Protocolo Microterminal Gertec (PMTG)

I. Introdução

O objetivo do presente documento é descrever o funcionamento do protocolo de comunicação dos microterminais Gertec, possibilitando o desenvolvimento de servidores.

Os microterminais TCP/IP GE7xx e MT7xx utilizam um protocolo de mensagens sobre a interface TCP/IP que permite a comunicação do mesmo com um servidor: o Protocolo Microterminal Gertec (PMTG). Para plataforma Microsoft Windows 32 bits, a Gertec fornece uma DLL (*Dynamic Link Library*), denominada PMTG.DLL, que gerencia este protocolo do servidor junto aos GE7xx/MT7xx. Para maiores detalhes sobre esta DLL procure em seu manual.

O Microterminal procurará o servidor no endereço IP configurado, tentando conectar-se à porta 6550. Assim, o servidor deverá abrir um socket em modo *listen* com esta porta. Quando um terminal conectar-se a este socket, o servidor deverá enviar o comando wGetIdentify descrito adiante para que o terminal reconheça a conexão.

II. Mensagens

A mensagem é qualquer informação lida ou escrita pelo microterminal no "*socket*" aberto para o servidor. A partir dessas mensagens, o terminal e servidor executam tarefas específicas, assim estas mensagens podem ser referenciadas também como comando. Na versão atual, todas as mensagens são binárias com identificador em número único (ID) em WORD (16 bits). Para cada mensagem gerada, há uma resposta com número de identificação consecutivo (ID+1), e dependendo da mensagem pode haver argumentos.

Composição

A composição da mensagem é apresentada na Tabela 1. Os primeiros dois bytes (WORD) representam o identificador da mensagem. Os próximos dois bytes (WORD) simbolizam o tamanho do argumento a ser lido, que vem em seguida. Assim, o PMTG exige que sejam lidas a cada interação com socket pelo menos duas WORDs. E caso o tamanho do argumento (**n**) seja diferente de zero, o *socket* deve ler o argumento em **n** bytes consecutivos na mensagem.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

ID COMANDO (WORD – 2 BYTES)	TAMANHO DO ARGUMENTO – n (WORD – 2 BYTES)	ARGUMENTO (n BYTES)

Tabela 1: Composição da mensagem

Exemplo

Por exemplo, quando o terminal está conectado e é pressionada a tecla 'a', é enviado para o servidor (valores em hexadecimal):

[1d][00][01][00][61]

Onde:

[1d][00]: ID COMANDO (001d = cGetCharTerm);

[01][00]: TAMANHO DO ARGUMENTO (0001 = 1 byte);

[61]: ARGUMENTO (1 byte = código ASCII da letra 'a').

Note que no caso das WORDs os bytes estão em formato *"little endian"*, isto é, o byte menos significativo ocupa o endereço menos significativo. Assim, o valor LONG WORD (4 bytes) **00000001** será enviado para o servidor como **[01][00][00][00]**.

No presente protocolo, os valores BOOL são representados por uma LONG WORD (quatro bytes). A seguinte terminologia é adotada:

TRUE = **00000001h**

FALSE = **00000000h**

"Void" equivale a "nenhum". WORD = 2 bytes, LONG WORD = 4 bytes.

Classificação

As mensagens estão classificadas em:

- **Atividade e Configuração TCP/IP**
- **Leitor de Cartão Magnético**
- **Teclado**
- **Display de Cristal Líquido**
- **Comunicação Serial**
- **Comunicação Paralela (LPT1)**

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

Tabela 2 – Identificação das Mensagens por Mnemônico e ID [decimal (hexadecimal)]

	Mnemônicos das Mensagens	ID Comando	Origem	GE750	GE760	MT740	MT720
TCP/IP	vLive	1 (0001)	**Servidor**	√	√	√	√
	wGetIdentify	3 (0003)	**Servidor**	√	√	√	√
	vSetSetupTCP	5 (0005)	**Servidor**	√	√	√	√
	vGetSetupTCP	7 (0007)	**Servidor**	√	√	√	√
	vSetExSetupTCP	81 (0051)	**Servidor**	√Δ	√	√	√
	vGetExSetupTCP	83 (0053)	**Servidor**	√Δ	√	√	√
	vRestart	89 (0059)	**Servidor**	√Δ	√	√	√
	vFTPMode	91 (005B)	**Servidor**	√Δ	√		
	vGetMAC	101(0065)	**Servidor**			√♣	√♣
CARD	vSetCard	9 (0009)	**Servidor**	√		√	
	bGetCard	11 (000B)	**Servidor**	√		√	
	bReadBuffCard	**13 (000D)**	***MicroT***	√		√	
TECLADO	vSetEnableKey	15 (000F)	**Servidor**	√	√	√	√
	bGetEnableKey	17 (0011)	**Servidor**	√	√	√	√
	vReset	19 (0013)	**Servidor**	√	√	√	√
	vSetCapsLock	21 (0015)	**Servidor**	√	√	√	√
	bGetCapsLock	23 (0017)	**Servidor**	√	√	√	√
	vSetNumLock	25 (0019)	**Servidor**	√	√	√	√
	bGetNumLock	27 (001B)	**Servidor**	√	√	√	√
	cGetCharTerm	**29 (001D)**	***MicroT***	√	√	√	√
	bProgramKbd	31 (001F)	**Servidor**	√	√	√	√
	vSetBeep	93 (005D)	**Servidor**		√	√	√
	vSetBeepKey	95 (005F)	**Servidor**		√	√	√
DISPLAY	vBackSpace	33 (0021)	**Servidor**	√	√	√	√
	vCarRet	35 (0023)	**Servidor**	√	√	√	√
	vLineFeed	37 (0025)	**Servidor**	√	√	√	√
	vFormFeed	39 (0027)	**Servidor**	√	√	√	√
	vGoToXY	41 (0029)	**Servidor**	√	√	√	√
	vGoToXYRef	43 (002B)	**Servidor**	√	√	√	√
	bReadEditString	**45 (002D)**	***MicroT***	√	√	√	√
	vSetEditString	47 (002F)	**Servidor**	√	√	√	√
	bGetEditString	49 (0031)	**Servidor**	√	√	√	√
	vDispStr	51 (0033)	**Servidor**	√	√	√	√
	vDispCh	53 (0035)	**Servidor**	√	√	√	√
	vDispClrLn	55 (0037)	**Servidor**	√	√	√	√
	vSetBack	99 (0063)	**Servidor**			√♣	√♣
COM1, 2, 3 e 4 *	vSetEnableSerial	57 (0039)	**Servidor**	√	√	√	√
	bGetEnableSerial	59 (003B)	**Servidor**	√	√	√	√
	bGetBinSerial	**61 (003D)**	***MicroT***	√	√	√	√
	bSendBinSerial	63 (003F)	**Servidor**	√	√	√	√
	bSetSetupSerial	65 (0041)	**Servidor**	√	√	√	√
	vGetSetupSerial	67 (0043)	**Servidor**	√	√	√	√
LPT1	cInitPrn	69 (0045)	**Servidor**	√		√†	
	cGetStatusPrn	71 (0047)	**Servidor**	√		√†	
	cSendPrn	73 (0049)	**Servidor**	√		√†	

** Portas COM disponíveis: **GE750**: 1 e 2; **GE760**: 1; **MT740**: 1, 2, 3 e 4; **MT720**: 1, 2 e 3*

† *Este modelo não possui LPT1.Sempre responderá erro de impressora.*

Δ *Somente a partir da versão 1.2*

♣ *Somente a partir da versão 2.2*

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

Descrição

a. Atividade e Configuração TCP/IP

Estas mensagens são utilizadas a qualquer momento durante a conexão, com exceção do wGetIdentify, que é aguardado pelo microterminal logo após ter realizado a conexão. Após o processo de conexão, esta mensagem é ignorada pelo microterminal por um critério de segurança. Todas estas mensagens são originárias do servidor.

vLive	0001h
Origem:	Servidor
Descrição:	Atividade do terminal
Argumento:	Void

Resposta

ID:	0002h
Argumento:	Void

A mensagem vLive é utilizada pelo servidor para verificar se o microterminal continua ativo. Para cada vLive com ID igual a 0001h, o microterminal responde com vLive+1 (0002h) sem nenhum argumento (void).

wGetIdentify	0003h
Origem:	Servidor
Descrição:	Solicita a Identificação do microterminal após ter conectado ao servidor
Argumento:	Void

Resposta

ID:	0004h
Argumento:	WORD

A identificação do microterminal é aguardada por este durante o processo de conexão ao servidor. Por isso, logo após a conexão ao servidor, este deve enviar o comando wGetIdentify (0003h). O Microterminal responde com wGetIdentify+1 (0004h), com argumento WORD (16bits), onde o byte mais significativo identifica o tipo de microterminal e o byte menos significativo identifica sua versão. Caso o servidor envie algum outro comando antes de wGetIdentify, o microterminal se reiniciará.

Identificação da versão de firmware: 1.0 = 0x10, 2.2 = 0x22 etc.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

Byte + significativo	Modelo
01h	GE750
02h	GE760
03h	MT740
04h	MT720

vSetSetupTCP	***0005h***
Origem:	Servidor
Descrição:	Configura os parâmetros TCP/IP do microterminal
Argumento:	TSetupTCP

Resposta

ID:	***0006h***
Argumento:	Void

Para configurar o microterminal *online*, envia-se o comando vSetSetupTCP, com o argumento TSetupTCP:

Tipo	*Nome*	*Descrição*
DWORD	microT_IP	Endereco IP do microterminal
DWORD	server_IP	Endereço IP do Servidor
DWORD	mskнet_IP	Endereço da Máscara de Rede
BOOL	bDHCP	Caso VERDADEIRO, o microterminal desconsidera seu endereço IP procurando o servidor DHCP (IP Dinâmico). Caso contrário, utiliza IP Fixo com os valores definidos.

Tabela 3: Estrutura TSetupTCP

Exemplo: ao enviar um comando para configurar o terminal com o endereço IP 192.168.0.120, endereço de servidor 192.168.0.48, máscara de rede 255.255.255.0 e IP dinâmico ativado, é enviado um pacote TCP com o seguinte conteúdo (capturado com o software Ethereal, apenas bytes de dados, valores em hexadecimal):

```
05 00 10 00 78 00 A8 C0 30 00 A8 00 00 FF FF FF 01 00 00 00
```

Onde:

[`05 00`] é o código do comando (vSetSetupTCP);

[`10 00`] é a quantidade de bytes do argumento (**10h** = **16d** bytes);

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

[78 00 A8 C0] é o IP do terminal (em decimal: **[120] [0] [168] [192]**. Observe a ordem);

[30 00 A8 C0] é o IP do servidor (em decimal: **[48] [0] [168] [192]**);

[00 FF FF FF] é a máscara de rede (em decimal: **[0] [255] [255] [255]**);

[01 00 00 00] é um valor booleano TRUE, indicando DHCP ativo.

Com os parâmetros dessa estrutura, o microterminal é reconfigurado, retornando o comando vSetSetupTCP+1, sem argumento (void). Ao receber esta mensagem, o servidor deve fechar a conexão TCP/IP com microterminal, para este reconectar com os novos parâmetros.

vGetSetupTCP	***0007h***
Origem:	Servidor
Descrição:	Solicita os parâmetros TCP/IP do microterminal
Argumento:	Void

Resposta

ID:	***0008h***
Argumento:	TSetupTCP

Para receber a configuração TCP/IP do microterminal, o servidor envia o comando vGetSetupTCP (0007h), sem argumento. O microterminal responderá com um vGetSetupTCP+1 (0008h), com o argumento TSetupTCP descrito na tabela 3.

vSetExSetupTCP	***0051h***
Origem:	Servidor
Descrição:	Configura os parâmetros TCP/IP estendidos do microterminal
Argumento:	TExSetupTCP

Resposta

ID:	***0052h***
Argumento:	Void

Para configurar o gateway, servidor de nomes e o nome do terminal *online*, envia-se o comando vSetExSetupTCP, com o argumento TExSetupTCP. O nome do terminal é uma string que pode ser utilizada para a identificação de cada terminal na rede, como, por exemplo, “Hortifruti”, “Caixa no.1” etc.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

Tipo	*Nome*	*Descrição*
DWORD	gateway	Endereco IP do gateway
DWORD	nameserver	Endereço IP do Servidor de Nomes
DWORD	myname[16]	Nome do terminal

Tabela 4: Estrutura TExSetupTCP

Exemplo: ao enviar um comando para configurar o terminal com o gateway 192.168.0.2, endereço de servidor de nomes 192.168.0.3 e nome do terminal "MT720", é enviado um pacote TCP com o seguinte conteúdo (capturado com o software Ethereal, apenas bytes de dados, valores em hexadecimal):

```
51 00 18 00 02 00 A8 C0 03 00 A8 C0 4D 54 37 32 30 00 12 00 3C 2C 38
7E 10 D9 15 00
```

Onde:

[51 00] é o código do comando (vSetExSetupTCP);

[18 00] é a quantidade de bytes do argumento (**18h** = **24d** bytes);

[02 00 A8 C0] é o IP do gateway (em decimal: **[2] [0] [168] [192]**);

[03 00 A8 C0] é o IP do servidor de nomes (em decimal**: [3] [0] [168] [192]**);

[4D 54 37 32 30 00 12 00 3C 2C 38 7E 10 D9 15 00] é a string do nome do terminal (ASCII: "**M T 7 2 0**". O sexto byte é **[00]**, que é o terminador da string. Assim, os demais bytes serão ignorados.

Com os parâmetros dessa estrutura, o microterminal é reconfigurado, retornando o comando vSetExSetupTCP+1, sem argumento (void). Ao receber esta mensagem, o servidor deve fechar a conexão TCP/IP com microterminal, para este reconectar com os novos parâmetros.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vGetExSetupTCP	***0053h***
Origem:	Servidor
Descrição:	Solicita os parâmetros TCP/IP estendidos do microterminal
Argumento:	Void

Resposta

ID:	***0054h***
Argumento:	TExSetupTCP

Para receber a configuração TCP/IP do microterminal, o servidor envia o comando vGetExSetupTCP (0053h), sem argumento. O microterminal responderá com um vGetExSetupTCP+1 (0054h), com o argumento TExSetupTCP descrito na tabela 4.

vRestart	***0059h***
Origem:	Servidor
Descrição:	Reinicia o terminal
Argumento:	LONG WORD

Resposta

ID:	***005Ah***
Argumento:	Void

Este comando reinicia o terminal. O argumento é uma senha de 32 bits igual a **5a33a5cc** H.

vFTPMode	***005Bh***
Origem:	Servidor
Descrição:	Coloca o terminal em modo FTP
Argumento:	void

Resposta

ID:	***005Ch***
Argumento:	Void

vFTPMode coloca o terminal em modo FTP.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vGetMAC	0065h
Origem:	Servidor
Descrição:	Obtém o MAC address do terminal
Argumento:	void

Resposta

ID:	0066h
Argumento:	ARG_STR

O comando vGetMAC requisita o endereço físico do terminal (MAC address). O endereço é retornado através de um argumento ARG_STR (tabela 10), ou seja, em formato de string terminada com NULL, por exemplo: 001D5B12345.

Não é possível alterar o MAC address do microterminal.

b. Leitor de Cartão Magnético

Os comandos referentes ao leitor de cartão magnético são três: vSetCard, vGetCard e vReadBuffCard. Os dois primeiros são originários do servidor, enquanto que o último é gerado pelo microterminal.

vSetCard	0009h
Origem:	Servidor
Descrição:	Habilita ou não o Leitor do Cartão Magnético
Argumento:	BOOL

Resposta

ID:	000Ah
Argumento:	Void

Para o microterminal reconhecer a passagem de cartão magnético, é necessário habilitar o leitor previamente. Para habilitar ou não o leitor, o servidor deve enviar a mensagem vSetCard (0009h), onde o argumento booleano quando TRUE habilita o leitor, e FALSE desabilita-o. Uma vez habilitado o leitor, este será desabilitado automaticamente após a passagem de um cartão ou após o pressionamento de alguma tecla.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vGetCard	***000Bh***
Origem:	Servidor
Descrição:	Retorna o estado do leitor de cartão magnético
Argumento:	Void

Resposta

ID:	***000Ch***
Argumento:	BOOL

O estado do leitor do cartão magnético é retornado pelo microterminal quando este recebe a mensagem vGetCard (oooBh), com argumento BOOL. Se o retorno for TRUE, o leitor está habilitado, e será FALSE quando não.

bReadBuffCard	***000Dh***
Origem:	Terminal
Descrição:	Retorna a leitura do cartão magnético
Argumento:	ARG_CARD

Resposta

ID:	***000Eh***
Argumento:	Void

Quando o leitor do cartão magnético está habilitado, e ocorre a passagem de cartão ou um tecla é pressionada a mensagem bReadBuffCard (oooDh) é enviada pelo microterminal, com a estrutura ARG_CARD como argumento:

Tipo	***Nome***	***Descrição***
BYTE [128]	card	Código Binário do Cartão, onde os códigos de cada trilha estão separados por pares de colchetes, o primeiro par de colchetes são da trilha 1 e o segundo da trilha 2. O primeiro dígito dentro de um par de colchetes é o número da trilha. Quando há erro de leitura do cartão, em card encontra-se a seguinte informação: [1][2]. Se o leitor for de apenas trilha 2, card conterá [1][leitura da trilha 2].
BOOL	status	Estado do Código. Se for FALSE, houve o cancelamento da passagem do cartão devido ao pressionamento de uma tecla. E se for TRUE ocorreu a passagem do cartão.

Tabela 5: Estrutura ARG_CARD

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

Se após a habilitação da leitura do cartão alguma tecla for pressionada, a leitura do cartão será cancelada. Após a leitura de um cartão, o leitor é desabilitado automaticamente, sendo necessário reabilitá-lo antes de se fazer uma nova leitura.

c. Teclado

Os comandos referentes ao teclado são nove, sendo oito gerados pelo servidor e um pelo microterminal.

vSetEnableKey	***000Fh***
Origem:	Servidor
Descrição:	Habilita ou desabilita a leitura do teclado
Argumento:	BOOL

Resposta	
ID:	***0010h***
Argumento:	Void

O comando vSetEnableKey (000Fh) habilita ou não a leitura do teclado. Este controle é passado pelo argumento BOOL também enviado pelo servidor.

vGetEnableKey	***0011h***
Origem:	Servidor
Descrição:	Retorna o estado de habilitação de teclas
Argumento:	Void

Resposta	
ID:	***0012h***
Argumento:	BOOL

Para verificar o estado de habilitação das teclas, utiliza-se o comando vGetEnableKey (0011h) enviado pelo servidor. O microterminal responde com vGetEnableKey (0012h), com argumento BOOL indicando se as teclas estão habilitadas ou não.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vReset	***0013h***
Origem:	Servidor
Descrição:	Reinicia o teclado
Argumento:	Void

Resposta

ID:	***0014h***
Argumento:	Void

Para reiniciar o teclado do microterminal, o servidor envia o comando vReset (0013h), e microterminal responde com vReset+1 (0014h).

vSetCapsLock	***0015h***
Origem:	Servidor
Descrição:	Habilita ou não o CAPSLOCK do teclado
Argumento:	BOOL

Resposta

ID:	***0016h***
Argumento:	Void

Para habilitar ou não CAPSLOCK, o servidor envia o comando vSetCapsLock (0015h), com argumento BOOL, onde TRUE habilita o CAPSLOCK e FALSE o desabilita.

vGetCapsLock	***0017h***
Origem:	Servidor
Descrição:	Retorna o estado do CAPSLOCK do teclado
Argumento:	Void

Resposta

ID:	***0018h***
Argumento:	Bool

Para reconhecer o estado do CAPSLOCK do teclado, o servidor envia o comando vGetCapsLock (0017h), então o microterminal responde com vGetCapsLock+1 (0018h), com argumento BOOL, onde TRUE representa CASPLOCK habilitado.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vSetNumLock	***0019h***
Origem:	Servidor
Descrição:	Habilita ou não o NUM LOCK do teclado
Argumento:	Bool

Resposta

ID:	***001Ah***
Argumento:	Void

Para habilitar ou não NUMLOCK, o servidor envia o comando vSetNumLock (0019h), com argumento BOOL, onde TRUE habilita o NUMLOCK.

vGetNumLock	***001Bh***
Origem:	Servidor
Descrição:	Retorna o estado do NUMLOCK do teclado
Argumento:	Void

Resposta

ID:	***001Ch***
Argumento:	Bool

Para reconhecer o estado do NUMLOCK do teclado, o servidor envia o comando vGetNumLock (001Bh), então o microterminal responde com vGetNumLock+1 (001Ch), com argumento BOOL, onde TRUE representa NUMLOCK habilitado.

vGetCharTerm	***001Dh***
Origem:	Microterminal
Descrição:	Retorna o código ASCII da tecla pressionada
Argumento:	BYTE

Resposta

ID:	***001Eh***
Argumento:	Void

Quando o teclado está habilitado e microterminal não está em edição de string e nem com leitor de cartão habilitado, ao pressionar uma tecla no microterminal o código ASCII da tecla é enviado ao servidor pelo comando cGetCharTerm (001Dh), com argumento BYTE. O servidor responde com cGetCharTerm+1 (001Eh).

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

bProgramKbd	***001Fh***
Origem:	Servidor
Descrição:	Programa os quatro códigos das teclas
Argumento:	ARG_CODE

Resposta

ID:	***0020h***
Argumento:	BOOL

Para programar o teclado 4 códigos (cada tecla pode gerar até 4 caracteres diferentes quando pressionada), o servidor utiliza o comando bProgramKbd (001Fh), com argumento ARG_CODE (Tabela 6), onde estão armazenados o código binário padrão Gertec para teclados 4 códigos. Ao terminar a programação, o microterminal responde com bProgramKbd com argumento BOOL, onde TRUE significa que o teclado foi corretamente reprogramado. O código binário pode ser gerado através de programas disponíveis em http://www.gertec.com.br , ou pode ser gerado com base na informação constante no manual do produto sobre o endereçamento das teclas.

Os 512 bytes enviados para o teclado são divididos em 4 páginas de 128 bytes. Assim, se quisermos que uma tecla que esteja na posição 0x30 retorne ENTER, por exemplo, colocamos o código correspondente ao ENTER na posição 30h do array de 512 bytes a ser enviado. Se além de ENTER quisermos que esta mesma tecla envie também um caracter "A", colocamos o código correspondente ao caracter "A" na posição 30h + 80h = B0h, ou seja, na segunda página. Os outros dois códigos correspondentes a esta tecla estariam nas posições 0x30 + 0x100 = 0x130 e 0x30 + 0x180 = 0x1b0. O mesmo procedimento é repetido para todas as teclas, e nas posições em que não se deseja nenhum retorno deve-se colocar zero (00h). Assim é montado o array de 512 bytes, bastando enviá-lo com o comando bProbramKbd. A tabela de códigos das teclas também se encontra no manual do produto.

Tipo	***Nome***	***Descrição***
BYTE [512]	code	Código padrão Gertec para programação do teclado 4 códigos.

Tabela 6: Estrutura ARG_CODE

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vSetBeep	***005Dh***
Origem:	Servidor
Descrição:	Liga / desliga o buzzer
Argumento:	BOOL

Resposta

ID:	***005Eh***
Argumento:	Void

Para controlar o buzzer, o servidor envia a mensagem vSetBeep (005Dh), onde o argumento BOOL, quando TRUE, liga o buzzer. Uma vez ligado o buzzer, este emitirá um apito contínuo até que o terminal receba o comando para desligá-lo, ou quando o terminal for reiniciado.

vSetBeepKey	***005Fh***
Origem:	Servidor
Descrição:	Habilita / desabilita o beep das teclas
Argumento:	BOOL

Resposta

ID:	***0060h***
Argumento:	Void

Para habilitar ou desabilitar o Beep de teclas, o servidor deve enviar a mensagem vSetBeepKey (005Fh), onde o argumento booleano quando TRUE habilita e FALSE desabilita.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

d. Display de Cristal Líquido (LCD)

As funções para controle do display do cristal líquido são 12, sendo uma gerada pelo microterminal.

vBackSpace	***0021h***
Origem:	Servidor
Descrição:	Apaga o caractere precedente ao cursor no display
Argumento:	Void

Resposta	
ID:	***0022h***
Argumento:	Void

Para apagar o caractere ao lado esquerdo do cursor, o servidor manda o comando vBackSpace (0021h) e microterminal responde com vBackSpace (0022h). Se o cursor estiver na primeira coluna de uma linha, nenhuma ação é executada no display.

vCarRet	***0023h***
Origem:	Servidor
Descrição:	Manda um Carriage Return para o display
Argumento:	Void

Resposta	
ID:	***0024h***
Argumento:	Void

Para enviar um Carriage Return para o display, o servidor deve mandar a mensagem vCarRet (0023h) e microterminal responde com vCarRet+1 (0024h).

O “Carriage Return”, ou “CR”, ou ainda “Retorno de Carro”, posiciona o cursor do display na primeira coluna da linha atual.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vLineFeed	***0025h***
Origem:	Servidor
Descrição:	Manda um Line Feed para o display
Argumento:	Void

Resposta

ID:	***0026h***
Argumento:	Void

Para enviar um Line Feed para o display, o servidor deve mandar a mensagem vLineFeed (0025h) e microterminal responde com vLineFeed+1 (0026h).

Caso o cursor não esteja na última linha do display, este comando fará o cursor deslocar-se para a linha abaixo, na mesma coluna. Se o cursor estiver na última linha, as linhas serão deslocadas para cima, e o cursor continuará na mesma posição do LCD.

vFormFeed	***0027h***
Origem:	Servidor
Descrição:	Manda um Form Feed (limpa) para o display
Argumento:	Void

Resposta

ID:	***0028h***
Argumento:	Void

Para enviar um Form Feed para o display, o servidor deve mandar a mensagem vFormFeed (0027h) e microterminal responde com 0028h. Este comando excluirá quaisquer caracteres que estejam em exibição e posicionará o cursor na primeira coluna da primeira linha.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vGoToXY	***0029h***
Origem:	Servidor
Descrição:	Desloca o cursor para uma posição absoluta
Argumento:	ARG_BYTE_BYTE

Resposta

ID:	***002Ah***
Argumento:	Void

Este comando move o cursor para uma posição especificada pelo argumento ARG_BYTE_BYTE:

Tipo	*Nome*	*Descrição*
BYTE	cArg1	Linha
BYTE	cArg2	Coluna

Tabela 7 – Estrutura ARG_BYTE_BYTE

vGoToXYRef	***002Bh***
Origem:	Servidor
Descrição:	Desloca o cursor para uma posição relativa
Argumento:	ARG_BYTE_BYTE

Resposta

ID:	***002Ch***
Argumento:	Void

Ao enviar este comando, o cursor será deslocado a partir da posição anterior de x linhas e y colunas especificadas em ARG_BYTE_BYTE (tabela 7), que poderão assumir valores negativos neste comando. Por exemplo, se o cursor está na linha 1, coluna 10 e é enviado o comando vGoToXYRef com os argumentos 1 e -1, o cursor será deslocado para a linha 1+1 = **2**, coluna 10-1 = **9**.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vSetEditString	***002Fh***
Origem:	Servidor
Descrição:	Habilita / desabilita a edição de string
Argumento:	ARG_STR_BOOL_BOOL

Resposta	
ID:	***0030h***
Argumento:	Void

Através de vSetEditString habilita-se / desabilita-se o modo de edição de string.

Neste modo, o usuário pode editar uma string através do teclado do equipamento ou através de um teclado auxiliar, podendo observar sua digitação através do display. Quando o usuário pressionar <ENTER>, esta string editada será enviada para o servidor. Esta função pode ser utilizada para a solicitação de digitação de dados como senhas, por exemplo, sendo disponível uma opção para que os caracteres digitados apareçam como asteriscos no display. A string enviada por este comando será exibida no display (ou serão exibidos asteriscos no lugar dos caracteres, caso assim deseje-se) e o usuário poderá alterá-la. Pode-se também enviar este comando com uma string nula (com um NULL na primeira posição de ***String***), quando se deseja simplesmente capturar alguma informação digitada pelo usuário. O pressionamento da tecla <ESC> interromperá a edição, e o terminal enviará uma string nula para o servidor através do comando vReadEditString tratado mais adiante.

Os argumentos são os seguintes:

Tipo	*Nome*	*Descrição*
BYTE [41]	String	String a ser Editada, terminada com caractere nulo.
BOOL	bOnOff	Se for TRUE, coloca o microterminal em modo de edição da string.
BOOL	bPassWord	Se for TRUE, a edição é feita substituindo os caracteres por asteriscos na exibição no display.

Tabela 8 – Estrutura ARG_STR_BOOL_BOOL

O próprio servidor poderá interromper a edição da string, enviando o comando vSetEditString com o argumento ***bOnOff*** igual a FALSE.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vReadEditString	***002Dh***
Origem:	Microterminal
Descrição:	Envia a string editada pelo microterminal
Argumento:	ARG_STR_BOOL

Resposta

ID:	***002Eh***
Argumento:	Void

Quando a edição da string termina, com pressionamento das teclas <ENTER> ou <ESC>, o microterminal envia o comando vReadEditString (002Dh), com argumento ARG_STR_BOOL (Tabela 8). E servidor responde com vReadEditString+1 (002Eh).

Tipo	*Nome*	*Descrição*
BYTE [41]	String	String Editada terminada com caractere nulo.
BOOL	Status	TRUE se foi pressionada a tecla ENTER; FALSE se foi pressionada ESC.

Tabela 9: Estrutura ARG_STR_BOOL

O servidor poderá interromper a edição da string enviando o comando vSetEditString com o argumento ***bOnOff*** igual a FALSE. Se a tecla ESC for pressionada, além de Status = 0, será enviada a String com um caractere nulo na primeira posição.

vGetEditString	***0031h***
Origem:	Servidor
Descrição:	Retorna o estado de edição de string
Argumento:	Void

Resposta

ID:	***0032h***
Argumento:	BOOL

Para que o servidor detecte se microterminal está em modo de edição, o servidor deve enviar o comando vGetEditString (0031h). O microterminal responderá com vGetEditString+1, com o argumento BOOL indicando o modo de edição (TRUE = modo de edição ativo).

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vDispStr	***0033h***
Origem:	Servidor
Descrição:	Exibe uma string no display
Argumento:	ARG_STR
Resposta	
ID:	***0034h***
Argumento:	Void

Para o envio de string para o display, o servidor utiliza o comando vDispStr (0033h), com argumento ARG_STR (Tabela 10). E o microterminal responde com vDispStr+1 (0034h).

Tipo	***Nome***	***Descrição***
BYTE [41]	String	String exibida, terminada com caractere nulo.

Tabela 10: Estrutura ARG_STR

Se a string enviada tender a ultrapassar a última coluna da linha atual, não haverá mudança de linha automática, e a string será truncada. Tão logo a string seja exibida, o cursor será posicionado na primeira coluna subseqüente ao último caractere da string, salvo quando este coincidir com a última posição da linha, caso no qual o cursor permanecerá na última coluna da linha em questão.

vDispCh	***0035h***
Origem:	Servidor
Descrição:	Exibe um caractere no display
Argumento:	BYTE
Resposta	
ID:	***0036h***
Argumento:	Void

Para exibir caractere na posição atual do cursor no display, o servidor utiliza o comando vDispCh (0035h), com argumento BYTE contendo o código ASCII do caractere. O microterminal responderá com vDispCh+1 (0036h).

Se o cursor estiver na última coluna de uma linha, o caractere enviado irá se sobrepor ao caractere que havia antes nesta posição, ou seja, não ocorrerá avanço automático de linha.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vDispClrLn	***0037h***
Origem:	Servidor
Descrição:	Apaga uma linha no display
Argumento:	BYTE

Resposta

ID:	***0038h***
Argumento:	Void

Para limpar um linha no display, o servidor envia a mensagem vDispClrLn (0037h), com o número da linha como argumento (BYTE), e o microterminal responde com vDispClrLn+1 (0038h). O cursor permanecerá na mesma posição em que estava antes da execução do comando.

vSetBack	***0063h***
Origem:	Servidor
Descrição:	Altera a condição da iluminação de fundo (backlight)
Argumento:	BOOL

Resposta

ID:	***0064h***
Argumento:	Void

Através da mensagem vSetBack é possível ligar e/ou desligar a iluminação de fundo do display nos microterminais MT7xx com firmware versão 2.2 ou superior. Com o argumento igual a TRUE o backlight é ligado, e é desligado se o argumento for FALSE.

Atenção: o microterminal não salva o estado do backlight quando alterado via servidor. Para ajustar o estado do backlight e salvá-lo, configure-o através de um teclado auxiliar. Para maiores informações consulte o manual do microterminal.

e. Comunicação Serial

A comunicação serial dos microterminais é contemplada por 6 comandos, sendo que um único tem o microterminal como origem.

Para saber quantas seriais há disponíveis em um determinado modelo, consulte o respectivo manual ou vide nota * da tabela 2.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

vSetEnableSerial	***0039h***
Origem:	Servidor
Descrição:	Habilita / desabilita a porta serial
Argumento:	ARG_BYTE_BOOL

Resposta

ID:	***003Ah***
Argumento:	Void

Para receber e enviar informação pela serial, a respectiva porta deve estar habilitada. O servidor habilita ou não uma porta serial pelo comando vSetEnableSerial (0039h), com argumento ARG_BYTE_BOOL (Tabela 11). E o microterminal responde com vSetEnableSerial+1 (003Ah).

Para a serial COM**x**, o argumento ***COM*** vale **x-1**, ou seja:

Para a COM**1**, o argumento ***COM*** é igual a **0** (zero); para a COM**2**, ***COM*** vale **1**; para a COM**3**, ***COM*** vale **2** e para a COM4, ***COM*** vale **3**.

Todas as seriais disponíveis são habilitadas ao ligar o microterminal. Esta configuração não é salva ao desligar.

Tipo	*Nome*	*Descrição*
BYTE	COM	Byte identificador da porta serial.
BOOL	bOnOff	Se for TRUE habilita a porta serial.

Tabela 11: Estrutura ARG_BYTE_BOOL

vGetEnableSerial	***003Bh***
Origem:	Servidor
Descrição:	Retorna o estado da porta serial
Argumento:	BYTE

Resposta

ID:	***003Ch***
Argumento:	ARG_BYTE_BOOL

Para o servidor identificar o estado da porta serial, ele envia o comando vGetEnableSerial (003Bh) com o argumento BYTE indicando qual porta conforme o explicado

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

em vSetEnableSerial, e o microterminal responde com vGetEnableSerial+1 (003Ch) com argumento ARG_BYTE_BOOL (Tabela 10), onde o argumento BYTE igual ao enviado pelo servidor e o argumento BOOL indica o estado da porta serial, sendo que o estado habilitado é representado pelo valor TRUE.

vGetBinSerial	***003Dh***
Origem:	Microterminal
Descrição:	Transmissão de dados recebidos pela serial para o servidor
Argumento:	ARG_COM_BIN_BYTE

Resposta	
ID:	***003Eh***
Argumento:	Void

Quando a porta serial está habilitada, e o microterminal recebe algum dado pela serial, seu conteúdo é armazenado num buffer interno e enviado ao servidor pelo comando bGetBinSerial (003Dh), com argumento ARG_COM_BIN_BYTE (Tabela 12). E o servidor responde com bGetBinSerial+1.

O argumento ***Bin*** sempre possui 256 bytes, mesmo que apenas um byte tenha sido lido da porta serial. Portanto, o servidor deve sempre ler o argumento ***Tam*** e ler os primeiros ***Tam*** bytes do argumento ***Bin***, descartando os demais.

Tipo	*Nome*	*Descrição*
BYTE	COM	Byte identificador da porta serial conforme o explicado em vSetEnableSerial.
BYTE[256]	Bin	Buffer Binário.
BYTE	Tam	Quantidade de bytes lidos.

Tabela 12: Estrutura ARG_COM_BIN_BYTE

Uma vez que a taxa de transmissão entre o servidor e o terminal é muito maior que a taxa de transmissão entre o terminal e o periférico serial, não é necessário um controle de fluxo para os dados que trafegam no sentido microterminal -> servidor, pois não é possível acontecer um estouro de buffer neste cenário.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

SendBinSerial	***003Fh***
Origem:	Servidor
Descrição:	Transmissão de dados do servidor para a serial do microterminal
Argumento:	ARG_COM_BIN_BYTE

Resposta

ID:	***0040h***
Argumento:	BOOL

Quando a porta serial está habilitada, o servidor pode transmitir dados pela porta serial do microterminal através do comando bSendBinSerial (003Fh), com argumento ARG_COM_BIN_BYTE (Tabela 12), e o microterminal responde com bSendBinSerial+1 (0040h), com argumento BOOL.

Quanto ao comportamento da serial, existe uma distinção entre os modelos GE750/GE760 e os modelos MT740/MT720.

Modelos GE750/GE760

Estes modelos sempre retornam bSendBinSerial+1 com o argumento BOOL igual a TRUE logo após o comando proveniente do servidor ser recebido.

Modelos MT740/MT720

Os modelos MT740 e MT720 retornam bSendBinSerial+1 com o argumento BOOL igual a FALSE se o servidor tentar transmitir dados para uma COM que o equipamento não tem ou se o buffer interno não tiver mais espaço para receber os dados. O envio desta resposta por parte do microterminal ocorre em um dos dois seguintes instantes (o que ocorrer primeiro):

a) Assim que os bytes enviados pelo servidor sejam transmitidos pela serial.

b) Assim que o servidor envie mais um bSendBinSerial.

Com a característica citada no item (a), é possível criar um controle de fluxo entre o servidor e o microterminal. Este controle será necessário quando estiver previsto o envio de mais de 512 bytes em um tempo menor do que o necessário para que estes bytes sejam transmitidos pela serial do microterminal. Deve-se lembrar que a taxa de transmissão (baud rate) da serial varia de 300 a 115.200 bits/s, enquanto a rede pode alcançar 100Mbits/s, o que indica que seja necessário um controle para que não ocorra estouro de buffer neste cenário, pois a informação chega ao terminal numa velocidade muito maior do que sai.

Para tanto, basta que o servidor transmita um pacote de dados com bSendBinSerial e espere pela resposta bSendBinSerial+1 do microterminal antes de transmitir mais dados. Assim, não haverá risco de estouro de buffer mesmo no envio contínuo de um volume de dados da ordem de megabytes para a serial.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

A característica do item (b) serve para compatibilizar com os microterminais GE750/GE760. Caso o servidor não espere pela resposta bSendBinSerial+1 e envie outro pacote de dados, o terminal responderá imediatamente com o bSendBinSerial+1 correspondente ao pacote anterior. Desta maneira, sempre haverá uma resposta do microterminal para cada comando do servidor. Este método é indicado apenas para pequenos volumes de dados conforme o exposto anteriormente.

vSetSetupSerial	***0041h***
Origem:	Servidor
Descrição:	Configura uma porta serial do microterminal
Argumento:	ARG_COM_SETUPSERIAL
Resposta	
ID:	***0042h***
Argumento:	BOOL

Para configurar cada uma das portas seriais, o servidor envia a mensagem bSetSetupSerial (0041h), com argumento ARG_COM_SETUPSERIAL (TABELA 12). E o microterminal responde com bSetSetupSerial+1 (0042h), com argumento BOOL, indicando se houve sucesso na configuração do microterminal.

Os microterminais retornam TRUE em caso de sucesso e FALSE em caso contrário.

Tipo	***Nome***	***Descrição***
BYTE	COM	Byte identificador da porta serial conforme o explicado em vSetEnableSerial.
DWORD	Baud	Define a velocidade de transmissão serial.
WORD	Bits	Define o número de bits para cada dado. Varia de 5 a 8.
WORD	Parity	Define a paridade utilizada. Utiliza as constantes listadas pela a Tabela 14.
WORD	Stops	Define o número de stop bits. Deve ser 1 ou 2.
BYTE	handshaking	Define o tipo de handshaking utilizado. Utiliza as constantes listadas pela Tabela 15.

Tabela 13: Estrutura ARG_COM_SETUPSERIAL

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

Valores válidos de baud rate: 300, 1.200, 2.400, 4.800, 9.600, 19.200, 38.400, 57.600 e 115.200 bauds.

Constante	*Paridade*
0	Nenhuma
1	Ímpar (Odd)
2	Par (Even)
3	Mark
4	Space

Tabela 14: Constantes de paridade

Constante	*Descrição*
0	Sem handshaking.
1	Com handshaking por hardware (por CTS e RTS).

Tabela 15: Constantes de handshaking

vGetSetupSerial	***0043h***
Origem:	Servidor
Descrição:	Solicita a configuração da porta serial do microterminal
Argumento:	BYTE

Resposta	
ID:	***0044h***
Argumento:	ARG_COM_SETUPSERIAL

Para o servidor conhecer a configuração de uma porta serial, o servidor envia um comando vGetSetupSerial (0043h), com argumento BYTE identificando a porta serial de interesse conforme o explicado em vSetEnableSerial. Então o microterminal responde com vGetSetupSerial+1 (0044h), com argumento ARG_COM_SETUPSERIAL.

Os microterminais MT740 e MT720 não respondem com vGetSetupSerial+1 caso o servidor solicite a configuração de uma porta serial inexistente.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

f. Comunicação Paralela (LPT1)

A comunicação paralela do microterminal é controlada por três comandos gerados pelo servidor.

Por motivos de compatibilidade, o microterminal MT740 responderá a estes comandos com a indicação de erro de impressora.

Os modelos GE760 e MT720 não retornarão resposta alguma caso recebam estes comandos.

cInitPrn	***0045h***
Origem:	Servidor
Descrição:	Incializa a impressora paralela
Argumento:	Void

Resposta	
ID:	***0046h***
Argumento:	BYTE

Para iniciar a impressora paralela, limpando de buffer interno de impressão, o servidor envia o comando cIniPrn (0045h). O microterminal retorna com cInitPrn+1 (0046h), com o argumento BYTE informando o status da impressora, conforme a descrição dos bits na tabela 15.

cGetStatusPrn	***0047h***
Origem:	Servidor
Descrição:	Obtém o estado da impressora paralela
Argumento:	Void

Resposta	
ID:	***0048h***
Argumento:	BYTE

Para retornar somente o estado da impressora, o servidor envia o comando cGetStatusPrn (0047h). E o microterminal retorna com cGetStatusPrn+1 (0048h) com argumento BYTE conforme a tabela 16.

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520

cSendPrn	***0049h***
Origem:	Servidor
Descrição:	Envia um conjunto de bytes para a paralela.
Argumento:	ARG_STR_BYTE

Resposta

ID:	***004Ah***
Argumento:	BYTE

Para imprimir pela paralela, o servidor envia o comando cSendPrn (0049h), com argumento ARG_STR_BYTE (Tabela 16). E microterminal retorna cSendPrn+1 (004Ah), com argumento BYTE que representa o estado da impressora.

Bit	***Função***
0	Esgotado o tempo de espera permitido (TIME OUT)
1 e 2	Não utilizado
3	Erro da entrada e saída (I/O ERROR)
4	Impressora selecionada (PRINTER SELECT)
5	Ausência de papel (OUT OF PAPER)
6	Impressora em Linha (ACK)
7	Impressora desocupada (NOT BUSY)

Tabela 16: Definição do byte de status da impressora.

Tipo	***Nome***	***Descrição***
BYTE[256]	Bin	Buffer Binário
BYTE	Tam	Tamanho do Buffer

Tabela 17: ARG_STR_BYTE

Rua Guaicurus, 145 – Vl. Conceição – Diadema - SP – CEP 09911-630
Tel.: 11 2173-6500 / Fax.: 11 2173-6520