

DESCRIPTION OF THE
TEKTRONIX GRAPHICS TERMINAL EMULATOR IN MS-DOS KERMIT V3.0
FOR IBM PCs

INTRODUCTION

The MS-DOS Kermit Tektronix terminal emulator has characteristic the real Tektronix 4010 and 4014 terminals plus extensive additions from DEC VT340 and Human Data Systems 2000/3000 series terminals. Please note the DEC VT340 can mix text and graphics on the same display but the hard design of IBM PC display adapters forbids this. Instead a best effort is done to simulate text and graphics together. Real Tektronix 401x terminals use storage tube technology which forbids erasure and colors; the MS-DOS Kermit emulator bends these rules in favor of raster display technology like the DEC VT340. The real Tektronix terminals display text by overwriting MS-DOS Kermit does this via command SET TERMINAL GRAPHICS CHARACTER-WRITE TRANSPARENT. Raster display devices fill each character cell with new data use option OPAQUE in the above command to achieve the same effect. A real terminal writes two screens of text by overwriting the second screen on right half of the first; MS-DOS Kermit sounds a beep and waits for a key and then starts on a fresh screen. VT340 Sixel graphics is supported, but ReGIS graphics is not.

The kind of display adapter is normally sensed automatically. But user can specify a type with command SET TERMINAL GRAPHICS; if the type present a fall-back to CGA or ordinary monochrome text is performed.

Screen coloring can be controlled from the Kermit prompt by the command SET TERMINAL GRAPHICS COLOR value, value, value. The "value" items are

```
0  white on black
1  for high intensity foreground
Foreground color (30-37) = 30 + sum of colors
Background color (40-47) = 40 + sum of colors
where colors are 1 = red, 2 = green, 4 = blue
```

Example: 1, 37, 44 bright(1) white(37) chars on a blue(44) background. These numbers can be separated by either commas or spaces. The default screen coloring is dim white characters on a black background, so that Print Screen does not make the paper all black (it makes an ink dot for each illuminated screen dot, without regard to foreground and background).

Command SET TERMINAL GRAPHICS CURSOR {ON, OFF} controls the occurrence of a box-like cursor character while in graphics terminal emulation mode default is ON. This cursor represents where the next text character would be drawn.

A mouse may be used to control the Tektronix GIN mode crosshair. It works in parallel with the keyboard arrow keys. A Microsoft compatible mouse driver is required, and a mouse menu program may be used when GIN mode is inactive. Within GIN mode any mouse menu is suspended. A typical DOS command to activate a mouse driver is MOUSE ON, or load MOUSE.SYS in file CONFIG.SYS. Mouse buttons are converted to be equivalent to pressing the Enter key.

Graphics screens are saved in display memory if the adapter has sufficient quantity. CGA screens cannot be saved, and VGA (640x480) can save the last 131 scan lines. Hercules adapters need graphics page 1 to an image; some clones do not seem to do this. The Wyse 700 display and special adapter do not allow screen saving. MS-DOS Kermit attempts to save the graphics screens whenever possible, even though this differs from DEC VT340 practice (DEC merges text and graphics modes, IBM-PCs cannot). Use = or keyboard verb \Kreset to do a full screen clear and terminal reset, echo ^L to just clear the screen.

The VT320 main text terminal emulator is designed to switch to Tektronix emulation whenever four commands are received:

1. ESC ^L Tektronix screen clear command
2. ESC [? 38 h VT340 command to enter Tek mode
3. DCS Pn q start of Sixel command string, Pn is a digit
4. ESC 1 same as ESC ^L

The third case results in the VT320 emulator feeding a ^L (screen clear) command to the Tek emulator, replaying the DCS lead-in characters, and replacing the Tek emulator temporarily in charge to read subsequent character. Tek sub-mode persists until either:

1. the user toggles terminal types manually (ALT white minus, or keyboard verb \Ktermtype)
2. ESC [? 38 I is read, VT340 command to exit Tek mode
3. ESC ^X is read
4. ESC 2 is read

Entering Tek mode by the third method (Sixel) forces character-writing to opaque, the other two methods do not.

ACTIONS TO RECEIVED CHARACTERS

Characters of 80h and above (those with their high bit on) are converted as follows:

80h..9fh converted to pair ESC <value - 40h>. These are C1 control codes. 0a0h..0ffh converted to <value - 80h> and become printable characters.

Control codes in C0 (no high bit) area

Name	ASCII chart	value hex	keyboard	operation
NUL	0/0	00h	^@	Ignored
ENQ	0/5	05h	^E	
BEL	0/7	07h	^G	Sound DEC style beep
BS	0/8	08h	^H	Backspace, move cursor left one character 8 dots, can be destructive
HT	0/9	09h	^I	Treated as a single space
LF	0/10	0ah	^J	Linefeed, move cursor down one line, 8 d
VT	0/11	0bh	^K	Vertical Tab, treated as a line feed
FF	0/12	0ch	^L	Formfeed, erase screen, Home cursor
CR	0/13	0dh	^M	Carriage return, move cursor to col 1
DC1	1/1	11h	^Q	XON flow control, resume communication
DC3	1/3	13h	^S	XOFF flow control, suspend communication
CAN	1/8	18h	^X	Return to text terminal, only if in sub-mode, else ignored if regular Tek terminal
SUB	1/10	1ah	^Z	Treated as a CAN
ESC	1/11	1bh	^[Escape, start escape seq, cancel any other
FS	1/12	1ch	^\	Enter point plotting mode
GS	1/13	1dh	^]	Enter line drawing mode
RS	1/14	1eh	^^	Enter incremental line drawing mode
US	1/15	1fh	^_	Enter Tek text mode (leave line/point drawing)

All others are ignored

Control codes in C1 (high bit set, 80h - 9fh) area

Name	ASCII chart	value hex	operation
DCS	9/0	90h	Expand to be ESC P Device Control String
ST	9/12	9ch	Expand to be ESC \ String Terminator
CSI	9/13	9dh	Expand to be ESC [Control Sequence Introducer

All others are expanded to be ESC <value-40h> and are generally ignored

Tektronix Coordinate system (the PC screen is scaled to fit a Tek screen)

The plot commands are characters which specify the absolute position to the beam. All moves except the one immediately after the GS character (Control -]) are with a visible trace.

For 4010-like devices - The positions are from 0 to 1023 for both X and although only 0 to 780 are visible for Y due to screen geometry. The screen is 10.23 by 7.80 inches, and coordinates are sent as 1 to 4 characters.

For 4014-like devices - The positions are from 0 to 4096, but each move is a multiple of 4 positions unless the high-resolution LSBXY are sent. makes it compatible with the 4010 in that a full sized plot fills the screen.

HI X, HI Y = High-order 5 bits of position

LOX, LOY = Middle-order 5 bits of position

LSBXY = Low-order 2 bits of X + low-order 2 bits of Y (4014 mode)

Hi Y	Lo Y	Hi X	LSBXY	Characters sent (Lo-X always sent)			
----	----	----	-----	-----			
Same	Same	Same	Same				Lo-X
Same	Same	Same	Di ff	LSB,	Lo-Y,		Lo-X 4014
Same	Same	Di ff	Same		Lo-Y,	Hi -X,	Lo-X
Same	Same	Di ff	Di ff	LSB,	Lo-Y,	Hi -X,	Lo-X 4014
Same	Di ff	Same	Same		Lo-Y,		Lo-X
Same	Di ff	Same	Di ff	LSB,	Lo-Y,		Lo-X 4014
Same	Di ff	Di ff	Same		Lo-Y,	Hi -X,	Lo-X
Same	Di ff	Di ff	Di ff	LSB,	Lo-Y,	Hi -X,	Lo-X 4014
Di ff	Same	Same	Same	Hi -Y,			Lo-X
Di ff	Same	Same	Di ff	Hi -Y,	LSB,	Lo-Y,	Lo-X 4014
Di ff	Same	Di ff	Same	Hi -Y,		Lo-Y,	Hi -X,
Di ff	Same	Di ff	Di ff	Hi -Y,	LSB,	Lo-Y,	Hi -X,
Di ff	Di ff	Same	Same	Hi -Y,		Lo-Y,	Lo-X
Di ff	Di ff	Same	Di ff	Hi -Y,	LSB,	Lo-Y,	Lo-X 4014
Di ff	Di ff	Di ff	Same	Hi -y,		Lo-Y,	Hi -X,
Di ff	Di ff	Di ff	Di ff	Hi -y,	LSB,	Lo-Y,	Hi -X,
Offset for byte:				20h	60h	60h	20h 40h

Note that LO-Y must be sent if HI-X has changed so that the TEKTRONIX k the HI-X byte (in the range of 20h-3fh) is HI-X and not HI-Y. LO-Y must also be sent if LSBXY has changed, so that the 4010 will ignore LSBXY and accept LO-Y. The LSBXY byte is 60h + MARGIN*10h + LSBY*4 + LSBX. (MARG

Control -] (GS) Tek coordinates. Enter Tek line plot mode.

The first move will be with beam off (a moveto command), subsequent coordinates will be reached with the beam on (a drawto command).

Note: this is also Kermit's Connect mode escape character so beware if typing GS by hand; SET ESCAPE to something else before the text Exit drawing upon reception of CR, LF, RS, US, FS, CAN.

Control -^ (RS) Tek coordinates. Enter Tek line incremental plot

RS space move with pen up (invisible)

RS P move with pen down (write dots)

RS <letter>

letter motion

A right (East)

B right and up (NE)

D up (North)

F left and up (NW)

letter motion

B left (West)

J left and down (SW)

H down (South)

I right and down (SE)

Exit drawing upon reception of CR, LF, RS, US, FS, CAN.

Example: RS <space> J J J means move three Tek positions left and down (three south west steps) with the pen up (move invisibly).

Control -\ (FS) Tek coordinates. Draw a dot at the coordinate. Point p mode. Like GS but does not join end points with lines.

Exit drawing upon reception of CR, LF, RS, US, FS, CAN.

Control -underline (_, US) Exit Tek line plot mode and return to text

Escape sequences (ESC intermediates Final)

Escape Seq	Mnemonic	Description of Action
ESC ^E		Request Tek status report
	Report is	
	20h Tek-X Tek-Y	0dh for non-text mode
	24h Tek-X Tek-Y	0dh for text mode
		Tek-X Tek-Y is Tek style cursor position
ESC ^L		Enter Tektronix sub-mode, clear Tek screen
ESC ^X		Turn on Bypass mode (no screen chars)
ESC ^Z		Turn on GIN crosshairs
ESC ?		Substitute for DEL, for 7-bit systems
ESC P		Device Control Sequence introducer (DCS) See below.
ESC Z		Report terminal type (as a VT320/VT340)
	Response is	
	ESC [? 63; 1; 2; 4; 8; 9; 15 c	a VT300 series, level 3, etc
ESC @ .. ESC M		(@, A, B, C, D, E, F, G, H, I, J, K, L, M) Select rectangular fill pattern 1..14. See ESC
ESC \		String Terminator (ST, of DCS items)
ESC ` .. ESC e, ESC x, .. ESC z		(accent grave, a, b, c, d, e, x, y, z)
		Select line drawing pattern
ESC letter	line type, bits,	least significant bit plotted f
accent	11111111	11111111
a	10101010	10101010
b	11110000	11110000
c	11111010	11111010
d	11111111	11001100
e	11111100	10010010
x	user defined (by ESC / Pn a)	
y	user defined (by ESC / Pn b)	
z	user defined (by ESC / Pn c)	

Control sequences (ESC / or ESC [or CSI)
(ESC / is an HDS 2000/3000 sequence, ESC [is ANSI form)

Control Seq	Mnemonic	Description of Action
ESC / P1; P2; ...; P8 C		Define user fill pattern. Use low 8-bit each Pn. P1 is top of fill, plotted lsb Pattern is 8x8 dots. Omitted Pn are 0's
ESC / P1; P2; ...; P8 D		Define second user fill pattern, as above
ESC / Pn a		Set user definable line drawing bit pattern
ESC / Pn b		Ditto, lsb drawn first, 16 bits overall
ESC / Pn c		Ditto
ESC / Pd d		Set pixel operation code
Pd	pixel operation	
0	draw 1's in foreground color, skip 0's	
1	draw 1's in background color, skip 0's	
2	XOR 1's with foreground color, skip 0's	
3	write absolute, 1's in foreground and 0's in background	
ESC / Pn h		Set, see table below

30-37 set foreground colors to Pn minus 30
 40-47 set background colors to Pn minus 40
 Colors are red=1, green=2, blue=4 and summations for oth
 Note: this stores new values in palette 0 (background) a
 palette 7 (foreground).

ESC [2; 2 \$ u Request VT340 color palette report
 Report is
 ESC P 2 \$ s <palette 0>/<palette 1>/...<palette 15> ESC \
 where <palette n> is the palette color in the RGB system -
 Pr; Pg; Pb for red, green, and blue percentages, re
 black is 0; 0; 0
 bold black is 20; 20; 20
 dim (regular) hue is 40, bold hue is 80.

NOTE: this report is a very long string and may not be acceptable to
 some communications channels (for example, Telnet). Further,
 host may request a report in the HLS system; MS-DOS Kermit alw
 reports in the RGB system.

Default VT340 color palettes are -

palette	b/w	color	palette	b/w	color
0 backgnd	black	black	8	dim grey	dim grey (bold
1	white	bold blue	9	grey	blue
2	white	bold red	10	grey	red
3	white	bold green	11	grey	green
4	white	bold magenta	12	grey	magenta
5	white	bold cyan	13	grey	cyan
6	white	bold yellow	14	grey	yellow/brown
7 foregnd	grey	grey	15	white	white (bold)

ESC [? 34 h Invoke macro Terminal S, if defined, exit
 connect mode.

ESC [? 34 l Invoke macro Terminal R, as above

ESC [? 38 l Exit Tek mode to text terminal emulator,
 if Tek mode were invoked from text emula
 by ESC [? 38 h or by a Sixel DCS.

ESC [? 256 n Request screen size report, MS-DOS Kermit
 Report is ESC [? 256; Ph; Pw; Pc n for graphics systems
 where Ph is screen height in dots
 Pw is screen width in dots
 Pc is number of colors (0, 1 or 16, for none, b/w, ega/v
 Report is ESC [? 24; 80; 0 n for pure text mono systems.

Device Control Strings (ESC P or DCS)

DCS P1; P2; P3 q string ST or in 7-bit form

ESC P P1; P2; P3 q string ESC \
 a Sixel Graphics command

P1 and P3 are ignored.

P2 = 0 or 2 means draw 0 bits in background, 1 means skip them.

string is a Sixel command string, containing mixtures of -

Sixel characters (3fh..7eh, lower 6 bits+3fh, displayed as six dots
 vertically, least significant bit at the top after subtracting 3
 "?" is all zeros, "@" is top line only, "~" is all 6 bits on.
 The initial Sixel char is placed at the top left of the current
 text cell, subsequent chars work to the right without wrapping.
 Writing below the screen bottom results in overwriting the botto
 strip.

! Pn sixel char Draw Sixel char Pn times (Pn is repeat c

" Pc; Pad; Ph; Pv Raster attributes (all ignored)
 \$ (dollar sign) Go to left margin
 - (minus) Go to left margin and 6 dots down
 Control characters Perform the function, stay in Sixel mode
 that LF increments by 8 dots (text cell)

Escape sequences are permitted within string and occur without disr

Pc; Pu; Px; Py; Pz Set palette color, as follows,
 Pc is color palette, 0..255 (0 is background, 7 is normal foreground only 0..15 are predefined.

Pu is color system, 1 = HLS, 2 = RGB

For Hue Lightness Saturation:

Px = Hue angle, 0-360 degrees. The colors are mapped around the color wheel in 60 degree segments as Hues:
 0-29 deg = blue, 30-89 = magenta (blue + red), 90-149 =
 150-209 = yellow (red + green), 210-269 = green,
 270-329 = cyan (green + blue), 330-359 = blue.

Py = Lightness, 0-100%, Pz = Saturation, 0-100%

Lightness	Sat = 51-100	Sat = 11-50	Sat = 0
86-100	bold white	bold white	bold wh
71-85	bold hue	bold white	bold wh
57-70	bold hue	grey (dim white)	grey
43-56	bold hue	dim hue	black
29-42	dim hue	grey	grey
14-28	dim hue	black	black
0-13	black	black	black

Note that Py = Pz = 50 gives the widest spectrum.

For RGB: Px = red, 0-100%, Py = green, 0-100%, Pz = blue, 0-100%
 If any color exceeds 50% then the bold bit is turned on the ensemble (IBM ega display adapter constraint for iRG)

Palette registers can be selected by the substring

Pc followed by a non-numeric character other than ";"
 and Pc is the palette register, 0..255.

Example of dynamic palette register selection -

ESC P ; 1 q AAAA#2BBBBB#3! 6C ESC \ (omit the spaces)

Displays Sixel char A four times in the default foreground color (palette 7), then B five times in colors of palette then C six times in palette 3 colors. The ";1" says skip coloring dots with 0 bits.

Exception from DEC: color of black always writes all dot black (an erasure or clearing).

Suggestion: when possible ask for a palette report and store the report change palettes as desired for a plot, and then restore the palettes.

Sixel character plotting begins at the upper left of the current text cell. Thus, either Tek or ANSI cursor steering commands can be employed to locate the starting position. ESC [..m coloring escape sequence occur withing a Sixel string and it acts on the current foreground/background colors and stores them in palettes 7 and 0. Sixel dots are stored by ORing the palette value with the palette value already existing in the dot, with the exception that all black writes black absolutely. At completion of a Sixel DCS the screen colors are reset to palette 7 and for foreground and background, respectively.

TIFF SCREEN DUMPS

These follow Aldus/Microsoft TIFF version 5.0 specifications. The output filename is always TEKPLTnn.TIF, where nn starts at 01 and increments by one for each new screen dump. The files are in uncompressed format so thus EGA/VGA screens yield 100KB+ files. Monochrome graphics are written as one bit per pixel TIFF type B format, with the screen divided into 25 fewer strips. EGA and VGA screens use TIFF type P palette format and a supplementary color palette to RGB table. They too are divided into 25 fewer strips and use four bits per pixel. The color palette contents are to match normal IBM PC display adapter conventions.

When dumping a screen via Control End or keyboard verb \Kdump the graphics screen needs to be visible; otherwise, an ordinary text screen is written to the Kermit.scn file.

TIFF v5 specifications are available directly from Aldus or Microsoft.

EXAMPLES

Tek 4010 line drawing:

Suppose <xy> is point y = 300, x = 500 in Tektronix coordinates. Split 10-bit coordinate into 5-bit groups, add the kind code to each. Send part last.

	HI - Y	LO - Y		HI - X	LO - X
Y=300d=012Ch=	01001	01100	X=500d=01F4h=	01111	10100
+kind code	+100000	+1100000	+kind code	+100000	+1000000
Binary	101001	01101100		101111	1000100
ASCII)	I		/	D

So <xy> = (500, 300) is sent or received in a GS command as ")I/D". An in C (program fragments):

```

-----
/* File tek.c. Creates binary output file tek.tst. Replay that file. */
/* writes a Tek test file 'tek.tst', Lattice C */
#include <stdio.h>
#include <math.h>

#define ESC 0x1b
#define FF 0x0c
#define CAN 0x18
#define FS 0x1c
#define GS 0x1d
#define US 0x1f
#define ESCZ 0x1a
#define RED 1
#define GREEN 2
#define BLUE 4
#define color(c) fputc(ESC, fp); fputc(' ', fp); fputc('1', fp); \
    fputc(';', fp); fputc('3', fp); fputc('0'+c, fp); fputc('m', fp);
FILE *fp;

main()
{
    int i, x, y, xc = 750, yc = 500;
    double radius = 125.0;

    if ((fp = fopen("tek.tst", "wb")) == NULL) /* write binary mode
        exit(1);

    fputc(ESC, fp); fputc(FF, fp); /* clear screen, enter graphics

```

```

for (i = 0; i < 40; i++) fputc('\0', fp);          /* padding */
                                                    /* for mode swit
fputc(GS, fp); coord(210, 500);                  /* moveto */
color(RED);
fputc(US, fp); fputs("shallow fan", fp);        /* text mode */
color(GREEN+RED);
fputc(GS, fp); coord(50, 500); coord(200, 400); /* drawto's */
coord(50, 500); coord(200, 450);
coord(50, 500); coord(200, 500);
coord(50, 500); coord(200, 550);
coord(50, 500); coord(200, 600);

fputc(GS, fp); coord(460, 500);
color(BLUE);
fputc(US, fp); fputs("steep fan", fp);
fputc(GS, fp); coord(400, 200); coord(400, 800);
coord(400, 500); coord(450, 200);
coord(400, 500); coord(450, 300);
coord(400, 500); coord(450, 400);
coord(400, 500); coord(450, 500);
coord(400, 500); coord(450, 600);
coord(400, 500); coord(450, 700);
coord(400, 500); coord(450, 800);
fputc(US, fp); fputc(' ', fp);

color(GREEN);
fputc(GS, fp);                                  /* simple circle */
for (i = 0; i <= 360; i++)
{
    x = radius * cos(PI * i / 180.0);
    y = radius * sin(PI * i / 180.0);
    coord(x+xc, y+yc);
}

color(GREEN+BLUE);
fputc(GS, fp); coord(75, 65);                  /* moveto */
fputc(US, fp); fputs("This is a house\n", fp); /* text mode */
fputc(GS, fp);                                  /* draw lines for house
coord(50, 50); coord(300, 50);
coord(300, 200); coord(50, 200);
coord(175, 250); coord(300, 200);
fputc(GS, fp); coord(50, 50); coord(50, 200);
color(RED+BLUE);

fputc(GS, fp); coord(350, 50);                /* do some point plotting */
fputc(FS, fp);                                  /* draw a dotted rectangle */
for (i = 350; i <= 600; i += 4) coord(i, 50);
for (i = 50; i <= 200; i += 4) coord(600, i);
for (i = 600; i >= 350; i -= 4) coord(i, 200);
for (i = 200; i >= 50; i -= 4) coord(350, i);
color(RED+GREEN+BLUE);
fputc(GS, fp); coord(50, 10);                 /* move to */
fputc(US, fp); fputs(" the end.", fp);        /* text mode */

fclose(fp);
exit(0);
}

coord(x, y)                                    /* package coordinates Tek style
int x, y;
{
    fputc((y / 32) + 32, fp);                 /* high y */
    fputc((y % 32) + 96, fp);                 /* low y */
    fputc((x / 32) + 32, fp);                 /* high x */
    fputc((x % 32) + 64, fp);                 /* low x */
}

```

}