

SCADA

É preciso contar novidades. Sempre. Senão ficamos repetitivos e ninguém nos convida nem mesmo para um velório mais divertido. Desses com mulher e amante
(Célius Aulicus - JB - 10/01/92).

Definições:

Scada = *Supervisory Control and Data Acquisition*

São os sistemas de supervisão de processos industriais que coletam dados do processo através de remotas industriais, principalmente Controladores Lógico Programáveis, formatam estes dados, e os apresenta ao operador em uma multiplicidade de formas. O objetivo principal dos sistemas SCADA é propiciar uma interface de alto nível do operador com o processo informando-o "em tempo real" de todos os eventos de importância da planta.

Hoje os sistemas de supervisão oferecem três funções básicas:

- a) Funções de supervisão:
Inclui todos as funções de monitoramento do processo tais como: sinóticos animados, gráficos de tendência de variáveis analógicas e digitais, relatórios em vídeo e impressos, etc.
- b) Funções de operação:
Atualmente os sistemas SCADA substituíram com vantagens as funções da mesa de controle. As funções de operação incluem: ligar e desligar equipamentos e seqüência de equipamentos, operação de malhas PID, mudança de modo de operação de equipamentos, etc.
- c) Funções de controle:
Controle DDC ("Digital Direct Control")
Alguns sistemas de supervisão possuem uma linguagem que permite definir diretamente ações de controle, sem depender de um nível intermediário de controle representado por remotas inteligentes. Todas as operações de entrada e saída são executadas diretamente através de cartões de I/O ligados diretamente ao barramento do micro, ou por remotas mais simples. Os dados são amostrados, um algoritmo de controle como um controlador PID por exemplo, é executado, e a saída é aplicada ao processo (ação direta sobre uma variável manipulada). Isto entretanto só é possível quando a velocidade do processo assim o permite. Em alguns casos requisitos de confiabilidade tornam desaconselhável este tipo de solução.

Controle Supervisório:

Nesta classe de sistemas, os algoritmos de controle são executados pela unidade terminal remota (UTR), mas os set-points para as malhas de controle são calculados dinamicamente pelo sistema de supervisão de acordo com o comportamento global do processo. Esta arquitetura possui maior confiabilidade que os sistemas DDC e traz a vantagem de atuar sobre um grande número de malhas de controle simultaneamente enquanto o operador geralmente só consegue atuar malha a malha com um sistema convencional. Geralmente é utilizada uma interface tipo sistema especialista para definição

das regras de controle a nível de supervisão. Este tipo de estratégia é muito utilizado para controle avançado na área mineral onde é comum o modelamento matemático da planta.

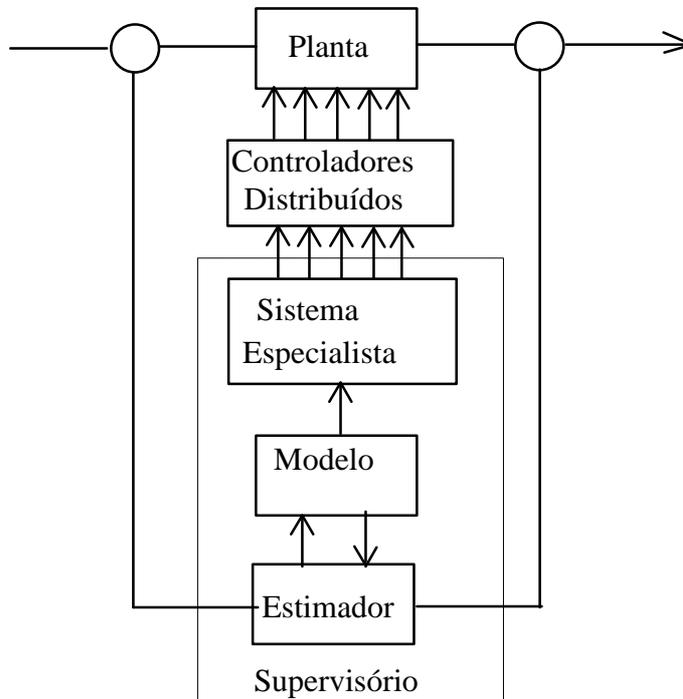


Figura 1: Controle supervísório

Os objetos de um supervísório:

Cada sistema SCADA irá gerenciar um certo número de objetos também denominados entidades que descrevem as variáveis de processo controladas e os elementos habituais de uma mesa de controle. Embora este número varie de um sistema para outro, existe um consenso em relação às suas características básicas.

A atividade de configuração de um supervísório compreende geralmente duas etapas:

- Definir cada variável de processo na base de dados
- Definir sinóticos, gráficos e relatórios.

Em todo o sistema existem basicamente duas variáveis simples ou primitivas e alguma variáveis compostas formadas a partir das primeiras.

Variáveis simples:

Variável analógica:

Descreve uma variável analógica de entrada. O CLP já converte esta leitura para um valor digital quantizado em 12 bits (0 ..4095).

Principais atributos:

- Tag: conjunto de até 12 caracteres descrevendo o nome da variável
- Texto: Descrição sucinta da variável
- Unidade de Engenharia: unidade de medida, por exemplo: ton./h, °C, metros, etc.
- Valor: último valor lido da variável
- Lim inferior: valor em UEng ser atribuído ao valor 0% da variável.
- Lim superior: valor em UEng a ser atribuído ao valor 100% da variável
- Limite HH: valor em UEng para alarme Muito Alto
- Limite H: valor em UEng para alarme Alto
- Limite L: valor em UEng para alarme Baixo
- Limite LL: valor em UEng para alarme Muito Baixo
- Histerese: valor percentual de histerese para geração de alarmes. Serve para evitar múltiplos alarmes quando o valor da variável cruza uma das faixas definidas pelos valores limites.
- Filtro: Somente quando o valor da variável variar mais que o limite especificado (filtro) é que o valor em tela será atualizado.
- Endereço: Corresponde ao endereço da variável no CLP.

Os valores limites definem 5 faixas de operação para a variável. Quando a variável cruza uma destas fronteiras um alarme é dado para o operador. Em geral os valores da variável na tela dependem das cores atribuídas a cada faixa. Estas cores em geral são as mesmas para todas as variáveis.

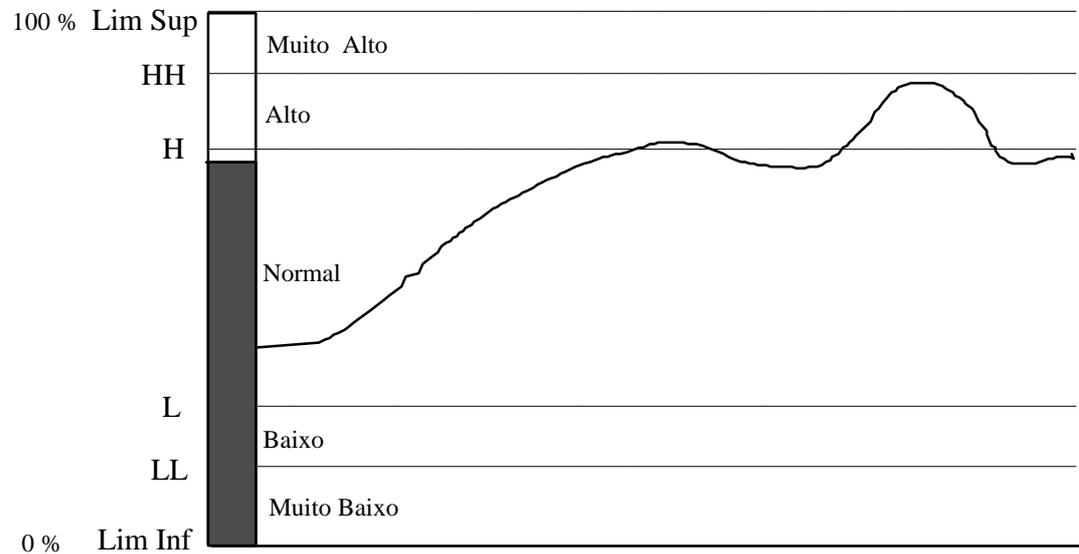


Figura 2: Faixas de uma variável analógica

Figura 3: Definição de parâmetros de variáveis analógicas

Variável Discreta:

Também chamada de variável digital. Corresponde a 1 bit de informação que pode assumir os valores 0 ou 1.

Principais atributos:

- Tag: nome da variável
- Texto: Descrição sucinta da variável
- Descrição do estado: texto a ser atribuído aos estados 0 e 1 da variável. Pode corresponder a 2 strings ou a um índice para uma tabela que contém os textos a serem associados a cada tipo de variável, por exemplo: Aberto/Fechado para válvulas, Ligado/Desligado para motores etc,
- Endereço: endereço da variável na memória do CLP na forma: CP:PAL:BIT

- Tipo: Diz se a variável desempenha alguma função importante como por exemplo: bit de Funcionando, Pronto para Partir ou Defeito de algum equipamento.
- Valor: valor corrente do ponto
- Alarme: Determina a classificação do alarme associado ao ponto: Crítico, Urgente, Alarme, Evento, Sem alarme. O sistema também define que tipo de ação será realizada para cada classe de alarme.

O valor normal para uma variável discreta é sempre convencionado para todas as variáveis. em conjunto para maior facilidade de interpretação por parte do operador e pessoal da manutenção.

Em alguns sistemas é possível definir agregados de variáveis discretas (*bit strings*), definir cores e textos para cada estado (2^n), etc.

Variáveis compostas:

Enquanto alguns sistemas possuem apenas um pequeno número de primitivas, outros possuem tipos mais complexos formados pelo agregado de várias primitivas fundamentais.

De uma forma ou de outra cada sistema sempre possui uma forma de representar todas as entidades que correspondem às entidades controladas no processo ou aos instrumentos da mesa de comando convencional:

T o t a l i z a d o r :

Totaliza uma variável analógica diretamente ou reproduz um valor totalizado pelo CLP. O input da totalização tanto pode ser um valor analógico (fluxo ou vazão) ou pulsos associados a uma variável discreta (totalizador de pulsos). Neste último caso, a cada pulso, um incremento pré determinado é aplicado ao valor totalizado. O operador pode a qualquer tempo zerar o valor totalizado. Assim é conveniente definir três atributos básicos:

- Valor totalizado sem reset: não pode ser zerado pelo operador.
- Valor totalizado com reset: pode ser zerado pelo operador ao final de um turno, dia, mês, etc
- Data de reset: armazena a data e hora da última ação de reset.

C o n t r o l a d o r P I D :

Descrevem um controlador PID independente de sua encarnação física. Um sistema supervisório moderno é capaz de comandar dezenas de malhas de controle que na verdade são implementadas no nível inferior de controle representado pelos CLP, Single loops e Multi loops.

Os parâmetros de configuração definidos são enviados para a memória do controlador. O operador pode então realizar a sintonia da malha durante o *start-up* do sistema, guiado por uma tela de tendência que plota os valores de variável medida, set-point e variável manipulada em função do tempo (*tuning*) ou usando uma ferramenta qualquer de sintonia.

A partir daí as únicas operações possíveis são mudar o modo de operação da malha ou o seu set-point.

Atributos típicos:

- Tag: identificação da malha de controle
- Texto: descrição da malha
- Variável de entrada: identificação da variável de entrada (medida) da malha.
- Constantes: K_c , T_i , T_d para sintonia da malha
- Modo: modo de operação da malha PID:
 - ❑ Modo Manual: O operador age diretamente sobre a variável de saída do bloco PID (variável manipulada)
 - ❑ Modo Automático: A ação se dá sobre o set-point da malha.
 - Modo Automático Local: O próprio operador define o valor do set-point.
 - Modo Automático Remoto: O set-point é definido externamente:
 - Pela variável de saída de outra malha em cascata (malha mestre)
 - Por um valor decorrente de algoritmo de otimização executado pelo supervisor ou CLP.

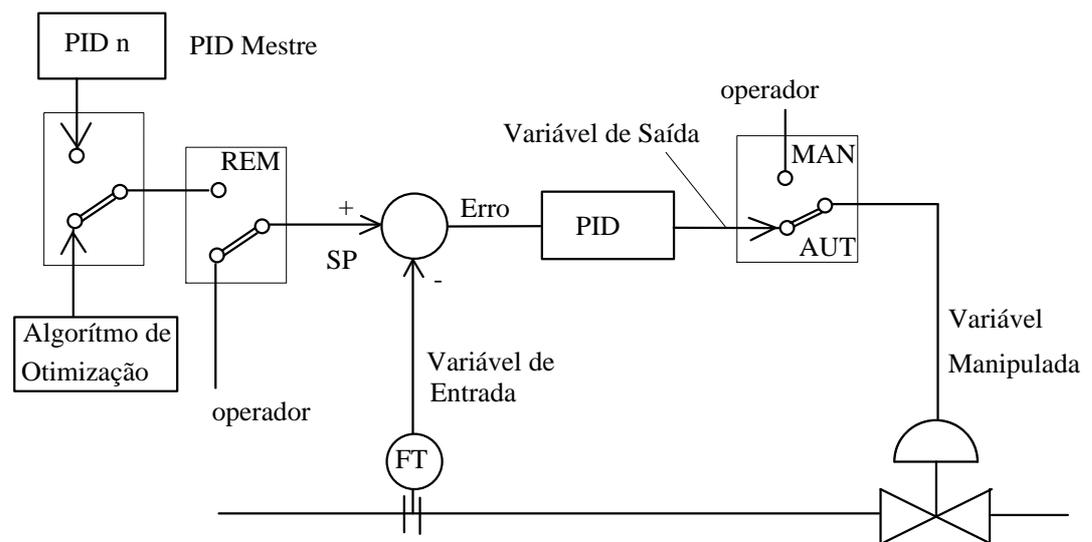


Figura 4: Diagrama de malha PID típica

- Identificação da Malha Mestre (quando aplicar)
- Endereço: Endereço base do bloco de parâmetros PID na memória do CLP
- Identificação do operador que definiu ou sintonizou a malha
- Banda morta
- Limite máximo e mínimo da variável de saída
- Polarização (bias)
- Etc.

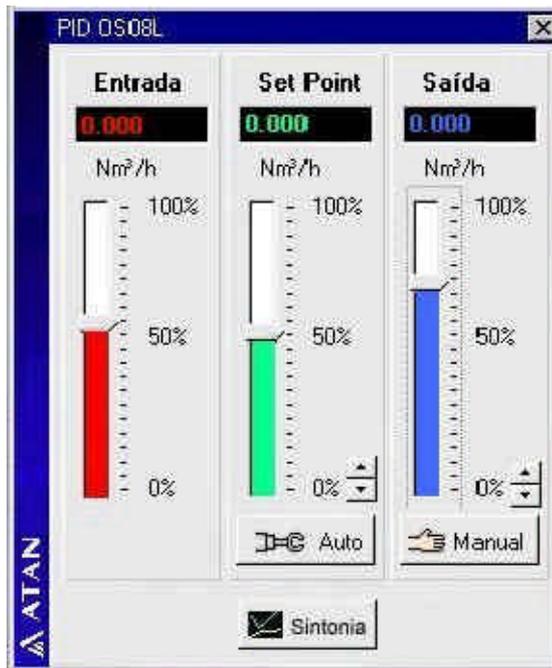


Figura 5: Janela de malha PID.

Equipamento:

Corresponde a um equipamento de processo qualquer: motor, classificador espiral, transportador de correia, reator, etc.

Seus principais atributos são:

- Tag: Identificação do equipamento
- Texto: Descrição do equipamento
- Bits de status:

Cada equipamento possui uma série de bits típicos de entrada para o supervisor, que em suma são função da filosofia de programação de funcionamento destes equipamentos a nível do CLP, por exemplo:

FUN - bit que avisa ao supervisor que o equipamento está funcionando

PPP - bit que sinaliza que o equipamento está pronto para partir (pronto para operar)

DEF - Bit que diz que o equipamento está com defeito (ou lógico de todos os bits de defeito do equipamento)

Além destes bits hoje é normal enviar para o supervisor todos os bits de defeito que discriminam qual o problema ocorrido. Estes bits são utilizados para alarme, diagnóstico, e para guias de operação no caso dos sistemas inteligentes.

MODO: um, dois ou três bits que discriminam o modo de operação do equipamento:

LOCAL/REMOTO: diz se o comando está sendo dado pela central de controle ou pelo painel de comando local do equipamento.

OPERAÇÃO/MANUTENÇÃO: diz se o equipamento está sendo comandado pela seqüência a qual pertence ou isoladamente para manutenção, ETC

- Bits de comando:
 - São bits de saída do supervisor para o equipamento permitindo realizar ações de comando:
 - LIGA: comando de liga para o equipamento.
 - DESLIGA: comando de desliga
 - MODO: definição de um novo modo para o equipamento.
- Horímetro: Fornece o total de horas de operação do equipamento
- Etc



Figura 6: Janela de equipamento com diagnósticos exibidos

Seqüência ou Grupo:

É um conjunto de equipamentos intertravados que pode ser tratado do ponto de vista lógico como uma única entidade. Cada seqüência é composta tipicamente por 8 ou mais equipamentos. Por exemplo: o conjunto de motores e outros subsistemas que compõem um moinho de bolas pode ser tratado de forma unificada pelo sistema, já que todos os equipamentos que o compõem partem e operam em conjunto de forma intertravada. Um equipamento pode pertencer a diversas seqüências diferentes. Por exemplo, um transportador de correia do pátio de um porto pode estar conectado a diversas rotas de transferência de minério de acordo com o tipo de minério sendo carregado, pier de embarque, etc.

Atributos típicos:

- Tag: Identificação da seqüência
- Texto: Descrição da seqüência
- Equipamentos: Relação dos equipamentos que compõem a seqüência.
- Bits de status: semelhantes aos bits de status de um equipamento:

Funcionando, Pronto para Partir, Defeito

- Modos de Operação:

Basicamente a seqüência pode operar em Remoto, onde todos os equipamento estão intertravados e são operados pelo supervisor, ou em Local onde cada equipamento é operado independentemente.. A partida se inicia pelo equipamento cabeça da seqüência,

- Bits de comando:

Servem para ligar desligar ou mudar o modo de operação de todos os equipamentos de uma seqüência.

Cálculo:

Nem todas as variáveis de interesse podem ser amostradas diretamente do processo. Às vezes estamos interessados no comportamento de uma variável que não é medida, mas que pode ser inferida a partir de diversas outras variáveis. Por exemplo calcular a potência ativa a partir dos valores de tensão, corrente e fator de potência.

Cálculos como estes em geral são definidos a partir de uma linguagem procedimental que deve conter no mínimo:

- Funções de acesso a valores de variáveis do processo.
- Funções aritméticas fundamentais: soma, subtração, multiplicação, divisão, mod.
- Funções booleanas: and, or, xor e not
- Funções transcendentais: trigonométricas, exponenciais, logarítmicas, radiciação, etc
- Instruções condicionais: if ...then... else...
- Lógica de parênteses
- Funções de manipulação de texto

O tipo retornado de um cálculo geralmente é um valor real, um valor booleano ou um string.

A chamada de um cálculo que envolve um outro cálculo deve referenciar sempre ao último valor calculado para se evitar chamadas recursivas infinitas.

A variável calculada, deve se comportar como um ponto qualquer amostrado diretamente do processo e permitir:

- Gerar valores instantâneos para exibição em sinóticos segundo diversas formas.
- Gerar valores para base de dados histórica.

Processadores de expressões também são úteis para diversas finalidades dentro de um sistemas de supervisão como por exemplo para criar campos virtuais em planilhas e geradores de relatórios.

Outras entidades podem ser definidas dependendo do tipo de processo como Chaves de Seleção, Chaves de Habilitação, Tanques, Constantes Paramétricas, Rotas, etc.

Principais funções de um supervisor:

Sinóticos:

Fornecem uma representação gráfica geral da planta em substituição aos painéis sinóticos tradicionais. Cada sinótico representa uma área do processo em um certo nível de detalhe. Para se obter uma visão mais detalhada de uma

determinada área pode-se recorrer a um novo sinótico, a um sinótico de hierarquia inferior (sub-sinótico), ou a uma visão de uma outra camada do mesmo sinótico (sistema "multi layer").

Para alguns tipos de processo, recomenda-se o uso de um sinótico tipo plano infinito que traz a representação global de uma sistema distribuído geograficamente, tal como um oleoduto, o sistema de controle de tráfego de uma cidade, um sistema de controle de subestações de trolebus, etc. Esta técnica é denominada *full-graphics* e requer um poder computacional muito maior por parte das estações de trabalho, para efetuar operações de *panning*, *zoom-in* e *zoom-out* em tempo real.

As telas possuem uma parte fixa denominada de máscara ou fundo e diversos campos dinâmico atualizados dinamicamente. Os primeiros sistemas supervisórios possuíam interface com o usuário semi gráfica. Todos os espaços vagos da tabela do gerador de caracteres eram preenchidos com símbolos especiais que permitiam representar os equipamentos de processo. Tanto as EPROMs da console do computador quanto dos terminais precisavam ser substituídas. Um conjunto de caracteres típico é mostrado no apêndice 1.

O sinótico era formado pela justaposição dos caracteres gráficos, como num quebra-cabeças. Os símbolos utilizados para um determinado tipo de processo: petroquímico, por exemplo, não serviam para representar um processo mineral ou de outra natureza.

No sistema gráfico ao contrário, o desenho é formado livremente pela combinação de entidades geométricas fundamentais como retas, retângulos, elipses e círculos, texto bitmapeado e vetorados ("stroke-fonts"), arcos, splines, curvas de bézier, etc.

Após definidos, os símbolos são armazenados numa biblioteca. Se a representação armazenada corresponde a descrição das entidades geométricas temos um sinótico orientado para geometria. Se o símbolo armazenado corresponde a uma configuração fixa de bits (mapa de bits), temos um editor *bitmapeado*.

O construtor de sinóticos é a ferramenta que permite ao usuário criar novos sinóticos. De preferência deve ser possível se usar o construtor com o sistema *on-line*. Alguns construtores são editores gráficos que definem duas estruturas de dados básicas: uma para a máscara e outra para os campos dinâmicos. Alguns construtores entretanto necessitam compilar a descrição de campos para obter um código executável para as animações.

Sinóticos com estrutura de dados geométrica (modelados) são mais flexíveis para modificações futuras e parecem ser uma tendência para sistemas SCADA atuais. Sinóticos bitmapeados são mais rápidos de se exibir e permitem definir um maior nível de detalhe para cada símbolo.

Atualmente os editores orientados para objetos são ainda mais flexíveis. Cada equipamento corresponde a um objeto. Os objetos podem ser transformados por translação, rotação e mudança de escala, podem ter qualquer uma de suas propriedades (atributos) alterada e ter ações complexas a ele associadas (*click actions*). O sistema se torna totalmente dirigido a eventos: eventos de processo e eventos de operação.



Figura 7: Sinóptico projetado com canhão.

Os elementos dinâmicos:

Exibem os status de variáveis do processo dinamicamente.

Representação de variável analógica:

O valor de uma variável analógica pode ter várias formas de representação:

a) Texto:

Exibe valor de engenharia da variável analógica. A cor do texto pode servir para codificar o status da variável: Muito Baixo, Baixo, Normal, Alto, Muito Alto.

- b) Barras horizontais e verticais:
Fornecem uma representação percentual do valor da variável. Podem ser utilizados para mostrar o enchimento de um silo, tanque, reator, etc.
- c) Deslocamento vertical, horizontal:
Realiza operação de translação de um objeto na tela em função do valor de uma variável de processo. Os valores de 0 a 100% são associados às posições dos extremos do percurso e o software realiza interpolações para valores intermediários.
- d) Rotação:
Efetua a rotação de um objeto: forno rotativo de cimento, pás de um ventilador, etc associando 0° ao valor 0% da variável e 360° ao valor de 100%.
- e) Tendência:
Exibe o gráfico dos últimos valores da variável em função do tempo.
- f) Mostradores Circulares:
São os Dials e Gauges que simulam os mostradores circulares convencionais.

Representação de uma variável de estado:

- a) Texto
Exibe o status da variável:
ABERTO/FECHADO
LOCAL/REMOTO
LIGADO/DESLIGADO
Para cada estado é possível definir a cor de apresentação do texto.
- b) Associação a cor (ou outro atributo) de um objeto:
A cor do objeto muda de acordo com o status da variável associada.

Na maior parte dos sistemas é desejável que a cor de um objeto seja função não apenas do estado de uma variável isolada, mas de um conjunto de variáveis:

Por exemplo: o estado de funcionamento de um equipamento pode estar codificado em dois bits: LOCAL / REMOTO / OPERAÇÃO / MANUTENÇÃO.
- c) Associação a um par de objetos complementares

Os dois objetos ocupam fisicamente a mesma posição no sinótico. Quando a variável está em 0 o objeto chave aberta, por exemplo, é exibido, quando está em 1 a chave é mostrada na posição fechada.

Os objetos podem ser modelados ou bitmapeados.

Para outras entidades o desenvolvimento de formas de representação é semelhante.

Indique as formas de representação que você julgaria importantes para as seguintes entidades compostas:

- a) Malha PID

- b) Totalizador

- c) Equipamento

- d) Seqüência

- e) Tanque

Objetos ativos:

Como o mouse hoje se tornou o principal dispositivo de seleção de opções de entrada para sistemas SCADA, a ação de se clicar um objeto é hoje a atitude default para se deflagrar qualquer tipo de ação referente ao objeto (equipamento, *faceplate* de controlador PID, etc).

Assim devemos clicar um objeto para:

- a) Alterar qualquer atributo do objeto, por exemplo: ligar/desligar, alterar modo de operação, forçar valor, inserir mensagem ao operador, etc.
- b) Solicitar ao sistema mais informações sobre o objeto, por exemplo: relatório de ponto, *help-on-line*, diagnóstico, guia operacional, etc.
- c) Exibir ou reconhecer alarmes do objeto
- d) etc.

Um dos objetos mais úteis em um sinótico são os *hot-links* que correspondem a textos ou figuras que quando clicadas chamam uma nova tela de sinótico permitindo a criação de acessos diretos, rápidos e complexos no estilo hipertexto.

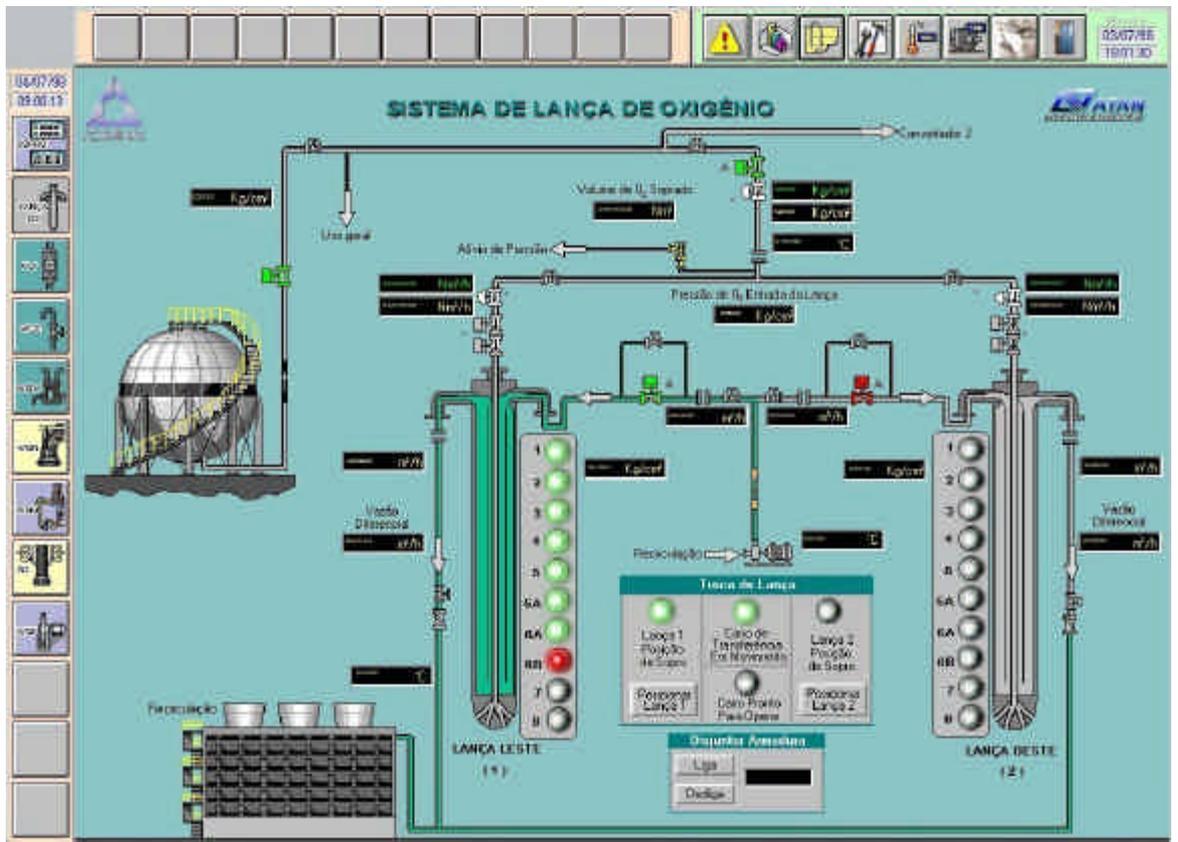
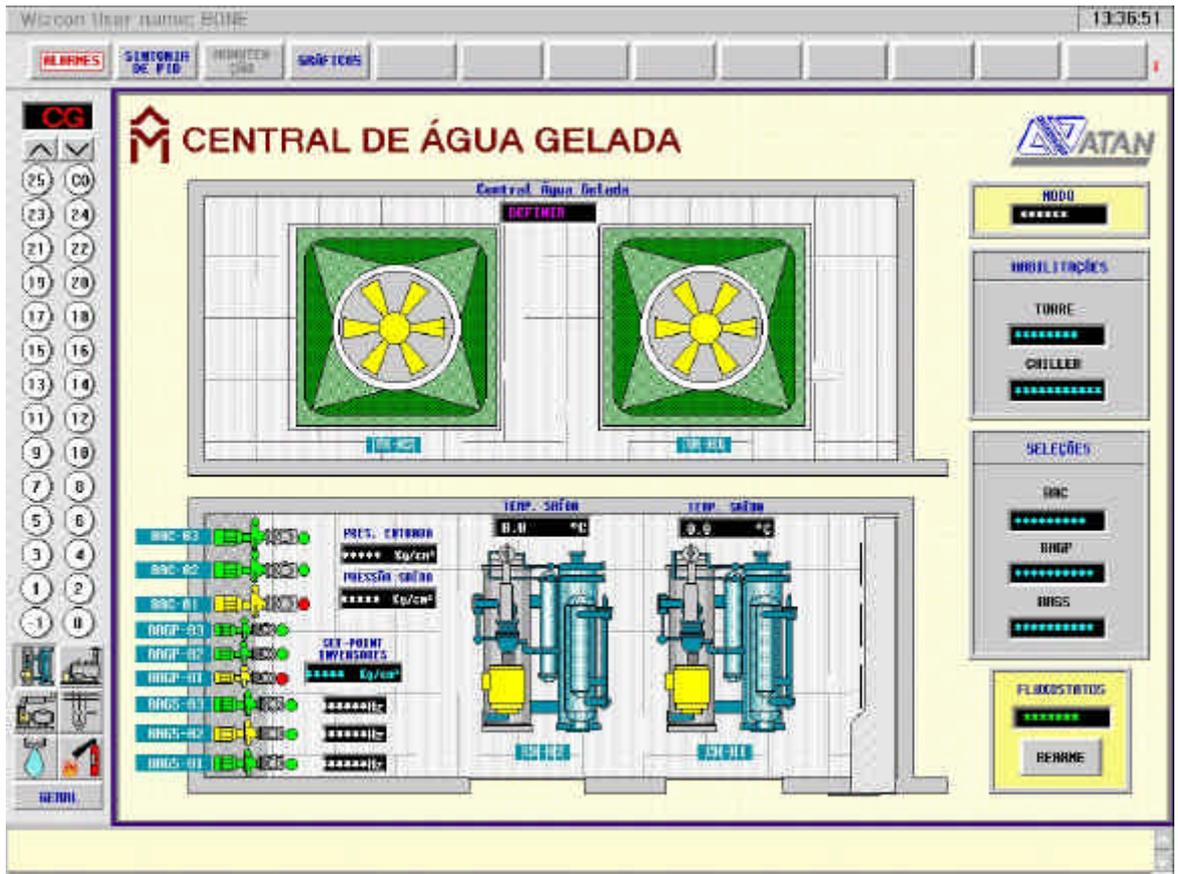


Figura 8: Sinótipos típicos

Operação

Para operar um equipamento devemos selecionar com o mouse um equipamento no sinóptico e solicitar a abertura da janela de operação. Esta janela nos permite realizar as operações básicas com um equipamento: ligar, desligar, solicitar um diagnóstico, rearmar defeitos, etc.



Figura 9: Janela de operação de equipamento

Hardcopy:

Realiza o *hardcopy* do sinóptico corrente.

Gerenciador de alarmes

Este módulo está presente em todos os sistemas SCADA. Ele recebe os eventos do processo e os registra identificando:

- data e hora do evento
- variável alarmada
- valor no momento do alarme
- descrição do evento
- data e hora de normalização do evento
- status do evento: alarmado, normalizado, reconhecido pelo operador

Os eventos são armazenados em um *buffer* circular em disco ou memória que armazena os últimos N eventos correspondendo geralmente a um turno de operação. Após este prazo, o arquivo pode ser salvo em disco para análise histórica, transferido para outro computador de maior capacidade de armazenamento ou simplesmente descartado.

A interface com operador se dá geralmente de duas formas:

Uma janela de alarmes exibe os x alarmes mais recentes (x geralmente varia de 2 a 4)

O operador pode solicitar a exibição de uma página de alarmes contendo cerca de 20 alarmes mais recentes e depois pagnar para frente e para trás até encontrar o evento de interesse.

Quando um alarme ocorre o operador é avisado através de uma buzina, música ou por um *speech maker*. O operador deve declarar que está ciente do problema, reconhecendo o alarme mais recente ou todos os alarmes simultaneamente. O alarme mais recente é então substituído na tela por outro alarme que ainda não tenha tido a oportunidade de se manifestar.

O status de um alarme pode ser convencionado como a seguir:

Cor da mensagem	Significado
Vermelho piscante	Alarme não reconhecido
Vermelho	Alarme já reconhecido
Amarelo	Normalizado e não reconhecido
Verde	Normalizado e reconhecido

Exemplo de linhas de alarme:

12/02 14:00	14:03	12-TC-34	Correia transportadora Desalinhada
12/02 14:05	xx:xx	12-BP-05 50°C	Temperatura do motor da bomba Muito Alta

Geralmente é possível imprimir o relatório de alarmes classificado por um campo como por exemplo, data e hora (alarmes mais recentes ou que ocorreram em um determinado intervalo), nome de variável, etc.

O campo de data e hora de normalização traz dificuldades para o projetista, já que uma variável pode ter seu estado normalizado diversas horas após ter entrado em alarme, o que forçará a busca do evento em arquivos já fechados.

Registrador de tendência

Plota um gráfico de tendência de uma dada variável, geralmente analógica, em função do tempo. Um gráfico de tendência pode ter de 1 a 8 penas.

O usuário deverá definir duas coisas:

- Quais variáveis devem ser amostradas pelo sistema sincronamente com um relógio e com qual frequência.
- Para cada gráfico, quais das variáveis registradas devem ser exibidas simultaneamente em uma mesma tela

Os períodos de amostragem que variam tipicamente de 100 ms a 1 hora devem ser escolhidos de acordo com a velocidade real do processo. É normal escolher um período para cada tipo de variável (temperatura, nível, pressão, etc).

Os dados são geralmente armazenados em um *buffer* circular de tamanho B bytes. O período total de armazenagem de dados irá depender do período de amostragem e de B.

Exemplo:

$B = 64 \text{ kbytes} = 64 * 1024 \text{ bytes} = 65536$

Tamanho do Registro: dado: double <----> 8 bytes / registro

Número total de registros = $65536/8 = 8192$ registros

Período de amostragem = 2 segundos

Tempo total armazenado = $8192 * 2 = 16384 \text{ s} = 273 \text{ min} = 4,5 \text{ h}$

$$T = P * B/R \text{ (s)}$$



Figura 10: Configuração do gráfico de tendência

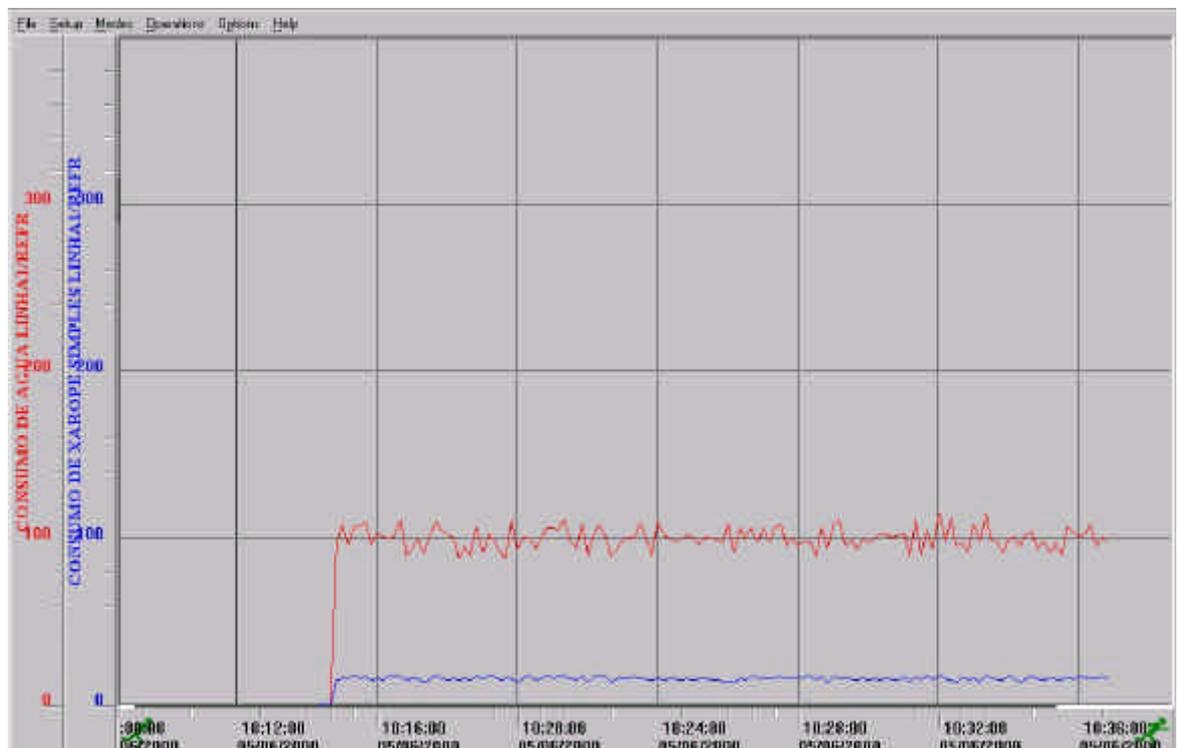


Figura 11: Gráfico de tendência

As operações possíveis neste tipo de gráfico são:

- a) Movimentação de cursor:
Um cursor gráfico vertical (dial) é passeado pela figura fornecendo a leitura dos pontos onde o cursor intercepta as curvas. Serve para examinar os valores críticos da variável e o momento em que um evento ocorre.
- b) Paginação:
Realiza paginação horizontal do gráfico para visualização dos pontos passados.
- c) Zoom horizontal e vertical
Realiza ampliação da parte do gráfico selecionada por uma janela de edição. A redução de escala pode ser feita para se retornar a escala original, ou para uma atingir uma escala reduzida, gerada pela compressão dos dados armazenados.

Tendência histórica:

Quando se deseja armazenar valores de variáveis em disco por longos períodos de tempo (até 1 ano), se recorre ao registro histórico. O registro histórico possui um período de amostragem tipicamente maior que o da tendência instantânea (alguns minutos). Os valores amostrados podem ser:

- Os valores instantâneos da variável no instante da amostragem histórica.
- Valores resultantes da compactação de amostras de tendência instantânea coletados a períodos inferiores.

Nos sistemas mais recentes, pode-se visualizar os gráficos de tendência instantânea e histórica de forma concatenada em um único sistema de coordenadas.

Hoje prefere-se realizar todas as funções históricas de um sistema SCADA em um software externo denominado PIMS (*Process/Plant Information Management System*). Os PIMS possuem melhores algoritmos de compressão de dados e armazenam tipicamente 5 anos de operação em um winchester de dimensões normais (20 Gbytes).

Os sistemas PIMS possuem uma riquíssima variedade de funções para manipulação dos registros históricos:

- a) Rápida importação de variáveis de processo através do *tag browser*. Você escolhe uma variável do dicionário de dados e arrasta o tag para o sinóptico. O número de penas simultâneas é ilimitado.
- b) Um gráfico de tendência configurado desta maneira pode ser salvo e constituir uma vista personalizada do processo de um operador em particular. Para salvar a vista deve-se apenas clicar em *Favorites*.
- c) Além das funções de *panning* e *zoom*, podemos inserir múltiplos cursores no gráfico para visualização dos valores das variáveis.
- d) Pode-se incluir o sinóptico em uma mensagem e enviá-la por *e-mail* a algum especialista que esteja em casa e queira analisar os dados.
- e) Podemos suspender a impressão de qualquer das penas por tempo indeterminado.
- f) Etc.

Gerador de relatórios

Até agora os únicos dados armazenados em disco foram os relativos aos registros de tendência instantânea e histórica. Evidentemente uma das principais funções dos sistemas SCADA reside na sua capacidade de armazenar dados e produzir relatórios de produção ao final de um turno, dia ou mês. Os relatórios de produção incluindo os relatórios de balanço de massa ou energia, demonstram quanto uma determinada planta produziu, quanto consumiu de insumos, de energia, etc e constituem o principal relatório de interesse gerencial. Outro alvo de interesse, desta vez por parte do pessoal da manutenção são os relatórios de monitoramento de equipamentos que dizem quando cada equipamento parou, por que parou e por quanto tempo ficou parado.

O usuário deve definir as variáveis que farão parte do relatório e o seu período (ou instante) de amostragem. A armazenagem de um dado pode estar vinculada a um evento e não apenas a um horário. Isto é muito comum em sistemas de batelada.

Depois deverá ser definido o formato do relatório e o instante de sua impressão (final do turno, dia, etc).

Um gerador de relatórios pode ter a forma de um editor WYSIWYG, (*What you see is what you get*), de uma planilha, pode ser definido por uma linguagem especial textual ou orientada para blocos de funções.

Construtor de estratégias de controle

Alguns sistemas supervisório possuem uma linguagem procedimental ou gráfica (blocos lógicos) para descrever procedimentos de controle a serem realizados a nível de supervisão. :

Basicamente as funções de um construtor de estratégias podem ser agrupadas nas seguintes bibliotecas:

- a) Biblioteca de entrada e saída
Realiza leitura e escrita de variáveis na RTU, solicita dados ao operador.
- b) Biblioteca de funções matemáticas:
Funções: funções convencionais + Max, Min, Média, Alarm, etc.
- c) Biblioteca de funções lógicas:
Funções: And, Or, Xor, Not, Pulso, etc.
- d) Biblioteca de funções de controle
Funções: PID, PI, etc.
- e) Biblioteca de funções de tratamento de sinais
Funções: *Lead_Lag*, Atraso, Interpolação linear em tabela, Filtro, Função Log após ganho e *offset*, Função Exp após ganho e *offset*, Aplicação de ruído, Totalização (integração), Derivação.

O construtor de estratégias tem sido totalmente substituído pelo módulo SoftLogic que programa as funções típicas de um CLP utilizando as linguagens do padrão IEC 61131-3.

Log de eventos

Registra no arquivo de *diário de bordo* todos os eventos relevantes de operação, com data, hora, descrição do evento e operador logado na hora do evento. Os eventos de interesse geralmente são:

- Eventos de configuração da base de dados
- Eventos de operação críticos tais como: ação sobre malhas de controle, partida e parada da planta ou de equipamentos críticos, etc

Este registro está vinculado a existência de um sistema de senhas para identificar cada operador que assuma a operação. O operador que deixa a operação deverá realizar o *logout* da estação enquanto o novo realiza o *login*, identificando-se.

Linguagem para definição de aplicações do usuário

Permite ao usuário criar seus próprios programas associados a ocorrência de algum tipo de evento. Os eventos podem ter diversas naturezas. Alguns supervisórios contemplam apenas parte destes eventos. Hoje a compreensão da natureza dos eventos possíveis de ocorrer tem trazido alto grau de generalidade às ferramentas. O quadro abaixo foi baseado na estrutura da linguagem Script do InTouch™ da Wonderware.

Quando lançar o programa ?

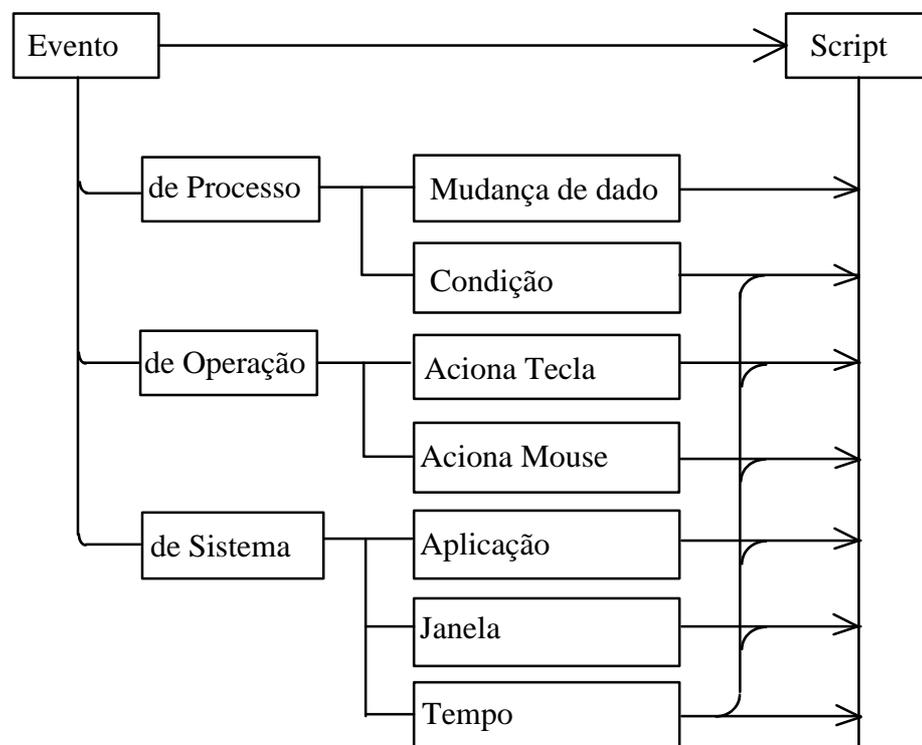


Figura 12: Eventos no InTouch

Os eventos possíveis são:

Variação de um dado de processo, atendimento de uma condição de processo, acionamento de uma tecla especial, click do mouse sobre um objeto ativo (*pushbutton*), lançamento de uma aplicação, abertura de uma janela,

Seja por exemplo o evento gerado pelo acionamento de uma tecla especial do teclado.

O programa (*Key Script*) pode ser programado para ser ativado:

- Ao se apertar a tecla
- Enquanto a tecla permanecer apertada a cada x ms (gera eventos de tempo).
- Ao se aliviar a tecla

O evento associado a uma condição (equação booleana envolvendo valores de variáveis de processo) causa a execução de um *Condition Script*:

- Quando a expressão se tornar verdadeira
- A cada x ms enquanto a expressão for verdadeira
- Quando a expressão se tornar falsa
- A cada x ms enquanto a expressão for falsa

E assim por diante.

Hoje praticamente todos os sistemas SCADA abandonaram linguagens próprias e padronizaram o Visual Basic ou o VBA da Microsoft como sua linguagem de geração de aplicações.

O que deve fazer o programa ?

As principais funções disponíveis devem incluir:

- Todas os operadores e funções matemáticas e lógicas inclusive:
mod, div, shr, shl
- Funções de manipulação de strings
- Comando condicional:
If condição **then**
then_statement
else
else_statement
Endif
- Acesso a todas as variáveis de processo:
Campos de valores
Campos de definição
Por exemplo: tag.campo_hh: valor do nível de alarme muito alto
- Criação de variáveis temporárias = dinâmicas = virtuais
- Reconhecimento de alarme de uma variável ou classe de variáveis
- Diálogo com o usuário:
Exibição de janela de mensagens e colocação de pergunta ao usuário.
- Envio de comando a remota:
Definição de uma variável de controle: set-point, variável controlada, etc
- Carga de programa ou receita na memória do CLP
- Ativação de nova aplicação
- Alteração do aspecto da I.H.M:

Alteração de visibilidade de janelas, do ponto de abertura ou outro atributo da próxima janela, etc

- l) Impressão de telas e relatórios
- m) Manipulação de campos de bits
- n) Acesso a variáveis do sistema: data, hora, operador corrente, etc
- o) Inclusão de comentários no programa
- p) Etc

Exemplo:

A linguagem Control Sequence Language (CSL) do sistema RealFlex™ da BJ Software.

Operadores:

Lógicos	AND; OR;
Relacionais	= <> > < >= <=
Aritméticos	+ - * / % Abs Fabs
Algébricos	Sqrt, Log, Sin, Cos, Tan, Rsin, Rcos, Rtan
Sistema	Hour, Minute, Month, Month_Day, Time, Week_day, Year_day

Funções:

CSL tag_12	Define procedimento CSL
Continue When (expressão)	Interrompe execução até expressão se tornar verdadeira
Display formato	Exibe formato para operador

Dsave entidade	Salva toda a base de dados correspondente a entidade em disco: dsave analog
Dstuff a = b	Força o valor da variável a sem passar pelo processamento de dados e alarmes do RealFlex. Atualiza o valor da variável em RAM e disco.
Stuff a = b	O mesmo que Dstuff porém sem atualizar o disco
EndCSL	Final de um procedimento CSL
Exit	Encerra processamento do CSL corrente e o desabilita
IF (expressão) <comandos_então> ELSE <comandos_senão> ENDIF	comando if / then / else até 3 níveis de aninhamento
Goto step n	Vá para o passo n

Goto [ponto_id] n1, n2,..n9	Se valor do ponto = nx então vá para step nx senão vá para step n1
Print_Screen tela [delay]	Exibe o sinótico pedido e realiza Hardcopy após delay segundos
Remove_Display	Fecha janela com último sinótico exibido
Report nome_do_relatório	Imprime relatório do sistema ou definido pelo usuário Ex: Report comm
Recipe nome_da_receita	Carrega receita solicitada
Return	Termina tarefa CSL sem alterar o seu estado de habilitação
Send a = b	Usado para enviar comandos às unidades de controle. este comando está sujeito a testes de validade: A PCU deve estar ativa O valor da variável deve estar dentro da faixa de validade
Set a = b	Atribui o valor da expressão b ao ponto a, através do processamento normal do RealFlex (checagem de alarmes / eventos, processamento histórico, etc). É usado para atribuir dados a um pseudo-ponto
Shell comando_qnx	Submete um comando ao QNX
Step n	Introduz <i>label</i> numérico
Wait n_seg	Espera n segundos antes de executar próxima instrução

O endereçamento do ponto neste sistema segue o formato:

[Entidade, PCU, Tag]campo

Exemplos:

[an, pcu_20, analógica_14]UN_ACK : acessa o campo de existência de alarmes não reconhecidos da variável analógica 14 da remota pcu_20

```
stuff [st, system, pump_on ] = on
Muda status da bomba para ligado na base de dados
```

CSL toca_buzina

Step 1

```
(alarme reconhecido ou equipamento on line)
if ((([st, bs25, bs25-status]UN_ACK = CSL_OFF) or
([st, bs25, bs25-status] = ONLINE))
Goto 3
endif
(alarme não reconhecido e estado = LOCAL)
if ((([st, bs25, bs25-status]UN_ACK = CSL_ON) and
```

```

    ([st, bs25, bs25-status] = LOCAL)
    Send [st, horn, bs25_horn] = ON (ligue buzina)
endif
Step 2
    *Wait 4 (espere 4 segundos)
    if [st, bs25, bs25-status]UN_ACK = CSL_ON (espera reconhecer alarme)
        Goto 2
    else
        Send [st, horn, bs25_horn] = OFF (desliga buzina)
        Goto 1
    endif
Step 3
    Wait 4
    Goto 1
ENDCSL

```

Escreva uma versão estruturada para o programa acima.

Arquitetura:

Os principais componentes da arquitetura de um sistema supervisório são:

Hardware:

- a) Microcomputador industrial ou workstation
- b) Dispositivo de entrada de dados:
 - Teclado de engenharia
 - Teclado Funcional
 - Mouse ou "Track-ball"
 - "Touch Screen"
- c) Dispositivo de comunicação com o operador:
 - Monitor
 - Terminal de vídeo
- d) Dispositivo de comunicação com o processo:
 - Saída serial convencional: RS 232-C
 - Cartão multiseriial inteligente
 - Cartão de comunicação direto com a rede de CLPs
- e) Dispositivos de comunicação com outros sistemas:
 - Cartões de redes
 - Distribuidores de conexão ("hubs")
- f) Outros periféricos:
 - Impressoras
 - Sinópticos tradicionais
 - Mesas de controle de "*back-up*" (em desuso)

Software:

- a) Pacote supervisório básico
 - Contém as funções básicas descritas no item anterior
- b) Pacote *batch*
 - Contém as funções de gerenciamento de processo de batelada
- c) Pacote SPC/SPQ
 - Contém as funções estatísticas para *Statistical Process Control*, basicamente plotagem automática das cartas de controle e geração de alarmes quando um determinado processo foge de seu comportamento normal.
- d) Gerador de relatórios (a exportação de dados para planilha Excel é a melhor opção).
- e) Linguagem de quarta geração para definição de programas pelo próprio usuário
- f) Sistema especialista de tempo real

O elemento chave na arquitetura descrita é sem dúvida a interface entre o supervisório e o CLP. O uso de cartões inteligentes foi um elemento imprescindível para aliviar a CPU principal da tarefa de conversão de protocolos, cálculo de CRC, e comunicação propriamente dita. O uso de cartões inteligentes que "falam" o mesmo protocolo do CLP traziam diversas vantagens:

- Evita o gargalo de comunicação representado pelo cartão RS-232 C

- Permite o uso de diversos protocolos concorrentes para conexão com diferentes tipos de equipamentos, diminuindo portanto a dependência do cliente em relação a um determinado fornecedor.
- Diminui a degradação do sistema com o aumento do número de variáveis de processo.

Hoje a comunicação entre SCADA e CLPs é feita necessariamente via rede de comunicação de dados.

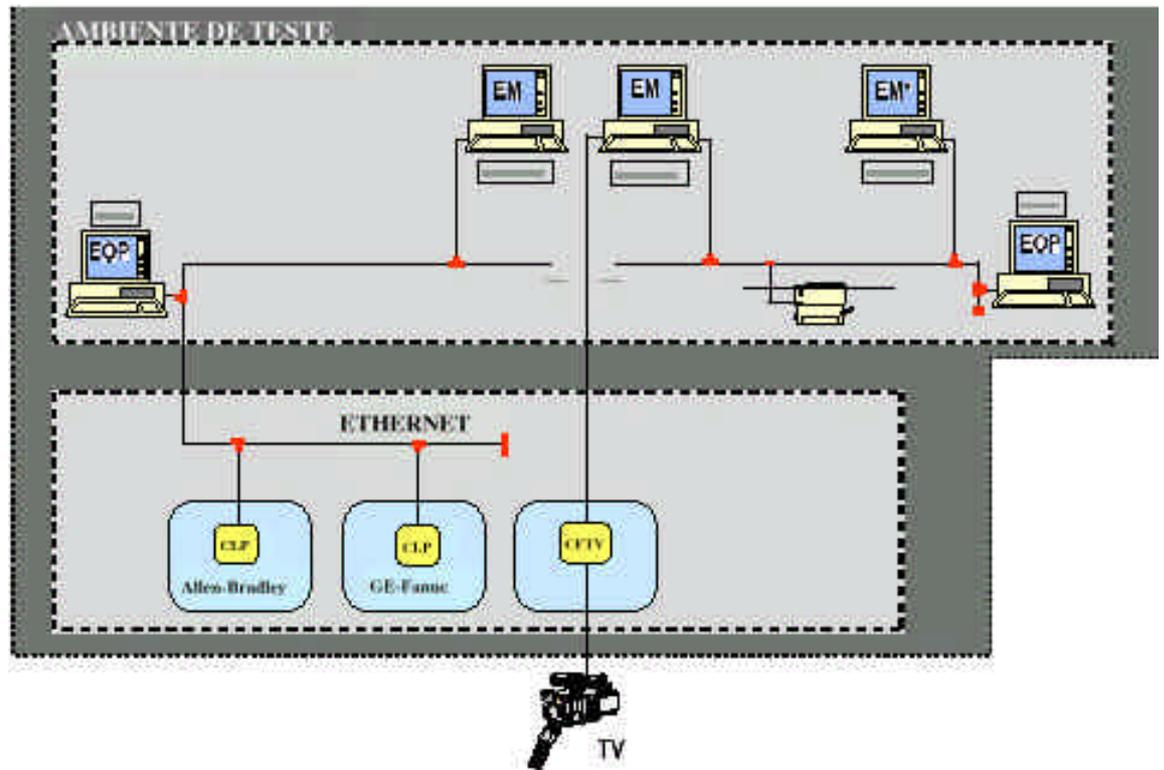


Figura 13: Arquitetura de um sistema SCADA

Requisitos da interface Homem Máquina:

O projeto da interface Homem Máquina de um sistema SCADA é baseado numa série de critérios ergonômicos tendo por base o relato de problemas de operação de diversos projetos anteriores [Shaw 87].

O principal objetivo da interface homem máquina é ajudar o operador a criar e manter uma imagem mental fiel e operacional do processo e seus componentes.

As principais dificuldades do operador moderno são:

- Deve basear suas ações em uma abstração da planta real. Deve transformar a interpretação dos dados em ações corretas.
- Não pode mais se especializar numa área específica do processo. Deve conhecer todas as áreas sob seu controle.
- Seus erros e omissões têm conseqüências muito maiores o que implica em maior pressão psicológica.
- O operador está submetido a uma postura de trabalho mais monótona o que conduz a fadiga.

Ao projetar uma I.H.M o projetista deve ter em mente:

- Diminuir a chance de erro do operador principalmente nos momentos de maior demanda operacional que coincide com o aumento do stress:
Representar as entidades de processo de forma única e consistente permitindo uma operação independente da encarnação física, sem surpresas para o operador.:

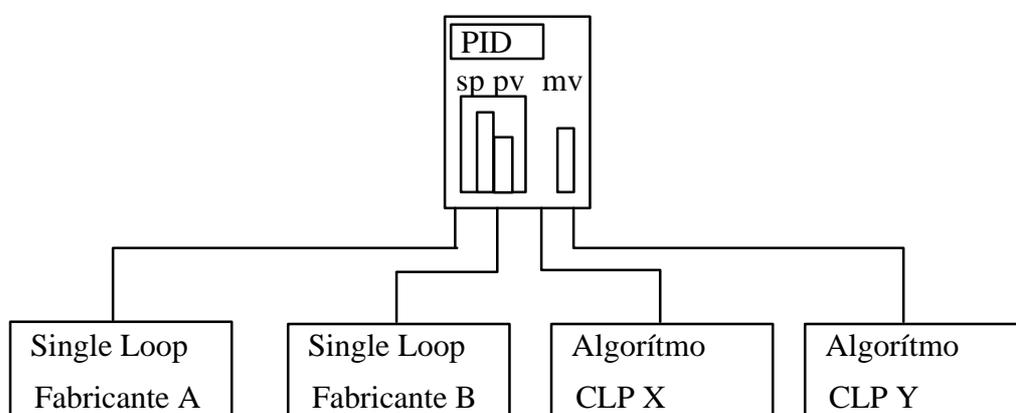


Figura 14: Conceito de objeto virtual

Embora os parâmetros exigidos para as 4 malhas mostradas possam ser diferentes, assim como os procedimentos de operação e sintonia das malhas, o operador enxergará apenas um tipo de PID (instrumento virtual). O processo se torna mais abstrato aos olhos do operador.

- Evitar as situações de monotonia que levam à desconcentração do operador. Sinópticos pouco representativos do processo e sem atrações de animação ou com muitos dados tabulares levam ao desinteresse.
- Evitar situações que acarretam cansaço.

- É sabido que muito elementos piscantes na tela trazem cansaço. A buzina de alarmes também deve ter uma tonalidade agradável e um nível não muito alto.
- d) Manter o operador sempre atento ao que realmente interessa.
Evitar excesso de informações na tela. Sinópticos muito cheios trazem excesso de informações que o operador não é capaz de processar. Os alarmes e informações devem obedecer ao critério da exceção.
Avalanches de alarmes devem ser evitadas. Pontos com alarmes crônicos devem ser desabilitados. Alarmes durante transitórios de partida e parada de equipamentos também. Para operações críticas como centros de operação de sistemas elétricos e centrais nucleares é recomendado o uso de sistemas especialistas para filtragem inteligente de alarmes [Kirshen 1992].
- e) Evitar consulta a referências externas ao sistema
Se o operador tiver dúvidas quanto a operação de elementos do sistema deverá consultar o próprio sistema (*Help on-line*). Se tiver dúvidas com relação a operação do processo deverá consultar um guia operacional *on-line* geralmente baseado em um sistemas especialista. Manuais são difíceis de se consultar numa emergência, e nunca estão disponíveis quando precisamos deles.

Conceitos ergonômicos para a construção de sinópticos: [Pettersson, 1989]

Os olhos tendem a se mover de:

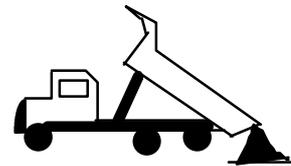
- Uma imagem grande para uma menor
- Uma cor saturada para uma não saturada
- Uma cor brilhante para um cor pastel
- Uma imagem colorida para outra monocromática
- Formas simétricas para formas assimétricas
- Algo que se move e pisca para uma imagem estática.

Logo ao construir um sinóptico devemos obedecer os seguintes critérios:

- Dar preferência a vídeos de 20"
- A construção do sinóptico deve ser bem balanceada: o número de elementos de informação por tela deve ser coerente com a capacidade humana de interpretá-los. Evite sinópticos congestionados ou vazios demais.
- O sistema gráfico deve propiciar:
 - Resolução suficiente para tornar a imagem legível (mínimo: 800 x 600 pixels)
 - Diversas cores simultâneas (16 .. 32 K cores)
 - Possibilidade de aplicação de texturas (*patterns*) sobre o desenho.
 - Caracteres com diversas formas e tamanho
 - Representação gráfica dinâmica (animações).
- Evite objetos grandes piscantes
- Deve haver redundância na forma de representar uma informação: valor, barras, enchimentos, etc. A representação mais natural é a mais indicada. Por exemplo, enchimento para tanques e silos, rotação para um forno de cimento ou britador de martelos, etc.

- ❑ Equipamentos devem ser desenhados de acordo com sua forma e tamanhos exatos. A representação fotográfica com excesso de detalhes, sombra, etc. é desaconselhável.

Isto é um caminhão



- ❑ A seqüência para ligar ou desligar equipamentos ou realizar ações de controle similares deve ser simples e intuitiva. Simplesmente selecione o objeto com o mouse e selecione a opção LIGA no menu. I.H.Ms de alguns terminais bancários que andam por aí são um bom contra exemplo.
- ❑ Mensagens devem ser claras, explícitas e auto suficientes.
Contra exemplo: Erro 46A: Execute o procedimento de emergência 78

GUIs - Graphical User Interfaces

O duelo entre interfaces gráficas padronizadas X interfaces gráficas proprietárias teve um único vencedor: as GUIs. Hoje o único problema reside na escolha do padrão:

Critérios:

- Deve rodar no S.O. e processador escolhidos
- Deve sobreviver a médio prazo
- Deve ter velocidade suficiente para aplicações de tempo real
- Deve permitir a transparência de aplicações via rede
- Deve ser independente de dispositivos gráficos de entrada/saída

Principais GUIs:

Fornecedor	GUI	SO
Microsoft	Windows	WNT/Windows 2000
OSF	Motif	UNIX/QNX4
AT&T/SUN	Open Look	UNIX/QNX2,4

OSF = Open Software Foundation = IBM + Digital + HP

O que é uma GUI ?

API
Windowing System
Imaging Model
SO

API = Application Program Interface

- Conjunto de chamadas de função
- Usuário especifica que janelas, Menus, ícones, etc irão aparecer na tela com a ajuda de um *toolkit*.

Windowing System = Gerenciador de janelas

- Conjunto de comandos para construir janelas, Menus, Dialog boxes, etc.
- Controle de criação, ajuste de tamanho, movimento e operação das janelas. A aplicação não tem de se preocupar se uma janela está sobre outra e de outros detalhes.

Imaging Model:

- Determina como linhas, textos, curvas, etc. vão aparecer na tela.

O padrão de interface do Windows é sem dúvida o padrão mais utilizado hoje, e é o padrão vencedor.

Tecnologias Web

A grande revolução nos sistemas SCADA se deu com a introdução de dois padrões: o OPC para comunicação do SCADA com os CLPs e demais dispositivos de controle, e os padrões da Web para disseminar informações a baixo custo para todas as áreas da empresa. Já em 1998 todos os supervisórios praticamente, ofereciam estações clientes leves que podiam exibir quaisquer dados da planta em um PC convencional dotado de um browser. Numa primeira fase tínhamos apenas a *feature de Internet publishing*, isto é, as informações eram disponibilizadas para leitura, mas o usuário não podia interagir com a planta, por exemplo dando comandos para a planta. Uma vez vencidas as barreiras iniciais de segurança, hoje não existem mais limitações e uma planta pode ser tanto telesupervisionada como telecomandada. Isto fez com que uma estação cliente Web pudesse ser fornecida a um custo extraordinariamente baixo.

Dentro deste novo contexto é hoje comum que informações de alarmes possam ser direcionadas para um e-mail, para um pager ou para um telefone celular, criando-se ainda mais flexibilidade para as atividades básicas.

Requisitos para escolha de um sistema supervisorio para uma determinada aplicação (Parâmetros de "benchmark"):

Para escolher entre sistemas supervisorios é necessário considerar-se a aplicação alvo, a plataforma desejada e as características de interligação a outros sistemas. Para pequenas aplicações (até 2000 pontos), a plataforma e o sistema operacional influem menos na escolha, mas para grandes aplicações tornam-se os elementos decisivos.

Benchmarks de fabricantes acentuam os pontos fortes do seu produto enquanto minimizam a importância dos demais pontos.

Principais características a verificar	
Sistema Operacional	Para grandes aplicações um SO multitarefas preemptivo é indispensável
Plataforma	Verificar se é multiplataforma, senão se a plataforma (PC ou Workstation) é adequada
Características gerais	Sinóticos, tendência, alarmes, diagnóstico, etc
I.H.M.	Verificar qual a GUI utilizada, ou se o sistema é proprietário.
Redes	Verificar qual a rede oferecida
Redes duais	Algumas aplicações requerem uso de redes redundantes
Sistemas especialistas	Se é desejável funções inteligentes tais como: <i>operator guidance</i> , alarmes inteligentes, operação automática, controle supervisorio, deve ser estudada a existência de um módulo SE no próprio sistema ou acoplado
Gerador de relatórios	O usuário deve ter a liberdade de definir seus próprios relatórios
Gerador de estratégia de controle	Se controle DDC for desejado, ou se pretende-se definir a estratégia de controle no próprio supervisorio.
Interface com bancos de dados relacionais, planilhas, etc	O SCADA deve propiciar interfaces com outros softwares gerenciais
Interface com CPD	Se é possível interligação com Mainframes e como é feita
Interligação a outras redes	Se protocolo TCP/IP está disponível
Interface com CLPs, Multi loops, etc	Se é possível uso de cartão inteligente, que cartão e características de velocidade
Abertura para novos aplicativos	Se é possível definir novas aplicações e personalizações em C ou outra linguagem.
Documentação	Se o fornecedor propicia documentação adequada, treinamento e suporte

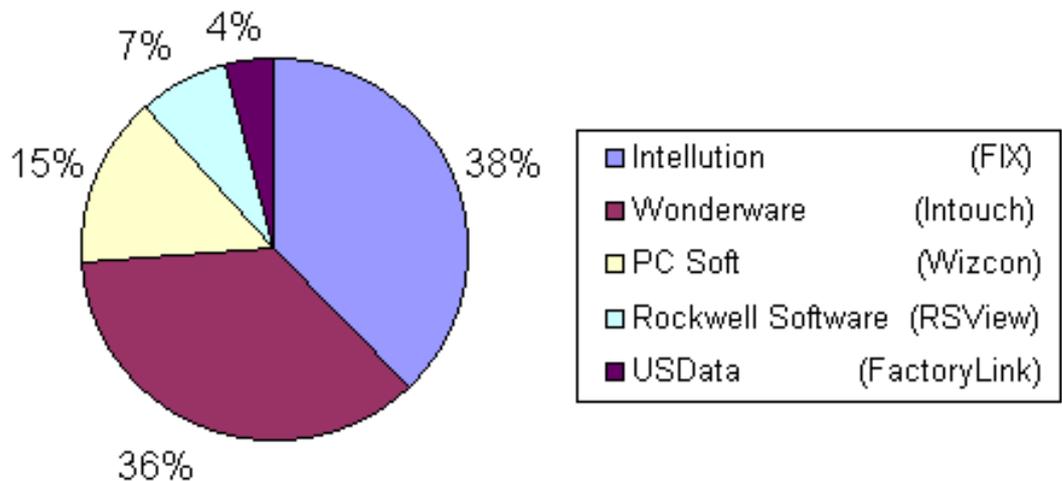


Figura 15: Marketshare de sistemas SCADA no Brasil em 1999 segundo www.automacao.net

		Honeywell	USDATA	PCSOFT	Rockwell	Wonderware	Intellution
CRITÉRIO	PESO	Scan3000	FactoryLink	Wizcon 7.x	RSView32 6.x	InTouch 7.x	Fix32 6.x
Ambiente Operacional	Elim.	OK	OK	OK	OK	OK	OK
Interface de Operação	15	32.40	39.45	41.25	37.50	38.55	48.15
Interface com o Campo	15	52.50	52.50	45.00	45.00	52.50	56.25
Arquitetura de Software	13	37.70	44.20	37.70	44.20	44.20	44.20
Configuração	11	33.00	32.89	38.61	40.59	36.19	41.36
Manutenção	10	31.00	36.00	40.00	33.50	35.00	40.00
Custo	10	17.50	17.00	20.50	27.50	31.00	30.50
Utilitários Disponíveis	8	14.00	19.20	32.00	26.00	32.00	32.00
Conectividade	8	16.80	20.80	18.40	20.00	20.00	25.60
Qualidade do Fornecedor	6	10.20	19.20	15.60	15.60	19.50	20.70
Documentação	4	10.40	10.40	10.40	10.40	10.40	10.40
TOTAL GERAL DE PONTOS		255.50	291.64	299.46	300.29	319.34	349.16

Figura 16: Resultado de *benchmarking* de teste realizado pela Petrobrás em 2000

Para este teste foram observados os seguintes critérios:

I Características fundamentais, em atendimento a alguns preceitos que nortearam a padronização atual, ou seja:

- Ser aberto e configurável;
- Possuir interface de operação;
- Possuir *drivers* de comunicação com os PLCs de uso do E&P;
- Possuir *toolkit* para desenvolvimento de *drivers* e outras aplicações.

II Características essenciais:

- Deter significativas parcelas do mercado;
- Ter efetiva representação com suporte no Brasil;
- Ter uma capacidade gráfica e funcional compatível com a padronização de IHM da E&P;
- Ter sido desenvolvido para ambiente Windows NT, ou seja, utilizar a sua capacidade de processamento de 32 bits;
- Utilizar a rede TCP/IP, como suporte para comunicação entre estações de operação e supervisão.

III Características desejáveis para os mesmos:

- Utilizar os recursos de janela próprios do Windows NT, tais como, *pop-up menus*, *drag & drop* e mais de uma janela de processo simultaneamente;
- Ser multiplataformas, ou seja, rodar em diferentes CPUs e/ou sistemas operacionais;
- Já ter sido utilizado, mesmo que em aplicações de pequeno porte e sob outros sistemas operacionais, pela E&P;
- Possuir infra-estrutura de conectividade, DDE, NetDDE , OLE e OPC ;
- Possuir *driver* de comunicação para o *software* de integração da E&P, o PI *-Plant Information*, da Oil Systems.

Um relatório completo dos testes é dado no documento: Relatório RL-3000.00 - 5520-850-PGT-003: Avaliação de Arquiteturas de Supervisão para Instalações Operacionais da E&P da Petrobrás.

Elementos de construção de sistemas SCADA:

Como seria impossível num primeiro curso estudar todos os elementos de projeto de um sistema supervisório, vamos apenas atentar para alguns pontos de interesse que servirão de base a execução dos trabalhos práticos:

Dicionário de Tags:

As variáveis do sistema ficam armazenadas em dois depósitos de dados principais (por entidade):

- Tabela em memória: Contém as variáveis de acesso mais contínuo. Não se pode perder tempo buscando em disco, as informações de alta frequência de acesso.
- Arquivo em disco:
- Contém as informações que só são acessadas por exceção.

Dois tipos de ações devem ser executadas em alta velocidade:

- a) Dado a identificação do ponto, achar o registro correspondente (índice na tabela e arquivo).
- b) Dado o Tag da variável achar o registro correspondente.

A estratégia a) ocorre quando o evento vem do processo. Por exemplo, o ponto 01.145.07 entrou em alarme. A partir da identificação da variável: CP/PAL/BIT somos capazes de achar o seu índice, ou através de uma simples fórmula (distribuição linear) ou consulta a uma *look-up table* tendo cp como índice de acesso.

A segunda situação ocorre quando o operador solicita dados ou muda um parâmetro de uma variável baseado no seu tag. A partir do tag devemos achar o índice da variável. Esta tarefa pode parecer simples. Bastaria montar uma tabela contendo os campos: tag, tipo da variável e índice, ordenada pelo campo de tag e realizar um algoritmo de procura binária.

Dado que um sistema de médias proporções, possui cerca de 10000 pontos, este algoritmo que possui tempo médio de acesso da ordem $n \log n$ seria lento demais.

Qual o número médio de acesso para se achar um tag na situação acima ?

Geralmente tags são organizados em uma tabela *hash*.

A partir dos caracteres do tag (8..12) é gerada uma chave por um algoritmo de máximo espalhamento (função *hash*). O número gerado é o índice da variável. Como dois tags diferentes podem ter a mesma assinatura, pode haver colisão se for determinado o mesmo local de armazenamento para as duas variáveis. Neste caso existem vários algoritmos possíveis. O mais simples, denominado *open hash*, procura a célula vazia mais próxima da célula calculada.

Exemplo de função hash:

```

int hash (char *tag, int maxtable)
{
    int aux;
    char *p;

    p = tag; aux = 0;
    while (*p ne '\0')
    { aux = (aux << 1) xorb *p;
      ++p }
    return (aux % maxtable)
}

```

maxtable deve ser um número primo.

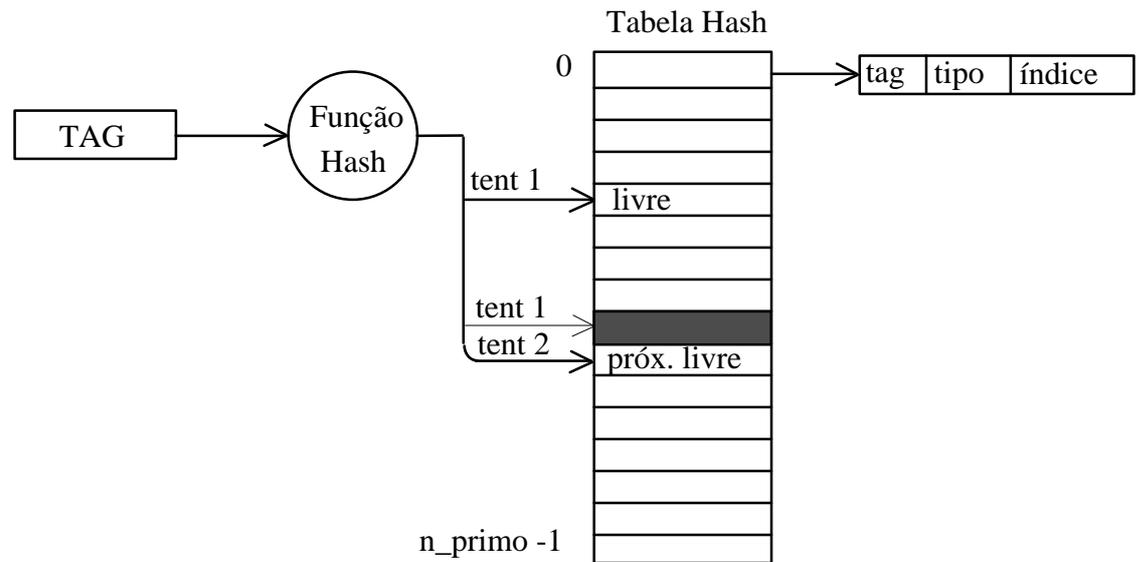


Figura 17: Armazenamento de tags em tabelas Hash

Implemente a função acima e calcule para uma tabela de tamanho 112 as chaves produzidas para os tags:

12-TC-54 18-BP-23a 01-PIC-29c 01-TC-55

Arquitetura de software:

A título de exemplo será mostrado o esquema de definição de uma variável analógica em um sistema SCADA convencional.

Entidade: Variável analógica

Estrutura de Dados

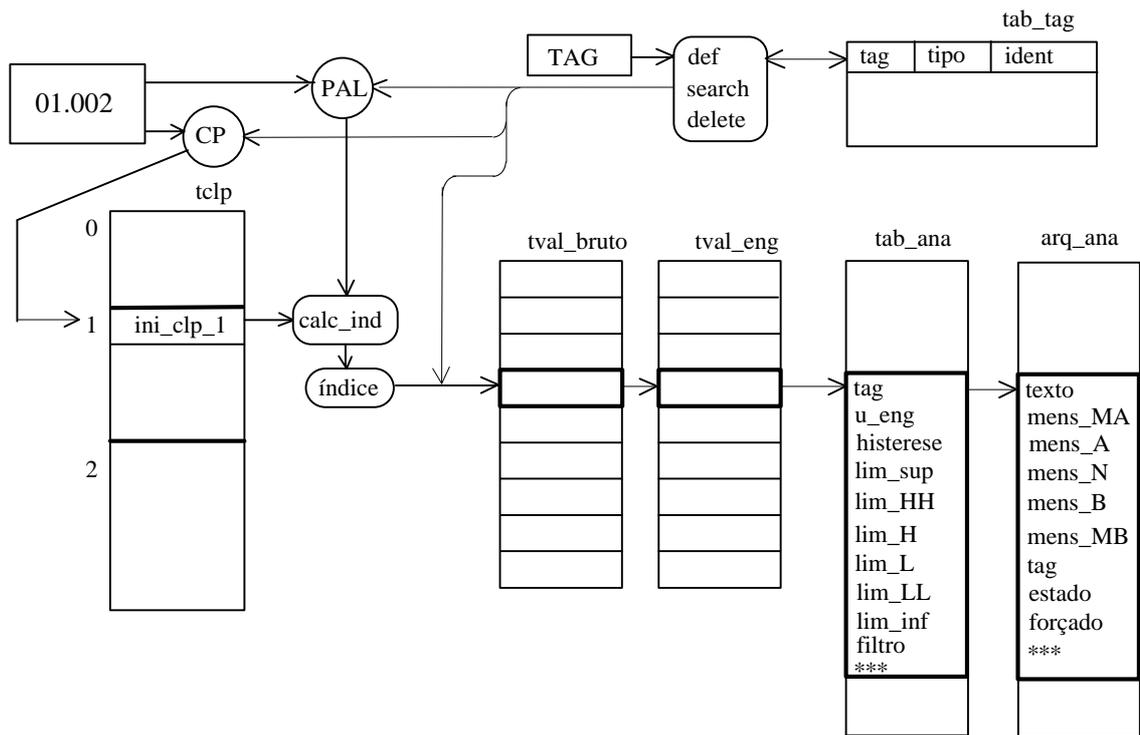


Figura 18: Armazenamento de tags em tabelas Hash

Descrição das funções de interface do drive entre um Controlador Lógico Programável (CLP) e o Computador de Supervisão

A interface de comunicação entre um sistema de operação/supervisão e um CLP é uma convenção que estabelece como os dados serão agrupados na memória do CLP para facilidade de acesso pelo computador principal (maior rapidez de leitura/escrita).

Dependendo da aplicação (novo projeto x reforma) pode inclusive não haver nenhuma convenção, sendo os dados colocados na memória do CLP de forma esparsa e não organizada.

De uma forma geral é interessante agrupar variáveis de natureza semelhante em uma mesma área, por exemplo: área de variáveis analógicas, área de parâmetros de malhas PID, etc.

Estas áreas cumprem também uma função de agrupamento lógico como por exemplo:
área de liga de equipamentos, etc.

As áreas de um determinado tipo de variável pode ser subdividida em novos tipos lógicos:

área de comandos:

- comandos de liga de equipamentos
- comandos de liga de sequências
- comandos de desliga de equipamentos
- comandos de desliga de sequências
- comandos de definição de modos de equipamentos (local/remoto, automático/manual, etc)
- parâmetros de variáveis analógicas
- parâmetros de malhas de controle
- valores de constantes de operação
- configuração de malhas PID (Manual/Remoto/Automático)
- set-points de malhas PID

área de status:

- estado de equipamentos
 - funcionando; pronto para partir; defeito: (FUN/PPP/DEF)
 - local/remoto; automático/manual;
 - bits de defeito particulares que são dependentes do estado do equipamento: desalinhamento de correia, sobrecarga no motor, etc.
- estado de seqüências
- estado de modo de operação de equipamentos
- valores de variáveis analógicas
- valores de totalizadores
- valores de saída de malhas de controle

obs: O bit geral de defeito corresponde à função OR de todos os demais bits de defeito de um determinado equipamento.

área de palavra chave

Corresponde a uma única palavra para cada CLP com a função de *watchdog timer*. Esta palavra é escrita periodicamente pelo computador e é lida e resetada pelo CLP a períodos fixos de tempo, digamos a cada 500ms. Se durante algum ciclo, a variável não estiver setada, o CLP sabe que o supervisor saiu do ar e pode tomar uma ação de emergência como por exemplo ativar um computador de stand-by ou soar um alarme.

A interface física:

A interface física enxerga a memória do CLP como uma tabela organizada em palavras ou bits.

O drive de um determinado CLP deve ser capaz de ler e escrever na sua memória, executando o protocolo particular daquele equipamento. Esta tarefa funciona como servidora de diversos clientes. Um destes clientes pode fazer o papel de interface lógica com o CLP.

Se a interface lógica receber uma ordem do tipo:

Ligar o equipamento 12-BP-03 (bomba de polpa), irá traduzir este comando como uma mensagem (send) para o drive físico do tipo:

Escrever 1 no bit 5 da palavra 3 do CLP 2.

Os comandos da interface física são basicamente:

- a) Ler uma palavra:
Read_word(clp_num, index);
A variável analógica geralmente ocupa os 12 primeiros bits de um inteiro (valor bruto)
- b) Ler um bloco de palavras:
Read_word_bloc(clp_num, index, num_words);
Lê num_words palavras da memória do CLP e as entrega à tarefa cliente.
- c) Escrever uma palavra:
Write_word(clp_num, index)
- d) Escrever um bloco de palavras:
Write_word_bloc(clp_num, index, num_words)
- e) Ler uma variável discreta:
Read_dig(clp_num, pal, bit);
- f) Ler um string de variáveis discretas:
Read_dig_bloc(clp_num, pal, bit, num_bits);

Lê num_bits bits da memória do CLP à partir do bit inicial indicado

- g) Escrever uma variável digital:
Write_dig(clp_num, pal, bit)
- h) Escrever um bloco de variáveis digitais:
Write_dig_bloc(clp_num, pal, bit, num_bits)

A numeração dos bits na palavra depende do CLP em questão.

Na maior parte dos casos a numeração começa em 0, onde o bit 0 corresponde ao bit menos significativo (LSB) da palavra.

Na maior parte dos casos uma simples porta serial do micro (RS232-C) é utilizada para conectar o computador com o CLP ou com um "gateway" que dá acesso à rede de CLPs.

Nos sistemas mais modernos, este protocolo é colocado *embedded* em uma placa de comunicação inteligente, que se interliga diretamente a rede (geralmente proprietária) do CLP. Neste último caso não ocorre o estrangulamento da comunicação causado pelo gargalo de 9600 bauds da linha serial.

Exemplo de organização da área de interface:

Área	Tamanho (palavras)	Descrição
Liga	64/16	Comandos de liga de equipamentos Comandos de liga de seqüências
Desliga	64/16	Comandos de desliga de equipamentos Comandos de desliga de seqüências
Modo	2 * 64/16	Comandos de definição de modos de equipamentos (2 bits por equipamento). Os nomes dos modos são configurados pelo usuário por classe de equipamentos.
Status	6 * 64/16	Bits de status de equipamentos Modo1 Modo2 FUN FUN_RÉ PPP DEF Bits de status de seqüências
Alarmes	64	Alarmes complementares Discriminam os alarmes de equipamentos e seqüências
Analógicas	128	Contém os valores correntes da variáveis analógicas: valor bruto de 0 a 4095.
Parâmetros de Analógicas	128 * 6	Contém os parâmetros de configuração das variáveis analógicas: Limite_Superior Limite_HH Limite_H Limite_L Limite_LL Limite_Inferior
Modo PID	16/8	Define o modo de operação de cada malha PID (2 bits por malha): MANUAL REMOTO AUTOMÁTICO

SP	16	Contém os valores de set-points de cada malha em percentual
SP remoto	16	Contém valores dos set-points remotos para cada malha de controle
PV PID	16	Contém o índice para a variável de entrada de cada malha
MV PID	16	Contém os valores das variáveis de saída de cada malha em percentual [0 a 99.9 %] Pode ser lida ou escrita pelo supervisor dependendo do estado da malha.
Parâmetros PID	16 * 10	Contém os parâmetros de configuração das malhas PID: Status_Modo KP Ti Td Banda_Morta Bias Limite_Inf_Saída Limite_Sup_Saída etc
Constantes Paramétricas	12	Constantes definidas pelo operador
Totalizadores	8 * 4	Contém os valores totalizados: Valor totalizado com Reset Valor totalizado sem Reset
Índice Totalizador	8	Índice para as variáveis analógicas totalizadas
Data Reset	8 * 2	Contém a data e hora do último reset para cada totalizador
Palavra Chave	1	Palavra chave
Reserva	N	Área reservada para expansão: Novas entidades, Receitas, etc

EXERCÍCIOS

- 1) Através de um sistema supervisorio convencional também é possível agir diretamente sobre uma variável manipulada, colocando-se uma malha em manual. Desenhe o esquema de ação sobre malhas de controle para os seguintes modos de operação:

Automático:

Manual:

- 2) Examine a tabela de caracteres semi gráficos dada no Apêndice 1 e responda:
- Como era desenhado o gráfico de barras com atualização dinâmica ?
 - Como era desenhada uma válvula de controle na posição vertical ?
E horizontal ?
 - Como era desenhada uma válvula de comando manual ?
 - Como era representado um moinho de bolas ?

- 3) Em um editor de sinóticos orientado para objetos, é possível se definir *click-actions* de diversas naturezas com o uso de uma linguagem procedimental.

Podemos por exemplo:

- Reconhecer um alarme
- Ativar um programa escrito em C
- Exibir um tópico de hipertexto
- Exibir um novo sinótico
- Exibir os alarmes referentes a uma classe.
- Executar um procedimento escrito em linguagem de 4ª geração
- Mudar o atributo de uma variável

Aponte as vantagens e limitações desta estratégia.

- 4) Um sistema SCADA possui uma linguagem de blocos funcionais. Uma expressão de gatilho deve ser definida para deflagrar a ação do bloco. Somente quando esta expressão adquire valor verdade TRUE, a função do bloco é ativada, transformando entradas em saídas.

Descreva que características deve possuir esta linguagem para ser capaz de representar as funções de controle típicas de sistemas SCADA.

Suponha que um bloco execute uma ação de emergência a ser deflagrador quando uma certa condição de variáveis de processo ocorrer (expressão booleana). Escreva um programa em linguagem C para:

- Interfacear com o usuário e receber a definição da expressão de gatilho
- Compilar a expressão de habilitação definida.
- Aplicar a ação de controle.

Faça um programa que demonstre o funcionamento do seu programa na tela.

Descreva as características que a linguagem de blocos deve possuir para ser útil em processos de batelada para:

- Deflagrar ações de controle na ocorrência de eventos.
- Temporizar ações.
- Enviar receitas ao CLP

- d) Amostrar o valor de variáveis de processo em disco ou tabela em memória.
- e) Gerar um relatório ao final da batelada.
- 5) Projete um sinóptico para o problema do reator estudado no capítulo de programação de CLPs. Desenhe uma máscara e indique os campos de atualização dinâmica. Leve em consideração todas as observações discutidas no capítulo sobre ergonomia. Se você tiver acesso a um sistema SCADA, construa o seu sinóptico e tire um *hardcopy* da tela.
- 6) Faça um programa que desenhe uma barra vertical proporcional ao valor de uma variável analógica e a atualize, caso o valor da variável varie cerca de 5%.
- a) Utilize um conjunto de caracteres semigráficos como definido no apêndice A
- b) Utilize as funções gráficas de um pacote gráfico qualquer.
- 7) Um determinado sistema supervisório muito usado na área petroquímica definiu um objeto composto denominado tanque. Liste os atributos que no seu entender tal objeto deva possuir.
- 8) Uma variável cálculo foi criada para servir de base a animação gráfica que simula a rotação de um forno de cimento. O forno deve girar na tela somente quando o bit de funcionando do motor do forno estiver setado e a velocidade de giro deve ser proporcional a velocidade do forno em rpm.
- a) Se você deseja apenas dar a idéia de movimento sem correlação com a velocidade real, qual a função mais simples a ser utilizada ?
- b) Defina este cálculo supondo uma linguagem como apresentada no objeto cálculo.
- c) Suponha que você não disponha da medida de velocidade e que quer realizar a animação baseado num valor nominal de 2 rpm. Que função temporal o seu processador de cálculo deve possuir para que a velocidade da atualização dinâmica seja coerente com o comportamento real do forno ?
- 9) Gerenciador de janela de alarmes:
Construa uma tarefa para gerenciamento de alarmes. Ela deve gerenciar 4 linhas na tela dinamicamente. Esta tarefa funciona como servidora, e recebe mensagens de outras tarefas clientes descrevendo eventos de alarme. As mensagens são enfileiradas e apenas as 4 mensagens mais recentes aparecem no vídeo. Quando o operador reconhece o alarme mais recente ele é retirado da tela e as demais mensagens são promovidas. O operador pode também reconhecer todos os alarmes simultaneamente. Construa uma tarefa de inicialização para criar o gerenciador de alarmes e 2 clientes. Ao acionar a tecla <A> do seu teclado, o cliente A gera um evento. A tecla funciona de modo análogo para o cliente B. A tecla <r> reconhece um alarme e a tecla <R> reconhece todos.
- 10) Discuta as vantagens e desvantagens de se implementar uma linguagem de comandos para usuário compilada ao invés de interpretada.

- 11) Construa um programa CSL para:
- Exibir uma tela de sinótico tão logo qualquer uma de três variáveis de processo críticas entre em alarme.
 - Imprimir um relatório automaticamente ao final de um dia
 - Totalizar a vazão de duas bombas em funcionamento exclusivo pelo motor que estiver ligado.
- 12) Considerando os critérios estabelecidos para uma linguagem de alto nível para uso pelo usuário, quais as funções não contempladas pela linguagem CSL ?
- 13) Num sistema dirigido por eventos, o supervisor deve checar o valor de variáveis analógicas contra os limites de configuração para detecção de alarmes. Como esta tarefa poderia ser transferida para o sistema de CLPs ?
- 14) Animação de um sinótico:

Construa um sinótico contendo no mínimo os seguintes objetos:

- Um tanque.
- Uma válvula tipo borboleta.
- Uma bomba.

Anime o sinótico usando uma tarefa para gerar valores segundo uma função qualquer.

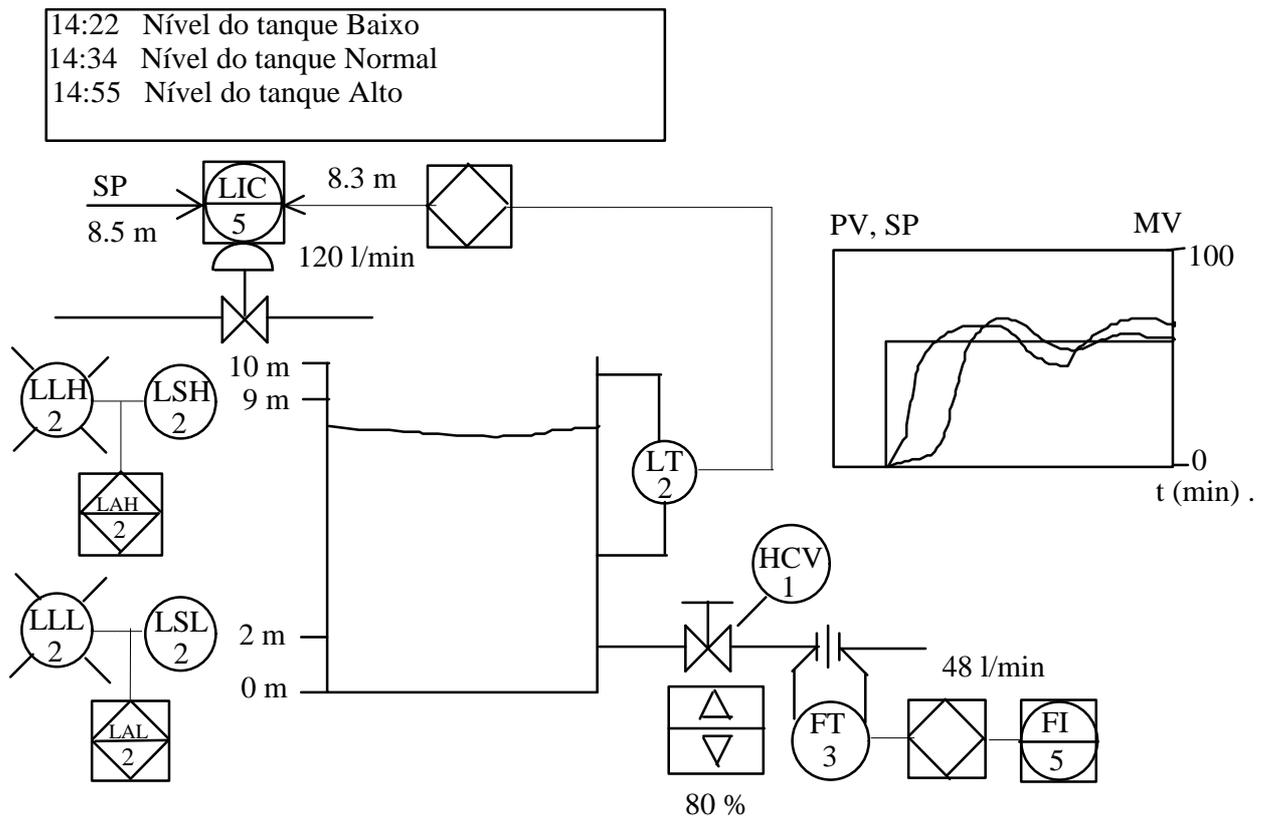
Outra tarefa deve atualizar os valores das variáveis no vídeo segundo as seguintes convenções e formas de representação:

- Nível do tanque: variável analógica.
 - valor numérico.
 - barra vertical indicando o seu grau de enchimento.
 - gráfico de tendência.
- Estado de funcionamento da bomba: variável discreta: a cor da bomba deve indicar o seu estado:
 - vermelho = defeito
 - amarelo = pronto para operar
 - verde = funcionandoDois botões com rótulos LIGA e DESLIGA devem permitir ao operador comandar o seu funcionamento .
- Estado da válvula: variável discreta.
 - A válvula possui dois estados: aberto e fechado.
 - O ícone ou objeto que a representa deve ser modificado de acordo com o seu estado instantâneo.

Uma tarefa de inicialização é responsável pela carga inicial da imagem do sinótico e pela criação das demais tarefas (simulação da planta, atualização da tela, interface com o usuário, etc).

- 15) Simulação de um processo contínuo

Controle DDC + sinótico animado



Um tanque cilíndrico de 5 metros de altura de 1,596 m de diâmetro, pode ser enchido através de uma tubulação ligada a uma válvula de controle, cuja vazão máxima é de 120 l/min. Uma válvula Manual permite esvaziar o tanque a uma razão de 60 l/min.

Projete um controlador PID discreto, que controle o nível da caixa através da manipulação da vazão de entrada. Use um período de amostragem de 100 ms para o algoritmo apresentado no paper "DDC Control Algorithms and their implementation" extraído do livro "Real Time Computer Control" de Stuart Bennet:

$$\begin{aligned}
 s_0 &= 0 \\
 s_n &= s_{n-1} + e_n \\
 m_n &= K_p * e_n + K_i * s_n + K_d * (e_n - e_{n-1}) + MV \quad (\text{Eq 4.7})
 \end{aligned}$$

onde:
 $K_p = K_c$
 $K_i = K_c * T_s / T_i$
 $K_d = K_c * T_d / T_s$

Desenhe um sinótico com as seguintes animações:

- a) O nível da caixa é atualizado dinamicamente sendo também exibido o seu valor em metros.

- b) Indicação da vazão de saída. Botões de controle de incremento e decremento permitem simular a abertura e o fechamento da válvula de saída em passos de 10% (é evidente que uma válvula manual só pode ser operada diretamente do campo).
- c) O valor do set-point de nível pode ser variado pelo operador.
- d) O valor da vazão de alimentação é indicado em unidade de engenharia e percentual.
- e) Um gráfico de tendência oferece uma plotagem do set-point, variável de entrada (nível), e variável manipulada com atualização dinâmica a cada 0.5s.
- f) Clicando com o mouse sobre o controlador PID abre-se uma janela para ajuste dos parâmetros K_C (ganho proporcional) ou PB (banda proporcional = $100/K_C$), T_i (*reset time* em min) e T_d (*derivative time* em min) para sintonia do controlador.
- g) Indique a condição de nível de tanque alto e baixo através de uma sinalização no próprio sinótico ou exibindo uma mensagem numa linha de alarmes.

Sintonize o controlador PID. Veja como o seu controlador reage a um degrau aplicado ao set-point. Analise o seu comportamento quando você aplica um degrau na carga deixando o set-point constante, por exemplo diminuindo a vazão de saída para 20 l/h.

Implementação:

- Uma tarefa de inicialização deverá criar todas as demais no início do programa.
- Uma tarefa deve gerenciar todas as interações com o usuário (I.H.M.).
- O controlador PID e a caixa devem ser modelados cada qual por uma tarefa independente.
- Concentre todas as atualizações do vídeo em uma única tarefa.
- Quem optar pela linha de alarmes, deverá construir uma tarefa gerenciadora de alarmes que exiba as 4 mensagens de alarmes mais recentes numa linha no topo da tela.

- 16) Especifique um relatório de paralisações de equipamentos.
- 17) Discuta as vantagens de se utilizar um *browser* como alternativa de visualização e operação em sistemas SCADA.
- 18) Qual a vantagem de se poder limitar os relatórios a serem desenvolvidos em um sistema SCADA e de transferir parte destas funções para o nível do MES ?

BIBLIOGRAFIA

- Sites sobre SCADA: <http://www.abpubs.demon.co.uk/scadasites.htm>
- KAAAYE Steve, "Designing and Integrating Workstations Into Plant Operations", I&C May, 1990
- KIRSHEN, D. S., and B.F. Wollenberg, "Inteligent Alarm Processing in Power Systems", Proceedings of the IEEE, vol 80, n°5, 1992.
- KUTOVA, Marcos André, OpenLook Overview da Especificação Funcional, Monografia de final de curso: Sistemas de Automação II., EEUFMG
- PAULA FILHO , Wilson de, e Ana Paola Amaral Duarte - Um editor orientado para geometria, Anais do x° Congresso da SUCESU
- PETTERSON Bo C, "Ergonomics: A Key to Reliable Process Operation", Control Engineering, August 1989:14 - 15.
- SEIXAS FILHO, Constantino, "Um sistema gráfico multitarefas para representação dinâmica de processos industriais", Anais do 8° Congresso Nacional de Automática (SBA), Setembro, 1990.
- SHAW, John . "Reducing Operator Error in Distributed Control Systems", Proc. of the ISA International Conference and Exhibit, Anaheim, Oct 1987
- SUN Microsystems, Inc., OPEN LOOK Graphical User Interface, Application Style Guidelines, Addison-Wesley Publishing Company, Inc.,1990
- USDATA, FactoryLink Technical Notes, version for Multitasked DOS, 1990
- WONDERWARE Software Development Corp., "InTouch" Man-Machine Interface - Application Generator, User Guide and Reference Manual, Rev D, May 1993.

APÊNDICE I

CONJUNTO DE CARACTERES SEMI GRÁFICOS

