Manual do Desenvolvedor

Terminal de Consulta 506

SC501GER.DLL V2.4

R00.22

Sumário

Inicializações da DLL 4
vInitialize 4
tc_startserver 4
_TermGertecServer 4
vFinalize 4
dll_version 4
Conversão de Tipos 4
TCinet_ntoa1 4
TCinet_addr1 4
Rotinas de Controle dos Terminais Conectados 5
GetTabConectados 5
iTypeTerm 5
Comandos de Rede 5
bEnviaVivo 5
bSendAllwaysLive 5
bSendCheckLive 5
bSendRestartSoft 6
bUpdateSoftware 6
bMandaConfig 6
bPedeConfig 6
bMandaExtConfig 6
bPedeExtConfig 7
bSendCardBonus 7
bSendCardNotFound 7
bSendProdPrice 7
bSendDisplayMsg 8
bSendImageFromFile 8
bSendImagePrice 8
bPedeParam 9
bMandaParam 9
bPedeUpdConfig 9
bMandaUpdConfig 9
bPedeWlanConfig 10
bMandaWlanConfig 10
bPedeMacAddr 10
Rotinas para receber dados dos terminais 10
bReceiveBarcode 10
bReceiveCardCode 11
bReceiveConfig 11

bReceiveExtConfig 11
GetResponseCtr 12
bReceiveParam 12
bReceiveUpdConfig 12
bReceiveWlanConfig 13
bReceiveMacAddrConfig 13

Inicializações da DLL

vInitialize

Primeira rotina que deve ser chamada para inicializar a DLL.

procedure vInitialize; stdcall;

tc_startserver

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

procedure tc_startserver; stdcall;
- 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; stdcall;
- Não retorna se houve erro ou não.
-Mantém a compatibilidade com versões anteriores.

vFinalize

Rotina que deve ser chamada ao encerrar a aplicação que utiliza a DLL.

procedure vFinalize; stdcall;

dll_version

Função que retorna a versão da DLL.

function dll_version: DWORD; stdcall;
Retorna um DWORD que representa a versão da DLL. A DLL verão 2.4 retorna: 0x02040000.

Conversão de Tipos

TCinet_ntoa1

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

procedure TCinet_ntoa(nIP : DWORD; var buf : byte); stdcall;
- nIP contém o valor de endereço de rede a ser convertido.
- buf é um ponteiro de string onde será escrito o IP.

TCinet_addr1

Converte um IP formatado por pontos em endereço de rede.

function TCinet_addr(var buf : byte): DWORD; stdcall;
- buf é um ponteiro de string com o IP a ser convertido.
- Retorna o IP convertido em endereço de rede.

Rotinas de Controle dos Terminais Conectados

GetTabConectados

Retorna uma estrutura com a lista de terminais conectados.

function GetTabConectados(nada: integer): TTABSOCK; stdcall;
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.

iTypeTerm

Retorna o tipo de terminal conectado.

function iTypeTerm(ID : integer): integer; stdcall;
- Retorna 1 se o Terminal for um TC502.
- Retorna 2 se o Terminal for um TC505.
- Retorna 0 se o Terminal for um TC501 ou um tipo desconhecido.

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); stdcall;

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); stdcall;

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; stdcall;

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; stdcall;

bUpdateSoftware

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

function bUpdateSoftware(ID: Integer): boolean; stdcall;

bMandaConfig

Envia configuração para o terminal.

function bMandaConfig(conftemp: PTCCONFIG): Boolean; stdcall;
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; stdcall;

bMandaExtConfig

Envia configuração para o terminal.

function bMandaExtConfig(conftemp: PTCEXTCONFIG): Boolean; stdcall;
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; stdcall;

bSendCardBonus

Envia Bonus do cartão.

function bSendCardBonus(ID: Integer; var Line1: byte, var Line2 : byte; TimeExhibition: WORD): Boolean;
stdcall;
- 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; stdcall;

bSendProdPrice

Envia Nome e Preço do produto.

function bSendProdPrice(ID: Integer; var NameProd, PriceProd : byte): Boolean; stdcall;
- NameProd: string com nome do produto;

- PriceProd: string com preço do produto.
bSendProdNotFound
Envia Mensagem de produto não encontrado;
function bSendProdNotFound(ID: Integer):boolean; stdcall;

bSendDisplayMsg

Envia Mensagem para o Display do terminal.

function bSendDisplayMsg(
ID: Integer;
var Line1, Line2 : byte;
TimeExhibition, TypeAnimation : WORD
): boolean; stdcall;
- 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;
var filename : byte;
width : integer ;
height : integer;
index : integer
): boolean; stdcall;
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.

bSendImagePrice

Envia Imagem, com duas linhas de mensagens, gerada pela DLL para o terminal. Pode ser utilizado para enviar a descrição (máximo de 20 caracteres) do produto e o preço (máximo de 20 caracteres) para o Display do terminal TC505.

function bSendImagePrice(ID: Integer; var NameProd: byte; var PriceProd : byte;
Tempo:integer; width,
height: integer): Boolean; stdcall;
NameProd: Texto que será mostrado na parte superior do display.
PriceProd: Texto que será mostrada na parte inferior do display.
Tempo: Tempo de exibição em segundos.
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.

bPedeParam

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

function bPedeParam(ID: Integer): boolean; stdcall;

bMandaParam

Envia configuração dos parâmetros extras para o terminal.

function bMandaParam(conftemp: PTCPARAMCONFIG): boolean; stdcall;
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; stdcall;

bMandaUpdConfig

Envia configuração dos parâmetros extras para o terminal.

function bMandaUpdConfig(conftemp: PTCUPDCONFIG): boolean; stdcall;
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);

bPedeWlanConfig

Envia comando para o terminal retornar sua configuração wireless.
Está função só funciona nos terminais de consulta com suporte à rede sem fio.

function bPedeWlanConfig(ID: Integer): boolean; stdcall;

bMandaWlanConfig

Envia configuração dos parâmetros extras para o terminal.
Está função só funciona nos terminais de consulta com suporte à rede sem fio.

function bMandaWlanConfig(conftemp: PTCWLANCONFIG): Boolean; stdcall;
PTCWLANCONFIG: ponteiro de uma estrutura do tipo TCWLANCONFIG.
type
TCWLANCONFIG = packed record
ID : integer;
usewifi : byte;
mode : byte;
ssid : array[0..21] of byte;
channel : byte;
wep : byte;
wepkey : array[0..21] of byte;
end;
- usewifi: 0 para desabilitar a rede sem fio (habilita a rede com cabo ethernet), 1 para habilitar;
- mode: 0 para infraestrutura, 1 para ad-hoc;
- ssid: SSID da rede;
- channel: Canal a ser utilizado, de 1 até 11 (somente no modo ad-hoc);
- wep: 0 desabilita utilização da chave WEP, 1 habilita;
- wepkey: Chave WEP a ser utilizada;

bPedeMacAddr

Envia comando para o terminal retornar seu endereço mac de rede.
Está função só funciona nos terminais de consulta V3.0 ou superior.

function bPedeMacAddr(ID: Integer; iface: byte): boolean; stdcall;
- iface: 0 retorna endereço MAC da interface ethernet
1 retorna endereço MAC da interface sem fio
9 retorna endereço MAC da interface que estiver sendo utilizada no momento

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: byte; var Nbr: integer): boolean;
stdcall;
- 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; stdcall;
- 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: TCCONFIG; stdcall;
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; stdcall;
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; stdcall;
- 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: TCPARAMCONFIG; stdcall;
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: TCUPDCONFIG; stdcall;
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);

bReceiveWlanConfig

Rotina que deve ser chamada periodicamente, para receber dados de configuração de endereço MAC do terminal (previamente requisitadas pelo servidor).
Está função só funciona nos terminais de consulta com suporte à rede sem fio.

function bReceiveWlanConfig: TCWLANCONFIG; stdcall;
PTCWLANCONFIG: ponteiro de uma estrutura do tipo TCWLANCONFIG.
type
TCWLANCONFIG = packed record
ID : integer;
usewifi : byte;
mode : byte;
ssid : array[0..21] of byte;
channel : byte;
wep : byte;
wepkey : array[0..21] of byte;
end;
- usewifi: 0 para desabilitar a rede sem fio (habilita a rede com cabo ethernet), 1 para habilitar;
- mode: 0 para infraestrutura, 1 para ad-hoc;
- ssid: SSID da rede;
- channel: Canal a ser utilizado, de 1 até 11 (somente no modo ad-hoc);
- wep: 0 desabilita utilização da chave WEP, 1 habilita;
- wepkey: Chave WEP a ser utilizada;

bReceiveMacAddrConfig

Rotina que deve ser chamada periodicamente, para receber dados de configuração de endereço MAC do terminal (previamente requisitadas pelo servidor).
Está função só funciona nos terminais de consulta com suporte à rede sem fio.

function bReceiveMacAddrConfig: TCMACADDRCONFIG; stdcall;
PTCMACADDRCONFIG: ponteiro de uma estrutura do tipo TCMACADDRCONFIG.
type
TCMACADDRCONFIG = packed record
ID : integer;
iface : byte;
macaddr : array[0..16] of byte;
end;
- iface: 0 se macaddr contém o valor da interface ethernet, 1 se macaddr contém o valor da interface sem fio;
- macaddr: Endereço mac no formato ASCII