

## Manual exemplo para Micro-Terminais Gradual.

Descrição dos exemplos para comunicação com os Micro-terminais da Gradual. Estaremos utilizando o exemplo desenvolvido em Delphi para os terminais TCP-IP, porém o mesmo se aplica à versão em Visual Basic e terminais modelos RS-485.

Primeiramente recomendamos estar utilizando o protocolo VT100 é mais eficiente em caso de acesso aos periféricos, o protocolo Gradual realiza a escrita na porta RS232 e Paralela byte a byte e devolve uma confirmação de escrita. O protocolo VT100 trabalha com envio por pacotes de dados e sem retorno.

Na estrutura do programa três rotinas merecem atenção especial: Gget(), Gtrtecla() e TrataTerminais();

### ➤ **Rotina Gget();**

Atribui ao programa uma máscara de entrada de teclado, exemplo:

Gget('99999') seta apenas números com 5 dígitos.

Gget('AAAAAAAAAA') seta apenas alfa com 10 caracteres

Gget('XXXXXXXXXXXXXXXX') seta qualquer caracter ASC ou Alfa-numérico com 16 caracteres

### ➤ **Rotina GtrTecla();**

Trata o teclado conforme a máscara atribuída:

Gget('99999') - a função Gtrtecla ao receber um número do teclado atribui a variável "Teclado[Terminal]" se receber Alfa este será desprezado, pois não pertence à máscara. Veja o código abaixo:

```
function TForm1.Gtrtecla(IndConexao,Terminal:Integer):Boolean;  
begin
```

```
...
```

#### **If tecla = Chr(27) Then**

Trata o recebimento da tecla DEL, se a variável pospic[Terminal] for igual a 1 é porque não foi digitado nada então volta um estado (veremos depois o que vem a ser estado). Se for diferente é porque foi digitado algo, sendo assim limpa toda a digitação.

#### **If tecla = Chr(8) Then**

Trata a tecla backspace, enviando a string 08h + ' ' + 08h onde, 08h faz o retorno do cursor ' ' imprime um espaço em branco na posição em que se encontra o cursor, neste caso, em cima do último caracter digitado. 08h faz o retorno do cursor novamente para dar o efeito de apagamento.

#### **If tecla = Chr(13) Then**

Trata a tecla "Enter", função retorna verdadeiro informando que a máscara se completou. Suponha que tenha setado um Gget('9999999999999') 13 números para código de barras, por exemplo, o leitor de código insere ao final do código o 13h, desta forma se tiver um código com menos de 13 dígitos a função entende que se completou aquele campo. Da mesma forma se for digitar e o campo possuir 6 dígitos e não 13, ao teclar o enter o restante da máscara é anulada.

#### **if gpict = #3 then**

Trata o retorno de peso, por padrão as balanças ao receber 05h envia pela serial o peso, também por padrão a balança coloca a informação da seguinte forma 02h + Peso + 03h ou 02h + 20h + Peso + 20h + Preço + 20h + Total + 03h . Note que apesar das informações serem diferentes, ambas terminam com 03h, sendo assim ao receber este caracter sabe-se que todo peso foi recebido. Se a balança for do tipo 02h + Peso + 03h pode-se usar a máscara Gget('X99999X'). Veja também tópico sobre Balança.

#### **If gpict = 'X' Then**

Trata as teclas conforme a máscara

"9" - caracter numérico

"A" - caracter alfabético

"X" - qualquer caracter

"." - posição decimal ( apenas em campo numérico )

"/" - separador de campo( apenas em campo numérico )

"\*" - para senha

```
end;
```

➤ **Rotina TrataTerminais();**

Onde é feito todo o processamento. Primeiramente deve-se ter em mente que se pode ter mais que 1 equipamento conectado ao mesmo sistema e que em um dado momento o sistema estará recebendo informação de n-terminais e principalmente que estes acesso se darão em momentos diferentes e caminhos diferentes, por exemplo:

Menu Principal

- Cadastro
- Pesquisa
- Venda

Sendo assim um terminal pode estar fazendo uma pesquisa, outro efetuando uma venda, outro enviando dados para impressora enfim, situações diferentes. Para controlar de maneira eficiente a solução oferecida é trabalhar com uma “máquina de estado”, sempre que o sistema for acionado por um terminal a função TrataTerminais será chamada e executará conforme o estado em que se encontrar aquele terminal em específico.

Veja o código da rotina TrataTerminais();

**procedure TForm1.TrataTerminais(IndConexao,Terminal:Integer);  
begin**

**Case estado[Terminal] of**

- ◆ *No estado 0, o display é apagado, em seguida a string "Senha:" é impressa no display é setado a máscara de entrada de teclado e finalmente incrementado o estado.*

**0:begin**

```
GClear(IndConexao);  
Gdisplay(IndConexao,'Senha:');  
Gget(Terminal,'****');  
estado[Terminal]:=1;
```

**end;**

- ◆ No estado 1, pega-se o retorno do estado 0, note que se teclado “DEL”, volta ao estado anterior (estado 0) senão deverá fazer o tratamento inerente ao estado, ou seja, o programa no estado 0 solicita a senha quando Gtrtecla retornar True deve-se pegar Teclado[terminal] verificar na base de dados se a informação Usuário/Senha é válida e seguir para o próximo estado, no exemplo se True vai para estado 2 senão para estado 99. Este estado é um estado “morto”, apenas para que o usuário veja que as informações “Usuário” e “Senha” não são válidas e para que pressione uma tecla qualquer para voltar.

**1:begin**

```
If Gtrtecla(IndConexao,Terminal)Then
```

```
begin
```

```
If (teclado[Terminal]="") or (teclado[Terminal]="#27)Then
```

```
estado[Terminal]:=0
```

```
else
```

```
If True Then // Neste caso Senha sempre será válida
```

```
begin
```

```
estado[Terminal]:=2;
```

```
end
```

```
else
```

```
begin
```

```
GClear(IndConexao);
```

```
Gdisplay(IndConexao,'Senha Inválida!');
```

```
estado[Terminal]:=99;
```

```
Gget(Terminal,'X');
```

```
end;
```

```
end;
```

**end;**

- ♦ Seguindo a lógica, nos estados pares envia-se mensagem para o display e muda o estado...

**2: begin**

```
Gclear(IndConexao);  
Gdisplay(IndConexao, "Mensagem:");  
Estado[Terminal]:=3;
```

**end;**

- ♦ E nos estados ímpares, pega-se o retorno do estado anterior, mudando o estado caso positivo ou voltando caso negativo ou então, mudando para um estado "morto".

**3: begin**

```
if GtrTecla () then  
begin  
    if Válido then  
        Avança ou vai para qualquer outro estado  
    else  
        volta ou vai para campo neutro;
```

**end;**

**end;**

**n-estado: begin**

.....

**end;**

**end;**

### ➤ **Balança**

Outro ponto a ser tratado é na questão de solicitação de peso, neste aspecto entramos em duas situações, como a balança trabalha e como vamos tratar o peso:

*Funcionamento:* O uso normal de balança é trabalhar por solicitação, neste caso o programa envia 05 em hexadecimal pela porta serial do terminal, a balança ao receber entende que se trata de uma solicitação de peso, pega o valor do buffer e escreve na porta serial. O terminal recebe este valor como digitação e envia para o sistema de forma e na velocidade que recebe. A outra forma, a balança fica o tempo todo escrevendo o peso, não é muito comum e se for utilizar este tipo, entre em contato com a Gradual para tratar a questão, note que o terminal recebe dados pela serial como se fosse digitação, isto significa que o terminal vai receber o peso o tempo todo e enviar o tempo todo para o sistema. Uma solução é ter a porta serial do terminal fechada e em determinado ponto do programa abrir pegar o peso e fechar novamente, o tratamento deverá ser de forma mais apurada, pois não se sabe em que parte o terminal começou a receber.

*Tratamento:* Algumas balanças quando instável ou não envia o peso ou envia um caracter indicando que o peso não é confiável. Do lado do sistema o programa deverá solicitar o peso e aguardar o recebimento, não recebendo este deve enviar novamente o pedido de peso. Neste aspecto devemos informar como o programa deverá se comportar:

- Sistema pede algumas vezes, não recebendo solicita ao usuário que digite o valor.
- Deixa o terminal aguardando até que a balança se recupere.

No exemplo temos um timer que fica varrendo o array `delai[l]` onde `l` será n-terminais. Toda vez que `delai[l]` for igual a zero ele solicita o peso novamente. O problema nesta solução é que se o peso não vier o programa fica travado. Deve-se ter um contador de pedidos, quando fizer n-pedidos o programa deve abortar e solicitar que o usuário digite o valor.

### ➤ Emulador

- Execute o Emulador.
- Clique em Terminal
- Selecione o protocolo de comunicação, lembrando que é mais interessante estar trabalhando com o protocolo VT100, o protocolo Gradual é indicado para os caso onde, o sistema vem dos modelos RS485. O protocolo Telnet emula o VT100.
- Selecione a forma que o terminal irá trabalhar, lembrando que o ideal e trabalhar como "Client", desta forma não precisará desenvolver rotinas para verificar se o terminal está na rede.
- Informe o IP da máquina onde estará sendo executado o sistema e a porta que o sistema irá abrir para a conexão.
- Clique "OK" duas vezes para exibir o Emulador.

Dica: Caso execute o Emulador e o Sistema na mesma máquina e queira executar mais de um emulador utilize o Localhost -> IP 127.0.0.1/ 127.0.0.2/ 127.0.0.3/

Sistema utilizando a Wtechlpt.dll pode ser utilizada com o emulador para desenvolvimento, use a Wtechlpt.dll versão 4, com a dll auxiliares. [http://www.gradual.com.br/programas/Wtechlpt\\_v4.htm](http://www.gradual.com.br/programas/Wtechlpt_v4.htm)

## Somente para terminais TCP-IP

**OBS:** Note que na rotina trataterminais os parâmetros IndConexao e Terminal servem para controle do terminal, isto é válido em Delphi 5.0 em Delphi 7.0 ou VB não é necessário, a questão toda é que em Delphi 5.0 quando o Socket é aberto a conexão é estabelecida por ordem de chegada, quando uma conexão é quebrada ele volta no final da fila, sendo assim seu índice no array de conexão será o último. Após alguns segundos a conexão perdida e definitivamente varrida e os índices se alteram novamente. Desta forma deve-se garantir o índice do terminal no array de conexões e a posição onde as informações relativas aquele terminal serão armazenadas.

Exemplo:

Ao abrir o socket o array está desta forma:

IP	Índice
192.168.0.10	0
192.168.0.13	1
192.168.0.20	2
192.168.0.15	3

O IP 192.168.0.10 cai em seguida volta, note que o mesmo IP está presente com índice 0 e 4

IP	Índice
<b>192.168.0.10</b>	<b>0</b>
192.168.0.13	1
192.168.0.20	2
192.168.0.15	3
<b>192.168.0.10</b>	<b>4</b>

Após alguns segundos, tempo não variável conforme rede, o índice 0 é fechado e todos os índices são re-allocated.

IP	Índice
192.168.0.13	0
192.168.0.20	1
192.168.0.15	2
<b>192.168.0.10</b>	<b>3</b>

O transtorno que isto gera é que pode-se digitar no terminal que inicialmente era 0 e ecoar no terminal com índice 1. Em nossas pesquisas não descobrimos, em Delphi 5.0, uma forma de se recuperar a mesma posição no array de conexão, em Visual Basic descobrimos que pode-se atribuir o mesmo endereço da conexão, sendo assim quando o equipamento é desligado e ligado novamente o índice será o mesmo, em Delphi 7.0 a conexão socket é feita por Thread da palheta Indy, então atribui-se o endereço da Thread. Em outras linguagens deverá ver o comportamento em caso de quebra de conexão para certificar-se que não terá este contratempo.

➤ **Delphi 7.0**

Para testar o exemplo em Delphi 7.0 deverá instalar o componente TSeverSocket .

component > install packages > add > C:\Arquivos de programas\Borland\Delphi7\Bin\dclsockets70.bpl

Link's para os exemplos:

➤ **Links**

**Para rede RS485 com comutadora Serial**

Terminais\_COM\_Delphi..... [http://www.gradual.com.br/programas/Terminais\\_COM\\_Delphi.zip](http://www.gradual.com.br/programas/Terminais_COM_Delphi.zip)

Terminais\_COM\_VB..... [http://www.gradual.com.br/programas/Terminais\\_COM\\_VB.zip](http://www.gradual.com.br/programas/Terminais_COM_VB.zip)

**Para rede RS485 com comutadora Paralela**

Terminais\_LPT\_Delphi ..... [http://www.gradual.com.br/programas/Terminais\\_LPT\\_Delphi.zip](http://www.gradual.com.br/programas/Terminais_LPT_Delphi.zip)

Terminais\_LPT\_VB ..... [http://www.gradual.com.br/programas/Terminais\\_LPT\\_VB.zip](http://www.gradual.com.br/programas/Terminais_LPT_VB.zip)

**Para terminais com comunicação Ethernet.**

Terminais\_IP\_VT100\_VB..... [http://www.gradual.com.br/programas/Terminais\\_IP\\_VT100\\_VB.zip](http://www.gradual.com.br/programas/Terminais_IP_VT100_VB.zip)

Terminais\_IP\_VT100\_Delphi..... [http://www.gradual.com.br/programas/Terminais\\_IP\\_VT100\\_Delphi.zip](http://www.gradual.com.br/programas/Terminais_IP_VT100_Delphi.zip)

Terminais\_IP\_Gradual\_Delphi..... [http://www.gradual.com.br/programas/Terminais\\_IP\\_Gradual\\_Delphi.zip](http://www.gradual.com.br/programas/Terminais_IP_Gradual_Delphi.zip)

Emulador Terminal:..... <http://www.gradual.com.br/programas/Emulador%20Terminal.zip>