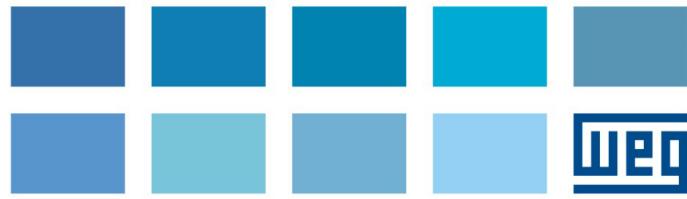


Centro de Treinamento de Clientes

MÓDULO 3 *Automação de Processos Industriais*



* “Material sujeito a alterações sem prévio aviso!”

Índice Analítico

| | |
|---|-----------|
| Controladores Lógicos Programáveis..... | 3 |
| Introdução | 3 |
| A Automação Industrial..... | 4 |
| Noções de Lógica Combinacional | 5 |
| 1.1.1 Operações Fundamentais | 5 |
| 1.1.1.1 Funções BOOLEANAS | 5 |
| 1.1.1.2 Operador “AND“ | 6 |
| 1.1.1.3 Operador “OR“ | 6 |
| 1.1.1.4 Operador “NOT“ | 6 |
| 1.1.1.5 Operador “NAND“ | 7 |
| 1.1.1.6 Operador “NOR“ | 7 |
| 1.1.1.7 Operador “XOR“ | 7 |
| Tipos de Sinais..... | 8 |
| 1.1.2 Sinais Analógicos | 8 |
| 1.1.3 Sinais Digitais | 8 |
| 1.1.3.1 Single bit | 8 |
| 1.1.3.2 Multi bit | 8 |
| Definição (IEC 1131-1)..... | 8 |
| Princípio de Funcionamento | 9 |
| Aspectos de Hardware | 9 |
| 1.1.4 Fonte de alimentação | 10 |
| 1.1.5 CPU | 10 |
| 1.1.6 Memórias | 11 |
| 1.1.7 Interfaces de Entrada/Saída | 11 |
| 1.1.8 Periféricos | 14 |
| 1.1.8.1 Terminal inteligente | 14 |
| 1.1.8.2 Microcomputadores | 15 |
| 1.1.8.3 Mini-programadores (terminais de bolso) | 15 |
| 1.1.8.4 Outros periféricos | 15 |
| 1.1.8.9 Interfaceamento de periféricos | 16 |
| Aspectos de Software | 16 |
| 1.1.10 Linguagens de programação | 17 |
| Sistemas Associados | 19 |
| 1.1.11 Redes de comunicação | 19 |
| 1.1.12 Supervisão e controle | 26 |
| Anexos | 30 |
| 1.1.13 - Manual do TPW3-PCLINK | 30 |
| 1.1.14 - Programação da Interface Homem-Máquina OP-07/OP-08 | 46 |
| 1.1.15 - Exercícios propostos | 61 |

CONTROLADORES LÓGICOS PROGRAMÁVEIS

Introdução

Esta apostila tem como objetivo prover uma visão geral das características e recursos hoje disponíveis no mercado de Controladores Programáveis (CP's), bem como, a sua aplicação nos diversos campos da automação industrial e controle de processos, onde as necessidades de flexibilidade, versatilidade, disponibilidade, alta confiabilidade, modularidade, robustez e baixos custos, o tornam uma excelente opção.

Mas, o que é um Controlador Programável ? Como surgiu ?

Mesmo antes da industrialização da eletrônica digital, os projetistas de comando elaboravam circuitos digitais como contatos programáveis. O programa era armazenado em plugs multi-pinôs e as instruções codificadas por meio de ligações elétricas entre os pinos destes plugs. Esses programas eram muito limitados, e, sua principal função era a seleção das operações das máquinas e/ou processos.

Desta forma, além de uma operacionalidade muito baixa, existiam outros problemas: alto consumo de energia, difícil manutenção, modificações de comandos dificultados e onerosos com muitas alterações na fiação ocasionando número de horas paradas, além das dificuldades em manter documentação atualizada dos esquemas de comando modificado.

Com a industrialização da eletrônica, os custos diminuíram, ao mesmo tempo em que a flexibilidade aumentou, permitindo a utilização de comandos eletrônicos em larga escala.

Mas alguns problemas persistiram, e quem sentia estes problemas de forma significativa era a indústria automobilística, pois a cada ano com o lançamento de novos modelos, muitos painéis eram sucateados pois os custos para alteração eram maiores do que a instalação de novos painéis.

Porém, em 1968 a GM através de sua Divisão Hidromatic preparou as especificações detalhadas do que posteriormente denominou-se Controlador Programável (CP). Estas especificações retratavam as necessidades da indústria, independentemente do produto final que iria ser fabricado. Em 1969 foi instalado o primeiro CP na GM executando apenas funções de intertravamento.

Historicamente os CP's tiveram a seguinte evolução: De 1970 a 1974, em adição às funções intertravamento e sequenciamento (lógica), foram acrescentadas funções de temporização e contagem, funções aritméticas, manipulação de dados e introdução de terminais de programação de CRT (Cathode Ray Tube).

De 1975 a 1979 foram incrementados ainda maiores recursos de software que propiciaram expansões na capacidade de memória, controles analógicos de malha fe-

chada com algoritmos PID, utilização de estações remotas de interfaces de E/S (Entradas e Saídas) e a comunicação com outros equipamentos “inteligentes”.

Com os desenvolvimentos deste período, o CP passou a substituir o microcomputador em muitas aplicações industriais.

Nesta década atual, através dos enormes avanços tecnológicos, tanto de hardware como de software, podemos dizer que o CP evoluiu para o conceito de controlador universal de processos, pois pode configurar-se para todas as necessidades de controle de processos e com custos extremamente atraentes.

A Automação Industrial

Antes de iniciarmos nosso estudo dos controladores programáveis, precisamos sedimentar alguns conceitos importantes. Um destes conceitos está relacionado com às respostas para algumas perguntas :

O que é controle ?

Conforme o dicionário (Aurélio Buarque de Holanda Ferreira) podemos definir a palavra controle como segue :

[Do fr. contrôle.] S. m. 1. Ato, efeito ou poder de controlar; domínio, governo. 2. Fiscalização exercida sobre as atividades de pessoas, órgãos, departamentos, ou sobre produtos, etc., para que tais atividades, ou produtos, não se desviam das normas preestabelecidas.

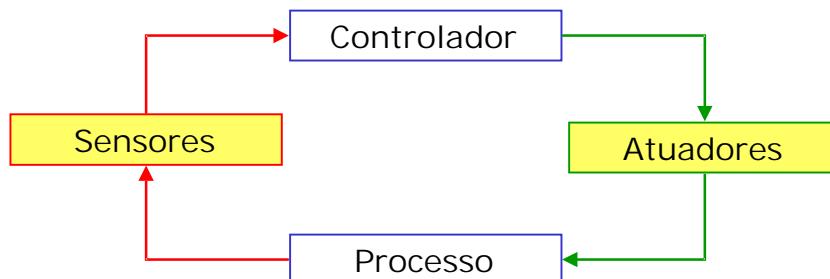


Figura 1- Diagrama de blocos de um sistema de automação

O controle, vendo sob o ponto de vista tecnológico, tem um papel importantíssimo no desenvolvimento de ações planejadas, modelando processos desde os mais simples até os mais complexos.

O que é automação industrial ?

Todas as vezes, relacionado a um processo, que introduzimos alguma nova técnica de controle estamos falando de automação industrial. Na verdade a utilização destas técnicas estará diretamente relacionada com o aumento de produtividade, qualidade, flexibilidade e confiabilidade. Note que o termo automação descreverá um conceito muito amplo, envolvendo um conjunto de técnicas de controle, das quais criamos um sistema ativo, capaz de fornecer a melhor resposta em funções das informações que recebe do processo em que está atuando. Dependendo das informações o sistema irá calcular a melhor ação corretiva à ser executada. Neste ponto podemos verificar as características relacionadas com os sistemas em malha fechada, também denominados sistemas realimentados (ver figura 1). A teoria clássica de controle define e modela, matematicamente, estas características dando uma conotação científica e tecnológica a este assunto.

Noções de Lógica Combinacional

Nesta seção iremos trabalhar alguns conceitos importantes para o desenvolvimento de um processo lógico de raciocínio que mas adiante nos permitirá compreender como serão relacionados todos os fatores relevantes à elaboração de projetos envolvendo controladores programáveis.

1.1.1 OPERAÇÕES FUNDAMENTAIS

A teoria matemática das proposições lógicas foi apresentada em 1854⁽¹⁾, pelo filósofo e matemático inglês George Boole (1815-1864), definindo assim os conceitos básicos da chamada álgebra de Boole para dois valores (sistema binário). Mas foi sómente em 1938⁽²⁾, que o engenheiro americano Claude Elwood Shannon, aplicou a teoria de Boole ao estudo e simplificação de funções usadas em telefonia, percebendo que as leis que regem as relações entre proposições lógicas eram as mesmas que se aplicavam para dispositivos de chaveamento de dois estados, já que estes dispositivos podem assumir os seguintes estados, como por exemplo : “ligado” ou “desligado”, “aberto” ou “fechado”, “potencial alto” ou “potencial baixo”, “verdadeiro” ou “falso”.

(1) Intitulado como *An Investigation of the Laws of Thought*

(2) Trabalho intitulado como *Symbolic Analysis of Relay and Switching*

1.1.1.1 FUNÇÕES BOOLEANAS

A álgebra de Boole está estruturada da seguinte maneira : Um conjunto **S**; três operações definidas sobre **S** (operação **E**, **OU** e **COMPLEMENTO**); Os caracteres **0** e **1**. Não abordaremos de forma detalhada os teoremas, postulados e leis desta teoria.

Mas a idéia de uma função lógica segue o mesmo conceito das funções da álgebra tradicional, onde uma função assume um único valor para cada combinação de valores possíveis assumidos pelas suas variáveis. Note, que na realidade uma função lógica (booleana) com n variáveis irá apresentar um total de combinações dadas por 2^n . Se adotarmos um procedimento formal para análise dos valores possíveis para uma função booleana chegaremos a conclusão que o processo seria bastante cansativo e

muito susceptível a erros, relacionados basicamente com a falta de atenção. Para facilitar esta análise foi proposta, pelo matemático inglês Charles Lutwidge Dodgson⁽³⁾ (1832-1898), uma forma tabular de representação conhecida como tabela verdade (truth table). A seguir mostraremos as equações algébricas e a tabela verdade dos operadores fundamentais da álgebra booleana.

(3) Cujo pseudônimo era Lewis Carroll, nome adotado quando escreveu o livro *Alice no País das Maravilhas*

1.1.1.2 OPERADOR “AND”

| Equação Algébrica | Tabela Verdade | | | | | | | | | | | | | | | |
|-------------------|---|------------------------|---|------------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| $F = A \cdot B$ | <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>$F = A \text{ AND } B$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> | A | B | $F = A \text{ AND } B$ | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| A | B | $F = A \text{ AND } B$ | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | |

1.1.1.3 OPERADOR “OR”

| Equação Algébrica | Tabela Verdade | | | | | | | | | | | | | | | |
|-------------------|--|-----------------------|---|-----------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| $F = A + B$ | <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>$F = A \text{ OR } B$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> | A | B | $F = A \text{ OR } B$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| A | B | $F = A \text{ OR } B$ | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | |

1.1.1.4 OPERADOR “NOT”

| Equação Algébrica | Tabela Verdade | | | | | | |
|-------------------|---|---|---------------------|---|---|---|---|
| $F = \bar{A}$ | <table border="1"> <thead> <tr> <th>A</th> <th>$F = \text{NOT } A$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table> | A | $F = \text{NOT } A$ | 0 | 1 | 1 | 0 |
| A | $F = \text{NOT } A$ | | | | | | |
| 0 | 1 | | | | | | |
| 1 | 0 | | | | | | |

1.1.1.5 OPERADOR “NAND”

| Equação Algébrica | Tabela Verdade | | | | | | | | | | | | | | | |
|----------------------------|---|-------------------------|---|-------------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| $F = \overline{A \cdot B}$ | <table border="1"> <thead> <tr> <th>A</th><th>B</th><th>$F = A \text{ NAND } B$</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | A | B | $F = A \text{ NAND } B$ | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| A | B | $F = A \text{ NAND } B$ | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | |

1.1.1.6 OPERADOR “NOR”

| Equação Algébrica | Tabela Verdade | | | | | | | | | | | | | | | |
|--|--|------------------------|---|------------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| $F = \overline{\overline{A} + \overline{B}}$ | <table border="1"> <thead> <tr> <th>A</th><th>B</th><th>$F = A \text{ NOR } B$</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | A | B | $F = A \text{ NOR } B$ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| A | B | $F = A \text{ NOR } B$ | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | |

1.1.1.7 OPERADOR “XOR”

| Equação Algébrica | Tabela Verdade | | | | | | | | | | | | | | | |
|---|--|------------------------|---|------------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| $F = \overline{A} \cdot B + A \cdot \overline{B}$ | <table border="1"> <thead> <tr> <th>A</th><th>B</th><th>$F = A \text{ XOR } B$</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | A | B | $F = A \text{ XOR } B$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| A | B | $F = A \text{ XOR } B$ | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | |

TIPOS DE SINAIS

1.1.2 SINAIS ANALÓGICOS

São sinais que variam continuamente no tempo conforme uma regra de comparação à uma referência definida.

Exemplos : potenciômetros, transdutores de temperatura, pressão, célula de carga, umidade, vazão, medidores, válvulas e atuadores analógicos, acionamentos de motores, etc.

1.1.3 SINAIS DIGITAIS

São sinais que variam continuamente no tempo assumindo apenas dois valores definidos e distintos. Podemos ainda encontrá-los subdivididos em dois tipos :

1.1.3.1 SINGLE BIT

Dispositivos deste tipo apresentam sinais que poderão ser representados por bits individuais.

Exemplos : botões, chaves seletoras, chaves fim-de-curso, pressostatos, termostatos, chaves de nível, contatos de relês, contatos auxiliares de contatores, alarmes, solenóides, lâmpadas, bobinas de relês, bobinas de contatores, etc.

1.1.3.2 MULTIBIT

Dispositivos deste tipo apresentam sinais representados por bits agrupados em conjunto, formando assim o que chamamos de “palavra binária”.

Exemplos : encoder absoluto, chave thumbwheel, etc.

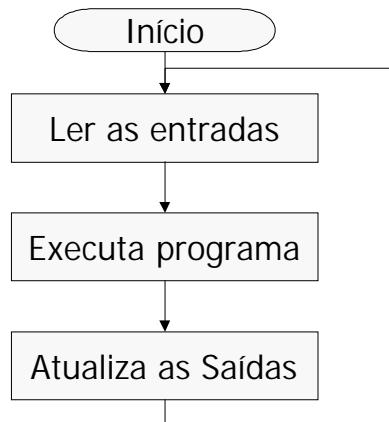
Definição de um controlador programável (IEC 61131-1)

Sistema eletrônico digital, desenvolvido para uso em ambiente industrial, que usa uma memória programável para armazenamento interno de instruções do usuário, para implementação de funções específicas, tais como, lógica, sequenciamento, temporização, contagem e aritmética, para controlar, através de entradas e saídas, vários tipos de máquinas e processos.

O CP e seus periféricos, ambos associados, são projetados de forma a poder ser integrados dentro de um sistema de controle industrial e finalmente usados a todas as funções a quais é destinado.

PRINCÍPIO DE FUNCIONAMENTO

O Controlador Programável, como todo sistema microprocessado, tem seu princípio de funcionamento baseado em três passos:



Com a partida, o CP executará as seguintes tarefas:

1. Transferirá os sinais existentes na interface de entrada para a memória de dados (RAM).
2. Iniciará a varredura do software aplicativo armazenando na memória de programa (SCAN), utilizando os dados armazenados na memória de dados. Dentro deste ciclo, executará todas as operações que estavam programadas no software aplicativo, como intertravamentos, habilitação de temporizadores/contadores, armazenagem de dados processados na memória de dados, etc...
3. Concluída a varredura do software aplicativo, o CP transferirá os dados processados (resultados de operações lógicas) para a interface de saída. Paralelamente, novos dados provenientes da interface de entrada irão alimentar a memória de dados.

ASPECTOS DE HARDWARE

O diagrama de blocos abaixo representa a estrutura básica de um controlador programável com todos os seus componentes. Estes componentes irão definir o que denominamos configuração do CLP.

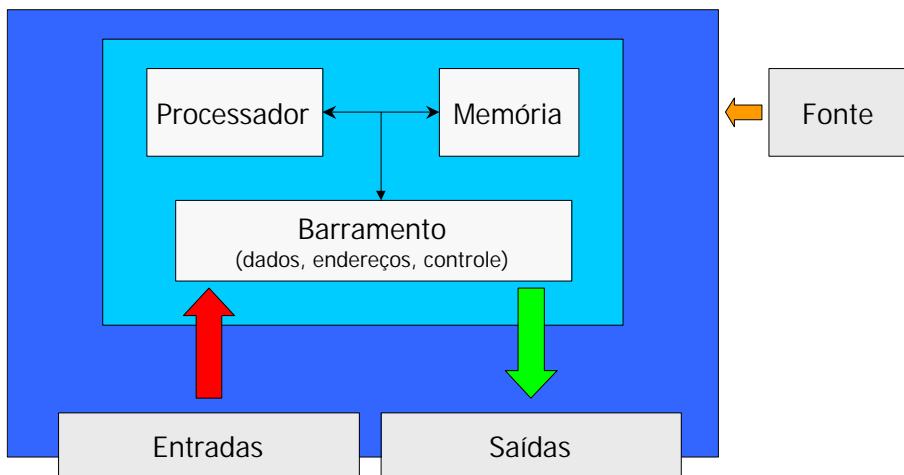


Figura 2 - Diagrama de blocos simplificado de um controlador programável

1.1.4 FONTE DE ALIMENTAÇÃO

A fonte fornece todos os níveis de tensão exigidos para as operações internas do CP (Ex.: CPU, Memória, E/S).

1.1.5 CPU

A CPU é o cérebro do sistema. Ela lê o sinal das entradas na memória de dados, executa operações aritméticas e lógicas baseadas na memória de programa, e gera os comandos apropriados para a memória de dados controlar o estado das saídas.

Abaixo são apresentadas algumas considerações e características principais:

Utiliza microprocessadores ou microcontroladores de 8,16 ou 32 bits e, em CP's maiores, um coprocessador (microprocessador dedicado) adicional para aumentar a capacidade de processamento em cálculos complexos com aritmética de ponto flutuante.

A maioria dos fabricantes de CP's especificam os tempos de varredura como função do tamanho do programa (p.e. 10ms/1k de programa), e situam-se na faixa de 0,3 até 10ms/k, caracterizando a existência de CP's rápidos e lentos.

Alguns fabricantes provêem recursos de hardware e software que possibilitam interrupções na varredura normal de forma a “ler” uma entrada ou “atualizar” uma saída imediatamente.

Recursos de auto-diagnose para detecção e indicação de falhas (Comunicação, memória, bateria, alimentação, temperatura, etc.) são também disponíveis em alguns CP's. Normalmente os indicadores estão localizados na parte frontal do cartão da UCP.

1.1.6 MEMÓRIAS

Memória de Dados : também conhecida como memória de rascunho. Serve para armazenar temporariamente os estados E/S, marcadores presets de temporizadores/ contadores e valores digitais para que o CPU possa processá-los. A cada ciclo de varredura a memória de dados é atualizada. Geralmente memória RAM.

Memória de Usuário : serve para armazenar as instruções do software aplicativo e do usuário (programas que controlam a máquina ou a operação do processo), que são continuamente executados pela CPU. Pode ser memória RAM, EPROM, E-PROM, NVRAM ou FLASH-EPROM.

1.1.7 INTERFACES DE ENTRADA/SAÍDA

O hardware, de E/S, freqüentemente chamado de módulos de E/S, é a interface entre os dispositivos conectados pelo usuário e a memória de dados. Na entrada, o módulo de entrada aceita as tensões usuais de comando (24VCC,110/220 VCA) que chegam e as transforma em tensões de nível lógico aceitos pela CPU. O módulo de saída comuta as tensões de controle fornecidas, necessárias para acionar vários dispositivos conectados.

Os primeiros CP's, como já mencionado anteriormente, eram limitados a interfaces de E/S discretas, ou seja, admitiam somente a conexão de dispositivos do tipo ON/OFF (liga/desliga, aberto/fechado, etc.), o que, naturalmente, os limitavam um controle parcial do processo, pois, variáveis como temperatura, pressão, vazão, etc., medidas e controladas através de dispositivos operados normalmente com sinais analógicos, não eram passíveis de controle. Todavia, os CP's de hoje, provêem de uma gama completa e variada de interfaces discretas e analógicas, que os habilitam a praticamente qualquer tipo de controle.

As entradas e saídas são organizadas por tipos e funções, e agrupadas em grupos de 2, 4, 8, 16 e até 32 “pontos” (circuitos) por interface (cartão eletrônico) de E/S. Os cartões são normalmente do tipo de encaixe e, configuráveis, de forma a possibilitar uma combinação adequada de pontos de E/S, digitais e analógicas.

A quantidade máxima de pontos de E/S, disponíveis no mercado de CP's, pode variar desde 16 a 8192 pontos normalmente, o que caracteriza a existência de pequenos, médios e grandes CP's.

Embora uma classificação de CP's devesse considerar a combinação de diversos aspectos (n.º de pontos de E/S, capacidade de memória, comunicação, recursos

de software e programação, etc.), para propósitos práticos, podemos considerar a seguinte classificação: Micro e Mini CP's; CP's de pequeno porte; CP's de médio porte; CP's de grande porte.

As figuras abaixo apresentam uma idéia básica de como estas interfaces são implementadas.

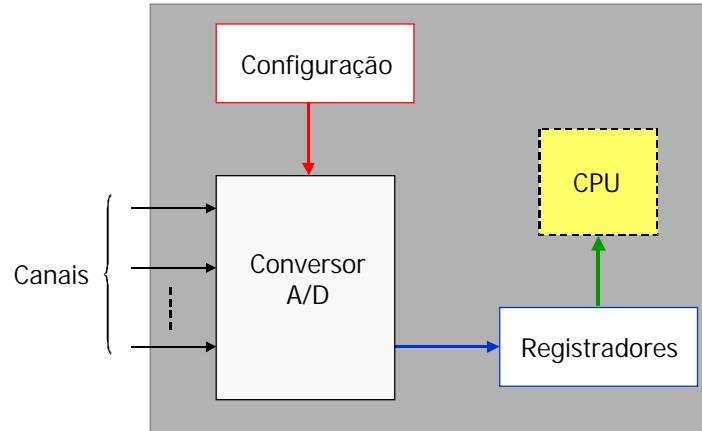


Figura 3 – Configuração típica de uma interface de entrada analógica.

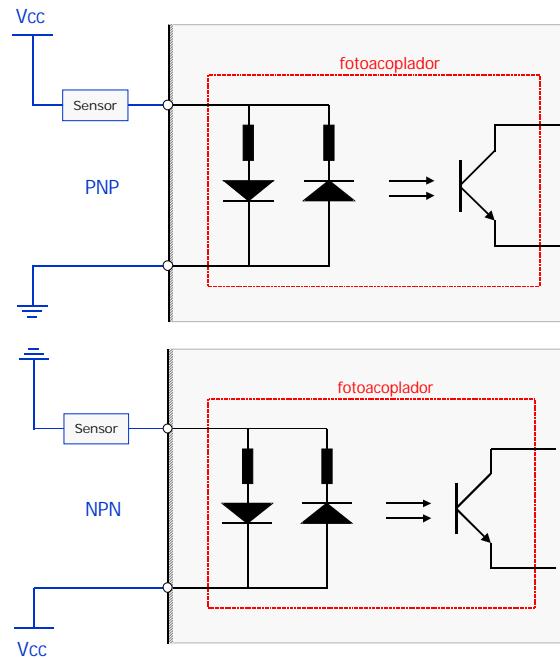


Figura 4 – Configurações típicas para interfaces de entrada digital.

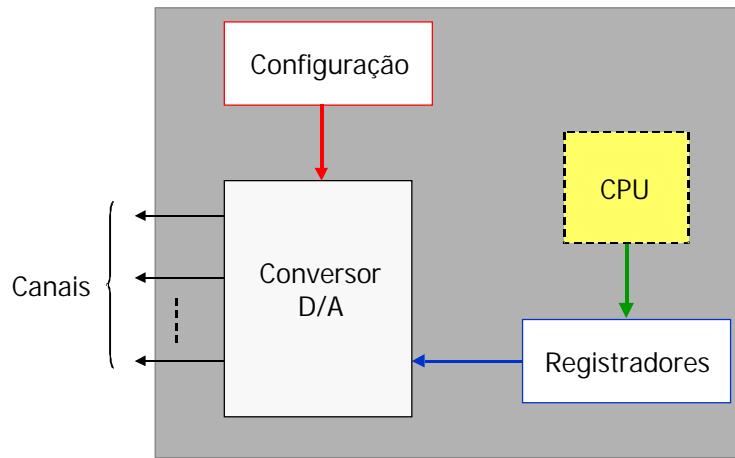


Figura 5 – Configuração típica para interfaces de saída analógica.

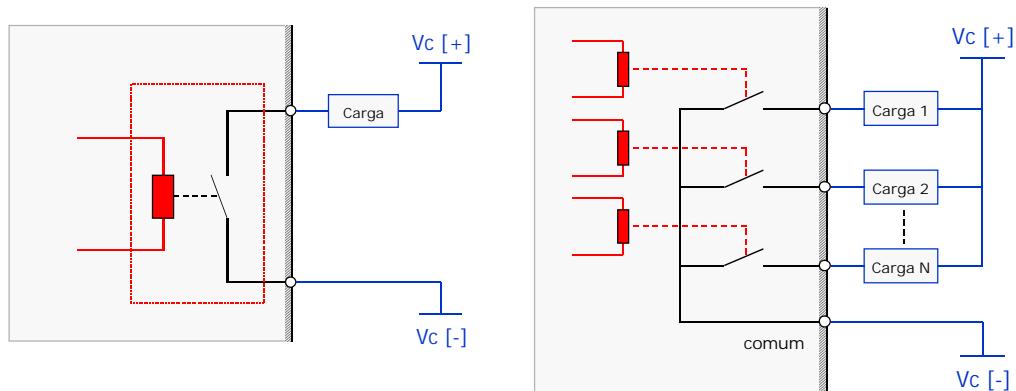


Figura 6 – Configurações típicas para interfaces de saída digital a relé.

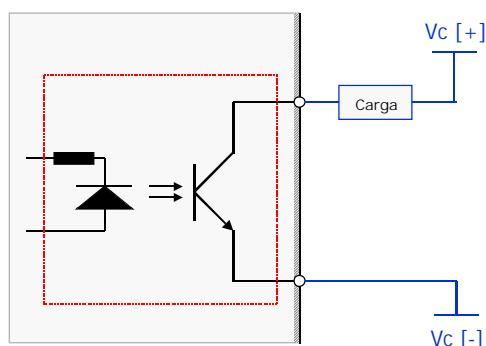


Figura 7 – Configuração típica para uma interface de saída digital a transistor.

1.1.8 PERIFÉRICOS

Dentre os diversos equipamentos periféricos ao CP's podemos destacar os de programação, que basicamente, tem por finalidade principal a introdução do programa de controle na memória do CP e a visualização e documentação do mesmo.

Os equipamentos de programação mais comumente utilizados são os seguintes:

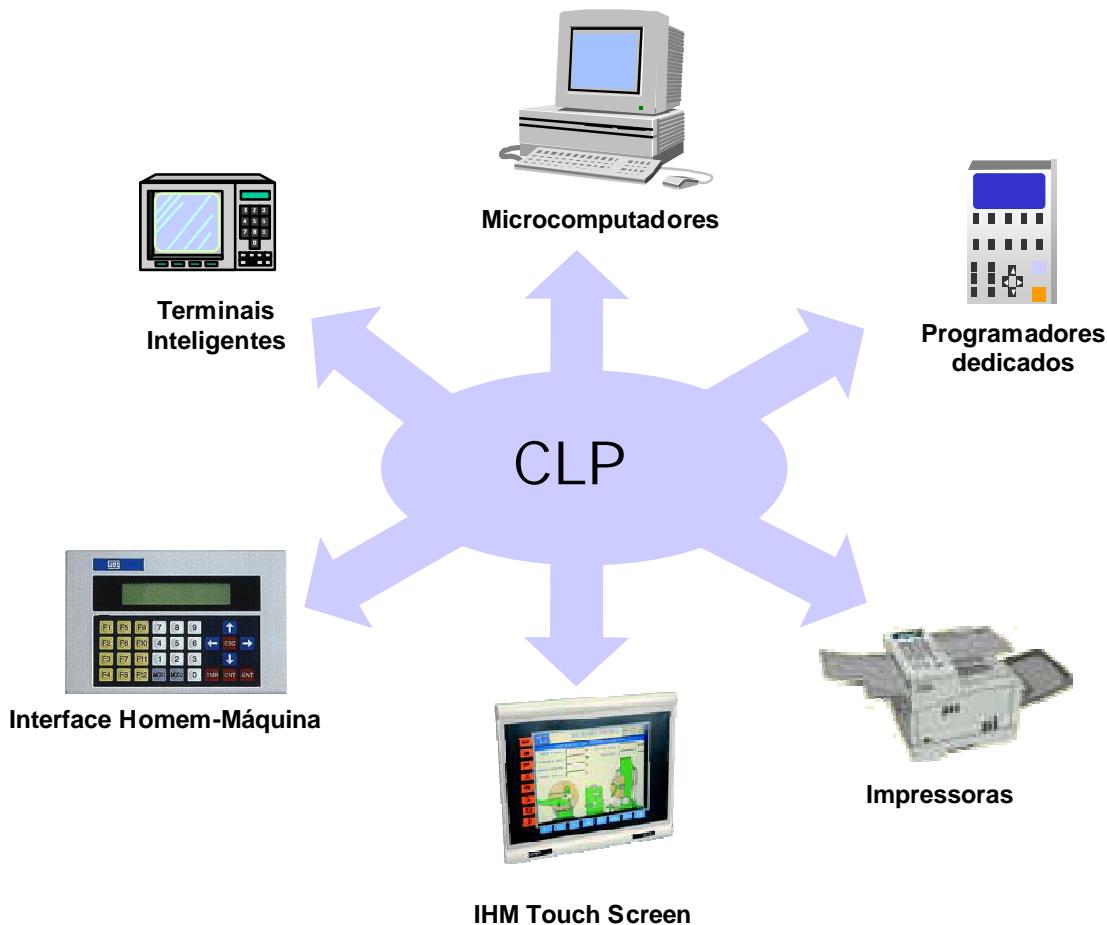


Figura 8- Periféricos mais utilizados

1.1.8.1 TERMINAL INTELIGENTE

Sendo microprocessado, é capaz de executar funções de edição de programas e outras independentemente da UCP do controlador. Ele possui sua própria me-

mória com software para criação, alteração e monitoração dos programas. A grande vantagem é a de poder também editar e armazenar os programas de controle sem estar acoplados ao CP. Esta capacidade é conhecida como programação “off-line”.

Em geral, estes terminais possuem acionadores de "Floppy-Disks" (discos flexíveis) e programadores de EPROM's o que possibilita também o arquivo de programas tanto em Floppy-Disks como em EPROM's.

Alguns terminais possuem ainda uma interface de rede o que permite acoplá-los às redes locais de comunicação. Este arranjo permite o terminal acessar qualquer CP na rede, alterar parâmetros ou programas, e monitorar quaisquer elementos sem estar acoplado diretamente a qualquer CP. Com software adequado, este arranjo pode permitir também um meio centralizado de aquisição e apresentação, inclusive gráfica, dos dados dos diferentes controladores da rede.

Uma desvantagem, é que estes terminais não são intercambiáveis entre diferentes fabricantes de CP's.

1.1.8.2 MICROCOMPUTADORES

Com o advento dos microcomputadores pessoais (PC's) e com a crescente utilização dos mesmos em ambientes industriais, a grande maioria dos fabricantes desenvolveram software especiais que possibilitaram utilizá-los também como programadores tanto “on line” como “off line”. A grande maioria destes software foram desenvolvidos com base na linha de micros compatíveis com os IBM-PC's, facilitando inclusive a compilação de programas em linguagens de alto nível (BASIC, C, PASCAL, etc.).

Há atualmente uma acentuada utilização destes equipamentos com CP's, principalmente como Interface Homem-Máquina/Processo no nível de Supervisão do controle de processos, tema este que abordaremos no capítulo 6.

1.1.8.3 MINI-PROGRAMADORES (TERMINAIS DE BOLSO)

São bastante compactos, assemelhando-se em muito com as calculadoras de mão. Este equipamento é preferencialmente utilizado para aplicação no campo, para testes e parametrização.

1.1.8.4 OUTROS PERIFÉRICOS

Ainda dentro da família de equipamentos periféricos aos CP's podemos destacar os seguintes:

INTERFACE HOMEM/MÁQUINA: Com dimensões reduzidas, são utilizados principalmente para introdução e visualização de dados e mensagens. São compostos

de um teclado numérico-funcional, muitas vezes do tipo membrana, e de display alfanumérico, sendo gerenciados por um microprocessador.

IMPRESSORAS: São utilizadas normalmente para prover cópia do programa de controle e geração de relatórios e mensagens ao operador. A comunicação é feita normalmente através de interfaces de comunicação serial padrão RS 232C.

1.1.9 INTERFACEAMENTO DE PERIFÉRICOS

COMUNICAÇÃO SERIAL: É a mais comumente utilizada para a maioria dos periféricos e é feita utilizando-se simples cabos de par traçado. Os padrões mais utilizados são o RS 232C, loop de corrente 20mA, e o RS-422/RS-485 em alguns casos.

RS-232C: Este padrão define basicamente as características dos sinais elétricos, bem como os detalhes mecânicos (pinagem) da interface.

É empregada para velocidades de transmissão de até 20k baud (bits/seg) e distância máxima de 15 metros. (Com a utilização dos modems esta distância pode ser ampliada).

RS-422/RS-485: É uma versão melhorada do padrão RS-232C. Ela possibilita, principalmente, o emprego de velocidade de transmissão de até 100k baud para distância de até 1200m, podendo alcançar velocidades da ordem de MBaud para distâncias menores.

LOOP DE CORRENTE 20mA: A interface de loop de corrente é idêntica a RS-232C e, evidentemente como é baseada em níveis de corrente em vez de tensão, possibilita o emprego em distâncias bem maiores. Muitos CP's oferecem ambos os padrões, RS-232C e loop de corrente.

Aspectos de Software

Além do número de pontos de E/S, o que determina a utilização de um CP são os recursos de software disponíveis, isto é, que funções ele pode executar. Todos os CP's possuem as seguintes funções básicas de software :

- Lógica E, OU e XOR;
- SET e RESET
- Temporização e contagem;
- Cálculos com aritmética básica (+,-,x,÷);
- Parênteses (para associação de lógicas);
- Comparação de valores;
- Registrador de deslocamento;
- Salto.

A medida que os CP's tem sua capacidade de processamento aumentada, surge a necessidade de funções de software mais avançadas, tais como:

- Cálculos com ponto flutuante;
- Cálculos integrais e trigonométricos;
- Malhas de controle PID;
- Posicionamento;
- Contagem rápida;
- Leitura de sinais analógicos;
- Leitura de sinais de temperatura;
- Linearização de sinais analógicos;
- Lógica fuzzy;
- Outros.

1.1.10 LINGUAGENS DE PROGRAMAÇÃO

A programação traduz as funções a serem executadas; para tanto ela deve ser a mais simples possível. Utilizando-se de linguagem específica, baseando-se na metotécnica, a linguagem de programação usa abreviações, figuras e números de tal forma a formar-se acessível a todos os níveis tecnológicos.

Os tipos de funções são associações lógicas (“E”, “OU”, etc), funções de memória (SET, RESET, etc), funções de contagem, temporização, aritméticas e outras mais específicas. A forma visual que a instrução se apresenta depende unicamente do tipo de sistema utilizado pelo programador. Seja por exemplo, a associação lógica “OU” entre duas informações que chamaremos de entradas por traduzirem informações do processo. O resultado desta associação será armazenado em uma memória para depois ser utilizado, na dependência da ordem de operação. Podemos representar essa associação na forma de diagrama de contatos (Ladder).

Podemos ainda representar a associação através de um esquema de funcionamento ou diagrama lógico.

As vantagens e desvantagens de cada uma das formas de linguagem de programação são dependentes dos conhecimentos do programador.

A linguagem mais difundida até agora tem sido o diagrama de contatos (LADDER), devido a semelhança com os esquemas elétricos usados para o comando convencional e a facilidade de visualização nas telas de vídeo dos programadores (CRT).

As funções aplicadas aos processadores de palavra (byte processor) são baseadas na mesma filosofia, porém as operações são de uma gama mais variada.

O Software pode apresentar-se de forma linear, onde o programa é varrido desde a primeira instrução até a última não importando-se com a necessidade ou não de ser executada parte do programa.

Essa programação linear é característica dos processadores mais simples (Bit Processor).

Outra forma de programação é a programação estruturada onde um programa principal é lido e, conforme a seqüência dos eventos, os blocos de programa e funções são executados.

A programação estruturada permite a otimização do Software adaptando assim as necessidades de cada comando, oferecendo ainda a possibilidade de utilização de subrotinas e subprogramas.

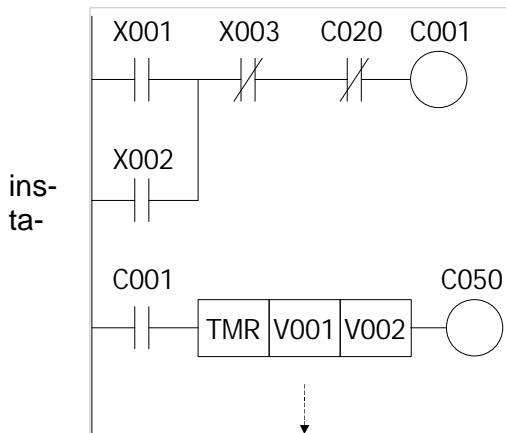
Alguns Softwares de programação permitem migrar de uma linguagem para outra. Como por exemplo, de Ladder para lista de instrução, de Ladder para diagrama lógico e vice versa.

Listas de instruções

| | |
|-----------------|--------------------|
| gua- (TP-02) | |
| | 0001 STR X001 |
| | 0002 OR X002 |
| | 0003 AND NOT X003 |
| | 0004 AND NOT C020 |
| | 0005 OUT C001 |
| | 0006 STR C001 |
| | 0007 TMR V001 V002 |
| | 0008 OUT C050 |
| | |
| | |

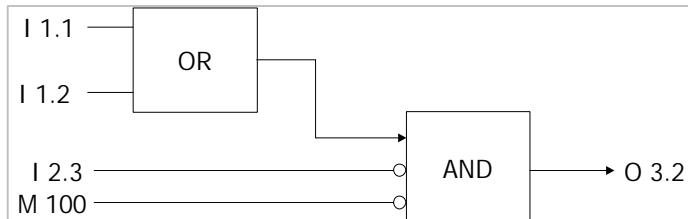
A lista ao lado mostra um exemplo da linguagem escrita na forma de mnemônicos booleanos - Weg).

Diagrama de contatos



Ao lado apresentamos um exemplo de instruções escritas na forma de diagrama de contatos ou Ladder (TP-02 – Weg).

Diagrama de blocos



Ao lado temos um exemplo de instruções escritas em diagrama de blocos ou blocos funcionais.

Sistemas Associados

Atualmente os controladores programáveis trabalham isoladamente, exceto em aplicações muito pequenas, e de maneira geral eles compõem com outros equipamentos um sistema integrado de controle. A seguir abordaremos algumas questões interessantes com relação a este aspecto.

1.1.11 REDES DE COMUNICAÇÃO

A utilização de processamento distribuído e de redes vem apresentando um crescimento significativo nos últimos anos. O advento dos chamados “sistemas distribuídos” tem exigido o desenvolvimento de novos modelos de programação e também de ferramentas apropriadas para o compartilhamento de recursos e de informações. Nos últimos anos vem-se verificando uma forte tendência de substituir sistemas computacionais centralizados, geralmente baseados em equipamentos de grande porte, por sistemas distribuídos, compostos por diversos similares de menor porte.

Para contribuir com essa tendência salienta-se que os processadores se tornaram muito mais baratos nos últimos anos, a modularidade obtida conduz a sistemas de mais fácil instalação, manutenção e expansão, além de permitir que o processo continue operando mesmo que um dos integrantes apresente uma falha. Esse controle distribuído, no entanto, somente é viável se todos os integrantes do sistema puderem trocar informações entre si de modo rápido e confiável. Dessa necessidade surgiu um campo vastíssimo de tecnologia em redes de comunicação. Diversos são os tipos, padrões, protocolos e centros de pesquisa e desenvolvimento em torno desse assunto.

Rede de comunicação é o conjunto de equipamentos e software utilizados para propiciar o trânsito de informações entre os diversos níveis hierárquicos e participantes de um processo industrial é chamado de rede de comunicação para automação. Atualmente, uma das características importantes de uma rede de comunicação é que ela seja aberta.

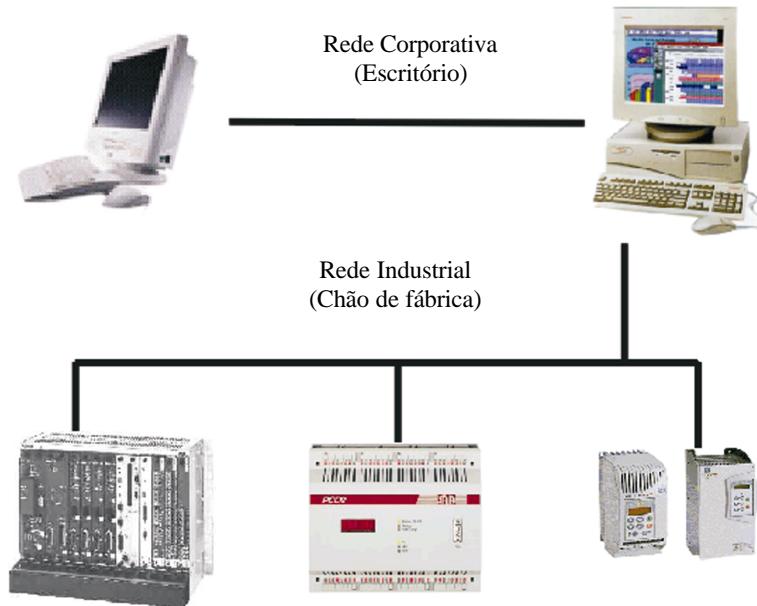


Figura 9- Trânsito de informações através de uma rede de comunicação

Vantagens do uso de redes abertas

- Flexibilidade para estender a rede e conectar diferentes módulos na mesma linha
- Cobertura de distâncias muito maiores que as conexões tradicionais
- Redução substancial da quantidade de cabos
- Ampliação do domínio da aplicação
- Redução global de custos
- Simplificação da instalação e operação
- Redução do custo de engenharia (acumulação do *know-how* devido à experiência)
- Disponibilidade de ferramentas para instalação e diagnose
- Possibilidade de conectar produtos de diferentes fabricantes

Requisitos para uma rede industrial

- Determinismo
- Flexibilidade
- Interoperabilidade
- Custo efetivo baixo
- Confiabilidade e segurança

- Facilidade de uso
- Solução completa para automação
- Ser aberto
- Padronizado
- Veloz

Grupos interessados em redes

Existem três categorias de profissionais particularmente interessados no uso das redes:

a) **Consumidores**: todos os operadores e administradores que no dia-a-dia utilizam o computador para fazer *setups* de sistemas, *download* de receitas e controle de processos, bem como acompanhamento de resultados;

b) **Integradores**: empresas que utilizam as ferramentas de *hardware* e *software* disponíveis no mercado para implementar soluções de automação usando redes de comunicação;

c) **Fornecedores**: empresas que desenvolvem equipamentos de visualização de dados, controle de variáveis, aquisição de dados e medição de grandezas para conexão direta em redes de comunicação. A Weg enquadra-se nos três grupos, pois é *consumidora* enquanto usa redes de comunicação nos mais diversos setores de fabricação e corporativo; é *integradora* enquanto fornece soluções completas de automação industrial baseadas em redes para seus clientes internos e externos e é *fornecedor* enquanto desenvolve equipamentos como o drive CFW-09, que podem ser diretamente ligados às modernas redes de comunicação.

Meios físicos, topologias e protocolos para redes

Como MEIO para tráfego das informações utilizam-se **cabos condutores elétricos** de diversos tipos, **fibras óticas** ou até mesmo **ondas de rádio**. A escolha do meio está associada a fatores como distância entre estações participantes, atenuação do sinal, imunidade a perturbações externas e velocidade de transferência das informações.

Tradicionalmente utilizam-se cabos condutores elétricos que apresentam bom desempenho em todos os quesitos de escolha. Observa-se no entanto uma tendência de uso da fibra ótica como meio de transporte para redes industriais, devido às suas excelentes características e à redução no seu custo devido ao crescente uso no setor de telecomunicações.

As redes de comunicação são elaboradas a partir de três tipos de topologia ou da combinação dessas.

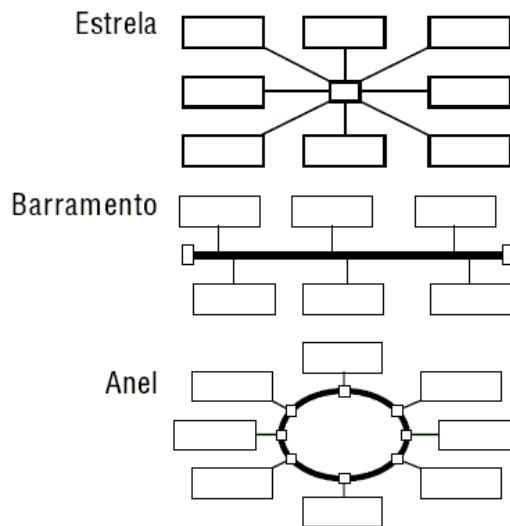


Figura 10-Tipos de topologia para interligação de equipamentos em rede

Protocolo

O protocolo de uma rede é o conjunto de regras e convenções de linguagem que são empregadas entre os participantes da rede para a troca de informações. Os protocolos, assim como as línguas, são muitos e cada um com suas particularidades. Existem no entanto alguns que se sobressaem devido à grande disseminação do seu uso e aprovação junto aos fabricantes de equipamentos, integradores e usuários.

Protocolos mais usados

Os protocolos atualmente aceitos são baseados em um documento desenvolvido pela ISO (*International Standards Organization*). Esse documento denominado OSI (*Open System Interconnection*) é um modelo de referência para o desenvolvimento de protocolos de comunicação. A estrutura do modelo OSI é baseada em 7 camadas. Softwares desenvolvidos com base nesse modelo são ditos ABERTOS, pois qualquer fabricante de equipamentos pode usá-lo para desenvolvimento de produtos que almejam ser usados em rede. Destacam-se hoje os seguintes padrões de rede que seguem o modelo OSI (Normalizado):

- PROFIBUS
- DeviceNET
- ETHERNET

Redes PROFIBUS

A PROFIBUS, de origem européia, é um padrão aberto de comunicação para um largo campo de aplicações em automação da manufatura, predial e processo. A independência de fabricante e abertura são garantidas pelo padrão PROFIBUS EN50170. Com o PROFIBUS, dispositivos de fabricantes diferentes podem se comunicar sem adaptações especiais de interface. Existem três tipos de PROFIBUS: a FMS, a DP e a PA. Dessa a DP é a mais utilizada.

PROFIBUS-FMS (*Fieldbus Message Specification*)

Solução de uso geral para tarefas de comunicação no nível célula. Serviços poderosos de FMS permitem largo campo de aplicações e proporcionam grande flexibilidade. Pode também ser usado para tarefas complexas e extensas de comunicação.

PROFIBUS-DP (*Decentralized Peripheria*)

Otimizado para alta velocidade e conexão de baixo custo. Esta versão de PROFIBUS é projetada especialmente para comunicação entre sistemas de controle de automação e I/O distribuído em nível de dispositivo.

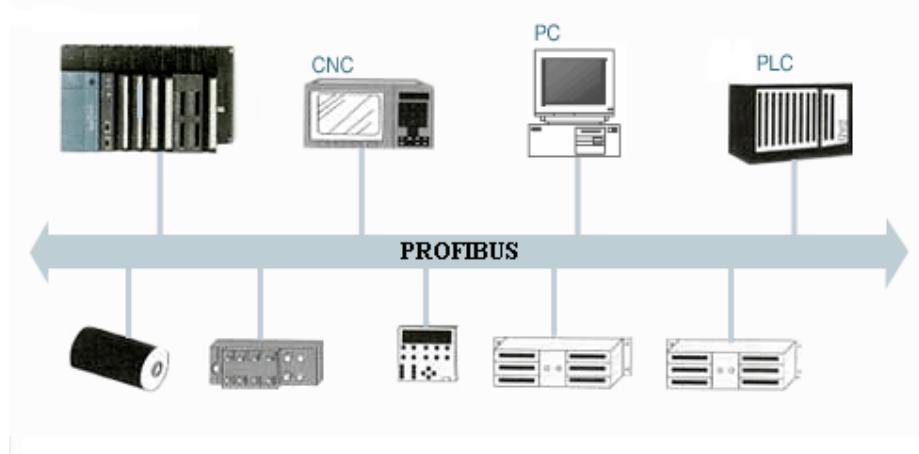


Figura 11- A rede Profibus-DP é a mais utilizada atualmente

PROFIBUS-PA (*Process Automation*)

Projetado especialmente para automação de processo (instrumentação). Permite que transmissores e atuadores sejam ligados em uma linha comum de rede regular em áreas intrinsecamente seguras. O PROFIBUS-PA permite comunicação de da-

dos e alimentação sobre a rede usando tecnologia de 2 fios de acordo com a Norma Internacional IEC 1158-2.

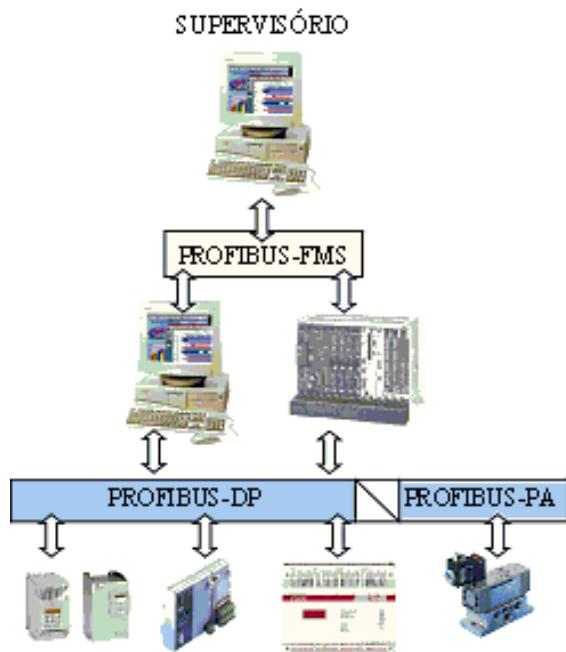


Figura 12- Estrutura típica baseada em rede PROFIBUS

Redes DeviceNET

A DeviceNET, de origem norte-americana, é um protocolo de comunicação para ligar dispositivos industriais como fins de curso, sensores fotoelétricos, manifolds, partidas de motor, sensores de processo, leitores de código de barra, drivers de frequência variável e interface de operador a uma única rede.

O DeviceNET é baseado num protocolo de comunicação chamado CAN. O CAN foi originalmente desenvolvido pela Bosch para o mercado de automóvel europeu para substituir os caros chicotes de cabo por um cabo em rede de baixo custo interligando componentes inteligentes como o computador de bordo, freios ABS, alarmes etc.

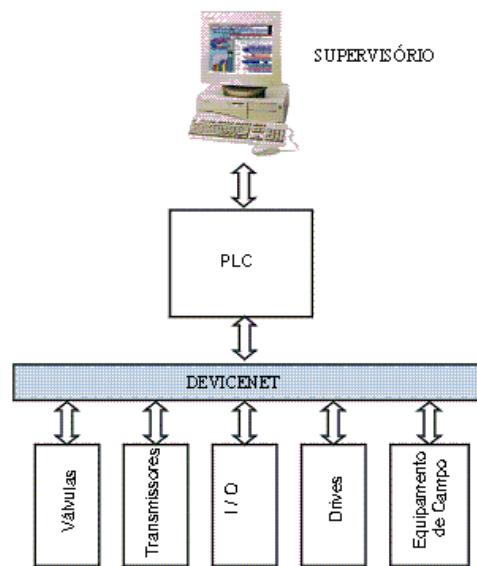


Figura 13 - Estrutura típica baseada em rede DeviceNET

Redes Ethernet

O padrão Ethernet é um dos mais populares e difundidos nas redes corporativas (escritórios) instaladas e certamente é o mais empregado em novos projetos. Sua popularidade deve-se a difusão em larga escala dos micro-computadores.

Ao contrário da PROFIBUS e DeviceNET, a ETHERNET não é determinística e ocorrem colisões de dados na rede. Isso do ponto de vista da automação é não recomendável pois pode comprometer o desempenho do sistema que está sendo controlado. Como o tempo não é tão crítico nas transações de informações no nível corporativo e de escritório, uma vez detectada uma colisão as mensagens são retransmitidas obedecendo uma lógica própria da ETHERNET. Na PROFIBUS e Device-NET as colisões não ocorrem pois o controlador de acesso ao meio entrega um *token* (ficha) ao integrante da rede que está na vez de transmitir. Existe um escalonamento pré-definido que torna o sistema determinístico.

O grande interesse das empresas e dos consumidores em geral tem levado a desenvolvimentos e aprimoramentos da ETHERNET que recentemente elevou a taxa de transmissão máxima de 10 para 100Mbits/s na chamada *Fast Ethernet*. Mas o trabalho não para por aí. Foi iniciado mais um grupo de trabalho com a denominação IEEE 802.3z e IEEE 802.3ab cujo objetivo é apresentar as especificações de mais um tipo de rede ETHERNET sinalizando em 1000Mbits/s, chamada de *Gigabit Ethernet*. O desenvolvimento do novo padrão começa a demonstrar que a era dos cabos elétricos pode estar chegando ao fim. As novas especificações já serão baseadas na tecnologia de *fibra ótica*.

Outra grande novidade é que o problema da falta de determinismo na rede Ethernet também vem sendo tratado e soluções baseadas na utilização de equipamentos especiais, chamados SMART HUB, estão começando a se tornar realidade. Os sistemas de automação de médio e grande porte que não sejam baseados em rede de comunicação já estão ultrapassados. Os modernos equipamentos como PCs, Drives (Soft-Starter, Inversores, Servo-drives), CCM inteligentes (CCM-03i) e CLPs já são concebidos voltados para este tipo de aplicação. A arquitetura utilizada nos projetos modernos de automação está buscando um único padrão ou um número menor de tipos de redes para interligar seus participantes. A arquitetura utilizada pela Weg em seus mais recentes projetos segue também essa tendência usando apenas dois níveis de rede.

O trabalho dos comitês normatizadores buscando a padronização dos protocolos, o aumento do uso e barateamento da fibra ótica como meio físico bem como o aumento da velocidade de transmissão irão contribuir significativamente para o uso generalizado das redes de comunicação para automação industrial.

| COMPARAÇÃO DE DESEMPENHO | | | |
|---|--------------------------------------|--------------------------------------|--------------------------------------|
| | ProfiBus | DeviceNet | Ethernet |
| Topologia | Barramento ou linha, (Estrela, Anel) | Barramento ou linha, (Estrela, Anel) | Barramento ou linha, (Estrela, Anel) |
| Máximo n.º componentes | 127 | 64 | 1024 |
| Método de Acesso ao meio | Token passing | Carrier-Sonac Multiple Access | CSMA/CD |
| Velocidade de Transmissão | Até 12Mbps | Até 500Kbps | Até 10Mbps |
| Tempo ciclo: 16 nós com 16 I/O digitais | T < 2ms | 2ms | Não informado |
| Tempo ciclo: 16 nós com 8 I/Os analóg. | T < 2ms | 10ms | Não informado |

CSMA/CD: Carrier Sense Multiple Access / Collision Detection

1.1.12 SUPERVISÃO E CONTROLE

Um sistema de controle de um máquina, conjunto de máquinas ou processo, pode configurar-se de diversas formas: individualmente, centralizado ou distribuído.

A definição e adoção da forma mais adequada, vai depender, também, de uma avaliação dos diversos aspectos envolvidos, tais como, complexidade do sistema, flexibilidade desejada, nível de redundância, integração, manutenibilidade, custo, etc. O

que vale a pena destacar, é que o controlador programável, independentemente da configuração adotada, aparece como uma excelente opção, como equipamento de controle. Os recursos de software para funções de sequenciamento e intertravamento, controle de malha aberta e fechada, bem como, a disponibilidade de distribuição do controle, através das redes de comunicação e dos seus módulos especiais inteligentes, permitindo, desta forma, a implementação parcial ou total da redundância no sistema, confirmam nossa afirmação.

Os sistemas modernos de automação industrial estão sendo baseados em arquiteturas verticalmente distribuídas, conforme a figura 3.

A implementação do nível de supervisão do controle do processo, ou seja, da interface homem-processo, assume, também, papel muito importante dentro desta estrutura hierárquica de controle. Evidentemente, existem várias maneiras de implementação, e a utilização de CP's, no nível de controle, possibilita tais opções. A utilização dos tradicionais painéis sinópticos de controle, em função da ocupação em demasia, muitas vezes, do espaço disponível, bem como, pela pouca flexibilidade para possíveis alterações, tem sido preterida.

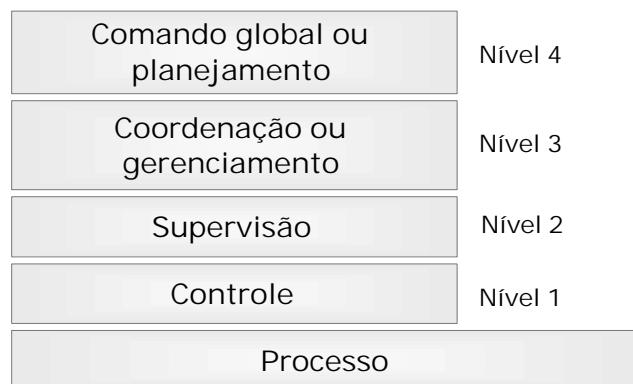


Figura 14 - Arquitetura de um sistema de automação

A opção que vem se apresentando como bastante atraente, sendo cada vez mais empregada, tanto no mercado internacional como brasileiro, é a utilização dos microcomputadores e seus periféricos, como interface homem-processo.

O êxito da utilização deste equipamento em conjunto com CP's, é decorrente da sua série de vantagens proporcionadas:

- ambiente de hardware e software próprio para o desenvolvimento de programas dedicados às funções de supervisão, tendo em vista, o bom suporte de software, especificamente linguagens de programação de alto nível;
- capacidade de memória e velocidade de processamento adequadas à maioria das aplicações;

- modularidade e portabilidade dos programas;
- facilidade de acréscimo de novas funções e de manutenção das já existentes;
- linguagem gráfica de fácil manuseio para construção de telas de sinópticos de processos, etc.;
- custo relativamente baixo.

A tendência verificada é a da utilização de microcomputadores compatíveis com o IBM-PC. A forma construtiva destes microcomputadores depende basicamente do local onde será instalado. Podendo ser um micro industrial de mesa, uma workstation com monitor e teclado incorporado, ou uma placa que pode ser conectada no próprio Rack do CP.

As principais funções implementadas pelo microcomputador são as seguintes:

- Apresentação de sinópticos do processo, com atualização dinâmica dos valores reais e teóricos das variáveis controladas;
- Apresentação de frontais de instrumentos, com informações relativas a cada malha, tais como, limites de alarme, ponto de ajuste (set-point), parâmetros de controle (ganhos), etc.;
- Registro de tendência (representação gráfica x tempo), em tempo real, das variáveis controladas;
- Registro de tendência histórica, através da armazenagem das informações anteriores, com apresentação sob solicitação ou freqüência pré-determinada;
- Registros de alarmes (ocorrências, conhecimento e retorno ao normal), e eventos (troca de estado das malhas, alteração de set-points, limites de alarmes, etc), com indicação da data, hora e descrição do evento ou alarme;
- Hard-copy das telas em impressoras;
- Manutenção de biblioteca de procedimentos padrão, para ser consultada pelo operador em caso de tomadas de decisão;
- entre outras.

A adoção de dois microcomputadores acoplados à rede de comunicação, com subdivisão de atribuições, bem como, a possibilidade de operação backup de cada um deles, ou seja, o controle integral de um no caso de uma falha do outro, é uma prática largamente empregada.

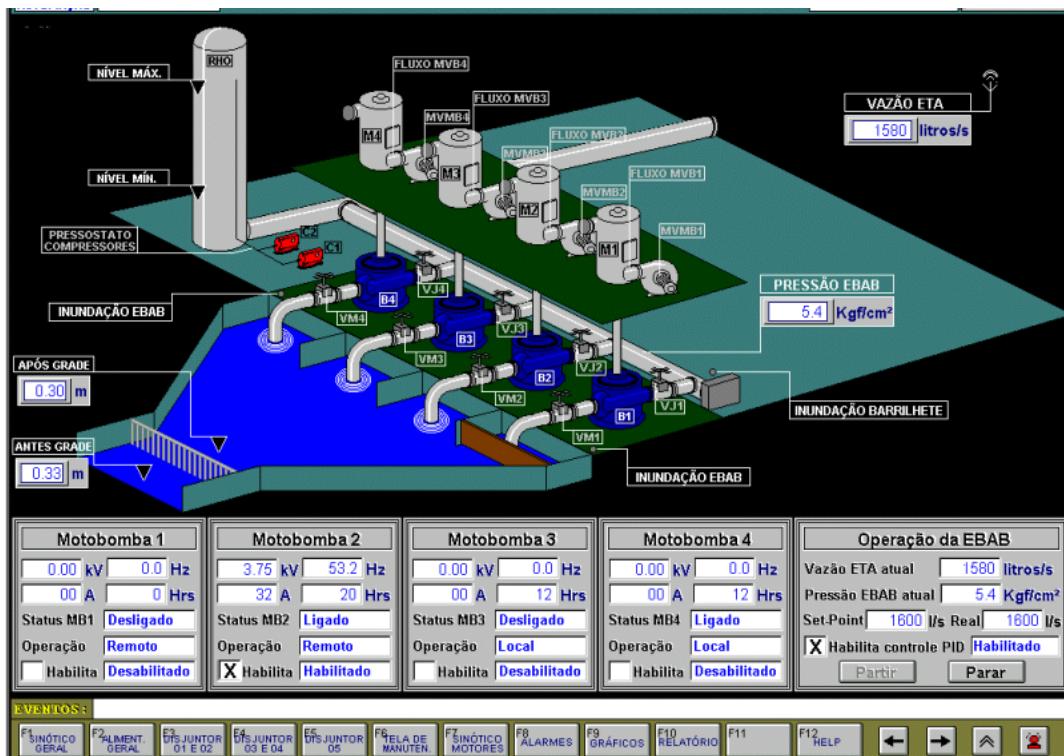


Figura 15 - Exemplo de tela utilizada em um sistema de supervisão

ANEXOS

1.1.13 - **MANUAL DO TPW3-PCLINK (SOFTWARE DE PROGRAMAÇÃO DO TPW-03)**

Requisitos de Sistema

O TPW3-PCLINK necessita dos seguintes requisitos básicos para funcionar corretamente :

- Um computador pessoal compatível com o IBM -PC com processador Pentium 133 MHz ou superior;
- Pelo menos 10 Mbytes de espaço livre de disco rígido;
- No mínimo 128 Mbytes de memória RAM
- Microsoft Windows 98 ou superior;

Nota : É aconselhável que se feche todos os outros programas/aplicativos abertos.

Instalação

Para instalar o programa basta seguir os seguintes passos :

- Ative seu browser e vá até o endereço www.weg.net
- Clique sobre o link [downloads e sistemas online](#);
- Na lista de softwares ou sistemas weg, escolha : [software de programação do controlador TPW-03](#)
- Em seguida, clique sobre o ícone: Fazer o Download;
- Aguarde o fim do processo e instale o aplicativo, executando o arquivo de SETUP;
- Siga as orientações do instalador até a conclusão da instalação.

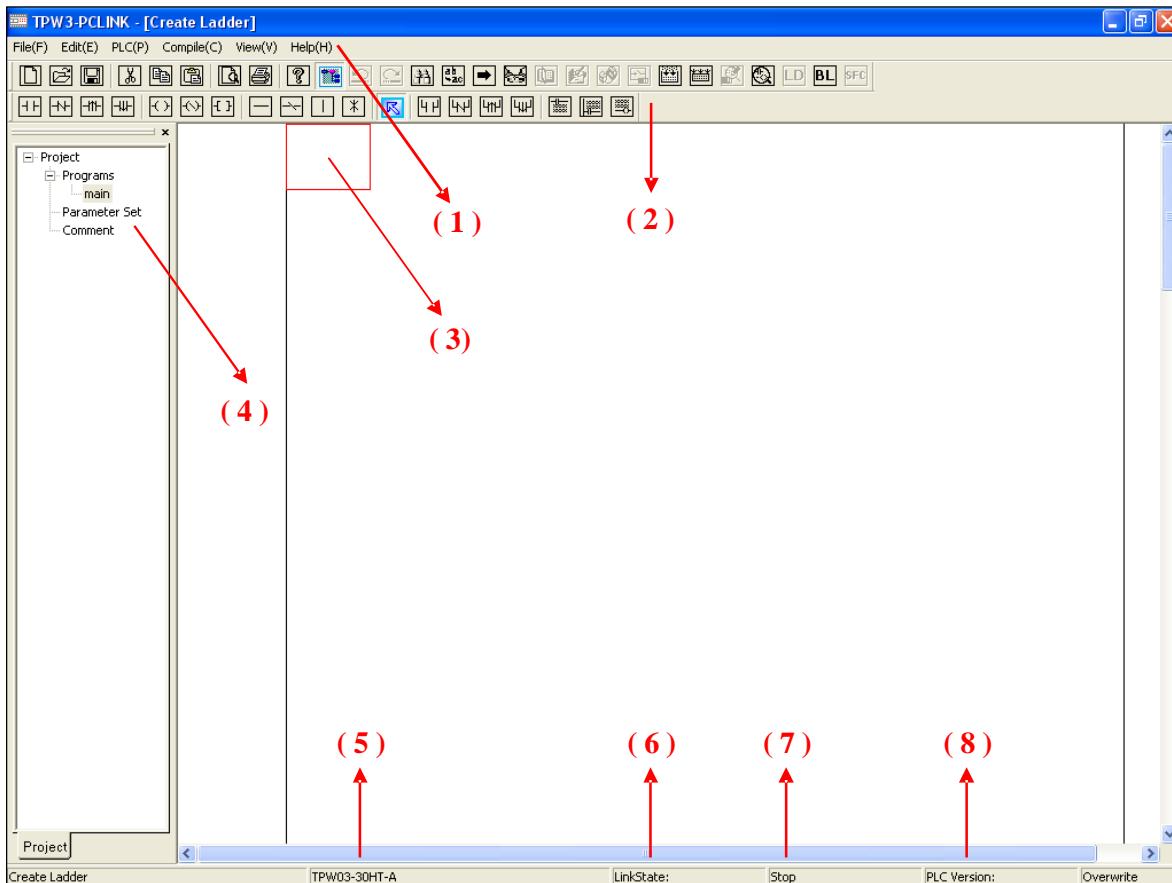
Nota : Caso ocorra algum problema ou falha durante a instalação do programa, entre em contato com a WEG Automação S/A (Fone: 0800 701 0701).

Visão Geral

O TPW3-PCLINK permite ao usuário criar o software aplicativo para toda linha TPW-03 de controladores programáveis. A seguir faremos uma descrição das principais características (telas, menus, comandos básicos, arquitetura de memória, etc.), bem como aplicaremos alguns exercícios para fixação dos conceitos e comandos.

Tela Principal

A figura abaixo mostra a tela principal do TPW3-PCLINK. Os detalhes referentes a cada uma das partes da interface é feita em seguida.



- (1) – Menu Principal
- (2) – Barra de Botões
- (3) – Cursor na área de edição do programa
- (4) – Opções do programa
- (5) – Indicação do módulo básico utilizado
- (6) – Porta de comunicação serial selecionada
- (7) – Status de operação do TPW -03
- (8) – Versão do hardware do TPW -03.

Opções do Menu Principal :

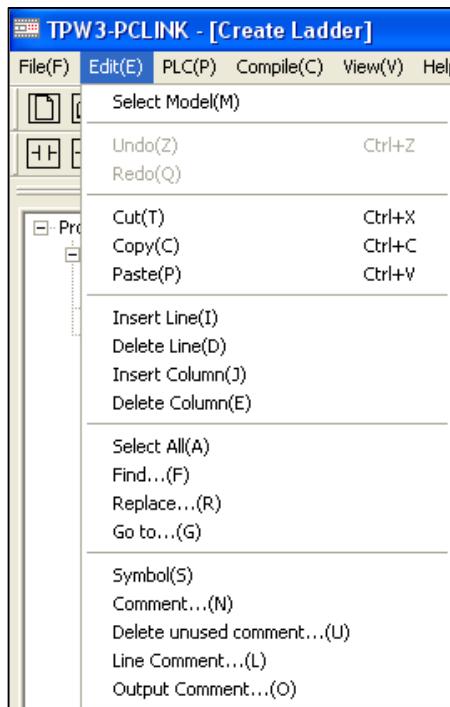
File Menu (Menu Arquivo)



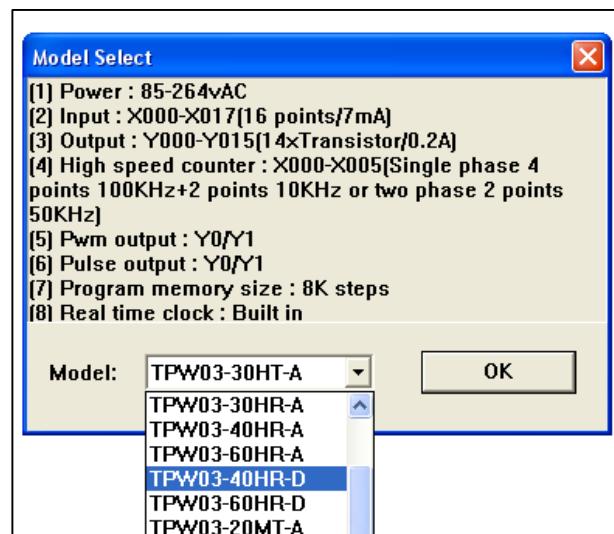
- **New (Novo)** : Cria um novo arquivo de programa, seleciona o modelo de TPW -03 e a linguagem de programação a ser utilizada;
- **Open (Abrir)** : Para abrir um arquivo de programa previamente criado, com extensão “*.tpc”;
- **Close (Fechar)** : Fecha a edição do programa atual, lembrando o usuário a salvar o programa. Esta opção não é habilitada quando o programa estiver em modo monitoração;
- **Save (Salvar)** : Para salvar o programa que está sendo editado;
- **Save As (Salvar Como)** : Para salvar o programa ativo com um outro nome e o local no computador;
- **Project Information (Informações do projeto)** : Podem ser inseridas informações relativas ao projeto, como o autor e um comentário;
- **Print (Imprimir)** : Para imprimir o programa aplicativo ou seus parâmetros;
- **Print Preview (Visualizar Impressão)** : Permite uma visualização antecipada do programa aplicativo ou seus parâmetros;
- **Print Setup (Ajuste de Impressão)** : Para selecionar e configurar dados, margens, cabeçalhos e a impressora;
- **Recent File List (Lista de arquivos recentes)** : Os 4 programas mais recentes são exibidos;
- **Exit (Sair)** : Para encerrar a edição e fechar o software de programação;

OBS: Alguns destes comandos são encontrados também na Barra de Botões, que permitem um acesso mais rápido e direto ao comando desejado.

Edit Menu (Menu Editar)

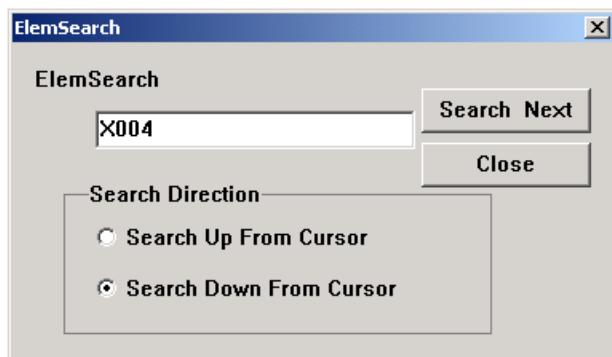


- **Select Model** (Tipo de PLC) : Define o tipo de unidade básica para o qual será criado o aplicativo;

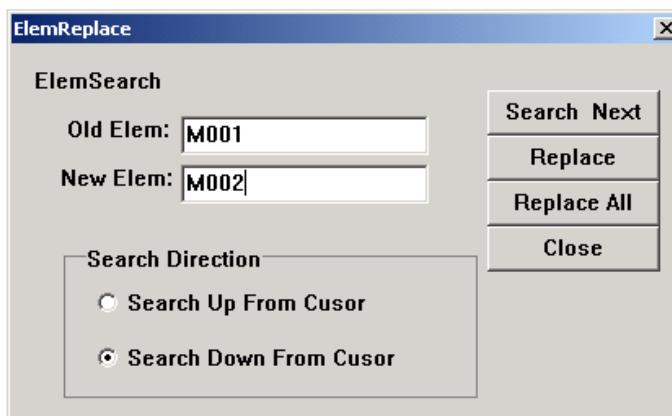


| | | | |
|---|----------------------------------|-----------------------|-----------------|
|  | Capacidade de memória disponível | TPW-03 - 20/30 pontos | 8 K instruções |
| | | TPW-03 – 40/60 pontos | 16 K instruções |

- **Undo** (Desfazer) : Para desfazer a última ação realizada no programa.
- **Redo** (Refazer) : Para recuperar a operação do comando “UNDO”.
- **Cut** (Recortar) : Recortar os componentes, as linhas ou as etapas selecionadas.
- **Copy** (Copiar) : Copiar os componentes, as linhas ou as etapas selecionadas
- **Paste** (Colar) : O conteúdo copiado (Copy) ou recortado (Cut), será colado na área selecionada
- **Insert Line** (Insere linha) : Introduzir uma linha em branco na área selecionada
- **Delete Line** (Apaga linha) : Apagar a linha na área selecionada
- **Insert Column** (Insere coluna) : Introduzir uma coluna na área selecionada
- **Delete Column** (Apaga coluna) : Apagar a coluna no local selecionado
- **Select All** (Selecionar tudo) : Selecionar todo o programa atual
- **Find...** (Procurar) : Encontra o componente de acordo com o endereço especificado

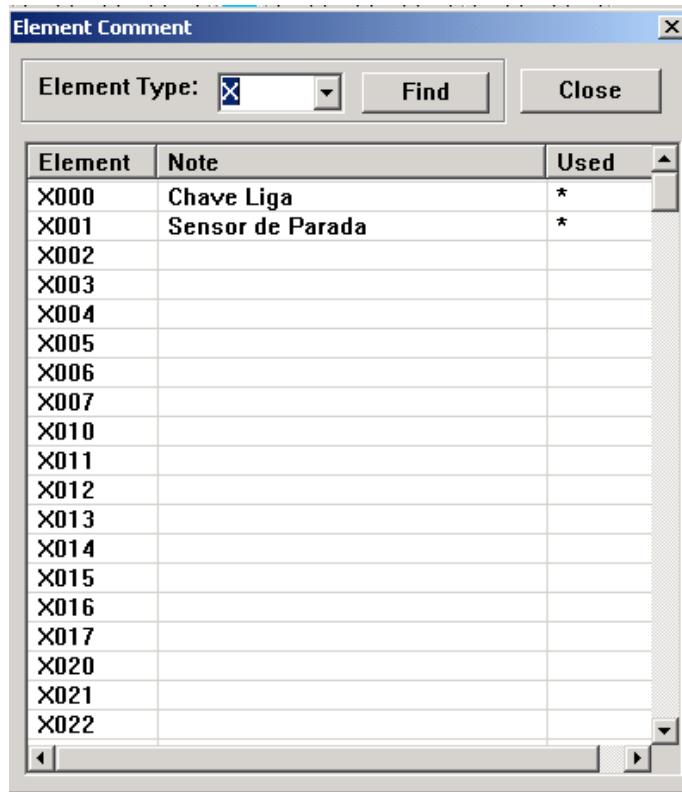


- **Replace...** (Substitui) : Substitui o endereço da instrução indicada por outro a ser especificado.



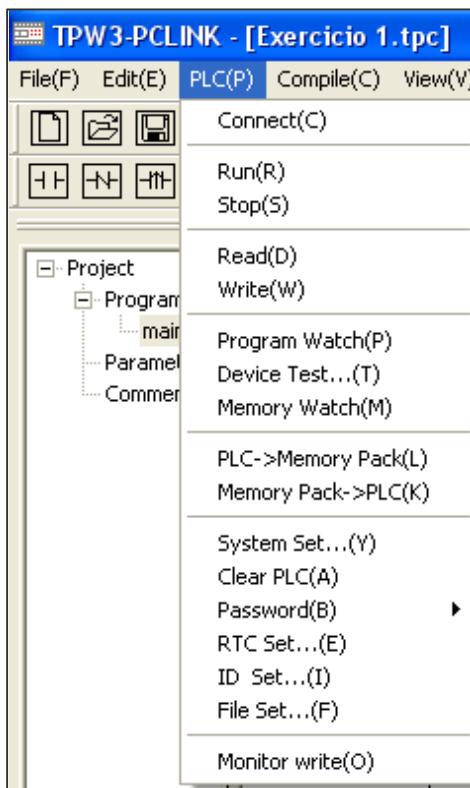
- **Go To...** (Ir para) : Leva o cursor para uma linha desejada no programa;

- **Symbol** (Símbolos) : Para atribuir comentários (Tag's) às instruções do programa (entradas, saídas, marcadore s, contadores, etc..). Também indica os endereços que já foram utilizados no programa.



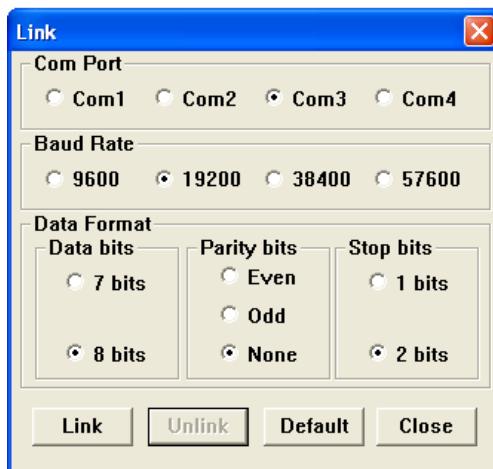
- **Comment...** (Comentário) : Mostra ou edita comentários dos endereços;
- **Delete unused comment...** (Apaga comentários não usados) : Apaga os comentários que não foram usados no programa;
- **Line Comment...** (Comentário de linha) : Mostra ou edita o início da linha de comentários, comportando em cada linha até 64 caracteres;
- **Output Comment...** (Comentário de saída) : Mostra ou edita uma linha de comentários para as saídas;

PLC Menu (Menu PLC)



Alguns comandos deste menu somente poderão ser utilizados quando o CLP estiver com uma conexão com o microcomputador (Comando “Connect” executado).

- **Connect (Conexão)** : Conecta (LINK) ou desconecta (UNLINK) o microcomputador PC ao PLC TPW -03 (LINK). Fazer a conexão significa estabelecer a comunicação serial entre os dois equipamentos. A porta de comunicação, taxa de transferência (baud rate) e o formato de dados podem ser selecionados conforme a configuração do PC;



- **Run** (Funcionar) : Coloca a CPU do CLP em modo execução. Inicia o ciclo de varredura (scan) do controlador;
- **Stop** (Para) : Paralisa a execução da CPU do CLP;



- Após este comando o led indicativo RUN ficará continuamente aceso;
- Caso ocorra uma indicação de erro verifique a memória de sistema do controlador para identificar o que está ocorrendo;

- **Stop** (Para) : Paralisa a execução da CPU do CLP;



Após este comando o led indicativo RUN começará a piscar avisando que o controlador não está executando a varredura.

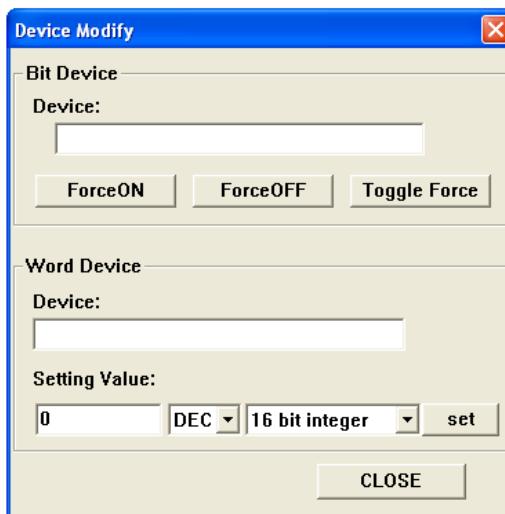
- **Read** (Ler) : Copia o programa do CLP para o PC;
- **Write** (Escrever) : Copia o programa do PC para o CLP;
- **Program Watch** (Monitorar Programa) : Monitora através do PC o status do programa do CLP, indicando o estado das instruções do programa (entradas, saídas, marcadores, etc) e o conteúdo de registradores;
- **Device Test...** (Teste de dispositivo) : habilita a possibilidade de modificar bit's ou word's:

Bit Device

Modifica (força) o status dos contatos/bobinas para ON ou OFF. Podem ser modificados os seguintes endereços: entradas (X), saídas (Y), contato auxiliar (M).

Word Device

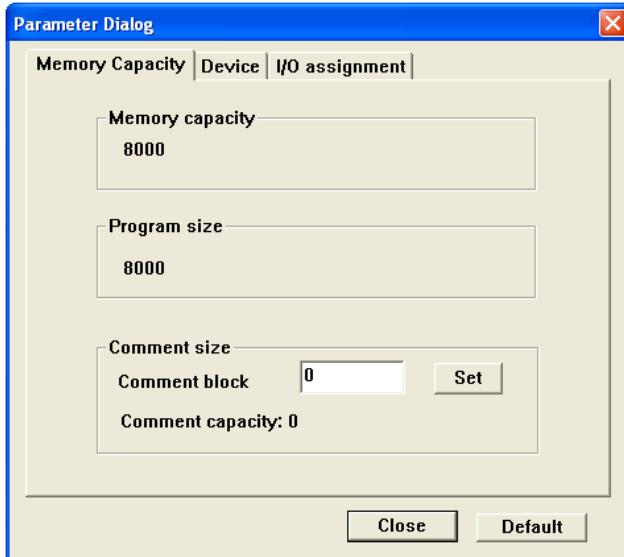
Os valores dos registradores podem ser alterados, conforme o valor ajustado nesta função.



- **Memory Watch** (Monitorar Memória) : Mostra a tabela com os valores reais dos registradores selecionados;
- **PLC -> Memory Pack** : Transfere o programa do CLP para a memória de Backup;
- **Memory Pack -> PLC** : Transfere o programa da memória de Backup para o CLP;
- **System Set** (Ajuste da Memória) : Podem ser ajustados dados da memória, conforme abaixo:

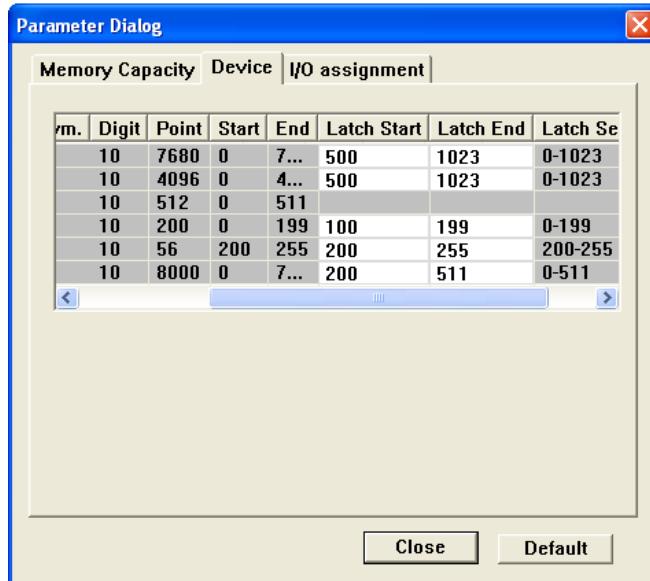
Memory Capacity

Nesta janela é possível ajustar uma faixa de memória do CLP para gravar os comentários das instruções do programa. Cada bloco de comentário consome 10 bytes da memória. Entende-se como bloco de comentário, cada comentário(symbol) feito a uma instrução. Os comentários de linha não são gravados.



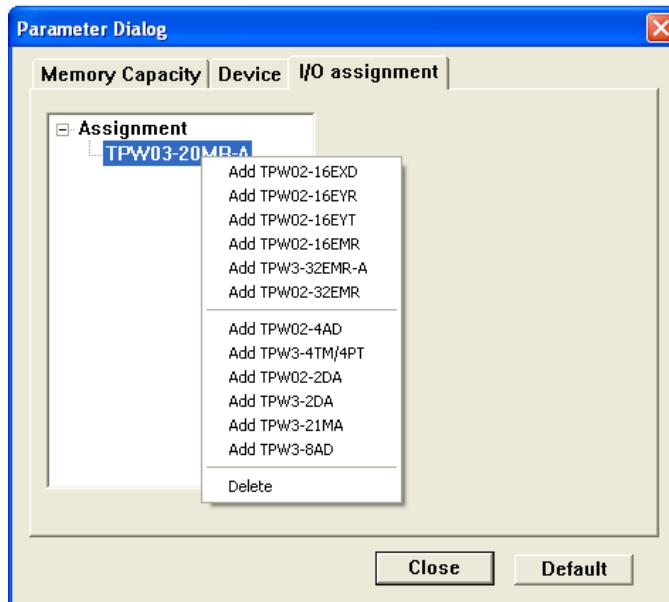
Device

Nesta janela é possível ajustar a memória retentiva do CLP. Os endereços que podem ser ajustados são: M, S, T, C e D.



I/O assignment

A janela do “I/O assignment”, é usada para adicionar os módulos de expansão. Com um clique no botão direito do mouse sobre o modelo da CPU utilizada, é possível selecionar os módulos a serem adicionados na configuração de uma determinada aplicação, já permitindo a visualização de como serão os endereços dos pontos de entrada/saída destes módulos de expansão.

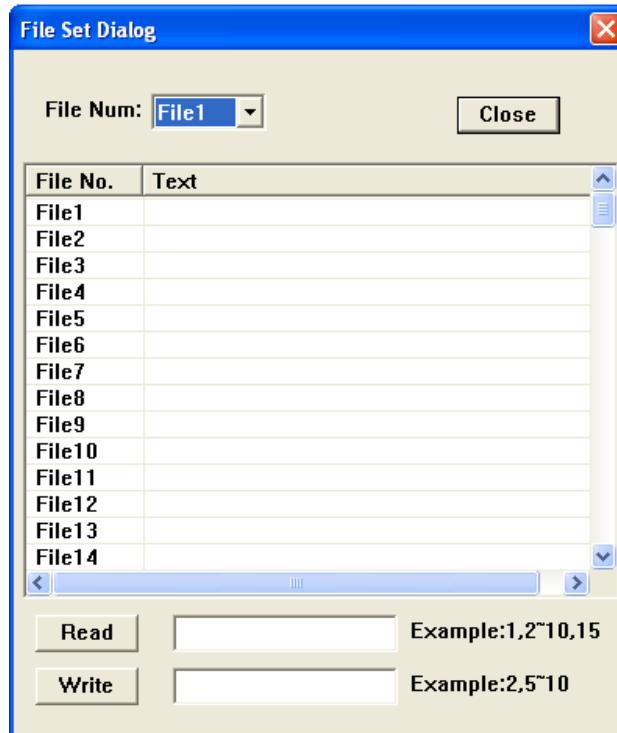


Ao inserir algum módulo de expansão, não esquecer de selecionar a opção “Param eter Set”, quando da escrita do programa para o CLP (comando “Write”)

- **Clear PLC** (Apagar PLC) : Limpa toda a memória do CLP(programas e ajustes feitos);
- **Password -> Register...** (Registro de Senha) : Insere a senha com possibilidade de selecionar o nível de proteção;
- **Password -> Disable...** (Desabilitar Senha) : Desabilita a senha inserida;
- **Password -> Delete...** (Apagar Senha) : Apaga a senha inserida;
- **RTC Set...** (Ajuste do RTC) : Ajuste do Relógio de Tempo Real do CLP, com opções de data e hora;

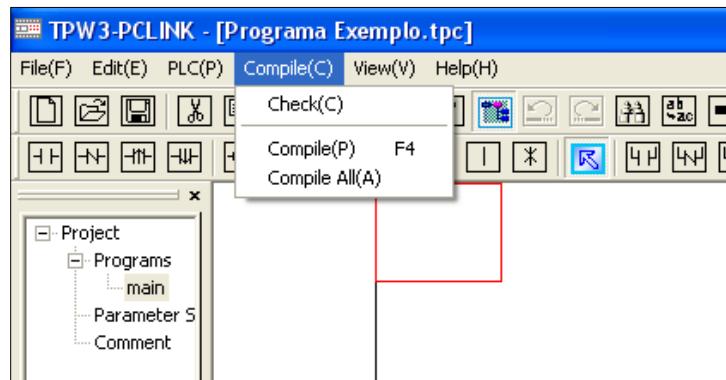


- **ID Set...** (Ajuste ID) : Altera o número (endereço) do CLP em uma rede;
- **File Set...** (Ajuste de arquivo) : Abre a janela para digitação das linhas a serem apresentadas no display da IHM (OP08);



- **Monitor Write** : Escreve o programa para o CLP com o CLP em modo RUN. Se esta função for habilitada o CLP será programado automaticamente no instante em que o programa é compilado (comando “Compile”).

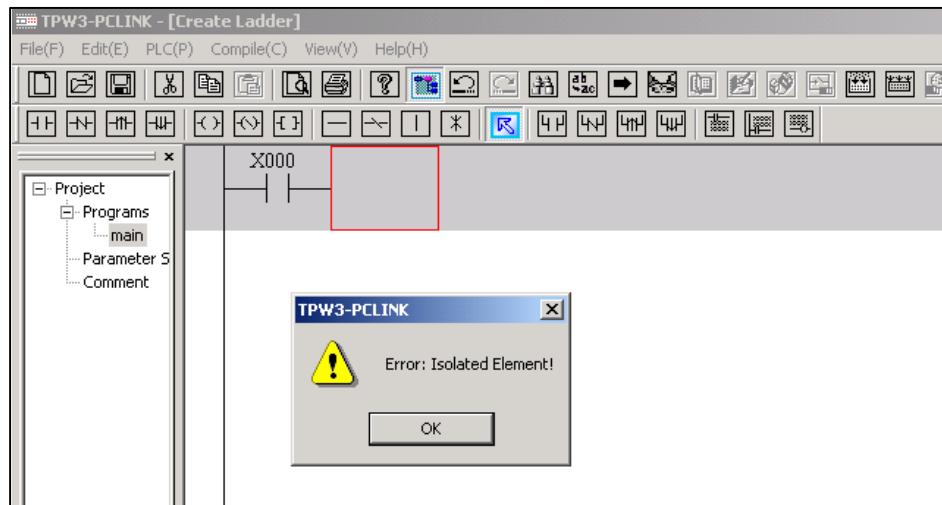
Compile Menu (Menu Compilar)



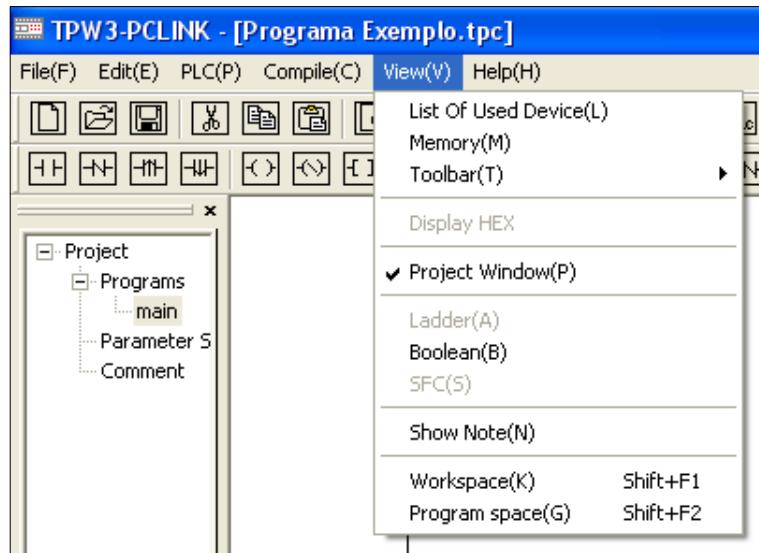
- **Check (Verificar)** : Verifica erros em todo o programa;
- **Compile (Compilar)** : Compila o programa atual;
- **Compile All (Compilar tudo)** : Compila todos os programas;



Quando o programa é compilado, ele é convertido para linguagem de máquina, que será utilizada na memória do CLP. Também quando executa-se o comando “compile” o software procura eventuais erros no aplicativo, alertando quando existirem e já indicando com o cursor o local onde encontra-se erro. Ver exemplo abaixo, onde existe um contato sem bobina na saída.



View Menu (Menu Exibir)



- **List of Used Device** (Lista dos componentes usados) : Apresenta uma lista com todos os endereços e seus respectivos comentários. Se o endereço estiver sendo utilizado no programa, este será marcado com um asterisco;

Element Comment

| Element Type: | Find | Close |
|---------------|------|-------|
| Element | Note | Used |
| X000 | LIGA | * |
| X001 | | |
| X002 | | |
| X003 | | |
| X004 | | |
| X005 | | |
| X006 | | |
| X007 | | |
| X010 | | |
| X011 | | |
| X012 | | |
| X013 | | |
| X014 | | |
| X015 | | |
| X016 | | |
| X017 | | |
| X020 | | |
| X021 | | |
| X022 | | |

- **Memory** (Memória) : Mostra a lista de instruções utilizadas no programa;

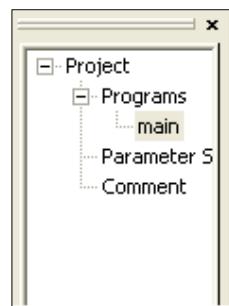
- **Toolbar à Main Toolbar** (Barra de Ferramentas principal) : Seleciona a barra de ferramenta principal;



- **Memory** (Memoria) : Mostra a lista de instruções utilizadas no programa;
- **Toolbar à SFC Edit Toolbar** (Barra de Ferramentas do Editor SFC) : Seleciona a barra de ferramentas para a programação em SFC (Linguagem de programação não habilitada);
- **Toolbar à Ladder Edit Toolbar** (Barra de Ferramentas do Editor Ladder) : Seleciona a barra de ferramentas para a programação em Ladder;

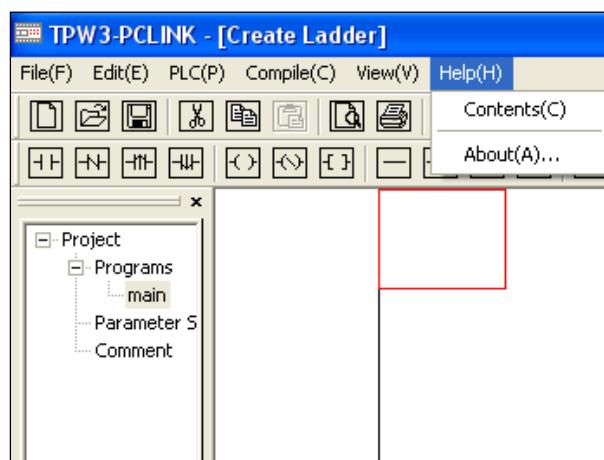


- **Project Window** (Janela do projeto): Se selecionado mostra a janela onde contém os dados do projeto;



- **Ladder** : Seleciona a linguagem de programação em Ladder;
- **Boolean** : Seleciona a linguagem de programação em Booleano (Lista de Instruções);
- **SFC** : Seleciona a linguagem de programação em SFC (não disponível);
- **Show Note** (Exibe notas): Se selecionado, mostra todos os comentários feitos no programa;

Help Menu (Menu Ajuda)



- **Contents** : ra os índices do software TPW3-PCLINK;
- **About (Sobre)**: Mostra a versão de software;

Ícones da Barra de Ferramentas



| Menu | Ícone | Atalho | Descrição |
|--------------------------|-------|--------|--|
| File(F) | | Alt+F | Seleciona o menu File |
| File à New(N) | | Crtl+N | Cria um novo programa |
| File à Open(O) | | Crtl+O | Abre um programa salvo |
| File à Save(S) | | Crtl+S | Salva a modificação no programa atual |
| Edit à Cut(T) | | Crtl+X | Recortar |
| Edit à Copy(C) | | Crtl+C | Copiar |
| Edit à Paste(P) | | Crtl+V | Colar |
| File à Print preview (V) | | | Visualiza o programa a ser impresso |
| File à Print(P) | | Crtl+P | Imprime o programa atual |
| Help à About(A) | | | Visualiza a versão do software |
| View à Project Window(P) | | | Mostra a janela com os dados do programa |
| Edit à Undo(Z) | | Crtl+Z | Desfaz a ultima ação feita no programa |
| Edit à Redo(O) | | | Recupera a operação do UNDO |
| Edit à Find(F) | | | Encontra o componente a partir do endereço |
| Edit à Replace(R) | | | Substitui o endereço da função |
| Edit à Go to(G) | | | Encontra a linha desejada no programa |
| PLC à Connect(C) | | | Conecta o PLC ao PC |
| PLC à Read(D) | | | Copia o programa do CLP para o PC |
| PLC à Write(W) | | | Escreve o programa do PC para o CLP |

| | | | |
|--------------------------|--|----|---|
| PLC à Program Watch(P) | | | Monitora o status do programa no CLP |
| PLC à Device Test(T) | | | Modifica o status dos contatos/bobinas |
| Compile à Compile(P) | | F4 | Compila o programa atual |
| Compile à Compile All(A) | | | Compila todos os programas do projeto |
| PLC à Monitor write(O) | | | Escreve o programa para o CLP com o CLP em modo RUN |
| Compile à Check(C) | | | Verifica erros em todo programa |

Ícones da Barra de Ferramentas Ladder



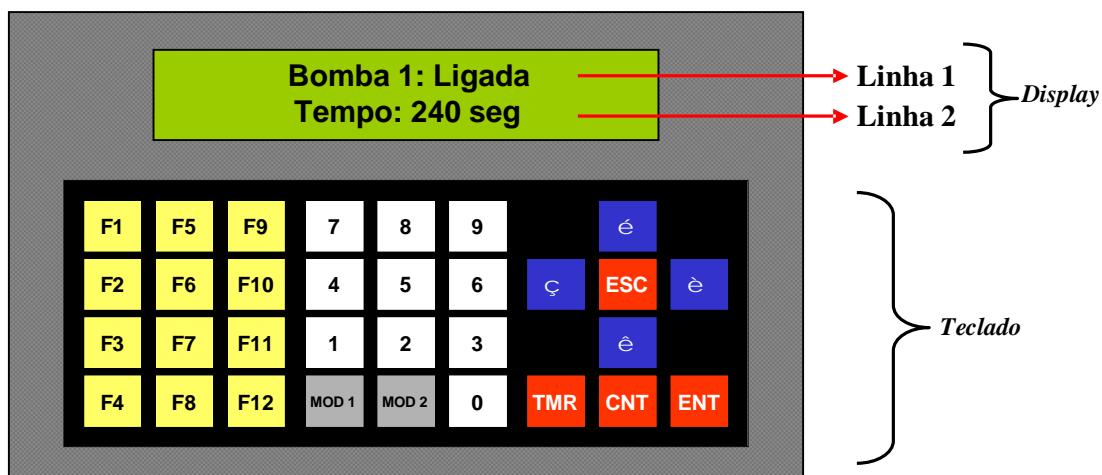
| Ícone | Descrição | Ícone | Descrição |
|-------|--------------------------------------|-------|------------------------------------|
| | Contato aberto | | Deleta vertical |
| | Contato fechado | | Seleciona a seta |
| | Pulso de subida | | Contato aberto abre interligação |
| | Pulso de descida | | Contato fechado abre interligação |
| | Saída | | Pulso de subida abre interligação |
| | Saída fechada | | Pulso de descida abre interligação |
| | Instrução de aplicação | | Mostrar nota no contato |
| | Linha horizontal (F5) | | Mostrar nota na linha |
| | Reverte o resultado da operação (F6) | | Mostrar nota na saída |
| | Linha vertical (F7) | | |

1.1.14 PROGRAMAÇÃO DA INTERFACE HOMEM-MÁQUINA OP-07/OP-08

As interfaces homem-máquina OP-07 e OP-08 são periféricos que permitem ao usuário efetuar alteração/visualização de valores de registradores, apresentação de textos e mensagens, comandos simples, em função do modo de operação definido.

Para que o TPW-03 reconheça a interface homem-máquina não se faz necessário nenhum tipo de ajuste em registrador, caso o TPW-03 não reconheça a IHM, será apresentada no display uma mensagem “PLC Time Out!”.

Composição da IHM:



Existe na memória de dados do TPW-03 um total de 1300 registradores que servirão para o armazenamento dos arquivos de texto.

| Arquivos de Texto PLC / File Set... | | | | | | | | | |
|--|-------|----|-------|----|---------------|-------|----|-------|----|
| Posição no LCD | 01 | 02 | 03 | 04 | 05 - 16 | 17 | 18 | 19 | 20 |
| File 001 | D2000 | | D2001 | | D2002 – D2007 | D2008 | | D2009 | |
| File 002 | D2010 | | D2011 | | D2012 – D2017 | D2018 | | D2019 | |
| File 003 | D2020 | | D2021 | | D2022 – D2027 | D2028 | | D2029 | |
| --- | --- | | --- | | --- | --- | | --- | |
| File 128 | D3270 | | D3271 | | D3272 – D3277 | D3278 | | D3279 | |
| File 129 | D3280 | | D3281 | | D3282 – D3287 | D3288 | | D3289 | |
| File 130 | D3290 | | D3291 | | D3292 – D3297 | D3298 | | D3299 | |

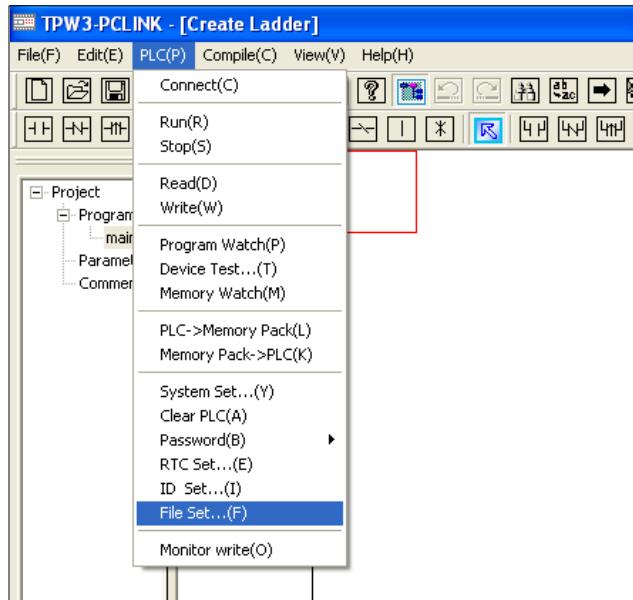
Os comandos podem ser realizados através de teclas de funções, que obedecem a um endereçamento pré-definido, conforme mostrado abaixo:

| Tecla de Função | Número do Relé Interno | Tecla de Função | Número do Relé Interno | Tecla de Função | Número do Relé Interno |
|-----------------|------------------------|-----------------|------------------------|-----------------|------------------------|
| F1 | M8280 | F9 | M8288 | TMR | M8296 |
| F2 | M8281 | F10 | M8289 | CNT | M8297 |
| F3 | M8282 | F11 | M8290 | ENT | M8298 |
| F4 | M8283 | F12 | M8291 | MOD1 | M8299 |
| F5 | M8284 | Seta Acima | M8292 | MOD2 | M8300 |
| F6 | M8285 | Seta Abaixo | M8293 | ESC | M8301 |
| F7 | M8286 | Seta Esquerda | M8294 | Reservado | M8302 |
| F8 | M8287 | Seta Direita | M8295 | Reservado | M8303 |

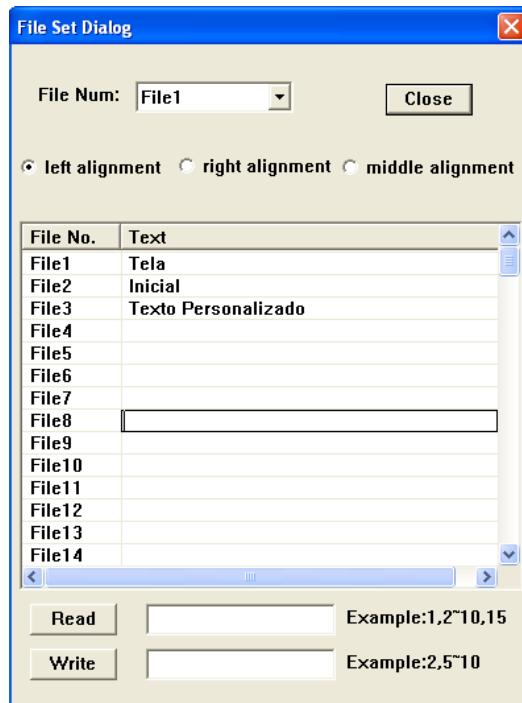
A programação do TPW-03 para utilização das interfaces homem-máquina está relacionada com o ajuste de determinados registradores. Para a configuração do display é necessário definir inicialmente o modo de operação desejado, e depois proceder aos ajustes dos registradores. A seguir apresentaremos como efetuar estes ajustes.

Para todos os modos onde se deve transferir um valor para ler um arquivo de texto, este arquivo deve ser digitado na tabela de textos.

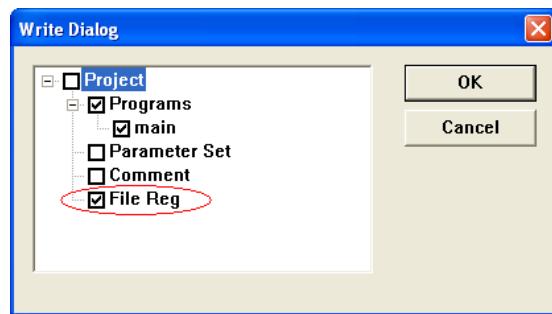
- 1- Em modo de edição vá ao menu PLC e selecione a opção “File Set...”.



2- Digite o texto no arquivo correspondente.



3- No download do programa (comando Write)selecione a opção “File Reg”.



MODO INICIAL

Utilização: Este modo pode ser aplicado para mostrar no display uma mensagem de apresentação, estado inicial da máquina ou processo, uma possível orientação ao operador, etc.

Programação: É feita diretamente no aplicativo ou através da edição dos registradores específicos.

Parâmetros:

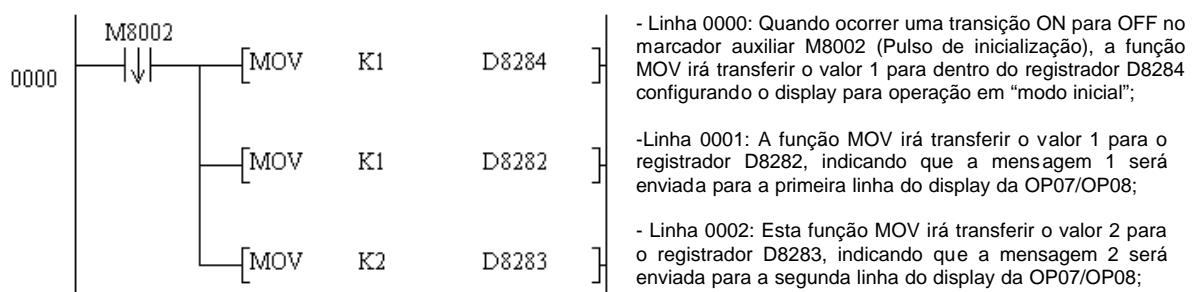
- Registrador D8284 – define o modo de operação do display da interface;
- Registrador D8282 – define qual das mensagens será apresentada na Linha 1 do display sob modo inicial de display;
- Registrador D8283 – define qual das mensagens será apresentada na Linha 2 do display sob modo inicial de display;

Seqüência de Programação:

- Ajustar o registrador D8284;
- Ajustar o registrador D8282;
- Ajustar o registrador D8283;

Exemplo de Programação:

| Mensagem (File) | Texto | Nº. da Linha do display |
|--------------------|-----------------------|-------------------------|
| 1 | WEG S/A. | 1 |
| 2 | Tecle "F1" para Ligar | 2 |



MODO DE DISPLAY F-192

Utilização: Este modo pode ser aplicado para apresentar no display mensagens de estado da máquina e/ou processo, mensagens de alarme, telas de ajuste ou visualização de parâmetros, etc.

Programação: É feita diretamente através da função TEXTP;

NOTAS:

1) O display da IHM OP-07/OP-08 é composto por duas linhas, denominadas de linha 1 e linha 2 (superior e inferior, respectivamente), com limite de 20 caracteres por linha em código ASCII, não sendo possível a acentuação ortográfica.

2) Escrita/Leitura de variáveis:

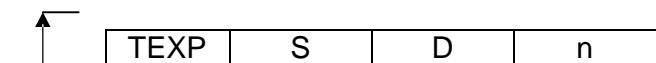
Quando se deseja enviar uma mensagem para a IHM OP-07/OP-08 com campos de leitura e/ou escrita de variáveis, deve-se inserir no texto da tela caracteres que são classificados como “especiais”. São eles:

- ? à Caractere para escrita na memória do CLP
- # à Caractere para leitura da memória do CLP

A ausência destes caracteres define um “texto simples”, que não tem a intenção de escrever ou ler da memória do CLP, geralmente utilizado para indicação de estado das máquinas/processos ou partes destes.

3) Em modo de display F-192 pode-se inserir apenas 1 campo de escrita/leitura por linha da IHM. Os campos de escrita/leitura da IHM OP-07/OP-08 em modo F-192 sempre trabalham com dados do tipo “Word” (16 bits). Dessa forma, apenas os primeiros 5 caracteres especiais serão reconhecidos como campos de escrita/leitura, podendo existir caracteres normais entre os caracteres especiais.

A função TEXTP:



A função TEXTP sempre é executada na mudança de estado das condições de entrada, de nível 0 para 1. Os campos tem as seguintes funções:

S à Define o número da mensagem (File) a ser parametrizado. Os dois registradores D8280 e D8281 trabalham com as linhas 1 e 2 do display, respectivamente.

D à Define a variável de escrita/leitura na tela da IHM. Para campos de leitura, o mesmo endereço declarado na função TEXTP terá o valor exibido na tela da IHM. Para campos de escrita, o endereço onde os dados digitados serão escritos será D + 1.

N à Define quantas linhas serão enviadas à tela da IHM.

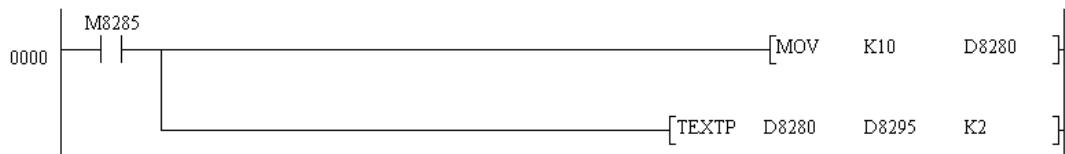
Exemplos de Programação:

A) Considere as seguintes condições:

| Mensagem (File) | Texto | Nº. da Linha do display |
|-----------------|------------------|-------------------------|
| 10 | WEG S/A. | 1 |
| 11 | Texto de Exemplo | 2 |



Como as mensagens (File) estão em seqüência, pode-se programar também de forma simplificada, conforme abaixo:



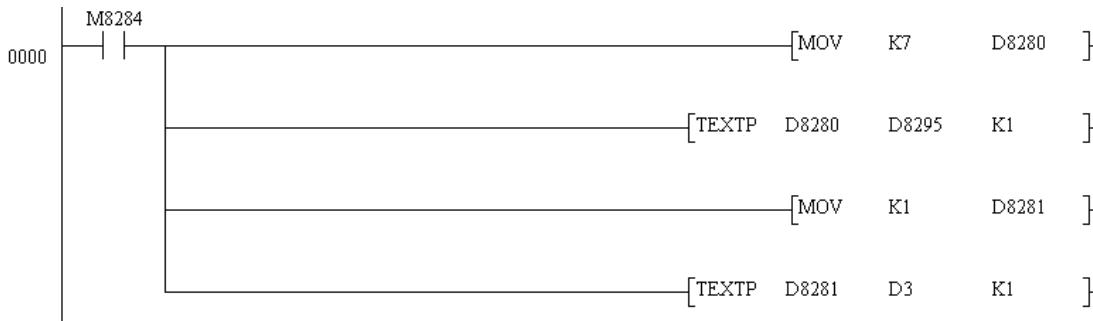
- Linha 0000: Esta função MOV irá transferir o valor 10 para o registrador D8280, indicando que a mensagem 10 será enviada para a linha 1 do display da OP07/OP08, através da função TEXTP;

- Linha 0001: Habilita a função TEXTP. O primeiro campo define a mensagem que irá aparecer na tela da IHM, neste caso o valor em D8280, arquivo FL10. O segundo campo define as variáveis de escrita/leitura, que neste caso não são utilizadas, pois não temos caracteres especiais na tela. O terceiro campo define quantas linhas esta função TEXTP irá configurar, como neste caso estamos utilizando duas linhas, a segunda linha será o arquivo definido em D8280 + 1, ou seja, o arquivo FL11;

Com o terceiro parâmetro em 2, os arquivos exibidos na tela da IHM ficam em seqüência. Para exibição de arquivos que não estejam em seqüência, se faz necessário declarar duas funções TEXTP, uma para cada linha, definindo seu respectivo arquivo.

B) Considere as seguintes condições:

| Mensagem (File) | Texto | No. da Linha do display |
|--------------------|---------------------|-------------------------|
| 7 | Escrita no Registro | 1 |
| 1 | D0004 = ????? | 2 |



- Linha 0000: Esta função MOV irá transferir o valor 7 para o registrador D8280, indicando que a mensagem 7 será enviada para a linha 1 do display da OP07/OP08, através da função TEXTP;

- Linha 0001: Esta função TEXTP irá enviar a mensagem configurada no registrador D8280 para a primeira linha do display da OP07/OP08. O segundo parâmetro define a variável de escrita/leitura, que neste caso não é utilizada, pois nesta mensagem não existem caracteres especiais. O terceiro parâmetro define que esta função TEXTP irá enviar apenas uma linha à tela da IHM;

- Linha 0002: a função MOV irá transferir o valor 1 para o registrador D8281, indicando que a mensagem 1 será enviada para a linha 2 do display da OP07/OP08, através da função TEXTP;

- Linha 0003: Esta função TEXTP irá enviar a mensagem configurada no registrador D8281 para a segunda linha do display da OP07/OP08. O segundo parâmetro define a variável de escrita/leitura, neste caso iremos escrever o valor no registrador D4, portanto a variável D3 deve ser declarada na função. O terceiro parâmetro define que esta função TEXTP irá enviar apenas uma linha à tela da IHM;

Nota: No FL07 não temos nenhum caractere especial (escrita ou leitura), portanto podemos utilizar qualquer variável em seu campo que a função TEXTP não terá efeito sobre a mesma;

C) Considere as seguintes condições:

| Mensagem (File) | Texto | No. da Linha do display |
|--------------------|---------------------|-------------------------|
| 25 | Leitura do Registro | 1 |
| 20 | D0009 = ##### | 2 |



- Linha 0000: Esta função MOV irá transferir o valor 25 para o registrador D8280, indicando que a mensagem 25 será enviada para a linha 1 do display da OP07/OP08, através da função TEXTP;

- Linha 0001: Esta função TEXTP irá enviar a mensagem configurada no registrador D8280 para a primeira linha do display da OP07/OP08. O segundo parâmetro define a variável de escrita/leitura, que neste caso não é utilizada, pois nesta mensagem não existem caracteres especiais. O terceiro parâmetro define que esta função TEXTP irá enviar apenas uma linha à tela da IHM;

- Linha 0002: a função MOV irá transferir o valor 20 para o registrador D8281, indicando que a mensagem 20 será enviada para a linha 2 do display da OP07/OP08, através da função TEXTP;

- Linha 0003: Esta função TEXTP irá enviar a mensagem configurada no registrador D8281 para a segunda linha do display da OP07/OP08. O segundo parâmetro define a variável de escrita/leitura, neste caso iremos ler o valor do registrador D9, portanto a variável D9 deve ser declarada na função. O terceiro parâmetro define que esta função TEXTP irá enviar apenas uma linha à tela da IHM;

Nota: No FL25 não temos nenhum caractere especial (escrita ou leitura), portanto podemos utilizar qualquer variável em seu campo que a função TEXTP não terá efeito sobre a mesma;

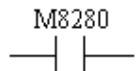


- Quando realizada qualquer alteração nos arquivos de mensagens “File”, não esquecer de selecionar a opção “File Reg” nas opções de download (comando Write)!

MODO 1 DE AJUSTE DO USUÁRIO (Receita de entrada de dados de duas variáveis)

Para ativar este modo devem-se executar os seguintes passos:

1 – Insira um contato para acionar a lógica. Exemplo: (M8280 = Tecla F1 da IHM)



2 – Insira em seu programa:

Uma função transferindo para D8284 o valor 4 (habilita modo de visualização).

Uma função transferindo o número do texto a ser visualizado na primeira linha para D8280.

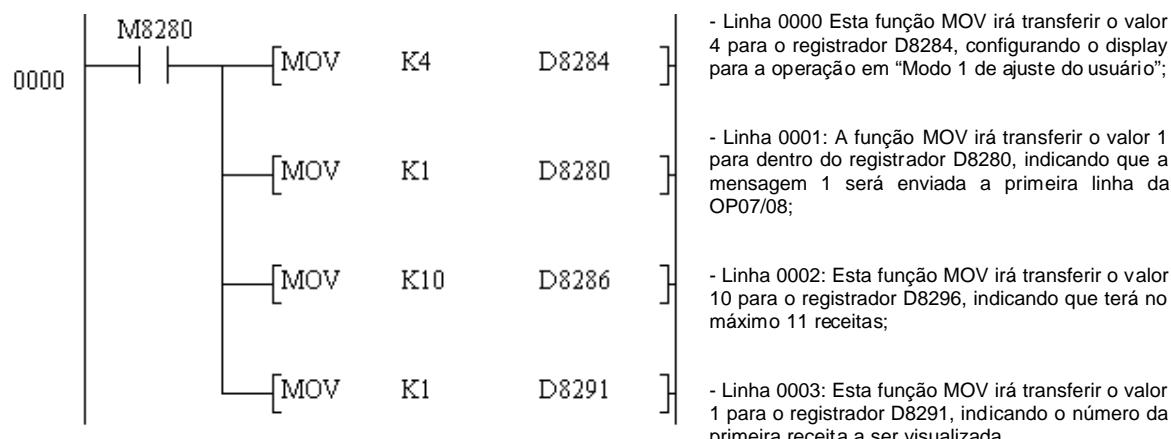
Uma função transferindo para D8286 o numero máximo de receitas (0 – 20).

Uma função transferindo para D8291 o número da primeira receita a visualizar (1 – D8296).

Tabela de Informações dos registradores:

| Nº. | Valor em D8291 | Valor de Ajuste 1 | Valor de Ajuste 2 |
|-----|----------------|-------------------|-------------------|
| 0 | 0 | D4124 | D4125 |
| 1 | 1 | D4126 | D4127 |
| ... | ... | ... | ... |
| 19 | 19 | D4162 | D4163 |

Exemplo:



Para este modo podemos alterar o formato dos campos de escrita/leitura, definindo o número de dígitos e a posição do ponto decimal. Os registradores D8297 e D8298 fazem esta configuração da seguinte forma:

D8297 / D8298 – Byte alto: Define quantos dígitos serão mostrados no campo (1 – 5)
 Byte baixo: Define a quantidade de casas decimais (0 – 4)

Exemplo: D8297 = H0502 – o campo exibirá 5 dígitos e 2 casas decimais.
 D8298 = H0301 – o campo exibirá 3 dígitos e 1 casa decimal.

Note que os valores citados nos exemplos acima estão em formato hexadecimal, isto facilita a transferência para os bytes alto e baixo, dispensando cálculos auxiliares para encontrar o valor equivalente em formato decimal. A função MOV fica da seguinte forma:



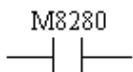
Outra configuração opcional é o sinal da variável no 1º campo de escrita/leitura, que pode ser negativo (-) ou positivo (+) de acordo com o registrador D8300.

D8300 = 1 à Sinal positivo (+)
 D8300 = 2 à Sinal negativo (-)

MODO 2 DE AJUSTE DE USUÁRIO: (Receita de entrada de dados de três variáveis)

Para ativar este modo devem-se executar os seguintes passos:

1 - Insira um contato para acionar a lógica. Exemplo: (M8280 = Tecla F1 da IHM)



2 – Insira em seu programa:

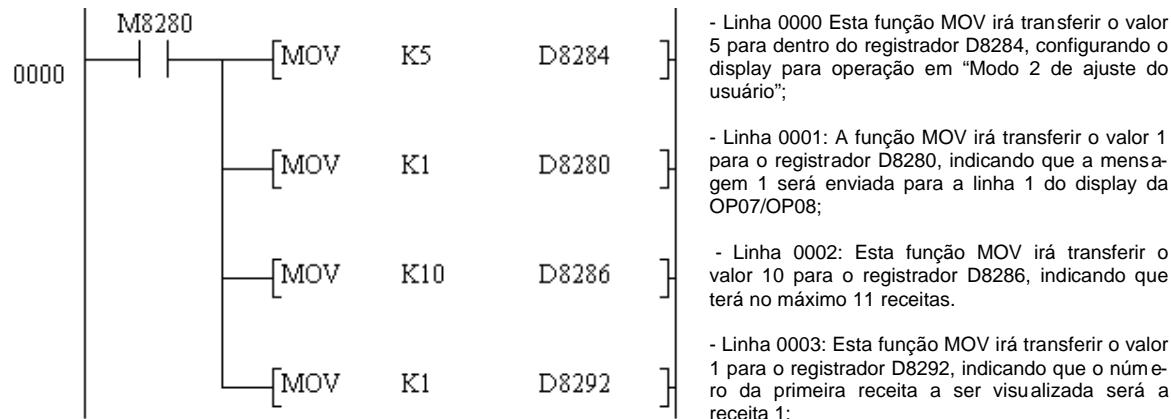
Uma função transferindo para D8284 o valor 5 (habilita modo de visualização).
 Uma função para transferir o número do texto a ser visualizado na primeira linha para D8280.
 Uma função transferindo para D8286 o numero máximo de receitas (0 – 19).
 E uma função transferindo para D8292 o numero da primeira receita (0 – D8286).

Para este modo podemos alterar o formato dos campos de escrita/leitura, conforme descrito no item 5.3.

Tabela de Informações dos registradores:

| Nº | Valor em D8292 | Valor de Ajuste 1 | Valor de Ajuste 2 | Valor de Ajuste 3 |
|-----|----------------|-------------------|-------------------|-------------------|
| 0 | 0 | D4164 | D4165 | D4166 |
| 1 | 1 | D4167 | D4168 | D4169 |
| ... | ... | ... | ... | ... |
| 19 | 19 | D4221 | D4222 | D4223 |

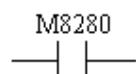
Exemplo:



MODO 3 DE AJUSTE DO USUÁRIO: (Visualização de uma variável e alteração do valor da mesma)

Para ativar este modo devem-se executar os seguintes passos:

Insira um contato para acionar a lógica. Exemplo: (M8280 = Tecla F1 da IHM)



2 – Insira em seu programa:

Uma função transferindo para D8284 o valor 6 (habilita modo de visualização).

Uma função transferindo para D8286 o número máximo de telas de leitura e entrada de dados (0 – 19).

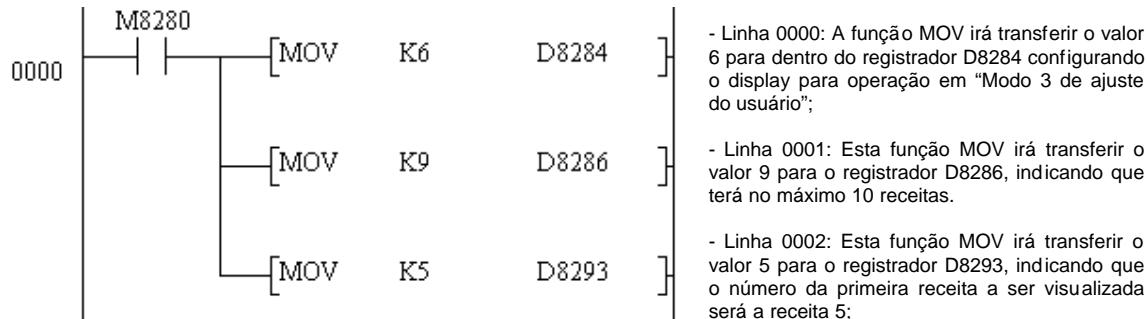
Uma função transferindo para D8293 o número da primeira tela a ser visualizada (0 – D8286).

Para este modo podemos alterar o formato dos campos de escrita/leitura, conforme descrito no modo anterior.

Tabela de Informações dos registradores:

| Nº | Valor em D8293 | Valor Atual | Valor de Ajuste |
|-----|----------------|-------------|-----------------|
| 0 | 0 | D4224 | D4224 |
| 1 | 1 | D4225 | D4225 |
| ... | ... | ... | ... |
| 19 | 19 | D4243 | D4243 |

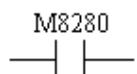
Exemplo:



MODO 4 DE AJUSTE DO USUÁRIO: (Visualização e alteração de estado de contatos)

Para ativar este modo devem-se executar os seguintes passos:

1 - Insira um contato para acionar a lógica. Exemplo (M8280 = Tecla F1 da IHM)



2 – Insira em seu programa:

Uma função transferindo para D8284 o valor 7 (habilita modo de visualização).

Uma função de transferência para transferir o número do texto a ser visualizado na primeira linha para D8280.

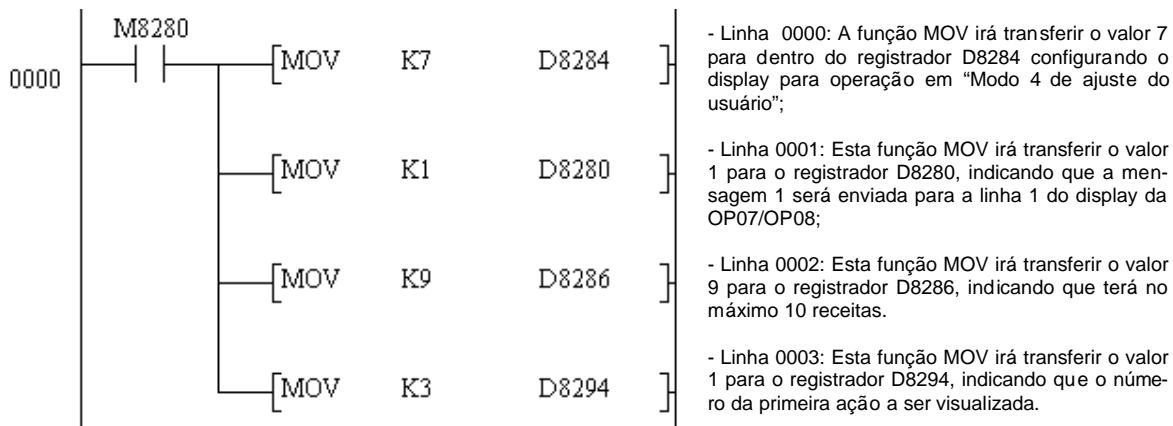
Uma função transferindo para D8286 o número máximo de ações (0 – 15).

Uma função transferindo para D8294 o número da primeira ação a ser visualizada (0 – D8286).

Tabela de Informações dos registradores:

| Nº | Valor em D8294 | Número do Marcador de Estado |
|-----|----------------|------------------------------|
| 0 | 0 | M1600 |
| 1 | 1 | M1601 |
| ... | ... | ... |
| 5 | 15 | M1615 |

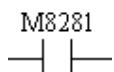
Exemplo:



MODO DE DISPLAY 1: (Visualização de três variáveis)

Para ativar este modo devem-se executar os seguintes passos:

- 1 - Insira um contato para acionar a lógica. Exemplo (M8081 = Tecla F2 da IHM)



- 2 – Insira em seu programa:

Uma função transferindo para D8284 o valor 8 (habilita modo de visualização). E uma função de transferência para transferir o número do texto a ser visualizado na primeira linha para D8280.
Para este modo podemos alterar o formato dos campos de escrita/leitura, conforme descrito no item 5.3.

Tabela de Informações dos registradores:

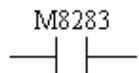
| Valor de Amostra 1 | Valor de Amostra 2 | Valor de Amostra 3 |
|--------------------|--------------------|--------------------|
| D4244 | D4245 | D4246 |

Exemplo:

**MODO DE DISPLAY 2: (Visualização de duas variáveis)**

Para ativar este modo devem-se executar os seguintes passos:

- 1 - Insira um contato para acionar a lógica. Exemplo (M8083 = Tecla F4 da IHM)

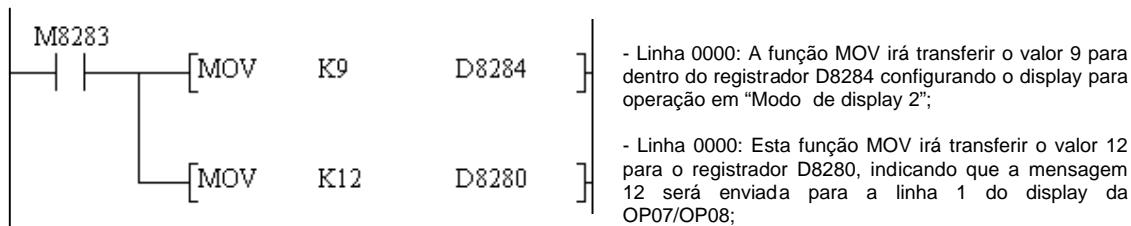


- 2 – Insira em seu programa:

E uma função transferindo para D8284 o valor 9 (habilita modo de visualização). Uma função de transferência para transferir o número do texto a ser visualizado na primeira linha para D8280.

| Valor de Amostra 1 | Valor de Amostra 2 |
|--------------------|--------------------|
| D4247 | D4248 |

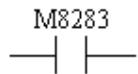
Exemplo:



MODO DE DISPLAY 3 : (Visualização de variáveis em formato binário)

Para ativar este modo deve-se executar os seguintes passos:

- 1 - Insira um contato para acionar a lógica. Exemplo (M8283 = Tecla F4 da IHM)

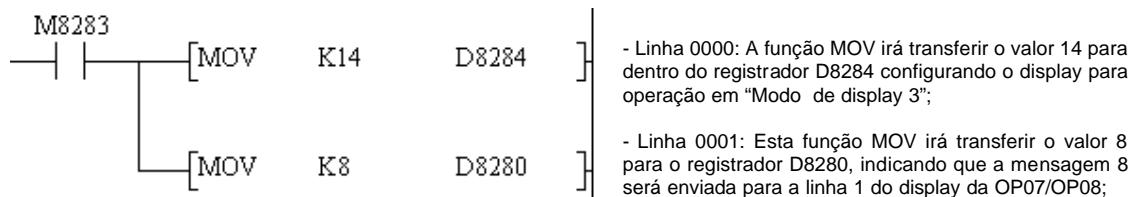


- 2 – Insira em seu programa:

E uma função transferindo para D8284 o valor 14 (habilita modo de visualização). Uma função de transferência para transferir o número do texto a ser visualizado na primeira linha para D8280.

| |
|-------------|
| Registrador |
| D4271 |

Exemplo:



1.1.15 - SOFTWARE DE PROGRAMAÇÃO ADP 6 - LINHA PWS

SOFTWARE DE PROGRAMAÇÃO ADP

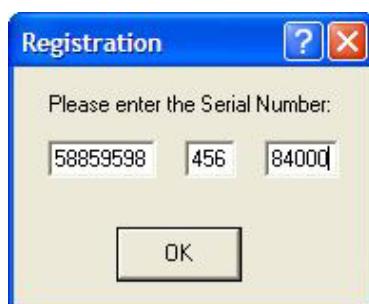
A licença do software ADP é gratuita, porém é necessário incluir um registro. Depois de instalado o programa, ao abri-lo pela primeira vez, será exibida a tela de registro do programa. Se não desejar registrar o programa no computador ou se preferir registrar mais tarde, clique em Skip.

Para registrar o software em seu computador clique em Register Now.



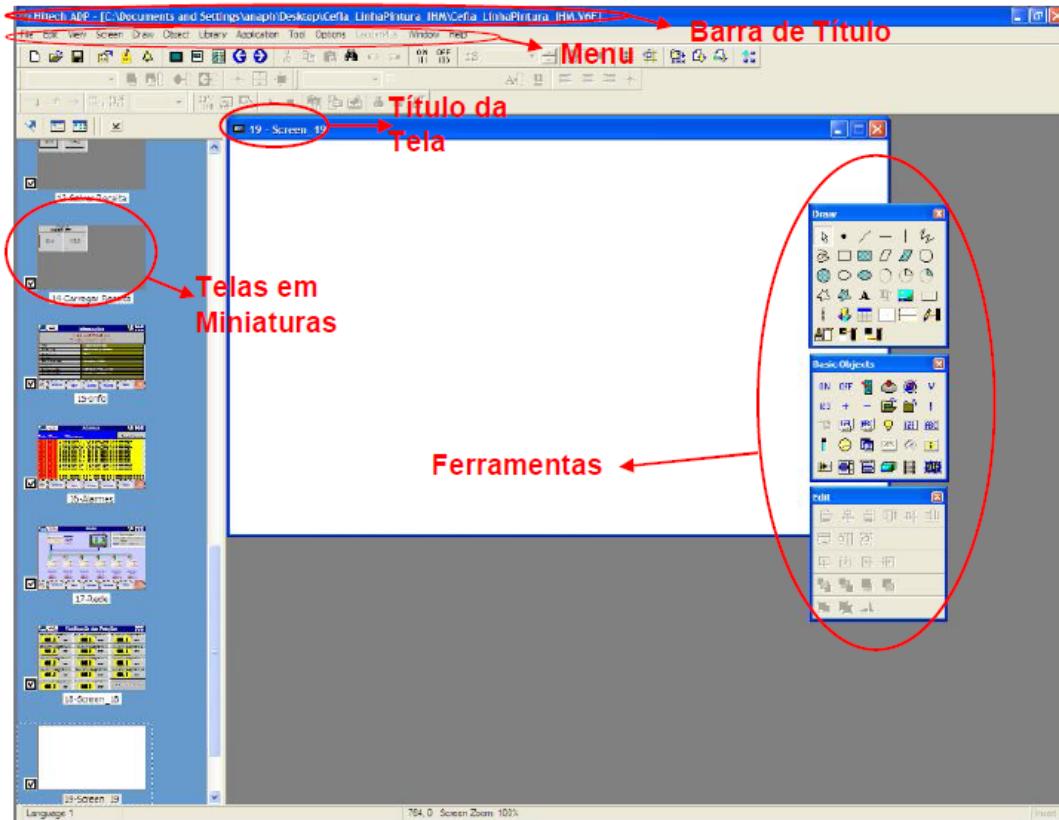
Solicitação de registro do software.

A licença para registro do software é fornecida pela Weg. Clique em Register Now e insira o código 58859598-456-84000. Para finalizar o registro clique OK.



Código de registro do ADP

Vamos analisar primeiramente sua estrutura básica, de telas quadros e menus, mostrando a disponibilidade dos recursos. Na figura abaixo, podemos visualizar a interface de programação do ADP, localizando as barras de menu e as principais caixas de ferramentas.

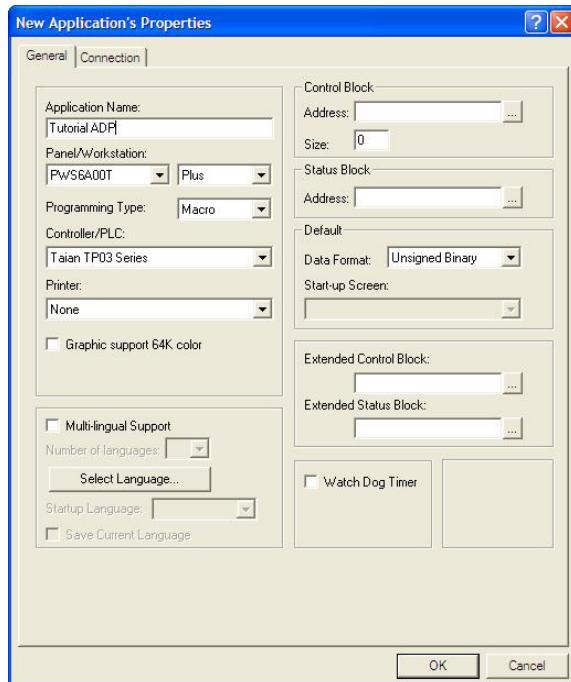


- Barra de Título. Na “Barra de Título” é visualizado o nome do diretório em que a aplicação atual está localizada.
- Menu. A barra de menus possui opções de acesso a todas funções do ADP, seleção de objetos, gerenciamento dos arquivos de programa, definição das funções de alarme, etc.
- Título da Tela. No título da tela podemos visualizar o número que a tela possui e o nome que recebeu.
- Telas em Miniaturas. No canto esquerdo da interface do software é possível visualizar as telas da aplicação, em formato de miniatura. Também chamado de *Screen Manager*, este quadro permite o gerenciamento das telas da aplicação (apagar, renomear, copiar, colar, etc).
- Ferramentas. São as barras de atalhos para os objetos e ferramentas do ADP.

DESENVOLVENDO UM PROJETO

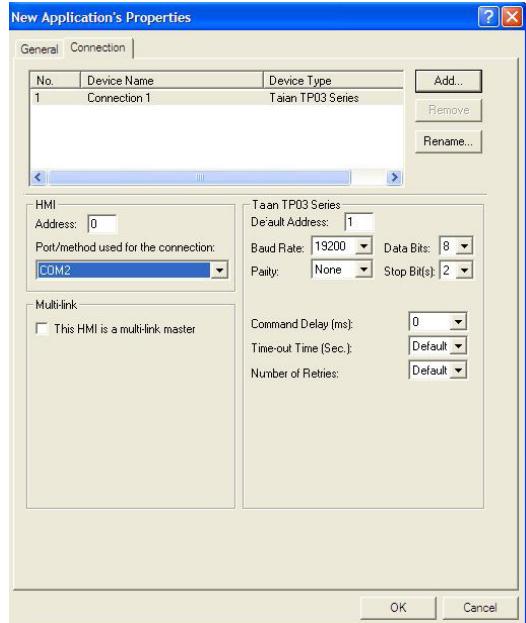
Criando uma Nova Aplicação

Primeiramente vamos dar início a uma nova aplicação, para isto clique no menu *File >>New*. A tela mostrada pela figura abaixo será aberta. O primeiro passo é definir o nome da aplicação, no campo *Application Name*. É necessária, primordialmente, a seleção do modelo da IHM utilizada, através do campo *Panel/Workstation*. Esta definição irá habilitar/desabilitar funções do software ADP conforme os recursos do modelo selecionado. No campo *Controller/PLC*, será selecionado o driver utilizado para a comunicação com o CLP ou outros dispositivos do sistema. Para o tutorial, utilizaremos a IHM PWS-6A00T em conjunto com o CLP TPW-03. Defina os campos conforme mostrado na figura abaixo.



Definições iniciais da aplicação

Após selecionar o driver de comunicação, automaticamente será exibida a aba *Connection*, onde será possível ajustar os parâmetros de comunicação, conforme as peculiaridades de cada driver. Esta etapa será posteriormente tratada, podendo ser omitida neste momento. Clique *OK* para confirmar a criação da aplicação.

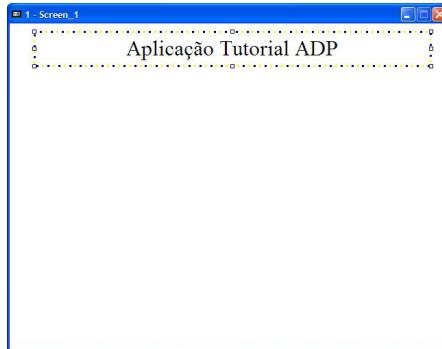


Configurações de Comunicação do driver TPW -03

Objeto de Texto

Após criar a aplicação, automaticamente será criada uma tela da IHM, conforme a resolução do modelo selecionado. Vamos aplicar um objeto de texto para colocar um título nesta tela.

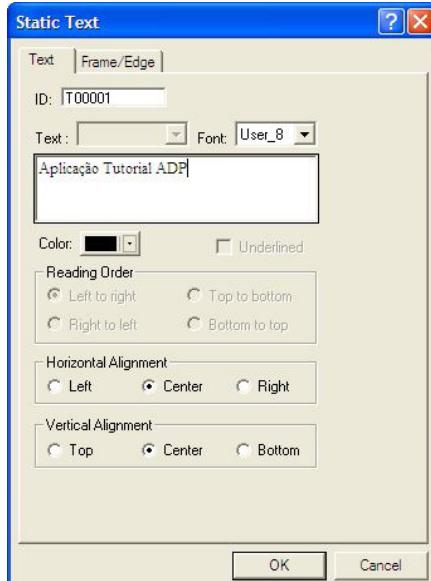
Com um clique no botão **A** iremos selecionar o objeto *Static Text*, inserindo-o na parte superior da tela, como mostrado pela figura abaixo.



Título inserido na tela

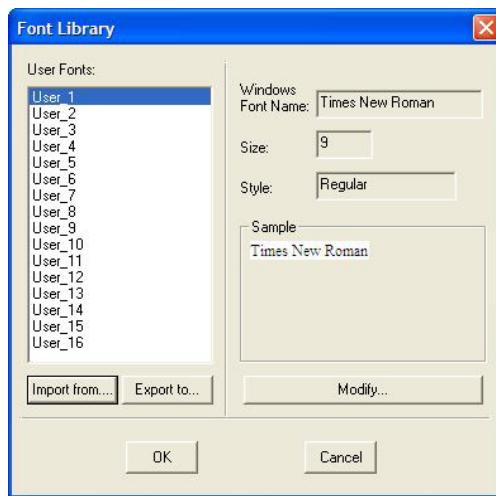
Para alterar o conteúdo do campo de texto, dê dois cliques no campo em que você deseja alterar. Será aberto o popup de propriedades do objeto *Static Text*, conforme visto na figura abaixo. Na opção *Font* você pode escolher o modelo da fonte para seu texto, logo abaixo existe um campo em branco onde será inserido o conteúdo do campo, ou seja, o título da tela. As opções *Horizontal Alignment* e *Vertical Alignment* são utilizadas

para o alinhamento do texto. Lembrando que o campo do texto deverá ser sempre maior que o texto, caso contrário ocorrerá um erro na compilação do programa.



Propriedades do objeto Static Text

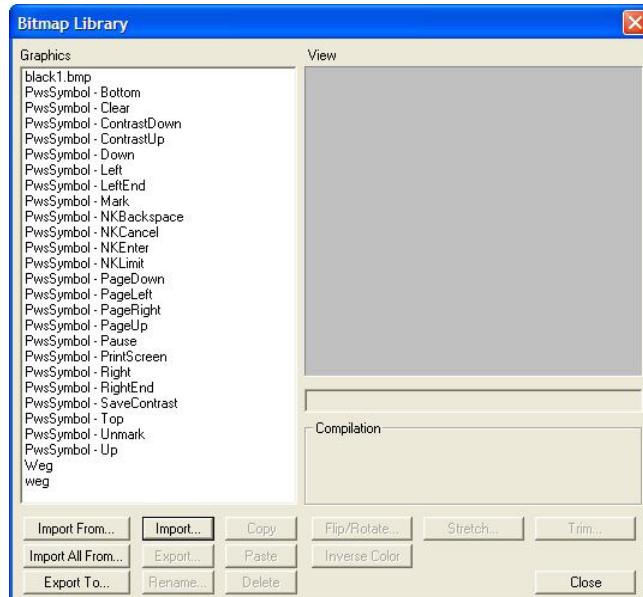
Existem dois tipos diferentes de fonte, o modelo padrão do ADP (8x8, 8x16, etc) e as fontes personalizadas (User_1, User_2, etc), que podem ser definidas pelo usuário, sendo configuradas através da biblioteca de fontes, disponível através do menu *Library >> Font Library*. A tela mostrada pela figura abaixo será aberta. Para alterar as configurações da fonte desejada, clique na opção *modify*. É possível definir o tipo de fonte, o tamanho, o estilo (negrito, itálico, etc) e efeitos adicionais (riscado e sublinhado). Estas definições serão válidas para qualquer objeto da aplicação que utilize as fontes personalizáveis. Para salvar as definições clique *OK*.



Biblioteca de fontes

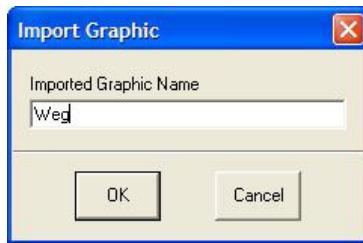
Importando Imagens

Da mesma forma como importamos fontes para a aplicação, podemos adicionar imagens. O ADP permite gerenciar a biblioteca de imagens, suportando importação dos formatos BMP, JPG, DWG e GIF. Esta biblioteca é individual por aplicação, ou seja, figuras adicionadas em uma aplicação não estarão disponíveis automaticamente em outra aplicação, a menos que sejam novamente adicionadas. Agora vamos importar uma imagem qualquer para ser utilizada na aplicação. Clique no menu *Library >> Bitmap Library*. Será mostrada a tela da figura abaixo.



Bitmap Library – biblioteca de imagens

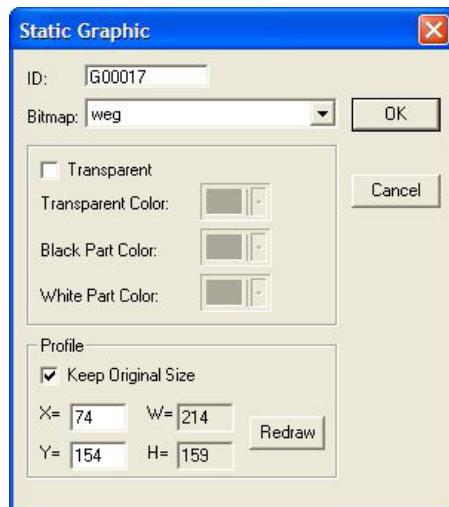
Clique em *Import*, em seguida abrirá uma tela para selecionar uma imagem desejada. Ao prosseguir em *Open*, será visualizada a tela da figura abaixo, onde será definido um nome interno para a imagem.



Definindo o nome da imagem importada

Após inserir a imagem na biblioteca, podemos adicioná-la na tela da IHM. Para isto, clique em para selecionar o objeto *Bitmap* e crie o objeto no local desejado. Em seguida, clique duas vezes sobre o objeto para visualizar suas propriedades, conforme figura

abaixo. Selecione então a imagem que importamos no campo *Bitmap* e clique *Ok* para salvar. Para redimensionar a imagem, desmarque a opção *Keep Original Size*, pois ela bloqueia o redimensionamento da imagem, mantendo o seu tamanho original. É possível definir também uma cor da imagem para ser reconhecida como transparente. Basta habilitar a opção *Transparent* e selecionar a cor desejada.

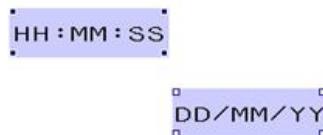


Seleção de uma imagem da biblioteca

Após a importação da imagem, o arquivo original com a mesma pode ser excluído de seu computador, pois a imagem estará armazenada na biblioteca de bitmaps, sendo salva juntamente com o arquivo da aplicação.

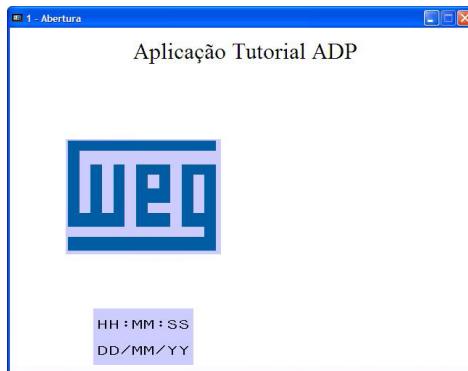
Inserindo campos de Data e Hora

No canto inferior da tela, iremos dispor os campos de hora e data. Para criar o campo de hora, clique no ícone *Time Display* e crie o objeto na tela. Repita o processo citado anteriormente para o campo de data, clicando em *Date Display* e criando-o abaixo do campo de hora. Não é necessário que os campos sejam criados exatamente com a mesma dimensão, pois existem recursos para alinhamento/dimensionamento. Pressione a tecla shift, clique sobre o campo de data e, ainda com a tecla shift pressionada, clique sobre o campo de hora, deixando os dois objetos selecionados. Antes de utilizar as opções de alinhamento, precisamos definir qual será o objeto base para a ação, para isto, pressione a tecla Ctrl e clique no campo de hora. A seleção do campo base ficará diferente dos demais campos selecionados, como visto na figura abaixo.



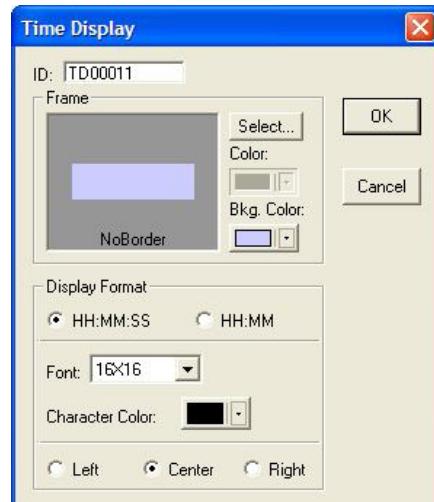
Campos de data e hora

Agora, clique no ícone Make Same Size para deixar os objetos com o mesmo tamanho. Para alinhar igualmente os dois campos, repita o processo acima e clique em um dos recursos disponíveis .



Tela com os campos de data/hora

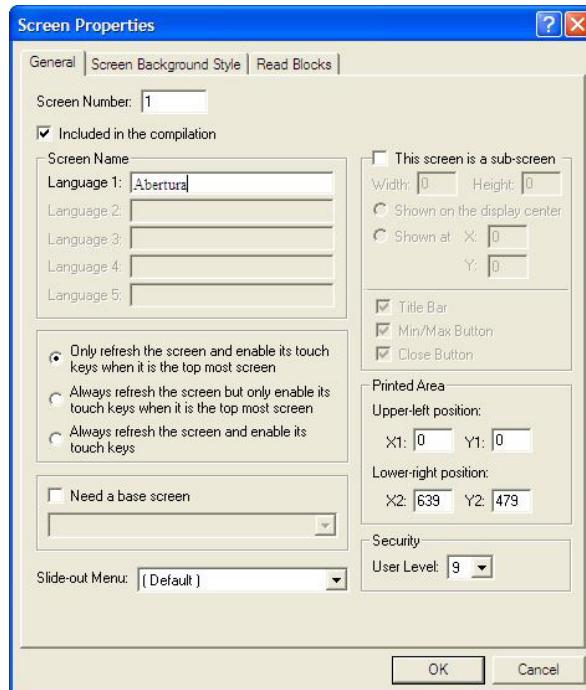
Em seguida, vamos editar o *Frame* dos objetos data e hora, que definem o estilo de exibição dos mesmos. Para isto, clique duas vezes sobre o campo de hora para abrir as propriedades do objeto. No quadro frame, pressione *Select* para escolher o estilo de frame do campo, vamos selecionar o primeiro o estilo *NoBorder*, em seguida clique *OK*. A opção *Color*, quando habilitada, serve para escolher a cor da borda do *frame*. O estilo *NoBorder* não possui bordas, portanto iremos selecionar apenas a cor do fundo, através da opção *Bkg. Color*. Como podemos observar na abaixo, temos dois modelos diferentes de horário, por padrão vamos deixar a primeira opção habilitada. Logo abaixo é possível modificar a fonte do texto, a cor e o alinhamento, em seguida clique *Ok*. Os mesmos processos serão seguidos para o campo de data.



Configurações do objeto Time Display

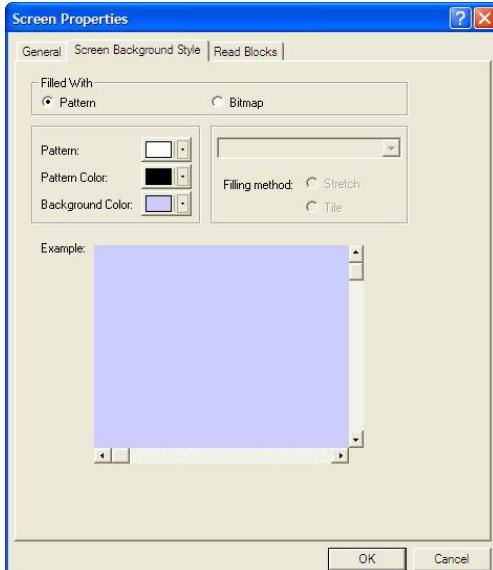
Propriedades da Tela

As telas da IHM possuem propriedades específicas, como cor de fundo, título, número de referência, etc. Iremos definir as propriedades de nossa primeira tela, clicando com o botão direito do mouse sobre a tela e escolhendo a opção *Screen Attributes*. A imagem da figura abaixo será mostrada. No campo *Screen Number* é possível mudar o número de referência da tela e na opção *Screen Name* podemos editar o nome da tela. O campo *Need a base screen* poderá ser ativado quando utilizarmos uma tela de base para esta tela, este assunto será retratado melhor posteriormente. Em *Slide-out Menu* podemos definir a ação das teclas 22 da IHM, quando houverem. Por padrão, o campo vem com o valor *Default*. Quando a função *This screen is a sub-screen for* habilitada é possível modificar o tamanho da tela, tornando-a um popup. Pode ser escolhida a localização que a mesma abrirá na IHM, selecionando a função *Shown at*, logo ao lado disponibilizaremos o número da localização desejada, conforme as coordenadas X/Y da tela. A barra de status do ADP fornece as coordenadas X/Y quando o cursor estiver na tela. Em *Security* podemos escolher o nível de acesso que a tela receberá. Por enquanto, deixaremos as opções conforme mostrado na figura abaixo.



Propriedades gerais da tela

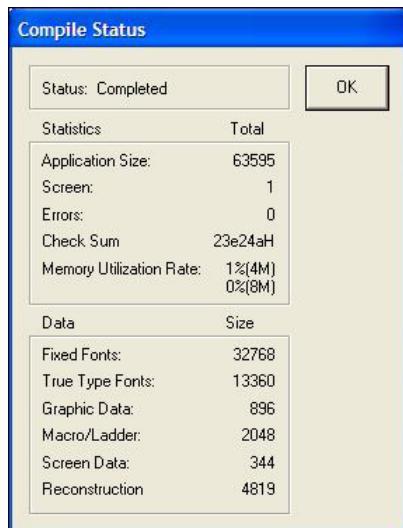
O próximo passo é modificar a cor do fundo de nossa tela, através da aba *Screen Background Style*. A opção *Pattern* disponibiliza alguns modelos de fundo e nos campos *Pattern Color* e *Background Color* podemos escolher a cor do fundo. Lembrando que a segunda opção somente será visualizada se o fundo possuir algum modelo com duas cores. Para confirmar as alterações realizadas clique em *OK*.



Seleção de cores para o fundo de uma tela

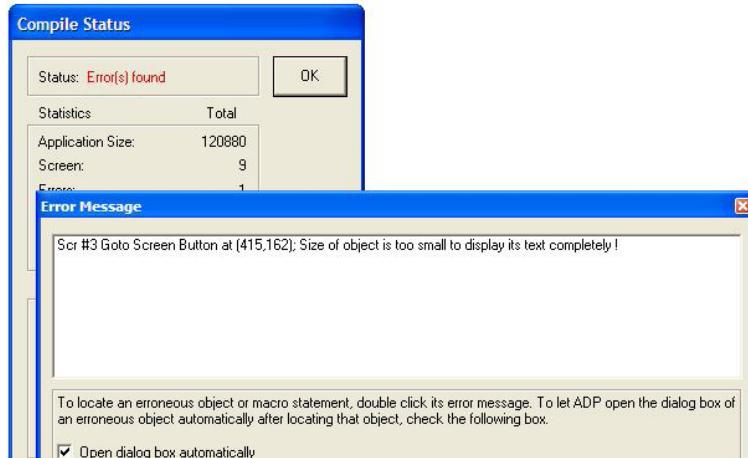
Simulação a Aplicação

O ADP permite executar a simulação do programa, auxiliando muito no desenvolvimento das aplicações, pois possibilita os testes de navegação e melhor visualização do layout das telas. Antes de rodar a simulação, precisamos salvar o programa, o que pode ser feito através do ícone Save. Escolha um caminho de sua preferência para armazenar os arquivos da aplicação. Para a simulação ocorrer, é necessário compilar o programa em *Application >> Compile*. Se a compilação for concluída com sucesso será mostrada a tela da figura abaixo.



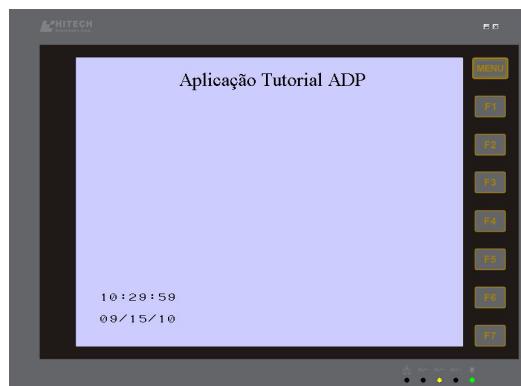
Compilação completada com sucesso

Caso ocorra algum erro, a mensagem *Error(s) found* aparecerá no campo *Status*. Clicando *OK*, um popup com a listagem dos erros será exibida, conforme figura abaixo. Com um duplo clique sobre o erro, o local de ocorrência do erro será automaticamente aberto. Após verificar e corrigir os erros compile o programa novamente.



Erros detectados na compilação

Para rodar a simulação, clique no menu *Tool >> Off-line Simulation*. A seguinte tela será visualizada.

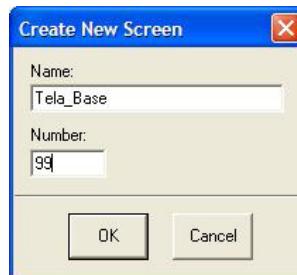


Simulação da aplicação

Criando Uma Nova Tela

Na aplicação desenvolvida neste tutorial, iremos criar várias telas, com funções específicas, tentando aproximar-se da estrutura de uma aplicação real. O ADP oferece um recurso muito útil, que permite a criação de telas base, que funcionam como um espelho, refletindo seus objetos na tela em que definimos referência a uma tela base. No próximo passo, criaremos uma tela de base e a personalizaremos para contemplar o layout das telas e, futuramente, os botões de navegação.

Para a criação de uma nova tela, clique em *Screen > New Screen*. Automaticamente abrirá uma tela com os campos *Name*, onde você colocará o nome desejado para a sua tela, e *Number*, para definir o número de referência da mesma. Sugestivamente, definiremos o número de referência desta tela como 99, para não interferir na seqüência de numeração das telas de processo da aplicação, que se iniciaram a partir da tela 01. Ao pressionar o botão *OK*, a tela aparecerá na lista de miniaturas do *Screen Manager*.



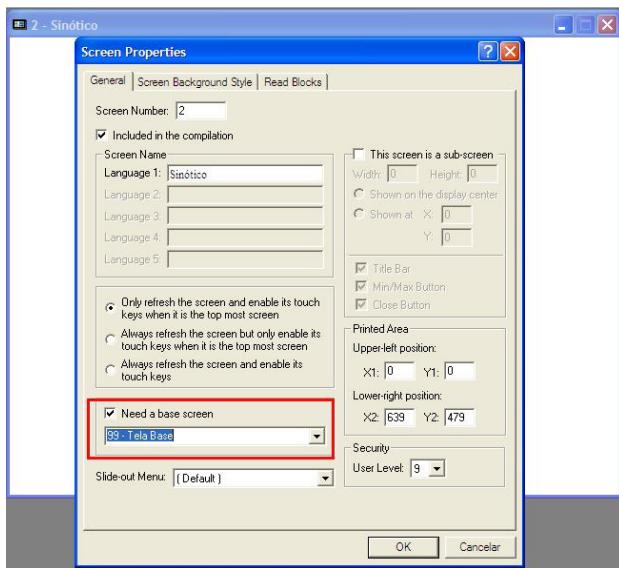
Definição do nome e número de uma nova tela

Após a criação da tela, vamos colocar um retângulo onde será disposto o título, clicando em *Solid Rectangle* e dimensionando o objeto. Para alterar as propriedades do retângulo, basta clicar duas vezes sobre o mesmo. Selecionado a função *Clipped*, no quadro *Type*, podemos configurar o arredondamento dos cantos. No quadro *Frame*, podemos alterar o estilo e a cor da linha de contorno. As opções de preenchimento podem ser alteradas no quadro *Fill*, que pode ser desabilitado, tornando o seu interior transparente. No quadro *Profile*, podemos modificar a localização do retângulo na tela. No canto direito da tela colocaremos o campo de hora e data e em volta um retângulo, seguindo os mesmos passos descritos anteriormente.



Tela base

Com a tela base concluída, iremos criar uma nova tela para aplicá-la. Da mesma forma como criamos a tela base, criaremos agora a tela “Sinótico”, conforme os passos descritos no início deste item. Após a criação de mais esta tela, iremos alterar suas propriedades para selecionar a tela base como referência. Clique o direito na tela “Sinótico” e acesse a opção *Screen Atributtes*. Marque a seleção *Need a base screen* e selecione a tela base – tela 99.



Associando a tela base à tela de sinótico

É importante ressaltar que somente os objetos criados na tela base serão visualizados nas telas em que a mesma for referenciada, ou seja, a cor do fundo pode ser alterada separadamente. Em seguida, crie o título “Sinótico” na parte superior da tela, dentro do retângulo.

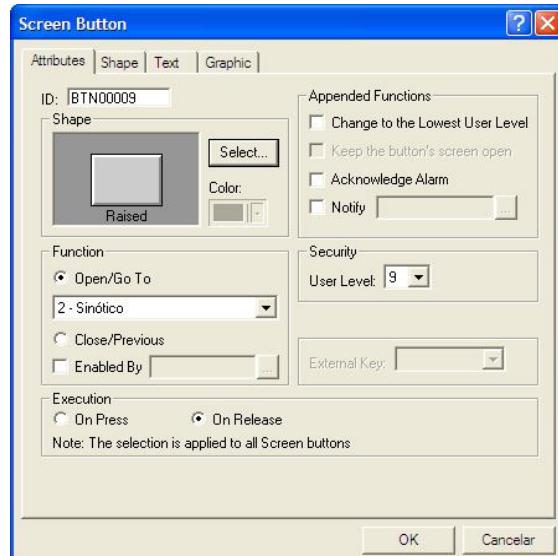


Tela base referenciada à tela de sinótico

Criando Botões de Navegação

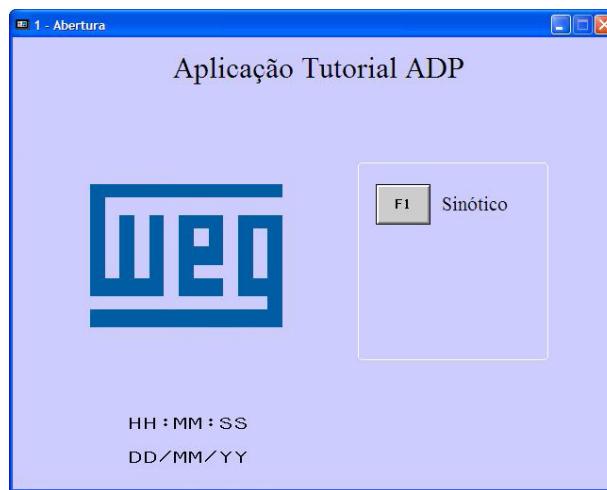
A tela de sinótico está criada, porém precisamos configurar um botão para chamar esta tela, a partir da tela inicial. Abrindo novamente a tela de abertura, vamos colocar um botão para chamar a tela de sinótico. Para isto clique em Goto Screen Button e crie o objeto na tela. Dê um duplo clique para acessar as propriedades do botão, que contém duas funções básicas, que podem ser selecionadas no quadro Function. No modo Open/Go To, o botão irá abrir a tela selecionado no Combo Box logo abaixo. No modo Close/Previous, o botão fechará a tela atual (quando estiver em um popup) ou

retornará a tela aberta anteriormente (no caso de telas). No nosso caso, selecionaremos a opção padrão Open/Go To, configurando o botão para abrir a tela “2 – Sinótico”. A aba *Shape* permite a configuração do formato e cores do botão. Na aba *Text*, podemos editar o texto explicativo do botão, que iremos definir como “F1”, pois posteriormente iremos configurar a navegação via teclas utilizando a tecla F1. Na aba *Graphic* podemos associar as imagens da biblioteca ao botão.



Propriedades do botão Goto Screen Button

Ao lado do botão criado, vamos indicar através do campo *Text* o nome da tela a ser chamada, como mostrada na figura abaixo. Em volta colocaremos um retângulo sem preenchimento através do ícone e ativaremos a função *Clipped*, logo ao lado, em *Corner* disponibilizaremos o valor de alteração do canto do retângulo.



Tela inicial com botão de navegação

Para contemplar a navegação, iremos adicionar os botões de chamada de tela na tela base, incluindo um botão para retornar a tela de abertura e outro para a tela de sinótico. Como podemos ver na figura abaixo, no botão de retorno à tela de abertura utilizaremos uma imagem, selecionada na aba *Graphic*, como citado anteriormente. Para todas as telas criadas vamos dispor, com o decorrer da aplicação, botões na tela de abertura e na tela base.



Tela base com botões de navegação

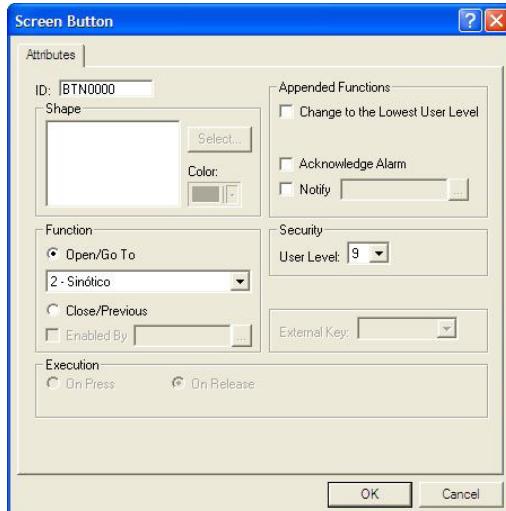
Navegação Através das Teclas da IHM

No próximo passo, iremos fazer a navegação da IHM através de suas teclas. O ADP permite a criação de perfis de teclas, os chamados *Slide-Out Menus*, que definem funções para as teclas e os textos explicativos exibidos. Quando o usuário edita uma tela, pode associar um perfil de teclas previamente criado, flexibilizando o método de associação das teclas com as funções desejadas, pois vários perfis podem ser criados e reutilizados. Para gerenciar os perfis de teclas, clique em *Application > Slide-out Menu*.



Tela de gerenciamento dos Slide -Out Menus

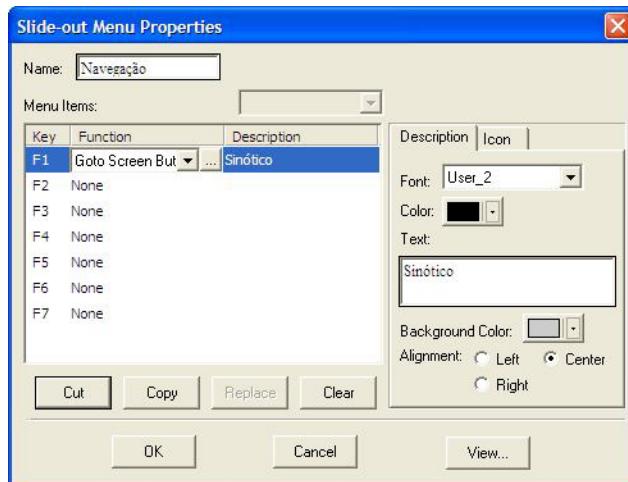
Clique em *New* para criar um perfil de teclas. Criaremos um perfil para a navegação entre telas da IHM. Primeiramente colocaremos o nome “Navegação” no campo *Name*, que define o nome do perfil de teclas. A tabela mostrada nesta tela relaciona as teclas disponíveis (conforme o modelo da IHM), as funções associadas e a descrição da tecla. No campo *Function* da tecla F1, selecione a função *Goto Screen Button*, para selecionar o botão de chamada de telas. Ao clicar no botão com reticências, serão abertas as propriedades do botão, onde será configurada a chamada da tela 2 - *Sinótico*.



Propriedades do Screen Button

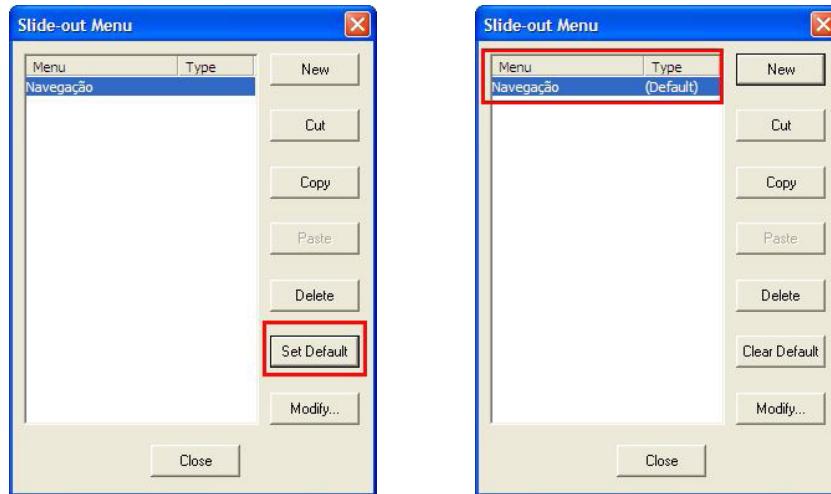
No campo *Description*, aparecerá a descrição de apresentação do botão, que será exibida quando o usuário pressionar a tecla *Menu* da IHM. Esta descrição pode ser editada no quadro à direita, através do campo *Text*. Também podemos selecionar diferentes

fontes e cores para este texto. Clique **OK** para confirmar as alterações realizadas neste perfil de teclas.

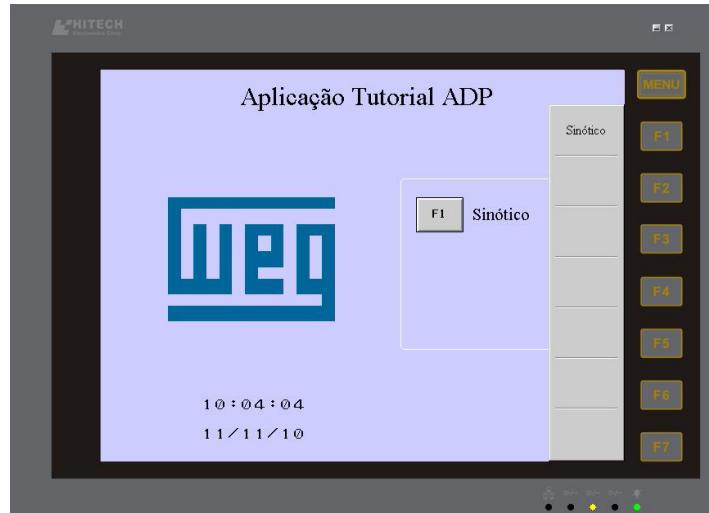


Propriedades do Slide-Out Menu

Após a criação deste perfil de teclas, precisamos associá-lo às telas desejadas para que ele seja executado. Como este perfil será utilizado para navegação entre as telas, podemos defini-lo como padrão para toda a aplicação, desta forma, não precisaremos alterar as propriedades de todas as telas até então criadas, visto que, quando a tela é criada, o *Slide-Out Default* (padrão) é automaticamente selecionado. Para definir um *Slide-Out* como padrão, selecione o *Slide-Out* desejado e clique no botão *Set Default*, conforme mostrado abaixo.



Seleção do perfil de teclas padrão



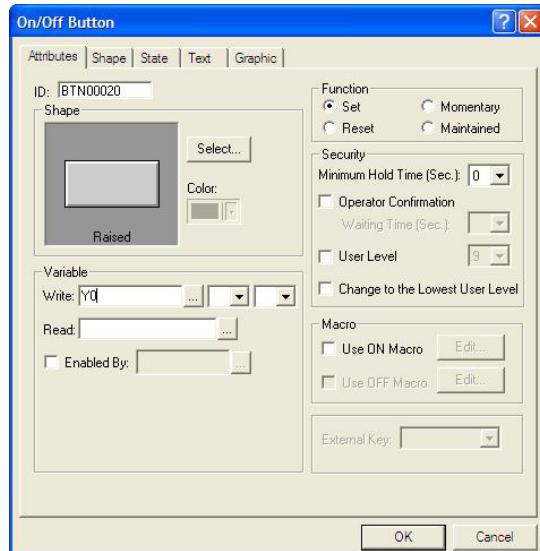
Exibição dos comandos de tecla quando pressionada tecla “MENU”

Criando Botões de Acionamento

Agora criaremos os botões para ligar e desligar a bomba, utilizando o objeto *On/Off Button*. Existem 4 tipos de botões de comando através do *On/Off Button*, sendo eles:

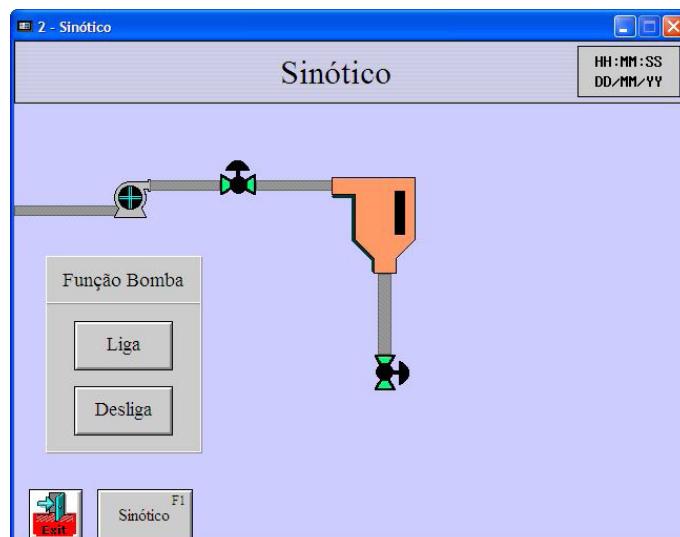
- *Set*: A variável selecionada vai para o estado ON e permanece ligada.
- *Reset*: A variável selecionada vai para o estado OFF e permanece desligada.
- *Maintained*: A variável terá seu estado invertido cada vez que o botão for pressionado.
- *Momentary*: A variável ficará ligada apenas enquanto o botão estiver sendo pressionado.

Estas opções podem ser selecionadas através das propriedades do objeto *On/Off Button*, ou pré-selecionadas através dos atalhos .. Crie um botão do tipo *Set*, clicando em *Set Button* e selecionando a variável Y0 no campo *Write*, nas propriedades do objeto, conforme figura abaixo.



Propriedades do On/Off Button

Podemos notar que o objeto também possui a propriedade *State*, acessível na aba *State*. É possível incorporar uma animação ao botão de comando. Esta animação terá como referência a variável informada no campo *Read*. Quando o campo *Read* estiver em branco, a mesma variável definida no campo *Write* ficará como referência para a animação. Na aba *Text* é possível definir um texto para o botão, lembrando que cada estado (0 ou 1) terá seu texto próprio. Defina o texto “Liga” para ambos os estados deste botão. Na aba *Graphic* podemos adicionar imagens ao botão. Da mesma forma como foi criado o botão “Liga” para a bomba, crie um botão “Desliga”, utilizando a função um *On/Off Button* com a função *Reset*. A tela ficará da seguinte forma:

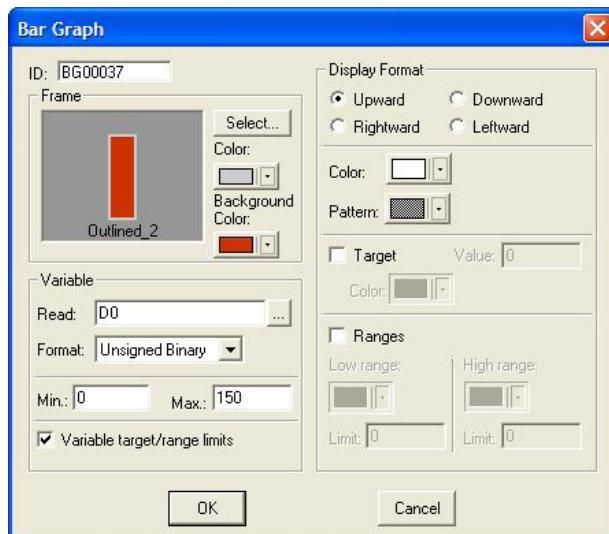


Tela de Sinótico contemplada com os comandos para a bomba

Bar Graph

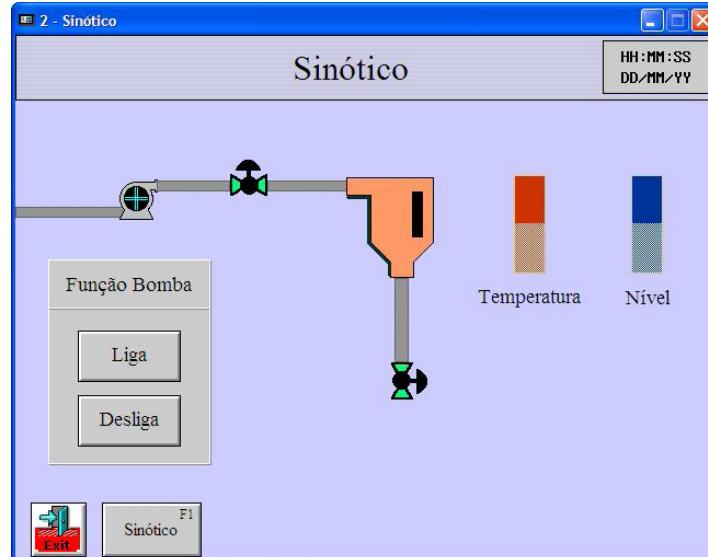
Explorando um pouco mais os objetos de animação disponíveis, iremos adicionar ao lado do desenho do tanque um objeto do tipo *Bar Graph*, que fará a indicação da temperatura e do nível do tanque. O objeto irá relacionar uma variável qualquer com uma barra de animação gráfica, que terá seu preenchimento de cor proporcional ao valor da variável configurada.

Para inseri-lo, clique no botão  *Normal Bar Graph*, e dimensione o objeto na tela. Acessando as propriedades do objeto, iremos configurar o campo *Read* para definir a variável base para a animação. Logo abaixo, indicaremos o valor mínimo e máximo que a variável poderá atingir. Este valor será relativo ao preenchimento do *Bar Graph*, de 0 a 100%. As cores de preenchimento serão definidas através do quadro *Frame*, sendo possível também alterar o estilo da barra de animação (com bordas, etc).



Propriedades do objeto Bar Graph

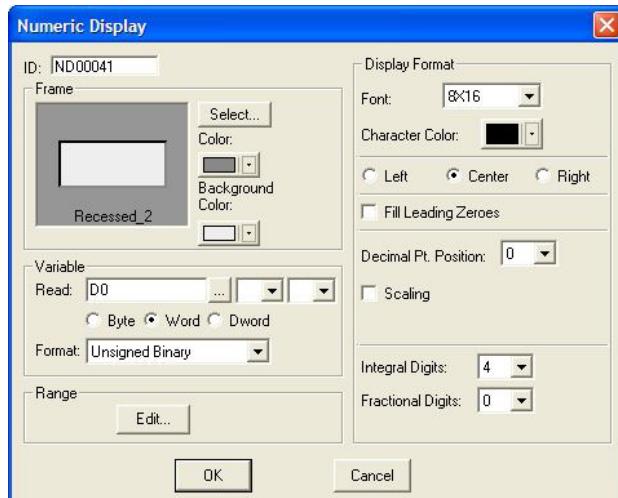
Em *Display Format*, escolheremos o estilo de preenchimento do *Bar Graph*, com as opções *Upward*, *Downward*, *Righward* e *Leftward*, respectivamente, para cima, para baixo, à direita e à esquerda. Esta opção refere-se à direção do preenchimento da cor de indicação da variável. Defina as propriedades do *Bar Graph* conforme a figura acima, com as variáveis D0 para a temperatura e D1 para o nível do tanque. Posteriormente, simularemos a alteração destas variáveis para verificar o funcionamento do *Bar Graph*.



Tela de Sinótico com os Bar Graphs de Temperatura e Nível

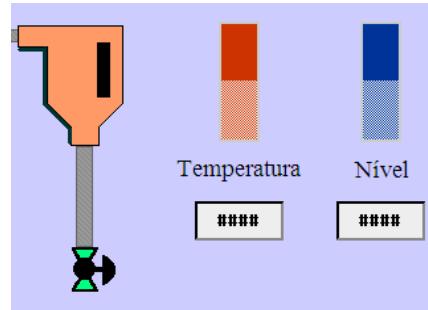
Exibindo Valores das Variáveis

Abaixo dos *Bar Graphs*, colocaremos um campo numérico com os valores absolutos de temperatura e nível. Para isto, iremos utilizar o objeto *Numeric Display*. Para adicionar este objeto na tela, clique no ícone de atalho *Numeric Display* e dimensione o objeto na tela. Para configurar o campo *Numeric Display*, clique duas vezes sobre o mesmo. O campo *Read* define a variável a ser exibida, permitindo ainda a seleção do formato de dados da variável. Através do quadro *Frame*, podemos personalizar a exibição do objeto, definindo cor de fundo, bordas, etc.



Propriedades do objeto Numeric Display

Em nossa aplicação, adicionaremos um campo de exibição de valor abaixo de cada *Bar Graph* de animação de temperatura e nível. Consequentemente, manteremos as mesmas variáveis de referência, que serão posteriormente simuladas.

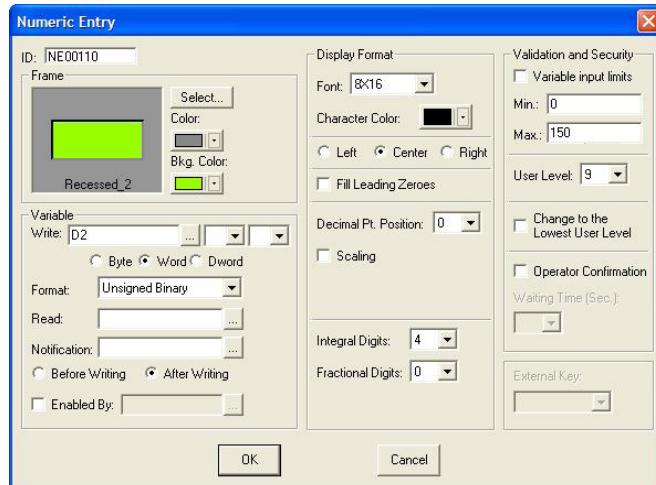


Inserção dos campos de exibição de valores numéricos

Criando Campos de Entrada de Valor

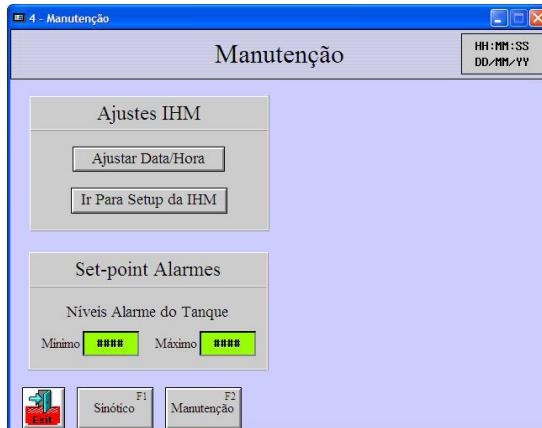
Ainda na tela de manutenção, criaremos o quadro *Set-point Alarms*, onde colocaremos dois campos para ajustar o valor máximo e mínimo do tanque, que posteriormente serão utilizados para gerar os alarmes. Após criar o quadro com o título, assim como fizemos para os ajustes gerais da IHM, utilizaremos o objeto *Numeric Entry*, que permite alterar valores de variáveis. Na tela de propriedades de *Numeric Entry* podemos escolher a quantidade de dígitos que o campo receberá em *Integral Digits*, conforme mostrado abaixo.

Ao lado direito, no menu *Validation and Security* disponibilizaremos o valor mínimo e máximo que o operador poderá digitar. Para o valor mínimo iremos referenciar a variável D2, e para o valor máximo a variável D3.



Propriedades do botão Numeric Entry

Como podemos notar, os campos de leitura foram definidos com a cor branca. Para auxiliar na identificação dos campos, para a entrada de valor utilizaremos a cor verde. Simule o programa e teste a entrada de valores. Para digitar um novo valor, clique sobre o campo de entrada. Um teclado numérico será aberto, permitindo a digitação do valor desejado.



Tela Manutenção com Set-point de Alarms



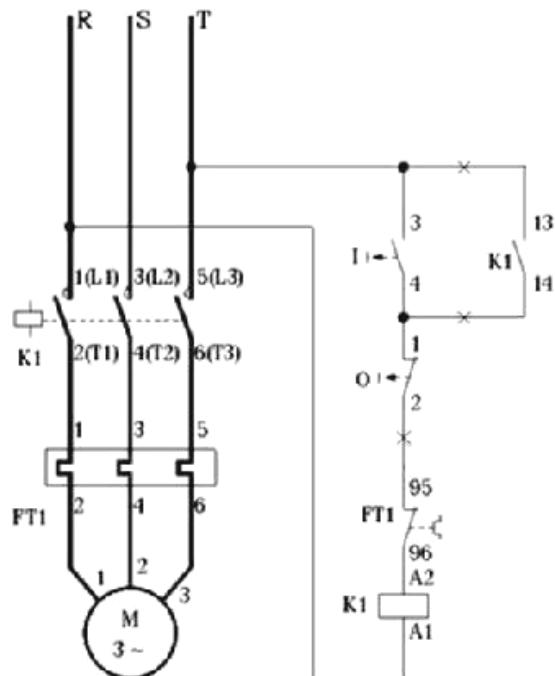
Teclado para entrada de valores

Criando Campos Para Entrada de Caracteres

Na mesma tela de manutenção, criaremos alguns ajustes pertinentes ao processo. Para isto, criaremos outro quadro, com o nome “*Set-point Processo*”, onde disponibilizaremos um campo para o operador colocar manualmente o nome do produto que estiver contido no tanque. A entrada de textos é permitida através do objeto *Character Entry*. Insira o objeto dentro do quadro recém criado. Na tela de propriedades do objeto *Character Entry* temos o campo *Number of Characters*, onde definiremos a quantidade máxima de caracteres que poderá ser digitada. Iremos definir a variável D100 para armazenar o texto digitado. Dependendo da quantidade de caracteres, o número de variáveis ocupadas irá aumentar. Neste caso, utilizando 20 caracteres, estaremos utilizando as variáveis D100 a D109 (1 variável para cada 2 caracteres).

1.1.16 - EXERCÍCIOS PROPOSTOS

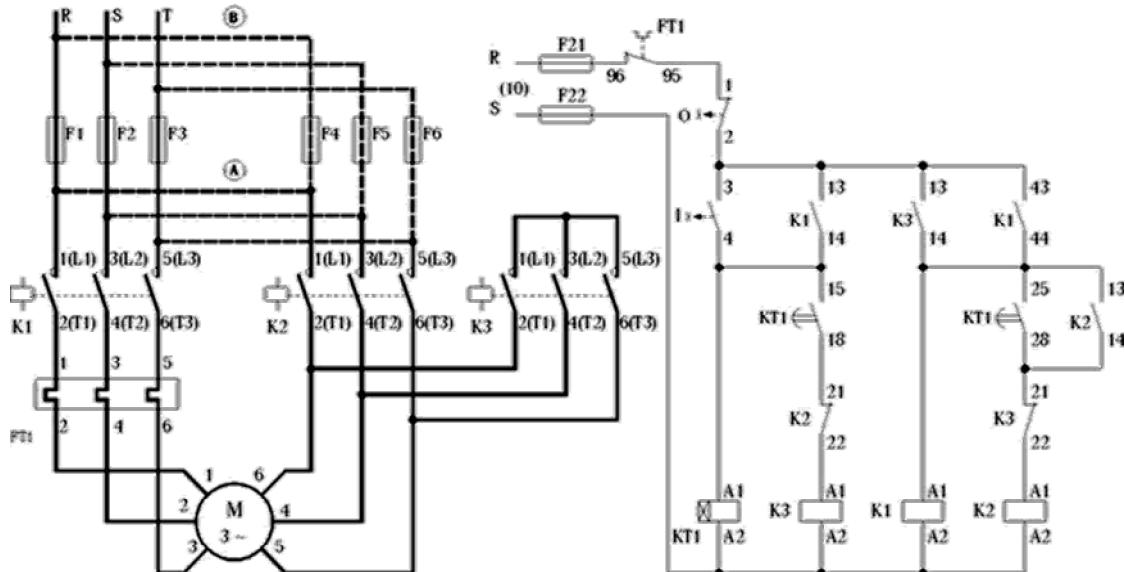
| | | |
|---|--------------------------------|---------------|
| Exercício no. 1 | Chave de partida direta | TPW-03 |
| Descriutivo : | | |
| <ul style="list-style-type: none"> - Ao ser pressionado o botão pulsador liga “I”; ativa a bobina do contator K1; - O motor é então acionado; - Ficando nesta condição até que seja pressionado o botão pulsador desliga “O” ou, ocorra a atuação do relé de sobrecarga; - Após o rearme do relé térmico a chave volta a condição inicial. <p>Condição inicial : botão pulsador I aberto; botão pulsador O fechado; FT1[97-98] aberto; FT1[95-96] fechado; bobina de K1 inativa (motor desligado).</p> | | |



| | | |
|------------------------|--|---------------|
| Exercício no. 2 | Chave de partida estrela-triângulo automática | TPW-03 |
| Descriptivo : | | |

- Ao ser pressionado o botão pulsador liga ‘T’;
- Ativa a bobina do relé temporizador Y-D, fechando o contato KT1[15-18];
- Ativando a bobina do contator K3;
- O contato K3[13-14] é então acionado, ativando a bobina do contator K1; (o contato K3[21-22] impede a possibilidade de ativação da bobina de K2 e o contato K1[43-44] permite ativação de K2 só depois que o contator K1 estiver ligado);
- Ficando nesta condição até que seja alcançado o tempo ajustado no relé temporizador Y-D (ajuste típico de 10 s);
- Quando então, o contato KT1[15-18] é novamente aberto; desativando a bobina do contator K3;
- 100 ms depois da abertura de KT1[15-18], o contato KT1[25-28] é fechado;
- Deste modo, caso o contato K3[21-22] esteja fechado e o contato K1[43-44] esteja fechado, a bobina do contator K2 é ativada;
- Ficando o motor ligado pelos contatores K1 e K2;
- Ficando nesta situação até que seja pressionado o botão pulsador desliga ‘O’ ou seja atuado o relé de sobrecarga;
- Após o rearme do relé térmico a chave voltará à sua condição inicial.

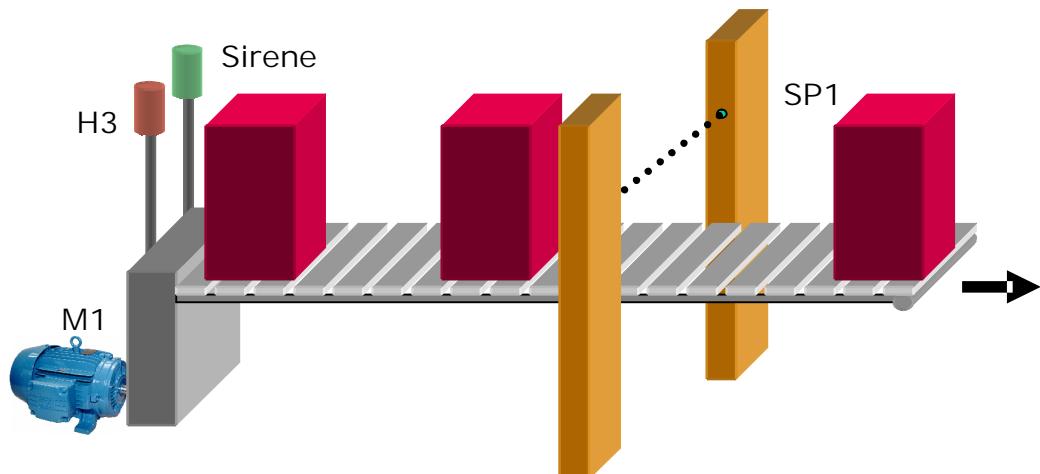
Condição inicial : botão pulsador I aberto; botão pulsador O fechado; FT1[97-98] aberto; bobina de K1 inativa, bobina de K2 inativa, bobina de K3 inativa, bobina do relé Y-Δ inativa (contatos KT1[15-18] e KT1[25-28] abertos).



| | | |
|------------------------|---------------------------|---------------|
| Exercício no. 3 | Contador de caixas | TPW-03 |
| Descriptivo : | | |

- Em uma esteira transportadora existe um sensor óptico (tipo barreira de luz) **SP1**;
- Este sensor indica a passagem de uma caixa do produto “X”;
- Todas as vezes que passarem por este sensor a quantidade de **15** caixas, a esteira deve ser parada e uma sirene ativada (avisando ao operador que um engradado foi completado);
- O operador retira manualmente o engradado completo e repõem um vazio para ser completado na próxima etapa;
- O processo é reiniciado através do operador pressionando o botão liga “I” (NA);
- **Obs.:** (1) A esteira é acionada pelo motor de indução trifásico **M1**, ligado através de uma chave de partida direta. (2) O processo deve ser interrompido caso seja desligado o motor, ou ocorra a atuação do relé de sobrecarga; (3) Caso ocorra a atuação do relé de sobrecarga, um sinalizador **H3** deve ser acionado de forma piscante (um pulso a cada 1 s); o valor da contagem nesta situação deverá ser armazenado, para que quando se reinicie o processo sejam contadas a quantidade desejada de caixas.

Condição inicial : O motor **M1** desligado; botão “I” não pressionado;



| Exercício no. 4 | Verificação do PH de uma mistura | TPW-03 |
|---|---|---------------|
| Desritivo : | | |
| <ul style="list-style-type: none"> - Em uma indústria química o PH de uma mistura deve ser mantido dentro de um valor específico; - Duas eletroválvulas controlam a entrada dos reagentes, EV1 para o reagente A e EV2 para o reagente B; - Para iniciar o processo é liberado o botão B1 (emergência – gira para soltar); - Então, a eletroválvula EV1 é acionada durante 30 s e a eletroválvula EV2 é acionada durante 45 s; - Também a eletroválvula EV3 (adição de água) é acionada, até que o sensor de nível SL1 seja acionado (indicando que o reservatório está cheio); - Assim que SL1 for ativado, a eletroválvula EV3 é desativada, e o motor M1 é acionado (agitador); - Após o acionamento do motor M1, deve-se aguardar 30 s para o início da monitoração do PH; - Para verificar esta condição um sensor SP1 (ver obs.) deverá ser lido; - Se a leitura do PH da mistura permanecer o equivalente a um valor entre 6,5 e 7,5 por mais de 180 s, o motor M1 é desligado e a eletroválvula EV4 é acionada, iniciando o escoamento do reservatório; - A eletroválvula EV4 ficará ativada até que o sensor de nível mínimo SL2 seja desativado; - Quando SL2 é desativado, a eletroválvula EV4 é desativada (terminando o escoamento); - Ocorre então, uma pausa de 10 s e o processo será reiniciado automaticamente, sendo interrompido apenas se novamente for pressionado o botão B1; - Obs.: O sensor de PH (SP1) é analógico, sinal elétrico 0 ... 10 Vcc proporcional à escala para PH de 0 ... 14; | | |

Condição inicial : reservatório vazio; todas as eletroválvulas desativadas; botão B1 ativado;

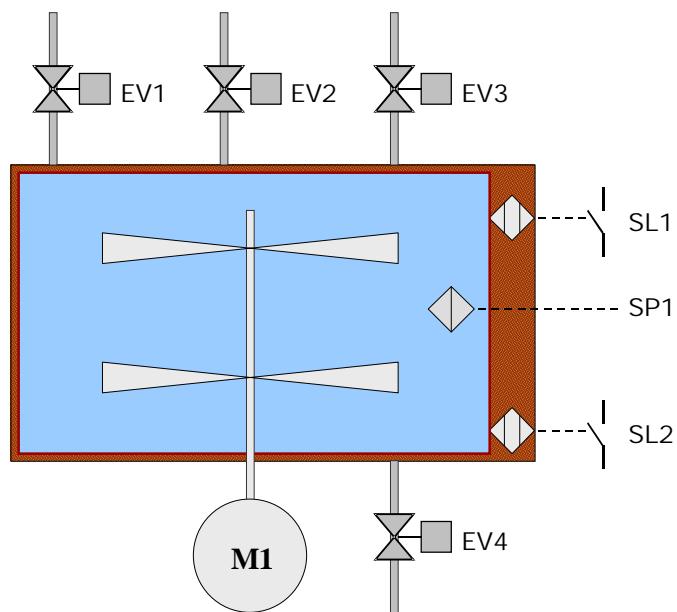
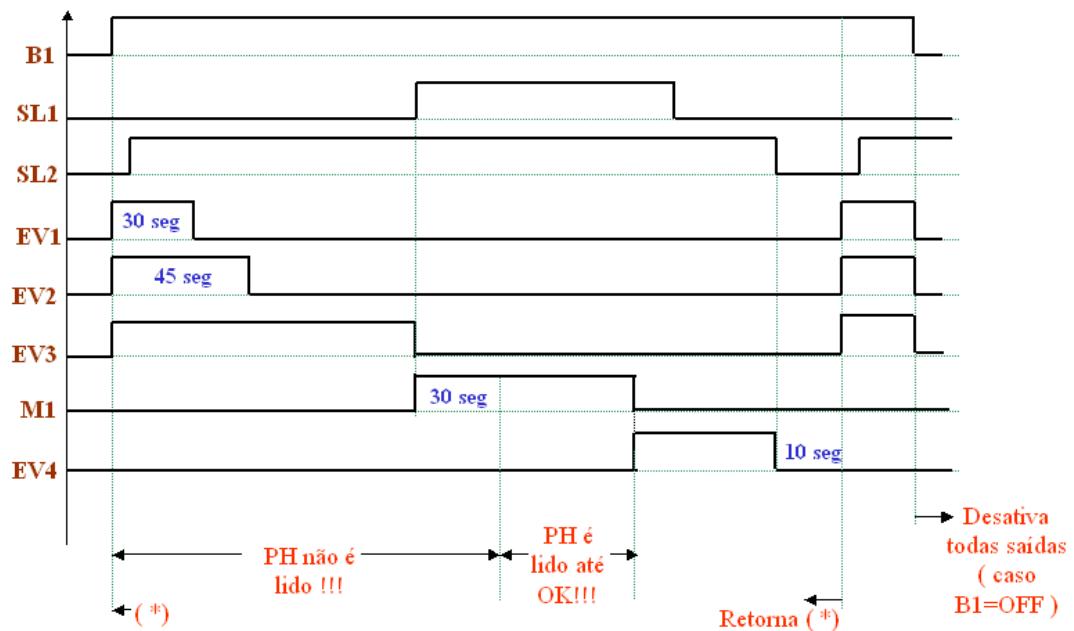


DIAGRAMA DE TEMPO



| Exercício no. 5 | Ajuste de velocidade através de saída analógica | TPW-03 |
|---|---|--------|
| <u>Descritivo :</u> | | |
| <ul style="list-style-type: none">- Considere um motor de 4 pólos (1800 RPM) acionado por um inversor de freqüência (CFW-XX);- Uma chave seletora Liga/Desliga S1 habilita o acionamento do motor.- O inversor está configurado para ter a referência de velocidade recebida através de uma de suas entradas analógicas (0 ... 10 Vcc);- Elaborar um aplicativo que permita ajustar a velocidade do motor, usando para isto um botão pulsador B1 (NA) para aumentar a velocidade e um pulsador B2 (NA) para diminuir a velocidade;- A velocidade do motor deverá variar de 10% a 100% de sua velocidade nominal; | | |

Condição inicial : Nenhum botão pressionado; motor desligado.

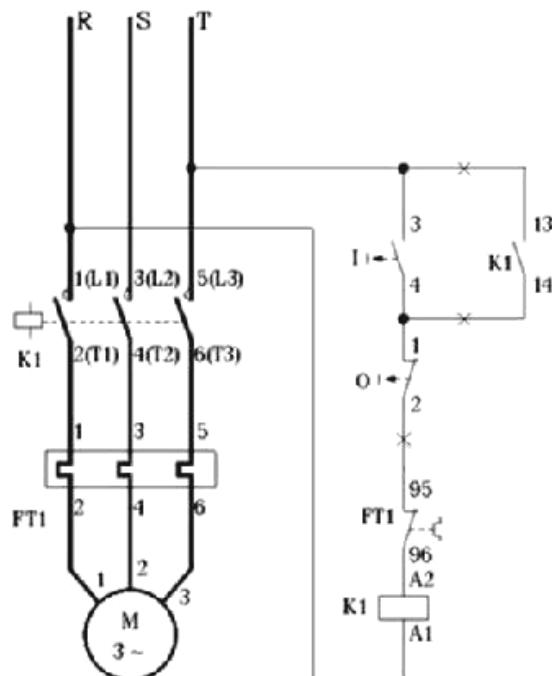
| | | |
|------------------------|---|-------------------|
| Exercício no. 6 | Chave de partida estrela-triângulo com IHM | TPW-03+PWS |
|------------------------|---|-------------------|

Descriptivo :

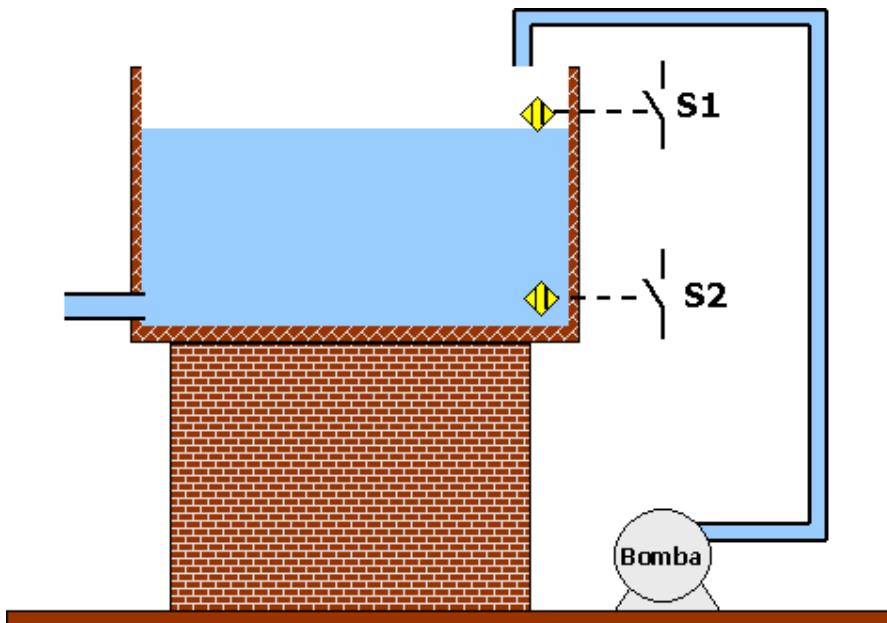
- Implemente um programa para partida estrela-triângulo;
- Acrescente as estruturas para indicar no display da PWS as seguintes telas:
- Ao energizar o TPW-03 mostrar uma tela inicial contendo:
 - Um único botão para ligar/desligar;
 - Campo para ajuste do tempo de partida;
 - Campo para visualização do tempo corrente.
- Deverá aparecer na IHM a condição de operação do motor, informando se está em partida ou em regime. Nestas telas, deverá ter a opção de retornar para a tela inicial.
- Indicar na IHM a atuação do FT1 caso este evento ocorra. Neste caso, a tela deverá permanecer até rearmar o relé, quando então o display deverá voltar para a tela inicial.

Condição inicial: nenhum botão pressionado; todas as bobinas dos contatores desativadas.

| | | |
|--|-----------------------|-------------|
| Exercício no. 1 | Partida direta | CLIC |
| Descriptivo : | | |
| <ul style="list-style-type: none"> - Considere a figura abaixo e elabore um aplicativo que reproduza o comando elétrico representado; | | |

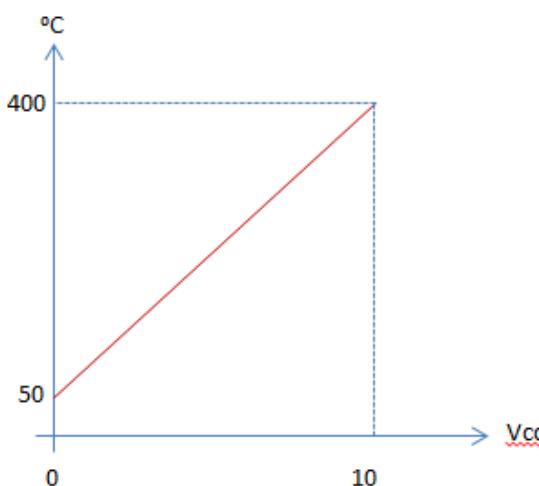


| | | |
|--|----------------------------------|-------------|
| Exercício no. 2 | Controle de nível (ON/FF) | CLIC |
| Descriptivo : | | |
| <ul style="list-style-type: none"> - Deseja-se controlar o nível de uma caixa d'água entre um valor máximo e mínimo; - Existem para isto dois sensores de nível, respectivamente, S1 (NA) (nível máximo) e S2 (NA) (nível mínimo); - Para enchermos esta caixa, usamos uma bomba centrífuga que será ligada ou desligada pelo CLIC® em função do nível da caixa; - Se o sensor S2 estiver aberto, a bomba é ligada; - Permanecendo assim até que o sensor S1 seja ativado; - Quando S1 é ativado, a bomba é desligada; - Permanecendo assim até que o sensor S2 abra novamente; - Este controle automático pode ser desligado manualmente por um interruptor L1. | | |



| Exercício no. 3 | Sistema de irrigação | CLIC |
|--|----------------------|------|
| <u>Descritivo :</u> | | |
| <ul style="list-style-type: none">- Temos que controlar a irrigação de uma determinada cultura, através de uma eletroválvula EV1;- Esta cultura deverá receber irrigação todos os dias, às 8:30 horas, durante 5 minutos, e às 17:30 horas, durante 10 minutos;- Ao pressionar um botão pulsador B1 (NA), a eletroválvula deverá ser acionada durante 15 minutos, em qualquer momento do dia;- Caso seja necessário, todo o sistema deverá ser desativado por uma chave seletora S0. | | |

| | | |
|---|--------------------------------|-------------|
| Exercício no. 4 | Acionamento de cilindro | CLIC |
| <u>Descriptivo :</u> | | |
| <p>- Implementar um aplicativo para acionar um cilindro de uma máquina ao pressionar um botão pulsador I. O tempo de atuação do cilindro deverá ser ajustado na tela inicial da IHM. Uma segunda tela indicará que o cilindro está acionado e o tempo corrente.</p> | | |

| Exercício no. 5 | Indicação de temperatura na IHM | CLIC | | | | | | | | |
|---|---------------------------------|------|---------|------------------|---|----|---|-----|----|-----|
| <u>Descritivo :</u> | | | | | | | | | | |
| <p>- Um sensor de temperatura envia um sinal (0-10Vcc) à entrada analógica 2 do CLIC, realizando uma leitura que varia de 50 a 400°C. Montar uma tela na IHM do CLIC para monitoração desta grandeza.</p>  <table border="1"><caption>Data points from the graph</caption><thead><tr><th>Vcc (V)</th><th>Temperature (°C)</th></tr></thead><tbody><tr><td>0</td><td>50</td></tr><tr><td>5</td><td>250</td></tr><tr><td>10</td><td>400</td></tr></tbody></table> | | | Vcc (V) | Temperature (°C) | 0 | 50 | 5 | 250 | 10 | 400 |
| Vcc (V) | Temperature (°C) | | | | | | | | | |
| 0 | 50 | | | | | | | | | |
| 5 | 250 | | | | | | | | | |
| 10 | 400 | | | | | | | | | |

| Exercício no. 6 | Comando para um misturador | CLIC |
|---|----------------------------|------|
| Descriptivo : | | |
| <ul style="list-style-type: none"> - Considere dois reservatórios, conforme a figura abaixo, contendo dois tipos diferentes de ingredientes, que deverão ser misturados a fim produzir uma massa. - A quantidade de material de cada ingrediente deverá ser ajustado na IHM em kg antes de iniciar o processo. - As balanças B1 e B2 fornecem um sinal de 0 a 10Vcc proporcional a um massa de 0 a 50kg. - Ao ser pressionado um botão liga “I” (pulsador NA), se os sensores de nível SL1 (NA) e SL2 (NA) estiverem fechados, as válvulas EV1 e EV2 deverão ser ativadas; - As válvulas EV1 e EV2 devem permanecer acionadas até que as respectivas balanças B1 e B2 atinjam as massas desejadas; - Assim que as duas válvulas EV1 e EV2 forem desativadas, ligam as válvulas EV3 e EV4 e o motor M1; - As válvulas EV3 e EV4 devem ser ativadas durante 15 s, e o motor M1 (misturador) deve ser acionado durante 2 min; - Quando o motor M1 desligar, a válvula EV5 é ativada por 15 s, terminando o ciclo de operação; - Este ciclo de operação será repetido somente se pressionando o botão “I” novamente, enquanto não faltar nenhum dos dois ingredientes. - Uma vez iniciado o processo, não poderá ser interrompido antes que um ciclo completo de operação seja executado. | | |

