Documentação
SC501GER.DLL V2.2

1. Inicializações da DLL.

vInitialize

Primeira rotina que deve ser chamada para inicializar a DLL.

procedure vInitialize;

tc_startserver

Rotina que faz com que o servidor espere por conexões de terminais.

procedure tc_startserver;

- Retorna 1 se o Servidor foi inicializado com sucesso.
- Retorna 0 se houve erro.

_TermGertecServer

Rotina que faz com que o servidor espere por conexões de terminais.

procedure _TermGertecServer;

- Não retorna se houve erro ou não.
-Mantém a compatibilidade com versões anteriores.

2. Conversão de Tipos

TCinet_ntoa

Converte um endereço de rede em um endereço IP formatado por pontos.

procedure TCinet_ntoa(nIP : DWORD; var buf : array of byte);

- nIP contém o valor de endereço de rede a ser convertido.
- buf é um ponteiro de string onde será escrito o IP.

TCinet_addr

Converte um IP formatado por pontos em endereço de rede.

function TCinet_addr(buf : array of byte): DWORD;

- buf é um ponteiro de string com o IP a ser convertido.
- Retorna o IP convertido em endereço de rede.

3. Rotinas de Controle dos Terminais Conectados

GetTabConectados

Retorna uma estrutura com a lista de terminais conectados.

function GetTabConectados(nada: integer): TTABSOCK;

```
type TTABSOCK = packed record
    TabSock: array[0..1023] of integer;
    TabIP: array[0..1023] of DWORD;
    NumSockConec: integer;
  end;
```

- nada: campo reservado, tem de ser 1.
- TabSock: lista com os SOCKETs dos terminais.
- TabIP: lista com os IPs dos terminais.
- NumSockConec: número de terminais conectados.

4. Comandos de Rede

No conjunto de rotinas deste capítulo, ID é número do SOCKET a ser enviado o comando.

bEnviaVivo

Envia comando de "vivo" para o terminal.

procedure bEnviaVivo (ID: Integer);

bSendAllwaysLive

Ao enviar este comando para o terminal, este não tenta se desconectar do servidor se o servidor deixar de enviar algum comando por mais de 12 segundos. Por padrão, o TC501 versão 2.0 já vem com esta opção habilitada.

procedure bSendAllwaysLive (ID: Integer);

bSendCheckLive

Este comando é o inverso do comando anterior, ou seja, ao envia-lo para o terminal, ao ficar por mais de 12 segundos sem receber nenhuma mensagem do servidor, o terminal faz um “ping” no servidor de 12 em 12 segundos. Se o servidor não responder depois de 10 “pings”, o terminal se desconecta e fica a procura do servidor.

function bSendCheckLive(ID: Integer): boolean;

bSendRestartSoft

Enviando este comando, o terminal é reiniciado. Uma boa sugestão, seria envia-lo após trocar seu IP (pela configuração remota), para que a configuração seja efetuada com sucesso imediatamente.

function bSendRestartSoft(ID: Integer): boolean;

bUpdateSoftware

Ao enviar este comando, o terminal tenta se atualizar remotamente, no endereço já pre-estabelecido em sua configuração.

function bUpdateSoftware(ID: Integer): boolean;

bMandaConfig

Envia configuração para o terminal.

function bMandaConfig(conftemp: PTCCONFIG): Boolean;

PTCCONFIG: ponteiro de uma estrutura do tipo TCCONFIG.

```
TCCONFIG = packed record
    ID        : integer;
    host      : array[0..21] of byte;
    endereco  : array[0..21] of byte;
    msknet    : array[0..21] of byte;
    texto1    : array[0..21] of byte;
    texto2    : array[0..21] of byte;
    texto3    : array[0..21] of byte;
    texto4    : array[0..21] of byte;
    tempoexib : byte;
  end;
```

- host: IP do servidor formatado por pontos;
- endereço: IP do terminal formatado por pontos;
- msknet: IP da mascara formatado por pontos;
- texto1/2/3/4: texto das linhas 1/2/3/4;
- tempoexib: tempo de exibição de mensagens;

bPedeConfig

Envia comando para o terminal retornar sua configuração atual.

function bPedeConfig(ID: Integer): boolean;

bMandaExtConfig

Envia configuração para o terminal.

function bMandaExtConfig(conftemp: PTCEXTCONFIG): Boolean;

PTCEXTCONFIG: ponteiro de uma estrutura do tipo TCEXTCONFIG.

```
TCEXTCONFIG = packed record
    ID         : integer;
    host       : array[0..21] of byte;
    endereco   : array[0..21] of byte;
    msknet     : array[0..21] of byte;
    gateway    : array[0..21] of byte;
    nameserver : array[0..21] of byte;
    tcname     : array[0..21] of byte;
    texto1     : array[0..21] of byte;
    texto2     : array[0..21] of byte;
    updserv    : array[0..99] of byte;
    upduser    : array[0..21] of byte;
    updpass    : array[0..21] of byte;
    tempoexib  : byte;
    dinamicip  : byte;
    buscaserv  : byte;
  end;
```

- host: IP do servidor formatado por pontos;
- endereço: IP do terminal formatado por pontos;
- msknet: IP da mascara formatado por pontos;
- gateway: IP do gateway formatado por pontos;
- nameserver: IP do servidor de nomes formatado por pontos;
- tcname: Nome do terminal;
- texto1: texto da primeira linha;
- texto2: texto da segunda linha;
- updserv: endereço do servidor de atualização;
- upduser: Nome do usuário (utilizado na atualização por FTP);
- updpass: Senha (utilizado na atualização por FTP);
- tempoexib: tempo de exibição de mensagens;
- dinamicip: IP dinâmico ou fixo;
- buscaserv: Busca Servidor;

bPedeExtConfig

Envia comando para o terminal retornar sua configuração atual. Função obsoleta, somente para TC501 versão 2.0

function bPedeExtConfig(ID: Integer): boolean;

bSendCardBonus

Envia Bonus do cartão.

function bSendCardBonus(ID: Integer; Line1, Line2 : array of byte; TimeExhibition: WORD): Boolean;

- Line1/2: string com as mensagens de cada linha do display.
- TimeExhibition: Tempo de exibição da mensagem.

bSendCardNotFound

Envia Mensagem de cartão não encontrado.

function bSendCardNotFound(ID: Integer): boolean;

bSendProdPrice

Envia Nome e Preço do produto.

function bSendProdPrice(ID: Integer; NameProd, PriceProd : array of byte): Boolean;

- NameProd: string com nome do produto;
- PriceProd: string com preço do produto.

bSendProdNotFound

Envia Mensagem de produto não encontrado;

function bSendProdNotFound(ID: Integer):boolean;

bSendDisplayMsg

Envia Mensagem para o Display do terminal.

```
function bSendDisplayMsg(
        ID: Integer;
        Line1, Line2 : array of byte;
        TimeExhibition, TypeAnimation : WORD
        ): Boolean;
```

- Line1/2: string com as mensagens de cada linha do display.
- TimeExhibition: Tempo de exibição da mensagem.
- TypeAnimation: Reservado, deve ser 48.

bSendImageFromFile

Envia Imagem (de um arquivo bmp) para o Display do terminal TC505.
Se a imagem for de tamanho diferente de 128x64 pixels ou for colorida a dll será convertida para o tamanho indicado nos parâmetros e para monocromática.

```
function bSendImageFromFile(
        ID: Integer;
        filename : array of byte;
        width : integer ;
        height : integer;
        index : integer
        ): Boolean;
```

filename: arquivo bmp com a imagem a ser enviada.
width : largura em pixels do display do terminal, para o TC 505 deve ser 128.
height: altura em pixels do display do terminal, para o TC 505 deve ser 64.

index: índice da imagem (0) para exibição imediata, (1 a 4) para loop de imagens.

BPedeParam

Envia comando para o terminal retornar seus parâmetros extras de configuração (IP dinâmico e busca servidor).

function bPedeParam(ID: Integer): boolean;

bMandaParam

Envia configuração dos parâmetros extras para o terminal.

function bMandaParam(conftemp: PTCPARAMCONFIG): boolean;

PTCPARAMCONFIG: ponteiro de uma estrutura do tipo TCPARAMCONFIG.

```
TCPARAMCONFIG = packed record
    ID        : integer;
    ipdinamico: byte;
    buscaserv : byte;
  end;
```

- ipdinamico: IP dinâmico (1) ou fixo (0);
- buscaserv: Busca servidor (1) ou não busca (0);

bPedeUpdConfig

Envia comando para o terminal retornar sua configuração de atualização.

function bPedeUpdConfig(ID: Integer): boolean;

bMandaUpdConfig,

Envia configuração dos parâmetros extras para o terminal.

function bMandaUpdConfig(conftemp: PTCUPDCONFIG): boolean;

PTCUPDCONFIG: ponteiro de uma estrutura do tipo TCUPDCONFIG.

```
TCUPDCONFIG = packed record
    ID         : integer;
    gateway    : array[0..21] of byte;
    nameserver : array[0..21] of byte;
    tcname     : array[0..21] of byte;
    updserv    : array[0..99] of byte;
    upduser    : array[0..21] of byte;
    updpass    : array[0..21] of byte;
  end;
```

- gateway: IP do gateway formatado por pontos;
- nameserver: IP do servidor de nomes formatado por pontos;
- tcname: Nome do terminal;
- updserv: endereço do servidor de atualização;
- upduser: Nome do usuário (utilizado na atualização por FTP);
- updpass: Senha (utilizado na atualização por FTP);

5. Rotinas para receber dados dos terminais

bReceiveBarcode

Rotina que deve ser chamada periodicamente, para ler dados do código de barras.

function bReceiveBarcode(var ID: integer; var Porta: integer;var buffer: PARRAYBYTE; var Nbr: integer): boolean;

- ID: valor do SOCKET que enviou o dado.
- Porta: porta que foi lido dados da serial.
- buffer: dados recebidos da serial.
- nbr: número de dados lido da serial.
- Retorna: true se recebeu algum dado, false se não tem nenhum dado a ser lido.

bReceiveCardCode

Rotina que deve ser chamada periodicamente, para ler dados do cartão magnético.

function bReceiveCardCode(var ID: integer; var buffer: PARRAYBYTE; var Nbr: integer): boolean;

- ID: número do SOCKET que enviou o dado.
- ptrilha: dados recebido da trilho 2 do cartão.
- errcode: 0 se não houve erro, outro valor se ocorreu algum erro.
- Retorna: true se recebeu algum dado, false se não tem nenhum dado a ser lido.

bReceiveConfig

Rotina que deve ser chamada periodicamente, para receber dados de configuração do terminal (previamente requisitadas pelo servidor).

function bReceiveConfig(var conftemp: PTCCONFIG): boolean;

PTCCONFIG: ponteiro de uma estrutura do tipo TCCONFIG.

```
TCCONFIG = packed record
    ID        : integer;
    host      : array[0..21] of byte;
    endereco  : array[0..21] of byte;
    msknet    : array[0..21] of byte;
    texto1    : array[0..21] of byte;
    texto2    : array[0..21] of byte;
    texto3    : array[0..21] of byte;
    texto4    : array[0..21] of byte;
    tempoexib : byte;
  end;
```

- host: IP do servidor formatado por pontos;
- endereço: IP do terminal formatado por pontos;
- msknet: IP da mascara formatado por pontos;
- texto1/2/3/4: texto das linhas 1/2/3/4;
- tempoexib: tempo de exibição de mensagens;

bReceiveExtConfig

Rotina que deve ser chamada periodicamente, para receber dados de configuração do terminal (previamente requisitadas pelo servidor). Função obsoleta, somente para TC501 versão 2.0.

function bReceiveExtConfig(var conftemp: PTCEXTCONFIG): boolean;

PTCEXTCONFIG: ponteiro de uma estrutura do tipo TCEXTCONFIG.

```
TCEXTCONFIG = packed record
    ID         : integer;
    host       : array[0..21] of byte;
    endereco   : array[0..21] of byte;
    msknet     : array[0..21] of byte;
    gateway    : array[0..21] of byte;
    nameserver : array[0..21] of byte;
    tcname     : array[0..21] of byte;
    texto1     : array[0..21] of byte;
```

```
    texto2     : array[0..21] of byte;
    updserv    : array[0..99] of byte;
    upduser    : array[0..21] of byte;
    updpass    : array[0..21] of byte;
    tempoexib  : byte;
    dinamicip  : byte;
    buscaserv  : byte;
  end;
```

- host: IP do servidor formatado por pontos;
- endereço: IP do terminal formatado por pontos;
- msknet: IP da mascara formatado por pontos;
- gateway: IP do gateway formatado por pontos;
- nameserver: IP do servidor de nomes formatado por pontos;
- tcname: Nome do terminal;
- texto1: texto da primeira linha;
- texto2: texto da segunda linha;
- updserv: endereço do servidor de atualização;
- upduser: Nome do usuário (utilizado na atualização por FTP);
- updpass: Senha (utilizado na atualização por FTP);
- tempoexib: tempo de exibição de mensagens;
- dinamicip: IP dinâmico ou fixo;
- buscaserv: Busca Servidor;

GetResponseCtr

Obtém ACK de comandos enviados para o terminal.

function GetResponseCtr(var pID: integer; var aresp :integer): boolean;

- pID: ID do terminal que enviou;
- aresp: Valor que identifica qual comando o terminal enviou.

bReceiveParam

Rotina que deve ser chamada periodicamente, para receber parâmetros extras de configuração do terminal (previamente requisitadas pelo servidor).

function bReceiveParam(var conftemp: PTCPARAMCONFIG): boolean;

PTCPARAMCONFIG: ponteiro de uma estrutura do tipo TCPARAMCONFIG.

```
TCPARAMCONFIG = packed record
    ID        : integer;
    ipdinamico: byte;
    buscaserv : byte;
  end;
```

- ipdinamico: IP dinâmico (1) ou fixo (0);
- buscaserv: Busca servidor (1) ou não busca (0);

bReceiveUpdConfig

Rotina que deve ser chamada periodicamente, para receber dados de configuração de atualização do terminal (previamente requisitadas pelo servidor).

function bReceiveUpdConfig(var conftemp: PTCUPDCONFIG): boolean;

PTCUPDCONFIG: ponteiro de uma estrutura do tipo TCUPDCONFIG.

```
TCUPDCONFIG = packed record
    ID         : integer;
    gateway    : array[0..21] of byte;
    nameserver : array[0..21] of byte;
    tcname     : array[0..21] of byte;
    updserv    : array[0..99] of byte;
    upduser    : array[0..21] of byte;
    updpass    : array[0..21] of byte;
  end;
```

- gateway: IP do gateway formatado por pontos;
- nameserver: IP do servidor de nomes formatado por pontos;
- tcname: Nome do terminal;
- updserv: endereço do servidor de atualização;
- upduser: Nome do usuário (utilizado na atualização por FTP);
- updpass: Senha (utilizado na atualização por FTP);