

Configurando a Ponte Rolante

Ataias Pereira Reis
Emanuel Pereira Barroso Neto

November 13, 2015

1 CONEXÕES IMPORTANTES

O presente documento tem por objetivo ser um guia prático para a configuração do sistema da ponte rolante a ser utilizado na validação experimental do controle de *risers* em malha fechada.

1.1 COMPONENTES

1.1.1 CONTROLADOR LÓGICO-PROGRAMÁVEL

O controlador lógico-programável (CLP - ver Figura 1.1) é a espinha dorsal da bancada. Ele é responsável por executar os comandos de controle vindos do computador sobre todos os elementos que estão conectados a ele.

O CLP utilizado é fabricado pela *Allen Bradley*, modelo Logix5560M03SE. Tal modelo possui memória lógica e de dados de 750 KiB, e memória de I/O de 494 KiB. Há quatro módulos no *chassis* do controlador:

- O próprio controlador;
- *SERCOS Interface*;
- DeviceNET;
- EtherNet/IP.

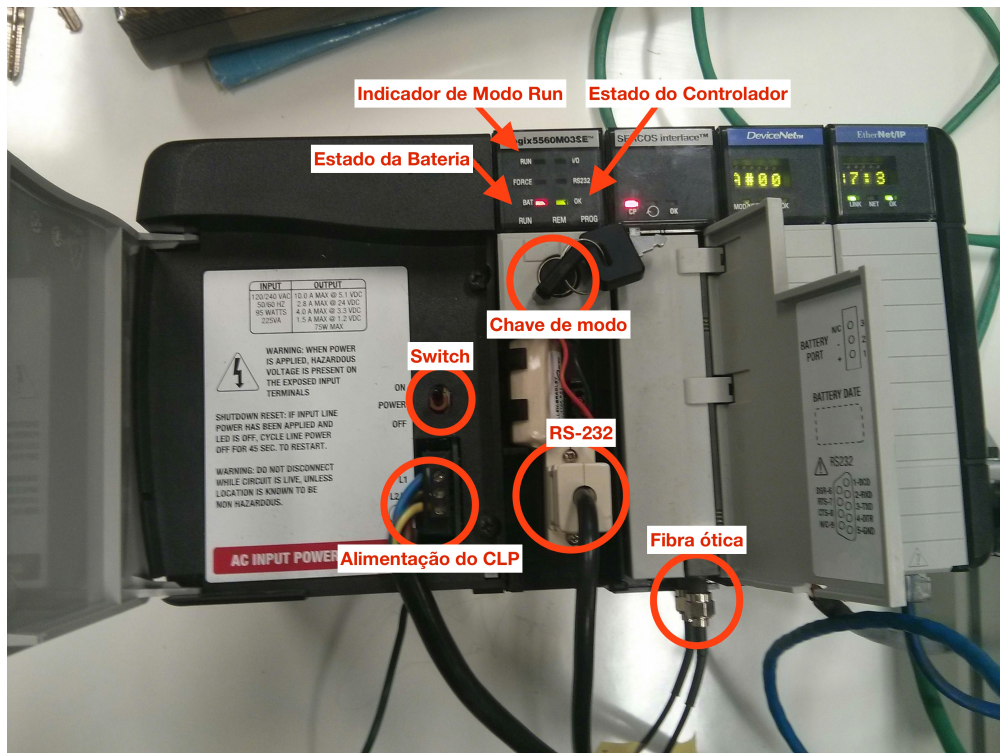


Figure 1.1: CLP com identificação de elementos

Além dos módulos, o controlador ainda possui um *switch* liga/desliga presente no *chassis*. No módulo Logix, há uma chave responsável por alterar o modo de funcionamento do mesmo. As posições possíveis dessa chave são:

- RUN;
- REM;
- PROG;

Na prática, o modo REM se divide em dois modos: REM RUN e REM PROG. A maneira de se diferenciar os dois é observar, no módulo Logix, o estado do LED indicador de modo RUN quando a chave estiver na posição REM.

No modo RUN, o controlador apenas roda o programa presente em sua memória; não há qualquer comunicação remota. No modo PROG, o controlador não roda nenhum programa; ele apenas pode receber um novo código. Nos modos REM, há a comunicação com o computador, permitindo verificar valores de variáveis de interesse e alterar, se necessário, o programa a ser rodado pelo controlador. O programa presente no controlador, em modo REM, só roda o código se estiver no modo REM RUN; se for necessário atualizar o programa, o modo deve ser o REM PROG.

1.1.2 DEVICENET

O DeviceNET é a rede responsável por estabelecer a conexão do CLP com os sensores indutivos. A rede é alimentada com 24 Volts externa, pois o módulo DeviceNET não fornece tal alimentação.

A rede DeviceNET também está conectada a um módulo de saídas, o *CompactBlock I/O* (Figura 1.2). Esse bloco, na presente configuração, apenas transmite a alimentação da fonte de 24 V para os sensores e o módulo DeviceNET. A alimentação da rede também foi conectada à alimentação da câmera.

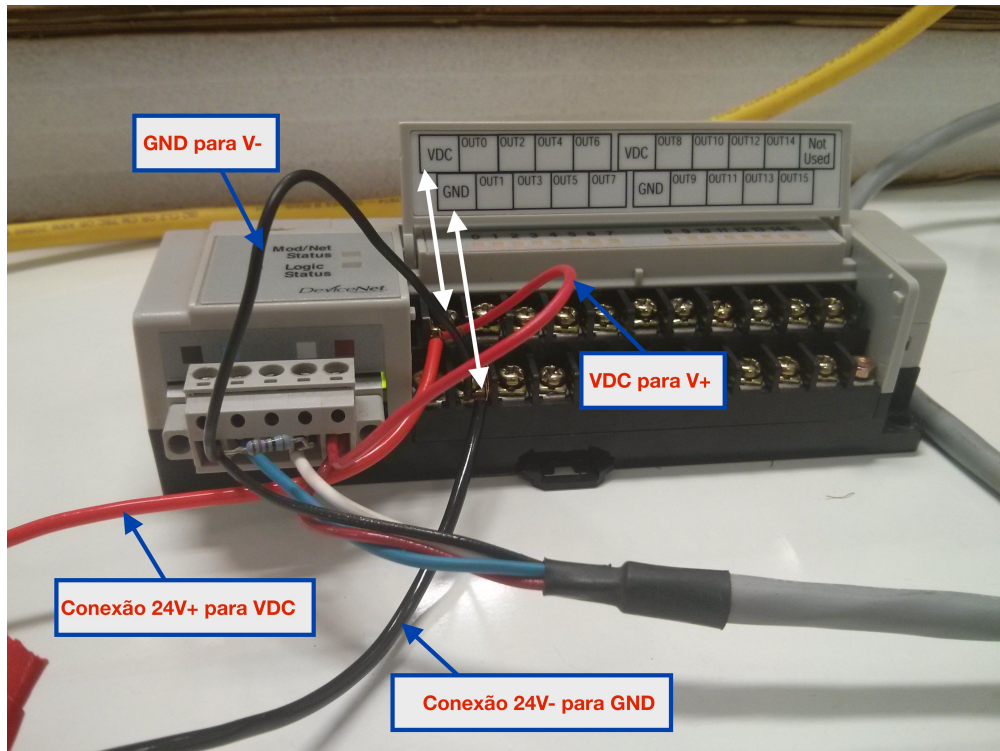


Figure 1.2: CompactBlock com identificação de elementos

1.1.3 SENSORES INDUTIVOS

Os sensores indutivos são elementos detectores de presença, particularmente de objetos metálicos. Eles funcionam através da variação de campo magnético ocasionada pela presença do objeto a ser identificado. Tal variação de campo magnético provoca uma variação de corrente dentro do sensor, alterando seu estado.

Na presente bancada, há 6 sensores indutivos da família 871TM, fabricados pela *Allen Bradley*. Eles são alimentados com tensão de 24 V, que está dentro dos limites padrão. São

sensores feitos de aço, adaptados a ambientes industriais.

1.1.4 CÂMERA

Cor do Fio	Descrição
Amarelo	RS-232 TX
Cinza	Remote Teach
Laranja	Product Change
Rosa	External Trigger
Preto	Discrete I/O #1
Vermelho	Discrete I/O #2
Branco	Discrete I/O #3
Azul-Claro	Discrete I/O #4
Violeta	RS-232 RX
Verde	RS-232 Signal Ground
Azul	Common (Signal Ground)
Marrom	10-30V DC

Table 1.1: Fiação da câmera

1.2 CUIDADOS COM AS CONEXÕES E COM O CONTROLADOR

Para que o sistema funcione corretamente, alguns cuidados devem ser tomados:

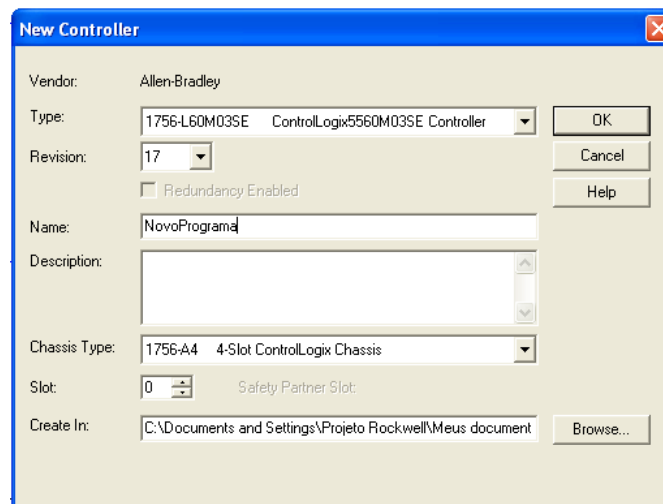
1. A rede DeviceNET deve ser alimentada corretamente; caso contrário, o módulo DeviceNET emitirá uma mensagem de erro “No Network Power”; caso essa mensagem seja mostrada, os sensores indutivos também não estarão alimentados, e não funcionarão.
2. O *switch* liga/desliga do controlador é protegido por uma tampa. É recomendável que, com o controlador em operação, que esta tampa esteja fechada.
3. Atentar para o estado do disjuntor industrial; ele deve estar ligado para que o motor funcione.
4. Não retirar a chave de modo do controlador; sem ela, pode ser impossível carregar um novo programa para o controlador ou mesmo rodar um programa já carregado.

2 CRIANDO UM NOVO PROGRAMA NO RSLOGIX

Esta seção tem por objetivo demonstrar os passos para criar um programa novo e fazer as configurações básicas para que um programa simples possa ser executado. Aqui, já se considera que o RSLinx tenha sido utilizado para verificar as conexões dos dispositivos. Note que, quando não mencionado o nome para se dar a um programa ou dispositivo, é livre a escolha, contanto que seja consistente.

2.1 ADICIONANDO DISPOSITIVOS

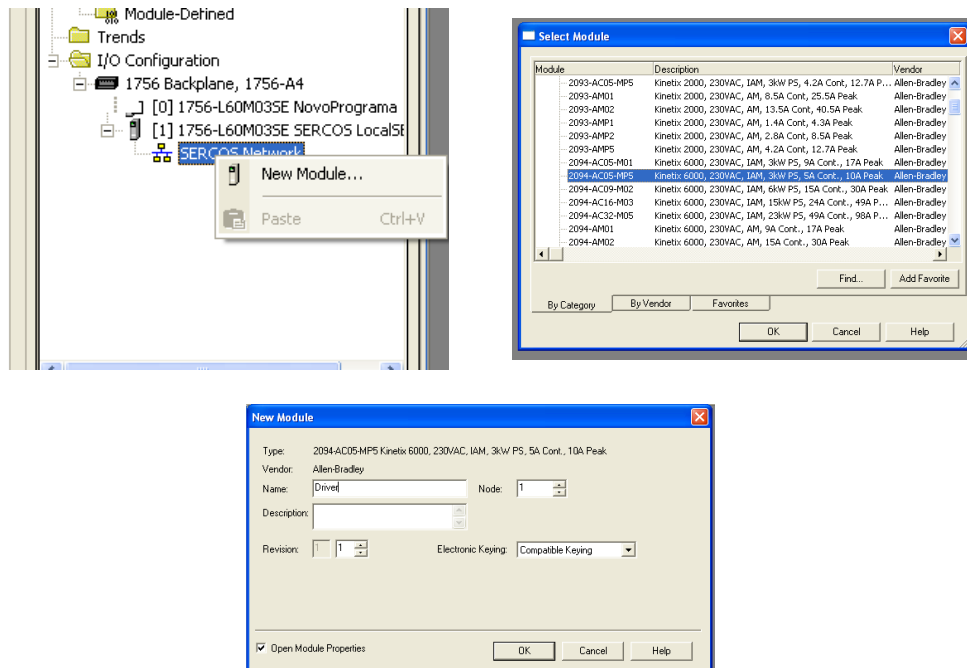
1. Crie um novo controlador (Ctrl+N) e configure da seguinte maneira



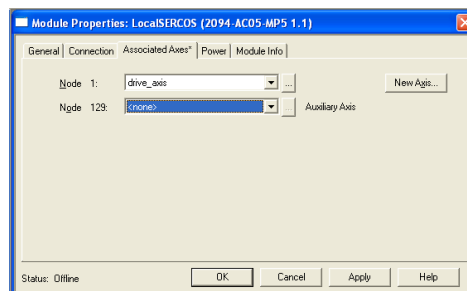
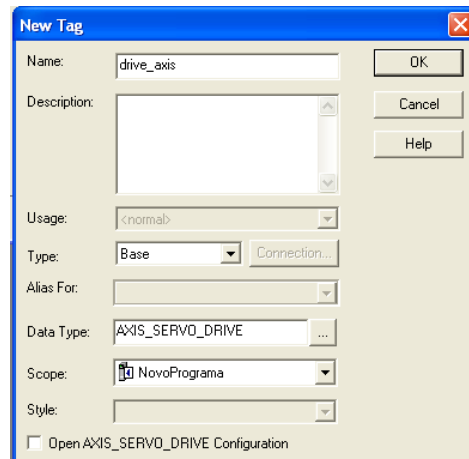
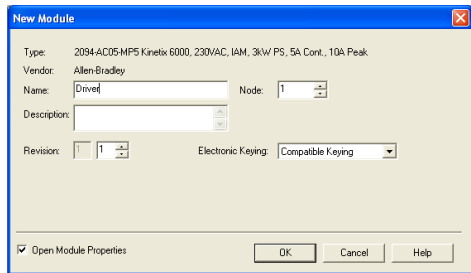
The 'New Controller' dialog box is shown with the following configuration:

- Vendor: Allen-Bradley
- Type: 1756-L60M03SE ControlLogix5560M03SE Controller
- Revision: 17
- ☐ Redundancy Enabled
- Name: NovoPrograma
- Description: (empty)
- Chassis Type: 1756-A4 4-Slot ControlLogix Chassis
- Slot: 0 (Safety Partner Slot)
- Create In: C:\Documents and Settings\Projeto Rockwell\Meus document

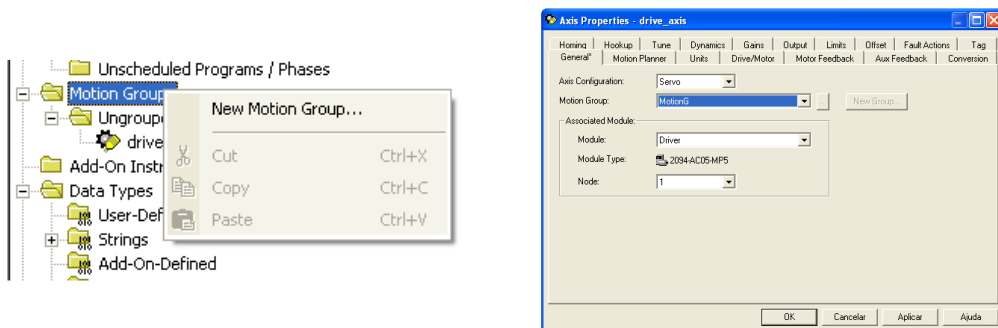
2. Adicione o driver 2094-AC05-MP5 no SERCOS



3. Abra as propriedades do módulo 2094-AC05-MP5: Agora, vá na aba *Associated Axes* e clique em *New Axis*. Dê um nome ao eixo e confirme. Após isso, associe o eixo ao nó 1. Veja figuras abaixo para clarificar dúvidas.



4. **Crie um motion group:** Vá em *Motion Group* e clique em *New Motion Group*. Dê um nome e clique e OK.

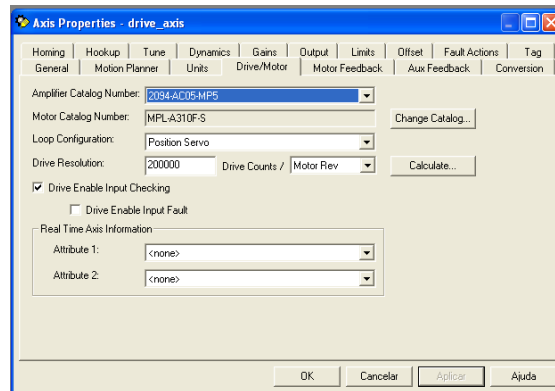


5. **Associar eixo ao Motion Group:** nas propriedades do eixo criado, adicione ao grupo. O resultado deve mostrar o eixo dentro do grupo de movimento.

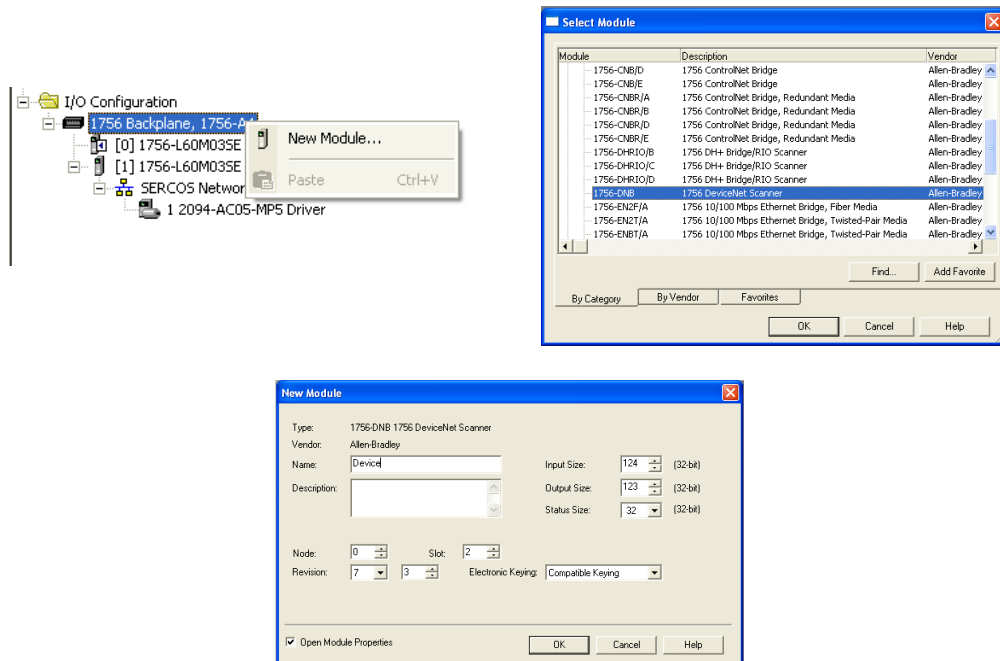


6. **Associar motor ao driver:** Vá nas propriedades do eixo, seleciona a aba *Drive/Motor*,

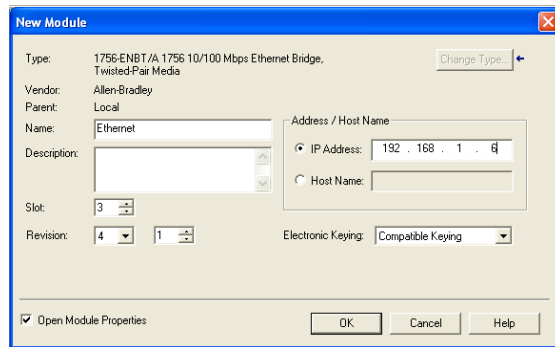
clique em *Change Catalog*, selecione MPL-A310F-S e clique em OK.



7. **Adicionar DeviceNet:** No backplane, clique em novo módulo. Daí, vá em *Communications* e adicione o 1756-DNB. Escolha a revisão 7 e clique em OK. Na sub-revisão, escolha 3 e pode clicar em OK para fechar as propriedades do módulo.



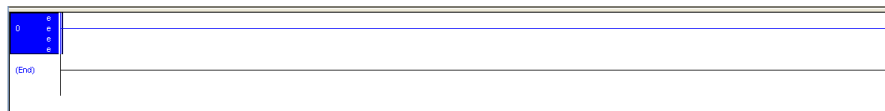
8. **Adicionar módulo Ethernet/IP:** crie um novo módulo no backplane e então vá em *Communications* e selecione 1756-ENBT/A e use revisão 4 com sub-revisão 1. O endereço IP pode ser definido como 192.168.0.4, por exemplo. (ou não? é definido no RSLinx?)



A configuração dos dispositivos foi terminada (talvez o que falte seja a câmera... não sei se é necessário que ela seja adicionada no programa).

2.2 PROGRAMANDO

1. **Abra o programa principal:** deve-se observar uma linha em branco de ladder.



2. **Crie o programa principal:** a Figura 2.1 mostra o programa principal. Note que as variáveis MAJ_1 e MAJ_2 devem ser criadas. Pode-se fazer isso clicando em cada uma delas com o botão esquerdo e escolhendo a opção apropriada. Configurações completas para o bloco MAJ estão na Figura 2.2. O bloco MSO liga o servomotor e MAJ muda a velocidade, veja a documentação dos blocos para maiores informações. O programa principal faz o carrinho se mexer em um sentido quando o sensor 2 for ativado e em outro sentido quando o sensor 5 for ativado.

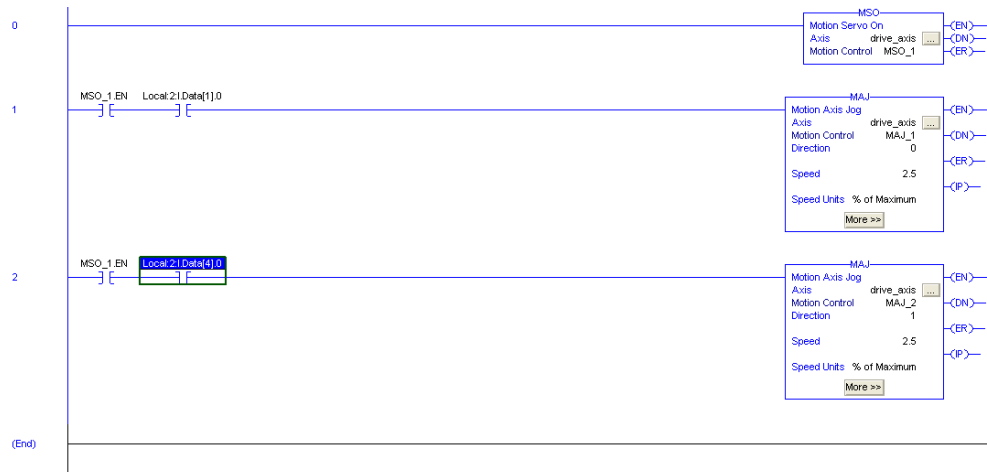
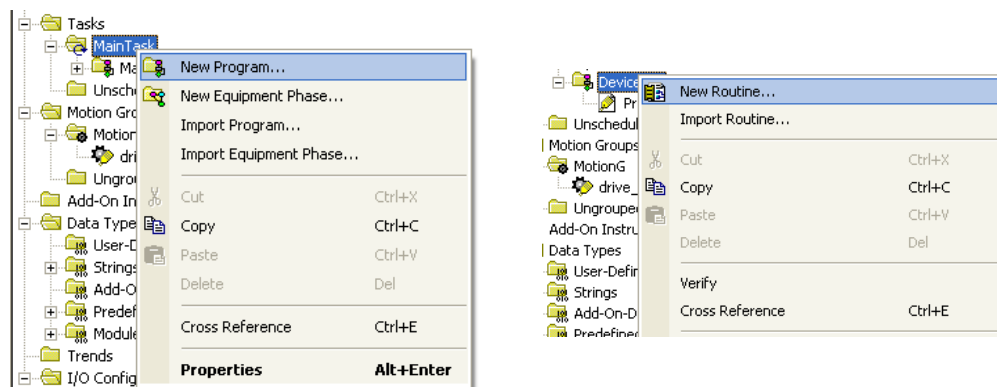


Figure 2.1: Programa principal



Figure 2.2: Configurações do MAJ

- Crie o programa que liga a rede DeviceNet:** crie um novo programa dentro de *Main-Task* e após isso crie uma rotina do tipo Ladder Diagram dentro desse novo programa. O programa que deve ser criado está na Figura 2.3.



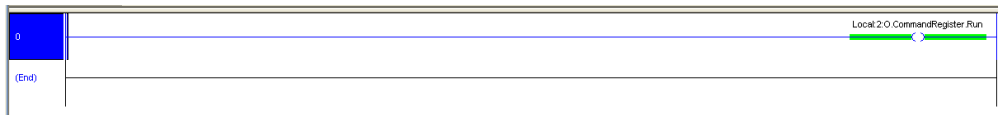
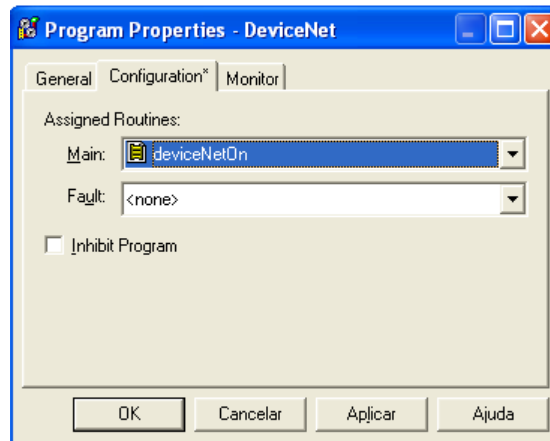


Figure 2.3: Programa que liga DeviceNet

4. **Torne a rotina criada uma rotina principal:** vá nas propriedades do programa, selecione a aba *Configuration* e escolha o nome da sua rotina criada como a rotina *Main*.



5. **Crie uma rotina de segurança:** crie um novo programa, crie uma nova rotina do tipo Ladder Diagram e então marque essa rotina como principal. Após isso, programe conforme a Figura 2.4. Nessa rotina, o motor para caso certos sensores sejam ativados.

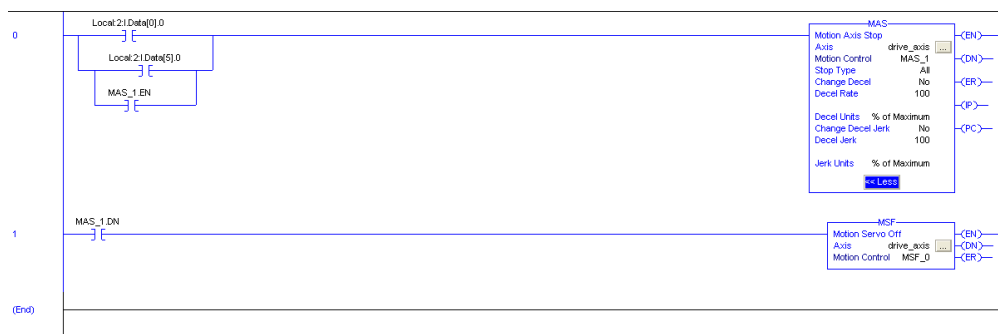


Figure 2.4: Rotina de segurança

6. **Execute o programa:** Para rodar o programa pela primeira vez, clique no menu “Communications” e depois “Who Active”. Para usar o RS para comunicação, selecione AB_DF1-1, DF1 e depois clique em download. Na primeira vez, lembre-se de verificar a opção “Designate this controller as CST master”.

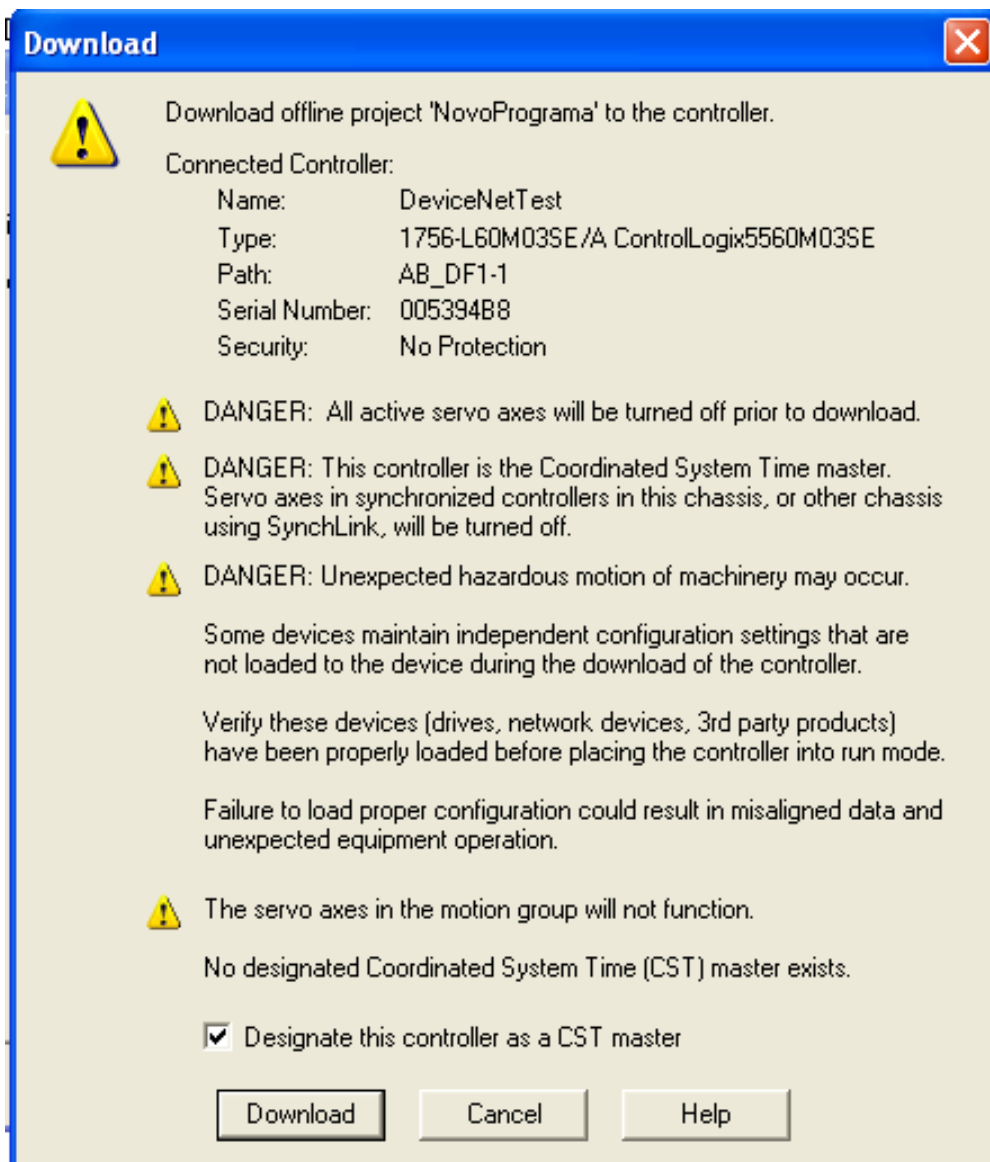


Figure 2.5: Mensagem que aparece quando se executa o programa e opção que aparece pela primeira vez

2.3 PROGRAMANDO II: TEXTO ESTRUTURADO

O controlador presentemente utilizado permite que programas razoavelmente grandes escritos em *ladder* possam ser executados e guardados em sua memória. O controlador 1560 possui cerca de 760 kB de memória. Porém, pode ser necessário se utilizar uma linguagem mais simples, uma vez que o espaço pode ser um problema e a linguagem *ladder* não é conveniente em casos que muitos parâmetros devem ser alterados por fora, ao mesmo tempo. Neste caso, torna-se interessante o uso de uma outra linguagem, puramente textual. Essa

linguagem é conhecida como texto estruturado (*Structured Text*).

Para se utilizar o texto estruturado, basta seguir os seguintes passos:

1. **Criar uma nova rotina:** Esse passo é semelhante à criação de uma nova rotina em *ladder*, mas a linguagem a ser selecionada é o texto estruturado.
2. **Programação em texto:** Para se programar em texto estruturado, deve-se ter em mente que o raciocínio é o mesmo empregado para o *ladder*; as *tags* utilizadas devem existir, por exemplo. Caso seja escrita uma *tag* ainda inexistente, o *RSLogix 5000* irá sublinhar a *tag* com uma linha vermelha, sinalizando que há um erro; basta clicar com o botão direito do *mouse* no texto e adicionar a *tag*, clicando na opção “*New Tag...*”. A partir desse passo, as configurações da nova *tag* são feitas do mesmo modo que na programação *ladder*.
3. **Texto estruturado x *ladder*:** Embora a maioria das instruções *ladder* possua um equivalente normal em texto estruturado, há algumas instruções que devem ser notadas por não existir em texto estruturado, por exemplo; há outras instruções que existem em texto, mas não em *ladder*. A tabela 2.1 mostra alguns exemplos:

Table 2.1: Relações entre instruções *ladder* e texto estruturado

Instrução <i>ladder</i>	Instrução em texto
-()- [OTE]	A [:=] X
-(L)- [OTL]	A := 1
-(U)- [OTU]	A := 0
-[]- [XIC]	IF A = 1 THEN ;Code END_IF;

3 CÂMERA

3.1 CONFIGURANDO A CÂMERA

Para que a câmera possa ser utilizada, é necessário que se estabeleça uma comunicação entre ela, o computador e o controlador, para que seja possível efetuar sua calibração e seu uso como sensor visual presente no experimento. Tendo em vista este objetivo, faz-se necessária a configuração de uma rede, que, neste experimento, será uma rede *Ethernet* com um *switch*. Os passos a seguir descrevem essa configuração.

1. **Configuração física da rede:** A câmera, o controlador e o computador devem estar ligados entre si através de cabos *Ethernet*. Todos os cabos devem ser conectados ao *switch*.
- 2.

3.

4.

3.2 CALIBRAÇÃO

A câmera permite fazer medidas de distâncias em pixels. De forma a se converter essa distância para milímetros, uma barra de alumínio com marcas e tamanho conhecido é utilizada. É importante primeiro calibrar o sistema, para se saber se há deformação de pixels significativa ao longo da distância de interesse. No PresencePlus P4 GEO 1.3, um programa com imagem de referência é feito, conforme Figura A barra de alumínio atualmente utilizada tem comprimento total de 532mm. Algumas marcas foram feitas e a Tabela 3.1 apresenta os resultados para cada seção. A distância entre duas marcas é de 10cm.

Table 3.1: Relações mm/px para diferentes seções da barra de alumínio

Seção 1	Seção 2	Distância (px)	mm/px
P0	P10	160	0.625
P10	P20	173	0.578
P20	P30	176	0.568
P30	P40	173	0.578
P40	P50	163	0.613
P0	PEND	893	0.596

O maior desvio da quantidade de milímetros por pixels das seções em relação à da barra inteira é de aproximadamente 4.93%. Há algumas imprecisões na maneira como os traços foram desenhados e é possível que o erro seja menor.