

- Funções / Comandos DLL V.9 _ Delphi para terminais TED famílias TC e TX

- Fonte de aplicação “SOMACALC” (calculadora)

1 - Host (aplicação) para os TED's “saída” _

Lista de comandos / funções :

- Abrindo comunicação _

function TEDStart(port: integer): integer; external DLL_LOCATION;

- Enviando mensagem ao display TED _

function TEDDisplaySend(ip, data: string): integer; external DLL_LOCATION;

- Enviando alarmes de buzzer (n beep's) ao TED _

function TEDBeep(ip: string; times: integer): integer; external DLL_LOCATION;

- Enviando dados à saída COM serial 1 do TED _

function TEDCom1Send(ip, data: string): integer; external DLL_LOCATION;

- Enviando dados à saída COM serial 2 do TED _

function TEDCom2Send(ip, data: string): integer; external DLL_LOCATION;

- Enviando comando de apagar display do TED _

function TEDDisplayClear(ip: string): integer; external DLL_LOCATION;

- Enviando comando de apagar lista de atalhos na paginação (veja comando abaixo) _

function TEDListClear(ip: string): integer; external DLL_LOCATION;

- Enviando lista de atalhos (sequência) para paginação _

function TEDListAdd(ip: string; data: string): integer; external DLL_LOCATION;

obs. Os TED's da família TX com visor (display) de 7 linhas, permitem paginação com 7 atalhos por página, até 4 páginas. Usando este comando, pode-se programar até 28 atalhos (acionados pelas teclas Enter+seta) estacionados na memória interna do TED ... veja explo. a seguir:

function TEDListAdd('192.168.0.10', 'PRODUTO A|PRODUTO B|PRODUTO C|' até 28)

- Enviando comando para acionamento de “headers” – preâmbulos das tarefas _

function TEDHeaderOn(ip: string): integer; external DLL_LOCATION;

- Enviando comando para desacionamento de “headers” – preâmbulos das tarefas _

function TEDHeaderOff(ip: string): integer; external DLL_LOCATION;

- Enviando comando de “on” (nível de saída = 1 – Vcc) da saída digital _

function TEDDigitalOn(ip: string): integer; external DLL_LOCATION;

- Enviando comando de “off” (nível de saída = 0 – gnd) da saída digital _

function TEDDigitalOff(ip: string): integer; external DLL_LOCATION;

- Enviando comando de leitura do status (nível 1 ou 0) da entrada digital _

function TEDDigitalRead(ip: string): integer; external DLL_LOCATION;

DLL_LOCATION _ ligação com as funções de mesmo nome que estão DENTRO da DLL.

2 - TED's para o Host (aplicação) “entrada” _

A recepção do Host (aplicação) vinda do TED – teclado alfanumérico / teclas de atalho via paginação / leitor de cód. de barras externo – USB / leitor cód. de barras interno (embutido) / leitor RFID interno (embutido) / porta serial 1 / porta serial 2 / entrada digital

```
procedure onTEDReceive(hnd: TTEDReceiveEvent); external DLL_LOCATION;
```

o que chega quando algo é recebido:

```
procedure recebido(ip, data: String; header: Integer);
```

```
begin
```

```
    Form1.RunProgram(ip, data);
```

```
end;
```

A partir das funções descritas, realizar a ligação _

```
procedure onTEDConnect(hnd: TTEDConnectEvent); external DLL_LOCATION;
```

```
procedure onTEDReceive(hnd: TTEDReceiveEvent); external DLL_LOCATION;
```

onTedConnect informa o que chega na porta 55555 e o onTedReceive informa o que chega na porta 8 (ou outra porta configurada antecipadamente no menu de configuração do TED)

3 – TRATAMENTO DAS PORTAS SERIAIS

- HABILITAR LEITURA DA PORTA SERIAL COM1

Assinatura da função

```
function TEDCom1On(ip: string): integer; external DLL_LOCATION;
```

Exemplo de uso

```
TEDCom1On('192.168.10.1');
```

- DESABILITAR LEITURA DA PORTA SERIAL COM1

Assinatura da função

```
function TEDCom1Off(ip: string): integer; external DLL_LOCATION;
```

Exemplo de uso

```
TEDCom1Off('192.168.10.1');
```

- HABILITAR LEITURA DA PORTA SERIAL COM2

Assinatura da função

```
function TEDCom2On(ip: string): integer; external DLL_LOCATION;
```

Exemplo de uso

```
TEDCom2On('192.168.10.1');
```

- DESABILITAR LEITURA DA PORTA SERIAL COM2

Assinatura da função

```
function TEDCom2Off(ip: string): integer; external DLL_LOCATION;
```

Exemplo de uso

```
TEDCom2Off('192.168.10.1');
```

- EXEMPLO SOMACALC

Obs.: No explo. de aplicação **SOMACALC** (calculadora) incluído nos anexos, são nomeados **conectado** e **recebido** conforme abaixo:

```
unit USoma;
```

```
interface
```

```
uses
```

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,  
Vcl.Graphics,
```

```
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, System.Generics.Collections,
```

```
Vcl.StdCtrls;
```

```
const
```

```
DLL_LOCATION = 'ted.dll';
```

```
//DLL_LOCATION = 'C:\Projetos\Colleter\DLL\ted.dll';
```

```
TEDStartMessages: array [0..1] of string = (
```

```
'Erro na abertura da porta!'
```

```
'Porta aberta com sucesso!'
```

```
);
```

```
type
```

```
TForm1 = class(TForm)
```

```
Label1: TLabel;
```

```
GroupBox4: TGroupBox;
```

```
Label4: TLabel;
```

```
Porta: TEdit;
```

```
Button1: TButton;
```

```
lblMensagem: TLabel;
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure FormCreate(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
procedure RunProgram(ip, data: String);
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
TTEDConnectEvent = procedure(ip: string);
```

```
TTEDReceiveEvent = procedure(ip, data: string; h dr: Integer);
```

```
function TEDStart(port: integer): integer; external DLL_LOCATION;
```

```
function TEDBeep(ip: string; times: integer): integer; external DLL_LOCATION;
```

```
function TEDDisplaySend(ip, data: string): integer; external DLL_LOCATION;
```

```
function TEDCom1Send(ip, data: string): integer; external DLL_LOCATION;
```

```
function TEDCom2Send(ip, data: string): integer; external DLL_LOCATION;
```

```
function TEDDisplayClear(ip: string): integer; external DLL_LOCATION;
```

```
function TEDListClear(ip: string): integer; external DLL_LOCATION;
```

```
function TEDListAdd(ip: string; data: string): integer; external DLL_LOCATION;
```

```
function TEDHeaderOn(ip: string): integer; external DLL_LOCATION;
```

```
function TEDHeaderOff(ip: string): integer; external DLL_LOCATION;
```

```
function TEDDigitalOn(ip: string): integer; external DLL_LOCATION;
```

```
function TEDDigitalOff(ip: string): integer; external DLL_LOCATION;
```

```
function TEDDigitalRead(ip: string): integer; external DLL_LOCATION;
```

```
procedure onTEDConnect(hnd: TTEDConnectEvent); external DLL_LOCATION;
```

```
procedure onTEDReceive(hnd: TTEDReceiveEvent); external DLL_LOCATION;
```

```
var
```

```
Form1: TForm1;
```

```
TEDs: TDictionary<String, Integer>; // IP + ID DO COMANDO ATUAL PARA O TED
```

```
Buffers: TDictionary<String, Integer>; // IP + DADO DO COMANDO ATUAL PARA O TED
```

```
implementation
```

```
{ $R *.dfm }
```

```
{ TForm1 }
```

```
procedure conectado(ip: String);
```

```
var
```

```
id: Integer;
```

```
begin
```

```
if (TEDs.TryGetValue(ip, id) = True) then Exit;
```

```
TEDs.AddOrSetValue(ip, 0);
```

```
Form1.RunProgram(ip, "");
```

```
end;
```

```
procedure recebido(ip, data: String; header: Integer);
```

```
begin
```

```
Form1.RunProgram(ip, data);
```

```
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    ret: integer;  
begin  
    onTEDConnect(conectado);  
    onTEDReceive(recebido);  
    ret := TEDStart(StrToIntDef(Porta.Text, 8));  
    lblMensagem.Caption := TEDStartMessages[ret];  
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    TEDs := TDictionary<String, Integer>.Create;  
    Buffers := TDictionary<String, Integer>.Create;  
end;
```

```
procedure TForm1.RunProgram(ip, data: String);  
var  
    id, a, b: integer;  
begin
```

```
    if (TEDs.TryGetValue(ip, id) = false) then Exit;
```

```
    if (id = 0) then
```

```
    begin
```

```
        TEDDisplayClear(ip);
```

```
        TEDBeep(ip, 1);
```

```
        TEDDisplaySend(ip, 'A:');
```

```
    end;
```

```
    if (id = 1) then
```

```

begin
  Buffers.AddOrSetValue(ip, StrToIntDef(data, 0));
  TEDBeep(ip, 1);
  TEDDisplaySend(ip, 'B:');
end;

```

```

if (id = 2) then
begin
  Buffers.TryGetValue(ip, a);
  b := StrToIntDef(data, 0);
  TEDBeep(ip, 1);
  TEDDisplaySend(ip, 'SOMA:' + IntToStr(a+b));
  TEDDisplaySend(ip, 'A:');
  id := 0;
end;

```

```

TEDs.AddOrSetValue(ip, id + 1);

```

```

end;

```

```

end.

```

Ressaltando que neste explo. **SOMACALC**

onTEDConnect e onTEDReceive espera cada um deles uma função para "LIGAR" e "RECEBER" os dados

São chamados **conectado** e **recebido**

```

procedure conectado(ip: String);

```

```

var

```

```

  id: Integer;

```

```

begin

```

```
if (TEDs.TryGetValue(ip, id) = True) then Exit;  
TEDs.AddOrSetValue(ip, 0);  
Form1.RunProgram(ip, "");  
end;
```

```
procedure recebido(ip, data: String; header: Integer);  
begin  
    Form1.RunProgram(ip, data);  
end;
```