Unidade Operacional
(CENTRO DE FORMAÇÃO PROFISSIONAL “JOSÉ INÁCIO PEIXOTO”)

CONTROLADOR LÓGICO PROGRAMÁVEL

FIEMG
CIEMG
SESI
SENAI
IEL
Sistema FIEMG

Presidente da FIEMG
Robson Braga de Andrade

Gestor do SENAI

Petrônio Machado Zica

Diretor Regional do SENAI e

Superintendente de Conhecimento e Tecnologia

Alexandre Magno Leão dos Santos

Gerente de Educação e Tecnologia

Edmar Fernando de Alcântara

Elaboração

Geraldo Stocler

Unidade Operacional

CENTRO DE FORMAÇÃO PROFISSIONAL “JOSÉ INÁCIO PEIXOTO”

FIEMG
CIEMG
SESI
SENAI
IEL

Sistema FIEMG

Sumário

Apresentação

"Muda a forma de trabalhar, agir, sentir, pensar na chamada sociedade do conhecimento. "
Peter Drucker

O ingresso na sociedade da informação exige mudanças profundas em todos os perfis profissionais, especialmente naqueles diretamente envolvidos na produção, coleta, disseminação e uso da informação.

O **SENAI**, maior rede privada de educação profissional do país,sabe disso , e ,consciente do seu papel formativo , educa o trabalhador sob a égide do conceito da competência***: "formar o profissional com responsabilidade no processo produtivo, com iniciativa na resolução de problemas, com conhecimentos técnicos aprofundados, flexibilidade e criatividade, empreendedorismo e consciência da necessidade de educação continuada*.**"**

Vivemos numa sociedade da informação. O conhecimento , na sua área tecnológica, amplia-se e se multiplica a cada dia. Uma constante atualização se faz necessária. Para o **SENAI**, cuidar do seu acervo bibliográfico, da sua infovia, da conexão de suas escolas à rede mundial de informações – internet- é tão importante quanto zelar pela produção de material didático.

Isto porque, nos embates diários,instrutores e alunos , nas diversas oficinas e laboratórios do **SENAI**, fazem com que as informações, contidas nos materiais didáticos, tomem sentido e se concretizem em múltiplos conhecimentos.

O **SENAI** deseja , por meio dos diversos materiais didáticos, aguçar a sua curiosidade, responder às suas demandas de informações e construir *links* entre os diversos conhecimentos, tão importantes para sua formação continuada !

Gerência de Educação e Tecnologia

Introdução

Este material foi desenvolvido para servir de suporte instrucional em um de treinamento sobre Controladores Lógicos Programáveis (CLPs), integrante da grade curricular de cursos técnicos de eletrônica e informática industrial ou cursos para suprimento de demanda de profissionais da indústria. Ele aborda conceitos, recursos, aplicações, procedimentos e aspectos operacionais relacionados com a arquitetura e programação de CLPs.

O tema, por si só, já é vastíssimo e além disso, do ponto de vista prático, o mercado oferece muitas opções em termos de equipamentos e recursos. Dessa forma procuramos centrar nossas abordagens em aspectos comuns de algumas marcas e modelos disponíveis.

1 - Histórico do CLP

O Controlador Lógico Programável, ou simplesmente PLC *(Programmable Logic Controller),* pode ser definido como um dispositivo de estado sólido - um Computador Industrial, capaz de armazenar instruções para implementação de funções de controle (seqüência lógica, temporização e contagem, por exemplo), além de realizar operações lógicas e aritméticas, manipulação de dados e comunicação em rede, sendo utilizado no controle de Sistemas Automatizados

Os principais blocos que compõem um PLC são:

- CPU *(Central Processing Unit* - Unidade Central de Processamento): compreende o processador (microprocessador, microcontrolador ou processador dedicado), o sistema de memória (ROM e RAM) e os circuitos auxiliares de controle;

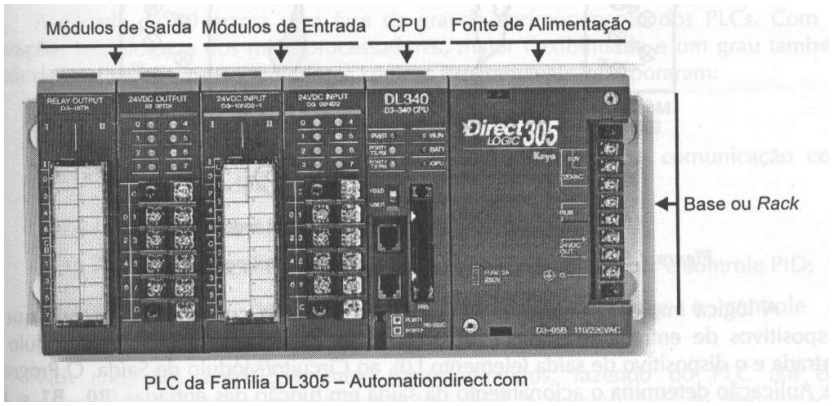
- Circuitos/Módulos de I/O *(InputlOutput* — Entrada/Saída): podem ser discretos (sinais digitais: 12VDC, 127 VAC, contatos normalmente abertos, contatos normalmente fechados) ou analógicos (sinais analógicos: 4-20mA, 0-10VDC, termopar);

- Fonte de Alimentação: responsável pela tensão de alimentação fornecida à CPU e aos Circuitos/Módulos de I/O. Em alguns casos, proporciona saída auxiliar (baixa corrente).

- Base ou *Rack:* proporciona conexão mecânica e elétrica entre a CPU, os Módulos de I/O e a Fonte de Alimentação. Contém o barramento de comunicação entre eles, no qual os sinais de dados, endereço, controle e tensão de alimentação estão presentes.

Pode ainda ser composto por Circuitos/Módulos Especiais: contador rápido (5kHz, 10kHz, 100kHz, ou mais), interrupção por hardware, controlador de temperatura, controlador PID, co-processadores (transmissão via rádio, posicionamento de eixos, programação BASIC, sintetizador de voz, entre outros) e comunicação em rede, por exemplo.

A figura a seguir mostra um PLC comercial.

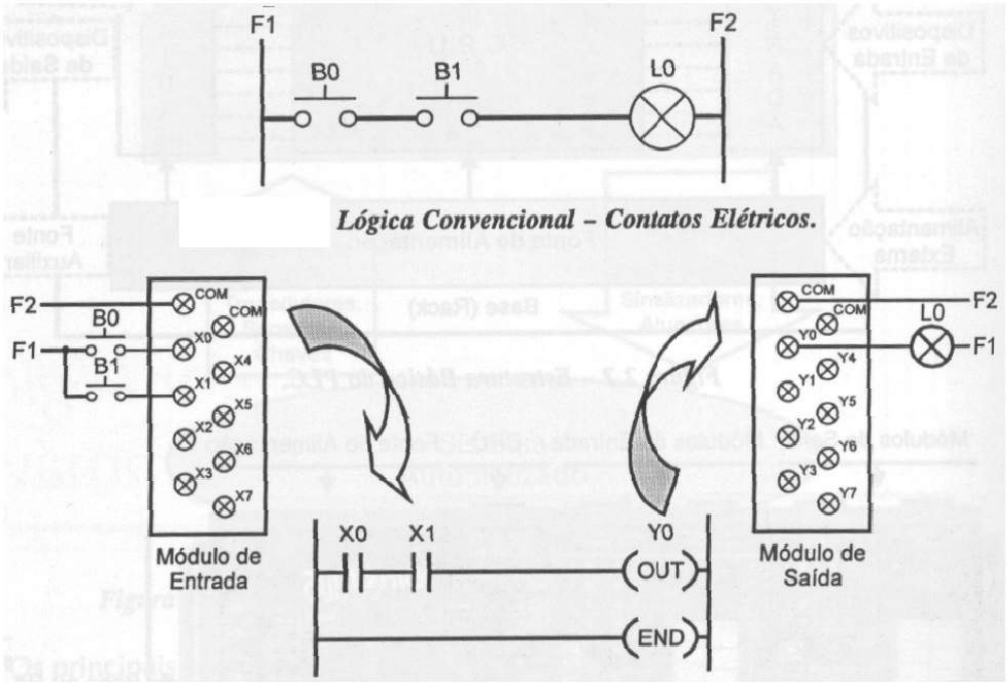


PLC da Família DL305 – Automationdirect.com

1.1 - Operação Básica do CLP

A CPU executa a leitura dos *status* (condições, estados) dos dispositivos de entrada meio dos Circuitos/Módulos de I/O. Esses *status* são armazenados na memória (RAM) para serem processados pelo Programa de Aplicação (desenvolvido pelo usuário e armazenado em memória RAM, EPROM ou EEPROM no PLC). Após a execução do Programa de Aplicação, o processador atualiza os *status* dos dispositivos de saída por meio dos Circuitos/Módulos de I/O, realizando a lógica de controle.

A programação do PLC é feita por meio de uma Ferramenta de Programação que pode ser um Programador Manual (Terminal de Programação, *Handheld Programmer),* ou um PC com Software de Programação específico (ambiente DOS® ou Windows®). A Linguagem Ladder *(RLL* - Relay *Ladder Logic,* Lógica de Contatos de Relê), muito popular entre os usuários dos antigos sistemas de controle a relês, é a mais utilizada. Esta linguagem é a representação lógica da seqüência elétrica de operação, como ilustrado nas figuras a seguir.



A lógica implementada pelo PLC é muito similar à convencional, sendo que os dispositivos de entrada (elementos B0 e B1) são conectados ao Circuito/Módulo de Entrada e o dispositivo de saída (elemento L0), ao Circuito/Módulo de Saída. O Programa de Aplicação determina o acionamento da saída em função das entradas (B0 . B1 = L0). Qualquer alteração desejada nesta lógica é realizada por meio de alterações no programa, permanecendo as mesmas ligações (conexões) nos Circuitos/Módulos de I/O.

1.2 - Histórico

Na década de 60, o aumento da competitividade fez com que a indústria automotiva melhorasse o desempenho de suas linhas de produção, aumentando tanto a qualidade como a produtividade. Fazia-se necessário encontrar uma alternativa para os sistemas de controle a relês. Uma saída possível, imaginada pela General Motors, seria um sistema baseado no computador.

Assim, em 1968 , a Divisão Hydramatic da GM determinou os critérios para projeto do PLC, sendo que o primeiro dispositivo a atender às especificações foi desenvolvido pela Gould Modicon em 1969.

As principais características desejadas nos novos equipamentos de estado sólido, com a flexibilidade dos computadores, eram:

- Preço competitivo com os sistemas a relês;
- Dispositivos de entrada e de saída facilmente substituíveis;
- Funcionamento em ambiente industrial (vibração, calor, poeira, ruídos);
- Facilidade de programação e manutenção por técnicos e engenheiros;
- Repetibilidade de operação e uso.

Inicialmente, os CLPs, ouPLCs eram chamados PCs - *Programmable Controllers,* mas com o advento dos Computadores Pessoais (PCs - *Personal Computers),* convencionou-se PLCs para evitar conflitos de nomenclatura. Originalmente os PLCs foram usados em aplicações de controle discreto *(onloff* - liga/desliga), como os sistemas a relês, porém eram facilmente instalados, economizando espaço e energia, além de possuírem indicadores de diagnósticos que facilitavam a manutenção. Uma eventual necessidade de alteração na lógica de controle da máquina era realizada em pouco tempo, apenas com 'mudanças' no programa, sem necessidade de alteração nas ligações elétricas.

A década de 70 marca uma fase de grande aprimoramento dos PLCs. Com as inovações tecnológicas dos microprocessadores, maior flexibilidade e um grau também maior de inteligência, os Controladores Lógicos Programáveis incorporaram:

1972 - Funções de temporização e contagem;

1973 - Operações aritméticas, manipulação de dados e comunicação com computadores;

1974 - Comunicação com Interfaces Homem-Máquina;

1975 - Maior capacidade de memória, controles analógicos e controle PID;

1979/80 - Módulos de I/O remotos, módulos inteligentes e controle de posicionamento.

Nos anos 80, aperfeiçoamentos foram atingidos, fazendo do PLC um dos equipamentos mais atraentes na Automação Industrial. A possibilidade de comunicação em rede (1981) é hoje uma característica indispensável na indústria. Além dessa evolução tecnológica, foi atingido um alto grau de integração, tanto no número de pontos como no tamanho físico, que possibilitou o fornecimento de minis e micros PLCs (a partir de 1982).

Atualmente, os PLCs apresentam as seguintes características:

- Módulos de I/O de alta densidade (grande número de Pontos de I/O por módulo);

- Módulos remotos controlados por uma mesma CPU;

- Módulos inteligentes (coprocessadores que permitem realização de tarefas complexas: controle PID, posicionamento de eixos, transmissão via rádio ou modem, leitura de código de barras);

- Software de programação em ambiente Windows® (facilidade de programação);

- Integração de Aplicativos Windows® (Access, Excel, Visual Basic) para comunicação com PLCs;

- Recursos de monitoramento da execução do programa, diagnósticos e detecção de falhas;

- Instruções avançadas que permitem operações complexas (ponto flutuante, funções trigonométricas);

- *Scan Time* (tempo de varredura) reduzido (maior velocidade de processamento) devido à utilização de processadores dedicados;

• Processamento paralelo (sistema de redundância), proporcionando confiabilidade na utilização em áreas de segurança;

• Pequenos e micros PLCs que oferecem recursos de hardware e de software dos PLCs maiores;

• Conexão de PLCs em rede (conexão de diferentes PLCs na mesma rede, comunicação por meio de Rede *Ethernet).*

O mercado recebe constantemente novos e melhores produtos que agregam valores, ao mesmo tempo que reduzem o custo das soluções baseadas em PLCs. Portanto, é indispensável uma atualização contínua por intermédio de contato com fabricantes e fornecedores, sendo a Internet uma ótima opção.

2 - Arquitetura do CLP

Conhecer a estrutura básica de cada Bloco que compõe o PLC, com suas particularidades e funções desempenhadas, auxilia na configuração e escolha do equipamento mais adequado à implementação de determinado Sistema Automatizado. De certa forma, influencia também no desenvolvimento do Programa de Aplicação.

2.1 - CPU - Unidade Central de Processamento

A CPU de um PLC compreende os elementos que formam a 'inteligência' do sistema: o Processador e o Sistema de Memória, além dos circuitos auxiliares de controle. O Processador interage continuamente com o Sistema de Memória por meio do Programa de Execução (desenvolvido pelo fabricante), interpreta e executa o Programa de Aplicação (desenvolvido pelo usuário), e gerência todo o sistema. Os circuitos auxiliares de controle atuam sobre os barramentos de dados *(data bus),* de endereços *(address bus)* e de controle *(control bus),* conforme solicitado pelo processador, de forma similar a um sistema convencional baseado em microprocessador.

2.2 - Processador

O desenvolvimento tecnológico de um PLC depende principalmente do Processador utilizado, que pode ser desde um microprocessador/controlada convencional - 80286, 80386, 8051, até um processador dedicado - *DSP (Digital Signa Processor* — Processador Digital de Sinais), por exemplo.

Atualmente, os Processadores utilizados em PLCs são dotados de alta capacidade computacional. Há CPUs que possuem processamento paralelo (sistema de redundância), no qual dois ou mais processadores executam o Programa de Aplicação, confrontando o resultados obtidas após o término de cada execução. Algumas Famílias de PLCs possuem Módulos Co-processadores, que auxiliam o Processador da CPU na execução de funções específicas (operações complexas).

Independente de sua tecnologia, o Processador é responsável pelo gerenciamento total do sistema, controlando os barramentos de endereços, de dados e de controle. Conforme determinado pelo Programa de Execução, interpreta e executa as instruções do Programa de Aplicação, controla a comunicação com dispositivos externos e verifica integridade de todo o sistema (diagnósticos). Pode operar com registros e palavras d instrução, ou de dados, de diferentes tamanhos (8, 16 ou 32 bits), determinado pelo tamanho de seu acumulador e pela lista de instruções disponíveis para cada CPU.

2.3 - Sistema de Memória

O Sistema de Memória da CPU é composto pela Memória do Sistema de Operação (Programa de Execução ou Firmware, e Rascunho do Sistema) e pela Memória de Aplicação (Programa de Aplicação e Tabela de Dados), conforme a figura a seguir.

Estrutura Básica do Sistema de Memória.

2.4 - Memória do Sistema de Operação

• Programa de Execução (Firmware): Constitui o programa desenvolvido pelo fabricante do PLC, o qual determina como o sistema deve operar, incluindo a execução do Programa de Aplicação, controle de serviços periféricos, atualização dos Módulos de I/O, etc. O Programa de Execução é responsável pela 'tradução' do Programa de Aplicação desenvolvido pelo usuário — em linguagem de alto nível, para instruções que o Processador da CPU possa executar — em linguagem de máquina. E armazenado em memória não volátil — tipo ROM, normalmente EPROM.

• Rascunho do Sistema: Trata-se de uma área de memória reservada para o armazenamento temporário de uma quantidade pequena de dados, utilizados pelo Sistema de Operação para cálculos ou controle (calendário e relógio internos, sinalizadores — flags — de alarmes e erros). Uma característica dessa área de memória é o acesso rápido, sendo do tipo RAM.

2.5 - Memória de Aplicação ou Memória do Usuário

• Programa de Aplicação: Nessa área é armazenado o programa desenvolvido pelo usuário para execução do controle desejado. Trata-se normalmente de memória EEPROM, podendo ser também EPROM, ou ainda RAM com bateria de segurança.

• Tabela de Dados: Essa área armazena dados que são utilizados pelo Programa de Aplicação, como valores atuais e de preset (pré-configurado) de temporizadores! Contadores e variáveis do programa, além dos status dos Pontos de Entrada e de Saída (Tabela de Imagem das Entradas e Tabela de Imagem das Saídas), que são lidas e escritas pelo Programa de Aplicação, respectivamente. A atualização desse status é realizada constantemente, refletindo as mudanças ocorridas nos Pontos de Entrada, e as atualizações das saídas são efetuadas pelo Programa de Aplicação. Cada Ponto de Entrada e de Saída, conectado aos Módulos de I/O, tem um endereço específico na Tabela de Dados, o qual é

acessado pelo Programa de Aplicação. Essa memória é do tipo RAM, podendo ser alimentada com bateria de lítio (memória retentiva).

Cada instrução que a CPU pode executar consome uma quantidade predeterminada de memória, expressa em bytes (8 bits) ou words (16 bits). Normalmente, as especificações técnicas de uma CPU indicam a quantidade de memória disponível para o usuário (memória variável - RAM, e memória de programação — EPROM, EEPROM ou RAM com bateria), podendo ser expressa em Kbytes ('capacidade física' de armazenamento da memória) ou em Kwords - palavras de programação ('capacidade lógica' de armazenamento da memória). No entanto, durante a configuração de um PLC, deve ser considerada a quantidade de palavras de programação, uma vez que nem sempre há relação direta entre a capacidade física (Kbytes) e a capacidade lógica (Kwords).

Conforme o fabricante e a Família (ou modelo) de PLC, a quantidade de memória destinada ao Programa de Aplicação pode ser configurada pelo usuário, ou seja, uma mesma CPU pode ser configurada para aceitar até 2Kwords de instruções, como até 4Kwords, por exemplo. Normalmente, quando existe esta possibilidade, a memória se apresenta na forma de cartuchos que são inseridos na CPU. Existem casos em que a CPU é fornecida com uma quantidade básica de memória, a qual pode ser expandida por meio desses "cartuchos".

Além da quantidade de memória, pode haver diferenças na forma de armazenamento dos dados. As características normalmente apresentadas nas especificações técnicas de unia CPU e que devem ser consideradas durante a sua configuração são:

- Capacidade de memória: quantidade máxima de memória que a CPU pode conter, sendo indicadas separadamente: Memória total para programa de aplicação e memória total para tabela de dados ou variáveis.

- Tipo de memória: forma de armazenamento do Programa de Aplicação. Algumas CPUs possibilitam a escolha do tipo de memória (EPROM ou EEPROM, por exemplo) para este fim.

- Bateria de backup: indica se a CPU permite utilização de bateria (de lítio) para manutenção da Tabela de Dados (Dados Retentivos), mesmo sem alimentação.

- Pontos de I/O total: quantidade máxima de Pontos de I/O que a CPU pode controlar. Conforme o caso, há limites para Pontos de Entrada e Pontos de Saída separadamente. Por exemplo, uma CPU pode controlar 640 Pontos de I/O, tendo no máximo 320 Pontos de Entrada e 320 Pontos de Saída.

- Tempo de processamento ou tempo de execução: tempo necessário para a CPU executar uma instrução booleana (contato ou bobina). Algumas CPUs podem apresentar tempo de execução para instruções booleanas relativamente alto, por serem indicadas ao processamento de operações mais complexas (operações aritméticas e trigonométricas). Pode ser expresso em 1 k de

instruções booleanas, incluindo, ou não, tempo de *overhead* (processamento executado pela CPU independente do Programa de Aplicação).

• Linguagem de programação: indica a(s) Linguagem(s) de Programação que pode ser utilizada. Apresenta o sistema operacional necessário para o Software de Programação para PC (DOS® ou Windows®, normalmente).

• Recursos de programação: indica os principais recursos disponíveis na CPU que podem ser utilizados. Por exemplo, pode apresentar a quantidade de temporizadores e contadores, operação com números inteiros ou números reais (ponto flutuante), rotinas internas para controle PID, existência de calendário/relógio internos, proteção por meio de senha (para acesso ao programa armazenado na memória) e sistema de diagnósticos, entre outros.

• Portas de comunicação: quantidade de portas de comunicação existentes na CPU, indicando tipo (RS-232 e/ou RS-422, por exemplo) e protocolos suportados.

Para casos em que a CPU apresenta-se como um módulo independente, deve-se considerar também o item *potência consumida da base,* o qual especifica a corrente que a CPU consome da Fonte de Alimentação, por meio do barramento da Base, para poder operar. Este valor é utilizado no Cálculo de Consumo de Potência durante a configuração do PLC.

2.6 - Circuitos/Módulos de I/O

A diferenciação de nomenclatura, Circuitos de I/O ou Módulos de I/O, deve-se ao tipo de PLC. No caso de PLCs Compactos — CPU e I/O alojados em um único invólucro, usa-se Circuitos de I/O. Para PLCs Modulares — CPU e I/O disponíveis de forma independente, usa-se Módulos de I/O. A partir deste ponto, é usado o termo Módulos de I/O indistintamente.

Os Módulos de I/O fazem a comunicação entre a CPU e o meio externo (por meio dos Dispositivos de Entrada e Saída), além de garantir isolação e proteção à CPU. De forma genérica, são divididos em Módulos de Entrada e Módulos de Saída. Para os PLCs modulares, há também os Módulos Combinados (Pontos de Entrada e de Saída no mesmo Módulo).

• Módulos de Entrada *(Input Modules):* recebem os sinais dos dispositivos de entrada, tais como: sensores, chaves e transdutores, e os convertem em níveis adequados para serem processados pela CPU.

• Módulos de Saída *(Output Modules):* enviam os sinais aos dispositivos de saída, tais como: motores, atuadores e sinalizadores. Esses sinais podem ser resultantes da lógica de controle, pela execução do Programa de Aplicação, ou podem ser 'forçados' pelo usuário, independente da lógica de controle.

Normalmente, os Módulos de I/O são dotados de:

• Isolação Óptica para proteção da CPU, Fonte de Alimentação e demais Módulos de I/O. Neste caso, não há conexão elétrica entre os dispositivos de entrada (chaves, sensores) ou de saída (atuadores, motores) e o barramento de comunicação da CPU.

• Indicadores de *Status* para auxílio durante a manutenção. Trata-se de LEDs *(Ligth Emitting Diodes* - Diodos Emissores de Luz) presentes na parte frontal dos Módulos de I/O que indicam quais Pontos de Entrada estão recebendo sinal dos dispositivos externos, e quais Pontos de Saída estão sendo atuados pela CPU. Há também a possibilidade de existirem indicadores de falhas, como, por exemplo, falta de alimentação externa, bloco de terminais desconectado, ou fusível interno queimado.

• Conectores Removíveis que reduzem o tempo de manutenção e/ou substituição dos Módulos de I/O, agilizando tais tarefas.

Os Módulos de I/O são classificados como Discretos (Digitais) ou Analógicos, existindo também os Especiais em algumas Famílias de PLCs.

Tratam sinais digitais *(on/off* - 0/1). São utilizados em sistemas seqüenciais e na maioria das aplicações com PLCs, mesmo como parte de sistemas contínuos.

Cada Ponto, de Entrada ou de Saída, dos Módulos Discretos corresponde a um *bit* de um determinado endereço da Tabela de Dados (Tabela de Imagem das Entradas e Tabela de Imagem das Saídas), a qual é acessada durante a execução do Programa de Aplicação.

A quantidade de pontos de um módulo determina sua densidade. Para os Módulos de Saída, quanto maior a densidade, menor a corrente que cada ponto pode fornecer.

2.7 - Módulos Discretos de Entrada

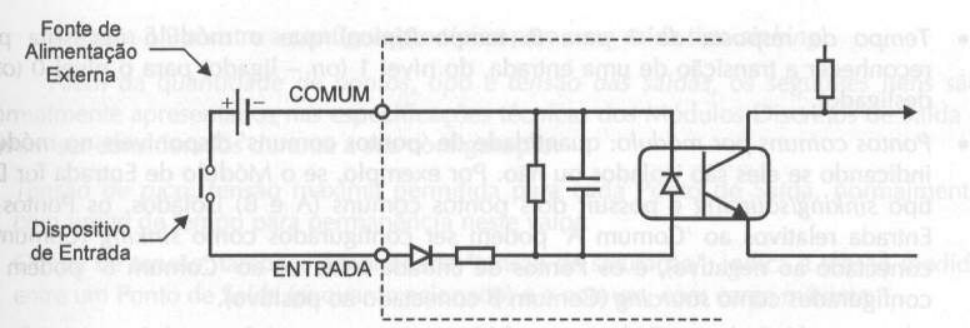
Os Módulos Discretos de Entrada normalmente apresentam as seguintes características:

• Filtros de sinal que eliminam problemas de *'bounces'* (pulsos indesejados, causados durante a abertura ou fechamento de contatos mecânicos - "rebatimentos".

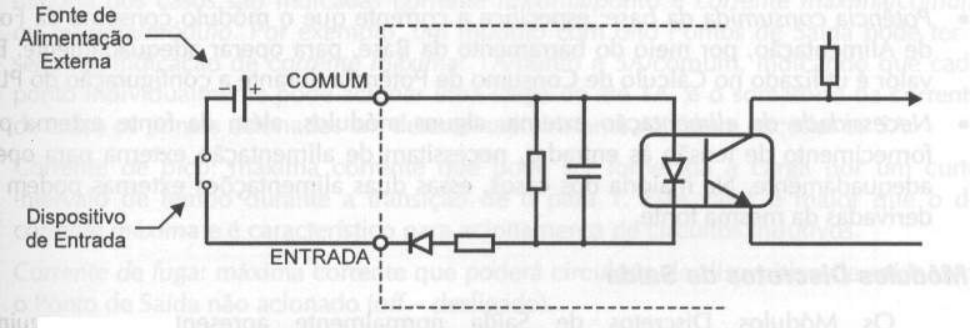
• Quantidade de pontos disponíveis: 8, 16, 32 ou 64.

• Tipo e faixa de tensão das entradas: AC (110V ou 220V), DC (12V, 24V ou 125V), AC/DC - *'either'* (12V, 24V, 110V), TTL ou 'contato seco'.

• As entradas DC podem ter configuração *current sinking* (consumidora de corrente - comum negativo), *current sourcing* (fornecedora de corrente - comum positivo) ou *current sinking/sourcing* (quando possuem um opto-acoplador com dois LEDs em anti-paralelo). Esta é uma característica determinante durante a configuração de um PLC, pois dependendo dos dispositivos de entrada utilizados (sensores NPN ou PNP, por exemplo), faz-se necessário optar por um ou outro tipo de entrada DC. Veja as figuras a seguir.



Configuração Típica de uma Entrada Tipo Sinking (Comum Negativo).



Configuração Típica de uma Entrada Tipo Sourcing (Comum Positivo).

Além da *quantidade de pontos, tipo* e *tensão das entradas,* os seguintes itens são normalmente apresentados nas especificações técnicas dos Módulos Discretos de Entrada e devem ser considerados durante a sua configuração:

• Tensão *máxima para nível O:* máxima tensão permitida para que o Módulo de Entrada reconheça como nível O *(off-* desligado).

• *Tensão mínima para nível 1:* mínima tensão necessária para que o Módulo de Entrada reconheça como nível 1 *(on* - ligado).

• *Tensão de pico:* máxima tensão permitida para cada Ponto de Entrada, normalmente com limite de tempo para permanência neste valor.

• *Corrente máxima em nível O:* máxima corrente que a entrada consome operando em nível 0.

• *Corrente mínima em nível 1:* mínima corrente necessária para que a entrada opere adequadamente em nível 1.

• *Corrente de entrada:* corrente típica de operação para uma entrada ativa (nível 1).

• *Impedância de entrada:* resistência que cada entrada representa para o dispositivo a ela conectado. Como esta não é linear, deve ser apresentada para algumas faixas de corrente.

• *Tempo de resposta de O para 1:* tempo (típico) que o módulo necessita para reconhecer a transição de uma entrada, do nível O *(off-* desligado) para o nível 1 (on- ligado).

• *Tempo de resposta de 1 para O:* tempo (típico) que *o* módulo necessita para reconhecer a transição de uma entrada, do nível 1 *(on* - ligado) para *o* nível O *(off-* desligado).

• *Pontos comuns por módulo:* quantidade de 'pontos comuns' disponíveis no módulo, indicando se eles são isolados ou não. Por exemplo, se o Módulo de Entrada for DC, tipo *sinkinglsourcing* e possuir dois pontos comuns (A e B) isolados, os Pontos de Entrada relativos ao 'Comum A' podem ser configurados como *sinking* (Comum A conectado ao negativo), e os Pontos de Entrada relativos ao 'Comum B' podem ser configurados como *sourcing* (Comum B conectado ao positivo).

• *Freqüência AC:* freqüência em que o módulo pode operar. Apenas para os Módulos de Entrada AC. Não se trata de freqüência de chaveamento (atuação) da entrada.

• *Potência consumida da base:* especifica a corrente que o módulo consome da Fonte de Alimentação, por meio do barramento da Base, para operar adequadamente. Este valor é utilizado no Cálculo de Consumo de Potência durante a configuração do PLC.

• *Necessidade de alimentação externa:* alguns módulos, além da fonte externa para fornecimento de tensão às entradas, necessitam de alimentação externa para operar adequadamente. Na maioria dos casos, essas duas alimentações externas podem ser derivadas da mesma fonte.

2.8 - Módulos Discretos de Saída

Os Módulos Discretos de Saída normalmente apresentam as seguintes características:

• Quantidade de pontos disponíveis: 4, 8, 12, 16, 32 ou 64.

• Tipo e faixa de tensão das saídas: AC - *triac* ou *scr* (24V, 110V ou 220V), DC -*transistor bipolar* ou *MOS-FET* (5V, 12V, 24V ou 125V) ou relê (AC e DC).

• As saídas DC podem ser tipo *sinking* (consumidora de corrente - comum negativo) ou *sourcing* (fornecedora de corrente - comum positivo).

• As saídas a relê podem ter contatos simples (um contato normalmente aberto), ou reversíveis (um contato normalmente aberto e outro normalmente fechado).

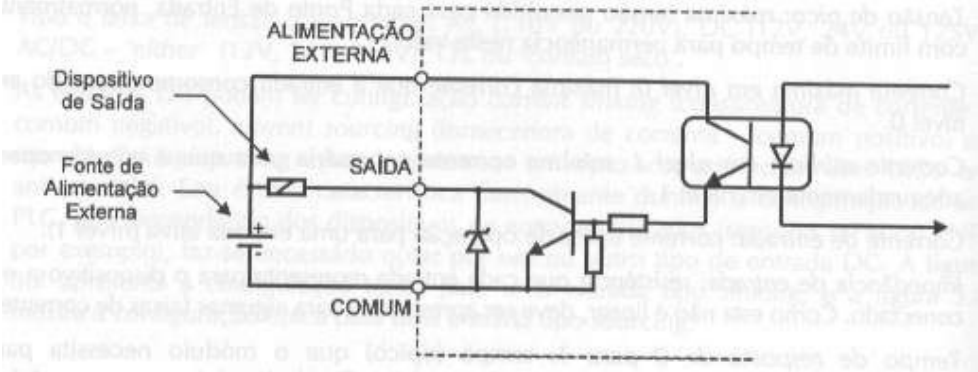
Além da *quantidade de pontos, tipo* e *tensão das saídas,* os seguintes itens são normalmente apresentados nas especificações técnicas dos Módulos Discretos de Saída e devem ser considerados durante a sua configuração:

• *Tensão de pico:* tensão máxima permitida para cada Ponto de Saída, normalmente com limite de tempo para permanência neste valor.

• *Queda de tensão:* também denominada “tensão de saturação", indica a tensão medida entre um Ponto de Saída (enquanto acionado) e o comum, com carga máxima.

• *Corrente máxima:* máxima corrente permitida para cada Ponto de Saída, normalmente indicada para cargas resistivas. Atenção especial deve ser dada a este item, pois na maioria dos casos são indicadas *corrente máxima/ponto* e *corrente máxima/comum* ou *máxima/módulo.* Por exemplo, um módulo com oito pontos de saída pode ter a seguinte indicação de *corrente máxima:* 1A/ponto e 5A/comum, indicando que cada ponto individualmente pode acionar uma carga de até 1A, e o somatório da corrente de todos os pontos acionados em determinado instante não deve exceder os 5A.

• *Corrente de pico:* máxima corrente que pode ser fornecida à carga por um curto intervalo de tempo durante a transição de O para 1. Este valor é maior que o de corrente *máxima* e é característico para acionamento de circuitos indutivos.



Configuração Típica de uma Saída Tipo Sinking (Comum Negativo) sem Fusível de Proteção.

• *Corrente de fuga:* máxima corrente que poderá circular pelo dispositivo de saída com o Ponto de Saída não acionado *(off* - desligado).

• *Carga mínima:* menor corrente que o Ponto de Saída deve fornecer à carga para operar adequadamente.

• *Tempo de resposta de 0 para 1:* tempo (típico) que o módulo necessita para realizar a transição de uma saída, do nível 0 *(off* - desligado) para o nível 1 *(on* - ligado).

• *Tempo de resposta de 1 para 0:* tempo (típico) que o módulo necessita para realizar a transição de uma saída, do nível 1 *(on* - ligado) para o nível 0 *(off* - desligado).

• *Pontos comuns por módulo:* quantidade de 'pontos comuns' disponíveis no módulo, indicando se eles são isolados ou não. Por exemplo, se for um Módulo de Saída a Relê e possuir dois pontos comuns (A e B) isolados, os Pontos de Saída relativos ao 'Comum A' podem ser configurados para operar com tensão DC, e os Pontos de Saída relativos ao 'Comum B' podem ser configurados para operar com tensão AC.

• *Freqüência AC:* freqüência em que o módulo pode operar. Apenas para os Módulos de Saída AC e Relê. Não se trata de freqüência de chaveamento (atuação) da saída.

• *Potência consumida da base:* especifica a corrente que o módulo consome da Fonte de Alimentação, por meio do barramento da Base, para operar adequadamente.

• *Necessidade de alimentação externa:* alguns módulos, além da fonte externa para fornecimento de tensão às saídas, necessitam de alimentação externa para operar adequadamente.

• *Fusíveis de proteção:* indica a existência ou não desses elementos, se são substituíveis e se estão localizados interna ou externamente ao módulo. Mesmo que os Módulos de Saída apresentem *fusíveis de proteção,* recomenda-se a utilização de proteção externa, por meio de fusíveis individuais para cada Ponto de Saída.

Outro fator importante durante a configuração dos Módulos de Saída relaciona-se ao acionamento dos dispositivos controlados. Não é recomendada a utilização de saídas a relê para *acionamentos cíclicos,* mesmo de baixa freqüência, ou acionamentos *rápidos,* devido à fadiga mecânica que eles podem sofrer.

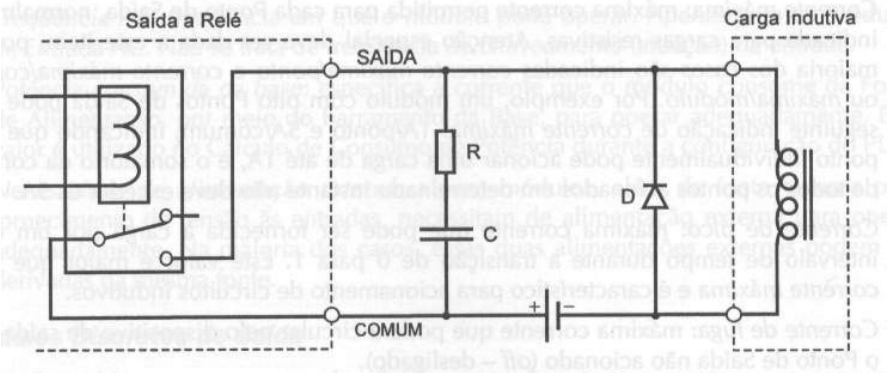
Porém, quando se utilizam saídas a relê para acionamento de cargas indutivas, recomenda-se a utilização de circuito RC - *snubber* (AC e DC) e diodo (apenas DC) para proteção dos contatos.

2.8 - Fonte de Alimentação

A Fonte de Alimentação desempenha importante papel na operação do sistema de um PLC. Além de fornecer todos os níveis de tensão para alimentação da CPU e dos Módulos de I/O, funciona como um dispositivo de proteção. Garante a segurança e a integridade da tensão de alimentação para todo o sistema, por meio do monitoramento constante dos níveis de tensão e de corrente fornecidos. Se esses níveis excederem os valores máximo ou mínimo permitidos, além do tempo especificado pelo fabricante, a fonte interage diretamente com o processador, gerando uma interrupção (por meio de uma seqüência de comandos) e fazendo com que a CPU pare a execução do Programa de Aplicação.

Atualmente, as Fontes de Alimentação dos PLCs utilizam tecnologia de chaveamento de freqüência (fontes chaveadas). Em alguns casos, a tensão de entrada não é fixa e nem selecionável pelo usuário, possuindo ajuste automático, proporcionando maior versatilidade e qualidade ao sistema. Há, também, Fontes de Alimentação com tensão de entrada DC (12V, 24Vou 125V) para aplicações específicas (automotivas, por exemplo).

As proteções externas recomendadas para a Fonte de Alimentação dos PLCs variam conforme o fabricante, mas basicamente consistem em transformadores de isolação ou supressores de ruídos para rede, aterramento adequado e conformidade com as normas técnicas locais.



Proteções Recomendadas para Saídas a Relés.

Em alguns casos, os Módulos de I/O necessitam, além das tensões fornecidas pela Fonte do PLC, de alimentação externa. A Fonte do PLC é responsável pela alimentação do circuito lógico dos Módulos de I/O, sendo que a fonte externa alimenta os circuitos de potência, ou circuitos externos - entrada ou saída (Módulos Discretos e Analógicos) ou ainda fornece um nível de tensão com maior capacidade de corrente para os Módulos Especiais.

Normalmente, as Fontes dos PLCs proporcionam saída auxiliar de tensão em 24VDC, com limite reduzido de corrente (na faixa de 300mA a 800mA). Essa saída pode ser utilizada para alimentação dos Módulos de I/O, desde que respeitado o limite de corrente.

A Fonte de Alimentação tem aspectos variados, conforme o fabricante e a Família de PLC. Pode apresentar-se em conjunto com a CPU, ou como um Módulo independente para ser conectado à Base, ou ainda ser parte integrante da própria Base.

As características normalmente apresentadas nas especificações técnicas de uma Fonte de Alimentação e que devem ser consideradas durante a sua configuração são:

• *Faixa da tensão de entrada:* AC (85-132V, 170-264V, 85-264V, por exemplo), DC (12V, 24V, 10-28V, 125V, por exemplo). Para as faixas de entrada em tensão DC observar também o *ripple* máximo permitido, geralmente menor que 10%.

• *Seleção da faixa de entrada: automática, por jumpers, ou por terminais de conexão.*

• *Potência fornecida: máxima corrente fornecida ao barramento da Base, normalmente relacionada à tensão de 5VDC, para alimentação dos Módulos de I/O e da CPU, se for o caso (CPU como módulo independente). Este valor é utilizado no Cálculo de Consumo de Potência durante a configuração do PLC.*

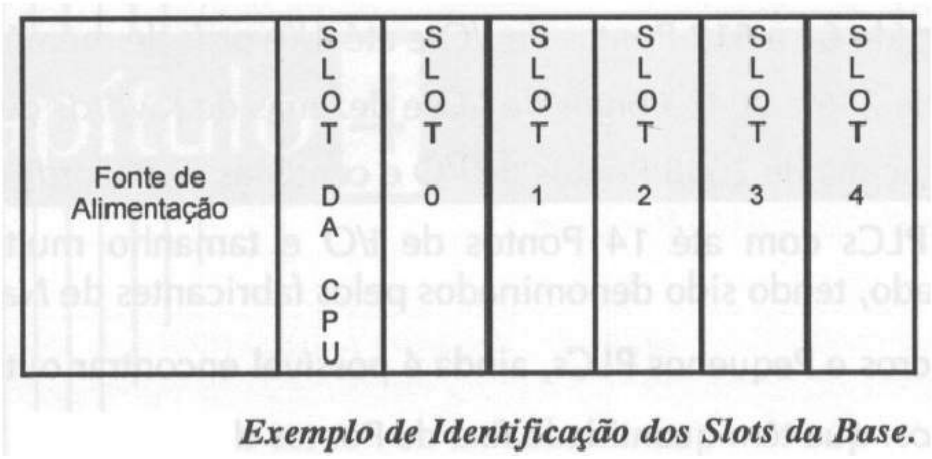
• *Saída auxiliar de 24VDC:* apresenta as características (tensão, corrente e *ripple)* da saída auxiliar de 24VDC. Apenas para fontes com alimentação AC.

2.9 - Base ou *Rack*

A Base, ou *Rack,* é responsável pela sustentação mecânica dos elementos que compõem o PLC. Contém o barramento que faz a conexão elétrica entre eles, no qual estão presentes os sinais de dados, endereço e controle - necessários para comunicação entre a CPU e os Módulos de I/O, além dos níveis de tensão fornecidos pela Fonte de Alimentação - necessários para que a CPU e os Módulos de I/O possam operar.

Cada posição da Base, possível de receber um Módulo de I/O ou a CPU - quando esta se apresentar como módulo independente, é denominada de *slot* (ranhura, abertura), e cada *slot* da Base tem uma identificação própria, conforme o fabricante. Por exemplo, a *Automationdirect.com* utiliza a seguinte nomenclatura para os *slots* da Base:

Nas Famílias em que a CPU apresenta-se como um módulo independente (Famílias DL205 e DL305), o primeiro slot ao lado da Fonte de Alimentação, denomina-se *slot da CPU,* não podendo ser ocupado por Módulos de I/O. Em casos específicos de Controle Baseado em PC, pode ser ocupado por Módulos Especiais de Comunicação (Módulo para Comunicação *Ethernet,* por exemplo). O primeiro *slot* ao lado da CPU denomina-se *slot 0,* o seguinte *slot 1,* e assim sucessivamente, conforme apresenta a figura a seguir.



Exemplo de Identificação dos Slots da Base.

Alguns Módulos de I/O ou Especiais podem ter restrições quanto ao posicionamento nos *slots* da Base. Porém, de forma geral, os Módulos Discretos e Analógicos podem ser posicionados livremente pelo usuário. As possíveis restrições de posicionamento são indicadas nos respectivos manuais técnicos.

Na maioria dos casos, uma mesma Família de PLC possui Bases com diferentes quantidades de *slots,* com o objetivo de atender às necessidades específicas de cada

2.10 - Classificação dos PLCs

Embora existam algumas divergências entre autores e fabricantes quanto aos critérios de classificação, os PLCs podem ser divididos em grupos específicos de acordo com a estrutura que apresentem (especificamente relacionada à quantidade de Pontos de I/O que a CPU pode controlar e a quantidade de memória de programação disponível):

- *Micros PLCs* (até 64 Pontos de I/O e até 2Kwords de memória)

- *Pequenos PLCs* (de 64 a 512 Pontos de I/O e até *4Kwords* de memória)

- *PLCs Médios* (de 256 a 2048 Pontos de I/O e dezenas de *Kwords* de memória)

- *PLCs* Grandes (acima de 2048 Pontos de I/O e centenas de *Kwords* de memória)

Em 1997, PLCs com até 14 Pontos de I/O e tamanho muito reduzido foram lançados no mercado, tendo sido denominados pelos fabricantes de *Nanos PLCs.*

Entre os Micros e Pequenos PLCs, ainda é possível encontrar outra divisão:

- *PLCs Compactos:* que têm quantidade fixa de Pontos de I/O.

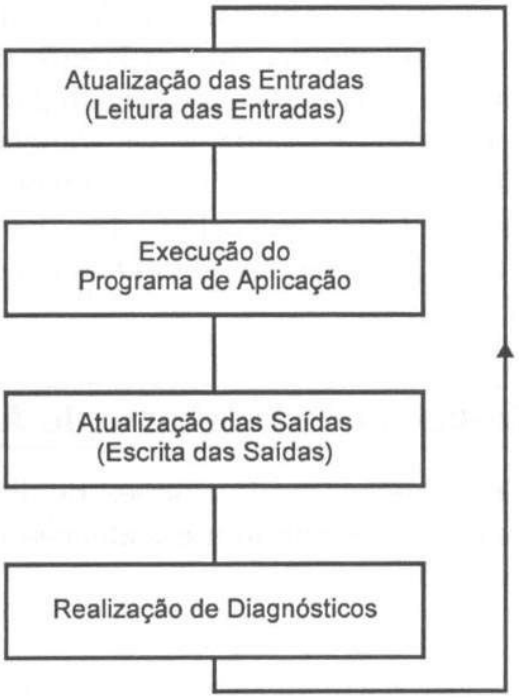
• *PLCs Modulares:* que permitem a configuração, por parte do usuário, da quantidade e combinação dos Pontos de I/O.

Em alguns PLCs Compactos, é possível a adição de Pontos de I/O por meio de 'blocos' de expansão, com limite determinado pelo fabricante, porém apresentam poucas opções de configuração (quantidade e tipo dos Pontos de I/O para cada bloco de expansão).

3 - Princípio de operação do CLP

3.1 - Ciclo de Execução do PLC

O Scan, que é o tempo de execução de um ciclo do PLC em modo de execução, pode ser descrito resumidamente pelo fluxograma apresentado na figura a seguir. Estes segmentos estão presentes em todos os PLCs disponíveis no mercado e definem o tratamento da informação *durante* a execução do Programa de Aplicação.



Fluxograma Básico do Sistema de Operação de um PLC (Modo de Execução).

A seguir, são descritos com mais detalhes os principais segmentos do fluxograma do sistema de operação do PLC.

3.2 - Atualização das Entradas - *Leitura das Entradas*

A CPU realiza a leitura de todos os pontos de entrada e armazena-os na tabela de imagem das entradas. Cada ponto de entrada corresponde a uma posição de memória específica (um *bit* de uma determinada *word).*

A tabela de imagem das entradas é acessada pela CPU durante a execução do programa de aplicação.

Após a execução deste segmento em um determinado *scan,* a Leitura das entradas será realizada apenas no *scan* seguinte, ou seja, se o *status* (condição) de um determinado ponto de entrada mudar após a leitura das entradas, ele só terá influência na execução do programa de aplicação no *scan*

seguinte, quando será percebida tal alteração.

Se uma determinada aplicação não puder 'esperar' este tempo (normalmente, da ordem de milisegundos) para reconhecimento da alteração dos pontos de entrada, utilizam-se instruções imediatas para construção da lógica de controle no programa de aplicação. Essas instruções acessam diretamente os pontos de entrada no momento em que são executadas. Há também as instruções imediatas de saída que, ao serem executadas, atualizam os pontos de saída e a tabela de imagem das saídas simultaneamente. A utilização de instruções imediatas aumenta o *Scan Time* (tempo de varredura, ou de execução) da CPU, pois além das operações de atualização das entradas e atualização das saídas, os módulos de I/O são acessados a cada execução de uma instrução imediata.

3.3 - Execução do Programa de Aplicação

Neste segmento, a CPU executa as instruções do Programa de aplicação, que definem a relação entre a condição das entradas e a atuação das saídas, ou seja, definem a lógica de controle a ser realizada.

A CPU inicia a execução do programa de aplicação a partir do primeiro *degrau* (Lógica de controle da linguagem ladder), executando-o da esquerda para a direita, e de cima para baixo, *rung a rung,* até encontrar a instrução END (FIM). Constrói, assim, uma nova tabela de imagem das saídas, gerada a partir da lógica executada.

3.4 - Atualização das Saídas - Escrita das Saídas

Após a execução do programa de aplicação, o conteúdo da Tabela de imagem das saídas, construída de acordo com a lógica executada, é enviado aos pontos de saída correspondentes.

3.5 - Realização de Diagnósticos

Neste segmento, a CPU realiza todos os diagnósticos do sistema, além de calcular o *Scan Time* (Tempo de varredura), atualizar Relês Especiais correspondentes e reinicializar o *Watchdog Timer* (Temporizador 'Cão-de-Guarda').

Entre os diagnósticos realizados, os mais importantes são o cálculo do Scan Time e o controle do *Watchdog Timer.* O Scan *Time* compreende o tempo consumido pela CPU para realizar todas as tarefas em cada *scan,* desde o início (atualização das entradas) até o término do ciclo (atualização das saídas). O *Watchdog Timer* armazena o tempo máximo permitido para execução de cada scan (normalmente definido pelo usuário). Se, em determinado *scan,* esse tempo for excedido (Erro Fatal), a CPU é forçada ao modo de programação e todas as saídas são desligadas. Caso contrário, o valor do Scan *Time é* armazenado em uma variável apropriada (para realização de estatísticas: Scan Time máximo e

mínimo, por exemplo) e juntamente com o *Watchdog Timer* é reinicializado, sendo controlados a cada *scan.*

Todos os erros diagnosticados, Fatais ou não Fatais, são indicados por *flags* (bits internos à CPU, que podem ser usados no programa de aplicação), e em alguns casos por LEDs externos (normalmente localizados na parte frontal da CPU e dos Módulos de I/O). Algumas CPUs dispõem, também, de uma variável destinada ao armazenamento do código de erro ocorrido durante a execução do último *scan.*

3.6 - Considerações Relacionadas ao *Scan Time*

Como apresentado, o scan do PLC é composto por diversos segmentos nos quais são realizadas tarefas específicas (determinadas pelo firmware). Para execução de cada segmento é consumida uma certa quantidade tempo, sendo que o somatório dos tempos determina o Scan *Time* (Tempo de varredura) o qual pode variar de um scan para outro.

Os fatores que têm influência direta sobre o Scan *Time* são:

- Quantidade de módulos e pontos de entrada (‘atualização das entradas’);

- Conexão de dispositivos(s) periférico(s) (‘atendimento a serviço periférico’);

- Tamanho do programa de aplicação e tipo das instruções utilizadas (‘execução do programa de aplicação’);

- Quantidade de módulos e pontos de saída (‘atualização das saídas’).

Independente da complexidade do programa de aplicação, há certos fundamentos da programação em linguagem Ladder que são imprescindíveis para um desenvolvimento adequado, os quais são válidos *genericamente a todos os PLCs.*

4 - Linguagens de Programação

A primeira linguagem criada para programação de PLCs foi a Linguagem Ladder.

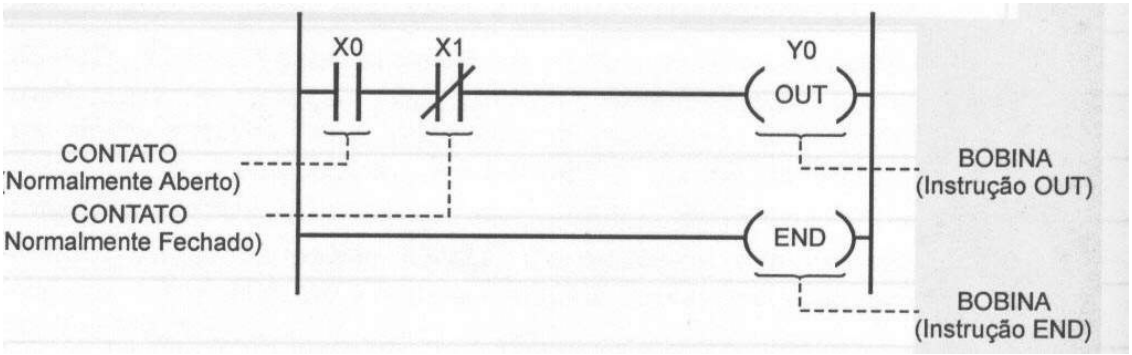
O fato de ser uma linguagem gráfica, baseada em símbolos semelhantes aos encontrados nos esquemas elétricos - contatos e bobinas, foi determinante para aceitação do PLC por técnicos e engenheiros acostumados com os sistemas de controle a relês. Provavelmente é ainda a mais utilizada.

Enquanto a Linguagem Ladder conquistava os Estados Unidos, a Linguagem de Lista de Instruções era amplamente difundida na Europa. Esta, por sua vez, é uma linguagem textual semelhante ao Assemble, e faz parte das linguagens básicas normalmente disponíveis em um PLC.

As Linguagens de Programação não se limitam apenas a estas duas. Atualmente, são encontrados no mercado PLCs que proporcionam programação por meio de Linguagem 'C' e BASIC, por exemplo. A Norma IEC 61131-3 define cinco Linguagens de Programação, entre as quais estão a Linguagem Ladder e a Linguagem de Lista de Instruções.

4.1 - Linguagem_Ladder *(LD - Ladder Diagram)*

O nome Ladder deve-se ã representação da linguagem se parecer com uma escada *(ladder),* na qual duas barras verticais paralelas são interligadas pela lógica de controle *(rung),* formando os degraus da escada. A figura a seguir apresenta um exemplo simples de programação em Linguagem Ladder.



Atualmente, os PLCs apresentam instruções sofisticadas. Além de simples contatos e bobinas, dispõem de contatos para detecção de borda de subida/descida (one *shot* —'disparo'), contatos de comparação, temporizadores, contadores, blocos de processamento (operações lógicas e aritméticas, manipulação de dados), controle total do fluxo de execução do programa *(foops For/Next, Goto, Stop,* sub-rotinas), interrupções (por hardware e por software) e blocos para manipulação de mensagens (ASCII, rede), por exemplo.

4.2 - Linguagem de Lista de Instruções *(IL - Instruction List)*

É uma linguagem de baixo nível, similar à assembler. Nessa Linguagem é permitida apenas uma operação por linha, como o armazenamento

de um valor em uma determinada variável, por exemplo. Sua utilização é viável em aplicações menores, ou para otimização de partes de uma aplicação mais complexa. A figura a seguir apresenta um exemplo simples de Programação em Linguagem de lista de Instruções.

Endereço	Instrução	Operando
0	STR	I0
1	ANDN	I1
2	OUT	O0
3	END	

4.3 - Ferramentas para Programação de PLCs

As principais Ferramentas para Programação disponíveis atualmente para as Famílias de PLCs encontrados no mercado são o Programador Manual (Handheld Programmer) e o Software de Programação para PC. Ambas ferramentas possuem recursos para monitoramento de condições internas à CPU (diagnósticos e erros, por exemplo), verificação da execução do Programa de Aplicação e controle sobre os Modos de Operação, entre outros.

Cada fabricante, e em alguns casos cada Família de PLC, tem suas Ferramentas de Programação próprias que não podem ser usadas para PLCs (ou CPUs) distintos.

Programador Manual *(Handheld Programmer)*

Esta é a ferramenta de menor custo e utilizada para pequenas alterações. Normalmente, possui um display de cristal líquido com duas linhas para apresentação das informações (endereço e dados do programa, condição dos Pontos de I/0 e diagnósticos internos, por exemplo) e um teclado de membrana para entrada dos dados.

O Programador Manual não é indicado para o desenvolvimento de todos os programas de aplicação, pois permite edição/alteração por meio de mnemônicos (Linguagem de Lista de Instruções) apenas. Porém, é bastante útil como ferramenta de manutenção para campo (‘chão de fábrica’, proporcionando visualização, monitoramento e alteração de parâmetros e do programa de aplicação rapidamente, com a vantagem de ser portátil e resistente ao ambiente industrial. É conectado à CPU do PLC por meio de cabo apropriado, pelo qual recebe a tensão de alimentação necessária à sua operação.

Algumas famílias de Micros PLCs permitem o desenvolvimento de programas apenas por intermédio dessa ferramenta de programação. Conforme o fabricante o *backup* (cópia de segurança) do programa de aplicação desenvolvido pode ser armazenado em cartões de memória tipo PCMCIA, ou em memórias tipo EEPROM, ambos instalados no próprio Programador Manual.

Software de Programação

É a Ferramenta mais poderosa disponível atualmente. Conforme o PLC, o Software de Programação opera em ambiente DOS® ou Windows®, sendo este o mais comum. Além de proporcionar edição/alteração do Programa de aplicação em ambiente gráfico (Linguagem Ladder, por exemplo) - mesmo para as versões DOS permite visualização e controle total do sistema; documentação e impressão da aplicação desenvolvida; várias formas de armazenamento de *backup* (disquete, HD, CD, etc.); e recursos avançados para depuração e manutenção. O PC deve atender ás configurações de hardware (processador, quantidade de memória RAM, espaço livre em HD, portas seriais) e de software (Sistema Operacional) indicadas pelo fabricante do PLC.

Conforme o Software de Programação, são disponíveis dois modos de operação:

- Offline (Sem Conexão): permite o desenvolvimento do programa de aplicação (edição, documentação, impressão) e configuração de parâmetros sem necessidade de conexão com a CPU do PLC.

- *Online* (Conectado): os recursos são disponíveis a partir da conexão com a CPU do PLC. Alguns Softwares de Programação permitem operação apenas neste modo, ou seja, todo o desenvolvimento deve ser realizado com o PC conectado ao PLC.

A comunicação entre o PC e a CPU do PLC é feita por meio de cabo apropriado, pela porta serial (RS-232) na maioria dos casos. Porém, algumas CPUs utilizam o padrão RS-422 e necessitam de conversor RS-232/RS-422 para conexão. Há ainda aquelas que utilizam padrão próprio e necessitam de interface dedicada instalada no PC.

Os recursos e facilidades que o Software de Programação oferece variam conforme o fabricante. Por exemplo, o Software de Programacão DirectSOFT da *Automationdirecr.com* opera em ambiente Windows (com versões para 16 e 32 bits), proporcionando nos modos *Offline* e *Online* poderosos recursos de edição, documentação e depuração/manutenção. Por utilizar plataforma Windows , permite a visualização de várias janelas simultaneamente, possibilitando que dois ou mais programas de aplicação sejam criados/editados ao mesmo tempo, e recursos de 'Marcar, Recortar, Colar' sejam utilizados entre eles. A comunicação com a CPU do PLC pode ser feita por porta serial padrão RS-232, ou por Modem, com busca e configuração automáticas em ambos os casos. Para comunicação via Modem - que permite a manutenção, alteração e atualização de aplicações a distância, são necessários dois Modems: um instalado no PC (interno ou externo) e outro instalado no PLC (externo), ambos configurados adequadamente.

5 - Soft RSLogix 500

5.1 - CLP SLC-500 da Allen Bradley

O SLC-500 é um controlador de estrutura modular básico, consiste de um chassi, fonte de alimentação, processador (CPU), Entrada/Saída (Módulos E/S). Possui características que anteriormente, só poderiam ser encontradas em controladores de grande porte. Possui a flexibilidade e a potência de um controlador de grande porte com o tamanho e a simplicidade de um de pequeno porte. O SLC-500 oferece mais opções de controle do que qualquer outro controlador programável de sua classe.

O chassi armazena o controlador e os módulos de E/S. A fonte de alimentação localiza-se no lado esquerdo do chassi. Todos os componentes se deslizam facilmente para dentro do chassi ao longo das guias. Não é necessário o uso de ferramentas para inserir ou remover o controlador ou os módulos de E/S. Podem ser conectados em um SLC até três chassis (30 ranhuras de E/S). Existem quatro tamanhos de chassis: 4 ranhuras, 7 ranhuras, 10 ranhuras e 13 ranhuras.

Os controladores de estrutura modular SLC-500 são projetados para atender desde aplicações independentes até grandes sistemas distribuídos e de aplicações simples até as mais complexas.

Recursos do controlador

Tamanho da memória – A memória do controlador de estrutura modular SLC-500 pode ser configurada tanto para armazenamento de dados quanto para armazenamento de programa. O tamanho da memória varia de 1K a 64K.

Pontos de E/S – O controlador SLC 5/01 suporta o endereçamento de até 3940 pontos de E/S. Os SLC 5/02, SLC 5/03, SLC 5/04 e SLC 5/05 suportam um endereçamento de 4096 pontos de E/S. Os controladores de estrutura modular SLC-500 são suportador por mais de 60 módulos de E/S diferentes, incluindo E/S digital e E/S inteligente.

Performance – Os controladores de estrutura modular SLC-500 são projetados tendo em vista o rendimento. O tempo de varredura do programa, para uma mistura típica de instruções, varia de 0,9 ms/K a 8,0 ms/K, dependendo do controlador. O tempo de varredura da E/S varia de 0,25ms a 2,6ms, dependendo do controlador.

5.1 - RSLogix 500

O software RSLogix 500 é um programa desenvolvido pela Rockwell Software para editar programas de aplicação dos CLPs da família SLC-500. Através dele é possível:

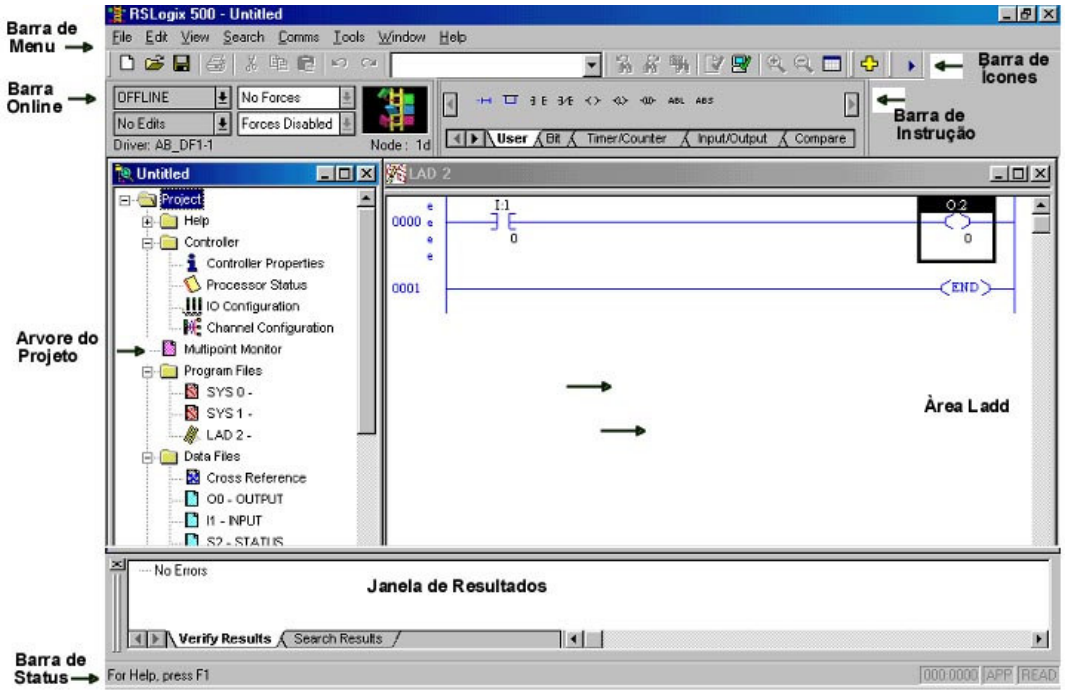
- Criar novos programas offline ou online.
- Enviar programas para o CLP (download).
- Ler programas do CLP (upload).
- Salvar as aplicações em disquete.
- Editar programas offline ou online.
- Imprimir programas.
- Impor condições de forçamento (forces) em E/S.
- Monitorar estados de programa online, verificando ou alterando parâmetros.

Requisitos de sistema

Este software foi desenvolvido para plataformas Windows 98, 2000 e XP. O Hardware mínimo é um microprocessador Pentium ou compatível com 16MB de RAM e 8MB disponível em disco rígido e uma porta serial RS232.

5.2 - Navegando no RSLogix 500

Quando você abrir um projeto no RSLogix 500, você terá a seguinte tela:



Barra de Título: Serve para mostrar o nome do programa e outras informações adicionais. No RS Logix *5,* além do título, ela pode mostra o nome do projeto.

Barra de Menu: Local onde são acessados todos os comandos que podem ser dados no programa. Basta clicar na opção para que o menu seja aberto.

Barra Online: Informa o modo de operação , e permite visualizar se há edições online ou forces. Você visualiza ainda o driver configurado no RS Linx e o nó da rede.

Barra de Ícones: Ela contém muitas funções que você irá utilizar repetidamente no desenvolvimento, e conferência da sua lógica de programa. A procura de instruções e/ou endereços aparece aí, bem como a verificação se o seu programa não possui erros.

Barra de Instruções: Mostra o mnemônico das instruções numa tabela de categorias. Quando você clica na categoria da barra de instruções, você muda a categoria trocando as instruções para as da categoria selecionada. Clique na instrução para inseri-la no seu programa Ladder.

Arvore do Projeto: Contém todos os parâmetros e arquivos do seu projeto. Você pode clicar no ícone desta árvore, e quando clicar com o botão da direita do mouse um menu de opções se abrirá. As opções que se abrirão poderão ser para renomear o arquivo de programas, abrir um programa ou revelar propriedades do arquivo de programas.

Área de Ladder: Nesta pane da janela de aplicação você verá os arquivos de programas em tempo real. É aqui que você editará o Ladder.

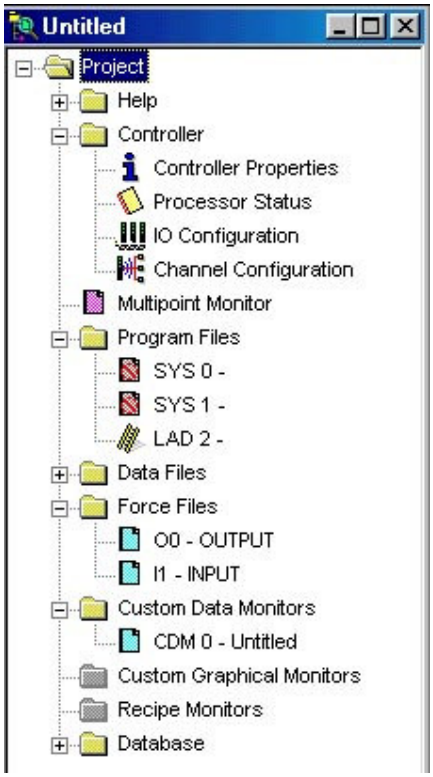
Janela de Resultados: Mostra os resultados da procura em todo o programa, ou a verificação de erros de projeto. Você pode alterar o tamanho desta janela ou deslocá-la na janela de aplicação.

Barra de Status: O campo da direita sempre informa o tipo de objeto quando há um selecionado. O campo da esquerda fornece informações sobre posição da linha no ladder e dá explicações curtas sobre as opções de menu e botões selecionados.

5.3 - A Árvore do Projeto

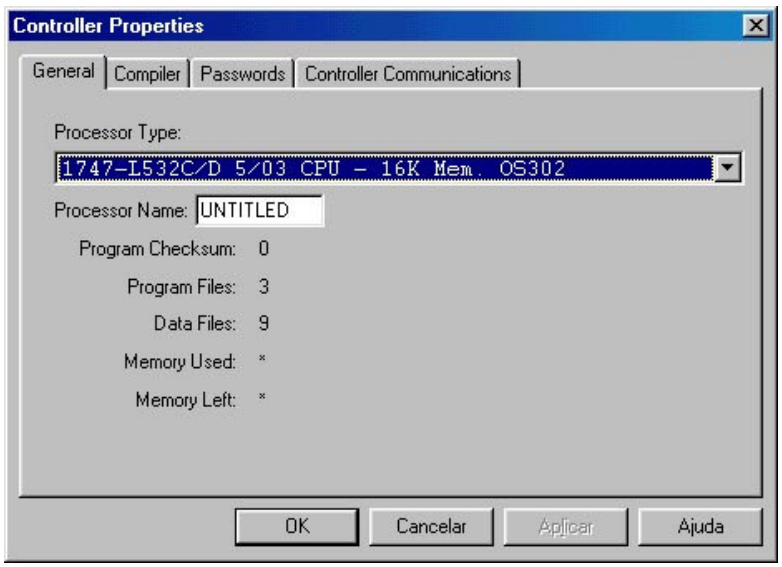
Na árvore do projeto temos todas as pastas e arquivos do seu projeto. As pastas são organizadas de forma a agrupar elementos afins. Para fechar uma pasta basta dar um clique no sinal de “+”, e para abri-la clique no sinal de “-“ .Como vemos na figura a seguir, as pastas são as seguintes:

1 - Controlier (controle),
2 - Program Files,
3 - Data Files,
4 - Force Files,
5 - Custom Data Monitors,
6 - Database.



A Pasta Controller
Controlier Properties

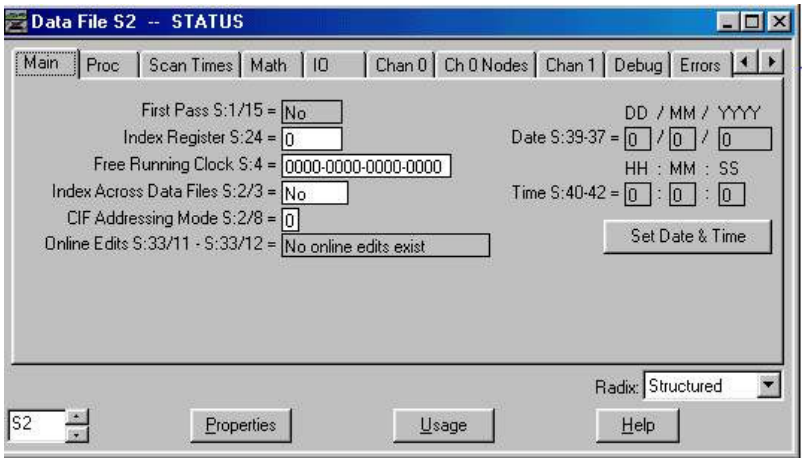
Neste item Controiler Properties (propriedades do PLC), temos a possibilidade de modificar o nome da aplicação, o modela da CPU, , senha, a driver, nó de comunicação e bloquear alguns acessos. Vide na figura abaixo que para mudar o Password, devemos selecionar com um clique na aba superior. O mesmo se dá para a configuração da comunicação.



Processor Status

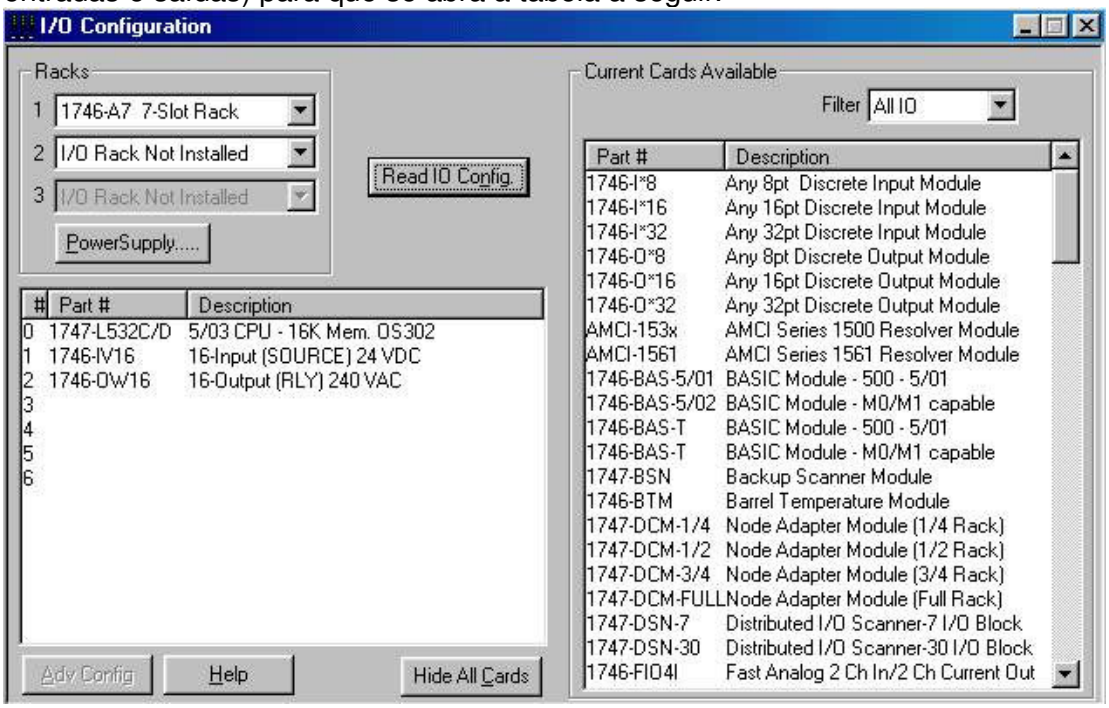
O tem Processor Status (estado da PLC) nos mostra de forma organizada, o arquivo de Status do PLC. Este é um dos arquivos mais importantes da aplicação, pois é nele que identificamos pôr exemplo:

- Ajuste do relógio e calendário interno,
- Ajuste e visualização da velocidade da Varredura (Scan),
- Flags aritméticos (Carry, Zero, Overflow e Signal),
- Situação das chaves (dip-switches) do fundo do Cassis,
- Falhas graves (Major) e de advertência (Minar) do PLC,
- Bits de bateria fraca,
- Presença ou **não** de forces,
- Habilitação ou não de varredura e reset de Racks.



I0 Contiguration

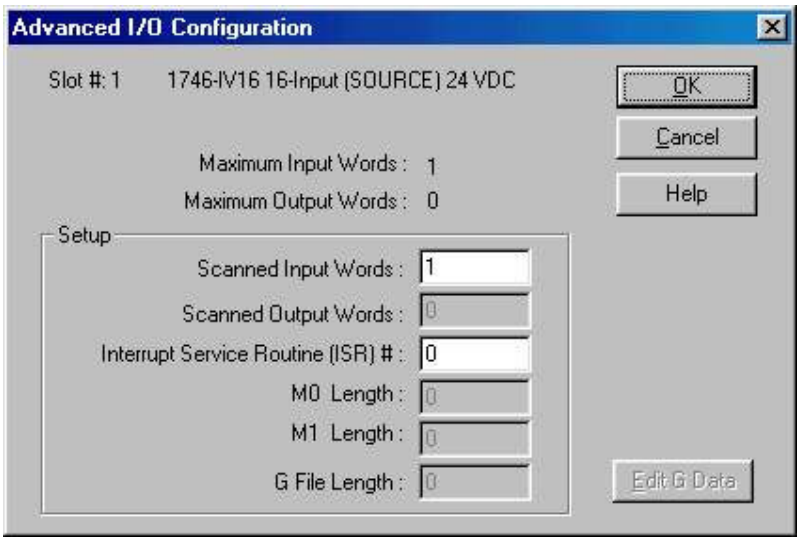
É muito importante especificarmos todos os racks e cartões que estão associados ao do projeto PLC. Clique no item LO Configuration (configuração das entradas e saídas) para que se abra a tabela a seguir.



Faça então a escolha dos Racks locais, que podem ser no máximo 3 ou até se atingir 30 slots. Para configurar o modelo dos Racks, clique no botão drag and drap e escolha o tamanho adequado.

Para configurar as cartões que ficarão nos slots, selecione em primeiro lugar o slot e depois clique duas vezes no campo Current Card Avaliable.

Há alguns cartões que necessitarão de configuração. Para isso clique duas vezes sobre ele e uma tela similar à mostrada a seguir, deverá ser configurada.

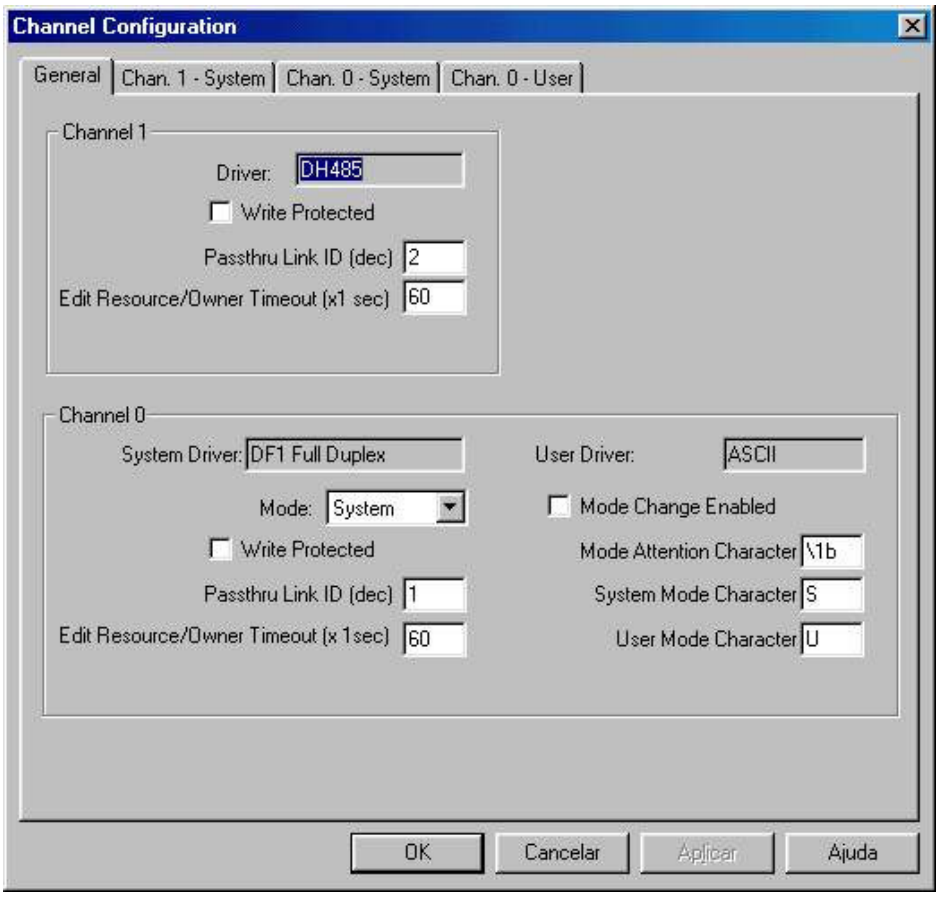


Channel Configuration

Cada modelo de CPU poderá apresentar variações da configuração mostrada a seguir, por isso escolheremos o PLC de maior quantidade de canais. Pode-se então a partir daí configurar os de menor complexidade.

O canal 0 é o canal responsável pela comunicação com o micro ponto a ponto, via RS 232 C. Pode-se mudar nesta opção a velocidade de comunicação (Baud Rate) do micro com o PLC.

O Canal 1 pode ser o canal que comunica com a rede DH+ (CPU 5/04) ou a rede Ethemet (CPU 5/05).



Passwords and Privileges

Apesar de não muito usual em ambientes industriais, existe a possibilidade de se criar senhas de acesso (Passwords) e privilégios de classe (Privileges).

As Passwords são usadas quando se precisa bloquear todo acesso ao programa. O privilégio é quando se quer criar níveis de proteção para algumas pessoas, bloqueando alguns acessos ou modificações a panes do programa.

A Pasta Program Files

A pasta Program Files nada mais é do que o arquivo dos programas da aplicação. Os arquivos são subdivisões do programa e podem ser chamados também de Subrotinas.

O primeiro arquivo (número 0) é o de sistema. Nele estão guardados o nome do programa, e as senhas, caso elas existam.

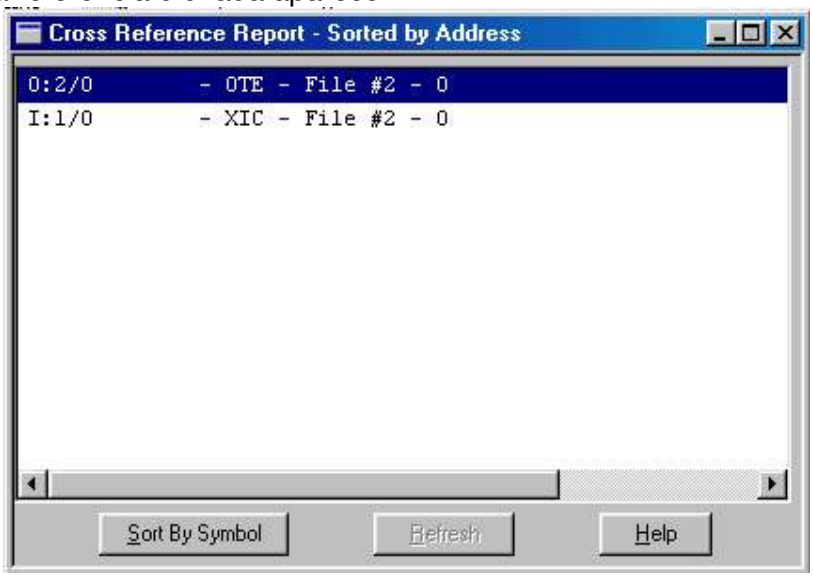
O arquivo número 1 é dedicado a um tipo de programação em blocos que associado ao Ladder permite uma melhor visualização do processo. O nome dado a este tipo de programação é SFC (Seqüência Functian Chart). Só nos PLC's da Família 5 é que teremos acesso a este arquivo.

Finalmente o arquivo de número 2, que se destina a conter a tipo de programação mais comum dos PLC's da Allen-Bradley (Rockwell), o Ladder. Não só ele, como os arquivos de 3 a 999 podem conter programa Ladder.

A Pasta Data Files

Cross Reference

A chamada Referência cruzada (Cross Reference), é uma cópia dos Diagramas de Relés, que possuíam em seu rodapé informações de onde encontrar os contatos, ou a bobina do relé. No programa Ladder, como não poderia ser diferente, tem-se um equivalente, que indica onde encontrar no programa todas as instruções relacionadas com um endereço. Veja a seguir coma a janela da referência cruzada aparece.



Data Files

A função do arquivo de dados (Data Files) é organizar a memória do PLC em partes distintas, para que assim possamos pesquisar e alterar de maneira mais rápida valores de bits e de palavras.

Podemos criar até mil arquivos mas eles por default são oito:

O0 - Arquivo de Saídas (Output) - Representa a tabela imagem das saidas fisicas do PLC.

I1 - Arquivo de Entradas (Input) - Representa a tabela imagem das entradas do PLC.

S2 - Arquivo dos estados do PLC (Status) - Vide mais detalhes na pasta Controlier, item Processor Status.

B3 - Bits auxiliares (Bit) - São os bits utilizados para a lógica interna do PLC. Eles trabalham no programa como se fossem relés auxiliares.

T4 - Arquivo de Temporizadores (Timer) - Se destinam a conter informações de bits de controle e parâmetros internos das instruções que trabalham com temporizadores.

C5 - Arquivo de Contadores (Counter) - Se destinam a conter informações de bits de controle e parâmetros internos das instruções que trabalham com Contadores.

R6 - Registradores de instruções avançadas (Register) – As instruções avançadas assim como os temporizadores e contadores precisam de um arquivo que possa guardar os seus bits controle e parâmetros. Só que como elas são menos usadas na programa, haverá apenas um arquivo comum para todas elas.

N7 - Arquivo Inteiros ou Naturais (Natural) - Considera-se este arquivo como se fosse a memória de armazenamento de valores. E usado em operações matemáticas ou em instruções que trabalhem com valores do formato de uma palavra, é que você usará os elementos deste arquivo. São considerados inteiros, porque nunca podem conter números maiores que os limites -32768 a 32767, ou fracionários.

F8 - Arquivo de Ponto Flutuante (Floating Poínt) - É um arquivo que também se destina a armazenar elementos na memória do PLC, mas a sua grandeza no que diz respeito a valores é bem maior que o anterior, alem de guardar números fracionários.

A Pasta Force Files

O arquivo de forces é uma representação em forma de tabela, de todos os forces que estão assinalados ou habilitadas na memória do PLC. Os arquivos são dois: Force de Saidas (O0) e Force de Entradas (I1).Vide mais informações no texto Como fazer um Force.

A Pasta Custom Data Monitor

Um recurso muito importante que o RS Logix 500 traz é sem dúvida a tabela de dados customizada. Nela podemos escolher os endereços a monitorar ou a modificar, sejam eles bits ou palavras. Dessa forma fica mais fácil monitorar o processo, sem precisar ficar deslocando o cursor pelo programa a procura de

um valor ou outro do processo. Pode-se criar várias tabelas e gravá-las para uma outra monitoração futura.

A Pasta Database

Esta pasta tem vários bancos de dados dos comentários do programa, onde podemos editar ou modificar a base de dados. Como se sabe os comentários são divididos em quatro tipos:

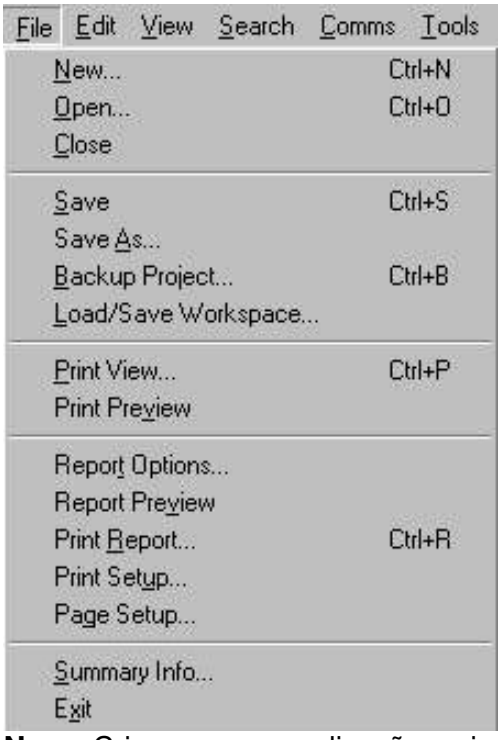
- Comentários de Linha (Rung Coments),
- Comentários de Instrução (Instrution Coments),
- Comentários de Endereço (Address Coments),
- Comentários Simbólicos (Symbols).

Os comentários de Linha são feitos para se organizar o Ladder, separando em grupos as panes do Ladder que dizem respeito a um determinado equipamento, setor do processo, ou simplesmente a uma Lógica particular.

Os comentários de Instrução e Endereço, também chamados de Description, são aqueles onde você pode descrever a função dentro do contexto do programa, ou o equipamento a que o endereço está associado.

Os comentários de Endereço são aqueles que vão direto para o endereço, sem se preocupar com a instrução que o endereço esta associado. Já os Comentários de Instrução, serão particulares para cada instrução, mesmo que ela tenha o mesmo endereço.

5.4 - O Menu File



New - Cria uma nova aplicação, acionando o comando ou o botão.

Open - Abre um arquivo de aplicação já existente, acionando o comando ou o botão.

Close - Fecha a aplicação que está sendo editada sem sair do RS Logix 500.

Save - Salva a aplicação que está sendo editada, acionando o comando ou o botão.

Save As.- Salva a aplicação que está sendo editada com um nome ou caminho que pode ser escolhido.

Backup Project.-. Cria uma cópia do programa original.

Load*/Save* workspace.- Carrega subrotinas prontas para ser utilizadas numa aplicação.

Print View.-. Imprime o Ladder que estiver selecionado na tela.

Print Preview - Permite a visualização da impressão do Ladder antes de ir para a impressora.

Report options - Mostra as opções que podem ser impressas, para a sua escolha.

Report Preview - Verifica como ficará a impressão antes mesmo de ir para a impressora.

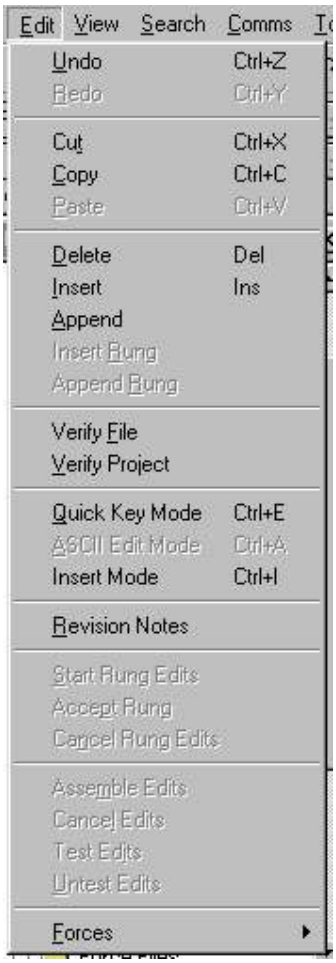
Print Report - Comando para que se imprima o que foi selecionado no comando Report Options.

Print Setup.-. Seleção das características da impressora.
Page Setup.-. Seleção das características da página.

Sumary Info.-. Mostra algumas das características da aplicação que se está trabalhando.

Exit - Fecha o RS Logix 500

5.5 - O Menu Edit



Undo - Desfaz o que foi apagado ou modificado voltando um ou vários passas para traz, de acordo com a quantidade de vezes que se aciona o comando ou o botão.

Redo - Refaz aquilo que foi apagado ou modificado voltando um ou vários passas para traz, de acordo com a quantidade de vezes que se aciona o comando ou o botão.

Cut - Propriedade de cortar uma instrução, linha ou arquivo de programa Ladder.

Copy - Propriedade de copiar uma instrução, linha ou arquiva de programa Ladder.

Paste - Propriedade de passar uma instrução, linha ou arquivo de programa Ladder que foi copiada ou cortada anteriormente.

Delete - Propriedade de apagar uma instrução, linha ou arquivo de programa Ladder.

Insert - Propriedade de inserir unia instrução, linha ou arquivo de programa Ladder.

Append - Propriedade de inserir uma instrução no programa Ladder.

Insert Rung - Propriedade de inserir uma linha no programa Ladder acima do cursor.

Append Rung - Propriedade de inserir uma linha no programa Ladder abaixo do cursor.

Verify File - Verificação do arquivo determinando se ele mesmo possui erros. Pode-se acionar o comando ou o botão.

Verify Project - Verificação do projeto determinando se ele mesmo possui erros. Pode-se acionar o comando ou o botão.

Quick Key Mode - Comando que proporciona o acesso imediato a edição do programa Ladder.

ASCII Edit Mode - Comando que proporciona o acesso imediato a edição da linha do programa Ladder através da digitação, ou alteração da mesma.

Insert Mode - Proporciona a entrada no modo de inserção no teclado.

Revision Notes - Permite visualizar alguma nota de revisão que tenha sido digitada na hora da criação da revisão do programa.

Start Rung Edits - Permite a edição de uma linha selecionada no programa.

Accept Rung - Confere se há algum erra na linha que foi editada e envia uma mensagem para a janela de resultados.

Cancel Rung Edits - Cancela as edições que estiverem em tramite, retornando com a linha original.

Assembly Edits - Este é o último passo para se efetivar as modificações online de um programa. Pode-se aceitar alteração de várias linhas simultaneamente.

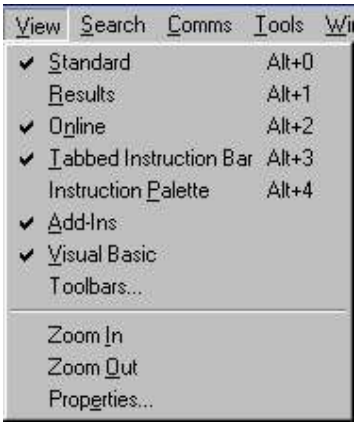
Cancel Edits - Cancela todas as modificações online nas linhas do programa de uma vez.

Test Edits - Este passo permite a verificação online das linhas que estão sendo modificadas, fazendo a execução das mesmas.

Untest Edits - Após ter sido verificada a lógica das linhas em edição (Test Edits) online pode-se voltar atrás, fazendo as linhas originais serem executadas.

Forces - Endereços de entrada e saída podem ser forçados a atingir valores, independente de seus estados físicos ou lógicos.

5.6 - O Menu View



Standard - Esta barra mostra os botões de comandos mais usuais do windows, verificação de arquivo ou projeto e procura (Find).

Results - Esta barra normalmente não aparece marcada como opção Default, porém se na hora da verificação do arquivo ou programa aparecerem erros ela aparecerá na tela. Outra maneira dela aparecer é quando fizermos a procura Find.

Online - A barra Online é de estrema importância na verificação e mudança dos modos de operação, verificação e ativação de Forces, verificação de edições pendentes no programa.

Instrution - Barra responsável pela edição de Ladder onde podemos escolher as instruções pôr categorias.

Instrution Palette - Trata-se de uma palheta de instruções onde escolhe-se a instrução através de um clique de mouse, ao invés de digitá-la.

Zoom In - Aumenta o tamanho do programa cada vez que se dá um comando ou se clica no botão.

Zoom Out - Diminui o tamanho do programa cada vez que se dá um comando ou se clica no botão.

Properties - Comando que faz abrir uma janela de acesso as características do programa Ladder. Permite mudar cores, Fontes, formato dos comentários, entre outras.

5.7 - O Menu Search

Find.-.Procura pela instrução/endereço onde o cursor está posicionado.

Replace.-.Muda a instrução/endereço onde o cursor está posicionado por uma nova Instrução/endereço.

Advanced Diagnosties.-. Tipo de procura que nos permite localizar partes do programa Ladder a partir de comentários de titulo.

Goto.-. Procura realizada a partir da linha, ou do arquivo que se quer localizar.

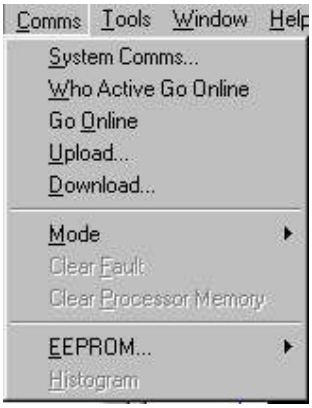
Find Next - Procura para frente da instrução/endereço onde o cursor está posicionado.

Find Previons - Procura para traz da instrução/endereço onde o cursor esta posicionado.

Next Error - Localiza para frente um erro de programa identificado pela verificação.

Prev Error - Localiza para frente um erro de programa identificado pela verificação.

5.8 - O Menu Comms



System Comms.-. Permite a seleção do driver e do nó de comunicação já configurado previamente no RS Linx. Tem acesso também a download e upload de programas.

Who Active Go Online - Acessa o RS Linx permitindo que se tenha uma visão da rede já configurada, e a partir dai entre Online no nó selecionado.

Go Online - Entra em comunicação com o nó configurado anteriormente.

Upload - Leva para o micro uma cópia do programa que está no nó acessado.

Download - Envia para o nó configurado, o programa que está no micro.

Mode.-. Muda o modo de trabalha do nó rede. Se existir uma CPU, ela deverá estar coma chave frontal na posição REM.

Clear Fault - Limpa as falhas maiores da CPU.

Clear Processor Memory - Limpa o programa da memória da CPU ou da memória do micro.

EEPROM.- Armazena o programa que está na memória da CPU na EEPROM instalada no PLC ou vice versa. Desta forma se tem um Backup do programa que roda no PLC.

Histogram - Monitoração de uma palavra ou bit de programa onde é permitido um acompanhamento de seus estados e do intervalo de permanência neste estado. É apresentado junto com a monitoração, um gráfico do estado x tempo.

5.8_ O Menu Tools

Options - É aberta uma janela de configuração do tamanho da linha de comentário, auto-salvamento, Browsers onde serão gravados os módulos do programa. Ele tem ainda em uma outra página, a réplica da janela de configuração dos drivers e nó de comunicação.

Delete Unused Memory - Comando recomendado quando se necessita fazer uma redução do tamanho do programa eliminando-se apenas partes desnecessárias do programa.

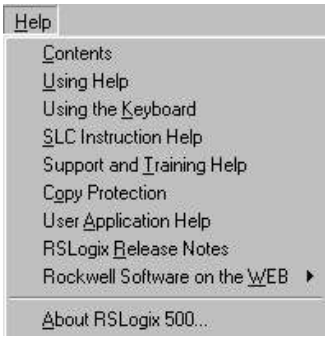
Database - Mostra um sub-menu que permite copiar, mover, deletar, modificar o formato da tabela de dados, etc...

5.9 - O Menu Window

Arrange - Quando acionado abre uma janela onde se seleciona a maneira de organizar as janelas do arquivo de aplicação. Esta organização pode ser Vertical, Horizontal, em Cascata. Podemos ainda retomar com as características defaut das janelas e barras de menu.

OBS: Existe ainda a possibilidade de seleção das janelas através do clique sobre o titulo da mesma que aparece logo abaixo do Arrange.

5.10 - O Menu Help



Contents - A abertura da janela de Help geral é feita quando se dá um clique nesta opção.

Using Help - Help do windows que lhe ensina com usar o Help.

Using the Keyboard - Trata-se de um help que traz um resumo das teclas de atalho utilizadas no software.

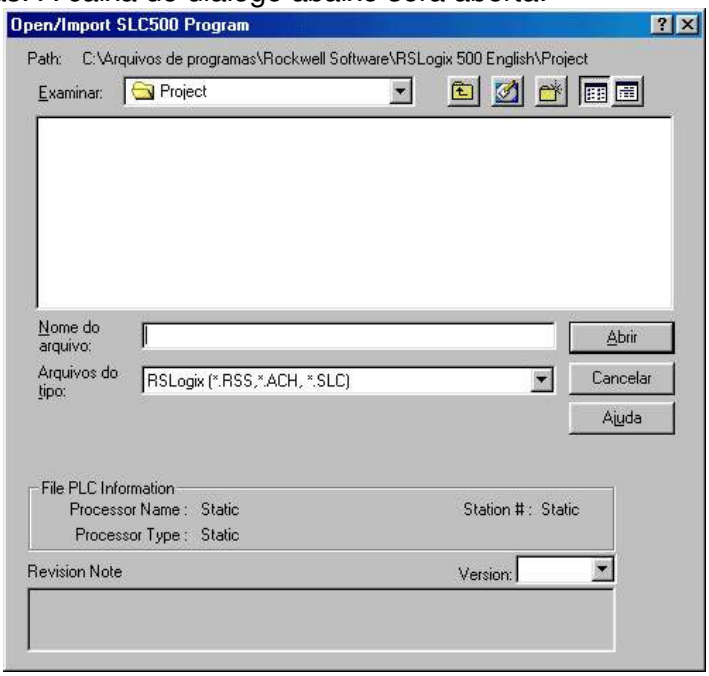
SLC Instrution Help - Oferece um Help de todas as instruções do SLC de forma a facilitar a sua programação.

Support and Training Help - Traz várias informações de como localizar suportes técnicos e treinamento para os softwares de Rockwell.

About RS Logix 500.-. Oferece informações gerais do RS Logix. São elas a revisão, número de série, contato telefônico e postal para dúvidas técnicas, etc...

5.11 - Passos para abrir um programa

1 - Dê um click na opção Open... do menu ou use o botão da barra de ferramentas. A caixa de diálogo abaixo será aberta:



2 - Escolha um tipo de arquivo, onde este pode ser vindo do RS Logix 500 ou de qualquer software anterior a ele. Pôr exemplo do APS.

3 - Depois clique em 0K para abri-lo.

5.12 - Passos para Editar um Programa

As Edições são feitas toda vez que estamos fazendo adequações do programa para que ele funcione de maneira a atender os requisitos que a produção estipulou. E certo que os ajustes as vezes são demorados e a melhor maneira de fazê-los é deixando a CPU em modo PROG (programação).

Há situações entretanto, em que não é possível parar o processo, assim as modificações deverão ser feitas em modo RUN, com o PLC Online.

CUIDADO: Observe que os modelos de CPU 5/01 e 5/02 não aceitam modificações online.

ATENÇÃO: Sempre que fizer edições Online, deve-se ter cuidado redobrada, pois qualquer descuido pode ser fatal para os equipamentos, ou para as pessoas envolvidas no processo.

Os passos para a edição online ou offline serão descritos a seguir, mas note que em modo RUN aumenta o número de passos para uma maior segurança.

1 -Entre na aplicação a ser editado pelo RS Logix 500. Siga os passos do item Como abrir um arquivo.

2 - Acesse o arquivo Ladder a ser editado dando um clique sobre o item definido para ele na pasta Program Files. Pôr exemplo LAD 3.

3 - Localize a linha no arquivo deslocando o cursor sobre o programa, ou utilizando o search para levá-lo a instrução pretendida.

4 - Clique duas vezes na borda lateral esquerda da linha, se quiser modificar a linha como um todo (Online serão necessários mais dois cliques). Aparecerá então toda a linha, pôr extenso. Modifique sobre o que você quiser e de <ENTER>.

5 – Em edição Offline para verificar se as alterações não possuem erros, e também sair do modo de edição, clique no botão:  . Em edição Online, clique no botão: 

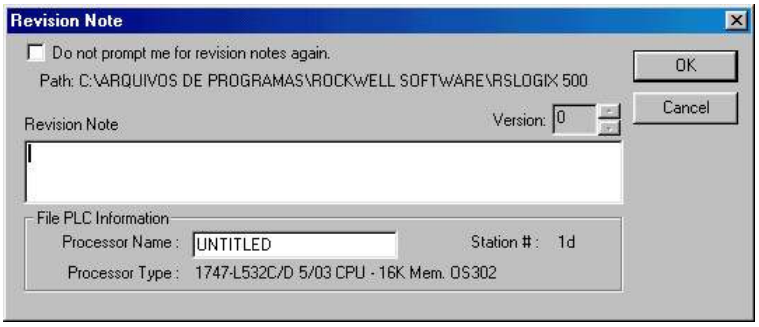
ATENÇÂO

1 - Deve-se testar a linha editada, para ter certeza que ela esta funcionando dentro da lógica prevista. Assim clique em:  e confirme com yes a pergunta.

2 - Se a lógica estiver ok, confirme as modificações com:  em seguida yes para confirmar. Dessa forma você estará saindo do modo de edição em definitivo.

5.13 - Passos para Salvar um programa

Para salvar a arquivo com o mesmo nome e no mesmo diretório, use a opção Sane do menu ou use o botão da barra de ferramentas. Será mostrada então a janela a seguir onde podemos optar pôr uma nova revisão e até fazer comentários da mesma.



5.14 - Passos para fazer Download

Quase sempre é preciso enviar o programa para a CPU, por causa de alguma falha, ou por alguma modificação mais radical que foi feita Offline. Para isso temos que fazer o Download. Faça então assim:

1 - Entre Offline no programa a ser enviado para a CPU.

2 - Na Barra Online abra a caixa de mudança de modo e clique no item Dowload (Vide figura).

ATENÇÃO: O Download deve ser feito sempre com a CPU no modo PROG, porque não é possível se transferir um programa com outro rodando no mesmo PLC.

5.15 - Passos para fazer Upload

Assim como é preciso fazer modificações na própria CPU, enquanto ela está trabalhando. Não se pode esquecer que uma cópia deste programa deve estar sempre na memória do micro, de forma que numa eventual perda do mesmo, possamos ter um back-up. Por isso devemos sempre fazer um Upload do programa para o micro. Veja como se faz:

1 - Entre Online, se já não estiver.

2 - Na Barra Online abra a caixa de mudança de modo e clique no item Upload (Vide figura).

5.16 - Passos para fazer Force

O Force é a maneira de obter um estado num endereço independente de qualquer outra condição. Os Forces só podem ser feitos para endereços de entrada e saída físicos (sempre Online), sendo assim é impossível forçar endereços lógicos. É o caso de bits auxiliares e bits com endereços de saídas.

Os Forces são muito perigosos, porque se implementados sem um prévio estudo de seus efeitos, podem ocasionar danos pessoais ou ao equipamento. Por isso saiba bem os seus efeitos.

Um Force de entrada é feito desconsiderando-se qualquer mudança no estado do campo. Portanto, se uma entrada é forçada para ON, não importa o que ocorra no campo, que ela manterá seu estado. Com relação a uma saída, acontece o mesmo, mas neste caso ela independe dos estados da lógica do programa.

Siga os passo abaixo para a perfeita implementação do Force.

1 - Acesse a linha de programa onde o Force será feito.

2 - Clique com o botão direto do mouse sobre a instrução a ser forçada. Nas opções que aparecerem escolha Force ON, ou Force 0FF.

3 - Para que o Force tenha efeito deve-se alterar na Barra Online a caixa que informa Forces Disable para Enable All Forces. Em seguida confirme com yes.

No caso de remoção dos forces retome a Barra Online, e altere do modo Forces Intalled para Remove All Forces.

Se você só quiser retirar um único Force, clique com o botão direito do mouse sobre a instrução e escolha a opção Remove Force.

5.17 - Passos para alterar o Modo de Operação

Os Modos de Operação são a maneira com que o PLC deve trabalhar. Existem três Modos: PROG, RUN e TEST. Sempre que quiser alterar o Modo pelo programa deve-se manter a chave frontal do PLC em REM.

• Modo PROG tem como finalidade podermos alterar o programa ou qualquer um de seus arquivos. Pode-se até mesmo fazer um Download. As saídas são totalmente desenergizadas, e o programa não executa o Scan.

• Modo RUN roda o programa e são limitadas as alterações no que diz respeito a arquivos. Não podemos fazer Download, mas o Upload é possível.

• Modo TEST o PLC roda o programa, sem entretanto energizar as saídas. Para arquivos, Upload e Download, valem as mesmas considerações que no modo RUN.

Deve-se seguir os passos:

1 - Estando no programa Online, clique na caixa seletora de Modo na Bana Online.

2 - Selecione o Modo desejado: PROG, RUN ou TEST.

3 - Em seguida confirme com Yes.

5.18 - Passos pa adicionar símbolos e comentários no programa

Os símbolos podem ter até 20 caracteres. Os caracteres podem ser as letras de A a Z e números de 0 a 9. O símbolo não pode começar com um caractere numérico. Os espaços não são permitidos.

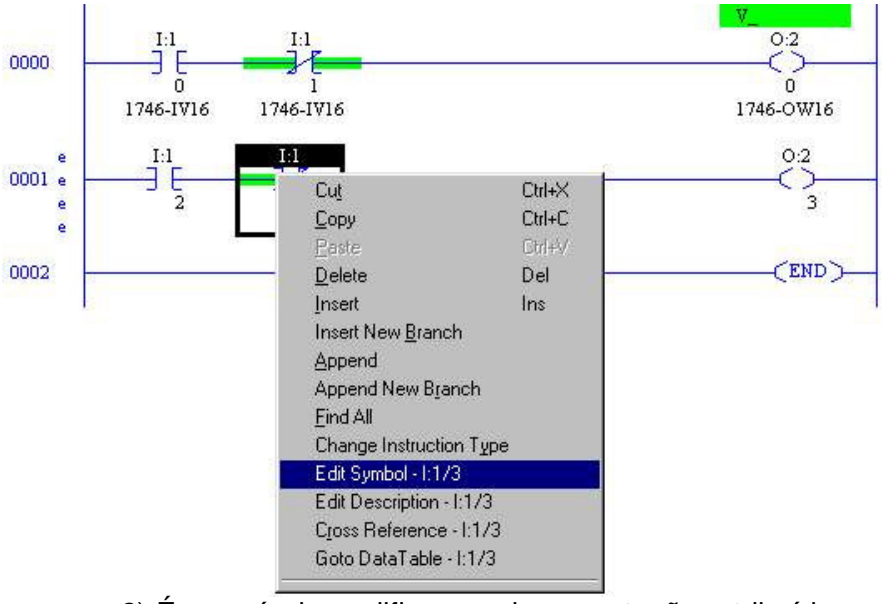
É possível utilizar vários métodos para adicionar símbolos e descrições aos endereços no banco de dados.

1) É possível abrir o arquivo de programa e adicionar a documentação diretamente à instrução endereçada.

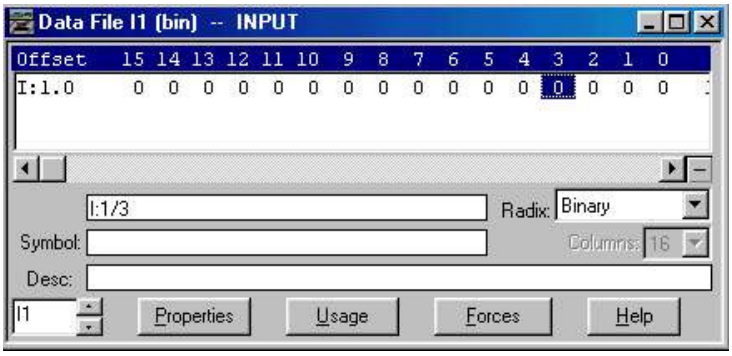
• Clique na instrução dentro do arquivo do programa que você deseja documentar.

• Clique com botão direito do mouse e selecione Editar Símbolo ou Editar descrição.

• No caso de Editar símbolo digite símbolo de sua escolha e pressione [ENTER], no caso de descrição digite a descrição e dique em 0K.

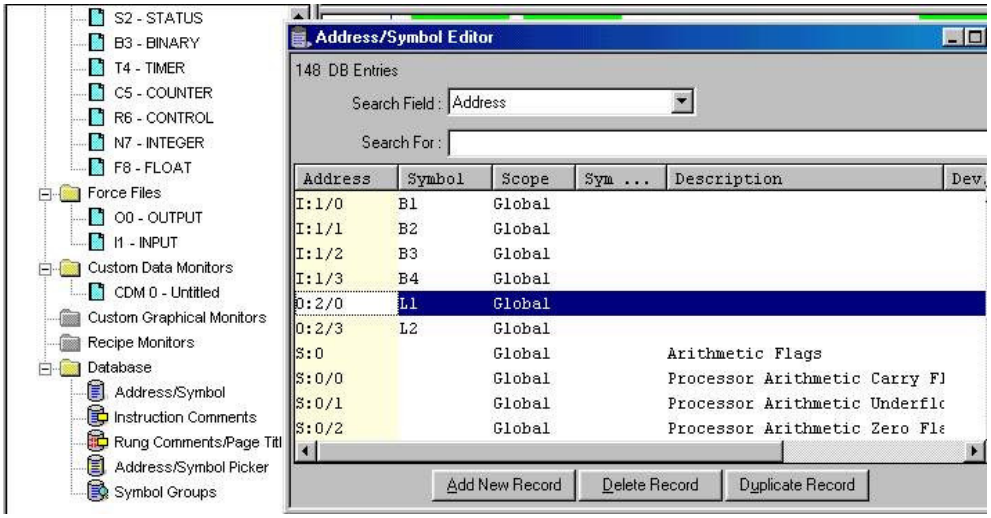


2) É possível modificar a documentação atribuída ao endereço no arquivo de dados.



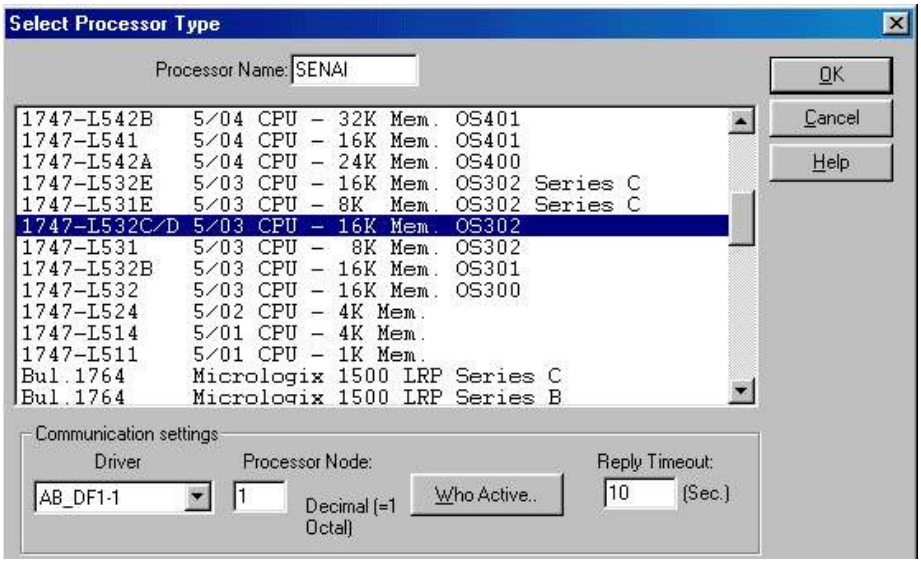
3) É possível modificar o banco de dados através de editor de banco de dados.

4) Os símbolos podem ser digitados sem precisar definir suas designações de endereços primeiro. Apenas dique em uma instrução e digite em uma instrução e digite um nome de símbolo em vez de um endereço. Em seguida, será possível atribuir endereços aos símbolos utilizados no programa no Editor de Banco de Dados. dique no ícone Endereço/Símbolo da pasta Banco de Dados da árvore de projetos para acessar o Editor de Banco de Dados.



5.18 - Passos para criar uma nova aplicação

1 - Para criar uma nova aplicação dê um dique na opção New... do menu ou use o botão da barra de ferramentas. A caixa de diálogo abaixo será aberta.



2 - A opção Processor Name deve ser preenchida com o nome sugerido para o projeto.

3 - Escolha a CPU que você irá trabalhar, assim como a série/revisão e tamanho da memoria.

4 - Se necessário, escolha o Driver e o nó de rede (Processor node) que será usado para comunicação. Vide informações abaixo para informar-se de como configurar o driver no RS Linx.

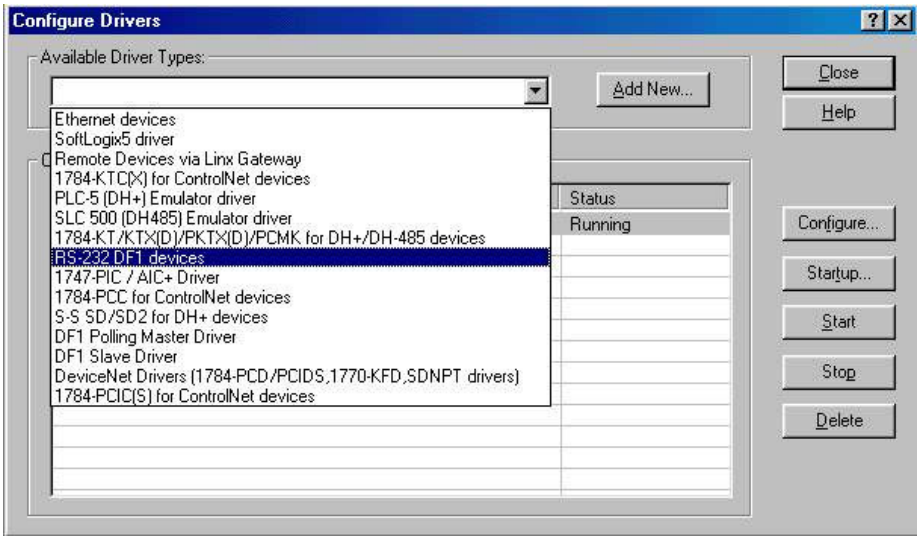
5 - Clique em 0K para que o novo arquivo seja aberto

5.19 - Configuração do driver no RS Linx

Para salvar uma aplicação do RS Logix no PLC, e imprescindível que se configure antes o driver no RS Linx. Para isso faça o seguinte:

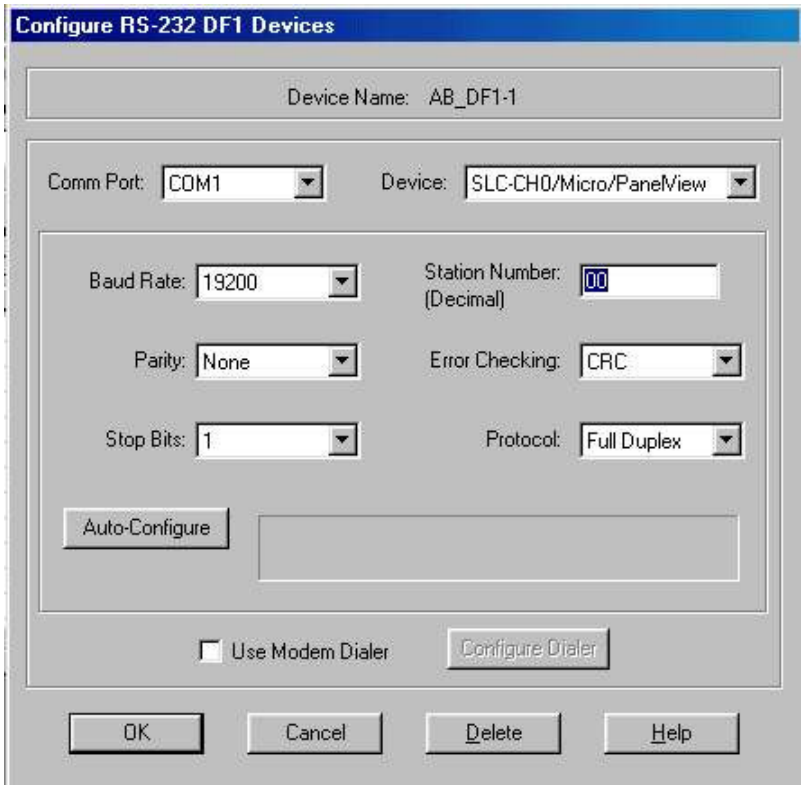
1 - Abra o Software RS Linx.

2 - Clique no botão  e escolha um dos drivers disponíveis (Avaliable Drivers Types). Pôr exemplo RS-232 DF1 Devices. Vide figura na seqüência.



3 - Adicione-o na lista de drivers configurados (Configured Drivers) com o botão Add New e OK na sequência.

4 - Aparecerá uma nova janela onde selecionaremos pôr exemplo a porta de comunicação (COM1, COM2,...) e o tipo de conexão fisica (Placa KT/KE, Canal 0 do PLC, etc...). Um exemplo está mostrado na figura a seguir.



5 - Configure em Comm. Port a porta de comunicação do Micro (COM1 ou COM2) e em Device a opção SLC-CH0/Micro/Panelview. Em seguida clique no botão Auto-configure, caso o cabo de comunicação da CPU esteja ligado ao micro. Isto fará com que as outras informações da janela sejam automaticamente lidas do PLC.

6 - Clique em 0K e retome a tela inicial do RS Linx.

7 - Minimize o software RS Linx, para que você possa comunicar no futuro com a CPU via RS Logix.

OBS: Caso você não tenha uma CPU na hora da configuração, você não deverá apertar Auto-configure

5.20 - Configuração do Driver no RS Logix 500

Abra o menu Tools, e clique na opção Options- system comms. Selecione o Driver da lista de drivers, e o número do nó da rede que se quer comunicar. A partir dai o seu micro está pronto para comunicar com o SLC-500.

5.21 - Passos para criar um Programa Ladder

Após ter criado uma nova aplicação siga os seguintes passos:

1 - Na Barra de Instruções clique em inserir degrau.

2 - Insira uma instrução da Barra de Instruções, escolhendo a categoria e a instrução que você precisar. Na categoria são abertas uma série de opções, basta clicar nas abas inferiores da Barra.

3 - Digite o endereço ou os parâmetros da instrução e dê <ENTER>.

4 - Repita os passos 2 e 3 até que a linha seja completada, não esquecendo que a edição deve ser feita sempre da esquerda para a direita. No caso de paralelos na linha, vide as informações a seguir.

5 - Se quiser uma nova linha, repita os passos 2, 3 e 4.

6 - Para verificar se o seu programa não possui erros, e também sair do modo de edição, dique no botão verificador de erros. 

Criação de Paralelos

O paralelo deve ser feito depois que a parte linear da linha é editada. Siga os seguintes passos:

1 - Posicione o cursor no lado esquerdo de onde o paralelo deverá aparecer.

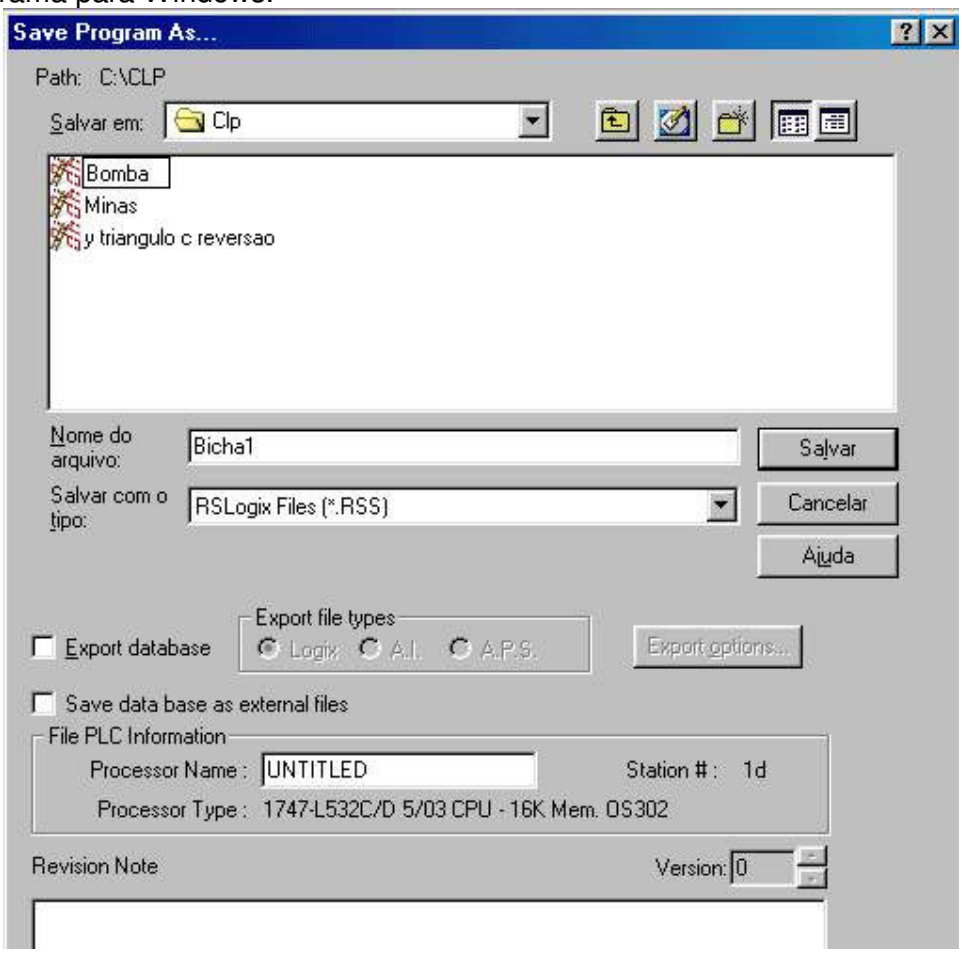
2 - Na Barra de Instruções dique no botão: 

3 - Em seguida clique e arraste com o mouse o lado direito do paralelo, envolvendo assim as instruções que ficarão dentro do paralelo (só solte quando a caixa vermelha ficar verde).

4 - Insira uma instrução da Barra de Instruções, escolhendo a categoria e a instrução que você precisar. Na categoria são abertas uma série de opções, basta clicar nas abas inferiores da Barra.

5 - Digite o endereço ou os parâmetros da instrução e dê <ENTER>.

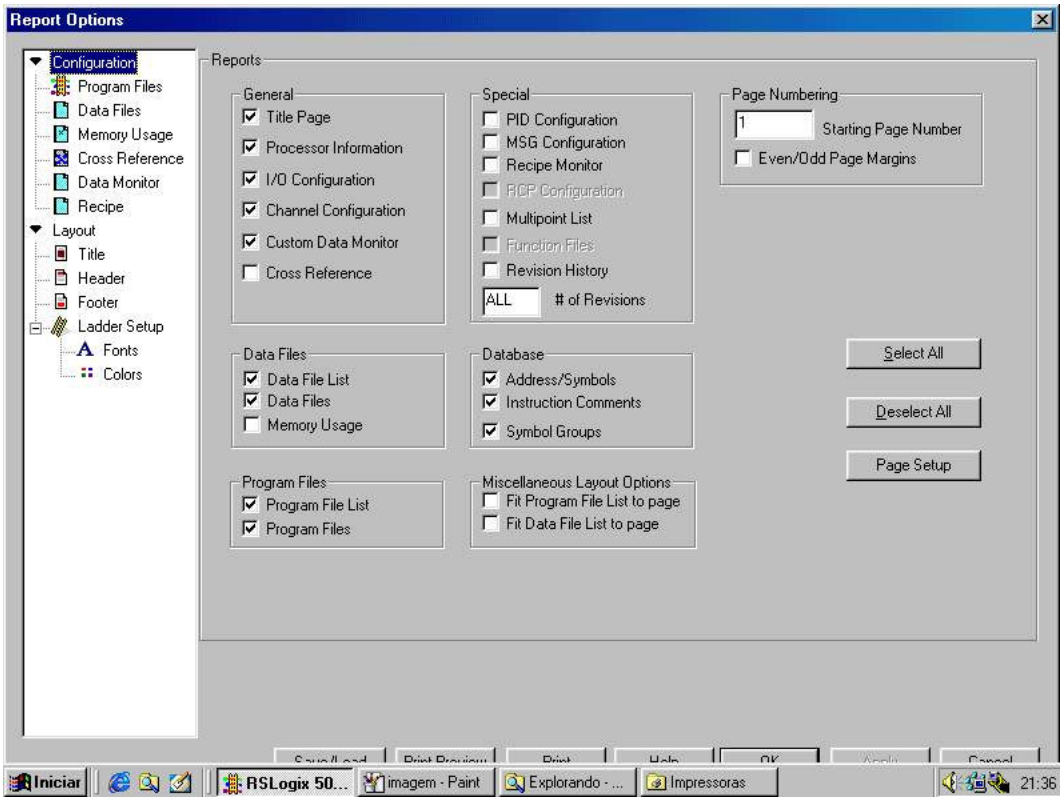
Para salvar o arquivo com outro nome ou em um diretório diferente use a opção Save As... do menu file. Opere essa caixa de diálogo como em qualquer outro programa para Windows.



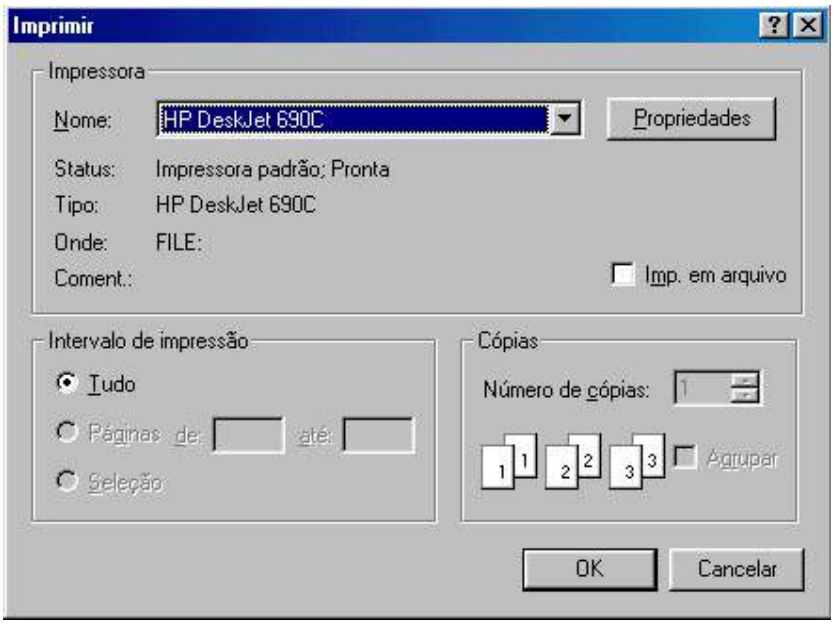
Passos para Imprimir um Projeto

1- Clique no comando Report options onde se terá acesso a tela das opções de impressão, e poderemos selecionar o que será impresso. Não pressione o botão ao lado antes de selecionar.

Dentre as opções as mais importantes está a pasta Program Files, onde podemos escolher a lista de programas, e a faixa do programa a ser impressa. Se quisermos todo o programa, basta manter em Program Files Range a seleção All Files. Há também os campos Reports (seleção geral), Title, Header and Footer (cabeçalho e rodapé), Data Files (arquivo de dados), Data Monitor (Tabela de monitoração) e Cross Reference (Referência Cruzada).



2 - Após selecionado dê um clique no botão **Print** ou vá até o comando **Print Reports**. Em seguida, como é de costume no windows ele mostrará a janela de configuração da impressora. Configure-a de acordo com a conveniência.

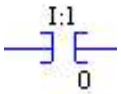


6 - Instruções para programação em Ladder

O conjunto de instruções do soft RSLogix 500 é muito completo, podendo encontrar as mais diversas instruções necessárias para uma aplicação de grande porte. A seguir será apresentado as principais instruções, caso necessite de informação sobre alguma outra instrução você encontrará no menu Help.

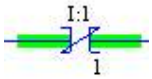
6.1 - Instruções básicas

Examinar se Energizado (XIC)



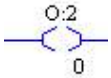
Examina o bit da tabela de dados I:1/0, o qual corresponde ao terminal 0 de um módulo de entrada localizado no cartão E/S 1. Se este bit da tabela de dados estiver energizado (1), a instrução é verdadeira.

Examinar se Desenergizado (XIO)



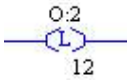
Examina o bit da tabela de dados I:1/1, o qual corresponde ao terminal 1 de um módulo de entrada localizado no cartão E/S 1. Se este bit da tabela de dados estiver desenergizado (0), a instrução é verdadeira.

Energizar Saída (OTE)



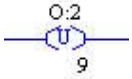
Se as instruções de entrada que antecedem esta instrução de saída na mesma linha passam a verdadeira, o bit 0:2/0 é energizado, o qual corresponde ao terminal 0 de um módulo de saída localizado no cartão E/S 2.

Energizar Saída com Retenção (OTL)



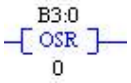
Se as condições de entrada anteriores a esta instrução de saída na mesma linha passam a verdadeira, o bit 0:2/12 é energizado, o qual corresponde ao terminal 12 de um módulo de saída localizado no cartão E/S 2.

Desernergizar Saída com Retenção (OTU)



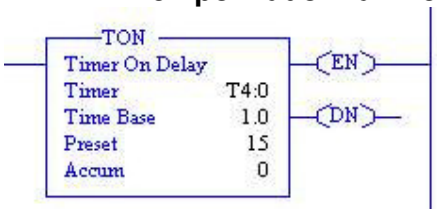
Se as condições de entrada anteriores a esta instrução de saída na mesma linha passam a verdadeira, o bit 0:2/9 é desenergizado, o qual corresponde ao terminal 9 de um módulo de saída localizado no cartão E/S 2. Isto é necessário para desenergizar um bit que foi energizado com retenção (OTL).

Subida do Monoestável (OSR)



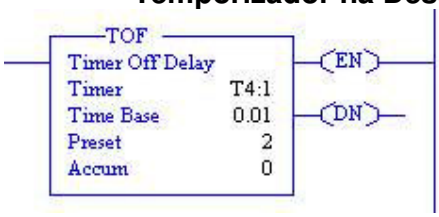
Esta é uma instrução de entrada condicional que dispara um evento para ocorrer uma vez. Se a condição de entrada for de falso para verdadeiro , a OSR é verdadeira durante uma varredura.

Temporizador na Energização (TON)



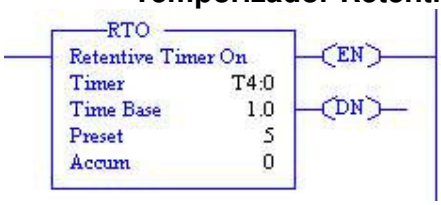
Se a condição de entrada se toma verdadeira, o temporizador começa a incrementar em intervalos selecionados (Time Base). Quando o valor acumulado (ACC) é maior ou igual ao valor pré-selecionado (Preset), o temporizador pára e energiza o bit de executado do temporizador (DN).

Temporizador na Desenergização (TOF)



Se a condição de entrada é falsa, o temporizador começa a incrementar em intervalos selecionados (Time Base). Quando o valor acumulado (ACC) é maior ou igual ao valor pré-selecionado (Preset), o temporizador pára e energiza o bit de executado do temporizador (DN).

Temporizador Retentivo (RTO)

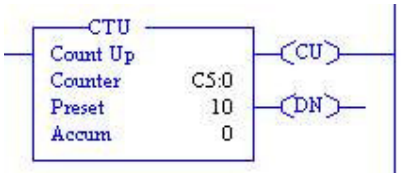


Se a condição de entrada se toma verdadeira, o temporizador começa a incrementar em intervalos selecionados (Time Base). Quando a linha passa a falsa, o temporizador pausa a temporização e retorna somente quando a linha for verdadeira. Quando o valor acumulado (ACC) é maior ou igual ao valor pré-selecionado (Preset), o temporizador pára e energiza o bit de executado do temporizador (DN).

Nos temporizadores existem os bit EN, DN e TT, o bit EN é verdadeiro quando a linha for verdadeira, o bit DN é verdadeiro quando o valor acumulado for igual ao pré-selecionado e o bit TT é verdadeiro durante a contagem de tempo.

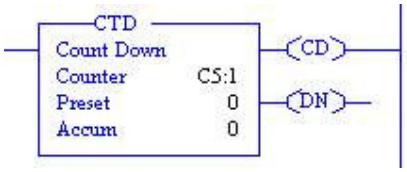
Quando for necessário usar o valor acumulado durante o programa deve se usar o seu endereço, como por exemplo: T4:0.ACC

Contador Crescente (CTU)



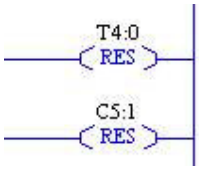
Se a condição de entrada se toma verdadeira, o contador inicia a contagem incrementando em 1 sempre que a linha passa de falsa para verdadeira. Quando o valor acumulado é maior ou igual ao valor pré-selecionado (Preset), o contador energiza o bit de executado (DN).

Contador Decrescente (CTD)



Se a condição de entrada se toma verdadeira, o contador inicia a contagem decrementando em 1 sempre que a linha passa de falsa para verdadeira. Quando o valor acumulado é maior ou igual ao valor pré-selecionado (Preset), o contador energiza o bit de executado (DN).

Rearme do Temporizador ou Contador (RES)



Se a condição de entrada se toma verdadeira, o valor acumulado (ACC) do temporizador ou contador é ressetado (=0).

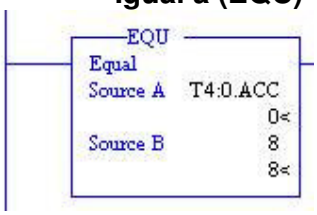
Nos contadores existem os bit CU, CD e DN, os bits CU e CD são verdadeiros quando a linha for verdadeira, o bit DN é verdadeiro quando o valor acumulado for maior ou igual ao pré-selecionado no contador.

Quando for necessário usar o valor acumulado durante o programa deve se usar o seu endereço, como por exemplo: C5:0.ACC.

Para se obter um contador crescente e decrescente (UP-DOW) usa-se dois contadores, um UP e um DOW, com o mesmo endereço.

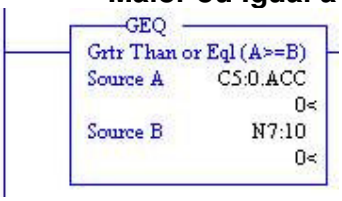
6.2 - Instruções de comparação

Igual a (EQU)



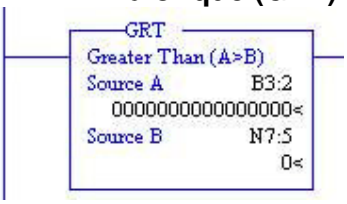
Se o valor em Source A é igual ao valor em Source B, esta instrução é verdadeira.

Maior ou Igual a (GEQ)



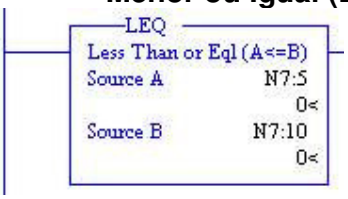
Se o valor em Source A é maior ou igual ao valor em Source B, esta instrução é verdadeira.

Maior que (GRT)



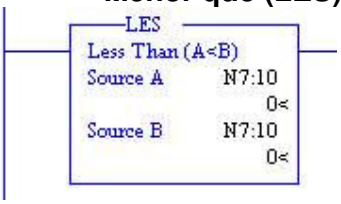
Se o valor em Source A é maior ao valor em Source B, esta instrução é verdadeira.

Menor ou Igual (LEQ)



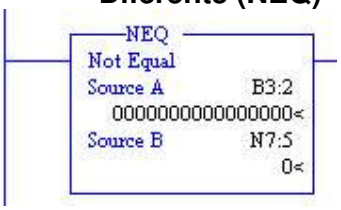
Se o valor em Source A é menor ou igual ao valor em Source B, esta instrução é verdadeira.

Menor que (LES)



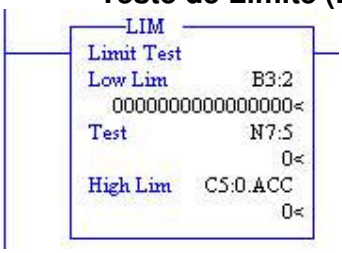
Se o valor em Source A é menor ao valor em Source B, esta instrução é verdadeira.

Diferente (NEQ)



Se o valor em Source A é diferente valor em Source B, esta instrução é verdadeira.

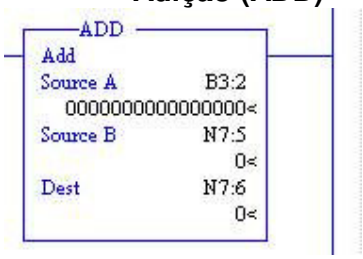
Teste de Limite (LIM)



Quando a instrução se torna verdadeira, ela testa se o valor no campo Test esta dentro ou fora de uma faixa especificada em limite inferior e limite superior. O valor da instrução é verdadeiro quando o valor de teste estiver entre os limites..

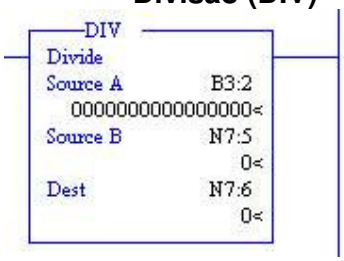
6.3 - Instruções matemáticas

Adição (ADD)



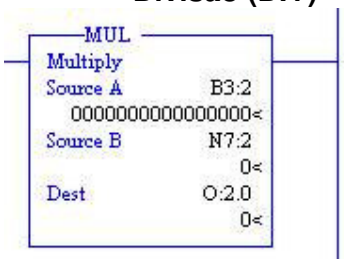
Quando a condição de entrada for verdadeira, some o valor do parâmetro Source A ao valor do parâmetro Source B e armazene o resultado no parâmetro Dest. Os dados podem ser valores ou endereços que contém valores, mas ambos não podem ser constantes.

Divisão (DIV)



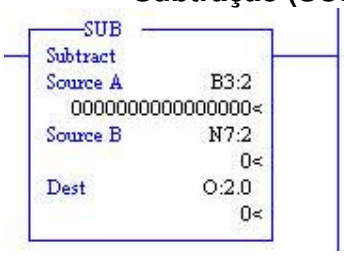
Quando a condição de entrada for verdadeira, divide o valor do parâmetro Source A pelo valor do parâmetro Source B e armazene o resultado no parâmetro Dest. Os dados podem ser valores ou endereços que contém valores, mas ambos não podem ser constantes.

Divisão (DIV)



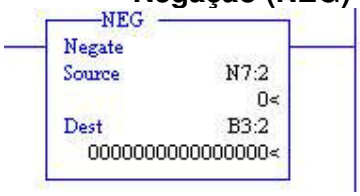
Quando a condição de entrada for verdadeira, multiplique o valor do parâmetro Source A pelo valor do parâmetro Source B e armazene o resultado no parâmetro Dest. Os dados podem ser valores ou endereços que contém valores, mas ambos não podem ser constantes.

Subtração (SUB)



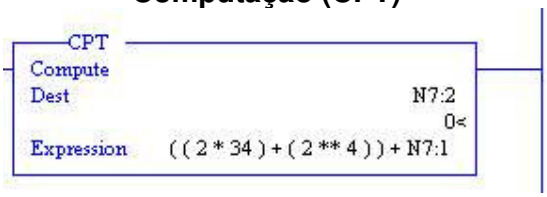
Quando a condição de entrada for verdadeira, o valor do parâmetro Source B é subtraído do valor do parâmetro Source A e o resultado é armazenado no parâmetro Dest. Os dados podem ser valores ou endereços que contém valores, mas ambos não podem ser constantes.

Negação (NEG)



Quando as condições da linha são verdadeiras, a instrução altera o sinal da origem e o coloca no destino. Os parâmetros de origem e destino devem ser endereços de palavras.

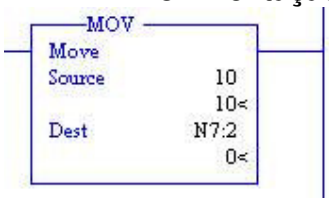
Computação (CPT)



Quando a condição de entrada for verdadeira, a operação é executada e o resultado é enviado ao destino.

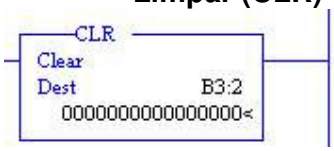
6.4 - Instruções de movimentação

Movimentação (MOV)



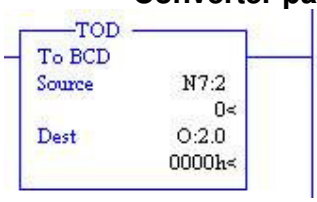
Quando a condição de entrada for verdadeira, uma cópia do parâmetro Source é movida para o parâmetro Dest. Assim, o valor original é eliminado no destino.

Limpar (CLR)



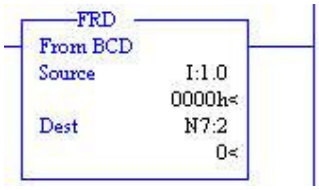
Quando a condição de entrada for verdadeira, essa instrução de saída zera todos os bits da palavra. O destino deve ser um endereço de palavra.

Converter para BCD (TOD)



Quando a condição de entrada for verdadeira, essa instrução de saída converte um valor de origem inteira de 16 bits para BCD e armazena-o no destino. Se o valor for negativo, o sinal é ignorado e a conversão ocorre como se o numero fosse positivo

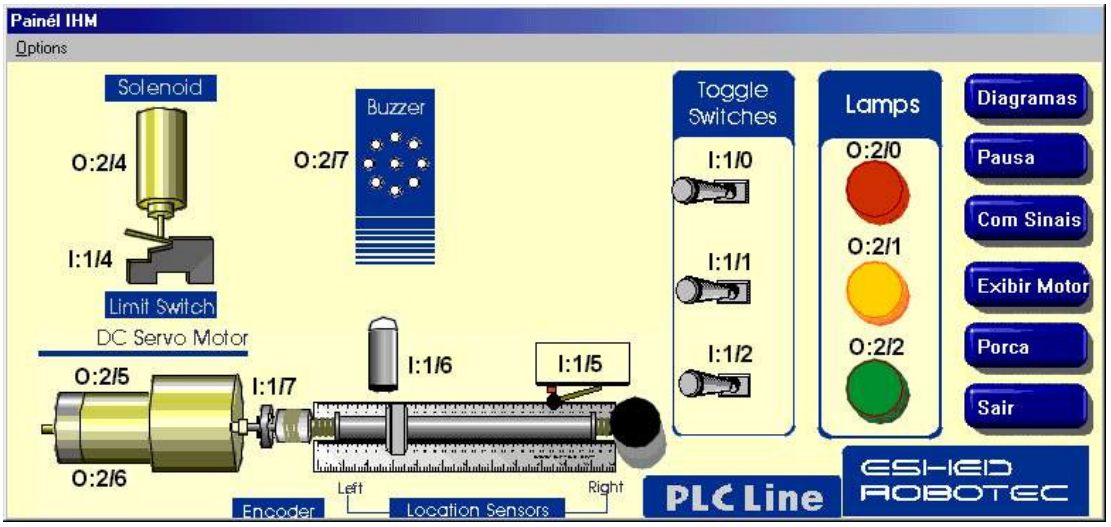
Converter de BCD para Inteiro (FRD)



Quando a condição de entrada for verdadeira, essa instrução de saída converte um valor BCD na origem para um inteiro e armazena-o no destino.

7 – Exemplos de programas

Os exemplos a seguir foram implementados em uma estrutura de hardware seguindo os endereços apontados na figura a seguir:

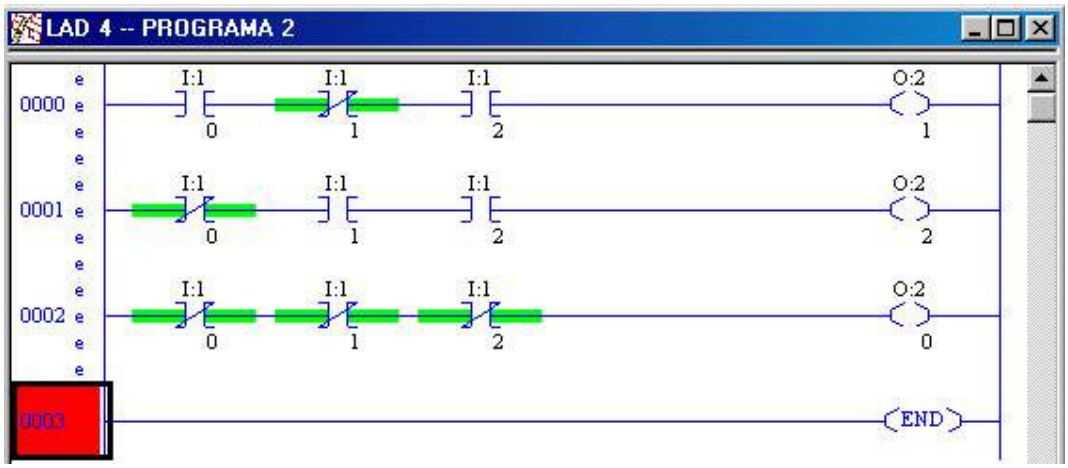


7.1 – Programa 1

O programa 1 visa nos mostrar a diferença entre as instruções Examine ON e Examine OFF. No primeiro degrau uma instrução Examine ON no endereço I:1/0 comanda o endereço de saída O:2/1. A saída será verdadeira quando a entrada for nível lógico 1, ou seja, receber tensão. Se a chave ligada na entrada for NA a saída será verdadeira quando a chave estiver acionada, mas se for NF a saída será verdadeira quando a chave não estiver acionada.

No segundo degrau uma instrução Examine OFF no endereço I:1/0 comanda o endereço de saída O:2/2. A saída será verdadeira quando a entrada for nível lógico 0, ou seja, não receber tensão. Se a chave ligada na entrada for NF a saída será verdadeira quando a chave estiver acionada, mas se for NA a saída será verdadeira quando a chave não estiver acionada.

7.2 – Programa 2



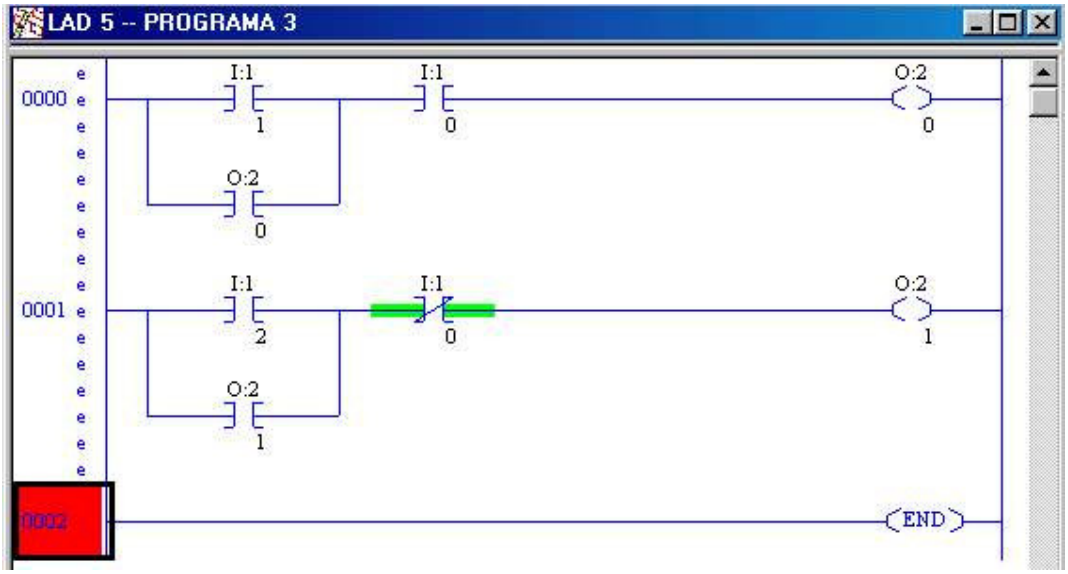
O programa 2 visa nos mostrar o funcionamento da lógica seqüencial, onde os endereços de saída serão comandados por uma seqüência de sinais de entrada. Considerando que todas as chaves são NA (I:1/0 = CH1, I:1/1 = CH2 e I:1/2 = CH3) teremos as seguintes condições para cada endereço de saída:

Para que o endereço O:2/1 seja verdadeiro é necessário que as chaves CH1 e CH3 estejam acionadas e a chave CH2 não esteja acionada.

Para que o endereço O:2/2 seja verdadeiro é necessário que as chaves CH2 e CH3 estejam acionadas e a chave CH1 não esteja acionada.

Para que o endereço O:2/0 seja verdadeiro é necessário que as chaves CH1, CH2 e CH3 não estejam acionadas.

7.3 – Programa 3

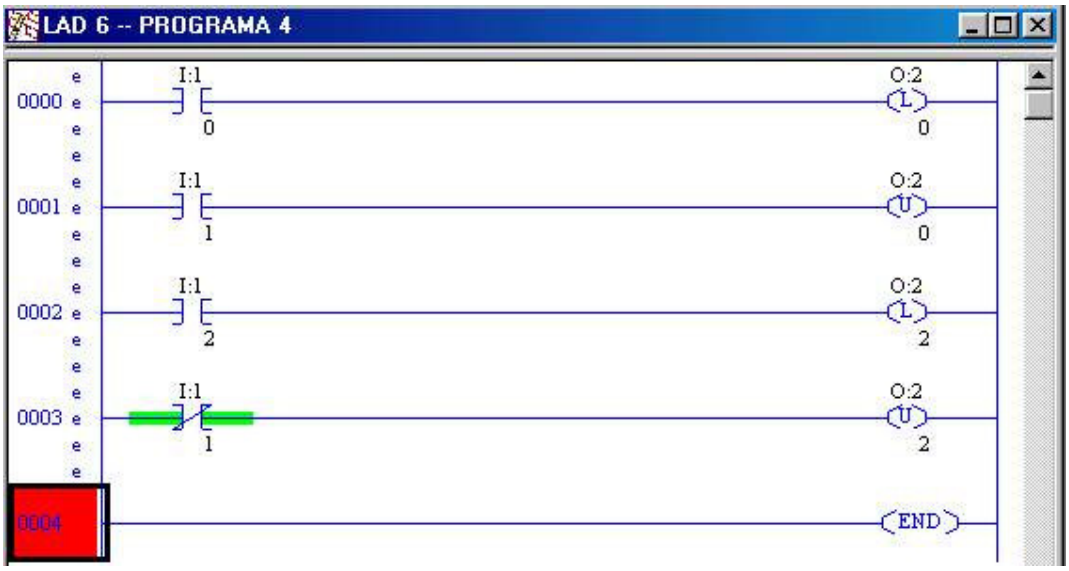


O programa 3 visa nos mostrar o funcionamento do comando por selo.

No primeiro degrau a chave de ligar (I:1/1) é NA e a chave de desligar (I:1/0) é NF. Para que o endereço de saída (O:2/0) mude de falso para verdadeiro é necessário que se pulse a chave de ligar, tornando a entrada I:1/1 verdadeira, enquanto a chave de desligar deverá estar sem acionamento. Sendo a chave de desligar NF o endereço I:1/0 será verdadeiro tornando verdadeiro o endereço O:2/0 que irá selar a saída, pois está em paralelo com o endereço da chave de ligar. Mesmo que a chave de ligar esteja sem acionamento, a saída será verdadeira até que se acione a chave de desligar, tornando falsa a saída.

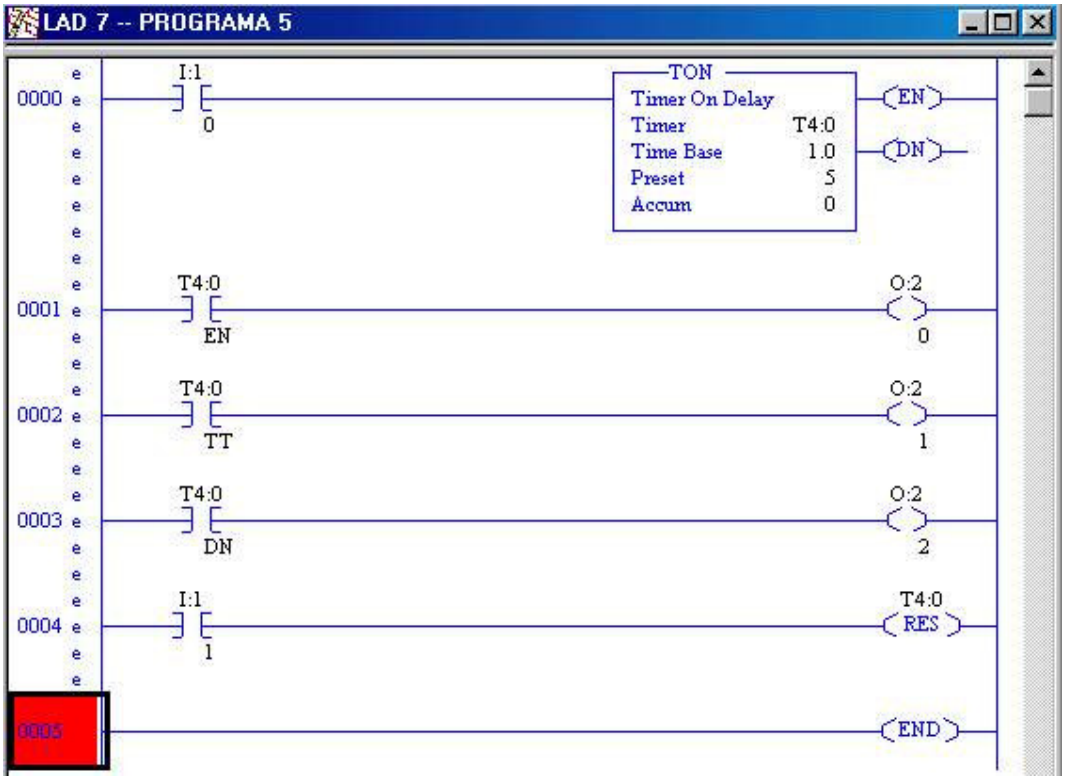
No segundo degrau a chave de ligar (I:1/1) e a chave de desligar (I:1/0) são NA. Para que o endereço de saída (O:2/0) mude de falso para verdadeiro é necessário que se pulse a chave de ligar, tornando a entrada I:1/1 verdadeira, enquanto a chave de desligar deverá estar sem acionamento. Sendo a chave de desligar NA o endereço I:1/0 será verdadeiro tornando verdadeiro o endereço O:2/0 que irá selar a saída, pois está em paralelo com o endereço da chave de ligar. Mesmo que a chave de ligar esteja sem acionamento, a saída será verdadeira até que se acione a chave de desligar, tornando falsa a saída.

7.4 – Programa 4



O programa 4 visa nos mostrar o funcionamento das instruções de retenção (L e U). Quando as condições que precedem a instrução liga (L) mudam de falso para verdadeiro o endereço desta se torna verdadeiro e permanece verdadeiro até que a instrução desliga (U) mude de falso para verdadeiro, as ações de ligar e desligar são executadas apenas com pulso, não necessita de retenção. A vantagem no uso desta instrução é que não é necessário o uso do contato de selo, porém é necessário duas lógicas para comandar um único endereço, uma lógica para ligar e uma para desligar.

7.5 – Programa 5



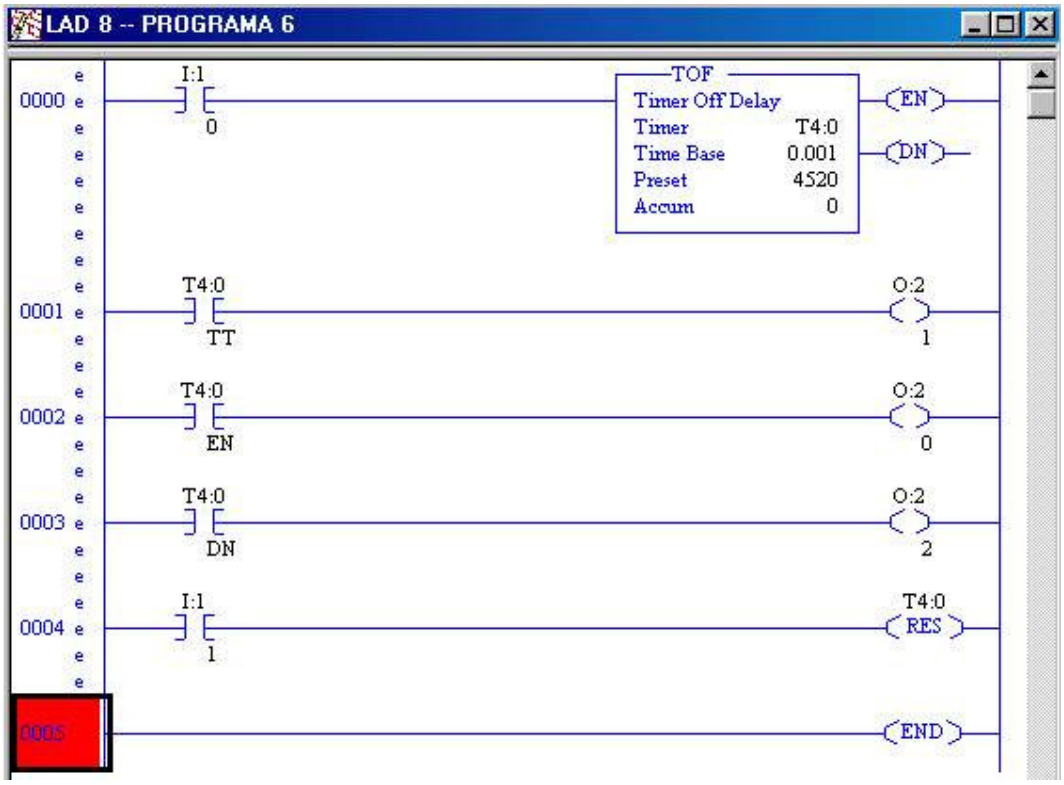
O programa 5 é um modelo de comprovação de funcionamento do temporizador para ligar (TON), onde se pode observar o funcionamento dos seus endereços auxiliares (EN, DN e TT) e da bobina reset (RES).

Quando as condições que precedem a instrução TON mudarem de falso para verdadeiro, o valor acumulado será incrementado a cada intervalo de tempo definido em time base. Quando o valor acumulado for igual ao valor do preset o bit DN será verdadeiro. Durante a contagem de tempo o bit TT é verdadeiro. O bit EN é verdadeiro quando as condições que precedem a instrução TON forem verdadeiras.

Se durante a contagem de tempo, as condições que precedem a instrução TON mudarem de verdadeiro para falso, o valor acumulado será ressetado. A instrução RES também pode ser usada para ressetar o valor acumulado.

Caso seja necessário o uso do valor acumulado em outra parte do programa seu endereço é o seguinte: T4:0.acc

7.6 – Programa 6



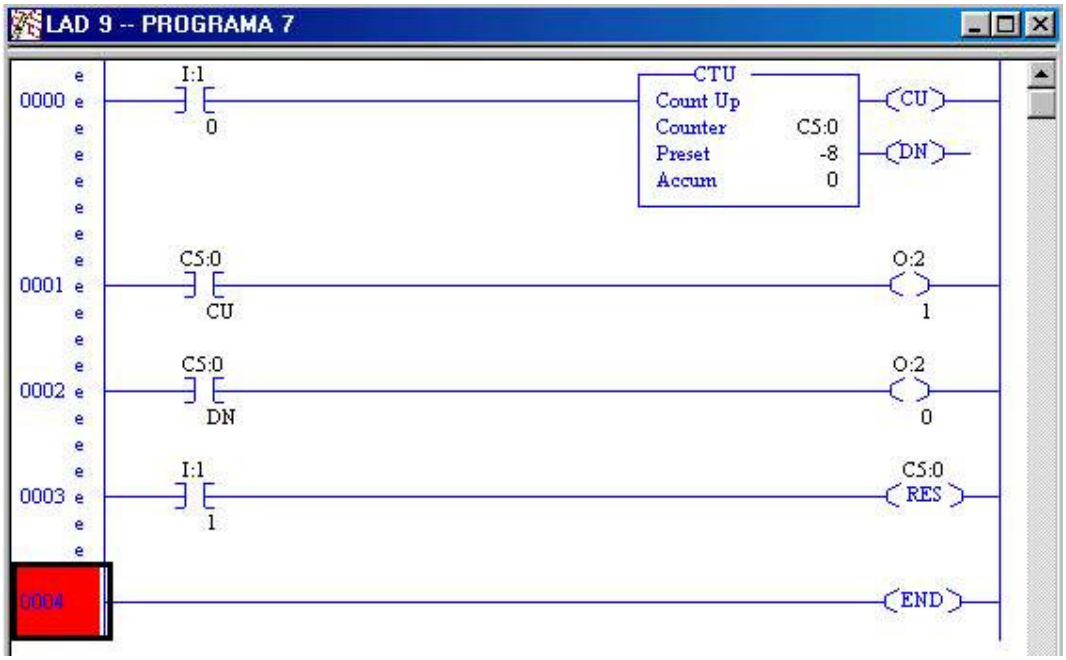
O programa 6 é um modelo de comprovação de funcionamento do temporizador para ligar (TON), onde se pode observar o funcionamento dos seus endereços auxiliares (EN, DN e TT) e da bobina reset (RES).

Quando as condições que precedem a instrução TOF mudarem de verdadeiro para falso, o valor acumulado será incrementado a cada intervalo de tempo definido em time base. Quando o valor acumulado for igual ao valor do preset o bit DN será verdadeiro. Durante a contagem de tempo o bit TT é verdadeiro. O bit EN é verdadeiro quando as condições que precedem a instrução TOF forem verdadeiras.

Se durante a contagem de tempo, as condições que precedem a instrução TOF mudarem de falso para verdadeiro, o valor acumulado será ressetado. A instrução RES também pode ser usada para ressetar o valor acumulado.

Caso seja necessário o uso do valor acumulado em outra parte do programa seu endereço é o seguinte: T4:0.acc

7.7 – Programa 7



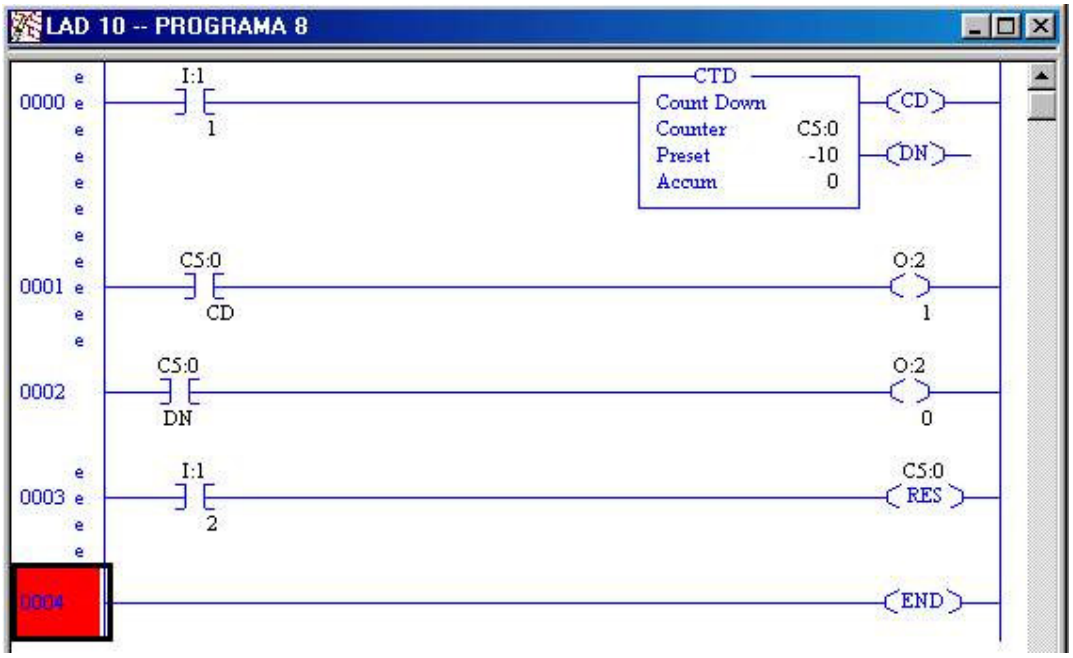
O programa 7 é um modelo de comprovação de funcionamento do Contador crescente (CTU), onde se pode observar o funcionamento dos seus endereços auxiliares (DN e CU) e da bobina reset (RES).

Quando as condições que precedem a instrução CTU mudarem de falso para verdadeiro, o valor acumulado será incrementado. Quando o valor acumulado for igual ou superior ao valor do preset o bit DN será verdadeiro. O bit CU é verdadeiro quando as condições que precedem a instrução TOF forem verdadeiras.

A instrução RES pode ser usada para ressetar o valor acumulado.

Caso seja necessário o uso do valor acumulado em outra parte do programa seu endereço é o seguinte: C5:0.acc

7.8 – Programa 8



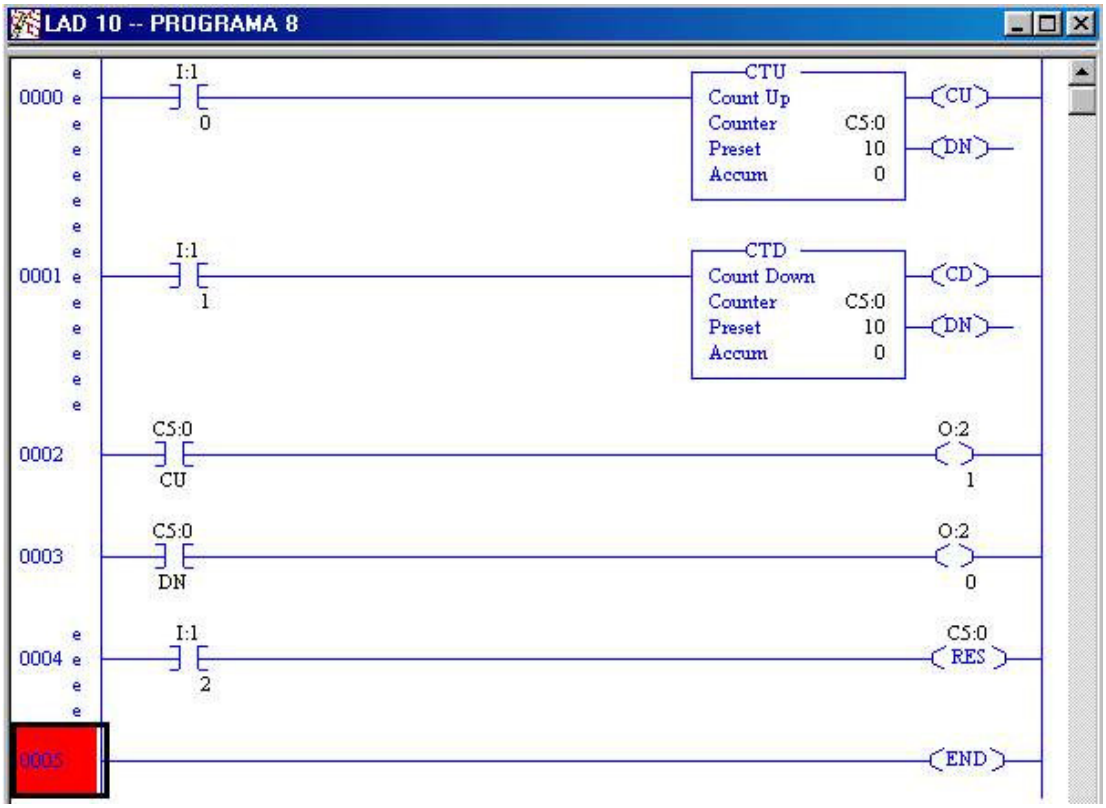
O programa 8 é um modelo de comprovação de funcionamento do Contador decrescente (CTD), onde se pode observar o funcionamento dos seus endereços auxiliares (DN e CD) e da bobina reset (RES).

Quando as condições que precedem a instrução CTD mudarem de falso para verdadeiro, o valor acumulado será decrementado. Enquanto o valor acumulado for igual ou superior ao valor do preset o bit DN será verdadeiro. O bit CD é verdadeiro quando as condições que precedem a instrução TOF forem verdadeiras.

A instrução RES pode ser usada para ressetar o valor acumulado.

Caso seja necessário o uso do valor acumulado em outra parte do programa seu endereço é o seguinte: C5:0.acc

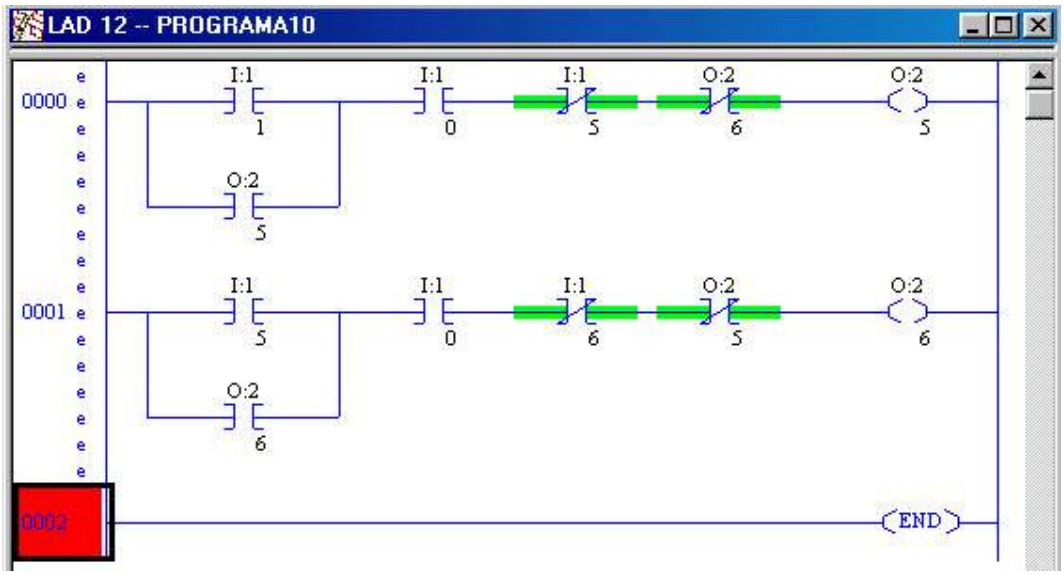
7.9 – Programa 9



O programa 9 é um modelo de comprovação de funcionamento do Contador crescente e decrescente (CTUD), que é formado a partir de dois contadores com o mesmo endereço um CTU e um CTD, onde se pode observar o funcionamento dos seus endereços auxiliares (DN e CU) e da bobina reset (RES).

Devido ao fato dos dois contadores possuírem o mesmo endereço (C5:0), eles compartilham os valores preset e acumulado. O contador crescente (CTU) é responsável pelo incremento do valor acumulado e o contador decrescente (CTD) é responsável pelo decremento. Quando o valor acumulado for maior ou igual ao valor do preset o bit DN será verdadeiro.

7.10 – Programa 10



O programa 10 é um exemplo de aplicação onde uma chave NA ligada no endereço I:1/1 ligará o motor para direita (O:2/5) e este irá girar até o limite direito (I:1/5) quando desligará o motor e o ligará para esquerda (O:2/6) até que chegue ao limite esquerdo (I:1/6) que fará o motor parar. A qualquer momento é possível parar o motor através da chave de desligar (I:1/0) que é NF.

Referências Bibliográficas

MANUAL TÉCNICO ALENN BRADLEY, 1785-6.8.2

GEORGINI, Marcelo, Automação Aplicada – Descrição e Implementação de Sistemas Seqüenciais com PLCs - 5ª Edição. São Paulo: editora Erica,2000

Edições FIEMG. CLP