TEKTRONIX INFORMATION DISPLAY PRODUCTS

TEKPLOT User's Manual

Rdok
1213

PROGRAM REFERENCE

DOCUMENT NO. 062-1383-00

PROGRAM IDENTIFICATION NO. 990032

J. Vlems.

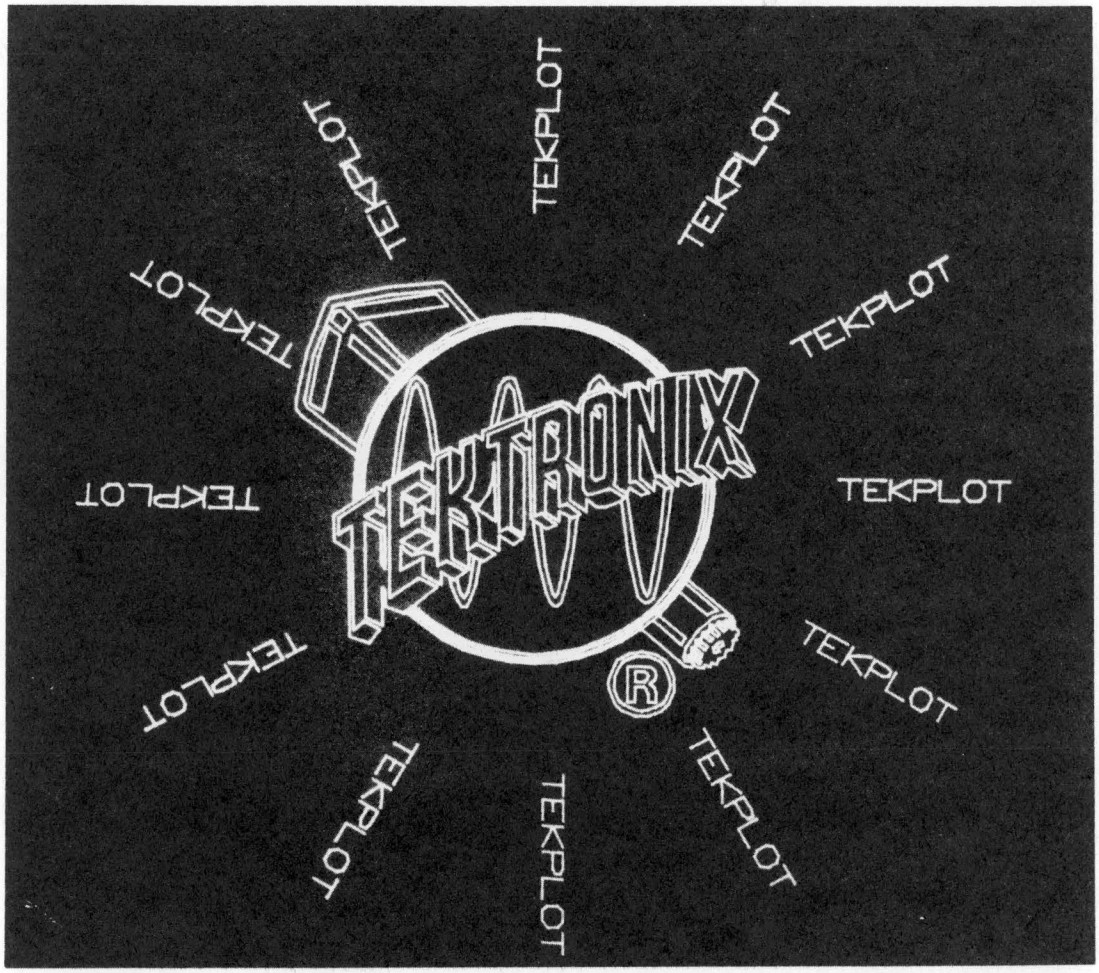
CONTENTS

This program is also available in the following machine-entry forms:

Please place all orders through your Tektronix Applications Engineer.

Tektronix, Inc. • P. O. Box 500 • Beaverton, Oregon 97005 • Phone 644-0161 • Cables: Tektronix

TEKPLOT



A FORTRAN SUBROUTINE PACKAGE

FOR THE

4002A GRAPHIC COMPUTER TERMINAL

TABLE OF CONTENTS

INTRODUCTION

The development of the direct view, bi-stable cathode ray storage tube (Tektronix 611) has been the precursor to a new type of computer terminal. Characterized by high resolution, high speed, and low cost graphics, this new terminal is significantly affecting the whole field of computer graphics. Utilizing an 11 inch direct view storage tube, the Tektronix 4002A Graphic Computer Terminal permits both graphics input and graphics output in addition to usual alphanumeric operation.

The TEKPLOT subroutine package is an extensive set of FORTRAN routines that facilitate the use of the 4002A. Coupled with TEKPLOT, the 4002A becomes a powerful tool for accomplishing computer graphics.

The TEKPLOT routines allow the user to think of the display screen as a "window" into two dimensional space. Under software control, this window may be translated to any particular area of interest, rotated any arbitrary number of degrees, and used as a magnifier.

Flexible conventions allow such window definition to be accomplished either by coordinate definition within a program or by user manipulation of the joystick based upon visual inspection of a plot.

The TEKPLOT routines also provide a natural mechanism for:

1. manipulating the terminal environment: erasing the screen, making a hard copy, determining character size and character font, etc.
2. plotting in any of the available modes, either absolutely or relative to the current position.
3. drawing axes, log axes, grids, and logarithmic grids.
4. performing graphics input of either points or regional coordinates.

and 5. drawing software generated characters of any desired size or orientation.

All TEKPLOT subroutines, with two exceptions, are written in a common subset of FORTRAN. Care has been taken to avoid singular statement types, word size pro-

blems, etc. in order to maximize transferability to any operating system that supports a FORTRAN compiler.

The routines communicate with one another through a central terminal status area in order to retain compatibility with systems that do not support labeled common. Finally, the subroutines are entirely separable making them suitable for immediate inclusion in the existing FORTRAN library.

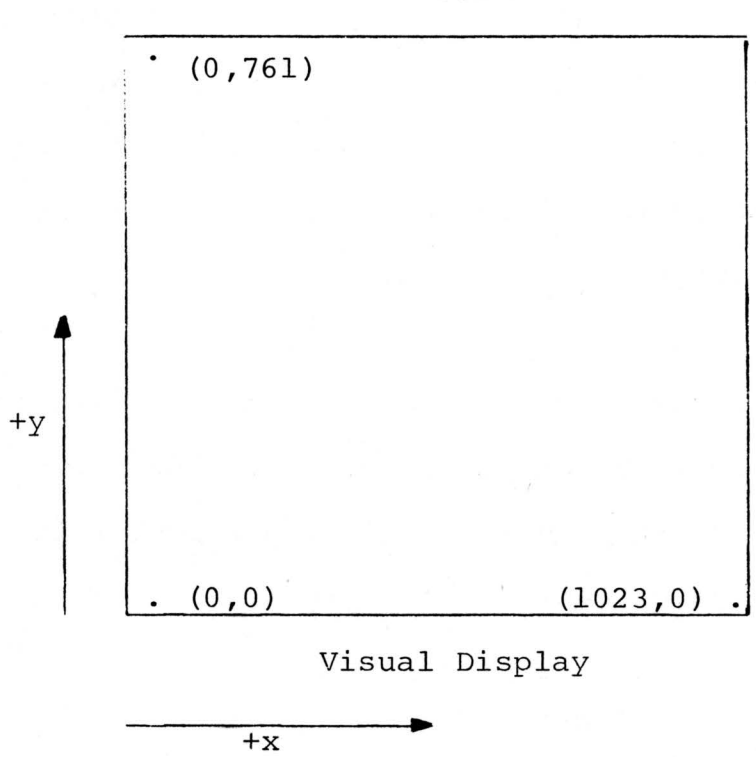
The two exceptions noted above are a character input and a character output routine. Only these two subroutines need be written in a manner particular to the I/O conventions of each operating system.

- NOTE -

There are minor spelling variations of TEKPLOT subroutine names on some time-sharing systems. For details, call or write: Any Tektronix Field Office (57 U.S. and 21 International, or Tektronix, Inc., Beaverton, Oregon 97005.

Description of TEKPLOT Subroutines

Points on the display area range from 0-1023 for X and 0-761 for Y. Point (0,0) is located at the bottom left of the screen.



In the following description, the term tekpoint refers to an (x,y) point in this range. The term scaled unit refers to an (x,y) point that will be scaled, translated, and rotated prior to plotting. Scale factors and axis mappings are established by the user's program with the scaling and axis manipulation routines.

Vectors drawn on the screen are straight line segments that connect the current position to the (x,y) point specified as a parameter in any of the plotting routines. This latter (x,y) point then becomes the new current graphics position.

It is necessary to stress that while tekpoint (0,0) is located at the lower left corner of the screen, the "home" position in alphanumeric mode is situated at the top left corner. This double convention allows the screen to be viewed as a page of textual information in alphanumeric mode, whereas in graphics mode the positive directions of the axes conform to the usual conventions for the first quadrant. (It will be seen in the ensuing description that the user is free to establish new axis conventions as he wishes.) In keeping with this double convention, the TEKPLOT routines distinguish between plotting and printing in the following way: the current position in graphics mode is not affected by an intervening printing that occurs in alphanumeric mode.

If the preceding conventions are not at once clear, a few moments experimentation with the terminal will make them seem obvious and natural.

The individual subroutines are described in the following paragraphs.

Utility Routines

The utility routines are used to manipulate the environment in which graphic and/or alphanumeric data will be displayed.

NAME: BEGIN

PURPOSE: Initializes the TEKPLOT routines and determines the delay required for ERASE and HCOPY.

USAGE: CALL BEGIN (JBAUD)

PARAMETERS: JBAUD - the Baud rate to be assumed for all further communication.

DESCRIPTION: BEGIN should always be the first subroutine to be called in any program; it should be called only once. BEGIN is used to initialize the TEKPLOT routines. Initialization is accomplished by presetting values in the Terminal Status Area (TSA). All TEKPLOT subroutines then communicate with one another through the TSA.

After calling BEGIN, the 4002A will be initialized to:

1. Alphanumeric mode
2. Current graphics position (0,0)
3. 1 tekpoint/scaled unit on both axes
4. An origin of (0,0)
5. Rotation of 0 degrees
6. Normal character size
7. Normal character font
8. A window size equal to the screen size

BEGIN also determines delay intervals required for the ERASE and HCOPY routines from the specified Baud rate.

SUBRS CALLED: TEK001, VECTOR, ALPHA

NAME: ERASE

PURPOSE: To erase the screen.

USAGE: CALL ERASE

DESCRIPTION: The ERASE subroutine erases the screen, places the terminal in alphanumeric mode, and positions the alpha cursor to the home position. ERASE does not affect the current graphics position.

SUBRS CALLED: HOME, CHOUT, TEK001

NAME: HOME

PURPOSE: Positions the alpha cursor to the upper left corner of the screen.

USAGE: CALL HOME

DESCRIPTION: HOME places the terminal in alphanumeric mode at the home position. The current graphics position is not affected.

SUBRS CALLED: ALPHA, CHOUT

NAME: TPAUSE

PURPOSE: To I/O bound a program.

USAGE: CALL TPAUSE

DESCRIPTION: The TPAUSE routine is a convenient way for a user to I/O bound himself in a time-sharing environment. TPAUSE places the terminal in alphanumeric mode at home position, then reads one character from the terminal, discards that character, and exits. The routine is intended for a user who wishes to pause at some point in his program in order to inspect his plotted output, and then resume execution by typing any character. TPAUSE does not affect the current graphics position.

SUBRS CALLED: HOME, CHIN

NAME: DOUBLE

PURPOSE: To place the 4002A into double size mode.

USAGE: CALL DOUBLE

DESCRIPTION: After calling DOUBLE, all subsequent plotting will be double intensity (i.e., broader vectors and points) and alpha-numeric characters will be twice as large.

SUBRS CALLED: CHOUT, TEK001

NAME: ITALIC

PURPOSE: To set italicized character mode.

USAGE: CALL ITALIC

DESCRIPTION: After calling ITALIC, all further alphanumeric characters will be italicized.

SUBRS CALLED: CHOUT, TEK001

NAME: NORMAL

PURPOSE: To reset double size and italicized modes.

USAGE: CALL NORMAL

DESCRIPTION: Calling NORMAL resets the mode in which double intensity graphics and double size and/or italicized characters occur.

SUBRS CALLED: VECTOR, POINT, TEK001, CHOUT

NAME: HCOPY *

PURPOSE: To make a hard copy.

USAGE: CALL HCOPY

DESCRIPTION: The HCOPY subroutine causes a hard copy of the current screen to be generated. After calling HCOPY, a delay will occur during the copying process.

HCOPY does not affect the user's current mode or current graphics location.

SUBRS CALLED: CHOUT, TEK001

* For use with the 4601 Hard Copy Unit

NAME:	SPAD
PURPOSE:	To "address" the scratch pad area.
USAGE:	Call SPAD
DESCRIPTION:	Calling SPAD sets the terminal to its normal alphanumeric mode and causes subsequent data received to be entered into the scratch pad area.
SUBRS CALLED:	CHOUT, NORMAL

NAME:	CLEAR
PURPOSE:	To "address" the Scratch Pad and clear its contents.
USAGE:	Call CLEAR
DESCRIPTION:	Calling CLEAR will "address" the Scratch Pad, clear the contents and cause subsequent received data to be entered starting at the left-most position.
SUBRS CALLED:	CHOUT, SPAD

NAME: EDIT

PURPOSE: To place the Scratch Pad in Edit mode.

USAGE: Call EDIT

DESCRIPTION: Calling EDIT places the terminal in Edit mode at the current Scratch Pad position and refreshes the contents, i.e. make the contents visable.

SUBRS CALLED: CHOUT

NOTE: The Scratch Pad must be "addressed" before calling EDIT.

NAME: REFRES

PURPOSE: To refresh the contents of the Scratch Pad, i.e. make visable.

USAGE: Call REFRES

DESCRIPTION: Calling REFRES causes the current contents of the Scratch Pad to be refreshed.

SUBRS CALLED: CHOUT

NOTE: The Scratch Pad must be "addressed" before calling REFRES.

Mode Routines

The mode routines are used to determine the mode in which all subsequent printing or plotting will take place. Five physical modes are available in the 4002A. These include:

a. alphanumeric

b. vector plot

c. point plot

d. incremental plot

e. graphics input

The TEKPLOT mode routines govern only the first three of these modes. Incremental plot mode is dealt with as a special case by the subroutine INCP, and graphics input mode is treated separately by the graphics input routines.

The 4002A functions as a teletype in alphanumeric mode. The terminal is automatically placed in this mode when it is turned on, and a carriage return* or the Home button will reset to this mode. When in alphanumeric mode, the current character position is signaled by the location of the flickering cursor.

* Available as a wire-strapped option

NAME: VECTOR

PURPOSE: To place the 4002A into vector mode.

USAGE: CALL VECTOR

DESCRIPTION: Calling VECTOR places the 4002A in vector mode and positions to the current graphics location. All subsequent plotting in this mode will be in the form of smooth lines connecting each specified point.

SUBRS CALLED: TEK001, TEK002, CHOUT

NAME: POINT

PURPOSE: To place the 4002A into point plot mode.

USAGE: CALL POINT

DESCRIPTION: Calling POINT places the terminal in point plot mode and positions to the current graphics location. All subsequent plotting in this mode will be in the form of individual points that are drawn at each specified coordinate.

SUBRS CALLED: CHOUT, TEK001

Vector mode is used to draw a smooth line between the current graphics position and the (x,y) point specified as a parameter to any of the plotting routines. This latter point then becomes the new current graphics position.

Point plot mode causes a point to appear at the (x,y) point specified as a parameter in a call to any of the plotting routines. This (x,y) point then becomes the new current graphics position.

NAME: ALPHA

PURPOSE: To place the 4002A into alphanumeric mode.

USAGE: CALL ALPHA

DESCRIPTION: Calling ALPHA places the terminal in alphanumeric mode. The alpha cursor will appear at the current (x,y) position.

SUBRS CALLED: CHOUT, TEK001

Axis Manipulation Routines

The axis manipulation routines are used to accomplish scaling, axis translation, axis transformation, and to draw a visible axis.

NAME: SCALE

PURPOSE: A call to SCALE determines scale factors and establishes an origin for any subsequent plotting. If SCALE is not called, then the indicated default values are assumed.

USAGE: CALL SCALE (XFACT, YFACT, XORG, YORG)

PARAMETERS:

XFACT - the number of tekpoints per scaled unit on the x-axis. The sign of XFACT determines the direction of the x-axis, i.e., if XFACT is negative, the positive direction of the x-axis is to the left. Default value = 1.

YFACT - the number of tekpoints per scaled unit on the y-axis. The sign of YFACT determines the direction of the y-axis. Default value = 1.

XORG - the origin on the x-axis expressed in tekpoints. Default value = 0.

YORG - the origin on the y-axis expressed in tekpoints. Default value = 0.

DESCRIPTION: SCALE may be called whether or not an explicit axis is drawn. XFACT and YFACT act as multiplicative factors that determine the conversion from tekpoints to scaled units. XORG and YORG act as linear displacements from tekpoint (0,0). For example, the physical origin of the 4002A is tekpoint (0,0) at the lower left corner of the screen. To establish a new origin at the center of the screen, the user may

```
CALL SCALE (1.,1.,512.,380.)
```

This new origin then obtains for all subsequent plotting or until the next call to SCALE.

SUBRS CALLED: TEK001

NAME: ROTATE

PURPOSE: By calling ROTATE, the user declares himself to be working in a new transformed axis that is rotated the specified number of degrees with respect to the Cartesian system of the screen. The center of rotation is the user's origin, scaled unit (0,0).

USAGE: CALL ROTATE (DEG)

PARAMETERS: DEG - the number of degrees (not radians) of rotation from the orthogonal system of the screen. Default value = 0. Positive rotation is in a counter clockwise direction.

DESCRIPTION: After calling ROTATE, all further plotting will occur in the new transformed coordinate system until the next call to rotate.

SUBRS CALLED: TEK001

NAME: WINDOW

PURPOSE: The WINDOW subroutine is used to set the window size for all subsequent plotting. Any plotting that extends beyond the current window will be clipped at the window margin.

USAGE: CALL WINDOW (XMIN, YMIN, XMAX, YMAX)

PARAMETERS:

XMIN - the minimum x value of a rectangular window. XMIN should be given in scaled, rotated units.

YMIN - the minimum y value of the window. YMIN should be given in scaled, rotated units.

XMAX - the maximum x value of the window, specified in scaled, rotated units.

YMAX - the maximum y value of the window, specified in scaled, rotated units.

DESCRIPTION: WINDOW converts the user's parameters to tekpoints and saves them in the terminal status area for future use. If the user's XMAX parameter turns out to be smaller than XMIN, then these are interchanged. Similarly, if YMAX is less than YMIN, then these are interchanged.

If the WINDOW parameters specify a window that is larger than the current screen size, then the designated window is collapsed to the screen size.

Because window parameters are specified in terms of scaled units, parameters may be set by visual inspection of a plot using the CURSIS subroutine.

WINDOW may be used to shrink the effective screen size in order to create protected

areas on the screen. The routine may also be used to display multiple plots on the same screen or bring disparate parts of a plot together for comparison.

SUBRS CALLED: TEK001

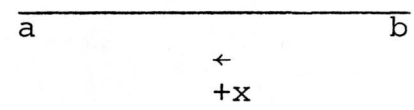
NAME: AXIS

PURPOSE: The AXIS subroutine is used to draw x and y axes, together with tic marks or grid lines, if desired. Each axis may be either linear or logarithmic (base 10).

USAGE: CALL AXIS (XLOW, YLOW, XLNG, YLNG, XTIC, YTIC, MARKX, MARKY)

PARAMETERS: XLOW - the starting point on the x axis, expressed in scaled, rotated units. XLOW must be the lowest value that the x axis will assume.

If, for example, an x-axis is to be drawn from point a to point b on the figure below,



and if the user has reversed the positive direction of the axis (by calling SCALE with negative XFACT), then b in the figure is in the starting position.

YLOW - the starting point on the y axis, expressed in scaled, rotated units. YLOW must be the lowest value that the y axis will assume.

XLNG - the length of the x axis, expressed in scaled units. XLNG is not a signed quantity; AXIS always takes the absolute value of the user's XLNG parameter.

YLNG - the length of the y axis, expressed in scaled units. YLNG is not a

signed quantity; AXIS always takes the absolute value of the user's YLNG parameter.

XTIC - the distance between tic marks on the x axis, expressed in scaled units. If XTIC is negative, then the x axis will be a log axis, and the absolute value of XTIC will be the distance between decades.

YTIC - the distance between tic marks on the y axis, expressed in scaled units. If YTIC is negative, then the y axis will be a log axis, and the absolute value of YTIC will be the distance between decades.

MARKX - determines axis annotation on the x axis. If MARKX = 0, then no axis annotation will be included. If MARKX > 0, then the x axis will be drawn with tic marks. If MARKX < 0, then the x axis will be drawn with grid lines.

MARKY - determines axis annotation on the y axis. If MARKY = 0, then no axis annotation will be included. If MARKY > 0, then the y axis will be drawn with tic marks. If MARKY < 0, then the y axis will be drawn with grid lines.

DESCRIPTION: AXIS converts the starting points, XLOW and YLOW, to their tekpoint equivalents to determine the correct starting values. XLNG and YLNG are converted to tekpoints and added to the starting points to determine the correct terminating values. If the user has called ROTATE, then the drawn axis will be appropriately rotated.

AXIS draws a rectangular axis passing

through the user's origin and then returns. The routine leaves the terminal in vector mode positioned at the origin. Each axis line specified by the AXIS parameters must pass through the user's origin; if not the routine returns without drawing an axis.

Example: The user wishes to establish an origin at the center of the screen. He wishes to draw axes that extend the length of the screen in either direction. Tic marks are to be 20 tekpoints apart on the x-axis and grid lines are to be 40 tekpoints apart on the y-axis. The following statements would suffice:

```
CALL SCALE (10.,10.,512.,380.)
CALL AXIS (-51.,-38.,102.,76.,2.,4.,1,-1)
```

SUBRS CALLED: TEK001, TEK002, TEK007, VECTOR

Character I/O Routines

The character I/O routines are not intended for use by the typical applications programmer. They are provided because the ASCII control characters needed to drive the 4002A are often not included in the FORTRAN character set.

Since the TEKPLOT package does provide the necessary control characters as required, the user need never concern himself with details of this type.

It should be noted that if a user elects to effect his own control of the terminal using character I/O, then the TEKPLOT routines cannot keep accurate track of current terminal status. For this reason, the user is advised to call ALPHA and then reinitialize his current graphics position after character I/O has occurred.

NAME: CHIN

PURPOSE: CHIN is a character input routine.

USAGE: CALL CHIN (J)

PARAMETERS: J - the ASCII value of a character input to the 4002A.

DESCRIPTION: CHIN is used to input a single character, J. For example, if the FORTRAN statement

```
CALL CHIN (J)
```

is executed, and the user depresses the BEL key, then the value 7 will be assigned to J.

SUBRS CALLED: None

NAME: CHOUT

PURPOSE: CHOUT is a character output routine.

USAGE: CALL CHOUT (J)

PARAMETERS: J - the ASCII value of a character to be output to the 4002A.

DESCRIPTION: CHOUT is used to output a single character, J. For example,

```
CALL CHOUT (7)
```

will result in a bell tone.

SUBRS CALLED: None

Plotting Routines

The plotting routines are used to plot data in the currently selected mode (vectors or points); the mode must be set prior to calling a plotting routine. Thus, a call to TPLOT while in vector mode will cause a line to be drawn from the current graphics position to the (x,y) point specified as a parameter to TPLOT. This latter (x,y) then becomes the new current graphics position. In a similar way, a call to TPLOT while in point plot mode causes a point to be drawn at the (x,y) position specified in the call; this (x,y) point then becomes the new current graphics position. (INCP is an exception to the above rules, as will be seen.)

Two types of plotting routines are provided. TPLOT takes as parameters an absolute (x,y) point in the user's established coordinate system. DELTA takes as parameters a (Δx, Δy) relative to the current position. Both TPLOT and DELTA commence at the current graphics positions, and both update the current graphics position after plotting has occurred.

NAME: TPLOT

PURPOSE: The TPLOT subroutine is used to plot to an absolute coordinate position in the current mode.

USAGE: CALL TPLOT (X, Y, IPEN, MARK)

PARAMETERS:

X - an X value, expressed in scaled, rotated units.

Y - a Y value, expressed in scaled, rotated units.

IPEN - determines whether the pen is up or down (z-axis off or on). If IPEN = 0, then the pen is up. If IPEN $\neq$ 0, then the pen is down.

MARK - the number of a data mark taken from the following list:

0 = no data mark
1 = small x
2 = small down arrow
3 = small up arrow
4 = small square
5 = small triangle
6 = small asterisk
11 = large x
12 = large down arrow
13 = large up arrow
14 = large square
15 = large triangle
16 = large asterisk

Any other number other than one of those from the above list is equivalent to no data mark, i.e., MARK = 0.

DESCRIPTION: The TPLOT routine converts the parameters X and Y to tekpoints and rotates them appropriately prior to plotting. If TPLOT has

been called with an X or Y that, after scaling and rotation, turns out to be outside the window limits, then the routine plots only that portion of the plot within the window.

Calling TPLOT in alpha mode is a legal operation, providing that the pen is up. The effect is to position the alpha cursor and then return the user to alpha mode. However, calling TPLOT while in alpha mode with the pen down will yield an undesirable result.

After calling TPLOT, the current graphics position is appropriately updated.

SUBRS CALLED: TEK001, TEK002, VECTOR, POINT, ALPHA, TEK006, CHOUT

NAME: DELTA

PURPOSE: The DELTA routine is used for local plotting, relative to the current graphics position. Plotting takes place in the current mode.

USAGE: CALL DELTA (DELX, DELY, IPEN, KRET)

PARAMETERS:

DELX - a quantity, ΔX, relative to the current X position, expressed in scaled, rotated units.

DELY - a quantity, ΔY, relative to the current Y position, expressed in scaled, rotated units.

IPEN - a pen value that governs whether the pen is up or down. If IPEN = 0, then the pen is up. If IPEN $\neq$ 0, then the pen is down.

KRET - determines whether DELTA will return to the original graphics position after plotting an increment of (Δx, Δy). If KRET = 0, then DELTA does not plot back. If KRET $\neq$ 0, then DELTA does plot back again.

DESCRIPTION: DELTA converts the parameters DELX and DELY to tekpoints and adds them to the current graphics position. DELTA then plots to this position in the current mode. If KRET $\neq$ 0, then DELTA positions back to the starting point again.

The same conventions that obtain for PLOT also apply to DELTA, viz. plotting is constrained to that portion of the plot within the current window; calling DELTA with pen down while in alpha mode will produce weird effects; and after calling

DELTA the current graphics position is appropriately updated.

SUBRS CALLED: TEK001, TEK002, VECTOR, ALPHA, POINT

NAME: INCP

PURPOSE: INCP is an incremental plotting routine that utilizes the incremental plot mode of the terminal. It is treated as a special case with respect to the user's current mode. In particular, the user may call INCP while in any mode; the routine will perform incremental plotting as specified and then return the user to his current mode.

USAGE: CALL INCP (JDIR, IPEN, NUM, KRET)

PARAMETERS:

JDIR - a variable or constant that specifies one of eight possible directions for the plotted increment. Directions are:

0 - N	4 - S
1 - NE	5 - SW
2 - E	6 - W
3 - SE	7 - NW

If JDIR is not one of the above directions, INCP just returns.

IPEN - determines whether the pen is up or down. If IPEN = 0, then the pen is up. If IPEN $\neq$ 0, the pen is down.

NUM - the number of increments to be plotted in the specified direction.

KRET - determines whether INCP will plot back to the original position after the increment has been plotted. If KRET = 0, then INCP will not plot back again. If KRET $\neq$ 0, then INCP does plot back.

DESCRIPTION: INCP does not update the current graphics position. The routine will commence incremental plotting at the current alpha position if the user is in alpha mode, or at the current graphics position if the user is in vector or point plot mode. In the latter case, the user is returned to his current graphics position after incremental plotting has been completed. Unlike the other routines, INCP does not constrain plotting to the current window. Off-screen plotting on the 4002A will "wrap around" to the other side.

Example: The user wishes to write the line

CASE 1: X^2+C

at the top of his plot, and then to plot a curve of some sort. The program might begin as follows:

```
            PROGRAM SUPRSCPT
            CALL BEGIN
            CALL ERASE
            WRITE (61,100)
   100      FORMAT (1H+,'CASE 1:X+ C')
   C        The + carriage control charac-
   C        ter suppresses the line feed.
            CALL INCP (0,0,7,0)
   C        The preceding statement posi-
   C        tions the alpha cursor to write
   C        a superscript.
            WRITE (61,200)
   200      FORMAT (9X, '2')
            CALL VECTOR
              :
              :
```

SUBRS CALLED: CHOUT, TEK001, TEK006, ALPHA, VECTOR, POINT

Graphics Input Routines *

The graphics input routines enable the user to input graphic information from the 4002A to his program. When a graphic input routine is called, a full screen crosshair cursor appears on the screen. This cursor may be positioned using the joystick adjacent to the terminal. After the cursor is positioned, the user causes position coordinates to be passed back to the graphics input routine by striking a key on the keyboard. When the coordinates have been sent to the computer, the graphics cursor disappears.

Since any key will cause position coordinates of the graphics cursor to be transmitted, the key may be thought of as special function buttons for graphics input mode. The user might choose one of several subroutines depending upon which key was depressed.

* This section is applicable only to units equipped with the Interactive Graphics Unit and Joystick.

NAME: CURSIS

PURPOSE: CURSIS is the basic graphics input routine. When called, CURSIS causes the graphics cursor to appear. The user then positions the cursor and depresses a key.

USAGE: CALL CURSIS (I, X, Y)

PARAMETERS:

I - an ASCII character code corresponding to the key that was depressed.

X - the location of the x-axis corresponding to the cursor position, converted to scaled, rotated units.

Y - the location of the y-axis corresponding to the cursor position, converted to scaled, rotated units.

DESCRIPTION: CURSIS reads the coordinate position of the cursor and converts it to a scaled coordinate (x,y). These scaled values are then assigned to the user's x and y parameter locations. The ASCII value of the key that was depressed is assigned to parameter location I. CURSIS may be called while the user is in any mode. After cursor coordinates have been input, the routine will return the user to his current mode.

SUBRS CALLED: TEK001, TEK004

NAME: SETGRD

PURPOSE: SETGRD is intended for use with the GRID routine. Calling SETGRD partitions the screen into a grid comprised of X*Y rectangles.

USAGE: CALL SETGRD (X, Y, IPEN)

PARAMETERS:

X - the number of regions along the x-axis. $X \geq 0$.

Y - the number of regions along the y-axis. $Y \geq 0$.

IPEN- if IPEN = 0, then SETGRD values will be retained, but no explicit grid will be drawn. If IPEN $\neq$ 0, then SETGRD values will be retained and an explicit X*Y grid will be drawn on the screen.

DESCRIPTION: It is often convenient to draw an explicit grid while debugging a program. When the program is debugged, IPEN can be set to zero and the drawn grid will be suppressed.

SETGRD does some internal rounding in the event that X or Y do not partition the axes uniformly. However, for best results with large X or Y, the user should choose a number of regions that is an integral divisor of the axis in question.

SETGRD may be called from any mode. The user is returned to his current mode by the routine. The parameters X and Y are not affected by the user's current scaling or rotation.

SUBRS CALLED: TEK001, TEK002, ALPHA, VECTOR, POINT

NAME: GRID

PURPOSE: GRID is a graphics input routine. It must be used in conjunction with SETGRD. When called, GRID causes the graphics cursor to appear. The user then positions the cursor and depresses a key. GRID returns a row and columns coordinate that is determined by the cursor location within the grid specified by the last call to SETGRD

USAGE: CALL GRID (J, JROW, JCOL)

PARAMETERS:

J - an ASCII character code corresponding to the key that was depressed.

JROW- the row of the current grid corresponding to the cursor position.

JCOL- the column of the current grid corresponding to the cursor position.

DESCRIPTION: GRID allows the user to graphically input a region coordinate in lieu of a point coordinate. It is useful for menu-picking or selecting areas of interest in graphs.

GRID reads the coordinate position of the cursor and converts this value to a (row, column) coordinate based upon the grid size specified in the user's last call to SETGRD. If the cursor falls exactly at the dividing line between two adjacent cells in the grid, the larger coordinate is selected.

Row and column coordinates returned by GRID range from 1 to X and Y respectively. That is, the first row is nominally called row 1; the first column is called column 1. An exception to this indexing

scheme occurs if SETGRD was called with X or Y = 0. In this case, GRID will return a zero row or column coordinate corresponding to the axis that had zero partitions. GRID may be called from any mode. The user is returned to his current mode by this routine.

The parameters JROW and JCOL are not affected by current scaling or rotation.

Example: The user wishes to write N lines of descriptive information on the screen. Each line corresponds to a choice. When the program is executed, an option will be selected by positioning the graphics cursor anywhere on the chosen line. Assuming the user has written the code to print N options, one per line, the code to govern menu-picking might look as follows:

```
C     The 4002A has 39 lines in
C     alphanumeric mode.
      CALL SETGRD (0.,39.,0)
      CALL GRID (KHAR,LINE,NULL)
C     At this point, LINE will have
C     a value equal to the index of
C     the chosen option.
```

SUBRS CALLED: TEK001, TEK004

Software Character Generator

The software character generator is used to draw a 48 character set. The character size and orientation of all software generated characters may be varied under program control.

NAME: WRITEX

PURPOSE: WRITEX is a software character generator. The characters to be written are supplied as parameters to this routine.

USAGE: CALL WRITEX (NUM, JARRAY)

PARAMETERS:

NUM - the number of characters to be generated.

JARRAY - an integer array that contains the ASCII codes of the characters to be generated, one character code per word.

DESCRIPTION: All characters are drawn in vector mode. The lower left corner of each character commences at the current graphics position. Upon completion, the current position will be updated to the lower left corner of the next character to be drawn.

WRITEX may be called from any mode. The routine will print characters as specified and then return the user to his current mode.

Normal character size is 8x8 tekpoints; normal print direction is horizontally left to right. WRITEX leaves a 1/4 character width between adjacent characters and a 1/2 character height between adjacent lines. The routine is affected by current scaling and rotation. Thus, it is possible to write smaller, larger, taller, or fatter characters, print at an angle, etc.

The WRITEX character set includes:

A through Z
0 through 9
space, -, /, =, ., +, [,]
A carriage return is represented by a $
A line feed is represented by a)
An underline is represented by a (
A backspace is represented by a <

Any character not included in the 48 character set listed above will be ignored.

Example: The user wishes to use WRITEX to generate the whole character set. He wishes to write the characters at the top of his screen in double size.

```
      PROGRAM CHRSET
      DIMENSION JARRAY (61)
      CALL BEGIN
      CALL ERASE
      CALL TPLOT (10.,700.,0,0)
C     Character generation begins at
C     the current graphics position.
C     It is possible to plot in ALPHA
C     mode as long as the z-axis is
C     off.
      CALL SCALE (2.,2.,0,0)
C     Double the character size.
      DO 100 I = 32,93
      JARRAY(I-31)=I
C     Characters not included in the
C     WRITEX character set will not
C     be printed.
100   CALL WRITEX (61,JARRAY)
      END
```

SUBRS CALLED: ALPHA, VECTOR, POINT, TEK001, TEK005

NAME: WRITEY

PURPOSE: WRITEY is a software character generator similar to WRITEX. However, WRITEY allows the user to supply local scaling and rotation parameters.

USAGE: CALL WRITEY (FACT, DEG, NUM, JARRAY)

PARAMETERS:

FACT - a local scaling factor that scales the normal character size.

DEG - a local rotation factor that determines the number of degrees of rotation from the orthogonal system of the screen. Position rotation is in a counter clockwise direction.

NUM - the number of characters to be generated.

JARRAY - an integer array that contains the ASCII codes of the characters to be generated, one character code per word.

DESCRIPTION: All characters are drawn in vector mode, starting at the current graphics position. Unlike WRITEX, however, WRITEY does not update the current position.

The only scaling factor that affects the character size for WRITEY is FACT. Character size is not influenced by the user's current SCALE.

The only rotation factor that affects WRITEY is DEG. The center of rotation will be the user's current position. This is dissimilar to the convention adopted

for WRITEX, where the center of rotation is the user's origin.

WRITEY may be called from any mode. The routine will draw characters as specified and then return the user to his current mode.

The WRITEY character set and character spacing conventions are identical to those established for WRITEX.

Example: The user wishes to use WRITEY to draw a rotated label such as the one that occurs in the frontispiece of this manual.

```
      PROGRAM LABEL
      DIMENSION JARRAY (9)
      CALL BEGIN
      CALL ERASE
      CALL TPLOT (512.,380.,0,0)
C     Establish current position at
C     center of the screen.
      JARRAY(1) = 32
      JARRAY(2) = 32
      JARRAY(3) = 84
      JARRAY(4) = 69
      JARRAY(5) = 75
      JARRAY(6) = 80
      JARRAY(7) = 76
      JARRAY(8) = 79
      JARRAY(9) = 84
      DO 100 I = 1,331,30
      DEG = I - 1
100   CALL WRITEY (2.,DEG,9,JARRAY)
C     WRITEY does not update the
C     current position.
      CALL HOME
      END
```

SUBRS CALLED: ALPHA, VECTOR, POINT, WRITEX, TEK001

NAME: XLATE

PURPOSE: XLATE converts the binary representation of a number to the equivalent ASCII representation of that number. The routine can be used to convert an integer variable to a form suitable for use with WRITEX or WRITEY.

USAGE: CALL XLATE (ICON, NUM, JARRAY)

PARAMETERS:

ICON - an integer variable or constant that is to be converted to its ASCII equivalent.

NUM - the number of significant digits, including the sign, in the translated result. NUM will be the number of characters that XLATE has stored in JARRAY.

JARRAY - an array where XLATE will store the translated result. JARRAY should be dimensioned large enough to contain one word for each decimal digit in ICON.

DESCRIPTION: XLATE suppresses leading zeros. If ICON is positive, the sign position, JARRAY(1), will be an ASCII space; if negative, the sign position will be an ASCII minus.

After calling XLATE, the assigned NUM and JARRAY parameters may be used as input parameters to WRITEX or WRITEY.

SUBRS CALLED: None

Internal Subroutines

Seven subroutines, TEK001 - TEK007, are used internally by the other routines described in previous paragraphs. These internal routines are not intended for external use by a user's program.

The internal routines include:

a. TEK001 - used to load from or store into the terminal status area. By using TEK001, the TEKPLOT package avoids cluttering the user's common area.

b. TEK002 - used to rotate and plot the user's current (x,y) points. If any point is off-screen, this subroutine computes the window intercept and plots to the window margin. If the vector passing from the last (x,y) to the current (x,y) does not pass through the window at all, TEK002 suppresses plotting.

c. TEK003 - used to partition an (x,y) point into high and low order bits, and tag bits, and send the result to the 4002A. If the high order bits have not changed since the last call to TEK003, then they are suppressed.

d. TEK004 - used to graphically input an (x,y) coordinate position together with a character.

e. TEK005 - used to generate a software character from a series of internally stored segments.

f. TEK006 - used to plot in incremental mode.

g. TEK007 - used to draw tic marks or grid lines for the AXIS routine.

Selected Examples

In this section several examples are presented. These are intended to clarify the use of the routines described in the preceding pages.

a. Example 1 uses TPLOT to draw a simple sine curve. Note the use of the loop variable to initialize the current graphics location.

b. Example 2 uses AXIS to draw two decades of a log axis. Then, for a fixed value of x, the log of points 1 through 9 and 10 through 100 are plotted.

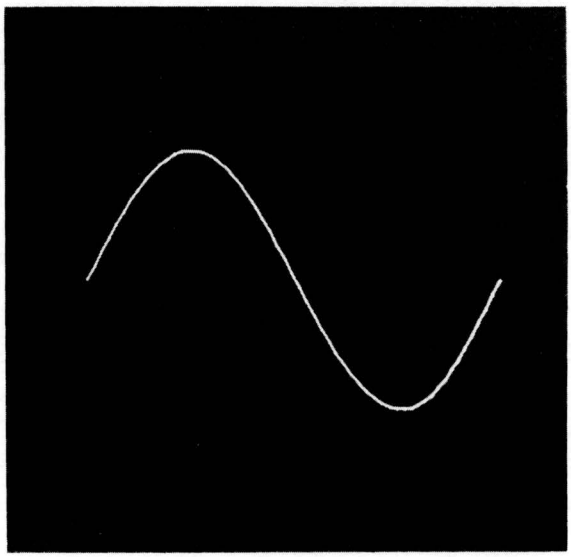
c. Example 3 uses AXIS to draw an axis. DELTA is then employed to label the axis using the hardware character generator.

d. Example 4 uses AXIS to draw the same axis that was specified in Example 3. AXIS labels in this example, however, are generated using the software character generator and XLATE.

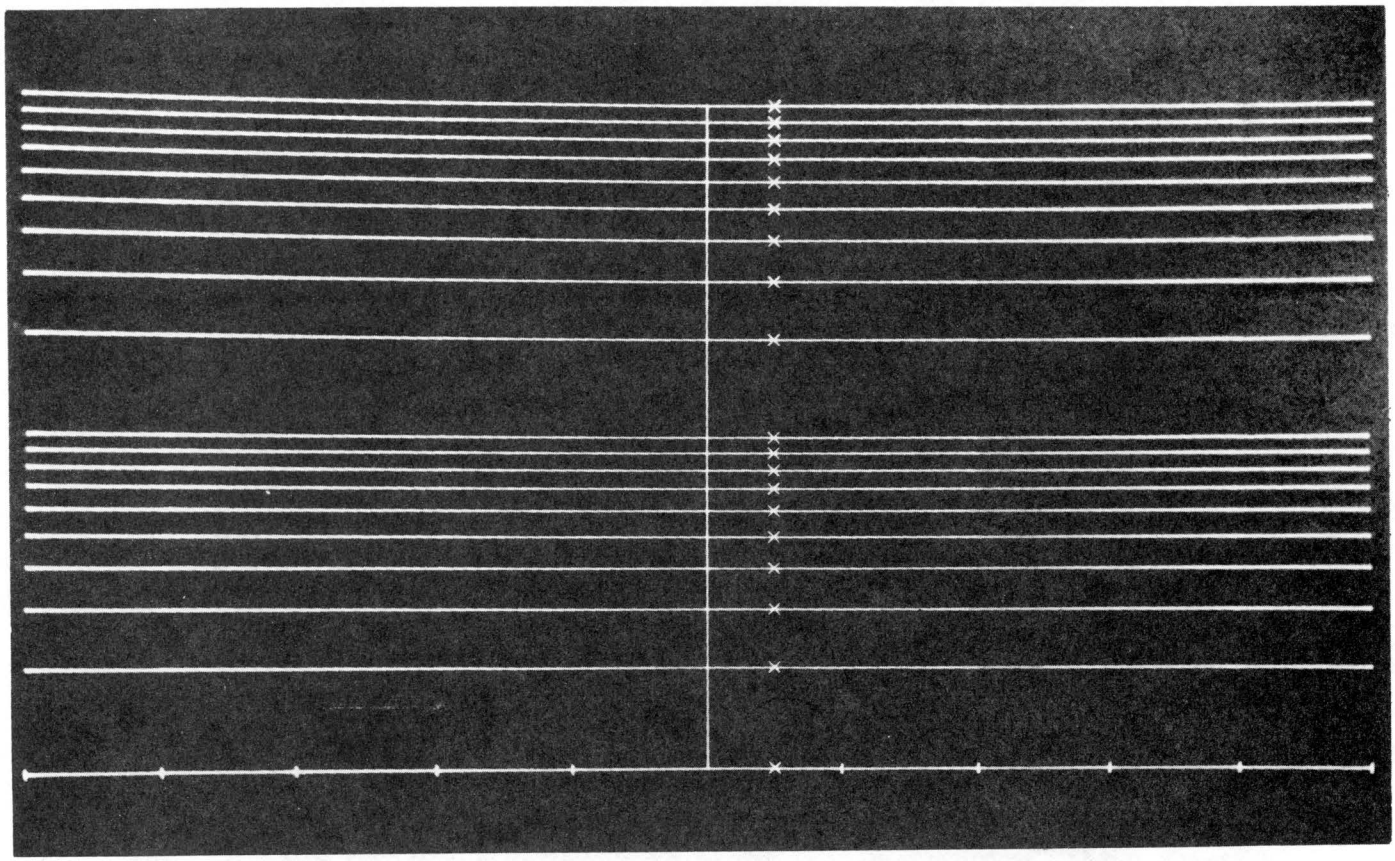
e. Example 5 shows how CURSIS may be used to graphically input a set of (x,y) points together with pen values from the 4002A. In this example the points are both plotted on the screen and saved in a file named REMEMBER.

f. Example 6 recalls the (x,y) points and pen values from a file named REMEMBER and plots these. The program then allows the use of graphics input to specify an area of interest. After defining such an area, the graph is replotted with a new origin and scale factors chosen such that the area of interest exactly fills the whole screen; the remainder of the plot is clipped at the screen margin. This process demonstrates the use of CURSIS to focus in on selected parts of a complicated plot.

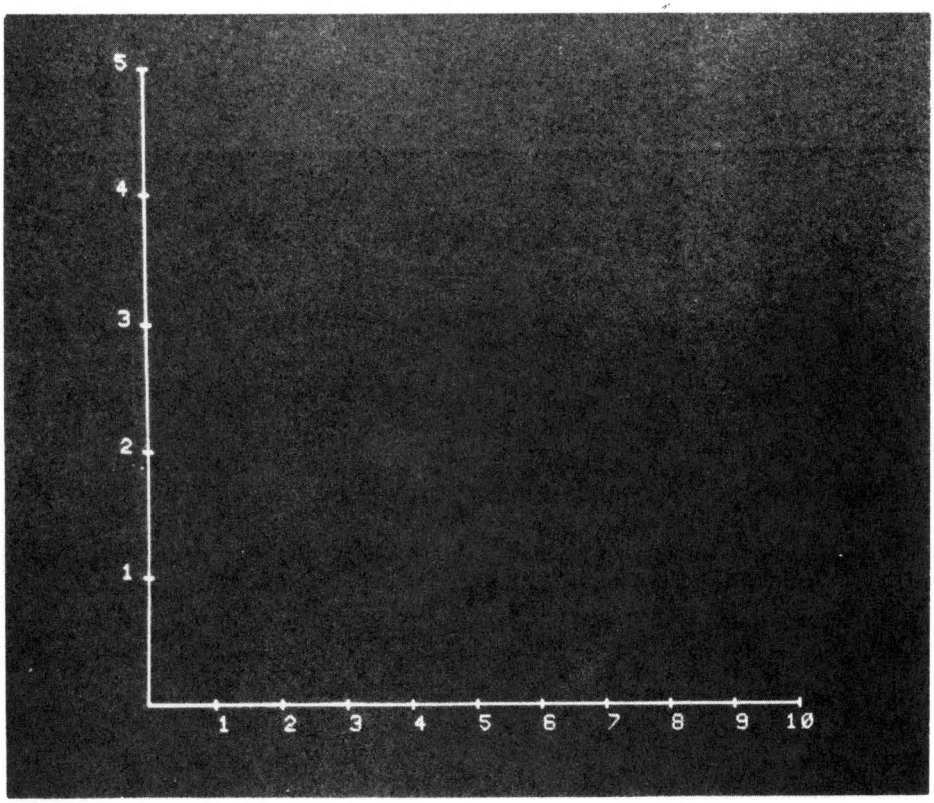
```
      PROGRAM SINE
C     PROGRAM TO PLOT A SINE CURVE
      CALL BEGIN(2400)
      CALL ERASE
      CALL VECTOR
      CALL SCALE(50.,100.,512.,380.)
C     SET ORIGIN TO MIDDLE OF THE SCREEN AND
C     SCALE Y FOR AMPLITUDE OF 100.
      DO 100 I=1,631,11
      IPEN=I-1
      X=I/100.
      Y=SIN(X)
100   CALL TPLOT(X,Y,IPEN,0)
C     NOTE THE TECHNIQUE OF USING THE LOOP VARIABLE, I,
C     TO INITIALIZE THE CURRENT GRAPHICS LOCATION FOR
C     THE FIRST VECTOR.
      CALL TPAUSE
      END
```



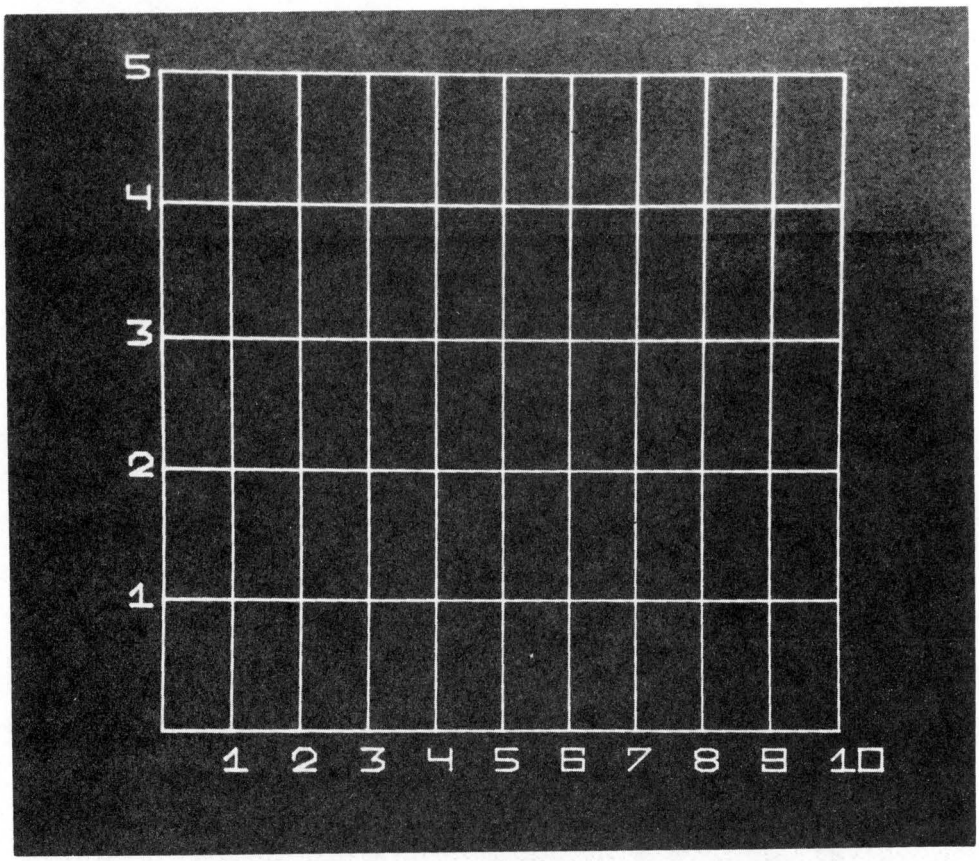
```
      PROGRAM LOGPLOT
      CALL BEGIN(2400)
      CALL SCALE(1.,250.,512.,100.)
C     SET SCALE FOR 250 TEKPOINTS/DECADE
C     ORIGIN TO X=512, Y=100
      CALL ERASE
      CALL AXIS(-500.,0.,1000.,2.,100.,-1.,1,-1)
C     X AXIS HAS TIC MARKS
C     Y AXIS IS A LOG AXIS, TWO DECADES WORTH
      CALL POINT
      X=50.
      DO 100 I=1,10
      Y=1
      Y=ALOG10(Y)
      CALL TPLOT(X,Y,L,L)
C     PLOT ON THE FIRST DECADE
100   CONTINUE
      DO 200 I=10,100,10
      Y=I
      Y=ALOG10(Y)
C     PLOT ON THE SECOND DECADE
      CALL TPLOT(X,Y,1,1)
200   CONTINUE
      CALL TPAUSE
      END
```



```
      PROGRAM LABEL1
C     PROGRAM TO LABEL AN AXIS.
C     USES THE ALPHA MODE OF THE 4002A.
      CALL BEGIN(2400)
      CALL ERASE
      CALL SCALE(10.,10.,200.,200.)
C     TEN TEKPOINTS/SCALED UNIT.
      CALL AXIS(0.,0.,50.,50.,5.,10.,1,1)
C     DRAW THE AXIS
C     REMEMBER THAT THE AXIS ROUTINE LEAVES THE
C     TERMINAL IN VECTOR MODE.
      CALL ALPHA
      CALL DELTA(-1.,-2.,0,0)
C     DELTA TO THE SOUTH OF THE AXIS.
C     NOW LABEL THE AXIS.
      DO 100 I=1,10
      CALL DELTA(5.,0.,0,0)
C     DELTA OVER TO THE NEXT TIC MARK ON THE X AXIS
100   WRITE(61,1) I
C     WRITE THE LABEL.
1     FORMAT(1X,I2)
C     NOTE THAT THE CARRIAGE RETURN AND LINE FEED
C     DO NOT AFFECT THE CURRENT GRAPHICS LOCATION
C     FOR THE NEXT DELTA.
C     NOW LABEL THE Y AXIS.
      CALL TPLOT(0.,0.,0,0)
C     PLOT BACK TO THE ORIGIN.
      CALL DELTA(-2.,0.,0,0)
C     DELTA TO THE WEST OF THE AXIS.
      DO 200 I=1,5
      CALL DELTA(0.,10.,0,0)
200   WRITE(61,2) I
2     FORMAT(1X,I1)
      CALL TPAUSE
      END
```



```
      PROGRAM LABEL2
C     PROGRAM TO LABEL AN AXIS.
C     LABEL2 USES THE SOFTWARE CHARACTER GENERATOR.
      DIMENSION ICHARS(8)
      CALL BEGIN(2400)
      CALL ERASE
      CALL SCALE(10.,10.,200.,200.)
C     10 TEKPOINTS/SCALED UNIT.
      CALL AXIS(0.,0.,50.,50.,5.,10.,-1,-1)
C     DRAW A GRID AXIS THIS TIME, JUST FOR VARIETY.
C     AXIS ROUTINE LEAVES THE TERMINAL IN VECTOR MODE,
C     POSITIONED AT THE ORIGIN.
      CALL ALPHA
      CALL DELTA(-2.5,-3.,0,0)
C     DELTA TO THE SOUTH OF THE AXIS.
      DO 100 I=1,10
      CALL DELTA(5.,0.,0,0)
C     DELTA OVER TO THE NEXT GRID LINE ON THE X AXIS.
      CALL XLATE(I,NUM,ICHARS)
C     TRANSLATE I TO ASCII.
100   CALL WRITEY(2.,0.,NUM,ICHARS)
C     WRITE THE LABEL.
C     NOW LABEL THE Y AXIS.
      CALL TPLOT(0.,0.,0,0)
C     PLOT BACK TO THE ORIGIN.
      CALL DELTA(-4.5.,-.5,0,0)
C     DELTA TO THE WEST OF THE Y AXIS.
      DO 200 I=1,5
      CALL DELTA(0.,10.,0,0)
      CALL XLATE(I,NUM,ICHARS)
200   CALL WRITEY)2.,0.,NUM,ICHARS)
      CALL TPAUSE
      END
```



```
      PROGRAM DIGITIZE
C     USE GRAPHICS INPUT TO DIGITIZE OR DRAW A PICTURE.
C     POINTS AND PEN VALUES ARE SAVED IN A FILE.
C     GRAPHICALLY INPUT POINTS ARE ACHOED BACK TO THE
C     4002A IN VECTOR MODE.
C        SPACE ECHOES AN INTENSIFIED VECTOR (PEN DOWN).
C        DEL ECHOES A DARK VECTOR (PEN UP).
C        BEL ENDS THE PROGRAM.
      DIMENSION XX(250),YY(250),JIPEN(250)
      CALL BEGIN(2400)
      CALL ERASE
C     THE NEXT STATEMENT EQUIPS LOGICAL UNIT 10 TO THE FILE
C     NAMED REMEMBER.
      CALL EQUIP(10,8HREMEMBER)
C     SAVE ALL POINTS IN A FILE NAMED REMEMBER.
C     REWIND 10
      JCT=0
      CALL SCALE(1.,1.,512.,380.)
      CALL VECTOR
100   CALL CURSIS(KHAR,X,Y)
C     INPUT A POINT.
      IF(KHAR.EQ.7) GO TO 200
C     END THE PROGRAM IF CHARACTER WAS A BELL
      IPEN=1
      IF(KHAR.EQ.127) IPEN=0
      CALL TPLOT(X,Y,IPEN,0)
C     ECHO THE INPUT BACK TO THE 4002A.
      JCT=JCT+1
      IF(JCT.GT.250) GOTO 200
      XX(JCT)=X
      YY(JCT)=Y
      JIPEN(JCT)=IPEN
      GO TO 100
200   DO 300 I=1,JCT
300   WRITE(10,400) XX(I),YY(I),JIPEN(I)
C     WRITE THE POINTS OUT ON THE FILE.
400   FORMAT(2(F8.2,2X),I1)
      END FILE 10
      CALL UNEQUIP(10)
      CALL HOME
      END
```

```
      PROGRAM MAGNIFY
C     PROGRAM TO READ A SET OF (X,Y) POINTS
C     FROM A FILE.
C     MAGNIFY PLOTS THE POINTS IN VECTOR MODE AND
C     THEN ALLOWS THE USER TO GRAPHICALLY DESIGNATE AN
C     AREA OF INTEREST. THE AREA OF INTEREST IS SPECIFIED
C     BY SURROUNDING AN AREA ON THE SCREEN WITH A GRAPHICALLY
C     INPUT RECTANGLE.  MAGNIFY THEN COMPUTES SCALE FACTORS
C     AND AN ORIGIN NECESSARY TO ENLARGE THE AREA OF INTEREST
C     TO THE SCREEN SIZE. THE POINTS ARE THEN REPLOTTED.
      DIMENSION XX(250),YY(250),JIPEN(250)
      CALL BEGIN(2400)
      CALL HOME
C     EQUIP THE FILE OF DIGITIZED POINTS.
      CALL EQUIP(10,8HREMEMBER)
      REWIND 10
      DO 100 I=1,250
      READ(10,1) X,Y,IPEN
1     FORMAT(2(FB.2,2X),I1)
      IF(EOF(10)) GO TO 200
      XX(I)=X
      YY(I)=Y
      JIPEN(I)=IPEN
100   JCT=I
200   CALL UNEQUIP(10)
500   XORG=512.
      YORG=380.
      XFACT=1.
      YFACT=1.
300   CALL SCALE(XFACT,YFACT,XORG,YORG)
      CALL ERASE
      CALL VECTOR
      DO 400 I=1,JCT
C     REPLOT THE POINTS.
400   CALL TPLOT(XX(I),YY(I),JIPEN(I),0)
C     NOW GRAPHICALLY INPUT THE AREA OF INTEREST.
      CALL CURSIS(K,XMIN,YMIN)
C     IF INPUT CHARACTER IS AN R, RETURN TO ORIGINAL SCALING.
      IF(K.EQ.82) GO TO 500
      CALL CURSIS(K,XMAX,YMAX)
      IF(K.EQ.82) GO TO 500
C     COMPUTE SCALING AND ORIGIN NECESSARY TO ENLARGE
C     THE AREA OF INTEREST TO THE SCREEN SIZE.
      XFACT=1023./ABS(XMAX-XMIN)
      YFACT=761./ABS(YMAX-YMIN)
      XORG=-XMIN*XFACT
      YORG=-YMIN*YFACT
      GO TO 300
      END
```