



# **TEC65\_32 DLL**

**VERSÃO 1.1.1999 (Gertec) - 25/08/1999**

## **Sumário**

- 1. INTRODUÇÃO
- 2. DOCUMENTAÇÃO
  - 2.1. Documentação de Referência
- 3. DESCRIÇÃO FUNCIONAL
  - 3.1. Interface DLL
  - 3.2. Mensagem do Windows
  - 3.3. Ambiente Operacional

## 1. INTRODUÇÃO

Este produto, **TEC65\_32.DLL**, é uma *DLL (Dynamic Link Library)* que provê acesso à funcionalidade do TECD65: *beeps* (contínuo ou de tecla); movimentação do cursor no display de cristal líquido (LCD); apresentação de caracteres; leitura de cartão magnético, das mudanças na chave de operador e do teclado.

## 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. DESCRIÇÃO FUNCIONAL

### 3.1. Interface DLL

**NOTA:** Será apresentada primeiramente a definição de importação da função em C++ e entre parênteses, na outra linha, em Visual Basic.

a) *UINT \_\_declspec(dllimport) \_\_stdcall OpenTec65 (void)*  
(Declare Function OpenTec65 Lib "tec65\_32.dll" () As Integer)

Habilita a DLL do TECD65 para início de operação. Deve ser chamada no início do aplicativo.

b) *void \_\_declspec(dllimport) \_\_stdcall CloseTec65 (void)*  
(Declare Sub CloseTec65 Lib "tec65\_32.dll" () )

Termina a operação com o TECD65 (desconecta a *DLL*). Deve ser chamada ao término do aplicativo.

c) *void \_\_declspec(dllimport) \_\_stdcall SetDisp (const UINT OnOff)*  
(Declare Sub SetDisp Lib "tec65\_32.dll" (ByVal OnOff As Integer) )

onde :

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

Os caracteres enviados ao TECD65 quando este comando estiver configurado para *ON* serão mostrados no display.

d) *void \_\_declspec(dllimport) \_\_stdcall TxKbd (BYTE dado)*  
(Declare Sub TxKbd Lib "tec65\_32.dll" (ByVal dado As Byte) )

onde :

*dado* = *Byte* a ser transmitido ( 0x00 a 0xFF ).

Transmite um determinado *byte* para o teclado.

e) *void \_\_declspec(dllimport) \_\_stdcall DispStr (LPSTR str)*  
(Declare Sub DispStr Lib "tec65\_32.dll" (ByVal str As String) )

onde :

*str* = Ponteiro para a *string* a ser enviada

Envia uma *string* de terminação nula (conjunto de caracteres) para o display.

f) void \_\_declspec(dllimport) \_\_stdcall DispCh (const char c)  
(Declare Sub DispCh Lib "tec65\_32.dll" (ByVal char As Byte) )

onde :

c = caracter a ser enviado para o display.

Envia um caracter para o display.

g) void \_\_declspec(dllimport) \_\_stdcall GoToXY (const UINT Col, const UINT Lin)  
(Declare Sub GoToXY Lib "tec65\_32.dll" (ByVal Col As Integer, ByVal Lin As Integer) )

onde :

Col = Nova coluna para onde posicionar o cursor (deve estar na faixa [1,16]).

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

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

h) void \_\_declspec(dllimport) \_\_stdcall BackSpace (void)  
(Declare Sub BackSpace Lib "tec65\_32.dll" () )

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

i) void \_\_declspec(dllimport) \_\_stdcall LineFeed (void)  
(Declare Sub LineFeed Lib "tec65\_32.dll" () )

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

j) void \_\_declspec(dllimport) \_\_stdcall FormFeed (void)  
(Declare Sub FormFeed Lib "tec65\_32.dll" () )

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

k) void \_\_declspec(dllimport) \_\_stdcall CarRet (void)  
(Declare Sub CarRet Lib "tec65\_32.dll" () )

Posiciona o cursor do display no início da linha.

l) void \_\_declspec(dllimport) \_\_stdcall SetPIN (const UINT OnOff)  
(Declare Sub SetPIN Lib "tec65\_32.dll" (ByVal OnOff As Integer) )

onde :

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

Habilita/Desabilita o funcionamento do PIN (acessório opcional).

m) void \_\_declspec(dllimport) \_\_stdcall GetMCRMsg (LPINT MCRMsg)  
(Declare Sub GetMCRMsg Lib "tec65\_32.dll" (MCRMsg As Any) )

onde :

*MCRMsg* = recebe o valor retornado por *RegisterWindowMessage* ("TEC65 - 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.

n) void \_\_declspec(dllimport) \_\_stdcall GetMCRBuf(LPSTR ReadBuf)  
(Declare Sub GetMCRBuf Lib "tec65\_32.dll" (ReadBuf As Any) )

onde:

*ReadBuf* = recebe o *buffer* do cartão (*MCR – magnetic card reader*). Se *MCR* retornar "[E]", *ReadBuf* [0] = 0, pois indica um erro de leitura por parte do leitor de cartão magnético.

Permite ao usuário, logo após receber uma mensagem do *MCR*, obter a *string* enviada pelo cartão, além de limpar o *buffer* do *MCR*.

o) void \_\_declspec(dllimport) \_\_stdcall Reset (void)  
(Declare Sub Reset Lib "tec65\_32.dll" () )

Força a reinicialização e auto-teste do teclado.

p) void \_\_declspec(dllimport) \_\_stdcall SetCard (const UINT OnOff)  
(Declare Sub SetCard Lib "tec65\_32.dll" (ByVal OnOff As Integer) )

onde :

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

Habilita/Desabilita o funcionamento do leitor de cartão magnético (*MCR*).

q) void \_\_declspec(dllimport) \_\_stdcall ProgSpecialChar (UINT CharSpecialNumber, LPINT ProgData)  
(Declare Sub ProgSpecialChar Lib "tec65\_32.dll" (ByVal CharSpecialNumber As Integer, ProgData As Any) )

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 *ProgData*[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, *ProgData*[7] deve ser zero.

r) void \_\_declspec(dllimport) \_\_stdcall SetNumL (const UINT OnOff)  
(Declare Sub SetNumL Lib "tec65\_32.dll" (ByVal OnOff As Integer) )

onde:

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

Altera o estado atual do *led* de NumLock do teclado.

s) void \_\_declspec(dllimport) \_\_stdcall SetCapsL (const UINT OnOff)  
(Declare Sub SetCapsL Lib "tec65\_32.dll" (ByVal OnOff As Integer) )

onde :

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

Altera o estado atual do *led* de CapsLock do teclado.

### **3.2.\_Mensagens do Windows**

Logo após o usuário passar o cartão magnético no *MCR*, o módulo **TEC65\_32.DLL** gera uma mensagem do Windows, referente à leitura do MCR :

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

Parâmetros da mensagem:

wParam = 0 ==> ou o cartão magnético possui uma cadeia vazia gravada ou o MCR retornou código de erro de leitura.

wParam <> 0 ==> o MCR recebeu a *string* com o formato esperado, i.e., "[<wParam dígitos>]" (no máximo, 150 caracteres).

### **3.3\_Ambiente Operacional**

Como requisitos mínimos de *hardware* e *software*, temos:

PC 486DX2 66MHZ com 16 MB de memória RAM  
Monitor SVGA  
Windows 95