

**CLP**

**Aplicativos de programação  
(Diagrama ladder/série MELSEC  
iQ-R)**

Este curso é destinado a usuários que possuam conhecimento básico do CLP série MELSEC iQ-R e desejam aprender mais sobre programação.

L(CTS)00689POR

Este curso é destinado para usuários que concluíram a Introdução à programação (Diagrama ladder) ou que possuem conhecimentos equivalentes. Este curso oferece conhecimentos sobre programação e resolução de problemas para CLP MELSEC iQ-R, uso avançado de dispositivos e programação de label.

Como pré-requisitos para este curso, você já deve ter concluído os seguintes cursos ou possuir conhecimentos equivalentes.

- Noções básicas da série MELSEC iQ-R
- Introdução à programação

O conteúdo do curso é explicado a seguir.

#### Capítulo 1 - Programação eficiente

Métodos e definições de programação eficiente

#### Capítulo 2 - Programação avançada





Uso avançado de dispositivos e programação de labels

#### Capítulo 3 - Resolução de problemas

Funções do software de engenharia usado para a resolução de problemas

#### Teste Final

Nota de aprovação: 60% ou superior é necessário

Ir para a próxima página		Ir para a próxima página.
Voltar para a página anterior		Voltar para a página anterior.
Mover-se para a página desejada		O "Índice" será exibido, permitindo que você navegue até a página desejada.
Sair do curso		Sair do curso.

**Precauções de segurança**

Quando você estiver aprendendo a operar os produtos reais, leia cuidadosamente as precauções de segurança dos respectivos manuais.

**Precauções neste curso**

As telas exibidas da versão de software que você utiliza podem ser diferentes das apresentadas neste curso. Este curso utiliza a seguinte versão do software:

- GX Works3 Versão 1.044W

## Capítulo 1 Programação eficiente

Este capítulo descreve as definições do software de engenharia para a programação eficiente e as noções básicas da programação de labels.

- 1.1 Fácil utilização de programas em sistemas diferentes
- 1.2 Ajuste da área da memória de acordo com o estado de uso do dispositivo
- 1.3 Utilização de nomes de etiqueta relacionados aos aplicativos
- 1.4 Melhora da legibilidade do programa

## 1.1 Fácil utilização de programas em sistemas diferentes

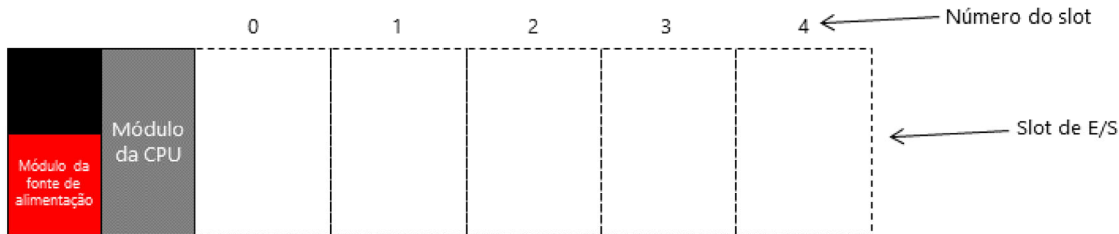
Esta seção descreve a atribuição eficiente do número de E/S para a fácil utilização de programas em sistemas diferentes.

### 1.1.1 Atribuição automática do número de E/S

Os números de E/S são atribuídos sequencialmente aos módulos instalados na unidade base a partir do slot mais próximo do módulo da CPU.

Os números de E/S são atribuídos em unidades de 16 pontos (0 a F).

O número de pontos disponível para a atribuição (a serem ocupados) varia de acordo com o tipo de módulo (16, 32, 64 e assim por diante).



No exemplo a seguir, cinco módulos de 16 pontos estão instalados.

	0	1	2	3	4	
						Número de pontos ocupados
Módulo da fonte de alimentação		16 pontos	16 pontos	16 pontos	16 pontos	
Módulo da CPU		00	10	20	30	40
		a	a	a	a	Número de E/S
		0F	1F	2F	3F	4F

## 1.1.1

### Atribuição automática do número de E/S

Quando os módulos com 16, 32 e 64 pontos ocupados são usados ao mesmo tempo, os números de E/S são atribuídos da seguinte forma:

		0	1	2	3	4
Módulo da fonte de alimentação	Módulo da CPU	16 pontos	32 pontos	64 pontos	32 pontos	16 pontos
		00	10	30	70	90
		a	a	a	a	a
		0F	2F	6F	8F	9F

Se há um slot vazio entre os módulos instalados, os números de E/S também são atribuídos ao slot vazio.

		0	1	2	3	4
Módulo da fonte de alimentação	Módulo da CPU	16 pontos	32 pontos	64 pontos	16 pontos	16 pontos
		00	10	30	70	80
		a	a	a	a	a
		0F	2F	6F	7F	8F

Slot vazio

Por padrão, os 16 pontos são atribuídos a um slot vazio. O número de pontos atribuídos pode ser alterado pela definição de parâmetro dentro do intervalo entre 0 e 1024 pontos (na unidade de 16 pontos).



## 1.1.1

### Atribuição automática do número de E/S

Os números de E/S de unidade base de expansão são atribuídos automaticamente seguindo o último número de E/S da unidade base principal.



## 1.1.2

## Atribuição fixa do número de E/S

Quando os números de E/S são atribuídos manualmente, os números de E/S atribuídos são fixados e inalterados mesmo se a configuração do módulo é alterada. Isso significa que o mesmo programa de controle pode ser utilizado para o mesmo controle independentemente da configuração do módulo.

### Atribuição automática

Antes dos módulos serem adicionados

Módulo da fonte de alimentação	Módulo da CPU	Módulo de entrada	Módulo de saída	Módulo de função inteligente	
		64 pontos	64 pontos	16 pontos	
		X00 a X3F	Y40 a Y7F	X/Y80 a X/Y8F	

Após os módulos serem adicionados  
(Um módulo de entrada de 32 pontos e um módulo de saída de 16 pontos são adicionados.)

Módulo da fonte de alimentação	Módulo da CPU	Módulo de entrada	Módulos adicionados		Módulo de função inteligente	
		64 pontos	Módulo de entrada 32 pontos	Módulo de saída 64 pontos	Módulo de saída 16 pontos	16 pontos
		X00 a X3F	X40 a X5F	Y60 a Y9F	YA0 a YAF	X/YB0 a X/YBF

Os números de E/S são reatribuídos após a adição dos módulos.

## 1.1.2 Atribuição fixa do número de E/S

### Atribuição manual para números de E/S fixos

Antes dos módulos serem adicionados

		Módulo de entrada	Módulo de saída	Módulo de função inteligente	
Módulo da fonte de alimentação	Módulo da CPU	64 pontos	64 pontos	16 pontos	
		X00 a X3F	Y40 a Y7F	X/Y80 a X/Y8F	

Após os módulos serem adicionados  
(Um módulo de entrada de 32 pontos e um módulo de saída de 16 pontos são adicionados.)

		Módulo de entrada	Módulo de entrada	Módulo de saída	Módulo de saída	Módulo de função inteligente	
Módulo da fonte de alimentação	Módulo da CPU	64 pontos	32 pontos	64 pontos	16 pontos	16 pontos	
		X00 a X3F	X90 a XAF	Y40 a Y7F	Y80 a YBF	X/Y80 a X/Y8F	

Módulos adicionados

Os mesmos números de E/S são atribuídos independentemente da adição de módulos.

Uma vez que os números de E/S dos módulos existentes permanecem inalterados, apenas os programas relacionados aos módulos adicionados devem ser adicionados ou modificados.

### 1.1.3

### Atribuição automática do número de E/S usando o diagrama de configuração do módulo

A configuração do módulo pode ser definida usando o diagrama de configuração do módulo do software de engenharia, MELSOFT GX Works3.

Selecione um nome de modelo do módulo e arraste e solte em um slot para colocar o módulo ali.

Os números de E/S são atribuídos sequencialmente aos módulos a partir do mais próximo ao módulo da CPU, conforme exibido no diagrama.

Selecione um módulo já colocado para visualizar o número de E/S inicial do módulo.

The screenshot displays the 'Module Configuration' window in MELSOFT GX Works3. On the left, a rack configuration diagram shows slots labeled 'POW', 'CPU 0', '1', '2', '3', and '4'. A red box highlights slot '1', and a red arrow points from a yellow callout box above it containing the text 'Arraste e solte um nome de modelo do módulo.' (Drag and drop a module model name). On the right, the 'Element Selection' panel is open, showing a list of modules. The 'RY42NT2P' module is highlighted with a red box. A red arrow points from this box to the 'Start XY' field in the 'Input the Configuration Detailed Information' section, which contains the value '0040'. A yellow callout box below this field contains the text 'Número de E/S inicial do módulo selecionado' (Initial I/O number of the selected module).

Module Model	Points	Type
RY40PT5B(S2S)	16	points(Source type)
RY40PT5P	16	points(Source type)
RY41NT2H	32	points(Sink type/High-Speed)
RY41NT2P	32	points(Sink type)
RY41PT1P	32	points(Source type)
RY41PT2H	32	points(Source type/High-Speed)
<b>RY42NT2P</b>	<b>64</b>	<b>points(Sink type)</b>
RY42PT1P	64	points(Source type)

Field	Value
Start XY	0040
Points	64 Points

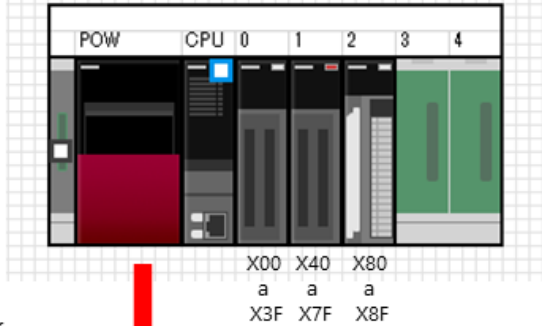
## 1.1.4

### Atribuição manual do número de E/S usando o diagrama de configuração do módulo

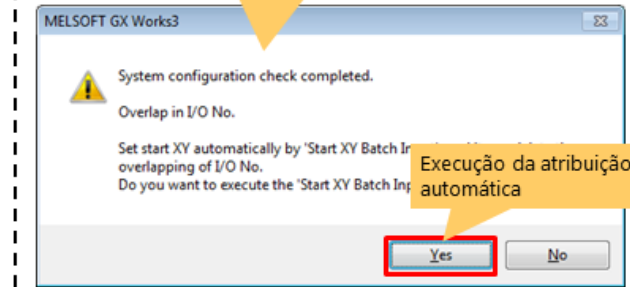
O exemplo a seguir descreve como atribuir números de E/S manualmente usando o diagrama de configuração do módulo do GX Works3.

Quando a configuração do módulo é alterada ao adicionar um novo módulo ao diagrama de configuração do módulo, o número de E/S do módulo adicionado se sobrepõe aos números existentes se a posição dos módulos existentes é alterada. Edite os números de E/S de maneira que eles não se sobreponham e determine a configuração do módulo. Uma mensagem de erro é exibida se a configuração do módulo é determinada com os números de E/S sobrepostos. Nesse momento, a atribuição automática pode ser executada na janela de mensagens de erro exibida.

Antes dos módulos serem adicionados

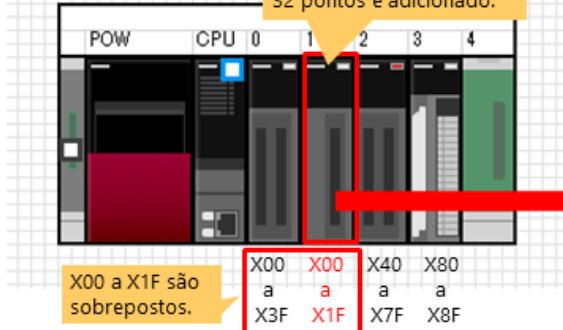


Uma mensagem de erro é exibida se a configuração do módulo é determinada com os números de E/S sobrepostos.



Após os módulos serem adicionados

Um módulo de entrada de 32 pontos é adicionado.



Altere o número de E/S inicial do módulo de entrada de 32 pontos de 0000 para 0090.

Captura de tela de uma janela de configuração de módulo. O texto da janela é:

Input the Configuration Detailed Information

RX41C4	
Start XY	0090
Points	32 Points

Determinação da configuração do módulo

## 1.2

## Ajuste da área da memória de acordo com o estado de uso do dispositivo

### 1.2.1

### Definições da área da memória do dispositivo/etiqueta

O número de pontos do dispositivo utilizado para os módulos da CPU varia de acordo com o tipo de módulo da CPU. O número inicial dos pontos do dispositivo é atribuído com base na capacidade da área do dispositivo do módulo da CPU.

A capacidade da área do dispositivo da R04CPU é de 40 mil words.

Reduzir as áreas não utilizadas aumenta o número de pontos do dispositivo a partir do valor inicial.

A imagem a seguir exibe a janela "Device/Label Memory Area Setting" (Definição da área da memória do dispositivo/etiqueta) como parâmetro para o ajuste da área da memória.

A capacidade da área do dispositivo aumenta quando a capacidade da área da etiqueta ou da área de armazenamento de arquivo é reduzida.

Além disso, a capacidade de toda a área da memória do dispositivo/etiqueta pode ser expandida usando um cassete SRAM estendido.

Item	Setting
<b>Device/Label Memory Area Setting</b>	
Extended SRAM Cassette Setting	Not Mounted
Device/Label Memory Area Capacity Setting	
Device Area	
Device Area Capacity	40 K Word
Label Area	
Label Area Capacity	30 K Word
Latch Label Area Capacity	2 K Word
File Storage Area Capacity	128 K Word
Device/Label Memory Configuration Confirmation	<Confirmation>
Device/Label Memory Area Detailed Setting	
Device Setting	<Detailed Setting>
Latch Type Setting of Latch Type Label	Latch (1)

Capacidade da área do dispositivo

## 1.2.2

## Definições do dispositivo

O número de pontos do dispositivo atribuídos a cada dispositivo pode ser alterado na janela "Device Setting" (Definição do dispositivo).

O valor inicial de alguns dispositivos é 0 pontos. Atribua o número de pontos ao usar esses dispositivos.

### Número de pontos do dispositivo:

Defina o número de pontos utilizado por cada dispositivo.

- Os valores iniciais são pré-atribuídos
- Os valores nas células brancas podem ser alterados
- Defina o número de pontos do dispositivo em unidades de 16 pontos
- 1K pontos significa 1024 pontos

Se o número total de pontos do dispositivo excede a capacidade do módulo da CPU, uma mensagem indicando a modificação da definição é exibida.

### Número total de pontos do dispositivo:

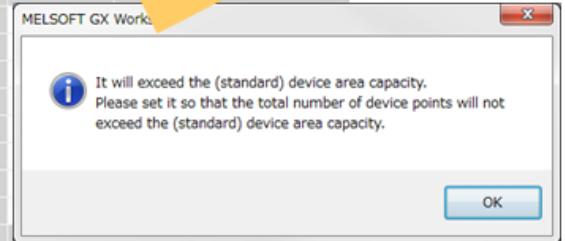
O número de pontos do dispositivo é convertido automaticamente em unidades de palavras.

### Número máximo de pontos do dispositivo = Capacidade do módulo da CPU

Por exemplo, a capacidade do módulo da CPU da R04CPU é de 40 mil palavras.

Item	Symbol	Points	Device	Latch (1)	Latch (2)
Input	X	12K			
Output	Y	12K	0 to 2047		
Internal Relay	M	12K	0 to 1228	No Setting	No Setting
Link Relay	B	8K	0 to 1FFF	No Setting	No Setting
Link Special Relay	SB	2K	0 to 7FF	No Setting	No Setting
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		
Long Timer	LT	1K	0 to 1023		
Protective Timer	ST	0			
		0			
		512	0 to 511		
		512	0 to 511		
		18K	0 to 18431		
		8K	0 to 1FFF		
		2K	0 to 7FF		
		8K	0 to 8191		
				No Setting	
Total Device			38.4K Word		
Total Word Device			34.5K Word		
Total Bit Device			62.0K Bit		

Janela "Device Setting" (Definição do dispositivo)

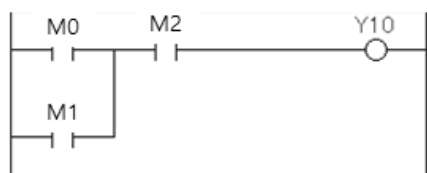


## 1.3 Utilização de nomes de etiqueta relacionados aos aplicativos

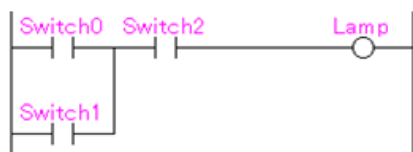
### 1.3.1 Vantagens do uso de labels

Os nomes dos dispositivos utilizados nos programas de controle devem consistir em uma letra e um número, como "M0" e "D5". Quando um nome de etiqueta é relacionado à aplicação, como "StartSwitch", o target do processamento fica evidente. Os nomes de label podem ser definidos livremente de acordo com as aplicações. Os usuários não precisam considerar os números do dispositivo como áreas em que as labels são utilizadas.

Programa que usa nomes do dispositivo



Programa que usa labels



As labels são classificadas nos dois seguintes tipos, de acordo com o escopo de utilização.

- **Labels globais**

Labels globais podem ser utilizadas para todos os programas em um projeto.

- **Labels locais**

Labels locais podem ser usadas apenas no programa para o qual estão registradas.

Ao usar labels para os dispositivos atuais (X, Y), os nomes dos dispositivos devem ser atribuídos a labels globais usando GX Works3.



## 1.3.2 Tipo de dados de labels

Um tipo de dados deve ser especificado para cada label para definir o intervalo dos valores a ser operado. Os tipos de dados incluem bit e inteiros, como mostrado abaixo.

Tipo de dados		Intervalo de dados
Tipo de bit		Estado on/off dos dispositivos de bits e estado verdadeiro/falso dos resultados de execução
Tipo inteiro	Palavra (sem sinal)	0 a 65.535
	Palavra (com sinal)	-32.768 a 32.767
	Double word (sem sinal)	0 a 4.294.967.295
	Double word (com sinal)	-2.147.483.648 a 2.147.483.647

Ao utilizar um tipo inteiro, selecione o tipo de palavra ou palavra dupla de acordo com o intervalo de dados e selecione o tipo com sinal ou sem sinal, de acordo com a necessidade de operar com valores negativos. Especifique o tipo de dados de uma label ao definir um nome de etiqueta usando GX Works3.

Label Name	Data Type
Switch0	Bit
Data0	Word [Unsigned]/Bit String [16-bit]
Data1	Double Word [Signed]

Especifique o valor do intervalo da label.

Janela de definição de label

### 1.3.3

## Nomes de labels que representam tipos de dados

Usar diferentes tipos de dados na fonte e no destino de transferência poderá causar um erro de conversão ou um resultado inesperado.

Confira abaixo um exemplo de programa de um caso semelhante.



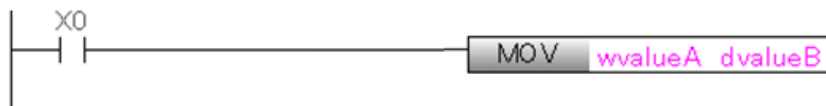
Valores de Double word não podem ser transferidos para uma label de tipo de word. Contudo, o tipo de dados não pode ser identificado pelo nome da label.

Portanto, os prefixos que representam o tipo de dados podem ser adicionados aos nomes de label para tornar os tipos de dados identificáveis visualmente.

Esse tipo de nomeação de label é chamado de notação húngara.

Tipo de dados		Intervalo de dados	Prefixo	Expansão de prefixo
Tipo de bit		Estado on/off dos dispositivos de bits e estado verdadeiro/falso dos resultados de execução	b	bit
Tipo inteiro	Palavra (sem sinal)	0 a 65.535	u	unsigned word (palavra sem sinal)
	Palavra (com sinal)	-32.768 a 32.767	w	signed word (palavra com sinal)
	Double word (sem sinal)	0 a 4.294.967.295	ud	unsigned double-word (Double word sem sinal)
	Double word (com sinal)	-2.147.483.648 a 2.147.483.647	d	signed double-word (Double word com sinal)

O exemplo de programa na parte superior desta página pode ser reescrito da seguinte forma usando a notação húngara:



Ao utilizar a notação húngara, inconsistências de tipo de dados podem ser identificadas no processo de escrita de um programa.

No restante do curso, os nomes de labels nos exemplos serão escritos na notação húngara.

## 1.3.4

## Utilizando labels prontas

Quando a configuração do módulo está definida no diagrama de configuração do módulo, as labels (labels do módulo) que representam os sinais do módulo ou os valores de definição que correspondem ao posicionamento de instalação do módulo são registrados automaticamente.

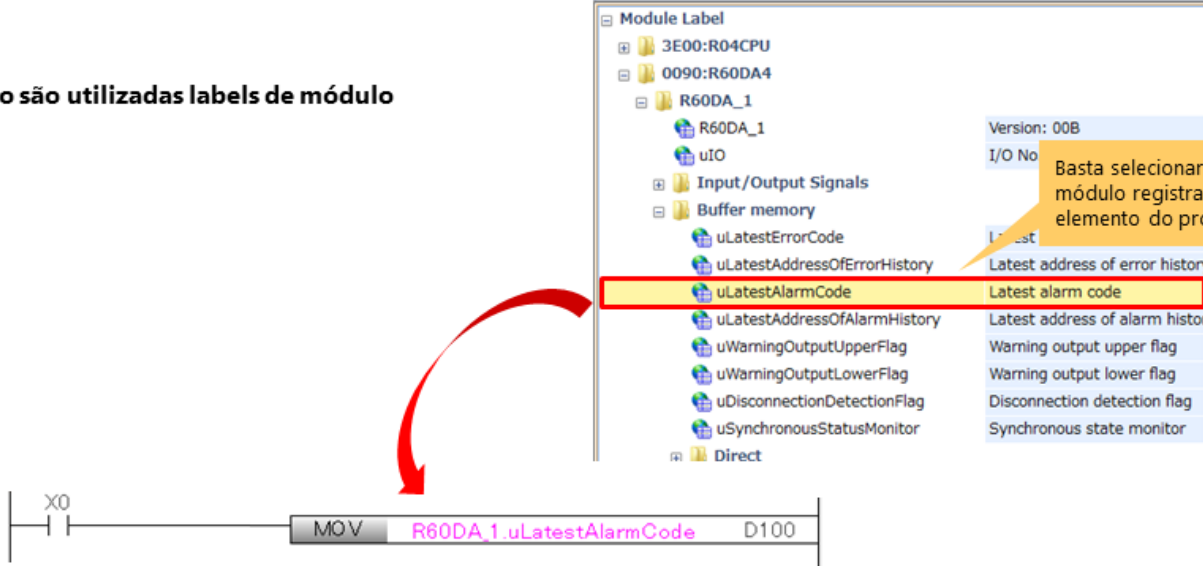
Para programar utilizando nomes do dispositivo, os números do dispositivo e os endereços da memória buffer que correspondem aos sinais devem ser verificados nos manuais, e isso leva tempo. O tempo de programação pode ser reduzido usando labels de módulo porque assim os usuários precisariam somente selecionar as labels na lista.

### Quando são utilizados nomes do dispositivo



Verifique e descreva a posição da instalação e o endereço da buffer memory.

### Quando são utilizadas labels de módulo



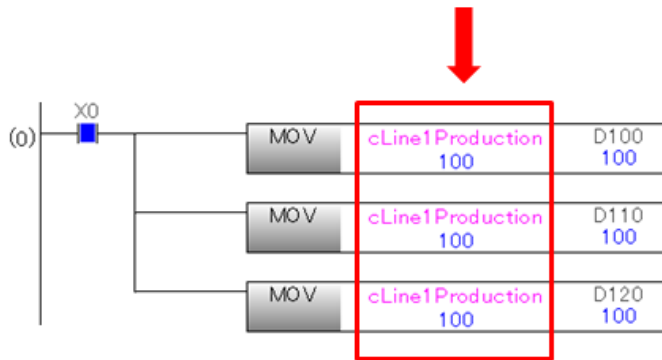
## 1.3.5

### Atribuindo constantes às labels

As constantes podem ser atribuídas às labels.

Quando as constantes são atribuídas às labels, os valores podem ser alterados sem a modificação do programa. A mesma constante utilizada para múltiplas labels pode ser alterada coletivamente.

Atribua a constante 100 à label "cLine1Production".



Atribua a constante 200.



Para atribuir uma constante a uma label, altere a classe que especifica a aplicação da label na janela de definição de labels.

Para labels locais, selecione "VAR\_CONSTANT".

Label Name	Data Type	Class	Initial Value	Constant
uData	Word [Unsigned]/Bit String [16-bit]	VAR_CONSTANT		100

Especifique a aplicação da label.

## 1.4

# Melhora da legibilidade do programa

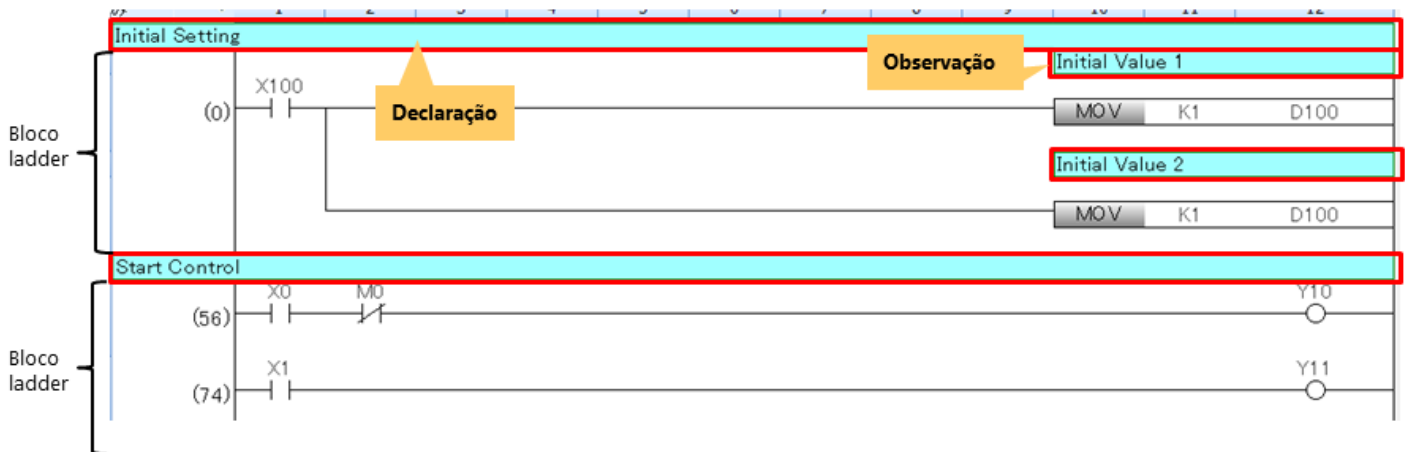
Comentários, como os detalhes do processamento e os nomes do dispositivo, podem ser adicionados a um programa. Comentários ajudam a explicar como um programa é executado.



O tipo de comentário pode ser selecionado de acordo com o elemento ou intervalo do programa.

No exemplo de programa acima, "comentários do dispositivo/label" foram adicionados para explicar as aplicações do dispositivo ou label e os tipos de dispositivos E/S conectados.

Além disso, o tipo de comentário inclui uma "declaração" que é adicionada ao bloco ladder para ajudar a explicar o fluxo de processamento e "observações" que ajudam a explicar os detalhes das bobinas e as instruções de aplicação.



Neste capítulo, você aprendeu:

- Atribuição do número de E/S
- Ajuste da área da memória
- Programação com labels
- Comentários nos programas

Pontos importantes

Atribuição do número de E/S	<ul style="list-style-type: none"><li>• Os números de E/S são atribuídos automaticamente aos slots a partir do mais próximo ao módulo da CPU</li><li>• Os programas podem ser usados em sistemas diferentes quando os números de E/S são atribuídos manualmente a módulos</li></ul>
Definições do dispositivo e definições da área da memória	<ul style="list-style-type: none"><li>• O número de pontos do dispositivo varia de acordo com o módulo da CPU</li><li>• O número de pontos do dispositivo pode ser aumentado se as áreas da memória não utilizadas são reduzidas</li><li>• O número de pontos do dispositivo de cada dispositivo pode ser alterado de acordo com o estado de utilização</li></ul>
Programação com labels	O target de processamento fica evidente quando labels são utilizadas
Comentários	Os detalhes e o fluxo de processamento tornam-se mais fáceis de entender quando comentários são adicionados

## Capítulo 2 Programação avançada

Este capítulo descreve o uso avançado de dispositivos e programação de labels.

- 2.1 Uso de dispositivo de palavra em unidades de bits
- 2.2 Ligando um dispositivo somente quando o estado de contato é alterado
- 2.3 Mantendo o tempo de medição do temporizador
- 2.4 Alterando a unidade de medida do temporizador
- 2.5 Operando múltiplos dispositivos (registro de índice)
- 2.6 Operando múltiplos valores (array)
- 2.7 Operando múltiplos valores (estrutura)
- 2.8 Mantendo o estado do dispositivo (retenção)
- 2.9 Mantendo o estado do dispositivo (registro de arquivos)
- 2.10 Usando dispositivos com funções e operações predeterminadas
- 2.11 Cálculo com números reais

## 2.1

# Uso de dispositivo de palavra em unidades de bits

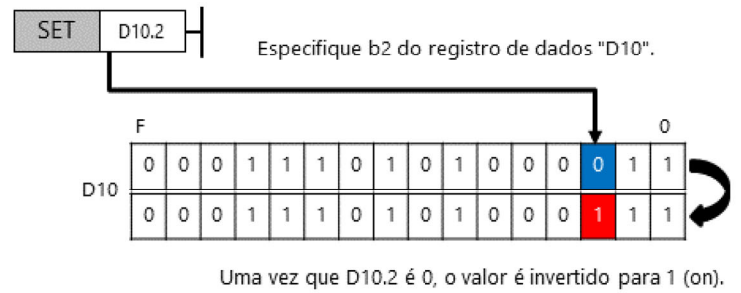
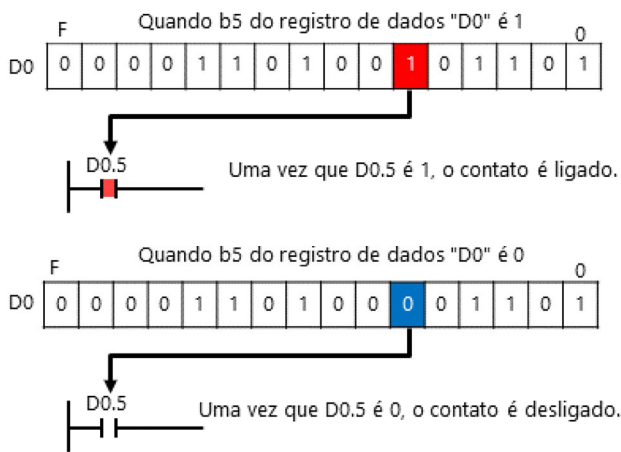
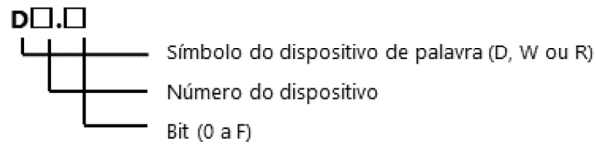
Dispositivos de palavra, como registro de dados, são geralmente utilizados em unidades de palavras, mas também podem ser utilizados em unidades de bits.

A unidade de bits é utilizada para especificar um bit específico no registro de dados (D).

Exemplo: registro de dados (D) 

0	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Formato de especificação de bit



Para utilizar labels, descreva como "uData.2" e "uData.5".



## 2.2

### Ligando um dispositivo somente quando o estado de contato é alterado

Um sinal que liga somente quando um scan na borda ascendente ou na borda descendente do contato pode ser especificado. Essa função é útil para controlar a subida e a descida pela condição de entrada.

#### Especificação da borda ascendente de contato

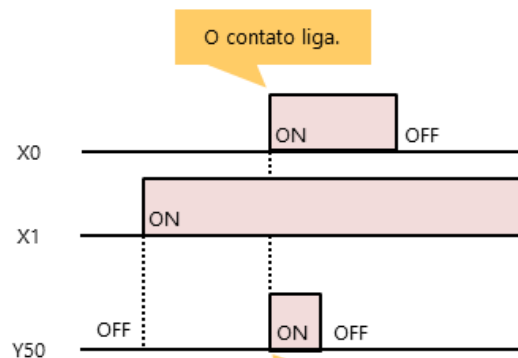


O sinal liga apenas para um scan, e o contato "X0" liga.

#### Especificação da borda descendente de contato

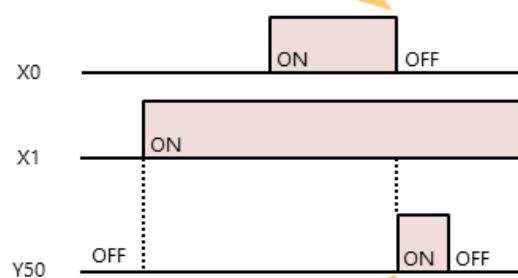


O sinal liga apenas para um scan, e o contato "X0" desliga.



Liga apenas para um scan

O contato desliga.



Liga apenas para um scan

## 2.3

# Mantendo o tempo de medição do temporizador

Esta seção descreve um dos dispositivos temporizadores, o temporizador retentivo, que pode manter o tempo de medição.

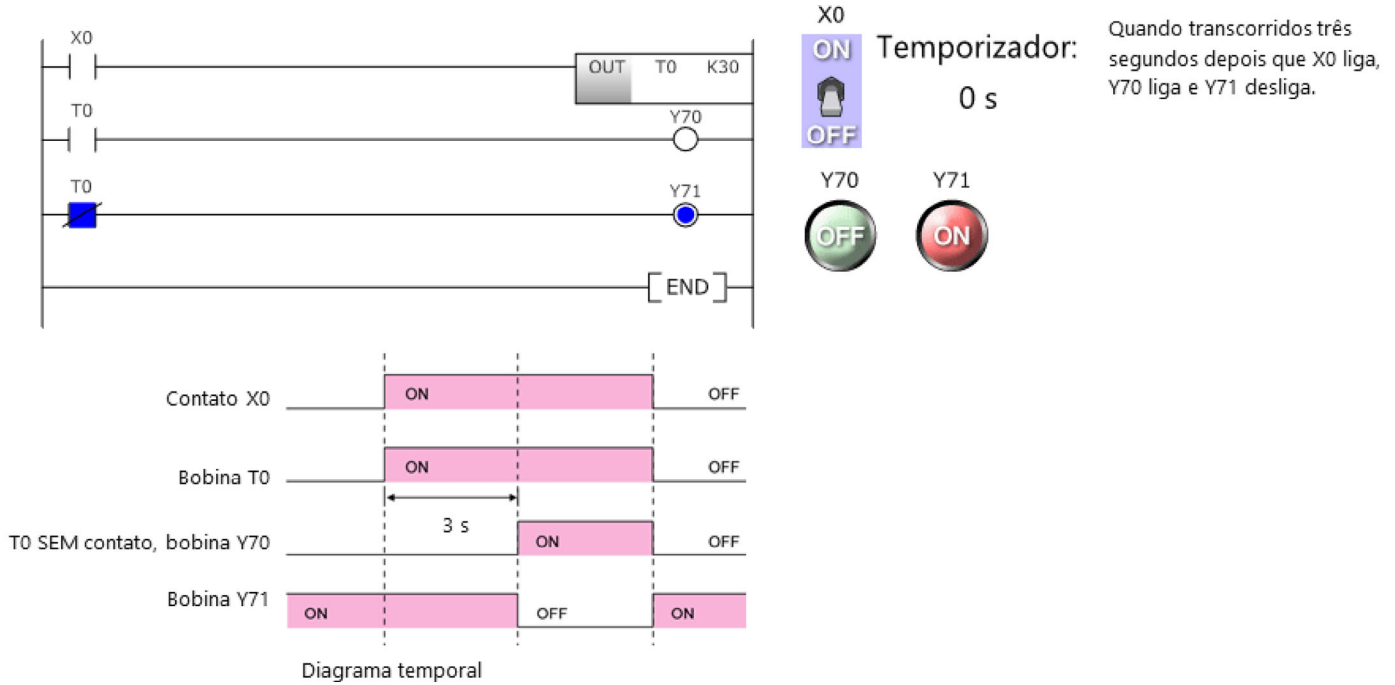
## 2.3.1

# Diferença entre o temporizador e o temporizador retentivo

Antes de explicar o temporizador retentivo, vejamos como o temporizador funciona.

O temporizador inicia a medição quando a bobina é ligada. Quando um tempo específico transcorre, o tempo para e o contato liga. Quando a bobina é desligada, o tempo medido é redefinido para "0". O símbolo do dispositivo do temporizador é "T".

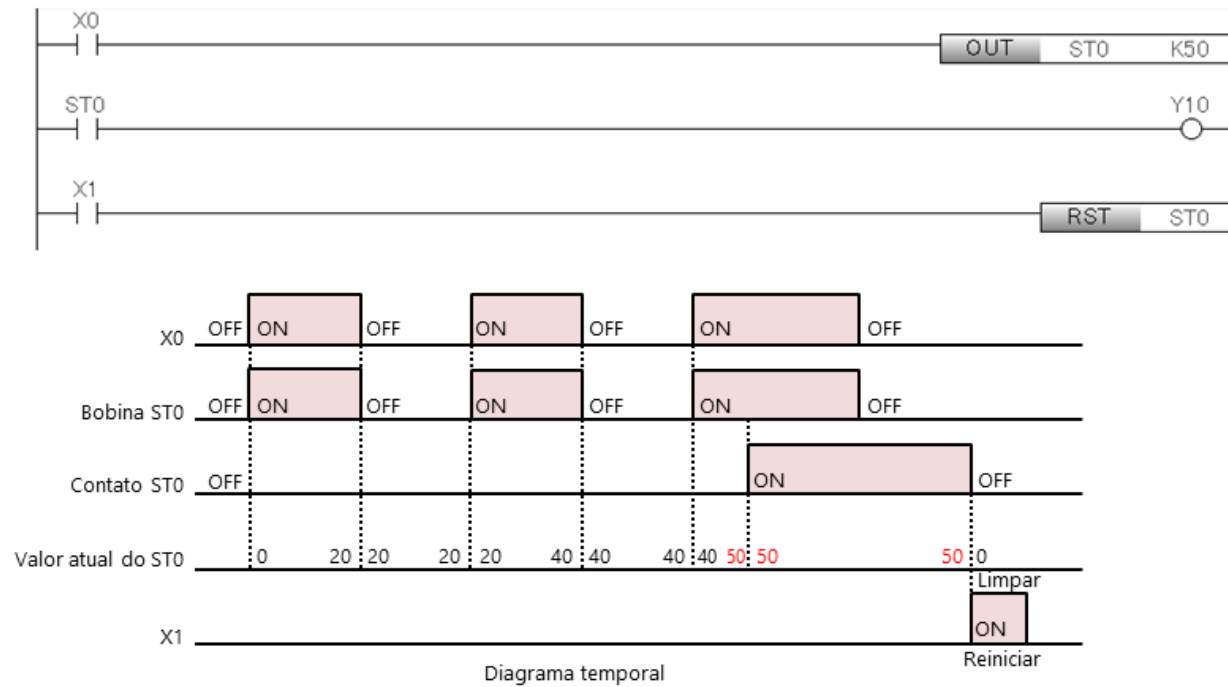
Use o interruptor de entrada no lado direito para ver como o temporizador funciona.



## 2.3.1

## Diferença entre o temporizador e o temporizador retentivo

O temporizador retentivo é útil para medir o tempo total de operação. O temporizador retentivo inicia a medição quando a bobina é ligada. Quando um tempo específico transcorre, o tempo para e o contato liga. Quando a bobina é desligada, o tempo de medição não é redefinido. Quando a bobina é ligada novamente, a medição reinicia a partir do valor mantido. O símbolo do dispositivo de um temporizador retentivo é "ST".



## 2.3.2

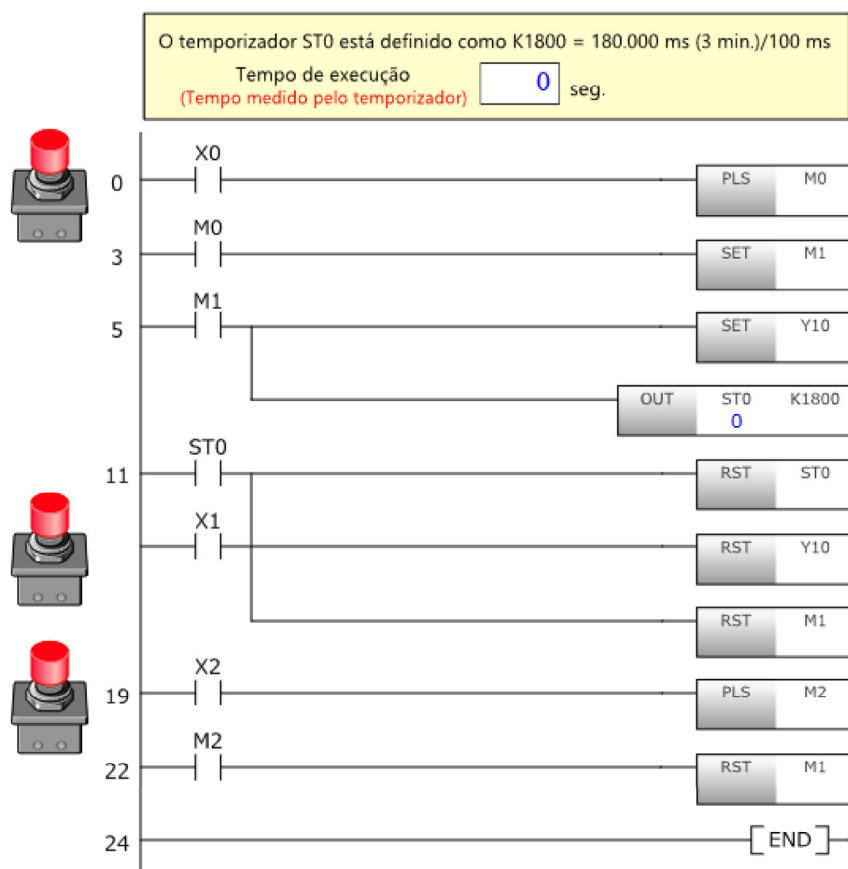
## Exemplo de programa do temporizador retentivo

Vejamos como o temporizador retentivo funciona fazendo uma simulação com uma máquina operante usando os interruptores de entrada (X0 a X2).

\*O temporizador retentivo (ST0) é definido em incrementos de 100 ms.



1. Quando X0 liga, a operação inicia.
2. Quando X2 liga, a operação é pausada e o valor atual é mantido.
3. Quando X0 liga novamente, a operação reinicia.
4. Quando X1 liga, a operação é finalizada e o valor atual é redefinido.



X0 a X2: Interruptores de entrada

Y10: Sinal inicial

## 2.3.3

## Definição do temporizador retentivo

O número de pontos utilizado pelo temporizador retentivo é "0" por padrão.

Antes de utilizar o temporizador retentivo, defina o número de pontos na "Device Setting" (Definição de dispositivo) do parâmetro da CPU com GX Works3.

No exemplo a seguir, 64 pontos (ST0 a ST63) estão definidos para o temporizador retentivo.

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
<b>Link Special Register SW</b>		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

### Utilizando uma label para especificar um temporizador

Defina "Timer" (Temporizador) como o tipo de dados quando uma label utiliza o dispositivo temporizador.

Label Name	Data Type
uTimer1	Timer
uTimer2	Retentive Timer

A definição do dispositivo é necessária para utilizar o temporizador retentivo, conforme descrito na seção anterior. No entanto, ela não é necessária para labels.

A unidade e o tempo de medição variam de acordo com o tipo de temporizador.

- Temporizador de alta velocidade (unidades de medição curtas)
- Temporizador de baixa velocidade (unidades de medição longas)
- O temporizador longo é capaz de medições de tempo longas

Cada um dos temporizadores acima tem a função de temporizador retentivo.

Tipo	Unidade de medição	Exemplo de programa	Operação	Retenção do valor atual
Temporizador de baixa velocidade	100 ms (padrão)		O temporizador de baixa velocidade T0 mede 5 segundos.	16 bits
Temporizador de alta velocidade	10,00 ms (padrão)		O temporizador de alta velocidade T1 mede 0,5 segundos.	
Temporizador retentivo de baixa velocidade	100 ms (padrão)		O temporizador retentivo de baixa velocidade ST0 mede 5 segundos.	
Temporizador retentivo de alta velocidade	10,00 ms (padrão)		O temporizador retentivo de alta velocidade ST1 mede 0,5 segundos.	
Temporizador longo	0,001 ms (padrão)		O temporizador longo LT0 mede 0,005 segundos.	32 bits
Temporizador retentivo longo			O temporizador retentivo longo LST0 mede 0,005 segundos.	

As unidades de medida iniciais são 100 ms para o temporizador de baixa velocidade, 10 ms para o temporizador de alta velocidade e 0,001 ms para o temporizador longo.

Consulte a próxima página para conferir como alterar a unidade de medição.

## 2.4

## Alterando a unidade de medição do temporizador

A unidade de medição do temporizador pode ser alterada na "Timer Limit Setting" (Definição do limite do temporizador) do parâmetro da CPU.

<b>Timer Limit Setting</b>	
Low Speed Timer/Low Speed Retentive Timer	100 ms
High Speed Timer/High Speed Retentive Timer	10.00 ms
Long Timer/Long Retentive Timer	0.001 ms



## 2.5

### Operando múltiplos dispositivos (registro de índice)

O registro de índice (Z) é utilizado em combinação com outro dispositivo para especificar indiretamente (modificar) o número do dispositivo do dispositivo de target de controle. O registro de índice é útil para simplificar programas pois pode descrever múltiplos dispositivos em um lote.

- O registro de índice é descrito com base em um símbolo do dispositivo e um número do dispositivo
- Número do dispositivo target de controle real = Símbolo do dispositivo (número do dispositivo + registro de índice)
- O número dos pontos do dispositivo para o registro de índice é 20 pontos (Z0 a Z19) por padrão

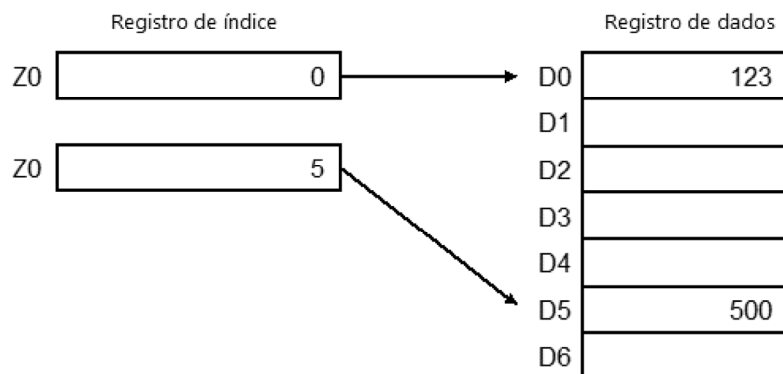
### 2.5.1

#### Exemplo de aplicação do registro de índice

Quando um dispositivo é descrito como "D0Z0", isso significa D(0+Z0).

Exemplo: quando Z0 é 0, o número do dispositivo é D0.

Quando Z0 é 5, o número do dispositivo é D5.



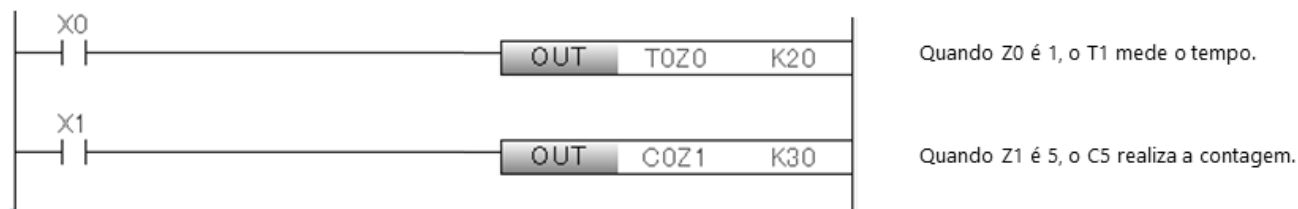
## 2.5.2

### Dispositivos que podem ser modificados pelo registro de índice

Dispositivos que podem ser modificados pelo registro de índice incluem:

Dispositivo de bits	X, Y, M, L, S, B, F
Dispositivo de palavra	T, C, D, R, W
Constante	K, H
Ponteiro	P

Observação: para contatos e bobinas utilizados em temporizadores e contadores, apenas o registro de índice "Z0" ou "Z1" pode ser utilizado.



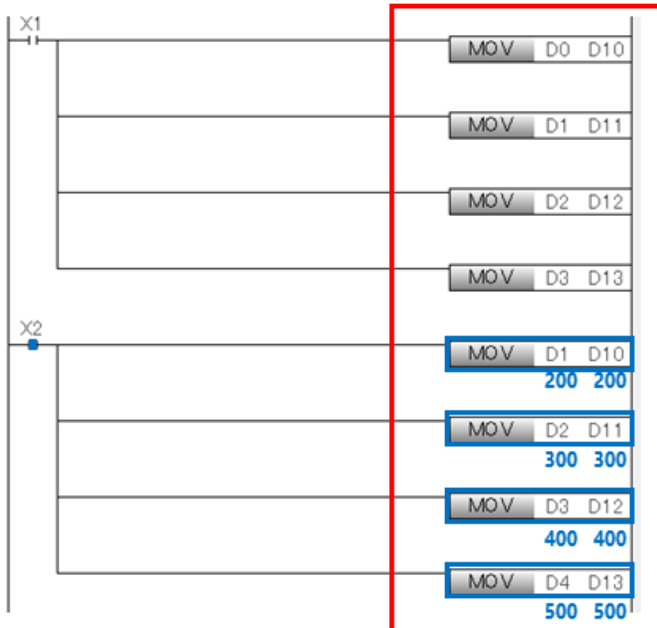
## 2.5.3

# Simplificação de programas utilizando o registro de índice

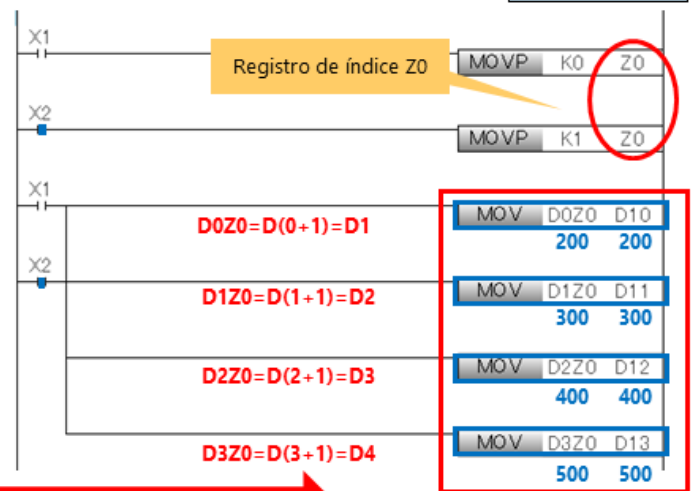
Os programas exibidos abaixo transferem os valores em "D0 a D4" para "D10 a D13" quando X1 ou X2 liga. Os programas (1) e (2) trarão o mesmo resultado. No programa (1), os dados são transferidos diretamente. No programa (2), os dados são transferidos indiretamente via registro de índice.

**Valores armazenados iniciais**  
D0=100  
D1=200  
D2=300  
D3=400  
D4=500

(1) Quando o registro de índice não é utilizado



(2) Quando o registro de índice é utilizado



O programa é simplificado.

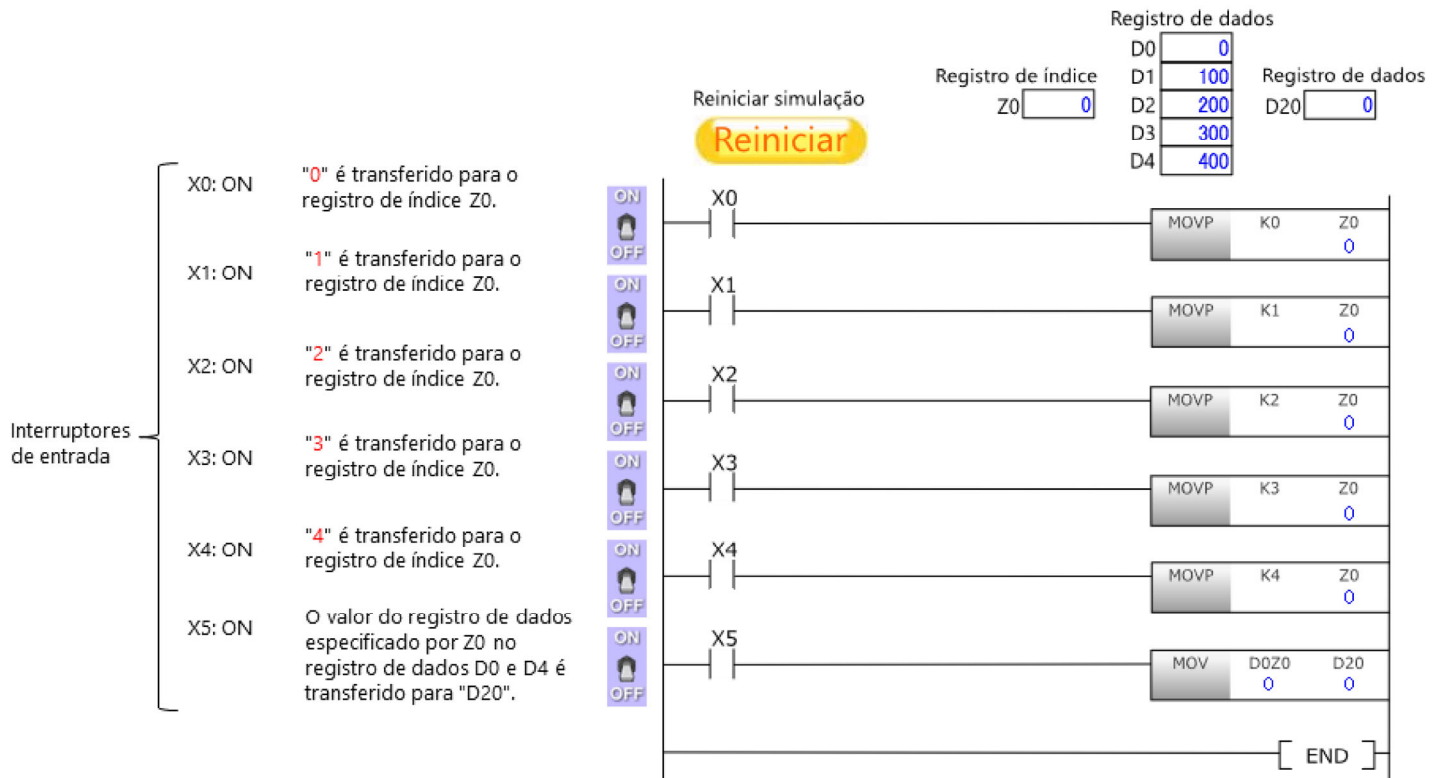
## 2.5.4

### Exemplo de programa do registro de índice

A operação do registro de índice Z0 pode ser simulada ligando os interruptores de entrada X0 a X5.

\*K0 a K400 já estão armazenados no registro de dados D0 a D4.

Ligue os interruptores de entradas X0 a X5 para verificar os valores armazenados em cada área do dispositivo.






## 2.6

## Operando múltiplos valores (array)

Ao usar um array, múltiplos valores podem ser operados por um nome de label.

No exemplo a seguir, os dados de volume de produção em uma fábrica de manufatura de automóveis são armazenados por destino.

Destino			
Volume de produção	35 unidades	75 unidades	65 unidades

Os dados de volume de produção por destino são atribuídos a uma label.

Quando um array não é utilizado, os nomes das labels devem ser criados para cada destino.

Contudo, ao utilizar um array, o volume de produção para múltiplos destinos pode ser atribuído e armazenado em um nome de label.

Quando um array não é utilizado

```
uProductionA
uProductionB
uProductionC
```



Quando um array é utilizado

```
uProduction
```

Labels individuais em um array são especificadas usando números de elemento. Números de elemento iniciam em [0].

```
uProduction [0]
```

Nome de label

Número de elemento



Destino (fila)

Country A	[0]	35
Country B	[1]	75
Country C	[2]	65



No exemplo de programa a seguir, o volume de produção planejado para Country A é transferido para outra label.

```
MOV uProduction[0] uShowProductionPlan
```

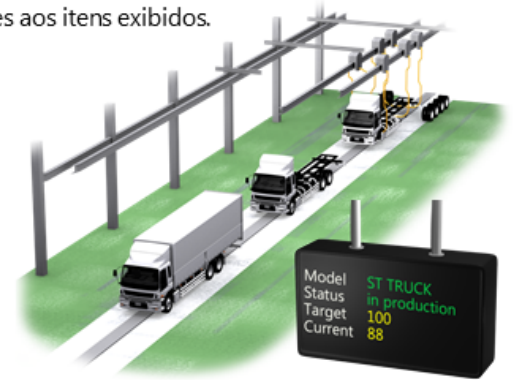
## 2.7

### Operando múltiplos valores (estrutura)

Ao usar uma estrutura, múltiplas labels relacionadas podem ser operadas por um nome de label. No exemplo a seguir, o estado de uma linha de produção de automóveis é exibido no Andon (painel de exibição).

A tabela a seguir lista os nomes de label, os valores e os tipos de dados correspondentes aos itens exibidos.

Item	Nome de label	Valor	Tipo de dados de label
Modelo	sModel	'ST TRUCK'	Tipo de string
Estado da operação	bStatus	'in production'	Tipo de bit
Target de volume de produção do dia	uPlanQty	'100' unidades	Tipo inteiro (palavra, não assinada)
Volume de produção atual	uActualQty	'88' unidades	Tipo inteiro (palavra, não assinada)



Se uma estrutura não é utilizada para uma fábrica que tem múltiplas linhas de produção, o nome da label deve ser alterado para cada linha.

A seguir, exemplos de nomes de label com nomes de linhas de produção adicionados.

**Primeira linha de produção**

```
s1stLineModel  
b1stLineStatus  
u1stLinePlanQty  
u1stLineActualQty
```

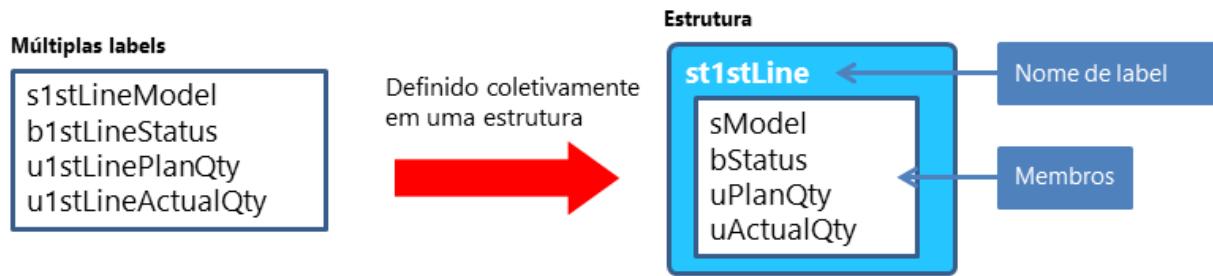
**Segunda linha de produção**

```
s2ndLineModel  
b2ndLineStatus  
u2ndLinePlanQty  
u2ndLineActualQty
```



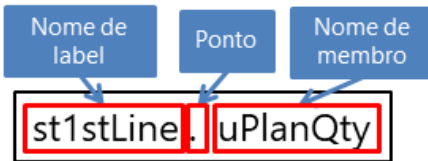
Conforme o número de linhas de produção aumenta, o número de labels a ser operado aumenta. Como resultado, o programa se torna mais longo e mais difícil de ler.

Uma estrutura possibilita que um nome de label represente múltiplas labels relacionadas a uma linha de produção. Portanto, as estruturas são usadas para organizar, armazenar e operar coletivamente as condições e especificações relacionadas a objetos ou matérias físicas, como dispositivos, equipamentos e peças.



Labels de estrutura são prefixadas com "st", que representa **st**structure (estrutura). As labels individuais definidas pela estrutura são chamadas de membros. Os tipos de dados de cada membro podem ser diferentes.

Para especificar cada membro em uma estrutura, adicione o nome do membro após o nome da label da estrutura com um ponto (.), como um delimitador intermediário.



No exemplo de programa a seguir, uma constante é atribuída a um membro da label do tipo de estrutura para a primeira linha de produção.

```
MOV K150 st1stLine.uPlanQty
```

## 2.8

### Mantendo o estado do dispositivo (retenção)

Esta seção descreve a função de retenção, que mantém os valores do dispositivo quando o módulo da CPU para a operação. Por exemplo, mesmo quando ocorre uma falha de potência que excede o tempo de falha de potência momentânea permitido, o controlador programável pode reiniciar o controle de sequências usando os dados mantidos no momento da parada da operação.

Se a função de retenção não é utilizada, os valores do dispositivo são limpos e redefinidos para os valores iniciais (desligado para dispositivos de bits e "0" para dispositivos de palavras) nos seguintes eventos:

- Desligamento
- Reinicialização pelo interruptor RUN/STOP/RESET (EXECUTAR/PARAR/REINICIAR)
- Uma falha de potência que excede o tempo de falha de potência momentânea permitido

### 2.8.1

#### Definindo a retenção em dispositivos

Defina o intervalo de retenção na janela "Device Setting" (Definição de dispositivo) do parâmetro da CPU. A seguir, um exemplo de definição de retenção do registro de dados, D0 a D128.

Latch (1)		Latch (2)		
No.	Device	Points (Decimal)	Start	End
1	D	129	0	128
2				

Dispositivo a ser retido

Número inicial do dispositivo retentivo

Número final do dispositivo retentivo



## 2.8.2

## Tipos de retenção e métodos de limpeza

Existem dois tipos de retenção (retenção (1) e retenção (2)), de acordo com os métodos de limpeza:

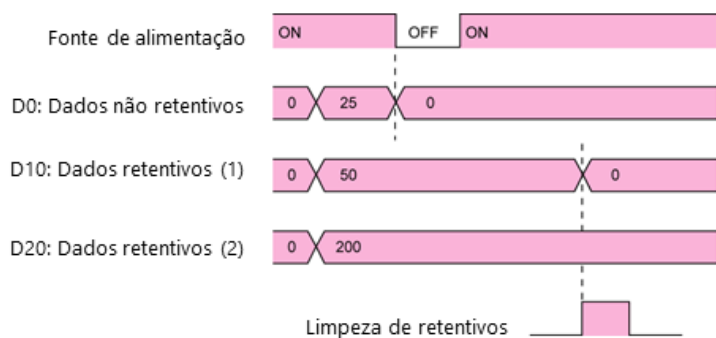
- Latch (1) (Retenção (1))

Os dados retidos podem ser limpos utilizando a função de operação da memória da CPU\* do GX Works3. Utilize a retenção 1 quando os dados retidos devem ser limpos no local de instalação.

- Latch (2) (Retenção (2))

Os dados retidos podem ser limpos utilizando uma instrução no programa. Utilize a retenção 2 quando os dados retidos não são limpos no local de instalação.

A seguir, o diagrama temporal da limpeza de retentivos.



No.	Device	Points (Decimal)	Start	End
1	D	129	0	128
2				

\*Execute a função selecionando [Online] (On-line) → [CPU Memory Operation] (Operação da memória da CPU) no menu do GX Works3.

### 2.8.3

### Definindo a retenção em labels

Para definir uma retenção em uma label, selecione o nome da classe que contém "RETAIN" na janela de definição de labels. Para labels locais, selecione "VAR\_RETAIN".

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

## 2.9

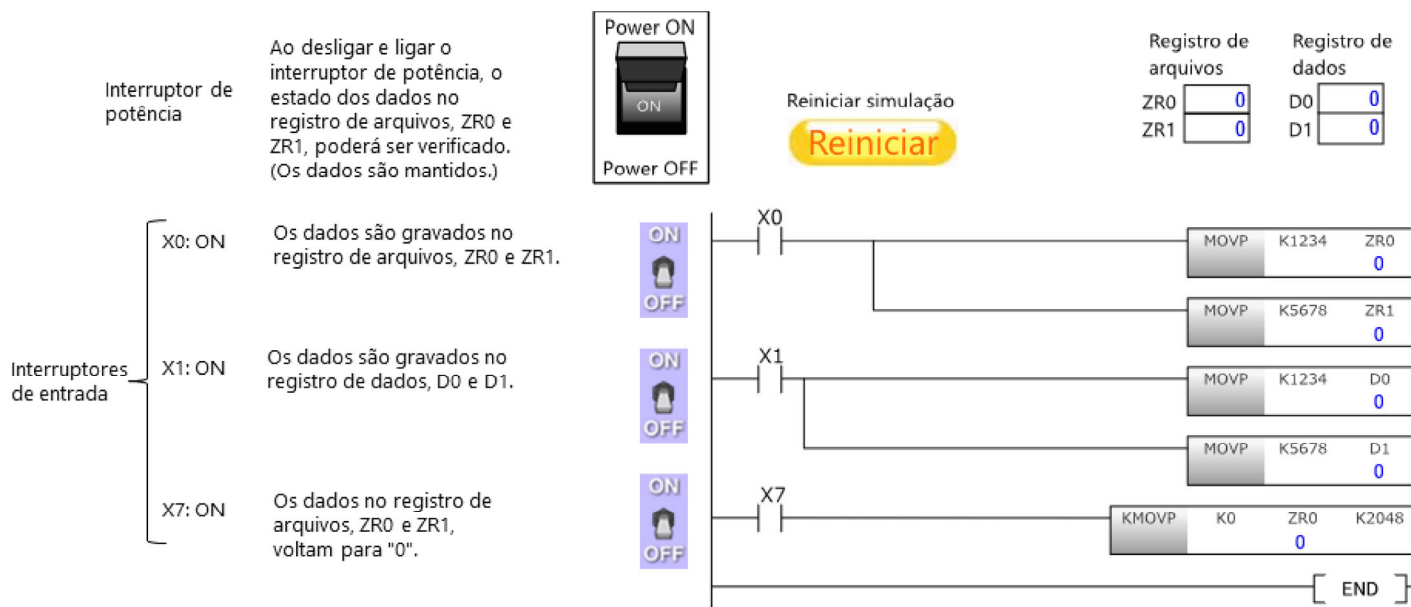
### Mantendo o estado do dispositivo (registro de arquivos)

- O registro de arquivos é um dispositivo de palavra utilizado para estender o registro de dados (D)
- Em comparação com o registro de dados, o registro de arquivos pode operar maiores quantidades de dados
- O registro de arquivos é armazenado na memória de dados do módulo da CPU ou em um cartão de memória
- Os dados armazenados no registro de arquivos serão operados mesmo quando o sistema estiver desligado ou quando o módulo da CPU for reiniciado.
- O registro de arquivos é útil para armazenar valores predefinidos, como valores padrão
- Para limpar os dados, escreva 0 para o intervalo do registro de arquivos ou limpe a memória do GX Works3
- O símbolo do dispositivo é "ZR"

### 2.9.1

#### Operação do programa ladder

A operação do registro de arquivos pode ser simulada ligando os interruptores de entrada e o interruptor de potência.



## 2.9.2

## Definição do registro de arquivos

Esta seção descreve a definição que especifica um registro de arquivos local como destino de armazenamento de cada programa. Selecione "File Register Setting" (Definição do registro de arquivos) no parâmetro da CPU e selecione "Use File Register of Each Program" (Utilizar o registro de arquivos de cada programa).

Item	Setting
<b>File Register Setting</b>	
Use Or Not Setting	Use File Register of Each Program
Capacity	
File Name	

Para gravar dados no controlador programável, a definição do registro de arquivos deve ser gravada em cada programa.

Defina o intervalo do registro de arquivos a ser gravado.

Module Name/Data Name	Detail	Title
Memory Card Parameter		
Remote Password	<input type="checkbox"/>	
Global Label	<input type="checkbox"/>	
Global Label Setting	<input type="checkbox"/>	
Program	<input type="checkbox"/>	Detail
MAIN	<input type="checkbox"/>	
Device Memory	<input type="checkbox"/>	
MAIN	<input type="checkbox"/>	Detail
File Register	<input type="checkbox"/>	Detail
MAIN	<input type="checkbox"/>	Detail
2018/05/08 10:35:52		-
Common Device Comment	<input type="checkbox"/>	2018/05/08 10:35:52 Not Calculated
COMMENT	<input type="checkbox"/>	2018/05/08 10:35:52 Not Calculated

## 2.10

## Usando dispositivos com funções e operações predeterminadas

O relé especial e o registro especial utilizados no módulo da CPU predeterminaram as funções e as operações.

O relé especial (SM) é o relé interno utilizado para as informações de bit (on/off) e o registro especial (SD) é o registro interno utilizado para as informações de palavra.

### 2.10.1

### Tipos de relé especial e registro especial

O relé especial e o registro especial são categorizados por seus tipos de informação. Os principais tipos estão listados abaixo. Em programas de usuários, o relé especial e o registro especial são utilizados como condições determinantes de controle. Eles também são utilizados na monitoração de operações, que pode ser realizada no monitor do dispositivo do GX Works3.

Informações de diagnóstico
Resultados de diagnóstico do módulo da CPU
Erros de diagnóstico e códigos de erro
Informações do sistema
Informações do sistema do módulo da CPU
Estado de operação do módulo da CPU, dados do relógio e outras informações
Relógio do sistema
Sinais do relógio e valores de contagem que são utilizados como elementos de temporização básicos
Diversos sinais do relógio (sempre ON, ON/OFF em intervalos específicos e outros sinais)

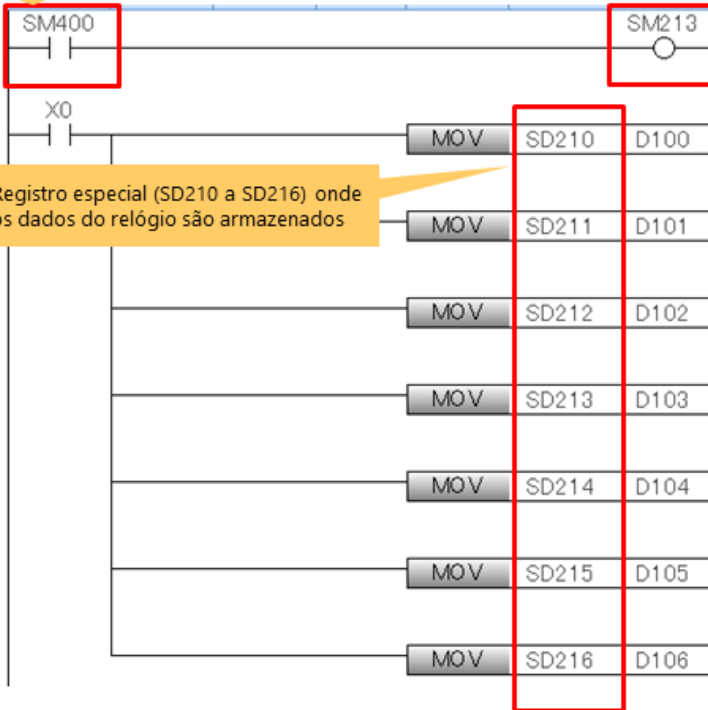
## 2.10.2

## Exemplo de programa de relé especial e registro especial

A seguir, um exemplo de programa de leitura dos dados do relógio do módulo da CPU utilizando o relé especial e o registro especial.

Relé especial (Always ON (Sempre ON))

Relé especial que requer a leitura dos dados do relógio do módulo da CPU



Registro especial (SD210 a SD216) onde os dados do relógio são armazenados

SM213 (Solicitação de leitura dos dados do relógio) liga durante a execução (RUN).

Dados do relógio (ano)

Dados do relógio (mês)

Dados do relógio (dia)

Dados do relógio (hora)

Dados do relógio (minuto)

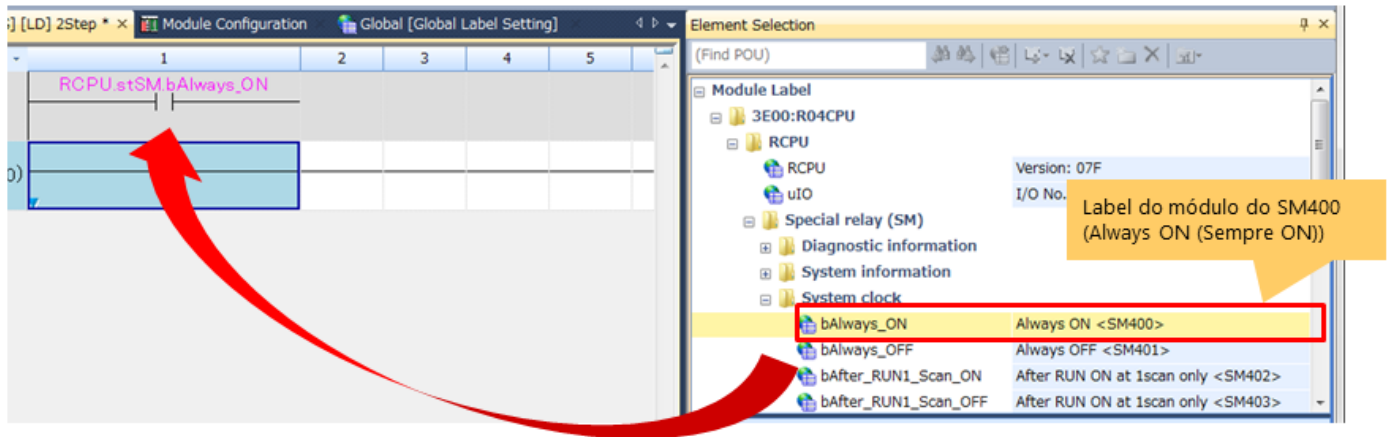
Dados do relógio (segundo)

Dados do relógio (dia da semana)

## 2.10.3

### Utilização de labels para o relé especial e o registro especial

O relé especial e o registro especial são preparados como labels de módulo no módulo da CPU. Eles podem ser utilizados bastando selecionar e colocar os nomes de label relevantes, sem verificar os números do dispositivo no manual.



## 2.11 Cálculo com números reais

### 2.11.1 Aplicação de números reais

- "Números reais" são valores numéricos com pontos decimais
- Inteiros são geralmente utilizados em programas de controle. Contudo, os números reais são utilizados em programas de controle de operação avançados, como funções trigonométricas e operações exponenciais, porque os valores numéricos com pontos decimais devem ser utilizados nesses programas
- Dados numéricos de números reais utilizados no módulo da CPU são chamados de "dados de ponto flutuante"

#### Observação:

- Um número real **sempre utiliza dois dispositivos de palavras consecutivos** (espaço de memória de 32 bits), independente do número \*
- Em programas de controle, **as instruções de operação dedicada** (como adição, subtração, multiplicação, divisão e funções especiais) que utilizam números reais são preparadas. As instruções de conversão entre números reais e inteiros também são preparadas

\*Se é necessário um cálculo mais preciso com um número maior de dígitos, utilize quatro dispositivos de palavra.

- Números reais que utilizam dois dispositivos de palavra são chamados de números reais de precisão simples
- Números reais que utilizam quatro dispositivos de palavra são chamados de números reais de precisão dupla

Este curso concentra-se em números reais de precisão simples.



## 2.11.2 Notação de números reais

"E" é utilizado para representar um número real.

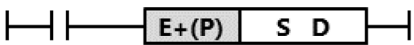
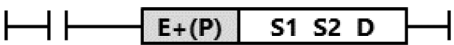
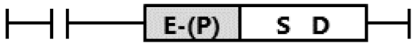
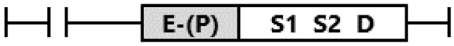
### Expressando uma constante com números reais

Para escrever uma constante, inicie com "E".

Expressão normal	Escreva um valor numérico normalmente. (Exemplo) Escreva 10,2345 como "E10.2345".
Expressão exponencial	Escreva um valor numérico como "(valor numérico) × 10 <sup>n</sup> ". (Exemplo) Escreva 1234,0 como "E1.234+3".

## 2.11.3

## Instruções de operação (adição e subtração)

Símbolo da instrução	Exemplo de ladder	
E+ (adicionando números reais de precisão simples)	 <p>A operação de número real "D + S = D" é executada.</p>	 <p>A operação de número real "S1 + S2 = D" é executada.</p>
E- (subtraindo números reais de precisão simples)	 <p>A operação de número real "D - S = D" é executada.</p>	 <p>A operação de número real "S1 - S2 = D" é executada.</p>

S (fonte): Dados antes da operação (constante, número do dispositivo)

D (destino): Destino dos dados após a operação (número do dispositivo)

P: Instruções a serem executadas na borda ascendente (de off para on)

S1 e S2: dois itens de dados são operados.

**Observação:**

Números reais e inteiros não podem ser misturados em uma operação.

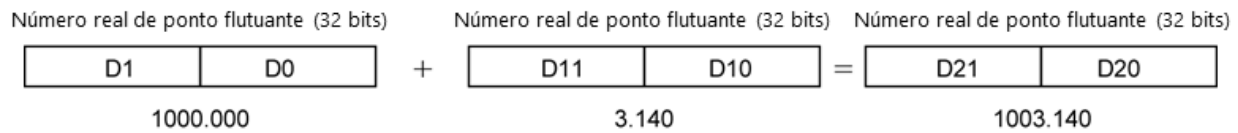
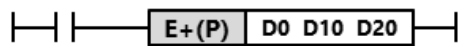
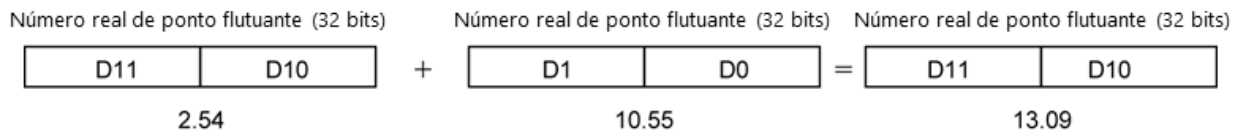
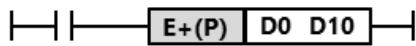
Para operações com números reais de precisão simples, S, S1 e S2 na expressão operacional devem ser números reais de precisão simples.

Números reais de precisão simples são armazenados em D.

### 2.11.3

## Instruções de operação (adição e subtração)

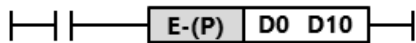
Exemplo de aplicação da instrução de adição



### 2.11.3

## Instruções de operação (adição e subtração)

Exemplo de aplicação da instrução de subtração



Número real de ponto flutuante (32 bits)



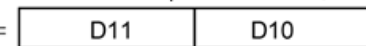
1000.000

Número real de ponto flutuante (32 bits)

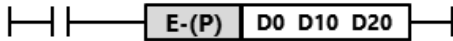


320.560

Número real de ponto flutuante (32 bits)



679.440



Número real de ponto flutuante (32 bits)



2.540

Número real de ponto flutuante (32 bits)



10.550

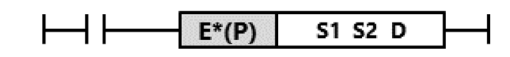
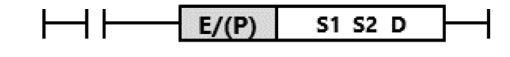
Número real de ponto flutuante (32 bits)



-8.010

## 2.11.4

## Instruções de operação (multiplicação e divisão)

Símbolo da instrução	Exemplo de ladder
E* (multiplicando números reais de precisão simples)	 A operação de número real "S1 * S2 = D" é executada.
E/ (dividindo números reais de precisão simples)	 A operação de número real "S1 / S2 = D" é executada.

S1, S2 (origem): dois itens de dados são operados.

D (destino): Destino dos dados após a operação (número do dispositivo)

P: Instruções a serem executadas na borda ascendente (de off para on)

### Observação:

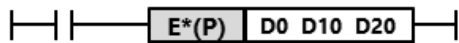
Para operações com números reais de precisão simples, S1 e S2 na expressão operacional devem ser números reais de precisão simples.

Números reais de precisão simples são armazenados em D.

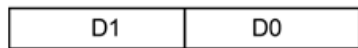
## 2.11.4

## Instruções de operação (multiplicação e divisão)

### Exemplo de aplicação da instrução de multiplicação



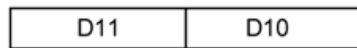
Número real de ponto flutuante (32 bits)



1000.000

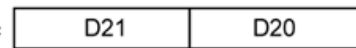
×

Número real de ponto flutuante (32 bits)



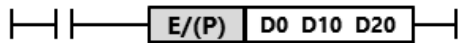
25.590

Número real de ponto flutuante (32 bits)

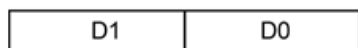


25590.000

### Exemplo de aplicação da instrução de divisão



Número real de ponto flutuante (32 bits)



1000.000

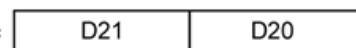
÷

Número real de ponto flutuante (32 bits)



25.590



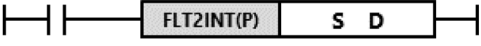

Número real de ponto flutuante (32 bits)



39.078

## 2.11.5

## Instruções de conversão entre números reais e inteiros

Símbolo da instrução	Exemplo de ladder	
<b>INT2FLT</b> (conversão de número inteiro para número real de precisão simples)	Um número inteiro (16 bits) é convertido em um número real (32 bits).    S (16 bits) é convertido e armazenado em D.	Um número inteiro (32 bits) é convertido em um número real (32 bits).    S (32 bits) é convertido e armazenado em D.
<b>FLT2INT</b> (conversão de número real de precisão simples em número inteiro)	Um número real (32 bits) é convertido em um número inteiro (16 bits).    S é convertido e armazenado em D (16 bits).	Um número real (32 bits) é convertido em um número inteiro (32 bits).    S é convertido e armazenado em D (32 bits).

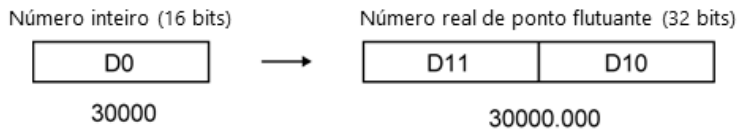
S (fonte): Dados antes da operação (constante, número do dispositivo)

D (destino): Destino dos dados após a operação (número do dispositivo)

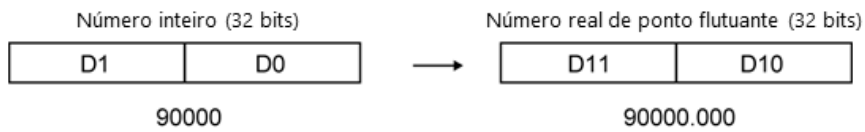
## 2.11.5

## Instruções de conversão entre números reais e inteiros

Exemplo de aplicação da instrução de conversão do número inteiro (16 bits) em número real (32 bits)



Exemplo de aplicação da instrução de conversão do número inteiro (32 bits) em número real (32 bits)

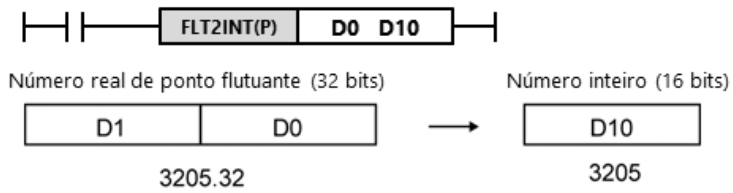




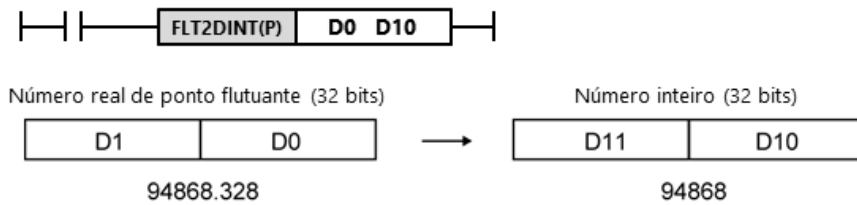
## 2.11.5

## Instruções de conversão entre números reais e inteiros

Exemplo de aplicação da instrução de conversão do número real (32 bits) em número inteiro (16 bits)



Exemplo de aplicação da instrução de conversão do número real (32 bits) em número inteiro (32 bits)



## 2.11.6

### Utilizando labels que representam números reais

Para utilizar labels para números reais, defina o tipo de dados como "Single Precision" (Precisão simples) ou "Double Precision" (Precisão dupla) na janela de definição de labels.

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

Neste capítulo, você aprendeu:

- Especificação de bit do dispositivo de palavra
- Especificação de borda ascendente ou descendente para contatos
- Temporizador retentivo
- Especificação da unidade de medição do temporizador
- Registro de índice
- Array
- Estrutura
- Retenção
- Registro de arquivos
- Relé especial, registro especial
- Cálculos com números reais

#### Pontos importantes

Temporizador retentivo	O tempo medido é mantido mesmo quando a bobina desliga e a medição é retomada quando a bobina liga novamente.
Unidade de medição do temporizador	A unidade de medição do temporizador pode ser alterada no parâmetro.
Registro de índice	Múltiplos dispositivos podem ser descritos em um lote.
Array	Múltiplos valores podem ser operados por um nome de label.
Estrutura	Múltiplas labels relacionadas podem ser operadas por um nome de label.
Retenção	<ul style="list-style-type: none"> <li>• Valores de dispositivos retidos são mantidos quando o módulo da CPU para a operação</li> <li>• Valores de dispositivos retidos são limpos pela operação da memória da CPU ou usando uma instrução de programa</li> </ul>
Registro de arquivos	<ul style="list-style-type: none"> <li>• Em comparação com o registro de dados, o registro de arquivos pode operar maiores quantidades de dados</li> <li>• Valores de dispositivos são mantidos quando o módulo da CPU para a operação</li> <li>• Os valores de dispositivo podem ser limpos pela operação da memória da CPU ou escrevendo 0 no intervalo de dispositivo</li> </ul>
Relé especial, registro especial	O estado interno do módulo da CPU, como as informações de diagnóstico e as informações de sistema, já foi armazenado nesses dispositivos.
Número real	<ul style="list-style-type: none"> <li>• Utiliza pelo menos dois dispositivos de palavra (32 bits)</li> <li>• Instruções de operação dedicada são fornecidas</li> <li>• Números reais e inteiros não podem ser misturados em uma operação</li> </ul>

## **Capítulo 3** Resolução de problemas eficiente

Este capítulo descreve funções de GX Works3 para a resolução de problemas eficiente.

- 3.1 Alteração temporária do intervalo do programa
- 3.2 Verificando a operação durante a alteração dos valores do dispositivo
- 3.3 Simulação da operação do programa

### 3.1

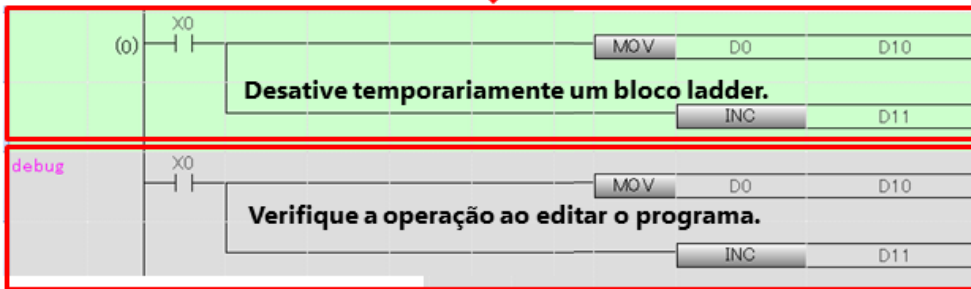
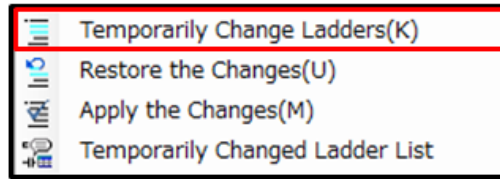
## Alteração temporária do intervalo do programa

Ao editar um intervalo amplo para a resolução de problemas do programa, será difícil desfazer todas as alterações.

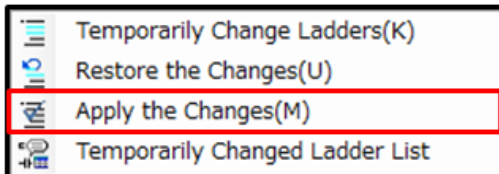
Ao desativar temporariamente o bloco ladder desejado, os usuários poderão usar uma cópia do programa para realizar a resolução de problemas sem alterar o original.

(Função Temporarily change ladders (Alterar ladders temporariamente))

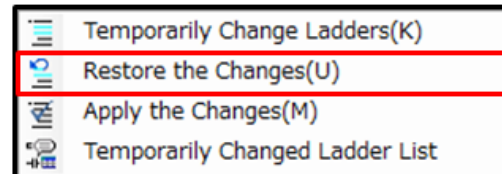
Após alterar o bloco ladder e verificar a operação com essa função, as alterações serão aplicadas se não houver problemas ou as alterações serão restauradas se houver um problema.



Quando não são encontrados problemas



Quando são encontrados problemas



## 3.2

### Verificando a operação durante a alteração dos valores do dispositivo

Ao executar um programa criado, o estado on/off dos dispositivos de bits e os valores armazenados nos dispositivos de palavra podem ser exibidos no editor do programa. (Função Monitor (Monitorar))

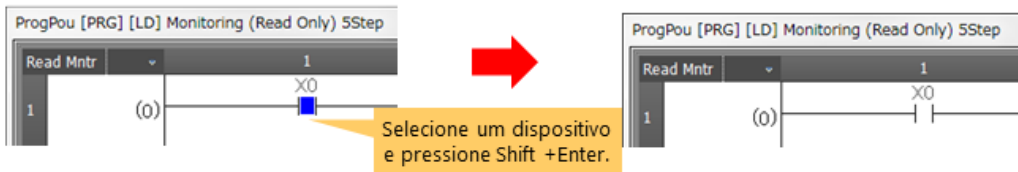
Com a função de monitorar, os usuários podem verificar com facilidade o estado de operação do programa.



Os valores atuais do dispositivo podem ser alterados à força durante a monitoração. (Alteração do valor atual)

Com a função de alterar o valor atual, as alterações podem ser feitas sem editar todo o programa ou sem executá-lo no próprio sistema.

O estado dos dispositivos de bits pode ser alterado no editor do programa.

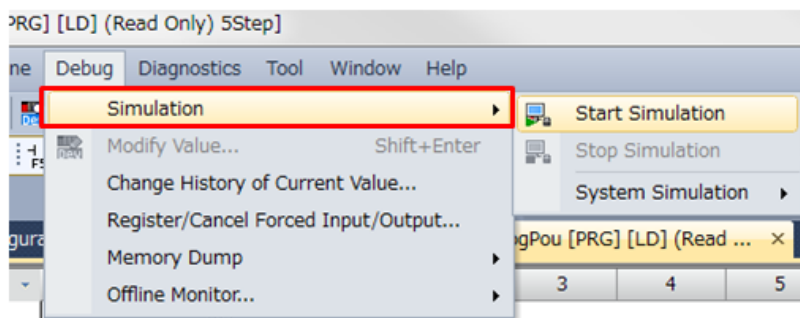


Ao utilizar a janela "Watch" (Observar), os dispositivos de palavra que devem ser monitorados poderão ser registrados e seus valores atuais poderão ser alterados.

Name	Current Value	Display Format	Data Type
X0	FALSE	BIN	Bit
D0	100	Decimal	Word [Signed]

Se um programa recém-criado é executado no próprio sistema, um erro inesperado pode ocorrer.

A operação do programa pode ser simulada sem a utilização do próprio controlador programável. (Função Simulation (Simulação))



Com a função de simulação, a operação do programa pode ser verificada como se o programa está sendo executado no próprio controlador programável.

Neste capítulo, você aprendeu:

- Alteração temporária de um bloco ladder
- Monitoração do programa e alteração do valor atual
- Simulação do programa

Pontos importantes

Alteração temporária de um bloco ladder	Um bloco ladder pode ser desativado temporariamente e uma cópia do programa pode ser utilizada para a resolução de problemas sem alterar o programa original.
Monitorar	<ul style="list-style-type: none"><li>• A execução do programa pode ser visualizada.</li><li>• A operação do programa pode ser verificada durante a alteração à força dos valores atuais do dispositivo.</li></ul>
Simulação	A operação do programa pode ser simulada sem a utilização do próprio controlador programável.



Chegamos ao fim deste curso de e-learning.

A seguir, um resumo deste curso.

- **Programação eficiente**

- Atribua números de E/S fixos aos módulos para utilizar programas com facilidade em sistema diferentes
- Ajuste as áreas da memória de acordo com o estado de uso do dispositivo
- Utilize labels para facilitar a programação e tornar a operação mais compreensível
- Adicione comentários para melhorar a legibilidade do programa

- **Programação avançada**

- Utilize o temporizador retentivo para manter o tempo medido
- Utilize o registro de índice, arrays ou estruturas para operar valores coletivamente
- Utilize a função de retenção e o registro de arquivos para manter o estado do dispositivo
- O relé especial e o registro especial que armazenam o estado interno do módulo da CPU são fornecidos
- Um número real é representado por dispositivos de duas ou quatro palavras. Números reais e inteiros não podem ser misturados em uma operação

- **Resolução de problemas eficiente**

Os usuários podem realizar o seguinte utilizando o GX Works3:

- Solucionar o problema de um programa sem alterar o programa original
- Visualizar como o programa é executado
- Simular a operação do programa

Para realizar a próxima etapa, faça os seguintes cursos sobre "estruturação", que parte o programa em camadas e componentes para que possam ser reutilizados com facilidade.

- Programação eficiente
- Programação eficiente (Prática) (lançamento em breve)



Qual afirmação a seguir sobre a atribuição de número de E/S de módulos é verdadeira?

Q1

- Números de E/S podem ser atribuídos manualmente a cada módulo para que não seja necessário modificar o programa quando as configurações do módulo são alteradas**
- Números de E/S atribuídos automaticamente não podem ser alterados**

Qual afirmação a seguir sobre a definição de pontos do dispositivo é verdadeira?

Q1

- Pelo menos um ponto deve ser atribuído a cada dispositivo, mesmo que o dispositivo não seja utilizado
- Os pontos podem ser atribuídos de acordo com o número de pontos utilizado

Qual afirmação a seguir sobre labels é verdadeira? (Múltiplas respostas)

Q1

- Usar labels ajuda a identificar o target do processamento e facilita a programação
- As labels que representam o sinais do módulo e os valores de definição são fornecidos
- Comentários podem ser adicionados aos elementos para melhorar a legibilidade do programa
- Uma vez que é possível atribuir constantes a labels, os valores podem ser alterados sem a modificação do programa

Complete o texto a seguir que descreve o temporizador retentivo.

O temporizador retentivo começa a medição quando o/a **(Q1)** liga (a bobina fica **(Q2)**).

O temporizador retentivo mantém o tempo medido mesmo quando a condição de entrada fica **(Q3)** e continua a medição a partir do valor mantido quando a condição de entrada fica **(Q4)** novamente.

O temporizador retentivo acaba a contagem quando o tempo medido atinge o valor definido. Nesse momento, o/a

Q1

-- Select --



Q2

-- Select --



Q3

-- Select --



Q4

-- Select --



Q5

-- Select --



Q6

-- Select --



Complete o programa de controle que executa o processamento a seguir.

- Utilize o temporizador retentivo (ST0) para medir a pontualidade do sinal de entrada X0 ou X1
- Quando a pontualidade de X0 ou X1 atinge 30 segundos, ligue a bobina Y70 e o indicador de tempo limite
- Quando X2 ligar, desligue o contato do temporizador retentivo (ST0) e limpe o tempo medido (valor atual)

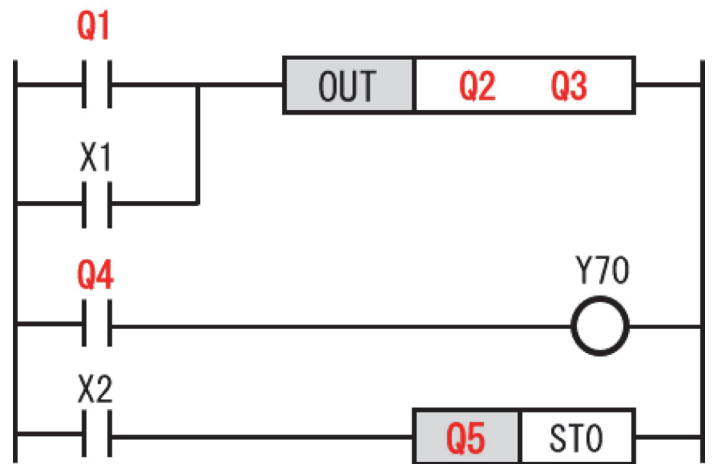
Q1

Q2

Q3

Q4

Q5



[ + ]

Selecione o valor armazenado no registro dados D20 quando X0 liga sob cada uma das seguintes condições no programa de controle abaixo.

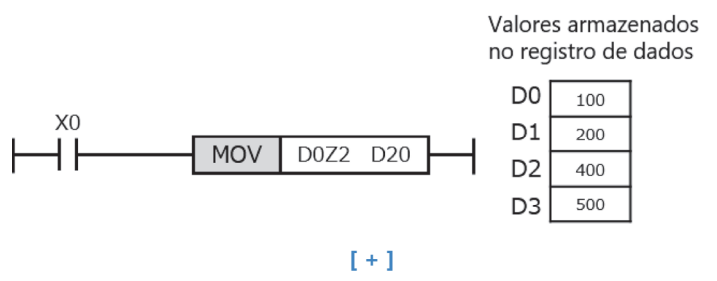
- Q1) Quando o valor armazenado em Z2 é "0"
- Q2) Quando o valor armazenado em Z2 é "1"
- Q3) Quando o valor armazenado em Z2 é "2"

Q1

Q2

Q3

Q4





Qual afirmação a seguir sobre como especificar um elemento de um conjunto é verdadeira?

Q1

- Adicionar um número de elemento ao final do nome da label
- Especificar um número do dispositivo indiretamente

Qual afirmação a seguir sobre estruturas é falsa?

Q1

- Estruturas são usadas para organizar e armazenar coletivamente as condições e as especificações relacionadas a objetos ou matérias físicas
- Ao usar estruturas, o processamento de grandes quantidades de dados pode ser descrito concisamente
- Os membros definidos em uma estrutura devem ter o mesmo tipo de dados

Qual afirmação a seguir sobre a função de retenção é verdadeira?

Q1

- Originalmente, os dispositivos têm uma função de manter valores
- A definição do parâmetro de manter valores necessita utilizar o software de engenharia

Complete o texto a seguir que descreve o registro de arquivos.

O registro de arquivos é um dispositivo de palavra usado para estender o (D) do registro de dados, e seu símbolo de dispositivo é **(Q1)**.

O registro de arquivos tem uma capacidade **(Q2)** que a do registro de dados, e os dados armazenados são **(Q3)** mesmo quando o sistema é desligado ou o módulo da CPU é reiniciado.

Q1

-- Select --



Q2

-- Select --



Q3

-- Select --



Q4

-- Select --



Qual afirmação a seguir sobre o relé especial e o registro especial é verdadeira?

Q1

- O estado interno do módulo da CPU já foi armazenado no relé especial e no registro especial e esses dispositivos são usados como condições de determinação em um programa de controle
- As funções especiais podem ser atribuídas livremente ao relé especial e ao registro especial

Complete o texto a seguir que descreve números reais (precisão simples).

- Um número real utiliza **(Q1)** dispositivo(s) de palavra e é armazenado em um espaço de memória de **(Q2)** bits.
- Os dados do valor numérico dos números reais são chamados de **(Q3)**. Por exemplo, 2,035 é descrito como **(Q4)** em um programa de controle.
- Números inteiros e reais **(Q5)** ser misturados em uma instrução de operação que lida com números reais.

Q1

-- Select --



Q2

-- Select --



Q3

-- Select --



Q4

-- Select --



Q5

-- Select --



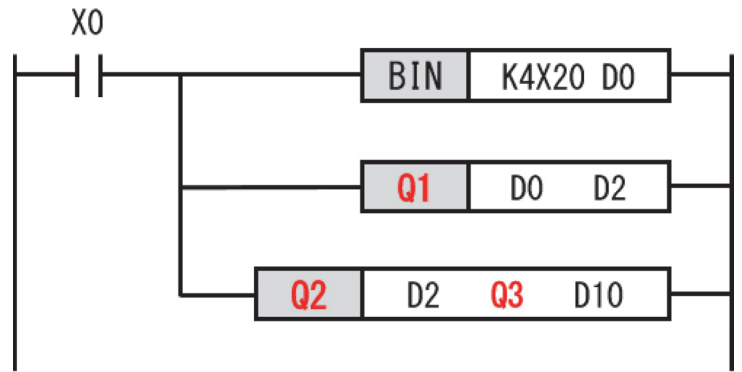
Complete o programa de controle que executa o processamento a seguir.

- Ler os dados de X20 a X2F (dados em BCD) quando X0 está ligado e armazenar em D0.
- Converter o valor em D0 para um número real e armazenar o valor em D2.
- Multiplicar o valor em D2 por 3,14 e armazenar o resultado em D10.

Q1

Q2

Q3



Qual afirmação a seguir sobre a resolução de problemas do programa de controle é verdadeira?

Q1

- A operação do programa pode ser simulada com segurança usando a função do software de engenharia.
- Para resolver o problema de um programa, ele deve ser executado no próprio sistema.



Você concluiu o Teste Final. Seus resultados são os seguintes.  
Para terminar o Teste Final, vá para a próxima página.

	1	2	3	4	5	6	7	8	9	10
Teste Final 1	✓									
Teste Final 2	✓									
Teste Final 3	✓									
Teste Final 4	✓	✓	✓	✓	✓	✓				
Teste Final 5	✓	✓	✓	✓	✓					
Teste Final 6	✓	✓	✓	✓						
Teste Final 7	✓									
Teste Final 8	✓									
Teste Final 9	✓									
Teste Final 10	✓	✓	✓	✓						
Teste Final 11	✓									
Teste Final 12	✓	✓	✓	✓	✓					
Teste Final 13	✓	✓	✓							
Teste Final 14	✓									

Total de perguntas: **35**

Respostas corretas: **35**

Porcentagem: **100 %**

**Limpar**

**Você concluiu o curso **Aplicativos de programação (Diagrama ladder/série MELSEC iQ-R).****

Muito obrigado por fazer este curso.

Esperamos que tenha gostado das lições e que as informações adquiridas sejam úteis no futuro.

Você pode rever o curso quantas vezes quiser.

**Revisar**

**Fechar**