



Prático 2 - Supervisório

Este trabalho deve ser feito por uma equipe de dois alunos.

Data de entrega: 24/06/2005 através de e-mail para o site: seixas@task.com.br

Deve ser enviado um arquivo .zip contendo os programas fontes e o executável.

O programa deverá ser codificado em Delphi ou em Visual C++

Você vai desenvolver uma aplicação que utilizará todos os recursos ensinados no curso.

Inicialmente desenvolva um processo denominado CLP para simular um Controlador Lógico Programável. O seu processo irá abrir uma janela de configuração para a definição de parâmetros de até 10 variáveis analógicas. Para cada variável o usuário pode escolher o tag (12 caracteres), o limite inferior (LIM_INF), o limite superior (LIM_SUP), os limites baixo (L), muito baixo (LL), alto (H) e muito alto (HH).

Também deverá definir um modelo de simulação para os valores da variável (PV). O usuário escolherá entre: ONDA_QUADRADA, RAMPA, ONDA_SENOIDAL ou VALOR_CONSTANTE. Para todos os sinais deverá determinar a amplitude e para os sinais periódicos determinar (período e amplitude).

Variável	TAG	LIM_INF	LL	L	H	HH	LIM_SUP
1	12-TT-02	20	80	110	450	480	930
2							
3							
...							
10							

Em seguida desenvolva o processo MONITOR.

Monitor através da thread INTERF_CLP irá enviar um comando de UPLOAD para CLP e CLP enviará para ele a tabela com as definições das variáveis analógicas.

Em seguida a INTERF_CLP irá pedir a cada período programável (default = 1s) a leitura de todas as PVs de CLP. Estes valores serão salvos em uma memória compartilhada. Será lido a cada transação, o índice da variável, o valor da PV e o time_stamp.

A comunicação entre MONITOR e CLP será através de um pipe com nome.

Outra thread de MONITOR denominada GERA_ALARM irá checar todos os valores das variáveis e ver se algum limite de alarme foi excedido. Neste caso GERA_ALARM irá formatar uma mensagem de alarme contendo data e hora, nome da variável, e descrição do alarme e irá enviar uma mensagem para o Mailslot "Alarm". Use o componente dado na apostila de Delphi para implementar Mailslots.

A thread/processo ALARME irá receber as mensagens neste mailslot, formatar uma mensagem e exibi-la em uma janela. ALARME irá disparar uma buzina até que todos os alarmes seja reconhecidos pelo operador. Um alarme não reconhecido é mostrado na janela em vermelho e quando reconhecido em verde. Use um botão para reconhecer todos os alarmes. Se quiser implementar um reconhecimento um a um dos alarmes, será recompensado.

Uma thread/processo chamada TREND irá ler periodicamente os valores das variáveis da tabela compartilhada em memória e irá traçar um gráfico de tendência das variáveis. O gráfico pode ser individual ou permitir o uso de até 8 penas para exibição no mesmo gráfico, você pode escolher. De toda forma o usuário pode visualizar quaisquer das variáveis para visualização. Você pode utilizar algum componente pré existente para traçar o gráfico de tendência.

Concentre todas as iterações de MONITOR com o usuário em uma thread IHM. IHM irá se comunicar com TREND através de mensagens padrão Windows (WM_USER). IHM será a thread principal que criará todas as demais.

Utilize timers para realizar as temporizações.

Quando o usuário quiser encerrar a simulação deverá acionar a tecla ESCAPE e toda a simulação terminará graciosamente.

Desenhe um diagrama didático mostrando todos os processos/threads, os mecanismos de comunicação adotados e os formatos de cada mensagem, como feito em sala de aula. Use este diagrama para guiar o desenvolvimento do seu projeto.

Todo desenvolvimento extra será recompensado com pontos de bônus.