



MANUAL

TEC44Dis DLL

Versão 1.0 32 bits - 17/10/99

Para Windows 95/98

Índice

1- INTRODUÇÃO	
2- DOCUMENTAÇÃO	1
2.1 Documentação de Referência	1
3- INSTALAÇÃO DO DRIVER	
4- DESCRIÇÃO FUNCIONAL	
4.1 Interface DLL	1
4.2 Mensagem do Windows	4
4.3 Ambiente Operacional	4

1- INTRODUÇÃO

Basicamente este produto, **TEC44Dis DLL Versão 1.0**, é uma DLL (Dynamic Link Library) que provê acesso às funcionalidades do TEC44Dis tais como: movimentar o cursor no LCD, apresentar caracteres, leitura de cartão magnético e do teclado e chave para seleção de acesso. Pode ser utilizada com qualquer tipo de compilador desde o Visual Basic até o Delphi passando pelo Borland C++ e Visual C.

2- DOCUMENTAÇÃO

2.1 Documentação de Referência

Para referências às características do teclado TECD65 consulte a respectiva documentação técnica contida no disquete que acompanha o produto.

3- INSTALAÇÃO DO DRIVER

O procedimento para a instalação do driver deve seguir os seguintes passos:

1 - Abra uma janela DOS.

2 - Copie o arquivo **WINDRVR.VXD** para o diretório **C:\WINDOWS\SYSTEM\VMM32**

Por exemplo:

copy [dir]\WINDRVR.VXD C:\WINDOWS\SYSTEM\VMM32

3 - Execute o aplicativo **WDREG.EXE** para que o Windows reconheça este driver no proximo boot. Utilize o aplicativo da seguinte maneira:

WDREG install

4 - Feche a janela DOS.

5 - Reinicie a sua maquina.

A partir deste momento já é possível utilizar a DLL para o acesso ao TEC44Dis através do driver instalado.

4- DESCRIÇÃO FUNCIONAL

4.1 Interface DLL

A seguir estão listadas as funções que podem ser utilizadas pelo programador para acesso as funcionalidades do TEC44Dis. É importante observar que o programador deve seguir uma seqüência para que o aplicativo possa ter total acesso à DLL e ao TEC44Dis. Esta seqüência deve ser: chamar a função **OpenTec44()** logo após carregar a DLL e chamar **CloseTec44()** logo antes de sair da aplicação. Não existem restrições às outras funções dentro da DLL.

1) *UINT __stdcall OpenTec44 (void);*

Habilita o TEC44Dis para início de operação. Deve ser chamada no início do aplicativo e antes de qualquer outra função.

2) *void __stdcall CloseTec44 (void);*

Termina a operação com o TEC44Dis. Deve ser chamada ao término do aplicativo.

3) *void __stdcall DispStr (LPSTR Str);*

onde :

Str = Ponteiro para a string a ser enviada

Envia uma string de terminação nula para o display na posição atual do cursor.

4) *void __stdcall DispCh (const char C);*

onde :

C = caracter a ser enviado para o display.

Envia um caracter para o display.

5) *void __stdcall GoToXY (const UINT Lin, const UINT Col);*

onde :

Lin = Nova linha para onde posicionar o cursor (deve estar na faixa [1,2]).

Col = Nova coluna para onde posicionar o cursor(deve estar na faixa [1,40]).

Move o cursor do display para a posição (Lin, Col).

6) *void __stdcall BackSpace (void);*

Apaga o caracter anterior à posição atual do cursor do display e desloca o cursor para esta posição.

7) *void __stdcall LineFeed (void);*

Posiciona o cursor do display no início da próxima linha, fazendo rolamento do display se estiver na segunda linha.

8) *void __stdcall FormFeed (void);*

Limpa o display e posiciona o cursor na linha 1, coluna1.

9) *void __stdcall CarRet (void);*

Posiciona o cursor do display no início da linha.

10) *void __stdcall SetPIN (const UINT OnOff);*

onde :

OnOff = indica o estado do comando (1 = ON , 0 = OFF)

Habilita/Desabilita o funcionamento do PIN;

11) *void __stdcall GetMCRMsg (LPINT MCRMsg);*

onde :

MCRMsg = recebe o valor retornado por RegisterWindowMessage ("Tec44 - MCR Message").

Permite ao usuário obter o número da mensagem que ele deve interceptar no seu loop de mensagens do windows de forma a saber que foi realizada uma leitura do cartão magnético.

12) void __stdcall GetMCRBuf(LPSTR ReadBuf);

onde :

ReadBuf = recebe o buffer do MCR.

Permite ao usuário, logo após receber uma mensagem do MCR, obter a string enviada pelo cartão no modelo de teclado codificado, além de limpar o buffer do MCR.

Os caracteres da trilha 1 estarão entre colchetes antecidos do caracter "1" (Ascii 0X31). Os caracteres da trilha 2 estão da mesma forma só que antecidos pelo caracter "2" (Ascii 0x32).

Exemplo: String do cartão: "[1xxxxxx][2yyyyyy]", onde x e y são os caracteres de cada trilha.

Se MCR retornar em ReadBuf "[1VK_F12]", "[2VK_F12]" ou [1][2], indica um erro de leitura por parte do leitor de cartão magnético.

13) void __stdcall GetMCRBufnc(LPSTR ReadBuf);

onde :

ReadBuf = recebe o buffer do MCR.

Permite ao usuário, logo após receber uma mensagem do MCR, obter a string enviada pelo cartão no modelo de teclado não codificado, além de limpar o buffer do MCR.

Os caracteres da trilha 1 estarão entre colchetes antecidos do caracter "1" (Ascii 0X31). Os caracteres da trilha 2 estão da mesma forma só que antecidos pelo caracter "2" (Ascii 0x32).

Exemplo: String do cartão: "[1xxxxxx][2yyyyyy]", onde x e y são os caracteres de cada trilha.

Se MCR retornar em ReadBuf "[1VK_F12]", "[2VK_F12]" ou [1][2], indica um erro de leitura por parte do leitor de cartão magnético.

14) void __stdcall Reset (void);

Força a reinicialização do teclado.

15) void __stdcall SetCard (const UINT OnOff);

onde :

OnOff = indica o estado do comando (1 = ON , 0 = OFF)

Habilita/Desabilita o funcionamento do leitor de cartão magnético;

16) void __stdcall ProgSpecialChar (UINT CharSpecialNumber, LPINT ProgData);

onde :

CharSpecialNumber = número do caracter especial (deve ser na faixa [0,7]).

ProgData = vetor contendo os dados a fim de programar o caracter especial.

Permite a programação de um dos 8 caracteres especiais.

Os caracteres apresentados no display são constituídos por células (matrizes) de 5 x 7 pontos (horizontal x vertical).

As linhas das matrizes dos caracteres são formadas, respectivamente, de cima para baixo, pelos bytes desde ProgData [0] até ucProgData [6], onde os pontos visíveis são representados pelos seus bits setados (atribuídos o valor 1), porém apenas os 5 bits menos significativos são utilizados.

Para ver o cursor (sublinhado) sob este caracter, ucProgData [7] deve ser zero.

17) void __stdcall SetNumL (void);

Ativa/Desativa o led de NumLock do teclado mudando seu estado também no sistema.

18) void __stdcall SetCapsL (void);

Ativa/Desativa o led de CapsLock do teclado mudando seu estado também no sistema.

19) void __stdcall BeepOn (const UINT OnOff);

onde :

OnOff = indica o estado do comando (1 = ON , 0 = OFF)

Ativa/Desativa o "beep" do teclado.

20) void __stdcall BeepKeyOn (const UINT OnOff);

onde :

OnOff = indica o estado do comando (1 = ON , 0 = OFF)

Ativa/Desativa o "beep" individual das teclas.

21) void __stdcall DispClrLn1 (void);

Apaga toda a linha 1 do display.

22) void __stdcall DispClrLn2 (void);

Apaga toda a linha 2 do display.

23) void __stdcall SetOpKey (const UINT OnOff);

onde :

OnOff = indica o estado do comando (1 = ON , 0 = OFF)

Ativa/Desativa o funcionamento da chave do operador. Esta função não esta disponível no tec44Dis, pois o mesmo não tem chave de operador.

24) void __stdcall ReadOpKey (LPSTR KeyBuf);

onde :

KeyBuf = Recebe o buffer da chave com o valor de sua posição atual.

Esta string tem o formato semelhante ao cartão magnético. O seu valor atual estará entre colchetes antecedido pelo caracter "t" (Ascii 0x74). Esta função não esta disponível no Tec44Dis, pois o mesmo não tem chave de operador.

Exemplo: [tx], onde x é o caracter da posição da chave.

25) void __stdcall SetEcho (const UINT OnOff, const UINT Password);

onde :

OnOff = indica o estado do comando (1 = ON , 0 = OFF)

Password = indica o estado de senha (1 = ON , 0 = OFF)

Estabelece que todo o caracter teclado também vai para o display. Se a variável Password estiver ligada, o caracter ecoado será o asterisco " * ".

26) void __stdcall EnableKbd (void);

Permite o funcionamento normal do teclado. O sistema recebe as teclas normalmente.

27) void __stdcall DisableKbd (void);

Inibe o teclado. Impede que as teclas cheguem ao sistema.

4.2 Mensagens de Windows

Logo após o usuário passar o cartão magnético no MCR, a DLL gera uma mensagem do Windows, referente à leitura do MCR. Para se ler as trilhas do cartão, basta chamar a função GetMCRBuf() para teclados codificados ou GetMCRBufnc() para modelos não codificados. Vários caracteres serão recebidos pela aplicação durante a leitura do cartão. Estes caracteres devem ser ignorados.

Mensagem do leitor de cartão magnético ==> obter o valor da mensagem com GetMCRMsg

Parâmetros da mensagem :

Esta mensagem não tem parâmetros.

4.3 Ambiente Operacional

Como requerimentos mínimos de hardware e software é necessário ter:

PC 486DX2 66mhz com 16 Mb de memória;

Monitor SVGA;

Windows 95.