

Cơ bản về Lập trình PLC

Khóa học này dành cho những học viên lập trình điều khiển bộ điều khiển khả trình lần đầu.

Khóa học này giải thích về lập trình, có thể được dùng cho bộ điều khiển khả trình MELSEC. Một trong các ngôn ngữ lập trình chính là biểu đồ ladder (Ladder Diagram, LD). Khóa học này sẽ trình bày các điểm căn bản của lập trình PLC dạng bậc thang (Ladder), trong đó bao gồm cả các lệnh chính.

Một số phần của khóa học này dựa theo khóa học cơ bản về bộ điều khiển khả trình MELSEC. Bạn nên tham gia khóa học cơ bản liên quan trước khi tham gia khóa học này.

Nội dung của khóa học này như sau.

Chương 1 - Lập trình PLC

Chương này trình bày các điểm căn bản của lập trình dạng bậc thang (Ladder).

Chương 2 - Các lệnh thiết bị bit

Chương này trình bày về các lệnh có trong các thiết bị bit (BẬT/TẮT).

Chương 3 - Các lệnh thiết bị từ

Chương này trình bày về các lệnh có trong các thiết bị từ (dạng số).

Chương 4 - Các lệnh tạo nhánh chương trình

Chương này trình bày về các lệnh tạo các chương trình nhánh.

Bài kiểm tra cuối khóa

Điểm đạt: Yêu cầu phải từ 60% trở lên.

Đến trang tiếp theo		Đến trang tiếp theo.
Trở lại trang trước		Trở lại trang trước.
Di chuyển đến trang mong muốn		"Mục lục" sẽ được hiển thị, cho phép bạn điều hướng đến trang mong muốn.
Thoát khỏi bài học		Thoát khỏi bài học.

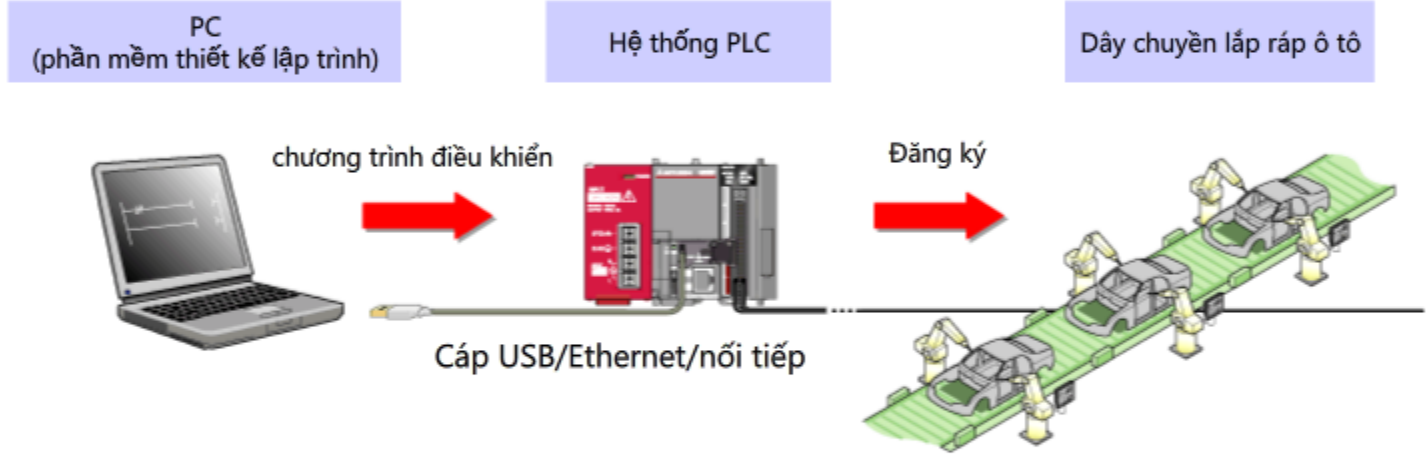
Biện pháp phòng ngừa an toàn

Khi bạn học tập dựa trên sử dụng các sản phẩm thực tế, hãy đọc kỹ các biện pháp phòng ngừa an toàn trong hướng dẫn sử dụng tương ứng.

Chương 1 Chương trình điều khiển

Các vận hành được thực hiện bởi bộ điều khiển khả trình được viết dưới dạng các chương trình điều khiển. Các chương trình này được đăng ký vào mô đun CPU, điều khiển các tín hiệu đầu vào và đầu ra (I/O) khác nhau. Các ngôn ngữ lập trình được sử dụng cho các bộ điều khiển khả trình bao gồm Trình lập trình PLC dạng bậc thang (Ladder), Danh sách lệnh (Instruction List (IL)) và Sơ đồ chức năng trình tự (Sequential Function Chart (SFC)).

Khóa học này sẽ trình bày các điểm căn bản của lập trình PLC dạng bậc thang (ladder), trong đó bao gồm cả các lệnh chính.



Trong khóa học này, phần mềm thiết kế bộ điều khiển khả trình, GX Works2 hoặc GX Works3, sẽ được dùng để tạo chương trình.

Để biết cách dùng phần mềm thiết kế bộ điều khiển khả trình, vui lòng tham gia khóa học "Khóa học Cơ Bản về GX Works2" hoặc "Engineering Software MELSOFT GX Works3 (Ladder)" (Phần mềm thiết kế MELSOFT GX Works3 (dạng bậc thang (Ladder))).

GX Works2 hỗ trợ MELSEC Sê-ri Q/L/F.

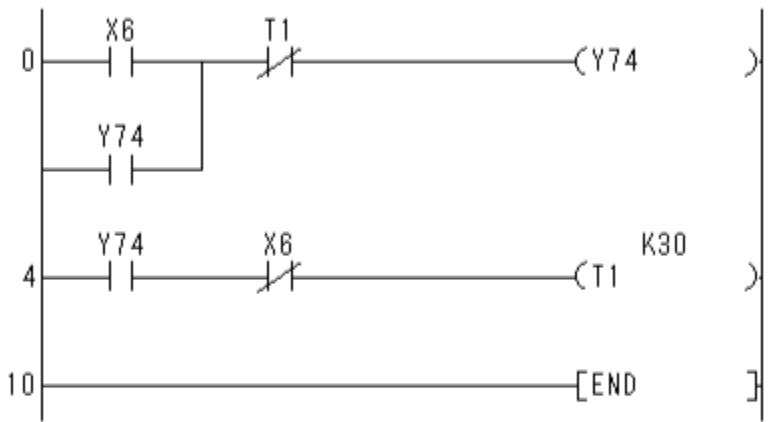
GX Works3 hỗ trợ MELSEC Sê-ri iQ-R/iQ-F.

1.1 Các ngôn ngữ lập trình

Các ngôn ngữ lập trình được sử dụng cho các bộ điều khiển khả trình bao gồm Trình lập trình PLC dạng bậc thang (Ladder), Danh sách lệnh (Instruction List (IL)) và Sơ đồ chức năng trình tự (Sequential Function Chart (SFC)).

Chương trình lập trình PLC dạng bậc thang (Ladder) là một sơ đồ logic dạng biểu đồ dựa trên mạch điện. Trong các chương trình lập trình PLC dạng bậc thang (Ladder), các biểu tượng biểu thị các lệnh được kết nối với các dòng, tương tự như sơ đồ mạch và có thể dễ dàng nhận biết các dòng vận hành.

Ngoài ra, lập trình PLC dạng bậc thang (Ladder) không yêu cầu các kiến thức lập trình đặc biệt như với các ngôn ngữ C hay BASIC và những ai đã có kinh nghiệm về mạch và rờ le điện đều có thể hiểu được dễ dàng.



Bảng dưới đây sẽ hiển thị chương trình tương tự trong IL. IL yêu cầu phải có chút kiến thức lập trình để mô tả vận hành dưới dạng các lệnh.

Bước số	Lệnh	Thiết bị
0	LD	X6
1	OR	Y74
2	ANI	T1
3	OUT	Y74
4	LD	Y74
5	ANI	X6
6	OUT	T1 K30
10	END	

Các chương trình cho bộ điều khiển khả trình có thể xử lý hai loại giá trị.

Bit	Bit được mô tả theo hai loại tín hiệu điện, BẬT và TẮT. Cũng có thể mô tả dưới dạng "1" (BẬT) và "0" (TẮT). Các giá trị bit thường được sử dụng để hiển thị các hiện trạng thiết bị I/O như công tắc và đèn.
Từ	Số và ký tự. Các giá trị từ thường được sử dụng để hiển thị số lượng và thời gian. *Khóa học này sẽ chỉ trình bày về số. Để biết chi tiết về ký tự được sử dụng làm giá trị từ, vui lòng tham khảo hướng dẫn sử dụng của sản phẩm có liên quan.

Các định dạng số dưới đây được sử dụng để hiển thị các giá trị.

- Thập phân
- Nhị phân
- Thập lục phân
- Bát phân

1.2.1**Ký hiệu hệ thập phân**

Trong ký hiệu hệ thập phân, đại lượng (số lượng) số được biểu thị bằng định dạng cơ số 10 từ "0 đến 9".

Trong bộ điều khiển khả trình MELSEC, các số thập phân đứng sau chữ "K".

Ví dụ: "K153" biểu thị số thập phân "153".

1.2.2

Ký hiệu hệ nhị phân

Trong khi ký hiệu hệ thập phân thường được dùng để biểu thị số lượng và thời gian thì các bộ điều khiển khả trình và bộ điều khiển cá nhân lại sử dụng dữ liệu nhị phân, kết hợp "0" và "1".

Bảng sau đây cho thấy sự tương ứng giữa các giá trị thập phân và nhị phân, tới số thập phân "8".

Thập phân	Nhị phân
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
⋮	⋮

Mỗi khi một lệnh có 1 từ được sử dụng trong một chương trình, nó sẽ được bộ điều khiển khả trình thực tế lưu và xử lý dưới dạng dữ liệu nhị phân 16 bit. Dữ liệu nhị phân 16 bit này tương ứng với "1 từ".

Ví dụ: số thập phân "157" được mô tả là "000000010011101" trong hệ nhị phân.

Trong ký hiệu hệ thập phân, các bit được ghi từ bên phải. (Bit xa nhất về phía bên phải là bit bắt đầu).

b15	~	b8	b7	b6	b5	b4	b3	b2	b1	b0	
0	...	0	1	0	0	1	1	1	0	1	← Vị trí bit
2^{16}		2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	← Nhị phân
32768		256	128	64	32	16	8	4	2	1	← Lũy thừa 2
											← Trọng số cho các giá trị thập phân

Để chuyển đổi các giá trị nhị phân sang thập phân, nhân mỗi trạng thái bit ("0" hoặc "1") nhân với trọng số tương ứng và cộng tất cả các kết quả lại.

$$\begin{aligned}
 &= 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\
 &= 128 + 16 + 8 + 4 + 1 \\
 &= 157
 \end{aligned}$$

Từ cách tính trên, có thể coi hệ nhị phân là định dạng số dựa trên trọng số.

1.2.3 Ký hiệu hệ thập lục phân

Trong ký hiệu hệ thập lục phân, độ lớn (số lượng) của một số được hiển thị sử dụng cơ số 16 hoặc được biểu thị bằng cách sử dụng 16 ký tự chữ và số: 0 đến 9 và A đến F. Mỗi số trong ký tự hệ thập lục phân tăng theo thứ tự 0, 1...9, A...E, rồi đến F. Khi giá trị vượt quá cơ số "F", giá trị đó sẽ được mang sang bên trái và trở thành "10".

Trong bộ điều khiển khả trình MELSEC, các số thập lục phân đứng sau chữ "H".

Ví dụ: "H4A9D" biểu thị số thập lục phân "4A9D".

Ký hiệu hệ nhị phân có thể dài và khó sử dụng trong các chương trình và trên hiển thị màn hình.

Trong trường hợp đó, ký hiệu hệ thập lục phân sẽ được sử dụng.

Một số trong giá trị thập lục phân có thể biểu diễn 4 bit (4 số) trong giá trị nhị phân.

Hình dưới đây hiển thị giá trị thập lục phân được biểu diễn như thế nào dưới dạng giá trị thập phân.

4	3	2	1	← Thứ tự số
4	A	9	D	← Giá trị thập lục phân
16^3	16^2	16^1	16^0	← Lũy thừa 16

$$= 4 \times 16^3 + A \times 16^2 + 9 \times 16^1 + D \times 16^0$$

(4096) (10) (256) (16) (13) (1)

$$= 19101$$

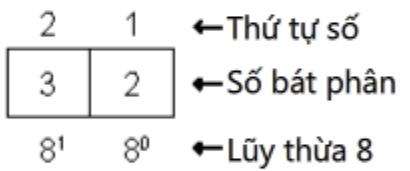
* Một số trong giá trị thập lục phân có thể biểu diễn 4 bit nhị phân.

Thập phân	Nhị phân	Thập lục phân
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
19101	0100 1010 1001 1101 4A9D ↑ ↑ ↑ ↑ ↑	

1.2.4 Ký hiệu hệ bát phân

Trong ký hiệu hệ bát phân, độ lớn (số lượng) của một số được biểu thị bằng định dạng cơ số 8 ("0 đến 7").
 Khi giá trị tăng từ "0", "1", "2" đến "7", giá trị đó được mang sang bên trái và trở thành "10".
 Ký hiệu hệ bát phân này được sử dụng cho các số thứ tự I/O trong MELSEC Sê-ri iQ-F/F.

Hình dưới đây hiển thị giá trị bát phân được biểu diễn như thế nào dưới dạng giá trị thập phân.



$$= 3 \times 8^1 + 2 \times 8^0$$

(8) (1)

$$= 26$$

* Một số trong giá trị bát phân có thể biểu diễn 3 bit nhị phân.

Thập phân	Nhị phân	Thập lục phân	Bát phân
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
16	1101	D	15
14	1110	E	16
15	1111	F	17

16	1000	10	20
17	10001	11	21
18	10010	12	22
.	.	.	.
.	.	.	.
.	.	.	.
26	011 010	1A	32

↑ ↑ ↑

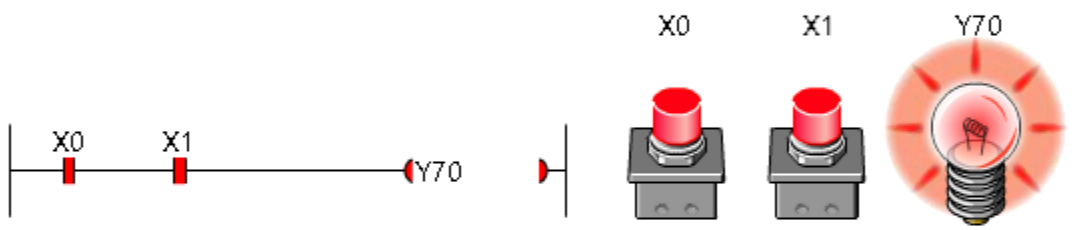
1.3 Cơ bản về lập trình PLC

Trong điều khiển trình tự, một chuỗi vận hành được thực hiện dựa trên các tín hiệu BẬT/TẮT nhận được từ thiết bị kết nối với mô đun đầu vào và sau đó các kết quả vận hành được đưa ra thiết bị kết nối với mô đun đầu ra.

Để thực hiện điều khiển này, một chương trình điều khiển phải có các điều kiện đầu vào và có các đầu ra, việc này sẽ được thực hiện khi thỏa mãn các điều kiện đầu vào.

Chương trình dưới đây sẽ đặt lệnh cho các điều khiển sau:

- Khi các công tắc loại bấm vào được kết nối với các bộ đấu nối dây X1 và X2 đều BẬT, bật bộ đấu nối dây Y70
- Kết quả vận hành được đưa ra bộ đấu nối dây Y70 để BẬT đèn được kết nối



Ấn đồng thời các công tắc X0 và X1 sẽ BẬT đèn Y70.

1.4 Thiết bị và số thứ tự I/O

Các chương trình được mô tả trong Chương 1.3 sử dụng các ký hiệu chữ và số như X0, X1 và Y70 để xác định thiết bị I/O. Các ký tự chữ và số được gọi là các số thứ tự I/O.

Chương này sẽ trình bày về các thiết bị và số thứ tự I/O cần thiết để lập các chương trình điều khiển.

MELSEC Sê-ri iQ-R/Q/L/iQ-F và MELSEC Sê-ri F sử dụng các định dạng khác nhau để biểu thị số hiệu thiết bị. Bảng dưới đây tóm tắt sự khác nhau này.

Các Sê-ri MELSEC	Bit			Từ
	X (số thứ tự đầu vào)	Y (số thứ tự đầu ra)	M (rờ le gắn trong)	D (thanh ghi dữ liệu)
Sê-ri iQ-R/Q/L	Thập lục phân	Thập lục phân	Thập phân	Thập phân
Sê-ri iQ-F/F	Bát phân	Bát phân	Thập phân	Thập phân

1.4.1 Số thứ tự I/O và tín hiệu I/O (MELSEC Sê-ri iQ-R/Q/L)

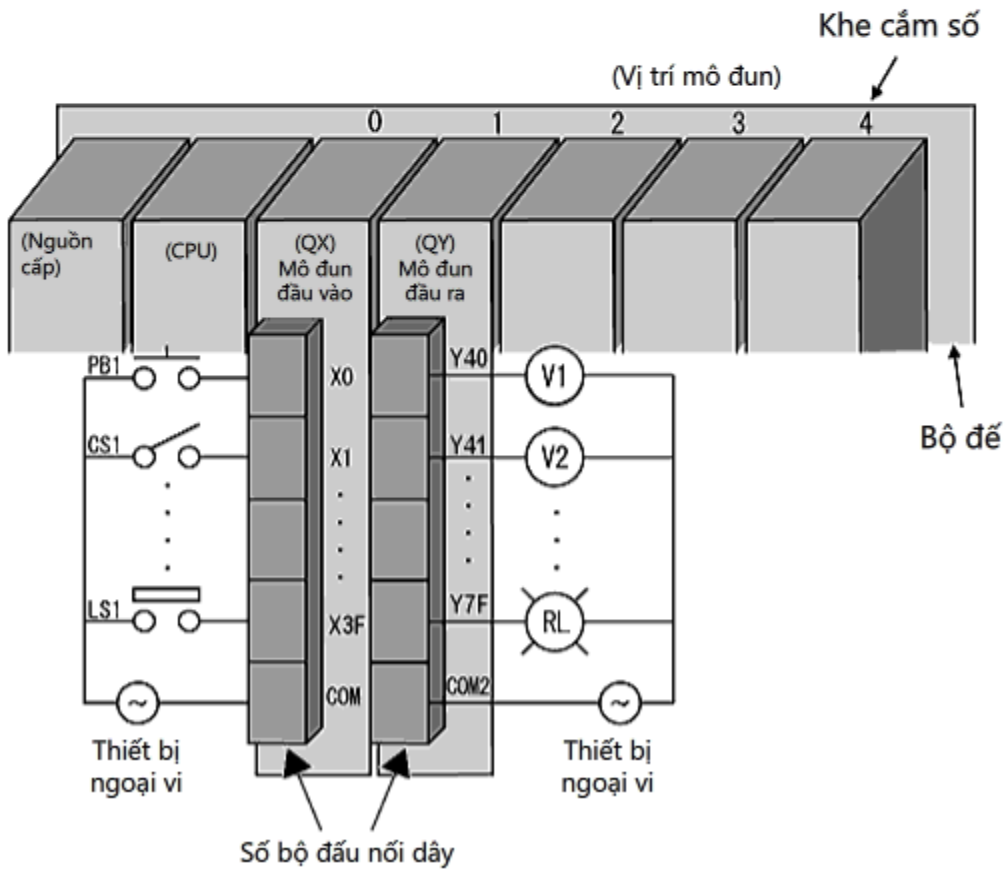
MELSEC Sê-ri iQ-R/Q/L

Một số thứ tự I/O bao gồm một chữ cái, biểu thị đầu vào (X) hoặc đầu ra (Y) và một giá trị thập lục phân biểu thị số bộ đấu nối dây.

Một số thứ tự I/O được xác định trước tiên dựa trên vị trí lắp đặt mô đun.

Sau đó, phạm vi số thứ tự I/O sẽ được xác định dựa trên số điểm I/O có thể quản lý của mô đun. (Số điểm I/O có thể quản lý tương ứng với số bộ đấu nối dây của mô đun I/O).

Hình dưới đây minh họa cách chỉ định các số thứ tự I/O cho mô đun đầu vào 64 điểm và mô đun đầu ra 64 điểm, các mô đun này được lắp tương ứng vào khe cắm số 0 và số 1.



1.4.2 Số thứ tự I/O và tín hiệu I/O (MELSEC Sê-ri iQ-F/F)

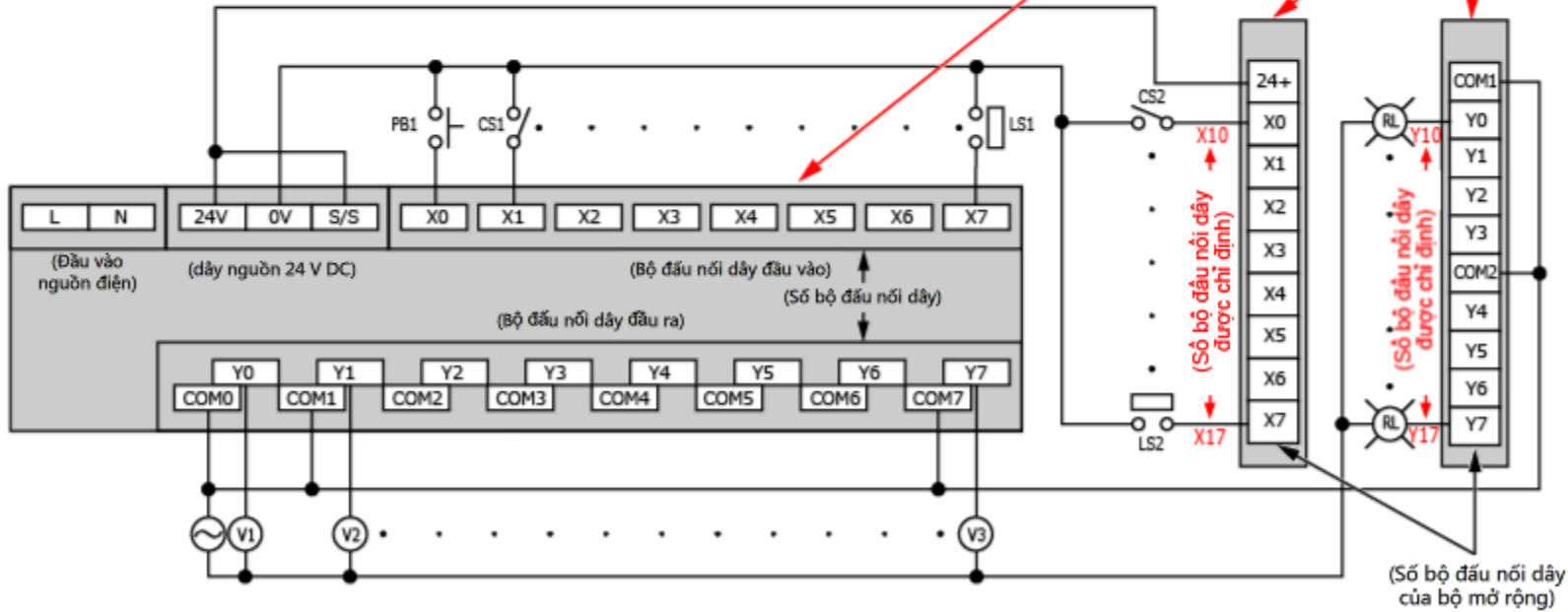
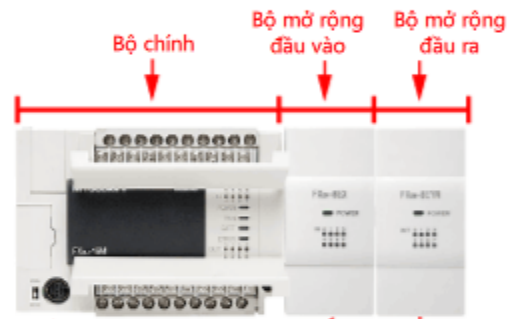
MELSEC Sê-ri iQ-F/F

Một số thứ tự I/O bao gồm một chữ cái, biểu thị đầu vào (X) hoặc đầu ra (Y) và một giá trị bát phân biểu thị số bộ đấu nối dây.

Một số thứ tự I/O trước tiên được xác định dựa trên số thứ tự I/O cuối của bộ chính hoặc khối mở rộng I/O đứng trước đó. Sau đó, phạm vi số thứ tự I/O sẽ được xác định dựa trên số điểm I/O có thể quản lý của bộ. (Số điểm I/O có thể quản lý tương ứng với số điểm I/O do bộ mở rộng I/O sử dụng).

Số đầu tiên của số thứ tự I/O luôn bắt đầu từ "0" đối với bộ mở rộng mới. Ví dụ: nếu số thứ tự I/O của bộ trước đó kết thúc tại X7, số thứ tự I/O của bộ tiếp theo sẽ bắt đầu từ X10.

Hình dưới đây minh họa cách chỉ định các số thứ tự I/O cho bộ mở rộng đầu vào 8 điểm và bộ mở rộng đầu ra 8 điểm, các bộ này được bổ sung vào bộ chính MELSEC-Sê-ri F.



Bộ chính, Bộ mở rộng đầu vào, Bộ mở rộng đầu ra

(Số bộ đấu nối dây được chỉ định)

(Số bộ đấu nối dây được chỉ định)

(Số bộ đấu nối dây của bộ mở rộng)

1.4.3 Thiết bị và số thứ tự I/O

Các hiện trạng của thiết bị nối với bộ được lưu trong khu vực bộ nhớ của bộ điều khiển khả trình được gọi là "thiết bị". Giống như đầu ra, thiết bị đầu ra hoạt động theo các hiện trạng của thiết bị.

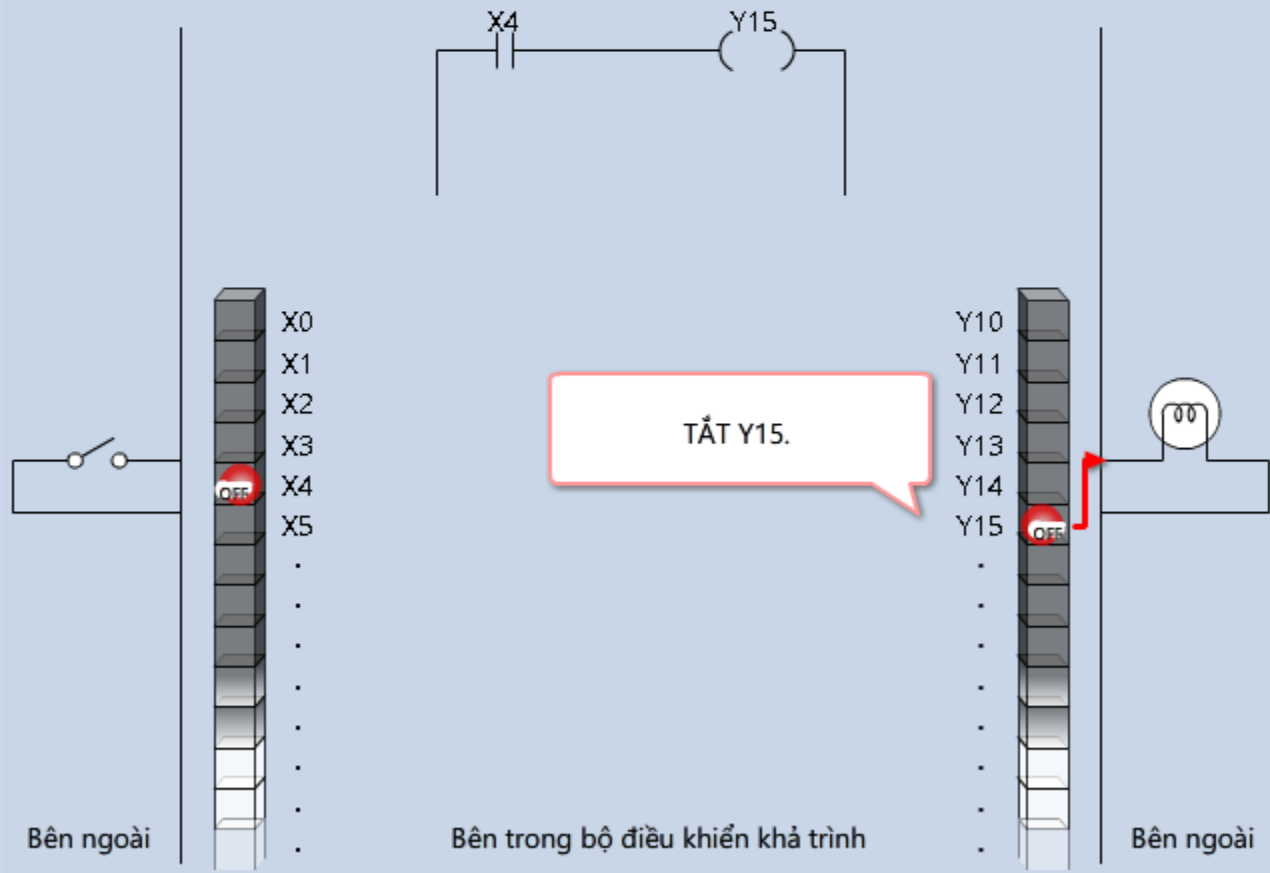
Như đã trình bày ở trên, các chương trình điều khiển thường được vận hành dựa trên các hiện trạng của thiết bị.

Các thiết bị lưu dữ liệu bit (BẬT/TẮT) như đầu vào (X) và đầu ra (Y) được gọi là "thiết bị bit".

Số hiệu thiết bị tương ứng với số thứ tự I/O.

Ví dụ: hiện trạng của bộ đầu nối dây được gán với số thứ tự I/O X0 được lưu trong thiết bị X0.

Cũng tương tự như vậy, hiện trạng của thiết bị Y10 tương ứng với bộ đầu nối dây được gán với số thứ tự I/O Y10.



Bên ngoài

Bên trong bộ điều khiển khả trình

Bên ngoài

1.4.4

Rờ le gắn trong

Chúng ta đã tìm hiểu về các thiết bị bit như X (đầu vào) và Y (đầu ra), tương ứng với các số gán cho các bộ đấu nối dây I/O của mô đun thực tế.

Còn một nhóm thiết bị bit khác, không liên quan đến các bộ đấu nối dây mô đun I/O và một thiết bị trong nhóm đó được gọi là "rờ le gắn trong (M)".

Các rờ le gắn trong (M) được biểu thị bằng định dạng thập phân, mặc dù thiết bị đầu vào (X) và thiết bị đầu ra (Y) được biểu thị ở hệ thập lục phân cho MELSEC Sê-ri iQ-R/Q/L và ở hệ bát phân cho MELSEC Sê-ri iQ-F/F.

Các rờ le gắn trong (M) được sử dụng chủ yếu để lưu dữ liệu bit tạm thời.

Ví dụ: có thể sử dụng các rờ le gắn trong (M) để lưu các kết quả tính toán vận hành để sử dụng trong thanh thang khác.

1.4.5 Thiết bị từ

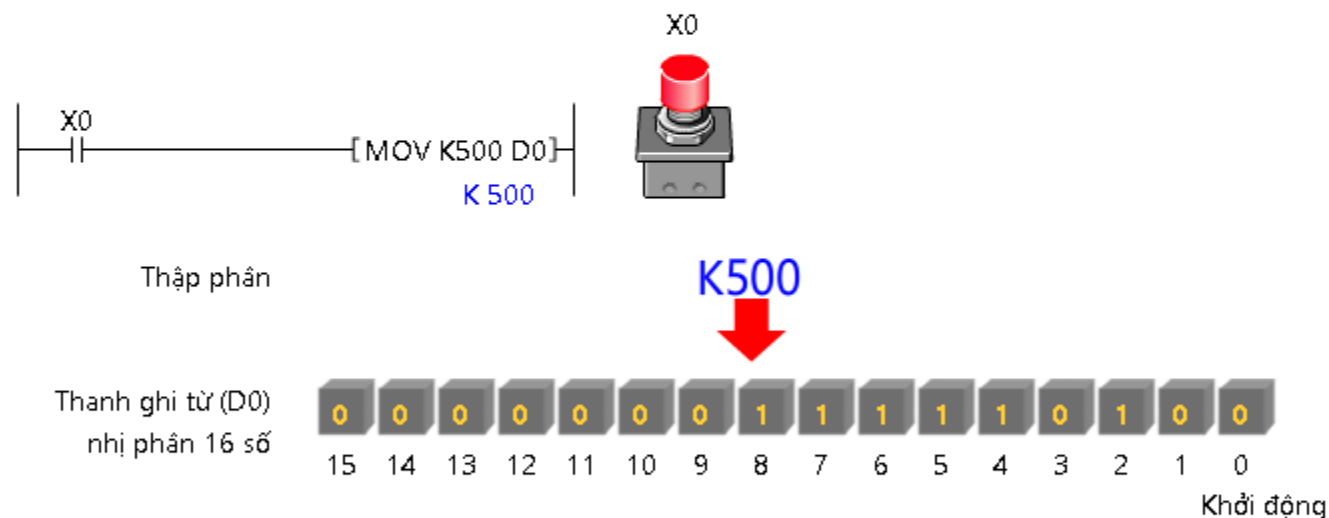
Chúng ta đã biết rằng các thiết bị lưu thông tin bit (BẬT/TẮT) được gọi là "thiết bị bit" và các thiết bị lưu thiết bị từ được gọi là "thiết bị từ".

"Thanh ghi dữ liệu" (D) là một trong các thiết bị từ thường được sử dụng. Một thanh ghi dữ liệu (D) có thể lưu 1 từ (16 bit) dữ liệu.

Ảnh động dưới đây minh họa cách sử dụng thanh ghi dữ liệu (D).

Trong chương trình này, BẬT X0 sẽ lưu "K500" (giá trị thập phân) thành D0.

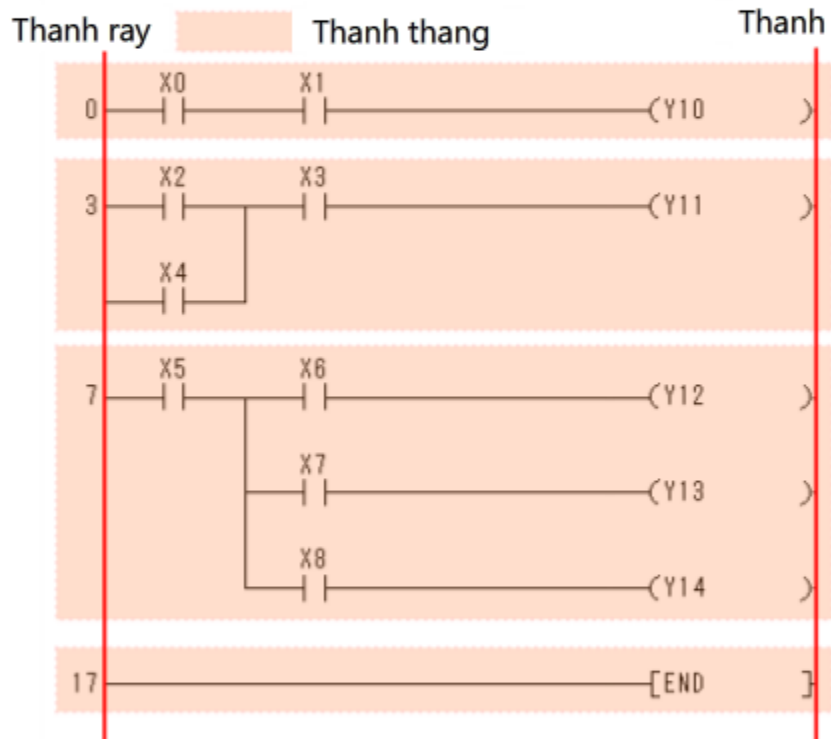
Lệnh MOV sẽ sao chép giá trị tới một thiết bị được chỉ định. (Nội dung chi tiết hơn sẽ được trình bày trong Chương 3.1). Trong phần này, các số được lưu trong các thanh ghi dữ liệu.



Nếu nhả nút và X0 TẮT, "K500" được lưu vào thanh ghi dữ liệu D0 sẽ được duy trì.

1.5 Lập chương trình điều khiển

Các chương trình điều khiển có các thanh ray ở cạnh xung trái và phải cùng các biểu tượng lệnh nối với các đường.
 Một khu vực từ lệnh -| |- nối với thanh ray bên trái sang lệnh-()- hoặc -[]- nối với thanh ray bên phải được gọi là thanh thang.
 Một số thanh thang này cấu thành chương trình điều khiển kết thúc bằng lệnh -[END]- hoặc -[FEND]-.



■ Sự khác nhau giữa các phần mềm thiết kế

Các lệnh -()- và -[]- là khác nhau giữa các phần mềm thiết kế, GX Works2 và GX Works3. Trong khóa học này, phần mô tả sử dụng các lệnh của GX Works2.

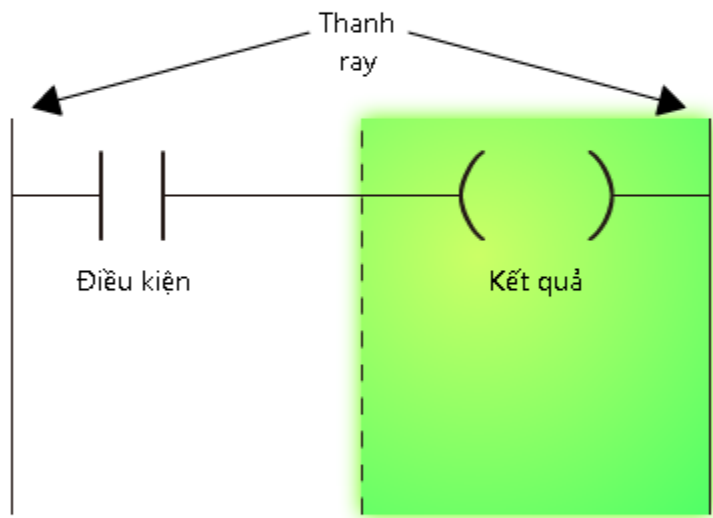
	GX Works2	GX Works3
-()-	<Y10 >	Y10
-[]-	{MOV K500 D0 }	

1.5.1 Biểu tượng lệnh

Như đã trình bày trong Chương 1.3, một bộ điều khiển khả trình phải được lập lệnh để thực hiện các vận hành nhất định khi thỏa mãn được điều kiện đầu vào được quy định trước. Đối với các lệnh đó, các biểu tượng lệnh được sử dụng để biểu thị các điều kiện đầu vào và các chi tiết đầu ra.

Một biểu tượng lệnh thường có một số hiệu thiết bị. Số hiệu thiết bị quy định một khu vực (thiết bị) lưu một hiện trạng, hiện trạng này sẽ được sử dụng để đánh giá điều kiện hoặc làm kết quả đầu ra.

Một thanh thang có các điều kiện và kết quả đầu ra. Các điều kiện được đặt bên trái và kết quả đầu ra được đặt bên phải. Các kết quả đầu ra có thể là tín hiệu BẬT/TẮT đơn giản hoặc là một lệnh riêng như tính toán hoặc sao chép hoạt động.



Trong biểu đồ ladder, có hai thanh ray song song. Các điều kiện được ghi ở bên trái. Các kết quả được ghi ở bên phải.

1.6

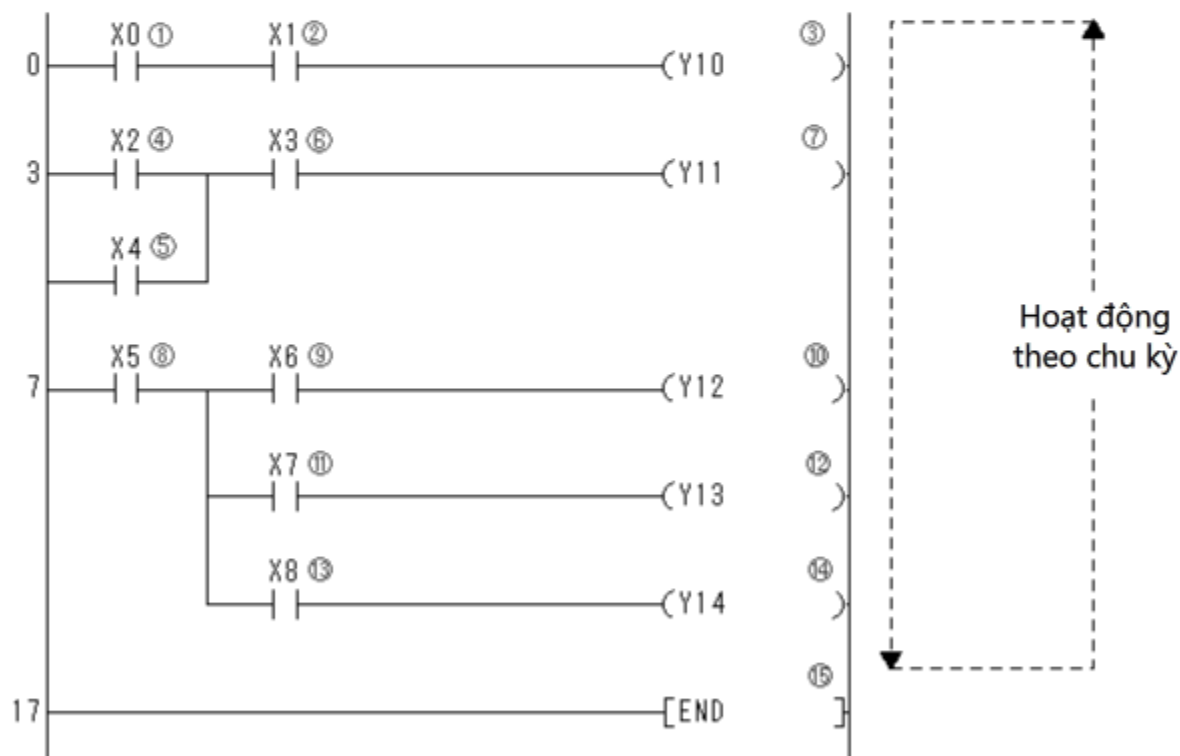
Quy trình thực hiện chương trình

Một chương trình bắt đầu từ lệnh khởi động bên trái và kết thúc bằng lệnh `-[END]-`. Sau khi đến lệnh `-[END]-`, việc thực hiện chương trình sẽ bắt đầu lại từ lệnh khởi động. Việc thực hiện lặp đi lặp lại này được gọi là "hoạt động theo chu kỳ".

Một chu kỳ trong hoạt động theo chu kỳ này được gọi là một lần "quét" và khoảng thời gian cần để xử lý một lần quét được gọi là "thời gian quét".

Hình dưới đây minh họa quy trình thực hiện lệnh.

Các lệnh được thực hiện từ trái sang phải trên mỗi thanh thang, sau đó từ thanh thang trên xuống dưới (Số 1, 2,...15 -> 1...).



1.7 Thời gian làm tươi

Như đã trình bày ở trên, thời gian quét là khoảng thời gian thực hiện một chuỗi các chương trình chỉ một lần. Thời gian quét cũng có thể được biểu thị dưới dạng:

$$\text{Thời gian quét} = \text{thời gian làm tươi} + \text{thời gian thực hiện chương trình} + \text{thời gian KẾT THÚC xử lý}$$

Thời gian làm tươi là khoảng thời gian đọc dữ liệu từ mô đun đầu vào sang thiết bị đầu vào (X) cộng với khoảng thời gian ghi dữ liệu vào mô đun đầu ra từ thiết bị đầu ra (Y).



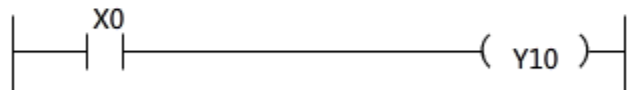
Lưu ý rằng các hiện trạng BẬT/TẮT của công tắc thực tế được đọc ngay và được lưu vào thiết bị đầu vào (X), đồng thời dữ liệu mới sẽ ghi đè các giá trị hiện tại trong khi làm tươi. Cũng tương tự như vậy, dữ liệu trong thiết bị đầu ra (Y) được ghi ngay vào mô đun đầu ra khi thực hiện lệnh.

(1) Làm mới I/O

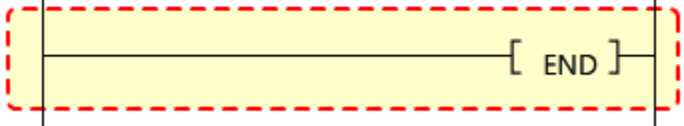
- Gửi hiện trạng BẬT/TẮT của các thiết bị đầu ra tới thiết bị đầu ra được kết nối
- Lưu hiện trạng BẬT/TẮT nhận được từ thiết bị đầu vào được kết nối vào các thiết bị đầu vào

Điều này có nghĩa là một tín hiệu sẽ chuyển từ TẮT sang BẬT, sau đó lại sang TẮT, tín hiệu này không bao giờ được nhận dạng là BẬT. Tuy nhiên, thời gian quét là rất ngắn so với độ dài tín hiệu. Hiếm khi bộ điều khiển khả trình bỏ sót thay đổi hiện trạng tín hiệu.

(2) Thực hiện chương trình



(3) END xử lý



END xử lý bằng lệnh END (Chi tiết đã bị xóa ở đây).

Chương 2 Các lệnh thiết bị bit



Chương này trình bày về các lệnh sử dụng các thiết bị bit (BẬT/TẮT).

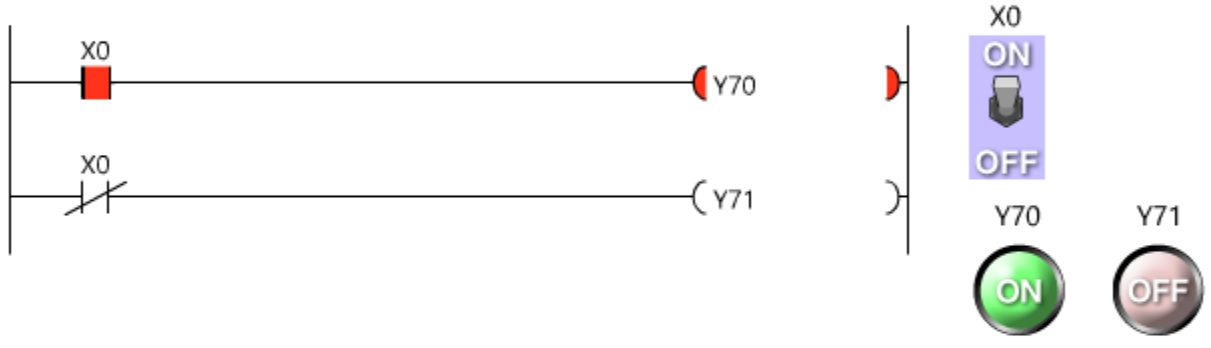
Các vận hành sử dụng thiết bị bit là các vận hành cơ bản nhất trong chương trình điều khiển.
Các đầu vào từ thiết bị đầu vào được sử dụng làm điều kiện để điều khiển thiết bị đầu ra.

2.1 Các điều kiện đầu vào và đầu ra

Các tiếp điểm thường mở (Normally open (NO)) và thường đóng (normally closed (NC)) được sử dụng làm các điều kiện đầu vào. Khi các điều kiện đầu vào được thỏa mãn, lệnh đầu ra cuộn cảm (lệnh OUT) sẽ được xuất ra. Khi các điều kiện đầu vào không được thỏa mãn, lệnh đầu ra cuộn cảm sẽ không được xuất ra. Lệnh tiếp điểm NO/NC và lệnh OUT là cách kết hợp lệnh chính được sử dụng trong chương trình điều khiển.

Chương trình ladder và vận hành

Mô phỏng các lệnh NO, NC và OUT bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.

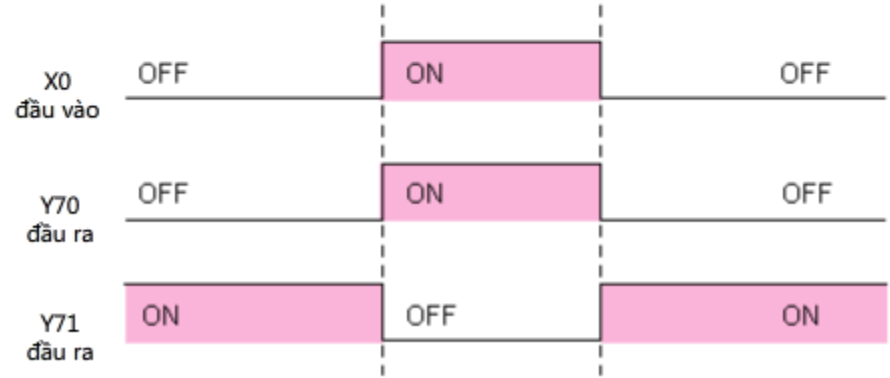


2.1 Các điều kiện đầu vào và đầu ra

■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	Tiếp điểm NO Dẫn điện khi hiện trạng thiết bị là BẬT.
	Tiếp điểm NC Dẫn điện khi hiện trạng thiết bị là TẮT (ngược lại với tiếp điểm NO).
	Đầu ra cuộn cảm (OUT) Khi điều kiện đầu vào trước đó đã được thỏa mãn, dữ liệu trong thiết bị cài sẵn sẽ được xuất ra.
	Lệnh kết thúc (END) Chỉ báo kết thúc chương trình. Một chương trình cần phải có lệnh END.

■ Biểu đồ thời gian



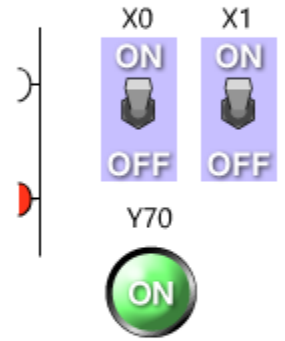
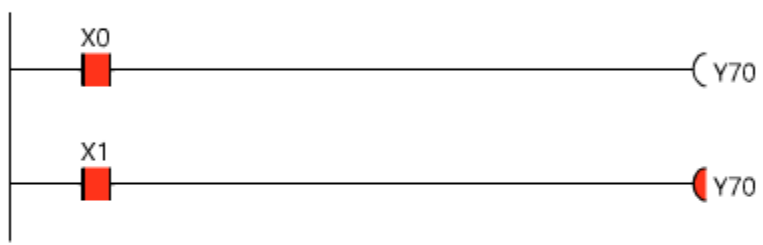
2.1.1 Sử dụng cùng số hiệu thiết bị cho các lệnh

Trên một thanh thang, chỉ có thể sử dụng một lệnh OUT với một số hiệu thiết bị. Nếu có nhiều lệnh OUT được sử dụng với một số hiệu thiết bị, chỉ có lệnh OUT sau là hợp lệ, khiến cho lệnh OUT trước trở thành không hợp lệ.

Chương trình ladder

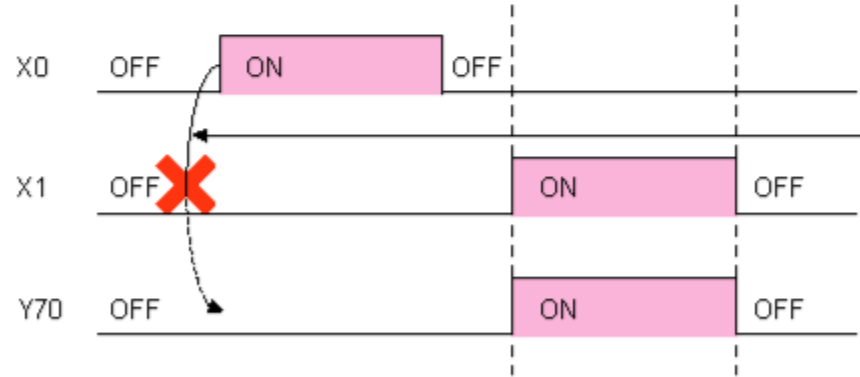
Mô phỏng hoạt động của hai lệnh có cùng số hiệu thiết bị bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.

Loại sử dụng này (sử dụng OUT Y70 cho hai lệnh) được gọi là "cuộn cảm nhân đôi".



Khi X1=ON, Y70=ON.

Biểu đồ thời gian



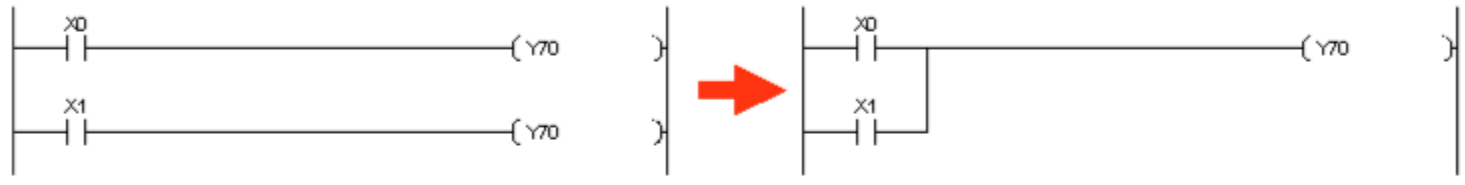
Điều kiện đầu vào X0 bị bỏ qua vì điều kiện đầu vào sau được ưu tiên.

2.1.1 Sử dụng cùng số hiệu thiết bị cho các lệnh

■ Ví dụ về cách sửa

Trong ví dụ này, điều kiện đầu vào "X1" được ưu tiên hơn và "X0" bị bỏ qua.

Bằng cách sửa thanh thang này thành thanh thang trong Hình B, thiết bị Y70 được BẬT khi một trong hai điều kiện đầu vào được thỏa mãn, tránh xung đột giữa hai lệnh OUT.



(Hình A)

(Hình B)

2.2

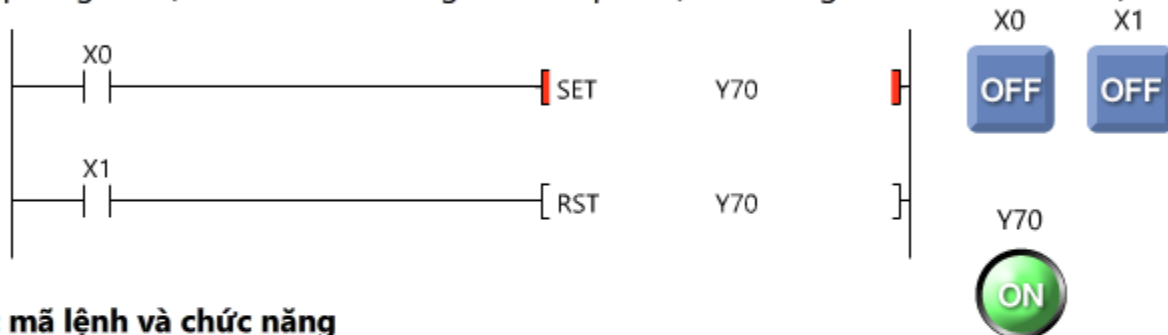
Duy trì/hủy các đầu ra

Không giống như lệnh OUT, lệnh duy trì vận hành (lệnh SET) sẽ duy trì hiện trạng đầu ra ngay cả khi điều kiện đầu vào trở thành không được thỏa mãn.

Để hủy đầu ra (TẮT), có thể thực hiện lệnh hủy duy trì vận hành (lệnh RST).

■ Chương trình ladder và vận hành

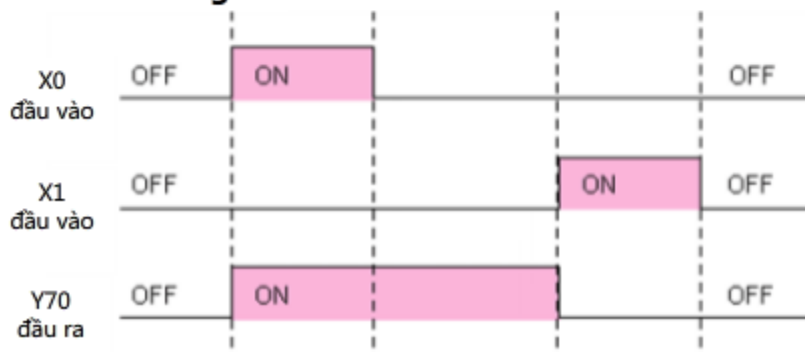
Mô phỏng các lệnh SET và RST bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.



■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	Lệnh duy trì vận hành (SET) Lệnh này sẽ BẬT thiết bị và duy trì hiện trạng BẬT (đầu ra). Đầu ra được duy trì ngay cả khi điều kiện đầu vào trở thành không được thỏa mãn.
	Lệnh hủy duy trì vận hành (RSET) Hủy hiện trạng BẬT và hủy đầu ra tới thiết bị được chỉ định.

■ Biểu đồ thời gian



2.2.1

Sự khác nhau giữa các lệnh OUT và SET



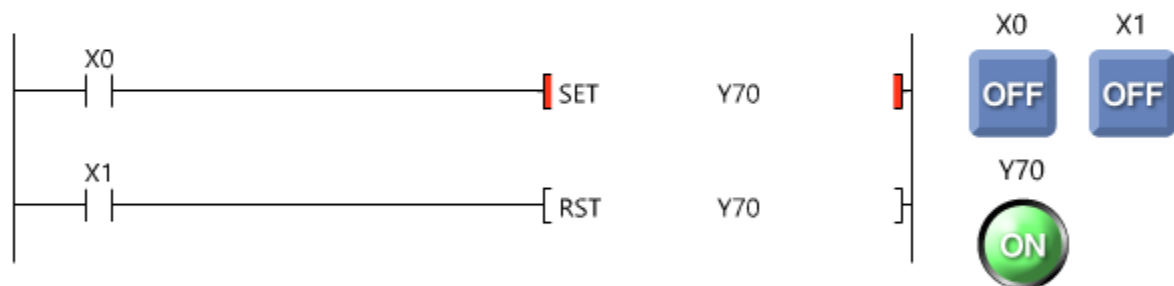
Mô phỏng sự khác nhau trong vận hành giữa các lệnh OUT và SET bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.

■ Lệnh OUT



Y70 BẬT khi điều kiện đầu vào được thỏa mãn.

■ Các lệnh SET/RST

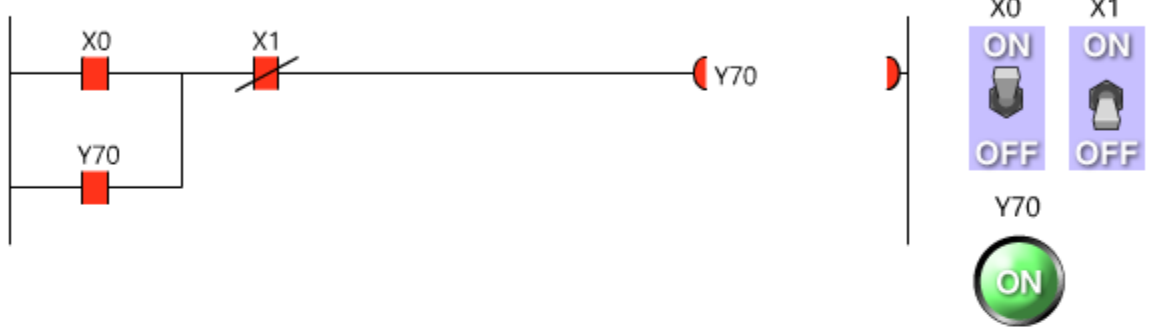


Khi điều kiện đầu vào được thỏa mãn, Y70 BẬT cho đến khi lệnh RST được thực hiện.

2.2.2 Thay các thang duy trì bằng lệnh SET

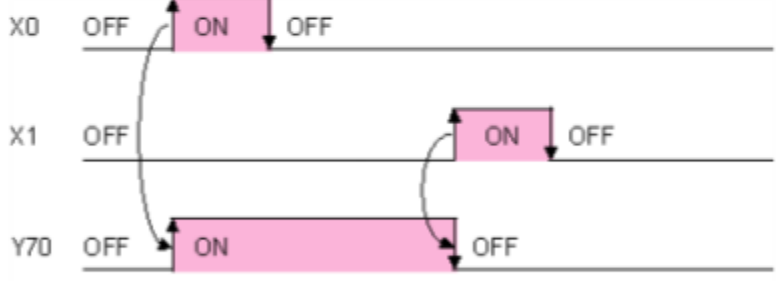
Chương trình lập trình PLC dạng bậc thang (Ladder) và vận hành

Mô phỏng hoạt động của thang duy trì bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.



Khi X0=BẬT và X1=TẮT, Y70=BẬT.
Y70 = BẬT (duy trì) cho đến khi X1=BẬT.

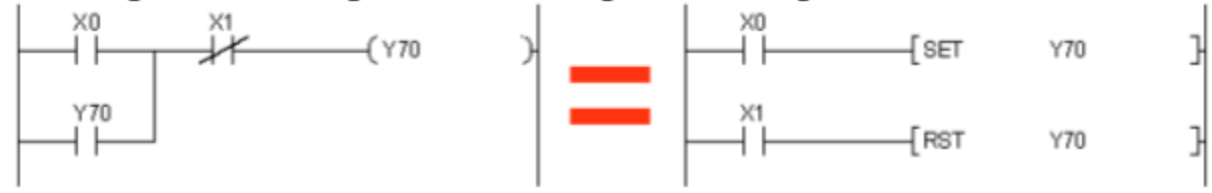
■ Biểu đồ thời gian



Ngay cả sau khi X10 TẮT, Y70 (cuộn cảm) vẫn BẬT (duy trì)

■ Thay bằng lệnh SET

Một chương trình ladder duy trì có thể được viết lại dưới dạng chương trình ladder với lệnh SET. Có thể đơn giản hóa chương trình ladder bằng cách sử dụng lệnh SET.



2.3 Thêm các điều kiện (logic AND)

Để có logic AND, các tiếp điểm NO/NC sẽ được đặt nối tiếp
Trong một logic AND, điều kiện sẽ được thỏa mãn khi nhiều tiếp điểm NO/NC nối tiếp với nhau đều là BẬT.

Chương trình ladder và vận hành

Mô phỏng vận hành logic AND bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải.



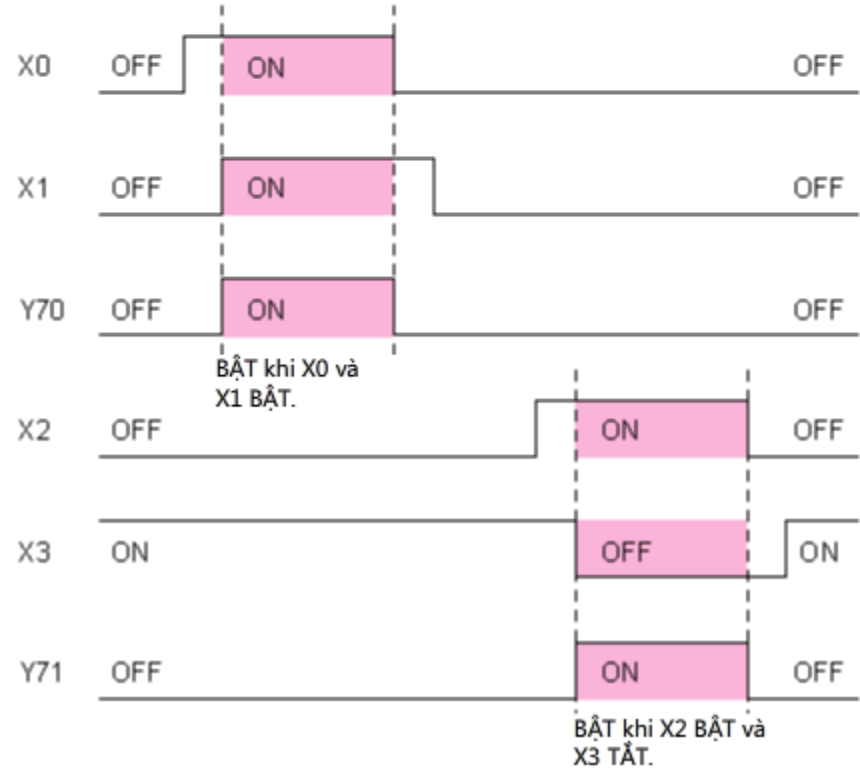
Khi X0 và X1 BẬT, Y70 BẬT.
Khi X2 BẬT và X3=TẮT, Y71=BẬT.

2.3 Thêm các điều kiện (logic AND)

■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	Kết nối nối tiếp tiếp điểm NO Tiếp điểm NO được kết nối nối tiếp (theo chiều ngang).
	Kết nối nối tiếp tiếp điểm NC Tiếp điểm NC được kết nối nối tiếp (theo chiều ngang).

■ Biểu đồ thời gian



2.4 Thêm các điều kiện (logic OR)

Để có logic OR, các tiếp điểm NO/NC phải được bố trí song song.
Trong một logic OR, điều kiện được thỏa mãn khi một trong các tiếp điểm NO/NC nối song song, là BẬT.

Chương trình ladder và vận hành

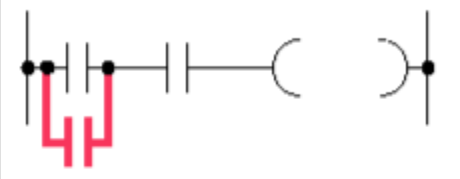
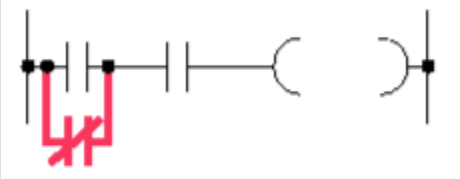
Mô phỏng vận hành logic OR bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải.



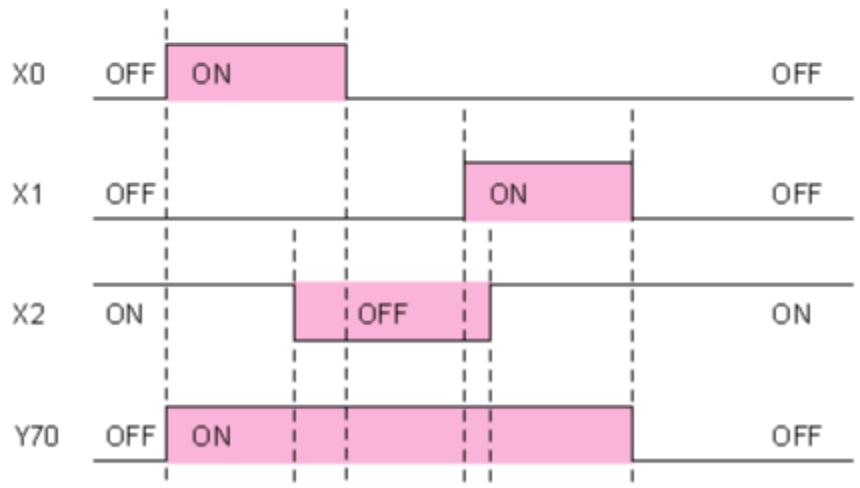
Y70 BẬT khi điều kiện sau được thỏa mãn. X0 BẬT, X1 BẬT hoặc X2 TẮT.

2.4 Thêm các điều kiện (logic OR)

■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	<p>Kết nối song song các tiếp điểm NO Tiếp điểm NO được kết nối song song (theo chiều dọc).</p>
	<p>Kết nối song song các tiếp điểm NC Tiếp điểm NC được kết nối song song (theo chiều dọc).</p>

■ Biểu đồ thời gian



2.5 Đầu ra dưới dạng xung

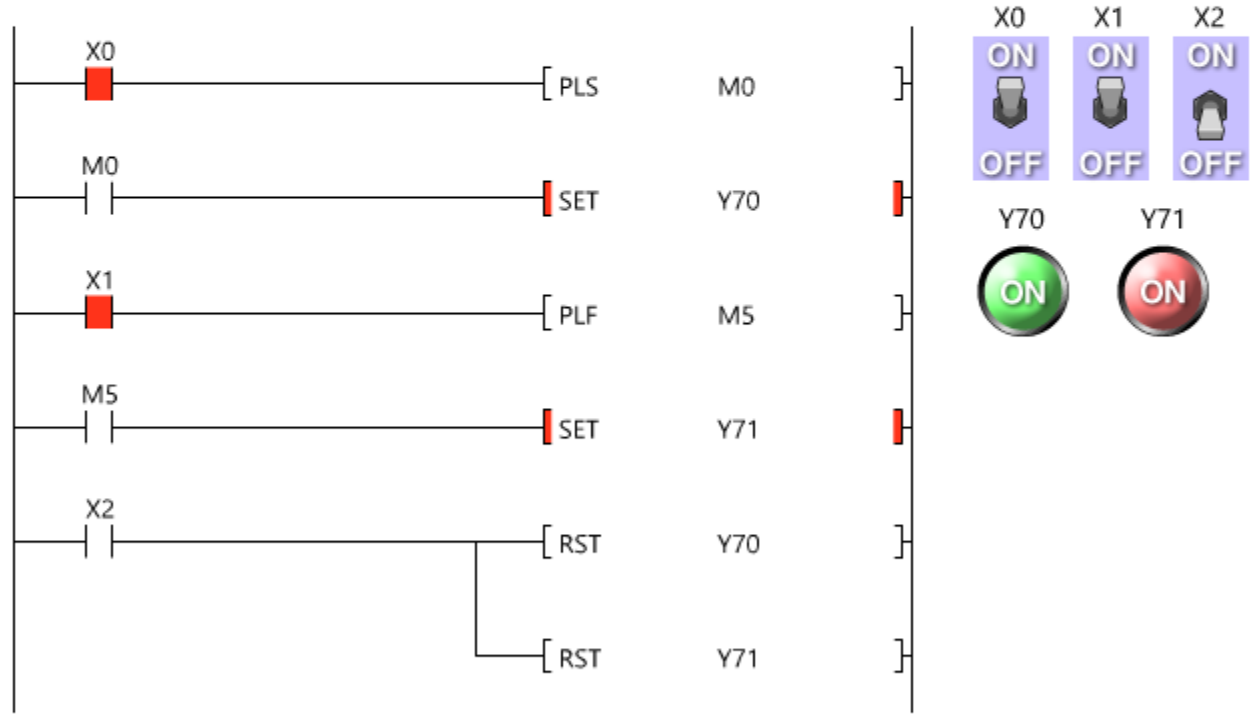
Không giống như lệnh OUT, lệnh cạnh xung lên (lệnh PLS) BẬT cuộn cảm cho một lần quét sau khi điều kiện đầu vào được thỏa mãn.

Ngược với lệnh PLS, lệnh cạnh xung xuống (lệnh PLF) BẬT cuộn cảm cho một lần quét sau khi điều kiện đầu vào trở thành không được thỏa mãn.

Cuộn cảm được BẬT bởi lệnh PLS/PLF sẽ trở về TẮT sau một lần quét.

Chương trình ladder và vận hành

Mô phỏng các lệnh PLS và PLF bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.



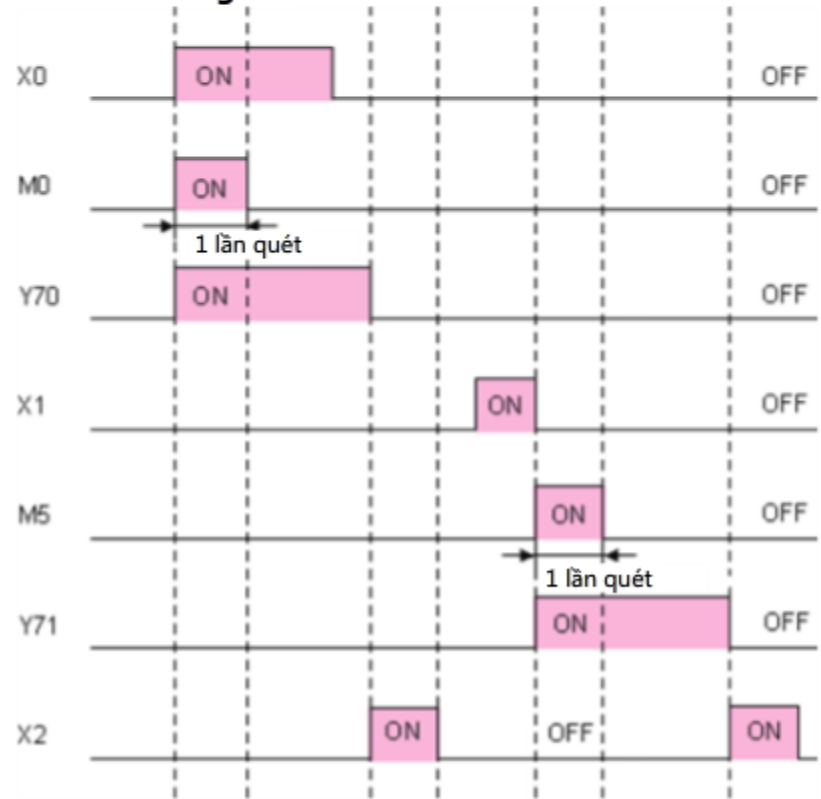
Tại cạnh xung lên X0 (TẮT sang BẬT), M0 BẬT cho 1 lần quét
TẠI cạnh xung xuống X1 (BẬT sang TẮT), M5 BẬT cho 1 lần quét

2.5 Đầu ra dưới dạng xung

Các mã lệnh và chức năng

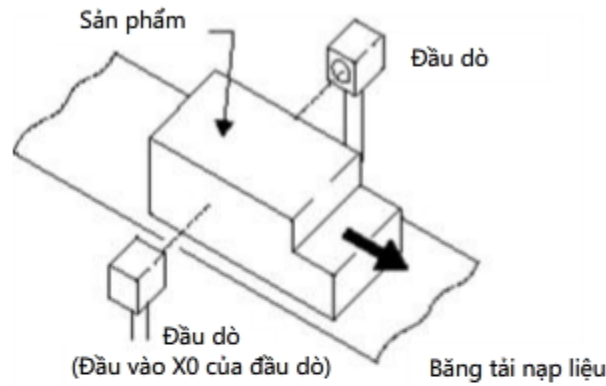
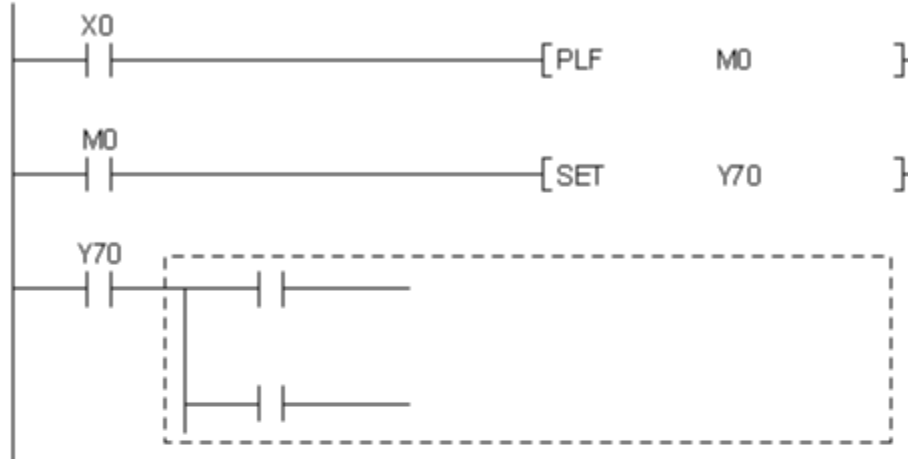
Biểu tượng	Chức năng
	Đầu ra tại cạnh xung lên (PLS) Dữ liệu được chuyển ra một thiết bị được chỉ định ở lần quét đầu tiên sau khi thỏa mãn điều kiện đầu vào.
	Đầu ra tại cạnh xung xuống (PLF) Dữ liệu được chuyển ra một thiết bị được chỉ định ở lần quét đầu tiên sau khi điều kiện đầu vào trở thành không thỏa mãn.

Biểu đồ thời gian



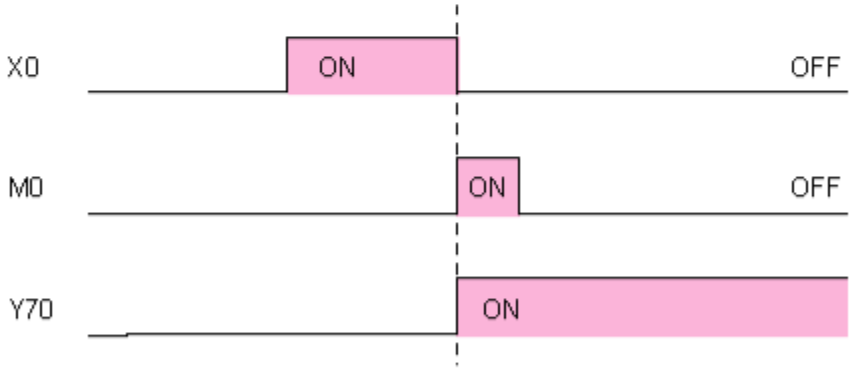
2.5.1 Ví dụ về ứng dụng của ngõ ra xung

Chương trình ladder



Ngõ ra xung được sử dụng để phát hiện dòng chuyển động của các chủ thể. Khi dòng sản phẩm được phát hiện, quy trình tiếp theo sẽ được khởi động.

Biểu đồ thời gian

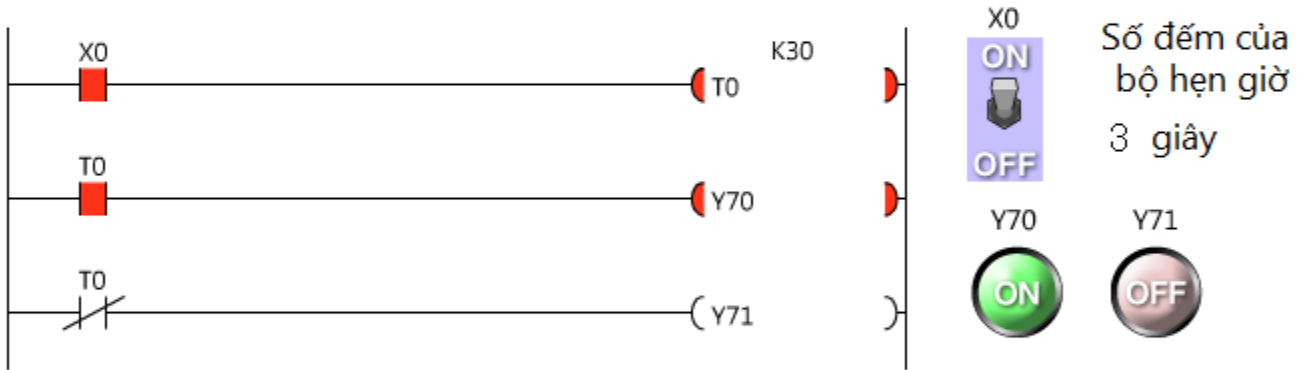


2.6 Đo thời gian

Lệnh OUT và thiết bị bộ hẹn giờ (T) được sử dụng để đo thời gian.
 Khi điều kiện đầu vào được thỏa mãn (BẬT), việc đo thời gian sẽ bắt đầu. Khi khoảng thời gian đạt đến giá trị được chỉ định, thiết bị bộ hẹn giờ (T) sẽ BẬT.
 Nếu điều kiện đầu vào không được thỏa mãn (TẮT) hoặc thiết bị bộ hẹn giờ (T) được cài lại bằng lệnh RST, thời gian trôi qua và đầu ra sẽ được khởi động.
 Có thể sử dụng hiện trạng thiết bị bộ hẹn giờ (T) làm điều kiện đầu vào trong các phần khác của chương trình.

Chương trình ladder và vận hành

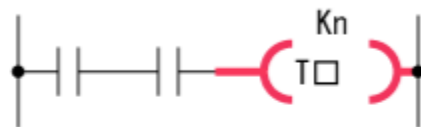
Mô phỏng vận hành bộ hẹn giờ bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.



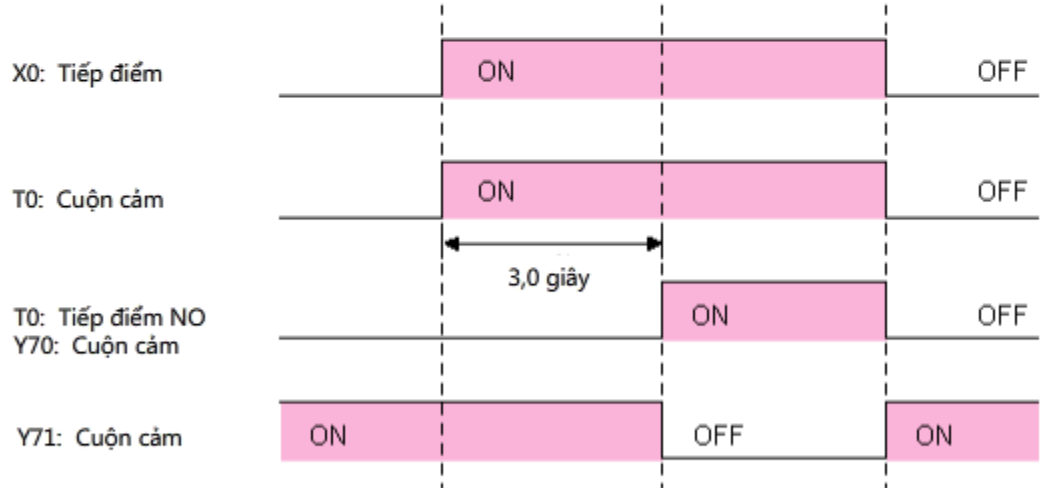
X0 BẬT, tiếp theo, sau 3 giây, Y70 BẬT và Y71 TẮT.

2.6 Đo thời gian

■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
 <p>□ : Số hiệu thiết bị</p>	<p>Vận hành bộ hẹn giờ</p> <p>Thiết bị bộ hẹn giờ (T) được sử dụng với một đầu ra cuộn cảm (OUT) để đo khoảng thời gian thỏa mãn điều kiện (đang BẬT).</p> <p>Việc hết thời gian cho phép sẽ xảy ra sau khoảng thời gian được quy định.</p> <p>Đồng thời khi hết thời gian cho phép, bộ hẹn giờ (T0) sẽ BẬT.</p> <p>Giá trị được cài cho bộ hẹn giờ được biểu thị bằng "Kn" (n: thập phân).</p> <p>Các bộ hẹn giờ thường được sử dụng làm bộ định thời độ trễ quá trình mở, quy định thời gian sau khi thỏa mãn một điều kiện nhất định.</p>

■ Biểu đồ thời gian

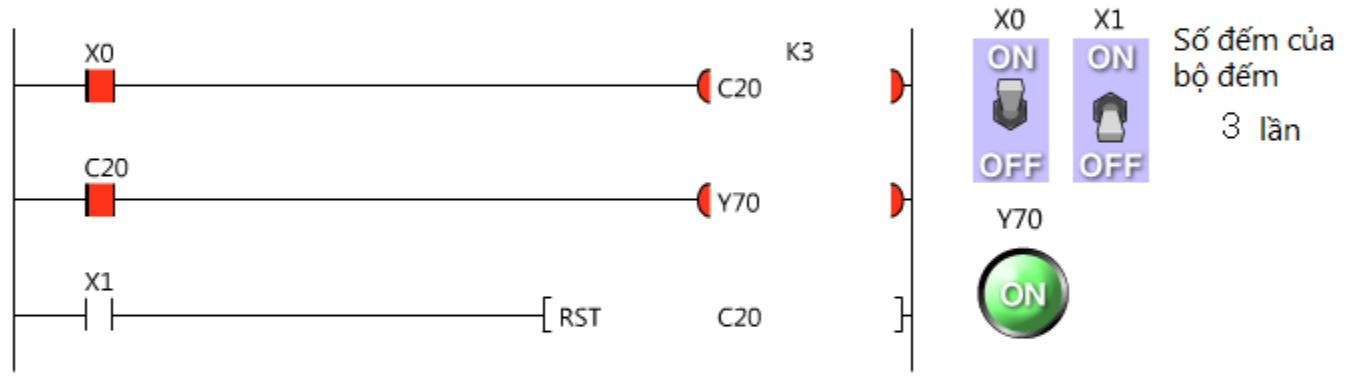


2.7 Đếm

Lệnh OUT và thiết bị bộ đếm (C) được sử dụng để đếm.
 Khi một điều kiện đầu vào được thỏa mãn, số đếm sẽ tăng lên và khi số đếm đạt đến một giá trị được quy định, thiết bị bộ đếm được quy định (C) sẽ BẬT.
 Nếu thiết bị bộ đếm (C) được đặt lại bằng lệnh RST, số đếm và hiện trạng thiết bị sẽ được khởi tạo.
 Có thể sử dụng hiện trạng thiết bị bộ đếm (C) làm điều kiện đầu vào trong các phần khác của chương trình.

Chương trình ladder và vận hành

Mô phỏng vận hành bộ đếm bằng cách nhấp chuột vào công tắc đầu vào hiển thị ở bên phải.



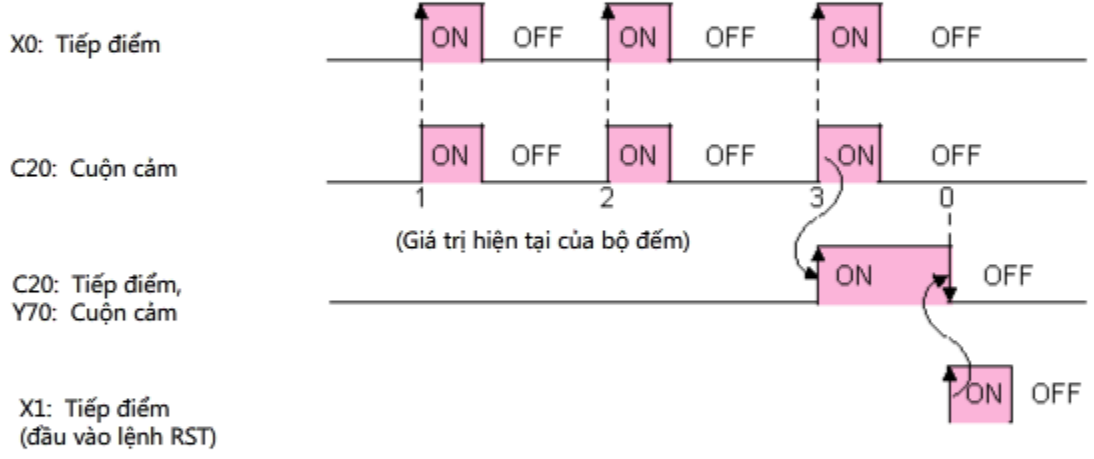
Giá trị trong C20 được tăng lên mỗi lần X0 BẬT. Khi số đếm đạt tới 3 (đếm lần lượt), Y70 sẽ BẬT.

2.7 Đếm

Các mã lệnh và chức năng

Biểu tượng	Chức năng
<p>□ : Số hiệu thiết bị</p>	<p>Bộ đếm Kết hợp với một đầu ra cuộn cảm (OUT), bộ đếm sẽ đếm thêm (lần lượt) đến số lần thỏa mãn điều kiện. Việc đếm lần lượt được thực hiện khi số đếm đạt đến số được chỉ định và tiếp điểm bộ đếm BẬT. Giá trị được cài cho bộ đếm được biểu thị bằng "Kn" (n: thập phân).</p>

Biểu đồ thời gian



Chương 3 Các lệnh thiết bị từ

Chương này giải thích về các lệnh sử dụng các thiết bị từ.

Các thiết bị từ được sử dụng trong kiểm soát thời gian, số đếm và các giá trị nhập vào từ thiết bị ngoại vi.

Các thiết bị từ có thể làm cho các chương trình điều khiển phản hồi nhạy hơn với vận hành thực tế.

- Mô phỏng các vận hành chương trình cơ bản để hiểu cách vận hành các lệnh chính
- Từ mô phỏng này, bạn sẽ hiểu được các vai trò của lệnh và việc xử lý được thực hiện trong bộ điều khiển khả trình

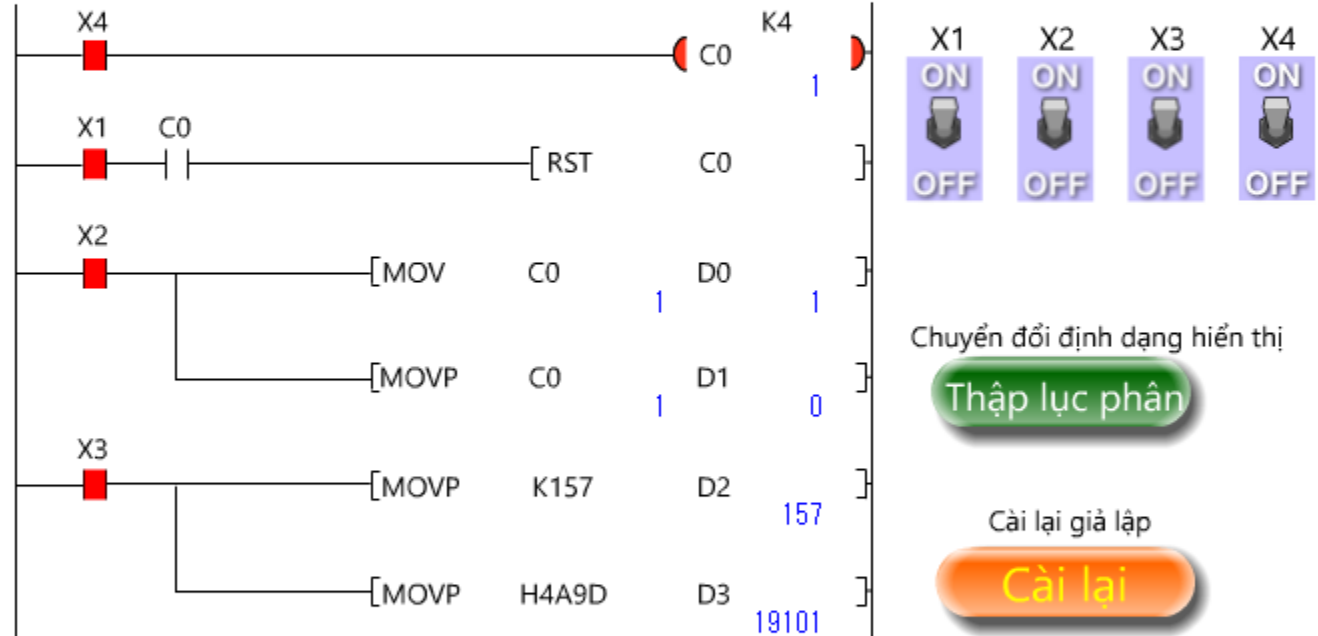
3.1 Chuyển dữ liệu sang thiết bị từ

Lệnh truyền dữ liệu 16 bit (MOV) sẽ chuyển (sao chép) một dữ liệu đơn vị 1 từ (16 bit) sang một thiết bị từ được chỉ định. Dữ liệu có thể chuyển có thể là một giá trị trong thiết bị hoặc có thể được chỉ định. Định dạng dữ liệu có thể chuyển có thể là thập phân hoặc thập lục phân.

Chương trình ladder và vận hành

Mô phỏng vận hành các lệnh sau bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải.

Mỗi số màu lam biểu thị giá trị (giá trị hiện tại) được lưu trong thiết bị.



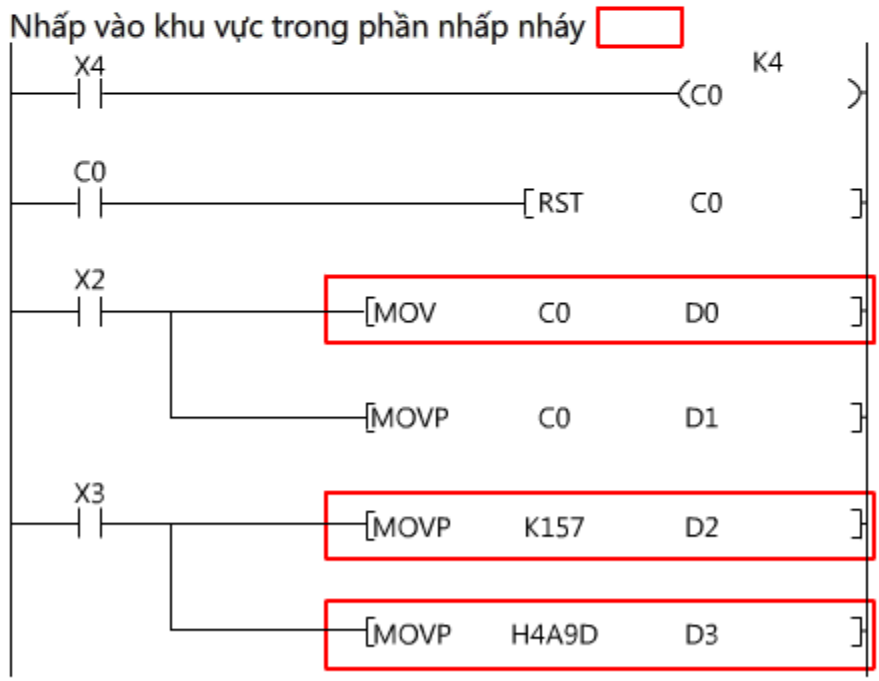
Khi bạn BẬT/TẮT X4 lặp lại nhiều lần, giá trị hiện tại của C0 sẽ tăng lên. (0, 1...4->0...).

3.1 Chuyển dữ liệu sang thiết bị từ

■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	<p>Truyền dữ liệu 16 bit (MOV) Khi điều kiện đầu vào được thỏa mãn, dữ liệu được chỉ định trong nguồn (S) được truyền (sao chép) sang thiết bị được chỉ định tại điểm đến (D).</p>
	<p>Truyền dữ liệu 16 bit (theo xung) (MOVP) Khi điều kiện đầu vào được thỏa mãn (TẮT thành BẬT), dữ liệu được chỉ định trong nguồn (S) được truyền (sao chép) sang thiết bị được chỉ định tại điểm đến (D).</p>

■ Chương trình ladder



3.1.1 Sự khác nhau giữa MOV và MOVP

Lệnh MOV được sử dụng để đọc liên tục các dữ liệu thay đổi. Trong khi đó, lệnh MOVP được sử dụng để truyền dữ liệu chỉ một lần, ví dụ như cài dữ liệu hoặc đọc dữ liệu khi xảy ra lỗi.



MOV

← Thời gian thực hiện →

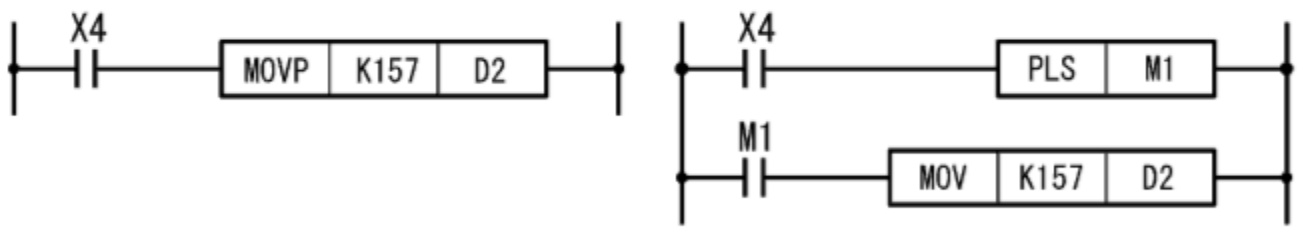
Dữ liệu được truyền mỗi lần quét khi điều kiện đầu vào được thỏa mãn.

MOVP

↔ Được thực hiện chỉ một lần tại cạnh xung lên

Dữ liệu chỉ được truyền tại cạnh xung lên của điều kiện đó. (Thực hiện một lần)

Các hình dưới đây minh họa hai chương trình cùng dẫn đến một vận hành giống nhau với các lệnh MOV và MOVP. Trong cả hai thanh thang, truyền dữ liệu được thực hiện khi X4 BẬT. Với lệnh MOVP, hoạt động có thể được thực hiện mà không sử dụng lệnh PLS quy định thực hiện vận hành cạnh xung lên.

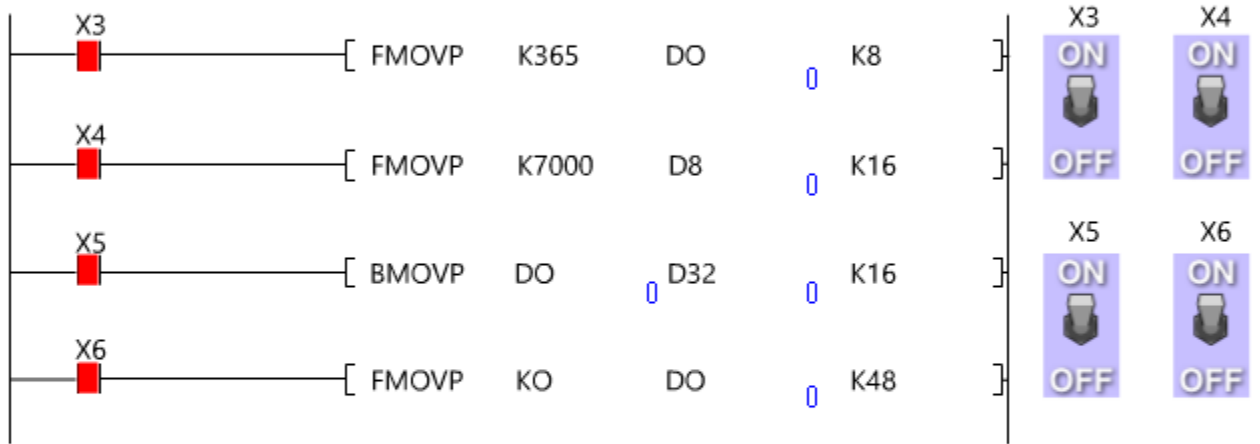


3.1.2 Chuyển dữ liệu sang nhiều thiết bị từ cùng một lúc

Các lệnh MOV/MOVP được sử dụng để truyền dữ liệu tới một thiết bị. Để truyền dữ liệu tới nhiều thiết bị có các số thứ tự liên tiếp, có thể sử dụng "lệnh truyền một loạt dữ liệu giống nhau" (FMOV) hoặc "lệnh truyền khối dữ liệu giống nhau" (BMOV).

Chương trình ladder và vận hành

Mô phỏng vận hành các lệnh sau bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải. Mỗi số màu lam biểu thị giá trị (giá trị hiện tại) được lưu trong thiết bị.



Trình giám sát thiết bị

DO	<input type="text" value="0"/>	D8	<input type="text" value="0"/>	D32	<input type="text" value="0"/>
D1	<input type="text" value="0"/>	D9	<input type="text" value="0"/>	D33	<input type="text" value="0"/>
D7	<input type="text" value="0"/>	D23	<input type="text" value="0"/>	D47	<input type="text" value="0"/>

Khi mỗi tín hiệu đầu vào BẬT, dữ liệu được chỉ định sẽ được truyền ngay lập tức.

LƯU Ý: Trên thanh thang thứ ba bắt đầu bằng X5, dữ liệu được truyền đi bằng lệnh BMOV.

3.1.2 Chuyển dữ liệu sang nhiều thiết bị từ cùng một lúc

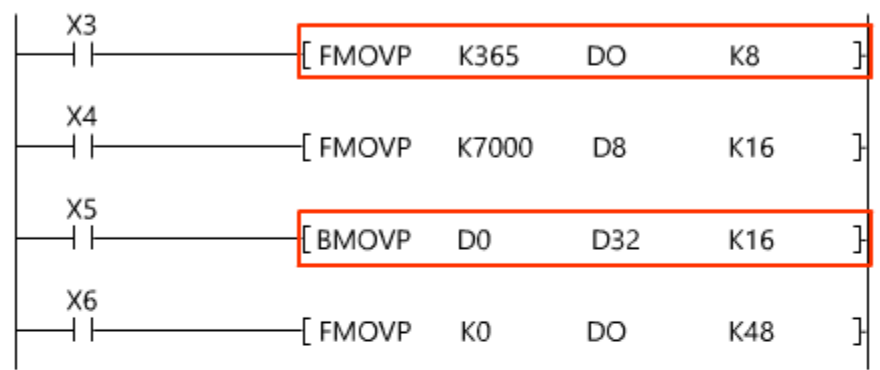
■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	<p>Truyền một loạt dữ liệu giống nhau (FMOV) Khi điều kiện đầu vào được thỏa mãn, dữ liệu được chỉ định trong nguồn (S) sẽ được truyền (sao chép) tới thiết bị được chỉ định bởi điểm đến (D) và tới "n" thiết bị theo sau D.</p>
	<p>Truyền một loạt dữ liệu giống nhau (theo xung) (FMOVP) Khi điều kiện đầu vào được thỏa mãn, dữ liệu được chỉ định trong nguồn (S) sẽ được truyền (sao chép) tới thiết bị được chỉ định bởi điểm đến (D) và tới "n" thiết bị theo sau D.</p>
	<p>Truyền khối dữ liệu giống nhau (BMOV) Khi điều kiện đầu vào được thỏa mãn, dữ liệu được chỉ định trong nguồn (S) và "n" thiết bị tiếp theo sẽ được truyền tới thiết bị được chỉ định bởi thiết bị (D) và tới "n" thiết bị tiếp theo.</p>
	<p>Truyền một loạt khối dữ liệu giống nhau (theo xung) (BMOVP) Tại cạnh xung lên của điều kiện, dữ liệu trong thiết bị do nguồn (S) chỉ định và "n" thiết bị tiếp theo sẽ được truyền tới thiết bị được chỉ định bởi thiết bị (D) và tới "n" thiết bị tiếp theo.</p>

3.1.2 Chuyển dữ liệu sang nhiều thiết bị từ cùng một lúc

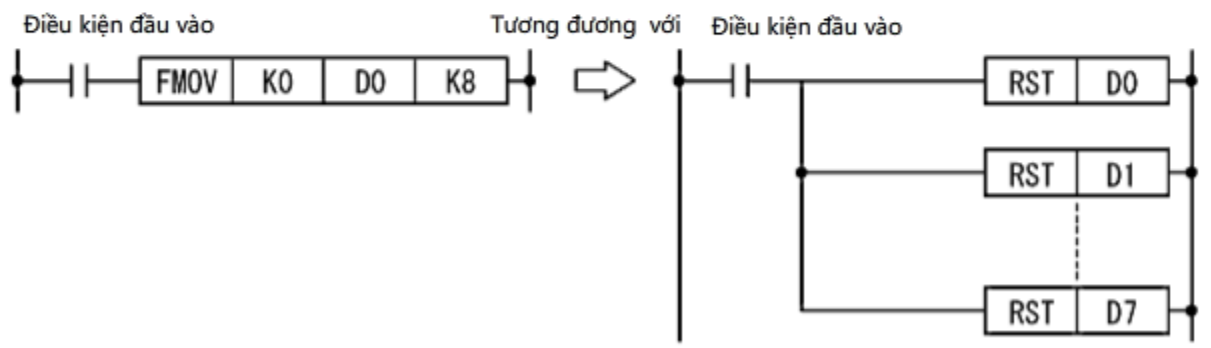
Chương trình ladder và vận hành

Nhấp vào khu vực trong phần nhấp nháy



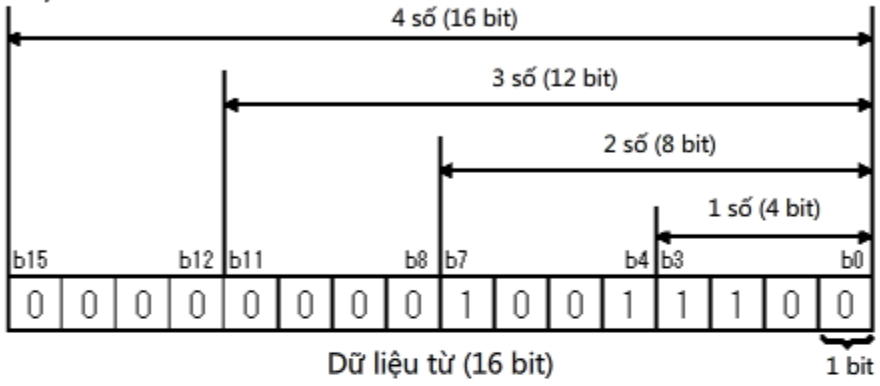
Ứng dụng các lệnh FMOV và BMOV

Lệnh FMOV thuận tiện cho việc xóa một lượng dữ liệu lớn ngay lập tức.



3.1.3 Số thiết bị bit

Bốn thiết bị bit được phân nhóm vào một số thiết bị bit để điều khiển thông tin bit trong một phạm vi nhất định (truyền dữ liệu, v.v).



■ Cách chỉ định một số thiết bị bit

Một số thiết bị bit được biểu thị dưới dạng "số thứ tự của các số" + "số thiết bị bắt đầu". Số thứ tự của số là một bội số của 4. Bảng dưới đây minh họa một số ví dụ. Sau đây là các ví dụ khi số thiết bị bắt đầu là "M0".

Phạm vi bit	Cách chỉ định
Dữ liệu 16 bit	K4M0 (16 bit, M0 đến M15)
Dữ liệu 32 bit	K8M0 (32 bit, M0 đến M31)

Số thiết bị bit (số bit) xác định phạm vi các giá trị số có thể sử dụng.

Số thiết bị bit	Phạm vi các giá trị số có thể sử dụng
K1 (4 bit)	0 đến 15
K2 (8 bit)	0 đến 255
K3 (12 bit)	0 đến 4095
K4 (16 bit)	-32768 đến 32767 Bit số 16 có thể được sử dụng cho ký hiệu dương/âm để biểu thị các giá trị âm.

3.1.3 Các ví dụ về truyền số thiết bị bit

Các lệnh truyền dữ liệu được sử dụng để truyền (sao chép) các số từ nguồn tới thiết bị điểm đến. Các ví dụ sau sẽ minh họa cách truyền dữ liệu được chỉ định.

(a) Thiết bị bit được chỉ định số → Thiết bị từ

Ví dụ) MOV K1X0 D0

(b) Thiết bị từ → Thiết bị bit được chỉ định số

Ví dụ) MOV D0 K2M0

(c) Các hằng số (các số được chỉ định trực tiếp) → Thiết bị bit được chỉ định số

Ví dụ) MOV H1234 K2M0

(d) Thiết bị bit được chỉ định số → Thiết bị bit được chỉ định số

Ví dụ) MOV K1X0 K2M0

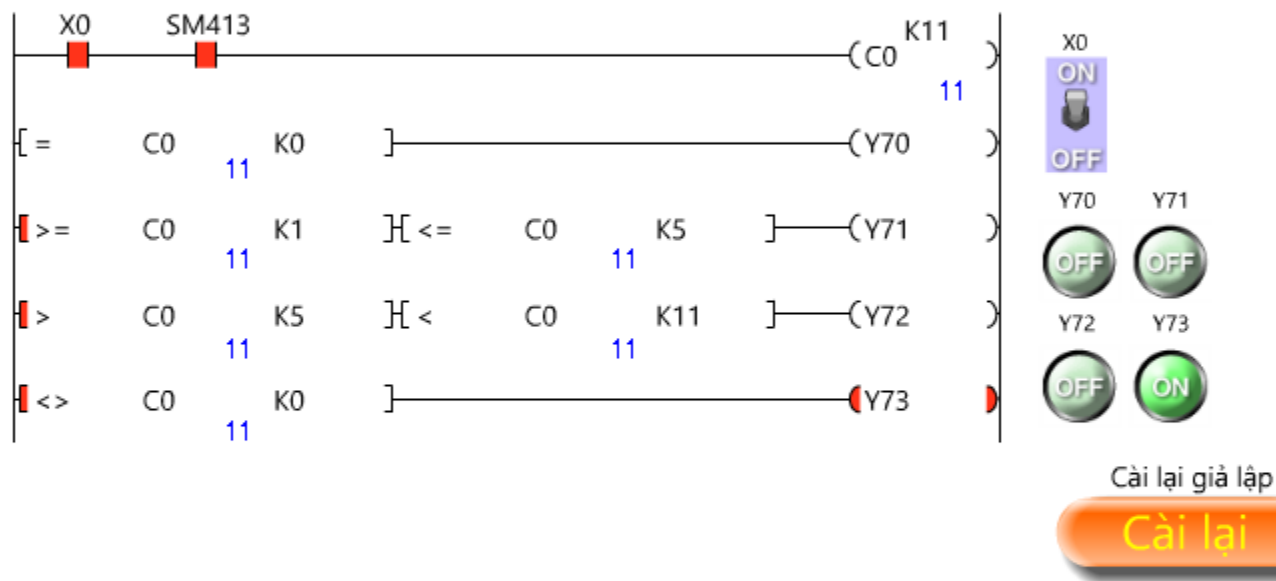
3.2

So sánh các giá trị số

Các lệnh vận hành so sánh được sử dụng để so sánh dữ liệu đơn vị-từ và dữ liệu được lưu trong các thiết bị từ. Khi một điều kiện (+) được thỏa mãn, lệnh tiếp theo sẽ được thực hiện.

Chương trình ladder và vận hành

Mô phỏng vận hành các lệnh sau bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải. Mỗi số màu lam biểu thị giá trị (giá trị hiện tại) được lưu trong thiết bị.



Y70 đến Y73 BẬT/TẮT phụ thuộc vào giá trị hiện tại của C0.

SM413 là một rờ le đặc biệt, nó được mô đun CPU BẬT hoặc TẮT với khoảng cách 1 giây. (2 giây xung đồng hồ) Khi X0 BẬT, C0 sẽ đếm thêm mỗi 2 giây.

* SM413 là một rờ le đặc biệt, nó BẬT hoặc TẮT với khoảng cách 1 giây (2 giây xung đồng hồ). Có thể sử dụng SM403 cho MELSEC Sê-ri iQ-R/Q/L/iQ-F. MELSEC-Sê-ri F không có rờ le 2 giây xung đồng hồ nhưng có M8011 (0,01 giây xung đồng hồ), M8012 (0,1 giây xung đồng hồ), M8013 (1 giây xung đồng hồ) và M8014 (1 phút xung đồng hồ).

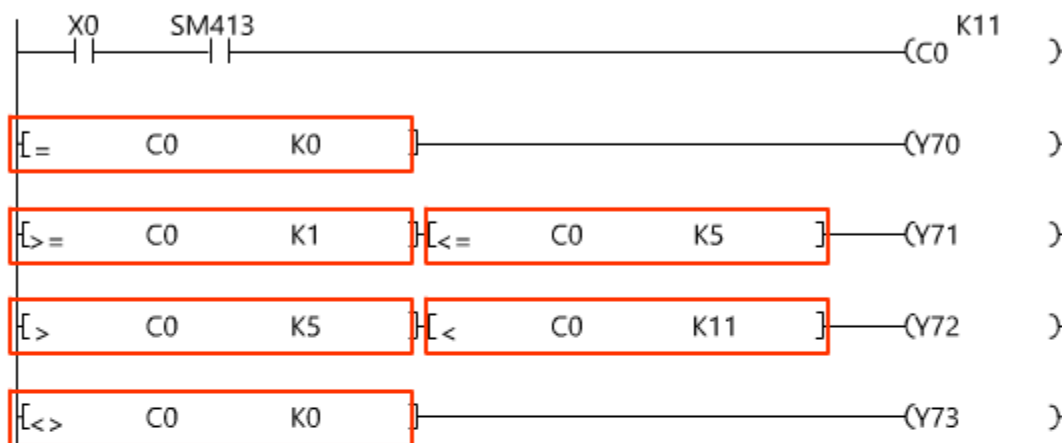
3.2 So sánh các giá trị số

■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	<p>So sánh dữ liệu nhị phân 16 bit. (=) Điều kiện được thỏa mãn khi NGUỒN 1 bằng NGUỒN 2.</p>
	<p>So sánh dữ liệu nhị phân 16 bit. (<) Điều kiện được thỏa mãn khi NGUỒN 1 NHỎ HƠN NGUỒN 2.</p>
	<p>So sánh dữ liệu nhị phân 16 bit. (>) Điều kiện được thỏa mãn khi NGUỒN 1 lớn hơn NGUỒN 2.</p>
	<p>So sánh dữ liệu nhị phân 16 bit. (<=) Điều kiện được thỏa mãn khi NGUỒN 1 bằng hoặc nhỏ hơn NGUỒN 2.</p>
	<p>So sánh dữ liệu nhị phân 16 bit. (>=) Điều kiện được thỏa mãn khi NGUỒN 1 bằng hoặc lớn hơn NGUỒN 2.</p>
	<p>So sánh dữ liệu nhị phân 16 bit. (<>) Điều kiện được thỏa mãn khi NGUỒN 1 không bằng NGUỒN 2.</p>

■ Chương trình ladder và vận hành

Nhấp vào khu vực trong phần nhấp nháy



SM413 là một rờ le đặc biệt, nó được mô đun CPU BẬT hoặc TẮT với khoảng cách 1 giây (2 giây xung đồng hồ). Rờ le đặc biệt (SM) là các thiết bị rờ le trong mô đun CPU. Mỗi rờ le đặc biệt thực hiện một vai trò nhất định.

3.3**Các thao tác thuật toán**

Phần này sẽ trình bày về các thao tác thuật toán cơ bản của thiết bị từ (số).

■ Cộng và trừ

Các thao tác thuật toán sử dụng các ký hiệu cộng (+) và trừ (-).

■ Nhân và chia

Các thao tác thuật toán sử dụng các ký hiệu nhân (*) và chia (/).

Các lệnh giữa MELSEC Sê-ri iQ-R/Q/L/iQ-F và MELSEC-Sê-ri F là khác nhau nhưng về khái niệm cơ bản thì giống nhau.

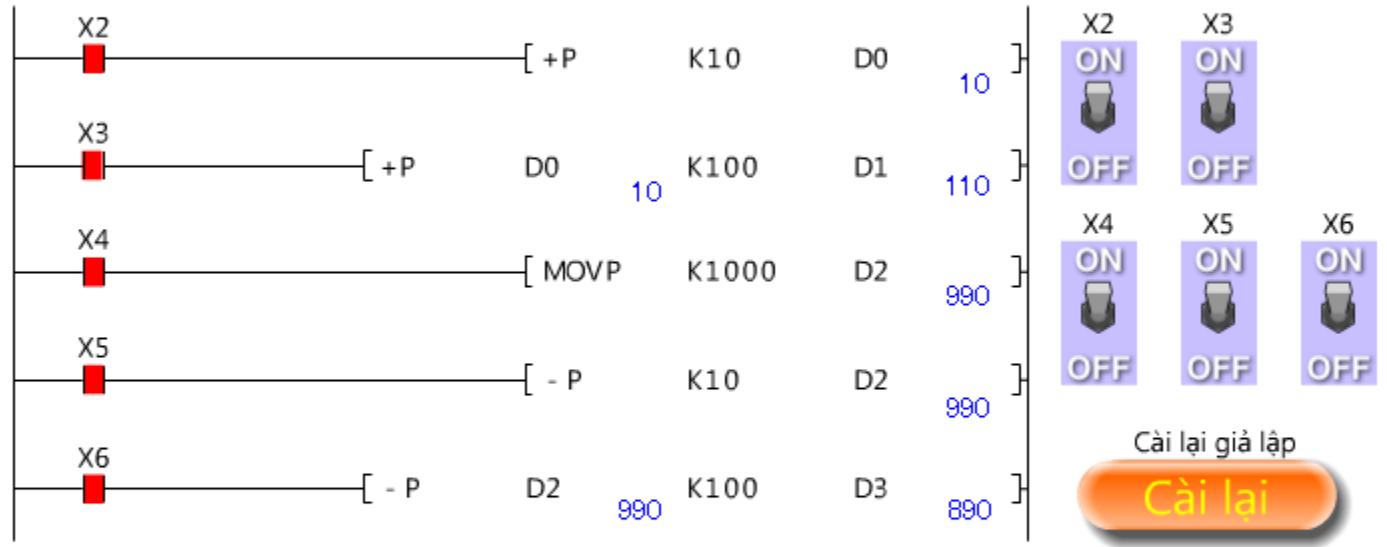
Phần này sẽ trình bày dựa trên các lệnh được dùng trong MELSEC Sê-ri iQ-R/Q/L /iQ-F.

3.3.1 Cộng và trừ

Sơ đồ dưới đây minh họa các lệnh thực hiện cộng và trừ và lưu giá trị đạt được trong các thiết bị được chỉ định.

Chương trình ladder và vận hành

Mô phỏng vận hành các lệnh sau bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải. Mỗi số màu lam biểu thị giá trị (giá trị hiện tại) được lưu trong thiết bị.

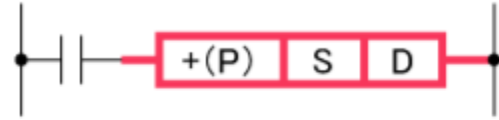
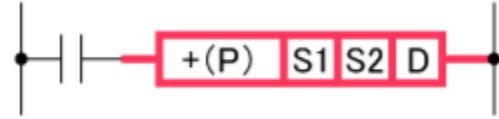
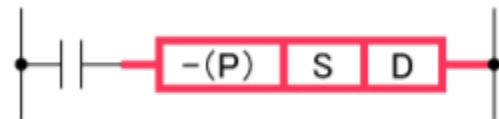
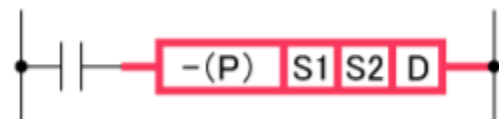


Khi mỗi tín hiệu đầu vào BẬT, thao tác thuật toán sẽ được thực hiện.

* Ví dụ dựa trên MELSEC Sê-ri iQ-R/Q/L/iQ-F.

3.3.1 Cộng và trừ

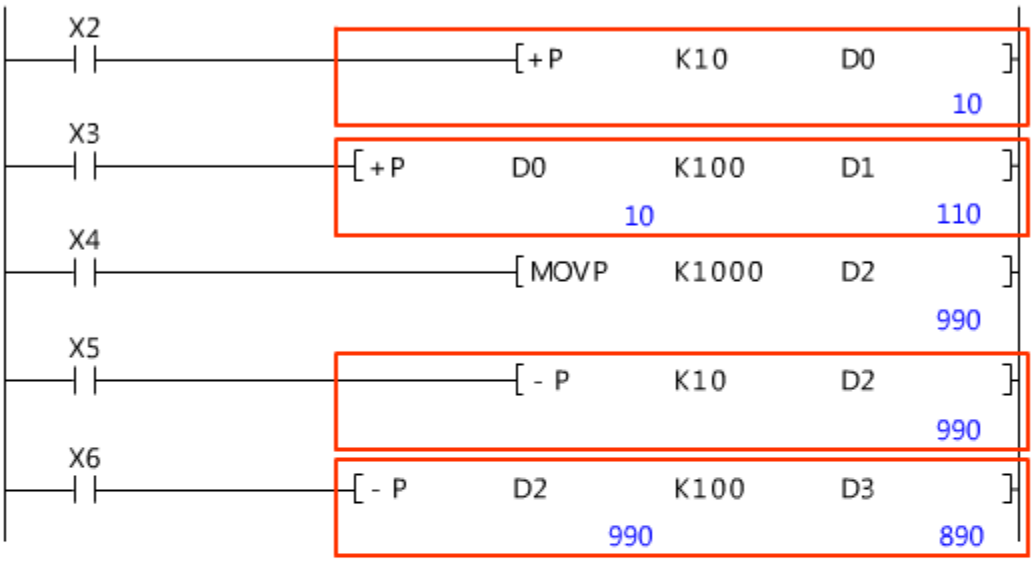
■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
	<p>Thêm dữ liệu nhị phân 16 bit</p> <ul style="list-style-type: none"> + S D : Thực hiện thao tác "D + S = D". + S1 S2 D : Thực hiện thao tác "S1 + S2 = D".
	
	<p>Trừ dữ liệu nhị phân 16 bit</p> <ul style="list-style-type: none"> + S D : Thực hiện thao tác "D - S = D". - S1 S2 D : Thực hiện thao tác "S1 - S2 = D".
	

3.3.1 Cộng và trừ

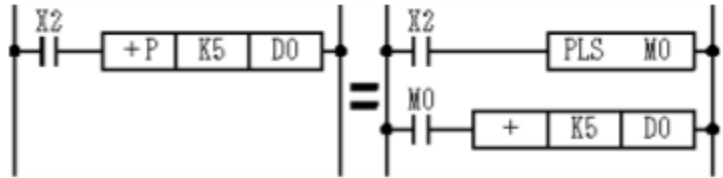
Chương trình ladder và vận hành

Nhấp vào khu vực trong phần nhập nháy



Lưu ý về các lệnh cộng và trừ

Trong các điều kiện bình thường, hãy sử dụng lệnh +P/-P để thực hiện cộng/trừ.
 Nếu sử dụng lệnh +/-, cộng/trừ sẽ được thực hiện lặp lại nhiều lần trong khi điều kiện đầu vào được thỏa mãn.
 Với một trong các thanh thang sau đây, thao tác cộng chỉ được thực hiện một lần khi X2 BẬT.



* Ví dụ dựa trên MELSEC Sê-ri iQ-R/Q/L/iQ-F.

3.3.2

Nhân và chia

Sơ đồ dưới đây minh họa các lệnh thực hiện nhân và chia và lưu giá trị đạt được trong các thiết bị được chỉ định.

■ Chương trình ladder và vận hành

Hãy mô phỏng vận hành các lệnh sau bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải. Mỗi số màu lam biểu thị giá trị (giá trị hiện tại) được lưu trong thiết bị.



Khi mỗi tín hiệu đầu vào BẬT, thao tác thuật toán sẽ được thực hiện.

* Ví dụ dựa trên MELSEC Sê-ri iQ-R/Q/L/iQ-F.

3.3.2

Nhân và chia



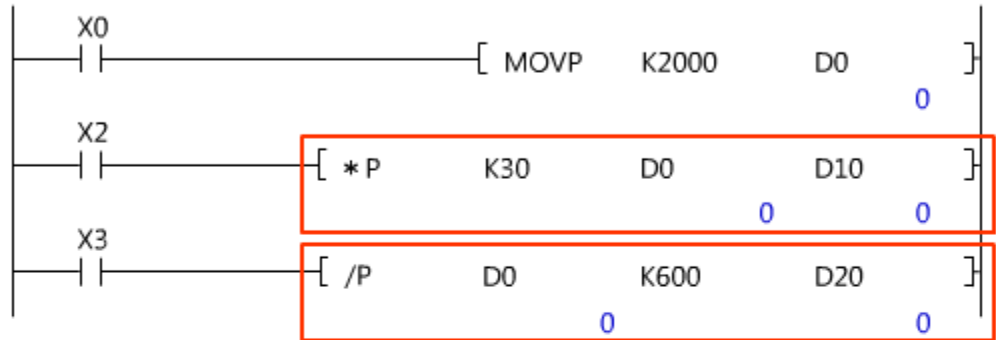
■ Các mã lệnh và chức năng

Biểu tượng	Chức năng
<p>The diagram shows an instruction format for multiplication. It consists of a 1-bit flag, a 2-bit register field (P), a 4-bit source register field (S1), a 4-bit source register field (S2), and a 4-bit destination register field (D). The entire instruction is enclosed in a red box.</p>	<p>Nhân dữ liệu nhị phân 16 bit (*) Thực hiện thao tác "$S1 \times S2 = (D+1 D)$". ("D+1" là thiết bị đứng sau D. Nếu D là D100 thì "D+1" là D101). Kết quả của thao tác là dữ liệu 32 bit, gồm có 2 đơn vị từ ("D" và "D+1").</p>
<p>The diagram shows an instruction format for division. It consists of a 1-bit flag, a 2-bit register field (P), a 4-bit source register field (S1), a 4-bit source register field (S2), and a 4-bit destination register field (D). The entire instruction is enclosed in a red box.</p>	<p>Chia dữ liệu nhị phân 16 bit Thực hiện thao tác "$S1/S2 = (D \text{ [thương số], } D + 1 \text{ [số dư])}$". ("D + 1" là thiết bị đứng sau D. Nếu D là D100 thì "D + 1" là D101). Kết quả của thao tác là một số nguyên.</p>

3.3.2 Nhân và chia

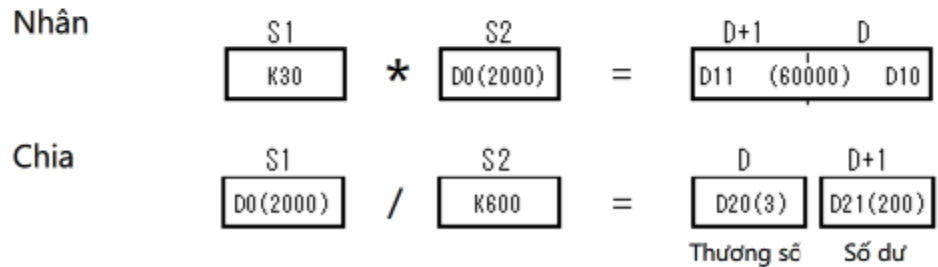
Chương trình ladder và vận hành

Nhấp vào khu vực trong phần nhập nhảy



Lưu ý về các lệnh nhân và chia

Để thực hiện lệnh nhân hoặc chia, hai thiết bị từ liên tiếp (D, D+1) được yêu cầu cho điểm đến (D).



* Ví dụ dựa trên MELSEC Sê-ri iQ-R/Q/L/iQ-F.

3.3.3 Sự khác nhau giữa MELSEC iQ-R/Q/L/iQ-F và MELSEC-F

Các biểu tượng của MELSEC Sê-ri iQ-R/Q/L/iQ-F khác với biểu tượng của MELSEC-Sê-ri F. Bảng dưới đây tóm tắt những điểm khác nhau chính này.

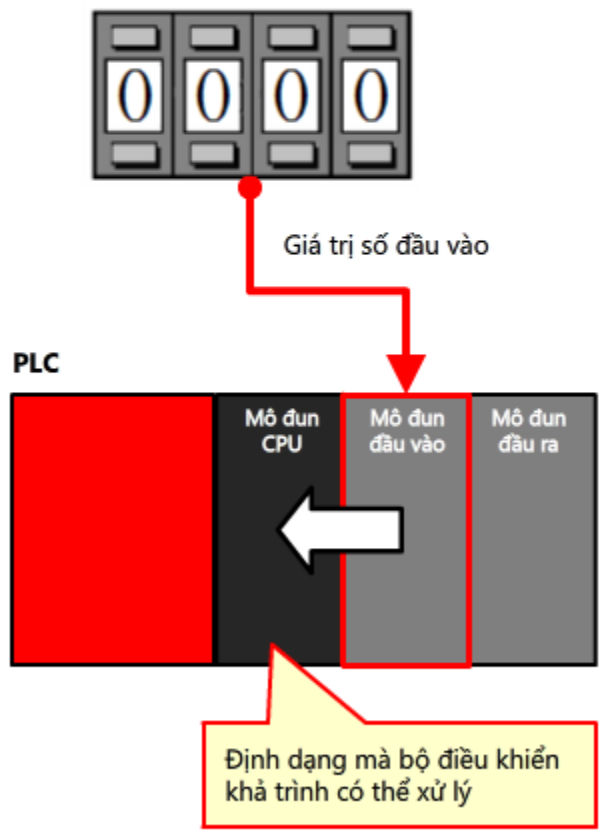
Thao tác thuật toán	Lệnh được sử dụng trong MELSEC Sê-ri iQ-R/Q/L/iQ-F	Lệnh được sử dụng trong MELSEC-Sê-ri F	Sự khác nhau
Cộng (+)			MELSEC Sê-ri iQ-R/Q/L/iQ-F: +(P) MELSEC-Sê-ri F: ADD(P)
Trừ (-)			MELSEC Sê-ri iQ-R/Q/L/iQ-F: -(P) MELSEC-Sê-ri F: SUB(P)
Nhân (*)			MELSEC Sê-ri iQ-R/Q/L/iQ-F: *(P) MELSEC-Sê-ri F: MUL(P)
Chia (/)			MELSEC Sê-ri iQ-R/Q/L/iQ-F: /(P) MELSEC-Sê-ri F: DIV(P)

3.4 Truyền/nhận dữ liệu giữa PLC và thiết bị I/O

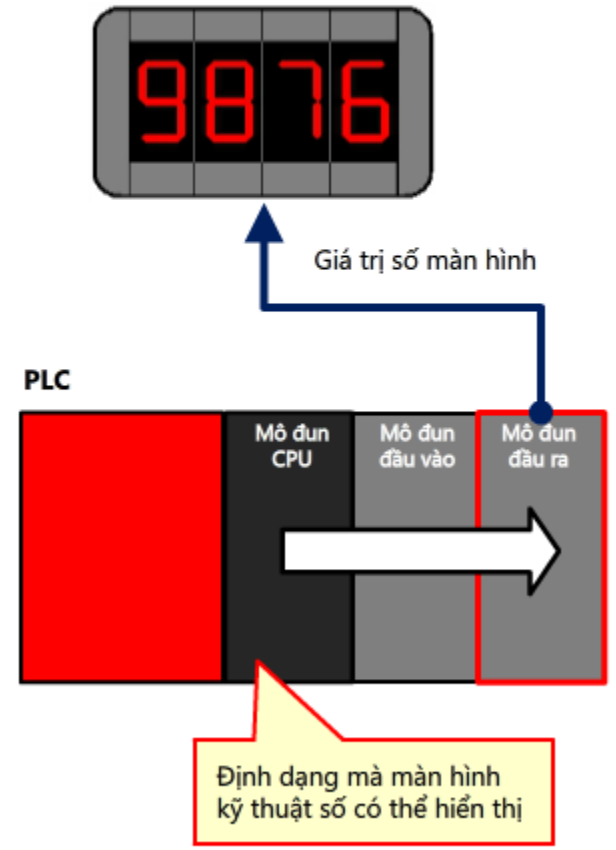
Công tắc đầu vào kỹ thuật số là thiết bị đầu vào giúp đưa dữ liệu vào bộ điều khiển khả trình dưới dạng giá trị số. Màn hình kỹ thuật số là thiết bị đầu ra hiển thị dữ liệu nhận được từ bộ điều khiển khả trình dưới dạng giá trị số.

Dữ liệu nhận được từ công tắc đầu vào kỹ thuật số phải được định dạng để bộ điều khiển khả trình có thể xử lý. Tương tự, đầu ra dữ liệu đến màn hình kỹ thuật số phải được định dạng thành dạng màn hình kỹ thuật số có thể đọc được.

Công tắc đầu vào kỹ thuật số



Màn hình kỹ thuật số



3.4.1 Nhận dữ liệu vào từ công tắc đầu vào kỹ thuật số

Để bộ điều khiển khả trình nhận được dữ liệu vào từ công tắc đầu vào kỹ thuật số, cần phải sử dụng lệnh BIN.

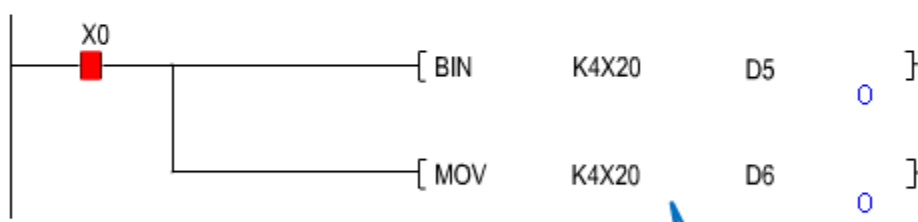
Các mã lệnh và chức năng

Biểu tượng	Chức năng
	Dữ liệu trong thiết bị (S) được định dạng thành một định dạng mà bộ điều khiển khả trình có thể xử lý được, sau đó được lưu vào thiết bị (D).

Chương trình ladder và vận hành

Mô phỏng vận hành các lệnh sau bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải. Mỗi số màu lam biểu thị giá trị (giá trị hiện tại) được lưu trong thiết bị.

D5 giữ lại dữ liệu đã nhận từ công tắc đầu vào kỹ thuật số sau khi dữ liệu được định dạng bằng lệnh BIN.
D6 giữ lại dữ liệu chưa định dạng nhận được từ công tắc đầu vào kỹ thuật số.



Công tắc ngõ vào số mã

<- Độ suy giảm

<- Bước tăng

X2F đến X20

Nếu dùng lệnh MOV, các số sẽ không khớp.

Cài lại giá lập
Cài lại

3.4.2 Hiển thị dữ liệu PLC trên màn hình kỹ thuật số

Để hiển thị dữ liệu của bộ điều khiển khả trình trên màn hình kỹ thuật số, cần dùng lệnh BCD.

Các mã lệnh và chức năng

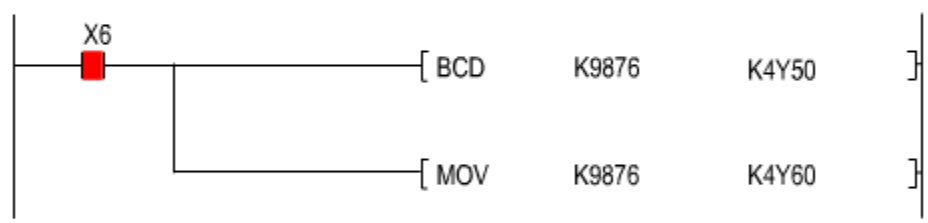
Biểu tượng	Chức năng
	Dữ liệu trong thiết bị (S) được định dạng thành định dạng có thể hiển thị trên màn hình kỹ thuật số, sau đó được lưu vào thiết bị (D).

Chương trình ladder và vận hành

Mô phỏng vận hành các lệnh sau bằng cách nhấp chuột vào các công tắc đầu vào hiển thị ở bên phải.

Màn hình kỹ thuật số trên cùng đang hiển thị dữ liệu được định dạng bằng lệnh BCD.

Màn hình kỹ thuật số ở cuối đang hiển thị dữ liệu chưa định dạng.



Nếu dùng lệnh MOV, các số sẽ không khớp.

Cài lại giá lập
Cài lại

Trong khóa học này, bạn đã tìm hiểu về:

- Khái niệm đầu vào và đầu ra đến/từ các bộ điều khiển khả trình
- Các lệnh chính điều khiển các bộ điều khiển khả trình
- Thông tin được bộ điều khiển khả trình MELSEC nhận sẽ được thực hiện trong các chương trình ladder trong bộ điều khiển khả trình đó và các kết quả thực hiện sẽ được truyền ra bên ngoài dưới dạng các đầu ra
- Sự khác nhau giữa các định dạng dữ liệu bit và từ
- Những nội dung cơ bản trong các chương trình điều khiển

Vui lòng tham gia khóa học "Engineering Software MELSOFT GX Works3 (Ladder)" (Phần mềm thiết kế MELSOFT GX Works3 (dạng bậc thang (Ladder))) để biết cách chỉnh sửa và đăng ký các chương trình với mô đun CPU MELSEC Sê-ri iQ-R/iQ-F.

Vui lòng tham gia khóa học "Khóa học Cơ Bản về GX Works2" để biết cách chỉnh sửa và đăng ký các chương trình với mô đun CPU MELSEC-Sê-ri Q/L/F.

Kiểm tra Bài kiểm tra cuối khóa

Bây giờ bạn đã hoàn thành tất cả các bài học trong khóa học **Cơ bản về lập trình PLC**, bạn đã sẵn sàng tham gia bài kiểm tra cuối khóa. Nếu bạn không rõ về bất cứ chủ đề nào được trình bày, vui lòng nhân cơ hội này xem xét lại các chủ đề đó.

Có tổng cộng 11 câu hỏi (54 mục) trong Bài kiểm tra cuối khóa này.

Bạn có thể làm bài kiểm tra cuối khóa nhiều lần tùy thích.

Làm thế nào ghi điểm bài kiểm tra

Sau khi chọn câu trả lời, hãy chắc chắn đã nhấp vào nút **Trả lời**. Câu trả lời của bạn sẽ bị mất nếu bạn tiếp tục mà không nhấp vào nút Trả lời. (Coi như là câu hỏi chưa được trả lời.)

Kết quả điểm số

Số lượng câu trả lời đúng, số lượng câu hỏi, tỷ lệ câu trả lời đúng, và kết quả đạt/hỏng sẽ xuất hiện trên trang điểm số.

Câu trả lời đúng : **5**

Tổng số câu hỏi : **5**

Tỷ lệ phần trăm : **100%**

Để vượt qua bài kiểm tra, bạn phải trả lời đúng **60%** các câu hỏi.

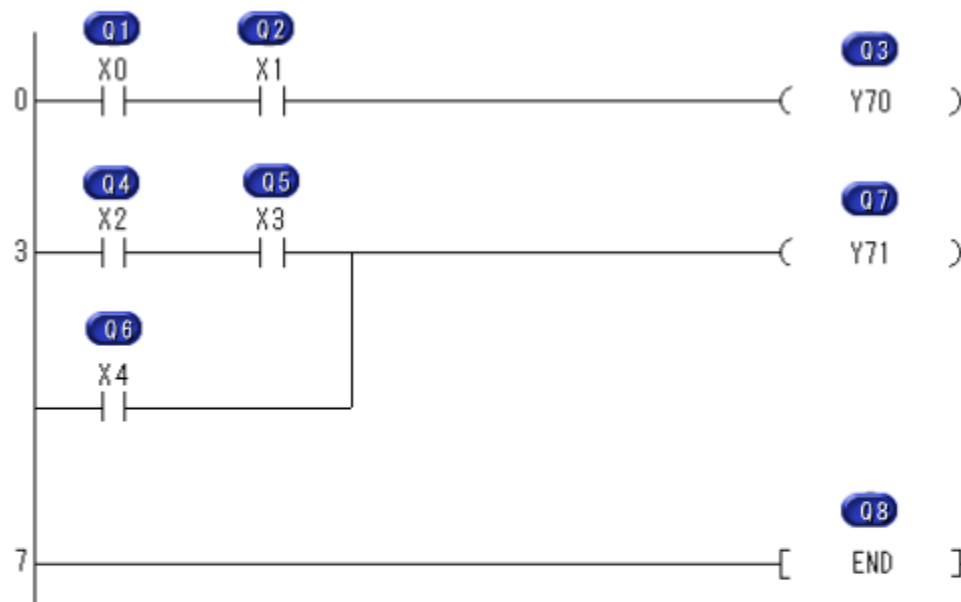
Tiếp tục

Xem lại

- Nhấp vào nút **Tiếp tục** để thoát khỏi bài kiểm tra.
- Nhấp vào nút **Xem lại** để xem lại bài kiểm tra. (Kiểm tra câu trả lời đúng)
- Nhấp vào nút **Thử lại** để làm lại bài kiểm tra một.

Kiểm tra Bài kiểm tra cuối khóa 1

Vui lòng đánh số các lệnh sau theo trình tự xử lý.



Q1 --Select-- ▾ Q2 --Select-- ▾ Q3 --Select-- ▾ Q4 --Select-- ▾

Q5 --Select-- ▾ Q6 --Select-- ▾ Q7 --Select-- ▾ Q8 --Select-- ▾

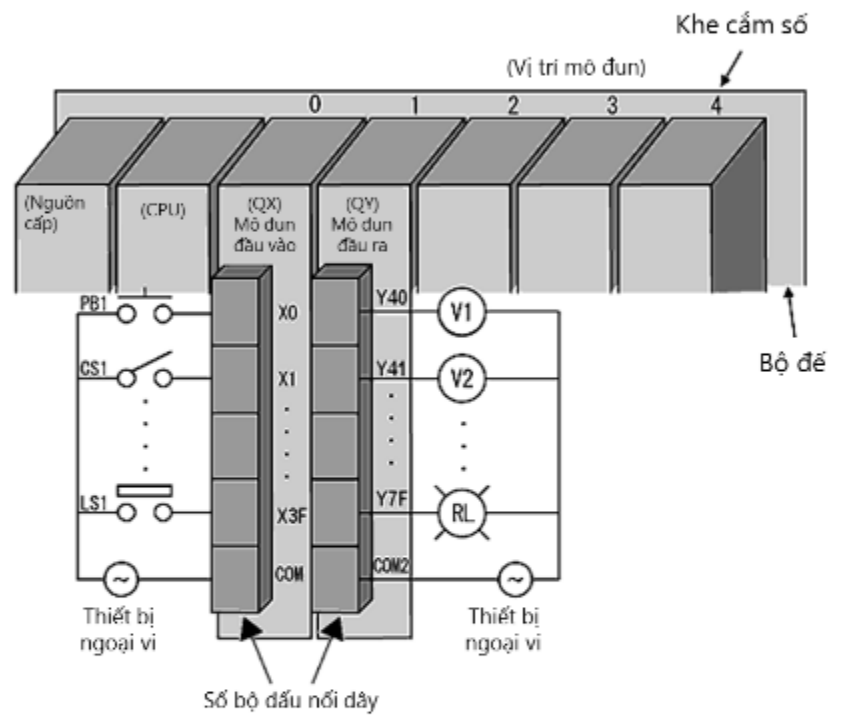
Trả lời

Quay lại

Kiểm tra Bài kiểm tra cuối khóa 2

Các câu dưới đây mô tả thiết bị I/O và tín hiệu I/O bên ngoài đến/từ các bộ điều khiển khả trình. Vui lòng hoàn thành các câu bằng cách chọn từ đúng.

- 1) Các số đầu vào và đầu ra cho các bộ điều khiển khả trình Sê-ri Q bắt đầu từ (--Select--) và nằm trong các giá trị (--Select--).
- 2) Các tín hiệu đầu vào và đầu ra được dùng các số giống nhau. Do đó, đầu vào có phần đầu là (--Select--) và đầu ra có phần đầu là (--Select--).
- 3) Các số được gán cho các tín hiệu được nhập vào từ thiết bị ngoại vi được xác định theo các điều kiện sau:
 - Vị trí trong bộ đế mà (--Select--) được lắp đặt trên
 - Số bộ đấu nối dây
- 4) Các số được gán cho các đầu ra (cuộn cảm) từ thiết bị ngoại vi được xác định theo các điều kiện sau:
 - Vị trí trong bộ đế mà (--Select--) được lắp đặt trên
 - Số bộ đấu nối dây



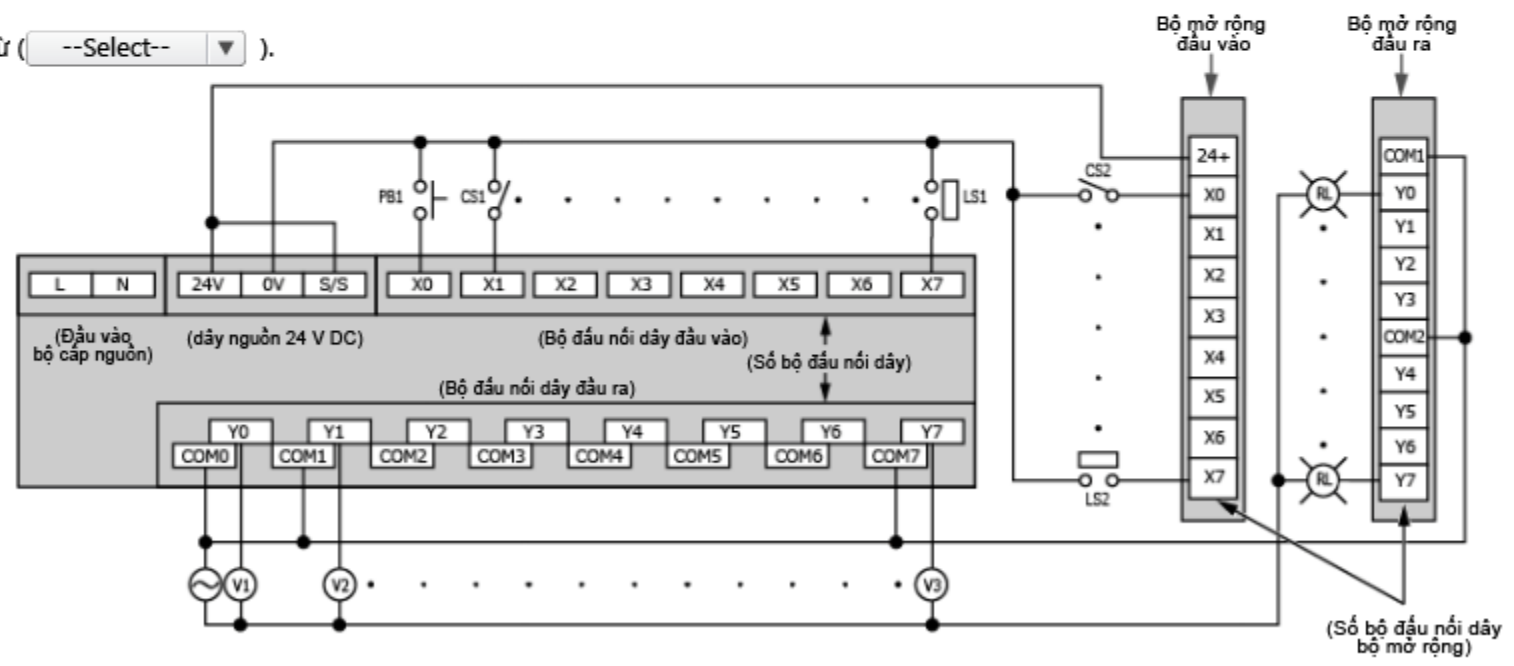
Trả lời

Quay lại

Kiểm tra Bài kiểm tra cuối khóa 3

Các câu dưới đây mô tả thiết bị I/O và số thứ tự I/O bên ngoài được chỉ định cho các bộ điều khiển khả trình. Vui lòng hoàn thành các câu bằng cách chọn từ đúng. (MELSEC-Sê-ri F)

- 1) Số thứ tự I/O cho các bộ điều khiển khả trình MELSEC-Sê-ri F bắt đầu từ (--Select--) và nằm trong các giá trị (--Select--).
- 2) Các tín hiệu đầu vào và đầu ra được dùng các số giống nhau. Do đó, đầu vào có phần đầu là (--Select--) và đầu ra có phần đầu là (--Select--).
- 3) Nếu bộ mở rộng I/O được thêm vào, bộ đó sẽ được gán một số theo sau số được gán cho trước đó (--Select--).
- 4) Số thứ tự I/O của một bộ sẽ bắt đầu từ số có số "0" trong chữ số đầu tiên. Số thứ tự I/O của bộ trước đó kết thúc tại X17, số thứ tự I/O của bộ tiếp theo sẽ bắt đầu từ (--Select--).

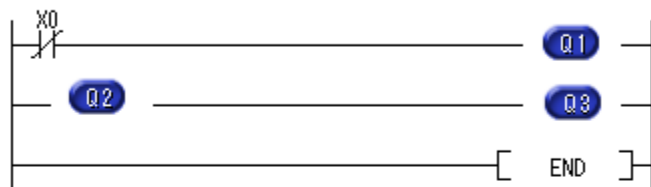


Kiểm tra Bài kiểm tra cuối khóa 4

Kéo và thả các lệnh phù hợp để hoàn thành chương trình thực hiện các vận hành sau:

Khi công tắc X0 TẮT, đèn A BẬT. (Y70 BẬT)

Khi công tắc BẬT, đèn B BẬT. (Y71 BẬT)



Q1 --Select-- ▼ Q2 --Select-- ▼ Q3 --Select-- ▼

Trả lời

Quay lại

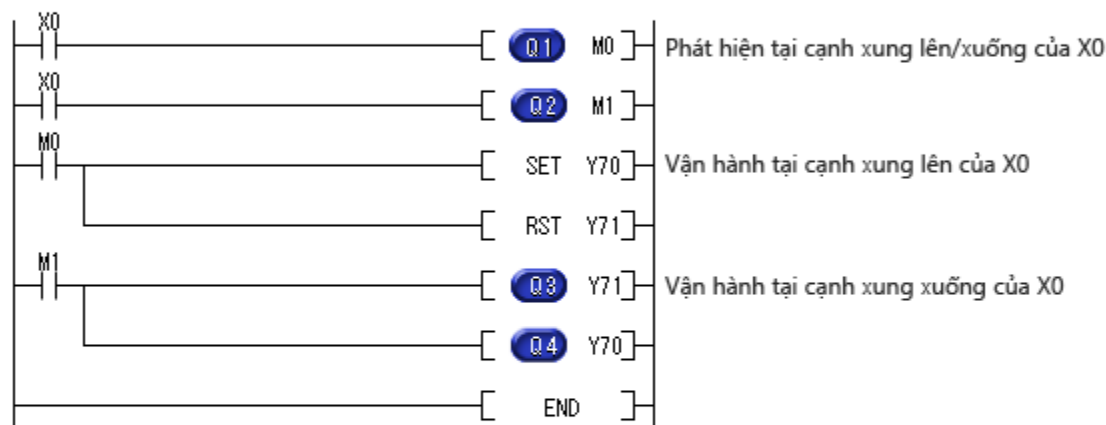
Kiểm tra Bài kiểm tra cuối khóa 5

Kéo và thả các lệnh phù hợp để hoàn thành chương trình thực hiện các vận hành sau:

Khi đang xử lý vật liệu, "tín hiệu đang xử lý" (X0) BẬT

Tại cạnh xung lên của "tín hiệu đang xử lý" (X0), đèn A BẬT (Y70 BẬT) và đèn B TẮT (Y71 TẮT).

Tại cạnh xung xuống của "tín hiệu đang xử lý" (X0), đèn B BẬT (Y70 BẬT) và đèn A TẮT (Y71 TẮT).



Q1 --Select-- ▾

Q2 --Select-- ▾

Q3 --Select-- ▾

Q4 --Select-- ▾

Trả lời

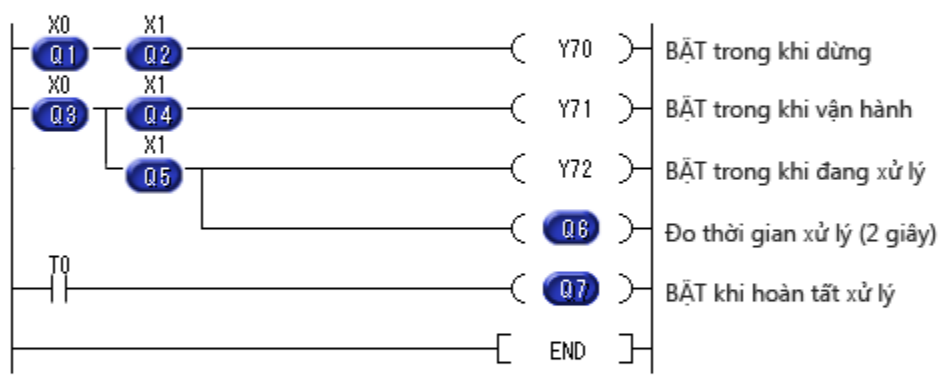
Quay lại

Kiểm tra Bài kiểm tra cuối khóa 6

Kéo và thả các lệnh phù hợp để hoàn thành chương trình thực hiện các vận hành sau:

BẬT đèn bằng cách BẬT/TẮT công tắc khởi động vận hành (X0) và công tắc khởi động xử lý (X1).
Đèn D sẽ BẬT sau khi BẬT cả hai công tắc được 2 giây.

[Khởi động vận hành (X0)]	[Công tắc khởi động xử lý (X1)]	[Đèn]
TẮT	TẮT	Đèn A (Y70 BẬT)
BẬT	TẮT	Đèn B (Y71 BẬT)
BẬT	BẬT	Đèn C (Y72 BẬT) và sau 2 giây, Đèn D (Y73 BẬT)



Q1 Q2 Q3 Q4
 Q5 Q6 Q7

Kiểm tra Bài kiểm tra cuối khóa 7

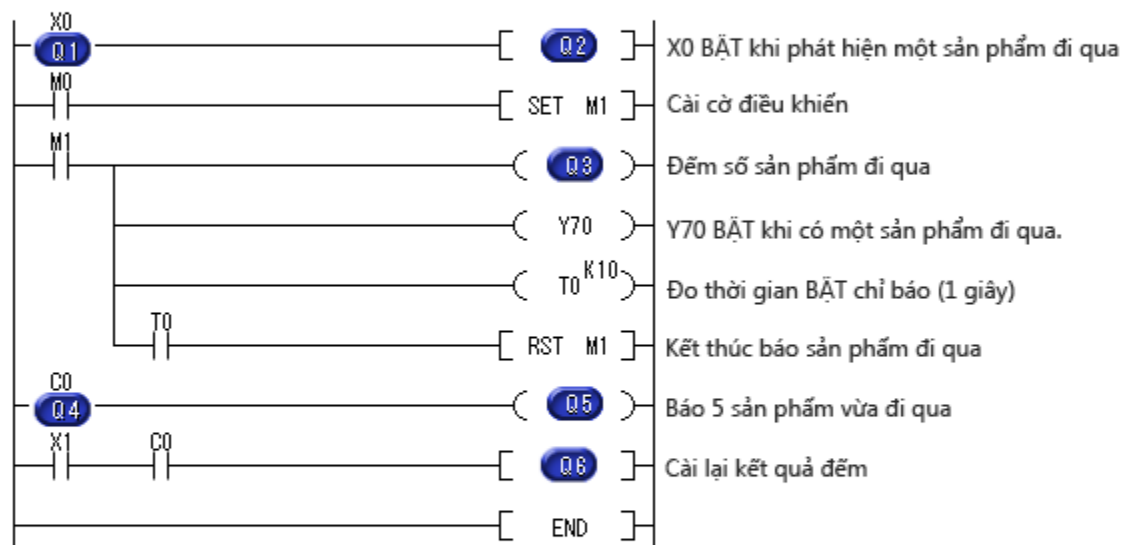
Kéo và thả các lệnh phù hợp để hoàn thành chương trình thực hiện các vận hành sau:

Khi một sản phẩm đi qua trên băng tải, tín hiệu X0 sẽ BẬT.

Khi một sản phẩm đi qua (sau 3 giây), đèn A sẽ BẬT. (Y70 sẽ BẬT trong 1 giây).

Sau khi 5 sản phẩm đi qua, đèn B sẽ BẬT. (Y71 BẬT)

Sau khi đèn B BẬT, công tắc xác nhận (X1) sẽ BẬT.



Q1 --Select-- ▼ Q2 --Select-- ▼ Q3 --Select-- ▼ Q4 --Select-- ▼

Q5 --Select-- ▼ Q6 --Select-- ▼

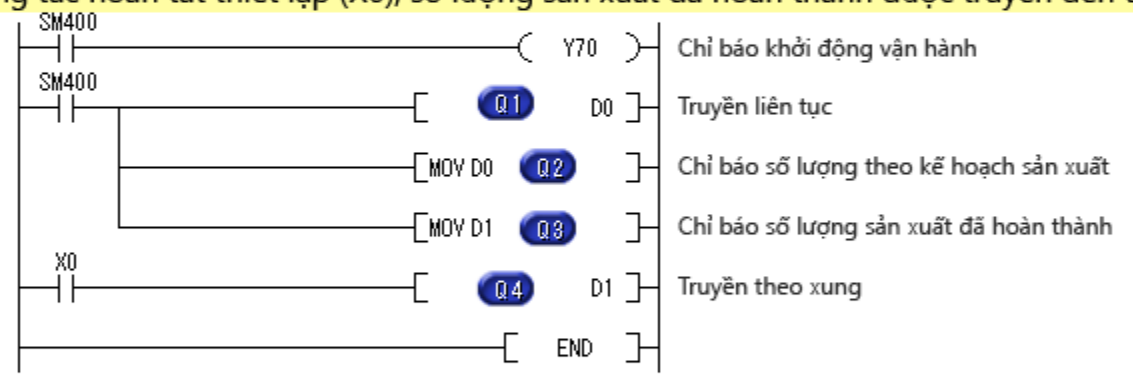
Trả lời

Quay lại

Kiểm tra Bài kiểm tra cuối khóa 8

Kéo và thả các lệnh phù hợp để hoàn thành chương trình thực hiện các vận hành sau:

- 1) Khi khởi động vận hành, đèn A sẽ BẬT. (Y70 BẬT)
- 2) Số lượng sản phẩm theo kế hoạch sản xuất được nhập bằng cách sử dụng các công tắc kỹ thuật số (X20-X2F). Số lượng được truyền tới thanh ghi dữ liệu D0 ngay khi được nhập.
- 3) Các giá trị được lưu trong thanh ghi dữ liệu (D0, D1) sẽ được truyền và cập nhật liên tục trên màn hình kỹ thuật số như dưới đây.
 - Y40-Y4F: Chỉ báo số lượng theo kế hoạch sản xuất (D0)
 - Y50-Y5F: Chỉ báo số lượng sản xuất đã hoàn thành (D1)
- 4) Các công tắc kỹ thuật số X30 đến X3F được sử dụng để nhập số lượng sản xuất đã hoàn thành. Khi BẬT công tắc hoàn tất thiết lập (X0), số lượng sản xuất đã hoàn thành được truyền đến thanh ghi dữ liệu D1.



* Trong chương trình này, lệnh MOV được sử dụng để truyền dữ liệu.
 * Để theo dõi D0 và D1, hãy sử dụng các giá trị thập lục phân.

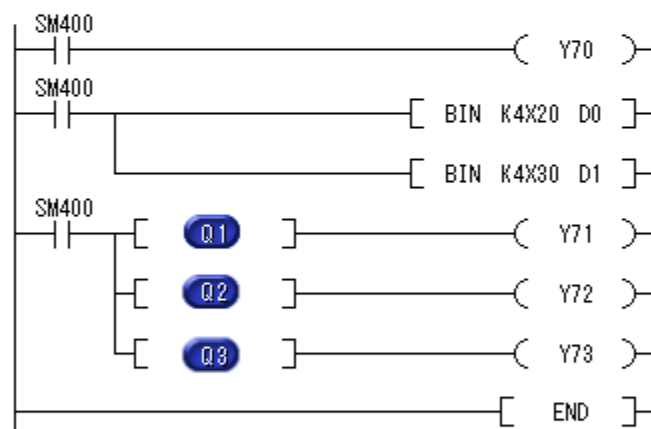
Q1 Q2 Q3 Q4

Kiểm tra Bài kiểm tra cuối khóa 9



Kéo và thả các lệnh phù hợp để hoàn thành chương trình thực hiện các vận hành sau:

- 1) Khi khởi động vận hành, đèn A sẽ BẬT. (Y70 BẬT)
- 2) Các vận hành sau đây sẽ được thực hiện.
 - Số lượng theo kế hoạch sản xuất A đã được nhập bằng cách sử dụng công tắc kỹ thuật số (X20-X2F), được định dạng và truyền tới thanh ghi dữ liệu D0.
 - Số lượng theo kế hoạch sản xuất B đã được nhập bằng cách sử dụng công tắc kỹ thuật số (X30-X3F), được định dạng và truyền tới thanh ghi dữ liệu D1.
 - Các thanh ghi dữ liệu D0 và D1 được so sánh với nhau và kết quả được chỉ báo bằng đèn.
 - D0>D1: Đèn B (Y71 BẬT/TẮT)
 - D0=D1: Đèn C (Y72 BẬT/TẮT)
 - D0<D1: Đèn D (Y73 BẬT/TẮT)



Chỉ báo khởi động vận hành

Nhận dữ liệu đầu vào

So sánh và chỉ báo kết quả

Kết thúc chương trình con

Q1 Q2 Q3

Trả lời

Quay lại

Kiểm tra Bài kiểm tra cuối khóa 10



Kéo và thả các lệnh phù hợp để hoàn thành chương trình thực hiện các vận hành sau:

- 1) Khi khởi động vận hành, đèn A sẽ BẬT. (Y70 BẬT)
- 2) Ban đầu, số lượng 100 theo kế hoạch sản xuất được lưu vào thanh ghi dữ liệu D0.
- 3) Mỗi lần có một sản phẩm được hoàn thành, thông tin sau sẽ được lưu vào các thanh ghi dữ liệu.

D1: Số lượng sản xuất đã hoàn thành (được đếm tại cạnh xung lên của X0)

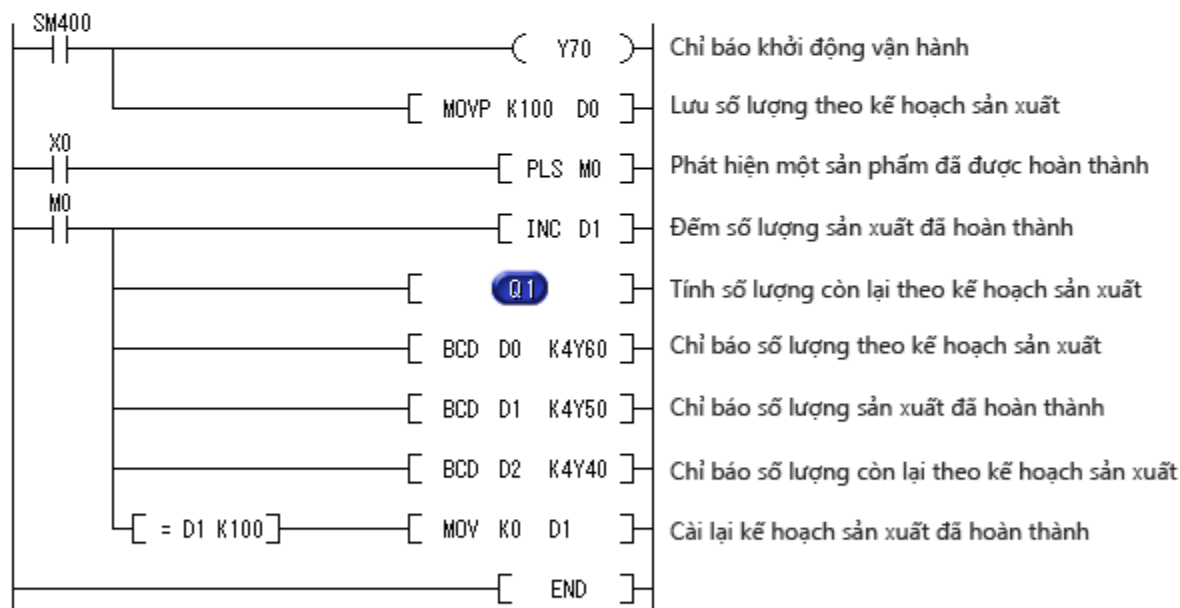
D2: Số lượng còn lại theo kế hoạch sản xuất ($D2 = D0 - D1$)

Màn hình kỹ thuật số hiển thị dữ liệu sau:

Y40-Y4F: Giá trị trong D2 (số lượng còn lại theo kế hoạch sản xuất (0 đến 100))

Y50-Y5F: Giá trị trong D1 (số lượng sản xuất đã hoàn thành (0 đến 100))

Y60-Y6F: Giá trị D0 (số lượng theo kế hoạch sản xuất (100))



Q1 --Select--

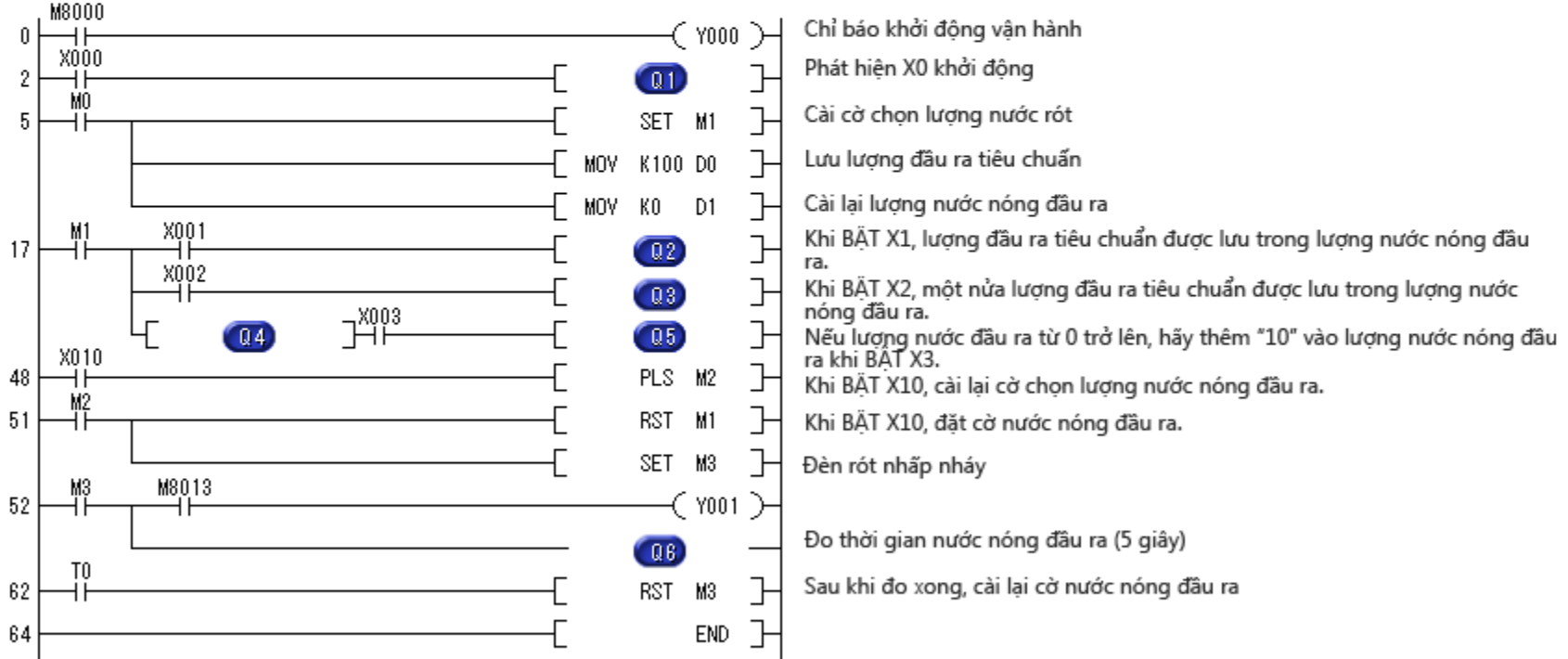
Trả lời

Quay lại

Kiểm tra Bài kiểm tra cuối khóa 11

Chương trình điều khiển được biểu thị dưới đây dành cho MELSEC-Sê-ri F và có các lệnh cùng rờ le đặc biệt. Kéo và thả các lệnh phù hợp để hoàn thành chương trình rớt nước nóng từ bình đun nước nóng:

- 1) Khi khởi động vận hành, đèn sẽ BẬT. (Y0 BẬT)
- 2) Tại cạnh xung lên khi khởi động vận hành bình đun nước nóng (X0 BẬT), "100" được lưu trong lượng nước tiêu chuẩn rớt ra D0 và "0" được lưu trong lượng nước nóng đầu ra D1. (Cài lại dữ liệu)
- 3) Chọn lượng nước nóng đầu ra
 Tại cạnh xung lên của X1, lượng đầu ra tiêu chuẩn D0 được lưu trong lượng nước nóng đầu ra D1.
 Tại cạnh xung lên của X2, một nửa lượng đầu ra tiêu chuẩn D0 được lưu trong lượng nước nóng đầu ra D1.
- 4) Nếu lượng nước nóng đầu ra D1 được chọn và có giá trị từ 0 trở lên, thì "+10" sẽ được cộng vào lượng nước nóng đầu ra D1 tại cạnh xung lên của X3, sau đó giá trị cộng thêm vào sẽ được lưu trong lượng nước nóng đầu ra.
- 5) Tại cạnh xung lên của đầu ra nước nóng (X10), đèn rớt nhấp nháy với khoảng thời gian 1 giây (Y1 lại BẬT/TẮT) và đo thời gian đầu ra nước nóng (T0) là 5 giây.
- 6) Đèn rớt nhấp nháy dừng lại sau khi đếm thời gian đầu ra nước nóng (T0).



Q1 Q2 Q3 Q4

Q5 Q6

Kiểm tra Điểm kiểm tra

Bạn đã hoàn thành Bài kiểm tra cuối khóa. Kết quả của bạn như sau.
Để kết thúc bài kiểm tra cuối khóa, hãy tiếp tục tới trang tiếp theo.

Câu trả lời đúng : **11**

Tổng số câu hỏi : **11**

Tỷ lệ phần trăm : **100%**

Tiếp tục

Xem lại

Chúc mừng bạn. Bạn đã đạt bài kiểm tra.

Bạn đã hoàn thành Khóa học **Cơ bản về lập trình PLC.**

Cảm ơn bạn đã tham gia khóa học này.

Chúng tôi hy vọng bạn thích các bài học và những thông tin bạn có được trong khóa học này sẽ hữu ích trong tương lai.

Bạn có thể xem lại khóa học này nhiều lần tùy ý.

Xem lại

Đóng