

Manual de Programação DLLs TED versões TC / TD / PD

Índice:

Programação / Comandos	2
Abrindo a comunicação (abreip)	2
Enviando mensagens ao display (envtmsg)	3
Recebendo dados do terminal (getcmt)	4
Tocando bips (envbeep)	5
Apagando o display (envclisp)	5
Escrevendo numa impressora pela porta serial auxiliar (envtimp)	6
Lendo uma balança ligada na porta serial auxiliar (retorno RX2)	6
Fluxograma exemplo para vários terminais	7

Programação / Comandos:

Para enviar e receber mensagens para o TED versões UDP você pode utilizar a DLL sockdelp.dll (para Delphi) ou sockvbcl.dll (para VB). Os comandos e funções fáceis e intuitivos de serem usados.

Abrindo comunicação com um terminal com a função “abreip” :

Abreip (x) - abre comunicação com um endereço IP e associa a ele um número de canal

x -string - Endereo IP em formato string (ex: 192.168.0.51)

Retorno: inteiro (sempre 1)

Declaração :

```
function abreip ( porta :shortstring):integer;  
external 'sockdelp.dll'
```

Exemplo de uso da função:

```
x = abreip ('192.168.0.51');
```

Para cada chamada “abreip” será associado um número de canal (uma espécie de índice) para o IP. Assim, o primeiro IP aberto passará a funcionar como canal 0, o segundo será canal 1, o terceiro canal 2 e assim por diante.

OBS: no caso do uso de IP dinâmico, esta função é chamada automaticamente pela DLL.

Enviando uma mensagem para o display do TED - função ENVTMSG

Cada IP que foi aberto com o abreip é associado a um número de canal. O primeiro aberto é o canal 0, o segundo é o canal 1 e assim por diante. Para enviar uma mensagem para um TED, o TED deve ser selecionado pelo número do seu canal na função ENVTMSG

ENVTMSG (conj, can, STRING) - envia mensagem para o display do TED

Conj - inteiro - (sempre 0 usado apenas para manter compatibilidade com versões anteriores)

Canal - inteiro - endereço associado ao IP destinatário da mensagem.

STRING - variável string - dado a ser enviado para o display do TED

Declaração:

```
function envtmsg ( conj,can :integer ; da : shortstring ):integer;
```

```
external 'sockdelp.dll'
```

Utilização :

```
X:=envtmsg ( 0,0,'Teste de envio');
```

RETORNO (X=número inteiro)

Se X=1, o comando foi bem sucedido

Se X=0, houve algum problema de comunicação de socket.

Recebendo uma mensagem digitada ou leitura de código de barras.

GETCNT (ptSTRING, ptCAN, conj) - lê mensagem vinda de todos os TEDs. Esta função deve ser colocada dentro de um TIMER , com intervalo de 1 milissegundo.

Conj - inteiro - (Colocar sempre conjunto=0)

ptCan - ponteiro para um inteiro - canal de onde veio a mensagem

ptSTRING - ponteiro para variável string - mensagem vinda do TED.

A função GETCNT retorna um valor 0 se não veio dado de nenhum TED e retorna 1 se existe algum dado vindo de algum canal (TED)

Na variável ptCAN será injetado pela DLL o número do canal que forneceu o dado (endereço) e na ptString será injetado o dado vindo do TED, independente de ser uma digitação ou uma leitura de código de barras.

Declaração :

```
function getcnt ( var dado:shortstring ; var canal:integer ; conj : integer ): integer;
```

```
external 'sockdelp.dll'
```

Utilização: (o GETCNT deve ser colocado dentro de um timer com intervalo de 1 milissegundo)

```
procedure TForm1.Timer1Timer(Sender: TObject);
```

```
begin
```

```
conjunto:=0;
```

```
canal:=100; // valor arbitrário (será modificado e retornado pela função)
```

```
if (getcnt ( dado,canal,conjunto ))=1 then
```

```
begin
```

```
Form1.edit1.text:='Recebi: '+dado + ' do canal:' + nntostr(canal);
```

```
envtmsg (conjunto,canal,'Dado recebido');
```

```
envbeep (conjunto,canal,1);
```

```
end;
```

```
end;
```

```
RETORNO (X=número inteiro)
```

Se getcnt retornou 1, existe algum dado vindo de algum TED. A variável 'CANAL' receberá o valor do canal que enviou o dado. A string recebida será armazenada em 'dado'

Se getcnt retornou 0, não existe dado nenhum, vindo de nenhum TED.

Fazendo o TED associado ao canal "n" tocar bips

x= ENVBEEP (conj, can, N) - faz o TED associado ao canal 'CAN' tocar N beeps

Conj - inteiro - sempre zero

Canal -inteiro - canal associado ao TED com a função abreporta

N -inteiro - número de beeps para o TED executar (variando de 1 a 9)

Declaração :

```
function envbeep ( cnj,can,n :integer ):integer;
```

```
external 'sockdelp.dll'
```

Utilização :

```
X:=envbeep ( 0,3,5); { Ted associado ao canal 3 tocará 5 beeps simples }
```

Apagando o display do TED

ENVCLDISP (conj, can) apaga o display do TED associado ao canal CAN

Conj - inteiro - Sempre zero

Canal -inteiro - canal que receberá o comando

Declaração:

```
function envcldisp ( cnj,can:integer ):integer;
```

```
external 'sockdelp.dll'
```

Utilização :

```
X:=envcldisp ( 0,3 ); Apaga o display do Ted associado ao canal 3 (3o IP aberto com abreip)
```

Escrevendo na porta serial auxiliar (impressoras seriais)

ENVTIMP (conj, can, STRING) - envia uma string para a porta serial auxiliar do TED

Conj - inteiro - (sempre 0 usado apenas para manter compatibilidade com versões anteriores)

Canal - inteiro - endereço associado ao IP destinatário da mensagem.

STRING - variável string - dado a ser enviado para o display do TED

Declaração:

```
function envtimp ( cnj,can :integer ; da : shortstring ):integer;  
external 'sockdelp.dll'
```

Utilização :

X:=envtimp (0,0,'Teste de envio'+chr(10)+chr(13)); // imprime mensagem na impressora serial e pula de linha

RETORNO (X=número inteiro)

Se X=1, o comando foi bem sucedido

Se X=0, houve algum problema de comunicação de socket.

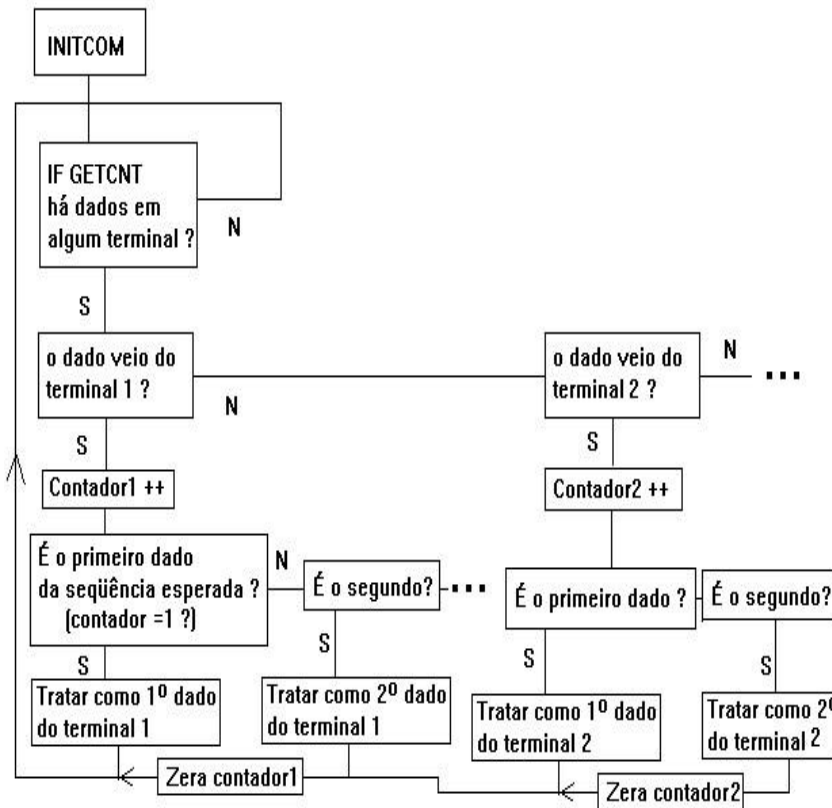
Lendo dados vindos da porta serial auxiliar (leitura de balanças)

Os terminais da família TED podem ler dados vindos pela porta serial auxiliar automaticamente. Não é necessário chamar um comando para ler o dado vindo pela linha RX do terminal. Quando chegar um dado vindo pelo RX, o terminal vai automaticamente enviar para o computador, como se tivesse sido digitado no teclado e pressionado Enter. Para ler o peso de uma balança serial que esteja configurada no protocolo ENQ, utilize a função “envtimp” para transmitir um código “asc 05” para a balança como feito no exemplo abaixo:

```
X:=envtimp ( 0,0, chr(05);
```

Com o comando acima será transmitido um ENQ para a balança (código ASC 05) e, assim, a balança irá transmitir o peso do prato para a linha RX. O dado será recebido pelo TED como se fosse uma digitação seguida de Enter.

Fluxograma exemplo de controle de vários microterminais na mesma rede



Obs : Note que só é necessária uma instrução _GETCNT, mesmo para vários terminais e vários dados de entrada.

Sugerimos que os contadores 1 e 2 do exemplo acima sejam elementos de uma matriz (ou vetor), indicando o status atual de cada terminal no sistema do usuário.