

RASTER TECHNOLOGIES
MODEL ONE
ALPHANUMERIC TERMINAL EMULATION
PROGRAMMING GUIDE

Revision 3.0 August 2, 1985

Alphanumeric Terminal Emulation Programming Guide

ALPHANUMERIC TERMINAL EMULATION PROGRAMMING GUIDE

August 2, 1985

Copyright 1985 by Raster Technologies, Inc. All rights reserved. No part of this work covered by the copyrights herein may be reproduced or copied in any form or by any means--electronic, graphic, or mechanical, including photocopying, recording, taping, or information and retrieval systems--without written permission.

NOTICE:

The information contained in this document is subject to change without notice.

RASTER TECHNOLOGIES DISCLAIMS ALL WARRANTIES WITH RESPECT TO THIS MATERIAL (INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE), EITHER EXPRESS OR IMPLIED. RASTER TECHNOLOGIES SHALL NOT BE LIABLE FOR DAMAGES RESULTING FROM ANY ERROR CONTAINED HEREIN, INCLUDING, BUT NOT LIMITED TO, FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS MATERIAL.

This document contains proprietary information which is protected by copyright.



Alphanumeric Terminal Emulation Programming Guide

Table of Contents

1.0 INTRODUCTION	4
2.0 ALPHANUMERIC TERMINAL EMULATION COMMANDS	6
3.0 WINDOW DEFINITION, SELECTION, AND DELETION	7
4.0 MOVING AROUND THE WINDOW	9
5.0 WINDOW AND TEXT CHARACTERISTICS	10



1.0 INTRODUCTION

This manual describes the alphanumeric terminal emulation commands. These commands use the Model One monitor as an alphanumeric terminal, thus allowing a Model One workstation to be operated with the Model One, a local keyboard, and a monitor. Under these circumstances, no local terminal is needed.

This manual includes 5 sections, as described below. Section 2.0 Alphanumeric Terminal Emulation Commands

This section describes this command:

ALPHEM Enables or disables terminal emulation; routes text from the keyboard and host to the selected window.

Section 3.0 Window Definition, Selection, and Deletion

This section describes the commands to set up the scrolling text windows as desired.

DEFWIN Defines the size and position of up to 8 scrolling text windows.

SELWIN Selects a previously defined window.

DELWIN Deletes a previously defined window.

GETWIN Returns the number of the currently selected window.

Section 4.0 Moving Around the Window

This section describes the commands for moving around the selected text window:

DIRCUR Moves the cursor to a specified character position.

GETPOS Returns the character position of the cursor.

MOVCUR Moves the cursor to a specified Model One coordinate position.

GETCUR Returns the Model One coordinate position of the cursor.

HOME Moves the cursor to character position (0,0).

SETCUR Enables or disables drawing of the cursor.

Section 5.0 Window and Text Characteristics

This section describes the commands used to define the window attributes, such as color and text size:

Alphanumeric Terminal Emulation Programming Guide

BOLD Enables or disables drawing of bold text.

OVRSTK Enables or disables overstriking of text.

SCROLL Enables or disables text scrolling.

SETSIZ Sets x and y scaling of text.

WRAP Enables or disables text wraparound.

A single command is used to start and stop the alphanumeric terminal emulator; it is described in this section. Use of the windows is described in detail in the following sections. When the Model One is factory-configured for a keyboard attached to the KEYBSIO port, the Model One coldstarts with window 0 defined and selected; all window attributes, such as bolding and text size, are set to their defaults. No other windows are defined. Any text that is typed at the keyboard will go to the text window.

The command ALPHEM ON is also executed during coldstart. This instructs the Model One to send all text from the keyboard to the window; if you wish to turn off the alphanumeric text emulation environment, you can use the command ALPHEM 0 or ALPHEM OFF.

3.0 WINDOW DEFINITION, SELECTION, AND DELETION

Up to eight scrolling text windows may be defined and used for alphanumeric terminal emulation. Window 0 is the hardware-scrolled window; it always uses the entire monitor area. The screen origin register (CREG 4) is used for scrolling, so that changes to CREG 4 will also affect window 0.

Up to eight software-scrolled windows--windows 1 through 7--may also be defined and used for alphanumeric text. These windows may be any size.

Each alphanumeric window is defined by number. When the window is defined, two diagonal corners are specified, thus giving the window size. (Of course, window 0 always occupies the entire screen, and so these corners are ignored for window 0.) The x and y scaling of the text for the window can also be defined.

Finally, each window has an associated write mask, which is specified the same way as with the WRMASK command.

The DEFWIN window,x1,y1,x2,y2,x size,y size,bitm,bankm command defines the text window specified by window. Any parameter with a value of zero (0) instructs the DEFWIN command to use the default or existing value. (There is no default for the window corners.) (This is shown in the examples below.) x1,y1 and x2,y2 specify the corners of the text window. Windows 1 through 7 can be any size; window 0 will always use the entire screen. x size and y size specify the text size. x size and y size are normally set to 1,1 (2,2 for the Model One/40 and the Model One/25 in 1K mode); this is equivalent to specifying a text size of 16 with the TEXTC command. x size and y size of 2,2 are equivalent to a text size of 32 with the TEXTC command. x size and y size may be different, allowing tall skinny text or short fat text.

bitm and bankm specify the write mask and are used exactly as they are used in the WRMASK command. Note: because a value of zero for any parameter instructs the DEFWIN command to use the default or existing value, it is impossible to specify a write mask of 0,0. This means, of course, that the write mask can never be set so that no information can be written in the alphanumeric windows.

Each window has a foreground, background, and cursor color. A set of three value registers is used for each window to specify these colors. Window 0 uses value registers 16, 17, and 18; window 1 uses value registers 19, 20, and 21; and so on through window 7, which uses value registers 40, 41, and 42. Foreground and background colors each can only be one of 64 colors.

The foreground color is used for text; the background color occupies the rest of the window. The cursor is an XOR cursor, so that the color in the value register is XORed with the background color to display the cursor. For example, if the background color is red and the foreground color is black, the cursor color value register must be set to cyan for the user to see a white cursor.

Alphanumeric Terminal Emulation Programming Guide

For example, to define window 1 to occupy the lower area of the monitor, use all bit planes and banks for text, and have a normal text size, you would use the command:

```
DEFWIN 1 -640,-512 640,-450 1,1 0,0
```

for a Model One/40 or a Model One/25 in 1K mode. For a Model One/60, you could use this command:

```
DEFWIN 1 -389,-288 389,-240 2,2 0,0
```

Then, to set up the desired colors for background, foreground and cursor, you would set value registers 19, 20, and 21.

Now, if you wish to redefine the area of window 1, you can use the command:

```
DEFWIN 1 -100,-100 -50,-50 0,0 0,0
```

which redefines the area of the window, and sets the text size and write mask back to the default values.

Once a window has been defined, it must be selected before information can be written to the window. Only one window can be selected at a time, and selection of the window "unselects" the previously selected window. The command SELWIN window selects a window; for example, SELWIN 2 selects window 2.

DELWIN window deletes the definition of a window. The window to be deleted cannot be selected.

GETWIN returns the number of the selected window. -1 is returned if no window is selected.

4.0 MOVING AROUND THE WINDOW

The alphanumeric text windows include a cursor. This cursor can be moved to any character position--row and column--or to a Model One coordinate position. For windows 1 through 7, the Model One coordinate position is rounded to the nearest character position; in window 0, the coordinate position is not rounded.

The commands DIRCUR *x,y* and HOME are used to move the cursor to character positions. DIRCUR will move the cursor to an arbitrary position; if it is outside the window, it will move the cursor as far as possible in the right direction. HOME always moves the cursor to character position (0,0)--the first row and first column. GETPOS returns the character position of the cursor.

MOVCUR moves the cursor to a Model One coordinate position; GETCUR returns the Model One coordinate position of the cursor.

SETCUR enables or disables the cursor.

5.0 WINDOW AND TEXT CHARACTERISTICS

In using the scrolling text windows, the boldness of the text, the size of the text, whether the text can be overstruck, and whether the text will wrap or scroll can all be set as desired.

BOLD ON or BOLD 1 turns on bolding of text. The boldness of text is in proportion to its size, as determined by the SETSIZ or DEFWIN commands. Bold text is the default. BOLD OFF or BOLD 0 disables bolding of text.

SETSIZ xscale,yscale specifies the x and y scaling of text. The default is 1,1. The new size overrides whatever was defined by the DEFWIN command; for windows 1 through 7, the entire window is re-initialized and any text currently in the window is lost. If either xscale or yscale is zero, the current value will be used.

OVRSTK ON or OVRSTK 1 enables overstriking of characters on the line. Previous characters in the line may then be overstruck without erasing that character. OVRSTK OFF or OVRSTK 0 disables overstriking; this is the default. For windows 1 through 7, only the last character in a given position will remain after the window is scrolled.

SCROLL ON and WRAP ON enables scrolling and wraparound of text. If scrolling is off—SCROLL OFF or SCROLL 0—characters wrap from the end of the bottom line to the beginning of the top line. If wraparound is off, the cursor stays in the last character position on the line until a carriage return is typed. The default is scrolling and wraparound of text.

In addition to these commands, a set of control characters can be used to move around the window and perform line functions. These are:

[CTRL-B]	Clear entire current line
[CTRL-E]	Move cursor up one pixel (window 0 only)
[CTRL-H]	Backspace
[CTRL-I]	Horizontal tab to next multiple of 8 characters
[CTRL-J]	Linefeed
[CTRL-K]	Vertical tab one line (upward linefeed)
[CTRL-L]	Formfeed: clears entire window
[CTRL-M]	Carriage return
[CTRL-T]	Clear to end of line
[CTRL-V]	Move cursor down one pixel (window 0 only)
[CTRL-Y]	Move cursor left one pixel (window 0 only)
[CTRL-\]	Home (move to character position 0,0)
[CTRL-]	Move cursor right one character position
[CTRL- <u>_</u>]	Move cursor right one pixel (window 0 only)

TECHNICAL NOTE
TEKTRONIX 4014 EMULATION ON
THE RASTER TECHNOLOGIES
MODEL ONE GRAPHICS CONTROLLER

Revision 3.0 December 2, 1983



TEKTRONIX EMULATOR TECHNICAL NOTE
December 2, 1983

Copyright 1983 by Raster Technologies, Inc. All rights reserved. No part of this work covered by the copyrights herein may be reproduced or copied in any form or by any means--electronic, graphic, or mechanical, including photocopying, recording, taping, or information and retrieval systems--without written permission.

NOTICE:

The information contained in this document is subject to change without notice.

RASTER TECHNOLOGIES DISCLAIMS ALL WARRANTIES WITH RESPECT TO THIS MATERIAL (INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE), EITHER EXPRESS OR IMPLIED. RASTER TECHNOLOGIES SHALL NOT BE LIABLE FOR DAMAGES RESULTING FROM ANY ERROR CONTAINED HEREIN, INCLUDING, BUT NOT LIMITED TO, FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS MATERIAL.

This document contains proprietary information which is protected by copyright.



1.0 INTRODUCTION

The Model One TEKEM command enters the TEKTRONIX 401X emulator for the Raster Technologies Model One. This emulator emulates the Tektronix models 4006, 4010, 4012, 4013, 4014, 4015, 4016, and 4054. When the Tektronix emulator is invoked, the Model One command set is unavailable and the 401X emulator is entered. Any host program/graphics package combination (e.g., Plot 10, GCS, DIGRAF) that has been used to drive the 401x terminal will then be able to drive the Model One.

This technical note is divided into the following sections:

TEKEM Command Syntax: section 2.0
Standard Emulation: section 3.0
Extensions to 401x Emulation: section 4.0
Escape Sequences: section 5.0

Each of the latter three categories is subdivided into host-controlled functions and keyboard-controlled local functions.

It is assumed that the reader of this document is familiar with the 4014 command set and the Model One.

NOTE: in using the Tektronix emulator (TEKEM), the ESCAPE key is used to invoke command sequences. Thus, if you are using the alphanumeric terminal for ESCAPE sequences, he may have some problems. In this case, these commands will correct the problem:

SPCHAR WARM OFF
SPCHAR BREAK OFF
SPCHAR KILL ON #15

The kill character is now CTRL-U; BREAK and WARMstart keys are disabled.

To use ESCAPE sequences from the host computer, you can disable all CTRL characters with the command:

SYSCFG SERIAL HOSTSIO CTRL OFF

This configuration is the default.

2.0 COMMAND SYNTAX

The Tektronix emulator is invoked with the Model One command:

TEKEM <bitflag>

where <bitflag> is an 8-bit parameter, the bits of which are set as follows:

Bit number (0=LSB) :	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-
	0	0	0	P	S	D	G	G

Here,

- S signals screen mode: it is 0 for normal screen mode and 1 if the terminal is to send an XOFF (^S) to the host upon a page-full condition. In the latter case, an XOFF (^Q) will be sent after the user clears the page.
- D signals duplex mode: it is 0 for full duplex and 1 for half duplex.
- GG signals graphics terminating character mode: it is 0 if no terminating character is to be sent after sending coordinates, it is 01 if a carriage return is to be sent, and it is 10 (=2 decimal) if a carriage return followed by EOT is to be sent.
- P signals that different line types will be treated differently. If P=0, all line types will be INSerted (pixel function INS). This is the default.

If P=1, dotted and dashed lines use pixel function CONDitional. Solid lines may be INSerted, XORed, or ORed, as determined by the ESCAPE sequences in section 5.

Thus, for example, TEKEM 6 would signal normal screen mode, half duplex mode, and <CR><EOT> to follow coordinate data.

3.0 STANDARD EMULATION

3.1 General Description

The TEKEM emulator is designed to provide 401X functionality. However, some "gray areas" arise from physical differences between the Model One and the Tektronix terminals. These differences are described in the next section.

3.2 Host-Controlled Functions

3.2.1 Addressability

The TEKEM emulator will respond to either 10-bit (1024x1024) or 12-bit (4096x4096) coordinate addressing. In the 12-bit mode, the "extra" byte sent by the host is ignored; the emulator will expect

HIY, LOY [EXTRA (ignored)], HIX, LOX.

In other words, the emulator will accept 12-bit addressing but supports 10 bits of address.

3.2.2 Graphic Input Mode (GIN)

Upon receipt of the sequence <ESC> <SUB>, the emulator will respond by displaying the crosshair. GIN will be terminated by the user typing any character. This character will NOT be displayed on the screen. GIN termination will generate 5 bytes in the following sequence:

Byte 1	character (GIN terminator)
Byte 2	HIX
Byte 3	LOX
Byte 4	HIY
Byte 5	LOY

This sequence will be terminated by a CR, a CR EOT, or by nothing, depending on the two "GG" bits of the TEKEM parameter.

3.2.3 Alpha Cursor Address

With the emulator in alpha mode, sending ESC ENQ from the host will return the terminal status and lower-left coordinate of the alpha cursor to the host. The first byte returned is the terminal status; after that, the bytes are identical to GIN. The emulator stays in alpha mode after returning the information.

3.2.4 Graphic Beam Address

With the emulator in graphics mode, sending ESC ENQ from the host will return the terminal status and graph beam coordinates to the host. The byte sequence is identical to that of the alpha cursor location. After returning the information, the emulator stays in graphics mode.

3.2.5 Crosshair Address

When displaying the crosshair cursor, sending the ESC ENQ sequence from the host returns the intersection point of the crosshair to the host. A 15 msec delay must occur between ESC SUB (enabling the crosshair) and ESC ENQ. Four address bytes are returned. After sending the bytes, the emulator returns to alpha mode.

3.2.6 Alpha/Graphics Mode

The emulator will be in alpha mode when it is first invoked. Upon receipt of US or CR, the emulator will return to alpha mode from whatever mode it was in. In alpha mode, all bytes received are treated as alphanumeric characters. After receiving GS, all further bytes received are treated as graphic information (addresses). The first vector drawn after receipt of GS is a MOVE (invisible); subsequent vectors are DRAWs (visible).

3.2.7 Line Fonts

Sending ESC, followed by the ASCII Decimal Equivalent (ADE) numbers below, will generate all subsequent vectors in the selected line font.

<u>ADE</u>	<u>Line Font</u>
96	solid (default)
97	dotted
98	dot-dash
99	short dash
100	long dash

3.2.8 Character Size

Sending ESC followed by the appropriate ADE will select the character size.

<u>ADE</u>	<u>Character Size</u>
56	large
57	medium
58	small
59	very small

3.2.9 Point Plot Mode

Sending FS will enter Point Plot Mode. In this mode, points are addressed normally; the endpoint (the addressed point) will be drawn.

3.3 Keyboard-Controlled Functions

3.3.1 Page Key

To emulate the PAGE/RESET key of the 401x, press CTRL-W followed by P (for page). The screen is erased and the alpha cursor moves to HOME (the upper left-hand corner).

3.3.2 Other Keyboard Functions

With the exception of margin (3-way toggle) and hard-copy, all other keyboard functions are supported.

4.0 EXTENSIONS TO THE 40XX COMMAND SET

4.1 Overview

In addition to providing all the features of the 401x graphic terminals, the emulator provides access to the powerful command set of the Model One. Some of the more popular commands such as changing the background color, changing the vector color, XHAIR color, areafill, and so forth, are made directly available. Most of these functions may be invoked either from the host or by keyboard entry. On the host, the commands may be sent through a routine that outputs ASCII strings (TOUTST for example in PLOT 10), or the user may write a set of routines that output specific command strings. Each of these extensions will be discussed.

4.2 Host-Controlled Functions

Throughout this section, the user must be aware that all parameters are ASCII numbers. For example, num in sections 4.2.1 and 4.2.2 is the ASCII numbers '0' through '4'.

4.2.1 Selected-Screen Quadrant

ESC DC4 (CTRL-T) 0 num selects the quadrant on the screen based on num. The range of num is 0 to 4. Number 0 selects the whole screen, while numbers 1-4 select the corresponding screen quadrants. Quadrant 1 is upper left, 2 is upper right, 3 is lower left, and 4 is lower right. When a quadrant is selected, all incoming data is scaled appropriately to fit in that quadrant. Any display manipulation command, such as erase, received from the host affects only that quadrant, thereby providing selective quadrant erasing.

The host binary command stream is:

[1BH] [14H] [30H] [num] (4 bytes)

This function is invoked by sending:

[ESC] [DC4 (CTRL-T)] [0] [num]

The range for num is 0 to 4.

4.2.2 View Screen Quadrant

ESC DC4 1 num selects the quadrant for viewing on full screen; each quadrant will be multiplied by a factor of 2 to fill the entire screen (2X zoom).

The host binary command stream is:

[1BH] [14H] [31H] [num] (4 bytes)

This function is invoked by sending:

[ESC] [DC4] [1] [num]

The range of num is 0 to 4.

4.2.3 Monitor Toggle

ESC DC4 2 toggles between the monitor and alpha terminal for display of text.

The host binary command stream is:

[1BH] [14H] [32H] (3 bytes)

This function is invoked by sending:

[ESC] [DC4] [2]

4.2.4 Termination

ESC DC4 3 terminates Tektronix emulation and returns to the Model One command environment.

The host binary command stream is:

[1BH] [14H] [03H] (3 bytes)

This function is invoked by sending:

[ESC] [DC4] [3]

4.2.5 Change Background Color

ESC DC4 4 N sets the background color to N. This color is used the next time the screen is cleared. The color is specified as:

MSB 7 6 5 4 3 2 1 0 LSB
X X R R G G B B

The host binary command stream is:

[1BH] [14H] [34H] [highN] [lowN] (5 bytes)

The function is invoked by sending:

[ESC] [DC4] [4] [N]

Note that N must be sent as two HEXASCII digits.

4.2.6 Change Foreground Color

ESC DC4 5 N sets the foreground color to N immediately. (N is described in the previous section.)

The host binary command stream is:

[1BH] [14H] [35H] [highN] [lowN] (5 bytes)

The function is invoked by sending:

[ESC] [DC4] [5] [N]

4.2.7 Change Crosshair Color

ESC DC4 6 N sets the crosshair color to N. (N is described in section 4.2.5.)

The host binary command stream is:

[1BH] [14H] [36H] [highN] [lowN] (5 bytes)

This function is invoked by sending:

[ESC] [DC4] [6] [N]

4.2.8 Area Fill

ESC DC4 7 performs an area fill, using the current color, starting at the current cursor position and bounded by any pixel in a different color. (See the AREAL command in the Model One documentation for further details.)

The host binary command stream is:

[1BH] [14H] [37H] (3 bytes)

This function is invoked by sending:

[ESC] [DC4] [7]

4.2.9 Change Look-Up Table Entry

ESC DC4 8 Entry Value changes a single look-up table entry to a specific value. Entry ranges from 0 to 3FH; value ranges from 0 to 3FH. (See the LUT8 command in the Model One documentation for further details.)

The host binary command stream is:

[1BH] [14H] [38H] [highE] [lowE] [highRV] [lowRV]
[highGV] [lowGV] [highBV] [lowBV] (11 bytes)

This function is invoked by sending:

[ESC] [DC4] [8] [Entry][Value]

4.2.10 Redraw and No Redraw: Stroke Terminal Simulation

ESC DC4 10 turns on redraw of vectors (simulation of a stroke terminal); ESC DC4 9 turns off redraw.

The host binary command stream is:

```
[1BH] [14H] [41H]   (3 bytes)   for REDRAW
[1BH] [14H] [39H]   (3 bytes)   for NO REDRAW
```

These commands are invoked by sending:

```
[ESC] [DC4] [A]     for REDRAW
[ESC] [DC4] [9]     for NO REDRAW
```

All vectors drawn after REDRAW are affected, until NO REDRAW is invoked.

4.2.11 Text Color

ESC DC4 11 N sets the text color to N. (N is described in section 4.2.5.)

The host binary command stream is:

```
[1BH] [14H] [42H] [highN] [lowN]   (5 bytes)
```

The command is invoked by sending:

```
[ESC] [DC4] [B] [N]
```

4.2.12 Redraw Rate

This command sets the rate of redraw for the redraw command (see section 4.2.10, above). ESC DC4 12 N sets the redraw rate to N. N=0 is not permitted.

The host binary command stream is:

```
[1BH] [14H] [43H] [highN] [lowN] (5 bytes)
```

The command is invoked by sending:

```
[ESC] [DC4] [C] [N]
```

The default rate is N=20.

4.3 Locally Controlled Functions

All local commands must be entered from the alpha terminal. Each local command is preceded by CTRL-W and is followed by the command letter and any required parameters.

These commands are available:

S num Set active sub-screen

V num Set viewing area to num

M Toggle text output between Model One and alphanumeric terminal
R Terminal Tektronix emulator and return to Model One
P Erase current subscreen (same as ESC FF)
B N Set background color to N
F N Set foreground color to N
C N Set character size to N
H Provide HELP (a list of available local commands)

5.0 ESCAPE SEQUENCES

This section compares the Model One Tektronix Emulator (TEKEM) and the actual Tektronix 4014 escape sequences.

Escape Sequence	TEKEM	TEK4014
ESC NUL		
ESC DC4 (^T)	Enter extension sequence	Stay in escape mode, interpret following characters as part of escape sequence
ESC ENQ (^E)	Same as TEK4014	Terminal status request
ESC BEL (^G)	Ring terminal bell	Ring terminal bell
ESC BS (^H)	Move left, no erase	Move left one space
ESC HT (^I)	Not applicable	Move one space right
ESC LF (^J)	Same as TEK4014	Set LCE
ESC VT (^L)	Not applicable	Move one line up
ESC FF (^L)	Same as TEK4014	Erase display; home; clear
ESC CR (^M)	Same as TEK4014	Set LCE; ignore further CRS
ESC SI (^O)	Not applicable	ASCII character set
ESC SO (^N)	Not applicable	Alternate character set
ESC ETB (~W)	Access local command set from A/N terminal	Select LOCAL commands
ESC CAN (^X)	Not applicable	Set bypass mode
ESC SUB (^Z)	Same as TEK4014	Set GIN, crosshair, and bypass modes
ESC ESC (csK)	Not applicable	Return to ANSI
ESC FS* (csL)	Same as TEK4014	Special point plot mode
ESC GS* (csM)	Same as TEK4014	Graph mode with dark vectors
ESC RS* (csN)	Not applicable	Incremental plot mode
ESC US* (csO)	Same as TEK4014	Alpha mode
ESC 8	78 characters, 46 lines	74 characters, 35 lines
ESC 9	86 characters, 46 lines	81 characters, 38 lines
ESC :	129 characters, 70 lines	121 characters, 58 lines
ESC ;	140 characters, 76 lines	133 characters, 64 lines
ESC (96), (104), (112)	Solid line, PIXFUN INS	
ESC (97), (105), (113)	Dotted line, PIXFUN COND	
ESC (98), (106), (114)	Dot-dash line, PIXFUN COND	
ESC (99), (107), (115)	Short dash line, PIXFUN COND	
ESC (100), (108), (116)	Long dash line, PIXFUN COND	
ESC (101), (109), (117)	Solid line, PIXFUN XOR	
ESC (102), (110), (118)	Solid line, PIXFUN OR	
ESC (103), (111), (119)	Solid line, PIXFUN INS	

*FS, GS, US, and RS may be sent with or without a preceding ESC.

In addition, the Model One Tektronix emulator does not support page full conditions; it does not stop displaying alphanumeric information when the screen is full. It does not honor the "make hard copy" request. The VIEW/HOLD operation no longer exists.



Raster Technologies

MODEL ONE/80

VT100 EMULATION GUIDE

Revision 1.0

August 26, 1985

Raster Technologies, Inc.
9 Executive Park Drive
North Billerica, MA 01862
(617) 667-8900



Chapter 1: Introduction to Using the VT100 EmulatorOVERVIEWPurpose

The VT100* emulator for the Model One/80 allows you to use the Model One/80 as a VT100 terminal. Thus you can operate a Model One/80 workstation with the Model One/80, a local keyboard, and a monitor. You do not need a local terminal.

Description of VT100 Window

The VT100 emulator uses one software scrolled text window. You can define the placement and text size for the window.

Limitations

The VT100 emulator has several limitations, due to limitations in architecture and available RAM.

- The maximum effective speed is 1200 baud. Faster baud rates may be selected, but will be bottlenecked by the code.
- The VT100 emulator has no dedicated bit planes. The user must control image memory so that the emulator does not overwrite it.
- The emulator does not include the VT100 extensions for the Model One/10. These extensions are listed in Chapter 2.
- The emulator does not include some of the features of the standard VT100 terminal, as described on the next page.

* VT100 is a trademark of Digital Equipment Corporation.



OVERVIEW, continuedStandard VT100 Features Which Are Not Implemented

The VT100 emulator does not have a SET-UP mode. You can, however, select many of the standard VT-100 SET-UP features by using the SYSCFG command or by sending escape sequences from the host.

The following SET-UP features are preset; you cannot change them.

<u>Feature</u>	<u>Setting</u>
Characters Per Line	80
Scroll	jump
Autorepeat	enabled
Cursor	underscore
Margin Bell	disabled
Key Click	enabled
Character Set	U.S.
Local Echo	disabled

You can set the baud rate, but the transmit speed and receive speed must be the same.

In addition, the VT100 emulator does not recognize escape sequences that control

- character size
- testing
- character set selection (no graphics characters), and
- mode for full screen reverse video.

Refer to Chapter 2 for a complete list of unrecognized escape sequences.

Compatibility with Alpha Emulator

You can use the VT100 emulator along with the existing alphanumeric terminal emulator.

The VT100 emulator uses window 8 of the alpha emulator.

Note: You do not need to understand how to use the alpha emulator in order to use the VT100 emulator.

Contents

The rest of Chapter 1 is organized into the following sections:

- System Configuration for Use of VT100 Emulator
- Definition of the VT100 Window: Placement and Text Size
- Controlling Image Memory
- Color and Video Attributes
- Memory Requirements for VT100 Emulation, and
- Relation of VT100 Emulator to Alpha Emulator.

SYSTEM CONFIGURATION FOR USE OF VT100 EMULATORIntroduction

You can configure your Model One/80 so that it powers up with the VT100 window defined and selected. Alternatively, you can configure your system to power up into the alpha emulator, or no emulator at all. This section describes the different ways that you can configure your system.

System Configurations

You can configure your system so that you powerup into the VT100 window, the alpha window, or no window. Use one of the following commands, followed by the SAVCFG command.

SYSCFG EMULATE ALPHA

Selects alpha window 0 for powerup and user display.

SYSCFG EMULATE OFF

Selects ALPHASIO for powerup and user display.

SYSCFG EMULATE VT100K

Selects the VT100 window for powerup and user display with keyboard supplied by Raster Technologies.

DEFINITION OF THE VT100 WINDOW: PLACEMENT AND TEXT SIZEIntroduction

This section describes the default powerup characteristics of the VT100 window, and how you can change them.

Definition of VT100 Window at Powerup

If your system is configured to power up into the VT100 window, then on powerup the VT100 window is defined to occupy the lower 8 bits of image memory from (-480,479) to (479,-480).

The VT100 window displays 80 columns and 24 rows of text.

The text size on powerup is 40 pixels high by 12 pixels wide.

Note: If you are familiar with alpha window commands, the Model One/80 performs the equivalent of the following commands at coldstart:

```
DEFWIN 8 -480 479 479 -480 2 4 255 7
SELWIN 8
ALPHEM ON
```

Redefining the VT100 Window

You can use the DEFWIN command to change the placement and text size of the window.

To change the window placement, you specify the upper left-hand coordinates.

To change the text size, you specify the x and y scaling factors.

The lower right-hand corner, and thus the physical size of the window, is automatically calculated from the text size.

Caution

If you execute the DEFWIN 8 command while the VT100 window is selected, the command reinitializes the entire window, erasing the text buffer and placing the cursor in character position (row 1, column 1).

Be careful not to define an upper left-hand corner and a text size that would extend the window beyond the screen.

DEFINITION OF THE VT100 WINDOW: PLACEMENT AND TEXT SIZE, continuedThe DEFWIN commandSyntax:

DEFWIN window, x1, y1, x2, y2, xsize, ysize, bitm, bankm

Parameters:

window	The window number; VT100 window is 8.
x1,y1	The upper left-hand coordinates.
x2,y2	Ignored. (Calculated automatically as noted below.)
xsize	The x scaling of text. Range is 1 (6 pixels) to 2.
ysize	The x scaling of text. Range is 1 (10 pixels) to 4.
bitm,bankm	Specify the write mask, as detailed in the next section.

How Emulator Calculates Lower Right Coordinates

The emulator calculates the lower right coordinates using the following formulas:

$$x2 = x1 + (xsize * 6 * 80) - 1$$

$$y2 = y1 - (ysize * 10 * 24) + 1$$

Default Value for Text Size

If you specify a value of zero for either of the text scaling parameters, then the previously specified values, or the default values are used.

The default powerup values for the scaling factors are xsize = 2 and ysize = 4. However, if you have not powered up into the VT100 window, the default values are xsize = 2 and ysize = 2.

CONTROLLING IMAGE MEMORYIntroduction

The Model One/80 has no overlay planes; thus there are no dedicated bit planes for the VT100 emulator. Because of this, you must control image memory yourself so that the VT100 emulator doesn't destroy it.

You control image memory by setting the write mask for the VT100 window.

Default Write Mask

The default for the VT100 emulator is that the lower 8 bits of image memory are available.

The write mask is set with the last two parameters of the DEFWIN command, bitm and bankm.

```
DEFWIN window, x1, y1, x2, y2, xsize, ysize, bitm, bankm
```

On powerup, bitm = 255 and bankm = 7.

Example of Changing the Write Mask

Suppose that you want the VT100 to draw only in bit plane 7. Then you would execute the following two commands:

```
LUTRMP 7 128 255 255 255 ; Guarantee that all VT100 text is
                           white, no matter what the background
                           color is.
```

```
DEFWIN 8 -480 479 479 -480 2 4 128 8
                                           ; Write enable only bit plane 7.
```

Reference

Refer to the WRMASK command for details on setting the write mask.

COLOR AND VIDEO ATTRIBUTESIntroduction

This section explains how color and video attributes are implemented for the VT100 emulator.

Colors on Powerup

The following chart shows which registers specify color for the VT100 emulator window.

VREG #	ATTRIBUTE	DEFAULT VALUE ON POWER UP	COLOR
40	text foreground color	255	white
41	background color	0	black
42	blink color	12	green
43	BOLD color	60	yellow

Changing the Colors

You can change the default colors by using the VLOAD command to load the value registers.

The cursor is drawn in XOR mode, so the actual color of the cursor depends upon the colors in the background.

Color changes do not become apparent until you next scroll or write text.

COLOR AND VIDEO ATTRIBUTES, continuedVideo Attributes

The following chart describes how video attributes are implemented for the VT100 emulator.

ATTRIBUTE	IMPLEMENTATION
UNDERLINING	Underline text.
REVERSE VIDEO	Swap foreground and background colors.
BOLD	Substitute VREG 43 for the foreground window color.
BLINK	Substitute VREG 42 for the foreground window color.
BOLD and BLINK together	Just implement BLINK.

Enabling Blinking

To enable blinking, you must put the value in VREG 42 into the blink table. Refer to the BLINKE command for details.

Caution: When you enable blinking, if there are any pixels outside of the text window which have the same value as the value in the blink table, then these pixels will also blink.

RESERVING MEMORY FOR THE VT100 EMULATOR

Introduction

This section describes the memory requirements for the VT100 emulator, and explains how to allocate additional memory if you want to use the alpha emulator along with the VT100 emulator.

Note: You can skip this section if you do not plan to use alpha windows along with the VT100 window.

Memory Requirements for VT100

On powerup, the VT100 emulator reserves 4K of alpha window area memory space. Both the VT100 emulator and the alpha emulator use this same area of memory. The default memory allocation for the alpha window area is 4K. Thus, if you want to use the alpha windows as well as the VT100 emulator, you must allocate additional memory for the alpha window area.

Reallocating Memory

You can use the MAP and CONFIG commands to reallocate memory for the alpha window area memory space.

The MAP command displays the current allocation for RAM space. To calculate the memory currently allocated for the alpha window area,

- execute the MAP command
- locate the display line labelled Alpha window area (line 4), and
- subtract the first value (displayed in hex) from the second value.

If you do not have enough memory reserved for the alpha window area, use the CONFIG command to reallocate RAM space.

For more information about the MAP and CONFIG commands, see the Model One/80 Command Reference.

Freeing Memory Space

If you are not going to use the VT100 emulator, you can enter the following sequence of commands to delete the VT100 window and free buffer space.

```
DEFWIN 0      ; Define alpha window 0 (using defaults).
SELWIN 0      ; Select alpha window 0 (using defaults).
DELWIN 8      ; Delete VT100 window 8.
```