

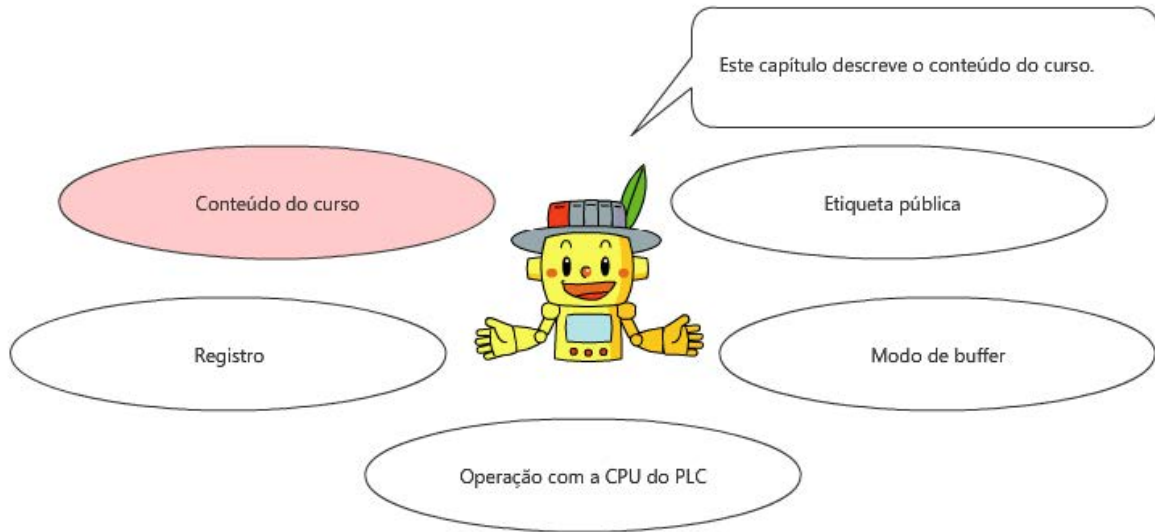
Controlador de sistema de servo

Introdução ao Módulo de Movimento da Série MELSEC iQ-R (RD78G(H) Controle de posicionamento)

Este treinamento destina-se aos participantes que querem estabelecer um sistema de motion controller utilizando o módulo de movimento da Série MELSEC iQ-R pela primeira vez.

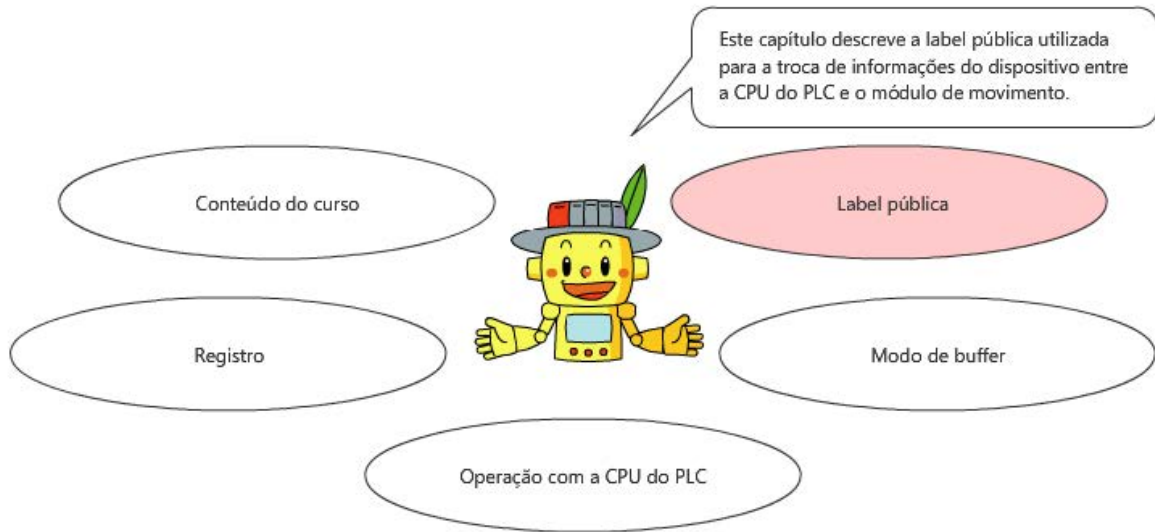
Clique no botão Avançar, no canto superior direito, para mudar para a página seguinte.

Este curso destina-se aos participantes que querem adquirir conhecimentos sobre o controle de posicionamento do sistema de motion controller utilizando o módulo de movimento da Série MELSEC iQ-R.



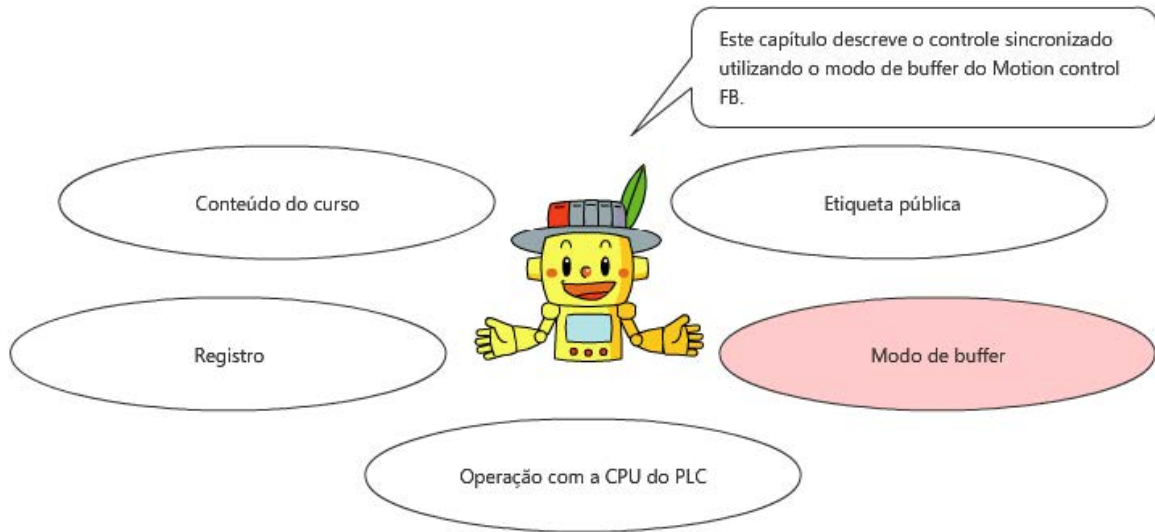
Este curso é uma continuação do curso Introdução ao Módulo de Movimento da Série MELSEC iQ-R (RD78G(H) Inicialização). Você deve concluir o curso de Inicialização antes de concluir este curso.

Este curso destina-se aos participantes que querem adquirir conhecimentos sobre o controle de posicionamento do sistema de motion controller utilizando o módulo de movimento da Série MELSEC iQ-R.



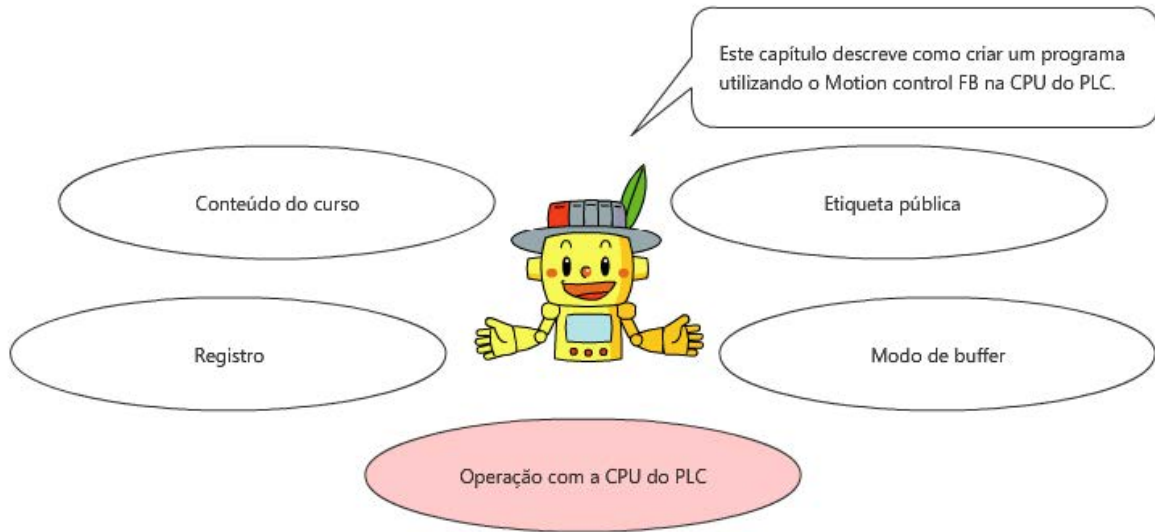
Este curso é uma continuação do curso Introdução ao Módulo de Movimento da Série MELSEC iQ-R (RD78G(H) Inicialização). Você deve concluir o curso de Inicialização antes de concluir este curso.

Este curso destina-se aos participantes que querem adquirir conhecimentos sobre o controle de posicionamento do sistema de motion controller utilizando o módulo de movimento da Série MELSEC iQ-R.



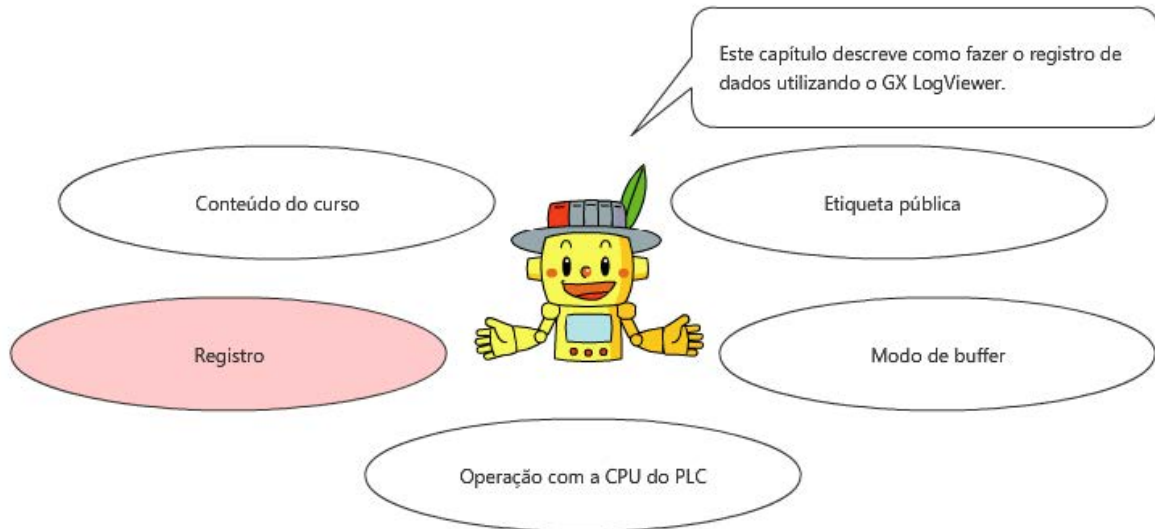
Este curso é uma continuação do curso Introdução ao Módulo de Movimento da Série MELSEC iQ-R (RD78G(H) Inicialização). Você deve concluir o curso de Inicialização antes de concluir este curso.

Este curso destina-se aos participantes que querem adquirir conhecimentos sobre o controle de posicionamento do sistema de motion controller utilizando o módulo de movimento da Série MELSEC iQ-R.



Este curso é uma continuação do curso Introdução ao Módulo de Movimento da Série MELSEC iQ-R (RD78G(H) Inicialização). Você deve concluir o curso de Inicialização antes de concluir este curso.

Este curso destina-se aos participantes que querem adquirir conhecimentos sobre o controle de posicionamento do sistema de motion controller utilizando o módulo de movimento da Série MELSEC iQ-R.



Este curso é uma continuação do curso Introdução ao Módulo de Movimento da Série MELSEC iQ-R (RD78G(H) Inicialização). Você deve concluir o curso de Inicialização antes de concluir este curso.

O conteúdo deste curso é fornecido a seguir.
Recomendamos que você comece pelo Capítulo 1.

Capítulo 1 - Conteúdo do curso

Este capítulo descreve o conteúdo do curso.

Capítulo 2 - Label pública

Este capítulo descreve a label pública utilizada para a troca de informações do dispositivo entre a CPU do PLC e o módulo de movimento.

Capítulo 3 - Modo de buffer

Este capítulo descreve o controle sincronizado utilizando o modo de buffer do Motion control FB.

Capítulo 4 - Operação com a CPU do PLC





Este capítulo descreve como criar um programa utilizando o Motion control FB na CPU do PLC.

Capítulo 5 - Registro de dados

Este capítulo descreve como fazer o registro de dados utilizando o GX LogViewer.

Teste final

4 seções no total (7 perguntas)

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

■Precauções de segurança

Ao utilizar os produtos reais para fins de aprendizado, leia atentamente as "Precauções de segurança" descritas no respectivo manual, e preste atenção à segurança e ao uso adequado.

■Precauções neste curso

As imagens das telas apresentadas no curso podem ser diferentes das imagens de seu software, dependendo da versão. Este curso utiliza as seguintes versões de software.


Para obter a última versão de cada software, verifique o site da Mitsubishi Electric FA.

MELSOFT GX Works3	Ver.1.066U	Motion Control Setting function	Ver.1.012N
GX LogViewer	Ver.1.106K		
MELSOFT MR Configurator2	Ver.1.110Q ou superior		

A versão de firmware da CPU do PLC deve ser 44 ou superior (46 ou superior para RD78GH).

A versão de firmware do módulo de movimento deve ser 10 ou superior.

Para atualizar a versão de firmware, consulte o site da MITSUBISHI ELECTRIC FA ou o manual de configuração do módulo.

O ícone  indica o manual de referência.

O conteúdo dos manuais descritos neste curso corresponde às seguintes versões.

Se as versões forem diferentes, o local da descrição e o conteúdo podem ser ligeiramente diferentes.

Nome do manual	N ° do manual	Versão
MELSEC iQ-R Motion Module User's Manual (Startup)	IB-0300406	E
MELSEC iQ-R Motion Module User's Manual (Application)	IB-0300411	E
MELSEC iQ-R Motion Module User's Manual (Network)	IB-0300426	E
MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks)	IB-0300431	E
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)	IB-0300533	C
MELSEC iQ-R Structured Text (ST) Programming Guide Book	SH-081483	F
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)	SH-081266	Z
MELSEC iQ-R CPU Module User's Manual (Application)	SH-081264	AK

A seção a seguir mostra a visão geral do curso.

Capítulo 1 Conteúdo do curso

Este capítulo descreve o conteúdo do curso.



Capítulo 2 Label pública

Este capítulo descreve a label pública utilizada para a troca de informações do dispositivo entre a CPU do PLC e o módulo de movimento.



Capítulo 3 Modo de buffer

Este capítulo descreve o controle sincronizado utilizando o modo de buffer do Motion control FB.



Capítulo 4 Operação com a CPU do PLC

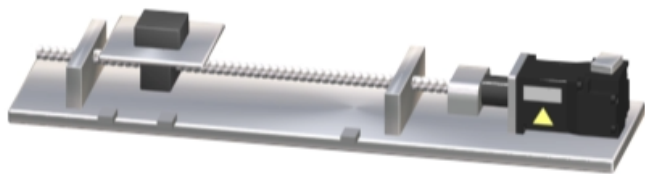
Este capítulo descreve como criar um programa utilizando o Motion control FB na CPU do PLC.



Capítulo 5 Registro de dados

Este capítulo descreve como fazer o registro de dados utilizando o GX LogViewer para verificar a operação do módulo de movimento.

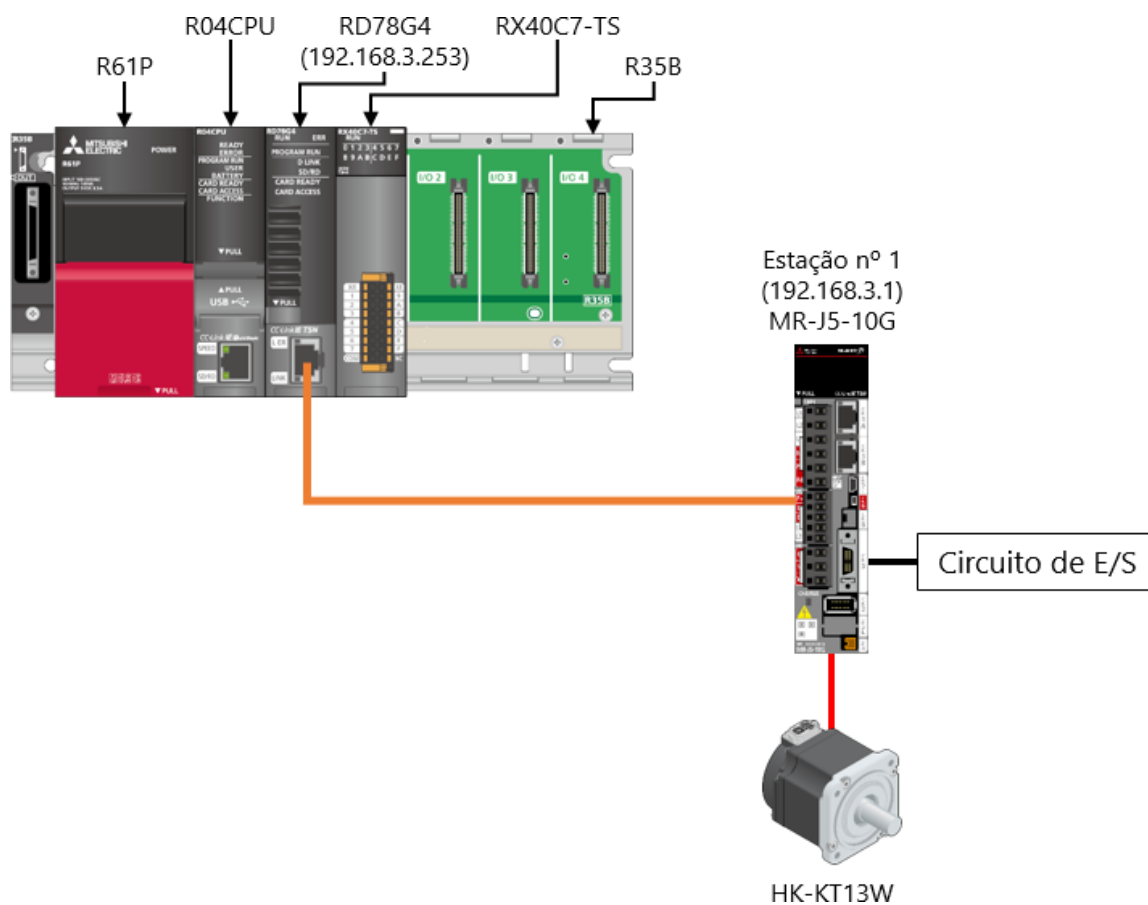
Este curso utiliza o mesmo mecanismo de fuso de esferas de eixo único utilizado no curso de Inicialização.



A configuração do sistema de destino é a seguinte.

Remova o módulo de entrada remota do sistema usado no curso de Inicialização, e adicione o módulo de entrada RX40C7-TS ao slot 1 da unidade base do controlador programável.

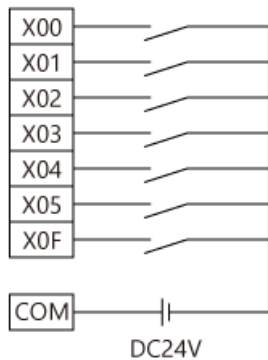
O nº da estação do servo amplificador MR-J5-10G foi alterado para 1, e o endereço IP foi alterado para 192.168.3.1.



A conexão elétrica da fonte de alimentação para o controlador programável e servo amplificador, e o método de conexão do servomotor, são os mesmos descritos no curso de Inicialização.

A seção a seguir mostra a conexão elétrica do circuito externo do módulo de entrada.

RX40C7-TS



(Nota)

X00: Servo desativado (OFF)

X01: JOG com rotação de avanço

X02: JOG com rotação de recuo

X03: Retorno à posição inicial (método do dog de proximidade)

X04: Controle de posicionamento (para o capítulo 2)

X05: Controle de posicionamento contínuo (para o capítulo 3)

X0F: Redefinição do erro

(Nota) Uma vez que o nº de I/O do RX40C7-TS é 0020H, X20 a X25 e X2F são utilizados no programa.

Quando o módulo de movimento é controlado pelo módulo de entrada do controlador programável, como no sistema usado no curso descrito no capítulo 1, a CPU do PLC e o módulo de movimento devem trocar informações sobre o dispositivo.

Existem dois métodos, descritos a seguir.

1. Utilizar labels públicas.
2. Utilizar a memória de buffer do módulo de movimento.

Este capítulo descreve como trocar dados utilizando labels públicas.

Baixe o exemplo de programa que será usado neste capítulo e no capítulo 3, clicando no link abaixo.

[RD78GBasic2_sample1.zip \(1.34MB\)](#)

[Ponto]

Ao utilizar a memória de buffer, copie os dados a serem trocados para a área do usuário (U□\G11478000 to G11997999).

(Exemplo de programa)

<CPU do PLC>

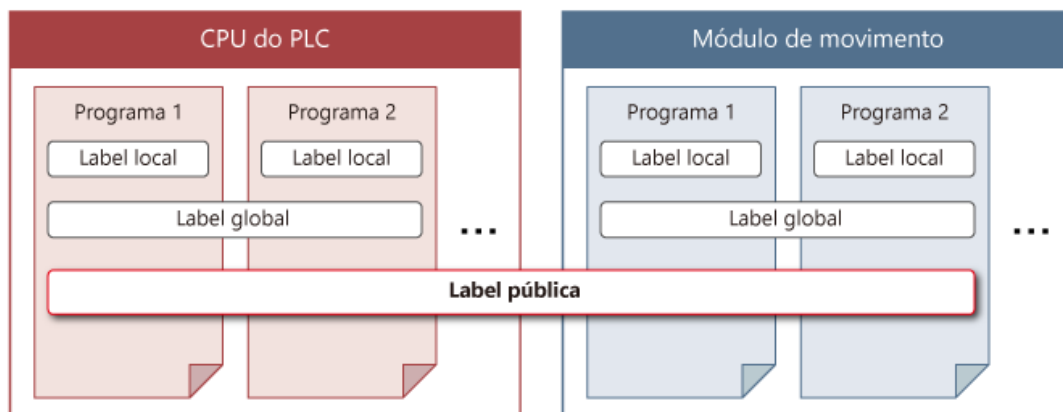
Início de posicionamento



<Módulo de movimento>

```
MC_MoveAbsolute_1(  
  Execute:= G11478000.0 ,  
  :  
  :  
);
```

A label pública é uma label compartilhada que pode ser usada no módulo de movimento e também na CPU do PLC. A seção a seguir mostra as áreas onde se pode aplicar a label local, label global e label pública.



(1) Como registrar labels públicas

Registre as labels públicas a partir das labels globais do módulo de movimento.

Certifique-se de que a coluna "Public Label" fique visível no editor de labels globais da tela Motion Module Setting Function.

Defina as labels a serem registradas como labels públicas como "Enabled".

Essa ação ativa a coluna "Motion Control Attribute".

Selecione se cada label deverá ser lida ou gravada da/para a CPU do PLC.

	Label Name	Data Type	Class	Initial	Constant	Japanese	English(Display Target)	Chinese	Remark	Public Label	Motion Control Attribute
1	G_bSVONCMD	Bit	VAR_GLOBAL				Servo ON			Enabled	WRITE (=> Motion)
2	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL				JOG Velocity			Enabled	WRITE (=> Motion)
3	G_bJogFwd	Bit	VAR_GLOBAL				JOG Forward			Enabled	WRITE (=> Motion)
4	G_bJogBwd	Bit	VAR_GLOBAL				JOG Backward			Enabled	WRITE (=> Motion)
5	G_bJogBusy	Bit	VAR_GLOBAL				JOG Busy			Enabled	READ (Motion =>)
6	G_lePosition0	FLOAT [Double Precision]	VAR_GLOBAL				Position0 Address			Disabled	-
7	G_bHomingCMD	Bit	VAR_GLOBAL				Homing Command			Enabled	WRITE (=> Motion)
8	G_bHomingDone	Bit	VAR_GLOBAL				Homing Done			Enabled	READ (Motion =>)
9	G_bHomingReq	Bit	VAR_GLOBAL				Homing Request			Enabled	READ (Motion =>)
10	G_bPosCMD	Bit	VAR_GLOBAL				Positioning Command			Enabled	WRITE (=> Motion)
11	G_bPosDone	Bit	VAR_GLOBAL				Positioning Done			Enabled	READ (Motion =>)
12	G_bPosReq	Bit	VAR_GLOBAL				Positioning Start Request			Enabled	READ (Motion =>)
13	G_bErrorReset	Bit	VAR_GLOBAL				Error Reset			Enabled	WRITE (=> Motion)
14	G_bContPosCMD	Bit	VAR_GLOBAL				Continuous Positioning Command			Enabled	WRITE (=> Motion)
15	G_bContPosReq	Bit	VAR_GLOBAL				Continuous Positioning Start Request			Enabled	WRITE (=> Motion)
16	G_bContPosDone	Bit	VAR_GLOBAL				Continuous Positioning Done			Enabled	READ (Motion =>)
17											

[Ponto]

Se a coluna da label pública não estiver visível, navegue para a direita na tabela.

(2) Tipos de dados que podem ser registrados como labels públicas

A tabela a seguir mostra os tipos de dados que podem ser registrados como labels públicas.

Tipo variável	Tipo	Seleção do array	Definição da label pública	Observações
Label global	Tipo simples	Não	○	<p>Não é possível atribuir definições para as seguintes labels e classe.</p> <p>■Label</p> <ul style="list-style-type: none"> • Label do tipo string • Label do tipo temporizador • Label do tipo contador • Label do tipo contador longa • Label do tipo temporizador retentiva • Label do tipo temporizador retentiva longa • Label do tipo temporizador longa <p>■Classe</p> <ul style="list-style-type: none"> • Classe VAR_GLOBAL_CONSTANT
		Sim	△(Nota 1,2)	
	Tipo de dados estruturados	Não	△(Nota 3)	
		Sim	△(Nota 1,2,4,5)	
	FB (Incluindo Motion control FBs)	Não	×	
		Sim	×	
Programa	-	-	×	
Label local de bloco de programa	-	-	×	
Tipo de dados estruturados	-	-	△(Nota 3,5)	
Dados estruturados do Motion control FB	-	-	△(Nota 6,7)	

(Nota)

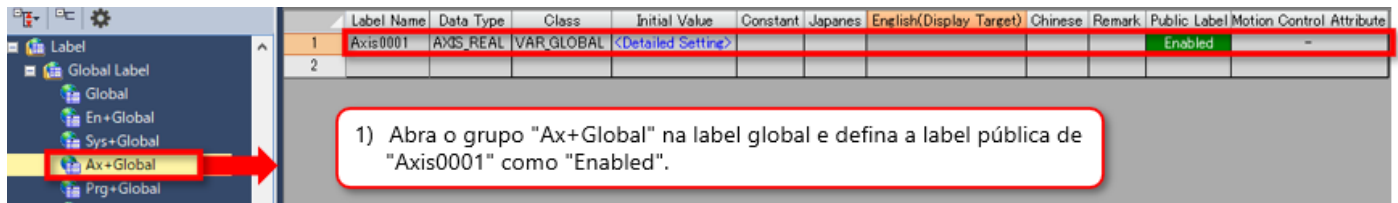
1. A definição da label pública não pode ser configurada para cada elemento de um array.
2. Quando se utiliza um array do tipo bit, a label pública não pode ser definida como "Enabled". (Nos dados estruturados, apenas o membro correspondente não pode ser definido como "Enabled".)
3. Quando se utiliza o tipo string como um membro do tipo de dados estruturados, o membro não pode ser definido como "Enabled".
4. Apenas os dados estruturados com no máximo quatro camadas podem se tornar públicos.
5. Quando se utiliza um array dos dados estruturados como membro dos dados estruturados, o membro não pode ser definido como "Enabled".
6. Ele pode ser usado no programa do PLCopen Motion control FB pelo módulo da CPU.
7. Quando se utiliza o tipo string em dados estruturados de um Motion control FB, o tipo dos dados estruturados do Motion control FB em si não pode ser definido.

(3) Como registrar dados estruturados como labels públicas

Para definir os membros de um tipo de dados estruturados preparado no sistema, como os dados do monitor do eixo, como label pública, registre as labels públicas pela camada dos dados estruturados, como apresentado abaixo.

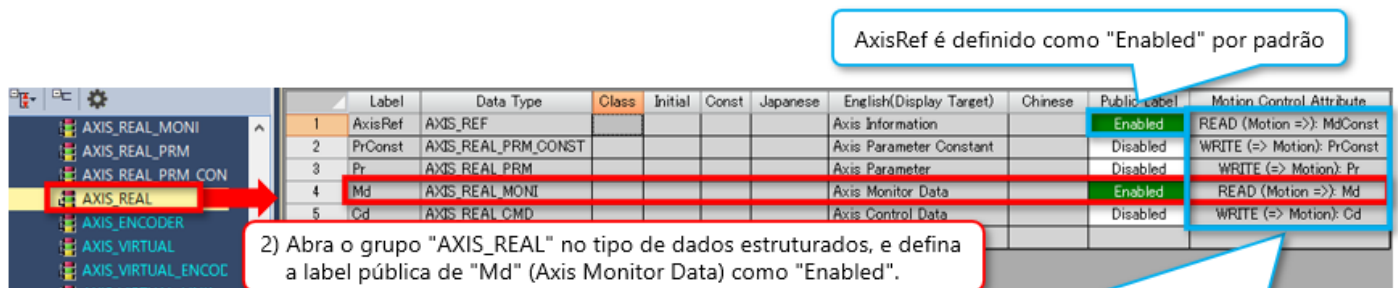
Este curso descreve como registrar os parâmetros Definir posição (SetPosition) e Definir velocidade (SetVelocity) e os dados do monitor (Md) do eixo de acionamento real (Axis_Real) como labels públicas.

[Como definir AxisName.Md.SetPosition (Posição atual do comando) e AxisName.Md.SetVelocity (Velocidade atual do comando) na label pública]



Label Name	Data Type	Class	Initial Value	Constant	Japanese	English(Display Target)	Chinese	Remark	Public Label	Motion Control Attribute
1	Axis0001	AXIS_REAL	VAR_GLOBAL	<Detailed Setting>					Enabled	-
2										

1) Abra o grupo "Ax+Global" na label global e defina a label pública de "Axis0001" como "Enabled".

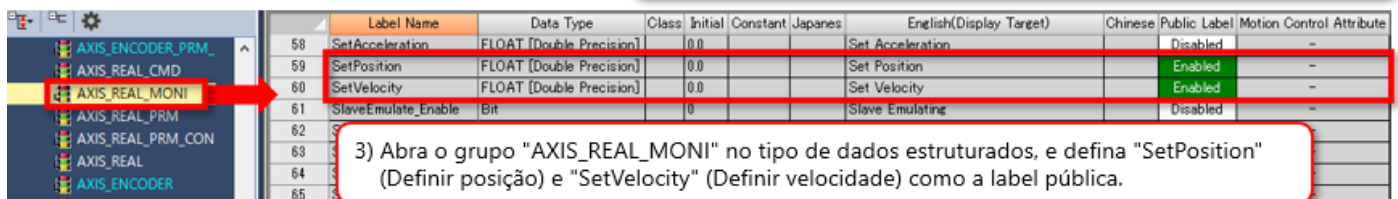


Label	Data Type	Class	Initial	Const	Japanese	English(Display Target)	Chinese	Public Label	Motion Control Attribute
1	AxisRef	AXIS_REF				Axis Information		Enabled	READ (Motion =>): MdConst
2	PrConst	AXIS_REAL_PRM_CONST				Axis Parameter Constant		Disabled	WRITE (=> Motion): PrConst
3	Pr	AXIS_REAL_PRM				Axis Parameter		Disabled	WRITE (=> Motion): Pr
4	Md	AXIS_REAL_MONI				Axis Monitor Data		Enabled	READ (Motion =>): Md
5	Cd	AXIS_REAL_CMD				Axis Control Data		Disabled	WRITE (=> Motion): Cd

AxisRef é definido como "Enabled" por padrão

2) Abra o grupo "AXIS_REAL" no tipo de dados estruturados, e defina a label pública de "Md" (Axis Monitor Data) como "Enabled".

O atributo do motion controller do tipo de dados estruturados é fixado.



Label Name	Data Type	Class	Initial	Constant	Japanese	English(Display Target)	Chinese	Public Label	Motion Control Attribute
58	SetAcceleration	FLOAT [Double Precision]	0.0			Set Acceleration		Disabled	-
59	SetPosition	FLOAT [Double Precision]	0.0			Set Position		Enabled	-
60	SetVelocity	FLOAT [Double Precision]	0.0			Set Velocity		Enabled	-
61	SlaveEmulate_Enable	Bit	0			Slave Emulating		Disabled	-
62									
63									
64									
65									

3) Abra o grupo "AXIS_REAL_MONI" no tipo de dados estruturados, e defina "SetPosition" (Definir posição) e "SetVelocity" (Definir velocidade) como a label pública.

(4) Refletindo as labels públicas

Selecione [Convert] → [Rebuild All] no menu.

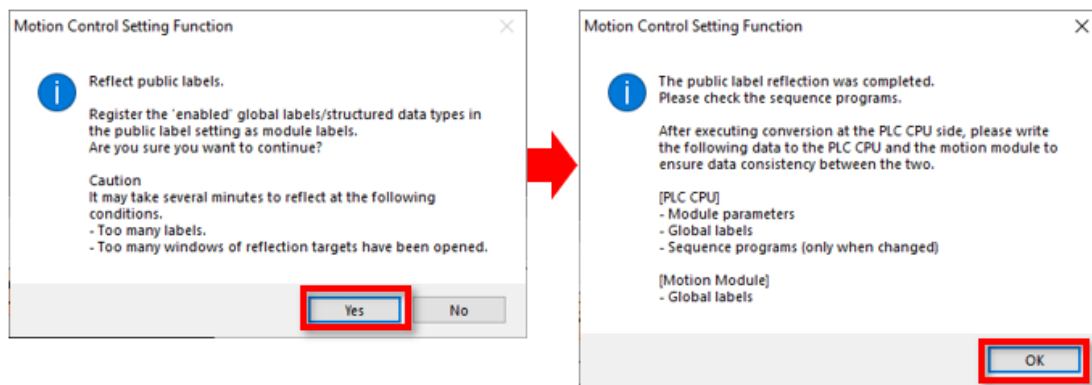
A capacidade livre da label pública é exibida na forma de informações na janela de saída.

No.	Result	Data Name	Category	Content	Error Code
1	Information	Public Label	Free Volume	99.88[%] (32728 [Word] = 32768 [Word] - (Global: 40 [Word]))	-

Quando o processo de reconstrução total estiver concluído com sucesso, selecione [Convert] → [Reflect Public Labels] no menu.

Clique em [Yes] na mensagem que aparece a seguir.

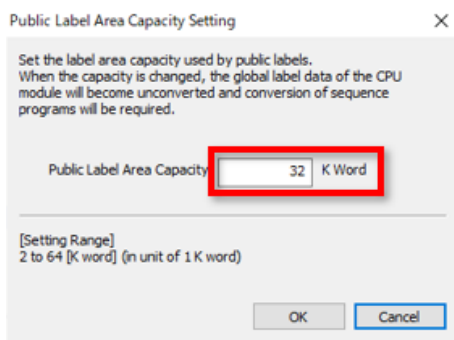
Quando aparecer uma mensagem indicando que as labels públicas foram refletidas com sucesso, clique no botão [OK].



(Nota) A capacidade de memória que pode ser usada para registrar labels públicas é de 32.000 palavras, por padrão.

Essa capacidade pode ser aumentada para até 64.000 palavras.

Para alterar a capacidade, defina o tamanho da memória em [Convert] → [Public Label Capacity Setting], no menu.



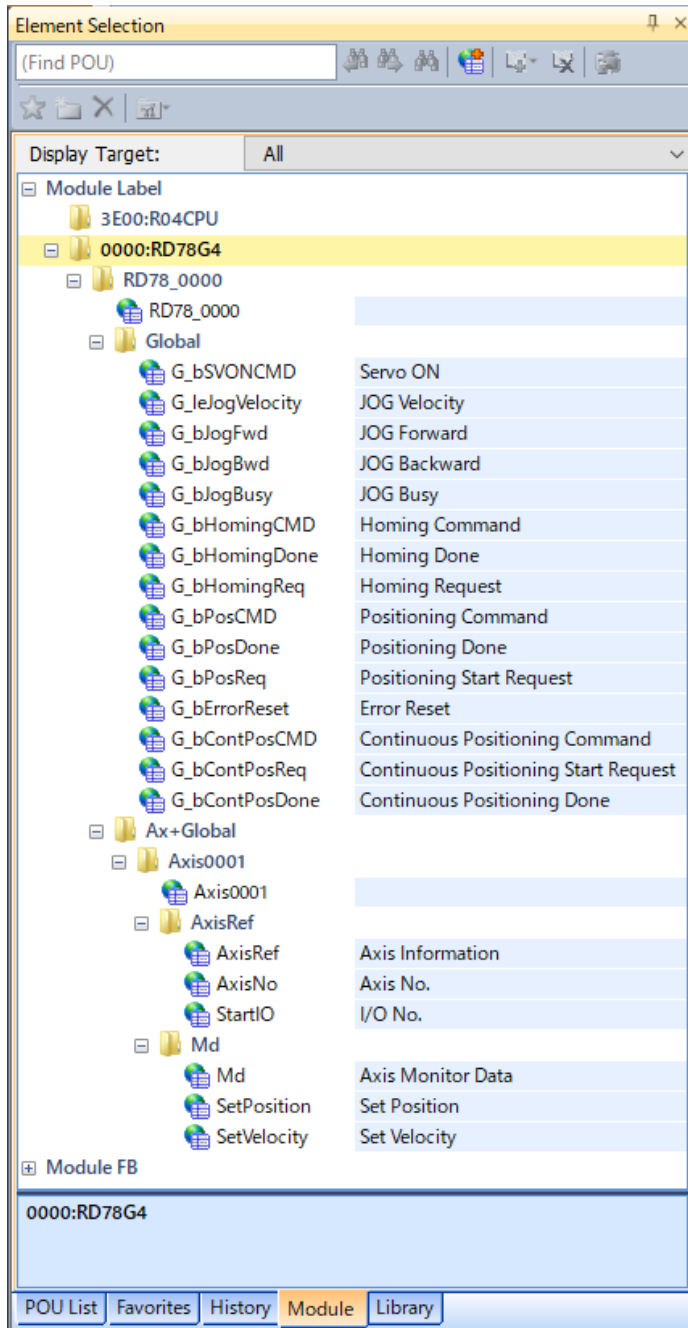
(5) Verificando as labels no lado da CPU do PLC

As labels públicas refletidas são registradas na label do módulo, no lado da CPU do PLC.

Selecione a label do módulo na janela Element Selection do GX Works3, e verifique se as labels públicas foram registradas sob [0000:RD78G4], em [Module Label].

Depois de alterar uma definição de label pública, sempre execute novamente "Reflect Public Labels".

Ao utilizar as labels públicas na CPU do PLC, reconstrua todos os programas.

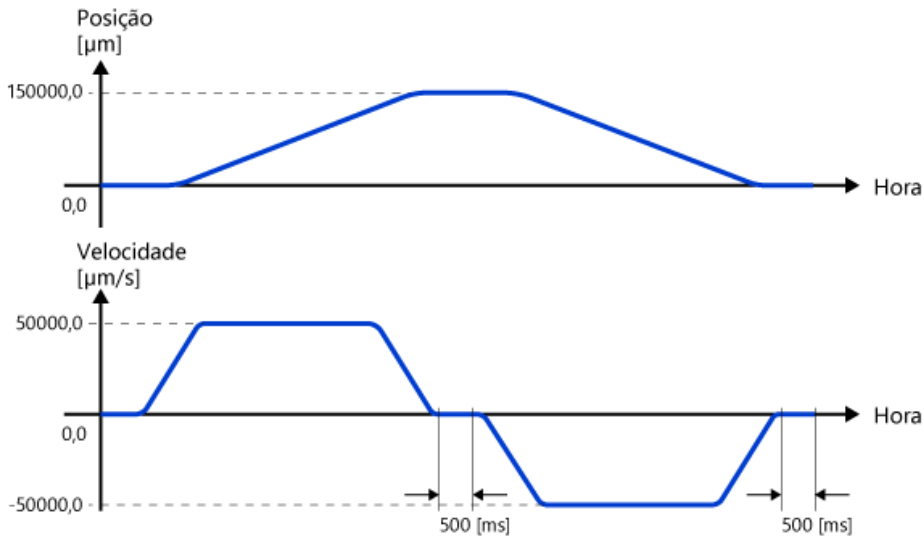


(1) Operação do exemplo de programa

Os sinais de entrada do exemplo de programa usado neste capítulo são atribuídos da seguinte forma.

Entrada	Operação
X20	Servo desligado (Nota)
X21	Operação JOG com rotação de avanço
X22	Operação JOG com rotação de recuo
X23	Retorno à posição inicial
X24	Controle de posicionamento
X25	Controle de posicionamento contínuo (Capítulo 3)

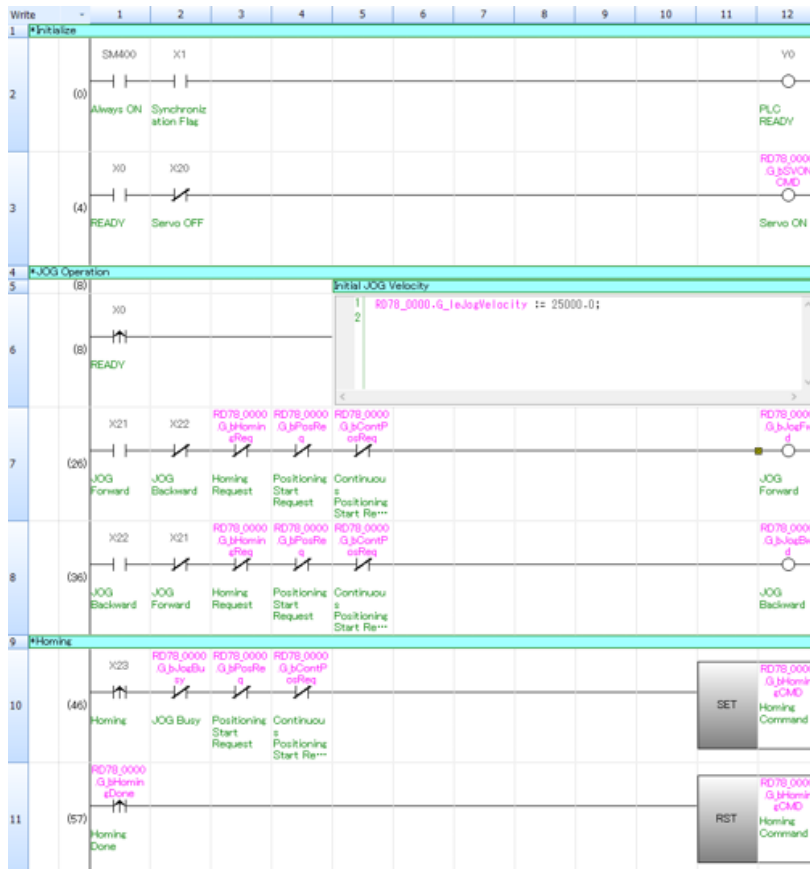
A seção a seguir mostra o padrão de operação do controle de posicionamento X24:



(Nota) Este exemplo de programa executa o servo ativado (ON) automaticamente, quando a CPU do PLC é definida como RUN. Quando a alimentação é ligada com os sinais de início ativados (ON), o servomotor pode ser ativado.

(2) Programa da CPU do PLC

1) MAIN (programa de scan, ladder)



Y0 é acionado em primeiro lugar.

Quando X0 é acionado, o comando servo ativado (ON) é executado.
 Acione X20 para executar o comando servo desativado (OFF).

Defina o valor inicial da velocidade de JOG.
 Este programa é usado com ST em linha.
 Uma vez que o atributo do motion controller da label global "G_JogVelocity" que armazena a velocidade de JOG é definido como "WRITE (-Motion)", o valor numérico deve ser definido na CPU do PLC.

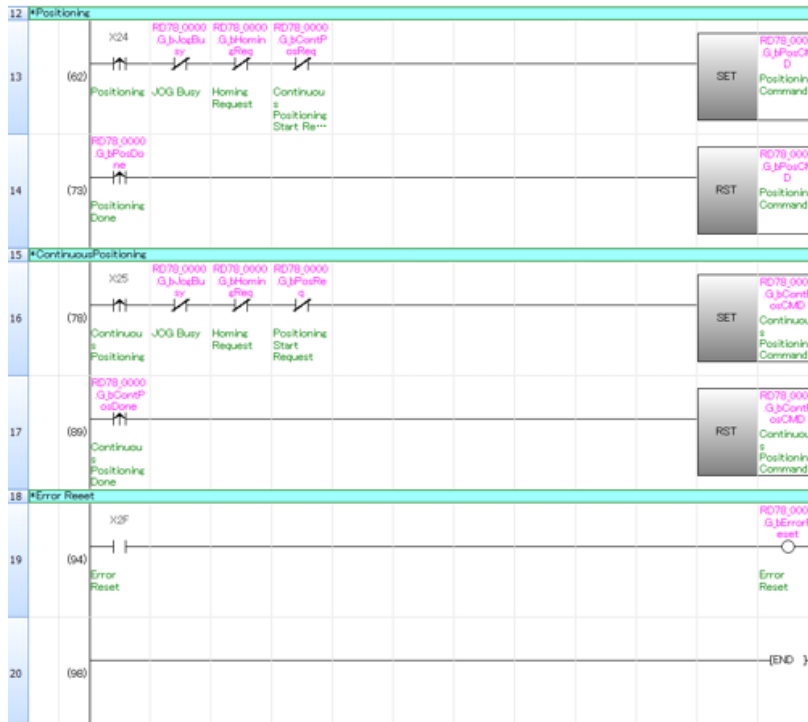
Aciona o sinal de início da operação JOG.
 Isso evita que a rotação de avanço e de recuo sejam iniciadas ao mesmo tempo.
 Um intertravamento é definido para evitar que a operação JOG seja iniciada durante a execução de outro programa.

A inicialização do retorno à posição inicial (X23) é retida na label pública G_bHomingCMD, e enviada ao módulo de movimento como condição de início do retorno à posição inicial.
 Um intertravamento é definido para evitar que o retorno à posição inicial seja iniciado durante a execução de outro programa.
 Após a recepção do sinal de que o módulo de movimento acionou o retorno à posição inicial, G_bHomingCMD é redefinido na borda de subida desse sinal.

(2) Programa da CPU do PLC

1) Parte de continuação de MAIN (programa de scan, ladder)

(Continuação da página anterior)



A borda de subida do início do controle de posicionamento (X24) é retida em `G_bPosCMD`, e enviada ao módulo de movimento como condição de início do controle de posicionamento. Um intertravamento é definido para evitar que o controle de posicionamento seja iniciado durante a execução de outro programa. Após a recepção do sinal de que o módulo de movimento acionou o retorno à posição inicial, `G_bPosCMD` é redefinido na borda de subida desse sinal.

O programa de início do posicionamento contínuo é descrito no capítulo 3. Para saber detalhes, consulte 3.4.

Quando X2F é ativado, os erros são redefinidos.

(2) Programa da CPU do PLC

2) MONITOR (ST, programa de scan)

SetPosition (Definir posição) e SetVelocity (Definir velocidade) do monitor do eixo, definidos como uma label pública, são armazenados nos dispositivos de palavras D0 e D2.

Uma vez que SetPosition e SetVelocity são do tipo número real de precisão dupla, são convertidos para o tipo palavra dupla com assinatura, para que possam ser facilmente manuseados pela CPU do PLC. (Nota)

Embora esses dispositivos de palavras não sejam usados neste curso, são usados para exibir dados em outros programas de sequência e no GOT, e para outros fins.

```
1 D0:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);  
2 D2:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);  
3
```

Especifique o tipo palavra dupla com assinatura com "D0:D".

(Nota) Quando o tipo número real de precisão dupla é convertido para o tipo palavra dupla com assinatura, se o valor a ser convertido estiver fora da faixa de -2147483648 a 2147483647, ocorrerá um erro de cálculo.

(3) Programa do módulo de movimento

1) ServoON_JOG (tipo de execução normal)

```

1 //-----Servo ON-----
2 MC_Power_1(
3     Axis      := Axis0001.AxisRef,
4     Enable    := TRUE,
5     ServoON   := G_bSVONCMD,
6     Busy      => bPowerBusy
7 );
8
9 //-----JOG Operation-----
10 //Initial Value Setting
11 IF (bPowerBusy) THEN
12     leJogAcceleration := 50000.0;
13     leJogDeceleration := 50000.0;
14     leJogJerk          := 0.0;
15 END_IF;
16
17 //JOG
18 MCv_Jog_1(
19     Axis      := Axis0001.AxisRef,
20     JogForward := G_bJogFwd,
21     JogBackward := G_bJogBwd,
22     Velocity   := G_leJogVelocity,
23     Acceleration := leJogAcceleration,
24     Deceleration := leJogDeceleration,
25     Jerk       := leJogJerk,
26     Busy       => G_bJogBusy
27 );
28

```

Recebe o sinal de servo ativado (G_bSVONCMD) da CPU do PLC e executa o servo ativado (ON).

Armazena os valores da aceleração JOG, desaceleração JOG, e jerk do JOG nas labels quando a saída Busy de MC_Power_1 é ativada.

Recebe os sinais de início de JOG e velocidade de JOG da CPU do PLC.

Retorna a saída Busy para a CPU do PLC.

(Nota) Neste exemplo de programa, os sinais de E/S dos FBs que não forem usados, ou que não tenham sido alterados em relação aos valores iniciais, serão omitidos.

- (3) Programa do módulo de movimento
 2) Posição inicial (tipo de execução normal)

```

1 //-----Homing Operation-----
2 //Initial Value Setting, Operation Start Request
3 IF G_bHomingCMD THEN
4   G_lePosition0 := 0.0 ;
5   G_bHomingReq := TRUE ;
6 ELSE
7   G_bHomingReq := FALSE ;
8 END_IF;
9
10 //Homing
11 MC_Home_1(
12   Axis      := Axis0001.AxisRef ,
13   Execute   := G_bHomingReq,
14   Position  := G_lePosition0 ,
15   Done      => bHomingDone ,
16   Busy      => G_bHomingBusy ,
17   CommandAborted => bHomingAborted ,
18   Error     => bHomingError
19 );
20
21 //Done Signal => PLC CPU
22 G_bHomingDone := bHomingDone OR bHomingAborted OR bHomingError;
23

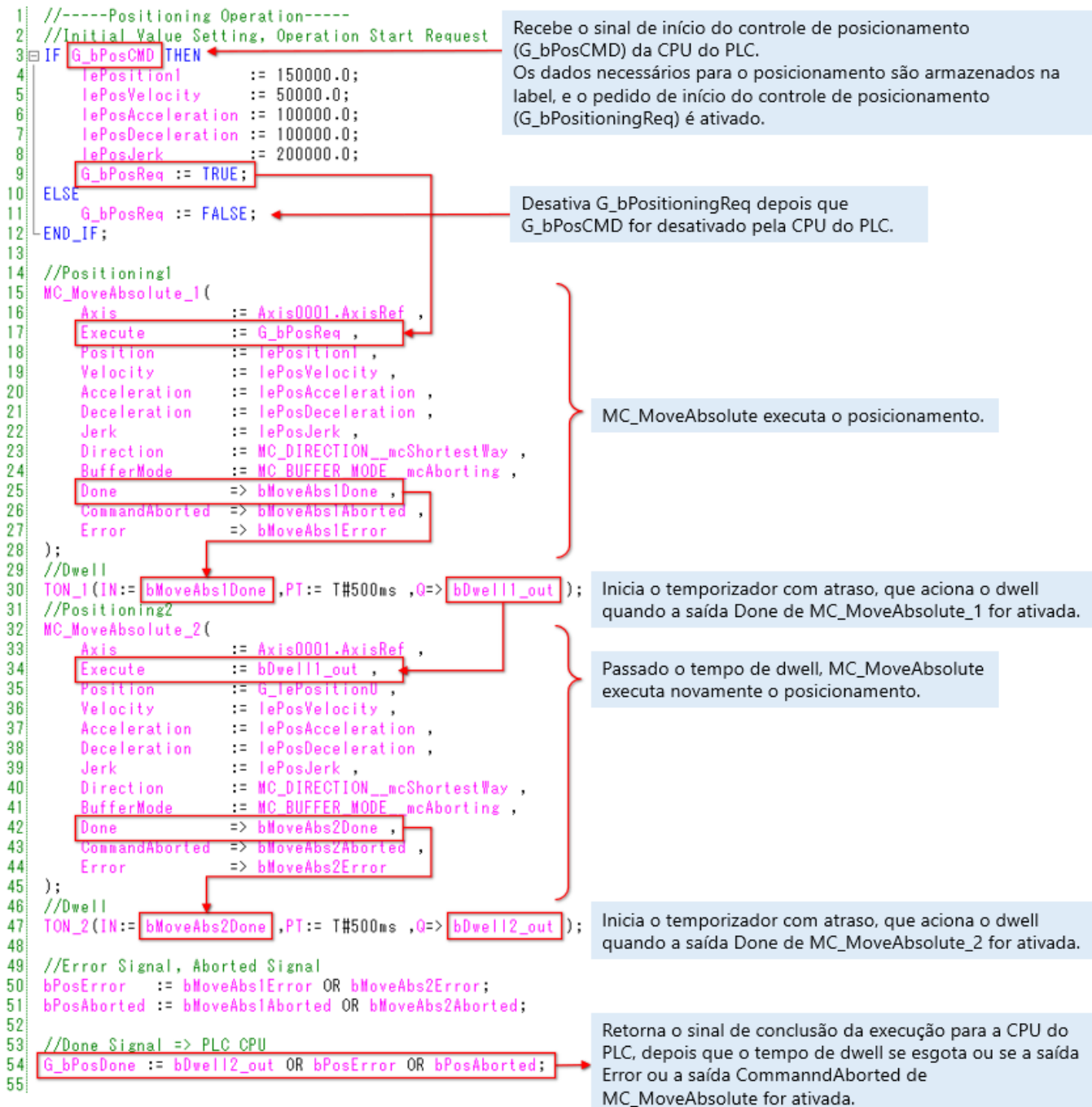
```

Recebe o sinal de comando do retorno à posição inicial (G_bHomingCMD) da CPU do PLC.
 Armazena o endereço da posição inicial na label e ativa o pedido de retorno à posição inicial (G_bHomingReq).

Desativa G_bHomingReq quando G_bHomingCMD é desativado.

Retorna o sinal de conclusão da execução para a CPU do PLC, depois que o retorno à posição inicial for concluído com sucesso (saída Done ON), a execução é interrompida (saída CommandAborted ON), ou ocorreu um erro (saída Error ON).

- (3) Programa do módulo de movimento
 3) Posicionamento (tipo de execução normal)



- (3) Programa do módulo de movimento
4) ErrorReset (tipo de execução normal)

```
1 //Axis Error Reset
2 MC_Reset_1(
3     Axis      := Axis0001.AxisRef ,
4     Execute   := G_bErrorReset
5 );
6
7 //System Error Reset
8 MCv_MotionErrorReset_1(
9     Execute   := G_bErrorReset
10 );
11
```

Recebe o sinal de redefinição do erro (G_bErrorReset) da CPU do PLC, e executa a redefinição do erro do eixo e a redefinição do erro do sistema.

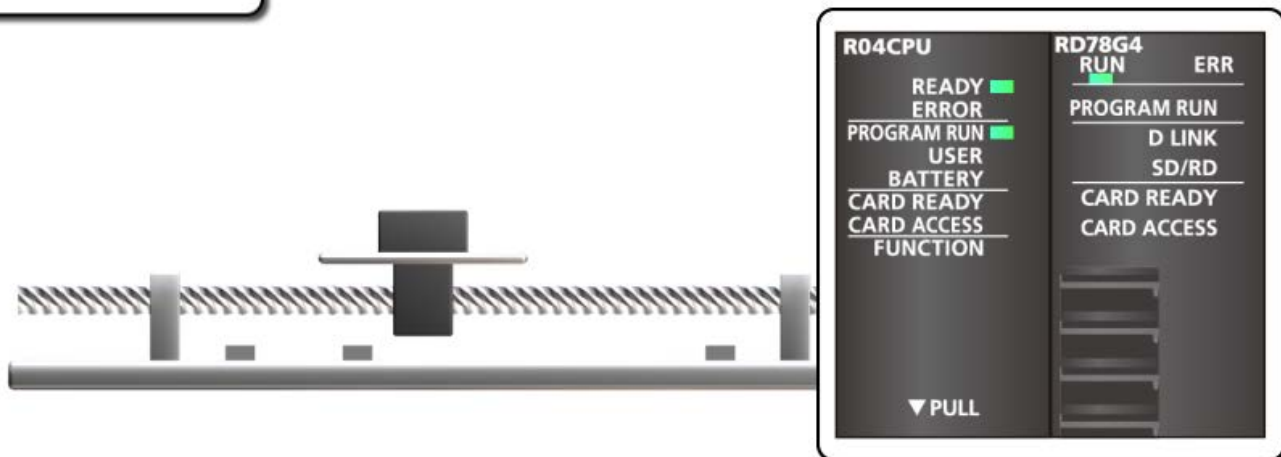
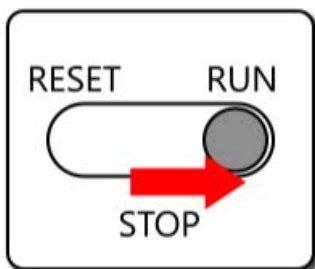
Grave o programa e os parâmetros na CPU do PLC e no módulo de movimento.

- 1) Depois que todos os programas da CPU do PLC forem reconstruídos, selecione [Online] → [Write to PLC], na barra de ferramentas do GX Works3, para gravar todos os dados na CPU do PLC.
- 2) Quando os parâmetros forem gravados na CPU do PLC, a comunicação com o módulo de movimento é ativada. Selecione [Online] → [Write to Module], na barra de ferramentas de Motion Control Setting Function, para gravar todos os dados no módulo de movimento.
- 3) Reinicie a CPU do PLC para concluir a gravação.

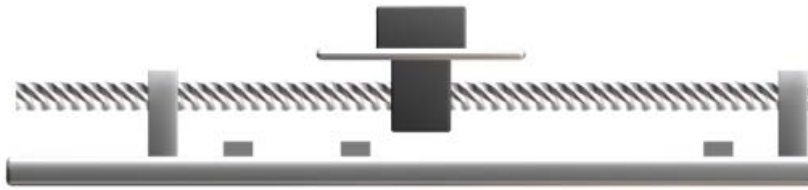
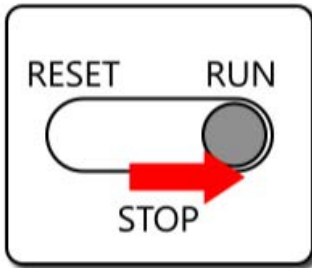
Clique no botão Executar, na área inferior esquerda da janela.



Verifique a operação do exemplo de programa.
Antes de iniciar a operação, certifique-se de que os programas da CPU
do PLC e do módulo de movimento estejam gravados.



Defina o botão RUN/STOP/RESET da CPU do PLC como RUN.
A luz READY e a luz PROGRAM RUN da CPU do PLC se acendem.
A luz RUN do módulo de movimento se acende.



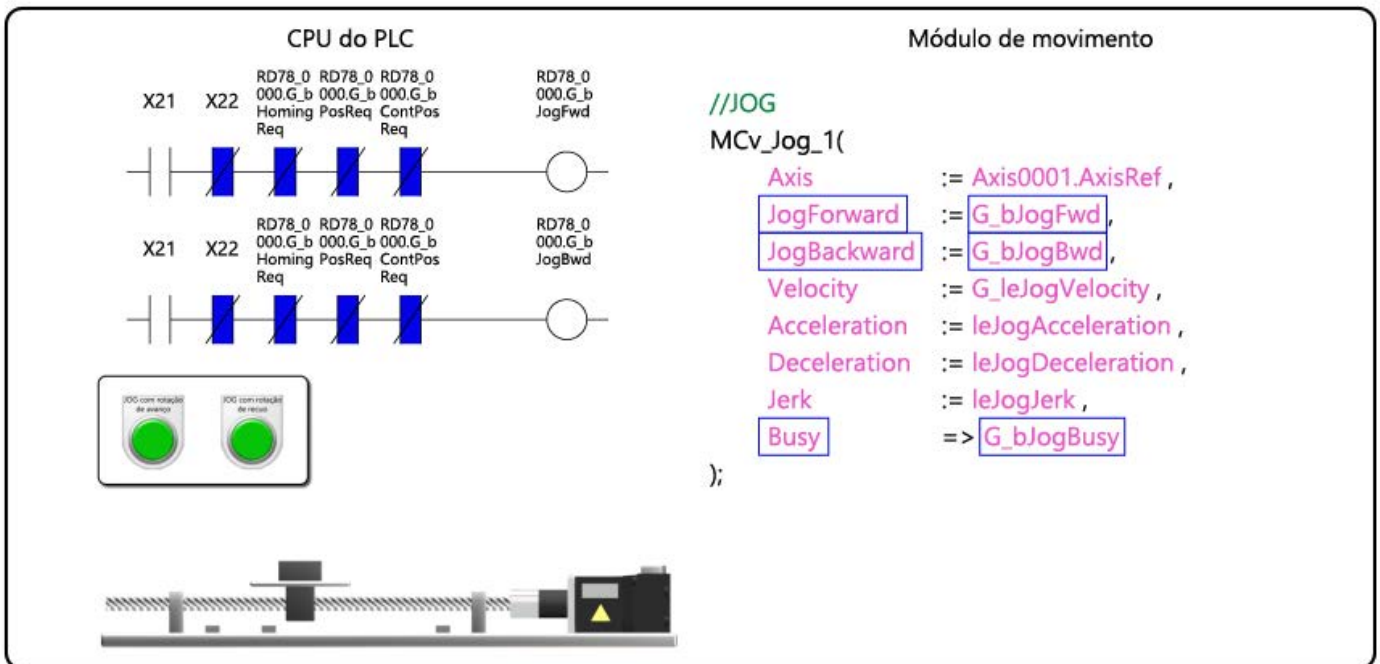
Espera até que a luz PROGRAM RUN do módulo de movimento se acenda.
 "r.01" é exibido no servo amplificador. (Os pontos se acendem.)
 O servomotor entra no estado de servo ativado (ON).



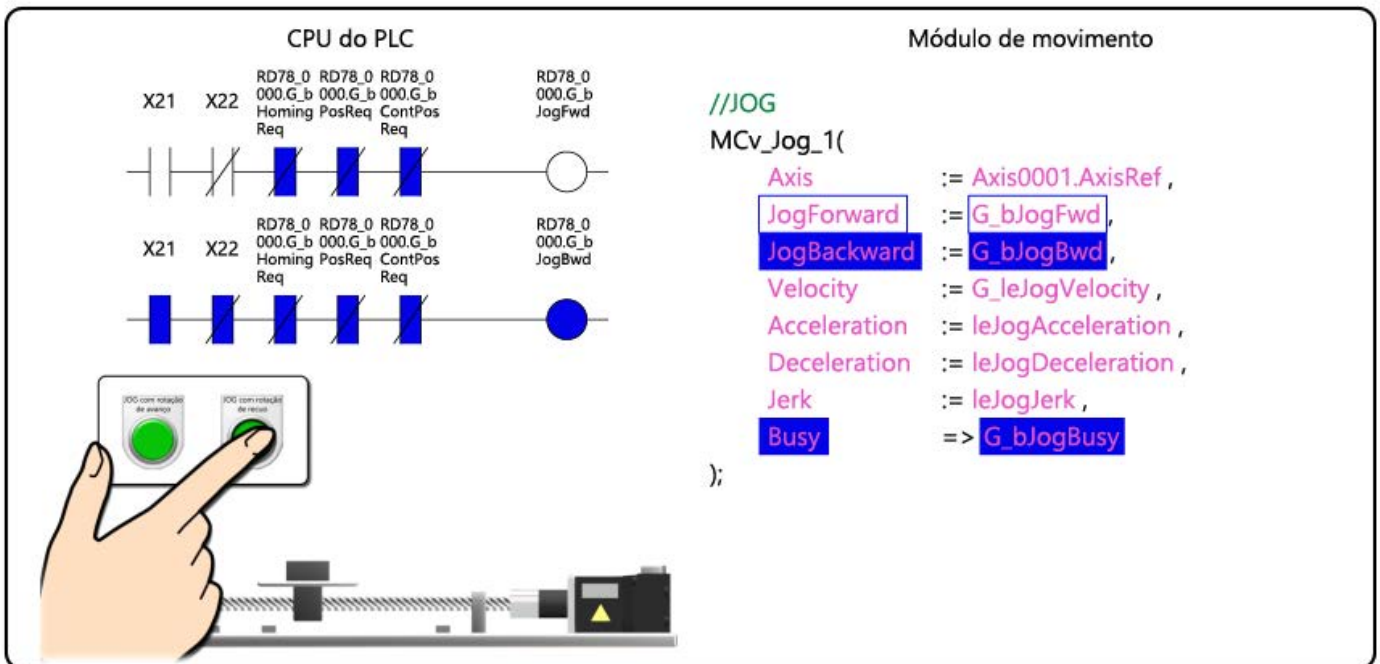
Acione X20 para executar o comando servo desativado (OFF).
"r.01" é exibido no servo amplificador. (Os pontos piscam.)
Desative X20 para executar o servo ativado (ON) novamente.



Ative o JOG com rotação de avanço (X21) para mover o eixo na direção de aumento do endereço, e desative-o para parar.
Ative o JOG com rotação de recuo (X22) para mover o eixo na direção de redução do endereço, e desative-o para parar.



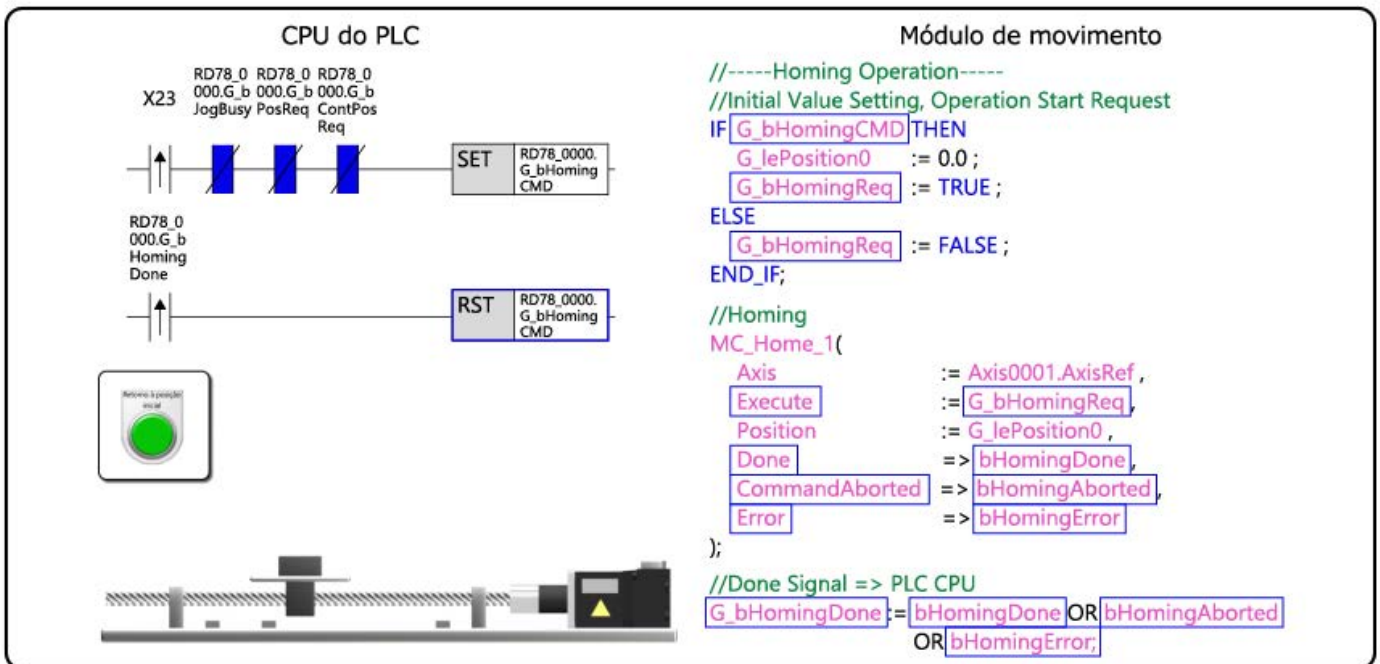
Verifique o monitor do programa.
 Quando X21 é ativado, "RD78_0000.G_bJogFwd" é ativado e "G_bJogFwd" é ativado, no lado do módulo de movimento.
 Quando a entrada JogForward de MCv_Jog_1 é ativada, o JOG com rotação de avanço é iniciado.



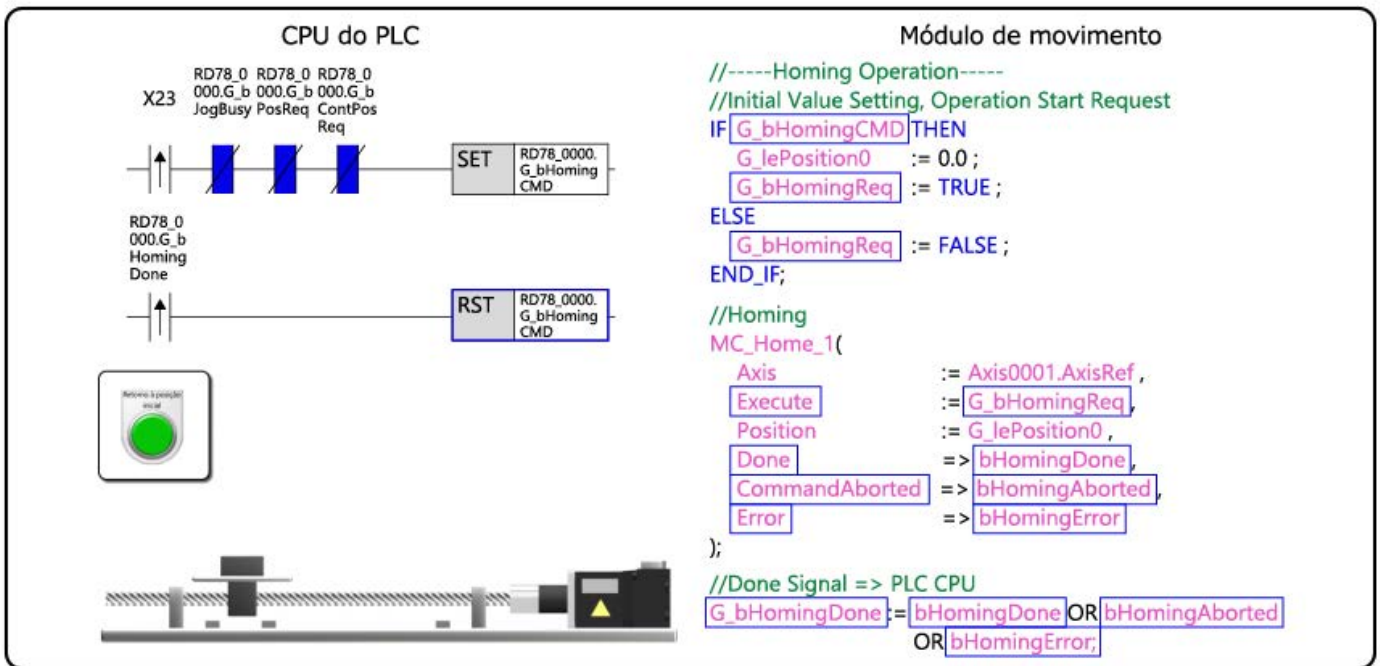
Quando X22 é ativado, "RD78_0000.G_bJogBwd" é ativado e "G_bJogFwd" é ativado, no lado do módulo de movimento. Quando a entrada JogBackward de MCv_Jog_1 é ativada, o JOG com rotação de recuo é iniciado.



Ative o retorno à posição inicial (X23) para iniciar esse procedimento.
Execute o retorno à posição inicial com o método do dog de proximidade (33 é subtraído de Pr.PT45)
O eixo para um pouco depois do dog,
e define esse ponto como a posição inicial.



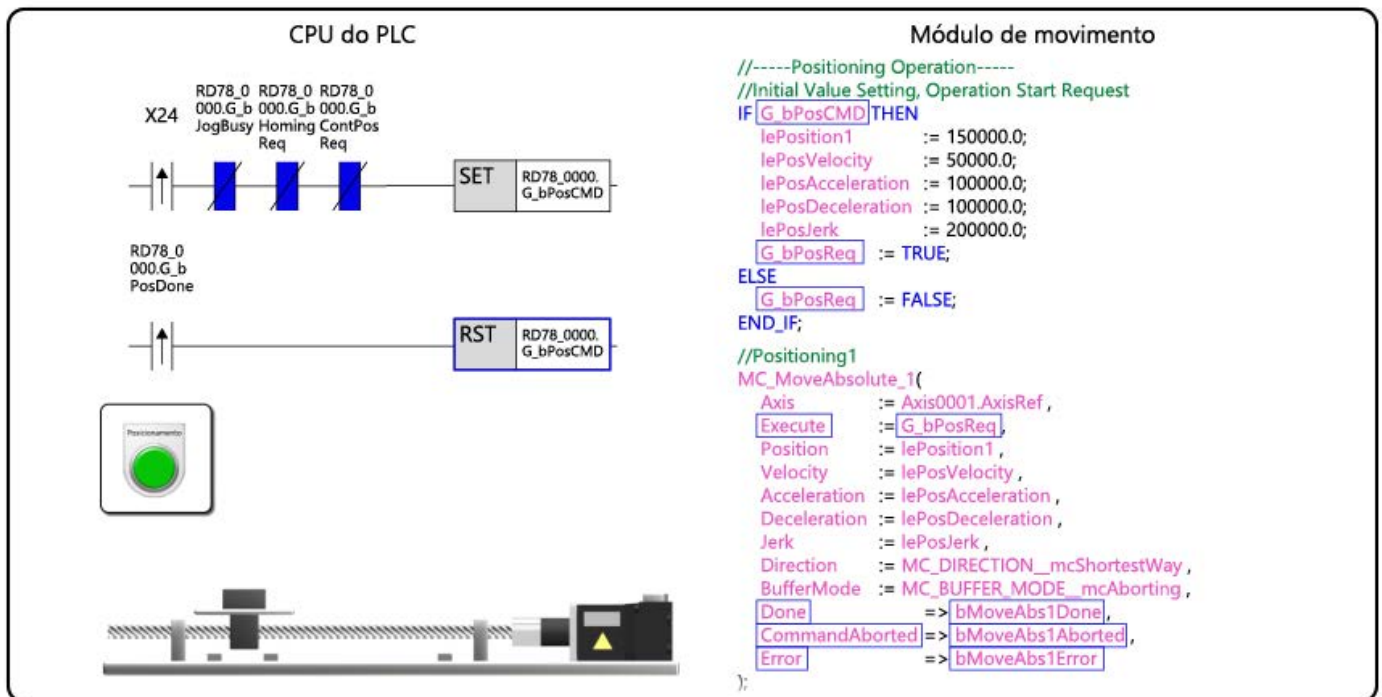
Verifique o monitor do programa.
Quando X23 é ativado, "RD78_0000.G_bHomingCMD" é definido.
"G_bHomingCMD" é ativado no lado do módulo de movimento, e
"G_bHomingReq", que o comando de execução de MC_Home_1, é
ativado.



Quando o retorno à posição inicial é concluído, a saída Done e "G_bHomingDone" é ativada.
"G_bHomingCMD", no lado da CPU do PLC,
é redefinido com o estado inicial.

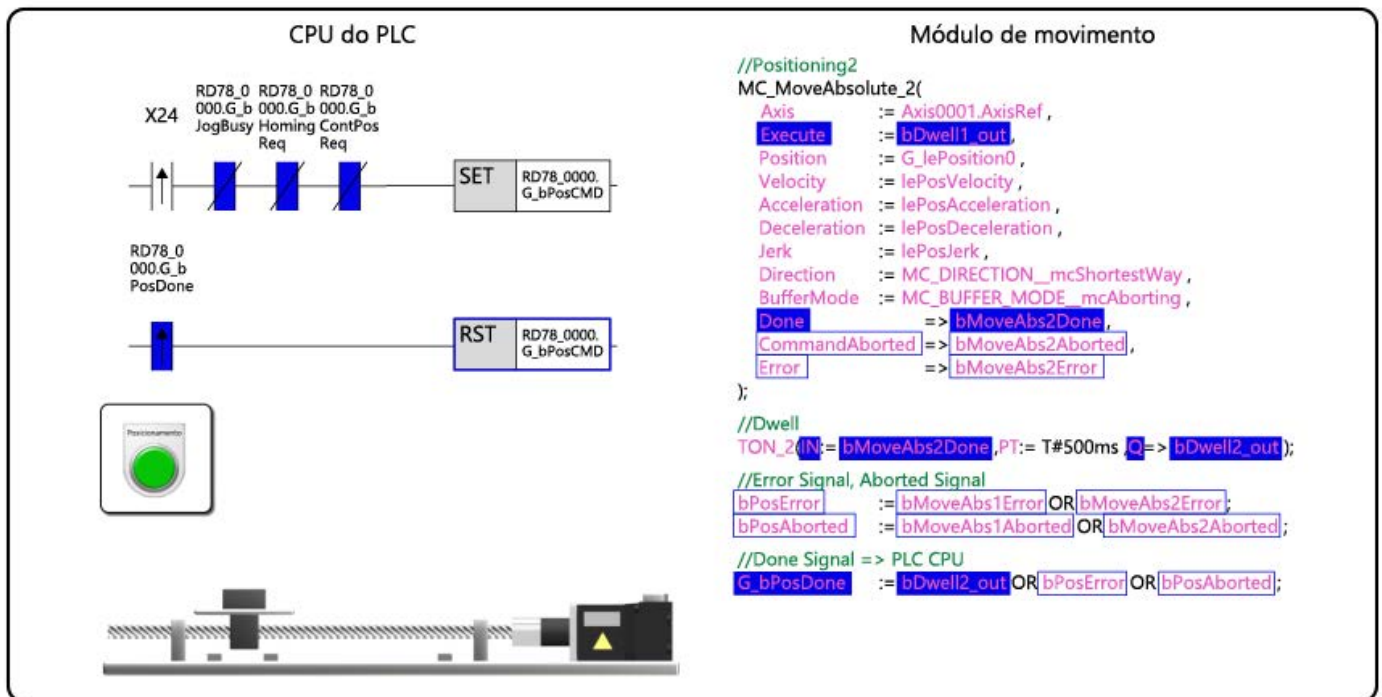


A ativação do início de posicionamento (X24)
inicia o movimento de ida e volta.
O eixo avança 150 mm e para durante 0,5 segundo,
recua 150 mm e para durante 0,5 segundo.

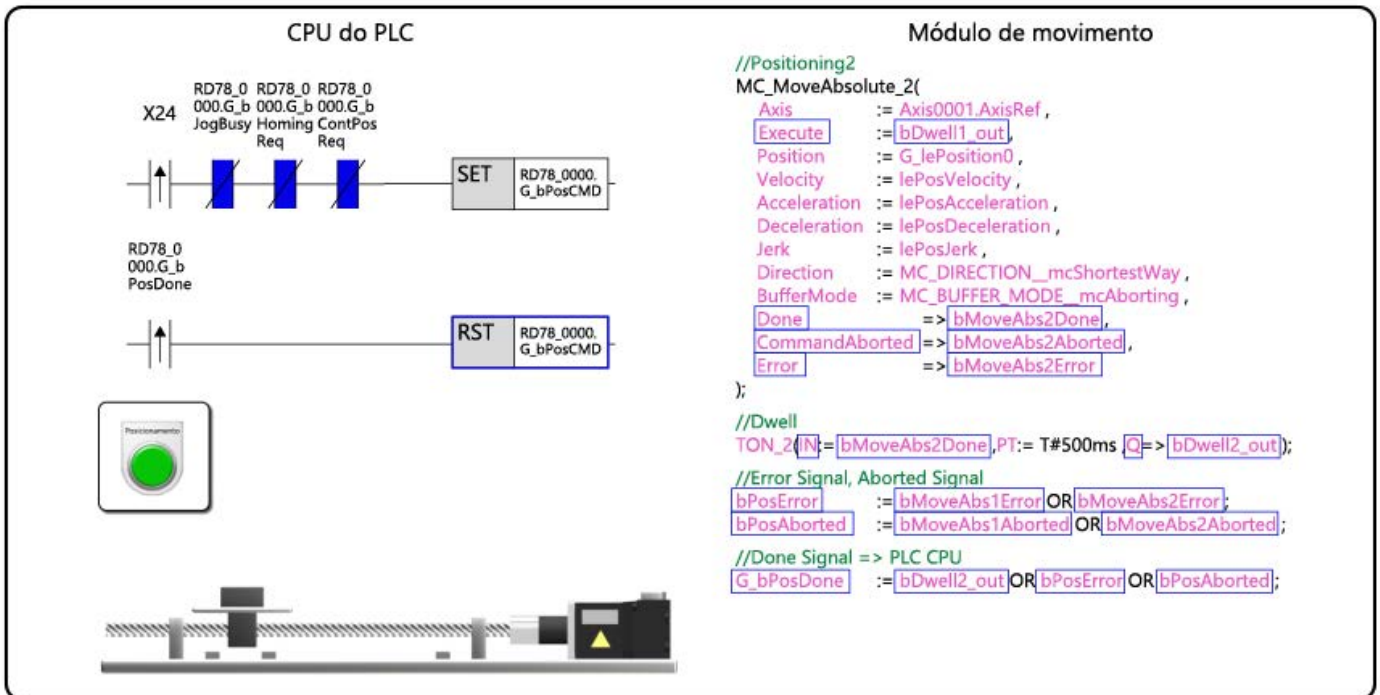


Verifique o monitor do programa.

Quando X24 é ativado, "RD78_0000.G_bPosCMD" é definido. "G_bPosCMD" é ativado no lado do módulo de movimento, e "G_bPosReq", que é o comando de execução de MC_MoveAbsolute_1, é ativado.



Depois que o movimento de ida e volta termina e o tempo de dwell se esgota, "G_bPosDone" é ativado.
 "G_bPosCMD", no lado da CPU do PLC, é redefinido com o estado inicial.



Isto conclui a verificação da operação.
Vá para a próxima página.

Neste capítulo, você aprendeu o seguinte:

- O que é uma label pública?
- Definição da label pública
- Exemplo de programa
- Gravando um programa
- Verificação da operação

Ponto

O que é uma label pública?	<ul style="list-style-type: none"> • A label pública é uma label compartilhada que pode ser usada no módulo de movimento e também na CPU do PLC.
Definição da label pública	<ul style="list-style-type: none"> • Registre as labels públicas a partir das labels globais do módulo de movimento. • Selecione se cada label deverá ser lida ou gravada da/para a CPU do PLC. • Para definir os membros de um tipo de dados estruturados preparado no sistema como label pública, registre as labels públicas pela camada do tipo de dados estruturados. • Depois de definir as labels públicas no módulo de movimento, reconstrua todos os programas e reflita as labels públicas. • As labels públicas são registradas na label do módulo, no lado da CPU do PLC.
Exemplo de programa	<ul style="list-style-type: none"> • Este capítulo descreveu o seguinte exemplo de programa: um programa de ladder da CPU do PLC que utiliza labels públicas para trocar o sinal de início do posicionamento e o sinal de conclusão do posicionamento.
Gravando um programa	<ul style="list-style-type: none"> • Grave os dados inicialmente na CPU do PLC, e em seguida no módulo de movimento.
Verificação da operação	<ul style="list-style-type: none"> • Você verificou a operação do exemplo de programa apresentado no vídeo.

O modo de buffer executa operações continuamente, iniciando vários FBs de operação de Motion control FBs. Ele pode ser definido com a entrada BufferMode do Motion control FB. É possível iniciar até dois FBs ao mesmo tempo, para cada eixo e grupo de eixos.

(Exemplo) MC_MoveAbsolute


```
MC_MoveAbsolute_1(
  Axis      := Axis0001.AxisRef ,
  Execute   := G_bPositioningReq ,
  ContinuousUpdate := FALSE ,
  Position  := lePosition1 ,
  Velocity  := lePosVelocity ,
  Acceleration := lePosAcceleration ,
  Deceleration := lePosDeceleration ,
  Jerk      := lePosJerk ,
  Direction := MC_DIRECTION_mcShortestWay ,
  BufferMode := MC_BUFFER_MODE__mcAborting ,
  Options   := 0 ,//mcAccDec
  Done      => bMoveAbs1Done ,
  CommandAborted => bMoveAbs1Aborted ,
  Error     => bMoveAbs1Error
);
```

- | | |
|---|---|
| 0 ou MC_BUFFER_MODE__mcAborting | ••• O FB em execução é interrompido, e o próximo FB é imediatamente executado. |
| 1 ou MC_BUFFER_MODE__mcBuffered | ••• Depois que a operação do FB em execução é concluída, o próximo FB é executado. |
| 2 ou MC_BUFFER_MODE__mcBlendingLow | ••• A velocidade alvo mais baixa do FB em execução e do FB que será colocado em armazenamento temporário é definida como a velocidade de comutação. |
| 3 ou MC_BUFFER_MODE__mcBlendingPrevious | ••• A velocidade alvo do FB em execução é definida como a velocidade de comutação. |
| 4 ou MC_BUFFER_MODE__mcBlendingNext | ••• A velocidade alvo do FB que será colocado em armazenamento temporário é definida como a velocidade de comutação. |
| 5 ou MC_BUFFER_MODE__mcBlendingHigh | ••• A velocidade alvo mais alta do FB em execução e do FB que será colocado em armazenamento temporário é definida como a velocidade de comutação. |

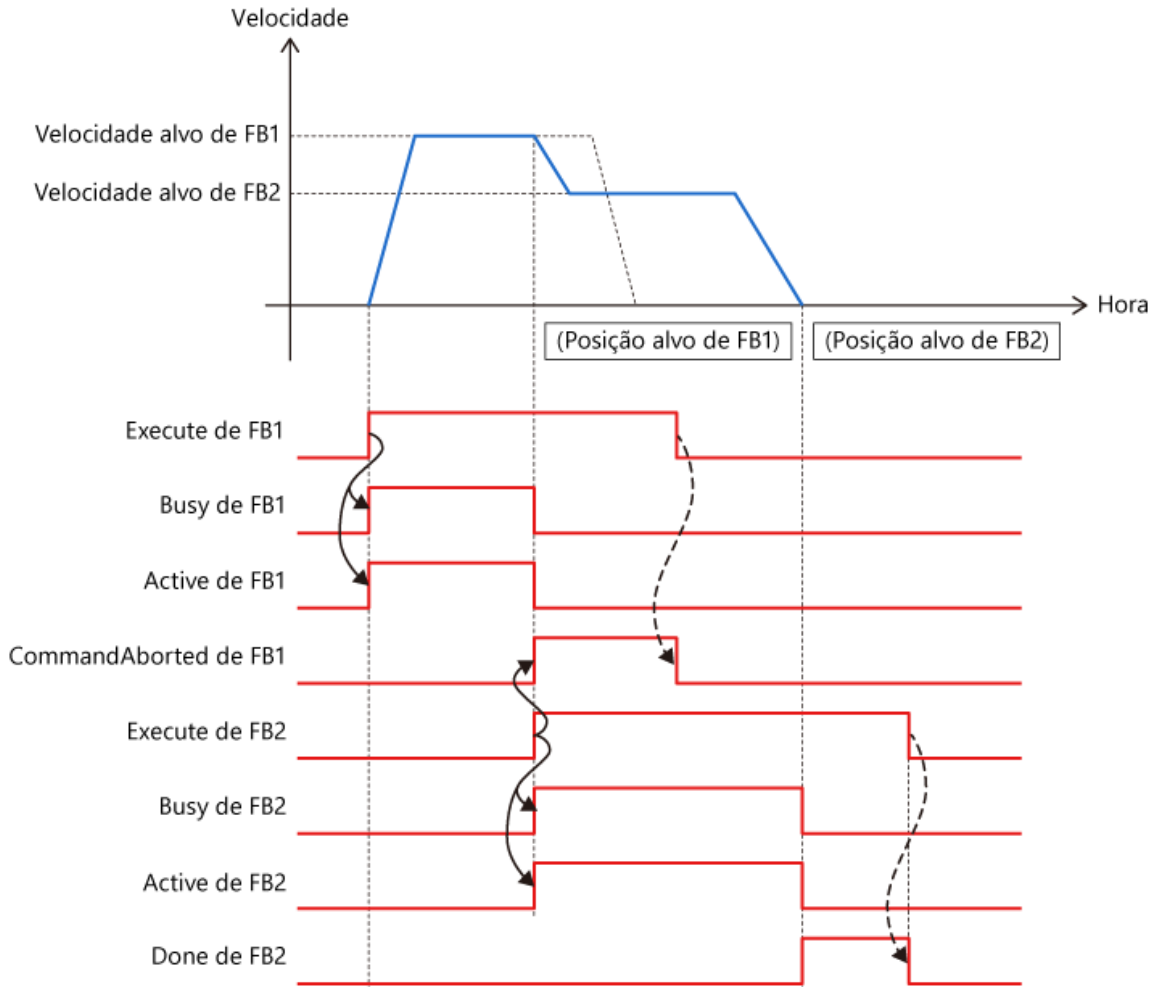
[Ponto]

Para a entrada de Direction e BufferMode, especifique números ou enumeradores ENUM começando com MC_BUFFER_MODE e MC_DIRECTION.

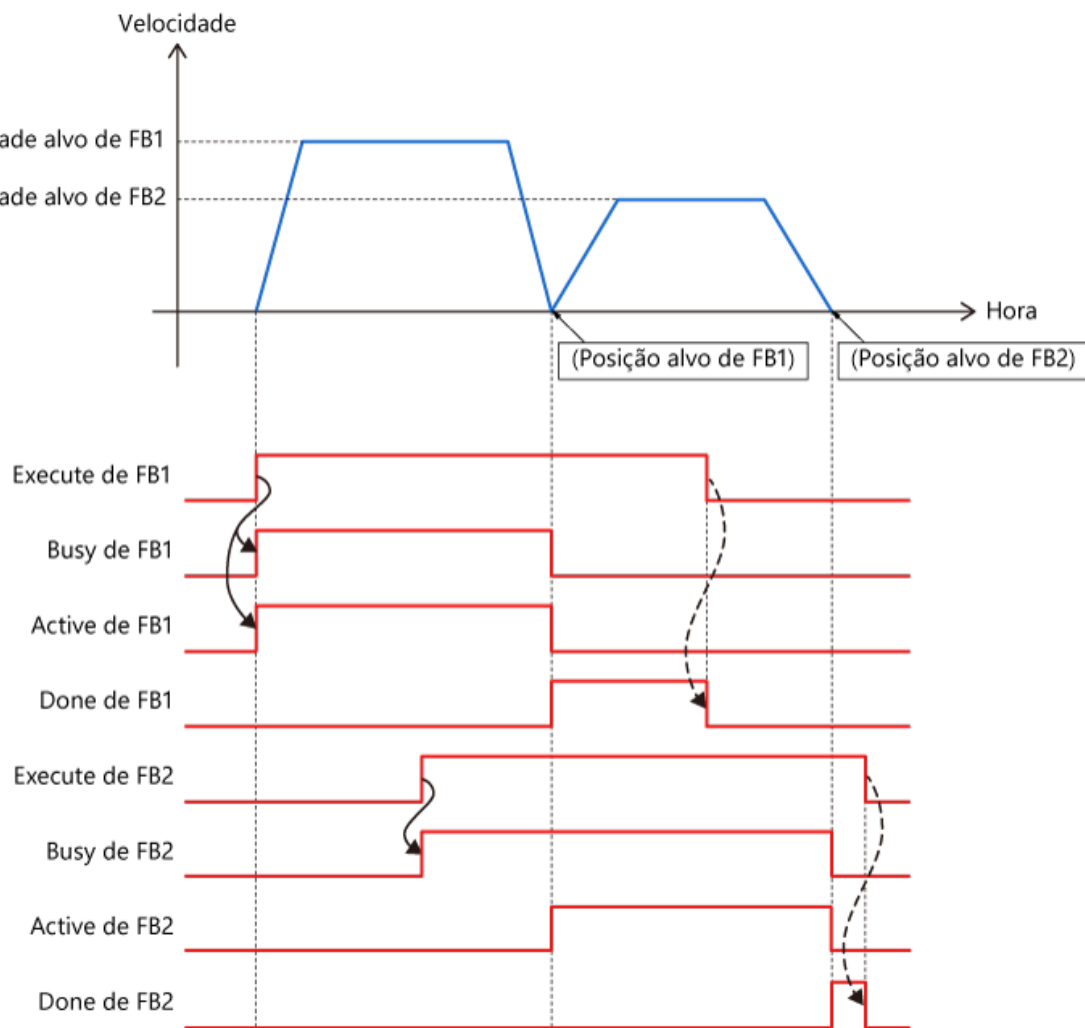
Para saber detalhes sobre os enumeradores ENUM, consulte o manual a seguir.

 MELSEC iQ-R Programming Manual (Motion Control Function Blocks)
 2 VARIABLES AND MOTION CONTROL FB
 2.2 List of Enumerators

O diagrama a seguir mostra a operação quando BufferMode é definido como 0: mcAborting. O FB em execução é interrompido, e o próximo FB é imediatamente executado.



O diagrama a seguir mostra a operação quando BufferMode é definido como 1: mcBuffered. Quando a operação do FB em execução é concluída, o próximo FB é executado.



Quando BufferMode é definido como mcBlending***, o próximo FB é executado continuamente, assim que se atinge a posição alvo do FB em execução.

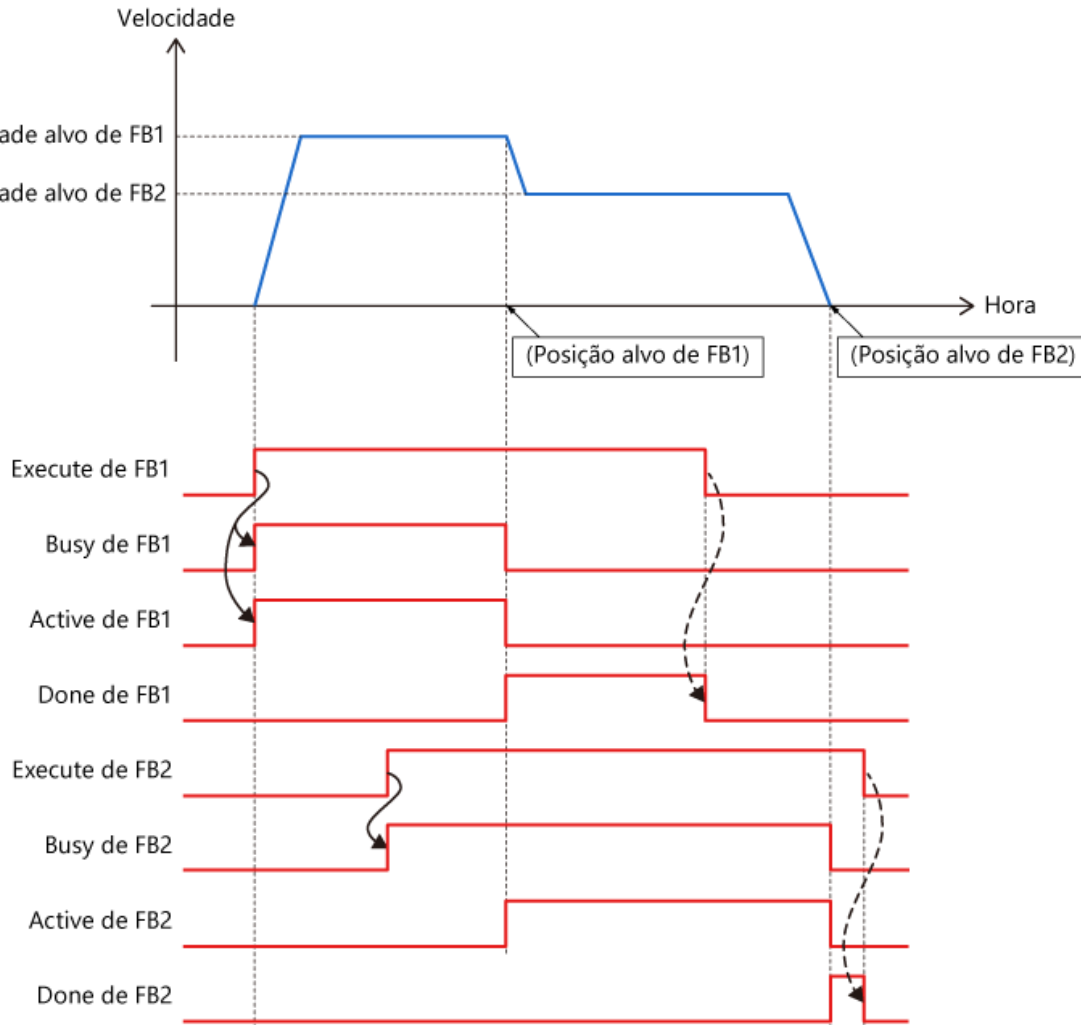
Na descrição a seguir, o primeiro FB a ser executado é FB1, e o FB a ser armazenado temporariamente é FB2.

(1) BlendingPrevious

O diagrama a seguir mostra a operação quando BufferMode é definido como 3: mcBlendingPrevious.

A operação é executada na velocidade alvo de FB1, até a posição alvo de FB1.

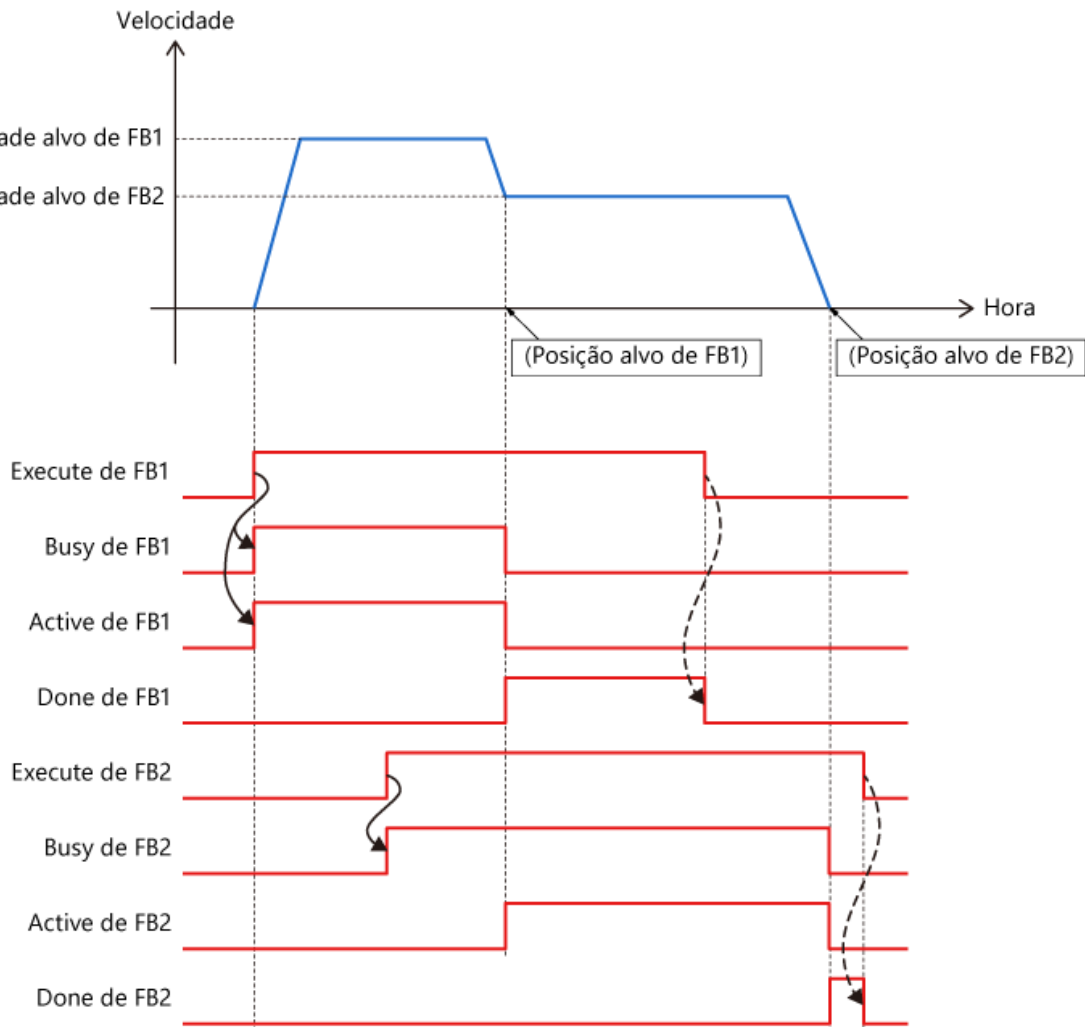
Quando a operação é alterada para FB2, a velocidade é alterada para a velocidade alvo de FB2, e ocorre a mudança para a posição alvo de FB2.



(2) BlendingNext

O diagrama a seguir mostra a operação quando BufferMode é definido como 4: mcBlendingNext.

A velocidade muda para a velocidade alvo de FB2 quando a operação atinge a posição alvo de FB1.



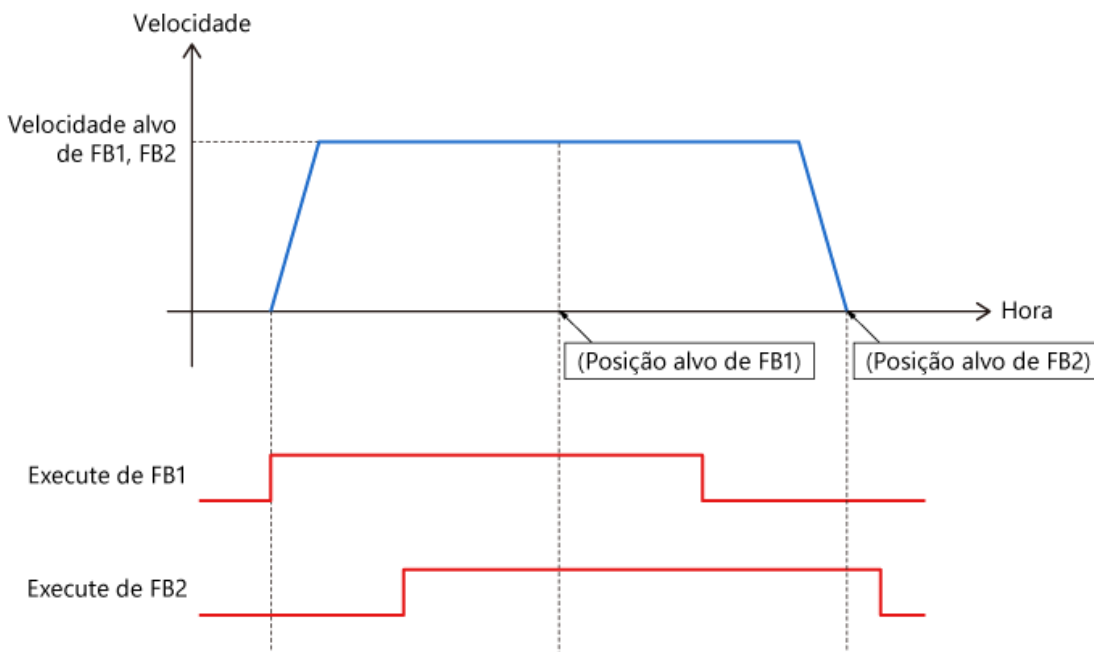
(3) BlendingLow, BlendingHigh

A operação quando BufferMode é definido como 2: mcBlendingLow ou 5: mcBlendingHigh varia, dependendo de qual das velocidade alvo de FB1 e FB2 for maior.

Definição do valor	Velocidade alvo de FB1 > Velocidade alvo de FB2	Velocidade alvo de FB1 < Velocidade alvo de FB2
2: mcBlendingLow	A mesma operação que BlendingPrevious	A mesma operação que BlendingNext
5: mcBlendingHigh	A mesma operação que BlendingNext	A mesma operação que BlendingPrevious

[Ponto]

O diagrama a seguir mostra a forma de onda da velocidade para BlendingPrevious, BlendingNext, BlendingHigh e BlendingLow, quando a velocidade alvo de FB1 e FB2 é a mesma.



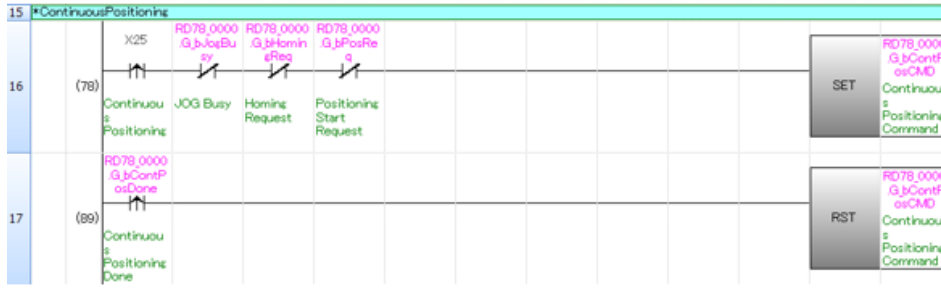
(1) Operação do exemplo de programa

Este capítulo utiliza o exemplo de programa visto no Capítulo 2.

Verifique a diferença na operação em modo de buffer de um programa que começa com X25.

Item	FB1 (MC_MoveAbsolute)	FB2 (MC_MoveAbsolute)
Endereço de posicionamento	75000,0 [μm]	150000,0 [μm]
Velocidade	50000,0 [$\mu\text{m}/\text{s}$]	25000,0 [$\mu\text{m}/\text{s}$]
Aceleração, desaceleração	100000,0 [$\mu\text{m}/\text{s}^2$]	50000,0 [$\mu\text{m}/\text{s}^2$]
Jerk	200000,0 [$\mu\text{m}/\text{s}^3$]	100000,0 [$\mu\text{m}/\text{s}^3$]

- (2) Programa da CPU do PLC
MAIN (programa de scan, ladder)



A borda de subida do início do controle de posicionamento contínuo (X25) é retida com G_bContPosCMD, e enviada ao módulo de movimento como condição de início do controle de posicionamento contínuo. Um intertravamento é definido para evitar que o controle de posicionamento seja iniciado durante a execução de outro programa. Após a recepção do sinal de que o módulo de movimento acionou o sinal de conclusão do retorno à posição inicial, G_bContPosCMD é redefinido na borda de subida desse sinal.

- (3) Programa do módulo de movimento
ContinuousPositioning (tipo de execução normal)

```

1  //-----Continuous Positioning Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bContPosCMD THEN
4      lePosition1 := 75000.0;
5      lePosVelocity1 := 50000.0;
6      lePosAcceleration1 := 100000.0;
7      lePosDeceleration1 := 100000.0;
8      lePosJerk1 := 200000.0;
9      lePosition2 := 150000.0;
10     lePosVelocity2 := 25000.0;
11     lePosAcceleration2 := 50000.0;
12     lePosDeceleration2 := 50000.0;
13     lePosJerk2 := 100000.0;
14     G_bContPosReq := TRUE;
15 ELSE
16     G_bContPosReq := FALSE;
17 END_IF;
18
19 FB1 //Positioning1
20 MC_MoveAbsolute_1(
21     Axis := Axis0001.AxisRef,
22     Execute := G_bContPosReq,
23     Position := lePosition1,
24     Velocity := lePosVelocity1,
25     Acceleration := lePosAcceleration1,
26     Deceleration := lePosDeceleration1,
27     Jerk := lePosJerk1,
28     Direction := MC_DIRECTION_ncShortestWay,
29     Done => bMoveAbs1Done,
30     Active => bMoveAbs1Active,
31     CommandAborted => bMoveAbs1Aborted,
32     Error => bMoveAbs1Error
33 );
34 FB2 //Positioning2
35 MC_MoveAbsolute_2(
36     Axis := Axis0001.AxisRef,
37     Execute := bMoveAbs1Active,
38     Position := lePosition2,
39     Velocity := lePosVelocity2,
40     Acceleration := lePosAcceleration2,
41     Deceleration := lePosDeceleration2,
42     Jerk := lePosJerk2,
43     Direction := MC_DIRECTION_ncShortestWay,
44     BufferMode := MC_BUFFER_MODE_ncBuffered,
45     Done => bMoveAbs2Done,
46     CommandAborted => bMoveAbs2Aborted,
47     Error := bMoveAbs2Error
48 );
49 //Dwell
50 SET(bMoveAbs2Done,bDwell_In);
51 TON_1(IN:= bDwell_In,PT:= T#100ms, Q=> bDwell_out);
52
53 //Error Signal, Aborted Signal
54 bError := bMoveAbs1Error OR bMoveAbs2Error;
55 bAborted := bMoveAbs1Aborted OR bMoveAbs2Aborted;
56
57 //Done Signal => PLC CPU
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
59 //Reset Dwell_In
60 RST(G_bContPosDone,bDwell_In);
61

```

Recebe o sinal de início do controle de posicionamento contínuo (G_bContPosCMD) da CPU do PLC. Os dados necessários para o posicionamento são armazenados na label, e o pedido de início do controle de posicionamento contínuo (G_bContPositioningReq) é ativado.

Desativa G_bContPositioningReq, depois que G_bContPosCMD é desativado.

FB1 (MC_MoveAbsolute_1) executa o posicionamento.

FB2 (MC_MoveAbsolute_2) é iniciado pela saída Active de FB1 (MC_MoveAbsolute_1). Em seguida, FB2 é colocado no armazenamento temporário, enquanto FB1 é executado.

Altere a entrada BufferMode de FB2 (MC_MoveAbsolute_2) e verifique a operação do modo de buffer.

O dwell que aciona o temporizador com atraso é iniciado pela saída Done de FB2 (MC_MoveAbsolute_2).

Retorna o sinal de conclusão da execução para a CPU do PLC, depois que o tempo de dwell se esgota ou se a saída Error ou a saída CommandAborted de MC_MoveAbsolute for ativada. Ao mesmo tempo, a entrada do tempo com atraso é redefinida.

Clique no botão Executar, na área inferior esquerda da janela.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBuffered ,  
45     Done      => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error     => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Verifique a operação do modo de buffer.

Abra "LinearInterpolation" no exemplo de programa e altere a entrada BufferMode de MC_MoveAbsolute_2 para verificar a operação do modo de buffer.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBuffered ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

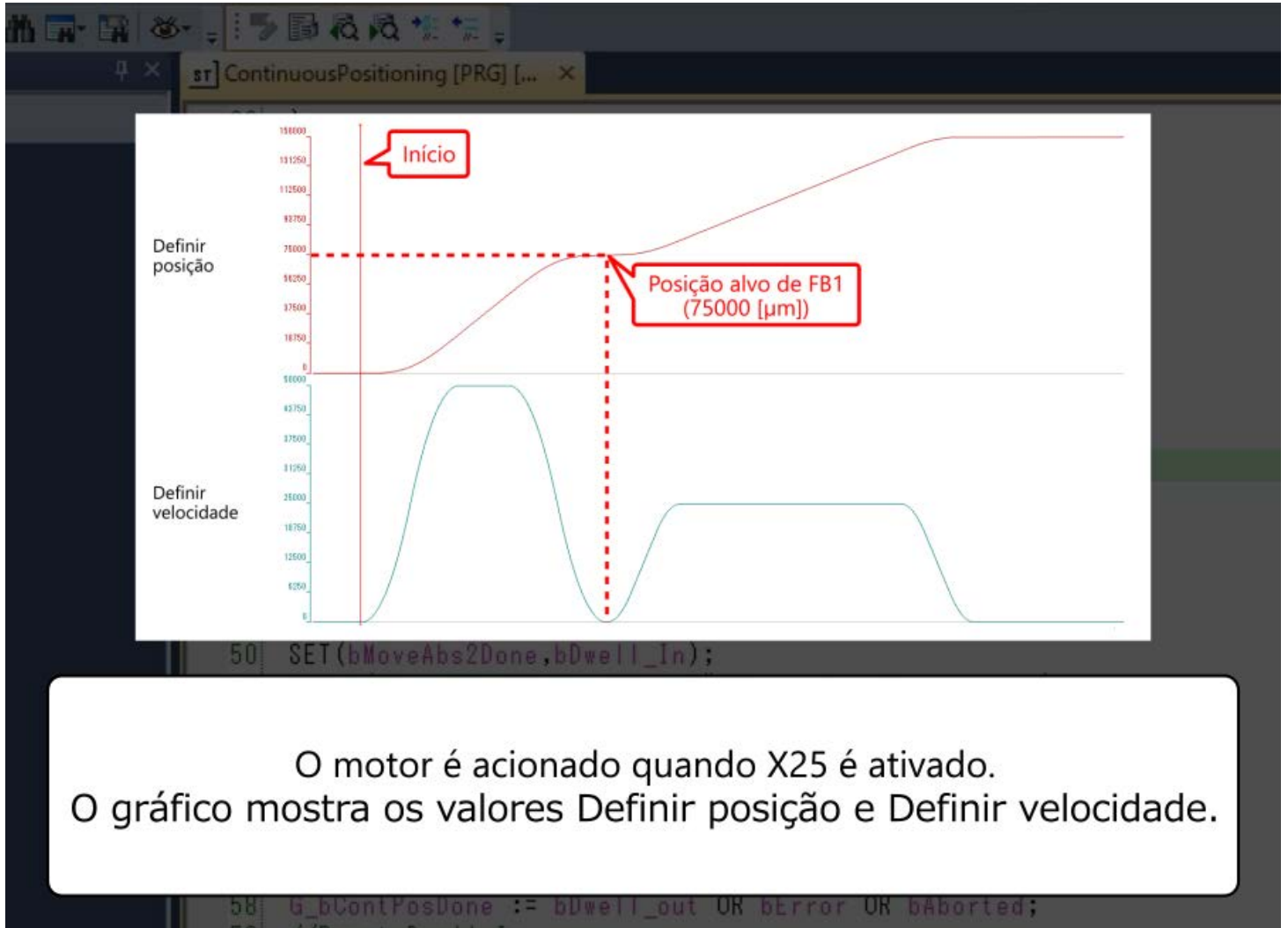
mcBuffered é predefinido quando o programa é baixado.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```

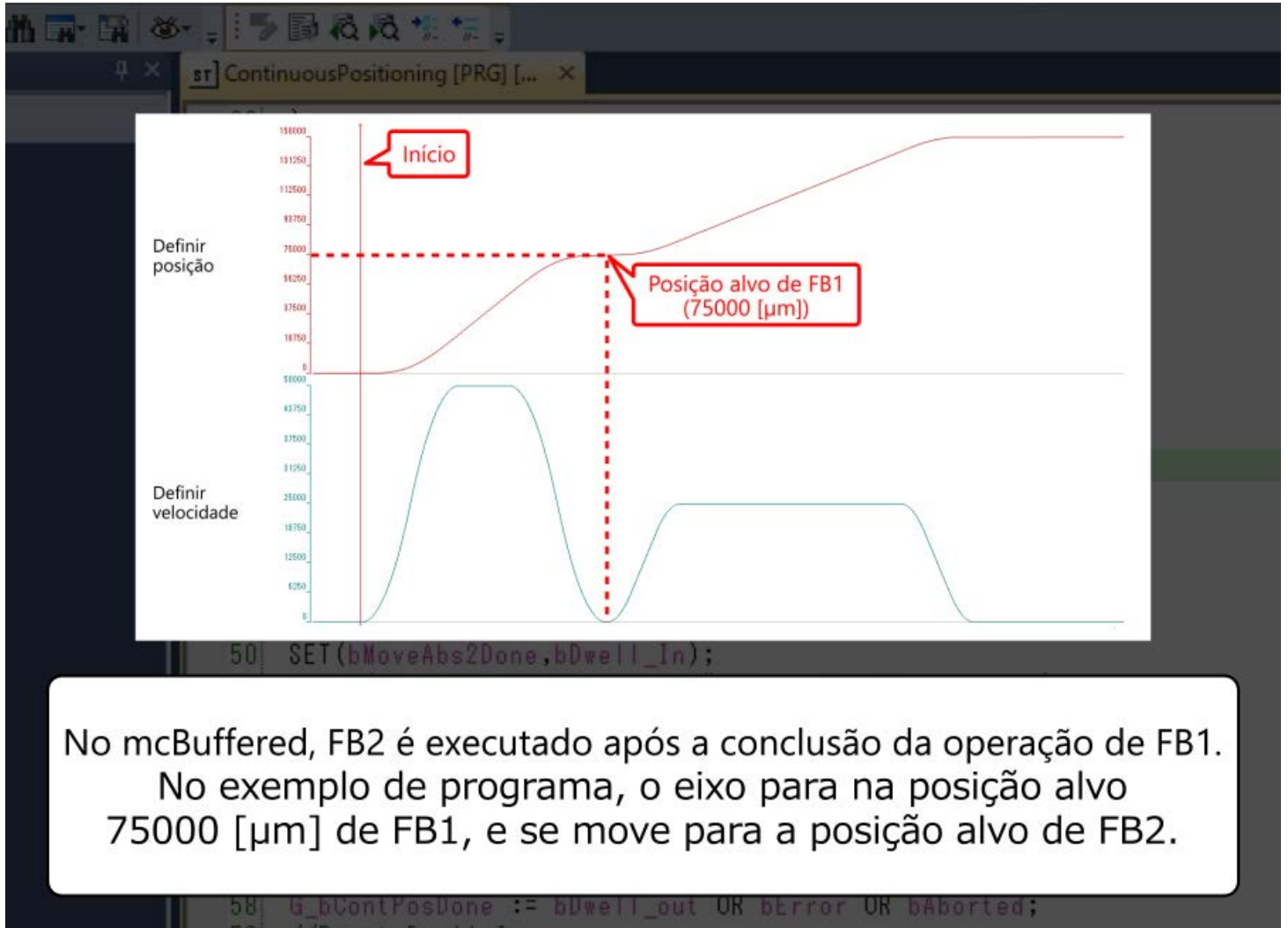
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcBuffered ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Em primeiro lugar, verifique a operação de mcBuffered. Verifique se a entrada BufferMode de MC_MoveAbsolute_2 está definida como "MC_BUFFER_MODE_mcBuffered", e grave o programa no módulo de movimento.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```



O motor é acionado quando X25 é ativado.
O gráfico mostra os valores Definir posição e Definir velocidade.

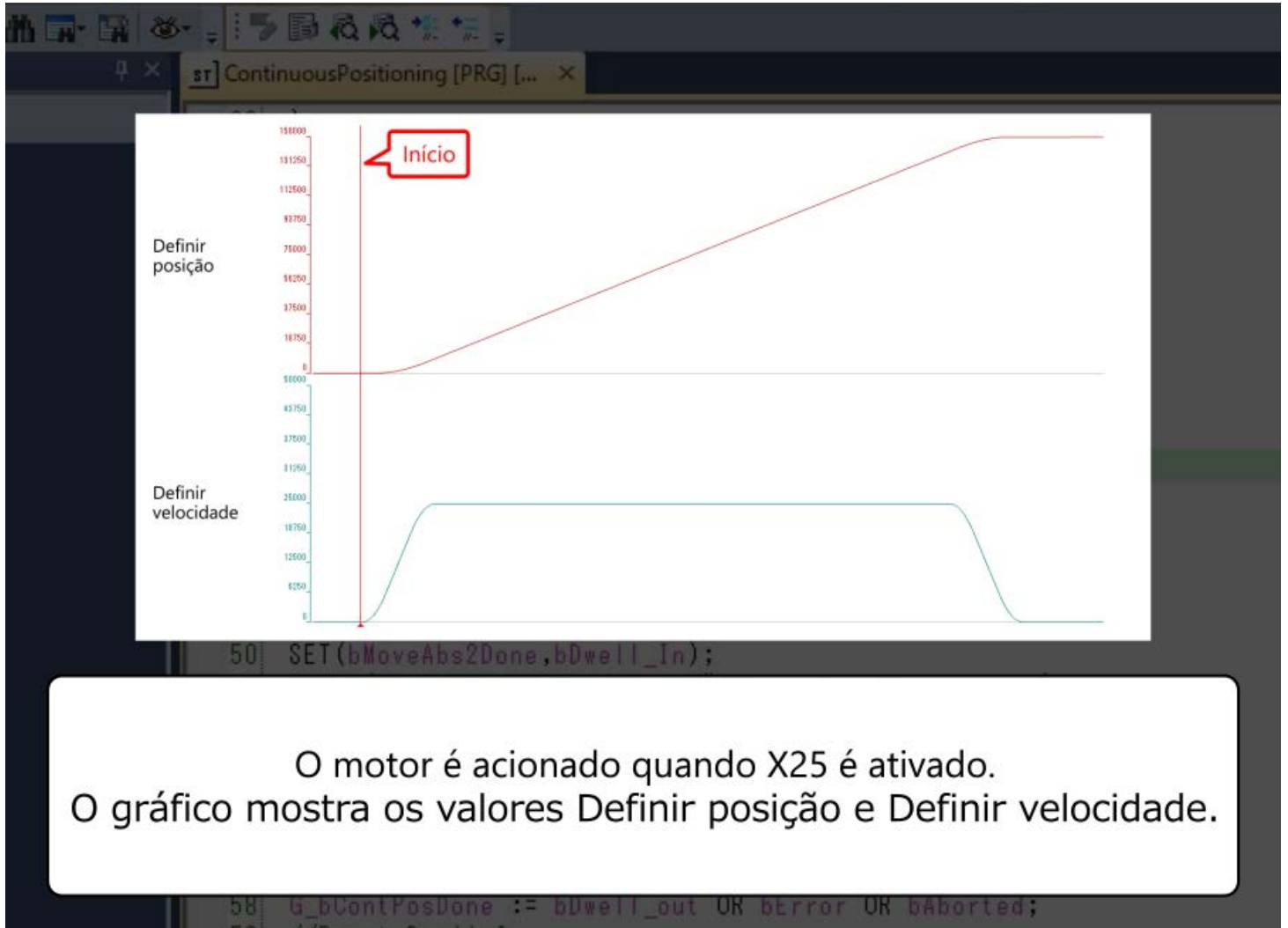


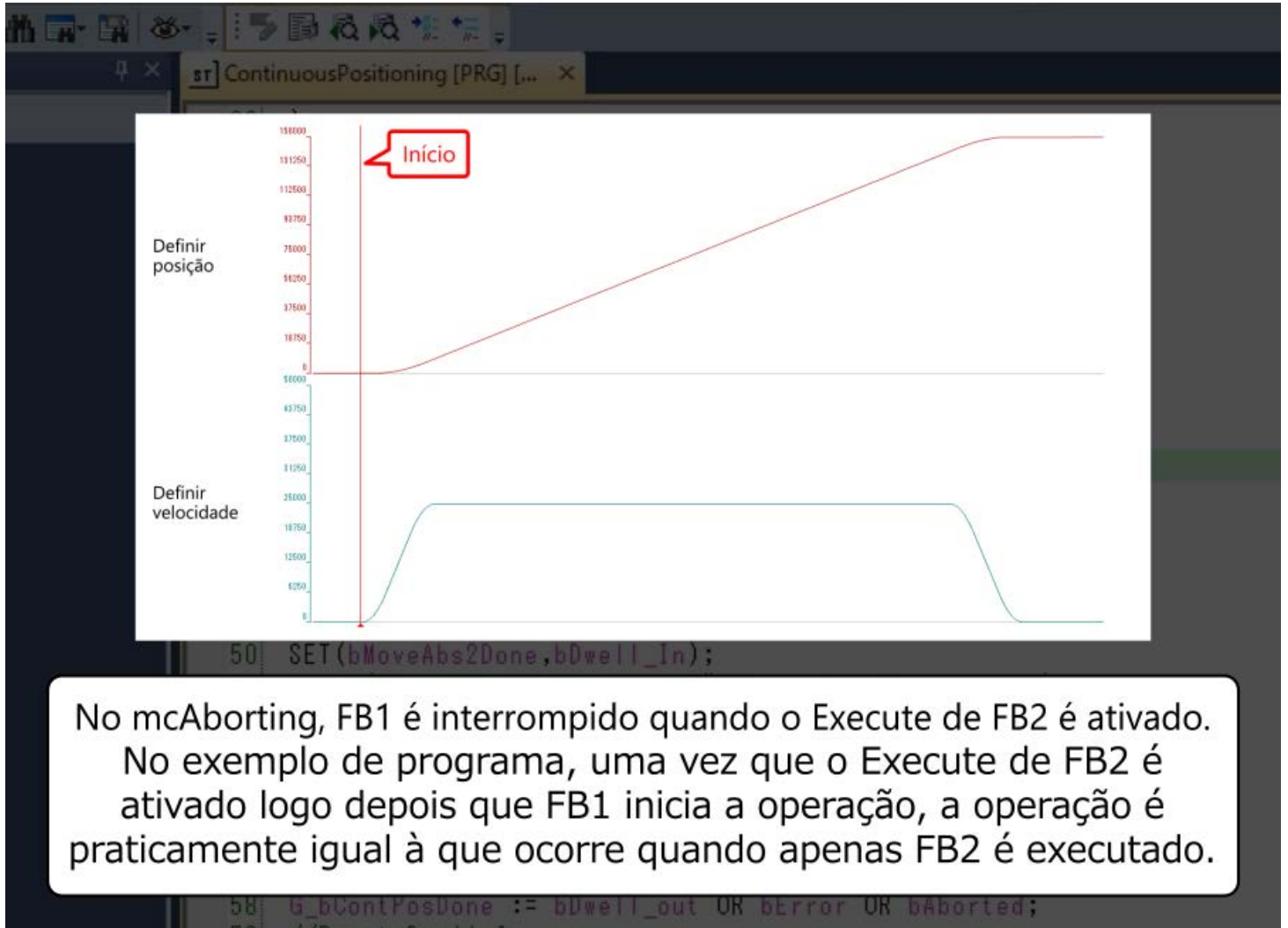
No mcBuffered, FB2 é executado após a conclusão da operação de FB1. No exemplo de programa, o eixo para na posição alvo 75000 [µm] de FB1, e se move para a posição alvo de FB2.

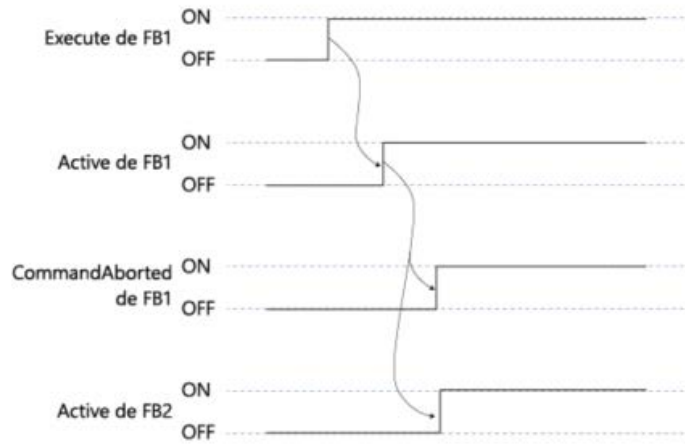

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcAborting ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

A seguir, verifique a operação de mcAborting.
Altere a entrada BufferMode de MC_MoveAbsolute_2 para "MC_BUFFER_MODE__mcAborting", reconstrua todos os programas e grave-os no módulo de movimento.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```





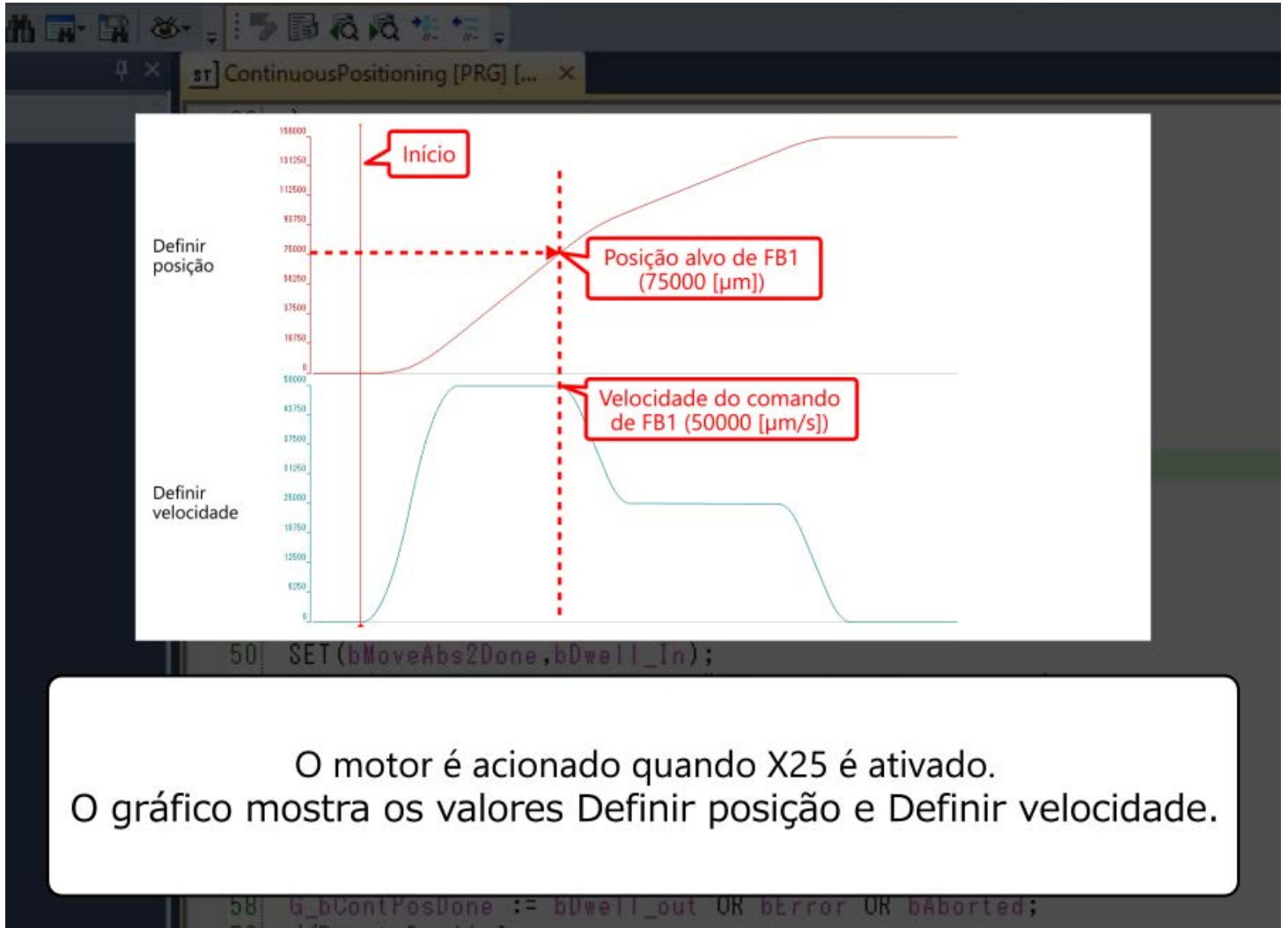


Verifique os sinais de E/S de FB1 e FB2 na hora de início.
A saída CommandAborted de FB1 é ativada,
indicando que FB1 é interrompido.

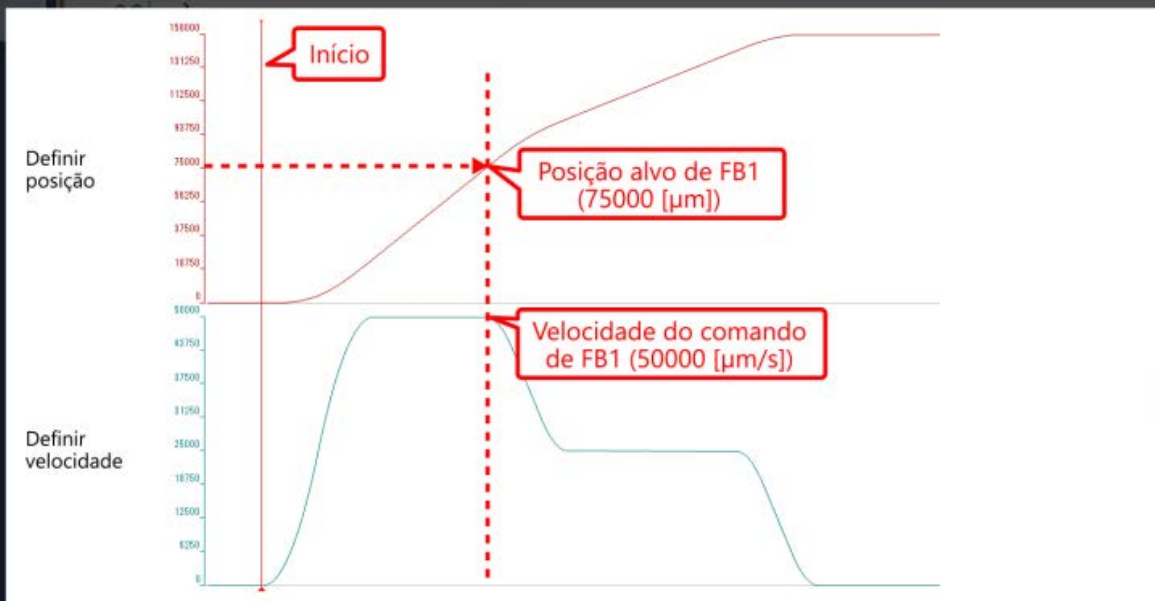
```
50 SET(bMoveAbs2Done,bDwell_In);  
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 /
```

```
33 );
34 //Positioning2
35 MC_MoveAbsolute_2(
36     Axis      := Axis0001.AxisRef ,
37     Execute   := bMoveAbs1Active ,
38     Position  := lePosition2 ,
39     Velocity  := lePosVelocity2 ,
40     Acceleration:= lePosAcceleration2 ,
41     Deceleration:= lePosDeceleration2 ,
42     Jerk      := lePosJerk2 ,
43     Direction := MC_DIRECTION__mcShortestWay ,
44     BufferMode := MC_BUFFER_MODE__mcBlendingPrevious ,
45     Done       => bMoveAbs2Done ,
46     CommandAborted => bMoveAbs2Aborted ,
47     Error      => bMoveAbs2Error
48 );
49 //Dwell
50 SET(bMoveAbs2Done,bDwell_In);
51
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```

A seguir, verifique a operação de mcBlendingPrevious. Altere a entrada BufferMode de MC_MoveAbsolute_2 para "MC_BUFFER_MODE__mcBlendingPrevious", reconstrua todos os programas e grave-os no módulo de movimento.



O motor é acionado quando X25 é ativado.
O gráfico mostra os valores Definir posição e Definir velocidade.



```
50 SET(bMoveAbs2Done,bDwell_In);
```

No `mcBlendingPrevious`, a velocidade do comando de FB1 é aplicada quando a posição alvo de FB1 é alcançada. Em seguida, a velocidade do comando de FB2 é aplicada, e o eixo se move para a posição alvo de FB2.

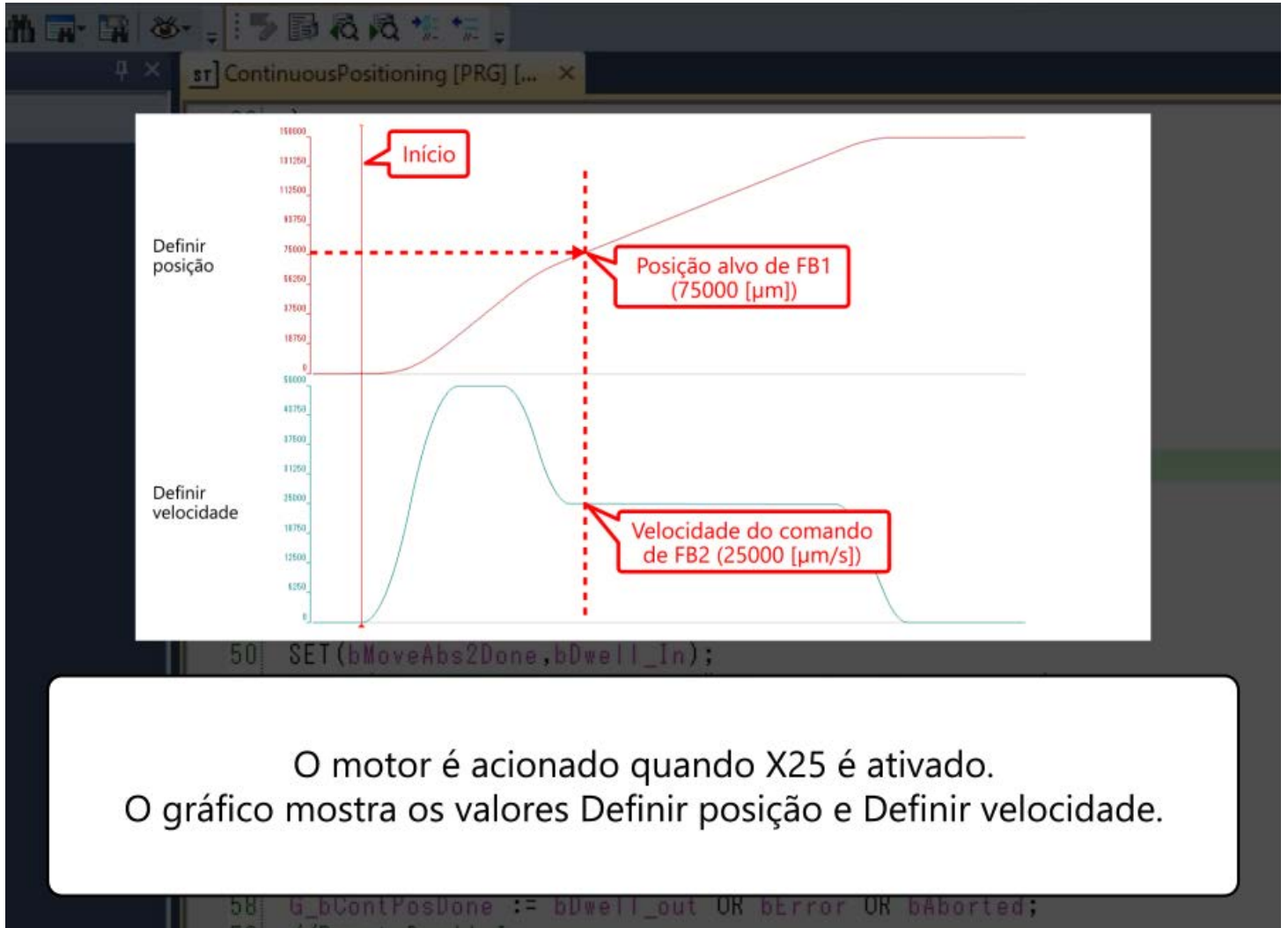
```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```

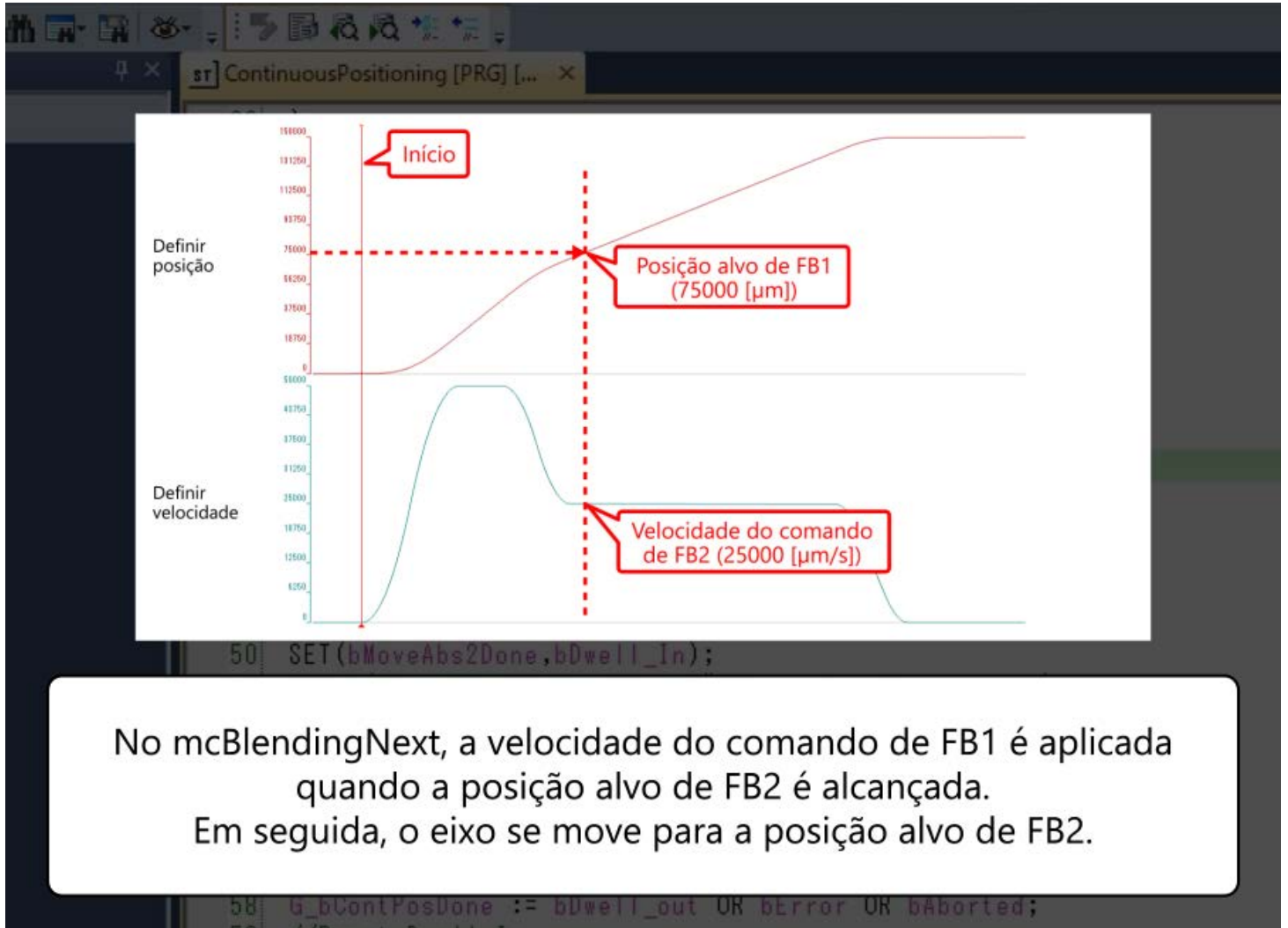


```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcBlendingNext ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

A seguir, verifique a operação de mcBlendingNext. Altere a entrada BufferMode de MC_MoveAbsolute_2 para "MC_BUFFER_MODE_mcBlendingNext", reconstrua todos os programas e grave-os no módulo de movimento.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```

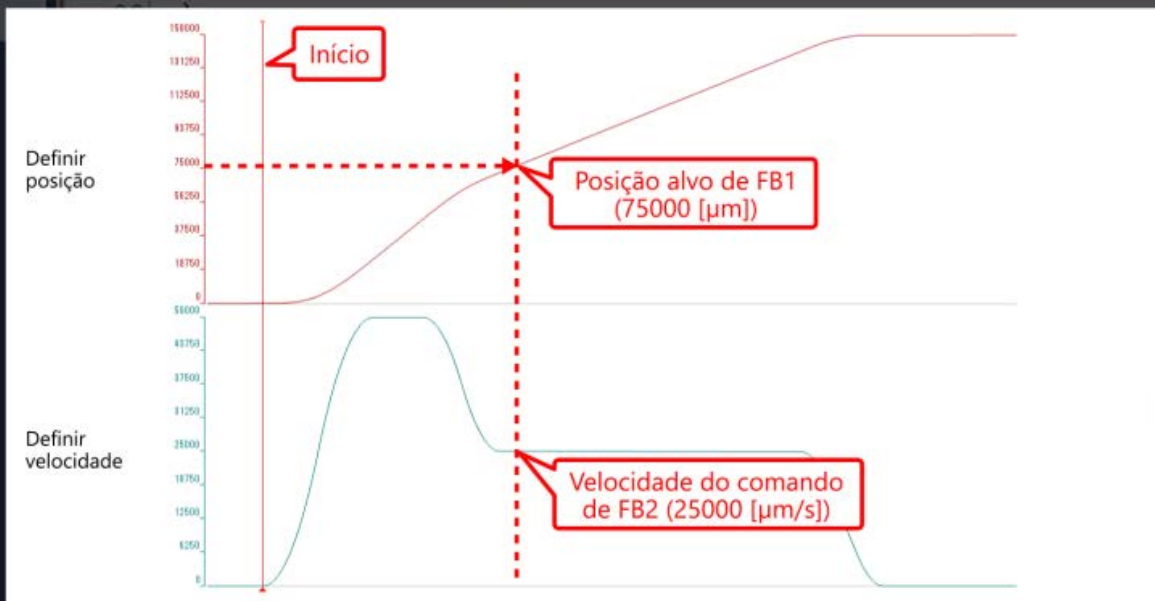


No mcBlendingNext, a velocidade do comando de FB1 é aplicada quando a posição alvo de FB2 é alcançada. Em seguida, o eixo se move para a posição alvo de FB2.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcBlendingLow ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

A seguir, verifique a operação de mcBlendingLow. Altere a entrada BufferMode de MC_MoveAbsolute_2 para "MC_BUFFER_MODE_mcBlendingLow", reconstrua todos os programas e grave-os no módulo de movimento.

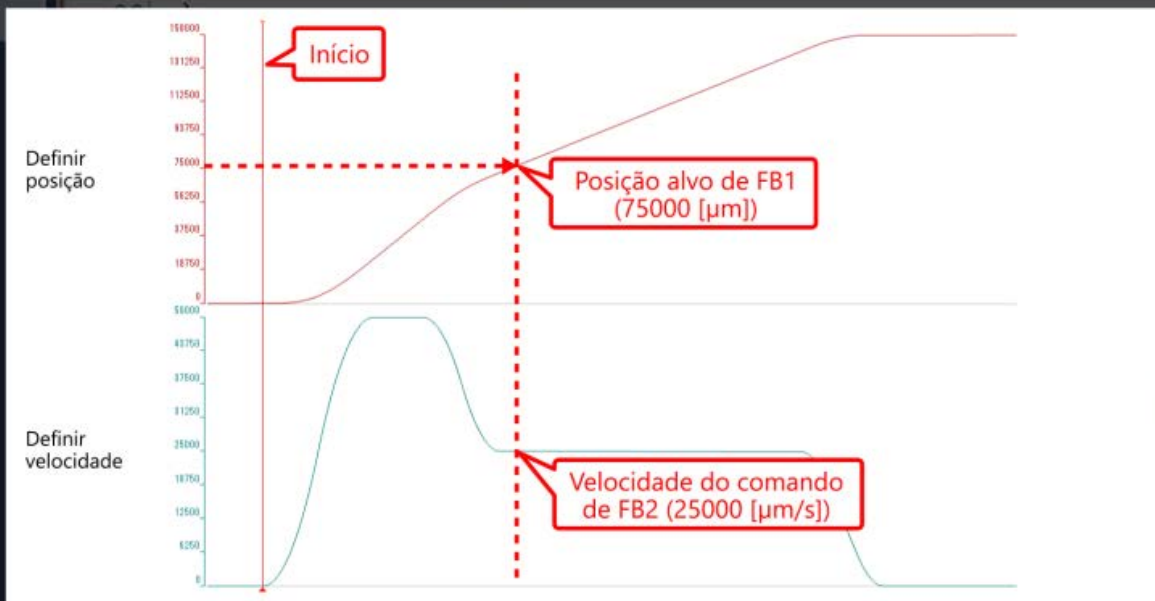
```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```



```
50 SET(bMoveAbs2Done,bDwell_In);
```

O motor é acionado quando X25 é ativado.
O gráfico mostra os valores Definir posição e Definir velocidade.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```



```
50 SET(bMoveAbs2Done,bDwell_In);
```

No mcBlendingLow, a velocidade mais lenta do comando é aplicada quando a posição alvo de FB1 é alcançada.

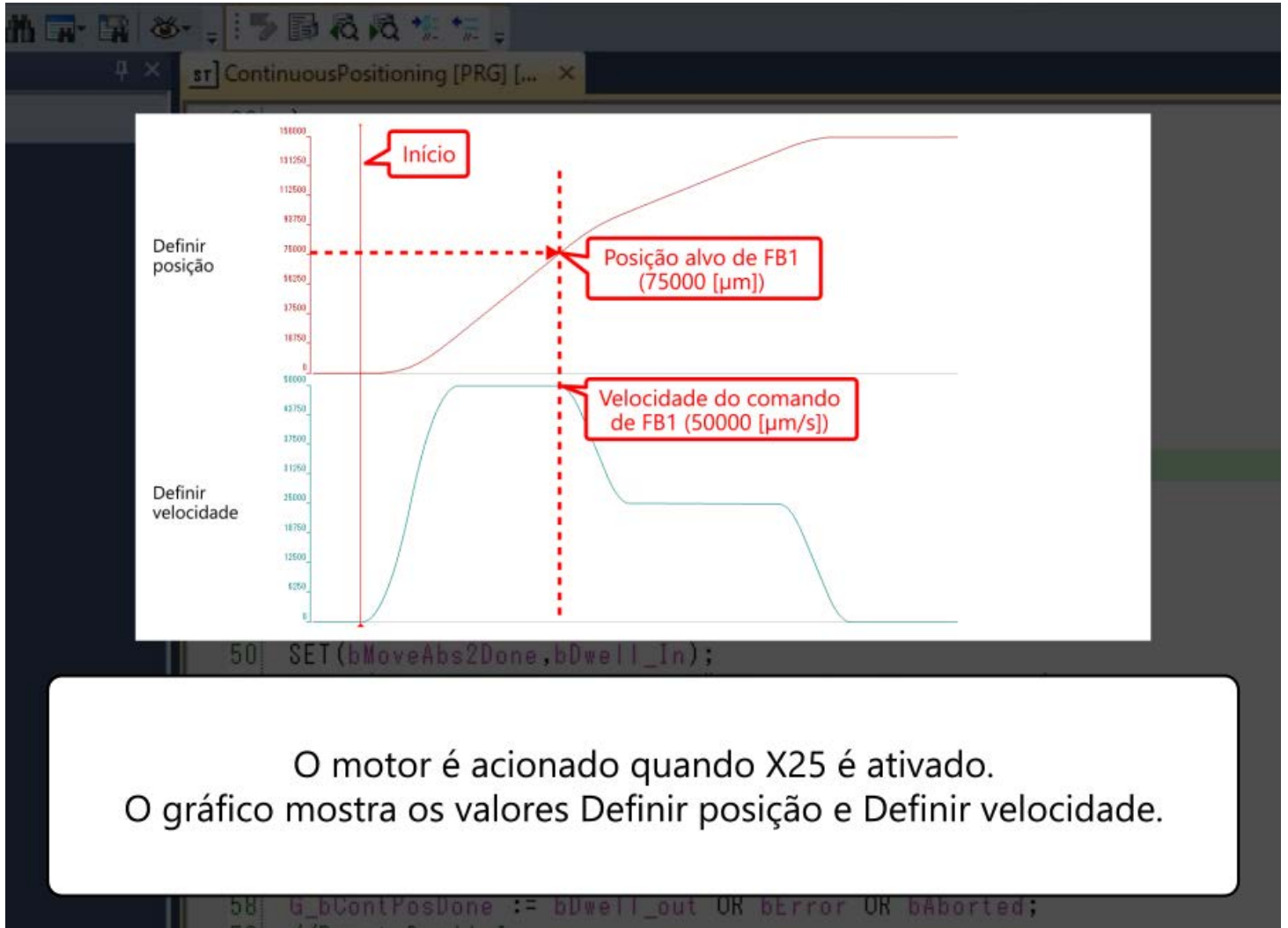
No exemplo deste curso, uma vez que a velocidade do comando de FB2 é mais lenta, a forma de onda da velocidade é a mesma de BlendingNext.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```

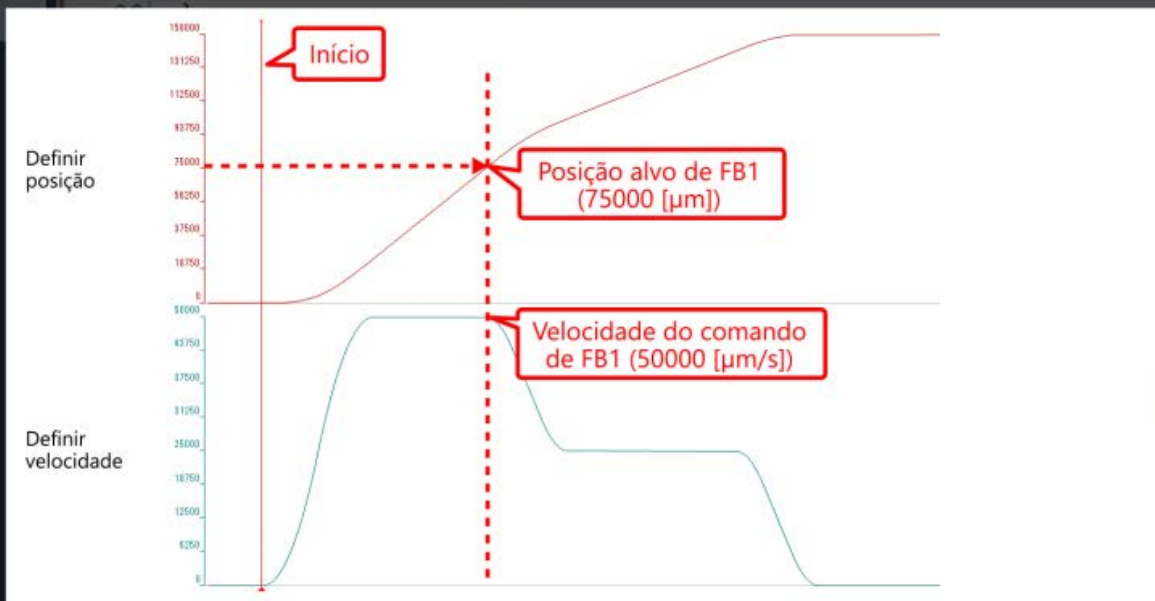
```
33 );
34 //Positioning2
35 MC_MoveAbsolute_2(
36     Axis      := Axis0001.AxisRef ,
37     Execute   := bMoveAbs1Active ,
38     Position  := lePosition2 ,
39     Velocity  := lePosVelocity2 ,
40     Acceleration:= lePosAcceleration2 ,
41     Deceleration:= lePosDeceleration2 ,
42     Jerk      := lePosJerk2 ,
43     Direction  := MC_DIRECTION_mcShortestWay ,
44     BufferMode := MC_BUFFER_MODE_mcBlendingHigh ,
45     Done       => bMoveAbs2Done ,
46     CommandAborted => bMoveAbs2Aborted ,
47     Error      => bMoveAbs2Error
48 );
49 //Dwell
50 SET(bMoveAbs2Done,bDwell_In);
```

Por último, verifique a operação de mcBlendingHigh. Altere a entrada BufferMode de MC_MoveAbsolute_2 para "MC_BUFFER_MODE_mcBlendingHigh", reconstrua todos os programas e grave-os no módulo de movimento.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
59 //Dwell
```

O motor é acionado quando X25 é ativado.
O gráfico mostra os valores Definir posição e Definir velocidade.

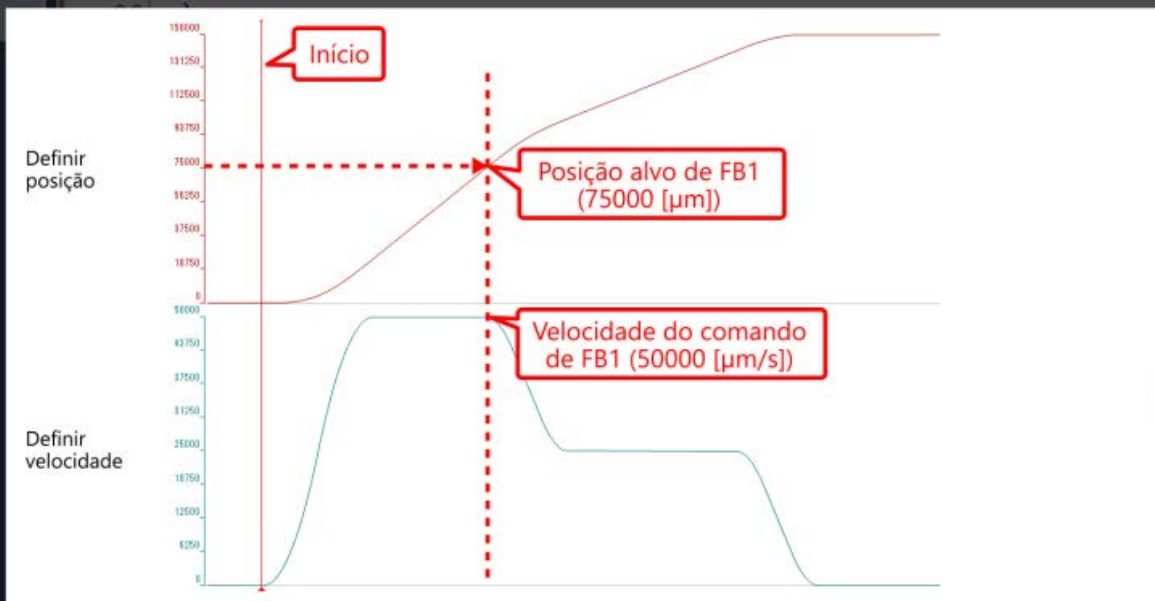


```
50 SET(bMoveAbs2Done,bDwell_In);
```

No mcBlendingHigh, a velocidade mais rápida do comando é aplicada quando a posição alvo de FB1 é alcançada.

No exemplo deste curso, uma vez que a velocidade do comando de FB1 é mais rápida, a forma de onda da velocidade é a mesma de BlendingPrevious.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```

Isto conclui a verificação da operação de BufferMode.
Vá para a próxima página.

```
50 SET(bMoveAbs2Done,bDwell_In);  
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 /
```

Neste capítulo, você aprendeu o seguinte:

- Anulação
- Armazenamento temporário
- Mesclagem
- Exemplo de programa
- Verificação da operação

Ponto

Anulação	<ul style="list-style-type: none"> • Quando um tipo de operação FB está em execução e o próximo FB do tipo operação é executado, o comando de Anulação interrompe o FB em execução e executa o próximo FB.
Armazenamento temporário	<ul style="list-style-type: none"> • Quando um FB do tipo operação está em execução e o próximo FB do tipo operação é executado, o comando Buffered espera até que o FB em execução seja concluído, para a seguir executar o próximo FB.
Mesclagem	<ul style="list-style-type: none"> • Quando um FB do tipo operação está em execução e o próximo FB do tipo operação é executado, o comando Blending executa o próximo FB sem interromper a operação do FB atual. • No comando Blending, existem quatro métodos de mudança de velocidade: BlendingLow, BlendingHigh, BlendingPrevious e BlendingNext.
Exemplo de programa	<ul style="list-style-type: none"> • Selecione o modo de buffer com a entrada BufferMode de um FB do tipo operação.
Verificação da operação	<ul style="list-style-type: none"> • Você verificou a diferença na operação de cada modo de buffer apresentado no vídeo.

Capítulo 4 | Operação com a CPU do PLC

Baixe o exemplo de programa que será usado neste capítulo, clicando no link abaixo.
O conteúdo do programa é o mesmo do exemplo de programa descrito nos capítulos 2 e 3.
O que muda é apenas o método de programação.

[RD78GBasic2_sample2.zip \(1.39 MB\)](#)

4.1 | Registrando a biblioteca de FBs do módulo de movimento

(1) Baixe a biblioteca de FBs

O Motion control FB pode ser usado no programa da CPU do PLC,
registrando-se a biblioteca de FBs correspondente ao módulo de movimento no GX Works3.

Baixe a biblioteca de FBs usando o link abaixo, e descompacte o arquivo ZIP no destino que desejar.

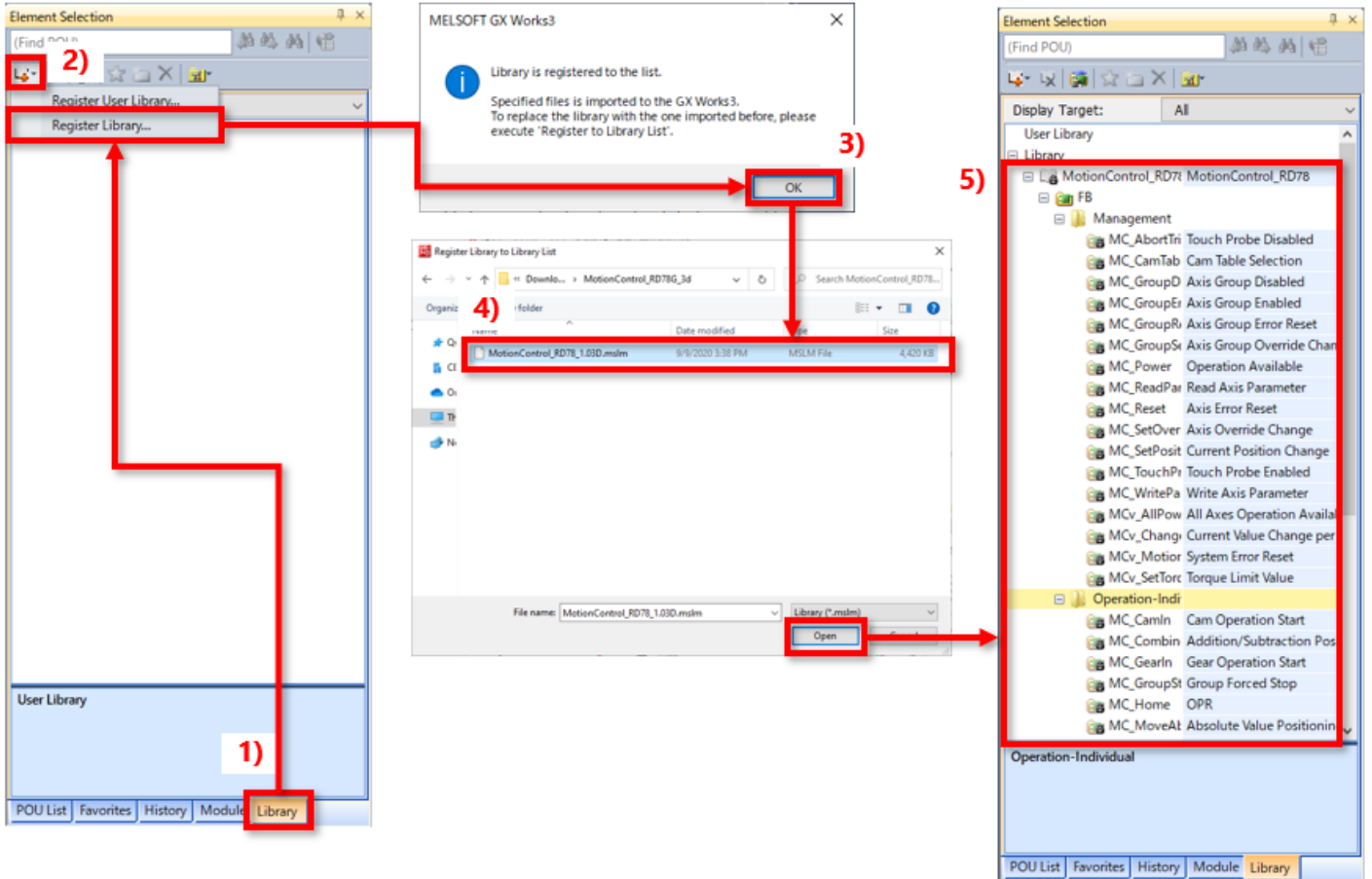
[MotionControl_RD78G_3d.zip\(4.29 MB\)](#)

(Nota) A última versão da biblioteca de FBs pode ser baixada no site global da MITSUBISHI ELECTRIC FA.

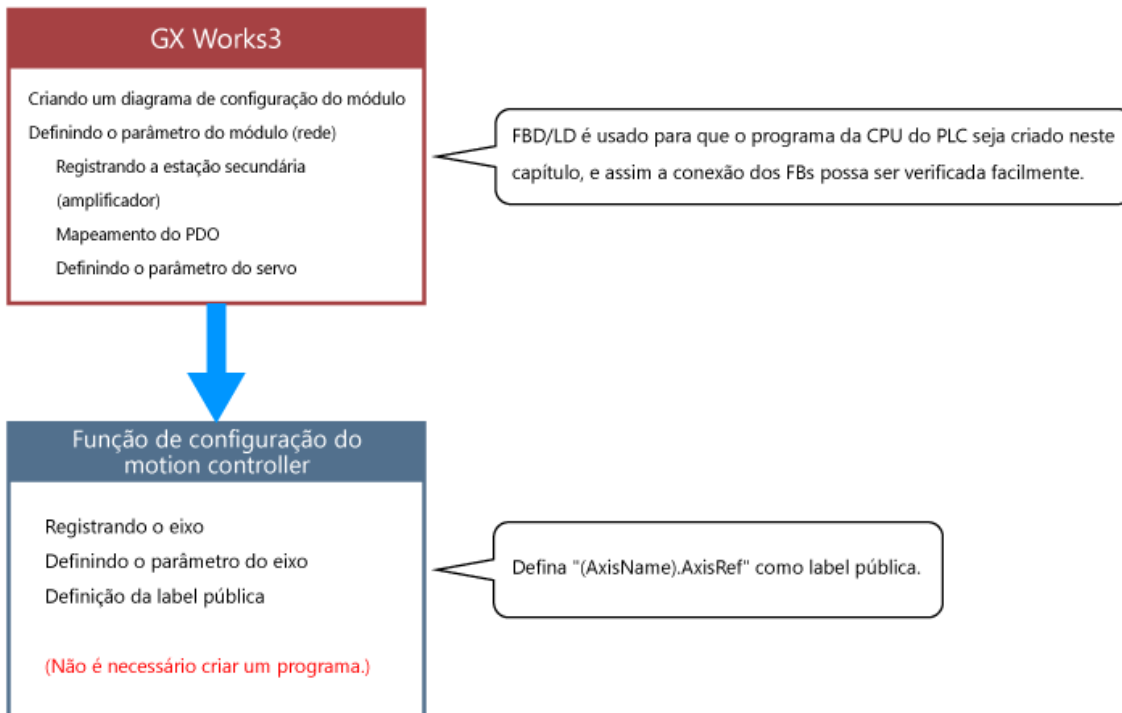
4.1 Registrando a biblioteca de FBs do módulo de movimento

(2) Registrando a biblioteca de FBs

- 1) Abra qualquer projeto no GX Works3, e abra a guia Library na janela Element Selection.
- 2) Clique no botão [Register to Library List], na parte superior, e selecione [Register Library].
- 3) Quando aparecer a mensagem "Library is registered to the list", clique em [OK].
- 4) Selecione o arquivo de biblioteca de FBs "MotionControl_RD78_****.mslm", e clique em [Open].
(**** indica a versão.)
- 5) O Motion control FB é registrado na biblioteca na janela Element Selection.



O procedimento de criação de um projeto é o mesmo descrito na seção anterior.



MELSOFT GX Works3 E: Sample.gx3 - [Global [Global Label Setting]]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Element Selection (Find POU) Display Target: All User Library Library

Label Name	Data Type	English (Display Target)	Access from External Device
G.JogFw	Bit	JOG Forward	<input type="checkbox"/>
G.JogBw	Bit	JOG Backward	<input type="checkbox"/>
G.JogVelocity	FLOAT (Double Precision)	JOG Velocity	<input type="checkbox"/>
G.JogBusy	Bit	JOG Busy	<input type="checkbox"/>

Clique no botão Executar.

Extended Display: Do Not Show Always

System label is reserved to be registered. System label is reserved to be released. The system label is already registered to the

To execute the Reservation to Register/Release for the system label, reflection to the system label database is required. It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
 * Only Q-R series/GOT 2000 series is available for system label Ver.2.
 * To execute Online Program Change, execute Online Program Change and save.

Reservation to Register System Label
 Reservation to Release System Label
 Import System Label

Not Reflected: 0
 Total: 0

Library POU... Favorit... History Mod... Library

Connection Des... Navigation Output Progress R04 Host Row 1 Column 1 CAP NUM

MELSOFT GX Works3 E: Sample.gx3 - [Global [Global Label Setting]]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Label Name	Data Type	English (Display Target)	Access from External Device
G.JogFW	Bit	JOG Forward	<input type="checkbox"/>
G.JogB#	Bit	JOG Backward	<input type="checkbox"/>
G.JogVelocity	FLOAT [Double Precision]	JOG Velocity	<input type="checkbox"/>
G.JogBusy	Bit	JOG Busy	<input type="checkbox"/>

Este vídeo mostra como criar um programa de FB (MCv_Jog) para a operação JOG, como exemplo.

Extended Display: Do Not Show Always

System label is reserved to be registered. System label is reserved to be released. The system label is already registered to the

To execute the Reservation to Register/Release for the system label, reflection to the system label database is required. Please execute 'Reflect to System Label Database'. It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
 * Only Q-R series/GOT 2000 series is available for system label Ver.2.
 * To execute Online Program Change, execute Online Program Change and save.

Reservation to Register System Label
 Reservation to Release System Label
 Import System Label

Not Reflected: 0
 Total: 0

Element Selection (Find POU) Display Target: All User Library Library

Library POU... Favori... History Mod... Library

Output Progress R04 Host Row 1 Column 1 CAP NUM

Registere o comando JOG, a velocidade de JOG e Jog busy na label global.
(A velocidade de JOG é registrada na label global, considerando que poderá ser definida a partir dos dispositivos externos, como o GOT.)

Label Name	Data Type	English/Display Target	Access from External Device
G.JogFW	Bit	JOG Forward	<input type="checkbox"/>
G.JogBW	Bit	JOG Backward	<input type="checkbox"/>
G.JogVelocity	FLOAT [Double Precision]	JOG Velocity	<input type="checkbox"/>
G.JogBusy	Bit	JOG Busy	<input type="checkbox"/>

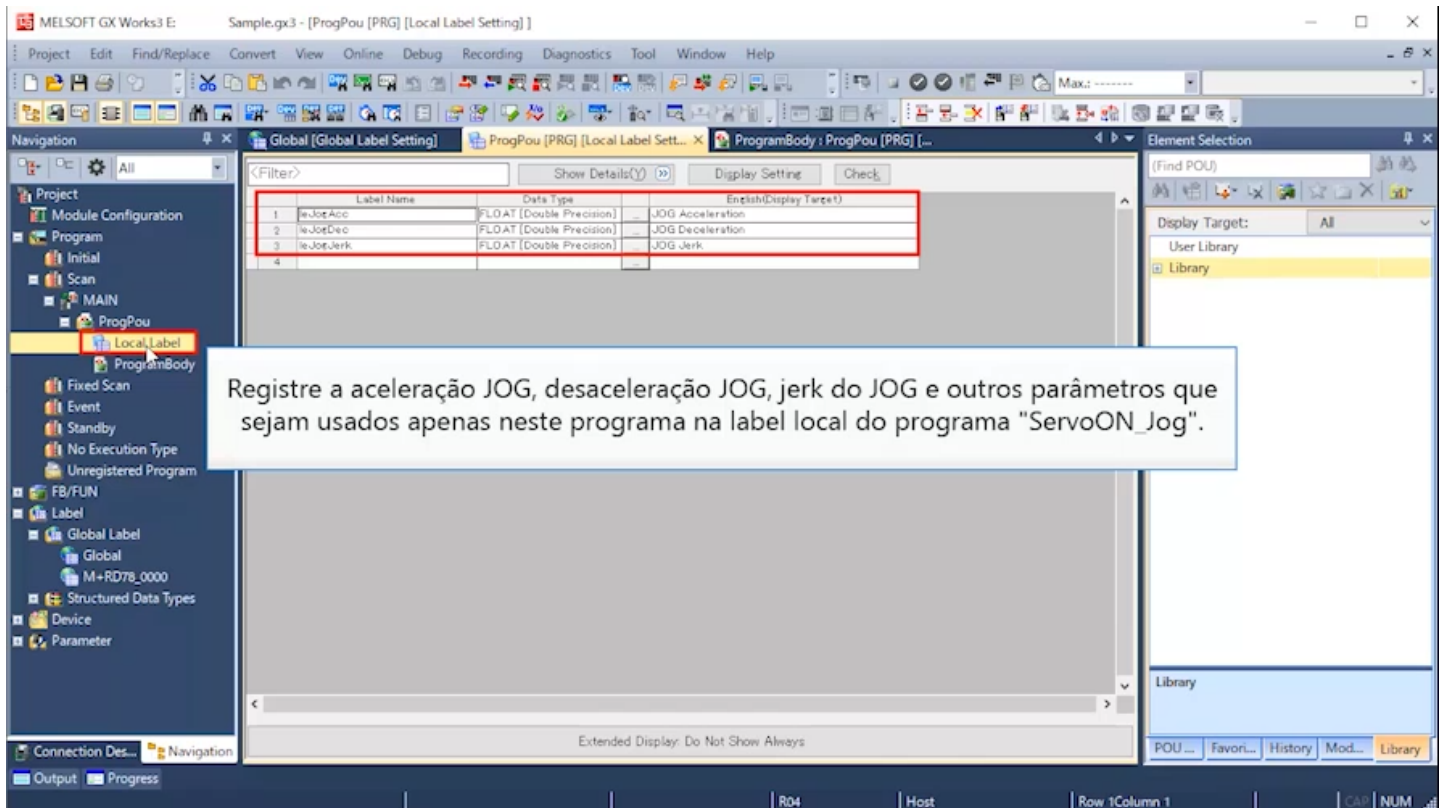
Extended Display: Do Not Show Always

System label is reserved to be registered. System label is reserved to be released. The system label is already registered to the

To execute the Reservation to Register/Release for the system label, reflection to the system label database is required. Please execute 'Reflect to System Label Database'. It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
* Only Q-R series/GOT 2000 series is available for system label Ver.2.
* To execute Online Program Change, execute Online Program Change and save.

Reservation to Register System Label
Reservation to Release System Label
Import System Label

Not Reflected: 0
Total: 0



Sample.gx3 - [ProgPou [PRG] [Local Label Setting]]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... x] ProgramBody : ProgPou [PRG] [...]

Label Name	Data Type	English/Display Target
!-JogAcc	FLOAT [Double Precision]	JOG Acceleration
!-JogDec	FLOAT [Double Precision]	JOG Deceleration
!-JogJerk	FLOAT [Double Precision]	JOG Jerk

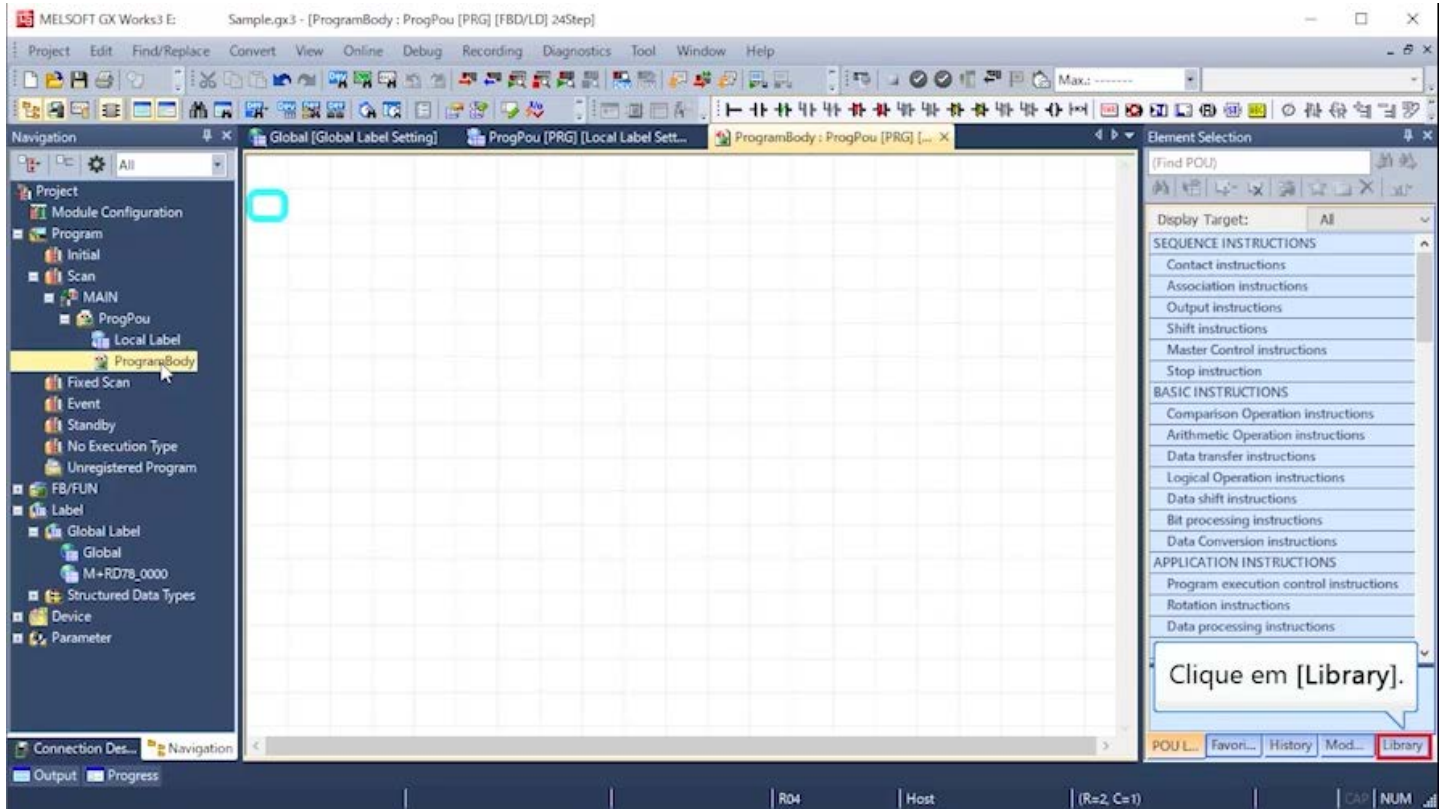
Element Selection (Find POU) Display Target: All User Library Library

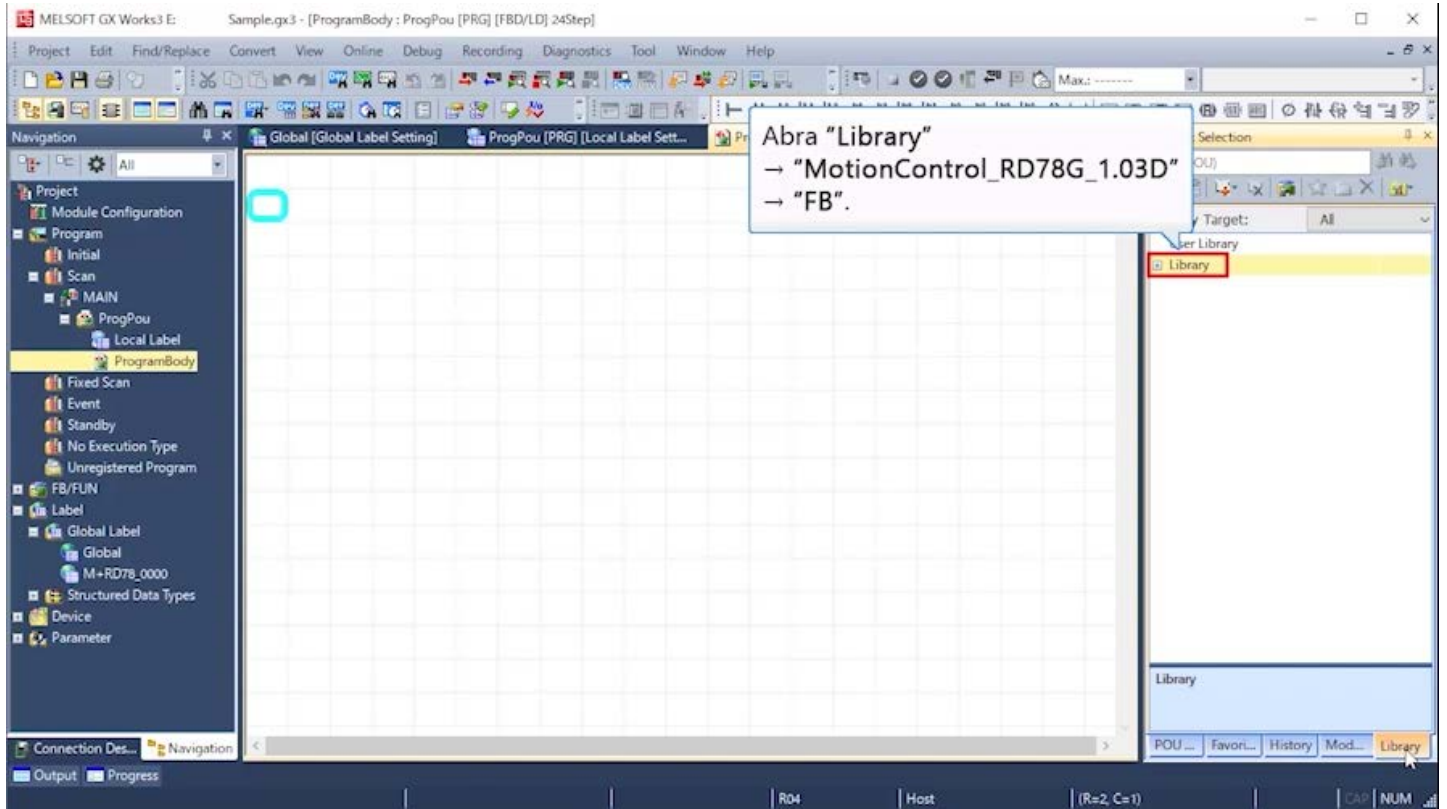
Library POU... Favorit... History Mod... Library

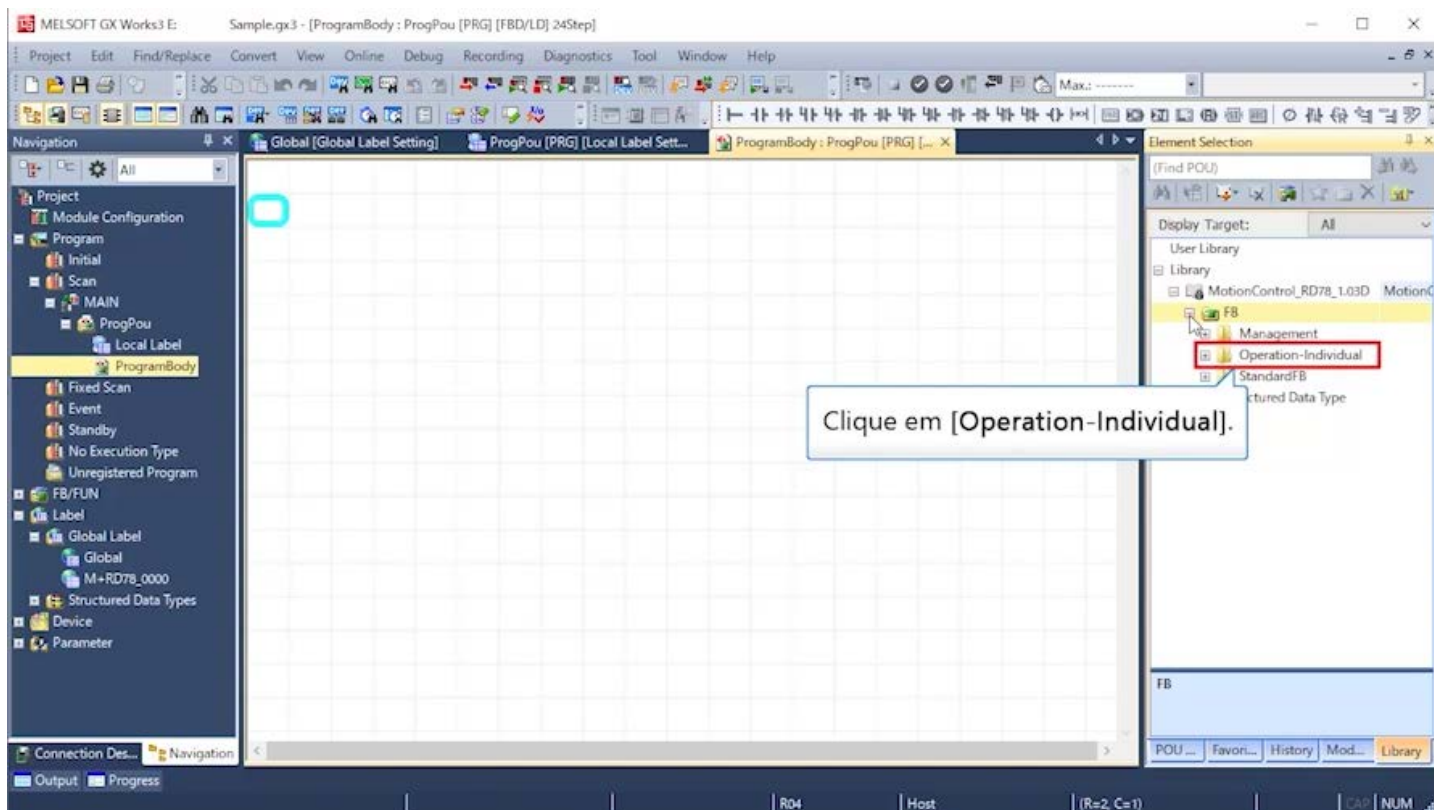
Extended Display: Do Not Show Always

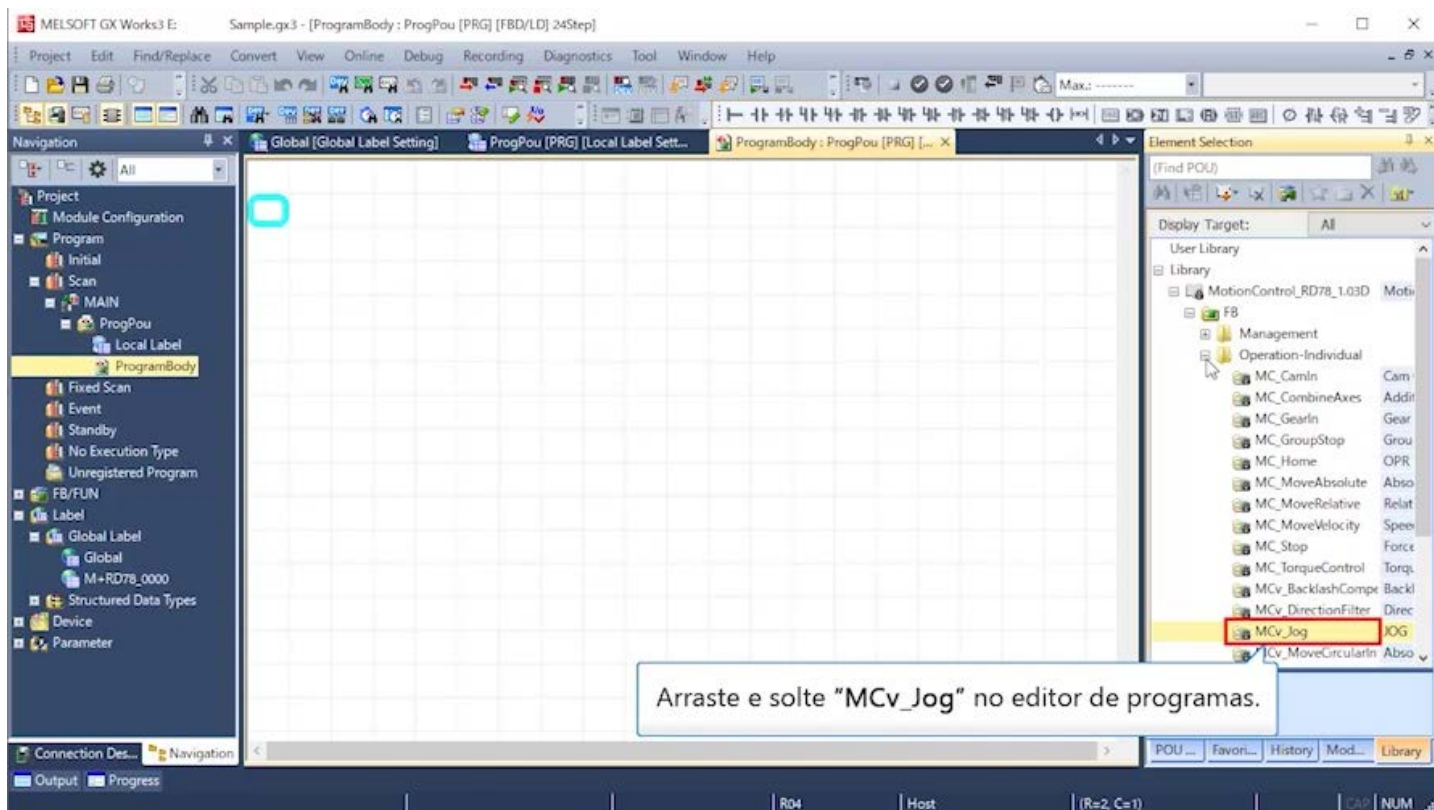
R04 Host Row 1Column 1 CAP NUM

Registere a aceleração JOG, desaceleração JOG, jerk do JOG e outros parâmetros que sejam usados apenas neste programa na label local do programa "ServoON_Jog".



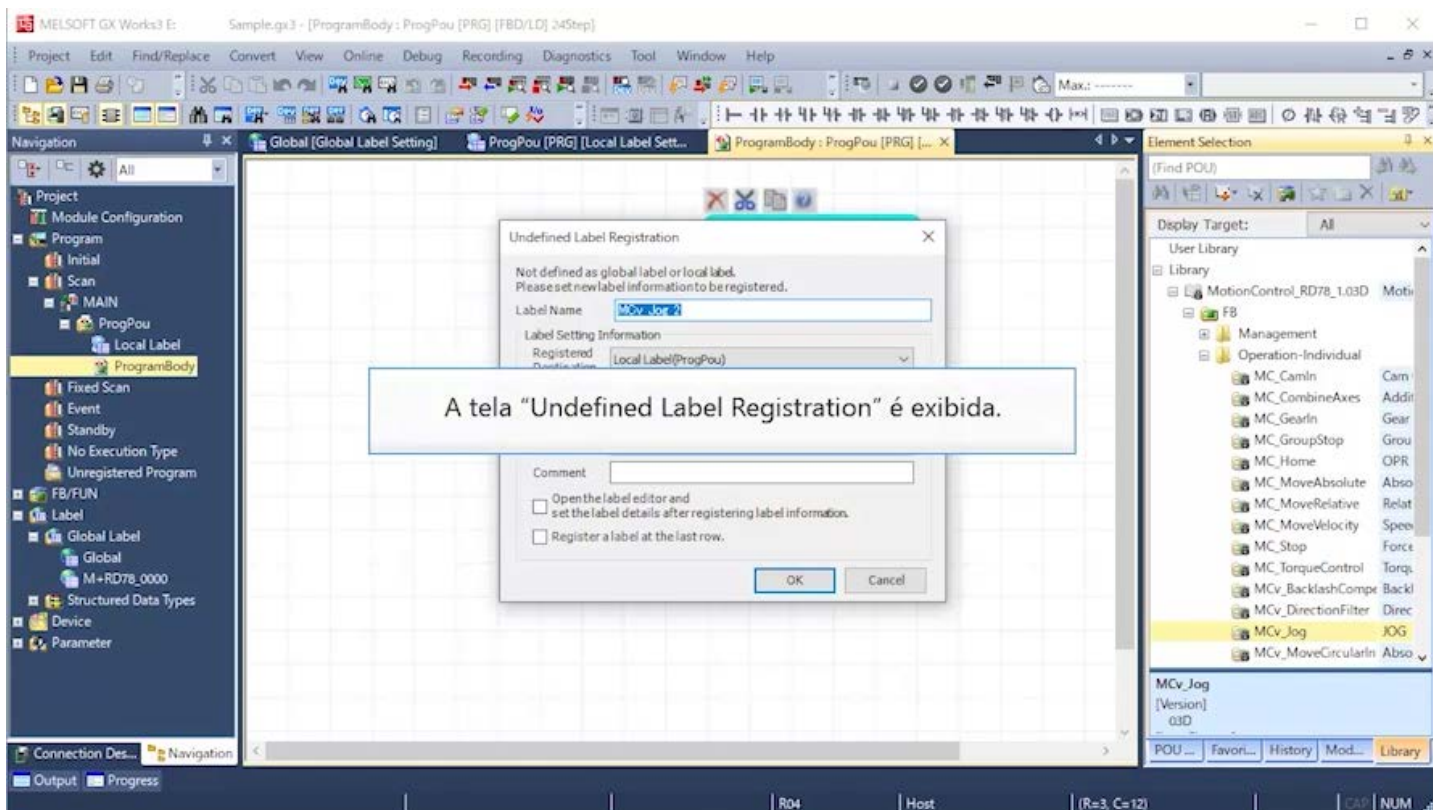






The screenshot displays the MELSOFT GX Works3 E software interface. The main window shows a ladder logic editor with a grid. On the left, the 'Navigation' pane shows the project structure, with 'ProgramBody' selected. On the right, the 'Element Selection' window is open, showing a library of motion control functions. The 'MCv_Jog' function is highlighted with a red box. A text box at the bottom of the screen contains the instruction: "Arraste e solte 'MCv_Jog' no editor de programas."

Arraste e solte "MCv_Jog" no editor de programas.



MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [... x]

Element Selection (Find POU)

Display Target: All

User Library

Library

- MotionControl_RD78_1.03D Moti
- FB
- Management
- Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addri
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Speec
 - MC_Stop Forcc
 - MC_TorqueControl Torq
 - MCv_BacklashComp Backl
 - MCv_DirectionFilter Direc
 - MCv_Jog JOG
 - MCv_MoveCircularIn Abso

MCv_Jog [Version] 03D

POU... Favori... History Mod... Library

Connection Des... Navigation

Output Progress

R04 Host (R=3, C=12) CAP NUM

Undefined Label Registration

Not defined as global label or local label. Please set new label information to be registered.

Label Name MCv_log_2

Label Setting Information

Registered Destination Local Label(ProgPou)

Class VAR

Data Type MCv_log

Constant

Comment

Open the label editor and set the label details after registering label information.

Register a label at the last row.

OK Cancel

Clique em [OK].

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a red square indicating the insertion point for a new element. A callout box with the text "Clique na área onde o elemento LD será inserido." (Click on the area where the LD element will be inserted.) points to this square. A blue-bordered box highlights the "MCv_Jog" element in the library, which is currently selected. The element's properties are visible in the "Element Selection" panel on the right, including "JogForward", "JogBackward", "Velocity", "Acceleration", "Deceleration", "Jerk", and "Options". The "User Library" panel on the right shows the "MotionControl_RD78_1.03D" library, with "MCv_Jog" selected under the "JOG" category. The status bar at the bottom indicates "R04", "Host", "(R=3, C=12)", and "CAP NUM".

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a callout box containing the text "Clique no ícone do elemento LD." (Click on the LD element icon). The callout box highlights a specific function block (FB) named "MDV_Jog_2" with the following parameters:

Parameter	Value
JogForward	Done
JogBackward	Busy
Velocity	Active
Acceleration	CommandAborted
Deceleration	Error
Jerk	ErrorID
Options	
Axis	1

The right-hand side of the interface shows the "Element Selection" panel with a tree view of the "MotionControl_RD78_1.03D" library. The "MCv_Jog" element is highlighted. The status bar at the bottom indicates "R04 | Host | (R=3, C=4) | CAP | NUM".

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a function block named "MCv_Jog" (MDV_Jog) being inserted. A callout box points to the LD element being added, with the text: "O elemento LD especificado é inserido." (The specified LD element is inserted.)

The interface includes a menu bar (Project, Edit, Find/Replace, Convert, View, Online, Debug, Recording, Diagnostics, Tool, Window, Help), a toolbar, and several panels:

- Navigation Panel:** Shows the project structure, including "ProgramBody" and "Local Label".
- Element Selection Panel:** Displays a list of motion control function blocks, such as "MCv_Jog" (JOG), "MCv_MoveRelative" (Relat), and "MCv_MoveAbsolute" (Abso).
- Main Workspace:** Shows the ladder logic diagram with the "MCv_Jog" block and its parameters (e.g., Acceleration, Deceleration, Jerk, Options, Axis).

The status bar at the bottom indicates the current position (R04), host status (Host), and coordinates (R=3, C=4).

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a callout box that reads "Clique duas vezes em [???]". The callout box is positioned over a network of logic involving a normally open contact labeled "???", a normally closed contact labeled "???", and a function block labeled "MCv_Jog_2". The function block has several input and output ports: Velocity, Acceleration, Deceleration, Jerk, Options, Done, Busy, Active, Error, and ErrorID. The output port "Axis" is also visible.

The left sidebar shows the Project tree structure, including "Program", "Scan", "Local Label", and "ProgramBody". The right sidebar shows the Element Selection panel, which lists the Motion Control library. The "MCv_Jog" function block is highlighted in the library, and its details are shown in the bottom right corner of the Element Selection panel.

The bottom status bar displays the current position and coordinates: R04 | Host | (R=3, C=8) | CAP | NUM.

MELSOFT GX Works3 E Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] **Insira o nome da label do dispositivo de bit.** Element Selection

Display Target: All

User Library

Library

- MotionControl_RD78_1.03D Moti
 - FB
 - Management
 - Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addri
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Speec
 - MC_Stop Force
 - MC_TorqueControl Torq
 - MCv_BacklashComp Backl
 - MCv_DirectionFilter Direc
 - MCv_Jog JOG**
 - MCv_MoveCircularIn Abso

MCv_Jog [Version] 03D

POU ... Favori... History Mod... Library

Global [Global Label Setting]

Global Label Setting

G_bJogBusy	BOOL	JOG Busy
G_bJogBW	BOOL	JOG Backward
G_bJogFW	BOOL	JOG Forward
G_leJogVelocity	LREAL	JOG Velocity

Settings...

MDv_jog_2 MDv_jog

JogForward Done

JogBackward Busy

Velocity Active

Acceleration CommandAborted

Deceleration Error

Jerk ErrorID

Options

Axis

1

Connection Des... Navigation

Output Progress

R04 Host (R=3, C=5) CAP NUM

The screenshot shows the MELSOFT GX Works3 E interface. The main workspace displays a ladder logic diagram with a callout box containing the text: "As labels já registradas nas labels globais são exibidas como alternativas." (Labels already registered in global labels are displayed as alternatives.)

The callout box points to a table of global labels:

Label	Type	Description
G_bJogBusy	BOOL	JOG Busy
G_bJogBW	BOOL	JOG Backward
G_bJogFW	BOOL	JOG Forward
G_leJogVelocity	LREAL	JOG Velocity

The diagram also shows a function block named MDV_Jog_2 with inputs: JogForward, JogBackward, Velocity, and Options. The Options input is connected to a block labeled Axis 1.

The right-hand side of the interface shows the Element Selection pane with a tree view of the User Library. The selected element is MCv_Jog, which is highlighted in yellow. The bottom status bar shows R04, Host, (R=3, C=5), CAP, and NUM.

The screenshot displays the MELSOFT GX Works3 E interface. The main workspace shows a ladder logic diagram with a normally closed contact labeled 'G_bJogFW' and a coil labeled 'G_bJogBW'. A red box highlights the coil, and a callout box contains the text: "Conecte o elemento LD à entrada JogForward do FB." To the right, a function block 'MDV_Jog_2' is shown with various inputs and outputs. The 'Element Selection' panel on the right lists the 'MCv_Jog' block under the 'MotionControl_RD78_1.03D' library.

Navigation: Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody: ProgPou [PRG] [...]

Element Selection: (Find POU) Display Target: All User Library Library MotionControl_RD78_1.03D Moti FB Management Operation-Individual MC_CamIn Cam MC_CombineAxes Addri MC_GearIn Gear MC_GroupStop Grou MC_Home OPR MC_MoveAbsolute Abso MC_MoveRelative Relat MC_MoveVelocity Speec MC_Stop Forcc MC_TorqueControl Torq MCv_BacklashComp Backl MCv_DirectionFilter Direc MCv_Jog JOG MCv_MoveCircularIn Abso

MCv_Jog [Version] 03D

POU... Favori... History Mod... Library

R04 Host (R=3, C=8) CAP NUM

Este pode ser o aspecto, quando o elemento LD for conectado à entrada JogForward e à entrada JogBackward.

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with two normally open contacts labeled 'G.JogFW' (ladder rungs 1 and 2) and two normally closed contacts labeled 'G.JogBW' (ladder rungs 3 and 4). These contacts are connected to the 'JogForward' and 'JogBackward' inputs of a block named 'MDV_Jog_2'. A red 'X' is placed over the 'JogBackward' input connection, and a blue square highlights the 'JogBackward' input label. The block has various outputs including Done, Busy, Active, CommandAborted, Error, ErrorID, and Axis. The software interface includes a navigation tree on the left, a top menu bar, and an element selection panel on the right showing a library of motion control functions.

The screenshot displays the MELSOFT GX Works3 E interface. The main workspace shows a ladder logic diagram with four normally open contacts labeled G_b.JogFW (1, 2, 3, 4) connected to a function block named MDV_Jog_2. The function block has inputs for JogForward, JogBackward, Velocity, and Active, and outputs for Done, Busy, and Active. A red square highlights the area between contacts 3 and 4, and a callout box with the text "Clique na área onde o elemento FBD será inserido." (Click in the area where the FBD element will be inserted.) points to this location. The right-hand side of the screen shows the Element Selection panel with a search filter for "POU" and a list of motion control function blocks, including MCv_Jog.

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Element Selection (Find POU)

Display Target: All

User Library

Library

- MotionControl_RD78_1.03D Moti
- FB
- Management
- Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addri
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Spees
 - MC_Stop Forcs
 - MC_TorqueControl Torq.
 - MCv_BacklashComp Backl
 - MCv_DirectionFilter Direc
 - MCv_Jog JOG
 - MCv_MoveCircularIn Abso

MCv_Jog [Version] 03D

POU... Favori... History Mod... Library

R04 Host (R=5, C=10) CAP NUM

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a function block labeled "G_LeJogVelocity". A callout box with the text "Insira a label global 'G_LeJogVelocity'." points to a "G_Le" label within the block. The function block has several inputs and outputs: "JogBackward", "JogForward", "Deceleration", "Jerk", "Options", "Active", "Error", "ErrorID", "Busy", and "CommandAborted". The "Active" output is currently set to "LREAL JOG Velocity". The "Setting..." button is visible below the block. The left sidebar shows the project structure, including "ProgramBody" and "Global Label". The right sidebar shows the "Element Selection" panel with a search bar and a "Display Target" dropdown set to "All". The bottom status bar shows "R04", "Host", "(R=6, C=8)", and "CAP NUM".

As labels já registradas nas labels globais são exibidas como alternativas.

The screenshot displays the MELSOFT GX Works3 E interface. The main workspace shows a ladder logic diagram for a motion control function block (MDV_Jog_2). The diagram includes several global labels (G_b.JogFW, G_b.JogBW, G_b.JogV) and a function block (MDV_Jog_2) with inputs like JogForward, JogBackward, and JogVelocity. A callout box highlights the text: "As labels já registradas nas labels globais são exibidas como alternativas." (Labels already registered in global labels are shown as alternatives). The interface also shows a navigation pane on the left with a tree view of the project structure, and a status bar at the bottom indicating the current position (R04) and host connection.

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with three normally open contacts labeled G_b.JogFW (1, 2, and 3) connected to the 'Velocity' input of a function block named MDV_Jog_2. A callout box with a white background and black border points to the 'Velocity' input, containing the text: "Conecte o elemento FBD à entrada Velocity do FB." The software's navigation pane on the left shows a tree view with 'ProgramBody' selected. The bottom status bar indicates 'R04 | Host | (R=6, C=8) | CAP | NUM'.

Conecte o elemento FBD à entrada Velocity do FB.

Quando a label indefinida é inserida, a tela "Undefined LabelRegistration" é exibida. Defina o tipo dos dados e o destino registrado.

Undefined Label Registration

Not defined as global label or local label.
Please set new label information to be registered.

Label Name: leJogAcc

Label Setting Information

Registered Destination: Local Label(ProgPou)

Class: VAR

Data Type: Word [Signed]

Constant:

Comment:

Open the label editor and set the label details after registering label information.

Register a label at the last row.

OK Cancel

Navigation: Project, Module Configuration, Program, Initial, Scan, ProgramPou, Local Label, ProgramBody, Fixed Scan, Event, Standby, No Execution Type, Unregistered Program, FB/FUN, Label, Global Label, M+RD78_0000, Structured Data Types, Device, Parameter.

Element Selection: (Find POU), Display Target: All, User Library, Library, MotionControl_RD78_1.03D, FB, Management, Operation-Individual, MC_CamIn, MC_CombineAxes, MC_GearIn, MC_GroupStop, MC_Home, MC_MoveAbsolute, MC_MoveRelative, MC_MoveVelocity, MC_Stop, MC_TorqueControl, MCv_BacklashComp, MCv_DirectionFilter, MCv_Jog, MCv_MoveCircularIn.

MCv_Jog [Version] 03D

POU... Favorit... History Mod... Library

R04 Host (R=7, C=8) CAP NUM

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Element Selection (Find POU)

Display Target: All

User Library

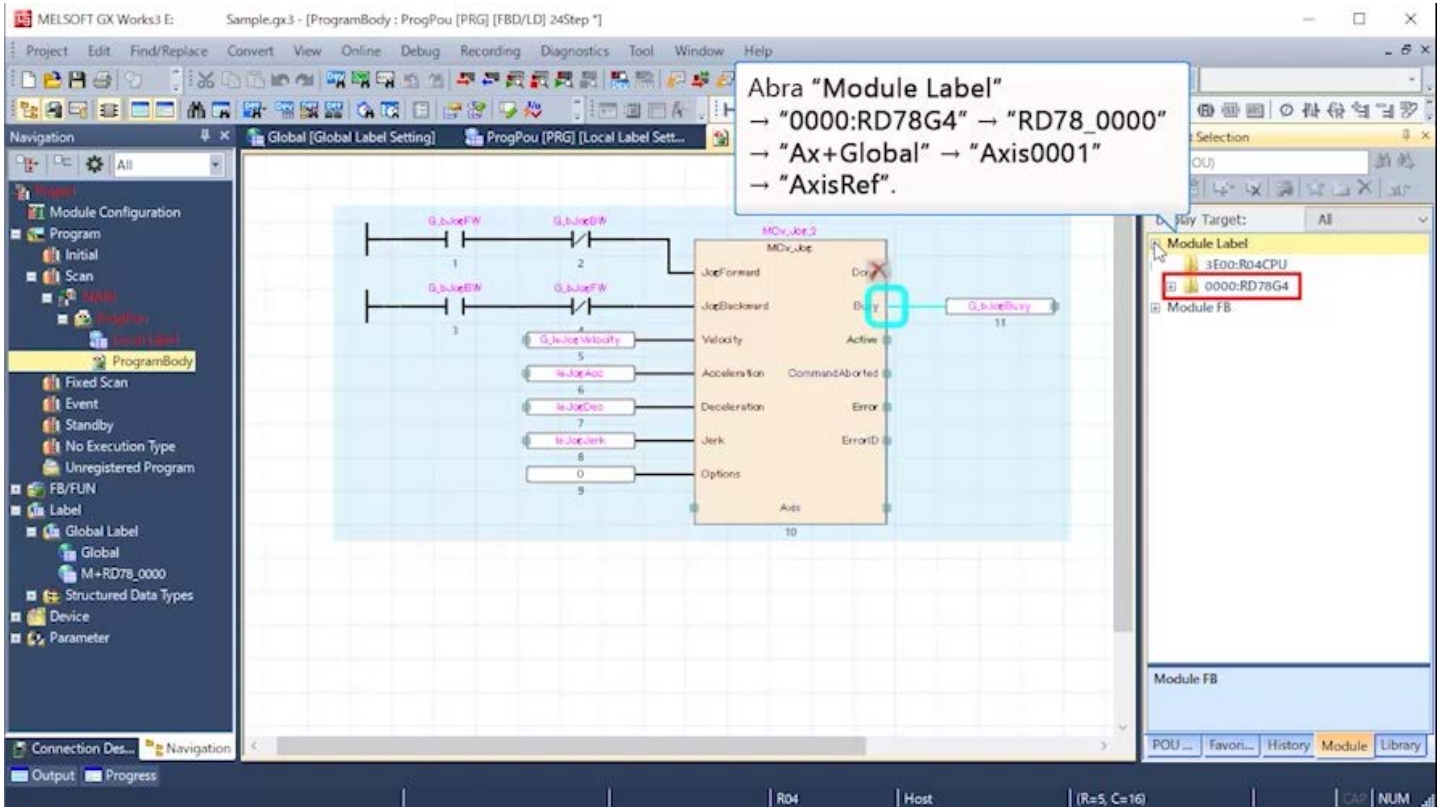
- Library
 - MotionControl_RD78_1.03D Moti
 - FB
 - Management
 - Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addit
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Spee
 - MC_Stop Forcc
 - MC_TorqueControl Torq
 - MCv_BacklashComp Backl
 - MCv_DirectionFilter Direc
 - MCv_Jog JOG

Connection Des... Navigation

Output Progress

R04 Host (R=5, C=16) CAP NUM

Clique em [Module].



The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram for a motion control function block (FB). The diagram includes a callout box with the following instructions:

- Abra "Module Label"
- "0000:RD78G4" → "RD78_0000"
- "Ax+Global" → "Axis0001"
- "AxisRef".

The diagram shows a function block named "MDv_x001" with various inputs and outputs. The inputs include "JocForward", "JocBackward", "Velocity", "Acceleration", "Deceleration", "Jerk", and "Options". The outputs include "Do", "Busy", "Active", "CommandAborted", "Error", and "ErrorID".

The right-hand side of the interface shows a tree view of the project structure. The "Module Label" section is expanded, showing the following items:

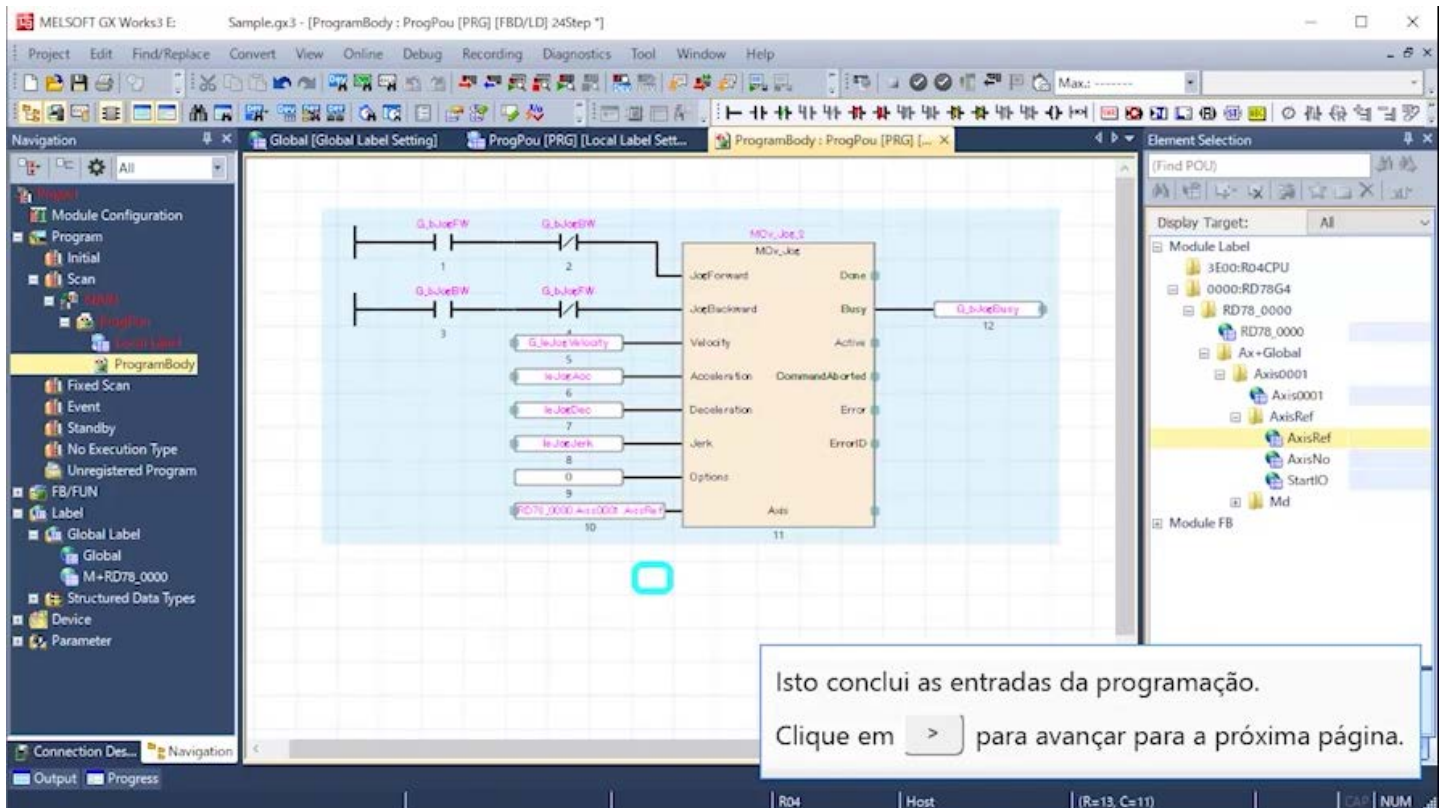
- 3E00-R04CPU
- 0000-RD78G4 (highlighted with a red box)
- Module FB

The bottom status bar shows the current position (R04), host status (Host), and other parameters (R=5, C=16).

Conecte os dados estruturados do tipo AxisRef registrados como label pública à entrada Axis. Arraste e solte "AxisRef" (informações sobre o eixo) no editor de programas.

AxisRef
軸情報

POU... Favorit... History Module Library



The screenshot displays the MELSOFT GX Works3 E interface. The main workspace shows a ladder logic diagram for a Motion Control Function Block (FB). The diagram includes the following elements:

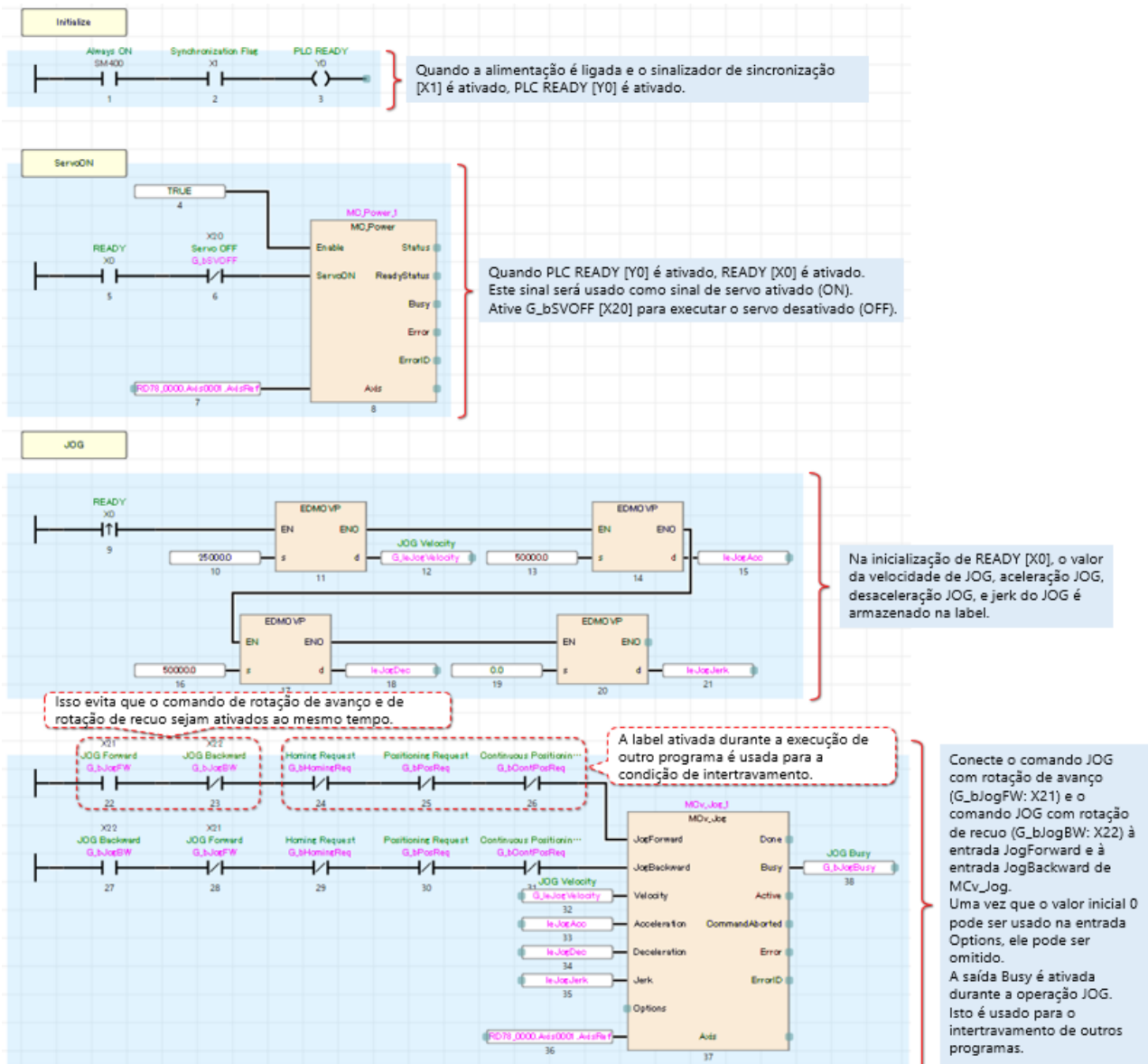
- Inputs:** G.JobFW (1), G.JobBW (2), G.JobBW (3), G.JobVelocity (4), W.JobAcc (5), W.JobDec (6), W.JobJerk (7), Options (8), and P078.0000 Axis0002 AxisRef (10).
- Function Block:** MDV_Job_2 (Motion Control Job 2).
- Outputs:** Done, Busy (G.JobBusy 12), Active, Command Aborted, and Error ID.

A callout box in the bottom right corner contains the following text:

Isto conclui as entradas da programação.
Clique em para avançar para a próxima página.

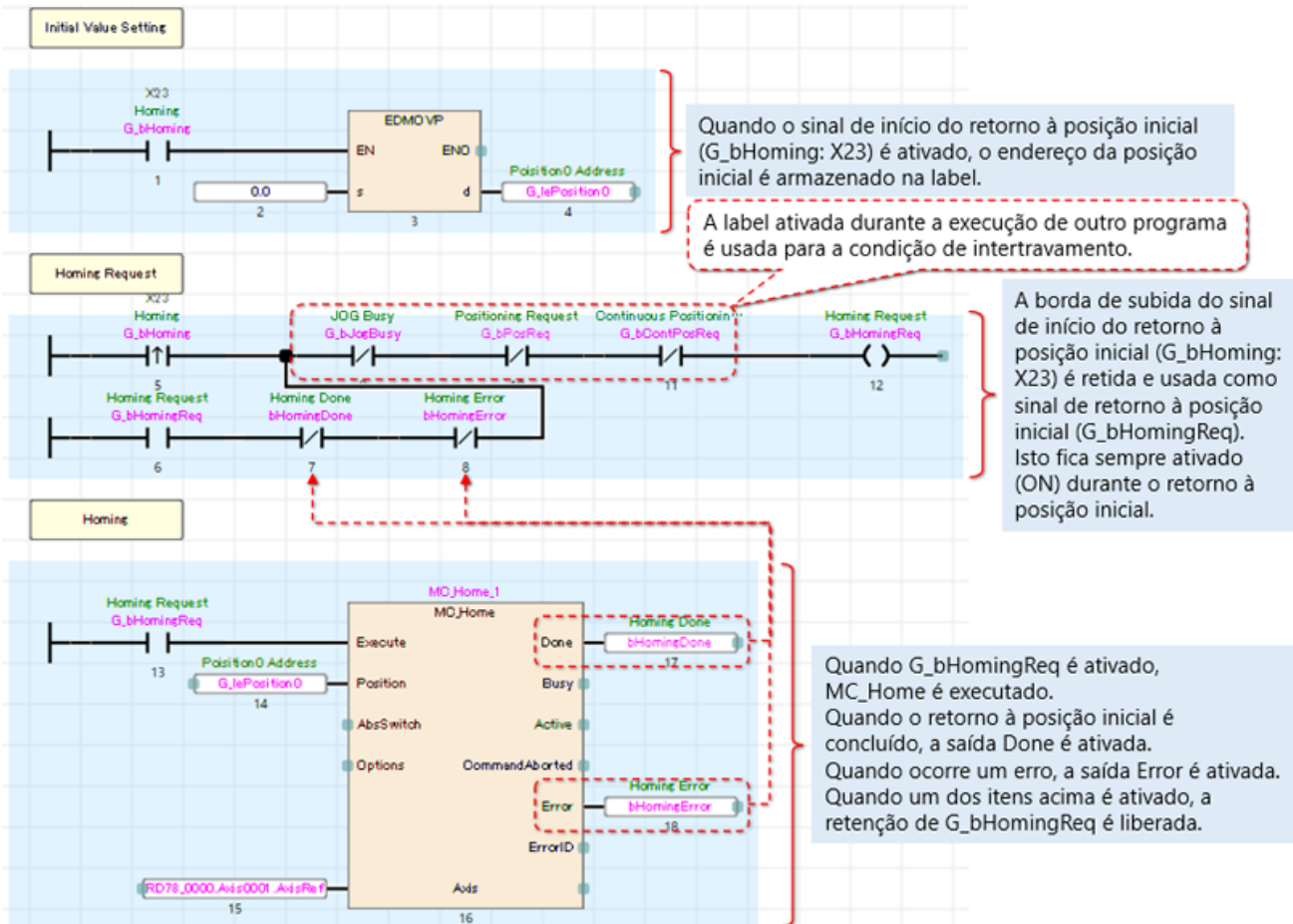
(1) ServoON_Jog

Esse programa executa o PLC já ativado (ON), o servo ativado (ON) e a operação JOG.



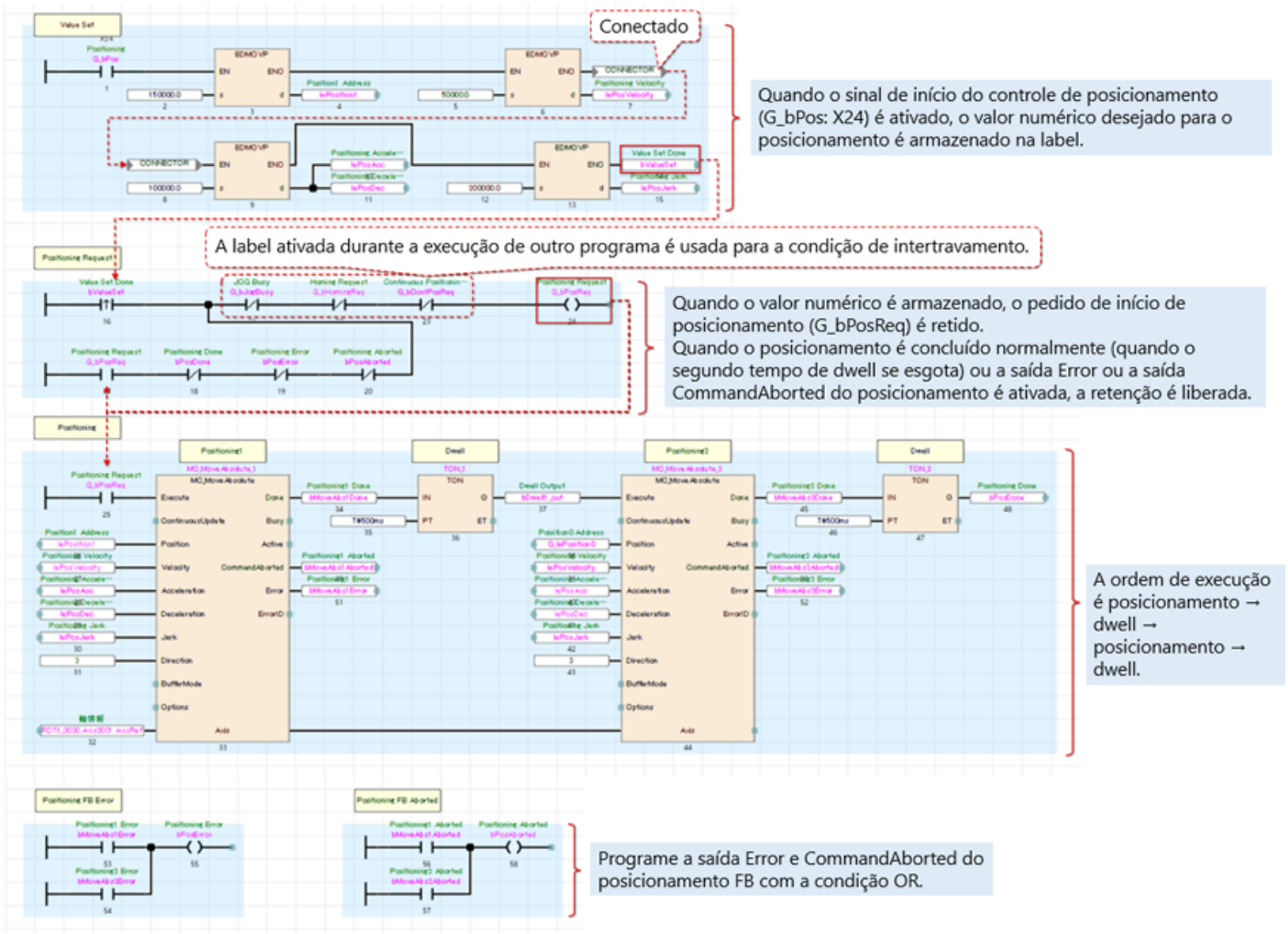
(2) Posição inicial

Esse programa executa o retorno à posição inicial.



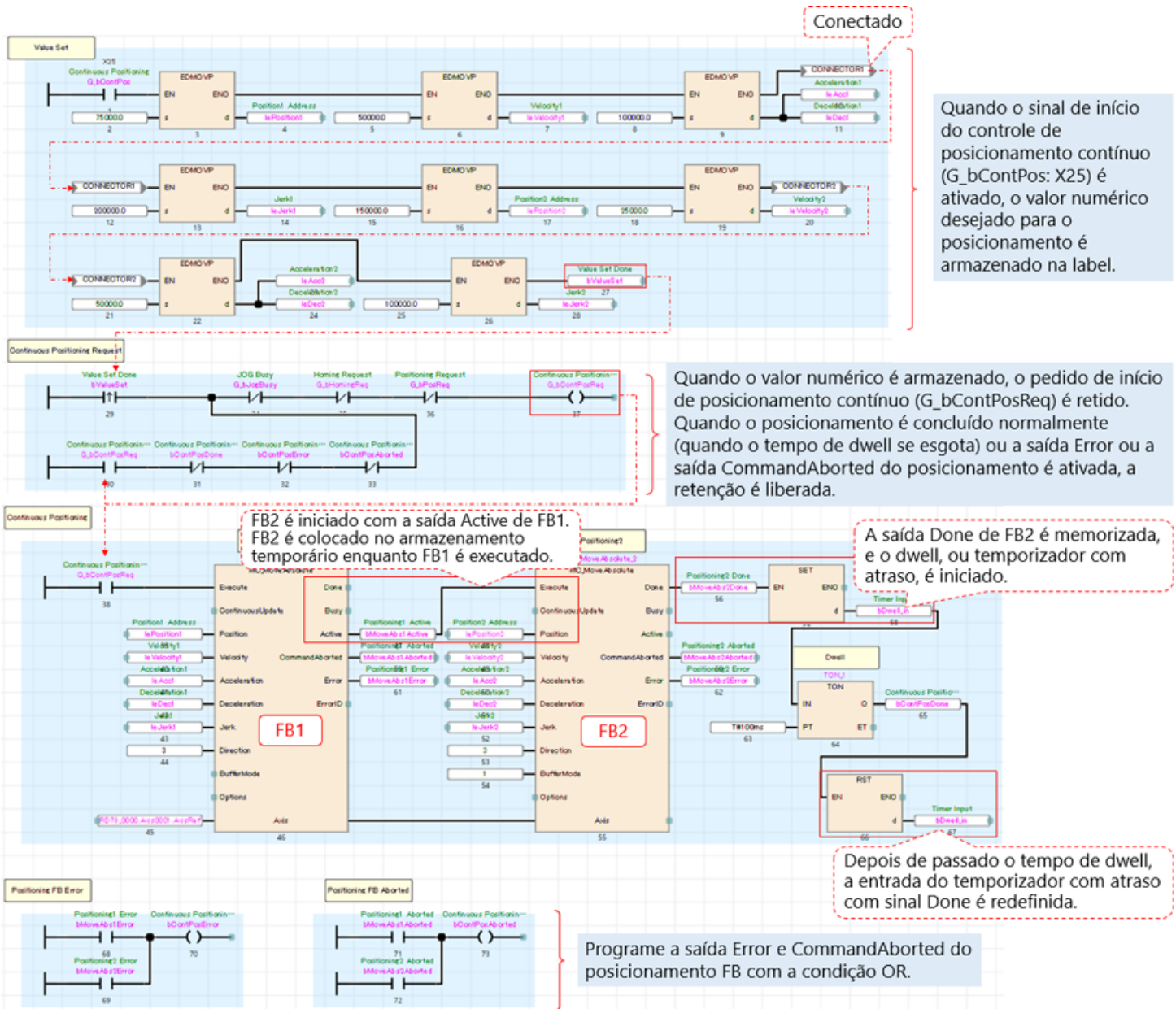
(3) Posicionamento

Esse programa executa a operação de posicionamento de eixo único.



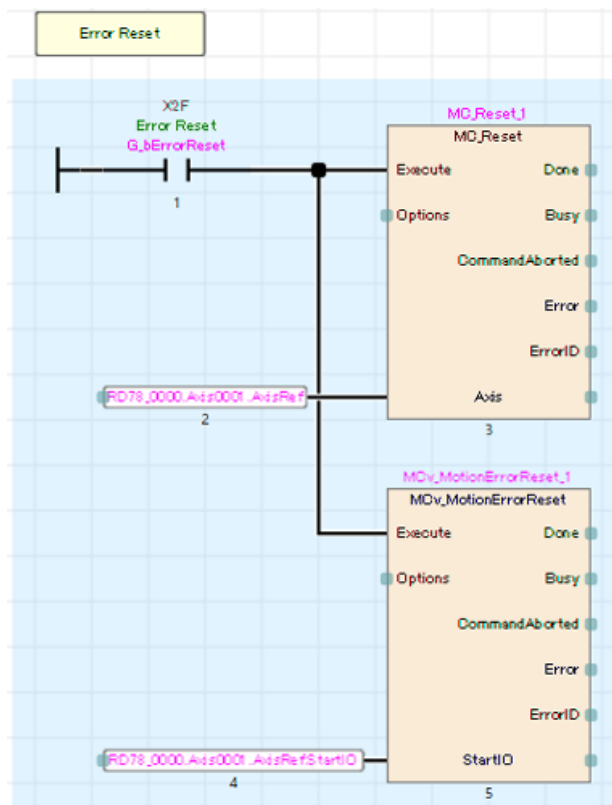
(4) ContinuousPositioning

Esse programa executa o posicionamento contínuo usando o modo de buffer.



(5) ErrorReset

Esse programa executa a redefinição do erro.



Quando o sinal de redefinição do erro (G_bErrorReset: X2F) é ativado, a redefinição do erro do eixo (MC_Reset) e a redefinição do erro do sistema (MCv_MotionErrorReset) são executadas.

(6) Monitor

Esse programa armazena SetPosition (Definir posição) e SetVelocity (Definir velocidade) da label global do monitor do eixo atribuída a D0 e D2 da CPU do PLC.

Uma vez que SetPosition e SetVelocity são do tipo número real de precisão dupla, são convertidos para o tipo palavra dupla com assinatura, para que possam ser facilmente manuseados pela CPU do PLC. (Nota)

Esses dispositivos de palavras não são usados neste tema.

Eles são usados para exibição em outros programas de sequência e no GOT, e para outros fins. Esse programa é descrito com ST.

```
1 G_dSetPosition := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);  
2 G_dSetVelocity := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);  
3
```



G_dSetPosition → D0
G_dSetVelocity → D2

(Nota) Quando o tipo número real de precisão dupla é convertido para o tipo palavra dupla com assinatura, se o valor a ser convertido estiver fora da faixa de -2147483648 a 2147483647, ocorrerá um erro de cálculo.

Grave o programa e os parâmetros na CPU do PLC e no módulo de movimento.

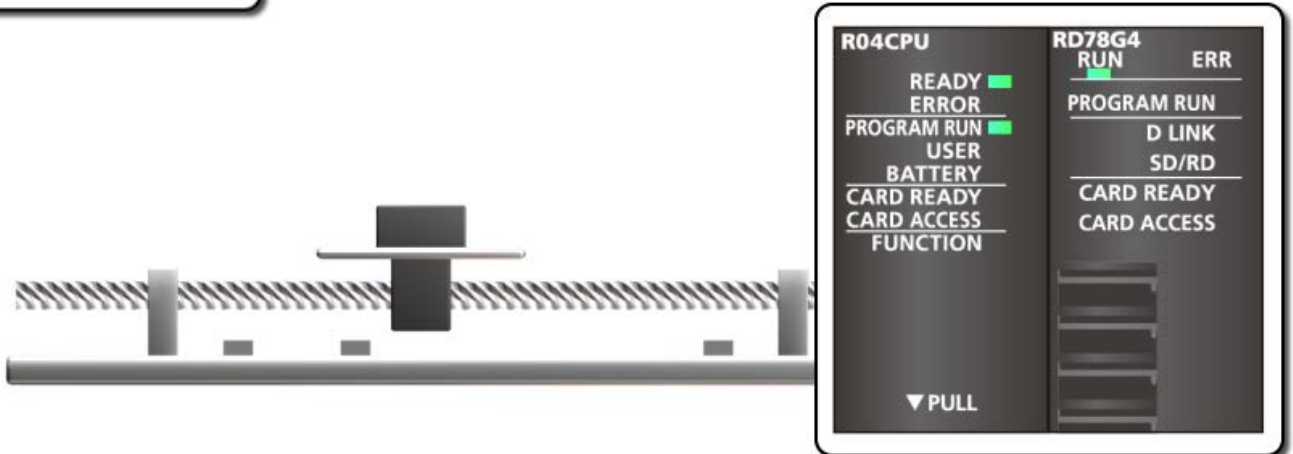
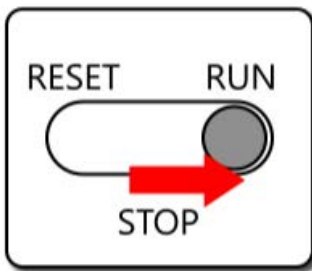
O programa é gravado apenas no módulo da CPU. As definições de parâmetros do eixo e label pública devem ser gravadas no lado do módulo de movimento.

- 1) Depois que todos os programas da CPU do PLC forem reconstruídos, selecione [Online] → [Write to PLC], na barra de ferramentas do GX Works3, para gravar todos os dados na CPU do PLC.
- 2) Quando os parâmetros forem gravados na CPU do PLC, a comunicação com o módulo de movimento é ativada. Selecione [Online] → [Write to Module], na barra de ferramentas de Motion Control Setting Function, para gravar todos os dados no módulo de movimento.
- 3) Reinicie a CPU do PLC para concluir a gravação.

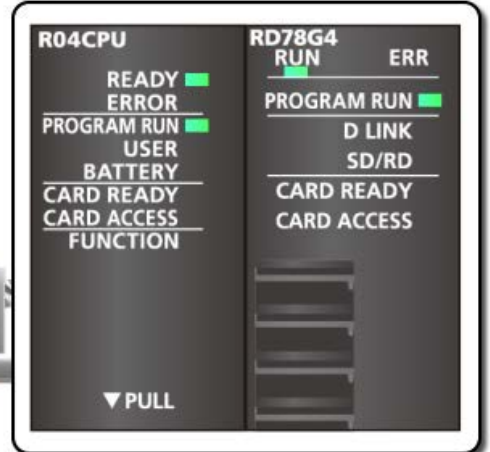
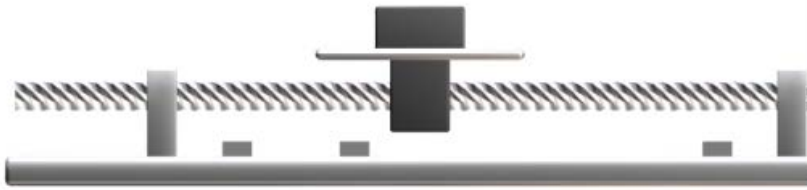
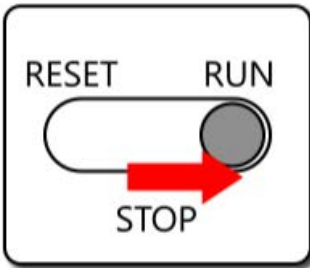
Clique no botão Executar, na área inferior esquerda da janela.



Verifique a operação do exemplo de programa.
Antes de iniciar a operação, certifique-se de que os programas e parâmetros sejam gravados na CPU do PLC e no módulo de movimento.



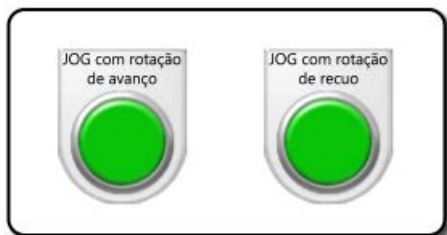
Defina o botão RUN/STOP/RESET da CPU do PLC como RUN.
A luz READY e a luz PROGRAM RUN do controlador programável se acendem.
A luz RUN do módulo de movimento se acende.



Espera até que a luz PROGRAM RUN do módulo de movimento se acenda.
 "r.01" é exibido no servo amplificador. (Os pontos se acendem.)
 O servomotor entra no estado de servo ativado (ON).



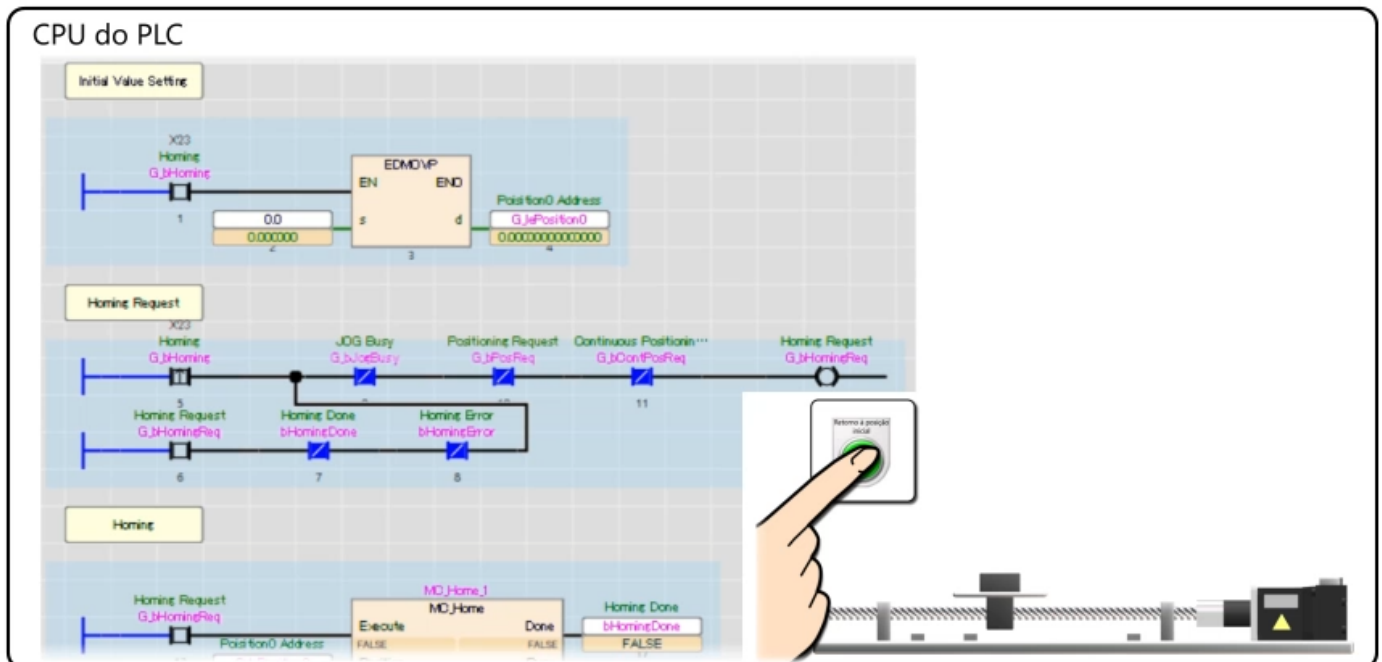
Acione X20 para executar o comando servo desativado (OFF).
"r.01" é exibido no servo amplificador. (Os pontos piscam.)
Desative X20 para executar o servo ativado (ON) novamente.



Ative o JOG com rotação de avanço (X21) para mover na direção de aumento do endereço, e desative-o para parar.
Ative o JOG com rotação de recuo do eixo (X22) para mover o eixo na direção de redução do endereço, e desative-o para parar.

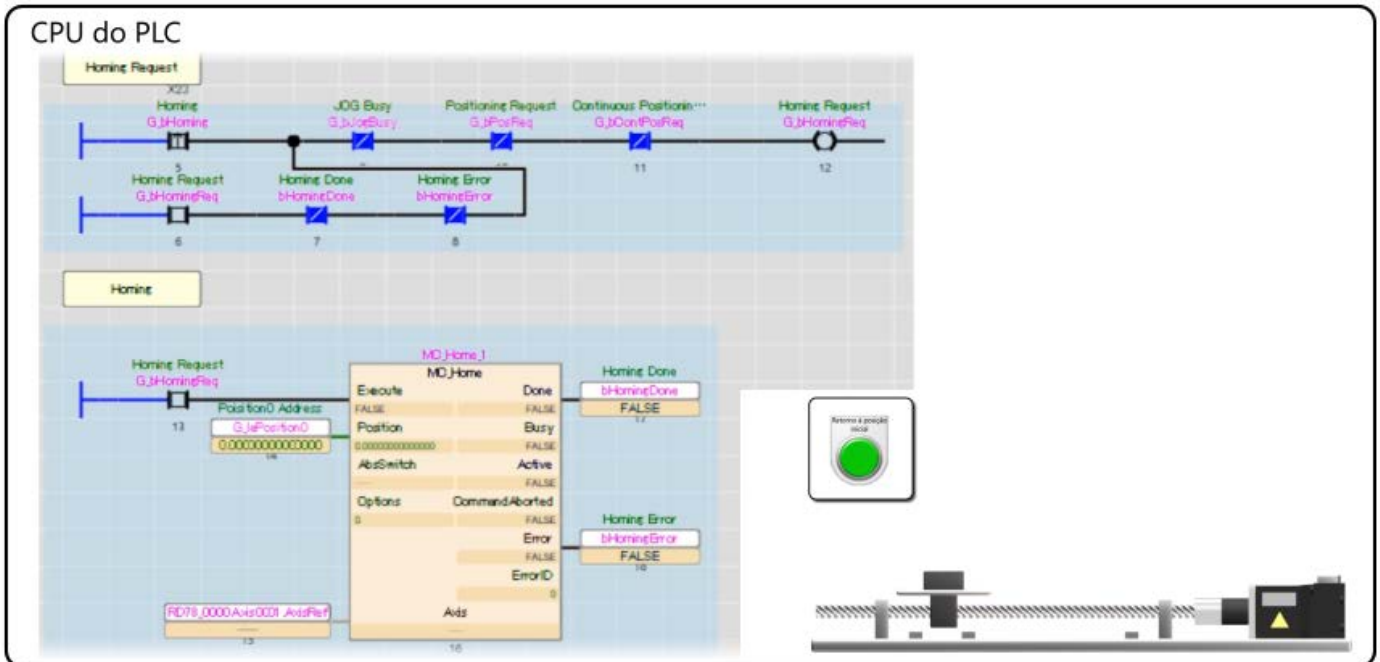


Ative o retorno à posição inicial (X23) para iniciar esse procedimento.
Execute o retorno à posição inicial com o método do dog de proximidade
(33 é subtraído de Pr.PT45)
O eixo para um pouco depois do dog,
e define esse ponto como a posição inicial.



Verifique o monitor do programa.

Quando X23 é ativado, o endereço da posição inicial é armazenado na label. "G_bHomingReq", que é o comando de execução de MC_Home_1, é ativado e retido.

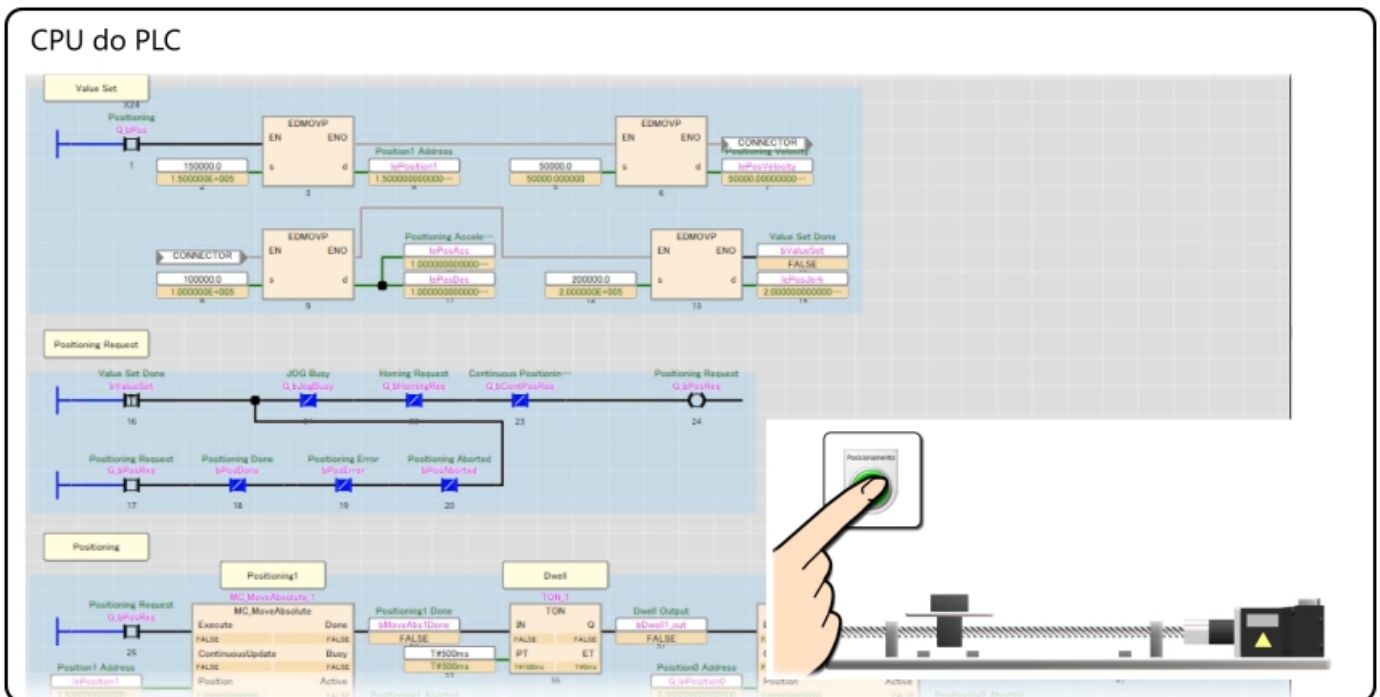


A operação de retorno à posição inicial é iniciada. Quando o retorno à posição inicial é concluído, a saída Done e "bHomingDone" são ativados, e a retenção de "G_bHomingReq" é cancelada.

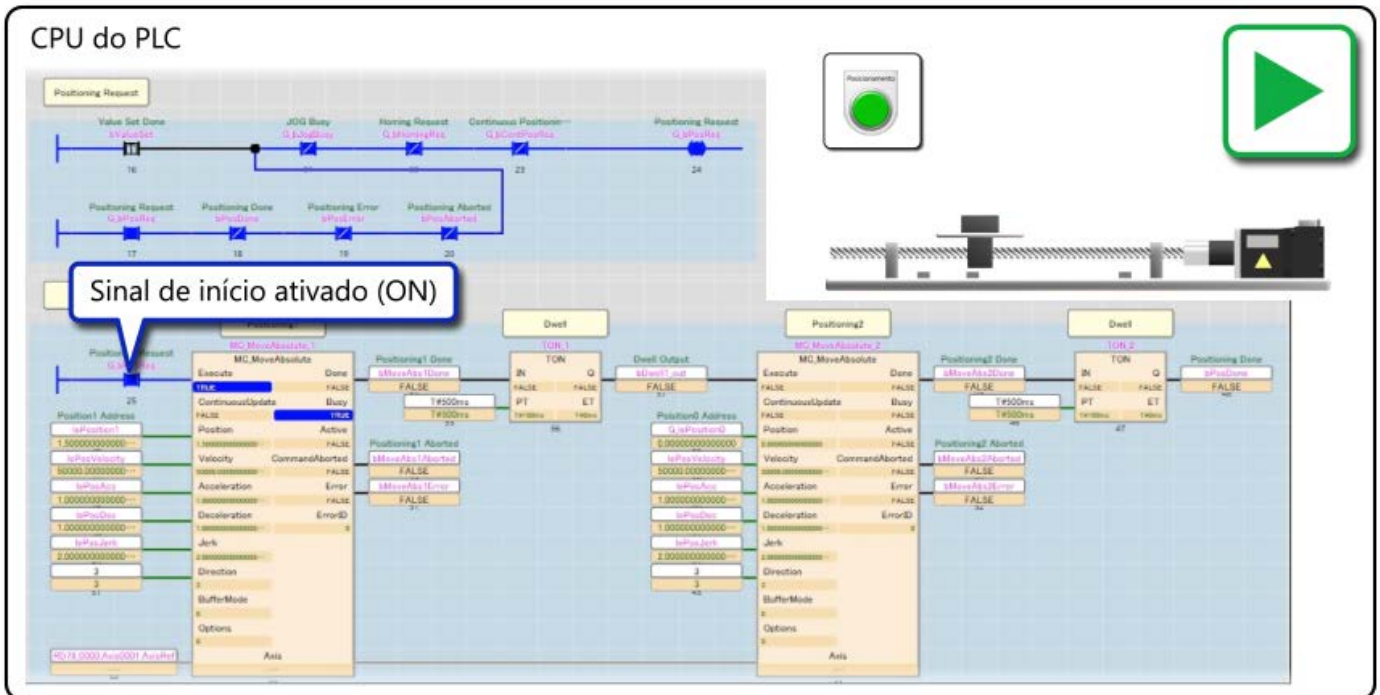


A ativação do início de posicionamento (X24) inicia o movimento de ida e volta.

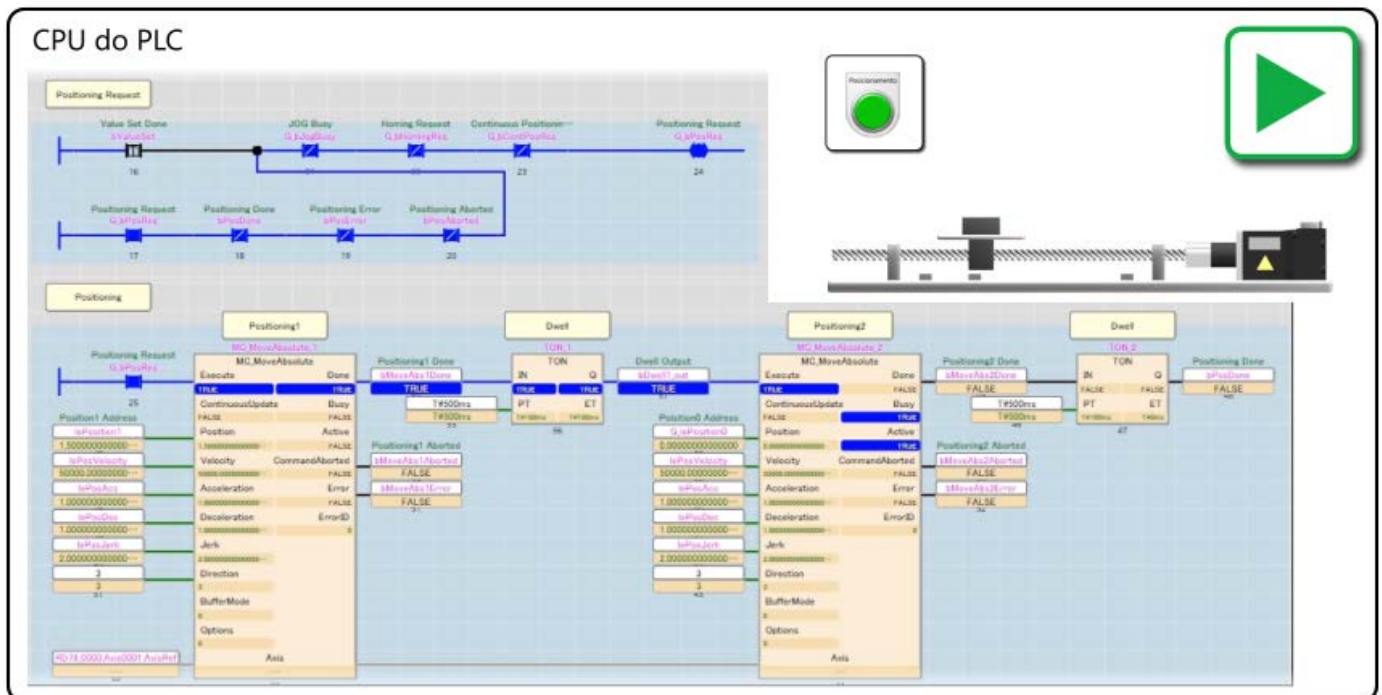
O eixo avança 150 mm e para durante 0,5 segundo, recua 150 mm e para durante 0,5 segundo.



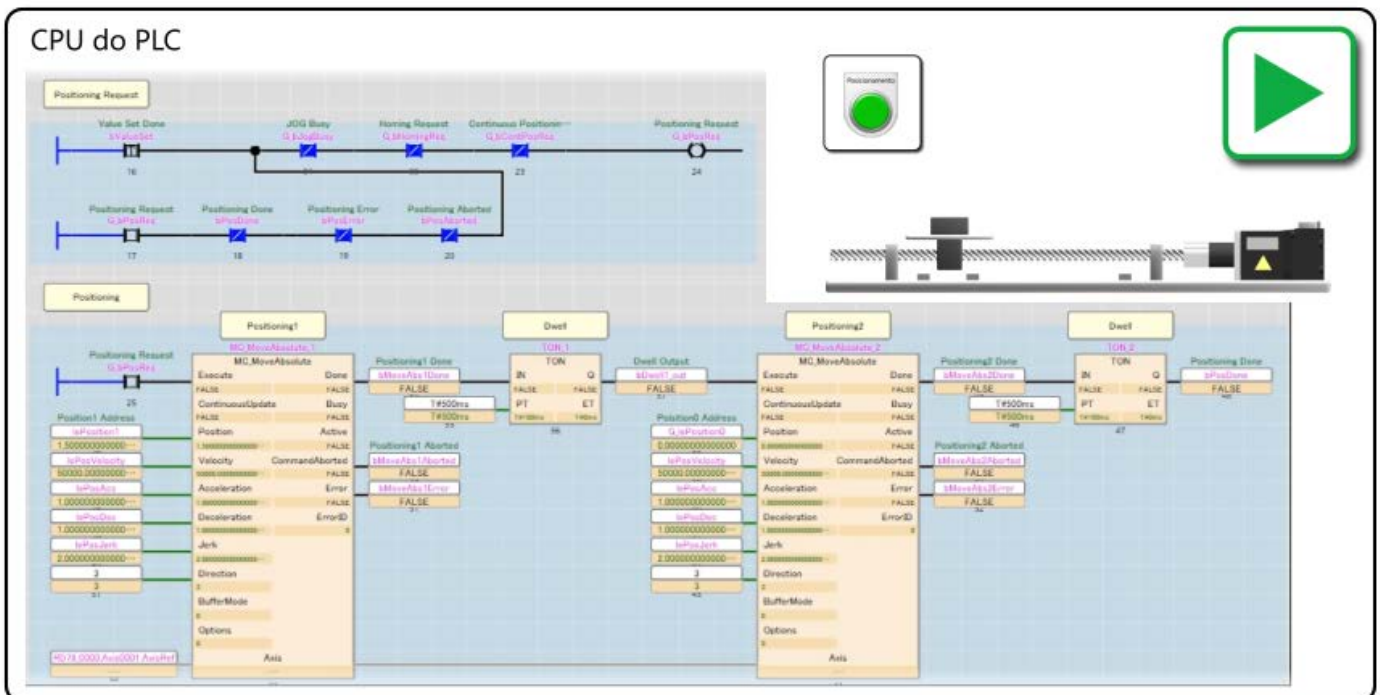
Verifique o monitor do programa.
 Quando X24 é ativado, os dados para posicionamento são armazenados em cada label, e "bValueSet" é ativado.
 "G_bPosReq", que é o comando de execução de MC_MoveAbsolute_1, é ativado e retido na borda de subida de "bValueSet".



Quando "G_bPosReq" é ativado, MC_MoveAbsolute_1 é iniciado, e o servomotor é acionado.



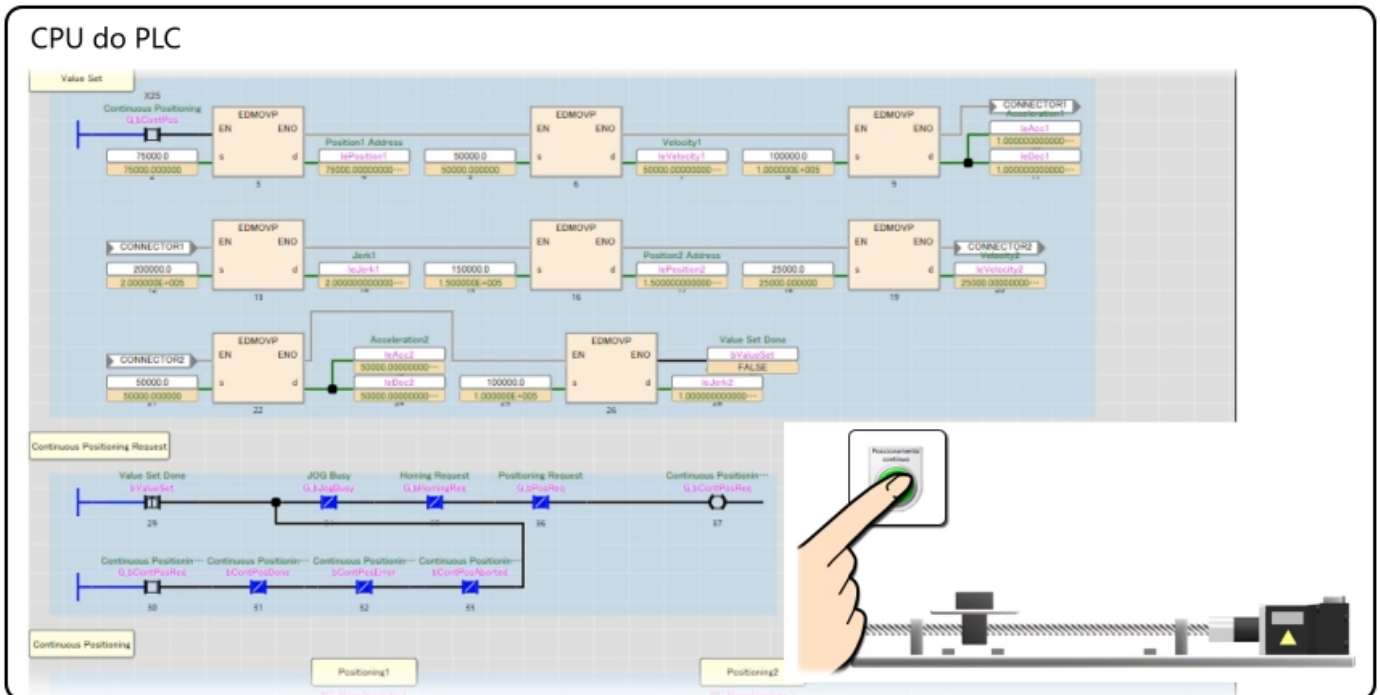
Quando o posicionamento feito por MC_MoveAbsolute_1 é concluído, TON_1, que é o dwell, é executado. Quando se passam 500 ms, MC_MoveAbsolute_2 é executado e o servomotor é acionado.



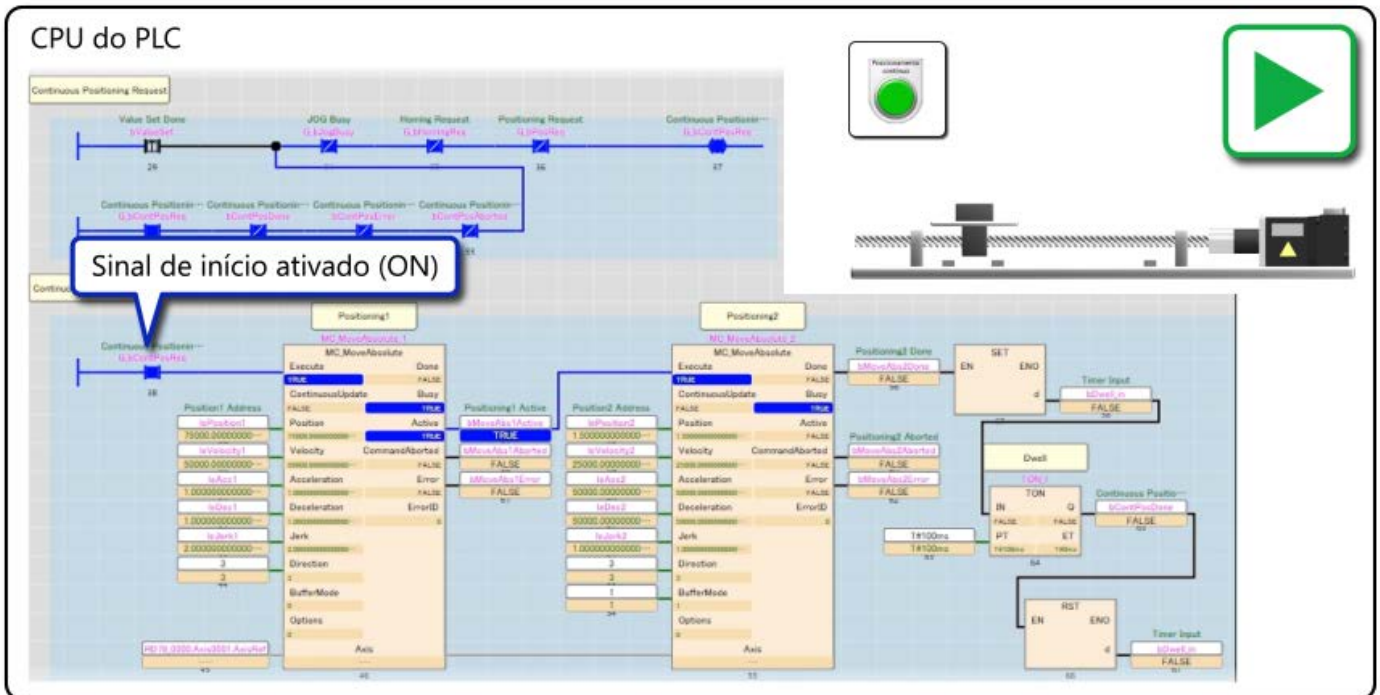
Quando o posicionamento feito por MC_MoveAbsolute_2 é concluído, TON_2, que é o dwell, é executado. Quando se passam 500 ms, a retenção de "G_bPosReq" é liberada, e redefinida com o estado inicial.



Ative o início de posicionamento contínuo (X25) para iniciar a operação do modo de buffer (mc_Buffered).

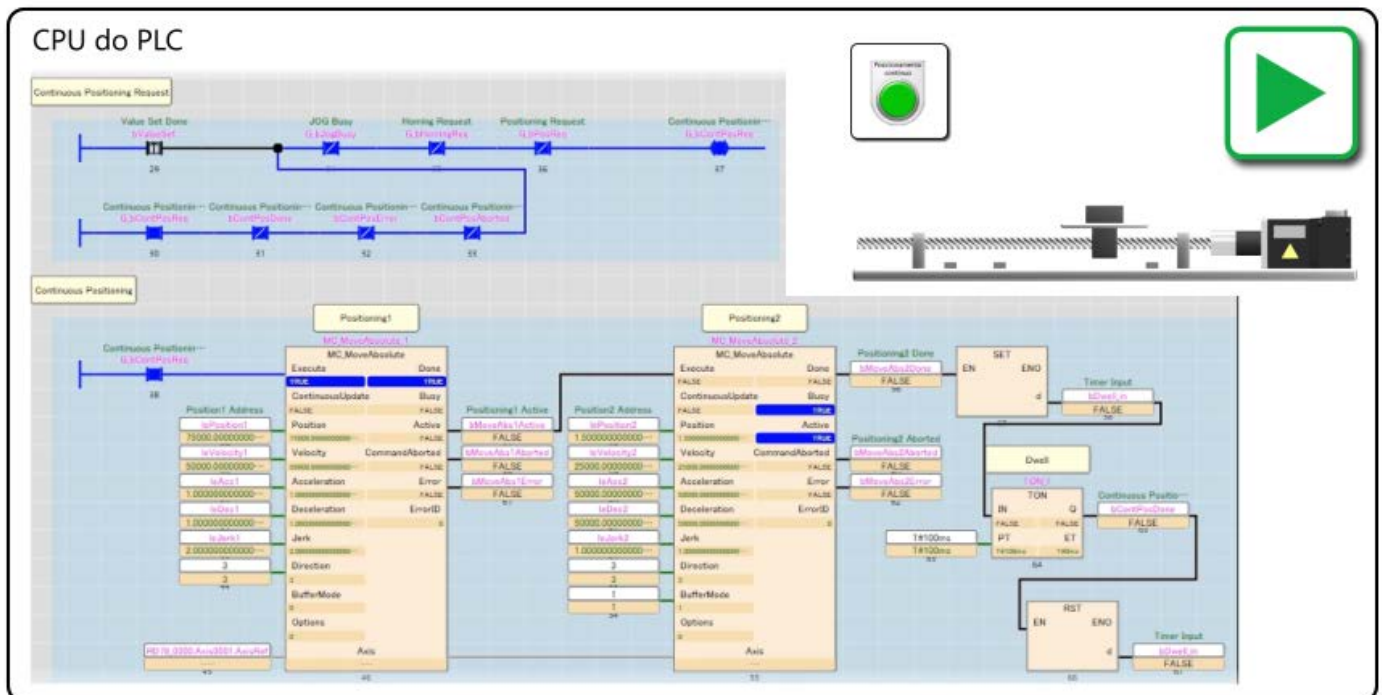


Verifique o monitor do programa.
 Quando X25 é ativado, os dados para posicionamento são armazenados em cada label, e "bValueSet" é ativado.
 "G_bContPosReq", que é o comando de execução de MC_MoAbsolute_1, é ativado e retido na borda de subida de "bValueSet".

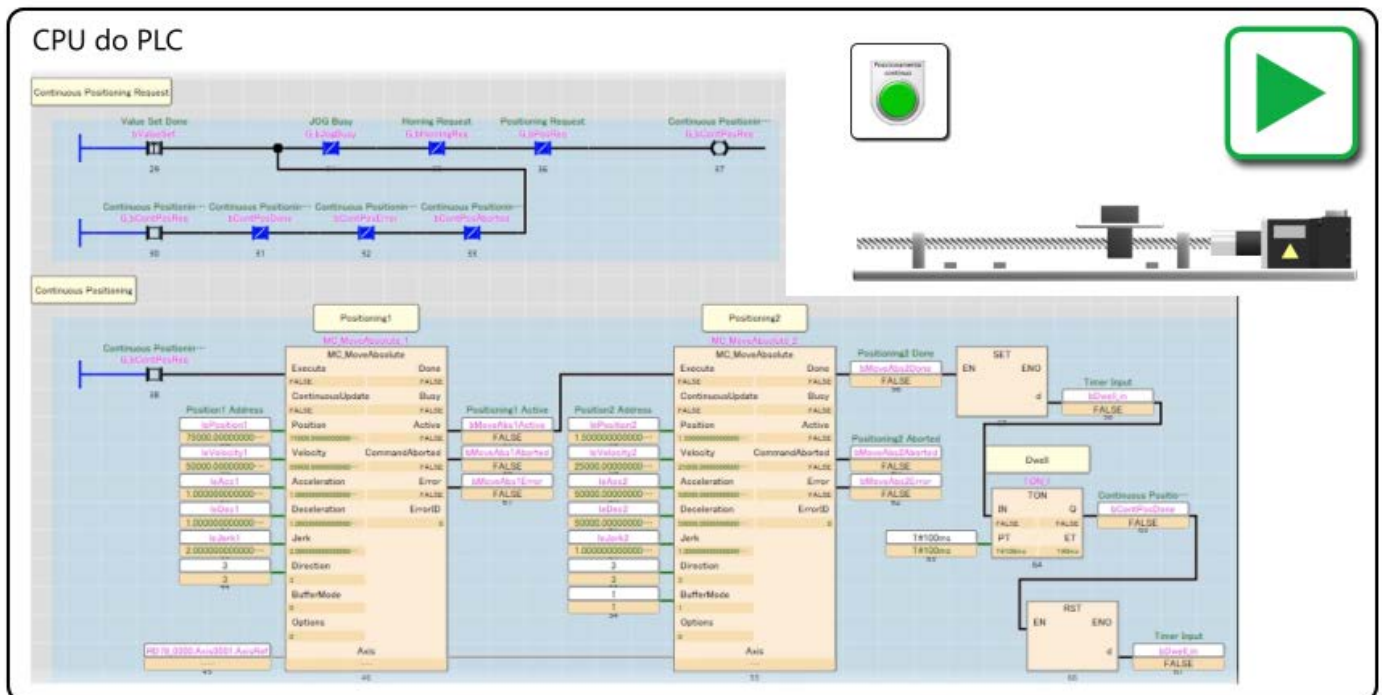


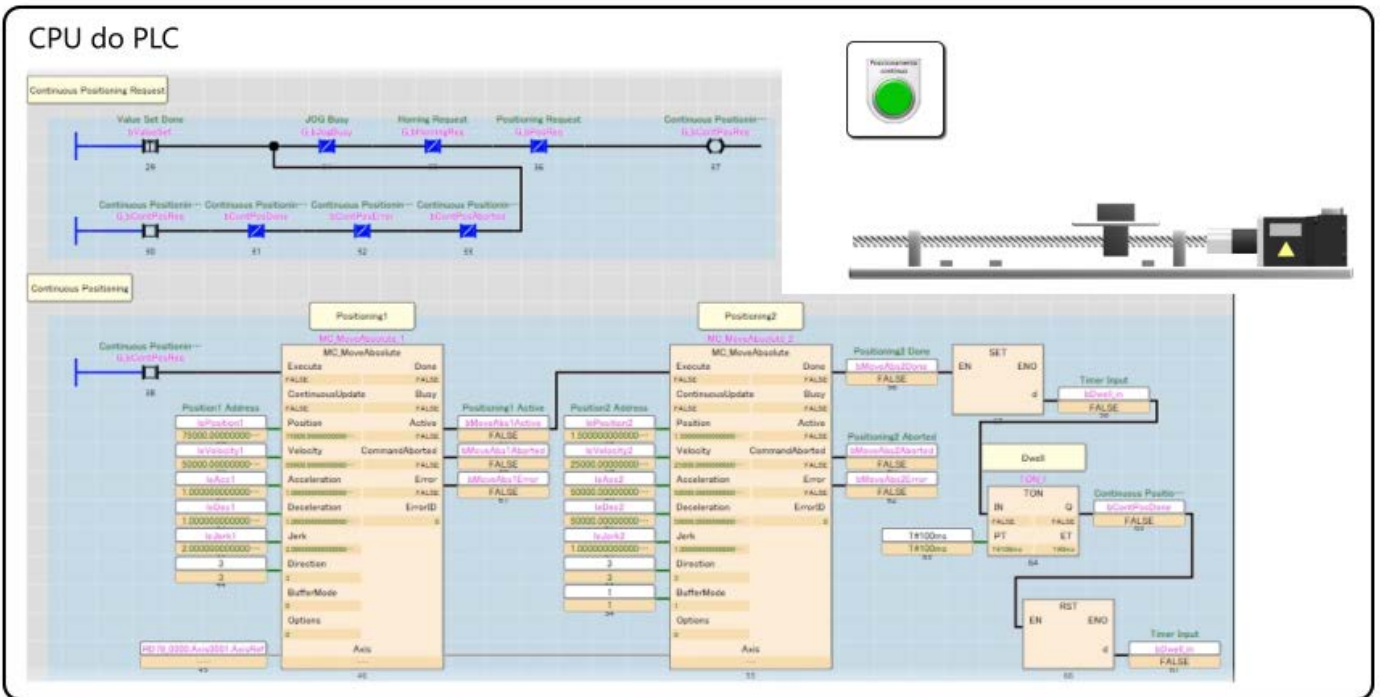
Quando "G_bContPosReq" é ativado, MC_MoveAbsolute_1 é iniciado e o servomotor é acionado.

Nesse momento, uma vez que a saída Active é o comando de execução de MC_MoveAbsolute_2, MC_MoveAbsolute_2 é colocado em armazenamento temporário.



Quando a operação de MC_MoveAbosolute_1 é concluída, o C_MoveAbsolute_2 em armazenamento temporário é executado. Quando a operação de MC_MoveAbosolute_2 é concluída, TON_1, que é o dwell, é executado.





Isto conclui a verificação da operação.
Vá para a próxima página.

Neste capítulo, você aprendeu o seguinte:

- Registrando a biblioteca de FBs do módulo de movimento
- Criando projetos
- Como usar o Motion control FB
- Descrição do exemplo de programa
- Verificação da operação do exemplo de programa

Ponto

Registrando a biblioteca de FBs do módulo de movimento	<ul style="list-style-type: none"> • A biblioteca de FBs deve ser registrada no GX Works3, para utilização do Motion control FB na CPU do PLC.
Criando projetos	<ul style="list-style-type: none"> • Configure os parâmetros dos eixos e outras definições, como na programação para o módulo de movimento.
Como usar o Motion control FB	<ul style="list-style-type: none"> • O Motion control FB pode ser inserido no editor de programas, arrastando-o e soltando-o na guia Library da janela Element Selection do GX Works3. • Conecte o contato e a label à entrada/saída do FB.
Descrição do exemplo de programa	<ul style="list-style-type: none"> • Você criou um programa semelhante aos exemplos de programas dos capítulos 2 e 3, usando apenas a CPU do PLC.
Verificação da operação do exemplo de programa	<ul style="list-style-type: none"> • Você verificou a operação do exemplo de programa apresentado no vídeo.

Capítulo 5 Registro

Este capítulo descreve como registrar os dados do módulo de movimento e exibi-los no gráfico.

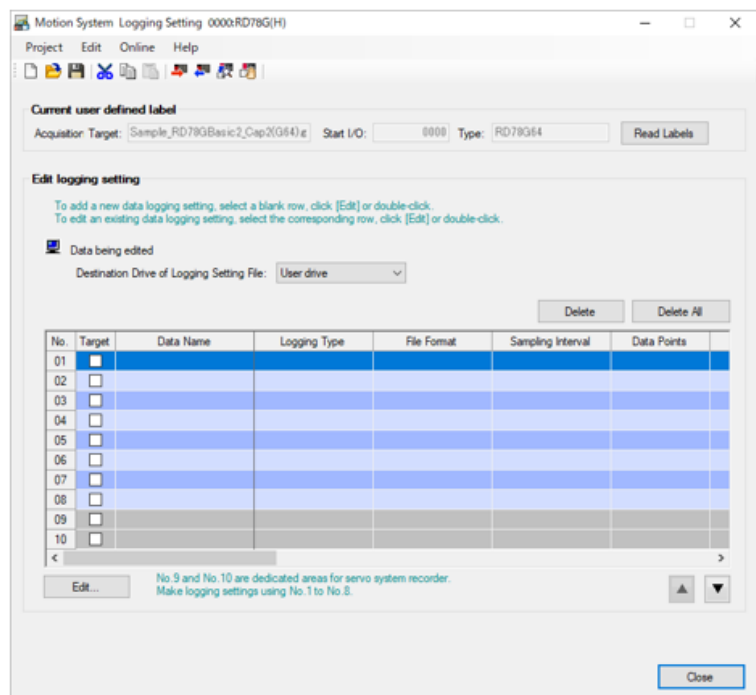
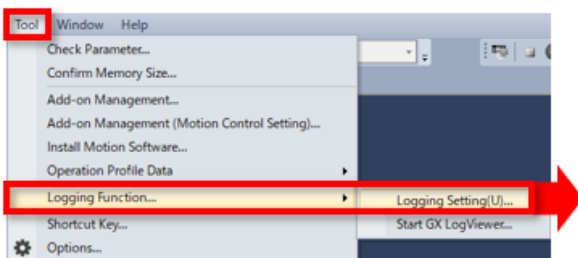
Neste curso, o programa de início de posicionamento do exemplo de programa nos capítulos 2 e 3 será registrado para ilustrar o procedimento.

(Nota) O programa do capítulo 4 não pode ser registrado com o procedimento descrito neste capítulo.

Deve-se usar "CPU Module Logging Configuration Tool".

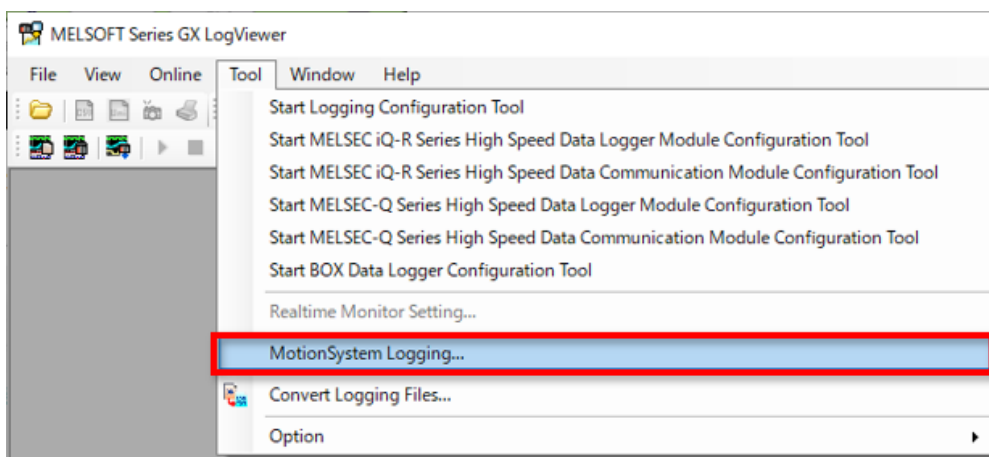
5.1 Iniciando a ferramenta de configuração do registro

Selecione [Tool] → [Logging Function] → [Logging Setting], na barra de ferramentas da tela Motion Control Setting Function. A ferramenta de configuração do registro do sistema de movimento é iniciada.



[Ponto]

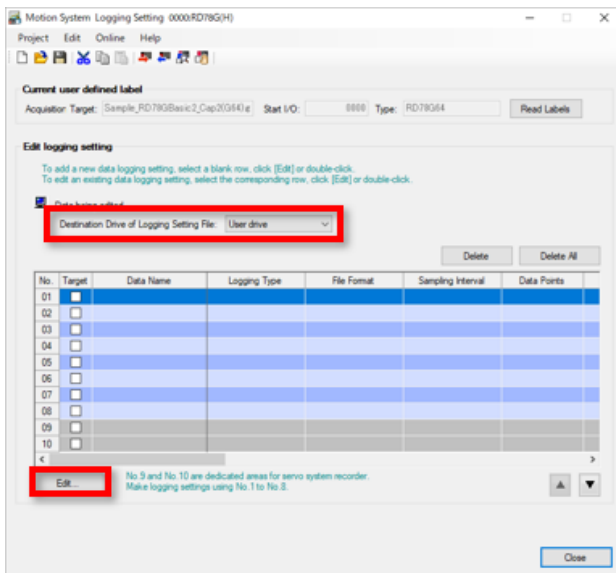
A ferramenta de configuração do registro do sistema de movimento pode ser iniciada em [Tool] → [MotionSystem Logging], no GX LogViewer.



(1) No campo de edição da configuração do registro da ferramenta de configuração do registro do sistema de movimento, defina onde os dados registrados serão salvos.

A seguir, clique no botão [Edit].

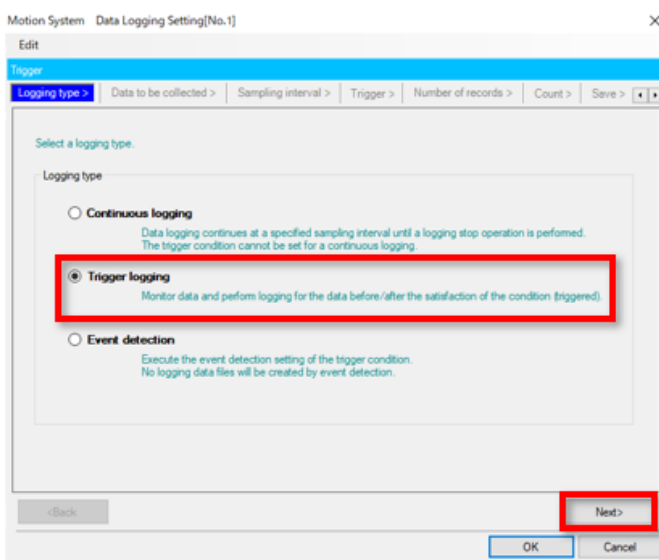
A tela Data Logging Setting é apresentada.



(2) Selecione [Logging type], entre: registro contínuo, registro por acionamento e detecção de evento.

Este curso descreve detalhes do registro por acionamento.

Selecione o registro por acionamento, e clique no botão [Next].



(3) A label dos dados a serem registrados é registrada em [Data to be collected].

1) Label global

Insira o nome da label global no campo de nome dos dados.
No campo de tipo de dados, selecione o tipo label.

2) Label local

Insira o nome dos dados no formato "nome do programa/nome da label local".
No campo de tipo de dados, selecione o tipo label.

3) Tipo de dados estruturados

Clique no botão [Label Input Assistant] e selecione o membro do tipo de dados estruturados na lista.

Selecione a opção na lista e clique no botão [Register] para refletir o processo nos dados que serão coletados.

Neste curso, os seguintes dados são registrados como exemplo.

Nome dos dados
G_bPosReq
Positioning/bMoveAbs1Done
Positioning/bDwell1_out
Positioning/bMoveAbs2Done
G_bPosDone
Axis0001.Md.SetPosition
Axis0001.Md.SetVelocity

Clique no botão [Next] quando o registro estiver concluído.

The screenshot shows the 'Motion System Data Logging Setting' dialog box. The 'Trigger' tab is active, and the 'Data to be collected' option is selected. The 'Label Input Assistant' dialog is open, showing a list of data items. The 'Next' button is highlighted in red. A green arrow points from the 'Next' button to the 'Label Input Assistant' dialog.

No.	Display	Type	Data Name	Data Type	Size (Word)	Axis Name
0001	<input checked="" type="checkbox"/>	VAR	G_bPosReq	Bit	1	
0002	<input checked="" type="checkbox"/>	VAR	Positioning/bMoveAbs1Done	Bit	1	
0003	<input checked="" type="checkbox"/>	VAR	Positioning/bDwell1_out	Bit	1	
0004	<input checked="" type="checkbox"/>	VAR	Positioning/bMoveAbs2Done	Bit	1	
0005	<input checked="" type="checkbox"/>	VAR	G_bPosDone	Bit	1	
0006	<input checked="" type="checkbox"/>	VAR	Axis0001.Md.SetPosition	Float (double precision)	4	Axis0001
0007	<input checked="" type="checkbox"/>	VAR	Axis0001.Md.SetVelocity	Float (double precision)	4	Axis0001
0008	<input type="checkbox"/>	VAR				
0009	<input type="checkbox"/>	VAR				
0010	<input type="checkbox"/>	VAR				
0011	<input type="checkbox"/>	VAR				

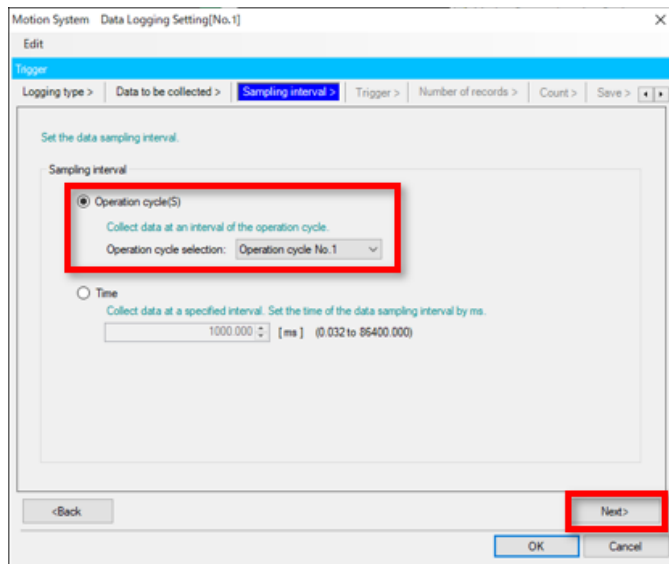
The 'Label Input Assistant' dialog shows the following settings:

- Type: Real Drive Axis
- Axis Name: Axis0001
- Structured Data Type: Monitor Data

Name	Data Name (Label)
Md Object Data_VelActualValue	*Md Io_VelActualValue
Md Jerk Limit Value	*Md JerkLimit
Md Control Cycle	*Md OperationCycle
Md Operation Setting at Overrun	*Md OverrunOperation
Md Current Position Restoration Status	*Md PosRestoration_Status
Md Execution Profile ID No.	*Md ProfileID
Md Set Acceleration	*Md SetAcceleration
Md Set Position	*Md SetPosition
Md Set Velocity	*Md SetVelocity
Md Slave Emulating	*Md SlaveEmulate_Enable
Md Start Permission at Homing Uncompleted	*Md StartableAtUnhomed
Md Deceleration at Stop	*Md StopMode_Deceleration
Md Stop Signal Signal Detection Method	*Md StopSignal_Detection

- (4) Defina o intervalo de amostragem em [Sampling Interval].
Neste curso, utilize o ciclo de operação nº 1 para a amostragem.

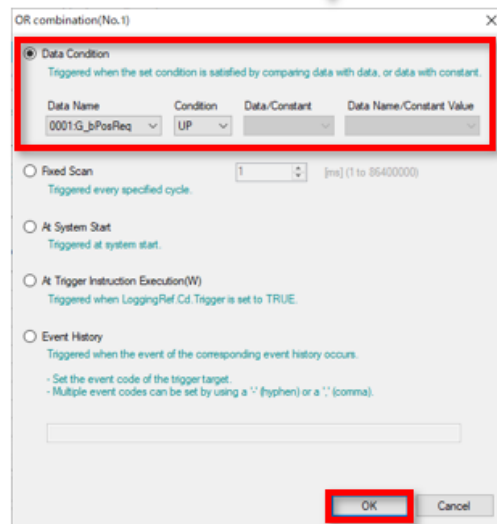
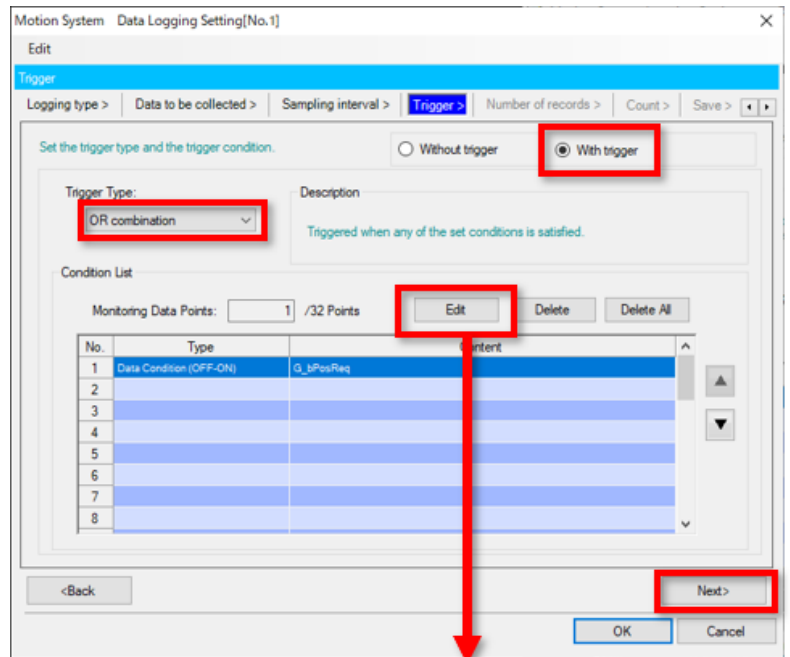
Depois de selecionar o intervalo de amostragem, clique no botão [Next].



(5) A condição para início do registro é definida em [Trigger].

Neste curso, a inicialização é o bit que o sinal de início de posicionamento utiliza para o acionamento.

- 1) Selecione [With Trigger].
- 2) Selecione "OR combination" para o tipo de acionamento.
- 3) Selecione o nº 1 na lista de condições, e clique no botão de edição. A subjanela é apresentada.
- 4) Selecione "Data Condition", e selecione "0001:G_bPosReq" como nome dos dados. Selecione "UP" como condição. Quando a seleção estiver concluída, clique no botão [OK].
- 5) Depois de retornar à tela original, clique no botão [Next].



- (6) O número de pontos de amostragem é definido em [Number of records]. Neste curso, o nº de registros (antes do acionamento) é definido como "500", e o nº de registros (depois do acionamento), como "19500". Quando a configuração estiver concluída, clique no botão [Next].

The screenshot shows the 'Data Logging Setting' dialog box with the 'Number of records' tab selected. The dialog has a breadcrumb trail: Logging type > Data to be collected > Sampling interval > Trigger > Number of records > Count > Save. The main area contains the following fields:

- No. of records (before trigger): 500 (Record) (0 to 299999)
- No. of records (after trigger): 19500 (Record) (1 to 300000)
- Total No. of records: 20000 (Record) (1 to 300000)

At the bottom, there are buttons for '<Back', 'Next>', 'OK', and 'Cancel'. The 'Next>' button is highlighted with a red box.

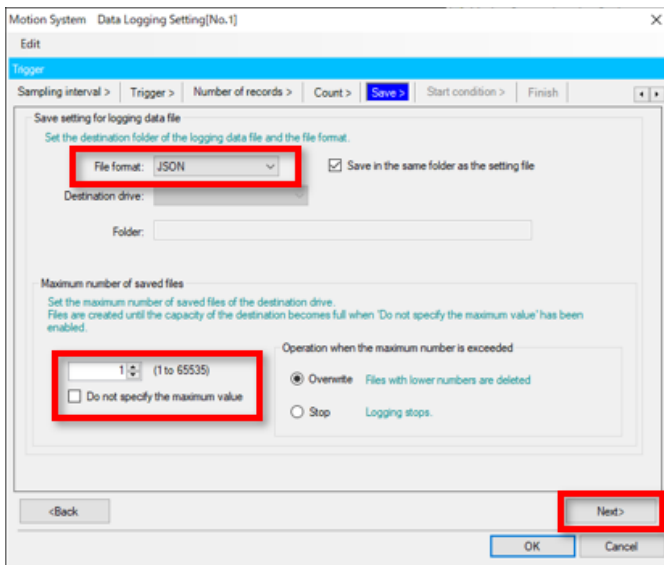
- (7) A contagem do registro é definida em [Count]. Neste curso, a contagem é definida como 1. Quando a configuração estiver concluída, clique no botão [Next].

The screenshot shows the 'Data Logging Setting' dialog box with the 'Count' tab selected. The breadcrumb trail is: Logging type > Data to be collected > Sampling interval > Trigger > Number of records > Count > Save. The main area contains the following options:

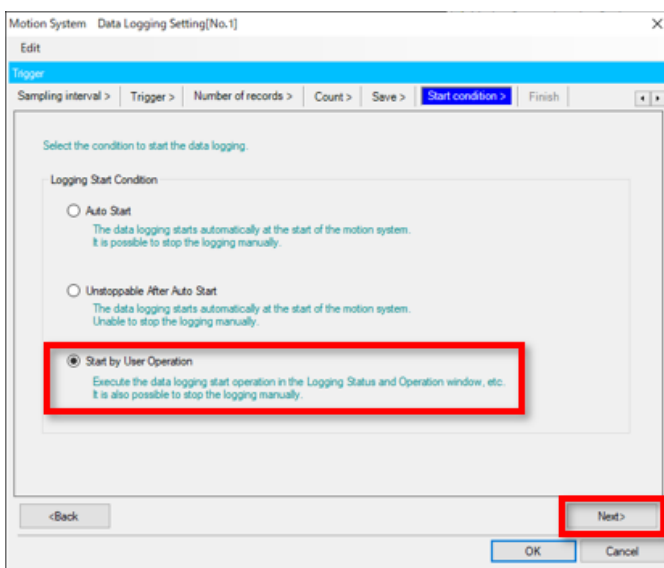
- Specified Count: 1 (1 to 32767)
Execute trigger logging repeatedly for the specified count.
The operation will be 'Overwrite' when the maximum number of saved files is exceeded.
- Specified Number of Saved Files
Execute the trigger logging repeatedly according to the maximum number setting of saved files.

At the bottom, there are buttons for '<Back', 'Next>', 'OK', and 'Cancel'. The 'Next>' button is highlighted with a red box.

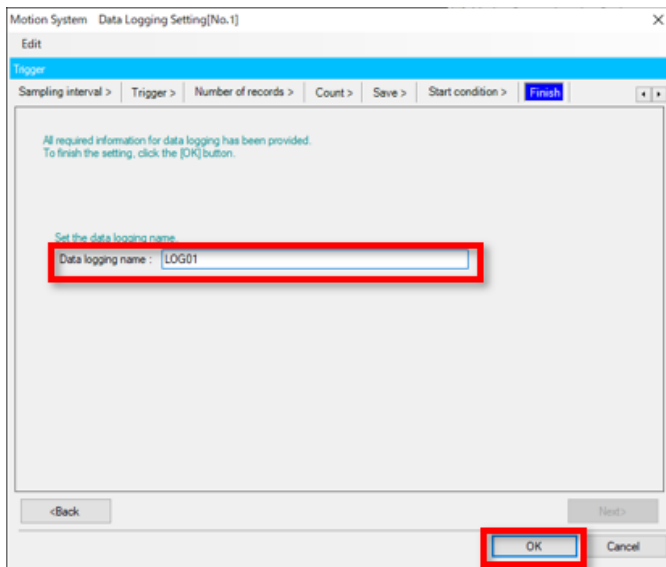
- (8) O formato do arquivo e o número de arquivos salvos dos dados do registro são definidos em [Save]. Neste curso, o valor padrão (formato: JSON, número de arquivos salvos: 1) é definido. Quando a configuração estiver concluída, clique no botão [Next].



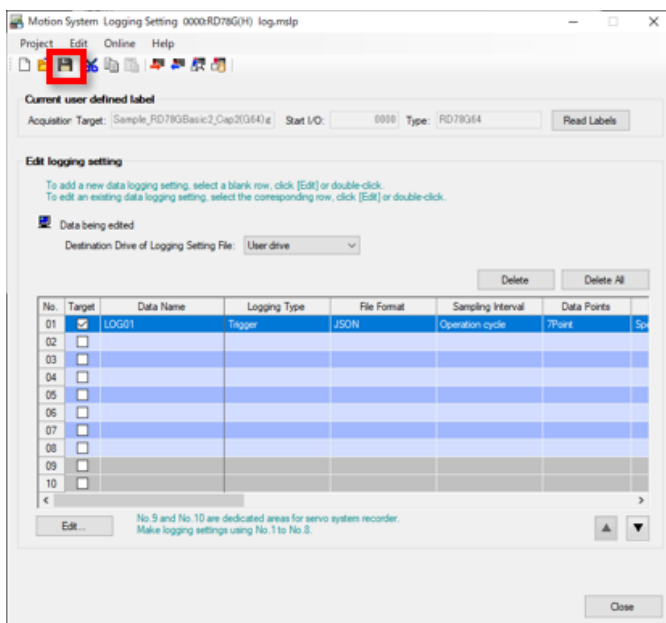
- (9) A condição para início do registro é definida em [Start condition]. Neste curso, "Start by User Operation" é definido. Quando a configuração estiver concluída, clique no botão [Next].



- (10) O nome do registro de dados é definido em [Finish].
Neste curso, o valor padrão (LOG01) é definido.
Quando a configuração estiver concluída, clique no botão [Next].



- (11) Retorne à ferramenta de configuração do registro do sistema de movimento.
As definições que foram configuradas podem ser salvas.
Clique no ícone de salvar, e salve-as no destino que desejar.



As informações de configuração do registro são gravadas.

Clique no ícone de gravação da configuração do registro, selecione a memória de destino e clique no botão [Write]. Uma janela de confirmação será apresentada.

Clique no botão [Yes] e continue. Quando a gravação estiver concluída, clique no botão [OK] e feche a tela.

The process is shown in four sequential screenshots:

- Motion System Logging Setting**: The 'Write' icon in the toolbar is highlighted with a red box.
- Write Logging Setting-0000:RD78G(H) (Host)**: The 'Target memory' dropdown is set to 'User drive' (highlighted with a red box). The 'Write' button at the bottom is also highlighted with a red box.
- Motion System Logging Configuration Tool**: A warning dialog box with a yellow triangle icon. The text reads: "Write the logging settings to the motion system. <Connection Destination> RD78G8. The following setting No., which has been included in the selected logging settings, already exists in the target memory of the write destination. - No.01. Writing settings will overwrite the logging setting in the target memory. Do you want to continue?". The 'Yes' button is highlighted with a red box.
- Motion System Logging Configuration Tool**: An information dialog box with a blue 'i' icon. The text reads: "Completed writing the logging setting data. <Caution> Logging is not started just by writing logging settings. Start logging in the 'Logging Status Display and Operation' window. To display the 'Logging Status Display and Operation' window, select [Online] -> [Logging Status Display and Operation]". The 'OK' button is highlighted with a red box.

Quando "Start by User Operation" estiver definido como 5.2 (9), clique no ícone [Logging Status and Operation] para exibir a tela [Logging Status and Operation] e iniciar o registro.

Quando o nome da configuração do registro dos dados que será executada for selecionado, clique no botão [Start], e o LoggingStatus mudará para "Waiting for trigger".

Quando o programa é executado nesse estado e a condição de acionamento é satisfeita (quando X24 é acionado, neste exemplo do curso), o status muda para "Triggered".

Quando o registro é concluído, o status muda para "CollectionCompleted", a partir de "Saving".

The image shows two screenshots of the Motion System software interface. The left screenshot is the 'Logging Setting' window, and the right is the 'Logging Status and Operation' window. A red arrow points from the 'Start' button in the right window to a flowchart on the right side of the image.

Logging Setting Window (Left):

- Current user defined label: Sample_RD790Basic2_Cap3(044) e Start I/O: 0000 Type: RD79064
- Destination Drive of Logging Setting File: User drive
- Table with columns: No., Target, Data Name, Logging Type, File Format, Sampling Interval, Data Points.

No.	Target	Data Name	Logging Type	File Format	Sampling Interval	Data Points
01	<input checked="" type="checkbox"/>	LOG01	Trigger	.LOG	Operation cycle	7988
02	<input type="checkbox"/>					
03	<input type="checkbox"/>					
04	<input type="checkbox"/>					
05	<input type="checkbox"/>					
06	<input type="checkbox"/>					
07	<input type="checkbox"/>					
08	<input type="checkbox"/>					
09	<input type="checkbox"/>					
10	<input type="checkbox"/>					

Logging Status and Operation Window (Right):

- Monitor status: Monitoring (Stop)
- User Drive Free Volume: 47852 MB
- RAM Drive Free Volume: 47852 MB
- SD Memory Card Free Volume: 47852 MB
- Logging status and operation: Display the executing logging status or execute a logging start/stop operation.
- Motion System Data table:

No.	Target	Data Name	Logging Type	Sampling Interval [ms]	Stopped
<input checked="" type="checkbox"/>	01	User drive	LOG01	Trigger	<input checked="" type="checkbox"/>
<input type="checkbox"/>	02	User drive			<input type="checkbox"/>
<input type="checkbox"/>	03	User drive			<input type="checkbox"/>
<input type="checkbox"/>	04	User drive			<input type="checkbox"/>
<input type="checkbox"/>	05	User drive			<input type="checkbox"/>
<input type="checkbox"/>	06	User drive			<input type="checkbox"/>
<input type="checkbox"/>	07	User drive			<input type="checkbox"/>
<input type="checkbox"/>	08	User drive			<input type="checkbox"/>
<input type="checkbox"/>	09	User drive			<input type="checkbox"/>
<input type="checkbox"/>	10	User drive			<input type="checkbox"/>

Logging Status Flowchart (Right):

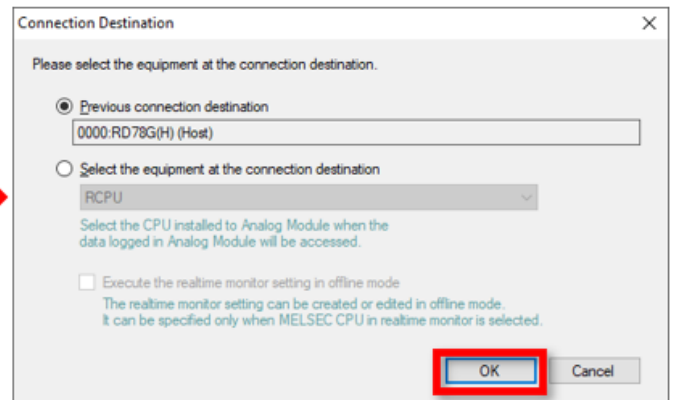
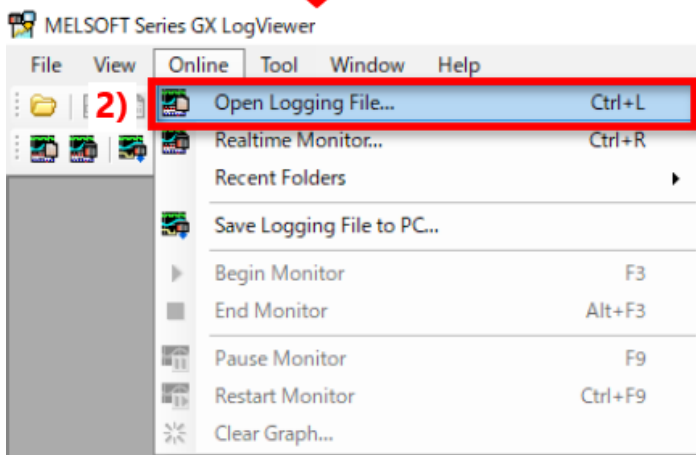
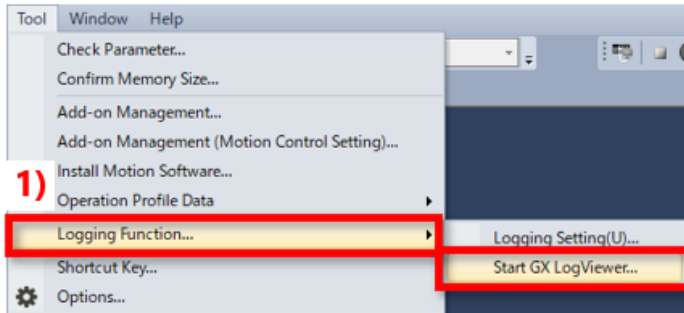
- Logging Status: Waiting for trigger
- Logging Status: Triggered
- Logging Status: Saving
- Logging Status: CollectionCompleted

GX LogViewer é usado para a leitura dos dados do registro.

Selecione [Tool] → [Logging Function] → [Start GX LogViewer], na barra de ferramentas da tela Motion Control Setting Function.

Quando o GX LogViewer for iniciado, selecione [Online] → [Open Logging File].

Selecione "0000:RD78G(H) (Host)" na tela Connection Destination. (Nota)

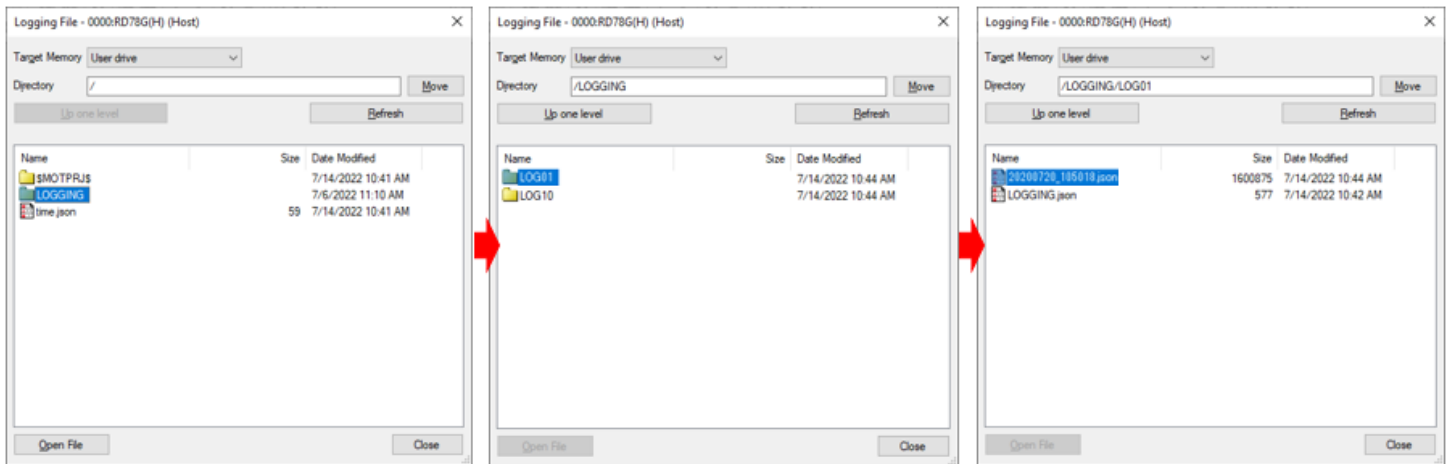


(Nota) Se o GX LogViewer já tiver sido iniciado e a comunicação com o módulo de movimento já estiver definida, essa tela não será exibida.

Selecione o arquivo de registro que será lido.

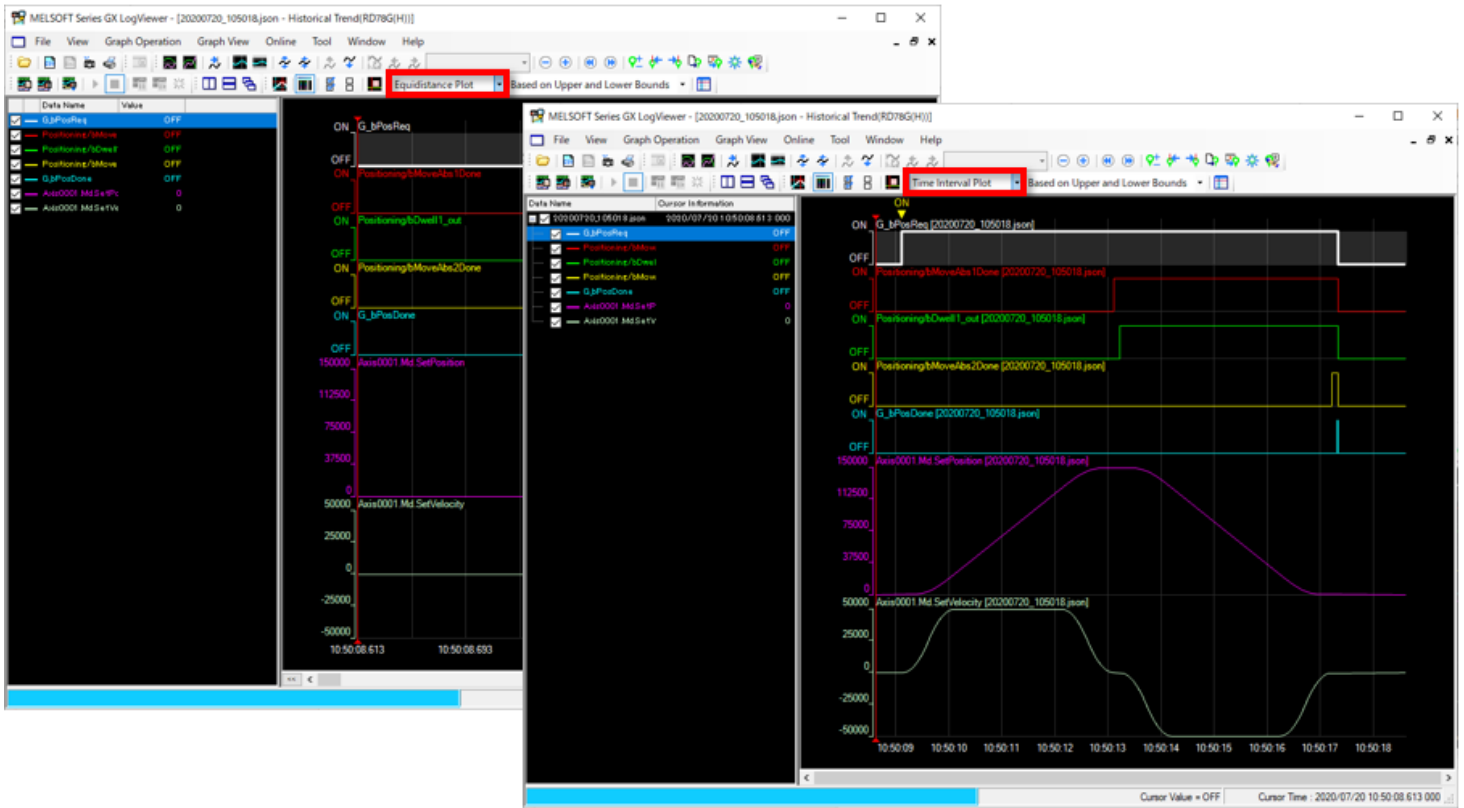
No exemplo deste capítulo, selecionamos a unidade do usuário, em "LOGGING" → "LOG1" → "(Logged date and time).json".

Selecione o nome do arquivo e clique no botão [Open File].



Os dados em forma de onda registrados no GX LogViewer são exibidos.

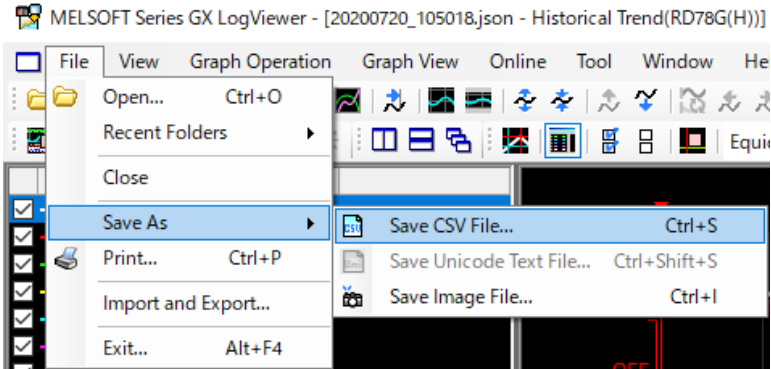
Quando o formato da plotagem é alterado de "Equidistance Plot" para "Time Interval Plot", é possível exibir toda a forma de onda registrada.



Os dados em forma de onda registrados podem ser salvos no formato de arquivo csv ou json.
(Quando o registro é no formato CSV, os dados podem ser salvos em um arquivo CSV.)

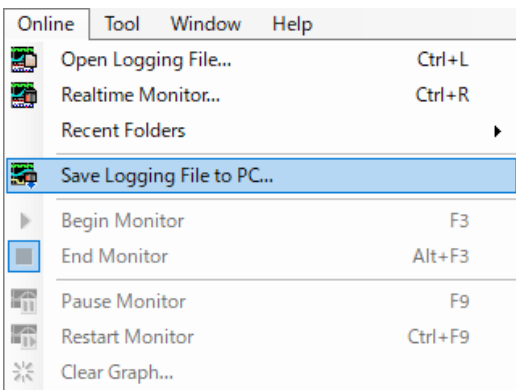
1) Salvando como um arquivo csv

Selecione [File] → [Save As] → [Save CSV File], na barra de ferramentas do GX LogViewer.



2) Salvando como um arquivo json

Selecione [Online] → [Save Logging File to PC], na barra de ferramentas do GX LogViewer.



Neste capítulo, você aprendeu o seguinte:

- Iniciando a ferramenta de configuração do registro
- Definição dos dados a serem registrados
- Gravando a configuração do registro
- Iniciando o registro
- Lendo os dados do registro
- Salvando os dados do registro

Ponto

Iniciando a ferramenta de configuração do registro	<ul style="list-style-type: none">• Inicie a ferramenta de configuração do registro do sistema de movimento da função de configuração do motion controller.
Definição dos dados a serem registrados	<ul style="list-style-type: none">• Defina os dados que serão registrados, as condições de acionamento e outros fatores, seguindo o procedimento que aparece na ferramenta de configuração do registro do sistema de movimento.
Gravando a configuração do registro	<ul style="list-style-type: none">• Grave os dados de configuração do registro no módulo de movimento, antes do registro.
Iniciando o registro	<ul style="list-style-type: none">• Quando a condição de início do registro for definida como "Start by User Operation", clique no botão de início, na tela "Logging Status and Operation", para começar o registro.
Lendo os dados do registro	<ul style="list-style-type: none">• GX LogViewer é usado para a leitura dos dados do registro.
Salvando os dados do registro	<ul style="list-style-type: none">• Os dados em forma de onda registrados podem ser salvos no formato de arquivo csv ou json.

Agora que você concluiu todas as lições do curso Introdução ao Módulo de Movimento da Série **MELSEC iQ-R (RD78G(H) Controle de posicionamento)**, já está pronto para fazer o teste final. Se tiver alguma dúvida sobre os tópicos abrangidos, aproveite esta oportunidade para rever esses tópicos.

Há um total de 4 perguntas (7 itens) neste Teste Final.

Você pode fazer o teste final quantas vezes quiser.

Resultados da pontuação

O número de respostas corretas, o número de perguntas, a porcentagem de respostas corretas, e o resultado de aprovação/reprovação aparecerá na página de pontuação.

		1	2	3	4	5	6	7	8	9	10	11	12	
Tentar novamente	Teste 1	✓	✓	✗	✓									Total de perguntas: 28
	Teste 2	✓	✓	✓	✓									Respostas corretas: 23
	Teste 3	✓												Porcentagem: 82 %
	Teste 4	✓	✓											
	Teste 5	✓	✓											
Tentar novamente	Teste 6	✓	✗	✗	✗									
	Teste 7	✓	✓	✓	✓									
	Teste 8	✓	✓	✓	✓	✓								
	Teste 9	✓												
Tentar novamente	Teste 10	✗												

Para receber aprovação no teste é necessário acertar **60%** das respostas.

Selecione as descrições corretas de uma label pública. (É possível selecionar várias respostas.)

A label pública é uma label compartilhada que pode ser usada no módulo de movimento e também na CPU do PLC.

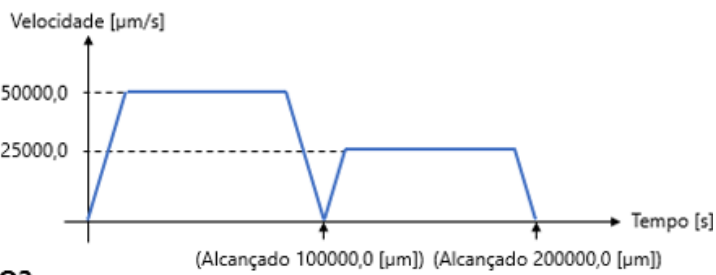
A label pública é registrada a partir da label global da CPU do PLC.

Quando a label global for definida como a label pública, selecione se a label será lida ou gravada de/para a CPU do PLC.

FB1 é executado em primeiro lugar, e em seguida FB2.
 Quando a posição alvo e a velocidade alvo de FB1 e FB2 forem aquelas apresentadas na tabela a seguir, selecione o modo de buffer que será executado a seguir.

	Posição alvo	Velocidade alvo
FB1	100000,0 [μm]	50000,0 [μm/s]
FB2	200000,0 [μm]	25000,0 [μm/s]

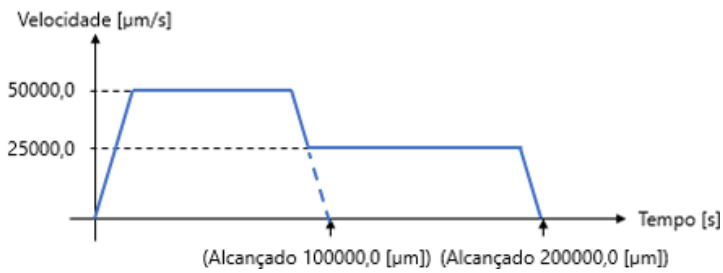
Q1



Q1

Q2

Q2



- Q1:** 1: mcAborting
 2: mcBuffered
 3: mcBlendingNext
 4: mcBlendingPrevious

- Q2:** 1: mcBlendingNext e mcBlendingHigh
 2: mcBlendingPrevious e mcBlendingHigh
 3: mcBlendingNext e mcBlendingLow
 4: mcBlendingPrevious e mcBlendingLow

Selecione as frases corretas, entre as apresentadas a seguir, para a programação com a CPU do PLC. (É possível selecionar várias respostas.)

A biblioteca de FBs deve ser registrada no GX Works3 para utilização do Motion control FB para o módulo de movimento na CPU do PLC.

Insira o motion control FB no editor de programas da árvore do projeto do GX Works3.

Não existem parâmetros a serem definidos para o módulo de movimento.

Preencha os espaços em branco selecionando as respostas adequadas.

- Inicie (Q1) para definir os dados a serem registrados.
- Grave os dados do registro em (Q2) para executar o registro.
- (Q3) é usado para ler os dados do registro e verificar a forma de onda.

Q1

Selecione a resposta adequada.

**Q2**

Selecione a resposta adequada.

**Q3**

Selecione a resposta adequada.



Q1: 1: CPU module logging configuration tool
2: Ferramenta de configuração do registro do sistema de movimento

Q2: 1: Módulo da CPU
2: Módulo de movimento
3: Servo amplificador

Q3: 1: MR Configurator2
2: GX LogViewer

Selecione as descrições corretas de uma label pública. (É possível selecionar várias respostas.)

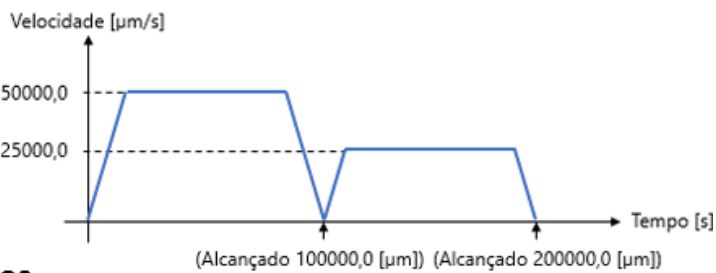
- A label pública é uma label compartilhada que pode ser usada no módulo de movimento e também na CPU do PLC.**
- A label pública é registrada a partir da label global da CPU do PLC.**
- Quando a label global for definida como a label pública, selecione se a label será lida ou gravada de/para a CPU do PLC.**

FB1 é executado em primeiro lugar, e em seguida FB2.

Quando a posição alvo e a velocidade alvo de FB1 e FB2 forem aquelas apresentadas na tabela a seguir, selecione o modo de buffer que será executado a seguir.

	Posição alvo	Velocidade alvo
FB1	100000,0 [μm]	50000,0 [μm/s]
FB2	200000,0 [μm]	25000,0 [μm/s]

Q1



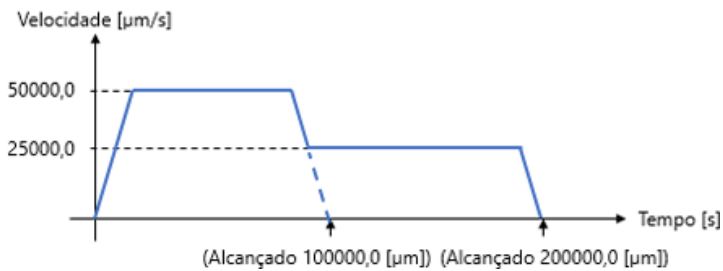
Q1

2: mcBuffered

Q2

3: mcBlendingNext e mcBlendingLow

Q2



- Q1:** 1: mcAborting
 2: mcBuffered
 3: mcBlendingNext
 4: mcBlendingPrevious

- Q2:** 1: mcBlendingNext e mcBlendingHigh
 2: mcBlendingPrevious e mcBlendingHigh
 3: mcBlendingNext e mcBlendingLow
 4: mcBlendingPrevious e mcBlendingLow

Selecione as frases corretas, entre as apresentadas a seguir, para a programação com a CPU do PLC. (É possível selecionar várias respostas.)



A biblioteca de FBs deve ser registrada no GX Works3 para utilização do Motion control FB para o módulo de movimento na CPU do PLC.



Insira o motion control FB no editor de programas da árvore do projeto do GX Works3.



Não existem parâmetros a serem definidos para o módulo de movimento.

Preencha os espaços em branco selecionando as respostas adequadas.

- Inicie (Q1) para definir os dados a serem registrados.
- Grave os dados do registro em (Q2) para executar o registro.
- (Q3) é usado para ler os dados do registro e verificar a forma de onda.

Q1

2: Ferramenta de configuração do registro do sistema de movimento

**Q2**

2: Módulo de movimento

**Q3**

2: GX LogViewer



Q1: 1: CPU module logging configuration tool
2: Ferramenta de configuração do registro do sistema de movimento

Q2: 1: Módulo da CPU
2: Módulo de movimento
3: Servo amplificador

Q3: 1: MR Configurator2
2: GX LogViewer

Você concluiu o teste final. Esta é sua área de resultados.
Para terminar o Teste Final, vá para a próxima página.

	1	2	3	4	5	6	7	8	9	10
Teste Final 1	✓									
Teste Final 2	✓	✓								
Teste Final 3	✓									
Teste Final 4	✓	✓	✓							

Total de perguntas: **7**
Respostas corretas: **7**
Porcentagem: **100 %**

Limpar

Você concluiu o curso "MELSEC iQ-R Series Motion Module Application (RD78G(H) Positioning Control)".

Muito obrigado por fazer este curso.

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

Você pode rever o curso quantas vezes quiser.

Revisar

Fechar