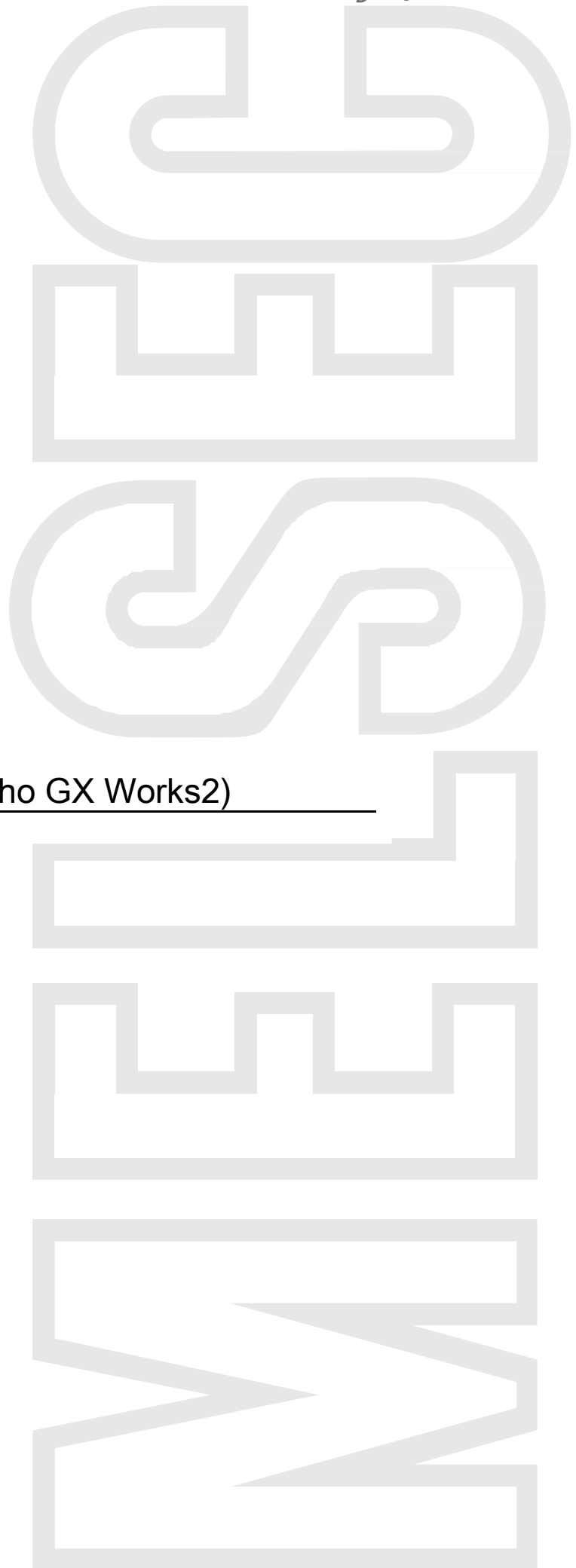


# mitsubishi

*Changes for the Better*

Bộ điều khiển khả trình  
Tài liệu hướng dẫn

Khóa học cơ bản về Q-series (cho GX Works2)





# • CẢN THẬN •

(Luôn đọc hướng dẫn trước khi sử dụng sản phẩm.)

Khi thiết kế hệ thống, luôn đọc tài liệu liên quan và cân nhắc cẩn thận đối với vấn đề an toàn. Trong phần này, cần chú ý đến những điểm sau và sử dụng sản phẩm thật cẩn thận

## [CHÚ Ý KHI THỰC HÀNH]

### CẢNH BÁO

- Không chạm vào tiếp điểm khi vẫn đang cắm điện để tránh bị điện giật
- Trước khi mở vỏ bọc an toàn, chắc chắn tắt nguồn hoặc đảm bảo an toàn
- Không chạm vào bộ phận di chuyển

### CẢN THẬN

- Làm theo chỉ dẫn của người hướng dẫn
- Không tháo rời khối module của máy hoặc thay đổi đấu nối điện mà không có sự đồng ý của hướng dẫn. Làm như vậy có thể dẫn đến hỏng, trục trặc, khiến bản thân bạn bị thương hoặc hỏa hoạn
- Tắt nguồn điện trước khi lắp vào hoặc tháo khối module ra. Không làm theo có thể dẫn đến sự cố hoặc điện giật
- Khi thiết bị Demo (ví dụ như bảng X/Y) phát ra những âm thanh/mùi bất thường, ấn vào “Power switch” hoặc “Emergency switch” để ngắt nguồn
- Khi có vấn đề xảy ra, thông báo cho người hướng dẫn sớm nhất có thể

KIỂM TRA

\*Mã sách được viết dưới đây bên trái của bìa sách.

Ngày in	Số sách	Kiểm tra
Tháng 10 2012	SH-081123ENG-A	Bản đầu tiên

Quyển sách này không có bản quyền công nghiệp hay bản quyền nào khác. Tập đoàn điện tử Mitsubishi không có trách nhiệm gì cho bất kể vấn đề gì liên quan đến bản quyền tài sản công nghiệp có thể xảy đến khi sử dụng nội dung trong quyển sách này.



## Mục lục

<b>CHƯƠNG 1</b>	<b>CƠ BẢN VỀ BỘ ĐIỀU KHIỂN KHẢ TRÌNH</b>	<b>1- 1 đến 1-14</b>
1.1	Chương trình.....	1- 1
1.2	Quy trình xử lí chương trình.....	1- 4
1.3	Cấu hình module MELSEC-QnUD.....	1- 5
1.4	Tín hiệu I/O ngoài và số I/O.....	1-11
1.5	Cấu hình hệ thống và số I/O của thiết bị DEMO.....	1-14
<b>CHƯƠNG 2</b>	<b>SỬ DỤNG GX Works2</b>	<b>2- 1 đến 2-64</b>
2.1	Tính năng trong GX Works2.....	2- 3
2.1.1	MELSOFT iQ Works .....	2- 7
2.2	Kiến thức căn bản để sử dụng GX Works2.....	2- 9
2.2.1	Cấu hình màn hình trong GX Works2 .....	2- 9
2.2.2	Chỉnh sửa ladder .....	2-11
2.2.3	Dự án.....	2-20
2.3	Thao tác trước khi tạo chương trình tuần tự ladder.....	2-22
2.3.1	Khởi động GX Works2.....	2-22
2.3.2	Tạo một dự án mới.....	2-23
2.4	Các bước chuẩn bị để khởi động CPU .....	2-25
2.5	Tạo một chương trình ladder .....	2-32
2.5.1	Tạo một chương trình ladder sử dụng phím chức năng.....	2-32
2.5.2	Tạo một chương trình sử dụng thanh công cụ.....	2-34
2.6	Chuyển đổi chương trình (chuyển đổi ladder) .....	2-36
2.7	Đọc/Viết dữ liệu đến/từ khối CPU điều khiển khả trình .....	2-37
2.8	Quan sát trạng thái chương trình ladder .....	2-40
2.9	Chuẩn đoán lỗi CPU điều khiển khả trình.....	2-43
2.10	Chỉnh sửa chương trình ladder .....	2-45
2.10.1	Điều chỉnh một phần của chương trình ladder.....	2-45
2.10.2	Vẽ/xóa đường thẳng .....	2-47
2.10.3	Chèn/xóa một hàng.....	2-50
2.10.4	Cắt/sao chép chương trình ladder .....	2-55
2.11	Kiểm tra dữ liệu .....	2-58
2.12	Lưu chương trình ladder.....	2-59
2.12.1	Lưu dự án mới tạo hoặc ghi đè lên dự án.....	2-59
2.12.2	Lưu một dự án với một tên khác.....	2-60
2.13	Đọc một dự án đã lưu.....	2-61
2.14	Mở dự án theo định dạng khác .....	2-62
2.15	Lưu dự án với định dạng khác .....	2-63
<b>CHƯƠNG 3</b>	<b>BIẾN VÀ THAM SỐ BỘ ĐIỀU KHIỂN KHẢ TRÌNH</b>	<b>3- 1 đến 3-6</b>
3.1	Biến.....	3- 1
3.2	Tham số.....	3- 3

4.1	Danh sách các lệnh được giải thích trong chương trình này .....	4- 1	
4.2	Sự khác nhau giữa <input type="checkbox"/> OUT <input type="checkbox"/> và <input type="checkbox"/> SET <input type="checkbox"/> / <input type="checkbox"/> RST .....	4- 4	
4.3	Đo bởi bộ định thời .....	4- 5	
4.4	Đếm bởi bộ đếm.....	4- 6	
4.5	<input type="checkbox"/> PLS <input type="checkbox"/> / <input type="checkbox"/> PLF .....	4-14	
4.6	<input type="checkbox"/> MC <input type="checkbox"/> / <input type="checkbox"/> MCR .....	4-20	
4.7	<input type="checkbox"/> FEND <input type="checkbox"/> / <input type="checkbox"/> CJ <input type="checkbox"/> / <input type="checkbox"/> SCJ <input type="checkbox"/> / <input type="checkbox"/> CALL <input type="checkbox"/> / <input type="checkbox"/> RET .....	4-24	
4.7.1	<input type="checkbox"/> FEND .....	4-24	
4.7.2	<input type="checkbox"/> CJ <input type="checkbox"/> / <input type="checkbox"/> SCJ .....	4-27	
4.7.3	<input type="checkbox"/> CALL(P) <input type="checkbox"/> / <input type="checkbox"/> RET .....	4-31	
4.8	Bài tập .....	4-35	
4.8.1	Bài tập 1 LD to NOP.....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST 1 .....	4-35
4.8.2	Bài tập 2 SET, RST .....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST2 .....	4-36
4.8.3	Bài tập 3 PLS, PLF.....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST3 .....	4-38
4.8.4	Bài tập 4 CJ, CALL, RET, FEND.....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST4 .....	4-39

5.1	Kí hiệu của giá trị (Dữ liệu).....	5- 1	
5.2	Lệnh truyền.....	5- 9	
5.2.1	<input type="checkbox"/> MOV (P) .....	5- 9	
5.2.2	<input type="checkbox"/> BIN (P) <input type="checkbox"/> .....	5-16	
5.2.3	<input type="checkbox"/> BCD (P) .....	5-18	
5.2.4	Ví dụ xác định con số cho thiết bị bit và truyền dữ liệu .....	5-21	
5.2.5	<input type="checkbox"/> FMOV (P) <input type="checkbox"/> / <input type="checkbox"/> BMOV (P) .....	5-22	
5.3	So sánh lệnh hoạt động .....	5-27	
5.4	Lệnh hoạt động số học.....	5-32	
5.4.1	<input type="checkbox"/> +(P) <input type="checkbox"/> / <input type="checkbox"/> -(P) .....	5-32	
5.4.2	<input type="checkbox"/> *(P) <input type="checkbox"/> / <input type="checkbox"/> /(P) .....	5-36	
5.4.3	Lệnh dữ liệu 32 bit và sự cần thiết của nó .....	5-41	
5.4.4	Ví dụ tính toán cho phép nhân và chia bao gồm số sau dấu phẩy.....	5-43	
5.5	Thanh ghi chỉ số và thanh ghi tệp.....	5-44	
5.5.1	Làm thế nào để sử dụng thanh ghi chỉ số Z .....	5-44	
5.5.2	Làm thế nào để sử dụng thanh ghi tệp R.....	5-46	
5.6	Cài đặt mở rộng của giá trị đặt cho bộ định thời/bộ đếm và Hiển thị mở rộng giá trị hiện tại .....	5-49	
5.7	Bài tập .....	5-51	
5.7.1	Bài tập 1 MOV .....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST5 .....	5-51
5.7.2	Bài tập 2 Chuyển đổi BIN và BCD .....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST6 .....	5-52
5.7.3	Bài tập 3 FMOV .....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST7 .....	5-53
5.7.4	Bài tập 4 Lệnh so sánh .....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST8 .....	5-54
5.7.5	Bài tập 5 Lệnh cộng và trừ.....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST9 .....	5-55
5.7.6	Bài tập 6 Lệnh nhân và chia.....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST10 .....	5-56
5.7.7	Bài tập 7 D-nhân và D-chia.....	<input type="checkbox"/> .....Tên dự án <input type="checkbox"/> QTEST11 .....	5-57

6.1	Kiểm tra chức năng trực tuyến .....	6- 1
6.1.1	Bật và tắt biến "Y" bằng cưỡng bức .....	6- 2
6.1.2	Đặt và khởi động lại biến "M" .....	6- 4
6.1.3	Đổi giá trị hiện tại của biến "T" .....	6- 5
6.1.4	Đọc các bước lỗi.....	6- 6
6.1.5	STOP và RUN từ xa .....	6- 7
6.2	Cưỡng bức chỉ định I/O bằng cài đặt Parameter .....	6- 8
6.3	Làm thế nào để sử dụng bộ định thời có nhớ.....	6-10
6.4	Thay thế loạt biến .....	6-12
6.4.1	Thay thế loạt biến .....	6- 12
6.4.2	Thay đổi loạt biến được xác định giữa tiếp điểm thường mở và tiếp điểm thường đóng.....	6- 13
6.5	Thay đổi chương trình trực tuyến .....	6-14
6.6	Đăng kí biến .....	6-15
6.7	Làm thế nào để tạo chú thích.....	6-16
6.8	Cài đặt bảo mật cho dự án.....	6-23
6.8.1	Cài đặt và đặt lại bảo mật cho dự án .....	6-24
6.8.2	Quản lí (thêm, xóa và thay đổi) người dùng .....	6-25
6.8.3	Đăng nhập vào dự án .....	6-29
6.8.4	Thay đổi xác thực truy nhập cho mỗi cấp truy nhập .....	6-30
6.9	Chức năng lấy mẫu truy dấu .....	6-31

7.1	Module chức năng thông minh.....	7- 1
7.2	Giao tiếp dữ liệu giữa module chức năng thông minh và CPU .....	7- 2
7.2.1	Các tín hiệu I/O đến CPU.....	7- 3
7.2.2	Giao tiếp dữ liệu trong module chức năng thông minh.....	7- 4
7.3	Giao tiếp trong module chức năng thông minh.....	7- 5
7.3.1	Phương thức giao tiếp trong module chức năng thông minh.....	7- 5
7.4	Hệ thống module chức năng thông minh trong thiết bị DEMO .....	7- 6
7.5	Module chuyển đổi Q64AD Tương tự/Số.....	7- 7
7.5.1	Tên các bộ phận .....	7- 7
7.5.2	Đặc điểm của chuyển đổi A/D.....	7- 8
7.5.3	Danh sách tín hiệu I/O và phân vùng bộ nhớ đệm.....	7- 9
7.5.4	Bổ sung hoặc cài đặt dữ liệu cho module chức năng thông minh .....	7-12
7.5.5	Bài tập với thiết bị DEMO.....	7-16
7.6	Module chuyển đổi Q62DAN Số/Tương tự .....	7-17
7.6.1	Tên các bộ phận.....	7-17
7.6.2	Đặc điểm của chuyển đổi D/A .....	7-18
7.6.3	Danh sách các tín hiệu I/O và phân vùng bộ nhớ đệm .....	7-19
7.6.4	Bổ sung hoặc cài đặt dữ liệu cho module chức năng thông minh .....	7-21
7.6.5	Bài tập với thiết bị DEMO.....	7-25

<b>CHƯƠNG 8    CHỨC NĂNG MÔ PHÒNG</b>	<b>8- 1 đến 8- 4</b>
---------------------------------------	----------------------

8.1 Chức năng mô phỏng .....	8- 1
8.2 Chức năng chạy /dừng .....	8- 1
8.3 Gỡ lỗi với chương trình ví dụ .....	8- 2
8.3.1 Giám sát và kiểm tra trạng thái thiết bị .....	8- 3

<b>CHƯƠNG 9    BẢO DƯỠNG</b>	<b>9- 1 đến 9-8</b>
------------------------------	---------------------

9.1 Lỗi phổ biến .....	9- 1
9.2 Bảo dưỡng.....	9- 2
9.3 Sản phẩm tiêu hao.....	9- 3
9.4 Vòng đời của role đầu ra.....	9- 4
9.5 Sản phẩm dự phòng.....	9- 5
9.6 Sử dụng sản phẩm hỗ trợ .....	9- 7

<b>PHỤ LỤC</b>	<b>App.- 1 đến App.- 80</b>
----------------	-----------------------------

Phụ lục 1 Chế độ điều khiển I/O .....	App.- 1
1.1 Chế độ điều khiển trực tiếp.....	App.- 1
1.2 Chế độ làm tự động làm mới.....	App.- 2
1.3 So sánh giữa chế độ trực tiếp và chế độ làm mới .....	App.- 3
Phụ lục 2 Rơ le đặc biệt .....	App.- 4
Phụ lục 3 Thanh ghi đặc biệt .....	App.- 5
Phụ lục 4 Ví dụ trình ứng dụng.....	App.- 6
4.1 Flip-flop ladder .....	App.- 6
4.2 Bất thời gian ngắn (one shot).....	App.- 8
4.3 Bộ định thời gian dài.....	App.- 9
4.4 Bộ định thời tắt trễ ( off-delay) .....	App.-10
4.5 Bộ định thời bật trễ (On-delay ) .....	App.-11
4.6 Ladder lặp ON-OFF.....	App.-12
4.7 Chống lặp đầu vào.....	App.-12
4.8 Ladder với tuyến chung.....	App.-13
4.9 Chương trình điều khiển thời gian.....	App.-14
4.10 Đồng hồ ladder.....	App.-15
4.10.1 Chức năng clock (Bổ sung).....	App.-16
4.11 Khởi động sao /Tam giác.....	App.-18
4.12 Hiện thị khoảng thời gian đã qua và đầu ra trước thời gian giới hạn.....	App.-19
4.13 Bộ định thời có nhớ.....	App.-20
4.14 Bộ định thời chuyển đổi thiết lập thời gian bên ngoài.....	App.-21
4.15 Thiết lập bộ đếm ngoài.....	App.-22
4.16 Đo thời gian hoạt động.....	App.-24
4.17 Đo chu trình.....	App.-24
4.18 Ví dụ ứng dụng của (D) C M L (P).....	App.-25
4.19 Chương trình hiện thị chia giá trị của 4 chữ số BIN đến 4 hàng chữ số thập phân .....	App.-26
4.20 Kiểm soát điều khiển tuyến.....	App.-29
4.21 Nén chương trình tuần tự sử dụng bộ đếm vòng.....	App.-31
4.22 Ví dụ ứng dụng điều khiển vị trí.....	App.-35
4.23 Ví dụ ứng dụng sử dụng chỉ số Z.....	App.-36
4.24 Ví dụ ứng dụng lệnh FIFO.....	App.-38

4.25	Ví dụ ứng dụng về bù dữ liệu.....	App.-41
4.26	Ví dụ về chương trình hoạt động tính căn bậc 2 của số liệu.....	App.-44
4.27	Ví dụ về chương trình hoạt động tính toán năng lượng của dữ liệu.....	App.-45
4.28	Chương trình sử dụng chuyển đổi kỹ thuật để nhập dữ liệu.....	App.-46
4.29	Hiển thị số lỗi và số lỗi sử dụng chương trình phát hiện lỗi.....	App.-47
Phụ lục 5	Bộ nhớ và tập tin được xử lý bởi CPU module.....	App.-51
Phụ lục 6	So sánh với GX Developer (thay đổi).....	App.-53
Phụ lục 7	Tùy chỉnh phím tắt.....	App.-62
Phụ lục 8	Chỉ mục.....	App.-64
Phụ lục 9	FB.....	App.-68
9.1	FB.....	App.-68
9.1.1	Chuyển đổi khối chức năng.....	App.-68
9.1.2	Ưu điểm của việc sử dụng FBS.....	App.-69
9.1.3	Thư viện FB.....	App.-71
9.1.4	Công cụ phát triển.....	App.-73
9.1.5	FB quy cách và biện pháp phòng ngừa.....	App.-73
9.2	Tạo chương trình bằng cách sử dụng thư viện FB.....	App.-74
9.2.1	Các chương trình được tạo ra.....	App.-74
9.2.2	Chuẩn bị trước khi dùng thư viện FB.....	App.-75
9.2.3	Nhập vào thư viện FB dự án.....	App.-76
9.2.4	Dán FBS.....	App.-77
9.2.5	Cài đặt tên của dán FBs.....	App.-78
9.2.6	Tạo ra đầu vào và đầu ra ladder.....	App.-79
9.2.7	Thực hiện sự chuyển đổi / biên dịch.....	App.-79
9.2.8	Ghi chương trình tuần tự.....	App.-80
9.2.9	Thao tác kiểm tra.....	App.-80

## GIỚI THIỆU

Quyển sách này nói về bộ điều khiển khả trình, phương pháp chỉnh sửa chương trình với GX Works2, hướng dẫn bên dưới và hướng dẫn ứng dụng để hiểu lập trình MELSEC-Q series. Hệ thống nhiều CPU khả dụng cho series MELSEC-Q với nhiều khối module CPU, nhưng quyển sách này chỉ nói về trường hợp một module CPU được sử dụng.

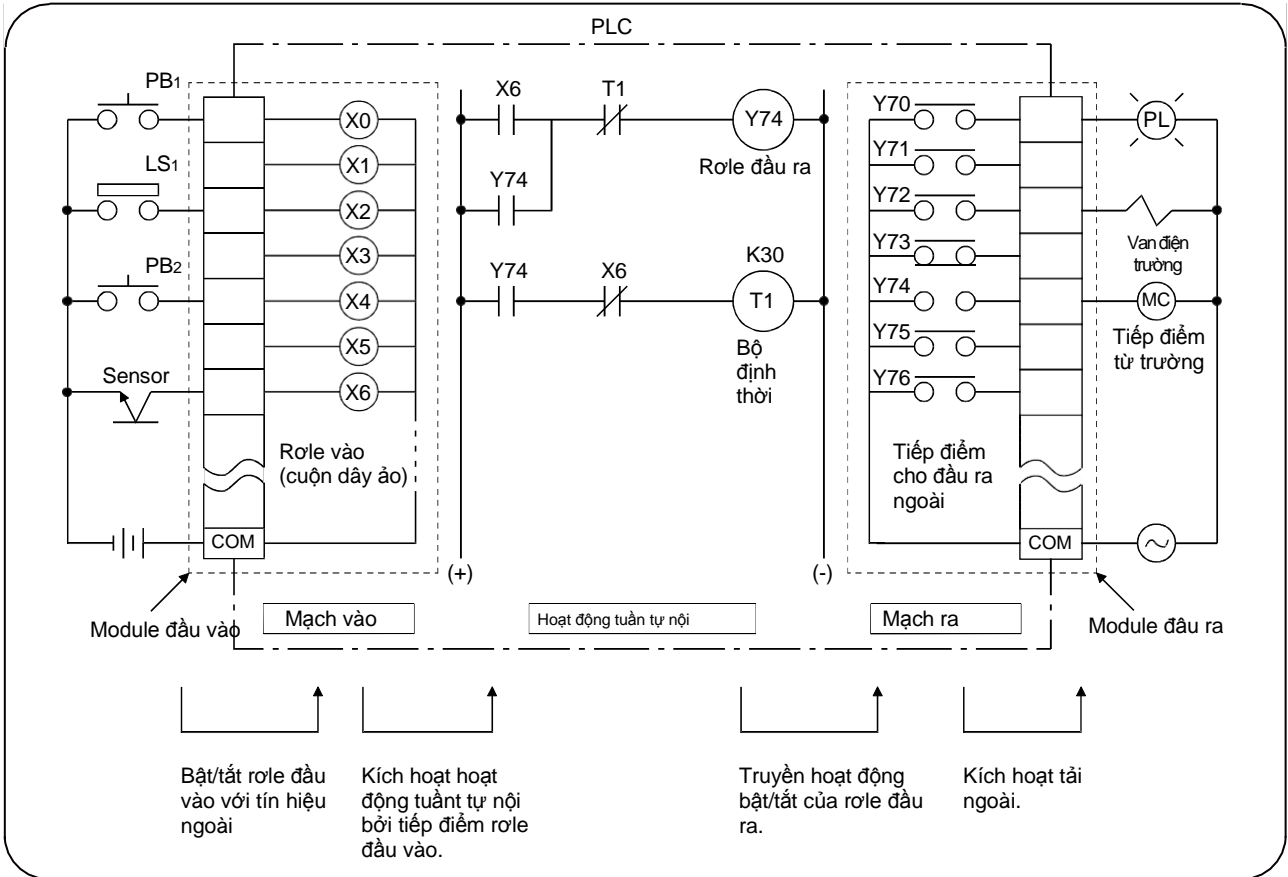
Những tài liệu liên quan có thể xem dưới đây:

- (1) QCPU User's Manual (Hardware Design, Maintenance and Inspection)  
..... SH-(NA)080483ENG  
Explains the hardware.
- (2) QnUCPU User's Manual (Function Explanation, Program Fundamentals)  
..... SH(NA)-080807ENG  
Explains the functions and programming method.
- (3) MELSEC-Q/L Programming Manual (Common Instruction)  
..... SH(NA)-080809ENG  
Explains details of each instruction.
- (4) GX Works2 Beginner's Manual (Simple Project)  
..... SH(NA)-080787ENG
- (5) GX Works2 Version 1 Operating Manual (Common)  
..... SH(NA)-080779ENG
- (6) GX Works2 Version 1 Operating Manual (Simple Project)  
..... SH(NA)-080780ENG
- (7) Before Using the Product  
..... BCN-P5782
- (8) Analog-Digital Converter Module User's Manual  
..... SH(NA)-080055
- (9) Digital-Analog Converter Module User's Manual  
..... SH(NA)-080054
- (10) I/O Module Type Building Block User's Manual  
..... SH(NA)-080042
- (11) MELSOFT GX Works2 FB Quick Start Guide  
..... L-08182ENG

# CHƯƠNG 1 CƠ BẢN VỀ BỘ ĐIỀU KHIỂN KHẢ TRÌNH

## 1.1 Chương trình

Nếu 1 bộ điều khiển khả trình là một mạch điều khiển hình thang. Nó được miêu tả bởi đầu vào ladder, đầu ra ladder, và các hoạt động tuần tự bên trong



Hình 1.1 Cấu hình bộ điều khiển khả trình

Một bộ điều khiển khả trình là thiết bị điện trung tâm giống như máy tính siêu nhỏ. Thực tế, một bộ điều khiển khả trình là tổ hợp của role, bộ định thời, và bộ đếm. Như hình 1.1, hoạt động tuần tự bên trong được thực thi bởi việc bật hoặc tắt cuộn dây. Bật/tắt của cuộn dây phụ thuộc vào điều kiện kết nối (theo nối tiếp hay song song) và dẫn đến công tắc mở hay công tắc đóng.

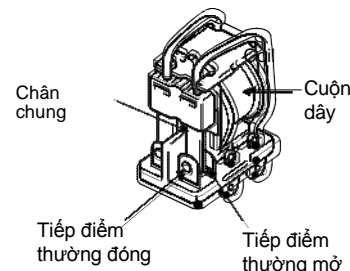
“Role”, còn được gọi là role điện từ, là một bộ chuyển để chuyển sang tín hiệu role. Role là thành phần quyết định để tạo nên một thang logic.

1) Cấp điện cuộn dây → Tạo từ trường

- Tiếp điểm thường mở đóng (dẫn điện)
- Tiếp điểm thường đóng mở

2) Ngắt điện cuộn dây → Giải từ trường

- Tiếp điểm thường mở mở (không dẫn điện)
- Tiếp điểm thường đóng đóng lại. (dẫn điện)



	Cuộn dây bật	Cuộn dây tắt
Tiếp điểm thường mở	Không dẫn	Conducted
Tiếp điểm thường đóng	Conducted	Không dẫn

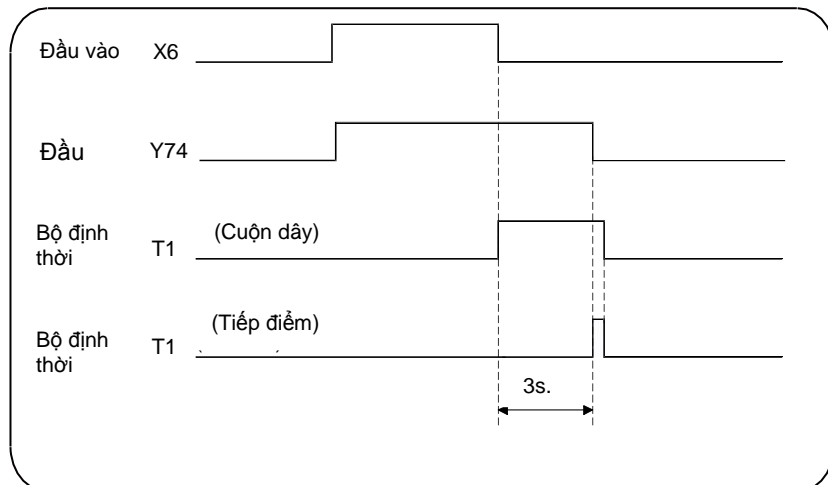
### Hoạt động tuần tự

Dưới đây là chu trình tín hiệu của hoạt động tuần tự bên trong của hình 1.1

- 1) Khi cảm biến bật lên, cuộn dây của rơle đầu vào X6 bị nhiễm từ
- 2) Cuộn dây của rơle đầu vào X6 nhiễm từ dẫn điện cho tiếp điểm thường mở X6 và cuộn dây của rơle Y74 nhiễm từ. (bộ định thời không bị nhiễm từ ở thời điểm này, tiếp điểm thường đóng vẫn dẫn điện).
- 3) Vì cuộn dây của rơle Y74 bị nhiễm từ, tiếp điểm lới ra ngoài Y74 dẫn điện và tiếp điểm từ (MC) được bật lên
- 4) Tắt cảm biến đi giải từ hóa cho cuộn dây của rơle đầu vào X6 và tiếp điểm thường mở X6 trở nên không dẫn điện.  
Vì tiếp điểm thường mở Y74 vẫn duy trì dẫn điện, cuộn dây vẫn bị nhiễm từ (hoạt động tự duy trì)
- 5) Khi cuộn dây của rơle đầu ra Y74 bị từ hóa (với tiếp điểm thường mở Y74 dẫn điện), tắt cảm biến đi (với tiếp điểm thường đóng X6 dẫn điện) từ hóa cuộn dây ở thời điểm T1 và bộ định thời bắt đầu đo thời gian  
Sau 3s (K30 nghĩa là 3.0s), tiếp điểm thường mở của bộ định thời dẫn điện và tiếp điểm thường đóng không dẫn điện.
- 6) Cuối cùng, cuộn dây của rơle đầu ra Y74 giải từ hóa và tải từ của tiếp điểm giảm xuống. Do đó, quá trình tự hoạt động của rơle đầu ra được giải phóng

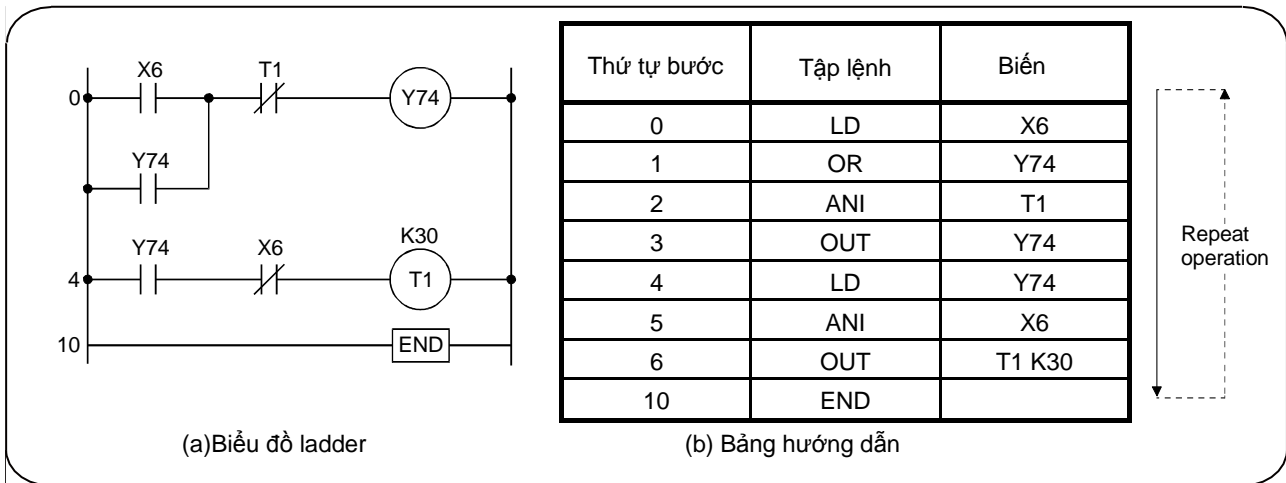
### Sơ đồ hoạt động

Biểu đồ thời gian dưới đây mô tả hoạt động của rơle đầu vào/ra theo thời gian.





Hoạt động tuần tự bên trong có thể coi như một chương trình của bộ điều khiển khả trình. Chương trình được lưu trong bộ nhớ chương trình tương tự như hướng dẫn dưới đây.

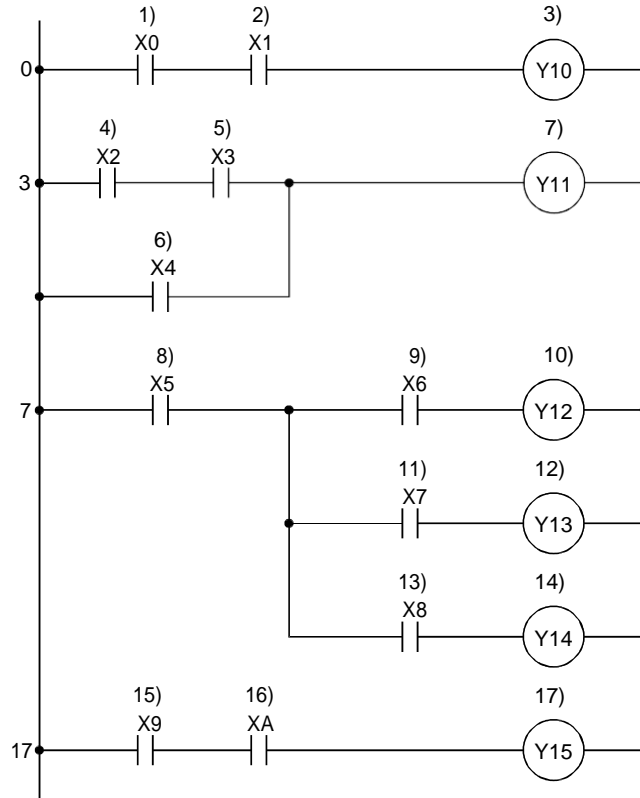


Hình 1.2 Chương trình

- Một chương trình là một tập hợp của rất nhiều tập lệnh và biến.
- Một lệnh gồm tập lệnh và biến. Thêm vào đó, lệnh thể hiện trình tự hoạt động. Thứ tự các lệnh là thứ tự hoạt động (tập lệnh cũng được gọi là lệnh)
- Các bước thực hiện thay đổi dựa trên loại lệnh hoặc phương pháp cài đặt để tạo giá trị sử dụng cho I/O và hoạt động (càng nhiều bước thực hiện thì hoạt động càng phức tạp)
- Lệnh lặp từ bước 0 đến từ lệnh END (nó còn được gọi là 'hoạt động lặp lại, "chu trình hoạt động" hoặc "quét"). Thời gian cần thiết cho một chu trình từ đầu đến cuối được gọi là chu trình hoạt động (chu kì quét)
- Số bước thực hiện từ bước 0 đến lệnh END là độ dài hay độ lớn của toàn bộ chương trình
- Chương trình được lưu vào bộ nhớ chương trình bên trong CPU. Hoạt động được thực thi theo khối hình thang Một khối hình thang thay đổi từ lúc bắt đầu thực thi lệnh (LD,LDI) đến lệnh OUT (bao gồm cả lệnh dữ liệu)

## 1.2 Quy trình xử lí chương trình

Chương trình hoạt động được thực thi tuần tự từ bước bắt đầu của chương trình theo thứ tự từ trái qua phải và từ trên xuống dưới (theo thứ tự 1,2.... 17) theo khối hình thang được chỉ ra như hình dưới.



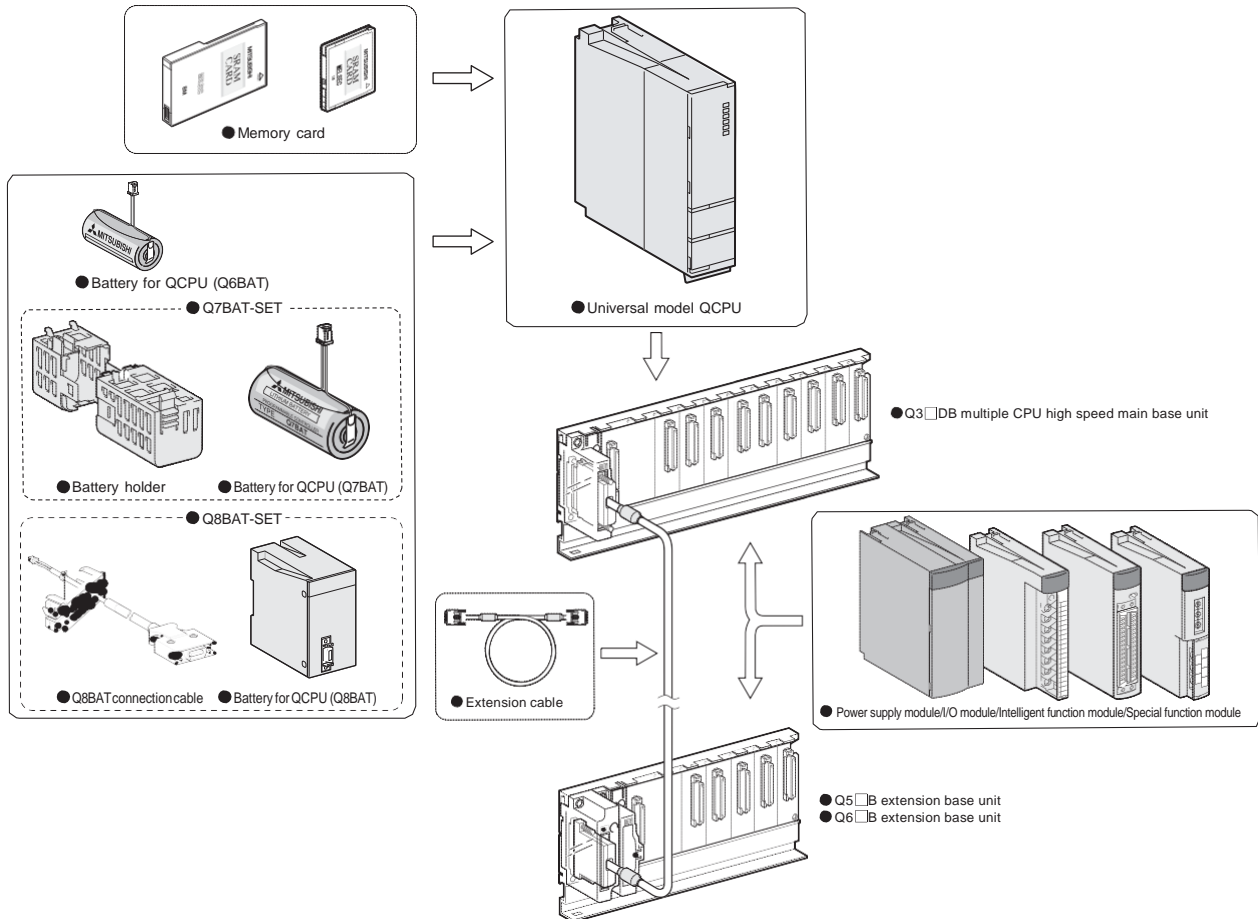
### 1.3 Cấu hình module MELSEC-QnUD

(1) Universal PLC

Universal PLC - QCPU được sử dụng cho việc đào tạo trong tài liệu này, vì thế, “QCPU” dùng để chỉ “ Universal model QCPU” trừ khi có chú ý khác.

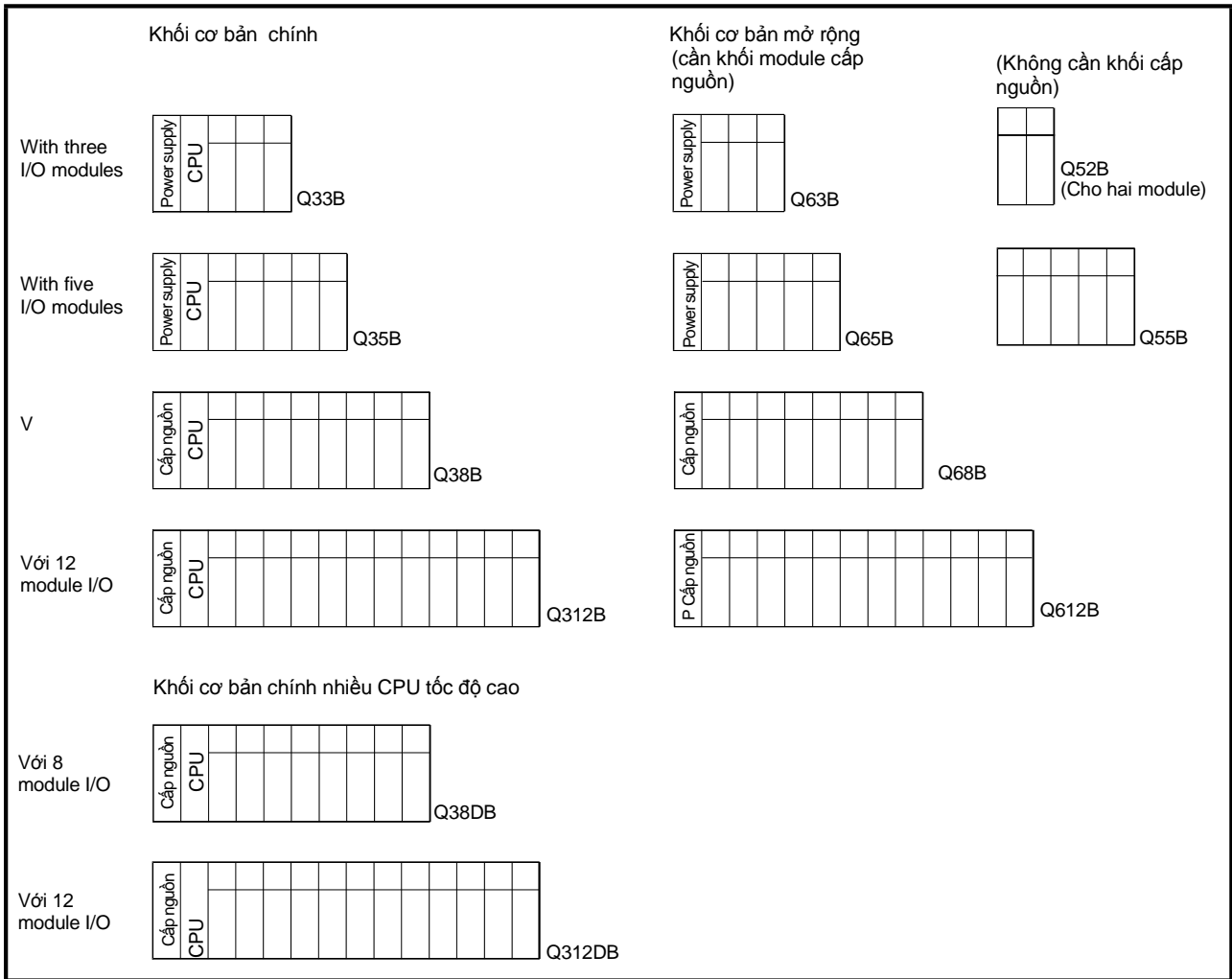
(2) Cấu hình cơ bản của một hệ thống điều khiển khả trình

Hình dưới là cấu hình của một bộ điều khiển khả trình thật sự



Hình 1.3 Cấu hình module MELSEC-QnUD (khi Q3DB được sử dụng)

Khối cơ bản



- Chức năng chính của khối cơ bản là: module cấp nguồn, CPU module, và module I/O, cung cấp nguồn 5VDC từ module cấp nguồn đến module CPU và module I/O, và truyền tín hiệu điều khiển đến mỗi module.

Module cấp nguồn

Tên module	Đầu vào	Đầu ra
Q61P	100V to 240VAC	5VDC 6A
Q62P	100V to 240VAC	5VDC 3A, 24VDC 0.6A
Q63P	24VDC	5VDC 6A
Q64P(N)	100V to 120V/AC200 to 240VAC	5VDC 8.5A
Q61P-D	100V to 240VAC	5VDC 6A

Module CPU

Loại CPU	Dung lượng (tối đa)	Tốc độ xử lý lệnh cơ bản	Số điểm I/O tối đa cho một khối điều khiển khả lập trình
Q00UJCPU	10K step	120ns	256 điểm
Q00UCPU	10K step	80ns	1024 điểm
Q01UCPU	15K step	60ns	1024 điểm
Q02UCPU	20K step	40ns	2048 điểm
Q03UD(E)CPU	30K step	20ns	4096 điểm
Q04UD(E)HCPU	40K step	9.5ns	
Q06UD(E)HCPU	60K step		
Q10UD(E)HCPU	100K step		
Q13UD(E)HCPU	130K step		
Q20UD(E)HCPU	200K step		
Q26UD(E)HCPU	260K step		
Q50UDEHCPU	500K step		
Q100UDEHCPU	1000K step		

Module I/O

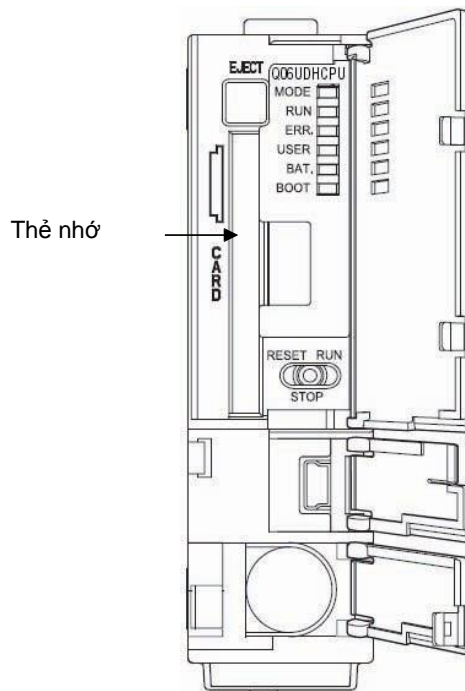
Định dạng		I/O điểm			
		8 điểm	16 điểm	32 điểm	64 điểm
Module đầu vào	120VAC	–	☒	–	–
	240VAC	☒	–	–	–
	24VDC (điểm chung dương)	–	☒	☒	☒
	24VDC (đầu vào tốc độ cao)	☒	–	–	–
	24VDC (tiếp tiếp điểm chung)	–	☒	☒	–
	5/12VDC	–	☒	☒	☒
Module đầu ra	Công tắc đầu ra	–	☒	–	–
	Công tắc đầu ra độc lập	☒	–	–	–
	Đầu ra Triac	–	☒	–	–
	Đầu ra transistor (sink)	☒	☒	☒	☒
	Đầu ra transistor (source)	–	☒	☒	–
I/O kết hợp		☒	–	☒	–

### Thẻ nhớ

Một QCPU được trang bị một bộ nhớ tích hợp sẵn như bình thường để lưu trữ các tham số và chương trình, do đó, chương trình có thể chạy mà không cần một thẻ nhớ

Thẻ nhớ cần trong những trường hợp trong bảng bên dưới.

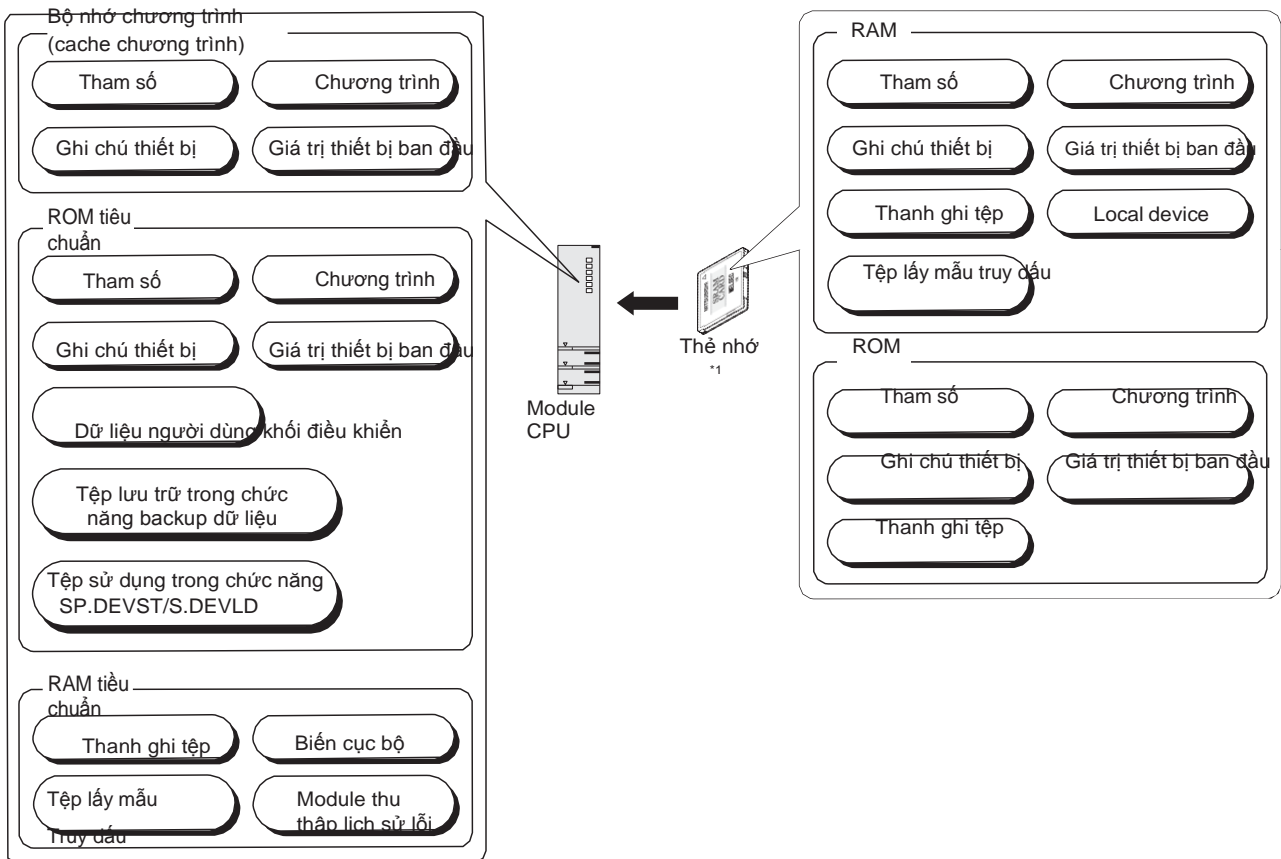
Loại	Mô tả
Card SRam	Dữ liệu có thể viết hoặc thay đổi bên trong bộ nhớ <ví dụ cách sử dụng> <ul style="list-style-type: none"> <li>• Để boot</li> <li>• Để lưu trữ dữ liệu mẫu truy dấu</li> <li>• Để lưu trữ dữ liệu theo dấu SFC</li> <li>• Để lưu trữ dữ liệu lịch sử lỗi</li> </ul>
Card Flash	Nội dung của bộ nhớ chương trình hoặc tệp nào đó có thể được ghi ở một thời điểm nhất định Dữ liệu mới được viết vào thay thế tất cả dữ liệu gốc. Dữ liệu có thể đọc bằng lệnh READ của chương trình <ví dụ cách sử dụng> <ul style="list-style-type: none"> <li>• Để boot</li> <li>• Khi thay đổi dữ liệu là không cần thiết</li> </ul>
Card ATA	Dữ liệu có thể được viết hoặc thay đổi trong chương trình Bộ điều khiển khả lập trình dữ liệu người dùng của một thẻ ATA có thể truy nhập bởi tệp lệnh access (như lệnh FWRITE) trong dãy chương trình qua định dạng CSV hoặc dạng nhị phân. <ví dụ cách sử dụng> <ul style="list-style-type: none"> <li>• Để boot</li> <li>• Cho bộ điều khiển khả trình dữ liệu người dùng (dữ liệu thông thường)</li> </ul>



- Phải có thẻ nhớ khi dung lượng dữ liệu vượt quá dung lượng bộ nhớ tích hợp, RAM tiêu chuẩn và ROM tiêu chuẩn
- Chọn card bộ nhớ theo độ lớn của chương trình hoặc theo loại dữ liệu được lưu
- Lắp pin dự phòng trước khi sử dụng card RAM loại SRAM. Dữ liệu card SRAM không thể lưu dự phòng trừ khi pin được lắp vào
- Format card bộ nhớ trước khi sử dụng nó
- Dữ liệu có thể viết vào card flash 100.000 lần, và với card ATA, dữ liệu có thể viết vào 1.000.000 lần.

< tham khảo: cấu hình bộ nhớ hệ thống của Universal Q-CPU >

Bộ nhớ của Universal Q-CPU bao gồm những khối sau.



\*1: Thẻ nhớ không thể sử dụng cho Q00UJCPU, Q00UCPU, Q01UCPU.

\*2: Q00UJCPU không có RAM tiêu chuẩn.

- Bộ nhớ chương trình:** Bộ nhớ lưu chương trình và tham số cho một module CPU hoạt động. Một chương trình hoạt động được thực thi bằng việc đưa chương trình được lưu trong bộ nhớ chương trình đến bộ nhớ cache chương trình.
- Bộ nhớ cache:** Một bộ nhớ cho chương trình hoạt động. Một chương trình hoạt động được thực thi bằng việc đưa chương trình được lưu trong bộ nhớ chương trình đến bộ nhớ cache chương trình.
- RAM tiêu chuẩn:** Một bộ nhớ dùng để sử dụng thanh ghi tệp, biến cục bộ, lấy mẫu truy dấu mà không cần thẻ nhớ. Sử dụng RAM tiêu chuẩn như là một thanh ghi tệp tạo khả năng truy nhập tốc độ cao như ghi dữ liệu. RAM tiêu chuẩn còn được sử dụng để lưu trữ cho module thu thập lịch sử lỗi.
- Standard ROM:** Một bộ nhớ dùng để lưu trữ dữ liệu như các tham số và chương trình.
- Thẻ nhớ (RAM):** Dùng để lưu trữ trên biến cục bộ, gỡ lỗi dữ liệu, dữ liệu truy dấu SFC, và dữ liệu lịch sử lỗi với các tham số và chương trình.
- Thẻ nhớ (ROM):** Card flash dùng để lưu trữ tham số, chương trình và thanh ghi tệp. Một card ATA lưu trữ các tham số, chương trình, và dữ liệu người dùng của bộ điều khiển khả trình (tệp bình thường).

## Điểm chú ý

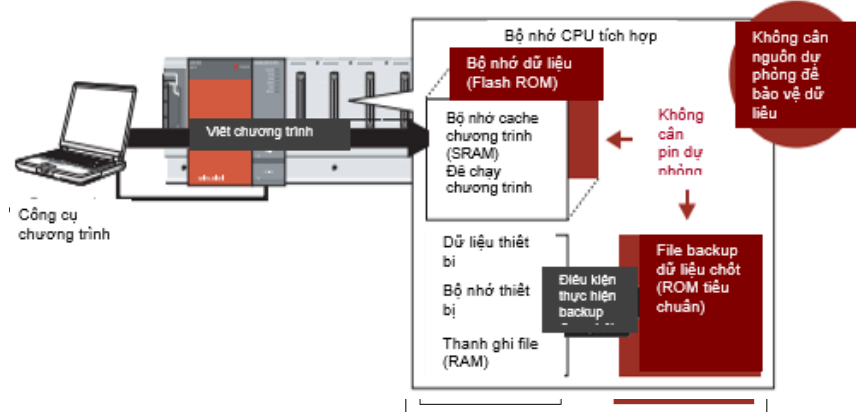
Backup an toàn cho lưu trữ dài hạn

Những chương trình và tệp số được tự động backup đến bộ nhớ chương trình (Flash ROM) mà không cần nguồn dự phòng. Nó ngăn cản việc mất dữ liệu chương trình và tệp số do mất nguồn.

Thời gian cho nguồn dự phòng cũng giảm đi đáng kể.

Thêm vào đó, dữ liệu quan trọng (như dữ liệu thiết bị) có thể back up đến ROM tiêu chuẩn để tránh mất dữ liệu do hết pin trong trường hợp đi vắng lâu ngày.

Dữ liệu backup được phục hồi tự động khi bật nguồn lần sau



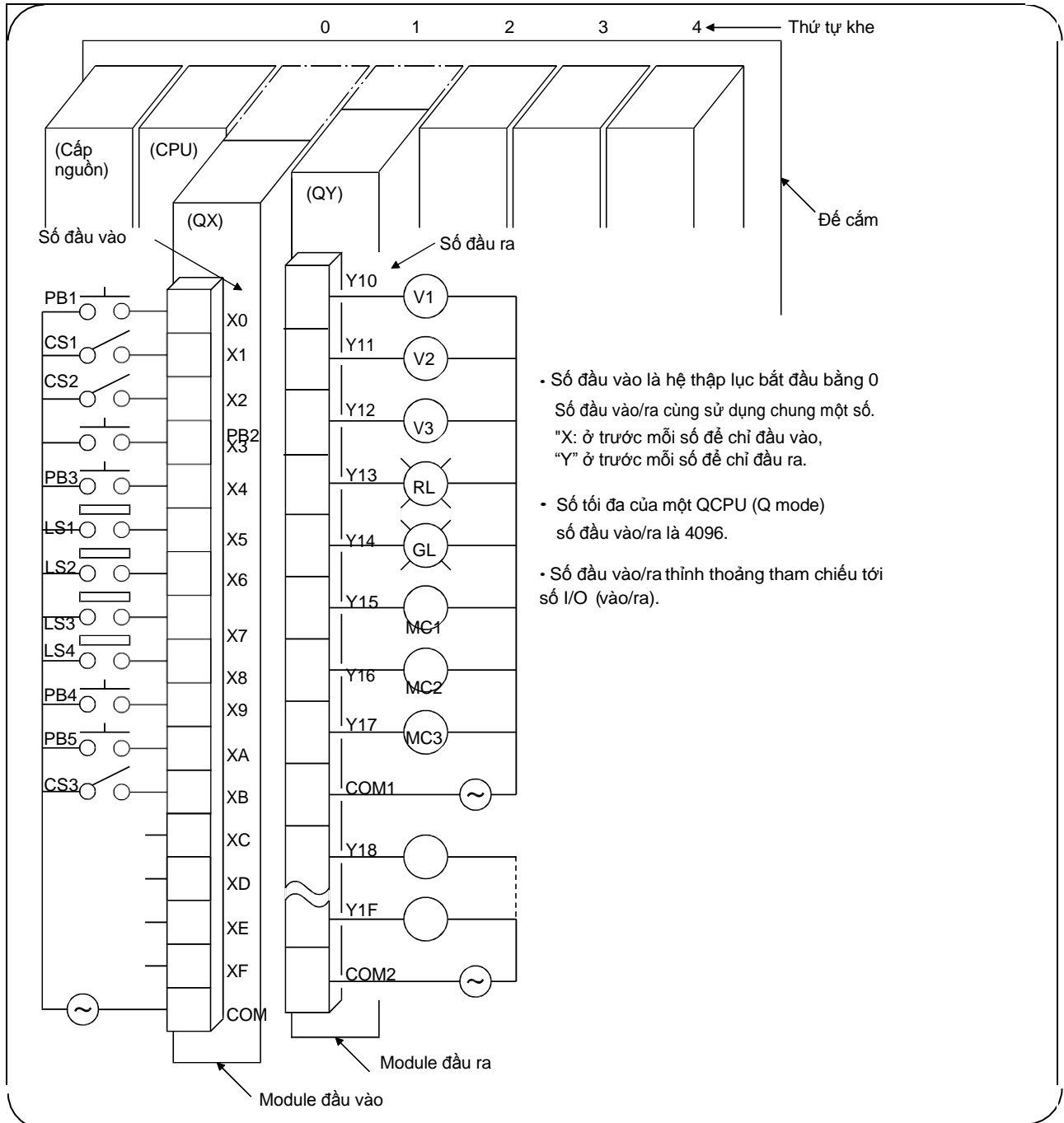


### 1.4 Tín hiệu I/O ngoại vi và số I/O

#### (1) Đi dây của địa chỉ I/O

Tín hiệu đầu ra của thiết bị ngoại vi là đầu vào bộ điều khiển, số đầu vào được quyết định bởi vị trí lắp đặt và số cuối của module đầu vào kết nối và sử dụng trong một chương trình.

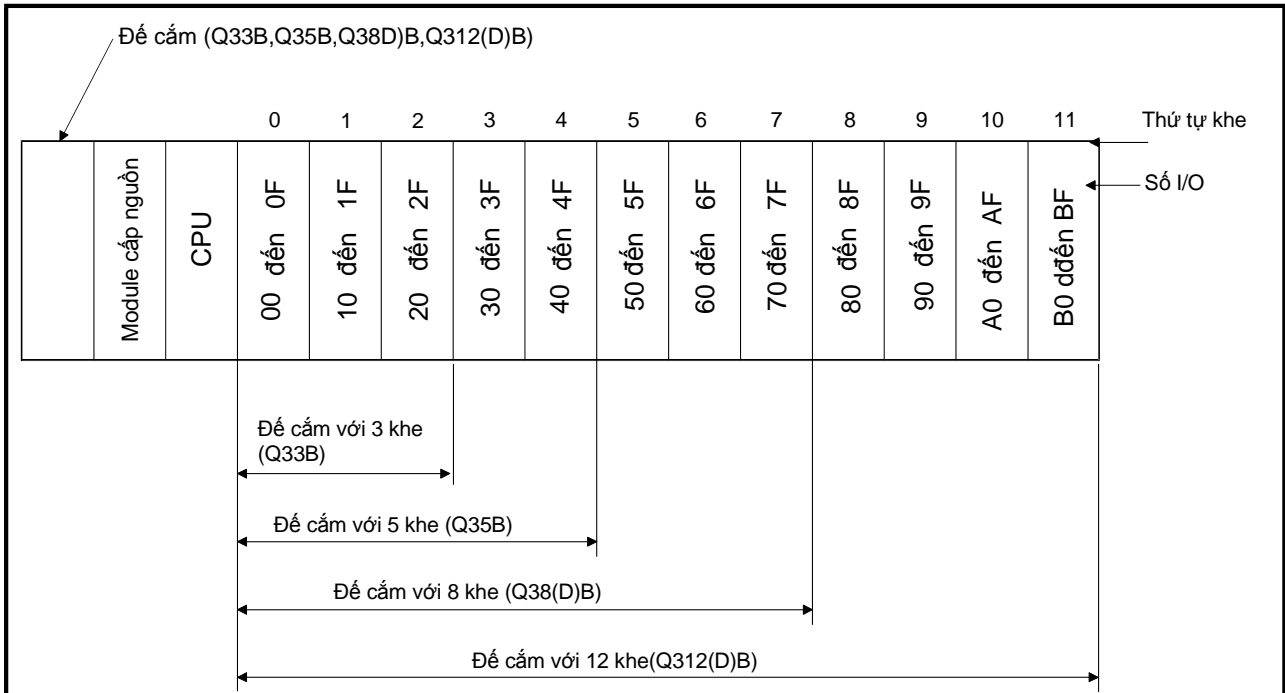
Hoạt động này dẫn đến đầu ra (cuộn dây), sử dụng số đầu ra mà số này được quyết định bởi vị trí cài đặt và số cuối của module đầu ra đến nơi mà module ngoài đầu ra được kết nối..



Hình 1.4 Đi dây biến I/O

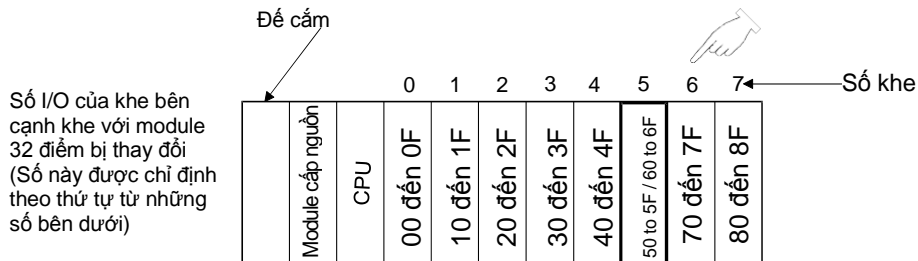
(2) Số I/O của đế cắm

Số I/O của những module I/O được đính vào một khối cơ bản như hình dưới.  
Cấu hình được sử dụng cho cả module I/O và module chức năng thông minh

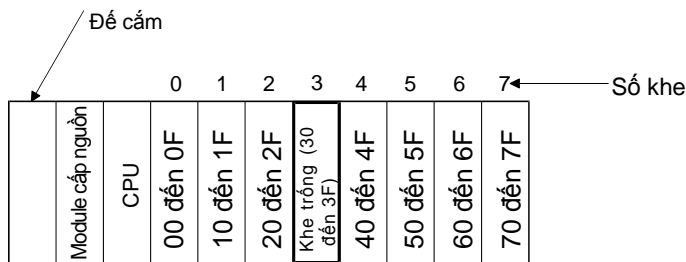


- Số I/O của một khe (một module) được sắp xếp theo thứ tự từ trên xuống trong đơn vị 16 điểm (0 đến F14). Như một chuẩn, module 16 điểm nên được gắn đến tất cả khe.

Ví dụ, hình dưới đây miêu tả số I/O của module 32 điểm được gắn vào khe số 5.

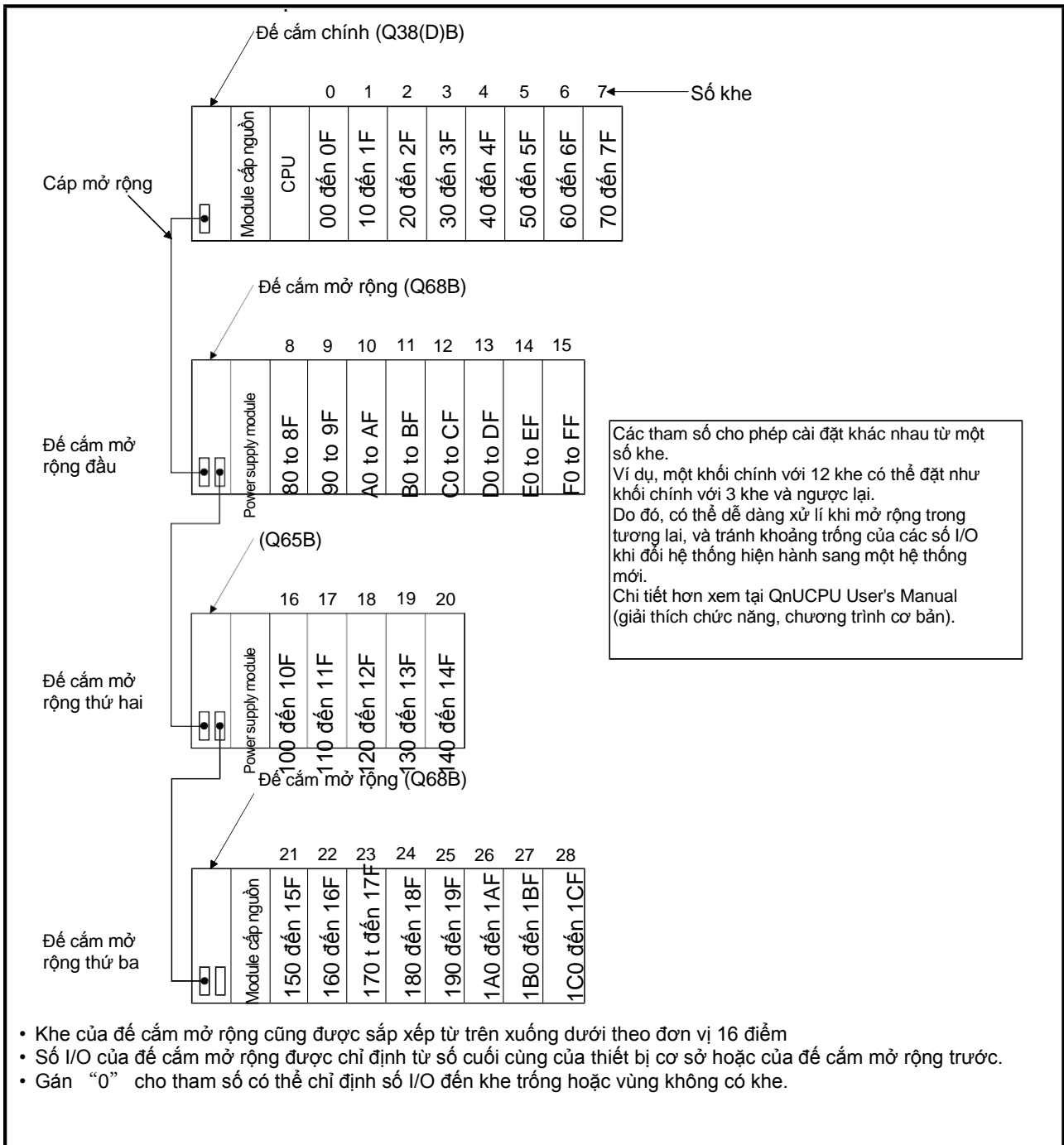


- Số I/O được đưa vào khe trống (một khe không có module I/O nào được gắn vào). Ví dụ, nếu khe thứ 3 còn trống, số I/O được đưa vào như hình dưới (trong phần cài đặt ban đầu) Số của điểm được chỉ định có thể thay đổi bằng cài đặt.



- Đối với cấu hình nhiều CPU (2 đến 4 CPUs), số I/O được chỉ định từ một khe ở bên cạnh khe mà CPU được gắn vào

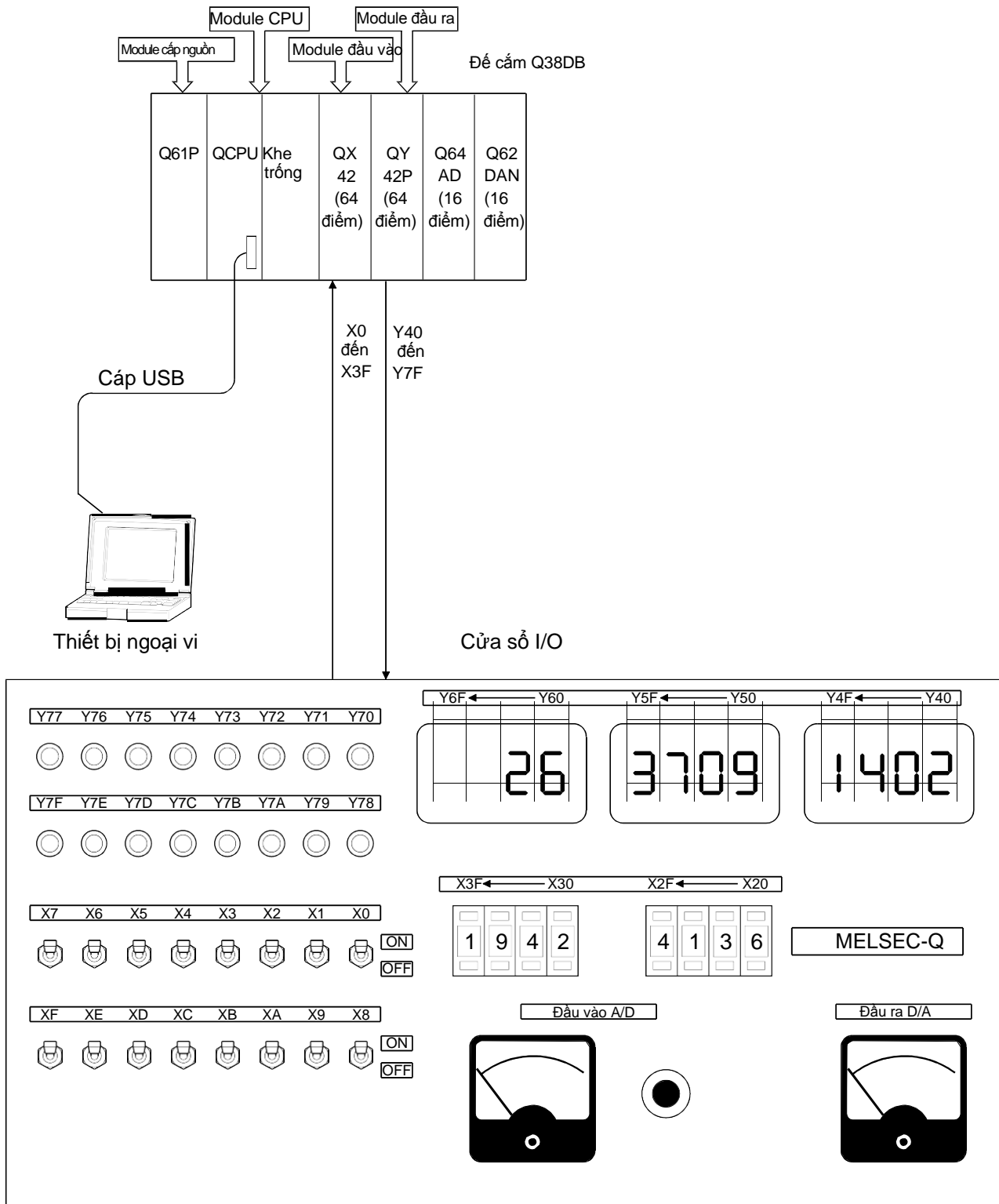
- (3) Số I/O của đế cắm mở rộng  
 Kết nối đến một đế cắm mở rộng khi số khe của đế cắm chính là không đủ.  
 Số I/O được chỉ định theo như cài đặt ban đầu  
 Cấu hình này được áp dụng cho cả module I/O và module chức năng thông minh.



Bảng dưới đây cho thấy số đế cắm mở rộng có thể có.

Loại CPU		Số trạng thái (bao gồm cả cái kết nối với GOT bằng kết nối bus)
Universal Model	Q00UJCPU	2
	Q00UCPU, Q01UCPU, Q02UCPU	4
	Khác với những loại trên	7

1.5 Cấu hình hệ thống và số I/O của thiết bị DEMO



## CHƯƠNG 2 : THAO TÁC TRÊN GXWORKS2

GX Works2 là một công cụ lập trình dùng để thiết kế, gỡ lỗi, và duy trì chương trình trên Window

GX Works2 đã cải thiện chức năng và khả năng thao tác, với những tính năng để sử dụng hơn khi so sánh với GX Developer đã có.

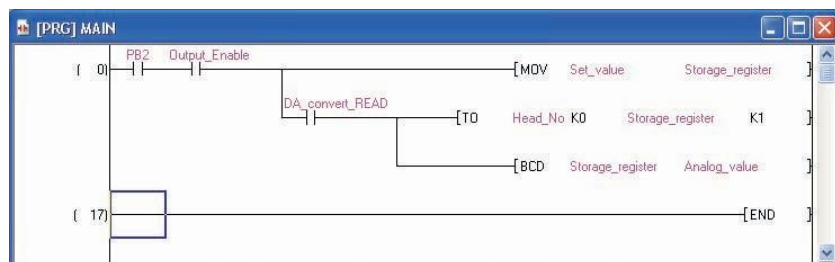
### ■ Chức năng chính của GX Works2

GX Works2 quản lý các chương trình và thông số đầu vào của dự án cho mỗi CPU điều khiển khả trình

#### • Lập trình

Chương trình có thể được tạo ra trong một Dự án đơn giản theo cách tương tự với GX Developer.

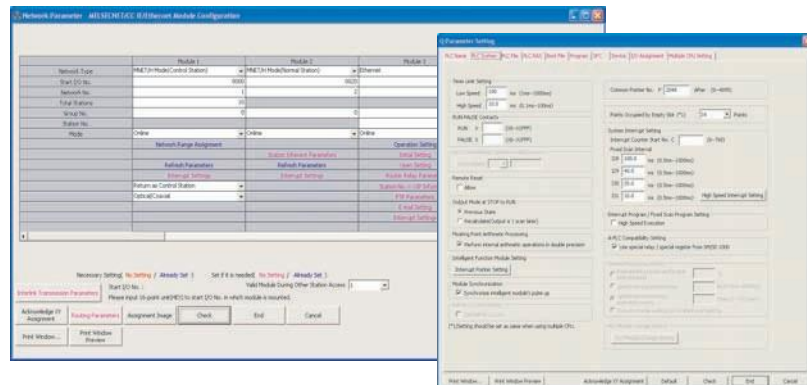
Lập trình cấu trúc trong một dự án cấu trúc cũng khả thi với GX Works2



#### • Cài đặt tham số

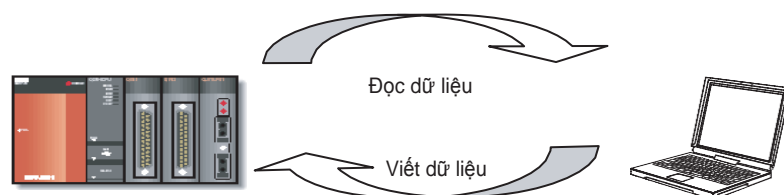
Tham số cho CPU điều khiển khả trình và tham số cấu hình mạng có thể được đặt với GX Works2.

Tham số cho khối chức năng thông minh cũng có thể được cài đặt.



#### • Viết/đọc dữ liệu đến/từ một CPU điều khiển khả trình

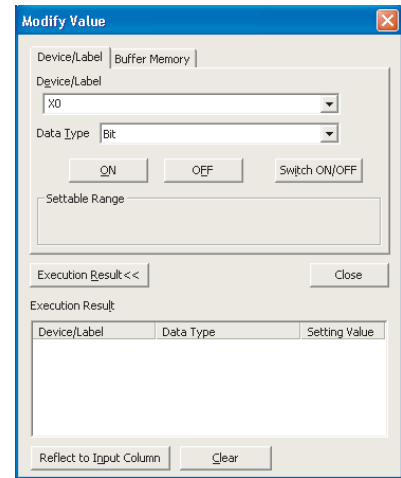
Tạo chương trình tuần tự có thể được viết/đọc từ một CPU điều khiển khả trình sử dụng Read từ PLC/Write của chức năng PLC. Đồng thời, với chương trình thay đổi chức năng trực tuyến, chương trình tuần tự có thể bị thay đổi ngay cả khi CPU điều khiển khả trình đang chạy (RUN).



- **Quan sát/soát lỗi**  
Tạo chương trình tuần tự có thể được viết cho CPU điều khiển khả trình và giá trị của thiết bị khi hoạt động của nó đang được theo dõi trực tuyến/ngoại tuyến.



Chương trình có thể theo dõi và soát lỗi

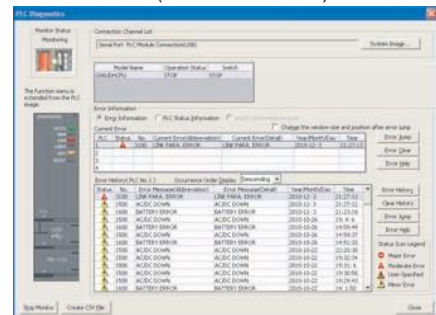


- **Chuẩn đoán**  
Trạng thái lỗi hiện tại và lịch sử lỗi của CPU điều khiển khả trình có thể được chuẩn đoán.  
Với chức năng chuẩn đoán, công việc khôi phục có thể được hoàn thành trong thời gian ngắn.  
Với chức năng theo dõi hệ thống (cho QCPU (Q mode)/LPCPU), thông tin cụ thể trong module chức năng thông tin có thể được lấy về. Điều này giúp cho rút ngắn thời gian phục hồi dữ liệu khi hệ thống đang lỗi.

Chuẩn đoán trạng thái khối điều khiển khả trình CPU (chuẩn đoán PLC)



Chuẩn đoán trạng thái khối điều khiển CPU (chuẩn đoán PLC)



## 2.1 Tính năng trong GX Works2

Phần này giải thích tính năng của GX Works2.

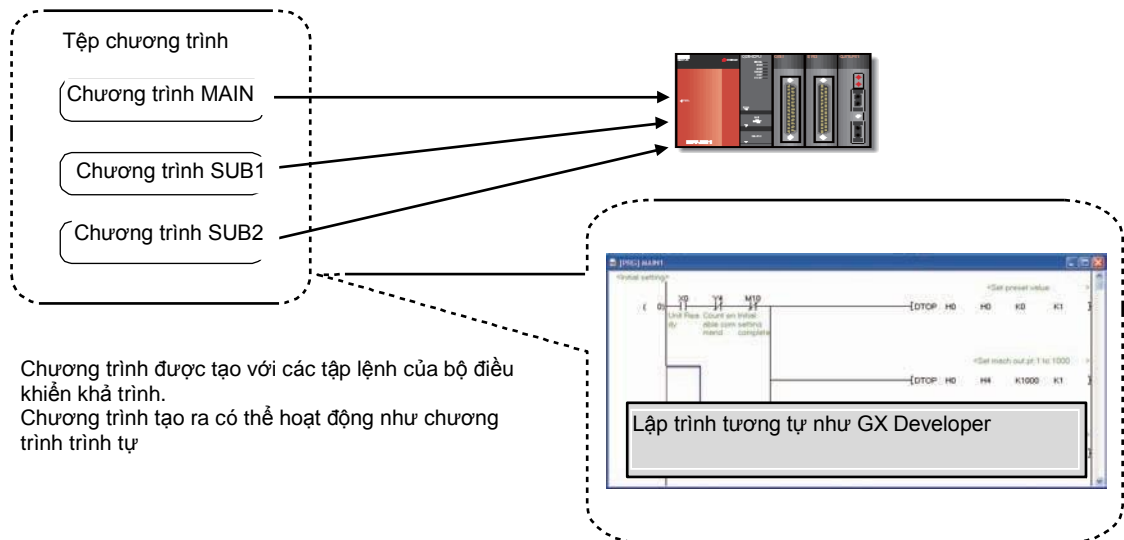
### (1) Loại dự án trong GX Works2

Trong GX Works2, loại dự án có thể được chọn từ những dự án đơn giản hoặc dự án cấu trúc.

#### (a) Dự án đơn giản

Dự án đơn giản tạo ra các chương trình tuần tự sử dụng tập lệnh cho CPU điều khiển khả trình Mitsubishi.

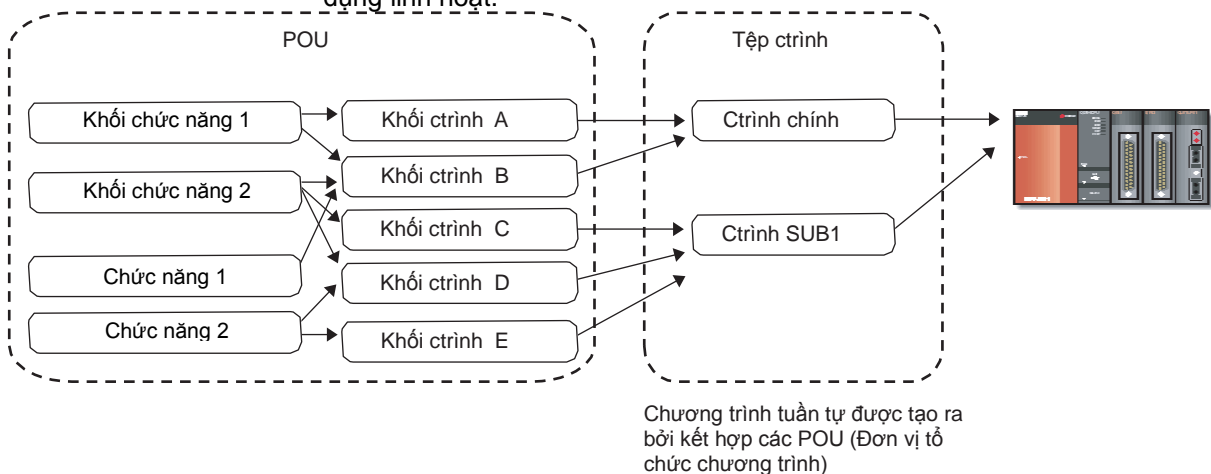
Chương trình trong một dự án đơn giản có thể được tạo ra tương tự như GX developer đã có.



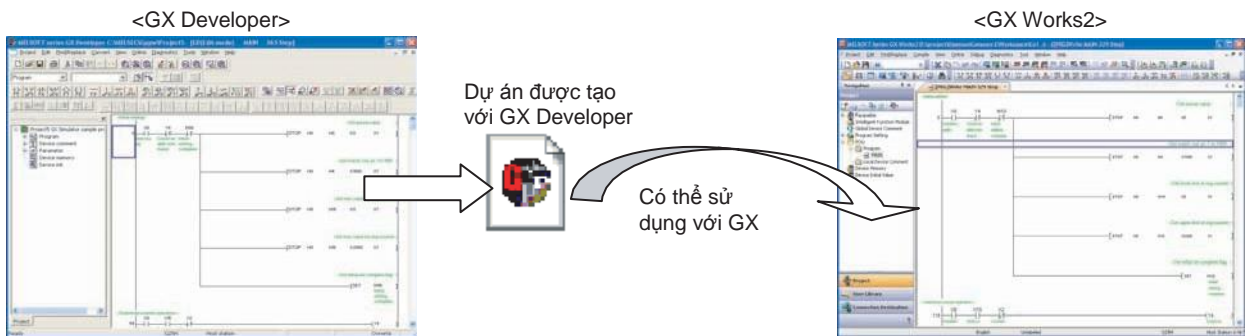
#### (b) Dự án cấu trúc

Trong một dự án cấu trúc, chương trình có thể được tạo ra bởi lập trình cấu trúc.

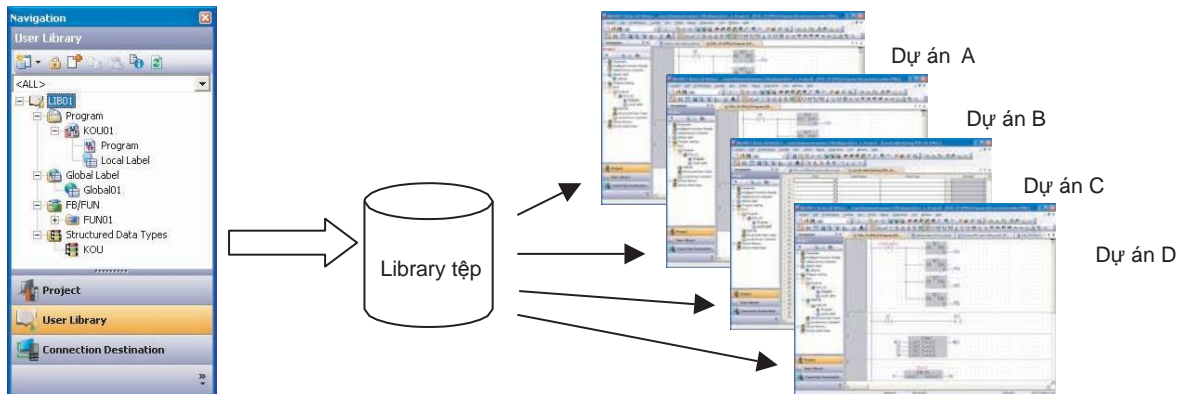
Bằng việc phân mảnh một chương trình điều khiển thành các chương trình thông dụng nhỏ, lập trình ra một chương trình có mức độ quản lý cao và sử dụng linh hoạt.



- (2) Tăng cường khả năng sử dụng của chương trình  
Những dự án được tạo bởi GX Developer đã có được tối ưu trong một dự án đơn giản. Tối ưu những thứ đã có giúp tăng hiệu năng của thiết kế chương trình.



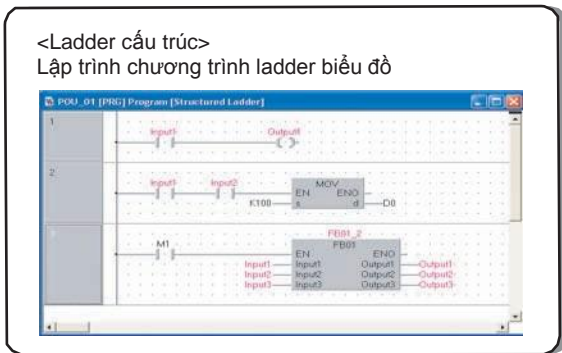
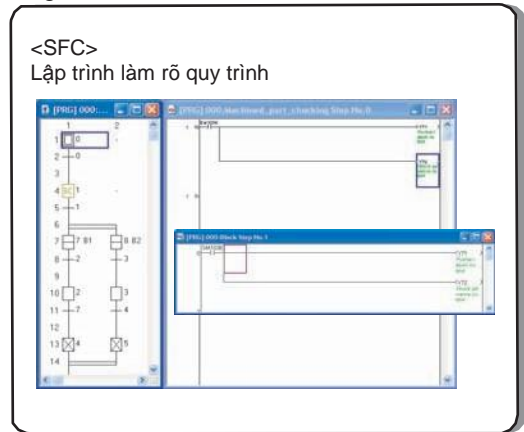
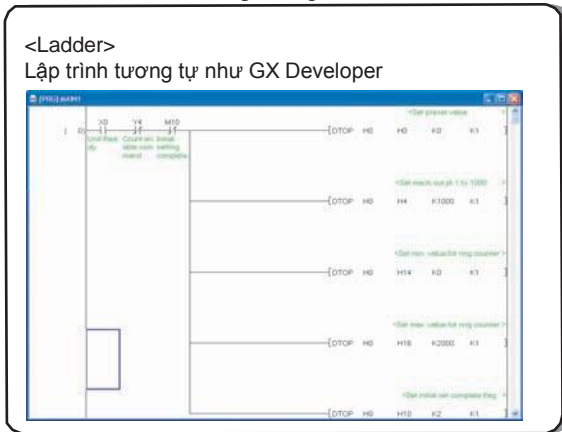
- (3) Chia sẻ Đơn vị tổ chức chương trình (POU) như một thư viện  
Trong một dự án cấu trúc, những chương trình, nhãn quốc tế, và cấu trúc thường sử dụng có thể được đăng kí như là thư viện người dùng. Tối ưu thư viện người dùng này giảm thời gian cần thiết để tạo ra những chương trình.





(4) Các ngôn ngữ lập trình

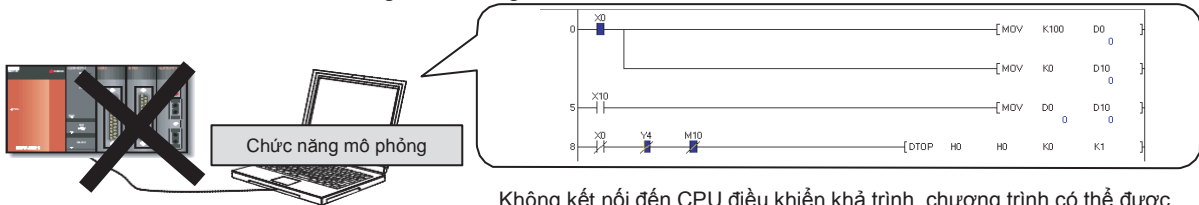
Có nhiều ngôn ngữ lập trình có thể sử dụng được với GX Works2 nên việc chọn ra ngôn ngữ tối ưu sẽ dựa vào khả năng điều khiển



(5) Tính năng khác

(a) Soát lỗi ngoại tuyến

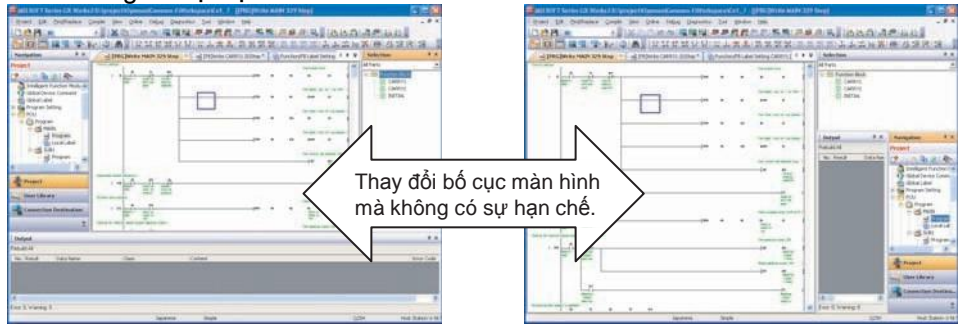
Soát lỗi ngoại tuyến sử dụng chương trình mô phỏng với GX Works2. Nó cho phép soát lỗi để đảm chương trình tuần tự tạo ra bảo hoạt động bình thường mà không kết nối GX Works2 đến CPU điều khiển khả trình.



Kết nối đến khối điều khiển khả trình là không bắt buộc

Không kết nối đến CPU điều khiển khả trình, chương trình có thể được theo dõi và soát lỗi tương tự như cách soát lỗi bằng CPU điều khiển khả trình.

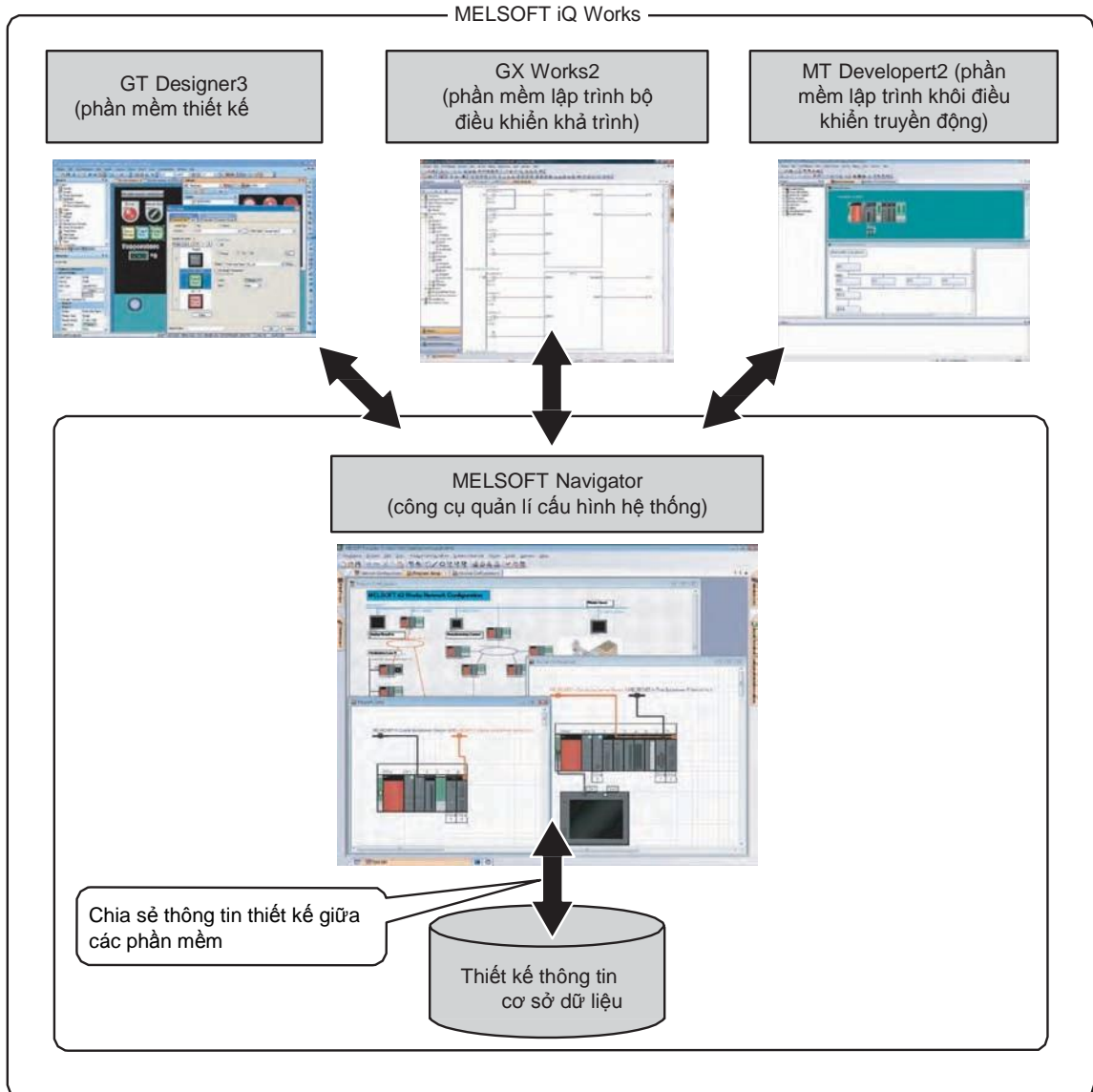
- (b) Bố cục màn hình có thể tùy chỉnh theo mong muốn của người dùng  
Cửa sổ lưu động cho phép thay đổi bố cục màn hình của GX Works2 mà không có sự hạn chế .



## 2.1.1 MELSOFT iQ Works

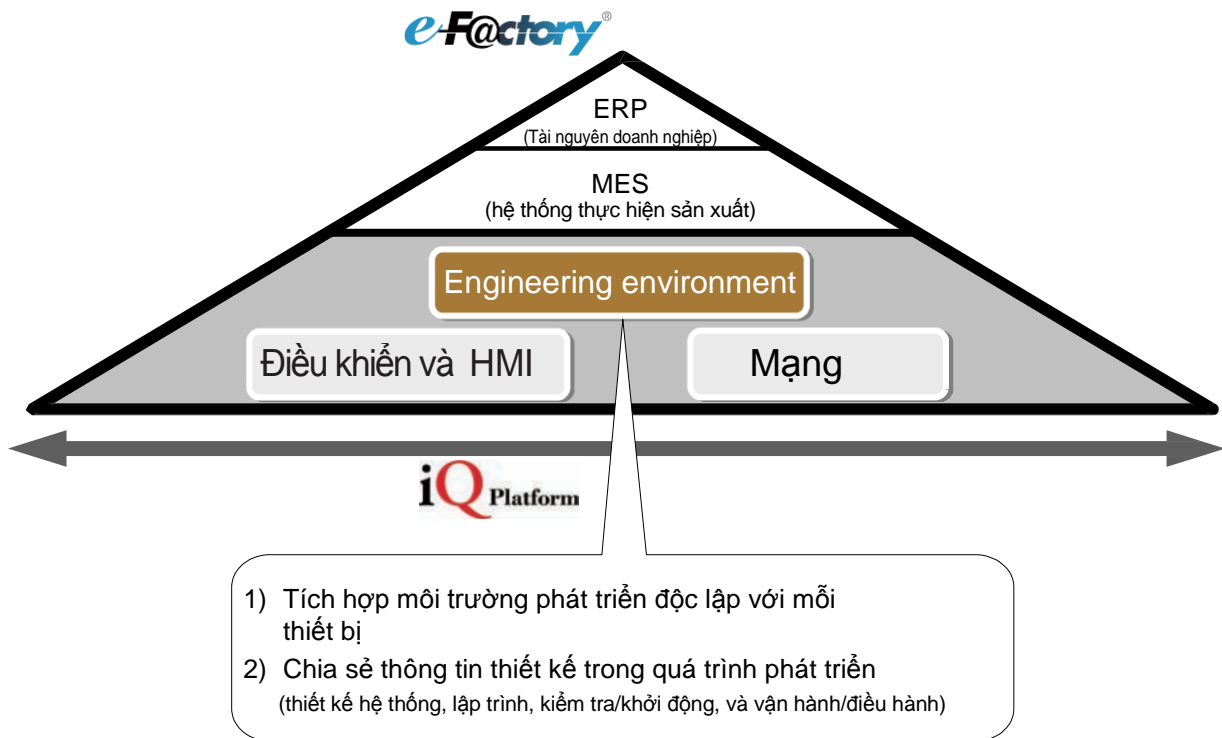
MELSOFT iQ Works tích hợp phần mềm kỹ thuật (GX Works2, MT Developer2, và GT Designer3).

Chia sẻ thông tin như thiết kế hệ thống và lập trình trong toàn bộ hệ thống điều khiển giúp cải thiện hiệu năng của việc thiết kế chương trình và hiệu năng của việc lập trình, giúp làm giảm giá thành.



Gợi ý	
	Để khởi động MELSOFT Navigator và mỗi phần mềm kỹ thuật, click vào nút Start và theo hướng dẫn dưới đây
• MELSOFT Navigator:	[Ứng dụng MELSOFT] → [MELSOFT iQ Works] → [MELSOFT Navigator]
• GX Works2:	[Ứng dụng MELSOFT] → [GX Works2] → [GX Works2]
• MT Developer2:	[Ứng dụng MELSOFT] → [MT Works2] → [MT Developer2]
• GT Designer3:	[Ứng dụng MELSOFT] → [GT Works3] → [GT Designer3]

[Mục đích của môi trường kĩ thuật]



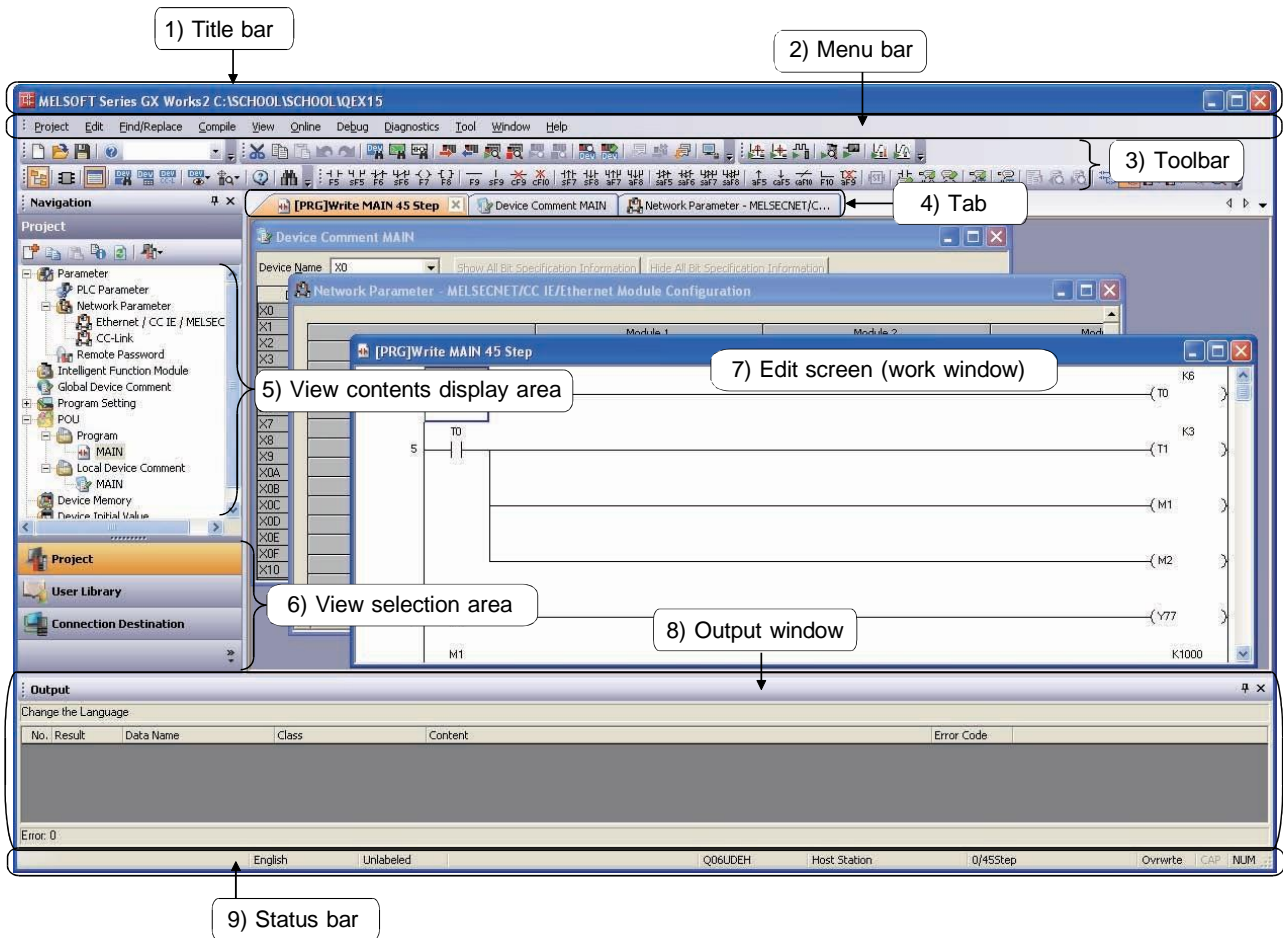
Gợi ý

**iQ Platform** là một FA khái niệm tích hợp của MITSUBISHI ELECTRIC.

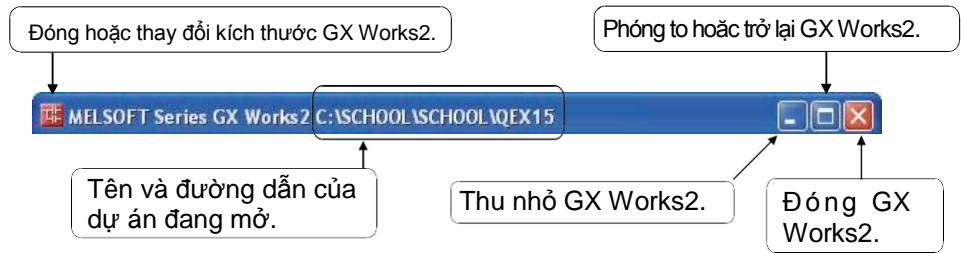
Tích hợp Q/tăng cường chất lượng/thông minh&nhanh/đột phá&Tìm kiếm

## 2.2 Kiến thức căn bản để sử dụng GX Works2

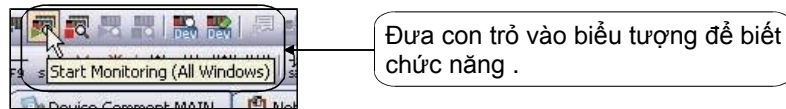
### 2.2.1 Cấu hình màn hình trong GX Works2



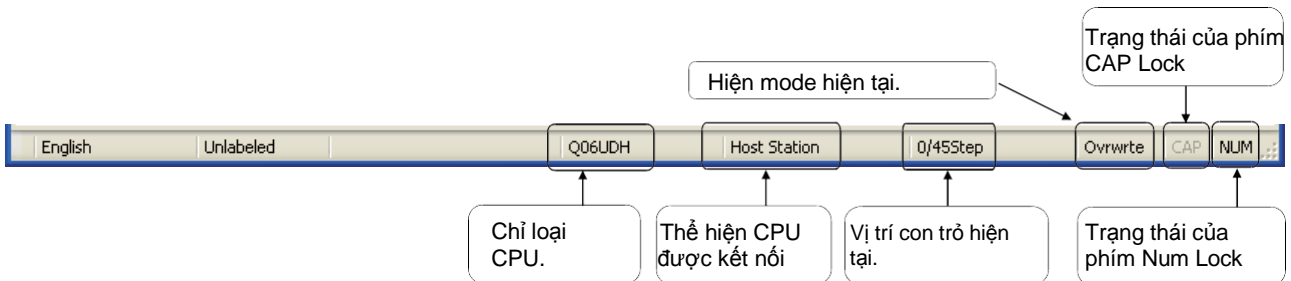
- 1) Thanh tiêu đề  
Thanh tiêu đề thể hiện tên của dự án đang sử dụng



- 2) Thanh Menu  
Thanh menu được sử dụng khá thường xuyên khi làm việc với GX Works2. Click vào thanh menu để chọn những chức năng khác nhau từ thanh menu
- 3) Thanh công cụ  
Thanh công cụ gồm những biểu tượng rất trực quan và dễ sử dụng cho những chức năng thông dụng. Nó khiến các thao tác nhanh hơn.



- 4) Tab  
Khi làm việc với nhiều cửa sổ đang mở, nó thể hiện theo dạng trình duyệt tab . Click vào tab sẽ kích hoạt cửa sổ làm việc tương ứng.
- 5) Xem nội dung hiển thị  
Xem nội dung hiển thị thể hiện nội dung của trình hiển thị hiện tại.
- 6) Xem vùng đang chọn  
Xem vùng đang chọn vùng thể hiện nội dung của vùng hiện tại
- 7) Màn hình chỉnh sửa (cửa sổ làm việc)  
Màn hình chỉnh sửa thể hiện nhiều loại màn hình như màn hình tạo chương trình và màn hình tạo comment để chỉnh sửa biểu đồ ladder, chú thích, và tham số.
- 8) Cửa sổ đầu ra  
Cửa sổ đầu ra thể hiện sự biên dịch và kiểm tra kết quả (như lỗi và cảnh báo).
- 9) Thanh trạng thái  
Thanh trạng thái thể hiện thông tin trạng thái của GX Works2.



## 2.2.2 Chỉnh sửa ladder

Phần này nói về màn hình làm việc của GX Works2 chỉnh sửa ladder và hoạt động cơ bản của nó.

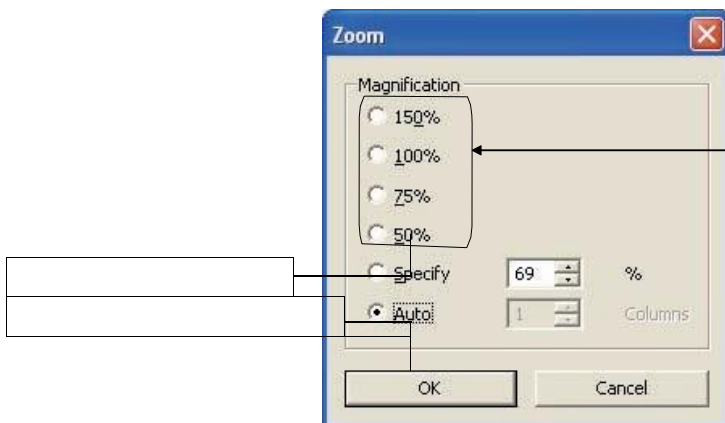
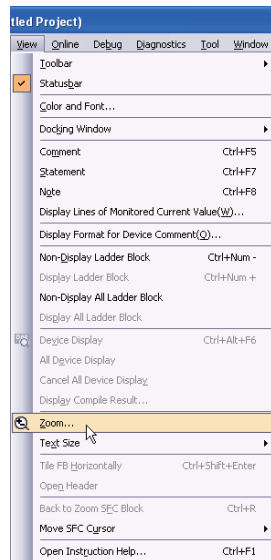
### (1) Chỉnh sửa màn hình



### (2) Thay đổi kích thước màn hình trên màn hình chỉnh sửa Kích thước màn hình của màn hình chỉnh sửa bị thay đổi.

1) Click [View] → [Zoom].

Hộp thoại Zoom xuất hiện.

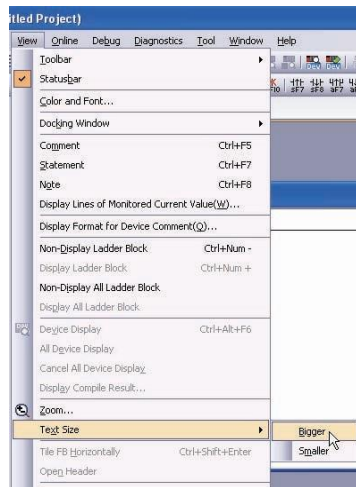


Thay đổi kích thước hiển thị theo chế độ phóng to theo %

Thay đổi kích thước hiển thị theo số phóng to tùy chọn (từ 50 đến 150%).

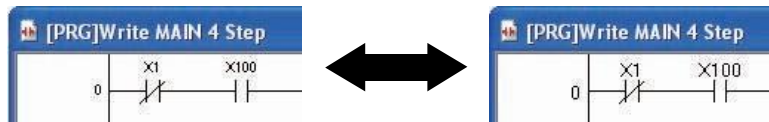
Thêm độ rộng của ladder tự động để hiển thị toàn bộ ladder.

(3) Thay đổi cỡ chữ trên màn hình chỉnh sửa  
Cỡ chữ trên màn hình có thể thay đổi



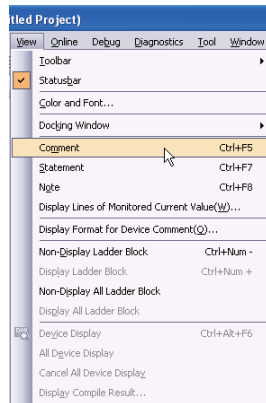
1) Chọn [View] → [Text Size] → [Bigger]/[Smaller].

Cỡ chữ được thay đổi mỗi lần thực hiện bước này trong phạm vi 10 lần..



(4) Hiện/ẩn chú thích

Ghi chú thiết bị (nhãn ghi chú), ghi chú, và biểu ngữ có thể hiện ra hoặc ẩn đi.



1) Chọn [View] → [Comment]/[Statement]/[Note].

**Gợi ý**

**Hiện/ẩn chú thích**  
Chú thích có thể hiện hoặc ẩn bằng thao tác sau.  
[Tool] → [Option] → "Program Editor" → "Ladder" → "Comment"

Comment Display Items

Device Comment       Note

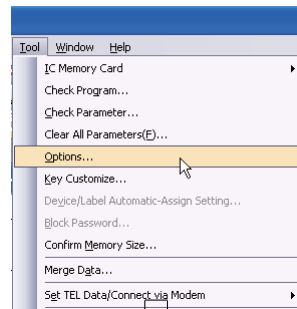
Statement

\* Chi tiết của thao tác này được giải thích trong trang sau



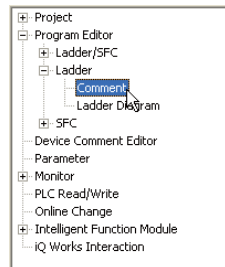
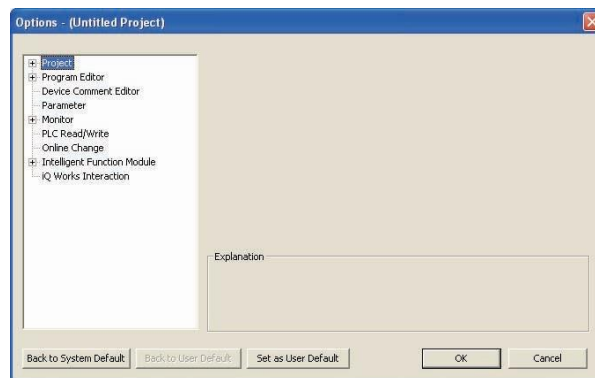
(5) Đặt số dòng và cột để hiện comment

Cài đặt Option cho phép thay đổi cỡ số hàng và cột để hiện ghi chú thiết bị



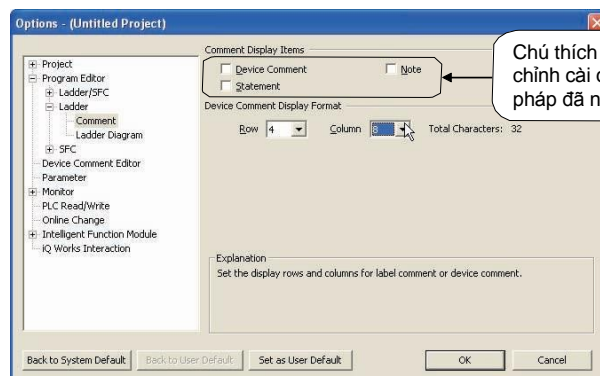
1) Click [Tool] → [Option].

Màn hình Option hiện lên..



2) Click "Program Editor" → "Ladder" → "Comment".

Màn hình để đặt định dạng thể hiện ghi chú thiết bị xuất hiện.

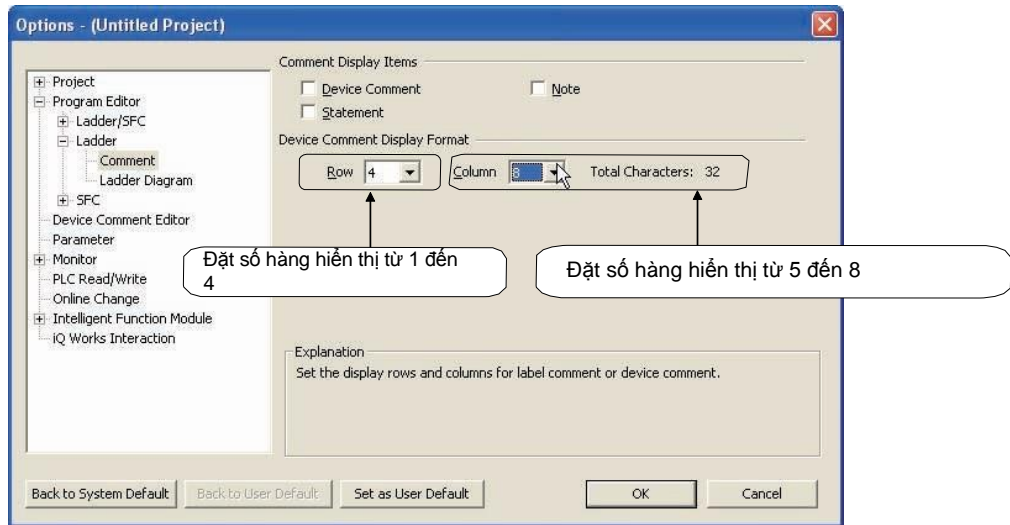


Chú thích hiện hoặc ẩn phụ thuộc vào chỉnh cài đặt phần này cùng với phương pháp đã nói đến ở trang trước.

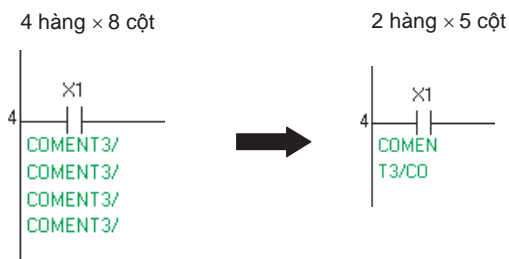


(Đến trang sau)

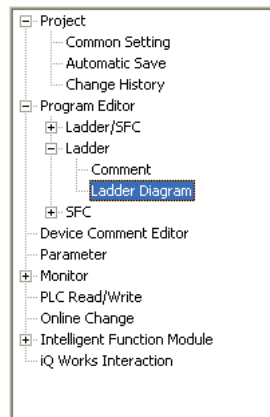
(Từ trang trước)



Ví dụ)

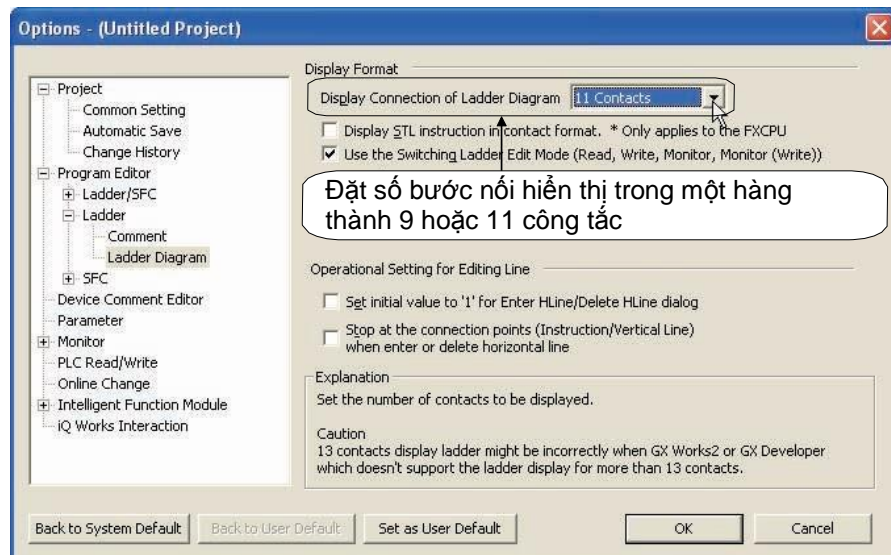


- (6) Cài đặt số công tắc hiển thị trong chương trình ladder  
Cài đặt phần này cho phép đổi số công tắc hiển thị trong 1 hàng.

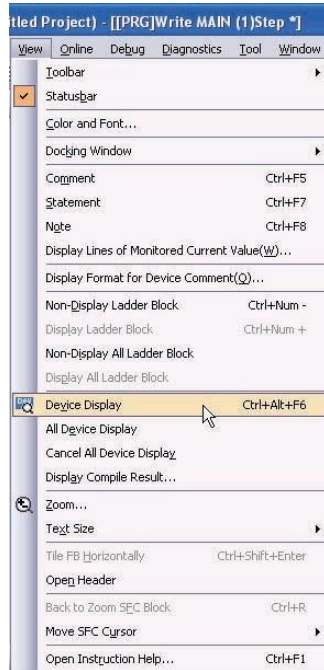


- 1) Click "Program Editor" → "Ladder" → "Ladder Diagram" trong màn hình Option.

Màn hình cài đặt Display Format cho biểu đồ ladder hiện ra.



- (7) Đổi tên nhãn hiển thị và thiết bị hiển thị  
 Một chương trình sử dụng nhãn có thể đổi giữa tên nhãn và tên thiết bị.  
 Chú thích nhãn và chú thích thiết bị được sử dụng sẽ cho chú thích trên màn hình tương ứng.  
 Những thiết bị sử dụng với trình biên dịch có thể chuyển đổi chế độ hiển thị tên nhãn sang hiển thị tên thiết bị.



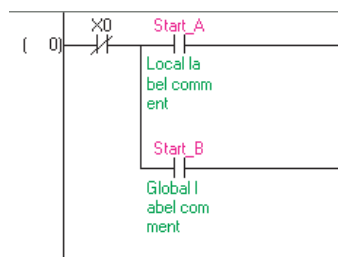
1) Click [View] → [Device Display].

Màn hình để cài đặt Display Format cho biểu đồ hình thang hiện ra.

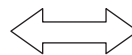
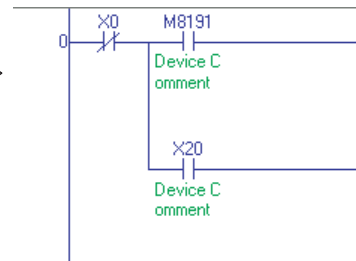


Ví dụ)

Hiện tên nhãn



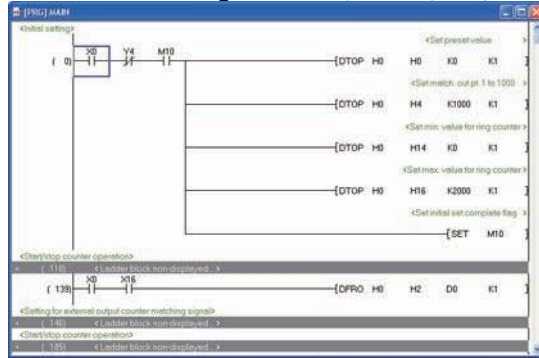
Tên thiết bị



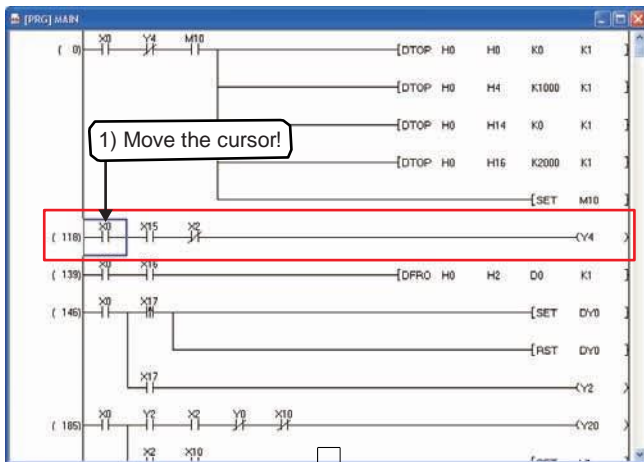
Gợi ý

Hiện/ẩn nhãn ghi chú và ghi chú thiết bị  
 Để kiểm tra chú thích nhãn và chú thích thiết bị, cài đặt phần chú thích hiển thị (xem phần 2.2.2(4)).

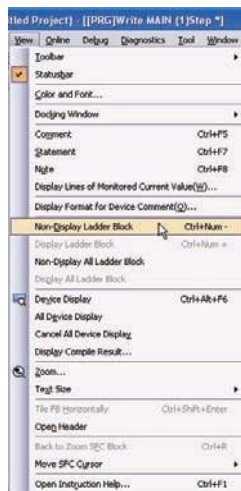
- (8) Ấn một khối ladder  
 Khối ladder sau khi biến đổi có thể ẩn đi  
 Khối ladder đi cùng với câu lệnh được đặt ẩn đi còn câu lệnh hiện lên.



(a) Ấn một khối ladder



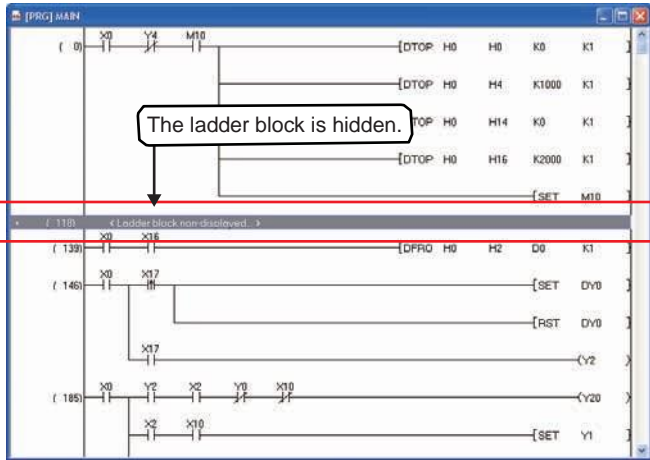
- 1) Đưa chuột đến khối ladder muốn ẩn.



- 2) Click [View] → [Non-Display Ladder Block].

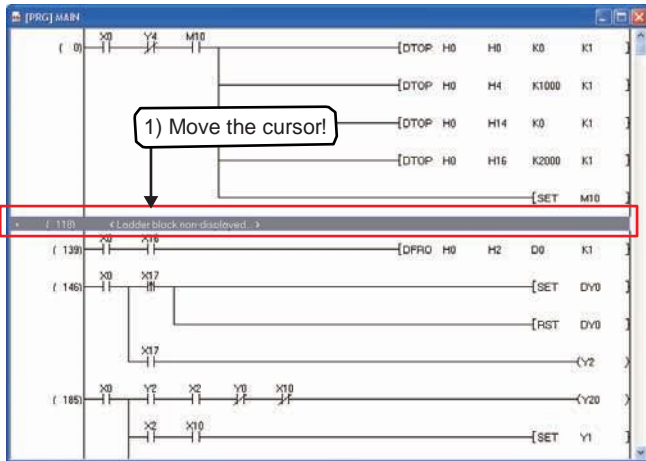
(Đến trang sau)

(Từ trang trước)



3) Khối ladder được chọn đã bị ẩn đi.

(b) Hủy bỏ lệnh ẩn khối ladder



1) Di chuyển chuột đến khối ladder ẩn hiện đang màu xám.

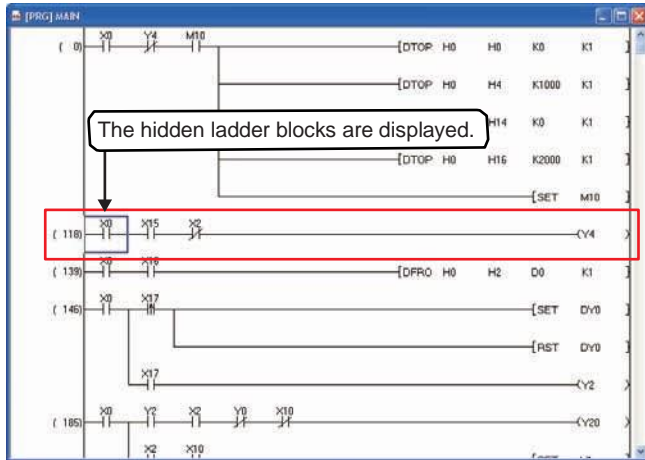


2) Click [View] → [Display Ladder Block].



(Đến trang sau)

(Từ trang trước)



3) Khối ladder được chọn đã bị ẩn đi..

#### Gợi ý

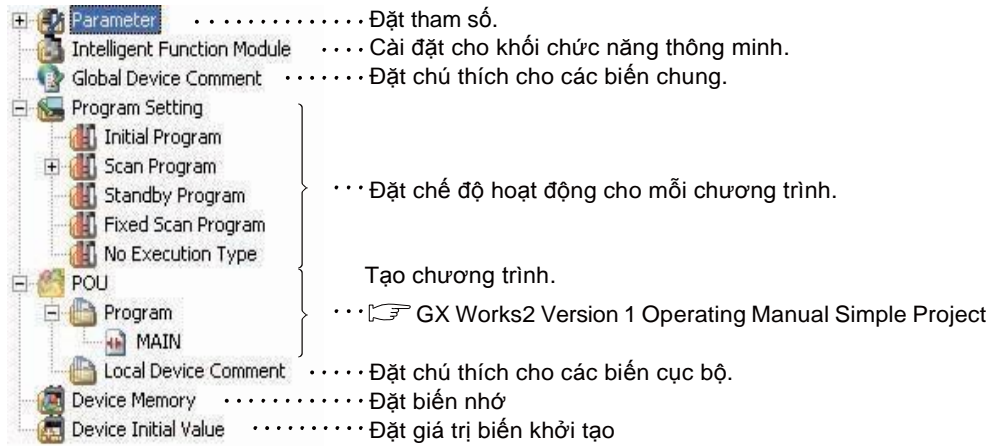
Hiện/ẩn khối ladder

- Nhiều khối ladder có thể được ẩn hoặc hiện..
- Tất cả khối ladder có thể hiện hoặc ẩn bằng việc [View] → [Display All Ladder Block]/[Non-Display All Ladder Block].
- Khối ladder còn có thể hiện hoặc ẩn bởi Phải chuột → [Displaying/hiding ladder **blocks**].

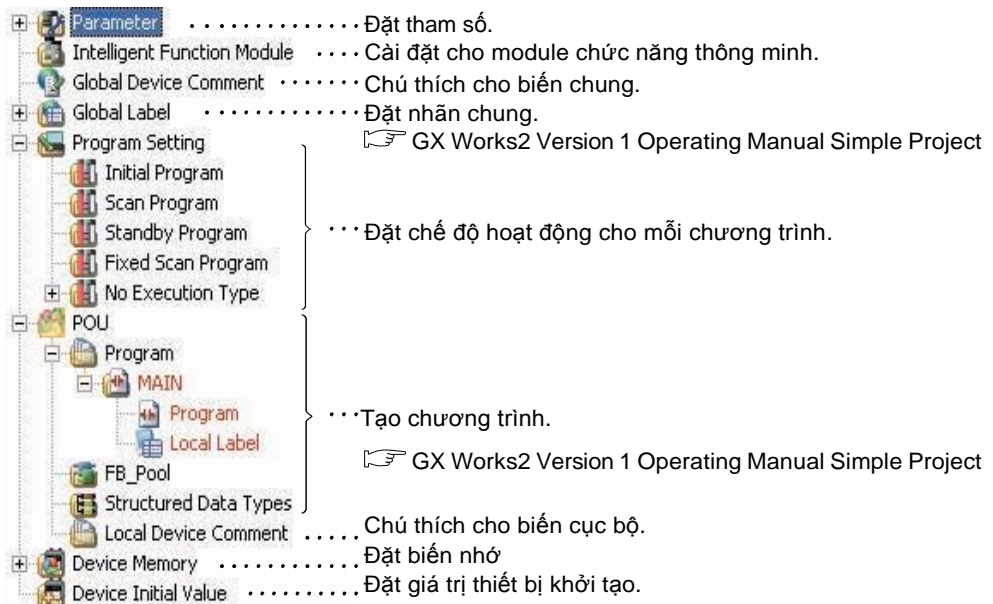
### 2.2.3 Dự án

Phần này giải thích cấu hình của một dự án được hiển thị theo dạng cây. Nội dung hiển thị khác nhau theo loại bộ điều khiển khả trình. Dưới đây là một ví dụ cho một dự án đơn giản của QCPU (Q mode)).

< Dự án đơn giản (không nhãn) >



< Dự án đơn giản (có nhãn) >





1) Một dự án với mỗi GX Works

Một GX Works2 có thể chỉnh sửa chỉ một dự án.

Để chỉnh sửa hai hay nhiều hơn cùng lúc, chạy số chương trình GX Works2 bằng với số dự án.

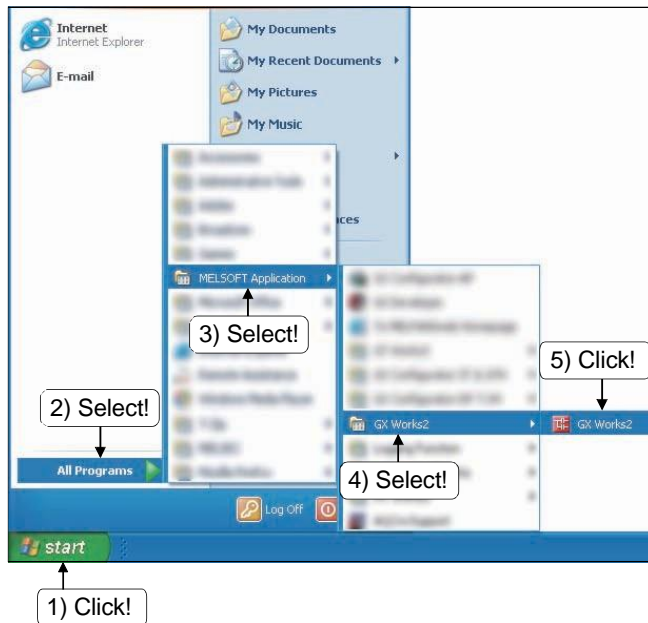
2) Chú thích biến


Chú thích biến của GX Works2 được phân loại vào chú thích biến chung và chú thích biến cục bộ.

Loại chú thích	Số chú thích	Mô tả
Chú thích biến chung	1	Một ghi chú thiết bị tự động được tạo ra khi một dự án được tạo Chú thích chung được đặt để sử dụng ghi chú thiết bị chung giữa nhiều chương trình.
Chú thích cục bộ	Bằng với số chương trình.	Một ghi chú thiết bị được tạo bởi người dùng. Không một ghi chú thiết bị cục bộ được có trước khi tạo một dự án mới. Vì vậy, hãy tạo một ghi chú thiết bị cục bộ nếu cần thiết. Đặt tên giống với chương trình tuần tự..... .....

## 2.3 Thao tác trước khi tạo chương trình tuần tự

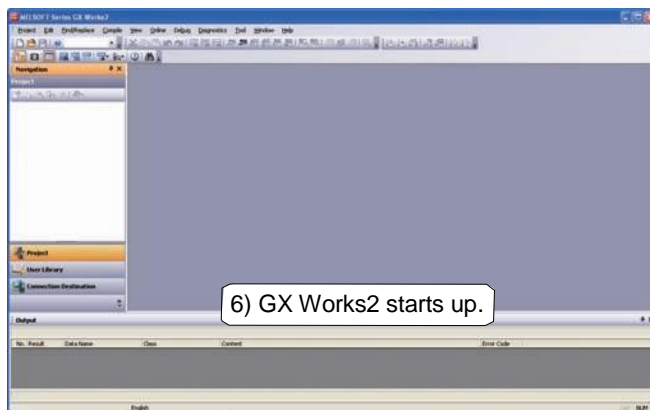
### 2.3.1 Khởi động GX Works 2



- 1) Click biểu tượng 
- 2) Chọn [All Programs].
- 3) Chọn [MELSOFT Application].
- 4) Chọn [GX Works2].

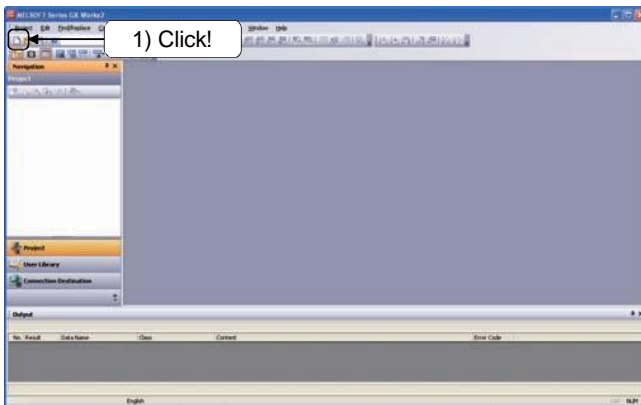
Đặt con trỏ chuột lên thư mục cần vào để hiện ra menu  
(Click trái chuột hay double-click chuột là không cần thiết)


- 5) Click [GX Works2].

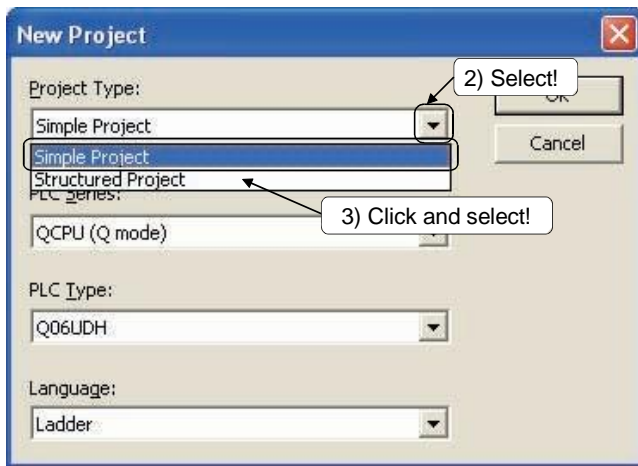


- 6) GX Works2 khởi động.

### 2.3.2 Tạo một dự án mới

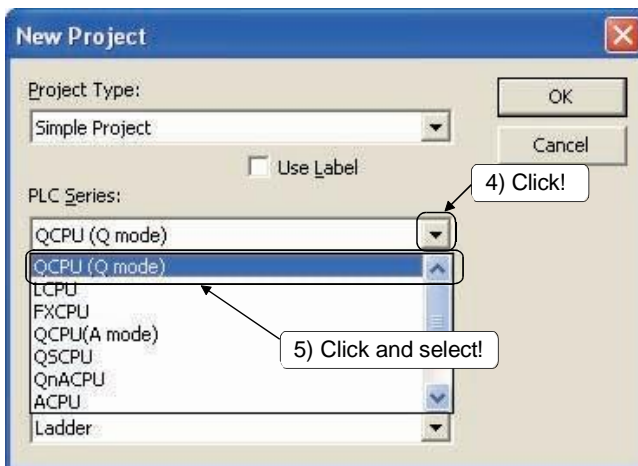


1) Click  trên thanh công cụ và chọn [Project] → [New Project] ( **Ctrl** + **N** ).



2) Click vào "Project Type".

3) Danh mục "Project Type" hiện lên. Chọn "Simple Project".



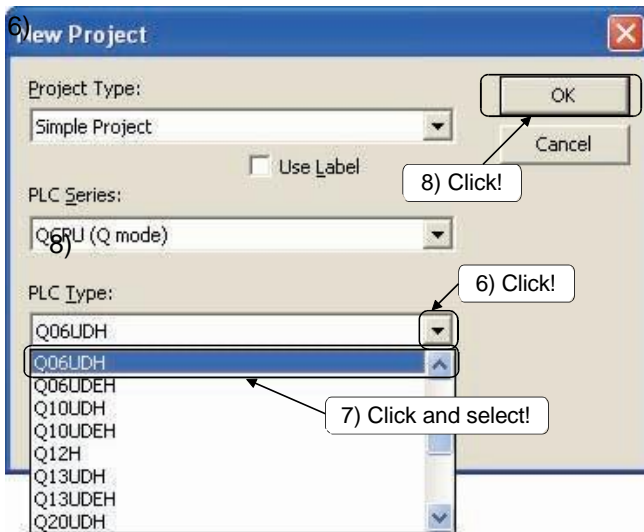
4) Click the "PLC Series" list button.

5) Danh mục "PLC Series" hiện lên. Select "QCPU (Q mode)".




(Đến trang sau)

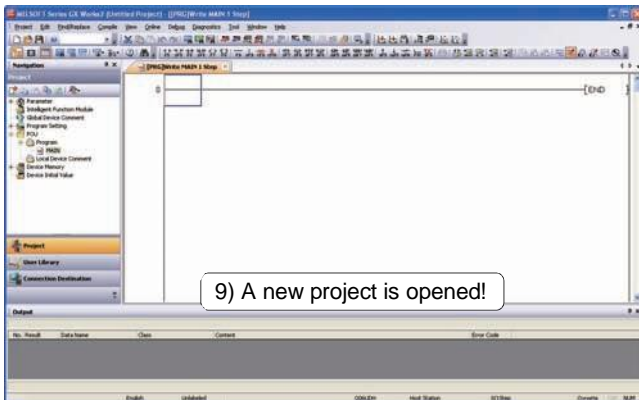
(Từ trang trước)



Click vào nút "PLC Type".

7) Danh sách "PLC Type" hiện ra. Chọn "Q06UDH"..

Click vào 

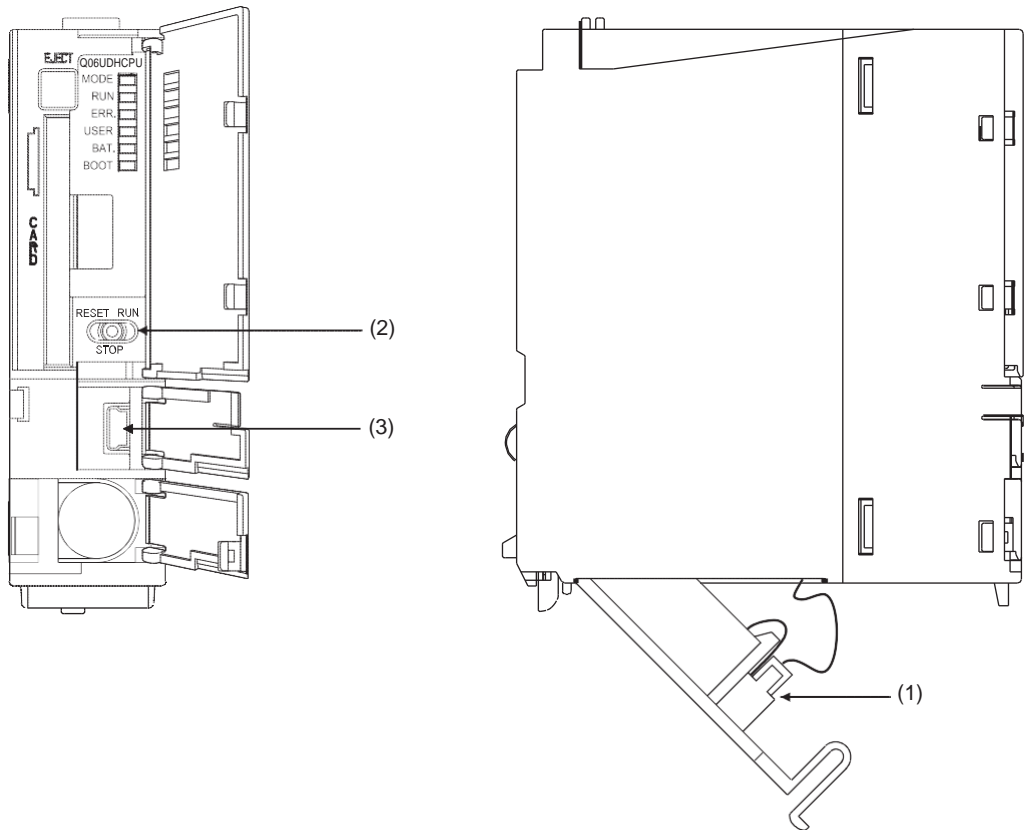


9) Một dự án mới được tạo ra.

## 2.4 Các bước chuẩn bị để khởi động CPU

Cần đặt chế độ công tắc và xóa bộ nhớ built-in trước khi viết một chương trình lên CPU.

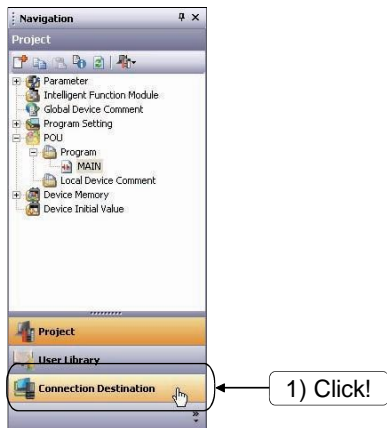
Kết nối và đặt chế độ công tắc từ (1) đến (3) như hình dưới (Hình vẽ bên dưới là một ví dụ của Q06UDHCPU).



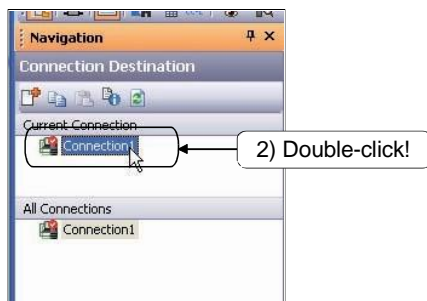
- (1) Kết nối đến nguồn  
Kết nối đến pin bởi vì pin bị tháo ra khi vận chuyển
- (2) Đặt công tắc  
Đặt công tắc RUN/STOP/RESET về vị trí STOP.
- (3) Kết nối cáp USB

#### (4) Cài đặt Connection Destination

Phần này giải thích làm thế nào để cài đặt kết nối giữa PC và PLC.

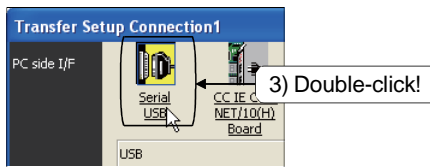


- 1) Click chọn "Connection Destination" trong vùng chọn lựa ở cửa sổ Navigation.

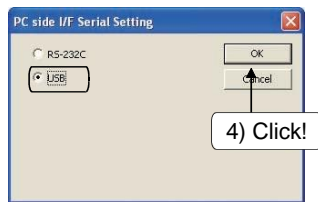


- 2) Cửa sổ Connection Destination hiện lên. Click đúp chuột "Connection1" trong Current Connection.

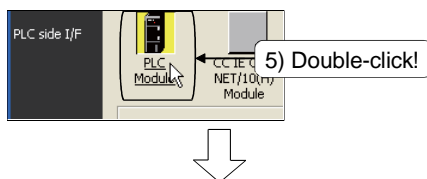
Hộp thoại Transfer Setup hiện lên.



- 3) Click đúp chuột "Serial USB" của PC side I/F.



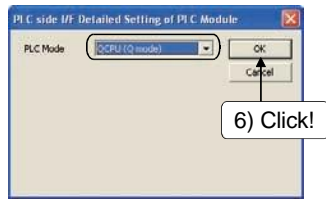
- 4) Hộp thoại PC side I/F Serial Setting hiện lên. Chọn "USB" và click vào nút **OK**



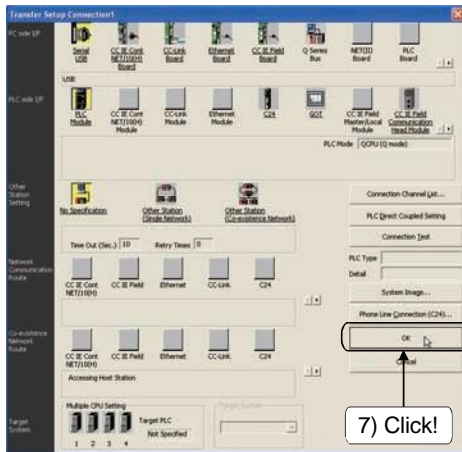
- 5) Nhấp đúp chuột vào "PLC Module" của PLC side I/F.

(Đến trang sau)

(Từ trang trước)



6) Hộp thoại PLC side I/F Detailed Setting of PLC Module hiện lên. Chọn "QCPU (Q mode)" và click chọn **OK**.

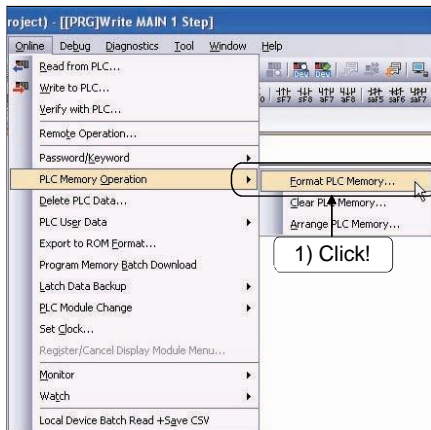


7) Click **OK**

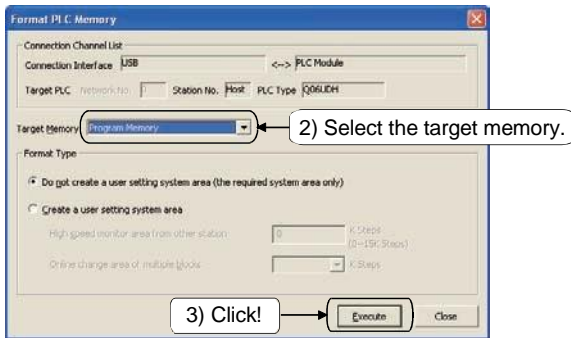
(5) Xóa bộ nhớ tích hợp trên CPU

Phần này mô tả làm thế nào để xóa bộ nhớ chương trình của QCPU

- 1) Click [Online] → [PLC Memory Operation] → [Format PLC Memory].



- 2) Hộp thoại Format PLC Memory hiện ra. Chọn "Program Memory" ở ô Target Memory.



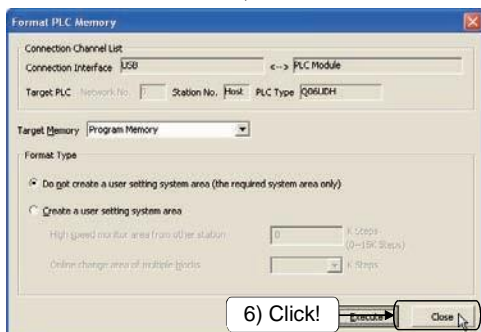
- 3) Click vào [Execute].



- 4) Click vào [Yes] để bắt đầu xóa..



- 5) Để hoàn tất quá trình xóa, khi hộp thoại bên trái hiện ra, chọn [OK].

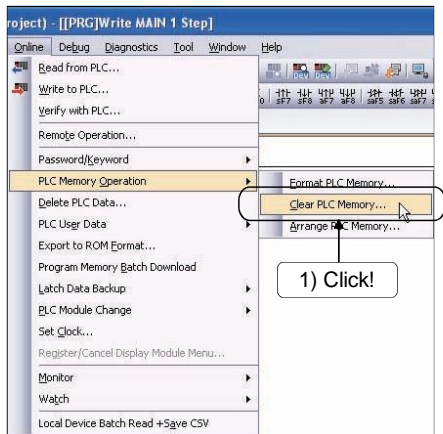


- 6) Click vào [Close] và đóng hộp thoại.

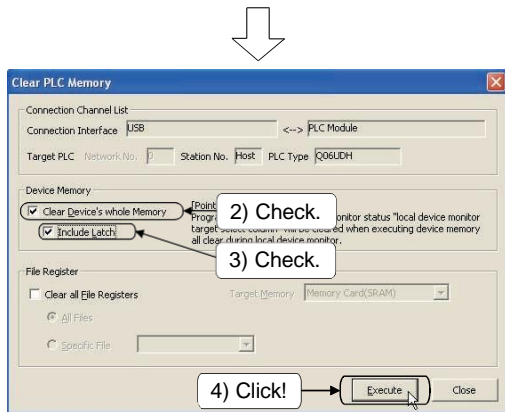


(6) Xóa bộ nhớ thiết bị từ CPU

Phần này mô tả làm thế nào để xóa hết biến nhớ từ QCPU..



- 1) Click [Online] → [PLC Memory Operation] → [Clear PLC Memory].



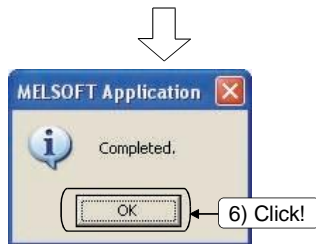
- 2) Hộp thoại Clear PLC Memory hiện ra. Kiểm tra "Clear Device's whole memory" đã được chọn.

- 3) Chọn "Include Latch".

- 4) Click vào **Execute**

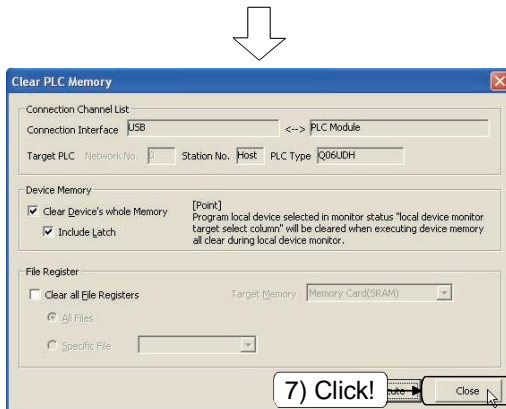


- 5) Click vào **Yes** để xóa biến chốt.



- 6) Khi lệnh xóa được hoàn tất hộp thoại ở bên trái hiện ra.

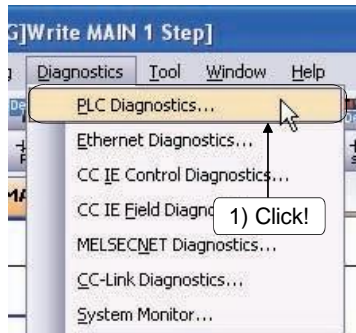
Click vào **OK**



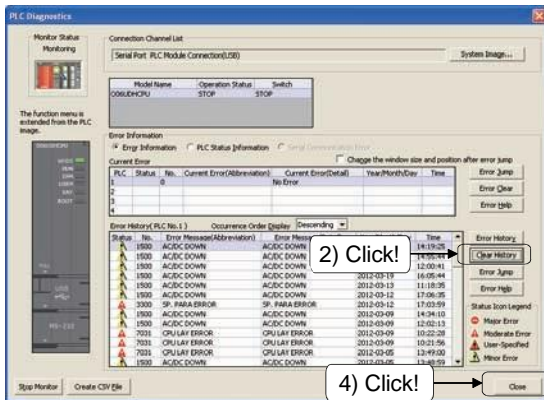
- 7) Click vào **Close** để đóng hộp thoại.

(7) Xóa lịch sử lỗi trong CPU

Phần này diễn giải làm thế nào để xóa dữ liệu lịch sử lỗi được lưu trên QCPU.



1) Click [Diagnostics] → [PLC Diagnostics].

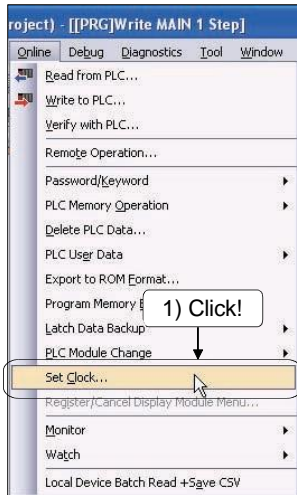


2) Hộp thoại PLC Diagnostics hiện ra.  
Click vào **Clear History** .

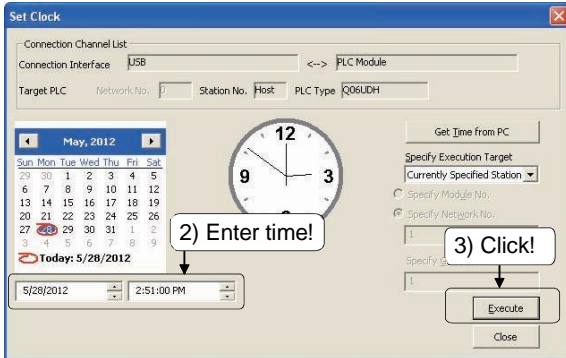
3) Hộp thoại xác nhận hiện ra.  
Click vào **Yes** ..

4) Click **Close** để đóng hộp thoại.

- (8) Đặt thời gian cho bộ điều khiển khả trình CPU.  
 Đặt ngày, tháng năm, giờ, phút, giây, số ngày một tuần cho đồng hồ trên bộ điều khiển khả trình.  
 Để sử dụng chức năng đồng hồ, sử dụng GX Works2 hoặc một chương trình tuần tự. Cài đặt hoặc đọc dữ liệu đồng hồ trong GX Works2.



- 1) Click [Online] → [Set Clock] để hộp thoại Set Clock hiện lên.

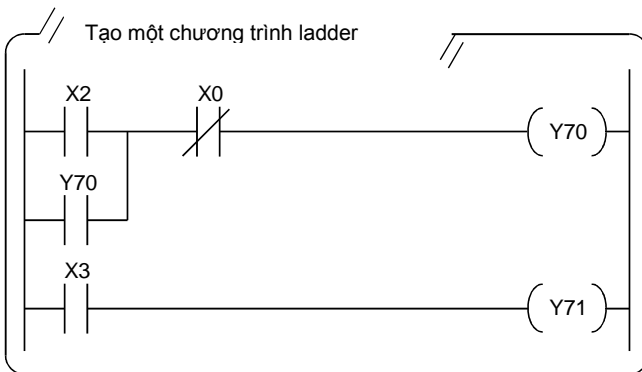


- 2) Đặt ngày, tháng, năm, giờ, phút, giây, và ngày trong tuần trong hộp thoại Set Clock.

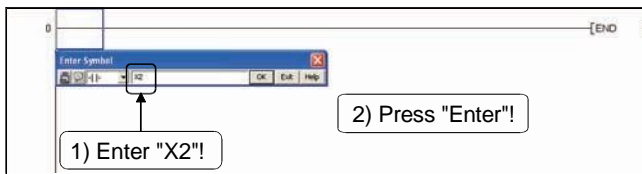
- 3) Click the **Execute** button.

## 2.5 Tạo một chương trình ladder

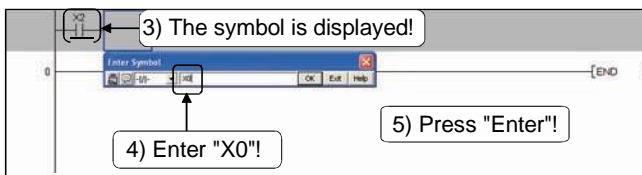
### 2.5.1 Tạo một chương trình ladder sử dụng phím chức năng (bàn phím)




Xem hướng dẫn bên dưới để tạo chương trình ladder như hình bên.



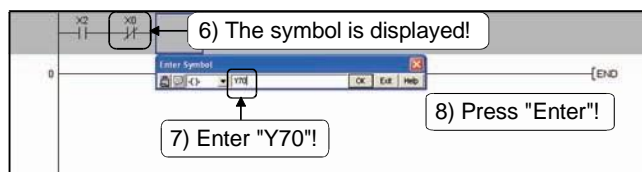
- 1) Nhấn **F5** để mở cửa sổ Enter Symbol.  
Nhập "X2".  
Nếu đánh sai bất kì kí tự nào, nhấn **Esc** và gõ lại.
- 2) Nhấn **Enter** để xác nhận.



- Nút OK hoặc Exit được sử dụng để xác nhận hoặc hủy bỏ.

3) Biểu tượng vừa nhập (  ) hiện ra.

- 4) Nhấn **Shift** + **F5**, và nhập "X0".
- 5) Nhấn **Enter** để xác nhận.



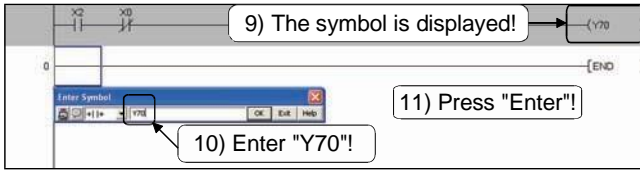
6) Biểu tượng vừa nhập (  ) is displayed.

- 7) Nhấn **F7**, và nhập "Y70".
- 8) Nhấn **Enter** để xác nhận..



(Đến trang sau)

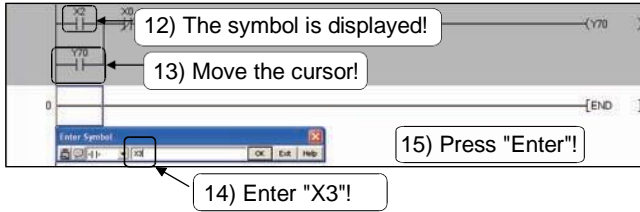
(Từ trang trước)



9) Biểu tượng vừa nhập (Y70) hiện ra.

10) Nhấn nút **F6**, và nhập "Y70".

11) Nhấn **Enter** để xác nhận.



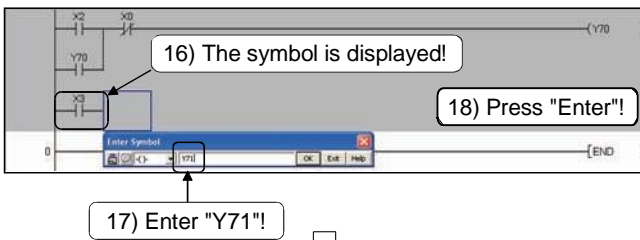
12) Biểu tượng vừa nhập (Y70) hiện ra.

13) Di chuyển chuột đến dưới biểu tượng



14) Nhấn **F5**, và nhập "X3".

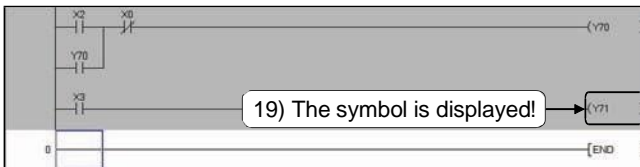
15) Nhấn **Enter** để xác nhận.



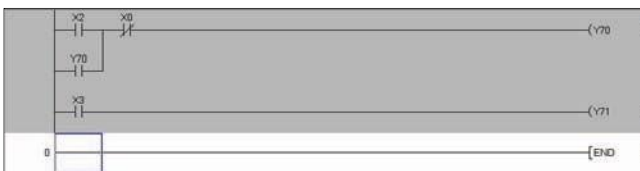
16) Biểu tượng vừa nhập (X3) hiện ra.

17) Nhấn **F7**, và nhập "Y71".

18) Nhấn **Enter** để xác nhận.

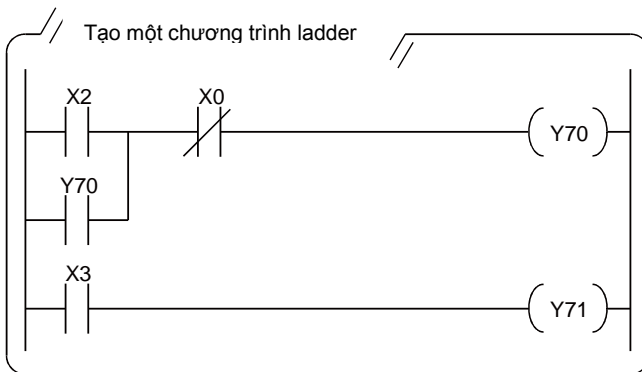


19) Biểu tượng vừa nhập (Y71) hiện ra.

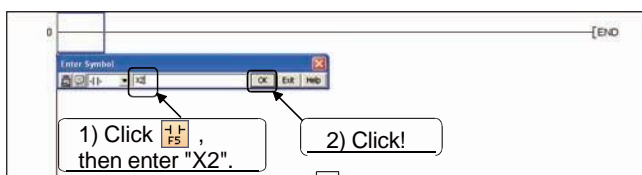





20) Kết thúc quy trình.

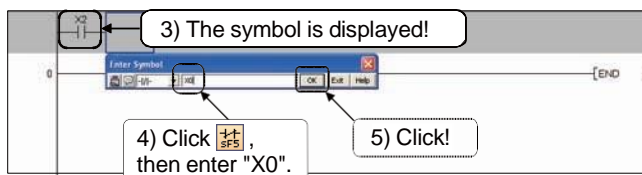
## 2.5.2 Tạo một chương trình ladder sử dụng thanh công cụ






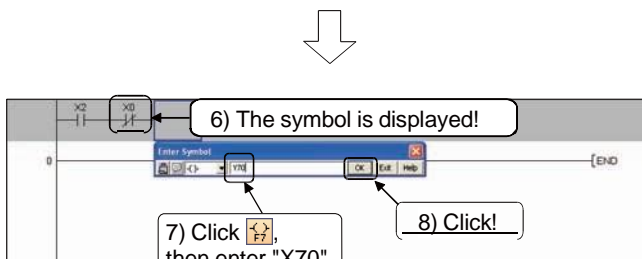
Làm theo những bước dưới đây để tạo ra một chương trình như hình bên trái.






- 1) Click  để mở cửa sổ Enter Symbol.  
Nhập "X2".  
Nếu trong lúc lần bấm nhấn vào nút nào đó, click  Exit
- 2) Click nút  để xác nhận.



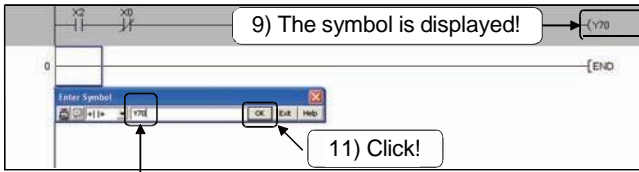
- 3) Biểu tượng vừa  hiện ra.  
nhập
- 4) Click  trên thanh công cụ và nhập "X0".
- 5) Click nút  .




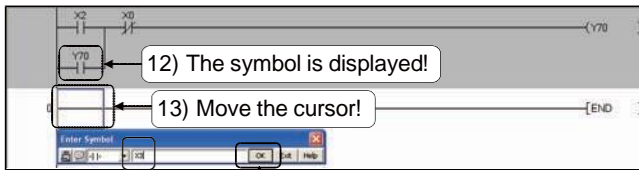
- 6) The entered symbol () is displayed.
- 7) Click  on the toolbar, and enter "Y70".
- 8) Click the  button.


(Đến trang sau)

(Từ trang trước)

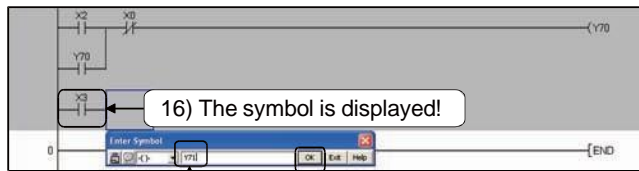



10) Click , then enter "X0".



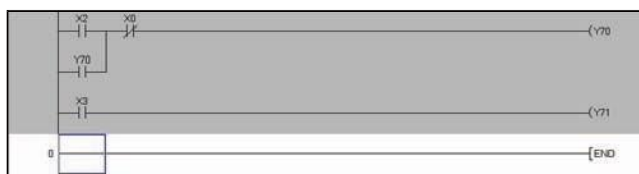
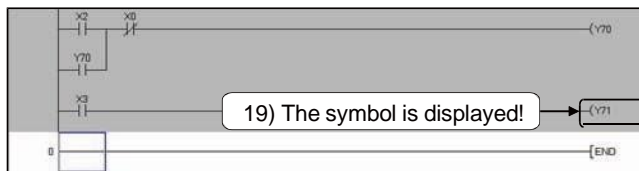
14) Click , then enter "X3".

15) Click!




17) Click , then enter "Y71".

18) Click!




9) Biểu tượng vừa nhập ( $\text{Y70}$ ) hiện ra.

10) Click  trên thanh công cụ và nhập "Y70".

11) Click nút  .


12) Biểu tượng vừa nhập ( $\text{X0}$ ) hiện ra.

13) Đưa chuột đến dưới biểu tượng  $\text{X0}$ .

14) Click  trên thanh công cụ và nhập "X3".

15) Click nút  .

16) Biểu tượng vừa nhập ( $\text{X3}$ ) hiện ra.

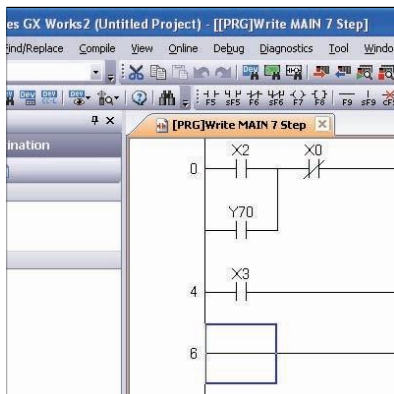
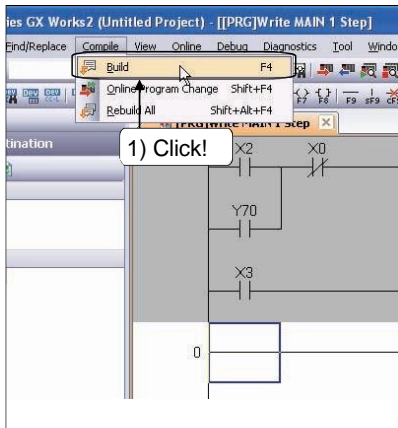
17) Click  trên thanh công cụ và nhập "Y71".

18) Click nút  .

19) Biểu tượng vừa nhập ( $\text{Y71}$ ) hiện ra.

20) Quy trình kết thúc.

## 2.6 Chuyển đổi chương trình (Chuyển đổi ladder)



1) Click [Compile] → [Build] ( **F4** ).

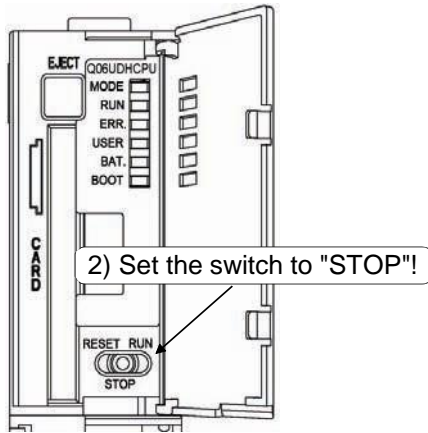
2) Chương trình ladder đã bị chuyển đổi.

Nếu lỗi xuất hiện trong suốt quá trình chuyển đổi, con trỏ sẽ tự di chuyển đến điểm lỗi của chương trình ladder. Kiểm tra những điểm đó và sửa lại chương trình nếu cần thiết.

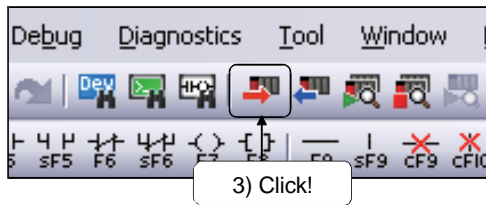



## 2.7 Đọc/Viết dữ liệu đến/từ khối CPU điều khiển khả trình

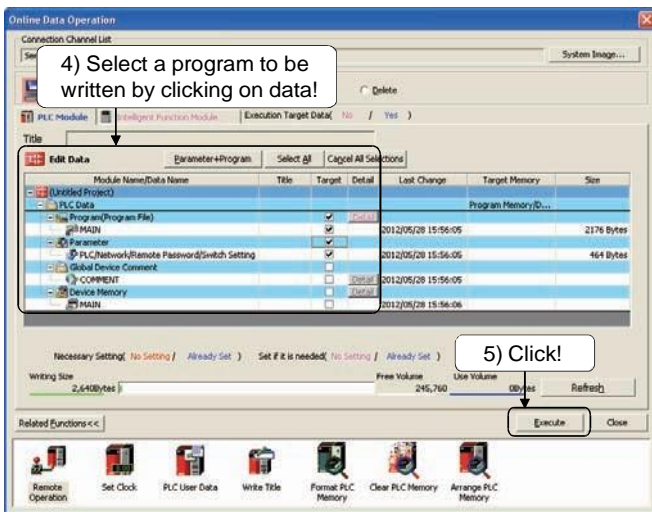
### (1) Viết dữ liệu đến CPU



- 1) Coi như chương trình ladder (chương trình tuần tự) đã được tạo ra với GX Works2 để thực hiện bước tiếp theo.
- 2) Đặt công tắc RUN/STOP/RESET trên CPU để STOP.

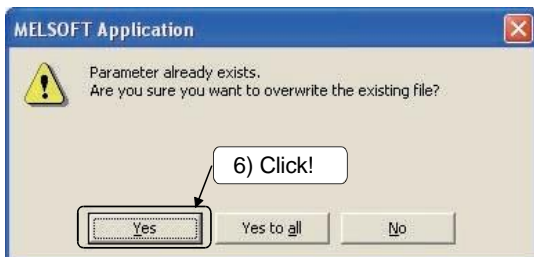


- 3) Click  trên thanh công cụ hoặc vào [Online] → [Write to PLC].



- 4) Từ thẻ "PLC Module, click để chọn chương trình và tham số để viết lên CPU hoặc vào **Parameter + Program** để chọn chương trình và tham số mục tiêu.

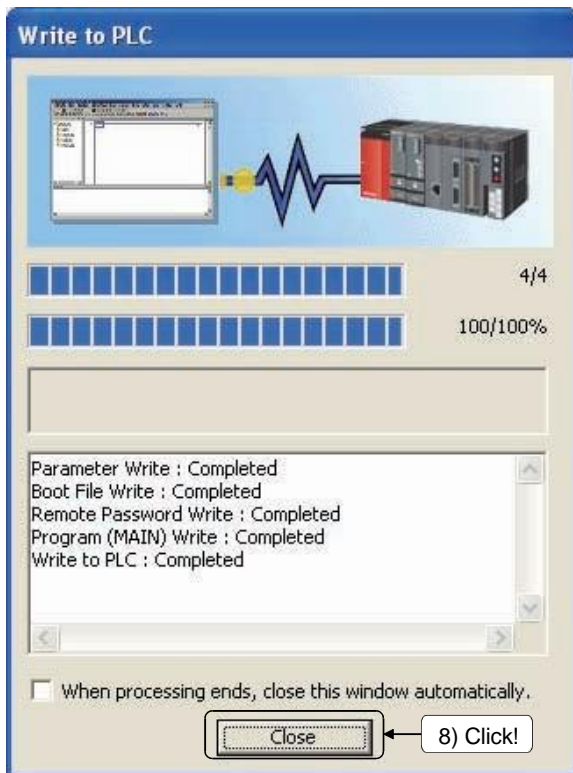
- 5) Click **Execute** để chấp nhận đã chọn .



- 6) Nếu một tham số hoặc chương trình đã được viết, hộp thoại xác nhận sẽ hiện ra để hỏi xem có xóa dữ liệu cũ không. Click **Yes** .

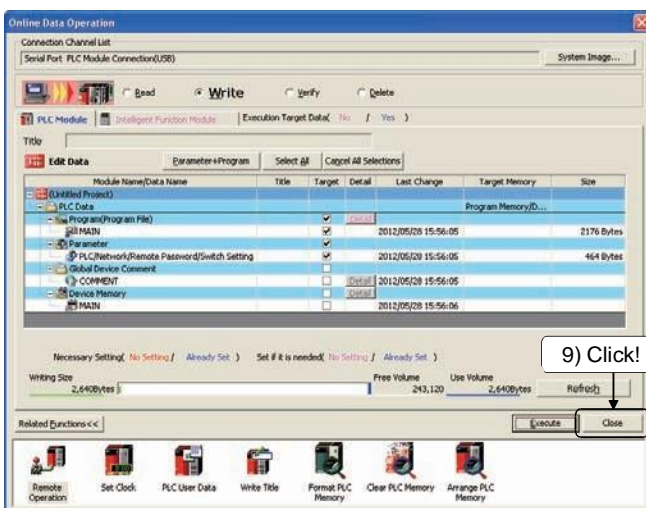
(Đến trang sau)

(Từ trang trước)



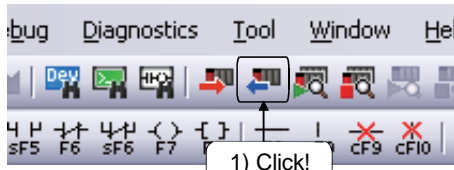
7) Hộp thoại thể hiện tiến trình hiện ra.


8) Thông báo "Completed" hiện ra khi quá trình viết hoàn tất. Sau đó click **Close**.

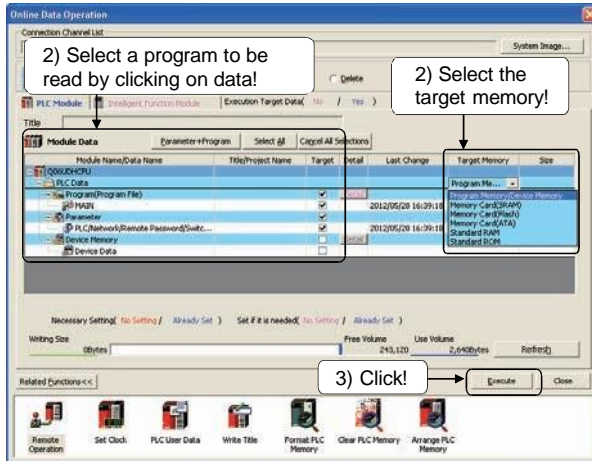


9) Click nút **Close** để đóng hộp thoại.

## (2) Đọc dữ liệu từ CPU



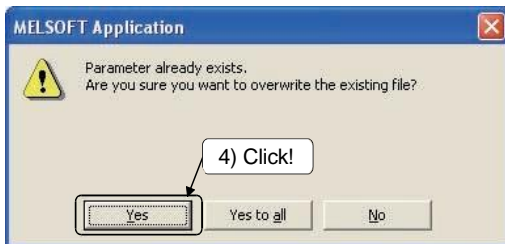
- 1) Click  trên thanh công cụ hoặc click [Online] → [Read from PLC].



- 2) Từ thẻ "PLC Module", click để chọn chương trình và tham số để đọc từ CPU. Hoặc click **Parameter + Program** để chọn tham số và chương trình mục tiêu.

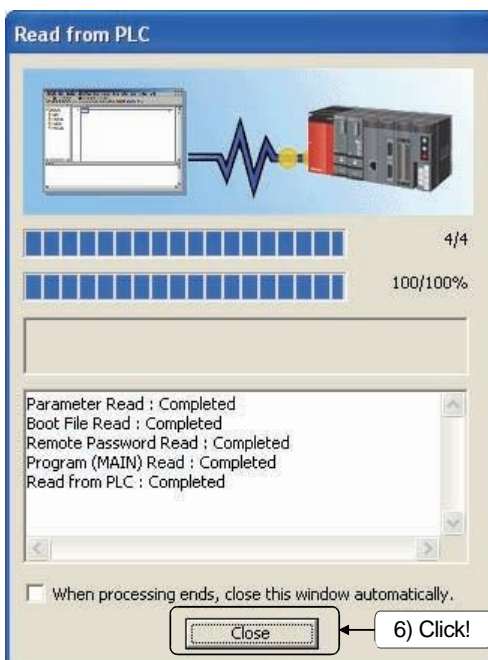
Chọn "Program Memory/Device Memory" cho "Target Memory".

- 3) Click **Execute** để chấp nhận đã chọn.



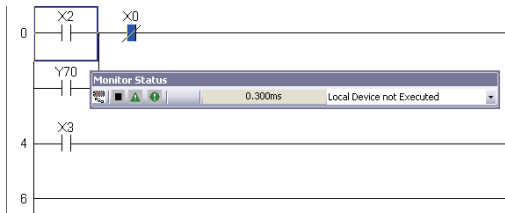
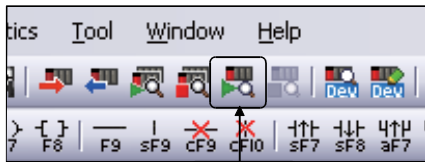
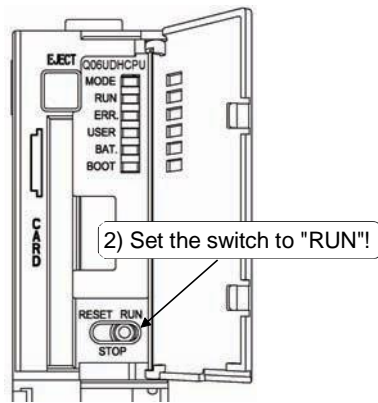
- 4) Nếu một chương trình hoặc tham số tồn tại, một hộp thoại xác nhận xóa dữ liệu cũ sẽ hiện lên. Click **Yes**.

- 5) Hộp thoại thể hiện tiến trình hiện ra.




- 6) Thông báo "Completed" hiện lên khi quá trình đọc hoàn tất. Sau đó, click **Close**.

## 2.8 Quan sát trạng thái chương trình ladder



1) Coi như chương trình ladder (chương trình tuần tự) đã được viết vào CPU điều khiển khả trình để tiếp tục bước sau.

2) Đặt công tắc RUN/STOP/RESET trên CPU để RESET (trong khoảng 1s.), đặt nó trở lại STOP, và đặt nó về RUN.

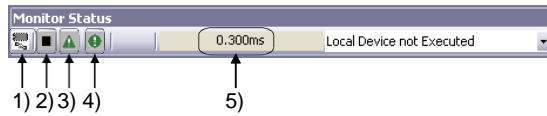
3) Click  trên thanh công cụ hoặc click [Online] → [Monitor] → [Start Monitoring].

4) Chọn menu khác kết thúc menu giám sát.

### Luyện tập

- 1) Xác nhận đồng hồ LED Y70 sáng lên bằng cách đặt công tắc chuyển mạch X2, và đồng hồ đó vẫn sáng sau khi công tắc chuyển mạch tắt đi.
- 2) Xác nhận đồng hồ LED Y70 tắt đi bằng cách ấn (đang bật) nút (công tắc chuyển mạch) X0, và đồng hồ không sáng lên khi nút (công tắc chuyển mạch) được giải phóng (tắt đi).
- 3) Bật công tắc chuyển mạch X3 làm bật đồng hồ LED Y71.

- (1) Trong mode giám sát, hộp thoại Monitor Status bên dưới hiện ra bất chấp trạng thái giám sát

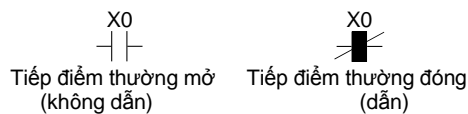


- 1) Trạng thái kết nối  
Thể hiện trạng thái kết nối giữa một CPU điều khiển khả trình và một máy tính cá nhân sử dụng chức năng mô phỏng được bắt đầu.
- 2) Trạng thái RUN/STOP  
Thể hiện trạng thái CPU điều khiển khả trình được điều khiển bởi công tắc khóa trên CPU điều khiển khả trình hoặc sự điều hành từ xa nhờ GX Works2.
- 3) Trạng thái ERR. (chuẩn đoán PLC)  
Thể hiện trạng thái lỗi của CPU điều khiển khả trình  
Click vào biểu tượng màn hình của màn hình chuẩn đoán PLC (\*1).
- 4) Trạng thái USER (chuẩn đoán PLC)  
Thể hiện trạng thái lỗi người dùng của CPU điều khiển khả trình.  
Click vào biểu tượng màn hình của màn hình chuẩn đoán PLC (\*1)
- 5) Chu kỳ quét  
Thể hiện chu kỳ quét tối đa của CPU điều khiển khả trình được giám sát.  
Bộ điều khiển khả trình Q-series thể hiện chu kỳ quét theo đơn vị 0.1 ms.

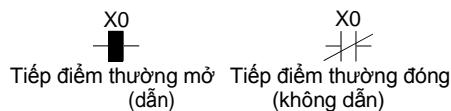
\*1: Đối với chuẩn đoán PLC, xem phần 2.8.

- (2) Trạng thái của ladder được thể hiện như bên dưới.

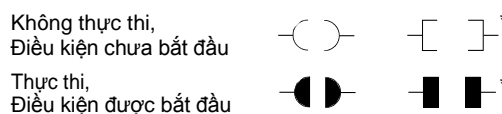
- 1) Hiện thị của một công tắc khi X0 = OFF



Hiện thị của một công tắc khi X0 = ON

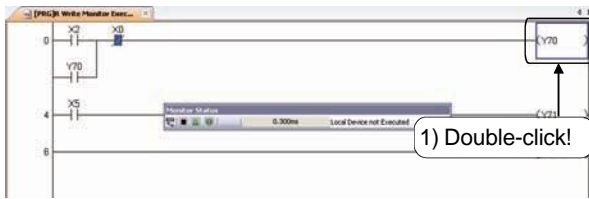


- 2) Hiện thị của một lệnh cuộn dây đầu ra, lệnh so sánh công tắc tương đương, và lệnh cuộn dây tương đương.

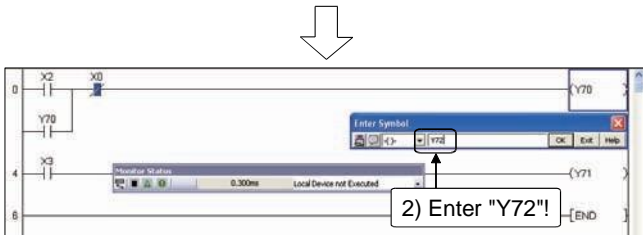


\*: Lệnh để so sánh công tắc tương đương và cuộn dây tương đương là SET, RST, PLS, PLF, SFT, SFTP, MC, FF, DELTA, và DELTAP.

(3) Sửa đổi ladder trong quá trình giám sát  
Phần này giải thích quy trình để sửa đổi Y70 thành Y72 trong quá trình giám sát.



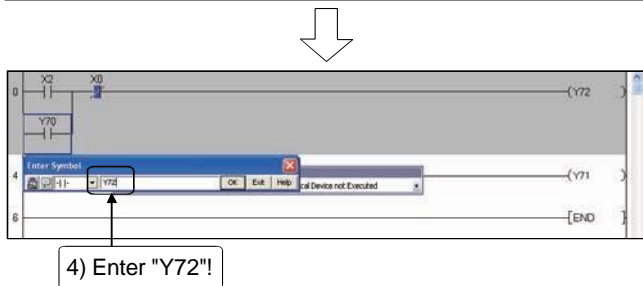
1) Click đúp (  $\text{---}(\text{Y70})\text{---}$  ).



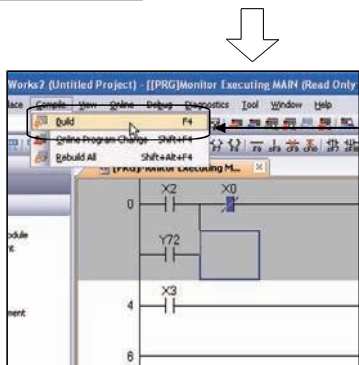
2) cửa sổ Enter Symbol hiện ra.

Nhập "Y72".

3) Nhấn nút  .

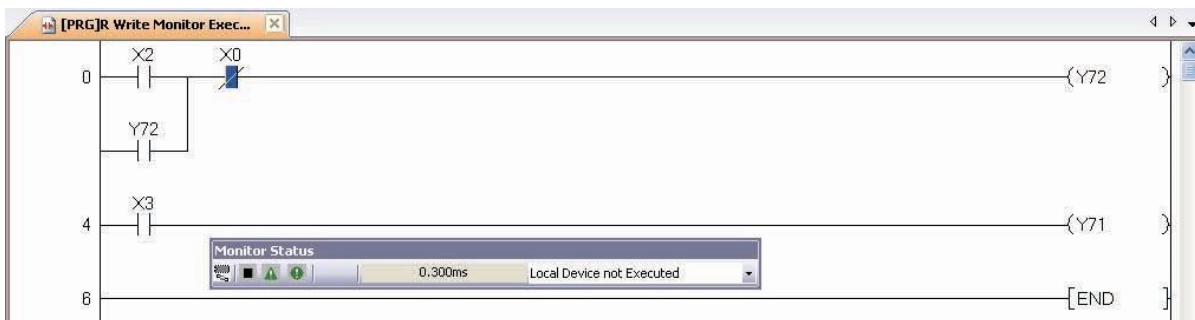


4) Click đúp (  $\text{---}(\text{Y70})\text{---}$  ) và chuyển "Y70" thành "Y72".



5) Click [Compile] → [Build] (  $\text{F4}$  ).

6) Quá trình sửa đổi hoàn tất



## 2.9 Chuẩn đoán lỗi CPU điều khiển khả trình



1) Click [Diagnostics] → [PLC Diagnostics].



2) Màn hình PLC Diagnostics hiện ra.

1) → Monitor Status Monitoring

2) → Connection Channel List

3) → Model Name, Operation Status, Switch

4) → The function menu is extended from the PLC image.

5) → Error Information

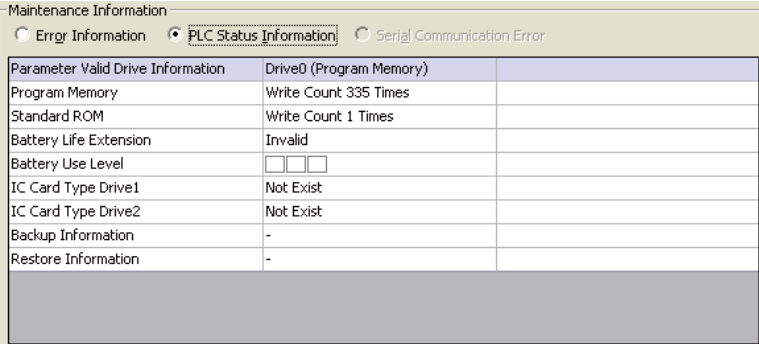
6) → Error History( PLC No.1 )

7) → Status Icon Legend

PLC	Status	No.	Current Error(Abbreviation)	Current Error(Detail)	Year/Month/Day	Time
1		0	No Error			
2						
3						
4						

Status	No.	Error Message(Abbreviation)	Error Message(Detail)	Year/Month/Day	Time
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-05-28	20:12:47
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-05-28	14:19:25
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-03-26	14:55:44
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-03-26	12:00:41
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-03-19	16:05:44
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-03-13	11:18:35
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-03-12	17:06:35
▲	3300	SP. PARA ERROR	SP. PARA ERROR	2012-03-12	17:03:59
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-03-09	14:34:10
▲	1500	AC/DC DOWN	AC/DC DOWN	2012-03-09	12:02:13
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-09	10:22:28
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-09	10:21:56
▲	7031	CPU LAY ERROR	CPU LAY ERROR	2012-03-05	13:49:00



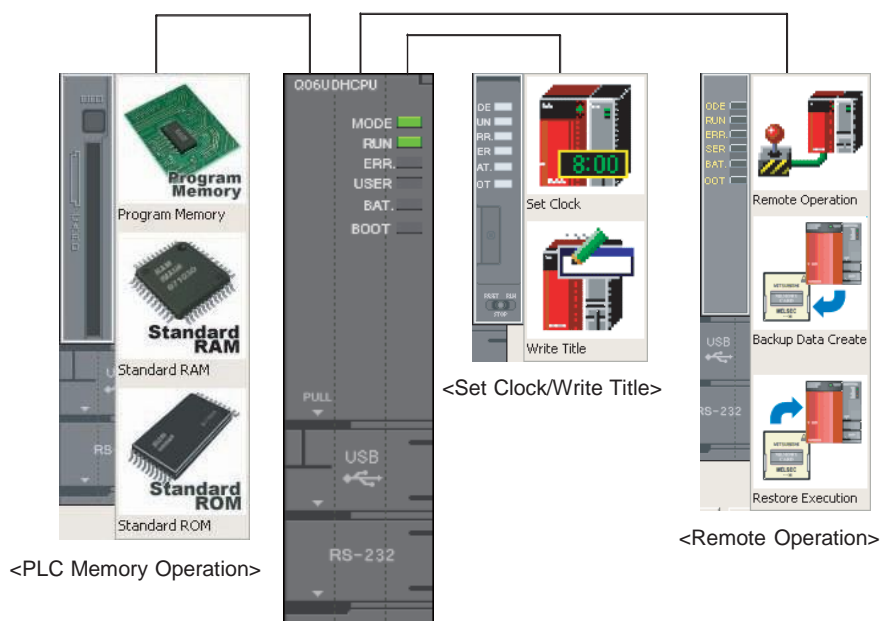
	Tên	Mô tả
1)	Trạng thái giám sát	Thể hiện trạng thái giám sát hiện tại
2)	Danh sách kênh kết nối	Thể hiện đường kết nối đã được đặt
3)	Trạng thái hoạt động CPU	<ul style="list-style-type: none"> <li>Với một hệ thống CPU đơn lẻ Thể hiện trạng thái hoạt động và trạng thái công tắc của CPU điều khiển khả trình.</li> <li>Với nhiều hệ thống CPU Thể hiện trạng thái hoạt động và trạng thái công tắc của CPU số 1 đến số 4.</li> <li>"Uninstallable/Blank" thể hiện một khe với không module nào lắp vào.</li> </ul>
4)	Hình ảnh CPU điều khiển khả trình	Thực hiện hoạt động trực tuyến của CPU điều khiển khả trình. (xem lại Gợi ý).
5)	Thông tin lỗi	Chọn cái này để thể hiện thông tin lỗi hiện tại của CPU điều khiển khả trình.
	Thông tin trạng thái PLC	Chọn cái này để thể hiện thông tin trạng thái của CPU điều khiển khả trình. 
6)	Lịch sử lỗi	Thể hiện lịch sử lỗi gần nhất bằng cách click vào nút <b>Error History</b> .
7)	Biểu tượng chú giải	Thể hiện biểu tượng trạng thái trên màn hình.

### Điểm chú ý

Thao tác trực tuyến

Chức năng PLC Memory Operation và chức năng Remote Operation có thể thực thi từ ảnh của CPU điều khiển khả trình.

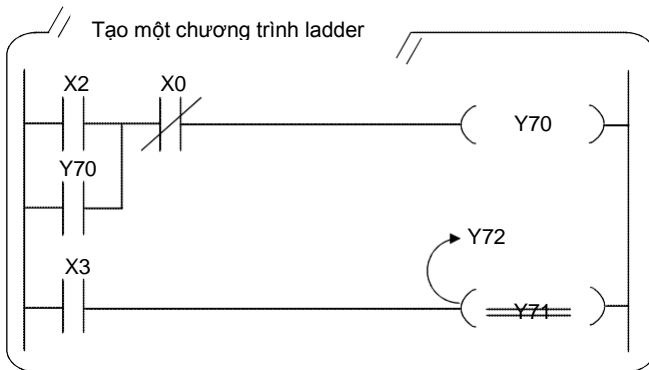
Khi con trỏ di chuyển đến ảnh của CPU điều khiển khả trình, menu chức năng được mở rộng. Click vào ảnh của CPU điều khiển khả trình để thể hiện những thành phần có thể đặt.



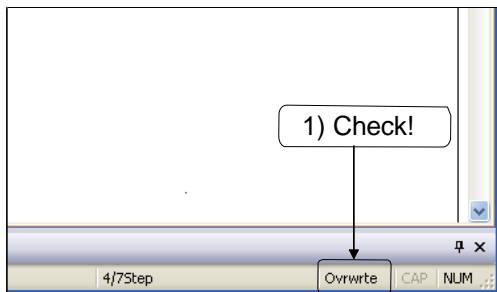


## 2.10 Chỉnh sửa chương trình ladder

### 2.10.1 Sửa đổi một phần của chương trình ladder



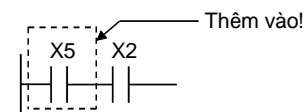
Phần này diễn giải làm thế nào để điều chỉnh một phần của chương trình ladder như hình bên.  
(OUT\_Y71 → OUT\_Y72)



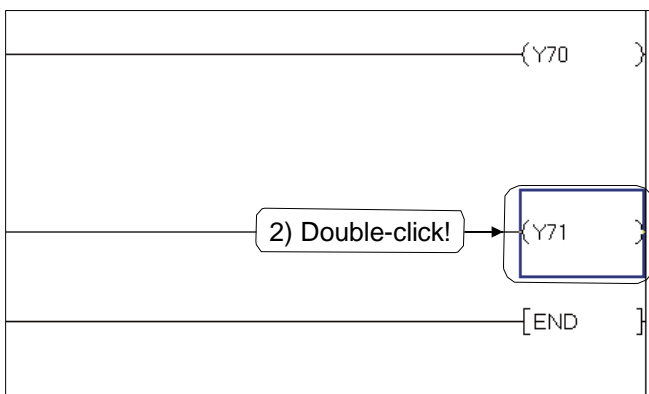
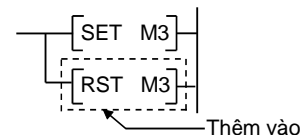
1) Xác nhận "Overwrite" được hiển thị ở phần bên phải bên dưới màn hình.

Nếu "Insert" hiển thị trên màn hình, click vào **Ins** để thay đổi hiển thị về "Overwrite".  
Nếu "Insert" hiển thị trên màn hình, công tắc hoặc cuộn dây được thêm vào biểu đồ.

<Khi sửa X2 thành X5>



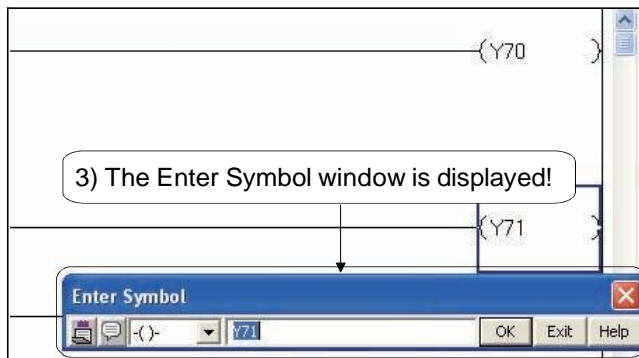
<Khi sửa SET thành RST>



2) Click đúp vào điểm để sửa lại.

(Đến trang sau)

(Từ trang trước)



3) Cửa sổ Enter Symbol hiện ra.

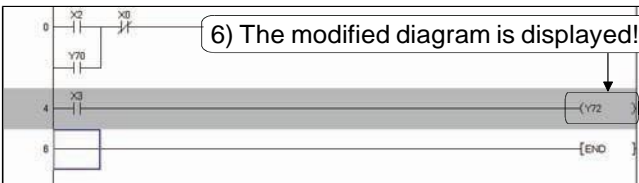


4) Click vào hộp chỉnh sửa và nhập "Y72".

4) Enter "Y72!"



5) Click!



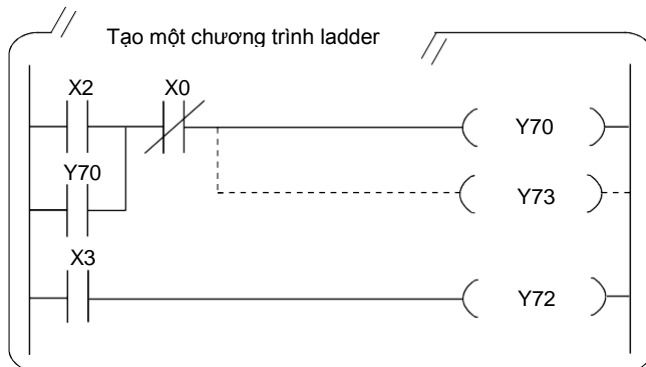
5) Click nút  để xác nhận thay đổi.

6) Chương trình ladder được điều chỉnh hiện ra.

7) Để chuyển đổi chương trình ladder, click [Compile] → [Build] (  ).


## 2.10.2 Vẽ/ xóa đường thẳng

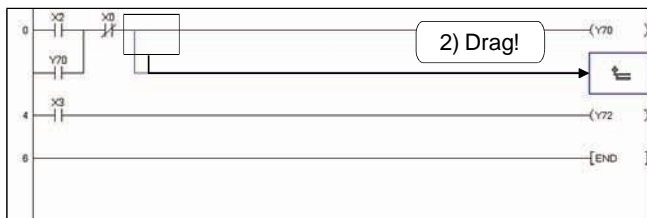
### (1) Vẽ đường thẳng



Phần này diễn giải làm thế nào để thêm một đường thẳng vào chương trình ladder như hình bên.

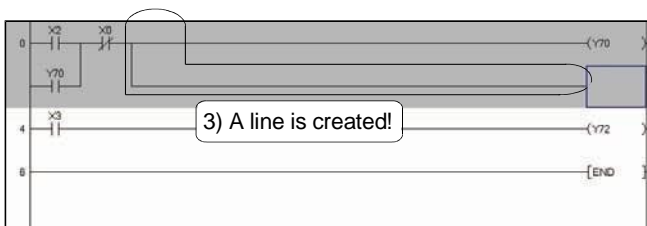


1) Click  ( Alt + F10 ) trên thanh công cụ.



2) Kéo con chuột từ điểm bắt đầu đến điểm kết thúc.

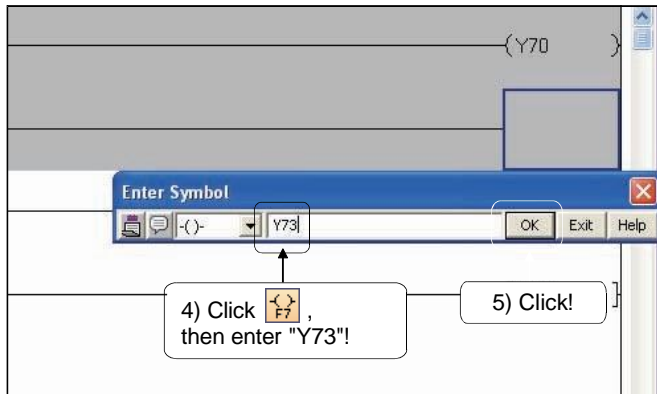
Một đường thẳng được tạo ra ở bên trái chuột.




3) Một đường thẳng được tạo ra khi trái chuột được thả ra.

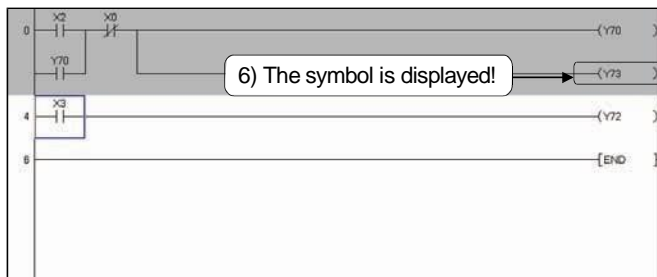
(Đến trang sau)

(Từ trang trước)



4) Click  trên thanh công cụ và nhập "Y73".

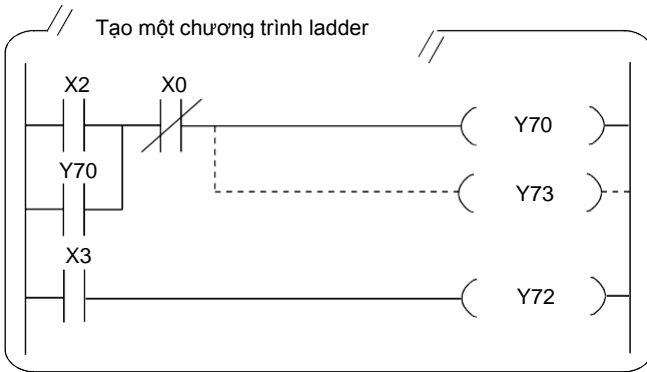
5) Click nút  .



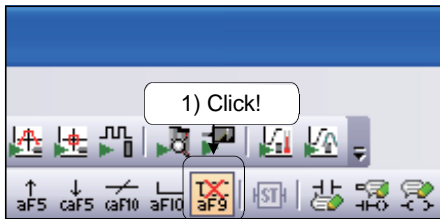
6) Biểu tượng vừa nhập ( Y73 ) hiện ra.

7) Để chuyển đổi chương trình ladder, click [Compile] → [Build] (  ).

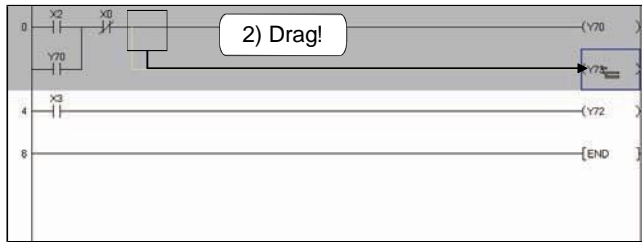
## (2) Xóa đường thẳng



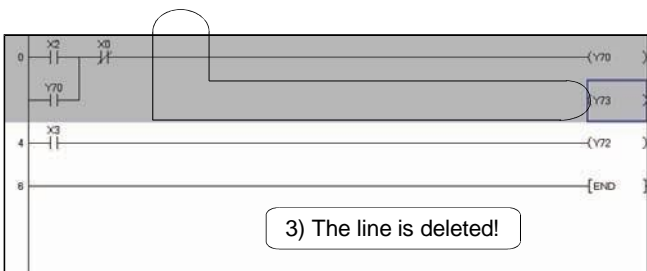
Thực hiện những bước dưới đây để xóa một đường thẳng như hình bên.



1) Click  ( **Alt** + **F9** ) trên thanh công cụ.

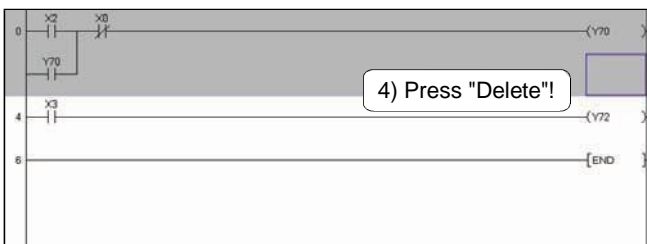


2) Kéo chuột từ điểm bắt đầu đến điểm kết thúc.



3) Đường thẳng bị xóa khi trái chuột thả ra.

Đường thẳng được vẽ cho lệnh END không thể xóa.

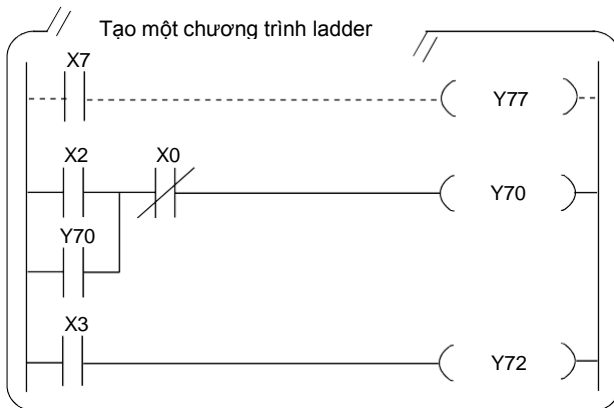


4) Nhấn nút **Delete** để xóa  $(Y73)$ .

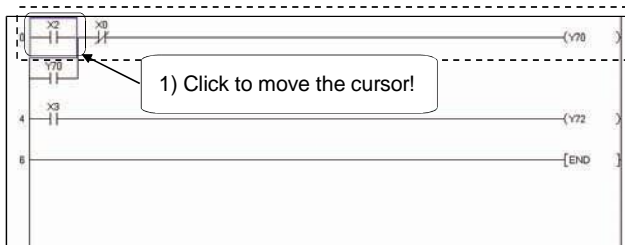
5) Để chuyển đổi chương trình ladder, click **[Compile]** → **[Build]** ( **F4** ).

## 2.10.3 Chèn/xóa một hàng

### (1) Chèn hàng

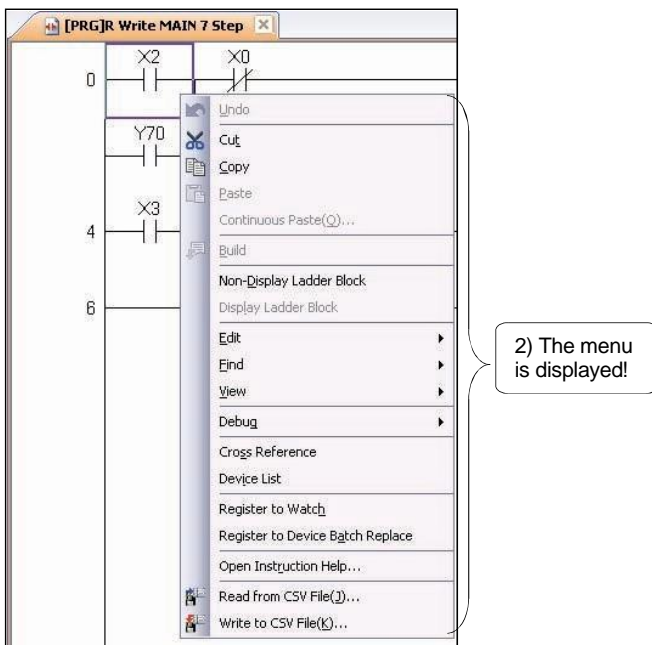


Phần này diễn giải làm thế nào để thêm một jangf vào chương trình ladder ở hình bên trái.



1) Click vào điểm bất kì của hàng để di chuyển con trỏ đến.

Một hàng được thêm vào trên hàng được chọn với con trỏ.

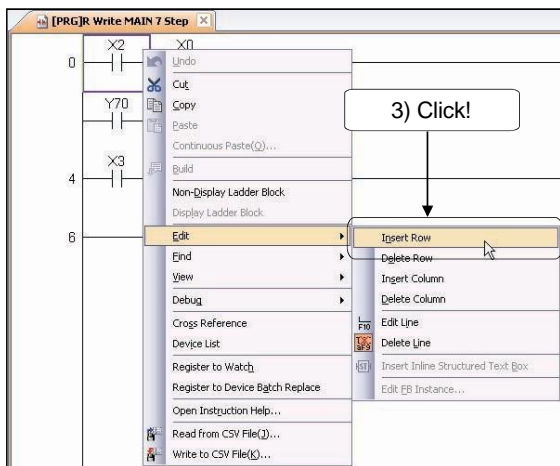


2) Phải chuột vào điểm bất kì trên màn hình tạo chương trình ladder để hiển thị menu.

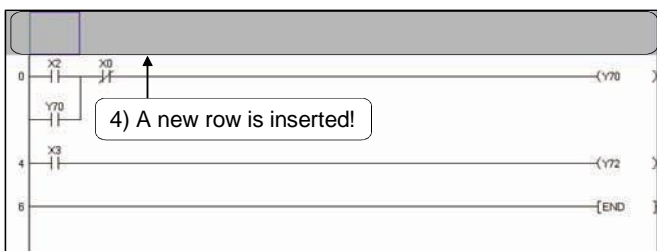


(Đến trang sau)

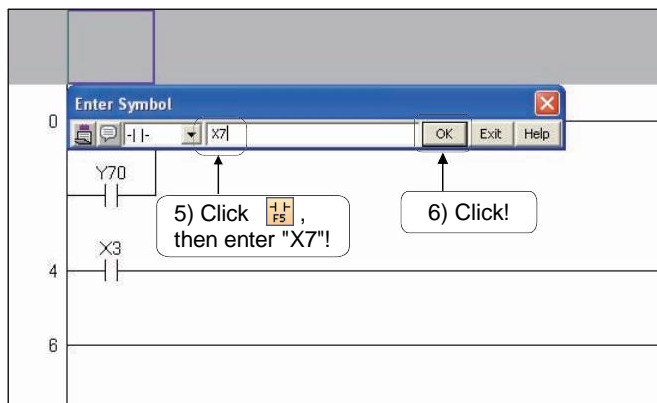
(Từ trang trước)



3) Chọn [Edit] → [Insert Row]  
( **Shift** + **Ins** ).



4) Một hàng mới được chèn vào trên hàng được chọn.



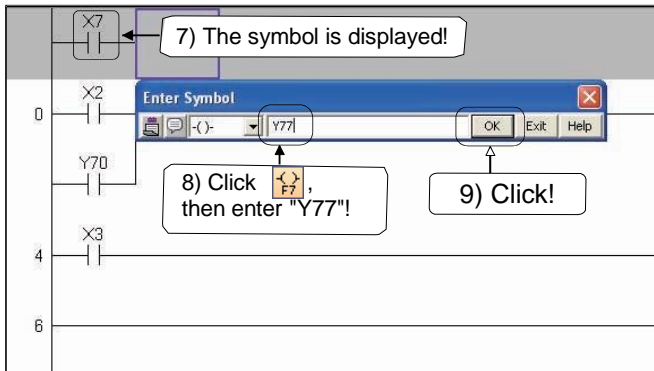
5) Click **FS** trên thanh công cụ để mở cửa sổ Enter Symbol. Nhập "X7".

6) Click nút **OK** để chấp nhận.




(Đến trang sau)

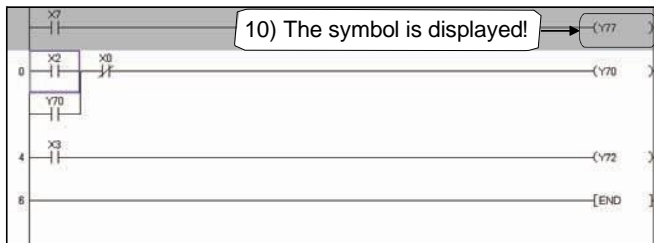
(Từ trang trước)



7) Biểu tượng nhập vào ( $\overline{X7}$ ) hiện ra.

8) Click  trên thanh công cụ, và nhập "Y77".

9) Click nút  .

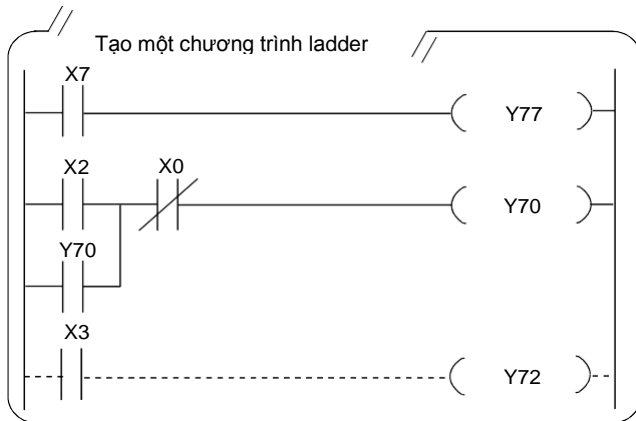


10) Biểu tượng vừa nhập ( $\overline{Y77}$ ) hiện ra.

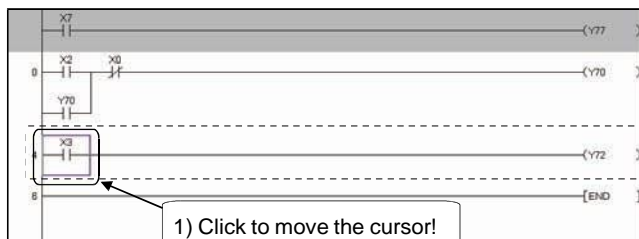
11) Để chuyển đổi chương trình ladder đã sửa, click  
[Compile] → [Build] (  ).



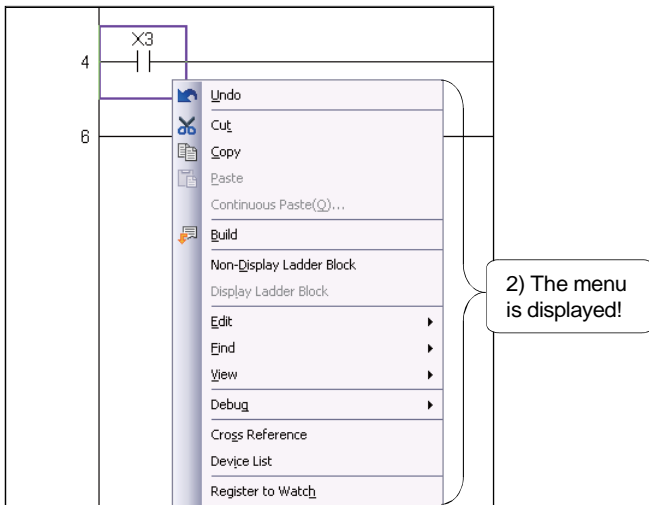
## (2) Xóa hàng



Phần này diễn giải làm thế nào để xóa hàng từ chương trình ladder như hình bên.



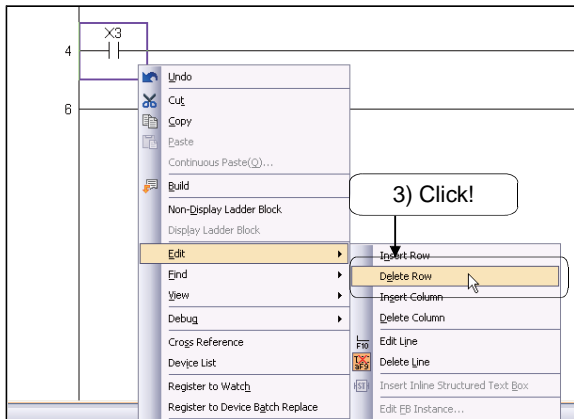
1) Đưa con trỏ đến hàng muốn xóa.



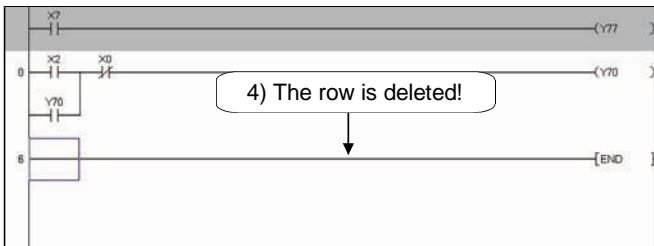
2) Kích chuột phải đến bất kì điểm nào trên màn hình tạo chương trình ladder để hiển thị menu.

(Đến trang sau)

(Từ trang trước)



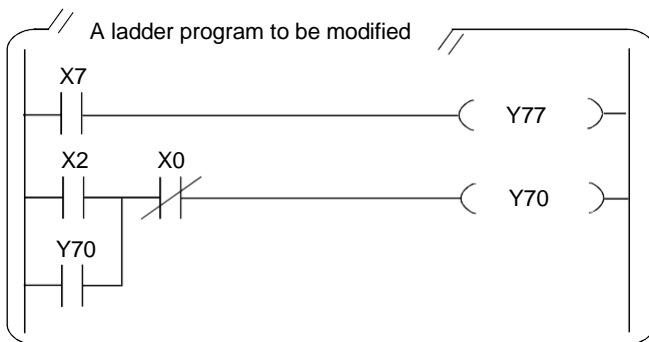
3) Chọn [Edit] → [Delete Row]  
( **Shift** + **Del** ).



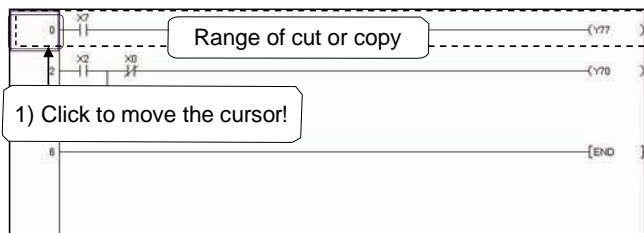
4) Hàng được chọn đã bị xóa.

5) Để chuyển đổi chương trình ladder, click  
[Compile] → [Build] ( **F4** ).

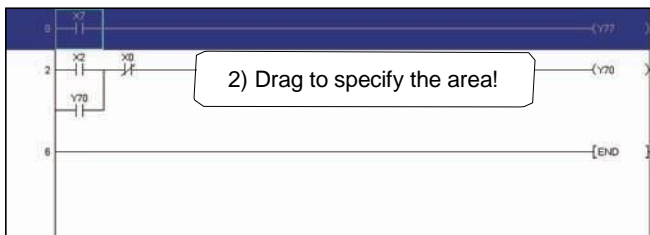
## 2.10.4 Cắt/sao chép chương trình ladder



Phần này diễn giải làm thế nào để sao chép và cắt chương trình ladder như hình bên.






- 1) Click vào điểm bắt đầu chương trình ladder muốn cắt để đặt con trỏ.



- 2) Kéo chuột bao phủ ladder để nhận diện vùng ladder muốn cắt.  
Vùng được chọn đã được chuyển màu.

Click vào số bước và kéo chuột sang ngang để nhận diện vùng trong đơn vị khối ladder.

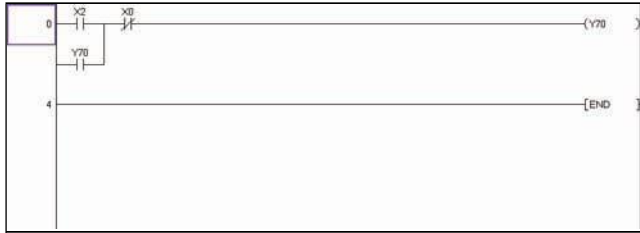


- 3) Click  trên thanh công cụ hoặc vào [Edit] → [Cut] (  +  ) để cắt vùng mong muốn

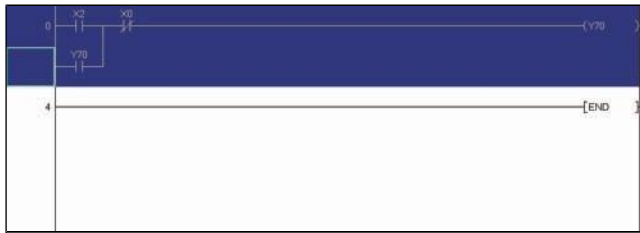


(Đến trang sau)

(Từ trang trước)

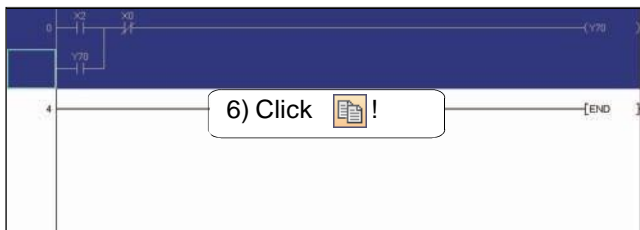



- 4) Click vào điểm bắt đầu của chương trình ladder muốn sao chép để đưa con trỏ tới.

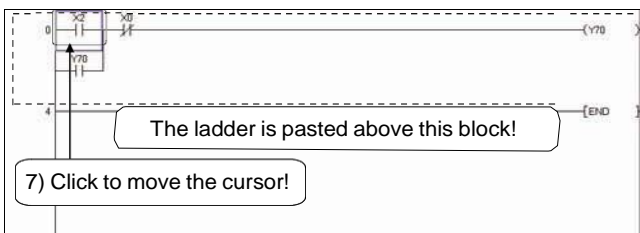


- 5) Kéo con trỏ bao trùm ladder để nhận diện vùng muốn cắt.  
Vùng được chọn được chuyển màu.

Click vào số bước và kéo chuột sang ngang để nhận diện vùng trong đơn vị khối ladder.



- 6) Click  trên thanh công cụ hoặc vào [Edit] → [Copy] ( **Ctrl** + **C** ) để sao chép vùng mong muốn.




- 7) Click vào bất kì khối ladder nào để đặt con trỏ đến ladder. Ladder muốn sao chép được dán vào trên hàng đặt con trỏ.

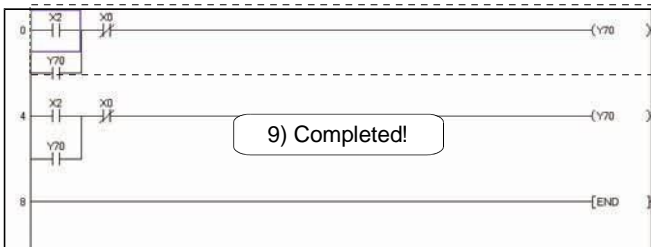


(Đến trang sau)

Từ trang trước)



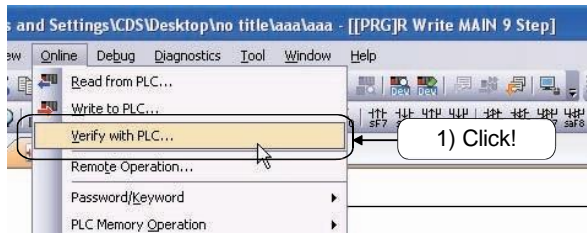
- 8) Click  trên thanh công cụ hoặc [Edit] → [Paste] ( **Ctrl** + **V** ) để dán vùng mong muốn.



- 9) Ladder bị cắt hoặc sao chép được dán vào.

## 2.11 Kiểm tra dữ liệu

Phần này diễn giải làm thế nào để kiểm tra dự án đang mở với dữ liệu trên CPU điều khiển khả trình.

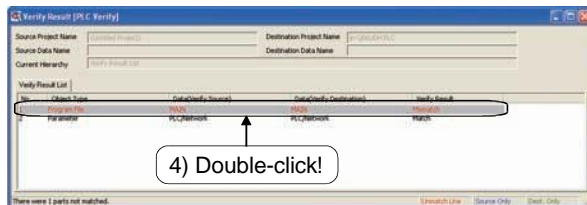


1) Click [Online] → [Verify with PLC].



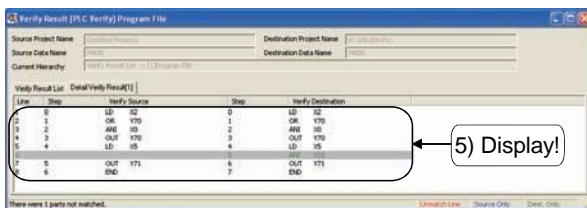
2) Hộp thoại Online Data Operation hiện ra. Click nút **Parameter + Program**.

3) Click nút **Execute**.



4) Kết quả kiểm tra hiển thị trên cửa sổ Verify Result.

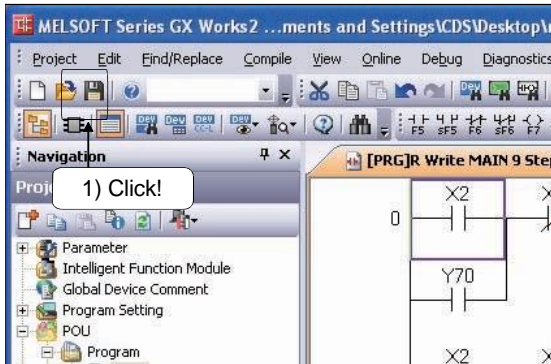
Để kiểm tra chi tiết dữ liệu, click đúp vào hàng tương ứng.




5) Chi tiết của kết quả được hiển thị.

## 2.12 Lưu chương trình ladder

### 2.12.1 Lưu dự án mới tạo hoặc ghi đè dự án

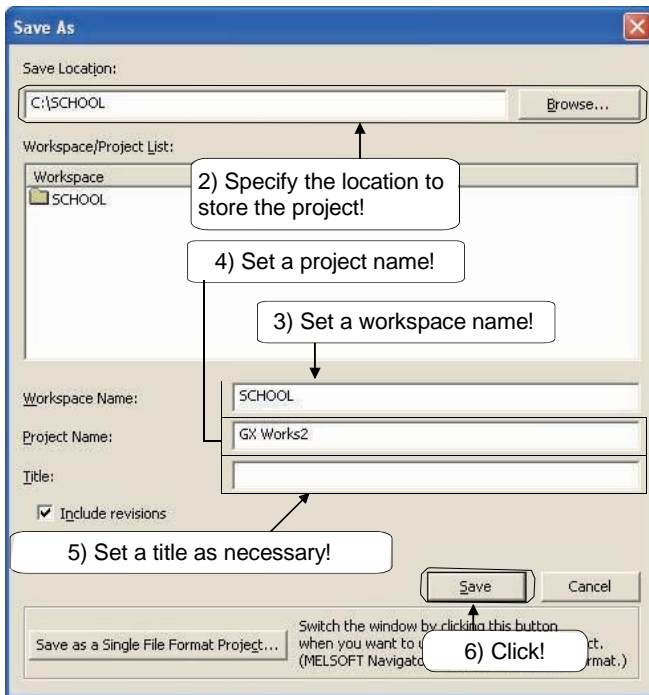


- 1) Click  trên thanh công cụ hoặc vào [Project] → [Save] (  +  ).

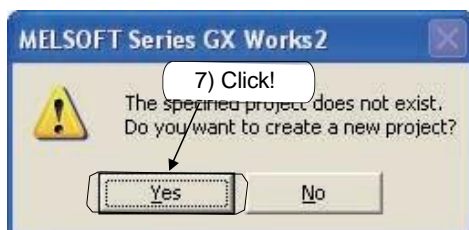
Lưu dự án đã có hoàn thành xong ở bước này



(Chỉ khi một dự án mới tạo được lưu)



- 2) Chỉ định địa điểm để lưu dự án .
- 3) Đặt tên vùng làm việc.
- 4) Đặt tên dự án.
- 5) Đặt một tiêu đề nếu cần thiết.
- 6) Click nút  để xác nhận.

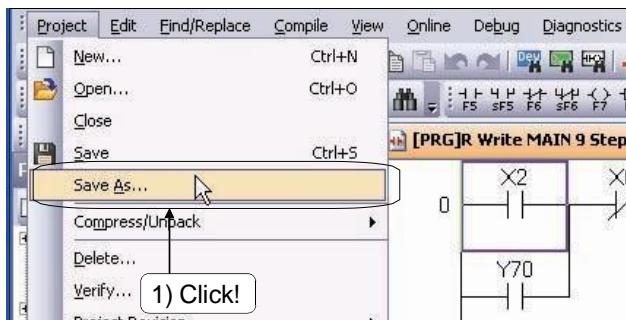


- 7) Click nút  .  
Dự án mới đã được lưu.

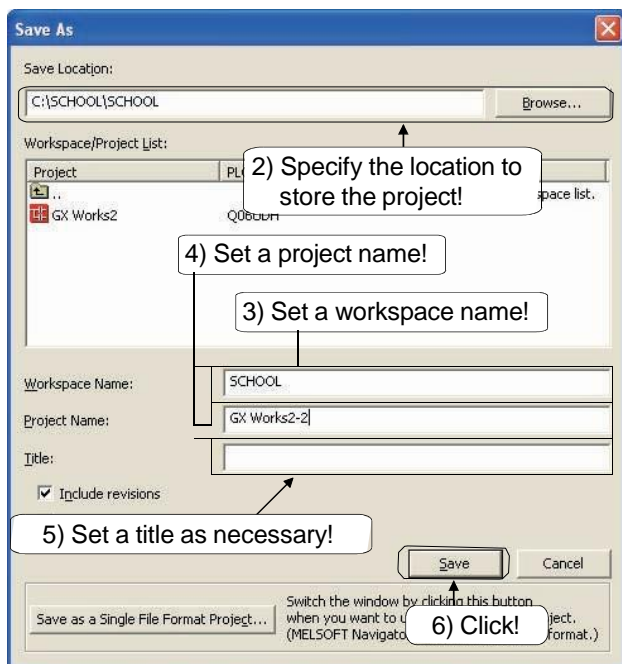
## Gợi ý

- Vùng làm việc  
Vùng làm việc cho phép GX Works2 quản lý nhiều dự án chỉ với một tên.
- Khi địa điểm lưu tồn tại  
Khi địa điểm lưu (vùng làm việc và dự án) tồn tại, thư mục chứa vùng làm việc được xác định trong "Workspace/Project List".
- Số kí tự cho một tên vùng làm việc, dự án, và tiêu đề  
Xác định tên vùng làm việc, tên dự án, và tiêu đề trong khoảng 128 kí tự  
Tuy nhiên, tổng số kí tự của tên đường dẫn địa điểm lưu + tên vùng làm việc + tên dự án phải nằm trong 150.

### 2.12.2 Lưu một dự án với một tên khác



1) Click [Project] → [Save as].



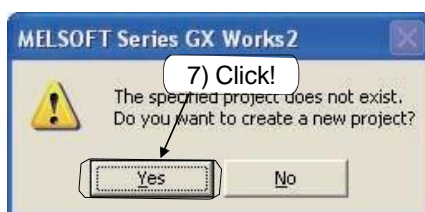
2) Chỉ định địa điểm để lưu dự án .

3) Đặt tên vùng làm việc.

4) Đặt tên dự án.

5) Đặt một tiêu đề nếu cần thiết.

6) Click nút **Save** để xác nhận.

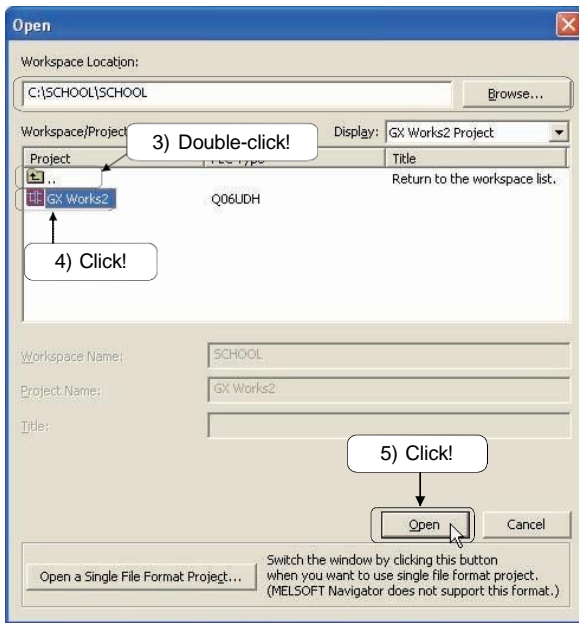



7) Click nút **Yes** .

Dự án mới đã được lưu.



## 2.13 Đọc một dự án đã lưu



1) Click  trên thanh công cụ hoặc vào [Project] → [Open] (  +  ).

2) Xác định địa điểm nơi dự án muốn đọc được lưu.

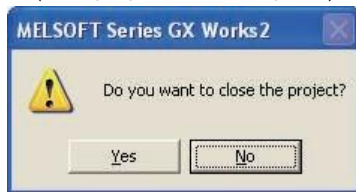
3) Click đúp vào vùng làm việc muốn đọc.

4) Click vào dự án muốn đọc.

5) Click vào nút đó để bắt đầu đọc dự án đã chọn.

Mỗi hộp thoại xác nhận được hiện ra trong những trường hợp dưới đây;

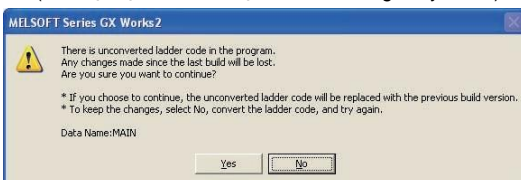
(Khi một dự án khác đã được mở)



.....Thoát khỏi dự án.

.....Giữ dự án mở.

(Khi một dự án khác được mở mà không chuyển đổi)



.....Thoát khỏi dự án mà không chuyển đổi dự án.

.....Giữ dự án mở.

(Tiếp tục chỉnh sửa chương trình ladder.)

(Khi dự án khác đang được mở mà chưa được lưu)



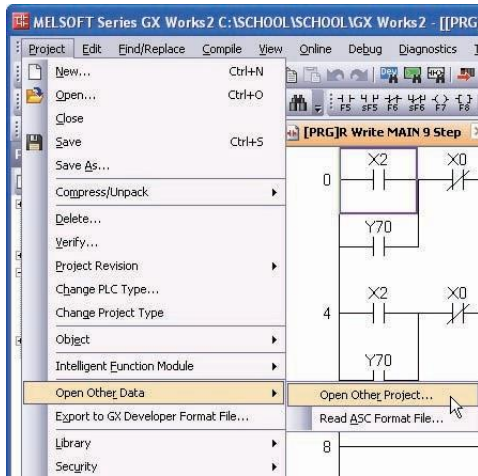
.....Thoát khỏi dự án sau khi lưu nó.

.....Thoát khỏi dự án mà không lưu nó.

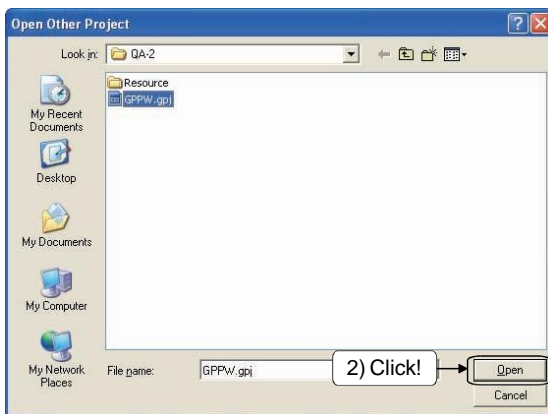
.....Giữ dự án mở.

## 2.14 Mở dự án theo định dạng khác

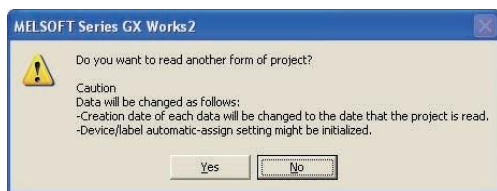
Phần này diễn giải làm thế nào để mở một dự án được tạo với GX Developer trong GX Works2.



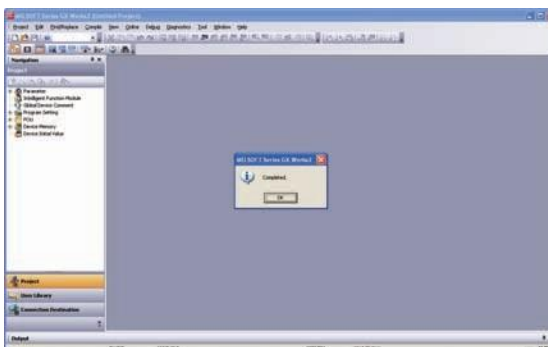
- 1) Click [Project] → [Open Other Data] → [Open Other Project].



- 2) Hộp thoại Open Other Project hiện ra. Xác định dự án muốn mở và click **Open**.



- 3) Thông báo ở bên trái hiện ra. Click nút **Yes**.



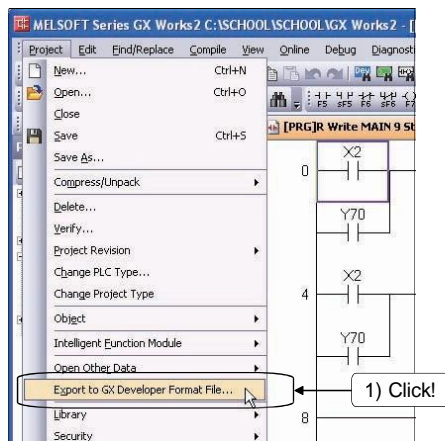
- 4) Dự án được tạo với GX Developer được mở.

## Gợi ý

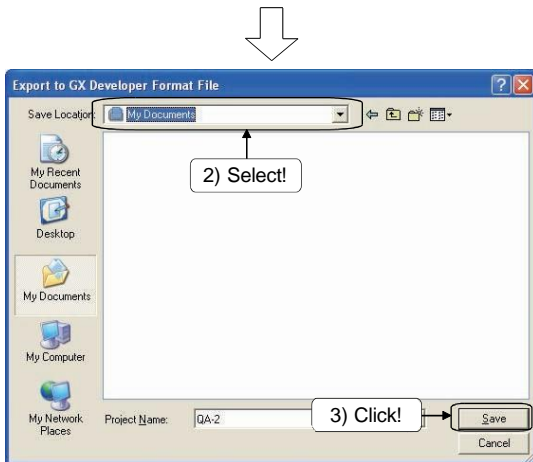
- Trạng thái sau khi một dự án với định dạng khác được mở  
Khi một dự án trong một định dạng khác được mở, dự án đang ở trạng thái chưa được biên dịch .  
Biên dịch tất cả chương trình trong dự án trước khi thực thi điều hành trực tuyến như viết dữ liệu và giám sát.  
Khi một trạng thái biên dịch lỗi xuất hiện, sửa lại chương trình tương ứng theo như hướng dẫn lập trình.

## 2.15 Lưu dự án với định dạng khác

Phần này diễn giả làm thế nào để lưu một dự án đơn giản của GX Works2 theo định dạng GX Developer.



- 1) Click [Project] → [Export to GX Developer Format Tệp].

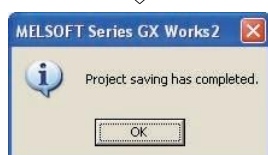


- 2) Hộp thoại Export to GX Developer Format Tệp. Xác định địa điểm để lưu dự án.

- 3) Nhập một tên dự án và click nút



- 4) Thông báo ở bên trái hiện lên.  
Click



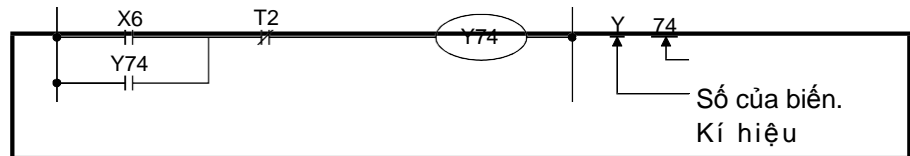
- 5) Dự án được lưu với định dạng GX Developer.

MEMO

## CHƯƠNG 3 BIẾN VÀ THAM SỐ BỘ ĐIỀU KHIỂN KHẢ TRÌNH

### 3.1 Biến

Biến là một phần tử ảo cho việc lập trình bộ điều khiển CPU, tương tự các phần tử (như tiếp điểm và cuộn cảm) mà nó viết nên một chương trình.



Các dạng		Mô tả	Ghi chú
X	Đầu vào	Gửi các lệnh và dữ liệu đến bộ điều khiển khả trình thông qua các thiết bị ngoài như nút bấm, chuyển mạch lựa chọn, chuyển mạch giới hạn và chuyển mạch số.	<ul style="list-style-type: none"> <li>• Biến bit</li> <li>• Chủ yếu là tín hiệu bật/tắt</li> </ul>
Y	Đầu ra	Các đầu ra đưa đến các cuộn dây, khởi động từ và các đèn chỉ thị và chỉ thị số.	
M	Rơ-le nội	Rơ-le phụ bên trong bộ điều khiển nó không thể đưa trực tiếp ra thiết bị ngoài.	
L	Rơ-le chốt	Rơ-le phụ hoạt động liên tục bên trong bộ điều khiển nó không thể đưa trực tiếp ra thiết bị ngoài.	
B	Rơ-le liên kết	Rơ-le trong cho liên kết dữ liệu nó không thể đưa trực tiếp ra thiết bị ngoài. Tại đây không được quy định bởi thông tin liên kết ban đầu mà được sử dụng như rơ-le trong.	
F	Bộ chỉ báo	Được sử dụng để phát hiện lỗi. Thiết lập một chương trình phát hiện lỗi trước và khởi động chương trình trong khi bộ điều khiển chương trình đang chạy để lưu trữ các giá trị số vào một thanh ghi riêng D.	
V	Rơ-le biên	Rơ-le trong lưu trữ các kết quả hoạt động	
SM	Rơ-le đặc biệt	Rơ-le trong mà lưu trữ trạng thái CPU.	
SB	Rơ-le liên kết đặc biệt	Rơ-le trong cho liên kết dữ liệu mà nó biểu thị một trạng thái giao tiếp hoặc lỗi.	
FX	Đầu vào chức năng	Rơ-le trong mà nó giữ dữ liệu bật tắt và được ghi bởi một lệnh gọi chương trình con trong một chương trình con.	
FY	Đầu ra chức năng	Rơ-le mà nó chuyển kết quả hoạt động từ chương trình con sang một nguồn gọi chương trình con.	
T(ST)	Bộ định thời	Bộ định thời bao gồm dạng: định thời tốc độ thấp, định thời tốc độ cao, bộ tích phân tốc độ thấp và bộ tích phân tốc độ cao.	<ul style="list-style-type: none"> <li>• Biến dạng từ</li> <li>• Chủ yếu là dữ liệu</li> <li>• Một từ gồm 16 bit.</li> <li>• Có thể được ghi bởi đầu vào ~.* (* = 0 đến F (ơ số 16)).</li> </ul>
C	Bộ đếm	Bộ đếm gồm hai dạng: bộ đếm cho các chương trình PLC và bộ đếm cho các chương trình PLC xử lý ngắt.	
D	Thanh ghi dữ liệu	Bộ nhớ lưu trữ dữ liệu bộ điều khiển khả trình.	
W	Thanh ghi liên kết	Thanh ghi dữ liệu cho liên kết dữ liệu.	
R	Thanh ghi tệp	Thanh ghi mở rộng cho thanh ghi dữ liệu, nó sử dụng RAM hoặc thẻ nhớ.	
SD	Thanh ghi đặc biệt	Thanh ghi lưu trữ trạng thái CPU.	
SW	Thanh ghi liên kết dữ liệu	Thanh ghi dữ liệu cho liên kết dữ liệu nó lưu trữ trạng thái giao tiếp và thông tin lỗi.	
FD	Thanh ghi chức năng	Thanh ghi trao đổi dữ liệu giữa nguồn gọi vòng lặp chương trình con này với chương trình con khác.	
Z	Thanh ghi chỉ số	Thanh ghi thay đổi biến (X, Y, M, L, B, F, T, C, D, W, R, K, H, và P)	

Các dạng		Mô tả	Ghi chú	
FD	Thanh ghi chức năng	Thanh ghi trao đổi dữ liệu giữa nguồn gọi vòng lặp chương trình con này với chương trình con khác.		
Z	Thanh ghi chỉ số	Thanh ghi thay đổi biến (X, Y, M, L, B, F, T, C, D, W, R, K, H, và P)		
N	Lồng	Hiển thị kiến trúc được lồng của bộ điều khiển chủ.		
P	Con trỏ	Xác định vị trí nhảy của các nhánh chỉ dẫn (CJ, SCJ, CALL và JMP).		
I	Con trỏ ngắt	Xác định một vị trí nhảy tương ứng với hệ số ngắt khi một hoạt động ngắt diễn ra.		
J	Mạng số. Đặc điểm thiết bị	Sử dụng để chỉ rõ số mạng trong lệnh liên kết dữ liệu.		
U	Số I/O .	Được sử dụng để chỉ ra số I/O trong module chức năng thông minh dành cho các lệnh		
K	Hằng số thập phân	Được sử dụng để chỉ ra các: giá trị thiết lập bộ đếm định thời, số con trỏ, số con trỏ ngắt, số chữ số của biến bit và các giá trị lệnh cơ bản.		
H	Hằng số thập lục phân	Được sử dụng để chỉ các giá trị lệnh cơ bản/ứng dụng.		
E	Hằng số thực	Được sử dụng để chỉ ra số thực của lệnh		
"Chuỗi kí tự"	Hằng số chuỗi kí tự	Được sử dụng để chỉ chuỗi kí tự như các lệnh		
Jn\X Jn\Y Jn\B Jn\SB	Biến liên kết trực tiếp	Biến có thể truy xuất trực tiếp đến một biến liên kết của một mô-đun mạng. (Làm mới các tham số thiết lập không được yêu cầu)		• Biến bit • Tín hiệu on/off .
Jn\W Jn\SW				• Biến dạng từ • Dữ liệu
Un\G	Mô-đun chức năng thông minh	Biến có thể truy xuất trực tiếp bộ nhớ đệm của một mô-đun chức năng thông minh.		• Một từ gồm 16 bit

### 3.2 Tham số

Tham số là các giá trị thiết lập cơ bản được gán vào bộ điều khiển khả trình để điều khiển các đối tượng.

Các tham số được phân chia thành các tham số PLC, tham số mạng, và mật khẩu từ xa như bên dưới.

\* Các vùng in đậm thể hiện các mục đã được sắp xếp trong cuốn sách này

Mục		Mô tả	
Tham số PLC	Tên PLC	Nhãn	Cài đặt (tên và ứng dụng) của bộ điều khiển khả trình CPU.
		Ghi chú	Cài đặt chú giải cho nhãn của bộ điều khiển CPU.
	Hệ thống PLC	Cài đặt giới hạn bộ định thời	Cài đặt giới hạn thời gian của bộ định thời tốc độ thấp và tốc độ cao.
		Tiếp điểm RUN-PAUSE	Cài đặt tiếp điểm cho điều khiển RUN và PAUSE của CPU
		Tiếp xúc hợp lệ thao tác sao lưu dữ liệu khóa.	Cài đặt các thiết bị tiếp điểm để thực hiện sao lưu dữ liệu khóa (Chỉ dành cho Universal QCPU)
		Thiết lập lại từ xa	Thiết lập để cho phép hoạt động cài lại từ xa nhờ GX Works2.
		Trạng thái đầu ra từ STOP đến	Cài đặt trạng thái đầu ra (Y) khi bộ điều khiển chuyển từ STOP sang RUN.
		Thuật toán xử lý số thực	Thiết lập để xử lý số thực (chỉ dành cho mô-đun hiệu suất cao QCPU)
		Thiết lập mô-đun chức năng thông minh	<ul style="list-style-type: none"> <li>Cài đặt tham số mô-đun</li> <li>Cài đặt số bắt đầu I/O</li> </ul>
		Số con trỏ chung	Cài đặt số bắt đầu của con trỏ được sử dụng như con trỏ chung
		Điểm chiếm giữ bởi khe trống	Cài đặt số khe trống cho đơn vị cơ bản chính hoặc mở rộng.
		Cài đặt ngắt hệ thống	<ul style="list-style-type: none"> <li>Cài đặt số bắt đầu của các bộ đếm ngắt</li> <li>Cài đặt khoảng thực hiện cho các con trỏ ngắt.</li> </ul>
		Cài đặt chương trình quét cố định/ Chương trình ngắt	Cài đặt thực hiện phá hủy tốc độ cao của một chương trình ngắt.
		Đồng bộ mô-đun	Thiết lập đồng bộ khởi động bộ điều khiển CPU với mô-đun chức năng thông minh
		Cài đặt tương thích A-PLC	Cài đặt để sử dụng chuỗi vùng nhớ đặc biệt/thanh ghi đặc biệt MELSEC-A
		Cài đặt xử lý dịch vụ	Cài đặt thời gian xử lý dịch vụ. (chỉ cho mô-đun quốc tế QCPU)
	Cài đặt thay đổi mô-đun	Cài đặt tham số này để thay thế mô-đun CPU sử dụng một thẻ nhớ (chỉ dành cho mô-đun QCPU).	
	Tập PLC	Thanh ghi tệp	<ul style="list-style-type: none"> <li>Cài đặt tệp thanh ghi tệp được sử dụng trong một chương trình</li> <li>Cài đặt để truyền dữ liệu đến ROM chuẩn ở hoạt động sao lưu dữ liệu chốt (chỉ dành cho mô-đun quốc tế QCPU)</li> </ul>
		Chú giải tệp được sử dụng trong một lệnh	Cài đặt tệp chú giải các biến mà nó được sử dụng trong một chương trình.
		Giá trị thiết bị ban đầu	Cài đặt giá trị thiết bị ban đầu được sử dụng trên bộ điều khiển CPU
		Tệp cho thiết bị cục bộ	Cài đặt tệp thiết bị cục bộ được sử dụng trong một chương trình
		Tệp sử dụng cho lệnh SP.DEVST/S.DEVLD	Cài đặt tệp lệnh đọc ghi ROM được sử dụng trong một chương trình (chỉ cho mô-đun quốc tế QCPU)
	RAS PLC	Cài đặt WDT (đồng hồ hẹn giờ)	Cài đặt WDT của bộ điều khiển chương trình CPU.
		Kiểm tra lỗi	Cài đặt để phát hiện lỗi được chỉ định
		Phương thức hoạt động khi có một lỗi	Cài đặt bộ điều khiển khả trình khi có một lỗi được phát hiện
		Quét hằng số	Cài đặt chu kỳ quét hằng số
		Lịch sử hồng học	Cài đặt đích lưu trữ lịch sử lỗi của bộ điều khiển CPU (chỉ dành cho mô-đun hiệu suất cao QCPU)
		Thời gian thực hiện chương trình tốc độ thấp	Cài đặt thời gian thực hiện chương trình tốc độ thấp trong mỗi chu kỳ quét. (chỉ dành cho mô-đun hiệu suất cao QCPU)

Mục		Mô tả	
Tham số PLC	Tập tin khởi động	Tùy chọn khởi động	Cài đặt để làm sạch chương trình khi khởi động lên.
		Cài đặt tập tin khởi động	Cài đặt dạng, tên dữ liệu, ổ đĩa nguồn truyền và ổ đĩa đích truyền trong tập tin khởi động
	Chương trình	Cài đặt tên tệp và các kiểu thực hiện khi một vài chương trình được ghi đến bộ khởi động CPU	
	SFC	Cài đặt phương thức khởi động và điều kiện khởi động của một chương trình SFC và phương thức đầu ra ở khối dừng	
	Biến	Số biến	Cài đặt số điểm được sử dụng cho mỗi thiết bị của bộ điều khiển chương trình CPU
		Chốt (1) bắt đầu/kết thúc	Cài đặt phạm vi chốt (số biến ban đầu/số biến kết thúc) với chuyển mạch RESET/L.CLR hoặc một thao tác mở khóa từ xa.
		Chốt (2) bắt đầu/kết thúc	Cài đặt phạm vi chốt (số biến ban đầu/số biến kết thúc) không xóa với chuyển mạch RESET/L.CLR hoặc một thao tác mở khóa từ xa.
		Khởi động/kết thúc biến cục bộ	Cài đặt phạm vi (số biến ban đầu/số biến kết thúc) của các biến như một thiết bị cục bộ.
		Cài đặt mở rộng thanh ghi tệp	Cài đặt mở rộng thanh ghi dữ liệu và thanh ghi liên kết. (chỉ dành cho mô-đun quốc tế QCPU)
		Cài đặt chỉ số cho biến ZR	Cài đặt số khởi động của Z đến 32 bit chỉ số, hoặc sử dụng thanh ghi chỉ số ZZ cho cài đặt 32-bit chỉ số (chỉ dành cho mô-đun quốc tế QCPU)
	Chỉ định I/O	Chỉ định I/O	Cài đặt tên, loại, số điểm của I/O và số khởi động I/O cho mỗi mô-đun được gán trên mỗi đơn vị cơ sở
		Cài đặt cơ bản	Cài đặt loại, số khe của đơn vị cơ bản, loại mô-đun cấp nguồn, và loại cáp mở rộng.
	Cài đặt đa CPU	Số PLC	Cài đặt số bộ điều khiển CPU trong hệ thống sử dụng nhiều CPU.
		Phương thức hoạt động	Cài đặt phương thức hoạt động của hệ thống đa CPU khi một lỗi dừng diễn ra trong bất kỳ CPU nào từ số 2 đến số 4.
		Trạm chủ	Cài đặt số CPU cho CPU chủ.
		Cài đặt khởi động đồng bộ đa CPU	Chọn mô-đun CPU được khởi động đồng bộ
		Thay đổi mô-đun online	Thiết lập để cho phép thay đổi mô-đun online trong hệ thống đa CPU.
		Chia sẻ I/O khi sử dụng nhiều CPU	Cài đặt để lấy lại trạng thái I/O của mô-đun I/O hoặc mô-đun chức năng thông minh được điều khiển bởi bộ điều khiển CPU khác.
		Cài đặt vùng giao tiếp (cài đặt làm mới)	Cài đặt chia sẻ bộ nhớ dữ liệu khả dụng giữa các CPU.
		Cài đặt vùng truyền dẫn đa CPU tốc độ cao	Cài đặt vùng người sử dụng, tự động làm mới, xác nhận chỉ định và vùng hệ thống.
	Cài đặt cổng Ethernet tích hợp sẵn	Cài đặt địa chỉ IP	Cài đặt địa chỉ IP và định dạng đầu vào của địa chỉ IP.
		Mã dữ liệu giao tiếp	Chọn mã nhị phân hoặc mã ASCII cho giao tiếp
		Nút cài đặt mở	Cài đặt giao thức hệ thống mở và số cổng trạm chủ.
		Nút cài đặt FTP	Cài đặt để sử dụng hàm FTP
		Nút cài đặt thời gian	Cài đặt để sử dụng hàm SNTP và định thời.
	Giao tiếp kiểu nối tiếp	Tốc độ truyền	Cài đặt tốc độ truyền
		Kiểm tra tổng	Cài đặt kiểm tra tổng.
Thời gian chờ truyền		Cài đặt thời gian chờ truyền	
Thay đổi trực tuyến		Cài đặt để cho phép thay đổi chương trình trực tuyến.	
Tham số mạng	Ehternet/CC IE/MELSECNET	Cài đặt tham số mạng cho Ehternet, MELSECNET/10, MELSECNET/H, và liên kết CC	
	CC-LINK	Cài đặt tham số cho CC-LINK	
Mật khẩu từ xa		Cài đặt mật khẩu để giới hạn truy nhập thông qua Ethernet hoặc mô-đun giao tiếp đặc biệt.	



- Khi GX Works2 khởi động, nó dùng các giá trị thiết lập sẵn như các tham số. Các giá trị này là mặc định (các giá trị ban đầu)
- Bộ điều khiển chương trình có thể chạy với các giá trị như cũ, tuy nhiên, thay đổi chúng trong một phạm vi xác định là cần thiết.

Ví dụ hoạt động: thay đổi chế độ hoạt động khi một lỗi tồn tại

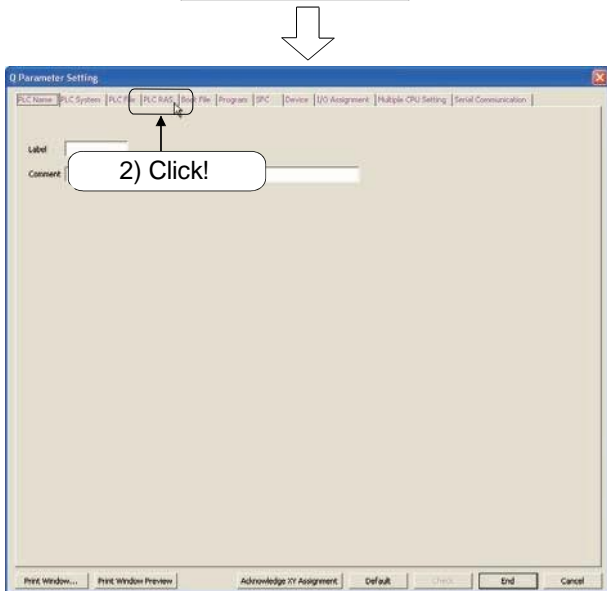
Khi lỗi tính toán gây ra, bộ điều khiển chương trình sẽ thay đổi trạng thái STOP ở giá trị mặc định, tuy nhiên, việc thay đổi các tham số sẽ duy trì bộ điều khiển CPU chạy.

Ví dụ lỗi tính toán

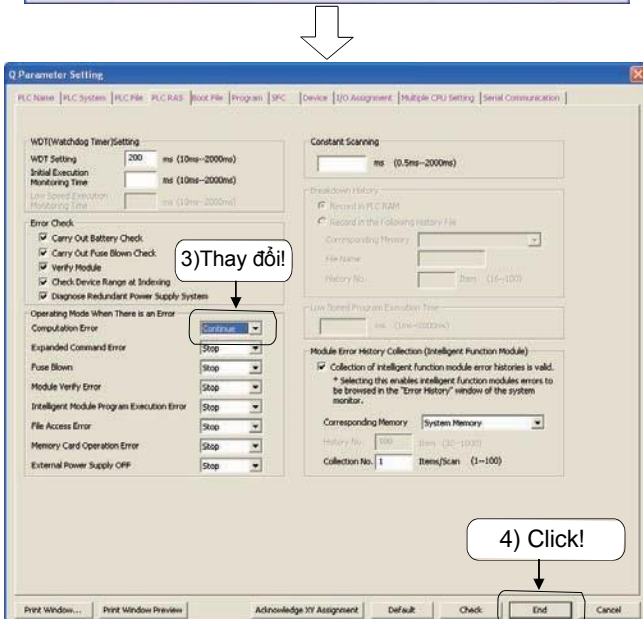
- Trong lệnh chia, quá trình xử lý chia dư 0 được thực hiện.



- 1) Click đúp "PLC parameter" trên cửa sổ Navigation



- 2) Hộp thoại cài đặt tham số Q sẽ xuất hiện. Click tab "PLC RAS" .



- 3) Thay đổi cài đặt "Computation Error" trong "Operation Mode When There Is an Error" sang "Continue"

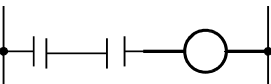
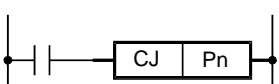
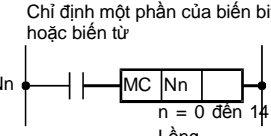
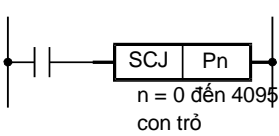
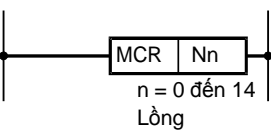
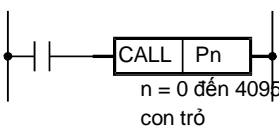
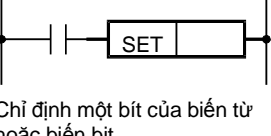
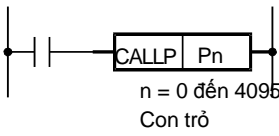
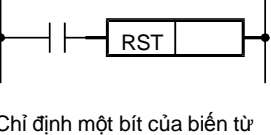
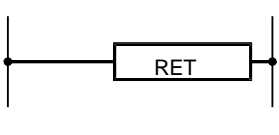
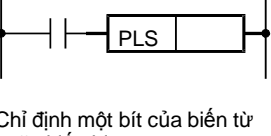
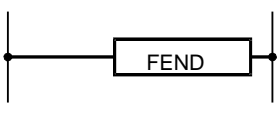
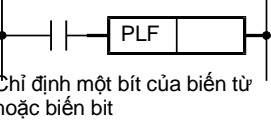
- 4) Click **End**

MEMO

## CHƯƠNG 4 CÁC LỆNH CƠ BẢN VÀ LỆNH TUẦN TỰ - PHẦN 1-

### 4.1 Danh sách các lệnh được giải thích trong chương này

Trong chương này giải thích các lệnh tuần tự và các lệnh cơ bản như bảng dưới đây:


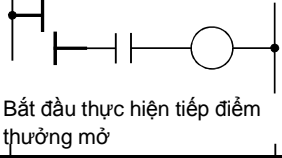

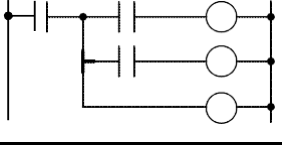

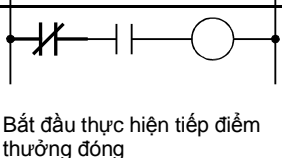

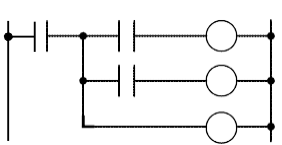




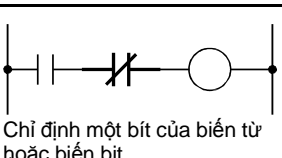


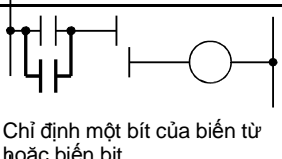

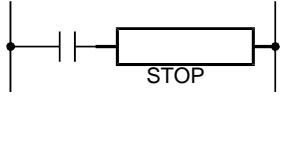

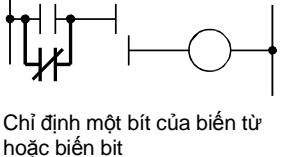



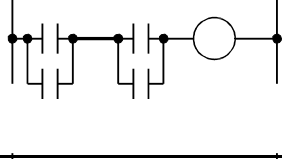

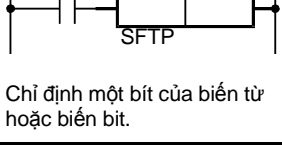

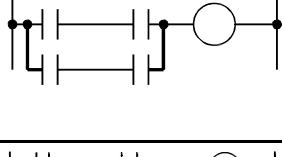

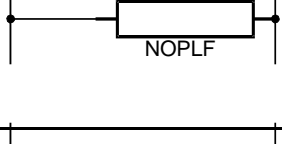

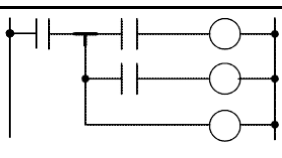

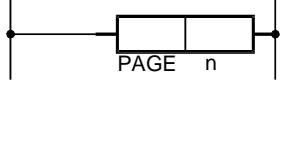
Kí hiệu lệnh (Tên)	Chức năng	Biểu tượng (Các biến được sử dụng)	Kí hiệu lệnh (Tên)	Chức năng	Biểu tượng (Các biến được sử dụng)
<b>OUT</b> Ra	Đầu ra cuộn dây	 Chỉ định một phần của biến bit hoặc biến từ	<b>CJ</b>	Điều kiện nhảy (không trở)	 n = 0 to 4095 con trở
<b>MC</b> Điều khiển chính	Khởi động điều khiển chính*1	 Chỉ định một phần của biến bit hoặc biến từ Nn n = 0 đến 14 Lồng	<b>SCJ</b>	Điều kiện nhảy (Nhảy sau khi quét)	 n = 0 đến 4095 con trở
<b>MCR</b> Reset điều khiển chính	Kết thúc kiểm soát chính	 n = 0 đến 14 Lồng	<b>CALL</b>	Gọi chương trình con	 n = 0 đến 4095 con trở
<b>SET</b> Cài đặt	Cài đặt các biến	 Chỉ định một bit của biến từ hoặc biến bit	<b>CALLP</b>	Gọi một chương trình con	 n = 0 đến 4095 Con trở
<b>RST</b> Cài đặt lại	Cài đặt lại các biến	 Chỉ định một bit của biến từ hoặc biến bit	<b>RET</b> Trở về	Trở về từ một chương trình con	
<b>PLS</b> Xung	Xung Tạo các xung cho một vòng chương trình khi một tín vào hiệu tắt	 Chỉ định một bit của biến từ hoặc biến bit	<b>FEND</b>	Kết thúc một chương trình thường xuyên chính	
<b>PLF</b> Pulf	Pulf Tạo các xung cho một vòng chương trình khi một tín vào hiệu tắt	 Chỉ định một bit của biến từ hoặc biến bit			

\*1: Trong GX Works2, trạng thái on/off của kiểm soát chính được hiển thị trong một thẻ trên màn hình.


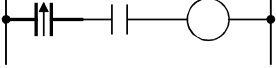

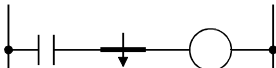

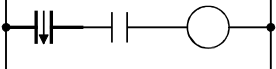

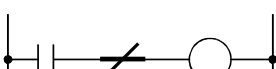

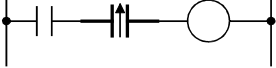

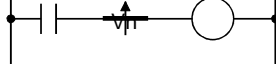

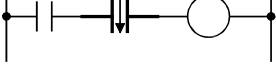

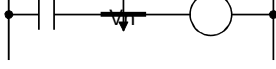

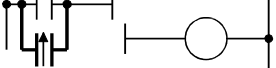

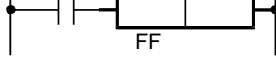

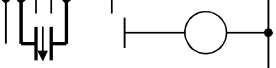



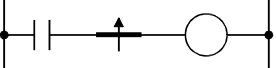

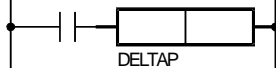
<Danh sách các lệnh không được giải thích trong chương này: phần 1>

"Giới thiệu: KHÓA HỌC PLC "các lệnh được trình bày bên dưới thì PLC dòng A cũng hỗ trợ .

Tham khảo "MELSEC-Q/L Programming Manual Common Instruction" để biết chi tiết hơn.

Kí hiệu lệnh (Tên)	Chức năng	Biểu tượng (Các biến được sử dụng)	Kí hiệu lệnh (tên)	Chức năng	Biểu tượng (Các biến được sử dụng)
 LD Tải	Thao tác lệnh logic (Bắt đầu thực hiện một tiếp điểm thường mở)	 Bắt đầu thực hiện tiếp điểm thường mở	 MRD Dây dẫn	Rẽ nhánh	
 LDI Tải ngược	Thao tác lệnh logic (Bắt đầu thực hiện một tiếp điểm thường đóng)	 Bắt đầu thực hiện tiếp điểm thường đóng	 MPP Pop	Kết thúc rẽ nhánh	
 AND And	Thao tác lệnh logic AND (Kết nối nối tiếp của các tiếp điểm thường mở.)	 Chỉ định một bit của biến từ hoặc biến bit	 NOP Nop	Được bỏ qua	Để trống hoặc xóa chương trình
 ANI And ngược	Thao tác lệnh logic ngược lệnh AND (Kết nối nối tiếp của các tiếp điểm thường đóng)	 Chỉ định một bit của biến từ hoặc biến bit	 END Kết thúc	Quá trình END kết thúc một chương trình	Phải được sử dụng như việc kết thúc một chương trình
 OR Or	Thao tác lệnh logic OR (Kết nối song song của các tiếp điểm thường mở)	 Chỉ định một bit của biến từ hoặc biến bit	 STOP	Dừng hoạt động	
 ORI Or ngược	Thao tác logic ngược OR (Kết nối song song của các tiếp điểm thường đóng)	 Chỉ định một bit của biến từ hoặc biến bit	 SFT Dịch	Dịch một bit của biến	 Chỉ định một bit của biến từ hoặc biến bit
 ANB Khối And	Thao tác lệnh AND giữa các khối logic (Kết nối nối tiếp của các khối)	 Chỉ định một bit của biến từ hoặc biến bit.	 SFTP Dịch P	Dịch một bit của biến ( với xung )	 Chỉ định một bit của biến từ hoặc biến bit.
 ORB Khối Or	Thao tác lệnh OR giữa các khối logic (Kết nối song song của các khối)	 Chỉ định một bit của biến từ hoặc biến bit.	 NOPLF	Bỏ qua (để một trang ngăn)	
 MPS Đầy	Khởi động một nhánh	 Chỉ định một bit của biến từ hoặc biến bit.	 PAGE	Bỏ qua (công nhận như bước số 0 của N trang)	

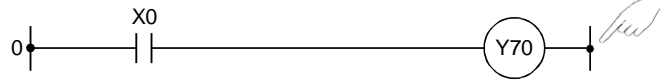
<Danh sách các lệnh không được giải thích trong chương này: phần 2>  
 Các lệnh được liệt kê dưới đây dành cho PLC dòng Q và không hỗ trợ cho PLC dòng A  
 Một phần được giải thích trong "Q Programming Applied Course".  
 Tham khảo "MELSEC-Q/L Programming Manual Common Instruction" để biết thêm chi tiết.

Kí hiệu lệnh (Tên)	Chức năng	Biểu tượng (các thiết bị được sử dụng)	Kí hiệu lệnh (Tên)	Chức năng	Biểu tượng (các thiết bị được sử dụng)
 LDP Tải P	Xung sườn lên	 Chỉ định một bit của biến từ hoặc biến bit	 MEF	Chuyển đổi kết quả thành xung sườn xuống	 Chỉ định một bit của biến từ hoặc biến bit
 LDF Tải F	Xung sườn xuống	 Chỉ định một bit của biến từ hoặc biến bit	 INV Nghịch đảo	Nghịch đảo kết quả xử lý	 Chỉ định một bit của biến từ hoặc biến bit
 ANDP And P	Nối nối tiếp tiếp điểm xung lên	 Chỉ định một bit của biến từ hoặc biến bit	 EGP Edge P	Chuyển đổi kết quả thành xung sườn xuống	 Chỉ định một bit của biến từ hoặc biến bit
 ANDF And F	Kết nối nối tiếp của xung xuống	 Chỉ định một bit của biến từ hoặc biến bit	 EGF Edge F	Chuyển đổi kết quả thành xung sườn xuống	 Chỉ định một bit của biến từ hoặc biến bit
 ORP Or P	Kết nối song song thêm xung lên	 Chỉ định một bit của biến từ hoặc biến bit	 FF	Nghịch đảo đầu vào của một biến	 Chỉ định một bit của biến từ hoặc biến bit
 ORF Or F	Kết nối song song thêm xung xuống	 Chỉ định một bit của biến từ hoặc biến bit	 DELTA Delta	Chuyển đổi đầu ra trực tiếp thành xung	 DY
 MEP	Chuyển kết quả thành xung sườn lên	 Chỉ định một bit của biến từ hoặc biến bit	 DELTAP Delta P	Chuyển đổi đầu ra trực tiếp thành xung	 DY

4.2 Sự khác nhau giữa **OUT** và **SET** / **RST**

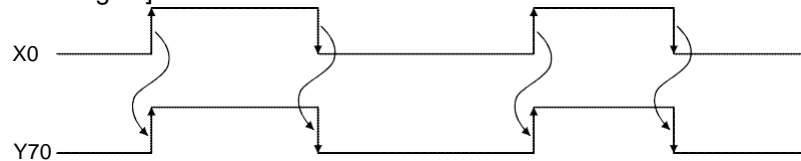
Tên dự án	QB-1
Tên dự án	MAIN

**Lệnh OUT**



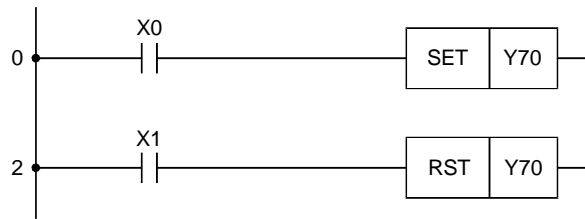
- Lệnh OUT biến được bật khi điều kiện đầu vào bật và biến tắt khi điều kiện đầu vào tắt.

[Biểu đồ thời gian]



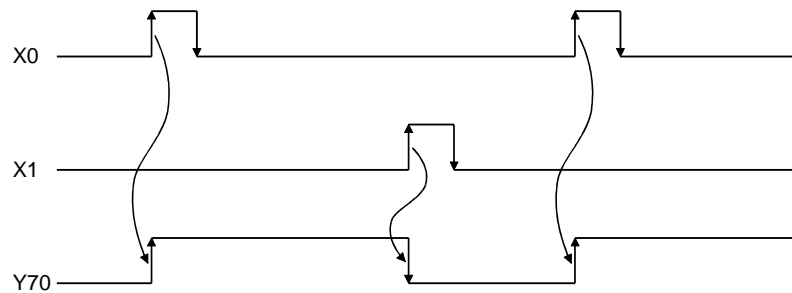
Tên dự án	QB-2
Tên dự án	MAIN

**Lệnh SET/RST**



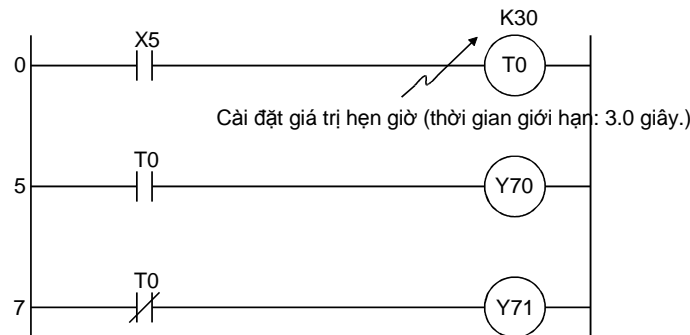
- Lệnh SET bật biến khi điều kiện đầu vào bật, và giữ trạng thái biến thậm chí khi điều kiện đầu vào tắt.  
Tắt biến khi sử dụng lệnh RST

[Biểu đồ thời gian]



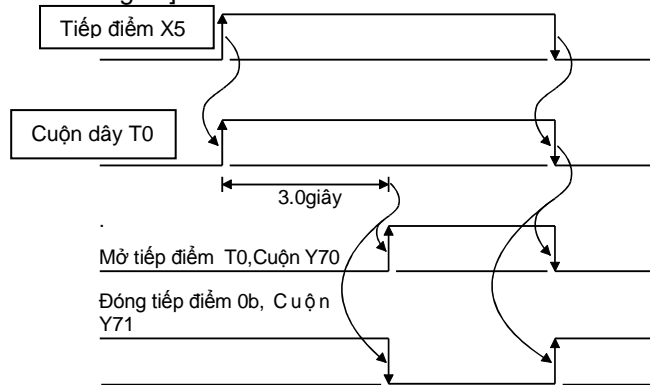
### 4.3 Đo bởi bộ định thời

Tên dự án	QB-3
Tên chương trình	MAIN



\*: OUT T là một lệnh 4 bước.

[Biểu đồ thời gian]



- Tiếp điểm định thời hoạt động trễ bởi việc cài đặt thời gian sau khi cuộn dây được cấp nguồn. (định thời trễ quá trình mở)
- Phạm vi cài đặt định thời từ K1 đến K32767. Định thời tốc độ thấp (100 ms) 0.1 đến 3276.7giây. Định thời tốc độ cao (10ms) 0.01 đến 327.67sec.
- Khi giá trị định thời được cài đặt là 0, nó được bật bởi việc thực hiện lệnh.

- Bốn dạng định thời khả dụng.

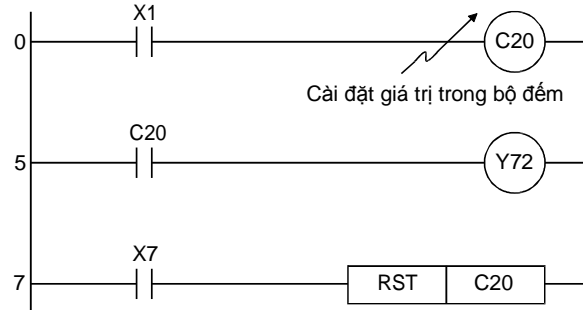
Dạng	Số định thời
Định thời tốc độ thấp..... Đếm thời gian theo đơn vị 100ms.	Mặc định T0 đến T2047 (2048)
Định thời tốc độ cao..... Đếm thời gian theo đơn vị 10ms.	
Định thời tốc độ thấp có nhớ..... Tích lũy thời gian theo đơn vị 100ms.	Mặc định: 0 Giá trị có thể được thay đổi sử dụng thông số.
Định thời tốc độ cao có nhớ..... Tích lũy thời gian theo đơn vị 10ms.	

- Thay đổi lệnh đầu ra (OUT) thành OUTH để chọn định thời tốc độ cao hoặc bộ định thời tốc độ cao có nhớ
- Để sử dụng bộ định thời có nhớ, cài đặt các tiếp điểm thiết bị cho bộ định thời có nhớ trong việc cài đặt thiết bị của tham số PLC.

Tham khảo mục 6.4 để xem giải thích về bộ định thời có nhớ

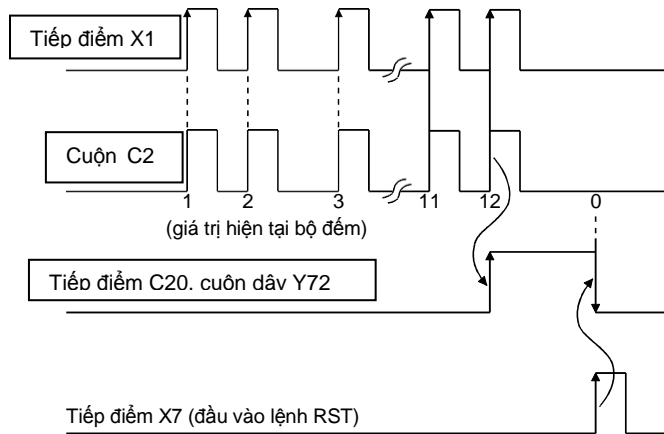
#### 4.4 Đếm bởi bộ đếm

Tên dự án	QB-4
Tên chương trình	MAIN



\*: OUT C là một lệnh 4 bước

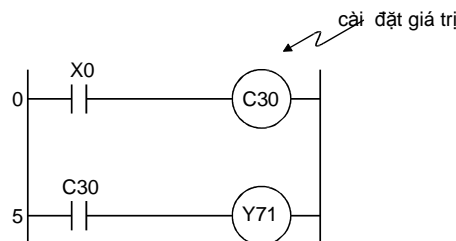
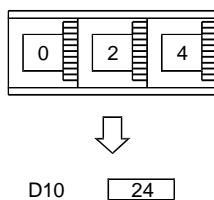
[Biểu đồ thời gian]



- Bộ đếm tiến đếm khi một tín hiệu đầu vào .
- Sau khi đến, các tín hiệu theo sau đầu vào không được đếm.
- Bộ đếm bắt đầu đếm, trạng thái tiếp điểm hoặc giá trị bộ đếm hiện tại không thay đổi cho đến tận khi lệnh RST được thực hiện
- Thực hiện lệnh RST trước khi bộ đếm quay trở lại 0.
- Phạm vi cài đặt bộ đếm từ K0 đến K32767. (K0 bật (đếm tiến) bằng việc thực hiện lệnh)

- Thêm và đó, Có thể sử dụng trực tiếp hằng số K, hoặc gián tiếp sử dụng biến D (thanh ghi dữ liệu )

Chuyển mạch số



- Bộ đếm C30 đếm nhánh trên của tín hiệu đầu vào X0 thành một số tương tự (như 24) được ghi bởi thanh ghi dữ liệu D10.
- Cấu hình gián tiếp hữu dụng cho các giá trị được chỉ rõ bởi một chuyển mạch số ngoài đến bộ đếm.

Cấu hình trực tiếp sử dụng thanh ghi dữ liệu D cũng có hiệu quả đối với bộ định thời

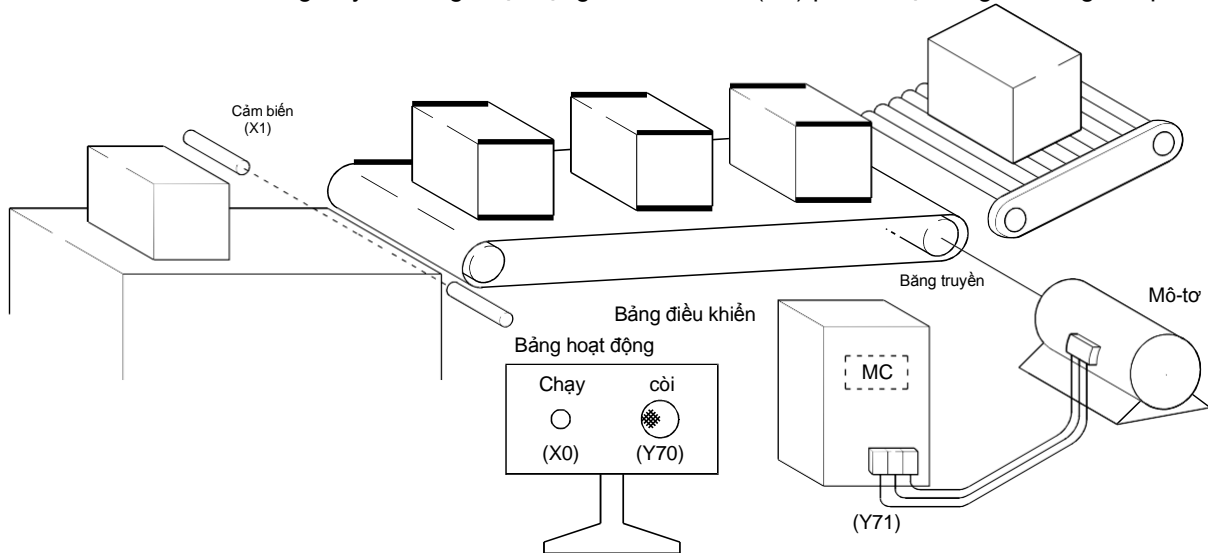


Tên dự án	QEX1
Tên chương trình	MAIN

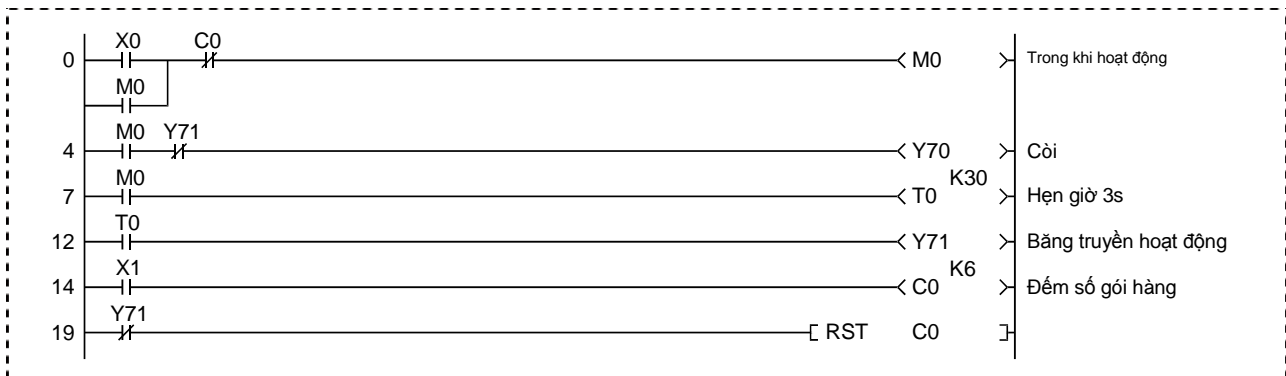
Ví dụ : Chương trình Ladder

Khi băng truyền hoạt động công tắc (X0) được bật, còi (Y70) kêu 3 hồi 2 lần và băng truyền (Y71) bắt đầu hoạt động.

Băng truyền dừng hoạt động khi cảm biến (X1) phát hiện 6 gói hàng đi qua.



Tạo chương trình ladder bên dưới và kiểm tra .



Trình tự thao tác

(1) Tạo một dự án

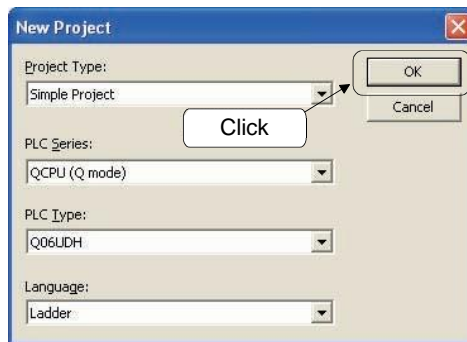
(a) Click  trên thanh công cụ.



(b) Hộp thoại New Project xuất hiện.

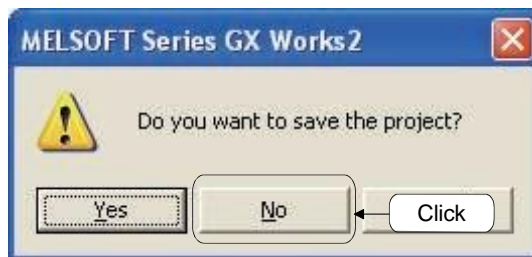
Cài đặt "Project Type" ở chế độ "Simple Project", "PLC Series" ở chế độ "QCPU (Q mode)",

Và "PLC Type" là "Q06UDH". Sau đó click



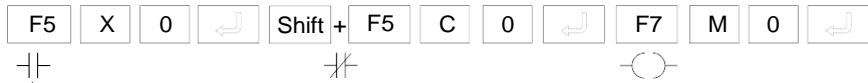
(c) Nếu dự án này đã tồn tại, hộp thoại xác nhận lưu dự án xuất hiện.

Click nút



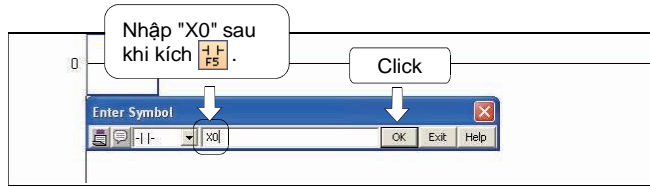
(d) Chuyển màn hình sang chế độ lập một dự án mới.

(2) Tạo một chương trình  
[sử dụng bàn phím]



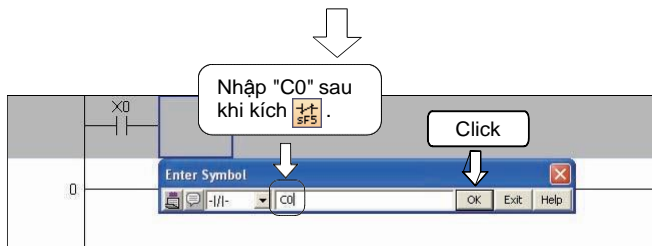
F4  
Chuyển

[Sử dụng các thanh công cụ]



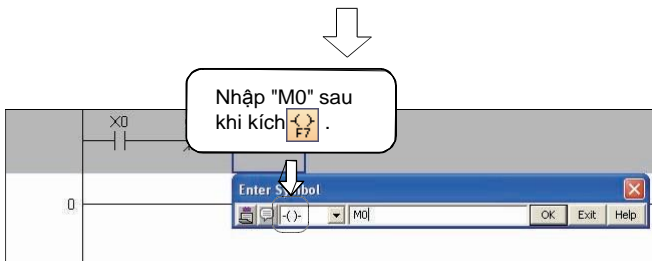
(a) Click trên thanh công cụ để mở cửa sổ Enter Symbol.

(b) Nhập "X0" trên bàn phím và click nút .



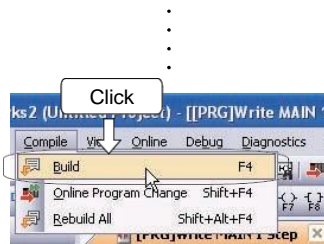
(c) Click trên thanh công cụ để mở cửa sổ Enter Symbol.

(d) Nhập "C0" trên bàn phím và click nút .



(e) Click trên thanh công cụ để mở cửa sổ Enter Symbol.

(f) Nhập "M0" trên bàn phím và click nút .




(g) Khi thiết lập mạch được hoàn thành, click [Compile] → [Build].

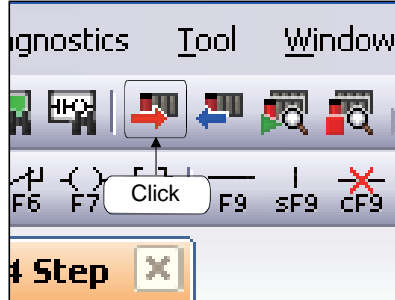
(3) Ghi một dự án cho bộ điều khiển khả trình

(a) Ghi một ladder được lập đến bộ nhớ trên bộ điều khiển.

Cài đặt chuyển mạch RUN/STOP của CPU sang STOP.

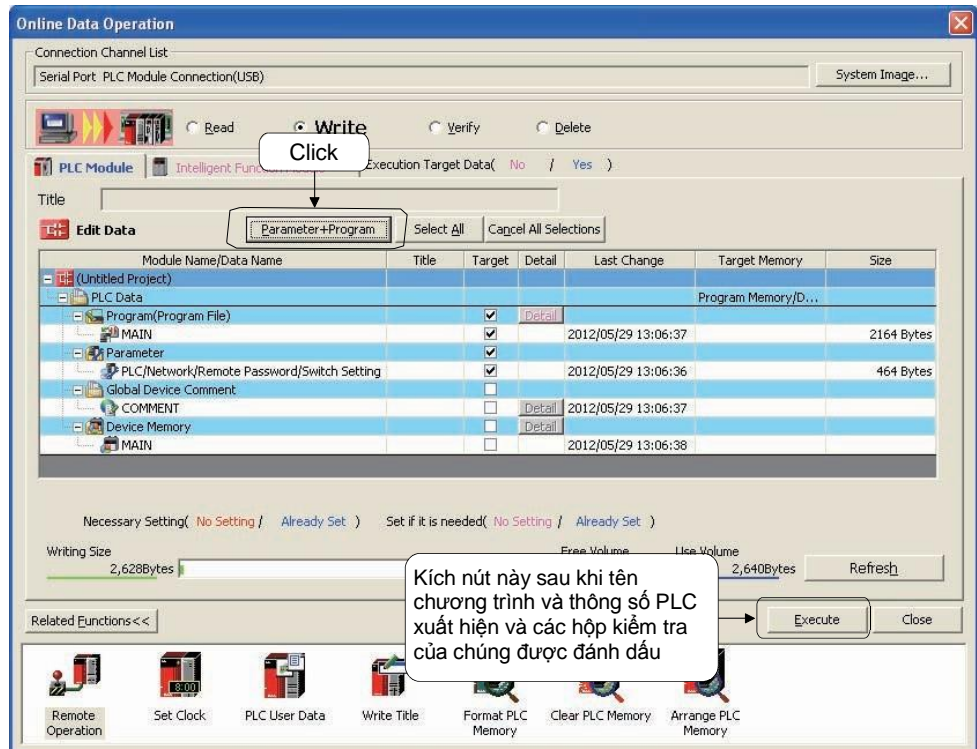
Click  trên thanh công cụ.

Hộp thoại Online Data Operation xuất hiện.

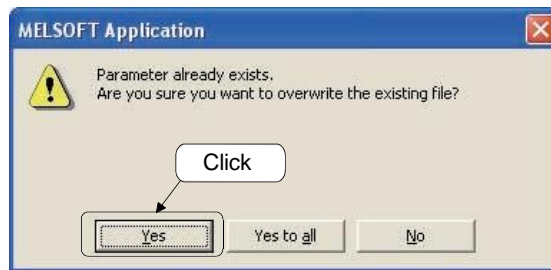


(b) Click  Parameter + Program . Các hộp kiểm tra chương trình đích và tham số đích hiện lên trong một cửa sổ được tự động đánh dấu (✓).

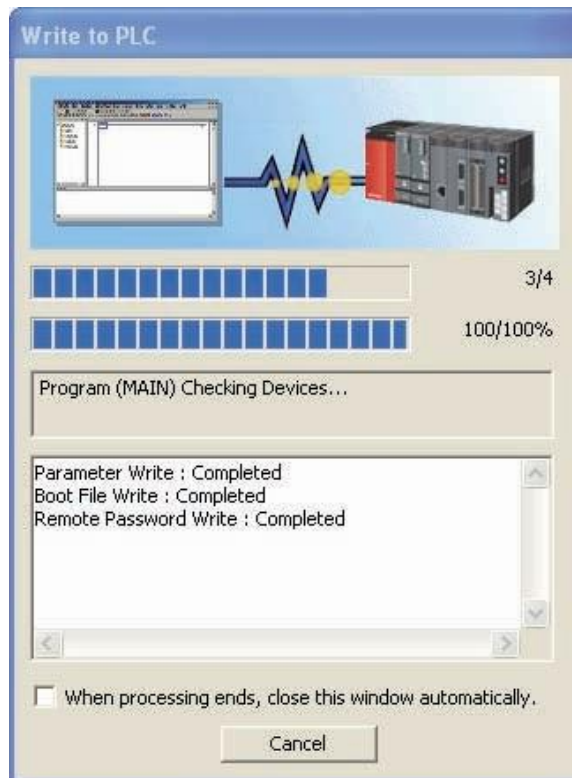
(c) Click  Execute .



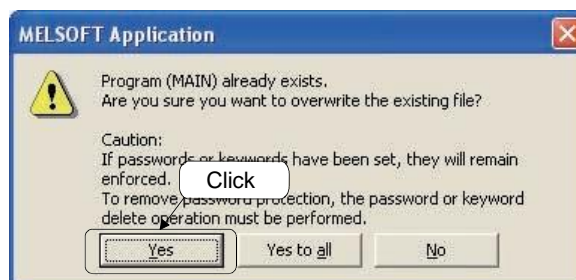
- (d) Nếu các tham số đã được ghi sẵn, một hộp thoại xác nhận đề tham số xuất hiện. Click



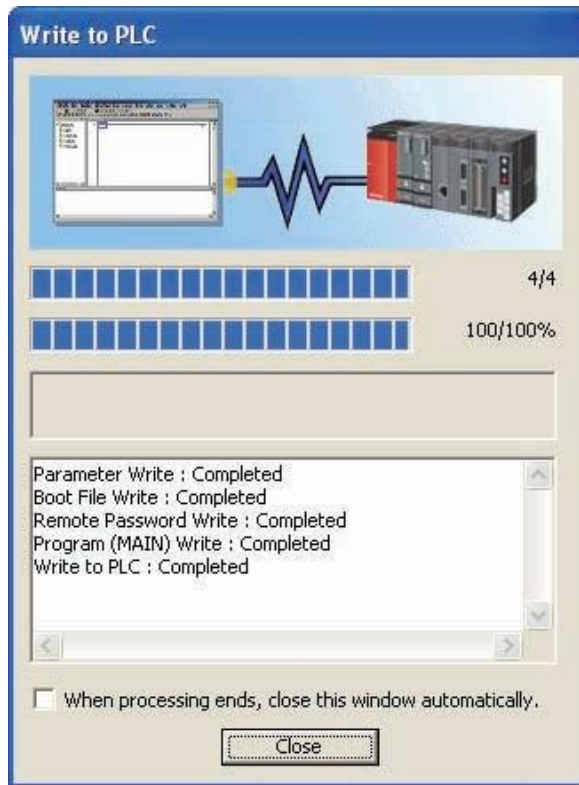
- (e) Hộp thoại ghi sang PLC xuất hiện



- (f) Nếu chương trình đã được ghi, hộp thoại xác nhận đề chương trình xuất hiện. Click



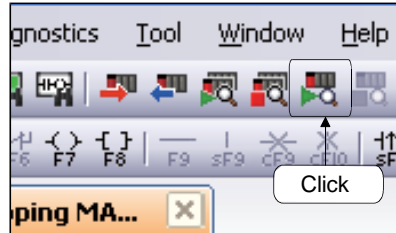
(g) Ghi một chương trình đến bộ điều khiển được hoàn thành.



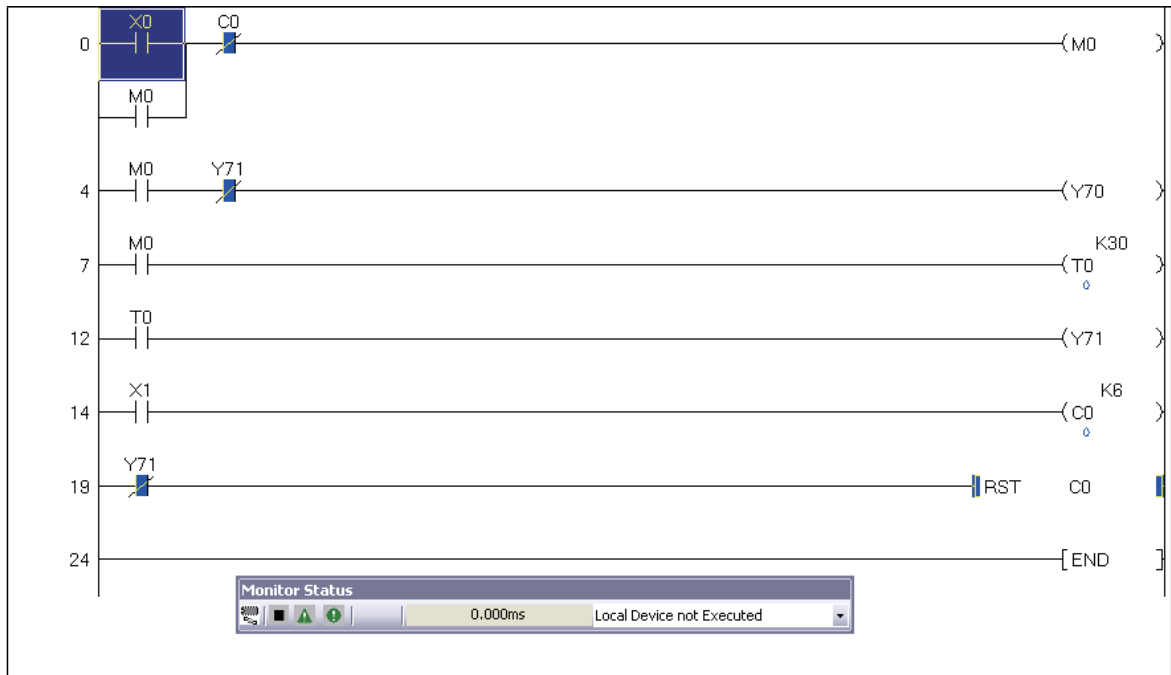
- (4) Giám sát một ladder  
Giám sát một ladder.

Giữ công tắc RESET/STOP/RUN trên CPU ở RESET nhiều hơn 1 giây, sau đó chuyển sang RUN.

- (a) Click  trên thanh công cụ.



- (b) Một màn hình ladder được sử dụng để giám sát ladder.



#### Hoạt động thực tế

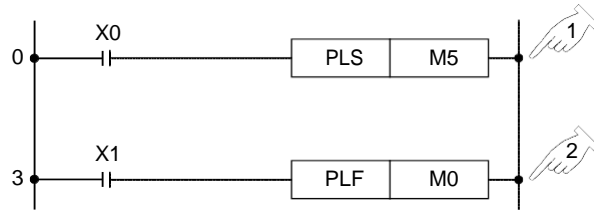
- 1) Bật công tắc (X0), bật Y70 và khởi động T0 cùng một lúc.
- 2) Khi bộ định thời T0 đếm 3 giây, Y70 tắt và Y71 bật cùng một lúc.
- 3) Bật hoặc tắt công tắc (X1). Bộ đếm C0 đếm số ON để tắt Y71 sau khi đếm 6 lần.

4.5 PLS Pulse ( Xung sườn lên khi có điều kiện đầu vào )

PLF Pulf (Xung sườn xuống khi có điều kiện đầu vào)

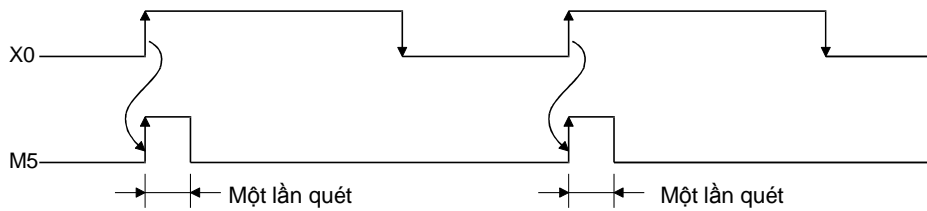


Tên dự án	QB-5
Tên chương trình	MAIN



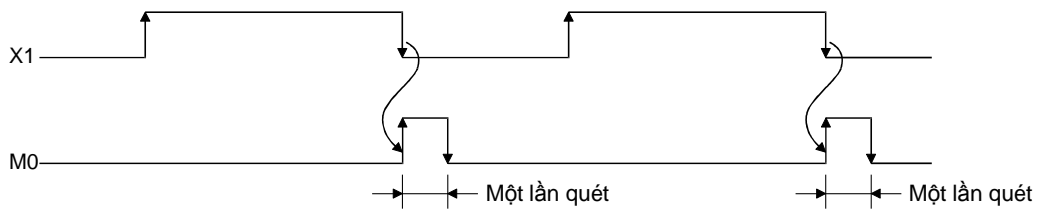
- 1 • Lệnh PLS cho phép ON biến trong khoảng một chu kỳ quét bắt đầu khi sườn lên của biến điều kiện đầu vào.

[Biểu đồ thời gian]



- 2 • Lệnh PLF cho phép ON biến trong khoảng một chu kỳ quét bắt đầu khi sườn xuống của biến điều kiện đầu vào.

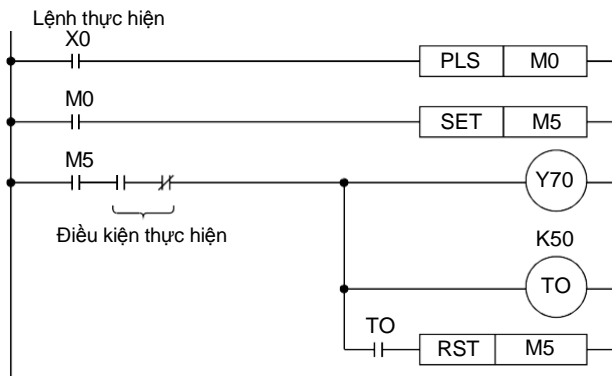
• [Biểu đồ thời gian]



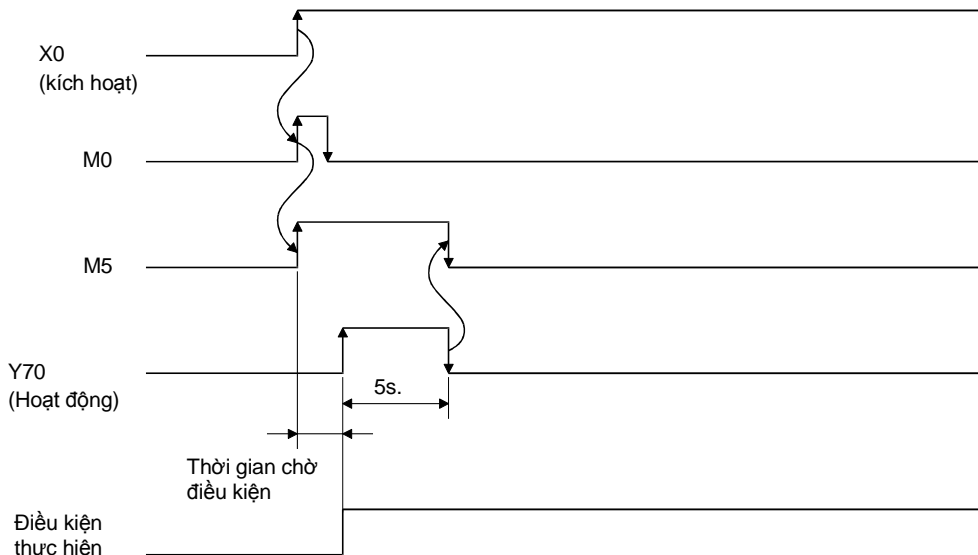


## Ứng dụng

- Các lệnh có thể được sử dụng trong chương trình dự phòng để chờ điều kiện hoạt động.

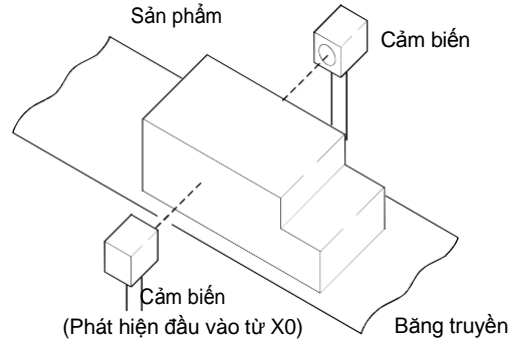
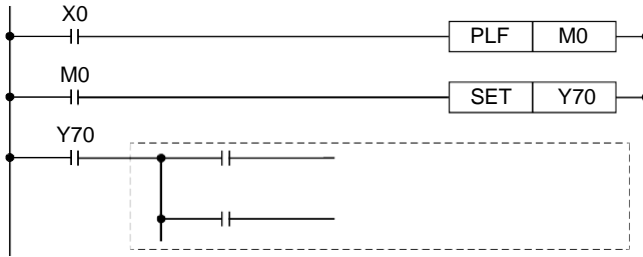


[Lược đồ thời gian]

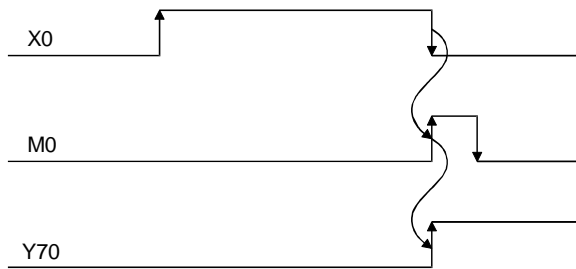


	Biến khả dụng												Số bước nhảy cơ bản							
	Biến trong (hệ thống hoặc người sử dụng)		Thanh ghi tệp	MELSECNET/10 (H) Direct Jn\		Mô-đun chức năng thông minh Un\G	Thanh ghi chỉ số	Hàng số		Con trỏ		Mức								
	Bit	Từ	R	Bit	Từ		Z	K	H	P	I	N								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>PLS</td><td>(D)</td></tr> <tr><td>PLF</td><td>(D)</td></tr> </table>	PLS	(D)	PLF	(D)	(D)	⌘	⌘	⌘	⌘	⌘	⌘									2
PLS	(D)																			
PLF	(D)																			

- Các lệnh có thể được sử dụng cho một chương trình để phát hiện sự chạy qua của các sản phẩm hoặc đối tượng.  
Sau khi một sản phẩm chạy qua được phát hiện, quá trình xử lý tiếp theo cho sản phẩm được bắt đầu..



[Lược đồ thời gian]

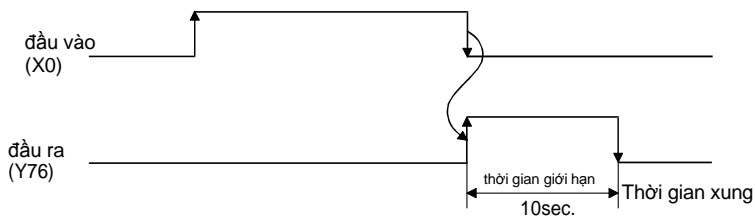


Các cách sử dụng khác nhau của PLS và PLF

Phần 1

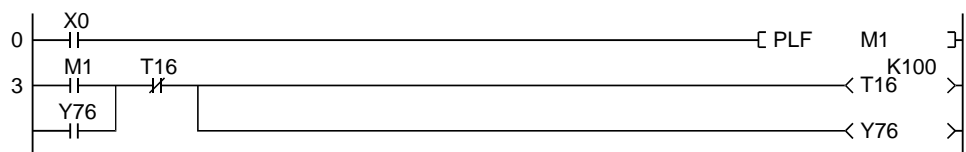
- Các lệnh có thể được sử dụng cho một chương trình mà chương trình đó thực hiện hoạt động đầu ra trên một chu kì khi tín hiệu đầu vào chuyển từ bật sang tắt.

[Lược đồ thời gian]



[Ví dụ chương trình]

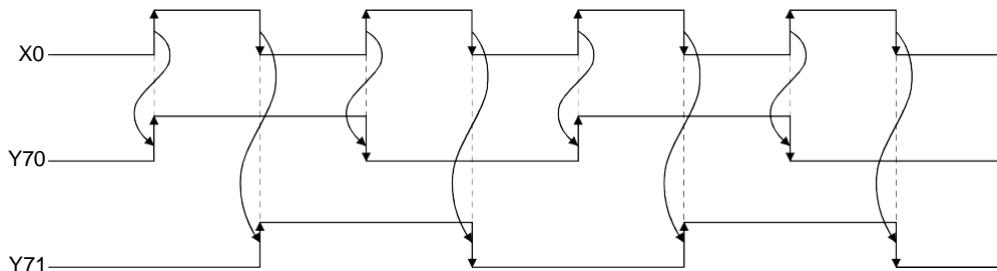
Tên dự án	QB-6
Tên chương trình	MAIN



- Một chương trình cho quá trình lặp như các trạng thái chuyển mạch on/off luân phiên nhau bởi việc nhấn các nút chuyển mạch có thể được thực hiện bằng các lệnh.

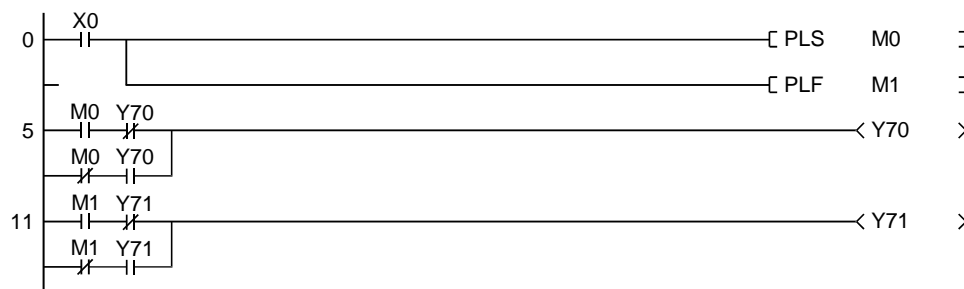
Nếu các lệnh PLS được sử dụng trong chương trình trên, thì sườn tăng được tạo ra khi nút bấm chuyển mạch được nhấn. Nếu lệnh PLF được sử dụng, thì sườn xuống được tạo ra khi chuyển mạch được giải phóng.

[Lược đồ thời gian]



[Ví dụ chương trình]

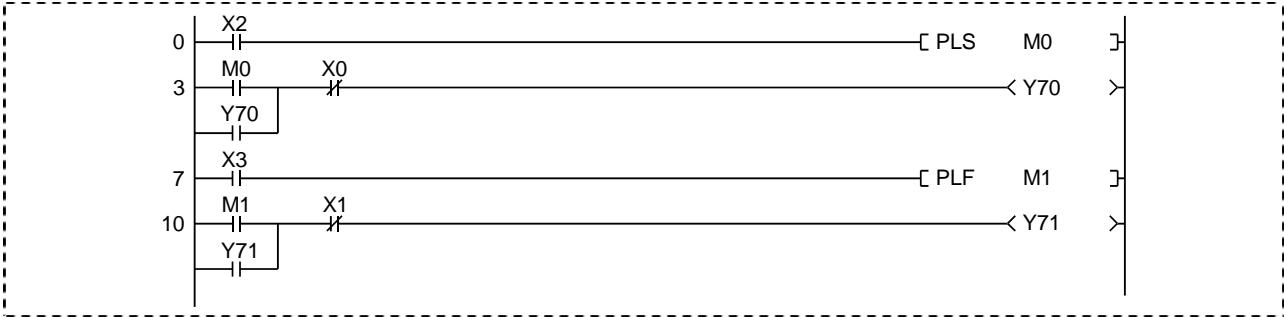
Tên dự án	QB-7
Tên chương trình	MAIN



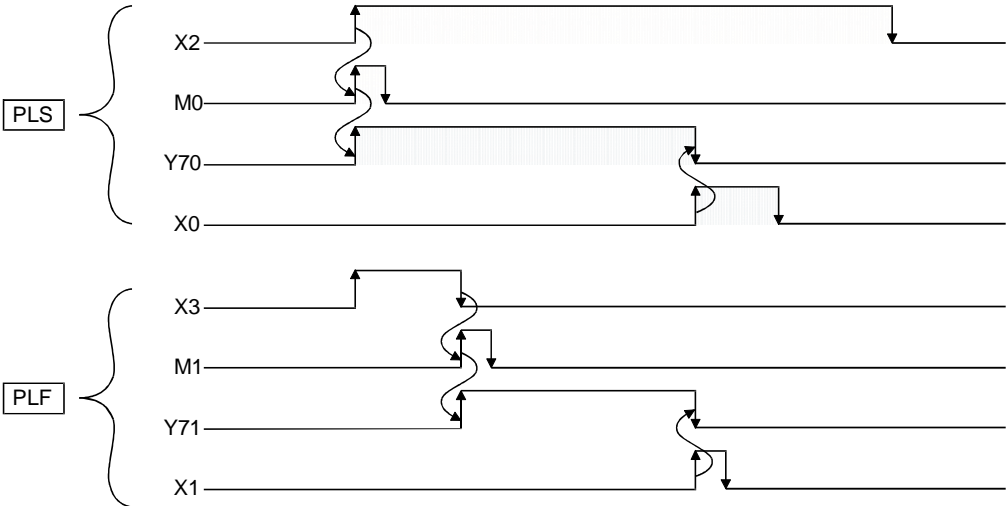
Tên dự án	QEX2
Tên chương trình	MAIN

Ví dụ Ladder

Thiết lập một ladder bên dưới và kiểm tra hoạt động của nó.

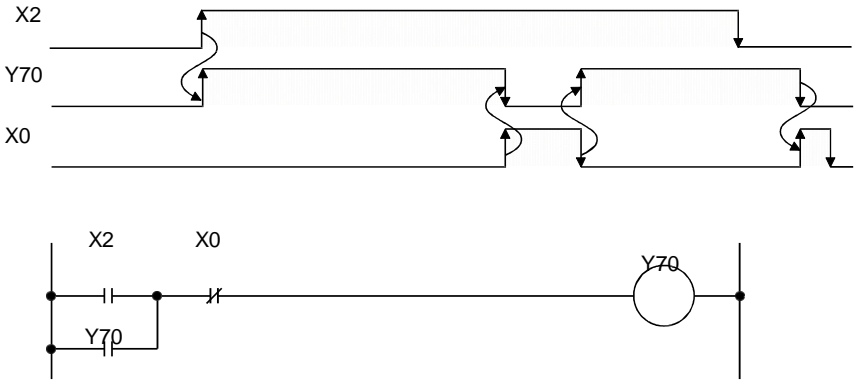


[Lược đồ thời gian]



**THAM KHẢO**

Dưới đây là biểu đồ thời gian của một ladder lên được lập trình sử dụng lệnh OUT. So sánh với một ladder lên sử dụng lệnh PLS.



**Trình tự hoạt động**

Các trình tự hoạt động dưới đây tương tự như **Trình tự hoạt động** trong mục 4.4.

- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Ghi dự án đến bộ điều khiển khả trình
- (4) Giám sát một ladder

**Hoạt động thực tế**

- Bật X2, bật Y70 và bật X0, tắt Y70. (Ngay cả khi X2 ở trạng thái bật, vẫn bật X0 và tắt Y70.)
- Bật X3, bật Y71 và bật X1, tắt Y71.

**Bài tập liên hệ** — Bài tập 3

<b>Chú ý</b>
--------------

Xử lý xung đầu vào không được yêu cầu đối với QCPU nó sử dụng một tiếp điểm dẫn

[Đối với A/AnSCPU]

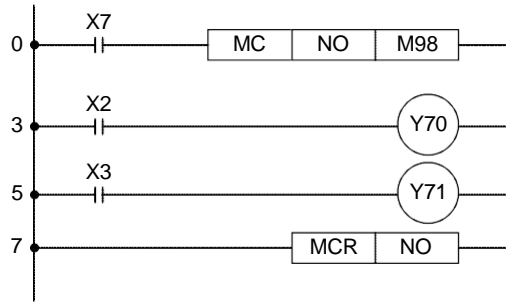
[Đối với QCPU]

Các lệnh hỗ trợ là: LDP, LDF, ANDP, ANDF, ORP và ORF.

4.6 **MC** Kiểm soát chính (Start)

**MCR** Cài đặt lại kiểm soát chính (End)

Tên dự án	QB-8
Tên chương trình	MAIN



- Chương trình trên là một chương trình cơ bản.
- MC**  $N$   $M$  sang **MCR**  $N$  (biểu thị "MC thành MCR" )

Số lồng sẵn sàng (N) cho "MC thành MCR" từ N0 đến N14.

- Chu kỳ quét được nhảy từ "MC sang MCR" hầu như không thay đổi.

Trạng thái biến của một chương trình được nhảy "MC thành MCR" trở nên như sau: Tất cả các thiết bị trong lệnh OUT được tắt.  
 Các biến trong lệnh SET, RST và SFT, bộ đếm và bộ định thời giữ nguyên trạng thái.  
 Định thời 100ms và 10ms được cài đặt lại về 0.

**Ứng dụng**

- Các lệnh có thể được sử dụng cho một chương trình đối với chuyển mạch hoạt động thủ công và tự động. (Xem ví dụ ladder.)

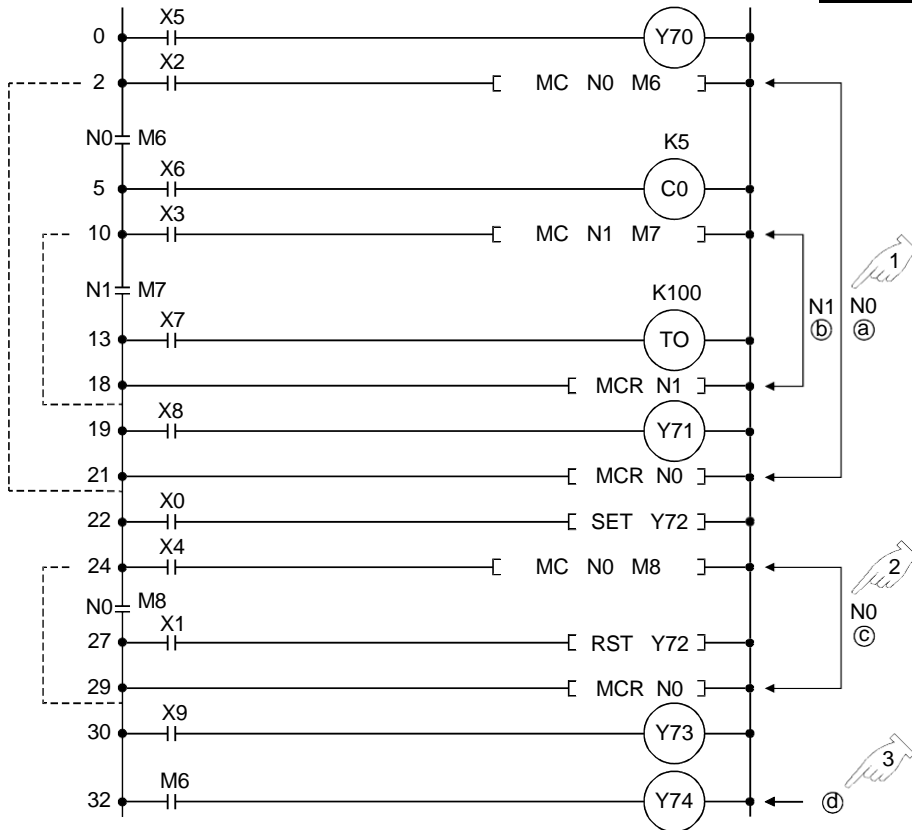
	Biến khả dụng											Số bước cơ bản			
	Biến trong (hệ thống hoặc người sử dụng)		Thanh ghi tập	MELSECNET/10 (H) Direct Jn\		Module chức năng thông minh Un\G	Thanh ghi chỉ số	Hàng số		Con trỏ			Mức		
	Bit	Từ	R	Bit	Từ		Z	K	H	P	I		N		
<b>MC</b> n $\text{\textcircled{D}}$	n														2
<b>MCR</b> n	$\text{\textcircled{D}}$	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$	$\text{\textcircled{P}}$								1

Số bước cơ bản của lệnh MC là 2 và của MCR là 1.

Ví dụ chương trình được lồng "MC với MCR"

- Các lệnh MC và MCR có thể được lồng với nhau như dưới đây.

Tên dự án	QB-9
Tên chương trình	MAIN



- 1 • Chương trình "MC với MCR" (a) được lồng dưới chương trình "MC với MCR" (b). Nó được gọi là kiến trúc lồng.

Trong trường hợp này

- (1) Chỉ định số lồng (N) của lệnh MC khi yêu cầu tăng
- (2) Chỉ định số lồng (N) của lệnh MCR được sử dụng đối với MC khi yêu cầu giảm

- 2 • Chương trình "MC với MCR" (a) có thể không phụ thuộc vào chương trình (c). Một số lồng tương tự (N) có thể được sử dụng trong suốt cả chương trình.

Số rơ-le trong (M) phải được thay đổi đối với mỗi lệnh.

- 3 • Như được trình bày trong chương trình (d). Số rơ-le trong M của MC được sử dụng một tiếp điểm.

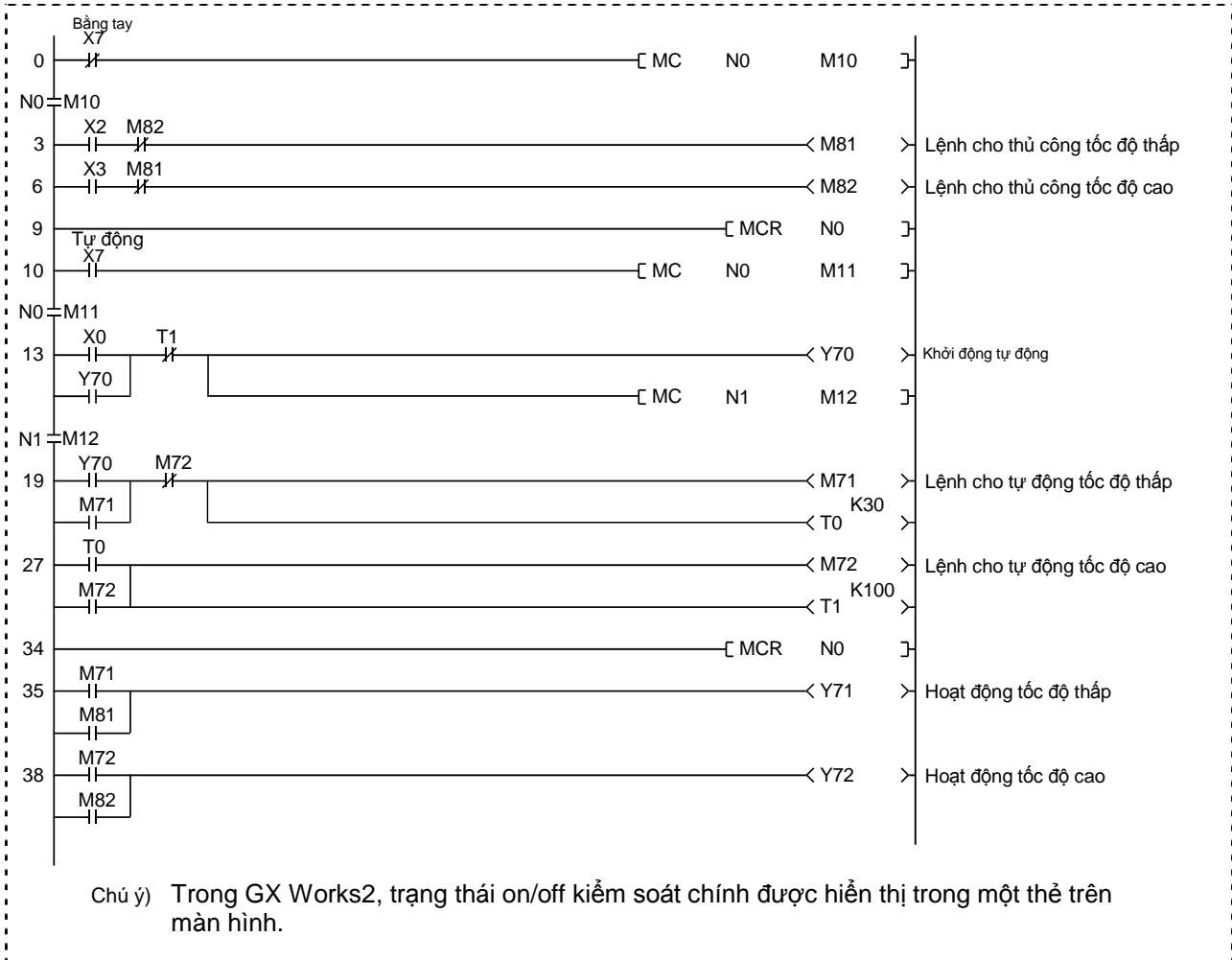
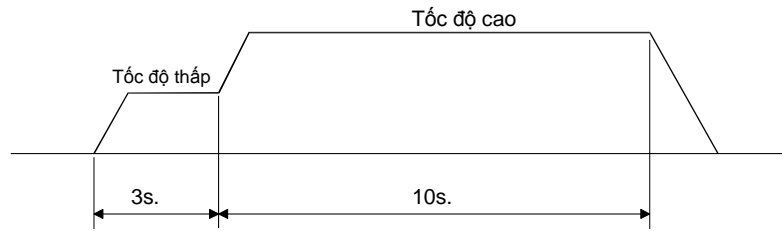
Chú ý) Trong GX Works2, trạng thái on/off của kiểm soát chính được hiển thị trong một thẻ trên màn hình.

Tên dự án	QEX3
Tên chương trình	MAIN

Ví dụ Ladder

Chương trình dưới đây chuyển giữa chế độ hoạt động bằng tay và tự động sử dụng lệnh MC và MCR.

- Khi chế độ hoạt động bằng tay được chọn bằng việc tắt X7;
  - 1) Điều chỉnh X2 để hệ thống chuyển sang chế độ làm việc tốc độ chậm.
  - 2) Điều chỉnh X3 để hệ thống chuyển sang chế độ làm việc tốc độ chậm.
- Khi chế độ hoạt động tự động được chọn bằng việc bật X7, hệ thống hoạt động trong chế độ tốc độ thấp trong 3 giây. Sau khi X0 được bật. Hệ thống chuyển sang chế độ tốc độ cao trong 10 giây rồi dừng.





#### Trình tự hoạt động

Các thủ tục dưới đây tương tự như **Operating Procedure** trong mục 4.4

- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Ghi dự án đến bộ điều khiển chương trình
- (4) Giám sát một ladder

#### Hoạt động thực tiễn

- Hoạt động thủ công được chọn bằng việc tắt chuyển mạch X7.  
Khi chuyển mạch X2 được bật, Y71 sáng và chế độ hoạt động tốc độ thấp được khởi động. Để chọn hoạt động tốc độ cao, bật chuyển mạch X3. Y72 sáng và chế độ hoạt động tốc độ cao khởi động.  
Hoạt động tự động được chọn bằng việc bật chuyển mạch X7.  
Khi chuyển mạch X0 được bật, Y70 sáng biểu thị chế độ hoạt động tự động được kích hoạt.  
Cùng thời điểm này, Y71 cũng sáng trong 3 giây, biểu thị hệ thống đang hoạt động ở chế độ tốc độ thấp. Sau khi 3 giây trôi qua, Y72 sáng trong 10 giây, biểu thị hệ thống hoạt động ở chế độ tốc độ cao. Sau đó dừng lại. (Y70, Y71, và Y72 dừng sáng lúc kết thúc.)

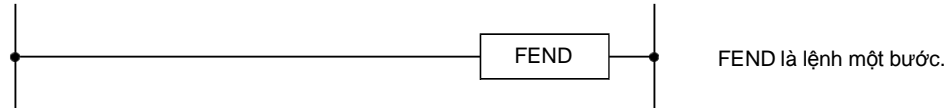
Chú ý
-------

Đối với lệnh MCR trong một khối chương trình lồng, tất cả kiểm soát chính trong một chương trình có thể được kết thúc chỉ với số lồng thấp nhất (N).
--

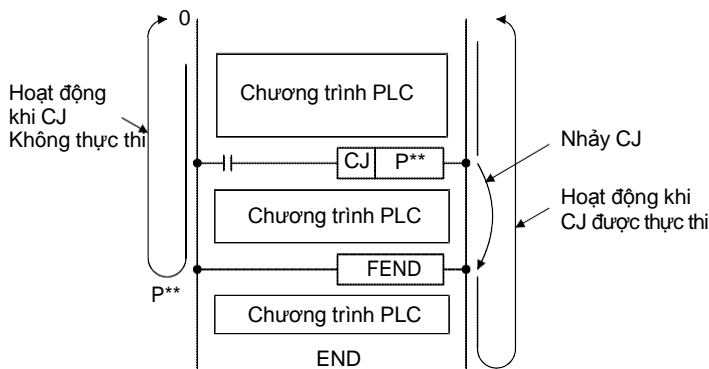
4.7 FEND / CJ / SCJ / CALL / RET

Tên dự án	QB-10
Tên chương trình	MAIN

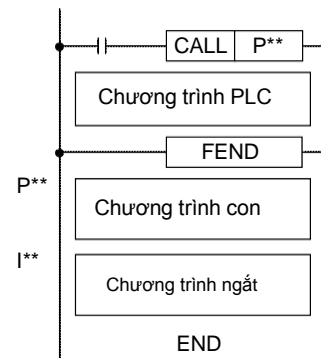
4.7.1 FEND F end



- Sử dụng lệnh FEND như lệnh END dưới điều kiện sau:
  - Khi mỗi chương trình PLC phải được hoạt động và kết thúc trong mỗi khối chương trình.  
Ví dụ: sử dụng lệnh FEND với các lệnh CJ và SCJ.
  - Khi sử dụng các chương trình con (các lệnh CALL và RET)
  - Khi sử dụng một chương trình ngắt.
- Sau mỗi lần thực hiện lệnh FEND, bộ điều khiển chương trình xử lý giá trị hiện thời của bộ định thời và bộ đếm và thực hiện kiểm tra tự chuẩn đoán, và khởi động lại từ bước 0.



(a) Khi hoạt động trong mỗi khối chương trình bởi lệnh CJ.



(b) Khi sử dụng các chương trình con và các chương trình ngắt

**Chú ý**

- Không có giới hạn số lệnh FEND trong một chương trình PLC, tuy nhiên, chúng không thể được sử dụng trong chương trình ngắt hoặc chương trình con.
- Lệnh FEND không thể được sử dụng để kết thúc một chương trình PLC chính hoặc con.  
Sử dụng lệnh END cho việc kết thúc toàn bộ chương trình.

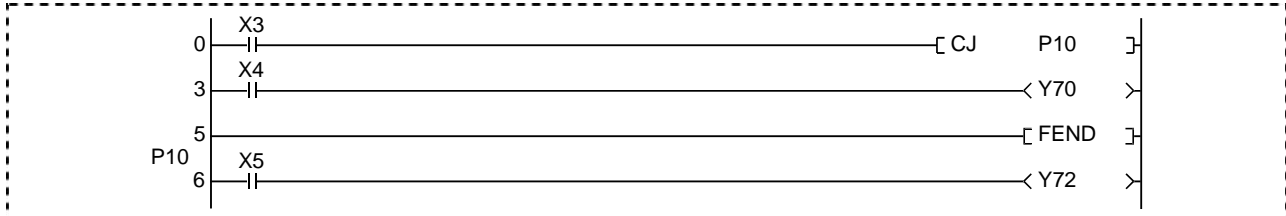
**Tham khảo**

Chương trình ngắt dừng quá trình hiện thời và xử lý ngắt trong lúc nhận một yêu cầu ngắt trong khi một chương trình thông thường đang được xử lý.

Tên dự án	QEX6
Tên chương trình	MAIN

Ví dụ Ladder

Tạo một Ladder bên dưới với GX Works2 và ghi nó đến CPU của một máy chứng minh. Sau đó kiểm tra lệnh FEND hoạt động chính xác hay không.



**Thủ tục hoạt động**

Thủ tục dưới đây tương tự như **Thủ tục hoạt động** trong mục 4.4

- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Ghi dự án đến bộ điều khiển chương trình
- (4) Giám sát một ladder

Hoạt động thực tế

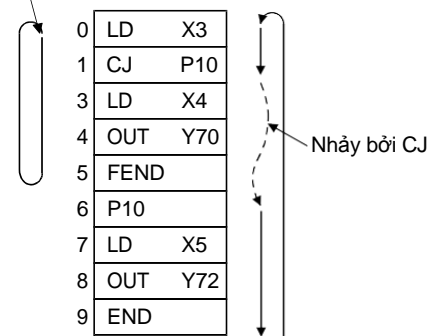
Xác nhận hoạt động của ladder mà nó được tạo với GX Works2 và được ghi tới CPU, bằng việc giám sát ladder trên màn hình .



- (1) Khi X3 tắt
  - (a) Hoạt động được thực hiện từ 0 đến FEND.
  - (b) Bật hoặc tắt X4 ,bật hoặc tắt Y70.
  - (c) Bật hoặc tắt X5, không thay đổi Y72.

- (2) Khi X3 bật
  - (a) Chương trình nhảy đến con trỏ P10 bởi lệnh CJ.
  - (b) Bật hoặc tắt X4, không thay đổi Y70.
  - (c) Bật hoặc tắt X5, bật hoặc tắt Y72.

Hoạt động khi X3 tắt

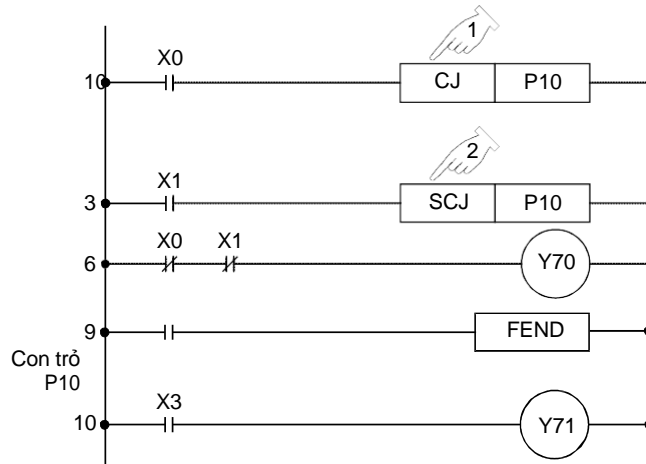


Bài tập liên hệ

—— Bài tập 4

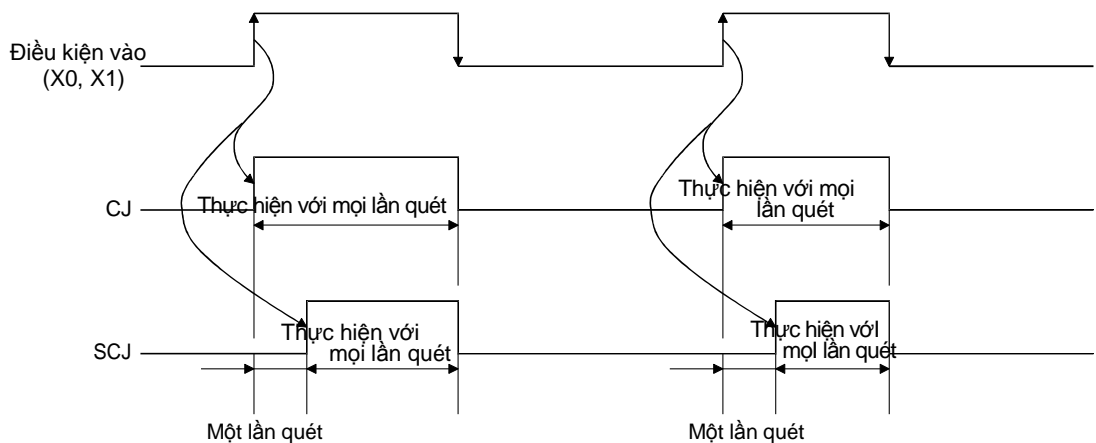
4.7.2 **CJ** (Nhảy có điều kiện: thực hiện ngay lập tức nhảy có điều kiện)

**SCJ** (Nhảy có điều kiện S: thực hiện nhảy có điều kiện sau một lần quét)



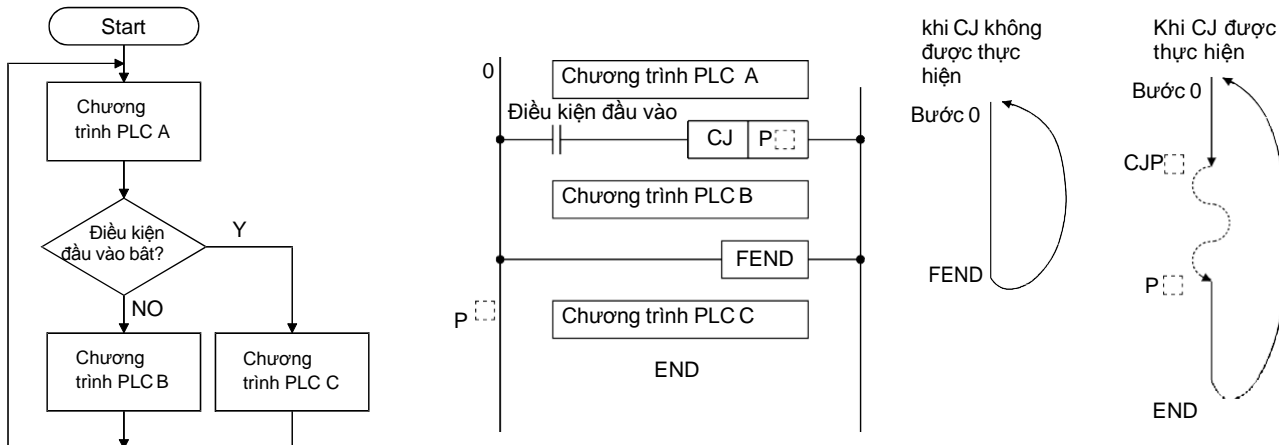
- 1 • Lệnh nhảy CJ thực hiện ngay lập tức việc nhảy một chương trình đến một địa chỉ được chỉ rõ (Số con trỏ) khi lệnh thực hiện bắt đầu. Khi lệnh kết thúc, thì chương trình sẽ không được nhảy.
- 2 • Lệnh SCJ không thực hiện nhảy chương trình khi lệnh thực hiện được bật. Từ lần quét tiếp theo, lệnh thực hiện nhảy chương trình đến một địa chỉ được chỉ rõ (Số con trỏ) Khi lệnh kết thúc, thì chương trình sẽ không được nhảy.
- Lệnh SCJ được sử dụng khi một số hoạt động phải được thực hiện trước khi nhảy một chương trình.  
Ví dụ: khi đầu ra cần được bật hoặc cài đặt lại trước

[Lược đồ thời gian]

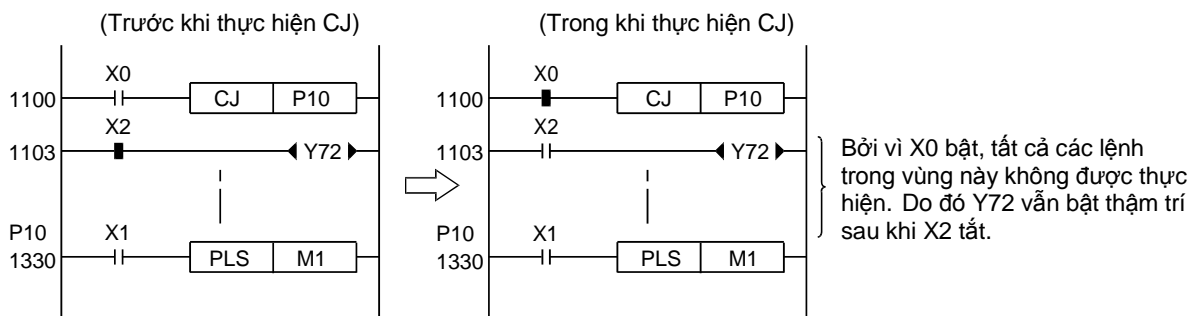


Chú ý

- Số con trở khả dụng cho cả lệnh CJ và SCJ là P0 đến P4095.
- Sử dụng lệnh FEND như bên dưới khi chương trình sử dụng lệnh CJ và SCJ phải được bao gồm trong mỗi khối chương trình. (xem mục 4.7.1 cho lệnh FEND.)



- Trạng thái của các ladder được bỏ qua nếu lệnh CJ vẫn như cũ.



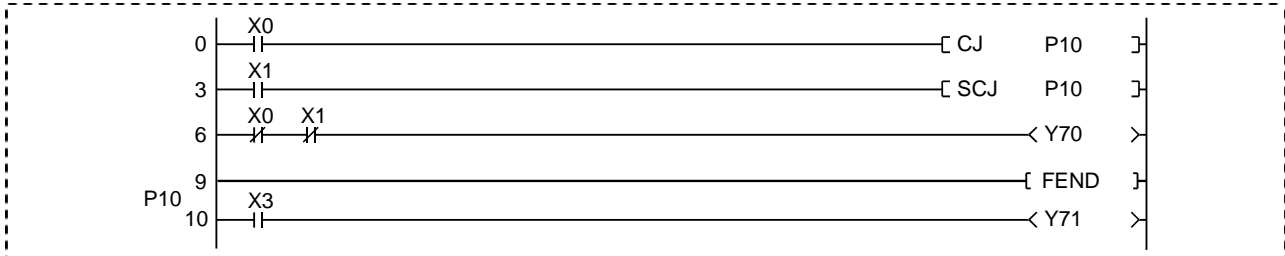
- Sau khi cuộn định thời được bật, nhảy bộ định thời của cuộn cảm mà nó sử dụng các lệnh CJ, SCJ hoặc JMP ngắt phép đo chính xác.

	Biến khả dụng											Số bước cơ bản					
	Biến trong (hệ thống hoặc người sử dụng)		Thanh ghi tệp	MELSECNET/10 (H) Direct Jn\		Mô-đun chức năng thông minh Un\G	Thanh ghi chỉ số	Hãng số			Con trở		Mức				
	Bit	Từ	R	Bit	Từ		Z	K	H	P	I			N			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%; text-align: center;">CJ</td><td style="width: 50%; text-align: center;">P**</td></tr> <tr><td style="width: 50%; text-align: center;">SCJ</td><td style="width: 50%; text-align: center;">P**</td></tr> </table>	CJ	P**	SCJ	P**		P											2
CJ	P**																
SCJ	P**																

Tên dự án	QEX4
Tên chương trình	MAIN

Ví dụ ladder

Tạo một ladder bên dưới với GX Works2 và ghi nó tới CPU. Sau đó kiểm tra sự khác nhau của các lệnh CJ và SCJ.



**Trình tự hoạt động**

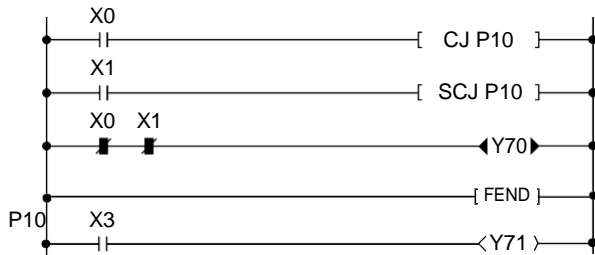
Thủ tục dưới đây tương tự như **Thủ tục hoạt động** trong mục 4.4

- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Ghi dự án đến bộ điều khiển khả trình
- (4) Giám sát một ladder

**Hoạt động thực tế**

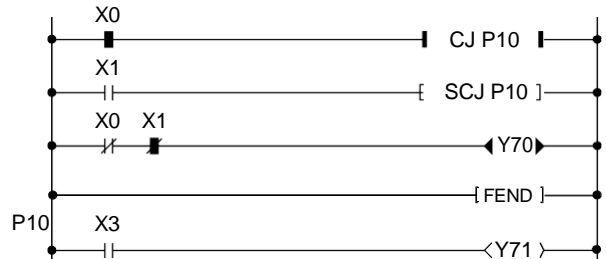
(1) Khi X0 và X1 tắt, lệnh CJ and SCJ không được thực hiện.  
 Ví dụ, Y70 bật.

[Trước khi thực hiện CJ và SCJ]



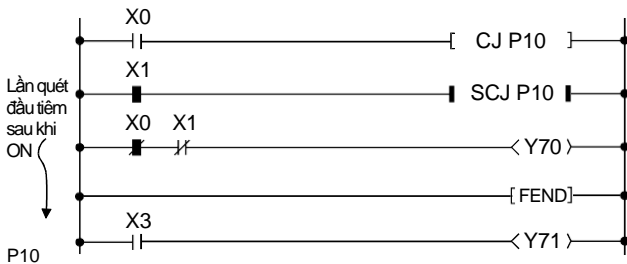
(2) Khi X0 được bật, lệnh CJ được thực hiện và chương trình nhảy đến con trỏ P10. Vì thế, Y70 vẫn bật.

[Trong khi thực hiện CJ] lần quét đầu tiên và các lần quét tiếp theo.

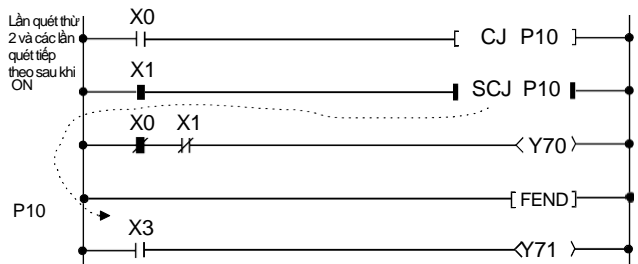


(3) Việc tắt X0 và bật X1 thực hiện lệnh SCJ và nhảy chương trình đến con trỏ P10 từ lần quét thứ 2. Vì thế, Y70 tắt.

[Trong khi thực hiện SCJ] Lần quét đầu tiên.

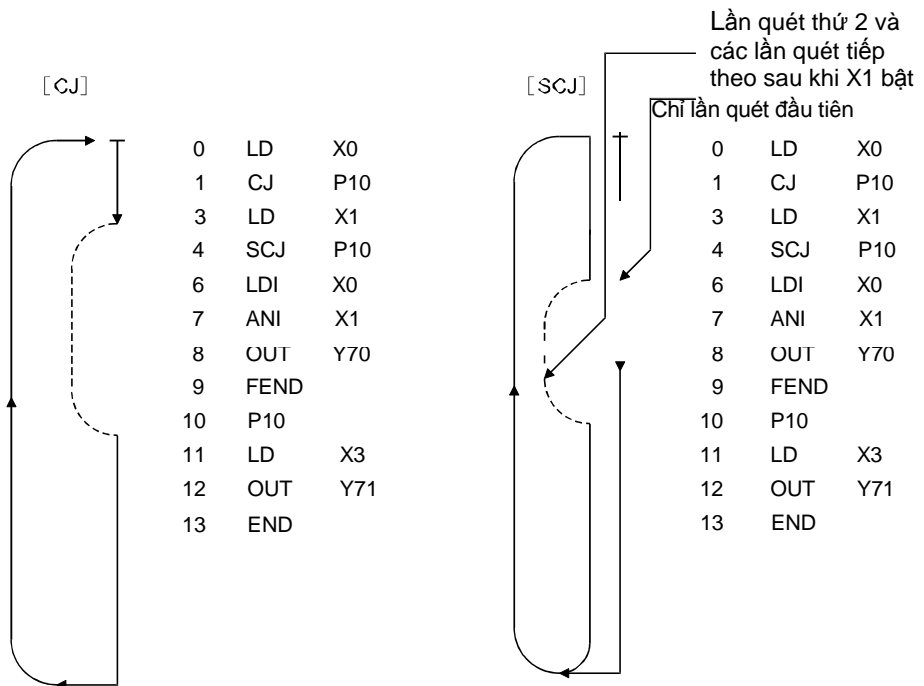


[Trong khi thực hiện SCJ] Lần quét thứ 2 và các lần quét tiếp theo.



(4) Y71 được bật hoặc tắt khi lệnh CJ và SCJ được thực hiện.

- Danh sách dưới đây chứng minh sự khác nhau giữa lệnh CJ và SCJ.

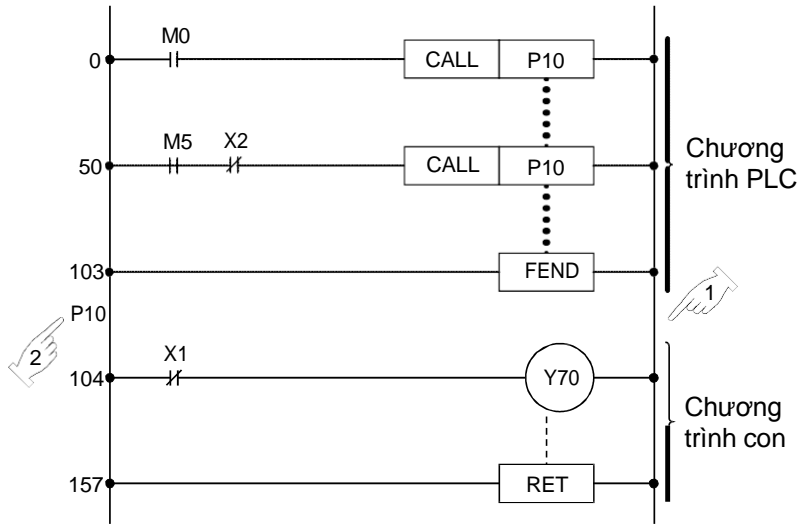


**Bài tập liên hệ**

— Bài tập 4



4.7.3 **CALL(P)** Gọi } Thực hiện một chương trình con  
**RET** Quay }  
 lại



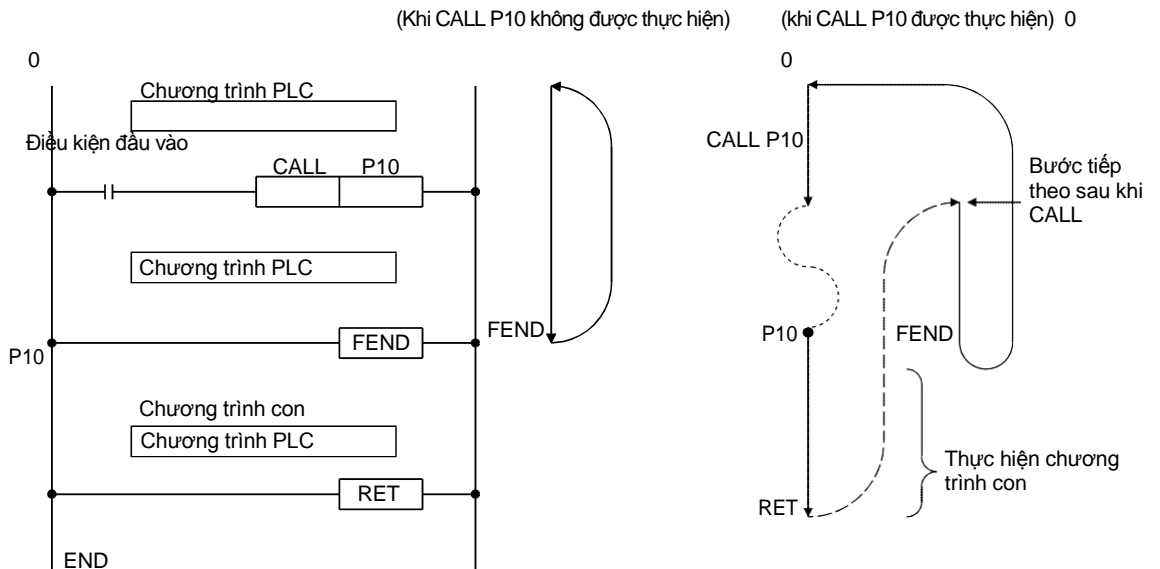
• Chương trình trên là một dạng cơ bản để thực hiện một chương trình con sử dụng lệnh CALL và RET.

Theo kiến trúc này, một lỗi xảy ra và bộ điều khiển chương trình tạm dừng.

1 → • Một chương trình con bao gồm các ladder cho việc thực hiện cùng một dữ liệu nhiều lần trong một chương trình.

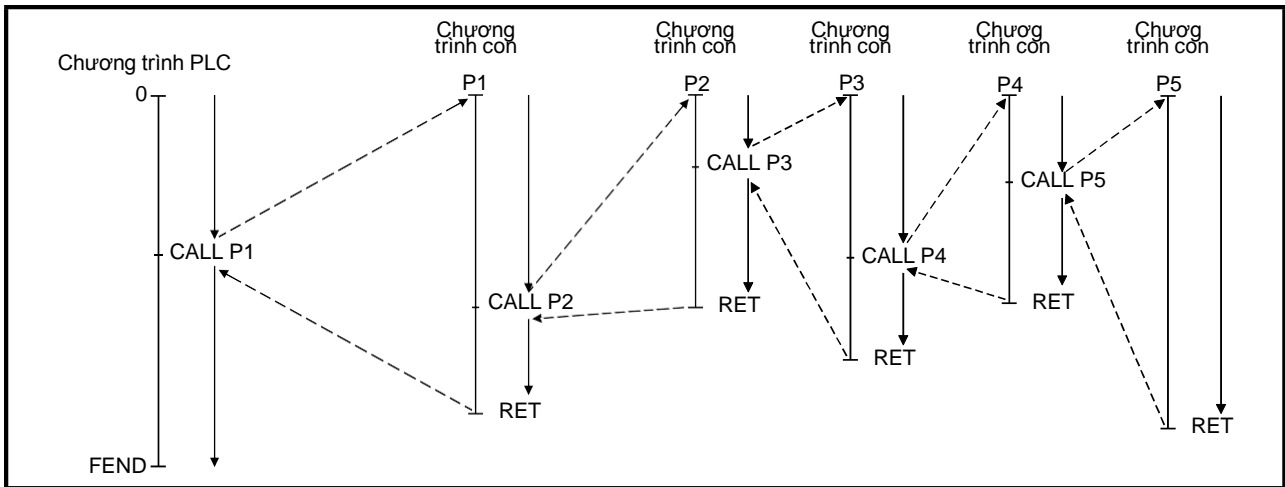
Một chương trình con bắt đầu ở con trở P [ ] và kết thúc bằng lệnh RET.

2 → • Số con trở P từ 0 đến 4095. (Tương tự như số con trở dùng cho lệnh CJ và SCJ)  
 • Một chương trình con được thực hiện như giản đồ bên dưới.

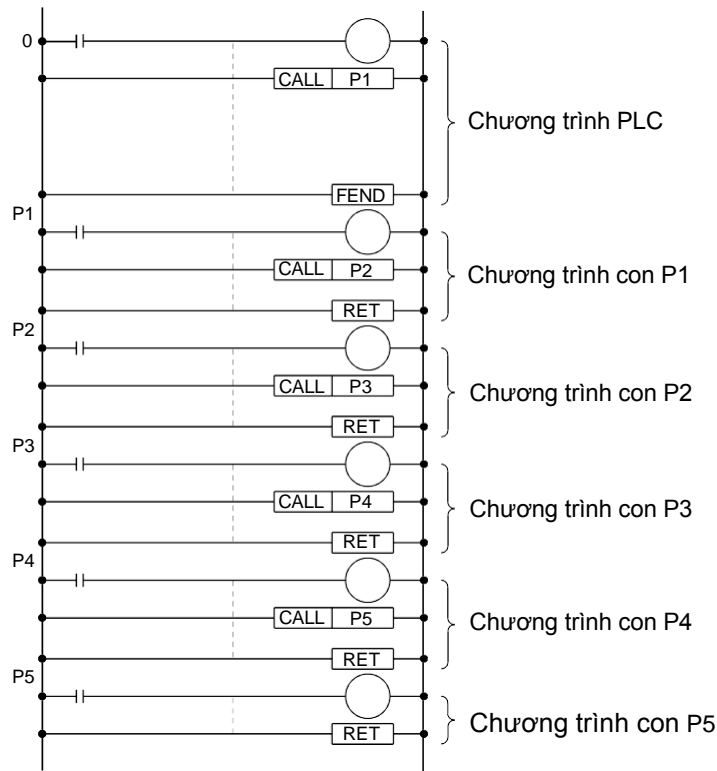


**Lồng**

• Lệnh CALL (P) có thể được lồng đến 16 cấp.



Mạch ladder dưới đây trình bày một chương trình được lồng bên trên.



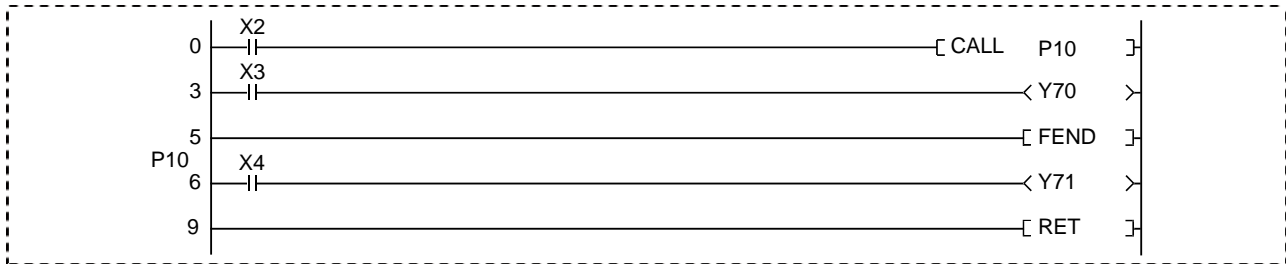
		Biến khả dụng											Số bước cơ bản		
		Biến trong (hệ thống hoặc người sử dụng)		Thanh ghi tệp	MELSECNET/10 (H) Direct Jn\		Mô-đun chức năng thông minh Un\G	Thanh ghi chỉ số	Hàng số		Con trỏ			Mức	
		Bit	Word	R	Bit	Word		Z	K	H	P	I		N	
CALL(P)	P**	P													2
RET															1

Số bước cơ bản của CALL (P) là 2, và của lệnh RET là 1.

Tên dự án	QEX5
Tên chương trình	MAIN

Ví dụ ladder

Tạo một ladder bên dưới với GX Works2 và ghi nó đến CPU. Sau đó kiểm tra các lệnh CALL và RET hoạt động chính xác hay không.



#### Trình tự hoạt động

Thủ tục dưới đây tương tự như **Thủ tục hoạt động** trong mục 4.4

- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Ghi dự án đến bộ điều khiển khả trình
- (4) Giám sát một ladder

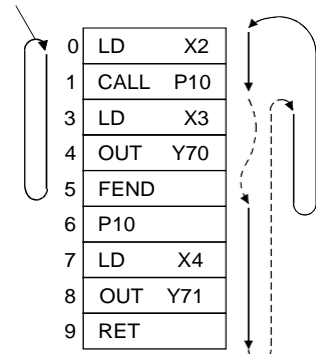
**Hoạt động thực tế**

Xác minh hoạt động của một ladder, mà nó được tạo với GX Works2 và được ghi tới CPU bằng việc giám sát ladder trên màn hình.



- (1) Khi X2 tắt
  - (a) Hoạt động được thực hiện từ 0 đến FEND.
  - (b) Bật hoặc tắt X3, bật hoặc tắt Y70.
  - (c) Bật hoặc tắt X4, dkhôn gthay đổi Y71.
- (2) Khi X2 bật
  - (a) Sau khi chương trình con của con trỏ P10 được thực hiện, hoạt động từ bước 3 đến FEND được thực thi.
  - (b) Bật hoặc tắt X3, bật hoặc tắt Y70.
  - (c) Bật hoặc tắt X4, bật hoặc tắt Y71.

Hoạt động khi X2 tắt



Hoạt động khi X2 bật

**Bài tập liên hệ** — Bài 4

Tên dự án	QTEST1
Tên chương trình	MAIN

#### 4.8 Bài tập

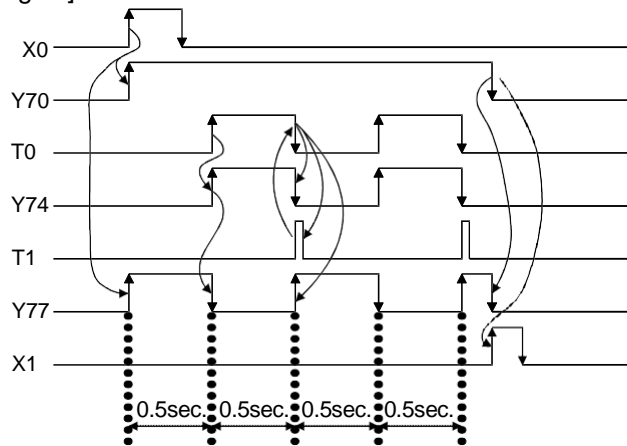
##### 4.8.1 Bài tập 1

#### LD đến NOP

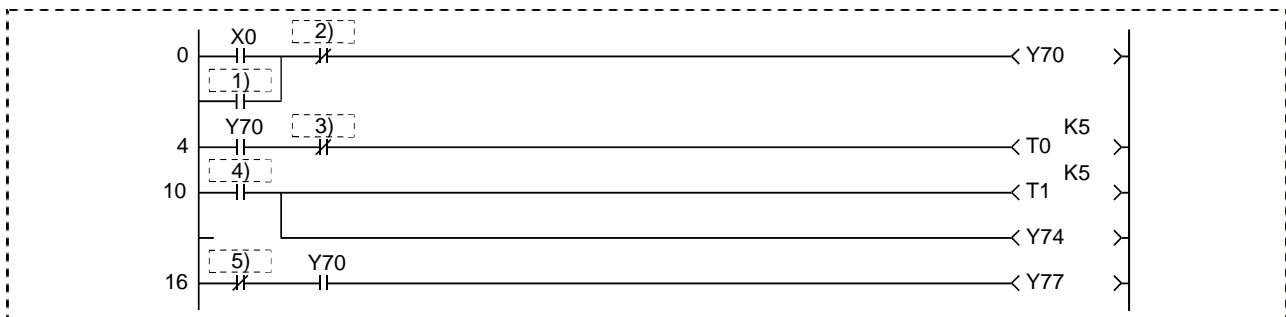
Khi X0 bật, Y70 tự duy trì, Y74 và Y77 lần lượt nhấp nháy trong 0.5 giây.

Khi X1 bật, Y70 tắt và Y74, Y77 cũng dừng.

[Lược đồ thời gian]



Tạo một chương trình bên dưới với GX Works2 và điền vào các ô trống . Sau đó, kiểm tra hoạt động sử dụng máy xác minh.



Tên dự án	QTEST2
Tên chương trình	MAIN

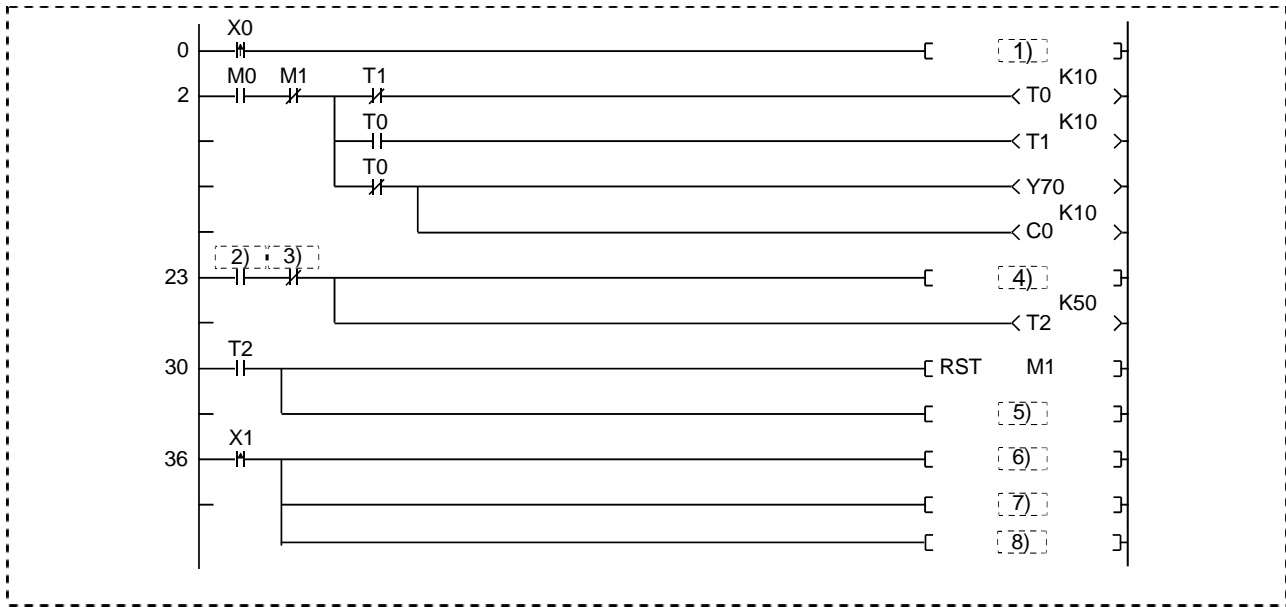
4.8.2 Bài tập 2

SET, RST

Khi X0 được bật, Y70 bắt đầu nhấp nháy khoảng 1 giây và dừng khoảng 5 giây tiếp sau nhấp nháy 10 lần, sau đó nhấp nháy lại.

Y70 có thể dừng nếu bật X1.

Tạo một chương trình bên dưới với GX Works2 và điền vào các ô trống [ ]. Sau đó, sau đó kiểm tra hoạt động bằng máy xác minh.

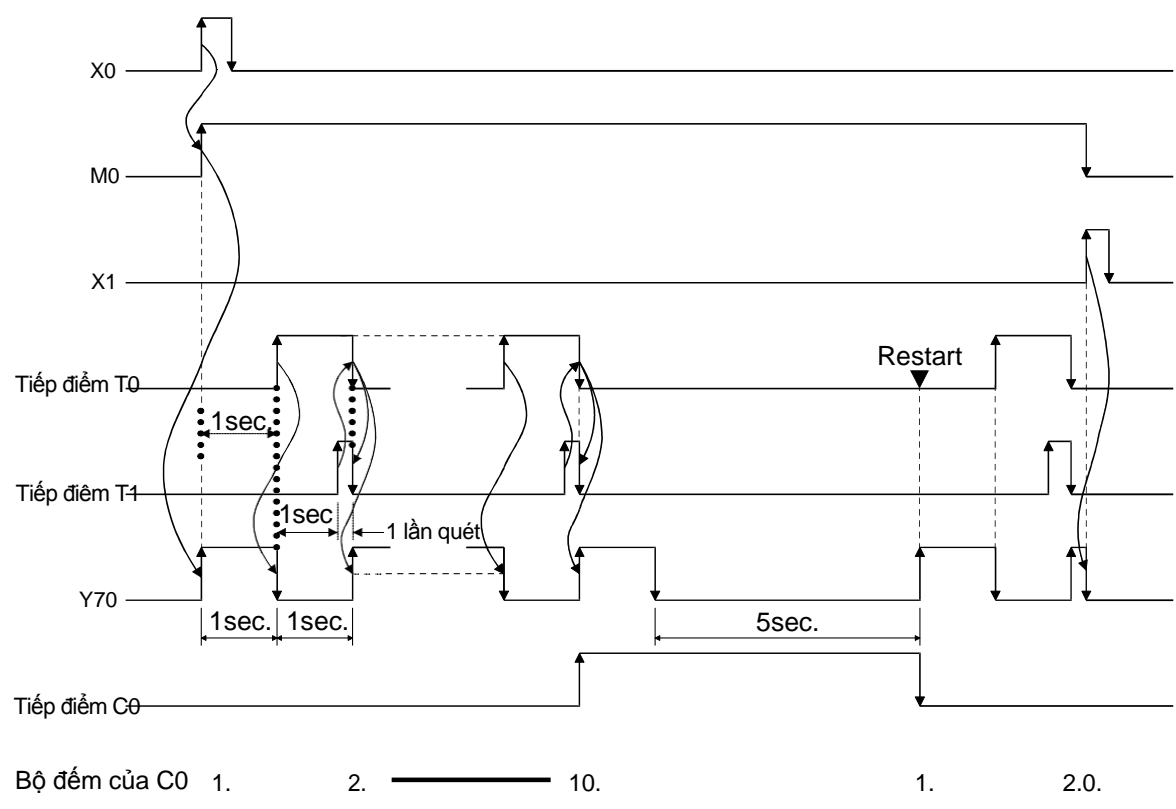


- 1) \_\_\_\_\_
- 2) \_\_\_\_\_
- 3) \_\_\_\_\_
- 4) \_\_\_\_\_

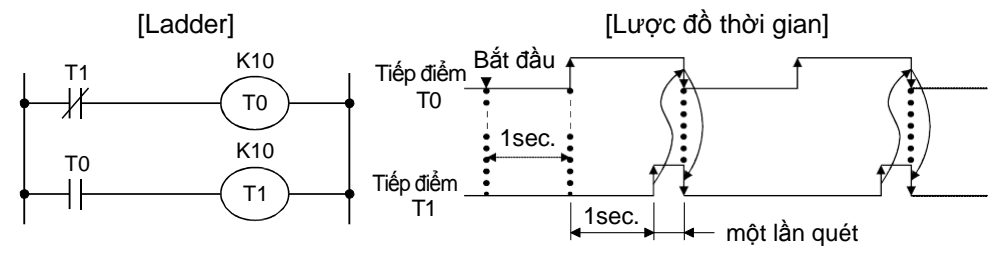
- 5) \_\_\_\_\_
- 6) \_\_\_\_\_
- 7) \_\_\_\_\_
- 8) \_\_\_\_\_

**Gợi ý**

(1) Dưới đây minh họa lược đồ thời gian của chương trình.



(2) Dưới đây là một ladder sáng nhấp nháy cơ bản và lược đồ thời gian.



**Tham khảo**

- Một ladder sáng nhấp nháy có thể được tạo sử dụng một rơ-le đặc biệt mà nó tạo ra xung đồng hồ như bên dưới.

SM413 (2-sec. clock)

Thay thế SM413 (2-sec. clock) bên trái, có thể sử dụng:

- SM409 (0.01-sec. clock)
- SM410 (0.1-sec. clock)
- SM411 (0.2-sec. clock)
- SM412 (1-sec. clock)
- SM414 (2n-sec. clock)
- SM415 (2n-msec. clock)

Xung đồng hồ bắt đầu từ OFF khi bộ điều khiển chương trình cài đặt lại hoặc nguồn được bật.

[Lược đồ thời gian]

The diagram shows a square wave pulse for Y70 with a 1-second period.

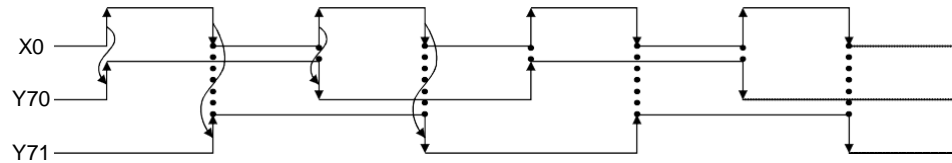
Tên dự án	QTEST3
Tên chương trình	MAIN

4.8.3 Bài tập 3

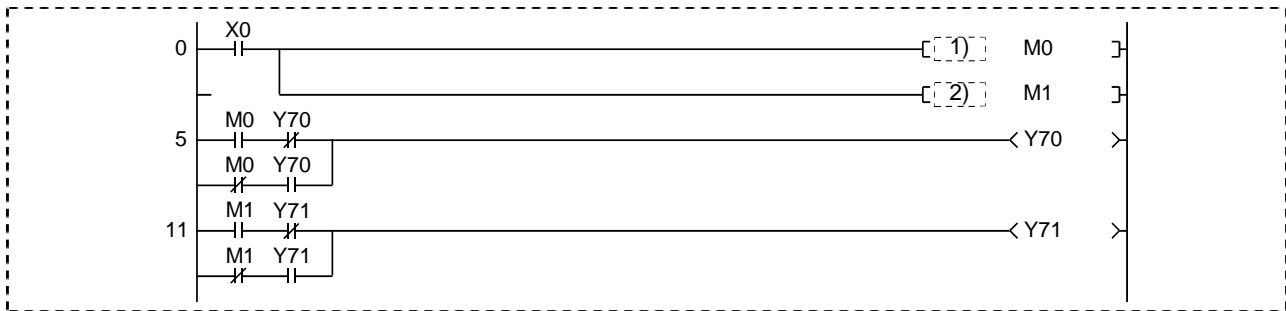
PLS, PLF

Y70 bắt đầu chuyển mạch ON và OFF luân phiên nhau khi X0 được bật và việc tắt X0 kích hoạt Y71 hoạt động tương tự như Y70.

[Lược đồ thời gian]



Tạo chương trình bên dưới với GX Works2 và điền vào khoảng trống [1] và [2]. Sau đó, kiểm tra hoạt động sử dụng máy xác minh.



- 1) \_\_\_\_\_
- 2) \_\_\_\_\_



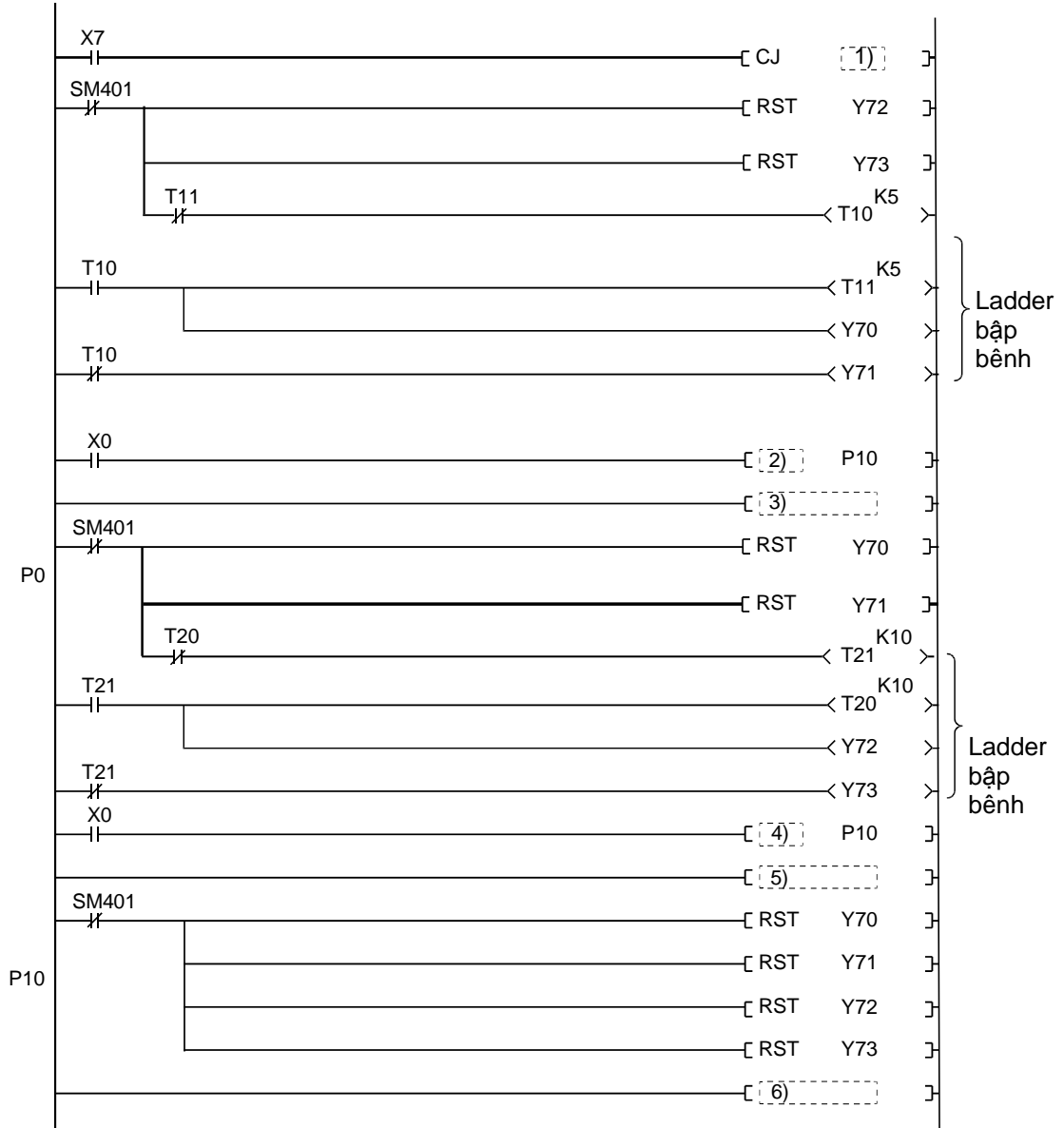
Tên dự án	QTEST4
Tên chương trình	MAIN

4.8.4 Bài tập 4

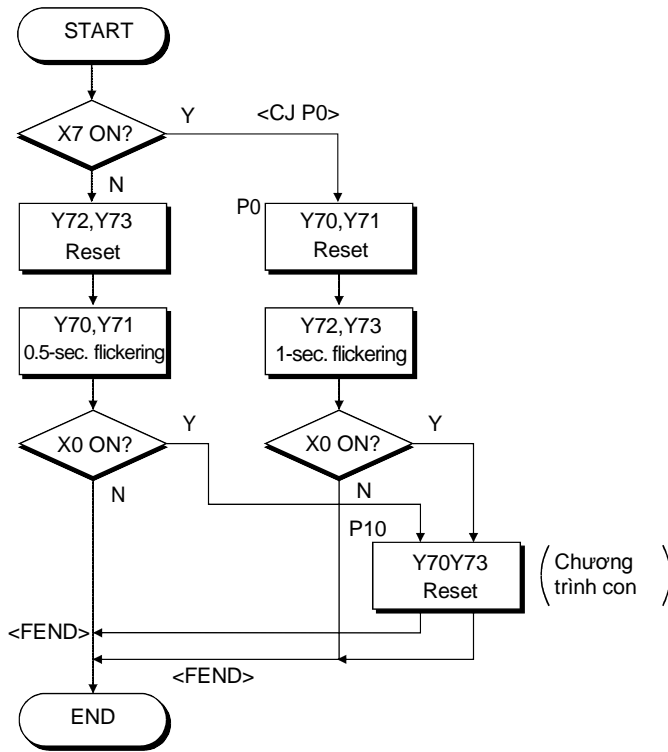
CJ, CALL, RET, FEND

Y70 và Y71 nhấp nháy 0.5 giây luân phiên nhau khi X7 tắt và X7 bật, Y72 và Y73 nhấp nháy luân phiên nhau khoảng 1.0 giây. Việc bật X0 sẽ thiết lập lại chế độ nhấp nháy hiện thời từ Y70 đến Y73.

Điền vào ô trống . Sau đó, kiểm tra hoạt động sử dụng máy xác minh.



Gợi ý



- 1) \_\_\_\_\_
- 2) \_\_\_\_\_
- 3) \_\_\_\_\_
- 4) \_\_\_\_\_
- 5) \_\_\_\_\_
- 6) \_\_\_\_\_

Đáp án cho bài tập chương 4

Bài số.		Đáp án
1	1)	Y70
	2)	X1
	3)	T1
	4)	T0
	5)	Y74
2	1)	SET M0
	2)	C0
	3)	Y70
	4)	SET M1
	5)	RST C0
	6)	RST M0
	7)	RST C0
	8)	RST M1
3	1)	PLS
	2)	PLF
4	1)	P0
	2)	CALL
	3)	FEND
	4)	CALL
	5)	FEND
	6)	RET

MEMO

## Chương 5    Lệnh cơ bản -Phần 2-

### 5.1   Kí hiệu của giá trị (dữ liệu)

CPU điều khiển khả trình biến đổi tất cả tín hiệu đầu vào thành tín hiệu ON hoặc OFF (logic lần lượt là 1 hoặc 0) để lưu trữ và xử lý chúng. Vì vậy, khối điều khiển khả trình thực thi hoạt động số sử dụng giá trị số được lưu với dạng logic 1 hoặc 0 (số nhị phân = BIN).

Trong cuộc sống hàng ngày, một số thập phân được sử dụng phổ biến và trong hệ thống đơn giản. Vì vậy, biến đổi thập phân sang nhị phân hoặc biến đổi ngược là cần thiết khi giá trị được viết hoặc đọc (giám sát) đến hoặc từ khối điều khiển khả trình. Hệ thống lập trình và một số lệnh có chức năng chuyển đổi thập phân sang nhị phân và từ nhị phân sang thập phân.

Phần này diễn giải làm thế nào để biểu thị giá trị (dữ liệu) dạng thập phân, nhị phân, thập lục phân và kí hiệu BCD, và làm thế nào để biến đổi chúng.

#### Thập phân

- Một hệ thập phân bao gồm 10 kí tự: 0, 1, 2, 3, 4, 5, 6, 7, 8, và 9 được sắp xếp theo thứ tự và giá trị (độ lớn).  
Sau khi một con số đến 9, số đó được đặt lại về 0 và con số bên trái của số đó được tăng thêm 1.
- Hình dưới diễn tả một số hệ thập phân (trong trường hợp này là 153) được biểu diễn như thế nào.

$$\begin{aligned} 153 &= 100+50+3 \\ &= 1 \times 100 + 5 \times 10 + 3 \times 1 \\ &= 1 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 \end{aligned}$$

Kí tự thập phân (0 đến 9)

"Độ lớn của con số"

n    : Con số (0, 1, 2...)

10   : Thập phân

- Trong khối điều khiển khả trình MELSEC-Q series, kí tự "K" được dùng để diễn tả giá trị dạng thập phân.

**Hệ nhị phân (BIN)**

- Hệ nhị phân bao gồm hai kí tự: 0 và 1 được sắp xếp theo thứ tự và giá trị (độ lớn). Sau khi một con số đến 1, số đó trở về 0 và con số bên cạnh (bên trái) được tăng thêm 1. Hai con số 0 và 1 còn được gọi là bit.

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

- Ví dụ bên dưới diễn giải làm thế nào để biến đổi một số nhị phân thành thập phân.

"10011101"

Biểu đồ dưới chỉ ra số nhị phân với mũ hai .

7	6	5	4	3	2	1	0	← Số bit
1	0	0	1	1	1	0	1	← Nhị phân
2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	← Số cơ bản với số mũ tương ứng
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	} = độ lớn bit
128	64	32	16	8	4	2	1	← ("nhị phân")

Số nhị phân được chỉ ra như sau.

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

Số nhị phân có thể được tính toán bằng cách thêm vào mỗi bit có mã bằng 1.

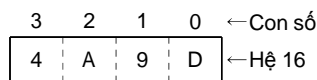
**Hệ thập lục phân**

- Hệ thập lục phân gồm 16 kí tự: 0 đến 9 và A đến F được sắp xếp theo thứ tự và giá trị (độ lớn). Sau khi một con số đến F, số đó được trở về 0 và con số bên cạnh (bên trái) tăng thêm một.

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

1 9 1 0 1	4 A 9 D	0 1 0 0   1 0 1 0   1 0 0 1   1 1 0 1
	↑ ↑ ↑ ↑	↑ ↑ ↑ ↑



"Số mũ"  
n: con số  
**16:** Thập lục phân

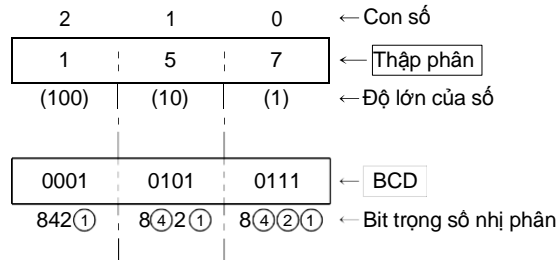
$$\begin{aligned}
 &= (4) \times 16^3 + (A) \times 16^2 + (9) \times 16^1 + (D) \times 16^0 \\
 &= 4 \times 4096 + 10 \times 256 + 9 \times 16 + 13 \times 1 \\
 &= 19101
 \end{aligned}$$

- Bốn bit của một số nhị phân bằng với một số của hệ thập lục phân.
- Trong khối điều khiển khả trình MELSEC-Q series, kí tự "H" dùng để chỉ một số hệ thập lục phân.
- Hệ thập lục phân thường sử dụng để thể hiện biến sau.
  - Đầu vào và đầu ra (X, Y)
  - Chức năng đầu vào và đầu ra (FX, FY)
  - Role liên kết (B)
  - Thanh ghi liên kết (W)
  - Role liên kết đặc biệt (SB)
  - Thanh ghi liên kết đặc biệt (SW)
  - Biến liên kết trực tiếp (Jn\X, Jn\Y, Jn\B, Jn\SB, Jn\W, Jn\SW)

**Hệ thập phân mã hóa nhị phân (BCD)**

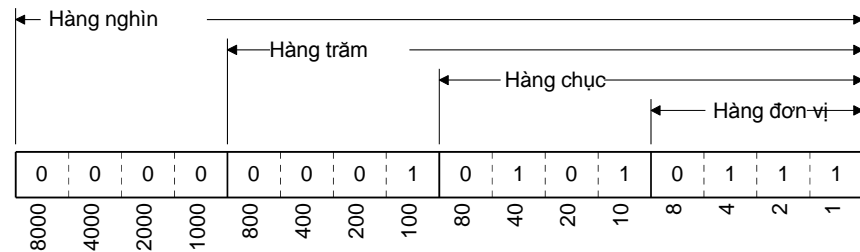
- Số hệ thập phân mã hóa nhị phân là "một hệ số sử dụng số nhị phân để biểu diễn một số thập phân".

Một số thập phân 157, để dễ hiểu hơn, được biểu diễn dưới đây.

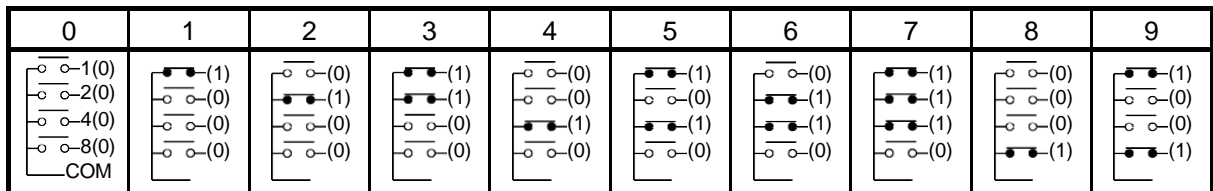


- Trong BCD, số thập phân từ 0 đến 9999 (số có 4 chữ số lớn nhất) có thể biểu diễn với 16 bit.

Biểu đồ dưới chỉ ra độ lớn bit của BCD.



- BCD được sử dụng cho những tín hiệu sau.
  - Tín hiệu đầu ra của công tắc số
  - Tín hiệu của hiển thị bảy thành phần (hiển thị số)



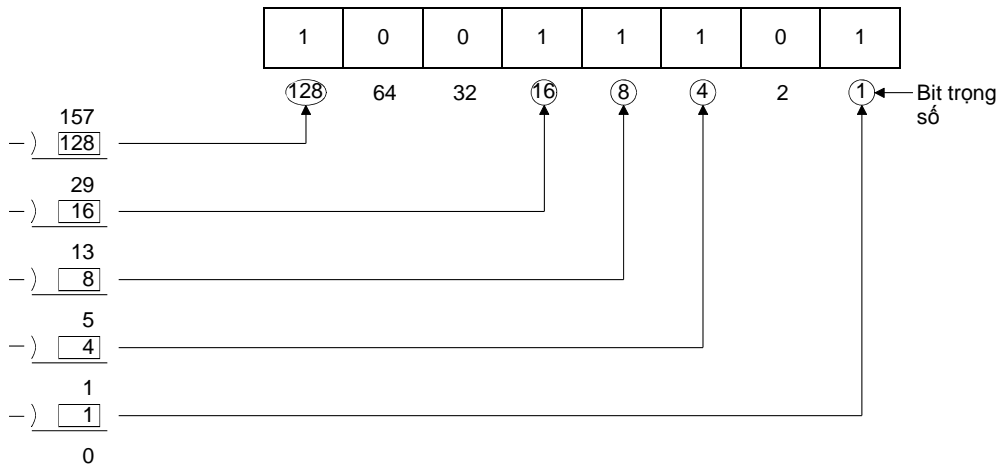
Mã công tắc số BCD



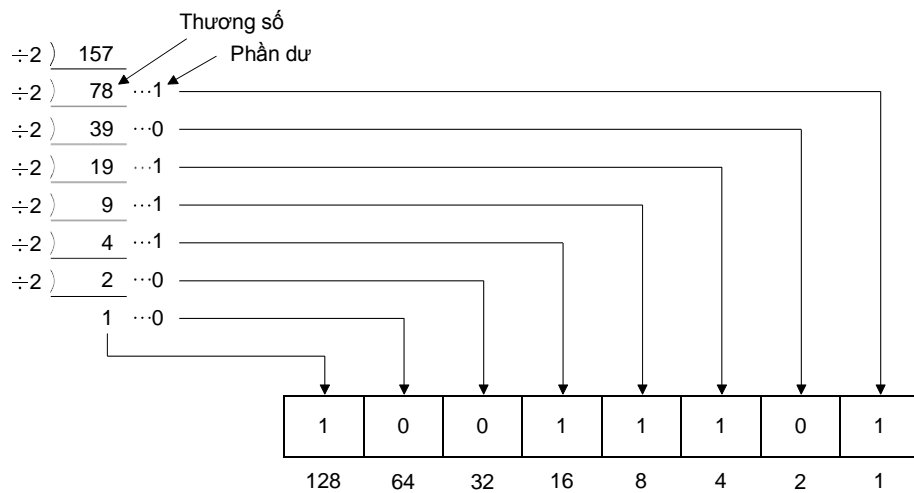
Làm thế nào để biến đổi số thập phân sang nhị phân

Trong ví dụ dưới, một số thập phân là 157 được biến đổi thành số nhị phân.

1)

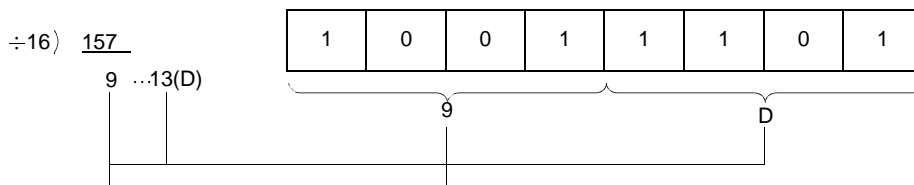


2)



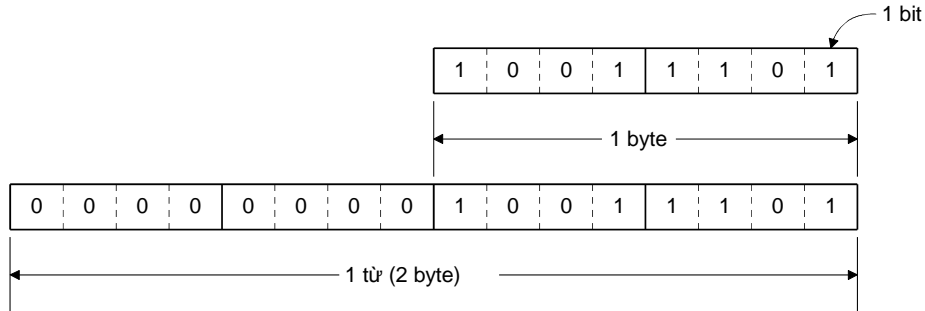
Làm thế nào để biến đổi một số thập phân thành số thập lục phân

Ở ví dụ dưới đây, một số thập phân là 157 được biến đổi thành một số thập lục phân 1)



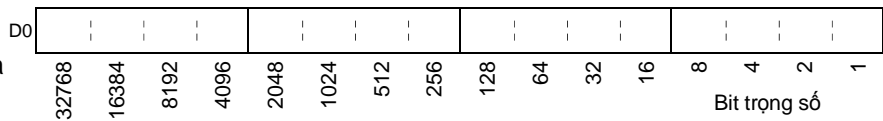
Giá trị số sử dụng bởi khối điều khiển khả trình MELSEC-Q series

- Thông thường, 8 bit được gọi là 1 byte, và 16 bits (2 byte) được gọi là 1 từ



- Thanh ghi của mỗi biến từ trong khối điều khiển khả trình MELSEC-Q series bao gồm 16 bit.

- Thanh ghi D
- Giá trị hiện tại của bộ định thời T
- Giá trị hiện tại của bộ đếm C
- Thanh ghi tệp R
- Thanh ghi liên kết W



- Hai khoảng dưới đây có thể được xử lý trong 16 bit (1 từ).

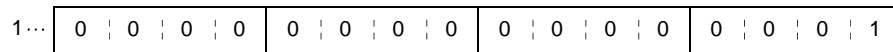
- 1) 0 đến 65535
- 2) -32768 đến +32767

- Khoảng 2) là hoàn toàn khả thi với khối điều khiển khả trình MELSEC-Q.

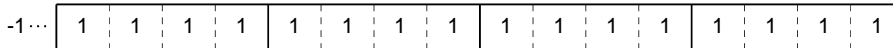
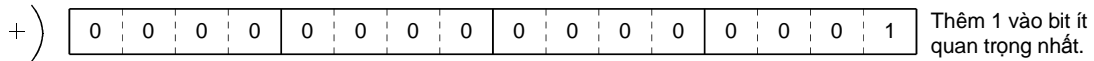
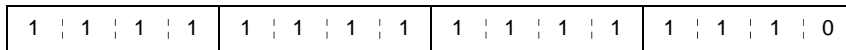
Số âm nhận hai phần bù ngược với số dương (1 to +32767).

- Trong phần hai phần bù, mỗi bit nhị phân bị nghịch đảo, và 1 được thêm vào bit ít quan trọng nhất. Ví dụ)

Làm thế nào để tính hai phần bù ngược với 1



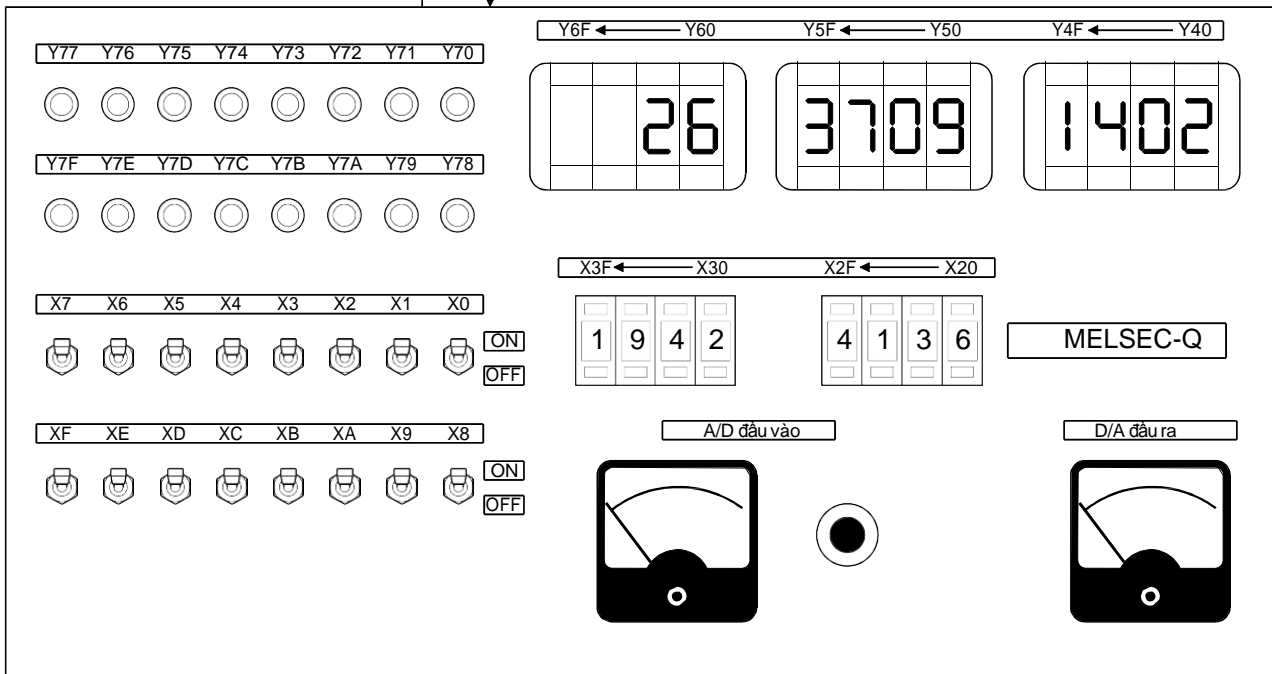
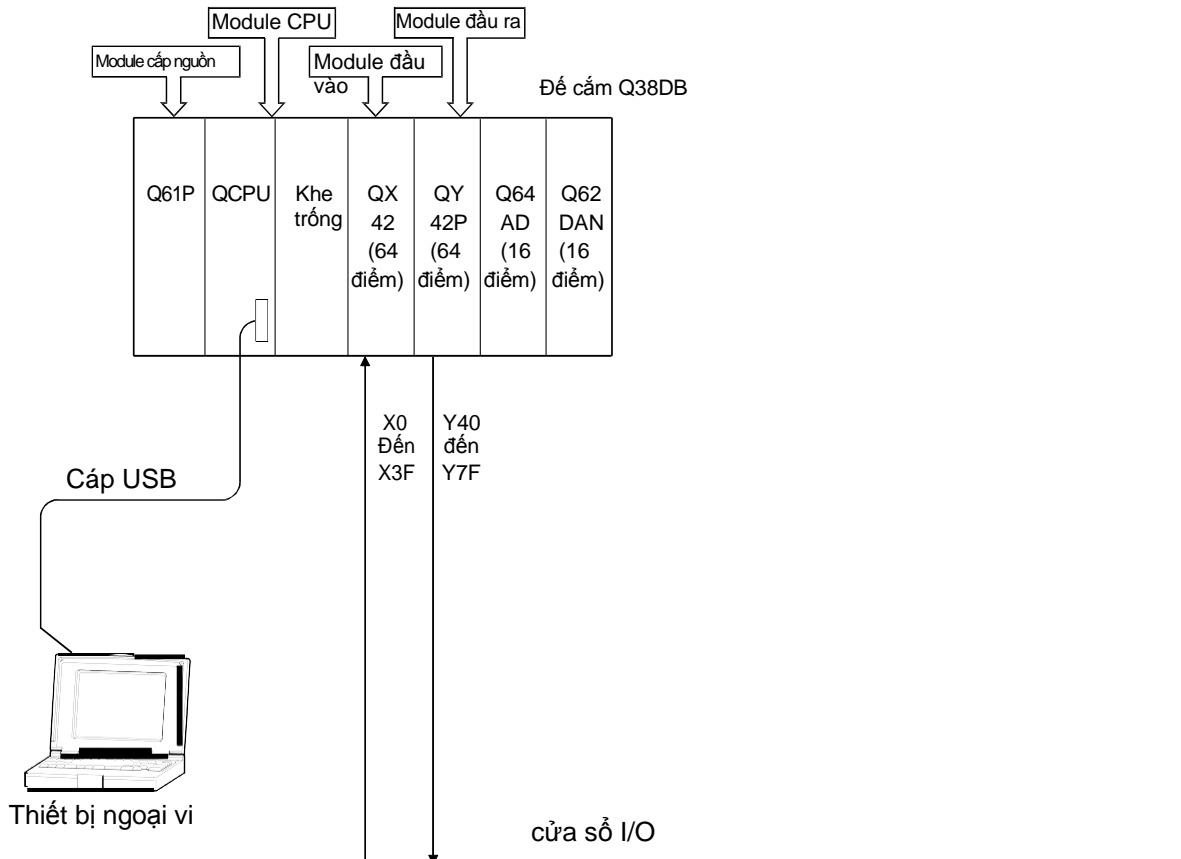
↓  
Nghịch đảo tất cả các bit



↑  
Bit quan trọng nhất là 1 nghĩa là số âm.  
Bit quan trọng nhất tương ứng với dấu của một số nhị phân.

BCD (hệ số BCD)	BIN (nhị phân)	K (thập phân)	H (hệ 16)
00000000 00000000	00000000 00000000	0	0000
00000000 00000001	00000000 00000001	1	0001
00000000 00000010	00000000 00000010	2	0002
00000000 00000011	00000000 00000011	3	0003
00000000 00000100	00000000 00000100	4	0004
00000000 00000101	00000000 00000101	5	0005
00000000 00000110	00000000 00000110	6	0006
00000000 00000111	00000000 00000111	7	0007
00000000 00001000	00000000 00001000	8	0008
00000000 00001001	00000000 00001001	9	0009
00000000 00010000	00000000 00001010	10	000A
00000000 00010001	00000000 00001011	11	000B
00000000 00010010	00000000 00001100	12	000C
00000000 00010011	00000000 00001101	13	000D
00000000 00010100	00000000 00001110	14	000E
00000000 00010101	00000000 00001111	15	000F
00000000 00010110	00000000 00010000	16	0010
00000000 00010111	00000000 00010001	17	0011
00000000 00011000	00000000 00010010	18	0012
00000000 00011001	00000000 00010011	19	0013
00000000 00100000	00000000 00010100	20	0014
00000000 00100001	00000000 00010101	21	0015
00000000 00100010	00000000 00010110	22	0016
00000000 00100011	00000000 00010111	23	0017
00000001 00000000	00000000 01100100	100	0064
00000001 00100111	00000000 01111111	127	007F
00000010 01010101	00000000 11111111	255	00FF
00010000 00000000	00000011 11101000	1000	03E8
00100000 01000111	00000111 11111111	2047	07FF
01000000 10010101	00001111 11111111	4095	0FFF
	00100111 00010000	10000	2710
	01111111 11111111	32767	7FFF
	11111111 11111111	-1	FFFF
	11111111 11111110	-2	FFFE
	10000000 00000000	-32768	8000

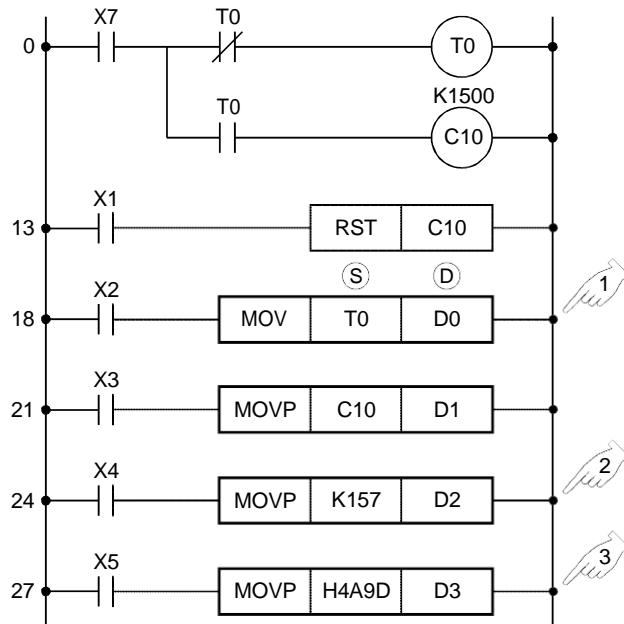
Cấu hình hệ thống và số I/O của bộ DEMO




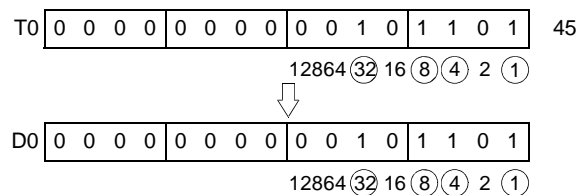
## 5.2 Lệnh truyền


Tên dự án	QB-11
Tên chương	MAIN

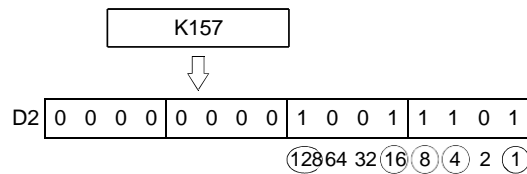
### 5.2.1 MOV (P) Truyền dữ liệu 16 bit



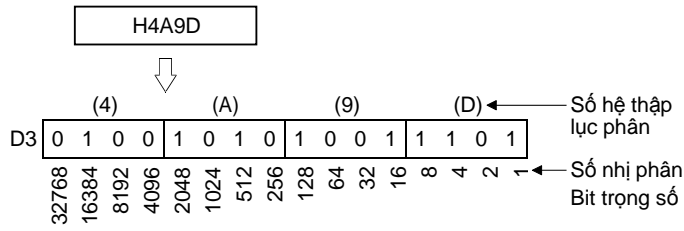
- 
 Khi điều kiện đầu vào bật lên, giá trị hiện tại của bộ định thời T0 được truyền tới thanh ghi dữ liệu D0.  
 (S)... Nguồn, (D)... Đích
- Giá trị hiện tại của T0 được lưu trong thanh ghi dưới dạng nhị phân (mã BIN). Và giá trị đó được truyền vào thanh ghi dữ liệu D0 ở dạng nhị phân (Mã này không bị biến đổi khi truyền.)



- 
 Khi điều kiện đầu vào bật lên, số thập phân 157 được truyền tới thanh ghi dữ liệu D2. Và giá trị được lưu trong trên thanh ghi dưới dạng nhị phân. Số thập phân (K) tự động được biến đổi thành nhị phân, và sau đó truyền đi.

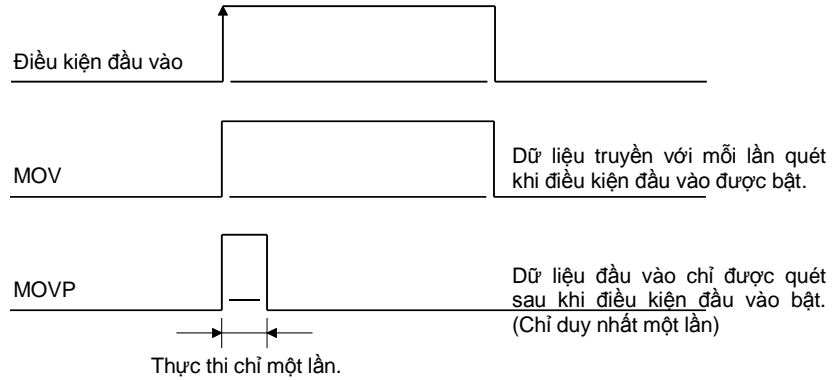


- 3 • Khi điều kiện đầu vào bật lên, số thập lục phân 4A9D được truyền vào thanh ghi D3.

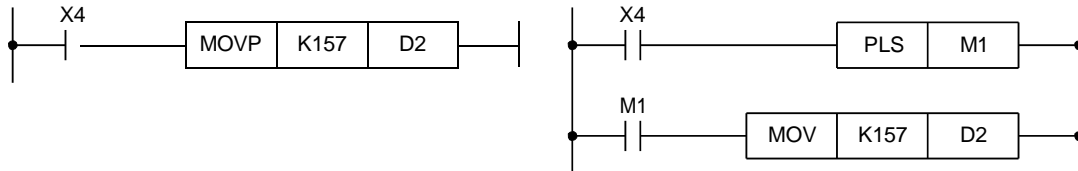


**Sự khác nhau giữa MOV và MOV P**

P của MOV P đại diện cho một xung.



- Lệnh **MOV** dùng để đọc dữ liệu đang thay đổi bất kì lúc nào.
- Lệnh **MOV P** dùng để truyền dữ liệu ngay lập tức như là cài đặt dữ liệu hay đọc dữ liệu tại một điểm lỗi.
- Cả hai chương trình ladder dưới đây có chức năng tương tự nhau .



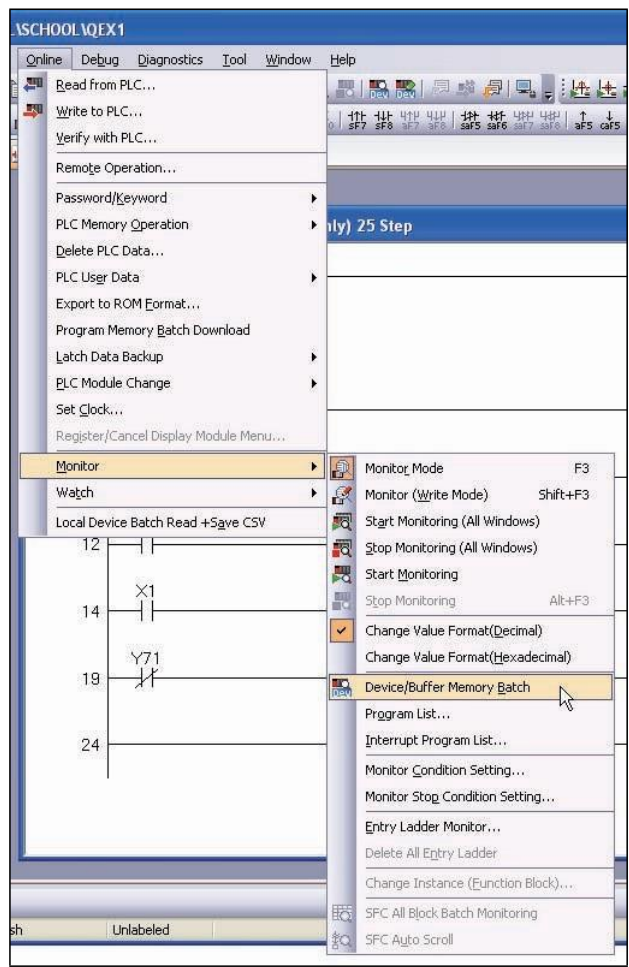
	Biến khả dụng											Số bước cơ bản				
	Biến trong (hệ thống hoặc người dùng)		Thanh ghi tệp	MELSECNET/10 (H) Direct Jn\		Module chức năng thông minh	Thanh ghi chỉ số	Hàng số			Mức					
	Bit	Từ	R	Bit	Từ		Z	K	H	P			I	N		
MOV	S	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	*
	D	☒	☒	☒	☒	☒	☒	☒								

\*: Số bước thay đổi dựa vào số biến sử dụng.

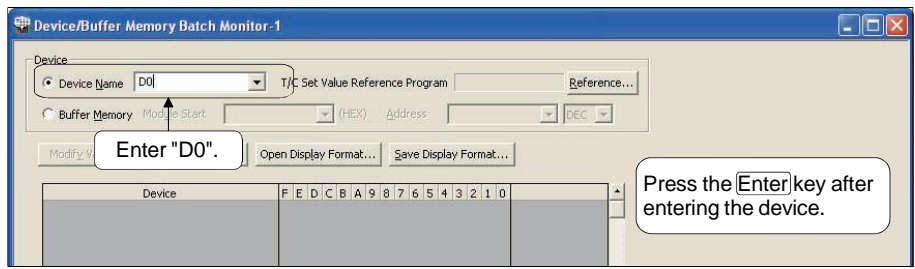
Kiểm tra

The CPU đang chạy.  
Đầu vào X2, X3, X4, X5, và X7 được bật.

- Giám sát nội dung thanh ghi D0 đến D3.
  - Sau khi viết dữ liệu đến khối điều khiển khả trình, click [Online] → [Monitor] → [Device/Buffer Memory Batch].  
Hộp thoại Device/Buffer Memory Batch Monitor hiện ra.



- Nhập "D0" vào cột Device Name của hộp thoại Device/Buffer Memory Batch Monitor và ấn  .



Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0		
D0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	40
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	157
D3	0	1	0	0	1	0	1	0	1	0	0	1	1	1	0	1		19101
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Giá trị hiện tại của bộ định thời và bộ đếm được giám sát (Nó đang thay đổi.)

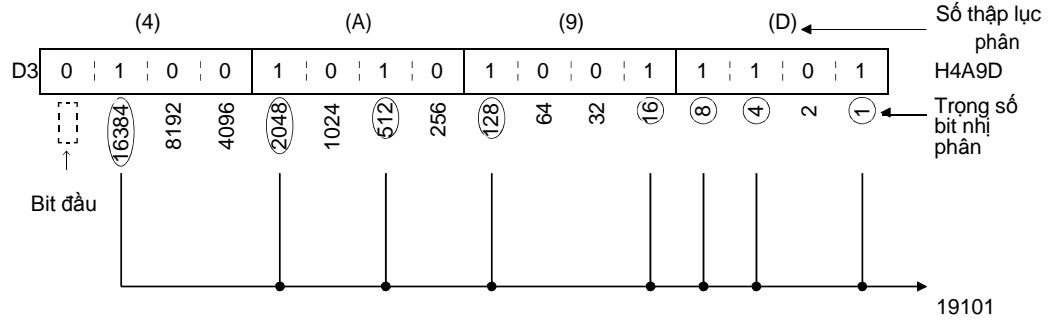
Chỉ ra số thập phân 157 (K157) được lưu.

Đây là một số thập phân tương đương với một thập lục phân 4A9D.

Indicates the word devices in the on/off of the bit units.

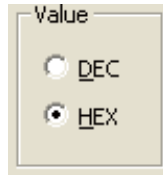
0 : OFF (0 in binary)

1 : ON (1 in binary)





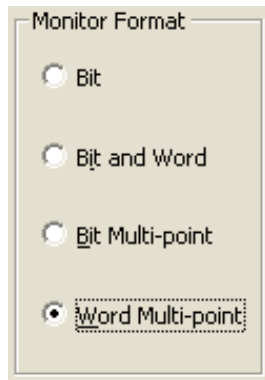
- Click nút Display Format .
- Thay đổi hiển thị.
- Chọn "HEX" cho trình theo dõi loạt biến.



[Màn hình Device/Buffer Memory Batch Monitor]

Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0023
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D2	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	009D
D3	0	1	0	0	1	0	1	0	1	0	0	1	1	1	0	1	4A9D
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

- Thay đổi hiển thị của giá trị số trong màn hình với kí hiệu nhị phân  
Chọn "Word Multi-point" trong Monitor Format cho trình theo dõi loạt biến.



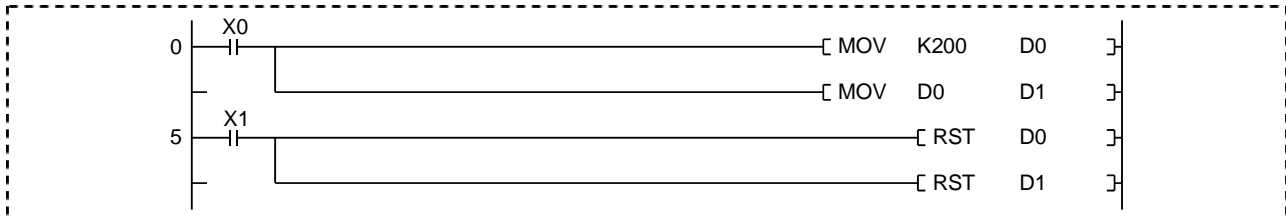
[Màn hình Device/Buffer Memory Batch Monitor]

Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	000D	0000	009D	4A9D	0000	0000	0000	0000
D8	0000	0000	0000	0000	0000	0000	0000	0000

Tên dự án	QEX7
Tên chương	MAIN

Ví dụ ladder

Tạo ladder sau với GX Works2 và viết nó lên CPU của bộ DEMO. Sau đó kiểm tra hoạt động lệnh MOV.



Quy trình hoạt động

Quy trình hoạt động dưới đây tương tự như Quy trình hoạt động trong phần 4.4.

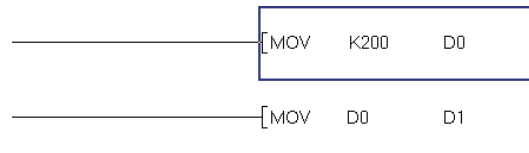
- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Viết dự án lên khối điều khiển khả trình.
- (4) Giám sát ladder.

- Làm thế nào để thay đổi lệnh truyền

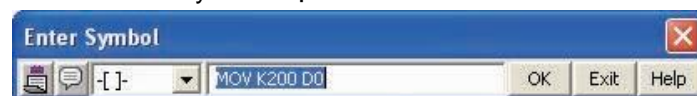
Để thay đổi lệnh truyền, làm theo hướng dẫn dưới đây.

Ví dụ: Thay đổi dữ liệu truyền K200 của [MOV K200 D0] thành K100

- 1) Click đúp vào lệnh để chỉnh sửa.



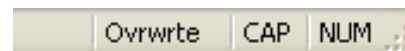
- 2) Cửa sổ Enter Symbol hiện ra



- 3) Viết "1" lên "2" của "MOV K200 D0".

- 4) Click nút  trên cửa sổ Enter Symbol.

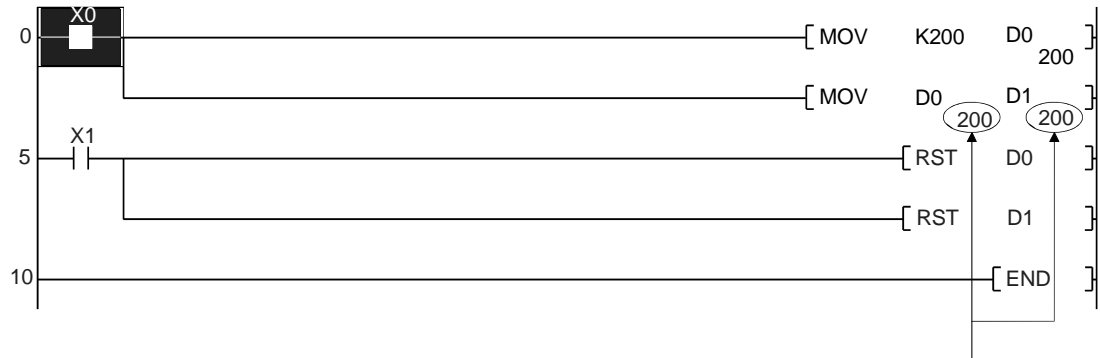
Dữ liệu ở [ ] có thể được chỉnh sửa với phương pháp Khi "Insert" hiện ra, ấn vào nút  để thay đổi nó thành trước khi chỉnh sửa.



- 5) Sau khi kết thúc chỉnh sửa, click [Compile] → [Build].

Luyện tập

Kiểm tra "200" hiển thị dưới D0 và D1 trên màn hình máy tính khi X0 của cửa sổ điều khiển của thiết bị Demo bật lên.



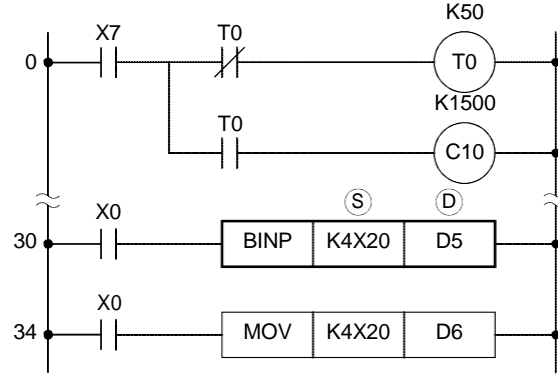
Khi X0 được bật, giá trị hiện tại của D0 và D1 trở thành 200.

Bài tập liên quan ---- Bài tập 5

Tên dự án	QB-12
Tên chương trình	MAIN

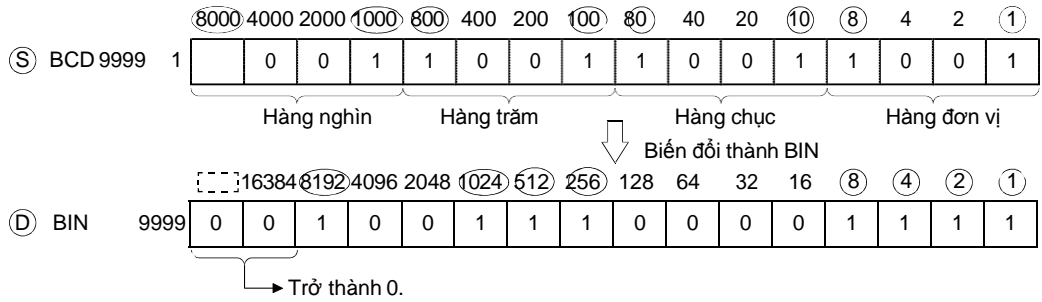
5.2.2 BIN (P) BCD → Lệnh biến đổi dữ liệu BIN

Hoạt động để đọc và viết dữ liệu sau bước 35

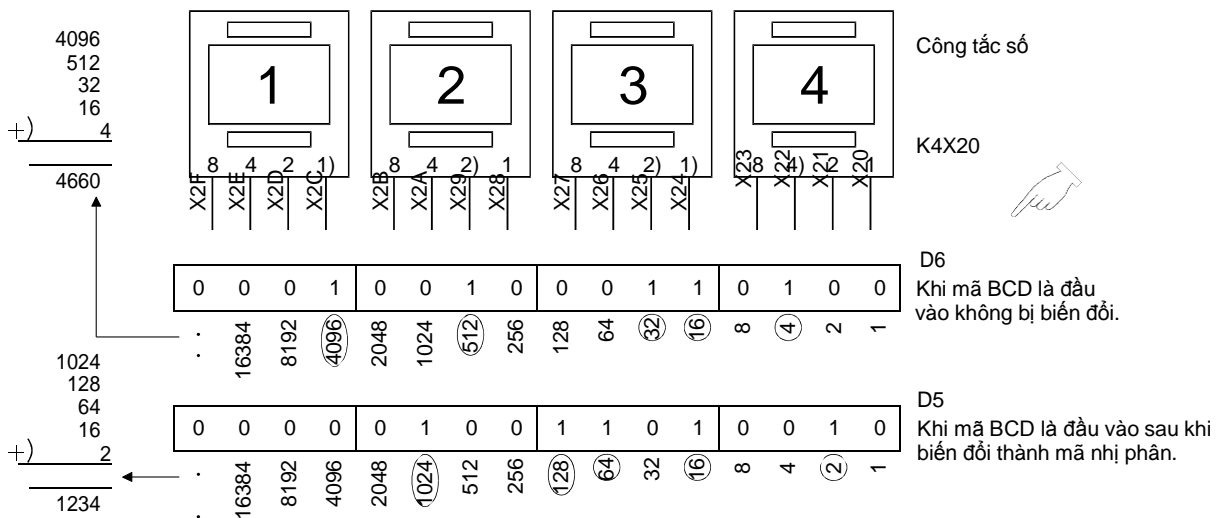


Kiểm tra sự khác nhau của lệnh BIN.

- Khi điều kiện đầu vào được bật, dữ liệu biến được xác định trong S được nhận điện là mã BCD, biến đổi thành nhị phân (mã BIN), và truyền đến biến xác định trong D.

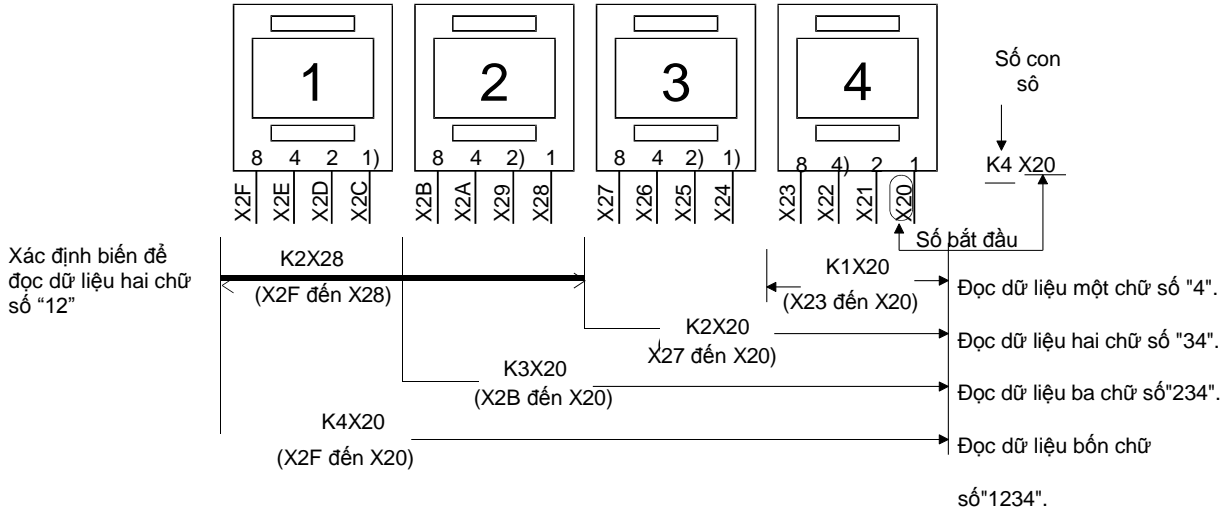


- Công tắc số nguyên gốc gạo mã BCD. Đó đó, lệnh BIN là cần thiết để viết dữ liệu từ công tắc số đến khối điều khiển khả trình.



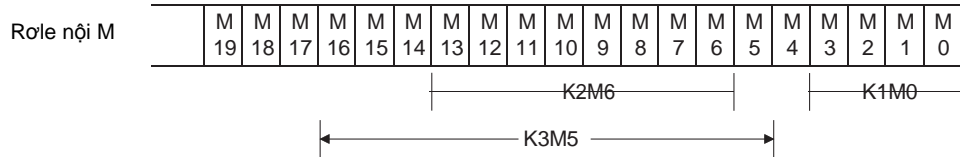
**K4X20**

- Biến từ D (thanh ghi dữ liệu), T (giá trị hiện tại bộ định thời), and C (giá trị hiện tại bộ đếm) gồm 16 bit (1 từ), và dữ liệu cơ bản được truyền giữa những đơn vị trong một biến.
- Thu thập 16 biến bit (như là X, Y, và M) nghĩa là xử lý biến từ. Số hiệu biến được cấp cho biến từ phải theo thứ tự liên tiếp.
- Với biến bit, dữ liệu được xử lý theo bốn đơn vị.



Khi mà biến bốn bit vẫn theo thứ tự liên tiếp, bất kì biến bit nào có thể xác định là cái đầu tiên.

- Biến bit khác có thể được xử lý theo cách tương tự như trên.

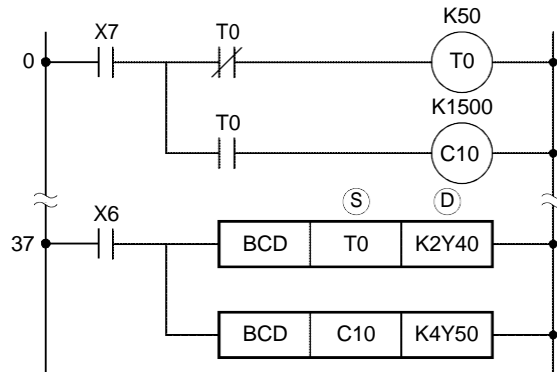


\* Một chương trình mẫu sử dụng công tắc số để nhập dữ liệu được cung cấp trong trang App. - 46.

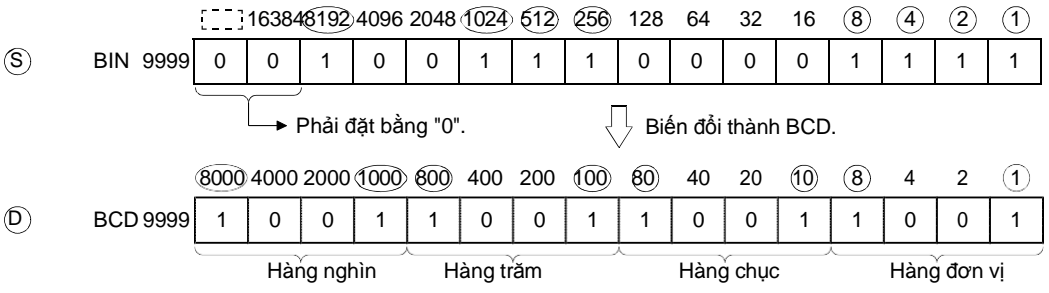
		Biến có thể áp dụng được											Con số	Số bước cơ bản									
		Biến trong (hệ thống hoặc người dùng)		Thanh ghi tệp	MELSECNET/10 (H) Direct Jn\		Module chức năng thông minh Un\G	Thanh ghi chỉ số	Hàng số		Con trỏ				Mức								
		Bit	Từ	R	Bit	Từ		Z	K	H	P	I				N							
MOV	S D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	K1 to K4	3

Tên dự án	QB-13
Tên chương	MAIN

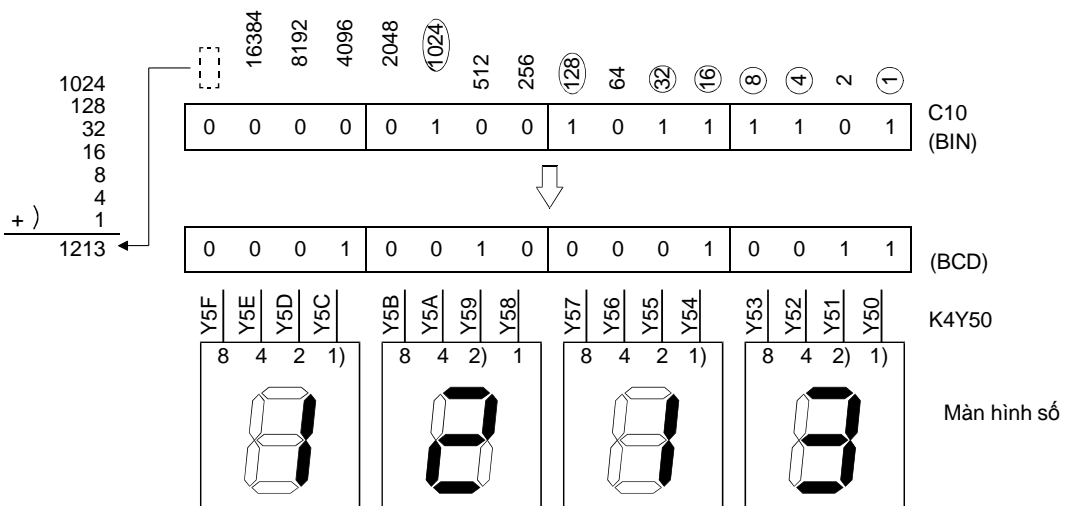
5.2.3 BCD (P) Lệnh biến đổi dữ liệu BIN → BCD



- Khi điều kiện đầu vào được bật, dữ liệu trong biến xác định trong (S) được nhận dạng là biến nhị phân (mã BIN), biến đổi thành thập phân mã hóa nhị phân (mã BCD), và truyền dữ liệu đến biến (D).

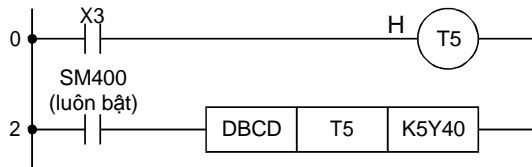
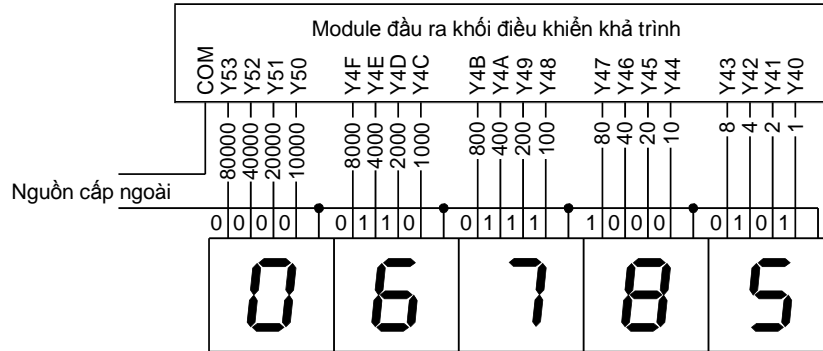


- Màn hình số thông thường hiển thị số theo mã BCD. Vì vậy, lệnh BCD là cần thiết để hiển thị dữ liệu của khối điều khiển khả trình (giá trị hiện tại của bộ định thời và bộ đếm, giá trị thanh ghi dữ liệu của kết quả hoạt động).



Lệnh BCD có thể hiển thị được

- Khoảng hiển thị được của dữ liệu với lệnh BCD (biến đổi từ BIN thành BCD) nằm giữa 0 và 9999. Bất kì dữ liệu nào nằm ngoài khoảng này sẽ gây lỗi. (Error code 4100: OPERATION ERROR)
- Để hiển thị giá trị hiện tại bộ định thời lớn hơn 9,999, sử dụng lệnh DBCD. Lệnh này có thể sử dụng giá trị tới 8 chữ số (lên đến 99,999,999).

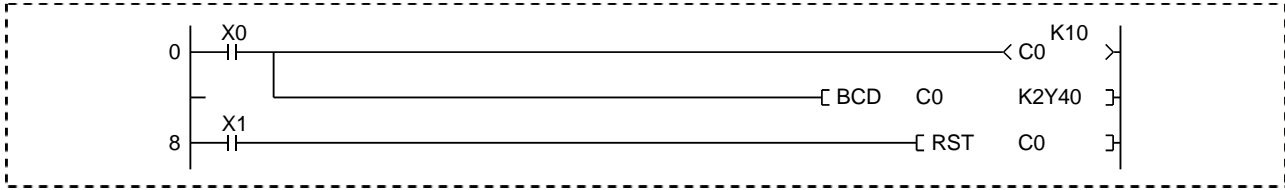


		Biến khả dụng											Số bước cá tnhiet				
		Biến trong (hệ thống hoặc người dùng)		Thanh ghi tập	MELSECNET/10 (H) Direct Jn\		Module chức năng thông minh Un\G	Thanh ghi chỉ số	Hàng số			Con trỏ		Mức			
		Bit	Word	R	Bit	Word		Z	K	H	P	I			N		
BCD	Ⓢ	Ⓧ	Ⓧ	Ⓧ	Ⓧ	Ⓧ	Ⓧ								K1	đế n	3
	Ⓧ	Ⓧ	Ⓧ	Ⓧ	Ⓧ	Ⓧ	Ⓧ										

Tên dự án	QEX8
Tên chương	MAIN

Ví dụ ladder

Tạo ladder sau với GX Works2 và viết nó vào CPU của thiết bị Demo. Sau đó kiểm tra lệnh BCD hoạt động tốt..



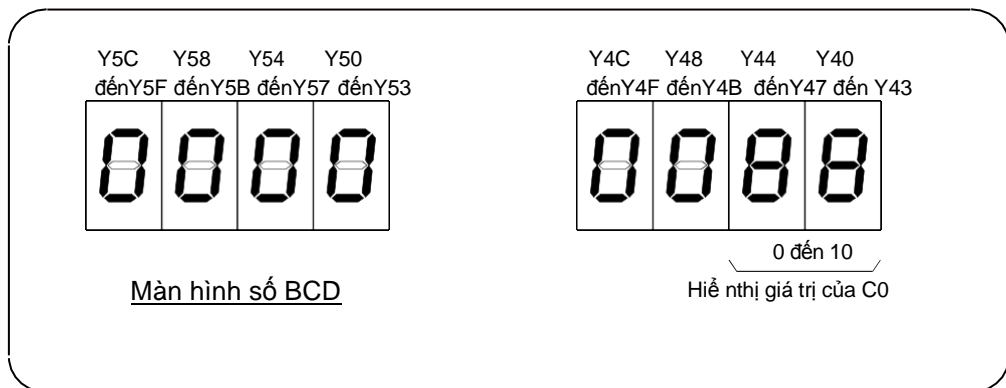
**Quy trình hoạt động**

Quy trình dưới đây tương tự như **Quy trình hoạt động** trong phần 4.4.

- (1) Tạo một dự án mới.
- (2) Tạo một chương trình
- (3) Viết dự án lên khối điều khiển khả trình
- (4) Giám sát ladder

**Luyện tập**

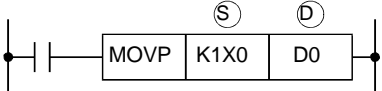
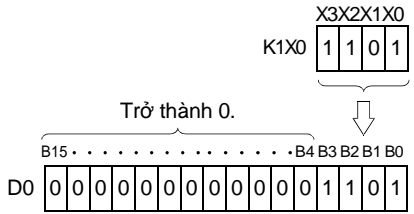
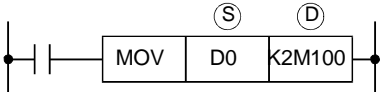
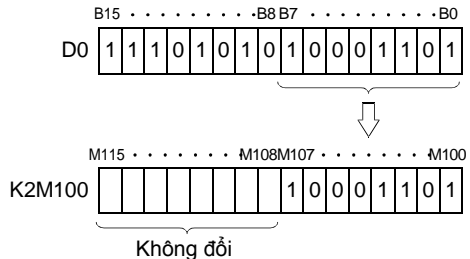
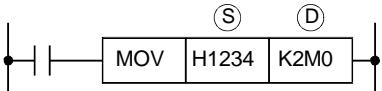
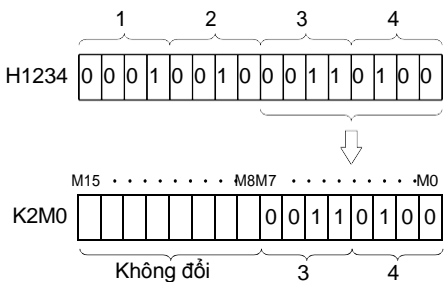
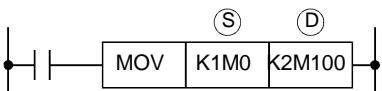
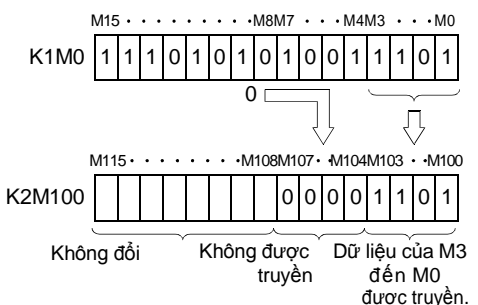
Kiểm tra bật X0 trên cửa sổ điều khiển một vài lần hiển thị giá trị của C0 trên màn hình số BCD từ Y40 đến Y47. Bật X1 lên khởi động lại C0.



**Bài tập liên quan** ---- Bài tập 6



5.2.4 Ví dụ xác định con số cho biến bit và truyền dữ liệu

Ví dụ chương trình	Xử lý
<p>Khi dữ liệu đích <math>\text{D}</math> là một biến từ.</p>  <ul style="list-style-type: none"> <li>• Nguồn: biến nguồn</li> <li>• Đích: biến đích</li> </ul>	<p>Xử lý</p> 
<p>Khi dữ liệu nguồn <math>\text{S}</math> là một biến từ</p> 	
<p>Khi dữ liệu nguồn <math>\text{S}</math> là một hằng số</p> 	
<p>Khi dữ liệu nguồn <math>\text{S}</math> là một biến từ</p> 	

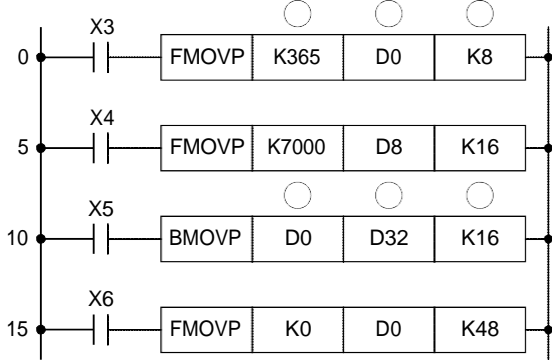
Tên dự án	QB-14
Tên chương	MAIN

5.2.5

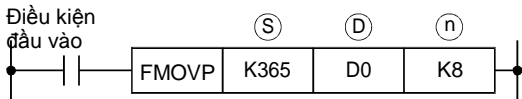
FMOV (P)

FMOV (truyền hàng loạt dữ liệu giống nhau)

BMOV (truyền hàng loạt dữ liệu giống nhau)



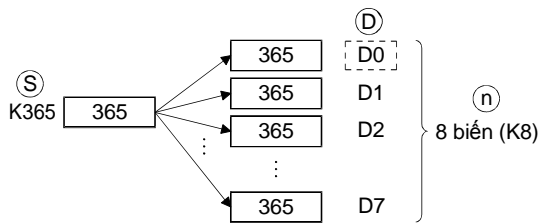
**Giải thích hoạt động**



**FMOV**

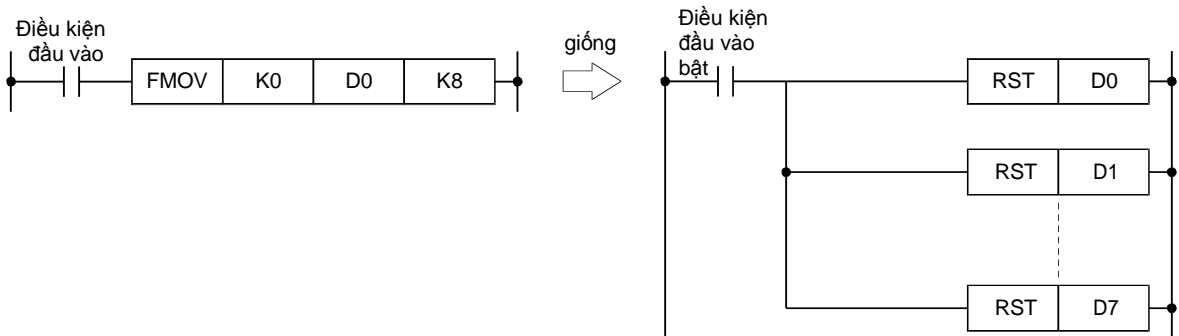
- Khi điều kiện đầu vào được bật, lệnh FMOV truyền dữ liệu được xác trong Biến  $\textcircled{S}$  đến biến được bắt đầu biến từ được xác định trong  $\textcircled{D}$  (số hiệu biến đích được xác định bởi  $\textcircled{n}$ ).

**Ví dụ** Lệnh FMOV thực thi hoạt động sau khi X3 được bật lên.

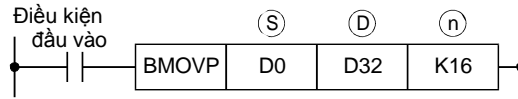


- Lệnh FOMV rất hữu dụng khi xóa một lượng lớn dữ liệu trong batch.

**Ví dụ**



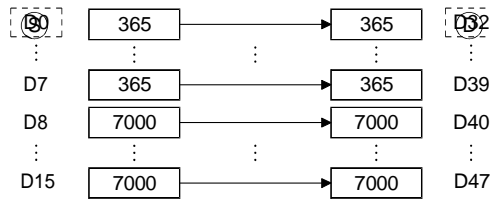
Lệnh FMOV có thể thay thế lệnh RST như hình trên



**BMOV**

- Khi điều kiện đầu vào được bật, lệnh BMOV truyền dữ liệu trong biến bắt đầu biến từ được xác định trong (S) đến biến bắt đầu từ biến xác định trong (D) trong lô (số thiết bị nguồn và thiết bị đích được xác định trong (n))

**Ví dụ** Lệnh BMOV thực thi hoạt động dưới đây khi X5 được bật lên.



- Lệnh BMOV hữu dụng cho những trường hợp sau:
  - Lưu trữ dữ liệu log
  - Lưu trữ dữ liệu quan trọng (như dữ liệu tự động hoá và số liệu đo) vào một vùng khóa. Nó giúp ngăn ngừa mất mát dữ liệu bởi việc mất nguồn.

	Biến khả dụng											Số bước cơ bản												
	biến trong (hệ thống hoặc người dùng)		Thanh ghi tệp	MELSECNET/10 (H) Direct Jn\		Module chức năng thông minh Un\G	Thanh ghi tệp	Hàng số			Con trỏ		Mức											
	Bit	Từ	R	Bit	Từ		Z	K	H	P	I			N										
<table border="1"> <tr> <td>FMOV</td> <td>S</td> <td>D</td> <td>n</td> </tr> <tr> <td>BMOV</td> <td>S</td> <td>D</td> <td>n</td> </tr> </table>	FMOV	S	D	n	BMOV	S	D	n	(S)	☒	☒	☒	☒	☒	☒	(Note)	(Note)	(Note)					K1	4
FMOV	S	D	n																					
BMOV	S	D	n																					
	(D)	☒	☒	☒	☒	☒	☒								đ									
	(n)	☒	☒	☒	☒	☒	☒	☒	☒	☒					n									

(Note) không khả dụng với lệnh BMOV .

Luyện tập

- Viết chương trình ở trang trước lên CPU, rồi chạy CPU.
- Theo hướng dẫn dưới đây để thực thi trình theo dõi loạt biến. Nội dung từ D0 đến D47 có thể được giám sát.  
Viết chương trình vào khối điều khiển khả trình Click [Online] →  
[Monitor] → [Device/Buffer Memory Batch].  
Nhập "D0" trong hộp thoại Device/Buffer Memory Batch Monitor sau đó ấn Enter .
- Click nút Display Format và chọn "Word Multi-point" để Monitor Format.  
→ Click nút OK .

[Giám sát màn hình]

Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

- 1) Bật X3.  
Dữ liệu số 365 được gửi đến tám thanh ghi từ D0 đến D7 trong lô.

Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	7000	7000	7000	7000	7000	7000	7000	7000
D16	7000	7000	7000	7000	7000	7000	7000	7000
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

- 2) Bật X4.  
Dữ liệu số 7000 được gửi đến 16 thanh ghi từ D8 đến D23 trong lô.

Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	7000	7000	7000	7000	7000	7000	7000	7000
D16	7000	7000	7000	7000	7000	7000	7000	7000
D24	0	0	0	0	0	0	0	0
D32	365	365	365	365	365	365	365	365
D40	7000	7000	7000	7000	7000	7000	7000	7000
D48	0	0	0	0	0	0	0	0

- 3) Bật X5.  
Nội dung 16 thanh ghi từ D0 đến D15 được gửi đến 16 thanh ghi của D32 đến D47 trong lô.

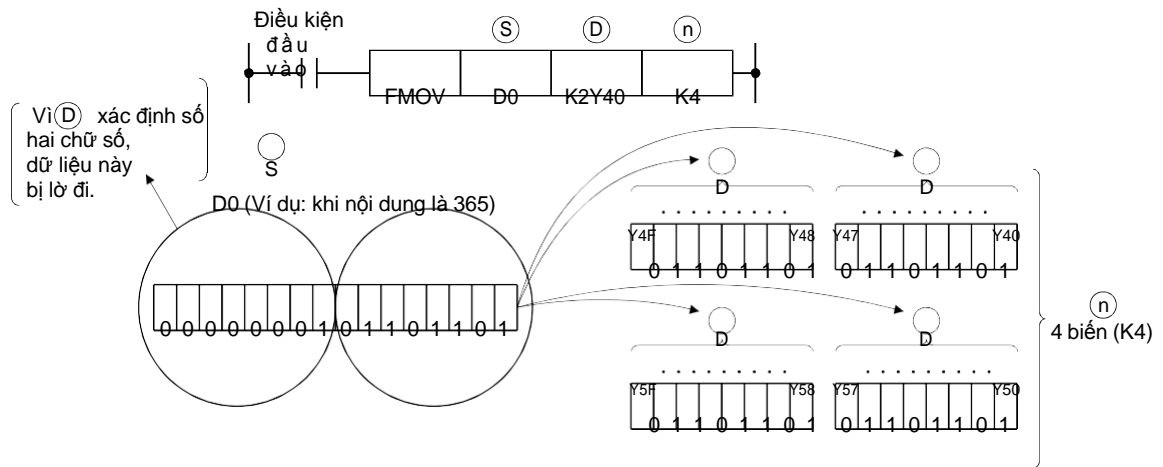
Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

- 4) Bật X6.  
"0" được gửi đến tất cả 48 thanh ghi từ D0 đến D47 trong lô. Nó nghĩa là 48 thanh ghi bị xóa.

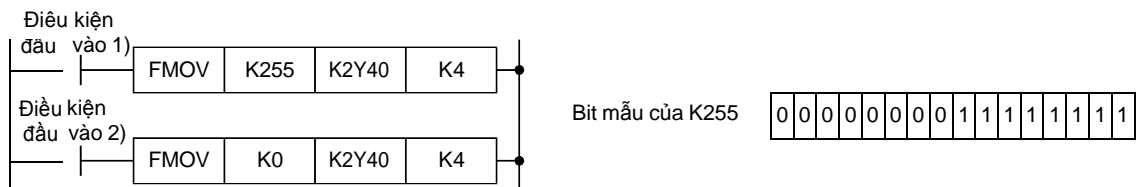
**Tham khảo**

- Nếu (D) là một biến từ, hoạt động diễn ra như sau;

**Lệnh FMOV**



- Giữa những biến từ Y40 đến Y5F, biến được xác định với "1" là đầu ra đầu tiên.
- Trong chương trình bên dưới, bật điều kiện đầu vào lên 1) Bật tất cả đầu ra Y40 đến Y5F và bật điều kiện đầu vào 2) Tắt nó đi.

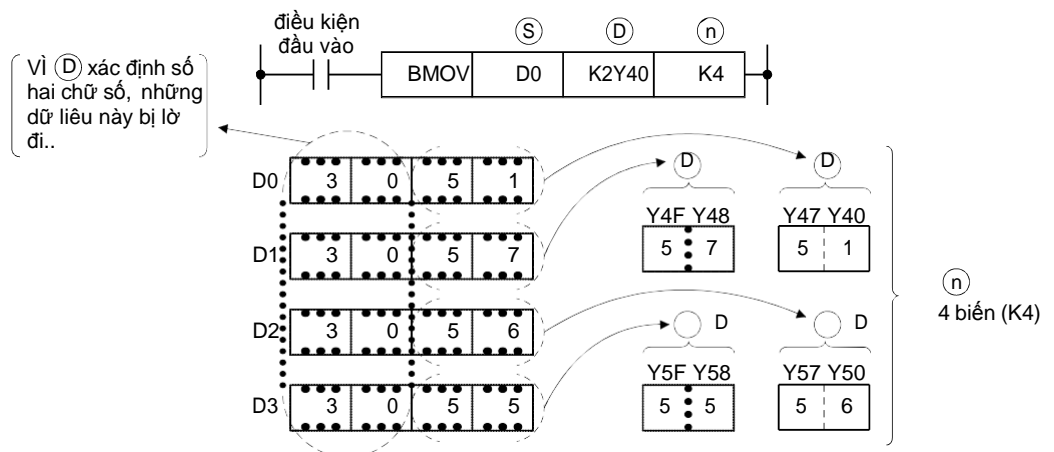


- Trong đơn vị bốn bit, để tắt đi;

- biến 16 bit và ít hơn → Lệnh MOV
- biến 32 bit và ít hơn → Lệnh DMOV
- biến nhiều hơn 32 bit → Lệnh FMOV



**Lệnh BMOV**

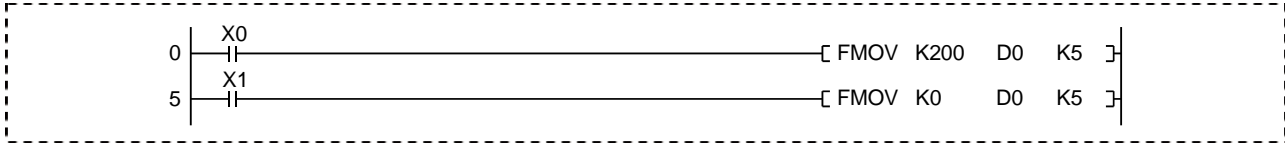


- Trong ví dụ trên, biến từ D0 đến D3 lưu tất cả biến (16 bit). Lệnh BMOV rất hữu dụng để hiển thị và giám sát hai con số cuối cùng thể hiện loại của nó.

Tên dự án	QEX9
Tên chương trình	MAIN

Ví dụ ladder

Tạo ladder dưới đây với GX Works2 và viết nó lên CPU của bộ DEMO. Sau đó kiểm tra lệnh FMOV hoạt động đúng.



**Quy trình hoạt động**

Quy trình dưới đây cũng giống với **Quy trình hoạt động** trong phần 4.4.

- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Viết dự án lên khối khả trình
- (4) Giám sát ladder

**Luyện tập**

Kiểm tra giá trị của biến từ D0 đến D4 trở thành 200 trên màn hình giám sát hàng loạt bằng cách bật X0 trên cửa sổ điều khiển của thiết bị Demo. Bật X1 xóa dữ liệu trong biến.

Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	200	200	200	200	200	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0

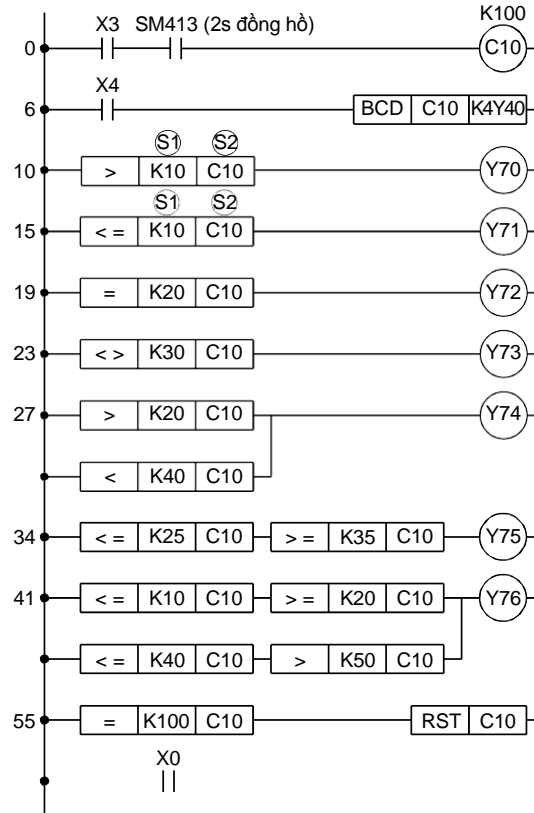
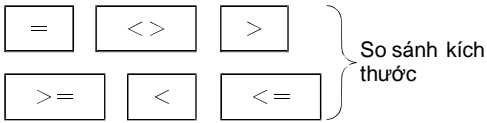
Thay đổi cài đặt của trình giám sát hàng loạt biến như dưới đây để hiển thị số thập phân, thập lục phân, hoặc nhị phân..

- Giá trị: DEC .....hiển thị số theo thập phân.
- Giá trị: HEX .....hiển thị số theo thập lục phân.
- Định dạng giám sát: Bit đa điểm .....hiển thị số theo nhị phân.

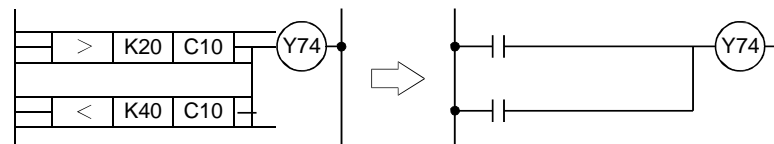
**Bài tập liên quan** ---- Bài tập số 7

Tên dự án	QB-15
Tên chương trình	MAIN

### 5.3 So sánh lệnh hoạt động



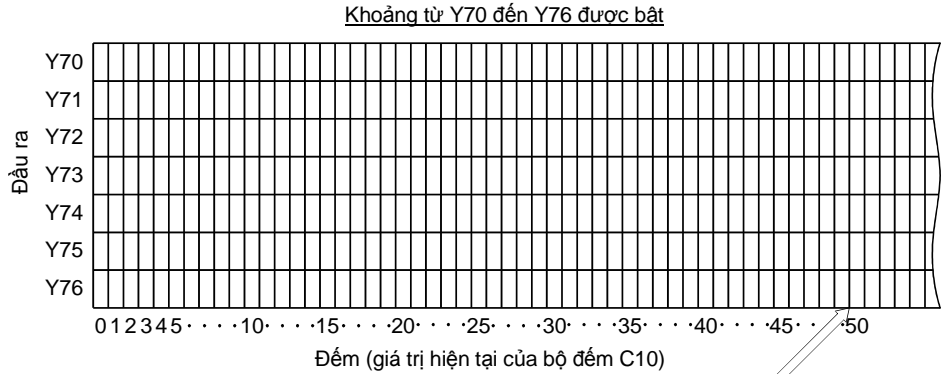
- So sánh lệnh hoạt động so sánh dữ liệu từ nguồn 1 (S1) và nguồn 2 (S2), và khiến thị bị trở nên dẫn điện khi gặp điều kiện.
- Lệnh có thể liên quan đến công tắc thường mở (—|—) vì nó chỉ dẫn khi điều kiện đến.



- $\boxed{= \quad S1 \quad S2}$  .....Dẫn khi nguồn 1 và nguồn 2 giống nhau.
- $\boxed{< \quad S1 \quad S2}$  .....Dẫn khi nguồn 1 nhỏ hơn nguồn 2.
- $\boxed{> \quad S1 \quad S2}$  .....Dẫn khi nguồn 1 lớn hơn nguồn 2.
- $\boxed{<= \quad S1 \quad S2}$  .....Dẫn khi nguồn 1 và nguồn 2 giống nhau hoặc khi nguồn 1 nhỏ hơn nguồn 2.
- $\boxed{>= \quad S1 \quad S2}$  .....Dẫn khi nguồn 1 và nguồn 2 giống nhau hoặc nguồn 1 lớn hơn nguồn 2.
- $\boxed{<> \quad S1 \quad S2}$  .....Dẫn khi nguồn 1 và nguồn 2 không giống nhau.

Luyện tập

- Viết chương trình vào CPU.
- Bật X3 và X4.
- C10 bắt đầu đếm. (đếm cứ mỗi 2s.) Giá trị bộ đếm hiện tại được hiển thị trên màn hình số (Y40 đến Y4F).
- Chắc chắn rằng biến Y70 đến Y76 bật như sau.



Sự khác nhau giữa  > và  >= ⇨

>	K50	C10	bằng 49.
>=	K50	C10	bằng 50.

- Bộ đếm được thiết kế để đặt lại mỗi 200s.
- Theo cách này, lệnh so sánh không chỉ so sánh không chỉ so sánh một dữ liệu và còn một dải cụ thể. Chức năng này được sử dụng khá phổ biến cho chương trình để đánh giá độ chấp nhận của sản phẩm.

		Biến khả dụng											Con số	Các bước cơ bản			
		Biến trong (hệ thống hoặc người dùng)		Thanh ghi tập	MELSECNET /10 (H) Direct Jn\		Module chức năng thông minh Un\G	Thanh ghi chỉ số	Hàng số		Con trỏ				Mức		
		Bit	Từ	R	Bit	Từ		Z	K	H	P	I			N		
Lệnh so sánh	S1	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	K1 đến K4	3
	S2	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ	Ⓟ		



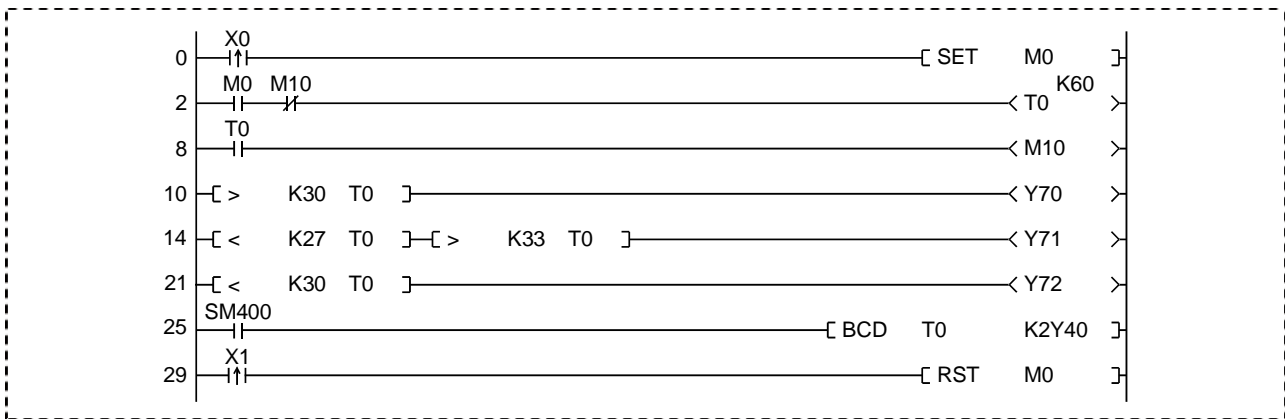
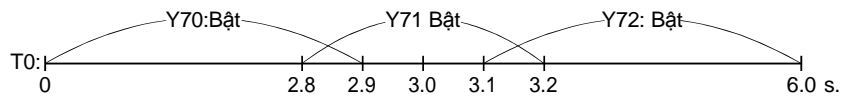
Tên dự án	QEX10
Tên chương trình	MAIN

Ví dụ ladder

Đọc ladder sau và viết nó lên CPU của bộ DEMO. Sau đó kiểm tra lệnh > and < hoạt động đúng.

0s. ≤ T0 < 3s. → Y70: ON, 2.7s. < T0 < 3.3s. →


Y71: ON, 3s. < T0 ≤ 6s. → Y72: ON



**Quy trình hoạt động**

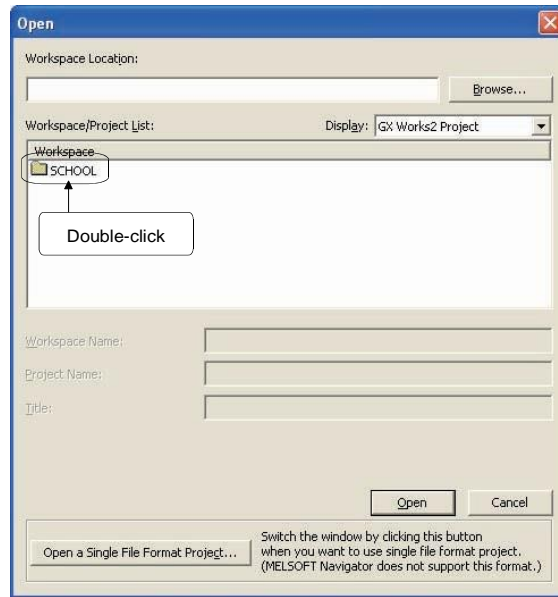
(1) Đọc dữ liệu

Đọc dữ liệu dự án.

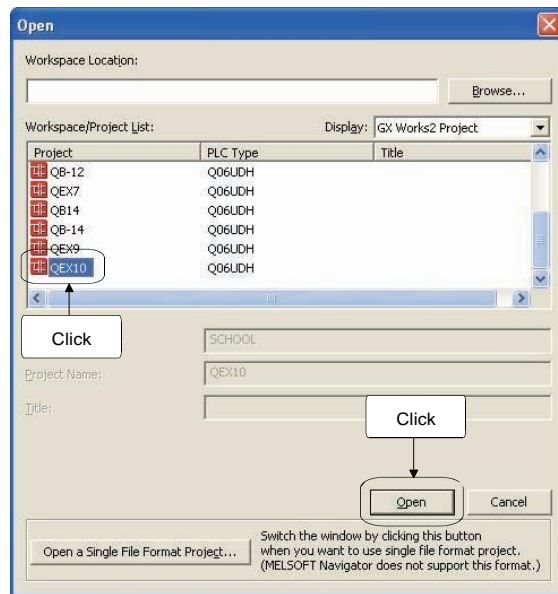
- Click  trên thanh công cụ.



- Hộp thoại Open hiện ra. Xác định đích để lưu.
- Click đúp vào vùng hoạt động hiện ra "SCHOOL".



- Click "QEX10" và click nút  .

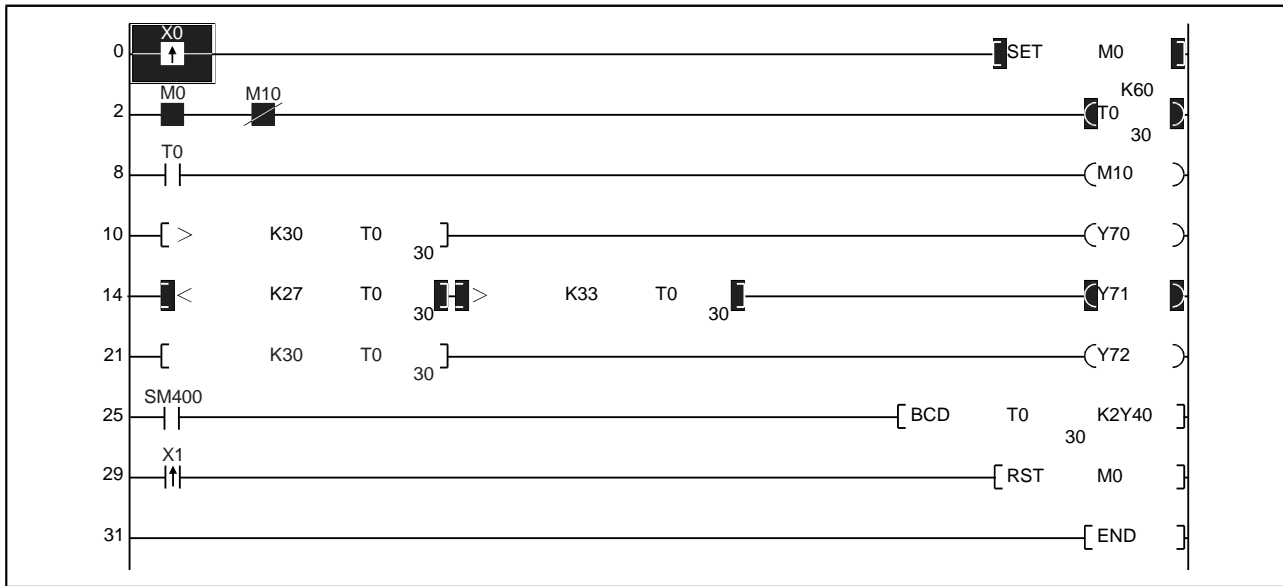


Quy trình hoạt động dưới đây tương tự như  trong phần 4.4.

- (2) Viết dự án lên khối khả trình
- (3) Giám sát ladder

Luyện tập

- Bật X0 và kiểm tra chương trình hoạt động đúng.



Bài tập liên quan ---- Bài tập 8

### 5.4 Lệnh hoạt động số học

Tên dự án	QB-16
Tên chương trình	MAIN

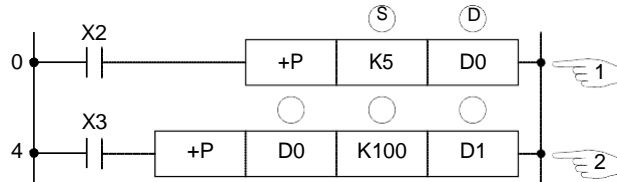
- 5.4.1 

+(P)
------

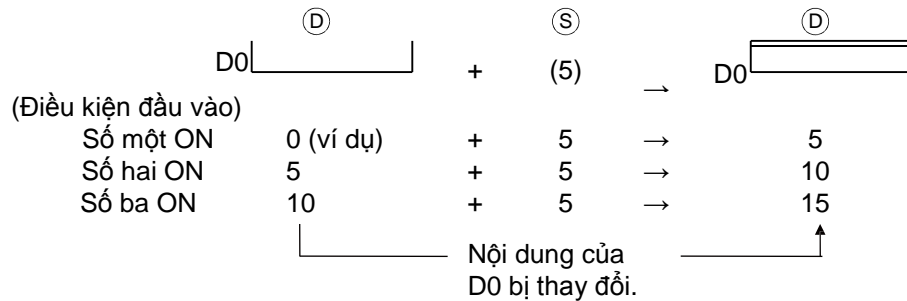
 Cộng dữ liệu BIN 16-bit  

-(P)
------

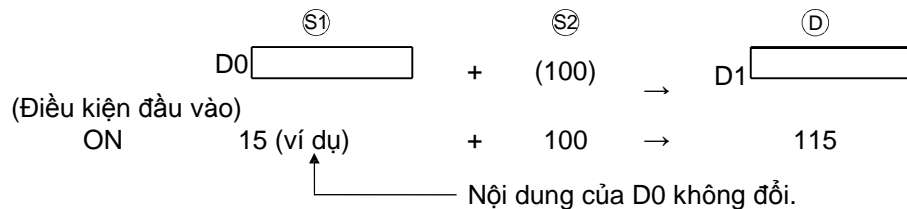
 Trừ dữ liệu BIN 16-bit



- 1 • Mỗi lần điều kiện đầu vào được bật, giá trị của biến được xác định trong  $\textcircled{D}$  được thêm vào nội dung của biến trong  $\textcircled{S}$  và nội dung được lưu trữ xác định trong  $\textcircled{D}$ .



- 2 • Khi điều kiện đầu vào được bật, giá trị của biến được xác định trong  $\textcircled{S1}$  được thêm vào nội dung biến xác định trong  $\textcircled{S2}$  và nội dung được lưu ở thiết xác định trong  $\textcircled{D}$ .



**Lưu ý**

- |      |
|------|
| +(P) |
|------|

 hay 

-(P)
------

 phải sử dụng cho lệnh cộng hoặc trừ.
- Khi 

+
---

 hay 

-
---

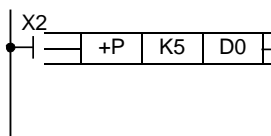
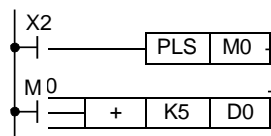
 được sử dụng, hoạt động cộng hoặc trừ được thực thi mỗi lần quét. Sử dụng 

+
---

 hay 

-
---

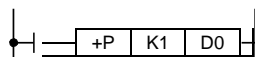
, toán hạng phải được chuyển thành xung trước.

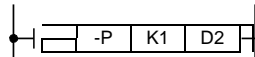
**Tham khảo**

- Hai lệnh dưới đây hoạt động tương tự như hoạt động cộng hoặc trừ.

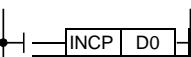
(Cộng)




(Trừ)



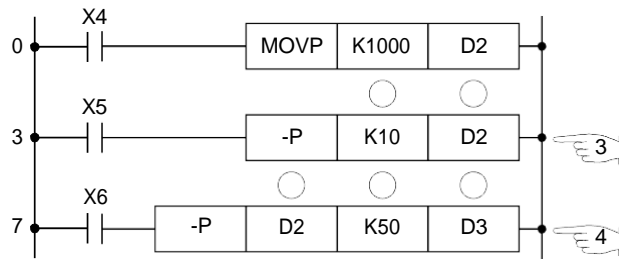
(Cộng)



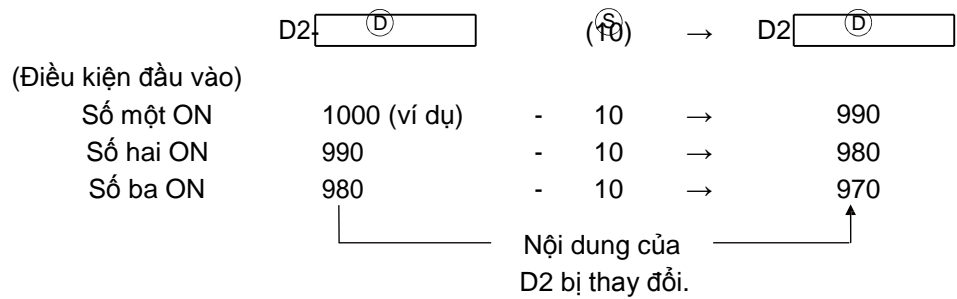
(Trừ)



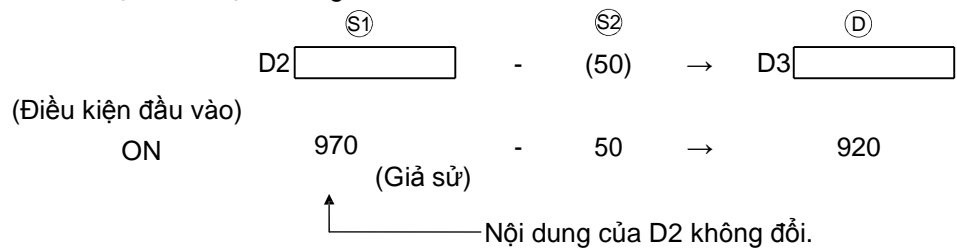
Tên dự án	QB-17
Tên chương trình	MAIN



- 3 • Mỗi khi điều kiện đầu vào được bật, giá trị của biến được xác định trong  $\textcircled{S}$  bị trừ đi biến từ xác định trong  $\textcircled{D}$  và kết quả được lưu trữ ở trong  $\textcircled{D}$ .



- 4 • Khi điều kiện đầu vào được bật, nội dung của biến được xác định trong  $\textcircled{S2}$  bị trừ nội dung biến được xác định trong  $\textcircled{S1}$  và kết quả được lưu ở nội dung biến được xác định trong  $\textcircled{D}$ .



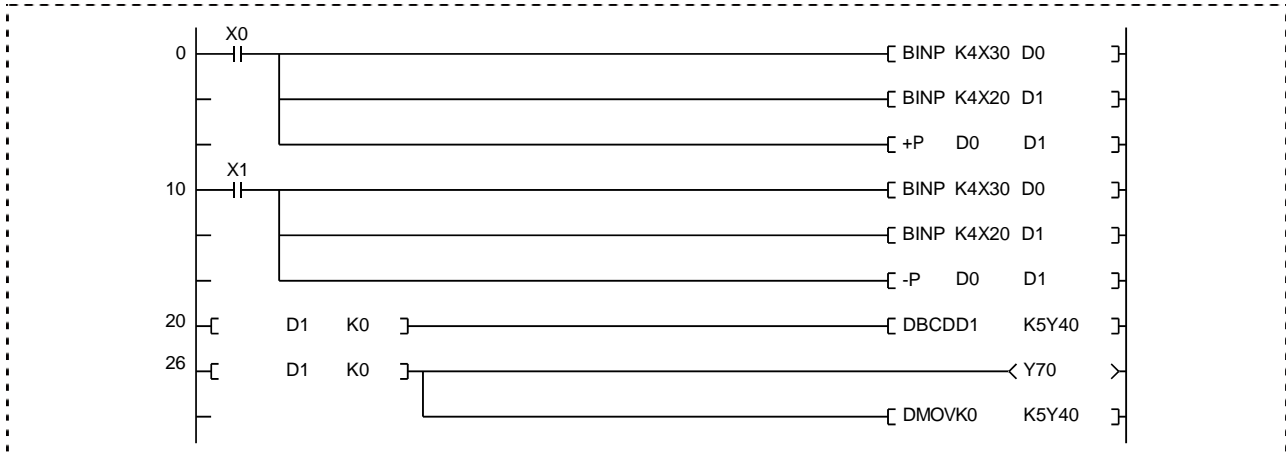
		Biến khả dụng										Con số	Số bước thực hiện				
		Biến trong (hệ thống hoặc người dùng)		Thanh ghi tệp	MELSECNET/10 (H) Direct Jn\		Module chức năng thông minh UniG	Thanh ghi chỉ số	Hàng số		Con trỏ			Mức			
		Bit	Word	R	Bit	Word		Z	K	H	P			I	N		
Lệnh cộng/trừ	$\textcircled{S}$ $\textcircled{D}$	$\textcircled{S1}$ $\textcircled{S2}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	K1	3
Lệnh cộng trừ	$\textcircled{S1}$ $\textcircled{S2}$ $\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	$\textcircled{D}$	đến K4	4

Số bước cơ bản là bốn cho  $\textcircled{S1}$   $\textcircled{S2}$   $\textcircled{D}$ .

Tên dự án	QEX11
Tên chương trình	MAIN

Ví dụ ladder

Tạo ladder dưới đây với GX Works2 và viết nó lên CPU của bộ DEMO. Sau đó kiểm tra lệnh trừ và cộng hoạt động bình thường.



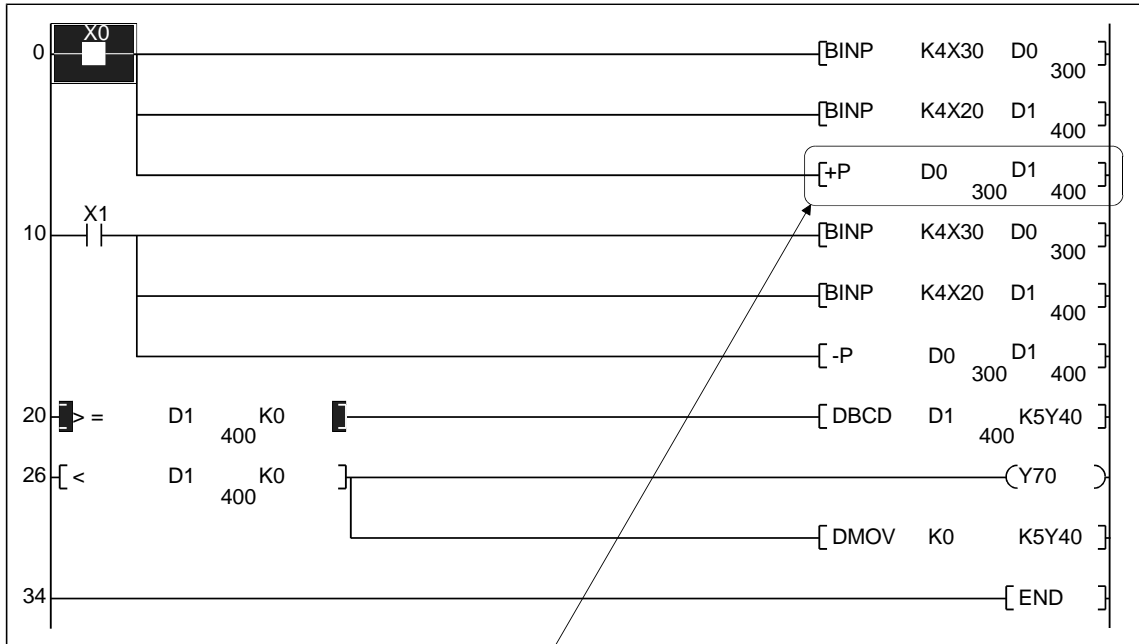
Quy trình hoạt động

Quy trình hoạt động dưới đây tương tự như Quy trình hoạt động trong phần 4.4.

- (1) Tạo một dự án mới
- (2) Tạo một chương trình mới
- (3) Viết dự án lên khối điều khiển khả trình
- (4) Giám sát ladder

Quy trình hoạt động

- (1) Khi X0 được bật, dữ liệu từ X30 đến 3F và X20 đến 2F được thêm vào, và kết quả đầu ra từ Y40 đến Y53.
- (2) Khi X1 được bật, dữ liệu từ X30 đến 3F bị trừ đi từ dữ liệu trong X20 đến 2F, và kết quả đầu ra từ Y40 đến Y53. Khi kết quả là giá trị âm, Y70 được bật và Y40 đến Y53 bị xóa về 0.

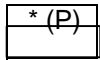


+	D0	D1	=	D1+D0 → D1
				100+30 → 400

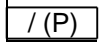
Bài tập liên quan ---- Bài tập 9

Tên dự án	QB-18
Tên chương trình	MAIN

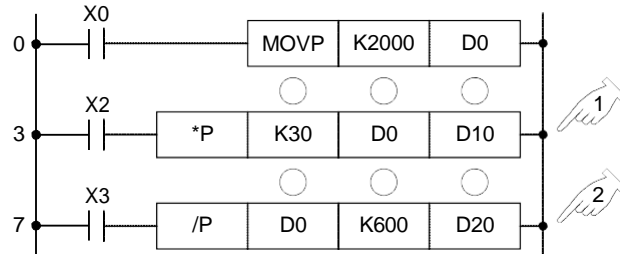
5.4.2 \* (P) Nhân dữ liệu BIN 16-bit



Nhân dữ liệu BIN 16-bit



Chia dữ liệu BIN 16-bit



- Khi điều kiện đầu vào được bật, nội dung của biến được xác định trong S1 được nhân lên với nội dung của biến xác định S2 và kết quả được lưu trong biến được xác định trong D.

$$\begin{array}{c} \text{S1} \\ \text{K30} \end{array} \times \begin{array}{c} \text{S2} \\ \text{D0} \\ \text{2000} \end{array} = \begin{array}{c} \text{D} \\ \text{D11} \quad \text{D10} \\ \text{60000} \end{array}$$

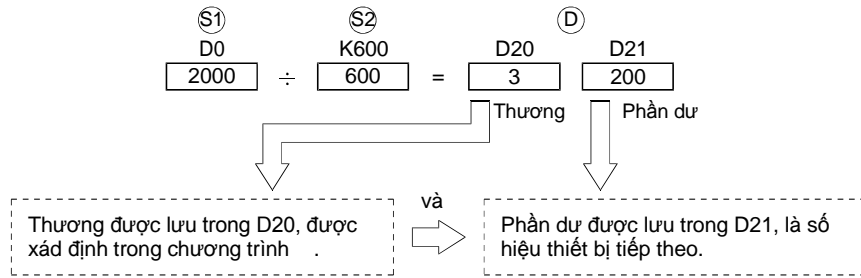
Để lưu kết quả của 16-bit dữ liệu × 16-bit dữ liệu.  
16 bit (1 từ) là không đủ.  
Vì vậy, D10 được xác định trong chương trình và số kế tiếp D11 tạo thành kết quả.

Biến này được coi như thanh ghi 32-bit để giữ kết quả. Hầu hết các bit còn lại của D10 (B15) không phải là bit quyết định âm hay dương. Đó cũng là một phần của dữ liệu.

Lệnh này tương tự như 32 bit để lập trình với tính toán kết quả của lệnh \* (P). (như lệnh DMOV và lệnh DBCD)



- 2 • Khi điều kiện đầu vào được bật, nội dung của biến được xác định trong S1 bị chia bởi nội dung của biến được xác định ở biến S2 và kết quả được lưu ở trong xác định là D.



Giá trị sau điểm thập phân của kết quả hoạt động bị lờ đi.

- Khi một biến bit xác định trong D, thương được lưu, nhưng phần dư thì không được lưu.
- Dưới đây là ví dụ để xử lý giá trị âm.

Ví dụ  $-5 / (-3) = 1, \text{ dư: } -2$   
 $5 / (-3) = -1, \text{ dư: } 2$

- Dưới đây là ví dụ của chia bởi số 0, hoặc chia cho 0 bởi một số.

Ví dụ  $0 / 0$  } Lỗi "OPERATION ERROR"  
 $1 / 0$  }  
 $0 / 1,$  Cả thương và phần dư đều bằng 0.

Luyện tập

- Viết chương trình lên CPU và chạy nó.
- Bật X0 và lưu "2000" (giá trị BIN) trong D0.
- Bật X2. Hoạt động dưới đây được thực thi.  
 Nếu "60000"(kết quả hoạt động D11 và D10 được hiểu là số nội 16-bit và chỉ D10 được giám sát, "-5536" được hiện ra. Để tránh điều này, làm theo hướng dẫn trong trang dưới đây.

S1 (K30) \* S2 (D0) = D (D11 D10)  
 (30) (2000) (60000)

- Bật X3.

S1 (D0) ÷ S2 (K600) = D (D21 D20)  
 (2000) (600) (200) (3)  
 Thương Phần dư

	Biến khả dụng											Con số Cacs bước cơ bản		
	Biến trong (hệ thống hay người dùng)		Thanh ghi tập	MELSECNET/10 (H) Direct Jn\		Module chức năng thông minhUn\G	Thanh ghi chỉ số	Hàng số		Con trỏ			Mức	
	Bit	Word	R	Bit	Word	Z	K	H	P	I	N			
Lệnh nhân/chia S1 S2 D	S1	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	K1 đến K4	*/ 4
	S2	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑		
	D	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑		

Số bước cơ bản cho lệnh nhân là ba hoặc bốn, và cho chia là bốn.

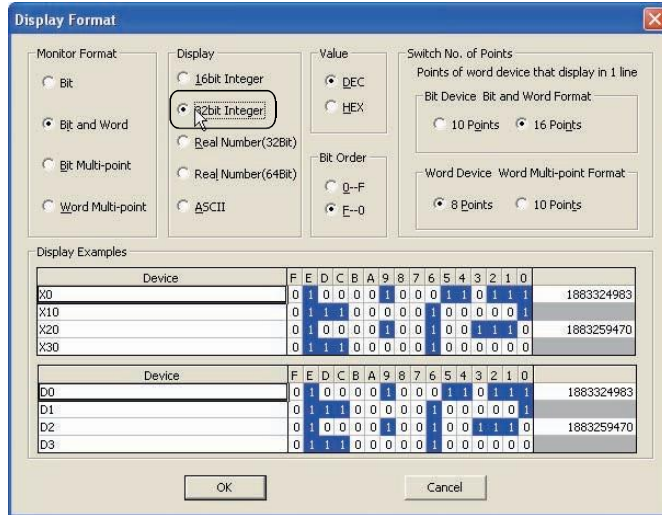
\*: Lệnh nhân thay đổi dựa trên số biến được sử dụng

- Làm thế nào để giám sát dữ liệu số nguyên 32-bit

Khi kết quả hoạt động của lệnh nhân nằm ngoài khoảng 0 đến 32,767, kết quả không thể hiện ra chính xác ngay cả khi số đó là số nguyên 16-bit và nội dung của thanh ghi thấp nhất được giám sát trong ladder.

Để giám sát những số đó chính xác, làm theo hướng dẫn dưới đây.

- Click nút **Display Format** ở trong hộp thoại Device/Buffer Batch Monitor và chọn "32bit Integer" của "Display".  
Click nút **OK**.



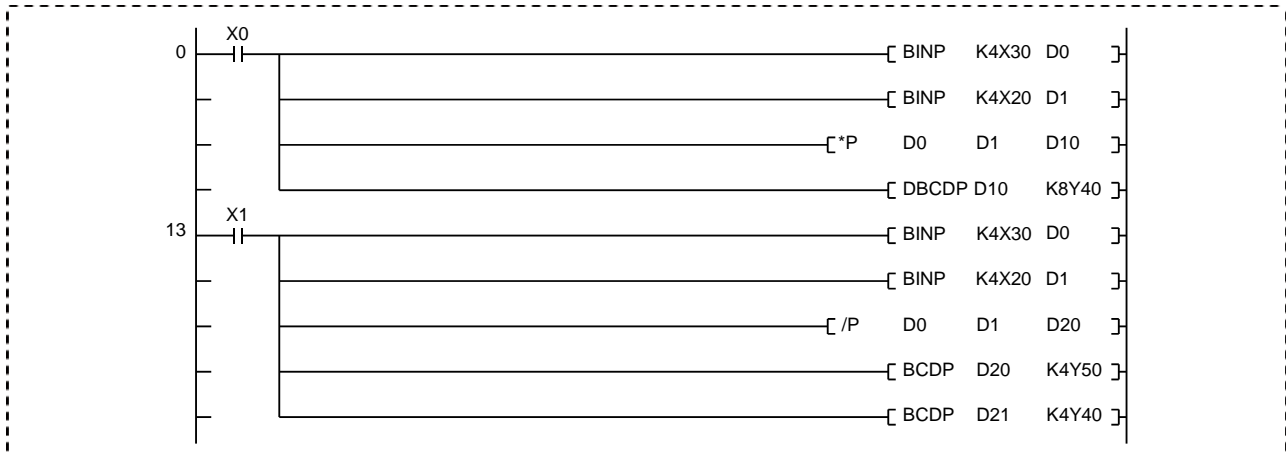
- Dữ liệu được giám sát

Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	2000
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	60000
D11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tên dự án	QEX12
Tên chương trình	MAIN

Ví dụ ladder

Tạo ladder dưới đây với GX Works2 và viết nó lên CPU của bộ DEMO. Sau đó kiểm tra lệnh nhân và chia hoạt động đúng.



**Quy trình hoạt động**

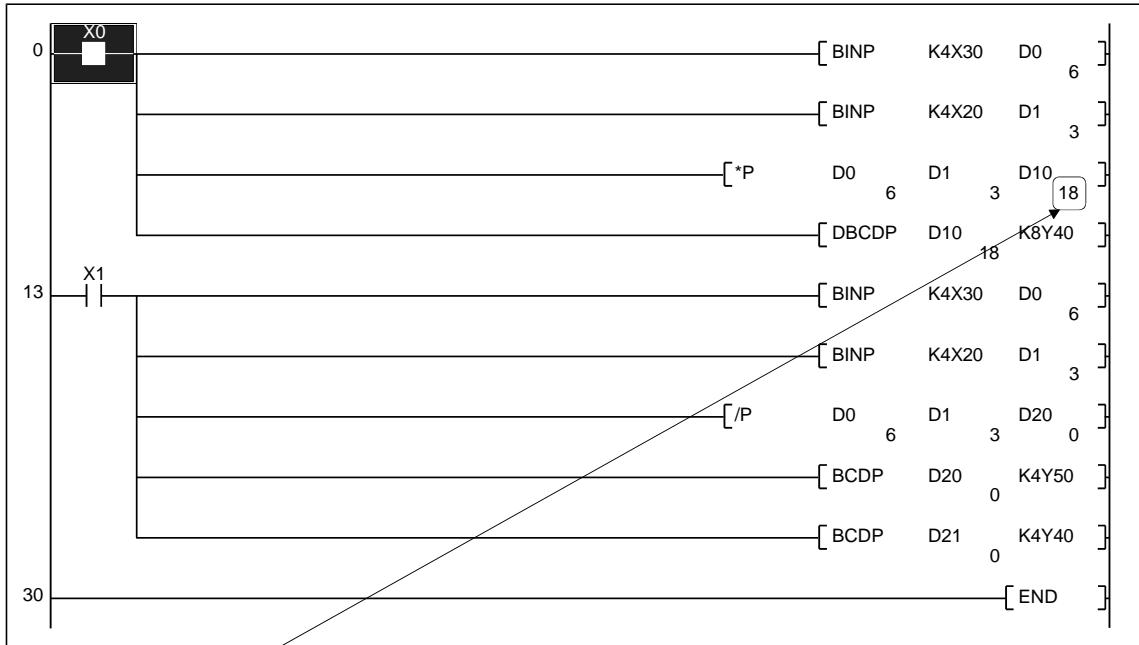
Quy trình hoạt động cũng tương tự như

**Quy trình hoạt động** trong phần 4.4.

- (1) Tạo một dự án mới
- (2) Tạo một chương trình
- (3) Viết dự án lên bộ điều khiển khả trình
- (4) Giám sát ladder

Luyện tập

- (1) Khi X0 được bật, dữ liệu từ X20 đến X2F được nhân với dữ liệu từ X30 đến 3F, và kết quả đầu ra từ Y40 đến 5F.
- (2) Khi X1 được bật, dữ liệu từ X30 đến X3F bị chia bởi dữ liệu từ X20 đến 2F. Thương số đầu ra từ Y50 đến 5F, và phần dư đầu ra từ Y40 đến 4F.



$D0 * D1 = D10$   
 $6 * 3 = 18$

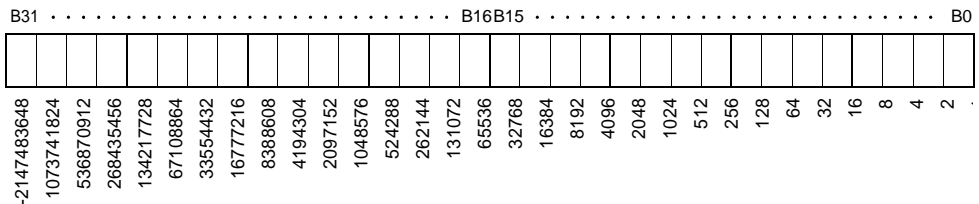
Bài tập liên quan ---- Bài tập 10, Bài tập 11

### 5.4.3 Lệnh dữ liệu 32 bit và sự cần thiết của nó

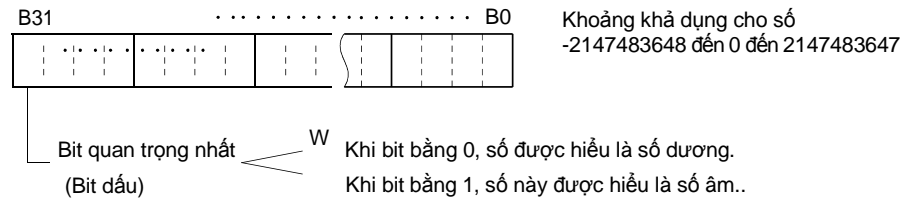
- Đơn vị nhỏ nhất của bộ nhớ dữ liệu của khối điều khiển khả trình Q-series là 1 từ gồm 16 bit. Vì thế, thông thường, dữ liệu được xử lý trong 1 từ cơ bản trong hoạt động truyền, so sánh, và số học.
- Bộ điều khiển khả trình Q-series có thể xử lý dữ liệu trong 2 từ cơ bản (32 bit). Trong trường hợp này, "D" được thêm vào đầu mỗi lệnh để chỉ ra lệnh này gồm 2 từ. Dưới đây là ví dụ.

Lệnh \ Dữ liệu	1 từ 16 bit	2 từ 32 bit
	Truyền	MOV(P) BIN(P) BCD(P)
So sánh	<, >, <= >=, =, <>	D<, D>, D<= D>=, D=, D<>
Hoạt động số học	+(P)	D+(P)
	-(P)	D-(P)
	*(P)	D*(P)
	/(P)	D/(P)
Khoảng khả dụng cho giá trị	-32,768 đến 32,767 Giá trị trong ngoặc là cho lệnh BIN(P), BCD(P). <div style="display: inline-block; vertical-align: middle; border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">                     0 đến 9,999                 </div>	-2,147,483,648 đến 2,147,483,647 Giá trị trong ngoặc là cho lệnh DBIN(P), DBCD(P). <div style="display: inline-block; vertical-align: middle; border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">                     0 đến 99,999,999                 </div>
Khoảng khả dụng cho con số	K1 đến K4	K1 đến K8

- Trọng số của cấu hình 32 bit như dưới đây::



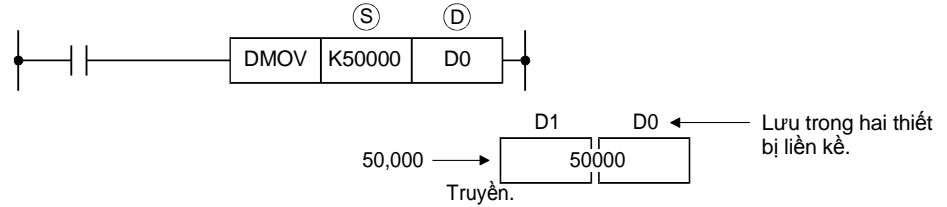
Trong trường hợp xử lý 16 bit dữ liệu, khối điều khiển khả trình xử lý 32 bit giá trị âm trong phần bù. Vì thế, bit quan trọng nhất B31 (B15 cho 16 bit dữ liệu) là một bit dấu



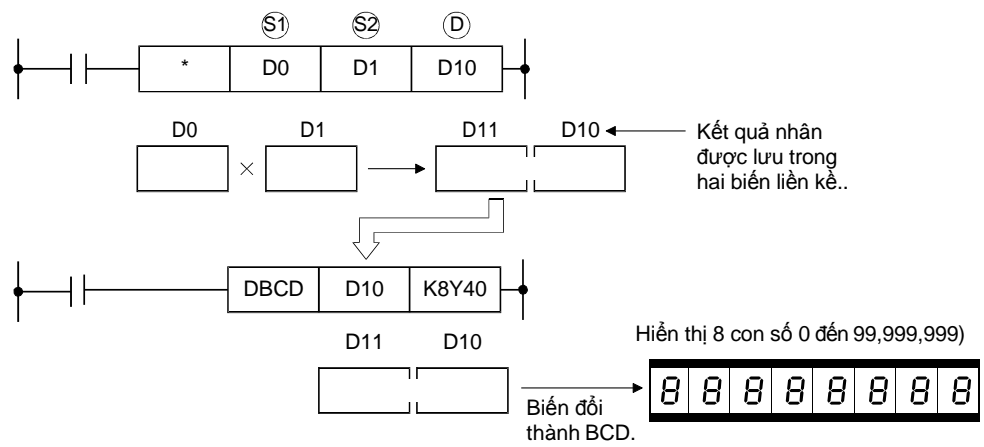
- Dù dữ liệu được xử lý trong 2 từ cơ bản (32 bit) hoặc không dựa vào kích thước của biến.

Trong hai trường hợp dưới đây, lệnh 2 từ phải được sử dụng.

- 1) Khi kích thước dữ liệu vượt khoảng (-32768 đến 32767) thì dữ liệu có thể được xử lý như 1 từ.

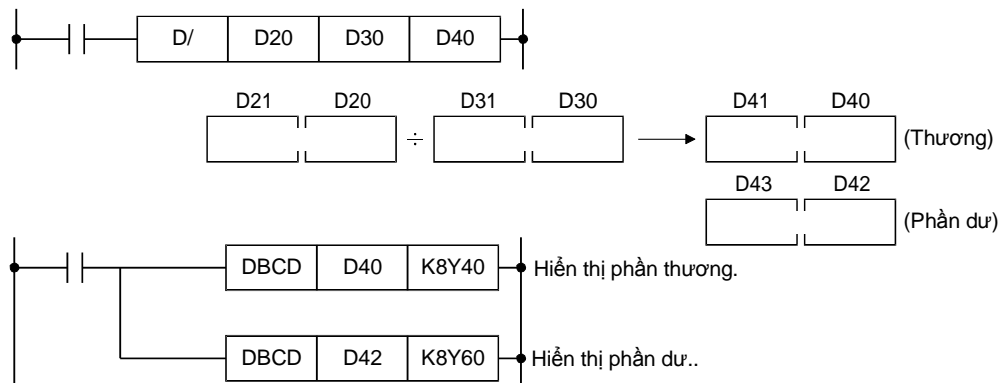


- 2) Khi kết quả của lệnh nhân 16 bit (1 tập lệnh) được truyền.



\*: Kết quả của dữ liệu nhân 32 bit là 64 bit.

- 3) Khi kết quả của lệnh chia 32 bit được sử dụng.



Tên dự án	QB-19
Tên chương trình	MAIN

5.4.4 Ví dụ tính toán cho phép nhân và chia bao gồm số sau dấu phẩy (khi nhân hoặc chia được sử dụng)

**Ví dụ 1** Ví dụ tính toán để quyết định trường hợp

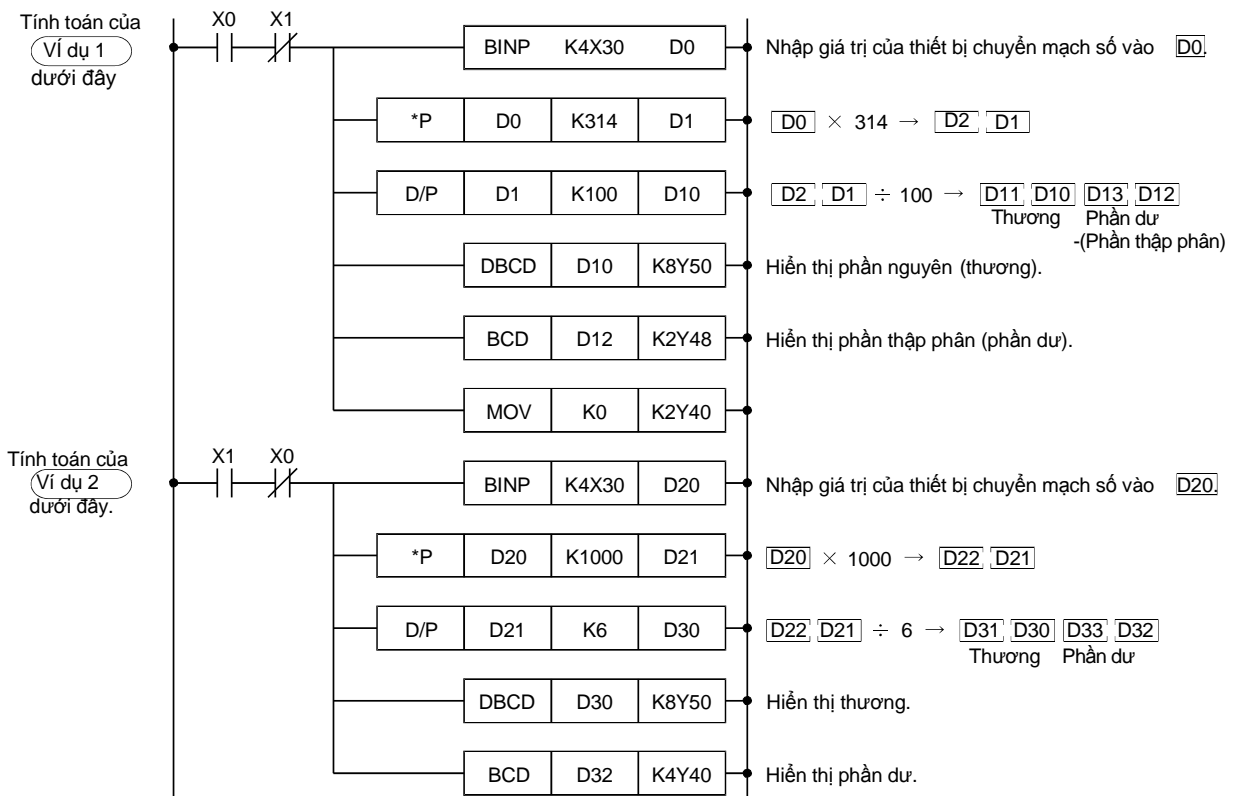
Giá trị số  $\times 3.14 \rightarrow$  Phần nguyên và Phần thập phân  
 (K4X30) (Làm tròn) (K8Y50) (K2Y48)

- Phương pháp lập trình  
 Xử lí số làm tròn như 314 ( $3.14 \times 100$ ), và sau đó chia kết quả cho 100.

**Ví dụ 2** Ví dụ tính toán để xử lí giá trị sau dấu phẩy (ví dụ là chia)

Giá trị số  $/ 0.006 \rightarrow$  Thương và Phần dư  
 (K4X30) (K8Y50) (K4Y40)

- Phương pháp lập trình  
 Để xử lí 0.006 như là phần nguyên 6, nhân cả số chia và số bị chia cho 1000.



**Chú ý**  
 QCPU có những lệnh có thể xử lí hoạt động dữ liệu số thực (dấu phẩy động) cho độ chính xác cao hơn  
 Khi nào lệnh còn được dùng, chú ý cẩn thận với điểm đặt của dấu phẩy thập phân như trên là không cần thiết

## 5.5 Thanh ghi chỉ số và thanh ghi tệp

### 5.5.1 Làm thế nào để sử dụng thanh ghi chỉ số Z

- Thanh ghi chỉ số (Zn) được sử dụng để nhận dạng số hiệu biến gián tiếp. Dữ liệu bổ sung ở thanh ghi chỉ số và nhận dạng số hiệu biến trực tiếp có thể được xem như là số hiệu biến.

Ví dụ

**D0Z0** → có thể hiểu như D (0+Z0)  
Số hiệu biến

Ví dụ, khi Z0 bằng 0, số hiệu biến trở thành D0.  
khi Z0 bằng 0, số hiệu biến trở thành D50.

- Z0 đến Z19 có thể sử dụng như thanh ghi chỉ số.
- Thanh ghi chỉ số (Zn) là một biến từ bao gồm 16 bit. Vì thế, khoảng kích thước dữ liệu khả dụng là -32768 đến +32767.
- Biến dưới đây có thể sử dụng cho chỉ số.

Biến bit ..... X, Y, M, L, S, B, F, Jn\X, Jn\Y, Jn\B, Jn\SB (như K4Y40Z0)

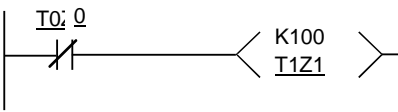
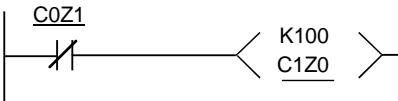
Biến từ.... T<sup>(Note)</sup>, C<sup>(Note)</sup>, D, R, W, Jn\W, Jn\SW, Jn\G (như D0Z0)

Hàng số..... K, H (như K100Z0)

Con trở ..... P

(Lưu ý) Chỉ những giá trị hiện tại có thể sử dụng cho bộ định thời và bộ đếm.

Những luật dưới đây được đưa ra để sử dụng thanh ghi chỉ số cho công tắc hoặc cuộn dây

Thiết bị	Mô tả	Ví dụ ứng dụng
T	• Chỉ Z0 và Z1 là khả dụng cho một công tắc và cuộn dây của bộ định thời.	
C	• Chỉ Z0 và Z1 là khả dụng cho một công tắc và cuộn dây của bộ định thời.	

#### Chú ý

Khi thanh ghi chỉ số được sử dụng với lệnh dữ liệu 32 bit

Zn và Zn+1 là mục tiêu để xử lí.

16 bit thấp tương ứng với thanh ghi chỉ số được xác định (Zn), và 16 bit cao tương ứng với thanh ghi chỉ số được xác định +1

Chỉ số 32 bit

(chỉ áp dụng với mô hình phổ biến QCPU (trừ Q00UJCPU))

Với phương pháp để xác định thanh ghi chỉ số cho chỉ số 32 bit có thể được chọn từ hai phương pháp sau.

- Xác định khoảng chỉ số sử dụng cho chỉ số 32 bit
- Xác định chỉ số 32 bit cho đặc tính “ZZ”

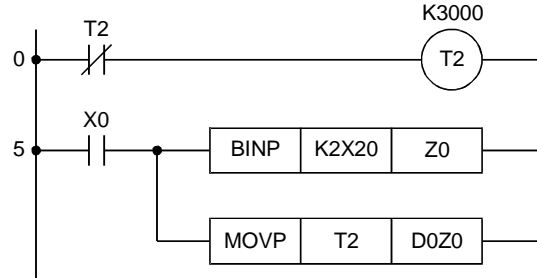
Xem phụ lục 8 để biết thêm chi tiết về chỉ số.



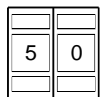
Ví dụ ứng dụng

- Viết dữ liệu lên thanh ghi dữ liệu với số được xác định trong chuyển mạch số.

Tên dự án	Thanh ghi chỉ số
Tên chương trình	MAIN



- Kiểm tra hoạt động của ladder thực thi trình theo dõi lô biến.
- Quy trình hoạt động tương tự như trong phần 5.2.1.  
Đặt số 2 con số bất kì ở cột chuyển mạch số (X27 đến X20) và bật X0.



X27 to X20  
Z0= 50  
D0Z0=D50

Device	+0	+1	+2	+3	+4	+5	+6	+7
D0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	2675	0	0	0	0	0
D56	0	0	0	0	0	0	0	0
D64	0	0	0	0	0	0	0	0

Giá trị hiện tại của T2 được truyền đến D50.

## 5.5.2 Làm thế nào để sử dụng thanh ghi tập R

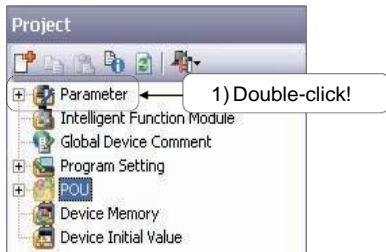
- Thanh ghi tập (R) gồm 16 bit cũng như thanh ghi dữ liệu (D).
- Đặt thanh ghi tập trong RAM tiêu chuẩn của QCPU hoặc một thẻ nhớ (card SRAM và card Flash). Thanh ghi tập được lưu trên card Flash chỉ có thể đọc từ chương trình.

Bộ nhớ chương trình	} Lưu tham số, chương trình, ghi chú thiết bị, và giá trị thiết bị ban đầu. (Thanh ghi tập không thể lưu.)
RAM tiêu chuẩn	
ROM tiêu chuẩn	} Lưu tham số, chương trình, ghi chú thiết bị, và giá trị thiết bị ban đầu. (Thanh ghi tập không thể lưu.)
Card bộ nhớ	
	} Lưu thanh ghi tập từ 1K đến 4086K. (Số tập thanh ghi tối đa lưu được dựa vào thẻ nhớ sử dụng.)

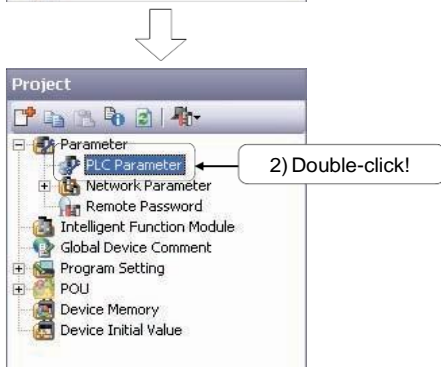
- Dữ liệu của thanh ghi tập vẫn còn sau khi khởi động lại hoạt động hoặc sau khi nguồn bị tắt.
- Để xóa dữ liệu, viết 0 vào thanh ghi tập với lệnh MOV(P) hoặc GX Works2.
- Sử dụng [Write to PLC] của GX Works2 hoặc một chương trình tuần tự để viết chương trình.
- Sử dụng [Export to ROM Format] của GX Works 2 để sao chép dữ liệu trong ROM tiêu chuẩn hoặc card Flash.
- Xác định vùng của thanh ghi 1K cơ bản (1024 điểm) với tham số.

### Ví dụ ứng dụng

- Đặt 32K điểm của thanh ghi tập R0 đến R3267 để sử dụng trong chương trình.



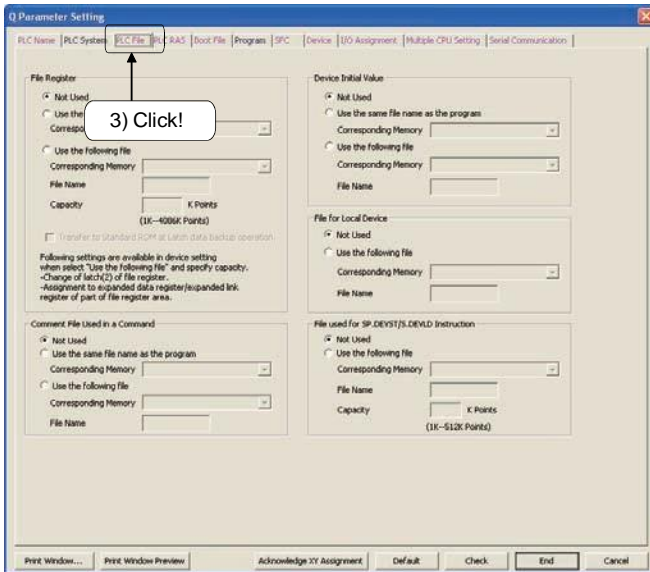
1) Click đúp "Parameter" trong danh mục dự án.



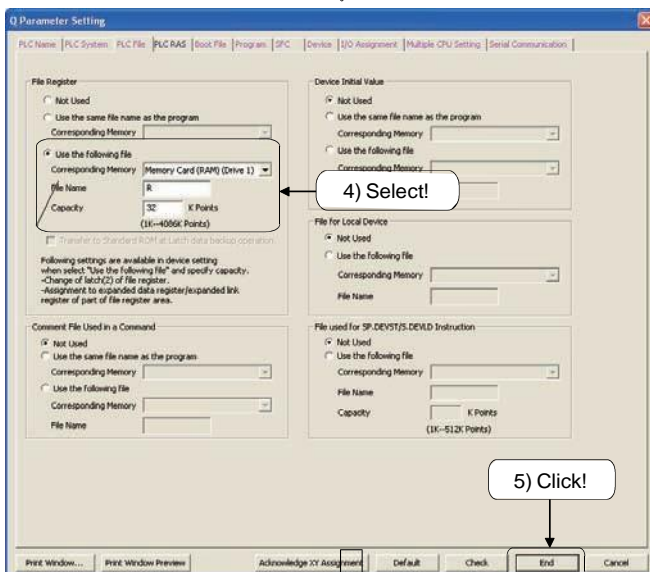
2) "PLC Parameter", "Network Parameter", và "Remote Mật khẩu" hiện ra. Click đúp "PLC Parameter"

(Đến trang sau)

(Từ trang trước)



- 3) Hộp thoại Q Parameter Setting hiện ra. Click vào thanh PLC Tập.



- 4) Kiểm tra hộp thoại "Use the following tệp" và chọn "Memory Card (RAM) (Drive 1)" cho "Corresponding Memory".

Nhập tham số dưới đây vào "Tệp Name" và "Capacity".

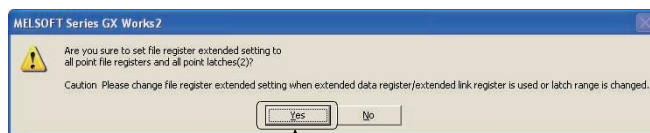
[Setting contents]

Tệp Name : R

Capacity 32

- 5) Sau khi phần cài đặt hoàn thành, click nút.

End



- 6) Thông báo bên trái hiện ra. Click vào nút

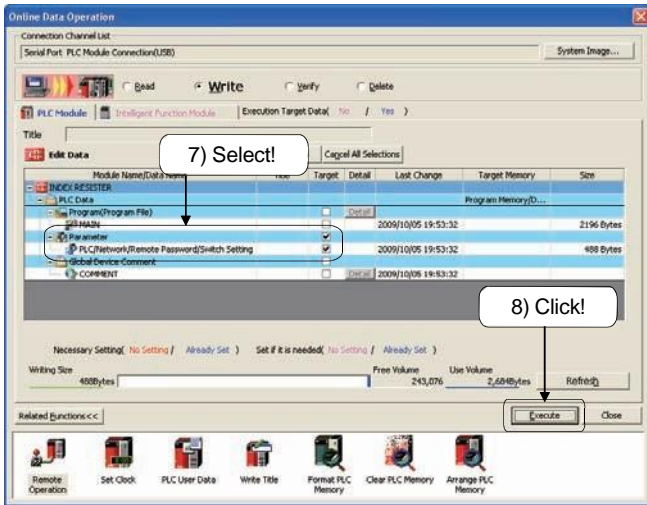
Yes

6) Click!



(Đến trang sau)

(Từ trang trước)



7) Click [Online] → [Write to PLC] để hiển thị hộp thoại Online Data Operation. Chọn "Parameter" trong thanh PLC Module.

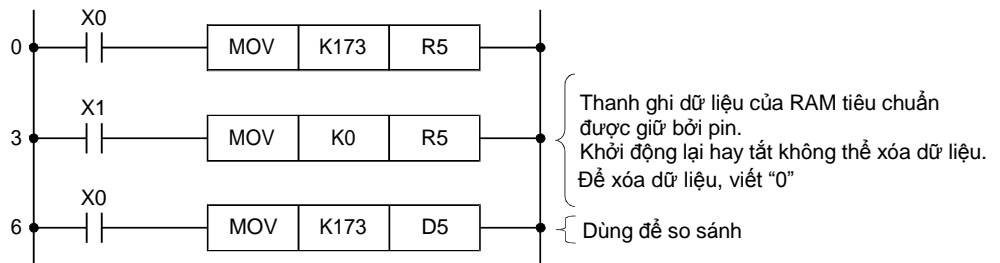
8) Click nút **Execute** để viết dữ liệu.

- Để xóa thanh ghi tập dữ liệu với chương trình. viết chương trình sau.

Cho quy trình hoạt động, xem phần 4.4

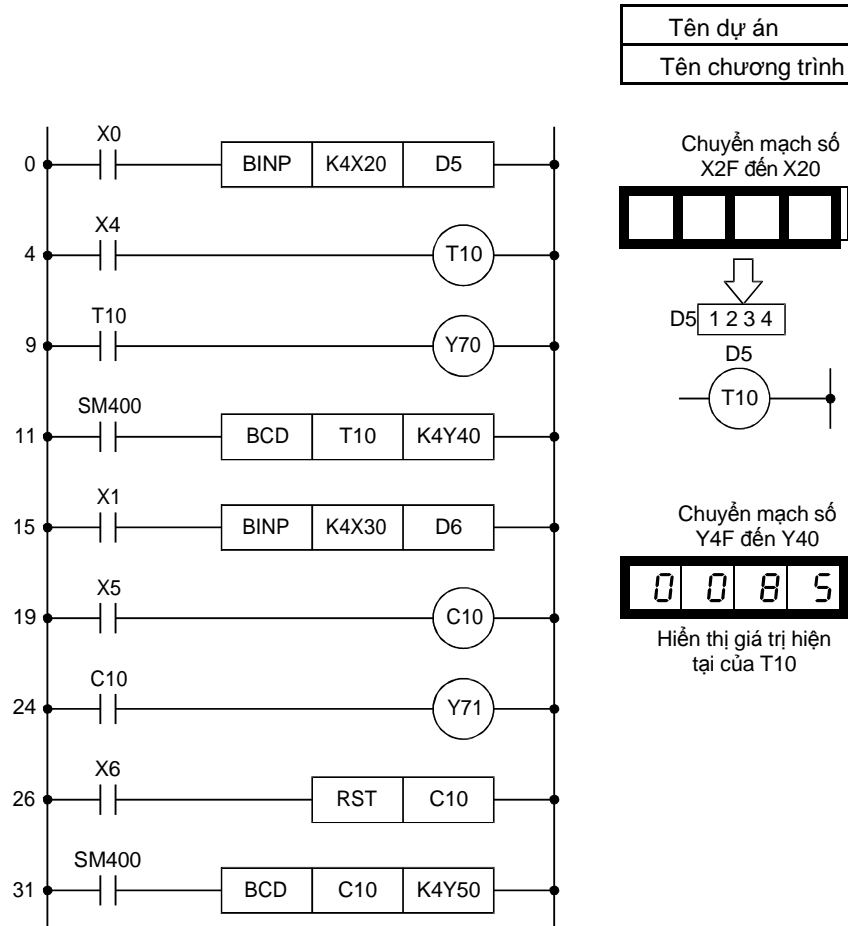
Bật X0 để viết dữ liệu, và bật X1 để xóa dữ liệu.

Tên dự án	Thanh ghi tập
Tên chương trình	MAIN



5.6 Cài đặt mở rộng của giá trị đặt cho bộ định thời/bộ đếm và hiển thị mở rộng của giá trị hiện tại

Bộ định thời và bộ đếm có thể được xác định bởi K (hàng số thập phân) trực tiếp hoặc bởi D (thanh ghi dữ liệu) gián tiếp. Trong chương trình bên dưới, bộ chuyển mạch số mở rộng có thể thay đổi giá trị đặt.



- Sau khi đọc chương trình vào GX Works2, viết nó vào khối điều khiển khả trình để kiểm tra nó hoạt động đúng.

Quy trình hoạt động

Bước (1) của quy trình dưới đây cũng giống với Quy trình hoạt động trong phần 5.3.

Bước (2) đến (4) của quy trình dưới đây cũng giống với Quy trình hoạt động trong phần 4.4.

- (1) Đọc dữ liệu
- (2) Tạo một chương trình
- (3) Viết một chương trình lên khối điều khiển khả trình
- (4) Giám sát ladder

Luyện tập

- (1) Cài đặt mở rộng với giá trị đặt bộ định thời và hiển thị của giá trị hiện tại
- Đặt giá trị thiết lập của bộ định thời trong chuyển mạch số (X20 đến 2F), và bật chuyển mạch X0.
  - Khi chuyển mạch X4 được bật, Y70 được bật lên sau khi thời gian được xác định với chuyển mạch số. (Ví dụ, Y70 bật lên sau 123.4s khi 

1	2	3	4
---	---	---	---

 được đặt.)
  - Màn hình số (Y40 đến 4F) hiển thị giá trị hiện tại của bộ định thời T10.
- (2) Cài đặt mở rộng của giá trị thiết lập bộ đếm và hiển thị của giá trị hiện tại
- Đặt giá trị thiết lập trong chuyển mạch số (X30 đến 3F), và bật chuyển mạch X1.
  - Bật vaf tắt chuyển mạch X5 lặp đi lặp lại. Khi X5 được bật cho thời gian được xác định với bộ chuyển mạch số (đếm lên), Y71 được bật.
  - Màn hình số (Y70 đến 5F) hiển thị giá trị hiện tại của bộ đếm C10 (thời gian X5 được bật lên).
  - Bật chuyển mạch X6 xóa bộ đếm C10 đến 0. Khi công tắc C10 đã được bật (đếm lên), công tắc được giải phóng.

Tên dự án	QTEST5
Tên chương trình	MAIN

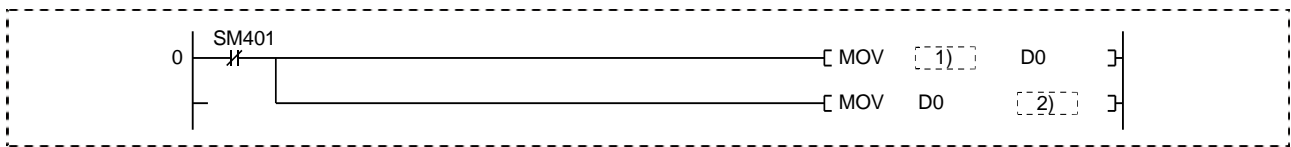
## 5.7 Bài tập

### 5.7.1 Bài tập 1 MOV

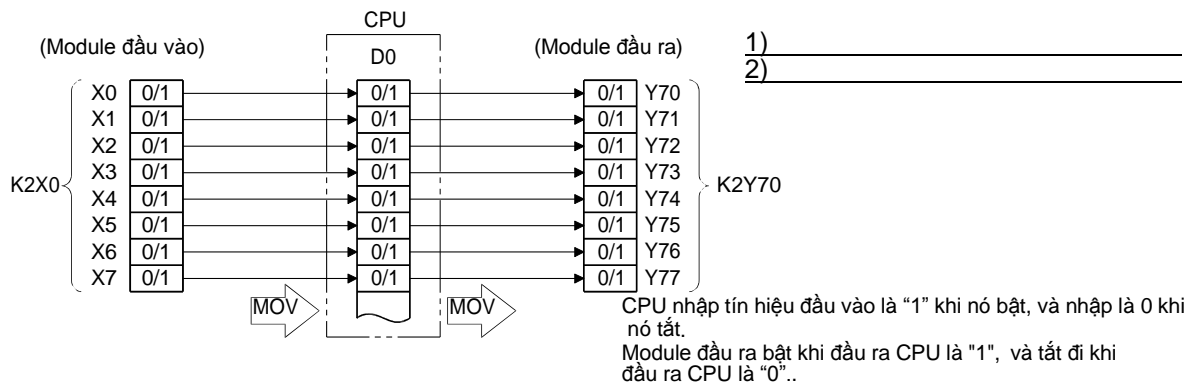
Truyền 8 trạng thái đầu vào (X0 đến X7) đến D0 khi đầu ra của chúng đến Y70 đến Y77.)

X0 → Y70  
 X1 → Y71  
 X2 → Y72  
 X3 → Y73  
 X4 → Y74  
 X5 → Y75  
 X6 → Y76  
 X7 → Y77

Tạo chương trình sau với GX Works2 bằng cách điền vào [ ]  
 Sau đó, kiểm tra hoạt động sử dụng thiết bị Demo.



Gợi ý.



So sánh

Hình dưới đây cho thấy một chương trình được tạo với lệnh tuần tự, không phải với lệnh MOV.

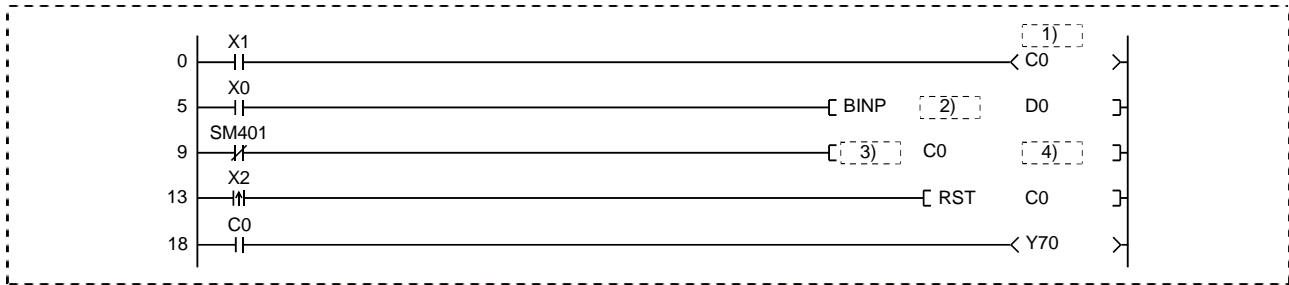


Tên dự án	QTEST6
Tên chương trình	MAIN

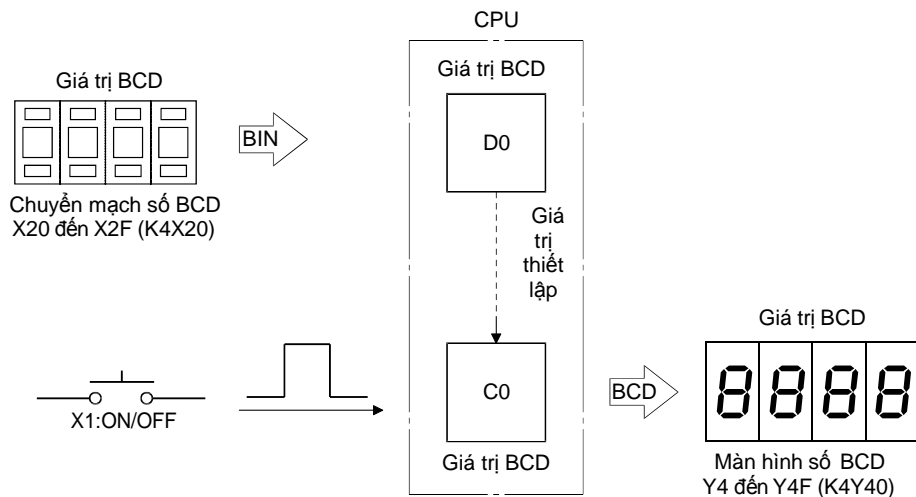
5.7.2 **Bài tập 2** Chuyển đổi BIN và BCD

Đầu ra là số lần X1 được bật trên màn hình kết nối với Y40 đến Y4F trên BCD. Như là một điều kiện trước hết, giá trị thiết lập của bộ đếm (C0) có thể làm đầu vào với chuyển mạch số (X20 đến X2F) và cài đặt này sẽ có giá trị khi bật X0.

Tạo chương trình sau với GX Works2 bằng cách điền vào [ 1 ] .  
Sau đó, kiểm tra hoạt động sử dụng thiết bị Demo.



Gợi ý



- 1) \_\_\_\_\_
- 2) \_\_\_\_\_
- 3) \_\_\_\_\_
- 4) \_\_\_\_\_

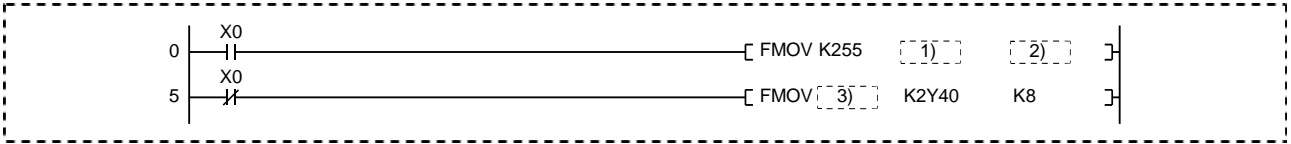


Tên dự án	QTEST7
Tên chương trình	MAIN

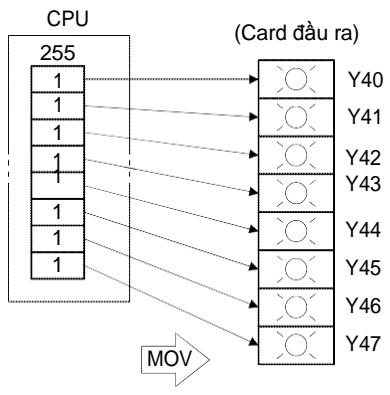
5.7.3 **Bài tập 3** FMOV

Tạo một chương trình mà khi bật X0 lên sẽ bật 64 bit đầu ra từ Y40 đến Y7F và tắt X0 đi sẽ tắt 64 bit đầu ra Y40 đến Y7F.

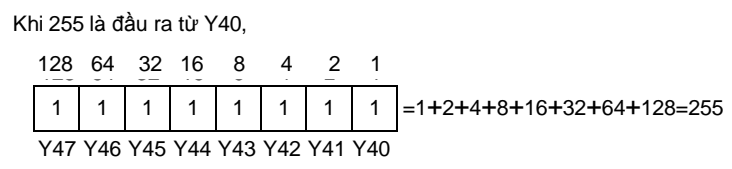
Tạo chương trình sau với GX Works2 bằng cách điền vào [     ]. Sau đó, kiểm tra hoạt động sử dụng thiết bị Demo.



Hint



Đầu ra từ CPU trong kí hiệu nhị phân là hàng số.

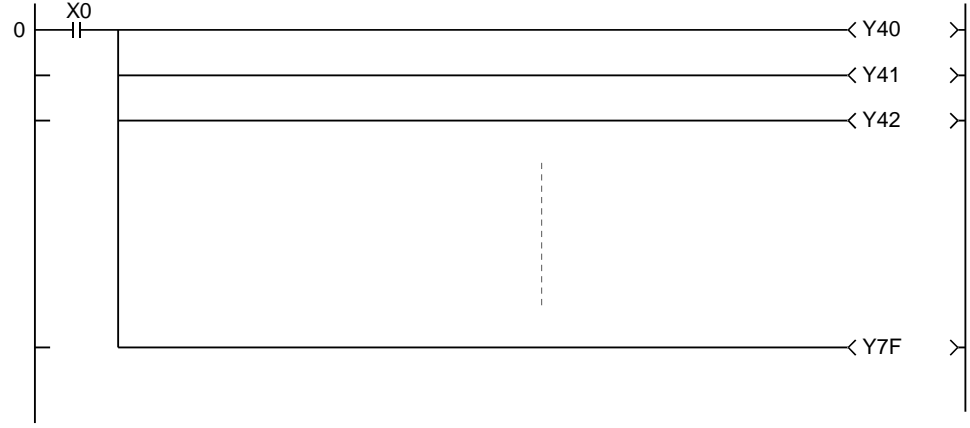


Trong bài này, một module đầu ra 64 điểm được sử dụng (Y40 đến Y7F). Có bao nhiêu khối cần thiết cho 255 ở đầu ra cơ bản ?

- 1) \_\_\_\_\_
- 2) \_\_\_\_\_
- 3) \_\_\_\_\_

So sánh

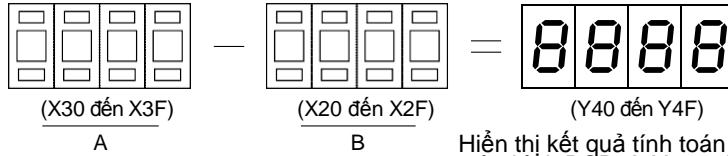
Hình dưới đây chỉ ra một chương trình được tạo ra với lệnh tuần tự, không phải với lệnh FMOV. 130 bước được sử dụng.



Tên dự án	QTEST8
Tên chương trình	MAIN

5.7.4 **Bài tập 4** Lệnh so sánh

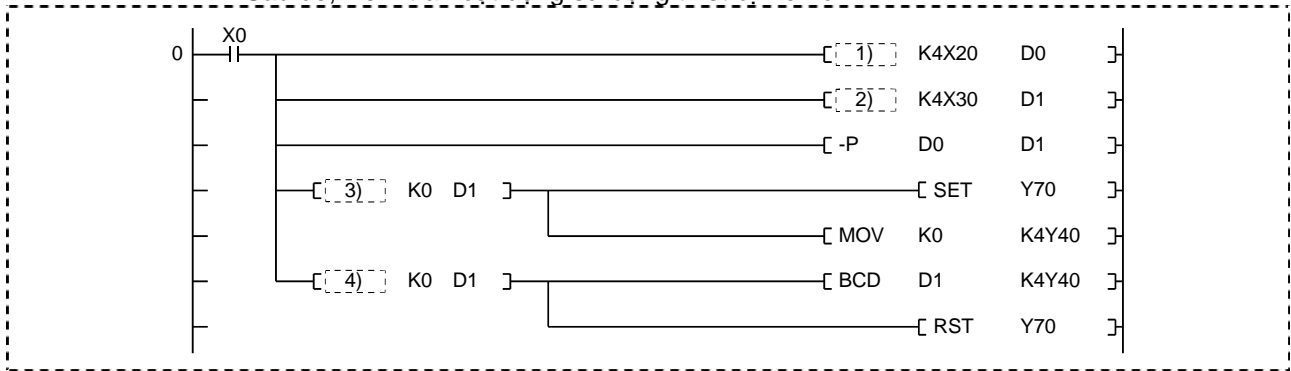
Sử dụng chuyển mạch số BCD, thực thi tính toán của (A - B) và hiển thị kết quả trên màn hình số BCD (Y40 đến Y4F).



Hiển thị kết quả tính toán từ A - B trên màn hình BCD từ Y40 to Y4F. Khi kết quả là một số âm, đảm bảo ăng mà hình hiển thị 0 và LED của Y70 được bật.

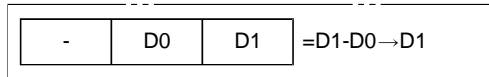
Điền vào chỗ trống [ ] .

Sau đó, kiểm tra hoạt động sử dụng thiết bị Demo.



Gợi ý

Kết quả hoạt động luôn là đầu ra từ CPU dạng nhị phân.



- 1) \_\_\_\_\_
- 2) \_\_\_\_\_
- 3) \_\_\_\_\_
- 4) \_\_\_\_\_

Tên dự án	QTEST9
Tên chương trình	MAIN

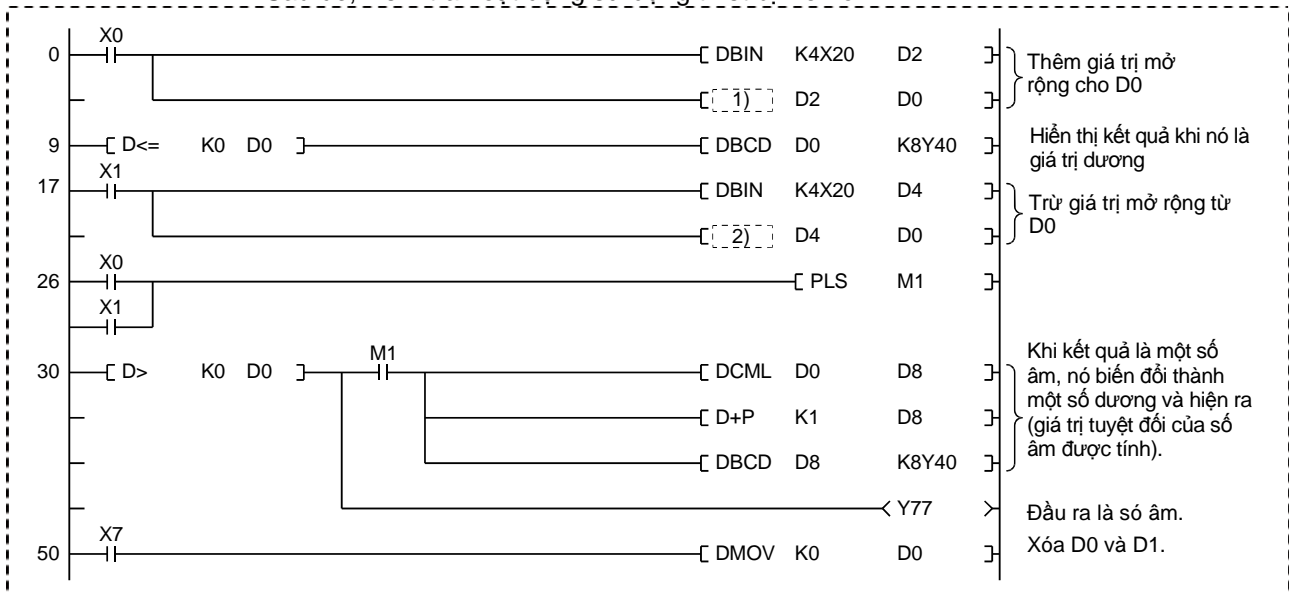
5.7.5 **Bài tập 5** Lệnh cộng và trừ

Tạo một chương trình:

- Nhập giá trị xác định bởi chuyển mạch số (X20 đến X2F) đến D3 và D2 (dữ liệu 32 bit) khi X0 được bật, thêm nó vào D1 và D0, và hiển thị kết quả trên màn hình (Y40 đến Y5F).
- Nhập giá trị xác định bởi chuyển mạch số (X20 đến X2F) đến D5 và D4 khi X1 được bật, lấy D1 và D0 trừ nó đi, và hiển thị kết quả.
- Khi kết quả là số âm, Y77 được bật, hai phần bù được quyết định từ kết quả để lấy giá trị tuyệt đối, và hiển thị.

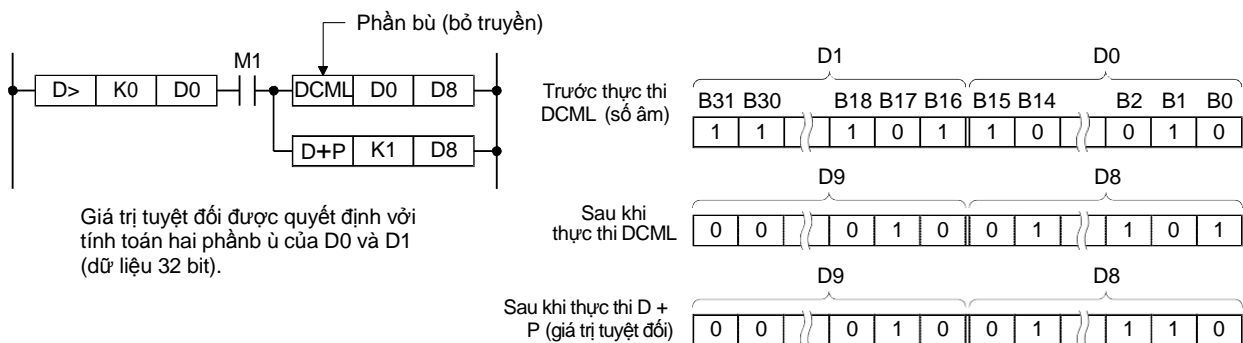
Điền vào chỗ trống: [ ] dưới đây.

Sau đó, kiểm tra hoạt động sử dụng thiết bị Demo.



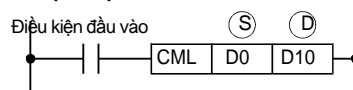
- \_\_\_\_\_
- \_\_\_\_\_

**Tham khảo**



**Ghi nhớ**

Lệnh CML dùng để đảo ngược bit mẫu của (S) và truyền dữ liệu này đến (D) Khi điều kiện đầu vào được bật..



Tên dự án	QTEST10
Tên chương trình	MAIN

5.7.6 Bài tập 6 Lệnh nhân và chia

Tạo một chương trình mà:

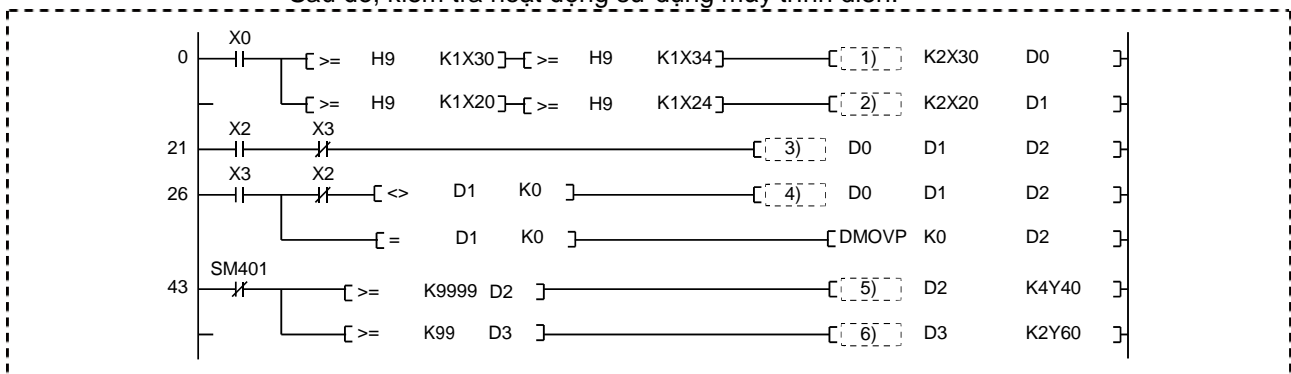
- Đặt dữ liệu cho nhân và chia khi X0 được bật.
- Nhân giá trị được xác định bởi chuyển mạch số X20 đến X27 bởi giá trị xác định bởi chuyển mạch số X30 đến X37 ở dạng nhị phân khi X2 được bật.
- Chia giá trị xác định bởi chuyển mạch số X30 đến X37 bởi giá trị xác định bởi chuyển mạch số X20 đến X27 ở dạng nhị phân khi X3 được bật.
- Đầu ra là kết quả của phép nhân hoặc chia cho ra màn hình BCD hiển thị Y40 đến Y4F và phần dư cho ra màn hình BCD Y60 đến Y67.

$(X30 \text{ đến } X37) \times (X20 \text{ đến } X27) \triangleright (Y40 \text{ đến } Y4F)$

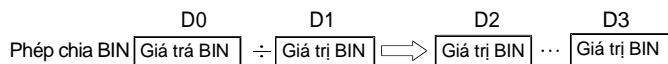
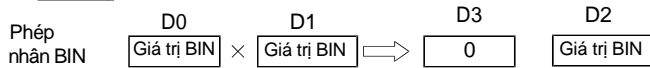
$(X30 \text{ đến } X37) / (X20 \text{ đến } X27) \triangleright (Y40 \text{ đến } Y4F) \dots (Y60 \text{ đến } Y67)$

Tạo một chương trình với GX Works2 điền vào ô trống [ ] dưới đây.

Sau đó, kiểm tra hoạt động sử dụng máy trình diễn.



Gợi ý



- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

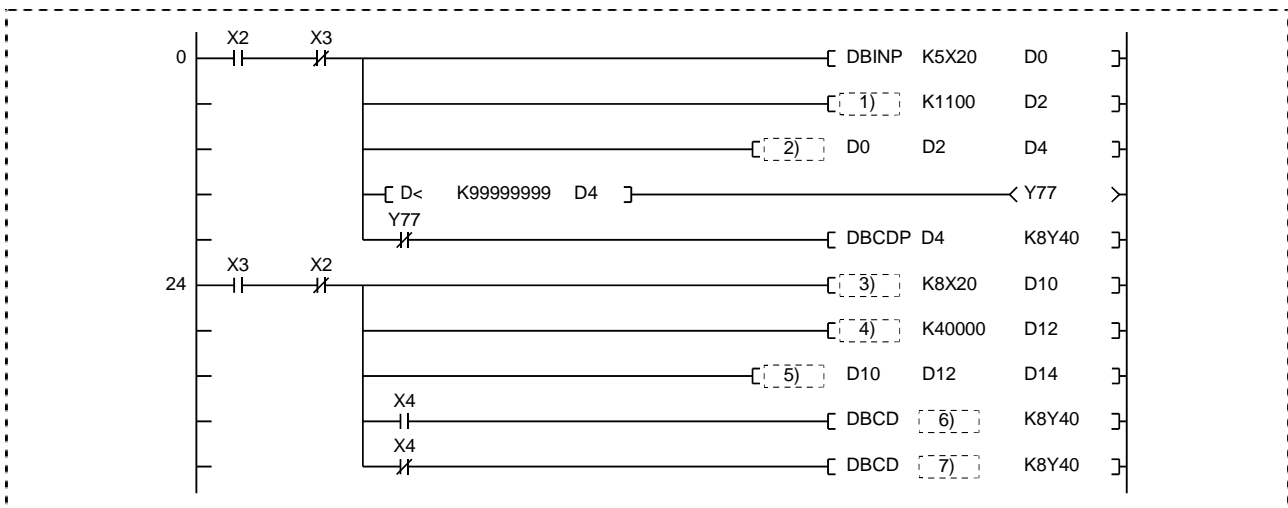
Tên dự án	QTEST11
Tên chương trình	MAIN

5.7.7 Bài tập 7 D-nhân và D-chia

Tạo chương trình mà:

- 1) Nhân giá trị thiết lập bởi chuyển mạch số 5 chữ số (X20 đến X33) bằng 1.100 dạng nhị phân khi X2 được bật. Khi kết quả là 99,999,999 hoặc ít hơn, nó được hiện trên màn hình (Y40 đến Y5F).
- 2) Chia giá trị thiết lập bởi chuyển mạch số 8 chữ số (Y40 đến Y5F) .  
 $(X20 \text{ đến } X33) \times 1100 \rightarrow (Y40 \text{ đến } Y5F)$   
 $(X20 \text{ đến } X3F) / 40000 \rightarrow$  Thương (Y40 đến Y5F) ... X4: ON  
 Phần dư (Y40 đến Y5F) ... X4: OFF

Tạo chương trình với GX Works2 điền vào chỗ trống [ ] dưới đây. Sau đó, kiểm tra hoạt động sử dụng thiết bị Demo.



- 1) \_\_\_\_\_
- 2) \_\_\_\_\_
- 3) \_\_\_\_\_
- 4) \_\_\_\_\_
- 5) \_\_\_\_\_
- 6) \_\_\_\_\_
- 7) \_\_\_\_\_

Đáp án cho bài tập chương 5

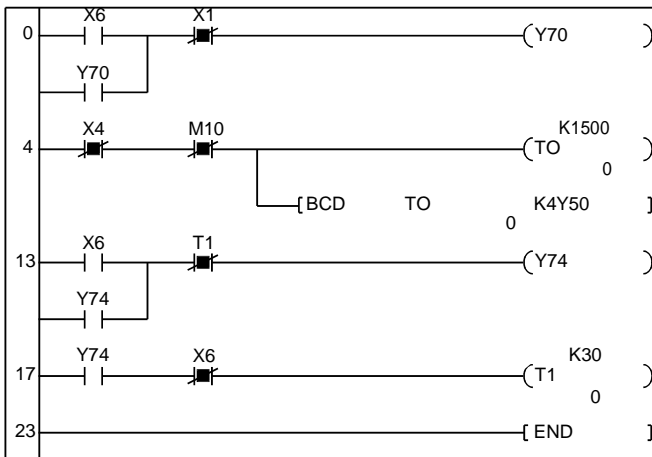
Bài tập Số.		Đáp án
1	1)	K2X0
	2)	K2Y70
2	1)	D0
	2)	K4X20
	3)	BCD
	4)	K4Y40
3	1)	K2Y40
	2)	K8
	3)	K0
4	1)	BINP
	2)	BINP
	3)	>
	4)	<=
5	1)	D + P
	2)	D - P
6	1)	BINP
	2)	BINP
	3)	*P
	4)	/P
	5)	BCD
	6)	BCD
7	1)	DMOV P
	2)	D*P
	3)	DBINP
	4)	DMOV P
	5)	D/P
	6)	D14
	7)	D16

## Chương 6 Làm thế nào để sử dụng các chức năng khác

### 6.1 Kiểm tra chức năng trực tuyến

Để chuẩn bị, làm theo hướng dẫn dưới đây.

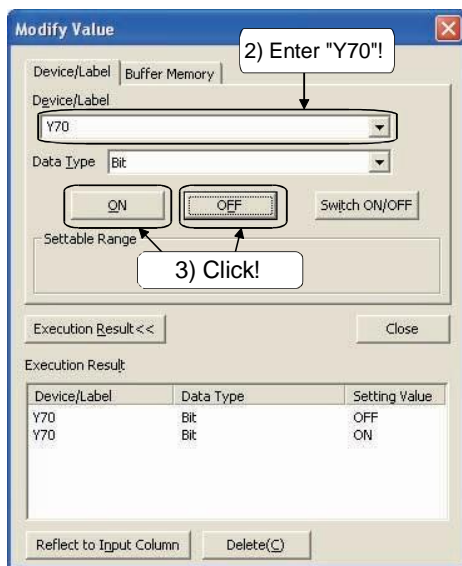
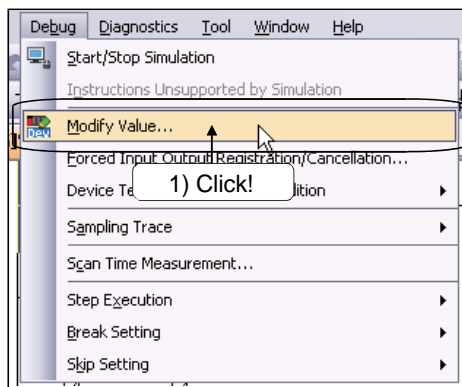
Tên dự án	QEX14
Tên chương trình	MAIN



Chi tiết về phương pháp hoạt động, xem lại chương 2.

- 1) Đọc một dự án với Works2.
- 2) Viết tham số và chương trình của dự án đang đọc vào CPU (bộ điều khiển khả trình). (CPU phải dừng lại.)
- 3) Đặt GX Works2 vào chế độ giám sát.
- 4) Xác nhận chương trình hiện trên màn hình.

### 6.1.1 Bật và tắt biến "Y" bằng cưỡng bức



Dừng CPU trước khi làm theo.

1) Click [Debug] → [Modify Value].

2) Hộp thoại Modify Value hiện ra. Nhập "Y70" vào hộp "Device/Label".

3) Click nút  hoặc  để bật hoặc tắt "Y70" bằng cưỡng bức.

#### Kiểm tra với bộ DEMO

1) Xác nhận trạng thái bật và tắt trên vùng Execution Result để biến đổi theo như click nút  hoặc . Thêm nữa, xác nhận LED của Y70 của thiết bị Demo bật và tắt theo hoạt động.

**Lưu ý**  
Khi CPU ở trạng thái RUN, kết quả hoạt động của chương trình được hiển thị theo độ ưu tiên. Vì thế, dừng CPU đầu tiên trước khi xác nhận với thiết bị Demo.



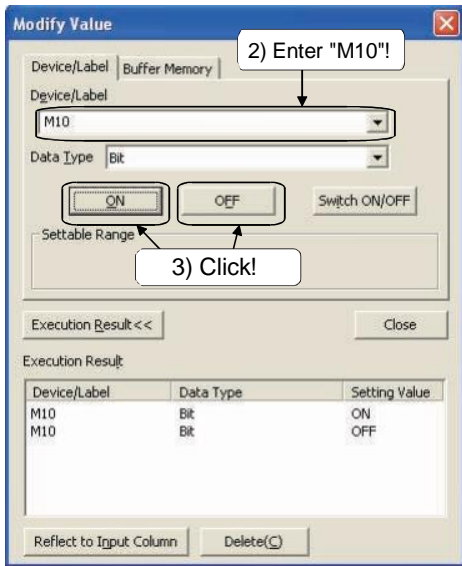
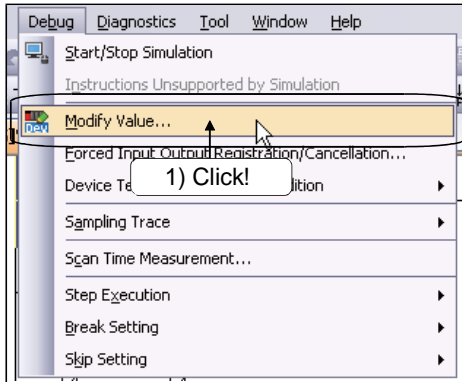
#### Gợi ý

Chức năng kiểm tra trong suốt quá trình giám sát ladder của GX Works2 cũng khả dụng cho cài đặt và khởi động lại công tắc, thay đổi giá trị hiện tại, và biến đầu ra bằng cường bức.

Click đúp vào một công tắc (nhấn vào nút  ) và giữ phím  ở trong màn hình giám sát GX Works2 biến đổi công tắc mở và đóng bằng cường bức.

Để hiển thị hộp thoại Modify Value, click đúp vào một biến từ ( giữ phím  ) và giữ phím  ở trong màn hình giám sát ladder của GXWorks2.

### 6.1.2 Đặt và khởi động lại biến "M"

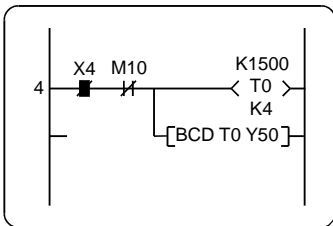


Kích hoạt CPU trước hoạt động này

- 1) Click [Debug] → [Modify Value].
- 2) Hộp thoại Modify hiện ra. Nhập "M10" trong hộp danh sách "Device/Label".
- 3) Click nút  ON hay  OFF để đặt hoặc khởi động lại "M10".

#### Kiểm tra bộ DEMO

Tắt X4 và kiểm tra dưới đây.



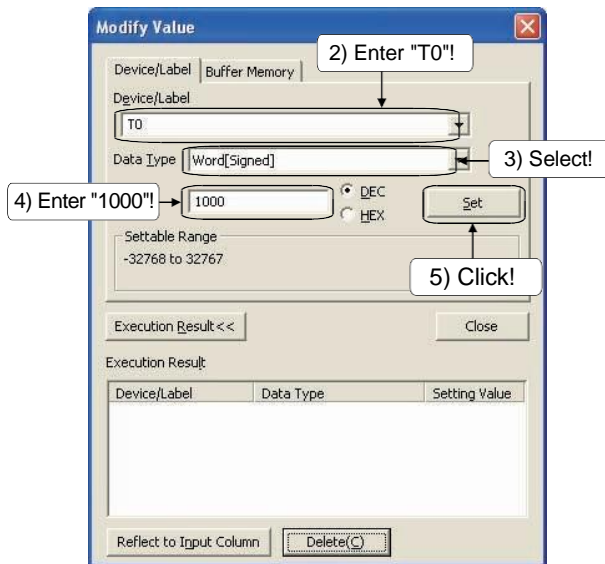
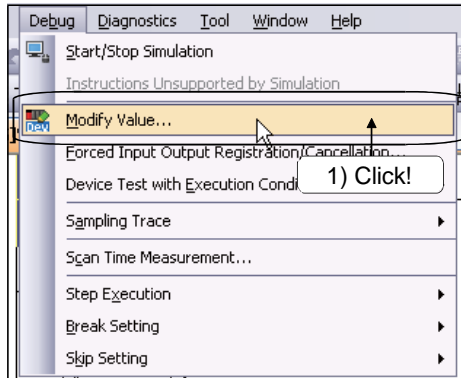
(Màn hình giám sát khi M10 được đặt)

- 1) Khi M10 được đặt,  $\overline{M10}$  trở thành không dẫn và giá trị hiện tại của bộ định thời T0 được xóa về 0. Kiểm tra giá trị trên màn hình số (Y50 đến Y5F) không đổi.
- 2) Khi M10 được đặt,  $M10$  dẫn và bộ định thời T0 bắt đầu đếm từ 0. Giá trị đếm tăng mỗi 10s. Xác nhận giá trị hiện trên màn hình (Y50-Y5F) tăng mỗi 10s.

#### Gợi ý

Với quy trình tương tự, thiết bị bit khác rơ le nội (M) có thể được đặt hoặc khởi động lại về 0.

### 6.1.3 Đổi giá trị hiện tại của biến "T"



Kích hoạt CPU trước khi hoạt động

1) Click [Debug] → [Modify Value].


2) Hộp thoại Modify Value hiện ra. Nhập "T0" vào hộp "Device/Label".

3) Chọn "Word[Signed]" từ danh sách "Data Type".

4) Nhập "1000" vào cột "Value"..

5) Sau khi cài đặt hoàn thành, click vào nút **Set** để thay đổi giá trị hiện tại của T0 đến 1000 bằng cưỡng bức.

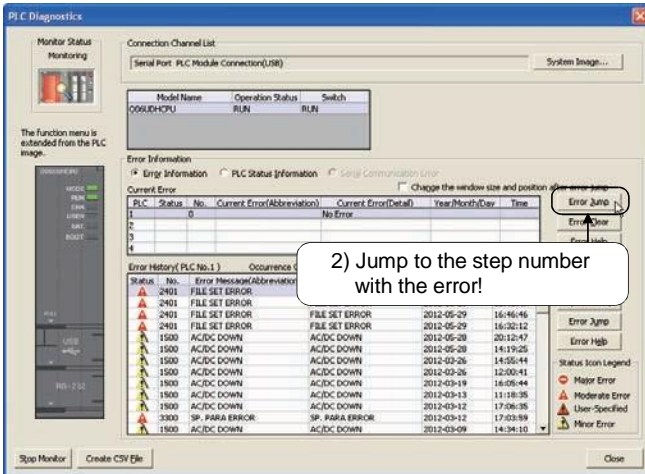
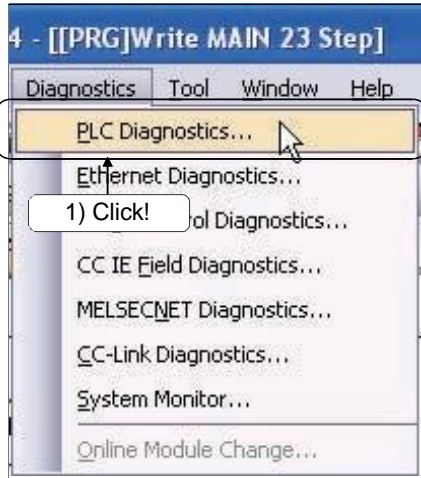
#### Kiểm tra với thiết bị Demo

1) Xác định giá trị trên màn hình số (Y50-Y5F) là 1000 khi khi  khóa được nhấn.

#### Gợi ý

Với quy trình tương đương, giá trị hiện tại của biến từ khác bộ định thời (T) cũng có thể thay đổi.

### 6.1.4 Đọc các bước lỗi



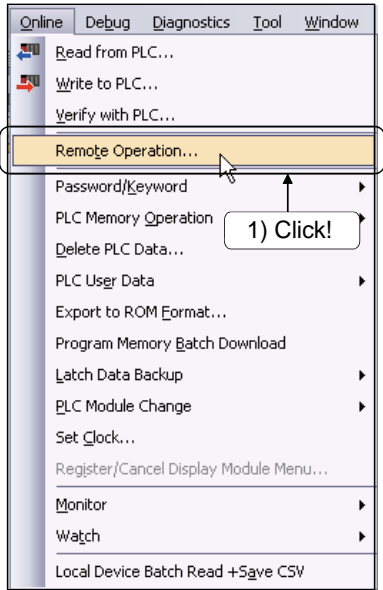
Kích hoạt CPU trước hoạt động này

1) Click [Diagnostics] → [PLC Diagnostics].

2) Hộp thoại The PLC Diagnostics hiện ra. Click nút **Error JUMP** để nhảy đến số bước chương trình tuần tự được đánh dấu ở nơi xảy ra lỗi được chọn

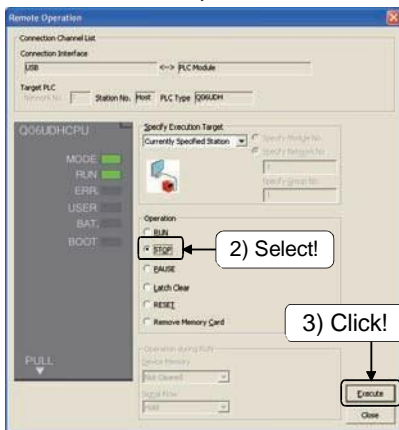
- Số thự tự lỗi hiện ra nếu có lỗi xuất hiện.
- "No Error" hiện ra nếu không có lỗi.

6.1.5 STOP và RUN từ xa



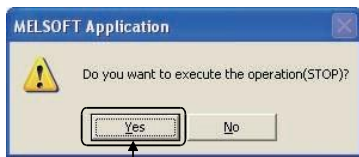
Kích hoạt CPU trước hoạt động này.

1) Click [Online] → [Remote Operation].



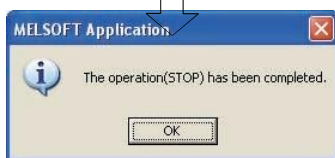
2) Hộp thoại Remote Operation hiện ra. Chọn "STOP" trong danh mục trong vùng Operation.

3) Sau khi cài đặt được hoàn thành, click nút **Execute** .



4) Tin nhắn "Do you want to execute the operation(STOP)?" hiện ra. Click nút **Yes** .

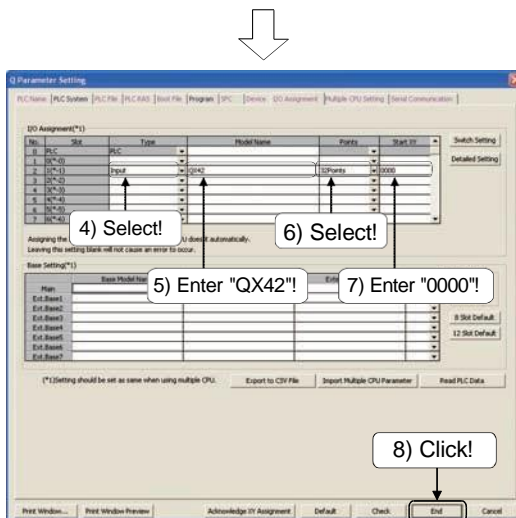
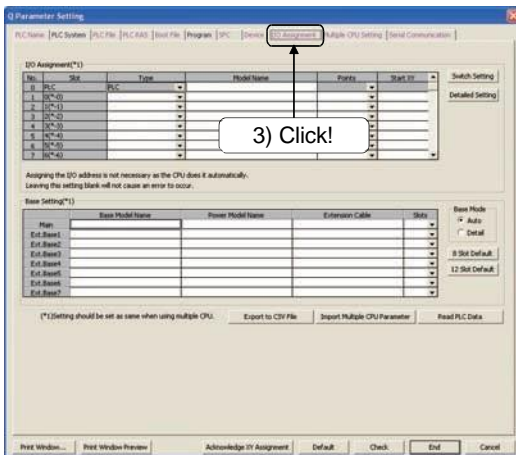
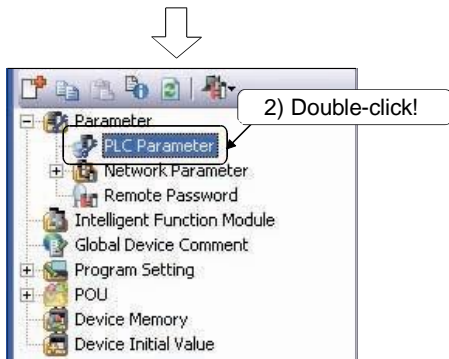
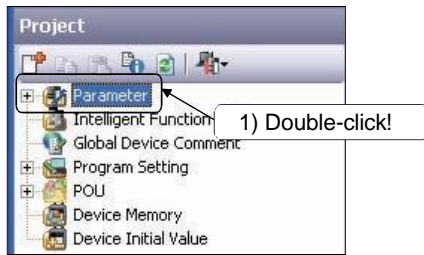
Hoạt động của CPU dừng lại



5) Chọn "RUN" trong bước 2), và thực hiện lại bước 2) đến 4) .

CPU, được dừng lại ở hoạt động trên, bắt đầu hoạt động lại

## 6.2 Cường bức chỉ định I/O bằng cài đặt Parameter



1) Click đúp "Parameter" ở danh mục dự án.

2) "PLC Parameter", thư mục "Network Parameter", và "Remote Mật khẩu" hiện ra. Click đúp vấp "PLC Parameter".

3) Hộp thoại Q Parameter Setting hiện ra. Click vào thanh "I/O Assignment".

4) Chọn "Input" từ danh sách trong cột "Type".

5) Nhập "QX42" vào cột "Model Name".


6) Chọn "32Điểm" từ danh mục của cột "Điểm".

7) Nhập "0000" trong cột "Start XY".

8) Sau khi cài đặt được hoàn thành, click **End**.

Sau khi bài tập này được hoàn thành, khởi động cài đặt bằng quy trình dưới đây.

1) Click nút **Default** trong hộp thoại Q Parameter Setting để khởi động cài đặt tham số.

2) Click  trên thanh công cụ và chỉ viết tham số lên CPU.

**Kiểm tra với thiết bị Demo**

Dừng CPU và click  trên thanh công cụ.

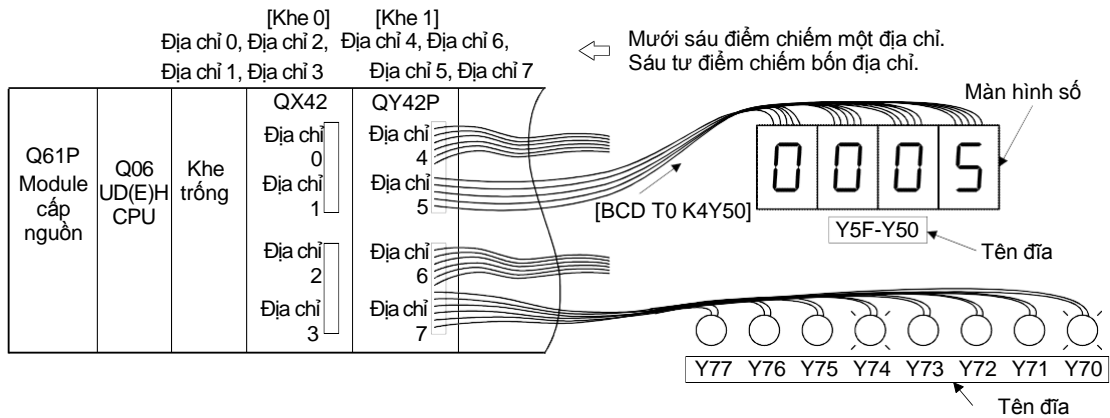
Hộp thoại Online Data Operation hiện ra. Click vào tham số của dữ liệu bị chỉnh sửa hiện tại, và click nút **Execute** để chỉ viết tham số lên CPU.

Sau đó, kích hoạt CPU và kiểm tra dưới đây.

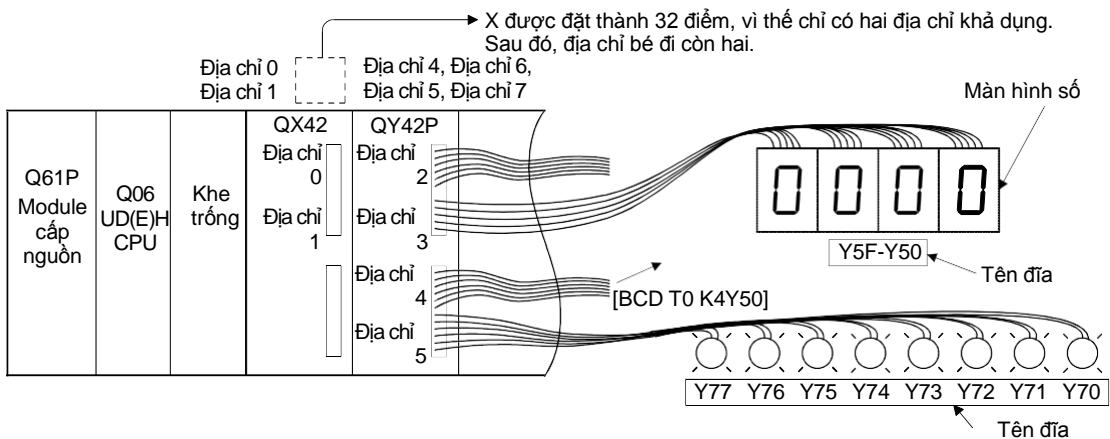
1) Giá trị hiện tại của biến mất từ màn hình số (Y50 đến Y5F). Sau đó, LED của Y70 đến Y77 bắt đầu nháy sáng cho đến khi giá trị thiết lập của Y70 đến Y77 đến mỗi giá trị biến được đặt.

2) Bật X6 để đưa tín hiệu đến Y70 và Y74 không làm tắt LED của Y70 và Y74.

**[Số I/O trước khi cưỡng bức chỉ định]**



**[Số I/O sau khi cưỡng bức chỉ định]**



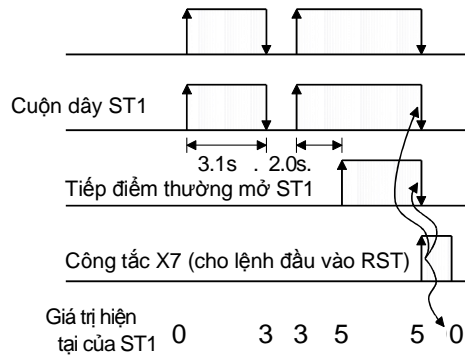
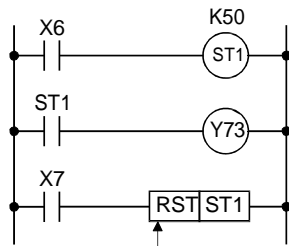
**Gợi ý**

- Địa chỉ số 7 được thay thế bằng địa chỉ số 5. Vì thế, giá trị hiện tại của bộ định thời được đưa đến đầu ra địa chỉ vừa chỉ định 5, và LED của Y70 đến Y77 được kết nối đến đèn địa chỉ số 5.
- Kết quả của đưa tín hiệu đầu ra đến Y70 hoặc Y74 không được hiển thị trên bất cứ màn hình nào vì địa chỉ số 7 của module đầu ra không còn tồn tại.
- Để hiển thị số biến địa chỉ thông thường, thay đổi số biến K4Y50 ⇨ K4Y30, và Y70 đến Y77 ⇨ Y50 đến Y57.

### 6.3 Làm thế nào để sử dụng bộ định thời có nhớ

Khi điều kiện đầu vào được bật, cuộn dây được cấp điện. Sau đó giá trị của bộ định thời có nhớ bắt đầu tăng lên. Khi giá trị hiện tại đến giá trị đặt, bộ định thời có nhớ không đếm nữa và công tắc bật lên. Khi điều kiện đầu vào bị tắt trong quá trình tăng lên, cuộn dây bị ngắt điện nhưng giá trị hiện tại vẫn được giữ. Sau đó điều kiện đầu vào được bật trở lại, cuộn dây được nạp lại được cấp điện và giá trị hiện tại được tích lũy.

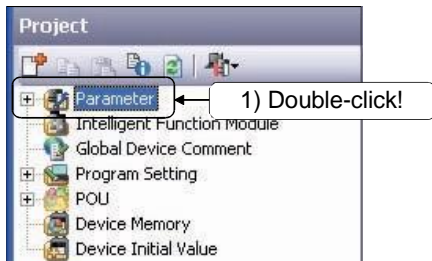
Tên dự án	Bộ đếm thời gian tích hợp lưu trữ
Tên chương trình	MAIN



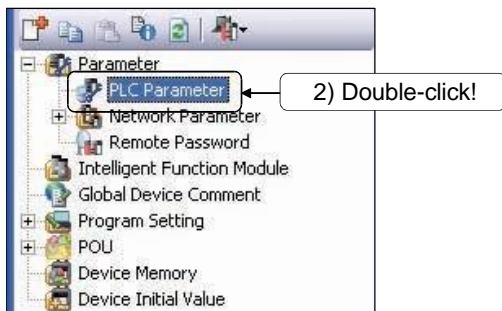
Khi sử dụng chương trình như bộ đếm thời gian tích hợp lưu trữ, xác định điểm của tham số trước.

Chỉ có lệnh RST là có thể tắt công tắc và xóa giá trị hiện tại sau khi bộ đếm thời gian tích hợp lưu trữ dừng lại.

Trong ví dụ dưới đây, bộ đếm thời gian tích hợp lưu trữ được đặt từ ST0 đến ST31.



1) Click đúp "Parameter" trong danh sách dự án.

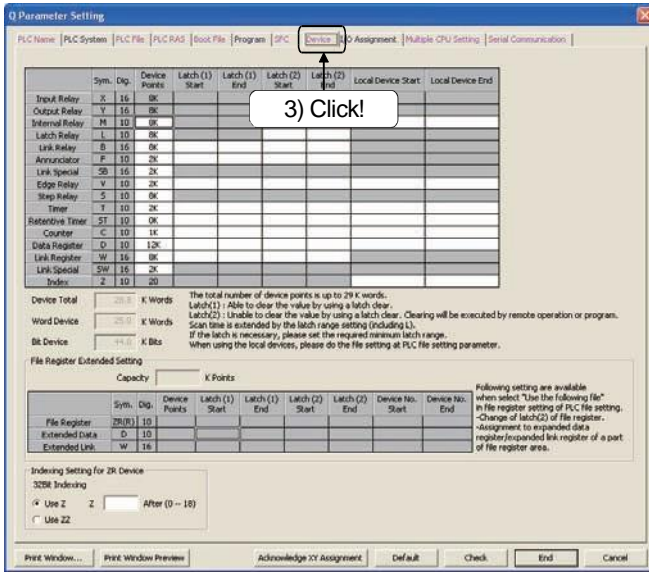


2) "PLC Parameter", "Network Parameter" folder, và "Remote Mật khẩu" hiện ra. Click đúp "PLC Parameter".

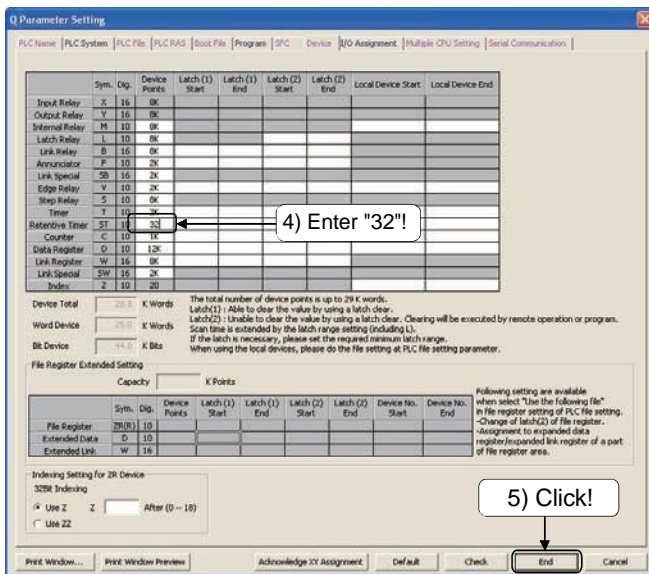
(Đến trang sau)



(Từ trang trước)



3) Hộp thoại Q Parameter Setting hiện ra. Click thanh "Device".



4) Click "Device Point" trong hàng "Retentive Timer", và nhập "32".

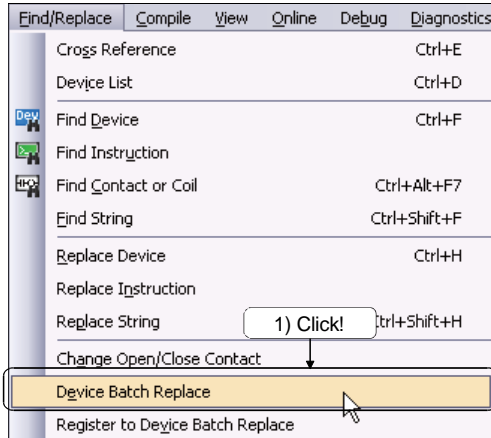
5) Sau khi cài đặt hoàn tất, click nút **End**.

5) Click!

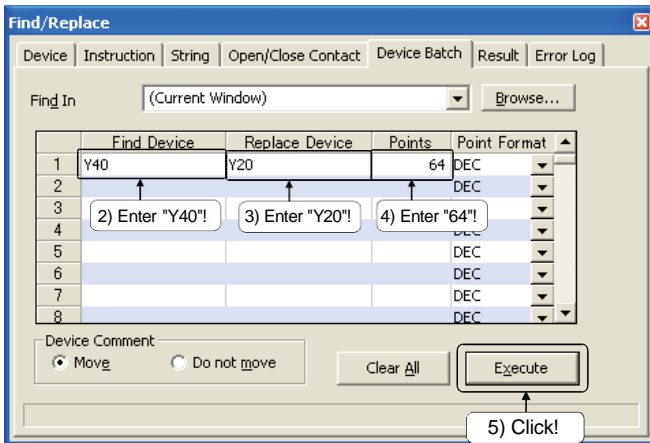
## 6.4 Thay thế loạt biến

### 6.4.1 Thay thế loạt biến

Phần này diễn giải làm thế nào để thay thế Y40 đến Y7F (64 biến) với Y20 đến Y5F (64 biến) theo lô.



1) Click [Find/Replace] → [Device Batch Replace].

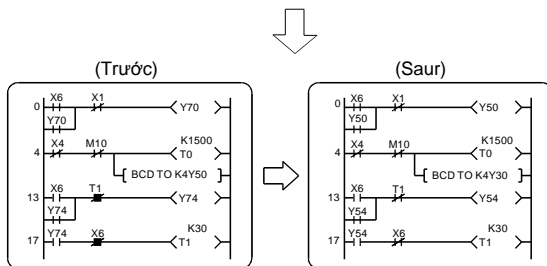


2) Hộp thoại Find/Replace hiện ra. Nhập "Y40" ở cột "Find Device".

3) Nhập "Y20" ở cột "Replace Device"..

4) Nhập "64" vào cột "Point".

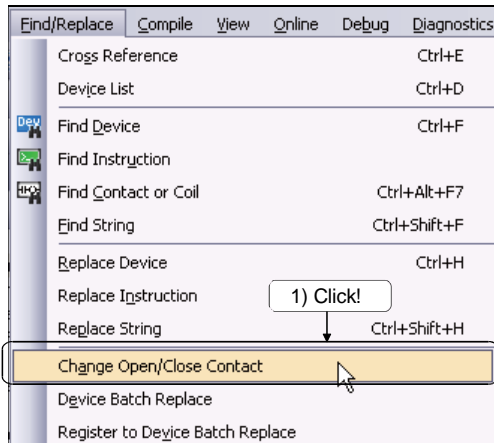
5) Sau khi cài đặt hoàn tất, click nút **Execute**.



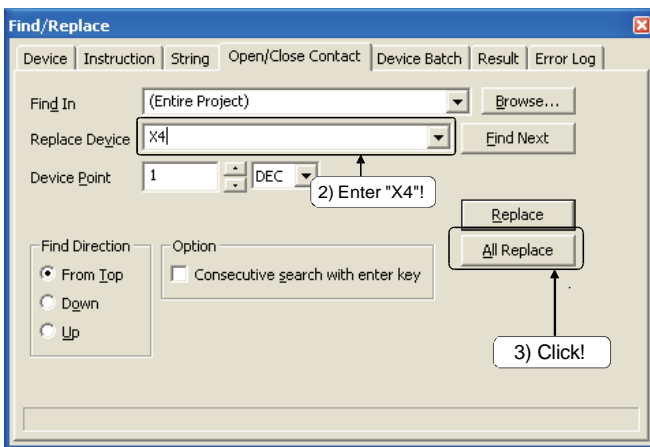
6) Xác nhận rằng số biến mục tiêu được thay thế.

## 6.4.2 Thay đổi loại biến được xác định giữa tiếp điểm thường mở và tiếp điểm thường đóng

Phần này diễn giải làm thế nào để thay đổi tiếp điểm thường mở của biến xác định thành tiếp điểm thường đóng và ngược lại.

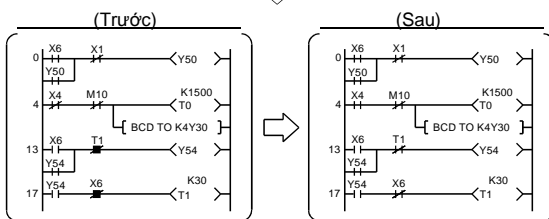


1) Click [Find/Replace] → [Change Open/Close Contact].



2) Hộp thoại Find/Replace hiện ra. Nhập "X4" trong danh mục "Replace Device".

3) Sau khi cài đặt hoàn tất, click nút **All Replace**.



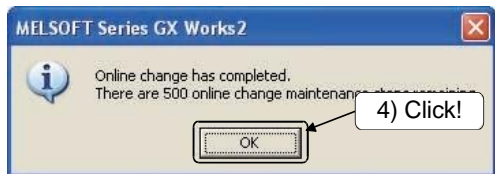
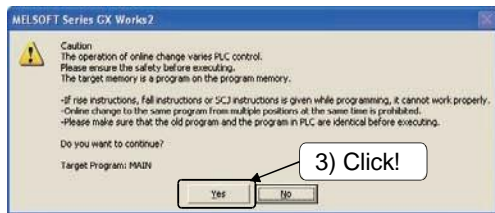
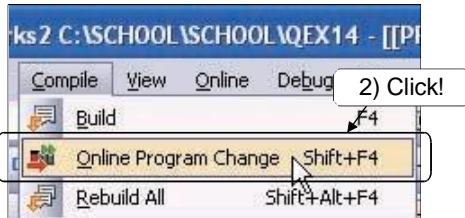
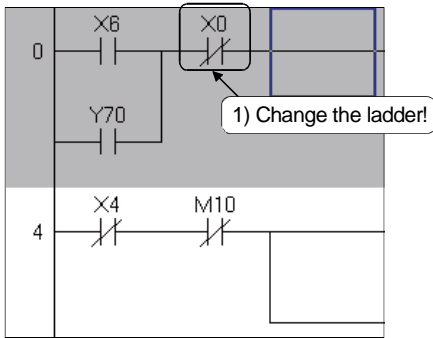
4) Xác nhận tiếp điểm thường mở được chuyển thành tiếp điểm thường đóng và ngược lại.

### Ghi chú

Trước khi luyện tập phần 6.5 sau phần này, viết chương trình trong máy tính cá nhân lên CPU.  
Đối với hoạt động viết, xem lại phần 2.7.

## 6.5 Thay đổi chương trình trực tuyến

Chức năng này dùng để viết chương trình lên CPU đang chạy.



Kích hoạt CPU trước hoạt động này

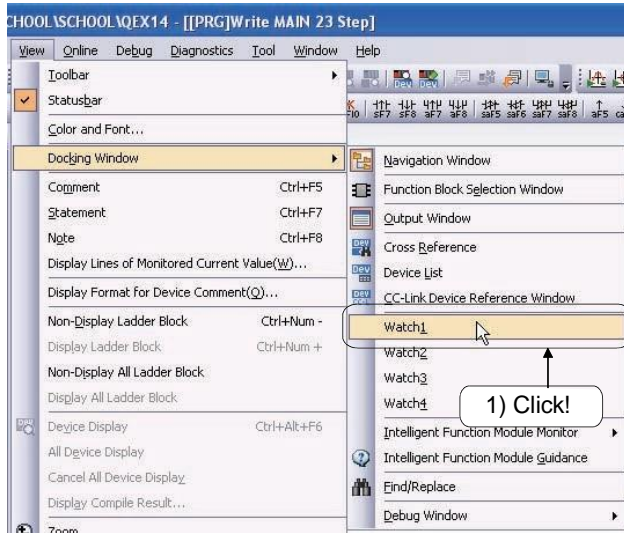
- 1) Thay đổi ladder.  
(Trong ví dụ, thay đổi "X1" thành "X0".)
- 2) Sau khi thay đổi, click [Compile] → [Online Program Change].
- 3) Hộp thoại "Caution" hiện ra. Click nút  để chấp nhận thay đổi.
- 4) Tin nhắn "Online change has completed." hiện ra. Click nút .

### Ghi chú

Thay đổi chương trình trực tuyến không thể thực thi khi chương trình trong CPU điều khiển khả trình và chương trình trong GX Works2 trước khi thay đổi không giống nhau. Vì thế, khi không biết chương trình giống hoặc không giống, kiểm tra chúng trước khi thay đổi với GX Works2, và thực thi thay đổi chương trình trực tuyến.

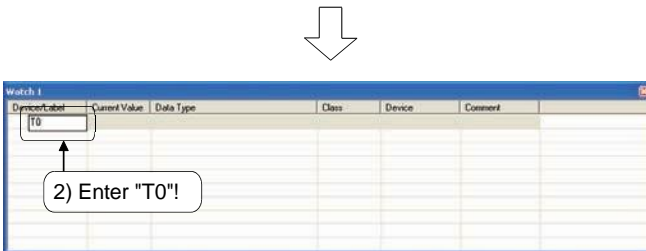
## 6.6 Đăng kí biến

Phần này diễn giải làm thế nào để đăng kí nhiều biến hoặc nhấn trong một màn hình và giám sát chúng cùng lúc..

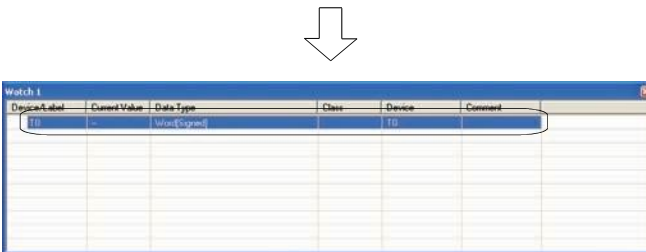


- 1) Click [View] → [Docking Window] → [Watch(1 to 4)].

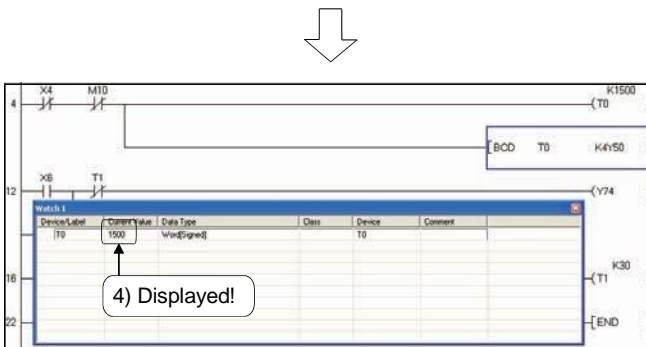
\* Trong ví dụ này, chọn "1".



- 2) Cửa sổ Watch 1 hiện ra. Chọn một hàng để chỉnh sửa. Nhập "T0" trong cột Device Label.



- 3) Biến đầu vào hoặc nhấn được đăng kí.



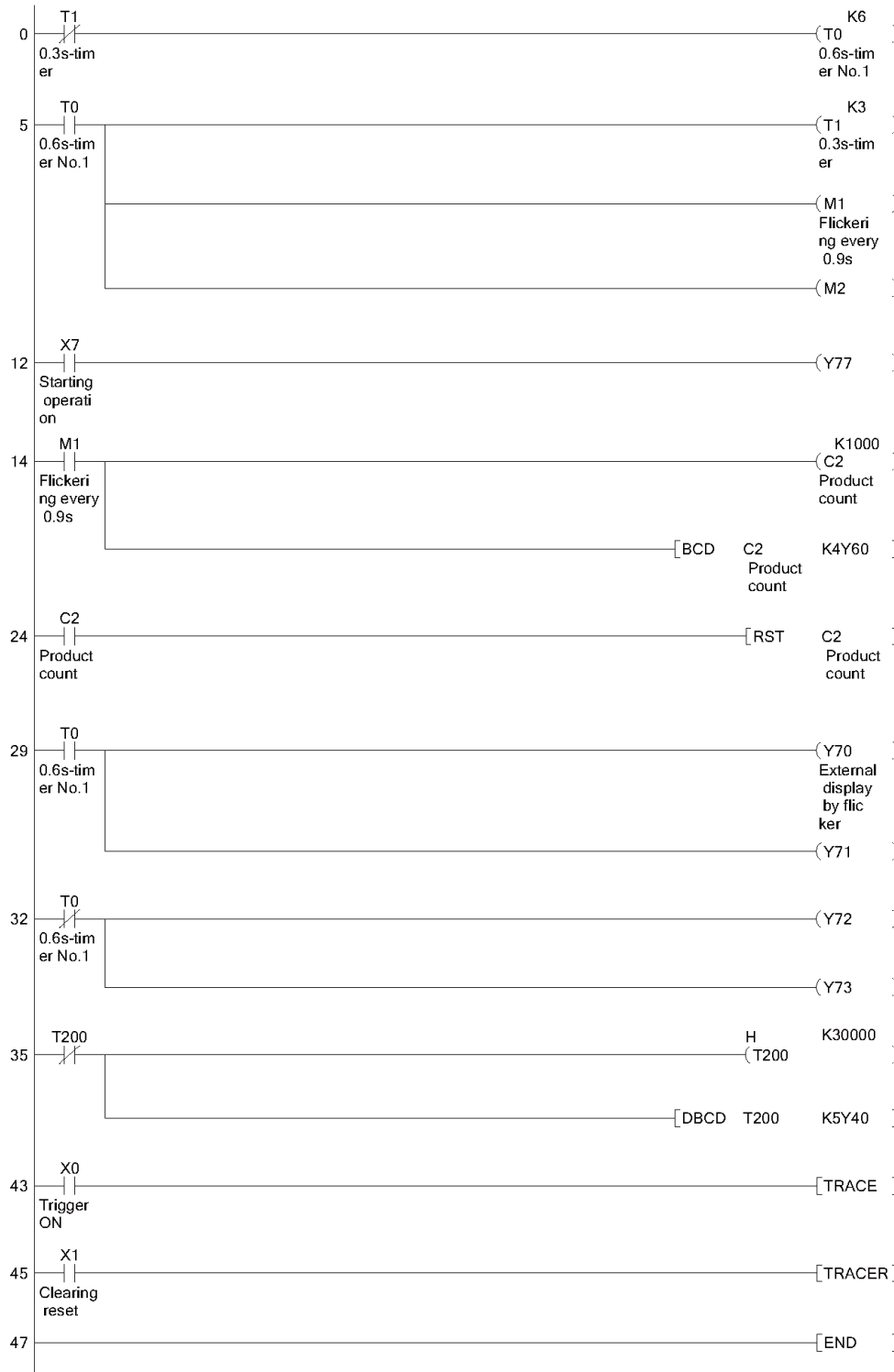
- 4) Click [Online] → [Watch] → [Start Watching].

Giá trị hiện tại của biến đăng kí hoặc nhấn được hiện ra trên cửa sổ.

### 6.7 Làm thế nào để tạo chú thích

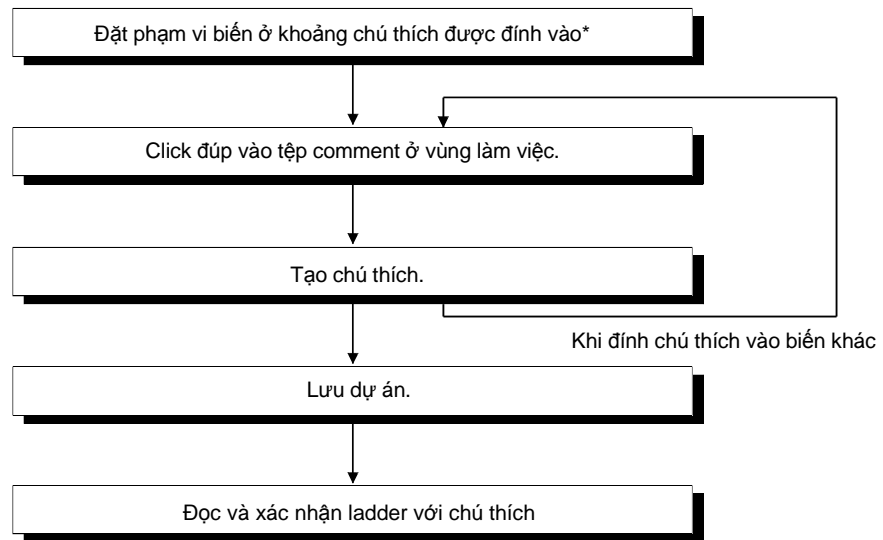
Tên dự án	QEX15
Tên chương trình	MAIN

Dưới đây là ví dụ của việc in ra ladder với chú thích.



Sử dụng bàn phím để nhập chương trình trên hoặc đọc nó từ một thư mục trên màn hình máy tính.

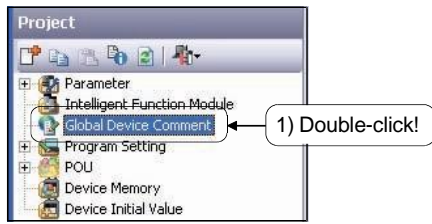
(1) Biểu đồ hoạt động khi tạo chú thích



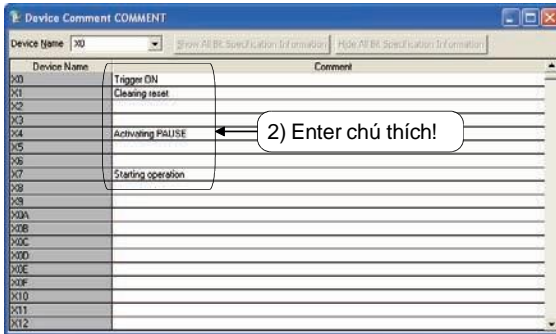
\*: Quy trình này cần thiết cho việc xác định phạm vi ghi chú thiết bị.

Gợi ý
Chú thích được sử dụng để hiển thị chức năng hoặc ứng dụng của mỗi biến Cho phép lên đến 32 ký tự

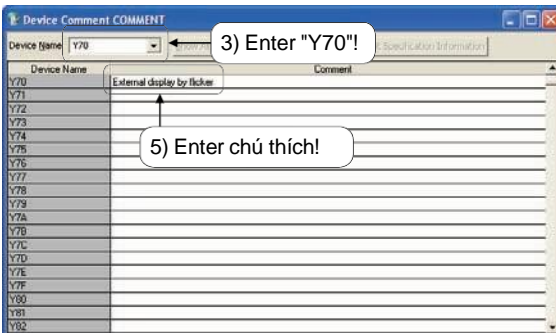
## (2) Tạo chú thích



- 1) Click đúp vào "Global Device Comment" trong danh sách dự án. Màn hình Device Comment hiện ra.



- 2) Click vào vùng chú thích và nhập một chú thích như bên trái.



- 3) Nhập "Y70" vào danh sách "Device Name" .

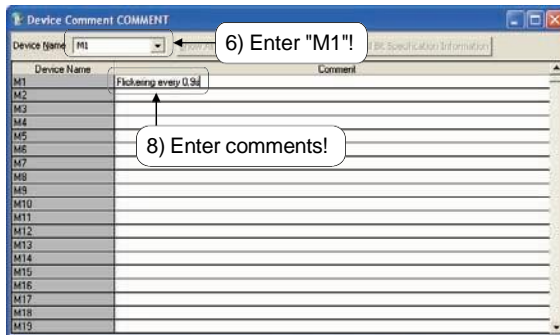
- 4) Nhấn nút  .

- 5) Click vào vùng chú thích và nhập một chú thích như bên trái.

(Đến trang sau)



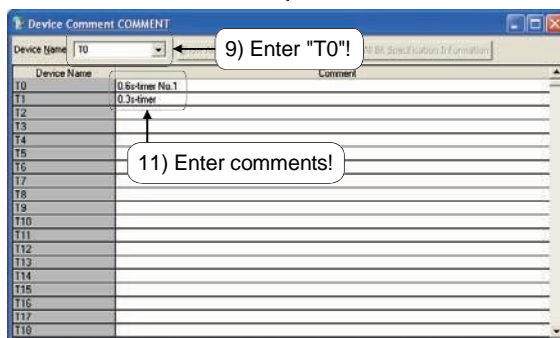
(Từ trang trước)



6) Nhập "M1" vào danh sách "Device Name".

7) Nhấn nút  .

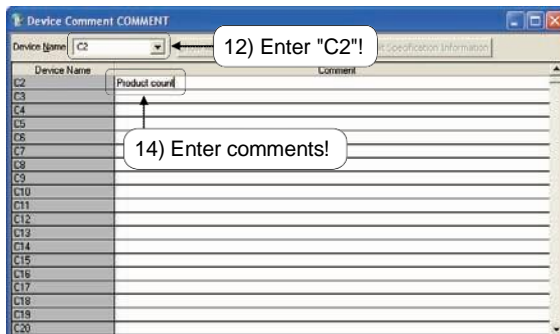
8) Click vào vùng chú thích và nhập một chú thích như bên trái.



9) Enter "T0" in the "Device Name" list box.

10) Nhấn nút  .

11) Click vào vùng chú thích và nhập một chú thích như bên trái.

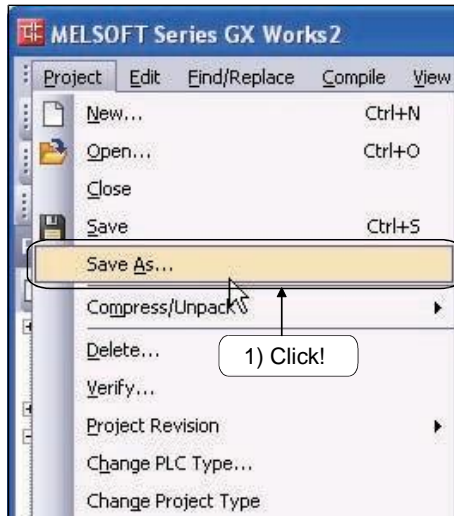


12) Nhập "C2" vào danh sách "Device name".

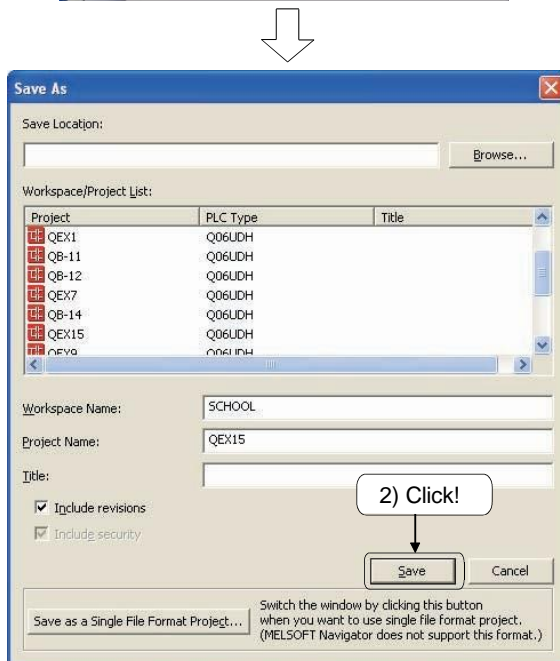
13) Nhấn nút  .

14) Click vào vùng chú thích và nhập một chú thích như bên trái.

### (3) Lưu chú thích



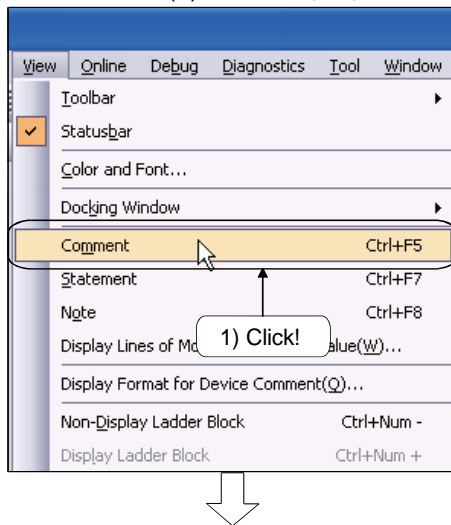
1) Click [Project] → [Save As].



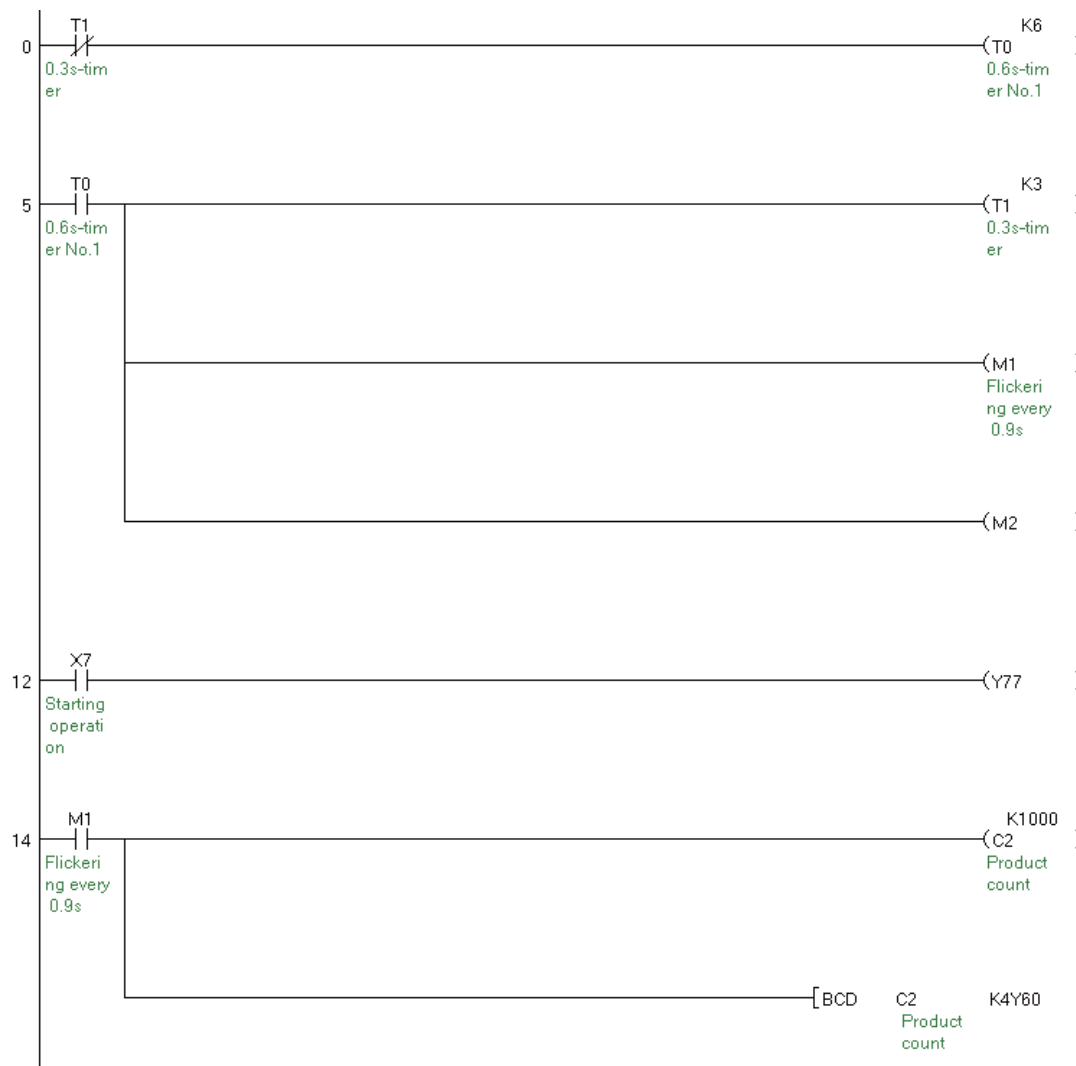
2) Hộp thoại Save As hiện ra. Xác định (hoặc chọn) một tên vùng làm việc và click nút **Save** .

(4) Hiển thị một ladder với chú thích trên màn hình GX Works2

1) Click [View] → [Comment].



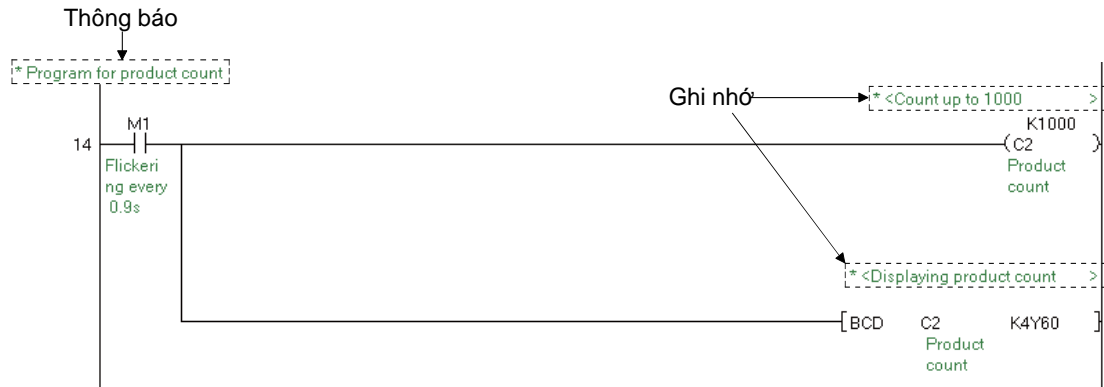
2) Chú thích được hiển thị trên màn hình ladder.



## Gợi ý

Cùng với ghi chú thiết bị, thông báo và ghi nhớ có thể tạo trên màn hình ladder.

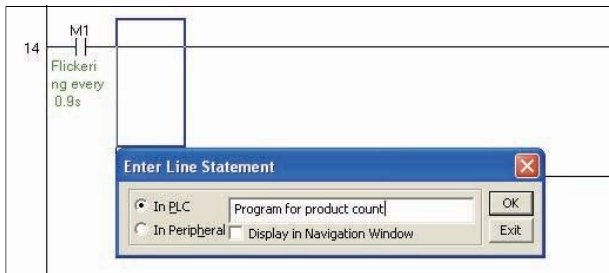
- Thông báo : Chú thích để diễn giải chức năng hoặc ứng dụng cho khối ladder. Có thể lên đến 64 ký tự.
- Ghi nhớ : Chú thích để diễn giải chức năng hoặc ứng dụng cho đầu ra và lệnh. Cho phép lên đến 32.



### • Tạo thông báo

Click và click đúp vào một biểu tượng nơi một chú thích được đính vào.

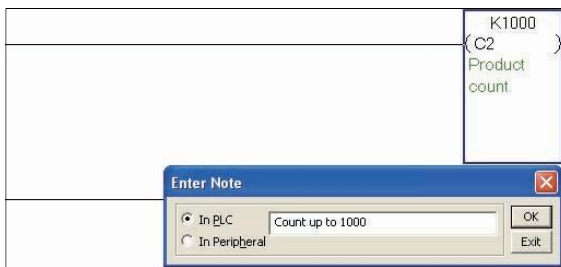
Hộp thoại Enter Line Statements hiện ra. Nhập vào một chú thích sau đó click nút  .



### • Tạo ghi nhớ

Click và click đúp vào một biểu tượng nơi một chú thích được đính vào.

Hộp thoại Enter Symbol hiện ra. Nhập vào một chú thích sau đó click nút  .



### • Chọn "In PLC" hoặc "In Peripheral" cho thông báo và ghi nhớ.

"In PLC"

: Dữ liệu của thông báo và ghi nhớ được lưu như một phần của chương trình. Nó cho phép dữ liệu được lưu trong CPU ở nhà máy. Tuy nhiên, cần rất nhiều dung lượng bộ nhớ chương trình của khối điều khiển khả trình.

"In Peripheral"

: Dữ liệu của thông báo và ghi nhớ được lưu trong thiết bị ngoại vị (máy tính cá nhân) tách biệt từ chương trình. Vì mỗi chương trình cần một bước bổ sung với mỗi địa điểm, cần ít dung lượng bộ nh chương trình trên khối điều khiển khả trình CPU. Tuy nhiên, khi chương trình được chỉnh sửa ở nhà máy, chương trình trên GX Works2 trong thiết bị ngoại vị (máy tính cá nhân) và chương trình trong khối điều khiển khả trình không giống nhau. Cần thận khi làm việc dữ liệu này.

## 6.8 Cài đặt bảo mật cho dự án

Phần này diễn giải làm thế nào để đặt bảo mật cho dự án để bảo vệ dự án và những dữ liệu trong dự án.

Đặt bảo mật hạn chế truy nhập cho dự án.

Thêm nữa, cài đặt bảo mật cấm những dữ liệu như POUs, ghi chú thiết bị, và tham số, được tạo ra bởi người dùng, từ sự chỉnh sửa hoặc tiết lộ lỗi cho những người dùng không có thẩm quyền.

Gợi ý	
<p>Cấp truy nhập và có quyền truy nhập                      Đặt cấp truy nhập cho mỗi người dùng để hạn chế truy nhập vào từng dữ liệu.                      Cấp truy nhập là một hoạt động trao quyền cho mỗi người dùng xem dự án.                      Năm cấp dưới đây khả dụng cho mỗi cấp truy nhập. Dữ liệu có thể được chỉnh sửa bởi một người dùng có quyền truy nhập thấp thì cũng sẽ có thể được chỉnh sửa bởi người dùng có mức truy nhập vào hơn.</p>	
Access level	Operating authority
Cao ↑ ↓ Thấp	Administrators <Cấp Administrator> Tất cả hoạt động đều được
	Developers (Level 3) <Cấp người phát triển>
	Developers (Level 2) Cài đặt bảo mật, truy nhập dữ liệu và một phần hoạt động bị giới hạn
	Developers (Level 1)
Users <Cấp điều hành> Chỉ truy nhập được vào dữ liệu dự án Dữ liệu không được đọc từ CPU điều khiển khả trình	
<p>&lt;Ví dụ&gt;                      Dữ liệu với quyền truy cập của người phát triển (cấp 2) có thể được chỉnh sửa bởi người dùng với nhà phát triển (cấp 2) hoặc cao hơn (Administrators, nhà phát triển (cấp 3), hoặc nhà phát triển (cấp 2)).</p>	

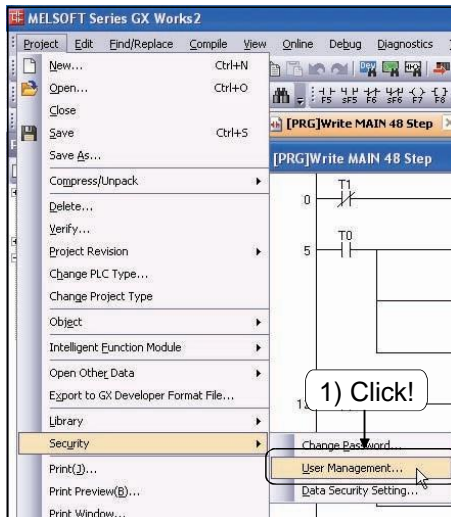
### 6.8.1 Cài đặt và đặt lại bảo mật cho dự án

Phần này diễn giải làm thế nào để đặt bảo mật cho một dự án mở và làm thế nào để đặt lại bảo mật.

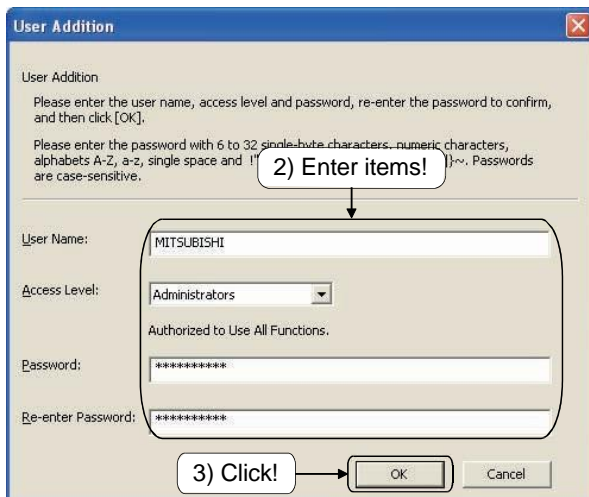
#### (1) Đặt bảo mật cho dự án

Đặt bảo mật cho một dự án.

Khi bảo mật được đặt cho dự án, cần có quyền người dùng khi dự án được mở lại .



1) Click [Project] → [Security] → [User Management].



2) Hộp thoại Use Addition hiện ra. Nhập những thứ dưới đây.

User Name : MITSUBISHI  
Mật khẩu : MITSUBISHI  
Re-enter Mật khẩu : MITSUBISHI

\* Khi người dùng hoặc mật khẩu đăng nhập bị mất, đăng nhập vào dự án là không thể. Không được nhập tên người dùng hoặc mật khẩu khác những cái trên.

3) Sau khi nhập chúng,click nút **OK** .

Bảo mật được đặt cho dự án.

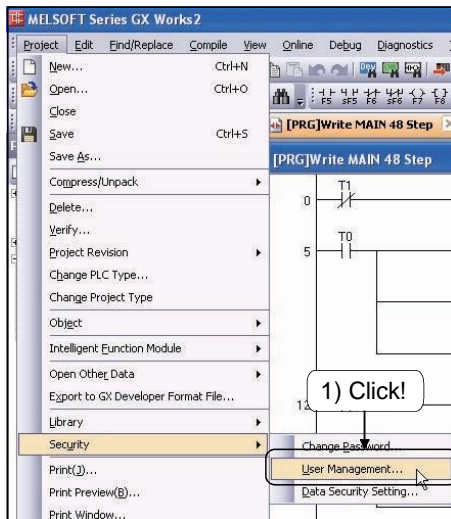
#### (2) Đặt lại bảo mật cho dự án

Xóa tất cả người dùng đặt lại bảo mật của một dự án và quay trở lại dự án mà không cần bảo mật (xem lại phần 6.8.2.)

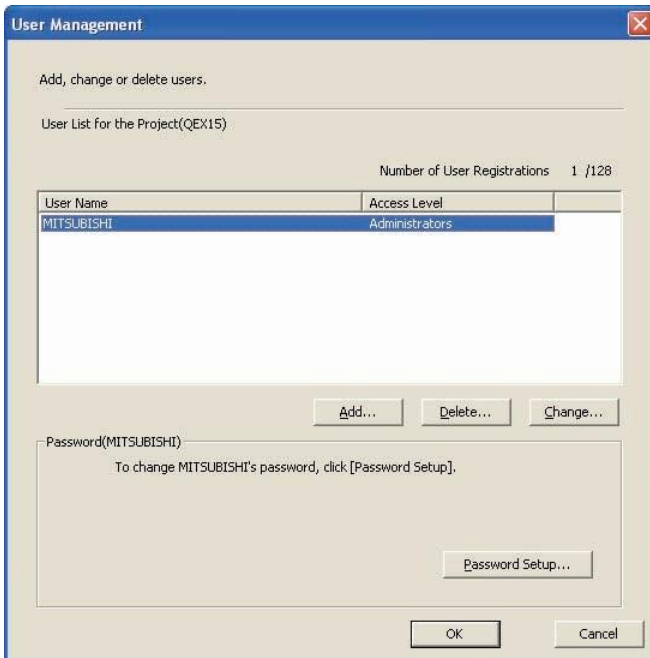
## 6.8.2 Quản lí (thêm, xóa, và thay đổi) người dùng

Phần này diễn giải làm thế nào để quản lí trạng thái đã được đăng kí của người dùng cho một dự án với bảo mật và làm thế nào để thêm, xóa và thay đổi người dùng. Chức năng này chỉ khả dụng khi người dùng đăng nhập vào một dự án với quyền truy cập "Administrators" hoặc "Developers".

[Hiện thị màn hình User Management ]



- 1) Click [Project] → [Security] → [User Management].



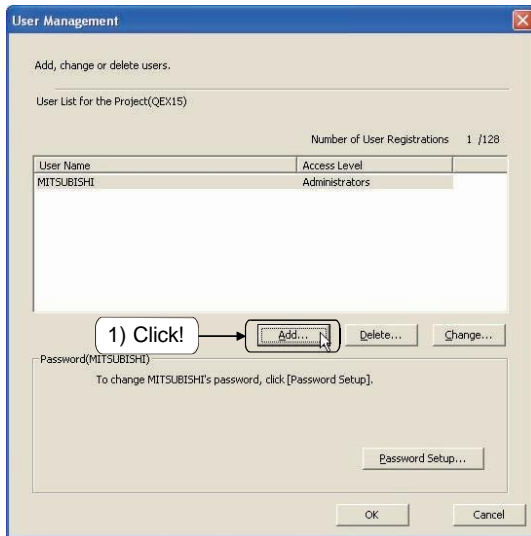
- 2) Hộp thoại User Management hiện ra.

Phương pháp cho thêm người dùng, thay đổi thông tin người dùng, thay đổi mật khẩu, và xóa người dùng được diễn giải ở trang sau.

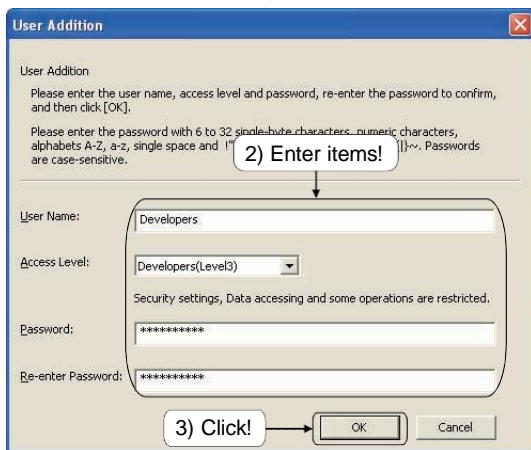
### [Thêm người dùng]

Thêm một người dùng vào một dự án với bảo mật.

Một người dùng có quyền truy cập cao hơn người dùng đang truy cập không thể thêm vào.



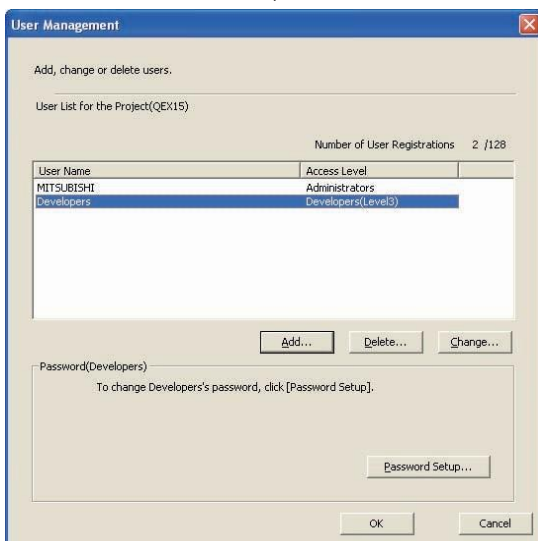
- 1) Click nút **Add...** Trên màn hình User Management.



- 2) Hộp thoại User Addition hiện ra. Nhập vào thông tin dưới đây.

User Name : Developers  
Access Level : Developers(Level3)  
Mật khẩu : Developers  
Re-enter Mật khẩu : Developers

- 3) Sau khi nhập chúng,click nút **OK** .



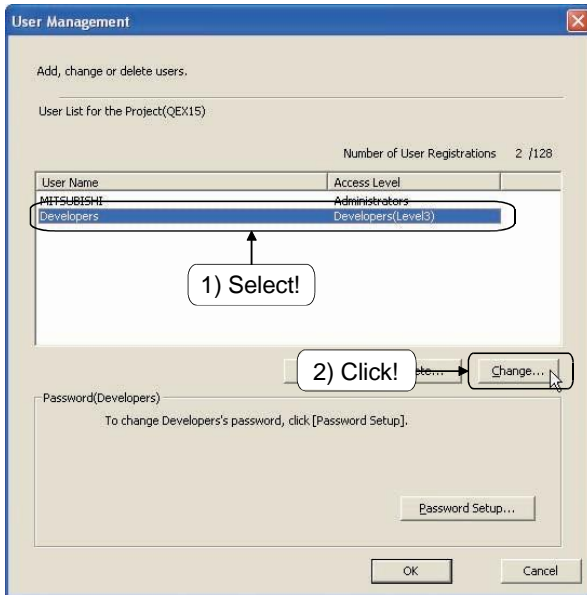
- 4) Người dùng (Developers(Level3)) được thêm.



[Thay đổi thông tin người dùng]

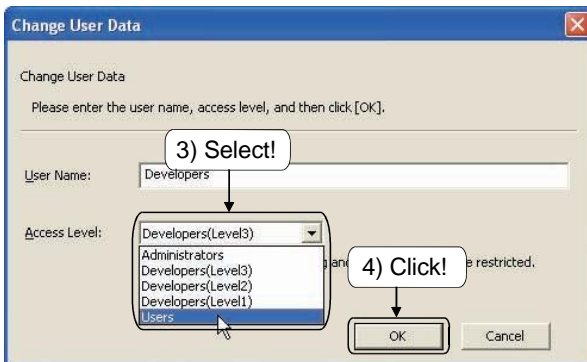
Thay đổi cấp truy nhập của người dùng được thêm vào từ trang trước "Developers(Level3)" vào "Users".

Thông tin của người dùng đăng nhập và của một người dùng có quyền truy nhập cao hơn người dùng đăng nhập là không thể thay đổi.



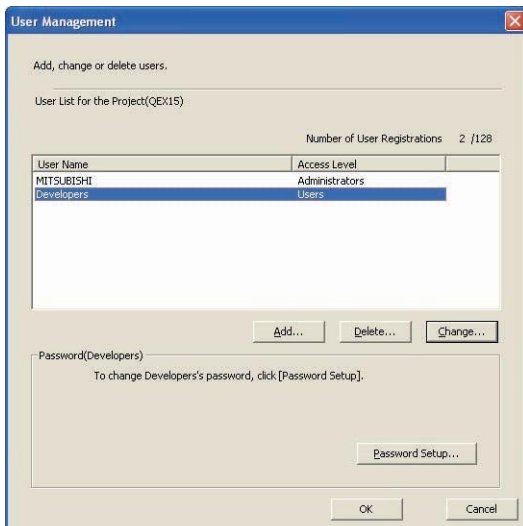
1) Chọn tên người dùng "Developers".

2) Click nút **Change...**.



3) Hộp thoại Change User Data hiện ra. Chọn "Users" từ danh sách "Access Level".

4) Sau khi chọn nó, click nút **OK**.

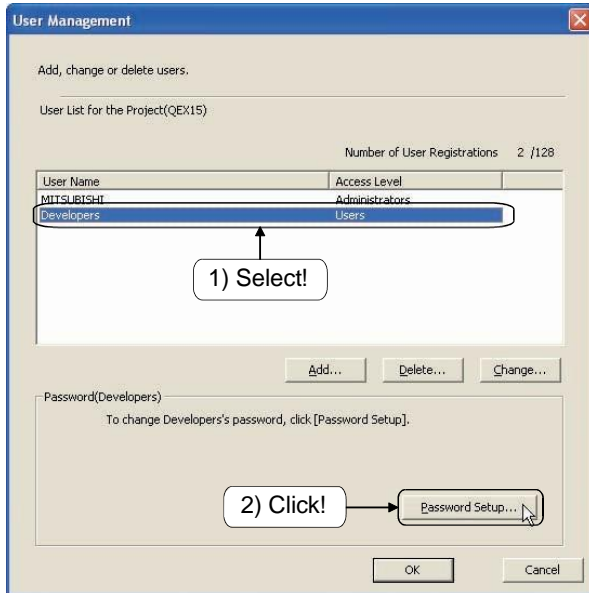


5) Cấp truy nhập của người dùng "Developers" bị thay đổi.

### [Đổi mật khẩu]

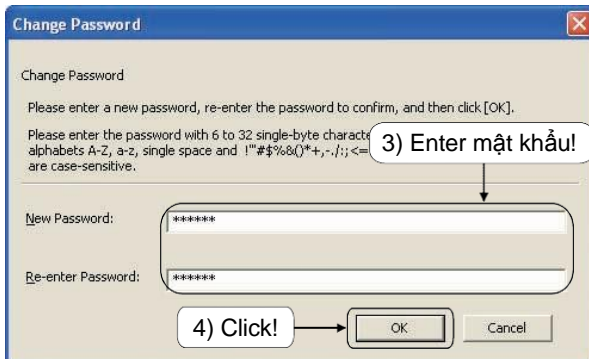
Thay đổi mật khẩu của một người dùng chọn trong danh sách trên màn hình User Management. Mật khẩu của người dùng đăng nhập và của một người dùng có cấp truy cập cao hơn người dùng đăng nhập là không thể thay đổi.

Để thay đổi mật khẩu của người dùng đăng nhập, click [Project] → [Security] → [Change Password].



1) Chọn tên người dùng "Developers".

2) Click nút **Password Setup** .



3) Hộp thoại Change Password hiện ra. Nhập những thông tin dưới đây.

New Password : Users1

Re-enter Password : Users1

4) Sau khi nhập chúng, click nút **OK** .

Mật khẩu của người dùng "Developers" bị thay đổi.

### [Xóa người dùng]

Xóa một người dùng được chọn trong danh sách với nút **Delete...** trên hộp thoại User Management.

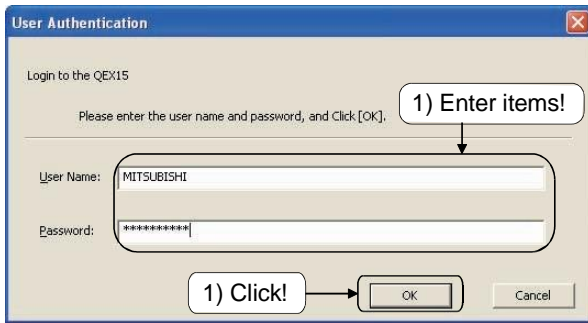
Người dùng đăng nhập hiện tại không thể xóa.

Tuy nhiên, khi người dùng đăng kí chỉ là "Administrators" và không có người dùng khác chuẩn bị xóa còn tồn tại, người dùng đăng nhập hiện tại có thể bị xóa.

Khi tất cả người dùng bị xóa, bảo mật được đặt lại.

### 6.8.3 Đăng nhập vào dự án

Xác thực người dùng là cần thiết cho mở một dự án với bảo mật.



- 1) Khi một dự án với bảo mật được mở, màn hình User Authentication hiện ra. Nhập người dùng và mật khẩu đăng nhập, và click nút  .

Nhập người dùng và mật khẩu dưới đây, được đặt trong phần 6.8.1.

User Name : MITSUBISHI  
Password : MITSUBISHI

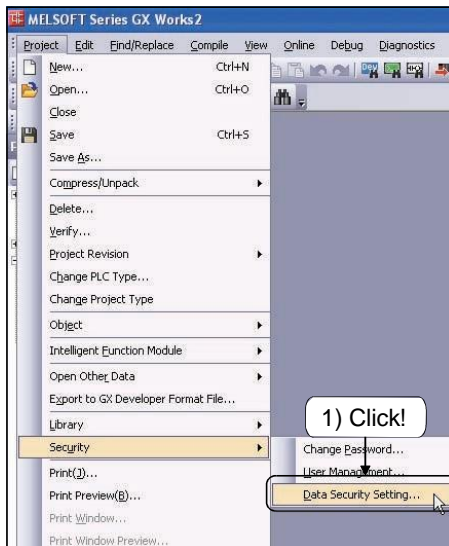
- 2) Dự án hiện ra.

### 6.8.4 Thay đổi xác thực truy nhập cho mỗi cấp truy nhập

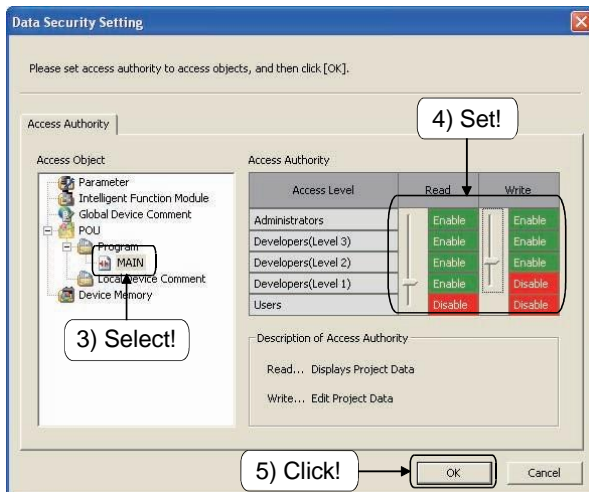
Phần này diễn giải làm thế nào để đặt một xác thực cho việc hiển thị và lưu dữ liệu cho mỗi cấp truy nhập.

Xác thực truy nhập của cấp cao hơn cấp của người dùng đăng nhập là không thể thay đổi.

Khi cấp truy nhập của người dùng đăng nhập hiện tại là "Users", xác thực truy nhập không thể thay đổi.



- 1) Click [Project] → [Security] → [Data Security Setting].



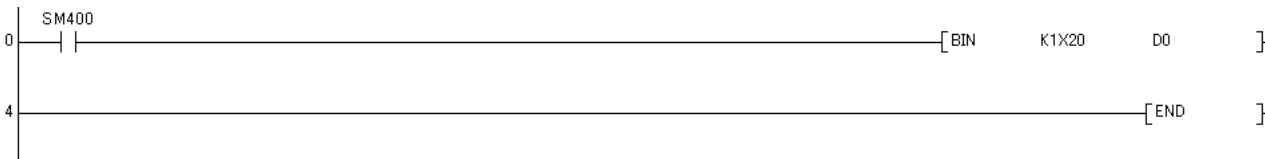
- 2) Hộp thoại Data Security Setting hiện ra.
- 3) Chọn mục tiêu từ Access Object.
- 4) Đặt "Enable" hoặc "Disable" để đọc và viết từ Access Authority cho mỗi cấp truy nhập bằng cách di chuyển con trượt.
- 5) Click nút  .

## 6.9 Chức năng lấy mẫu truy dấu

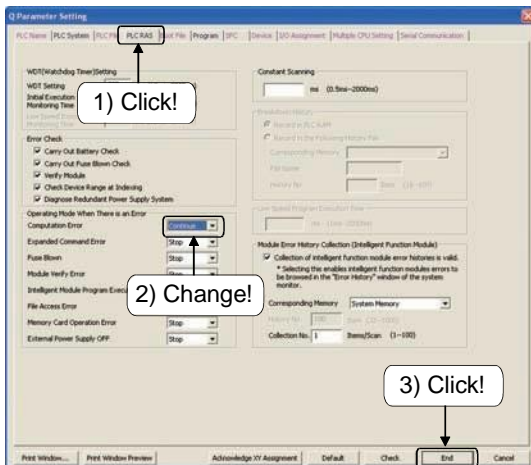
Chức năng này được sử dụng để lấy dữ liệu ở một thời điểm nhất định để tìm xem giá trị biến thay đổi như thế nào trong suốt quá trình chương trình hoạt động và truy dấu sự thay đổi hiện ra theo thời gian. Chi tiết của chức năng lấy mẫu truy dấu, xem lại hướng dẫn của mỗi module CPU.

Trong ví dụ này, giá trị biến tại điểm lỗi xảy ra được lấy.

Tên dự án	TRACE
Tên chương trình	MAIN

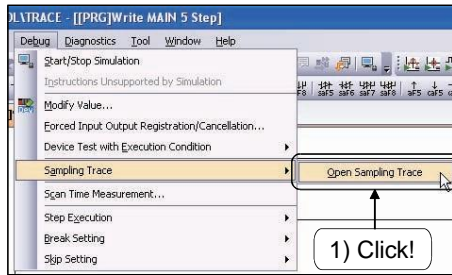


Để chuẩn bị, làm theo hướng dẫn dưới đây.

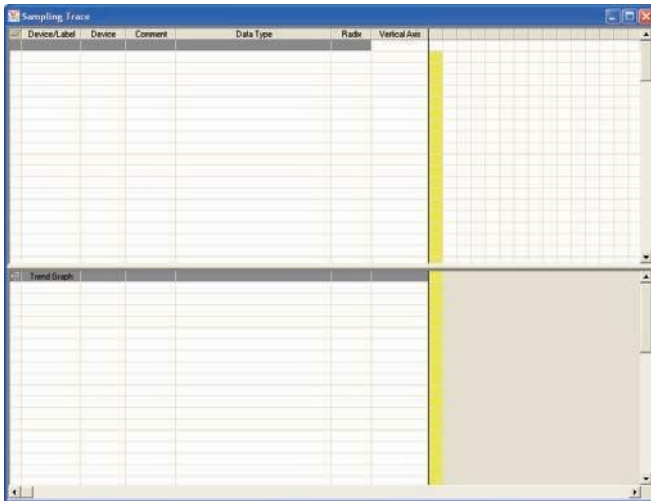


- 1) Click thanh "PLC RAS trên hộp thoại Q Parameter Setting.
- 2) Chọn "Continue" từ danh sách Computation Error trong Operating Mode When There is an Error area.
- 3) Click nút **End** .
- 4) Viết tham số và chương trình lên CPU.

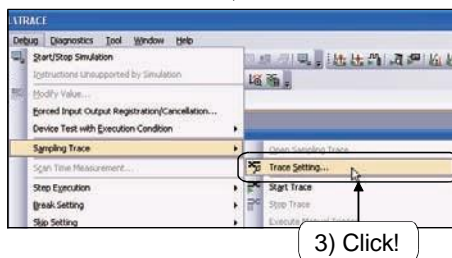
(1) Cài đặt lấy mẫu truy dấu



- 1) Click [Debug] → [Sampling Trace] → [Open Sampling Trace].



- 2) Màn hình Sampling Trace hiện ra.

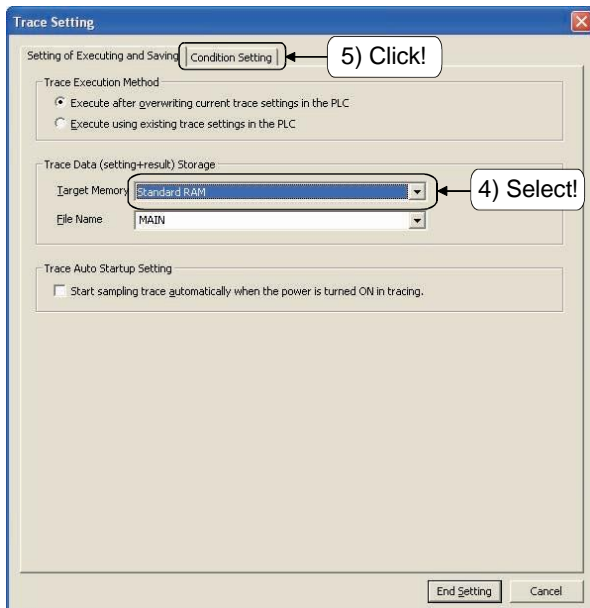


- 3) Click [Debug] → [Sampling Trace] → [Trace Setting].



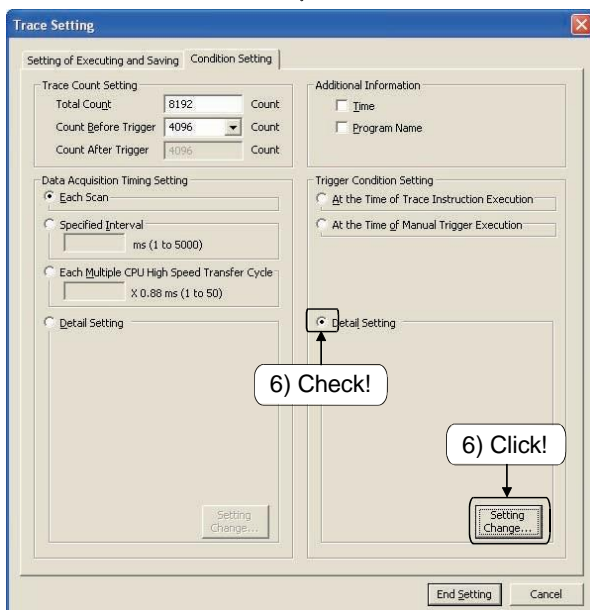
(Đến trang sau)

(Từ trang trước)



4) Hộp thoại Trace Setting hiện ra. Chọn "Standard RAM" từ danh sách "Target Memory".

5) Click vào thanh Condition Setting.

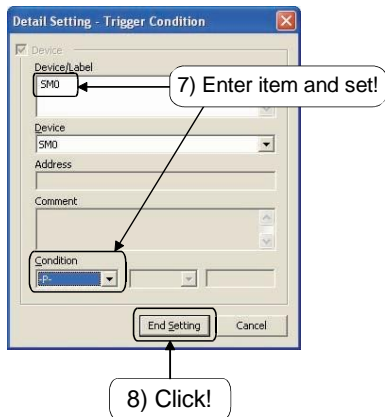


6) Kiểm tra "Detail Setting" ở trong vùng Trigger Condition Setting và click the **Setting Change** .



(Đến trang sau)

(Từ trang trước)

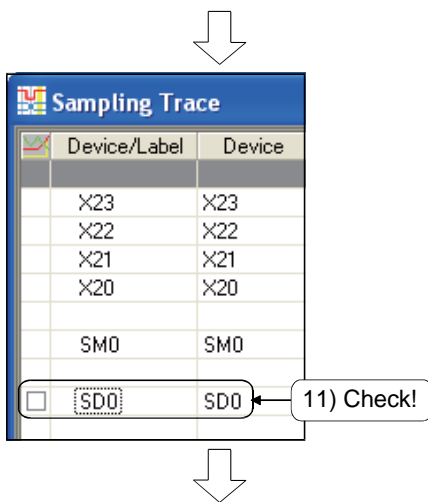


7) Hộp thoại Detail Setting - Trigger Condition hiện ra. Cài đặt những phần dưới đây  
Trong ví dụ này, đặt xuất hiện lỗi là điều kiện bắt đầu.

Device/Label : SM0  
Condition : -P-

8) Click nút **End Setting** .

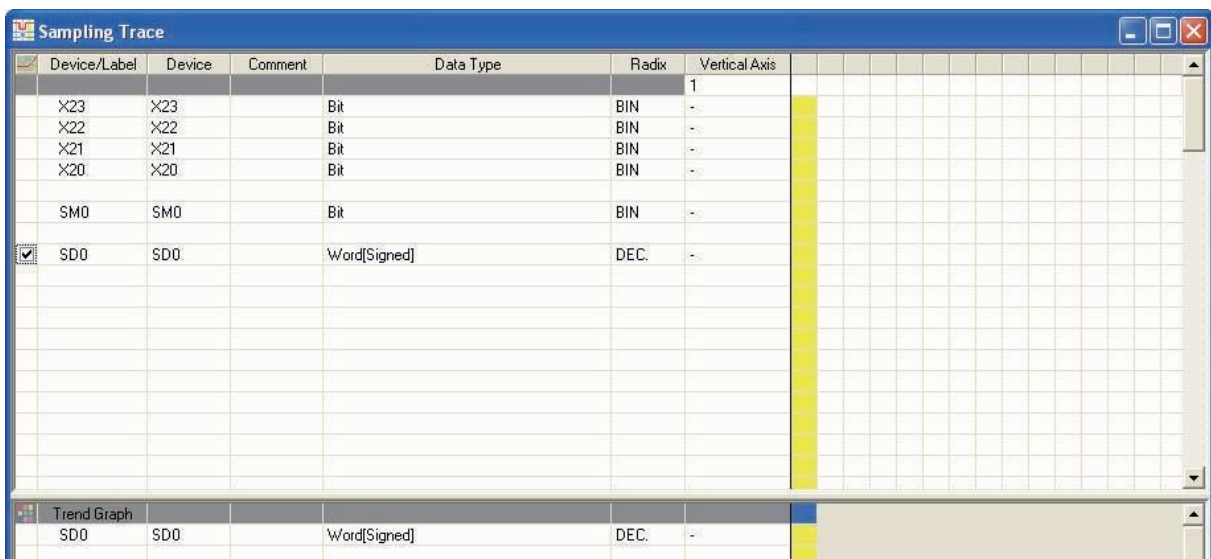
9) Hộp thoại Detail Setting - Trigger Condition biến mất. Click vào nút **End Setting** để đóng hộp thoại Trace Setting.



10) Đặt biến để truy dấu trên màn hình Sampling Trace như bên trái.

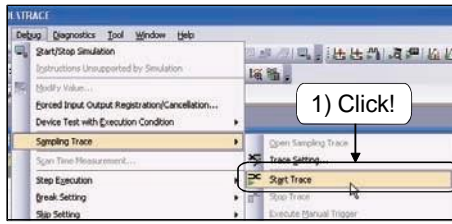
11) Kiểm tra hộp thoại hiện ra đồ thị xu hướng của SD0.

12) "SD0" hiện ra trên vùng đồ thị xu hướng trên màn hình Sampling Trace.

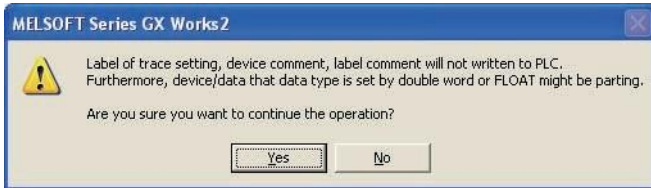




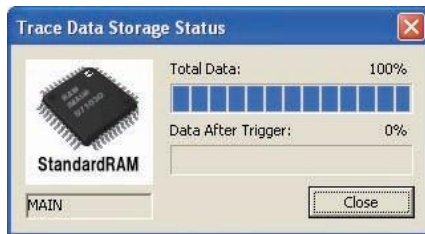
(2) Bắt đầu lấy mẫu truy dấu



1) Click [Debug] → [Sampling Trace] → [Start Trace].

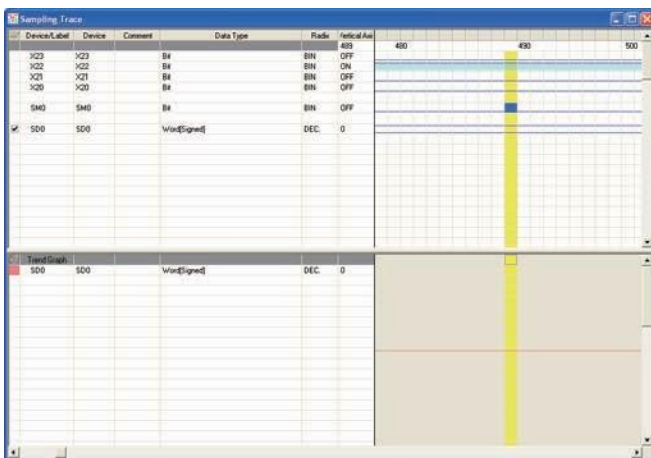


2) Tin nhắn xuất hiện ở bên trái hiện ra . Click nút **Yes** .



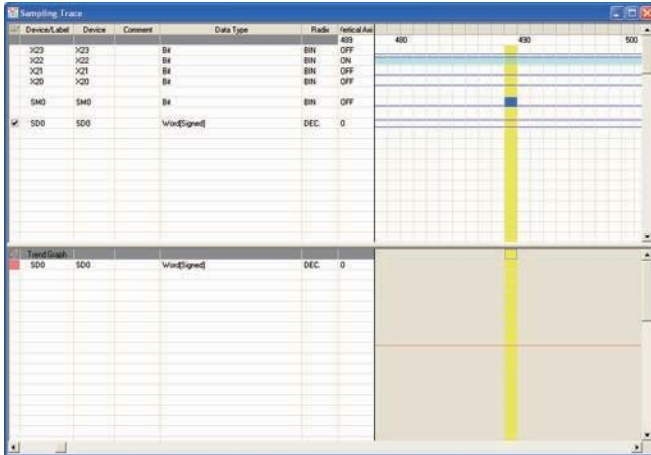
3) Màn hình Trace Data Storage Status hiện ra khi lấy mẫu truy dấu bắt đầu.

Sau khi xác nhận tổng dữ liệu lên đến 100%, thao tác công tắc số để tạo ra lỗi.



4) Kết quả truy dấu được hiện lên trên màn hình Sampling Trace.

### (3) Kiểm tra kết quả truy dấu

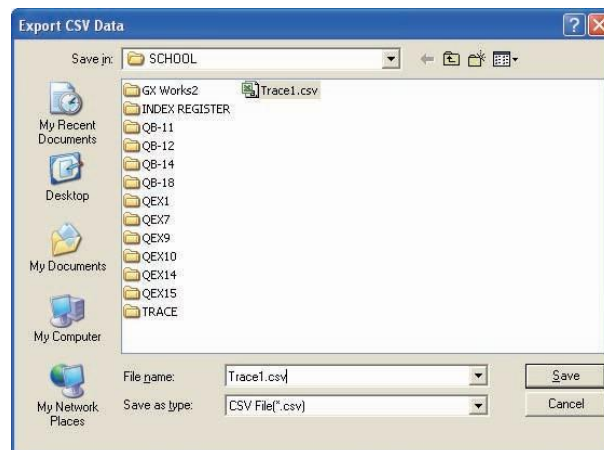


- 1) Kéo xuống màn hình đồ thị xu hướng đến điểm bắt đầu để kiểm tra giá trị biến tại điểm xảy ra lỗi.

#### Gợi ý

Lưu dữ liệu truy dấu đến một máy tính cá nhân

Click [Debug] → [Sampling Trace] → [Export CSV Data]. Hộp thoại dưới đây hiện ra.



Sau khi nhập tên tệp, click vào nút  .

## CHƯƠNG 7 CHƯƠNG TRÌNH MODULE CHỨC NĂNG THÔNG MINH

### 7.1 Module chức năng thông minh

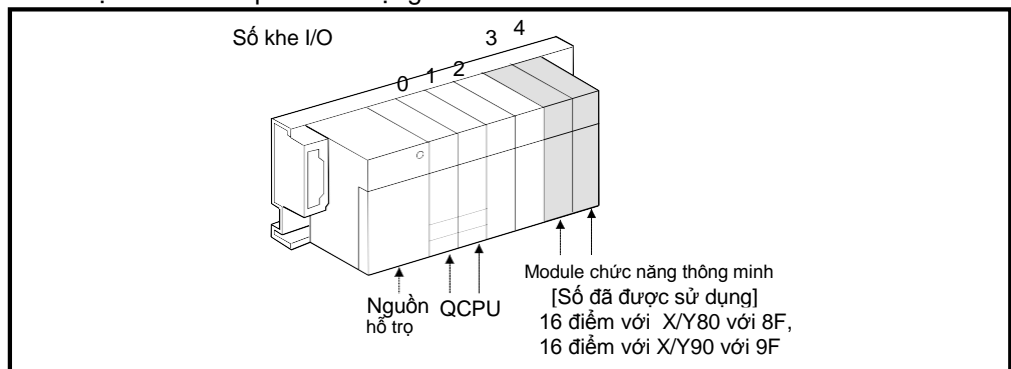
- (1) Module chức năng thông minh có dạng:  
 Trên bộ điều khiển lập trình CPU (sau này được gọi tắt là QCPU), một vài chức năng không được hỗ trợ hoặc bị giới hạn trong quá trình sử dụng. Module chức năng thông minh hỗ trợ những chức năng này thay cho QCPU.  
 Do đó người sử dụng cần phải chọn một module chức năng thông minh để đáp ứng được các mục đích sử dụng.  
 QCPU phù hợp với module QCPU có chức năng thông minh. Bảng dưới đây sẽ đưa ra những ví dụ những module chức năng thông minh.

Bảng 7.1 Ví dụ về module chức năng thông minh

Tên module	Số lượng điểm I/O có thể quản lý	Chức năng	Điện áp và dòng điện tiêu thụ
Module chuyển đổi tương tự sang số (Q64AD)	16 điểm	Module chuyển đổi ở đầu vào; 0 đến 20mA → 0 đến 4000 (Ở chế độ phân giải tiêu chuẩn) 0 đến ±10V → 0 đến ±4000 (Ở chế độ phân giải tiêu chuẩn)	5VDC  0.63A
Module chuyển đổi số sang tương tự (Q62DAN)	16 điểm	Module chuyển đổi ở đầu ra; 0 đến 4000 → 0 đến 20mA (Ở chế độ phân giải tiêu chuẩn) 0 đến ±4000 → 0 đến ±10V (Ở chế độ phân giải tiêu chuẩn)	5VDC  0.33A  24VDC  0.12A

- (2) Cách sử dụng module chức năng thông minh với các CPU

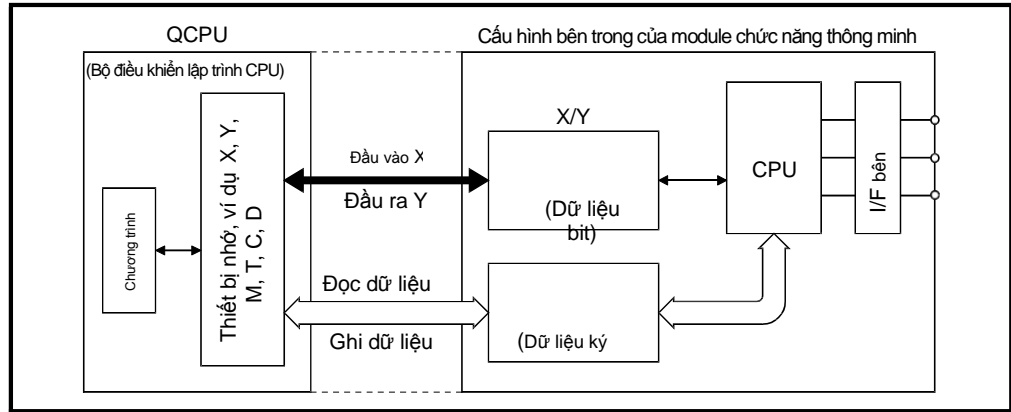
Với một module chức năng thông minh có thể được cài đặt trên bất kỳ khe I/O nào trên một đế cắm và phần mở rộng của nó.



Hình 7.1. Lắp đặt module chức năng thông minh

## 7.2 Giao tiếp dữ liệu giữa module chức năng thông minh và CPU

Một module chức năng thông minh và CPU sẽ trao đổi hai dạng dữ liệu chính  
Dữ liệu bit: Tín hiệu được sử dụng có đầu vào là X và đầu ra là Y  
Dữ liệu ký tự: 16 bit hoặc 32 bit dữ liệu

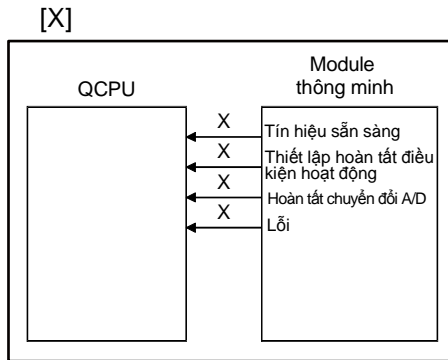


Hình 7.2 Cấu hình bên trong của module chức năng thông minh

### 7.2.1 Các tín hiệu I/O đến CPU

Với 1 bit tín hiệu được chuyển đổi giữa QCPU và module chức năng thông minh sử dụng đầu vào là X và đầu ra là Y.

X và Y không mang ý nghĩa giống như bên trong các I/O nhưng các ký hiệu ở đây có thể sử dụng chương trình tuần tự để đại diện một cách độc quyền cho các tín hiệu I/O của module chức năng thông minh. Lưu ý rằng, số I/O được ấn định theo vị trí của khe mà module chức năng thông minh được lắp đặt.



Hình 7.3

Tín hiệu X của module chức năng thông minh

X trong chương trình tuần tự đại diện cho các tín hiệu này được chuyển đến QCPU từ module chức năng thông minh. Những tín hiệu này được tạo ra trên module chức năng thông minh. Chú ý, X được sử dụng như một cách để liên lạc với chương trình này. Một số tín hiệu được ví dụ cụ thể dưới đây:

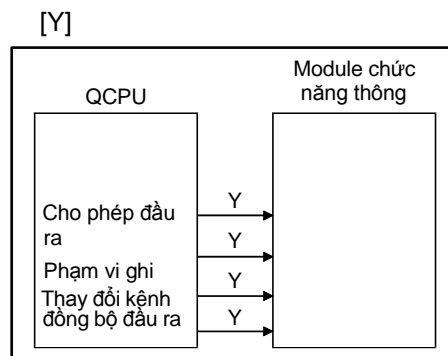
(1) Tín hiệu READY

Tín hiệu này thông báo tới QCPU rằng một module chức năng thông minh có thể khởi động bình thường bật nguồn và sẵn sàng hoạt động.

(2) Thiết lập điều kiện hoạt động đã hoàn tất

Đây là tín hiệu được sử dụng như là điều kiện khóa liên động để chuyển đổi thiết lập điều kiện hoạt động theo yêu cầu (Y9) mở/tắt khi chế độ cài đặt được thay đổi

- Thiết lập chuyển đổi A/D hoạt động/không hoạt động (địa chỉ bộ nhớ đệm 0: Un\G0)
- CH thời gian trung bình/số lần trung bình (địa chỉ bộ nhớ đệm 1 đến 8: Un\G1 đến Un\G8)
- Thiết lập quá trình trung bình (địa chỉ bộ nhớ đệm 9: Un\G0)

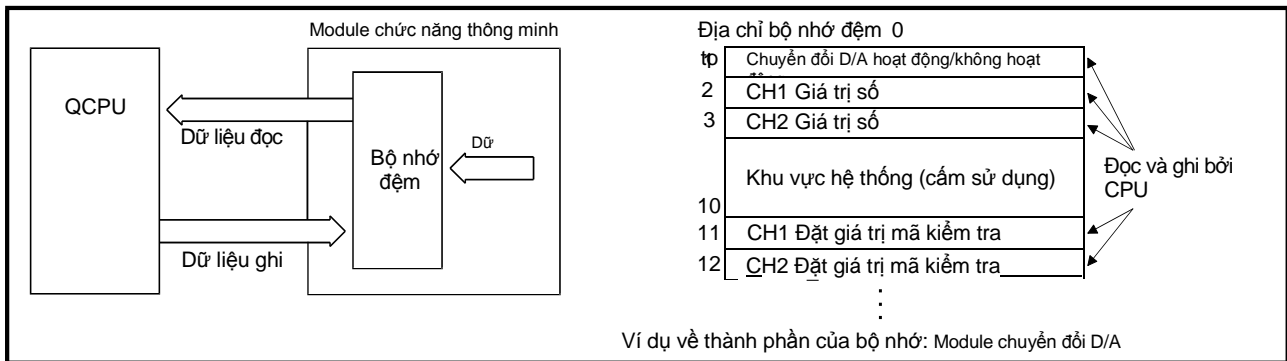


Hình 7.4 Tín hiệu Y của CPU

SET, RST, hoặc OUT-Y yêu cầu tín hiệu ra xuất phát từ một QCPU đến module chức năng thông minh. Những tín hiệu này được tạo ra trên một QCPU. Chú ý rằng, chúng được sử dụng như những sợi dây hoặc dùng để liên lạc trong các chương trình, (Ví dụ) Đầu ra của module bộ chuyển đổi D/A đưa ra một hướng dẫn cho biết giá trị tương tự đã được chuyển đổi sang giá trị số.

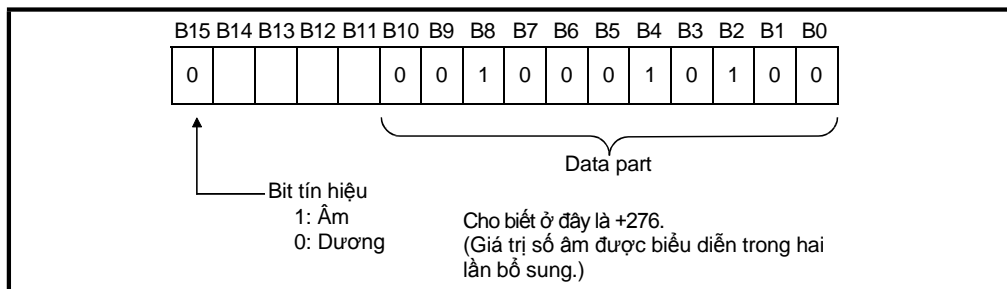
### 7.2.2 Giao tiếp dữ liệu trong module chức năng thông minh

Dữ liệu được phát hoặc thu với 16 bit hoặc 32 bit. Những module chức năng thông minh có đều có một bộ nhớ đệm để lưu trữ những dữ liệu này



Hình 7.5 Bộ nhớ đệm

- (1) QCPU có thể đọc và ghi dữ liệu từ bộ nhớ đệm. Chú ý rằng, một vài module có thể ghi dữ liệu bộ nhớ đệm từ thiết bị ngoại vi thông qua một giao diện.
- (2) Trong bộ nhớ đệm, khoảng trống của một ký tự (16bit) được dự trữ cho mỗi địa chỉ của module chức năng thông minh duy nhất. Địa chỉ nhớ nhất là 0, và những địa chỉ này được sử dụng riêng mới một module đích để đọc hoặc ghi. Đơn vị nhỏ nhất là một ký tự. Dữ liệu từ 17 bit đến 32 bit tương đương như 2 (32 bit) ký tự dữ liệu



Hình 7.6 Hình ảnh ví dụ của bộ nhớ đệm (module chuyển đổi D/A)

Hình 7.6 cho thấy 16 bit bộ nhớ đệm của một module chuyển đổi D/A, nơi mà giá trị số được ghi ra. Con số này thu được từ số lượng giá trị số của một QCPU ghi vào bộ nhớ đệm trong khoảng -4096 đến +4095 trong hệ nhị phân (dài 16 bit)

- (3) Bộ nhớ đệm là RAM.

### 7.3 Giao tiếp trong module chức năng thông minh

#### 7.3.1 Phương thức giao tiếp trong module chức năng thông minh

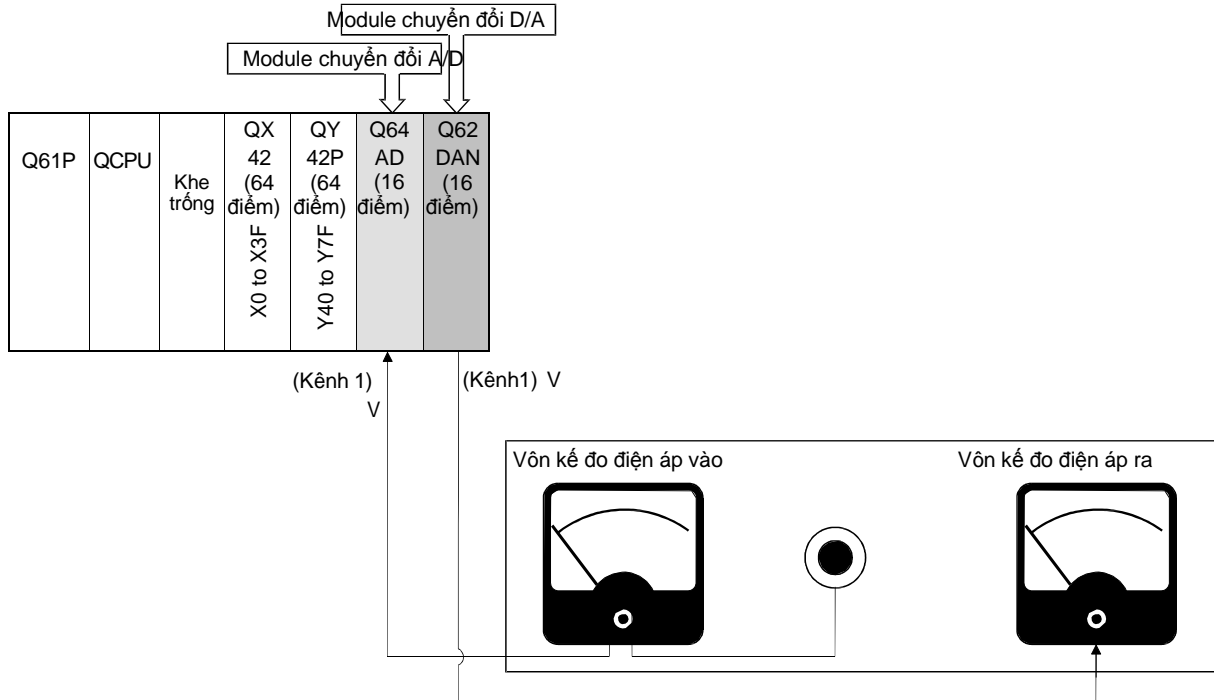
Bảng dưới đây cho biết các phương thức liên lạc giữa QCPU với một module chức năng thông minh.

Bảng 7.2 Phương thức giao tiếp dữ liệu chức năng thông minh

Phương thức liên lạc	Chức năng	Phương thức thiết lập
Thiết lập ban đầu, Tự động làm mới thiết lập	<p>Thực hiện các cài đặt ban đầu và tự động làm mới thiết lập của các module chức năng thông minh. Những cài đặt này cho phép ghi/đọc dữ liệu từ những module chức năng thông minh không phân biệt các chương trình truyền thông thông minh hoặc địa chỉ bộ nhớ đệm.</p> <p>Ví dụ: Q64AD được sử dụng trong module chuyển đổi A/D.</p> <ul style="list-style-type: none"> <li>Cài đặt ban đầu: <ul style="list-style-type: none"> <li>Thiết lập kích hoạt chuyển đổi A/D: hoạt động/không hoạt động</li> <li>Xử lý đặc điểm kỹ thuật lấy mẫu/trung bình.</li> <li>Trung bình thời gian /số lần cấu hình trung bình</li> <li>Thời gian trung bình/số lần cấu hình trung bình (Dữ liệu tự động làm mới thiết lập được lưu trữ trong các tham số của module chức năng thông minh trên một QCPU)</li> </ul> </li> <li>Làm mới thiết lập tự động: Một biến trên QCPU lưu trữ những dữ liệu sau: <ul style="list-style-type: none"> <li>Đầu ra số từ Q64AD</li> <li>Giá trị lớn nhất và nhỏ nhất của Q64AD</li> <li>Mã lỗi</li> </ul> </li> </ul> <p>(Đặt dữ liệu tự động làm mới thiết lập được lưu trữ trong</p>	Sử dụng GX Works2.
Biến địa chỉ giá trị ban đầu	<p>Việc ghi dữ liệu trong cài đặt ban đầu của biến địa chỉ trên module chức năng thông minh đến module chức năng thông minh ở những thời điểm sau:</p> <ul style="list-style-type: none"> <li>Tại thời điểm bật nguồn của một QCPU</li> <li>Tại thời điểm reset</li> <li>Tại thời điểm chuyển từ chế độ STOP sang RUN</li> </ul>	Sử dụng GX Works2 để xác định phạm vi của các thiết bị module chức năng thông minh (U□\G□).
Chỉ dẫn FROM/TO	Đọc hoặc ghi dữ liệu đi hoặc đến bộ nhớ đệm trên module chức năng thông minh	Sử dụng chỉ dẫn này trong một chương trình tuần tự.
Biến địa chỉ module chức năng thông minh (U□\G□)	<p>Trực tiếp xử lý bộ nhớ đệm trên một module chức năng thông minh như một Biến địa chỉ của QCPU</p> <p>Không giống như chỉ dẫn FROM/TO, yêu cầu này chỉ là một chỉ dẫn của quá trình xử lý dữ liệu được đọc từ module chức năng thông minh</p>	Chỉ định biến địa chỉ này như một biến trong chương trình tuần tự
Lệnh chuyên dụng của module chức năng thông minh	Được sử dụng để đơn giản hóa các chương trình của module chức năng thông minh để dễ tiếp cận với người sử dụng.	Sử dụng chỉ dẫn này trong một chương trình tuần tự

#### 7.4 Hệ thống module chức năng thông minh trong thiết bị DEMO

Sử dụng module chuyển đổi A/D hoặc D/A để chuyển đổi từ tín hiệu tương tự sang tín hiệu số, được nhập với âm lượng hoặc chuyển đổi số trên bộ DEMO.

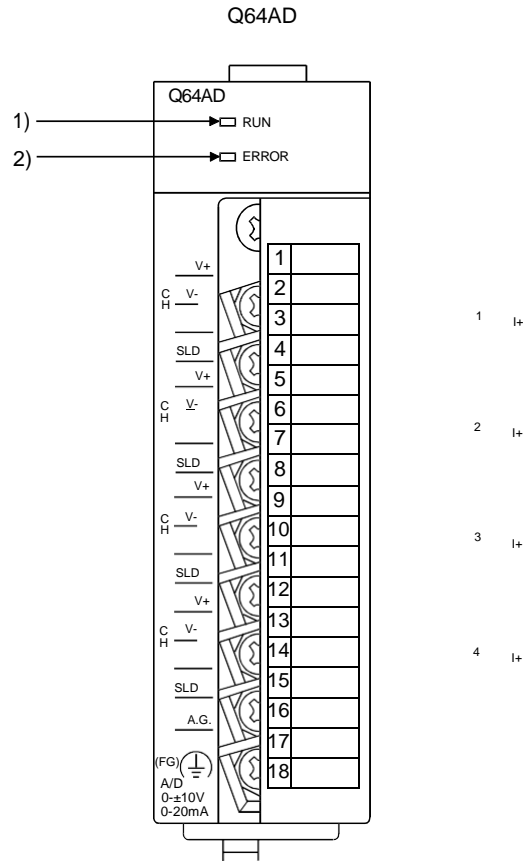




## 7.5 Module chuyển đổi Q64AD Tương tự/Số

### 7.5.1 Tên các bộ phận

Dưới đây sẽ giải thích về các bộ phận của Q64AD. Chi tiết về Q64AD, hãy xem qua hướng dẫn sử dụng.

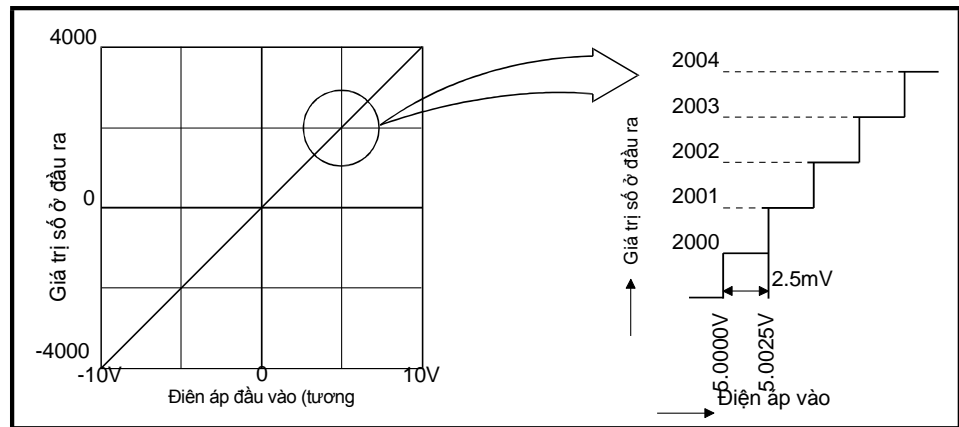


TT	Tên và hình dạng	Mô tả
1)	RUN LED	Cho biết tình trạng hoạt động của các module chuyển đổi A/D Bật : Hoạt động bình thường Nhấp nháy : Chế độ cài đặt Ngưỡng/Độ lợi Tắt : Mất nguồn điện hoặc thiết bị giám sát lỗi hẹn giờ
2)	ERROR LED	Cho biết lỗi và tình trạng hiện tại của module chuyển đổi A/D Bật : Đã xuất hiện lỗi Tắt : Hoạt động bình thường Nhấp nháy : Hệ thống chuyển đổi có lỗi Các giá trị khác 0 được thiết lập để chuyển đổi thành 5 trên module chức năng thông minh.

## 7.5.2 Đặc điểm của chuyển đổi A/D

### (1) Đặc điểm của điện áp đầu vào

(Trong chế độ phân giải tiêu chuẩn, tín hiệu tương tự có điện áp vào khoảng -10 đến 10V)

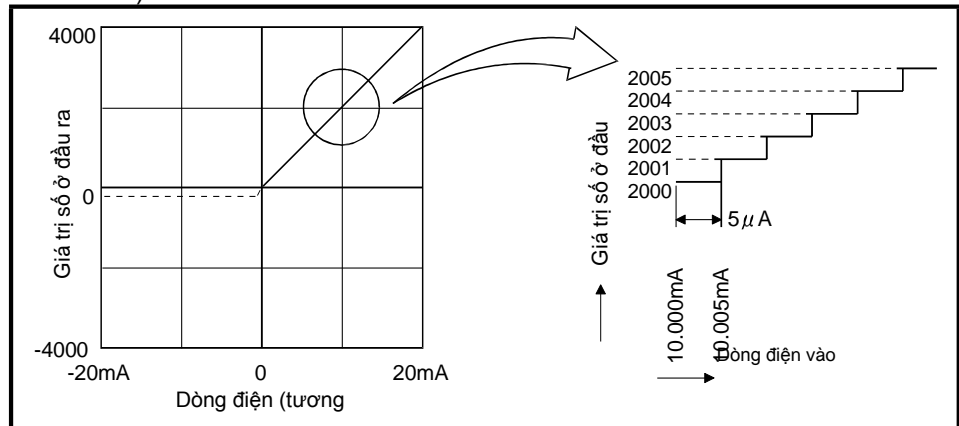


Hình 7.12 Đặc điểm của chuyển đổi A/D (Điện áp vào)

Module chuyển đổi A/D chuyển đổi các giá trị tương tự được nhập từ những thiết bị khác sang các giá trị số, vì vậy các CPU có thể xử lý những giá trị này. Trên điện áp đầu vào, ví dụ, module chuyển đổi A/D chuyển đổi từ -10V sang -4000 và từ 10V sang 4000. Điều này có nghĩa các module chuyển đổi từ điện áp vào là 2.5mV sang giá trị số là 1, và loại bỏ các giá trị nhỏ hơn 2.5mV.

### (2) Đặc điểm dòng điện đầu vào của bộ chuyển đổi A/D

(Ở độ phân giải tiêu chuẩn, tín hiệu analog có dòng điện vào khoảng 0 đến 20mA)



Hình 7.13 Đặc điểm của chuyển đổi A/D (Dòng điện vào)

Các module chuyển đổi dòng điện 0mA ở đầu vào chuyển thành 0 ở đầu ra và 20mA đến 4000. Điều này có nghĩa là các module này chuyển đổi một cường độ 5  $\mu$ A sang giá trị số là 1, và loại bỏ các giá trị nhỏ hơn 5 $\mu$ A.

#### LƯU Ý

Giá trị điện áp và dòng điện tương đương với giá trị số là 1 do chuyển đổi A/D (ở độ phân giải tối đa) khác nhau tùy thuộc vào các thiết lập của chế độ phân giải (1/4000, 1/12000, 1/16000) hoặc sự quy định khoảng đầu ra.

### 7.5.3 Danh sách các tín hiệu I/O và phân vùng bộ nhớ đệm

#### (1) Danh sách các tín hiệu I/O

Bảng dưới đây cho thấy một danh sách các tín hiệu I/O cho các module chuyển đổi A/D. Chú ý rằng số lượng I/O (X/Y) xuất hiện trong phần này và về sau là những giá trị khởi đầu số I/O cho module chuyển đổi A/D được đặt là 0.

Hướng tín hiệu: CPU ← module chuyển đổi A/D		Hướng tín hiệu: CPU → module chuyển đổi A/D	
Thứ tự biến (vào)	Tên tín hiệu	Thứ tự biến (ra)	Tên tín hiệu
X0	Module đã sẵn sàng	Y0	Cấm sử dụng *1
X1	Cờ bù nhiệt độ bị hao hụt	Y1	
X2	Cấm sử dụng *1	Y2	
X3		Y3	
X4		Y4	
X5		Y5	
X6		Y6	
X7		Y7	
X8	Cờ đánh dấu ở chế độ phân giải cao	Y8	
X9	Cờ đánh dấu thiết lập hoàn tất điều kiện hoạt động	Y9	Yêu cầu điều kiện hoạt động
XA	Cờ đánh dấu chế độ thiết lập ngưỡng/độ lợi	YA	Yêu cầu phạm vi sử dụng
XB	Cờ đánh dấu thay đổi kênh hoàn tất	YB	Yêu cầu thay đổi kênh
XC	Cấm sử dụng *1	YC	Cấm sử dụng *1
XD	Cờ hoàn tất làm mới giá trị lớn nhất/nhỏ nhất	YD	Yêu cầu làm mới giá trị lớn nhất/nhỏ nhất
XE	Cờ hoàn tất chuyển đổi A/D	YE	Cấm sử dụng *1
XF	Cờ báo lỗi	YF	Yêu cầu xóa sạch lỗi

#### Chú thích

\*1: Những tín hiệu không cho phép hoạt động bởi chúng chỉ được hoạt động dưới sự cho phép của hệ thống. Nếu chúng được bật/tắt bởi chương trình tuần tự sẽ dẫn đến hoạt động của module chuyển đổi A/D không được đảm bảo.

(2) Phân vùng bộ nhớ đệm (Q64AD)

Trong mục này sẽ giải thích sự phân vùng bộ nhớ đệm của Q64AD

Chú ý

Không ghi dữ liệu vào vùng hệ thống hoặc vô hiệu hóa vùng ghi dữ liệu từ các chương trình tuần tự. Làm như vậy có thể gây ra sự cố.

Phân vùng bộ nhớ đệm (Q64AD) (1/2)

Địa chỉ		Mô tả	Giá trị mặc định	Read/write <sup>*1</sup>
Hệ 16	Hệ 10			
0H	0	Thiết lập chuyển đổi A/D hoạt động/không hoạt động	0	R/W
1H	1	CH1 Thời gian trung bình/số lần trung bình	0	R/W
2H	2	CH2 Thời gian trung bình/số lần trung bình	0	R/W
3H	3	CH3 Thời gian trung bình/số lần trung bình	0	R/W
4H	4	CH4 Thời gian trung bình/số lần trung bình	0	R/W
5H	5	Khu vực hệ thống	-	-
:	:			
8H	8			
9H	9	Quá trình thiết lập trung bình	0	R/W
A	10	Cờ chuyển đổi A/D hoàn tất	0	R
B	11	CH1 Xuất ra giá trị số	0	R
CH	12	CH2 Xuất ra giá trị số	0	R
DH	13	CH3 Xuất ra giá trị số	0	R
E	14	CH4 Xuất ra giá trị số	0	R
FH	15	Khu vực hệ thống	-	-
:	:			
12H	18			
13H	19	Mã lỗi	0	R
14H	20	Phạm vi cài đặt (CH1 to CH4)	0	R
15H	21	Khu vực hệ thống	-	-
16H	22	Chế độ thiết lập Ngưỡng/Độ lợi: Đặc tính Ngưỡng	0	R/W
17H	23	Chế độ thiết lập Ngưỡng/Độ lợi: Đặc tính Độ lợi	0	R/W

\*1: Cho biết đọc và ghi các chương trình tuần tự được kích hoạt.

R: Chế độ Đọc

W: Chế độ Ghi

Phân vùng bộ nhớ đệm (Q64AD) (2/2)

Địa chỉ		Mô tả	Giá trị mặc định	Read/writ e <sup>*1</sup>
Hệ 16	Hệ 10			
18H	24	Khu vực hệ thống	-	-
:	:			
1DH	29			
1EH	30	CH1 Giá trị lớn nhất	0	R/W
1FH	31	CH1 Giá trị lớn nhất	0	R/W
20H	32	CH2 Giá trị lớn nhất	0	R/W
21H	33	CH2 Giá trị lớn nhất	0	R/W
22H	34	CH3 Giá trị lớn nhất	0	R/W
23H	35	CH3 Giá trị lớn nhất	0	R/W
24H	36	CH4 Giá trị lớn nhất	0	R/W
25H	37	CH4 Giá trị lớn nhất	0	R/W
26H	38	Khu vực hệ thống	-	-
:	:			
9DH	157			
9EH	158	Cài đặt chế độ chuyển đổi	0	R/W
9FH	159			
A0H	160	Khu vực hệ thống	-	-
:	:			
C7H	199			
C8H	200	Vượt qua thiết lập phân loại dữ liệu <sup>*2</sup>	0	R/W
C9H	201	Khu vực hệ thống	-	-
CAH	202	CH1 Thiết lập giá trị ngưỡng vận chuyển công nghiệp <sup>*2</sup>	0	R/W
CBH	203	CH1 Thiết lập giá trị độ lợi vận chuyển công nghiệp <sup>*2</sup>	0	R/W
CCH	204	CH2 Thiết lập giá trị ngưỡng vận chuyển công nghiệp <sup>*2</sup>	0	R/W
CDH	205	CH2 Thiết lập giá trị độ lợi vận chuyển công nghiệp <sup>*2</sup>	0	R/W
CEH	206	CH3 Thiết lập giá trị ngưỡng vận chuyển công nghiệp <sup>*2</sup>	0	R/W
CFH	207	CH3 Thiết lập giá trị độ lợi vận chuyển công nghiệp <sup>*2</sup>	0	R/W
D0H	208	CH4 Thiết lập giá trị ngưỡng vận chuyển công nghiệp <sup>*2</sup>	0	R/W
D1H	209	CH4 Thiết lập giá trị độ lợi vận chuyển công nghiệp <sup>*2</sup>	0	R/W
D2H	210	CH1 Thiết lập phạm vi người sử dụng giá trị ngưỡng <sup>*2</sup>	0	R/W
D3H	211	CH1 Thiết lập phạm vi người sử dụng giá trị độ lợi <sup>*2</sup>	0	R/W
D4H	212	CH2 Thiết lập phạm vi người sử dụng giá trị ngưỡng <sup>*2</sup>	0	R/W
D5H	213	CH2 Thiết lập phạm vi người sử dụng giá trị độ lợi <sup>*2</sup>	0	R/W
D6H	214	CH3 Thiết lập phạm vi người sử dụng giá trị ngưỡng <sup>*2</sup>	0	R/W
D7H	215	CH3 Thiết lập phạm vi người sử dụng giá trị độ lợi <sup>*2</sup>	0	R/W
D8H	216	CH4 Thiết lập phạm vi người sử dụng giá trị ngưỡng <sup>*2</sup>	0	R/W
D9H	217	CH4 Thiết lập phạm vi người sử dụng giá trị độ lợi <sup>*2</sup>	0	R/W

\*1: Cho biết đọc và ghi các chương trình tuần tự được kích hoạt.

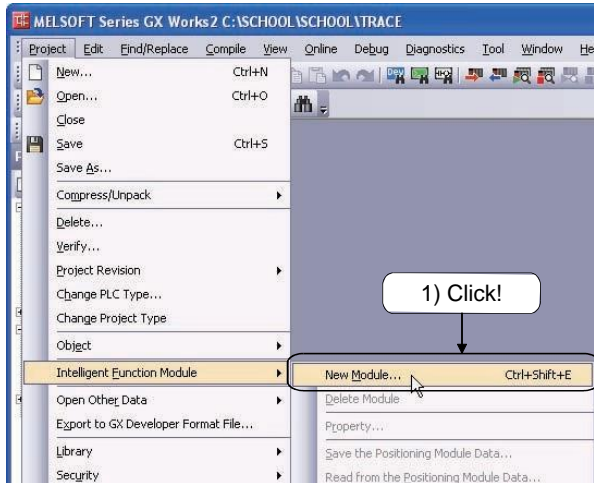
R: Chế độ đọc

W: Chế độ ghi

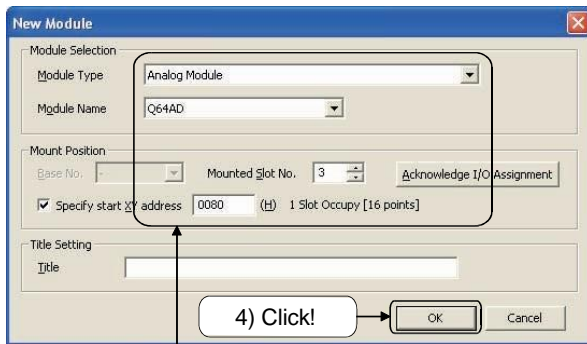
\*2: Khu vực sử dụng để khôi phục các cài đặt phạm vi sử dụng ngưỡng/độ lợi khi các thay đổi module trực tuyến được thi hành.

## 7.5.4 Bổ sung hoặc cài đặt dữ liệu cho module chức năng thông minh

Phần này sẽ giải thích làm thế nào để thiết lập dữ liệu cho module chức năng thông minh. Sau khi các module được thêm vào, thiết lập các cài đặt dữ liệu (tham số và cài đặt chuyển đổi) của module chức năng thông minh.



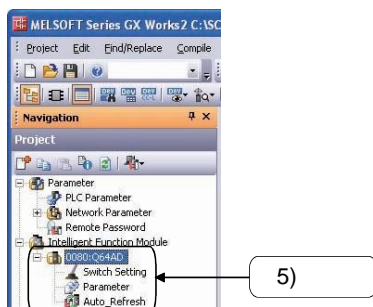
- 1) Click [Project] → [Intelligent Function Module] → [New Module].



- 2) Hộp thoại New Module được hiển thị.
- 3) Thiết lập tham số module chuyển đổi A/D  
Modulee Type : Analog Modulee  
Modulee Name : Q64AD  
Mounted Slot No. : 3  
(Specify start XY address: 0080)

- 4) Click nút **OK**

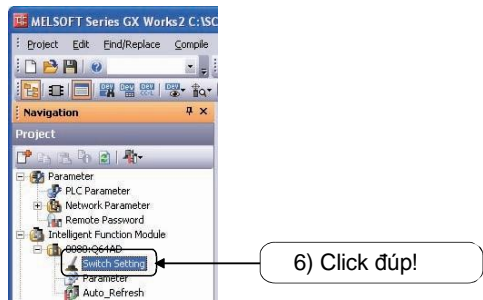
3) Đặt



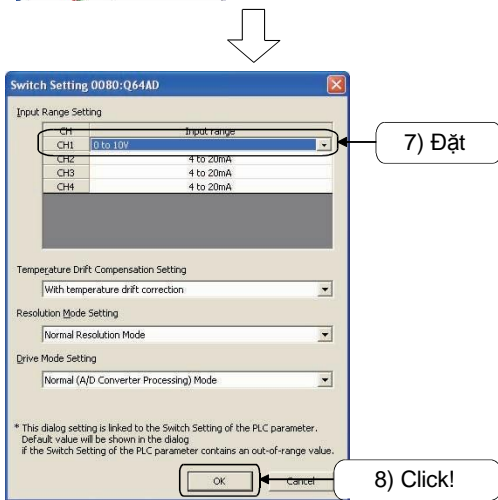
- 5) Các dữ liệu module chức năng thông minh định được thêm vào cửa sổ Project.

(Đến trang sau)

(Từ trang trước)

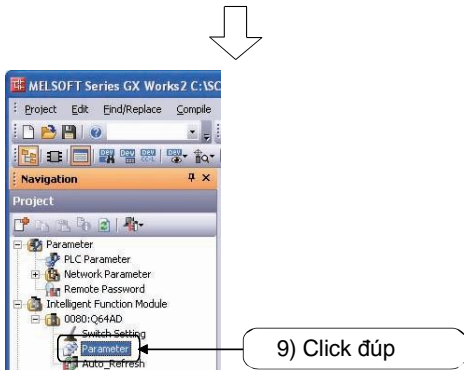


6) Click đúp vào Switch Setting.

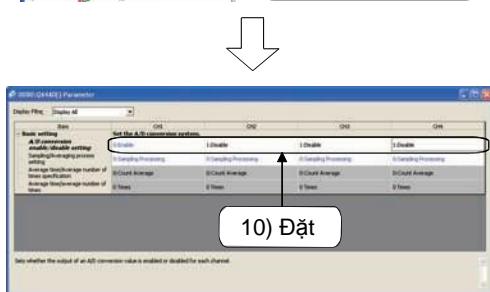


7) Hộp thoại Switch Setting được hiển thị. Thiết lập phạm vi đầu vào cho CH1 là "0 đến 10V".

8) Click nút



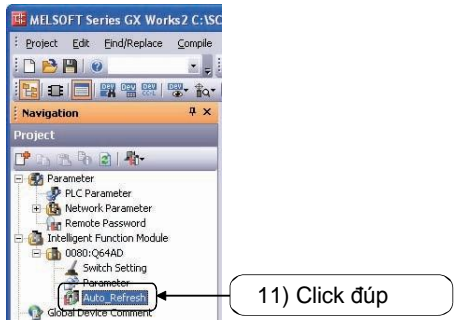
9) Click đúp Parameter.



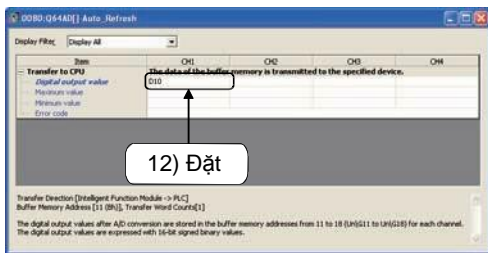
10) Hộp thoại Parameter được hiển thị. Chọn "A/D conversion enable/disable setting" cho CH2 đến CH4 chọn "1:Disable". (Chỉ CH1 được sử dụng.)

(Đến trang sau)

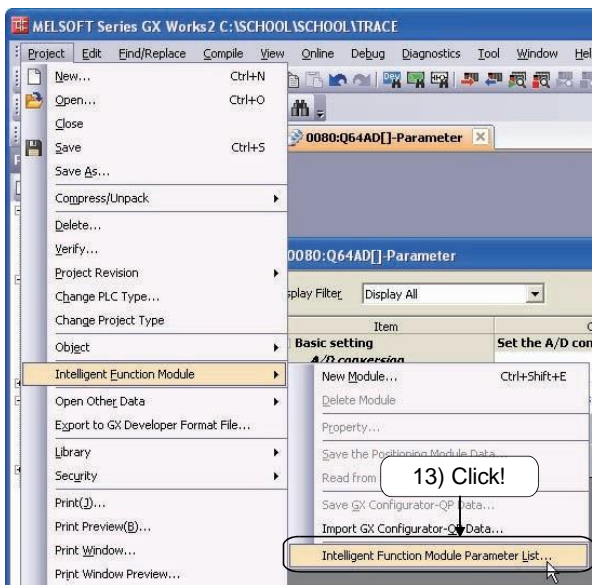
(Từ trang trước)



11) Click đúp Auto\_Refresh.



12) Màn hình Auto\_Refresh được hiển thị.  
Giá trị đầu ra Set Digital của CH1 là "D10".

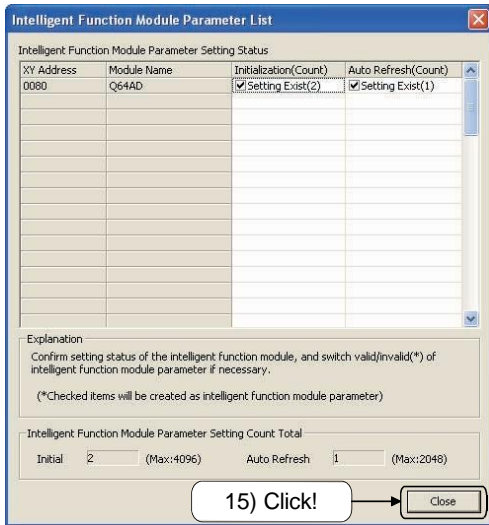


13) Click [Project] → [Intelligent Function Module] → [Intelligent Function Module Parameter List].

(Đến trang sau)



(Từ trang trước)



14) Kiểm tra xem “Setting Exits” được chọn trong Initialization (Count) và Auto Refresh (Count) cho Q64AD trong hộp thoại Intelligent Function Module Parameter List

15) Click nút

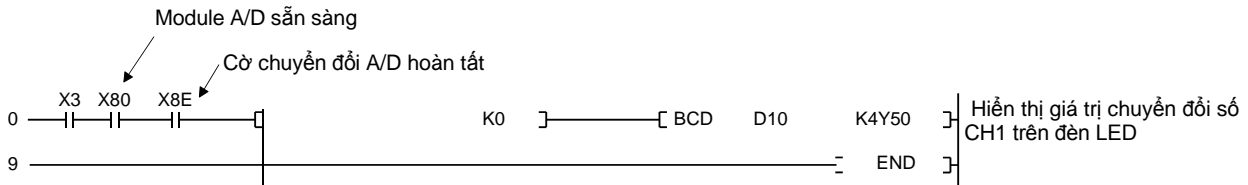
### 7.5.5 Bài tập với thiết bị Demo

#### (1) Chương trình tuần tự

Một chương trình tuần tự thực việc xử lý lấy mẫu điện áp đầu vào là tín hiệu tương tự thông qua CH1 của Q64AD, sau đó bộ chuyển đổi các giá trị tương tự sang giá trị số.

Đặt XY ban đầu của Q64AD là 80 như giải thích trước.

Tên dự án	Q64AD
Tên chương trình	MAIN



X80: Module tín hiệu sẵn sàng

X8E: Hoàn tất cờ chuyển đổi A/D

Tại thời điểm bật nguồn hoặc reset của bộ điều khiển lập trình CPU, cờ này được đánh dấu nếu bộ chuyển đổi A/D sẵn sàng hoạt động. Chuyển đổi A/D được thực hiện một lần khi cờ này được đánh dấu.

#### (2) Hoạt động của thiết bị Demo Dừng

CPU và click  trên thanh công cụ

Hộp thoại Online Data Operation được hiển thị. Click nút

, sau đó click nút  để ghi dữ liệu đến CPU

Sau đó, kích hoạt các CPU và kiểm tra các mục dưới đây

- (a) Bật X3 và thay đổi điện áp đầu vào cho một module chuyển đổi A/D cùng với âm lượng trên thiết bị Demo

Các giá trị tương tự được nhập vào trên kênh 1 (CH1) của Q64AD được lưu trữ trong bộ nhớ đệm (với giá trị số). Với các thiết lập tự động làm mới, QCPU sẽ đọc các giá trị số đã lưu trữ và lưu trữ chúng trong dữ liệu đăng ký D10.

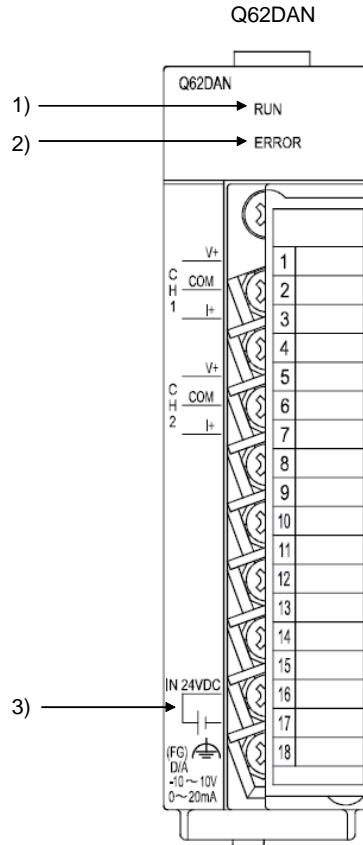
- (b) Bất kể khi nào giá trị tương tự là "-1" hoặc nhỏ hơn, 0 sẽ được đặt

- (c) Các giá trị số được hiển thị trên màn hình hiển thị (Y50 đến Y5F)

## 7.6 Module chuyển đổi Số/Tương tự Q62DAN

### 7.6.1 Tên của các bộ phận

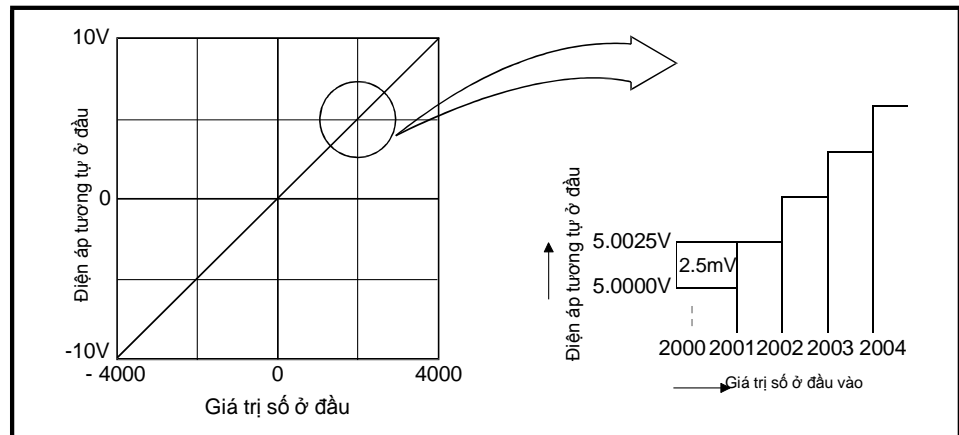
Dưới đây sẽ giải thích về các bộ phận của Q62DAN. Chi tiết về Q64AD, hãy xem qua hướng dẫn sử dụng.



TT	Tên và hình dạng	Mô tả
1)	RUN LED	Cho biết tình trạng hoạt động của các module chuyển đổi D/A Bật : Hoạt động bình thường Nhấp nháy : Chế độ cài đặt Ngưỡng/Độ lợi Tắt : Mất nguồn điện hoặc thiết bị giám sát lỗi hẹn giờ
2)	ERROR LED	Cho biết lỗi và tình trạng hiện tại của module chuyển đổi D/A Bật : Đã xuất hiện lỗi Tắt : Hoạt động bình thường Nhấp nháy : Hệ thống chuyển đổi có lỗi Các giá trị khác 0 được thiết lập để chuyển đổi thành 5 trên module chức năng thông minh
3)	Nguồn cung cấp hỗ trợ thiết bị đầu cuối	Thiết bị đầu cuối kết nối với một nguồn hỗ trợ bên ngoài 24VDC

## 7.6.2 Đặc điểm của chuyển đổi D/A

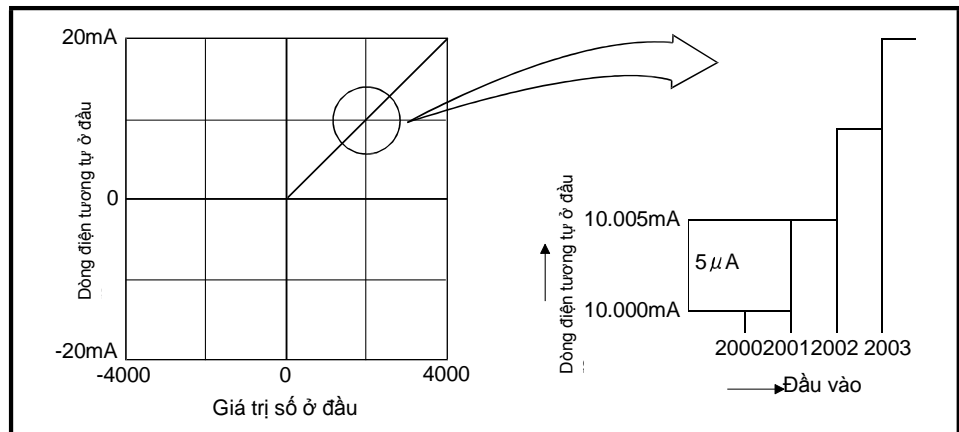
- (1) Đặc điểm của điện áp đầu vào  
(Trong chế độ phân giải tiêu chuẩn, tín hiệu tương tự có điện áp vào khoảng -10 đến 10V)



Hình 7.14 Đặc điểm của chuyển đổi D/A (Điện áp ra)

Module chuyển đổi D/A sẽ chuyển đổi các giá trị số được nhập từ một QCPU thành những giá trị tương tự và sau đó xuất chúng. Ví dụ, những module chuyển đổi giá trị số -4000 sang giá trị tương tự là -10V và 4000 sang 10V sau đó xuất ra. Điều này có nghĩa rằng các module chuyển đổi các giá trị đầu vào số là 1 sang giá trị tương tự là 2.5mV, và loại bỏ các giá trị số đầu vào trong hệ thập phân.

- (2) Đặc điểm dòng điện đầu vào của bộ chuyển đổi D/A  
(Ở độ phân giải tiêu chuẩn, tín hiệu tương tự có dòng điện vào khoảng 0 đến 20mA)



Hình 7.15 Đặc điểm của chuyển đổi D/A (Dòng điện ra)

Với dòng ở đầu ra, các module chuyển đổi giá trị số từ 0 sang 0mA và từ 4000 thành 20mA. Điều này có nghĩa các module chuyển đổi giá trị số đầu vào là 1 thành tín hiệu tương tự là 5  $\mu$ A, và loại bỏ các giá trị số đầu vào thuộc hệ thập phân.

### Chú thích

Giá trị điện áp và dòng điện tương đương với giá trị số là 1 do chuyển đổi D/A (ở độ phân giải tối đa) khác nhau tùy thuộc vào các thiết lập của chế độ phân giải (1/4000, 1/12000, 1/16000) hoặc sự quy định khoảng đầu ra.

### 7.6.3 Danh sách các tín hiệu I/O và sự phân vùng bộ nhớ đệm

#### (1) Danh sách các tín hiệu I/O

Bảng dưới đây cho thấy danh sách các tín hiệu I/O cho các module chuyển đổi D/A. Những lời giải thích này được dựa trên Q68DAVN, Q68DAIN, Q68DAV và Q68DAI với 8 kênh đầu ra tương tự (CH1 đến CH8).

Chú ý rằng số lượng I/O (X/Y) xuất hiện trong phần này và về sau là những giá trị khởi đầu số I/O cho module chuyển đổi D/A được đặt là 0.

Hướng tín hiệu	Module chuyển đổi D/A → Module CPU	Hướng tín hiệu	Module CPU → Module chuyển đổi D/A
Thứ tự biến	Tên tín hiệu	Thứ tự biến	Tên tín hiệu
X0	Module đã sẵn sàng	Y0	Cấm sử dụng <sup>*1</sup>
X1	Cấm sử dụng <sup>*1</sup>	Y1	CH1 Cờ đầu ra hoạt động/không hoạt động
X2		Y2	CH2 Cờ đầu ra hoạt động/không hoạt động
X3		Y3 <sup>*2</sup>	CH3 Cờ đầu ra hoạt động/không hoạt động
X4		Y4 <sup>*2</sup>	CH4 Cờ đầu ra hoạt động/không hoạt động
X5		Y5 <sup>*2</sup>	CH5 Cờ đầu ra hoạt động/không hoạt động
X6		Y6 <sup>*2</sup>	CH6 Cờ đầu ra hoạt động/không hoạt động
X7		Y7 <sup>*2</sup>	CH7 Cờ đầu ra hoạt động/không hoạt động
X8		Cờ đánh dấu ở chế độ phân giải cao	Y8 <sup>*2</sup>
X9	Cờ đánh dấu thiết lập hoàn tất điều kiện hoạt động	Y9	Yêu cầu điều kiện hoạt động
XA	Cờ đánh dấu chế độ thiết lập ngưỡng/độ	YA	Người sử dụng yêu cầu phạm vi ghi
XB	Cờ đánh dấu thay đổi kênh hoàn tất	YB	Yêu cầu thay đổi kênh
XC	Cờ đánh dấu hoàn tất thay đổi giá trị	YC	Yêu cầu đặt giá trị thay đổi
XD	Cờ đánh dấu chế độ đồng bộ đầu ra	YD	Yêu cầu đồng bộ đầu ra
XE	Cấm sử dụng <sup>*1</sup>	YE	Cấm sử dụng <sup>*1</sup>
XF	Cờ lỗi	YF	Yêu cầu xóa sạch lỗi

#### Chú thích

\*1: Những tín hiệu không cho phép hoạt động bởi chúng chỉ được hoạt động dưới sự cho phép của hệ thống. Nếu chúng được bật/tắt bởi chương trình tuần tự sẽ dẫn đến hoạt động của module chuyển đổi D/A không được đảm bảo.

\*2: Đối với Q62DAN và Q62DA, Y3 đến Y8 bị cấm sử dụng.  
Đối với Q64DAN và Q64DA, Y5 đến Y8 bị cấm sử dụng.

## (2) Phân vùng bộ nhớ đệm (Q62DAN)

Trong mục này sẽ giải thích sự phân vùng bộ nhớ đệm của Q62DAN

Chú ý
-------

<p>Không ghi dữ liệu vào vùng hệ thống hoặc vô hiệu hóa vùng ghi dữ liệu từ các chương trình tuần tự. Làm như vậy có thể gây ra sự cố.</p>
--

Địa chỉ		Mô tả	Giá trị mặc định <sup>*1</sup>	Read/write <sup>*2</sup>
Hệ 16	Hệ 10			
0H	0	Chuyển đổi D/A hoạt động/không hoạt động	3H	R/W
1H	1	CH1 Giá trị số	0	R/W
2H	2	CH2 Giá trị số	0	R/W
3H	3	Khu vực hệ thống	-	-
:	:			
AH	10			
BH	11	CH1 Đặt giá trị mã kiểm tra	0	R
CH	12	CH2 Đặt giá trị mã kiểm tra	0	R
DH	13	Khu vực hệ thống	-	-
:	:			
12H	18			
13H	19	Mã lỗi	0	R
14H	20	Phạm vi cài đặt (CH1 to CH2)	0H	R
15H	21	Khu vực hệ thống	-	-
16H	22	Chế độ thiết lập Ngưỡng/Độ lợi: Đặc tính Ngưỡng	0	R/W
17H	23	Chế độ thiết lập Ngưỡng/Độ lợi: Đặc tính Độ lợi	0	R/W
18H	24	Điều chỉnh đặc giá giá trị Ngưỡng/Độ lợi	0	R/W
19H	25	Khu vực hệ thống	-	-
:	:			
9DH	157			
9EH	158	Chế độ cài đặt chuyển đổi	0	R/W
9FH	159		0	R/W
A0H	160	Khu vực hệ thống	-	-
:	:			
C7H	199			
C8H	200	Vượt qua thiết lập phân loại dữ liệu <sup>*3</sup>	0	R/W
C9H	201	Khu vực hệ thống	-	-
CAH	202	CH1 Thiết lập giá trị ngưỡng vận chuyển công nghiệp <sup>*3</sup>	0	R/W
CBH	203	CH1 Thiết lập giá trị độ lợi vận chuyển công nghiệp <sup>*3</sup>	0	R/W
CCH	204	CH2 Thiết lập giá trị ngưỡng vận chuyển công nghiệp <sup>*3</sup>	0	R/W
CDH	205	CH2 Thiết lập giá trị độ lợi vận chuyển công nghiệp <sup>*3</sup>	0	R/W
CEH	206	CH1 Thiết lập phạm vi người sử dụng giá trị ngưỡng <sup>*3</sup>	0	R/W
CFH	207	CH1 Thiết lập phạm vi người sử dụng giá trị độ lợi <sup>*3</sup>	0	R/W
D0H	208	CH2 Thiết lập phạm vi người sử dụng giá trị ngưỡng <sup>*3</sup>	0	R/W
D1H	209	CH2 Thiết lập phạm vi người sử dụng giá trị độ lợi <sup>*3</sup>	0	R/W

\*1: Đây là các giá trị thiết lập ban đầu sau khi nguồn được bật hoặc hoặc bộ điều khiển lập trình CPU được reset.

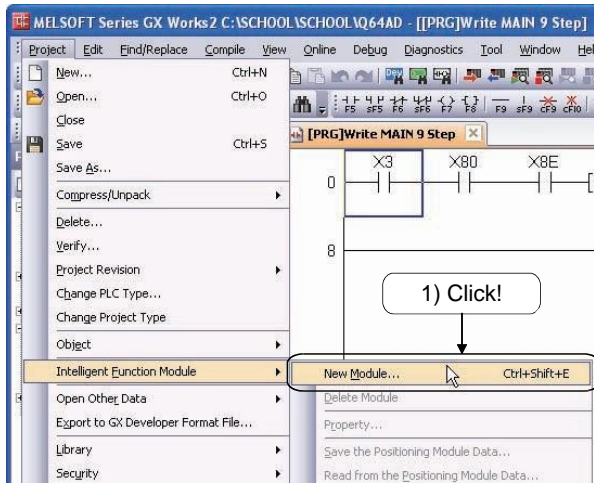
\*2: Cho biết đọc và ghi các chương trình tuần tự được kích hoạt

R: Chế độ đọc

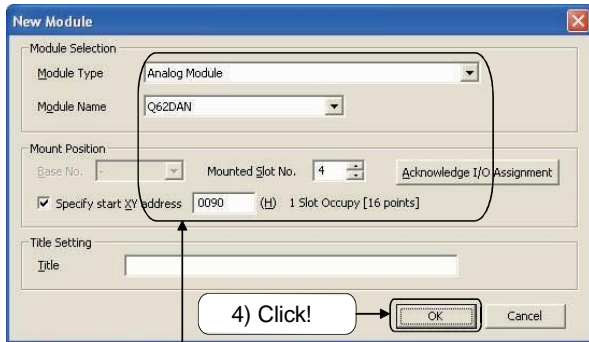
W: Chế độ ghi

\*3: Khu vực sử dụng để khôi phục các cài đặt phạm vi sử dụng ngưỡng/độ lợi khi các thay đổi module trực tuyến được thi hành.

### 7.6.4 Bổ sung hoặc cài đặt dữ liệu cho module chức năng thông minh



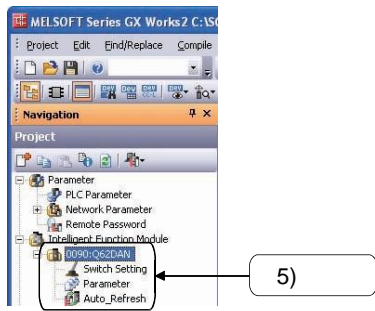
- 1) Click [Project] → [Intelligent Function Module] → [New Modulee].



- 2) Hộp thoại New Modulee được hiển thị.
- 3) Thiết lập tham số module chuyển đổi A/D.  
Modulee Type : Analog Modulee  
Modulee Name : Q62DAN  
Mounted Slot No. : 4  
(Đặt địa chỉ khởi tạo XY: 0090)

- 4) Click nút **OK**

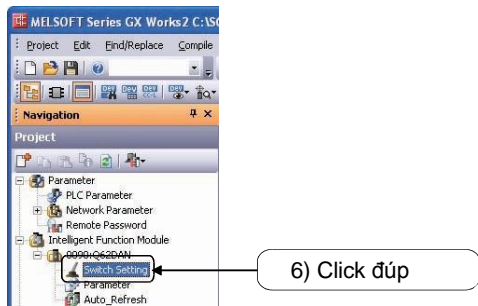
3) Đặt



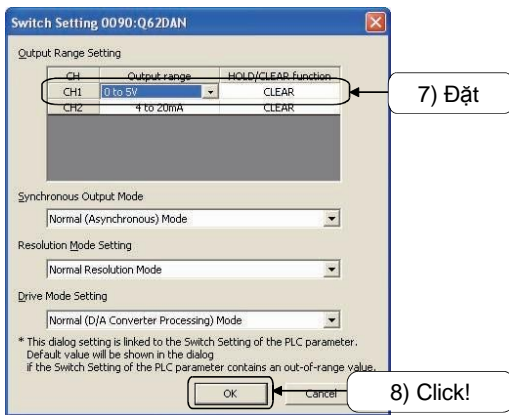
- 5) Các dữ liệu module chức năng thông minh định được thêm vào cửa sổ Project.

(Đến trang sau)

(Từ trang trước)

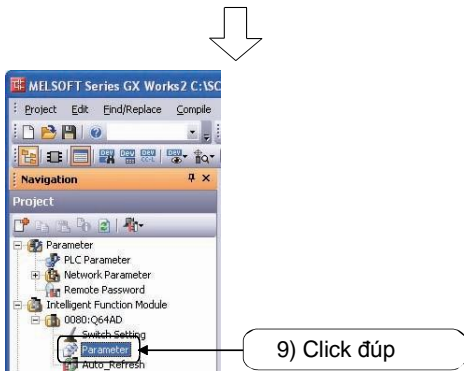


6) Click đúp Switch Setting.

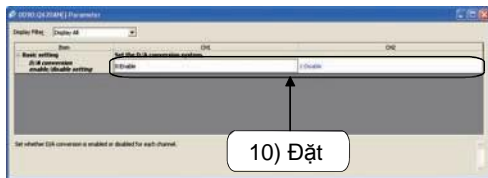


7) Hộp thoại Switch Setting được hiển thị. Thiết lập phạm vi đầu vào cho CH1 "0 đến 5V".

8) Click nút



9) Click đúp Parameter.

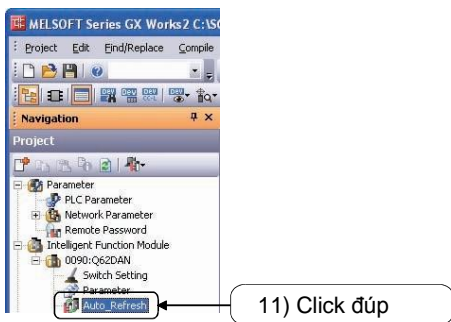


10) Hộp thoại Parameter được hiển thị. Chọn "D/A conversion enable/disable setting" cho CH1 chọn "0:Enable". (chỉ CH1 được sử dụng.)

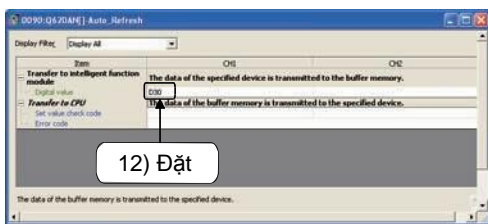
(Đến trang sau)



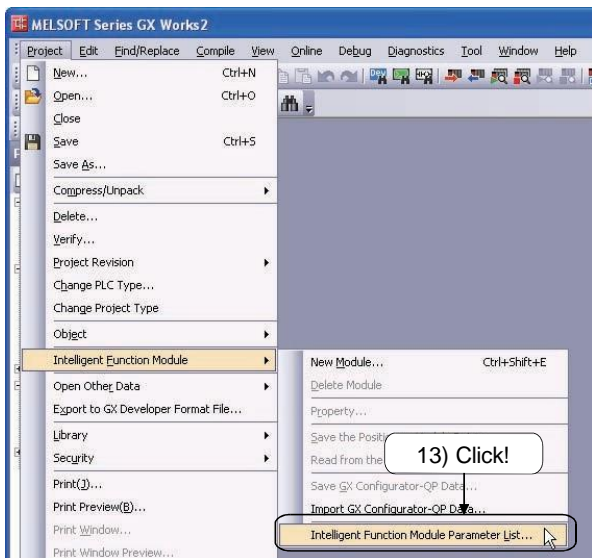
(Từ trang trước)



11) Click đúp Auto\_Refresh.



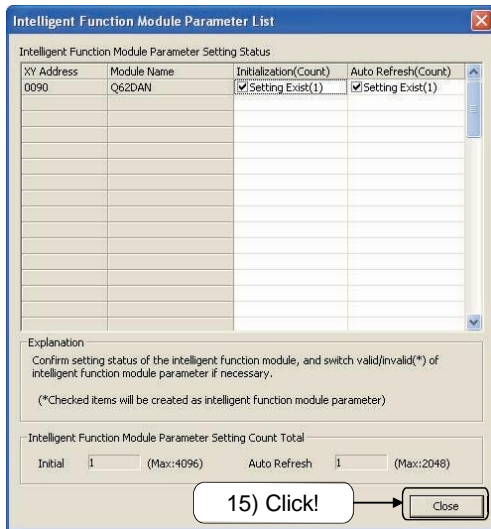
12) Màn hình Auto\_Refresh được hiển thị. Giá trị đầu ra Set Digital của CH1 là "D30".



13) Click [Project] → [Intelligent Function Module] → [Intelligent Function Module Parameter List].

(Đến trang sau)

(Từ trang trước)



14) Kiểm tra xem “Setting Exits” được chọn trong Initialization (Count) và Auto Refresh (Count) cho Q64AD trong hộp thoại Intelligent Function Modulee Parameter List.

15) Click nút [Close]

### 7.6.5 Bài tập với thiết bị Demo

(1) Chương trình tuần tự

Một chương trình tuần tự thực hiện chuyển đổi các giá trị số sang giá trị tương tự.

Đặt XY ban đầu của Q62DAN là 90 và giá trị số của CH1 là D30 như giải thích trước.

Tên dự án	Q62DAN
Tên chương	MAIN



X90: Module tín hiệu sẵn sàng

Tại thời điểm bật nguồn hoặc reset của bộ điều khiển lập trình CPU, tín hiệu này được bật nếu bộ chuyển đổi D/A sẵn sàng hoạt động. Bộ chuyển đổi D/A được thực hiện một lần khi tín hiệu được bật lên.

Y91: CH1 Cờ đánh dấu đầu ra hoạt động/không hoạt động

Lựa chọn cờ này được đánh dấu hay không trên mỗi kênh khi có giá trị đầu ra ở bộ chuyển đổi D/A hoặc có các giá trị ngưỡng.

ON: giá trị chuyển đổi D/A, OFF: giá trị ngưỡng

(2) Hoạt động của cá thiết bị Demo

Dừng CPU và click  trên thanh công cụ.

Hộp thoại Online Data Operation được hiển thị. Click nút

để ghi dữ liệu đến CPU

Sau đó, kích hoạt các CPU và kiểm tra các mục dưới đây.

(a) Bật X2 để cho phép kết quả đầu ra D/A của CH1

(b) Điện áp đầu ra theo X3 đến X5

(c) Màn hình OUTPUT của Vôn kế của D/A hiển thị các giá trị điện áp chính là kết quả đầu ra của module chuyển đổi D/A.

MEMO

## Chương 8 Chức năng mô phỏng

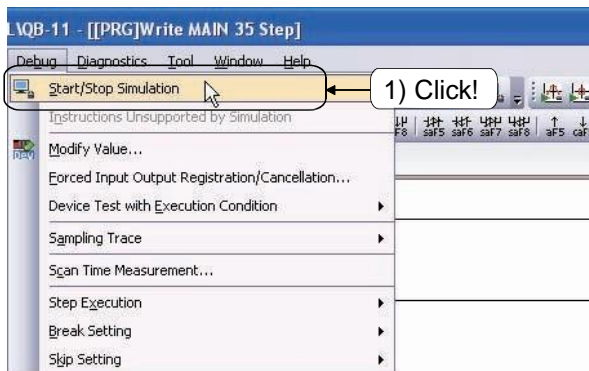
### 8.1 Chức năng mô phỏng

Chức năng mô phỏng dùng để gỡ lỗi một chương trình tuần tự sử dụng bộ điều khiển khả trình ảo trên máy tính cá nhân.

Chương trình tuần tự tạo ra có thể ngay lập tức Gỡ lỗi mà không cần kết nối đến CPU điều khiển khả trình.

Ghi nhớ
An toàn và phòng tránh lỗi của chức năng mô phỏng
1) Chức năng mô phỏng mô phỏng CPU điều khiển khả trình thật sự để Gỡ lỗi một chương trình tuần tự tạo ra. Tuy nhiên, chức năng này không đảm bảo hoạt động của chương trình tuần tự bị debug.
2) Chức năng mô phỏng sử dụng bộ nhớ để mô phỏng dữ liệu đầu vào và ra đến/từ module I/O và module chức năng thông minh. Một vài lệnh, chức năng, và bộ nhớ thiết bị không được hỗ trợ. Vì thế, kết quả hoạt động được lấy từ khối điều khiển khả trình ảo có thể khác với những cái lấy ra từ CPU điều khiển khả trình thật sự.

### 8.2 Chức năng chạy /dừng



1) Click [Debug] → [Start/Stop Simulation].

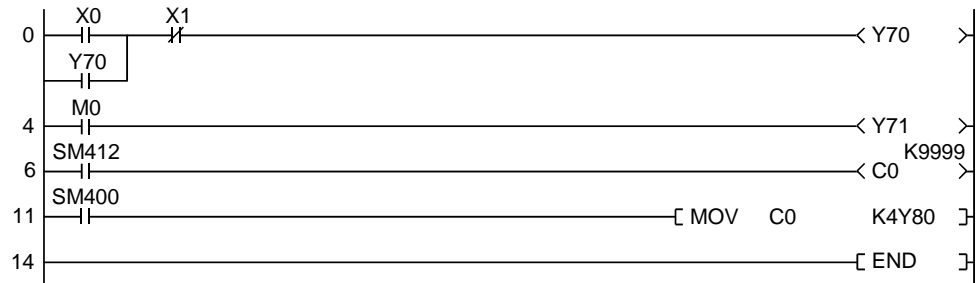
2) Màn hình GX Simulator2 hiện ra và mô phỏng bắt đầu.

3) Để dừng mô phỏng lại, click lại [Debug] → [Start/Stop Simulation] .

### 8.3 Gỡ lỗi với chương trình ví dụ

Sử dụng ví dụ dưới đây để luyện tập.

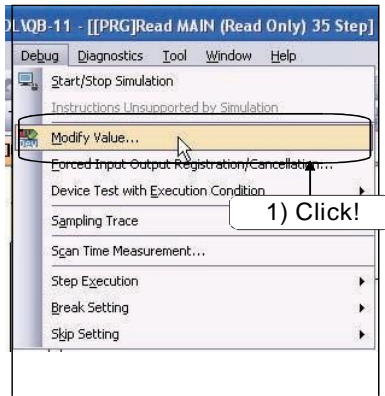
<<Chương trình ví dụ>>



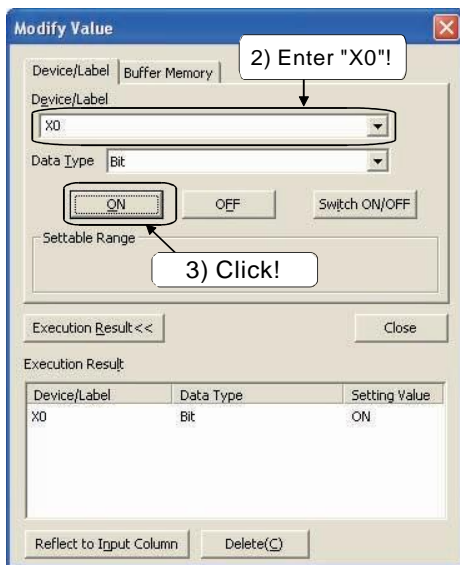
### 8.3.1 Giám sát và kiểm tra trạng thái biến

Phần này diễn giải làm thế nào để quan sát trạng thái biến, bật biến bit bật/tắt bằng cường bức và thay đổi giá trị biến từ.

- (1) Bật biến bit bật/tắt bằng cường bức  
Ở ví dụ dưới đây, "X0" bị cường bức bật lên.



- 1) Click [Debug] → [Modify Value].



- 2) Hộp thoại Modify Value hiện ra. Nhập đầu vào "X0" vào danh sách "Device/Label".
- 3) Click nút  để bật "X0" lên bằng cường bức



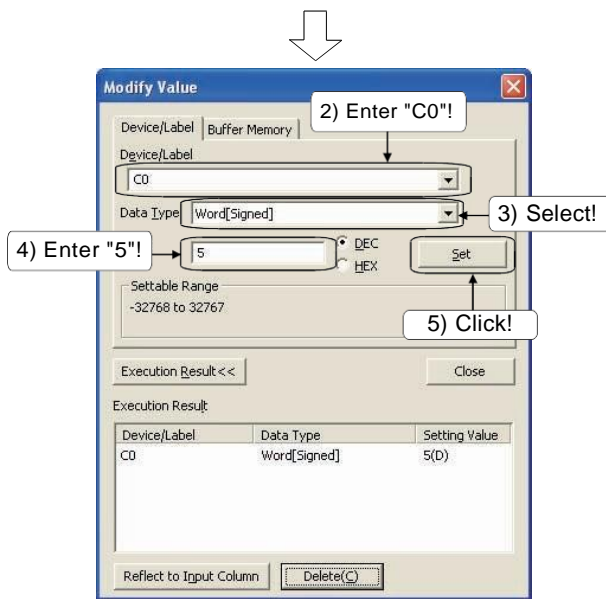
- 4) Kết quả của biến được bật lên được phản ánh trên màn hình giám sát ladder.

(2) Thay đổi giá trị biến từ

Ở ví dụ dưới đây, giá trị biến từ "C0" được đổi thành "5".



1) Click [Debug] → [Modify Value].

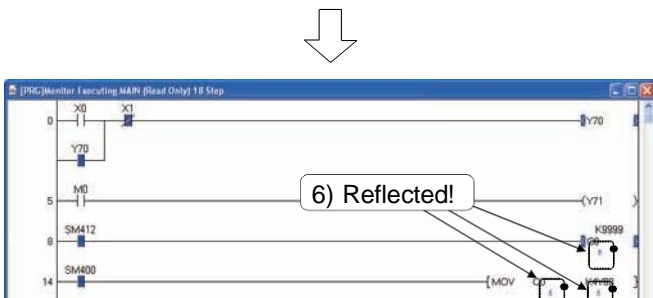


2) Hộp thoại Modify Value hiện ra. Nhập đầu vào "C0" vào danh sách "Device/Label".

3) Chọn "Word[Signed]" từ danh sách "Data Type".

4) Nhập "5" vào cột "Value".

5) Sau khi cài đặt hoàn tất, click vào nút **Set** để thay đổi giá trị hiện tại bằng vũ lực của C0 thành 5



6) Thay đổi giá trị của "C0" thành "5" được phản ánh trên màn hình giám sát ladder.



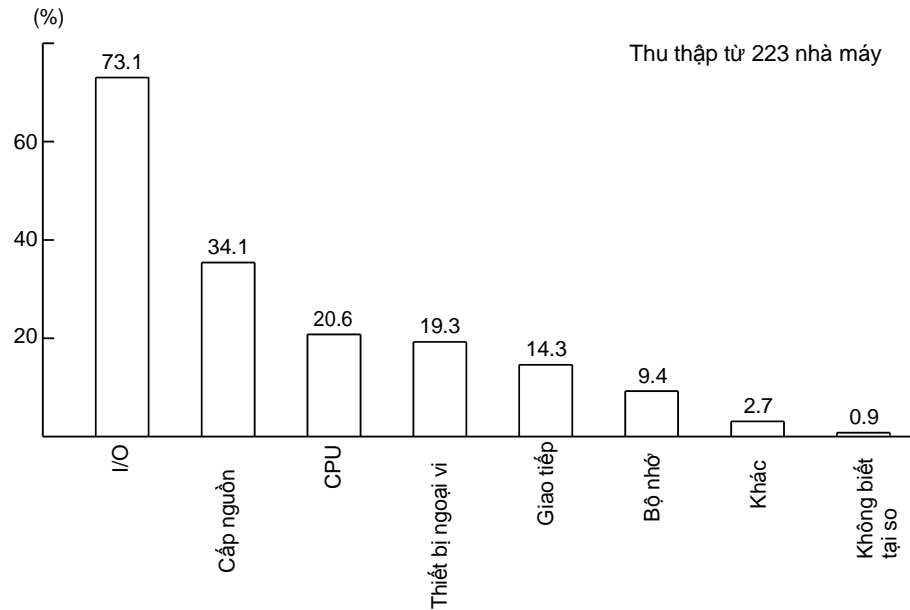
## Chương 9 Bảo dưỡng

### 9.1 Lỗi phổ biến

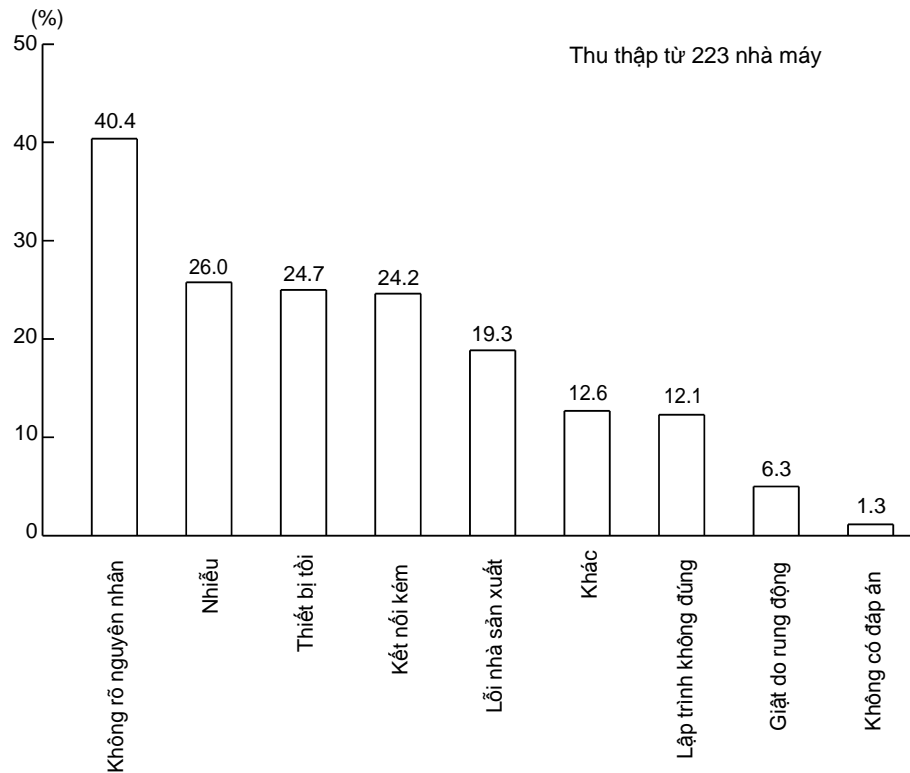
Biểu đồ cột dưới đây chỉ ra tỉ lệ của thành phần lỗi và nguyên nhân của lỗi bộ điều khiển khả trình.

[Nguồn: điều tra tạo bởi JEMA (The Japan Electrical Manufacture's Association)]

Hình 9.1 Thành phần lỗi trên khối điều khiển khả trình (cho phép nhiều đáp án)



Hình 9.2 Nguyên nhân lỗi của khối điều khiển khả trình (cho phép nhiều đáp án)



9.2 Bảo dưỡng

Để giữ khối điều khiển khả trình trong điều kiện hoạt động tốt nhất, làm theo kiểm tra dưới đây hàng ngày và theo chu kì.

(1) Kiểm tra hàng ngày

Bảng danh sách dưới đây liệt kê những mục cần được kiểm tra hàng ngày.

Bảng 9.1 Kiểm tra hàng ngày

Mục	Mục kiểm tra	Nội dung kiểm tra	Tiêu chí đánh giá	Thang đo	
1	Lắp đặt của đế cắm	Kiểm tra vít cố định không bị lỏng và bảo vệ không bị rơi ra	Vít và bảo vệ phải được gắn vào an toàn	Vặn chặt lại vít.	
2	Lắp đặt module I/O	Kiểm tra modul không bị sai vị trí và móc cố định module được gắn an toàn	Móc cố định module phải được gắn và lắp đặt an toàn.	Gắn móc cố định module an toàn. Vặn chặt vít	
3	Điều kiện kết nối	Kiểm tra độ lỏng của vít tiếp điểm.	Vít không bị lỏng.	Vặn chặt lại vít tiếp điểm	
		Kiểm tra khoảng cách giữa những loại đầu nối không hàn	Khoảng cách chính xác phải được cung cấp giữa những đầu nối không hàn.	Đặt khoảng cách chính xác.	
		Kiểm tra phần kết nối của cáp	Kết nối không bị lỏng	Vặn chặt lại vít cố định bộ kết nối	
4	Module chỉ LED	Module cấp nguồn "POWER" LED	Kiểm tra LED được bật.	LED phải được bật. (Lỗi nếu LED bị tắt)	Xem lại QCPU (Q mode) User's Manual.
		CPU "RUN" LED	Kiểm tra LED được bật ở trạng thái RUN	LED phải được bật. (Lỗi nếu LED bị tắt)	
		CPU "ERROR" LED	Kiểm tra LED bị tắt	LED phải được tắt (Lỗi nếu LED được bật hoặc nháy)	
		CPU "BAT.ARM" LED	Kiểm tra LED bị tắt	LED phải được tắt (Lỗi nếu LED được bật)	
		LED đầu vào	Kiểm tra LED được bật và tắt	LED phải được bật khi nguồn đầu vào được bật LED phải được tắt khi nguồn đầu vào được tắt (Lỗi nếu LED không bật hoặc tắt như chỉ thị bên triền)	
		LED đầu ra	Kiểm tra LED được bật và tắt.	LED phải được bật khi nguồn đầu ra được bật LED phải được tắt khi nguồn đầu ra được tắt (Lỗi nếu LED không bật hoặc tắt như chỉ thị bên triền)	

(2) Kiểm tra định kì

Bảng danh sách dưới đây liệt kê các mục phải được kiểm tra một hoặc hai lần mỗi nửa năm cho đến một năm. Khi biến được đặt lại vị trí hoặc chỉnh sửa, hoặc bố trí đi dây bị thay đổi, thực hiện kiểm tra này.

Bảng 9.2 Kiểm tra định kì

Mục	Mục kiểm tra	Nội dung kiểm tra	Tiêu chí đánh giá	Thang đo	
1	Nhiệt độ môi trường	Đo nhiệt độ và độ ẩm với nhiệt kế và ẩm kế Đo nồng độ khí ăn mòn.	0 đến 55 °C	Khi khối điều khiển khả trình được sử dụng trên bảng mạch, độ ẩm trên bảng mạch là độ ẩm môi trường.	
	Độ ẩm môi trường		5 đến 95% RH *1		
	Môi trường		Khí ăn mòn không được phép có		
2	Điện áp cấp nguồn	Đo điện áp xuyên suốt các đầu cuối của 100/200VAC.	85 đến 132VAC	Thay đổi khối cấp nguồn.	
			170 đến 264VAC		
3	Chỉ thị	Lồng, làm sạch	Di chuyển module ra để kiểm tra độ lỏng và làm sạch	Module phải được lắp đặt an toàn.	Vận chặt ví nếu CPU, I/O hoặc module cấp nguồn bị lỏng, cố định nó bằng đinh.
		Sự dính vào của bụi và vật liệu ngoài	Kiểm tra bằng mắt	Bụi và vật liệu ngoài không được phép có	
4	Điều kiện kết nối	Lồng vít đầu nối	Vận chặt vít với một tuốc-nơ-vít	Vít phải chặt	Vận chặt vít kết nối
		Khoảng cách giữa những loại đầu nối không hàn	Kiểm tra bằng mắt	Khoảng cách chính xác phải được đặt giữa những đầu nối không hàn	Đặt khoảng cách chính xác.
		Lồng bộ kết nối	Kiểm tra bằng mắt.	Bộ kết nối không được lỏng	Vận chặt vít cố định bộ kết nối.
5	Pin	Kiểm tra SM51 hoặc SM52 bị tắt với GX Works2.	(Bảo dưỡng phòng ngừa)	Ngay cả khi dung lượng pin thấp không hiển thị, thay thế pin bằng một pin mới nếu vòng đời hoạt động của pin bị hết hạn.	
6	Sản phẩm phụ tùng	Cài đặt sản phẩm trên một khối điều khiển khả trình thật sự và kiểm tra hoạt động	Hoạt động phải đúng với đặc điểm kĩ thuật.	Sử dụng sản phẩm thông thường trên khối điều khiển khả trình như một sản phẩm phụ tùng	
7	Kiểm tra chương trình được lưu	So sánh chương trình được lưu với chương trình đang chạy	Hai chương trình phải giống nhau	Đúng nếu tìm thấy bất cứ sự khác nhau nào.	
8	Quạt (bộ trao đổi nhiệt)	Trạng thái vòng quay Tiếng vòng quay Tắc	Quạt phải quay mà không có âm thanh bất thường nào Quạt phải quay mà không bị tắc.	Thay thế nếu có bất kì lỗi nào tìm thấy. Sạch	
9	I/O tương tự	Kiểm tra giá trị offset/gain (ngưỡng/độ lợi)	Giá trị phải giống với tham số kĩ thuật (giá trị thiết kế)	Đúng nếu tìm thấy bất cứ sự khác nhau nào	

\*1: Khi module AnS Series được sử dụng trong hệ thống, tiêu chí đánh giá sẽ từ 10 đến 90% RH.

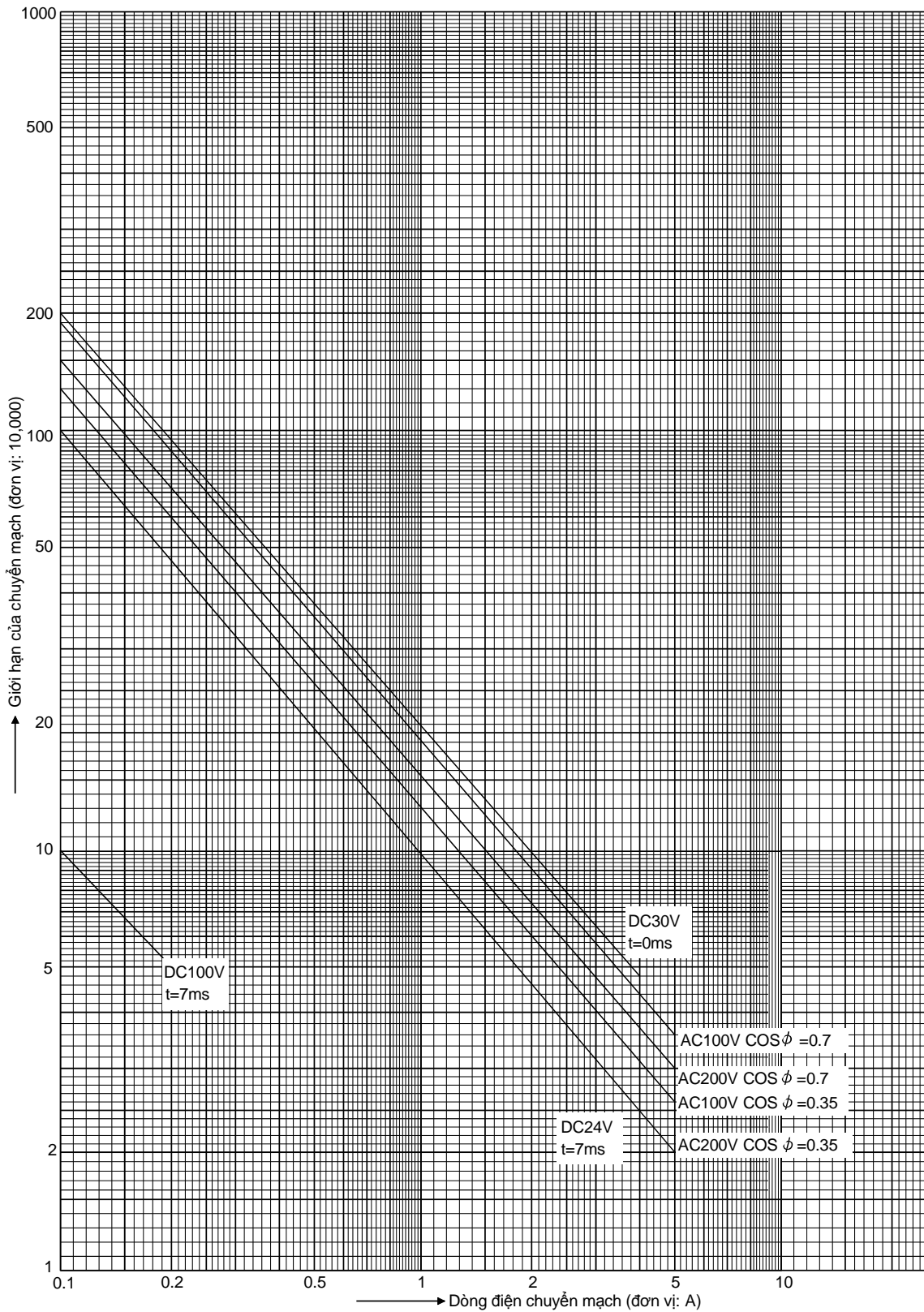
### 9.3 Sản phẩm tiêu hao

Pin dự phòng trên khối điều khiển khả trình là Sản phẩm tiêu hao.

9.4 Vòng đời của rơle đầu ra

Rơle đầu ra của module bị tiêu tốn bởi hoạt động chuyển mạch.

Một rơle được lắp trực tiếp lên bảng in của module đầu ra là cần thiết để thay thế module đầu ra của chính nó sau khi tiêu thụ.



Hình 9.3 Đặc tính tuổi thọ của công tắc rơle đầu ra (QY10, QY18A)

## 9.5 Sản phẩm dự phòng

Sản phẩm dự phòng rất dễ đặt mua qua trung tâm dịch vụ Mitsubitshi hoặc đại diện Mitsubitshi ở Nhật Bản. Do đó, sản phẩm dự phòng có thể được chuẩn bị ngay cả sau khi có sự cố. Tuy nhiên, lưu ý rằng sản phẩm liên quan đến bên ngoài như xuất sản phẩm, sản phẩm dự phòng phải được gửi trước.

- (1) Loại dễ thay thế  
Thay thế module loại khối block rất dễ. Chỉ cần thay thế module khi nó có lỗi
- (2) Loại bộ nhớ  
Để sử dụng thẻ nhớ RAM cơ bản hoặc SRAM cơ bản, cần có pin dự phòng. ROMs tiêu chuẩn, card Flash, và card ATA không cần pin để sử dụng, bên cạnh đó, bộ nhớ này ngăn cản chương trình thay đổi do lỗi liên quan đến con người. Bộ nhớ này nên được sử dụng trên sản phẩm để xuất
- (3) Giảm số loại module  
Giảm số loại module rất hiệu quả cho việc giảm số lượng loại sản phẩm dự phòng.
- (4) Đặt trước điểm I/O  
Bằng việc không sử dụng hết điểm I/O trên 16-, 32-, và 64-điểm module I/O nhưng đặt trước 10% đến 20% của chúng, có thể chỉ cần thay đổi đi dây và chương trình (tín hiệu I/O) thay vì thay thế module lỗi với một module dự phòng khi mà module dự phòng không có.
- (5) Tạo tài liệu  
Bởi vì chương trình tuần tự rất dễ chỉnh sửa, sự không nhất quán giữa một chương trình đang hoạt động và tài liệu có thể xảy ra (ví dụ biểu đồ ladder, danh sách chương trình).  
Tiếp tục cập nhật tài liệu.  
Để làm điều đó, sử dụng máy in sẽ rất hiệu quả.
- (6) Am hiểu thiết bị ngoại vi  
Thuần thực thiết bị ngoại vi như một máy tính cá nhân, GX Works2 giúp phục hồi nhanh từ một sự cố.

(7) Sản phẩm dự phòng

Table 9.3 Spare products

	Tên sản phẩm	Số lượng	Điểm
1	Pin	Một hoặc hai	Vòng đời của pin lithium là khoảng 5 năm. Vì thế, không nên lưu trữ trong kho thường xuyên nhưng khi cần thiết thì phải đặt mua. Tuy nhiên, giữ trong kho <b>một hoặc hai phòng trừ sự cố.</b>
2	Module I/O	Một với mỗi loại module	Lưu ý rằng module I/O thường gặp lỗi trong quá trình kiểm tra.. Cũng lưu ý rằng công tắc module đầu ra được sử dụng trong thời gian dài.
3	Module CPU	Một với mỗi loại model	Module CPU và thẻ nhớ là những thành phần cốt lõi của một bộ điều khiển khả trình, nghĩa là một lỗi của nó sẽ khiến cả hệ thống ngừng trệ.
4	Thẻ nhớ	Một với mỗi loại model	
5	Module cấp nguồn	Một với mỗi loại model	Tương tự như trên. Khi nhiệt độ của module cấp nguồn tăng nhanh, và nhiệt độ môi trường tăng cao có thể làm ngắn đi thời gian sống của nó.

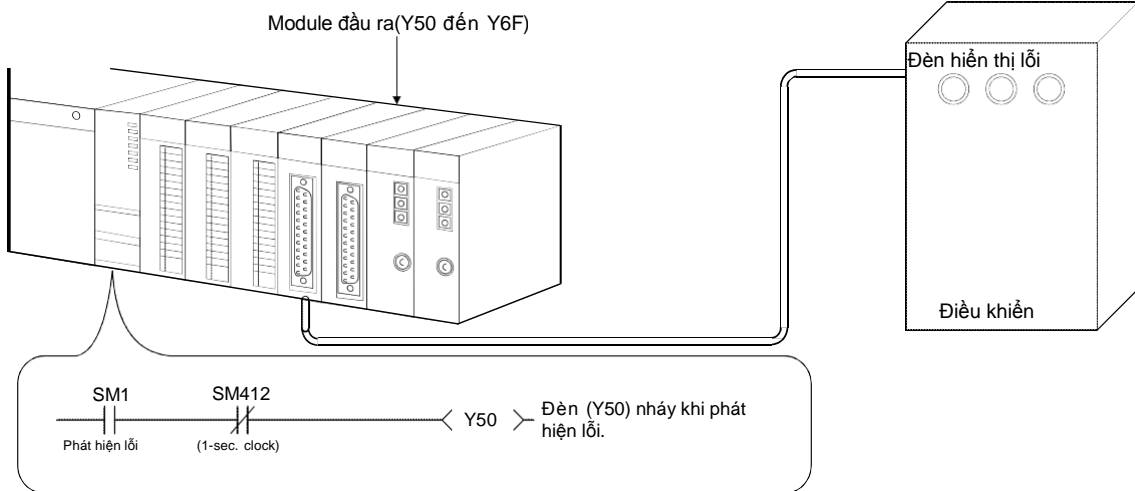
## 9.6 Sử dụng sản phẩm hỗ trợ

Dưới đây là ví dụ của sản phẩm hỗ trợ trong trường hợp hệ thống hoặc thiết bị sử dụng khối điều khiển khả trình thông báo một lỗi tìm thấy hoặc trạng thái hoạt động đến người điều hành hoặc người bảo dưỡng trong suốt quá trình điều khiển tự động.

### 1. Hiển thị lỗi sử dụng đèn thông thường

Kết nối đèn lỗi đến module đầu ra của khối điều khiển khả trình thế nên đèn nhấp khi phát hiện một lỗi.

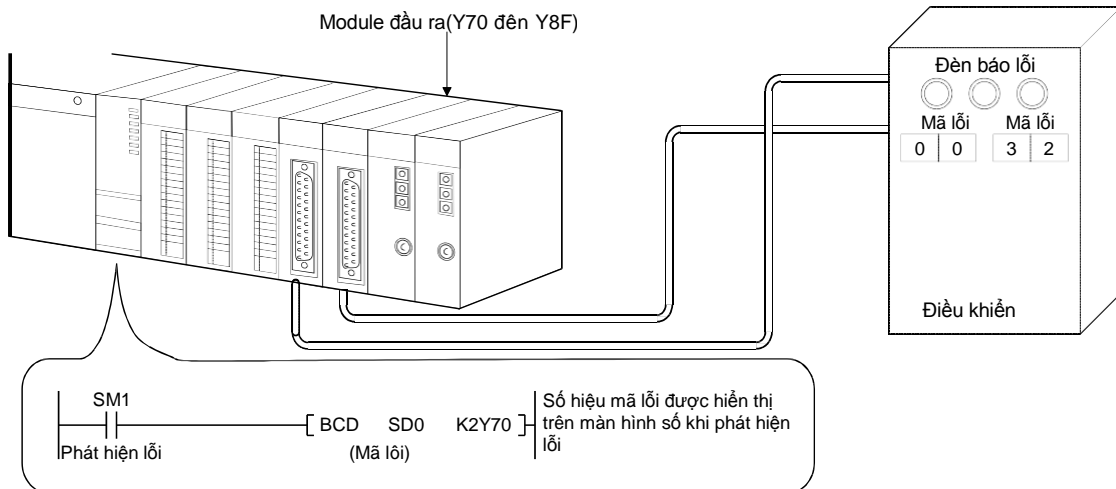
#### Đèn nhấp



### 2. Hiển thị mã lỗi trên màn hình số thông thường

Kết nối màn hình số đến module đầu ra của khối điều khiển khả trình vì thế nên số mã lỗi của lỗi phát hiện được sẽ được chỉ ra trên màn hình kỹ thuật số.

#### Hiển thị số học



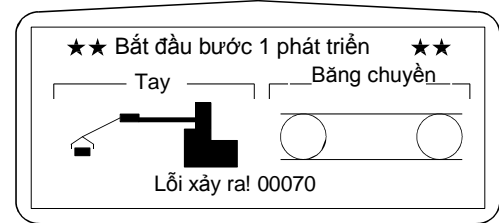
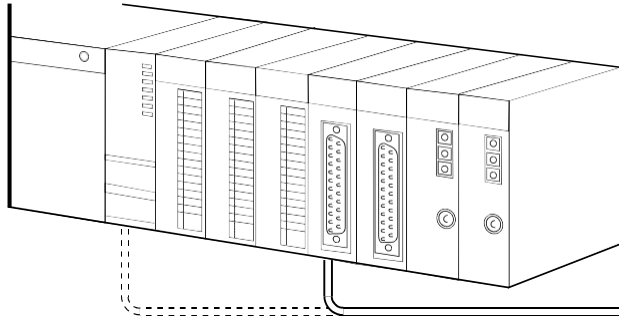
#### Lưu ý

Chương trình trên không thể thực thi khi có một lỗi báo dừng..

3. **Hiển thị nội dung của lỗi phát hiện trên màn hình**

Thông tin chi tiết lỗi của khối điều khiển khả trình có thể được hiển thị trên một màn hình CRT, plasma, và tinh thể lỏng..

Màn hình hiển thị



MELSEC-Q trợ giúp rất nhiều loại GOTs (Màn hình hiển thị đồ họa).

Cùng với chức năng hiển thị lỗi, GOTs có rất nhiều chức năng hữu dụng như giám sát đồ họa, giám sát ladder, giám sát thiết bị, công tắc bảng chạm, và chức năng in .

(Xem lại mục lục để biết thêm chi tiết.)



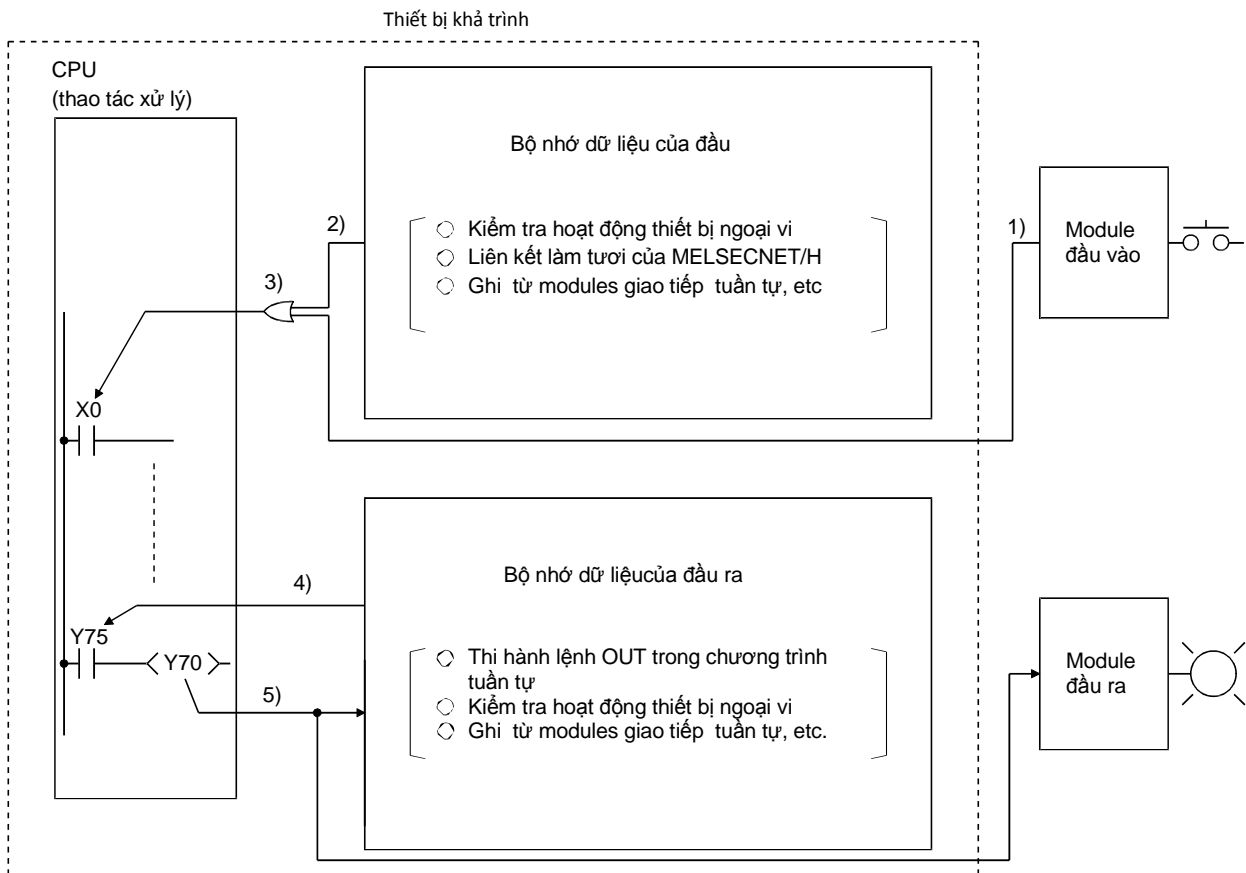
## Phụ lục

### Phụ lục 1 Chế độ điều khiển I/O

CPU hỗ trợ hai loại chế độ điều khiển: Chế độ điều khiển trực tiếp và chế độ điều khiển tự động làm mới.

#### Phụ lục 1.1 Chế độ điều khiển trực tiếp

Trong chế độ trực tiếp, tín hiệu đầu vào được nhập đến thiết bị điều khiển khả trình mỗi khi chúng được nhập vào và được xử lý như thông tin nhập vào. Kết quả hoạt động của chương trình được xuất ra bộ nhớ ngoài và module đầu ra. Đồ thị sau đây cho thấy luồng của I / O dữ liệu trong chế độ trực tiếp.



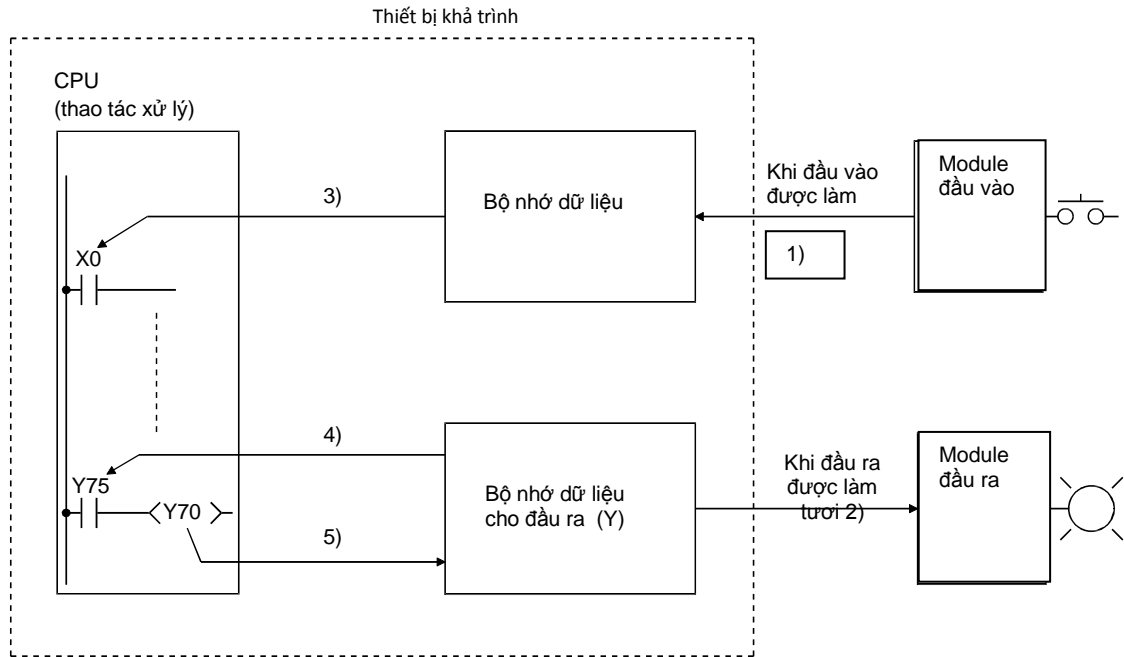
- Khi câu lệnh công tắc dữ liệu đầu vào được thi hành: Hoạt động OR được thi hành ở thông tin đầu vào 1), từ input module và thông tin đầu vào 2) trong bộ nhớ dữ liệu. Sau đó kết quả được dùng làm thông tin đầu vào 3) thực hiện tại chương trình tuần tự.
- Khi câu lệnh công tắc dữ liệu đầu ra được thi hành: Đầu ra thông tin 4) được đọc từ bộ nhớ dữ liệu cho đầu ra (Y), và chương trình tuần tự được thi hành.
- Khi câu lệnh đầu ra OUT được thi hành: Kết quả hoạt động 5) của chương trình tuần tự được xuất đến output module, và được lưu trữ trong bộ nhớ dữ liệu cho đầu ra (Y).
- Khi QCPU thực hiện I/O trong chế độ trực tiếp, chương trình tuần tự sử dụng DX cho đầu vào và DY cho đầu ra.
- Khi QCPU thi hành I / O trong chế độ trực tiếp, chương trình tuần tự sử dụng Sx truyền vô tuyến tầm xa cho dữ liệu nhập vào DX và cho đầu ra DY.

Phụ lục 1.2 Chế độ tự động làm mới

Ở chế độ tự động làm mới, tất cả các thay đổi xảy ra trong module đầu vào được xuất đến đầu vào của bộ nhớ dữ liệu trong thiết bị khả trình CPU trước mỗi lần quét. Dữ liệu trong bộ nhớ dữ liệu được bộ nhớ dữ liệu được dùng cho hoạt động này.

Kết quả hoạt động của chương trình cho đầu ra (Y) được lưu đến đầu ra bộ nhớ dữ liệu ở mọi hoạt động. Tất cả dữ liệu lưu trữ trong đầu ra bộ nhớ dữ liệu là một khối - dữ liệu ra đến module đầu ra sau khi triển khai hướng dẫn END.

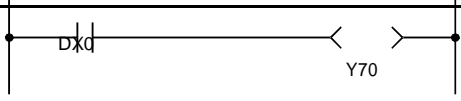
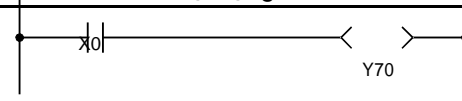
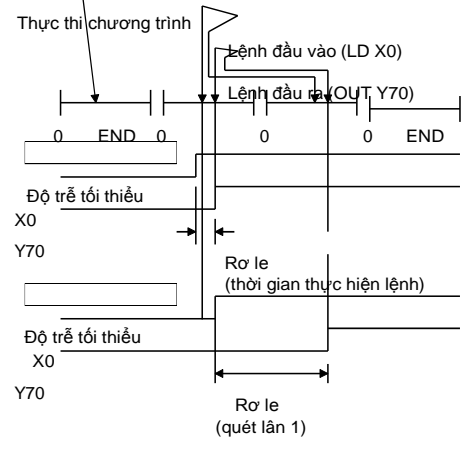
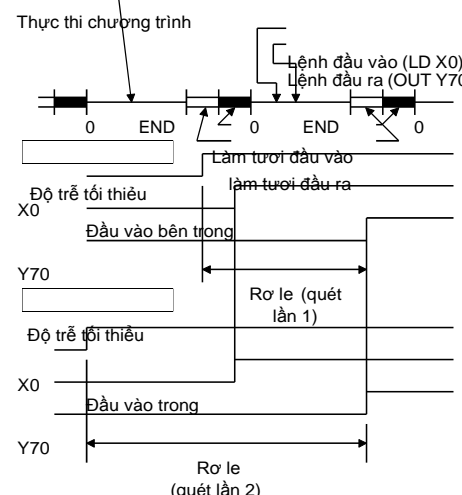
Đồ thị sau đây cho thấy luồng của I / O dữ liệu trong chế độ tự động làm mới.



- Đầu vào chế độ tự động làm mới  
Dữ liệu đầu vào trong module đầu vào là khối - đọc 1 ) trước khi thi hành của bước 0, và lưu trữ đến bộ nhớ dữ liệu cho đầu vào ( X ).
- Đầu ra làm tươi  
Dữ liệu 2 ) trong bộ nhớ dữ liệu cho đầu ra ( Y ) là khối - đầu ra đến module đầu ra trước khi thi hành của bước 0.
- Khi lệnh địa chỉ công tắc đầu vào được thi hành :  
Dữ liệu đầu vào được đọc từ bộ nhớ dữ liệu cho đầu vào ( X ) 3 ), và chương trình tuần tự được thi hành.
- Khi lệnh địa chỉ công tắc đầu ra được thi hành :  
Dữ liệu đầu ra 4 ) được đọc từ bộ nhớ dữ liệu cho đầu ra ( Y ), và chương trình tuần tự được thi hành.
- Khi đầu ra lệnh OUT được thi hành :  
Kết quả thao tác của chương trình tuần tự 5 ) được lưu trữ trong bộ nhớ dữ liệu cho đầu ra ( Y ).

Phụ lục 1.3 So sánh giữa chế độ trực tiếp và chế độ tự động làm mới cho đầu ra ( Y ).

Ở ví dụ sơ đồ bậc ladder cho dưới, bật đầu vào X0 bật đầu ra Y70.

Mục	Chế độ trực tiếp	Chế độ tự động làm mới
1. Ví dụ ladder		
2. Độ trễ phản hồi từ lúc đầu vào được thay đổi đến khi đầu ra thay đổi theo.	 <p>Thực thi chương trình</p> <p>Lệnh đầu vào (LD X0)</p> <p>Lệnh đầu ra (OUT Y70)</p> <p>Độ trễ tối thiểu X0</p> <p>Độ trễ tối thiểu Y70</p> <p>Rơ le (thời gian thực hiện lệnh)</p> <p>Rơ le (quét lần 1)</p> <ul style="list-style-type: none"> <li>• Thời gian trễ khoảng từ 0 (chỉ trong thời gian thực hiện của lệnh) đến 1 quét.</li> <li>• Thời gian trễ là từ 0 đến 1 quét.</li> </ul>	 <p>Thực thi chương trình</p> <p>Lệnh đầu vào (LD X0)</p> <p>Lệnh đầu ra (OUT Y70)</p> <p>Độ trễ tối thiểu X0</p> <p>Độ trễ tối thiểu Y70</p> <p>X0</p> <p>Y70</p> <p>Đầu vào bên trong</p> <p>Đầu vào trong</p> <p>Làm tươi đầu vào</p> <p>Làm tươi đầu ra</p> <p>Rơ le (quét lần 1)</p> <p>Rơ le (quét lần 2)</p> <ul style="list-style-type: none"> <li>• Thời gian trễ khoảng từ 1 đến 2 quét.</li> <li>• Thời gian trễ là từ 1 đến 2 quét.</li> </ul>
3. Thời gian thực hiện lệnh I / O	<ul style="list-style-type: none"> <li>• Chế độ trực tiếp cần thời gian dài hơn chế độ làm tươi vì một thiết bị khả trình truy cập module I / O</li> </ul>	<ul style="list-style-type: none"> <li>• Nói chung, chỉ cần thời gian ngắn vì thiết bị khả trình truy cập bộ nhớ dữ liệu.</li> </ul>
4. thời gian quét	<ul style="list-style-type: none"> <li>• Thời gian quét là dài hơn thời gian thực hiện của lệnh I / O.</li> <li>• Thời gian quét thực tế là thời gian thực hiện chương trình.</li> </ul>	<ul style="list-style-type: none"> <li>• Thời gian quét ngắn hơn thời gian thực hiện lệnh I / O.</li> <li>• Thời gian quét thực tế là tổng thời gian của thực hiện chương trình, chuyển giao dữ liệu nhập vào, và chuyển giao dữ liệu đầu ra.</li> </ul>

Phụ lục 2 Rơ le đặc biệt

Mạch Rơ le đặc biệt ( SM ) là mạch rơ le bên trong có tác dụng là cố định ở bộ điều khiển khả trình. Vì lý do này, thanh ghi đặc biệt không thể sử dụng giống như những thanh ghi bên trong khác được dùng trong chương trình tuần tự. Tuy nhiên, bit của rơ le đặc biệt có thể bật hoặc tắt khi cần thiết để điều khiển module CPU.

Sau đây là cách để đọc khoản mục trong danh sách.

Để biết thêm chi tiết của mạch rơ le đặc biệt, tham khảo bản hướng dẫn sử dụng Phần cứng QCPU, Bảo trì và kiểm duyệt.

Mục	Mô tả
Số	• Biểu thị các số rơ le đặc biệt .
Tên	• Biểu thị các tên rơ le đặc biệt
Ý nghĩa	• Biểu thị nội dung của rơ le đặc biệt.
Giải thích	• Giải thích nội dung đặc điểm của rơ le đặc biệt
Thiết lập bởi (khi thiết lập)	<ul style="list-style-type: none"> <li>• Biểu thị các thiết lập bên và thiết lập thời gian cho thanh ghi đặc biệt &lt;thiết lập bởi&gt;</li> <li>S : thiết lập bởi hệ thống</li> <li>U : thiết lập bởi người sử dụng (trong chương trình tuần tự hoặc kiểm tra quá trình hoạt động của các thiết bị ngoại vi)</li> <li>S/U : thiết lập bởi hệ thống và người dùng.</li> <li>&lt;thiết lập khi&gt; Chỉ biểu thị khi hệ thống đã thiết lập xong.</li> <li>Mọi xử lý END : Thiết lập trong mỗi xử lý END</li> <li>Ban đầu : Thiết lập trong xử lý ban đầu (sau khi nguồn bật hoặc trạng thái chuyển từ Stop sang Run )</li> <li>Trạng thái thay đổi : Thiết lập khi trạng thái hoạt động được thay đổi</li> <li>Lỗi : thiết lập nếu lỗi xảy ra</li> <li>Thực thi lệnh : Thiết lập khi lệnh được thi hành</li> <li>Yêu cầu : Thiết lập khi được yêu cầu bởi người dùng ( sử dụng rơ le đặc biệt )</li> <li>Khi hệ thống được chuyển mạch : Thiết lập khi hệ thống được chuyển mạch ( giữa hệ thống kiểm soát và hệ thống dự phòng)</li> </ul>
WACPU M9□□□ trong ứng	<ul style="list-style-type: none"> <li>• Biểu thị rơ le đặc biệt (M9□□□) Được hỗ trợ bởi ACPU. ("M9□□□ định dạng thay đổi " biểu thị cho 1 ứng dụng đã thay đổi. Không tương thích với Q00J/Q00/Q01, và QnPRH.)</li> <li>• "New" biểu thị thêm một Q - chuỗi CPU.</li> </ul>
i t e CPU tương ứng , T h	<ul style="list-style-type: none"> <li>Biểu thị CPU module hỗ trợ rơ le đặc biệt</li> <li>QCPU : tất cả Q-chuỗi CPU modules</li> <li>Q00J/Q00/Q01 : Mô hình cơ bản</li> <li>QCPU Qn(H) : Mô hình hiệu suất cao QCPU</li> <li>QnPH : Quy trình CPU</li> <li>QnPRH : Dự thừa CPU</li> <li>QnU : Mô hình phổ biến QCPU</li> <li>Tên module CPU : Chỉ mô hình CPU xác định (Ví dụ: Q02U)</li> </ul>

TTTham khảo tài liệu hướng dẫn của này:

- Đối với các mục liên quan đến mạng -> Tài liệu hướng dẫn sử dụng đối với mỗi module mạng
- Đối với chương trình SFC → Hướng dẫn lập trình MELSEC-Q/L/QnA (SFC).

GỢI Ý
<p>Không thay đổi giá trị của rơ le đặc biệt thiết lập bởi hệ thống sử dụng chương trình hoặc bằng thao tác kiểm tra.                      Làm vậy rất có thể dẫn đến kết quả hệ thống xuống hay gặp sự cố giao tiếp.</p>

Phụ lục 3 Thanh ghi đặc biệt

Thanh ghi đặc biệt ( SD ) là thanh ghi bên trong có ứng dụng là cố định ở thiết bị khả trình. Vì lý do này, thanh ghi đặc biệt không thể sử dụng giống như thanh ghi bên trong khác được dùng trong chương trình trình tuần tự. Tuy nhiên, dữ liệu có thể được ghi cho thanh ghi đặc biệt để điều khiển module CPU khi cần. Dữ liệu được lưu trữ trong dạng nhị phân nếu chưa xác định.

Sau đây là cách để đọc khoản mục trong danh sách.

Để biết thêm chi tiết của mạch rơ le đặc biệt, tham khảo bản hướng dẫn sử dụng Phần cứng QCPU, Bảo trì và kiểm duyệt.

Mục	Mô tả
Số	• Biểu thị các số rơ le đặc biệt .
Tên	• Biểu thị các tên rơ le đặc biệt
Ý nghĩa	• Biểu thị nội dung của rơ le đặc biệt.
Giải thích	• Giải thích nội dung đặc điểm của rơ le đặc biệt
Thiết lập bởi (khi thiết lập)	<ul style="list-style-type: none"> <li>• Biểu thị các thiết lập bên và thiết lập thời gian cho thanh ghi đặc biệt &lt;thiết lập bởi&gt;</li> <li>S : thiết lập bởi hệ thống</li> <li>U : thiết lập bởi người sử dụng (trong chương trình tuần tự hoặc kiểm tra quá trình hoạt động của các thiết bị ngoại vi)</li> <li>S/U : thiết lập bởi hệ thống và người dùng.</li> <li>&lt;thiết lập khi&gt; Chỉ biểu thị khi hệ thống đã thiết lập xong.</li> <li>Mọi xử lý END : Thiết lập trong mỗi xử lý END</li> <li>Ban đầu : Thiết lập trong xử lý ban đầu (sau khi nguồn bật hoặc trạng thái chuyển từ Stop sang Run )</li> <li>Trạng thái thay đổi : Thiết lập khi trạng thái hoạt động được thay đổi</li> <li>Lỗi : thiết lập nếu lỗi xảy ra</li> <li>Thực thi lệnh : Thiết lập khi lệnh được thi hành</li> <li>Yêu cầu : Thiết lập khi được yêu cầu bởi người dùng ( sử dụng rơ le đặc biệt )</li> <li>Khi hệ thống được chuyển mạch : Thiết lập khi hệ thống được chuyển mạch ( giữa hệ thống kiểm soát và hệ thống dự phòng)</li> </ul>
QCPU M9□□□ trong ứng dụng	<ul style="list-style-type: none"> <li>• Biểu thị rơ le đặc biệt (M9□□□) Được hỗ trợ bởi QCPU.</li> <li>("M9□□□ định dạng thay đổi " biểu thị cho 1 ứng dụng đã thay đổi.</li> <li>Không tương thích với Q00J/Q00/Q01, và QnPRH.)</li> <li>• "New" biểu thị thêm một Q - chuỗi CPU.</li> </ul>
QCPU tương ứng	<ul style="list-style-type: none"> <li>Biểu thị CPU module hỗ trợ rơ le đặc biệt</li> <li>QCPU : tất cả Q-chuỗi CPU modules</li> <li>Q00J/Q00/Q01 : Mô hình cơ bản</li> <li>QCPU Qn(H) : Mô hình hiệu suất cao QCPU</li> <li>QnPH : Quy trình CPU</li> <li>QnPRH : Dự thừa CPU</li> <li>QnU : Mô hình phổ biến QCPU</li> <li>Tên module CPU : Chỉ mô hình CPU xác định (Ví dụ: Q02U)</li> </ul>

Tham khảo tài liệu hướng dẫn tại đây

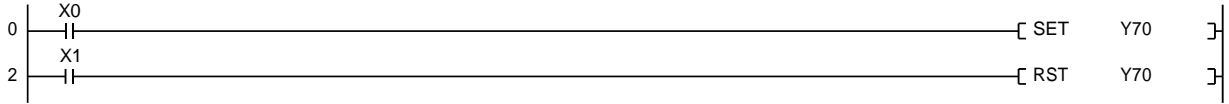
- Đối với các mục liên quan đến mạng -> Tài liệu hướng dẫn sử dụng đối với mỗi module mạng
- Đối với chương trình SFC → Hướng dẫn lập trình MELSEC-Q/L/QnA (SFC).

<b>GỢI Ý</b>
<p>Không thay đổi giá trị của rơ le đặc biệt được thiết lập bởi hệ thống sử dụng chương trình hoặc bằng thao tác kiểm tra.          Làm thế rất có thể dẫn đến kết quả hệ thống dừng hay gặp sự cố giao tiếp.</p>

Phụ lục 4 Ví dụ trình ứng dụng

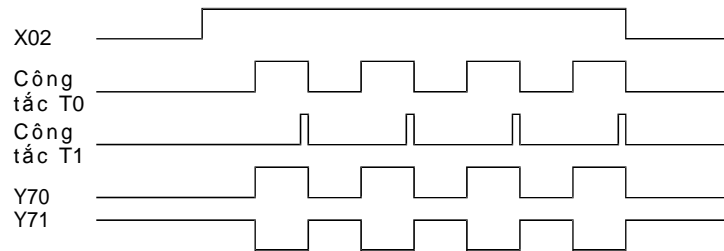
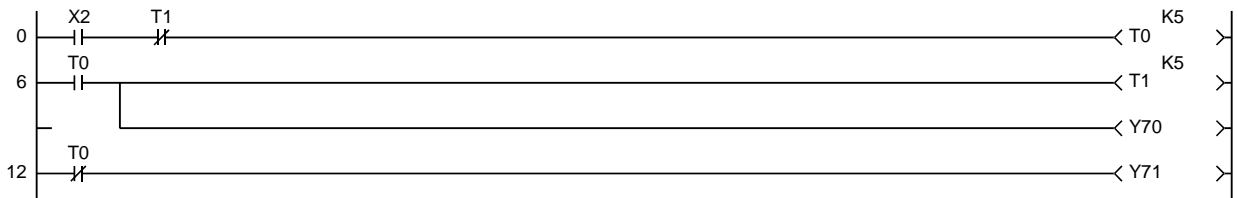
Phụ lục 4.1 Flip-flop ladder

(1) Y70 bật khi X0 được bật, và tắt khi X1 được bật.



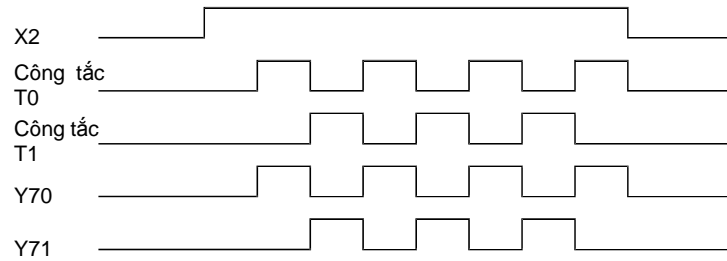
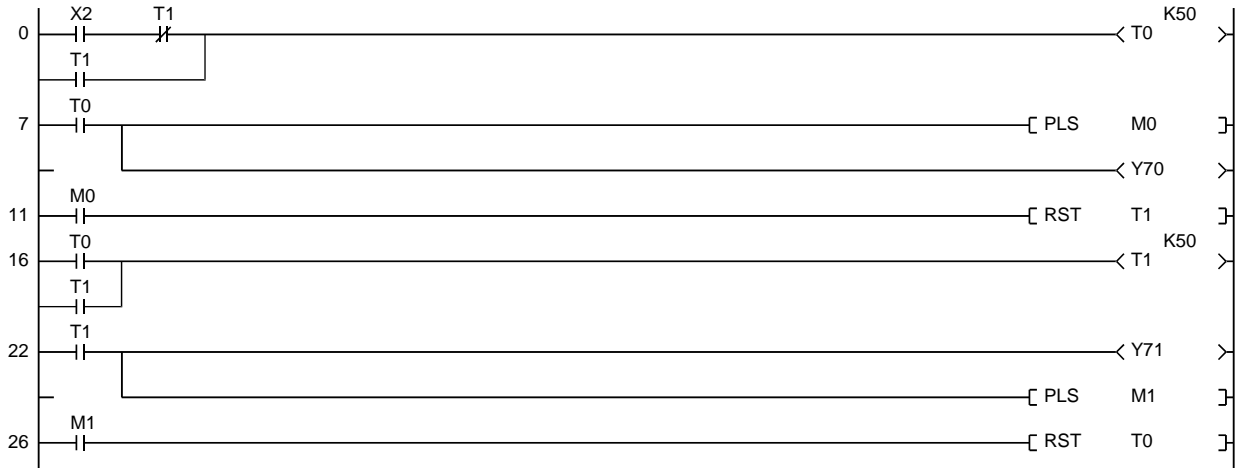
(2) Khi X2 được bật, Y71 tắt nếu Y70 đang mở, và bật nếu Y70 không làm việc. Thao tác mạch Flip-flop này được lặp lại.

Tên dự án	QA-16
Tên chương trình	MAIN



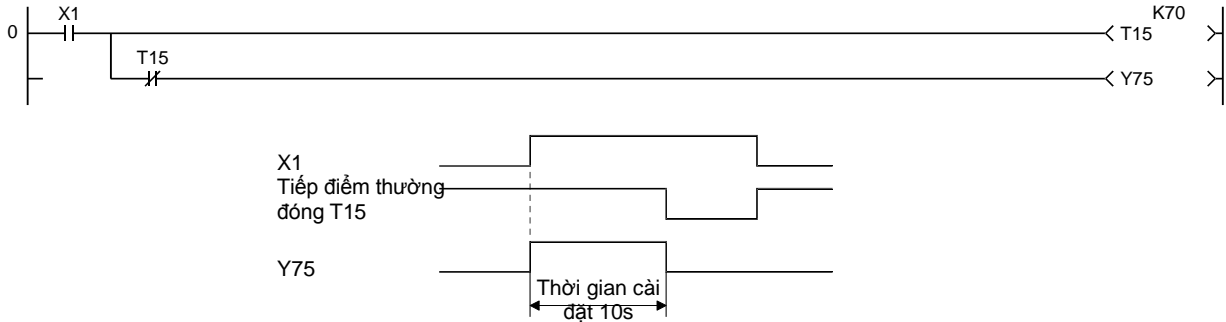
(3) Hoạt động Flip -flop bắt đầu khi X2 được bật. Trong hoạt động này, Y70 bật nếu T0 đang mở, và Y71 bật nếu T1 đang mở. ( Chu kỳ : 10s. ).

Tên dự án	QA-17
Tên chương trình	MAIN

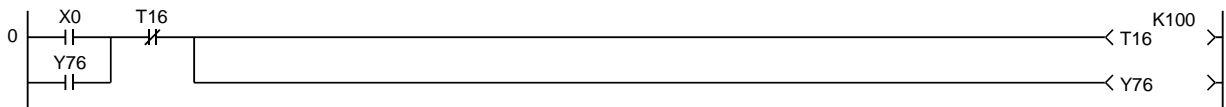


Phụ lục 4.2 Bật thời gian ngắn (one shot).

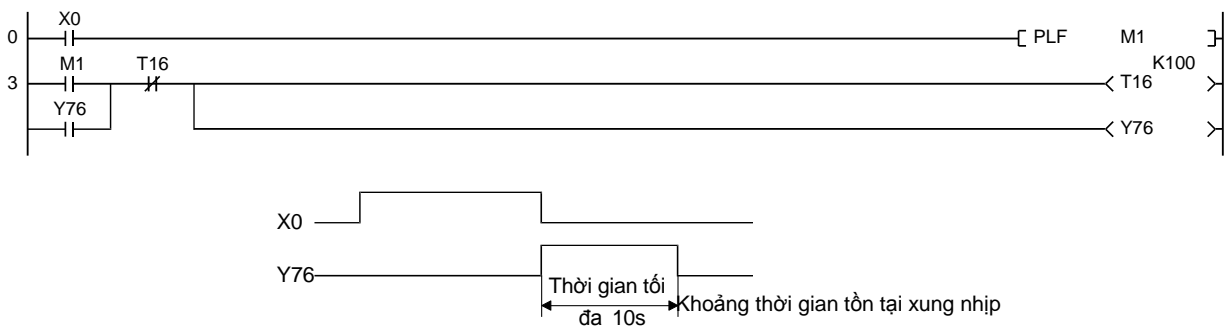
- (1) Sau khi đầu vào X1 được bật, đầu ra cũng khởi động và tiếp tục trong một khoảng thời gian xác định. ( Thời gian cho đầu vào phải dài hơn thời hạn ấn định.)



- (2) Ngay khi đầu vào X0 được bật, thì Y76 được bật trong thời gian nhất định.



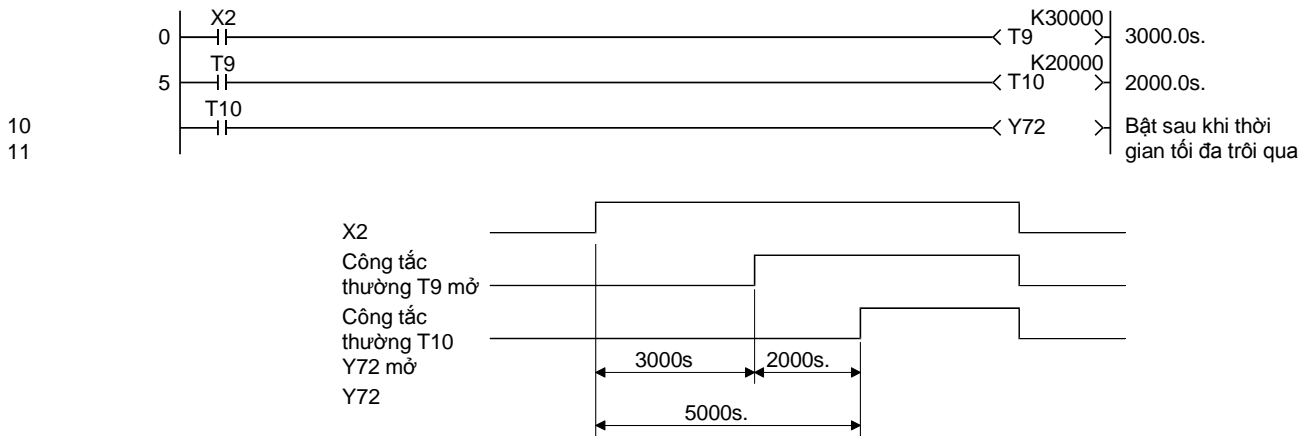
- (3) Đầu ra chắc chắn bắt đầu và tiếp tục khi X0 đầu vào được chuyển từ bật sang tắt.





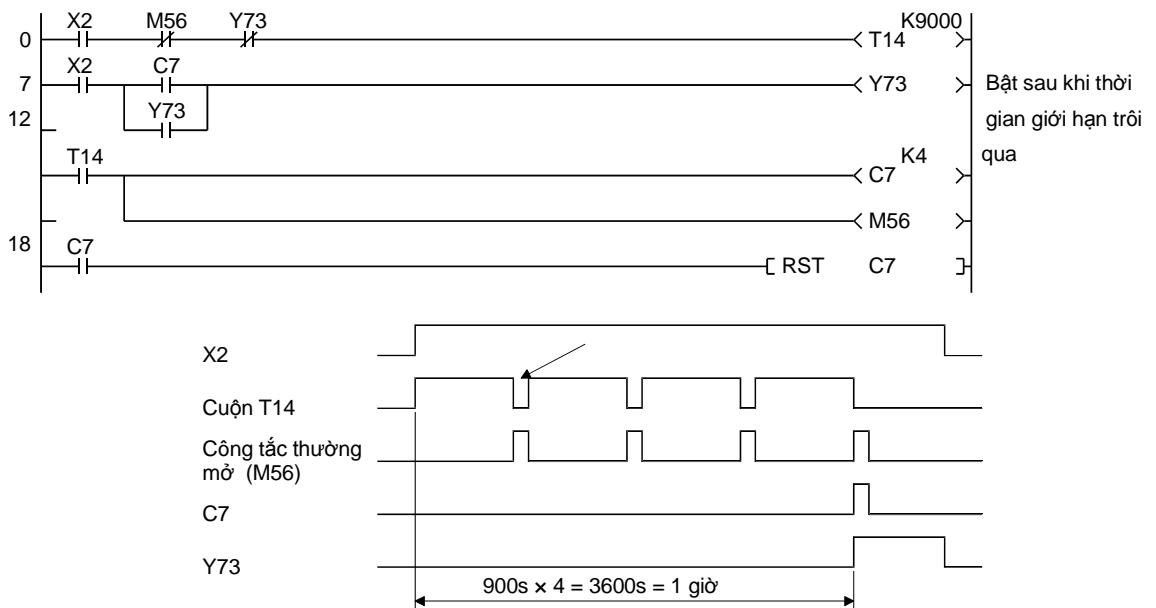
Phụ lục 4.3 Bộ định thời gian dài

(1) Thời gian cần đạt được bằng cách kết nối các bộ định thời nối tiếp.



(2) Thời gian cần đạt được bằng cách sử dụng bộ định thời và bộ đếm.  
Thời gian giới hạn của bộ định tuyến  $\times$  giá trị thiết lập của bộ đếm = Bộ định thời gian lớn ( lưu ý rằng độ chính xác của bộ định thời được tích lũy. )

Tên dự án	QA-18
Tên chương trình	MAIN



(Chú ý) Thời gian đủ được thu với bộ đếm C7, nó đếm số lần tạm ngưng của bộ định thời T14.

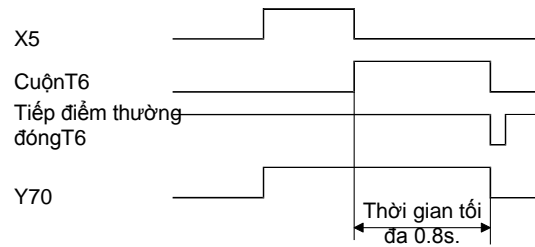
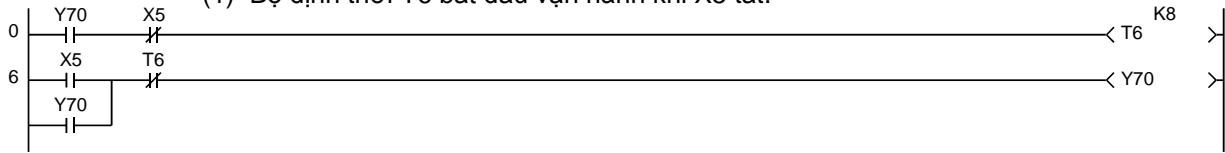
M56 thiết lập lại T14 sau khi thời gian tạm ngưng. Với C7, đầu ra Y73 là tự - tiếp năng lượng trong khi bộ đếm đang được hoạt động. Với Y73, T14 thiết lập lại và bộ đếm thời gian sau bị tạm dừng.

Phụ lục 4.4 Bộ định thời tắt trễ ( off-delay)

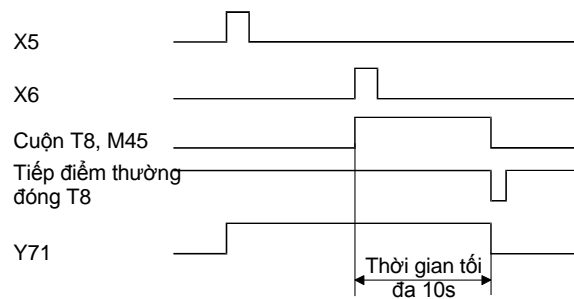
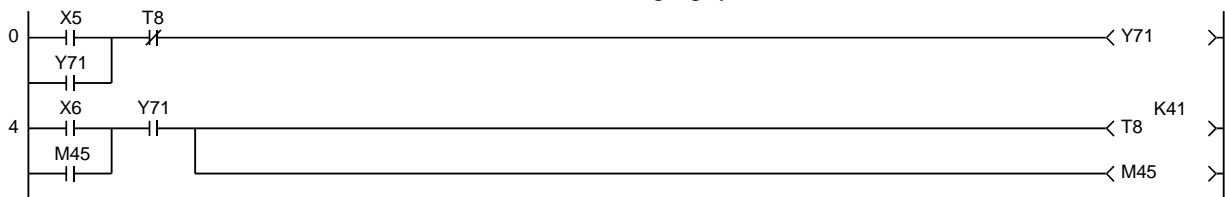
MELSEC - Q không quản lý bộ định thời tắt trễ. Cấu hình bộ định thời tắt trễ ra như

sau:

(1) Bộ định thời T6 bắt đầu vận hành khi X5 tắt.



(2) Bật X5 ngay khi thiết lập hoạt động sẵn sàng.  
Bộ định thời T8 bắt đầu hoạt động ngay khi X6 bật.

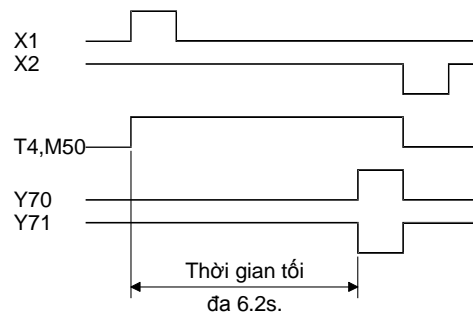
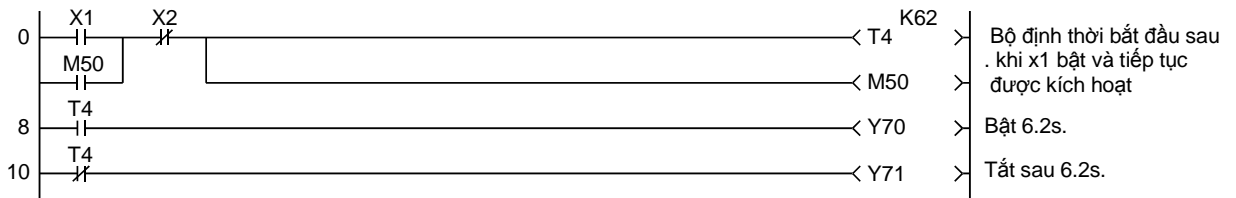


(Chú ý) Ladder bên trên sử dụng như một ladder trễ quá trình đóng khi bắt đầu vào X5 và X6.  
M45 tương đương với địa chỉ công tắc tạm thời của T8.

Phụ lục 4.5 Bộ định thời bật trễ (On-delay )

Bộ định thời bật trễ của bộ điều khiển khả trình hoạt động dễ dàng với đầu vào liên tiếp nhau. Rơ le M được sử dụng với đầu vào tạm thời.

Tên dự án	QA-19
Tên chương trình	MAIN



(Chú ý ) Ladder bên trên sử dụng như ladder on delay ngay khi đầu vào X1 và X2 được bật.

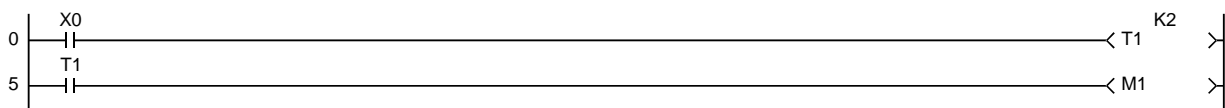
Phụ lục 4.6 Ladder lập ON-OFF

Ở ladder On - OFF, Y70 bật khi X1 được bật, và tắt khi X1 được bật lại.



Phụ lục 4.7 Chống lặp đầu vào

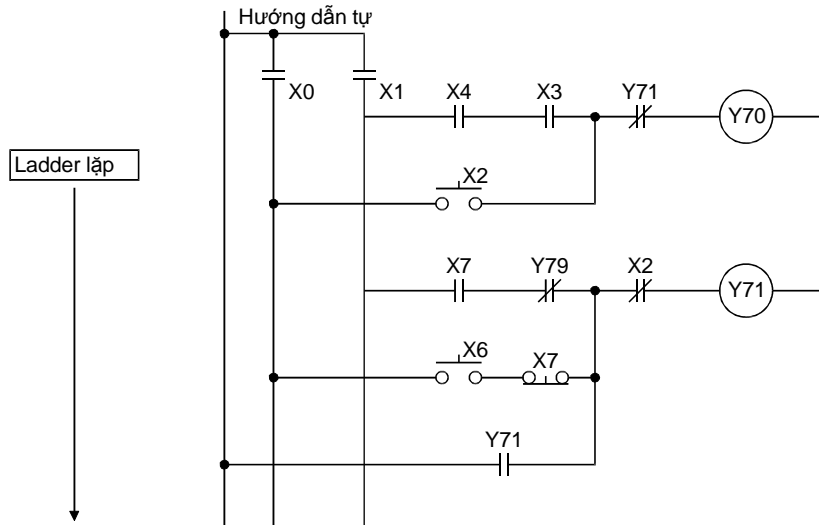
Bộ định thời thiết lập sao cho nó bắt đầu ở đầu ra khi đầu vào giữ được trên cho 0.2s.



M1 bật khi X0 giữ được trên cho 0.2s, hoặc dài hơn. Do đó, sử dụng M1 thay cho X0 khi tạo ra chương trình.

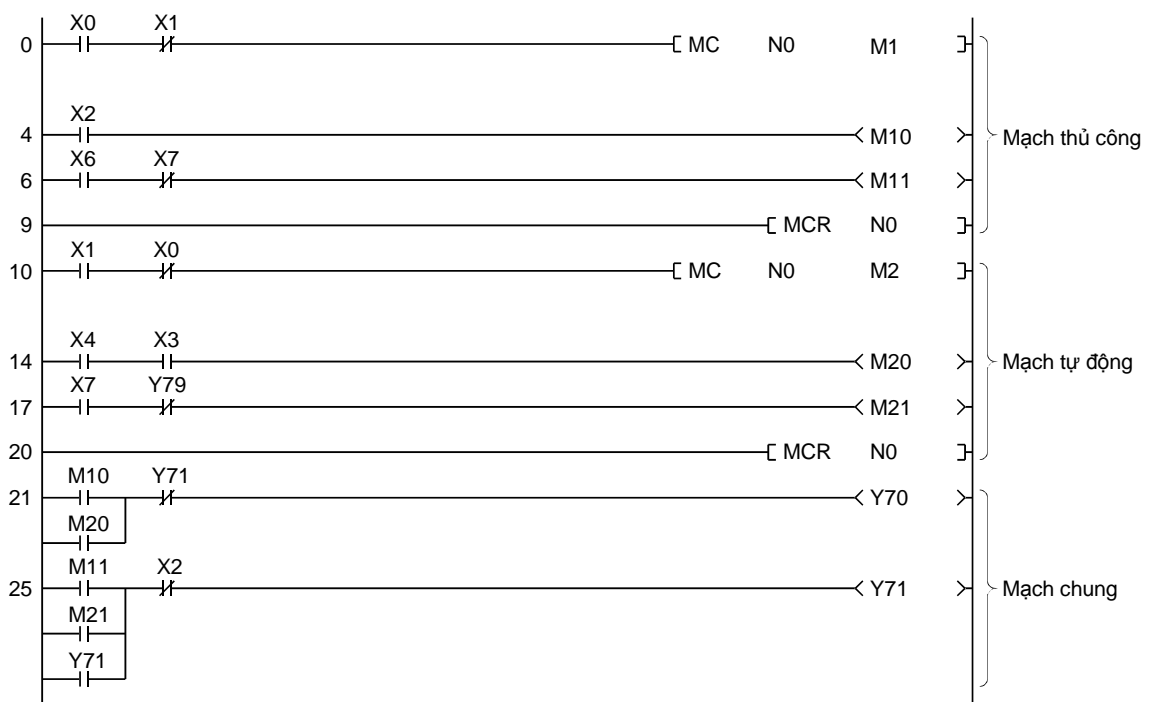
Phụ lục 4.8 Ladder với tuyến chung

Ladder sau không thể hoạt động như bình thường. Để ladder dễ điều khiển, thì sử dụng lệnh điều khiển chính ( MC, MCR ) trong chương trình.



Tên dự án	QA-1
Tên chương trình	MAIN

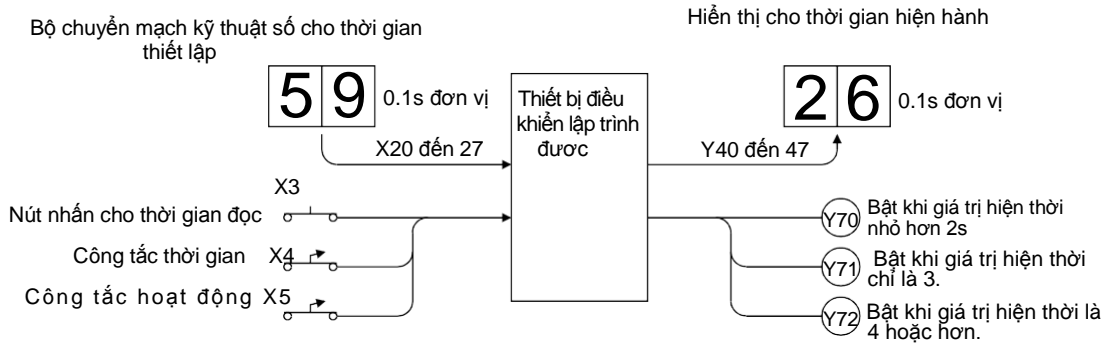
Chương trình tuần tự với lệnh điều khiển chính



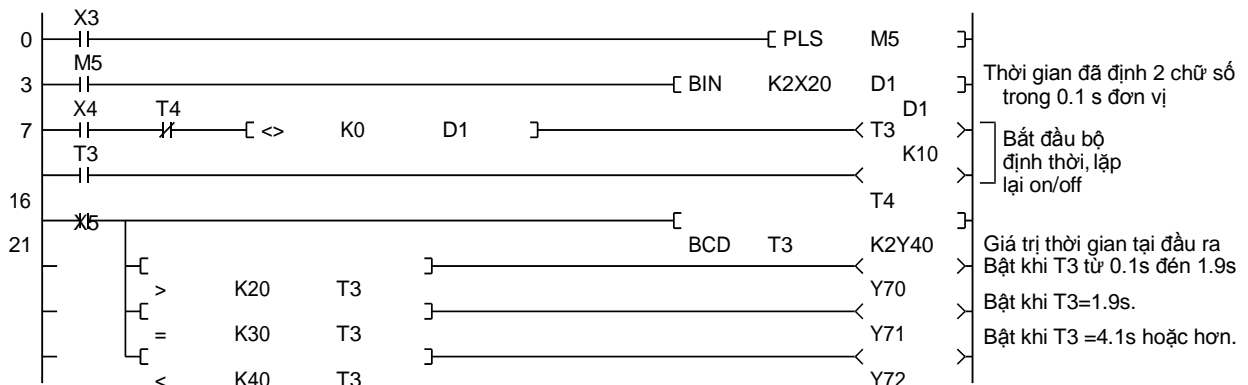
Chú thích ) Trong GX Works2, trạng thái bật / tắt của bộ điều khiển chính được hiển thị trong thẻ tiêu đề đi cùng màn hình thiết bị hiển thị.

Phụ lục 4.9 Chương trình điều khiển thời gian

Giá trị thời gian được thiết lập bởi hai chữ số của bộ chuyển mạch kỹ thuật số. Hiện tại khoảng thời gian đã qua được hiển thị trên Y40 đến Y47 trong khi đầu ra Y70 đến Y72 bật sau khi thời gian giới hạn đã trôi qua. Thao tác này cứ thế được lặp lại.



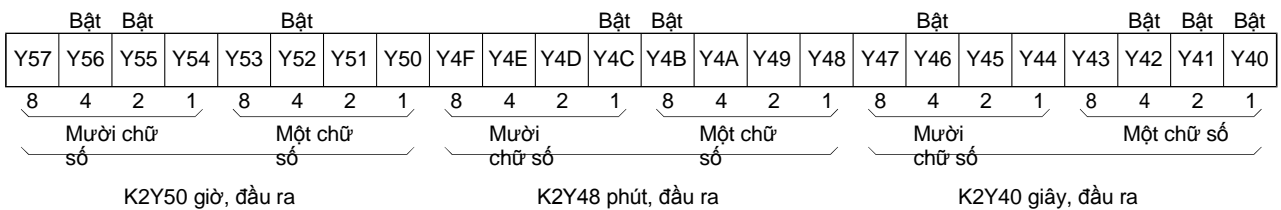
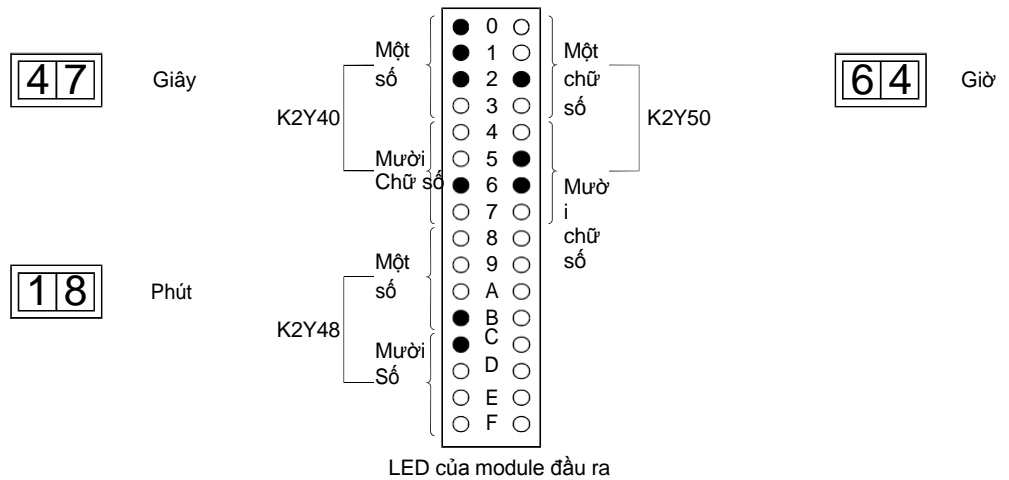
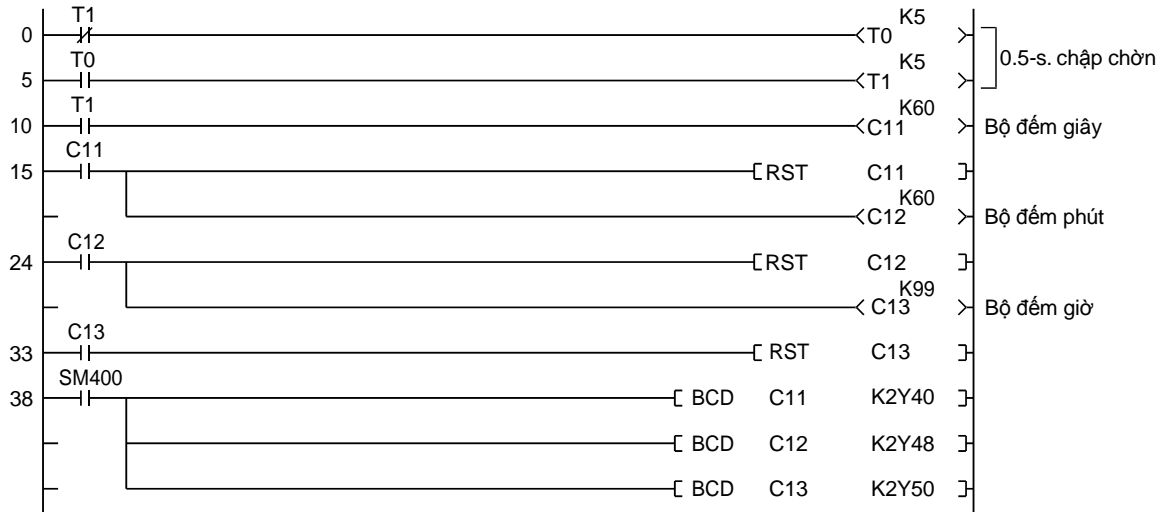
Tên dự án	QA-2
Tên chương trình	MAIN



Phụ lục 4.10 Đồng hồ ladder

Dữ liệu đồng hồ như là giờ, phút, và giây là đầu ra đến màn hình kỹ thuật số.

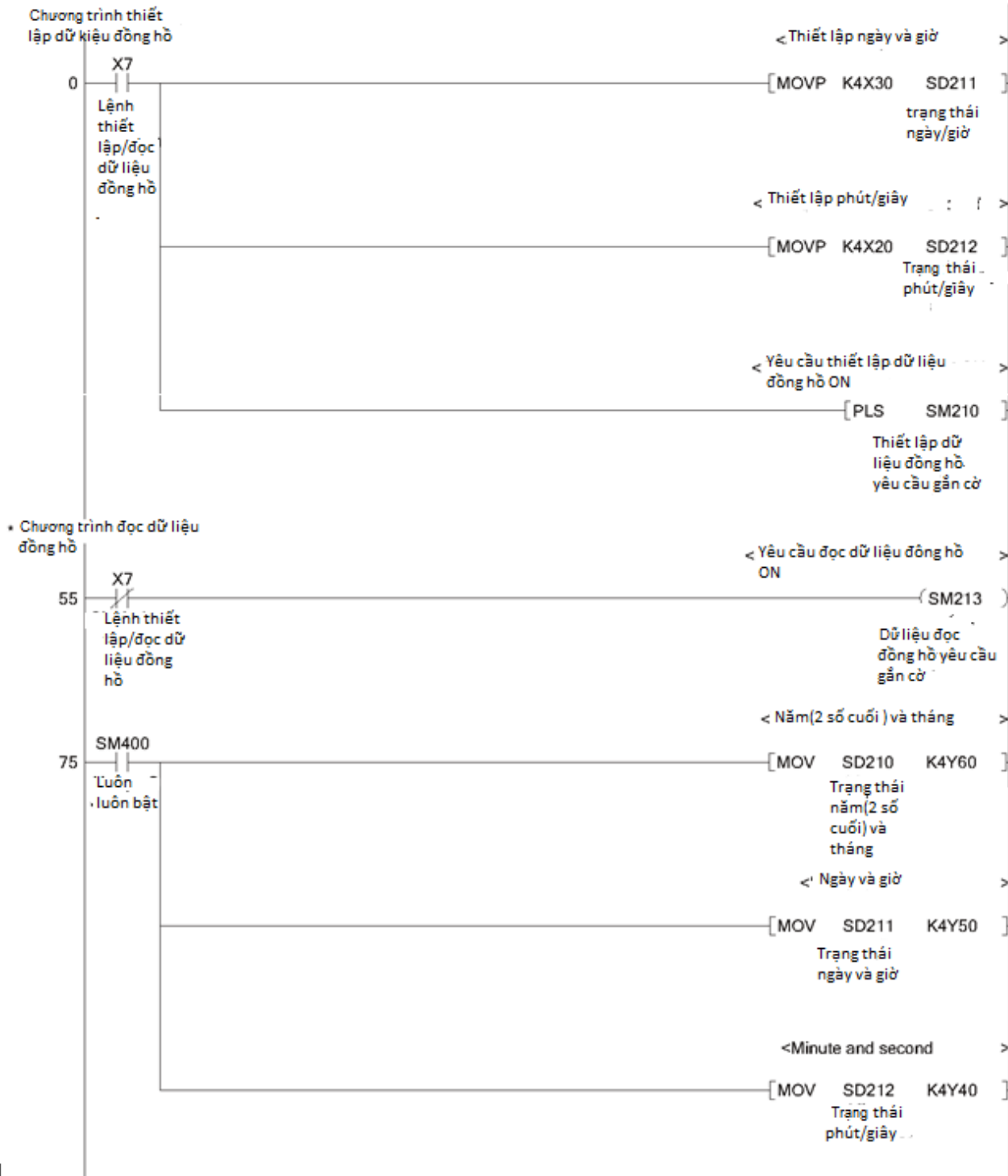
Tên dự án	QA-3
Tên chương trình	MAIN



Phụ lục 4.10.1 Chức năng clock (Bổ sung)

Ladder sau đây hiển thị thời gian thiết lập GX Works2 đến máy chứng minh Q.

Tên dự án	QEX13
-----------	-------



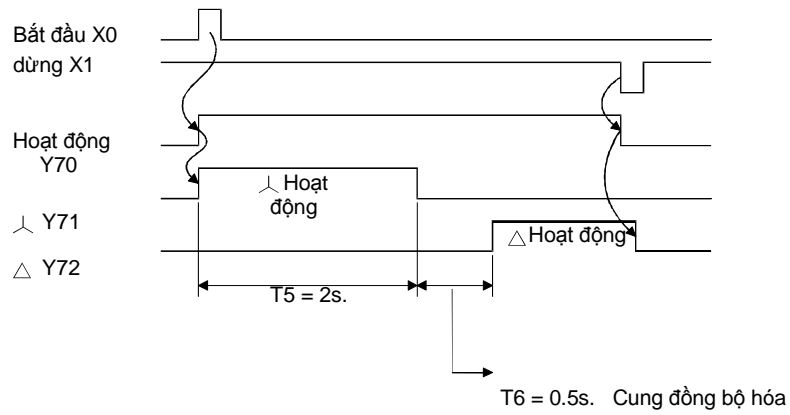


118	[=	H2000	SD213 ]		Hiện thị chhur nhật	(Y76)	)
				Năm(2 số cuối ) /ngày trong tuần	<   Hiện thị thứ hai		>
133	[=	H2001	SD213 ]			(Y75)	)
				Năm (2 số cuối)/ngày trong tuần			
					Hiện thị thứ ba		
148	[=	H2002	SD213 ]			(Y74)	)
				Năm (2 số cuối )/ngày trong tuần	< Hiện thị thứ tư		>
163	[=	H2003	SD213 ]			(Y73)	)
				Năm (2 số cuối )/ngày trong tuần	< Hiện thị thứ năm		>
179	[=	H2004	SD213 ]			(Y72)	)
				Năm (2 số cuối )/ngà y trong tuần	< Hiện thị thứ sáu		>
195	[=	H2005	SD213 ]			(Y71)	)
				Năm (2 số cuối )/ngày trong tuần	block		
					< Hiện thị thứ bảy		>
210	[=	H2006	SD213 ]			(Y70)	)
				Năm (2 số cuối )/n gày trong tuần			
226						[END	]

Phụ lục 4.11 Khởi động sao / Tam giác:

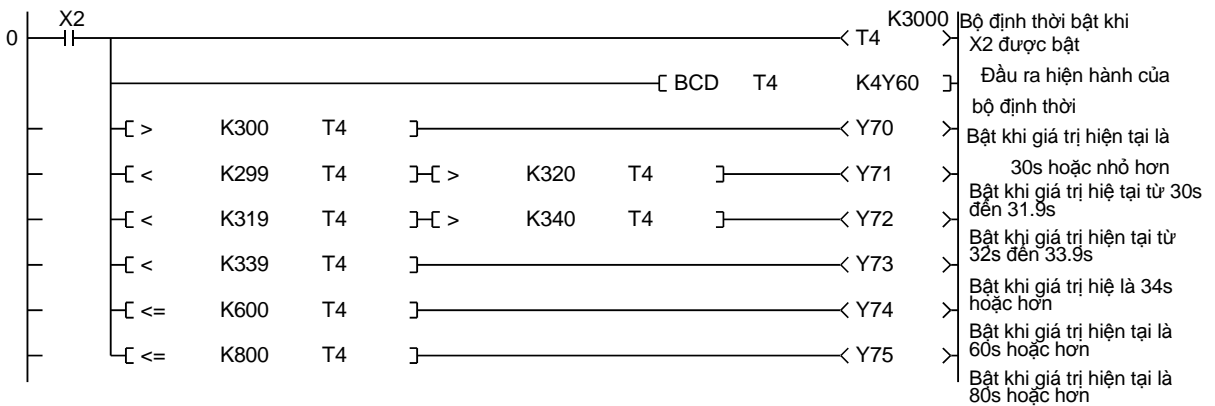
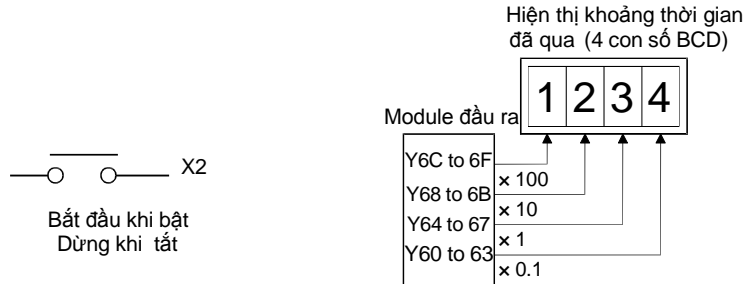
Bật công tắc khởi động hoạt động  $\sphericalangle$ . Sau hoạt động  $\sphericalangle$  đã trôi qua, chế độ hoạt động 6 được kích hoạt thông qua trạng thái cung đồng bộ hóa.

Tên dự án	QA-20
Tên chương trình	MAIN



Phụ lục 4.12 Hiển thị khoảng thời gian đã qua và đầu ra trước thời gian giới hạn

Các đầu ra Ladder sau đây, thời gian trôi qua trong bộ định thời trên màn hình LED, và cho thấy thiết lập thời gian đã được đạt đến. Hệ thống này cũng có thể được đưa vào trong các bộ đếm.

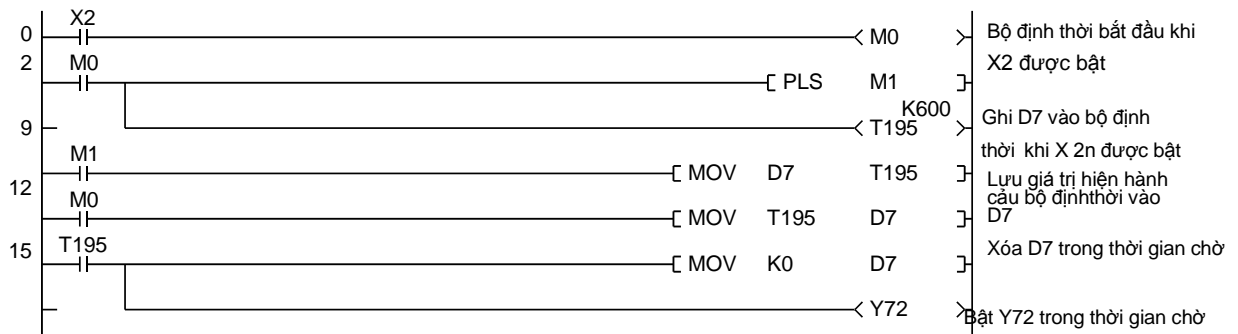


Phụ lục 4.13 Bộ định thời có nhớ

Đầu vào X2 chuyển đổi liên tục giữa bật và tắt. Số lần bật của X2 được tích lũy và Y72 bật theo giá trị tích lũy n.

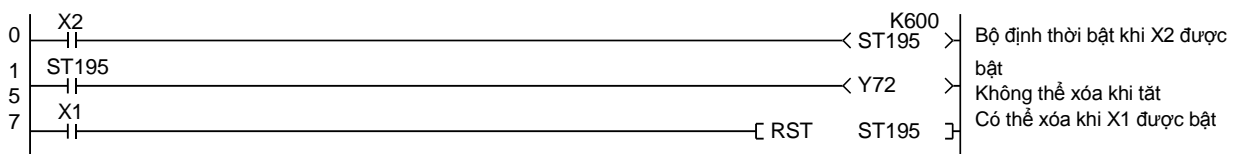
(1) Cho ladder tích lũy giá trị không có bộ định thời có nhớ

Tên dự án	QA-21
Tên chương trình	MAIN



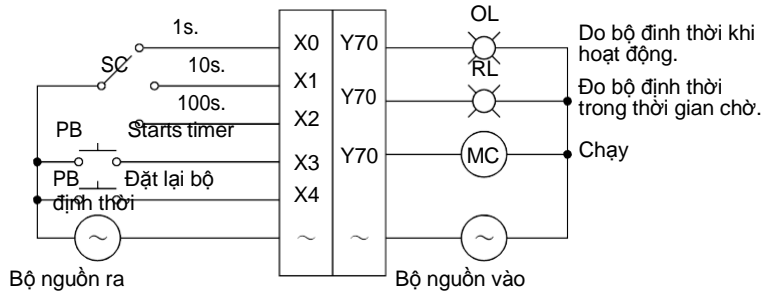
(2) Khi bộ định thời có nhớ được gán trong điều chỉnh thiết bị của bộ định thời lưu giữ tham biến PLC (ST) : 224 điểm ( ST0 đến ST223 )

Tên dự án	QA-8
Tên chương trình	MAIN

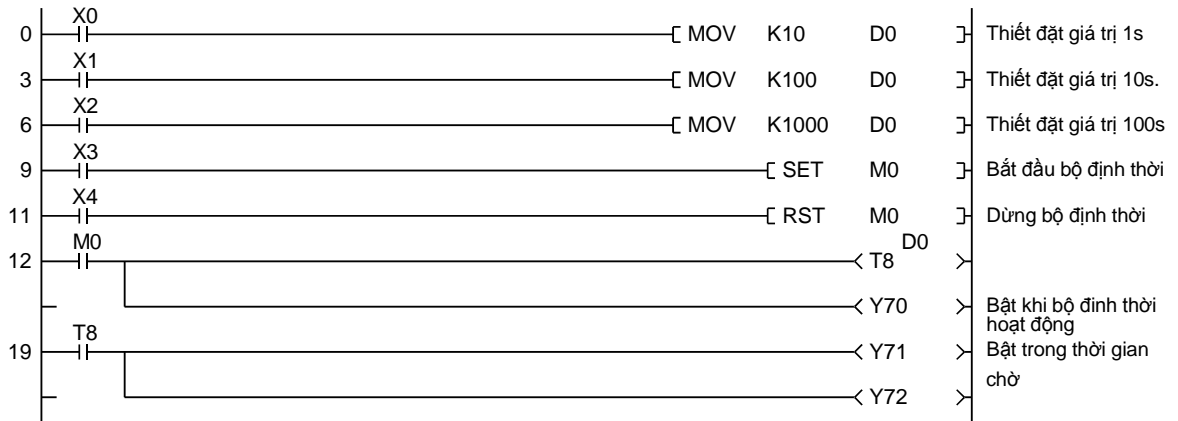


Phụ lục 4.14 Bộ định thời chuyển đổi thiết lập thời gian bên ngoài

- (1) Với công - tắc ngoài, giá trị để được thiết lập bởi một bộ định thời có thể được chọn lựa từ ba loại ; 1s. , 10s. , và 100s.  
 Thiết bị tính giờ được kích hoạt và thiết lập lại với công tắc ấn.



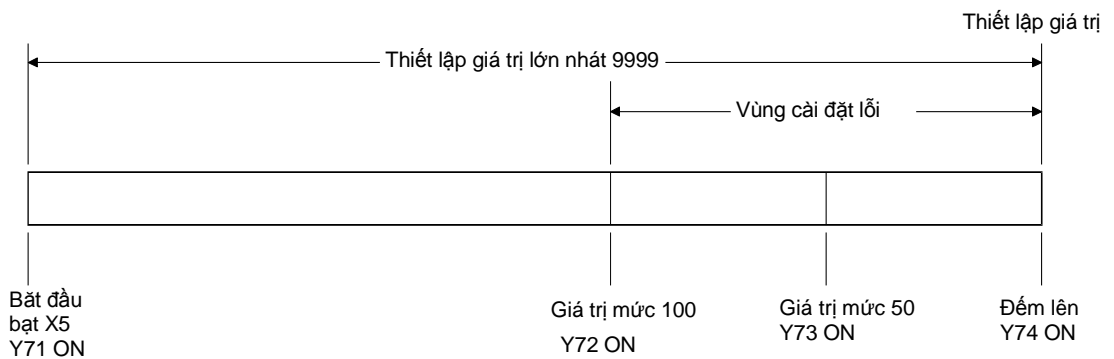
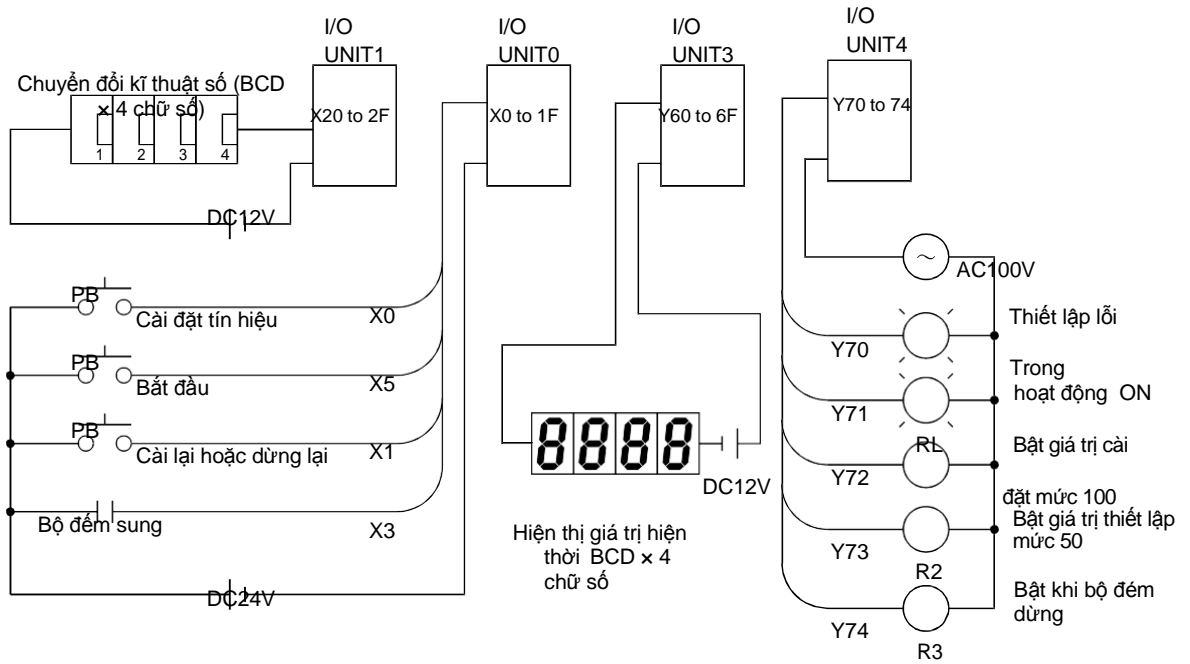
Tên dự án	QA-22
Tên chương trình	MAIN



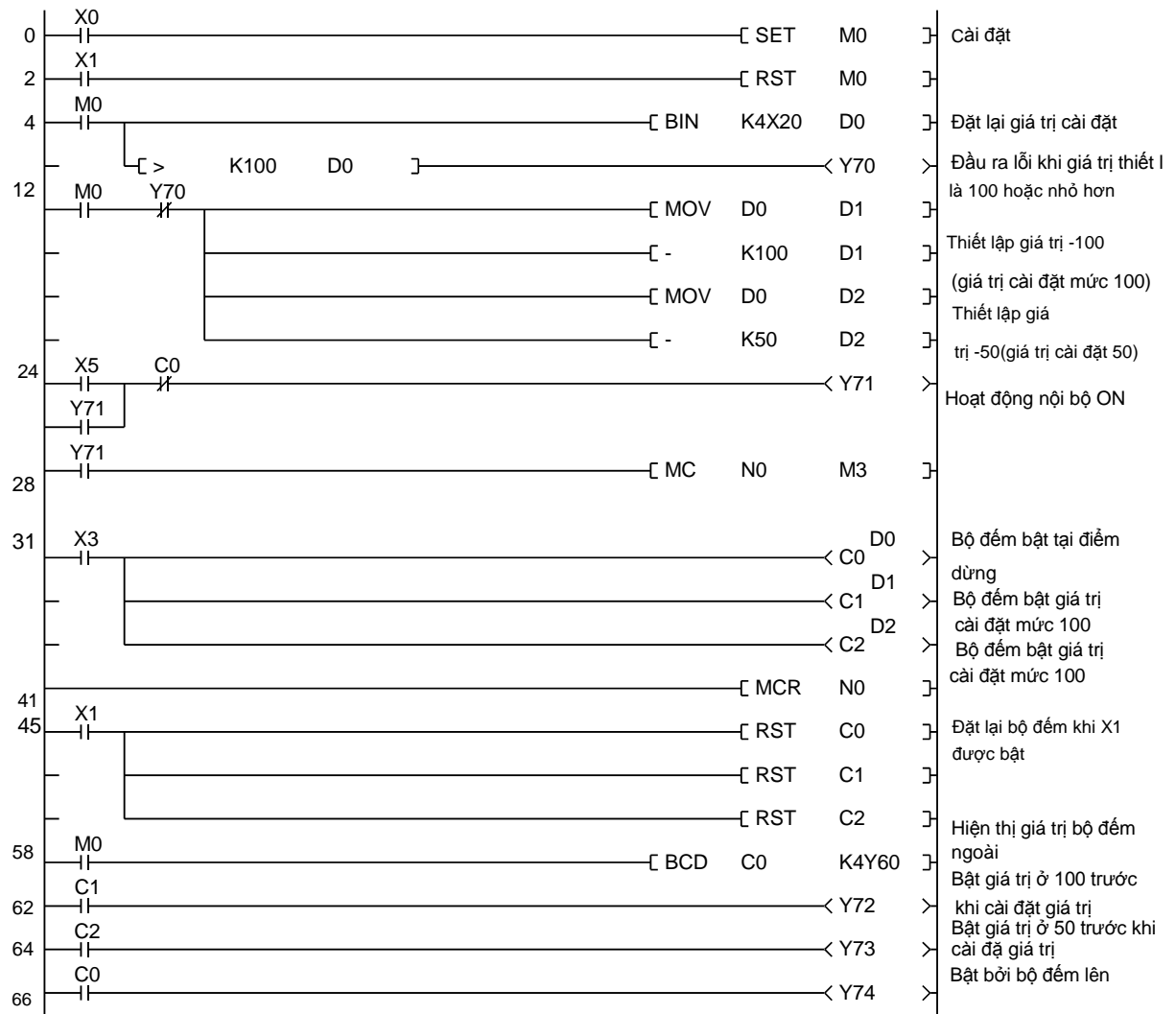
Phụ lục 4.15 Thiết lập bộ đếm ngoài

Với bộ chuyển mạch kỹ thuật số ngoài có 4 số, bộ đếm có thể thiết lập từ xa và giá trị hiện hành của chúng được hiển thị trong 4 chữ số. Ngoài ra mỗi lần đếm - lên, dữ liệu đầu ra của bộ định thời khi nó đạt đến giá trị mức 100 giá trị thiết lập và giá trị mức 50 giá trị thiết lập.

Lưu ý rằng thiết lập lỗi xảy ra nếu giá trị thiết lập của bộ đếm nhỏ hơn 100.



Tên dự án	QA-4
Tên chương trình	MAIN

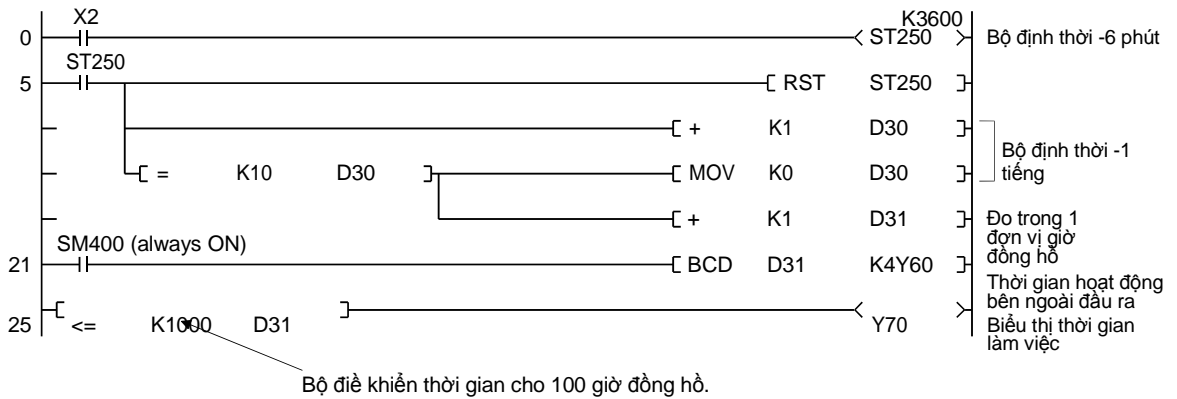


(Lưu ý ) Trong GX Works2, trạng thái bật/tắt của bộ điều khiển chính được hiện thị qua các thẻ tiêu đề trên màn hình.

Phụ lục 4.16 Đo thời gian hoạt động

Thiết lập thời gian hoạt động cho một bộ điều khiển đích là hữu ích cho việc phán đoán thời gian của bộ phận thay thế và bôi trơn. Các bộ đếm thời gian ST và thanh ghi D phải có nguồn điện dự phòng để có thể tiếp tục hoạt động khi mất điện. Với các nội dung của D31 (trong một giờ) được hiển thị ra bên ngoài, chương trình có thể hoạt động như một bộ đếm thời gian.

Tên dự án	QA-23
Tên chương trình	MAIN

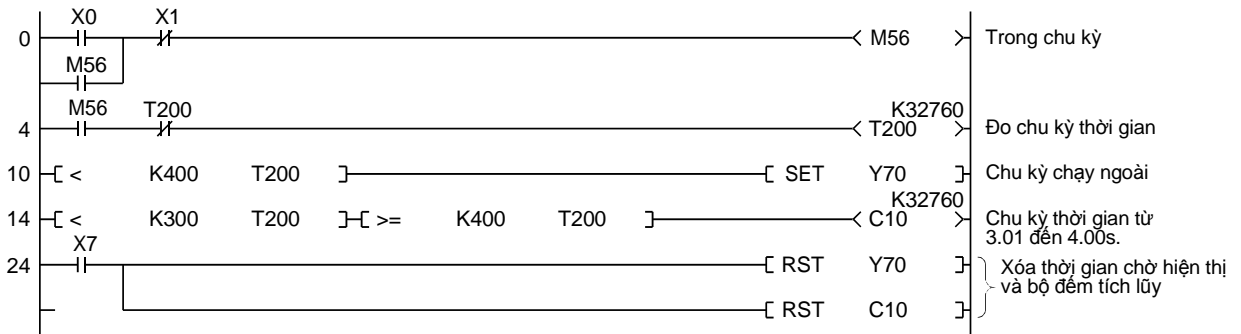


Phụ lục 4.17 Đo chu trình

Đo thời gian hoạt động của một bộ điều khiển đích (từ lúc bắt đầu đến kết thúc) cho phép hiển thị chu kỳ chờ và quản lý thời gian trễ.

Ladder sau với các lệnh <, >, và = được sử dụng để xác định trạng thái của và T200 chỉ ra chu kỳ chờ và đo khoảng thời gian trễ bằng các bộ đếm.

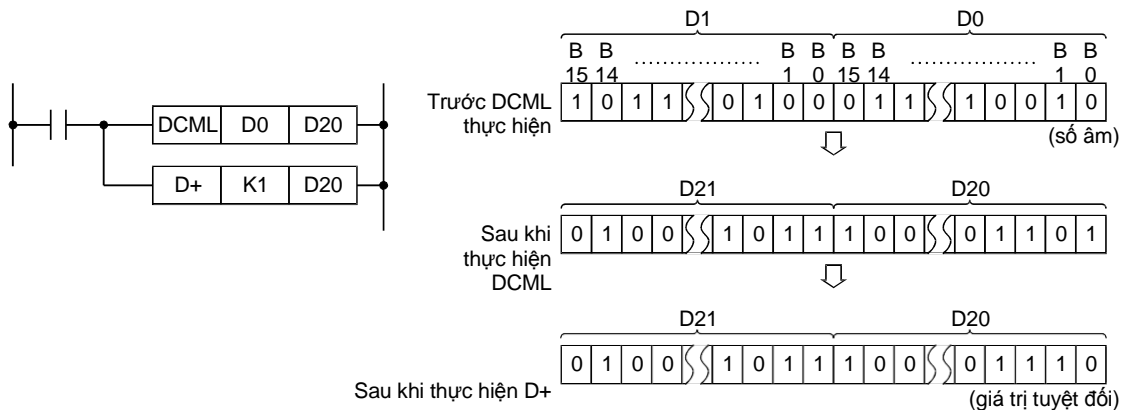
Tên dự án	QA-24
Tên chương trình	MAIN





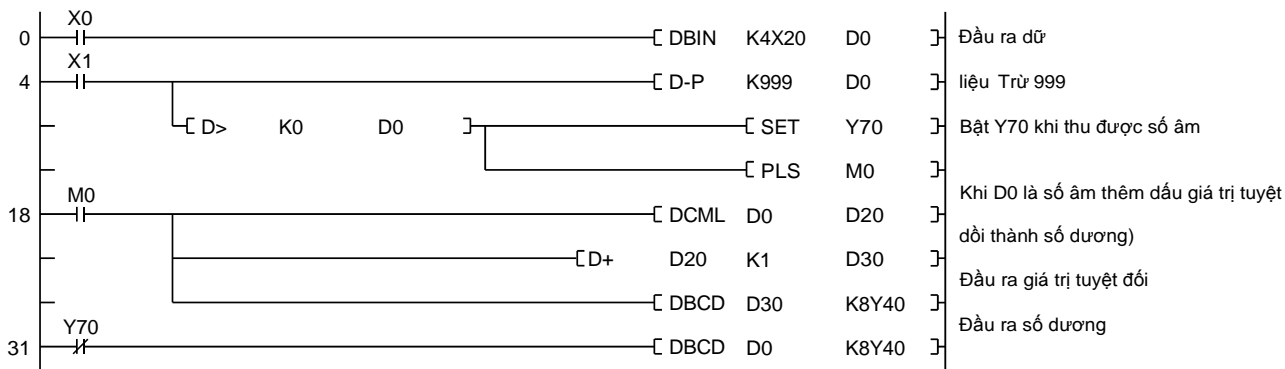
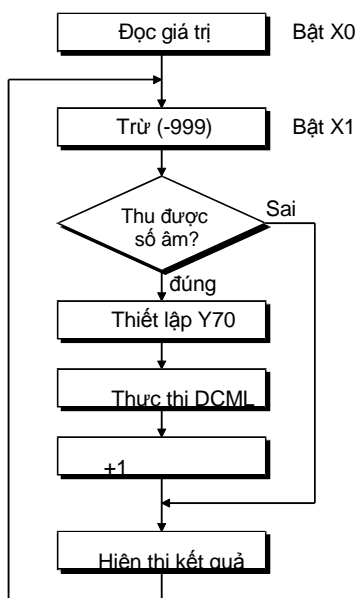
Phụ lục 4.18 Ví dụ ứng dụng của (D) C M L (P) Phản bù

Hình sau đây sẽ giải thích làm thế nào để có được giá trị tuyệt đối của giá trị âm -32.768 hoặc nhỏ hơn (-2147483648, 32 bit dữ liệu).



(Ví dụ)

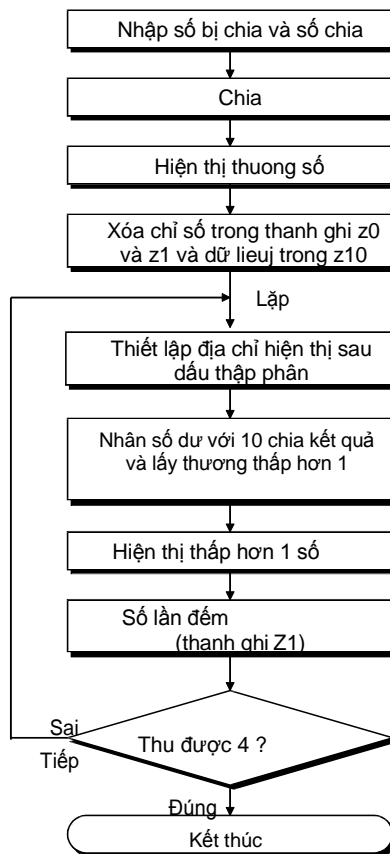
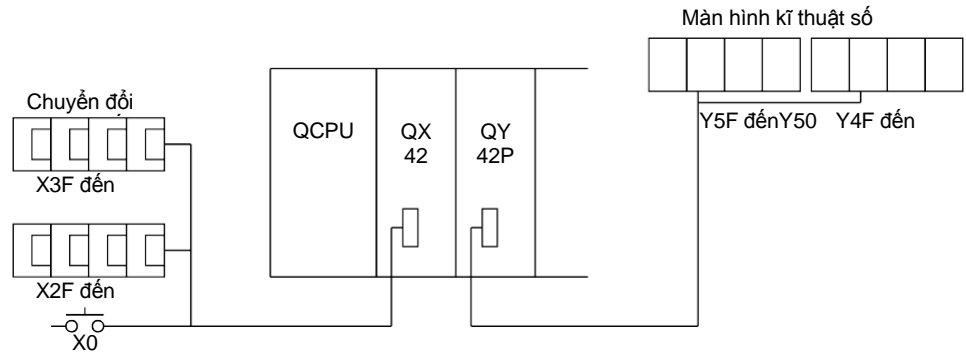
Mỗi lần X1 được bật, 999 trừ một giá trị và kết quả được hiển thị. Khi các giá trị kết quả là âm, đầu ra Y70 được bật, và các giá trị tuyệt đối của được hiển thị.



Phụ lục 4.19 Chương trình hiển thị chia giá trị của 4 chữ số BIN đến 4 hàng chữ số thập phân.

(1) Ví dụ 1

Chương trình sẽ hiển thị các kết quả bằng cách sử dụng số bị chia và số chia được xác định riêng trong hai thiết bị chuyển mạch kỹ thuật số 4 chữ số trên hai màn hình có 4-chữ số (một phần không thể tách rời và phần thập phân).

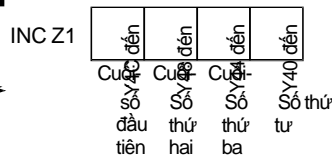


Số bị chia chuyển kĩ thuật số X30 đến X3F → D0

Số chia chuyển kĩ thuật số X20 đến X2F → D1

$(D0) / (D1) = (D2) \dots (D3)$   
Thương Số dư

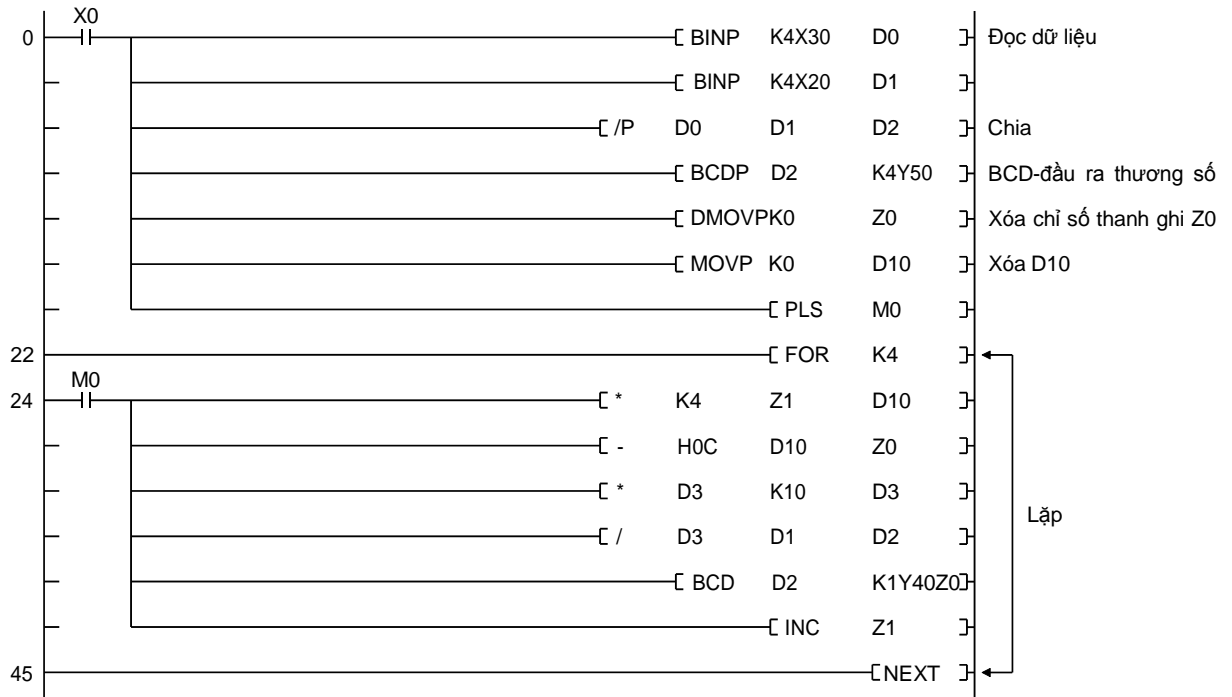
$4 \times (Z1) \rightarrow (D10) \quad HC-(D10) \rightarrow (Z0)$   
 1st time  $4 \times 0 \rightarrow 0 \quad HC-K0 \rightarrow HC$   
 2nd time  $4 \times 1 \rightarrow 4 \quad HC-K4 \rightarrow H8$   
 3rd time  $4 \times 2 \rightarrow 8 \quad HC-K8 \rightarrow H4$   
 4th time  $4 \times 3 \rightarrow 12 \quad HC-K12 \rightarrow H0$   
 $(D3) \times 10 \rightarrow (D3)$   
 $(D3) / (D1) = (D2) \dots (D3)$



Chương trình tuần tự của ví dụ 1

Câu lệnh For-Next được thực hiện để phân chia mỗi chữ số thập phân riêng và 4 chữ số thập phân được hiển thị Print K4Y40.

Tên dự án	QA-5
Tên chương trình	MAIN

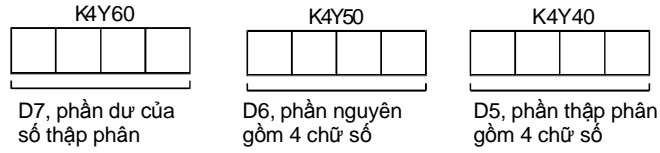


Thực hiện lệnh INC Z1 bổ sung thêm Z1.

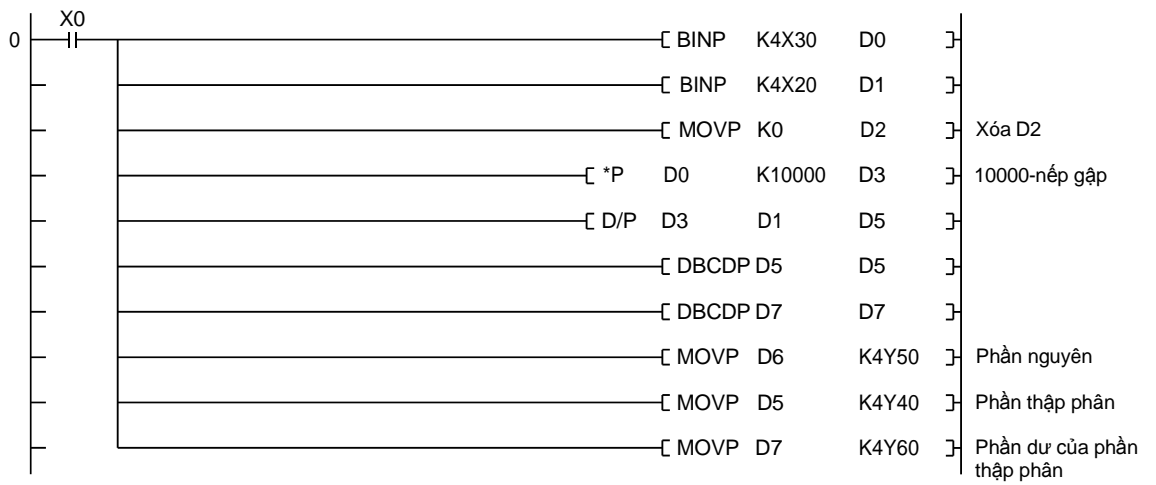
(2) Ví dụ 2

Trong ví dụ 2, D0 được chia D1 lấy D5 trong 4 chữ số thập phân.

Các số bị chia D0 được nhân với 10000. Kết quả của phép chia sử dụng được chuyển đổi thành một giá trị BCD và đầu ra với màn hình kỹ thuật số ngoại vi.



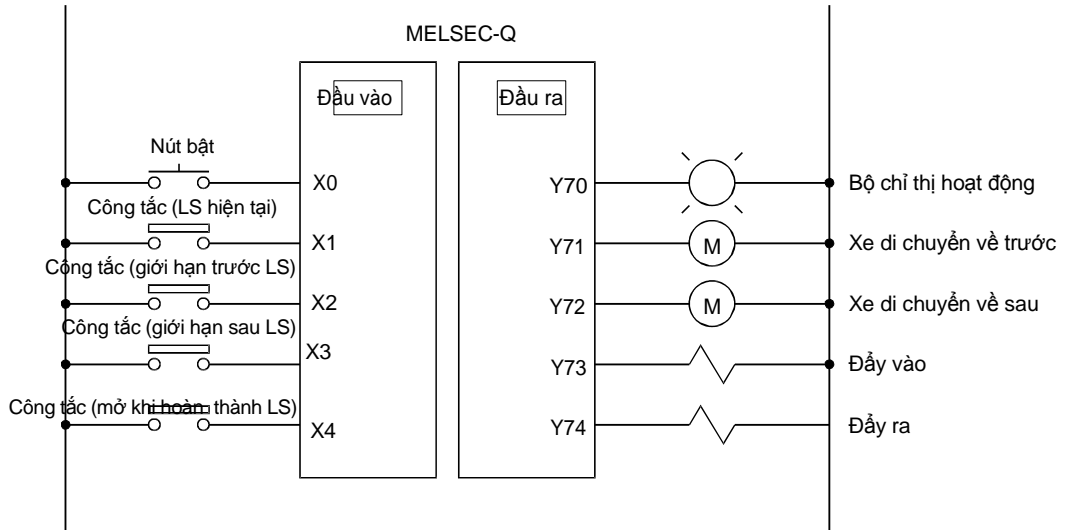
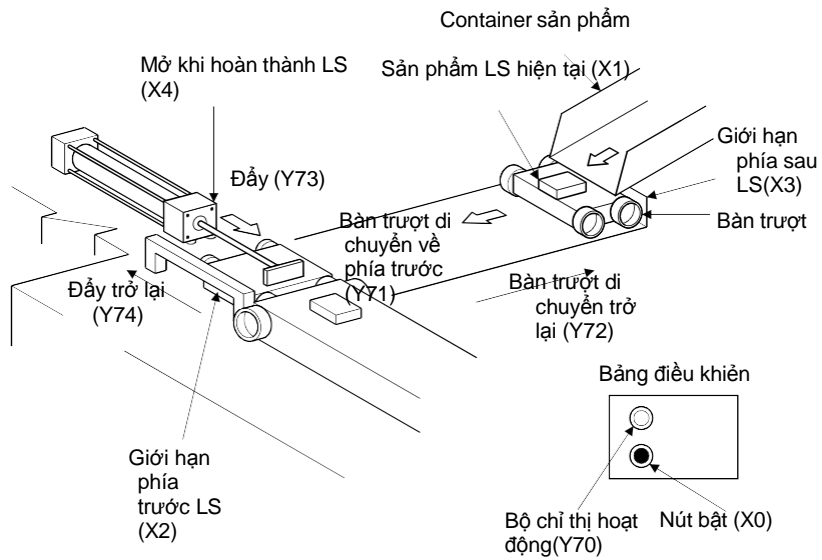
Tên dự án	QA-6
Tên chương	MAIN



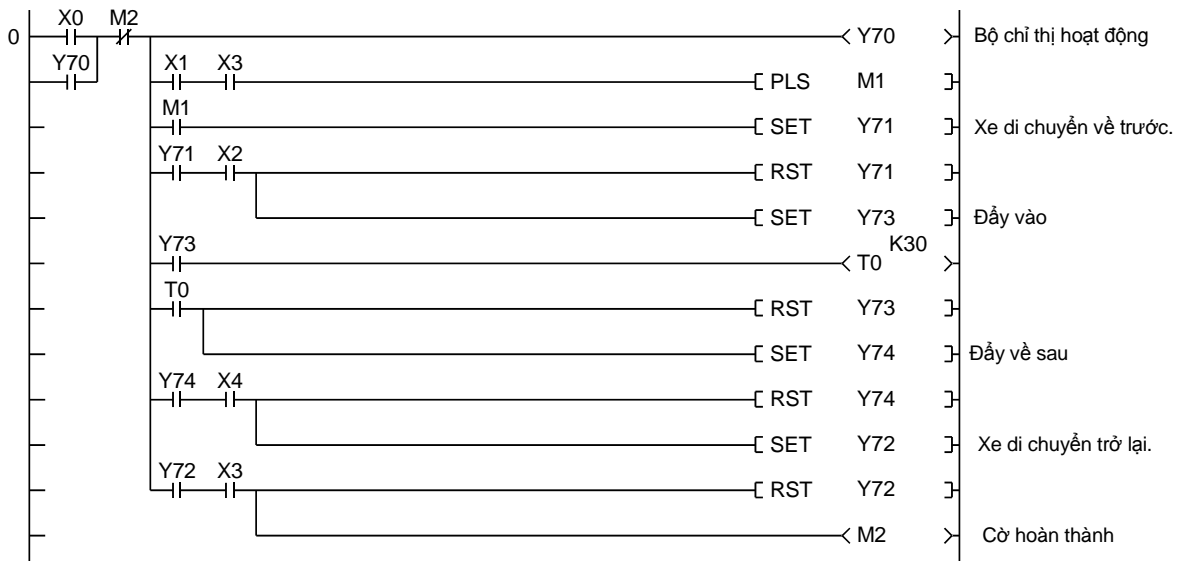
Phụ lục 4.20 Kiểm soát điều khiển tuyến

Sau đây là một ví dụ về quá trình điều khiển tuần tự sử dụng một bàn trượt để chở sản phẩm (nguyên liệu).

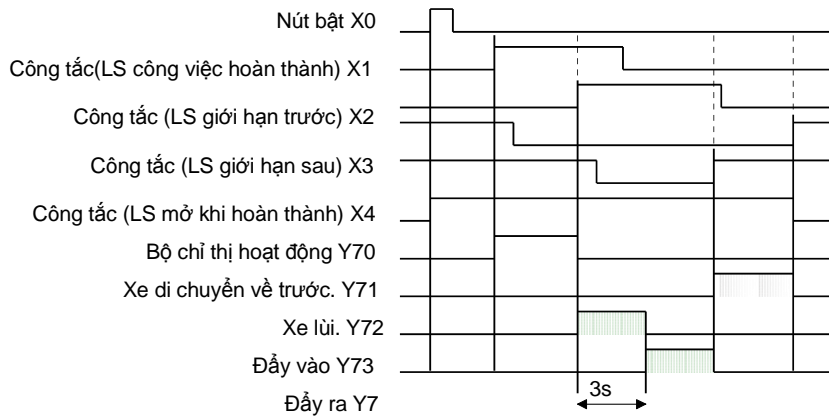
Chuỗi các thao tác thực hiện trong một chu kỳ như sau; sản phẩm được đặt lên băng trượt, băng trượt tiến lên, băng trượt dừng lại ở giới hạn chuyển tiếp, cánh tay đẩy công việc đến phía băng chuyển khác, và băng trượt quay lại đến giới hạn về phía sau.



Tên dự án	QA-10
Tên chương trình	MAIN

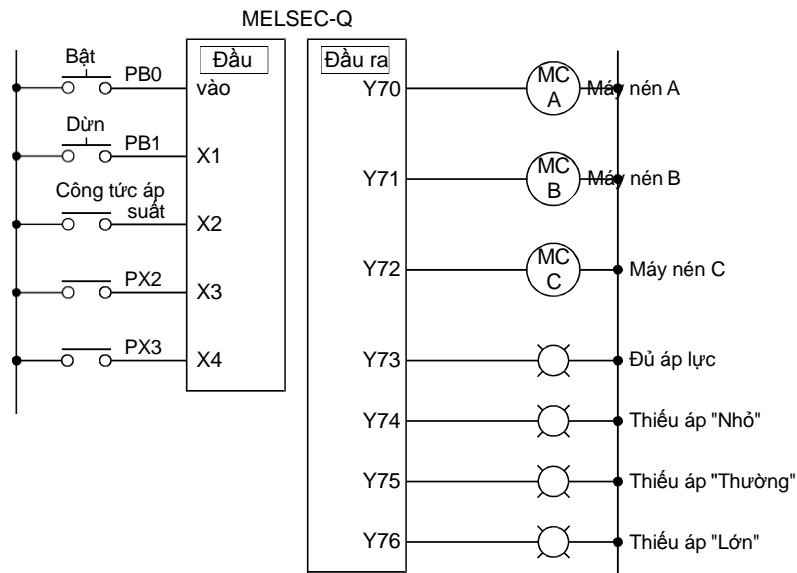
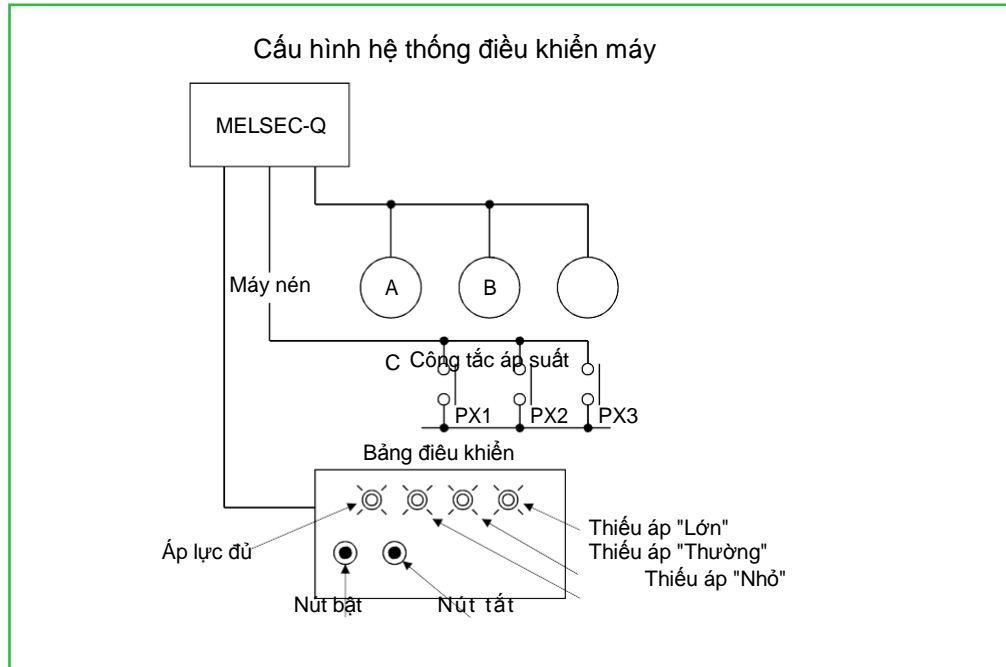


Biểu đồ thời gian



Phụ lục 4.21 Nén chương trình tuần tự sử dụng bộ đếm vòng

Hệ thống này điều khiển và quản lý áp lực với ba máy nén. Số lượng máy nén hoạt động cùng lúc phụ thuộc vào mức độ thiếu hụt áp. Để cân bằng số lượng của mỗi máy nén khí, máy nén được kích hoạt theo thứ tự đã thiết lập.



**Giải thích hoạt động**

(1) Các công tắc áp suất (X2, X3, và X4) ban đầu ở trạng thái tắt. Trong trạng thái này, bật công tắc khởi động (X0) kích hoạt ba máy nén hoạt động cùng với nhau, và khi áp lực thu được đã đủ (X2, X3, và X4 bật), ba máy nén dừng lại. Đây là hoạt động cơ bản của hệ thống này.

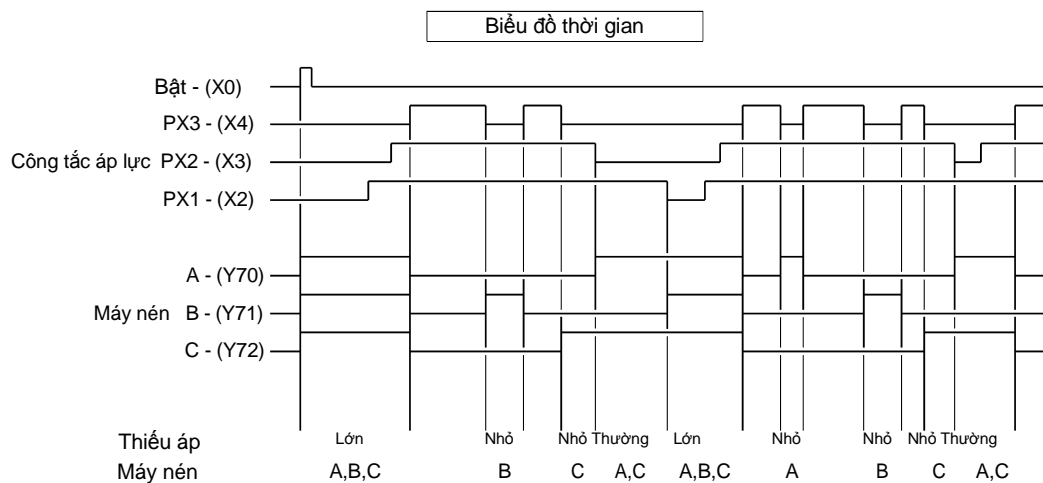
Nếu tất cả các máy nén dừng lại với đủ áp lực hoặc thiếu áp lực "nhỏ" được phát hiện (X4 tắt), một máy nén sẽ được kích hoạt để cung cấp áp lực cho đến khi đủ áp lực thu được.

Trong phản ứng với tình trạng thiếu áp lực, các máy nén kích hoạt tại thời điểm này được bật theo thứ tự từ A đến. Lưu ý rằng việc chuyển (X1) sang dừng lại luôn sẵn sàng cho việc dừng máy nén bất cứ lúc nào.

(2) Nếu một máy nén không cung cấp đủ áp lực và mức độ thiếu hụt áp sẽ tăng lên đến "Thường" (X3 tắt), máy nén thứ hai sẽ được kích hoạt để hỗ trợ các máy nén đầu tiên. Máy nén thứ hai này sẽ là máy nén C nếu máy nén A đã đi vào hoạt động, A nếu B đã đi vào hoạt động, và B nếu C đã đi vào hoạt động.

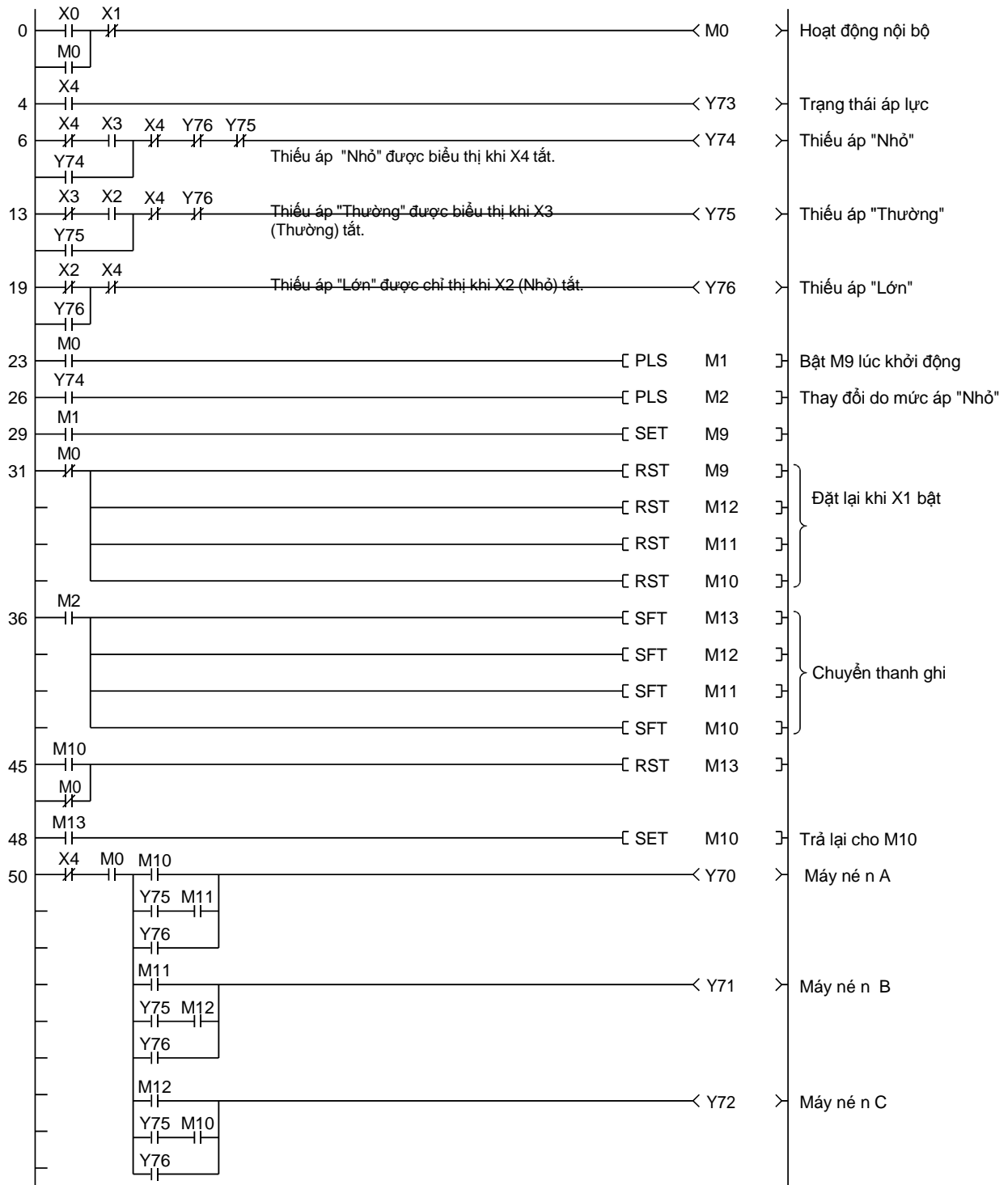
(3) Nếu cả hai máy nén không cung cấp đủ áp lực thì mức độ thiếu hụt áp sẽ tăng lên để "Lớn" (X2 tắt), máy nén cuối cùng cũng sẽ được kích hoạt. Khi chỉ có một máy nén hoạt động và áp lực thiếu hụt ngay lập tức sẽ tăng từ "Nhỏ" đến "Lớn", thì hai máy nén còn lại sẽ được kích hoạt cùng lúc.

(4) Khi hai hoặc ba máy nén hoạt động, chúng sẽ tiếp tục hoạt động cùng nhau cho đến khi đủ áp lực thu được. Sau đó, chúng sẽ dừng lại (X4 lợt về).



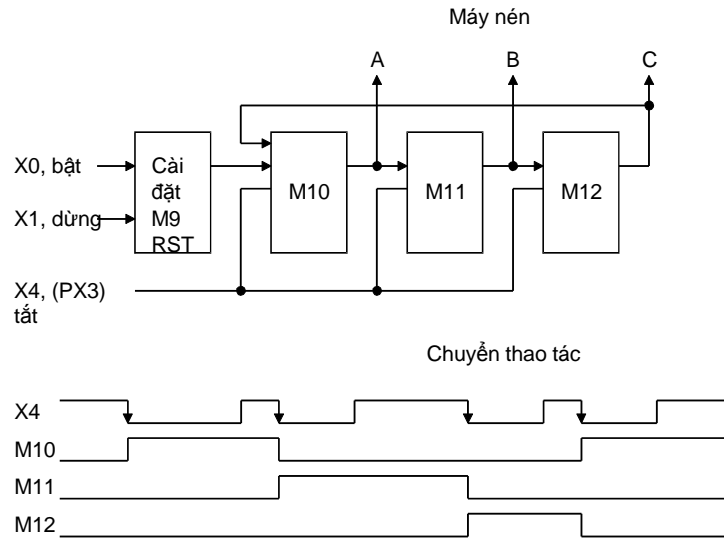


Tên dự án	QA-11
Tên chương trình	MAIN



Sau thao tác cơ bản, một bộ nén được kích hoạt trong phản ứng khi thiếu hụt áp lực phát hiện. Để sử dụng ba bộ nén đều nhau, chúng được kích hoạt theo thiết lập thứ tự. Điều khiển này được kích hoạt bằng 3 - giai đoạn cung bộ đếm ( thanh ghi cung dịch chuyển) M10 đến M12.

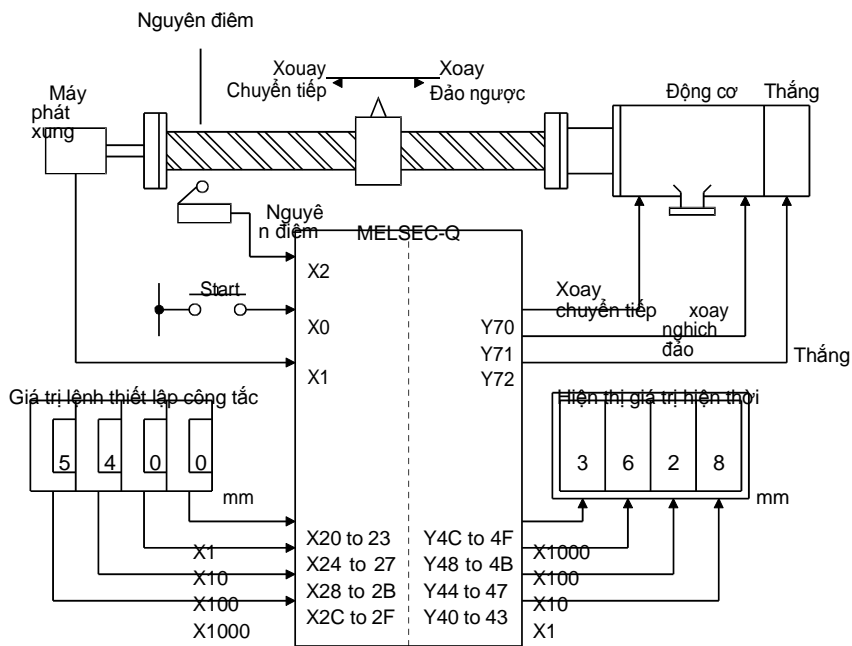
Tín hiệu chuyển được tạo ra khi thiếu áp lực được tìm ra ( X04 các khóa chuyển từ bật sang tắt).



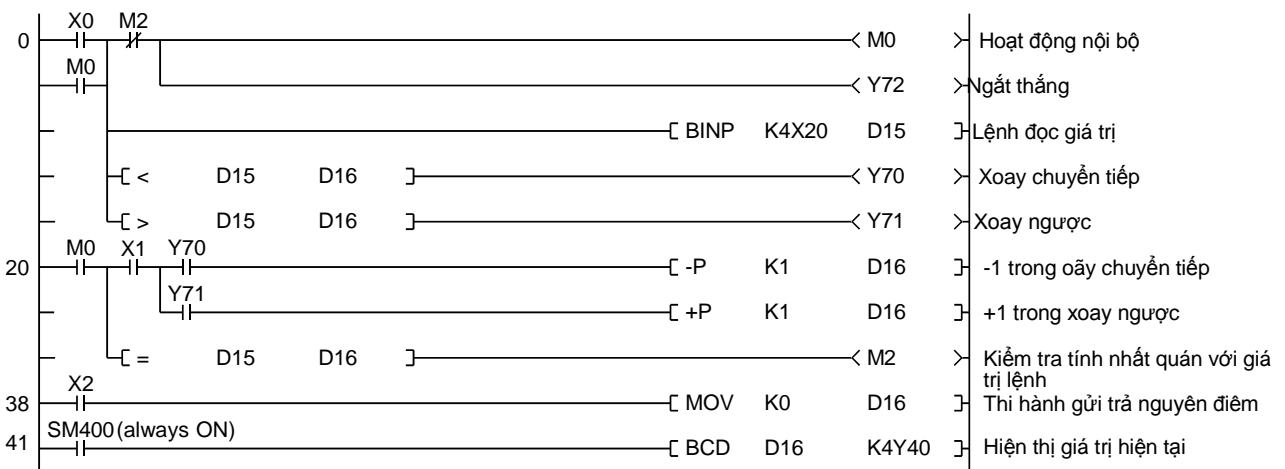
Phụ lục 4.22 Ví dụ ứng dụng điều khiển vị trí

Sau đây là ví dụ của hệ thống định vị với máy phát xung đầu ra xung nhịp mỗi động cơ, thẳng, và đơn vị của khoảng cách.

Trong hệ thống này, giá trị lệnh là thiết lập với bộ chuyển mạch kỹ thuật số, và giá trị lệnh thiết lập này được so sánh với giá trị hiện hành vào lúc khởi động để xác định hướng chỉ định, chuyển tiếp hoặc đảo ngược, xoay động cơ. Giá trị hiện hành ở thanh ghi D16 trừ 1 trong hướng chuyển tiếp, và tăng lên 1 trong hướng đảo ngược. Định vị hoàn tất khi giá trị lệnh trùng khớp giá trị hiện hành. Giá trị hiện hành được chuyển thành giá trị BCD sao cho vị trí hiện hành được biểu diễn trong 4 - chữ số các số thập phân.

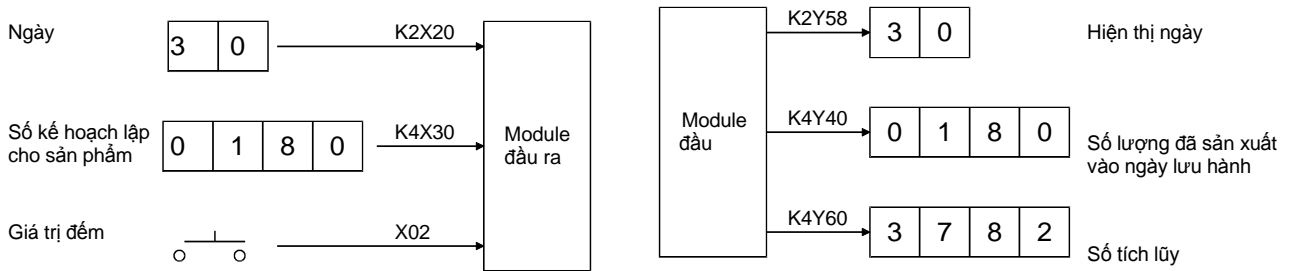


Tên dự án	QA-26
Tên chương trình	MAIN



Phụ lục 4.23 Ví dụ ứng dụng sử dụng chỉ số Z

- (1) Số sản phẩm được sản xuất được đếm mỗi ngày trong một tháng, và số kết quả được lưu trữ đến các thanh ghi tương ứng của ngày ( D1 đến D31 ).
- (2) Số lập kế hoạch để sản xuất sản phẩm là đầu vào của bộ chuyển mạch kỹ thuật số ngoài. Sản xuất ngừng lại khi số này được hoàn thành.
- (3) Ngày cũng được xác định với bộ chuyển mạch kỹ thuật số ngoài.
- (4) Số tích lũy sản phẩm được sản xuất trong tháng này cũng như số sản phẩm được sản xuất vào ngày lưu hành được hiển thị ngoài.



Số sản phẩm được sản xuất vào ngày lưu hành được đếm bằng C5. Số sản phẩm tích lũy được sản xuất được đếm bằng C6.  
 Ngày được nhập vào thanh ghi chỉ số Z để gián tiếp xác định dữ liệu thanh ghi phù hợp với ngày sử dụng D0Z0.  
 Khi Z0 bằng 30, D0Z0 trở thành 0 + 30, xác định D30.

[Thiết bị / Vùng đệm Bộ nhớ Bó Thiết bị hiển thị màn hình]

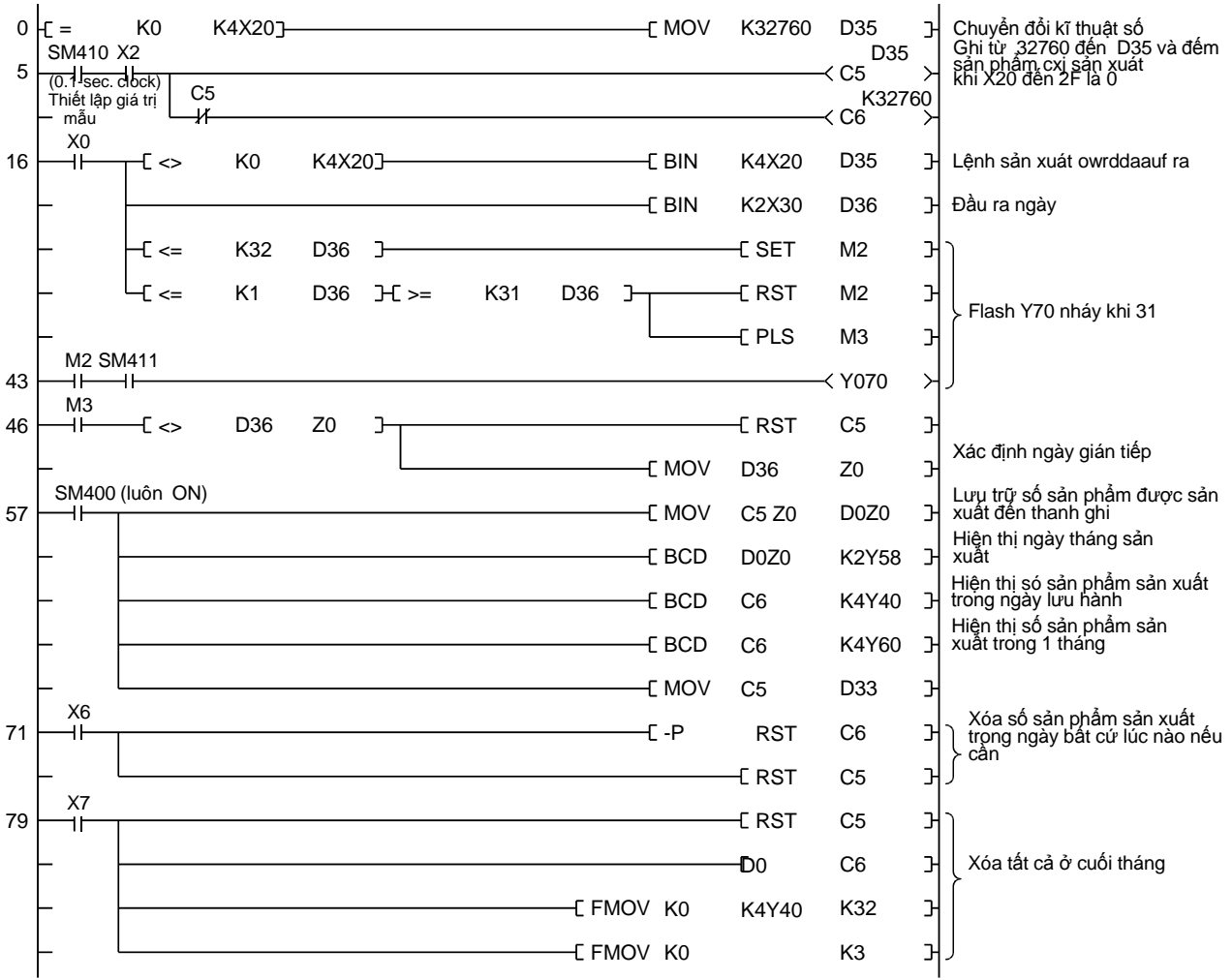
Device	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	159
D2	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	145
D3	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	168
D4	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	144
⋮																	
D29	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	213
D30	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	180
D31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D33	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1	0	3782
D34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D35	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	180
D36	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	30
D37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Annotations on the right side of the table:

- Arrow pointing to rows D1-D31: Lưu số kết quả mỗi ngày từ 1 đến 31 ở D1 đến D31.
- Arrow pointing to row D33: Số tích lũy
- Arrow pointing to row D35: Số kế hoạch của sản
- Arrow pointing to row D36: Ngày

Kết quả sản xuất của mỗi ngày từ 1 đến 31 được lưu trữ trong D1 đến D31, có sẵn trong dữ liệu sản xuất.

Tên dự án	QA-7
Tên chương trình	MAIN



FMOV	K0	D0	K32
------	----	----	-----

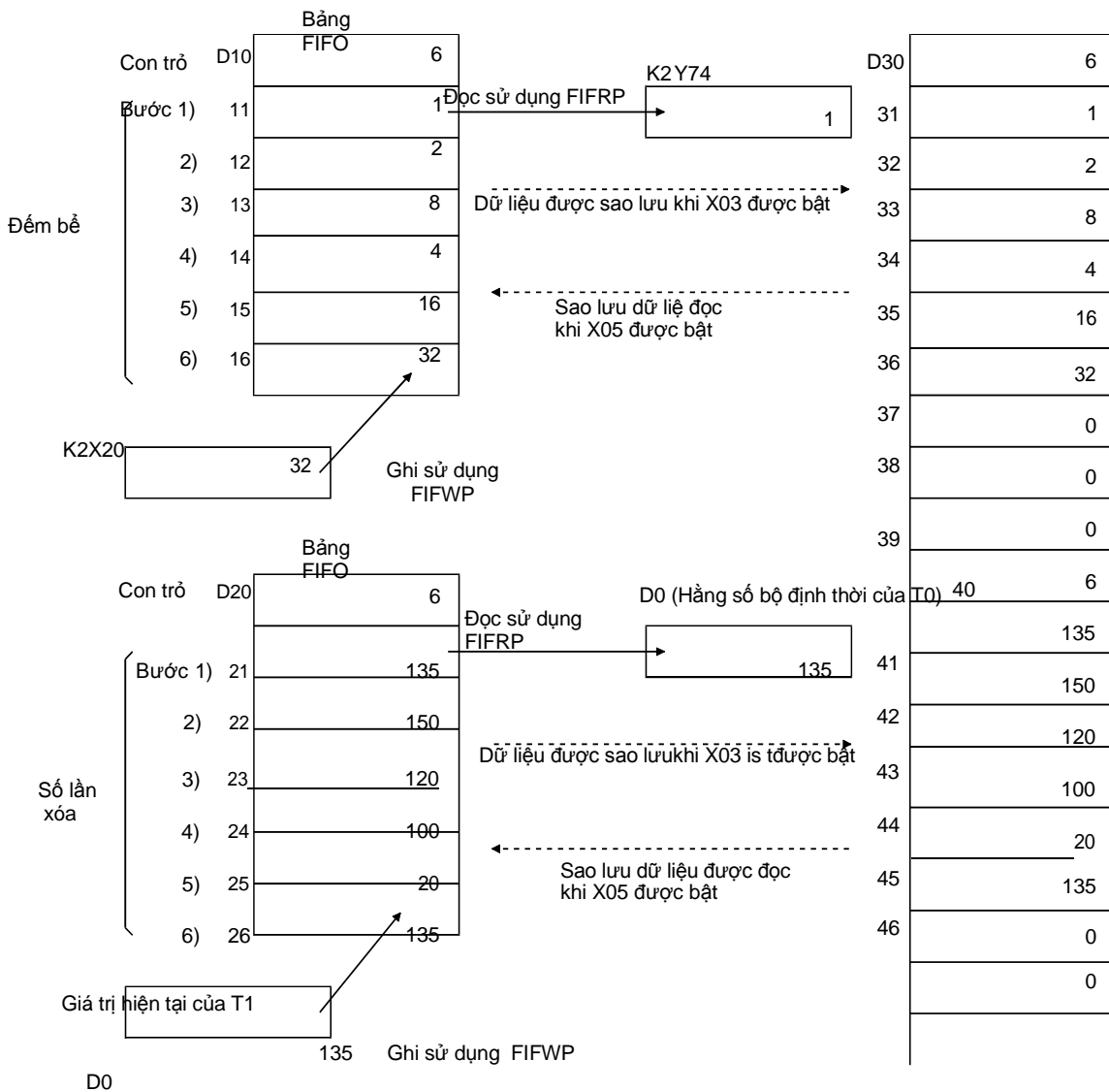
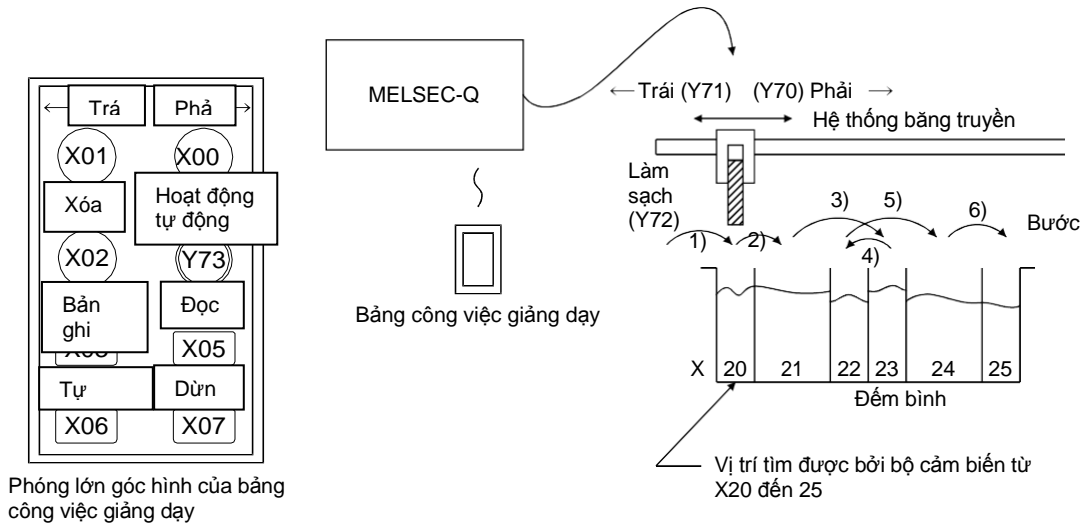
 Chuyển giao dữ liệu cùng 1 lúc từ 0 đến D0 đến D31.

FMOV	K0	K4Y40	K3
------	----	-------	----

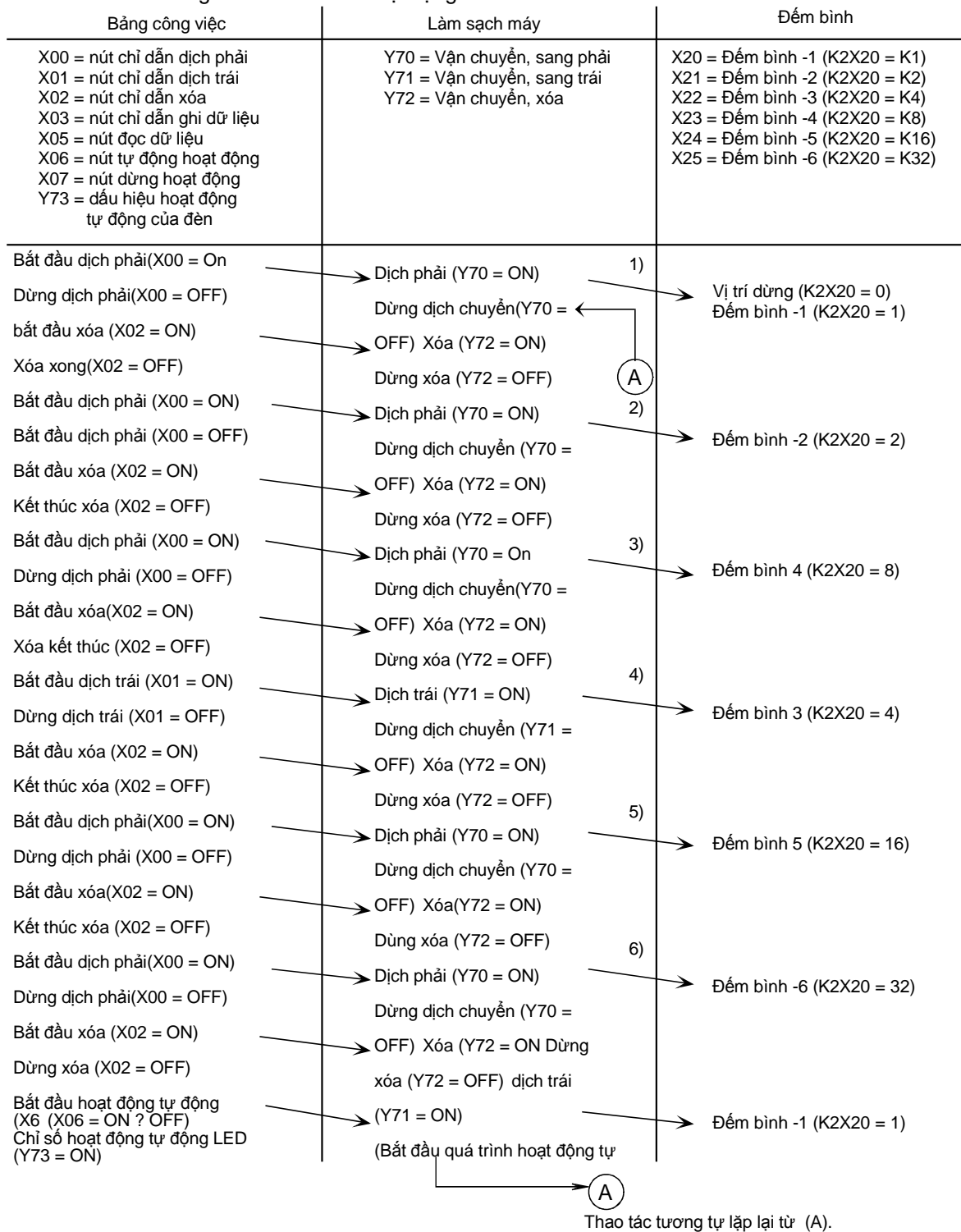
 Chuyển giao dữ liệu cùng 1 lúc 0 to K4Y40, K4Y50, và K4Y60.

Phụ lục 4.24 Ví dụ ứng dụng lệnh FIFO

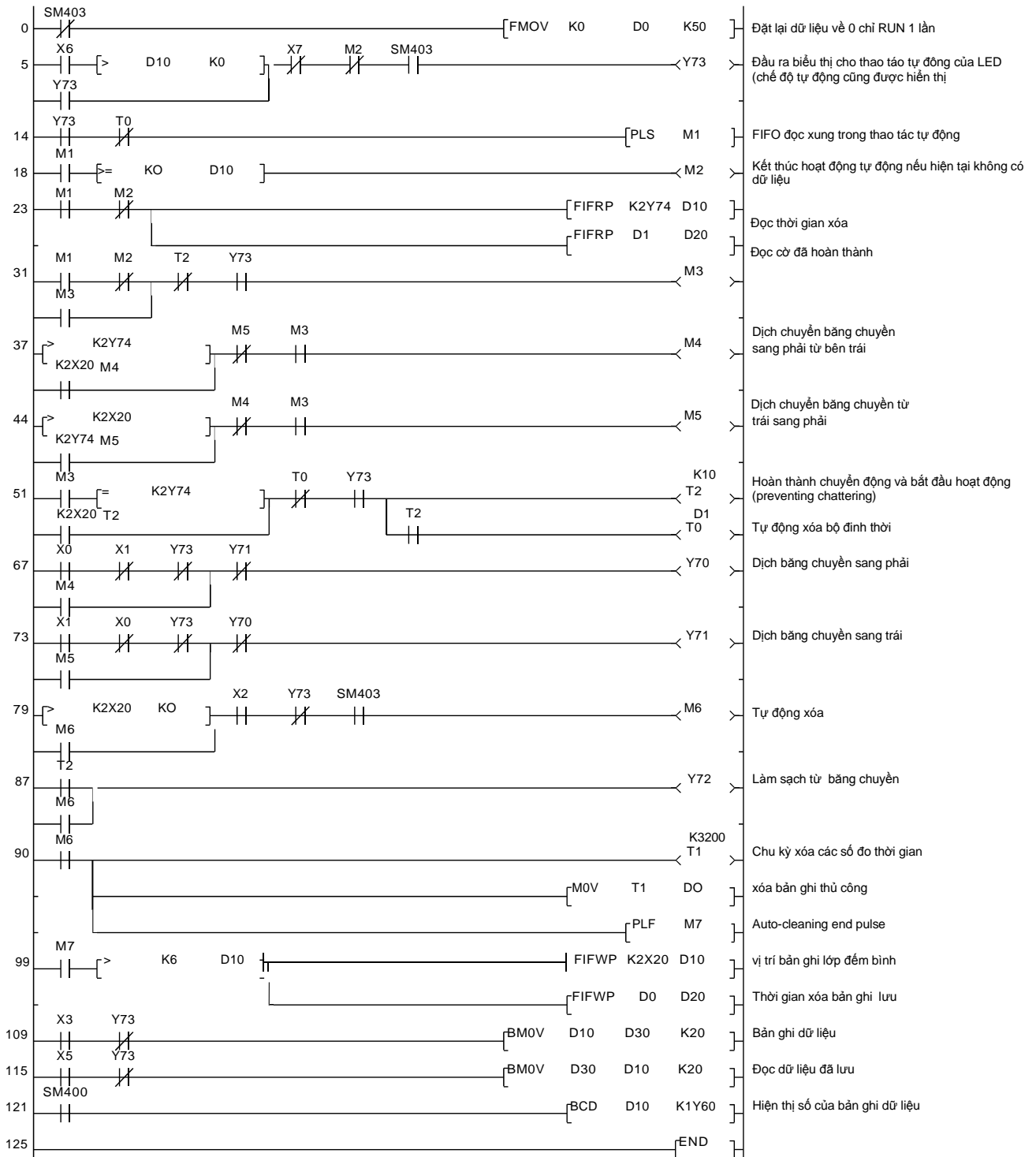
Hướng dẫn các gói công việc và thời gian làm việc của nó có thể được lưu trữ và lặp lại bởi máy móc sau này.



Mẫu các thao tác từ hướng dẫn cho thao tác tự động



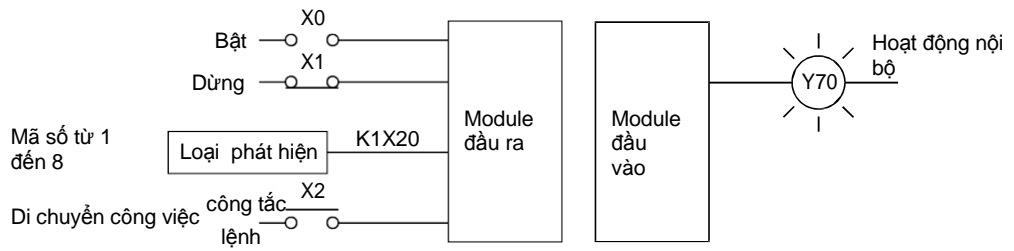
Tên dự án	QA-9
Tên chương trình	MAIN





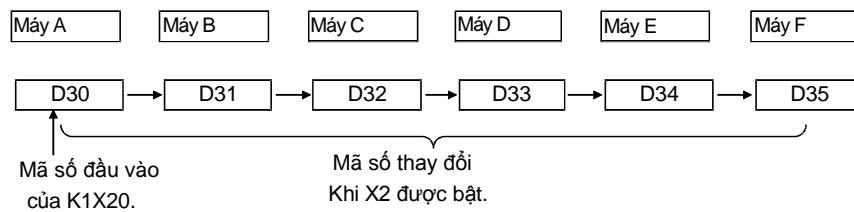
Phụ Lục 4.25 Ví dụ ứng dụng về bù dữ liệu

Công việc được truyền tải cùng với mã số của chúng, và dữ liệu đăng ký của máy móc chế biến được phân tích để máy làm việc theo mã số của chúng.

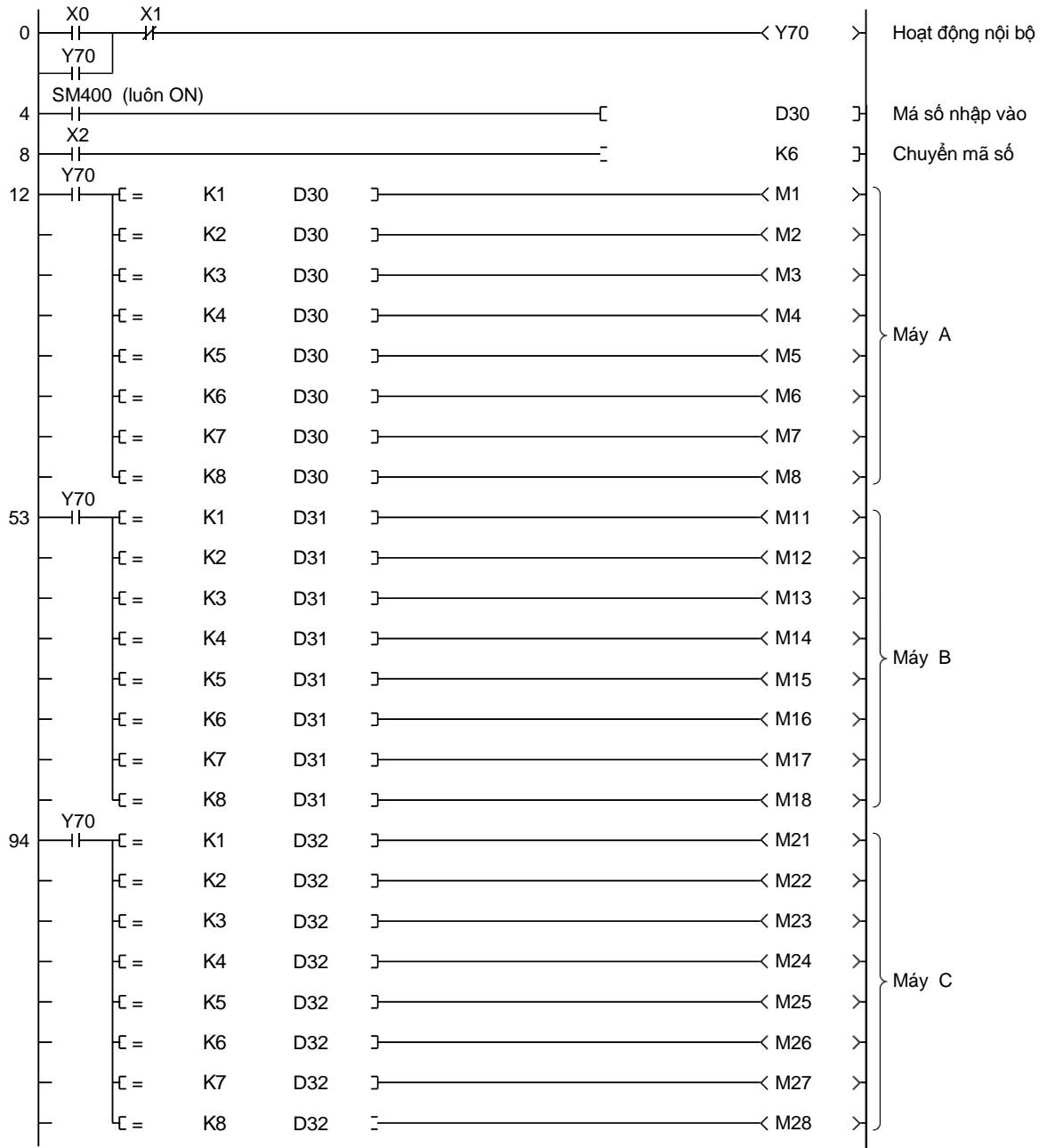


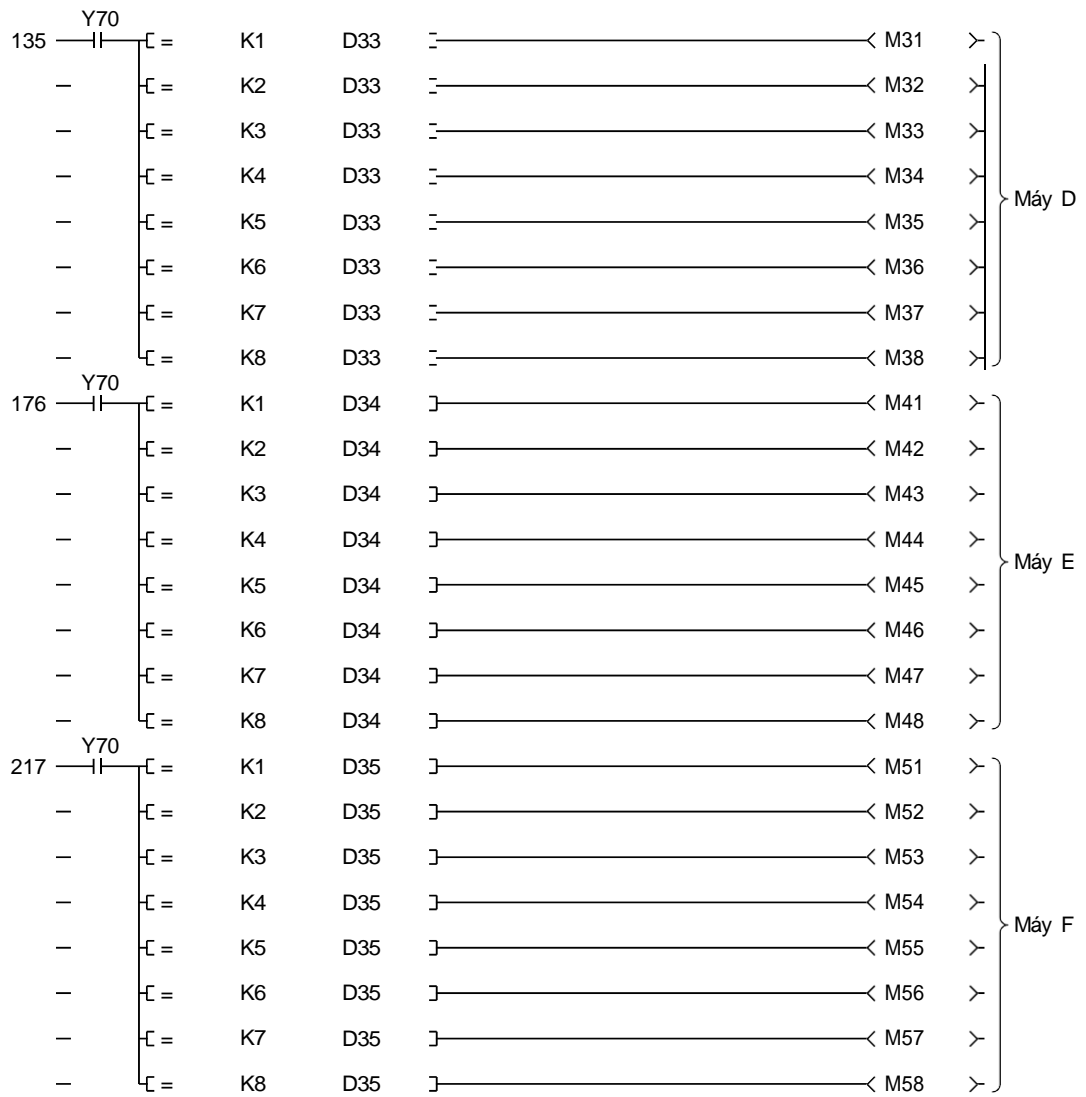
Máy	Thanh ghi dữ	Mã 1	Mã 2	Mã 3	Mã 4	Mã 5	Mã 6	Mã 7	Mã 8
A	D30	M1	M2	M3	M4	M5	M6	M7	M8
B	D31	M11	M12	M13	M14	M15	M16	M17	M18
C	D32	M21	M22	M23	M24	M25	M26	M27	M28
D	D33	M31	M32	M33	M34	M35	M36	M37	M38
E	D34	M41	M42	M43	M44	M45	M46	M47	M48
F	D35	M51	M52	M53	M54	M55	M56	M57	M58

Một mã số được lưu trong thanh ghi, và M tương ứng với số lượng lưu trữ được kích hoạt để máy làm việc.



Tên dự án	QA-12
Tên chương trình	MAIN

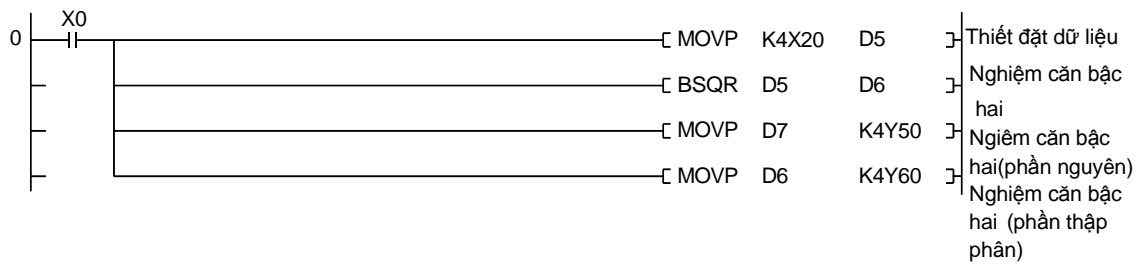




Tên dự án	QA-14
Tên chương trình	MAIN

Phụ Lục 4.26 Ví dụ về chương trình hoạt động tính căn bậc 2 của số liệu

Các dữ liệu được lưu trữ trong D5 được tính toán thành căn bậc 2 và kết quả được lưu trữ trong D6 và D7.



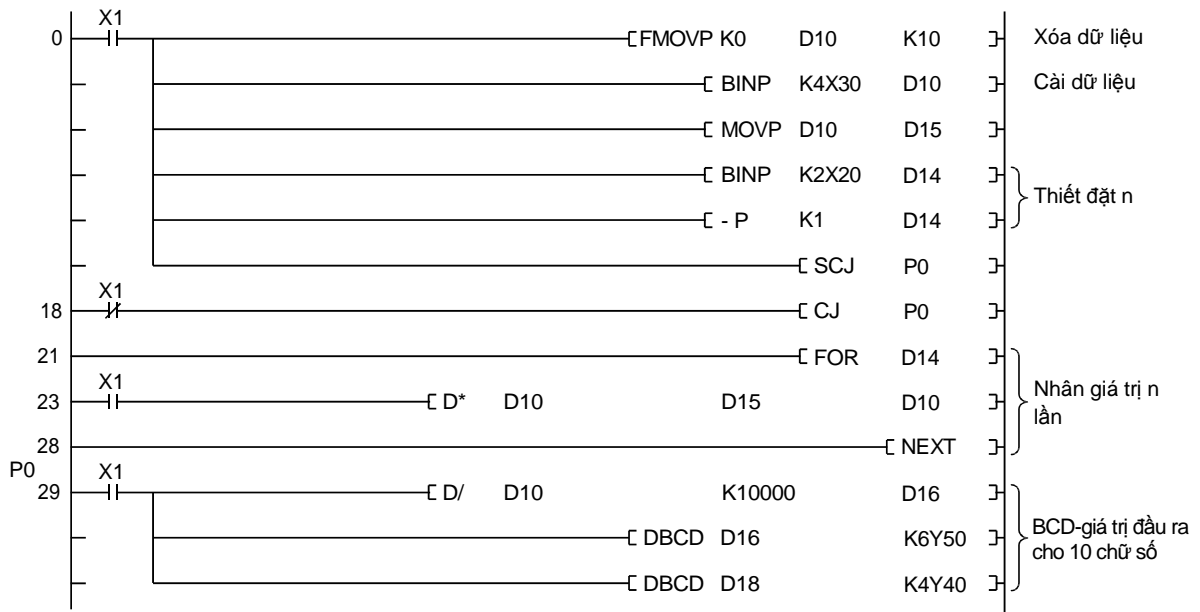
kết quả việc tính căn bậc hai được lưu trữ như sau: .

$\frac{\sqrt{D5}}{0 \text{ to } 9999}$ <small>(BCD giá trị)</small>	$\frac{D6}{0 \text{ to } 9999}$ <small>(BCD giá trị)</small>	$\cdot \frac{D7}{0 \text{ to } 9999}$ <small>(BCD giá trị)</small>	————— Làm tròn đến số thập phân thứ 5. Như vậy, số thập phân thứ 4 được làm tròn ±1.
---	--	--	--

**Lưu ý:**  
 QCPUs cung cấp hướng dẫn hoạt động căn bậc hai cho dữ liệu trong một số thực (dấu chấm) định dạng.

Tên dự án	QA-15
Tên chương trình	MAIN

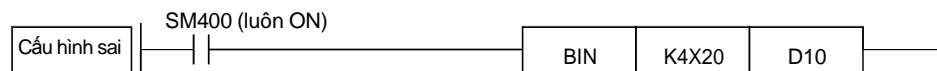
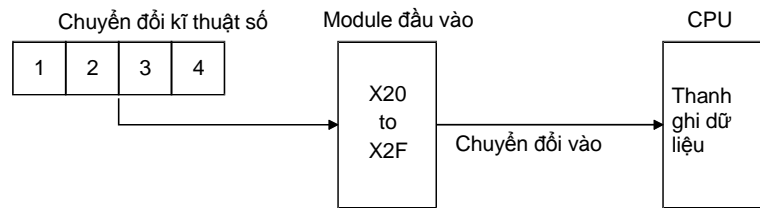
Phụ lục 4.27 Ví dụ về chương trình hoạt động tính toán năng lượng của dữ liệu  
 Một giá trị được lưu trữ trong D10 và được tính năng lượng n-th của nó ("n" là 1 giá trị được lưu trữ trong D14) và kết quả được lưu trong D10.



**CHÚ**

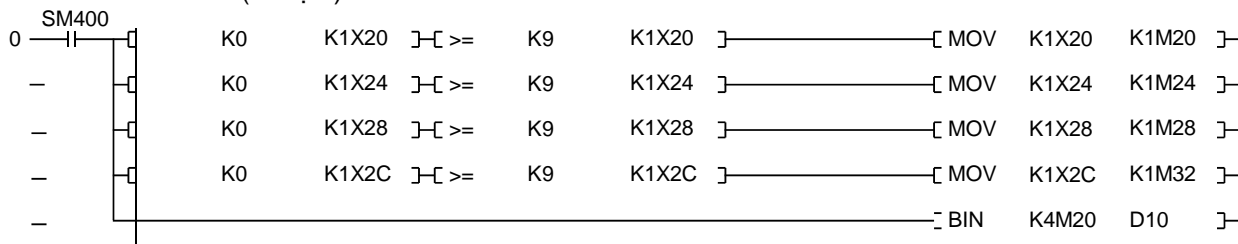
Một hoạt động lỗi xảy ra khi một giá trị trong D10 vượt quá 2147483647..

Phụ lục 4.28 Chương trình sử dụng chuyển đổi kỹ thuật để nhập dữ liệu  
 Khi 1 giá trị của bộ chuyển đổi kỹ thuật số luôn luôn là đầu vào và được lưu trữ ở D10 của bộ điều khiển khả trình.

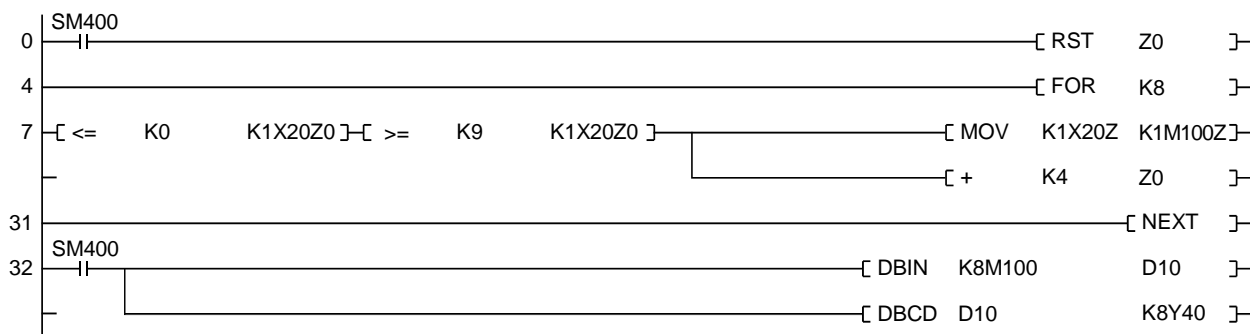


Trong chương trình trên, việc thay đổi giá trị của bộ chuyển đổi kỹ thuật số trong khi bộ điều khiển khả trình đang RUN có thể gây ra các mã khác từ 0-9 tùy thuộc vào thời gian của sự thay đổi mà có thể gây ra một lỗi hoạt động của CPU. Để tránh điều này, viết chương trình như sau:

(Ví dụ 1) Đối với 4 chữ số từ X20 đến X2F



(Ví dụ 2) Đối với 8 chữ số từ X20 đến X3F



Phụ Lục 4.29 Hiển thị số lỗi và số lỗi sử dụng chương trình phát hiện lỗi.

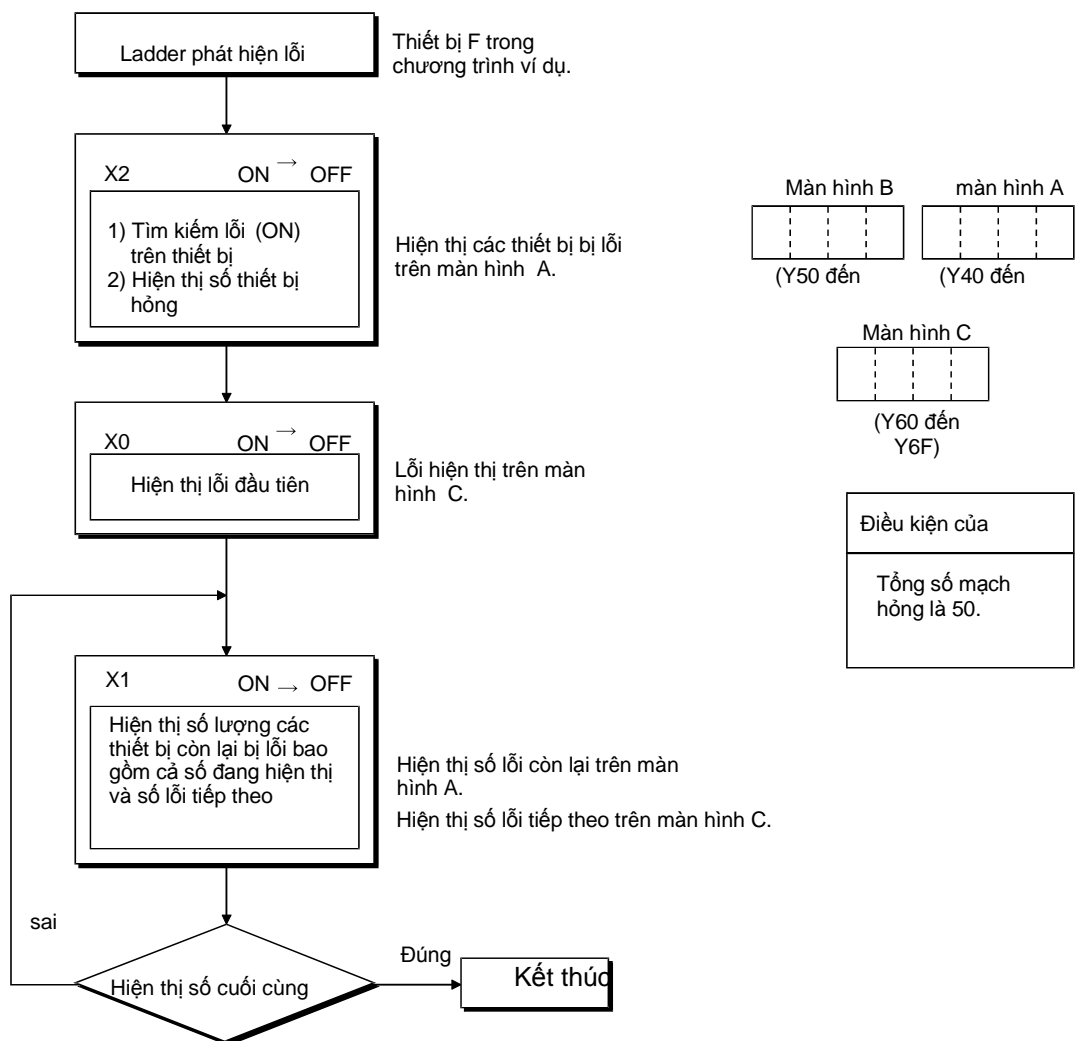
Chương trình tuần tự sau đây sẽ hiển thị số lượng bit thiết bị được bật (như X, M, và F) trong số rất nhiều các bit thiết bị được sử dụng liên tục, cùng với số thiết bị của chúng.

[Ví dụ ứng dụng]

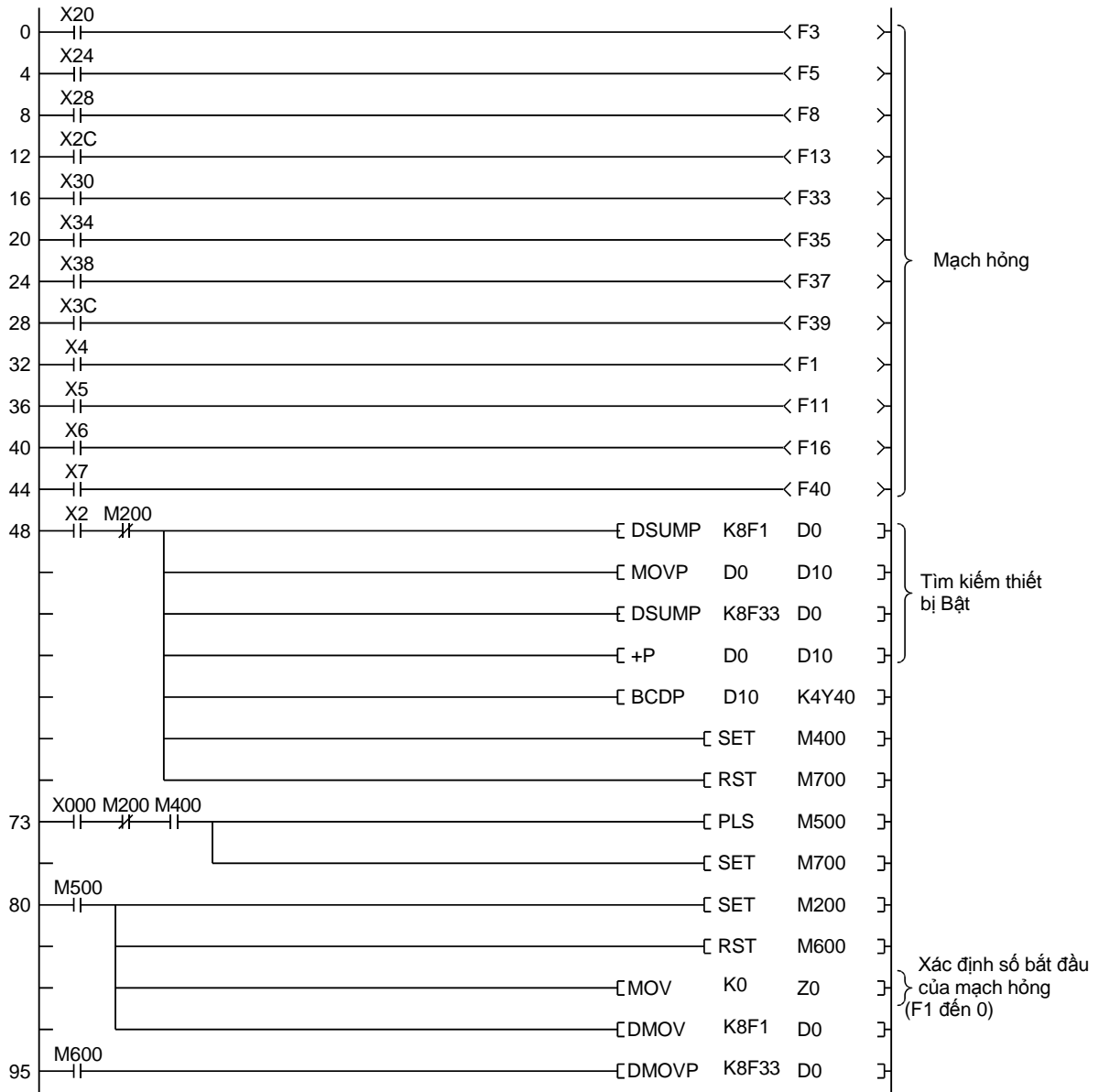
Khi M hay F được sử dụng như một thiết bị đầu ra của một chương trình phát hiện lỗi, sử dụng các chương trình sau đây để có được một số lỗi nhất định từ các lỗi.

[Chuỗi dòng chảy chương trình]

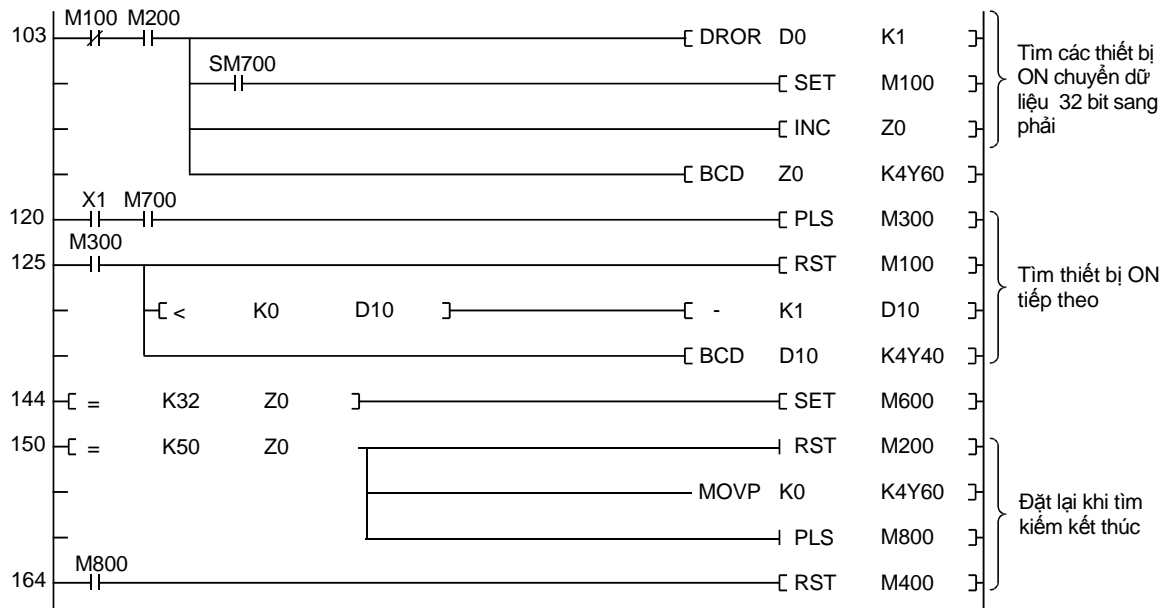
(Phương thức hoạt động)



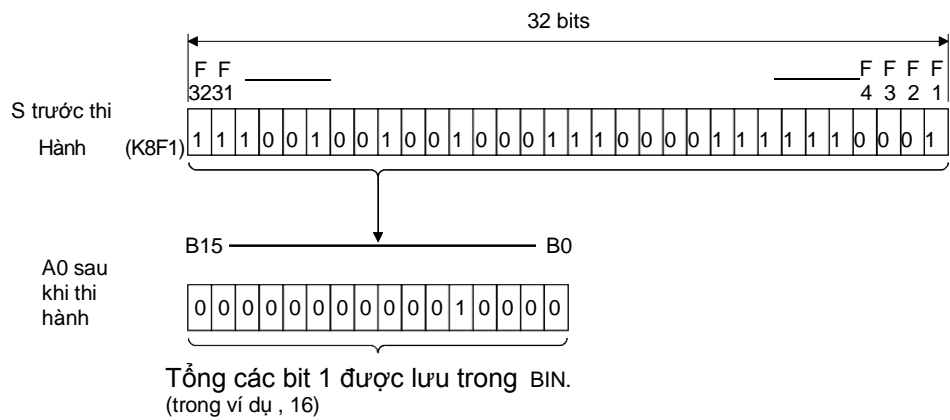
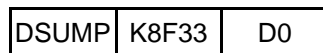
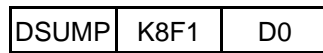
Tên dự án	QA-31
Tên chương trình	MAIN



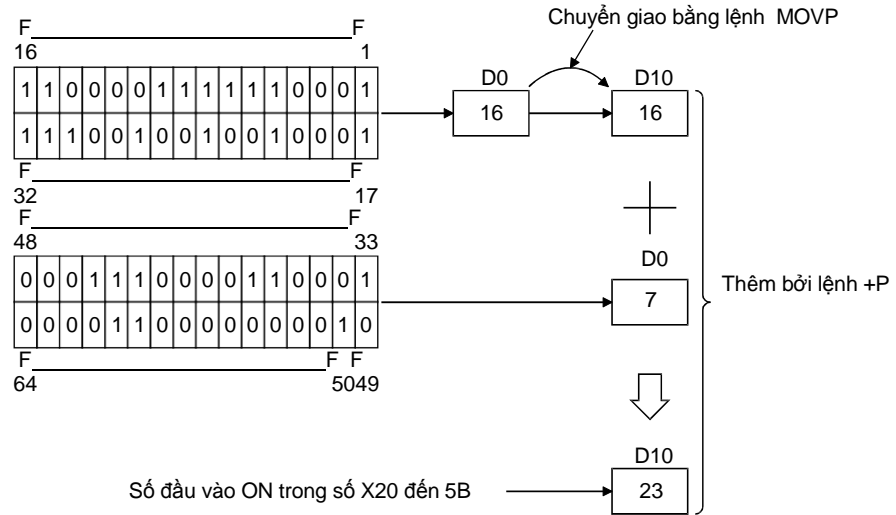




(1) Tìm các thiết bị Bật

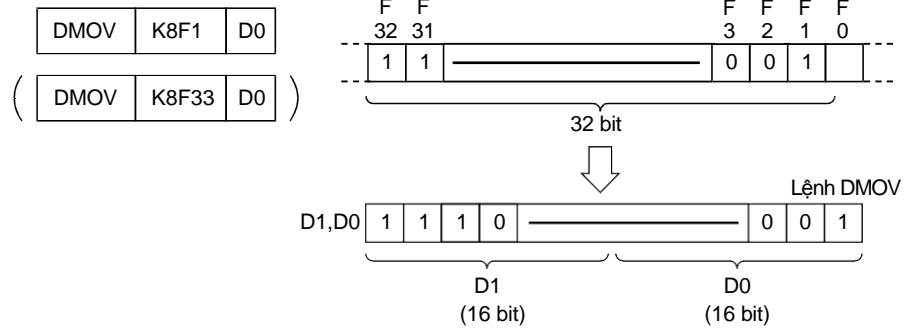


Bật X2 lưu trữ số của các bit được bật trong số F1 đến F64 đến D10 và hiển thị nó.

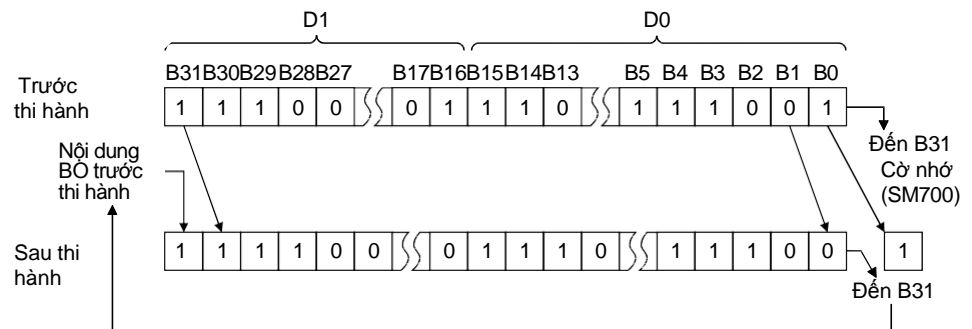


(2) Tìm kiếm thiết bị bật chuyển dữ liệu sang phải

DROR	D0	K1
------	----	----



DROR	D0	K1
------	----	----



- (a) Bật X0 thiết lập chuyển dữ liệu (D0 và D1). Sau đó, dữ liệu được thay đổi 1 bit tại mỗi lần scan cho đến khi bit được bật được phát hiện. Khi bit được bật được tìm ra, dịch chuyển dừng lại vì scan ( SM700 bật ), và số tích lũy của chuyển đổi ( tương đương số hiệu thiết bị ) được hiển thị.
- (b) Mỗi lần X1 được bật, bit tiếp theo được bật được tìm ra và số hiệu thiết bị phát hiện được hiển thị. Đồng thời, 1 bị trừ bởi bit được bật đã được thu trước để hiển thị các số còn lại của bit được bật.

Phụ lục 5 Bộ nhớ và tập tin được xử lý bởi CPU module

- Dữ liệu được lưu trữ trong bộ nhớ.

Bảng dưới đây liệt kê các dữ liệu và số ổ đĩa có thể được lưu trữ trong bộ nhớ chương trình, bộ nhớ RAM tiêu chuẩn, ROM tiêu chuẩn, và thẻ nhớ.

Mục	Bộ nhớ gắn liền với module CPU			Bộ nhớ (RAM)	Bộ nhớ (ROM)		Tên tập tin và phần mở rộng	Nhận xét
	Bộ nhớ chương trình	RAM chuẩn	ROM chuẩn	Thẻ SRAM	Thẻ Flash	Thẻ ATA		
	Ổ đĩa 0 <sup>*1</sup>	Ổ đĩa 3 <sup>*1</sup>	Ổ đĩa 4 <sup>*1</sup>	Ổ đĩa 1 <sup>*1</sup>	Ổ đĩa 2 <sup>*1</sup>			
Tham biên	O	X	O	O	O	O	PARAM.QPA	1 data/ Ổ đĩa
Module chức năng thông minh	O	X	O	O	O	O	IPARAM.QPA	1 data/ Ổ đĩa
Chương trình	⊙	X	O <sup>*3</sup>	O <sup>*4</sup>	O <sup>*4</sup>	O <sup>*4</sup>	***.QPG	-
Chú thích thiết bị	O <sup>*5</sup>	X	O <sup>*6</sup>	O <sup>*6</sup>	O <sup>*6</sup>	O <sup>*6</sup>	***.QCD	-
Giá trị thiết bị	O	X	O	O	O	O	***.QDI	-
Dữ liệu thiết bị	X	X	O	X	X	X	***.QST	-
Thành ghi tập tin	X	O <sup>*7*8</sup>	X	O	O <sup>*9</sup>	X	***.QDR	-
Thiết bị nội bộ	X	O <sup>*7</sup>	X	O	X	X	***.QDL	1 data/CPU module
Lấy mẫu tập tin	X	O <sup>*7</sup>	X	O	X	X	***.QTD	-
Dữ liệu lỗi	X	X	X	X	X	X	***.QFD	-
Tập tin lưu trữ dữ liệu	X	X	O	X	X	X	DEVSTORE.QST	-
Module lưu trữ lỗi file	X	O	X	X	X	X	IERRLOG.QIE	-
Sao lưu file	X	X	X	O	O	O	MEMBKUP0.QBP	-
Dữ liệu người dùng thiết bị tự động lập trình	X	X	O	X	X	O <sup>*10</sup>	***.***	-
Thiết lập người dùng	O	X	X	X	X	X	-	-

⊙: Yêu cầu, O: Cất giữ, X: Không thể cất giữ

\*1: Một số ổ đĩa được sử dụng để xác định một bộ nhớ được viết / đọc bởi các thiết bị bên ngoài bằng cách sử dụng một chương trình tuần tự hoặc giao thức MC.

Kể từ khi tên bộ nhớ được sử dụng để xác định các mục tiêu bộ nhớ trong GX Works2, số lượng ổ đĩa cần phải được xem xét.

\*2: Lưu trữ các thông số chức năng Module thông minh trong các ổ đĩa cùng với các thông số.

Khi chúng được lưu trữ trong các ổ đĩa khác nhau, các thông số chức năng Module thông minh không có hiệu lực.

\*3 Một chương trình được lưu trữ trong ROM tiêu chuẩn không thể được thực thi. Lưu trữ các chương trình vào bộ nhớ chương trình trước khi thực hiện.

\*4: Để thực hiện một chương trình được lưu trữ trong thẻ nhớ, làm cho các thiết lập trong Boot tab File của cửa sổ Thiết lập thông số Q

\*5: Các ý kiến thiết bị không thể được đọc bởi một chương trình hướng dẫn trong chuỗi

\*6 Đọc từ một chương trình đòi hỏi nhiều trình tự quét..

\*7: Chỉ có mỗi một trong những tập tin đăng ký, một thiết bị địa phương, và lấy mẫu dấu vết tập tin có thể được lưu trữ trong bộ nhớ RAM tiêu chuẩn

\*8: Đối với số lượng đăng ký có thể lưu trữ tập tin, hãy tham khảo hướng dẫn sử dụng chức năng Giải thích QnUCPU của người sử dụng, chương trình cơ bản.

\*9: Một chương trình tuần tự chỉ cho phép đọc. Không có dữ liệu có thể được viết từ các chương trình tuần tự.

\*10: Dữ liệu có thể được viết hoặc đọc các hướng dẫn sau đây.

- SP.FREAD (batch-đọc dữ liệu từ các tập tin chỉ định trong thẻ nhớ.)

- SP.FWRITE (batch-ghi dữ liệu vào tập tin chỉ định trong thẻ nhớ.)

\*11: Thiết lập một khu vực được sử dụng bởi hệ thống.

- Khả năng bộ nhớ và sự cần thiết của việc định dạng.
- Các bảng dưới đây liệt kê các khả năng bộ nhớ và sự cần thiết của định dạng của mỗi bộ nhớ.

		Q00UCPU	Q00UCPU	Q01UCPU	Q02UCPU	Q03UD(E)CPU	Định dạng
Bộ nhớ chương trình		10K bước (40K byte)		15K bước (60K byte)	20K bước (80K byte)	30K bước (120K byte)	*1
ROM tiêu chuẩn		256K byte	512K byte			1024K byte	Không cần
RAM tiêu chuẩn		-	128K byte			192K byte	*1
Thẻ nhớ	Thẻ SRAM	-			Q2MEM-1MBS: 1M byte Q2MEM-2MBS: 2M byte Q3MEM-4MBS: 4M byte Q3MEM-8MBS: 8M byte		Cần (sử dụng GX Works2.)
	Thẻ Flash	-			Q2MEM-2MBF: 2M byte Q2MEM-4MBF: 4M byte		Không cần
	Thẻ ATA	-			Q2MEM-8MBA: 8M byte Q2MEM-16MBA: 16M byte Q2MEM-32MBA: 32M byte		Cần (sử dụng GX Works2.)

		Q04UD(E)H CPU	Q06UD(E)H CPU	Q010UD(E)H CPU	Q13UD(E)H CPU	Q20UD(E)H CPU	Q26UD(E)H CPU	Định dạng
Bộ nhớ chương trình		40K bước (160K byte)	60K bước (240K byte)	100K bước (400K byte)	130K bước (520K byte)	200K bước (800K byte)	260K bước (1040K byte)	*1
ROM tiêu chuẩn		1024K byte			2048K byte	4096K byte		Không cần
RAM tiêu chuẩn		256K byte	768K byte	1024K byte		1280K byte		*1
Thẻ nhớ	Thẻ SRAM	Q2MEM-1MBS: 1M byte Q2MEM-2MBS: 2M byte Q3MEM-4MBS: 4M byte Q3MEM-8MBS: 8M byte						Cần sử dụng GX Works2.)
	Thẻ Flash	Q2MEM-2MBF: 2M byte Q2MEM-4MBF: 4M byte						Không cần
	Thẻ ATA	Q2MEM-8MBA: 8M byte Q2MEM-16MBA: 16M byte Q2MEM-32MBA: 32M byte						Cần(sử dụng GX Works2.)

\*1: Khi nội dung bộ nhớ trở nên không xác định trong tình trạng ban đầu hoặc do sự kết thúc vòng đời của pin, bộ nhớ sẽ tự động định dạng sau khi các bộ lập trình điều khiển được tắt nguồn và sau đó thiết lập lại. Hãy chắc chắn để định dạng bộ nhớ trong GX Works2 trước khi sử dụng.

Phụ lục 6.

So sánh với GX Developer (thay đổi)

(1) Hỗ trợ CPU modules

Bảng sau đây liệt kê các module CPU được hỗ trợ trong GX Works2

Chuỗi thiết bị khả trình	Loại thiết bị khả trình
QCPU (Mô hình Q)	Mô hình hiệu suất cao QCPU (Q02, Q02H, Q06H, Q12H, Q25H)
	Universal Q-CPU (Q00UJ, Q00U, Q01U, Q02U, Q03UD, Q03UDE, Q04UDH, Q04UDEH, Q06UDH, Q06UDEH, Q10UDH, Q10UDEH, Q13UDH, Q13UDEH, Q20UDH, Q20UDEH, Q26UDH, Q26UDEH, Q50UDEH, Q100UDEH)

Bảng sau đây liệt kê các module CPU mà không được hỗ trợ trong GX Works2. Sử dụng GX Developer cho module CPU sau.

Chuỗi thiết bị khả trình	Loại thiết bị khả trình
QCPU (Mô hình Q)	Mô hình cơ bản QCPU (Q00J, Q00, Q01)
	Quy trình CPU (Q02PH, Q06PH, Q12PH, Q25PH)
	Dư thừa CPU (Q12PRH, Q25PRH)
	Nhập / xuất từ xa(QJ71LP21, QJ71BR11)
QCPU (chuyển động)	Tất cả loại thiết bị khả trình
QCPU (mô hình A)	Tất cả loại thiết bị khả trình
QSCPU	Tất cả loại thiết bị khả trình
QnACPU	Tất cả loại thiết bị khả trình
ACPU	Tất cả loại thiết bị khả trình
Bộ điều khiển chuyển động (QCPU)	Tất cả loại thiết bị khả trình
CNC (M6, M7)	Tất cả loại thiết bị khả trình

(2) Các tính năng không được hỗ trợ

Bảng sau đây liệt kê các tính năng mà không được hỗ trợ trong GX Works2.

Sử dụng GX Developer, GX Simulator, hoặc GX Configurator cho các tính năng sau đây.

Chức năng không được hỗ trợ		Thay thế S/W
Chức năng trực tuyến	Chức năng TEL	
Chức năng khắc phục lỗi cho chương trình ladder	Chức năng Điều kiện hiển thị / thiết lập điều kiện Thiết bị hiển thị ngừng	GX Developer GX Simulator
	Chức năng đo thời gian quét	
	thi hành chức năng bỏ qua / Các linh kiện / Bước	
Chức năng khắc phục lỗi cho chương trình ST	Chức năng khắc phục lỗi	
	Chức năng điểm dừng	
Chức năng thông minh lập trình module	Chức năng hỗ trợ FB giao thức	GX Configurator-SC
Chức năng thông minh khắc phục lỗi module	Chức năng hỗ trợ khắc phục lỗi	
Chức năng trực tuyến để định vị module	Chức năng tìm vết	GX Configurator-QP
	Chức năng trình giám sát hệ thống	
	Chức năng chế độ kiểm tra	
Chức năng khởi tạo giá trị thiết bị	Chức năng đăng kí bộ nhớ thiết bị	GX Developer
	Chức năng in	
Chức năng mật khẩu	Chức năng việc đăng kí mật khẩu cho dữ liệu trong dự án	GX Developer
Tương tác với GX Explorer	Khởi động bằng GX Explorer	GX Developer
Tương tác với Nhà phát triển PX	PX Developer	GX Developer
Tương tác với GX Converter	I/O chức năng với GX Converter	GX Developer
Nhập vào định dạng in MEDOC	Nhập vào trong định dạng in MEDOC	GX Developer
Chức năng trực tuyến	Chẩn đoán sự cố module thông minh từ trình giám sát hệ thống	GX Developer GX Configurator
Thống kê chức năng tìm vết	Thống kê chức năng lần tìm ra có điều kiện trên số bước	GX Developer

(3) Các loại dự án được hỗ trợ

Bảng sau đây liệt kê các loại dự án được hỗ trợ trong GX Works2.

Loại dự án	Mô tả
Dự án đơn giản ( không có nhãn )	<p>Tương đương với dự án " không sử dụng sử dụng nhãn" của Nhà phát triển GX.</p> <p>1) Khi Dự án được tạo ra trong " không sử dụng nhãn" của Nhà phát triển GX được đọc với GX Works2, dự án trở thành Simple dự án( không có nhãn ).</p> <p>2) Khi Dự án được tạo ra trong Simple dự án( không có nhãn ) của GX Works2 được đọc với Nhà phát triển GX, dự án trở thành dự án " không sử dụng nhãn".</p>
Dự án đơn giản ( với nhãn )	<p>Tương đương với dự án " Sử dụng nhãn" của Nhà phát triển GX.</p> <p>1) Khi dự án được tạo ra trong " Sử dụng nhãn" của GX Nhà phát triển được đọc với GX Works2, dự án trở thành Simple dự án( với nhãn ).</p> <p>2) Khi dự án được tạo ra trong dự án Đơn giản ( với nhãn ) của GX Works2 được đọc với Nhà phát triển GX, dự án trở thành dự án " Sử dụng nhãn".</p>
Dự án dự án	<p>Ở GX Works2, " lập trình có cấu trúc" luôn sẵn sàng. Lập trình có cấu trúc thu được trong khi tạo ra POUs và tổ hợp chúng ( thanh ghi tác vụ trong tập tin chương trình ).</p> <p>1) Khi dự án được tạo ra trong " Sử dụng nhãn" với ST của Nhà phát triển GX được đọc với GX Works2, dự án trở thành " Structured Dự án".</p> <p>2) Dự án được tạo ra trong " Cấu trúc Dự án" của GX Works2 không thể đọc với Nhà phát triển GX.</p>

(a) Sử dụng chức năng dự án

Trước khi sử dụng các chức năng trong dự án GX Works2, xem xét các biện pháp phòng ngừa sau đây.

Chức năng	Mô tả ( sự khác biệt giữa Nhà phát triển GX và GX Works2 )	
	GX Developer	GX Works2
Bảo vệ dự án	Bằng cách cài đặt các dự án như "chỉ theo dõi", các dự án có thể được bảo vệ trên mỗi máy tính cá nhân.	Bằng cách thiết lập dự án như " chỉ - đọc với chức năng " Security ", dự án -bởi - dự án sự bảo vệ là hiện đang có sẵn.
Thay đổi loại dự án	Loại dự án không thể thay đổi từ " không sử dụng nhãn" thành " Sử dụng nhãn".	Những thay đổi loại hình dự án sau đây đang có sẵn. 1) Từ "Dự án cơ bản(không nhãn)" đến " Dự án cơ bản (có nhãn)" 2) Từ "Dự án cơ bản (với nhãn)" đến "Cấu trúc dự án". * Loại dự án không thể thay đổi trực tiếp từ " dự án cơ bản (với nhãn )" đến "Cấu trúc dự án"
Đọc định dạng dự án Nhà phát triển GX	Chọn [Dự án] → [Open Other Dự án] có thể đọc định dạng protect Nhà phát triển GX.	
Đọc dự án cấu hình GX - định dạng QP.	Chọn [Dự án] → [Intelligent Function Module] → [Import GX Configurator-QP Data] có thể đọc dự án cấu hình GX - định dạng QP..	
Sao chép dữ liệu dự án này sang dự án khác	Nó được kích hoạt trên các dự án hộp thoại copy	Sao chép và dán những thứ hiện có trong cửa sổ window.

(4) Ngôn ngữ lập trình được hỗ trợ bởi mỗi loại dự án.

Bảng sau đây liệt kê ngôn ngữ lập trình được hỗ trợ bởi mỗi dự án loại GX Works2.

Kiểu dự án	Hỗ trợ ngôn ngữ lập trình
Simple dự án (không có nhãn)	Ladder, SFC (MELSAP3)
Simple dự án (có nhãn)	Ladder, SFC (MELSAP3) * Chương trình hỗ trợ phần tử : nhãn, cấu trúc, khối chức năng
Cấu trúc dự án	Ladder, SFC (MELSAP3), cấu trúc ladder, ST * Chương trình hỗ trợ phần tử : nhãn, cấu trúc, khối chức năng, block hàm, thư viện

Ngôn ngữ lập trình sau không được hỗ trợ đưa vào GX Works2. Nhà phát triển GX sử dụng cho ngôn ngữ lập trình sau.

Kiểu dự án	Hỗ trợ ngôn ngữ lập trình
Danh sách	1) Khi GX Works2 đọc chương trình tạo ra với danh sách đưa vào nhà phát triển GX, nó có thể được hiển thị hay hiệu chỉnh đưa vào ladder. 2) Khi nhà phát triển GX đọc chương trình tạo ra với ladder đưa vào GX Works2, nó có thể được hiển thị hay hiệu chỉnh đưa vào danh sách
MELSAP-L	1) Khi GX Works2 đọc chương trình tạo ra với MELSAP-L đưa vào nhà phát triển GX, nó có thể được hiển thị hay hiệu chỉnh đưa vào ladder. 2) Khi nhà phát triển GX đọc chương trình tạo ra với SFC ( MELSAP3 ) đưa vào GX Works2, nó có thể được hiển thị hay hiệu chỉnh đưa vào MELSAP-L.

(a) Sử dụng ngôn ngữ ladder

Trước khi sử dụng ngôn ngữ ladder đưa vào GX Works2, xem xét phòng ngừa sau.

Chức năng	Mô tả ( khác nhau giữa nhà phát triển GX và GX Works2	
	GX Developer	GX Works2
Chương trình tính đàn hồi thiết bị biệt hiệu	Nó được cho phép bằng " Biệt hiệu" hàm	Sử dụng " Nhân".
Đoạn một phần chương trình vào POU ( marcos)	Nó được cho phép bằng " định nghĩa macro / nhập" chức năng	Sử dụng " khối chức năng"
Phát hiện / Thay thế câu lệnh / thiết bị / phòng thí nghiệm chữ L	Phát hiện được cho phép bằng trực tiếp gõ câu lệnh / thiết bị / nhãn đưa vào" Đọc phương thức".	Nhấn phím Space trên bộ lắp ladder cho phép phát hiện đơn.
Kiểm tra sử dụng trạng thái của thiết bị / nhãn	Nó được kích hoạt bằng chức năng" Danh sách Tham chiếu chéo" và" chức năng" Danh sách Thiết bị Đã sử dụng"	Chọn [Find/Replace] → [Cross Reference], or [Find/Replace] → [Device List].
Hợp nhất chương trình	Được cho phép bằng" Hợp nhất vào Số liệu"	Sử dụng sao chép và dán trên bộ lắp nhãn
Kiểm tra	Khối không tương ứng	Cửa sổ Kết quả Kiểm tra thể hiện rõ những thứ sau đây:" không gì có thể sánh bằng vùng chương trình"," nguồn kiểm tra chỉ chứa đựng chương trình" và" dự định kiểm tra chỉ chứa đựng chương trình ".



(b) Sử dụng SFC ( MELSAP3 ) ngôn ngữ

Trước khi dùng SFC ( MELSAP3 ) ngôn ngữ đưa vào GX Works2, xem xét phòng ngừa sau.

Chức năng	Mô tả ( khác nhau giữa nhà phát triển GX và GX Works2 )	
	GX Developer	GX Works2
Thay đổi số khối	Nó được cho phép bằng" sao chép và dán" chức năng đưa vào danh sách khối	Mỗi số liệu khối được hiển thị đưa vào cửa sổ Dự án, và số khối có thể được thay đổi đưa vào quyền sở hữu của số liệu mỗi khối.  * Chọn [ Nhìn ] → [ Mở SFC Blocklist ] có thể biểu lộ danh sách block đương lượng để
Tự động cuộn	Sơ đồ khối tin có thể mở bằng khởi hành khối	Chọn lọc [View] → [Open Zoom/Start Destination Block] có thể mở nó.
Thay đổi số khối	Nó được cho phép bằng" sao chép và dán" chức năng đưa vào danh sách khối	Mỗi số liệu khối được hiển thị đưa vào cửa sổ Đề án, và số khối có thể được thay đổi đưa vào quyền sở hữu của số liệu mỗi
Vận hành mở / chuyển tiếp điều kiện chương trình	Di chuyển con trỏ trên biểu đồ SFC có thể hiển thị thu phóng ( vận hành tại đầu ra / điều kiện chuyển tiếp ).	Chọn [View] → [Mở Zoom/Start Destination Block] có thể mở. Hoặc nhấn đúp trong khi ấn phím Crt cũng có thể mở nó
	Multiple zooms ( thao tác tại đầu ra /điều kiện chuyển tiếp ) có thể hiển thị cùng một lúc . * Thay đổi" Thiết lập của Zoom Display" tùy chọn có thể chuyển đổi hiển thị trong cửa sổ giống như phương pháp của Nhà phát triển GX	

(c) Sử dụng nhãn

Trước khi dùng nhãn đưa vào GX Works2, xem xét biện pháp phòng ngừa sau..

Chức năng	Mô tả ( khác nhau giữa nhà phát triển GX và GX Works2 )	
	GX Developer	GX Works2
Thiết bị kiểm tra tự động được phân công cho nhãn	Nó được cho phép bằng" Biểu diễn áp đặt thiết bị" chức năng của bộ lắp nhãn	Kiểm tra bộ lắp ladder bằng cách chọn [View] → [Device Display].
Import/Export thiết bị chú thích đến nhãn	Nó được cho phép bằng" thiết bị chú thích đầu vào " chức năng và" thiết bị chú thích đầu ra" chức năng.	Sử dụng sao chép và dán trên nhãn người biên tập và thiết bị chú thích của người biên tập
Sử dụng trở - loại nhãn	Con trở cục bộ được gán.	Cho dự án với kí hiệu, 2048 trở được xác lập mặc định trong" Con trở thông thường NO." trong"Hệ thống PLC " cũ đích của Tham biến PLC .
Không thể dùng được từ riêng cho tên nhãn	Định nghĩa của các từ riêng là khác giữa GX Nhà phát triển và GX Works2	

(d) Sử dụng khối chức năng

Trước khi dùng khối chức năng đưa vào GX Works2, xem xét biện pháp phòng ngừa sau.

Chức năng	Mô tả
Sử dụng khối chức năng tạo ra ladder.	Khối chức năng tạo ra với ladder có thể được dùng trong chương trình ladder, chương trình ST, và đầu ra thao tác chương trình SFC.  *Khi sử dụng khối chức năng đã tạo với ladder cho chương trình ST, chọn [ Công cụ ] → [ Tùy chọn ] → [ Biên dịch ] → [ thiết lập cơ bản ] → " gọi khối chức năng Enable ' từ ladder đến Structured Ladder / FBD ' và ' từ Structured Ladder / FBD hoặc ST đến ladder".
Sử dụng khối chức năng tạo ra cấu trúc ladder.	Khối chức năng tạo ra với cấu trúc ladder có thể được dùng cho chương trình ladder, cấu trúc chương trình ladder và chương trình ST.
Sử dụng khối chức năng tạo ra ST.	Hàm block tạo ra với ST có thể được dùng cho chương trình ladder, cấu trúc chương trình ladder, và chương trình ST.  * Khi sử dụng Hàm block tạo ra với ST cho chương trình ladder , chọn lọc [Tool] → [ Options] → [Compile] → [Basic Setting] →" cho phép hàm block gọi ' từ ladder đến Cấu trúc ladder / FBD ' và ' từ Cấu trúc ladder / FBD hoặc ST tới ladder ".
Khi quyền lựa chọn" Cho phép khối chức năng kết nối ' từ ladder đến Cấu trúc sơ đồ ladder/ FBD ' và ' từ Cấu trúc ladder/ FBD hoặc ST tới sơ đồ	Khi VAR_IN_OUT đầu vào biến số và đầu ra biến số có nhãn khác nhau / thiết bị, giá trị biến số đầu vào luôn bằng giá trị biến số đầu ra.

(5) Sử dụng chú giải biến địa chỉ

Trước khi dùng chú giải biến địa chỉ đưa vào GX Works2, xem xét phòng ngừa sau.

Chức năng	Mô tả ( khác nhau giữa nhà phát triển GX và GX Works2 )	
	GX Developer	GX Works2
Xóa biến địa chỉ của thiết bị chưa được sử dụng	Nó được kích hoạt bằng cách xóa chú thích chức năng.	Sau khi kiểm tra các thiết bị không sử dụng bằng cách chọn Find/Replace] → [Device List], xóa chú giải biến địa chỉ trực tiếp.
Mẫu chú thích	Nhận xét mẫu của relay / đăng ký đặc biệt đặc biệt được cung cấp theo dạng dự án.	Chú thích của rơ le đặc biệt / thanh ghi đặc biệt và module chức năng thông minh có thể được nhập bởi chức năng " Nhập vào Sample Comment " trên trình soạn thảo.

(6) Sử dụng bộ nhớ biến địa chỉ

Trước khi sử dụng các bộ nhớ biến địa chỉ trong GX Works2, xem xét các biện pháp phòng ngừa sau đây.

Chức năng	Mô tả
Hiện thị bộ nhớ biến địa chỉ	Nhiều dây thiết bị có thể được hiển thị trong một cửa sổ. * Bằng cách chọn "All Range" khi thiết bị này là đầu vào, tất cả các phạm vi biến địa chỉ có thể được hiển thị trong một cửa sổ trong cùng một cách như của GX Developer.
Copy and past device memory data to Excel	Để sao chép và bộ nhớ thiết bị nhập dữ liệu vào Excel, chọn [Tool] → [đọc từ Excel File] / [Viết cho Excel File].

- (7) Sử dụng thiết bị giá trị ban đầu  
 Trước khi sử dụng thiết bị giá trị ban đầu trong GX Works2, xem xét các biện pháp phòng ngừa sau đây.

Chức năng	Mô tả ( khác nhau giữa nhà phát triển GX và GX Works2 )	
	GX Developer	GX Works2
Số lượng tối đa của dữ liệu giá trị khởi đầu của thiết bị đã được tạo lập	Chỉ có một tập hợp các dữ liệu có thể được tạo ra.	Lên đến 800 bộ dữ liệu có thể được tạo ra.
Hạn chế số lượng thiết bị	Số thiết bị phải có trong những điểm tối đa của mỗi bộ điều khiển lập trình của thiết bị.	Số lượng thiết bị phải nằm trong phạm vi thiết lập thiết bị của các thông số PLC.
Viết cho PLC / đọc từ PLC IC thẻ nhớ ghi / đọc	Chỉ có 1 dữ liệu có thể được đọc và viết.	Nhiều dữ liệu được lựa chọn có thể được đọc và viết.

- (8) Sử dụng chức năng trực tuyến  
 Trước khi sử dụng các chức năng trực tuyến tại GX Works2, xem xét các biện pháp phòng ngừa sau đây.

Chức năng	Mô tả ( khác nhau giữa nhà phát triển GX và GX Works2 )	
	GX Developer	GX Works2
Thiết lập điểm đến kết nối	Một đề án có thể chỉ chứa một tập hợp các "điểm đến kết nối" thông tin.	Một đề án có thể chứa nhiều bộ "đích kết nối" thông tin. Để thay đổi các "điểm kết nối" thông tin, chọn "Connecting Destination" trong cửa sổ đề án.
Viết / đọc dữ liệu đến / từ các module chức năng thông minh	Dữ liệu có thể được viết hoặc đọc đến / từ các module CPU và các chức năng module thông minh cùng một lúc.	Chọn [online] → [write to PLC] ghi dữ liệu vào module CPU và các chức năng module thông minh cùng một lúc. Cách chọn [online] → [Read from PLC] đọc dữ liệu từ module CPU và các chức năng module thông minh cùng một lúc.
Ghi dữ liệu vào ROM Flash của module CPU	Nó được kích hoạt bởi các chức năng "PLC write (Flash ROM)".	Các chức năng "PLC write (Flash ROM)" được tích hợp trong chức năng " write to PLC" Chọn [online] → [write to PLC].
Vận hành từ xa cửa sổ PLC để dự đoán Hệ thống cửa sổ màn hình.	Chọn [online] → [Operation Remote ] và [Diagnostics] → [System Monitor] / [PLC Diagnostics] bây giờ có thể hiển thị các hình ảnh Module và các trạng thái vận hành bộ điều khiển lập trình CPU là dễ dàng nhận thấy. Các hoạt động từ xa, hoạt động bộ nhớ, và thiết lập đồng hồ có thể được bắt đầu từ cửa sổ PLC Diagnostics.	

Chức năng	Mô tả
Đọc từ PLC	Thông tin mạng tính biểu tượng trong định dạng Developer GX không bao gồm các chương trình SFC. Đọc thông tin mạng tính biểu tượng về "đề án đơn giản (không có) Nếu các thông tin mạng tính biểu tượng của GX Developer hoặc GX IEC Developer được đọc ra, đề án sẽ không được soạn thảo.

(9) Sử dụng màn hình / chức năng debug

Trước khi sử dụng các chức năng theo dõi / debug trong GX Works2, xem xét các biện pháp phòng ngừa sau đây.

Chức năng	Mô tả
Nhập màn hình thiết bị	Các chức năng "nhập màn hình thiết bị" bây giờ là một cửa sổ hạn chế như một chức năng "xem" để nó có thể được hiển thị mà không chồng chéo với các chương trình soạn thảo, Device / nhân đã được kích hoạt để được nhập bằng cách kéo và thả từ trình soạn thảo chương trình và bật / tắt tình trạng của thiết bị và giá trị hiện tại của các thiết bị có thể được sửa đổi trên cửa sổ màn hình.
Thiết bị giám sát hàng loạt Đệm giám sát hàng loạt	Các chức năng "thiết bị giám sát hàng loạt" và "đệm theo dõi hàng loạt bộ nhớ" đang tích hợp để nhận ra khả năng hoạt động tương tự. Bật / tắt tình trạng của thiết bị và giá trị hiện tại của các thiết bị có thể được sửa đổi trên cửa sổ màn hình.
Giám sát và kiểm tra các chức năng Module thông	Để sử dụng các chức năng giám sát hoặc kiểm tra cho FL-net (OPCN-2) giao diện đơn vị và AS-i đơn vị làm chủ, thực hiện chức năng "watch" và "Device / Buffer màn hình hàng loạt bộ nhớ".

(10) Sử dụng chức năng in ấn

Trước khi sử dụng các chức năng in ấn trong GX Works2, xem xét các biện pháp phòng ngừa sau đây.

Chức năng	Mô tả
Thêm in các thông tin như báo cáo và chú thích thiết bị	Hình ảnh hiển thị được in hoặc xem trước. Để in thêm các thông tin như một bản tường trình và nhận xét thiết bị, đưa các mục thông tin trên màn hình và sau đó chọn [DỰ ÁN] → [Print Window] / [Print Preview Window].

(11) Sao chép lưu dữ liệu đề án

Trước khi sao chép dữ liệu dự án được lưu trong GX Works2, xem xét các biện pháp phòng ngừa sau đây..

Chức năng	Mô tả (khác nhau giữa GX Developer và GX Works2)	
	GX Developer	GX Works2
Sao chép dữ liệu dự án được lưu	Dữ liệu dự án được lưu có thể được sao chép bằng cách sao chép các tập tin trong thư mục tên dự án.	Sao chép tất cả các thư mục tên workspace và "workspacelist.xml" được tạo ra trong hệ thống cấp bậc giống như các thư mục tên workspace.

(12) Tính tương thích với GX Nhà phát triển

Cho tính tương thích giữa GX Nhà phát triển và GX Works2, xem xét biện pháp phòng ngừa sau..

Chức năng	Mô tả
Mở dự án trong các định dạng khác	Trước khi mở một dự án GX Developer "Sử dụng nhân" trong đó có một chương trình và khối chức năng có cùng tên, thì thay đổi tên dữ liệu trong GX Developer. Tên chức năng của ngôn ngữ ST khác nhau giữa GX Developer và GX Works2. Biên dịch chương trình và sửa lỗi.
Dự án xuất đến tập tin định dạng GX Developer	Dự án có thể áp dụng được là sau đây; 1) Simple dự án (không nhân) 2) Compiled Simple dự án (có nhân) dự án sử dụng nhân trong ngôn ngữ SFC được thi hành.  Dự án có thể được lưu trong định dạng Developer GX khi không có trường hợp nào trong số các trường hợp dưới đây được áp dụng. 1) Không có thiết lập được thiết lập. 2) Chiều dài của tên nhân vượt quá 16 ký tự. 3) Tên Nhân chứa tên thiết bị hoặc từ dành riêng. 4) Ký tự không hợp lệ được sử dụng. 5) Kiểu dữ liệu không được hỗ trợ bởi Nhà phát triển GX được sử dụng. 6) Giá trị không phải là hằng số được dùng trong hằng số. Dữ liệu thanh ghi đến nhân toàn cục là thiết lập như "Auto External" với mọi nhân cục bộ.

## (13) Tính tương thích với GX IEC Developer

Cho tính tương thích giữa GX IEC Developer và GX Works2, xem xét biện pháp phòng ngừa sau..

Chức năng	Mô tả
Mở dự án trong các định dạng khác	Tên chức năng của ngôn ngữ ST khác giữa GX IEC Nhà phát triển và GX Works2. Biên dịch chương trình và sửa lỗi.
Thư viện người dùng	Trước khi dùng thư viện người dùng GX IEC Developer mà mật khẩu là ẩn định thành, hủy bỏ mật khẩu trong GX IEC Developer.

## (14) Thao tác then chốt

Phần này giải thích sự khác biệt của thao tác then chốt giữa Nhà phát triển GX và GX Works2.

Chức năng		Mô tả	Phím tắt		
			GX Developer	GX Works2	
Sửa	Chế độ đọc	Kích hoạt chế độ đọc	Shift + F2	- (*1)	
	Chế độ ghi	Kích hoạt chế độ ghi	F2	- (*1)	
Tìm / Thay thế	Tham chiếu chéo	Hiện thị tham chiếu chéo	-		
	Danh sách thiết bị	Hiện thị danh sách thiết bị	-		
Chuyển	Chuyển đổi ( tất cả chương trình được chỉnh sửa )	Chuyển đổi tất cả chương trình được chỉnh sửa.	Ctrl + Alt + F4	-	
View	Danh sách dữ liệu Dự án	Thiết bị chuyển mạch hiển thị / không hiển thị danh sách dữ liệu dự án.	Alt + O	-	
	Chuyển đổi giữa Dự án danh sách dữ liệu và cửa sổ	Thiết bị chuyển mạch giữa Dự án danh sách dữ liệu và mỗi cửa sổ.	Alt + 7	-	
	Chuyển đổi giữa laddered và list	Thiết bị chuyển mạch giữa cửa sổ ladder và cửa sổ danh sách.	Alt + F1	-	
Thực tuyến	Giám sát	Giám sát ( tất cả cửa sổ )	Các ladder giám sát của tất cả chương trình mở.	Ctrl + F3	-
		Giám sát ( ghi chế độ )	Kích hoạt chế độ ghi trong thiết bị ladder giám sát.	Shift + F3	- (*2)
		Ngừng Giám sát ( tất cả cửa sổ )	Ngừng ladder giám sát với mọi chương trình mở.	Ctrl + Alt + F3	-
	Khắc phục lỗi	Thử nghiệm thiết bị	Bật hoặc tắt các thiết bị thử công hoặc thay đổi giá trị hiện tại.	Alt + 1	-
		Bỏ thi hành	Thi hành chương trình tuần tự chọn lọc trong bỏ qua thi hành.	Alt + 2	-
		Thi hành từng phần	Chương trình trình tuần tự thi hành từng phần.	Alt + 3	-
		bước thi hành	Thi hành bộ điều khiển lập trình được Cpu trong bước thi hành.	Alt + 4	-
	Thao tác từ xa	Thi hành thao tác từ xa.	Thi hành thao	-	

\*1: Trong GX Works2, chuyển mạch trình soạn thảo ladder đến chế độ đọc / ghi chế độ là không cần thiết. Ladder có thể chỉnh sửa bất kỳ lúc nào..

\*2: : Trong GX Works2, chuyển mạch trình soạn thảo ladder đến thiết bị hiển thị ( ghi chế độ ) trong thiết bị hiển thị ladder là không cần thiết..

Ngay cả trong việc giám sát ladder, ladder có thể được chỉnh sửa và ghi vào bộ điều khiển lập trình được trong trạng thái RUN.

Phụ lục 7 Tùy chỉnh phím tắt

Các phím tắt của mỗi chức năng có thể được tùy chỉnh.  
 Phím tắt tùy chỉnh có thể được đăng ký như là một khuôn mẫu và sử dụng.  
 Màn hình hiển thị

Chọn [Tool] → [Key Customize].



Mục	Mô tả
Phím tắt	-
Mục	Chọn mục từ danh sách nhóm phân loại bằng cửa sổ.
Lệnh	Chọn tên chức năng phím tắt được thay đổi.
Phím hiện hành	Hiện thị phím tắt được gán cho chỉ lệnh được lựa chọn.
Nhấn phím để gán	Xác định phím tắt mới để được gán. Nhấn một phím ( s ) trên bàn phím gán phím ( s ). Ví dụ) <b>Ctrl</b> + <b>5</b>
Hiện hành	Hiện thị tên menu mà phím tắt nhập được gán. Khi phím đã được gán cho chức năng khác, tên chức năng được hiển thị.
Khuôn mẫu	<ul style="list-style-type: none"> <li>• Chọn khuôn mẫu của phím tắt từ hộp danh sách.</li> <li>• Thiết lập mặc định</li> <li>• Thiết lập mặc định được thiết lập .</li> <li>• Thiết lập Định dạng GPPA</li> <li>• Thiết lập phím tắt tại lập trình ladder là thay đổi cùng thiết lập với ở GPPA.</li> </ul>

Nút màn hình

Assign

Gán phím tắt. Phím tắt gán được hiển thị trong " Current Key".

Delete

Xoá bỏ phím tắt đã chọn ở " Current Key ".

Register Current Setting as Template...

Màn hình Enter Template Name được hiển thị

Thanh ghi phím tắt gán như là một khuôn mẫu với tên.

Khuôn mẫu thanh ghi được hiển thị trong " Template ". "

Apply

Khuôn mẫu của phím tắt chọn lọc được áp dụng.

Delete

Xoá bỏ khuôn mẫu được chọn trong " Template ".

Import...

Nhập vào tập tin mẫu đã được lưu ( \*. gks ) và thêm nó vào " Template ".

Export...

Lưu khuôn mẫu được chọn trong " Khuôn mẫu" như tập tin mẫu ( \*. gks ).

Phụ lục 8 Chỉ mục

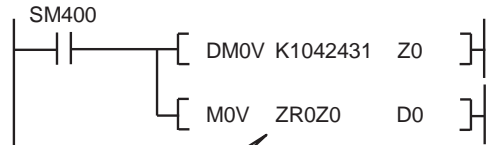
Với QCPU Universal (không bao gồm Q00UJCPU), mở rộng thanh ghi chỉ số đến 32 bit cho phép việc lập chỉ số cho tất cả thanh ghi vùng nhớ file.

số se-ri truy cập thanh ghi định dạng tập tin

ZR0
ZR1
⋮
⋮
ZR32767
ZR32768
⋮
⋮
⋮
ZR4184063

Conventional area to which indexing can be used

Area to which indexing can be used of Universal model QCPU



Chỉ số số thứ tự truy cập thanh ghi định dạng file ( ZR ) với 32 -bit, sử dụng thanh ghi chỉ số ( Z )

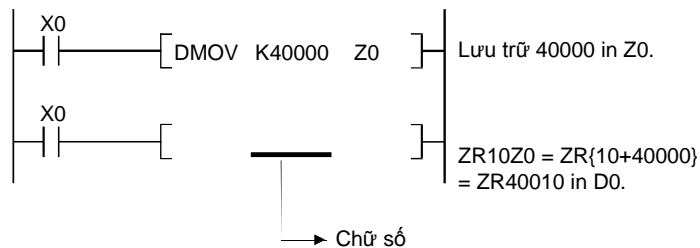
Phương pháp để xác định thanh ghi chỉ mục 32 bit có thể được chọn lựa theo hai phương pháp sau:

- Xác định phạm vi chỉ số được dùng chỉ 32 bit.
- Xác định chỉ mục 32 bit sử dụng " ZZ" .

(1) Khi xác định phạm vi số được dùng cho chỉ số 32 bit

(a) Mỗi thanh ghi chỉ số có thể thiết lập giữa - 2147483648 và 2147483647.

Sau đây cho thấy ví dụ chỉ số.



(b) Phương pháp kỹ thuật

Cho chỉ số với thanh ghi chỉ số 32 bit, xác định số bắt đầu của thanh ghi chỉ số để có thể sử dụng trên các tab Device của màn hình cài đặt thông số PLC trong GX Works2.

<b>GỢI Ý</b>
<p>Khi số bắt đầu của thanh ghi chỉ số đã được sử dụng được thay đổi trên tab Device của màn hình cài đặt thông số PLC, không thay đổi hoặc không viết các thông số vào bộ điều khiển lập khả trình, Chắc chắn chương trình sẽ ghi tham biến và bộ điều khiển lập trình được.</p> <p>Khi tham biến bị buộc phải ghi vào bộ điều khiển khả trình, xảy ra. LỖI CAN'T EXE. PRG. ( Mã lỗi : 2500 )</p>



- (c) Biến địa chỉ mà chỉ mục có thể được sử dụng  
Chỉ số có thể chỉ được sử dụng cho các biến địa chỉ hiển thị dưới đây.
- ZR : số sê – ri của thanh ghi tập tin định dạng truy cập
  - D: Mở rộng dữ liệu thanh ghi
  - W: Mở rộng đường dẫn thanh ghi

(d) Phạm vi Sử dụng của thanh ghi chỉ mục

Bảng dưới đây cho thấy phạm vi sử dụng của các thanh ghi chỉ số cho lập chỉ số với các chỉ số thanh ghi 32-bit.

Đối với chỉ số của thanh ghi chỉ số 32 bit, thanh ghi chỉ số này xác định ( Zn ) và thanh ghi chỉ số tiếp theo của thanh ghi xác định ( Zn + 1 ) được sử dụng. Chắc chắn thanh ghi chỉ số này không chồng lấp thanh ghi chỉ số đã được sử dụng.

Giá trị thiết lập	Thanh ghi chỉ số đã sử dụng	Giá trị thiết lập	Thanh ghi chỉ số đã sử dụng
Z0	Z0, Z1	Z10	Z10, Z11
Z1	Z1, Z2	Z11	Z11, Z12
Z2	Z2, Z3	Z12	Z12, Z13
Z3	Z3, Z4	Z13	Z13, Z14
Z4	Z4, Z5	Z14	Z14, Z15
Z5	Z5, Z6	Z15	Z15, Z16
Z6	Z6, Z7	Z16	Z16, Z17
Z7	Z7, Z8	Z17	Z17, Z18
Z8	Z8, Z9	Z18	Z18, Z19
Z9	Z9, Z10	Z19	Không thể được xác định.

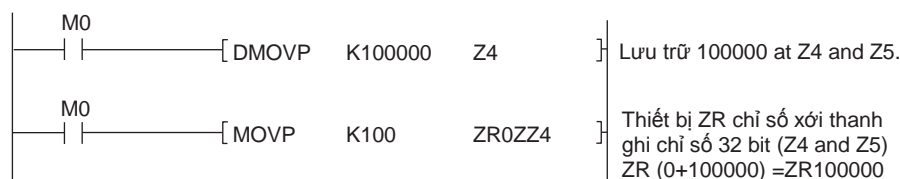
- (e) Ví dụ sau đây cho thấy chỉ số và quy trình thực tế. ( Khi Z0 ( 32 -bit ) là 100000 và Z2 ( 16 -bit ) là - 20 )

Ví dụ ladder	Thiết bị xử lý thực tế						
	<p>Mô tả</p> <table border="0"> <tr> <td>ZR1000Z0</td> <td>...</td> <td>ZR(1000+100000)=ZR101000</td> </tr> <tr> <td>D30Z2</td> <td>...</td> <td>D(30-20)=D10</td> </tr> </table>	ZR1000Z0	...	ZR(1000+100000)=ZR101000	D30Z2	...	D(30-20)=D10
ZR1000Z0	...	ZR(1000+100000)=ZR101000					
D30Z2	...	D(30-20)=D10					

(2) Khi xác định 32 bit sử dụng kỹ thuật " ZZ"

(a) Một chỉ mục thanh ghi có thể xác định 32 bit bằng cách sử dụng kỹ thuật " ZZ" như là " ZR0ZZ4".

Ví dụ sau đây cho thấy chỉ mục 32 bit với kỹ thuật " ZZ" .



(b) Phương pháp kỹ thuật

Chỉ số 32 bit sử dụng kỹ thuật " ZZ", chọn " Use ZZ" trong [ Chỉ số Thiết lập cho Thiết bị ZR ] trong tab Device trong màn hình thiết lập thông số PLC.

(c) Biến địa chỉ có thể sử dụng được chỉ số

Chỉ số có thể được sử dụng cho các biến địa chỉ hiển thị dưới đây

- ZR: số sê – ri của thanh ghi tập tin định dạng truy cập
- D: Mở rộng dữ liệu thanh ghi
- W: Mở rộng đường dẫn thanh ghi

(d) Phạm vi Sử dụng của thanh ghi chỉ mục

Bảng dưới đây cho thấy phạm vi sử dụng của các thanh ghi chỉ số cho lập chỉ số với các chỉ số thanh ghi 32-bit.

Đối với chỉ số của thanh ghi chỉ số 32 bit, thanh ghi chỉ số này xác định ( Zn ) và thanh ghi chỉ số tiếp theo của thanh ghi xác định ( Zn + 1 ) được sử dụng. Chắc chắn thanh ghi chỉ số này không chồng lấp thanh ghi chỉ số đã được sử dụng.

Kỹ thuật ZZ	Thanh ghi chỉ số đã sử dụng	" Kỹ thuật ZZ	Thanh ghi chỉ số đã sử dụng
<input type="checkbox"/> ZZ0	Z0, Z1	<input type="checkbox"/> ZZ10	Z10, Z11
<input type="checkbox"/> ZZ1	Z1, Z2	<input type="checkbox"/> ZZ11	Z11, Z12
<input type="checkbox"/> ZZ2	Z2, Z3	<input type="checkbox"/> ZZ12	Z12, Z13
<input type="checkbox"/> ZZ3	Z3, Z4	<input type="checkbox"/> ZZ13	Z13, Z14
<input type="checkbox"/> ZZ4	Z4, Z5	<input type="checkbox"/> ZZ14	Z14, Z15
<input type="checkbox"/> ZZ5	Z5, Z6	<input type="checkbox"/> ZZ15	Z15, Z16
<input type="checkbox"/> ZZ6	Z6, Z7	<input type="checkbox"/> ZZ16	Z16, Z17
<input type="checkbox"/> ZZ7	Z7, Z8	<input type="checkbox"/> ZZ17	Z17, Z18
<input type="checkbox"/> ZZ8	Z8, Z9	<input type="checkbox"/> ZZ18	Z18, Z19
<input type="checkbox"/> ZZ9	Z9, Z10	<input type="checkbox"/> ZZ19	Không xác định

\*:  chỉ ra t tên thiết bị (ZR, D, W) cho chỉ số đích

- (e) Ví dụ sau đây cho thấy việc lập chỉ số 32-bit với kỹ thuật "ZZ" và thiết bị xử lý thực tế.  
 Các chỉ số kỹ thuật 32-bit với kỹ thuật "ZZ" áp dụng cho các chức năng sau của GX Works2)

Ví dụ ladder	Thiết bị xử lý thực tế
	<p>Mô tả {        ...        ZR1000Z0 · ZR(1000+100000)=ZR101000        D30Z2      D(30-20)=D10</p>

- (f) Chức năng có sẵn cho kỹ thuật "ZZ"  
 Các chỉ số kỹ thuật 32-bit với kỹ thuật "ZZ" áp dụng cho các chức năng sau của GX Works2.

No.	Tên chức năng và mô tả
1	Xác định biến địa chỉ trong chương trình lệnh
2	Màn hình nhập
3	Thử nghiệm biến địa chỉ
4	Thử nghiệm biến địa chỉ với điều kiện
5	Thiết lập điều kiện màn hình
6	Tìm lỗi ( Điểm phát hiện ( xác định thiết bị ), thiết bị phát hiện

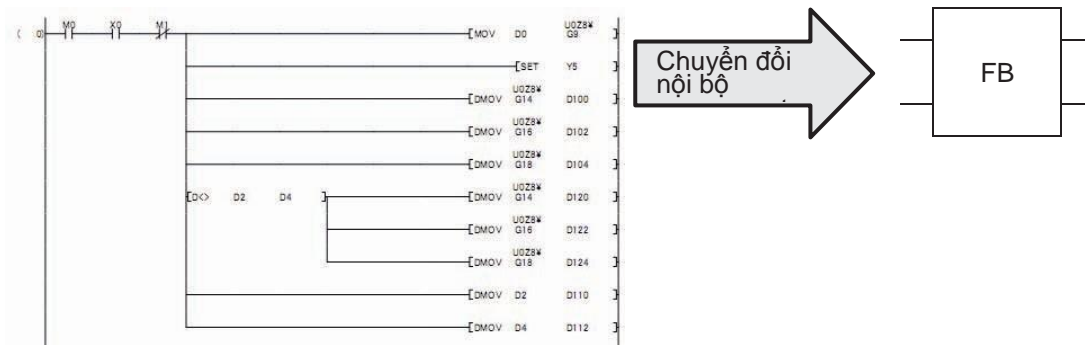
GỢI Ý
ZZn không thể được sử dụng đơn lẻ như là biến địa chỉ "DMOV K100000 ZZ0". Khi giá trị thiết lập của thanh ghi chỉ số xác định 32-bit với kỹ thuật "ZZ", thiết lập giá trị của Zn (Z0 đến Z19). Một mình ZZn không thể là đầu vào cho từng chức năng của GX Works2.

Để biết chi tiết, hãy tham khảo QnUCPU User's Manual Function Explanation, Program Fundamentals và MELSEC-Q/L Programming Manual (điều kiện lệnh).

Phụ lục 9 FB

Phụ lục 9.1 FB

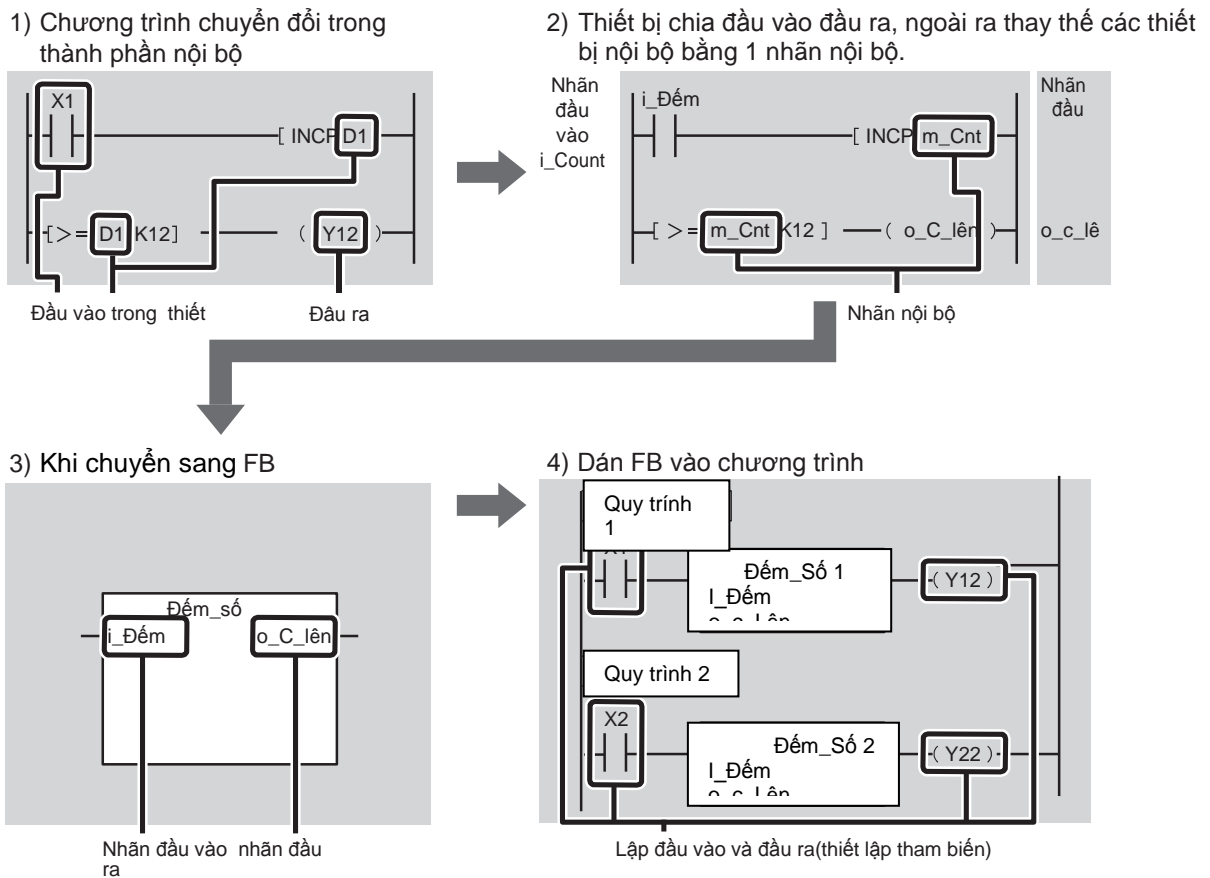
FB là một từ viết tắt cho một khối chức năng được thiết kế để chuyển đổi một khối ladder, được sử dụng nhiều lần trong một chương trình tuần tự, một thành phần (FB) được sử dụng trong một chương trình tuần tự. Điều này không chỉ làm tăng hiệu quả của các chương trình phát triển mà còn giảm lỗi lập trình để nâng cao chất lượng chương trình.



Hình vẽ Ứng dụng. 9.1 Chuyển đổi chương trình trình tuần tự thành phần .

Phụ lục 9.1.1 Sự chuyển đổi thành thành phần

Phần sau giải thích quy trình cho chuyển đổi chương trình đơn trong thành phần nội bộ.



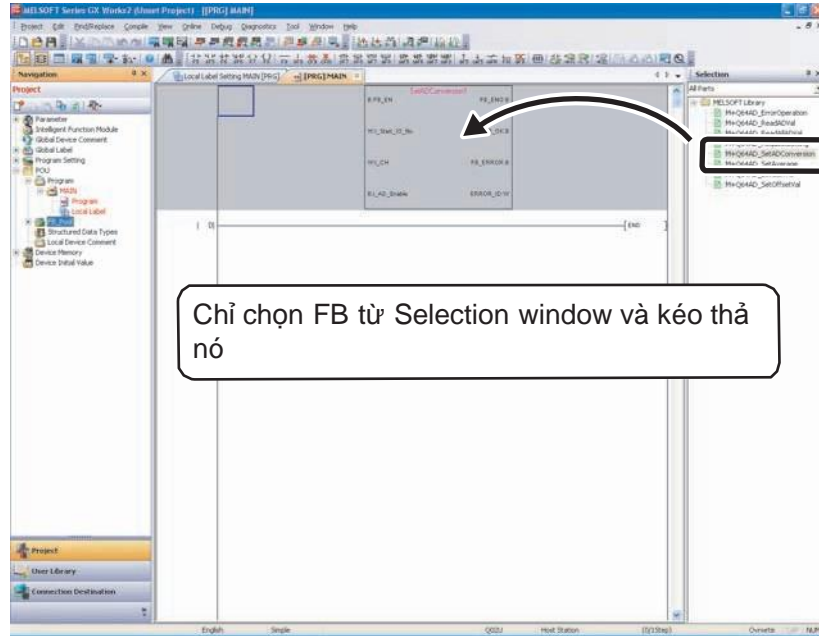
Hình vẽ Ứng dụng. 9.2 Luồng của sự chuyển đổi thành phần nội bộ.

Phụ lục 9.1.2 Ưu điểm của việc sử dụng FBS

Phần này giới thiệu những ưu điểm của việc tạo ra các chương trình bằng cách sử dụng FBS

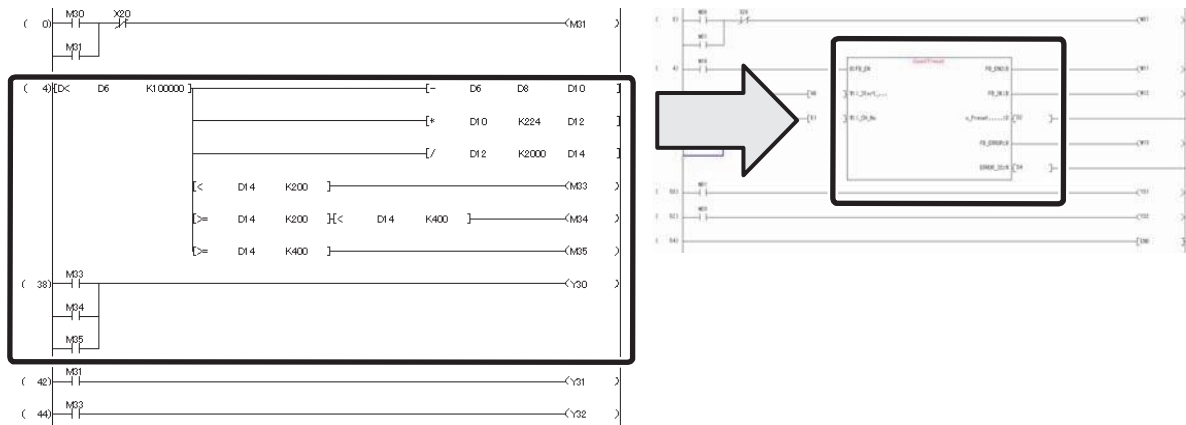
(2) Chương trình đơn giản

Chương trình trình tuần tự có thể đã được tạo lập bằng cách dán FBS. Điều này giảm đáng kể phát triển chương trình man-hours. ( Thư viện FB cung cấp bởi Công ty Mitsubishi Electric..)



( 2 ) Đọc đơn giản

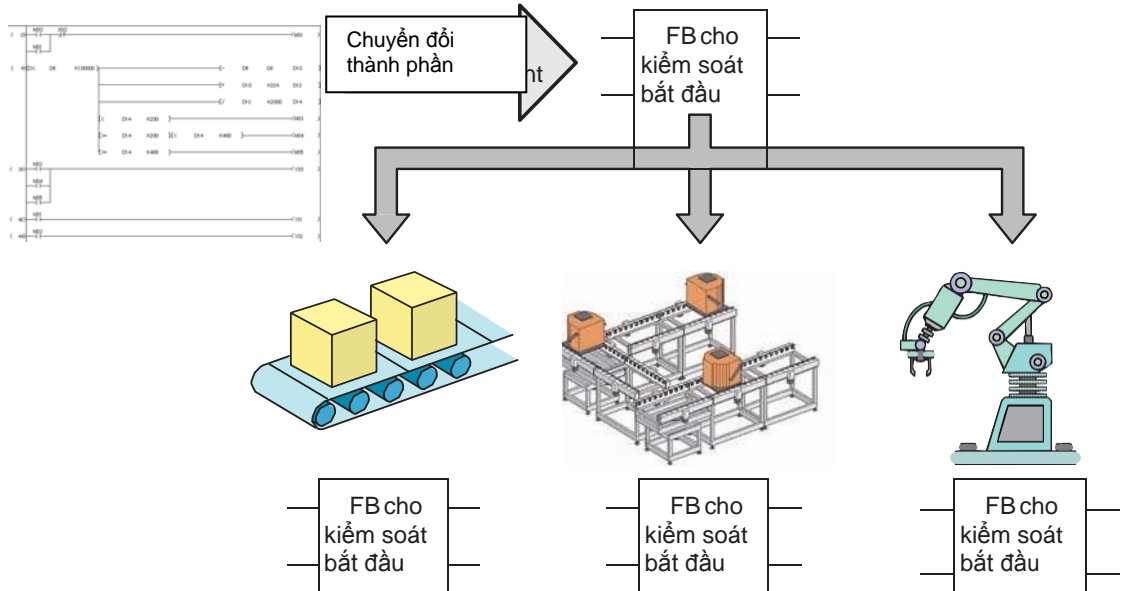
Sử dụng một FB tạo ra một chương trình đơn giản với chỉ một cái "box" (FB), một đầu vào, và một đầu ra để tạo ra một chương trình tuần tự dễ đọc.



### (3) Sử dụng lại

Chuyển đổi chương trình chuẩn trong các thành phần cho phép chương trình có thể được sử dụng lại nhiều lần.

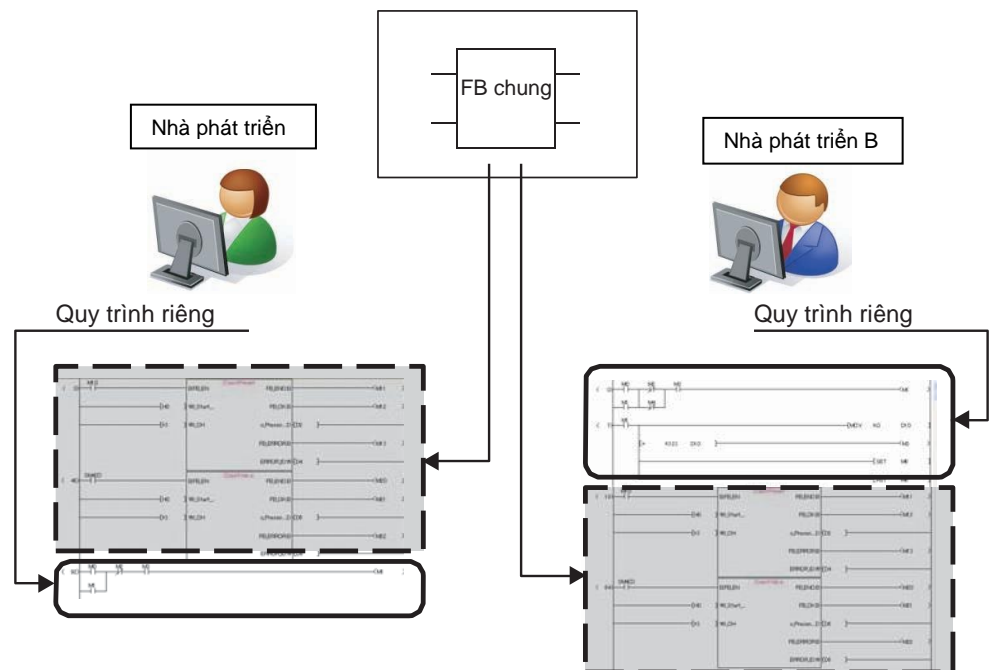
Kết quả là hoạt động sao chép chương trình trình tuần tự và điều chỉnh thiết bị, thường đã yêu cầu trước đây, sẽ không cần thiết.



### (4) Cải thiện chất lượng

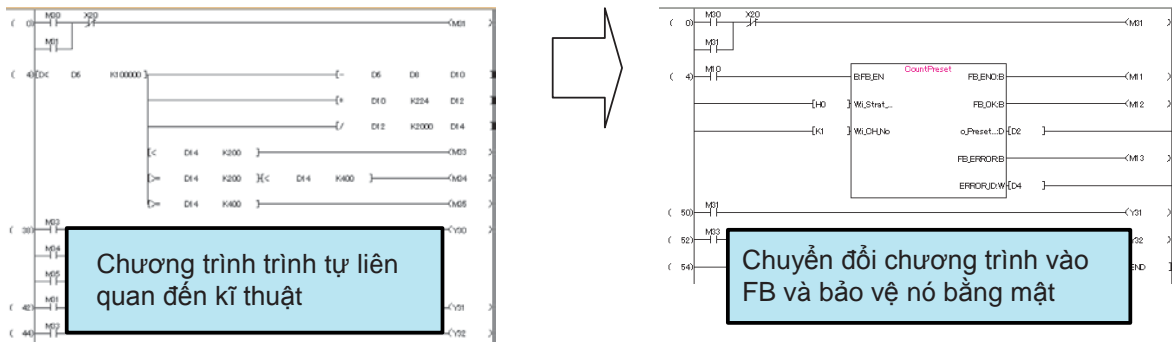
Chuyển đổi chương trình chuẩn vào thành phần như một FB để sử dụng lại chương trình cho phép phát triển chương trình với chất lượng không thay đổi, không dựa vào kỹ năng công nghệ của nhà phát triển chương trình.

Khi nhà phát triển A và B đang phát triển chương trình tuần tự cho thiết bị khác, sử dụng FB tương tự cho xử lý chung tạo cơ hội cho nhà phát triển tạo ra chất lượng không thay đổi của chương trình tuần tự.



(5) Tài sản được bảo vệ

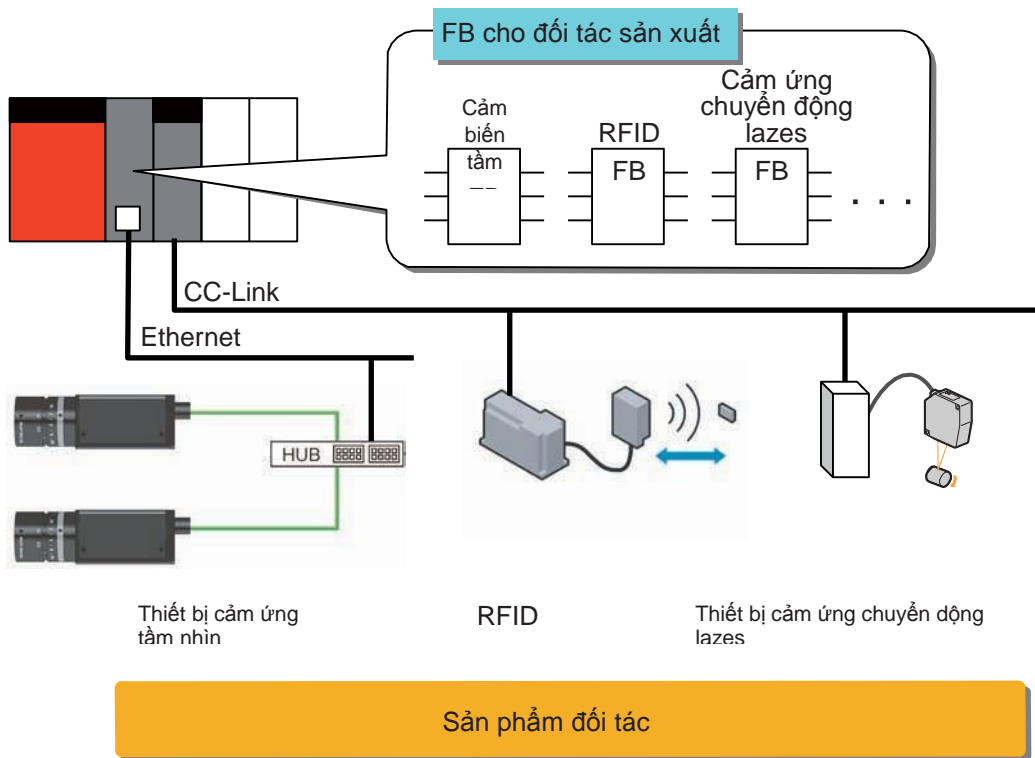
Do thiết lập mật khẩu khóa, FB tạo ra có thể được bảo vệ sao cho nó không thể xem được.



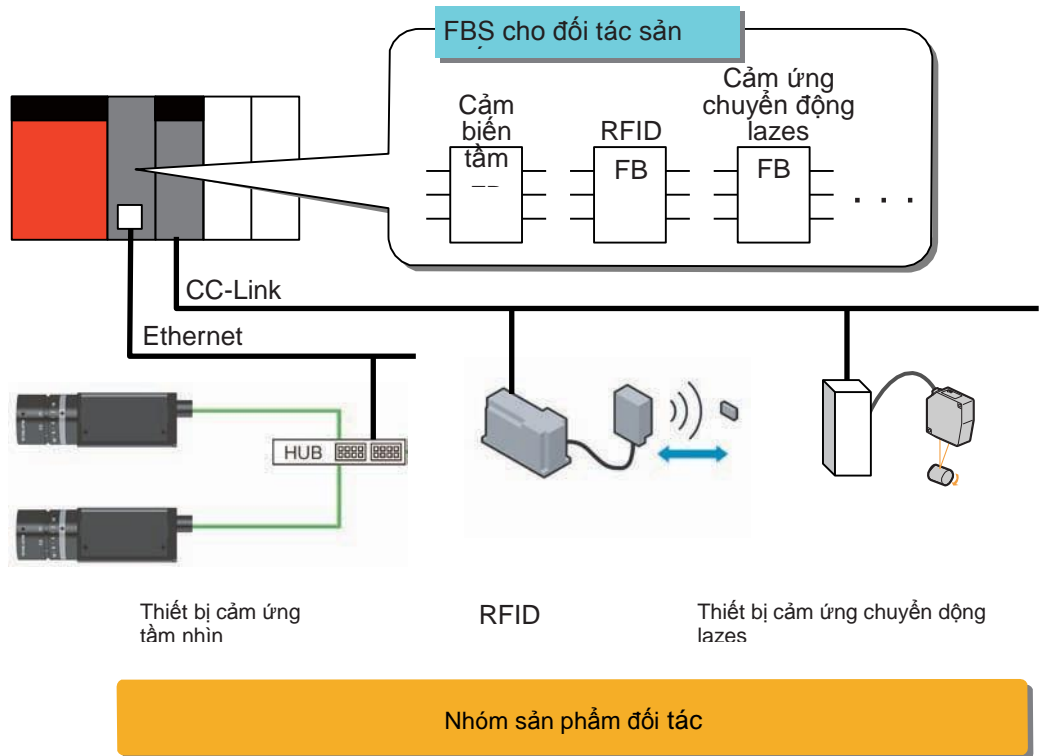
Phụ lục 9.1.3 Thư viện FB

Một thư viện FB là tập hợp các FBs đang sử dụng trong GX Works2 ( dự án Đơn giản ). Sử dụng thư viện FB cho phép dễ dàng sắp đặt và hoạt động của modules MELSEC-Q / L và sản phẩm đối tác.

<Ví dụ của module MELSEC-Q /L>



<Ví dụ sản phẩm của đối tác >



(1) Đội hình thư viện FB

Thư viện FB bao gồm " FBs cho MELSEC-Q / L module" và " FBs cho sản phẩm đối tác".

(2) Làm sao để có được thư viện FB

Thư viện FB có thể tải từ trang chủ của Mitsubishi Electric FA.

URL <http://www.mitsubishielectric.co.jp/fa/index.html>

Cho thủ tục để đạt được thư viện FB, tham khảo Ứng dụng. 9.2.2" Chuẩn bị trước khi dùng thư viện FB



#### Phụ lục 9.1.4 Công cụ phát triển

GX Works2 ( dự án Đơn giản ) ver 1.12 N hoặc sau đó phải phát triển chương trình trình tự sử dụng FBs.

GỢI Ý
-------

Tùy vào thư viện FB , hỗ trợ phiên bản GX Works2 có thể khác nhau. Đối với mỗi chi tiết nói về trang tải xuống của mỗi FB
---

#### Phụ lục 9.1.5 FB quy cách và biện pháp phòng ngừa

Quy cách và biện pháp phòng ngừa sau phải được hiểu rõ trước khi sử dụng FBs.

1. FB không thể được dùng trong FB khác.
2. Bởi vì một quy trình cụ thể được thêm vào FB khi một FB được bố trí, số lượng các bước tăng khi so sánh với một ladder được tạo ra mà không có một FB
3. FBs không thể được dùng trong chương trình gián đoạn.
4. FBS mà thực hiện không hoàn chỉnh trong vòng một lần quét không thể sử dụng trong các lệnh vòng lặp FOR đến NEXT theo hoặc các chương trình con.

## Phụ lục 9.2 Tạo chương trình bằng cách sử dụng thư viện FB

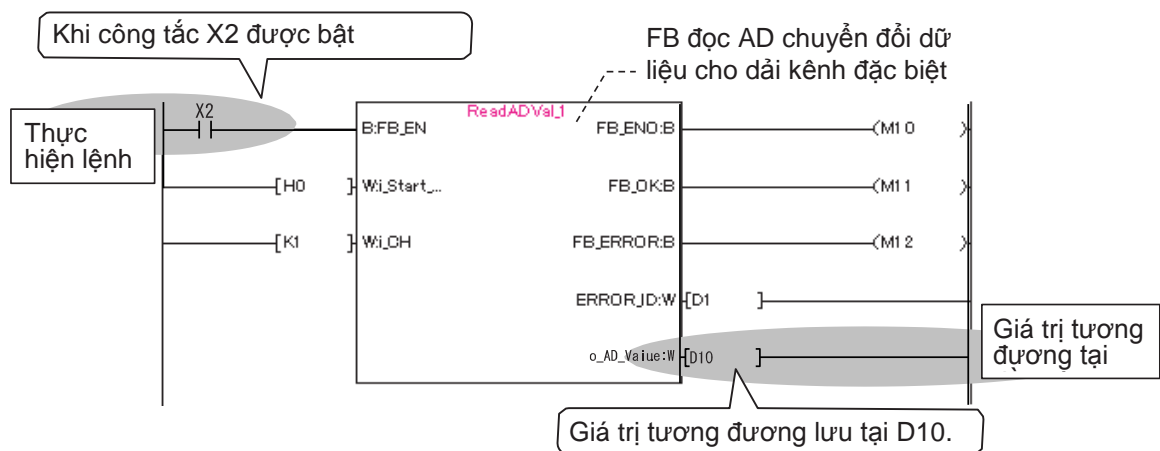
Phần này giải thích thủ tục cho tạo chương trình bằng cách sử dụng thư viện FB

### Phụ lục 9.2.1 Các chương trình được tạo ra

Phần này giải thích cách sử dụng thư viện FB với ví dụ nhập vào giá trị tương tự từ module đầu vào tương tự.

Ví dụ) Đọc giá trị tương tự đến D10 từ module đầu vào tương tự ( Q64AD ) khi công - tắc ( X2 ) được bật.

Chương trình có thể dễ dàng được tạo ra bằng cách sử dụng một thư viện FB như sau.



#### GỢI Ý

FB tạo bởi người dùng khác với FB trong thư viện FB.

Đối với các phương pháp sáng tạo của FB mới nhất, tham khảo "MELSOFT GX Works2 FB Quick Start Guide"

## Phụ lục 9.2.2 Chuẩn bị trước khi dùng thư viện FB

Trước khi dùng thư viện FB, liên lạc nhà phân phối của bạn để nhận được nó.  
(Thư viện FB sẽ không được cài đặt khi cài đặt GX Works2.

Theo giải thích thủ tục, hoạt động sử dụng thư viện FB cho Q64AD như một kiểu mẫu..

- 1) Khi các tập tin thu được từ các nhà phân phối của bạn là một tập tin định dạng zip, giải nén "q64ad\_v100a.zip".



- 2) Nhấp đúp "setup.exe" trong "q64ad v100a".

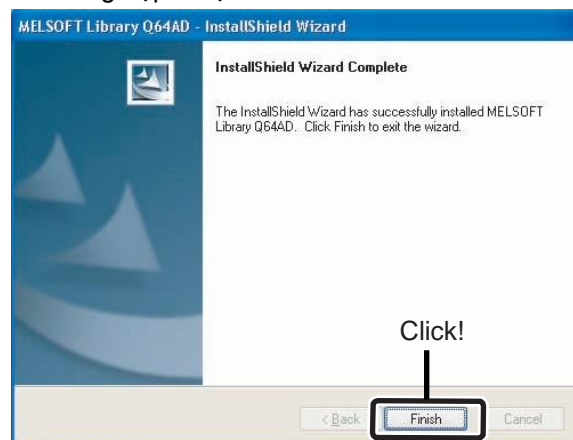


- 3) Màn hình cho cài đặt được hiển thị. Làm theo lời chỉ dẫn để hoàn tất cài đặt.



- 4) Hộp thoại sau đây được hiển thị khi cài đặt hoàn tất. Nhấp vào nút

để đóng hộp thoại

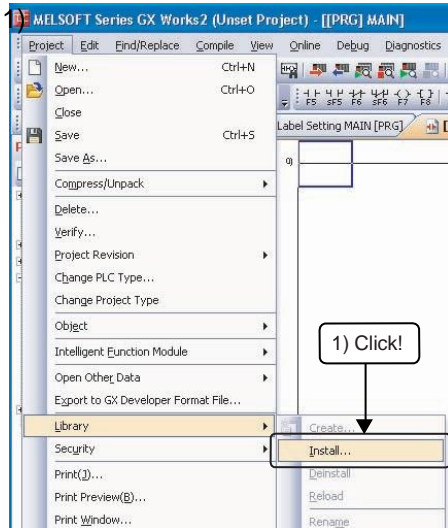


Hoàn thành việc chuẩn bị trước khi sử dụng các thư viện FB.

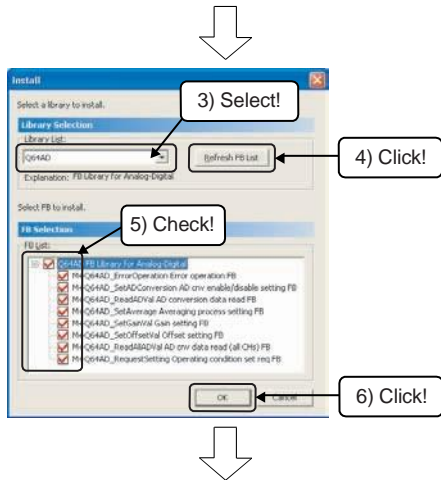
Phụ lục 9.2.3 Nhập vào thư viện FB dự án

Phần này giải thích làm thế nào để nhập vào một thư viện FB cho module đầu vào analog (Q64AD) để được dán vào chương trình vào một dự án.

Tạo dự án mới trước các hoạt động sau. (Hãy tham khảo phần 2.3.2)



1) Click [Dự án] → [Library] → [Install].



2) Hộp thoại Lắp đặt được hiển thị.

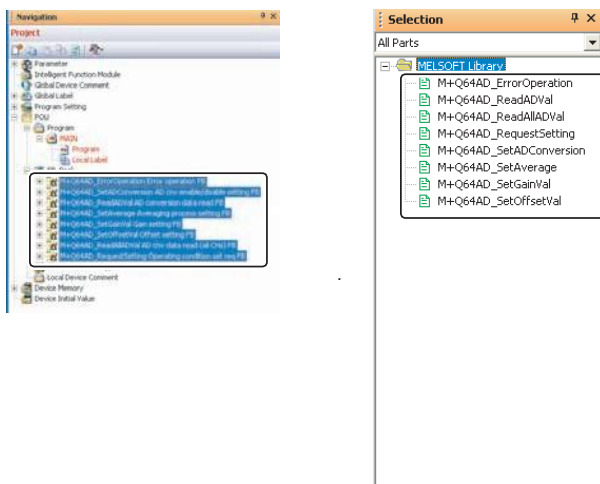
3) Chọn " Q64AD" từ Thư viện Danh mục.

4) Nhấp vào **Refresh FB List**

5) Kiểm tra thư viện để nhập khẩu.

6) Nhấp vào **OK**

7) Các FBS nhập vào được trình bày dưới FB\_Pool trong khung nhìn Dự án và hiển thị trong cửa sổ lựa chọn.

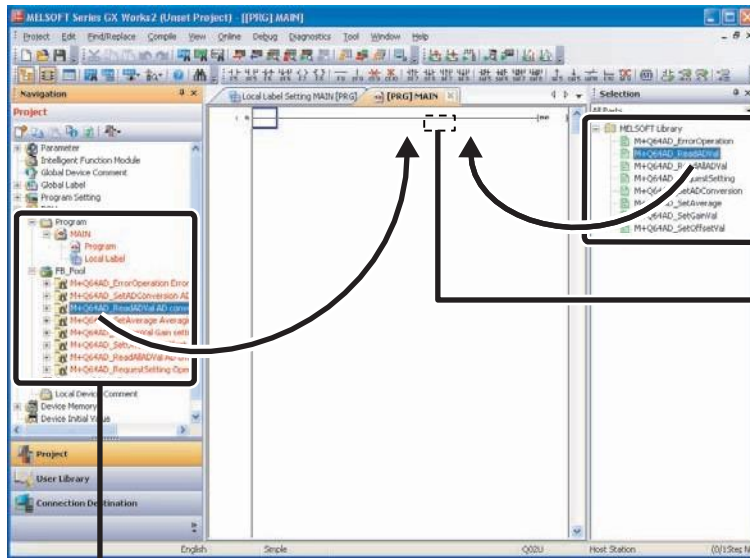


## Phụ lục 9.2.4 Dán FBS

Kéo và thả FBs để được dán đến cửa sổ chương trình từ góc nhìn Dự án hay cửa sổ Lựa chọn. ( Kéo và thả từ góc nhìn Dự án là có thể từ GX Works2 1.24A hoặc sau đó. )

Quy trình nghiệp vụ

- 1) Dán " M + Q64AD ReadADVal" đến cửa sổ chương trình.



Cửa sổ lựa chọn

Từ cửa sổ lựa chọn hoặc xem dự án, kéo và thả một FB đến nơi FB sẽ được dán.

Cửa sổ Dự án



- 2) Không kết nối được với server. Bạn có thể thử lại sau vài giây



Để biết thêm chi tiết của thiết lập, tham khảo Đến trang sau.

### Phụ lục 9.2.5 Cài đặt tên của dán FBs

Khi thư viện FB được dán đến cửa sổ chương trình, hộp thoại tên đầu vào của FBs được dán( tên đối tượng FB ) được hiển thị.

Tên đối tượng là một tên để phân biệt các FB.

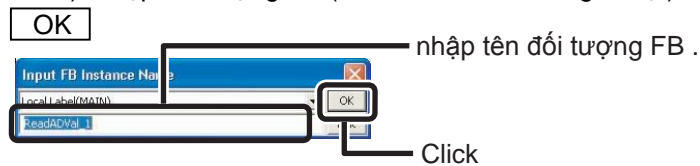
Tên tạm thời được tự động ấn định thành tên giải pháp. Để sử dụng tên, đóng hộp thoại bằng bấm

Bảo đảm tên cũng không tồn tại trong chương trình cùng lúc khi thay đổi tên.

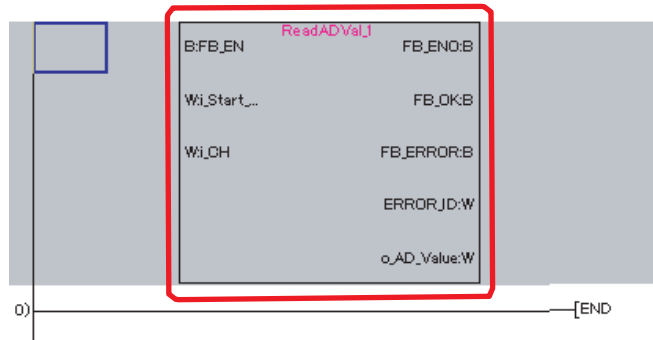
Ở phần này, mặc định được sử dụng.

Quy trình nghiệp vụ

1) Nhập tên tượng FB (" ReadADVal 1" trong ví dụ ) và bấm nút .



2) Các FB được dán vào cửa sổ chương trình.



#### GỢI Ý

Khi nhập vào tên đối tượng, lưu ý các điểm sau.

- Trường hợp nhạy cảm
- Chữ cái đầu tiên không thể là số.
- Số lượng ký tự tối đa cho tên đối tượng là 16.

Lỗi xảy ra nếu click OK với các thiết lập sau.

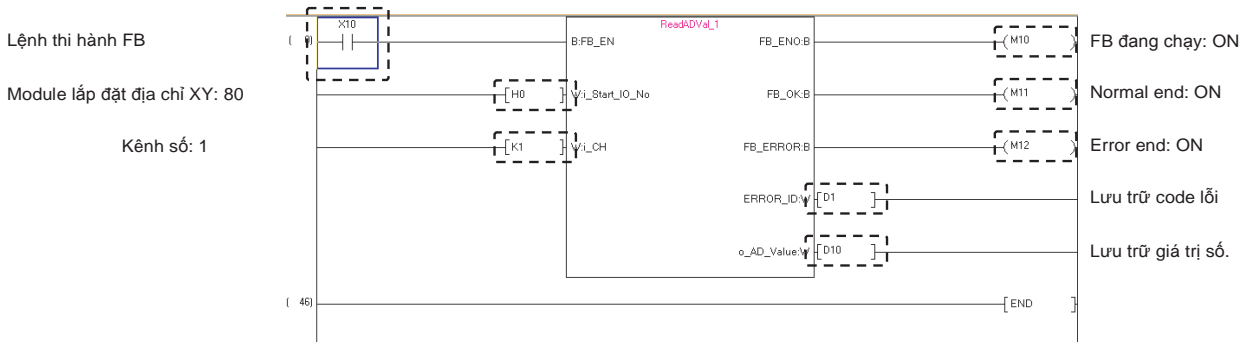
(Khi viết chữ cái đầu tiên là số).



**Phụ lục 9.2.6 Tạo ra đầu vào và đầu ra ladder**

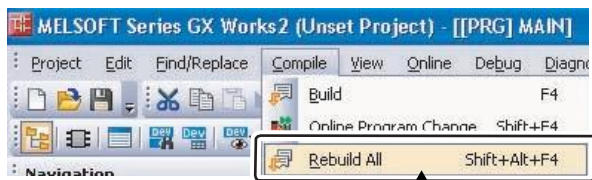
Tạo ra phần ladder đầu vào và phần ladder đầu ra của FB dán để hoàn chỉnh chương trình.

Tham khảo hình vẽ sau và nhập thông tin.



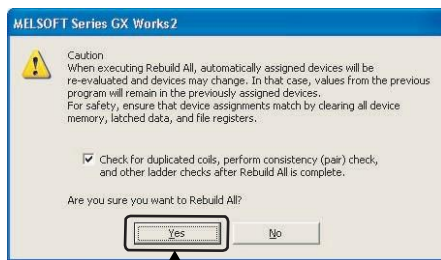
**Phụ lục 9.2.7 Thực hiện sự chuyển đổi / biên dịch**

Sự chuyển đổi / biên dịch phải được thi hành thành chương trình hoàn chỉnh. Sau đây giải thích làm thế nào để chuyển đổi / biên dịch tất cả các chương trình.



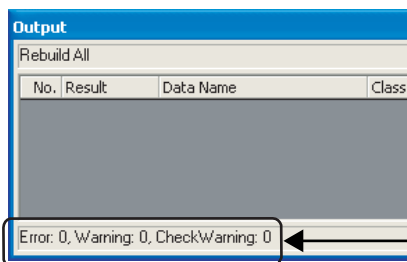
1) Nhấp vào [Compile] → [Rebuild All].

1) Click!



2) Thông báo bên trái được hiển thị. Click nút **Yes**.

2) Click!



3) Kết quả biên dịch được hiển thị trong cửa sổ đầu ra

3)Hiển thị!

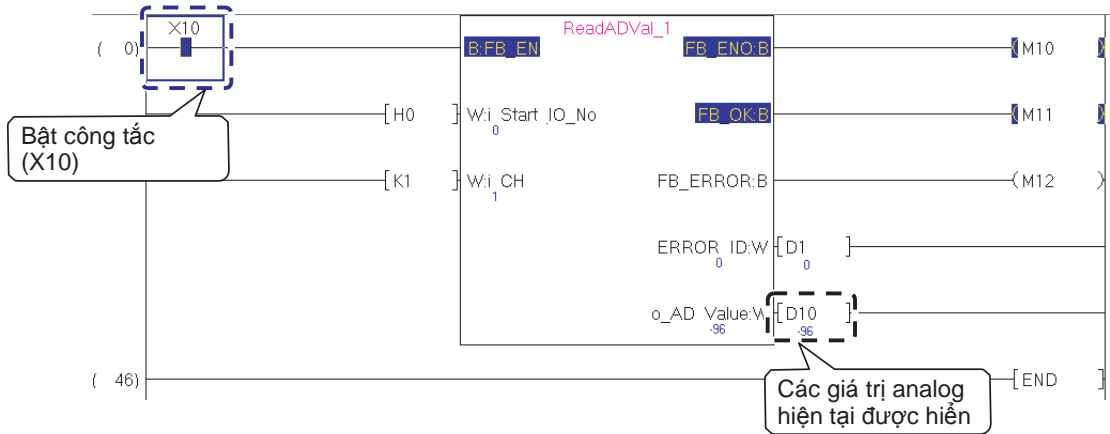
Phụ lục 9.2.8 Ghi chương trình tuần tự

Đối với các thủ tục để ghi chương trình tuần tự, tham khảo phần 2.7 ( 1 )" Ghi dữ liệu đến CPU".

Phụ lục 9.2.9 Thao tác kiểm tra

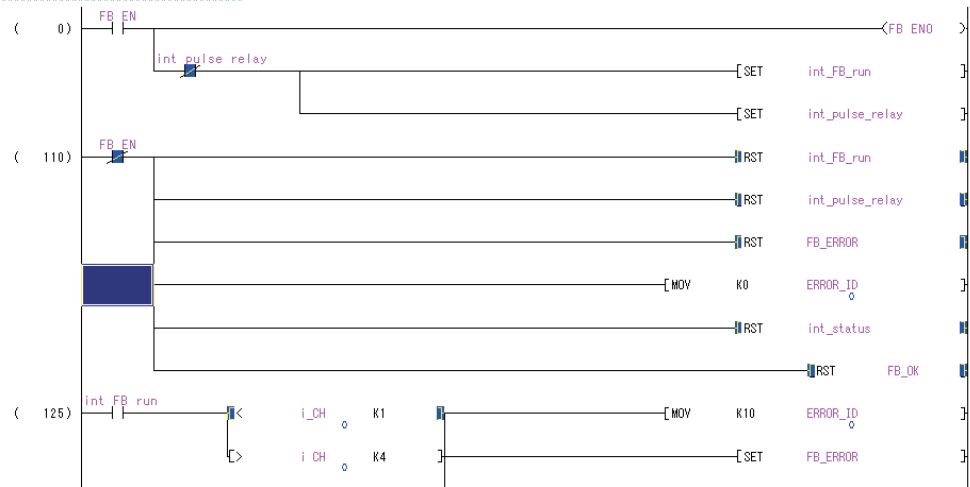
Đối với các thủ tục để kiểm tra thao tác của chương trình đã tạo ra, tham khảo phần 2.8 Giám sát Status Chương trình Ladder.

Bật công - tắc ( X2 ) và xác nhận giá trị tương tự được đọc.



Nhấp đúp chuột FB trong chương trình tuần tự trên màn hình cho phép giám sát của trạng thái chương trình tuần tự trong FB.

\*\*\*\*\*  
 \* FB name: M+Q64AD\_ReadADVal  
 \* Function: AD conversion data read  
 \* Version: Ver1.01B  
 \*\*\*\*\*







# Tài liệu hướng dẫn khối điều khiển khả trình Mitsubishi

## Khóa học cơ bản Q-series (cho GX Works2)

MODEL	SCHOOL-BASIC-GXW2-E
MODEL CODE	13JW55
SH(NA)-081123ENG-A(1210)MEE	



HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN  
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.