PLC

Ứng dụng lập trình (Bản vẽ nguyên lý mạch điện/Sê-ri MELSEC iQ-R)

Khóa học này nhằm giúp người dùng hiểu được kiến thức cơ bản về bộ điều khiển khả trình sê-ri MELSEC iQ-R và muốn tìm hiểu về bước lập trình tiếp theo.

L(CTS)00688VIE

Giới thiệu Mục đích khóa học

Khóa học này nhằm giúp người dùng đã hoàn tất Kiến thức cơ bản lập trình (Bản vẽ nguyên lý mạch điện) hoặc người có kiến thức tương đương. Khóa học này cung cấp kiến thức về việc lập trình hiệu quả và gỡ rối cho bộ điều khiển khả trình sê-ri MELSEC iQ-R, sử dụng nâng cao thiết bị và lập trình nhãn.

Để được tham gia khóa học này, bạn phải hoàn thành các khóa học sau đây hoặc có kiến thức tương đương.

- Kiến thức cơ bản về Sê-ri MELSEC iQ-R
- Cơ bản về lập trình

Giới thiệu Cấu trúc khóa học

Nội dung của khóa học này như sau.

Chương 1 - Lập trình hiệu quả

Phương pháp và thiết lập để lập trình hiệu quả

Chương 2 - Lập trình nâng cao

Sử dụng nâng cao các thiết bị và lập trình nhãn

Chương 3 - Gỡ rối hiệu quả

Các chức năng của phần mềm kỹ thuật được sử dụng để gỡ rối hiệu quả

Bài kiểm tra cuối khóa

Điểm đạt: Bắt buộc phải đúng từ 60% trở lên

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

Biện pháp phòng ngừa an toàn

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

Biện pháp phòng ngừa trong khóa học này

Màn hình hiển thị của phiên bản phần mềm mà bạn sử dụng có thể khác với các màn hình trong khóa học này. Khóa học này sử dụng phiên bản phần mềm sau:

- GX Works3 Phiên bản 1.044W

Chương 1 Lập trình hiệu quả

Chương này mô tả thiết lập của phần mềm kỹ thuật để lập trình hiệu quả và kiến thức cơ bản về lập trình nhãn.

1.1 Sử dụng dễ dàng hơn các chương trình trong các hệ thống khác nhau

1.2 Điều chỉnh vùng nhớ theo như hiện trạng sử dụng thiết bị

1.3 Sử dụng các tên nhãn có liên quan đến các ứng dụng

1.4 Cải thiện khả năng đọc chương trình

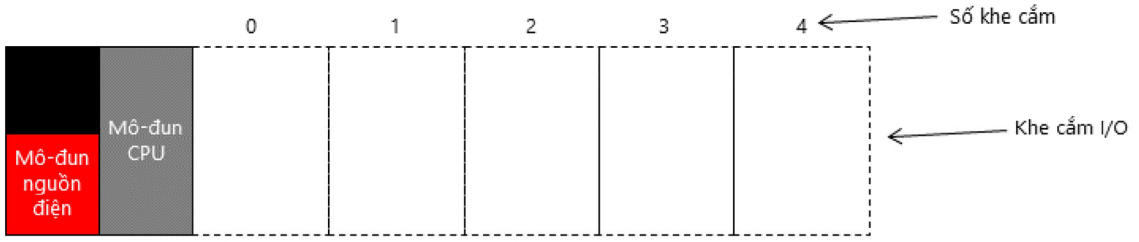
1.1 Sử dụng dễ dàng hơn các chương trình trong các hệ thống khác nhau

Phần này mô tả sự phân giao số thứ tự I/O hiệu quả để sử dụng sớm hơn các chương trình trong các hệ thống khác nhau.

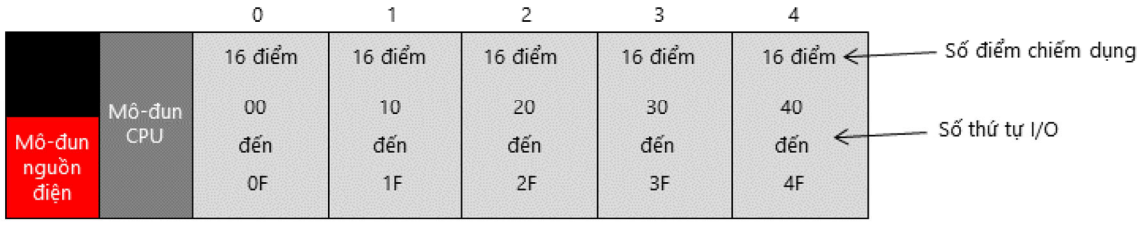
1.1.1 Phân giao số thứ tự I/O tự động

Số thứ tự I/O được chỉ định theo trình tự cho các mô-đun được lắp đặt trên thanh gắn mô đun, bắt đầu từ khe cắm gần mô-đun CPU nhất.
Số thứ tự I/O được chỉ định trong các đơn vị 16 điểm (0 đến F).

Số lượng điểm có sẵn được chỉ định (chiếm) thay đổi tùy thuộc vào loại mô-đun (16, 32, 64, v.v.).



Trong ví dụ sau, năm mô-đun 16 điểm được cài đặt.



1.1.1 Phân giao số thứ tự I/O tự động

Khi các mô-đun có 16, 32 và 64 điểm được chỉ định được sử dụng đồng thời, số thứ tự I/O được chỉ định như sau:

		0	1	2	3	4
Mô-đun nguồn điện	Mô-đun CPU	16 điểm 00 đến 0F	32 điểm 10 đến 2F	64 điểm 30 đến 6F	32 điểm 70 đến 8F	16 điểm 90 đến 9F

Nếu có khe cắm trống giữa các mô-đun được cài đặt, số thứ tự I/O cũng được chỉ định cho khe cắm trống đó.

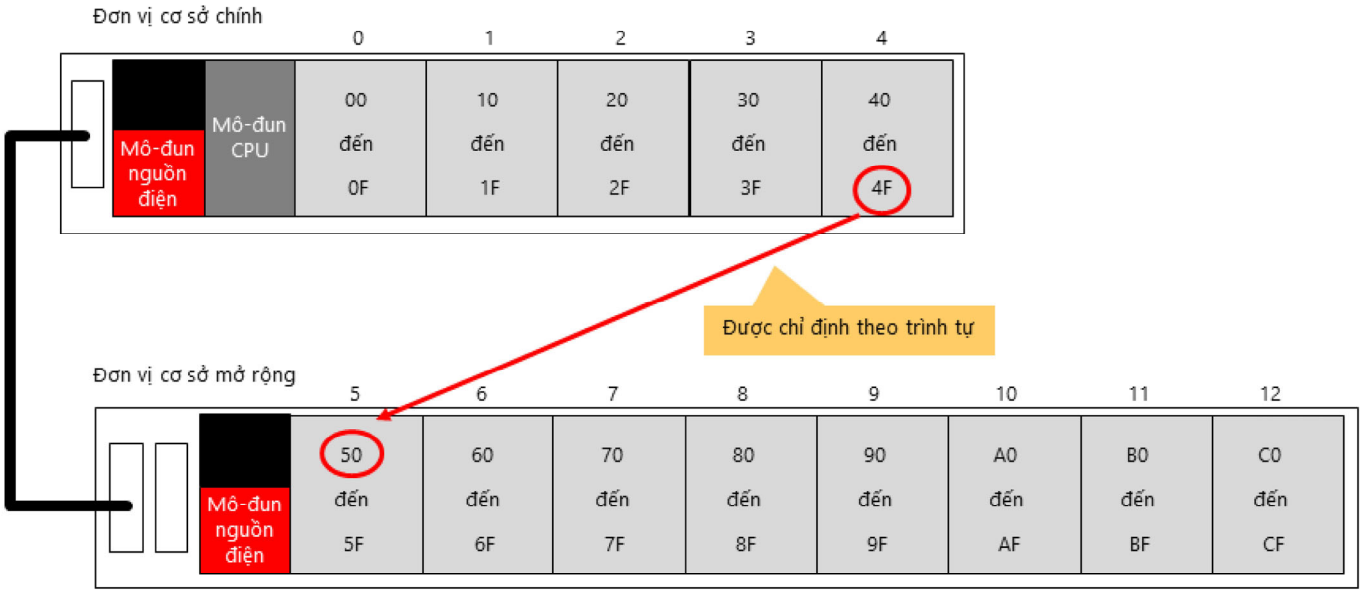
		0	1	2	3	4
Mô-đun nguồn điện	Mô-đun CPU	16 điểm 00 đến 0F	32 điểm 10 đến 2F	64 điểm 30 đến 6F	16 điểm 70 đến 7F	16 điểm 80 đến 8F

Khe cắm trống

Theo mặc định, 16 điểm được chỉ định cho một khe cắm trống. Số lượng điểm được chỉ định có thể được thay đổi theo thiết lập thông số trong phạm vi từ 0 đến 1024 điểm (trong đơn vị 16 điểm).

1.1.1 Phân giao số thứ tự I/O tự động

Số thứ tự I/O của đơn vị cơ sở mở rộng được chỉ định tự động, sau số thứ tự I/O cuối cùng của thanh gắn mô đun chính.

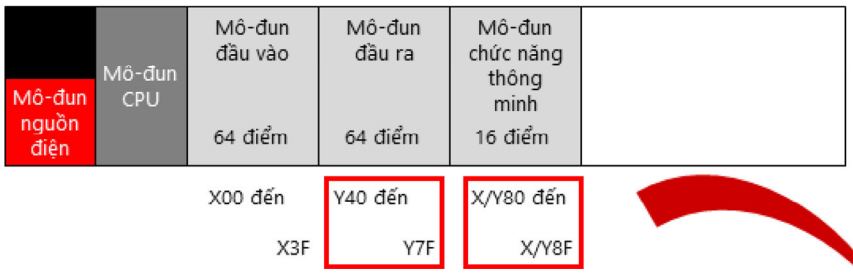


1.1.2 Phân chia số thứ tự I/O cố định

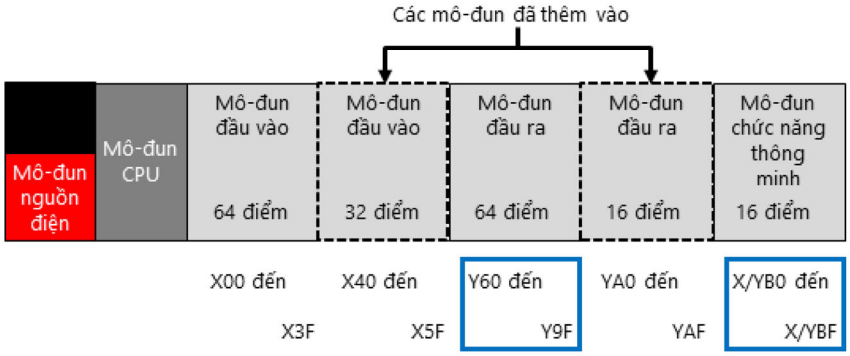
Khi số thứ tự I/O được chỉ định theo cách thủ công, số thứ tự I/O được chỉ định cố định và không thay đổi ngay cả khi thay đổi cấu hình mô-đun. Điều này có nghĩa là có thể sử dụng cùng chương trình điều khiển cho cùng một chức năng điều khiển bất kể cấu hình mô-đun là gì.

Phân giao tự động

Trước khi các mô-đun được thêm vào



Sau khi các mô-đun được thêm vào
(Một mô-đun đầu vào 32 điểm và một mô-đun đầu ra 16 điểm được thêm vào.)

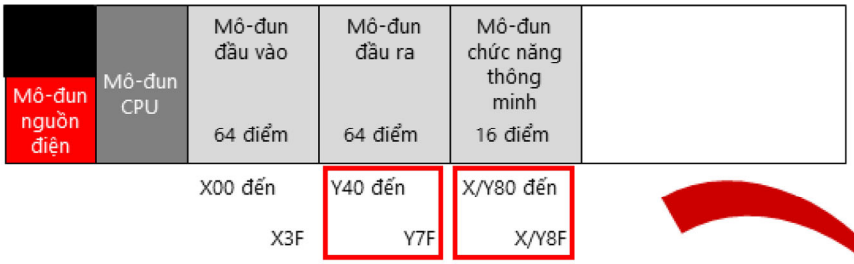


Số thứ tự I/O được chỉ định lại sau khi thêm vào các mô-đun.

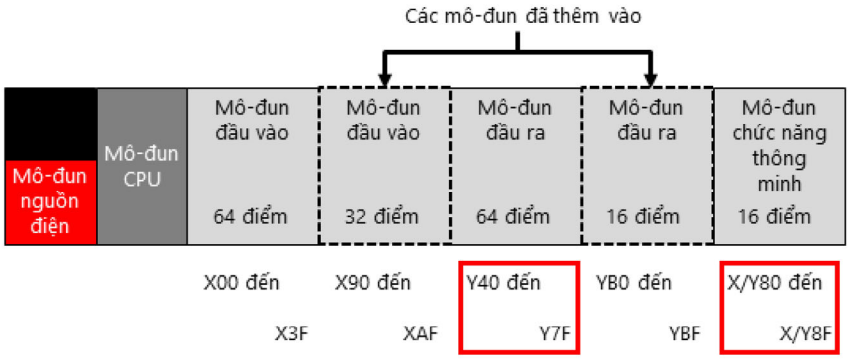
1.1.2 Phân chia số thứ tự I/O cố định

Phân chia thủ công cho các số thứ tự I/O cố định

Trước khi các mô-đun được thêm vào



Sau khi các mô-đun được thêm vào
(Một mô-đun đầu vào 32 điểm và một mô-đun đầu ra 16 điểm được thêm vào.)

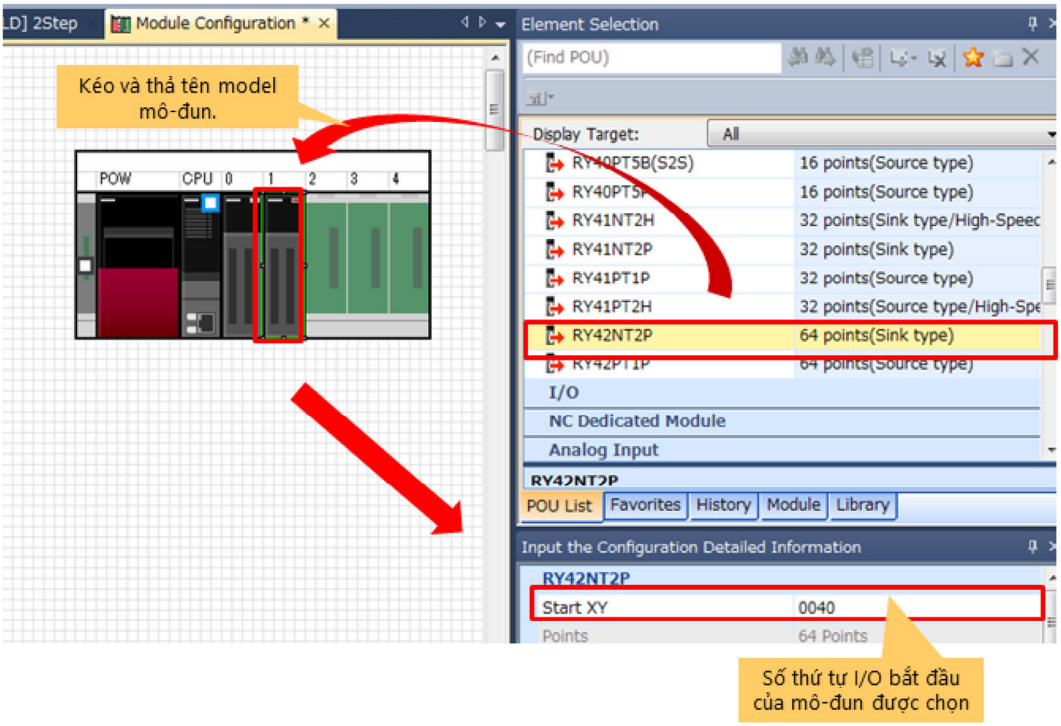


Các số thứ tự I/O tương tự được chỉ định bất kể việc thêm vào các mô-đun.

Vì số thứ tự I/O của mô-đun hiện tại không đổi, chỉ các chương trình có liên quan đến các mô-đun đã thêm vào mới cần được thêm vào hoặc sửa đổi.

1.1.3 Phân chia số thứ tự I/O tự động bằng sơ đồ cấu hình mô-đun

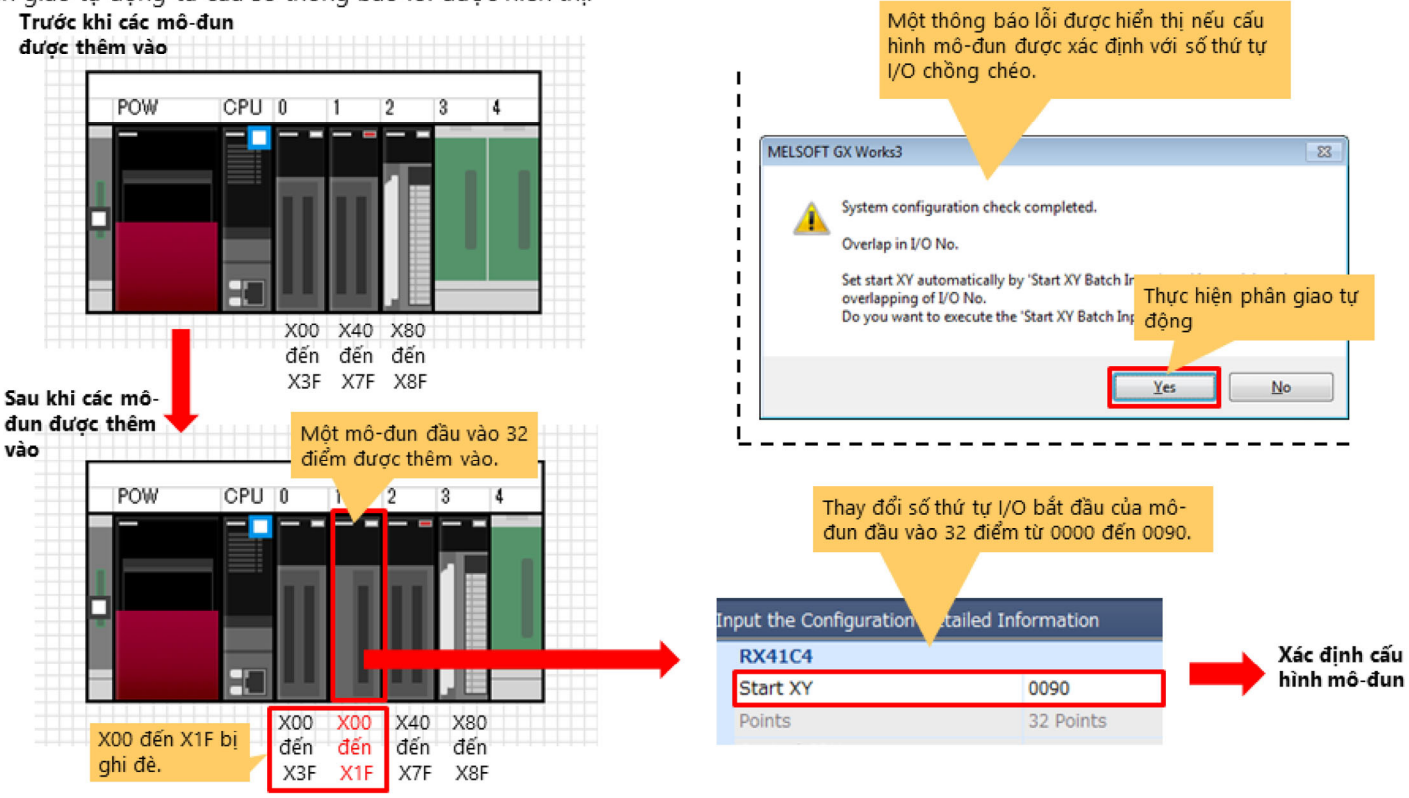
Có thể thiết lập cấu hình mô-đun bằng sơ đồ cấu hình mô-đun của phần mềm kỹ thuật, MELSOFT GX Works3.
Chọn tên model mô-đun, kéo và thả nó vào một vị trí để đặt mô-đun tại đó.
Số thứ tự I/O được chỉ định theo trình tự vào các mô-đun, bắt đầu từ mô-đun gần mô-đun CPU nhất như được hiển thị trong sơ đồ.
Chọn một mô-đun đã đặt để xem số thứ tự I/O bắt đầu của mô-đun.



1.1.4 Phân giao số thứ tự I/O thủ công bằng sơ đồ cấu hình mô-đun

Ví dụ sau mô tả cách chỉ định số thứ tự I/O theo cách thủ công bằng sơ đồ cấu hình mô-đun của GX Works3.

Khi cấu hình mô-đun được thay đổi bằng cách thêm một mô-đun mới vào sơ đồ cấu hình mô-đun, số thứ tự I/O của mô-đun đã thêm vào đè lên số hiện tại nếu vị trí của mô-đun hiện tại được thay đổi. Chỉnh sửa số thứ tự I/O để chúng không bị ghi đè và xác định cấu hình mô-đun.
Một thông báo lỗi được hiển thị nếu cấu hình mô-đun được xác định với số thứ tự I/O chồng chéo. Lúc này, có thể thực hiện phân giao tự động từ cửa sổ thông báo lỗi được hiển thị.



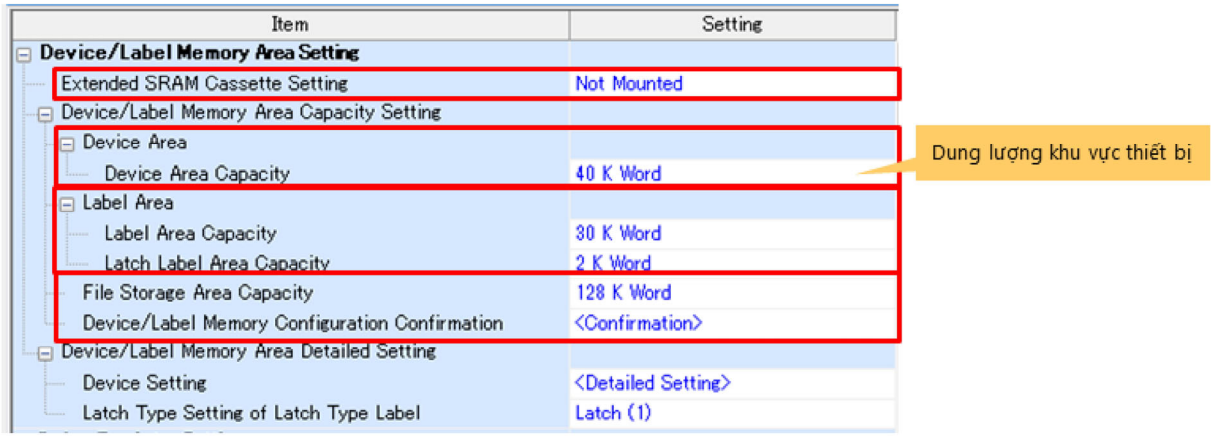
1.2 Điều chỉnh vùng nhớ theo như hiện trạng sử dụng thiết bị

1.2.1 Thiết lập vùng nhớ của thiết bị/nhãn

Số điểm của thiết bị được sử dụng cho mô-đun CPU thay đổi theo loại mô-đun CPU. Số điểm ban đầu của thiết bị được chỉ định dựa trên dung lượng khu vực thiết bị của mô-đun CPU.

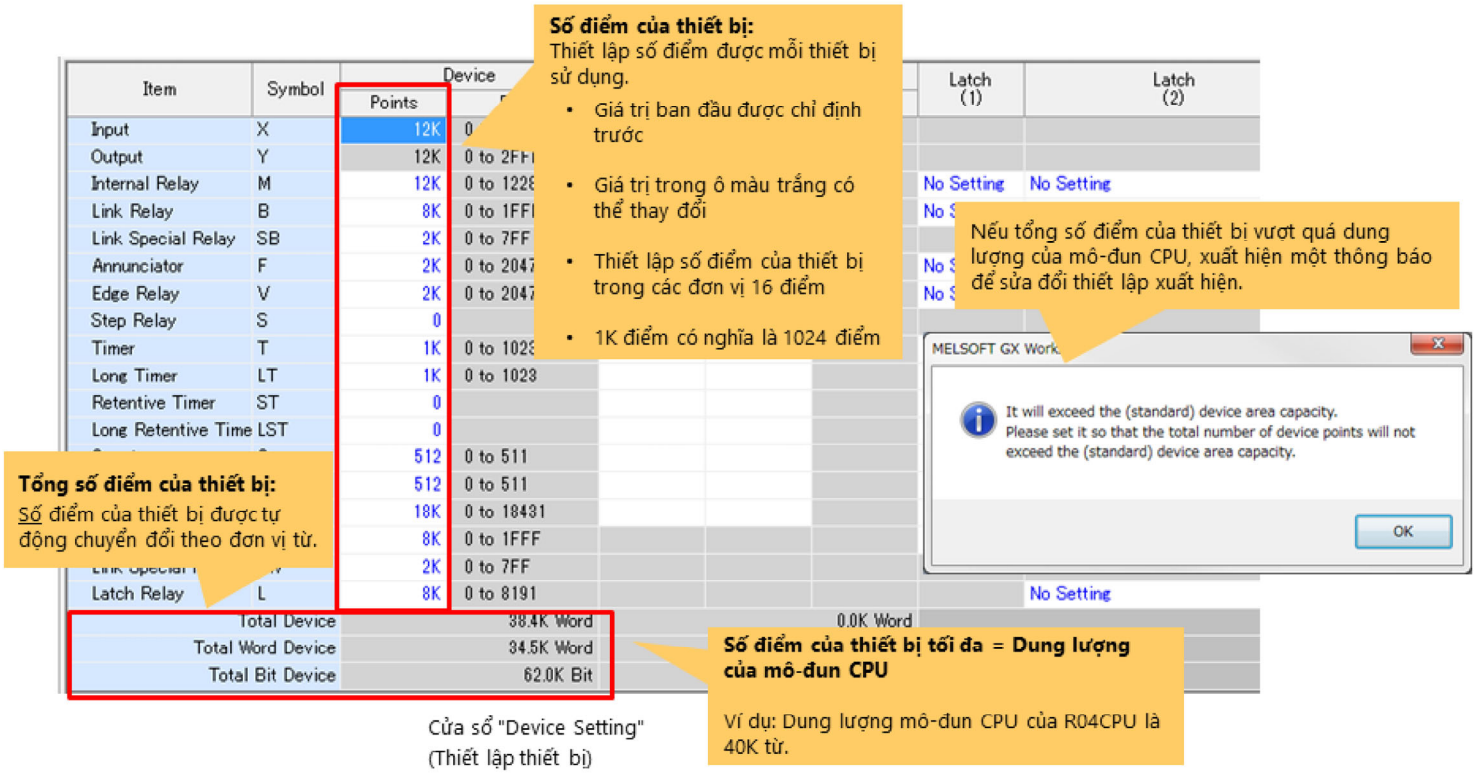
Dung lượng của khu vực thiết bị của R04CPU là 40K thanh ghi.
Việc giảm các khu vực không được sử dụng sẽ làm tăng số điểm của thiết bị so với giá trị ban đầu.

Hình sau hiển thị cửa sổ "Device/Label Memory Area Setting" (Thiết bị khu vực bộ nhớ của thiết bị/nhãn) dưới dạng các thông số để điều chỉnh khu vực bộ nhớ.
Dung lượng của thiết bị tăng lên khi dung lượng của khu vực nhãn hoặc bộ nhớ tệp giảm đi.
Ngoài ra, dung lượng của toàn bộ khu vực bộ nhớ của thiết bị/nhãn có thể được mở rộng bằng cách sử dụng băng từ SRAM mở rộng.



1.2.2 Thiết lập thiết bị

Số điểm của thiết bị được chỉ định cho mỗi thiết bị có thể được thay đổi trong cửa sổ "Device Setting" (Thiết lập thiết bị). Giá trị ban đầu của một số thiết bị là 0 điểm. Chỉ định số điểm khi sử dụng thiết bị đó.



Item	Symbol	Device Points	Latch (1)	Latch (2)
Input	X	12K		
Output	Y	12K		
Internal Relay	M	12K	No Setting	No Setting
Link Relay	B	8K		
Link Special Relay	SB	2K		
Annunciator	F	2K		
Edge Relay	V	2K		
Step Relay	S	0		
Timer	T	1K		
Long Timer	LT	1K		
Retentive Timer	ST	0		
Long Retentive Time	LST	0		
		512		
		512		
		18K		
		8K		
		2K		
Latch Relay	L	8K		No Setting
Total Device		38.4K Word	0.0K Word	
Total Word Device		34.5K Word		
Total Bit Device		62.0K Bit		

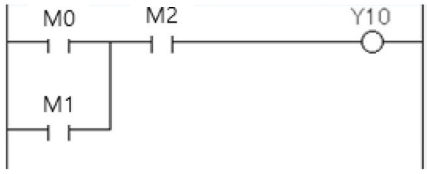
Cửa sổ "Device Setting" (Thiết lập thiết bị)

1.3 Sử dụng các tên nhãn có liên quan đến các ứng dụng

1.3.1 Lợi thế của việc sử dụng các nhãn

Tên thiết bị được sử dụng trong chương trình điều khiển phải bao gồm một chữ cái và một số, như "M0" và "D5". Khi tên nhãn có liên quan đến ứng dụng, như "StartSwitch", mục tiêu xử lý trở nên rõ ràng.
Tên nhãn có thể được thiết lập tự do theo ứng dụng. Người dùng không cần xem xét số hiệu thiết bị cho các khu vực có sử dụng các nhãn.

Chương trình sử dụng tên thiết bị



Chương trình sử dụng nhãn

Các nhãn được phân loại thành hai loại sau theo phạm vi sử dụng.

- **Nhãn chung**
  Nhãn chung có thể được sử dụng cho tất cả các chương trình trong một dự án.

- **Nhãn cục bộ**
  Nhãn cục bộ chỉ có thể được sử dụng trong chương trình có đăng ký nhãn đó.

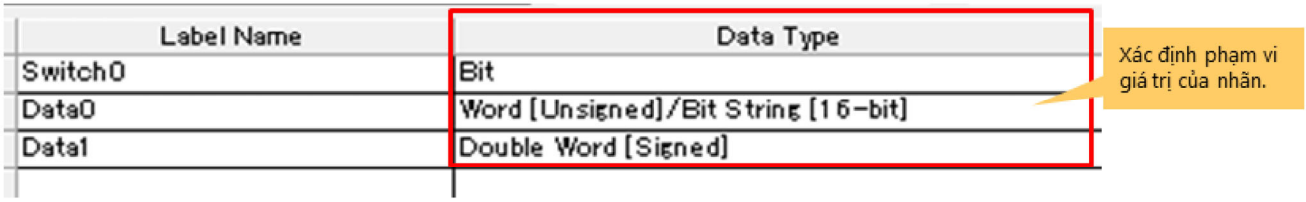
Khi sử dụng các nhãn cho các thiết bị thực tế (X, Y), tên thiết bị phải được chỉ định cho các nhãn chung bằng cách sử dụng GX Works3.

1.3.2 Kiểu dữ liệu của các nhãn

Phải xác định kiểu dữ liệu cho mỗi nhãn để xác định phạm vi của giá trị cần được xử lý.
Kiểu dữ liệu bao gồm bit và số nguyên, như chỉ ra dưới đây.

Kiểu dữ liệu		Phạm vi dữ liệu
Kiểu bit		Trạng thái bật/tắt của thiết bị bit và trạng thái đúng/sai của kết quả thực hiện
Kiểu số nguyên	Từ (không dấu)	0 đến 65.535
	Từ (có dấu)	-32.768 đến 32.767
	Thanh ghi đôi (32 bit) (không dấu)	0 đến 4.294.967.295
	Thanh ghi đôi (32 bit) (có dấu)	-2.147.483.648 đến 2.147.483.647

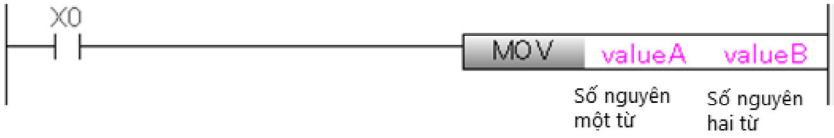
Khi sử dụng kiểu số nguyên, chọn loại độ dài dữ liệu một từ hoặc 2 từ theo phạm vi dữ liệu, và chọn loại có dấu hoặc không dấu theo tính cần thiết để xử lý các giá trị âm.
Xác định kiểu dữ liệu của một nhãn khi thiết lập tên nhãn bằng GX Works3.



Cửa sổ thiết lập nhãn

1.3.3 Tên nhãn thể hiện kiểu dữ liệu

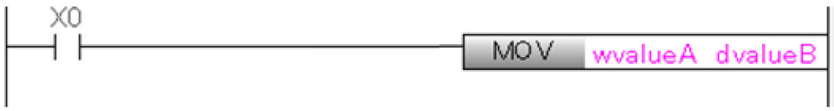
Sử dụng các kiểu dữ liệu khác nhau trên nguồn chuyển và điểm đến có thể gây ra lỗi chuyển đổi hoặc kết quả không mong muốn. Bên dưới là ví dụ chương trình của trường hợp như vậy.



Các giá trị hai từ không thể được chuyển đến nhãn kiểu một từ. Tuy nhiên, kiểu dữ liệu không thể được xác định theo tên nhãn.
Do đó, các tiền tố thể hiện kiểu dữ liệu có thể được thêm vào tên nhãn để giúp xác định rõ ràng kiểu dữ liệu.
Kiểu đặt tên nhãn này được gọi là ký hiệu Hungary.

Kiểu dữ liệu		Phạm vi dữ liệu	Tiền tố	Mở rộng tiền tố
Kiểu bit		Trạng thái bật/tắt của thiết bị bit và trạng thái đúng/sai của kết quả thực hiện	b	**b**it
Kiểu số nguyên	Từ (không dấu)	0 đến 65.535	u	**u**nsigned word (từkhông dấu)
	Từ (có dấu)	-32.768 đến 32.767	w	signed **w**ord (từ có dấu)
	Độ dài dữ liệu 2 từ (32 bit) (không dấu)	0 đến 4.294.967.295	ud	**u**nsigned **d**ouble-word (hai từ không dấu)
	Độ dài dữ liệu 2 từ (32 bit) (có dấu)	-2.147.483.648 đến 2.147.483.647	d	signed **d**ouble-word (hai từ có dấu)

Ví dụ chương trình trên đầu trang có thể được ghi như sau bằng cách sử dụng ký hiệu Hungary:



Bằng cách sử dụng ký hiệu Hungary, các trường hợp không thống nhất về kiểu dữ liệu có thể được xác định trong quá trình ghi chương trình.

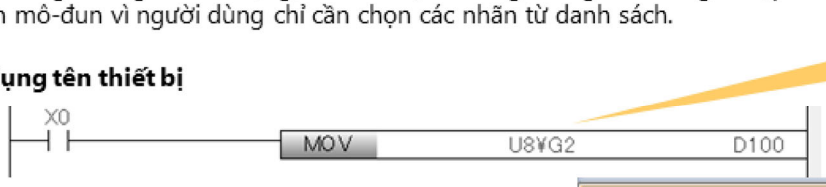
Trong phần còn lại, tên nhãn trong các ví dụ được ghi theo ký hiệu Hungary.

1.3.4 Sử dụng nhãn đã chuẩn bị

Khi cấu hình mô-đun được thiết lập trong sơ đồ cấu hình mô-đun, các nhãn (nhãn mô-đun) thể hiện các ký hiệu mô-đun hoặc thiết lập các giá trị tương ứng với vị trí cài đặt mô-đun được đăng ký tự động.

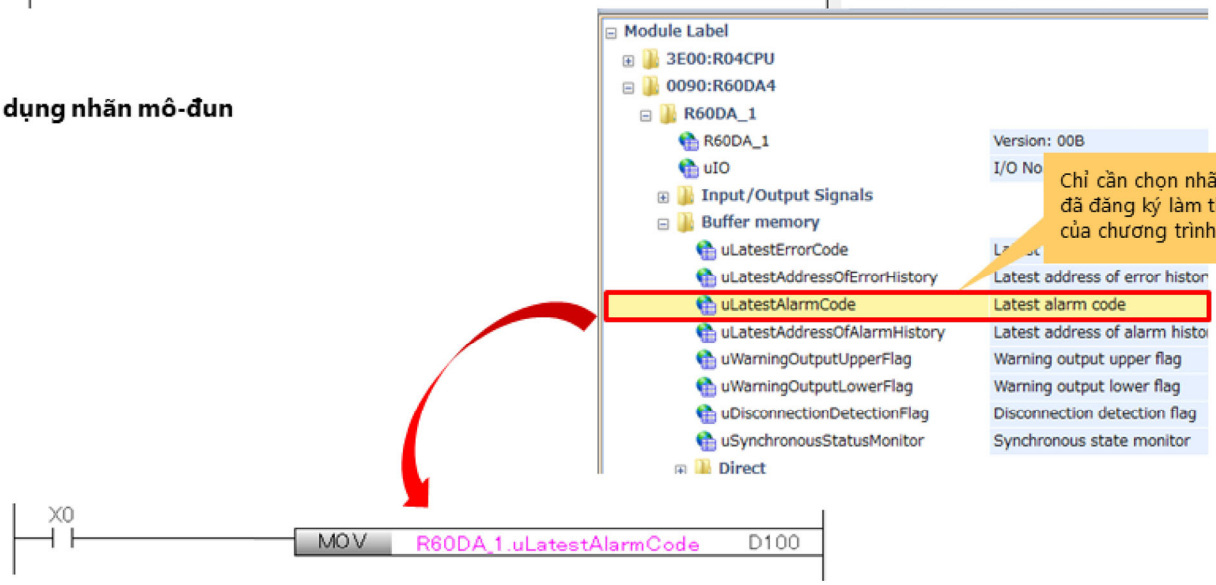
Để lập trình bằng cách sử dụng tên thiết bị, số hiệu thiết bị và địa chỉ bộ nhớ đệm tương ứng với các ký hiệu phải được kiểm tra trong hướng dẫn sử dụng, sẽ mất một khoảng thời gian. Thời gian lập trình có thể giảm xuống bằng cách sử dụng các nhãn mô-đun vì người dùng chỉ cần chọn các nhãn từ danh sách.

Khi sử dụng tên thiết bị



Kiểm tra và mô tả vị trí cài đặt và địa chỉ bộ nhớ đệm.

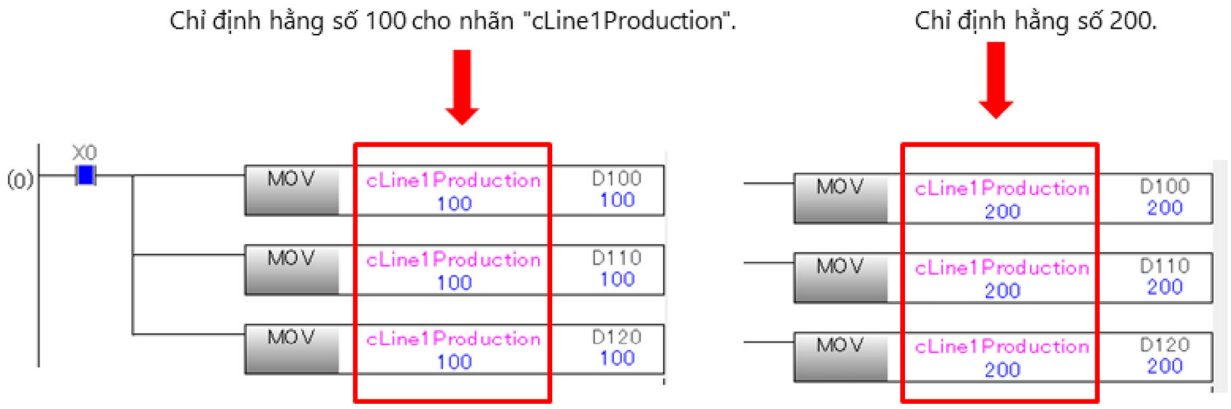
Khi sử dụng nhãn mô-đun



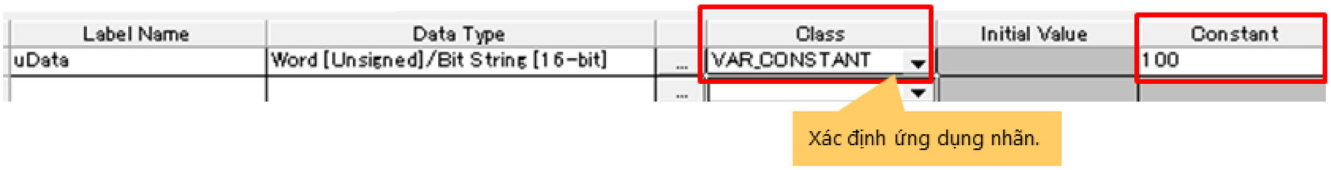
Chỉ cần chọn nhãn mô-đun đã đăng ký làm thành tố của chương trình.

1.3.5 Chỉ định hằng số cho các nhãn

Hằng số có thể được chỉ định cho các nhãn.
Khi các hằng số được chỉ định cho các nhãn, nên các giá trị có thể được thay đổi mà không cần sửa đổi chương trình.
Có thể thay đổi chung hằng số giống nhau được sử dụng cho nhiều nhãn.



Để chỉ định một hằng số cho nhãn, hãy thay đổi lớp xác định ứng dụng nhãn trên cửa sổ để thiết lập nhãn.
Đối với các nhãn cục bộ, hãy chọn "VAR_CONSTANT".

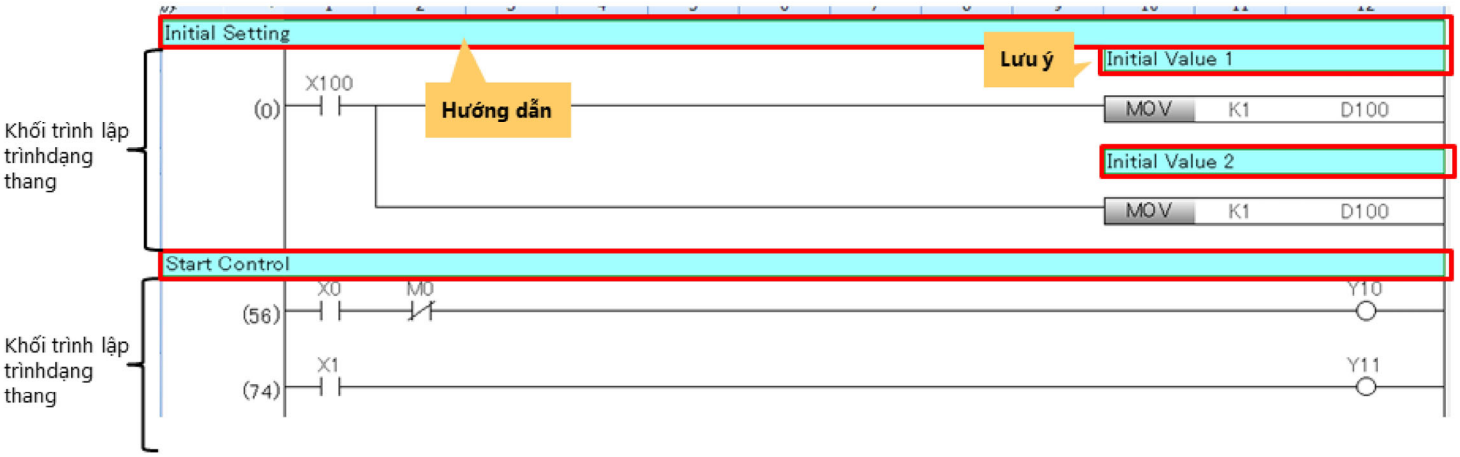


1.4 Cải thiện khả năng đọc chương trình

Có thể thêm các chú thích, như thông tin chi tiết xử lý và tên thiết bị, vào chương trình.
Các chú thích giúp làm rõ cách chương trình hoạt động.

Có thể chọn kiểu chú thích theo thành tố hoặc phạm vi của chương trình.
Trong ví dụ chương trình trên, "chú thích thiết bị/nhãn" đã được thêm vào để làm rõ các ứng dụng của thiết bị hoặc nhãn và các kiểu của thiết bị I/O được kết nối.

Ngoài ra, kiểu chú thích bao gồm một "hướng dẫn" được thêm vào khối trình lập trình dạng thang để giúp làm rõ luồng xử lý và các "lưu ý" giúp làm rõ thông tin chi tiết về cuộn cảm và hướng dẫn ứng dụng.



1.5 Tóm tắt chương này

Trong chương này, bạn đã tìm hiểu về:

- Phân giao số thứ tự I/O
- Điều chỉnh khu vực bộ nhớ
- Lập trình với các nhãn
- Các chú thích trong chương trình

Các điểm quan trọng

Phân giao số thứ tự I/O	• Số thứ tự I/O được chỉ định tự động vào các vị trí, bắt đầu từ mô-đun gần mô-đun CPU nhất • Các chương trình có thể được sử dụng trong các hệ thống khác nhau khi số thứ tự I/O cố định được chỉ định cho các mô-đun theo cách thủ công
Thiết lập thiết bị và thiết lập khu vực bộ nhớ	• Số điểm của thiết bị khác nhau tùy thuộc vào mô-đun CPU • Số điểm của thiết bị có thể tăng lên bằng cách giảm các khu vực bộ nhớ không được sử dụng • Số điểm của thiết bị cho mỗi thiết bị có thể được thay đổi theo hiện trạng sử dụng
Lập trình với các nhãn	Mục tiêu xử lý trở nên rõ ràng khi sử dụng các nhãn
Chú thích	Thông tin chi tiết và luồng xử lý trở nên dễ hiểu hơn khi thêm vào các chú thích

Chương 2 Lập trình nâng cao

Chương này mô tả cách sử dụng nâng cao các thiết bị và lập trình nhãn.

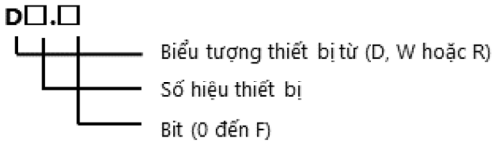
2.1 Sử dụng thiết bị từ theo đơn vị bit

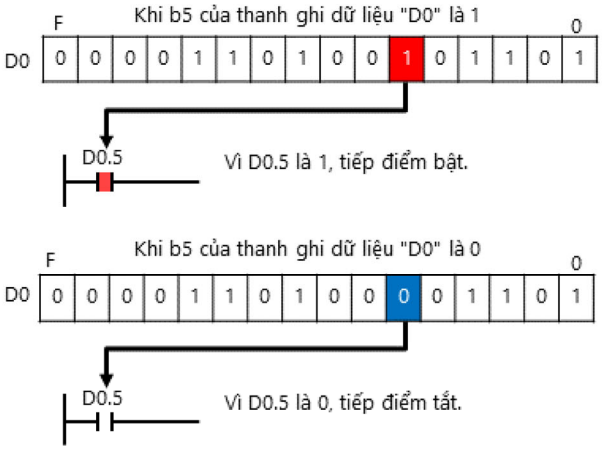
Các thiết bị Word, như thanh ghi dữ liệu, thường được sử dụng theo đơn vị word, nhưng chúng cũng có thể được sử dụng theo đơn vị bit.
Đơn vị bit được sử dụng để xác định một bit đặc biệt trong thanh ghi dữ liệu (D).

Ví dụ) Thanh ghi dữ liệu (D)

0	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1

Định dạng đặc tính bit



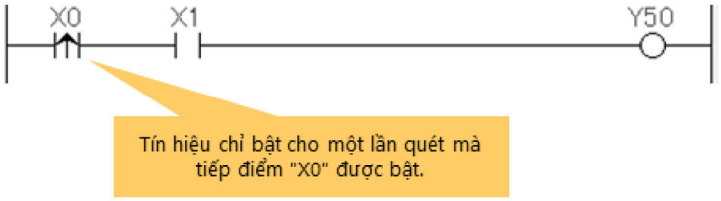


Để sử dụng các nhãn, thực hiện mô tả dưới dạng "uData.2" và "uData.5".

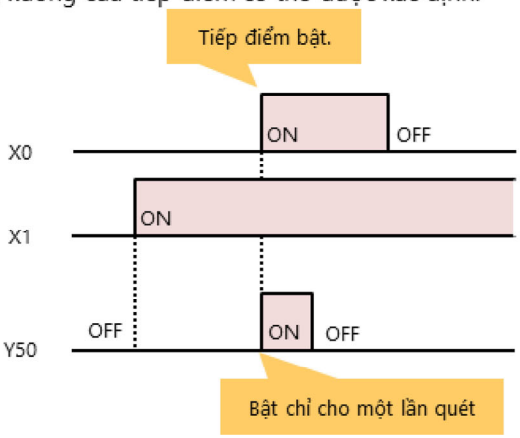
2.2 Chỉ bật thiết bị khi trạng thái tiếp xúc thay đổi

Một tín hiệu chỉ bật cho một lần quét trên sườn xung lên hoặc trên sường xung xuống của tiếp điểm có thể được xác định. Chức năng này hữu ích để kiểm soát độ tăng và giảm theo điều kiện đầu vào.

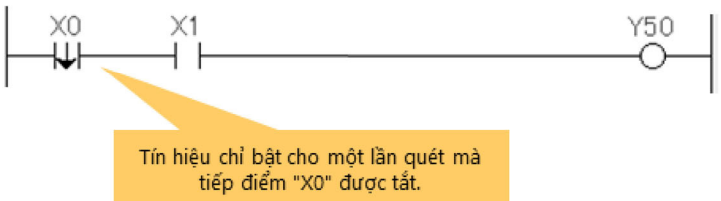
Đặc tính cạnh sườn lên của tiếp điểm



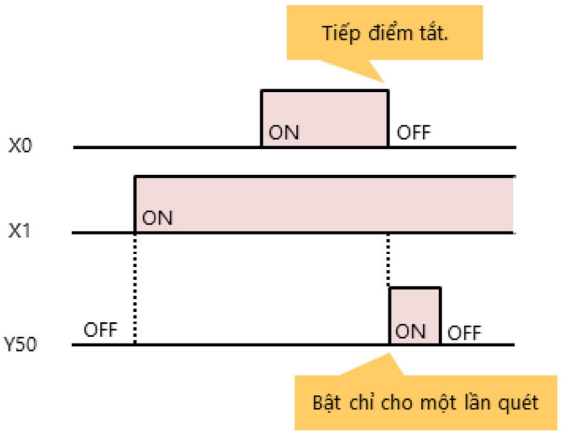
Tín hiệu chỉ bật cho một lần quét mà tiếp điểm "X0" được bật.



Đặc tính cạnh sườn xuống của tiếp điểm



Tín hiệu chỉ bật cho một lần quét mà tiếp điểm "X0" được tắt.



2.3 Giữ lại thời gian đo của bộ hẹn giờ

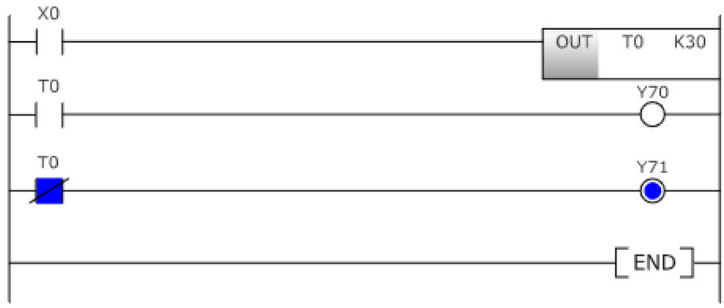
Phần này mô tả một trong những thiết bị của bộ hẹn giờ, bộ hẹn giờ có khả năng nhớ các sự kiện, có thể giữ lại thời gian đo.

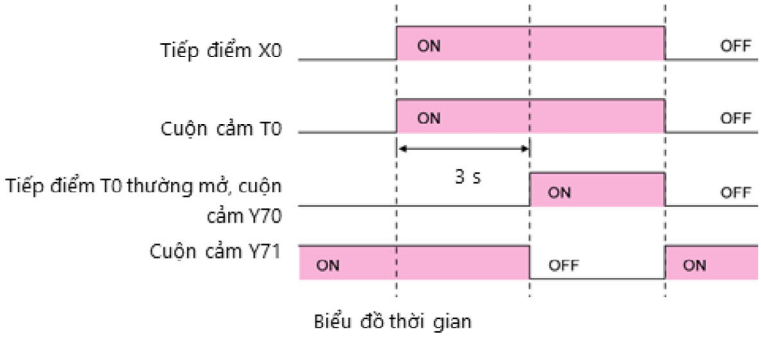
2.3.1 Sự khác biệt giữa bộ hẹn giờ và bộ hẹn giờ có khả năng nhớ các sự kiện

Trước khi giải thích về bộ hẹn giờ có khả năng nhớ các sự kiện, hãy xem xét cách bộ hẹn giờ hoạt động.

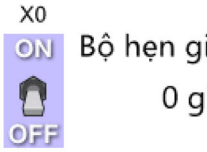
Bộ hẹn giờ bắt đầu phép đo khi cuộc cảm bật. Khi đã vượt quá khoảng thời gian xác định, hết thời gian và tiếp điểm được bật. Khi cuộn cảm tắt, thời gian đo được thiết lập về "0". Biểu tượng thiết bị của bộ hẹn giờ là "T".

Sử dụng công tắc đầu vào ở bên phải để xem cách bộ hẹn giờ hoạt động.





Biểu đồ thời gian



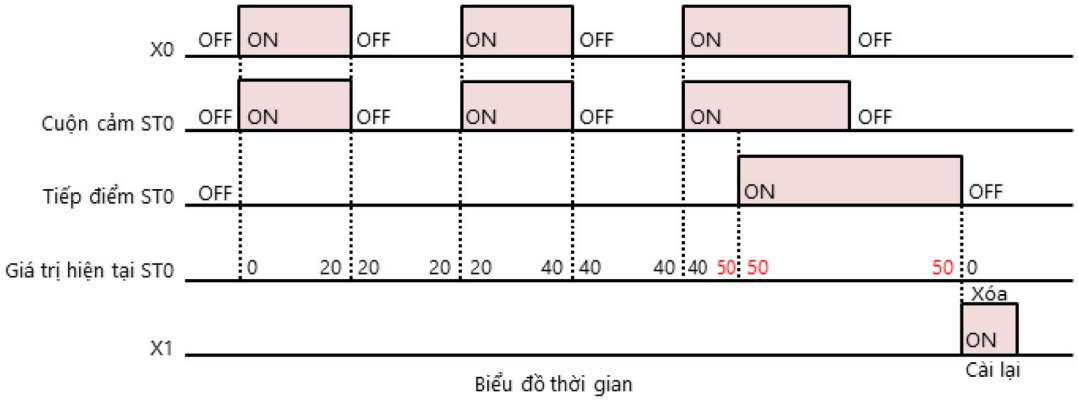
Bộ hẹn giờ:

0 giây

Ba giây sau khi X0 bật, Y70 bật và Y71 tắt.

2.3.1 Sự khác biệt giữa bộ hẹn giờ và bộ hẹn giờ có khả năng nhớ các sự kiện

Bộ hẹn giờ có khả năng nhớ các sự kiện hữu ích trong việc đo tổng thời gian hoạt động. Bộ hẹn giờ có khả năng nhớ các sự kiện bắt đầu phép đo khi cuộc cảm bật. Khi đã vượt quá khoảng thời gian xác định, hết thời gian và tiếp điểm được bật. Khi cuộn cảm tắt, thời gian đo không được thiết lập lại. Khi cuộn cảm bật lại, phép đo bắt đầu lại từ giá trị đã lưu. Biểu tượng thiết bị của bộ hẹn giờ có khả năng nhớ các sự kiện là "ST".



Biểu đồ thời gian

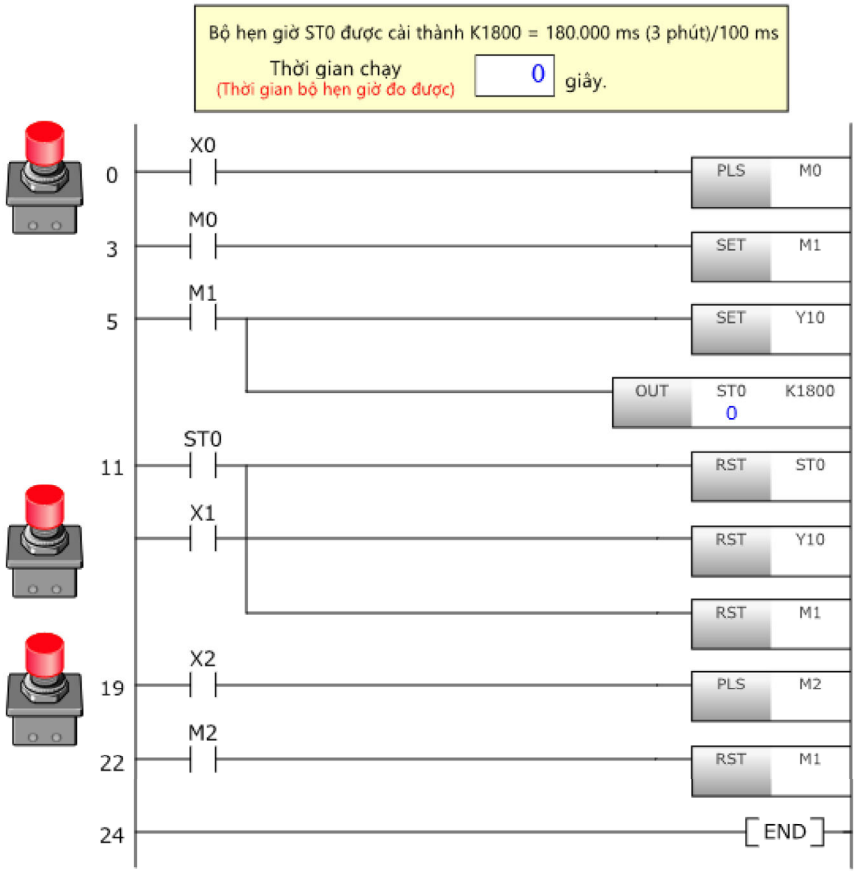
2.3.2 Ví dụ chương trình về bộ hẹn giờ có khả năng nhớ các sự kiện

Hãy xem xét cách bộ hẹn giờ có khả năng nhớ các sự kiện hoạt động bằng cách giả lập một máy đang vận hành bằng cách sử dụng công tắc đầu vào (X0 đến X2).
*Bộ hẹn giờ có khả năng nhớ các sự kiện (ST0) được thiết lập theo bước tăng 100 ms.

1. Khi X0 bật, quá trình vận hành bắt đầu.
2. Khi X2 bật, quá trình vận hành tạm dừng và giá trị hiện tại được lưu lại.
3. Khi X0 bật lại, quá trình vận hành bắt đầu lại.
4. Khi X1 bật, quá trình vận hành kết thúc và giá trị hiện tại được thiết lập lại.

X0 đến X2: Chuyển đổi đầu vào

Y10: Tín hiệu bắt đầu



2.3.3 Thiết lập cho bộ hẹn giờ có khả năng nhớ các sự kiện

Số điểm được sử dụng bởi bộ hẹn giờ có khả năng nhớ các sự kiện là "0" theo mặc định.
Trước khi sử dụng bộ hẹn giờ có khả năng nhớ các sự kiện, thiết lập số điểm trong "Device Setting" (Thiết lập thiết bị) của thông số CPU bằng GX Works3.

Trong ví dụ sau, 64 điểm (ST0 đến ST63) được thiết lập cho bộ hẹn giờ có khả năng nhớ các sự kiện.

Item	Symbol	Device		Local Device			Latch (1)	Latch (2)
		Points	Range	Start	End	Points		
Input	X	12K	0 to 2FFF					
Output	Y	12K	0 to 2FFF					
Internal Relay	M	12K	0 to 12287				No Setting	No Setting
Link Relay	B	16K	0 to 3FFF				No Setting	No Setting
Link Special Relay	SB	16K	0 to 3FFF					
Annunciator	F	2K	0 to 2047				No Setting	No Setting
Edge Relay	V	2K	0 to 2047				No Setting	No Setting
Step Relay	S	0						
Timer	T	1K	0 to 1023				No Setting	No Setting
Long Timer	LT	1K	0 to 1023				No Setting	No Setting
Retentive Timer	ST	64	0 to 63				No Setting	No Setting
Long Retentive Time	LST	0					No Setting	No Setting
Counter	C	512	0 to 511				No Setting	No Setting
Long Counter	LC	512	0 to 511				No Setting	No Setting
Data Register	D	18K	0 to 18431				No Setting	No Setting
Link Register	W	8K	0 to 1FFF				No Setting	No Setting
Link Special Registe	*SW*	2K	0 to 7FF					
Latch Relay	L	8K	0 to 8191					No Setting
Total Device			39.9K Word			0.0K Word		
Total Word Device			34.6K Word			0.0K Word		
Total Bit Device			84.2K Bit			0.0K Bit		

2.3.4 Sử dụng một nhãn để xác định bộ hẹn giờ

Thiết lập "Timer" (Bộ hẹn giờ) thành kiểu dữ liệu khi nhãn sử dụng thiết bị bộ hẹn giờ.

Label Name	Data Type
uTimer1	Timer
uTimer2	Retentive Timer

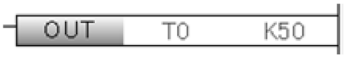
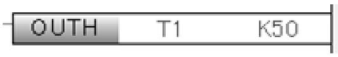
Thiết lập thiết bị cần thiết để sử dụng bộ hẹn giờ có khả năng nhớ các sự kiện như mô tả trong phần trước đó. Tuy nhiên, không cần cho các nhãn.

2.4 Thay đổi đơn vị đo của bộ hẹn giờ

Đơn vị và thời gian đo khác nhau tùy thuộc vào kiểu bộ hẹn giờ.

- Bộ hẹn giờ tốc độ cao (đơn vị đo ngắn)
- Bộ hẹn giờ tốc độ thấp (đơn vị đo dài)
- Bộ hẹn giờ dài có khả năng thực hiện các phép đo thời gian dài

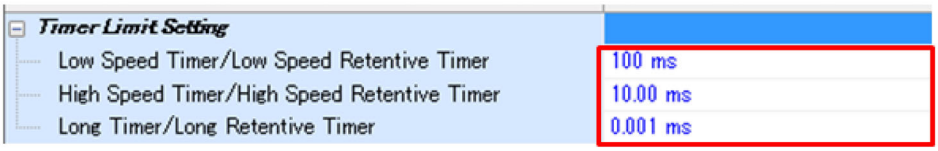
Mỗi bộ hẹn giờ trên đều có chức năng nhớ các sự kiện.

Loại	Đơn vị đo	Ví dụ chương trình	Hoạt động	Giữ lại giá trị hiện tại
Bộ hẹn giờ tốc độ thấp	100 ms (mặc định)		Bộ hẹn giờ tốc độ thấp T0 đo 5 giây.	16 bit
Bộ hẹn giờ tốc độ cao	10,00 ms (mặc định)		Bộ hẹn giờ tốc độ thấp T1 đo 0,5 giây.	
Bộ hẹn giờ có khả năng nhớ các sự kiện tốc độ thấp	100 ms (mặc định)		Bộ hẹn giờ có khả năng nhớ các sự kiện tốc độ thấp ST0 đo 5 giây.	
Bộ hẹn giờ có khả năng nhớ các sự kiện tốc độ cao	10,00 ms (mặc định)		Bộ hẹn giờ có khả năng nhớ các sự kiện tốc độ cao ST1 đo 0,5 giây.	
Bộ hẹn giờ dài	0,001 ms (mặc định)		Bộ hẹn giờ dài LT0 đo 0,005 giây.	32 bit
Bộ hẹn giờ có khả năng nhớ các sự kiện dài			Bộ hẹn giờ có khả năng nhớ các sự kiện dài LST0 đo 0,005 giây.	

Đơn vị đo ban đầu là 100 ms đối với bộ hẹn giờ tốc độ thấp, 10 ms đối với bộ hẹn giờ tốc độ cao và 0,001 ms đối với bộ hẹn giờ dài.
Hãy xem trang tiếp theo để biết cách thay đổi đơn vị đo.

2.4 Thay đổi đơn vị đo của bộ hẹn giờ

Đơn vị đo của bộ hẹn giờ có thể được thay đổi trong "Timer Limit Setting" (Thiết lập giới hạn bộ hẹn giờ) của thông số CPU.



2.5 Xử lý nhiều thiết bị (thanh ghi chỉ mục)

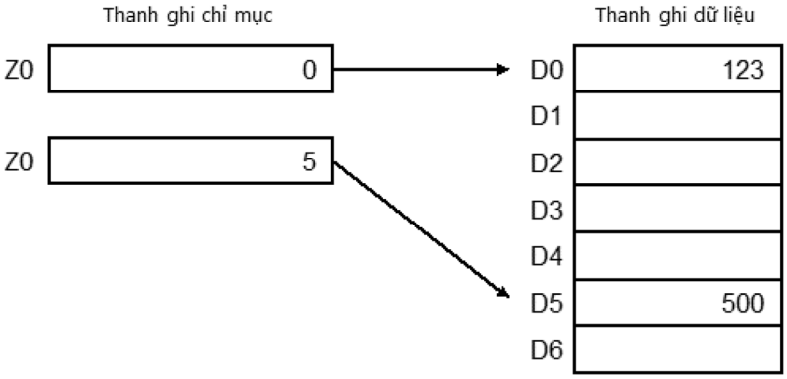
Thanh ghi chỉ mục (Z) được sử dụng kết hợp với một thiết bị khác để trực tiếp xác định (sửa đổi) số hiệu thiết bị của thiết bị điều khiển mục tiêu. Thanh ghi chỉ mục rất hữu ích để đơn giản hóa các chương trình vì nó có thể mô tả nhiều thiết bị trong một lô.

- Thanh ghi chỉ mục được mô tả sau biểu tượng thiết bị và số hiệu thiết bị
- Số hiệu thiết bị điều khiển mục tiêu thực tế = Biểu tượng thiết bị (số hiệu thiết bị + thanh ghi chỉ mục)
- Số điểm của thiết bị cho thanh ghi chỉ mục theo mặc định là 20 điểm (Z0 đến Z19)

2.5.1 Ví dụ về ứng dụng của thanh ghi chỉ mục

Khi thiết bị được mô tả là "D0Z0", nó có nghĩa là D(0+Z0).

Ví dụ) Khi Z0 có giá trị là 0, số hiệu thiết bị là D0.
Khi Z0 có giá trị là 5, số hiệu thiết bị là D5.

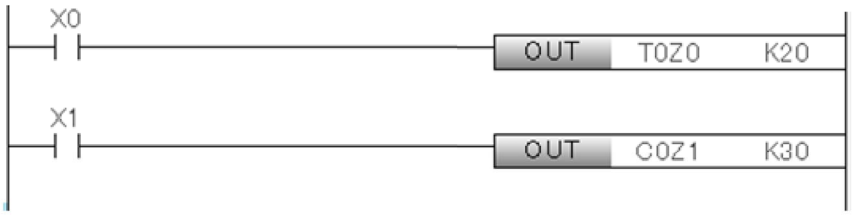


2.5.2 Các thiết bị có thể được sửa đổi bằng thanh ghi chỉ mục

Các thiết bị có thể được sửa đổi bằng thanh ghi chỉ mục bao gồm:

Thiết bị bit	X, Y, M, L, S, B, F
Thiết bị từ	T, C, D, R, W
Hằng số	K, H
Con trỏ	P

Lưu ý) Đối với các tiếp điểm và cuộn cảm được sử dụng trong bộ hẹn giờ và bộ đếm, chỉ có thể sử dụng thanh ghi chỉ mục "Z0" hoặc "Z1".



Khi Z0 là 1, T1 đo thời gian.

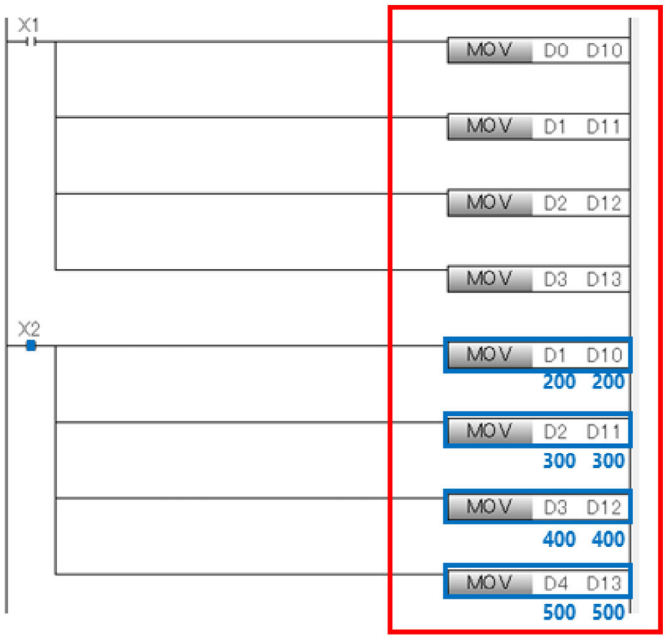
Khi Z1 là 5, C5 tiến hành đếm.

2.5.3 Đơn giản hóa các chương trình bằng thanh ghi chỉ mục

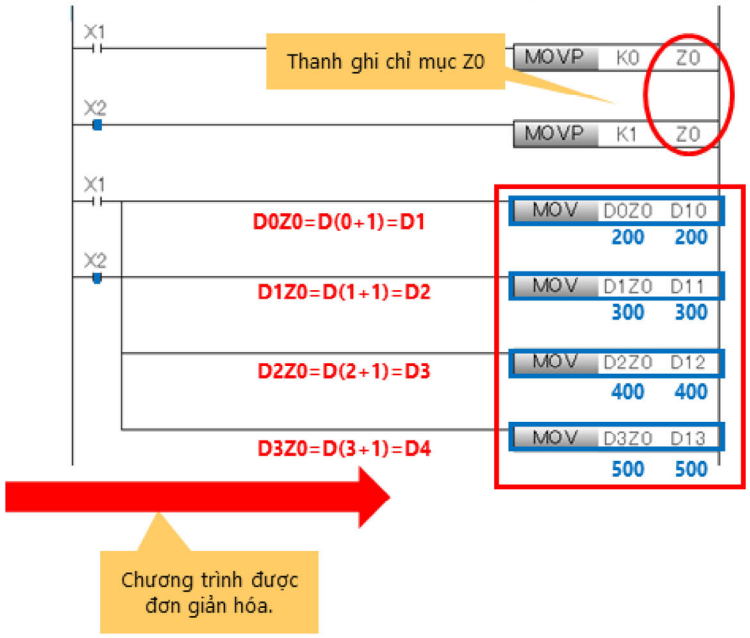
Các chương trình hiển thị bên dưới chuyển các giá trị trong "D0 đến D4" thành "D10 đến D13" khi X1 hoặc X2 bật. Các chương trình (1) và (2) sẽ mang lại kết quả tương tự. Trong chương trình (1), dữ liệu được chuyển trực tiếp. Trong chương trình (2), dữ liệu được chuyển trực tiếp trong thanh ghi chỉ mục.

Giá trị được lưu ban đầu
D0=100
D1=200
D2=300
D3=400
D4=500

(1) Khi thanh ghi chỉ mục không được sử dụng



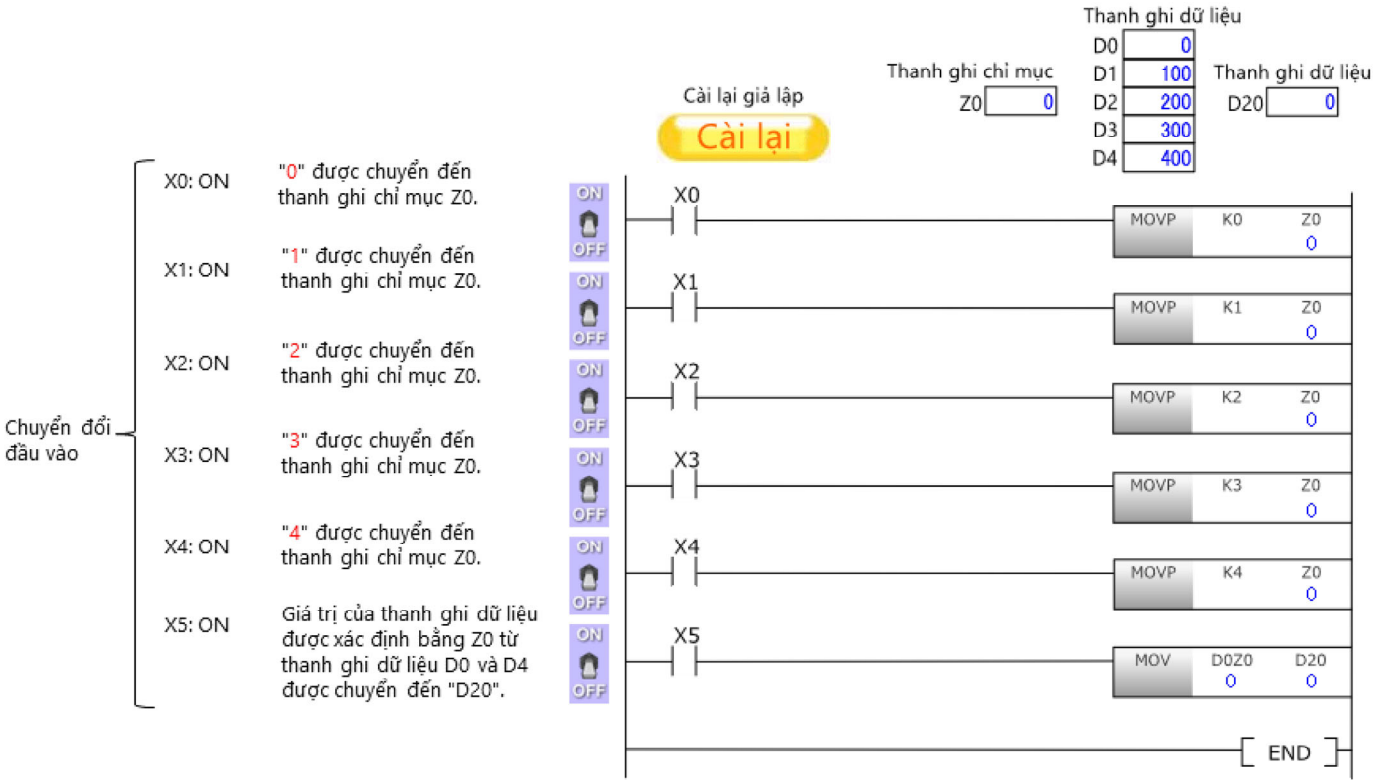
(2) Khi thanh ghi chỉ mục được sử dụng



Chương trình được đơn giản hóa.

2.5.4 Ví dụ chương trình của thanh ghi chỉ mục

Cách thanh ghi chỉ mục Z0 hoạt động có thể được giả lập bằng cách bật công tắc đầu vào X0 đến X5.
*K0 đến K400 đã được lưu trong thanh ghi dữ liệu D0 đến D4.
Bật công tắc đầu vào X0 đến X5 để kiểm tra các giá trị được lưu vào mỗi khu vực thiết bị.



2.6 Xử lý nhiều giá trị (dãy)

Bằng cách sử dụng một dãy, có thể xử lý nhiều giá trị bằng một tên nhãn.
Trong ví dụ sau, dữ liệu về khối lượng sản xuất tại một nhà máy sản xuất ô tô được lưu theo điểm đến.

Điểm đến	Country A	Country B	Country C
Khối lượng sản xuất	35 đơn vị	75 đơn vị	65 đơn vị

Dữ liệu về khối lượng sản xuất theo điểm đến được chỉ định cho một nhãn.
Khi không sử dụng một dãy, phải tạo các tên nhãn cho mỗi điểm đến.
Tuy nhiên, bằng cách sử dụng một dãy, khối lượng sản xuất cho nhiều điểm đến có thể được chỉ định cho và được lưu trong một tên nhãn.

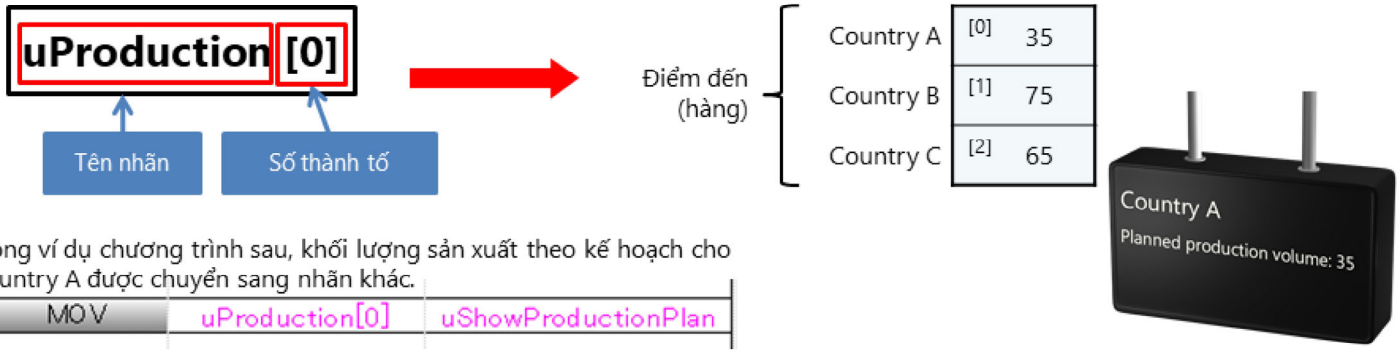
Khi không sử dụng một dãy

uProductionA
uProductionB
uProductionC

Khi sử dụng một dãy

uProduction

Các nhãn riêng trong dãi được xác định bằng các số thành tố. Số thành tố bắt đầu từ [0].



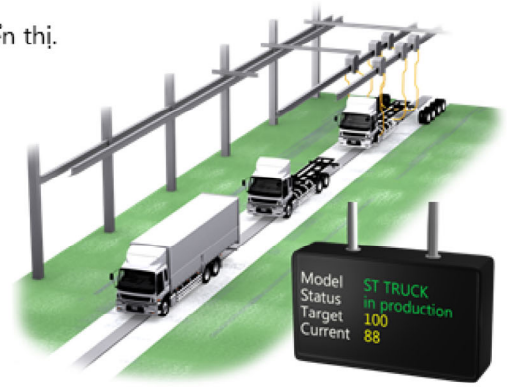
Trong ví dụ chương trình sau, khối lượng sản xuất theo kế hoạch cho Country A được chuyển sang nhãn khác.

2.7 Xử lý nhiều giá trị (cấu trúc)

Bằng cách sử dụng một cấu trúc, có thể xử lý nhiều nhãn có liên quan bằng một tên nhãn.
Trong ví dụ sau, trạng thái của dây chuyền sản xuất ô tô được hiển thị trên Andon (bảng hiển thị).

Bảng sau liệt kê các tên nhãn, giá trị và loại dữ liệu tương ứng với các mục được hiển thị.

Mục	Tên nhãn	Giá trị	Kiểu dữ liệu của nhãn
Model	sModel	'ST TRUCK'	Kiểu chuỗi
Trạng thái hoạt động	bStatus	'in production'	Kiểu bit
Khối lượng sản xuất mục tiêu hôm nay	uPlanQty	'100' đơn vị	Kiểu số nguyên (từ, không dấu)
Khối lượng sản xuất hiện tại	uActualQty	'88' đơn vị	Kiểu số nguyên (từ, không dấu)



Nếu không sử dụng cấu trúc cho nhà máy có nhiều dây chuyền sản xuất, phải thay đổi tên nhãn cho mỗi dây chuyền.
Sau đây là các ví dụ về tên nhãn với tên dây chuyền sản xuất được thêm vào.

Dây chuyền sản xuất đầu tiên

s1stLineModel
b1stLineStatus
u1stLinePlanQty
u1stLineActualQty

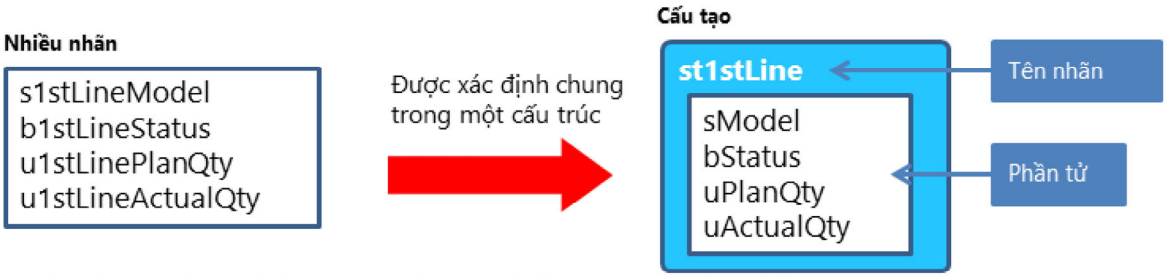
Dây chuyền sản xuất thứ hai

s2ndLineModel
b2ndLineStatus
u2ndLinePlanQty
u2ndLineActualQty

Khi số dây truyền sản xuất tăng lên, số nhãn cần được xử lý cũng tăng lên. Do đó, chương trình sẽ dài hơn và khó đọc hơn.

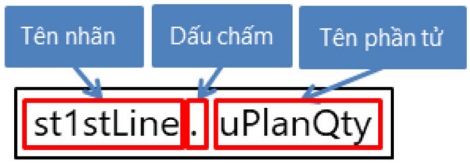
2.7 Xử lý nhiều giá trị (cấu trúc)

Cấu trúc cho phép một tên nhãn thể hiện nhiều nhãn liên quan đến một dây chuyền sản xuất.
Do đó, các cấu trúc được sử dụng để tổ chức, lưu trữ và xử lý chung các điều kiện và thông số kỹ thuật liên quan đến các chủ thể hoặc vấn đề thực tế như thiết bị, dụng cụ và phôi.

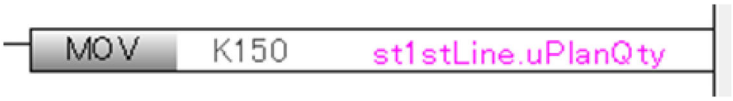


Các nhãn cấu trúc được cố định trước với "st" thể hiện **st**ructure (cấu trúc).
Các nhãn riêng được xác định theo cấu trúc được gọi là phần tử. Kiểu dữ liệu của mỗi phần tử có thể khác nhau.

Để xác định mỗi phần tử trong một cấu trúc, thêm tên phần tử sau tên nhãn cấu trúc với một dấu chấm (.) làm dấu phân cách ở giữa.



Trong ví dụ chương trình sau, một số nguyên được chỉ định cho một phần tử của nhãn kiểu cấu trúc cho dây chuyền sản xuất đầu tiên.



2.8 Giữ lại trạng thái thiết bị (khóa)

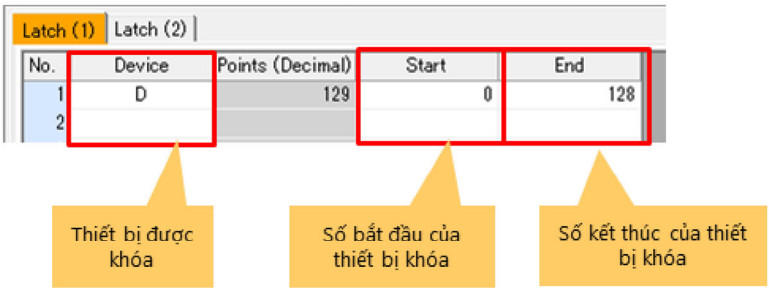
Phần này mô tả chức năng khóa để giữ lại các giá trị thiết bị khi mô-đun CPU dừng hoạt động.
Ví dụ: ngay cả khi xảy ra tình trạng mất nguồn vượt quá thời gian mất nguồn tạm thời cho phép, bộ điều khiển khả trình có thể khởi động lại chức năng kiểm soát tuần tự bằng cách sử dụng dữ liệu được lưu giữ khi dừng hoạt động.

Nếu chức năng khóa không được sử dụng, các giá trị thiết bị bị xóa và thiết lập lai thành giá trị ban đầu (tắt đối với các thiết bị bit và "0" đối với thiết bị từ) trong các trường hợp sau:

- Tắt nguồn
- Thiết lập lại bằng công tắc RUN (CHẠY)/STOP (DỪNG)/RESET (THIẾT LẬP LẠI)
- Mất nguồn quá thời gian mất nguồn tạm thời cho phép

2.8.1 Thiết lập khóa trên thiết bị

Thiết lập phạm vi khóa trong cửa sổ "Device Setting" (Thiết lập thiết bị) của thông số CPU.
Sau đây là ví dụ thiết lập khóa của thanh ghi dữ liệu, D0 đến D128.



2.8.2 Kiểu khóa và phương pháp xóa

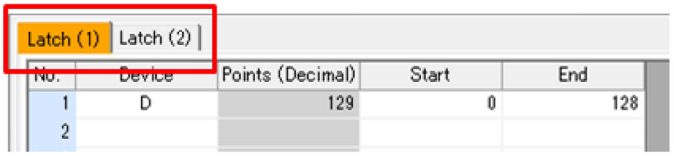
Có hai kiểu khóa (khóa (1) và khóa (2)) theo phương pháp xóa.

- Latch (1) (Khóa (1))

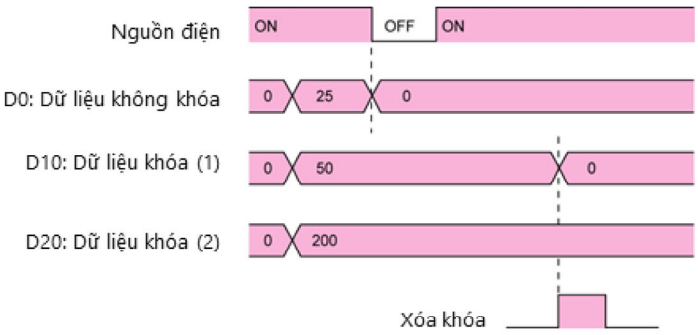
  Có thể xóa dữ liệu đã khóa bằng cách sử dụng chức năng vận hành bộ nhớ CPU* của GX Works3.
  Sử dụng khóa 1 khi cần xóa dữ liệu đã khóa tại điểm cài đặt.

- Latch (2) (Khóa (2))

  Có thể xóa dữ liệu đã khóa bằng cách sử dụng hướng dẫn trong chương trình.
  Sử dụng khóa 2 khi không cần xóa dữ liệu đã khóa tại điểm cài đặt.



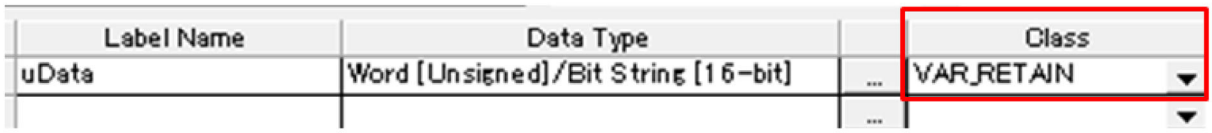
Sau đây là biểu đồ thời gian xóa khóa.



*Thực hiện chức năng này bằng cách chọn [Online] (Trực tuyến) → [CPU Memory Operation] (Hoạt động của bộ nhớ CPU) từ menu GX Works3.

2.8.3 Thiết lập khóa trên nhãn

Để thiết lập khóa trên nhãn, hãy chọn tên lớp có chứa "RETAIN" trên cửa sổ để thiết lập nhãn.
Đối với các nhãn cục bộ, hãy chọn "VAR_RETAIN".



Label Name	Data Type		Class
uData	Word [Unsigned]/Bit String [16-bit]	...	VAR_RETAIN
		...	

2.9 Giữ lại trạng thái thiết bị (thanh ghi tập tin)

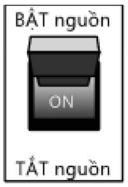
- Thanh ghi tập tin là một thiết bị từ được sử dụng để mở rộng thanh ghi dữ liệu (D)
- So sánh với thanh ghi dữ liệu, thanh ghi tập tin có thể xử lý số lượng dữ liệu lớn hơn
- Thanh ghi tập tin được lưu trong bộ nhớ dữ liệu của mô-đun CPU hoặc trong thẻ nhớ
- Dữ liệu được lưu trong thanh ghi tập tin sẽ được giữ lại ngay cả khi hệ thống bị tắt nguồn hoặc mô-đun CPU được thiết lập lại. Thanh ghi tập tin hữu ích để lưu trữ các giá trị được xác định trước, như các giá trị tiêu chuẩn
- Để xóa dữ liệu, ghi 0 cho phạm vi thanh ghi tập tin hoặc xóa bộ nhớ khỏi GX Works3
- Biểu tượng thiết bị là "ZR"

2.9.1 Hoạt động của chương trình ladder

Cách thanh ghi tập tin hoạt động có thể được giả lập bằng cách bật công tắc đầu vào và công tắc nguồn.

Công tắc nguồn

Bằng cách tắt và bật công tắc nguồn, có thể kiểm tra trạng thái dữ liệu trong thanh ghi tập tin, ZR0 và ZR1. (Dữ liệu được giữ lại.)



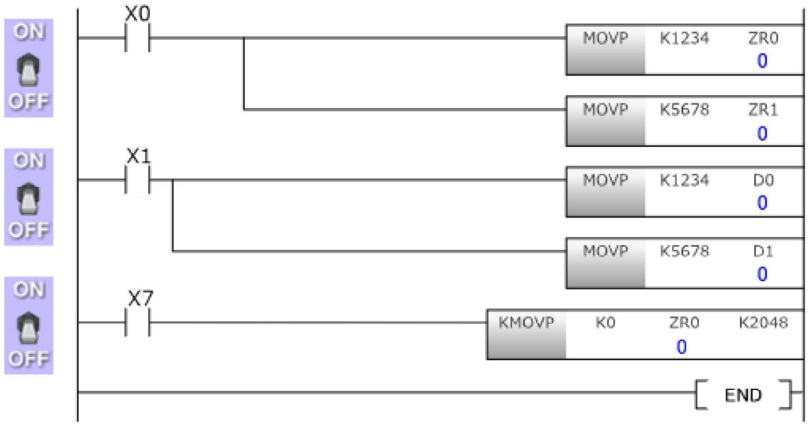
Cài lại giả lập

Cài lại

Thanh ghi tập tin		Thanh ghi dữ liệu	
ZR0	0	D0	0
ZR1	0	D1	0

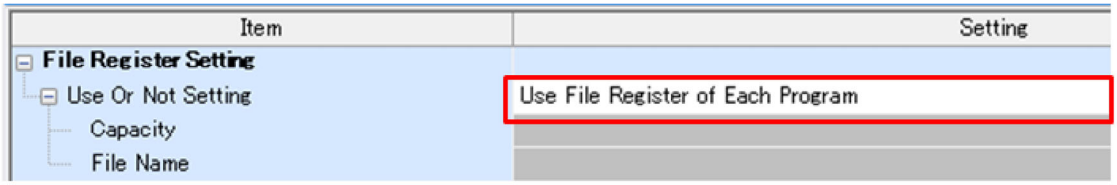
Chuyển đổi đầu vào

- X0: ON — Dữ liệu được ghi vào thanh ghi tập tin, ZR0 và ZR1.
- X1: ON — Dữ liệu được ghi vào thanh ghi dữ liệu, D0 và D1.
- X7: ON — Dữ liệu trong thanh ghi tập tin, ZR0 và ZR1, được đưa trở lại giá trị "0".

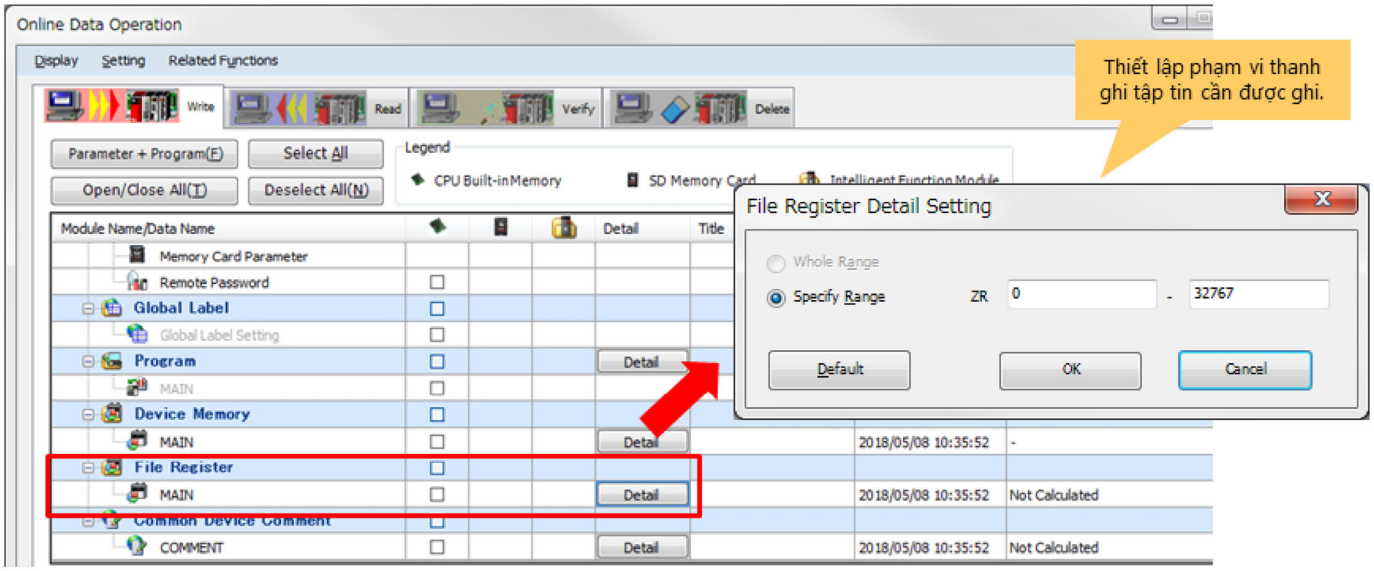


2.9.2 Thiết lập cho thanh ghi tập tin

Phần này mô tả thiết lập xác định một thanh ghi tập tin cục bộ làm điểm đến lưu trữ của mỗi chương trình. Chọn "File Register Setting" (Thiết lập thanh ghi tập tin) trong thông số CPU và chọn "Use File Register of Each Program" (Sử dụng thanh ghi tập tin của mỗi chương trình).



Để ghi dữ liệu vào bộ điều khiển khả trình, thiết lập thanh ghi tập tin phải được ghi vào mỗi chương trình.



2.10 Sử dụng các thiết bị có các chức năng và hoạt động được xác định trước

Rơ-le đặc biệt và thanh ghi đặc biệt được sử dụng trong mô-đun CPU có các chức năng và hoạt động được xác định trước.
Rơ-le đặc biệt (SM) là rơ-le gắn trong được sử dụng cho thông tin bit (bật/tắt) và thanh ghi đặc biệt (SD) là thanh ghi gắn trong được sử dụng cho thông tin từ.

2.10.1 Các loại rơ-le và thanh ghi đặc biệt

Rơ-le và thanh ghi đặc biệt được phân loại theo loại thông tin của chúng. Các loại chính được liệt kê bên dưới.
Trong các chương trình người dùng, rơ-le đặc và thanh ghi đặc biệt được sử dụng như điều kiện xác định cho chức năng kiểm soát. Chúng cũng được sử dụng để giam sát hoạt động, có thể được thực hiện trên trình giám sát thiết bị của GX Works3.

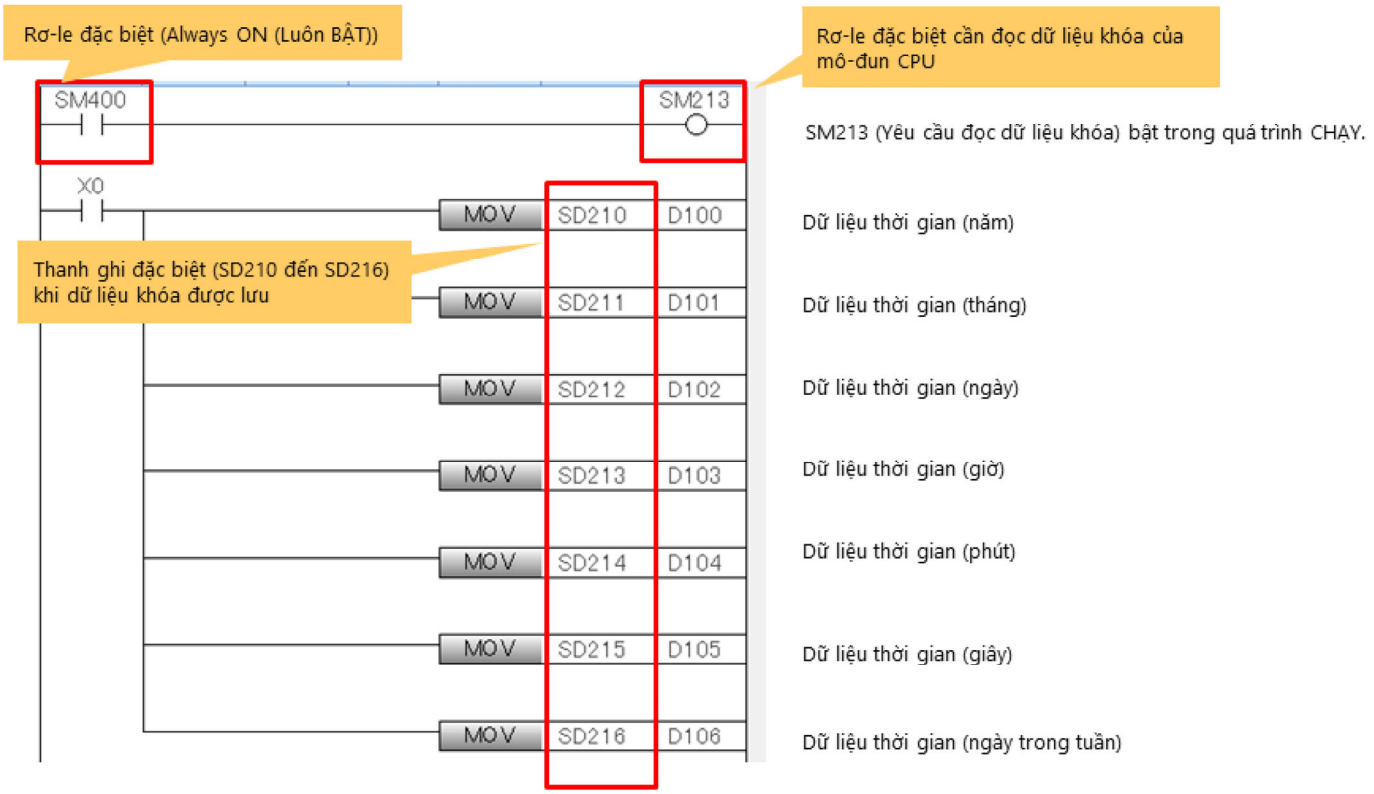
Thông tin chuẩn đoán
Kết quả chuẩn đoán của mô-đun CPU
Lỗi chuẩn đoán và mã lỗi

Thông tin hệ thống
Thông tin hệ thống của mô-đun CPU
Trạng thái hoạt động của mô-đun CPU, dữ liệu thời gian và các thông tin khác

Thời gian hệ thống
Các tín hiệu thời gian và giá trị đếm được sử dụng làm thành tố thời gian cơ bản
Các tín hiệu khá khác nhau (luôn ON (BẬT), ON/OFF (BẬT/TẮT) trong khoảng thời gian xác định và các tín hiệu khác)

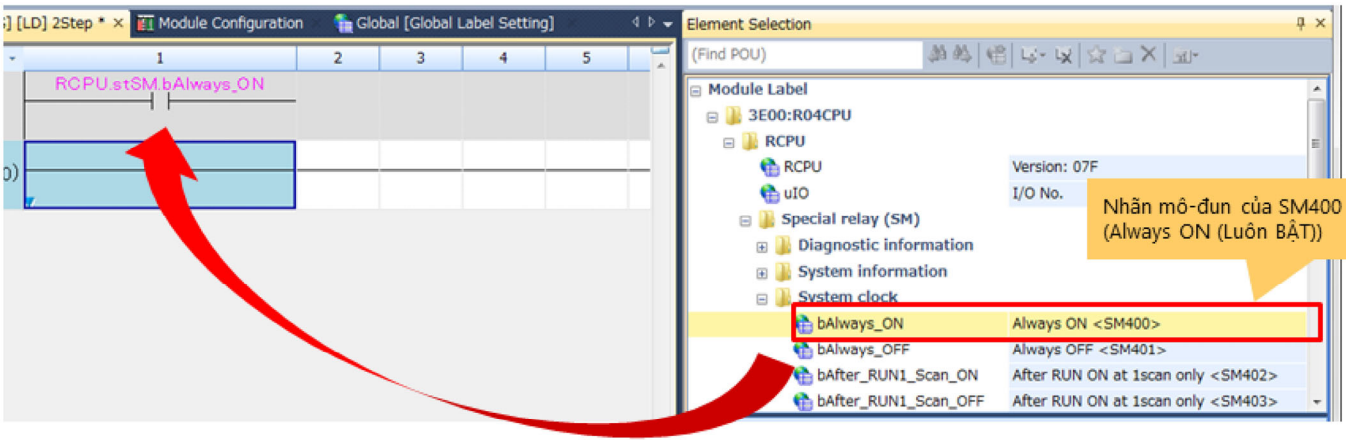
2.10.2 Ví dụ chương trình về rơ-le và thanh ghi đặc biệt

Sau đây là ví dụ chương trình để đọc dữ liệu khóa của mô-đun CPU bằng cách sử dụng rơ-le và thanh ghi đặc biệt.



2.10.3 Sử dụng các nhãn cho rơ-le và thanh ghi đặc biệt

Rơ-le và thanh ghi đặc biệt được chuẩn bị như các nhãn mô-đun trong mô-đun CPU. Chúng có thể được sử dụng đơn giản bằng cách chọn và đặt tên nhãn liên quan, không cần kiểm tra số hiệu thiết bị trong hướng dẫn sử dụng.



2.11 Tính toán bằng các số thực

2.11.1 Sử dụng các số thực

- "Số thực" là các giá trị số có dấu thập phân
- Số nguyên thường được sử dụng trong các chương trình điều khiển. Tuy nhiên, số thực được sử dụng trong các chương trình để kiểm soát hoạt động nâng cao như hàm số lượng giác và hoạt động theo cấp số nhân vì các giá trị số có dấu thập phân cần được xử lý trong các chương trình đó
- Dữ liệu số của số thực được xử lý trong mô-đun CPU là "dữ liệu dấu phẩy động"

Lưu ý:

- Một số thực luôn sử dụng hai thiết bị từ liên tiếp (dung lượng bộ nhớ 32 bit), bất kể số nào*

- Trong các chương trình điều khiển, hướng dẫn phép tính riêng (như cộng, trừ, nhân, chia và các hàm số đặc biệt) xử lý số thực đã được chuẩn bị. Hướng dẫn chuyển đổi giữa số nguyên và số thực cũng được chuẩn bị

*Nếu cần tính toán chính xác hơn với số lượng số quan trọng lớn hơn, hãy sử dụng bốn thiết bị từ.

- Các số thực sử dụng hai thiết bị từ được gọi là số thực có độ chính xác đơn
- Các số thực sử dụng bốn thiết bị từ được gọi là số thực có độ chính xác kép

Khóa học này tập trung vào các số thực có độ chính xác đơn.

2.11.2 Chú giải cho các số thực

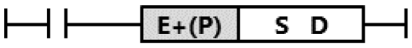
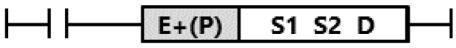
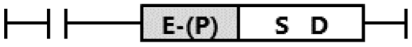
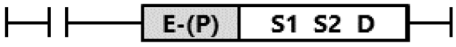
"E" được sử dụng để thể hiện một số thực.

Biểu thị một hằng số với các số thực

Để ghi hằng số, hãy bắt đầu với chữ cái "E".

Biểu thức thông thường	Ghi giá trị số thành. (Ví dụ) Ghi 10,2345 thành "E10.2345".
Biểu thức hàm mũ	Ghi giá trị số thành "(giá trị số) × 10^n". (Ví dụ) Ghi 1234,0 thành "E1.234+3".

2.11.3 Hướng dẫn phép tính (cộng và trừ)

Biểu tượng hướng dẫn	Ví dụ về trình lập trình dạng thang	
E+ (Cộng các số thực có độ chính xác đơn)	 Phép tính số thực "D + S = D" được thực hiện.	 Phép tính số thực "S1 + S2 = D" được thực hiện.
E- (Trừ các số thực có độ chính xác đơn)	 Phép tính số thực "D - S = D" được thực hiện.	 Phép tính số thực "S1 - S2 = D" được thực hiện.

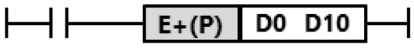
S (nguồn): Dữ liệu trước khi thao tác (không đổi, số hiệu thiết bị)
D (điểm đến): Điểm đến của dữ liệu sau khi thao tác (số hiệu thiết bị)
P: Hướng dẫn cần được thực hiện trên cạnh xung tăng (từ tắt đến bật)
S1 và S2: Hai mục dữ liệu cần được thao tác.

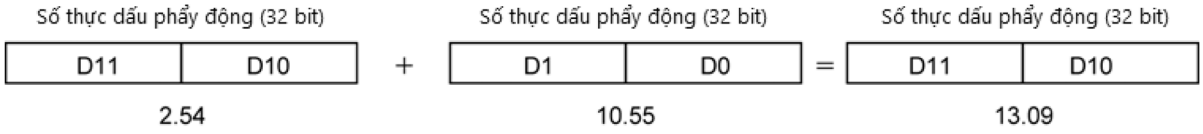
Lưu ý:

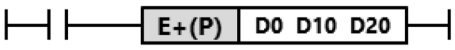
Không thể sử dụng chung số nguyên và số thực trong một hoạt động.
Đối với các phép tính số thực có độ chính xác đơn, S, S1 và S2 trong biểu thức tính phải là số thực có độ chính xác đơn.
Các số thực có độ chính xác duy nhất được lưu trong D.

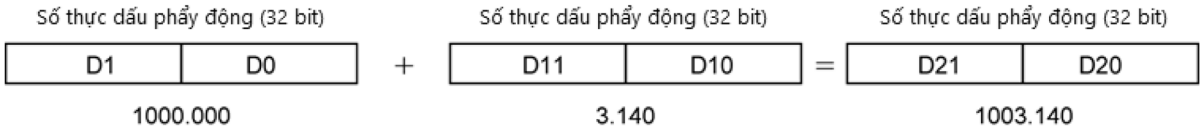
2.11.3 Hướng dẫn phép tính (cộng và trừ)

Ví dụ về ứng dụng của phép cộng và trừ



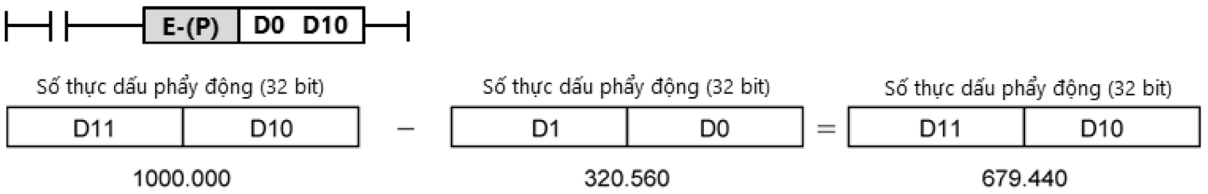


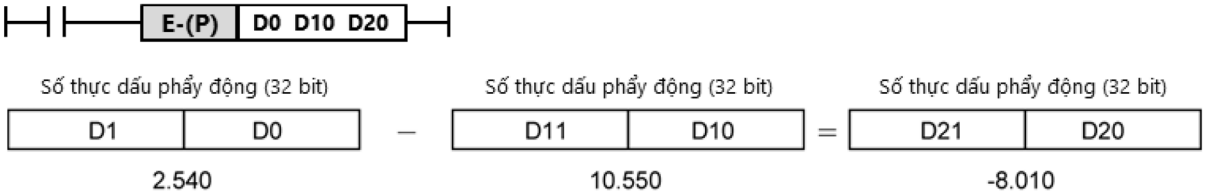




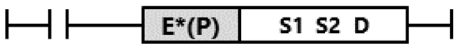
2.11.3 Hướng dẫn phép tính (cộng và trừ)

Ví dụ về ứng dụng của phép trừ





2.11.4 Hướng dẫn phép tính (phép nhân và chia)

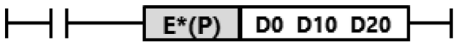
Biểu tượng hướng dẫn	Ví dụ về trình lập trình dạng thang
E* (Nhân các số thực có độ chính xác đơn)	 Phép tính số thực "S1 * S2 = D" được thực hiện.
E/ (Chia các số thực có độ chính xác đơn)	 Phép tính số thực "S1 / S2 = D" được thực hiện.

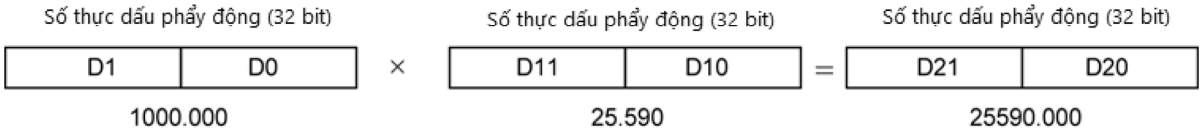
S1, S2 (nguồn): Hai mục dữ liệu cần được thao tác.
D (điểm đến): Điểm đến của dữ liệu sau khi thao tác (số hiệu thiết bị)
P: Hướng dẫn cần được thực hiện trên cạnh xung tăng (từ tắt đến bật)

Lưu ý:
Đối với các phép tính số thực có độ chính xác đơn, S1 và S2 trong biểu thức tính phải là số thực có độ chính xác đơn. Các số thực có độ chính xác duy nhất được lưu trong D.

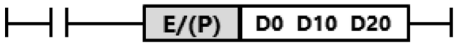
2.11.4 Hướng dẫn phép tính (phép nhân và chia)

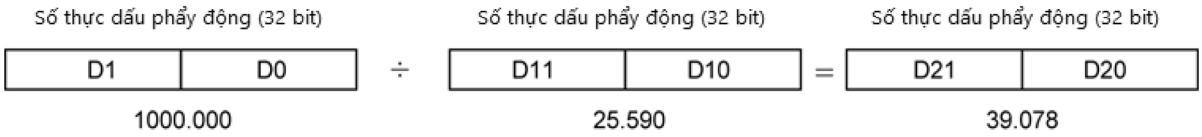
Ví dụ về ứng dụng của phép nhân





Ví dụ về ứng dụng của phép chia





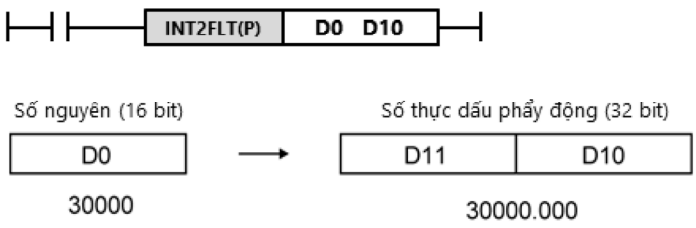
2.11.5 Hướng dẫn chuyển đổi giữa số nguyên và số thực

Biểu tượng hướng dẫn	Ví dụ về trình lập trình dạng thang	
INT2FLT (Chuyển đổi số nguyên thành số thực có độ chính xác đơn)	Số nguyên (16 bit) được chuyển đổi thành số thực (32 bit).  S (16 bit) được chuyển đổi và lưu trong D.	Số nguyên (32 bit) được chuyển đổi thành số thực (32 bit).  S (32 bit) được chuyển đổi và lưu trong D.
FLT2INT (Chuyển đổi số thực có độ chính xác đơn thành số nguyên)	Số thực (32 bit) được chuyển đổi thành số nguyên (16 bit).  S được chuyển đổi và lưu trong D (16 bit).	Số thực (32 bit) được chuyển đổi thành số nguyên (32 bit).  S được chuyển đổi và lưu trong D (32 bit).

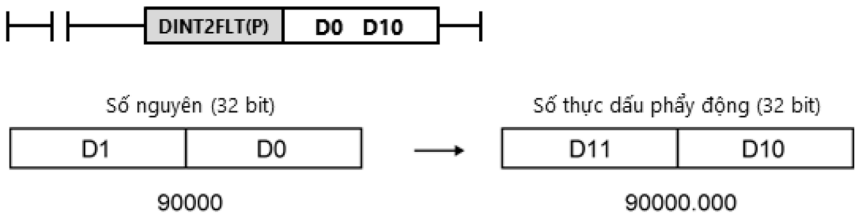
S (nguồn): Dữ liệu trước khi thao tác (không đổi, số hiệu thiết bị)
D (điểm đến): Điểm đến của dữ liệu sau khi thao tác (số hiệu thiết bị)

2.11.5 Hướng dẫn chuyển đổi giữa số nguyên và số thực

Ví dụ về ứng dụng của hướng dẫn chuyển đổi số nguyên (16 bit) thành số thực (32 bit)

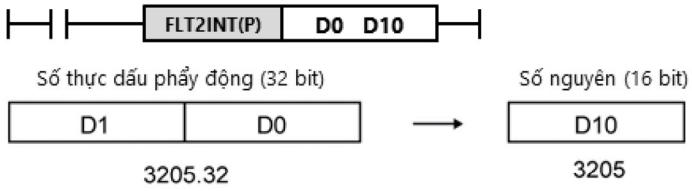


Ví dụ về ứng dụng của hướng dẫn chuyển đổi số nguyên (32 bit) thành số thực (32 bit)

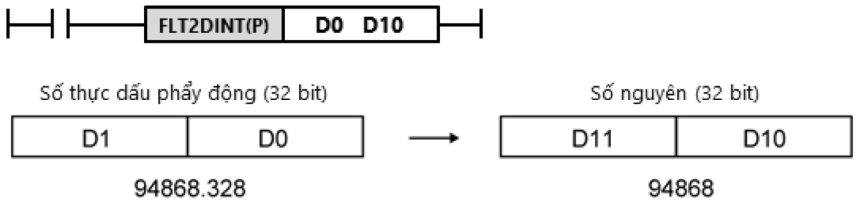


2.11.5 Hướng dẫn chuyển đổi giữa số nguyên và số thực

Ví dụ về ứng dụng của hướng dẫn chuyển đổi số thực (32 bit) thành số nguyên (16 bit)



Ví dụ về ứng dụng của hướng dẫn chuyển đổi số thực (32 bit) thành số nguyên (32 bit)



2.11.6 Sử dụng các nhãn thể hiện số thực

Để sử dụng các nhãn cho số thực, hãy thiết lập kiểu dữ liệu thành "Single Precision" (Độ chính xác đơn) hoặc "Double Precisition" (Độ chính xác kép) trên cửa sổ để thiết lập nhãn.

Label Name	Data Type
eData	FLOAT [Single Precision]
leData	FLOAT [Double Precision]

2.12 Tóm tắt chương này

Trong chương này, bạn đã tìm hiểu về:

- Thông số kỹ thuật bit của thiết bị từ
- Đặc tính cạnh xung tăng hoặc giảm cho các tiếp điểm
- Bộ hẹn giờ có khả năng nhớ các sự kiện
- Đặc tính đơn vị đo của bộ hẹn giờ
- Thanh ghi chỉ mục
- Dãy
- Cấu tạo
- Khóa
- Thanh ghi tập tin
- Rơ-le đặc biệt, thanh ghi đặc biệt
- Tính toán bằng các số thực

Các điểm quan trọng

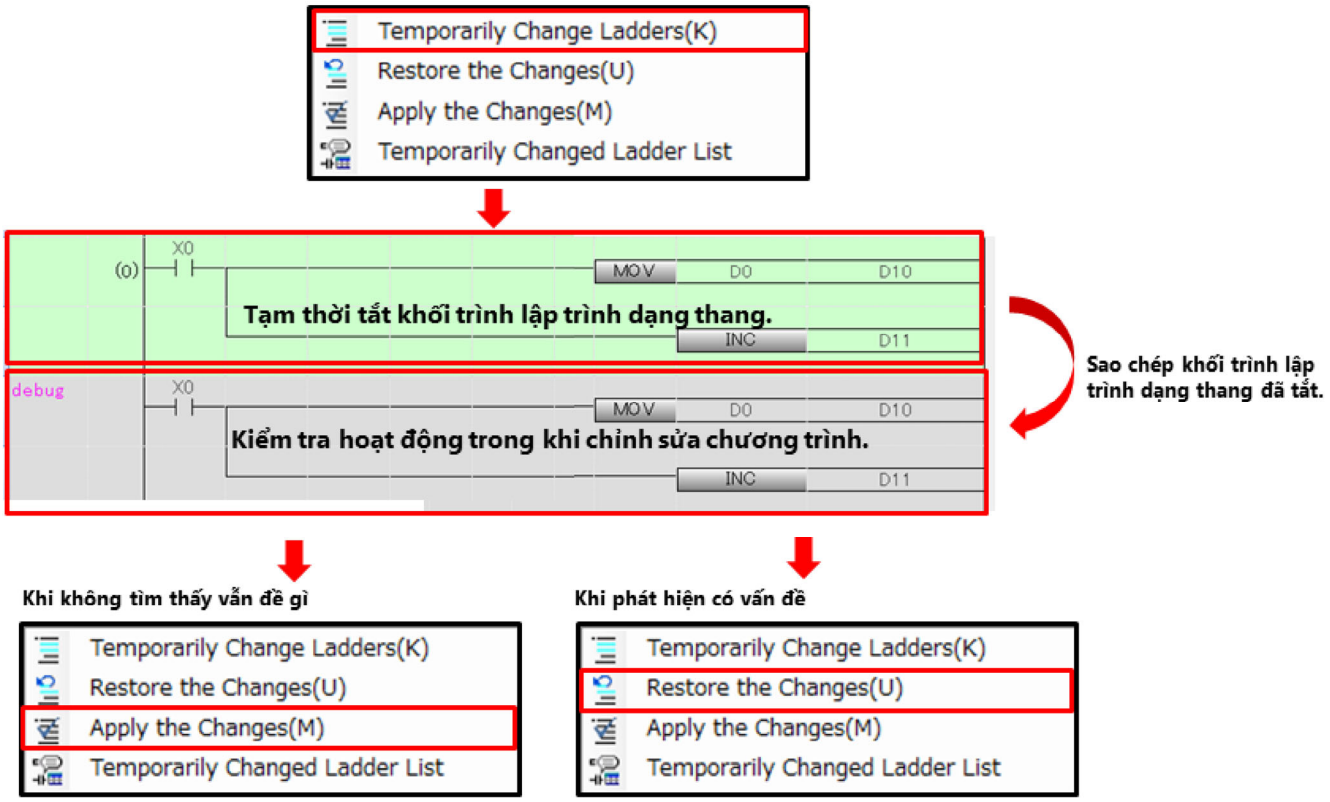
Bộ hẹn giờ có khả năng nhớ các sự kiện	Thời gian đo được được giữ lại ngay cả khi cuộn cảm tắt và phép đo tiếp tục khi cuộn cảm bật lại.
Đơn vị đo của bộ hẹn giờ	Đơn vị đo của bộ hẹn giờ có thể được thay đổi trong thông số.
Thanh ghi chỉ mục	Nhiều thiết bị có thể được mô tả trong một lô.
Dãy	Có thể xử lý nhiều giá trị bằng một tên nhãn.
Cấu tạo	Có thể xử lý nhiều nhãn có liên quan bằng một tên nhãn.
Khóa	• Các giá trị thiết bị đã khóa được giữ lại khi mô-đun CPU dừng hoạt động • Các giá trị thiết bị đã khóa bị xóa do hoạt động của bộ nhớ CPU hoặc sử dụng hướng dẫn chương trình
Thanh ghi tập tin	• So sánh với thanh ghi dữ liệu, thanh ghi tập tin có thể xử lý số lượng dữ liệu lớn hơn • Các giá trị thiết bị được giữ lại khi mô-đun CPU dừng hoạt động • Có thể xóa các giá trị thiết bị bằng hoạt động bộ nhớ CPU hoặc ghi 0 vào phạm vi thiết bị
Rơ-le đặc biệt, thanh ghi đặc biệt	Trạng thái bên trong của mô-đun CPU, như thông tin chuẩn đoán và thông tin hệ thống, đã được lưu trong những thiết bị này.
Số thực	• Sử dụng ít nhất hai thiết bị từ (32 bit) • Hướng dẫn phép tính riêng được cung cấp • Không thể sử dụng chung số nguyên và số thực trong một hoạt động

Chương 3 Khắc phục lỗi hiệu quả

Chương này mô tả các chức năng của GX Works3 để khắc phục lỗi hiệu quả.

3.1 Thay đổi tạm thời phạm vi chương trình

Khi chỉnh sửa phạm vi chương trình rộng để khắc phục lỗi, sẽ rất khó hoàn tác tất cả các thay đổi.
Bằng cách tạm thời tắt khóa trình lập trình dạng thang mong muốn, người dùng có thể sử dụng bản sao của chương trình để khắc phục lỗi mà không cần thay đổi bản gốc.
(Chức năng Temporarily change ladders (Tạm thời thay đổi trình lập trình dạng thang))
Sau khi thay đổi khối trình lập trình dạng thang và kiểm tra hoạt động với chức năng này, thay đổi được áp dụng nếu không có vấn đề gì hoặc thay đổi được khôi phục nếu có vấn đề.



Khi không tìm thấy vấn đề gì

Khi phát hiện có vấn đề

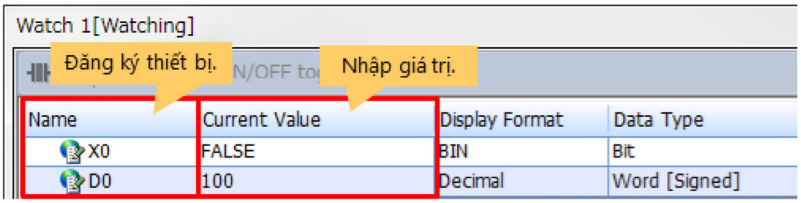
3.2 Kiểm tra hoạt động khi thay đổi giá trị thiết bị

Khi chạy chương trình đã tạo lập, trạng thái bật/tắt của các thiết bị bit và giá trị được lưu trong thiết bị từ có thể được hiển thị trong trình chỉnh sửa chương trình. (Chức năng Monitor (giám sát))
Với chức năng giám sát, người dùng có thể dễ dàng kiểm tra trạng thái hoạt động của chương trình.

Giá trị hiện tại của thiết bị có thể bị buộc phải thay đổi trong quá trình giám sát. (Thay đổi giá trị hiện tại)
Với chức năng thay đổi giá trị hiện tại, có thể thực hiện thay đổi mà không cần chỉnh sửa toàn bộ chương trình hay chạy nó trên hệ thống thực.

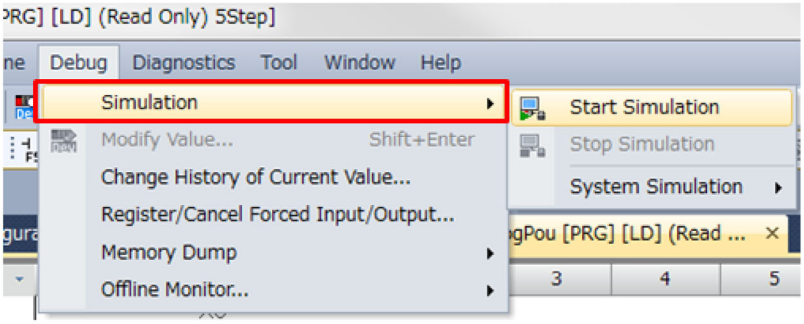
Trạng thái của thiết bị bit có thể được chuyển đổi trên trình chỉnh sửa chương trình.

Sử dụng cửa sổ "Watch" (Xem), có thể đăng ký các thiết bị từ cần được giám sát và có thể thay đổi giá trị hiện tại của chúng.



3.3 Giả lập hoạt động của chương trình

Nếu một chương trình mới được tạo lập được chạy trên hệ thống thực, có thể xảy ra lỗi không mong muốn.
Có thể giả lập hoạt động của chương trình mà không cần sử dụng một bộ điều khiển khả trình thực tế.
(Chức năng Simulation (giả lập))



Với chức năng giả lập, có thể kiểm tra hoạt động của chương trình như thể chương trình đang chạy trên bộ điều khiển khả trình thực tế.

3.4 Tóm tắt chương này

Trong chương này, bạn đã tìm hiểu về:

- Tạm thời thay đổi khối trình lập trình dạng thang
- Giám sát chương trình và thay đổi giá trị hiện tại
- Giả lập chương trình

Các điểm quan trọng

Tạm thời thay đổi khối trình lập trình dạng thang	Có thể tạm thời tắt khối trình lập trình dạng thang và có thể sử dụng bản sao chương trình để khắc phục lỗi mà không cần thay đổi chương trình gốc.
Giám sát	• Có thể trực quan hóa cách chương trình chạy. • Có thể kiểm tra hoạt động của chương trình khi buộc thay đổi giá trị thiết bị hiện tại.
Giả lập	Có thể giả lập hoạt động của chương trình mà không cần sử dụng một bộ điều khiển khả trình thực tế.

3.5 Tóm tắt khóa học này

Đây là phần kết thúc khóa học e-learning.

Sau đây là phần tóm tắt khóa học này.

- **Lập trình hiệu quả**
  - Chỉ định số thứ tự I/O cố định cho các mô-đun để dễ dàng sử dụng các chương trình tron các hệ thống khác nhau
  - Điều chỉnh các khu vực bộ nhớ theo như hiện trạng sử dụng thiết bị
  - Sử dụng các nhãn để làm cho việc lập trình trở nên dễ dàng hơn và thao tác dễ hiểu hơn
  - Thêm các chú thích để nâng cao khả năng có thể đọc của chương trình
- **Lập trình nâng cao**
  - Sử dụng bộ hẹn giờ có khả năng nhớ các sự kiện để giữ lại thời gian đã đo
  - Sử dụng thanh ghi chỉ mục, các dãy hoặc cấu trúc để xử lý chung các giá trị
  - Sử dụng chức năng khóa và thanh ghi tập tin để giữ lại trạng thái thiết bị
  - Rơ-le và thanh ghi đặc biệt lưu trữ trạng thái bên trong của mô-đun CPU được cung cấp
  - Số thực được thể hiện qua hai hoặc bốn thiết bị từ. Không thể sử dụng chung số nguyên và số thực trong một hoạt động
- **Gỡ rối hiệu quả**

  Người dùng có thể thực hiện các hoạt động sau bằng cách sử dụng GX Works3:
  - Gỡ rối chương trình mà không cần thay đổi chương trình gốc
  - Trực quan hóa cách chương trình chạy
  - Giả lập hoạt động của chương trình

Để thực hiện bước tiếp theo, hãy tham gia các khóa học sau về "cấu trúc", chia chương trình thành các lớp và thành phần để bạn có thể dễ dàng sử dụng.

- Lập trình hiệu quả
- Lập trình hiệu quả (Thực hành) (sẽ được phát hành)

Bài Kiểm tra

Bài kiểm tra cuối khóa

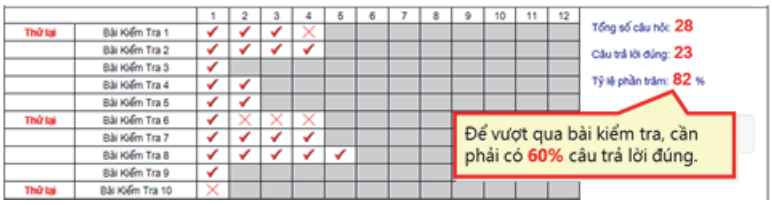
Vì bạn đã hoàn thành tất cả các bài học về của khóa học Ứng dụng lập trình (Bản vẽ nguyên lý mạch điện/Sê-ri MELSEC iQ-R), bạn đã sẵn sàng tham gia bài kiểm tra cuối khóa. Nếu bạn không rõ về bất cứ chủ đề nào được trình bày, vui lòng nhân cơ hội này xem xét lại các chủ đề đó.

Có tổng cộng 14 câu hỏi (35 mục) trong Bài kiểm tra cuối khóa này.

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

Kết quả điểm số

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



Bài Kiểm tra

Bài kiểm tra cuối khóa 1

Câu nào sau đây là đúng về phân giao mô-đun của số thứ tự I/O?

H1

- Số thứ tự I/O có thể được chỉ định thủ công cho mỗi mô-đun để chương trình không cần được sửa đổi khi cấu hình mô-đun được thay đổi
- Không thể thay đổi số thứ tự I/O được chỉ định tự động

Bài Kiểm tra **Bài kiểm tra cuối khóa 2**

Câu nào sau đây là đúng về thiết lập điểm thiết bị?

H1

- Phải chỉ định ít nhất một điểm cho mỗi thiết bị ngay cả khi thiết bị đó không được sử dụng.
- Có thể chỉ định các điểm theo số điểm được sử dụng.

Bài Kiểm tra **Bài kiểm tra cuối khóa 3**

Câu nào sau đây là đúng về các nhãn? (Câu trả lời trắc nghiệm)

H1

- [ ] Việc sử dụng các nhãn giúp xác định mục tiêu xử lý và làm cho việc lập trình dễ dàng hơn
- [ ] Các nhãn thể hiện các tín hiệu mô-đun và các giá trị thiết lập được cung cấp
- [ ] Có thể thêm các chú thích vào các thành tố để cải thiện khả năng có thể đọc được của chương trình
- [ ] Vì các hằng số có thể được chỉ định cho các nhãn, nên các giá trị có thể được thay đổi mà không cần sửa đổi chương trình

Bài Kiểm tra

Bài kiểm tra cuối khóa 4

Hoàn thành văn bản sau mô tả bộ hẹn giờ có khả năng nhớ các sự kiện.

Bộ hẹn giờ có khả năng nhớ các sự kiện bắt đầu phép đo khi bật (H1) (cuộn cảm (H2)).
Bộ hẹn giờ có khả năng nhớ các sự kiện lưu giữ lại thời gian đo được ngay cả khi điều kiện đầu vào (H3) và tiếp tục phép đo từ giá trị được lưu giữ khi điều kiện đầu vào (H4) lại.
Bộ hẹn giờ có khả năng nhớ các sự kiện hết thời gian cho phép khi thời gian đo được đạt đến giá trị thiết lập, tại điểm

H1 -- Select --

H2 -- Select --

H3 -- Select --

H4 -- Select --

H5 -- Select --

H6 -- Select --

Bài Kiểm tra

Bài kiểm tra cuối khóa 5

Hoàn thành chương trình điều khiển thực hiện việc xử lý sau.

• Sử dụng bộ hẹn giờ có khả năng nhớ các sự kiện (ST0) để đo thời gian bật của tín hiệu đầu vào X0 hoặc X1
• Khi thời gian bật X0 hoặc X1 đạt đến 30 giây, bật cuộn cảm Y70 và đèn chỉ báo hết thời gian cho phép
• Khi X2 bật, tắt tiếp điểm của bộ hẹn giờ có khả năng nhớ các sự kiện (ST0) và xóa thời gian đã đo được (giá trị hiện tại)

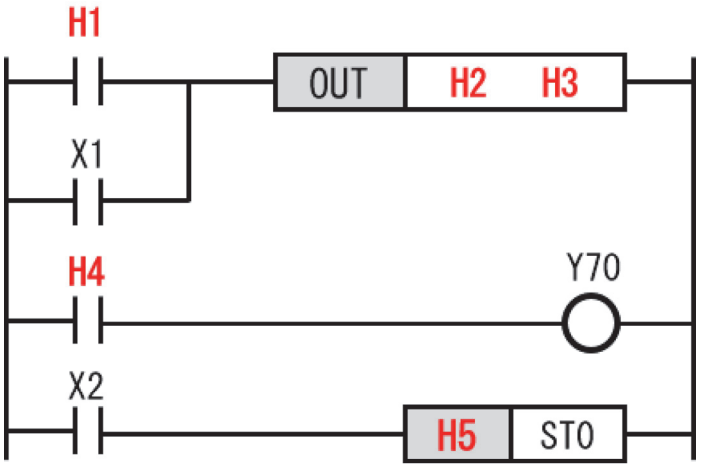
H1 -- Select --

H2 -- Select --

H3 -- Select --

H4 -- Select --

H5 -- Select --



[+]

Bài Kiểm tra

Bài kiểm tra cuối khóa 6

Lựa chọn giá trị đã lưu trong thanh ghi dữ liệu D20 khi X0 bật theo mỗi điều kiện sau trong chương trình điều khiển bên dưới.

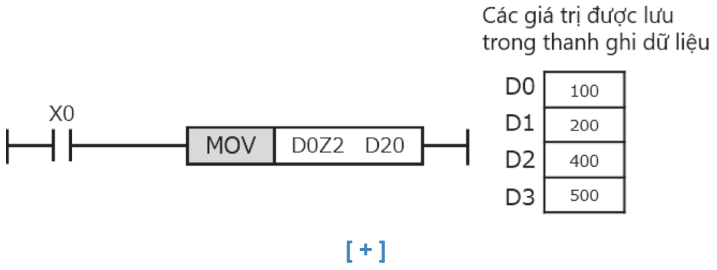
H1) Khi giá trị được lưu trong Z2 là "0"
H2) Khi giá trị được lưu trong Z2 là "1"
H3) Khi giá trị được lưu trong Z2 là "2"

H1 -- Select --

H2 -- Select --

H3 -- Select --

H4 -- Select --



Các giá trị được lưu trong thanh ghi dữ liệu

D0	100
D1	200
D2	400
D3	500

[+]

Bài Kiểm tra

Bài kiểm tra cuối khóa 7

Câu nào sau đây là đúng về cách xác định một thành tố của một dãy?

H1

- Thêm một số thành tố vào cuối tên nhãn
- Trực tiếp xác định số hiệu thiết bị

Bài Kiểm tra **Bài kiểm tra cuối khóa 8**

Câu nào sau đây là sai về các cấu trúc?

H1

- Các cấu trúc được sử dụng để tổ chức và lưu trữ chung các điều kiện và thông số kỹ thuật liên quan đến các chủ thể hoặc vấn đề thực tế
- Bằng cách sử dụng các cấu trúc xử lý khối lượng lớn dữ liệu có thể được mô tả chính xác
- Các thành phần được xác định trong một cấu trúc phải có cùng kiểu dữ liệu

Bài Kiểm tra **Bài kiểm tra cuối khóa 9**

Câu nào sau đây là đúng về chức năng chốt?

H1

- Các thiết bị thường có chức năng lưu giữ lại các giá trị
- Cần thiết lập tham số để lưu giữ lại các giá trị bằng cách sử dụng phần mềm kỹ thuật

Bài Kiểm tra

Bài kiểm tra cuối khóa 10

Hoàn thành văn bản sau mô tả thanh ghi tập tin.

Thanh ghi tập tin là một thiết bị từ (kiểu dữ liệu 32 bits) được sử dụng để mở rộng thanh ghi dữ liệu (D) và biểu tượng thiết bị là (H1).
Thanh ghi tập tin có khả năng (H2) so với thanh ghi dữ liệu và dữ liệu đã được lưu giữ (H3) ngay cả khi hệ thống bị tắt nguồn hoặc mô đun CPU được cài lại.

H1 -- Select --

H2 -- Select --

H3 -- Select --

H4 -- Select --

Bài Kiểm tra | Bài kiểm tra cuối khóa 11

Câu nào sau đây là đúng về rơ-le đặc biệt và thanh ghi đặc biệt?

H1

- Hiện trạng bên trong của mô đun CPU đã được lưu trong rơ-le đặc biệt và thanh ghi đặc biệt, và những thiết bị này được sử dụng như những điều kiện xác định trong chương trình điều khiển.
- Các chức năng đặc biệt có thể được gán tự do cho rơ-le đặc biệt và thanh ghi đặc biệt.

Bài Kiểm tra

Bài kiểm tra cuối khóa 12

Hoàn thành văn bản sau mô tả các số thực (độ chính xác duy nhất).

• Một số thực sử dụng (H1) thiết bị từ (kiểu dữ liệu 32 bits) và được lưu trong dung lượng bộ nhớ (H2) bit.
• Dữ liệu giá trị số của số thực được gọi là (H3). Ví dụ: 2,035 được mô tả là (H4) trong chương trình điều khiển.
• Các số nguyên và số thực (H5) được kết hợp trong một hướng dẫn tác nghiệp xử lý các số thực.

H1 -- Select --

H2 -- Select --

H3 -- Select --

H4 -- Select --

H5 -- Select --

Bài Kiểm tra

Bài kiểm tra cuối khóa 13

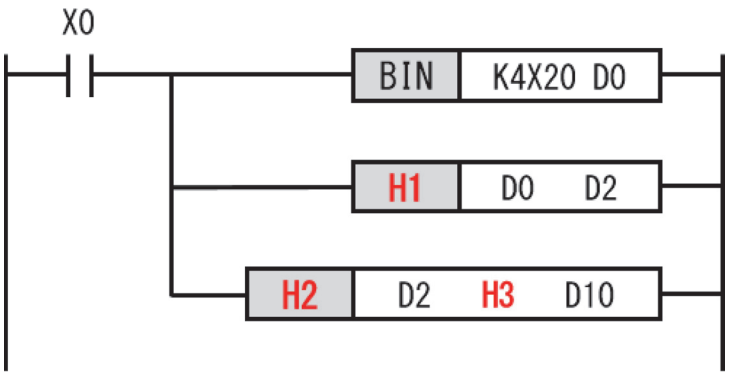
Hoàn thành chương trình điều khiển thực hiện việc xử lý sau.

- Đọc dữ liệu từ X20 đến X2F (dữ liệu BCD) khi X0 được bật và lưu nó trong D0.
- Chuyển đổi giá trị trong D0 thành số thực và lưu giá trị đã được chuyển đổi trong D2.
- Nhân giá trị trong D2 với 3,14 và lưu kết quả trong D10.

H1 -- Select --

H2 -- Select --

H3 -- Select --



[+]

Bài Kiểm tra | Bài kiểm tra cuối khóa 14

Câu nào sau đây là đúng về việc gỡ rối chương trình điều khiển?

H1

- Có thể giả lập hoạt động của chương trình một cách an toàn bằng cách sử dụng chức năng của phần mềm kỹ thuật.
- Để gỡ rối chương trình, việc gỡ rối phải được thực hiện trong hệ thống thực.

Bài Kiểm tra

Điểm kiểm tra

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

		1	2	3	4	5	6	7	8	9	10
	Bài kiểm tra cuối khóa 1	✓									
	Bài kiểm tra cuối khóa 2	✓									
	Bài kiểm tra cuối khóa 3	✓									
	Bài kiểm tra cuối khóa 4	✓	✓	✓	✓	✓	✓				
	Bài kiểm tra cuối khóa 5	✓	✓	✓	✓	✓					
	Bài kiểm tra cuối khóa 6	✓	✓	✓	✓						
	Bài kiểm tra cuối khóa 7	✓									
	Bài kiểm tra cuối khóa 8	✓									
	Bài kiểm tra cuối khóa 9	✓									
	Bài kiểm tra cuối khóa 10	✓	✓	✓	✓						
	Bài kiểm tra cuối khóa 11	✓									
	Bài kiểm tra cuối khóa 12	✓	✓	✓	✓	✓					
	Bài kiểm tra cuối khóa 13	✓	✓	✓							
	Bài kiểm tra cuối khóa 14	✓									

Tổng số câu hỏi: 35

Câu trả lời đúng: 35

Tỷ lệ phần trăm: 100 %

Xóa

Bạn đã hoàn thành khóa học Ứng dụng lập trình (Bản vẽ nguyên lý mạch điện/Sê-ri MELSEC iQ-R).

Cảm ơn bạn đã tham gia khóa học này.

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

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

Xem lại

Đóng