



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PMI - DEPARTAMENTO DE ENGENHARIA DE MINAS E PETRÓLEO

PEA - DEPARTAMENTO DE ENGENHARIA DE ENERGIA E AUTOMAÇÃO ELÉTRICAS

Eduardo Poleze, Deinar Mercaldi Rafani, Marcos Yukio Yamaguchi
Prof. Dr. Carlos Frederico Meschini, Prof. Dr. Cícero Couto de Moraes

LABORATÓRIO DE ELETROTÉCNICA GERAL

EXPERIMENTO: CONTROLADOR LÓGICO PROGRAMÁVEL

Aluno

Nº. USP

2018

1. INTRODUÇÃO

1.1.Histórico

No fim da década de 1960, os circuitos integrados permitiram o desenvolvimento de minicomputadores, que foram logo utilizados para controle on-line de processos industriais.

Em 1969, surgiram os primeiros controladores baseados numa especificação da General Motors, devido à grande dificuldade de mudar a lógica de controle de painéis de comando a cada mudança na linha de montagem. Tais mudanças implicavam em altos gastos de tempo e dinheiro.

Sob a liderança do engenheiro Richard Morley, foi preparada uma especificação que refletia as necessidades de muitos usuários de circuitos de comando com contadores, não apenas da indústria automobilística, mas de toda a indústria manufatureira. Nascia, assim, um equipamento bastante versátil e de fácil utilização, que vem se aprimorando constantemente e diversificando cada vez mais os setores industriais e suas aplicações.

Na década de 1970, os controladores passaram a ter microprocessadores, sendo denominados Controladores Lógicos Programáveis (CLPs). Na década de 1980, houve aperfeiçoamento das funções de comunicação dos CLPs, que passaram a ser utilizados em rede.

Desde o seu aparecimento até hoje, muita coisa evoluiu, como a variedade de tipos de entradas e saídas, o aumento da velocidade de processamento, a inclusão de blocos lógicos complexos para tratamento das entradas e saídas e, principalmente, o modo de programação e a interface com o usuário.

1.2.Conceito de automação

A palavra *automation* foi inventada pelo *marketing* da indústria de equipamentos na década de 1960. O neologismo, sem dúvida sonoro, buscava enfatizar a participação do computador no controle automático industrial.

O que significa automação hoje? Entende-se por automação qualquer sistema, *apoiado em computadores*, que substitua o trabalho humano, em favor da segurança das pessoas, da qualidade dos produtos, da rapidez da produção ou da redução de custos, assim aperfeiçoando os complexos objetivos das indústrias e dos serviços. Exemplos: automação da mineração; da manufatura metálica; de processos químicos contínuos; bancária; metroviária; aeroportuária.

É comum pensar que a automação resulta somente do objetivo de reduzir custos de produção. Isto não é verdade: ela decorre mais de necessidades como maior nível de qualidade,

expressa por especificações numéricas de tolerância, maior flexibilidade de modelos para o mercado, maior segurança pública e dos operários, menores perdas materiais e de energia, mais disponibilidade e qualidade da informação sobre o processo e melhor planejamento e controle da produção.

A automação envolve a implantação de sistemas interligados e assistidos por redes de comunicação, compreendendo sistemas supervisórios e interfaces homem-máquina que possam auxiliar os operadores no exercício da supervisão e da análise dos problemas que porventura venham a ocorrer.

1.3. Por que usar o CLP?

O *Controlador Lógico Programável (CLP)* é um dispositivo digital que controla máquinas e processos. Utiliza uma memória programável para armazenar instruções e executar funções específicas: energização/desenergização, temporização, contagem, sequenciamento, operações matemáticas e manipulação de dados.

O desenvolvimento dos CLPs começou em 1968 em resposta a uma necessidade constatada pela General Motors. Naquela época, frequentemente se consumiam dias ou semanas para se alterar um sistema de controle baseado em contadores, e isso ocorria sempre que se mudava um modelo de carro ou se introduziam modificações na linha de montagem. Para reduzir esse alto custo, a GM especificou um sistema com a flexibilidade de um computador, que pudesse ser programado e mantido pelos engenheiros e técnicos nas fábricas. Também era preciso que suportasse o ar poluído, a vibração, o ruído elétrico e os extremos de umidade e temperatura encontrados normalmente num ambiente industrial. Os primeiros CLPs foram instalados em 1969, com sucesso quase imediato. Funcionando como substitutos de circuitos de contadores, eram mais confiáveis que os sistemas originais. Permitiam reduzir os custos dos materiais, da mão-de-obra, da instalação e da localização de falhas; reduziam as necessidades de fiação e os erros a ela associados. Além disso, os CLPs ocupavam menos espaço que os contadores, temporizadores e outros componentes de controle utilizados anteriormente.

Os CLPs caracterizam-se por:

- Robustez adequada aos ambientes industriais (geralmente não incluem vídeo);
- Programação por meio de computadores pessoais (PCs);
- Linguagens amigáveis para o projetista de automação de eventos discretos;
- Permitir tanto o controle lógico quanto o controle dinâmico (P+I+D);
- Incluir modelos capazes de conexões em grandes redes de dados

Um Controlador Lógico Programável automatiza uma grande quantidade de ações, com precisão, confiabilidade, rapidez e pouco investimento. Informações de entrada são analisadas, decisões são tomadas, comandos são transmitidos, tudo concomitantemente com o desenrolar do processo.

Os CLPs são encontrados em diversos setores industriais como *Metalurgia, Engenharia Mecânica e Automotiva, Indústrias Químicas, Indústrias Petroquímicas, Agricultura e Indústria Alimentícia*

1.4.Arquitetura do CLP

Um CLP é constituído basicamente de:

- Fonte de alimentação;
- Unidade Central de Processamento (UCP);
- Memórias dos tipos fixo e volátil;
- Dispositivos de entrada e saída;
- Terminal de programação.

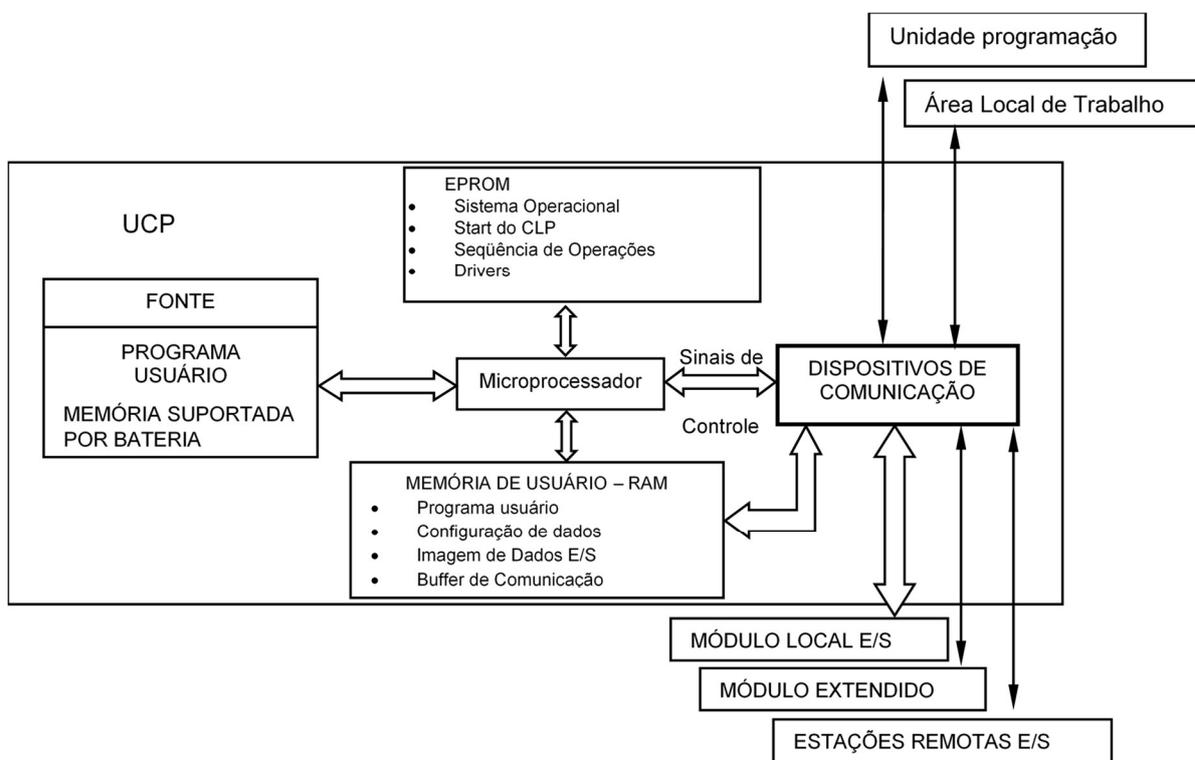


Figura 1 – Arquitetura de um CLP.

Fonte de Alimentação

Converte corrente alternada em contínua para alimentar o controlador. Caso falte energia, há uma bateria que impede a perda do programa do usuário. Ao retornar a energia, o programa se reinicia.

UCP - Unidade Central de Processamento

Responsável pela execução do programa do usuário e pela atualização da memória de dados e da memória-imagem das entradas e saídas.

Memória EPROM

Contém programa elaborado pelo fabricante (*firmware*), que faz o *start-up* do controlador, armazena dados e gerencia a sequência de operações. Este tipo de memória não é acessível ao usuário do controlador programável.

Memória do Usuário

Armazena o programa aplicativo do usuário. A CPU processa esse programa e atualiza a memória de dados internos e a de imagem E/S.

Memória de Dados

Encontram-se aqui dados referentes ao processamento do programa do usuário, isto é, uma tabela de valores manipuláveis.

Memória-Imagem das Entradas e Saídas

Memória que reproduz o estado dos periféricos de entrada e saída.

- Circuitos das entradas são provenientes de chaves, seletoras, limitadoras...
- Circuitos das saídas são destinados a dar partida em motores, solenoides...

A correspondência entre níveis 0 e 1 e níveis de tensão varia conforme a necessidade; por exemplo, pode ocorrer: nível 0 para 0 volt e nível 1 para 115 volts CA.

Módulos de Entrada e Saída (E/S)

Permitem a conexão de entradas digitais e analógicas provenientes de sensores e medidores ao controlador e a conexão de saídas analógicas e digitais para o acionamento de atuadores e para sinalização.

1.5.Linguagem de diagrama de contatos (Ladder)

Nos circuitos de relés ou contadores, cada contato, ao assumir os estados fechado ou aberto, representa uma variável booleana, ou seja, uma variável que assume dois valores: verdadeiro ou falso.

Pela facilidade do desenho e da inspeção de circuitos, e pela longa experiência e tradição dos engenheiros projetistas dos quadros de comando elétrico, uma das primeiras técnicas de programação dos CLPs foi chamada de linguagem “*de relés*” ou “*ladder*” (que quer dizer em escada ou cascata). Essa técnica mantém regras e símbolos tradicionais do projeto de quadros de comando.

Assim, o diagrama *ladder* parte de duas linhas verticais, também chamadas de barras de alimentação. Cada representação de causalidade é feita por uma linha horizontal. Esta linha, por sua vez, é formada por, pelo menos, um elemento controlado (bobina de relé) e um conjunto de condições para o controle desse elemento (rede de contatos).

O diagrama *ladder* é apenas uma representação lógica, trabalhando somente com símbolos, não considerando a tensão envolvida nas barras de alimentação nem a intensidade da corrente pelo circuito. Os contatos e outros dispositivos no diagrama estão em cada momento abertos ou fechados e as bobinas, por consequência, ficam desenergizadas ou energizadas.

O Controlador Lógico Programável examina a continuidade de cada linha, isto é, verifica se todas as variáveis de entrada são verdadeiras. Trata-se de uma “**continuidade lógica**”.

Serão detalhadas as principais instruções e comandos utilizados nesta linguagem.

1.6.Instruções e comandos da linguagem *Ladder*

No *ladder* todos os tipos de instruções pertencem a dois grandes grupos: instruções de entrada e instruções de saída. Nas instruções de entrada são formuladas perguntas enquanto que as instruções de saída executam algum tipo de ação em função das respostas afirmativas ou negativas das instruções de entrada, que estão representadas na mesma linha lógica da instrução de saída.

A CPU do Controlador Programável executa todas as instruções, começando pela primeira instrução da primeira linha do programa indo até a última instrução da última linha do programa do usuário.

Uma instrução de bit pode ser de entrada ou de saída. Durante a execução de uma **instrução de entrada** o estado de um bit em um determinado endereço é examinado. Durante a execução de uma instrução de saída de bit o estado de um bit de um determinado endereço é

alterado para 0 ou 1, conforme haja ou não continuidade lógica da linha com a qual a instrução está relacionada. As principais instruções de bit são apresentadas a seguir:

Instrução NA (XIC) (eXamine If Closed)

A CPU executa esta instrução verificando o valor do bit endereçado pela instrução. Se o bit endereçado pela instrução estiver no estado lógico 0 a instrução retorna com o valor lógico falso e portanto não há continuidade lógica no trecho do *ladder* em que a instrução está inserida. Se o bit endereçado pela instrução estiver no estado lógico 1, a instrução retorna com o valor lógico verdadeiro e portanto há continuidade lógica no trecho do *ladder* em que a instrução está inserida. A representação desta instrução na linguagem *ladder* juntamente com a sua operação faz com que a mesma seja comumente interpretada como um contato normalmente aberto de um relé.

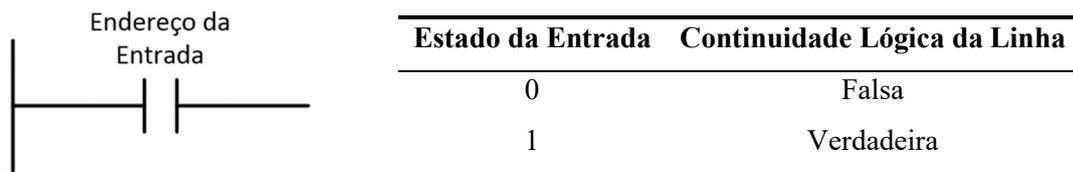


Figura 2 – Instrução NA.

Instrução NF (XIO) (eXamine If Open)

A CPU executa esta instrução verificando o valor do bit endereçado pela instrução. Se o bit endereçado pela instrução estiver no estado lógico 1 a instrução retorna com o valor lógico falso e portanto não há continuidade lógica no trecho do *ladder*. Se o bit endereçado pela instrução estiver no estado lógico 0 a instrução retorna com o valor lógico verdadeiro e portanto há continuidade lógica no trecho do *ladder*. A representação gráfica desta instrução juntamente com a sua operação faz com que ela seja interpretada como um contato normalmente fechado de um relé. **Porém, convém ressaltar que, apesar da analogia, é uma instrução lógica e não um contato de circuito elétrico.**

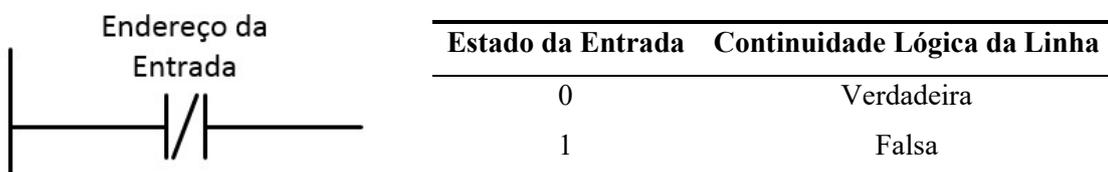


Figura 3 – Instrução NF.

Instrução de Saída – Bobina Energizada (OTE) (Output Terminal Energize)

A CPU executa esta instrução verificando se há ou não continuidade lógica na linha que antecede a instrução. Caso haja continuidade lógica da linha, o bit endereçado pela instrução será colocado no estado lógico 1. Se não houver continuidade na linha o bit endereçado pela instrução será colocado no estado lógico 0.

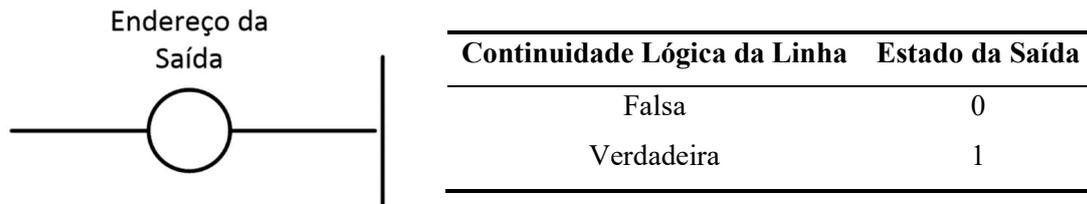


Figura 4 – Bobina Energizada.

2. OBJETIVOS

Essa atividade experimental tem como objetivo:

- 1) Introduzir os Controladores Lógicos Programáveis (CLPs) e apresentar a motivação histórica por trás do seu desenvolvimento;
- 2) Apresentar a arquitetura básica dos CLPs;
- 3) Introduzir a programação de CLPs usando a linguagem *ladder*;
- 4) Apresentar as principais instruções de *bit* da linguagem *ladder*.

Após conclusão deste relatório, o aluno deverá ser capaz de:

- 1) Compreender as vantagens que o uso do CLP trouxe para a indústria;
- 2) Compreender os princípios da programação de CLPs com a linguagem *ladder*;
- 3) Analisar e compreender alguns requisitos de projetos de automação;
- 4) Configurar o ambiente de programação com o *software* “*Connected Components Workbench*”;
- 5) Aplicar as principais instruções de *bit* da linguagem *ladder* na resolução de problemas de automação.

3. MATERIAIS

Os equipamentos usados nesta atividade experimental estão listados a seguir:

Kit Didático – CLP

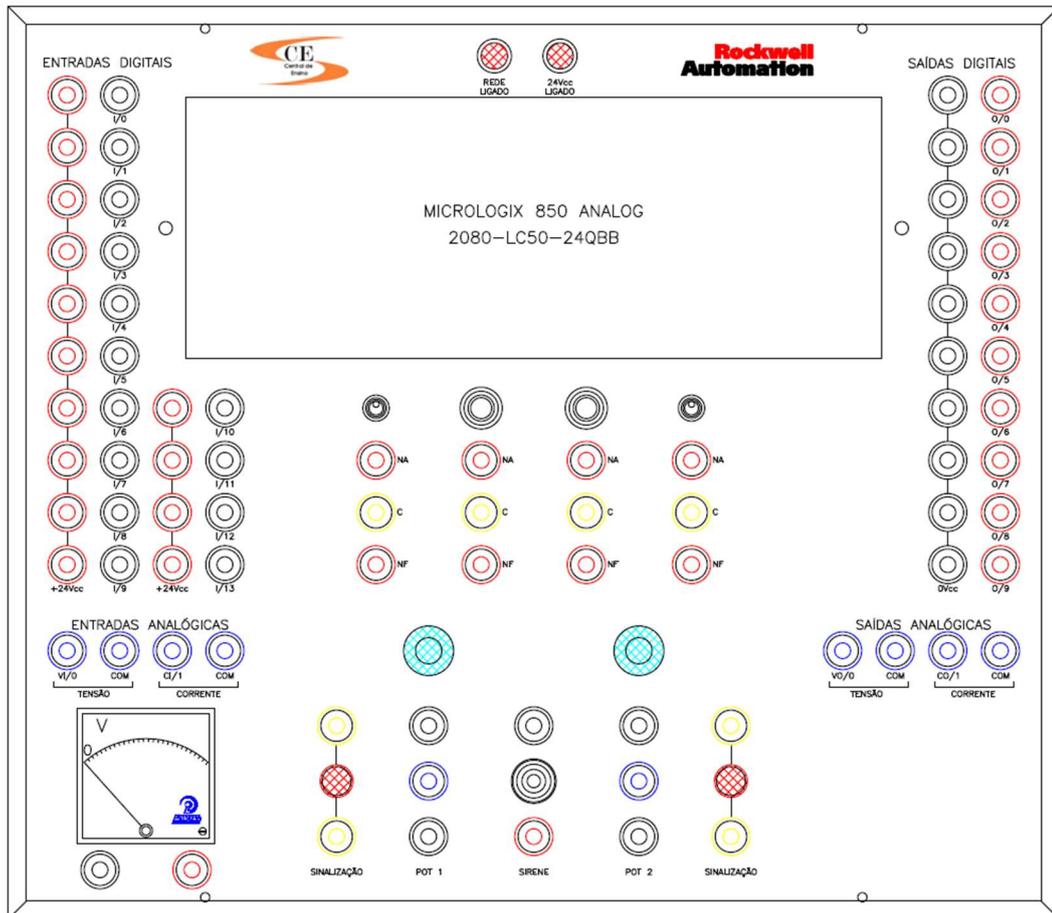


Figura 5 – Equipamentos utilizados na atividade experimental.

4. ROTEIRO DE LABORATÓRIO

4.1. Familiarização com o kit didático

Nesta seção é explicado como os elementos do kit didático devem ser utilizados. Recomenda-se ler com atenção as informações abaixo antes de passar para as atividades práticas.

Atenção:

Você **pode** fazer conexões ou modificações com o kit energizado, desde que:

- Faça uma conexão/modificação de cada vez;
- Não deixe as pontas dos cabos banana-banana conectados ao kit soltas.

Utilizando as entradas digitais:

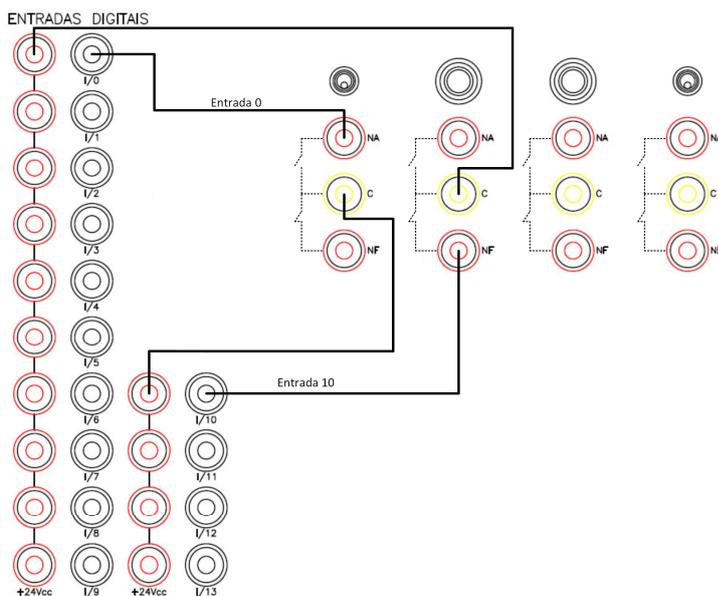


Figura 6 – Exemplo de como conectar as entradas digitais.

O kit didático permite o acesso a 14 entradas digitais, enumeradas de I/0 a I/13. Para utilizar qualquer uma delas, você deve fazer as conexões conforme demonstrado na figura ao lado. No exemplo, a entrada I/0 está vinculada com a chave no borne NA. Com a chave na posição representada, o nível lógico reconhecido pelo CLP é 0 (o circuito está aberto – não há tensão na entrada). Caso a chave seja acionada, o nível lógico passará a ser 1 (o circuito será fechado – haverá uma tensão de 24 V na entrada). A entrada I/10 está vinculada com o botão no borne NF. Nesse caso, com o botão liberado, o nível lógico reconhecido pelo CLP é 1 (o circuito está fechado – 24 V na entrada). Caso o botão seja pressionado, o nível lógico passará a ser 0 (o circuito será aberto – sem tensão na entrada)

Utilizando as saídas digitais:

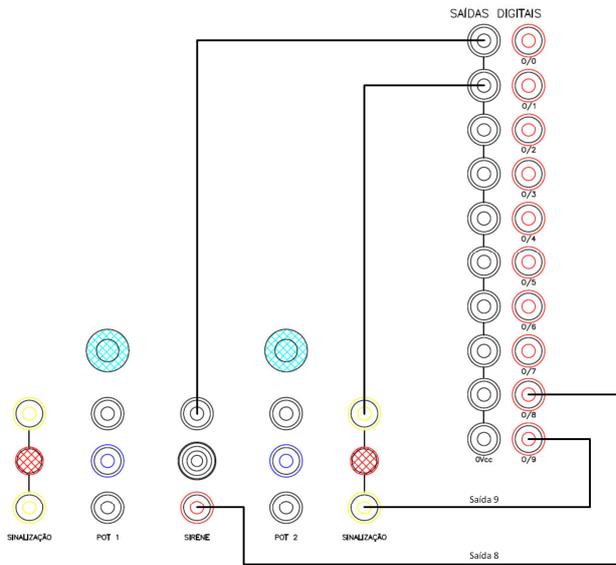


Figura 7 – Exemplo de como conectar as saídas digitais.

As saídas digitais do kit serão utilizadas para fazer o acionamento dos elementos de sinalização (lâmpadas e a sirene). A conexão deve ser feita conforme indicado na figura ao lado. **Embora não esteja explicitamente representado no painel (ambos bornes são amarelos), as lâmpadas também possuem polaridade. Assim, caso você tenha dado o comando para ligar alguma delas e isso não ocorreu, tente inverter as conexões nos bornes.**

4.2. Configuração do ambiente de programação

Esta seção apresenta os procedimentos de configuração do CLP instalado no kit didático no *Connected Components Workbench Standard Edition (CCW)*, que é um software gratuito desenvolvido pela Rockwell Automation e que permite a programação de CLPs da família Micro800. Siga os passos a seguir:

Passo 1) Abra o CCW e clique na opção “Novo” para abrir a configuração de um novo projeto

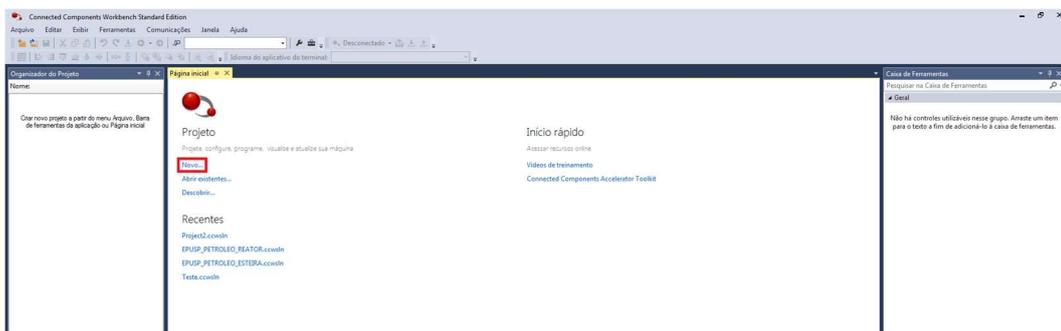


Figura 8 – Tela inicial do CCW.

Passo 2) Na janela “Novo projeto”, escolha o nome do projeto e clique em “OK”

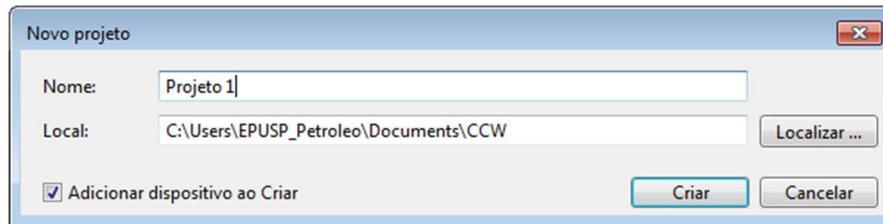


Figura 9 – Janela “Novo projeto”.

Passo 3) Na janela “Adicionar dispositivo”, clique em “Controladores”, depois em “Micro850” e escolha a opção 2080-LC50-24QBB, que é o controlador instalado no kit didático. Em seguida, clique em “Selecionar”:

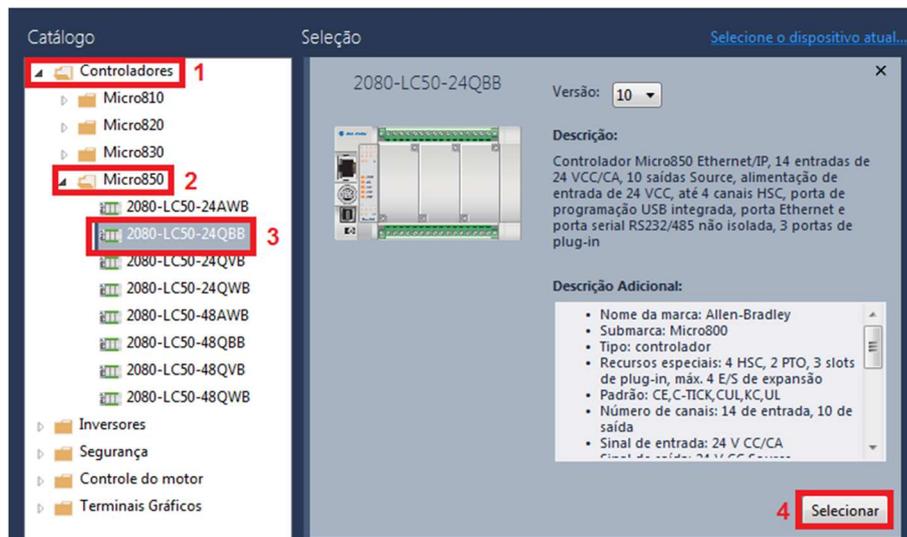


Figura 10 – Catálogo de dispositivos.

Passo 4) Na janela seguinte, clique em “Adicionar ao projeto” e aguarde a inicialização do projeto que você acabou de criar. Ao final, você se deparará com a tela abaixo. No canto superior esquerdo está o “Organizador do Projeto”, onde ficam estruturados todos os dispositivos e outros elementos que fazem parte do seu projeto. No caso, como dispositivo há somente o controlador Micro850. No canto inferior da tela, estão todas as opções de configuração disponíveis para este controlador.

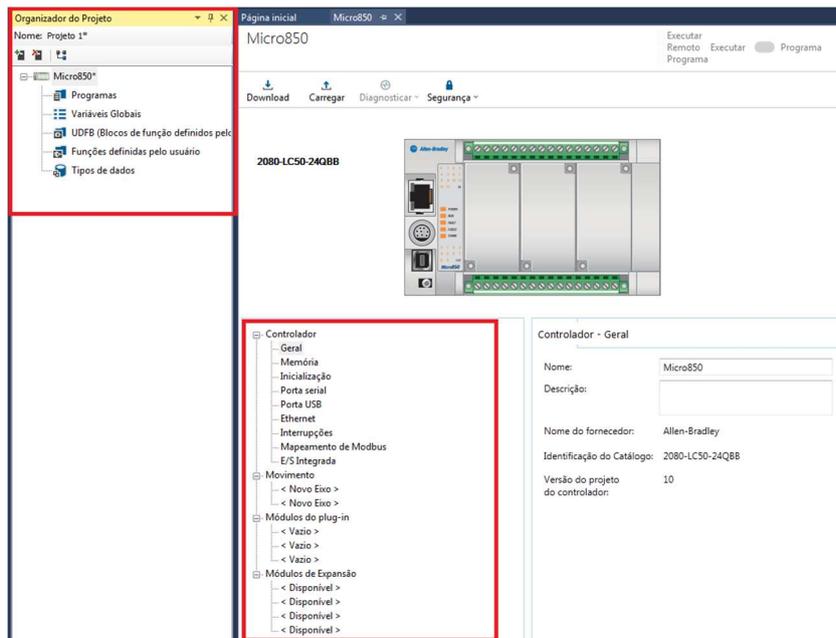


Figura 11 – Visualização do “Organizador do Projeto” (à esquerda) e das opções de configuração para o controlado (abaixo).

Passo 5) A última etapa é configurar o caminho de conexão com o CLP. Clique na caneta ao lado de “Caminho da Conexão” (1). Em seguida, clique em “Localizar”.

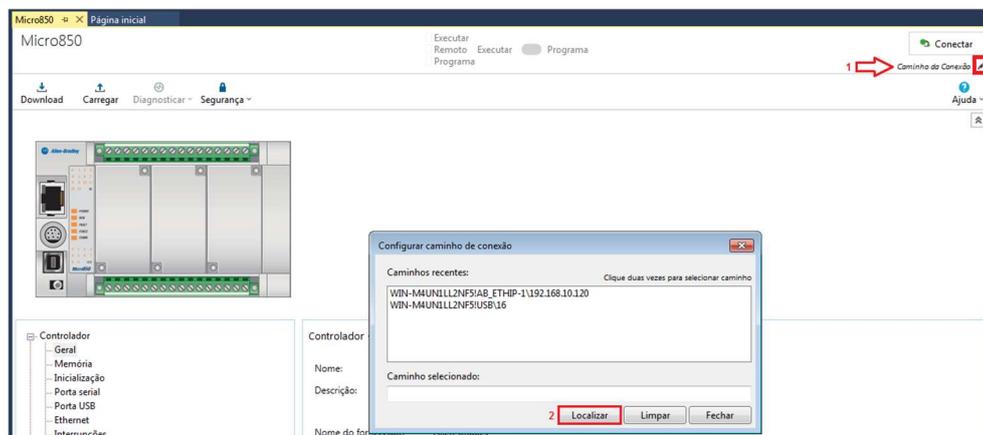


Figura 12 – Janela para configuração do caminho de conexão.

Passo 6) Na janela “Navegador da conexão”, clique no (+) ao lado da opção USB e selecione o controlador Micro850. Em seguida, clique em “OK” e em “Fechar”. A configuração do dispositivo no ambiente CCW está concluída.

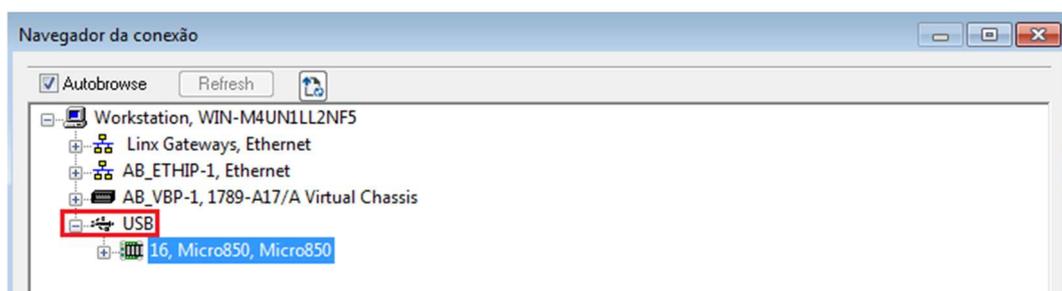


Figura 13 – Janela do “Navegador da conexão”.

4.3. Programa tutorial

Esta seção apresentará os procedimentos para a implementação de um programa simples em *ladder* usando o ambiente CCW, por meio de um **projeto guiado**. Siga atentamente os passos a seguir, pois eles introduzirão conceitos fundamentais para a resolução dos projetos propostos nas seções seguintes. Certifique-se que seu primeiro programa está funcionando antes de prosseguir para as próximas etapas da atividade experimental.

Entradas e saídas digitais

Passo 1) No “Organizador do Projeto”, clique com o botão direito em “Programas” e adicione um novo diagrama *ladder* (“Novo LD: diagrama de escada”)

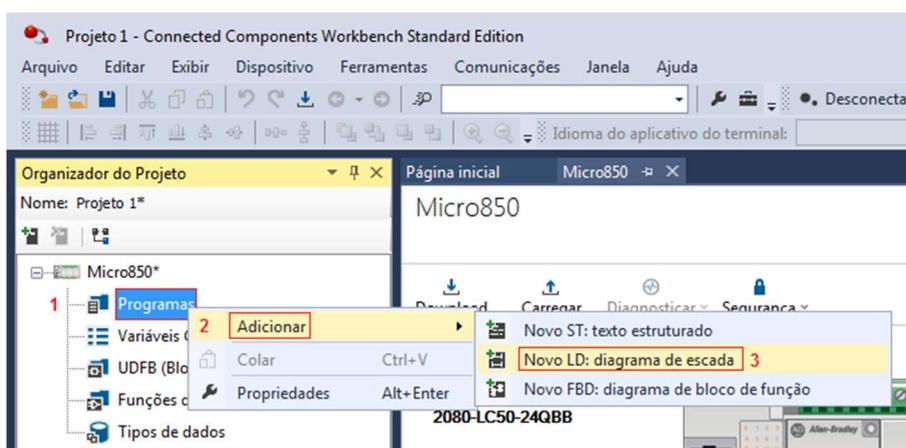


Figura 14 – Criação de um novo programa do tipo *ladder*.

Passo 2) Vamos fazer um sistema Liga/Desliga para uma das lâmpadas do painel. Para isso, clique no “Contato direto” (também chamado de NA ou XIC), na “Caixa de Ferramentas”, e arraste para o diagrama *ladder*, conforme a figura abaixo.

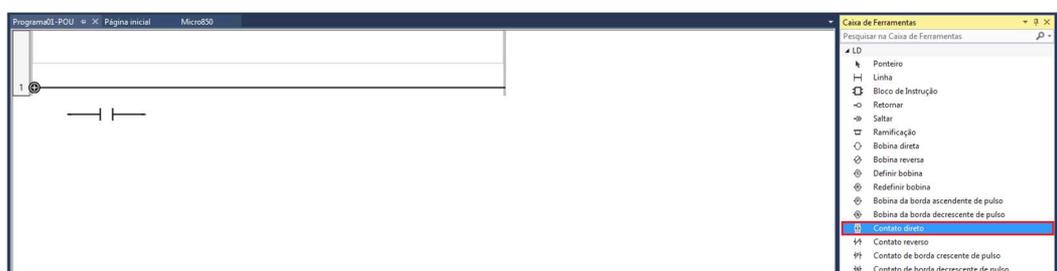


Figura 15 – Inclusão de um “Contato direto” (NA) na primeira linha do diagrama.

OBS: Caso a “Caixa de Ferramentas não esteja visível, clique na opção  localizada no menu superior (ou utilize as teclas de atalho Ctrl+Alt+X).

Passo 3) Após inserir o “Contato direto”, será aberta a janela do “Seletor de variáveis”. Clique na aba “E/S – Micro850”, onde estão listadas todas as entradas e saídas digitais configuradas no controlador. Navegue até encontrar a variável “_IO_EM_DI_00”, que é a entrada digital 00 (DI – Digital Input). É uma boa prática incluir um “Alias”, que neste caso nada mais é do que uma breve explicação a respeito do uso daquela entrada na lógica do seu programa. Escreva “Botão Liga” e clique em “Ok”

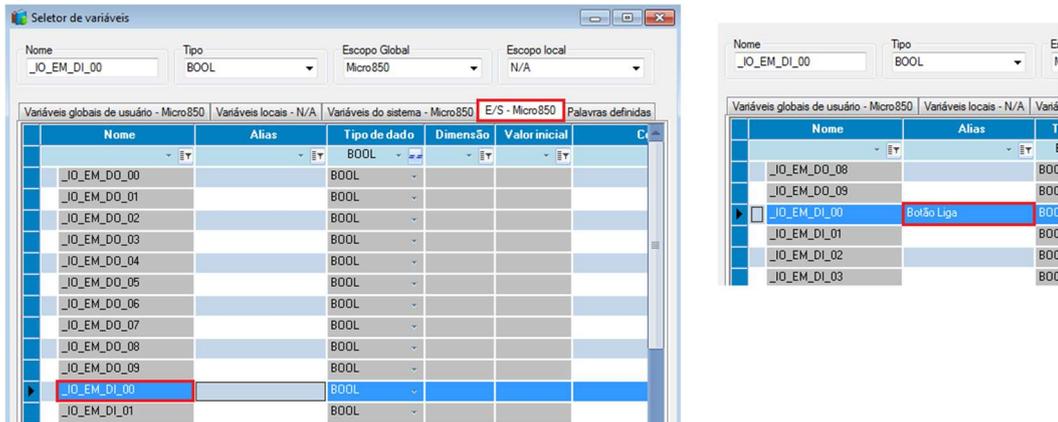


Figura 16 – Janela do seletor de variáveis: seleção da entrada digital 00 (à esquerda) e atribuição de um “Alias” para descrever sua função no programa (à direita).

Observe que a variável é do tipo BOOL, ou seja, pode assumir somente os estados 1 ou 0 (ligado ou desligado).

Passo 4) Ao lado do “Contato direto”, insira uma “Bobina direta” (também chamada de OTE). O procedimento é o mesmo que foi realizado no passo 1).

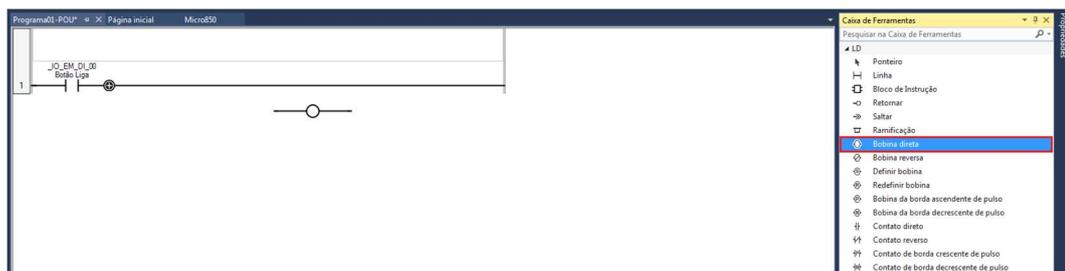


Figura 17 – Inclusão de uma “Bobina direta” (OTE) no diagrama.

Passo 5) Na janela do “Seletor de variáveis”, vá na aba “E/S – Micro 850” e selecione a saída digital _IO_EM_DO_00 (DO – Digital Output). Defina o “Alias” como “Lâmpada L1”.

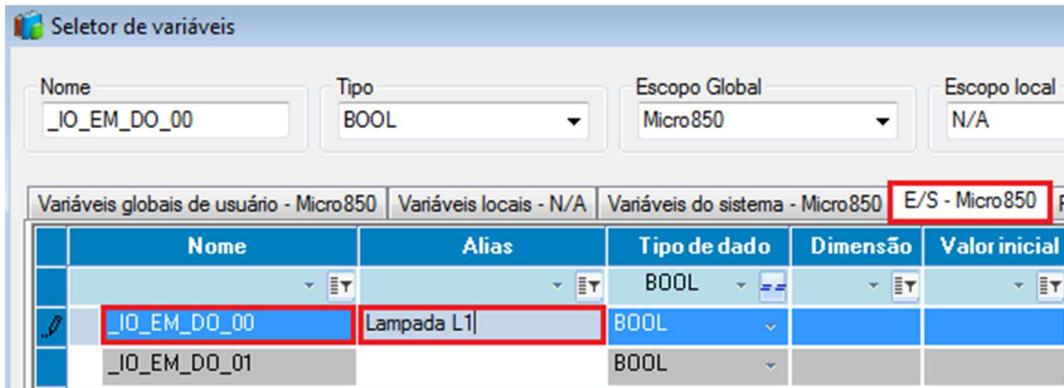


Figura 18 – Atribuição da saída digital 00 para a “Bobina direta” inserida no passo 4)

Passo 6) Vamos fazer o **contato de selo** para impedir o desligamento da lâmpada ao se liberar o “Botão Liga”. Para isso, insira uma “Ramificação” na linha, conforme mostrado a seguir:

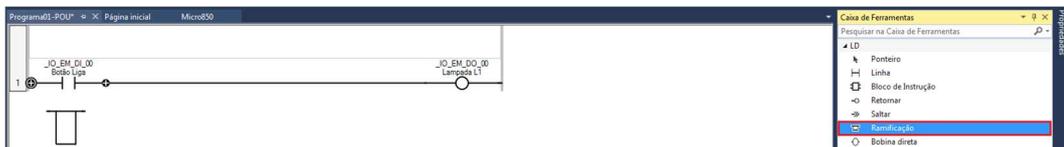


Figura 19 – Inclusão de uma “Ramificação” abaixo do “Botão Liga”.

Passo 7) Em seguida, insira um “Contato direto” na ramificação e referencie-o com a saída “Lâmpada L1”, conforme representado na figura abaixo:

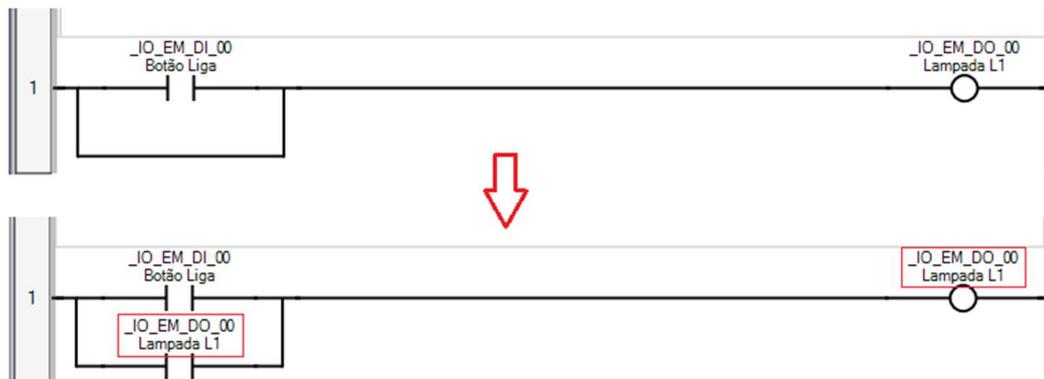


Figura 20 – Inclusão do contato de selo no diagrama.

Passo 8) Por fim, vamos inserir o “Botão desliga” no diagrama *ladder*. Utilize para isso um “Contato direto” e vincule-o com a entrada digital 01 (`_IO_EM_DI_01`). É uma boa prática inserir um comentário acima da linha explicando sua função dentro do programa. Para isso, dê um duplo clique na área acima da linha e complete com a frase destacada na figura a seguir.

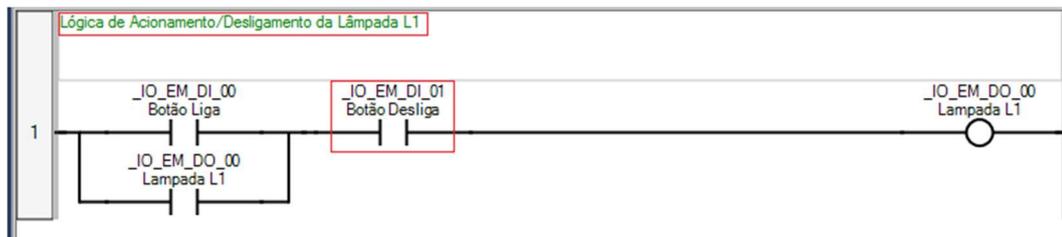


Figura 21 – Inclusão do “Botão Desliga” no diagrama.

Passo 9) A lógica do botão liga/desliga está pronta. Agora é necessário realizar as conexões no painel conforme ilustrado abaixo:

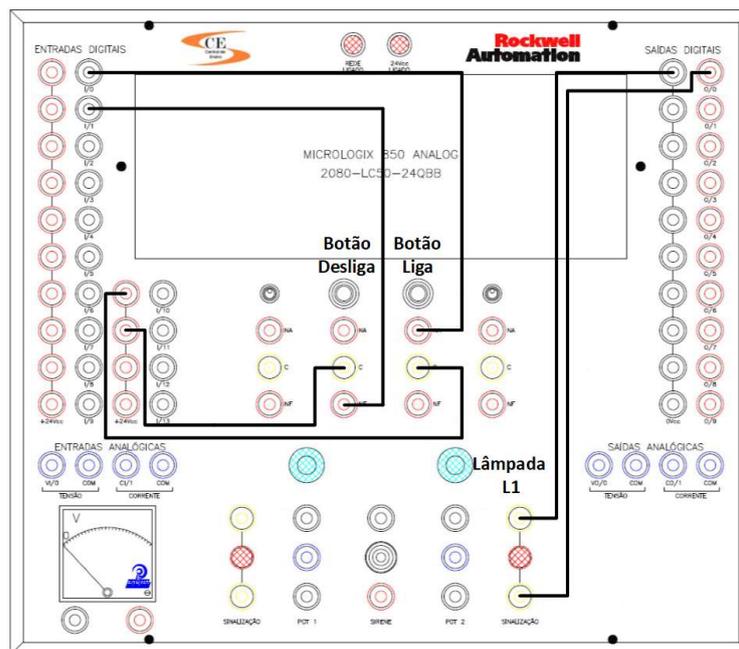


Figura 22 – Esquema das ligações no painel para o teste do programa.

Passo 10) Vamos testar o programa. Para isso, no “Organizador de Projeto” clique com o botão direito no seu programa e escolha a opção “Compilar”. Aguarde e verifique que o processo ocorre sem apresentar erros. Em seguida, clique com o botão direito novamente no seu programa e escolha a opção “Download” para descarrega-lo no controlador. Observe que **download significa enviar o programa do computador para o CLP.**

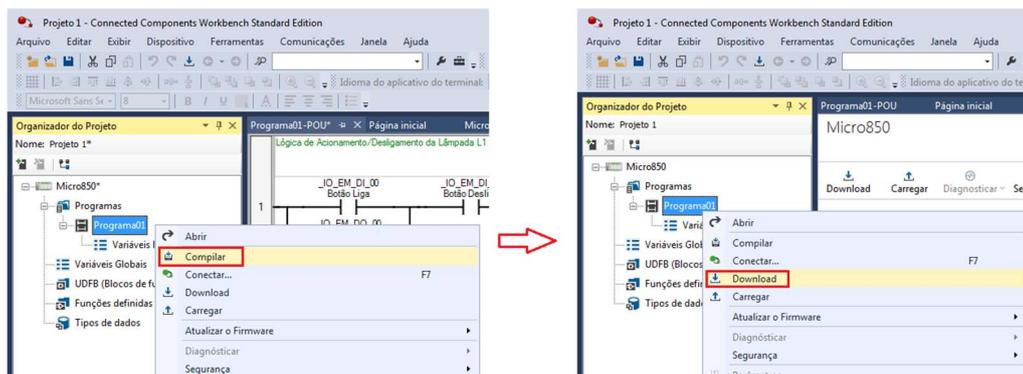


Figura 23 – Compilação do programa (à esquerda) e solicitação do “Download” (à direita).

Passo 11) Aparecerão duas mensagens: a primeira para confirmar se você deseja fazer o download, alertando que o processo sobrescreve o programa atual no controlador. Escolha “Download” e aguarde. Ao final do processo aparecerá a segunda mensagem perguntando se você deseja mudar o controlado para o modo execução remota. Escolha a opção “Yes”.

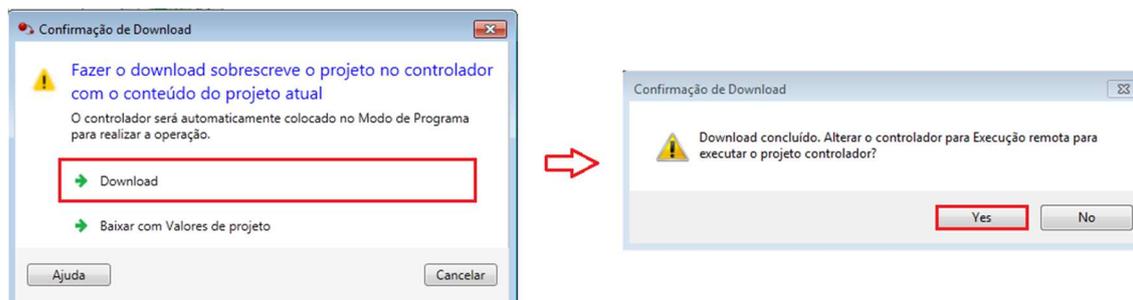


Figura 24 – Confirmação para a realização do “Download” (à esquerda) e confirmação para passar o controlador para o modo de “Execução Remota”.

Passo 12) Teste o funcionamento do programa e verifique que o estado é indicado no diagrama *ladder*. Em seguida, preencha a Tabela 1, considerando o valor 1 para estado ativado (permite a continuidade lógica da linha) e o valor 0 para estado não ativado

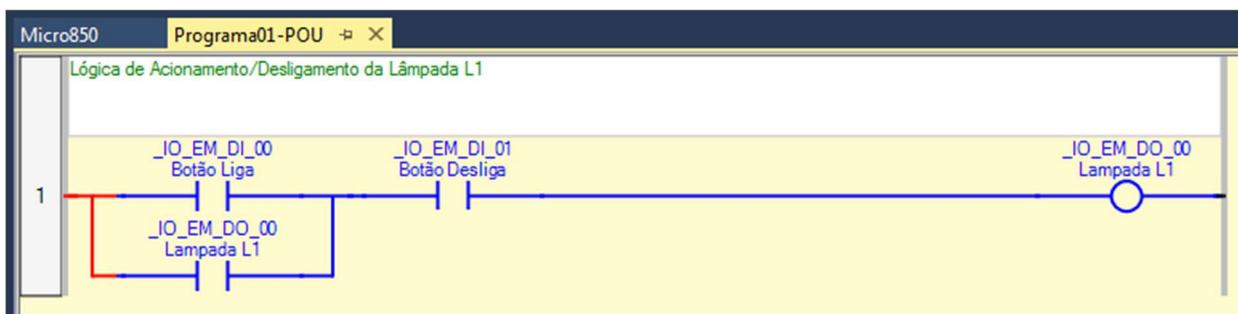


Figura 25 – Diagrama exibido no CCW após a realização do “Download”. Indica quais entradas e saídas estão acionadas e desligadas.

Tabela 1 – Estado do sistema em função das entradas.

Entrada	Instruções (<i>ladder</i>)		Saídas (<i>ladder</i>)	Lâmpadas
	Desliga DI_01	Liga DI_00	L1 DO_00	L1
Botão Liga Pressionado				
Botão Liga Pressionado e solto				
Botão Desliga pressionado				
Botão Desliga pressionado e solto				
Ambos pressionados				

OBS: Para modificar o programa, é necessário desconectar o CCW do controlador. Para isso,

clique em “Conectado”, no menu superior. Em geral, edição do programa enquanto conectado é possível. Entretanto, o CCW em sua versão gratuita não permite por limitações de licença.

4.4. Análise de lógica com intertravamento em *ladder*

Os intertravamentos são vínculos lógicos estabelecidos entre os equipamentos para que o sistema funcione de maneira correta e segura. Um exemplo simples de intertravamento ocorre nos elevadores, nos quais o movimento de subida ou descida só deve acontecer caso a porta esteja fechada.

Nesta etapa da experiência, você deve implementar e analisar o sistema com intertravamento descrito na Figura 26. Não esqueça de incluir a conexão da saída DO_00 para a lâmpada L2 no kit didático.

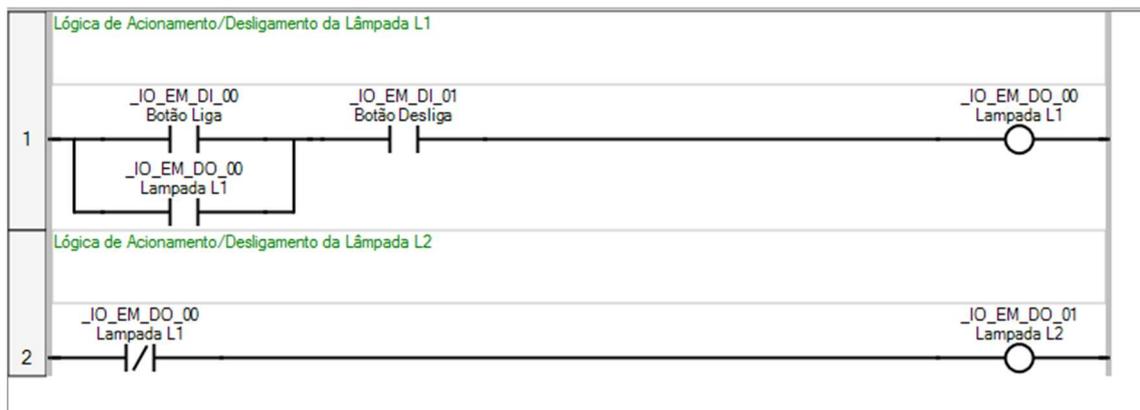


Figura 26 – Diagrama *ladder* com intertravamentos a ser implementado.

Preencha a tabela de estados a seguir, indicando 1 para ativado e 0 para não ativado.

Tabela 2 – Estado do sistema com intertravamento em função das entradas.

Entrada	Instruções (<i>ladder</i>)		Saídas (<i>ladder</i>)		Lâmpadas	
	Desliga DI_01	Liga DI_00	L1 DO_00	L2 DO_01	L1	L2
Botão Liga Pressionado						
Botão Liga Pressionado e solto						
Botão Desliga pressionado						
Botão Desliga pressionado e solto						
Ambos pressionados						

4.5. Alteração na lógica com intertravamento

Nesta etapa da atividade experimental, você deve modificar **apenas o ladder** para que o sistema anterior passe a operar com as funções das lâmpadas L1 e L2 trocadas.

Em seguida, registre as diferenças observadas no comportamento do sistema na Tabela 3.

Tabela 3 – Estado do sistema com intertravamento modificado.

Entrada	Instruções (<i>ladder</i>)		Saídas (<i>ladder</i>)		Lâmpadas	
	Desliga DI_01	Liga DI_00	L1 DO_00	L2 DO_01	L1	L2
Botão Liga Pressionado						
Botão Liga Pressionado e solto						
Botão Desliga pressionado						
Botão Desliga pressionado e solto						
Ambos pressionados						

4.6. Projeto – Automação de um sistema de distribuição de água

Atenção:

Você deve apresentar todo o raciocínio e as considerações empregadas para resolver os exercícios propostos a seguir no relatório. Além disso, registre o diagrama *ladder* da solução por meio de desenhos ou capturas de tela. Demonstre o funcionamento do sistema para o professor.

Projete e simule a automação de um reservatório de água, conforme figura abaixo. A bomba B1 deve ser ligada para manter a água do reservatório de distribuição entre os níveis indicados pelos sensores $N_{\min 2}$ e $N_{\max 2}$, que acusam a presença de água no nível determinado. No painel, utilize uma lâmpada para indicar quando o motor está ligado ou desligado. Observe que B1 não pode ser acionada caso o nível do reservatório principal esteja abaixo de $N_{\min 1}$. Nesse caso, deverá ser acionado um alarme de falta de água (sirene). Para simular os sensores de nível, utilize as chaves disponíveis no painel e um dos botões “NA”. Assuma que os sensores de nível são acionados, ou seja, retornam **nível lógico 1** ao CLP, na **presença de água**.

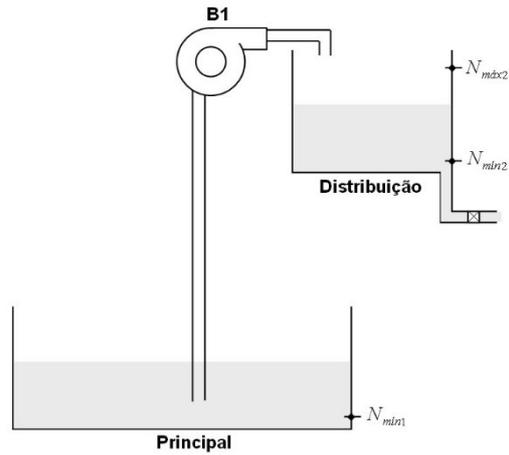


Figura 27 – Sistema de distribuição de água.

Observe a operação do sistema e preencha a tabela verdade fornecida a seguir. Considere o valor 0 como desligado e o valor 1 como ligado. Indique com um X os estados impossíveis de serem alcançados na prática.

Tabela 4 – Estado do sistema em função das entradas.

N_{min1}	N_{min1}	N_{min1}	Bomba	Alarme
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

5. RELATÓRIO

No relatório deverá constar o roteiro da atividade experimental preenchido com todos os resultados coletados e as respostas dos itens solicitados a seguir. Busque sempre deixar claro o raciocínio adotado e estabelecer um paralelo entre a teoria vista em sala de aula e os resultados obtidos no laboratório.

Parte 1 – Com base nos resultados obtidos nas seções 4.3, 4.4 e 4.5 da atividade experimental, explique o funcionamento de cada um dos sistemas implementados e faça o que se pede nas alíneas a seguir:

- a) Na seção 4.3, você implementou sistema Liga/Desliga para uma das lâmpadas do kit didático. Nesse caso, o botão Liga era NA e o botão Desliga era NF. Assumindo que as características de funcionamento devem ser mantidas, como deveria ser o diagrama *ladder* do sistema caso os botões fossem do tipo NF?
- b) Dê pelo menos dois exemplos de intertravamentos de segurança que devem ser incluídos em sistemas reais.
- c) Imagine que o sistema da seção 4.4 tenha sido implementado com contadores. Quais dificuldades seriam encontradas para realizar a modificação solicitada na seção 4.5, nesse caso? Comente as vantagens que a introdução dos CLPs na indústria trouxe em relação à realização de modificações/manutenções nos sistemas.

Parte 2 – Apresente todo o raciocínio e as considerações empregadas para resolver o projeto **Automação de um sistema de distribuição de água**, proposto na seção 4.6 da atividade experimental. Não se esqueça de incluir o diagrama *ladder* da solução.

5.1. Projetos adicionais

Atenção:

Você deve apresentar todo o raciocínio e as considerações empregadas para resolver os projetos adicionais no relatório. Registre o diagrama *ladder* da solução por meio de desenhos. **Não é necessário implementar os projetos abaixo no CLP.**

Projeto 1 - Automatizar o sistema de distribuição de água de um edifício com reservatório secundário. A bomba B1 deve ser preferencialmente ligada para regular o nível de água no reservatório de distribuição de modo a deixá-lo entre $N_{\min 2}$ e $N_{\max 2}$. B1 só deve ser ligada se o sensor $N_{\min 1}$ não acusar nível baixo no reservatório. Caso o nível da distribuição esteja baixo, mas B1 estiver impossibilitada de ligar, B2 deverá ser acionada juntamente com um alarme sonoro para avisar da falta de água no reservatório principal. B2 também não poderá ser acionada se o nível do reservatório secundário estiver baixo. B3 é responsável por manter o nível de água no reservatório secundário entre os níveis $N_{\min 3}$ e $N_{\max 3}$. B3 só pode ser ligado caso o reservatório principal não esteja com o nível baixo. Desenhe o *ladder* do sistema.

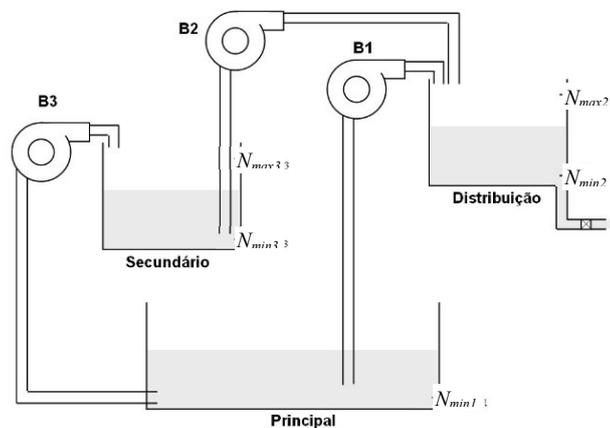


Figura 28 – Sistema de distribuição com três reservatórios.

Projeto 2 - Deve-se comandar um motor por meio de um programa *ladder* para abertura e fechamento de um portão a partir de dois postos de comando. Considere os seguintes requisitos de funcionamento:

1. Quando o portão estiver fechado e houver um acionamento para a sua abertura (comando “Abrir”), ele deve abrir totalmente e o motor deve desligar quando isso ocorrer (ou seja, quando atingir o sensor de fim de curso “Aberto”);
2. Da mesma forma, quando o portão estiver aberto e houver um acionamento para o seu fechamento, ele deve fechar totalmente e o motor deve desligar quando isso ocorrer; (ou seja, quando atingir o sensor de fim de curso “Fechado”);
3. Deve ser incluído um botão “Emergência” para interromper o funcionamento do sistema a qualquer momento nos dois postos de comando.
4. Quando a porta estiver abrindo, deve acender uma luz verde nos pontos de comando;

5. Quando a porta estiver fechando deve acender uma luz amarela nos pontos de comando;
6. O motor não deve ser comandado no sentido de abrir e fechar simultaneamente em hipótese alguma, pois isso causaria um curto-circuito.

Tabela 5 – Lista de entradas e saídas do sistema.

Entradas	Saídas
Botão Abrir (Posto de Comando 1)	Motor Sentido Direto (Abrir)
Botão Abrir (Posto de Comando 2)	Motor Sentido Reverso (Fechar)
Botão Fechar (Posto de Comando 1)	Sinalização Verde (Posto de Comando 1)
Botão Fechar (Posto de Comando 2)	Sinalização Verde (Posto de Comando 2)
Botão Emergência (Posto de Comando 1)	Sinalização Amarela (Posto de Comando 1)
Botão Emergência (Posto de Comando 2)	Sinalização Amarela (Posto de Comando 2)
Sensor de Fim de Curso Aberto	
Sensor de Fim de Curso Fechado	

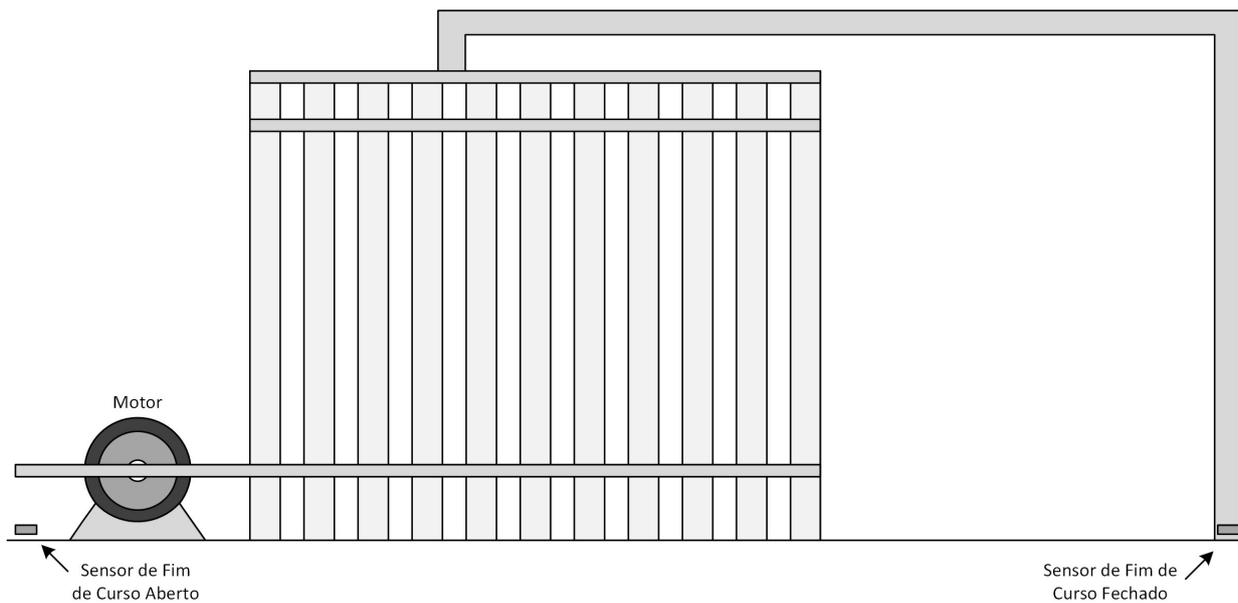


Figura 29 – Portão a ser comandado.