

APOSTILA DE TREINAMENTO

GX Works3

TPB-GXW3



www.mitsubishielectric.com/fa/br_pt/

Data da Revisão	Nome do Arquivo	Revisão
Jan/2018	Treinamento GX Works3(A)	1ª Edição
Jun/2018	Treinamento GX Works3(B)	Revisão de exercício Anexo de criação de FB's Anexo de criação de descrição de capítulos e lista de manuais relevantes.

Contato de Suporte Técnico

Assistência telefônica disponível em (11) 4689-3000.

Assistência por e-mail disponível em cat@mitsubishielectric.com.br.

Ressalva

Este manual não implica em garantia ou implementação de direitos para propriedade industrial ou implementação de outros direitos. A Mitsubishi Electric não é responsável por problemas de propriedade industrial causados pelo uso ou mau uso do conteúdo deste manual.

© 2015 Mitsubishi Electric do Brasil Com. e Serviços Ltda.

Sumário

Apresentação da Aula	6
<i>Objetivos do Curso.....</i>	<i>6</i>
<i>Pré-requisitos.....</i>	<i>6</i>
<i>Duração do Curso.....</i>	<i>6</i>
1. INTRODUÇÃO AO GX WORKS3	9
1.1 <i>Introdução</i>	9
1.2 <i>Funcionalidades do GX Works3</i>	10
2. PROJETO NO GX WORKS3	13
2.1 <i>Criando um projeto novo</i>	13
2.2 <i>Compilação do Programa</i>	16
2.2.1 <i>Árvore de projeto</i>	16
2.2.2 <i>Compilação do Projeto no GX Works3</i>	17
2.2.3 <i>Compilação do programa no GX Works3</i>	18
2.3 <i>Comunicação do GX Works3 com o PLC</i>	19
2.3.1 <i>Definição da rota de comunicação</i>	19
2.3.2 <i>Escrita de programa para o PLC</i>	23
2.4 <i>Habilitando o Simulador do PLC.....</i>	24
3. OPERAÇÕES OFFLINE/ONLINE.....	26
3.1 <i>Operação de programa ladder.....</i>	26
3.2 <i>Modos de leitura e escrita</i>	27
3.3 <i>Monitoração de valores de dispositivos/labels.....</i>	31
3.3.1 <i>Device/Buffer Memory Batch Monitor.....</i>	32
3.3.2 <i>Watch.....</i>	33
3.4 <i>Alteração de valores de dispositivos/labels.....</i>	35

3.5	<i>Alteração de ladder online</i>	36
3.6	<i>Variáveis do PLC: Dispositivos e labels</i>	37
3.6.1	<i>Dispositivos (Devices)</i>	37
3.6.2	<i>Labels</i>	39
3.7	<i>Criação de uma linha de ladder.....</i>	43
3.7.1	<i>Inserir nova linha de ladder.....</i>	44
3.7.2	<i>Desenhar e apagar linhas verticais (derivações) ou horizontais.....</i>	45
3.7.3	<i>Inserir instruções especiais</i>	46
3.8	<i>EXERCÍCIOS</i>	48
4.	INSTRUÇÕES BINÁRIAS.....	49
4.1	<i>Instruções de contato de pulso e detecção de borda.....</i>	49
4.1.1	<i>Contato de pulso de subida e descida</i>	49
4.1.2	<i>Instruções MEP e MEF</i>	51
4.1.3	<i>Instrução INV</i>	52
4.2	<i>Instruções SET e RST (Reset).....</i>	52
4.3	<i>Instrução FF (Alternate)</i>	53
4.4	<i>Exercício.....</i>	54
5.	TEMPORIZADORES E CONTADORES	57
5.1	<i>Temporizadores</i>	57
5.1.1	<i>Temporizadores OUT T, OUTH T, OUTHS T</i>	57
5.1.2	<i>Temporizadores retentivos OUT ST, OUTH ST, OUTHS ST</i>	61
5.2	<i>Contadores.....</i>	63
5.3	<i>EXERCÍCIOS</i>	65
6.	INSTRUÇÕES DE DADOS NUMÉRICOS	66
6.1	<i>Introdução</i>	66
6.2	<i>Representação de constantes numéricas.....</i>	68

6.3 Instrução MOV	69
6.4 Instruções de comparação de magnitude.....	70
6.5 Instrução de soma	71
6.6 Instrução de subtração	71
6.7 Instrução de multiplicação	72
6.8 Instrução de divisão.....	73
6.8.1 Exercícios.....	74
6.9 Inline ST para expressões matemáticas.....	75
6.9.1 Exercícios.....	1
6.10 Function Block	2
6.10.1 Exercícios.....	4
7. DIAGNÓSTICOS	5
7.1 Introdução	5
7.2 Função [Device List]	5
7.3 Função [Cross Reference].....	6
7.4 Função [Find/Replace Device/Label].....	7
7.5 Função [System Monitor...] (Exclusivo iQ-R).....	7
7.6 Função [Module Diagnostics (CPU Diagnostics) ...].....	8
7.6.1 Erros e alarmes	9
7.6.2 Firmware	9
7.7 Função [Ethernet Diagnostics...].....	10
7.8 Exercícios.....	12
Anexo A - Importação de projetos do GX Works2.....	14

Apresentação da Aula

Seja bem-vindo ao Treinamento GX Works3.

Objetivos do Curso

Ao final deste curso de treinamento, o estudante deve ser capaz de:

- Entender as ferramentas do software
- Editar um programa em ladder utilizando o software GX Works3
- Configurar os parâmetros dos CLP's de qualquer linha que o software atenda
- Criar um programa utilizando temporizadores, contadores e as principais instruções de programação.
- Simular e identificar possíveis problemas com as ferramentas do software.

Pré-requisitos

Antes de frequentar esta aula, é altamente recomendado que o estudante tenha conhecimentos de elétrica industrial e de programação em linguagem ladder genérica para controladores lógico-programáveis (PLC).

Esta aula é focada no software GX Works3 e sua programação em ladder.

Duração do Curso

Este curso é planejado para a duração de um dia de aula.

Descrição dos Capítulos

CAPÍTULO 1 – Introdução ao GX Works3

Nesse capítulo são apresentadas as ferramentas e funcionalidades do GX Works3.

CAPÍTULO 2 – Projeto no GX Works3

Este capítulo explica os primeiros passos para criar um projeto para o GX Works3.

CAPÍTULO 3 – Operações Offline/Online

Este capítulo explica as operações de programa em ladder, monitoração e labels.

CAPÍTULO 4 – Instruções binárias

Este capítulo explica as instruções básicas de operação de bit e detecção de borda.

CAPÍTULO 5 – Temporizadores e contadores

Este capítulo trata dos temporizadores e contadores no FX5.

CAPÍTULO 6 – Instruções de dados numéricos

Este capítulo detalha as principais instruções de manipulação de dados numéricos.

CAPÍTULO 7 – Diagnósticos

Nesse capítulo são apresentadas as ferramentas de diagnósticos e falhas.

Lista de Manuais Relevantes

JY997D58201 MELSEC iQ-F FX5 User's Manual (Startup)

(Especificação de performance, procedimentos prévios à operação e solução de problemas do módulo de CPU)

JY997D55301 MELSEC iQ-F FX5U User's Manual (Hardware)

(Descreve os detalhes de hardware da CPU FX5U, incluindo entradas e saídas, especificações, fiação, instalação e manutenção)

JY997D55401 MELSEC iQ-F FX5 User's Manual (Application)

(Descreve os conhecimentos básicos requeridos para o desenvolvimento do programa, funções do módulo de CPU, dispositivos/labels e parâmetros)

JY997D55701 MELSEC iQ-F FX5 Programming Manual (Program Design)

(Descreve a especificação do ladder, ST, FBD/LD e outros programas e labels)

SH-081263ENG MELSEC iQ-R CPU Module User's Manual (Startup)

(Especificação de performance, procedimentos prévios à operação e solução de problemas do módulo de CPU)

SH-081264ENG MELSEC iQ-R CPU Module User's Manual (Application)

(Descreve os conhecimentos básicos requeridos para o desenvolvimento do programa, funções do módulo de CPU, dispositivos/labels e parâmetros)

SH-081215ENG GX Works3 Operating Manual

(Configuração do sistema, configuração de parâmetros e operações online)

1. INTRODUÇÃO AO GX WORKS3

1.1 *Introdução*

O software GX Works3 é a ferramenta de programação para os novos PLCs FX5 (linha iQ-F) e R (linha iQ-R). Através do GX Works3 é possível parametrizar a CPU, construir o programa em formação de acordo com a norma IEC 61131-3 e parametrizar os módulos inteligentes (alguns módulos tem possibilidade de parametrização diretamente no módulo).

O GX Works3 faz parte do pacote iQ Works2 e pode ser instalado nas plataformas conforme a tabela abaixo.

Item	MELSOFT GX Works3,
Personal computer	Windows® supported personal computer
CPU	Intel® Core™ 2 Duo Processor 2 GHz or more
Available hard disk capacity	5 GB
Display resolution	1024 x 768 pixels or higher
Communication interface (When a CPU module is directly connected.)	USB port, Ethernet port
Required memory	
64-bit edition	2 GB or more recommended
32-bit edition	1 GB or more recommended
OS (English version)	
Microsoft® Windows® 8.1 Operating System	●
Microsoft® Windows® 8.1 Pro Operating System	●
Microsoft® Windows® 8.1 Enterprise Operating System	●
Microsoft® Windows® 8 Operating System	●
Microsoft® Windows® 8 Pro Operating System	●
Microsoft® Windows® 8 Enterprise Operating System	●
Microsoft® Windows® 7 Starter Operating System	●
Microsoft® Windows® 7 Home Premium Operating System	●
Microsoft® Windows® 7 Professional Operating System	●
Microsoft® Windows® 7 Ultimate Operating System	●
Microsoft® Windows® 7 Enterprise Operating System	●
Microsoft® Windows Vista® Home Basic Operating System	●*1
Microsoft® Windows Vista® Home Premium Operating System	●*1
Microsoft® Windows Vista® Business Operating System	●*1
Microsoft® Windows Vista® Ultimate Operating System	●*1
Microsoft® Windows Vista® Enterprise Operating System	●*1
Microsoft® Windows® XP Professional Operating System SP3	●*1
Microsoft® Windows® XP Home Edition Operating System SP3	●*1

*1: The 64-bit edition is not supported.

1.2 Funcionalidades do GX Works3

- No GX Works3, o sistema do PLC com os módulos é montado de forma gráfica, dentro do próprio software, facilitando a configuração. Os módulos são adicionados arrastando os módulos de uma lista lateral.

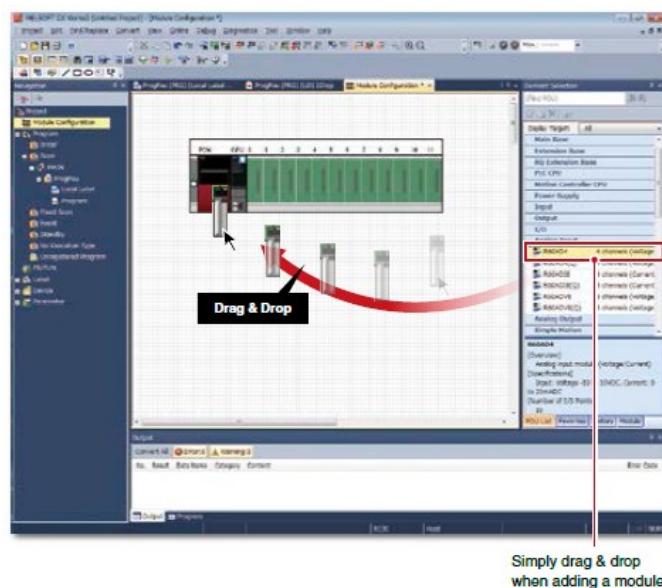


Figura 1: Montagem do sistema de forma gráfica.

- Os parâmetros dos módulos inteligentes são acessados com um duplo clique sobre o módulo que se deseja configurar, no sistema gráfico montado.

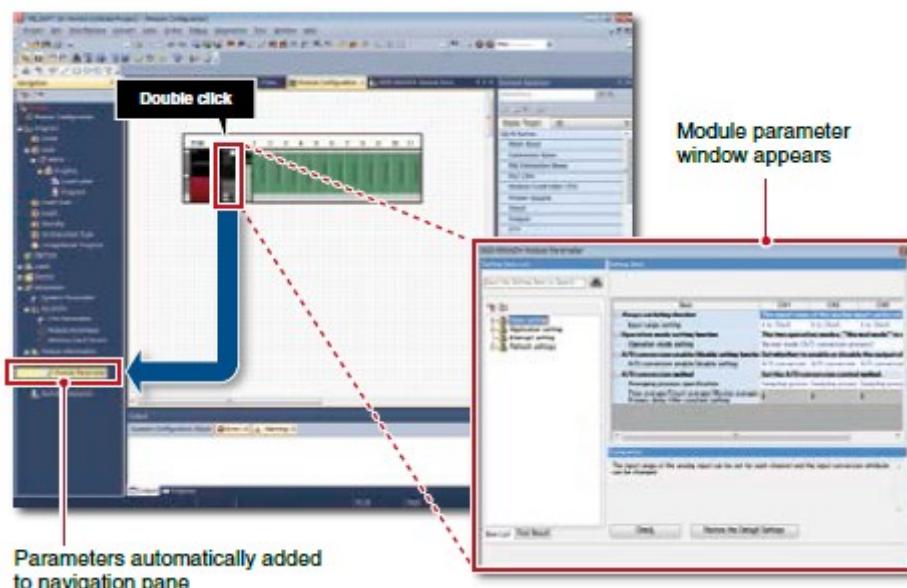


Figura 2: Acesso aos parâmetros dos módulos inteligentes.

- ✧ As principais linguagens da norma IEC são suportadas (SFC estará disponível em breve).

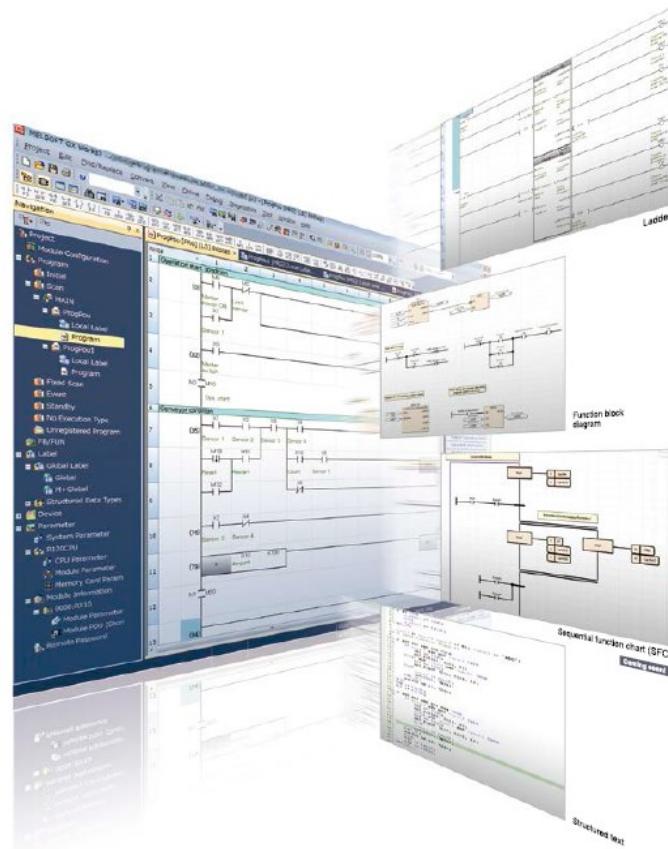


Figura 3: Linguagens de programação de acordo com a norma IEC.

- ✧ Em breve, serão possíveis comparações de programação de dois projetos, com destaque para linhas de programação modificadas, adicionadas ou removidas de um projeto em relação a outro comparado.

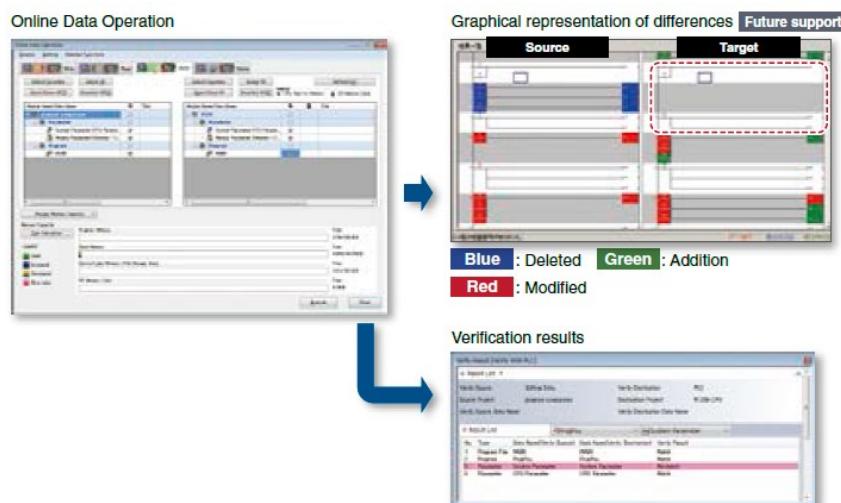


Figura 4: Facilidade na comparação da programas do PLC.

- ✧ É possível colocar comentários de variáveis em múltiplas linguagens e modificar a exibição no programa, de acordo com a conveniência.

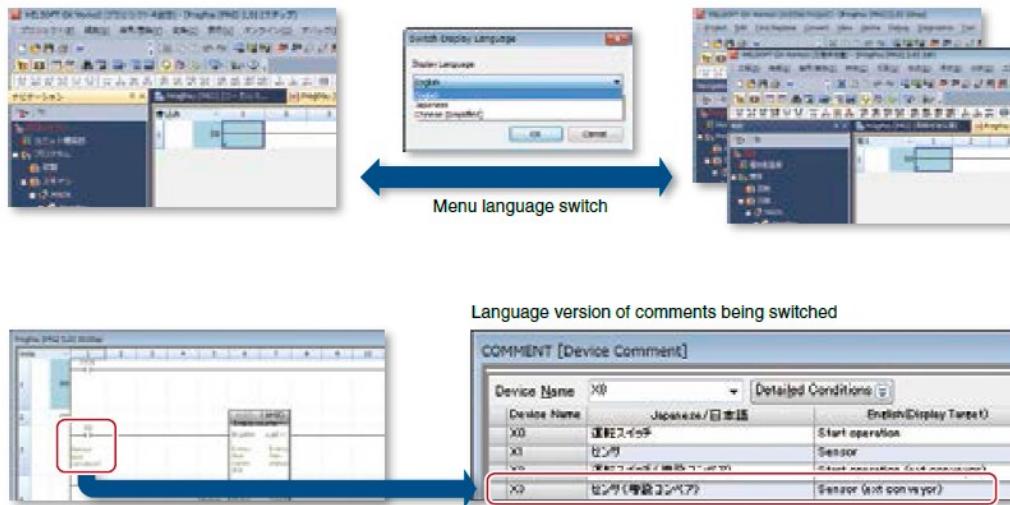


Figura 5: Comentários em multi-linguagens.

- ✧ Para manter a compatibilidade com séries anteriores de PLC, o GX Works3 inclui o GX Works2 e GX Developer em seu pacote de instalação.

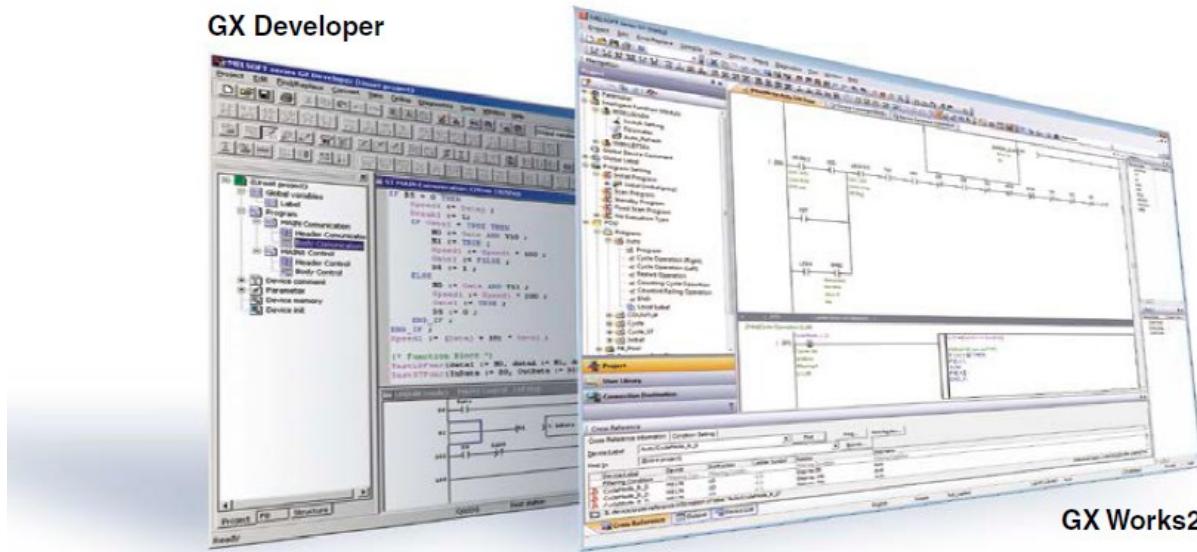


Figura 6: Inclusão dos softwares GX Works2 e GX Developer no GX Works3.

2. PROJETO NO GX WORKS3

Neste capítulo, o participante aprenderá a criar um projeto para GX Works3 com a configuração de hardware.

2.1 Criando um projeto novo

Para criar um novo projeto no software GX Works3 (GXW3), entre no menu Iniciar do Windows e localize [MELSOFT Application] → [GX Works3], em um ícone como o ilustrado na Figura 7 (abaixo).



Figura 7: Ícone do GX Works3.

Na janela que aparece, selecione pelo menu suspenso [Project] → [New...], como ilustrado na Figura 8 (abaixo).

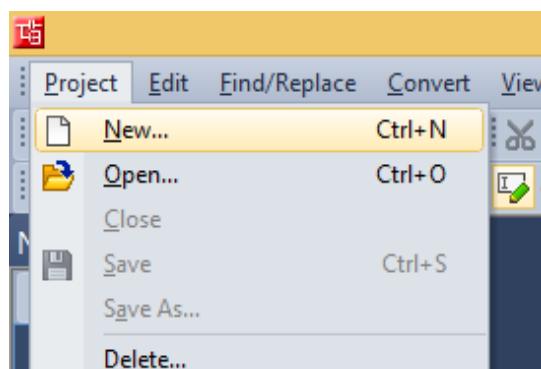


Figura 8: Caminho para criar um novo projeto no GXW3.

Em seguida, selecione a série [Series], tipo de CPU [Type] e linguagem de programação [Program Language]. Neste treinamento será abordada a linguagem de programação ladder. Dessa forma, configure a caixa de diálogo como ilustrado abaixo, de acordo com a CPU que deseja trabalhar, e clique no botão [OK].

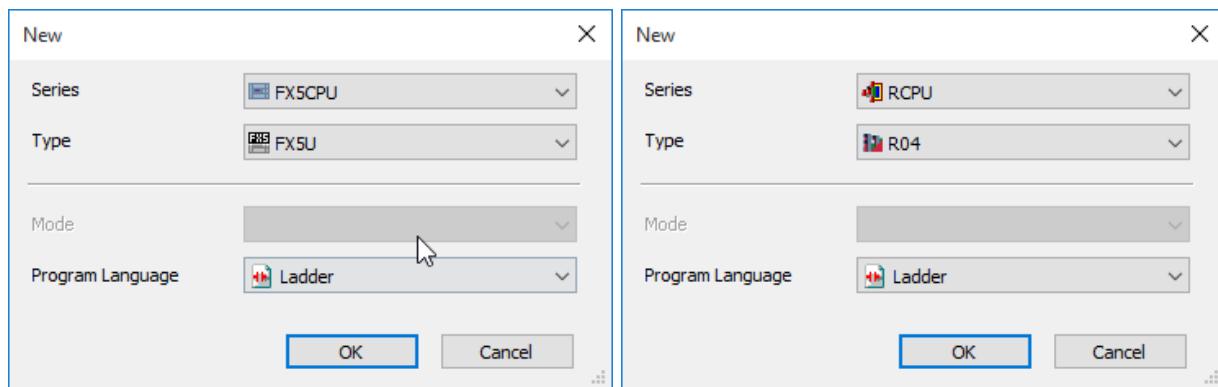
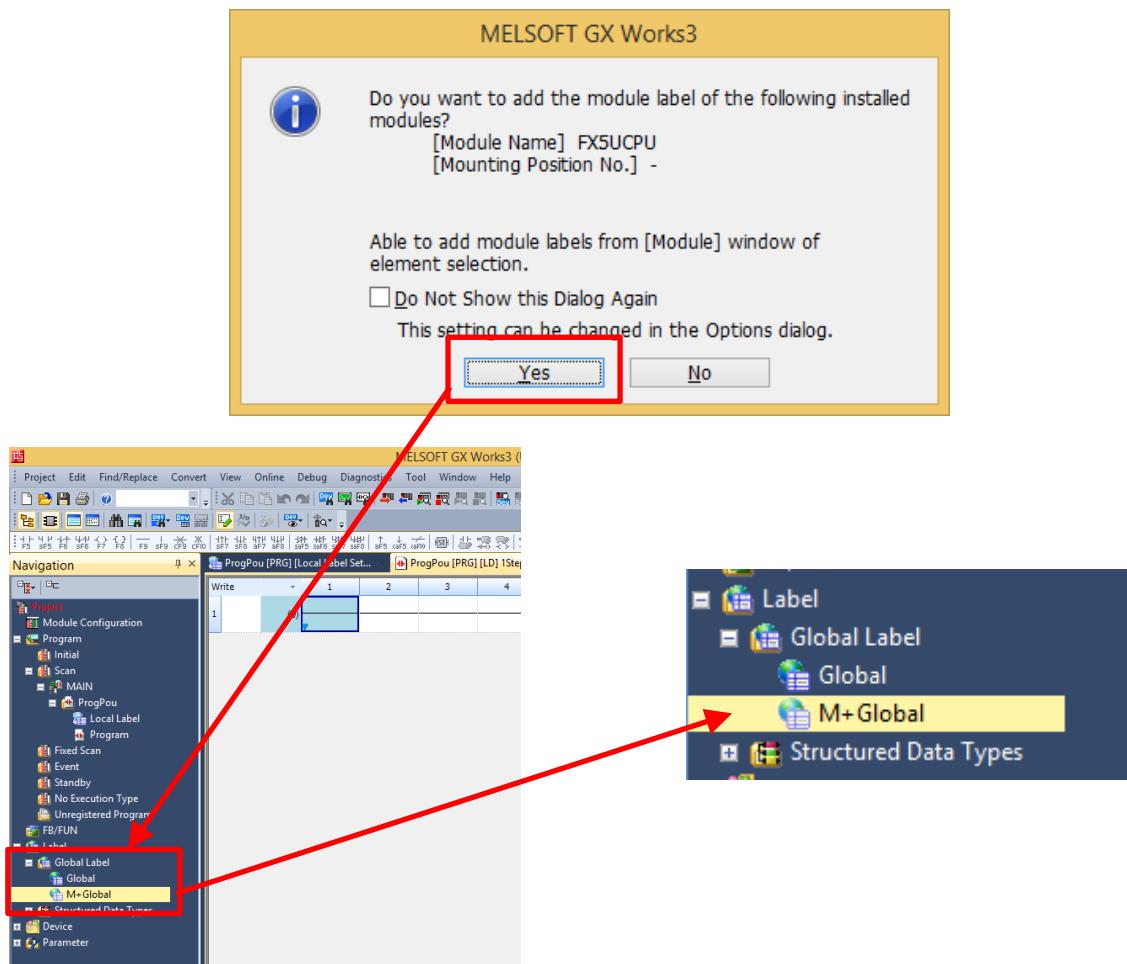


Figura 9: Seleção de projeto, a figura a esquerda para CPU FX5, e a figura a direita para CPU R, com programação em ladder.

Na sequência, uma janela de diálogo como a ilustrada abaixo aparece, perguntando se o usuário deseja incluir variáveis dedicadas dos módulos de CPU aos labels globais. Selecione [Yes] para incluir essas variáveis/labels.



Agora, deverá aparecer uma janela como a ilustrada abaixo.

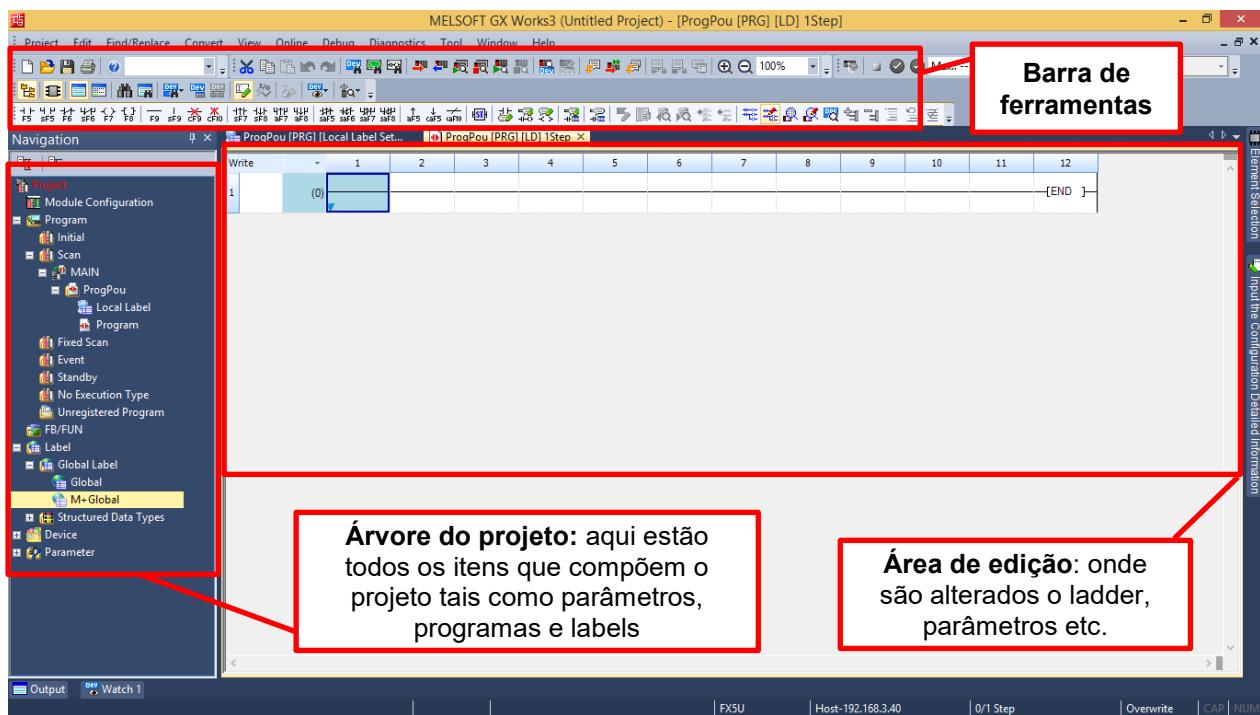


Figura 10: Janela do GX Works3.

2.2 Compilação do Programa

2.2.1 Árvore de projeto

Na árvore de Projeto encontrasse todas as informações necessárias para fazer com que a CPU opere normalmente.

Na Figura 11 você irá notar que existe um item em vermelho. Sempre que realizar a alteração de programa ou parâmetro, o item alterado ficará nessa cor. E, se houver algum item nessa cor, é necessário que você compile o projeto.

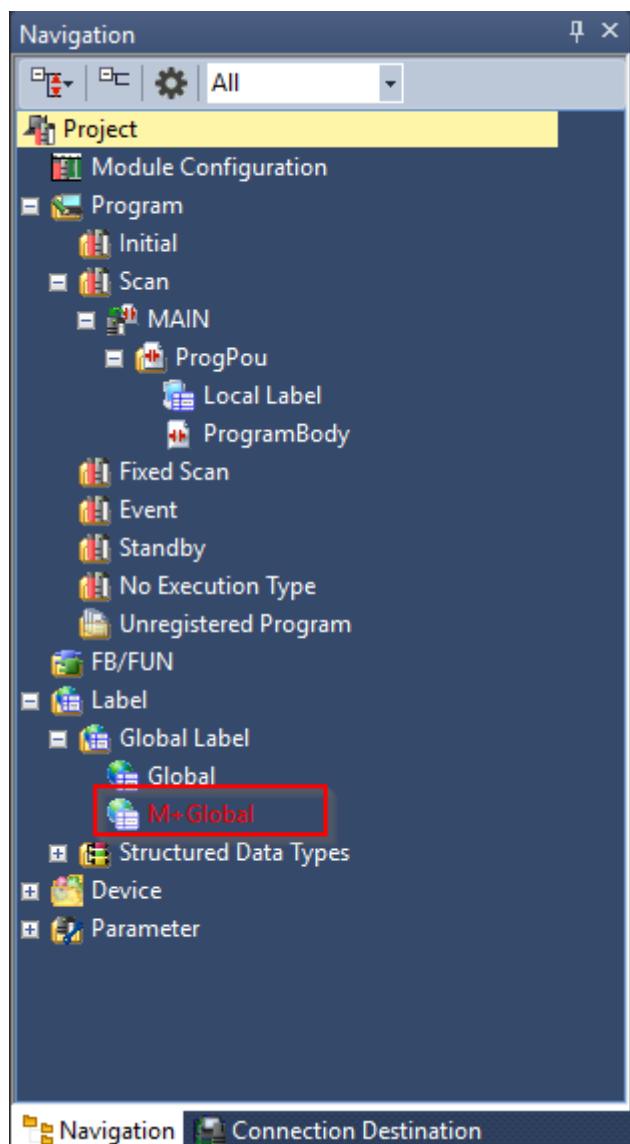


Figura 11: Árvore de Projeto

2.2.2 Compilação do Projeto no GX Works3

O procedimento de compilação do projeto deve ser realizado toda vez que houver alguma alteração no mesmo e for fazer a escrita dessas informações no CLP. A Figura 12 mostra onde localizar esse recurso dentro do software.

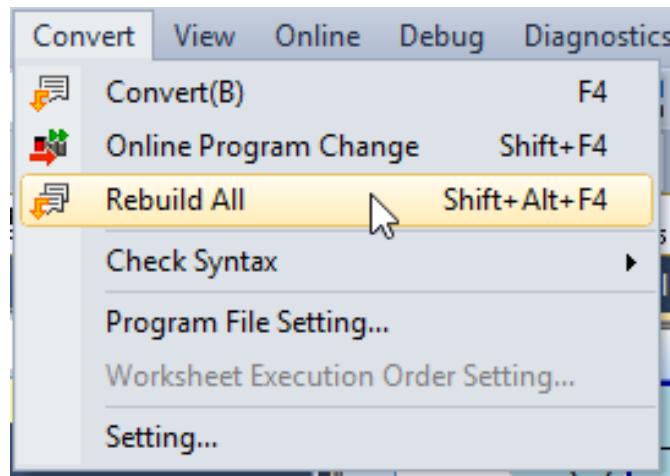


Figura 12: Acesso a função de compilação do programa.

Após clicar em [Rebuild All] a tela da Figura 13 irá aparecer. Clique em [Ok] para iniciar a operação de compilação.

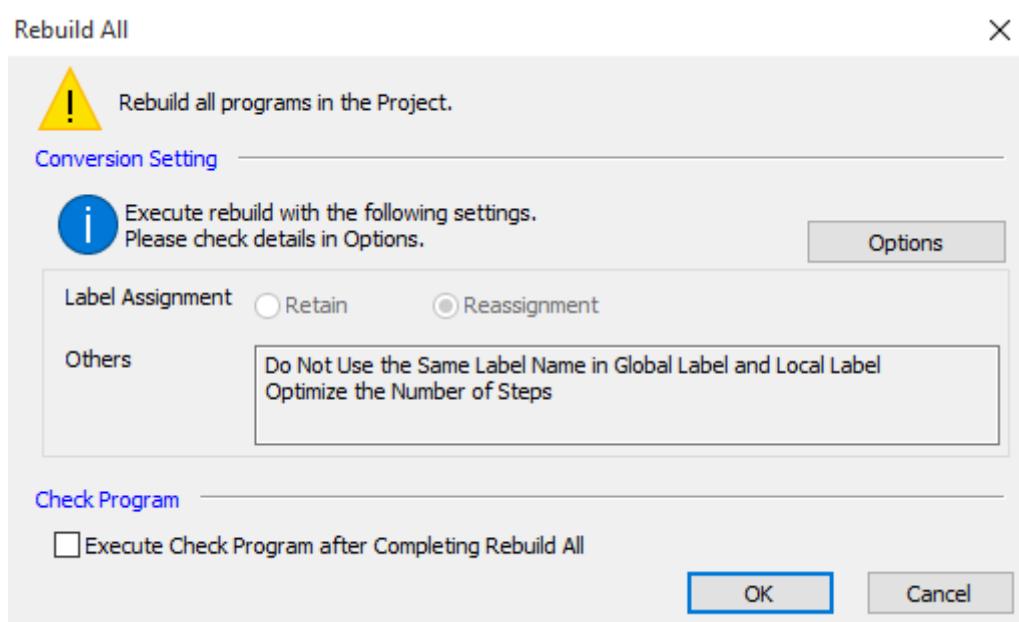


Figura 13: Janela de Compilação (Rebuild all)

Ao finalizar o processo, se não for localizado erros de compilação, a tela fechará automaticamente e na arvore de projetos não terá nenhum ícone em vermelho. Caso haja algum erro de compilação a Janela Output irá aparecer, conforme indicado na Figura 14. Para descobrir onde se encontra o erro dê um duplo click sobre a linha demarcada.

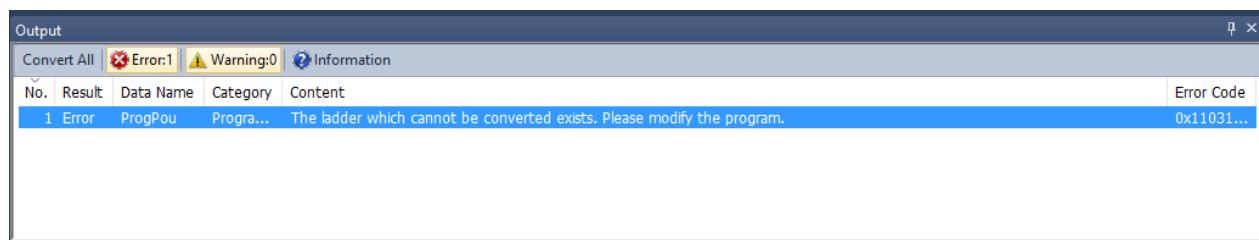


Figura 14: Janela Output com erros de compilação

2.2.3 Compilação do programa no GX Works3

Para compilar apenas a área de ladder do programa aberto, utilize a ferramenta localizada em >>[Convert]>>[Convert] como demonstra a Figura 15, ou o atalho rápido <F4>.

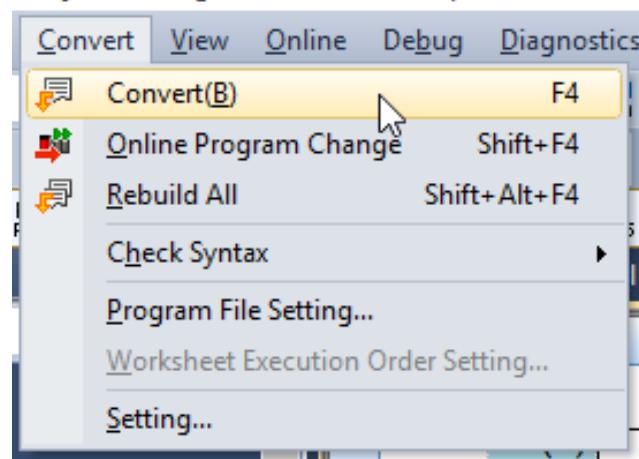


Figura 15: Compilação de programa

2.3 Comunicação do GX Works3 com o PLC

2.3.1 Definição da rota de comunicação

Antes de iniciar a comunicação com o PLC para escrever/ler programas ou monitorar o PLC é necessário definir uma rota de comunicação entre o PC e o PLC. Em outras palavras, é preciso definir qual interface de comunicação será utilizada (serial, Ethernet etc.) e as configurações relacionadas a essa interface.

Para realizar essa definição, entre, pelo menu suspenso em [Online] → [Current Connection Destination] conforme ilustrado na Figura 16:..

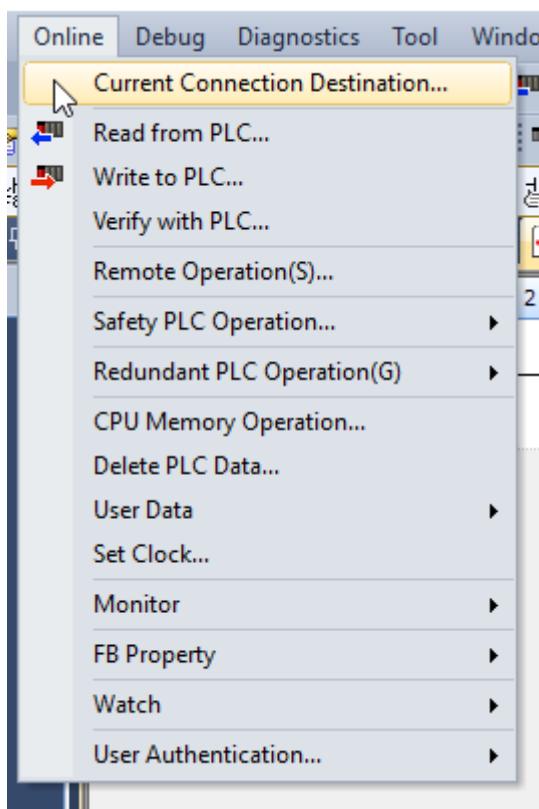


Figura 16: Acesso a janela de conexão com o CLP

Em seguida, a janela ilustrada na Figura 17 aparece para a definição da interface do lado do PC, do PLC e a ligação entre ambas. Como primeiro passo, selecione [Ethernet Board] como a interface do PC e [PLC Module] do lado do PLC.

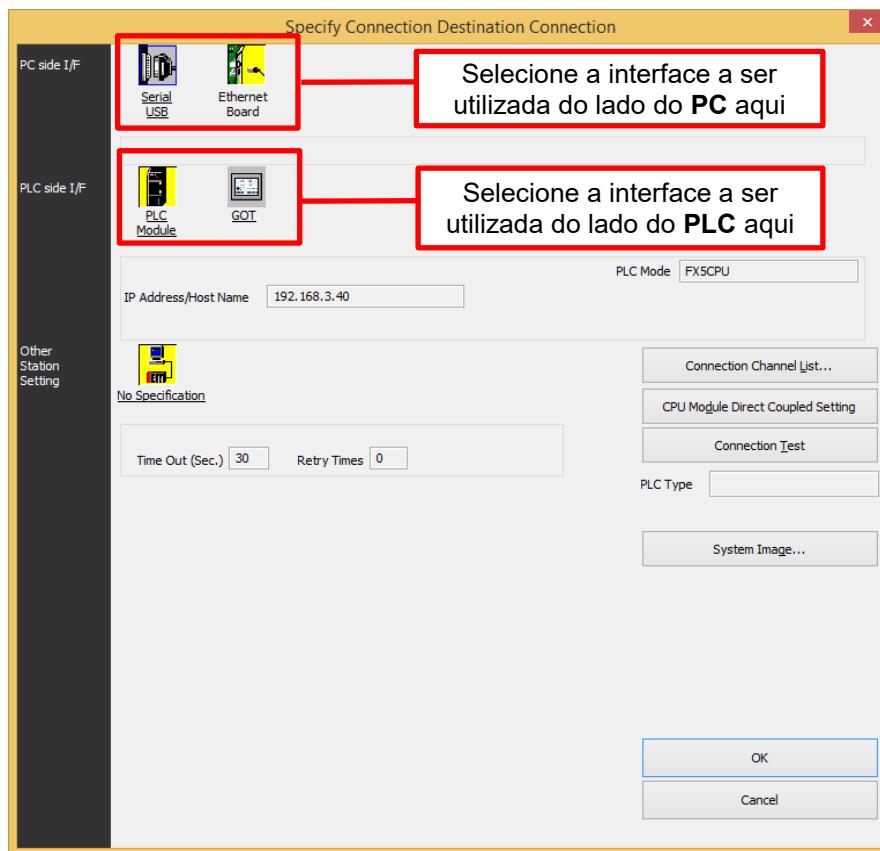


Figura 17: Janela de especificação da rota de comunicação [Specify Connection Destination Connection].

Em seguida, dê um duplo clique sobre o ícone [PLC Module] – na janela ilustrada na Figura 17 – para configurar os detalhes da ligação da interface Ethernet entre o PLC e o PC. Uma janela como a da parte inferior da Figura 18, aparece. Para o treinamento, configure comunicação direta (sem passar por hub) como ilustrado na figura e clique em [OK]. Essa configuração dispensa configuração do endereço IP.

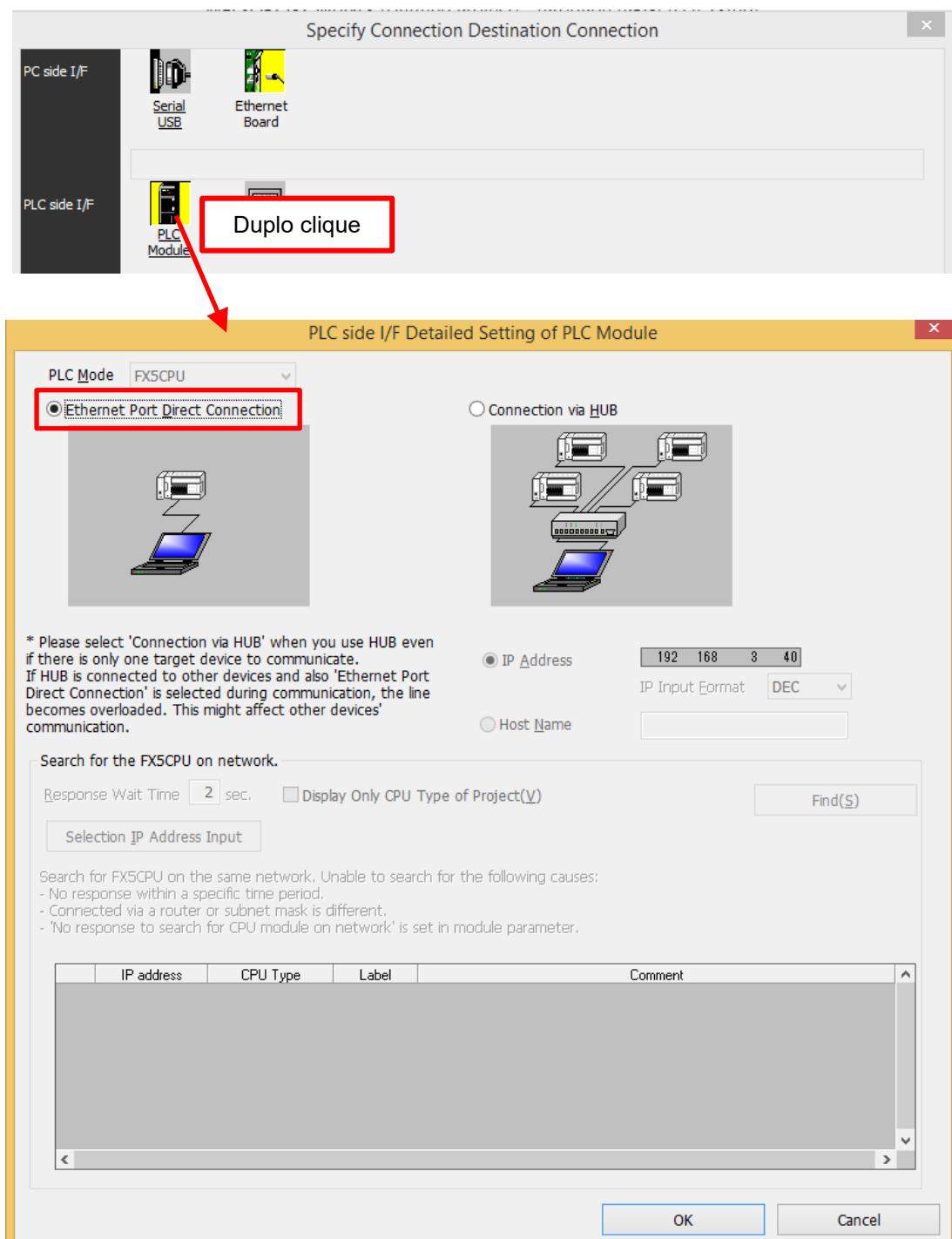


Figura 18: Janela de configuração da comunicação Ethernet com a seleção de comunicação direta.

Após clicar [OK] na janela da Figura 18, deverá aparecer novamente a janela anterior de definição das interfaces do PC e PLC (Figura 17). Nessa janela, pressione o botão [Connection Test] para verificar se a configuração realizada está OK, como ilustrado na Figura 19.

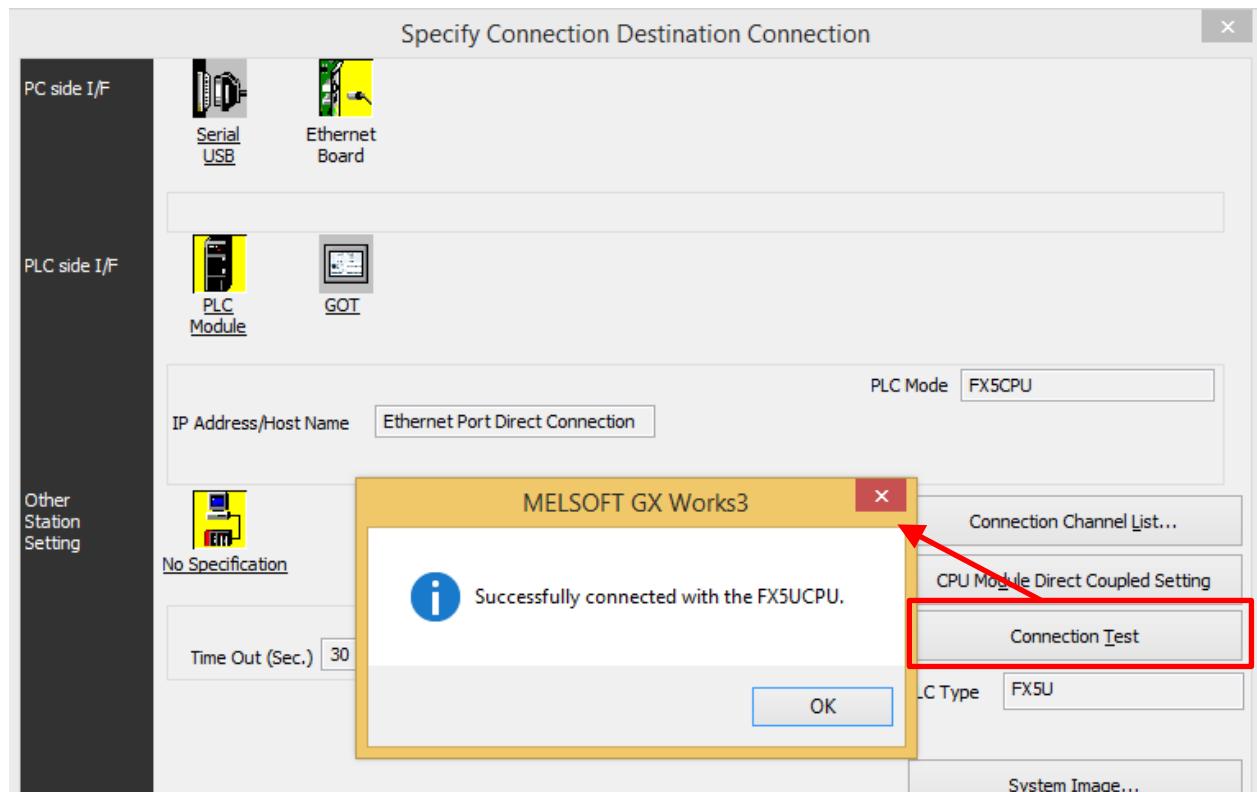


Figura 19: Janela de configuração da comunicação Ethernet com a seleção de comunicação direta.

Após testar a comunicação clique [OK] na caixa de diálogo e na janela [Specify Connection Destination Connection] para confirmar a seleção de rota de comunicação realizada.

O tempo de timeout padrão da comunicação Ethernet é de 30 segundos, dessa forma, se após apertar o botão [Connection Test] a janela congelar sem exibir uma janela de diálogo como a exibida na Figura 19, aguarde. Caso apareça uma janela de diálogo informando uma falha de comunicação, reveja as configurações explicadas nesta seção e verifique se o cabo Ethernet e suas conexões estão corretas.

2.3.2 Escrita de programa para o PLC

Uma vez configurada a rota de comunicação (vide seção 2.3.1), é possível escrever o programa criado para o PLC com o GX Works3, acessando-se no menu suspenso o caminho [Online] → [Write to PLC...]. A janela ilustrada na Figura 20 aparece. Clique no botão [Parameter + Program] para selecionar os dados de programa e parâmetro para serem transferidos para o PLC e, em seguida, clique no botão [Execute] para executar a escrita no PLC.

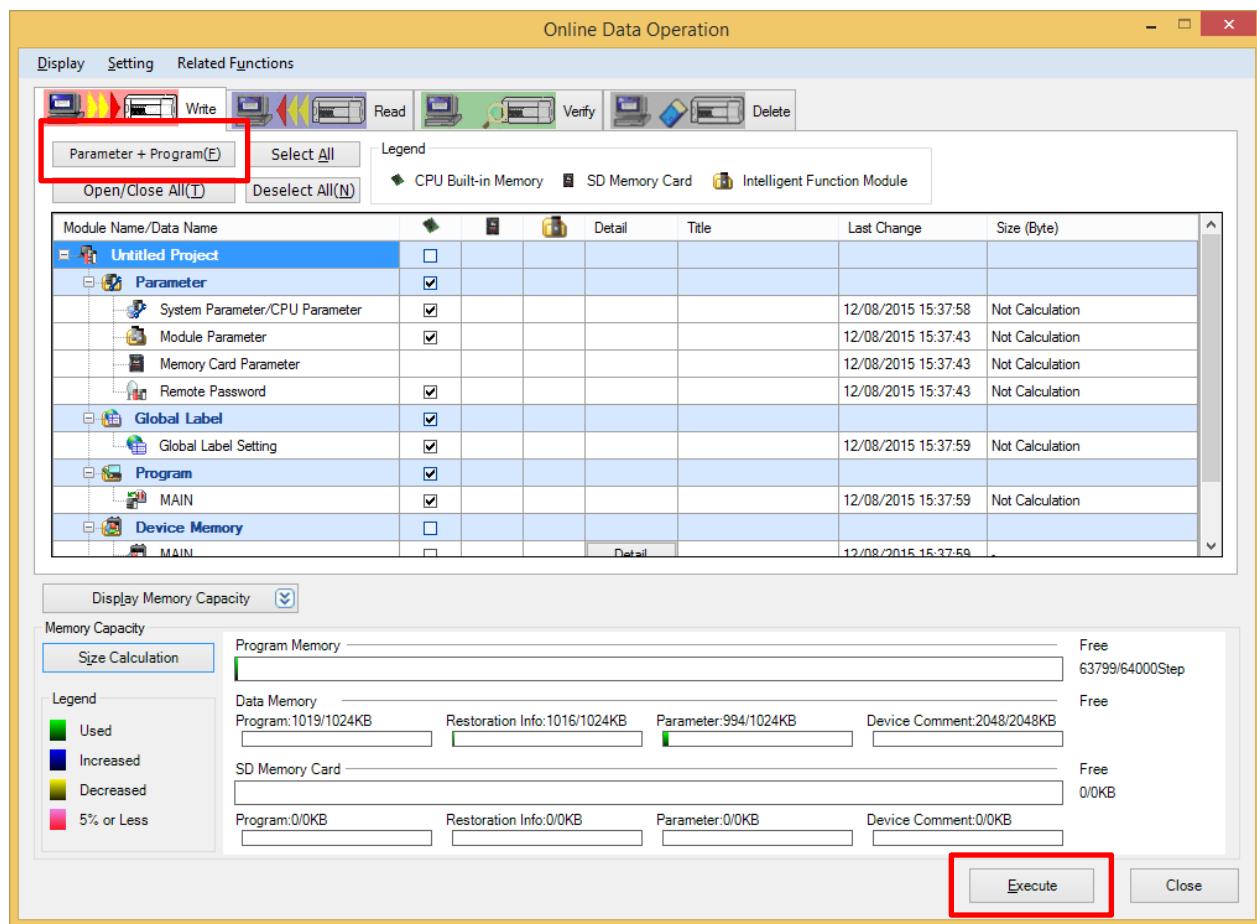


Figura 20: Janela de transferência do programa para o PLC, aba [Write].

2.4 Habilitando o Simulador do PLC

No software GX Works3 é possível utilizar o simulador do PLC. Importante lembrar dois pontos:

1. Ao usar o simulador do CLP, toda e qualquer operação a ser utilizada no modo online será com o CLP Virtual, mesmo que esteja com o CLP Real conectado.
2. Ao utilizar um CLP virtual o tempo de Scan dependerá do sistema operacional que o seu computador possui.

Para habilitar o CLP Virtual realize a compilação do projeto e siga o caminho indicado na Figura 21

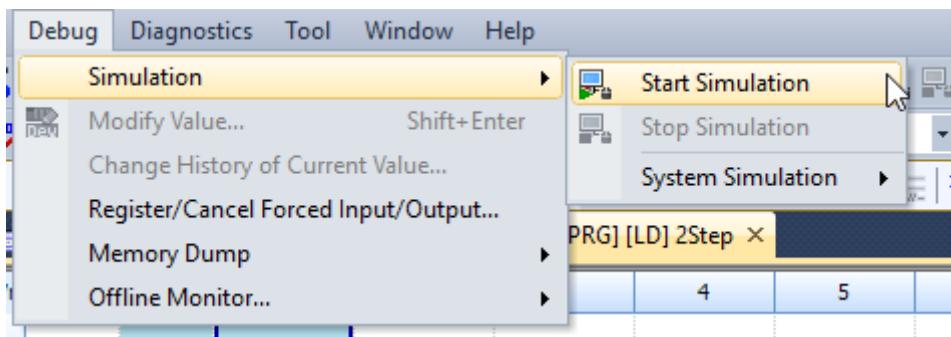


Figura 21: Habilitando simulador

A tela da Figura 20 e da Figura 22. A tela da Figura 22(esquerda) é o GX Simulator3 que será a interface do seu CLP Virtual. Siga o procedimento descrito no item 2.3.2 desta apostila de treinamento. Ao final do processo, a tela de interface do CLP Virtual ficará igual a Figura 22(direita).

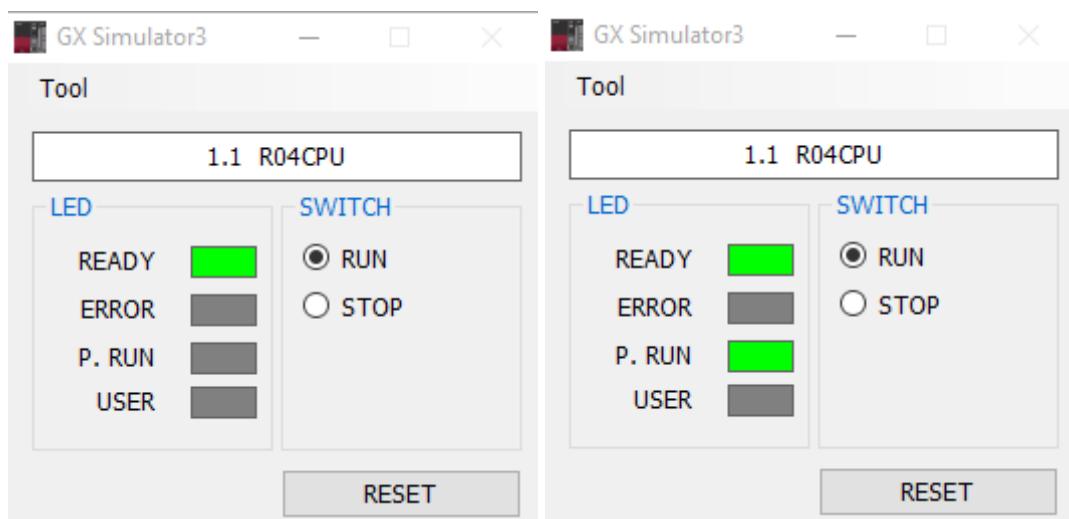


Figura 22: Tela do GX Simulator3, a esquerda antes de escrever Parâmetro+Programa, a direita após escrever o Parâmetro+Programa

3. OPERAÇÕES OFFLINE/ONLINE

3.1 Operação de programa ladder

Para editar o ladder, será utilizado o programa em ladder criado inicialmente com um novo projeto, chamado de programa “MAIN”. Para abrir o programa “MAIN”, localize e abra na árvore do projeto, o seguinte caminho: [Project] → [Program] → [Scan] → [MAIN] → [ProgPou] → [Program]. A Figura 23 (abaixo) ilustra esse caminho.

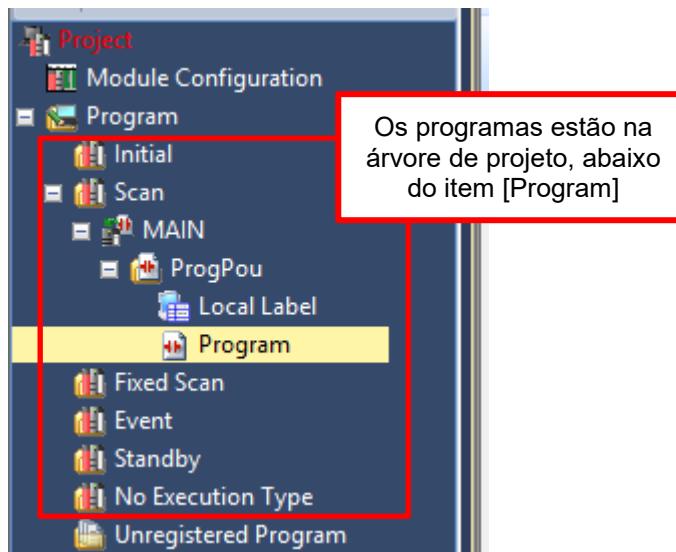


Figura 23: Programas na árvore de projeto.

Como pode ser observado na Figura 23, os programas estão abaixo do item [Program] na árvore de projeto. Cada item abaixo de [Program] – [Initial], [Scan], [Fixed Scan], [Event], [Standby] e [No Execution Type] – agrupam programas que tenham o mesmo modo de execução, como na tabela abaixo.

Tipo de execução	Descrição
Initial	Programas executados apenas uma vez quando o PLC é colocado em RUN
Scan	Programas que rodam constantemente enquanto a CPU estiver em RUN
Fixed Scan	Programas que rodam com tempo de execução (varredura, scan) fixo enquanto a CPU estiver em RUN
Event	Programas que rodam apenas quando houver a ocorrência de um evento programado

Standby	Programas que nunca rodam, ficam aguardando um comando para passarem para outro tipo de execução
No Execution Type	Programas que foram criados e não classificados quanto ao modo de execução

Por hora, no treinamento, será utilizado apenas o programa “MAIN” em modo de execução [Scan]. Para iniciar a edição desse programa, dê duplo clique em [Program], abaixo de [MAIN] como ilustrado na Figura 23 (página anterior). A janela do GX Works3 deverá se apresentar como ilustrado na Figura 24.

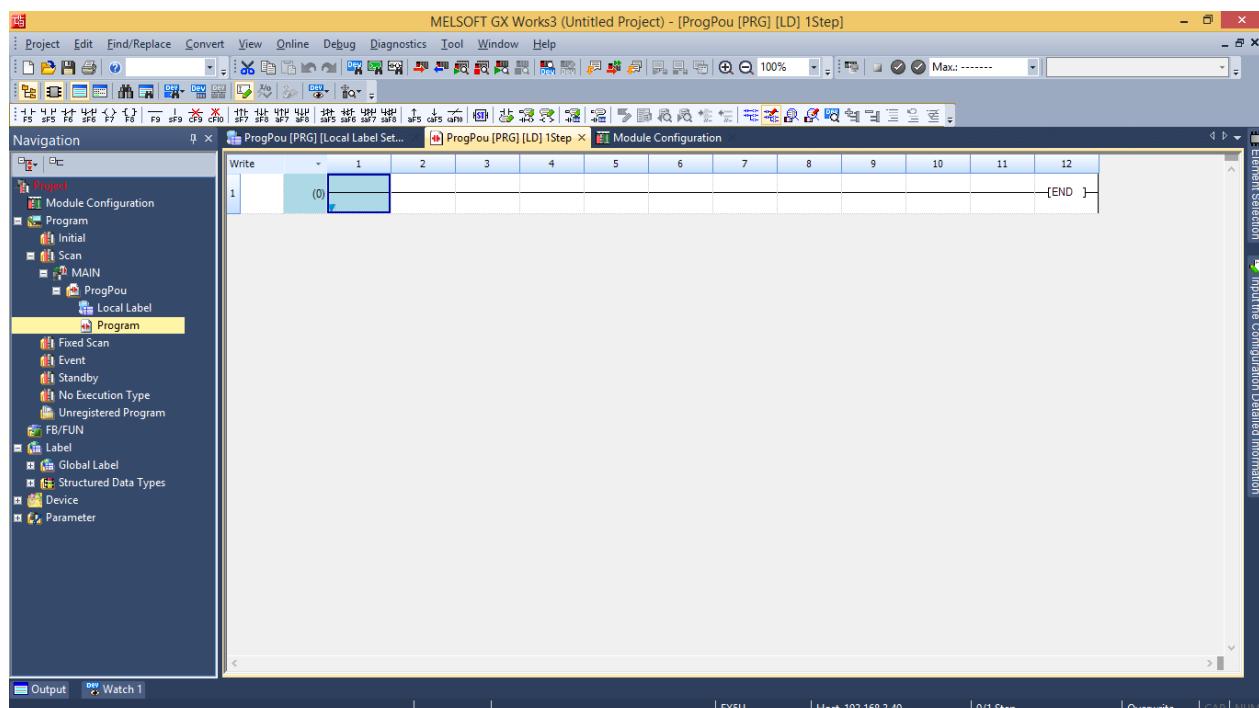


Figura 24: Programa “MAIN” aberto no editor de ladder.

3.2 Modos de leitura e escrita

Para editar o *ladder*, clique na área de edição do *ladder* para utilizar os botões de contato existentes na barra de ferramentas, como será explicado posteriormente.

A saber, há quatro modos de operação da área de edição do ladder, que podem ser acessados através dos botões da barra de ferramentas demonstrados na Figura 25.



Figura 25: Modos de operação

1) Modo de leitura [Read Mode]

Em [Read Mode], cujo modo é de apenas de leitura do programa, o cursor de edição do programa aparece “cheio”, conforme a figura abaixo.



Figura 26: Aspecto do cursor de edição do ladder em [Read Mode].

2) Modo de Escrita [Write Mode]

Em [Write Mode], o modo de permissão de modificar o programa, o cursor de edição do programa aparece “vazado”, conforme a figura abaixo.

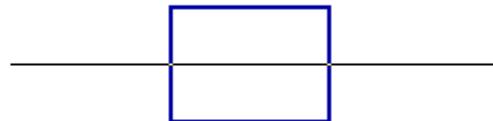


Figura 27: Aspecto do cursor de edição do ladder em [Write Mode].

3) Modo de monitoramento [Monitor Mode]

Em [Read Mode], cujo modo é de apenas de leitura do programa, o cursor de edição do programa aparece “cheio”, conforme a Figura 26, e é possível monitorar o estado dos bits e registradores conforme demonstrado na Figura 29.



Figura 28: Aspecto do cursor de edição do ladder em [Monitor Mode].



Figura 29: Monitoração do programa aberto no editor.

4) Modo de Escrita online[Monitor Write Mode]

Em [Monitor Write Mode], o modo de permissão para modificar o programa em modo online, será solicitado para que você confirme se deseja que ao Compilar o programa (item 2.2.3 Compilação do programa no GX Works3) seja feita a troca online na CPU, de acordo com a Figura 30.

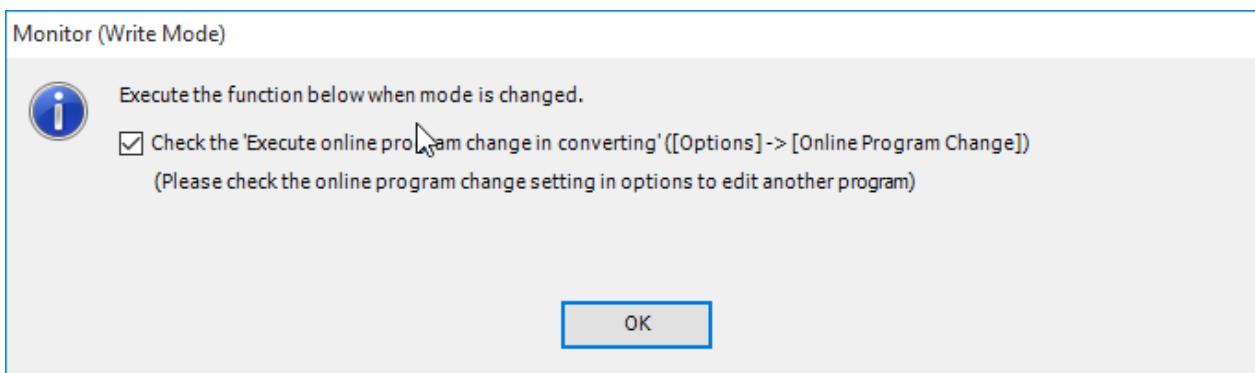


Figura 30: Confirmação ao inicializar o modo de Escrita Online

O cursor de edição aparece “vazado”, conforme a Figura 31 e Figura 29.

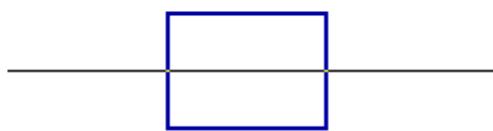


Figura 31: Aspecto do cursor de edição do ladder em [Monitor Write Mode].

3.3 Monitoração de valores de dispositivos/labels

Dentro do GX Works3, existem ferramentas de monitoramento para monitorar os valores armazenados nas variáveis do PLC (dispositivos e labels). Para tal, há duas ferramentas: *Watch* e *Device Batch* (Device/Buffer Memory Batch Monitor).

Device batch destina-se a monitoração de uma área contínua de memória (faixa de dispositivos). Para usar essa função, selecione [Online] → [Monitor] → [Device/Buffer Memory Batch Monitor], conforme Figura 32. Essa função será detalhada na seção 3.3.1 (página 32).

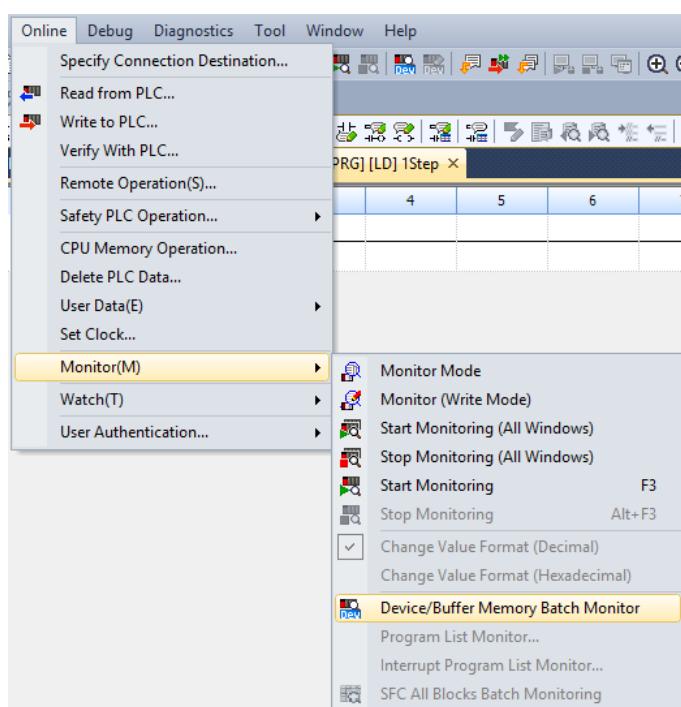


Figura 32: Acesso à monitoração de dispositivos [Device/Buffer Memory Batch Monitor].

“Watch” destina-se a monitoração de dispositivos variados de forma individual. Para usar essa função selecione [Online] → [Watch] → [Register to Watch Window] → [Watch Window 1]. Essa função será detalhada na seção 3.3.2 (página 33).

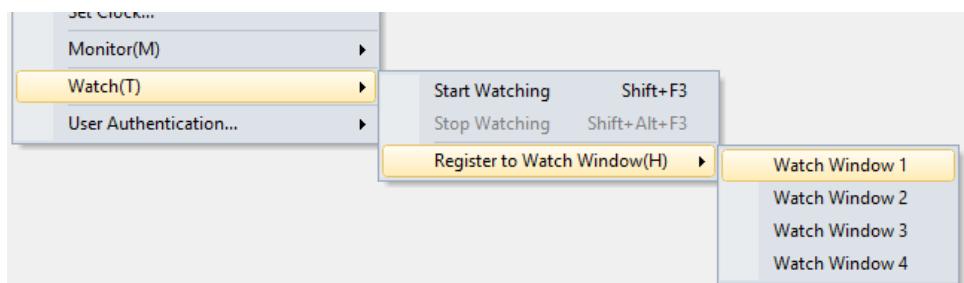


Figura 33: Acesso à monitoração de dispositivos [Device/Buffer Memory Batch Monitor].

3.3.1 Device/Buffer Memory Batch Monitor

Através da ferramenta de monitoração Device Batch (Device/Buffer Memory Batch Monitor), é possível monitorar uma batelada de dispositivos sequencialmente.

A Figura 34 ilustra um exemplo de monitoração de dispositivos a partir do SM400 usando a ferramenta Device Batch.

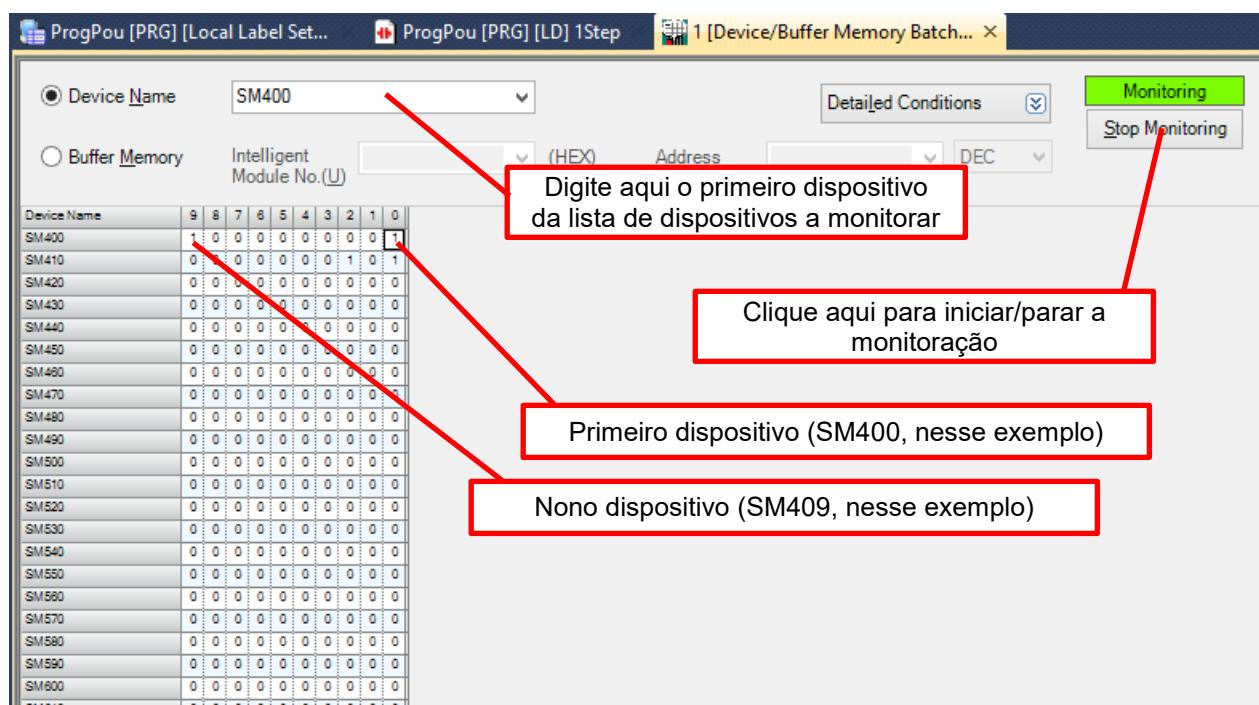


Figura 34: Janela [Device/Buffer Memory Batch Monitor].

O usuário deve digitar o dispositivo de cabeçalho da batelada no local indicado na Figura 34 (acima). Em seguida, o usuário deve pressionar <Enter> ou clicar em [Start Monitoring] (mesmo botão que [Stop Monitoring] na Figura 34 para iniciar a monitoração).

3.3.2 Watch

Através dessa ferramenta de monitoração, é possível monitorar diversos tipos de dispositivos diferentes em tipo e endereço na mesma tela. Para exibir a janela [Watch], é necessário estar na tela de edição do ladder e acessar o caminho [Online] → [Watch] → [Register to Watch Window] → [Watch Window 1] (ou, [Watch Window 2], [Watch Window 3], [Watch Window 4] – são 4 janelas diferentes que podem ser simultaneamente exibidas). O aspecto da exibição da tela de ladder combinado com a janela [Watch] é exibido na.

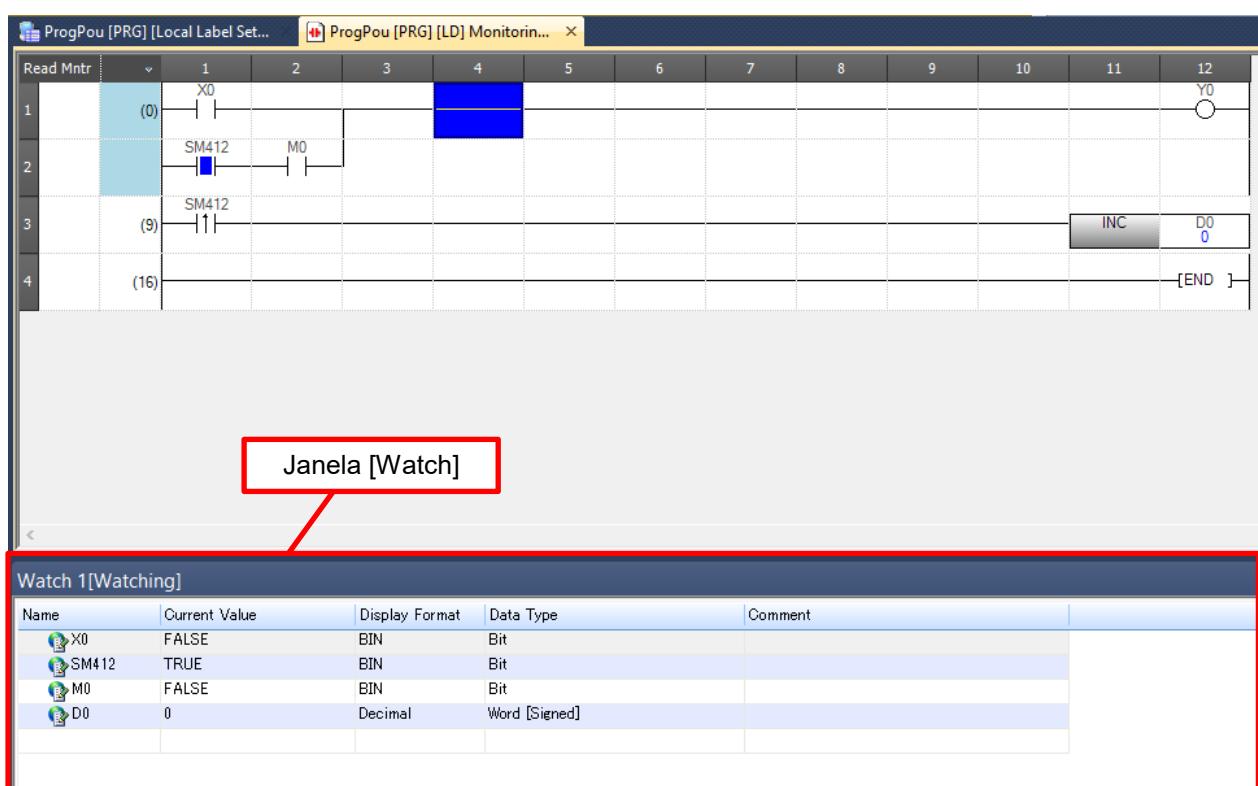


Figura 35: Janela [Watch] combinada com a exibição do editor de ladder.

Para registrar os dispositivos, dê um duplo clique sobre uma das linhas (primeira coluna) da janela [Watch]. O cursor para entrada de dados aparece. Digite o dispositivo necessário de ser monitorado e pressione <Enter> no teclado. Repita o processo nas próximas linhas dessa janela, dependendo da quantidade de dispositivos a serem monitorados. O aspecto da janela deve ser semelhante ao exibido na Figura 37. A janela [Watch] pode ser ocultada ou mantida aberta, para modificar essa condição utilize o botão de tachinha, como ilustrado na Figura 37.

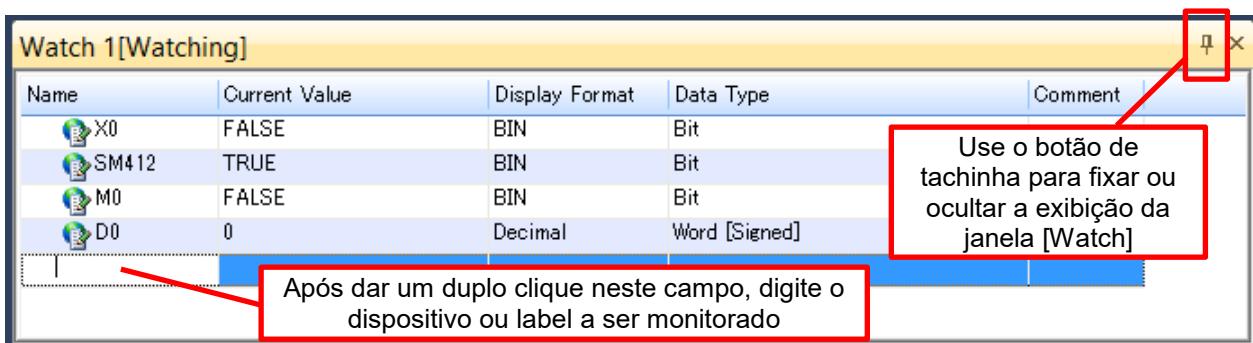


Figura 36: Registro, na janela [Watch], de um dispositivo/label existente no ladder.

Outra forma de registrar um dispositivo na janela [Watch] é, na área de edição do ladder, clicar com o botão direito do mouse sobre um dispositivo/label que se deseja incluir na janela [Watch] e selecionar [Register to Watch Window] → [Watch 1] (ou [Watch 2] a [Watch 4] dependendo da janela que se deseja incluir o dispositivo), como ilustrado no exemplo da Figura 37.

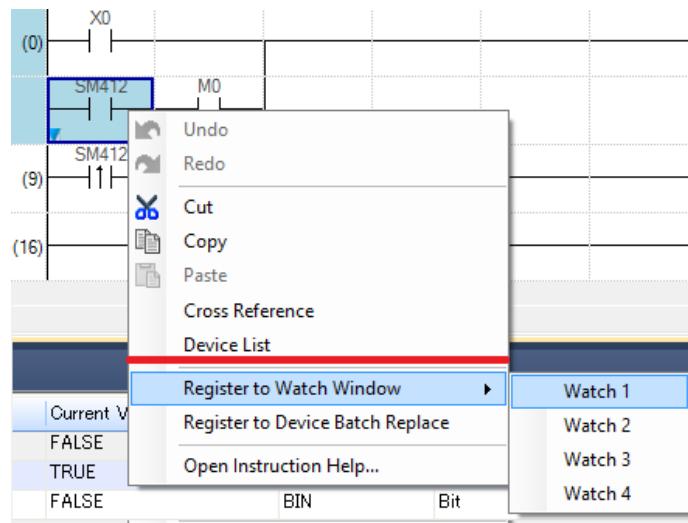


Figura 37: Registro, na janela [Watch], de um dispositivo/label existente no ladder.

Uma vez registrados os dispositivos, clique com o botão direito do mouse sobre as linhas dos dispositivos da janela [Watch] e, no menu que aparece, selecione a opção [Start Watch]. A monitoração dos dispositivos deve iniciar-se.

3.4 Alteração de valores de dispositivos/labels

Uma variável (dispositivo/label) pode ter o seu valor alterado **desde que o programa não imponha um estado (por escrita cíclica na execução do programa) ou valor para essa variável**. Caso não haja uma instrução escrevendo ciclicamente o valor em uma variável, o valor dessa pode ser alterado através da janela [Watch]. Para alterar, clique sobre o valor atual da variável, exibido na janela [Watch], e digite o novo valor a ser escrito na variável, como ilustrado na Figura 38.

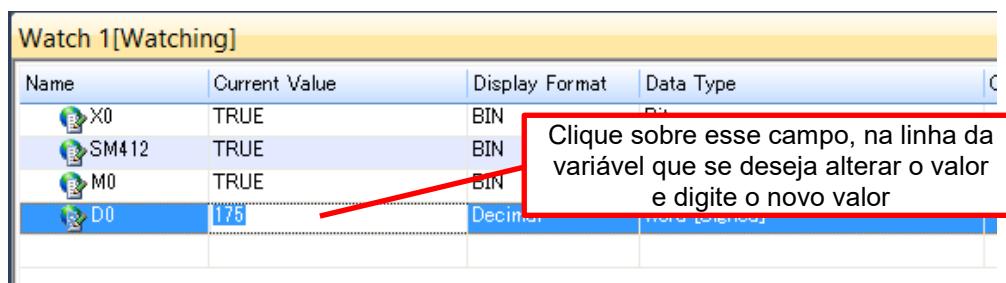


Figura 38: Alteração de valores de variáveis pela janela [Watch].

No caso de variáveis do tipo bit, há a alternativa de colocar o cursor da área de edição do ladder sobre o contato associado ao bit que se deseja alterar o valor e pressionar simultaneamente **<Shift> + <Enter>** para inverter seu estado. Com isso, o estado do bit é invertido sempre que essa ação é repetida.

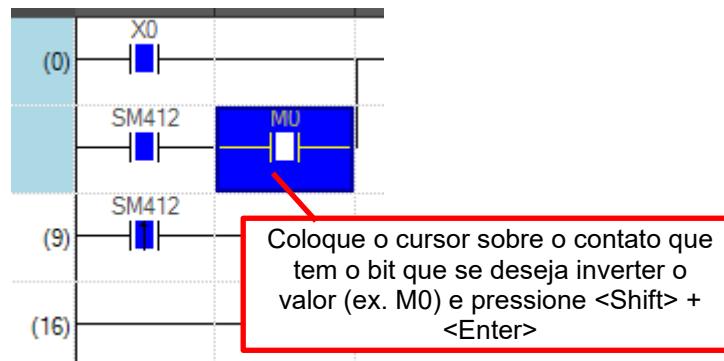


Figura 39: Alteração de valores de bit pelo ladder (inversão de estado).

3.5 Alteração de ladder online

É possível realizar alterações no ladder exibido no GX Works3 e, sem a necessidade de escrever todo o programa no PLC, apenas alterá-lo de acordo com as modificações realizadas no editor. Para realizar tal operação, faça a modificação necessária no editor do programa ladder e, antes de converter/compilar o programa, aperte <Shift> + <F4> ou, pelo menu suspenso, acesse [Convert] → [Online Program Change], como ilustrado na figura 40.

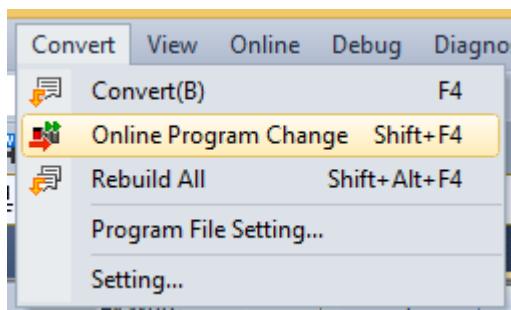


Figura 40: Menu para alteração de programa online.

3.6 Variáveis do PLC: Dispositivos e labels

3.6.1 Dispositivos (Devices)

A tabela a seguir descreve os tipos de dispositivos e suas funções como variáveis do PLC.

Dispositivo (Device)	Detalhes
X	Entradas digitais. Correspondem aos endereços marcados nos bornes de entrada digital do PLC. Endereçado no FX5 de forma octal (X0...X7, X10...X17 etc.) e no R de forma hexadecimal (X0...XF, X10...X1F)
Y	Saídas digitais. Correspondem aos endereços marcados nos bornes de saída digital do PLC. Endereçado no FX5 de forma octal (Y0...Y7, Y10...Y17 etc.) e no R de forma hexadecimal (Y0...YF, Y10...Y1F)
M	Relés internos ou auxiliares. Variáveis tipo bit para construção de lógicas auxiliares ou para receber valores de periféricos como IHMs ou supervisórios (botões etc.). Endereçado em decimal (M0...M9, M10 etc.)
B	Relés de rede. Inicialmente designados para troca de dados tipo bit em rede com outros equipamentos. Quando não utilizados com essa função, podem ter a mesma função do dispositivo M. Endereçado em hexadecimal (B0 a B0F, B10 etc.)
SB	Relés especiais de rede (bits), designados especificamente para receber informações de diagnóstico de rede. Endereçado em hexadecimal (SB0 a SB0F, SB10 etc.)
F	Anunciadores. São variáveis tipo bit que tem função de serem utilizados para armazenamento de estado de falhas operacionais da aplicação programada pelo usuário. Quando um dispositivo F é acionado, entra em um histórico que pode ser acessado pelo usuário como histórico de falhas operacionais da máquina/aplicação. Endereçado em decimal (F0...F9, F10 etc.)
S	Step relays. São bits utilizados na construção de programas em Step Ladder ou SFC. Endereçado em decimal (S0...S9, S10 etc.)
T	Timers ou temporizadores. São áreas de memória específicas para uso com instruções de temporizadores. No FX5: OUT T, OUTH T, OUTHS T, e no R OUT T, OUTH T. Endereçado em decimal (T0...T9, T10 etc.)
LT	Timers ou temporizadores exclusivos da série R. São áreas de memória específicas para uso com instruções de temporizadores.
ST	Timers ou temporizadores retentivos. São áreas de memória específicas para uso com instruções de temporizadores retentivos. No FX5: OUT T, OUTH T, OUTHS T, e no R OUT T, OUTH T. Endereçado em decimal (ST0...ST9, ST10 etc.)
C	Contadores de 16 bits. São áreas de memória específicas para uso com instruções de contadores (OUT C). Endereçado em decimal (C0...C9, C10 etc.)
LC	Long Counters ou contadores de 32 bits. São áreas de memória específicas para uso com instruções de contadores (OUT LC). Endereçado em decimal (LC0...LC9, LC10 etc.)
D	Registradores de 16 bits. São áreas de memória de 16 bits para armazenamento de valores numéricos ou strings (cadeia de caracteres). Podem ser combinadas para armazenamento de valores numéricos em 32

	bits ou para armazenamento de strings. Endereçado em decimal (D0...D9, D10 etc.)
L	Latch relays ou Relés auxiliares retentivos por memória não volátil. Variáveis tipo bit para construção de lógicas auxiliares ou para receber valores de periféricos como IHMs ou supervisórios (botões etc.) com retentividade de informação em caso de perda de alimentação ou reset do PLC. Endereçado em decimal (L0...L9, L10 etc.)
R	File Register (16 bits). São áreas de memória de 16 bits para armazenamento de valores numéricos ou strings (cadeia de caracteres) adicional aos registradores D. Podem ser combinadas para armazenamento de valores numéricos em 32 bits ou para armazenamento de strings. Quando retentivos por bateria são geralmente usados para armazenamento de receitas. Endereçado em decimal (R0...R9, R10 etc.)
W	Registradores de rede (16 bits). São áreas de memória de 16 bits para armazenamento de valores numéricos ou strings (cadeia de caracteres) para troca de dados entre equipamentos via rede. Podem ser combinadas para armazenamento de valores numéricos em 32 bits ou para armazenamento de strings. Endereçado em hexadecimal (W0...W0F, W10 etc.)
SW	Registradores especiais de rede (16 bits, words), designados especificamente para receber informações de diagnóstico de rede. Endereçado em hexadecimal (SW0 a SW0F, SW10 etc.)

Existe ainda, alguns dispositivos de funções específicas para diagnóstico e auxílio à programação. Esses dispositivos não tem as configurações de quantidade e memória alteráveis. A tabela abaixo detalha os dois tipos de dispositivo desse gênero.

Dispositivo (Device)	Detalhes
SM	Relés ou bits especiais. Têm funções fixas e predeterminadas que podem ser conhecidas utilizando-se o menu de ajuda (help) do GX Works3.
SD	Registradores especiais (16 bits, words). Têm funções fixas e predeterminadas que podem ser conhecidas utilizando-se o menu de ajuda (help) do GX Works3.

Alguns dos principais bits especiais:

Dispositivo (Device)	Detalhes
SM400	Bit sempre ligado (ON)
SM401	Bit sempre desligado (OFF)
SM402	Bit ligado apenas na primeira varredura após a CPU entrar em modo RUN
SM412	Bit pulsante (clock) de período 1 segundo (0,5 s OFF / 0,5 s ON)

3.6.2 Labels

Os labels são variáveis do PLC que não são predeterminadas em áreas de memórias fixas, cabe ao usuário programador determinar um nome para o label e atribuir a ele uma função ou quantidade de memória. Esse processo é feito na declaração de variáveis que será visto mais adiante.

Os labels apresentam-se em dois tipos: locais e globais. Os labels locais são variáveis internas a um determinado programa. Essas variáveis não são acessíveis a outros programas ou equipamentos periféricos e são utilizadas para organizar dados que interessam apenas ao processamento interno de um programa. Os labels globais são variáveis acessíveis a todos os programas e periféricos de forma compartilhada. Essas variáveis são utilizadas quando se deseja compartilhar informações entre programas ou receber/transmitir dados com periféricos como IHMs e supervisórios. A Figura 41 ilustra a localização das listas de declaração dos labels local (no exemplo, para o programa “MAIN”) e global.

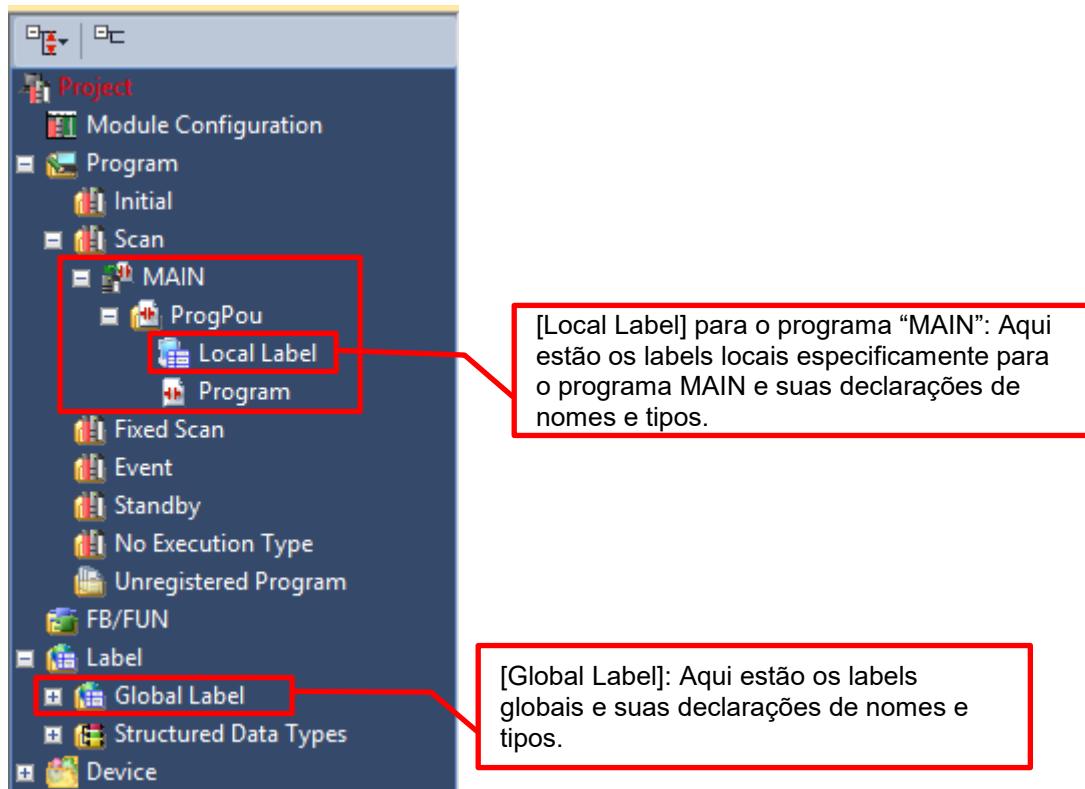


Figura 41: Localização dos labels globais e locais na árvore do projeto.

A Figura 42 (abaixo) ilustra a declaração de labels locais.

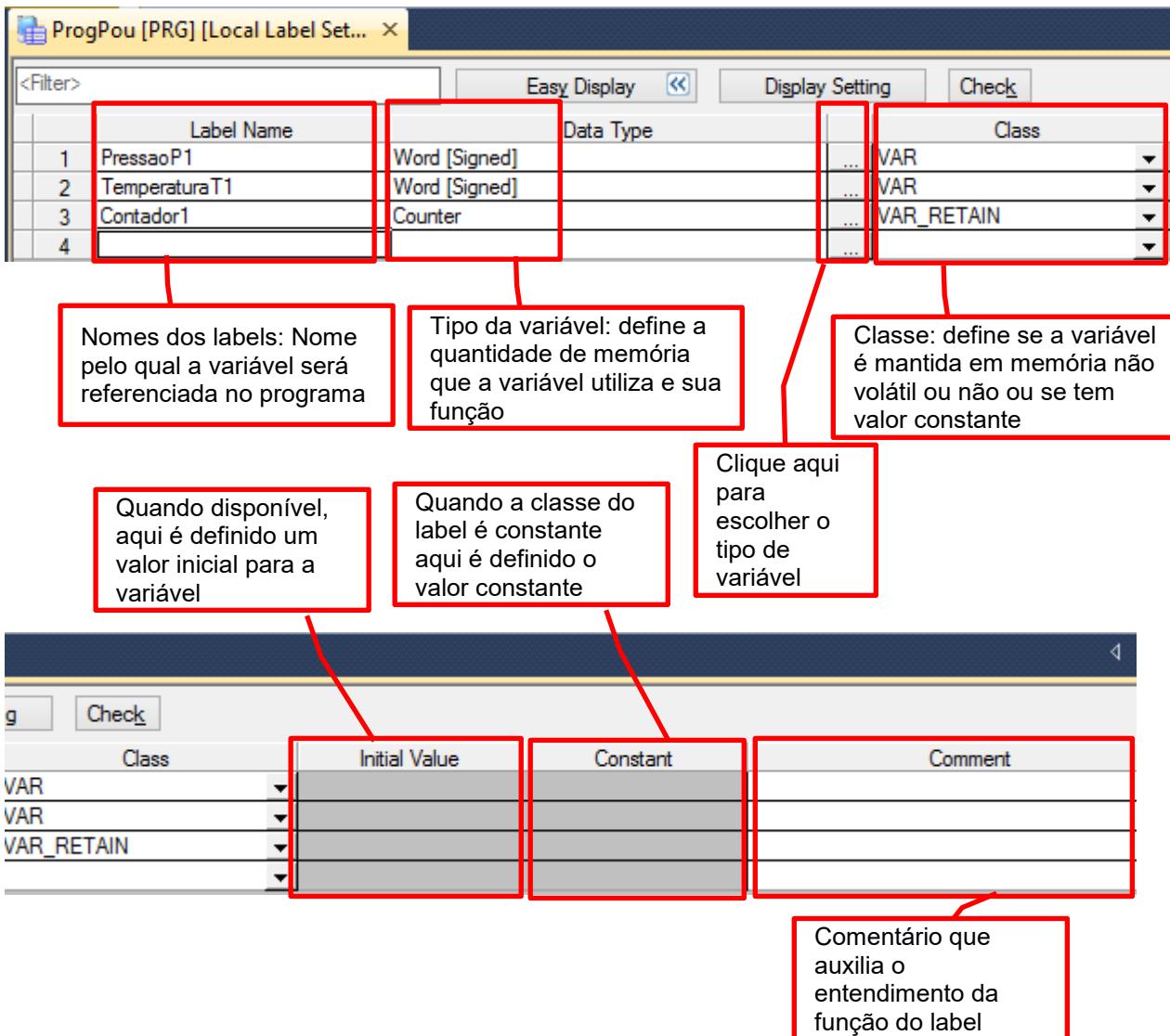


Figura 42: Declaração de labels locais.

No caso dos labels globais, vale tudo o que foi descrito para os labels locais, porém há algumas opções adicionais, como ilustrado na Figura 43 (abaixo).

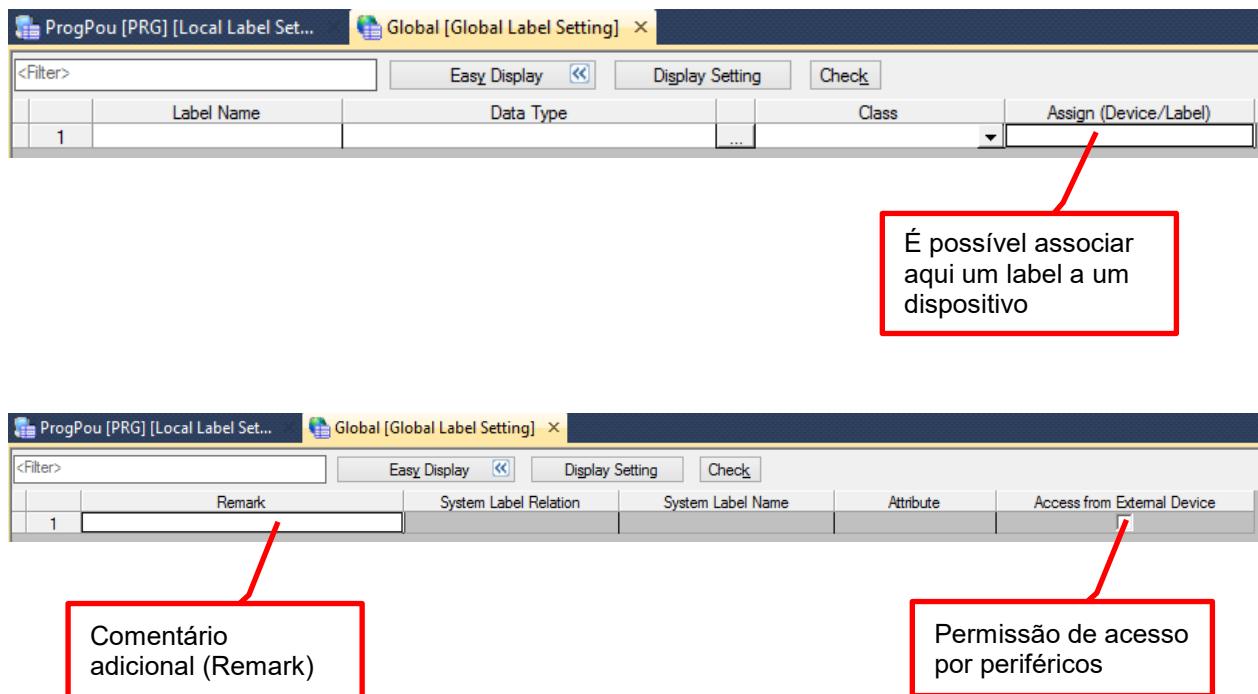


Figura 43: Declaração de labels globais.

Na tabela abaixo, estão as descrições para os tipos de variáveis (Data Type).

Tipo de variável (Data Type)	Detalhes
Bit	Informação de bit. A variável pode receber uma informação binária TRUE (1, ON) ou FALSE (0, OFF)
Word [Signed]	Informação de 16 bits com sinal. A variável pode receber valores entre -32768 e +32767
Word [Unsigned]/Bit string (16-bit)	Informação de 16 bits sem sinal. A variável pode receber valores entre 0 e 65535
Double Word [Signed]	Informação de 32 bits com sinal. A variável pode receber valores entre -2147483648 e +2147483647.
Double Word [Unsigned]/Bit string (32-bit)	Informação de 32 bits sem sinal. A variável pode receber valores entre 0 e +4294967296.

FLOAT [Single Precision]	Informação em ponto flutuante de 32 bits (número decimal com vírgula, com quantidade flutuante de casas decimais). A variável pode receber valores entre -2 ¹²⁸ to -2 ⁻¹²⁶ , 0, 2 ⁻¹²⁶ to 2 ¹²⁸
TIME	Variável de 32 bits para armazenamento de valor de tempo no formato IEC (T#). Pode armazenar valores entre T#-24d20h31m23s648ms a T#24d20h31m23s647ms
String	Variável para armazenamento de cadeia de caracteres. Tem tamanho variável e pode armazenar entre 0 a 255 caracteres (o tamanho é definido entre parênteses na frente da declaração "String", ex. String(20))
Pointer	Ponteiro para salto do programa.
Timer	Variável para uso exclusivo de função temporizador com instruções de bobina OUT T, OUTH T ou OUTHS T.
Retentive Timer	Variável para uso exclusivo de função temporizador retentivo com instruções de bobina OUT ST, OUTH ST ou OUTHS ST.
Counter	Variável para uso exclusivo de função contador com instruções de bobina OUT C.
Long Counter	Variável para uso exclusivo de função contador de 32 bits com instruções de bobina OUT C.

Na tabela a seguir, são descritas as classes utilizáveis com declaração de labels globais e locais.

Classe (Class)	Detalhes
VAR(_GLOBAL)_CONSTANT	Classe de label que tem valor fixo, constante.
VAR(_GLOBAL)	Variável não retentiva por memória não volátil
VAR(_GLOBAL)_RETAIN	Variável retentiva por memória não volátil

3.7 Criação de uma linha de ladder

Para inserir um contato ou outro elemento, podemos utilizar a barra de ferramenta ou podemos utilizar atalhos no teclado do PC.



Exemplos de atalho de teclado

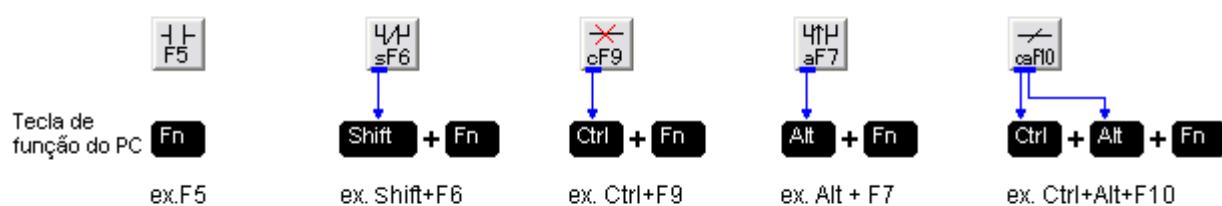


Figura 44: Barra de ferramenta para edição do ladder e atalhos do teclado.

Exemplo: Para inserir um contato NA de entrada “X0”, pressione a tecla <F5>. Aparecerá a seguinte janela:

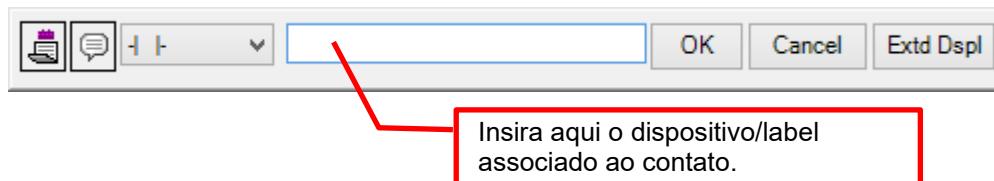


Figura 45: Janela para inserção de uma instrução ladder.

Digite o dispositivo “X0” na região indicada e pressione <Enter> ou clique no botão [OK].

Obs: Não é necessário digitar em letras maiúsculas ou colocar zeros na frente da numeração, o programa fará as mudanças necessárias automaticamente.

Em qualquer programa, a instrução “END” aparece incondicionalmente na última linha. Caso o usuário insira um contato ou qualquer outro elemento, o software automaticamente deslocará a instrução “END” para baixo.

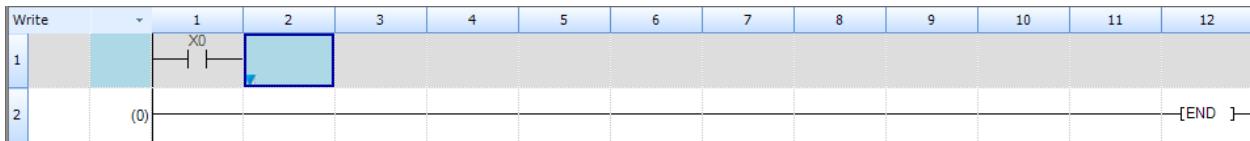


Figura 46: Inserção de um contato NA (NO) em uma lógica ladder.

Em seguida, insira uma bobina “Y1”. Pressione <F7>, digite “Y1” e pressione <Enter>.

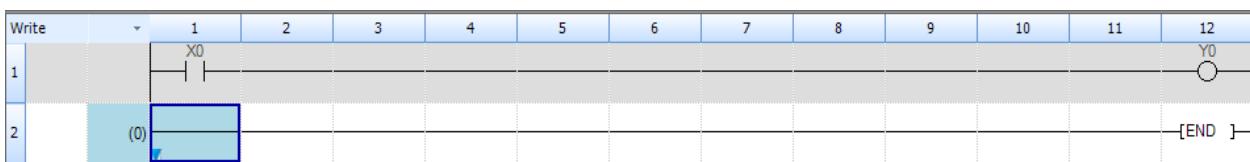


Figura 47: Inserção de uma bobina em uma lógica ladder.

Observe que o cursor se movimenta automaticamente para linha de baixo.

A área acinzentada que aparece durante a programação significa que essa parte do programa foi alterada e, portanto será necessário compilar (converter) o programa. Para compilar o programa pressione <F4> ou na barra de ferramenta clique no botão .

Ao converter o programa a área de edição deve apresentar o seguinte aspecto:

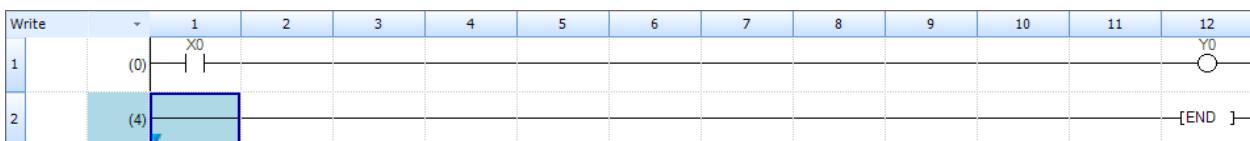


Figura 48: Linha de ladder convertida/compilada.

Caso o ladder apresente algum erro aparecerá uma janela [Output] na parte inferior com detalhes sobre a falha de conversão ocorrida.

3.7.1 Inserir nova linha de ladder

Caso queira inserir linha de *ladder* em qualquer parte do programa, basta posicionar o cursor na linha abaixo da posição onde deseja inserir uma nova linha e pressione <Shift> + <Insert>.



Figura 49: Inserção de nova linha (row) de ladder.

Outra opção é utilizar o menu superior [Edit] → [Insert Row].

3.7.2 Desenhar e apagar linhas verticais (derivações) ou horizontais

Em caso de necessidade de inserir um contato ou algum outro elemento em paralelo com uma linha de ladder existente, será necessário desenhar linhas verticais. Em outras situações será necessário desenhar linhas horizontais.

Para tanto, posicione o cursor do ladder onde se deseja acrescentar uma linha vertical e mantenha a tecla <Ctrl> pressionada, juntamente com uma tecla de direção <→>, <↓>, <↑> ou <←> dependendo do sentido que se necessitar desenhar linhas. A Figura 49 ilustra o procedimento.

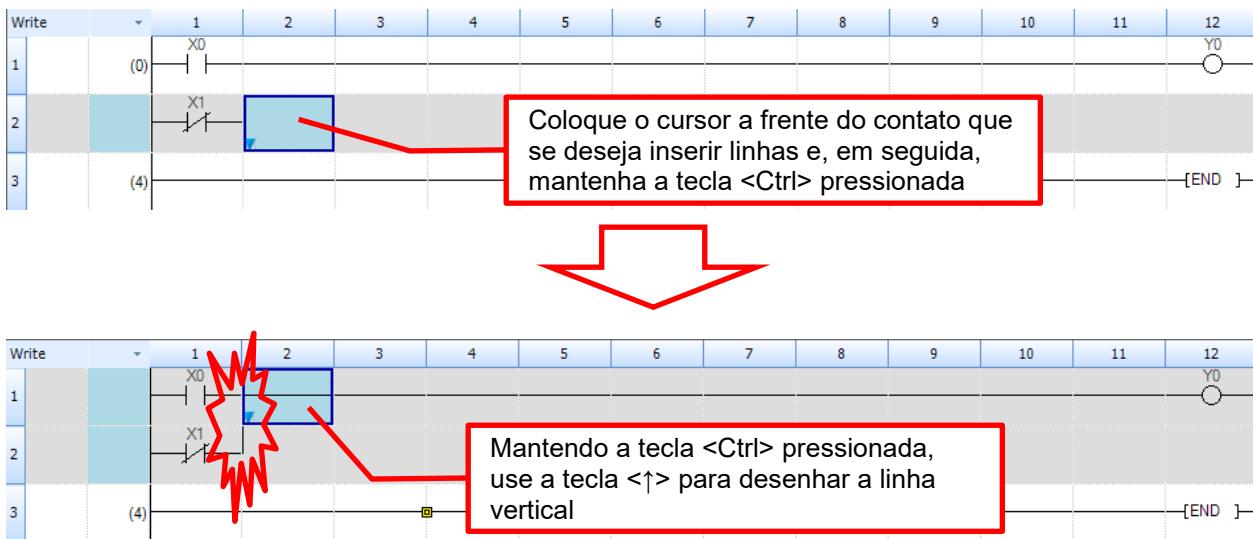


Figura 50: Inserção de linhas de ligação de lógicas no ladder.

Outras formas alternativas de inserção de linhas estão disponíveis na barra de ferramentas, como ilustrado na Figura 51.

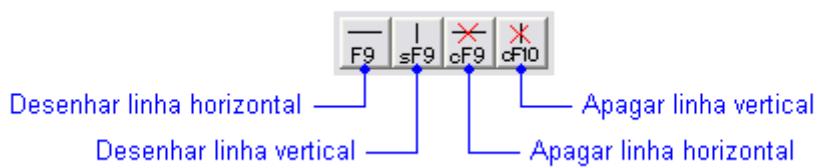


Figura 51: Ferramenta alternativas de inserção de linhas de ligação das lógicas ladder.

Caso queira utilizar uma ferramenta específica para desenhar apenas linhas horizontais, use o atalho <F9>. Aparecerá a seguinte janela:

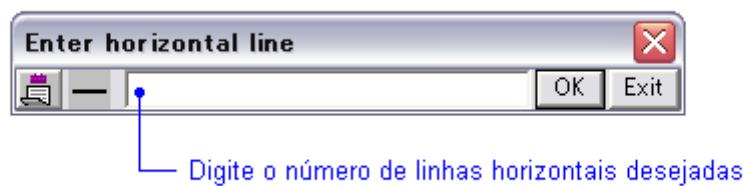


Figura 52: Janela da ferramenta de inserção de linha <F9>.

Ao pressionar <Enter> ou clicar em [OK] sem ter digitado nenhum número no espaço indicado, por *default*, o programa irá desenhar uma única linha onde o cursor estiver.

Para linhas verticais o procedimento será o mesmo. Selecione a ferramenta através do atalho: <Shift> + <F9>. Note que a linha vertical será desenhada para baixo onde o cursor estiver.

O procedimento para apagar apenas linhas horizontais (<Ctrl> + <F9>) ou apenas linhas verticais (<Ctrl> + <F10>) será o mesmo.

3.7.3 Inserir instruções especiais

Na programação de CLP da Mitsubishi, contatos e bobinas representam apenas estados imediatos (não retentivos) de entradas e saídas. No caso de utilizar instruções especiais ou instruções de manipulação de dados dos registradores será necessário utilizar a seguinte ferramenta para inserir tais instruções no programa.



Pressione a tecla <F8>

Figura 53: Ferramenta para inserção de instruções especiais de ladder.

Aparecerá a janela ilustrada na Figura 54. Caso conheça a instrução, digite-a tecle <Enter> ou clique [OK].

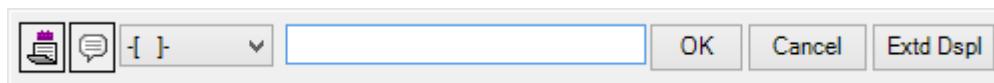


Figura 54: Janela para inserção de instrução especial.

Caso não saiba a instrução a digitar e os seus argumentos, clique em [Extd Dspl]. Aparecerá a janela ilustrada na Figura 55, onde pode-se fazer uma busca pela instrução desejada ou abrir detalhes sobre a instrução utilizando-se o botão [Manual].

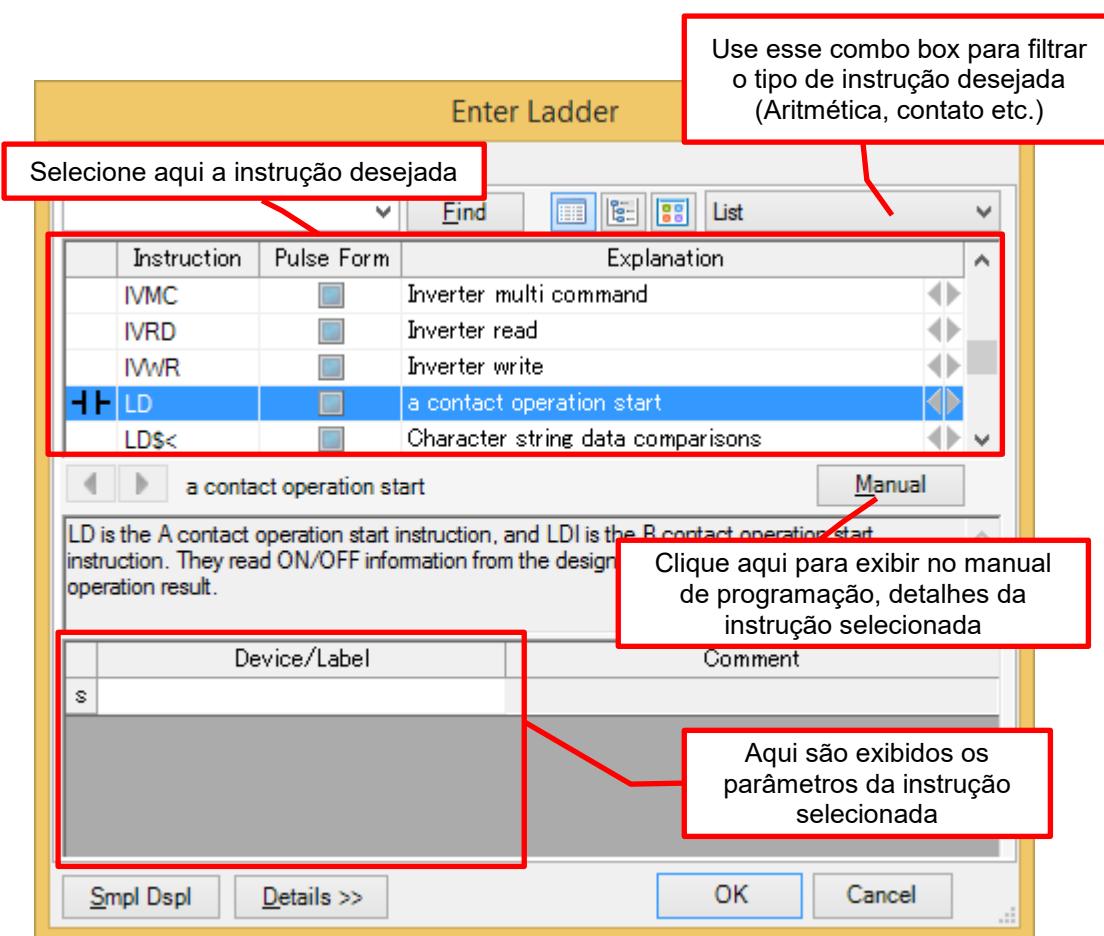


Figura 55: Janela detalhada [Extd Dspl] para entrada da instrução especial.

3.8 EXERCÍCIOS

4.5.1. Construir no GX Works3, o seguinte programa ladder.



4. INSTRUÇÕES BINÁRIAS

4.1 Instruções de contato de pulso e detecção de borda

Por vezes em uma lógica, há necessidade de detectar quando um bit altera o seu estado de ligado para desligado ou vice-versa (detecção de borda ou flanco de subida e descida). Para esses casos, há duas instruções que podem ser utilizadas. A primeira delas é o contato de pulso de subida ou de descida, quando é necessário detectar a borda de um único bit ou, a segunda opção, que é o uso das instruções MEP (detecção de borda de subida de uma lógica) e MEF (detecção de borda de descida de uma lógica).

4.1.1 Contato de pulso de subida e descida

Os contatos de pulso de subida e descida, quando associados a um bit, detectam a ocorrência de uma borda desse bit associado. Quando houver essa detecção, o contato retorna para a lógica um pulso que tem duração de uma varredura do programa (um ciclo de execução do programa). O contato de pulso de subida pode ser inserido pelo teclado usando <Shift> + <F7> ou pela barra de ferramentas pelo ícone . No caso do contato para borda de descida, utiliza-se <Shift> + <F8> ou .

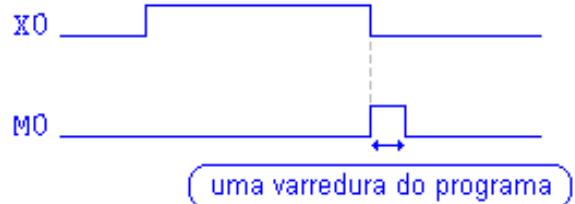


Figura 56: Exemplo das instruções de contato de pulso.

4.1.2 Instruções MEP e MEF

As instruções MEP e MEF são úteis quando se deseja detectar a borda de subida ou de descida de uma lógica, caso em que não se pode usar os contatos de pulso. As instruções MEP e MEF apresentam-se no ladder com a representação de uma flecha para cima (MEP) ou uma flecha para baixo (MEF), cruzada com a linha de ladder, como ilustra a figura. Para inserir uma MEP, use <Alt> + <F5> ou . No caso da MEF, use <Ctrl> + <Alt> + <F5> ou .

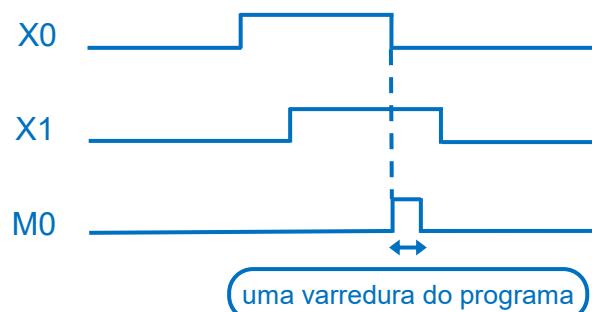
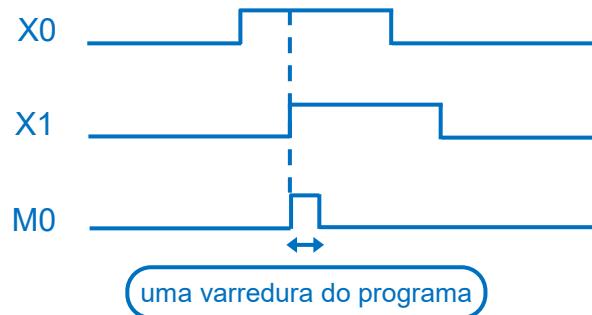
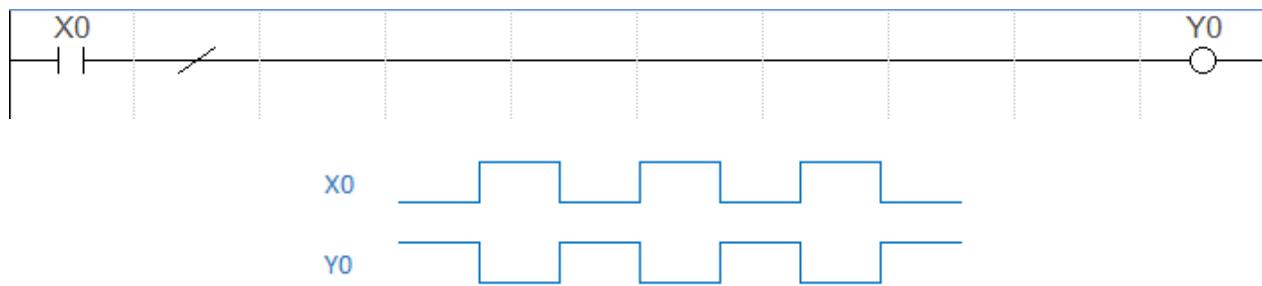


Figura 57: Exemplo instruções MEP e MEF.

4.1.3 Instrução INV

A instrução INV é útil quando você precisa fazer uma lógica para inverter o sinal da saída, sem ter que realizar muitos comandos com contatos NA e NF.



4.2 Instruções SET e RST (Reset)

A instrução SET é uma saída ativa com retenção, ela é ativada com um conjunto verdadeiro de condições. A instrução RST é uma saída especial vinculada (desativa), ela é ativada como um resultado de um conjunto verdadeiro de condições. Aparece sempre como último contato na linha de programação. Um exemplo é exibido na Figura 58.

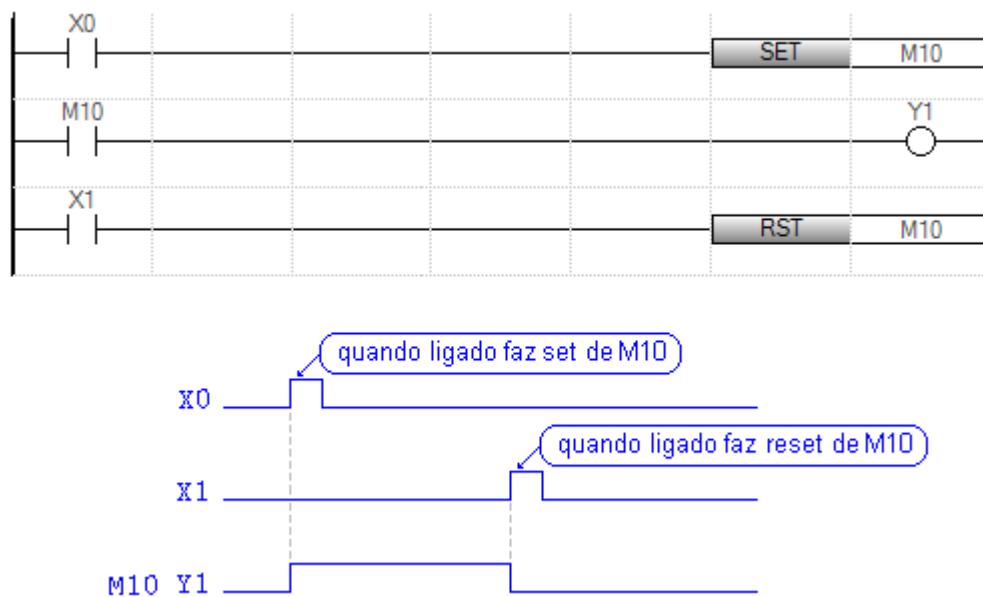


Figura 58: Exemplo instrução SET e RST.

4.3 Instrução FF (Alternate)

A instrução FF é uma saída que deve alternar o estado do bit selecionado de acordo com a borda de subida da entrada.

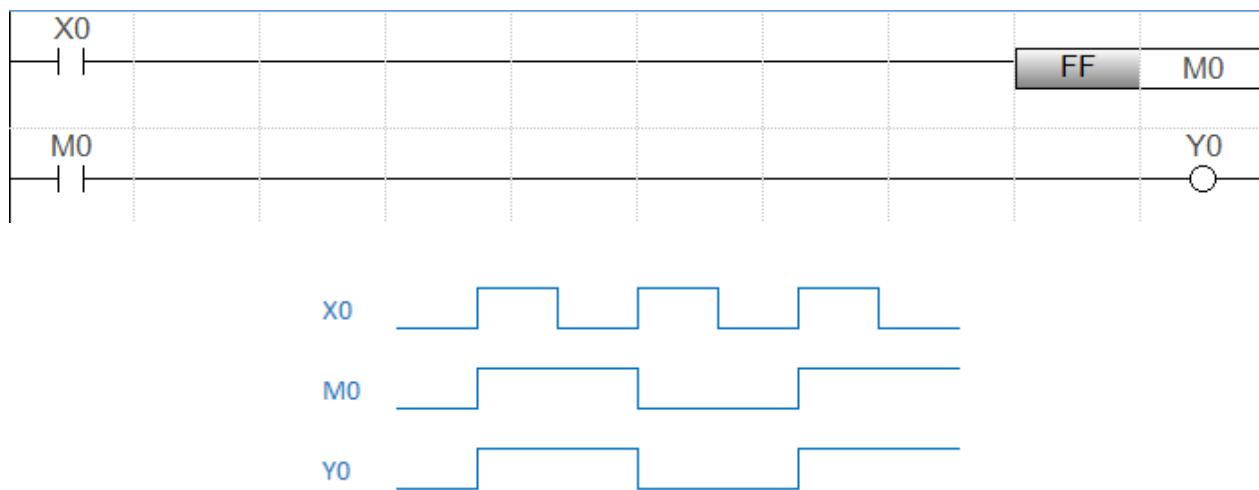


Figura 59: Exemplo de instrução FF

4.4 Exercício

1) Faça um programa onde a esteira se movimente para frente (Y12) e para trás (Y12), apenas se, o modo AUTO (0) estiver acionado e o botão START (X22) seja pressionado. Quando em funcionamento, a esteira deve se acionar a lâmpada RUN (Y21). Ao pressionar o botão RESET (X21), toda a operação deve parar.

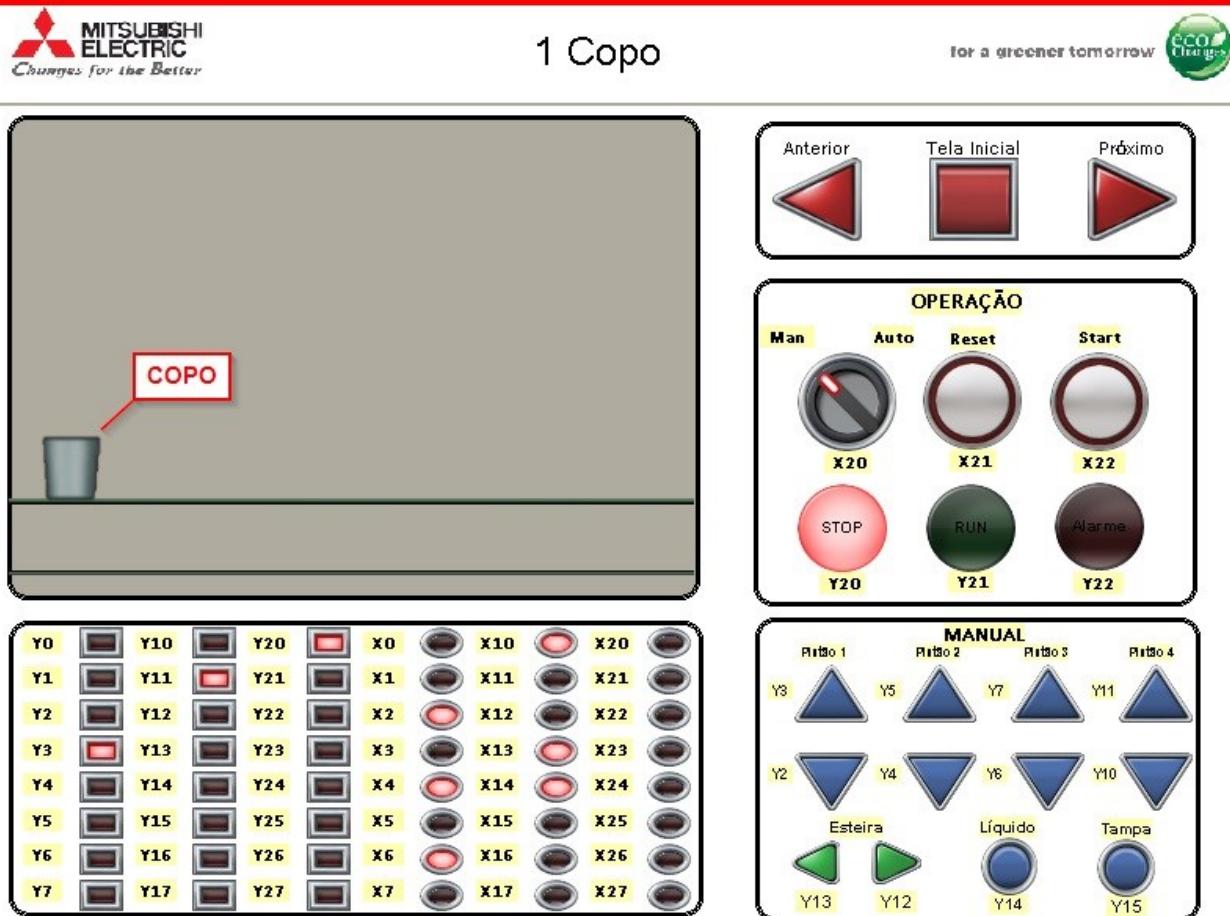


Figura 60: Tela 1 Copo do projeto treinamento

2) Faça um programa onde o PISTÃO 1 suba e desça até acionar os sensores, apenas se, o modo AUTO (X20) estiver acionado e o botão START (X22) seja pressionado. Quando em funcionamento, o PISTÃO 1 ativa a lâmpada RUN (Y21). Ao pressionar o botão RESET (X21), toda a operação deve parar, o PISTÃO 1 deve voltar a sua posição inicial e a lâmpada STOP (Y20) deve ser acionada.

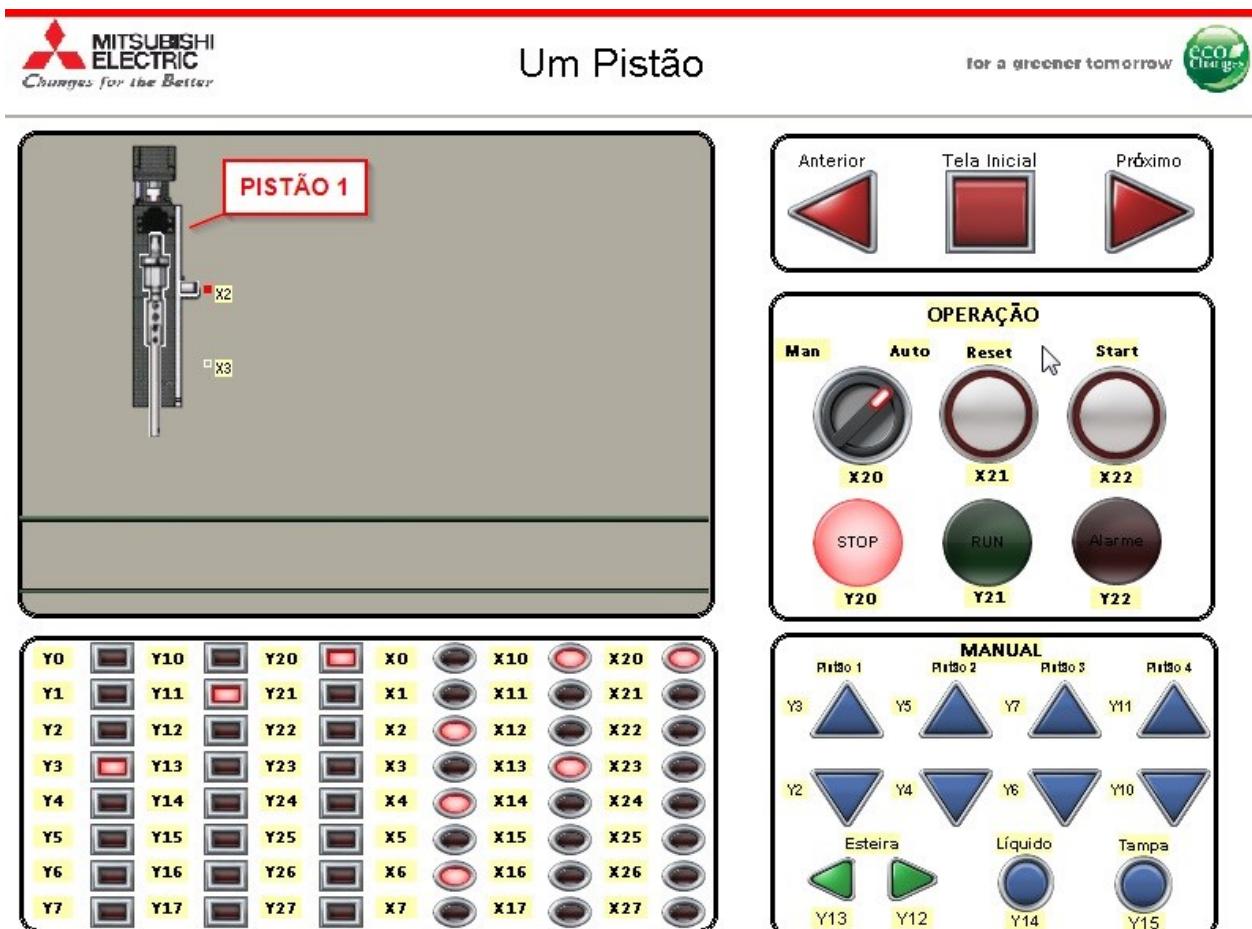


Figura 60: Tela Um Pistão do projeto treinamento

3) Faça um programa onde o PISTÃO 1 desça, até o sensor X3 acionar, cada vez que um copo passar. Esta operação ocorre, apenas se, o modo AUTO (X20) estiver acionado e o botão START (X22) seja pressionado. Quando em funcionamento, o PISTÃO 1 ativa a lâmpada RUN (Y21). Ao pressionar o botão RESET (X21), toda a operação deve parar, o PISTÃO 1 deve voltar a sua posição inicial e a lâmpada STOP (Y20) deve ser acionada.

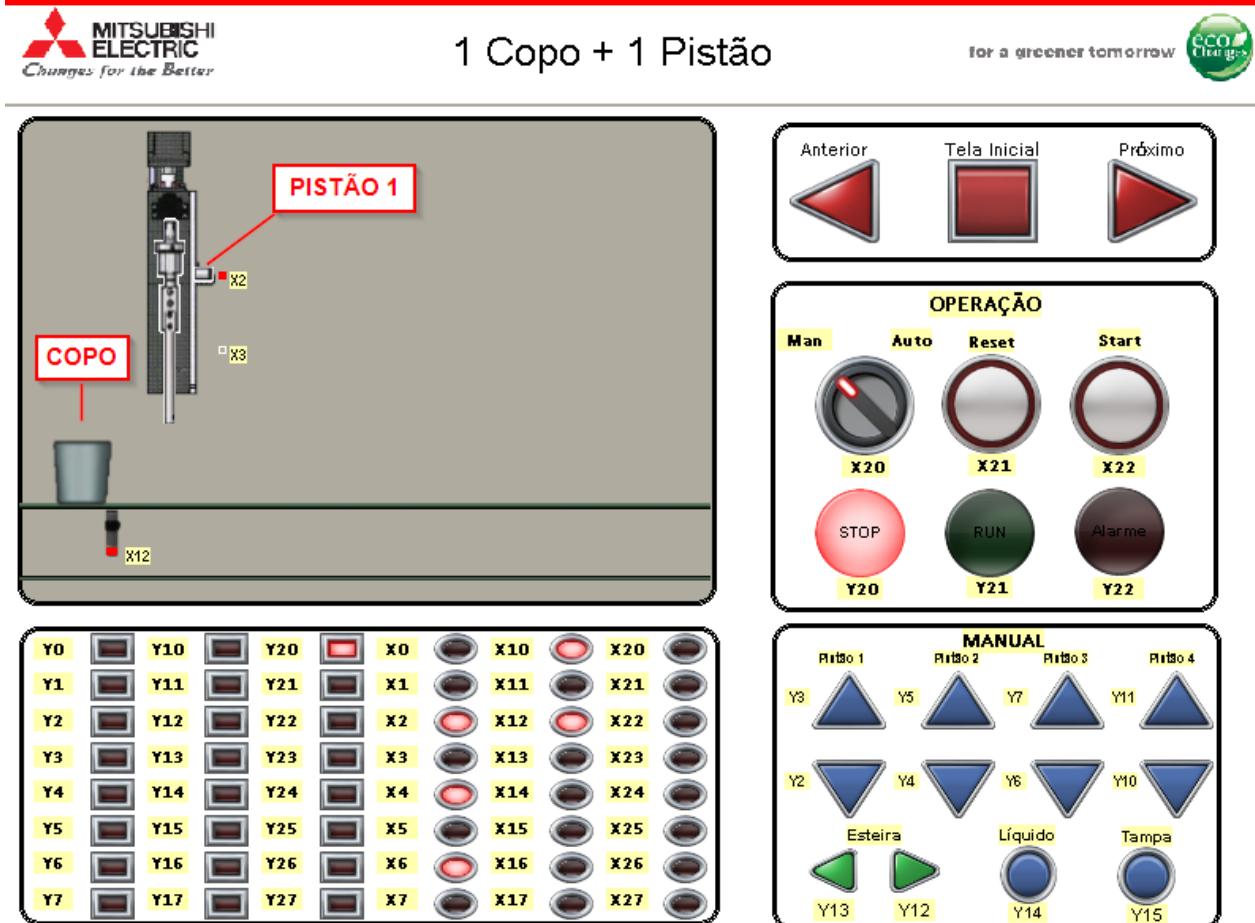


Figura 61: Tela 1 copo+1 Pistão Treinamento

5. TEMPORIZADORES E CONTADORES

5.1 Temporizadores

Temporizadores são instruções destinadas a registro de tempo no PLC. No FX5/GX Works3 há uma variedade de temporizadores que podem ser utilizados. Neste treinamento serão explicados alguns dos principais temporizadores, conforme tabela a seguir.

Instrução	Detalhes
OUT T	Temporizador on-delay low speed Mitsubishi
OUTH T	Temporizador on-delay Mitsubishi
OUTHS T	Temporizador on-delay high speed Mitsubishi
OUT ST	Temporizador retentivo on-delay low speed Mitsubishi
OUTH ST	Temporizador retentivo on-delay Mitsubishi
OUTHS ST	Temporizador on-delay high speed Mitsubishi

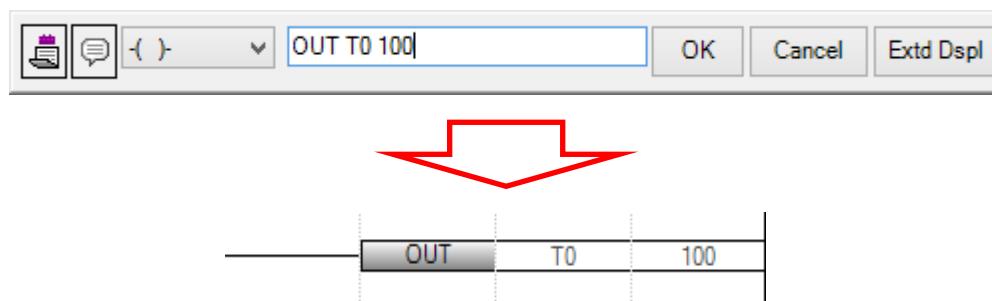
5.1.1 Temporizadores OUT T, OUTH T, OUTHS T

Os temporizadores OUT T, OUTHT e OUTHS T são temporizadores on-delay em que a diferença entre eles é a base de tempo. Esses temporizadores têm o set point como um valor inteiro (Word [Signed]) e, esse valor inteiro é multiplicado pela base de tempo do temporizador para determinar o tempo a ser atingido. As bases de tempo dos temporizadores com OUT são dadas na tabela abaixo. Esses temporizadores necessitam de um endereço tipo “Timer”, que pode ser um dispositivo T ou um label do tipo “Timer”.

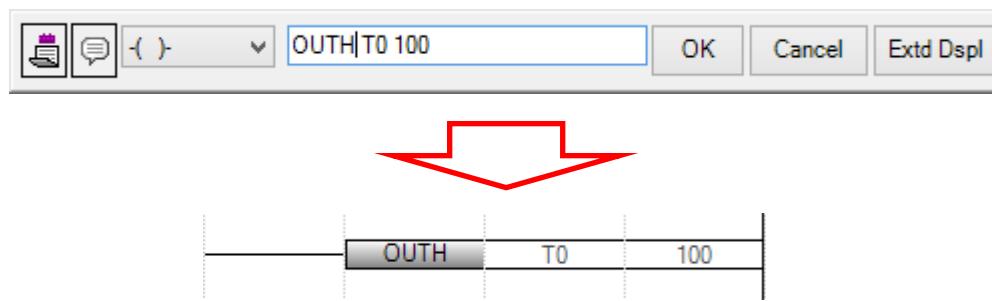
Instrução	Base de tempo / limite de set point em segundos
OUT T	100 ms / 0,1 a 3276,7 s
OUTH T	10 ms / 0,01 a 327,67 s
OUTHS T	1 ms / 0,001 a 32,767 s

No GX Works3, os temporizadores são adicionados utilizando o atalho de instrução especial <F8> (ou bobina <F7>) e digitando-se OUTH, OUTHS ou OUT, espaço o endereço do dispositivo temporizador (T) e o set point que pode ser constante (valor numérico) ou variável (nesse caso precisa ser declarado um registrador ou label tipo “Word [Signed]”). São ilustrados exemplos na Figura utilizando dispositivo T e, na Figura , utilizando label.

Temporizador low speed OUT



Temporizador OUTH



Temporizador high speed OUTHS

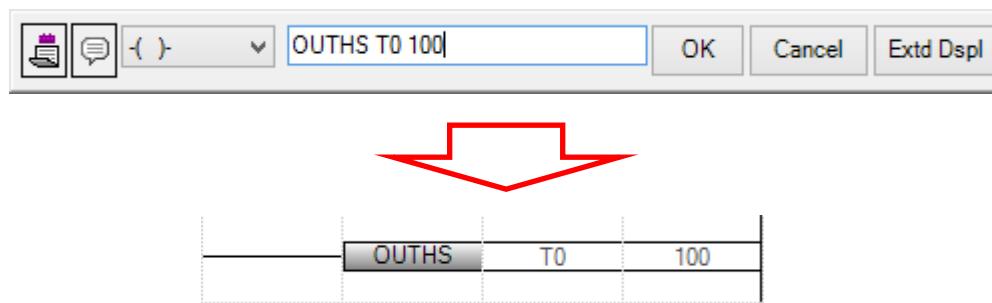


Figura 61: Exemplos de entrada de temporizadores OUT T no ladder com variável dispositivo.

Temporizador OUT, OUTH ou OUTHS com label (declaração da variável label)

Label Name		Data Type	Comment
1	Tempo_1	Timer	...
2			...

É possível usar OUT, OUTH ou OUTHS com label

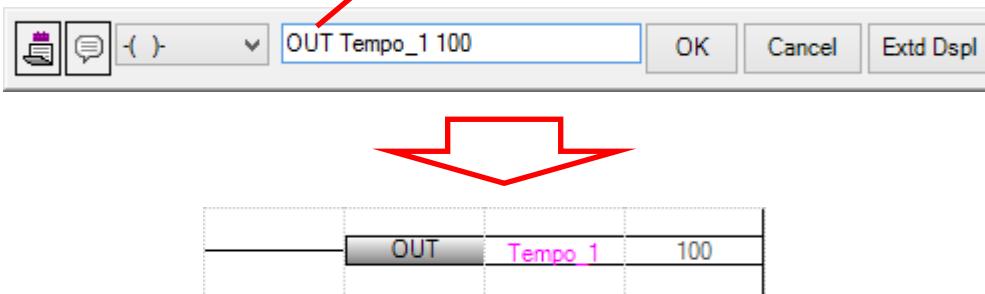


Figura 62: Exemplos de entrada de temporizadores OUT T no ladder com variável label.

Operação do temporizador OUT T

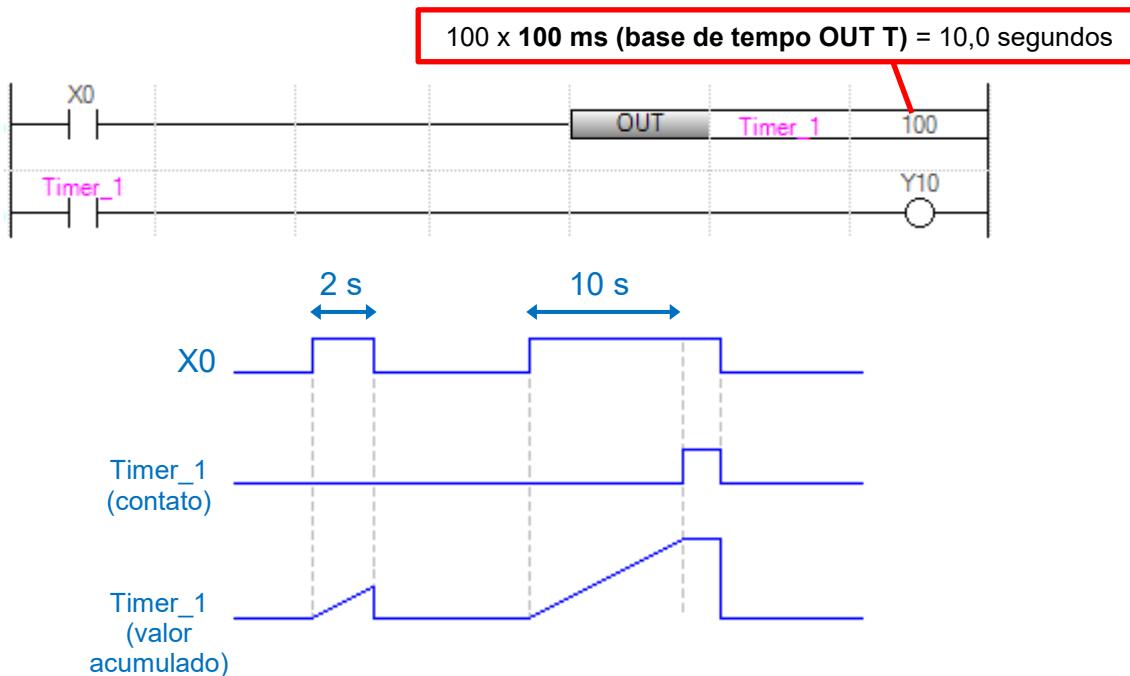


Figura 63: Operação do temporizador OUT T.

Operação do temporizador OUTH T

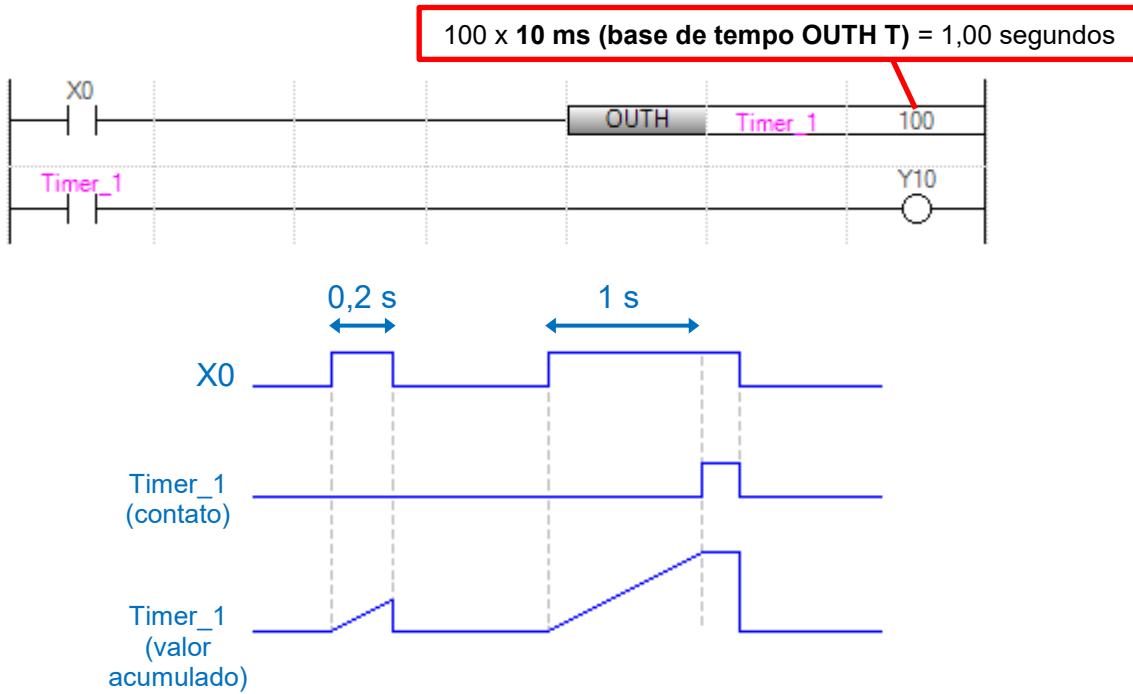


Figura 64: Operação do temporizador OUTH T.

Operação do temporizador OUTHS T

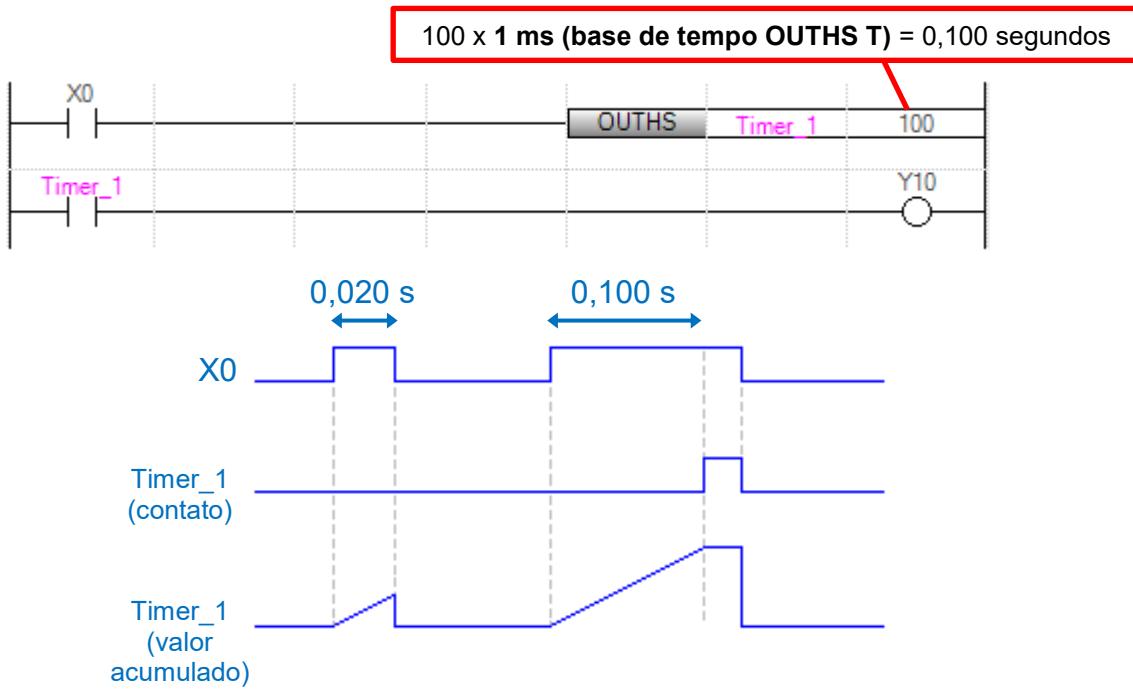


Figura 65: Operação do temporizador OUTH T.

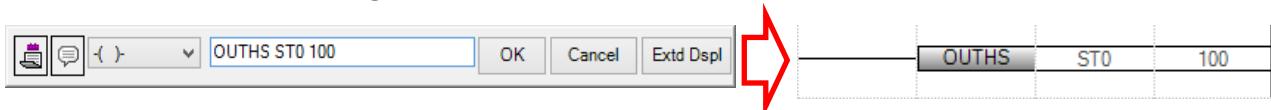
5.1.2 Temporizadores retentivos OUT ST, OUTH ST, OUTHS ST

Os temporizadores OUT ST, OUTH ST e OUTHS ST são temporizadores on-delay, assim como os temporizadores OUT T, OUTH T e OUTHS T porém, a diferença é que são retentivos, ou seja, se uma entrada que aciona o temporizador desligar, o valor acumulado do temporizador não volta a zero, se mantém no valor atual e, continuará temporizando a partir desse valor quando a entrada que aciona o temporizador voltar a ligar. Da mesma forma que os temporizadores não retentivos, têm o set point como um valor inteiro (Word [Signed]) e, esse valor inteiro é multiplicado pela base de tempo do temporizador para determinar o tempo a ser atingido. Esses temporizadores necessitam de um endereço tipo “Retentive Timer”, que pode ser um dispositivo ST ou um label do tipo “Retentive Timer”.

Instrução	Base de tempo / limite de set point em segundos
OUT ST	100 ms / 0,1 a 3276,7 s
OUTH ST	10 ms / 0,01 a 327,67 s
OUTHS ST	1 ms / 0,001 a 32,767 s

A seguir, será dado um exemplo dos temporizadores retentivos usando a instrução OUTHS ST. Esse exemplo aplica-se às instruções OUTH ST e OUTHS ST, alterando-se as bases de tempo.

Temporizador retentivo high speed OUT com dispositivo ST



Temporizador retentivo high speed OUT com label

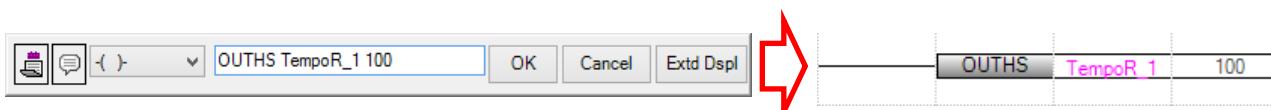
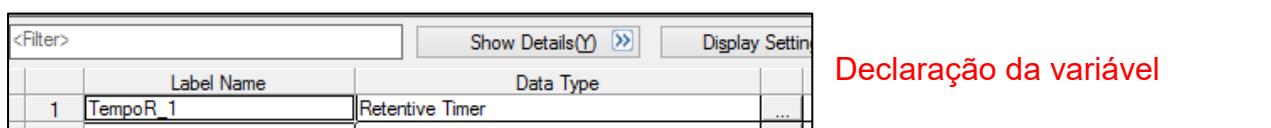


Figura 66: Exemplos de entrada de temporizadores OUT ST no ladder.

Para reiniciar os temporizadores retentivos, deve-se utilizar uma instrução reset (RST).

5.2 EXERCÍCIOS

1) Faça um programa onde o PISTÃO 2 deve descer, até o sensor X5 acionar. Ao mesmo tempo que isso ocorre, o botão LÍQUIDO (Y14) deve ser acionado, para que o PISTÃO 3 desça, e acione o sensor X7. Após 2 segundos da descida dos PISTÕES 2 e 3, ambos devem voltar as suas posições iniciais. O funcionamento dos pistões deve ocorrer toda vez que o sensor **X13** for acionado. Quando em funcionamento os pistões e a esteira acionam a lâmpada RUN (Y21). Todas essas funções devem ocorrer em modo AUTO (**X20**) e, apenas se, o botão START (**X22**) for pressionado. Ao pressionar o botão RESET (X21), toda a operação deve parar, os PISTÕES 2 e 3 devem voltar as suas posições iniciais e a lâmpada STOP (Y20) deve ser acionada.

Temporizador

For a greener tomorrow

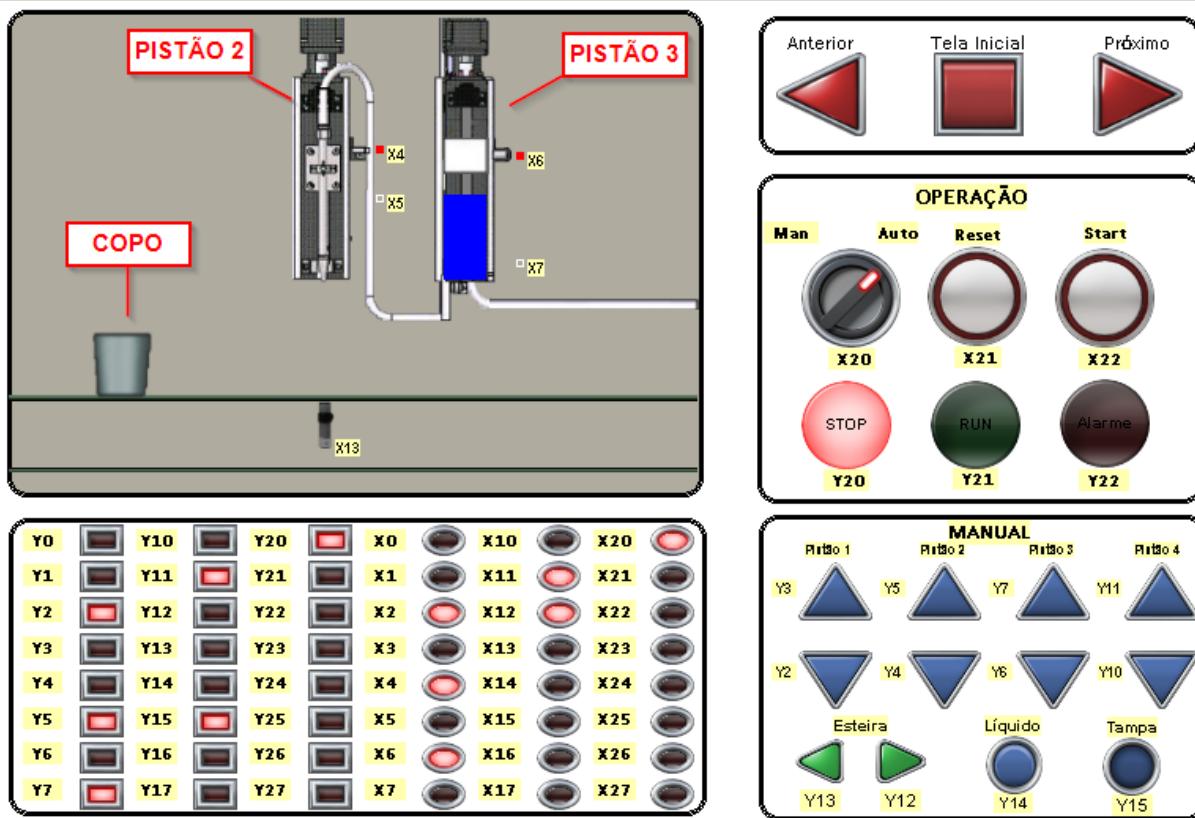


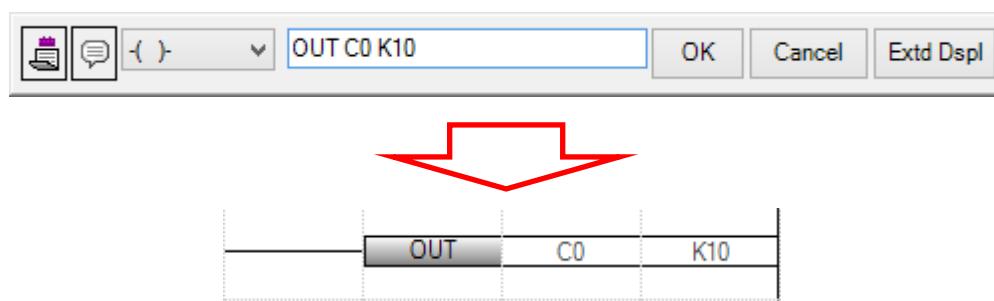
Figura 69: Exemplos de operação do contador.

5.3 Contadores

Os contadores são instruções tipo bobina destinadas a acumulação de valor em função de pulsos gerados por uma variável tipo bit. Os contadores na CPU FX5 estão disponíveis em dois tipos: “Counter” (16 bits, de 0 a 32767) e “Long Counter” (32 bits sem sinal, de 0 a 4294967295). O que determina se o contador é do tipo “Counter” ou “Long Counter” é o tipo de variável associada a instrução OUT.

Contador de 16 bits (Counter)

Com dispositivo



Com label

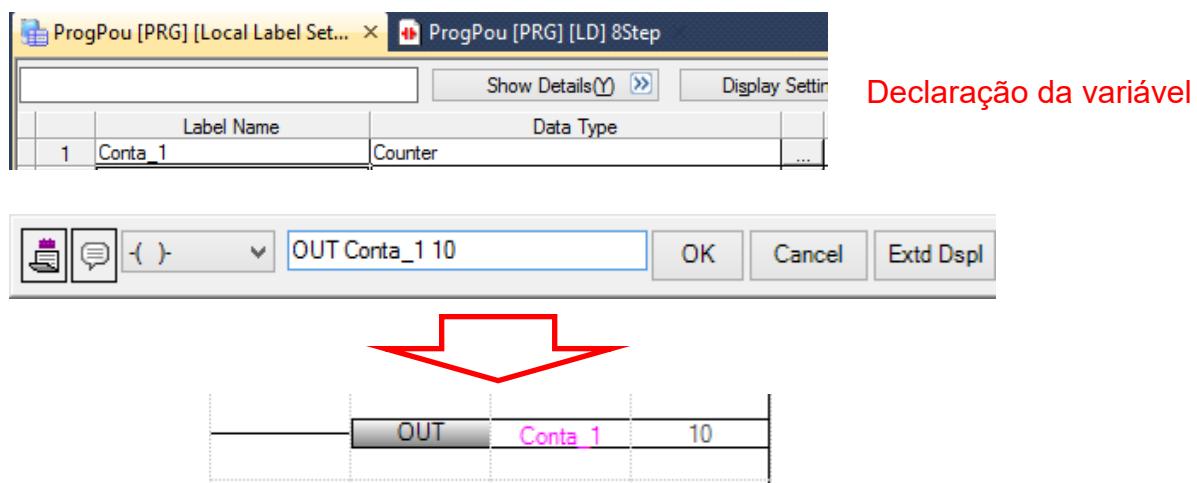
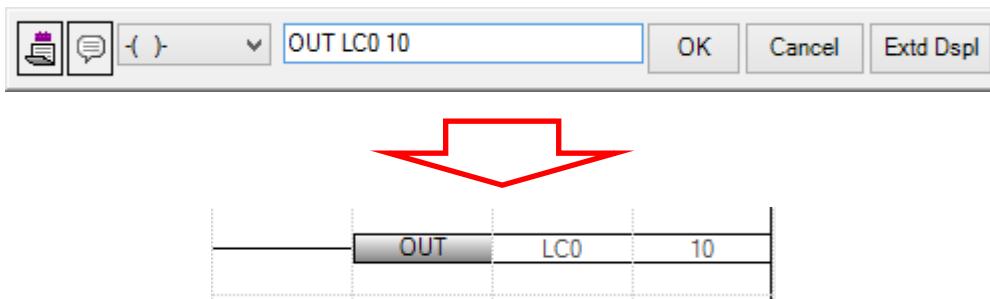


Figura 67: Exemplos de entrada de contadores OUT C no ladder.

Contador de 32 bits sem sinal (Long Counter)

Com dispositivo



Com label

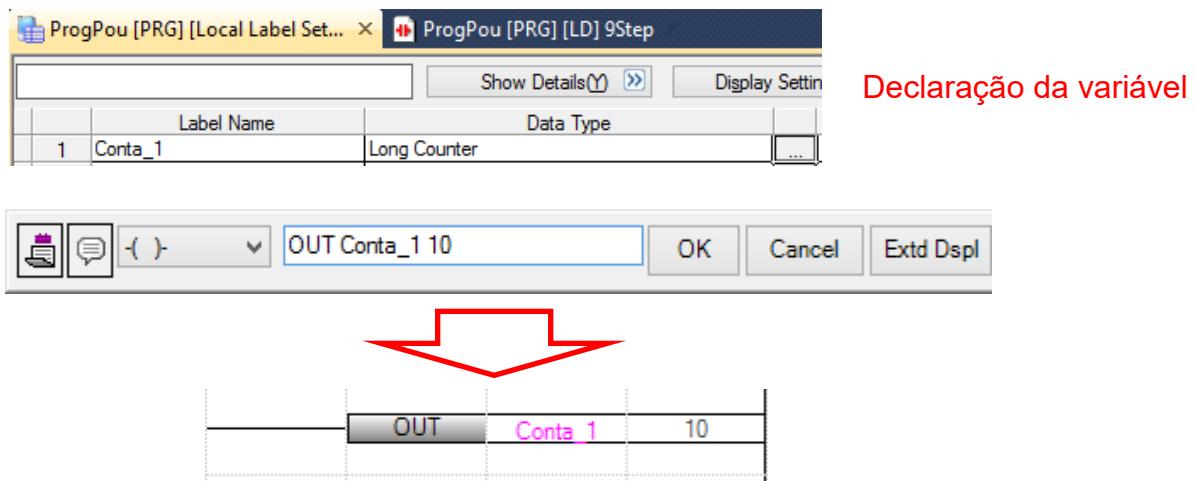


Figura 68: Exemplos de entrada de contadores OUT LC no ladder.

Operação do contador

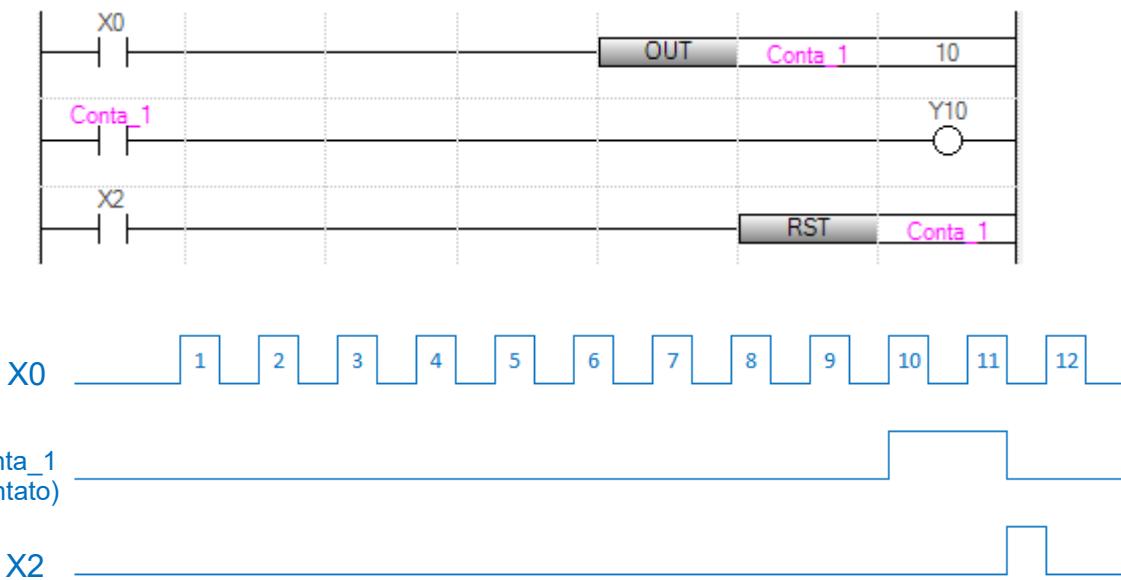
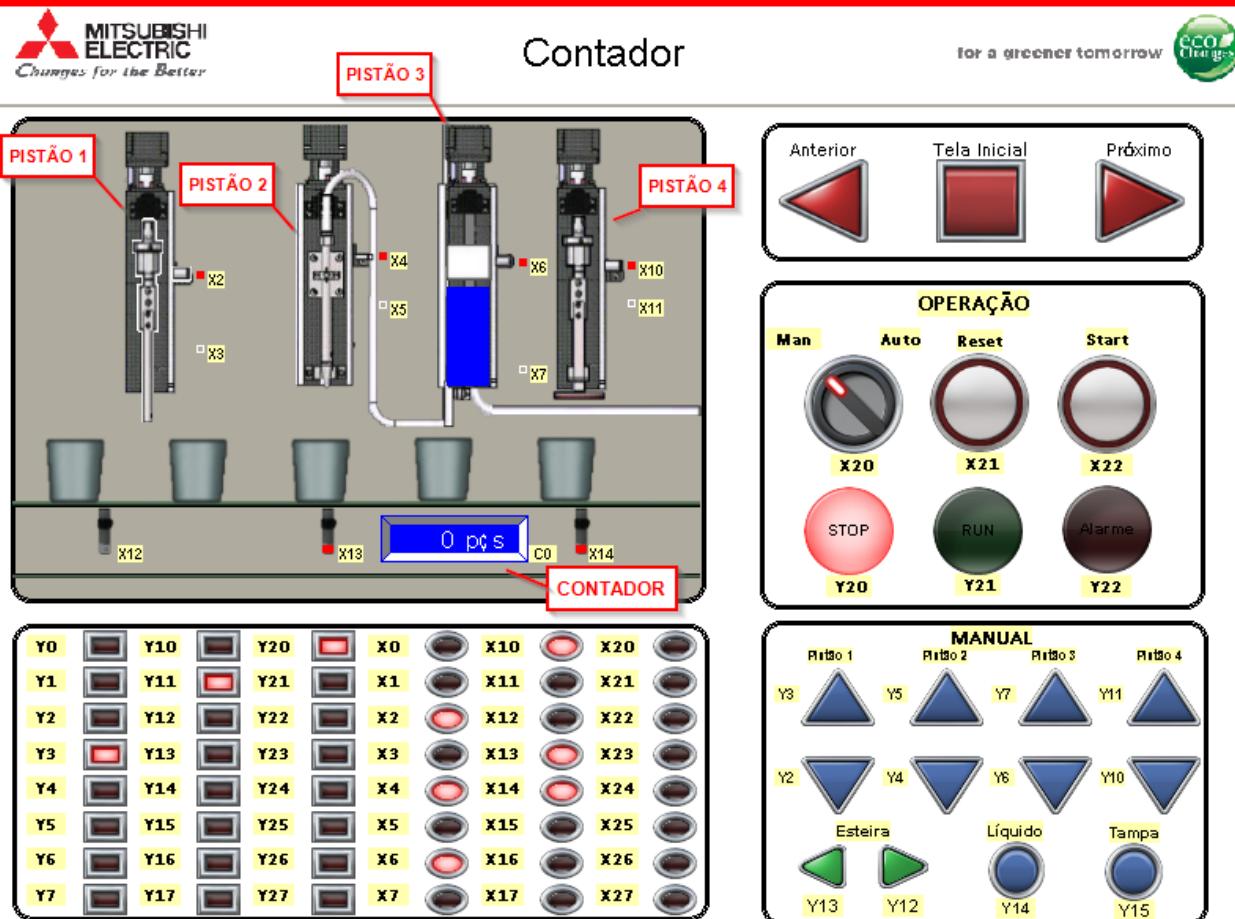


Figura 69: Exemplos de operação do contador.

5.4 EXERCÍCIOS

2) Faça um programa onde o PISTÃO 1 desça, até o sensor X3 acionar, cada vez que um copo passar. O PISTÃO 2 deve descer, até o sensor X5 acionar. Em seguida o botão LÍQUIDO (Y14) deve ser acionado, para que o PISTÃO 3 desça, acione o sensor X7, e os PISTÕES 2 e 3, retornem a suas posições originais. Após isso, o PISTÃO 4, deve descer até acionar o sensor X11 e o botão TAMPA (X15) deve ser pressionado. Então o PISTÃO 4 retorna a sua posição original. A cada copo pronto, o contador deve incrementar um. Quando em funcionamento os pistões e a esteira acionam a lâmpada RUN (Y21). Todas essas funções devem ocorrer em modo AUTO (X20) e, apenas se, o botão START (X22) for pressionado. Ao pressionar o botão RESET (X21), toda a operação deve parar, os PISTÕES 1,2,3 e 4 devem voltar as suas posições iniciais e a lâmpada STOP (Y20) deve ser acionada.



6. INSTRUÇÕES DE DADOS NUMÉRICOS

Neste capítulo serão detalhadas as principais instruções de manipulação de dados numéricos do FX5.

6.1 *Introdução*

Os dados numéricos, em qualquer máquina digital – como os PLCs – são armazenados e processados como um número em formato binário, ou seja, uma notação numérica composta de dígitos de 1 e 0. Cada dígito do número binário é um bit da memória e, portanto, o range de valores que pode ser armazenado em uma área de memória depende de quantos bits essa área é composta.

Outro fato importante a se notar é que, os dados numéricos utilizados pelo homem se apresentam em diversos formatos, sendo que os mais utilizados são diferentes do formato binário (geralmente decimal, variando entre inteiro positivo, inteiro negativo, real etc). Sendo o formato binário a única forma de armazenamento e processamento possível no PLC, é necessária uma conversão dos valores utilizados pelo homem em notação binária. Para cada formato de dado há uma conversão para o formato binário diferente e, em alguns casos, há mais de uma forma possível de conversão para o binário.

Com isso, concluímos que o valor binário armazenado em uma determinada área de memória de dados apenas pode ser interpretado ou convertido se for conhecida a formatação dos dados representados em binário nessa área e que, não se pode processar dados com informações de áreas formatadas diferentemente, sem antes converter ao menos uma delas para que fiquem com o mesmo formato.

Nos PLCs Mitsubishi, a interpretação do formato de dados armazenado em uma área de memória é dependente da instrução de processamento utilizada. Para formatos de dados que tenham possibilidade de realizar o mesmo tipo de processamento, como por exemplo uma soma, há uma instrução de soma específica para cada formato (há uma soma para inteiros representados em 16 bits, inteiros representados em 32 bits, reais etc).

Em particular, o PLC da série iQ-F FX5, pode manipular através de instruções, dados dos tipos (algumas instruções não preveem todos os formatos):

- a) Números inteiros codificados em formato binário de 16 bits com sinal;
- b) Números inteiros codificados em formato binário de 16 bits sem sinal;
- c) Números inteiros codificados em formato binário de 32 bits com sinal;
- d) Números inteiros codificados em formato binário de 32 bits sem sinal;
- e) Números inteiros em codificação BCD (Decimal Codificado em Binário);
- f) Números reais representados em notação de ponto flutuante de simples precisão IEEE-754 (32 bits);
- g) Cadeias de caracteres codificadas em formato ASCII (*American Standard Code for Information Interchange*);
- h) Notação de tempo IEC, expressa como T#12d4h3m2s1ms (dias, horas, minutos, segundos e milissegundos).

Utilizando o software GX Works2 para programação em ladder, a interpretação dos dados de acordo com os tipos citados anteriormente, depende da instrução empregada. Na tabela abaixo é exemplificada essa metodologia com a instrução de soma para diversos formatos binários

Instrução de soma	Formato numérico
B+	Soma de valores numéricos formatados em BCD 4 dígitos
DB+	Soma de valores numéricos formatados em BCD 8 dígitos
+	Soma de valores numéricos formatados inteiros de 16 bits com sinal
+_U	Soma de valores numéricos formatados inteiros de 16 bits sem sinal
D+	Soma de valores numéricos formatados inteiros de 32 bits com sinal
D+_U	Soma de valores numéricos formatados inteiros de 32 bits sem sinal
E+	Soma de valores numéricos formatados em ponto flutuante de simples precisão (32 bits)

No caso de utilização de variáveis (áreas de memória para armazenamento de dados) do tipo dispositivo (device), é importante lembrar que para os dados que ocupam 32 bits, não existe um dispositivo de 32 bits, são utilizados dois registradores em endereços contíguos (por exemplo, D0 e D1) para o armazenamento desse tipo de dado numérico.

Dessa forma, instruções como a D+ e E+ ocupam dois registradores contíguos para o processamento e armazenamento do resultado, como no exemplo a seguir, da Figura 62.

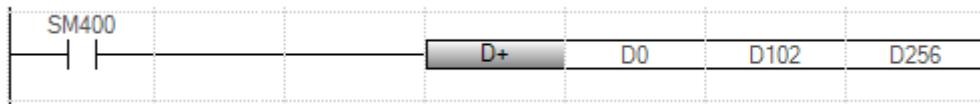


Figura 62: Exemplo de processamento de instrução de 32 bits com dispositivos.

No exemplo da Figura 62, é importante notar que para a soma serão processados os dados de dois registradores contíguos de cada uma das parcelas, ou seja, o dado armazenado em D0-D1 combinados e D102-D103 combinados e o resultado será armazenado em D256-D257 combinados formando 32 bits. **Assim é importante que o registrador D257 não seja utilizado com outra função que não seja o armazenamento da parte mais significativa dos 32 bits do dado formado em D256-D257 combinados.**

Nas seções a seguir, serão detalhadas as principais instruções que manipulam dados numéricos.

6.2 Representação de constantes numéricas

No FX5 é possível trabalhar com representações de constantes numéricas em diversos formatos. Esses formatos são determinados por um prefixo adicionado ao número. Na tabela abaixo, as representações de constantes numéricas e seus significados.

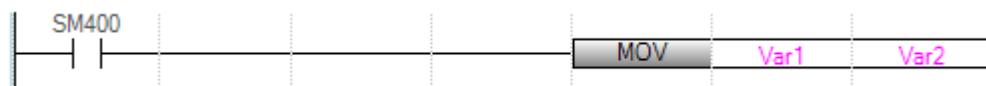
Sufixo	Descrição	Exemplo
K ou (vazio)	Constante decimal	K-30000 ou -30000
H ou 16#	Constante hexadecimal	H4FD ou 16#4FD
8#	Constante octal	8#17
2#	Constante binária	2#101101
E ou (vazio, com valor numérico utilizando ponto)	Constante real em ponto flutuante. O sinal de “+” ou “-” após o valor numérico indica o valor do expoente de 10 em notação científica (ex.: E1.6+4 = 1.6 x 10 ⁴)	E10 ou 10.0 E1.6+4 ou 1.6+4
‘ ‘ (entre aspas simples)	Cadeia de caracteres (string)	‘ABC’, ‘!34@’
T# d h m s ms	Constante de tempo	T#30s (= 30 seg) T#4h2m (= 4 hrs e 2 min)

É possível usar “_” na constante numérica para facilitar a visualização, por exemplo “12_3” é processado como “123” no PLC.

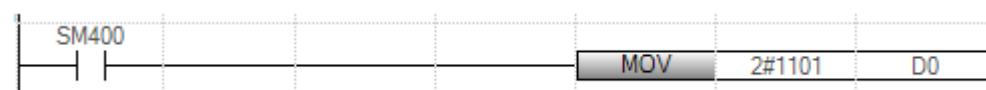
6.3 Instrução MOV

É utilizada para mover dados entre variáveis ou escrever. A Figura 63 ilustra um exemplo.

O valor armazenado em Var1 é copiado em Var2 (Word[Signed])



O valor binário 1101 é armazenado nos bits menos significativos de D0



O valor armazenado em D40 é copiado em D50

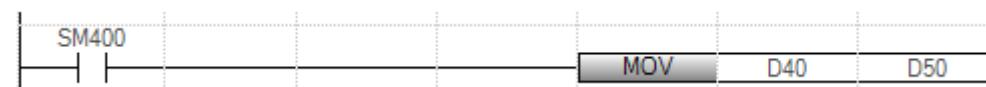


Figura 63: Instrução MOV, exemplo no formato 16 bits com sinal.

Instrução MOV para outros formatos numéricos:

Instrução de soma	Formato numéricico
MOVB	Copia dados de 1 bit
DMOV	Copia dados de 32 bits
EMOV	Copia dados em formato de ponto flutuante

6.4 Instruções de comparação de magnitude

A operação de comparação realiza comparação numérica de magnitude entre dois valores. Essas comparações são feitas em forma de contato que é ativado quando a condição imposta se torna verdadeira. A aplicação das instruções de comparação é a mesma dos contatos comuns.

As instruções de comparação são:

>(maior)	>=(maior ou igual)
<(menor)	<= (menor ou igual)
= (igual)	<>(diferente)

A Figura 64 ilustra um exemplo desse tipo de instrução.



Figura 64: Exemplo de instrução de comparação.

No exemplo acima enquanto o valor armazenado em Var1 for menor ou igual que 4, a instrução de comparação acima atuará como um contato e a bobina Y10 ficará ativada.

Essas instruções de comparação podem ser colocadas não só como um único contato, mas também introduzido em lógicas.

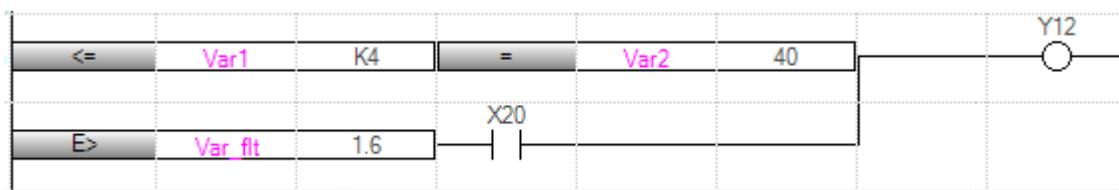


Figura 65: Exemplo de instruções de comparação combinadas em uma lógica.

6.5 Instrução de soma

A instrução “ + ” é utilizada para somar o valor de duas variáveis.



Figura 66: Exemplo da instrução de soma.

Neste caso, o valor de Var1 é adicionado a Var2 e o resultado é armazenado em Var3 (variáveis tipo Word[Signed] – inteiro de 16 bits). Para outros formatos numéricos, veja tabela abaixo.

Instrução	Formato numérico
B+	Soma de valores numéricos formatados em BCD 4 dígitos
DB+	Soma de valores numéricos formatados em BCD 8 dígitos
+	Soma de valores numéricos formatados inteiros de 16 bits com sinal
+_U	Soma de valores numéricos formatados inteiros de 16 bits sem sinal
D+	Soma de valores numéricos formatados inteiros de 32 bits com sinal
D+_U	Soma de valores numéricos formatados inteiros de 32 bits sem sinal
E+	Soma de valores numéricos formatados em ponto flutuante de simples precisão (32 bits)

6.6 Instrução de subtração

A instrução “ - ” é utilizada para subtrair o valor de duas variáveis.



Figura 67: Exemplo da instrução de subtração.

No exemplo da Figura 67, o valor de Var1 é subtraído de Var2 e o resultado é armazenado em Var3 (variáveis tipo Word[Signed] – inteiro de 16 bits). Para outros formatos numéricos, veja tabela a seguir.

Instrução	Formato numérico
B-	Subtração de valores numéricos formatados em BCD 4 dígitos
DB-	Subtração de valores numéricos formatados em BCD 8 dígitos
-	Subtração de valores numéricos formatados inteiros de 16 bits com sinal
_U	Subtração de valores numéricos formatados inteiros de 16 bits sem sinal
D-	Subtração de valores numéricos formatados inteiros de 32 bits com sinal
D_U	Subtração de valores numéricos formatados inteiros de 32 bits sem sinal
E-	Subtração de valores numéricos formatados em ponto flutuante de simples precisão (32 bits)

6.7 Instrução de multiplicação

A instrução “ * ” é utilizada para multiplicar o valor de duas variáveis.



Figura 68: Exemplo da instrução de multiplicação.

No exemplo da Figura 68 o valor de Var1 é multiplicado por Var2 (variáveis tipo Word[Signed] – inteiro de 16 bits) e o resultado é armazenado em Var3_32 (variável tipo Double Word[Signed] – inteiro de 32 bits). **Repare que a instrução de multiplicação em 16 bits resulta em um valor de 32 bits. Isso ocorre por questão de processamento da instrução de multiplicação de valores inteiros.** Para outros formatos numéricos, veja tabela abaixo.

Instrução	Formato numérico
B*	Subtração de valores numéricos formatados em BCD 4 dígitos
DB*	Subtração de valores numéricos formatados em BCD 8 dígitos
*	Subtração de valores numéricos formatados inteiros de 16 bits com sinal
_U	Subtração de valores numéricos formatados inteiros de 16 bits sem sinal
D*	Subtração de valores numéricos formatados inteiros de 32 bits com sinal
D*_U	Subtração de valores numéricos formatados inteiros de 32 bits sem sinal
E*	Subtração de valores numéricos formatados em ponto flutuante de simples precisão (32 bits)

6.8 Instrução de divisão

A instrução “ / ” é utilizada para dividir o valor de uma variável por outra.

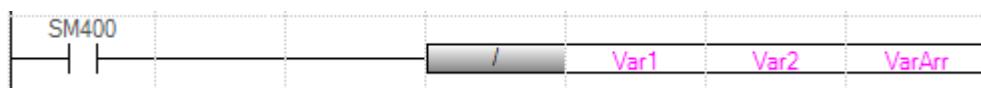


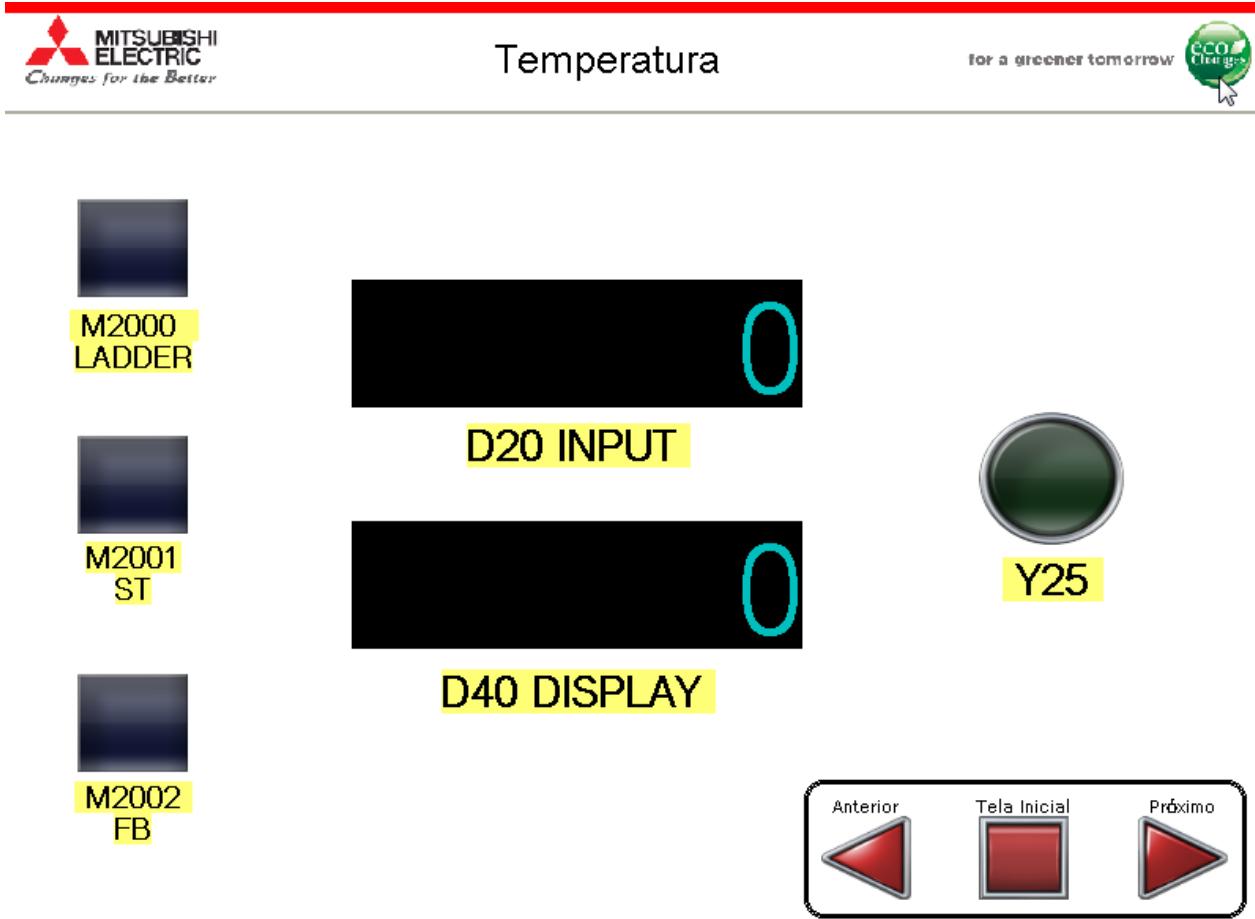
Figura 69: Exemplo da instrução de divisão.

No exemplo da Figura 69, o valor de Var1 é dividido por Var2 (variáveis tipo Word[Signed] – inteiro de 16 bits) e o resultado é armazenado em um array de 2 variáveis de 16 bits (variável de arranjo contendo 2 posições de tipo Word[Signed]). Isso ocorre porque a instrução de divisão do exemplo, é uma instrução para dados inteiros, dessa forma, a divisão poderá ter resto. A variável tipo array com duas posições servirá para que, na primeira posição seja armazenado o resultado e, na segunda posição, seja armazenado o resto da divisão.

Para divisão de outros tipos de formatos numéricos, veja a tabela abaixo.

Instrução	Formato numéricico
B/	Subtração de valores numéricos formatados em BCD 4 dígitos
DB/	Subtração de valores numéricos formatados em BCD 8 dígitos
/	Subtração de valores numéricos formatados inteiros de 16 bits com sinal
/_U	Subtração de valores numéricos formatados inteiros de 16 bits sem sinal
D/	Subtração de valores numéricos formatados inteiros de 32 bits com sinal
D/_U	Subtração de valores numéricos formatados inteiros de 32 bits sem sinal
E/	Subtração de valores numéricos formatados em ponto flutuante de simples precisão (32 bits)

6.8.1 Exercícios



Deseja-se incluir no sistema uma formula para converter temperatura de Fahrenheit em Celsius, de acordo com a formula abaixo em números inteiros:

$$T_{\circ C} = \frac{5}{9} \cdot (T_{\circ F} - 32)$$

- a) Faça a lógica de operação no CLP utilizando o M2000 (ladder) em modo on, considerando $T_{\circ F}$ (D20) é o input e o $T_{\circ C}$ (D40) o output
- b) Implemente uma lógica para que, se a temperatura em Celsius (D40) chegar a 98°C (arredondado para baixo de 210°F) ou mais, a saída Y20 deve ligar

6.9 *Inline ST para expressões matemáticas*

Há um recurso de programação no GX Works3 que permite que expressões matemáticas como equações possam ser construídas no programa sem necessitar construir um programa extenso com operação por operação matemática da expressão individualmente. Esse recurso denomina-se “Inline ST”.

Para introduzir o Inline ST, pressione **<Ctrl> + ** ou clique em  na barra de ferramentas com o cursor na área de edição do ladder onde se deseja adicionar a expressão matemática. A Figura 70 ilustra um exemplo de uso do recurso de programação Inline ST. Nesse exemplo, a expressão $Var1 = (Var2 + 300) / 4$ é inserida no Inline ST e é executada quando X20 estiver ligado liga.

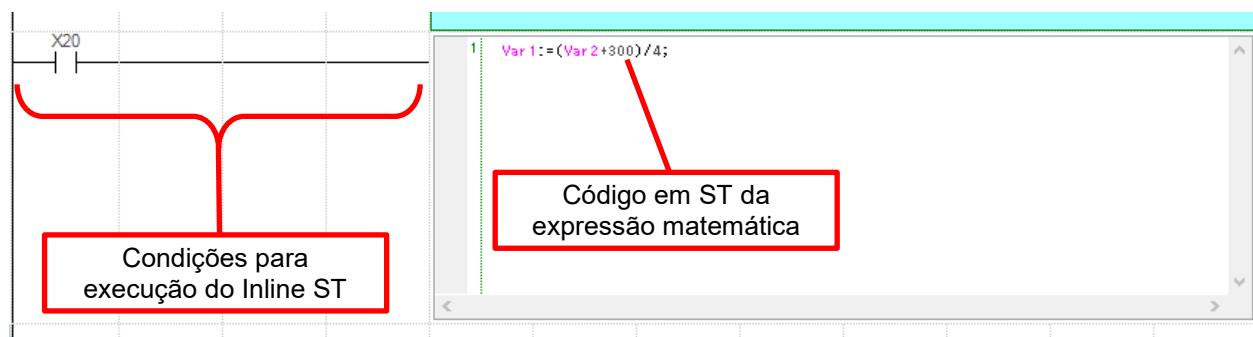
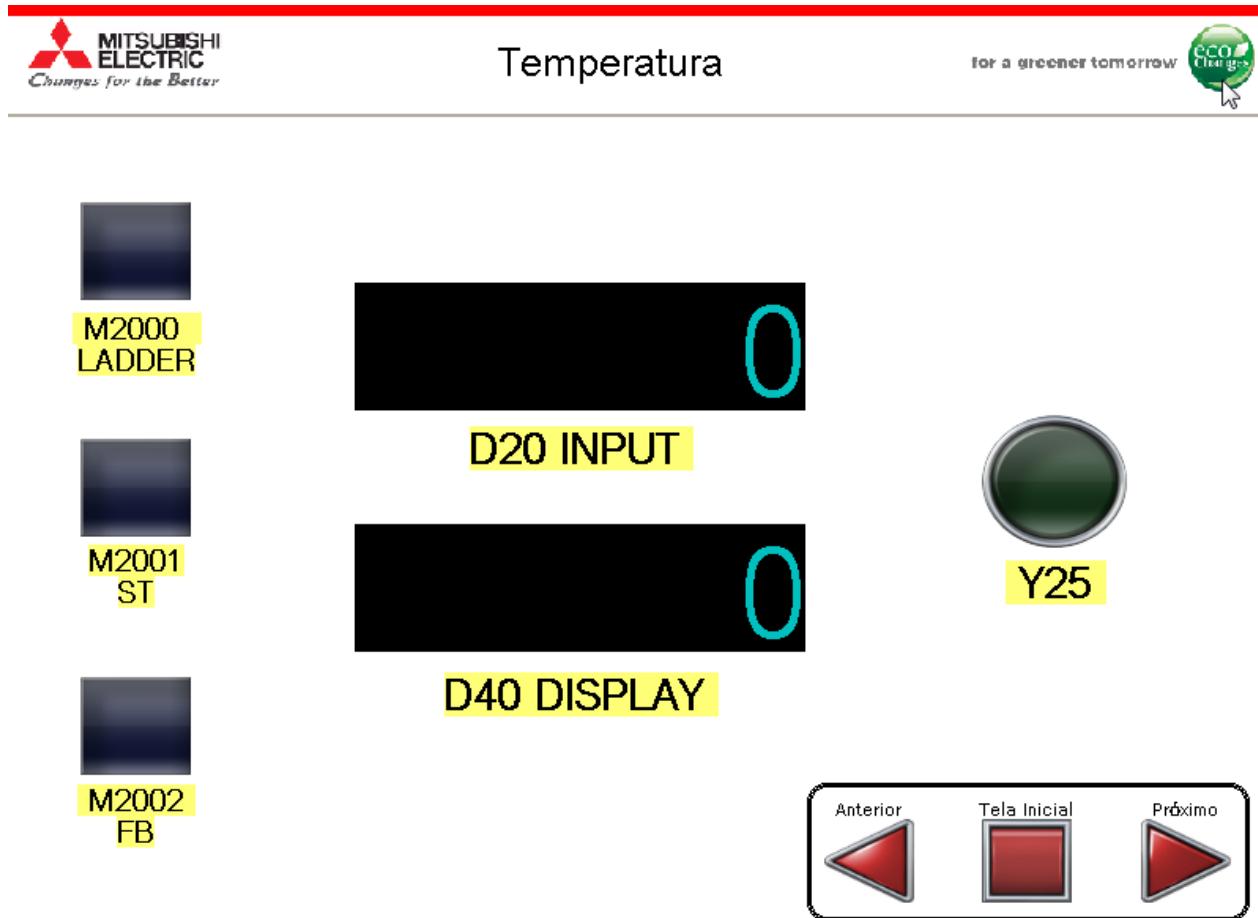


Figura 70: Exemplo de uso do recurso Inline ST.

6.9.1 Exercícios



- 1) Refaça o exercício 6.9, utilizando o M2001 (texto estruturado) em modo on.

6.10 Function Block

Há um recurso de programação no GX Works3 que permite que qualquer tipo de lógica repetitiva possa ser criada no programa sem a necessidade da construção de um programa extenso. Esse recurso denomina-se “Function Block”.

Para introduzir o Function Block, vá até a barra Navigation na árvore Project, e clique com o botão direito em FB/FUN. Selecione ADD NEW DATA. Na janela que abrirá, selecione o data type, o data name, program language e mude o "Use EN/ENO" para Yes. Abrirá uma tela onde será desenvolvida a lógica que criará o Function Block.

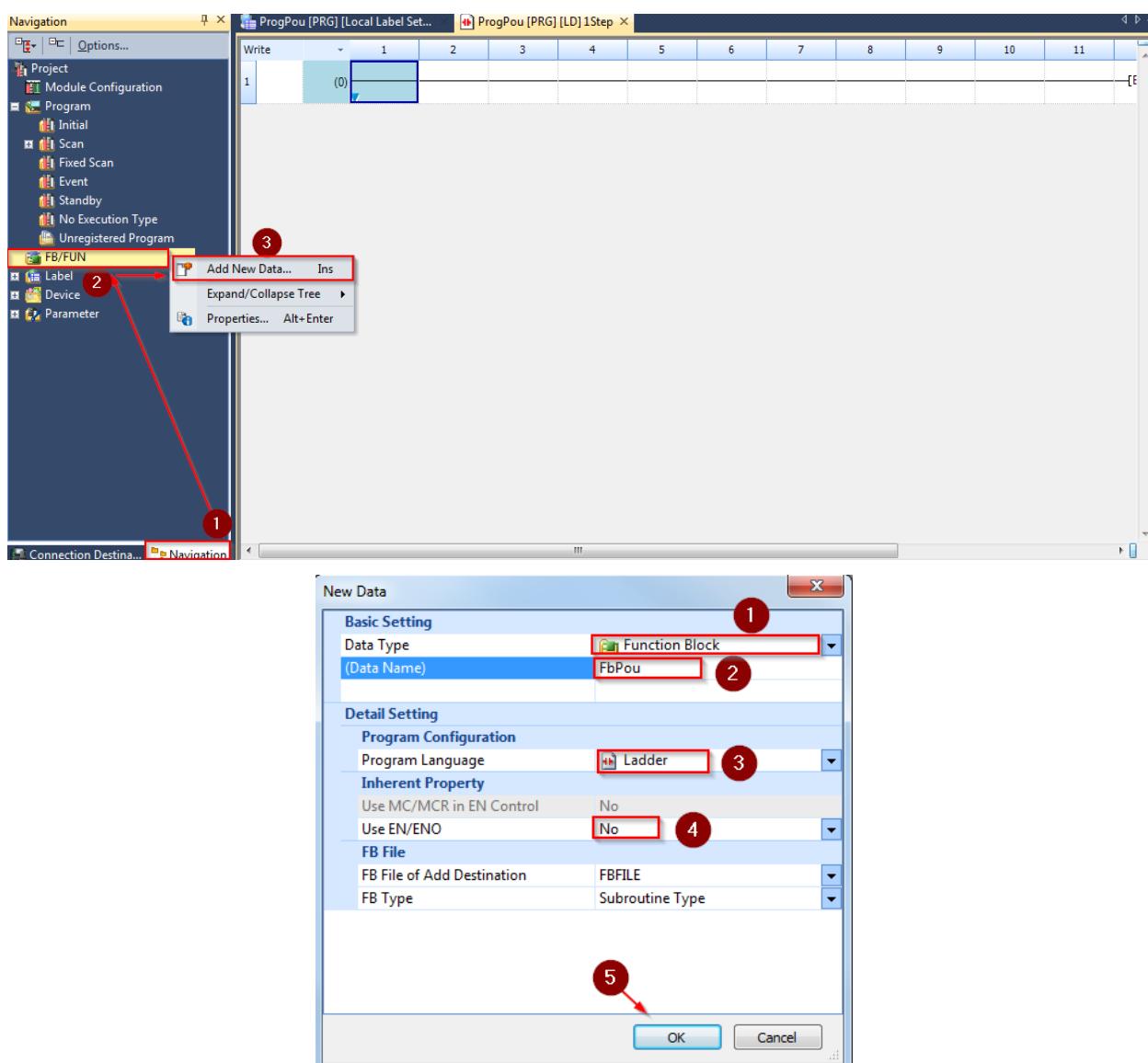
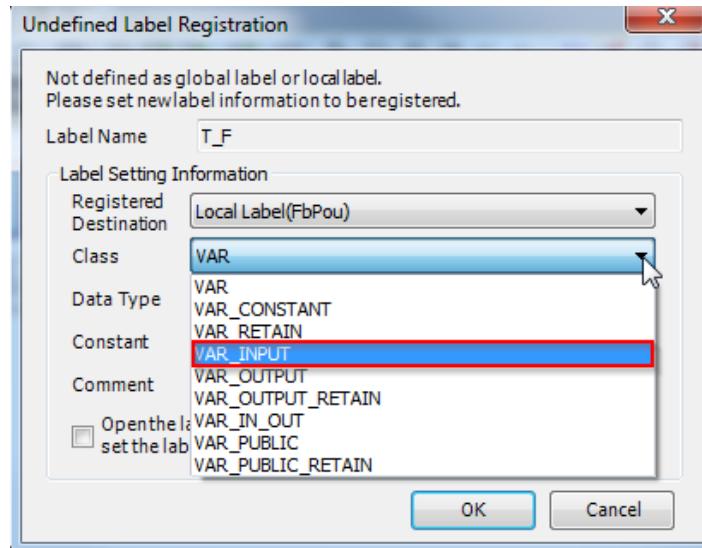


Figura 78: Passos criação do Function Block

No processo de criação da lógica, será necessário a utilização de labels . Ao inserir uma nova label, uma nova janela abrirá e será necessário selecionar qual o tipo de label desejada.



Alguns tipos de variáveis disponíveis:

- VAR é usado para variável que existe apenas dentro da função.
- VAR_CONSTANT é usado para um valor numérico fixo, constante.
- VAR_INPUT é usado como uma entrada de dados
- VAR_OUTPUT é usado como saída de dados.

Após a criação do Function Block, vá até a árvore de projeto e clique em FB/FUN. Abaixo do FBFILe aparecerá o Function Block criado. Clique sobre ele, e arraste até a área de edição ladder.

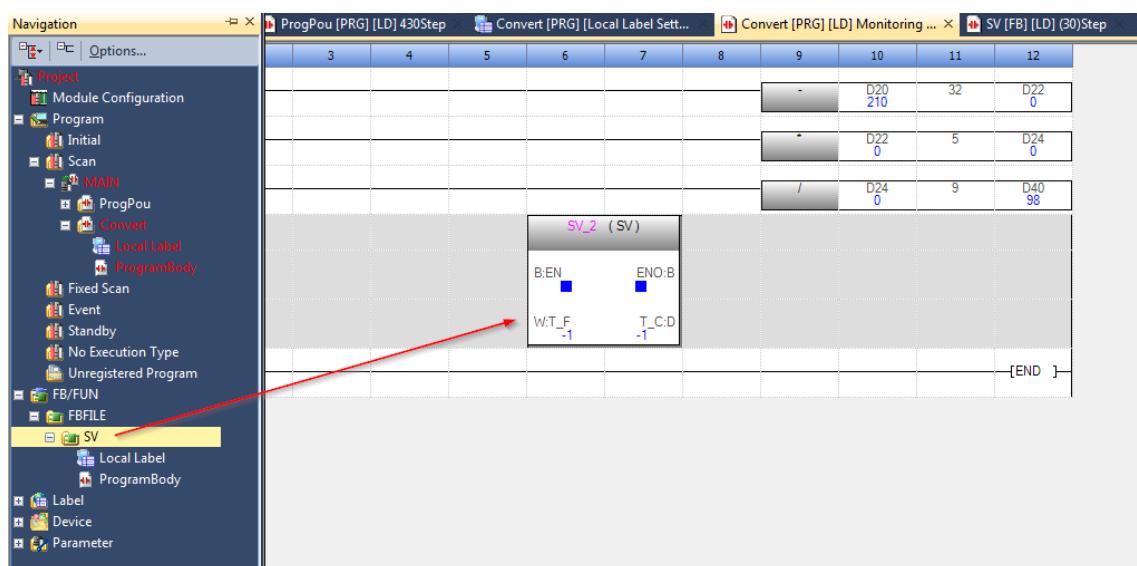


Figura 79: Colocando o FB na área de edição ladder

6.10.1 Exercícios

Temperatura

For a greener tomorrow 

 M2000 LADDER	 D20 INPUT	 Y25
 M2001 ST	 D40 DISPLAY	
 M2002 FB		 Anterior Tela Inicial Próximo

- 1) Refaça o exercício 6.9, utilizando o M2002 (texto estruturado) em modo on.

7. DIAGNÓSTICOS

7.1 Introdução

Neste capítulo iremos estudar as funções que estão relacionadas ao controle e entendimento do programa, como gerar diagnósticos de falhas que podem ocorrer na CPU.

Para isso iremos usar as ferramentas que se encontram nos menus demonstrados na Figura 71.

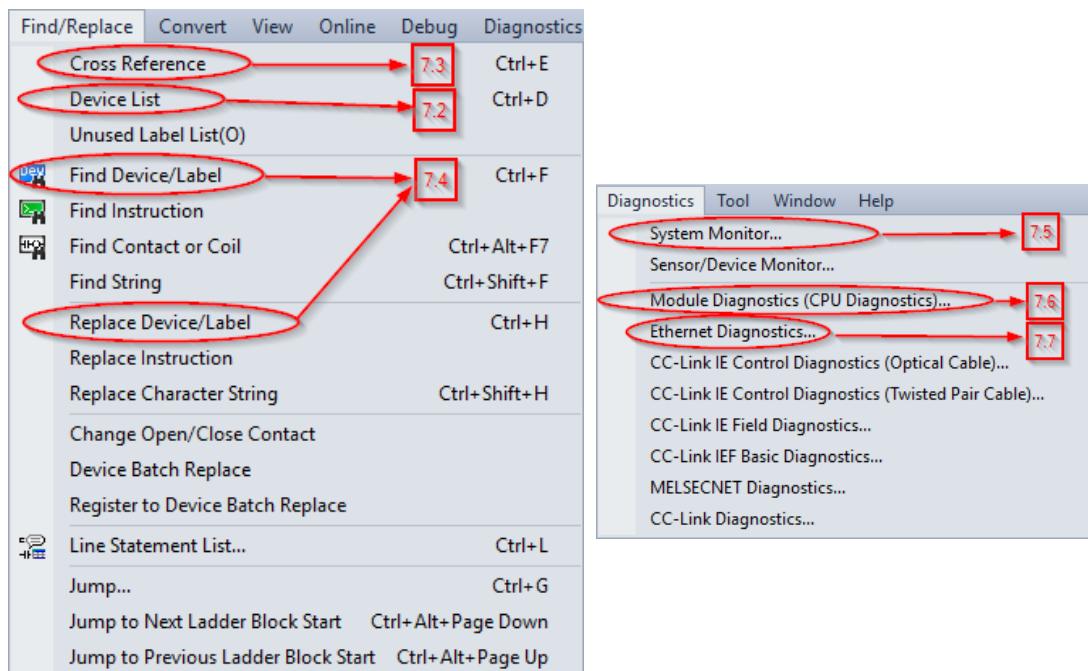
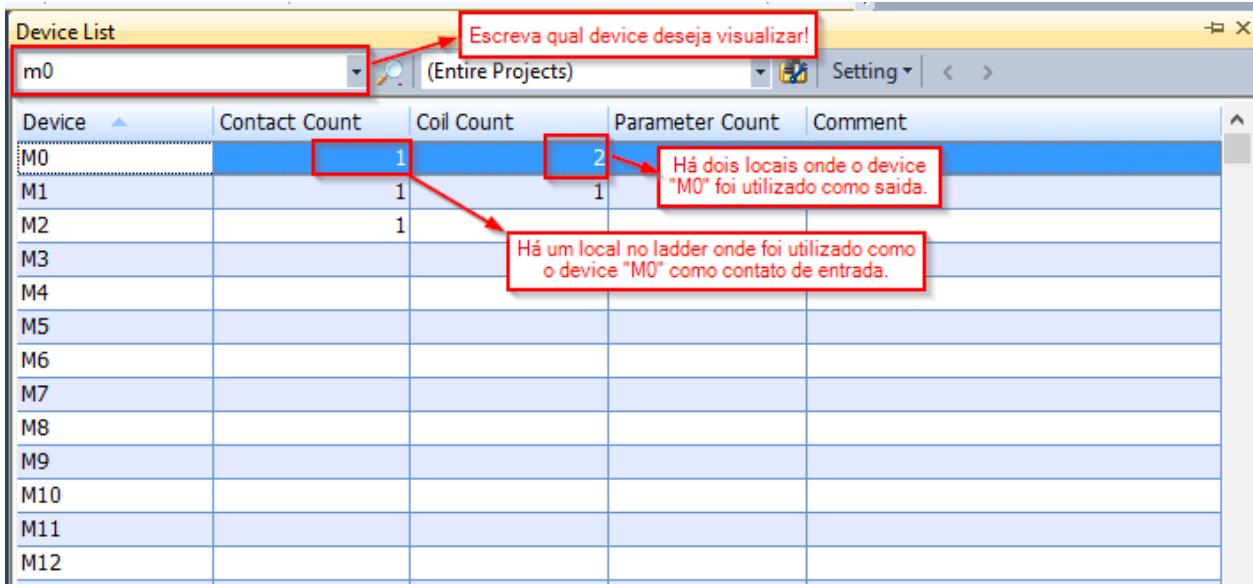


Figura 71: A esquerda Menu [Find/Replace], a direita Menu [Diagnostics]

7.2 Função [Device List]

Ao acessar a função de [Device List] descrito na Figura 71 você terá a sua disposição a tela da Figura 72, onde poderá facilmente visualizar qual device está sendo usado e como está sendo usado, sendo essas informações separadas em três colunas: Sinal de Entrada [Contact Count], Sinal de Saída [Coil Count] e Parâmetros [Parameter Count].



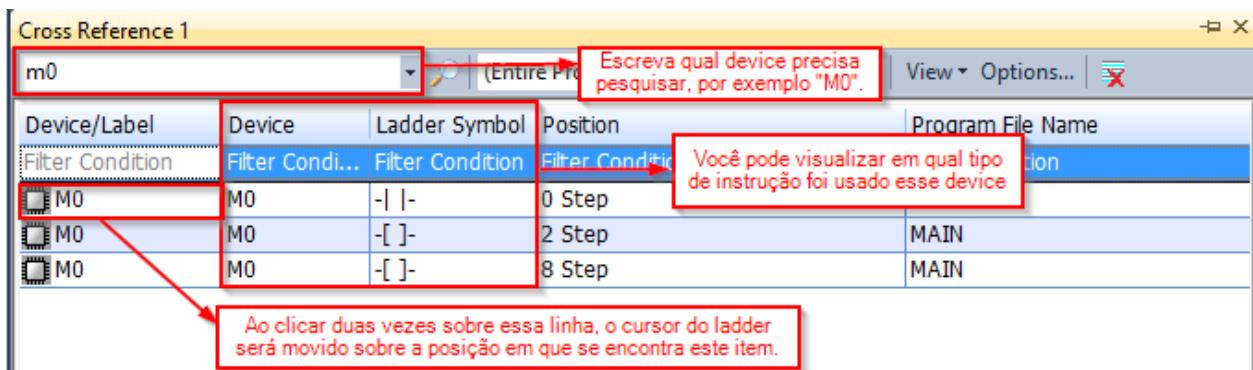
The screenshot shows a table titled "Device List" with columns: Device, Contact Count, Coil Count, Parameter Count, and Comment. A search bar at the top left contains "m0". A red box highlights the search bar with the text "Escreva qual device deseja visualizar!". The table rows are M0, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, and M12. The "Contact Count" column for M0 has a red box around the value "1" with the annotation "Há um local no ladder onde foi utilizado como o device 'M0' como contato de entrada.". The "Parameter Count" column for M0 has a red box around the value "2" with the annotation "Há dois locais onde o device 'M0' foi utilizado como saída."

Device List				
Device	Contact Count	Coil Count	Parameter Count	Comment
M0	1		2	Há dois locais onde o device "M0" foi utilizado como saída.
M1	1		1	
M2	1			
M3				
M4				
M5				
M6				
M7				
M8				
M9				
M10				
M11				
M12				

Figura 72: Janela da função [Device List]

7.3 Função [Cross Reference]

Dentro da função [Cross Reference], também conhecida como referência cruzada, é possível visualizar todos os locais onde o device alvo foi utilizado dentro do Projeto.



The screenshot shows a table titled "Cross Reference 1" with columns: Device/Label, Device, Ladder Symbol, Position, and Program File Name. A search bar at the top left contains "m0". A red box highlights the search bar with the text "Escreva qual device precisa pesquisar, por exemplo 'M0'.". The table rows are M0, M0, and M0. The "Position" column for the first row has a red box around "0 Step" with the annotation "Você pode visualizar em qual tipo de instrução foi usado esse device". A red arrow points from the bottom right annotation to the "Position" column of the first row. Another red box highlights the "Position" column of the first row with the annotation "Ao clicar duas vezes sobre essa linha, o cursor do ladder será movido sobre a posição em que se encontra este item.".

Cross Reference 1				
Device/Label	Device	Ladder Symbol	Position	Program File Name
Filter Condition	Filter Cond...	Filter Condition	Filter Condition	MAIN
<input checked="" type="checkbox"/> M0	M0	- -	0 Step	
<input checked="" type="checkbox"/> M0	M0	-[]-	2 Step	MAIN
<input checked="" type="checkbox"/> M0	M0	-[]-	8 Step	MAIN

Figura 73: Janela da função [Cross Reference]

7.4 Função [Find/Replace Device/Label]

Com a função [Find/Replace] é possível realizar a pesquisa de device's rapidamente assim como a substituição destes por outros sem ter que olhar a lógica de ladder por completo.

Na Figura 74 está a janela desta função com um breve descriptivo de como realizar a operação.

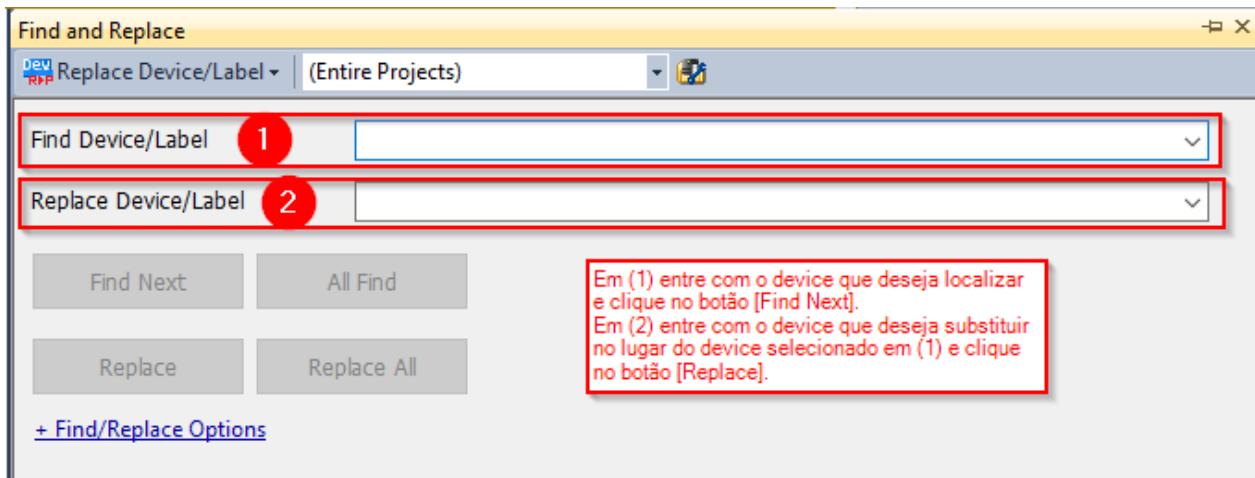


Figura 74: Janela da função [Find/Replace Device/Label]

7.5 Função [System Monitor...] (Exclusivo iQ-R)

Através da função [System Monitor...] é possível monitorar todos os módulos da CPU iQ-R e facilmente identificar quais módulos estão com erro. Ao clicar sobre o modulo com erro, uma tela com os diagnósticos se abre informando qual erro e possível solução.

A Figura 75 mostra a tela com os recursos de monitoramento.

Obs.: Recurso disponível apenas para projetos utilizando CLP da série iQ-R.

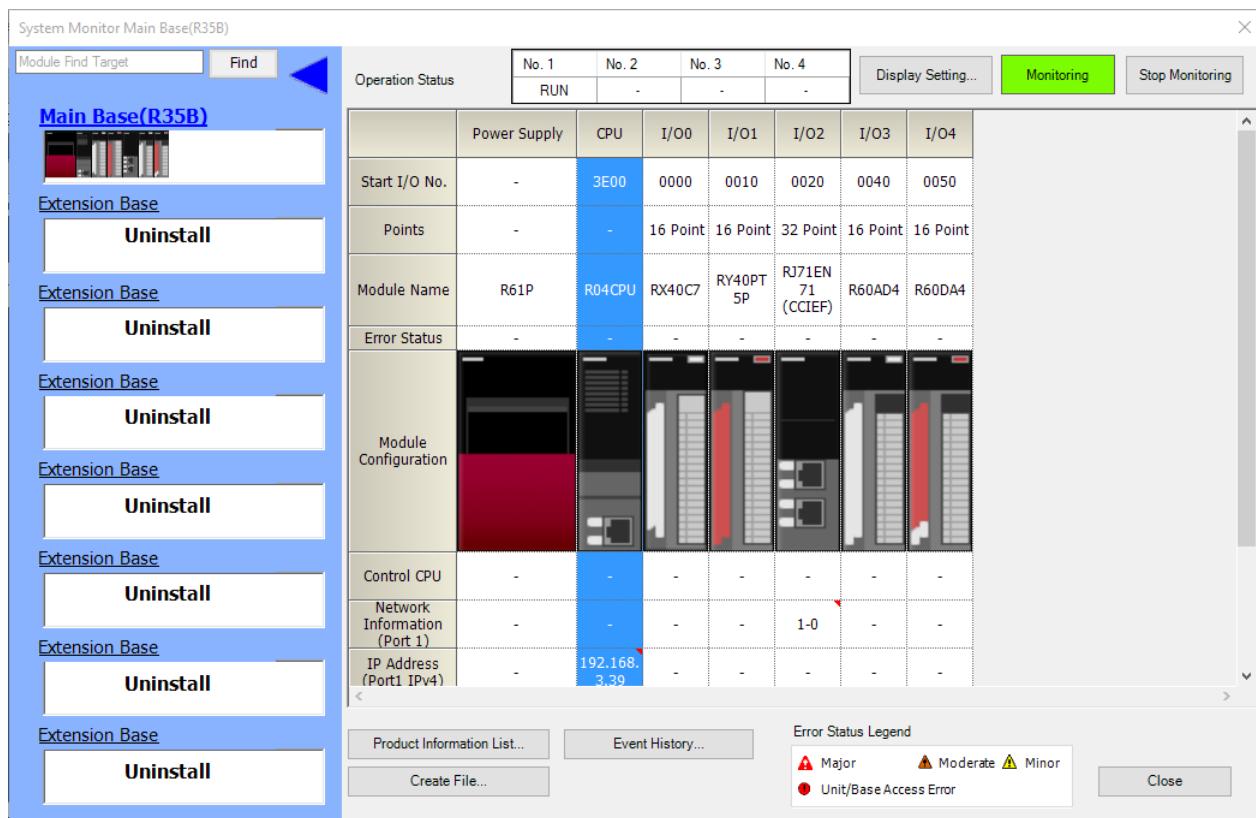


Figura 75: Janela da função [System Monitor]

7.6 Função [Module Diagnostics (CPU Diagnostics)...]

Essa função serve para verificarmos qual firmware está instalado dentro da CPU, seu Status, e caso houver algum tipo de erro, pode ser utilizado para localizar onde e como solucionar o erro desta CPU.

A seguir, na Figura 76 estão as janelas do [Module Diagnostics (CPU Diagnostics)...] para cada uma das séries de CPU do software GX Works3.

Note que são semelhantes quanto ao funcionamento.

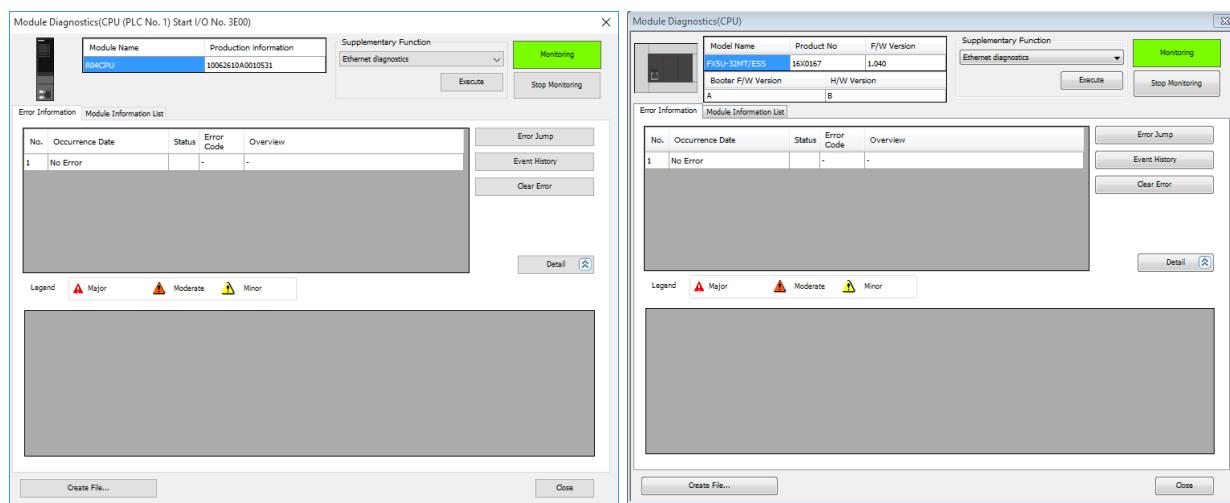


Figura 76: Janela do Module Diagnostics para iQ-R(esquerda) e iQ-F(direita)

7.6.1 Erros e alarmes

Na Figura 77 está o detalhe da janela referente aos erros e alarmes que podem ocorrer na CPU. Se houver algum erro seu código e histórico ficara registrado aqui. Ao selecionar uma linha de erro, ao clicar no botão [Error Jump] o cursor de ladder será enviado sobre a posição que houve o erro (caso o erro seja de ladder).

No botão [Detail] você encontrará informações sobre o erro, como possíveis causas e soluções.



Figura 77: Detalhe de erros da Janela do Module Diagnostics (CPU Diagnostics)

7.6.2 Firmware

Com os PLC da Mitsubishi é possivel visualizar o firmware através do detalhe demonstrado na Figura 78, no campo [F/W Version]. Dependendo da versão de

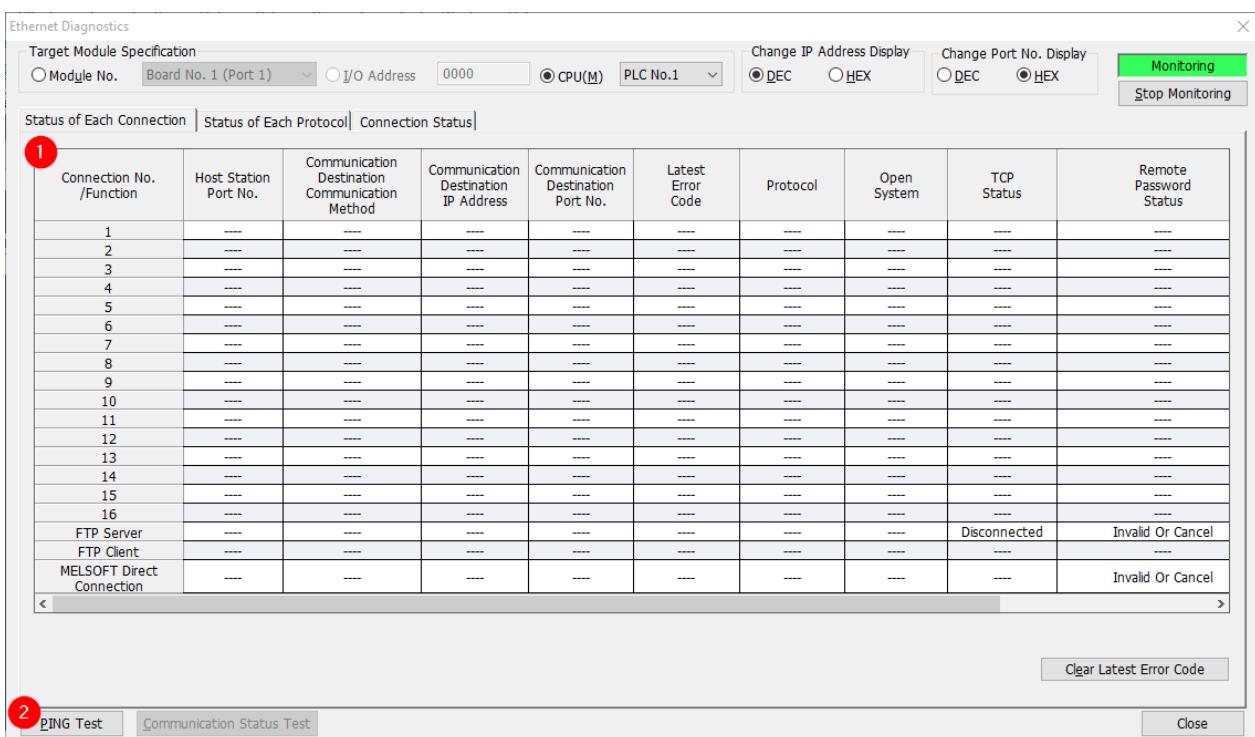
Firmware, algumas funções da sua CPU podem estar desabilitadas, sendo necessário fazer o upgrade da versão de Firmware, explicado no manual de cada PLC.

Model Name	Product No	F/W Version
FX5U-32MT/ESS	16X0167	1.040
Booter F/W Version	H/W Version	
A	B	

Figura 78: Detalhe de Firmware da Janela do Module Diagnostics (CPU Diagnostics)

7.7 Função [Ethernet Diagnostics...]

Quando se utiliza a comunicação Ethernet das CPU's, eventualmente pode se encontrar dificuldades de estabelecer uma conexão entre dois equipamentos. Para isso, no lado da CPU série iQ-F ou iQ-R possuem o recurso de diagnosticar o Status de cada conexão criada na porta Ethernet através da janela na Figura 79.



The screenshot shows the 'Ethernet Diagnostics' window with the following details:

- Target Module Specification:**
 - Module No.: Board No. 1 (Port 1)
 - I/O Address: 0000
 - CPU(M): Selected
 - PLC No.1: Selected
- Status of Each Connection:** This tab is selected, showing a table with 17 rows. The first row is highlighted with a red circle labeled '1'. The last row (FTP Server) is also highlighted with a red circle labeled '2'.

Connection No. /Function	Host Station Port No.	Communication Destination Communication Method	Communication Destination IP Address	Communication Destination Port No.	Latest Error Code	Protocol	Open System	TCP Status	Remote Password Status
1	---	---	---	---	---	---	---	---	---
2	---	---	---	---	---	---	---	---	---
3	---	---	---	---	---	---	---	---	---
4	---	---	---	---	---	---	---	---	---
5	---	---	---	---	---	---	---	---	---
6	---	---	---	---	---	---	---	---	---
7	---	---	---	---	---	---	---	---	---
8	---	---	---	---	---	---	---	---	---
9	---	---	---	---	---	---	---	---	---
10	---	---	---	---	---	---	---	---	---
11	---	---	---	---	---	---	---	---	---
12	---	---	---	---	---	---	---	---	---
13	---	---	---	---	---	---	---	---	---
14	---	---	---	---	---	---	---	---	---
15	---	---	---	---	---	---	---	---	---
16	---	---	---	---	---	---	---	---	---
FTP Server	---	---	---	---	---	---	---	Disconnected	Invalid Or Cancel
FTP Client	---	---	---	---	---	---	---	---	---
MELSOFT Direct Connection	---	---	---	---	---	---	---	---	Invalid Or Cancel
- Buttons:**
 - Clear Latest Error Code
 - PING Test
 - Communication Status Test
 - Close

Figura 79: Janela da Função [Ethernet Diagnostics]

No item número (1) da Figura 79 é possível visualizar todas as conexões que a IHM pode realizar e seus respectivos status.

No item número (2) é possível enviar um comando de PING através da porta Ethernet da CPU, para um IP que você pode configurar, podendo verificar assim se a tal IP pode ser acessada pelo PLC através dessa conexão, como demonstra a Figura 80. Caso haja algum erro pode ser considerado cabeamento ou erro de configuração entre as portas ethernet.

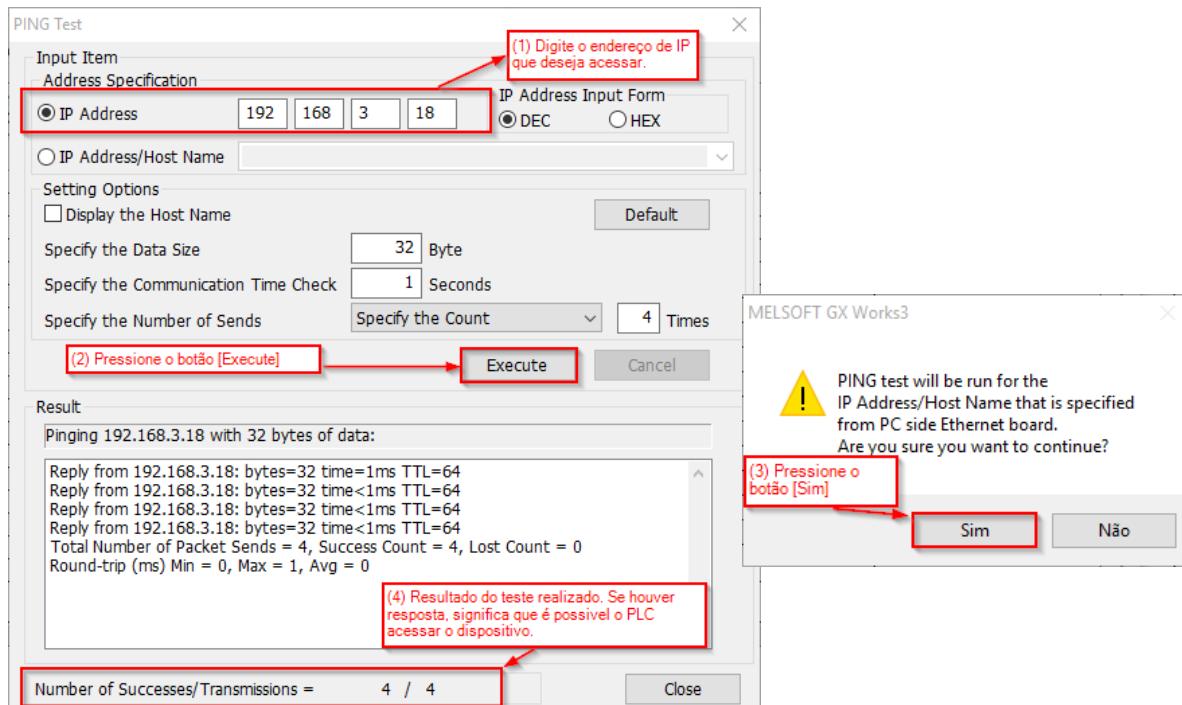


Figura 80: Janela da função [PING Test]

7.8 Exercícios

Foram feitas algumas modificações num processo e o programador cometeu alguns erros. O programa abaixo deve realizar a seguinte operação, assim que ligar a máquina:

- a-) Realizar o cálculo informado nos comentários e armazenar o resultado em D320;
- b-) Ligar e desligar Y2E de acordo com o clock imposto pelos temporizadores;
- c-) Ligar Y2F assim que o cálculo atingir o valor 2 e parar Y2E

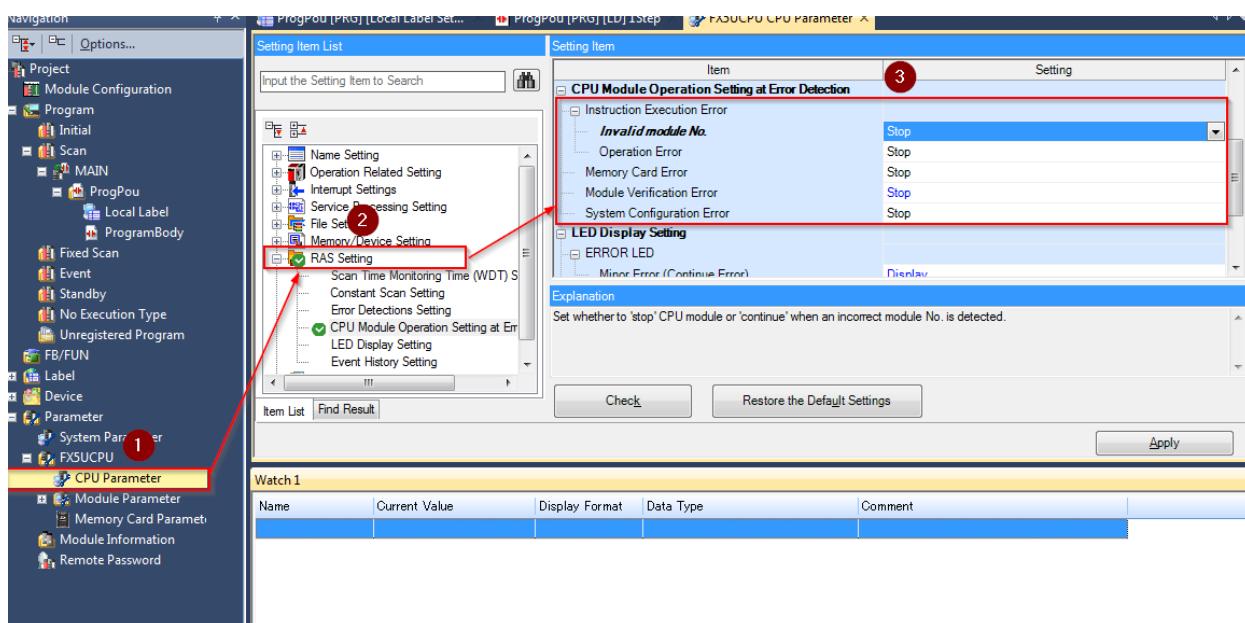
→ O que não funciona:

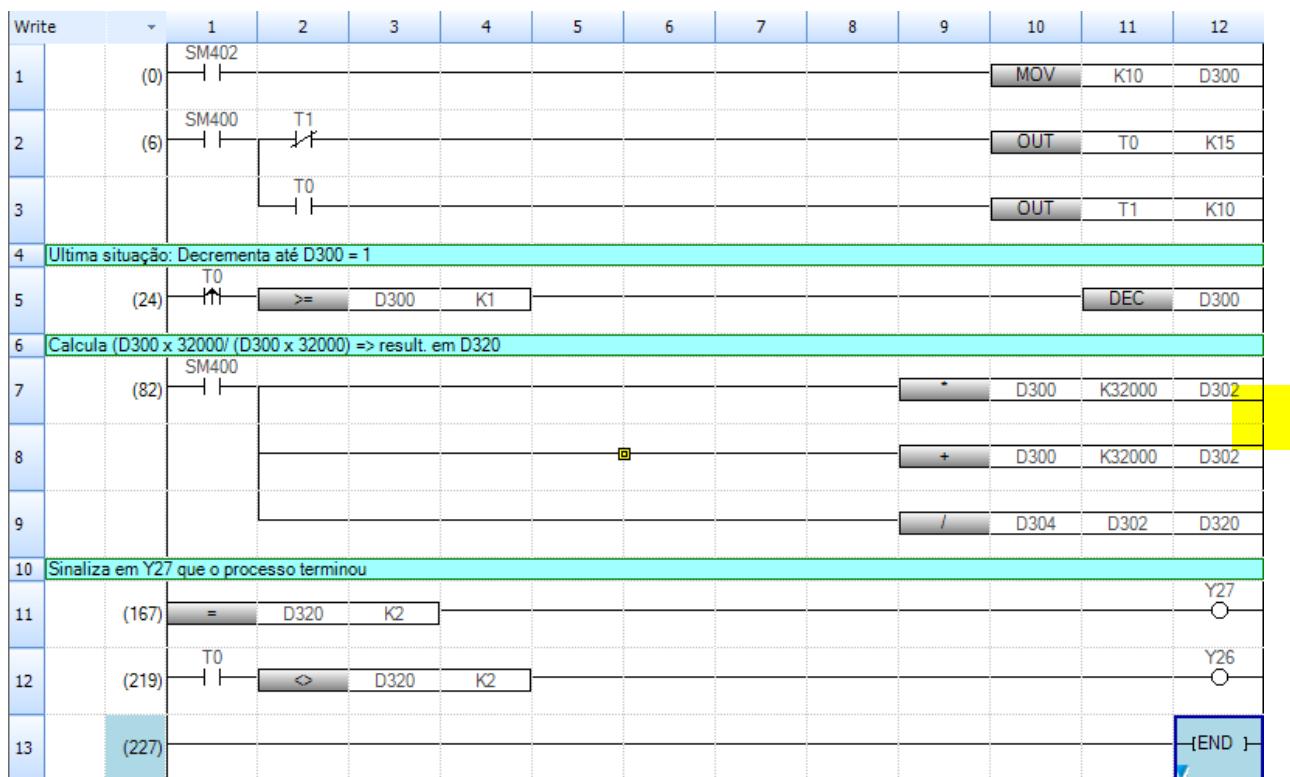
O PLC está entrando em falha no final do processo;

O PLC não sinaliza fim de processo em Y2F apesar do processo ter terminado;

Tente descobrir onde estão as falhas e sugira como consertá-las! Repare que os comentários são relativos ao que deveria ocorrer no programa.

- A seguir, o programa e os parâmetros:





*Dica: Descarregue o programa na CPU e veja o que acontece!

Anexo A - Importação de projetos do GX Works2

É possível importar projetos realizados no GX Works2 para o GX Works3. Dependendo do tipo de CPU utilizada no projeto original, o GX Works3 converterá o programa para os tipos de CPU, conforme tabela abaixo. Importante lembrar que, caso o projeto do GX Works2 tenha sido configurado com CLP diferente da tabela abaixo, não irá funcionar a conversão.

Module type before change	Module type after change
Universal model QCPU/Universal model High-speed Type QCPU	R120CPU
FXCPU(FX3U/FX3UC)	FX5UCPU

Para importar projetos do GX Works2, acesse, no meu suspenso, o caminho [Project] → [Open Other Format File] → [Open GX Works2 Format Project]. Uma janela para selecionar o arquivo do projeto do GX Works2, como a ilustrada abaixo.

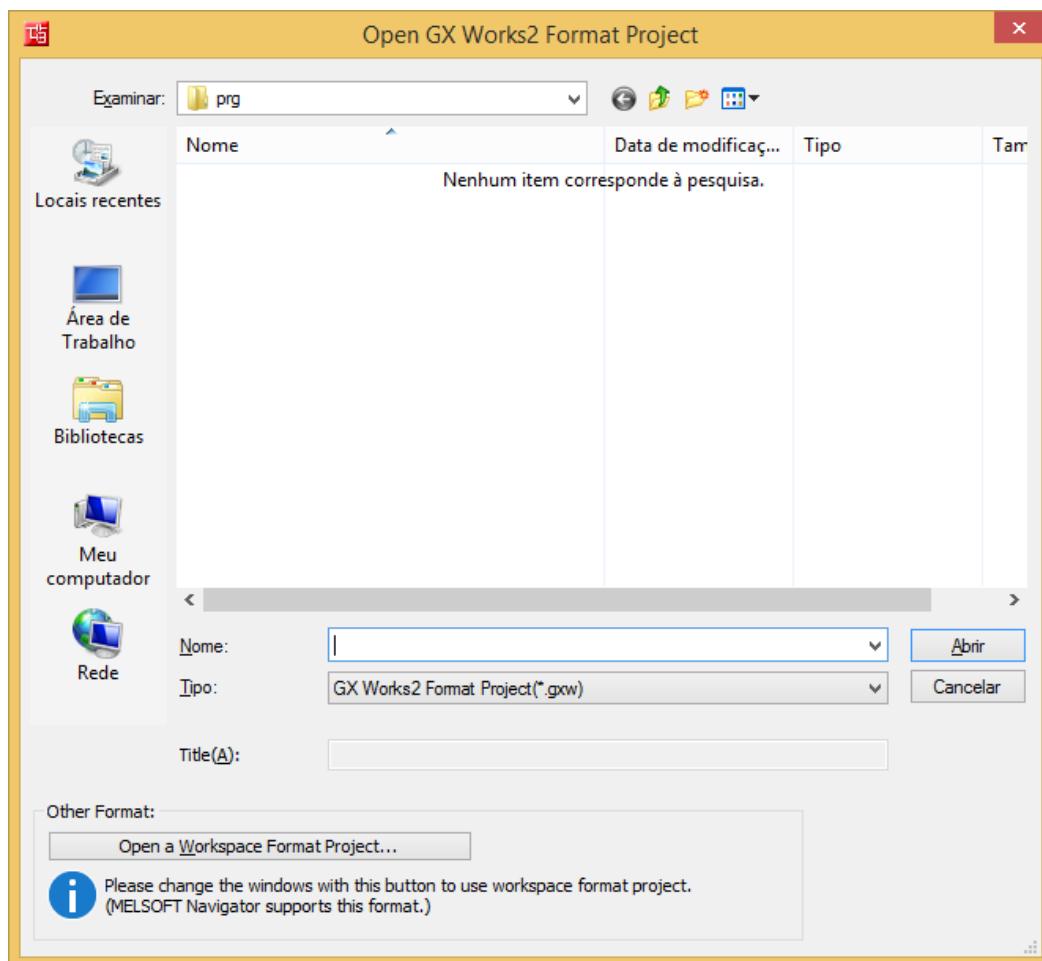


Figura 81: Janela de importação de projetos do GX Works2.

É importante notar que alguns itens, após a importação do projeto, precisam ser verificados, conforme a tabela abaixo.

Operation at changing module type	Setting items of GX Works2		Remarks
Changed in accordance with the target module type	<ul style="list-style-type: none"> • PLC Parameter <ul style="list-style-type: none"> PLC Name PLC System^{*1,*2} PLC File Setting PLC RAS^{*1} Boot File Program^{*1} SFC^{*3} Device^{*1,*2} I/O Assignment^{*4} Multiple CPU Setting^{*1,*4} Built-in Ethernet Port Setting • Network Parameter <ul style="list-style-type: none"> CC-Link IE setting Ethernet setting CC-Link setting^{*2} • Intelligent function module parameter <ul style="list-style-type: none"> QD75 Positioning module^{*2,*4} • FB/FUN <ul style="list-style-type: none"> Structured Ladder/FBD program^{*5} SFC program^{*1,*3} Device comment for devices other than SM/SD^{*6} 		Check the settings after the change.
Returns to default/ Data are deleted	<ul style="list-style-type: none"> Options (other than "Device Comment Reference/Reflection Target") System label Device comments of SM/SD^{*6} Connection destination Remote password 		Set with GX Works3.
Deleted	<ul style="list-style-type: none"> • PLC Parameter <ul style="list-style-type: none"> Communication Head Setting Built-in I/O Function Setting Serial Communication Built-in Serial Setting Adapter Serial Setting Memory Capacity Special Function Block Positioning Operation Setting • Redundant Parameters <ul style="list-style-type: none"> Operation mode setting Tracking setting • Network Parameter <ul style="list-style-type: none"> MELSECNET/10 and MELSECNET/H setting MELSECNET/H Remote I/O setting • Programs of user library that are not registered to the program setting Task settings Device/Label Automatic-Assign Setting Project Revision Device memory (Only when the module type is changed to FX5UCPU) 		Not supported by GX Works3.

*1 Check/set with GX Works3 after the change since some items are returned to the default or their data are deleted.

*2 The data are deleted when the module type is changed to FX5UCPU.

*3 To read SFC programs, GX Works2 of which version is 1.53SH or later requires to be installed in a personal computer.

*4 The items which are not supported by GX Works3 are deleted.

*5 To read Structured Ladder/FBD programs, GX Works2 of which version is 1.519R or later requires to be installed in a personal computer.

*6 The data are moved to SM8000 or higher/SD8000 or higher when the module type is changed to FX5UCPU.