

BOLETIM DE INFORMAÇÕES TÉCNICAS
BIT 17 - CA100.DLL / CA600.DLL
Atualização: 20/10/2000

1. INTRODUÇÃO

A maioria dos desenvolvedores utilizam atualmente uma linguagem visual para criar os aplicativos para as plataformas Windows e acessar bancos de dados relacionais. A CA100.DLL e a CA600.DLL são ferramentas especialmente desenvolvidas para este tipo de usuários.

A CA100.DLL é uma biblioteca de funções que podem ser chamadas a partir de uma outra aplicação. Foi desenvolvida para comunicação de dados entre um computador e um terminal ou catraca da PASSO com interpretador UACLIP, através de uma única serial do computador, tanto em conexão RS-485 ou RS-232.

A CA600.DLL é uma versão equivalente utilizada quando o meio de comunicação é por rádio frequência. Todas as informações fornecidas para a CA100.DLL são válidas para a CA600.DLL.

2. CARACTERÍSTICAS

As funções das DLL's podem ser chamadas de qualquer programa desenvolvido para plataforma Windows 32 bits (Windows 95, Windows 98, Windows NT).

Para auxiliar os desenvolvedores na utilização desta DLL estão disponíveis na internet exemplos nas linguagens Visual Basic e Delphi. É importante destacar que o Visual Basic trabalha com 32 bits a partir da versão 5.0 e o Delphi a partir da versão 2.0.

Outras linguagens compatíveis com as características descritas acima podem ser utilizadas para importar as funções da DLL. Como por exemplo Cobol, versão para 32 bits e C++ Builder.

3. CONFIGURAÇÃO DA REDE

Para configurar a rede deve ser utilizado o programa CF100.EXE ou WCF100.EXE, mesmo quando trabalhando em conexão RS-232. Estes programas geram o arquivo PC100.CFG que contém as informações necessárias para funcionamento dos equipamentos ligados à rede URANET. Os itens que devem ser configurados são "Terminais", "Comunicação" e "Aplicativos". Os outros itens são desconsiderados pela CA100.DLL.

4. A APLICAÇÃO PARA OS TERMINAIS/CATRACAS

Para comunicar o terminal carrega um programa em memória, desenvolvido na linguagem UACLIP, esta linguagem tem uma estrutura parecida com o Clipper com algumas funções especialmente desenvolvidas para os equipamentos da PASSO.

Numerosos exemplos são fornecidos em UACLIP nos quais os programadores podem se basear para desenvolver as suas aplicações. Sempre que houver dúvidas sobre o uso de alguma função nesta linguagem é possível contar com a ajuda do suporte técnico da PASSO.

Utilizando esta DLL é possível utilizar todas as funções dos equipamentos sem perder as vantagens de uma CPU inteligente que permite leitura de cartões de barras, scanner; comunicação

com balanças, impressoras; etc. As funções de comunicação com o PC ficam limitadas nas seguintes: FREAD, FWRITE, DATE e TIME.

As funções FREAD e FWRITE implementadas para escrita e leitura em arquivo texto no PC100 são utilizadas pela DLL para troca de dados entre os microterminais/catracas e a aplicação que roda no PC.

5. A APLICAÇÃO NO PC

O uso e chamada das funções da DLL estão descritos nos exemplos em Delphi e em VB, existentes na seção utilitários da página da Passo na internet ().

Estes exemplos descrevem como os dados provenientes dos terminais deverão ser lidos/escritos utilizando as funções ReadFromTerm e WriteToTerm da DLL.

A troca de informações através de strings permite ao desenvolvedor criar um protocolo próprio dando-lhe total autonomia e possibilitando o uso de todos os recursos da linguagem utilizada como também o acesso a qualquer banco de dados com o qual esteja familiarizado.

A DLL também fornece informações sobre problemas na porta de comunicação através do uso de funções descritas nos exemplos.

Se o desenvolvedor utilizar Visual Basic, podem ocorrer erros ao se utilizar o botão stop da barra de ferramentas, quando o programa estiver sendo rodado dentro do ambiente. Se este problema ocorrer, é necessário criar um botão ou menu para finalizar a aplicação, o qual deverá chamar a função EndCom antes do Unload Me e End, e finalizar a aplicação sempre por ali, não utilizando mais o botão Stop.

6. FUNÇÕES DA DLL

long StartCom():

OBJETIVO: Abre a porta de comunicação configurada no pc100.cfg e cria os encadeamentos de comunicação, que iniciam a varredura dos microterminais em background.

ENTRADA: vazio.

SAÍDA: Zero caso bem sucedida
Código do erro se mal sucedida.

long EndCom():

OBJETIVO: Finaliza os encadeamentos de background e fecha a porta de comunicação.

ENTRADA: vazio.

SAÍDA: Zero caso bem sucedida.
Código do erro se mal sucedida.

long SendToTerm(inteiro iTerm, string de 200 caracteres Buffer):

OBJETIVO: Envia para o buffer de saída a string a ser enviada para o terminal definido pelo primeiro parâmetro da função.

ENTRADA: iTerm=>Ordem em que o terminal aparece no arquivo de configuração pc100.cfg, conforme declaração feita no software de configuração cf100 ou wcf100.
Buffer=>Ponteiro para string a ser transmitida.

SAÍDA:

Retorna um inteiro longo com o número de bytes enviados se bem sucedida, caso contrário retorna EOF.

long ReadFromTerm(inteiro iTerm, String de 200 caracteres Buffer):

OBJETIVO: Fornece para a aplicação a última string transmitida pelo terminal armazenada no buffer de recepção.

ENTRADA: iTerm=>Ordem em que o terminal aparece no arquivo de configuração pc100.cfg, conforme declaração feita no software de configuração cf100 ou wcf100.
Buffer=>Ponteiro para string recebida do terminal.

SAÍDA:

Retorna um inteiro longo com o tamanho em bytes da string enviada pelo terminal.

long Suspende():

OBJETIVO: Paralisa a execução dos encadeamentos de background.

ENTRADA: vazio.

SAÍDA: Zero caso bem sucedida.
Código de erro se mal sucedida.

long Ativa():

OBJETIVO: Ativa novamente encadeamentos suspensos pela função Suspende.

ENTRADA: vazio.

SAÍDA: Zero caso bem sucedida.
Código de erro se mal sucedida.

long CommError():

OBJETIVO: Essa função é válida apenas para a ca100.dll.

Indica se houve erro na comunicação serial, lido do registrador LSR da porta de comunicação.

A utilização desta função é importante para detectar possíveis problemas de rede, possibilitando que sejam corrigidos antes que estes cheguem a um nível que torne impossível a operação do sistema.

Em placas mãe com BIOS Award, aconselha-se que a cada erro de comunicação encontrado a porta de comunicação seja fechada e aberta novamente através das funções EndCom e StartCom.

O arquivo CommErros.txt, no diretório da aplicação, fornece informações mais detalhadas sobre os erros: tipo de erro ocorrido, o número do terminal em que ele ocorreu, e a data e hora em que ocorreu.

ENTRADA: vazio.

SAÍDA:

0 - Nenhum erro.

Maior que zero - Erros detectados

long NTerm():

OBJETIVO: retorna o número de terminais declarados no arquivo de configuração.

ENTRADA: vazio

SAÍDA: número de terminais declarados no arquivo de configuração PC100.CFG.
zero se a porta de comunicação ainda não foi aberta.

7. ARQUIVOS GERADOS PELA DLL:

7.1. CommErrors.txt:

Esse arquivo é gerado apenas pela ca100.dll e no mesmo são registrados todos os erros detectados pela porta de comunicação do PC.

O primeiro campo indica o tipo de erro, o segundo indica o número do terminal em comunicação no momento da ocorrência do erro, e finalmente tem-se a data e a hora em que o erro ocorreu.

Exemplo: CE_FRAME 00200 Wed Oct 18 17:33:40 2000

7.2 Reini.txt:

Esse arquivo é gerado apenas pela ca600.dll e no mesmo são registradas todas as reinicializações da porta de comunicação do PC.

A porta será reinicializada caso a dll detecte algum mal funcionamento na mesma.

Exemplo: Reiniciou - Tue Oct 17 09:01:14 2000

7.3 Excecoes.txt:

Nesse arquivo serão registradas quaisquer exceções detectadas e tratadas durante a execução da dll, bem como a data e hora de sua ocorrência.