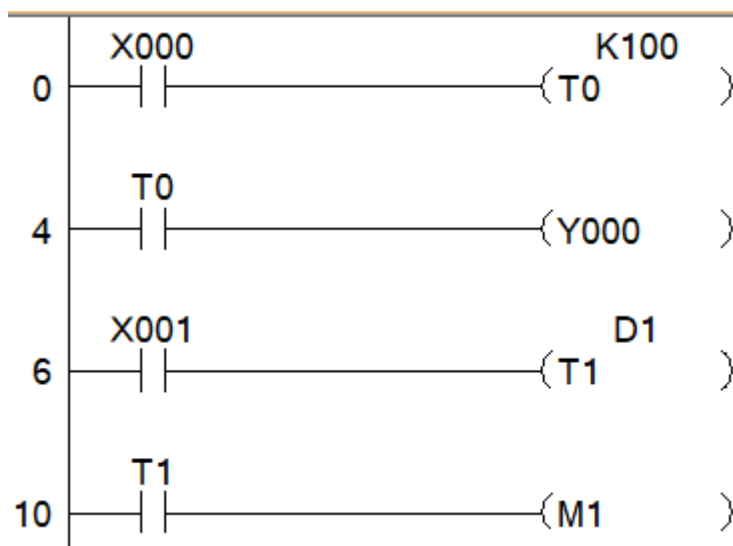
Về năng lực tự chủ và trách nhiệm:

- Có khả năng tự lập trình sử dụng các bộ timer và hướng dẫn người khác thực hành;
- Tự đánh giá được kết quả thực hành của bản thân.

II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu chung**

Timer là bộ tạo thời gian trễ giữa tín hiệu vào và tín hiệu ra, nên trong điều khiển thường được gọi là khâu trễ.

Kí hiệu bộ Timer:

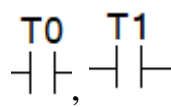


Hình 2.7-1. Ký hiệu các bộ Timer

T_i : Tên các Timer, $i = 0 \div 511$ (PLC FX 3U có 512 Timer)

K100: Giá trị đặt (hằng số)

D1: thanh ghi lưu giá trị đặt (có thể thay đổi được)

T0 T1
 : Tiếp điểm của bộ Timer tương ứng

Khi tín hiệu đầu vào được ON, các Timer cộng và đếm các xung clock theo chu kỳ 1ms, 10ms, 100ms,... (tùy loại Timer) trong PLC. Khi giá trị đếm đạt được giá trị cài đặt, tiếp điểm ngõ ra của timer bật ON.

2. Phân loại

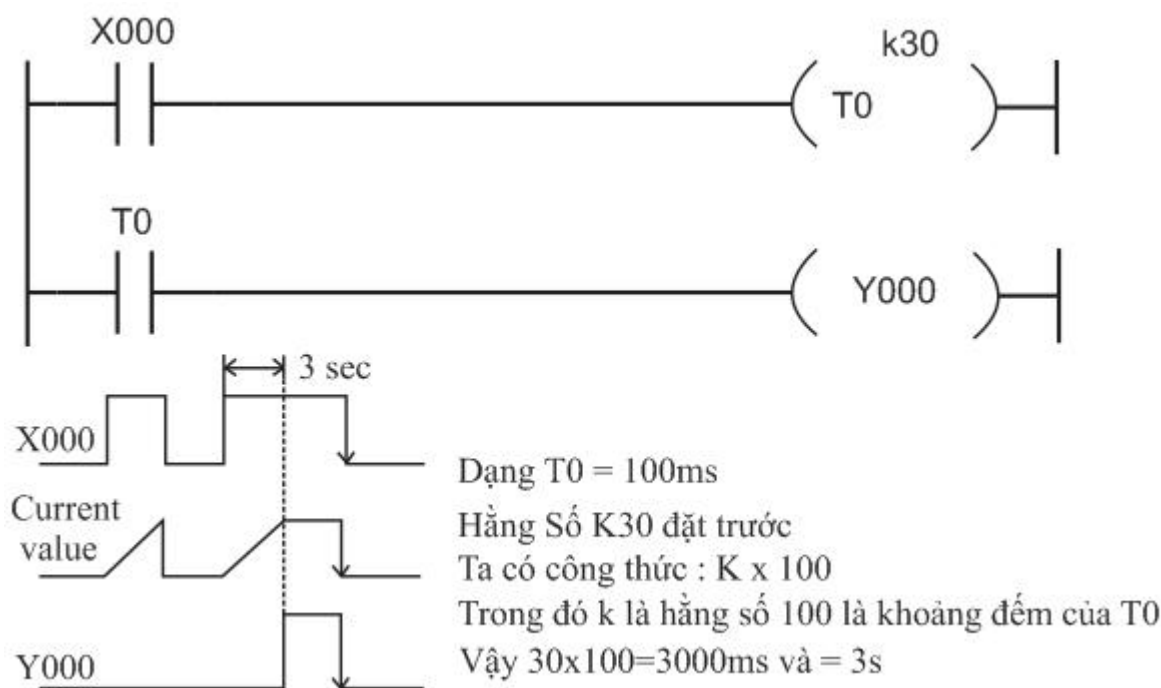
PLC Mitsubishi FX3U có 512 bộ Timer:

T0 ÷ T199 (200 điểm)	T200 ÷ T245 (46 điểm)	T246 ÷ T249 (4 điểm) Loại chột	T250 ÷ T255 (6 điểm) Loại chột	T256 ÷ T511 (256 điểm)
100ms	10ms	1ms	100ms	1ms

Timer loại chột sẽ lưu được giá trị đếm xung hiện hành khi ngắt tín hiệu duy trì cho timer hoặc khi mất nguồn điện. Khi có điện trở lại, bộ Timer sẽ tiếp tục đếm xung bắt đầu từ giá trị hiện hành đang lưu.

3. nguyên lý hoạt động

3.1. Timer loại thông thường



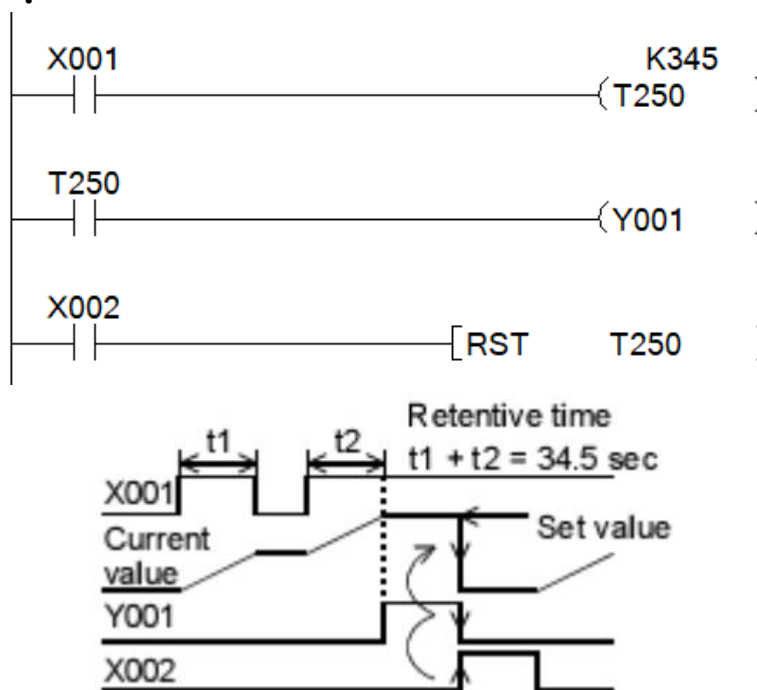
Hình 2.7-2. Nguyên lý hoạt động Timer loại thông thường

Khi ngõ vào điều khiển X0 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, đồng thời bật ON Y0.

Nói một cách khác, tiếp điểm ngõ Y0 bật ON sau 3s khi X0 ON. Khi ngõ vào điều khiển X0 OFF hoặc khi mất nguồn điện, timer được reset và các tiếp điểm ngõ ra trở về trạng thái đầu.

Cũng có thể Reset Timer về 0 bằng lệnh [RST T0]

3.2. Timer loại chốt



Hình 2.7-3. Nguyên lý hoạt động Timer loại chốt

Khi ngõ vào điều khiển X1 ON, Timer T250 sẽ bắt đầu đếm giá trị bắt đầu từ giá trị hiện hành của T250 với xung clock 1ms.

Khi giá trị đếm được bằng với giá trị cài đặt là K345, tiếp điểm ngõ ra của timer bật ON.

Khi ngõ vào điều khiển X0 OFF hoặc khi mất nguồn điện, timer sẽ dừng đếm xung và lưu giá trị đếm hiện hành. Khi có điện trở lại, X0 được ON thì Timer sẽ tiếp tục đếm từ giá trị hiện hành đã lưu.

Khi X2 ON, giá trị đếm của Timer bị reset về 0; tiếp điểm đầu ra của Timer T250 cũng OFF, đầu ra Y1 cũng OFF.

III. NỘI DUNG THỰC HÀNH

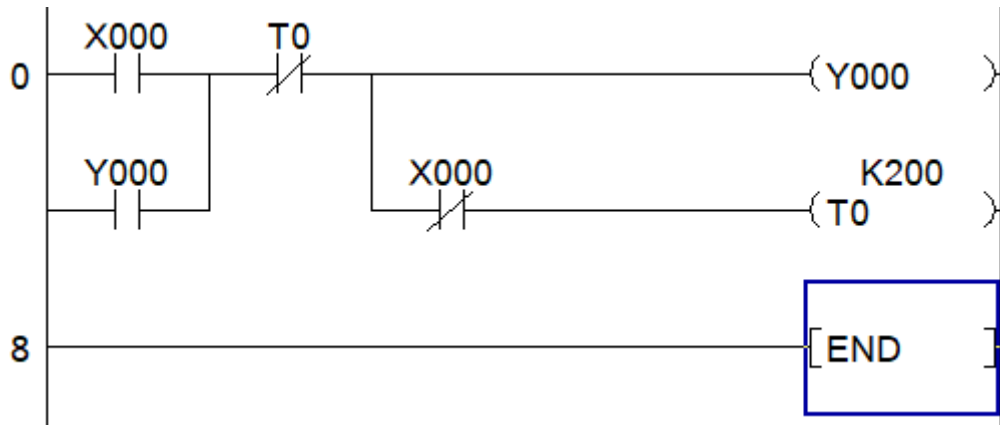
1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

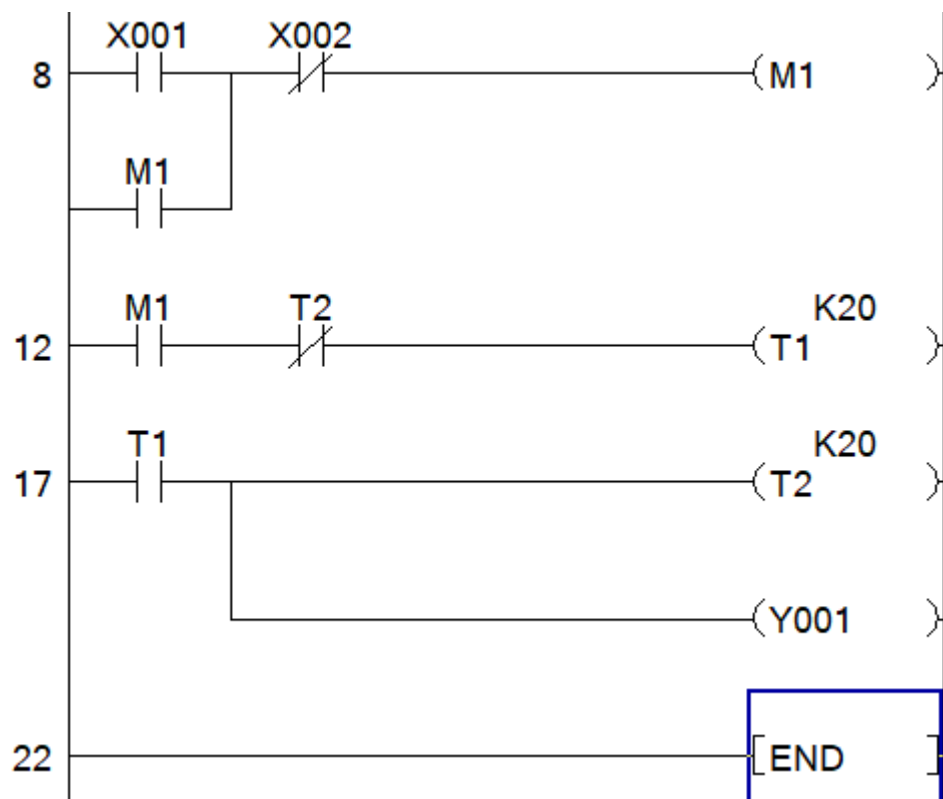
STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
2	Dây cắm	1 bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Các bước thực hiện

2.1. Lập chương trình tạo 1 Off-delay timer



2.2. Lập chương trình phát xung



2.3. Thực hành với các timer có chốt

Lập lại yêu cầu ở phần 2.1, 2.2 và sử dụng các timer có chốt

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Thời gian trễ không đúng	- Chọn sai bộ timer - Sai giá trị đặt	- Chọn lại timer - Tính toán lại giá trị đặt

IV. TIÊU CHÍ ĐÁNH GIÁ

- Chọn đúng loại timer []
- Tính toán được giá trị đặt []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Các loại timer và nguyên lý hoạt động
3. Lập trình thực hành các lệnh timer
4. Trình tự vận hành và kết quả từng bước

VI. CÂU HỎI KIỂM TRA

1. trình bày nguyên lý hoạt động các loại timer?
2. Lập chương trình điều khiển tuần tự 3 băng tải vận chuyển hàng?

Bài thực hành số 8

LẬP TRÌNH COUNTER

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Nhận biết được ký hiệu, cú pháp lệnh của các bộ Counter
- Trình bày được nguyên lý hoạt động, chức năng của các bộ Counter.

Về kỹ năng:

- Sử dụng được thành thạo các lệnh Counter;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình sử dụng các câu lệnh lập trình Counter.

Về năng lực tự chủ và trách nhiệm:

- Có khả năng tự lập trình sử dụng các bộ Counter và hướng dẫn người khác thực hành;
- Tự đánh giá được kết quả thực hành của bản thân.

II. TÓM TẮT LÝ THUYẾT

1. Giới thiệu chung

Counter là bộ đếm được sử dụng để đếm các cạnh (sườn) xung của tín hiệu đầu vào.

Ký hiệu bộ counter:



Hình 2.8-1. Ký hiệu các bộ counter

C_i : Tên các counter, $i = 0 \div 234$ (PLC FX 3U có 235 bộ counter)

C_i : lưu giá trị đếm hiện hành, được sử dụng như các thanh ghi trong các lệnh di chuyển dữ liệu hoặc lệnh toán học.

K100: Giá trị đặt (hằng số)

D1: thanh ghi lưu giá trị đặt (có thể thay đổi được)

C_0 C_1

$\neg|$, $|$: Tiếp điểm của bộ counter tương ứng

2. Phân loại

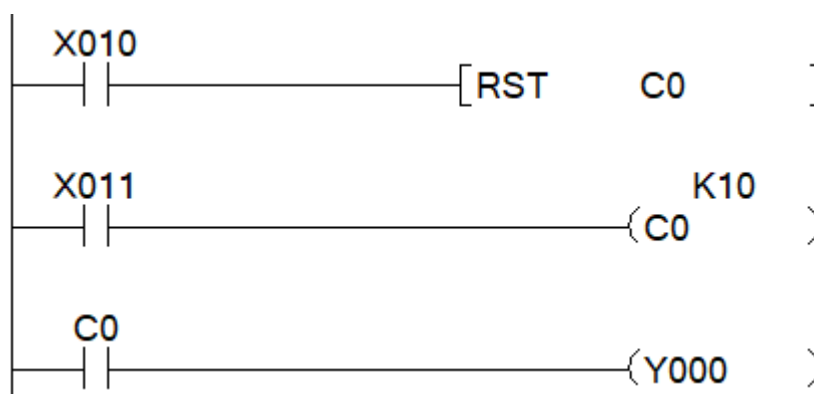
Bộ đếm tiến 16 bit đếm từ 0 đến 32767		Bộ đếm hai chiều 32 bit đếm từ -2,147,483,648 +2,147,483,647	
Loại thường	Loại chốt	Loại thường	Loại chốt
C0 ÷ C99	C100 ÷ C199	C200 ÷ C219	C220 ÷ C234
100 bộ	100 bộ	20 bộ	15 bộ

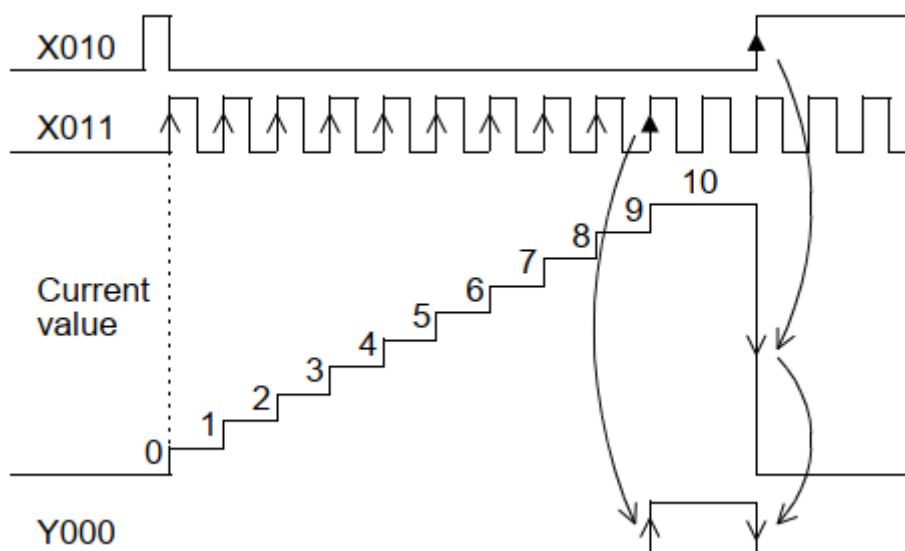
Bộ đếm loại chốt có thể lưu được giá trị đếm hiện hành khi mất nguồn điện cung cấp cho PLC

Mục	Bộ đếm 16-bit	Bộ đếm 32-bit
Hướng đếm (chiều đếm)	Đếm lên	Đếm lên và đếm xuống, có thể được chuyển hướng đếm nhờ các Role phụ trợ (M)
Giá trị cài đặt	1 ÷ 32767	-2,147,483,648 ÷ +2,147,483,647
Xác định giá trị cài đặt	Hàng số (K) hay một thanh ghi dữ liệu	Hàng số (K) hay một cặp thanh ghi dữ liệu
Thay đổi giá trị hiện hành	Không thay đổi khi đã đặt giá trị đặt	Có thể thay đổi ngay cả sau đạt giá trị đặt (bộ đếm vòng)
Tiếp điểm ngõ ra	ON sau khi đạt giá trị đặt	ON khi giá trị hiện hành bằng giá trị đặt ở chiều đếm lên; OFF khi giá trị hiện hành bằng giá trị đặt ở chiều đếm xuống
Thanh ghi giá trị hiện hành	16 bit	32 bit

3. Nguyên lý hoạt động

3.1. Counter đếm lên 16 bit





Hình 2.8-2. Nguyên lý hoạt động counter đếm lên 16 bit

Mỗi khi có sườn lên của đầu vào X011, giá trị hiện hành sẽ tăng lên 1 đơn vị (nếu giá trị hiện hành còn nhỏ hơn giá trị đặt K10). Khi đã đạt giá trị đặt K10 thì giá trị hiện hành không thay đổi cho dù có xung đầu vào X011.

Khi giá trị đếm đạt giá trị đặt thì tiếp điểm C0 của bộ đếm ON, Y0 cũng ON. Khi X010 ON thì lập tức giá trị đếm bị reset về 0, tiếp điểm C0 OFF.

Giá trị đặt cũng có thể được cài đặt thông qua 1 thanh ghi dữ liệu (D_i).

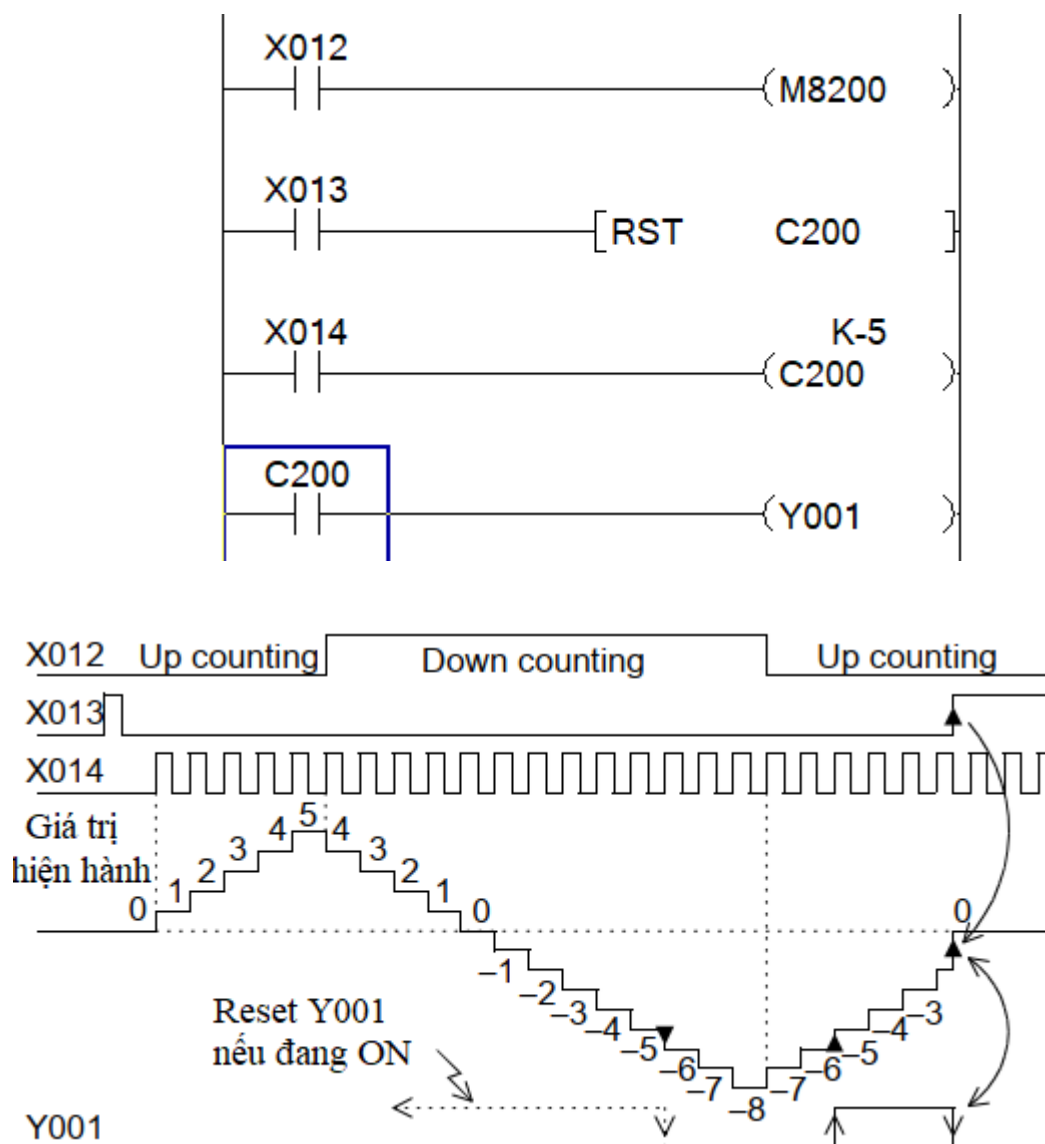
Counter loại chốt sẽ lưu được giá trị đếm hiện hành khi mất điện, khi có điện trở lại, counter sẽ thực hiện đếm tiếp từ giá trị đã lưu.

3.2. Counter đếm lên/xuống 32 bit

Chiều đếm lên (xuống) của counter C_i được quyết định bởi giá trị OFF (ON) của rơ le phụ trợ tương ứng M8_i, với i = 200 ÷ 234

Counter No.	Counting direction switching relay	Counter No.	Counting direction switching relay	Counter No.	Counting direction switching relay	Counter No.	Counting direction switching relay
C200	M8200	C209	M8209	C218	M8218	C227	M8227
C201	M8201	C210	M8210	C219	M8219	C228	M8228
C202	M8202	C211	M8211	C220	M8220	C229	M8229
C203	M8203	C212	M8212	C221	M8221	C230	M8230
C204	M8204	C213	M8213	C222	M8222	C231	M8231
C205	M8205	C214	M8214	C223	M8223	C232	M8232
C206	M8206	C215	M8215	C224	M8224	C233	M8233
C207	M8207	C216	M8216	C225	M8225	C234	M8234
C208	M8208	C217	M8217	C226	M8226		

Bảng 2-8.1 Rơ le phụ trợ chuyển hướng đếm counter



Hình 2.8-3. Nguyên lý hoạt động counter đếm lên, xuống 32 bit

M8012: chiều đếm, OFF - đếm lên; ON - đếm xuống.

Giá trị đặt: -5 (có thể đặt số âm hoặc số dương); giá trị đặt cũng có thể đặt gián tiếp qua thanh ghi.

Mỗi khi có sườn lên của X014 thì giá trị hiện hành sẽ tăng (nếu M8012 OFF) hoặc giảm (nếu M8012 ON) 1 đơn vị.

Khi giá trị hiện hành bằng giá trị đặt là -5 thì tiếp điểm đầu ra bộ đếm sẽ được set ON nếu đếm lên hoặc reset OFF nếu đếm xuống

Khi có tín hiệu reset X013, giá trị hiện hành của bộ đếm bị reset về 0, tiếp điểm đầu ra của bộ đếm bị reset OFF.

VII. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	PLC Mitsubishi dòng FX3u	1 bộ
2	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
3	Dây điện (1.0mm)	5m
4	Bộ đồ nghề dụng cụ	1 bộ

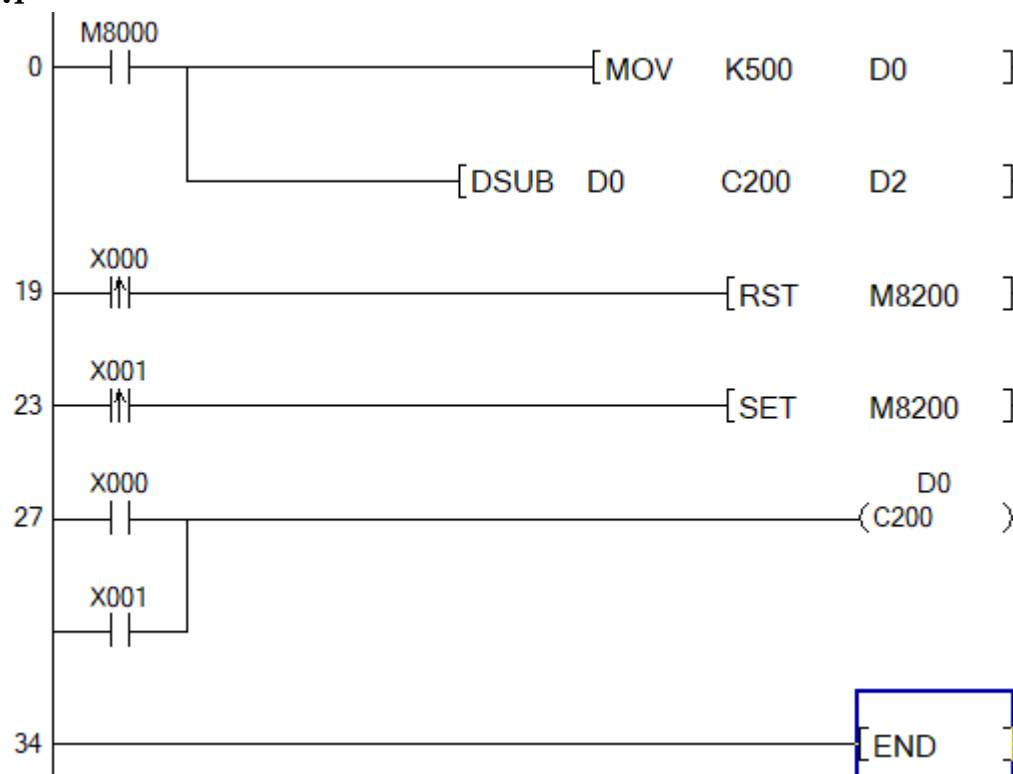
2. Các bước thực hiện

2.1. Yêu cầu: Lập trình giám sát số lượng xe trong bãi đỗ xe và thông báo số lượng vị trí đỗ còn trống?

Phân tích:

X001	Nhận tín hiệu từ cảm biến giám sát xe vào
X002	Nhận tín hiệu từ cảm biến giám sát xe ra
C200	Bộ đếm lên – xuống; Khi có xe vào thì đếm tăng, xe ra thì đếm giảm.
D0	Tổng số vị trí xe của bãi
D2	Số lượng vị trí đỗ xe còn trống

2.2. Lập trình



3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Counter 32 bit không đếm được lên hoặc xuống	Chọn sai cặp counter và rơ le phụ trợ điều khiển hướng đếm (M_i)	- Chọn lại cho đúng theo <i>Bảng 2-8.1</i>

IV. TIÊU CHÍ ĐÁNH GIÁ

- Chọn đúng loại counter []
- Lập trình đếm các sự kiện []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Các loại counter và nguyên lý hoạt động
3. Lập trình thực hành các lệnh counter
4. Trình tự vận hành và kết quả từng bước

VI. CÂU HỎI KIỂM TRA

1. Trình bày nguyên lý hoạt động các loại counter?
2. Lập chương trình đếm số lượng chai nước sản xuất được và đưa đến băng tải để đóng thùng theo số lượng 6 chai/thùng?

Bài thực hành số 9

LẬP TRÌNH ĐẾM SẢN PHẨM VÀ HIỂN THỊ LED 7 THANH**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

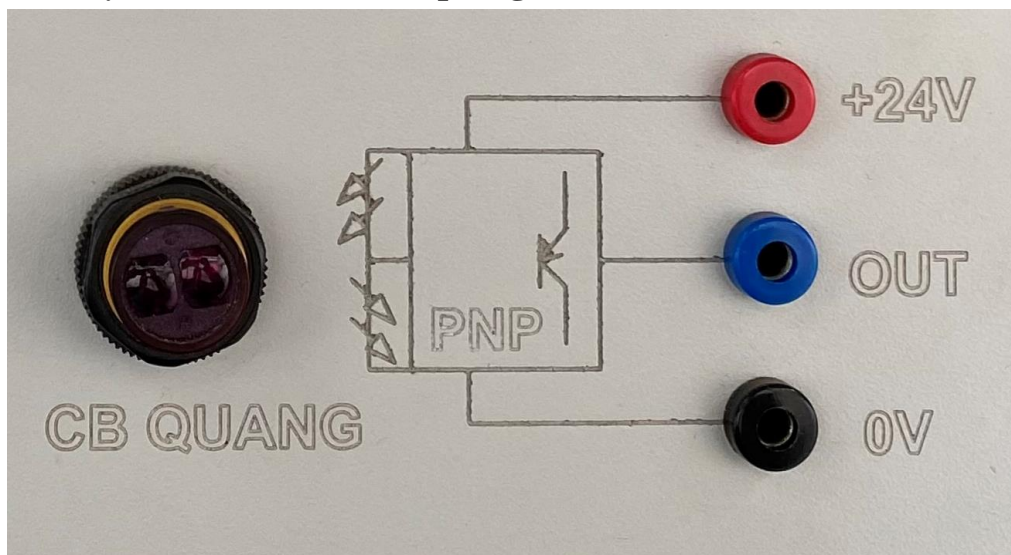
Phân loại và trình bày được nguyên lý hoạt động của các loại cảm biến hồng ngoại và Led 7 thanh.

Về kỹ năng:

- Đo, kiểm tra được các loại cảm biến hồng ngoại và Led 7 thanh;
- Vẽ được sơ đồ và đấu nối được các linh kiện, thiết bị theo sơ đồ;
- Lập trình đếm sản phẩm và hiển thị trên Led 7 thanh
- Khắc phục được một số sai hỏng thường gặp.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình đấu nối, lập trình;
- Có khả năng tự đấu nối, lập trình hoặc làm việc theo nhóm;
- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu module cảm biến quang**

Hình 2.9-1. Mô đun cảm biến quang

Trên mô hình đang sử dụng loại cảm biến quang PNP



Hình 2.9-2. Cảm biến quang

- Cảm biến loại này có 3 đầu dây:

Nâu: + nguồn 5 – 25VDC

Xanh: GND nguồn

Đen: Output (Dây tín hiệu ra)

- Nguyên lý hoạt động: Cảm biến có 2 đầu thu và phát tia hồng ngoại. Khi cấp điện, đầu phát sẽ liên tục phát ra tia hồng ngoại, nếu gặp vật cản thì tia hồng ngoại này sẽ bị phản xạ trở lại và tác động lên đầu thu.

- Cách đo kiểm tra cảm biến:

Cấp nguồn cho cảm biến (Nâu: + 24VDC, Xanh: GND);

Dùng đồng hồ VAO để thang đo 50VDC, que đỏ đặt vào dây đen (Output) của cảm biến, que đen đặt vào dây xanh (GND).

Khi chưa có vật cản, kim chỉ 0V, khi có vật cản ở đầu cảm biến, nếu kim chỉ 24V thì cảm biến còn hoạt động tốt, nếu kim vẫn chỉ 0V thì cảm biến đã hỏng.

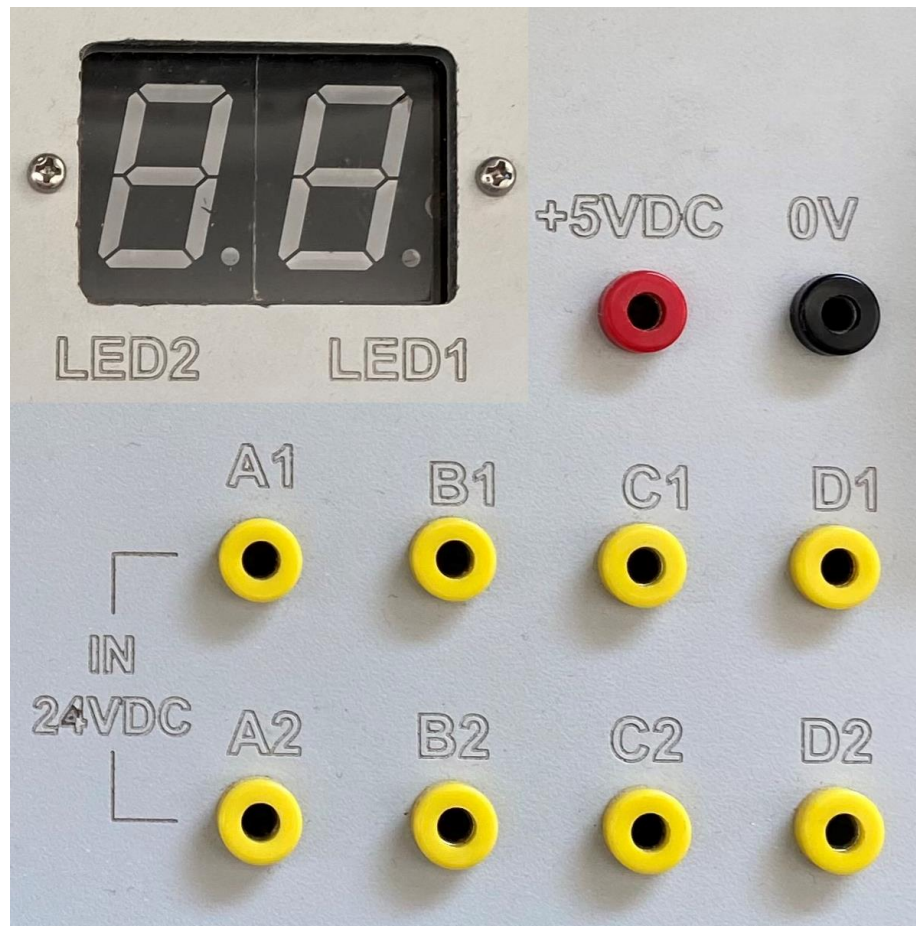
2. Giới thiệu module Led 7 thanh

2.1. Module Led 7 thanh: bao gồm 2 Led, đã được ghép nối với IC giải mã

- Để Led hoạt động được cần cấp nguồn 5VDC vào 2 cọc cắm (đỏ đen)

- Các Chân điều khiển của Led 1 lần lượt là A1, B1, C1, D1; Led 2 lần lượt là A2, B2, C2, D2 (trọng số lần lượt từ cao xuống thấp). Các chân này được điều khiển bởi đầu ra của PLC thông qua Rơ le trung gian.

- Có thể kiểm tra hoạt động của Led bằng cách cấp điện 24VDC vào các chân A_i , B_i , C_i , D_i và quan sát hiện thị của Led.



Hình 2.9-3. Module Led 7 thanh

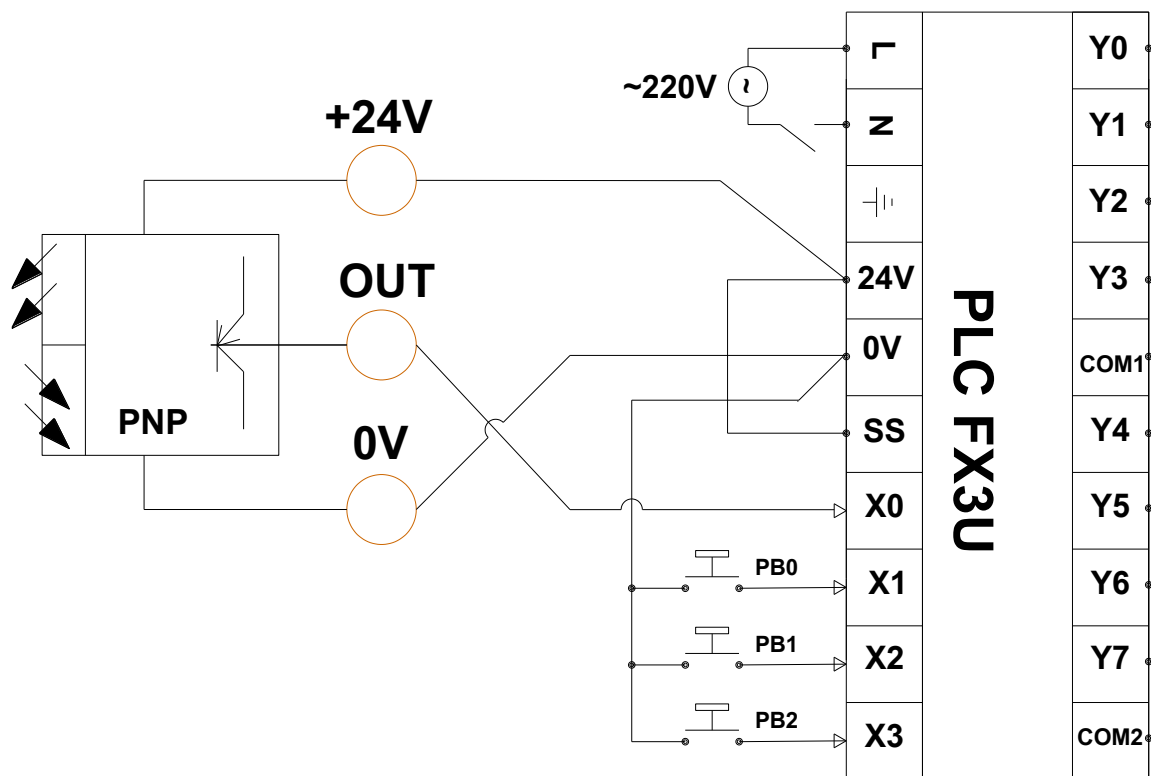
2.2. Bảng giá trị điều khiển Led 7 thanh

- Bảng giá trị điều khiển cho Led 1

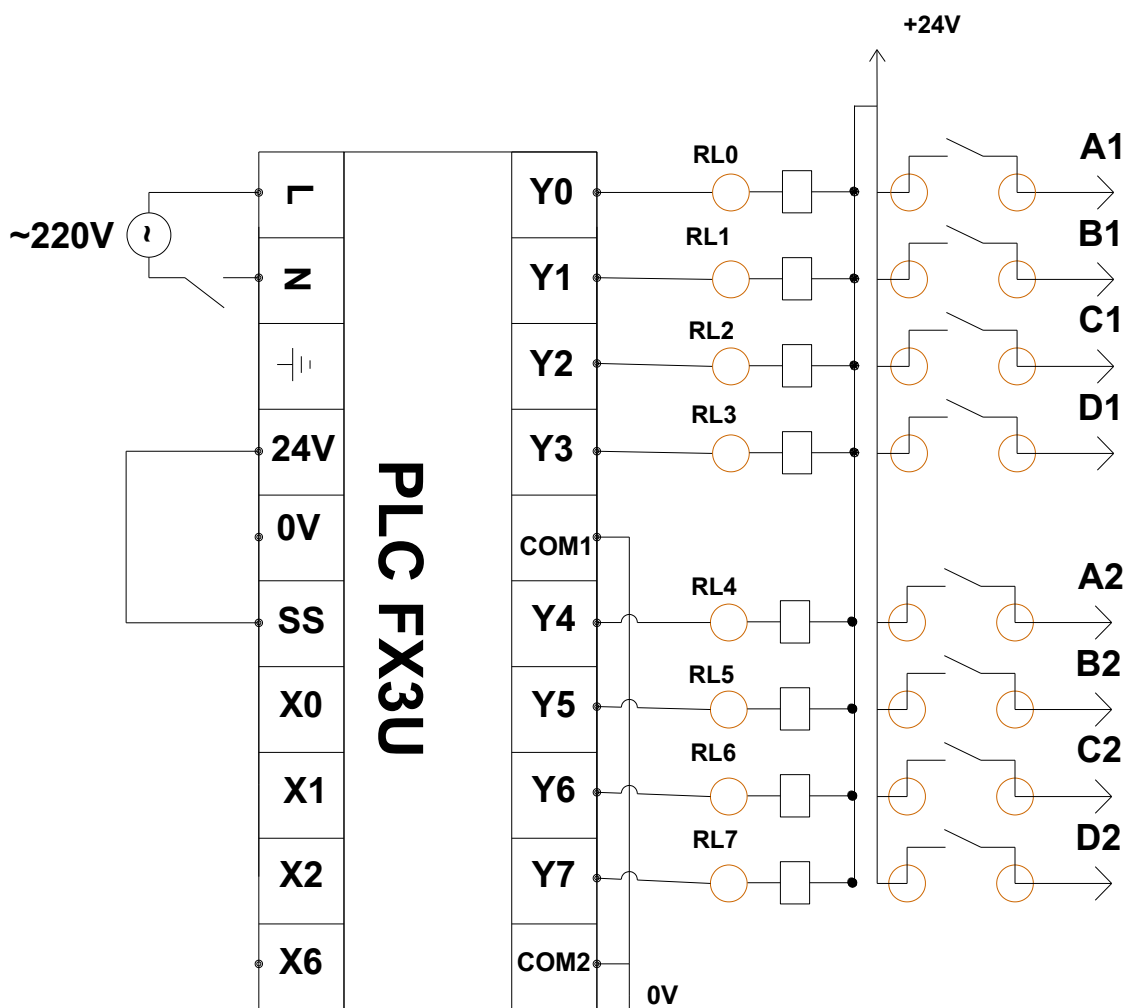
Số hiển thị	A1	B1	C1	D1	Ghi chú
0	0	0	0	0	0 – 0VDC 1 – 24VDC
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	

- Tương tự với giá trị điều khiển cho Led 2

3. Sơ đồ kết nối



Hình 2.9-4. Sơ đồ kết nối cảm biến với đầu vào PLC



Hình 2.9-5. Sơ đồ kết nối đầu ra PLC điều khiển Led 7 thanh

III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

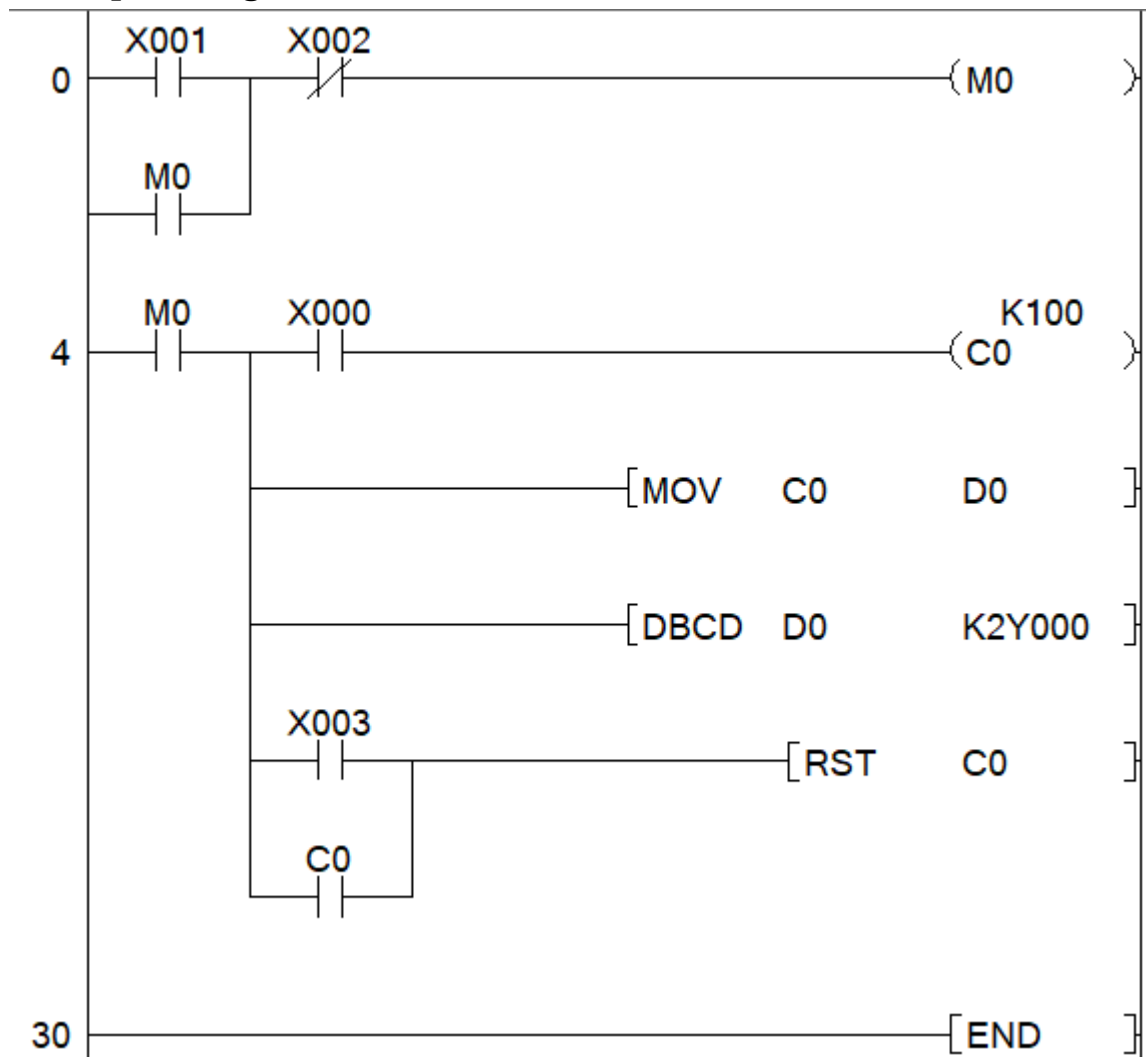
2. Các bước thực hiện

2.1. Đấu nối thiết bị theo sơ đồ

- Sử dụng dây cắm đầu nối các module với nhau theo sơ đồ hình 2.9-4, và hình 2.9-5;

- Kiểm tra các kết nối trước khi đóng điện, sử dụng đồng hồ VAO đo nguội kiểm tra nếu cần thiết.

2.2. Lập chương trình điều khiển



2.3. Vận hành chạy thử

- Tải chương trình vào PLC.
- Nhấn Pb0 để bắt đầu khởi động hệ thống đếm, dùng tay hoặc vật thể đưa qua đầu cảm biến và quan sát 2 Led 7 thanh.
- Nhấn Pb1 để dừng hệ thống đếm.
- Nhấn Pb2 để xóa bộ đếm.

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Bộ đếm không hoạt động	- Hỏng cảm biến - Đầu nối đầu vào xung, chân S/S	- Đo, kiểm tra cảm biến - Đầu nối theo đúng sơ đồ hình 2.9-4
LED 7 thanh không hiển thị hoặc hiển thị không đúng	- Nguồn cấp 5VDC - Thứ tự các chân điều khiển không đúng	- Kiểm tra lại nguồn cấp - Đầu nối theo đúng sơ đồ hình 2.9-5

IV. TIÊU CHÍ ĐÁNH GIÁ

- Thao tác đo, kiểm tra cảm biến, Led 7 thanh []
- Đầu nối thiết bị theo sơ đồ []
- Chương trình lập trình đếm sản phẩm []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Nguyên lý hoạt động cảm biến quang và cách điều khiển Led 7 thanh
3. Sơ đồ kết nối mạch đếm sản phẩm và hiển thị led 7 thanh
4. Chương trình điều khiển
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

1. Nguyên lý hoạt động và sơ đồ đầu nối của cảm biến hồng ngoại NPN, PNP?
2. Lập bảng trạng thái điều khiển Led 7 thanh sử dụng IC giải mã?

Bài thực hành số 10

LẬP TRÌNH BỘ ĐẾM TỐC ĐỘ CAO ĐỌC XUNG ENCODER TƯƠNG ĐỐI

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Phân loại và trình bày được nguyên lý hoạt động của các loại bộ đếm tốc độ cao;
- Trình bày được cấu tạo, nguyên lý hoạt động của encoder tương đối.

Về kỹ năng:

- Đấu nối được phần cứng để đọc xung encoder tương đối;
- Lập trình đọc xung encoder tương đối theo các chế độ khác nhau;
- Khắc phục được một số sai hỏng thường gặp.

Về năng lực tự chủ và trách nhiệm:

- Có khả năng tự lập trình sử dụng các bộ đếm tốc độ cao và hướng dẫn người khác thực hành;
- Tự đánh giá được kết quả thực hành của bản thân.

II. TÓM TẮT LÝ THUYẾT

1. Bộ đếm tốc độ cao HSC

1.1. Giới thiệu chung

HSC là bộ đếm tốc độ cao 32 bit, đếm một pha hoặc hai pha hai chiều lên/xuống, dùng để đọc các xung tín hiệu có tần số cao.

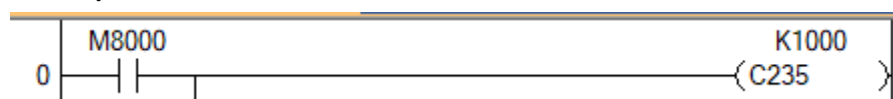
HSC thường được dùng để đọc tín hiệu của encoder trong các ứng dụng điều khiển, giám sát tốc độ, vị trí.

HSC được chia thành 2 loại:

Đếm cứng: dùng để đếm các xung tốc độ cao từ thiết bị bên ngoài đưa về PLC.


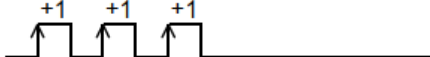





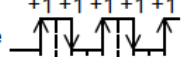
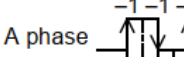
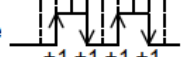
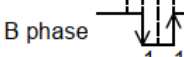
Đếm mềm: dùng trong quá trình xử lý ngắt CPU.

Ký hiệu các bộ HSC:



Bộ đếm HSC được sử dụng giống như bộ đếm thông thường:

1.2. Các loại HSC và dạng tín hiệu vào

Loại		Dạng tín hiệu vào	Chiều đếm
1 - pha 1- ngõ vào đếm C235 ÷ C8245		LÊN/XUỐNG 	Chiều đếm được quyết định bằng cách bật ON hay OFF M8235 ÷ M8245 ON: đếm xuống OFF: đếm lên
1 - pha 2 - ngõ vào đếm C246 ÷ C250		LÊN  XUỐNG 	Bộ đếm tự động thực hiện việc đếm lên hay xuống tùy thuộc vào công nhận xung đầu vào Có thể kiểm tra M8246 ÷ M8250 để biết chiều đếm: ON: đếm xuống OFF: đếm lên
2 - pha 2 - ngõ vào đếm C251 ÷ C255	Đếm 1 cạnh	A phase  B phase  Đếm lên  Đếm xuống 	Bộ đếm tự động thực hiện việc đếm lên hay xuống tùy thuộc vào việc thay đổi trạng thái ngõ vào A/B Chiều đếm có thể kiểm tra bởi M8251 ÷ M8255 ON: đếm xuống OFF: đếm lên
	Đếm 4 cạnh	A phase  B phase  Đếm lên  Đếm xuống 	

1.3. Các bit điều khiển của HSC

Loại	Bộ đếm	Bit đặc biệt	Đếm lên	Đếm xuống
1 - pha 1- ngõ vào đếm	C235	M8235	ON	OFF
	C236	M8236		
	C237	M8237		
	C238	M8238		
	C239	M8239		
	C240	M8240		
	C241	M8241		

	C242	M8242		
	C243	M8243		
	C244	M8244		
	C245	M8245		

Loại	Bộ đếm	Bit đặc biệt	ON	OFF
1 - pha 2- ngõ vào đếm	C246	M8246	Đếm lên	Đếm xuống
	C247	M8247		
	C248	M8248		
	C249	M8249		
	C250	M8250		
2 - pha 2- ngõ vào đếm	C251	M8251		
	C252	M8252		
	C253	M8253		
	C254	M8254		
	C255	M8255		

1.4. Quy định đầu vào cho mỗi bộ HSC

	Counter No.	Classifica- tion	Input terminal assignment							
			X000	X001	X002	X003	X004	X005	X006	X007
1-phase 1-count input	C235 ^{*1}	H/W ^{*2}	U/D							
	C236 ^{*1}	H/W ^{*2}		U/D						
	C237 ^{*1}	H/W ^{*2}			U/D					
	C238 ^{*1}	H/W ^{*2}				U/D				
	C239 ^{*1}	H/W ^{*2}					U/D			
	C240 ^{*1}	H/W ^{*2}						U/D		
	C241	S/W	U/D	R						
	C242	S/W			U/D	R				
	C243	S/W					U/D	R		
	C244	S/W	U/D	R						S
	C244(OP) ^{*3}	H/W ^{*2}								U/D
	C245	S/W			U/D	R				
C245(OP) ^{*3}	H/W ^{*2}									U/D
1-phase 2-count input	C246 ^{*1}	H/W ^{*2}	U	D						
	C247	S/W	U	D	R					
	C248	S/W				U	D	R		
	C248(OP) ^{*1*3}	H/W ^{*2}				U	D			
	C249	S/W	U	D	R					S
	C250	S/W				U	D	R		S
2-phase 2-count input ^{*4}	C251 ^{*1}	H/W ^{*2}	A	B						
	C252	S/W	A	B	R					
	C253 ^{*1}	H/W ^{*2}				A	B	R		
	C253(OP) ^{*3}	S/W				A	B			
	C254	S/W	A	B	R					S
	C255	S/W				A	B	R		S

2. Encoder tương đối

2.1. Giới thiệu

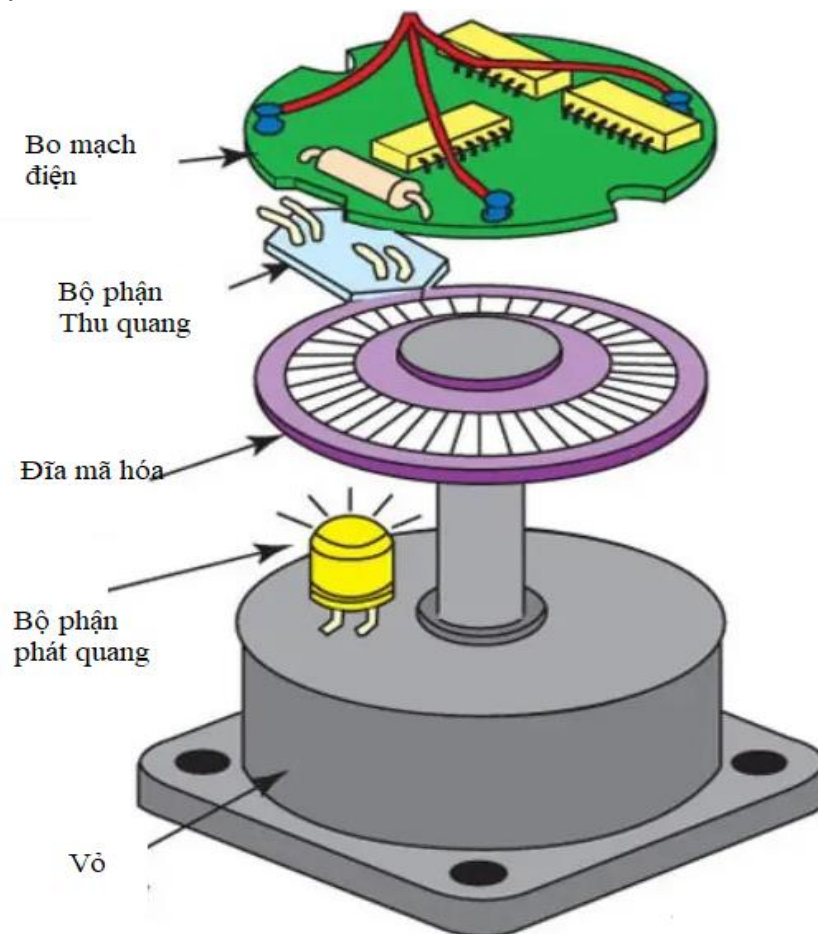
Encoder hay còn gọi là bộ mã hóa vòng quay, biến các chuyển động cơ thành các xung điện. Encoder là bộ phận rất quan trọng trong các máy CNC, các dây chuyền sản xuất, giúp giám sát được tốc độ, vị trí động cơ.

Encoder thường được lắp ở phía sau động cơ servo, gắn trên trục động cơ, gắn trên trục băng tải.

Ví dụ encoder Omron E6B2 – CWZ6C 100P/R, có thông số như sau:

	Resolution	600 P/R (xung/vòng)
	Brown	5 – 24VDC
	Blue	0V
	Black	Out A
	White	Out B
	Orange	Out Z
	Loại đầu ra: NPN Colecto hở	

2.2. Cấu tạo



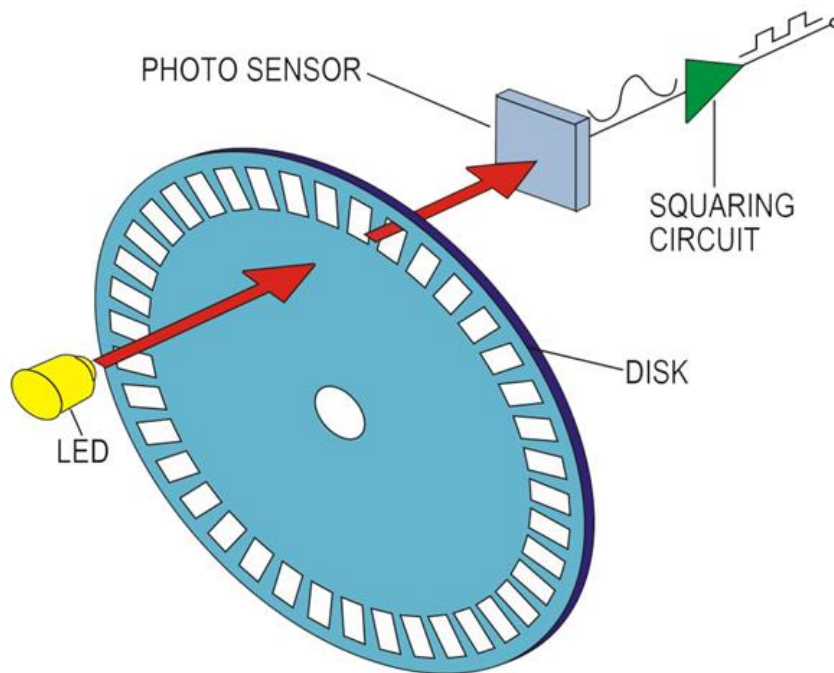
Hình 2.10-1. Cấu tạo encoder tương đối

Cấu tạo encoder tương đối gồm các bộ phận chính như sau:

- Đĩa mã hóa: được khoét lỗ, được gắn với trục quay
- Bộ phận phát quang: là 1 led dùng làm nguồn phát sáng
- Bộ phận thu quang: thường là các tranito thu quang, được đặt thẳng hàng với led phát
- Bo mạch điện: giúp xử lý, khuếch đại tín hiệu.

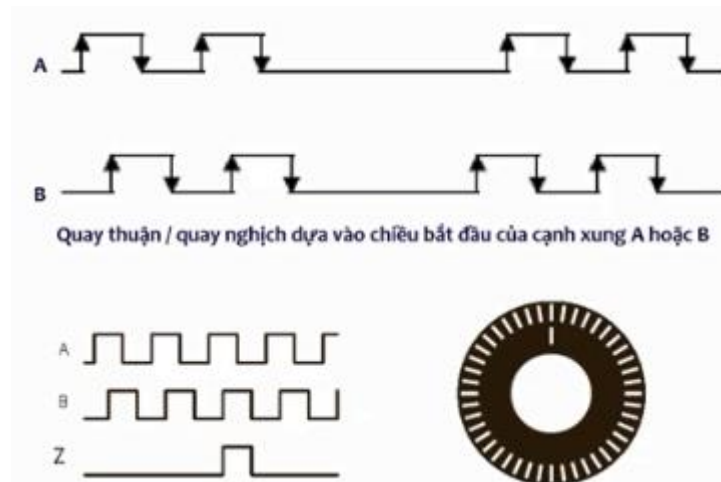
Số lượng xung trên 1 vòng quay của encoder phụ thuộc vào số lỗ trên đĩa quay. Mỗi đầu ra xung (A, B, Z) sẽ có 1 bộ thu và hệ thống lỗ khác nhau.

2.3. nguyên lý hoạt động



Hình 2.10-2. Nguyên lý hoạt động encoder tương đối

Khi đĩa mã hóa quay, các lỗ khoét trên đĩa sẽ quét qua đường ánh sáng chiếu từ bộ phận phát đến bộ phận thu, bo mạch điện sẽ biến đổi thành các xung điện ở đầu ra A, B, Z.



Hình 2.10-3. Các xung đầu ra

Đầu ra xung A, B có số xung/vòng giống nhau, xung có chu kỳ lệch nhau 90 độ, dùng để xác định chiều quay.

Đầu ra Z chỉ có 1 xung/vòng.

2.4. Cách đo kiểm tra encoder

- Cấp nguồn VDC (5V hoặc 24V tùy loại) cho encoder
- Cách 1: dùng đồng hồ VAO thang đo VDC, đặt que đỏ vào đầu ra (A, B hoặc Z), que đen nối 0V. Sau đó xoay nhẹ trục encoder. Nếu thấy kim đồng hồ lên xuống 24v, 0v là Encoder còn tốt. (Chú ý, đối với xung Z chỉ có 1 xung 1 vòng);
- Cách 2: Nối các chân đầu ra A, B, Z vào đầu vào X0, X1, X2 của PLC.
 - + PLC nối đầu vào kiểu Sink nếu encoder loại NPN, kiểu Source với encoder loại PNP.
 - + Sau đó xoay nhẹ trục encoder sẽ thấy đèn báo X0, X1, X2 nhấp nháy thì encoder còn tốt.

III. NỘI DUNG THỰC HÀNH

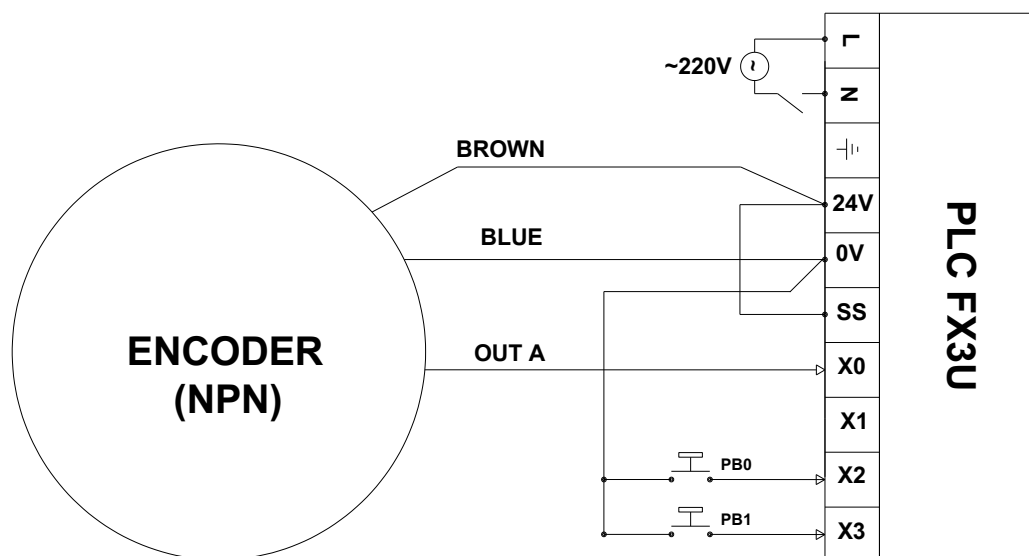
1. Chuẩn bị dụng cụ, thiết bị

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Encoder Omron E6B2 – CWZ6C	1 cái
4	Bộ đồ nghề dụng cụ	1 bộ

2. Các bước thực hiện

2.1. Thực hành với bộ đếm 1 pha 1 ngõ vào

2.1.1. Sơ đồ kết nối



Hình 2.10-4. Sơ đồ kết nối 1 pha 1 ngõ vào

2.1.2. Lập trình

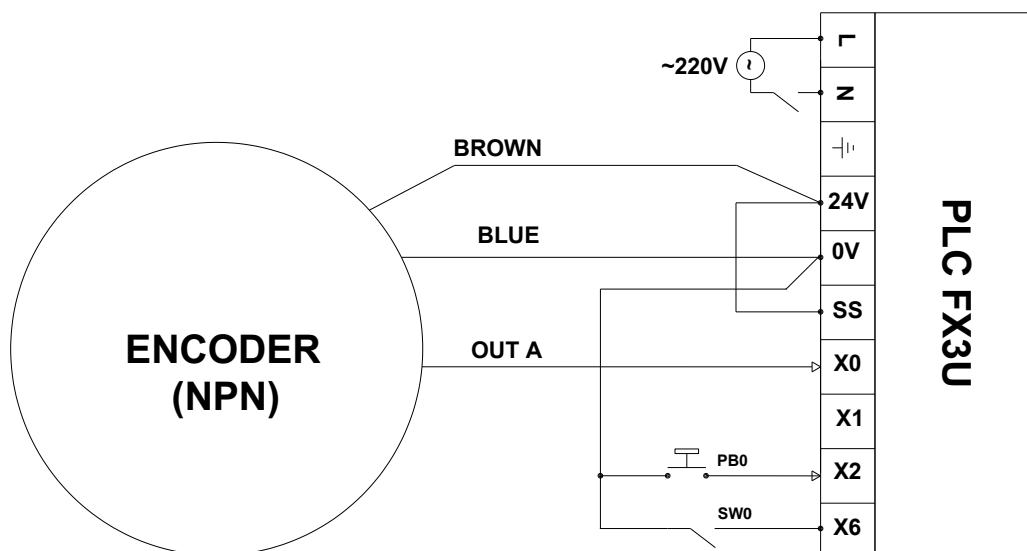
Chương trình lập trình:



- Nút ấn “Set” X₂ để xác định chiều đếm.
- Nút ấn reset X₃ để xóa bộ đếm.
- Quay encoder và quan sát giá trị hiện hành các bộ HSC.
- Chú ý: việc đếm tăng hay giảm không phụ thuộc vào chiều quay của Encoder

2.2. Thực hành với bộ đếm 1 pha 2 ngõ vào

2.2.1. Sơ đồ kết nối

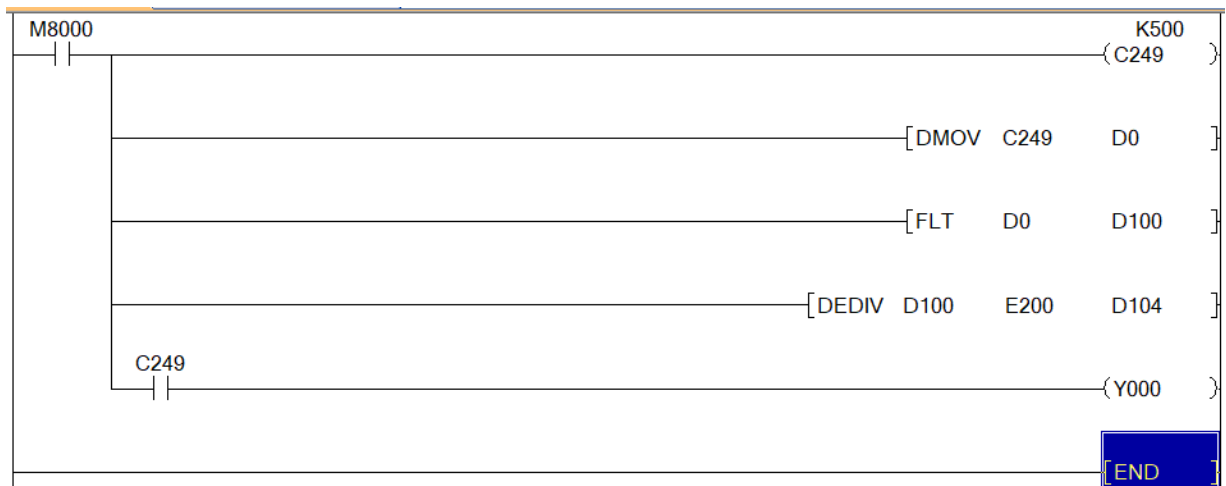


Hình 2.10-5. Sơ đồ kết nối 1 pha 2 ngõ vào

Ví dụ sử dụng bộ đếm C249:

- Đếm lên nối X₀, đếm xuống nối X₁, Reset X₂, Chân Set nối X₀₀₆
- Công tắc SW0 set ON bộ đếm hoạt động, set OFF bộ đếm dừng hoạt động.
- **Chú ý: không nối đồng thời xung A vào X₀ và xung B vào X₁.**

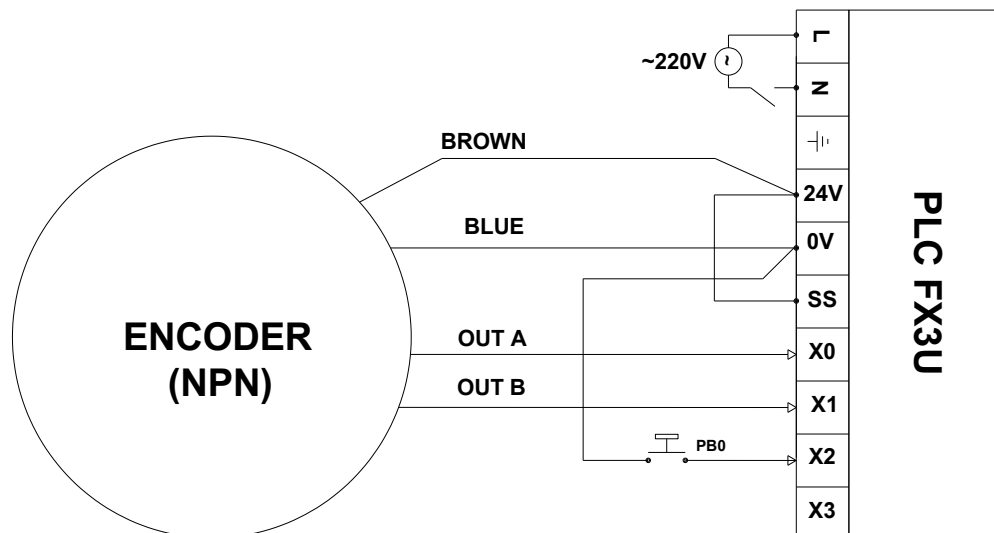
2.2.2. Chương trình lập trình



2.3. Thực hành với bộ đếm 2 pha 2 ngõ vào

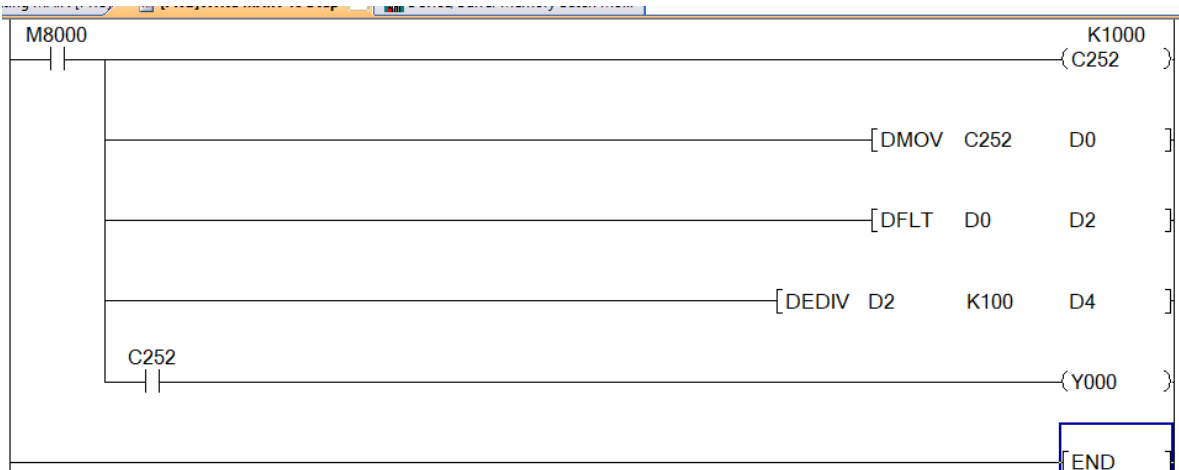
2.3.1. Sơ đồ kết nối

- Sử dụng bộ đếm C252: xung A nối X0, B nối X1, nút reset PB0 nối X2.
- Bộ đếm tự động đếm tăng hoặc giảm tùy thuộc vào chiều quay của encoder.



Hình 2.10-6. Sơ đồ kết nối 2 pha 2 ngõ vào

2.3.2. Chương trình lập trình



3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Bộ đếm không hoạt động	- Hỏng encoder - Đầu nối đầu vào xung, chân S/S, chân Reset, set	- Kiểm tra encoder - Đầu nối theo đúng sơ đồ, chú ý bảng Quy định đầu vào cho mỗi bộ HSC tại mục 1.4
Bộ đếm không chính xác	Đặt tiếp điểm đầu vào bộ đếm là các đầu vào X	Đặt tiếp điểm đầu vào bộ đếm là M8000 (always ON)

IV. TIÊU CHÍ ĐÁNH GIÁ

- Thao tác đo, kiểm tra encoder []
- Sơ đồ đầu nối bộ đếm theo các chế độ []
- Chương trình lập trình sử dụng các bộ đếm []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi sử dụng bộ đếm []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Nguyên lý hoạt động encoder
3. Sơ đồ kết nối sử dụng bộ đếm tốc độ cao ở các chế độ
4. Chương trình điều khiển
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

Viết chương trình đo tốc độ quay động cơ có trục gắn với trục của encoder E6B2 – CWZ6C 100P/R?

Bài thực hành số 11

ĐỌC CẢM BIẾN MÀU KEYENCE CZ-V1**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

Trình bày được cấu tạo, nguyên lý hoạt động và ứng dụng của cảm biến màu KEYENCE CZ-V1.

Về kỹ năng:

- Dạy được cho cảm biến nhận biết các màu sắc;
- Đấu nối được phần cứng kết nối cảm biến màu Keyence CZ-V1 với PLC theo sơ đồ nguyên lý;
- Lập được chương trình cho PLC đọc dữ liệu từ cảm biến và hiển thị màu sắc tương ứng;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình thực hành kết nối cảm biến màu.

Về năng lực tự chủ và trách nhiệm:

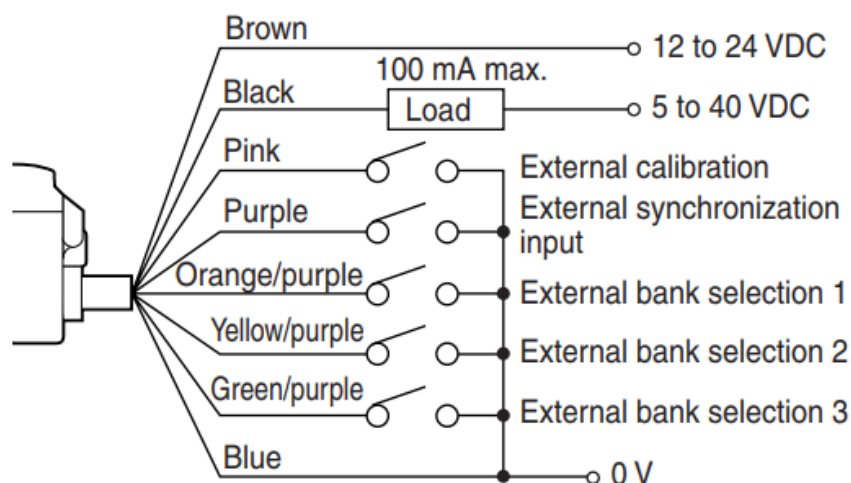
- Có khả năng tự lập trình đọc cảm biến màu và hướng dẫn người khác thực hành;
- Tự đánh giá được kết quả thực hành của bản thân.

II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu cảm biến màu keyence CZ-V1**

- Cảm biến màu keyence CZ –V1 có thể phát hiện được 8 màu sắc khác nhau bằng cách dạy cho cảm biến.



Hình 2.11-1. Cảm biến CZ-V1



Hình 2.11-2. Sơ đồ nối dây

2. Dạy màu cho cảm biến

- Nối dây cho cảm biến như hình 2.13-2.
- Các chân Orange (cam), Yellow (vàng), Green (xanh) được điều khiển qua các mức Logic 1 (để hở), 0 (nối 0 V) nhờ các công tắc hoặc nối trực tiếp.
- Nối đầu rò màu sắc (dùng chung với cảm biến quang), gạt lẫy → ấn 2 dây vào lỗ cảm biến → khóa lẫy.
- Lần lượt đặt các tổ hợp 3 dây lựa chọn (select) SL₁, SL₂, SL₃, ứng với mỗi lần (tổ hợp), đặt đầu rò lại gần màu cần xác định, sau đó nhấn nút “Set” để dạy cho cảm biến nhớ màu đó .

Wire color / Channel	Cam SL1	Vàng SL2	Xanh lá SL3
A	1	1	1
B	0	1	1
C	1	0	1
D	0	0	1
E	1	1	0
F	0	1	0
G	1	0	0
H	0	0	0

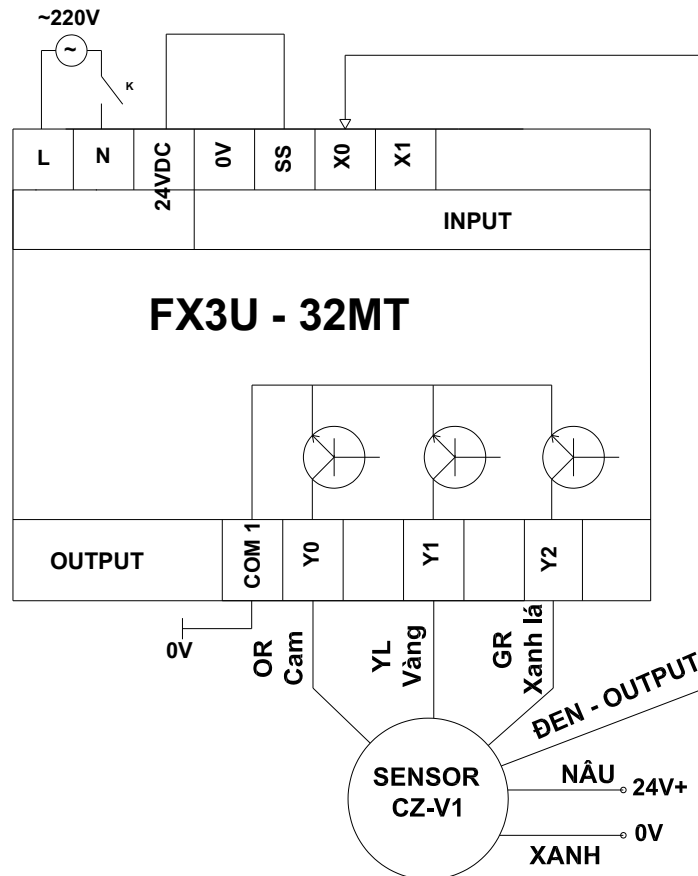
- Sensor sẽ nhớ được 8 màu tương ứng với 8 tổ hợp như bảng trên.
- Ứng với mỗi tổ hợp 3 đầu vào cam/vàng/xanh lá thì trên màn hình của cảm biến sẽ hiển thị kênh màu tương ứng
- Ví dụ: + Hở mạch cả 3 dây sẽ hiển thị kênh AC...

+ Kín mạch cả 3 dây sẽ hiển thị kênh HC...

- Sau khi đã dạy màu cho cảm biến xong, nếu đưa đầu rò đến gần 1 màu bất kỳ đã được dạy, và lần lượt thay đổi tổ hợp 3 đầu rò cam/ vàng/ xanh lá. Nếu tổ hợp đúng với màu đó thì sẽ có tín hiệu ra ở dây đen (đầu ra là dạng NPN collector hở)

III. NỘI DUNG THỰC HÀNH

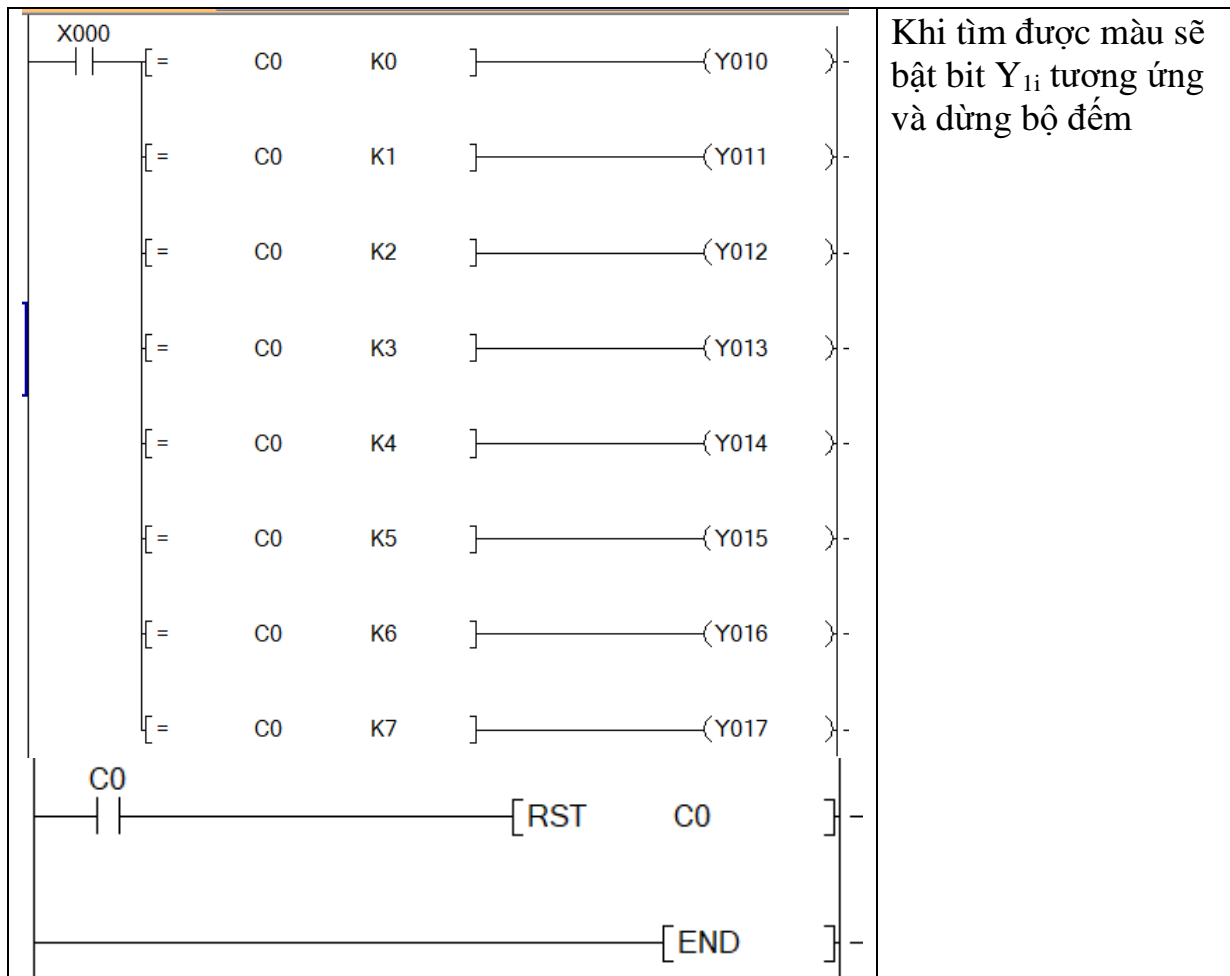
1. Sơ đồ nối dây



Hình 2.11-3. Sơ đồ nối dây cảm biến CZ-V1 với PLC

2. Viết chương trình

<pre> M8002 -----[MOV K0 C0] -----[MOV K0 K2Y000] M8012 =[K2Y010 K0]----- (C0) K8 M8000 -----[MOV C0 K2Y000] </pre>	<p>Khởi tạo, xóa C0, K2Y0</p>
	<p>Xung 100ms đếm C0</p>
	<p>Hoặc [Mov C0 K1 Y000] để nạp giá trị cho 3 bit Y0, Y1, Y2 thay đổi trạng thái đầu vào cảm biến</p>



IV. TIÊU CHÍ ĐÁNH GIÁ

- Thao tác đấu nối, dạy cho cảm biến màu []
- Sơ đồ đấu nối PLC với cảm biến màu []
- Chương trình lập trình phân biệt màu sắc []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi sử dụng bộ đếm []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Thông số kỹ thuật và nguyên lý hoạt động cảm biến màu
3. Sơ đồ kết nối PLC với cảm biến màu
4. Chương trình PLC phân biệt màu sắc
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

Viết chương trình phân loại sản phẩm theo màu sắc?

Bài thực hành số 12

XỬ LÝ TÍN HIỆU ANALOG SỬ DỤNG MODULE FX2N-5A**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Nhận biết được ký mã hiệu của module analog;
- Trình bày được cấu tạo, nguyên lý hoạt động và ứng dụng của module FX2N-5A.

Về kỹ năng:

- Sử dụng được thành thạo các lệnh cơ bản đúng chức năng;
- Đấu nối được phần cứng vào ra theo sơ đồ nguyên lý;
- Lập được chương trình đọc, ghi dữ liệu sử dụng PLC dòng FX và module FX2N-5A;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình sử dụng các câu lệnh lập trình.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình đấu nối, lập trình xử lý tín hiệu analog;
- Có khả năng tự đấu nối, lập trình hoặc làm việc theo nhóm;
- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

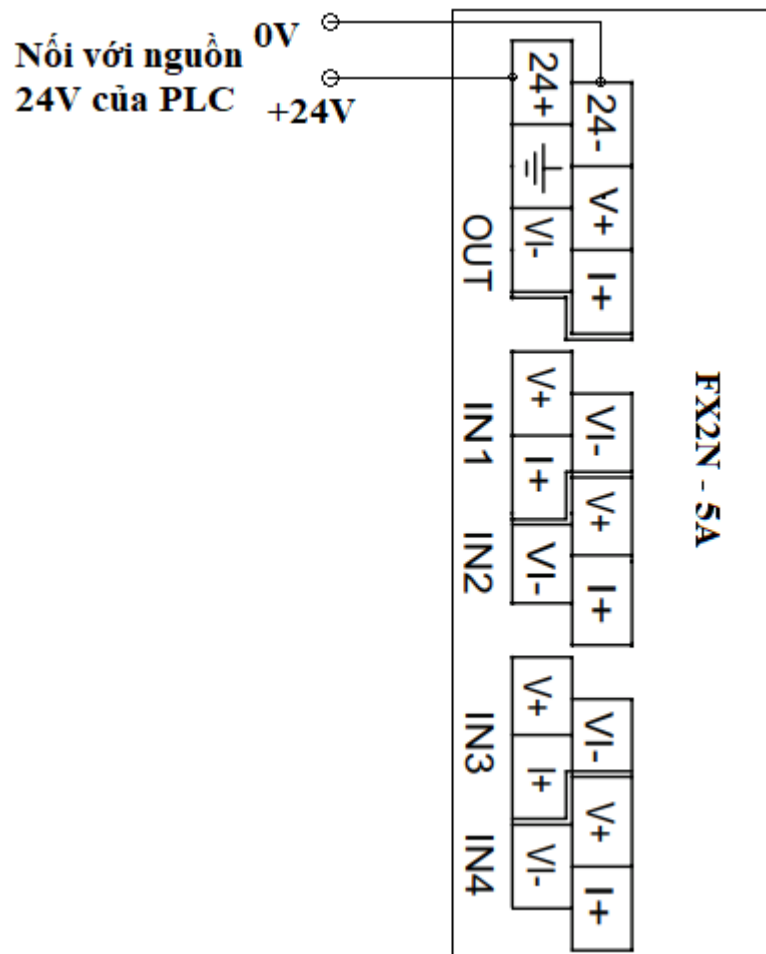
II. TÓM TẮT LÝ THUYẾT**1. Thông số kỹ thuật của module FX2N-5A**

- Module FX2N-5A có thể kết nối với FX0N/1N/2N/2NC, FX3G/3GC, FX3u và FX3uc của PLC Mitsubishi
- Tùy theo Seri PLC mà có thể kết nối tối đa 2, 4 hoặc 8 modul Fx2N-5A.
Ví dụ:
 - + FX2N, 3G, 3GE, FX3U, 3UC kết nối được tối đa với 8 Fx2N-5A
 - + Riêng FX3UC - 32MT-LT chỉ kết nối được tối đa với 7 module Fx2N-5A.

Thời gian lấy mẫu ;à 1ms ($T_m=1ms$)

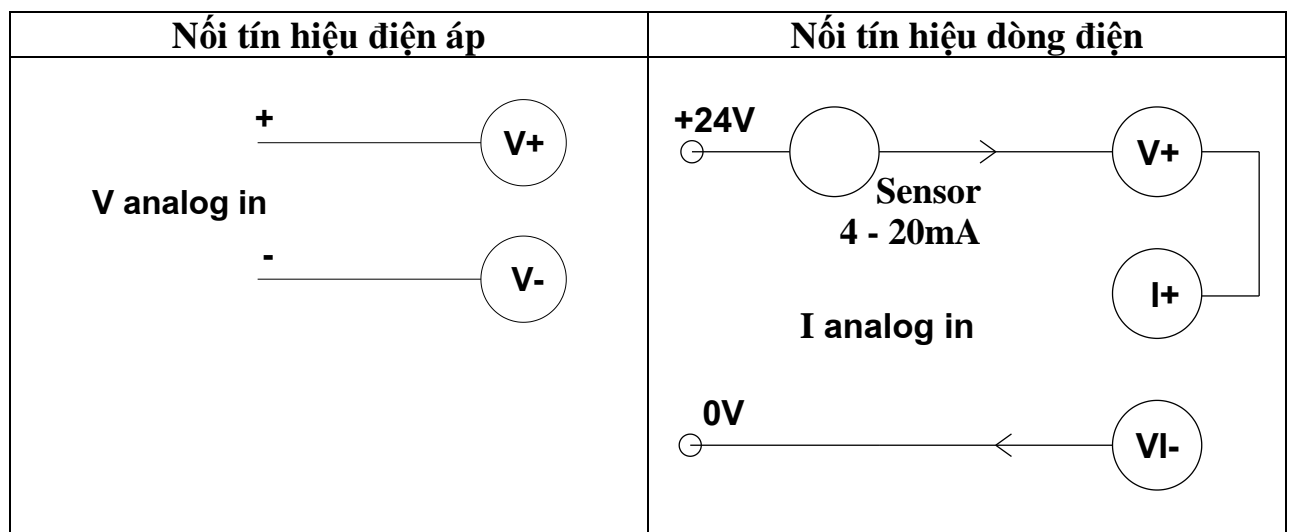
2. Sơ đồ đấu nối

- Gồm 4AD và 1DA, sơ đồ như sau:



Hình 2.12-1. Sơ đồ chân module FX2N – 5A

- Tín hiệu điện áp $\pm 10\text{v}$, tín hiệu dòng điện $\pm 20\text{mA}$, tùy thuộc vào chế độ cài đặt;
- Địa chỉ hóa các modul bằng số thứ tự từ 0 đến 7 (0 là module gần PLC nhất, bỏ qua modul I/O)
- Phương pháp nối dây tín hiệu:



Hình 2.12-2. Sơ đồ đấu dây các loại nguồn tín hiệu vào FX2N – 5^a

3. Chức năng của các thanh ghi trong bộ nhớ đệm BFM

3.1. Đặc điểm các thanh ghi

- Các thanh ghi này (16 bit) lưu trữ các tham số thiết lập chế độ làm việc khi xử lý tín hiệu và lưu trữ các giá trị chuyển đổi tín hiệu

- **Chú ý:** Không thực hiện các phép toán cộng, trừ,... trong các thanh ghi này. Nếu cần thực hiện tính toán, phải lấy dữ liệu ở các thanh ghi chuyển sang thanh ghi ở vùng nhớ dữ liệu (thanh ghi D₀, D₁, ... D_n) để tính toán sau đó nạp trở lại vào chúng.

- Các thanh ghi trong bộ nhớ đệm được đánh số lần lượt theo thứ tự số nguyên từ 0, 1, 2...n (dùng chữ K để chỉ dẫn)

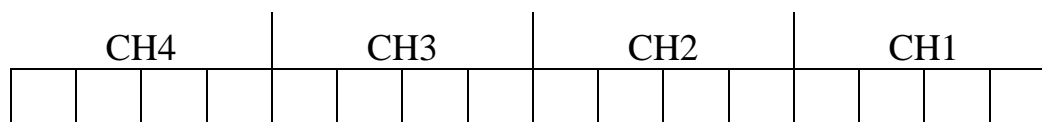
Ví dụ: K₀ thanh ghi số 0

K₁ thanh ghi số 1

3.2. Ý nghĩa của các thanh ghi như sau

a. Thanh ghi #0 (K0):

- Xác định chế độ làm việc của tín hiệu vào Analog cho các kênh từ CH₁ - CH₄



- Giá trị khởi tạo là H0000 Mỗi giá trị của từng kênh có ý nghĩa sau:

+ **0:** cho tín hiệu vào -10v ÷ +10v, hiển thị -32000 ÷ 32000

+ **1:** cho tín hiệu dòng từ 4 ÷ 20mA, hiển thị 0 ÷ 32000

(Nếu tín hiệu vào < 2mA thì đưa ra một cảnh báo ở thanh ghi K28)

+ **2:** cho tín hiệu dòng vào -20 ÷ +20mA, hiển thị -32000 ÷ 32000

+ **3:** cho tín hiệu điện áp vào -100 ÷ +100mV, hiển thị -3200 ÷ 32000

+ **4:** cho tín hiệu điện áp vào -100 ÷ +100mV, hiển thị -2000 ÷ +2000

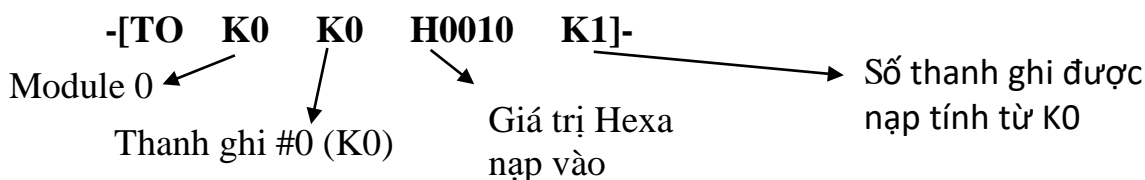
+ **5:** cho tín hiệu điện áp vào -10v ÷ +10V, hiển thị -10000 ÷ +10.000

.....

+ **C, D, E:** không dùng (không hợp lệ), modun sẽ phục hồi giá trị hợp lệ của lần trước đó.

+ **F:** Khóa kênh, không cho làm việc. Khi không sử dụng nên nạp = F để giảm tần số quét.

- Muốn nạp giá trị vào thanh ghi này ta không dùng lệnh Mov thông thường mà phải dùng lệnh TO:



Sau lệnh trên thanh ghi số $\neq 0$ (của Modun 0 – Modun gần PLC nhất) được nạp giá trị H0010;

Tức là, ba kênh CH₁, CH₃, CH₄ làm việc ở chế độ 0 (thực hiện áp $\neq 10\text{v}$), kênh CH₂ làm việc ở chế độ 1 ($4 \div 20\text{mA}$)

Có thể thay giá trị H0010 = K2 ($0010_{(\text{HEX})} = 2_{(\text{DEC})}$)

- Giá trị này (H0010) được nạp vào duy nhất thanh ghi $\neq 0$ (do tham số cuối cùng là K1 = 1), (Nếu là K2 thì sẽ được nạp cả vào thanh ghi $\neq 1$).

b. Thanh ghi #1 (K1):

- Xác định chế độ làm việc của đầu ra Analog nên chỉ tồn tại kênh CH₁

- Ý nghĩa của các giá trị nạp vào kênh CH₁ như sau

+ **0**: đầu ra xuất ra $-10\text{v} \div +10\text{v}$, ứng với số nhập vào thanh ghi $\neq 14$ từ: $-32000 \div +32000$

+ **1**: đầu ra xuất ra $-10\text{v} \div +10\text{v}$, ứng với số nhập vào: $-2000 \div 2000$

+ **2**: đầu ra xuất ra $4 \div 20\text{mA}$, ứng với số nhập vào: $0 \div 32000$

+ **3**: đầu ra xuất ra $4 \div 20\text{mA}$, ứng với số nhập vào: $0 \div 1000$

c. Thanh ghi #2, 3, 4, 5:

- Là các thanh ghi lưu số lần lấy mẫu để đưa ra giá trị trung bình, giá trị khởi tạo của các thanh ghi này là 8 (8 lần)

#2 cho CH₁

#3 cho CH₂

#4 cho CH₃

#5 cho CH₄

- Nếu giá trị thanh ghi #2 bằng 8, CPU sẽ lấy 8 mẫu, sau đó chia ra giá trị trung bình: $8 \times 1\text{ms} = 8\text{ms}$ sẽ hiển thị giá trị trung bình

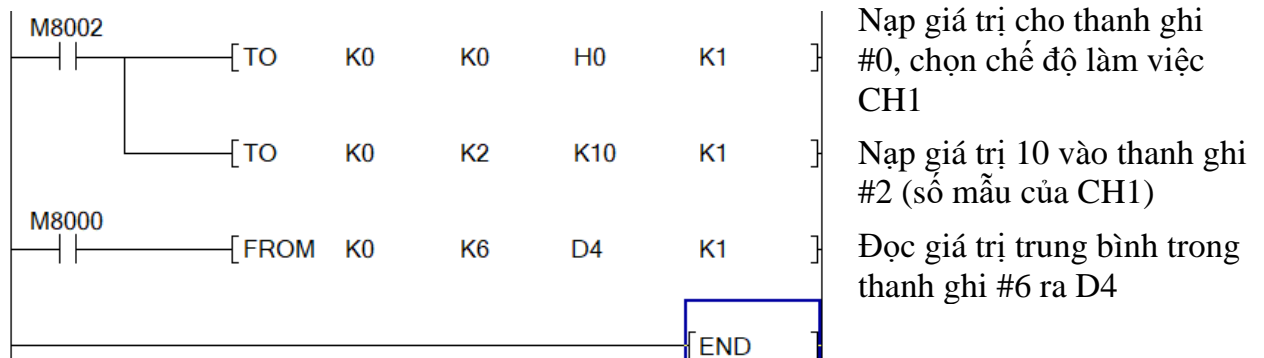
- Ta có thể thay đổi giá trị khởi tạo này từ $1 \div 256$

Ví dụ: nạp giá trị 200 vào thanh ghi #2 thì: cứ mỗi 200ms sẽ xuất ra giá trị trung bình.

d. Các thanh ghi #6, 7, 8, 9:

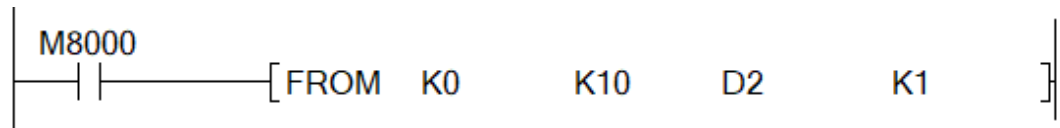
- Là các thanh ghi lần lượt lưu giá trị trung bình của các mẫu tại kênh CH₁, CH₂, CH₃, CH₄

Ví dụ: Muốn đọc các dữ liệu trung bình sau 10 lần lấy mẫu ra thanh ghi D₄ của kênh CH₁, modun FX2N-5A nằm gần CPU nhất (modun 0) ta viết lệnh như sau:



d. Các thanh ghi ≠ 10, 11, 12, 13

- Lưu các giá trị tức thời, lần lượt của CH₁, CH₂, CH₃, CH₄ (sau 1 chu kỳ lấy mẫu $T_m=1ms$ đưa ra kết quả 1 lần)
- Ví dụ: Đọc giá trị tức thời kênh CH₁ của modul 0 ra thanh ghi D₂



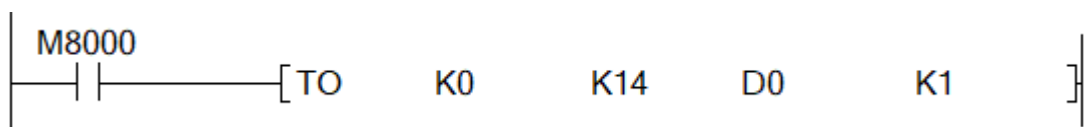
- Ví dụ: Nạp đồng thời số lần lấy mẫu K=10 vào các thanh ghi #2, #3, #4, #5 của modul 0:



Nạp giá trị 10 vào 4 thanh ghi liên tiếp kể từ thanh ghi #2 (K2)

e. Thanh ghi #14:

- Lưu giá trị số nguyên cần nhập vào để chuyển đổi sang tín hiệu Analog đầu ra trên kênh CH₁.
- Để nạp giá trị ta dùng lệnh T0:



Câu lệnh trên sẽ nạp giá trị lưu trong D0 vào thanh ghi #14 của module 0.

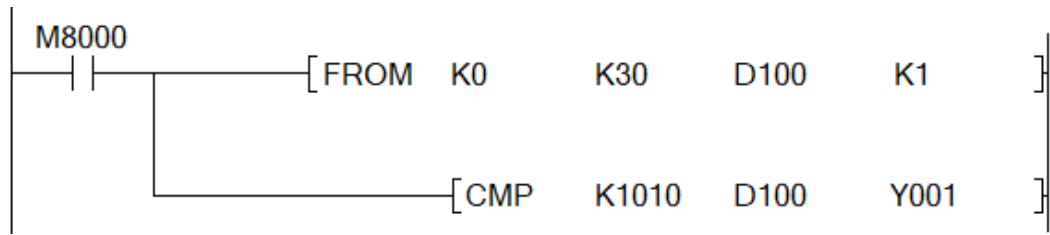
Giá trị D₀ phải phù hợp với chế độ làm việc đã xác định bởi thanh ghi #1 cho kênh analog ra CH₁ (xem mục **b.**).

f. Thanh ghi #30:

- Lưu mã ID của modul Analog.

Ví dụ: Modul FX2N-5A có ID “1010”

Câu lệnh kiểm tra mã của modul “0” xem có phải là loại FX2N-5A hay không, nếu đúng thì Y001 = 0; sai thì Y001 = 1;



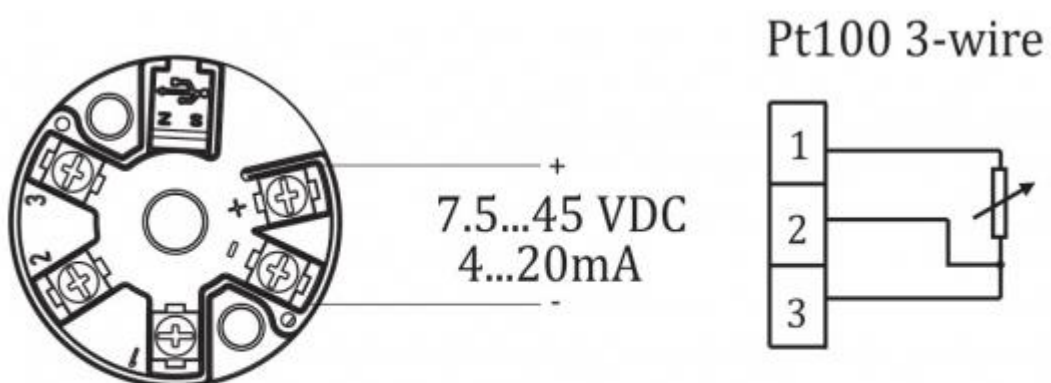
Đọc mã mô đun ra thanh ghi D100 và so sánh với K1010, nếu khác thì Y001 ON, bằng thì Y001 OFF

III. NỘI DUNG THỰC HÀNH

1. Bài tập ví dụ 1: Đấu nối cảm biến PT100 + Tranducer MST110 với PLC FX3u + Module FX2N – 5a để đo và hiển thị nhiệt độ nước.

1.1. Giới thiệu cảm biến PT100 và Tranducer MST110

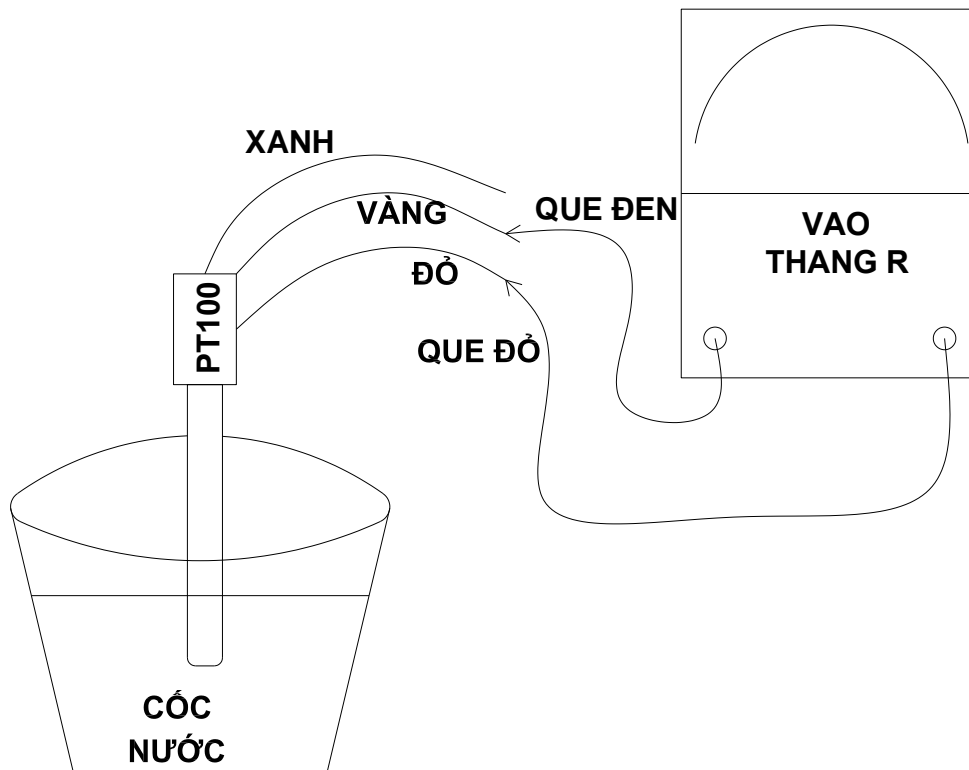
Bộ Tranducer MST110 giúp chuyển đổi tín hiệu từ cảm biến Pt100 thành tín hiệu ra dòng chuẩn từ: 4 – 20mA



MST110 with 4...20mA output

Hình 2.12-3. Sơ đồ đấu nối Pt100 với Tranducer

1.2. Kiểm tra cảm biến nhiệt độ Pt100



Hình 2.12-4. Khảo sát đo điện trở của Pt100

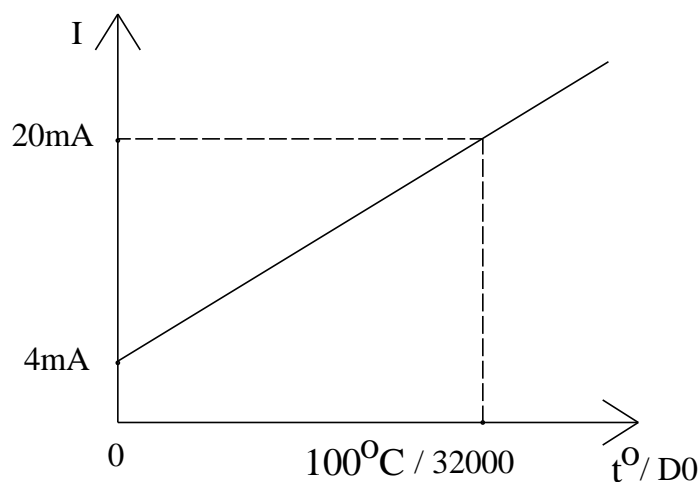
Kết quả:

- Ở nhiệt độ môi trường (20°C): $R = 110\Omega$
- Nước sôi 100°C: $R = 140\Omega$
- Điện trở đo giữa 2 dây vàng – xanh là 1Ω

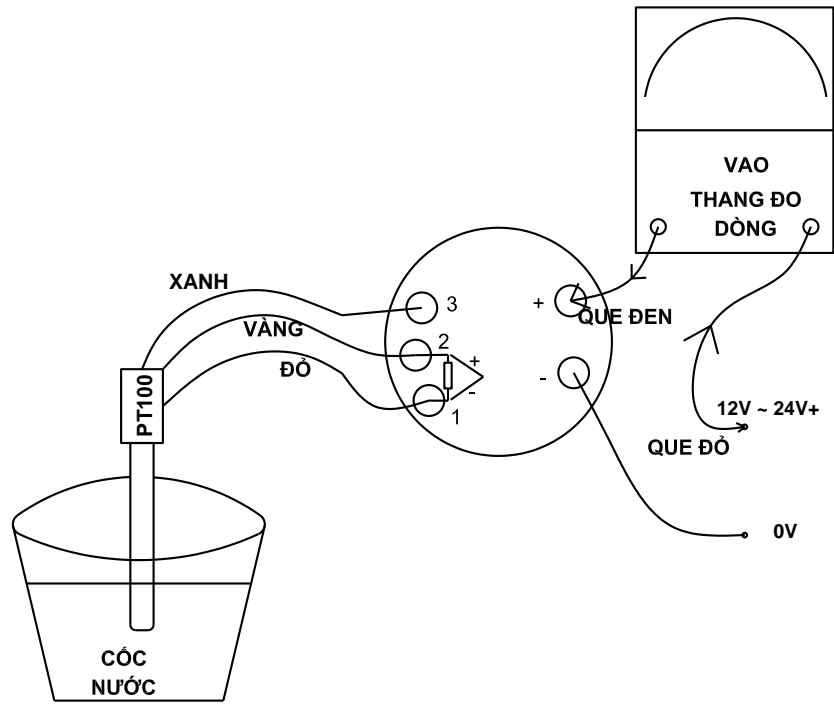
Kết quả không thay đổi khi đổi que 2 đầu que đo đỏ/đen.

1.3. Thí nghiệm đo dòng điện của Transducer

- Cho đá vào cốc ($t^{\circ} \sim 0^{\circ}\text{C}$), dùng VAO (loại số) để ở thang đo dòng, que đo đặt vào +24V như hình vẽ. Quan sát thấy VAO hiển thị 0,004A (4mA), D0 ~ 0.
- Đun sôi nước ($t^{\circ} \sim 100^{\circ}\text{C}$), VAO hiển thị 0,02A (20mA), D0 ~ 3200

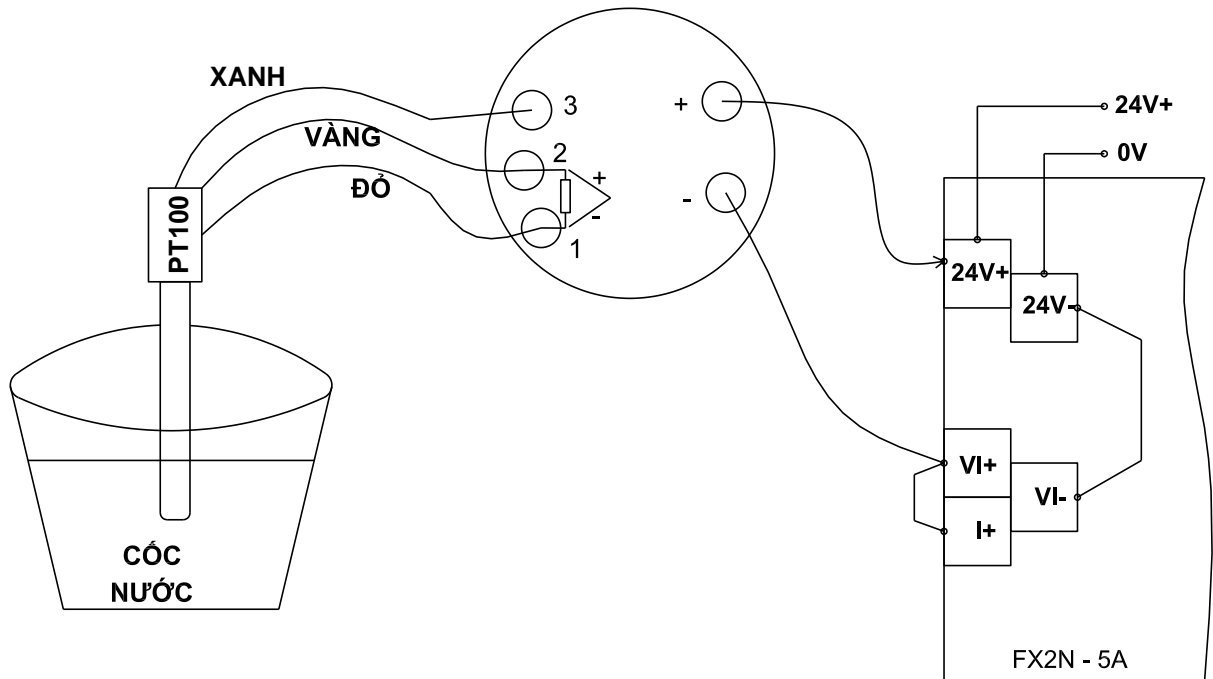


Hình 2.12-5. Đồ thị chuyển đổi tín hiệu ở chế độ 1



Hình 2.12-6. Sơ đồ thí nghiệm đo dòng điện của transducer

1.4. Viết chương trình đo và hiển thị nhiệt độ



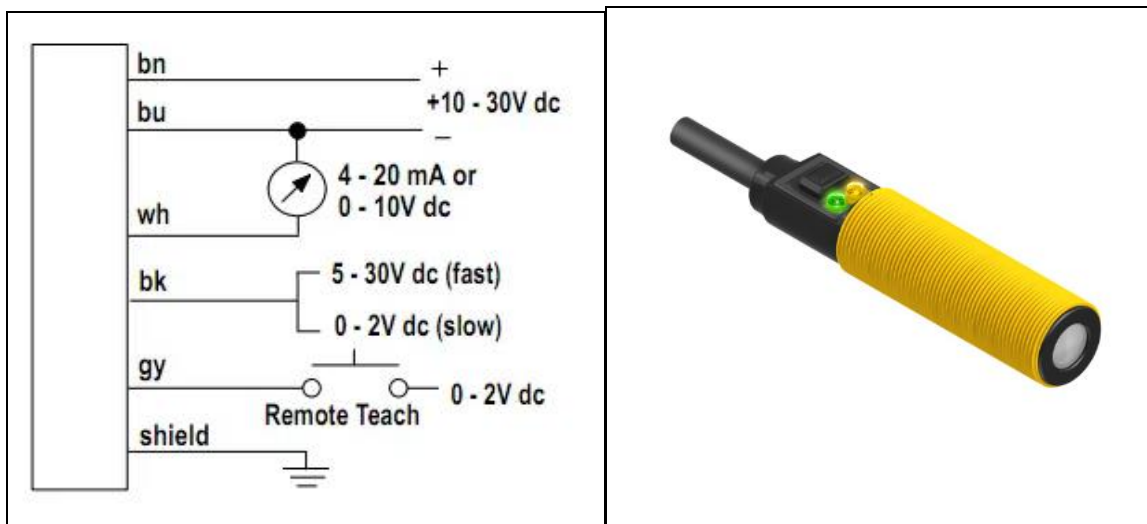
Hình 2.12-7. Sơ đồ nối dây transducer với module FX2N-5A

Chương trình lập trình:

<pre> M8002 ---[TO K0 K0 H1 K1] ---[TO K0 K2 K100 K1] M8000 ---[FROM K0 K6 D0 K1] ---[FLT D0 D4] ---[DEDIV D4 E320 D8] ---[END] </pre>	<p>Nạp H1 vào thanh ghi #0 cho module 1, kênh CH1 để đọc analog vào từ 4 – 20mA</p> <p>Nạp 100 (số lần lấy mẫu) vào K2</p>
	<p>Lấy dữ liệu từ K6 ra D0</p>
	<p>Đổi D0 ra số thực, lưu vào D4</p>
	<p>Chia D4 cho 320 để tính toán ra nhiệt độ tương ứng</p>

2. Bài tập ví dụ 2: Lập trình đo khoảng cách bằng cảm biến siêu âm S18UIA - Banner - USA

2.1. Thông số kỹ thuật cảm biến siêu âm

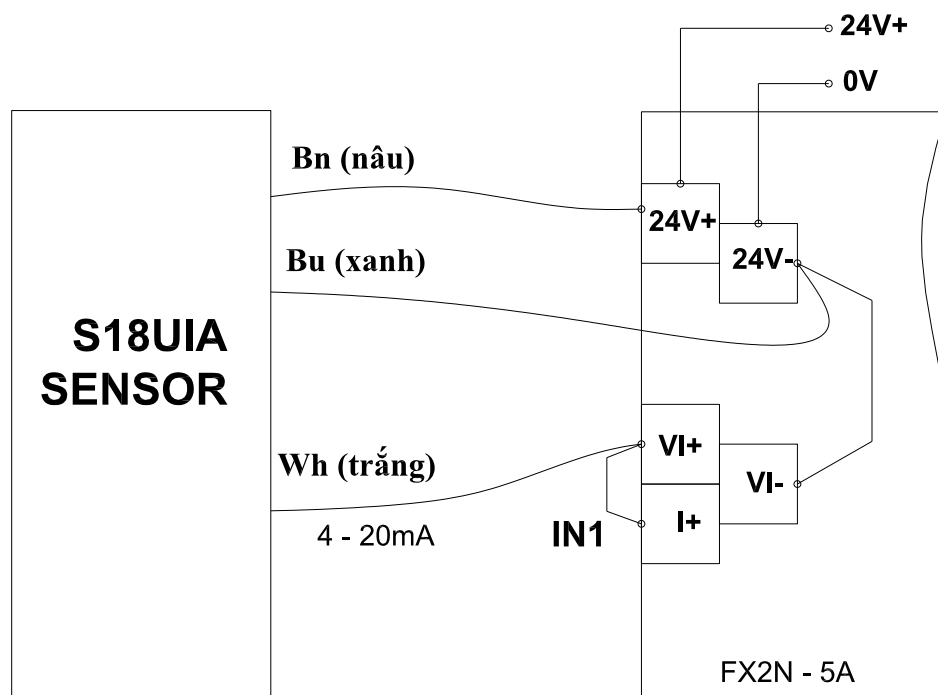


Hình 2.12-8. Cảm biến siêu âm S18UIA

Thông số:

- Điện áp nguồn: 10 – 30VDC
- Khoảng cách đo: 30mm – 300mm
- Độ phân giải: 2,5ms / 1mm
- Thời gian đáp ứng ngõ ra: 2,5ms nếu dây đen nối 5 – 50 VDC, 30ms nếu dây đen nối 0 – 2 VDC.

2.2. Sơ đồ đấu nối



Hình 2.12-9. Sơ đồ đấu nối cảm biến với module FX2N-5A

Sau khi đấu nối cảm biến với module analog xong, ta tiến hành kiểm tra cảm biến như sau:

- Viết chương trình đọc giá trị cảm biến như ở bài tập 1
- Đặt vật cách đầu cảm biến 30mm và quan sát giá trị D0
- Đặt vật cách đầu cảm biến 300mm và quan sát giá trị D0
- Sau đó tính toán giá trị Ex và thay vào câu lệnh: `-[DEDIV D4 Ex D8]-` để tính ra khoảng cách đo được.

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không đọc được giá trị vào D0	- Nạp các giá trị không đúng: thứ tự module, chọn chế độ, thanh ghi chức năng	- Kiểm tra lại giá trị nạp theo chức năng ở mục II.3
Giá trị tính toán không đúng với giá trị thực tế	Đo kiểm tra cảm biến chưa chính xác	Cần đo, kiểm tra xác định 2 giá trị min, max (như cách xác định ở 2 bài tập 1, 2)

IV. TIÊU CHÍ ĐÁNH GIÁ

- Thao tác đo, kiểm tra cảm biến nhiệt, cảm biến siêu âm []
- Đấu nối các thiết bị theo sơ đồ []

- Chương trình lập trình []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi sử dụng module analog []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Ký mã hiệu module analog
3. Sơ đồ kết nối sử dụng cảm biến nhiệt Pt100, cảm biến siêu âm S18UIA với module analog
4. Chương trình điều khiển
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

Lập chương trình xuất tín hiệu analog ra từ: 0 – 10 VDC?

Bài thực hành số 13

**THỰC HÀNH CÀI ĐẶT VÀ ĐIỀU KHIỂN
BIẾN TẦN MITSUBISHI FR-E700****I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Nhận biết được ký mã hiệu của biến tần Mitsubishi;
- Trình bày được ý nghĩa các tham số cài đặt cơ bản và trình tự cài đặt các tham số đó

Về kỹ năng:

- Cài đặt được các tham số cho biến tần chạy theo các chế độ khác nhau;
- Đấu nối được phần cứng theo sơ đồ nguyên lý;
- Lập được chương trình điều khiển chạy thuận – nghịch và dừng biến tần;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình sử dụng các câu lệnh lập trình.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình thực hành cài đặt biến tần;
- Có khả năng tự cài đặt và hướng dẫn được người khác cài đặt tham số biến tần;
- Tự đánh giá được kết quả thực hành của bản thân.

II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu biến tần Mitsubishi FR-E700****1.1. Ký mã hiệu và thông số kỹ thuật**

- Mã hiệu: FR – E720 – 0.75K

E720: Lớp điện áp 200V, 3 pha. Cho phép cấp 1 pha 220V đầu vào

0.75K: công suất biến tần.

E740: điện áp 3 pha 400V

E720S: loại điện áp vào 1 pha 200V

- Thông số kỹ thuật:

+ Điện áp cung cấp 3 pha : 200...240V, +/-10%, 48...63Hz;

+ Tần số ra: 0-500 Hz; Hệ số công suất 0.98

+ Dải công suất: 0.75 Kw (tùy theo loại)

+ Khả năng chịu quá tải: 150% trong 60s, 200% trong 3s.

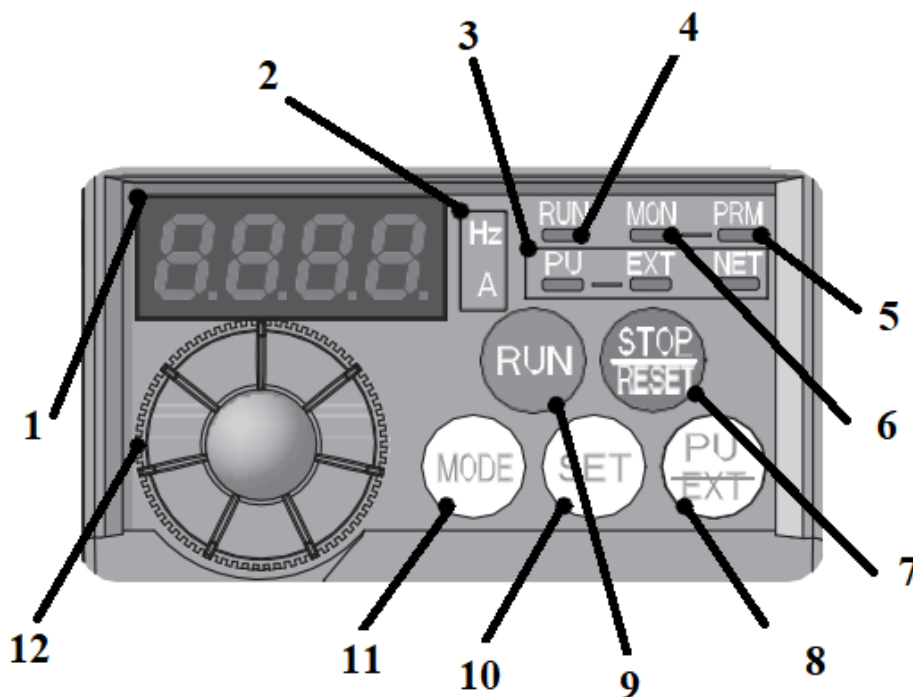
+ Chế độ điều khiển: Forward/Reveres, Multi speed, PID control, truyền thông...

- + Chức năng bảo vệ động cơ khi quá tải, ngắt mạch khi đang hoạt động.
- + Có chân kết nối điện trở thắng cho ứng dụng cần dừng nhanh.
- + Có thể gắn thêm card mở rộng I/O, card truyền thông.
- + Tích hợp thêm cổng kết nối màn hình rời, cổng USB kết nối với PC



Hình 2.13-1. Biến tần Mitsubishi FR – E720

1.2. Giới thiệu chức năng mặt điều khiển của biến tần



Hình 2.13-2. Bảng điều khiển biến tần Mitsubishi FR – E720

1 - Màn hình led 7 thanh: hiển thị các tham số cài đặt, các thông số về dòng điện, điện áp, tần số khi hoạt động.

2 – Đèn báo thông số nào được hiển thị trên led khi hoạt động:

- + Đèn Hz sáng → hiển thị tần số
- + Đèn A sáng → hiển thị dòng điện
- + Cả 2 đèn tắt → hiển thị điện áp

3 – Nhóm đèn báo chế độ hoạt động

- + Đèn PU sáng: chế độ điều khiển tại chỗ
- + Đèn EXT sáng: chế độ điều khiển từ bên ngoài
- + Đèn NET: chế độ điều khiển qua truyền thông mạng

4 – Đèn báo Run:

- + Sáng khi chạy thuận;
- + Nhấp nháy khi chạy nghịch

5 – Đèn báo chế độ giám sát

6 – Đèn báo chế độ cài đặt tham số

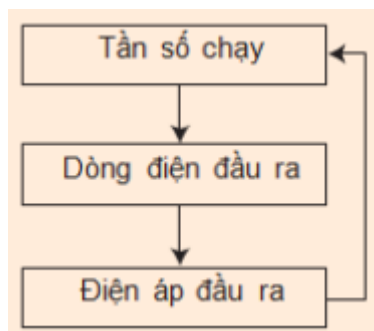
7 - STOP và RESET lỗi

8 – Chuyển đổi chế độ hoạt động PU/EXT

9 – RUN

10 – SET:

- + Nhấn để xác nhận giá trị các tham số cài đặt
- + Nếu nhấn khi đang chạy thì sẽ chuyển đổi các thông số giám sát:



11 – MODE: dùng để chuyển chế độ thiết lập, cài đặt tham số

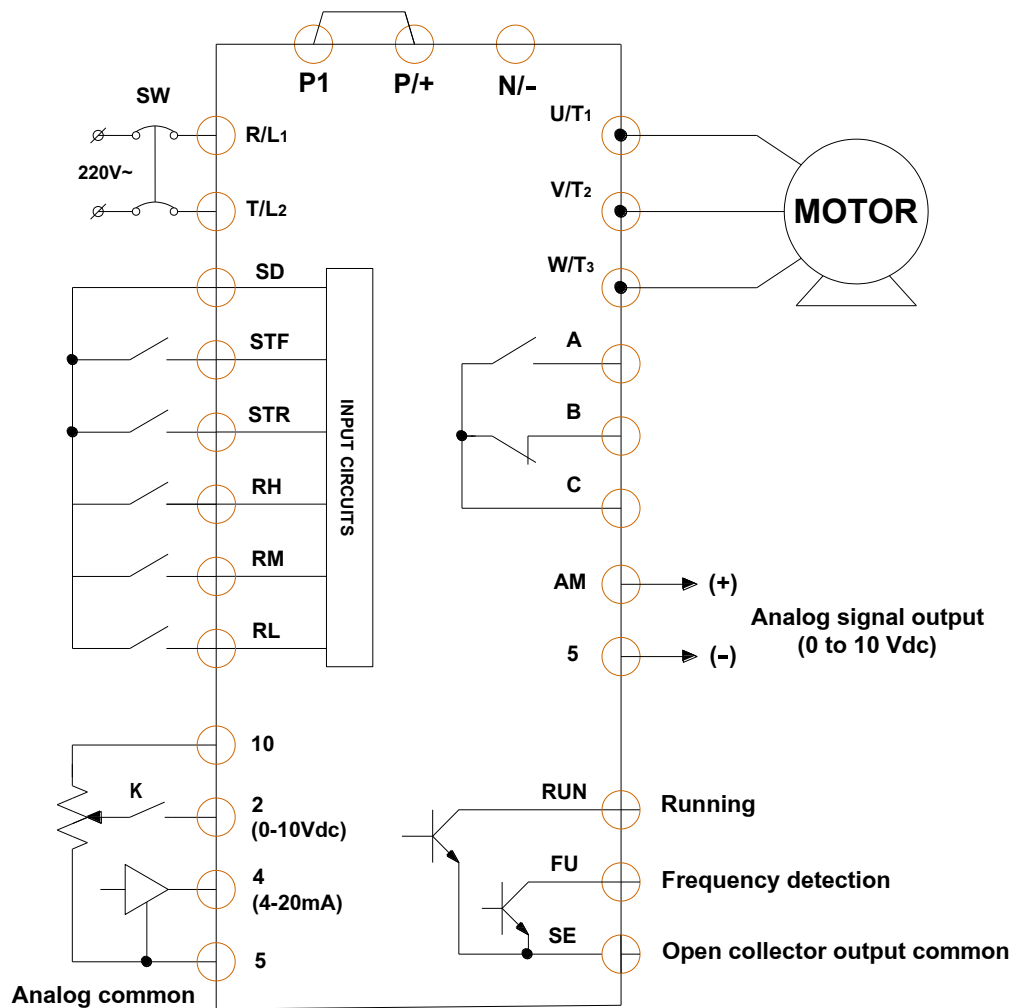
12 – Núm xoay: dùng để thay đổi cài đặt tần số, thiết lập tham số cài đặt

2. Sơ đồ đấu nối biến tần

Biến tần đã được đấu dây ra các cọc cắm trên Module biến tần của bàn thực hành.

Ý nghĩa các ký hiệu thường dùng:

STF	Quay thuận
STR	Quay nghịch
RH, RM, RL	Lựa chọn các cấp tốc độ Cao, trung bình, thấp
MRS	Stop
RES	Reset
SD	Nguồn 0VDC Làm chân chung để điều khiển ở chế độ SINK
PC	Nguồn 24VDC Làm chân chung để điều khiển ở chế độ SOURCE
Chân 10 và 5	Nguồn 5VDC và GND
Chân 2	Nhận tín hiệu điều khiển dạng điện áp
Chân 4	Nhận tín hiệu điều khiển dạng dòng



Hình 2.13-3. Sơ đồ đấu nối biến tần

3. Các tham số cài đặt cơ bản

Tham số	Ý nghĩa
P. 0	Tăng momen khởi động 0 ÷ 30% (5%)
P. 1	F_{max} : Tần số lớn nhất (50 hoặc 60Hz là phù hợp)
P. 2	F_{min} : Tần số nhỏ nhất (5Hz)
P. 3	Tần số cơ bản (50Hz)
P. 4	Tần số chạy tốc độ cao – khi RH đóng
P. 5	Tần số chạy tốc độ trung bình – khi RM đóng
P. 6	Tần số chạy tốc độ thấp – khi RL đóng
P. 7	Thời gian tăng tốc
P. 8	Thời gian giảm tốc
P. 72	Tần số PWM (0, 1, ..., 15), tương đương với dải tần số PWM từ 0,7 ÷ 14,5 kHz. Nếu P. 72 = 0: động cơ chạy kêu to.
P. 73	Thiết lập điện áp analog vào chân 2: (0, 1, 10, 11)

	<p>0: $V_{in} = 0 \div 10V$ (không cho phép đảo chiều bằng tín hiệu) 1: $V_{in} = 0 \div 5V$ (không cho phép đảo chiều bằng tín hiệu) 10: $V_{in} = 0 \div 10V$ (cho phép đảo chiều bằng tín hiệu) 11: $V_{in} = 0 \div 5V$ (cho phép đảo chiều bằng tín hiệu)</p>
P. 77	<p>0: Chỉ cho phép sửa dữ liệu khi đang dừng 1: Cấm ghi, sửa dữ liệu 2: Cho phép sửa dữ liệu trong lúc chạy và dừng</p>
P. 78	<p>0: Cho phép chạy 2 chiều thuận, ngược 1: Cấm chạy ngược 2: Cấm chạy thuận</p>
P. 79	<p>Thiết lập chế độ hoạt động (0, 1, ...,7) 0: chạy 2 chế độ PU (bàn phím + nút vận) và EXT (tù xa), dùng phím PU/EXT để chuyển đổi 2 chế độ hoạt động 1: Chỉ chạy chế độ PU - trực tiếp trên bàn phím 2: Chạy từ xa với tín hiệu Analog vào chân 2, chạy thuận, nghịch qua công tắc STF, STR. 3: Chạy một trong các tốc độ cao, trung bình, thấp qua các công tắc RH, RM, RL. Chạy thuận, nghịch qua STF, STR <i>* Khi P. 79 =2 thì ấn PU không chuyển được về chế độ sửa các tham số khác, muốn sửa các tham số thì cần đặt P. 79 = 0</i></p>
P. 340	<p>Chọn chế độ truyền thông: (0, 1, 10) 0: không truyền thông, chế độ hoạt động được thiết lập trong P. 79 1: Chỉ thiết lập chế độ truyền thông NET; 10: Thiết lập 2 chế độ hoạt động PU và NET</p>

III. NỘI DUNG THỰC HÀNH

1. Chuẩn bị dụng cụ, thiết bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u gồm: module biến tần, PLC, động cơ	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Các bước thực hiện

2.1. Cài đặt tham số biến tần

- Xem và kiểm tra tham số:

Cấp điện → nhấn MODE để chuyển sang chế độ PRM (Parameter) → xuất hiện P. 0 → xoay núm xoay để chuyển đến tham số (P.) tương ứng

→ Nhấn SET để xem giá trị → Nếu nhấn SET tiếp thì nhảy sang P. Kế tiếp.

- Cài đặt tham số: Khi cài đặt tham số thì biến tần phải ở chế độ STOP

→ Cấp điện → nhấn MODE để chuyển sang chế độ PRM → xoay núm xoay để chuyển đến tham số P. 79 → nhấn SET → xoay núm xoay để cài đặt giá trị bằng 0 → nhấn SET để kết thúc (cần đặt P. 79 = 0 thì mới cho phép sửa các tham số khác

→ Xoay núm vặn chuyển đến tham số P. cần cài đặt

→ Nhấn SET để hiển thị giá trị của P. đó

→ Xoay núm vặn để thay đổi giá trị

→ Nhấn SET 2 lần để thiết lập giá trị mới cho P.

→ Đặt lại giá trị cho P. 79 như lúc đầu.

→ Ngắt nguồn điện của biến tần để nhớ các thay đổi vừa thực hiện.

2.2. Điều khiển tại chỗ

Đây là phương pháp dùng bảng điều khiển trên mặt biến tần để điều khiển.

- Bước 1: Cài đặt P. 79 = 0; Ngắt nguồn điện biến tần rồi bật lại.

- Bước 2: nhấn MODE để chuyển về chế độ giám sát MON để quan sát các thông số. Nhấn SET để chuyển quan sát các thông số giám sát khác: Hz, A, U.

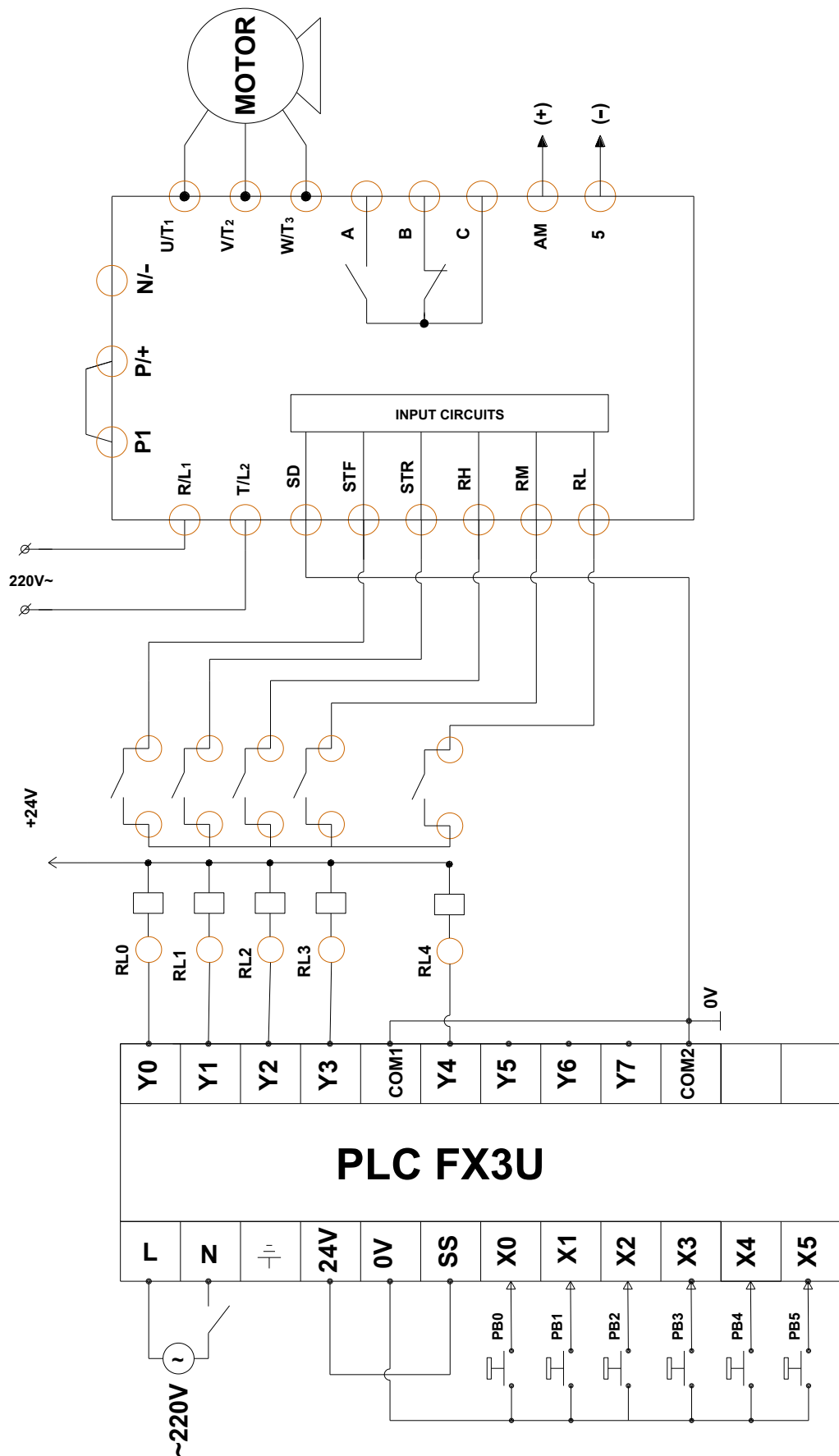
- Bước 3: nhấn PU/EXT để đèn PU sáng (chạy tại chỗ)

- Bước 4: Nhấn Run để chạy, điều chỉnh tần số bằng núm xoay, nhấn STOP để dừng.

2.3. Điều khiển từ xa EXT chạy 3 cấp tốc độ

2.3.1. Sơ đồ đấu nối

Kết nối đầu và ra các module thực hành theo sơ đồ sau:



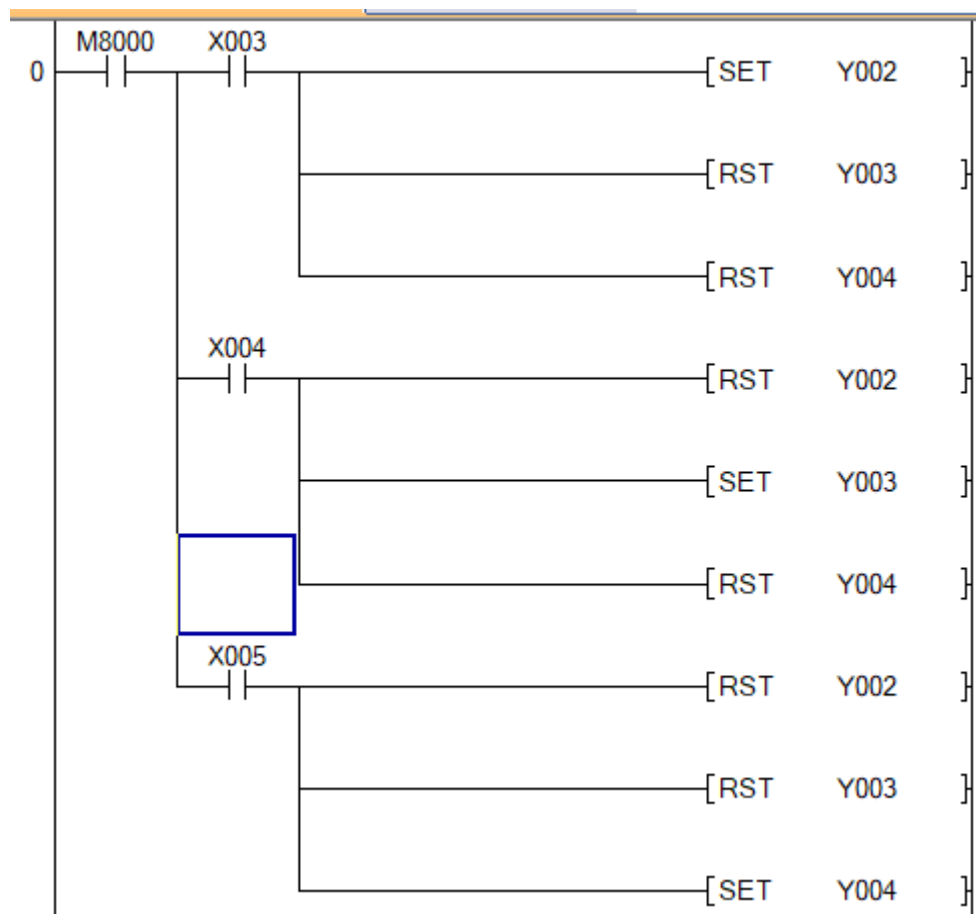
Hình 2.13-4. Sơ đồ đấu nối vào ra PLC – biến tần

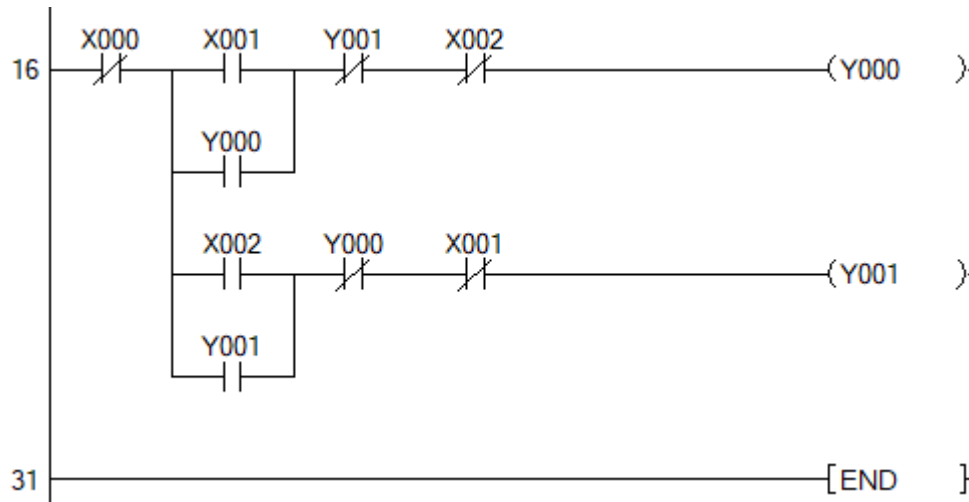
2.3.2. Cài đặt biến tần

- Cài đặt P. 79 = 0 để cho phép cài đặt các tham số khác
- Cài đặt các tham số các cấp tốc độ:
 - + Tốc độ cao: P. 4 = 60
 - + Tốc độ trung bình P. 5 = 30
 - + Tốc độ thấp: P. 6 = 10
- Cài đặt P. 79 = 3 để cho phép chạy các cấp tốc độ
- Ngắt nguồn cấp cho biến tần rồi bật trở lại để biến tần nhớ các tham số cài đặt.

2.3.3. Lập chương trình điều khiển

Đầu vào		Đầu ra	
X000	Dừng	Y000	RL0 → STF
X001	Chạy thuận	Y001	RL1 → STR
X002	Chạy ngược	Y002	RL2 → RH
X003	Tốc độ cao	Y003	RL3 → RM
X004	Tốc độ trung bình	Y004	RL4 → RL
X005	Tốc độ thấp		



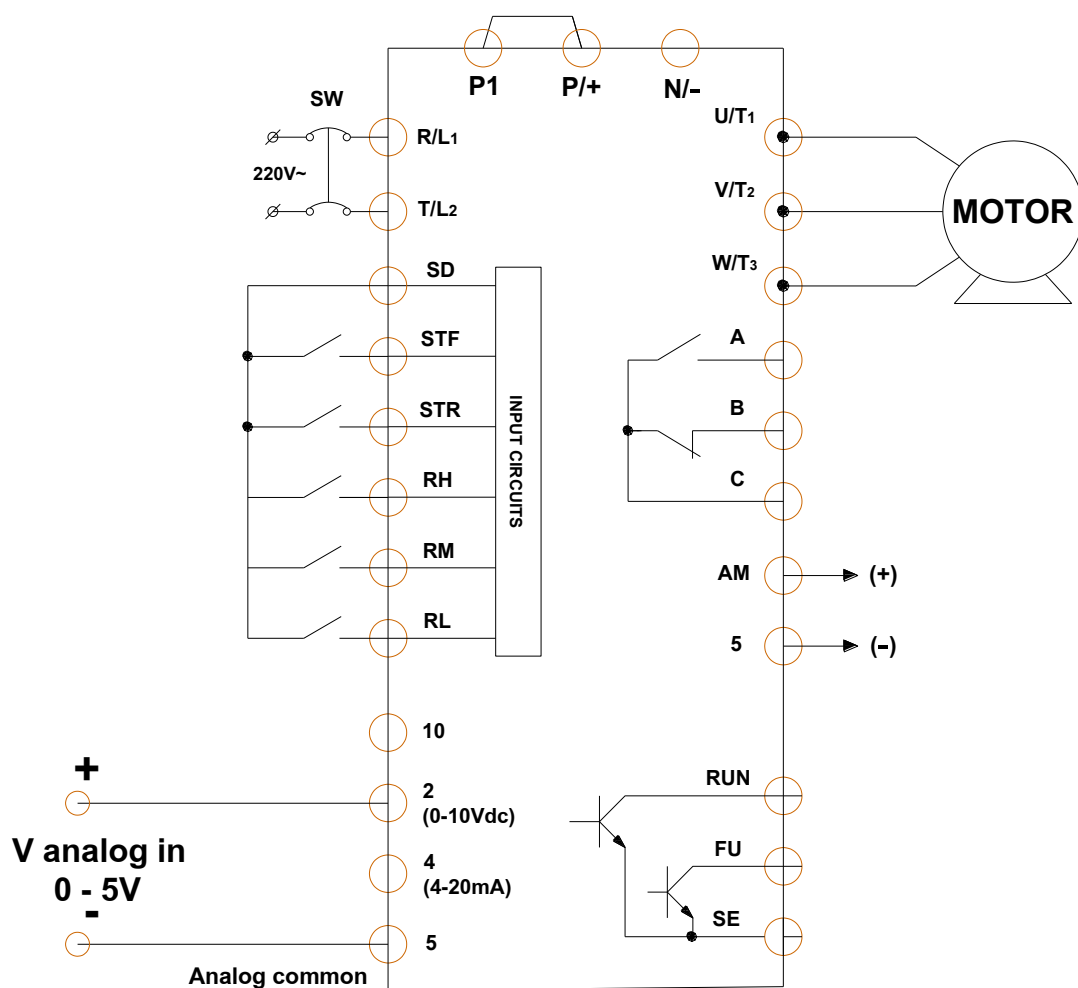


2.3.4. Vận hành

- Tải chương trình cho PLC
- Vận hành chạy thuận, ngược và chạy các cấp tốc độ khác nhau

2.3. Điều khiển từ xa EXT sử dụng tín hiệu analog

2.3.1. Sơ đồ đấu nối



Hình 2.13-5. Sơ đồ đấu điều khiển biến tần bằng tín hiệu analog ngoài

2.3.2. Cài đặt tham số liên quan:

- Cài đặt P. 79 = 2; (cần đặt P. 79 = 0 và cài đặt các tham số khác trước, sau đó mới đặt P. 79 = 2).

- Bảng cài đặt tham số P. 73, P. 184, P. 267 theo các chế độ khác nhau:

P. 184	P. 73	Tín hiệu vào chân số 2	Tín hiệu vào chân số 4	Chế độ chạy đảo chiều quay
Khác 4	0	0 ÷ 10V	Không sử dụng	Không cho phép đảo chiều bằng tín hiệu
	1	0 ÷ 5V		cho phép đảo chiều bằng tín hiệu
	10	0 ÷ 10V		
	11	0 ÷ 5V		
4	0	Không sử dụng	P. 267: 0: 4 ÷ 20mA 1: 0 ÷ 5V 2: 0 ÷ 10V	không cho phép đảo chiều bằng tín hiệu
	1			Cho phép đảo chiều bằng tín hiệu
	10			
	11			

* **Ví dụ 1:** Cài đặt biến tần để đưa tín hiệu điều khiển dòng 4 ÷ 20mA vào chân 4 và 5:

P. 79 = 2
 P. 73 = 0 hoặc 1
 P. 184 = 4
 P. 267 = 0
 Chú ý: cần gạt công tắc trên bo mạch để chuyển V → I:

I

V

* **Ví dụ 2:** Cài đặt biến tần để đưa tín hiệu điều khiển dạng điện áp 0 ÷ 10V vào chân 2 và 5; không dùng đảo chiều quay bằng tín hiệu điều khiển:

P. 79 = 2

P. 73 = 0

P. 184 đặt khác giá trị 4

P. 267 không quan tâm

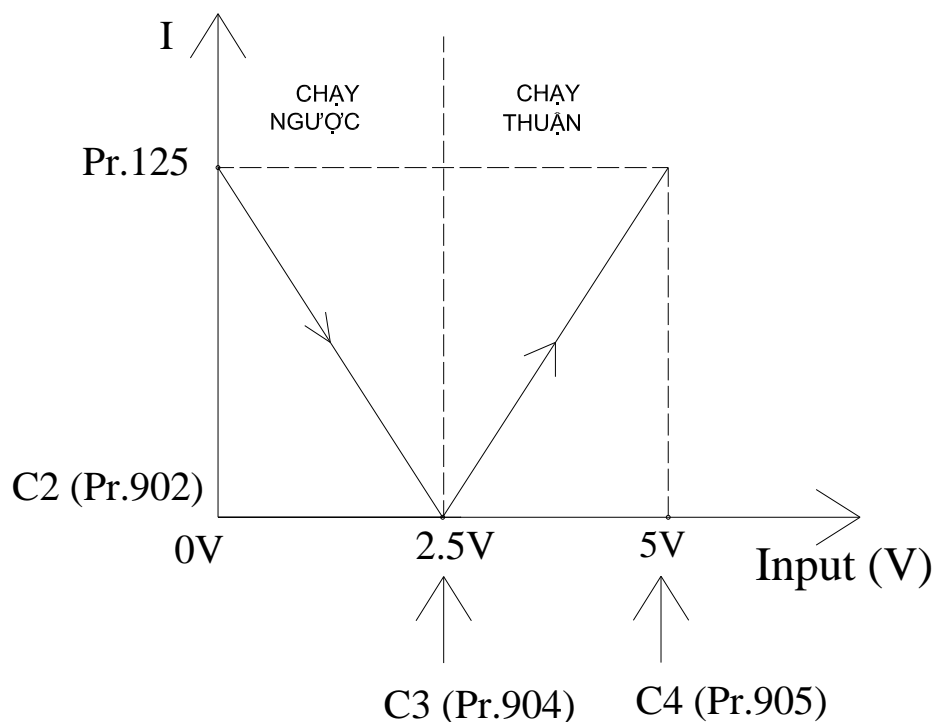
Đảo chiều bằng các chuyển mạch STF và STR.

* **Ví dụ 3:** Cài đặt biến tần để đưa tín hiệu điều khiển dạng điện áp 0 ÷ 5V vào chân 2 và 5; trong đó với điện áp từ 0 ÷ 2,5V động cơ quay ngược, 2,5 ÷ 5V động cơ quay thuận:

+ P. 79 = 2

+ P. 184 = 0









+ P. 73 = 11



Hình 2.13-6. Minh họa điều khiển đảo chiều bằng tín hiệu analog

- + P. 125 = 60 (Hz) : Tần số lớn nhất
- + C2 = 0 (Hz) điểm bắt đầu; C2 là parameter P. 902
- + C3 = 2.5V hoặc 50% (P.241 = 1 thì cài đặt giá trị V/mA, = 0 là cài đặt %)
- + C4 = 5V hoặc 100%
- + Trình tự cài đặt:

1. STOP biến tần, nhấn PU/EXT để chọn PU	
2. Nhấn MODE để chọn PRM, xuất hiện P0	
3. Xoay núm vặn để xuất hiện C...	
4. Nhấn SET cho đến khi xuất hiện C---	
5. Xoay núm vặn đến các giá trị C2, C3 Nhấn SET để hiển thị giá trị các tham số	

Vặn núm xoay để thay đổi giá trị cài đặt Nhấn SET để xác nhận cài đặt	
6. Xoay núm vặn để xuất hiện 	 →  Terminal 2 input is selected  Terminal 4 input is selected
7. Nhấn SET để hiển thị giá trị V/mA hoặc % (tùy thuộc P.241)	 → 
8. Vặn chiết áp nguồn analog và đo $U_{vào} = 5V$, nhập giá trị 5 (V) hoặc 100% (tùy thuộc P.241)	 → 
9. Nhấn SET để cài đặt giá trị này (đèn nhấp nháy 2 lần rồi tắt)	

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Biến tần không cho cài đặt tham số	Chưa cài đặt đúng P. 79	Đặt P.79 = 0
Không đảo chiều động cơ được bằng các tiếp điểm STF, STR	Chưa cài đặt đúng P. 78	Đặt P.78: 0: Cho phép chạy 2 chiều thuận, ngược 1: Cấm chạy ngược 2: Cấm chạy thuận
Không đảo chiều động cơ được bằng tín hiệu analog ngoài	Chưa cài đặt đúng P. 184, P. 73, P. 267, C2, C3, C4	Cài đặt lại các tham số như ví dụ 3 mục 2.3.2.
Biến tần không lưu được tham số đã cài đặt	Chưa khởi động lại biến tần	Ngắt nguồn rồi cấp lại cho biến tần khởi động lại để nhớ các cài đặt

IV. TIÊU CHÍ ĐÁNH GIÁ

- Thao tác cài đặt biến tần []
- Sơ đồ, và thao tác đấu nối thiết bị []
- Chương trình lập trình điều khiển biến tần []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi sử dụng biến tần []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Ý nghĩa các tham số cơ bản
3. Sơ đồ đấu nối phần cứng cho các chế độ điều khiển
4. Chương trình điều khiển
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

Cài đặt và điều khiển biến tần bằng tín hiệu dòng 4 – 20mA, có đảo chiều bằng tín hiệu?

Bài thực hành số 14

**LẬP TRÌNH ĐIỀU KHIỂN BIẾN TẦN MITSUBISHI FR-E700
SỬ DỤNG MODULE ANALOG FX2N-2DA****I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Trình bày được ý nghĩa các thanh ghi chức năng của Module analog Fx2N-2DA và các câu lệnh đọc ghi dữ liệu cho module;

- Trình bày được ý nghĩa và trình tự cài đặt các tham số để biến tần hoạt động ở chế độ điều khiển analog từ bên ngoài.

Về kỹ năng:

- Cài đặt được các tham số cho biến tần hoạt động ở chế độ điều khiển analog từ bên ngoài;

- Đấu nối được phân cứng theo sơ đồ nguyên lý;

- Lập được chương trình điều biến tần sử dụng module analog FX2N – 2DA;

- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình sử dụng các câu lệnh lập trình.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình đấu nối, lập trình điều khiển biến tần;

- Có khả năng làm việc độc lập và làm việc nhóm. Hướng dẫn được người khác thực hành.

- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu module analog FX2N - 2DA****1.1. Giới thiệu chung**

- Module analog FX2N - 2DA sử dụng để biến đổi tín hiệu số 12 bit thành tín hiệu tương tự dạng dòng hoặc áp:

+ Tín hiệu dạng điện áp: $0 \div 10\text{VDC}$

+ Tín hiệu dạng dòng điện: $4 \div 20\text{mA}$

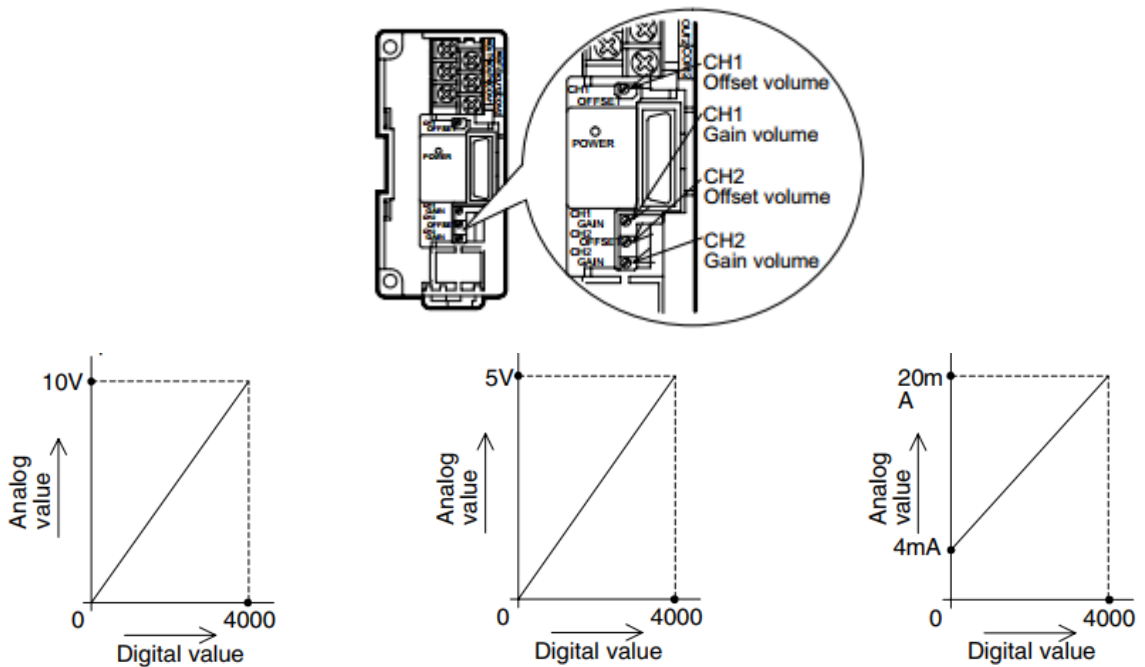
- Module analog FX2N - 2DA có thể kết nối với các dòng PLC như: FX0N, FX1N, FX2N, FX2NC, FX3G, FX3GC, FX3U và FX3UC.

- Kết nối với PLC bằng cáp mở rộng (kết nối bên hông PLC)

- Giá trị analog đầu ra biến thiên từ $0 \div 10\text{VDC}$ (hoặc $4 \div 20\text{mA}$) tùy theo giá trị số nạp cho module trong dải từ: $0 \div 4000$.



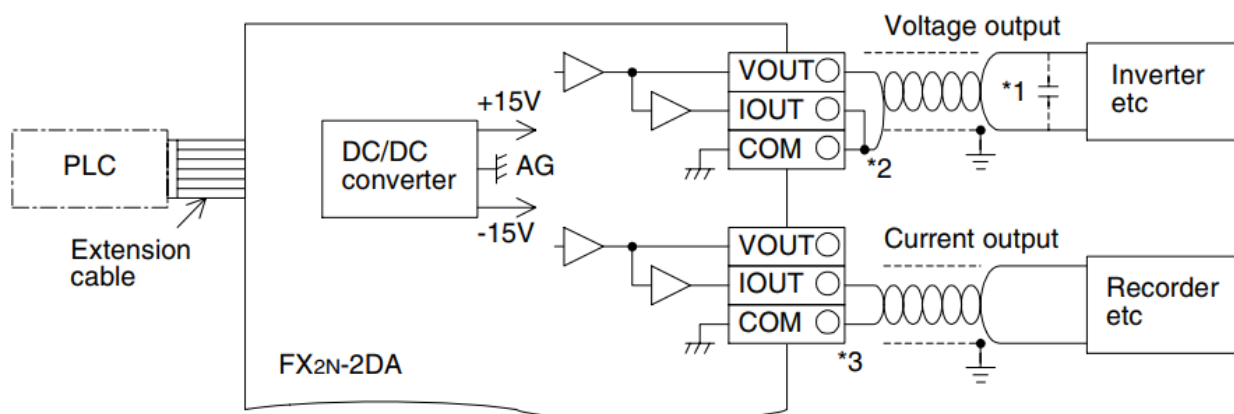
Hình 2.14-1a. Module analog FX2N – 2DA



Hình 2.14-1b. Nút vặn để hiệu chỉnh dải tín hiệu

Chú ý: Có thể sử dụng các nút vặn Gain và Offset để điều chỉnh dải điện áp xuất ra các kênh từ 0 ÷ 10VDC hoặc từ 0 ÷ 5VDC

1.2. Sơ đồ đấu dây



Hình 2.14-2. Sơ đồ kết nối vào ra FX2N- 2DA

1.3. Các thanh ghi đặc biệt

Sử dụng 2 thanh ghi 16 bit: #16 và #17, cụ thể như sau:

Thanh Ghi	b15 ÷ b8	b7 ÷ b3	b2	b1	b0
#16	Không sử dụng	Dữ liệu sử dụng cho đầu ra (8 bit)			
#17	Không sử dụng	bit lựa chọn lưu 8 bit thấp của dữ liệu	Bit cho phép CH1 bắt đầu hoạt động	Bit cho phép CH2 bắt đầu hoạt động	

- Thanh ghi #16: thanh ghi này chỉ sử dụng 8 bit thấp để lưu dữ liệu, mà giá trị lớn nhất của dữ liệu cần nạp vào module là 4000 (cần 12 bit lưu trữ). Do đó dữ liệu trước khi lưu vào #16 cần được chia thành 2 phần: 8 bit thấp và 4 bit cao (tùy thuộc vào bit b2 của thanh ghi #17).

- Thanh ghi #17:

- + Bit b0: cho phép CH2 bắt đầu hoạt động khi b0 chuyển trạng thái từ 1 → 0
- + Bit b1: cho phép CH1 bắt đầu hoạt động khi b1 chuyển trạng thái từ 1 → 0
- + Bit b2: cho phép lưu trữ 8 bit thấp của dữ liệu nạp vào thanh ghi #16 khi b2 chuyển trạng thái từ 1 → 0

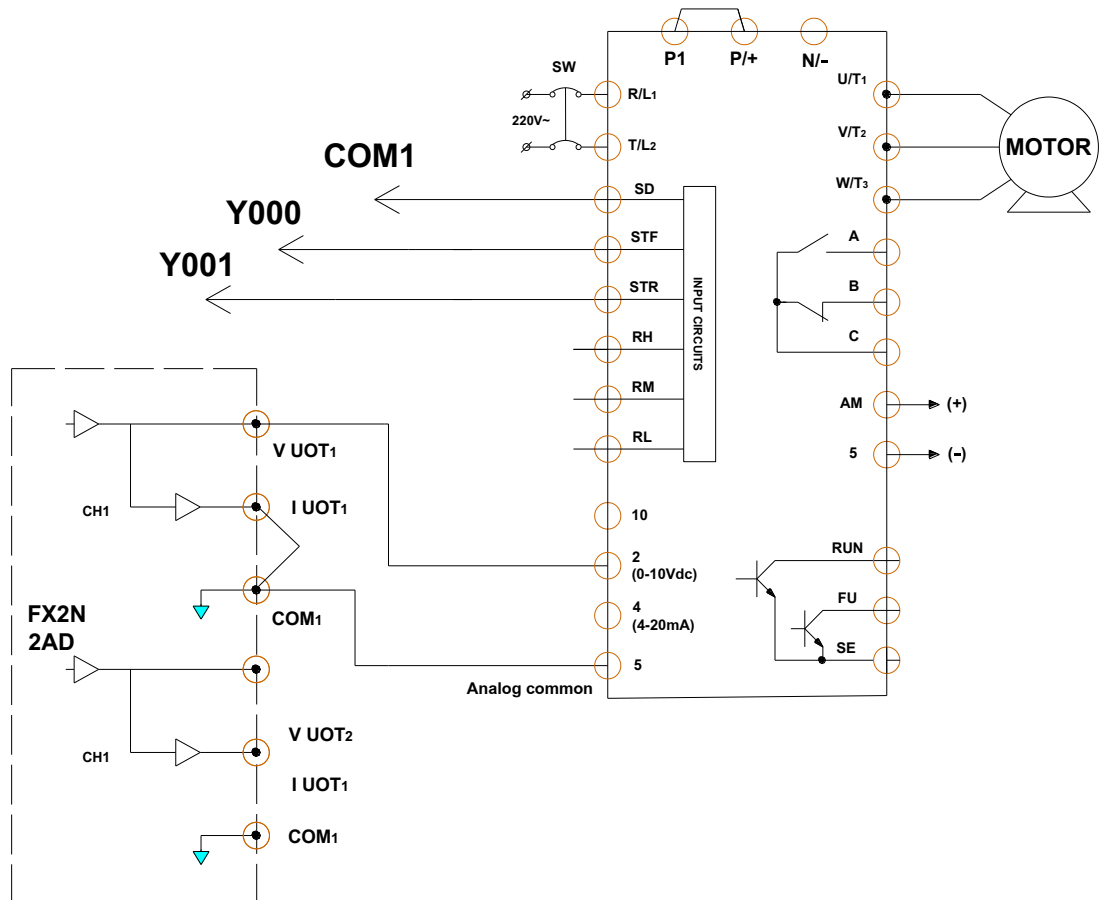
III. NỘI DUNG THỰC HÀNH

1. Công tác chuẩn bị

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u gồm: Module PLC, module analog FX2N – 2DA, module biến tần, PLC, động cơ	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Trình tự thực hiện

2.1. Sơ đồ đấu nối



Hình 2.14-3. Sơ đồ đấu nối thực hành

2.2. Cài đặt tham số biến tần

Cài đặt biến tần để đưa tín hiệu điều khiển dạng điện áp $0 \div 10V$ vào chân 2 và 5; Đảo chiều bằng các chuyển mạch STF và STR.

P. 79 = 2

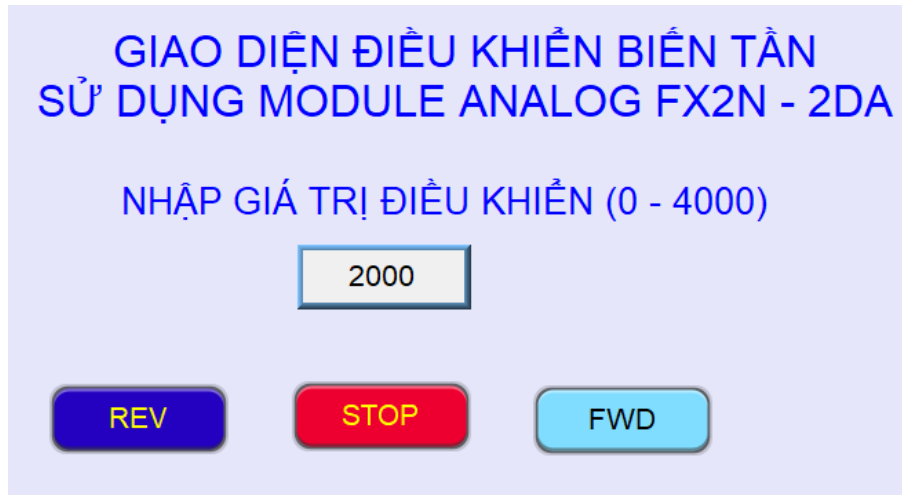
P. 73 = 0

P. 184 đặt khác giá trị 4

2.3. Thiết kế giao diện HMI điều khiển

Thiết kế giao diện điều khiển bao gồm:

Tên chức năng	Thanh ghi / Bit	Kiểu đối tượng
Ô nhập số nguyên (0 – 4000)	D20	16 bit Unsigned
Nút dừng	M0	Momentary
Nút chạy thuận	M1	Momentary
Nút chạy ngược	M2	Momentary



Hình 2.14-4. Giao diện điều khiển biến tần

2.4. Lập chương trình điều khiển

	<p>a- Nạp dữ liệu từ D20 vào 16 bit trung gian: M15 ÷ M0</p> <p>b- Ghi 8 bit thấp của dữ liệu vào FX2N-2DA</p> <p>c- Lưu 8 bit thấp của dữ liệu</p> <p>d- ghi 4 bit cao của dữ liệu vào FX2N-2DA</p> <p>e- Đưa điện áp điều khiển ra kênh CH1</p>
	<p>Điều khiển chạy thuận, ngược và dừng biến tần</p>

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không đảo chiều động cơ được bằng các tiếp điểm STF, STR	Chưa cài đặt đúng P. 78	Đặt P.78: 0: Cho phép chạy 2 chiều thuận, ngược 1: Cấm chạy ngược 2: Cấm chạy thuận
Không điều khiển được từ HMI	Lỗi kết nối HMI - PLC	Kiểm tra lại cài đặt khi thiết kế giao diện
Không xuất được tín hiệu analog điều khiển ra biến tần	- Kết nối phần cứng - Chương trình lập trình	- Kiểm tra kết nối: <i>Hình 2.14-3</i> - Kiểm tra chương trình điều khiển: mục 2.4.

IV. TIÊU CHÍ ĐÁNH GIÁ

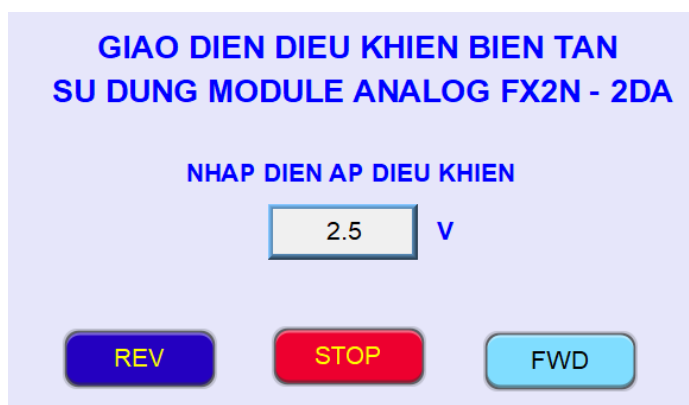
- Thao tác cài đặt biến tần []
- Sơ đồ, và thao tác đấu nối thiết bị []
- Chương trình lập trình điều khiển biến tần sử dụng module FX2N-2DA []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi sử dụng bộ đếm []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Giới thiệu module FX2N-2DA
3. Sơ đồ kết nối PLC - FX2N-2DA - Biến tần
4. Chương trình điều khiển
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

1. Thiết kế giao diện và lập chương trình điều khiển để nhập vào giá trị điện áp điều khiển biến tần.



Bài thực hành số 15

LẬP TRÌNH ĐO VÀ HIỂN THỊ TỐC ĐỘ ĐỘNG CƠ**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Trình bày được trình tự đo và hiển thị tốc độ động cơ;
- Trình bày được ý nghĩa và trình tự cài đặt các tham số để biến tần hoạt động ở chế độ điều khiển analog từ bên ngoài.

Về kỹ năng:

- Cài đặt được các tham số cho biến tần hoạt động ở chế độ điều khiển analog từ bên ngoài;
- Đấu nối được phần cứng theo sơ đồ nguyên lý;
- Thiết kế được giao diện HMI để điều khiển và giám sát tốc độ động cơ;
- Lập được chương trình đo và hiển thị tốc độ động cơ;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình thực hành.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình thực hành;
- Có khả năng làm việc độc lập và làm việc nhóm;
- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu chung về bộ thực hành**

Hình 2.15-1. Bộ thực hành đo hiển thị tốc độ động cơ

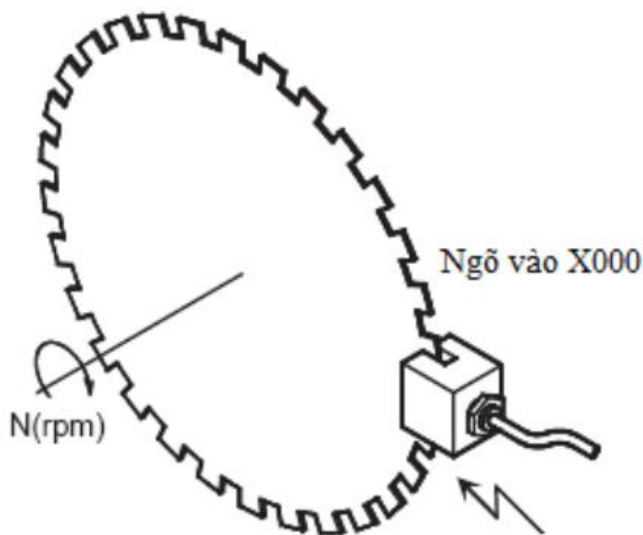
Bộ thực hành bao gồm các khối:

- Màn hình HMI
- PLC và module Analog
- Biến tần và động cơ
- Encoder: gắn trên trục của băng tải; loại 100 P/R, NPN. (chi tiết xem tại

bài thực hành số 10.

2. Phương pháp tính tốc độ động cơ

2.1. Lệnh đọc tốc độ xung



Hình 2.15-2. Minh họa cơ cấu đếm xung của encoder

- Lệnh đo tốc độ xung: **-[SPD S1 S2 D0]-**

Trong đó: + S1: đầu vào PLC kết nối với xung encoder (X000)

+ S2: Khoảng thời gian đếm xung, đơn vị là ms

+ D0: lưu số đếm được trong khoảng thời gian S2

2.2. Công thức tính tốc độ động cơ N (vòng/phút):

$$N = \frac{60 \times D0}{n \times S2} \times 1000 \text{ (vòng/phút)}$$

Trong đó: N: tốc độ động cơ (vòng/phút)

n: số xung/vòng của encoder (p/r)

Ví dụ: tính tốc độ quay động cơ với: n = 360 p/r, khoảng thời gian đếm xung S2 = K100.

$$N = \frac{60 \times D0}{360 \times 100} \times 1000 = \frac{10 \times D0}{6} \text{ (vòng/phút)}$$

III. NỘI DUNG THỰC HÀNH

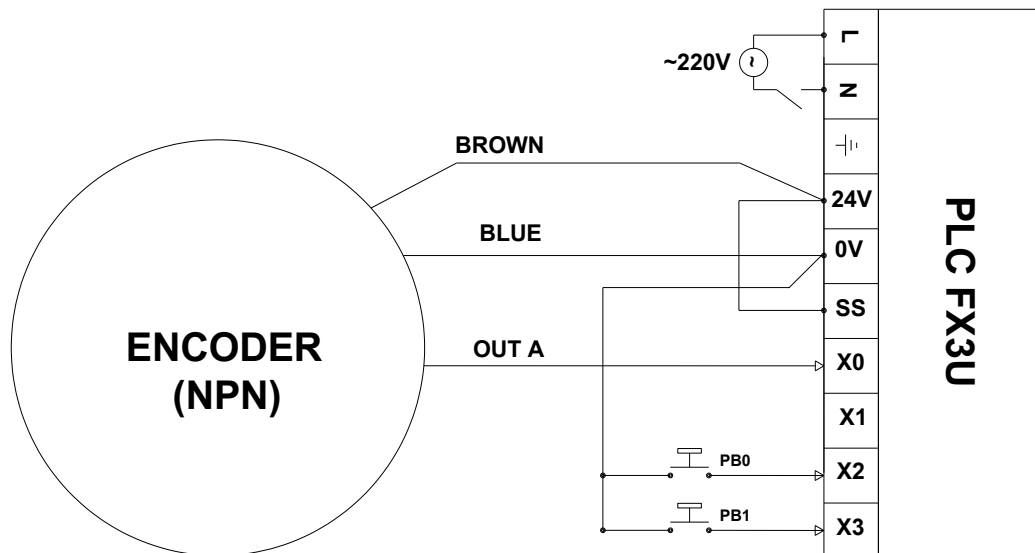
1. Công tác chuẩn bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

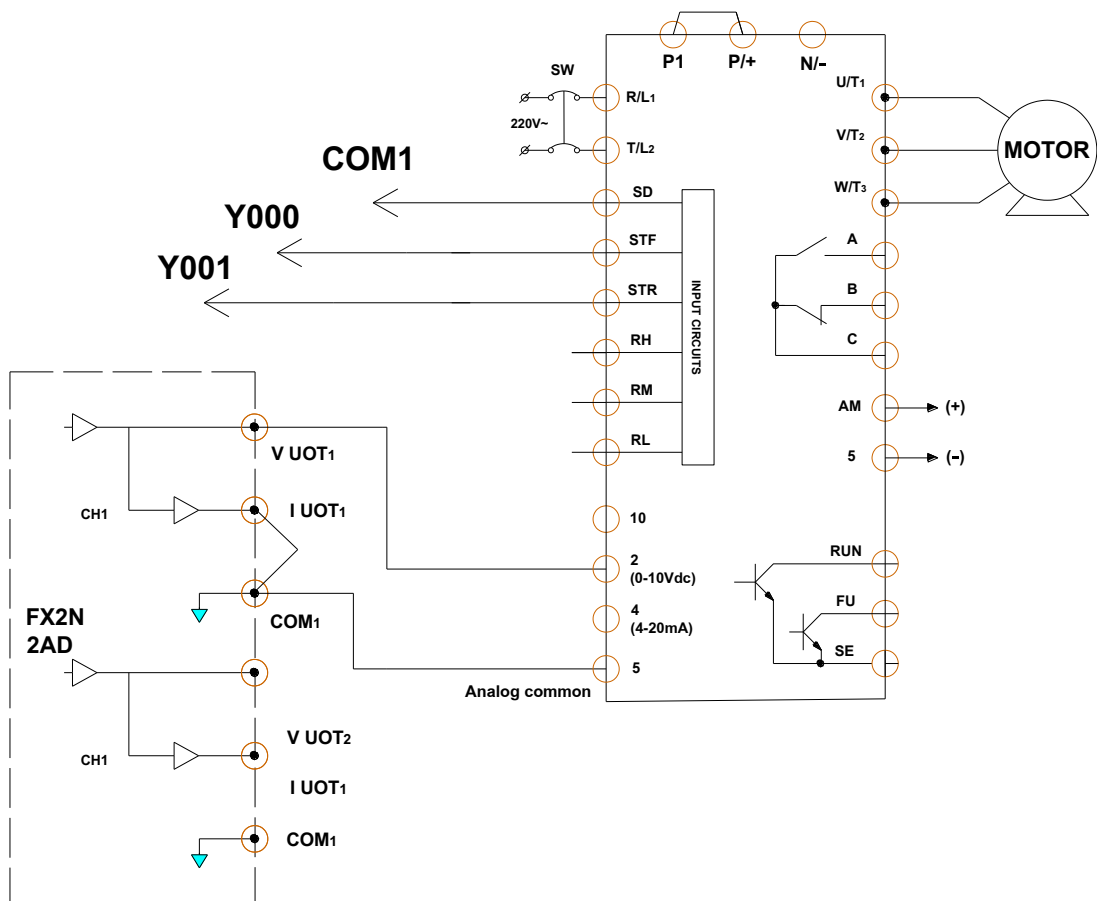
STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u gồm: Module PLC, module analog FX2N – 2DA, module biến tần, PLC, động cơ, Encoder, HMI	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Trình tự thực hiện

2.1. Sơ đồ đấu nối



Hình 2.15-3. Sơ đồ đấu nối encoder – PLC



Hình 2.15-4. Sơ đồ đấu nối đầu ra PLC, FX2N-2DA và biến tần

2.2. Cài đặt tham số biến tần

Cài đặt biến tần để đưa tín hiệu điều khiển dạng điện áp $0 \div 10V$ vào chân 2 và 5; Đảo chiều bằng các chuyển mạch STF và STR.

P. 79 = 2

P. 73 = 0

P. 184 đặt khác giá trị 4

2.3. Thiết kế giao diện HMI điều khiển

Thiết kế giao diện điều khiển bao gồm:

Tên chức năng	Thanh ghi / Bit	Kiểu đối tượng
Ô nhập số nguyên (0 – 4000)	D20	16 bit Unsigned
Ô hiển thị tốc độ	D14	16 bit Unsigned
Nút dừng	M0	Momentary
Nút chạy thuận	M1	Momentary
Nút chạy ngược	M2	Momentary

GIAO DIỆN ĐIỀU KHIỂN BIẾN TẦN SỬ DỤNG MODULE ANALOG FX2N - 2DA

NHẬP GIÁ TRỊ ĐIỀU KHIỂN (0 - 4000)

0

TỐC ĐỘ HIỆN TẠI

0

v/p

REV

STOP

FWD

Hình 2.14-4. Giao diện điều khiển biến tần

2.4. Lập chương trình điều khiển

<pre> M8000 ├── [MOV D20 K4M100] ├── [TO K0 K16 K2M100 K1] ├── [TO K0 K17 H4 K1] ├── [TO K0 K17 H0 K1] ├── [TO K0 K16 K1M108 K1] ├── [TO K0 K17 H2 K1] └── [TO K0 K17 H0 K1] </pre>	<p>a</p> <p>b</p> <p>c</p> <p>d</p> <p>e</p>	<p>a- Nạp dữ liệu từ D20 vào 16 bit trung gian: M15 ÷ M0</p> <p>b- Ghi 8 bit thấp của dữ liệu vào FX2N-2DA</p> <p>c- Lưu 8 bit thấp của dữ liệu</p> <p>d- ghi 4 bit cao của dữ liệu vào FX2N-2DA</p> <p>e- Đưa điện áp điều khiển ra kênh CH1</p>
---	--	---

	<p>Đoạn chương trình đọc xung từ encoder và tính toán tốc độ vòng quay</p>
	<p>Điều khiển chạy thuận, ngược và dừng biến tần</p>

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không hiển thị hoặc hiển thị không đúng tốc độ lên HMI	Do tính toán không đúng Do đặt thanh ghi trên HMI không đúng với trong lập trình	Kiểm tra chương trình lập trình theo công thức mục II.2.2

IV. TIÊU CHÍ ĐÁNH GIÁ

- Thao tác cài đặt biến tần []
- Sơ đồ, và thao tác đấu nối thiết bị []
- Thiết kế giao diện điều khiển []
- Chương trình lập trình đo và hiển thị tốc độ động cơ []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi sử dụng bộ đếm []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Phương pháp đo và tính toán tốc độ động cơ
3. Sơ đồ kết nối PLC - FX2N-2DA - Biến tần
4. Thiết kế giao diện và chương trình điều khiển
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

1. Thiết kế giao diện và lập chương trình điều khiển để nhập vào HMI giá trị là tốc độ quay (vòng/phút).

Bài thực hành số 16

LẬP TRÌNH TRUYỀN THÔNG MODBUS RTU GIỮA PLC FX3U VỚI BIẾN TẦN MITSUBISHI FR-E700 SỬ DỤNG MODULE FX3U-485BD

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Trình bày được các khái niệm cơ bản về truyền thông modbus;
- Phân biệt được các kiểu truyền thông modbus cơ bản.

Về kỹ năng:

- Đấu nối được phần cứng theo sơ đồ nguyên lý;
- Cài đặt được các tham số cho biến tần hoạt động ở chế độ truyền thông;
- Thiết kế được giao diện HMI và lập được chương trình truyền thông được với biến tần;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình thực hành.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình đấu nối thiết bị và lập trình điều khiển;
- Có khả năng tự đấu nối, lập trình hoặc làm việc theo nhóm;
- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.
- Hướng dẫn được các thành viên trong nhóm thực hành.

II. TÓM TẮT LÝ THUYẾT

1. Khái niệm về truyền thông Modbus

- Có 3 loại:
 - + Modbus RTU
 - + Modbus TCP (còn gọi là IP, hay Modbus Ethernet hay TCP/IP)
 - + Modbus ASC II
- Modbus RTU là giao thức mở, sử dụng đường truyền vật lý RS232 hoặc RS485 và mô hình dạng Master Slaver

2. So sánh truyền thông RS232 và RS485

RS 232	RS485
<ul style="list-style-type: none"> - Nguyên tắc: So sánh điện áp Tx và Rx với GND (3 dây) 	<ul style="list-style-type: none"> So sánh điện áp giữa 2 dây A và B: $\Delta U = V_A - V_B$ (cân 2 dây)

TX _____
 RX _____
 GND _____

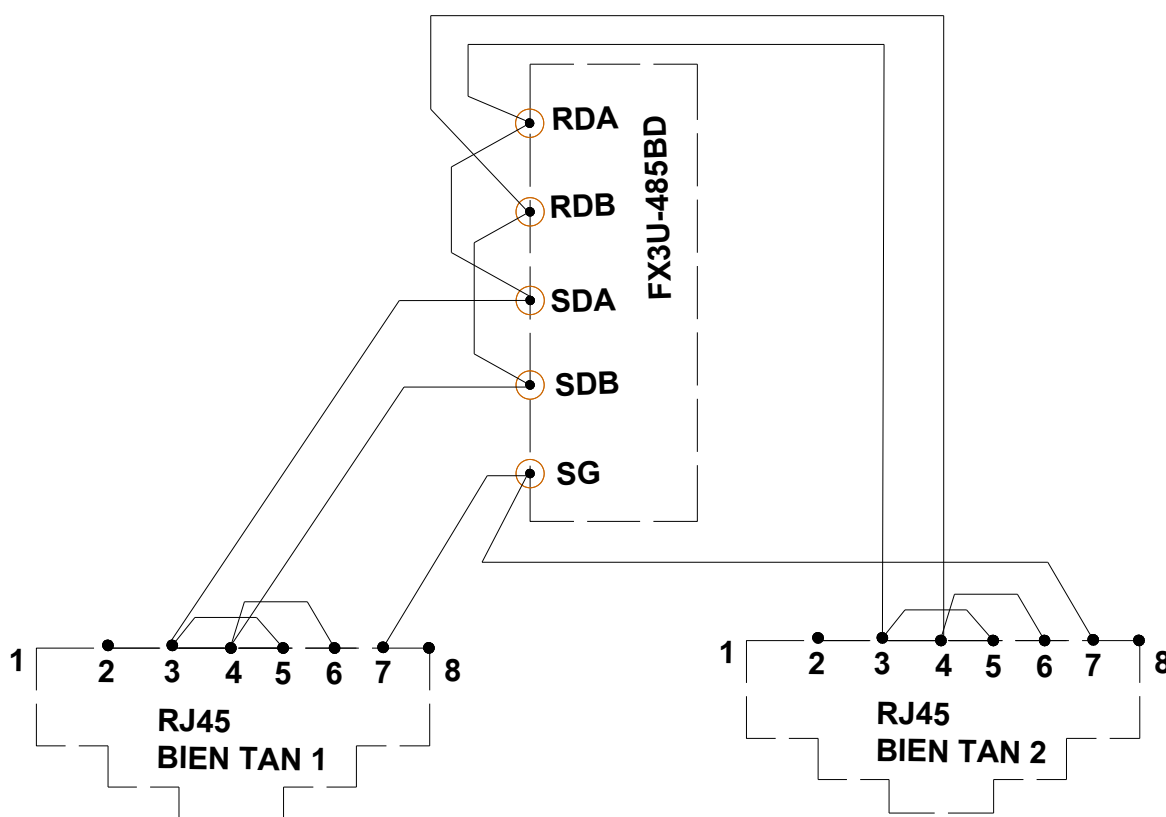
- Bị suy hao khi truyền xa, khoảng cách tối đa 15m
- Kiểu truyền nối tiếp
- Kết nối qua cổng COM

A _____
 B _____

- Ít bị suy hao khi truyền xa, khoảng cách truyền tối đa 1200m
- Kiểu truyền nối tiếp
- R485 được coi là phát triển của RS232

3. Sơ đồ kết nối FX3u - 485BD với biến tần

- Module FX3u - 485BD được ghép nối với PLC qua giắc cắm bên trái PLC.
- Trên biến tần E700 sử dụng cổng kết nối RJ45 (giống với cổng Ethernet).
- Sử dụng cáp mạng Lan để kết nối Module FX3u - 485BD với biến tần theo sơ đồ kết nối như sau:



Hình 2.16-1. Sơ đồ kết nối cáp truyền thông

4. Các thanh ghi chức năng của biến tần

Thanh ghi	Ý nghĩa
HOED	Thanh ghi nhập tần số chạy của biến tần Tần số $f = (HOED / 100)$
HOFA	Thanh ghi nhập giá trị điều khiển chạy thuận,

	ngược, dừng (8 bit): <table border="1" style="margin-left: 20px;"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> b1 = 1 → chạy thuận b2 = 1 → chạy ngược	b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0		
H7A	Thanh ghi trạng thái (8 bit): <table border="1" style="margin-left: 20px;"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> b1 = 1: đang chạy thuận b2 = 1: đang chạy ngược b7 = 1: báo lỗi	b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0		
H6F	Lưu giá trị tần số hiện tại								
H70	Lưu giá trị dòng điện hiện tại								

III. NỘI DUNG THỰC HÀNH

1. Công tác chuẩn bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

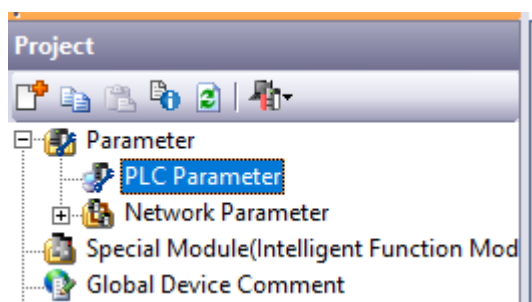
STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành PLC Mitsubishi dòng FX3u gồm: Module PLC, module truyền thông FX3u-485BD, module biến tần, PLC, động cơ, HMI	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Trình tự thực hiện

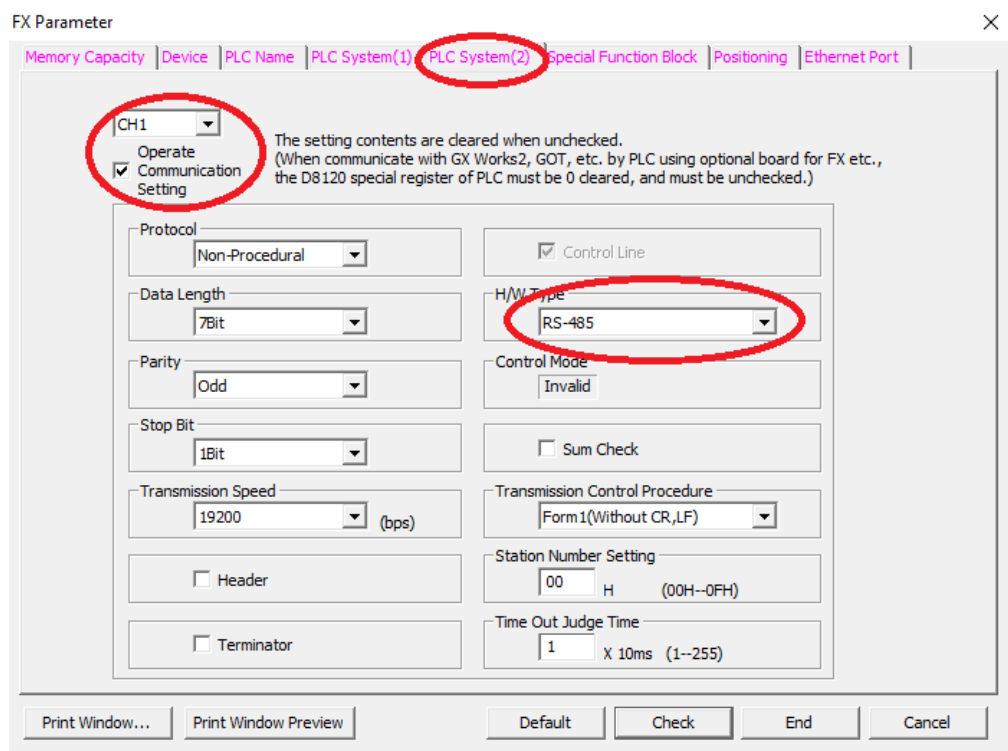
2.1. Cài đặt tham số

2.1.1. Cài đặt tham số PLC

- Khởi động Gx-Works 2, tại cửa sổ project vào parameter → PLC parameter



- Tại cửa sổ parameter chọn System 2 và cài đặt các tham số như hình dưới:



Hình 2.16-2. Cài đặt parameter cho PLC

2.1.2. Cài đặt tham số biến tần

Tham số	Ý nghĩa	Giá trị
P.117	Địa chỉ biến tần	1
P.118	Tốc độ truyền	19200
P.119	Stop bit	10
P.120	Parity	2
P.78	Cho phép quay 2 chiều	0
P. 79	Chọn chế độ hoạt động	0
P. 549	Chế độ hoạt động	0
P. 340	Chế độ truyền thông	10
P. 121, 122, 123		9999
P. 124		1

- Nếu có 2 biến tần trở lên thì cài đặt tương tự, chỉ khác địa trạm

2.2. Lập chương trình điều khiển

<p>M8000</p> <pre> [IVDR K1 H0ED D0 K1] [IVDR K1 H0FA D2 K1] [IVCK K1 H7A D4 K1] [IVCK K1 H6F D6 K1] [DIV D6 K100 D8] [IVCK K1 H70 D10 K1] [FLT D10 D12] [DEDIV D10 K100 D14] [END] </pre>	<p>Nhập tần số D0 vào thanh ghi H0ED trên biến tần 1</p> <p>Nạp D2 vào H0FA để ĐK chạy thuận, ngược, dừng</p> <p>Đọc trạng thái biến tần ra thanh ghi D4</p> <p>Đọc tần số ra thanh ghi D6</p> <p>Đọc giá trị dòng điện ra thanh ghi D10</p>
---	--

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không truyền thông được	<ul style="list-style-type: none"> - Cáp kết nối không đúng - Cài đặt tham số - Câu lệnh lập trình 	<ul style="list-style-type: none"> - Kiểm tra lại cáp kết nối theo đúng sơ đồ - Kiểm tra lại tham số cài đặt cho PLC và biến tần - Kiểm tra câu lệnh lập trình: địa chỉ trạm biến tần, các thanh ghi chức năng

IV. TIÊU CHÍ ĐÁNH GIÁ

- Thao tác cài đặt biến tần []
- Sơ đồ, và thao tác đấu nối thiết bị []
- Chương trình lập trình truyền thông với biến tần []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi sử dụng bộ đếm []

V. BÁO CÁO THỰC HÀNH

1. Tên bài

2. Các thanh ghi chức năng đặc biệt của biến tần
3. Sơ đồ kết nối cáp truyền thông
4. Chương trình điều khiển
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

1. Thiết kế giao diện điều khiển và giám sát các tham số truyền thông trên HMI.

Bài thực hành số 17

ĐIỀU KHIỂN ĐỘNG CƠ SERVO MITSUBISHI**I. MỤC TIÊU**

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Nhận biết được ký mã hiệu của các loại driver và động cơ servo mitsubishi;

- Trình bày được trình tự các bước cài đặt tham số driver.

Về kỹ năng:

- Đấu nối được phần cứng theo sơ đồ nguyên lý;

- Cài đặt được các tham số cơ bản cho driver servo;

- Điều khiển được động cơ servo chạy ở chế độ Jog và bằng encoder;

- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình thực hành.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình thực hành;

- Có khả năng làm việc độc lập và làm việc nhóm;

- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

II. TÓM TẮT LÝ THUYẾT**1. Giới thiệu driver dòng MR-J3-xA và động cơ servo HF-KP**

- Driver dòng MR - J3-xA là các driver chạy xung:

MR-J3-10A: driver công suất 100W, nguồn cấp 220V 1 pha hoặc 3 pha

MR-J3-20A: driver công suất 200W, nguồn cấp 220V 1 pha hoặc 3 pha

MR-J3-40A: driver công suất 400W, nguồn cấp 220V 1 pha hoặc 3 pha

- Dòng driver này thường được sử dụng cùng với động cơ loại HF-KPx có công suất tương ứng:

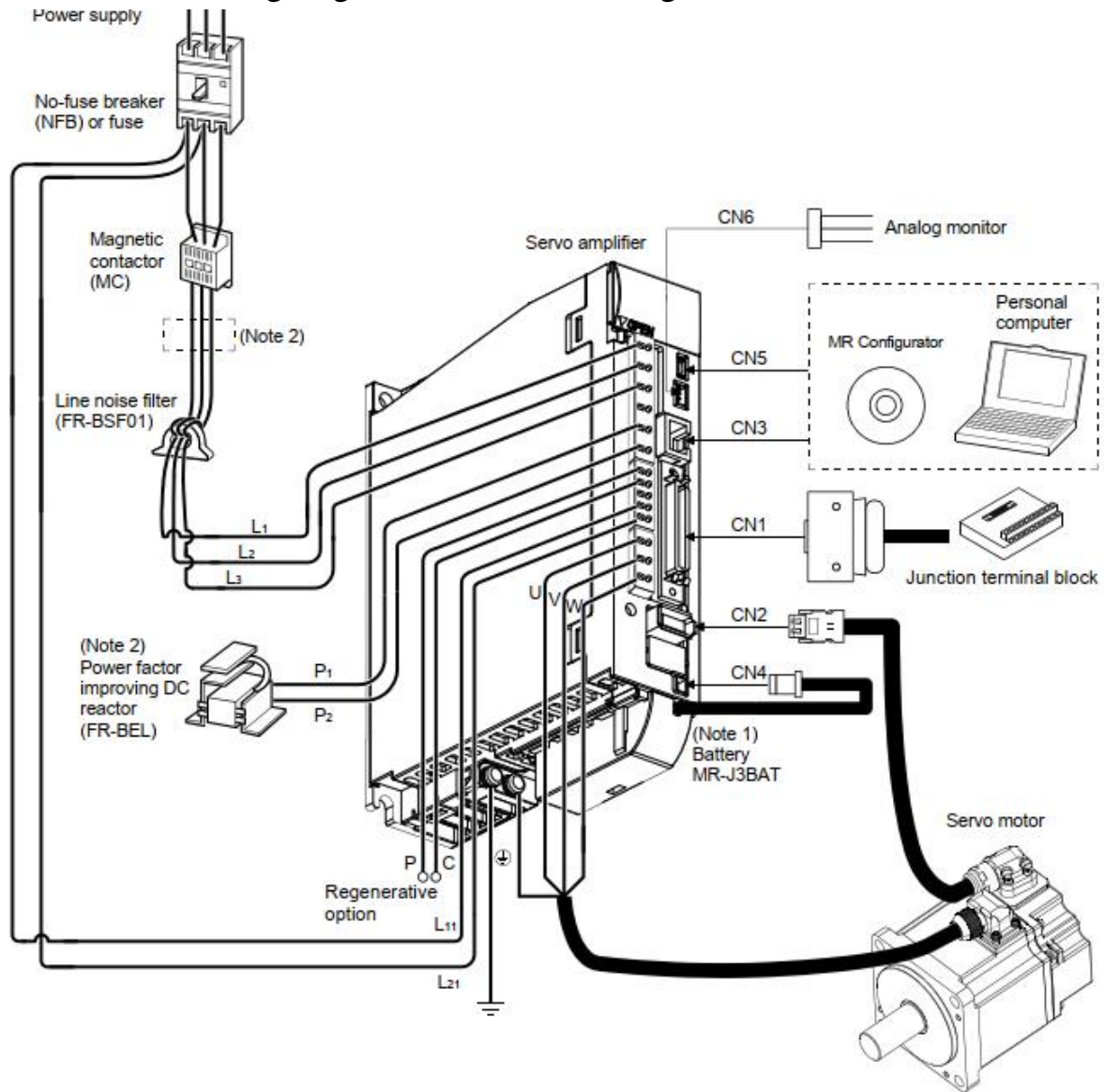
Driver	Động cơ	Xung điều khiển / Vòng
MR-J3-10A	HF-KP13	262144 pulse/rev
MR-J3-20A	HF-KP23	
MR-J3-40A	HF-KP43	

2. Sơ đồ nối dây driver**2.1. Sơ đồ đấu nối tổng quát**

- Giắc CN1: 50 pin, là giắc IN/OUT điều khiển (Xem chi tiết tại hình **Hình 2.17-2**)

- Giắc CN2: giắc kết nối với encoder của động cơ

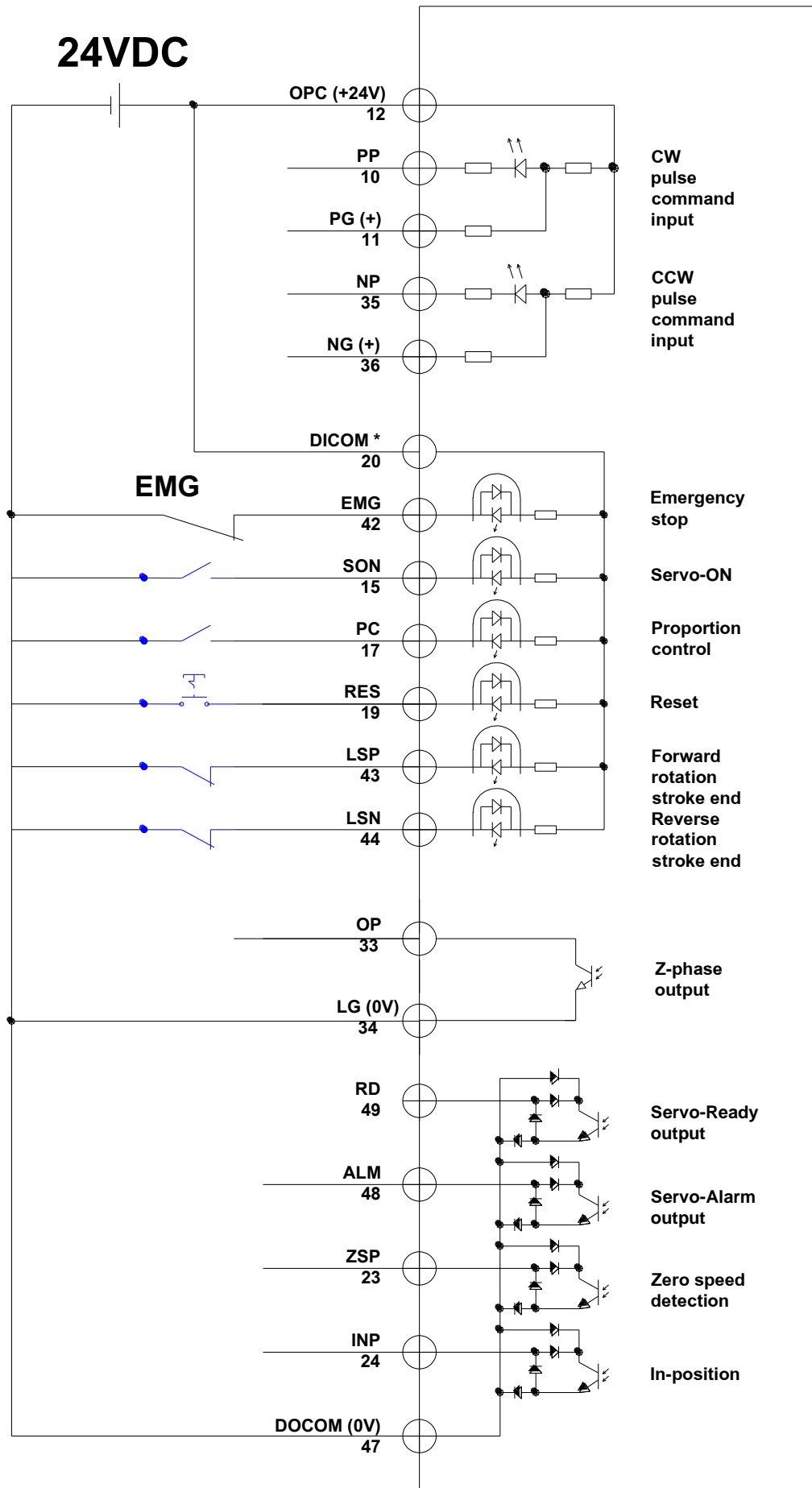
- Giắc CN3, CN5: giắc kết nối máy tính để cấu hình (nạp, rút parameter) của driver
- Giắc CN4: kết nối nguồn pin dự phòng, dùng trong chế độ điều khiển vị trí
- Giắc CN6: giắc giám sát tín hiệu Analog



Hình 2.17-1. Sơ đồ kết nối tổng quát

2.2. Sơ đồ đấu nối giắc CN1

- Đây là giắc 50 pin để cấp xung và các tín hiệu điều khiển
- Có 2 cách nối nguồn:
 - + Nối +24V vào chân 20 (kiểu SINK), tức là các chân 10, 11, 42, 45, 17, 19... đã chờ sẵn +24V, phù hợp với đầu vào là encoder dạng collector hở.



Hình 2.17-2. Sơ đồ chân giắc CNI

Chân chức năng	Ý nghĩa
20 – DICOM	+24VDC
OPC	+24VDC
46, 47	0V
10 – PP	Chân điều hướng
35 - NP	Chân điều hướng
42 – EMG	Dừng khẩn cấp
15 - SON	Servo ON
17 – PC	Chế độ điều khiển tỷ lệ
19 – RES	Reset
43 – LSP	Giới hạn hành trình chạy thuận
43 - LSN	Giới hạn hành trình chạy ngược
33 – OP	Xung Z từ encoder của động cơ
34 – LG	0V
48 – ALM	Cảnh báo

3. Cài đặt các tham cơ bản

3.1. Phương pháp cài đặt tham số

- Bước 1: Nhấn Mode 3 lần để xuất hiện PA01;
- Bước 2: Nhấn Δ (up) hoặc ∇ (down) để chọn PA tương ứng;
- Bước 3: Nhấn Set 2 lần để hiện thị và nhấp nháy giá trị của PA đó, nhấn set 1 lần thì không nhấp nháy:

- + Nếu giá trị cần nhập chỉ có 5 chữ số \rightarrow Nhấn (hoặc giữ) Δ hoặc ∇ để chọn đến giá trị cần nhập \rightarrow nhấn set để lưu \rightarrow nhấn Δ hoặc ∇ để thoát;
- + Nếu giá trị cần nhận có 6 chữ số trở lên, ví dụ 665536 thì nhập như sau:

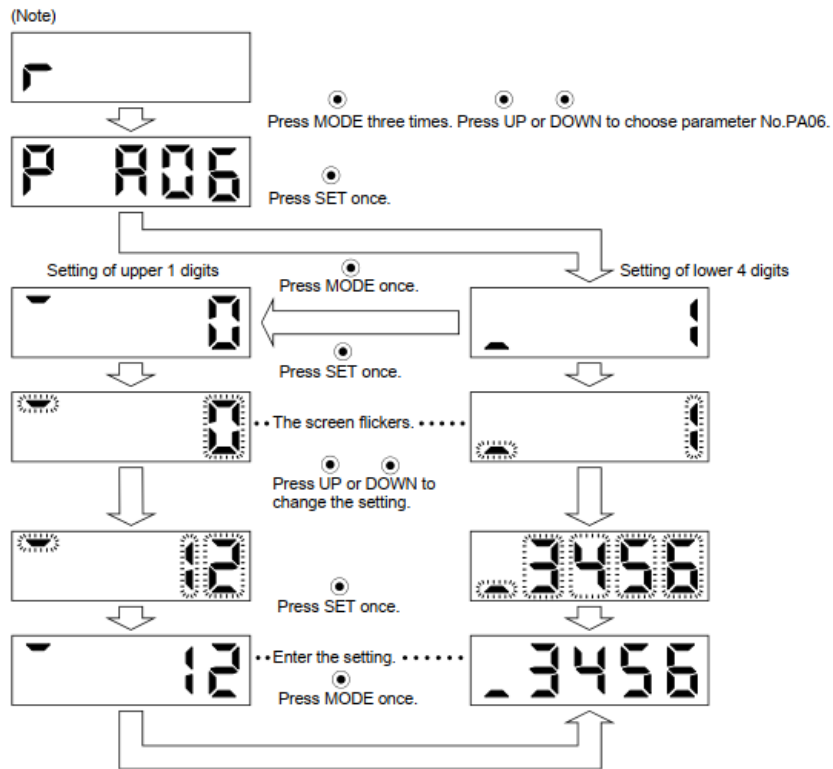
\rightarrow Nhập 4 số 5536 trước, với dấu gạch ngang ở dưới \rightarrow nhấn Set để lưu:

_5536

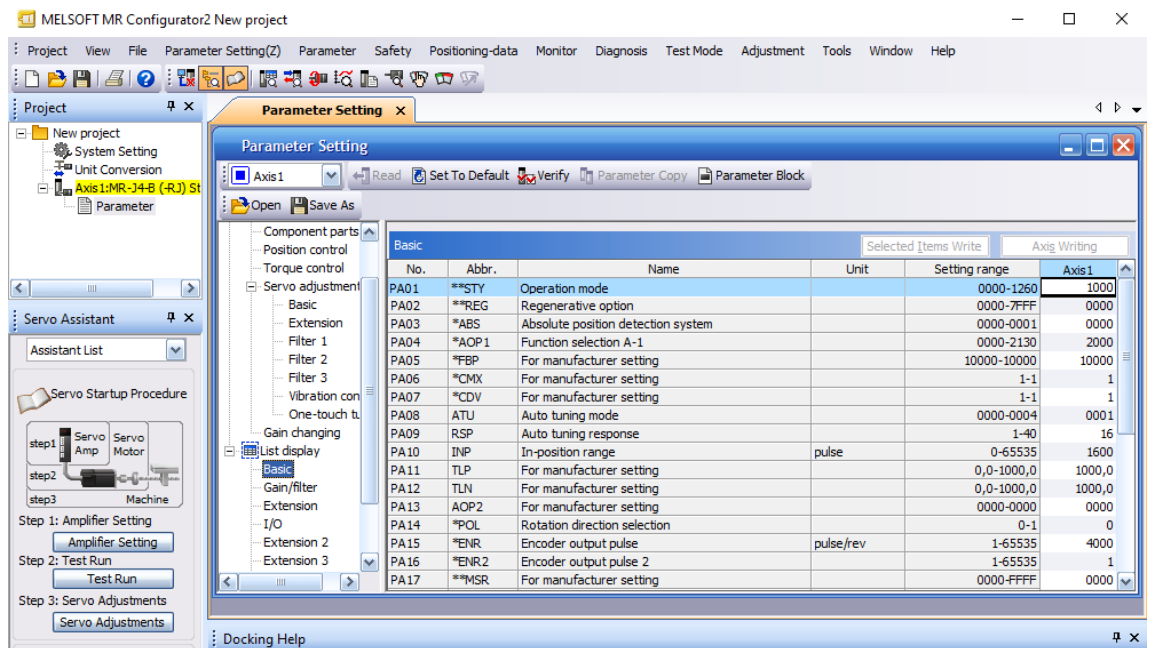
\rightarrow Nhấn MODE Nhập tiếp 2 số 66, với dấu gạch ngang ở trên \rightarrow nhấn Set để lưu

- 66

\rightarrow Nhấn Δ hoặc ∇ để thoát



Ngoài ra cũng có thể sử dụng phần mềm MR Configurator trên máy tính để đọc/ghi tham số cho driver, hoặc thực hiện các chế độ chạy JOG từ máy tính



Hình 2.17-3. Giao diện phần mềm MR configurator

3.2. Ý nghĩa các tham số cài đặt cơ bản

Tham số	Ý nghĩa
PA01	Cài đặt chế độ điều khiển <div style="display: flex; align-items: center; gap: 10px;"> <div style="border: 1px solid black; padding: 2px 5px;">0</div> <div style="border: 1px solid black; padding: 2px 5px;">0</div> <div style="border: 1px solid black; padding: 2px 5px;">0</div> </div> <p style="margin-left: 40px;">0: Điều khiển vị trí 1: Điều khiển vị trí và tốc độ</p>

	<p>2: Điều khiển tốc độ 3: điều khiển tốc độ và mô men</p>																						
PA05	<p>Số xung/vòng quay - 0: cần nhập $PA06 = CMX$ $PA07 = CDV$ - Giá trị khác: giá trị nhỏ nhất là 1000, là số xung điều khiển động cơ quay hết 1 vòng + Khi PA05 khác 0 thì PA06, PA07 bị vô hiệu</p>																						
PA06	Giá trị CMX: tham số của động cơ (tra thông số động cơ)																						
PA07	Giá trị CDV: số xung cần phát để động cơ quay 1 vòng																						
PA08	Auto turning mode, giá trị khởi tạo là 0001H																						
PA09	Momen hãm																						
PA13	<p>Chế độ lựa chọn xung đầu vào và phương thức hoạt động của motor:</p> <table border="1"> <thead> <tr> <th>Setting</th> <th>Pulse train form</th> <th>Forward rotation command</th> <th>Reverse rotation command</th> </tr> </thead> <tbody> <tr> <td>0010h</td> <td>Forward rotation pulse train Reverse rotation pulse train</td> <td> </td> </tr> <tr> <td>0011h</td> <td>Signed pulse train</td> <td> </td> </tr> <tr> <td>0012h</td> <td>A-phase pulse train B-phase pulse train</td> <td> </td> </tr> <tr> <td>0000h</td> <td>Forward rotation pulse train Reverse rotation pulse train</td> <td> </td> </tr> <tr> <td>0001h</td> <td>Signed pulse train</td> <td> </td> </tr> <tr> <td>0002h</td> <td>A-phase pulse train B-phase pulse train</td> <td> </td> </tr> </tbody> </table> <p>PA13=10: - Xung điều khiển là xung đơn, nối với chân 10 → động cơ chạy thuận (chân 35 phải ở mức cao)</p>	Setting	Pulse train form	Forward rotation command	Reverse rotation command	0010h	Forward rotation pulse train Reverse rotation pulse train		0011h	Signed pulse train		0012h	A-phase pulse train B-phase pulse train		0000h	Forward rotation pulse train Reverse rotation pulse train		0001h	Signed pulse train		0002h	A-phase pulse train B-phase pulse train	
Setting	Pulse train form	Forward rotation command	Reverse rotation command																				
0010h	Forward rotation pulse train Reverse rotation pulse train																						
0011h	Signed pulse train																						
0012h	A-phase pulse train B-phase pulse train																						
0000h	Forward rotation pulse train Reverse rotation pulse train																						
0001h	Signed pulse train																						
0002h	A-phase pulse train B-phase pulse train																						

	<p>- Xung điều khiển nối chân 35, chân 10 ở mức cao → động cơ chạy ngược</p> <p>PA13=11: có 2 trường hợp</p> <p>1. Nếu xung A vào chân 10:</p> <ul style="list-style-type: none"> - Chân 35 nối mức cao → động cơ chạy thuận - Chân 35 nối mức thấp → động cơ chạy ngược <p>→ chân 35 là chân điều hướng</p> <p>2. Nếu xung A vào chân 10, xung B vào chân 35 (xung A, B lệch nhau 90 độ)</p> <ul style="list-style-type: none"> - Encoder quay thuận → động cơ chạy thuận - Encoder quay ngược → động cơ chạy ngược <p>PA13=12:</p> <p>- Nối lần lượt xung A, B vào chân 10 và 35, thì động cơ quay tương tự như trường hợp 2 của PA13=11, nhưng tỷ số tăng gấp đôi do động cơ được dịch chuyển ở cả 2 sườn xung <i>(nếu chỉ đấu 1 dây vào chân 10 hoặc chân 35 thì động cơ không chạy)</i></p>																												
PA14	Lựa chọn hướng quay																												
PA19	<p>Khóa/cho phép ghi các tham số (parameter):</p> <p>000: chỉ cho phép cài đặt các tham số cơ bản PA_ 00A: Khóa hết tất cả các tham số 00C: cho mở PA, PB, PC, PD 00F: mở tất cả, bao gồm cả PE, PO</p>																												
PD01	<p>Cho phép các tín hiệu vào, tự động bảo vệ động cơ:</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th rowspan="2">Signal name</th> <th colspan="2">Initial value</th> </tr> <tr> <th>BIN</th> <th>HEX</th> </tr> </thead> <tbody> <tr> <td>Servo-on (SON)</td> <td>0</td> <td rowspan="4">0</td> </tr> <tr> <td></td> <td>0</td> </tr> <tr> <td></td> <td>0</td> </tr> <tr> <td></td> <td>0</td> </tr> <tr> <td>Proportion control (PC)</td> <td>0</td> <td rowspan="4">0</td> </tr> <tr> <td>External torque limit selection (TL)</td> <td>0</td> </tr> <tr> <td></td> <td>0</td> </tr> <tr> <td></td> <td>0</td> </tr> <tr> <td>Forward rotation stroke end (LSP)</td> <td>0</td> <td rowspan="2">0</td> </tr> <tr> <td>Reverse rotation stroke end (LSN)</td> <td>0</td> </tr> </tbody> </table> <p style="margin-left: 20px;">BIN 0: Used as external input signal BIN 1: Automatic ON</p>	Signal name	Initial value		BIN	HEX	Servo-on (SON)	0	0		0		0		0	Proportion control (PC)	0	0	External torque limit selection (TL)	0		0		0	Forward rotation stroke end (LSP)	0	0	Reverse rotation stroke end (LSN)	0
Signal name	Initial value																												
	BIN	HEX																											
Servo-on (SON)	0	0																											
	0																												
	0																												
	0																												
Proportion control (PC)	0	0																											
External torque limit selection (TL)	0																												
	0																												
	0																												
Forward rotation stroke end (LSP)	0	0																											
Reverse rotation stroke end (LSN)	0																												

0	0	0	0	b11	b10					b5	b4		b2		
---	---	---	---	-----	-----	--	--	--	--	----	----	--	----	--	--

b2 = 0 → Servo ON
 = 1 → Servo OFF
b4 = 0/1 → Proportion control ON/OFF
b5: giới hạn momen từ bên ngoài
b10 (LSP):
 = 0 → tự động dừng khi quá hành trình tiến (sử dụng giới hạn hành trình bên ngoài)
 = 1 → không tự động dừng khi quá hành trình tiến (phải dừng bằng cách lập trình từ PLC)
b11 (LSN):
 = 0 → tự động dừng khi quá hành trình lùi (sử dụng giới hạn hành trình bên ngoài)
 = 1 → không tự động dừng khi quá hành trình lùi (phải dừng bằng cách lập trình từ PLC)
Ví dụ:
PD01 = 0C00:
 - Động cơ sẽ không tự động dừng khi quá hành trình theo cả 2 chiều
 - Bật tắt động cơ bằng công tắc nối với chân SON
PD01 = 0004:
 - Không cần có công tắc Servo On, cấp xung là chạy
 - Tự động dừng khi quá hành trình theo cả 2 chiều

*** Điều kiện cần để motor chạy:**

- Chân EMG phải được đóng, nếu mở sẽ báo lỗi E6
- LSP và LSN phải được đóng (nối với COM – 0V ở chế độ SINK)

III. NỘI DUNG THỰC HÀNH

1. Công tác chuẩn bị

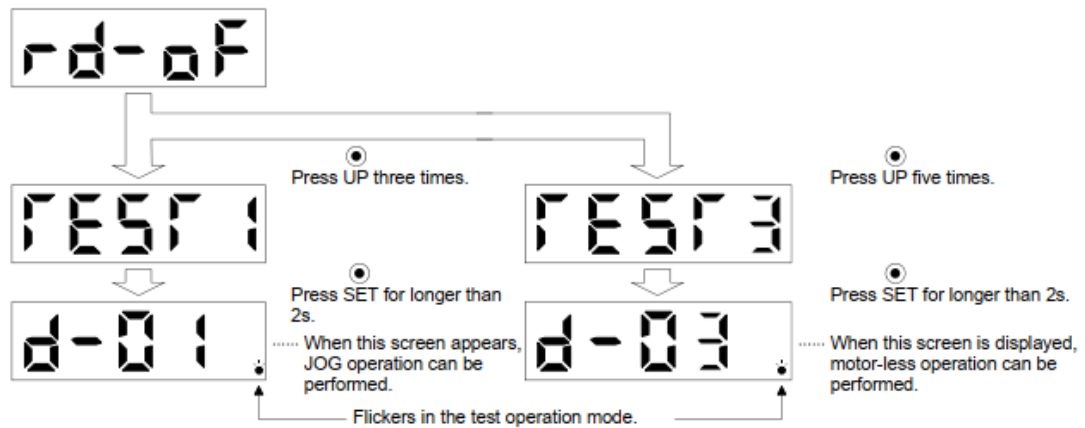
STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành động cơ servo Mitsubishi	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Trình tự thực hiện

2.1. Thực hiện chạy JOG

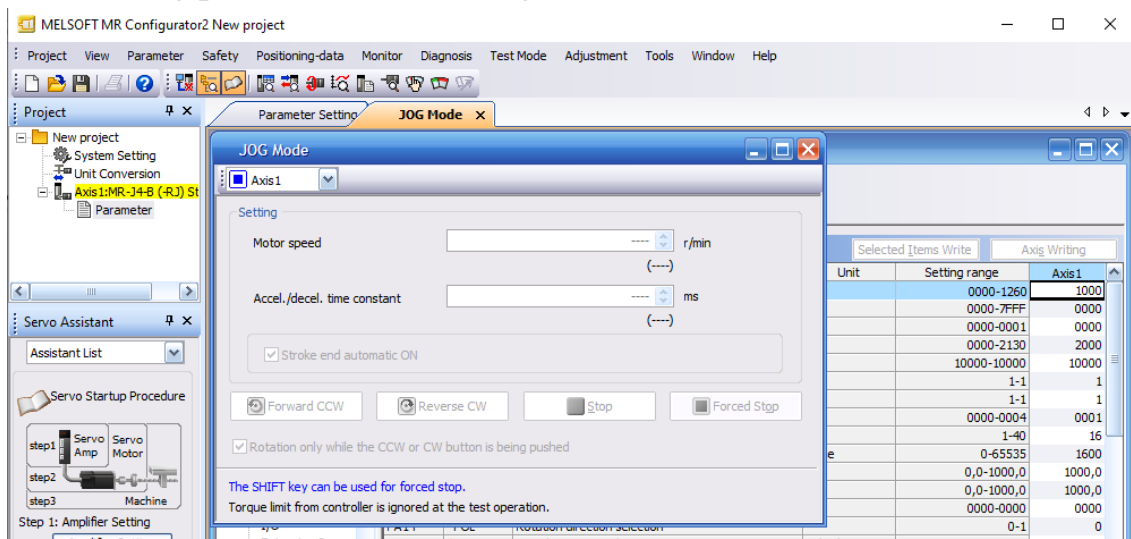
- Sử dụng phím bấm trên driver:
- + Đóng các tiếp điểm EMG, LSP, LSN, SON
- + Sử dụng phím MODE để chọn đến chế độ chạy JOG
- + Nhấn và giữ phím up/down để chạy động cơ thuận/ngược
- + Để thoát chế độ chạy JOG cơ thể ngắt nguồn của driver hoặc nhấn MODE

để chuyển sang màn hình d-01, nhấn giữ SET trong 2 giây.



Hình 2.17-3. Các bước chạy JOG

- Sử dụng phần mềm MR Configurator:



Hình 2.17-4. Chạy JOG trên phần mềm

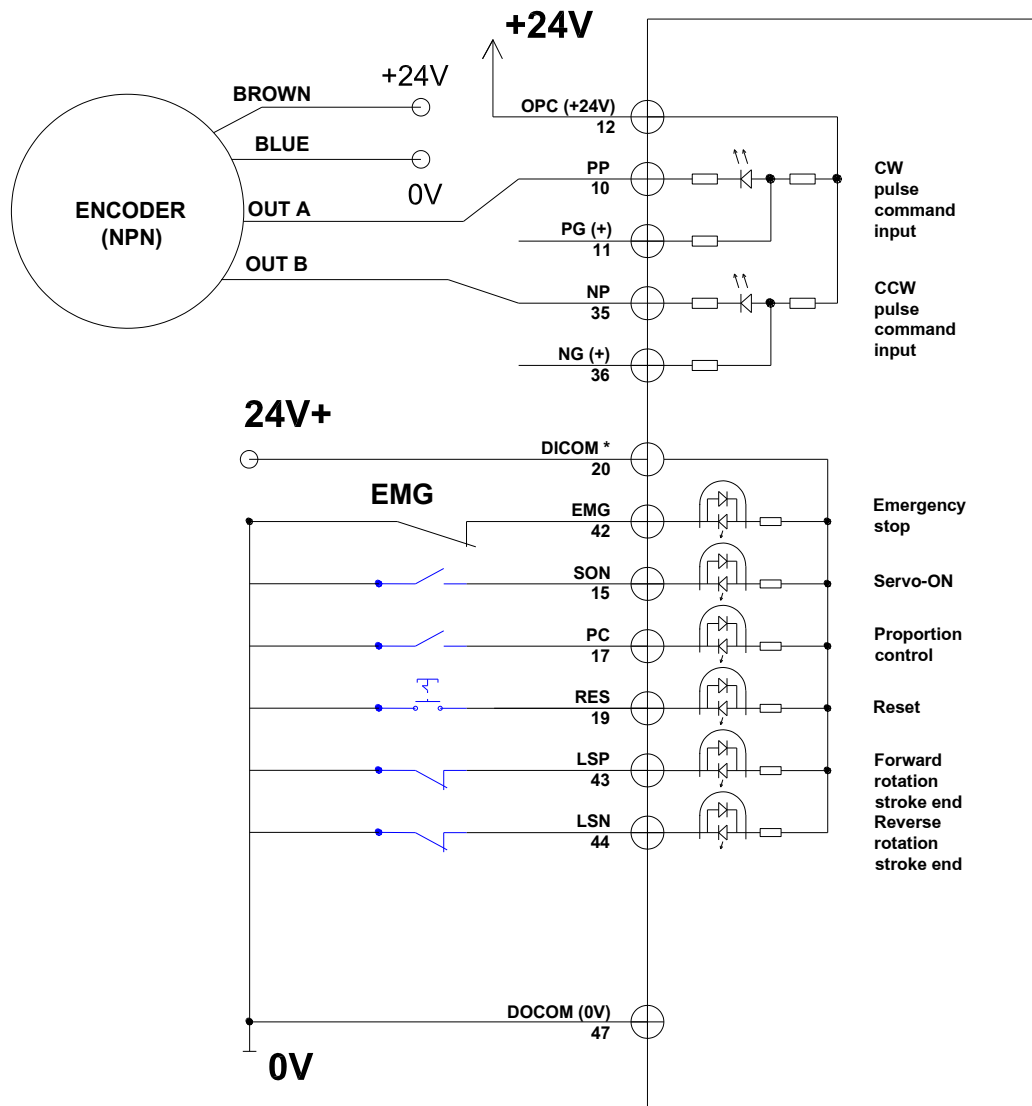
2.2. Điều khiển động cơ servo bằng tay quay phát xung

2.2.1. Sơ đồ kết nối

- Sử dụng tay quay phát xung E6B2-CWZ6C-200P/R



Hình 2.17-5. Tay quay phát xung



Hình 2.17-6. Sơ đồ nối tay quay phát xung với driver

2.2.2. Cài đặt tham số

Giả sử ta muốn phát 200 xung thì động cơ quay được 1 vòng (360°), cần tính toán các tham số như sau:

- Tính tỉ số: $\frac{CMX}{CDV}$

CMX : là tham số của động cơ (tra bảng)

CDV : số xung cần phát để động cơ quay 1 vòng

+ Với động cơ loại HK-KP13 (23, 43), tra tài liệu được $CMX = 262144$ p/r

+ $CDV = 200$ (theo yêu cầu đầu bài)

$$\frac{CMX}{CDV} = \frac{262144}{200} = \frac{65536}{50}$$

+ Cài đặt: PA05 = 0
PA06 = 65536
PA07 = 50

- Cài đặt: PD01 = 0004

- Cài đặt: PA13 = 11

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không xuất hiện một số tham số	Chưa đặt đúng PA19	Đặt lại PA19, cho phép mở các tham số
Động cơ hoạt động không đúng theo yêu cầu	Tính toán $\frac{CMX}{CDV}$ khụng đỳng	Tính toán và cài đặt lại PA05, PA06, PA07

IV. TIÊU CHÍ ĐÁNH GIÁ

- Sơ đồ, và thao tác đấu nối thiết bị []
- Tính toán các tham số []
- Thao tác cài đặt tham số driver []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi điều khiển động cơ servo []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Các tham số cài đặt và ý nghĩa
3. Sơ đồ kết nối cáp truyền thông
4. Trình tự cài đặt các tham số
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

Cài đặt cho driver để khi tay quay phát xung quay hết 4 vòng thì động cơ quay được 1 vòng?

Bài thực hành số 18

LẬP TRÌNH ĐIỀU KHIỂN VỊ TRÍ SỬ DỤNG PLC MITSUBISHI VÀ SERVO MR-J3-10A

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Phân biệt được các cơ cấu truyền động;
- Nhận biết được các lệnh sử dụng để điều khiển vị trí;
- Trình bày được trình tự các bước cài đặt tham số driver.

Về kỹ năng:

- Đấu nối được phần cứng theo sơ đồ nguyên lý;
- Cài đặt được các tham số cơ bản cho driver servo;
- Lập được chương trình PLC điều khiển vị trí động cơ servo;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình

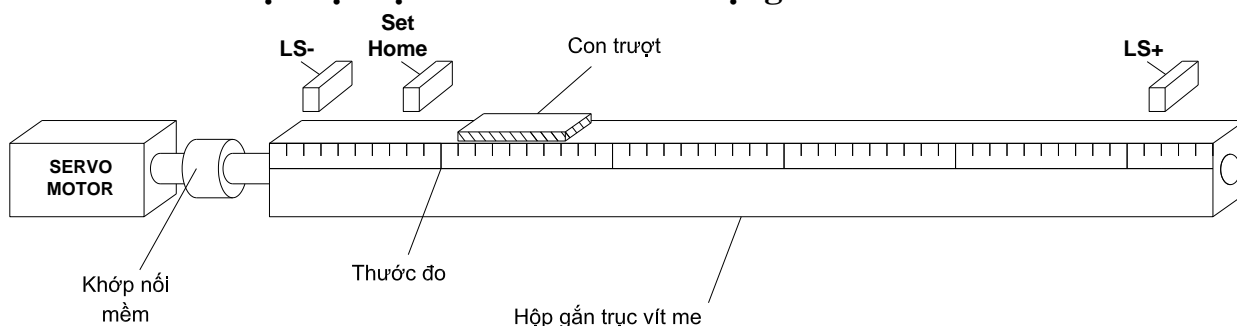
thực hành.

Về năng lực tự chủ và trách nhiệm:

- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình thực hành;
- Có khả năng làm việc độc lập và làm việc nhóm;
- Hướng dẫn được các thành viên khác trong nhóm cài đặt và lập trình điều khiển động cơ servo;
- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

II. TÓM TẮT LÝ THUYẾT

1. Giới thiệu bộ thực hành điều khiển động cơ servo



Hình 2-18.1. Cơ cấu truyền động động cơ servo – trục vít me – con trượt

- Khi động cơ quay được 1 vòng, trục vít me cũng quay theo 1 vòng, khi đó con trượt được gắn với đai ốc sẽ chuyển động tịnh tiến (trượt dọc theo thanh ray) 1 khoảng là S (mm):

- Thông số trục vít me:
- + Có 1 đầu mỗi ren (1 đường ren)
- + Bước vít me: $S = 5\text{mm}$

+ Hành trình $L_{\max} = 500\text{mm}$

- Nếu trục vít me có n đường ren, thì khi trục vít quay được 1 vòng sẽ tạo ra 1 bước xoắn:

$$m = n.S \text{ (mm)}$$

- Bộ 03 cảm biến dạng collector hở:

LS- (NC): Cảm biến giới hạn hành trình lùi Over -

LS+ (NC): Cảm biến giới hạn hành trình lùi Over +

Set home (NO): Cảm biến về gốc

2. Cách tính bước dịch chuyển trên 1 xung

- Độ phân giải: k (mm/xung) là khoảng dịch chuyển của con trượt trên 1 xung điều khiển.

- Số xung cần cài đặt để động cơ quay hết 1 vòng: x (xung)

- Khi phát 1 xung thì động cơ (trục vít) quay: $1/x$ (vòng), do hộp số có tỉ số truyền là 1, số ren là 1)

- Trục vít quay 1 vòng thì con trượt dịch chuyển 1 khoảng là S (mm), do đó khi phát 1 xung thì con trượt sẽ dịch chuyển 1 khoảng là S/x (mm)

Hay nói cách khác, ta có độ phân giải là:

$$k = \frac{S}{x} \text{ (mm/xung)}$$

3. Một số lệnh điều khiển vị trí

3.1. Lệnh phát xung PLSY / DPLSY

- PLSY là lệnh thực thi với 16 bit, DPLSY thực thi với 32 bit

- Cú pháp lệnh:

-[PLSY D0 D4 Y0]-

Trong đó:

D0: Tần số xung (xung/giây)

D4: Tổng số xung phát ra

Y0: cổng phát xung

- Chiều quay được xác định bởi tín hiệu ở Y4

- Thanh ghi D8410 (L-byte thấp) và D8411 (H-byte cao) lưu trữ số xung đã phát ở cổng Y0.

- Thanh ghi D8412 (L-byte thấp) và D8413 (H-byte cao) lưu trữ số xung đã phát ở cổng Y1.

3.2. Lệnh phát xung PLSR / DPLSR

- PLSR/DPLSR là các lệnh phát xung có thời gian tăng tốc, giảm tốc

- Cú pháp lệnh:

-[PLSY D0 D4 D8 Y0]-

Trong đó:

D0: Tần số xung (xung/giây)

D4: Tổng số xung phát ra

D8: Thời gian tăng tốc và thời gian giảm tốc (ms)

Y0: cổng phát xung

3.3. Lệnh di chuyển tương đối DRVI / DDRVI

- Cú pháp lệnh:

-[DRVI D0 D4 Y0 Y4]-

- Trong đó:

D0: Tổng số xung phát ra

D4: Tần số phát xung (Hz)

Y0: cổng phát xung (Fx3U phát được 4 cổng Y0, Y1, Y2, Y3)

Y4: Hướng: + Nếu $D0 > 0 \rightarrow Y4$ ON, động cơ quay thuận

+ Nếu $D0 < 0 \rightarrow Y4$ OFF, động cơ quay ngược

- Lệnh này sẽ thực hiện phát hết số xung thanh ghi D0, khi đó con trượt sẽ dịch chuyển 1 khoảng so với vị trí hiện tại.

- **D8341 & D8340** lưu giá trị tổng số xung đã phát (giá trị hiện hành) ra ở cổng Y0.

- Bit **M8340**: báo trạng thái của Y0, ON nếu vẫn còn xung, OFF nếu đã phát hết xung.

3.4. Lệnh di chuyển tuyệt đối DRVA / DDRVA

- Cú pháp lệnh:

-[DRVA D0 D4 Y0 Y4]-

- Trong đó:

D0: Tổng số xung phát ra so với vị trí ban đầu (gốc 0)

D4: Tần số phát xung (Hz)

Y0: cổng phát xung (Fx3U phát được 4 cổng Y0, Y1, Y2, Y3)

Y4: Hướng: + Nếu $D0 > 0 \rightarrow Y4$ ON, động cơ quay thuận

+ Nếu $D0 < 0 \rightarrow Y4$ OFF, động cơ quay ngược

- Lệnh này sẽ thực hiện phát hết số xung bằng D0 – D8340 (D8340 là giá trị hiện hành)

Ví dụ:

+ Lần 1 nhập $D0 = 1000$, lệnh thực hiện phát 1000 xung $\rightarrow D8340 = 1000$;

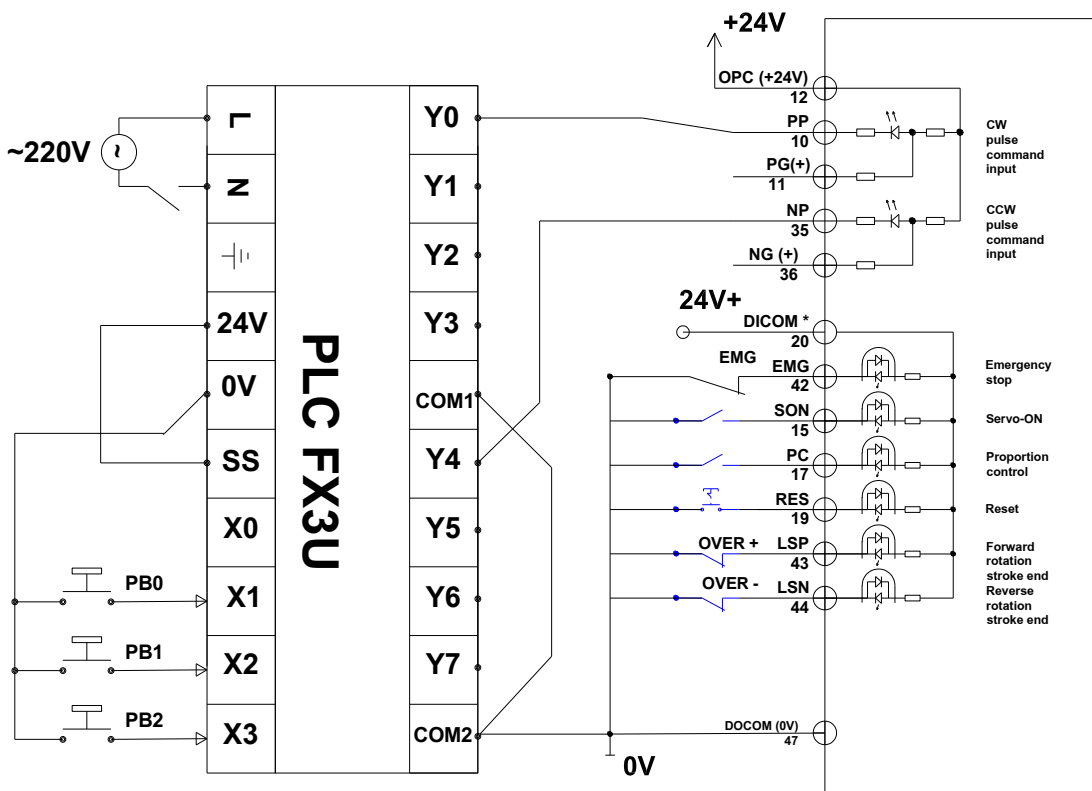
+ Lần 2, nếu $D0$ vẫn = 1000 \rightarrow con động cơ không chạy (vì $D0 - D8340 = 0$, không có xung nào được phát ra)

+ Lần 3, nhập $D0 = 1500 \rightarrow$ số xung phát ra là:

$D0 - d8340 = 1500 - 1000 = 500$ xung

→ động cơ chạy thêm 500 xung

4. Sơ đồ đấu nối tổng quát



Hình 2-18.2. Sơ đồ kết nối PLC – Driver servo

III. NỘI DUNG THỰC HÀNH

1. Công tác chuẩn bị

- Sử dụng cho 1 nhóm không quá 3 sinh viên

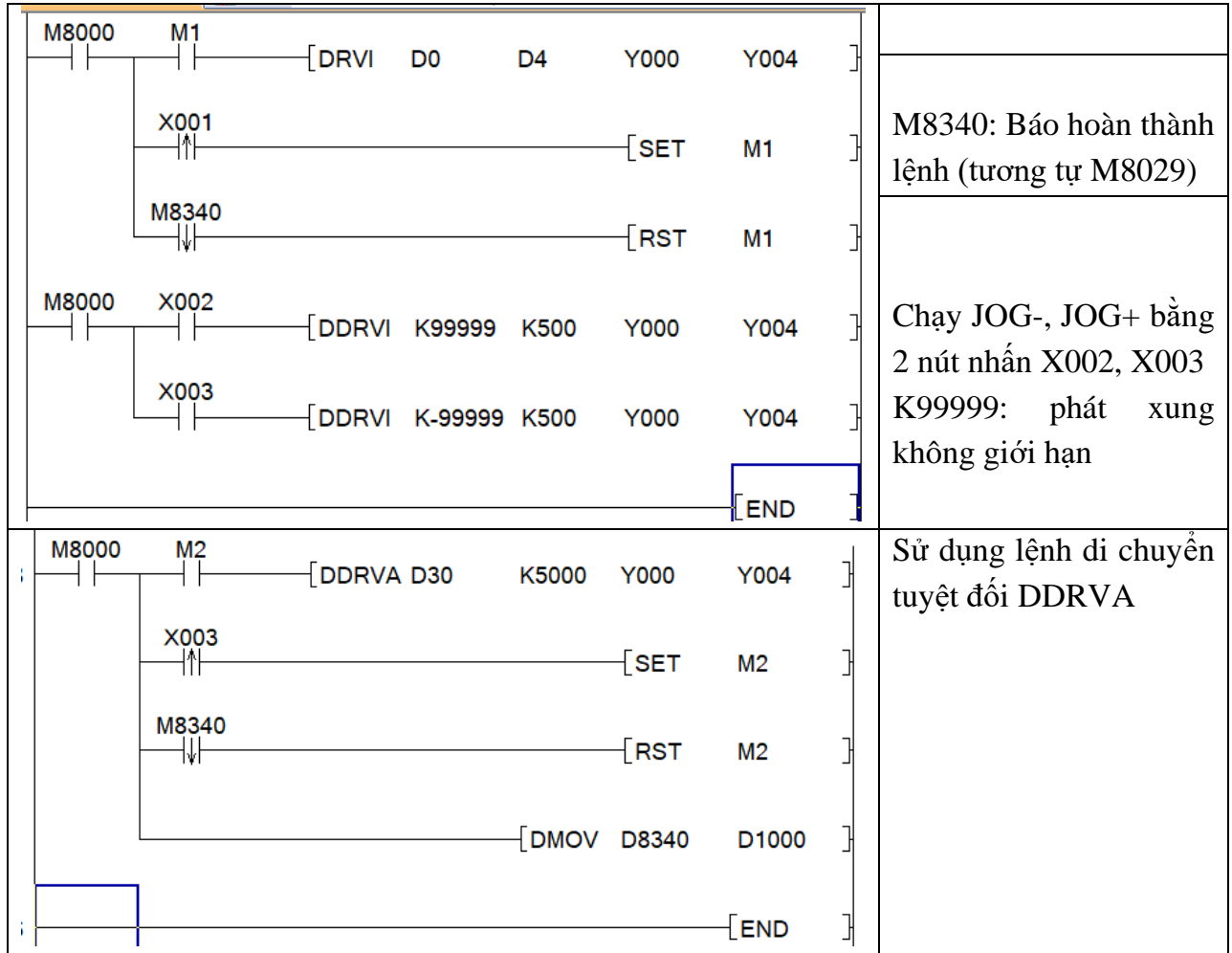
STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành động cơ servo Mitsubishi	1 bộ
2	Dây cắm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Trình tự thực hiện

2.1. Thực hành lệnh phát xung PLSY / DPLSY

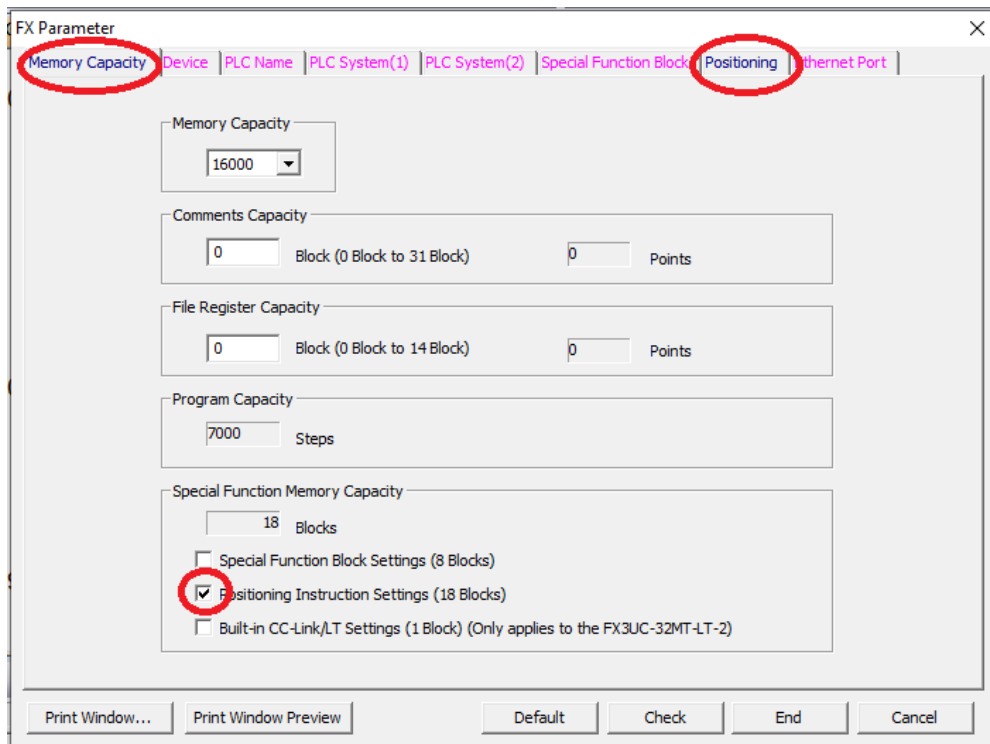
2.1.1. Cài đặt tham số:

PD01 = 0004	Set ON servo, động cơ chạy khi có xung; Sử dụng các giới hạn hành trình bên ngoài để bảo vệ
PA13 = 11	Chế độ xung điều khiển (bài thực hành số 17)
PA14 = 0/1	Đôi hướng quay 0: Y004 ON/OFF → động cơ quay thuận/ngược 1: Y004 ON/OFF → động cơ quay ngược/thuận

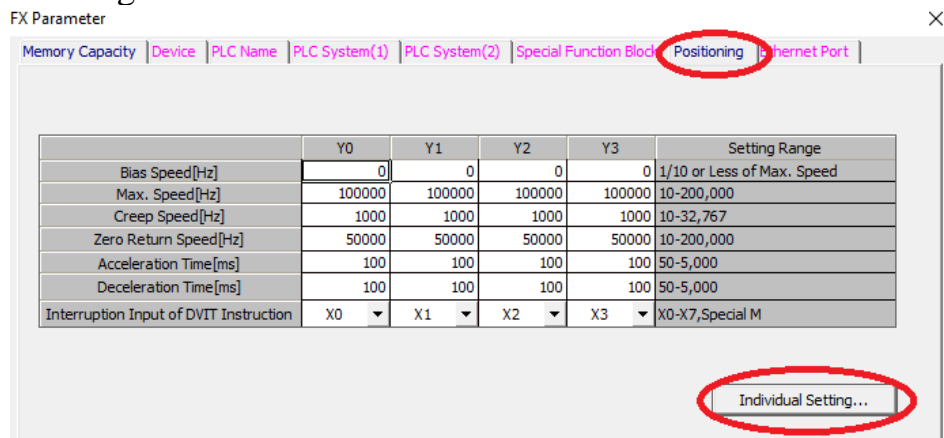


2.3. Thực hành lệnh TBL / DTBL

- **TBL / DTBL** là các lệnh gọi các lệnh đã được định nghĩa sẵn

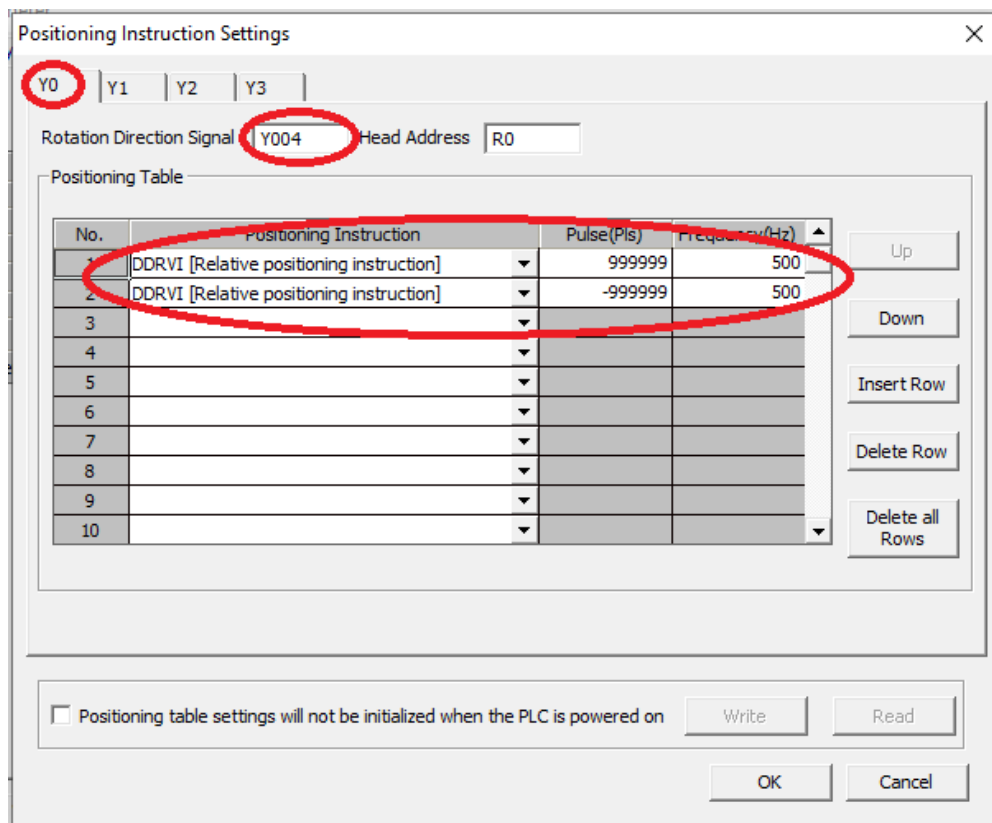


- Chọn PLC Parameter → tại cửa số Memory capacity tick chọn Possitioning Instruction Setting



- Chọn sang tab Positioning → chọn Individual Setting:

- Chọn cặp Y0 – Y4 (cặp ngõ ra phát xung – điều hướng) → nhập các lệnh DDRVI, DDRVA vào bảng → OK → Check → End



→ Chương trình sử dụng lệnh TBL / DTBL:

M8000	X002	[DDRVI K99999 K500 Y000 Y004]			
	X003	[DDRVI K-99999 K500 Y000 Y004]			
M8000	X002	[DTBL Y000 K1]			
	X003	[DTBL Y000 K2]			

2 dòng lệnh DTBL thay thế cho 2 dòng lệnh phía trên, K1, K2 là thứ tự lệnh đã định nghĩa trong bảng Positioning Instruction

3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không xuất hiện một số tham số	Chưa đặt đúng PA19	Đặt lại PA19, cho phép mở các tham số
Động cơ hoạt động không đúng theo yêu cầu	Tính toán $\frac{CMX}{CDV}$ khụng đỳng	Tính toán và cài đặt lại PA05, PA06, PA07

IV. TIÊU CHÍ ĐÁNH GIÁ

- Sơ đồ, và thao tác đấu nối thiết bị

[]

- Tính toán các tham số []
- Thao tác cài đặt tham số driver []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi điều khiển động cơ servo []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Các tham số cài đặt và ý nghĩa
3. Sơ đồ kết nối cáp truyền thông
4. Trình tự cài đặt các tham số
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

Nêu sự khác biệt giữa lệnh di chuyển tương đối và tuyệt đối?

Bài thực hành số 19

LẬP TRÌNH NGUYÊN ĐIỂM

SỬ DỤNG PLC MITSUBISHI VÀ SERVO MR-J3-10A

I. MỤC TIÊU

Học xong bài học này, người học có khả năng:

Về kiến thức:

- Trình bày được trình tự thực hiện nguyên điểm;
- Trình bày được trình tự các bước cài đặt tham số driver.

Về kỹ năng:

- Đầu nối được phần cứng theo sơ đồ nguyên lý;
- Cài đặt được các tham số cơ bản cho driver servo;
- Lập được chương trình PLC thực hiện nguyên điểm;
- Biên dịch, tìm lỗi và khắc phục được một số lỗi thường gặp trong quá trình

thực hành.

Về năng lực tự chủ và trách nhiệm:

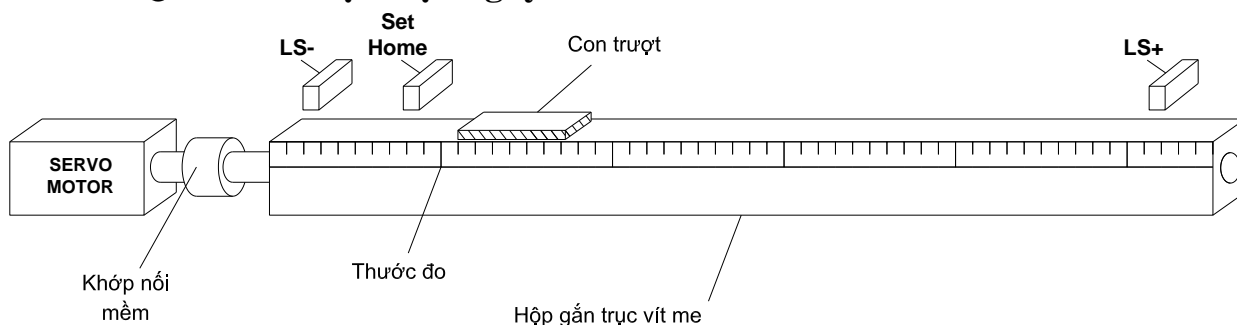
- Có tác phong công nghiệp, tỉ mỉ, cẩn thận trong quá trình thực hành;
- Có khả năng làm việc độc lập và làm việc nhóm;
- Hướng dẫn được thành viên khác thực hành cài đặt tham số và lập trình

điều khiển.

- Tự đánh giá được kết quả thực hành của bản thân và của nhóm.

II. TÓM TẮT LÝ THUYẾT

1. Quá trình thực hiện nguyên điểm



Hình 2-19.1. Cơ cấu truyền động động cơ servo – trục vít me – con trượt

- Nguyên điểm hay còn có tên gọi khác là về gốc (set home) là hoạt động bắt buộc đối với các máy CNC trước khi thực hiện gia công.

- Để trục có thể về gốc được thì bắt buộc phải có cảm biến Set home (dạng NO) như hình vẽ.

- Mục tiêu của hoạt động nguyên điểm là tìm vị trí xung Z ở tại vòng quay khi gặp cảm biến Set home

- Khi gặp cảm biến Set home thì động cơ phải quay chậm lại và bắt đầu dò vị trí xung Z (Creep Speed):

+ **Trường hợp 1:** Ở một số máy CNC thì sau khi tìm được xung Z lần 1, trục sẽ dừng ngay và kết thúc dò tìm gốc, xóa thanh ghi D8340 về 0;

+ **Trường hợp 2:** sau khi tìm được xung Z lần 1 thì động cơ tiếp tục quay cho đến khi chạm LS-, rồi quay ngược lại gặp cảm biến Set home, sau đó dò xung Z lần 2 và kết thúc quá trình về gốc khi tìm được xung Z.

- Điều kiện để về gốc được:

+ LS- (NC): Cảm biến giới hạn hành trình lùi Over -

+ LS+ (NC): Cảm biến giới hạn hành trình tiến Over +

+ Set home (NO): Cảm biến về gốc

+ Xung Z của encoder động cơ phải được đưa về PLC

Nếu LS- và LS+ được đưa về PLC thì cần cài đặt cho PD01 = 0C04, và cần lập trình bảo vệ quá hành trình cho các bit M8343 (LSH), M8344 (LSL)

- Sau khi về gốc thì:

+ Bit M8029 sẽ ON (Short pulse)

+ Thanh ghi D8341 & D8340 được xóa = 0

+ Bit M8340 Off

- **Bit M8342:** quyết định hướng về gốc:

+ M8342 = 0: Con trượt dịch chuyển sang trái để tìm gốc → điểm gốc sẽ nằm bên trái cảm biến Set home

+ M8342 = 1: Con trượt dịch chuyển sang phải để tìm gốc, khi gặp LS+ sẽ đảo chiều và chuyển động về phía cảm biến Set home → Khi gặp Set home thì đảo chiều và dò xung Z → điểm gốc sẽ nằm bên phải cảm biến Set home.

- **Bit M8349:** = 0/1 → cho phép/ngắt phát xung ra Y0, dùng trong các trường hợp dừng khẩn cấp

2. Lệnh về gốc:

- Cú pháp lệnh

-[DSZR X001 X003 Y0 Y4]-

Trong đó:

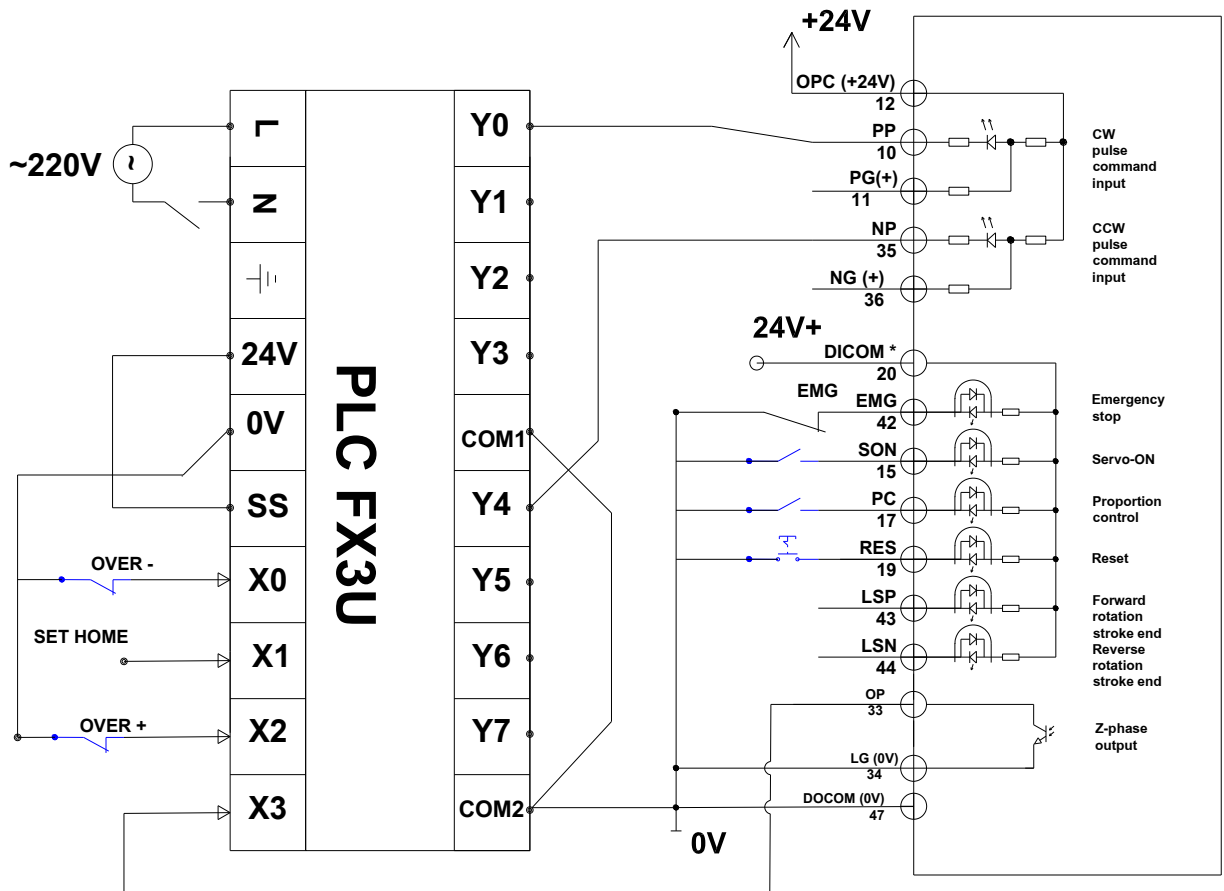
+ X001: cảm biến Set home (NO)

+ X003: xung Z (OP-33)

+ Y000: phát xung

+ Y004: điều hướng

3. Sơ đồ đấu nối



Hình 2-18.2. Sơ đồ kết nối PLC – Driver servo

III. NỘI DUNG THỰC HÀNH

1. Công tác chuẩn bị

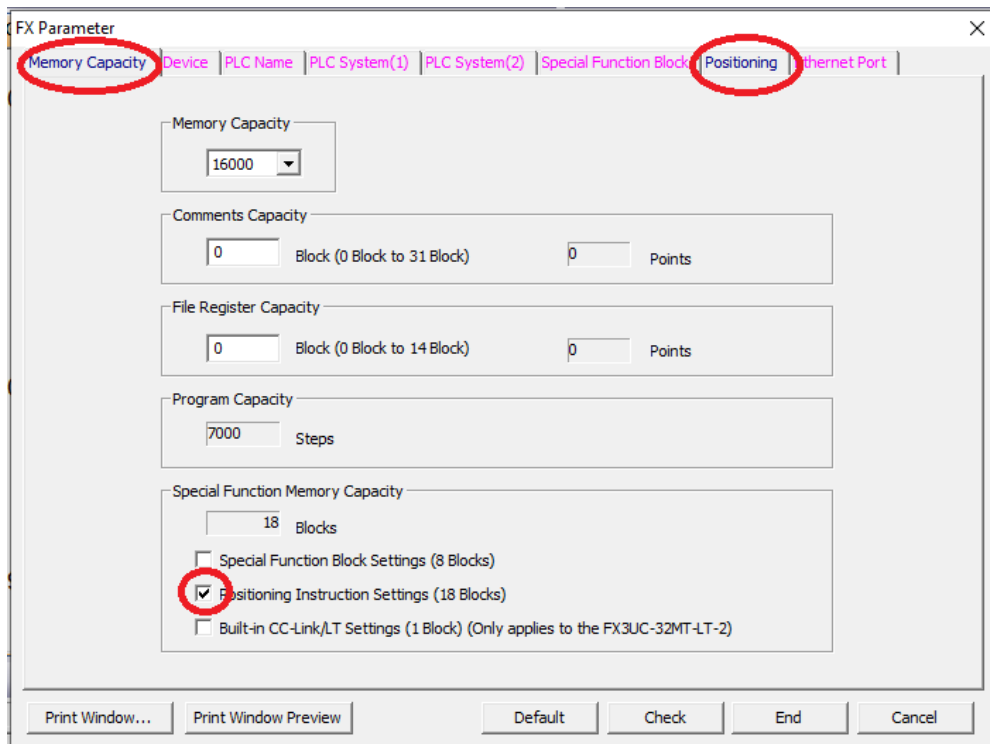
- Sử dụng cho 1 nhóm không quá 3 sinh viên

STT	Tên thiết bị, dụng cụ	Số lượng
1	Bộ thực hành động cơ servo Mitsubishi	1 bộ
2	Dây cảm mô hình	1 Bộ
3	Bộ đồ nghề dụng cụ	1 bộ

2. Trình tự thực hiện

2.1. Cài đặt parameter cho PLC

- Chọn Parameter/PLC Parameter → tại cửa số Memory capacity tick chọn Positioning Instruction Setting



- Chọn sang tab Positioning và đặt như sau:

	Y0	Y1	Y2	Y3	Setting Range
Bias Speed[Hz]	0	0	0	0	1/10 or Less of Max. Speed
Max. Speed[Hz]	20000	20000	20000	20000	10-200,000
Creep Speed[Hz]	100	100	100	100	10-32,767
Zero Return Speed[Hz]	300	300	300	300	10-200,000
Acceleration Time[ms]	100	100	100	100	50-5,000
Deceleration Time[ms]	100	100	100	100	50-5,000
Interruption Input of DVIT Instruction	X0	X1	X2	X3	X0-X7, Special M

+ Creep Speed: tốc độ dò xung Z sau khi gặp cảm biến Set home;

+ Zero Return Speed: tốc độ lúc bắt đầu nhấn nút về gốc

+ Thanh ghi D8348/D8349 lưu thời gian tăng tốc/giảm tốc

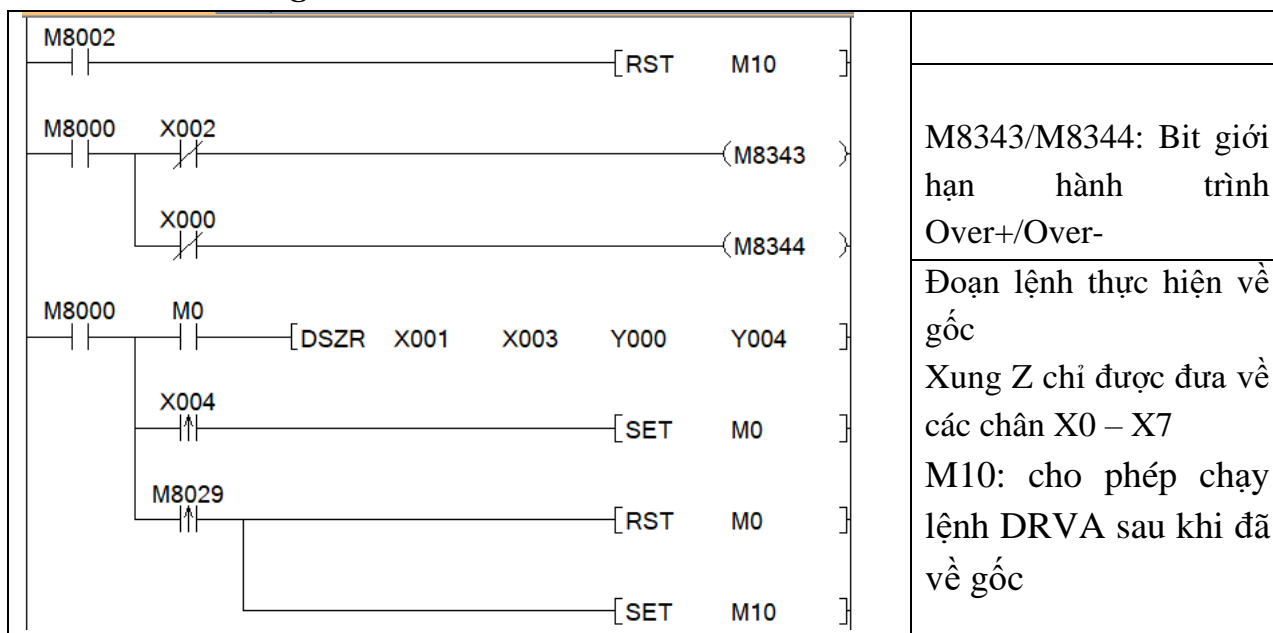
- Chọn End để kết thúc

* Chú ý khi tải chương trình vào PLC cần tick chọn để tải cả Parameter.

2.2. Cài đặt tham số Driver servo:

PD01 = 0C04	Set ON servo, động cơ chạy khi có xung; Các công tắc giới hạn hành trình đưa về PLC
PA05 = 0	
PA06 = 262144	Đặt 50 để động cơ quay hết 1 vòng
PA07 = 50	
PA13 = 11	Chế độ xung điều khiển (bài thực hành số 17)
PA14 = 1	Hướng quay của motor cùng chiều kim đồng hồ khi số xung nhập vào là dương, con trượt dịch chuyển sang phải

2.3. Chương trình điều khiển



3. Một số sai hỏng thường gặp

Hiện tượng	Nguyên nhân	Biện pháp khắc phục
Không về gốc được	Tham số cài đặt không đúng	Đặt lại các tham số cho PLC và driver
	Cảm biến Set home	Kiểm tra lại cảm biến, kiểm tra sơ đồ đấu nối

IV. TIÊU CHÍ ĐÁNH GIÁ

- Sơ đồ, và thao tác đấu nối thiết bị []
- Thao tác cài đặt tham số PLC và driver []
- Lập chương trình điều khiển []
- Vận hành mạch điện đúng trình tự, đảm bảo an toàn []
- Khắc phục một số lỗi thường xảy ra khi lập trình nguyên điểm []

V. BÁO CÁO THỰC HÀNH

1. Tên bài
2. Các tham số cài đặt và ý nghĩa
3. Sơ đồ kết nối cáp truyền thông
4. Trình tự cài đặt các tham số
5. Trình tự vận hành và kết quả từng bước.

VI. CÂU HỎI KIỂM TRA

Lập chương trình về gốc, điều khiển trực chạy JOG và di chuyển theo tọa độ mong muốn?

TÀI LIỆU THAM KHẢO

- [1] Tài liệu của hãng Mitsubishi, Fx3U Series Programmable Controllers.
- [2] Bùi Hồng Huế, Nguyễn Văn Doanh. Hướng dẫn thực hành PLC S7-200. Nhà xuất bản Xây dựng – 2011.
- [3] Bùi Hồng Huế. Giáo trình điện công nghiệp. Nhà xuất bản Xây dựng – 2003
- [4]. Nguồn internet.