NASA TM-100331

NASA Technical Memorandum

NASA TM- 100331

GRAPHICS SOFTWARE TOOL FOR VT TERMINALS (VTGRAPH)

By Caroline Wang

Software and Data Management division
Information and Electronic Systems Laboratory
Science and Engineering Directorate

June 1988

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA TM-100331		

4. TITLE AND SUBTITLE	5. REPORT DATE
Graphics Software Tool for VT Terminals (VTGRAPH)	June 1988
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT #
Caroline Wang	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	11. CONTRACT OR GRANT NO.
	13. TYPE OF REPORT & PERIOD COVERED
12. SPONSORING AGENCY NAME AND ADDRESS	
National Aeronautics and Space Administration Washington, DC 20546	Technical Memorandum
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

Prepared by Software and Date Management Division, Information and Electronic Systems Laboratory, Science and Engineering Directorate

16. ABSTRACT

VTGRAPH is a graphics software tool for using DEC/VT or VT compatible terminals. It allows the user to deal with computer environments which use VT terminals for window management and graphics systems.

VTGRAPH was developed using the Re'Gis Graphics set and it was written in FORTRAN language. It provides window management and PLOT10-like package plus color or shade capability.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Graphics Software Tool	Unclassified-Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	44	NTIS

MSFC - Form 3292 (Rev. December 1972)

For sale by National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

TECHNICAL MEMORANDUM

GRAPHICS SOFTWARE TOOL FOR VT TERMINALS (VTGRAPH)

I. INTRODUCTION

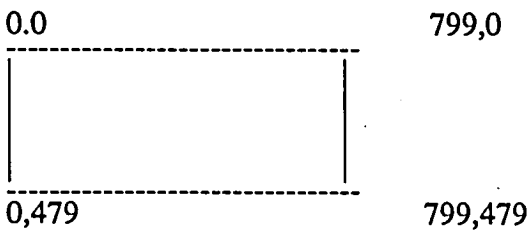
VTGRAPH is a software tool for using DEC/VT or VT compatible terminals. It allows the user to deal with many computer environments which use VT terminals for window management and graphics systems. VTGRAPH was designed by the author as a transportable software package to run on any computer using VT or VT compatible terminals.

VTGRAPH was developed using the ReGis Graphics set and it was written in FORTRAN language. ReGis provides a full range of graphics capabilities.

VTGRAPH provides a window management system for VT100, VT200, and VT300 terminals, and it also provides a PLOT10-like package plus color or shade capability for VT240, VT241, and the new VT300 terminals.

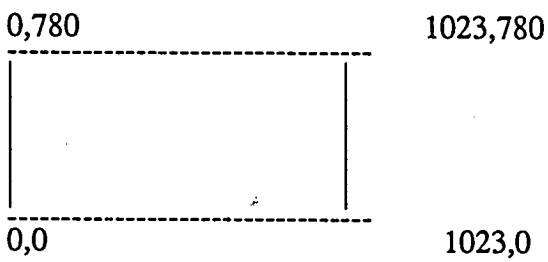
There are two types of screen coordinates which include:

1) Window management system



This type is for window management and chart drawing.

2) PLOT10 compatible screen coordinates for drawing.



This type is for PLOT10 compatible programs.

II. PURPOSE

VT terminals are widely used by Government and industry. They are the terminals used most often at Marshall Space Flight Center. VTGRAPH was designed to provide ease of use in a wide range of window management and graphics capabilities.

VTGRAPH is a FORTRAN or C callable library, and it was designed for PLOT10 compatibility plus color window management. It is transportable for many different computers which use VT terminals.

The differences between VT100's, VT200's, or VT300's are that the VT100 and VT102 only provide window capability, VT125 provides black and white graphics, VT240 provides shaded graphics, and VT241 provides for color graphics and color drawing. The new VT300 terminals also support the full range of ReGis Graphics set and is a VT240 compatible system.

With this graphics package, the user can easily design more friendly user interface programs and design PLOT10 programs on VT terminals with different computer systems.

III. VTGRAPH OVERVIEW

A. VTGRAPH Basic Capability

The basic VTGRAPH capabilities are:

1) Window management.

2) PLOT10 compatible drawing.

3) Generic program routine for two-dimensional plotting.

4) Generic program routine for three-dimensional plotting.

5) Color graphics or shaded graphics.

B. VTGRAPH Library

The VTGRAPH Library is as follows:

INITIALIZATION and EXIT:

INIT
VTOUT
INITT
RESET
FINITT

Color Graphics Utility (VT200s, VT300s)

BACK
CIRC
CLR
COLR
CPOS
CURS
CURV
DENT
DLAY
DOT
DRAW
MOVE
PUTC
RECT
TEXT

PLOT10 Compatible Window Coordinates:

Drawing Utility:

MOVABS
DRWABS
PNTABS
DSHABS
MOVEA
DRAWA
POINTA
DSHA

Virtual Window:

VWINDO
DWINDO

Screen Window:

SWINDO
TWINDO

VT Window Utility Routines:

ANMODE
GMOD
AOFF
ANCHO
ANSTR
NEWPAGE
CURPOS
CHRSIZ
ERASE
BELL
BLINK
CLRLIN

Generic Plotting Routines:

PLOT2D_I
PLOT2D_R
PLOT3D_I
PLOT3D_R

Initialization and EXIT:

Subroutine INIT

Initialization of the terminal.
The screen is erased and the cursor moves to the HOME position.

Subroutine VTIN

This routine accomplishes the same function as INIT, but it does not clear the screen. It can be used to re-enter the VTGRAPH mode.

Subroutine VTOUT

When terminating a program.
Set VT into regular mode.

Subroutine INITT (BAUD); for PLOT10 compatible software

This routine works the same as INIT except it is a PLOT10 compatible routine and is for black and white terminals only.
IBAUD is a dummy argument here, but it is the transmission rate in characters per second for PLOT10 package.

Subroutine RESET; for PLOT10 compatible software

This routine accomplishes the same function as INITT, but it does not call for a new page.

Subroutine FINITT (IX,IY); for PLOT10 compatible software

Used when terminating a program.
It returns the terminal to alphanumeric mode and moves the cursor to a point that will not interfere with any previous output.
IX: The screen x-coordinate of position to which the beam is moved before program termination.

IY: The screen y-coordinate of the beam termination position.

Color Graphics Utility

These utilities are for VT200s and VT300s terminals.

The window range for VT200s are from upper left (0,0) to lower right (799,479).

Subroutine BACK (color)

Color: Background color code

0: Dark
1: Blue
2: Pink
3: Tan

CALL BACK (color)

Subroutine CIRC (IX,IY,IR,IFILL)

This subroutine draws a circle using current color.

IX,IY: Screen position of the center of the circle.
IR: Radius of the circle.
IFILL: Fill flag (0 is unfilled, 1 is filled)

CALL CIRC(IX,IY,IR,IFILL)

Example:

```
Call color (2); red color
Call circ(200,200,50,1)
```

Subroutine CLR

This subroutine clears the VT241 screen

CALL CLR

Subroutine COLR (jcolor)

This subroutine sets the current drawing color.

Foreground drawing color code:

jcolor:
0: Dark
1: Blue
2: Red
3: Green

CALL COLOR (jcolor)

Subroutine CPOX(IX,IY)

IX,IY: Current drawing position

IX,IY are integer*2
Window range from upper left (0,0) to lower right (799,479)

CALL CPOS(IX,IY)
CPOS works the same as MOVE.

Subroutine CURS (jcode)

This subroutine performs a cursor operation.

jcode=0: turn cursor off
=1: turn cursor on

CALL CURS(jcode)

Subroutine CURV(XARRAY,YARRAY,INPTS,ICFLAG)

This subroutine draws either an open or closed curve connecting the specified points.

XARRAY: X array, integer*2
YARRAY: Y array, integer*2
INPTS: number of points, integer*2
ICFLAG: logical*1, closed/open flag (0 is open, 1 is closed)

CALL CURV(XARRAY,YARRAY,INPTS,ICFLAG).

Subroutine DENT(IX,IY,STRING,SLEN,IS,IA,ICA)

This subroutine displays a text string using the current font of the specified size and rotation.

IX,IY: [integer*2] starting position of the string
STRING: [logical*1] text string
SLEN: [integer*2] the length of the text string
IS: [integer*2] character size code (0:16)
IA: [integer*2] angle rotation of the string
ICA: [integer*2] angle rotation of the character

CALL DENT (IX,IY,STRING,SLEN,IS,IA,ICA)

Window range from upper left (0,0) to lower right (799,479).

Subroutine DLAY(SECS)

This subroutine delays the execution of the display buffer by the specified number of seconds.

SECS: [real*4] number of seconds

CALL DLAY(SECS)

Subroutine DOT(IX,IY)

This subroutine draws a single pixel dot at the specified screen position in the current color.

IX,IY: [integer*2] dot position

CALL DOT(IX,IY)

Subroutine DRAW(IX,IY)

This subroutine draws a single pixel width line from the current drawing position to the specified screen position in the current color.

IX,IY: [integer*2] drawing cursor posiiton

CALL DRAW(IX,IY)

Subroutine DSHD(IX,IY,STRING,SLEN,W,H,IA)

This subroutine displays a text string using the current font of the specified size and rotation.

Subroutine MOVE(IX,IY)

This subroutine moves the current drawing position to the position specified.

IX,IY: [integer*2] the new current drawing position.

CALL MOVE(IX,IY)

Subroutine PUTC(IX,IY)

This subroutine positions the cursor at the specified location.

IX,IY: [integer*2] new screen position of the cursor.

CALL PUTC(IX,IY)

Subroutine RECT(IX,IY,IDX,IDY,IFILL)

This subroutine draws a filled or unfilled rectangle.

IX,IY: [integer*2] reference corner of the rectangle
IDX: [integer*2] width of the rectangle in pixels
IDY: [integer*2] height of the rectangle in pixels
IFILL: [integer*2] fill flag (0 is unfilled, 1 is filled)

CALL RECT(IX,IY,IDX,IDY,IFILL)

Subroutine TEXT(STRING,SLEN)

This subroutine displays a text string using current font.

STRING: [logical*1] text string
SLEN: [integer*2] number of characters of the string.

CALL TEXT(STRING,SLEN)

PLOT10 Compatible Window Coordinates

Drawing Utility:

Subroutine MOVABS(IX,IY); for VT125 or VT200s; PLOT10 compatible routine.

Calling sequence: CALL MOVABS(IX,IY)

Moves the point to absolute position (X,Y).

Subroutine DRWABS(IX,IY); for VT125 and VT200s; PLOT10 compatible routine.

DRWABS generates a vector from the current beam position to the coordinates given and updates the appropriate variables in the terminal status area.

CALL DRWABS(IX,IY)

Example:

```
CALL MOVABS(100,100)
CALL DRWABS(200,200)
```

Subroutine PNTABS(IX,IY): for VT125 and VT200s

Moves to the coordinates given as arguments and displays a point there.

Subroutine MOVEA, DRAWA, and POINTA; are analogous to MOVABS, DRWABS, and PNTABS, but they are for virtual windows.

Calling sequence:

```
CALL MOVEA (X,Y)
CALL DRAWA (X,Y)
CALL POINTA (X,Y)
```

X: The horizontal virtual (real) coordinate to which a bright or dark vector is drawn or at which a point is displayed.

Y: The vertical virtual (real) coordinate to which a bright or dark vector is drawn or at which a point is displayed.

DSHA, DSHABS works the same as DRAWA and DRWABS except they draw dash lines.

Subroutine DSHABS(IX,IY,L); draws dash pattern.

IX: X position
IY: Y position
L: pattern number (1 to 4)

DSHABS works like DRWABS for absolute location.

Subroutine DSHA(IX,IY,L); draws dash pattern.

IX: X position
IY: Y position
L: pattern number (1 to 4)

DSHA works like DRAWA for relative location.

Example:

```
CALL MOVABS(100,100)
CALL DSHABS(200,100,4)
CALL MOVABS(100,200)
CALL DRWABS(200,200)
CALL MOVEA(X1,Y1)
CALL DSHA(X2,Y2,2)
CALL MOVEA(X3,Y3)
CALL DRAWA(X4,Y4)
```

Virtual Window:

Subroutine VWINDO (XMIN, XRANGE, YMIN, YRANGE)

DWINDO (XMIN, XMAX, YMIN, YMAX) for VT125, VT200s

PLOT10 compatible routine.

The Terminal Control System uses one of two subroutines to define the virtual window, VWINDO, DWINDO.

XMIN: The minimum horizontal user coordinate
YMAX: The maximum horizontal user coordinate
YMIN: The minimum vertical user coordinate
YMAX: The maximum vertical user coordinate
XRANGE: The horizontal extent of the rectangle
YRANGE: The vertical extent of the rectangle

Screen Window:

The terminal control system uses one of two subroutines to define the screen window. SWINDO, TWINDO

SWINDO (MINX,LENX, MINY, LENY)
TWINDO (MINX,MAXX,MINY,MAXY)

MINX: The minimum horizontal screen coordinate
MAXX: The maximum horizontal screen coordinate
MINY: The minimum vertical screen coordinate
MAXY: The maximum vertical screen coordinate
LENX: The horizontal extent of the rectangle
LENY: The vertical extent of the rectangle

Example:

```
CALL DWINDO(-20.5, 100.5, 32.3, 200.3)
```

```
CALL TWINDO(100,1000,50,700)
CALL VWINDO(-20.5, 80.0, 32.3, 68.0)
CALL SWINDO(100, 900, 50, 650)
```

VT Window Utility Routines

Subroutine ANMODE

Allows terminal control system to monitor alphanumeric (A/N) mode. This allows user to output A/N data.

```
CALL ANMODE
```

Subroutine GMOD

Set special graphics characters mode.

If the special graphics set is selected, the graphics for ASCII codes OCT137 to OCT176 will be replaced by special graphics characters.

Subroutine AOFF

Attributes off.

This is used after the reverse image was set and need to return to the regular image.

Subroutine ANCHO (ICHAR)

A/N character output

ICHAR: An integer which represents a 7-bit ASCII character, not a control character.

Example:

```
CHARACTER TO BE DISPLAYED
ICHAR='A'

CALL ANCHO(ICHAR)
```

Subroutine ANSTR (NCHAR, NADE); PLOT10 compatible

This routine accomplished the same as function TEXT, but it is a PLOT10 compatible routine.

A/N String output

NCHAR: The number of characters to be output

NADE: An array containing the ASCII decimal integer equivalents for the characters to be output.

Example:

```
CALL ANSTR(12,'THIS IS TEST')
```

ANSTR is a PLOT10 compatible routine, it works the same as TEXT routine in VTGRAPH.

Subroutine NEWPAGE; vor VT100s and 200s; PLOT10 compatible.

Erases the terminal screen and returns the alphanumeric cursor to the HOME position.

Subroutine CURPOS(IX,IY); for VT100s and VT200s

IX: The line number
IY: The column number

Subroutine RIMAGE; for VT100s and VT200s

Reverse image

CALL RIMAGE

Subroutine CHRSIZ(ISIZE); for VT100s and VT200s; PLOT10 compatible

Changing the character size

ISIZE: an integer number for character size.

ISIZE=1 to 9 for VT terminal
ISIZE=1 to 4 for PLOT10 4010 emulater.

Subroutine ERASE; for VT100s and VT200s; PLOT10 compatible

The terminal screen may be erased without changing the mode or beam position.

CALL ERASE

Subroutine BELL; for VT100s and VT200s; PLOT10 compatible

Ring the Bell.

CALL BELL

Subroutine BLINK; for VT100s and VT200s

Blink the character.

CALL BLINK.

Subroutine CLRLIN(Code); for VT100s and VT200s; clear the line.

Code=0; from cursor to end of line

Code=1; from beginning of line to cursor

Code=2; entire line containing cursor

CALL CLRLIN(1)

Generic Plotting Routines

These routines were developed for generic two-dimensional and three-dimensional graphics.

The user needs to generate two-dimensional and three-dimensional data points and store them into the buffer array (x,y) or (x,y,z) for future drawing.

Two-Dimensional Plotting:

The maximum number of points is 1000 for this package.

Subroutine PLOT2D_I (NAME,IXMIN,IXMAX,IYMIN,IYMAX,IPTN,ICLR)

This subroutine automatically scales and plots the 2D data file, and also allows the user to select minimum and maximum X and Y range for different view of the plots.

Format of the file: 2I10 or two integer numbers with comma in between.

NAME: Data file name
IXMIN: minimum of x range
IXMAX: maximum of x range
IYMIN: minimum of y range
IYMAX: maximum of y range
IPTN: drawing dash pattern
  code 0: solid line
  code 1 through 4: dash lines; integer number
ICLR: set color
  code 0: dark
  code 1: blue
  code 2: red
  code 3: green

Subroutine PLOT2D_R (NAM,XMIN,XMAX,YMIN,YMAX,IPTN,ICLR)

This subroutine works the same as PLOT2D_I except the x and y points are real numbers.

Format of the file: 2F15.5 or two real numbers with comma in between.

NAM: Data file name
XMIN: x minimum; real number
XMAX: x maximum; real number

YMIN: y minimum; real number
YMAX: y maximum; real number
IPTN: pattern code; integer number
ICLR: color code; integer number

Subroutine PLOT3D_I (NAM, IXMIN, IXMAX, IYMIN, IYMAX, IZMIN, IZMAX, IPTN, ICLR)

This subroutine plots x,y,z 3D data file, it converts x,y,z points into projectory points to plot on the normal screen. It also allows the user to select the projectory x,y range for the plot.

Format of the file: A1 1X, 3I10 or three integer numbers with comma in between.

NAM: 3D data file name
IXMIN: x minimum; integer
IXMAX: x maximum; integer
IYMIN: y minimum; integer
IYMAX: y maximum; integer
IZMIN: z minimum; integer
IZMAX: z maximum; integer
IPTN: pattern code; same as PLOT2D_I
ICLR: color code; same as PLOT2D_I

Example data file:
D, 100, 200, 100
D, 150, 250, 100
M, 200, 100, 200
D for drawing and M for moving.

Subroutine PLOT3D_R (NAM, XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX, IPTN, ICLR)

This subroutine plots x,y,z 3D data file. It converts x,y,z points into projectory points to plot on the normal screen. It also allows the user to select the projectory x,y range for the plot.

Format of the file: A1 1X, 3F15.5 or three real numbers with comma in between.

NAM: 3D data file name
XMIN: x minimum; real
XMAX: x maximum; real
IYMIN: y minimum; real
IYMAX: y maximum; real
IZMIN: z minimum; real
IZMAX: x maximum; real
IPTN: pattern code; same as PLOT2D_I
ICLR: color code; same as PLOT2D_I

Example data file:
D, 100.5, 200.7, 100.8
D, 150.6, 250.8, 100.9
M, 200.45, 100.91, 200.5
D for drawing and M for moving.

C. Software and Hardware Requirement

The software and hardware requirements are as follows:

1. Software requirement: ReGis graphics set which all the VT terminals support; FORTRAN compiler.

2. Hardware requirement: VT or VT compatible terminals, time sharing or micro computers.

IV. VTGRAPH USER'S GUIDE

A. VTGRAPH Installation Guide

1. Make sure all the VTGRAPH library routines are there.

VTAMOD.FOR
VTANCHO.FOR
VTANMODE.FOR
VTSTR.FOR
VTAOFF.FOR
VTBACK.FOR
VTBELL.FOR
VTBLINK.FOR
VTCHRSIZ.FOR
VTCIRC.FOR
VTCLIP.FOR
VTCLR.FOR
VTCLRLIN.FOR
VTCOLR.FOR
VTCPOS.FOR
VTCURPOS.FOR
VTCURS.FOR
VTCURV.FOR
VTDENT.FOR
VTDLAY.FOR
VTDOT.FOR
VTDRAW.FOR
VTDRAWA.FOR
VTDRWABS.FOR
VTDSHA.FOR
VTDSHABS.FOR
VTDSHD.FOR
VTDWINDO.FOR
VTERASE.FOR
VTFINITT.FOR
VTGMOD.FOR
VTINIT.FOR

VTINITT.FOR
VTMOVABS.FOR
VTMOVE.FOR
VTMOVA.FOR
VTNEWPAG.FOR
VTPLOT2D_I.FOR
VTPLOT2D_R.FOR
VTPLOT3D_I.FOR
VTPLOT3D_R.FOR
VTPOINTA.FOR
VTPUTC.FOR
VTRECT.FOR
VTRESET.FOR
VTRIMAGE.FOR
VTSWINDO.FOR
VTTEXT.FOR
VTTWINDO.FOR
VTIN.FOR
VTOUT.FOR
VTVWINDO.FOR
VTWINDOW.FOR

2. Create a library file

   Example for VAX system: Library/create VTGRAPH

3. Load all the VTGRAPH package BIN files into VTGRAPH library.

   Example for VAX system: Library/insert .....

4. Set up user's program.

   Example for VAX system: Link user_program,VTGRAPH.Olb/lib

B. User's Program Set Up Procedure

1. Create user's program.
2. Compile user's program.
3. Link user's program object file with VTGRAPH library

   Example for VAX system

```
For MAIN
For SUB1

Edit MAIN.COM
MAIN,SUB1,VTGRAPH.OLB/LIB
EXIT

Link @ MAIN

Run MAIN
```

C. VTGRAPH User's Program Examples

VTGRAPH user's program examples are given on the following pages of computer printout.

```
c
c       ****************************************************************
c       *                                                              *
c       *         THIS IS A DEMONSTRATION PROGRAM WITH USING VT_GRAPH   *
C       *                                                              *
C       ****************************************************************
c
c       Caroline Wang
c
c


1000    FORMAT($,1H+,'VTGRAPH DEMO PROGRAM')
1001    FORMAT($,1H+,' 1. VT241 COLOR DEMONSTRATION')
1002    FORMAT($,1H+,' 2. VT240 SHAD DEMONSTRATION')
1003    FORMAT($,1H+,' 3. PLOT10 COMPATIBLE DRAWING')
1004    FORMAT($,1H+,' 4. 3D PLOT')
1005    FORMAT($,1H+,' 5. DASH LINE PATTERN')
1009    FORMAT($,1H+,' 6. TERMINATE')
1010    FORMAT($,1H+,'  INPUT OPTION:')
2000    FORMAT(I5)
c
c
c
10      continue
        call reset
        IL=2
        IC=20
        CALL RIMAGE
        CALL CURPOS(IL,IC)
        WRITE(5,1000)
        CALL AOFF
        IC=5
        IL=IL+3
        CALL CURPOS(IL,IC)
        WRITE(5,1001)
        IL=IL+1
        CALL CURPOS(IL,IC)
        WRITE(5,1002)
        IL=IL+1
        CALL CURPOS(IL,IC)
        WRITE(5,1003)
        IL=IL+1
        CALL CURPOS(IL,IC)
        WRITE(5,1004)
        IL=IL+1
        CALL CURPOS(IL,IC)
        WRITE(5,1005)
        IL=IL+1
        CALL CURPOS(IL,IC)
        WRITE(5,1009)
        IL=IL+3
        CALL CURPOS(IL,IC)
        WRITE(5,1010)
        READ(5,2000) IOPT
        GO TO(1,2,3,4,5,6) IOPT
```

```
1       CONTINUE
        CALL VT_COLOR
        call curpos(23,5)
        PAUSE 'key in CONTINUE to continue'
        GO TO 1Ø
2       CONTINUE
        CALL VT_COLOR
        call curpos(23,5)
        pause 'key in CONTINUE to continue'
        GO TO 1Ø
3       CONTINUE
        CALL VT_PLOT1Ø
        call curpos(23,5)
        GO TO 1Ø
4       CONTINUE
        CALL VT_PLT3D
        PAUSE 'KEY IN CONTINUE TO CONTINUE'
        CALL CURPOS(23,5)
        GO TO 1Ø
5       CONTINUE
        CALL VT_PATTERN
        call curpos(23,5)
        pause 'key in CONTINUE to continue'
        GO TO 1Ø
6       CONTINUE
        STOP
        END
```

```
C       THIS ROUTINE SHOWS THE VT_GRAPH COLOR CAPABILITY
C       ON VT240, VT241 TERMINALS OR VT300'S TERMINALS
        SUBROUTINE VT_COLOR
        implicit integer(a-z)
        integer*2 x(10),y(10),ix,iy,jcode
        real x1,y1
        data x/150,468,406,195,289,453,578,531,312,276/
        data y/402,379,329,359,179,299,279,329,289,369/
        call init
        type *,' s(e)'
        call vtout
        do 11 i=1,3
11      call bell
        call init
        call clr
        jcode=1
        call colr(jcode)
        call dshd(150,100,'MSFC',4,10,13,0,0)
        call bell
        jcode=2
        call colr(jcode)
        call dshd(30,300,'V T G R A P H',13,6,6,0,0)
        call colr(3)
        call dshd(100,400,'by Caroline Wang/EB44',21,3,3,0,0)
        call bell
        type *,' s(t120)'
        type *,' s(e)'
        jcode=3
        call colr(jcode)
        call rect(0,0,799,479,1)

        jcode=0
        call colr(jcode)
        call rect(117,120,565,280,1)
        call rect(7,5,785,466,0)
        call rect(9,6,781,466,0)
        type *,' t(d0,s2)'
        call move(212,59)
        call text('FORTRAN DRAW PACKAGE',20)
        call move(150,89)
        call text('PROVIDES THESE CAPABILITIES',27)
        call move(234,429)
        call text('TURN CURSOR ON/OFF',18)
        call putc(x(1),y(1))
        call curs(1)
        jcode=3
        call colr(jcode)
                type *,' s(c1)'
                do 10 i=2,10
                call bell
                dx=x(i)-x(i-1)
                dy=y(i)-y(i-1)
                x1=x(i-1)
```

```
                yl=y(i-1)
                do 15 j=1,15
                        ix=xl
                        iy=yl
                        call putc(ix,iy)
                        xl=xl+dx/15.
                        yl=yl+dy/15.
15              continue
                call curs(1)
                call putc(x(i),y(i))
                call rect(x(i),y(i),2,2,1)
                call curs(1)
10      continue
        jcode=3
        call colr(jcode)
        call rect(117,429,563,40,1)
        jcode=0
        call colr(jcode)
        call move(296,429)
        type *,' t(d0,s2)w(c)'
        call text('DRAW CURVES',11)
        type *,' w(v)'
        jcode=0
        call colr(jcode)
        call move(117,414)
        call rect(117,120,565,280,1)
        jcode=1
        call colr(jcode)
        call move(x(1),y(1))
        call curv(x,y,10,0,2)
        call demo_window('DRAW TARGET',11)
        call demo_target
        call demo_window('TEXT SAMPLE',10)
        CALL demo_text
        call vtout
        end
```

```
c       This routine plots 2D data file.
c       It uses plot10 compatible routines.
c       x :integer*2
c       y :integer*4
c       data points need to be less than 1000
c
c       developed by Caroline Wang
c
c       EB44/MSFC NASA
c       (205)-544-3887
        SUBROUTINE VT_PLOT10
        LOGICAL*1 NAM(30)
        dimension x(1000),y(1000)
        integer xx(1000)
        integer*4 yy(1000)
        COMMON /PT/X,Y,IR
        COMMON /VTGRPH/MINX,MAXX,MINY,MAXY,XMIN,XMAX,YMIN,YMAX
        COMMON /COLOR/ICLR
c
c
c
c
1000    FORMAT(2I10)
1002    FORMAT(I3)
4000    FORMAT(1X,'INPUT FILE NAME=(The demo data file: histf.dat)')
4001    FORMAT(Q,30A1)
1001    FORMAT(1X,'OPTION:',/,
     *          1X,'1. GO',/,
     *          1X,'2. STOP')
1003    FORMAT(1X,'OPTION:',/,
     *          1X, '1. BLUE',/,
     *          1X, '2. RED',/,
     *          1X, '3. GREEN')
2000    FORMAT(10X,'XMIN,XMAX=',2F10.2,/,
     *         10X,'YMIN,YMAX=',2F10.2)
c
c
c
        WRITE(5,4000)
        READ(5,4001) IC,NAM
        NAM(IC+1)=0
        OPEN(UNIT=4,TYPE='OLD',NAME=NAM,ACCESS='SEQUENTIAL',
     *       FORM='FORMATTED',RECORDSIZE=80)
        IR=1
        READ(4,1000) XX(1),YY(1)
        X(1)=XX(1)
        Y(1)=YY(1)
        XMIN=X(1)
        XMAX=X(1)
        YMIN=Y(1)
        YMAX=Y(1)
1       CONTINUE
        IR=IR+1
```

```
      READ(4,1000,END=2)XX(IR),YY(IR)
      X(IR)=XX(IR)
      Y(IR)=YY(IR)
      GO TO 1
2     CONTINUE
      IR=IR-1
      DO 11 I=1,IR
      IF(X(I) .GT. XMAX) XMAX=X(I)
      IF(X(I) .LT. XMIN) XMIN=X(I)
      IF(Y(I) .GT. YMAX) YMAX=Y(I)
      IF(Y(I) .LT. YMIN) YMIN=Y(I)
11    CONTINUE
      XMI=XMIN
      XMA=XMAX
      YMI=YMI
      YMA=YMAX
4     CONTINUE
      XMIN=XMI
      XMAX=XMA
      YMIN=YMI
      YMAX=YMA
      WRITE(5,1001)
      READ(5,1002) ID
      GO TO (21,22) ID
21    write(5,1003)
      READ(5,1002) ICLR
      WRITE(5,2000) XMIN,XMAX,YMIN,YMAX
      CALL PLOT
      WRITE(5,2000) XMIN,XMAX,YMIN,YMAX
      PAUSE 'key in CONTINUE to continue'
      GO TO 4
22    CONTINUE
      CLOSE(UNIT=4)
      END
```

```
C
C
C
C
C       ********************************
C
C       *
C       *     P L O T
C       *
C       ********************************
C
C
C
C       PURPOSE: This routine plot 2D data file and automatically
C                scale the virture window size.
C
C       DEVELOPED BY CAROLINE  WANG
C
C
        SUBROUTINE PLOT
        dimension x(1000),y(1000)
        COMMON /IRG/IXMIN,IXMAX,IYMIN,IYMAX
        COMMON /VTGRph/MINX,MAXX,MINY,MAXY,XMIN,XMAX,YMIN,YMAX
        COMMON /PT/X,Y,IR
        COMMON /IPT/IX,IY
        common /color/iclr
        LOGICAL IFLAG
c
c
c
2001    FORMAT(1X,'OPTION:',/,
     *         1X,'1. USE DEFAULT MIN AND MAX',/,
     *         1X,'2. SELECT MIN AND MAX VALUE')
2002    FORMAT(I2)
2003    FORMAT(1X,'INPUT Y MIN AND MAX IN REAL NUMBER=')
2004    FORMAT(2f10.2)
2103    FORMAT(1X,'INPUT XMIN AND XMAX IN REAL NUMBER=')
2104    FORMAT(2f10.2)
2014    FORMAT(1X,1H',f10.2,1H')
c
c
c
        WRITE(5,2001)
        READ(5,2002) IOPT
        GO TO (31,32) IOPT
31      CONTINUE
        GO TO 3
32      CONTINUE
        WRITE(5,2103)
        READ(5,2104) XMIN,XMAX
        WRITE(5,2003)
        READ(5,2004) YMIN,YMAX
```

```
3       CONTINUE
        YINC=(YMAX-YMIN)/4.
        XINC=(XMAX-XMIN)/4.
        CALL INIT
        call clr
        call colr(iclr)
        CALL DWINDO(XMIN,XMAX,YMIN,YMAX)
        CALL TWINDO(100,1000,100,650)
        YY=Y(1)
        CALL MOVEA(XMIN,YY)
        type *,xmin,yy
        IXMIN=XMIN
        IXMAX=XMAX
        DO 11 I=1,IR
        XX=X(I)
        YY=Y(I)
        CALL DRAWA(XX,YY)
11      CONTINUE
        CALL MOVEA(XMIN,YMIN)
        XX=XMAX
        CALL DRAWA(XX,YMIN)
        CALL DRAWA(XX,YMAX)
        CALL DRAWA(XMIN,YMAX)
        CALL DRAWA(XMIN,YMIN)
        DO 33 I=1,3
        XX=XINC*I+XMIN
        YY=YMIN
        CALL MOVEA(XX,YY)
        CALL DRAWA(XX,YMAX)
33      CONTINUE
        DO 44 I=1,3
        XX=XMIN
        YY=YINC*I+YMIN
        CALL MOVEA(XX,YY)
        CALL DRAWA(XMAX,YY)
44      CONTINUE
        IXINC=(1000-100)/4
        DO 35 I=1,3
        IX=(100+I*IXINC-50)
        IY=50
        call anmode
        CALL MOVABS(IX,IY)
        CALL ANMODE
        XXX=XINC*I+XMIN
        WRITE(5,2014) XXX
35      CONTINUE
        IYINC=(650-100)/4
        DO 36 I=1,3
        IY=(100+IYINC*I)
        IX=10
```

```
        CALL MOVABS(IX,IY)
        YYY=YINC*I+YMIN
        CALL ANMODE
        WRITE(5,2014) YYY
36      CONTINUE
        CALL ANMODE
        call amod
        RETURN
        END
```

```
c
c
c       Developed by: Caroline Wang
c
c
        subroutine demo_text
        logical*1 string(80),quote
        integer*2 ifill,w,h,a
        data quote/1H'/
        jcode=2
        call colr(jcode)
        call dshd(125,164,'0123456789012345678901234567890
     * 0123456789',40,1,2,0,0)
        call putc(125,180)
        call draw(671,180)
        jcode=3
        call colr(jcode)
        call dshd(125,199,'MEM     EC MEMORY MGT GMT DDD/HH:MM:SS',
     *              37,1,2,0,1)
        call dshd(125,229,'MM3A02 34 C CCHCHF',18,1,2,0,1)
        jcode=1
        call colr(jcode)
        call dshd(343,229,'DIS DIS DIS DIS *',17,1,2,0,1)
        jcode=3
        call colr(jcode)
        call dshd(125,279,'   FUNC CONF FX',15,1,2,0,1)
        call dshd(125,329,'1',1,1,2,0,1)
        call dshd(159,329,'RUN    ',7,1,2,0,1)
        call dshd(125,359,'2',1,1,2,0,1)
        call dshd(156,359,'REMOVE ',7,1,2,0,1)
        jcode=2
        call colr(jcode)
        call dshd(125,389,'XYZ >EC MESSAGE LINE NN HH:MM:SS',32,
     *              1,2,0,1)
        jcode=1
        call colr(jcode)
        call dshd(125,389,'XYZ >EC MESSAGE LINE NN HH:MM:SS',32,
     *              1,2,0,1)
        jcode=3
        call colr(jcode)
        call dshd(420,389,'SCRATCH PAD LINE',16,1,2,0,1)
        call demo_window('TEXT',4)
        jcode=2
        call colr(jcode)
        call dshd(150,140,'WOW!',4,15,17,0,0)
        jcode=3
        return
        end
```

```
        subroutine demo_target

        implicit integer(a-z)
c
c
        k=0
        do 100 i=0,14
                do 10 j=0,2
                if(k .gt. 4) k=0
                call colr(k)
                k=k+1
                IR=9*I+3*J
                call circ(398,274,IR,0)
10      continue
100     continue
        k=0
        do 20 i=1,15
                if(k .ge. 4)k=0
                call colr(k)
                k=k+1
                call circ(398,274,141-9*i,1)
20      continue

        end
```

```
c
c
c       Caroline Wang
c
c
c
        subroutine demo_window(string,slen)
        logical*1 string(80)
        integer*2 slen
        type *,' w(v)'
        jcode=3
        call colr(jcode)
        call rect(117,429,563,40,1)
        jcode=0
        call colr(jcode)
        call move(296,429)
        type *,' t(d0,s2)w(c)'
        call text(string,slen)
        type *,' w(v)'
        jcode=0
        call colr(jcode)
        call move(117,120)
        call rect(117,120,565,290,1)
        return
        end
```

```
C
C       CAROLINE WANG
C
        SUBROUTINE VT_PATTERN
        CALL INIT
        CALL CLR
        CALL COLR(1)
        CALL MOVABS(50,100)
        CALL DRWABS(400,100)
        CALL MOVABS(50,150)
        CALL DSHABS(400,150,1)
        CALL MOVABS(50,200)
        CALL DSHABS(400,200,2)
        CALL MOVABS(50,250)
        CALL DSHABS(400,250,3)
        CALL MOVABS(50,300)
        CALL DSHABS(400,300,4)
        CALL COLR(2)
        CALL MOVABS (450,100)
        CALL DRWABS(800,100)
        CALL MOVABS(450,150)
        CALL DSHABS(800,150,1)
        CALL MOVABS(450,200)
        CALL DSHABS(800,200,2)
        CALL MOVABS(450,250)
        CALL DSHABS(800,250,3)
        CALL MOVABS(450,300)
        CALL DSHABS(800,300,4)
        CALL COLR(3)
        CALL MOVABS(50,350)
        CALL DRWABS(400,350)
        CALL MOVABS(50,375)
        CALL DRWABS(400,375,1)
        CALL MOVABS(50,400)
        CALL DSHABS(400,400,2)
        CALL MOVABS(50,425)
        CALL DSHABS(400,425,3)
        CALL MOVABS(50,450)
        CALL DSHABS(400,450,4)
        CALL VTOUT
        END
```

V. FUTURE PLANS

The major plans for the future are to build more generic application program tools for using VTGRAPH. VTGRAPH handles the window management system PLOT10 compatible routines, 2D and 3D graphics. In the near future, it will support more advanced versions of 3D graphics such as showing multiple viewpoints.

The author has 15 years experience in software research and development, with the last couple of years concentrating in AI applications. She realizes the importance of using friendly user interface and generic application programs.

She designed the VTGRAPH package to provide the user better tools to work with on application software.

APPENDIX

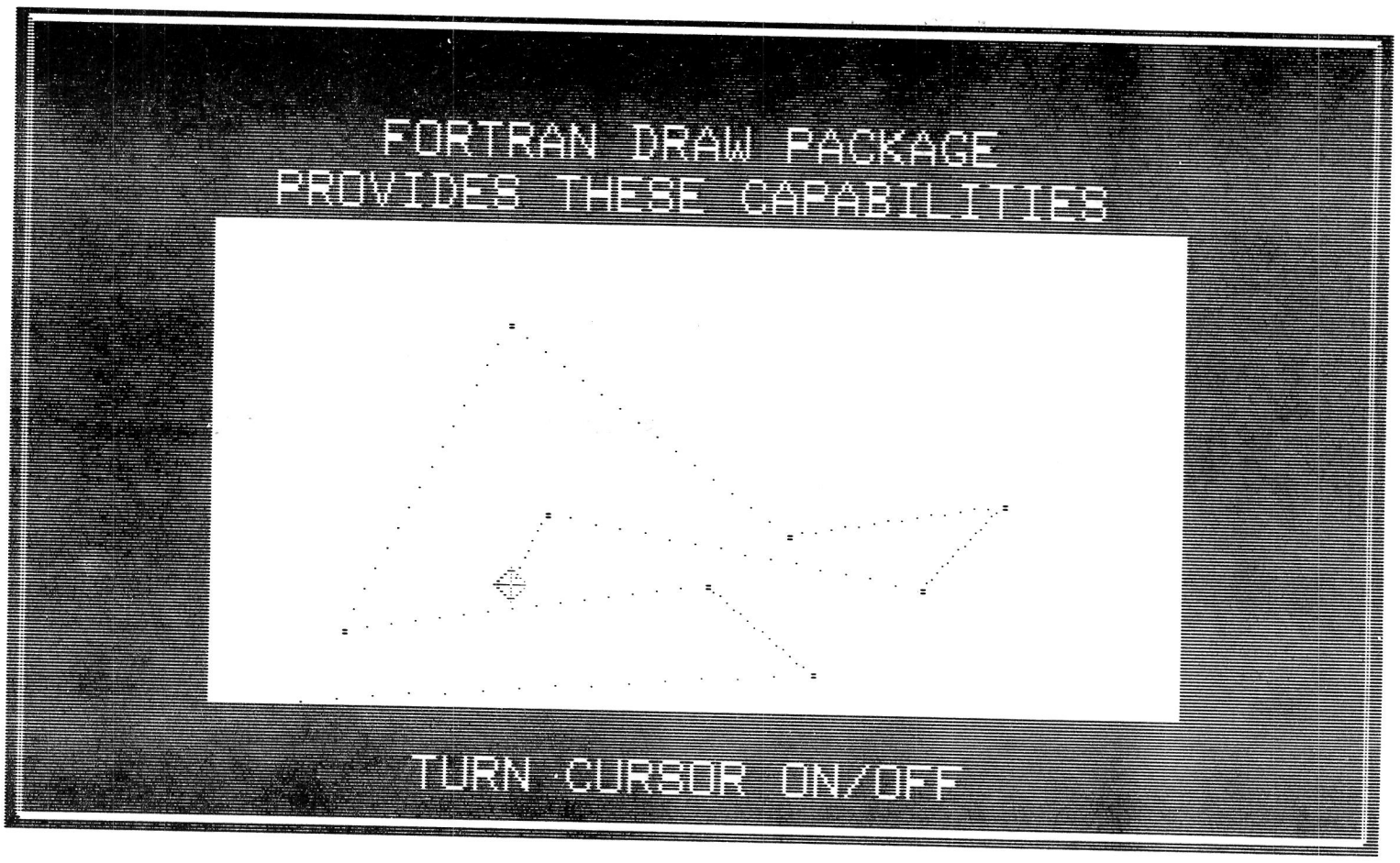
VKGRAPH DEMO PROGRAM

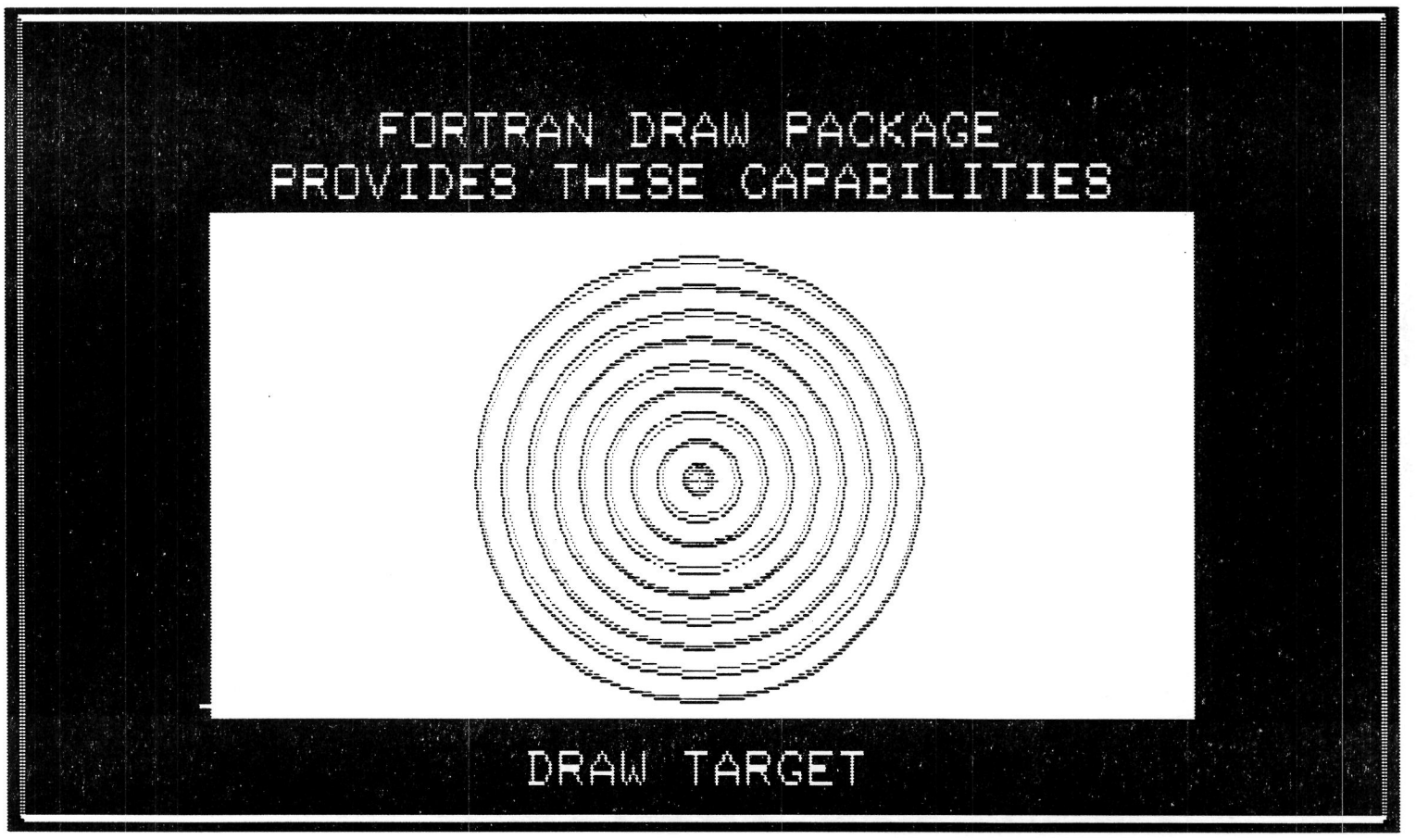
1. VT241 COLOR DEMONSTRATION
2. VT240 SHAD DEMONSTRATION
3. PLOT10 COMPATIBLE DRAWING
4. DASH LINE PATTERN
5. TERMINATE

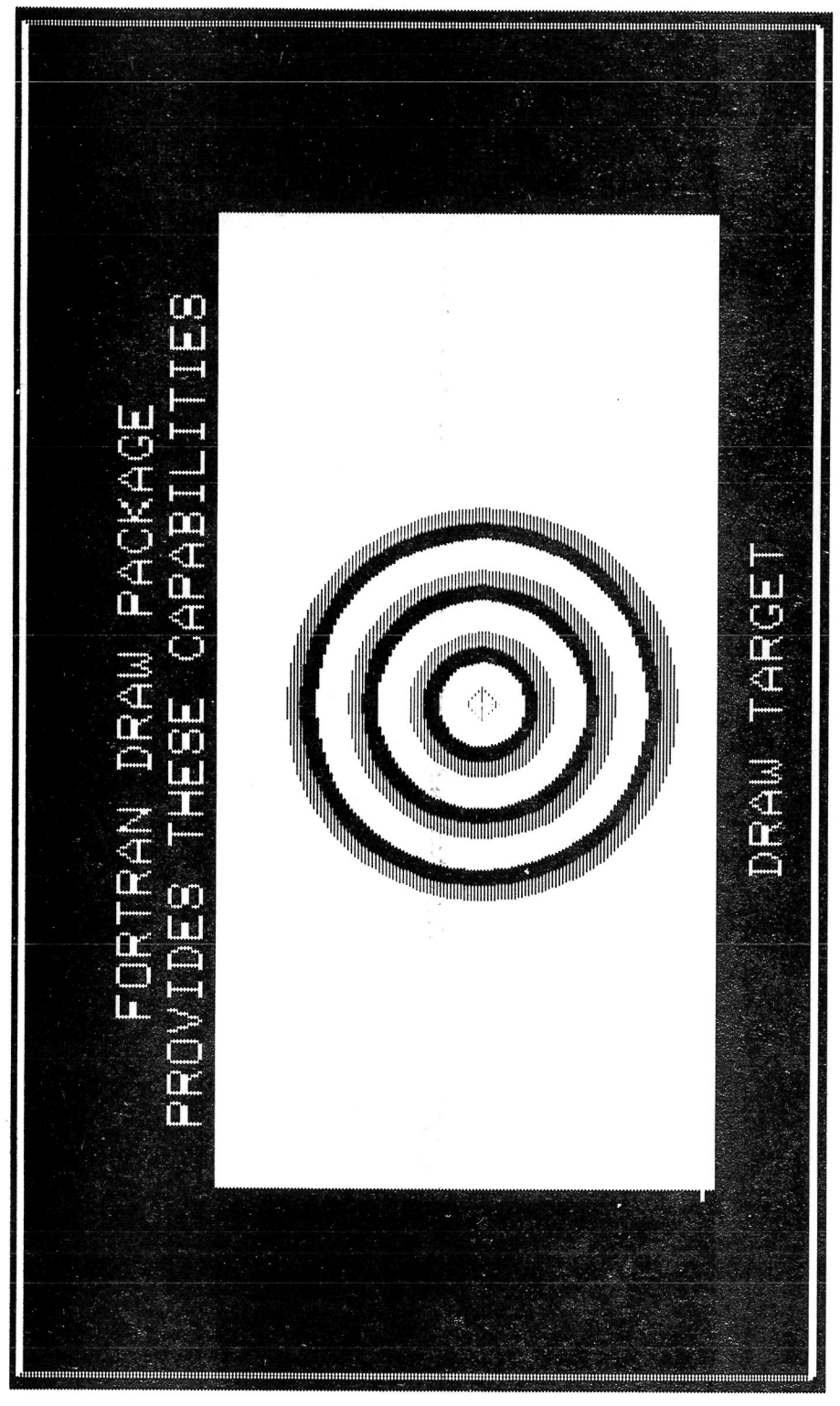
INPUT OPTION:

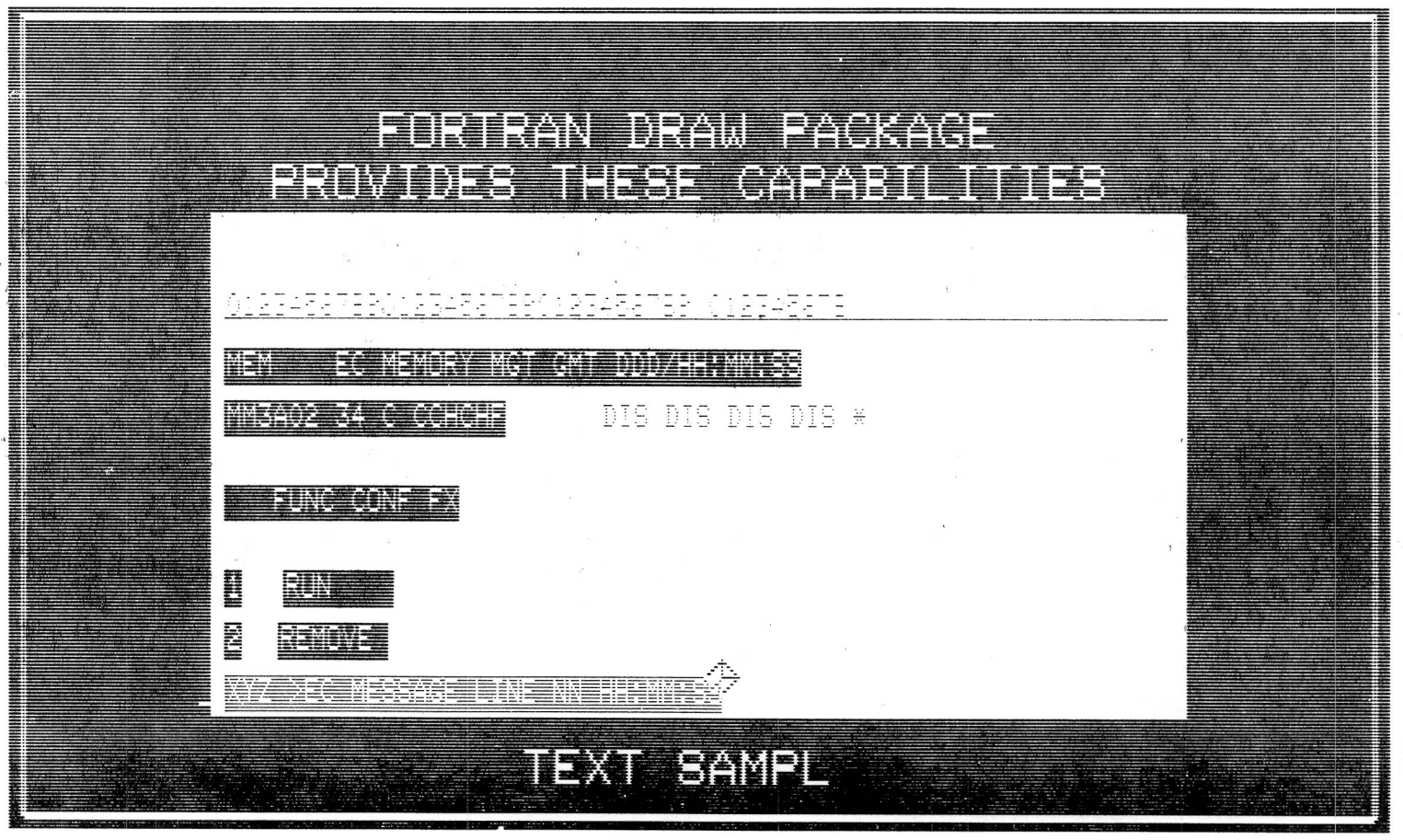
VTGRAPH

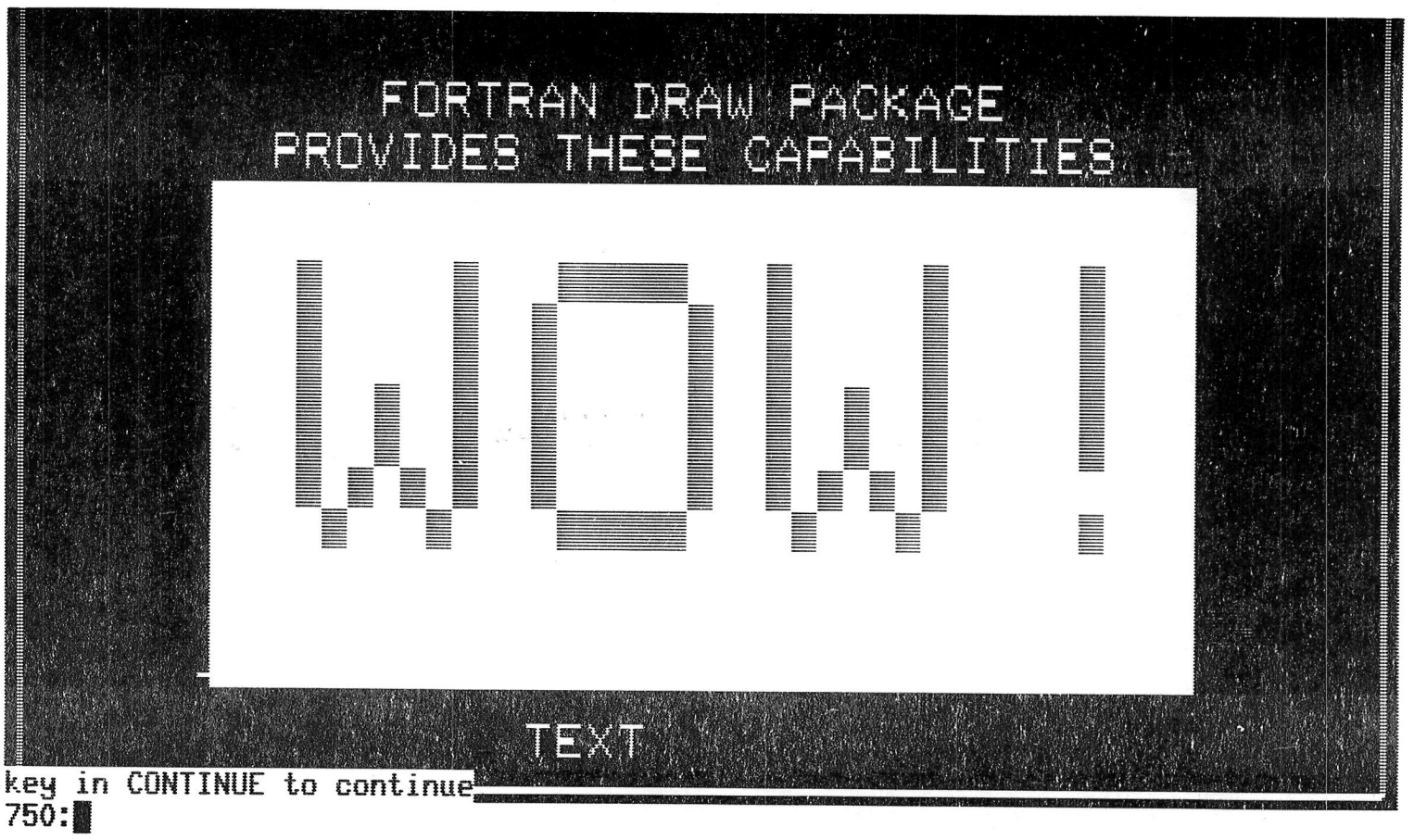
by Caroline Wang/EB44

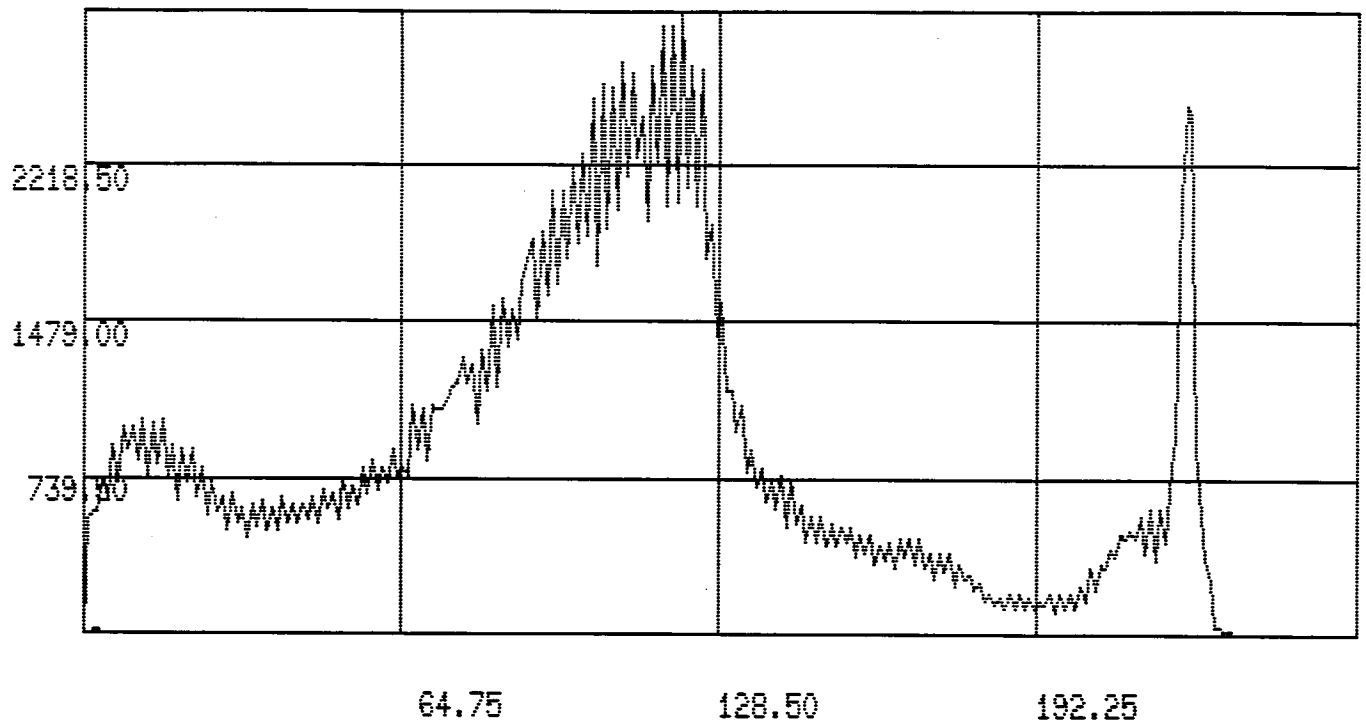

FORTRAN DRAW PACKAGE
PROVIDES THESE CAPABILITIES
TURN CURSOR ON/OFF


FORTRAN DRAW PACKAGE
PROVIDES THESE CAPABILITIES
DRAW TARGET

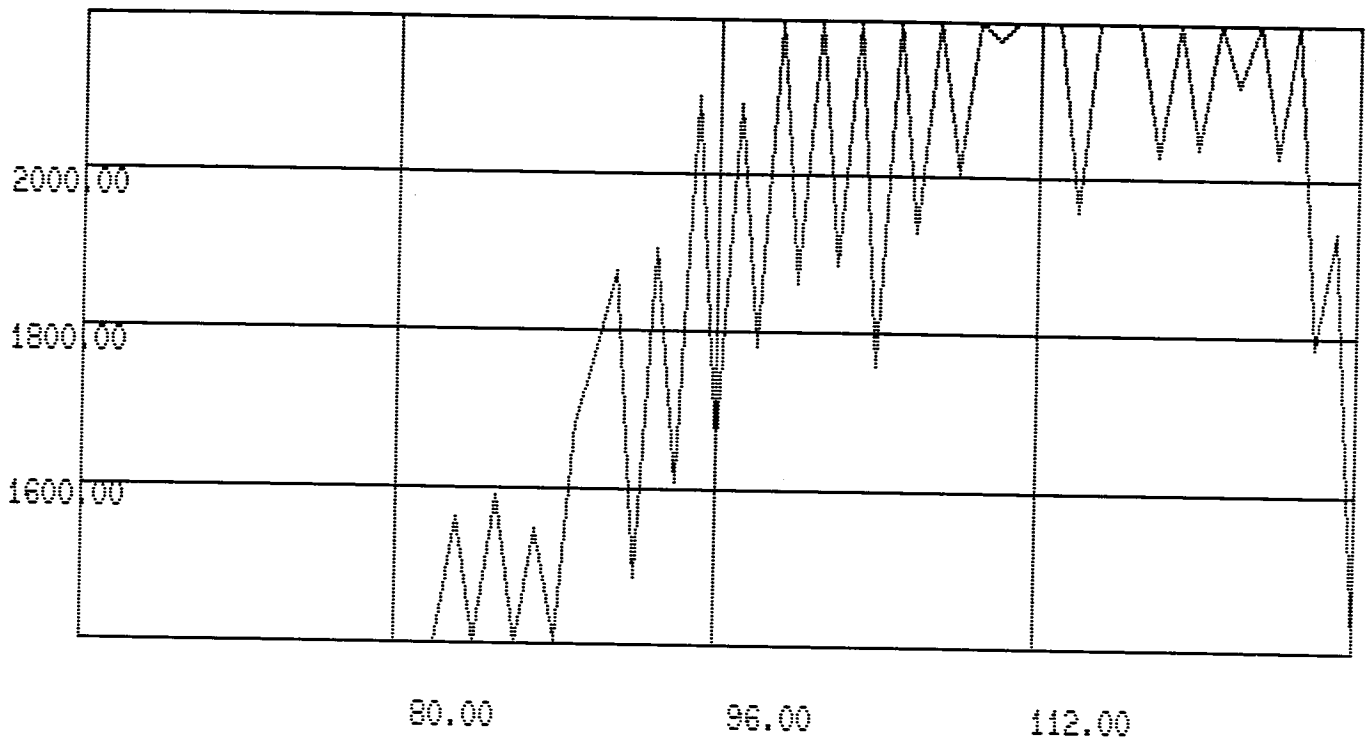

FORTRAN DRAW PACKAGE
PROVIDES THESE CAPABILITIES
DRAW TARGET


FORTRAN DRAW PACKAGE
PROVIDES THESE CAPABILITIES
MEM EC MEMORY MGT GMT DDD/HH:MM:SS
DIS DIS DIS DIS
FUNC CONF FX
1 RUN
2 REMOVE
TEXT SAMPL

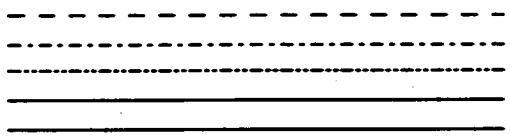

FORTRAN DRAW PACKAGE
PROVIDES THESE CAPABILITIES
WOW!
TEXT
key in CONTINUE to continue
750:

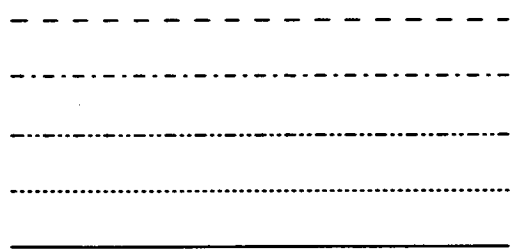



```
      XMIN,XMAX=      1.00    256.00
      YMIN,YMAX=      0.00   2958.00
key in CONTINUE to continue
750:
```




```
        XMIN,XMAX=     64.00     128.00
        YMIN,YMAX=   1400.00    2200.00
key in CONTINUE to continue
750:continue
```





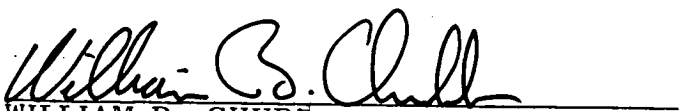
key in CONTINUE to continue
750:

APPENDIX

GRAPHICS SOFTWARE TOOL FOR VT TERMINALS (VTGRAPH)

By Caroline Wang

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.



WILLIAM B. CHUBB
Director, Information and Electronic
Systems Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE: 1988 - 530-050/80067