



PLC

GX Works2 Avançado

Este curso fornece conhecimento das funções para a melhoria do ambiente de desenvolvimento do local de design que apresentam problemas relativos à "produtividade", "qualidade", "gerenciamento do projeto" e "medidas de segurança". O curso destina-se a programadores de sequência que já usam o MELSOFT GX Works2.

Introdução **Objetivo do curso**

Este curso fornece conhecimento das funções para a melhoria do ambiente de desenvolvimento do local de design que apresentam problemas relativos à "produtividade", "qualidade", "gerenciamento do projeto" e "medidas de segurança". O curso destina-se a programadores de sequência que já usam o MELSOFT GX Works2.

Introdução Estrutura do curso

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

Capítulo 1 - Melhoria do ambiente de desenvolvimento usando o GX Works2

Você aprenderá sobre os problemas enfrentados no local de design e o ambiente de desenvolvimento necessário para solucioná-los.

Capítulo 2 - Programação

Você aprenderá as funções usadas para programação.

Capítulo 3 - Depuração

Você aprenderá as funções usadas para depuração.

Capítulo 4 - Gerenciamento do projeto e medidas de segurança

Você aprenderá sobre as funções para gerenciamento do projeto e medidas de segurança no estágio de desenvolvimento e manutenção após o início da operação do sistema.

Teste final

Grau de aprovação: 60% ou superior

Introdução Como utilizar esta ferramenta de e-Learning

Ir para a próxima página		Ir para a próxima página.
Voltar para a página anterior		Voltar para a página anterior.
Mover-se para a página desejada		O "Índice" será exibido, permitindo que você navegue até a página desejada.
Sair do curso		Sair do curso. A janela, como a tela de "Conteúdo", e o curso serão fechados.

Introdução **Precauções para utilização**

Precauções de segurança

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

Precauções neste curso

- As telas exibidas da versão de software que você utiliza podem ser diferentes das apresentadas neste curso.

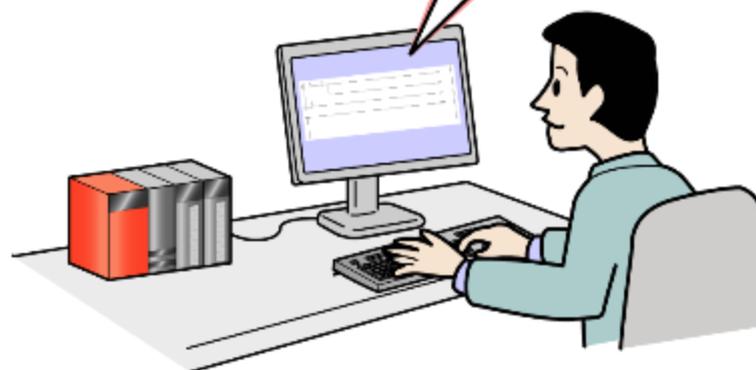
Capítulo 1 Melhoria do ambiente de desenvolvimento usando o GX Works2

Etapas de aprendizagem do Capítulo 1

Este curso destina-se a programadores que trabalham no desenvolvimento de programas de sequência. Você aprenderá como usar as funções excelentes do **MELSOFT GX Works2** para solucionar problemas do local de design relacionados à **"produtividade"**, **"qualidade"**, **"gerenciamento do projeto"** e **"segurança"**.

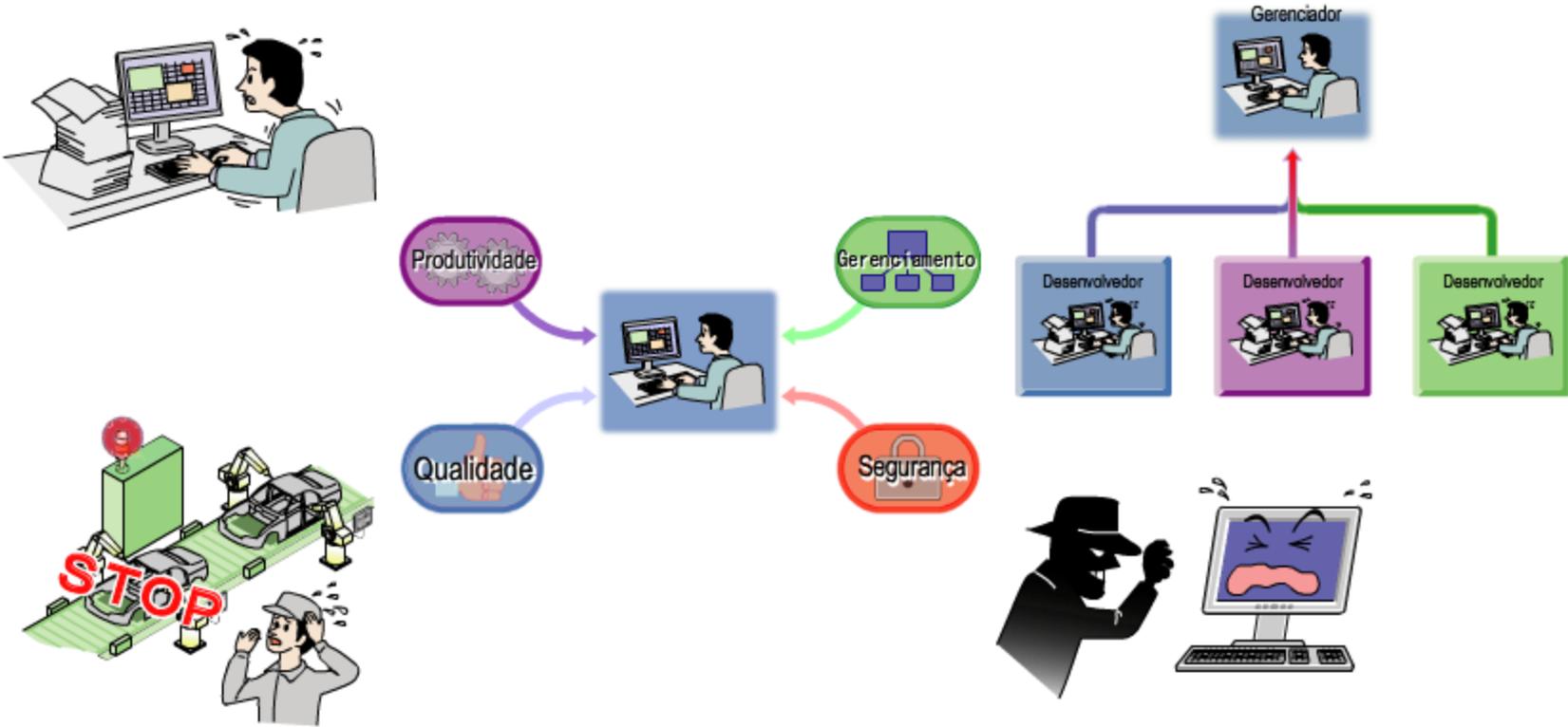
No Capítulo 1, você aprenderá sobre os problemas enfrentados no local de design e o ambiente de desenvolvimento necessário para solucioná-los.

- 1.1 Problemas enfrentados pelo local de design
 - 1.1.1 Melhoria da produtividade
 - 1.1.2 Melhoria da qualidade
 - 1.1.3 Gerenciamento do projeto
 - 1.1.4 Medidas de segurança
- 1.2 Procedimento de aprendizagem



1.1 Problemas enfrentados pelo local de design

Ao mesmo tempo que o local de design é necessário para a melhoria da produtividade e redução dos custos de desenvolvimento, também deve garantir que os programas desenvolvidos sejam de alta qualidade. O local de design também deve realizar o gerenciamento do projeto que permite o desenvolvimento da equipe e garante a rápida recuperação no caso de problemas. Também são necessárias medidas de segurança porque os programas de seqüência incluem dados e conhecimento importantes.



1.1.1 Melhoria da produtividade

O local de design é necessário para desenvolver programas de sequência que lidem com um processamento complexo de ampla escala.

No entanto, o custo do desenvolvimento é maior conforme aumenta a duração do período de desenvolvimento.

Assim, você é solicitado a melhorar a produtividade do desenvolvimento do programa a fim de reduzir os custos de desenvolvimento.

O GX Works2 fornece as seguintes funções para solucionar esse problema:

- Label
- Function block
- Valor inicial do dispositivo e memória do dispositivo*
- Texto estruturado em linha
- Importação do comentário de amostra

* A série MELSEC-F não pode configurar os valores iniciais do dispositivo.

Produtividade



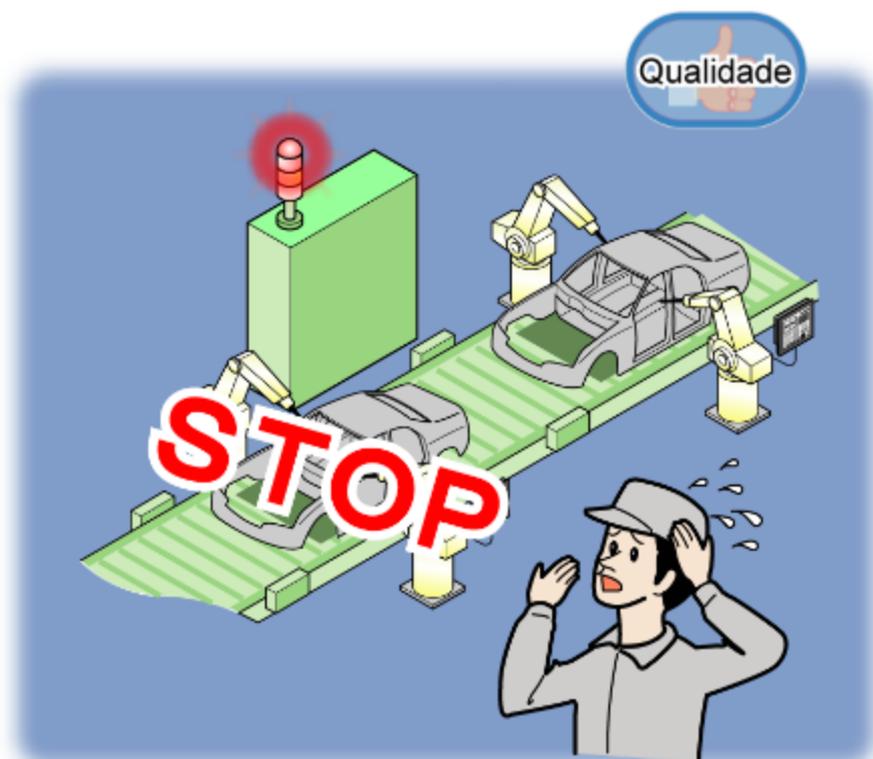
1.1.2 Melhoria da qualidade

Um erro no programa de sequências pode causar uma parada do sistema, falha ou acidente, que pode parar a produção e resultar na perda de lucros e confiabilidade.

Você é solicitado a desenvolver programas de qualidade que não apresentem erros.

O GX Works2 fornece as seguintes funções para solucionar esse problema:

- Label
- Function block
- Valor inicial do dispositivo e memória do dispositivo*
- Comentário
- Importação do comentário de amostra
- Exibir
- Referência cruzada
- Rastreamento de amostragem*
- Teste de dispositivo condicionado executável*
- Função de execução de step*
- Definição do sistema de E/S



* A série MELSEC-F não pode usar as definições de valor inicial do dispositivo, rastreamento de amostragem, teste do dispositivo condicionado executável e função de execução de step.

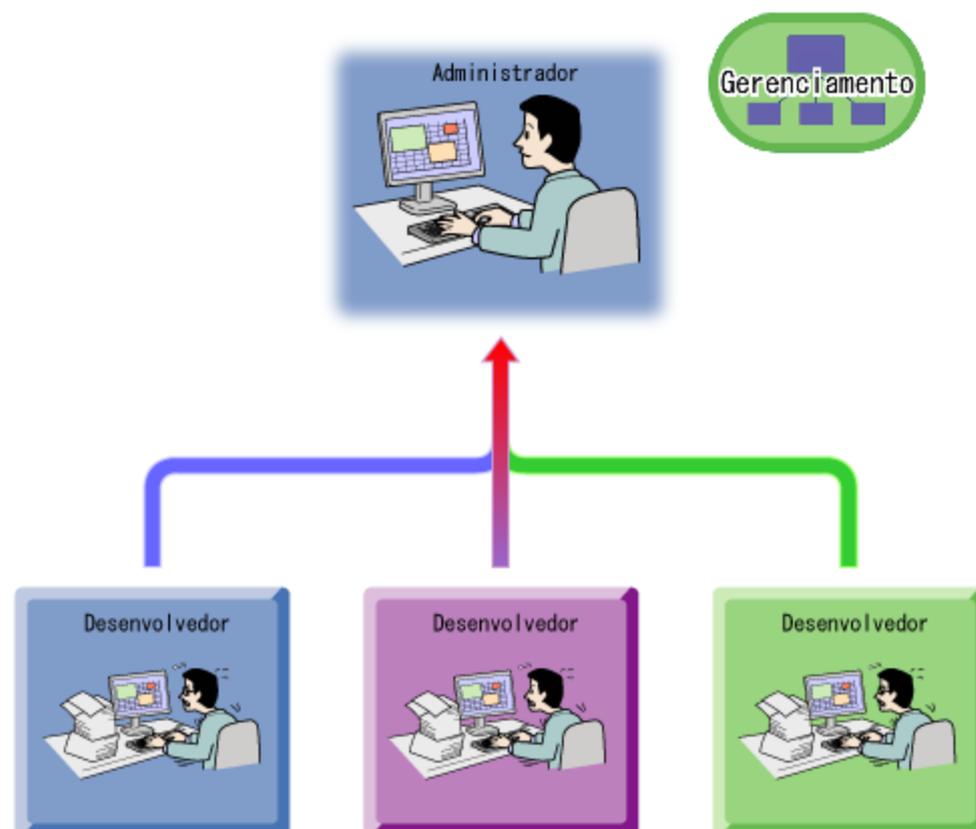
1.1.3 Gerenciamento do projeto

Um programa de seqüências de ampla escala normalmente é desenvolvido por uma equipe de programadores. Do ponto de vista da autoridade individual de gerenciamento e confidencialidade dos dados, os limites devem ser estabelecidos para a range de dados acessíveis e funções utilizáveis.

Além disso, para garantir a rápida recuperação no caso de perda do programa devido à falha no controlador programável, você é solicitado a implementar o gerenciamento da versão do programa e backup do programa periódico.

O GX Works2 fornece as seguintes funções para solucionar esses problemas:

- Segurança
- Histórico de revisão
- Verificação do PC



1.1.4 Medidas de segurança

O programa de sequências inclui dados e conhecimento estrategicamente importantes.

A **dispersão** de conhecimento e de dados do programa para o exterior poderia ter um efeito devastador nos negócios.

A **modificação não autorizada** do programa pode levar a problemas de produção como parar o sistema.

Devem ser tomadas as medidas de segurança apropriadas para prevenir contra esses problemas.

O GX Works2 fornece a seguinte função para solucionar esses problemas:

- Segurança



1.2 Procedimento de aprendizagem

Neste curso, você aprenderá sobre as funções do GX Works2 seguindo o procedimento para desenvolvimento do sistema efetivo.

(1) Programação Capítulo 2



(2) Depuração Capítulo 3



(3) Gerenciamento do projeto e medidas de segurança Capítulo 4

<Explicação de ícones>

Os ícones exibidos nas páginas dos Capítulos 2, 3 e 4 correspondem às funções respectivas de melhoria, conforme exibido a seguir.



Função usada para melhorar a produtividade do programa



Função usada para melhorar a qualidade do programa



Função usada para melhorar o gerenciamento do projeto



Função usada para medidas de segurança

Capítulo 2 Programação

Etapas de aprendizagem do Capítulo 2

No Capítulo 2, você aprenderá as funções usadas para programação.

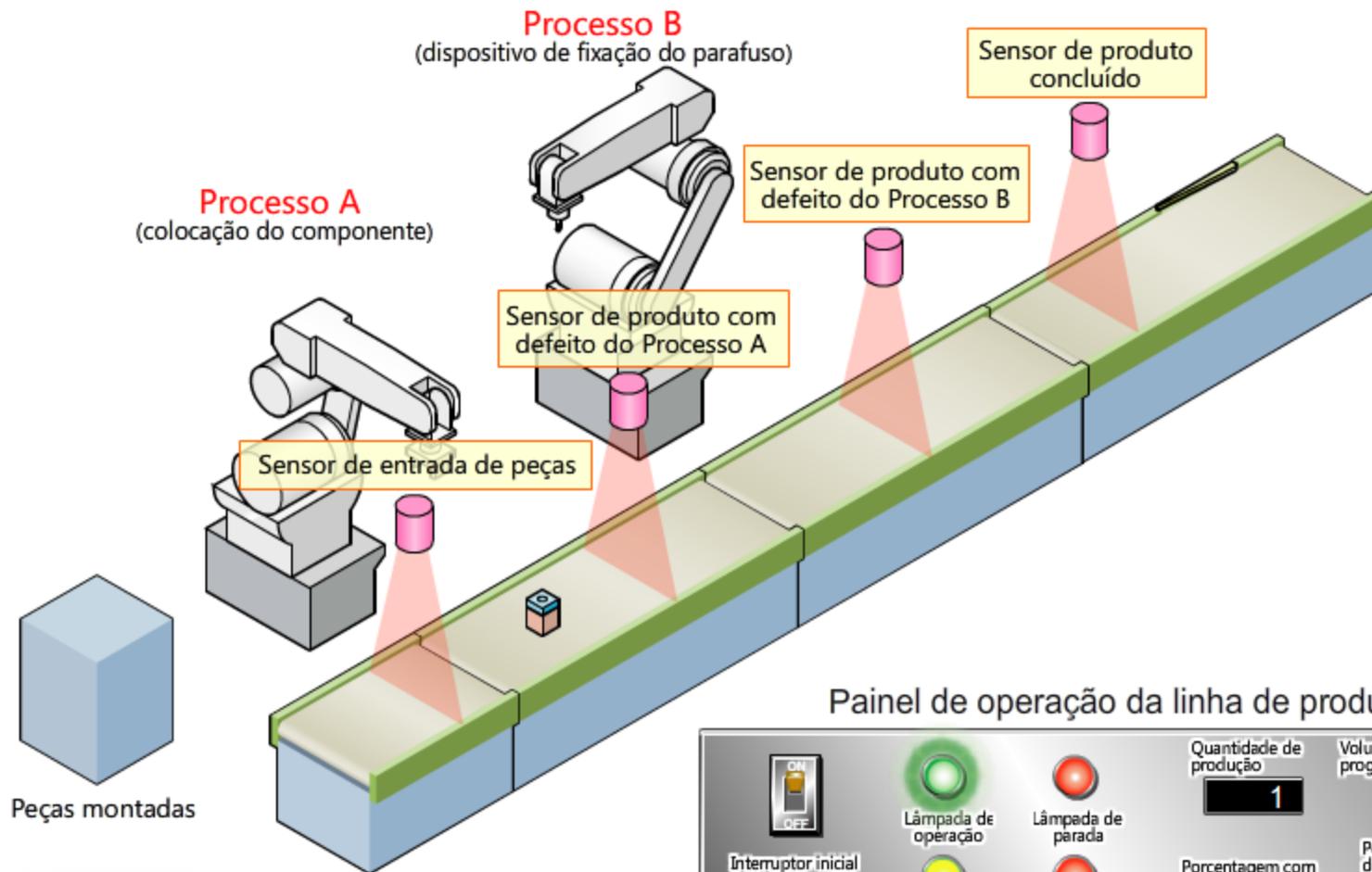
O GX Works2 fornece muitas funções diferentes para garantir uma programação eficiente e melhorar a qualidade dos programas.

- 2.1 Substituição do nome do dispositivo com um nome associado à aplicação
 - 2.1.1 Tipos de label
 - 2.1.2 Tipos de aplicações de label e valores armazenados
 - 2.1.3 Registro de label
 - 2.1.4 Atribuição automático de labels a dispositivos
- 2.2 Organização dos ladder blocks usados repetidamente como function blocks para desvio
 - 2.2.1 Criação e colocação dos function blocks
 - 2.2.2 Utilização da biblioteca de function blocks
- 2.3 Alteração dos valores iniciais do dispositivo sem correção do programa
- 2.4 Simplificação dos programas ladder
- 2.5 Criação de programas que sejam fáceis de compreender e ler
 - 2.5.1 Escrevendo um comentário para cada ladder block
 - 2.5.2 Escrevendo um comentário para cada saída (bobina, instrução de aplicação)
- 2.6 Tornando fácil a leitura de programas que contêm relés/registros especiais

Capítulo 2 Introdução do sistema usado para aprendizagem neste curso



Este curso emprega o seguinte sistema hipotético para ajudá-lo a aprender como usar as funções do GX Works2:



Ao pressionar o botão "Voltar para o topo", você volta para a primeira seleção de operação.

Voltar para o topo

Painel de operação da linha de produção

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Quantidade de produção	Volume de produção programado
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	20
Interruptor inicial	Lâmpada de operação	Lâmpada de parada	Porcentagem com defeito do Processo A	Porcentagem com defeito do Processo A permissível
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10
<input type="checkbox"/>	Quantidade de produção alcançada	Lâmpada de erro de porcentagem com defeito do Processo A	Porcentagem com defeito do Processo B	Porcentagem com defeito do Processo B permissível
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	5
Reset de quantidade de produção	Lâmpada de erro de porcentagem com defeito do Processo B			

2.1 Substituição do nome do dispositivo com um nome associado à aplicação

Produtividade

Qualidade

Um **dispositivo** usado no programa de seqüências recebe um nome que consiste de uma letra seguida de um número, por exemplo "M0" ou "D5".

Assim, o nome do dispositivo não fornecer qualquer pista quanto à aplicação do dispositivo.

Um programa de ampla escala usa muitos tipos de diferentes de dispositivos, o que significa que, durante a programação, você precisa verificar continuamente os documentos de design do sistema para determinar a aplicação dos dispositivos. Isso reduz a eficiência do trabalho e afeta de forma adversa a qualidade do programa devido aos erros ao selecionar dispositivos.

As **"labels"** podem ser usadas para solucionar esses problemas.

Em vez de usar um nome do dispositivo, você pode usar um nome (label) que indique a aplicação real, como, por exemplo, **"Sinal de início da linha de produção"**.

Para esse tipo de nome, os caracteres japoneses (hiragana, katakana, e kanji) podem ser usados, além dos caracteres alfanuméricos.

<Instrução para definir "M0", que contém o sinal de início da linha de produção, como LIGADO através da instrução de DEFINIÇÃO>

Dispositivo	—[SET M0]—
Label	—[SET Sinal de início da linha de produção]—

O uso desse tipo de label é eficaz na criação de um programa fácil de ler, aumentando a eficiência de desenvolvimento do programa e prevenindo contra erros de entrada do dispositivo.



2.1.1 Tipos de label

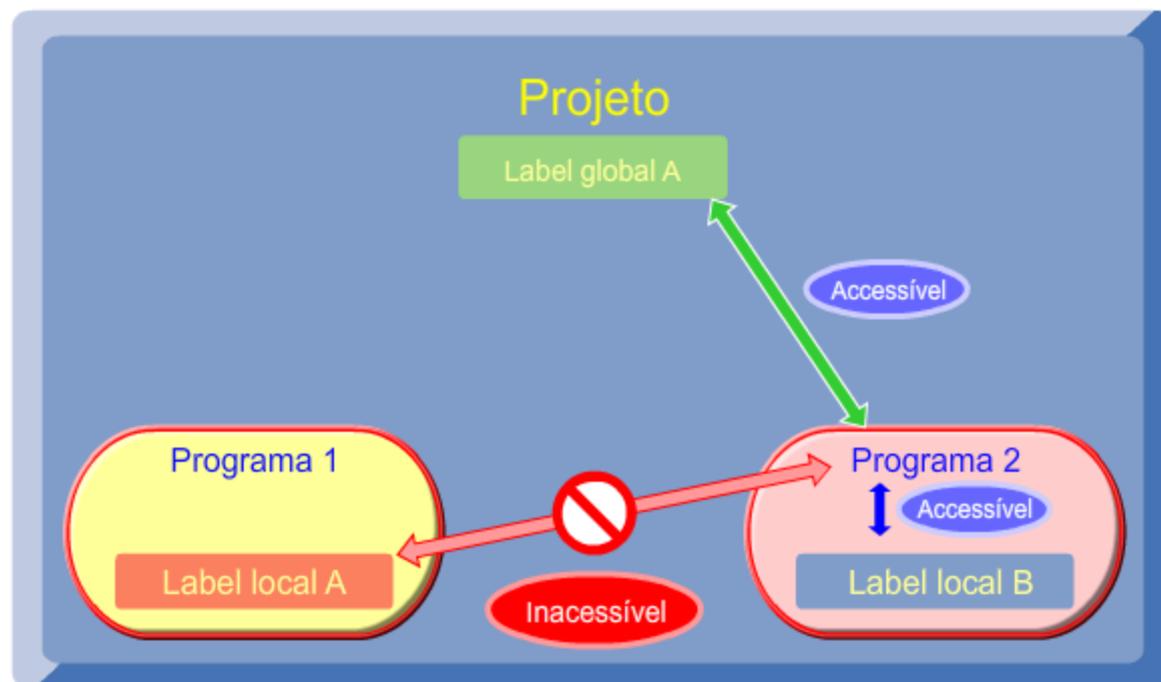
Há dois tipos de labels: "**label global**" e "**label local**".

<Label global>

As labels globais são usadas para um projeto inteiro e podem ser acessadas por qualquer programa naquele projeto.

<Label local>

As labels locais são usadas em um programa específico e somente podem ser acessadas pelo programa no qual a label está registrada.



"Program 1" (Programa 1) e "Program 2" (Programa 2) são botões. Pressione ambos os programas para ver podem acessar os dois tipos de labels.

2.1.2 Tipos de aplicações de label e valores armazenados

Produtividade

Qualidade

Ao registrar uma label, especifique a aplicação da label e o tipo de valor que pode ser armazenado usando "**Classe**" e "**Tipo de dados**".

<Classe>

A classe indica a range e aplicação de uso de uma label.

As classes que podem ser selecionadas variam de acordo com o tipo de label.

Classe	Uso pretendido	Área de configuração da label		
		Label global	Local do programa label	Function block label local
VAR_GLOBAL	A label comum pode ser usada pelos programas e pela function blocks em um projeto	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
VAR_GLOBAL_CONSTANTE	A label comum com uma constante pode ser usada pelos programas e pela function blocks em um projeto	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
VAR	A label que pode ser usada pelos programas e pela function blocks para a qual a label é pretendida	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
VAR_CONSTANT	A label com uma constante que pode ser usada pelos programas e pela function blocks para a qual a label é pretendida	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
VAR_RETAIN	Etiqueta (tipo trava) que pode ser usada por programas e blocos de função para os quais a etiqueta se destina	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
VAR_INPUT	A label usada para a entrada de uma function blocks para a qual a label é pretendida * O valor não pode ser alterado em um componente do programa.	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
VAR_OUTPUT	A label usada para a saída de uma function blocks para a qual a label é pretendida	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
VAR_IN_OUT	A etiqueta usada para a entrada e saída de um function block para o qual a etiqueta é pretendida * O valor pode ser alterado em um componente do programa.	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

2.1.2 Tipos de aplicações de label e valores armazenados

Produtividade

Qualidade

<Tipo de dados>

O tipo de dados refere-se ao tipo de valor armazenado no label.

O tipo de dados atribuído a um label indica o tipo e a range de valores que podem ser armazenados no label e no dispositivo correspondente.

Os tipos de dados que podem ser usados com os programas ladder são listados a seguir:

Tipo de dados	Descrição	Largura de bit	Range de valores armazenados
Bit	LIGADO ou DESLIGADO é armazenado. Corresponde ao dispositivo "M"	1 bit	1: LIGADO, 0: DESLIGADO
Palavra (assinada)	Um número inteiro sem frações é armazenado.	16 bits	-32768 até 32767
Palavra dupla (assinada)	Corresponde ao dispositivo "D"	32 bits	-2147483648 até 2147483647
FLOAT (precisão única)	É armazenado um número real incluindo as frações.	16 bits	-2^{128} até -2^{-126} , 0, 2^{-126} até 2^{128}
FLOAT (precisão dupla)	Corresponde ao dispositivo "D"	32 bits	-2^{1024} até -2^{-1022} , 0, 2^{-1022} até 2^{1024}
String	Uma string é armazenada. Corresponde ao dispositivo "D"	Variável	Até 255 caracteres
Temporizador	LIGA quando o tempo especificado é atingido. Corresponde ao dispositivo temporizador "T"		
Temporizador retentor	LIGA quando o tempo especificado é atingido. Corresponde ao dispositivo temporizador retentor "ST"		
Contador	LIGA quando a contagem especificada é atingida. Corresponde ao dispositivo do contador "C"		
Ponteiro	Uma posição de início de sub-rotina é armazenada. Corresponde ao dispositivo ponteiro "P"		

2.1.3

Registro de label

Produtividade

Qualidade

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [Local Label Setting MAIN [PRG]]

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

Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
 - Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool
- Structured Data Type
- Local Device Commen

Project

User Library

Connection Destination

	Class	Label Name	Data Type	Constant
1	VAR	Production_qty_attained	Bit	...
2	VAR	Parts_input_start_flag	Bit	...
3	VAR	Start_reject_pct_calc	Bit	...
4				...
5				...
6				...
7				...
8				...
9				...
10				...
11				...
12				...
13				...
14				...
15				...
16				...
17				...
18				...
19				...
20				...
21				...
22				...
23				...
24				...
25				...
26				...

Finalize as definições para registro do label.

Clique em  para continuar.

English

Simple

Q03UDE

Host Station

Line N...



2.1.4 Atribuição automática de labels a dispositivos

Produtividade

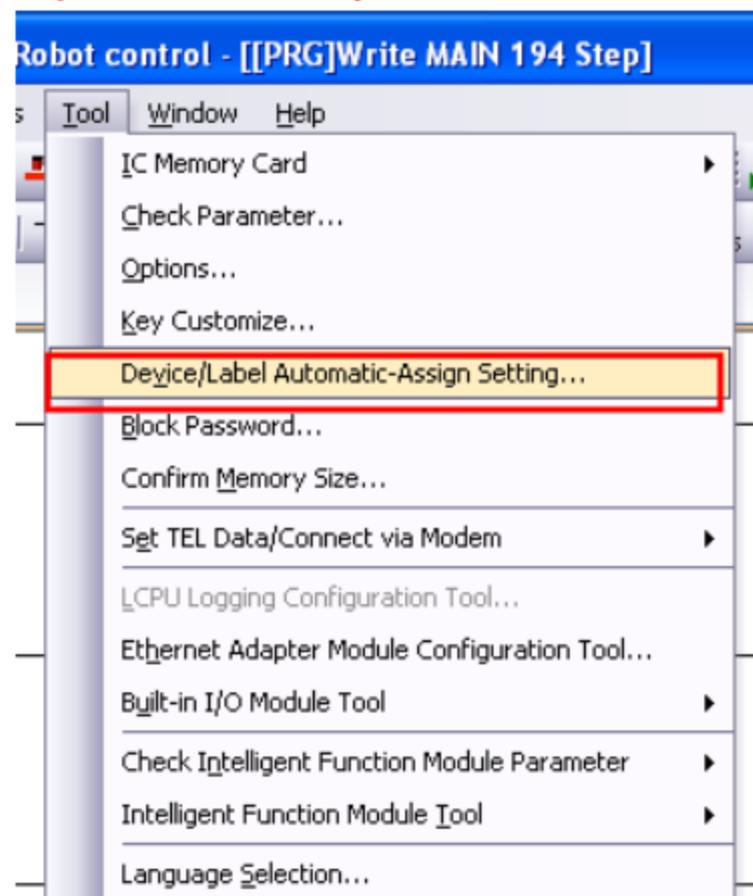
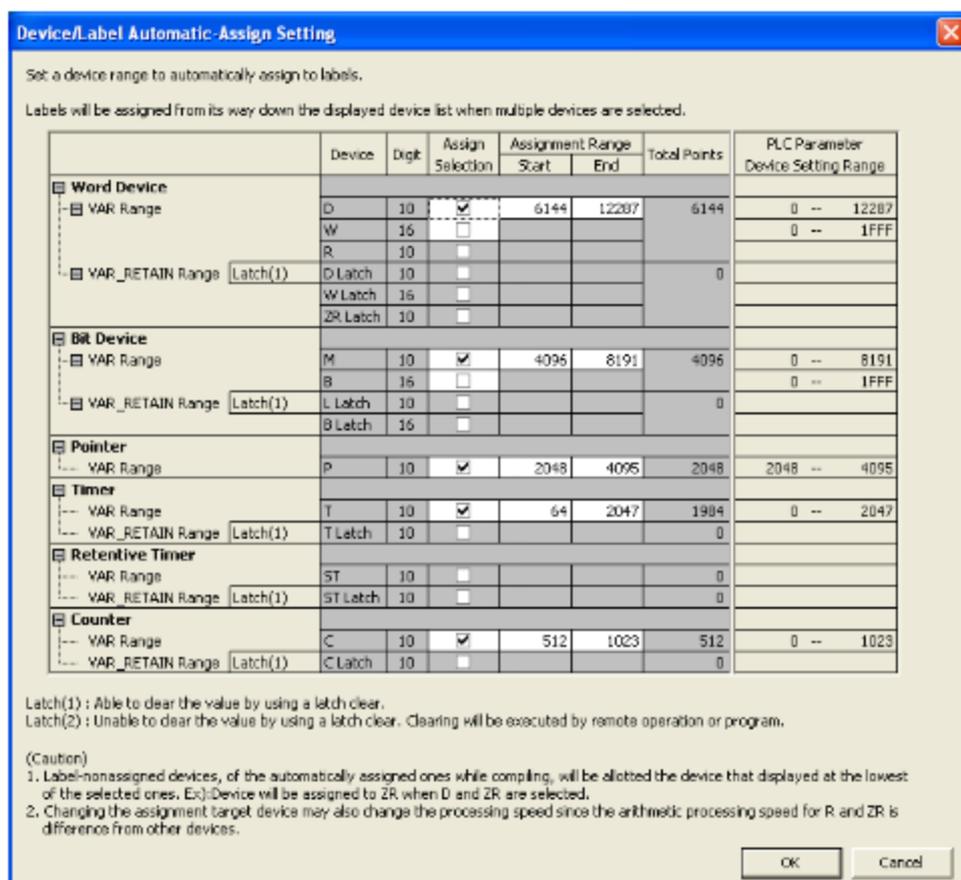
Qualidade

Converter um programa automaticamente atribui o dispositivo apropriado ao label de acordo com a classe e o tipo de dados. Ao utilizar um label, não é necessário saber qual dispositivo é atribuído.

Utilize **"Device/Label Automatic-Assign Setting"** (Configuração de atribuição automática de dispositivo/label) para alterar a range de dispositivos atribuídos ao label.

* Esta tela é a janela de definição do dispositivo de atribuição automática para as séries MELSEC-Q e MELSEC-L. A tela pode diferir para a série MELSEC-F.

<Iniciando a "Device/Label Automatic-Assign Setting" (Definição de atribuição automática de dispositivos/labels)>
No menu de GX Works2, selecione [Tool] (Ferramentas) – [Device/Label Automatic-Assign Setting] (Definição de atribuição automática de dispositivos/labels).



2.2

Organização dos ladder blocks usados repetidamente como function blocks para desvio

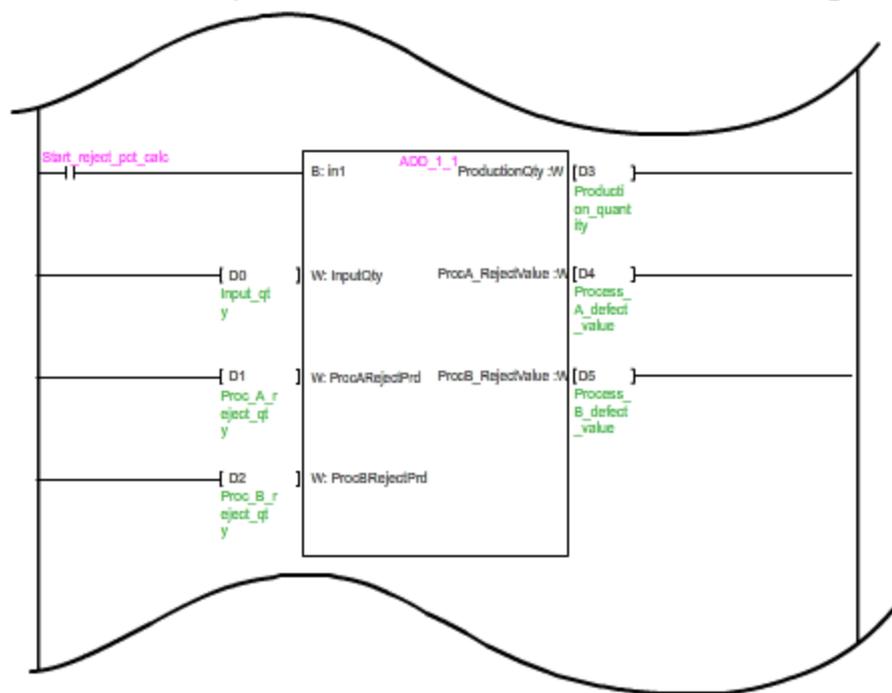
Em um programa em larga escala, alguns ladder blocks podem ser usados repetidamente. Ou alguns dos mesmos ladder blocks podem ser usados em programas diferentes. A eficiência do trabalho não pode ser aprimorada se você tiver que inserir o mesmo ladder block sempre que necessário. Além disso, se um defeito for encontrado em um ladder block comum, cada block deverá ser localizado e corrigido. Dessa forma, haverá uma perda de tempo significativa.

Use o **"Function Block (FB)"** para solucionar esses problemas.

Um ladder block que é usado repetidamente pode ser **organizado como um function block**, que pode ser usado em todo programa. A eficiência do trabalho é significativamente aprimorada.

Se um defeito é encontrado em um bloco comum, tudo o que você precisa fazer é corrigir o function block.

Assim, você não apenas economiza tempo, mas também evita o risco de não corrigir um dos blocos.



Pressione o botão para assistir à animação desde o início.



2.2.1

Criação e colocação dos function blocks

Produtividade

Qualidade

Visto que o function block será usado em diferentes programas, dispositivos reais (como X, Y e Z) não podem ser usados no function block. Esses dispositivos devem ser substituídos por **labels** como você aprendeu na Seção 2.1.

O procedimento para organizar ladder blocks em um function block é exibido em uma animação.

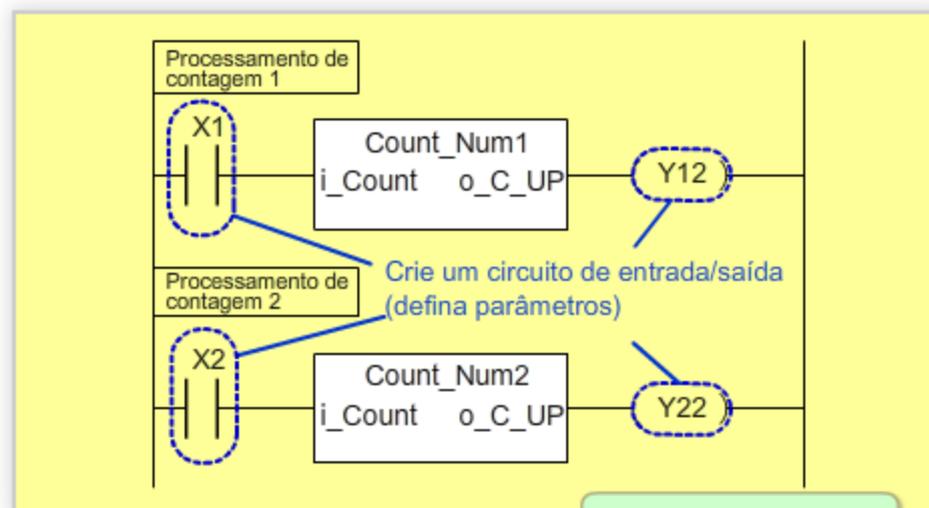
1. Prepare um programa que será disposto em uma function block.

2. Divida o programa em entrada e saída e divida os dispositivos internos por labels internas para fazer uma function block.

3. Posicione a function block criada no programa usando arrastar e largar.

4. Atribua parâmetros aos labels de entrada e saída da function block que você posicionou no programa.

5. Crie um circuito de entrada que passe o parâmetro para a label de entrada e um circuito de saída que receba o parâmetro do label de saída, antes e depois da function block.



Pressione o botão para voltar para o primeiro fluxo.



2.2.1 Criação e colocação dos function blocks



MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN (188)Step *]

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

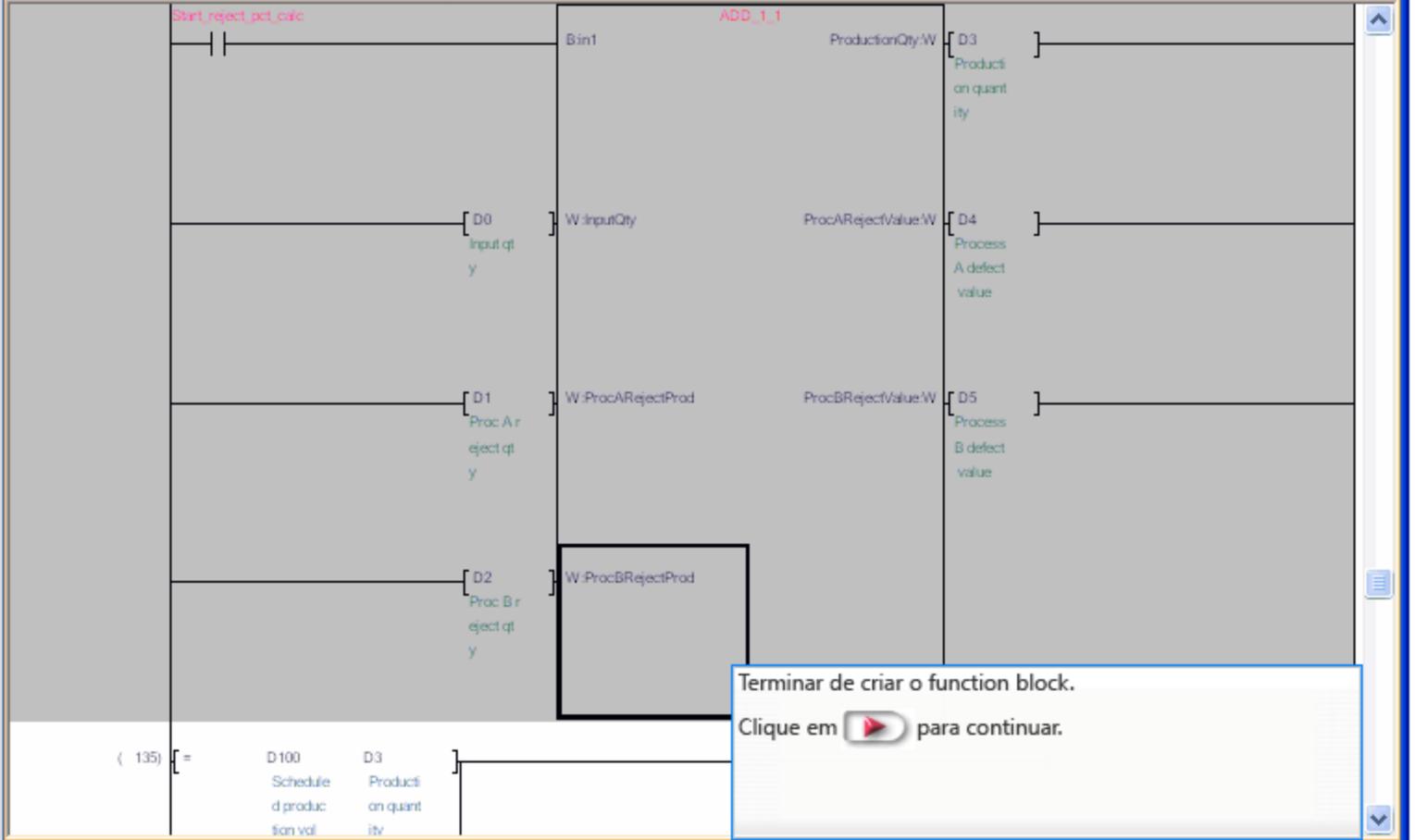


Navigation

[PRG]Write MAIN (188)Ste... Global Label Setting Global1 Local Label Setting MAIN [PRG] Function/FB Lat

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
 - Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - ADD_1
 - Program
 - Local Label
 - Structured Data Type



Terminar de criar o function block.
 Clique em para continuar.



2.2.2 Utilização da biblioteca de function blocks

Produtividade

Qualidade

Você aprendeu como criar um function block.

A Mitsubishi Electric organizou os programas de controle de diversos módulos como CPUs, módulos de entrada/saída analógicos, redes e módulos de posicionamento como function blocks, disponíveis sem custo como a **"Biblioteca da FB"**. Utilizar a FB Library elimina a necessidade de desenvolver programas de controle de módulo, que, tradicionalmente, era de responsabilidade do usuário. Até mesmo os módulos com os quais você não está familiarizado podem ser facilmente introduzidos.

Você pode fazer download da Biblioteca de FB na página **"MELSOFT Library Download"** (Download da biblioteca MELSOFT) no **MITSUBISHI ELECTRIC FA Website**.

<Antes>



<Se você usar a biblioteca de FB>



* Não há uma biblioteca de FB fornecida para a série MELSEC-F.
Os modelos suportados estão listados na página de download. Verifique o modelo suportado antes de fazer o download.

2.3 Alteração dos valores iniciais do dispositivo sem correção do programa

O valor inicial ou constante de um dispositivo é normalmente definido usando a instrução MOV antes do processamento do programa principal.

Neste caso, o programa deve ser diretamente corrigido cada vez que a operação de programa for alterada, de acordo com a aplicação do sistema.

Este método não só leva tempo, mas também há o risco de erros de correção ou falhas para fazer uma correção.

Use o "**Valor inicial do dispositivo**" para solucionar esses problemas.

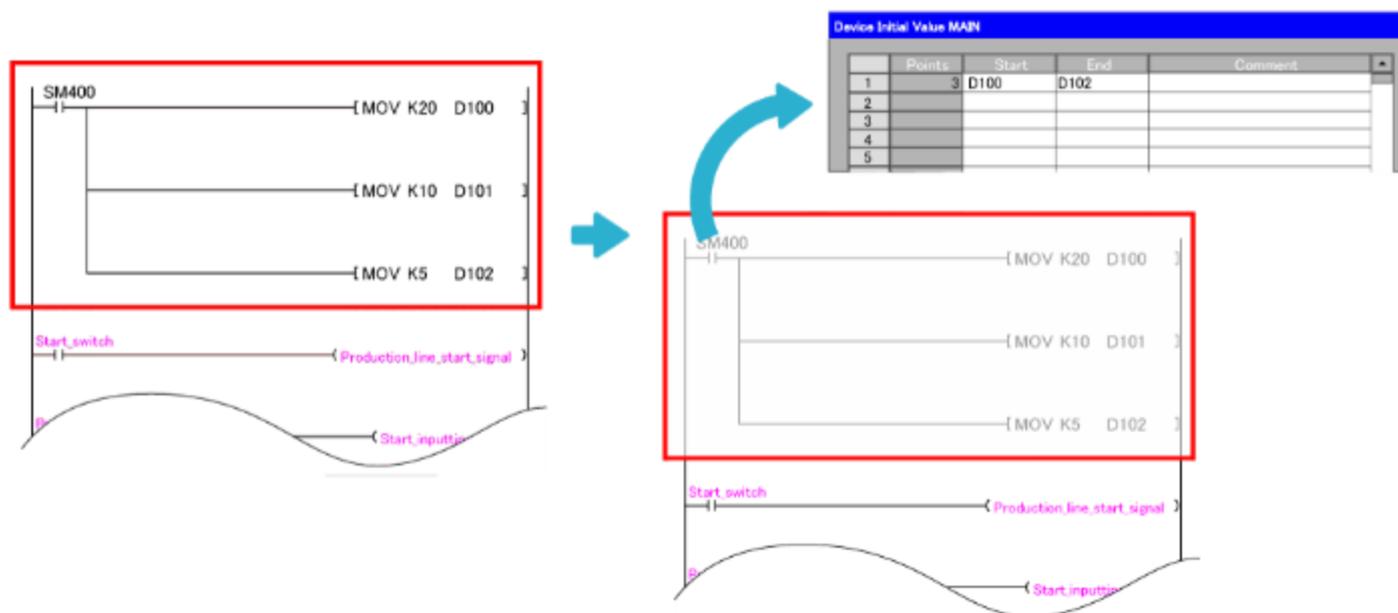
A utilização da função GX Works2 garante o gerenciamento adequado dos valores iniciais do dispositivo e elimina a necessidade de fazer correções de programa, assim lhe permitindo criar programas bem mais eficientes.

Além disso, esta função elimina a necessidade de um programa para definir o valor inicial do dispositivo, que reduzirá o volume do programa (uso da memória) e diminuirá o risco de falha.

Para o valor inicial do dispositivo, especifique a range de dispositivos para definir os valores iniciais.

Os valores iniciais reais são armazenados na **memória do dispositivo** e são atribuídos à range de dispositivos especificada.

Com áreas de memória do dispositivo preparadas de acordo com as aplicações do sistema, os valores iniciais do dispositivo podem ser alterados simplesmente alterando a área de memória do dispositivo que será atribuída.



2.3 Alteração dos valores iniciais do dispositivo sem correção do programa



MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN (166)Step *]

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



Navigation

[PRG]Write MAIN (166)Ste... Global Label Setting Global1 Local Label Setting MAIN [PRG]

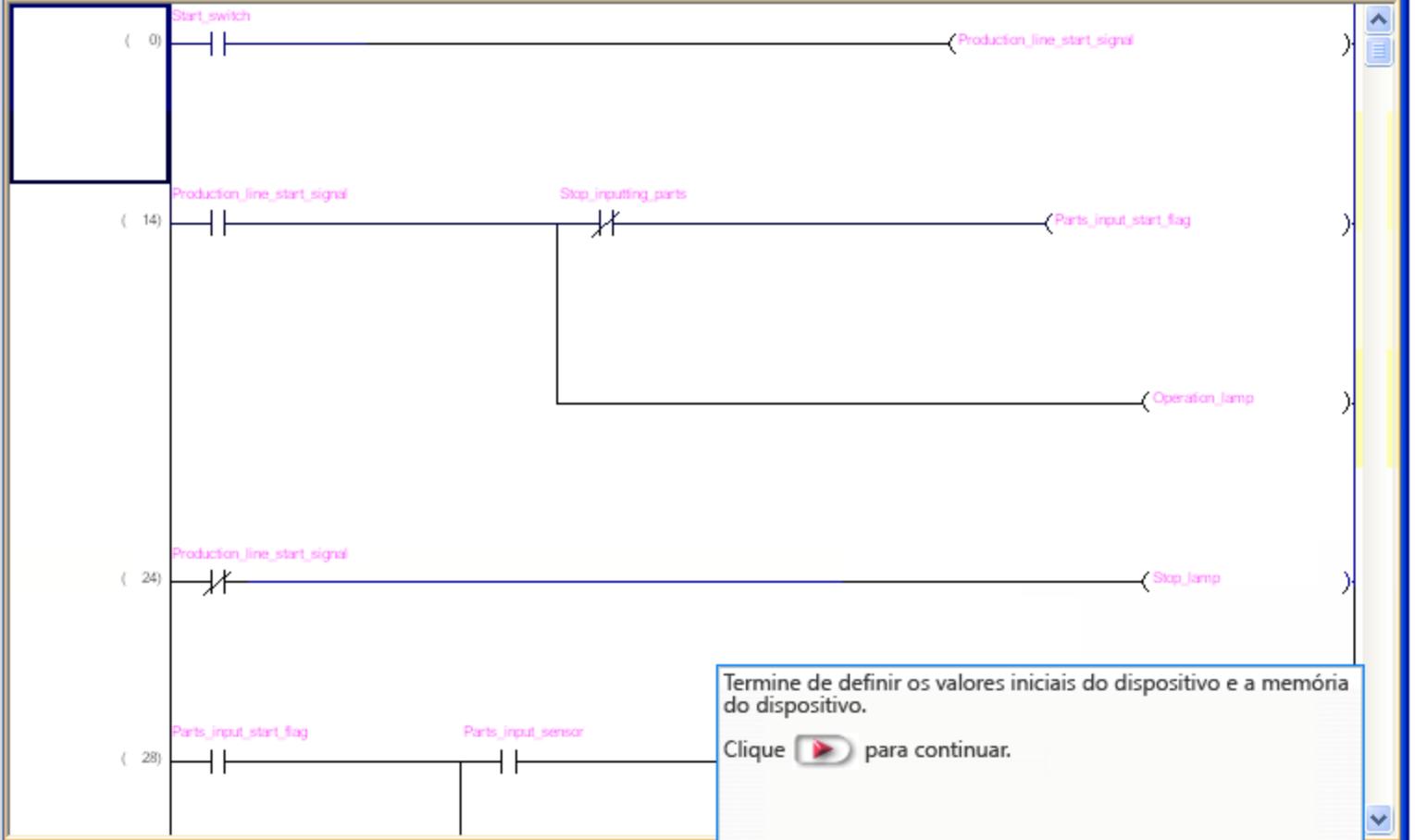
Project

- Global1
- Program Setting
- POU
- Program
 - MAIN
 - Program
 - Local Label
- FB_Pool
- Structured Data Type
- Local Device Commen
- Device Memory
 - MAIN
 - ProdQty1
- Device Initial Value
 - MAIN

Project

User Library

Connection Destination



Termine de definir os valores iniciais do dispositivo e a memória do dispositivo.

Clique para continuar.



2.4 Simplificação dos programas ladder



Em um programa ladder de larga escala, a ligação de dispositivos, instruções e linhas de controle podem se tornar bastante complicadas, dificultando compreender que processamento o programa realiza. Especialmente, os cálculos numéricos, como uma fórmula concluída em uma linha única, devem ser programadas usando uma combinação de instruções.

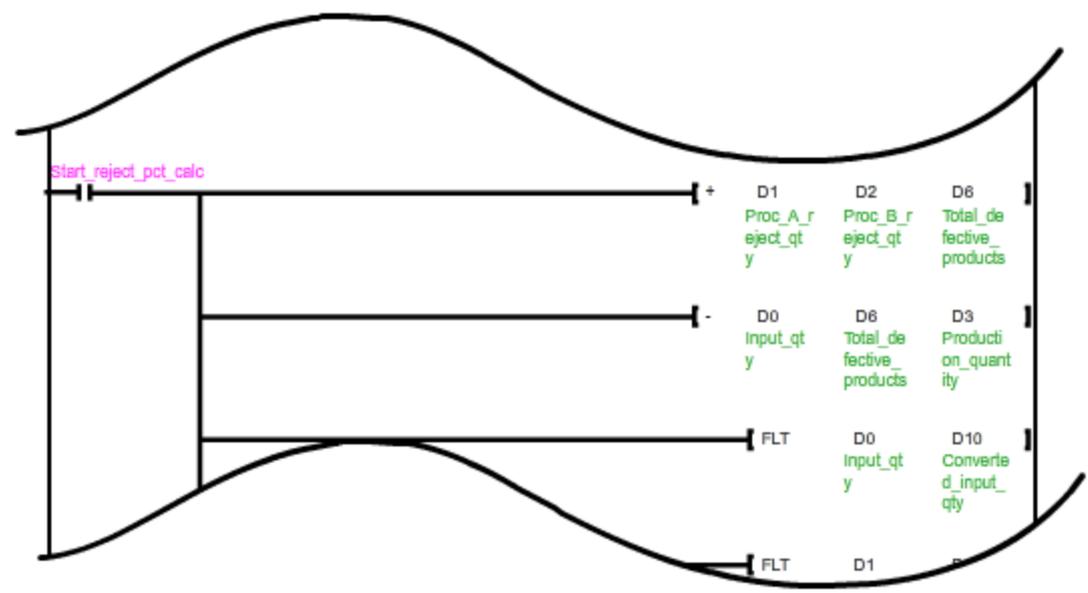
Use o "Texto estruturado em linha" para solucionar esses problemas.

Um programa ladder é substituído por um programa que é parcialmente escrito na linguagem de estrutura de texto (ST). A linguagem ST é uma linguagem de programação de controle de sequência similar à linguagem C usada para programação de software de computadores.

Os cálculos numéricos podem ser escritos usando fórmulas, de modo que os programadores que não estiverem familiarizados com C podem usar o texto estruturado em linha.

A seguinte figura mostra um exemplo no qual parte do programa ladder para o sistema é substituído com texto estruturado em linha.

Você pode ver que o último programa complicado agora é fácil de compreender.



2.4

Simplificação dos programas ladder

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 114 Step]

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

Navigation [PRG]Write MAIN 114 Step Global Label Setting Global1 Local Label Setting MAIN [PRG]

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Type
 - Local Device Commen
- Device Memory
- Device Initial Value

Start reject pct calc

(75)

|

|

```

INT_TO_REAL_E(Start_reject_pct_calc,D2,inp2);
IF inp0>E0 THEN
resultA:=(inp1/inp0)*E100;
END_IF;
REAL_TO_INT_E(Start_reject_pct_calc,resultA,D4);
IF (inp0-inp1)>E0 THEN
resultB:=(inp2/(inp0-inp1))*E100;
END_IF;
REAL_TO_INT_E(Start_reject_pct_calc,resultB,D5);

```

(87)

[

=

D100	D3
Schedule	Producti
d produc	on quant
tion vol	ity
ume	

Production_qty_ attained

(94)

|

|

D4	D101
Process	Process
A defect	A defect
value	\$resho
	ld

Proc_A_reject_pct_error_lamp

Termine de definir o texto estruturado em linha.
Clique  para continuar.

2.5 Criação de programas que sejam fáceis de compreender e ler

Produtividade

Qualidade

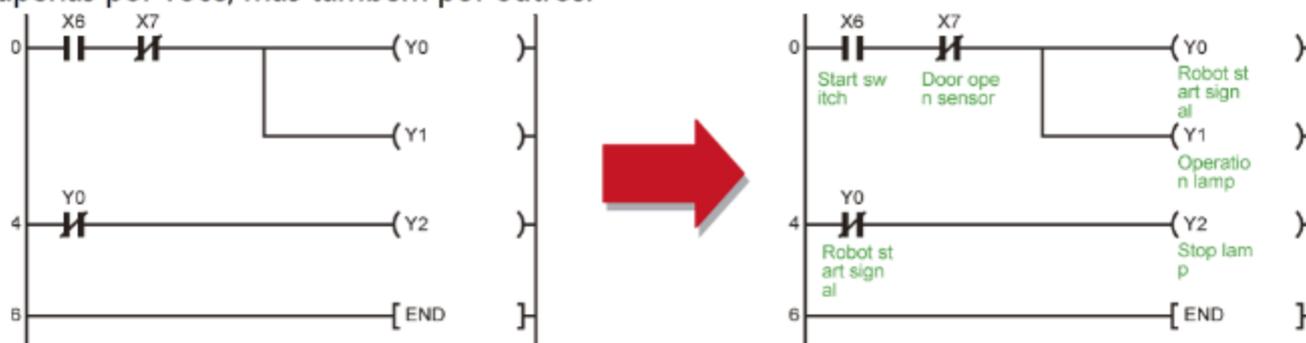
Você pode ter dificuldade de entender os detalhes de controle em um programa de grande escala apenas olhando para o programa. Os seguintes problemas podem ocorrer como resultado:

- Você faz erros de programa (como fornecer instruções ou dispositivos incorretos).
- Você não conseguiu encontrar as causas dos erros de programa.
- Alguém que assuma a programação não consegue entender os detalhes de controle.

Utilize **"Comentário"** para solucionar esses problemas.

Lembretes, tais como informações de controle e nomes de dispositivo, podem ser anexados ao programa para que os detalhes do controle possam ser facilmente compreendidos.

Estes comentários devem ser inseridos sempre que possível para criar programas que sejam fáceis de compreender, não apenas por você, mas também por outros.



O GX Works2 permite que os seguintes comentários sejam inseridos.

Tipo de comentário	Range de comentário
Comentário do dispositivo	Um comentário pode ser anexado a um dispositivo. Este comentário indica a aplicação de cada dispositivo e o tipo de dispositivo de E/S conectado.
Statement	Um comentário pode ser anexado a um ladder block. Este comentário faz do fluxo de processamento fácil de entender.
Nota	Um comentário pode ser anexado a uma instrução de bobina/aplicação no programa. Este comentário torna o conteúdo das instruções de uma bobina (saída) e de uma aplicação fáceis de compreender.

2.5.1

Escrevendo um comentário para cada ladder block

Produtividade

Qualidade

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Write MAIN 145 Step]

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

Navigation

Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

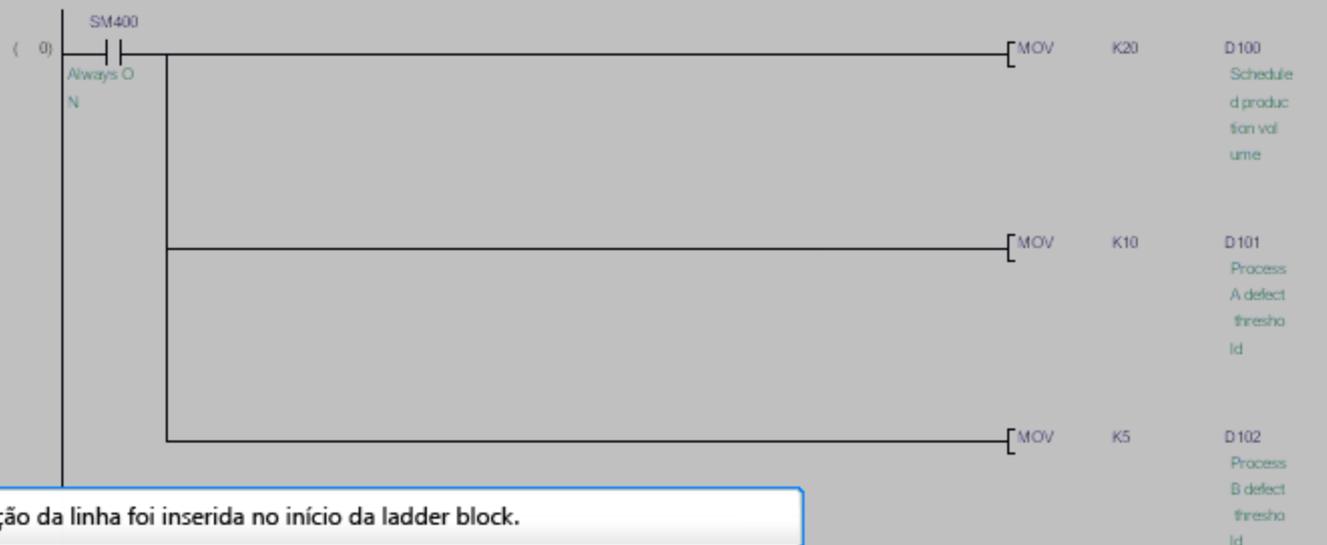
Project

User Library

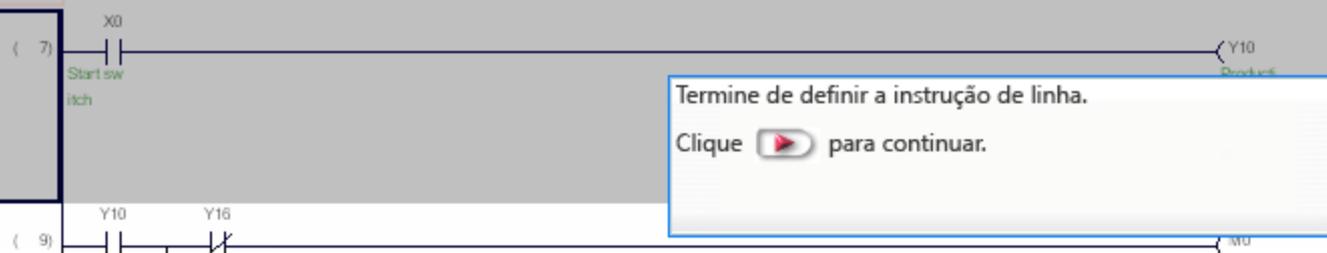
Connection Destination

[PRG]Write MAIN 145 Step

Scheduled prod qty setting



Prod line start



English

Simple

Q03UDE

Host Station

(7/14)



2.5.2

Escrevendo um comentário para cada saída (bobina, instrução de aplicação)

Produtividade

Qualidade

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 181 Step]

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

Navigation

Project

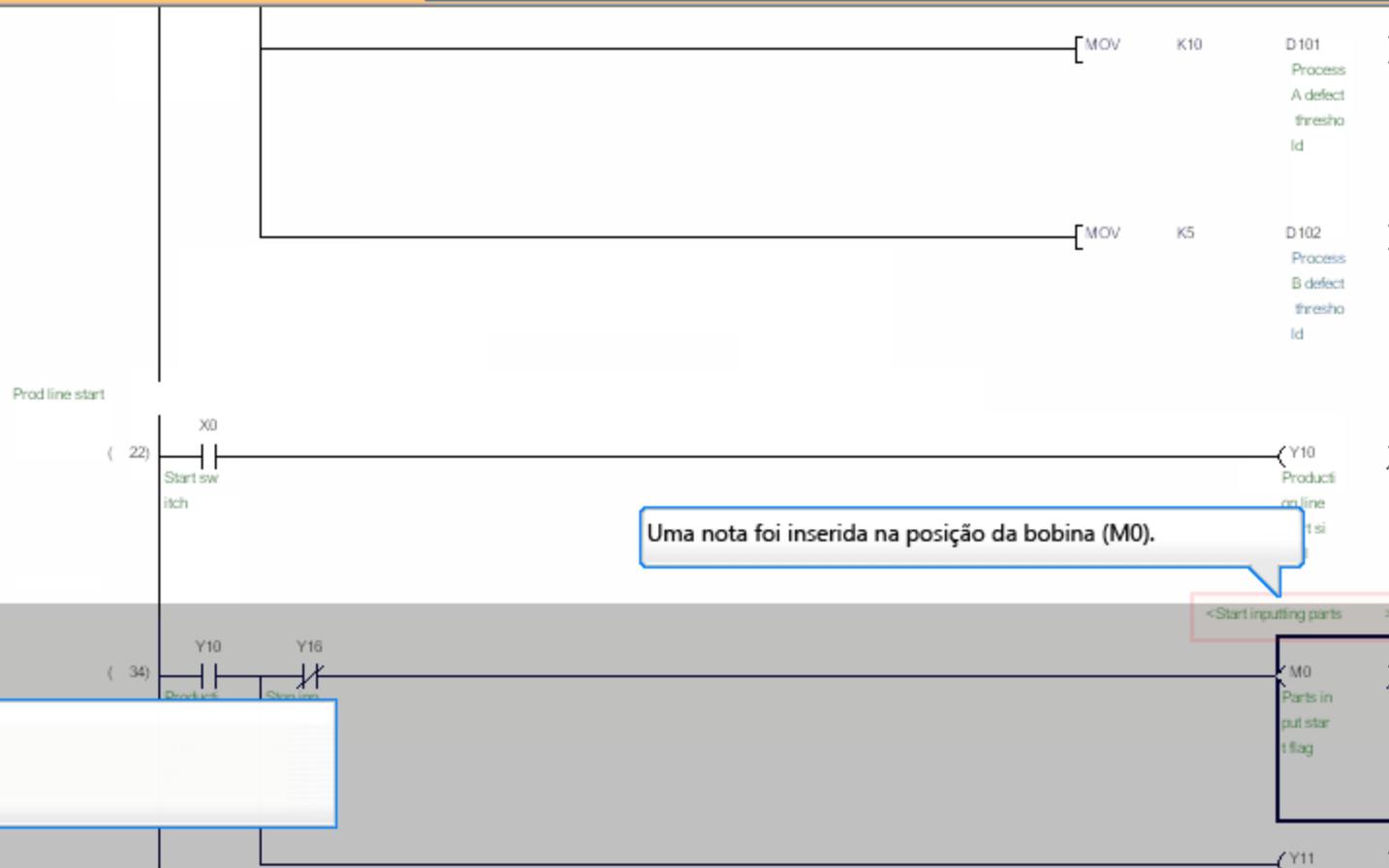
- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

Termine de definir a nota.

Clique  para continuar.

[PRG]Write MAIN 181 Step



English

Simple

Q03UDE

Host Station

(37 M...

2.6

Tornando fácil a leitura de programas que contêm relés/registros especiais

Se relés especiais, registros especiais e/ou dispositivos de módulo de função inteligente são usados em um programa, pode ser difícil de entender todas as aplicações e funções desses dispositivos. Você precisa ler o programa com o manual na sua mão.

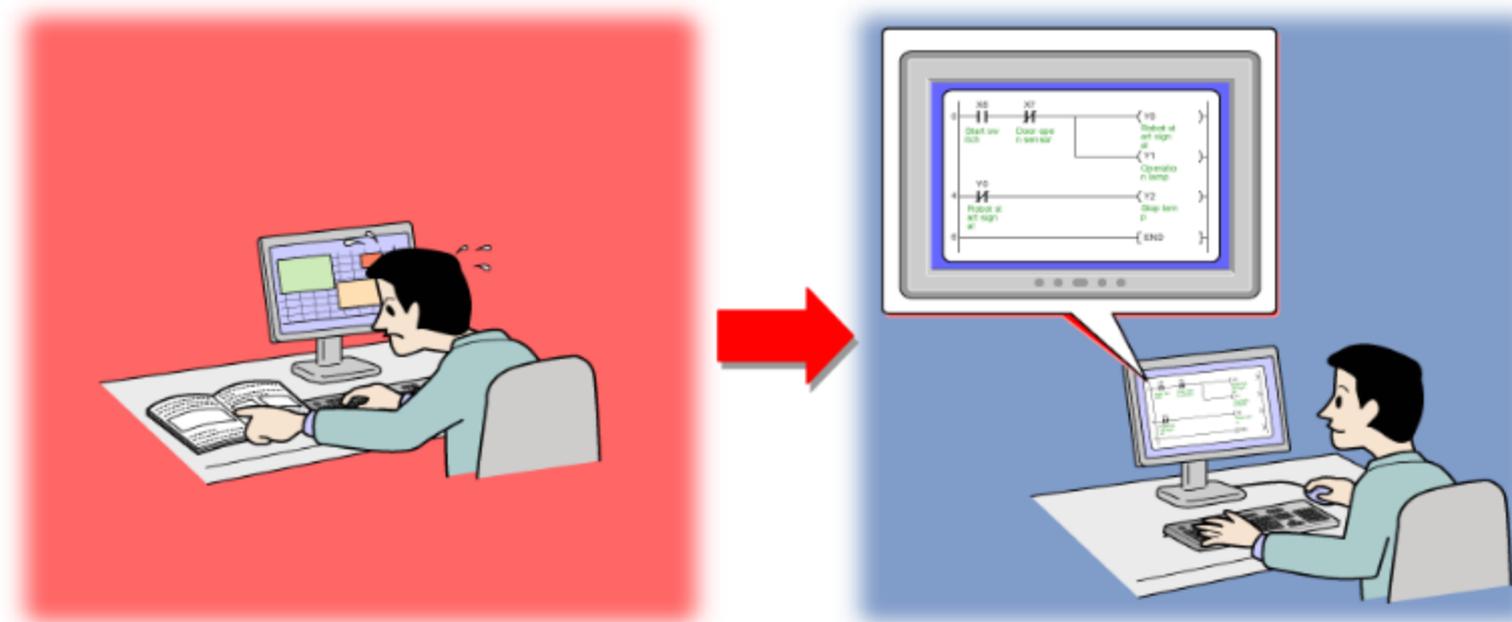
Embora o programa seja mais fácil de ler se um comentário é anexado ao dispositivo, será requerido tempo e esforço considerável para anexar os comentários se são usados vários dispositivos.

Utilize "**Comentário de amostra**" para solucionar esses problemas.

O GX Works2 fornece comentários de amostra descrevendo as aplicações e funções de relés especiais, registros especiais e dispositivos do módulo de função inteligente.

Utilizar esses comentários de amostra torna muito mais fácil anexar comentários aos dispositivos para tornar o programa fácil de ler.

Os comentários de amostra podem ser modificados conforme necessário.



2.6

Tornando fácil a leitura de programas que contêm relés/registros especiais

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 194 Step]

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

Navigation

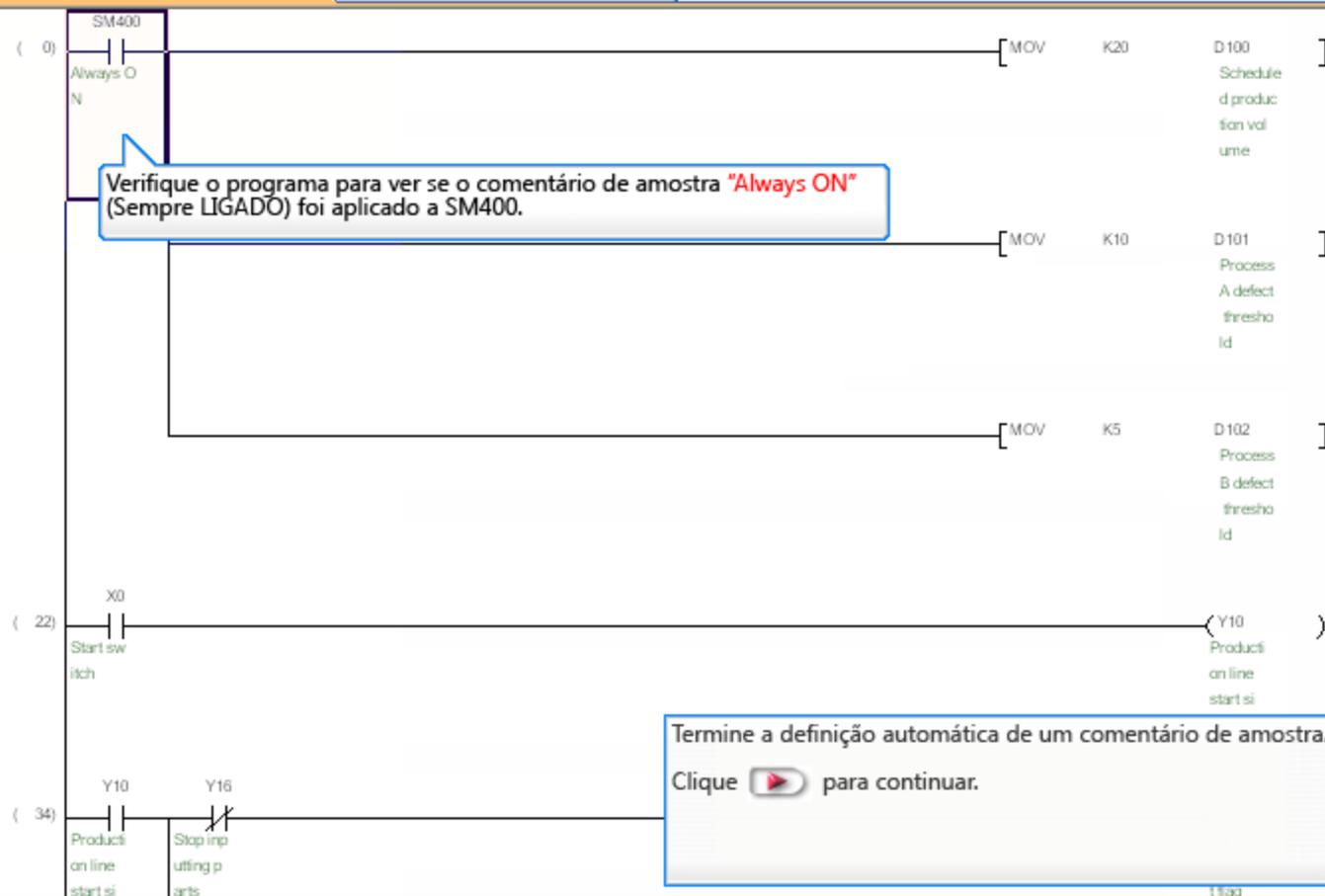
Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination



English

Simple

Q03UDE

Host Station

(15 N)

Capítulo 3 Depuração

Etapas de aprendizagem do Capítulo 3

No Capítulo 3, você aprenderá as funções usadas para depuração.

O GX Works2 fornece várias ferramentas diferentes de monitoração e depuração para corrigir erros (bugs).

Crie programas sem erros e de qualidade usando estas ferramentas de monitoramento e depuração.

- 3.1 Monitoramento apenas de dispositivos e rótulos alvo
- 3.2 Verificação do status de utilização de dispositivos e rótulos
- 3.3 Coleta de informações sobre a mudança de valor do dispositivo ao longo do tempo
- 3.4 Alteração dos valores iniciais do dispositivo sem correção do programa
- 3.5 Operação do programa de depuração passo a passo
 - 3.5.1 Usar a função de execução de step
- 3.6 Simulação do funcionamento de um dispositivo externo
 - 3.6.1 Inserir valores de dispositivo para a definição do sistema de E/S
 - 3.6.2 Usar o formato do diagrama temporal para definição do sistema de E/S

3.1

Monitoramento apenas de dispositivos e labels alvo

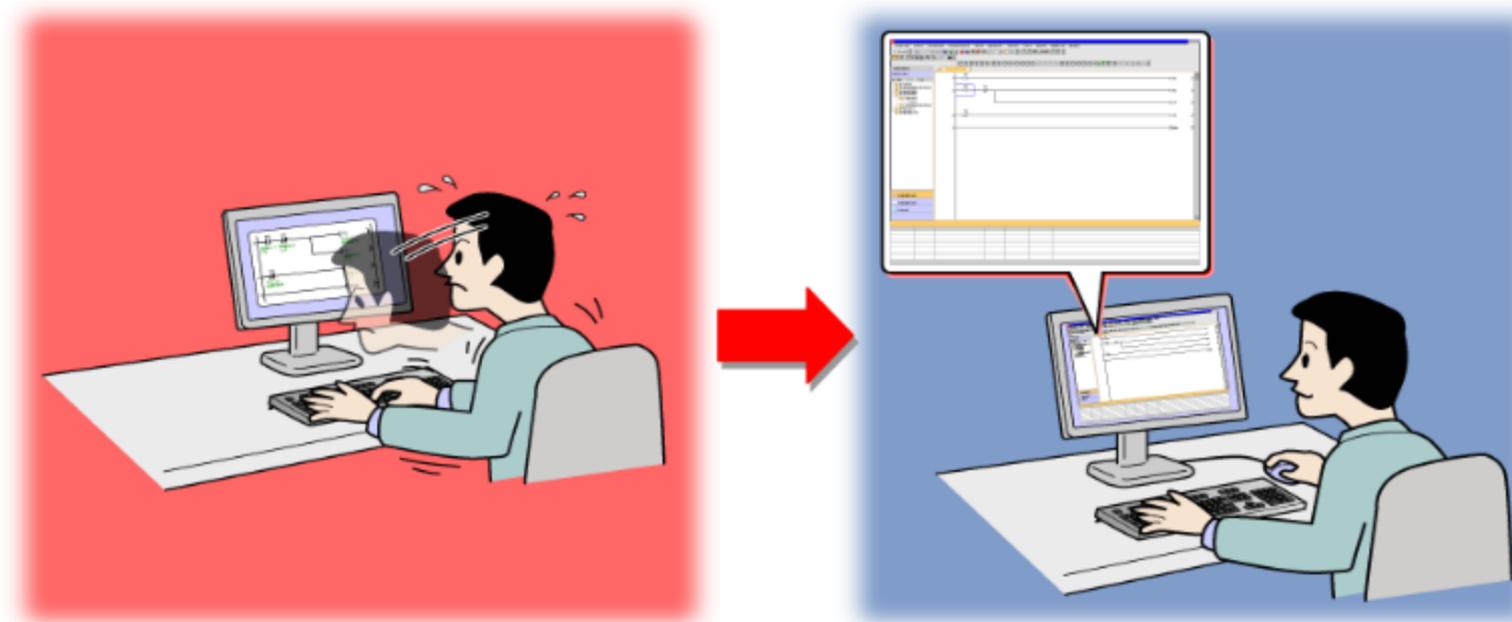
Qualidade

Um programa usa várias instruções e dispositivos. Além disso, o comprimento vertical longo de um programa significa que apenas parte dele é exibida ao mesmo tempo na tela de monitoração de um computador. Portanto, o monitor ladder sozinho não pode monitorar toda a operação.

Utilize **"Exibir"** para solucionar esses problemas.

Esta função pode ser usada para monitorar apenas os dispositivos e os labels que você especificou com antecedência.

Duas ou mais exibições podem ser criadas para garantir que cada range seja monitorada.



3.1

Monitoramento apenas de dispositivos e labels alvo

Qualidade

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

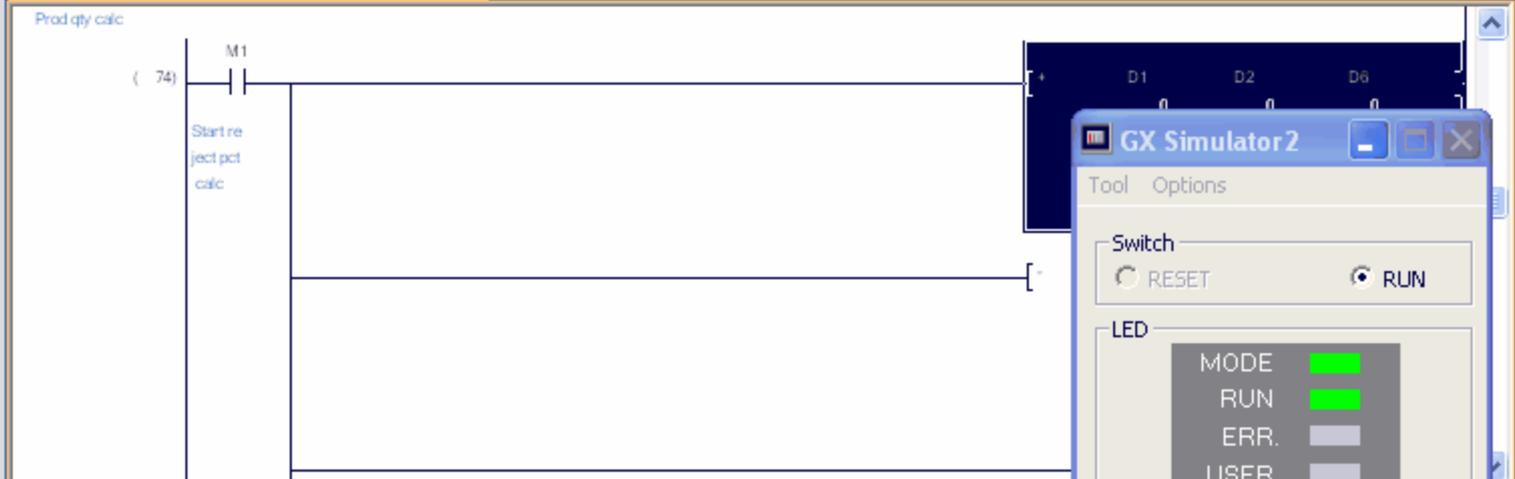
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG]Monitor Executing M...

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program



GX Simulator 2

Tool Options

Switch

RESET RUN

LED

MODE

RUN

ERR.

USER

Watch 1 Os valores presentes dos dispositivos registrados com Watch 1 (Exibir 1) são monitorados.

Device/Label	Current Value	Data Type	Class	Device	Comment
X0	0	Bit		X0	Start switch
X1	0	Bit		X1	Parts input sensor
X2	0	Bit		X2	
X3	0	Bit		X3	

Termine o registro e o monitorização Watch (Exibir).

Clique para continuar.

Watch 1 Watch 2

English

Simple

Q03UDE

Simulation

(84 N)



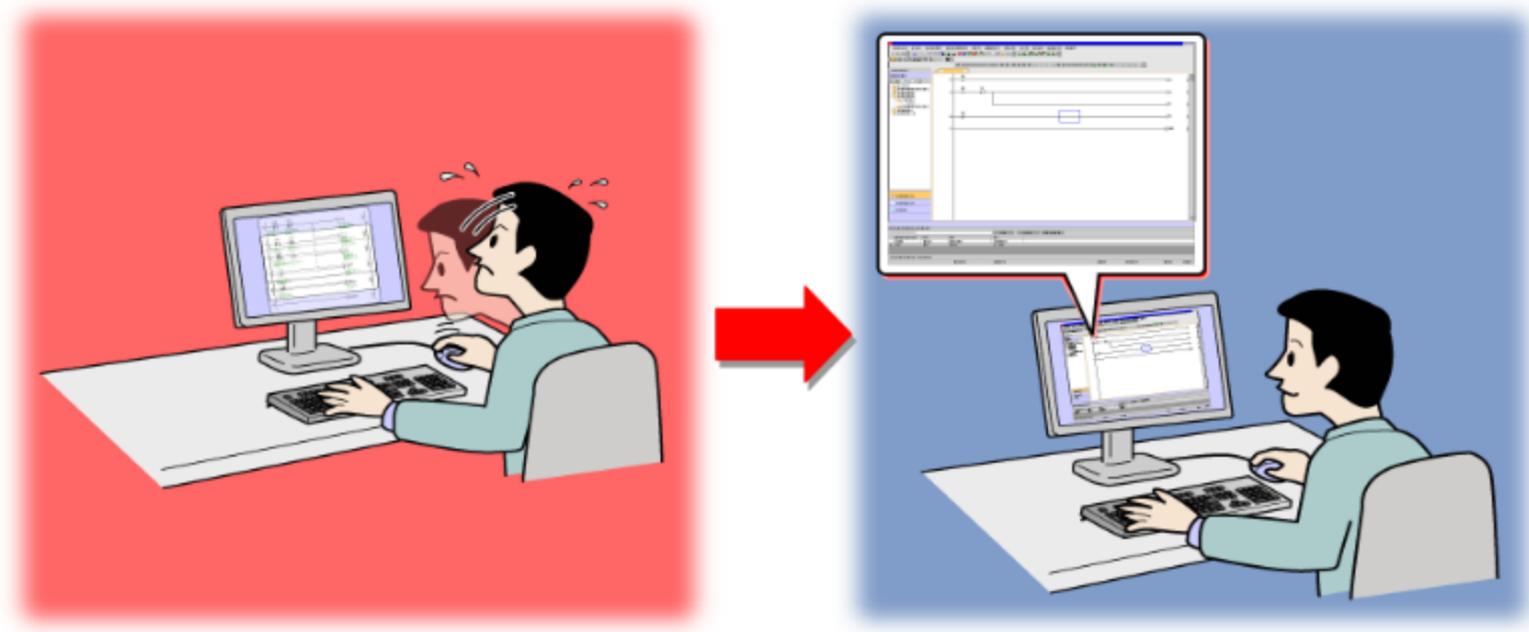
3.2

Verificação do status de utilização de dispositivos e labels



Um programa utiliza os mesmos dispositivos e labels em locais diferentes.
Você pode querer verificar o status de utilização desses dispositivos e labels, comparando-os entre os locais.

Utilize "**Referência cruzada**" para fazer isso.
Esta função relaciona as localizações dos dispositivos e dos labels que correspondem às condições de busca para que você possa compará-las e verificar a utilização incorreta.



3.2

Verificação do status de utilização de dispositivos e labels

Qualidade

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 194 Step]

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

Navigation [PRG]Write MAIN 194 Step

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
- Program
 - MAIN

Prod qty calc

(74) M1 Start re ject pct calc

O cursor do programa moveu-se para a posição da etapa nº 74.

Device/Label	Device	Instruction	Ladder Symbol	Position	Data Name
D1	D1	+P	-[]-	Step No.63	
D1	D1	+	-[]-	Step No.84	
D1	D1	FLT	-[]-	Step No.92	

5: device/cross reference information of label "D1"

Termine usando a Referência cruzada.
Clique para continuar.

English Simple Q03UDE Host Station (84 M...

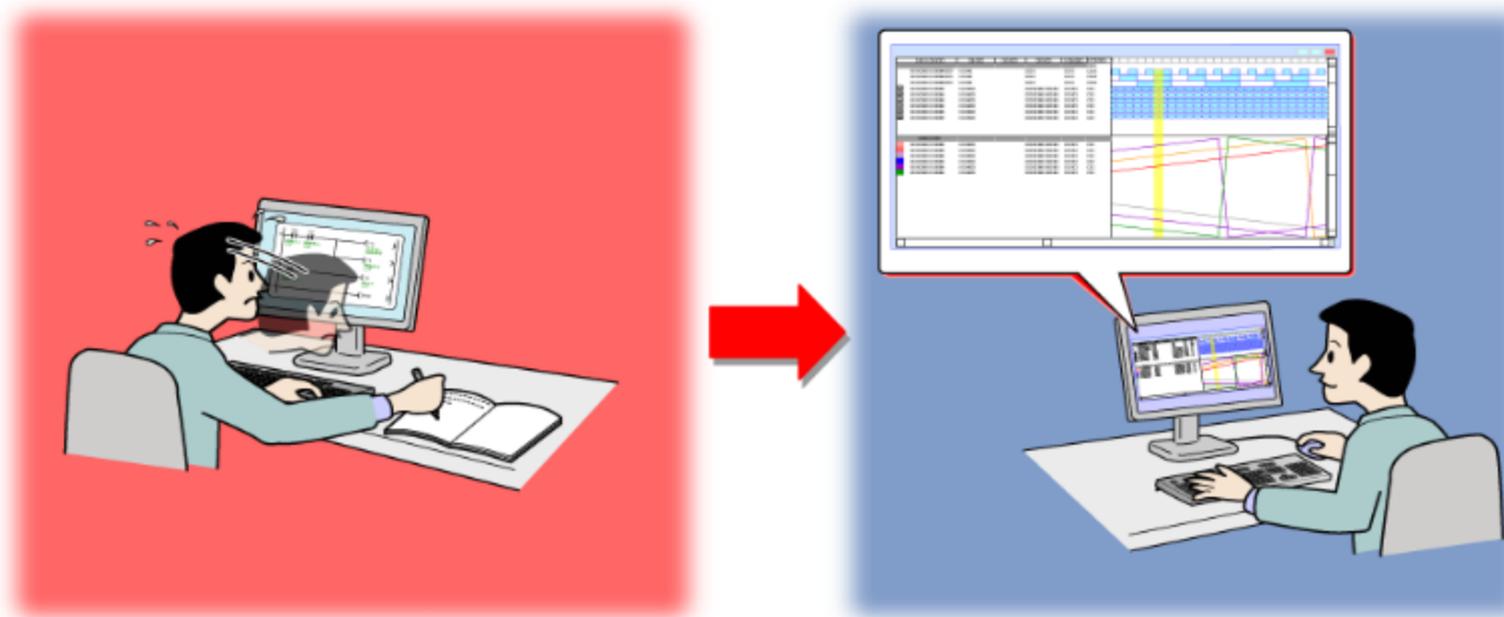
3.3

Coleta de informações sobre a mudança de valor do dispositivo ao longo do tempo

Qualidade

Você pode desejar confirmar se as alterações no dispositivo e nos valores de label estão dentro da range do projeto ou executam a troubleshooting verificando as mudanças que ocorreram em caso de falha.

Utilize o "**Rastreo de amostragem**" neste caso. (Apenas para as séries MELSEC-Q e MELSEC-L)
Esta função pode ser usada para monitorar e gravar as alterações nos valores de dispositivo e label ao longo do tempo. O registro pode ser salvo em arquivos.



3.4 Alteração dos valores iniciais do dispositivo sem correção do programa

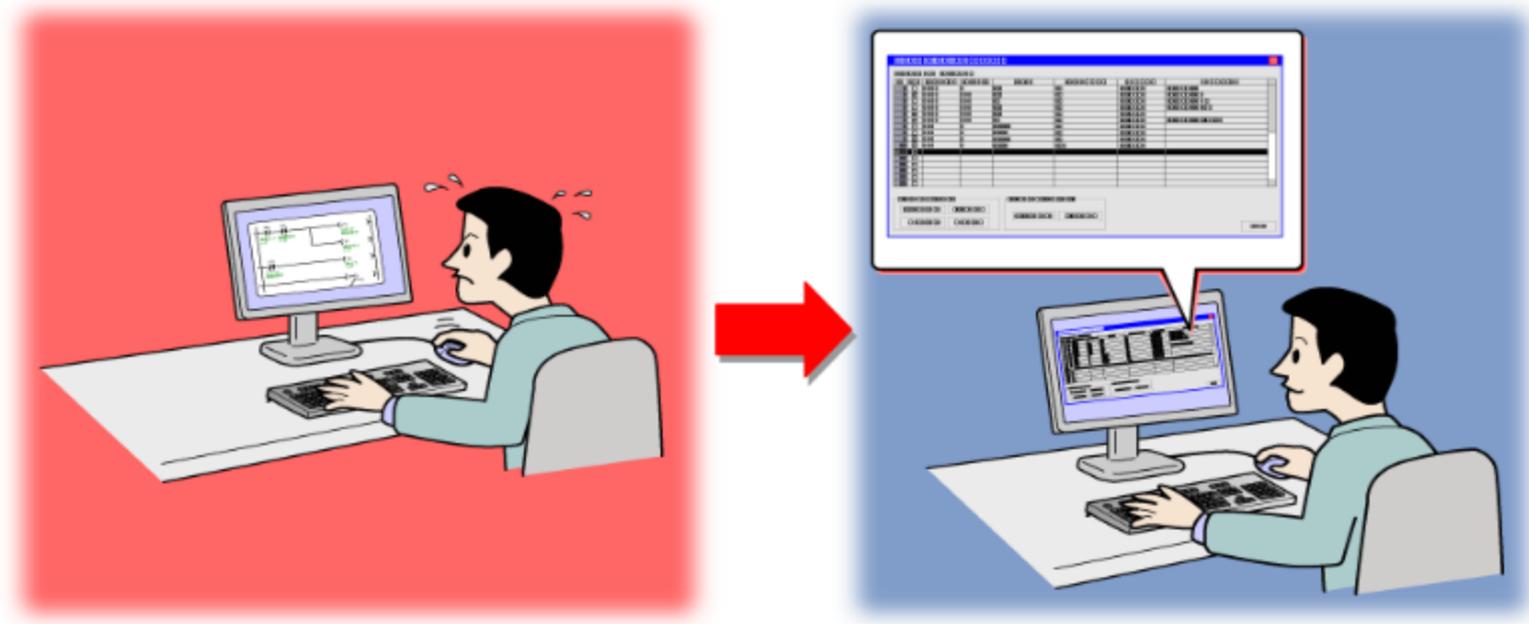


Ao realizar a depuração, você pode desejar mudar à força os valores do dispositivo para verificar a diferença na operação do programa.

Contudo, isso significa que você deve modificar o programa a cada vez que o valor do dispositivo for alterado, dependendo muito tempo e esforço.

Além disso, se você esquecer de definir o programa modificado de volta para as definições originais, isso pode causar outra falha.

Utilize o **"Teste do dispositivo condicionado executável"** neste caso. (Apenas para as séries MELSEC-Q e MELSEC-L)
Esta função altera os valores do dispositivo após a execução do número de step especificado com antecedência, sem ter de modificar o programa.



3.4 Alteração dos valores iniciais do dispositivo sem correção do programa



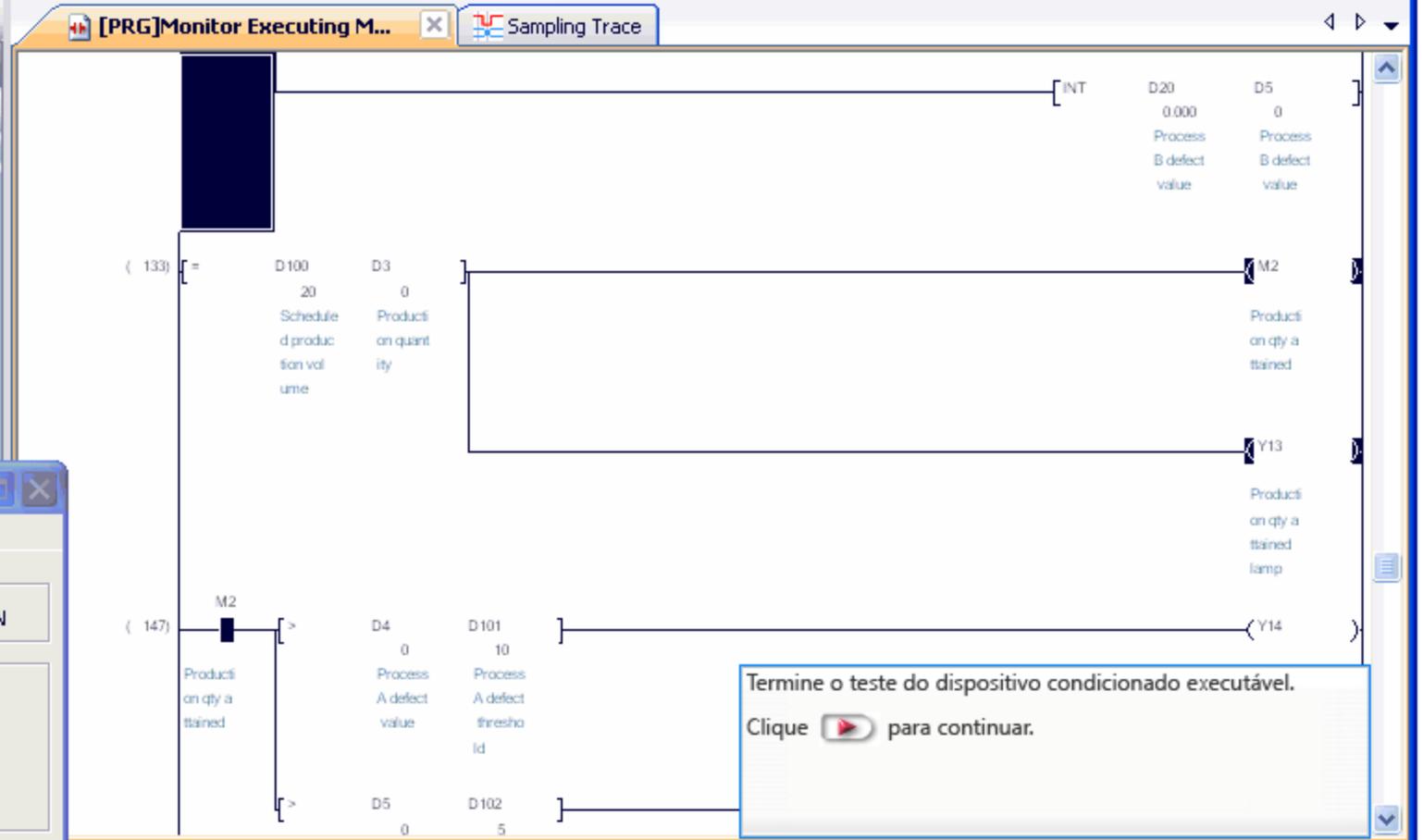
MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

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

Toolbar with various icons for file operations, editing, and simulation control.

Navigation Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool



GX Simulator 2

Tool Options

Switch: RESET STOP RUN

LED:

- MODE
- RUN
- ERR.
- USER

Termine o teste do dispositivo condicionado executável.
 Clique para continuar.

Simulation control buttons: play, stop, back, forward, etc.

3.5 Operação do programa de depuração passo a passo

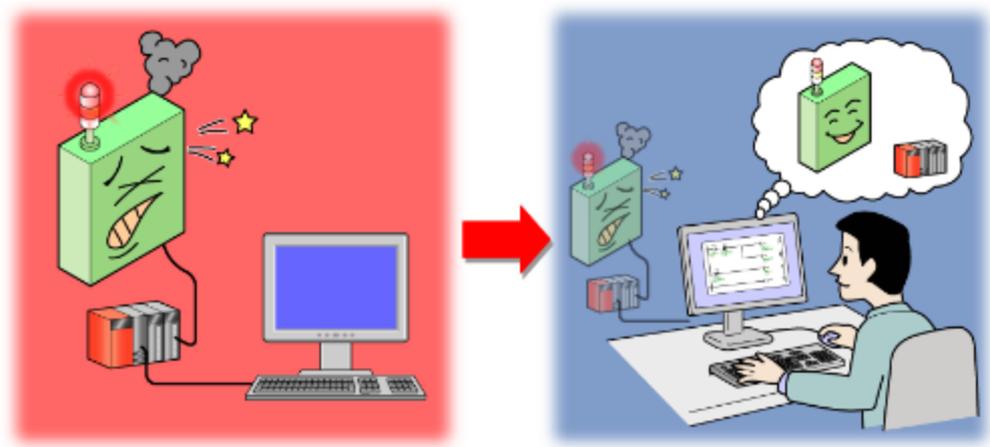


Durante a depuração, você pode desejar confirmar a execução de instrução em cada step ou verificar se há alterações em um valor de dispositivo. No entanto, a depuração passo a passo pode ser difícil devido à velocidade rápida de execução do programa (tempo de scan).

Utilize a **"Função de execução de step"** para resolver este problema. (Apenas para as séries MELSEC-Q e MELSEC-L) Esta função permite a execução do programa em um passo de cada vez e, portanto, implementa a depuração passo a passo. Utilize a função de execução de step juntamente com a **função de simulação**. (A função de execução de step não pode ser utilizada para depuração no PLC real.)

As funções a seguir podem ser usadas pela função de execução de step.

Função	Descrição
Execução de interrupção	Esta função executa o programa até que sejam satisfeitas as condições de interrupção especificadas. A execução do programa é interrompida quando são satisfeitas as condições de interrupção. Utilize um ponto de interrupção e um dispositivo de interrupção para especificar as condições de interrupção.
Execução de step	Esta função executa o programa passo a passo.
Execução parcial	Esta função executa o programa somente a partir do local especificado.



3.5.1 Usar a função de execução de step

Qualidade

Especifique um **ponto de interrupção** e **dispositivo de interrupção** como o local inicial e condição de start de depuração, respectivamente.

Você também pode especificar um **intervalo para ignorar** dentro do qual você deseja evitar temporariamente a execução do programa. *(Apenas para as séries MELSEC-Q e MELSEC-L)*

Quando as condições de interrupção forem satisfeitas após a **execução da interrupção** ter sido iniciada, a execução do programa é interrompida.

Depois disso, ao executar a operação do programa passo a passo com a **função de execução de step**, verifique se há alterações no valor de dispositivo para localizar uma falha.

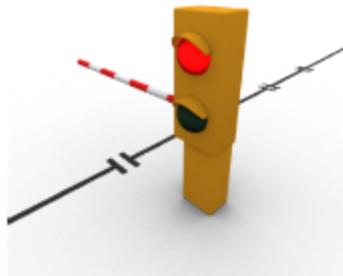
<Ponto de interrupção>

Defina um ponto de interrupção no qual você deseja interromper a execução do programa. Especifique isso em unidades de etapas. Até 64 pontos de interrupção podem ser definidos em todo o projeto.



<Dispositivo de interrupção>

Defina um dispositivo de interrupção com base na execução do programa que é interrompido quando o valor do dispositivo ou do label satisfaz a condição especificada. Podem ser definidos dispositivos de até 16 bits e/ou palavras.



<Ignorar intervalo>

Defina um intervalo dentro do qual o programa não deve ser executado, em unidades de ladder blocks, durante a execução de step. Até 16 ranges podem ser definidas em todo o projeto.



<Aplicação ignorar intervalo>

Um ponto de falha pode ser diminuído usando **ignorar intervalo**.

Realize a execução da interrupção com e sem a definição ignorar intervalo.

Se uma falha ocorrer somente quando a definição ignorar intervalo for liberada, isso indica que o intervalo inclui uma falha.

3.5.1 Usar a função de execução de step



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Read MAIN (Read Only) 194 Step]

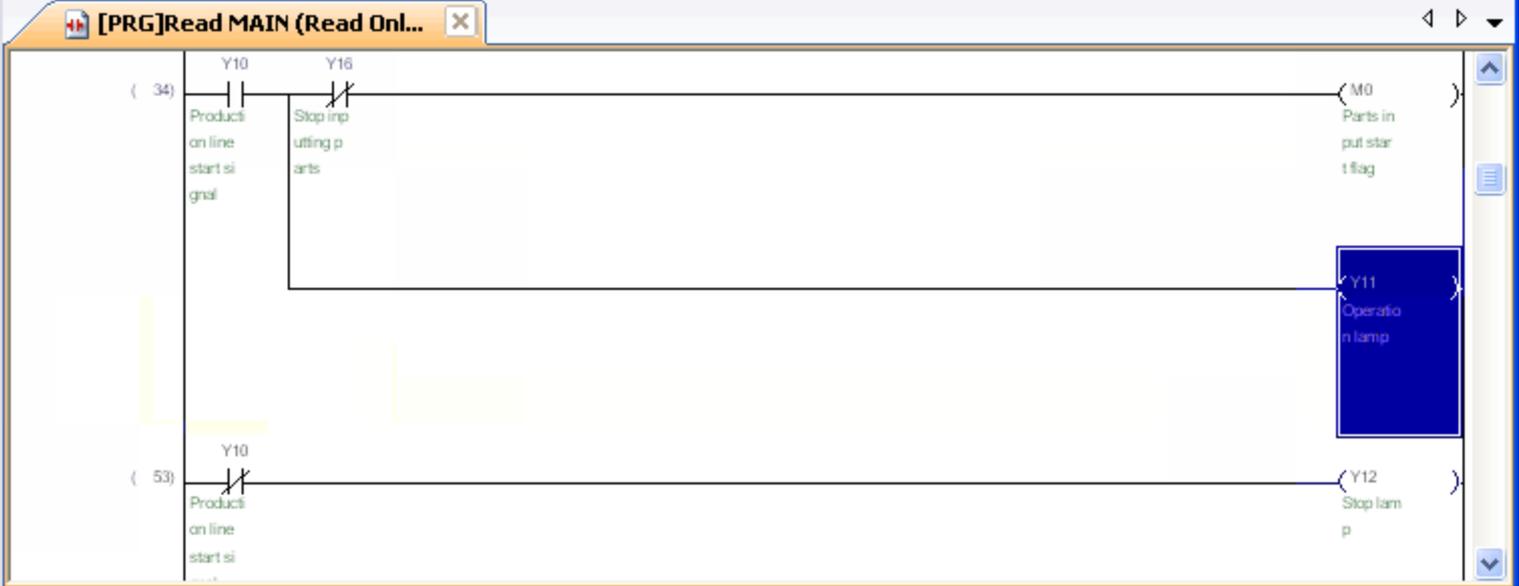
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation icons: Save, Undo, Redo, Copy, Paste, etc.

Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
- Program
 - MAIN



Break Device

New Condition Cancel Cancel All

Combination

Judge each break device (OR condition) Judge all break devices (AND condition)

Enable/Disable	Comparative Source (Device/Label)	Condition	Comparative Target (Value/Device/Label)	Comparative Type

A depuração foi concluído. Finalizar a função de execução de step
 Clique para continuar.

Skip Range Break Point Break Device

English Simple Q03UDE Host Station (52 N)

Navigation icons: Stop, Play, Previous, Next, Home

3.6

Simulação da operação de um dispositivo externo

Qualidade

Não é possível executar a depuração em conjunto com a operação de um dispositivo externo em um ambiente de desenvolvimento em que não pode ser usado um dispositivo externo, como a função de simulação.

Para resolver este problema, um programa de depuração que simula o funcionamento de um dispositivo externo é convencionalmente adicionado.

No entanto, não apenas despende-se tempo e esforço consideráveis para criar um programa de simulação, mas também é necessário modificar o programa quando se muda a operação.

Utilize a "**Definição do Sistema E/S**" para resolver este problema.

Esta função pode simular a operação de um dispositivo externo sem usar um programa de depuração.

A operação de um dispositivo externo pode ser facilmente definida ou alterada na janela de definição. Portanto, a criação/modificação de um programa convencional não é necessária.

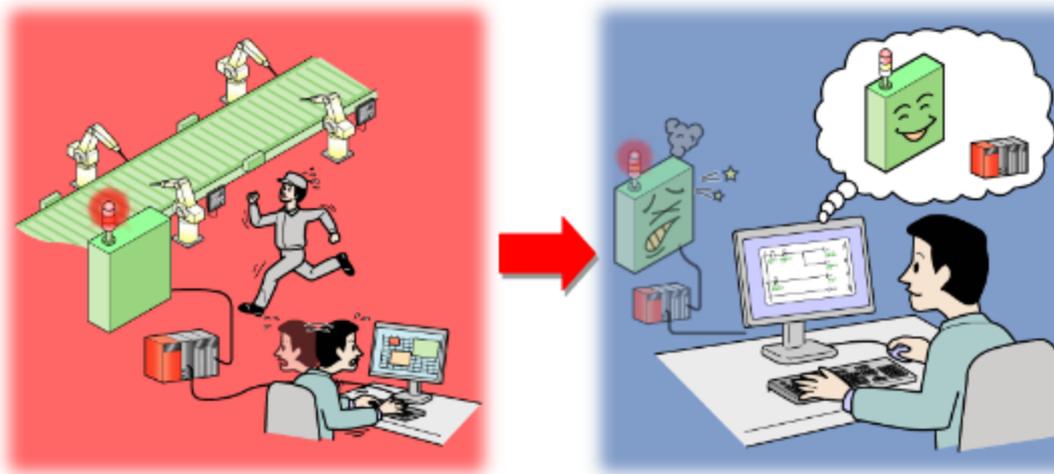
A operação de um dispositivo externo pode ser definida nas seguintes duas formas:

<**Definição dos valores do dispositivo**>

O valor de dispositivo especificado pode ser alterado no momento definido pelo temporizador depois que forem satisfeitas as condições especificadas.

<**Definição no formato de diagrama temporal**>

Uma mudança de dispositivo que foi definida no formato do diagrama temporal especificado pode ser realizada quando forem satisfeitas as condições especificadas.



3.6.1

Inserir valores de dispositivo para a definição do sistema de E/S

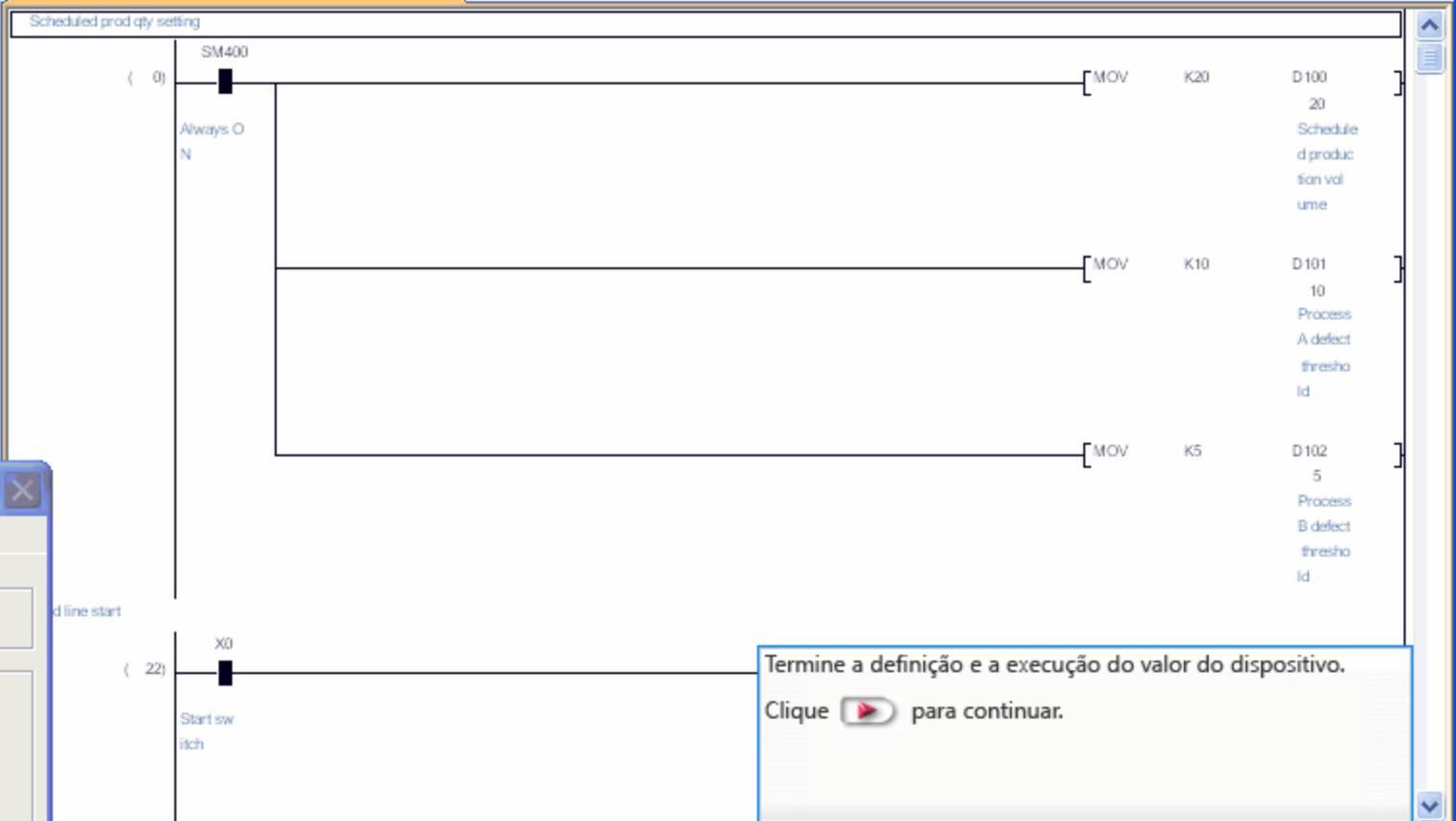


MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step]

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



Navigation [PRG]Monitor Executing M...



Simulation (0/ N)



3.6.2

Usar o formato do diagrama temporal para definição do sistema de E/S

Qualidade

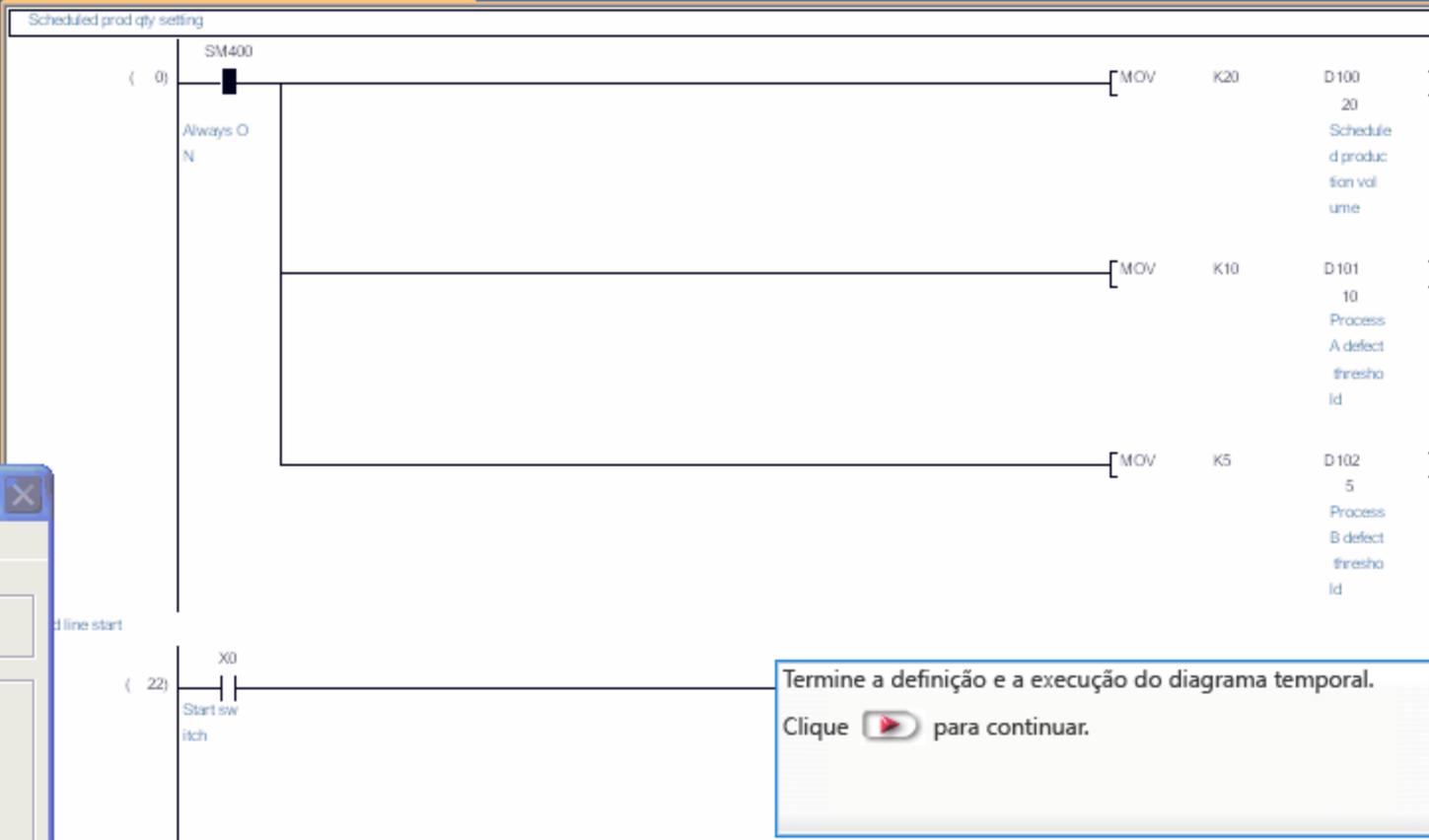
MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

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

Navigation [PRG]Monitor Executing M...

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Type



Termine a definição e a execução do diagrama temporal.

Clique  para continuar.

ish

Simple

Q03UDE

Simulation

(0/ N)



Capítulo 4 Gerenciamento do projeto e medidas de segurança

Etapas de aprendizagem no Capítulo 4

No Capítulo 4, você vai aprender sobre as funções usadas para o gerenciamento do projeto e as medidas de segurança.

- 4.1 Evitar dispersão de conhecimento e modificação não autorizada de programas
 - 4.1.1 Limitar os dados acessíveis por cada usuário
- 4.2 Backup do projeto e gerenciamento de versões
- 4.3 Comparação de programas salvos em controlador programável e computador pessoal

4.1 Evitar dispersão de conhecimento e modificação não autorizada de programas



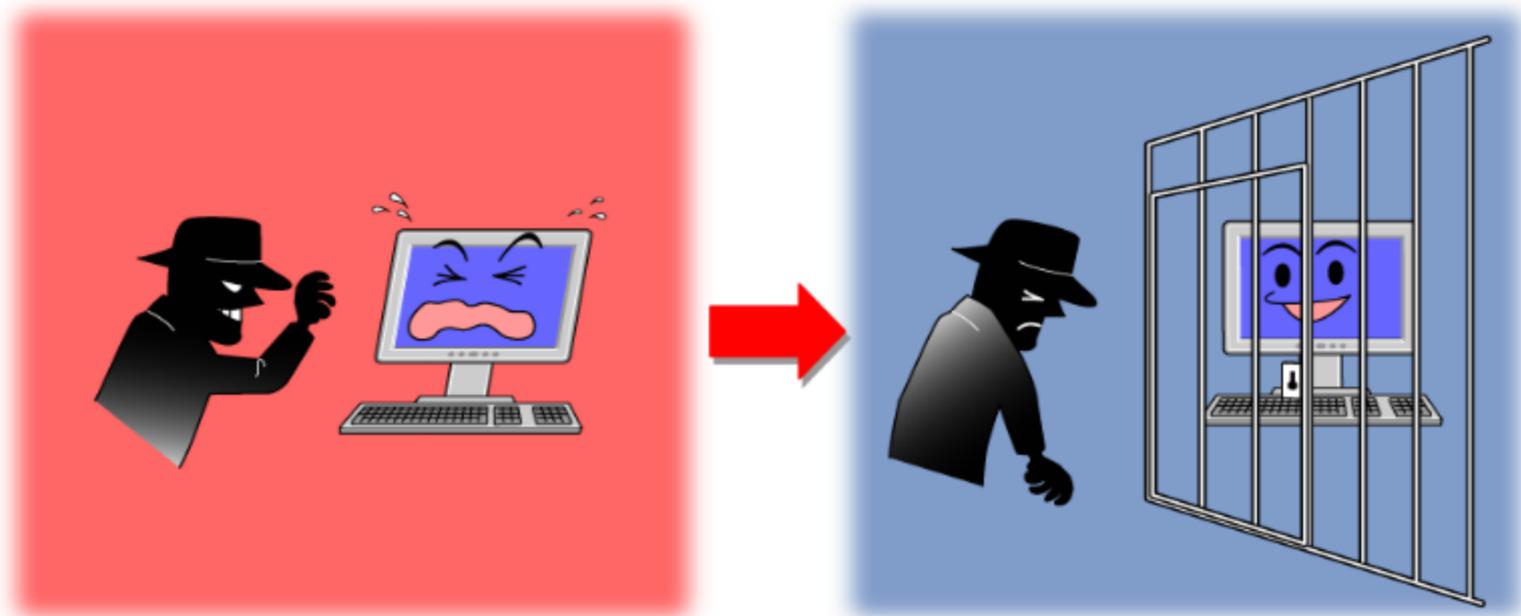
O programa de sequências inclui dados e conhecimento estrategicamente importantes. A dispersão de conhecimento e de dados do programa para o exterior poderia ter um efeito devastador nos negócios. A modificação não autorizada do programa pode levar a problemas de produção como parar o sistema.

Utilize a "**Segurança**" para resolver esses problemas.

Esta função limita os usuários que podem acessar cada projeto com proteção por senha.

Ele também pode limitar o intervalo de dados ou de funções que cada usuário pode acessar ou operar.

Assim, a função impede que usuários não autorizados naveguem ou editem programas.



4.1.1

Limitar os dados acessíveis por cada usuário



Um programa de sequências em grande escala é frequentemente desenvolvido por dois ou mais programadores que compartilham o trabalho.

Em caso de desenvolvimento em equipe, o intervalo de dados acessíveis e funções disponíveis deve ser gerenciado corretamente de acordo com a range de trabalho e o nível de habilidades de cada programador e a confidencialidade dos dados manipulados por cada programador.

Este gerenciamento de acesso pode ser implementado definindo **níveis de acesso** de segurança.

<Nível de acesso>

Podem ser definidos privilégios de operação de dados incluídos no projeto para cada usuário.

Os cinco níveis de acesso a seguir podem ser definidos.

Nível de acesso		Autoridade de operação	
Alta Baixa	Administradores	Nível de administrador	Autorizados a utilizarem todas as funções.
	Desenvolvedores (Nível3)	Nível de desenvolvedor	Definições de segurança, acesso de dados e algumas operações são restritas.
	Desenvolvedores (Nível2)		
	Desenvolvedores (Nível1)		
	Usuários	Nível de operador	Está disponível apenas a navegação pelos dados do projeto. Não é possível ler a partir da CPU do PLC.

4.1.1

Limitar os dados acessíveis por cada usuário



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Read MAIN (Read Only) 194 Step]

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



Navigation [PRG]Read MAIN (Read Onl...

Project

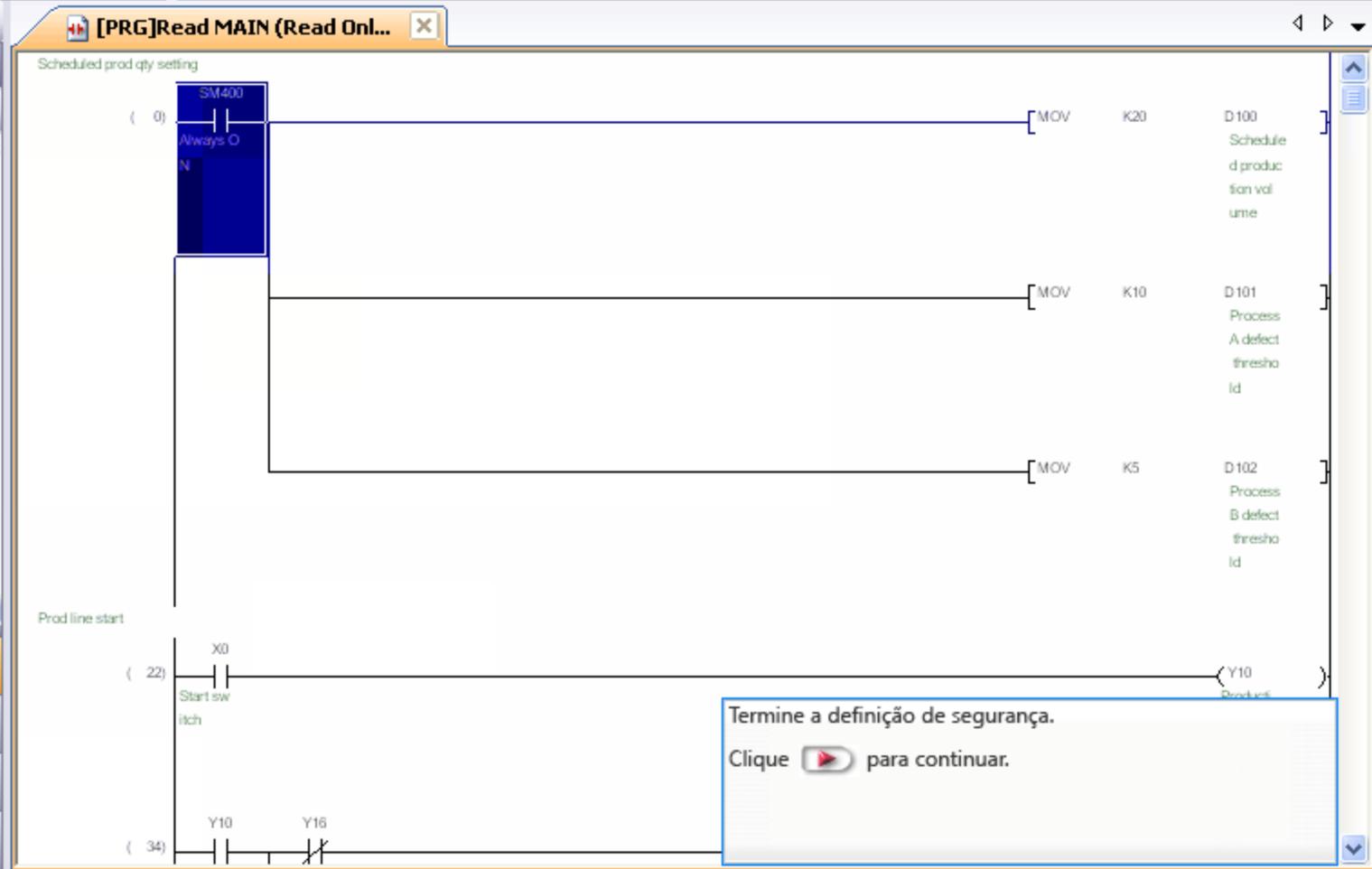
- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value



Project

User Library

Connection Destination



Termine a definição de segurança.
 Clique para continuar.

English Simple User Q03UDE Host Station (15 N...



4.2

Backup do projeto e gerenciamento de versões



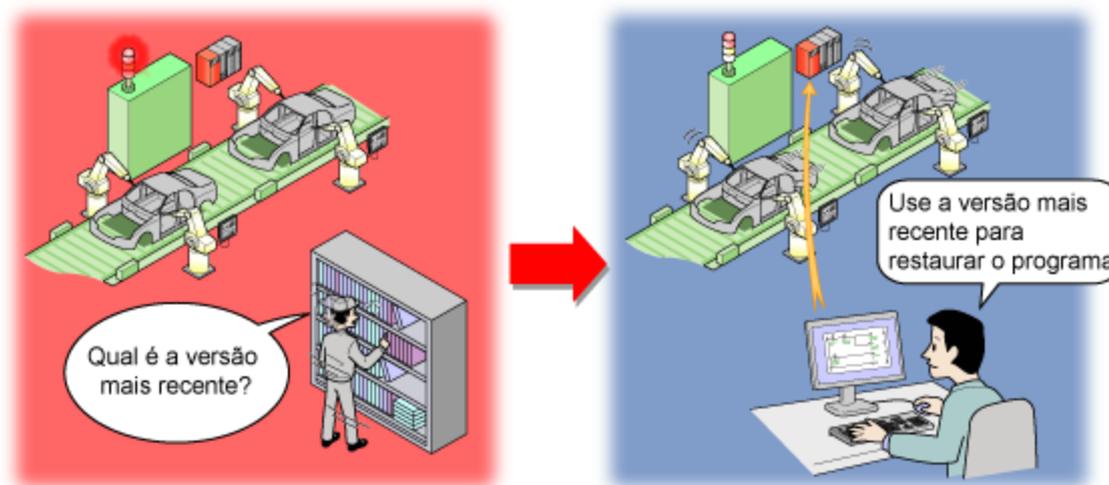
Um programa importante poderia ser perdido devido à falha da CPU do controlador programável. Se um programa de backup não estiver disponível, há pouca esperança de uma recuperação rápida. Mesmo se um programa de backup estiver disponível, seria difícil determinar se a versão é a mesma que o programa perdeu, a menos que o gerenciamento de versões esteja protegido. Para estar preparado para circunstâncias inesperadas, é necessário realizar um backup periódico e garantir o gerenciamento de versões.

Utilize **"Alterar histórico"** para resolver esses problemas.

Esta função pode registrar até 100 conjuntos de alteração de históricos (número de histórico, data/hora, usuário, título, comentário) do projeto.

Os dados do projeto no momento da gravação também têm seu backup realizado ao mesmo tempo.

O gerenciamento da versão assegurado pela função alterar histórico permite que você restaure programas perdidos, verifique as versões do programa e, assim, implemente uma recuperação rápida em caso de circunstâncias inesperadas.





4.2 Backup do projeto e gerenciamento de versões

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control (Revision No.3 : Revision B) - [[PRG]Write MAIN 194 Step]

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

Navigation [PRG]Write MAIN 194 Step

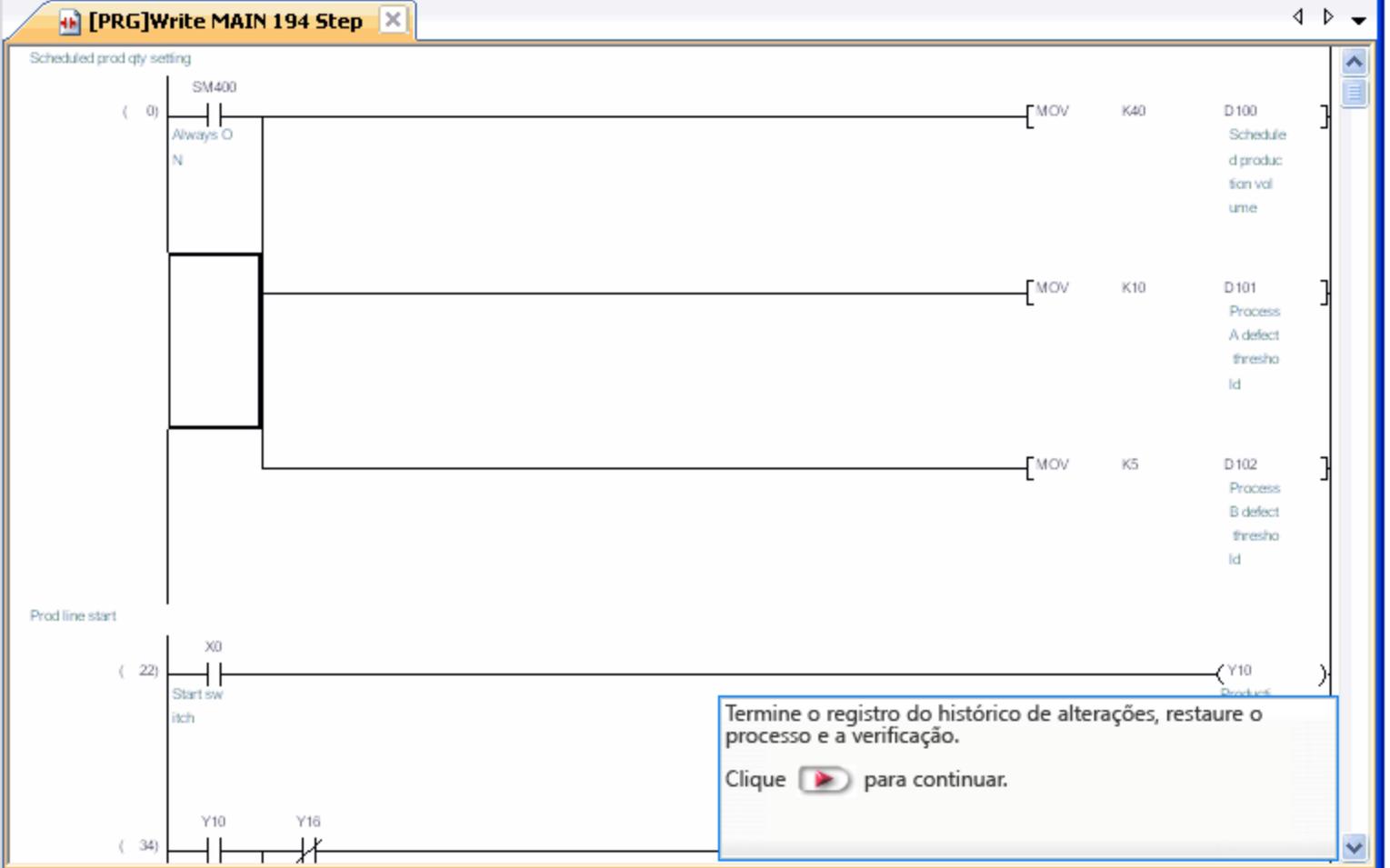
Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool
- Structured Data Types
- Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination



Termine o registro do histórico de alterações, restaure o processo e a verificação.

Clique para continuar.

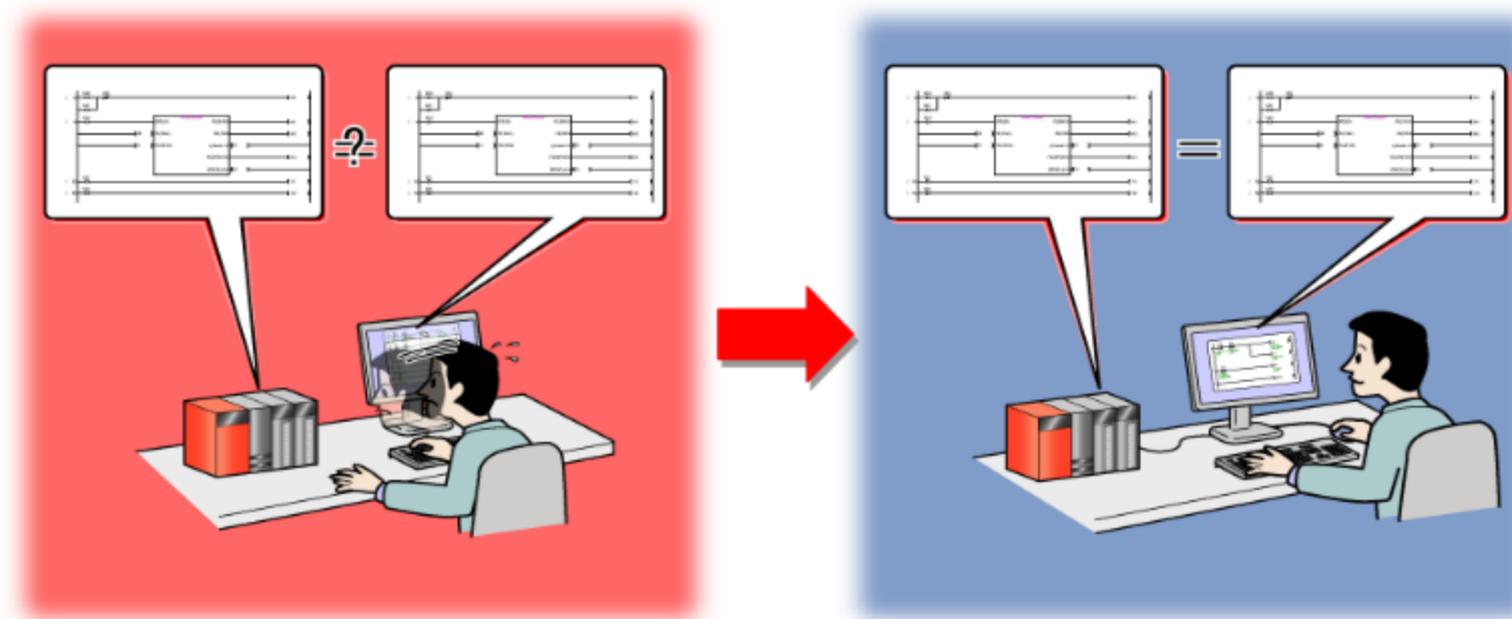
Navigation icons: Home, Play, Stop, Previous, Next

4.3 Comparação de programas salvos no controlador programável e computador pessoal



Normalmente, os programas são salvos em um PC no ambiente de desenvolvimento e também são gravados no PLC. Esses dois programas não são sempre os mesmos. Executar apenas uma verificação visual para ver se eles são os mesmos pode resultar em erros.

Utilize **"Verificar com PLC"** para resolver este problema. Esta função pode verificar se o programa aberto pelo GX Works2 corresponde ao programa escrito no PLC.



4.3 Comparação de programas salvos no controlador programável e computador pessoal

Gerenciamento

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Write MAIN 194 Step]

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



Navigation [PRG]Write MAIN 194 Step Verify Result [PLC Verify]

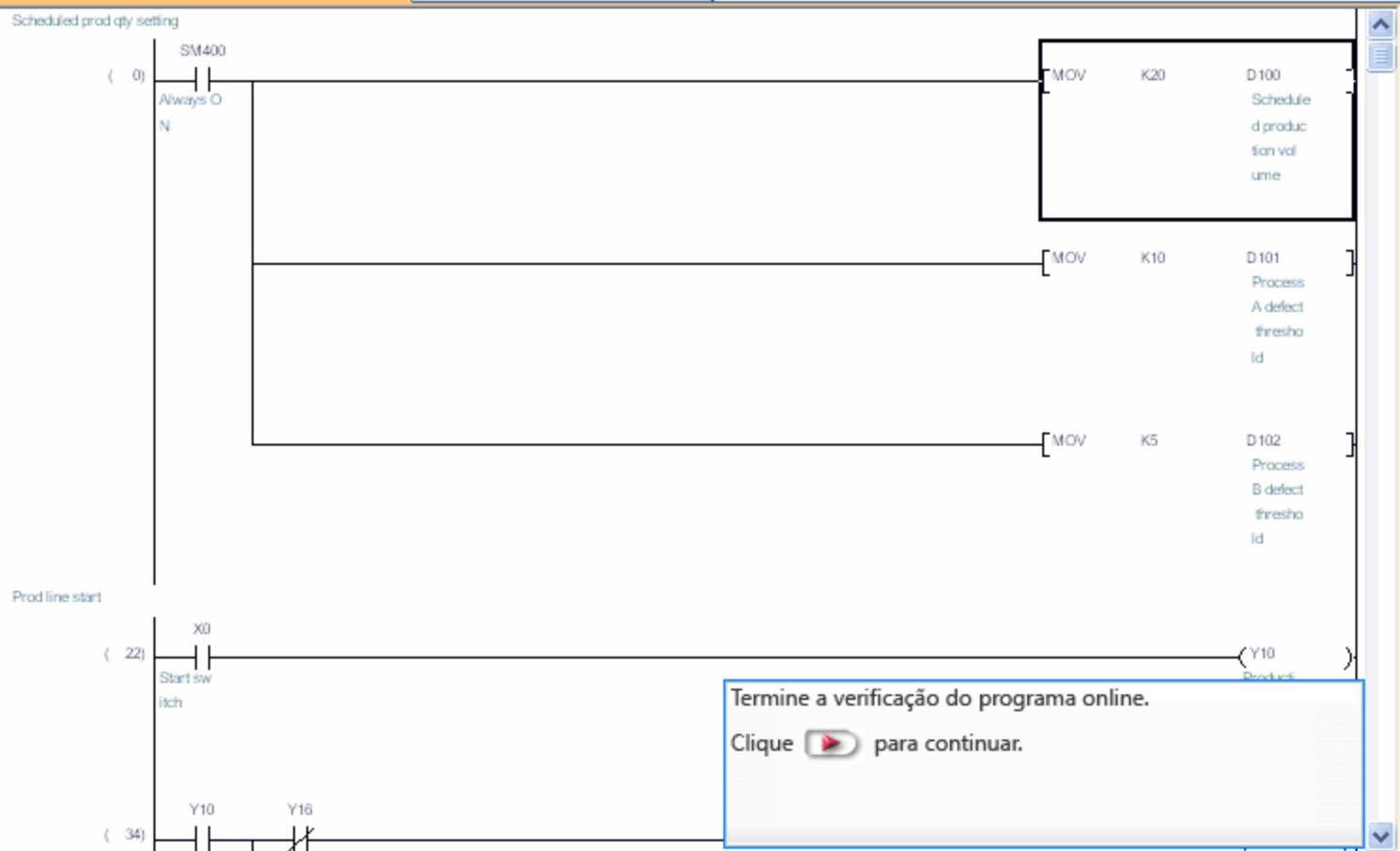
Project

- Parameter Prod line control s
 - Intelligent Function Module
 - Global Device Comment
 - Global Label
 - Program Setting
 - POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination



Termine a verificação do programa online.

Clique  para continuar.

English

Simple

Q03UDE

Host Station

(16 N)



Teste Teste final

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

O Teste Final é composto por 8 perguntas (8 itens).
Você pode fazer o teste final quantas vezes desejar.

Como é feita a pontuação do teste

Depois de selecionar a resposta, não se esqueça de clicar no botão **Gravar Pontuação**. Se você continuar sem clicar no botão Gravar Pontuação, sua resposta será perdida. (Considerada como uma pergunta não respondida.)

Resultados da pontuação

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

Respostas corretas: 2

Total de perguntas: 9

Porcentagem: 22%

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

Continuar

Rever

Repetir

- Clique no botão **Continuar** para sair do teste.
- Clique no botão **Rever** para revisar o teste. (Verificação das respostas corretas)
- Clique no botão **Repetir** para refazer o teste.

Teste**Teste final 1**

Qual das seguintes funções ativa a programação eficiente usando os ladder blocks usados repetidamente como componentes compartilháveis? (Escolha uma.)

- Texto estruturado em linha
- Label
- Function block

Gravar Pontuação

Voltar

Teste**Teste final 2**

Qual das seguintes funções pode criar programas fáceis de ler alterando os nomes dos dispositivos para nomes associados com sua aplicação? (Escolha uma.)

- Comentário do dispositivo
- Label
- Nota

Gravar Pontuação

Voltar

Teste**Teste final 3**

Qual das seguintes funções pode criar programas fáceis de ler fornecendo informações sobre processamento para cada ladder block? (Escolha uma.)

- Comentário do dispositivo
- Statement de linha
- Nota

Gravar Pontuação

Voltar

Teste**Teste final 4**

Qual das seguintes é a explicação correta para a função "Verify with PLC" (Verificar com PLC)? (Escolha uma.)

- Compara o programa sendo editado com um programa registrado no histórico de alterações.
- Compara o programa sendo editado com um programa selecionado salvo no computador.
- Compara o programa sendo editado com um programa escrito na CPU do PLC.

Gravar Pontuação

Voltar

Teste **Teste final 5**

Qual das seguintes é a explicação correta para a função "I/O System Setting" (Definição do sistema de E/S)?
(Escolha uma.)

- Simula a operação do equipamento externo de E/S no computador pessoal durante a depuração.
- Controla remotamente a operação do equipamento externo de E/S do computador pessoal durante a depuração.
- Simula a operação da CPU do PLC no computador pessoal durante a depuração.

Gravar Pontuação

Voltar

Teste**Teste final 6**

Qual das seguintes é a explicação correta para a função "Change history" (Alterar histórico)? (Escolha uma.)

- Registra a operação do GX Works2 passo a passo para que possa ser livremente restaurada depois.
- Registra as informações do histórico e faz o backup do projeto para ativar a verificação e a restauração depois.

Gravar Pontuação

Voltar

Teste**Teste final 7**

Qual das seguintes funções pode ser usada durante a depuração para alterar apenas os valores do dispositivo sem modificar o programa? (Escolha uma.)

- Execução de interrupção
- Definição do sistema de E/S
- Teste de dispositivo condicionado executável

Gravar Pontuação

Voltar

Teste**Teste final 8**

Suponha que o projeto inclui dois programas, A e B, e que você usa a função "label".
Qual dos seguintes tipos de labels pode ser acessado pelo programa B? (Escolha dois.)

- Label global
- Label local do programa A
- Label local do programa B

Teste**Pontuação do teste**

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

Respostas corretas: **0**

Total de perguntas: **8**

Porcentagem: **0%**

Você não passou no teste.

Você concluiu o curso **PLC GX Works2 Avançado**.

Muito obrigado por fazer este curso.

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

Você pode revisar o curso quantas vezes quiser.

Rever

Fechar