



FUMEP – Fundação Municipal de Ensino de Piracicaba

EEP – Escola de Engenharia de Piracicaba

COTIP – Colégio Técnico Industrial de Piracicaba

CONTROLADORES LÓGICO PROGRAMÁVEIS - LADDER

Prof. Msc. Marcelo Eurípedes da Silva

Piracicaba, 15 de Fevereiro de 2007

1 - Introdução

O objetivo desta apostila é o de apresentar a linguagem de programação de Controladores Lógico Programáveis (CLPs) denominada de diagrama de contatos, ou Ladder. A palavra “Ladder” em inglês significa “escada”, nome este dado por causa da similaridade da linguagem com o objeto de uso diário. Esta similaridade será observada posteriormente, no decorrer do curso.

A simbologia na linguagem de programação Ladder segue a padrões e normas internacionais, apesar de há uma pequena variação em alguns símbolos dentre os diferentes fabricantes. Mais uma vez deve-se citar que o aluno não deve ficar preocupado com este detalhe, pois o que importa realmente, é o raciocínio lógico que leva a programação, e não o símbolo propriamente dito.

Nos próximos parágrafos tem-se a definição de CLP, mostrando um histórico da tecnologia. A partir do capítulo 2 pode-se encontrar a parte prática do curso.

1.1 - Definição dos CLPS

Os CLPs ou Controladores Lógico Programáveis podem ser definidos, segundo a norma ABNT, como um equipamento eletrônico-digital compatível com aplicações industriais.

Os CPLs também são conhecidos como PLCs, do inglês: Programmable Logic Controller.

O primeiro CLP data de 1968 na divisão de hidramáticos da General Motors.

Surgiu como evolução aos antigos painéis elétricos, cuja lógica fixa tornava impraticável qualquer mudança extra do processo.

A tecnologia dos CLPs só foi possível com o advento dos chamados Circuitos Integrados e da evolução da lógica digital.

Trouxe consigo as principais vantagens:

- a) fácil diagnóstico durante o projeto
- b) economia de espaço devido ao seu tamanho reduzido
- c) não produzem faíscas
- d) podem ser programados sem interromper o processo produtivo
- e) possibilidade de criar um banco de armazenamento de programas
- f) baixo consumo de energia
- g) necessita de uma reduzida equipe de manutenção

- h) tem a flexibilidade para expansão do número de entradas e saídas
- i) capacidade de comunicação com diversos outros equipamentos, entre outras

1.2 - Histórico da Tecnologia

Historicamente os CLPs podem ser classificados nas seguintes categorias:

1ª GERAÇÃO: Programação em Assembly. Era necessário conhecer o hardware do equipamento, ou seja, a eletrônica do projeto do CLP.

2ª GERAÇÃO: Apareceram as linguagens de programação de nível médio. Foi desenvolvido o “Programa monitor” que transformava para linguagem de máquina o programa inserido pelo usuário.

3ª GERAÇÃO: Os CLPs passam a ter uma entrada de programação que era feita através de um teclado, ou programador portátil, conectado ao mesmo.

4ª GERAÇÃO: É introduzida uma entrada para comunicação serial, e a programação passa a ser feita através de micro-computadores. Com este advento surgiu a possibilidade de testar o programa antes do mesmo ser transferido ao módulo do CLP, propriamente dito.

5ª GERAÇÃO: Os CLPs de quinta geração vem com padrões de protocolo de comunicação para facilitar a interface com equipamentos de outros fabricantes, e também com Sistemas Supervisórios e Redes Internas de comunicação.

1.3 - Princípio de Funcionamento

Conforme a Figura 1.1 abaixo, o CLP funciona de forma seqüencial, fazendo um ciclo de varredura em algumas etapas. É importante observar que quando cada etapa do ciclo é executada, as outras etapas ficam inativas. O tempo total para realizar o ciclo é denominado **CLOCK**. Isso justifica a exigência de processadores com velocidades cada vez mais altas.

Início: Verifica o funcionamento da C.P.U, memórias, circuitos auxiliares, estado das chaves, existência de um programa de usuário, emite aviso de erro em caso de falha. Desativa todas as saídas.



Figura 1.1 – Ciclo de Varredura de um CLP

Verifica o estado das entradas: Lê cada uma das entradas, verificando se houve acionamento. O processo é chamado de ciclo de varredura.

Compara com o programa do usuário: Através das instruções do usuário sobre qual ação tomar em caso de acionamento das entradas o CLP atualiza a memória imagem das saídas.

Atualiza as saídas: As saídas são acionadas ou desativadas conforme a determinação da CPU. Um novo ciclo é iniciado.

1.4 - Estrutura Básica de um CLP

Fonte de alimentação: Converte a tensão da rede de 110 ou 220 VCA em +5VCC, +12VCC ou +24VCC para alimentar os circuitos eletrônicos, as entradas e as saídas.

Unidade de processamento: Também conhecida por CPU, é composta por microcontroladores ou microprocessadores (Intel 80xx, motorola 68xx, PIC 16xx). Endereçamento de memória de até 1Mega Byte, velocidades de clock de 4 a 30 MHz, manipulação de dados decimais, octais e hexadecimais.

Bateria: Utilizada para manter o circuito do relógio em tempo real. Normalmente são utilizadas baterias recarregáveis do tipo Ni - Ca.

Memória do programa supervisor: O programa supervisor é responsável pelo gerenciamento de todas as atividades do CLP. Não pode ser modificado pelo usuário e fica normalmente em memórias do tipo PROM, EPROM, EEPROM.

Memória do usuário: Espaço reservado ao programa do usuário. Constituída por memórias do tipo RAM, EEPROM ou FLASH-EPROM. Também pode-se utilizar **cartuchos de memória**, para proporcionar agilidade e flexibilidade.

Memória de dados: Armazena valores do programa do usuário, tais como valores de temporizadores, contadores, códigos de erros, senhas, etc. Nesta região se encontra também a **memória imagem das entradas – a saídas**. Esta funciona como uma tabela virtual onde a CPU busca informações para o processo decisório.

Os circuitos auxiliares atuam em caso de falha do CLP, são:

POWER ON RESET: desliga todas as saídas assim que o equipamento é ligado, isso evita que possíveis danos venham a acontecer.

POWER DOWN: monitora a tensão de alimentação salvando o conteúdo das memórias antes que alguma queda de energia possa acontecer.

WATCH DOG TIMER: o cão de guarda deve ser acionado em intervalos periódicos, isso evita que o programa entre em “loop”.

1.5 - Classificação dos CLPs

Os CLPs podem ser classificados segundo a sua capacidade:

Nano e micro CLPs: possuem até 16 entradas e a saídas. Normalmente são compostos por um único módulo com capacidade de memória máxima de 512 passos.

CLPs de médio porte: capacidade de entrada e saída em até 256 pontos, digitais e analógicas. Permitem até 2048 passos de memória.

CLPs de grande porte: construção modular com CPU principal e auxiliares. Módulos de entrada e saída digitais e analógicas, módulos especializados, módulos para redes locais. Permitem a utilização de até 4096 pontos. A memória pode ser otimizada para o tamanho requerido pelo usuário.

1.6 - Tipos de Variáveis

Durante o decorrer do curso serão utilizadas variáveis discretas e analógicas, pois esta mixagem é permitida neste tipo de linguagem. As variáveis *analógicas* são aquelas que variam continuamente com o tempo, conforme mostra a figura 1.1(a). Elas são comumente encontradas em processos químicos advindas de sensores de pressão, temperatura e outras variáveis físicas. As variáveis *discretas*, ou digitais, são aquelas que variam discretamente com o tempo, como pode ser visto na figura 1.1(b).

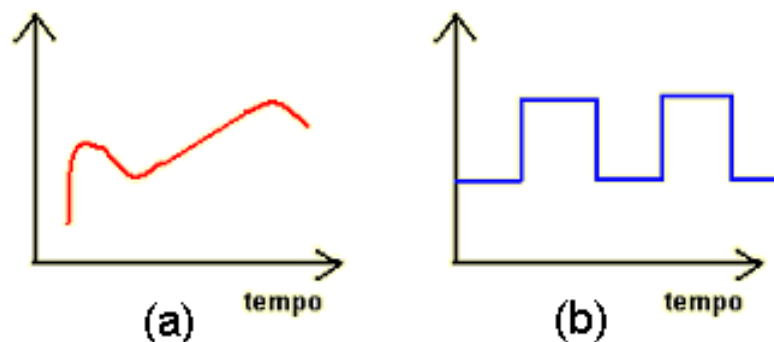


Figura 1.2 – Variáveis analógicas e digitais

Alguns tópicos do controle discreto, ou a automação com variáveis discretas, já foi estudado pelo aluno na disciplina de comandos elétricos. Esta é fundamental e básica, pois a finalidade da automação de qualquer sistema está no acionamento de atuadores, que irão exercer um trabalho físico no sistema controlado, evitando assim a intervenção humana.

As variáveis controladas pelo CLP podem ser divididas em entradas, advindas dos sensores e saídas, correspondendo aos atuadores. Alguns exemplos são mostrados nos próximos parágrafos.

1.7 - Entradas discretas

São aquelas que fornecem apenas um pulso ao controlador, ou seja, elas têm apenas um estado ligado ou desligado, nível alto ou nível baixo, remontando a álgebra booleana que trabalha com uns e zeros. Alguns exemplos são mostrados na figura 1.2, dentre elas: as botoeiras (1.2a), válvulas eletro-pneumáticas (1.2b), os pressostatos (1.2c) e os termostatos (1.2d).

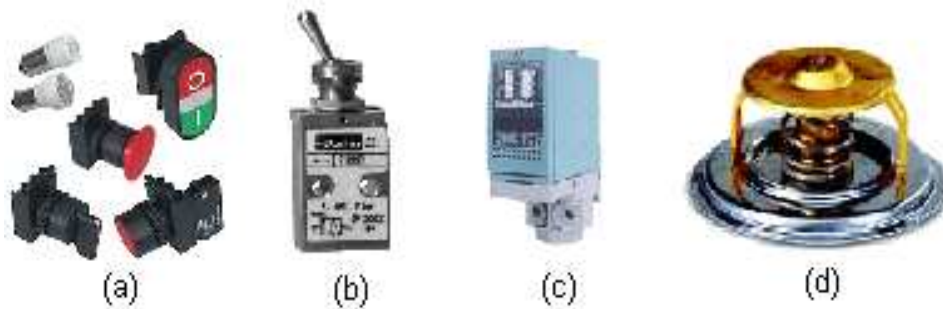


Figura 1.2 – Entradas discretas

1.8 - Entradas analógicas:

Como o próprio nome já diz, elas medem as grandezas de forma analógica. Para trabalhar com este tipo de entrada os controladores tem conversores analógico-digitais (A/D). Atualmente no mercado os conversores de 10 bits são os mais populares. As principais medidas feitas de forma analógica são a temperatura e pressão. Na figura 1.3 mostram-se, como exemplo, sensores de pressão ou termopares.

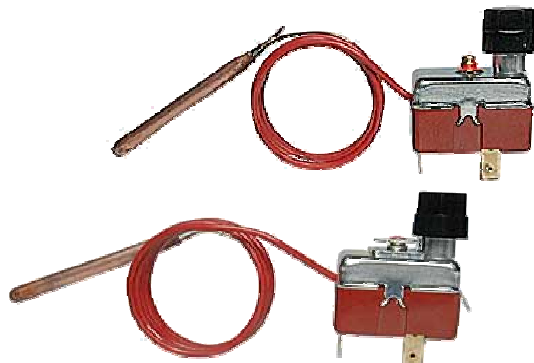


Figura 1.3 – Exemplos de entradas analógicas – Termopares

1.9 - Saídas discretas

São aquelas que exigem do controlador apenas um pulso que determinará o seu acionamento ou desacionamento. Como exemplo têm-se elementos mostrados na figura 1.4: Contatores (1.4a) que acionam os Motores de Indução (1.4b) e as Válvulas Eletropneumáticas (1.4c).

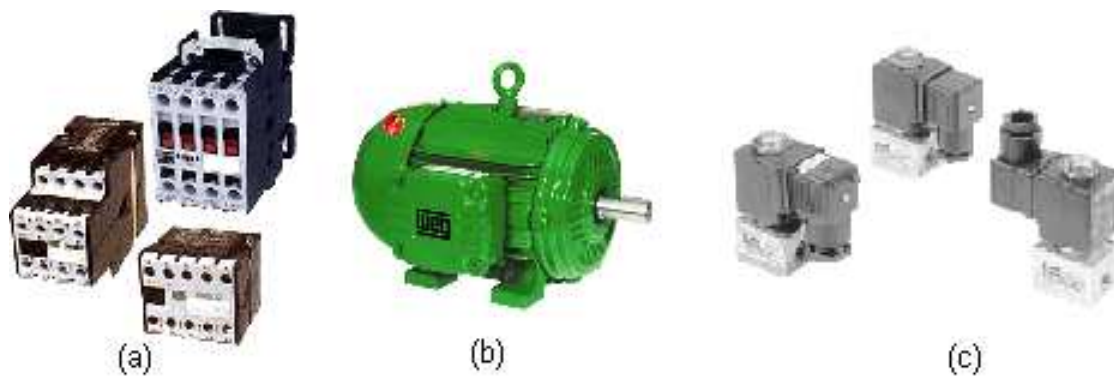


Figura 1.4 – Exemplos de saídas discretas

1.10 - Saídas analógicas

Como dito anteriormente, de forma similar o controlador necessita de um conversor digital para analógico (D/A), para trabalhar com este tipo de saída. Os exemplos mais comuns são: válvula proporcional, acionamento de motores DC, displays gráficos, entre outros.

2 – Elementos Básicos de um programa em Ladder

A linguagem Ladder foi a primeira que surgiu na programação dos Controladores Lógico Programáveis (CLPs), pois sua funcionalidade procurava imitar os antigos diagramas elétricos, utilizados pelos Técnicos e Engenheiros da época. O objetivo era o de evitar uma quebra de paradigmas muito grande, permitindo assim a melhor aceitação do produto no mercado.

O diagrama de contatos (Ladder) consiste em um desenho formado por duas linhas verticais, que representam os pólos positivo e negativo de uma bateria, ou fonte de alimentação genérica. Entre as duas linhas verticais são desenhados ramos horizontais que possuem chaves. Estas podem ser normalmente abertas, ou fechadas e representam os estados das entradas do CLP. Dessa forma fica muito fácil passar um diagrama elétrico para linguagem Ladder. Basta transformar as colunas em linhas, como se mostra nas figuras 2.1 e 2.2, para o caso de uma simples partida direta.

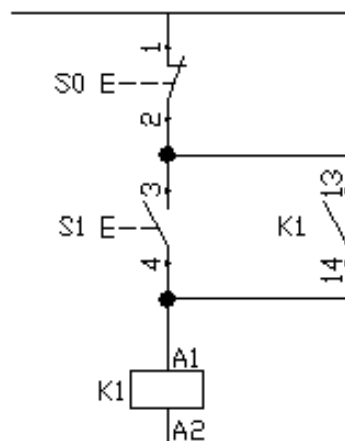


Figura 2.1 – Diagrama elétrico de uma partida direta

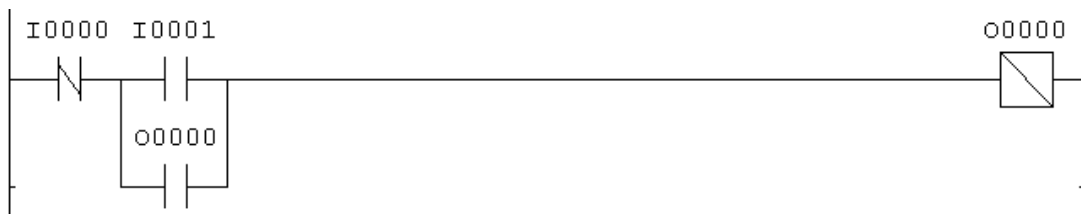


Figura 2.2 – Diagrama elétrico de uma partida direta

Não se deve esquecer de ligar as botoeiras e contatores, que são os elementos de comando, externamente ao CLP. Para o caso deste comando as ligações elétricas são mostradas na figura 2.3. É importante observar que o relé foi colocado para permitir a existência de dois circuitos diferentes, o de comando composto por uma tensão contínua de 24 V, e o circuito de potência, composto por uma tensão alternada de 220 V.

Ainda no CLP a letra "I" significa entrada (Input) e a letra "O" significa saída (Output).

Deve-se lembrar sempre que em painéis elétricos o CLP está inserido na parte de comando do mesmo.

Deve-se lembrar sempre que em painéis elétricos o CLP está inserido na parte de comando do mesmo.

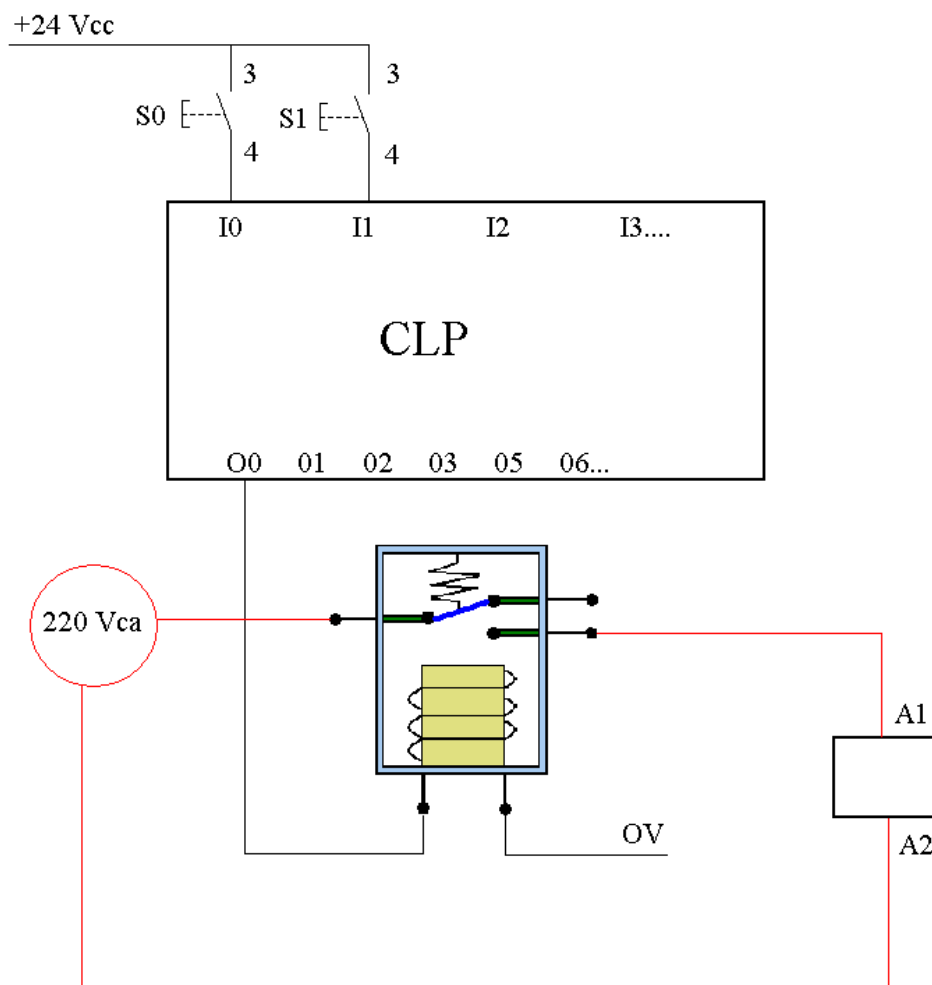


Figura 2.3 – Exemplo de ligação para acionamento de um contator, como no caso do comando direto de um motor de indução trifásico

O mesmo procedimento de conversão pode ser feito com para uma partida de motores com reversão, como mostram as figuras 2.4 e 2.5 a seguir.

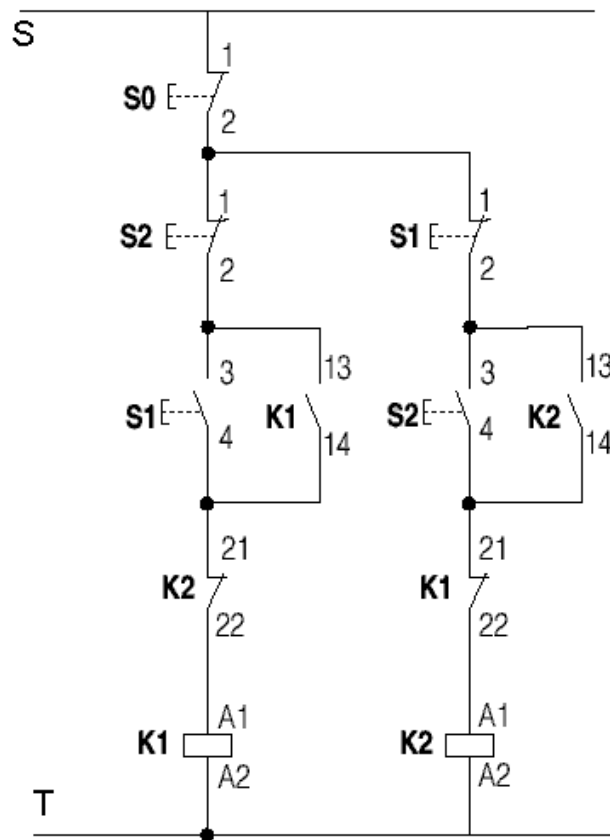


Figura 2.4 – Diagrama de comando para uma partida com reversão

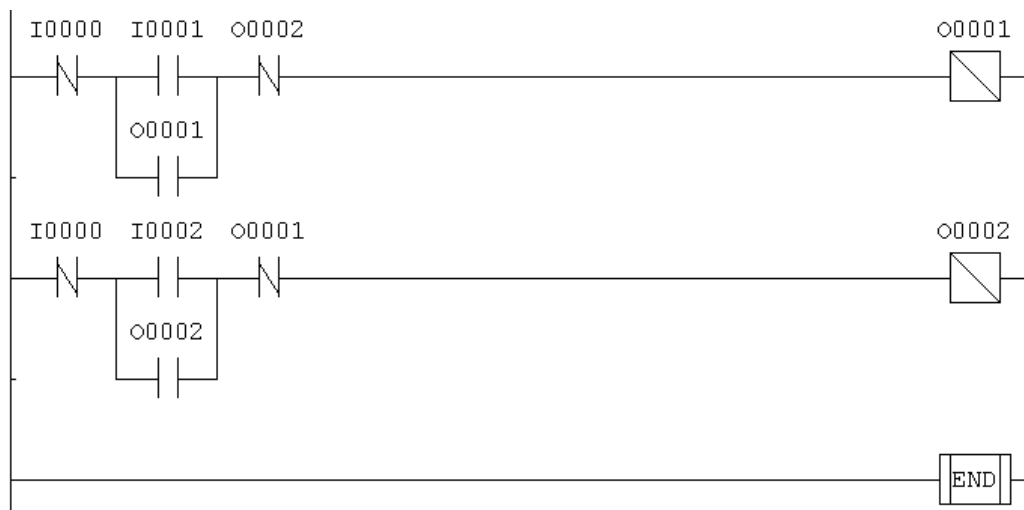
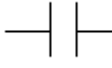
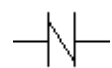
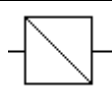


Figura 2.5 – Programa em Ladder para uma partida com reversão

Na figuras 2.4 e 2.5 podem-se observar os elementos básicos de comando, que são os selos dados pelos contatos abertos de O1 e O2, e também os intertravamentos dados pelos contatos fechados de O1 e O2.

Observando os dois exemplos dados, pode-se definir agora os elementos essenciais em uma programação Ladder:

Tabela 2.1 – Elementos Básicos em Ladder		
Nomeclatura	Abreviação	Símbolo
Contato Normalmente Aberto	NA	
Contato Normalmente Fechado	NF	
Bobina ou Saída	---	

Com os elementos básicos montam-se diversas combinações importantes, mostradas nos próximos itens.

2.1) Funções Lógicas em Ladder

As funções lógicas são estudadas em todos e quaisquer elementos. A combinação entre os contatos NA e NF servem de importante orientação para o projetista e programador de circuitos lógicos.

A) Função “E” (AND)

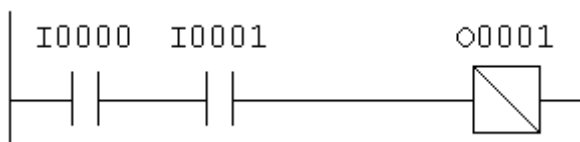


Tabela verdade		
I1	I2	Q1
0	0	0
0	1	0
1	0	0
1	1	1

B) Função “OU” (OR)

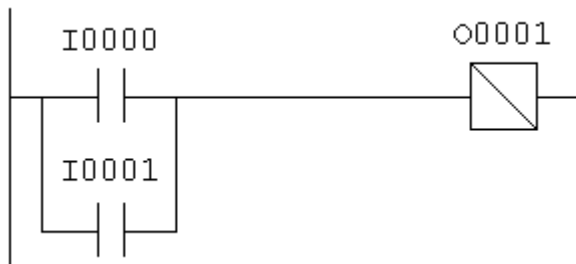


Tabela Verdade

I1	I2	Q1
0	0	0
0	1	1
1	0	1
1	1	1

D) Função “Não E” (NAND)

Segundo a álgebra de Boole: $\overline{A * B} = \overline{A} + \overline{B}$. Assim a função NAND consiste em dois contatos NF em paralelo.

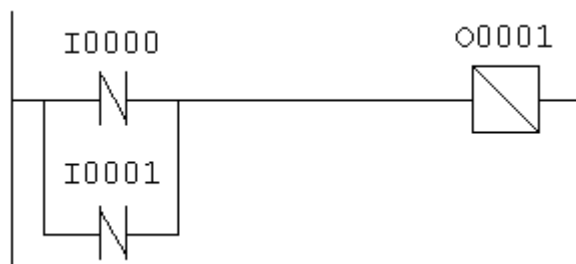


Tabela verdade

I1	I2	Q1
0	0	1
0	1	1
1	0	1
1	1	0

D) Função “Não OU” (NOR)

Segundo a álgebra de Boole: $\overline{A + B} = \overline{A} * \overline{B}$. Assim a função NOR consiste em dois contatos NF em série.

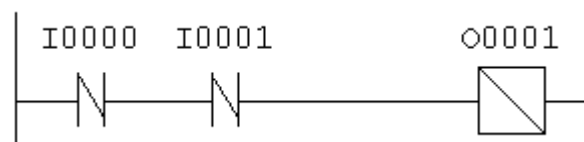


Tabela verdade

I1	I2	Q1
0	0	1
0	1	0
1	0	0
1	1	0

2.2) Circuitos de Selo

Os selos são as combinações mais básicas entre elementos, destinados a manter uma saída ligada, quando se utilizam botoeiras.

A) Selo com prioridade no ligamento

Com as duas chaves pressionadas o circuito sempre estará ligado.

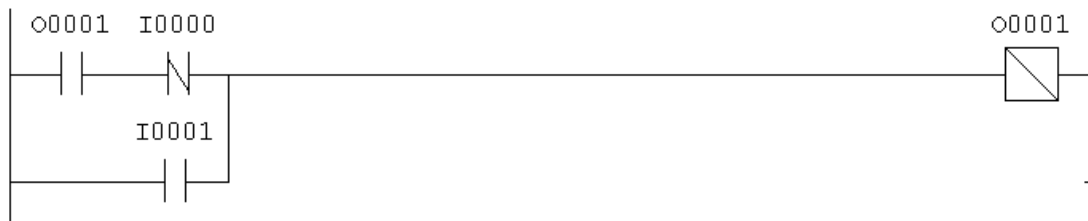


Figura 2.6 – Selo com prioridade no ligamento

A) Selo com prioridade no desligamento

Com as duas chaves pressionadas o circuito sempre estará desligado. É o mais utilizado por questões de segurança.

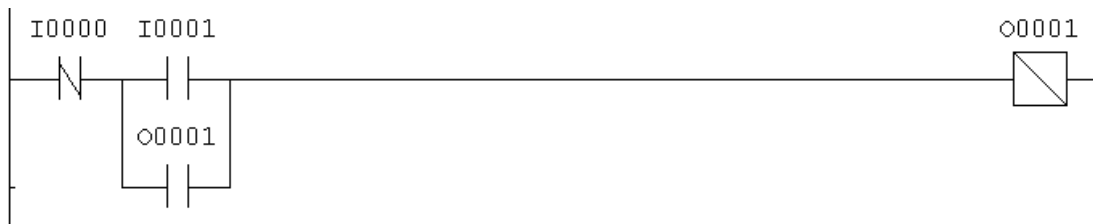


Figura 2.7 – Selo com prioridade no desligamento

2.3) Instruções de ‘SET’ e ‘RESET’

A instrução de “SET” liga uma saída e mantém a mesma ligada mesmo que a alimentação da entrada seja retirada. Para se desligar a saída utiliza a instrução “RESET”. A figura 2.8 mostra um exemplo da utilização destas instruções na partida direta de um motor. O programa na figura 2.8 é equivalente ao programa mostrado na figura 2.7.

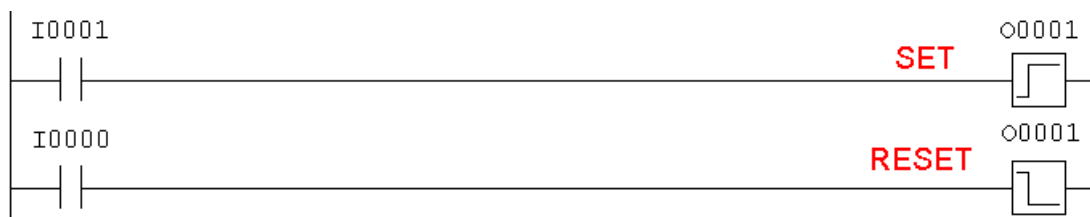


Figura 2.8 – Utilização das Instruções de SET e RESET

2.4) Circuitos de Detecção de Borda

Existem situações em que é necessário registrar não o estado da entrada, mas sim o instante em que essa entrada comuta. Isso é realizado pelos circuitos de detecção de borda, que podem detectar o flanco ascendente (instante de ativação da entrada) quanto o flanco descendente (instante de desativação da entrada).

Estes circuitos se aproveitam do modo de operação do CLP onde a varredura é feita através de uma linha de cada vez. A figura 2.9 mostra o exemplo de detecção de borda durante a subida.

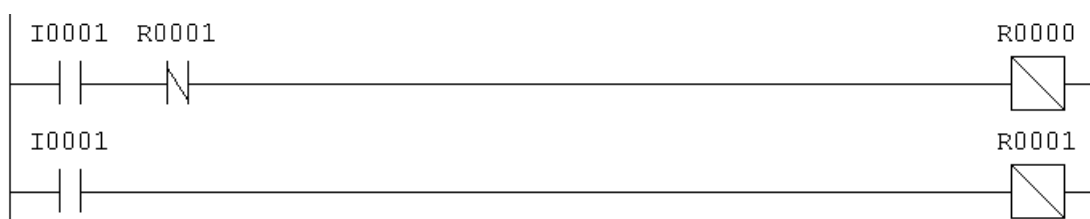


Figura 2.9 – Circuito de detecção de borda

Uma aplicação prática deste circuito é quando se deseja ativar e desativar uma saída com um único pulsador (ou botoeira). O circuito completo para este tipo de operação de operação é mostrado na figura 2.10.

É importante notar que no programa da figura 2.10 nota-se que utilizou-se a letra “R” na saída e não “O”. A diferença é que “R” significa “**Relé de contato auxiliar**”, ou seja quando se aciona “R” nenhuma saída externa ao CLP é ligada. Este relé representa uma “memória” interna do CLP e como o próprio nome já diz, serve somente para auxiliar na lógica do programa. Este elemento é muito utilizado em programação com diagramas de contato.

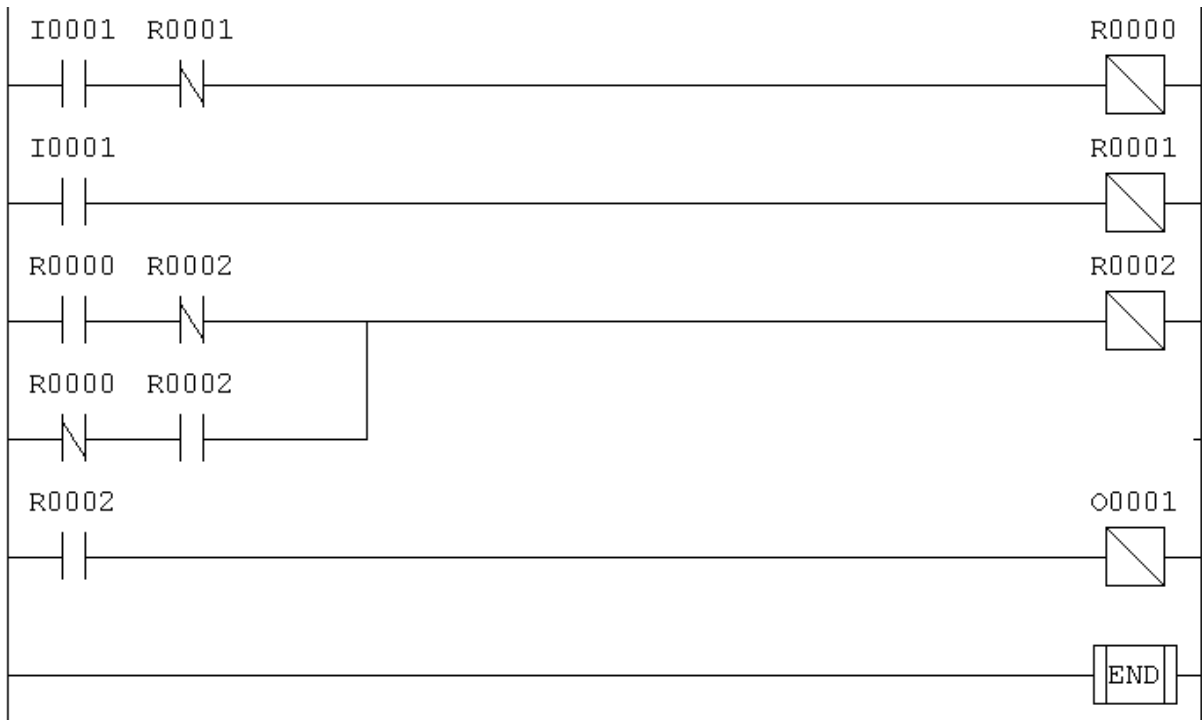


Figura 2.10 – Circuito de detecção de borda

Para facilitar a programação, o CLP apresenta as funções de detecção de borda e acionamento com um único pulsador. A tabela 2.2 apresenta os símbolos destas funções.

Tabela 2.2 – Elementos de pulso		
Nomeclatura	Abreviação	Símbolo
Flip Flop SET de borda (aciona somente na borda de subida do sinal de entrada)	--	
Flip Flop RESET de borda (aciona somente na borda de subida do sinal de entrada)	--	
Flip Flop de Pulso (liga e desliga a saída dependendo do estado anterior)	---	

2.5) Mensagens na IHM do CLP

Todos os Controladores Lógicos Programáveis (CLPs) tem como opcional a chamada Interface Homem Máquina (IHM). Esta normalmente consiste de um teclado para entrada de dados e uma tela (display) onde pode-se visualizar dados e mensagens destinadas ao operador, técnico de manutenção ou programador.

Como exemplo tem-se o seguinte problema: deve-se fazer um programa em ladder para comandar uma partida com reversão, de forma a mostrar na IHM do controlador, as mensagens de “sentido-horário”, “anti-horário” e “operação ilegal”. Esta última deve aparecer em três condições distintas: o operador pressiona as duas botoeiras de forma simultânea, o motor gira no sentido anti-horário e ele pressiona a botoeira do sentido contrário e vice-versa. O programa correspondente é mostrado na figura 2.11.

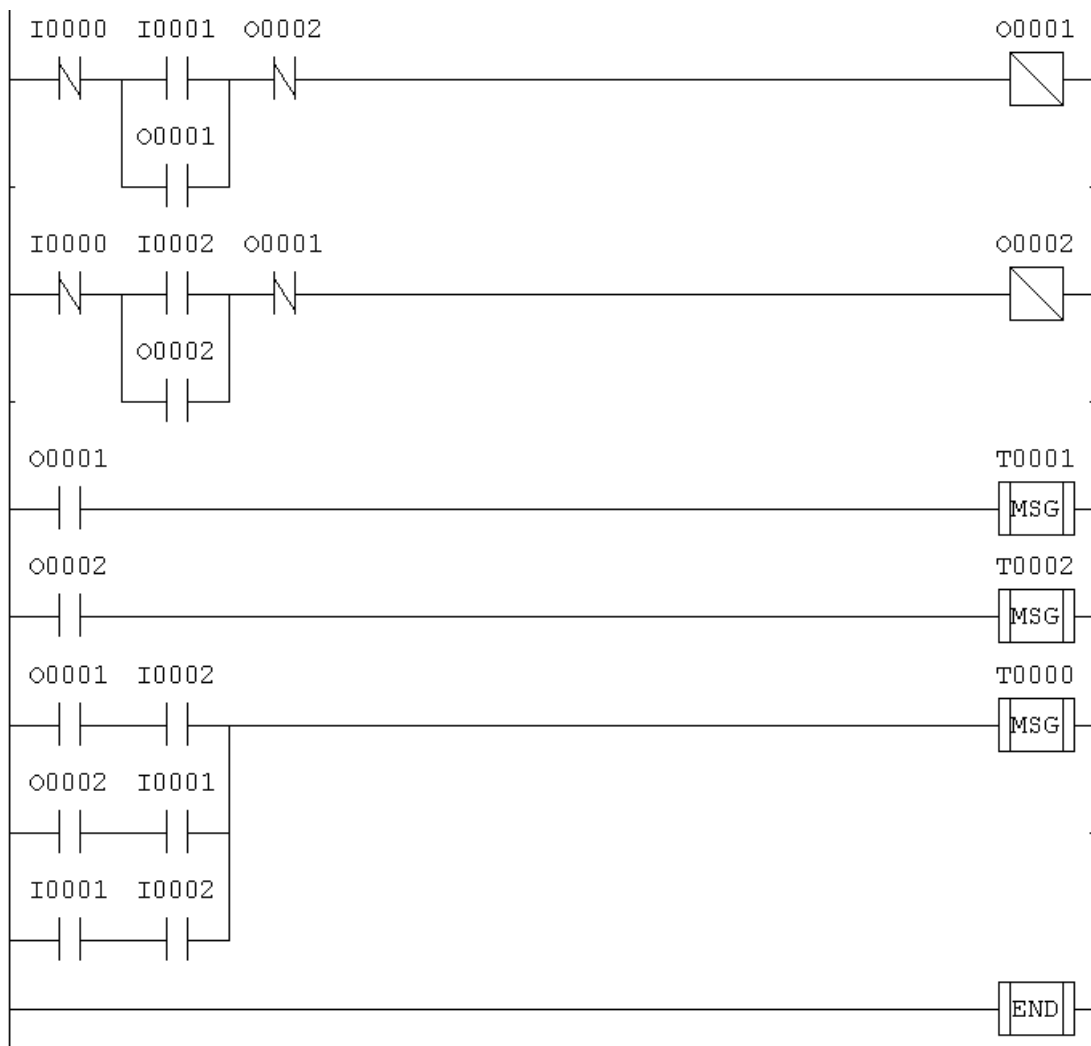


Figura 2.11 – Programa para partida com reversão mensagens

Na tela de inserção de mensagens foram inseridas:

- T0000 → operação ilegal
- T0001 → Sentido Horário
- T0002 → Sentido Anti-horário

Deve-se lembrar que o CLP não pode mostrar duas mensagens na tela ao mesmo tempo, assim ele prioriza as mensagens por ordem crescente de inserção, ou seja, T0000 tem prioridade sobre T0001, que por sua vez tem prioridade sobre T0002, e assim por diante. No programa da figura 2.11 a mensagem de operação ilegal tem prioridade sobre as demais, pois ela pode aparecer mesmo quando existe outra mensagem sobre a tela, assim sendo ela foi inserida em T0000.

2.6) Particularidades do CLP utilizado neste curso

Como já foi dito anteriormente, a linguagem de programação é padronizada. Entretanto os diferentes fabricantes apresentam certas diferenças e particularidades. Estas são mostradas, de forma resumida, para o CLP utilizado no laboratório que é o HI ZAP500. Maiores detalhes devem ser consultados no manual do fabricante. Algumas características deste controlador são:

- Alimentação de 8 a 38 Vcc
- Consumo: 2,5 Watts
- Temperatura de operação de 0 a 65 °C
- 10 canais de I/O (Entradas / Saídas)
- Interface IHM com teclado de 15 teclas e 10 Led's Programáveis

O ambiente de desenvolvimento é composto por um Menu Principal com as seguintes opções:

- **Projeto:** carrega um projeto existente ou cria um novo
- **Programa:** edita um programa em Ladder

- **Controlador:** gera um código executável para ser enviado ao controlador ZAP500
- **Depurador:** permite a visualização do programa em funcionamento no micro-controlador
- **Supervisão:** gera a supervisão de blocos de controle (PID) ou gráficos de tendência ("Trend Variáveis")

Para maiores detalhes de como inserir um programa no controlador o aluno deve consultar o Anexo 1 da apostila. Os principais operadores utilizados neste CLP são:

- **I:** Representa as entradas digitais. As diferentes entradas são distinguidas através de números seqüenciais. Ex: I1, I2, I3,.....
- **O:** Representa as saídas digitais. As diferentes saídas são distinguidas através de números seqüenciais. Ex: O1, O2, O3,.....
- **R:** Representa um contato auxiliar. Estes não tem conexão direta com o meio físico (processo) e são úteis na definição das lógicas. Também são diferenciados através de números seqüenciais. Ex: R1, R2, R3,.....
- **M:** Memória destinada a guardar valores inteiros de 16 bits, ou seja, números de -32768 a $+32768$
- **D:** Memória destinada a armazenar valores reais, ou seja, números de -10^{-38} a $+10^{+38}$
- **K:** Memória destinada a armazenar uma constante inteira de 16 bits. Pode assumir valores de -999 a $+9999$
- **H:** Representa o valor de uma constante Hexadecimal. Pode assumir valores na faixa de 0 a $FFFF$.
- **Q:** Representa o valor de uma constante real. Pode ser usado com números na faixa de -10^{-38} a $+10^{+38}$
- **E:** Representa um canal de entrada analógica. Usado pelo bloco de movimentação "MOV", quando se deseja lê o valor de uma entrada analógica do controlador em uma memória.

- **S**: Representa um canal de saída analógica. Usado pelo bloco de movimentação “MOV”, quando se deseja movimentar o valor de uma memória a uma saída.
- **T**: Representa um elemento de sinalização de eventos, como por exemplo, para identificar uma mensagem. Identifica uma lógica qualquer, presente em um programa. Representa um malha de controle PID

Exercícios do Capítulo 2

E2.1) Faça um programa para comandar um motor de indução trifásico em partida direta de modo que o operador deve pressionar duas botoeiras simultaneamente para acionar o mesmo. Desenhe as ligações elétricas externas ao CLP para este comando.

E2.2) Programa a partida com reversão utilizando as funções de SET e RESET.

E2.3) Programe uma partida com reversão de modo que o operador possa ligar e desligar o motor com apenas uma botoeira para cada sentido de rotação.

E2.4) Um motor de indução trifásico tem de ser ligado e desligado através de dois lugares diferentes em partida direta. Faça um programa em Ladder para comandar este motor.

E2.5) No exercício 2.4 deve-se acrescentar as seguintes mensagens:

- a) Se o motor for ligado no ponto 1: “Ligado em 1”
- b) Se o motor for ligado no ponto 2: “Ligado em 2”
- c) Se o motor estiver desligado: “Motor em espera”

E2.6) Faça um programa em Ladder para comandar um motor elétrico com reversão, de modo que para acionar o sentido horário o operador deve pressionar duas botoeiras de forma simultânea. Para acionar o sentido anti-horário basta pressionar uma botoeira. No sentido horário basta o operador retirar as duas mãos para que o motor pare. No sentido anti-horário o operador deve pressionar uma botoeira S0 para interromper o funcionamento.

E2.7) Dois motores elétricos trifásicos devem ser ligados em partida direta, de modo que:

- a) O primeiro motor pode ser comandado através de um ponto apenas;
- b) O segundo motor pode ser ligado em dois pontos de forma independente.

Os motores devem ter o funcionamento independente um do outro. Faça um programa em Ladder para este comando.

E2.8) Deve-se acionar dois motores elétricos de modo que o primeiro pode ser ligado de forma independente e o segundo só pode ser ligado se o primeiro também estiver ligado. O primeiro motor funciona em partida direta e o segundo motor funciona em partida com reversão.

E2.9) Deseja-se comandar um cilindro pneumático de dupla ação com avanço por botoeira e retorno automático no fim do curso. Especifique os elementos pneumáticos, faça um desenho do sistema pneumático, mostre as ligações das entradas e saídas com o CLP e finalmente faça um programa em Ladder para este comando.

E2.10) Incremente o programa do exercício (E2.9) de modo que ao pressionar a botoeira de avanço o sistema funcione ininterruptamente, até que a botoeira de desligamento seja pressionada. Você deve acrescentar algum elemento ao circuito pneumático?

3 - Temporizador e recursos avançados da IHM

O temporizador é um elemento fundamental na programação em Ladder, pois a partir deste elemento pode-se montar diversas aplicações.

O símbolo do temporizador para o CLP HI ZAP 500 é mostrado na figura 3.1. Este tem duas entradas E1 e E2, uma saída S1. A saída é acionada após a contagem do tempo programado em M2, que pode ser uma memória inteira, real ou uma constante. A memória M1 é de uso próprio do controlador e se destina a fazer a contagem regressiva do tempo, quando o temporizador estiver acionado.

A entrada E1 destina-se a paralização da contagem do tempo e a entrada E2 zera a contagem do mesmo. O comportamento do controlador, conforme a combinação das entradas é mostrado na tabela 3.1.

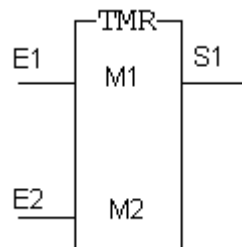


Figura 3.1 – Bloco Temporizador do CLP HI ZAP500

E1	E2	Temporizador
0	0	Zerado em espera
0	1	Paralisado
1	0	Zerado
1	1	Contando

Outra característica importante do temporizador é que ele conta tempos de 0,01s. Assim para programar 5s seria necessário fazer $M1=500$, como mostra a equação 3.1.

$$M1 = \frac{\text{tempo desejado}}{0,01} = \frac{5}{0,01} = 500 \quad (3.1)$$

3.1) Partida Seqüencial de motores

Uma aplicação interessante do temporizador consiste na partida seqüencial de motores. Para exemplificar pode-se fazer um programa para acionar 3 motores de forma que o primeiro seja acionado imediatamente, o segundo deve ser acionado 5s após o primeiro e o terceiro 8s após o primeiro. O programa é apresentado abaixo, nota-se que a aplicação é simples e direta. Esta estratégia é muito utilizada para evitar picos de consumo de energia e corrente, caso os motores partissem todos simultaneamente.

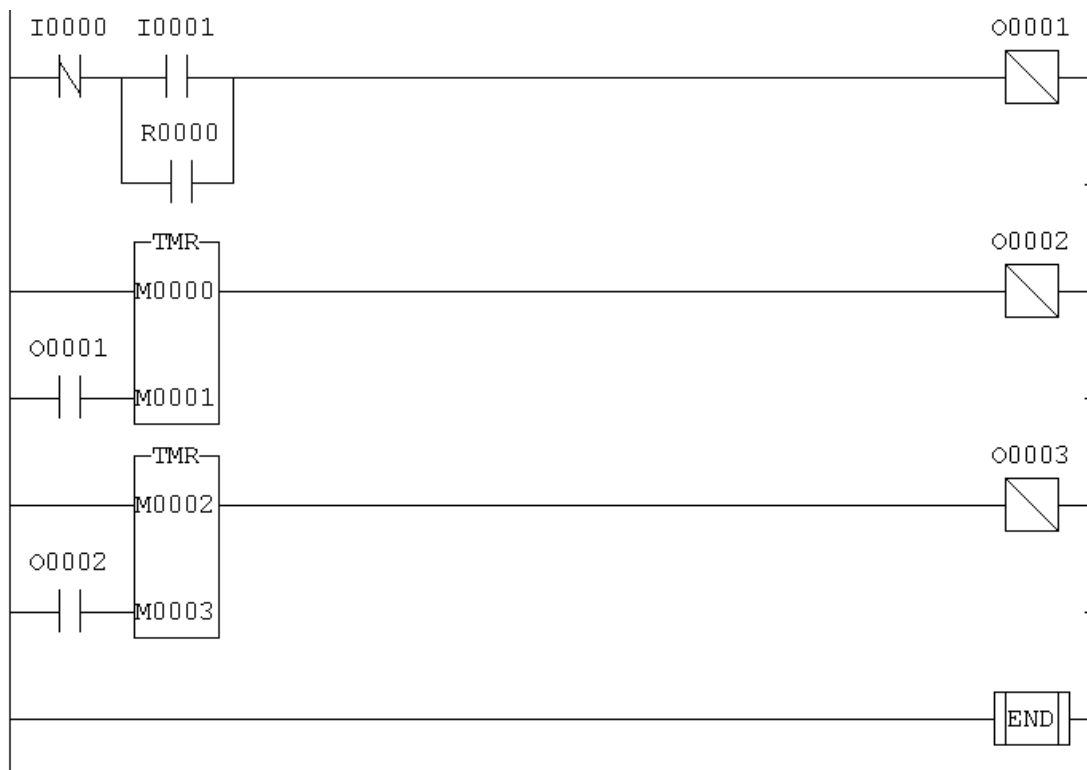


Figura 3.2 – Programa para uma partida seqüencial

Deve-se lembra que M1=500 e M3=800, para a correta contagem do tempo.

3.2) Partida Y/Δ de motores

Uma outra aplicação do temporizador é no comando de uma partida Y/Δ. Neste programa deve-se lembrar que são utilizados 3 contatores de forma que um contator está sempre ligado. O contator Y primeiramente e desliga após o tempo programado. O contator Δ liga somente após o tempo programado. Deve haver um intertravamento entre os contatores Y/Δ.

Uma das formas de confeccionar o programa é fazendo a cópia do diagrama elétrico, já visto na disciplina de comandos elétricos. O programa resultante é mostrado na figura 3.3.

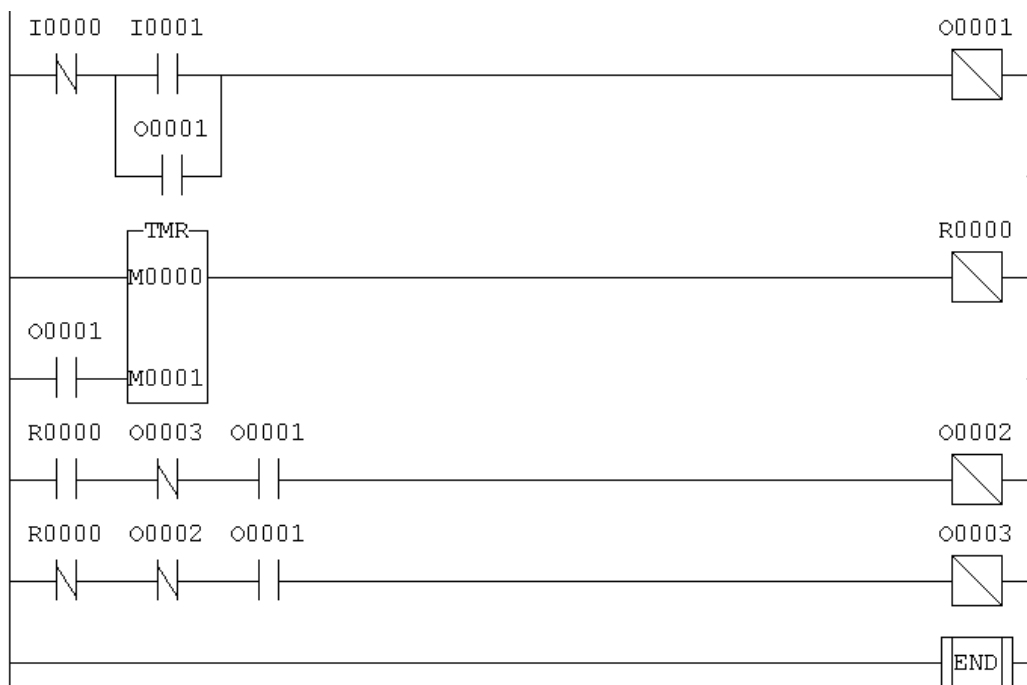


Figura 3.3 – Programa para uma partida Estrela-Triângulo

No programa da partida estrela-triângulo os elementos são: I0 está conectada a botoeira de desligamento, I1 à botoeira de ligamento, O1 ao contator fixo, O2 ao contator triângulo e O3 ao contator estrela.

3.3) Partida Estrela-Triângulo com programação da Interface-Homem-Máquina (IHM)

Na partida estrela-triângulo o tempo de comutação entre os contatores é um parâmetro programável, que pode mudar dependendo da carga acionada. Assim em certas situações é adequado permitir que o operador programe este tempo. Como ele não terá no ambiente fabril um micro computador disponível, é interessante que o mesmo faça através da IHM do controlador. Isso é possível a partir da tela de memórias, como mostra a figura 3.4, onde é possível permitir a programação do valor desejado durante a execução do programa.



Figura 3.4 – Tela de programação de Memórias

Para que operador não necessite fazer contas durante esta programação, uma outra função utilizada é a função de Multiplicação, na guia de funções matemáticas. O programa é apresentado na figura 3.5.

Os valores de M2 e M3 são multiplicados e transferidos a M1. Assim pode-se habilitar o operador para programar o valor de M2 e fazer M3 constante e igual a 100. Dessa forma o tempo pode ser digitado diretamente em segundos. Este tipo de programação utiliza recursos mais avançados na IHM do CLP.

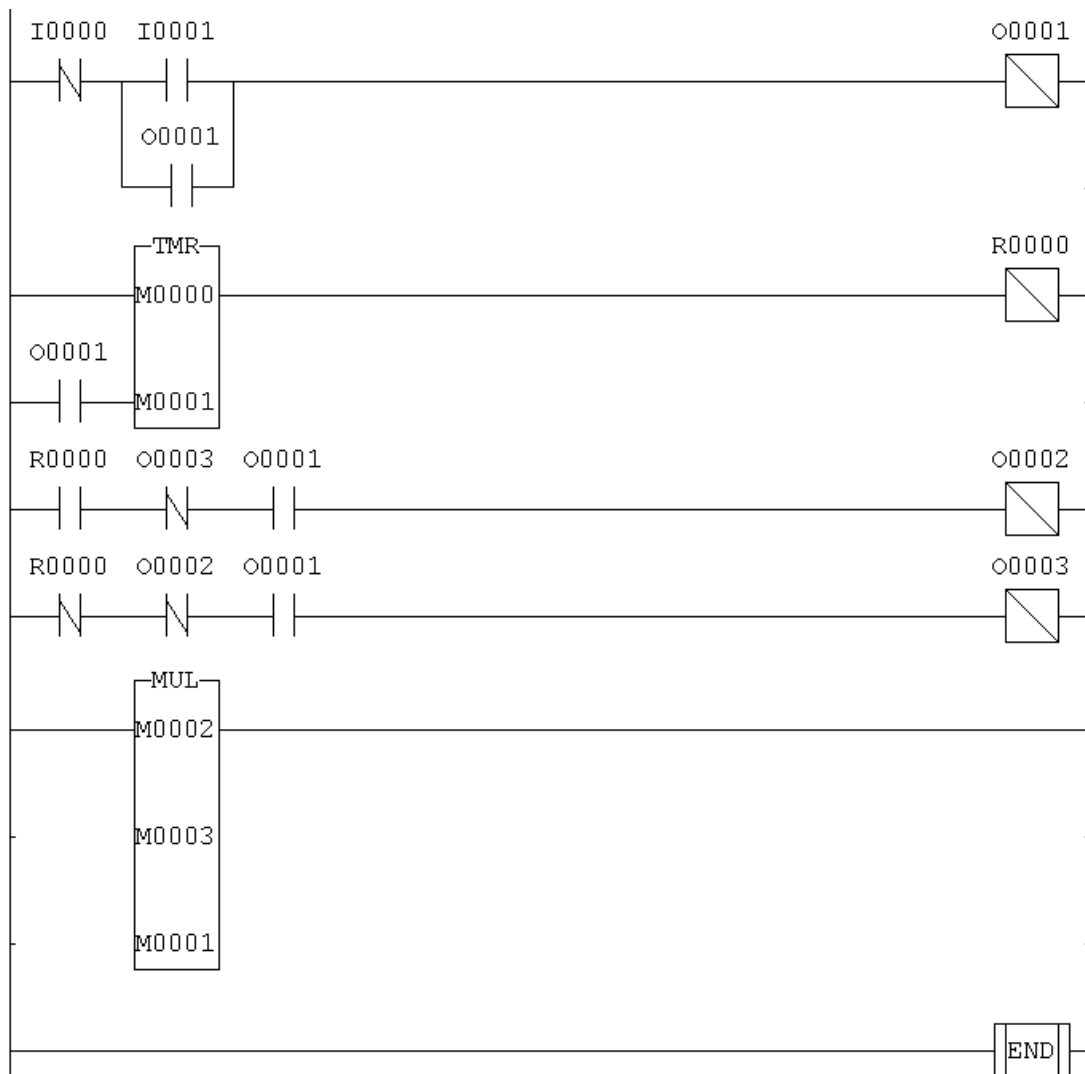


Figura 3.5 – Partida Estrela-Triângulo com Multiplicador

3.4) Segurança no comando de prensas

No comando de prensas antigamente utilizavam-se duas botoeiras em série para evitar que o operador estivesse com uma mão livre ao acionar a mesma. Entretanto eles começaram a prender um dos botões com um peso, tornando a estratégia ineficaz. Nos comandos modernos faz-se com que as botoeiras devam ser acionadas em um intervalo de tempo menor que 3s, por exemplo, caso contrário a prensa não aciona. O programa em Ladder para acionamento é mostrado na figura 3.6.

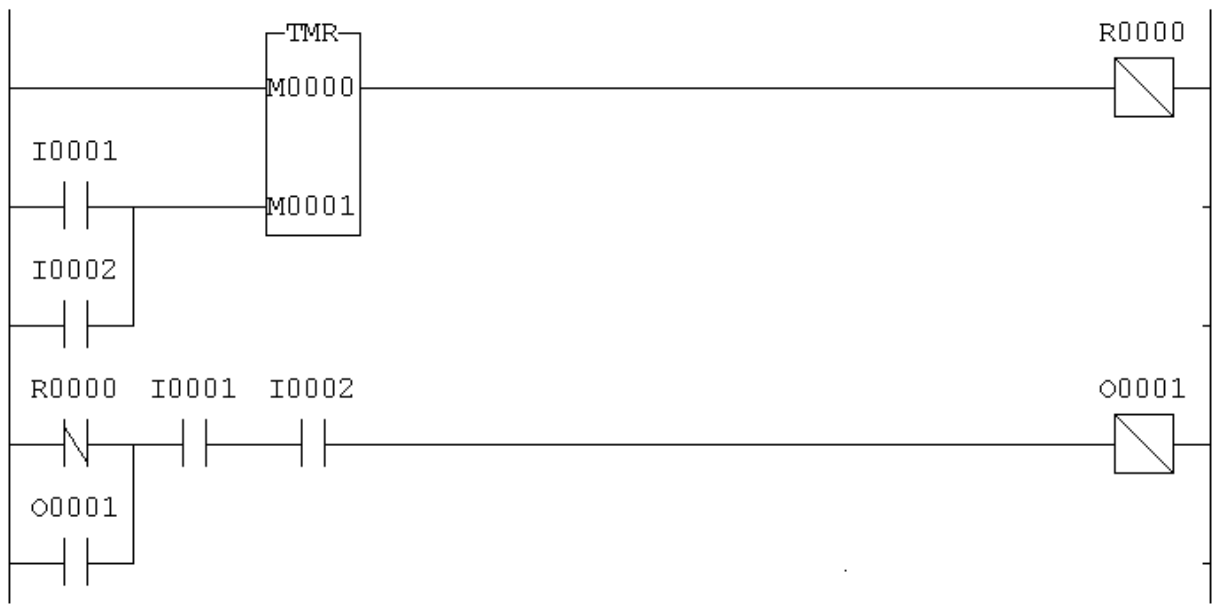


Figura 3.6 – Segurança em prensa com temporizador

Exercícios do Capítulo 3

E3.1) No programa para partida Y/ Δ acrescente as mensagens na IHM do CLP: “motor em estrela”; “motor em triângulo” e “motor em espera”.

E3.2) Uma válvula de descarga, normalmente fechada, deve ser ligada através de uma botoeira. Após o acionamento do operador, esta deve ficar aberta por apenas 10s. Faça um programa em ladder para este comando.

E3.3) Dois motores devem ser acionados de modo que ao pressionar a botoeira os dois ligam simultaneamente, e o segundo motor desliga 6s após o acionamento do primeiro.

E3.4) Dois motores de indução trifásicos devem ser comandados de modo que o segundo motor só pode ser ligado se o primeiro estiver funcionando em sentido anti-horário. O segundo motor funciona em partida direta e também só aciona 5s depois que o operador tiver pressionado a botoeira.

E3.5) Complementar o programa do exercício 3.4 com as mensagens: “motor 1 horário”; “motor 2 anti-hor”, “motores 1 e 2” e “operação ilegal” caso o operador tente ligar o motor 2 quando o motor 1 estiver em sentido horário.

E3.6) Elaborar um programa PLC para controlar dois relés (R1 e R2) de tal maneira que R1 pode atuar de forma independente e R2 só pode atuar se R1 estiver ligado, mas pode continuar ligado após o desligamento de R1. Os relés são ligados pelas botoeiras L1 e L2, e são desligados pelas botoeiras D1 e D2.

E3.7) Confeccionar um programa em Ladder para comandar um cilindro pneumático de dupla ação, com retorno automático, de modo que para retornar haja uma espera de 5s.

E3.8) Fazer um programa para comandar 3 saídas de modo que ao pressionar a botoeira:

- a) a primeira saída liga instantaneamente;
- b) a segunda saída liga “8s” após a primeira;
- c) a terceira saída liga “5s” após a segunda.

4 - Introdução ao trabalho com variáveis analógicas

As variáveis analógicas são mais utilizadas em controles de processos químicos. Exemplos clássicos são as caldeiras de vapor, as usinas de açúcar, e indústrias de papel e celulose onde estão envolvidas variáveis tais como pressão, temperatura e vazão. Os dados fornecidos pelos sensores podem ser utilizados de forma simples, para interromper o funcionamento de uma caldeira por exemplo, ou de forma complexa, em controles Proporcionais-Integrais-Derivativos, onde se busca atingir um alvo, ou “set-point”.

No CLP utilizado as as entradas analógicas tem resolução de 10 bits, e são configuráveis para sinais de correntes de 0 a 20 ma, 4 a 20 ma, ou tensão de 0 a 5 Vdc ou 1 a 5 Vdc. A configuração dependerá do sensor que está sendo ligado a referida entrada. A saída analógica tem resolução de 9 bits com sinal de corrente de 4 a 20 ma e impedância máxima de 1Kohm.

As entradas analógicas do CLP HI ZAP 500 estão localizadas no módulo de expansão ZEM530, são identificadas pela letra “E”, num total de 4 entradas: E0000 a E0003.

A leitura de dos valores analógicos é feita através da função MOV. Outras funções que podem ser utilizadas são encontradas nas guias “movimento”, “matemáticas” e “comparação”. Na tabela 4.1 encontram-se os símbolos de algumas destas funções. Maiores detalhes devem ser consultados no manual fabricante.

Tabela 4.1 – Exemplo de funções utilizadas com variáveis analógicas	
Nome e função	Símbolo
→ Movimentação de Variáveis (MOV) → Transfere o valor lido em E0000 para a memória inteira M0001. É utilizada na leitura de variáveis analógicas e também para transferir o valor para uma saída analógica.	
→ Soma de variáveis (ADD) → Soma o valor das memórias inteiras M0001 e M0002 colocando o resultado em M0003. Pode ser usado, assim como as outras funções matemáticas, para somar valores já lidos de entradas analógicas.	
→ Comparação maior que (>) → Testa se o valor da memória M0001 é maior que o valor da memória M0002. Se isto for verdadeiro pode acionar uma saída especificada pelo usuário.	

Como exemplo de manipulação de variáveis analógicas tem-se o programa mostrado na figura 4.1, cujo propósito é o de ler uma variável, comparar com um valor pré-estabelecido e acionar uma saída qualquer (no caso 01).

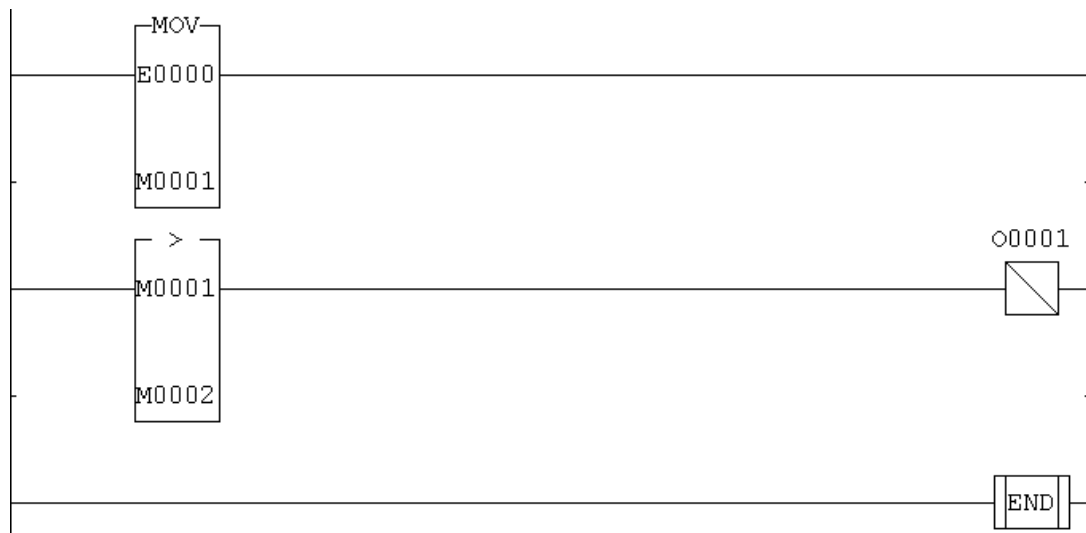


Figura 4.1 – Leitura e comparação de uma variável analógica

Uma aplicação prática seria no uso em caldeiras, onde pode-se ler a entrada de um sensor de pressão e/ou temperatura, interrompendo o funcionamento da mesma em caso de valores altos.

Outra aplicação, para este tipo de leitura, seria na proteção de motores elétricos, onde poderia ser colocado um sensor de temperatura na carcaça do mesmo. Em caso de aquecimento este deve ser desligado. A figura 4.2 mostra um programa para fazer tal comando.

No programa apresentado o sensor de temperatura é ligado na entrada E0000, caso o valor medido seja maior que um valor especificado em M0001, o motor para. O contator de potência que aciona o motor é ligado na saída O0001.

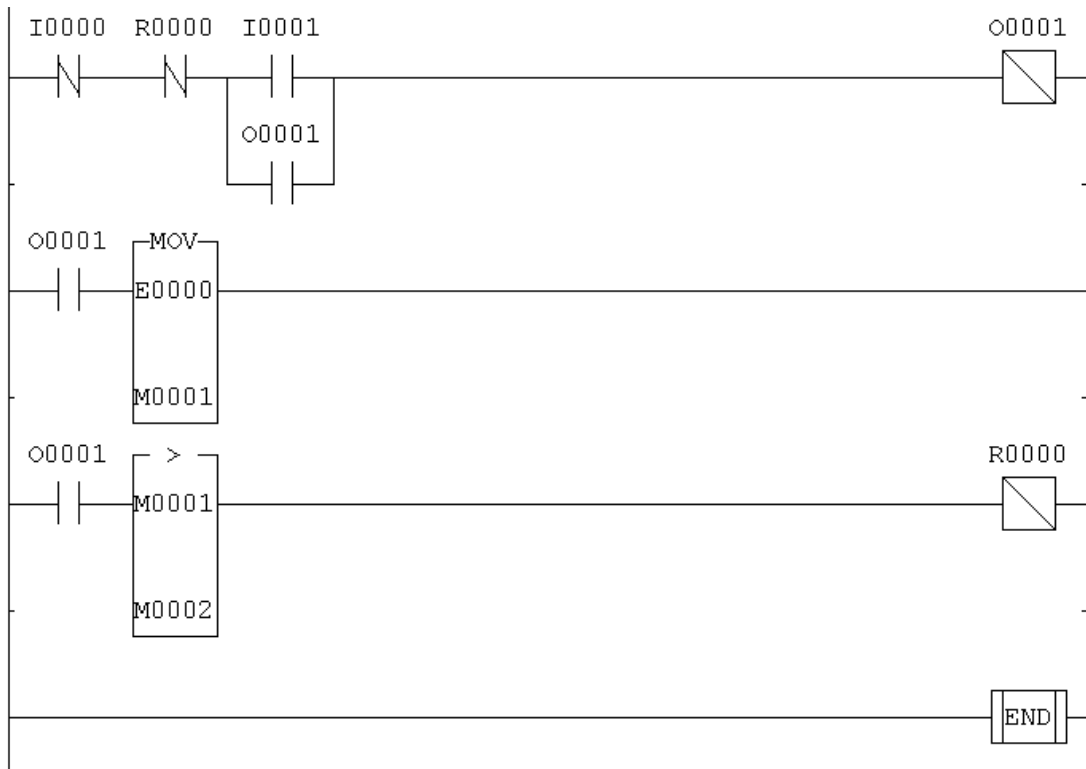


Figura 4.2 – Aplicação na proteção de motores elétricos

Exercícios do capítulo 4

E4.1) Incremente o programa de proteção de motores, dado na teoria, de modo que antes de se desligar o motor haja uma espera de 5s. Isto porque o funcionamento do motor em sobre-corrente é garantido pelo fabricante do mesmo por um intervalo de tempo determinado.

E4.2) Incremente o programa do exercício E4.1, inserindo as mensagens: “motor em espera”, “sobrecarga” e “motor em estado crítico”, caso a temperatura esteja em 90% do valor limite.

E4.3) Incremente o programa do exercício E4.2, de modo que o operador possa ajustar na IHM do CLP os valores da temperatura de interrupção e do tempo de espera.

Referências Bibliográficas

Silveira, Paulo R. da; Santos, Winderson E.; “*Automação e Controle Discreto*”; Editora Erica; São Paulo; 1988.

Rosário, João Maurício; “Princípios de Mecatrônica”; Editora Pearson Prentice Hall; São Paulo; 2005.

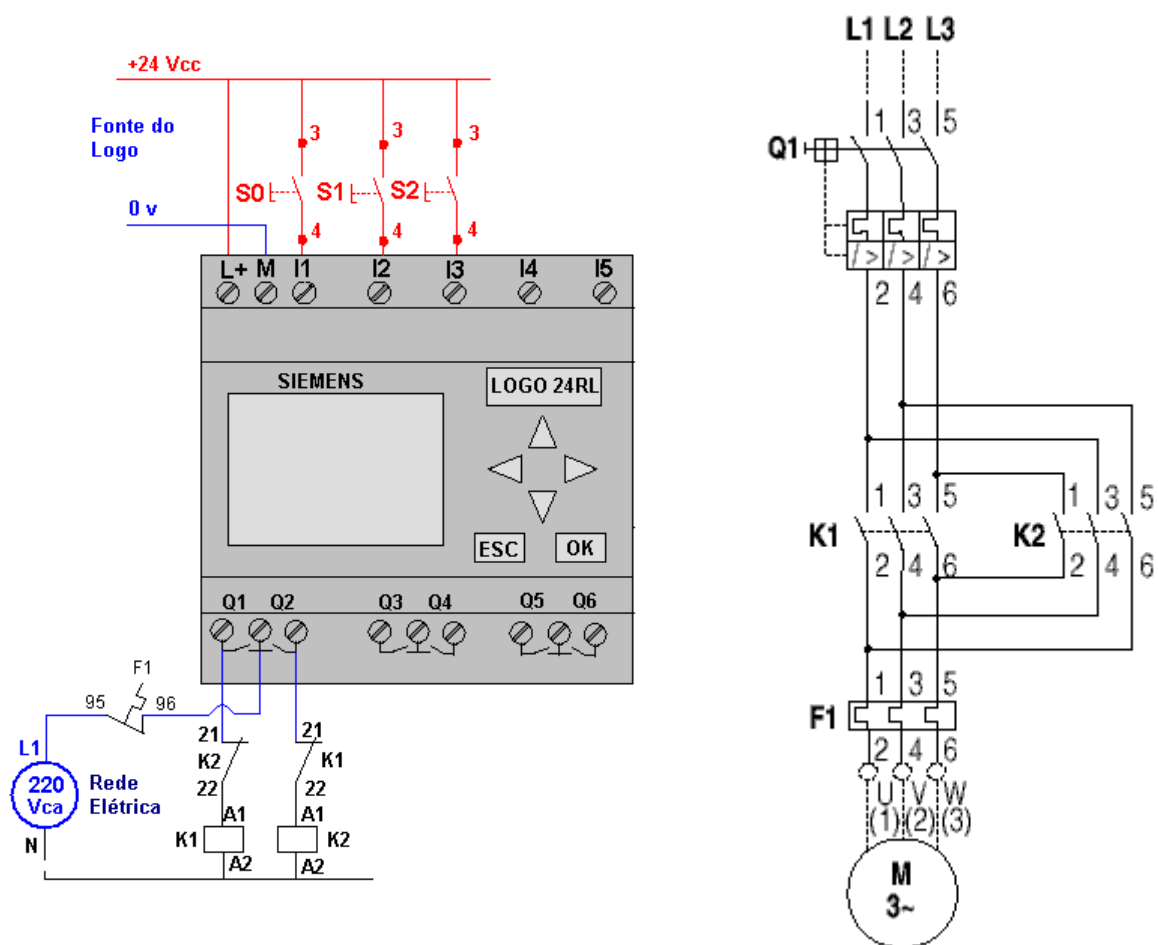
Natale, Ferdinando; “Automação Industrial”; Editora Érica; São Paulo; 1995

Moraes, Cícero Couto de; Castrucci, Plínio de Lauro; “Engenharia de Automação Industrial”, Editora LTC,;Rio de Janeiro; 2001

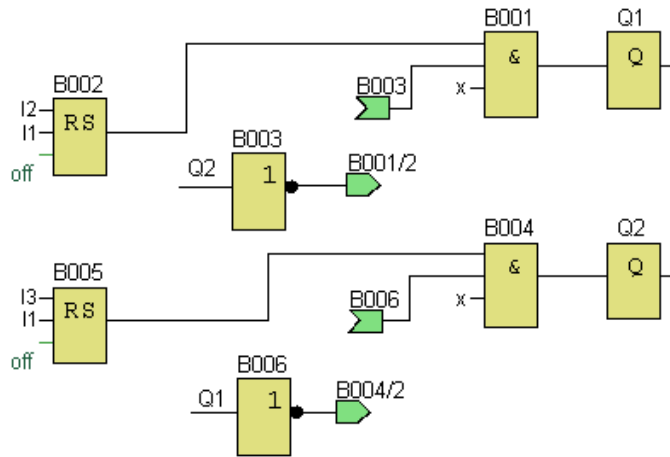
Manuais e Catálogos dos Fabricantes

ANEXO I – LABORATÓRIO – LIGAÇÕES ELÉTRICAS EXTERNAS AO CLP

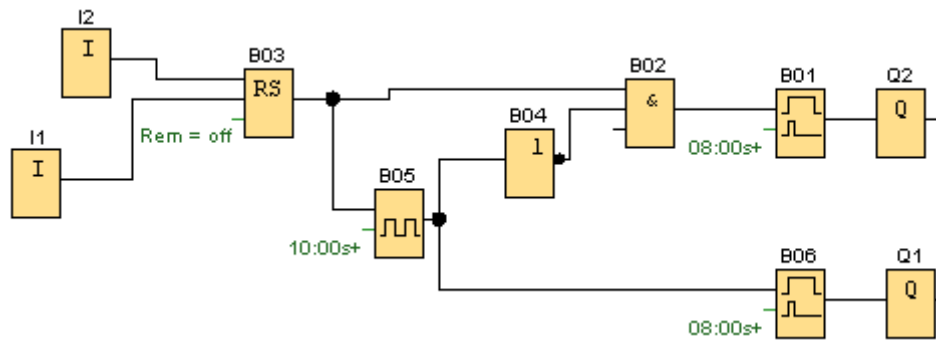
- 1) Fazer a ligação elétrica do comando.
- 2) Inserir o programa para uma partida simples com reversão no logo e testar.
- 3) Inserir o programa para partida automática no Logo e testar.
- 4) Fazer a ligação elétrica da potência, testando-a separadamente. Se tudo estiver funcionando separadamente, testar o conjunto.
- 5) Elaborar um relatório a ser entregue na próxima aula.



Partida com reversão simples – parada obrigatória

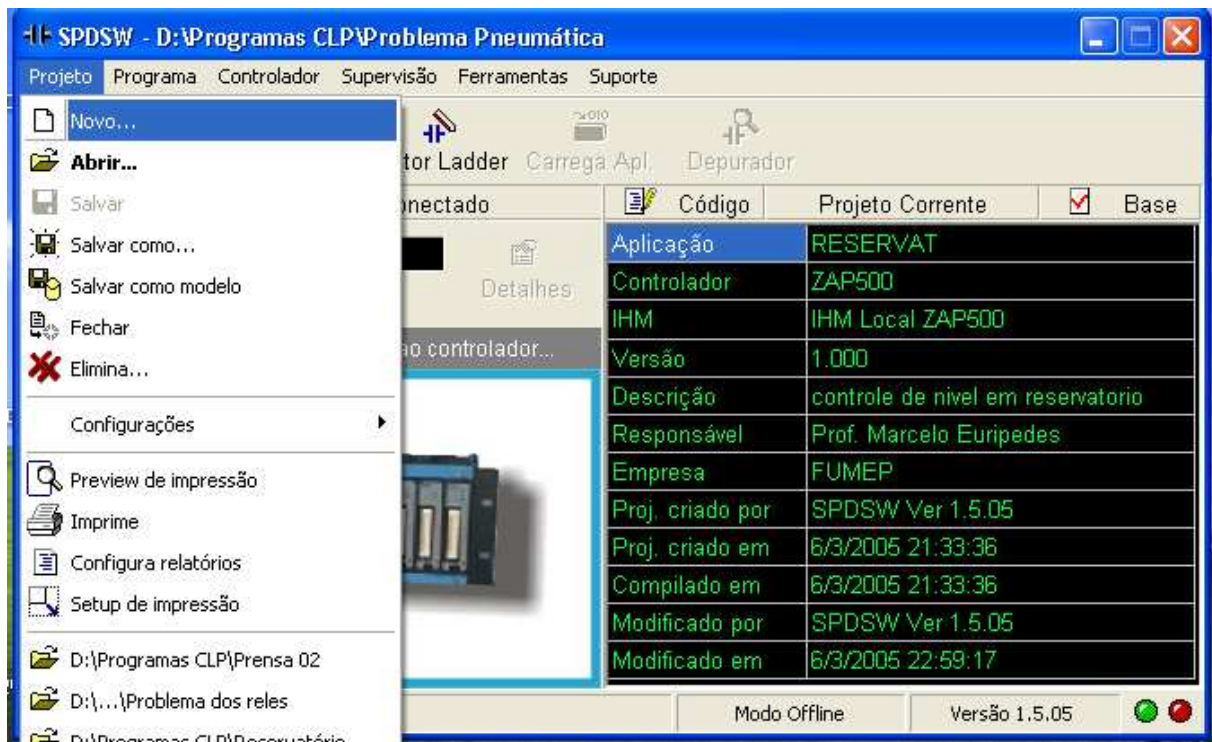


Partida com reversão automática

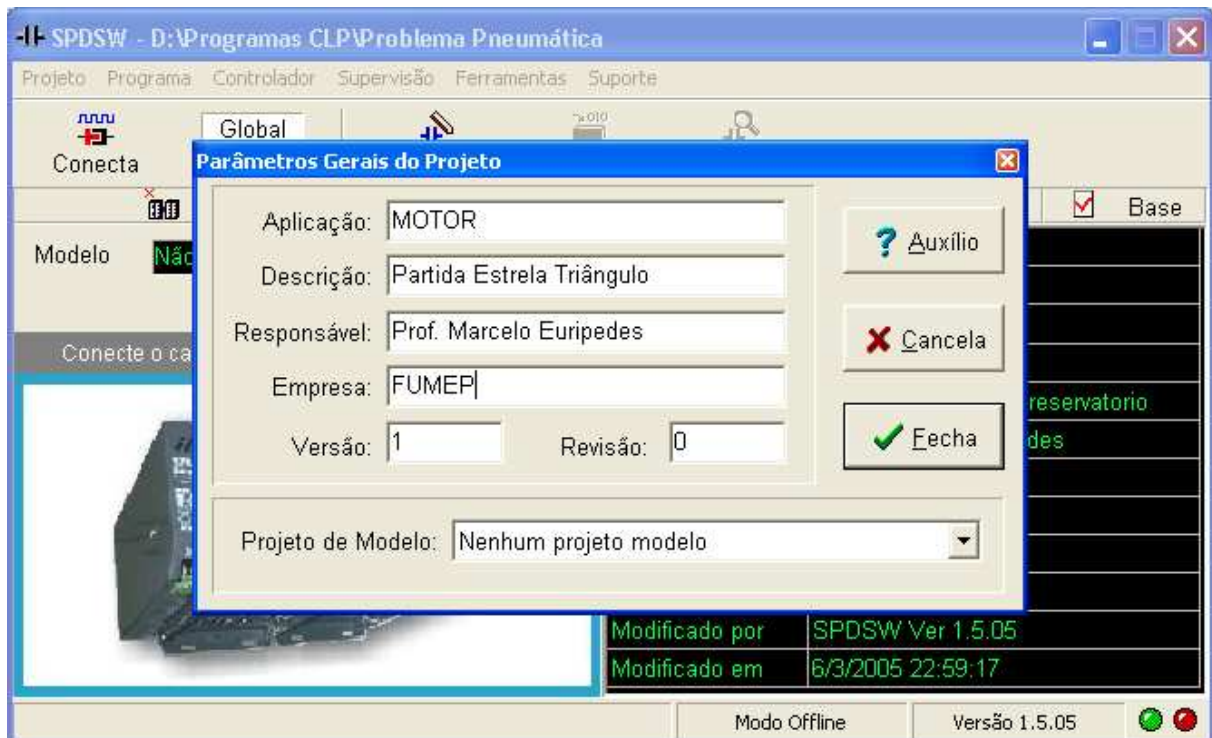


ANEXO II – COMO INICIAR UM PROJETO NOVO NO CLP HI ZAP 500

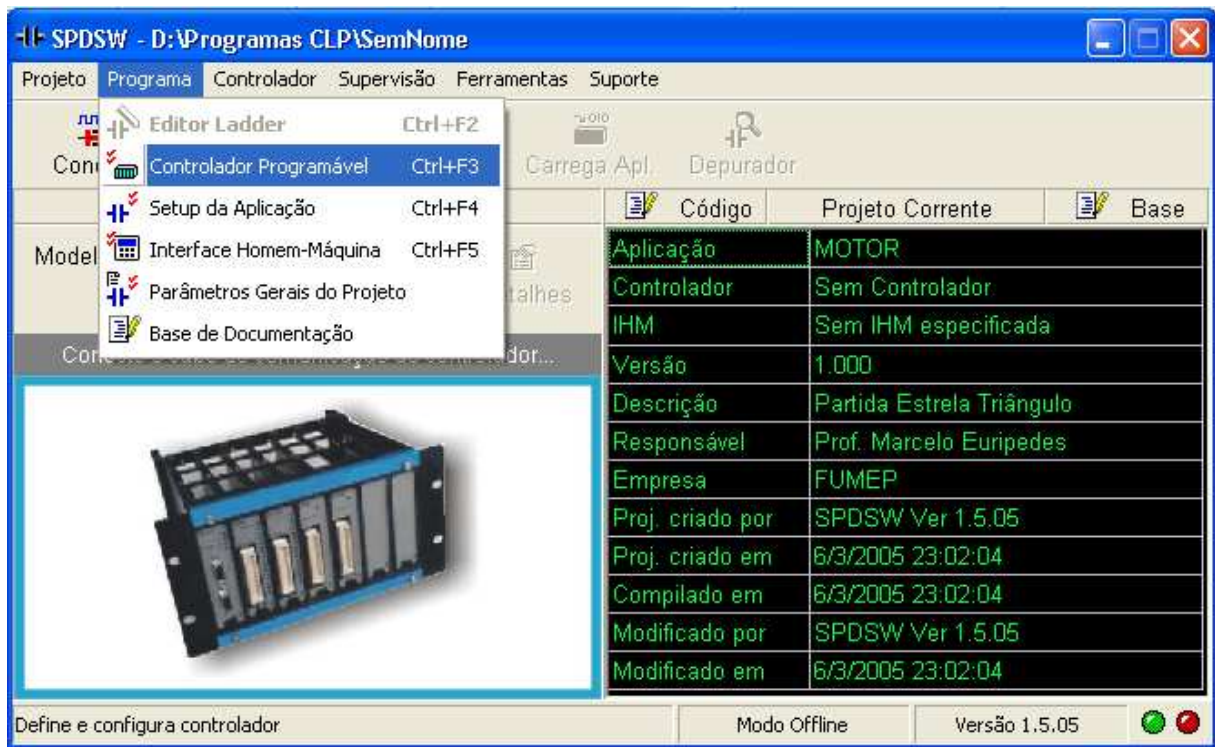
1) Clique na Guia Projeto → Novo



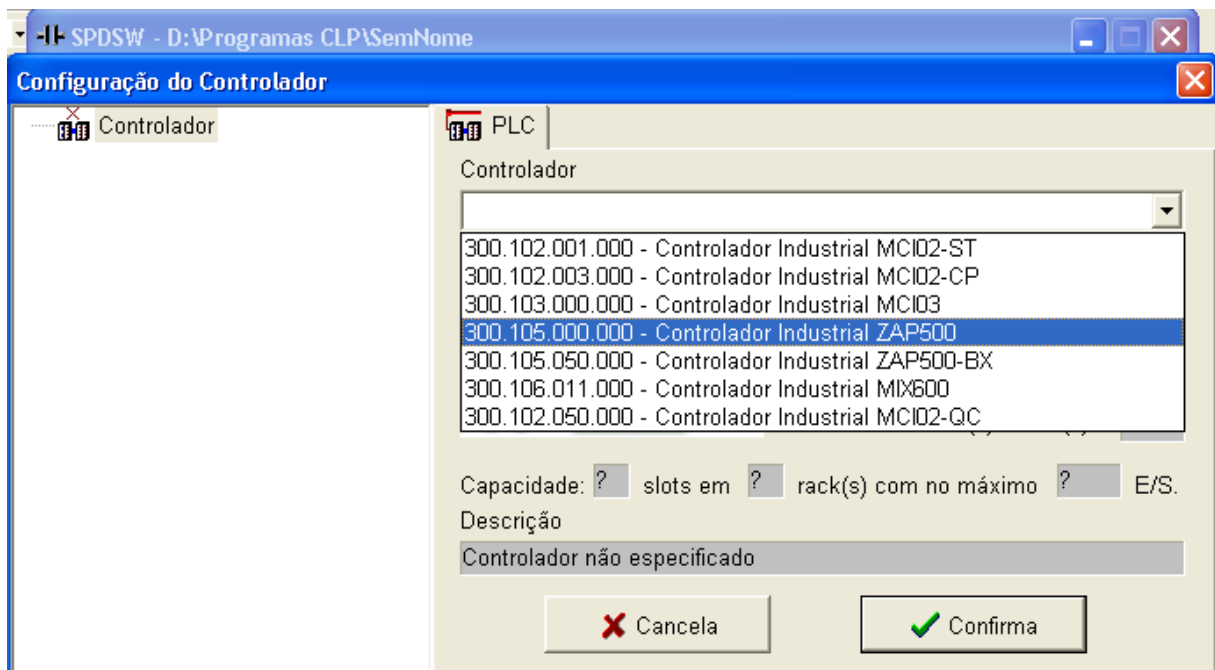
2) Preencha os parâmetros gerais do Projeto. Deixe na parte inferior: Nenhum projeto modelo



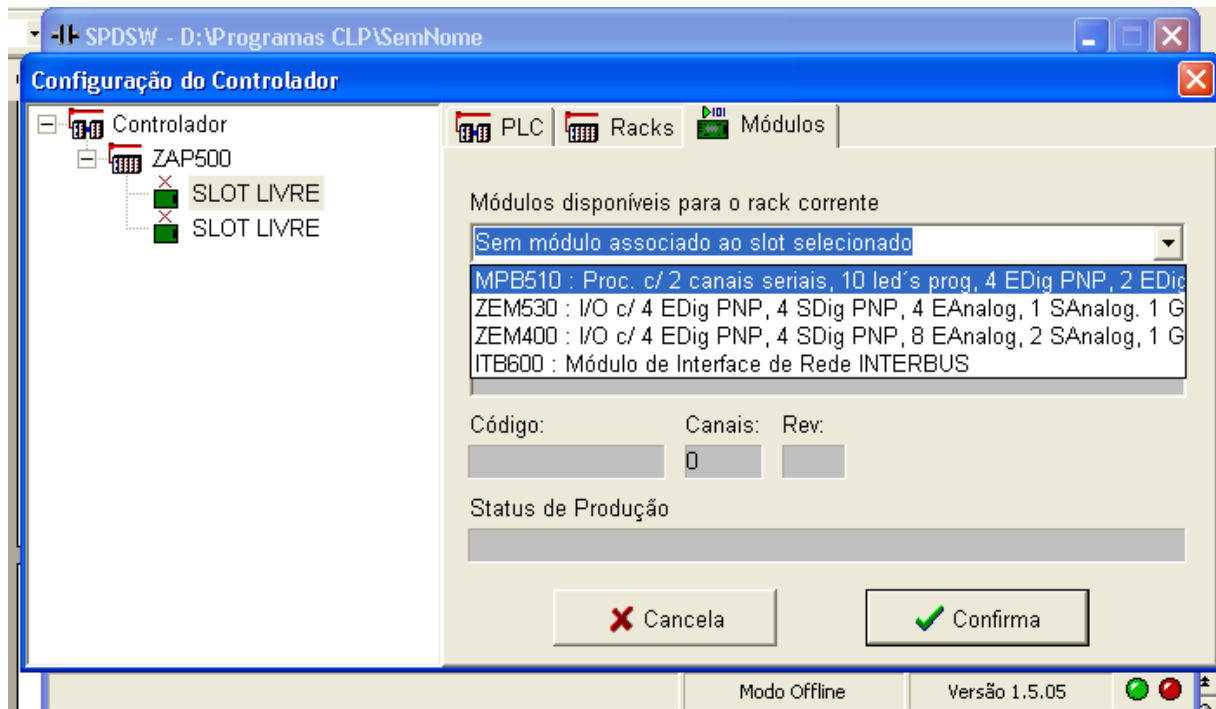
3) Clique na guia Programa → Controlador Programável



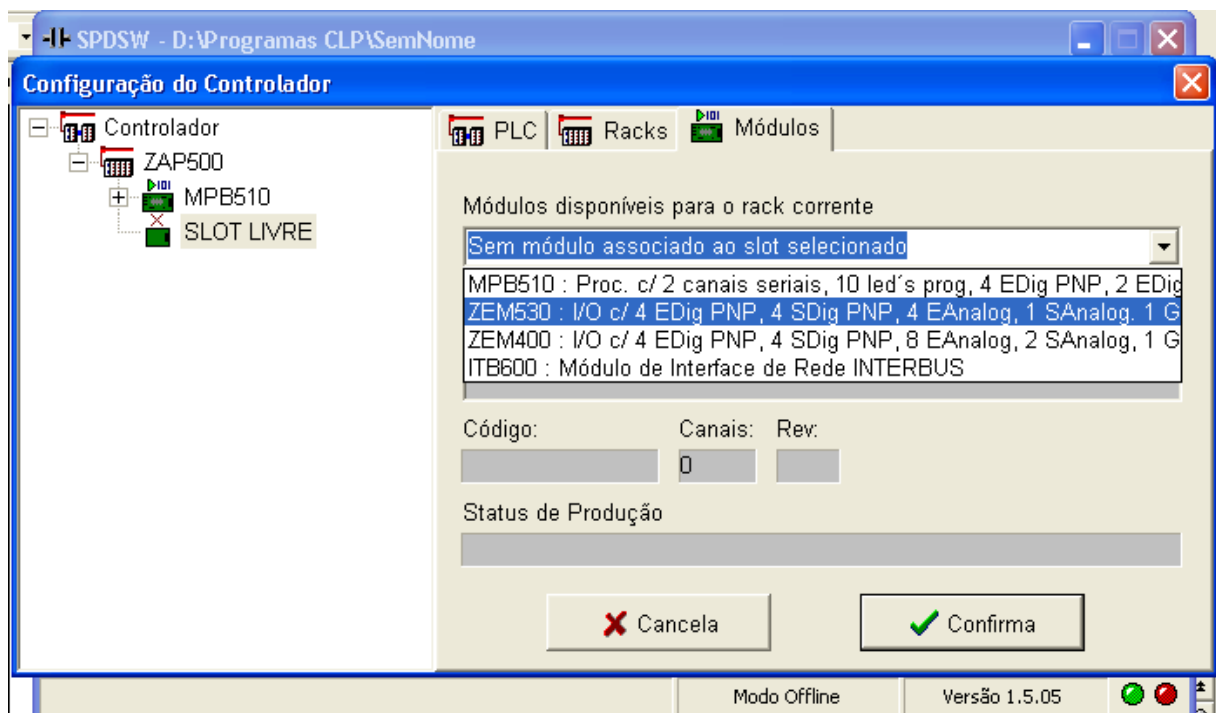
4) Clique em Controlador e depois no canto direito selecione Controlador Industrial ZAP500



5) Clique no primeiro Slot Livre e depois selecione a opção MPB510: Proc c 2 canais seriais...



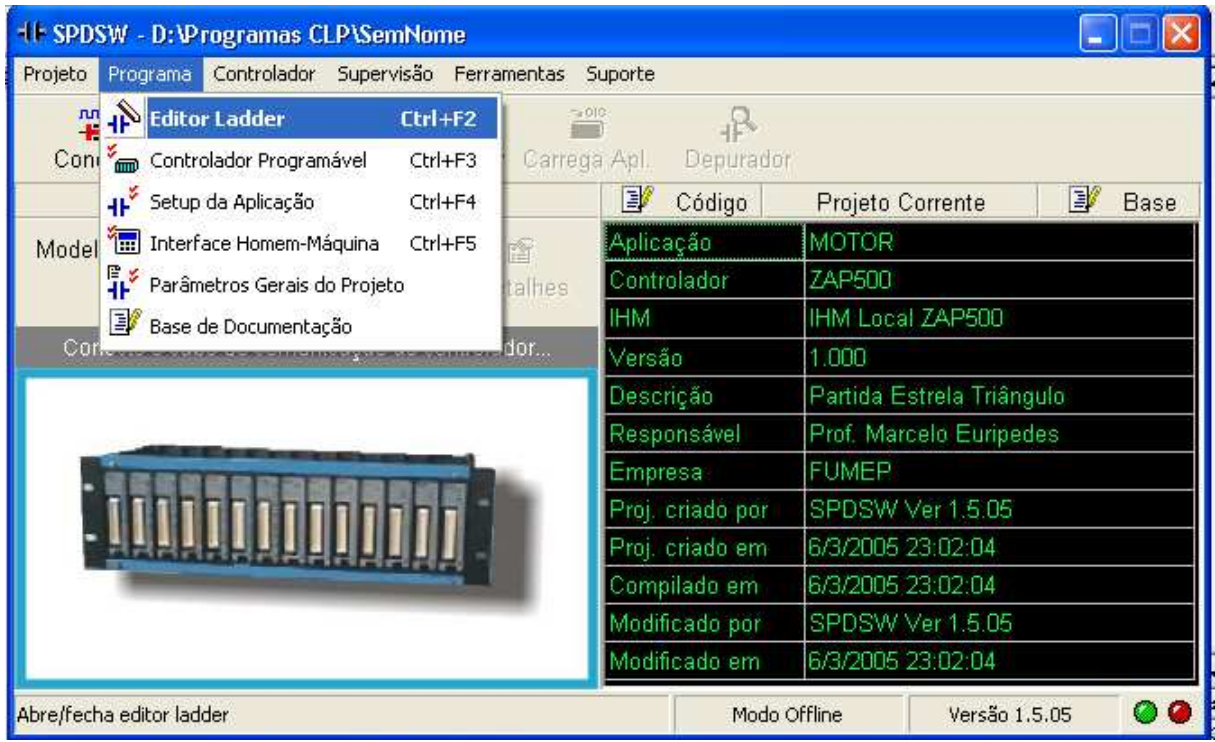
6) Clique no segundo Slot Livre e depois na opção ZEM530: I/O c/ 4 Edig PNP...
Clique em "SIM" e depois em "Confirma" no canto inferior da tela.



7) Clique em Programa → Interface Homem Máquina. Na tela que se abrir selecione a IHM local ZAP500. Depois clique em “Aplica” e logo após em “Fecha”.



8) Finalmente clique em Programa → Editor Ladder. Você está pronto para Inserir o seu programa no CLP HI ZAP 500.



**ESTA PÁGINA FOI DEIXADA EM BRANCO
INTENCIONALMENTE**