

PLC

Introdução ao GX Works2

Este curso (e-learning) foi projetado para aqueles que utilizam o software GX Works2 pela primeira vez para criar programas de sequência.

Introdução**Objetivo do curso**

Este curso proporciona conhecimentos básicos para uso do software GX Works2 para programação, depuração e verificação da operação de um controlador programável (PLC). Este curso destina-se àqueles que criam programas de sequência para dispositivos de controle das séries MELSEC-Q, MELSEC-L e MELSEC-F.

O conteúdo do curso é explicado a seguir.
Recomendamos que você comece pelo Capítulo 1.

Capítulo 1 - Método de controle do sistema do PLC

A linguagem e o software de programação utilizados são apresentados aqui.

Capítulo 2 - Criação do programa

Você aprenderá a criar um programa com base em itens de controle e na configuração do hardware.

Capítulo 3 - Programação

Você aprenderá a programar utilizando o software dedicado GX Works2.

Capítulo 4 - Depuração

Você aprenderá a escrever programas de sequência no módulo de CPU e a depurá-los.

Capítulo 5 - Teste final

Pontuação para aprovação: 60% ou mais.

Introdução**Como utilizar esta ferramenta de e-Learning**

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. A janela, como a tela de "Conteúdo", e o curso serão fechados.

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.

Capítulo 1 Método de controle do sistema do PLC

Este curso foi projetado para pessoas que trabalham com software de engenharia. Este curso destina-se àqueles que criam programas de sequência para dispositivos de controle das séries MELSEC-Q, MELSEC-L e MELSEC-F.

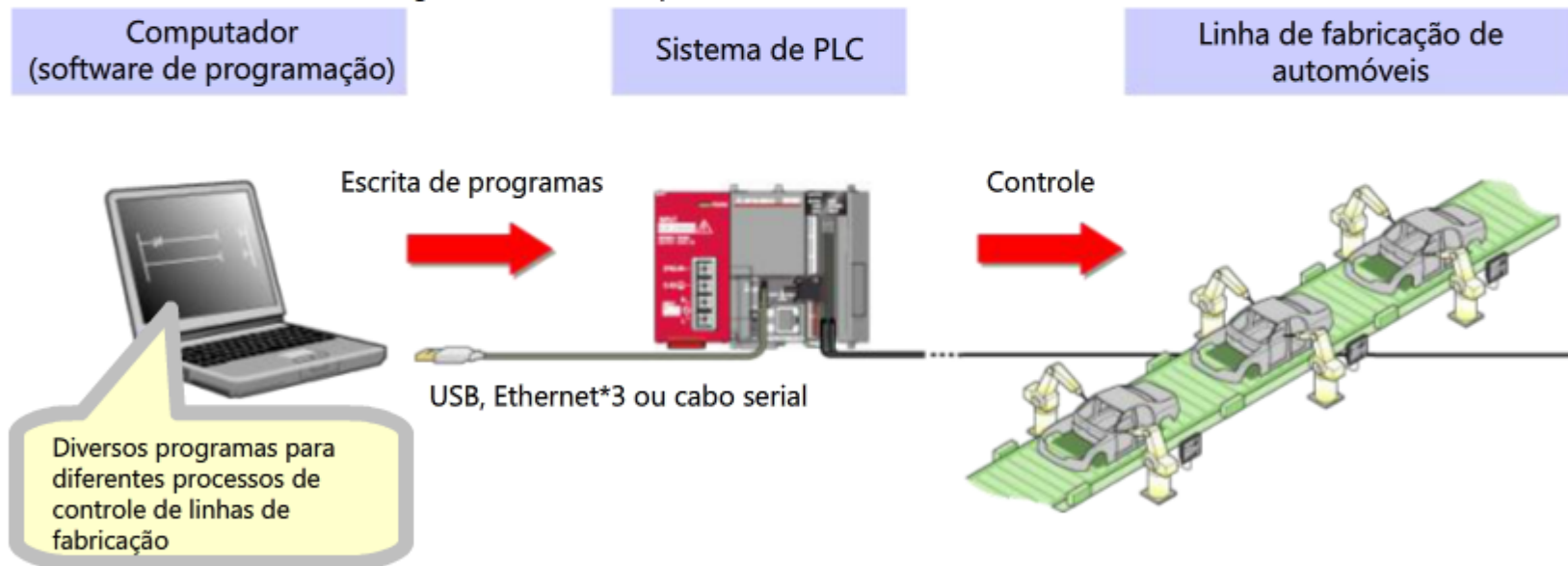
O GX Works 2 (GXW2) utiliza linguagens de programação internacionalmente padronizadas, incluindo Sequential função Chart (SFC), Instruction List (IL)*1, Ladder Logic, função Block Diagram (FBD)*2 e Structured Text (ST).

Os programas são desenvolvidos usando um computador pessoal que executa o "software de engenharia", GX works2, e normalmente são escritos para a CPU do controlador programável através de cabo USB, cabo Ethernet*3 ou cabo serial. O módulo de CPU pode ser reprogramado quantas vezes forem necessárias, para se adaptar a eventuais mudanças a serem feitas no controle desejado.

*1 Plano futuro para o GX works2.

*2 Atualmente denominada Structured Ladder no GX works2, conformidade prevista com IEC.

*3 Ethernet é uma marca comercial registrada da Xerox Corp.



Neste curso, o programa de exemplo utiliza Ladder Logic (uma das linguagens de programação de PLC mais populares). Embora o exemplo utilize um PLC de série L, o conteúdo deste curso aplica-se igualmente aos sistemas de série Q.

O método básico de controle é o mesmo também para a série MELSEC-F, mas algumas das operações e funções são diferentes.

1.1

Procedimento de construção do sistema do PLC

Este curso de e-learning abrange as etapas de criação do software (apresentadas em verde) necessárias para implementar um sistema de controlador programável.

Projeto do hardware

(1) Projeto do sistema Curso básico de MELSEC-Q/MELSEC-L



(2) Seleção do produto Curso básico de MELSEC-Q/MELSEC-L



(3) Preparação avançada Curso básico de MELSEC-Q/MELSEC-L



(4) Instalação e conexão elétrica Curso básico de MELSEC-Q/MELSEC-L



(5) Verificação da conexão elétrica ... Curso básico de MELSEC-Q/MELSEC-L

**Criação do software**

(6) Criação do programa Capítulo 2



(7) Programação Capítulo 3



(8) Depuração Capítulo 4



(9) Operação

Escopo deste curso

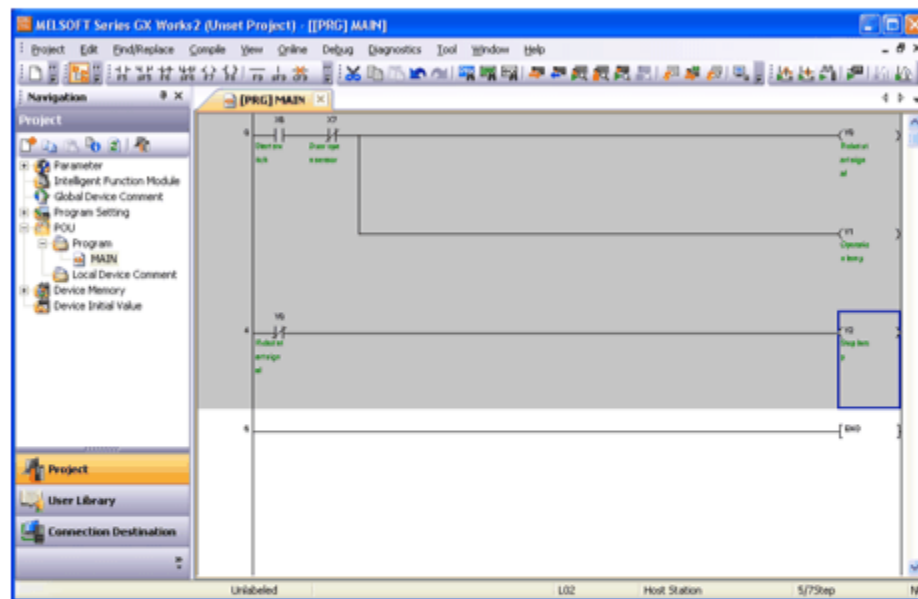
1.2

Requisitos para a programação

Este curso dá enfoque a como utilizar o software de engenharia GX Works2 de controlador programável para desenvolver o programa do sistema de exemplo.

A seção abaixo enumera algumas das principais funções do GX Works2.

- Gerenciamento de memória e arquivos
- Desenvolvimento de programas de controlador programável
- Gerenciamento da documentação dos programas (comentários, etc.)
- Leitura de dados de escrita (especialmente programas) de/para o módulo de CPU
- Verificação da operação dos programas
 - Simulação do software do hardware de PLC
 - Forçar a ativação/desativação de I/O
 - Monitorar a I/O e o status do endereço de memória
- Executar tarefas de manutenção e resolução de problemas



1.3

Configuração da tela do GX Works2

A configuração da tela do GX Works2 é apresentada abaixo.

Posicione o cursor do mouse em uma moldura vermelha para exibir sua respectiva função.

The screenshot displays the MELSOFT Series GX Works2 software interface. The main window shows a ladder logic diagram for a sub-program labeled [PRG] SUB. The diagram includes the following components:

- Input X6 (Stop Sw) with a normally closed contact.
- Input X7 (Door open sensor) with a normally open contact.
- Output Y0 (Robot start signal) with a coil.
- Output Y1 (Operation lamp) with a coil.
- Output Y2 (Stop lamp) with a coil.
- End of program (END).

A red rectangular box highlights a specific component in the diagram, which is a normally open contact labeled Y0 (Robot start signal). This box is positioned over the contact in the lower part of the diagram, where it is connected to the Y2 (Stop lamp) output.

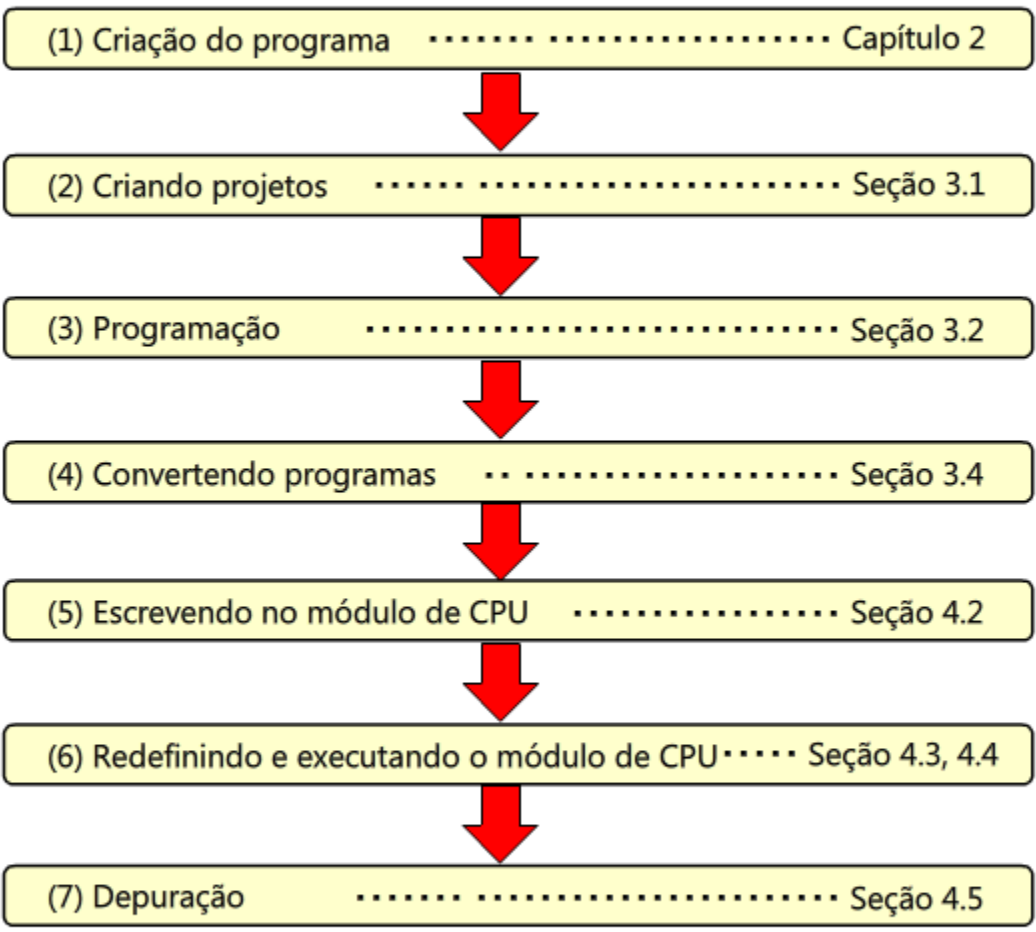
The interface also shows a navigation pane on the left with the following structure:

- Project
 - Parameter
 - Intelligent Function Module
 - Global Device Comment
 - Program Setting
 - POU
 - Program
 - MAIN
 - SUB
 - Local Device Comment
 - Device Memory
 - Device Initial Value
- Project
- User Library
- Connection Destination

The status bar at the bottom indicates the current step is 5/7 Step.

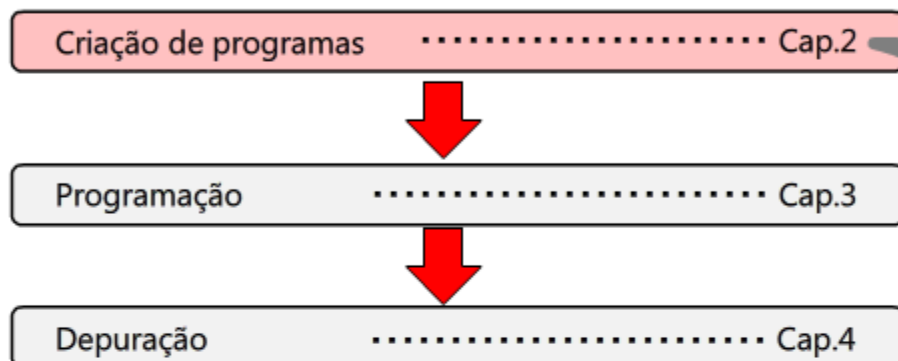
1.4 Procedimento de criação de programas de sequência

Crie um programa de sequência de acordo com o seguinte procedimento.



Capítulo 2 Criação de dados da tela

No Capítulo 2, você aprenderá a criar programas, inclusive a definir o conteúdo de controle e convertê-lo em um programa.



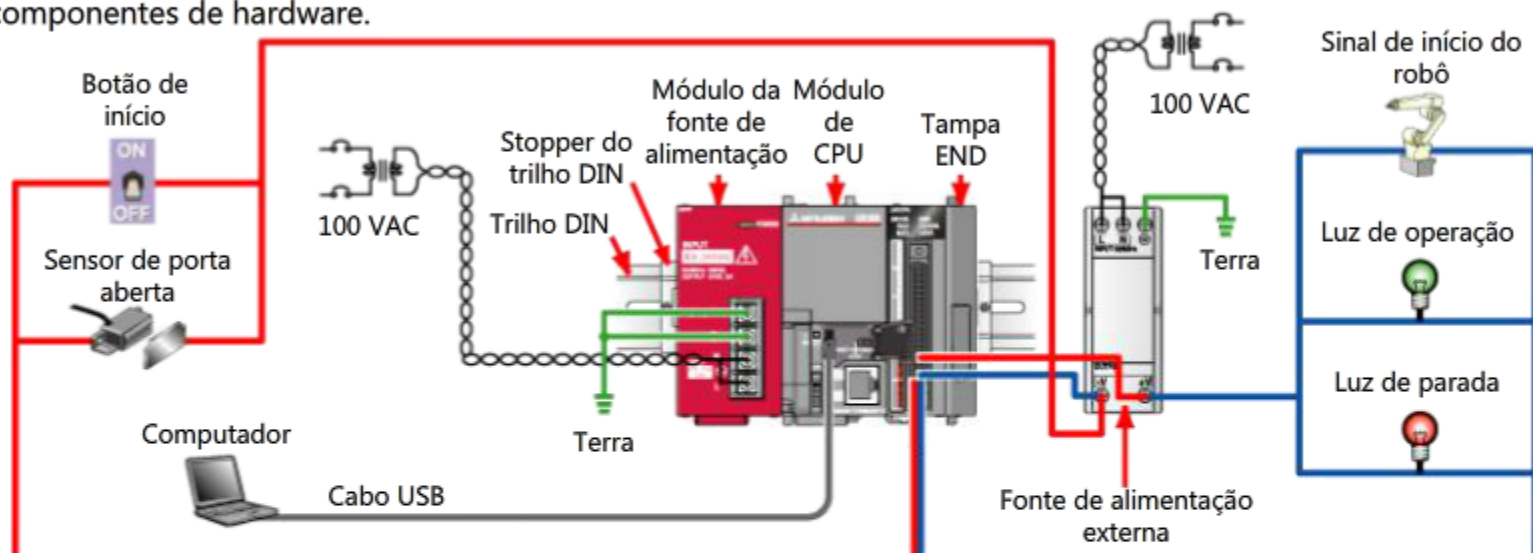
Etapas do aprendizado no Capítulo 2

- 2.1 Configuração do hardware de exemplo
Sistema usado para o curso
- 2.2 Definindo itens de controle
- 2.3 Criando uma tabela de correspondência de I/O
Dispositivos e números de dispositivos
- 2.4 Criando um programa

2.1 Configuração do hardware do sistema de exemplo utilizado no curso

Neste curso, você construirá um sistema de PLC (daqui para frente denominado "sistema de exemplo"), que liga o robô de acordo com um procedimento.

Um diagrama da configuração de hardware do sistema de exemplo é apresentado abaixo, com uma lista de componentes de hardware.



Item	Componente	Modelo	Descrição
Sistema de PLC	Módulo da fonte de alimentação	L61P	Alimenta os módulos, que incluem o módulo de CPU e I/O.
	Módulo de CPU	L02CPU	Controla o sistema de PLC.
	Tampa END	L6EC	Acoplada no lado direito do bloco do sistema.
	Cabo USB	MR-J3USBCBL3M	Conecta o computador onde o GX Works2 está instalado ao módulo de CPU.
	Computador	—	Executa o GX Works2.
Fonte de alimentação externa	—	—	Alimenta os dispositivos de I/O externos.
Equipamentos de I/O externos	Botão	—	Defina como ON (Ativar) para ligar o controle.
	Sensor	—	Detecta se a porta está aberta ou fechada.
	Robô	—	Opera de acordo com os sinais de controle.
	Duas luzes	—	Acendem-se de acordo com o status da operação.

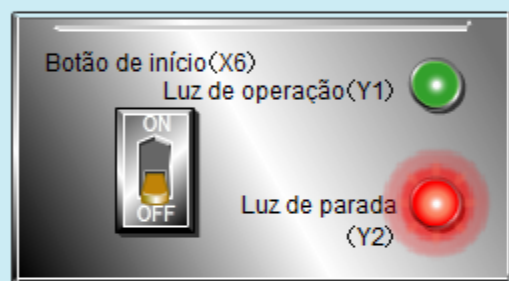
A primeira etapa da criação de um programa é identificar os dispositivos a serem controlados e os dispositivos I/O necessários para o controle desejado. No sistema de exemplo, é feito o controle da ativação e desativação da operação de um robô. O robô será impedido de ser ativado se a porta para a cerca de segurança estiver aberta, e será parado se a porta for aberta durante a operação.

Veja a animação abaixo para entender melhor como funciona o sistema de exemplo.

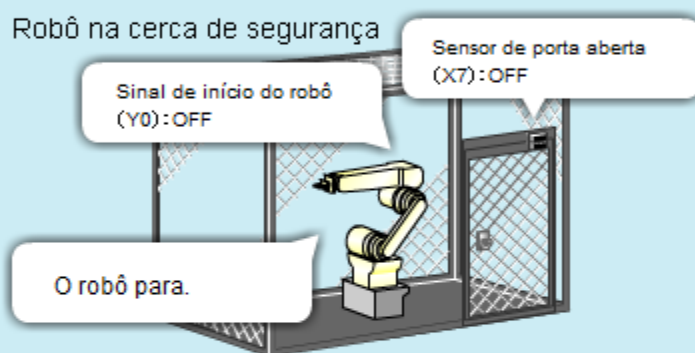
Operação do sistema de exemplo

Clique no interior do círculo vermelho

Painel de controle do robô



Robô na cerca de segurança



Quando se define o **botão de início (X6)** como OFF (Desativado), o **sinal de início do robô (Y0)** é desligado e o robô interrompe a operação. Simultaneamente, a **luz de operação (Y1)** do painel de controle se apaga, e a **luz de parada (Y2)** se acende.

Repetir



Anterior

2.3 Criando uma tabela de correspondência de dispositivos I/O e números de dispositivos

É recomendável criar uma tabela que inclua todos os dispositivos I/O e registros utilizados em um PLC e suas informações correspondentes, para qualquer programa que seja criado. Isso reduz a probabilidade de ocorrência de erros durante a criação e o processo de programação, servindo para melhorar a eficiência da programação. Se já existir uma tabela de correspondência para o sistema, como uma que tenha sido criada pela pessoa que configurou o hardware, utilize-a.

A tabela abaixo é uma tabela de correspondência para o sistema de exemplo utilizado neste curso

Nome do dispositivo I/O	Nº do dispositivo	Tipo de I/O	Tipo de dispositivo	Descrição
Botão de início	X6	Entrada	Bit	Este botão inicia ou para a operação do robô.
Sensor de porta aberta	X7	Entrada	Bit	Este sensor verifica se a porta da cerca de segurança do robô está aberta. Quando a porta é aberta, o sensor é ativado. Quando a porta é fechada, o sensor é desativado.
Sinal de início do robô	Y0	Saída	Bit	Quando esse sinal é ativado, o robô inicia a operação.
Luz de operação	Y1	Saída	Bit	Essa luz se acende quando o robô está operando.
Luz de parada	Y2	Saída	Bit	Essa luz se acende quando o robô está parado.

* Se forem usados dados de palavra, o valor inicial, o range de definição (limites inferior e superior), os tipo dos dados (assinado, real, etc.), e o comentário devem ser incluídos na tabela. Essas informações serão úteis para criar e modificar os programas.

2.4

Criando programas

Crie um programa usando a linguagem Ladder Logic, com base nos itens de controle e na tabela de correspondência de I/O. O programa ladder e a tabela de correspondência I/O criados para o sistema de exemplo são apresentados a seguir.

Programa Ladder

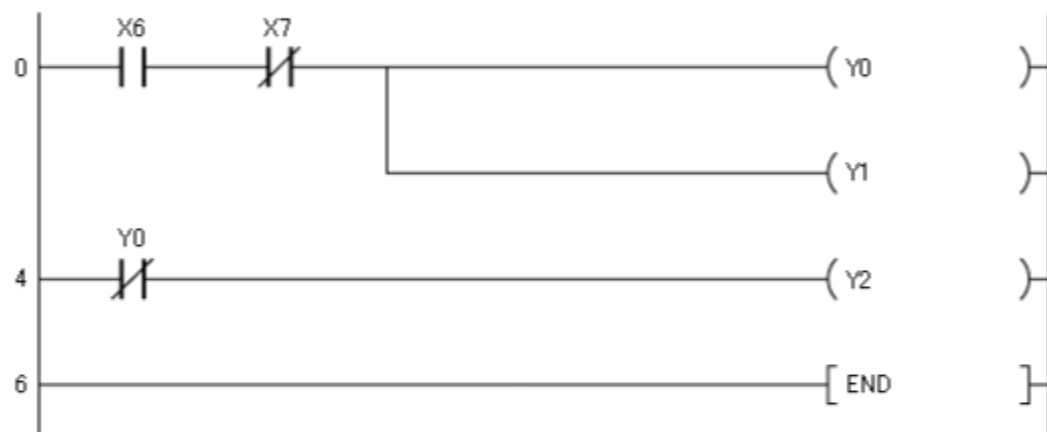
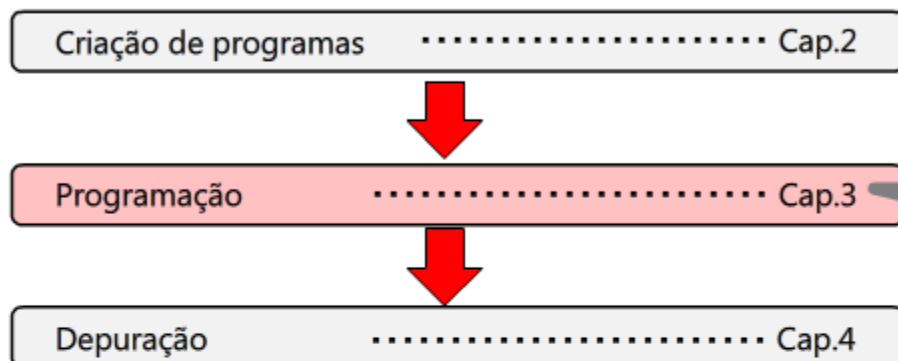


Tabela de correspondência de I/O

Nome do dispositivo I/O	Tipo	Nº do dispositivo
Botão de início	Entrada	X6
Sensor de porta aberta	Entrada	X7
Sinal de início do robô	Saída	Y0
Luz de operação	Saída	Y1
Luz de parada	Saída	Y2

Capítulo 3 Programação

No Capítulo 3, você aprenderá a programar o programa criado com o GX Works2.



Etapas do aprendizado no Capítulo 3

- 3.1 Criando projetos
- 3.2 Criando programas
- 3.3 Facilitando a compreensão dos programas
- 3.4 Convertendo programas para a forma executável
- 3.5 Salvando projetos

3.1

Criando projetos

A primeira etapa para se escrever um programa é criar um projeto.
O projeto é uma coleção de dados que o GX Works2 utiliza para gerenciar os programas.
A tabela a seguir enumera os principais componentes de um projeto.

Tipo dos dados	Descrição
Programa	O código fonte e o código compilado para operações de sequência da CPU.
Comentário	Um tipo de documentação apresentado dentro do programa. Consulte a Seção 3.3 "Facilitando a compreensão dos programas" para obter detalhes.
Parâmetro	Contêm a maioria ou todas as informações sobre definição e configuração de um sistema.
Configuração da transferência	As informações sobre a rota de conexão necessárias para estabelecer comunicações entre o sistema que executa o GX Works2 e o módulo de CPU.

Programa Ladder

O GX Works2 permite selecionar os dois seguintes tipos de projetos.
O programa de exemplo deste curso utiliza o tipo "simple project".

Tipo de projeto	Descrição
Simple project	Este tipo de projeto é compatível com projeto anteriores do GX Developer. Os simple projects podem ser convertidos em projetos estruturados posteriormente, mas o contrário não ocorre.
Projeto estruturado	Esses projetos podem utilizar uma linguagem de programação adicional, denominada Structured Ladder. Além disso, os programas podem ser separados em várias partes pequenas, e os pedaços de código utilizados frequentemente podem ser facilmente modularizados e reutilizados por meio de uma biblioteca de usuários. Da mesma forma, os labels podem ser modularizados para uma fácil reutilização. Isso pode aumentar a eficiência da programação e da depuração, especialmente para projetos muito grandes.

Labels

Os labels são nomes criados pelos usuários que se tornam as designações dos endereços dos dispositivos. Podem ser usados globalmente, localmente ou em todo o sistema, quando implementados juntamente com o MELSOFT Navigator. Os simple projects podem ser criados com ou sem o recurso de utilização de labels. No projeto de exemplo, os labels não são utilizados.

Para começar a criar o projeto de exemplo, faça as seguintes configurações.

Antes de criar um projeto, é necessário saber a série do controlador programável e o nome do modelo, bem como o tipo de projeto a ser utilizado.

Item	Descrição
Project type (Tipo de projeto)	O tipo de projeto determina quais recursos estão disponíveis quando se escrevem os programas. Nesse exemplo, escolha "simple project".
Use label (Usar label)	Se o recurso de escrever programas utilizando labels for necessário, assinale este item. O programa de exemplo não utiliza labels. Por isso, deixe esta caixa desmarcada.
PLC series (Série de PLC)	A série de PLC determina os modelos disponíveis para seleção na lista suspensa do tipo de PLC. Nesse exemplo, escolha "LCPU."
PLC type (Tipo de PLC)	O tipo de PLC determina a forma como o compilador converte os programas dos usuários em códigos da máquina. Escolha o modelo de PLC que será programado; nesse caso, "L02."
Programming language (Linguagem de programação)	A linguagem de programação determina o tipo do primeiro programa criado automaticamente (MAIN). Também é possível adicionar outros programas que utilizam outras linguagens, posteriormente. Nesse exemplo, escolha "Ladder".

Leia a próxima página, que apresenta a simulação do processo de criação de um novo projeto.

3.1

Criando projetos

MELSOFT Series GX Works2

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project



Project

User Library

Connection Destination

New Project

Project Type:

Simple Project

OK

Cancel

 Use Label

PLC Series:

LCPU

PLC Type:

L02

Language:

Ladder

Um novo projeto foi criado.

Clique em  para continuar.

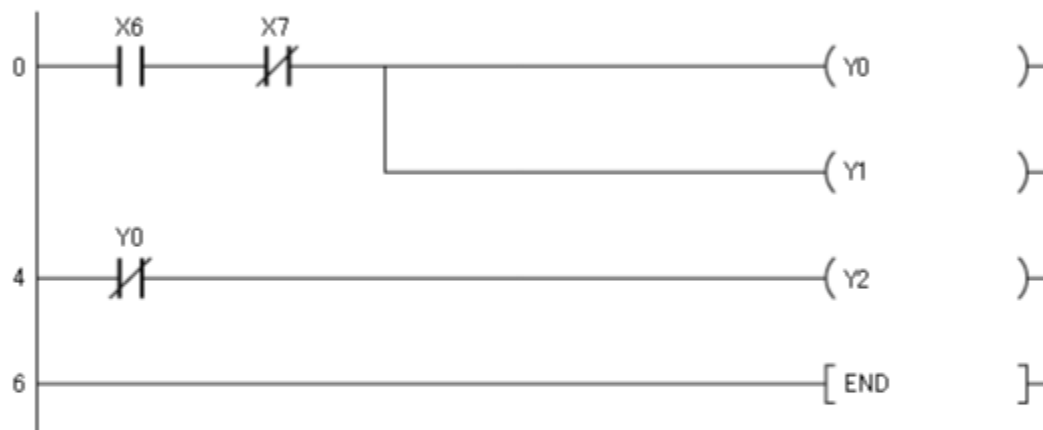
3.2 Criando programas

Depois de criarmos um projeto, vamos criar um programa.

Crie o programa a seguir e aprenda as operações básicas (inserir instruções, alterar, eliminar, copiar/colar e inserir/eliminar linha de regra).

O programa criado para o sistema de exemplo no Capítulo 2 é apresentado a seguir.

Programa para o sistema de exemplo



Na próxima página, experimente criar este programa usando a janela simulada.

3.2 Criando programas

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value


Unlabeled

L02

Host Station

0/15Step

NL

O programa de circuito em Ladder está concluído.
Clique em  para continuar.

3.3

Facilitando a compreensão dos programas

Em seu estado atual, a representação visual do programa contém apenas dispositivos, instruções, linhas e números de etapas. Quando se olha para um programa complexo, pode ser difícil determinar o que ele está fazendo.

- Erros de programação difíceis de encontrar, como números de dispositivo ou instruções incorretos.
- De forma geral, análise operacional, depuração e expansão de programas difíceis de executar.
- Se o criador do programa original não puder mais fazer a manutenção do programa, a tarefa de aprender como ele funciona pode ser avassaladora, e talvez impossível.

Contramedidas

Inclua a **documentação** no programa, para permitir que qualquer pessoa entenda rapidamente como o programa funciona. Como boa prática, todos os programadores devem adicionar comentários detalhados de seus programas, para que eles mesmos e outras pessoas possam entendê-los melhor.



O GX Works2 permite inserir três tipos de comentários.

Para obter mais detalhes, consulte o Manual de Simple Projects do GX Works2.

Tipo de comentário	Descrição do comentário
Comentário sobre o dispositivo	Insira até 32 caracteres a serem exibidos abaixo do dispositivo selecionado (I/O ou outro endereço da memória).
Instrução	Insira até 64 caracteres por instrução para serem adicionados no topo do bloco do ladder selecionado (acima do número da etapa). Cada bloco do ladder pode conter várias instruções.
Nota	Insira até 32 caracteres a serem exibidos acima da bobina selecionada ou a instrução da aplicação.

A próxima página simula o processo de adição de comentários sobre o dispositivo ao programa de exemplo.

3.3

Facilitando a compreensão dos programas

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project


- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

0 X6 Start switch X7 Door open sensor Y0 Robot start signal

4 Y0 Robot start signal Y2 Stop lamp

6 [END]

Unlabeled L02 Host Station 5/75Step

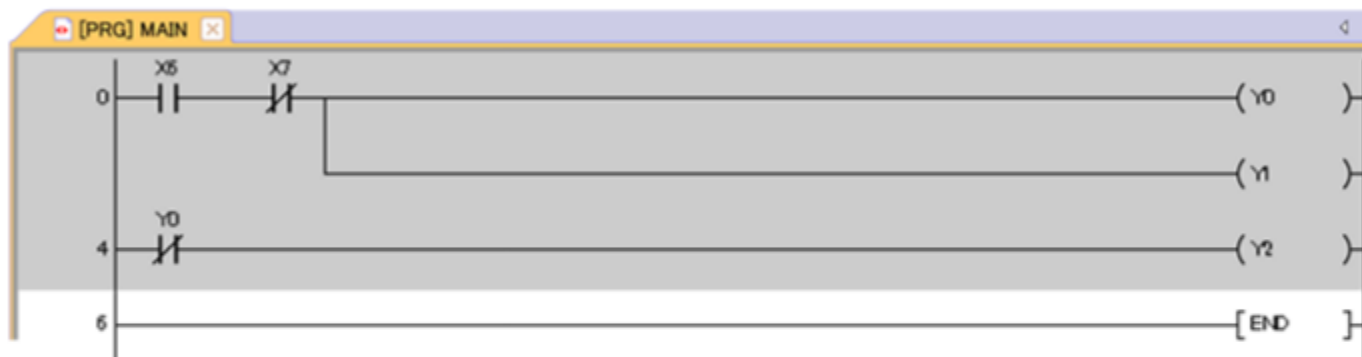
A entrada do comentário sobre o dispositivo foi concluída.
Clique em  para continuar.

3.4

Convertendo programas para a forma executável

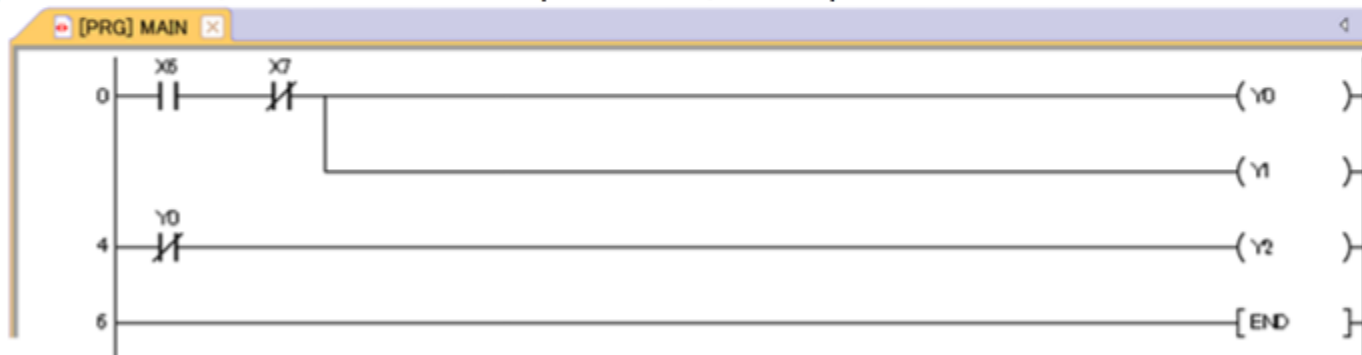
Depois de terminar o programa, você precisa convertê-lo para uma forma que possa ser executada no módulo de CPU. Os programas não convertidos não podem ser executados ou salvos.

A cor de fundo dos programas não convertidos é cinza, como mostra a imagem abaixo.



Converter

Após a conversão, a cor de fundo muda para branco, como apresentado abaixo.



Na próxima página, experimente converter o programa usando a janela simulada.

3.4

Convertendo programas para a forma executável



MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled

0 X8 X7

Start switch Door open sensor

Y0 Robot start signal


Y1 Operation lamp

4 Y0 Robot start signal

Y2 Stop lamp

6 [END]

Quando o programa é convertido, a cor de fundo muda de cinza para branco.

O programa é convertido.
Clique em  para continuar.

Unlabeled L02 Host Station 5/75Step

3.5 Salvando projetos

Depois que a conversão do programa for concluída, salve o projeto, incluindo os programas. Se o GX Works2 for encerrado sem que o projeto seja salvo, os programas associados serão descartados, por isso recomendamos salvar o projeto regularmente. Ao salvar um novo projeto, especifique os seguintes tipos de informações sobre o projeto. (Isso não é necessário para salvar sobrescrevendo).

Você deve incluir informações que tornem mais fácil que outras pessoas entendam o conteúdo de controle do programa, o nome do sistema, etc.

Item	Obrigatório	Descrição
Save destination path (Salvar caminho de destino)	✓	Especifique a pasta em que a área de trabalho será alocada.
Workspace/project list (Área de trabalho/lista de projetos)		Se já existirem uma ou mais áreas de trabalho na pasta especificada em "Save destination path" (Salvar caminho de destino), as áreas de trabalho existentes serão listadas.
Workspace name (Nome da área de trabalho)	✓	Especifique o nome da área de trabalho, usando até 128 caracteres.
Project name (Nome do projeto)	✓	Especifique o nome do projeto, usando até 128 caracteres.
Title (Título)		Especifique o título do projeto, usando até 128 caracteres. Esse parâmetro é útil quando se quer atribuir um nome longo, que não cabe no campo "Project name" (Nome do projeto).

A **área de trabalho** é uma pasta para gerenciamento de vários projetos.

A seção abaixo mostra um exemplo da utilização de uma área de trabalho. (Os projetos são gerenciados para cada tipo de veículo da linha de fabricação de automóveis).

Workspace name (Nome da área de trabalho)	Project name (Nome do projeto)	Title (Título)
Linha de fabricação de automóveis	Linha de fabricação do Tipo A	Programa de operação normal para controlar a linha de fabricação do tipo A
	Linha de fabricação do Tipo B	Programa de operação normal para controlar a linha de fabricação do tipo B
	Linha de fabricação do Tipo C	Programa de operação normal para controlar a linha de fabricação do tipo C

- Notas:**
- Se um projeto contendo um programa não convertido for salvo, apenas o programa não convertido será descartado. Antes de salvar um projeto, faça a conversão do programa, como visto na Seção 3.4.
 - Especifique o caminho de destino, o nome da área de trabalho e o nome do projeto, de forma que o número total de caracteres não ultrapasse 150.

Na próxima página, experimente salvar o projeto usando a janela simulada.

3.5

Salvando projetos

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled

L02

Host Station


6/75Step

NL

0 Start Switch (X6) Door open sensor (X7) Robot start signal (Y0)

4 Robot start signal (Y0) Stop lamp (Y2)

6 END

O projeto é salvo.
Clique em  para continuar.

Capítulo 4 Depuração

No Capítulo 4, você aprenderá a escrever programas de sequência no módulo de CPU e a depurá-los.

Criação de programas Cap.2



Programação Cap.3



Depuração Cap.4

Etapas do aprendizado no Capítulo 4

4.1 Depuração

4.1.1 Depuração de um programa sem utilizar o módulo de CPU

4.1.2 Alteração do status de um dispositivo I/O

4.1.3 Monitoração do status do dispositivo

4.2 Escrevendo programas no módulo de CPU

4.3 Ativando programas escritos

4.4 Executando programas

4.5 Depurando programas

4.6 Verificando a operação do sistema de PLC

4.7 Operando o sistema de PLC

4.8 Conclusão

Quando se termina de escrever um programa ou segmento de programa, é necessário testar o código para verificar se ele está operando como desejado.

Os defeitos de software (quando o código escrito não apresenta o desempenho desejado) são chamados de "bugs", e o processo de encontrar a causa do comportamento não desejado e corrigi-lo é chamado de "depuração."

O teste e a depuração são etapas essenciais da criação de programas.

Particularmente nos controladores programáveis pois, se houver bugs, eles podem causar a parada do sistema, danos ao equipamento ou outros acidentes.

A tabela a seguir lista algumas das funções do GX Works2 que podem ajudar no processo de depuração.

Nome da função	Descrição
Simulator (Simulador)	Essa função é usada para simular a execução de um programa, mesmo sem um módulo de CPU. Essa função pode ser usada para depuração em um ambiente que não tenha módulo de CPU disponível.
Monitor	Essa função permite monitorar o status de execução e o status de cada dispositivo durante a execução do módulo de CPU. Existem várias funções de monitoração disponíveis, dependendo da aplicação, como monitoração no ladder, monitoração apenas de dispositivos registrados, e monitoração de todos os dispositivos em lote.
Change current value (Alterar valor atual)	Essa função serve para forçar a alteração do status do dispositivo (bit; ON ↔ OFF, palavra: valor atual) durante a execução do módulo de CPU. Essa função é útil para mudar o valor atual de um dispositivo de palavra ou o status de um relé interno.
Forced input output registration/cancellation (Registro/cancelamento forçado de entrada/saída)	Essa função força a alteração do status (ON ↔ OFF) de um dispositivo I/O registrado, durante a execução do módulo de CPU. Para a depuração ou verificação da operação com um módulo de CPU sozinho, essa função pode ser usada como substituta de um botão.

Essas funções sobre o processo de depuração são explicadas mais detalhadamente nas demais partes deste Capítulo.

Notas sobre a depuração

Não efetue tarefas de depuração se o controlador programável estiver conectado a dispositivos I/O físicos.

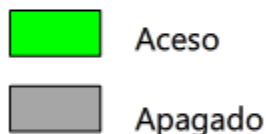
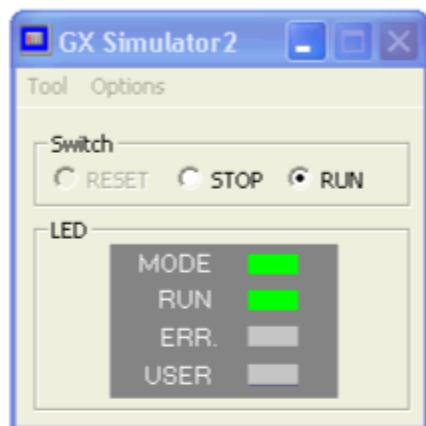
Os bugs do programa, dispositivos I/O forçados, ou alterações do valor da palavra podem causar danos nos equipamentos externos ou algo pior.

Se um sistema de PLC desconectado não estiver disponível, utilize a função de simulador.

4.1.1

Depuração de um programa sem utilizar o módulo de CPU

Se um módulo de CPU não estiver disponível para depuração, utilize a **função de simulador**. Um programa pode ser executado em um módulo de CPU virtual oferecido pelo software, sem utilizar um módulo de CPU real.



Item	Estado	Descrição
Botão	RUN (EXECUTAR)	Executa o módulo de CPU virtual.
	STOP (PARAR)	Para o módulo de CPU virtual.
	RESET	Efetua o reset do módulo de CPU virtual. (Ativado apenas no estado STOP)
LED	MODE (MODO)	Indica o status de MODE (MODO) da CPU virtual.
	RUN (EXECUTAR)	Indica o status de execução da CPU virtual. •Aceso: estado RUN (EXECUTAR) •Apagado: estado STOP (PARAR)
	ERR	Indica o status de erro do módulo de CPU virtual. Se houver um erro, o LED se acenderá ou piscará.
	USER (USUÁRIO)	Indica se ocorreu um erro do usuário na CPU virtual. Acende-se ou pisca quando ocorre um erro.

Notas sobre a utilização da função de simulador

- A depuração por meio da função de simulador não garante que o programa de sequência funcionará corretamente após a depuração.
- A função de simulador executa a entrada/saída de dados com módulos de I/O utilizando a memória de simulação. A função não aceita algumas instruções, funções e memória do dispositivo. Portanto, os resultados da operação com a função de simulador podem ser diferentes dos obtidos com o módulo de CPU real.

Na próxima página, experimente usar a função de simulador com a janela simulada.

4.1.1

Depuração de um programa sem utilizar o módulo de CPU

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled


L02

Host Station

6/75Step

NL

```
graph TD
    R0[0] --- X6[Start Switch X6]
    R0 --- X7[Door open sensor X7]
    R0 --- Y0[Robot start signal Y0]
    R4[4] --- Y0_2[Robot start signal Y0]
    R4 --- Y2[Stop lamp Y2]
    R6[6] --- END[END]
```

Agora você aprendeu como utilizar o recurso de simulação.
Clique em  para continuar.

4.1.2

Alteração do status de um dispositivo I/O

Ao depurar um programa de sequência com um módulo de CPU ao qual nenhum dispositivo I/O está conectado, ou utilizando a função de simulador, utilize a função **Forced input output registration/cancellation (Registro/cancelamento forçado de entrada/saída)** para alterar o estado ON/OFF (Ativado/Desativado) de um dispositivo I/O.

O status dos dispositivos I/O registrados pode ser forçadamente alterado para ON (Ativado) ou OFF (Desativado) com o software.

(séries MELSEC-Q e MELSEC-L): Na tela "Forced Input Output Registration/Cancellation" (Registro/Cancelamento da saída e entrada reforçada)

(série MELSEC-F): Na tela "Modify Value" (Modificar valor)



tela Forced Input Output Registration/Cancellation (Registro/Cancelamento de saída e entrada reforçada) (séries MELSEC-Q e MELSEC-L)

tela Modify Value (Modificar valor) (série MELSEC-F)

Para alterar os estados de outros dispositivos

Para alterar o dispositivo atual de um dispositivo de palavra ou o estado ON/OFF (Ativado/desativado) de um relé interno, utilize a **função alterar valor atual**.

Para obter detalhes, consulte o manual.

4.1.3

Monitoração do status do dispositivo

Quando a simulação é iniciada, a monitoração começa automaticamente. Para entrar no modo de monitoração quando conectado a uma CPU de controlador programável real, basta clicar em Online, Monitor e Start Monitoring (Iniciar monitoração). Ou então, utilize o atalho do teclado, F3.

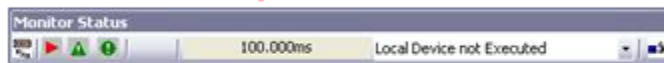
Durante o modo de monitoração, os valores e status de todos os dispositivos usados no programa podem ser vistos sobrepostos ao código do programa. Isso permite que o usuário veja os valores mudando, incluindo os efeitos da utilização da função "Forced input output registration/cancellation (Registro/cancelamento forçado de entrada/saída)".

Além disso, a barra **Monitor Status** (Status do monitor) aparece, contendo informações básicas para determinar o status da CPU ou da CPU virtual. Consulte a tabela abaixo para entender as informações fornecidas pela barra **Monitor Status** (Status do monitor).

Quando conectado ao módulo de CPU



Ao utilizar a função de simulador



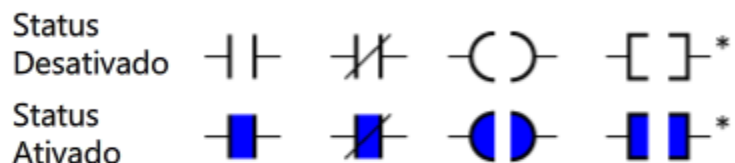
Estado	Ícone/indicação	Descrição
Connection status (Status da conexão)	Quando conectado ao módulo de CPU	Exibe o status da conexão com o módulo de CPU ou a função de simulador.
	Ao utilizar a função de simulador	
RUN/STOP status (Status de EXECUTAR/PARAR)	RUN (EXECUTAR)	Exibe o status da execução da CPU (RUN - EXECUTAR ou STOP - PARAR).
	STOP (PARAR)	
ERR. status (Status de ERRO)	ERR. apagado	Exibe o status de erro do módulo de CPU.
	ERR. aceso	
	↔ ERR piscando	
USER status (Status de USUÁRIO)	USER apagado	Exibe o status de erro do usuário do módulo de CPU.
	USER aceso	
	↔ USER piscando	
Scan time (Tempo de scan)		Exibe o tempo máximo de scan do módulo de CPU que está sendo monitorado.
Unsupported instruction presence/absence status (Status de presença/ausência de instrução não suportada)	Existe uma instrução não suportada.	Exibe se existe uma instrução não suportada quando se executa a função de simulador.
	Não existe uma instrução não suportada.	Clique nesse ícone para abrir a janela Unsupported Instruction/Device (Instrução/dispositivo não suportado).

4.1.3 Monitoração do status do dispositivo

Durante o modo de monitoração, o status atual de todos os dispositivos do programa tornam-se visíveis.

Exibição do status do dispositivo de bit (Ativado/Desativado)

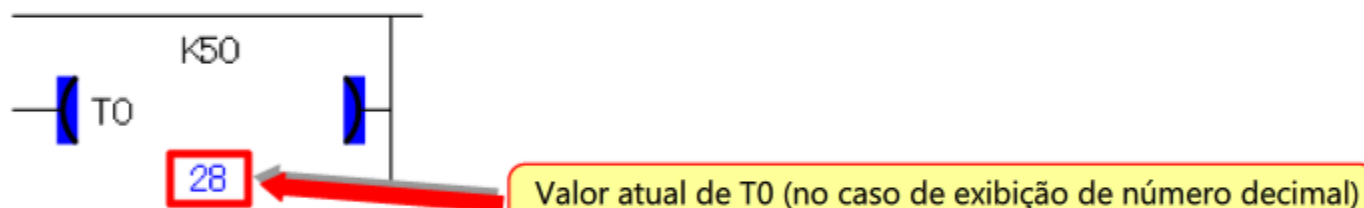
O status Ativado/Desativado é exibido durante a monitoração, da forma apresentada abaixo.



* Esse tipo de exibição aplica-se apenas a instruções SET, RST, PLS, PLF, SFT, SFTP, MC e de comparação de tipos de contato. Note que, para a instrução RST, é exibido apenas o status Ativado/Desativado.

Exibição do valor atual do dispositivo de palavra (exibição de números decimais/hexadecimais)

O valor atual durante a monitoração é exibido da forma abaixo.



Monitorando apenas dispositivos específicos

Quando se monitora um programa muito grande ou complexo, pode ser uma boa ideia monitorar apenas alguns dispositivos de interesse. Para isso, o GX Works2 inclui janelas de observação que permitem que o usuário adicione facilmente os dispositivos nos quais está interessado, veja seu status atual e modifique seus valores durante a monitoração. Para obter detalhes, consulte o Manual de Operação do GX Works2 (Comum).

Watch 1					
Device/Label	Current Value	Data Type	Class	Device	Comment
X7	--	Bit		X7	Door open sensor
Y0	--	Bit		Y0	Robot start signal
Y1	--	Bit		Y1	Operation lamp
Y0	--	Bit		Y0	Robot start signal
Y2	--	Bit		Y2	Stop lamp
Y0	--	Bit		Y0	Robot start signal

4.2

Escrevendo programas no módulo de CPU

Antes de fazer qualquer depuração usando um módulo de CPU real, coloque a CPU no **modo de PARADA**, certifique-se de que a conexão à CPU tenha sido feita e escreva os programas e parâmetros na memória do programa.

Como se vê na tela abaixo, as principais funções da janela **Write to PLC (Escrever no PLC)** permitem que o usuário selecione os arquivos que deseja escrever, escolha seu local e confirme a capacidade de memória da CPU. Os três botões acima da lista de arquivos permitem que o usuário selecione rapidamente os arquivos a serem escritos. O mais comum, utilizado na simulação a seguir, é **"Parameter+Program (Parâmetro + programa)."**

Lista de arquivos

Capacidade de memória

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
Robot_Control						
PLC Data					Program Memory/D...	
Program(Program File)		<input checked="" type="checkbox"/>				
Parameter		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2256 Bytes
PLC(Network/Remote Password/Switch Setting)		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2936 Bytes
Global Device Comment		<input type="checkbox"/>				
COMMENT		<input type="checkbox"/>		2010/08/05 15:25:26		
Device Memory		<input type="checkbox"/>				
HAZU		<input type="checkbox"/>		2010/08/05 15:22:25		

Necessary Setting: No Setting / Already Set Set if it is needed: No Setting / Already Set

Writing Size: 5,192Bytes Free Volume: 76,728 Use Volume: 5,192Bytes Refresh

Related Functions: c.c

Execute Close

Na próxima página, experimente escrever no módulo de CPU usando a janela simulada.

4.2

Escrevendo programas no módulo de CPU

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled

L02

Host Station


6/75Step

NL

0 Start Switch (X6) Door open sensor (X7) Robot start signal (Y0)

4 Robot start signal (Y0) Stop lamp (Y2)

6 END

O programa é escrito no módulo de PLC.
Clique em  para continuar.

4.3 Ativando programas escritos

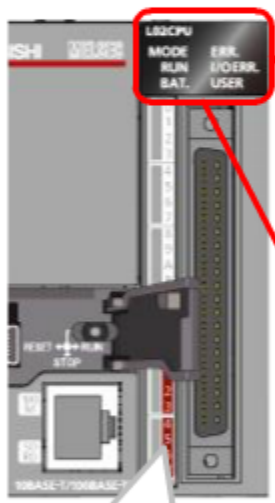
(série MELSEC-F): A seguinte operação não é necessária.

(séries MELSEC-Q e MELSEC-L): A seguinte operação é necessária.

Depois escrever um programa no módulo de CPU, faça o **reset** do módulo de CPU.
Os programas escritos só podem ser ativados quando se faz o reset do módulo de CPU.

* Essa operação não é necessária se a função de simulador for usada para a depuração.

Faça o reset do módulo de CPU da seguinte forma:



Botão RESET/STOP/RUN (RESET/PARAR/EXECUTAR)

- (1) Pressione e mantenha pressionado o botão RESET/STOP/RUN (RESET/PARAR/EXECUTAR) do painel dianteiro do módulo de CPU na posição RESET (por 1 segundo ou mais).

[Reset em andamento]

L02CPU	MODE (MODO) : Aceso em verde
MODE ■ ERR.	RUN (EXECUTAR) : Apagado
RUN I/OERR.	ERR. : Piscando
BAT. USER	



- (2) Solte o botão depois que o LED MODE (MODO) e o LED ERR. piscando se apagarem.

[Reset concluído]

L02CPU	MODE (MODO) : Aceso em verde
MODE ■ ERR.	RUN (EXECUTAR) : Apagado
RUN I/OERR.	ERR. : Apagado
BAT. USER	



- (3) O botão volta para a posição STOP (PARAR) para concluir o reset.

4.4 Executando programas

séries MELSEC-Q e MELSEC-L

Depois de fazer o reset, execute o programa.
Coloque o módulo de CPU no **status RUN (EXECUTAR)**, como se segue, para executar o programa.

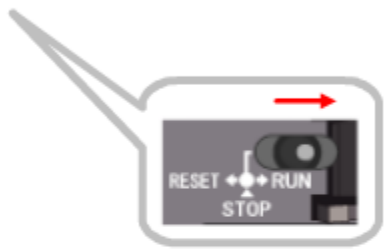
* Essa operação não é necessária se a função de simulador for usada para a depuração.

(1) Vire o botão RESET/STOP/RUN (RESET/PARAR/EXECUTAR) do painel dianteiro do módulo de CPU para a posição RUN (EXECUTAR).



Exibição do LED no status STOP (PARAR)

L02CPU	MODE (MODO) :Aceso em verde
MODE	ERR.
RUN	I/OERR.
BAT.	USER
RUN (EXECUTAR) :Apagado	



(2) Se o LED RUN (EXECUTAR) se acender em verde, o programa está sendo executado normalmente.

Exibição do LED no status RUN (EXECUTAR)

L02CPU	MODE (MODO) :Aceso em verde
MODE	ERR.
RUN	I/OERR.
BAT.	USER
RUN (EXECUTAR): Aceso em verde	

série MELSEC-F

Depois de escrever um programa na unidade principal, altere a unidade principal para o estado RUN (EXECUTAR) conforme a seguir para executar o programa. (A operação de reset não é necessária.)

(1) Gire o interruptor RUN/STOP (EXECUTAR/PARAR) no painel frontal da unidade principal para a posição RUN (EXECUTAR).



Exibição do LED no status STOP (PARAR)

(2) Se o LED RUN (EXECUTAR) acender, o programa será executado normalmente.



Exibição do LED no status RUN (EXECUTAR)

4.5

Depurando programas

Depois de executar o módulo de CPU, utilize a função "Forced input output registration/cancellation (Registro/cancelamento forçado de entrada/saída)" para mudar o status de cada dispositivo e monitorar o resultado (saída) no ladder.

(Exemplo de tela das séries MELSEC-Q e MELSEC-L)

The screenshot shows the MELSOFT GX Works2 interface. A dialog box titled "Forced Input Output Registration/Cancellation" is open, displaying a table for device registration. The table has columns for No., Device, and ON/OFF status. The first two rows are populated: No. 1 with Device X6 and ON/OFF ON; No. 2 with Device X7 and ON/OFF OFF. The rest of the rows are empty. The dialog also includes buttons for "Register FORCE ON", "Register FORCE OFF", "Cancel Registration", "Update Status", "Batch Cancel Registration", and "Close".

No.	Device	ON/OFF	No.	Device	ON/OFF
1	X6	ON	17		
2	X7	OFF	18		
3			19		
4			20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

Na próxima página, experimente depurar o programa usando a janela simulada.

4.5

Depurando programas

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

1.000ms Local Device not Executed

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

[[PRG] MAIN

0 Start Switch Door open sensor


Y0 Robot start signal

Y1 Operation lamp

4 Y0 Robot start signal

Y2 Stop lamp

6 END

A depuração do programa foi concluída.
Clique em  para continuar.

Unlabeled L02 Host Station 6/75Step

4.6

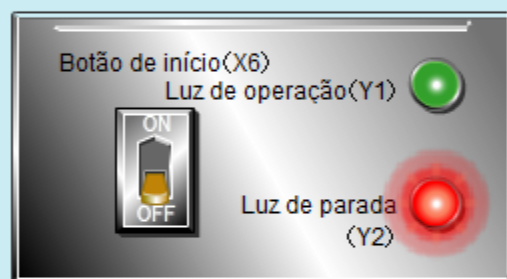
Verificando a operação do sistema de PLC

Depois de terminar a depuração do programa, escreva-o no sistema de PLC real, para finalmente verificar sua operação. Opere o equipamento de I/O real para confirmar se o programa funciona da forma planejada. Mesmo durante a operação do equipamento de I/O, o status de cada dispositivo pode ser verificado com a função de monitoração do GX Works2.

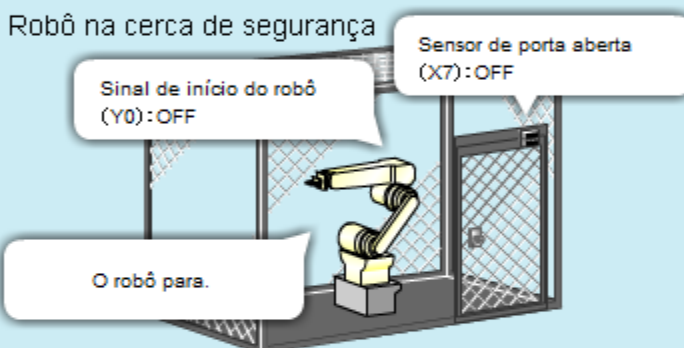
Operação do sistema de exemplo

 Clique no interior do círculo vermelho

Painel de controle do robô



Robô na cerca de segurança

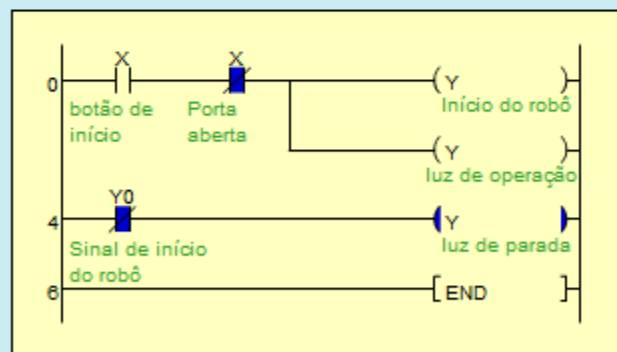


Quando se define o **botão de início (X6)** como OFF (Desativado), o **sinal de início do robô (Y0)** é desligado e o robô interrompe a operação. Simultaneamente, a **luz de operação (Y1)** do painel de controle se apaga, e a **luz de parada (Y2)**

Repetir



Anterior

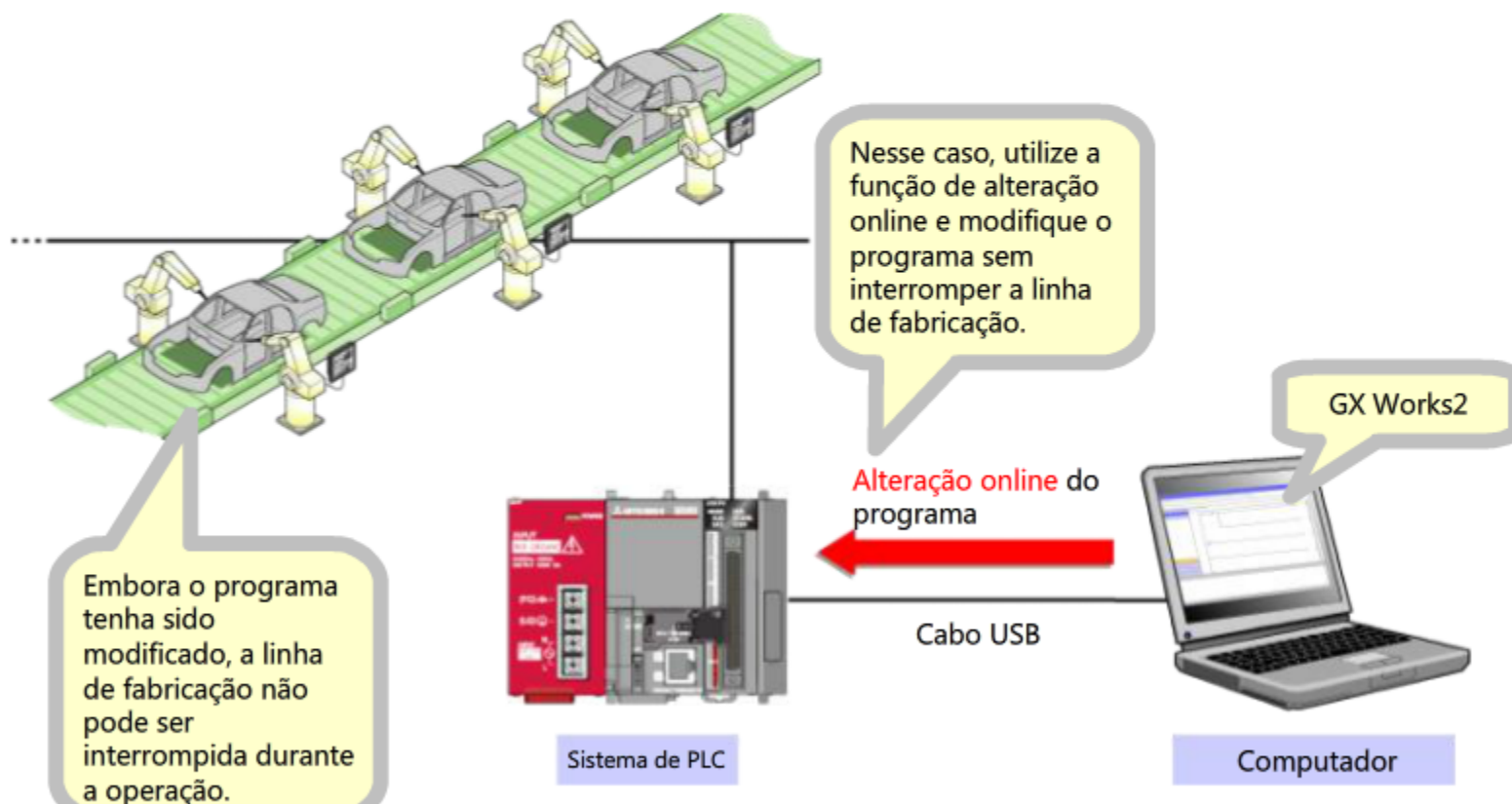


Depois de terminar a verificação da operação, execute o sistema de PLC para iniciar a operação.

Se o programa precisar ser modificado no sistema de execução

Pode ser necessário modificar o programa, por exemplo, para corrigir um bug ou expandir o sistema, após o início da operação do sistema. Normalmente, o sistema (módulo de CPU) precisa ser parado para se escrever um programa modificado, mas isso nem sempre é possível. Para solucionar esse problema, o GX Works oferece uma função de alteração online, utilizada para escrever programas se, para o módulo de CPU em execução.

Exemplo: Linha de fabricação de automóveis funcionando 24 horas



Na próxima página, experimente usar a função de alteração online com a janela simulada.

4.7

Operando o sistema de PLC

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

1.000ms Local Device not Executed

Navigation

Project


- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
- Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

```
graph TD
    subgraph Rung_0 [0]
        X0[X0 Start Switch] --- Y0[Y0 Robot start signal]
    end
    subgraph Rung_4 [4]
        Y0[Y0 Robot start signal] --- Y2[Y2 Stop lamp]
    end
    subgraph Rung_6 [6]
        END[END]
    end
```

A alteração online do programa modificado foi concluída.
Clique em  para continuar.

Unlabeled L02 Host Station 2/75Step

4.8**Conclusão**

Terminamos a explicação básica sobre a criação do software do controlador programável.

Neste curso, você aprendeu:

- Os elementos necessários para a programação de um sistema de PLC
- Algumas diretrizes básicas para a criação de programas, incluindo o uso de comentários
- Como utilizar o GX works2 para executar as tarefas básicas de programação de um PC
- Algumas técnicas utilizadas para depurar programas de PLC

Agora que você concluiu todas as lições do curso **PLC Introdução ao GX Works2**, está pronto para fazer o teste final. Se tiver qualquer dúvida sobre os tópicos abrangidos, aproveite esta oportunidade para revê-los.

O Teste Final é composto por 5 perguntas (15 itens).

Você pode fazer o teste final quantas vezes desejar.

Como é feita a pontuação do teste

Depois de selecionar a resposta, não se esqueça de clicar no botão **Resposta**. Sua resposta será perdida se você continuar sem clicar nesse botão. (O sistema assumirá que essa pergunta não foi respondida).

Resultados da pontuação

O número de respostas corretas, o número de perguntas, a porcentagem de respostas corretas e o resultado (aprovado/reprovado) aparecem na página de pontuação.

Respostas corretas: 2

Total de perguntas: 9

Porcentagem: 22%

Para passar no teste, você precisa responder corretamente a 60% das perguntas.

Continuar

Rever

Repetir

- Clique no botão **Continuar** para sair do teste.
- Clique no botão **Rever** para rever o teste. (Verificar a resposta correta)
- Clique no botão **Repetir** para refazer o teste.

O programa que estava a seu encargo foi assumido por outra pessoa, que teve dificuldades em entender os itens de controle. Qual seria a contramedida adequada para evitar esse problema?

- Usar a função de comentário do GX Works2, adicionar um título e explicação apropriados ao programa.
- Explicar oralmente os itens de controle para o novo encarregado.
- Evitar utilizar programas grandes e complexos.
- Transferir a tabela de correspondência para dispositivos I/O e números de dispositivos juntamente com o programa.

Gravar Pontuação

Voltar

Complete o procedimento de programação adequado.

Etapa 1 Criação do programa

Etapa 2 (Q1)

Etapa 3 (Q2)

Etapa 4 Converter programas

Etapa 5 Salvar projetos

Etapa 6 (Q3)

Etapa 7 (Q4)

Etapa 8 Executar o módulo de CPU (RUN - EXECUTAR)

Etapa 9 (Q5)

Etapa 10 Verificar a operação do sistema de PLC

Gravar Pontuação

Voltar

Preencha os espaços das seguintes explicações do que deve ser feito após a conclusão de um programa.

Depois que o programa é escrito, deve ser testado para confirmar que funciona da forma esperada.

Um () (quando se escreve código que não executa a função pretendida) é denominado

um () e o processo de detecção da causa e sua correção é denominado

().

Esse processo é uma etapa essencial da criação de programas.

Selecione a aplicação de apropriada de cada função do GX Works2.

Função	Aplicação
Simulação	--Select--
Forced input output registration/cancellation (Registro/cancelamento forçado de entrada/saida)	--Select--
Change current value (Alterar valor atual)	--Select--
Ladder monitor (Monitorar ladder)	--Select--
Watch (Observar)	--Select--

Gravar Pontuação

Voltar

Selecione a descrição correta da função de alteração online.

- A função para automaticamente CPU, escreve um programa na CPU, e automaticamente aciona a CPU.
- A função compara o programa no módulo de CPU em execução com o programa aberto pelo GX Works2.
- A função permite escrever um programa no módulo de CPU após a parada segura do módulo de CPU
- A função permite escrever um programa no módulo de CPU em execução sem pará-lo.

Gravar Pontuação

Voltar

Teste**Pontuação no Teste**

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

Respostas corretas: **0**

Total de perguntas: **5**

Porcentagem: **0%**

Você não passou no teste.

Você concluiu o curso **PLC Introdução ao GX Works2**.

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.

Rever

Fechar