

A11105 973915 S

Reference

Publications



NBS TECHNICAL NOTE 1184

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

Utility Programs for Producing Camera Ready Illustrations on a Microcomputer and a Laboratory Plotter

QC
100
U5753
No.1184
1984

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards¹ was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities² — Radiation Research — Chemical Physics — Analytical Chemistry — Materials Science

THE NATIONAL ENGINEERING LABORATORY provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering² — Manufacturing Engineering — Building Technology — Fire Research — Chemical Engineering²

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

Programming Science and Technology — Computer Systems Engineering.

¹Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Washington, DC 20234.

²Some divisions within the center are located at Boulder, CO 80303.

Ref
QC
100
.U5753
No. 1184
1984

Utility Programs for Producing Camera Ready Illustrations on a Microcomputer and a Laboratory Plotter

C. E. Dick

Center for Radiation Research
National Measurement Laboratory
National Bureau of Standards
Washington, DC 20234



U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued January 1984

National Bureau of Standards Technical Note 1184
Natl. Bur. Stand. (U.S.), Tech. Note 1184, 59 pages (Jan. 1984)
CODEN: NBTNAE

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1984

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

TABLE OF CONTENTS

Abstract	1
Introduction	2
Hardware	3
Software	3
Plotter Geometry and Variables	5
Running Draftsman	6
Making a Drawing	8
Sample Illustrations	9
Conclusion	12
Appendix A	13
References	16
Table 1: Main DRAFTSMAN Program Segments	17
Table 2: Instruction Files	18
Table 3: DRAFTSMAN/BPLOT Correspondence	20
Figure Captions	21
Listings	29

Utility Programs for Producing Camera Ready Illustrations
on
A Microcomputer and a Laboratory Plotter

C. E. Dick
National Bureau of Standards
Washington, DC 20234

ABSTRACT

A collection of software routines is described that allows the user to prepare camera ready illustrations in the laboratory or office environment. These routines are written in APPLESOFT BASIC and 6502 assembly code for the Apple II microcomputer which is interfaced with an inexpensive digital plotter. Provisions are made to draw figures composed of straight and curved line segments, letter the figures with a variety of graphic arts fonts, and save the figures on disk for later plotting or revision.

Key words: Apple II computer; camera ready illustrations; digital plotter; graphic arts; microcomputer.

INTRODUCTION

In the last few years, microcomputers capable of controlling inexpensive laboratory plotters have made possible the production of camera ready illustrations in the laboratory and office environment. Such a system has several inherent advantages; it provides the author with rapidly generated illustrations at low cost; it permits updating of drawings and figures as conditions warrant; and it eliminates costly and time consuming drafting and hand lettering by graphics artists.

The ingredients of such an in-house facility, in addition to the microcomputer and plotter are:

- a) a collection of mechanical drafting routines, and
- b) a repertory of graphic arts fonts in digitized form.

The second of these requirements, a repertory of graphic arts fonts, has been discussed in a previous report (1) which describes the storage and retrieval of the Hershey fonts (2,3) of occidental characters on the Apple II^{*} computer system operating under DOS 3.3 (4). The first item, a collection of drafting routines, is the subject of this report which describes a program named DRAFTSMAN which is written in Applesoft Basic (5). This program provides the capability of producing illustrations complete with text on a digital plotter, editing them if necessary, and storing them on floppy disk for later recall. In addition, an assembly language program called BPLOTT^{*} is provided to interface the Apple II to a Hiplot (6) digital plotter. Although this program is designed to drive a Hiplot plotter, it can be modified if necessary to interface with other types of plotters. This collection of programs is designed to operate on an Apple II computer equipped with at least 48K of memory. The following sections describe the system in detail and include examples of the types of illustrations that can be produced using this system.

* In this report, certain commercial products are referenced by name. These references are for informational purposes only, and do not imply that these products are necessarily the best for the purpose and do not imply endorsement by NBS.

HARDWARE

As presently implemented in our laboratory, the system operates on an Apple II computer system which is interfaced with a Houston Instrument Hiplot plotter (Model DMP-2) through an Apple RS-232 serial interface (7) operating at 9600 baud. This plotter has a useable plotting area of 184 x 260 mm (7.25 x 10.25 in.) with resolution steps of 0.127 mm (0.005 in.) in both the horizontal and vertical directions. The movement of the pen is controlled by the series of ASCII characters p..w, which move the pen in the directions shown in Fig. 1, and the characters y,z which raise or lower the pen respectively. (Parenthetically, all figures in this report were drawn on this system.) For example, outputting the character string "zpppqrrry" to the plotter will lower the pen, advance the carriage 3 units up, 1 unit at an angle of $+45^{\circ}$, 3 units to the right, and finally, raise the pen. Although the interface program named BPLOT described in the Software Section was written to generate these specific ASCII characters for this plotter, this program was written as an assembly language routine not only for execution speed but so it could be modified to support other plotters if desired.

The Apple computer is a standard model Apple II or Apple II+ with at least 48 kbytes of RAM. The serial I/O interface card is installed in expansion slot 2 although any slot other than 0 could be used with an appropriate modification of the driver software of BPLOT. In addition, to facilitate the editing process, we have a printer interfaced to the computer in slot #1 although this is not necessary for the execution of the programs.

SOFTWARE

The software package consists of several parts as mentioned above. Listing 1 gives a catalog of the diskette containing this system. As can be seen from an examination of this listing, the primary routines for generating the illustrations are contained in an Applesoft basic program called DRAFTSMAN which is the greeting

program for the diskette. In addition, there are twenty two text files which contain the digitized Hershey fonts (1), the binary file RPLLOT which contains the output driver, and five binary files labeled INSTR.1 to INSTR.5 which are used to supply instructions to users of the system but are normally not needed to execute the programs. These files occupy 263 sectors of the diskette leaving approximately 200 sectors available for storing illustrations produced using the system.

The Applesoft code of DRAFTSMAN is given in Listing 2. A Pascal extension of this program which includes routines to input data from peripheral devices, edit the data, and plot it is currently being debugged and documented (8). The variables used in DRAFTSMAN are documented in Listing 3 which was produced by the variable documentation program of Wagner (9). Reference to this listing will be helpful to understand the program structure of DRAFTSMAN.

As can be seen from an examination of the program, DRAFTSMAN is composed of a number of inter-related segments. The principal segments and their corresponding line numbers are given in Table 1. In addition, in Table 1, a name is listed for each segment of the program. Although these names are meant only for reference in the discussions that follow, some of them are used as procedure names in the forthcoming Pascal version (8).

In addition to the DRAFTSMAN program, the binary program BPLLOT is needed to interface between DRAFTSMAN and the Hiplot plotter. An assembled listing of this program as compiled with the Microproducts assembler (10) is discussed in some detail in Appendix A. This program uses double precision values for the x- and y-coordinates to calculate the output string of ASCII characters which are stored in a character table beginning at memory location 34304 (\$8600). A command to output this character string results in the sequential outputting of these characters using the ROM based software supplied as part of the Apple serial I/O card located in slot #2 (7). (A CALL to the location \$C200

results in the outputting on the RS-232 line of whatever byte is contained in the Apple's 6502 microprocessor's accumulator.)

Finally, the remaining files are used to supply the appropriate Hershey fonts (1,2) to the program or to provide the user with directions on how to communicate with the program. The uses of these files will be more fully documented in the sections which describe the execution of the program.

PLOTTER GEOMETRY AND VARIABLES

In order to understand the operation of the DRAFTSMAN program it is necessary to be familiar with the coordinates of the plotter system. Figure 2 illustrates these coordinates and some of the variables used in the software programs. The origin of the coordinate system is in the lower left hand corner (0,0) and the x- and y-axis form an orthogonal coordinate system in the usual sense. Since the plotter has a resolution of 200 steps per inch, the variables XD and YD are used to record the current pen position from the origin. Thus in Fig. 2, for example, the point (3,2) which is 3 inches from the origin in the x-direction and 2 inches in the y-direction, has an XD value of 600 (3×200) and a YD value of 400 (2×200). For convenience in returning the pen to known locations in the plot, the program allows for the definition of any point as a reference origin. These reference points may be stacked up to 9 levels deep and the variable R keeps track of this stacking. The arrays H(R) and V(R) are used to store the offset of the last defined reference location from the one preceding it. Thus in Fig.2, if the point (3,2) is defined as the current origin, the value of R is incremented ($R=1$) and the offsets to the previous origin (0,0: $R=0$) are stored as $H(0)=600$ and $V(0)=400$. If the pen is subsequently moved to a new location, (6,3), the variables XD and YD contain the plotter offset to the last defined origin as shown on the figure. The variables LX and LY are used to store the total offset of the current pen location from the lower left corner of the plot, (0,0), and are used to keep the pen from overflowing the plotter limits. Since the useful plotting

area of this particular plotter is 10.25 in. horizontally by 7.25 in. vertically, the program ensures that the values of LX and LY are in the ranges from 0-2050 and 0-1450 respectively. Finally, for convenience, the program also recognizes the directions L,R,U, and D as shown in Fig.2.

RUNNING DRAFTSMAN

The DRAFTSMAN program is executed with the usual Applesoft RUN command or by booting the system with the diskette in the boot disk drive. After the presentation of the logo, the user is asked whether directions are needed. If the directions are requested, the program sequentially loads the binary files INSTR.1 to INSTR.5 into the portion of memory reserved for the first page of text (\$0400-\$07FF) where they are automatically displayed by the monitor. Table 2 gives a listing of the instructions that are displayed during this process. As can be seen from this table, the program is designed to recognize mnemonic commands issued by the user to move the pen on the plotter or to perform certain disk or utility commands. These instructions will be covered in detail in the following section.

After the direction cycle is completed, the binary output driver is loaded by the program segment Bload and variables are assigned to allow the program to communicate with this I/O and vector generator routine. In addition, the user is requested to position the plotter carriage in the lower left corner (the location 0,0 of Fig.2). The program is designed so that all variables are initialized with this location as a starting point. Finally, the user is requested to input the drive number of the diskette which will be used to either store or retrieve drawings.

The Initialize section of the program completes the remaining variable initialization and inquires whether the drawing has been previously saved on the diskette. If the drawing is on diskette, the Load section of the program inputs the text file containing the instruction set describing that drawing and the values of the variables shown in Fig. 2 at the time the drawing was saved. At

the end of the load, the user has the option of specifying whether the drawing is a new print <N> or a continuation of an old print <C>. If the new print option is chosen, the previously saved drawing will be redrawn up to the point where it had been saved and control will be returned to the user. If the continuation option is chosen, the pen can be placed in the appropriate location and control is immediately returned to the user to append further instructions. If the drawing is not one that has previously been saved, control is immediately returned to the user for the creation of a new drawing.

The Input section of the DRAFTSMAN program is the section in which the user enters the actual instructions to be executed. These instructions are input as text strings into an array named B\$(J) which has a maximum size of 475 elements. In the rare case where a drawing might take more than 475 instructions to specify, it can be broken down into two or more parts which can be sequentially plotted using the continuation option described above. Instructions are input in response to the prompt "INSTRUCTION NO. <J>" and are terminated by a carriage return.

Upon termination of the input instruction, the instruction string is decoded and the appropriate flags are set to indicate the operation to be performed. These flags are decoded by the Decode section and branches are made to various subroutines (Fonts, Text, Circle, X-Y etc.) to calculate the values of the x and y offset coordinates and the pen position (up,down). These values are then passed to the Out section which calls the appropriate routines in BPLOT to calculate and output the appropriate ASCII text string to the plotter. After the current instruction is decoded and executed, the next instruction is requested and the process begins again.

The program is halted through the Exit routine which is called by the input instruction "Q" and which then gives the user the option to save the current drawing. The remaining sections are transparent to the user and are used to handle DOS errors

(Error), check for input syntax errors (Syntax), or to edit the instructions (Edit).

MAKING A DRAWING

As described above, instructions are input in the form of easy to remember mnemonics which are interpreted by the program to perform the desired functions. These mnemonics, listed in Table 2, can be divided into the following categories:

- a) Instructions which change the pen position or move the carriage on the plotter. These instructions are those listed under the MULTIPLE COMMAND category of Table 2. In general, these commands may be combined to provide multiple instructions in one line. For example the command PD3.1CR is interpreted to move the carriage 3.1 centimeters to the right with the pen down (i.e. a 3.1 cm long line will be drawn on the paper.) The DIRECT MOVEMENT command X#Y# is also of this type and is listed in a separate category in the table for emphasis since it is frequently used. It should also be noted that the pen may also be moved at an angle for a given distance by the A## command when combined with a distance as in PU3IA45 which will move the carriage 3 inches at an angle of $+45^{\circ}$ to the positive x-axis with the pen up. Special instructions for the circle command, @, are listed in Table 2 and allow the drawing of circles, ellipses, and arcs of circles or ellipses. It is also not necessary to specify the scale factor, I or C, in each command as this factor will remain at its last specified value unless explicitly changed. The default value for this parameter is inches. In the same fashion the pen is also left in its last position when drawing lines unless explicitly changed. The pen default is "pen up", and the pen returns to this value at the completion of the arrow, circle, erase, and home commands.
- b) Instructions which are used to perform specific actions and are listed in Table 2 under CONTROL COMMANDS. These commands are entered as listed in the Table except for the dotted line

command, *, which may be combined with an optional dot length and a command from the commands above. (For example, the instruction .05*IX2Y1 will draw a dotted line from the current carriage position to the point 3 inches to the right and 1 inch up with .05 inch long dots and spaces.) The font command is used to enter the Hershey fonts from the diskette and the text and size commands are used to enter text or compute the length of a text string. Table 2 has information on the use of these commands. Figure 3 illustrates the Hershey characters that are currently available to the program. Finally, the erase command, E, will remove the last instruction input from the command sequence and reverse the pen movements associated with the deletion. If this command was one which plotted a line with the pen down however, that line will still be on the paper and must be manually erased or the figure redrawn.

- c) Finally, there are the utility commands, ?, V, and Q. The ? command results in the display of the instruction files INSTR.1 through INSTR.5 on the monitor. The verify command, V, allows the user to print out all the instructions entered prior to its execution on either the monitor or a printer and edit them for errors if necessary. Finally, the quit command, Q, terminates the input phase and allows for the saving on diskette of the instruction set for the drawing. None of these commands is saved as a permanent part of the instruction array and are thus not part of the permanent record of the drawing.

SAMPLE ILLUSTRATIONS

In order to understand the procedures associated with the production of a finished drawing, this section will present some examples of illustrations made using the system. Figure 4 depicts a stylized drawing of a soup can produced with the DRAFTSMAN program. The instruction set given in Listing 4 was used to generate this drawing. As can be seen from this listing, a series of 23 instructions was used for this particular figure. With the

pen in the initial position, the lower left corner of the plotting area, Instructions 0 and 1 define this location as the origin. The next instruction requests the input of the character fonts designated in Instructions 3 - 7. These fonts are thus made available for the input of text to the program and are taken in this particular case from the Index fonts of Fig.3. After the fonts are loaded, the carriage is moved with the pen up 3 inches in both the x and y directions (#8). This point is then designated as a new origin (#9) and the carriage is then moved up 2 inches (#10). With this point as the center, Instruction #11 is used to draw an ellipse with a major axis of 2 inches and a minor axis of 0.7 inches to represent the top of the can. At the completion of the ellipse, the pen is left in the up position, and Instruction No. 12 moves the pen to the right hand edge of the ellipse. From this point a 3 inch line is drawn to represent the side of the can (#13). Notice that a pen position command is necessary in this instruction to change the status of the pen. Instruction No. 14 moves the carriage with the pen up to the other side of the can and that side is subsequently drawn (#15). The H(ome) instruction then moves the carriage to the last defined origin which is on the centerline of the can and 1 inch above its base. This instruction has the effect of decrementing the arrays that refer to the origin so that the next home command will refer back to the point at the origin of the plotter (the lower left hand corner). Instruction No. 17 moves the carriage to the center of the can's base. From this point, the front edge of the can is drawn (#18) as a solid line, and the hidden back edge is drawn as a dotted line (#19). The carriage is then moved to the center of the can (#20). At this point in the input sequence, a size command was used to determine the length of the text string that was to be entered here. This command does not appear in the stored instruction sequence but it returned the value 1.24 inches for this length. To center the label, the pen is then moved .62 inches left (#21) and the text is entered. Stored text commands always begin with W to indicate that Words are to be plotted and

the last two numbers indicate the size and direction to plot the text. In this case the size is 4 and the direction is 0 (horizontal). Vertical characters are indicated by a 6 in the last character. The figure is now complete and the carriage is returned to the origin of the plotter by the H(ome) command(#23).

This drawing has several errors, the dotted ellipse representing the bottom of the can (#19) has the wrong minor axis, and the label reads SOAP instead of SOUP (#22). Figure 4b shows the can plotted correctly with the errors corrected by use of the V(erify) utility. The instructions for this print are given in Listing 5. Notice that Instruction No. 8 has also been changed to start the figure at an x value of 7 inches so that both figures could be drawn on the same print. Also, a number of Instructions have been added to draw a border around the figures and label the two versions as (a) and (b).

Figures 5 thru 7 illustrate some drawings that have been produced using these programs. Figure 5 illustrates a drawing produced for a machinist to fabricate an adapter flange for a vacuum system used in our laboratory. Of particular interest is the use of the >(arrow) command to draw arrows on some of the lines. This command is designed to draw an arrow with a specified size either in the direction of last carriage movement (>) or antiparallel to that direction (<). Figure 6 shows a pie chart which details the usage of one of the accelerators in our laboratory. The draw at an angle command (A##) was used in this print to divide the circle into the pie shaped segments. Finally, Figure 7 depicts a schematic of the safety interlock system for an electron accelerator. This figure utilizes essentially the entire gamut of instructions available to produce a schematic electrical circuit drawing which can be saved on diskette as a permanent part of the documentation for the accelerator. These figures are meant to be illustrative of some of the capabilities of the system. Ultimately, the uses are limited by the imagination of the user.

CONCLUSION

We have developed a software system that allows the Apple II computer system to produce illustrations on an inexpensive digital plotter. This system can be used to produce camera ready figures for inclusion in technical reports, for documentation, or for preparing vue-graphs or machinists prints. The figures can be constructed of straight line segments or portions of ellipses or circles and can be labeled with any of more than 700 characters and graphic symbols. The system can be improved in several aspects by rewriting some of the routines in assembly language for faster execution, particularly the segments used to generate circles and ellipses. The interface driver to the plotter is written in 6502 assembly code and can be modified to interface to a variety of laboratory plotters. At present, a Pascal version of this system is being incorporated into a multisegmented program designed not only to provide drafting capabilities but data input and manipulation as well.

APPENDIX A

The Apple II computer and the Hiplot plotter are interfaced through the binary I/O program BPLOT. This program is designed to take the values of the x and y offsets and the proper pen position (up or down) generated by DRAFTSMAN and generate the appropriate series of ASCII characters to control the plotter as shown in Fig. 1. Although these functions could be written in Applesoft, Basic routines to generate these characters are prohibitively slow, so the driver was written in 6502 assembly code. This choice has the added advantage of permitting easier modification to interface with different plotters without disturbing the fundamental DRAFTSMAN program. Listing 6 gives a compiled version of BPLOT using the Microproducts Assembler/Editor (10).

The DRAFTSMAN program interacts with the output driver BPLOT through the series of variables and subroutines listed in Table 3. The DRAFTSMAN program calculates the x and y offsets to the next point to be plotted and converts them to double precision binary numbers which have the high and low bytes HL, HH, VL, and VH which are poked into the proper locations in BPLOT. The variable PC is used to indicate the pen status, ANGLE specifies the angular rotation (0=hor., 6=vert.), TIME specifies the delay between successive character outputs, XVEC and YVEC are used to input the Hershey character vectors, and SIZE may be used to specify the character size.

The program, BPLOT, is composed of a number of segments that generate a table of ASCII characters to control the Hiplot plotter, and output these characters to the plotter. The main program, DRAFTSMAN, calculates the offset from the current location to the next point as double precision (16 bit) binary numbers which are poked into the locations \$85D6-\$85D9 as shown in Listing 7. In the case of the Hershey characters, the X and Y-offsets are poked into the single precision locations \$8418 and \$842C. In this latter case, a subsequent CALL LGEN (line 8720) converts these single precision values to double precision and stores them in the appropriate locations. In both cases, the

Vector Generator routine is called (either directly from line 7140 or as an extension of LGEN) and a table of ASCII characters is generated and stored in a table beginning at location \$8600. This table is generated by the assembly language routine beginning at location \$843F and is modeled after the Basic language routine given in Listing 7. This routine which is included in the Hiplot instruction manual (6) calculates the best straight line between the current pen location and the location specified by the X- and Y-offsets. Listing 7 indicates the correspondence between the variables used in the Basic program and the variables and locations of the BPLLOT vector generator routine. The output table begins with the pen control character ("y" or "z" to indicate pen up or down), and contains a series of characters from the ASCII set "p".."w" to control the motion of the carriage. The table ends with the ASCII character "x" to signify the end of the vector. This routine makes use of standard double precision routines to ADD (\$8599), SUBTRACT(\$85BC), and MOVE(\$85C7) double precision binary numbers.

Once the table corresponding to the plotter movement is generated, the PLOT routine (\$83B8) outputs one character at a time to the plotter through the RS-232 interface installed in slot #2. Each character is output the number of times specified by the parameter SIZE with a selectable delay between successive characters. The table is output character by character until the occurrence of the character "x" which signifies the end of the table. At this point, the output is terminated, various locations are restored to their initial values, and control is returned to the calling program.

To modify this program for plotters other than the Hiplot plotter, it is necessary to replace the vector generator subroutine with a subroutine that will generate the appropriate characters for the plotter of interest. In addition, the BPLLOT program contains two routines to accept a single character (CHAR) and output it via the RS-232 serial port (ALPHA). Although these

locations are not used in DRAFTSMAN, they are available for other programs which might incorporate the BLOT program.

REFERENCES

1. C.E. Dick and J. Hilsenrath, "Utility Programs for Generating the Hershey Character Fonts on Microcomputers and Laboratory Plotters," NBS Technical Note 1176, June 1983, National Bureau of Standards, Washington, DC, 20234.
2. A.V. Hershey, "Calligraphy for Computers," NWL Technical Report No. 2101, August 1967, U.S. Naval Surface Weapons Laboratory, Dahlgren, VA.
3. N.M. Wolcott and J. Hilsenrath, "A Contribution to Computer Typesetting Techniques: Tables of Coordinates for Hershey's Repertory of Occidental Type Fonts and Graphics Symbols," NBS Special Publication No. 424, National Bureau of Standards, Washington, DC, 20234.
4. "DOS 3.3", copyright 1980, Apple Computer Inc., 10260 Bandley Drive, Cupertino, CA, 95014. Apple software product #A2L0036.
5. "Applesoft", copyright 1978, Apple Computer Inc., 10260 Bandley Drive, Cupertino, CA, 95014. Apple software product #A2L0006.
6. The plotter utilized is a Hiplot Model DMP-2 Digital Plotter, manufactured by the Houston Instrument Co., One Houston Square, Austin, TX, 78753.
7. Serial Interface Card, Apple Computer Inc., 10260 Bandley Drive, Cupertino, CA, 95014. Apple hardware product #A2L0008.
8. C.E. Dick, A Pascal language program for data retrieval, storage, reduction, and plotting on a microcomputer and laboratory plotter." In preparation.
9. "Apple-doc", software copyright 1979, Roger Wagner, Southwestern Data Systems, Box 982, Santee, CA, 92071.
10. "6 Character Label Editor/Assembler", Version 6.1, software copyright 1979, Microproducts Inc., 2107 Artesia Blvd., Redondo Beach, CA, 90278.

Table 1. A listing of the main program segments and the corresponding line numbers for DRAFTSMAN.

Name	Lines	Program Function
Logo	10-180	Initialization, sets HIMEM, defines the DOS token (D\$), and prints the logo.
Instr.	500-560	Retrieves the user instructions from binary files INSTR.1 to INSTR.5 and displays them.
Bload	1000-1100	BLOADS BPLOT, initializes variables pointing to locations in BPLOT, and requests disk #.
Init.	2000-2200	Main program start and variable init.
Input	3000-3390	Sets save flag (SF) if input is from disk. Main instruction input routine, requests input from keyboard, decodes it and sets appropriate plotting flags.
Decode	3500-3730	Decodes flags and instructions given in the instruction input loop at 3000-.
Exit	4000-4060	Exit routines, saves appropriate variables to prepare to save illustration on diskette.
Fonts	5000-5350	Inputs Hershey fonts from appropriate files.
Load	6000-6180	Inputs saved drawing from disk.
Save	6500-6610	Saves drawing to text file on disk.
Out	7000-7190	Output vector generator, calls appropriate routines in BPLOT to generate and output ASCII character string to the plotter.
Text	8000-8190	Inputs text string to be plotted on plotter.
Hershey	8200-8920	Decodes text string, calculates coordinates for Hershey characters, and outputs them to plotter.
Display	9000-9070	Displays text characters as decoded.
X-Y	10000-10050	Decodes XY input instructions.
Rho	11000-11030	Calculates rho values.
Error	12000-12120	Error handling routines.
Circle	13000-13180	Decodes circle, ellipse, and arc inputs.
Dot	15000-15070	Decodes dotted line output instructions.
Edit	16000-16300	Instruction editing and printing routines.
Syntax	20000-20030	Checks input for illegal characters.

Table 2. A listing of the instructions input to the user from the files INSTR.1 through INSTR.5.

DRAFTSMAN INSTRUCTIONS

COMMANDS ARE GIVEN BY TYPING IN KEY LETTERS THAT THE INTERPRETER CAN TRANSLATE TO MOVE THE PEN ON THE HILOT PLOTTER. DIMENSIONS CAN BE SPECIFIED EITHER IN INCHES OR IN CENTIMETERS. COMMANDS OTHER THAN DISK COMMANDS MAY BE COMBINED IN ANY ORDER IN ONE STATEMENT SUCH AS PU3.5IA30 WHICH PARSES AS A PEN UP MOVE OF 3.5 INCHES AT AN ANGLE OF 30 DEG. THIS MAY ALSO BE WRITTEN A30PU3.5I ETC. TO REVIEW COMMANDS ENTER <?> IN RESPONSE TO PROMPT. TO EDIT COMMANDS ENTER <V> IN RESPONSE TO PROMPT.

KEY	COMMAND	EXAMPLE
MULTIPLE COMMANDS		
A##	MOVE AT ANGLE ##	A45PU3I
@##	DRAW CIRCLE W//RAD.##	@3.1C
C,I	CENTIMETERS, INCHES	PU3CR
PU,PD	PEN CONTROL UP/DOWN	PD2CD
R,L	MOVE RIGHT, LEFT	PU1IL
U,D	MOVE UP, DOWN	PD2CD
>	DRAW ARROW	.5I>
CONTROL COMMANDS		
*	DRAW DOTTED LINE	.05*1IR
E	ERASE LAST LINE	E
O	SET CURRENT LOCATION AS ORIGIN	O
H	RETURN TO LATEST ORIGIN	H
T	INSERT TEXT	T
PA#	CHANGE PEN SIZE (PAUSE)	PA2
F	CHOOSE LETTER FONT	F
S	CALC. LENGTH OF TEXT	S
DIRECT MOVEMENT		
X#Y#	MOVE IN STRAIGHT LINE TO POINT X+DX, Y+DY	TX3Y4.5

SPECIAL INSTRUCTIONS FOR F(ONT),P(AUSE),T(EXT) AND S(IZE) COMMANDS.

THE F COMMAND ALLOWS THE CURRENT FONT TO BE CHANGED. FOLLOW THE PROMPTS GIVEN.

THE PA COMMAND GIVES A BUILT IN PAUSE TO ALLOW THE USER TO CHANGE THE PEN OR FOR ANY OTHER REASON.

TO ENTER TEXT VIA THE T COMMAND, THE DEFAULT MODE IS ALL CAPS, WHICH ARE SHOWN IN INVERSE VIDEO. TO CHANGE TO LOWER CASE AND VICE VERSA, USE THE <ESC> KEY AS A TOGGLE. THE MODE REMAINS THE SAME UNTIL TOGGLED AGAIN. GREEK LETTERS AND MATH SYMBOLS ARE ENTERED BY PRECEDING THE APPROPRIATE KEY WITH EITHER A <CTRL-G> OR A <CTRL-S> AND ARE SHOWN FLASHING.

THE S COMMAND CALCULATES THE LENGTH OF A TEXT STRING FOR CENTERING PURPOSES.

SPECIAL INSTRUCTIONS FOR THE @(CIRCLE) COMMAND. THE @ COMMAND WILL DRAW CIRCLES OF A GIVEN RADIUS OR ELLIPSES WITH SPECIFIED MAJOR AND MINOR AXES.

TO DRAW A CIRCLE WITH A GIVEN RADIUS, SIMPLY SPECIFY THE RADIUS AND THE @ SYMBOL: E.G. @2 OR 2@ WILL DRAW A 4 IN. DIAMETER CIRCLE.

TO DRAW AN ELLIPSE, SPECIFY THE MAJOR AND MINOR AXES SEPARATED BY THE @ SYMBOL: E.G. @2@1 OR 2@1@ WILL DRAW AN ELLIPSE WITH A 4 IN. MAJOR AXIS AND A 2 IN. MINOR AXIS.

TO DRAW AN ARC OF A CIRCLE, SPECIFY THE ANGULAR RANGE SEPARATED THE ^ SYMBOL: E.G. 2@^30^120 OR @2^30^120 WILL DRAW A 2 IN. ARC BETWEEN 30 AND 120 DEGREES.

SPECIAL INSTRUCTIONS FOR *(DOTTED LINE) COMMAND.

ANY LINE CAN BE DOTTED BY INSERTING THE * CHARACTER AND THE DOT LENGTH IN THE COMMAND STRING. FOR EXAMPLE, THE COMMAND .05*1R OR *.05R1 WILL DRAW A DOTTED LINE TO THE RIGHT WITH .05 IN. DOT UNITS.

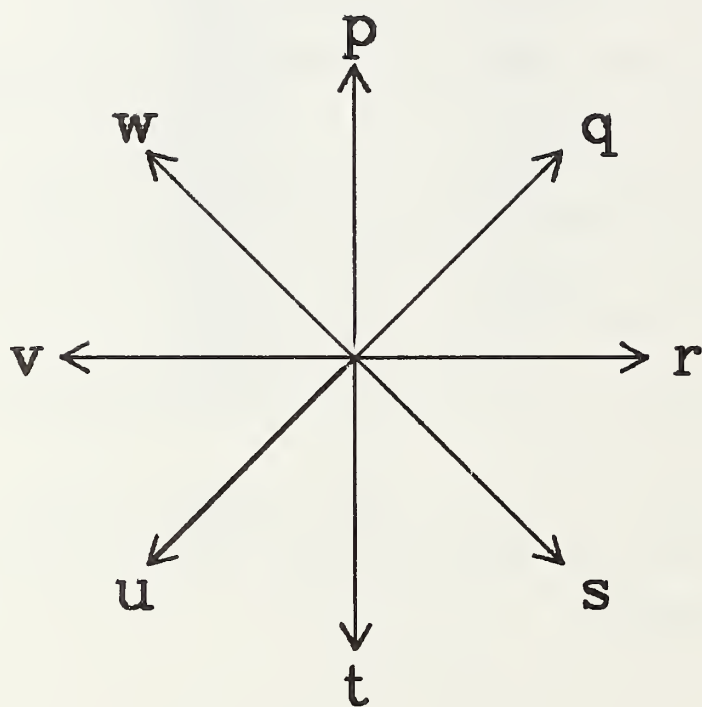
CAUTION!! CIRCLES AND ELLIPSES MAY ALSO BE DOTTED, BUT THE DOT LENGTH HAS NO MEANING AND YOU MAY GET A DIV BY ZERO ERROR. TO AVOID THIS, JUST USE THE DOT SYMBOL <*>, FOLLOWED BY THE CIRCLE SYMBOL <@> AND THE RADIUS SUCH AS *@.5. SIMILARLY, IN THE CASE OF ELLIPSES AND ARCS, USE *@1@2 AND *@.5^30^120.

Table 3. Correspondence of locations between DRAFTSMAN and BLOT.

DRAFTSMAN VARIABLE	LOCATION decimal	BLOT VARIABLE	LOCATION hexadecimal
OUT	33720	PLOT	\$83B8
ALPHA	33793	ONECHR	\$8401
SIZE	33801	SIZE	\$8409
ANGLE	33802	ANGLE	\$840A
CHAR	33803	CHAR	\$840B
TIME	33804	DELAY	\$840C
LGEN	33805	LETTER	\$840D
PC	33811	PEN	\$8412
XVEC	33816	XVEC	\$8418
YVEC	33836	YVEC	\$842C
VGEN	33855	GO	\$843F
VL	34262	INPUT	\$85D6
VH	34263	YH	\$85D7
HL	34264	XL	\$85D8
HH	34265	XH	\$85D9

FIGURE CAPTIONS

- Figure 1. A schematic illustration of the carriage movement and pen position control generated on the Hiplot plotter upon receipt of the ASCII characters shown.
- Figure 2. A representation of the geometry and variables used to define the plotter carriage location.
- Figure 3. The Hershey characters available to the DRAFTSMAN program from the 22 text files on the diskette.
- Figure 4. A stylized drawing of a soup can. Figure 4a was drawn with the instruction set of Listing 4 and contains several errors. In Fig. 4b, the errors have been corrected as given in Listing 5.
- Figure 5. An example of a vacuum adapter flange produced by the software system.
- Figure 6. An example of a pie chart produced by the system.
- Figure 7. An example of an electrical wiring diagram produced by the system.



y : pen up
z : pen down

FIGURE 1.

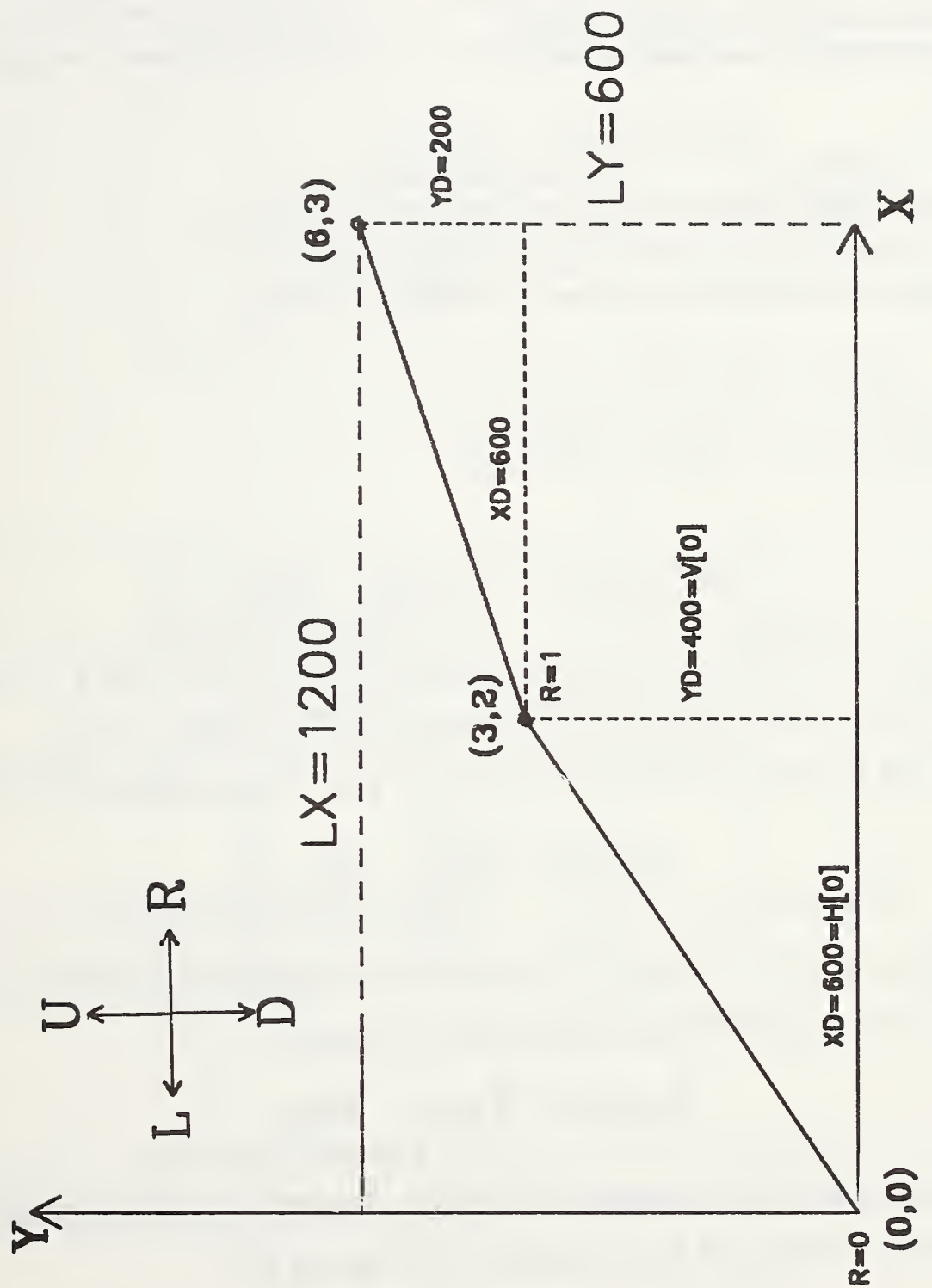


FIGURE 2.

Simplex Font: Size 1

!"#\$%&'()*+,-./0123456789:;<=>
 @ABCDEFGHIJKLMNPOQRSTUVWXYZ±∫
 °abcdefghijklmnpqrstuvwxy{ }
 ΩαβγδεζηθικλμνξοπρστυφχψωΣΠΓΘ

Cartographic Font: Size 2

!"#\$%&'()*+,-./0123456789:;<=>
 @ABCDEFGHIJKLMNPOQRSTUVWXYZ±∫
 °abcdefghijklmnpqrstuvwxy{ }
 ΩαβγδεζηθικλμνξοπρστυφχψωΣΠΓΘ

Index Font: Size 2

!"#\$%&'()*+,-./0123456789:;<=>?
 @ABCDEFGHIJKLMNPOQRSTUVWXYZ±∫
 °abcdefghijklmnpqrstuvwxy{ }
 ΩαβγδεζηθικλμνξοπρστυφχψωΣΠΓΘ

Math Symbols: Size 2

±∓×÷≠≡<>≤≥∞~^
 √[]→↑←↓∂∇§∫∫∞†‡

Complex Font: Size 2

!"#\$%&'()*+,-./0123456789:;<=>?
 @ABCDEFGHIJKLMNPOQRSTUVWXYZ
 °abcdefghijklmnpqrstuvwxy{ }
 ΩαβγδεζηθικλμνξοπρστυφχψωΣΠΓΘ

Script Font: Size 2

!"#\$%&'()*+,-./0123456789:;<=>?
 @ABCDEFGHIJKLMNPOQRSTUVWXYZ±∫
 °abcdefghijklmnpqrstuvwxy{ }
 ΩαβγδεζηθικλμνξοπρστυφχψωΣΠΓΘ

Gothic Font: Size 2

!"#\$%&'()*+,-./0123456789:;<=>?
 @ABCDEFGHIJKLMNPOQRSTUVWXYZ±∫
 °abcdefghijklmnpqrstuvwxy{ }
 ΩαβγδεζηθικλμνξοπρστυφχψωΣΠΓΘ

FIGURE 3.

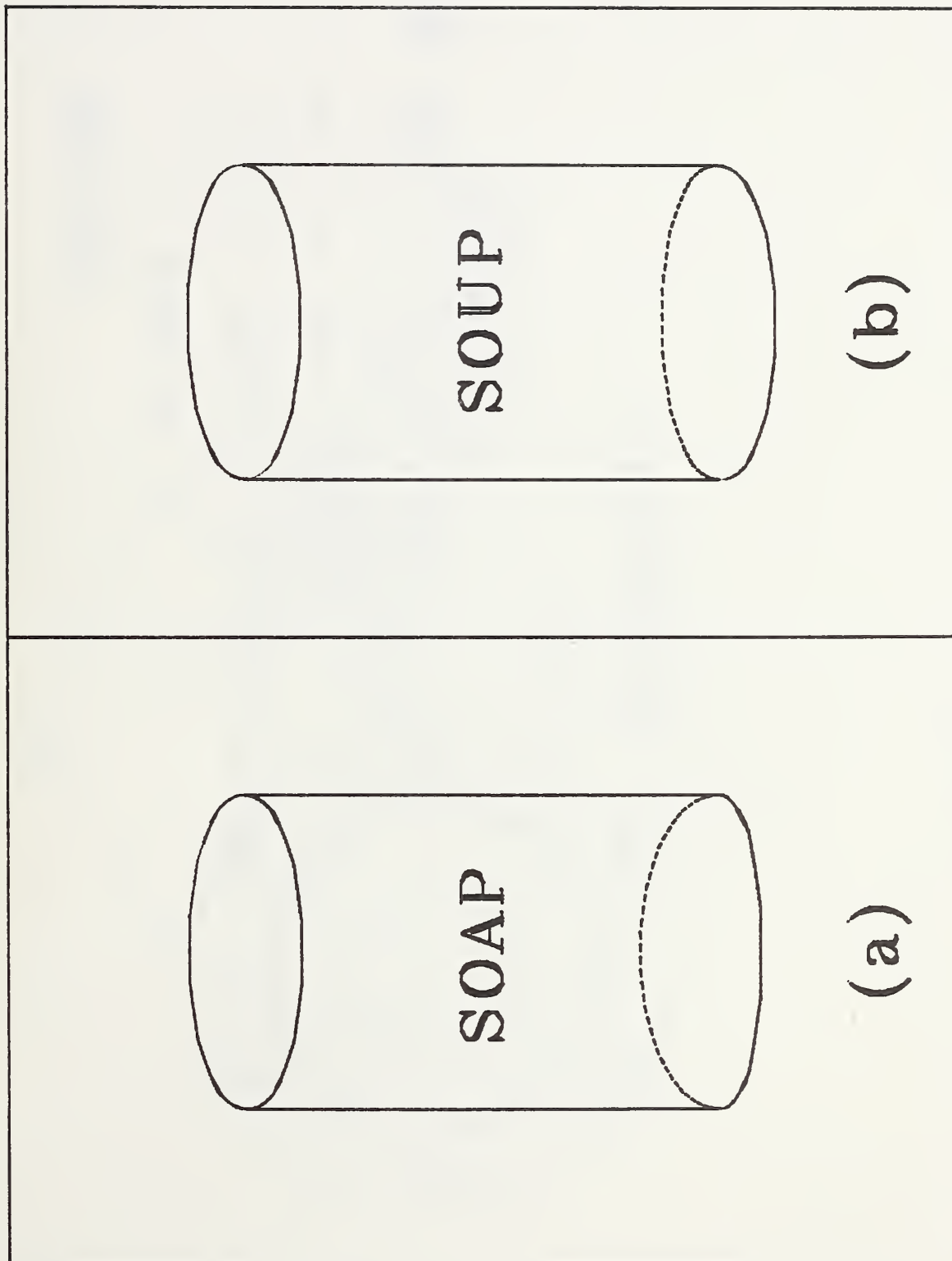
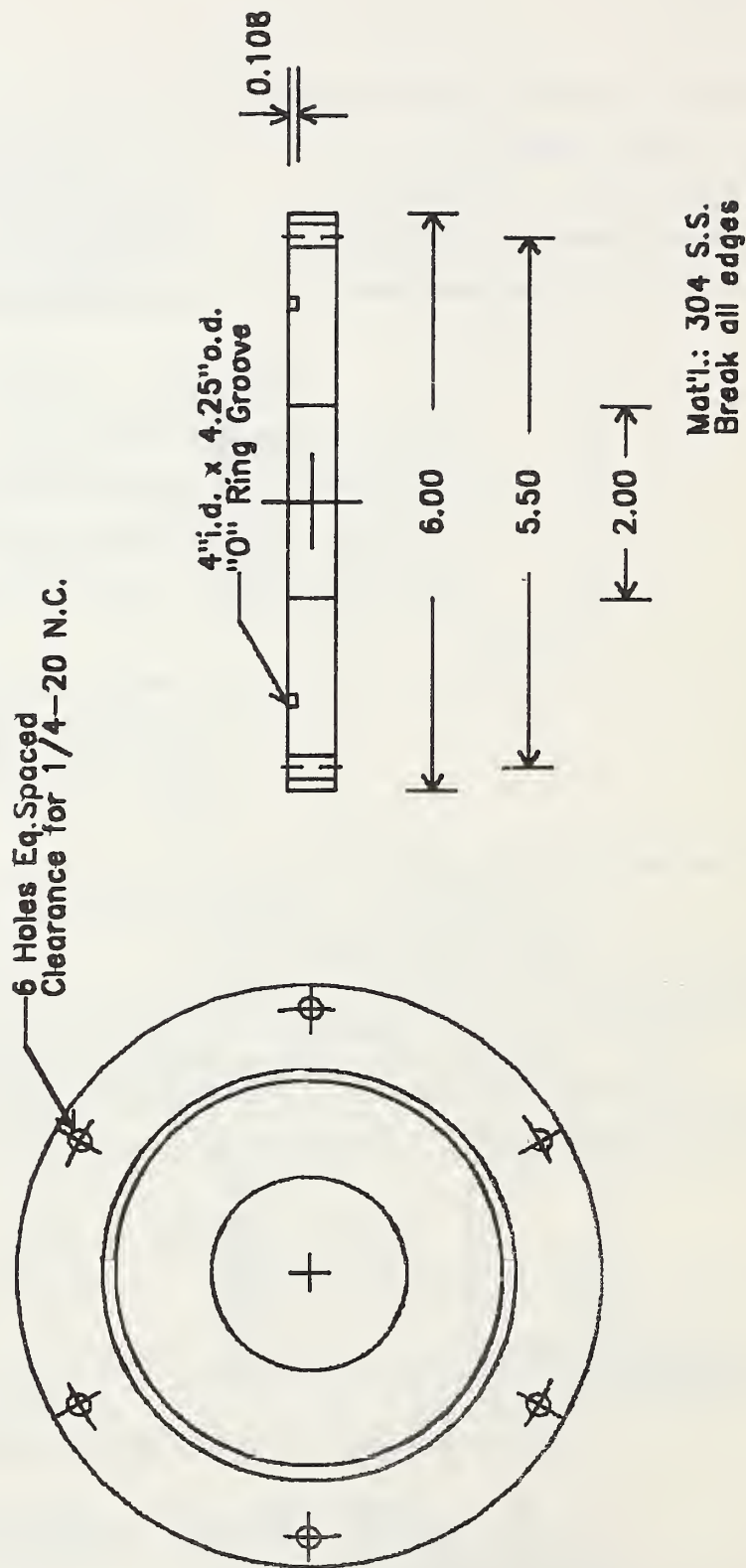


FIGURE 4.



Adapter Flange

Scale 1:2

FIGURE 5.

Accelerator Usage

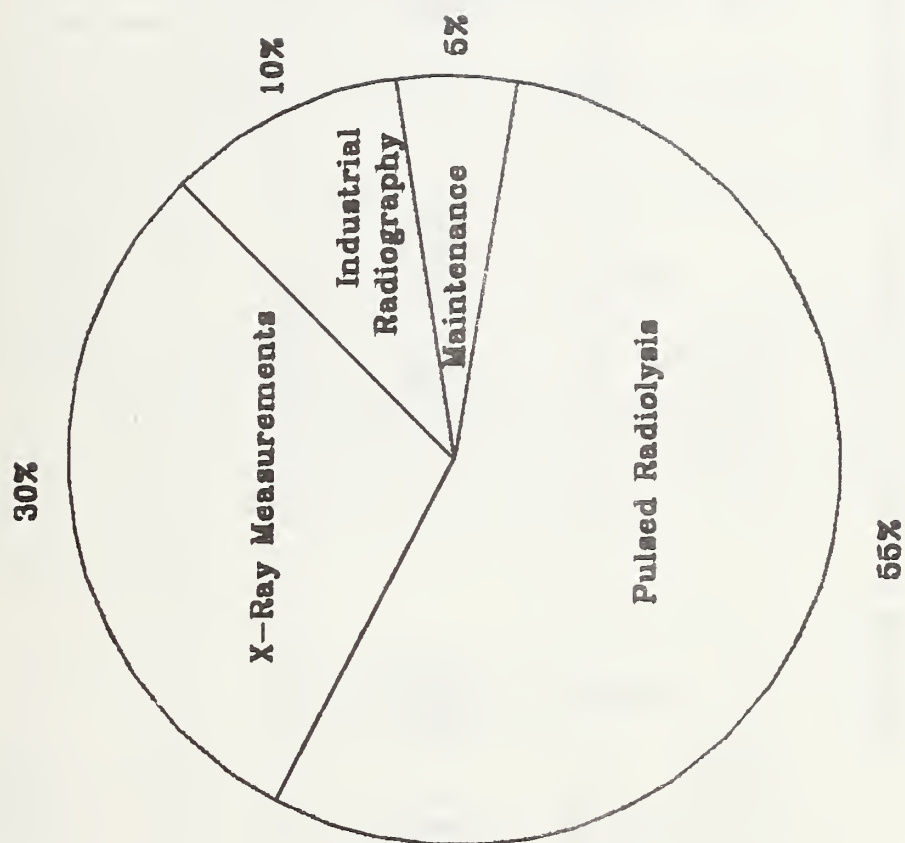
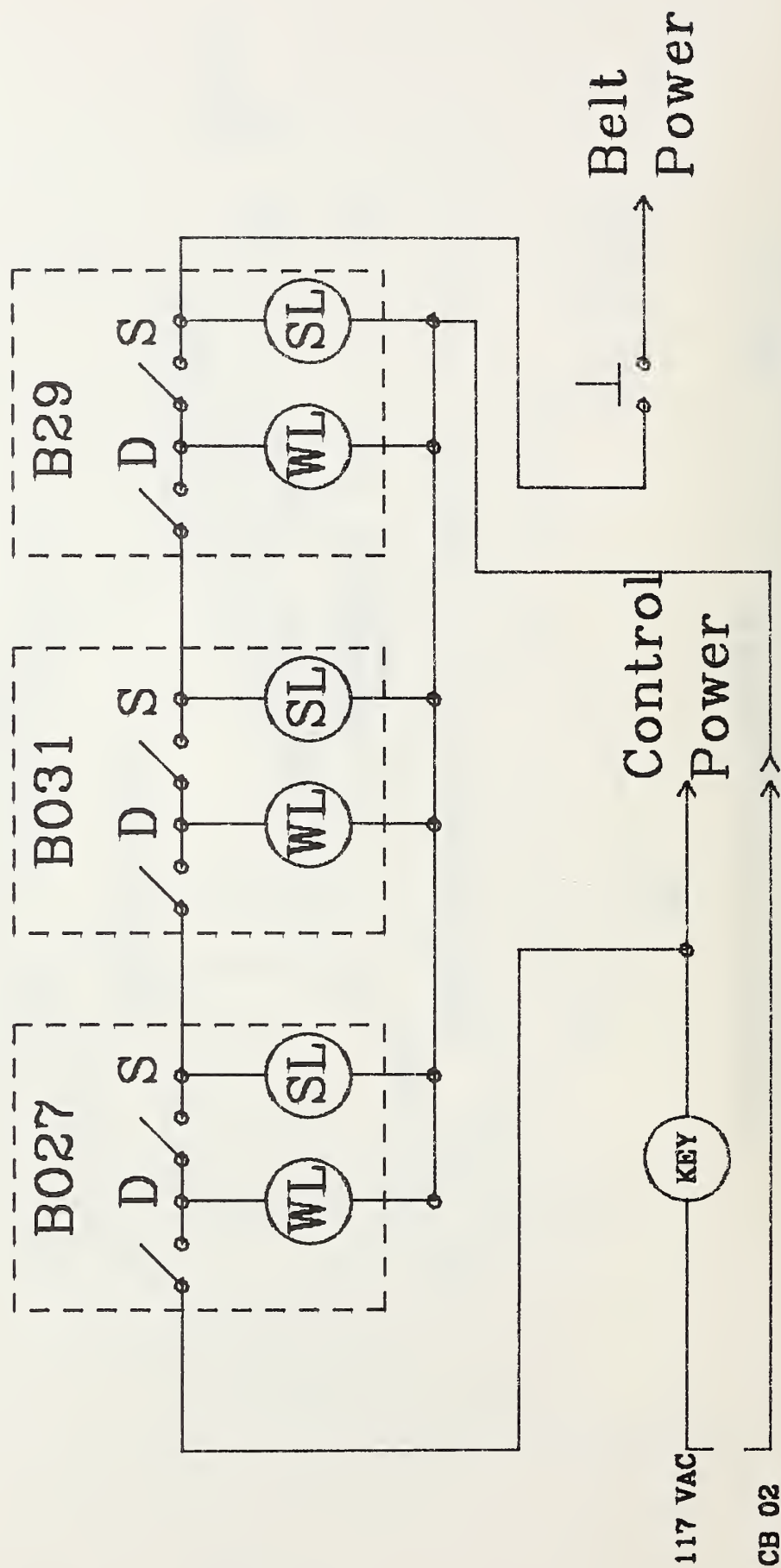


FIGURE 5.

4-MeV Electron Van de Graaff Safety Interlock System



D:Door Switch; S:Scram Switch; WL:Warning Light; SL:Scram Light

LISTINGS

1. A catalog of the diskette containing the software system.
2. The Applesoft Basic listing of the main program, DRAFTSMAN.
3. A listing of the variables used in DRAFTSMAN and the line numbers where they occur. This listing was produced with the variable documentation program of Wagner (9).
4. The instruction set used to produce the drawing of a can as shown in Fig. 4a. This illustration has several errors which are corrected as shown in Fig. 4b.
5. The edited version of the instruction set of Listing 4 which was used to produce the corrected can drawing of Fig. 4b.
6. The compiled listing of the assembly language I/O driver program BPLOTT.
7. A listing of a Basic program to calculate the best line between two points which has been used in BPLOTT to generate the appropriate string of ASCII characters to control the plotter. The comments indicate the correspondence between this program and the appropriate locations in BPLOTT.

LISTING 1.

CATALOG

DISK VOLUME 254

*A 052 DRAFTSMAN
*T 007 INDEX NUMBERS
*T 008 INDEX CAPS
*T 009 INDEX SMALL
*T 009 INDEX GREEK
*T 005 SIMPLEX NUMBERS
*T 005 SIMPLEX CAPS
*T 006 SIMPLEX SMALL
*T 006 SIMPLEX GREEK
*T 007 COMPLEX NUMBERS
*T 009 COMPLEX CAPS
*T 009 COMPLEX SMALL
*T 010 COMPLEX GREEK
*T 004 CARTO NUMBERS
*T 005 CARTO CAPS
*T 005 CARTO SMALL
*T 010 GOTHIC NUMBERS
*T 020 GOTHIC CAPS
*T 011 GOTHIC SMALL
*T 008 SCRIPT NUMBERS
*T 012 SCRIPT CAPS
*T 009 SCRIPT SMALL
*T 007 MATH SYMBOLS
*B 004 BPLOT
*B 006 INSTR.1
*B 005 INSTR.2
*B 005 INSTR.3
*B 005 INSTR.4
*B 005 INSTR.5
T 006 TEST
T 004 SOAP
T 005 SOUP

]

LISTING 2.

```

LIST
10 HIMEM: 33719
20 ONERR GOTO 12000
30 HOME :BELL$ = CHR$(7):D$ = CHR$(4)
40 A$ = "*****THE ELECTRONIC DRAFTSMAN"
50 HTAB (5): VTAB (10): PRINT "*****"
60 FOR I = 1 TO 5: HTAB (5): VTAB (10 + I): PRINT "*": HTAB (35): VTAB (1
    0 + I): PRINT "*": NEXT I
70 HTAB (5): VTAB (16): PRINT "*****"
80 HTAB (15): VTAB (12): PRINT "WELCOME TO"
90 FOR I = 1 TO LEN (A$) - 23: HTAB (8): VTAB (14): PRINT ( MID$ (A$,I,2
    4)):BELL$: FOR J = 1 TO 50: NEXT J: NEXT I
100 HTAB (15): VTAB (20): PRINT "COPYRIGHT 1981"
110 HTAB (20): VTAB (21): PRINT "C. E. DICK"
120 HTAB 25: VTAB 22: PRINT "REV. 11.3.2.83"
130 VTAB (24): PRINT "HIT ANY KEY TO BEGIN:":BELL$:BELL$: GET A$
140 HOME : VTAB (5): PRINT "THIS PROGRAM IS DESIGNED TO ALLOW YOU TOCREAT
    E YOUR OWN MECHANICAL OR SCHEMATIC DRAWINGS, LETTER THEM, AND SAVE TH
    EM ON DISK FILES FOR LATER RECALL.
150 PRINT BELL$
160 PRINT BELL$: INPUT "DO YOU NEED DIRECTIONS ?":Q$: IF LEFT$ (Q$,1) =
    "Y" THEN GOSUB 500
170 IF LEFT$ (Q$,1) < > "N" THEN 160
180 GOTO 1000
500 REM INSTRUCTIONS
510 FOR I = 1 TO 5
520 PRINT D$:"BLOAD INSTR.":I
530 VTAB 24: PRINT "STRIKE ANY KEY TO CONTINUE...": GET Q$
540 PRINT BELL$: HOME
550 NEXT
560 HOME : SPEED= 255: RETURN
1000 HOME : PRINT "NOW LOADING BINARY PLOTTING PROGRAMS"
1010 PRINT D$:"BLOAD BLOAD"
1020 PRINT : PRINT "POSITION PEN IN LOWER LEFT CORNER": PRINT "STRIKE ANY
    KEY WHEN READY-->":BELL$: GET Q$:LX = 0:LY = 0
1030 DIM B$(502),X$(140)
1040 OUT = 33720:LGEM = 33905:ALPHA = 33793:VGEN = 33855
1050 PC = 33911:XVEC = 33816:VVEC = 33836
1060 SIZE = 33801:ANGLE = 33802:CHAR = 33803:TIME = 33904
1070 VL = 34262:VH = 34263:HL = 34264:HH = 34265
1080 XX = 200: POKE TIME,25
1090 DF = PEEK (43624)
1100 PRINT BELL$: INPUT "DISK DRIVE NO. OF DATA:":DD: IF DD < 1 OR DD > 2
    THEN PRINT "ERROR...": GOTO 1100
2000 REM *** MAIN PROGRAM ***
2100 HOME :J = 0:PI = 3.14159264:R = 0:DOT = 0
2200 PRINT BELL$: INPUT "IS DRAWING ON DISK ?":Q$: IF LEFT$ (Q$,1) = "Y"
    THEN SF = 1: GOTO 6000
2210 Q$ = ""
3000 REM PROGRAM INSTR. LOOP
3010 A$ = "":R$ = "":RHO = 0:CF = 0: IF SF THEN HTAB (8): VTAB (10): PRINT
    "EXECUTING INSTRUCTION #":K
3020 IF SF THEN PRINT : FOR VV = 1 TO 255: PRINT " ": NEXT : HTAB (1): VTA
    (12): PRINT "CURRENT INSTR.":B$(K): PRINT : PRINT "PEN POSITION X=":
    LX / 200:" Y=":LY / 200: GOTO 3060
3030 PRINT : PRINT "PEN IS PRESENTLY AT X=":LX / 200:" Y=":LY / 200
3035 IF J = 475 THEN B$(J) = "Q": GOTO 4010

```

```

3040 PRINT BELL$; PRINT "INSTRUCTION NO. ("J;" "): INPUT " "E$(J)
3050 GOSUB 20000
3060 DEF FN C(I) = ASC ( MID$ (E$(J),I,1))
3070 IF J > 450 THEN PRINT BELL$;BELL$;"WARNING!! ONLY "475 - J;" INSTRU
CTIONS LEFT"
3080 FOR I = 1 TO LEN (E$(J)):E$ = MID$ (E$(J),I,1)
3090 IF I > LEN (E$(J)) THEN 3540
3100 IF E$ = "V" THEN GOTO 14000
3110 IF E$ = "7" THEN GOSUB 500: GOTO 3090
3120 IF E$ = "*" AND RHO = 0 THEN GOSUB 11000:E$ = "*": IF RHO = 0 THEN
RHO = 0.1
3130 IF E$ = "*" THEN DOT = 1:DL = RHO * INT (XX):RHO = 0: GOTO 3390
3140 IF E$ = "Q" THEN CF = 1: GOSUB 13000: GOTO 3390
3150 IF E$ = "U" THEN A = 90: GOTO 3390
3160 IF E$ = "S" THEN LF = 1: GOTO 3020
3170 IF E$ = "D" THEN A = 270: GOTO 3390
3180 IF E$ = ">" THEN AF = 1: GOTO 3390
3190 IF E$ = "<" THEN AF = 1:A = A + PI: GOTO 3390
3200 IF E$ = "R" THEN A = 0: GOTO 3390
3210 IF E$ = "L" THEN A = 180: GOTO 3390
3220 IF E$ = "I" THEN XX = 200: GOTO 3390
3230 IF E$ = "O" THEN OF = 1: GOTO 3390
3240 IF E$ = "C" THEN XX = 78.74: GOTO 3390
3250 IF E$ = "P" THEN I = I + 1: IF MID$ (E$(J),I,1) = "D" THEN PF = 1: GOTO
3390
3260 IF E$ = "P" AND MID$ (E$(J),I,1) = "A" THEN PRINT BELL$;"CHANGE PE
N TO SIZE # "; VAL ( MID$ (E$(J),I + 1,1)): PRINT BELL$;"HIT ANY KEY
TO CONTINUE ": GET A$:A$ = " ": GOTO 3710
3270 IF E$ = "P" THEN PF = 0: GOTO 3390
3280 IF E$ = "Q" THEN PRINT : PRINT "PLEASE STAND BY...":BELL$;MM = FRE
(0):I = LEN (E$(J)):XX = 200: GOTO 3390
3290 IF E$ = "T" THEN E$ = "W": GOSUB 8000: GOTO 3390
3300 IF E$ = "F" THEN FF = 1: GOTO 3390
3310 IF E$ = "H" THEN HF = 1: GOTO 3390
3320 IF E$ = "E" THEN EF = 1: GOTO 3390
3330 IF E$ = "W" THEN R$ = MID$ (E$(K),2, LEN (E$(K)) - 3):IV = VAL ( RIGHT$
(E$(K),1)):IW = INT (( VAL ( RIGHT$ (E$(K),2))) / 10): GOSUB 9170: GOTO
3710
3340 IF E$ = "A" THEN A$ = A$ + MID$ (E$(J),I + 1,1):I = I + 1: IF I < LEN
(E$(J)) AND ( FN C(I) > 45 AND FN C(I) < 58) THEN 3340
3350 IF E$ = "A" THEN A = VAL (A$): IF I < LEN (E$(J)) THEN I = I - 1
3360 IF E$ = "A" THEN GOTO 3390
3370 IF E$ = "X" OR E$ = "Y" THEN GOSUB 10000
3380 GOSUB 11000
3390 NEXT : IF E$ = "Q" THEN 4010
3500 REM FLAG DECODING ROUTINES
3510 IF E$ = "W" THEN 3710
3520 IF PF AND NOT (SF) THEN GOSUB 5000:FF = 0:J = J - 1: GOTO 3710
3530 IF PF AND SF THEN FOR I = 1 TO 5:K = K + 1:N$(I) = E$(K): NEXT I: GOSUB
5220:FF = 0: GOTO 3710
3540 IF (OF) OR (CF) THEN H(R) = XD:V(R) = YD:R = R + 1:XD = 0:YD = 0:OF =
0: IF NOT (CF) THEN 3710
3550 IF XYF THEN GOSUB 7000:RHO = SQR (X * X + Y * Y):A = ATN (Y / X):
XYF = 0: GOTO 3710
3560 IF EF THEN X = - X:Y = - Y:PF = 0: GOSUB 7020:EF = 0:J = J - 1: GOTO
3710
3570 IF HF AND R > 0 THEN PF = 0:PX = - XD:PY = - YD: GOSUB 7030:R = R -
1:YD = V(R):XD = H(R):HF = 0: GOTO 3710
3580 IF HF AND R = 0 THEN HF = 0: GOTO 3710
3590 IF CF THEN LIMIT = 1: FOR I = A(1) * 2 / PI TO .99 + A(2) * 2 / PI:X

```



```

      = COS (I * PI / 2) * AX(1):Y = SIN (I * PI / 2) * AX(2): GOSUB 700
0: IF X OR Y THEN I = .99 + A/27 * 2 / PI
3600 IF CF THEN NEXT I:LIMIT = 0: IF X OR Y THEN R = R - 1:YD = V(R):XD =
H(R):HF = 0:CF = 0: GOTO 3710
3610 IF CF THEN PP = 0:X = AX(1) * COS (A(1)):Y = AX(2) * SIN (A(1)): GOSUB
7000:PP = 1: FLASH :DA = PI / (40 * (AX(1) + AX(2))): PRINT : PRINT "
CALCULATING CIRCLE COORDINATES": IF NOT (DOT) THEN CALL OUT
3620 IF CF THEN X = AX(1) * (COS (A(1) + DA) - COS (A(1))):Y = AX(2) *
(SIN (A(1) + DA) - SIN (A(1))): IF DOT THEN PP = NOT (PP)
3630 IF CF THEN GOSUB 7000:A(1) = A(1) + DA: IF A(1) <= A(2) THEN 3620

3640 IF CF AND NOT (DOT) THEN GOSUB 7160
3650 IF CF THEN CF = 0:DOT = 0:X = 0:Y = 0:HF = 1: NORMAL : GOTO 3570
3660 IF NOT (AF) THEN A = A * PI / 180
3670 X = RHO * COS (A):Y = RHO * SIN (A)
3680 IF AF THEN PP = 1:X = RHO * COS (A + 5 * PI / 6):Y = RHO * SIN (A +
5 * PI / 6): GOSUB 7000:X = RHO * COS (A + 1.5 * PI):PP = 0
3690 IF AF THEN Y = RHO * SIN (A + 1.5 * PI): GOSUB 7000:X = RHO * COS
(A + PI / 4):Y = RHO * SIN (A + PI / 4):PP = 1: GOSUB 7000:AF = 0:PP
= 0: GOTO 3710
3700 GOSUB 7000
3710 IF NOT (SF) THEN J = J + 1
3720 RHO = 0: IF SF THEN RETURN
3730 GOTO 38010
4000 REM EXIT ROUTINES
4010 B$(J) = STR$(LX):B$(J + 1) = STR$(LY):B$(J + 2) = STR$(XD):B$(J
+ 3) = STR$(YD):B$(J + 4) = STR$(R):J = J + 5
4020 FOR II = 0 TO 10:B$(J) = STR$(H(II)):B$(J + 1) = STR$(V(II)):J =
J + 2: NEXT II:B$(J) = "END"
4030 PRINT BELL$: INPUT "WANT TO SAVE DRAWING?":B$: IF LEFT$(B$,1) = "Y
" THEN GOSUB 4500
4040 PRINT BELL$: INPUT "FINISHED DRAWING? ":Q$: IF LEFT$(Q$,1) = "N" THEN
IF (LX OR LY) THEN PRINT BELL$: INPUT "RETURN PEN TO (0,0)? ":B$: IF
LEFT$(B$,1) = "Y" THEN PP = 0:PX = - LX:PY = - LY: GOSUB 7050:XD =
0:YD = 0
4050 IF LEFT$(Q$,1) = "N" THEN GOTO 2000
4060 IF LEFT$(Q$,1) = "Y" THEN PRINT BELL$: PRINT "---BYE---": END
4070 GOTO 4030
5000 REM LETTER FONTS
5010 HOME : FOR I = 1 TO 130:X$(I) = " ": NEXT
5020 PRINT : PRINT TAB( 10): INVERSE : PRINT "INPUT LETTER FONTS": PRINT
: NORMAL : PRINT TAB( 11):"FONTS AVAILABLE:"
5030 PRINT : PRINT TAB( 10):"C1> INDEX": PRINT TAB( 10):"C2> SIMPLEX"
: PRINT TAB( 10):"C3> COMPLEX": PRINT TAB( 10):"C4> CARTOGRAPHIC"
: PRINT TAB( 10):"C5> SCRIPT": PRINT TAB( 10):"C6> GOTHIC": PRINT
: PRINT
5040 I = 1
5050 N$ = " NUMBERS": PRINT : INPUT "NUMBER FONT (1-60):":N:N = INT (N): IF
N < 1 OR N > 6 THEN PRINT "REINPUT ": GOTO 5050
5060 GOSUB 5140:
5070 N$ = " CAPS": PRINT : INPUT "UPPER CASE FONT (1-60):":N:N = INT (N): IF
N < 1 OR N > 6 THEN PRINT "REINPUT ": GOTO 5070
5080 GOSUB 5140
5090 N$ = " SMALL": PRINT : INPUT "LOWER CASE FONT (1-60):":N:N = INT (N):
IF N < 1 OR N > 6 THEN PRINT "REINPUT ": GOTO 5090
5100 GOSUB 5140
5110 N$ = " GREEK": PRINT : INPUT "GREEK FONT (1-30):":N:N = INT (N): IF N
< 1 OR N > 3 THEN PRINT "REINPUT ": GOTO 5110
5120 GOSUB 5140
5130 PRINT :N$(5) = "MATH SYMBOLS": PRINT N$(5):" ALSO BEING LOADED": GOTO

```

```

5210
5140 IF N = 1 THEN N$(I) = "INDEX"
5150 IF N = 2 THEN N$(I) = "SIMPLEX"
5160 IF N = 3 THEN N$(I) = "COMPLEX"
5170 IF N = 4 THEN N$(I) = "CARTO"
5180 IF N = 5 THEN N$(I) = "SCRIPT"
5190 IF N = 6 THEN N$(I) = "GOTHIC"
5200 N$(I) = N$(I) + N$: I = I + 1: RETURN
5210 FOR I = 1 TO 5: J = J + 1: E$(J) = N$(I): NEXT I: J = J + 1
5220 PRINT : FLASH : PRINT "NOW LOADING FONTS": NORMAL
5230 FOR I = 1 TO 2: PRINT N$(I): " " : NEXT I: PRINT
5240 FOR I = 3 TO 4: PRINT N$(I): " " : NEXT I: PRINT : PRINT "MATH SYMB
OLS"
5250 N = 0: FOR I = 1 TO 5
5260 PRINT D$:"OPEN ";N$(I);".D":DF
5270 PRINT D$:"READ ";N$(I)
5280 FOR JJ = 1 TO 32
5290 INPUT X$(N)
5300 N = N + 1
5310 NEXT JJ
5320 PRINT D$:"CLOSE ";N$(I)
5330 NEXT I
5340 MM = FRE (0): HOME :N$ = ""
5350 RETURN
6000 REM INPUT DRAWING
6010 PRINT BELL$;: INPUT "FILENAME OF DRAWING?":N$
6020 IF N$ = 0$ THEN E = K - 28:R = VAL (E$(E + 4)): GOTO 6120
6030 PRINT D$:"OPEN ";N$:".D":DD
6040 PRINT D$:"READ ";N$
6050 FOR K = 0 TO 502
6060 INPUT L$:E$(K) = ""
6070 FOR II = 1 TO L$: INPUT MID:E$(K) = E$(K) + CHR$(MID): NEXT II
6080 IF E$(K) = "END" THEN LX = VAL (E$(K - 27)):LY = VAL (E$(K - 26)):
XD = VAL (E$(K - 25)):YD = VAL (E$(K - 24)):R = VAL (E$(K - 23)):E
= K - 27
6090 IF E$(K) = "END" THEN K = K - 22: FOR II = 0 TO 10:H(II) = VAL (E$(
K)):V(II) = VAL (E$(K + 1)):K = K + 2: NEXT II:K = 502
6100 NEXT K
6110 PRINT D$:"CLOSE ";N$
6120 PRINT BELL$;: INPUT "CONT. (C) OR NEW PRINT (N)?":Q$: IF LEFT$ (Q$,
1) = "N" THEN 6160
6130 SF = 0: FOR J = E TO 0 STEP - 1: IF ((N$ < 0) AND (E$(J) = "F"))
THEN FOR I = 1 TO 5:N$(I) = E$(I + J): NEXT I: GOSUB 5220:J = 0
6140 NEXT J:J = E: PRINT BELL$:"SHALL I MOVE PEN FROM 0,0 TO ";LX / 200:"
";LY / 200: INPUT A$: IF LEFT$ (A$, 1) = "Y" THEN PX = LX:PY = LY:P
= 0: GOSUB 7100
6150 GOTO 3010
6160 LX = 0:LY = 0: PRINT BELL$:"POSITION PEN IN LOWER LEFT CORNER": PRINT
"STRIKE ANY KEY WHEN READY": GET A$
6170 FOR K = 0 TO E - 1:J = K: GOSUB 3010
6180 NEXT K:SF = 0:J = E: GOTO 3010
6500 REM SAVE DRAWING
6510 PRINT BELL$;: INPUT "FILENAME?":O$
6520 PRINT D$:"OPEN ";O$:".D":DD
6530 PRINT D$:"DELETE ";O$
6540 PRINT D$:"OPEN ";O$
6550 PRINT D$:"WRITE ";O$
6560 FOR K = 0 TO J:LL = LEN (E$(K))
6570 PRINT LL
6580 FOR II = 1 TO LL: PRINT ASC (MID$ (E$(K),II,1)): NEXT II

```

```

6590 NEXT K
6600 PRINT D#;"CLOSE ";C#
6610 RETURN
7000 REM *** OUTPUT ROUTINES ***
7010 REM ** ENTER WITH X,Y,PP **
7020 PX = X * XX;PY = Y * XX
7030 IF PX - INT (PX) > .5 THEN PX = INT (PX) + 1
7040 IF PY - INT (PY) > .5 THEN PY = INT (PY) + 1
7050 PX = INT (PX);PY = INT (PY); IF TX OR TY THEN 7080
7060 LX = LX + PX;LY = LY + PY; IF LX < 0 OR LX > 2050 OR LY < 0 OR LY > 1
450 THEN PRINT : PRINT "PLOTTER LIMITS EXCEEDED...";BELL#;BELL#;BELL
#:LX = LX - PX;LY = LY - PY;J = J - 1; RETURN
7070 IF CF AND LIMIT THEN X = 0;Y = 0;LX = LX - PX;LY = LY - PY; RETURN
7080 IF ((DOT) AND (NOT (CF))) THEN GOSUB 15000; RETURN
7090 XD = XD + PX;YD = YD + PY
7100 XH = INT (PX / 256);XL = PX - XH * 256
7110 YH = INT (PY / 256);YL = PY - YH * 256
7120 IF XH < 0 THEN XH = XH + 256
7130 IF YH < 0 THEN YH = YH + 256
7140 POKE VH,YH; POKE VL,YL; POKE NH,XH; POKE HL,XL; CALL VGEN
7150 IF CF AND NOT (DOT) THEN 7190
7160 POKE CHAR,PP + 121; POKE TIME,255; CALL ALPHA; IF CF THEN POKE TIME
,30; CALL OUT; GOTO 7190
7170 POKE TIME,32; IF PP = 0 THEN POKE TIME,1
7180 CALL OUT
7190 RETURN
8000 REM *** TEXT GENERATOR ***
8010 HOME
8020 PRINT "CHARACTER STRING: ";LC = 0;SC = 0; INVERSE
8030 IF OS# > "" THEN LL = LEN (OS#) - 1;R# = OS#; FOR IC = 1 TO LL: GOSUB
8220: NEXT : PRINT : PRINT "SIZE ";RIGHT# (R#,1); INPUT "OK ? ";A#; IF
LEFT# (A#,1) = "Y" THEN IW = VAL (RIGHT# (R#,1));OS# = "";R# = MID#
(R#,1,LL); GOTO 8130
8040 IF OS# < " " AND LEFT# (A#,1) = "N" THEN PRINT : PRINT "CHARACTE
R STRING: ";OS# = "";R# = ""
8050 PRINT BELL#; GET T#; IF T# = CHR# (13) THEN LC = 0; NORMAL : GOTO
8110
8060 IF T# = CHR# (7) OR T# = CHR# (19) THEN FLASH :R# = R# + T#;SC =
1; GOTO 8050
8070 IF T# = CHR# (27) THEN LC = NOT (LC); GOTO 8050
8080 IF NOT (LC) AND NOT (SC) THEN INVERSE
8090 IF LC AND NOT (SC) AND ASC (T#) > 33 THEN NORMAL :T# = CHR# (ASC
(T#) + 128)
8100 PRINT T#;R# = R# + CHR# (ASC (T#));SC = 0; GOTO 8050
8110 PRINT : PRINT BELL#; INPUT "SIZE (IN TENTHS) ";IW;IW = INT (IW); IF
IW < 1 OR IW > 5 THEN PRINT BELL#;"RE-INPUT "; GOTO 8110
8120 IF LF THEN W = 0; GOTO 8190
8130 PRINT BELL#; INPUT "HORIZ.CHD OR VERT.CHD CHARACTERS ?";L#; IF L# <
> "H" AND L# < > "V" THEN PRINT BELL#;"REINPUT "; GOTO 8130
8140 IV = 0; IF L# = "V" THEN IV = 04
8150 E#(J) = E# + R# + CHR# (48 + IW) + CHR# (48 + IV);W = 0
8160 IF E#(TEMP) = "V" THEN RETURN
8170 IF NOT (SF) THEN WF = 1
8180 IF LEN (R#) = 0 THEN PRINT BELL#;"ERROR-REINPUT: "; GOTO 8030
8190 PRINT
8200 REM TEXT STRING DECODER
8210 PRINT : PRINT : FOR IC = 1 TO LEN (R#)
8220 P# = MID# (R#,IC,1);P = ASC (P#); IF P = 7 OR P = 19 THEN IC = IC +
1;P# = MID# (R#,IC,1)
9500 REM *** MAKE HERSHEY VECTORS ***

```

```

8510 IF P# = "" THEN PRINT BELL#;"ERROR-REINPUT ":R# = "": GOTO 8030
8520 L = ASC (P#) - 32
8530 IF (P = 7 OR P = 19) AND (L > 95) THEN L = L - 128
8540 IF P = 7 THEN L = L + 34
8550 IF P = 19 THEN L = L + 96
8560 IF L > 159 THEN L = L - 96
8570 IF OS# > "" THEN GOTO 9020
8580 IF L > 159 OR L < 0 THEN PRINT BELL#;"REINPUT ":R# = "": GOTO 8030

8590 IF (WF) OR LF THEN GOTO 8770
8600 GOSUB 9000
8610 PF = 0
8620 IF IV = 0 THEN XD = XD + (IW * (ASC (MID# (X$(L),1,1)) - 93)): IF
SF THEN LX = LX + (IW * (ASC (MID# (X$(L),1,1)) - 93))
8630 IF IV = 3 THEN YD = YD + (IW * (ASC (MID# (X$(L),1,1)) - 93)): IF
SF THEN LY = LY + (IW * (ASC (MID# (X$(L),1,1)) - 93))
8640 FOR G = 2 TO LEN (X$(L)) STEP 1
8650 DX = ASC (MID# (X$(L),G,1)) - 93
8660 DY = ASC (MID# (X$(L),G + 1,1)) - 93
8670 IF DX = 34 AND LEN (X$(L)) > 4 THEN PF = NOT (PF)
8680 IF DX = 34 THEN G = G - 1: NEXT G
8690 DX = DX * IW: DY = DY * IW
8700 IF DX < 0 THEN DX = 256 + DX
8710 IF DY < 0 THEN DY = 256 + DY
8720 POKE XVEC,DX: POKE YVEC,DY: POKE PC,PF + 121: CALL LGEN
8730 NEXT G
8740 POKE ANGLE,IV: POKE TIME,24 + 2 * IW + 160 / 2 * IW
8750 CALL OUT: POKE SIZE,1: POKE ANGLE,0: POKE TIME,25
8760 NEXT IC: HOME: RETURN
8770 W = W + ASC (MID# (X$(L),1,1)) - 93
8780 NEXT IC
8790 PRINT
8800 W = W * IW
8810 PRINT BELL#: PRINT "LENGTH OF STRING IS "W / XX: IF XX = 200 THEN
PRINT " INCHES"
8820 IF XX = 78.74 THEN PRINT " CENTIMETERS"
8830 IF LF THEN LF = 0: OS# = R# + CHR# (48 + IW): GOTO 8010
8840 PRINT BELL#: INPUT "IS THIS LENGTH O.K. (Y) OR (N):": Q#: IF LEFT# (
Q$,1) < > "Y" AND LEFT# (Q$,1) < > "N" THEN GOTO 8840
8850 IF LEFT# (Q$,1) = "N" THEN GOTO 8900
8860 IF IV = 0 THEN LX = LX + W: IF LX > 2050 THEN LX = LX - W: LIMIT = 1
8870 IF IV = 06 THEN LY = LY + W: IF LY > 1450 THEN LY = LY - W: LIMIT = 1

8880 IF LIMIT THEN LIMIT = 0: PRINT BELL#:BELL#:BELL#;"PLOTTER LIMITS EXC
EDED...": GOTO 8900
8890 WF = 0: GOTO 8210
8900 PRINT: PRINT BELL#:: INPUT "REINPUT(R) OR ABORT(A)?": Q#:R# = "": IF
LEFT# (Q$,1) = "R" THEN PRINT BELL#: PRINT "REINPUT: ": GOTO 8030
8910 IF LEFT# (Q$,1) < > "A" THEN GOTO 8900
8920 GOTO 3040
9000 REM TEXT GENERATOR DISPLAY
9010 VTAB (20)
9020 NORMAL
9030 IF P = 7 OR P = 19 THEN FLASH
9040 IF L > 31 AND L < 33 THEN INVERSE
9050 IF OS# = "" THEN HTAB (IC)
9060 PRINT P#:: NORMAL
9070 RETURN
10000 REM XY ROUTINES
10010 IF NOT (XYF) THEN X = 0: Y = 0

```



```

10020 XYF = 1: IF E# = "X" AND RHO < > 0 THEN X = RHO:R# = "":RHO = 0: RETURN
10030 IF E# = "Y" AND RHO < > 0 THEN Y = RHO:R# = "":RHO = 0: RETURN
10040 IF E# = "X" AND RHO = 0 THEN I = I + 1: E# = MID# (B#(J),I,1): GOSUB
11000: X = RHO:RHO = 0:R# = "": RETURN
10050 IF E# = "Y" AND RHO = 0 THEN I = I + 1: E# = MID# (B#(J),I,1): GOSUB
11000: Y = RHO:RHO = 0:R# = "": RETURN
11000 REM CALCULATION OF RHO VALUE
11010 IF E# > "," AND E# < ":" THEN R# = R# + E#:RHO = VAL (R#)
11020 I = I + 1: E# = MID# (B#(J),I,1): IF E# < "-" OR E# > "9" OR I > LEN
(B#(J)) THEN I = I - 1:R# = "": RETURN
11030 GOTO 11010
12000 REM ERROR HANDLING ROUTINES
12010 GOOF = PEEK (222)
12020 IF GOOF = 133 AND XYF THEN A = ATN (Y / 1.E - 6):XYF = 0: GOTO 371
0
12030 PRINT : PRINT BELL#:"ERROR #":GOOF:" ENCOUNTERED IN LINE ": PEEK (2
18) + 256 * PEEK (219): PRINT "ERROR HANDLER ACTIVE...":BELL#: FOR Z
= 1 TO 200: NEXT
12040 IF GOOF = 53 THEN PRINT : PRINT BELL#:"NEED FONTS TO CONTINUE":BEL
L#: FOR ZZ = 1 TO 1000: NEXT ZZ: E#(J+6) = E#(J): HOME: E#(J) = "F":
GOSUB 5000: RESUME
12050 IF (GOOF = 5) OR (GOOF = 4) THEN DN = PEEK (43624):DN = NOT (DN -
1) + 1: POKE 43624,DN: IF DD < > DN THEN DD = DN: GOTO 4030
12060 IF GOOF = 132 THEN PRINT "DIVISION BY ZERO ERROR---NEED RADIUS TO
DRAW CIRCLE": GOTO 2040
12070 IF GOOF = 9 THEN PRINT BELL#:"DISK FULL, INSERT NEW DISKETTE":BELL#
12080 IF GOOF = 10 THEN PRINT BELL#:"WARNING!!, ":N#:" IS WRITE PROTECTE
D": PRINT "DO YOU WISH TO DESTROY OLD FILE": INPUT A#: IF LEFT# (A#
,1) = "N" THEN GOTO 4030
12090 IF LEFT# (A#,1) = "Y" THEN PRINT B#:"UNLOCK ":N#
12100 IF GOOF = 255 THEN PRINT "BREAK IN LINE NO. ": PEEK (219) + 256 *
PEEK (219): PRINT : PRINT "WARM RESTART AT 3000": STOP
12110 IF GOOF = 77 THEN MM = FRE (0)
12120 RESUME
13000 REM CIRCLE AND ELLIPSE DECODING
13010 ZZ = 1
13020 A(1) = 0:A(2) = 2 * PI
13030 AX(1) = RHO:AX(2) = RHO:RHO = 0
13040 I = I + 1: E# = MID# (B#(J),I,1)
13050 IF I > LEN (B#(J)) THEN RETURN
13060 IF ((E# < > ",.") AND (E# < "0" OR E# > "9") AND (E# < > "G") AND
(E# < > "^")) THEN PRINT "SYNTAX ERROR--REINPUT..":CF = 0: GOTO 300
0
13070 IF E# = "G" THEN 13040
13080 IF I > LEN (B#(J)) THEN RETURN
13090 IF E# = "C" THEN XX = 73.74: GOTO 13040
13100 IF E# = "I" THEN XX = 200: GOTO 13040
13110 IF E# < > "^" THEN 13130
13120 IF I = LEN (B#(J)) THEN RETURN
13130 IF RHO = 0 THEN GOSUB 11000
13140 A(ZZ) = RHO * PI / 180:RHO = 0: IF A(1) > A(2) THEN A(2) = A(2) + 2 *
PI
13150 ZZ = ZZ + 1: GOTO 13040
13160 GOSUB 11000
13170 IF AX(1) THEN AX(2) = RHO:RHO = 0: GOTO 13040
13180 GOTO 13030
15000 REM DOTTED LINE OUTPUT
15010 IF DL = 0 THEN DL = 30

```

```

15020 TX = PX:TY = PY:DCT = 0
15030 IF ((TX = 0) AND (TY = 0)) THEN PP = 0:DL = 0: RETURN
15040 IF ABS (TX) > ABS (TY) THEN PX = DL * SGN (PX):PY = DL * SGN (PY) * ABS (TY / TX)
15050 IF ABS (TX) < ABS (TY) THEN PY = DL * SGN (PY):PX = DL * SGN (PX) * ABS (TX / TY)
15060 IF ((ABS (PX) > ABS (TX)) OR (ABS (PY) > ABS (TY))) THEN PX = TX:PY = TY
15070 PP = NOT (PP): GOSUB 7030:TX = TX - PX:TY = TY - PY: GOTO 15030
16000 HOME : PRINT : PRINT TAB( 12):"EDITING ROUTINES"
16010 PRINT BELL$: INPUT "DUMP INSTRUCTION SET TO PRINTER?"A$: IF LEFT$(A$,1) = "Y" THEN 16230
16020 PRINT BELL$: INPUT "EDIT ALL (A) OR SELECTED (S) INSTR.?"A$
16030 IF LEFT$(A$,1) = "E" THEN GOSUB 16110: GOTO 3000
16040 IF LEFT$(A$,1) < > "A" THEN 16020
16050 HOME :ZZ = 0
16060 PRINT "CURRENT INSTRUCTIONS ARE---"
16070 PRINT " NO. INSTRUCTION---"
16080 FOR II = 1 TO 10: PRINT ZZ: TAB( 7):B$(ZZ): IF ZZ = J - 1 THEN GOSUB 16200: GOTO 3000
16090 ZZ = ZZ + 1: NEXT
16100 GOSUB 16200: GOTO 16070
16110 PRINT BELL$: INPUT "EDIT INSTRUCTION NO.?"Z
16120 PRINT "CURRENT INSTRUCTION IS:"B$(Z):BELL$
16130 PRINT BELL$: INPUT "CHANGE IT?"A$: IF LEFT$(A$,1) < > "Y" THEN 16140
16140 INPUT "DO YOU WISH TO ENTER A TEXT STRING?"A$: IF LEFT$(A$,1) = "Y" THEN OS$ = "":R$ = "":B$ = "W":TEMP = J:J = Z: GOSUB 3020:J = TEMP: GOTO 16140
16150 INPUT "REPLACE WITH:"B$(Z)
16160 INPUT "FINISHED?"A$
16170 IF LEFT$(A$,1) = "N" THEN 16110
16180 IF LEFT$(A$,1) = "Y" THEN RETURN
16190 GOTO 16140
16200 INPUT "ARE THESE OK?"A$: IF LEFT$(A$,1) = "N" THEN 16110
16210 IF LEFT$(A$,1) < > "Y" THEN 16200
16220 RETURN
16230 IF N$ = "" THEN PRINT : INPUT "FILENAME OF DRAWING:"N$
16240 PR# 1: PRINT CHR$(9):"90M": PRINT CHR$(12)
16250 PRINT "FILENAME:"N$: PRINT : PRINT
16260 FOR ZZ = 0 TO J - 1: PRINT "INSTR. NO.":ZZ,B$(ZZ): NEXT ZZ
16270 PRINT CHR$(12): PR# 0: PRINT
16280 INPUT "RETURN TO EDITOR (E) OR INTERPRETER (I)?"A$: IF LEFT$(A$,1) = "E" THEN 16020
16290 IF LEFT$(A$,1) = "I" THEN 3000
16300 GOTO 16230
20000 REM ILLEGAL CHARACTER CHECKS
20010 FOR I = 1 TO LEN (B$(J)):TC$ = MID$(B$(J),I,1)
20020 IF TC$ < "*" OR TC$ = "=" OR TC$ = "/" OR TC$ = "C" OR TC$ = "J" OR TC$ = "K" OR TC$ = "Z" OR TC$ = "B" OR TC$ = "N" OR TC$ = "M" THEN PRINT BELL$:"ILLEGAL CHARACTER ...":I = LEN (B$(J)): NEXT : GOTO 3040
20030 NEXT : RETURN

```


LISTING 3.

```

*****
*
*           DRAFTSMAN
*
*  --)TABLE OF VARIABLES(--
*
*****

```

```

%Q(*)  A  A$  A(*)  AF  ALPHA  ANGLE  AX(*)  B$  B(*)  BELL$  OF  CHAR  B$
DA  DD  DF  DL  DN  DOT  DX  DY  E  EF  FF  G  GDEF  H(*)  HF  HH  HL  I  IC
II  IV  IW  J  JJ  K  L  L$  LP  LC  LF  LGEN  LIMIT  LL  LX  LY  MID  MM  N
N$  N(*)  O$  OF  OS$  OUT  P  P$  PC  PI  PP  PY  Q$  R  R$  RND  SD
SF  SIZE  T$  TC$  TEMP  TIME  TX  TY  V(*)  VGEN  VH  VL  VV  W  WF  X
X(*)  XD  XH  XL  XVEC  XX  XYF  Y  YD  YH  YL  YVEC  Z  ZZ
END OF VAR. LIST

```

```

*****
*
*           DRAFTSMAN
*
*  --)TABLE OF VARIABLES(--
*
*****

```

```

%Q(*) - FUNCTION = ASC(MID$(Q$,I,1))
3040 3340 3340

```

```

A - ANGLE IN DEGREES
3150 3170 3190 3190 3200 3210 3350 3550 3660 3660 3670 3670 3680
3680 3680 3690 3690 3690 12020

```

```

A$ - STRING VARIABLE USED MAINLY FOR INPUTTING ANSWERS AND FOR LOGO
40 90 90 130 3010 3260 3260 3340 3340 3350 6140 6140 6160 8030
8030 8040 12080 12080 12090 16010 16010 16020 16030 16040 16130
16130 16140 16140 16160 16170 16180 16200 16200 16210 16280 16280
16290

```

```

A(*) - ANGULAR LIMITS FOR ARCS A(1)=STARTING ANGLE A(2)=ENDING ANGLE
3610 3610 3620 3620 3620 3620 3630 3630 3630 3630 13020 13020
13140 13140 13140 13140 13140

```

```

AF - ARROW FLAG 1=SET
3180 3190 3660 3680 3690 3690

```

ALPHA - ONE CHARACTER OUTPUT ROUTINE IN BPL0T
1040 7160

ANGLE - ANGLE FOR BPL0T OUTPUT ROUTINE
1060 8740 8750

AX(*) - AXES FOR ELLIPSES AX(1)=MAJOR AXIS AX(2)=MINOR AXIS
3590 3590 3610 3610 3610 3620 3620 13030 13030 13170 13170

B# - CHARACTER IN INPUT INSTRUCTION STRING
3080 3100 3110 3120 3120 3130 3140 3150 3160 3170 3180 3190 3200
3210 3220 3230 3240 3250 3260 3270 3280 3290 3290 3300 3310 3320
3330 3340 3350 3360 3370 3370 3390 3510 4030 4030 4040 4050 4610
8150 10020 10030 10040 10040 10050 10050 11010 11010 11010 11020
11020 11020 13040 13060 13060 13060 13060 13060 13070 13090 13100
13110 16140

B#(*) - INPUT INSTRUCTION STRING ARRAY 475 MAX.
1030 3020 3025 3040 3060 3080 3080 3090 3250 3260 3260 3280 3330
3330 3330 3330 3340 3340 3350 3530 4010 4010 4010 4010 4010 4020
4020 4020 5210 6020 6060 6070 6070 6080 6080 6080 6080 6080 6080
6090 6090 6090 6130 6130 6560 6580 9150 9160 10040 10050 11020
11020 12040 12040 12040 13040 13050 13080 13120 16080 16120 16150
16260 20010 20010 20020

BELL# - CHR#(7) SOUNDS SPEAKER
30 90 130 130 150 160 540 1020 1100 2200 3040 3070 3070 3260
3260 3280 4030 4040 4040 4060 6010 6120 6140 6160 6510 7060 7060
7060 8050 8110 8110 8130 8130 8180 8510 8580 8810 8840 8880 8880
8880 8900 8900 12030 12030 12040 12040 12070 12070 12080 16010
16020 16110 16120 16130 20020

CF - CIRCLE FLAG 1=SET
3010 3140 3540 3540 3590 3600 3600 3610 3620 3630 3640 3650 3650
7070 7080 7150 7160 13060

CHAR - OUTPUT CHARACTER IN BPL0T
1060 7160

D# - CHR#(4) DOS CONTROL
30 520 1010 5260 5270 5320 6030 6040 6110 6520 6530 6540 6550
6600 12090

DA - DELTA ANGLE FOR DRAWING CIRCLES AND ELLIPSES
3610 3620 3620 3630

DD - DISK DRIVE NUMBER 1 OR 2
1100 1100 1100 6030 6520 12050 12050

DF - CURRENT DISK DRIVE IN USE
1090 5260

DL - DOT LENGTH FOR DOTTED LINES
3130 15010 15010 15030 15040 15040 15050 15050

DN - DISK DRIVE NUMBER
12050 12050 12050 12050 12050 12050

DOT - DOTTED LINE TOKEN 1=SET

2100 3130 3610 3620 3640 3650 7090 7150 15020

DX - DELTA X

8650 8670 8680 8690 8690 8700 8700 8700 8720

DY - DELTA Y

8660 8690 8690 8710 8710 8710 8720

E - END OF FILE TOKEN

6020 6020 6080 6130 6140 6170 6180 12020

EF - ERASE FLAG 1=SET

3320 3560 3560

FF - FONT FLAG 1=SET

3300 3520 3520 3530 3530

G - LOOP INDEX FOR TEXT STRING DECODER

8640 8650 8660 8680 8690 8690 8730

GOOF - ERROR NUMBER PEEK(222)

12010 12020 12030 12040 12050 12050 12060 12070 12080 12100 12110

H(*) - HORIZONTAL HOME ARRAY

3540 3570 3600 4020 6090

HF - HOME FLAG 1=SET

3310 3570 3570 3590 3590 3600 3650

HH - HORIZONTAL HIGH BYTE

1070 7140

HL - HORIZONTAL LOW BYTE

1070 7140

I - GENERAL LOOP INDEX

60 60 60 60 70 70 70 70 510 520 3060 3060 3080 3080 3090 3250
3250 3250 3260 3260 3280 3340 3340 3340 3340 3340 3340 3350 3350
3350 3530 3530 3530 3590 3590 3590 3590 3600 5010 5010 5040 5140
5150 5160 5170 5180 5190 5200 5200 5200 5200 5210 5210 5210 5230
5230 5230 5240 5240 5240 5250 5260 5270 5320 5330 6130 6130 6130
10040 10040 10040 10050 10050 10050 11020 11020 11020 11020 11020
11020 13040 13040 13040 13050 13080 13120 20010 20010 20020

IC - LOOP INDEX FOR HERSEY CHARACTER DECODER

9030 8210 8220 8220 8220 8220 8760 8780 9050

II - LOOP INDEX

4020 4020 4020 4020 6070 6070 6090 6090 6090 6090 6580 6580 6580
16080

IV - CHARACTER DIRECTION INDICATOR 00=HOR. 06=VERT.

3330 9140 9140 8150 8620 8630 8740 8860 8870

IW - CHARACTER SIZE

3330 8030 8110 8110 8110 8110 8110 8150 8620 8620 8630 8630 8690
8690 8740 8740 8800 8830

J - LOOP INDEX MAINLY FOR INPUT INSTRUCTION DECODING

90 90 2100 3035 3035 3040 3040 3060 3070 3070 3090 3090 3090

3250 3260 3260 3280 3340 3340 3350 3520 3520 3560 3560 3710 3710
 4010 4010 4010 4010 4010 4010 4010 4020 4020 4020 4020 4020 4020
 5210 5210 5210 5210 6130 6130 6130 6170 6140 6140 6170 6190 6560
 7060 7060 8150 10040 10050 11020 11020 12040 12040 12040 12040
 13050 13080 13120 14090 14140 14140 14140 14240 20010 20010 20020

JJ - LOOP INDEX
 5290 5310

K - LOOP INDEX MAINLY FOR DISK INPUT
 3010 3020 3330 3330 3330 3330 3530 3530 3530 3530 4020 4050 4060 4070
 4070 4080 4080 4080 4080 4080 4080 4080 4080 4080 4080 4080 4080
 4090 4090 4090 4100 4170 4170 4170 4180 4560 4560 4580 4580

L - HERSCHEY VECTOR INPUT INDEX
 8520 8530 8530 8530 8540 8540 8550 8550 8560 8560 8560 8560 8580 8580
 8620 8620 8630 8630 8640 8650 8660 8670 8770 9040 9040

L# - CHARACTER SIZE AND ORIENTATION STRING
 8130 8130 8130 8140

LB - LENGTH OF FILE BUFFER
 4060 4070

LC - LOWER CASE FLAG 1=SET
 8020 8050 8070 8070 8080 8090

LF - LENGTH FLAG 1=SET
 3160 8120 8580 8830 8830

LDEN - LETTER GENERATOR ADDRESS IN BPLOT
 1040 8720

LIMIT - PLOTTER LIMIT CHECK FLAG 1=SET
 3590 3600 7070 8860 8870 8880 8890

LL - LENGTH OF TEXT STRING
 4560 4570 4580 8030 8030 8030

LX - X LOCATION FOR LIMIT CHECK
 1020 3020 3030 4010 4040 4040 6080 6140 6140 6140 7060 7060 7060
 7060 7060 7060 7070 7070 8620 8620 8860 8860 8860 8860 8860

LY - Y LOCATION FOR LIMIT CHECK
 1020 3020 3030 4010 4040 4040 6080 6140 6140 6140 7060 7060 7060
 7060 7060 7060 7070 7070 8630 8630 8870 8870 8870 8870 8870

MID - INPUT CHARACTER BUFFER FROM DISK
 4070 4070

MM - CLEAR THE GARBAGE VARIABLE
 3280 5340 12110

N - FONT NUMBER INPUT
 5050 5050 5050 5050 5050 5070 5070 5070 5070 5070 5070 5070 5070
 5090 5090 5110 5110 5110 5110 5110 5140 5150 5160 5170 5180 5190
 5250 5290 5300 5300

N# - FONT NAME / FILE NAME
 5050 5070 5090 5110 5200 6010 6020 6030 6040 6110 6130 12080

12090 14230 14230 14250

N#(*) - FONT NAME ARRAY

3530 5130 5130 5140 5150 5160 5170 5190 5190 5200 5200 5210 5230
5240 5260 5270 5320 6130

O# - OLD FILE NAME

6020 6130 6510 6520 6530 6540 6550 6600

OF - ORIGIN FLAG 1=SET

3230 3540 3540

OS# - OLD TEXT STRING

8030 8030 8030 8030 8040 8040 8570 8830 9050 16140

OUT - OUTPUT ROUTINE FOR BPL0T

1040 3610 7160 7190 8750

P - ASC (P#)

8220 8220 8220 8530 8530 8540 8550 9030 9030

P# - HERSEY CHARACTER NUMBER AS STRING

8220 8220 8220 8510 8520 9060

PC - PEN CONTROL ADDRESS IN BPL0T

1050 8720

PI - 3.14159

2100 3190 3410 3660 3680 3680 3680 3680 3680 3680 3680 13020 13140
13140

PF - PEN POSITION FLAG 0=UP 1=DOWN

3250 3270 3570 3610 3610 3620 3620 3680 3680 3680 3680 4040 6140
7160 7170 8610 8670 8670 8720 15030 15070 15070

PX - PLOTTER X DEFLECTION

3570 4040 6140 7020 7030 7030 7030 7030 7050 7050 7060 7060 7070
7090 7100 7100 15020 15040 15040 15050 15050 15060 15060 15070

PY - PLOTTER Y DEFLECTION

3570 4040 6140 7020 7040 7040 7040 7040 7050 7050 7060 7060 7070
7090 7110 7110 15020 15040 15040 15050 15050 15060 15060 15070

Q# - STRING MAINLY FOR QUESTION ANSWER INPUT

160 160 170 530 1020 2200 2200 4040 4040 4040 4040 4060 6120
6120 8840 8840 8840 8850 8900 8900 8910

R - ORIGIN(HOME) NUMBER

2100 3540 3540 3540 3540 3570 3570 3570 3570 3570 3580 3600 3600
3600 3600 4010 6020 6080

R# - RADIAL DEFLECTION STRING

3010 3330 8030 8030 8030 8030 8030 8040 8060 8060 8100 8100 8150
8180 8210 8220 8220 8510 8580 8830 8900 10020 10030 10040 10050
11010 11010 11010 11020 16140

RHD - VAL(R#)

3010 3120 3120 3120 3130 3130 3550 3670 3670 3680 3680 3680 3690
3690 3690 3720 10020 10020 10020 10030 10030 10030 10040 10040
10040 10050 10050 10050 11010 13030 13030 13030 13130 13140 13140

13170 13170

SC - SPECIAL CHARACTER FLAG 1=SET

9020 9060 9080 9090 9100

SF - SAVE FLAG 1=SET

2200 3010 3020 3520 3530 3710 3720 4130 4180 8170 8620 8630

SIZE - SIZE OF HERSHEY CHARACTER

1060 8750

T# - TEXT INPUT STRING CHARACTER

9050 9050 9060 9060 9060 9070 9070 9070 9070 9100 9100

TC# - TEMPORARY CHARACTER FOR SYNTAX CHECK

20010 20020 20020 20020 20020 20020 20020 20020 20020 20020 20020

TEMP - TEMPORARY VARIABLE

9160 13140 13140

TIME - OUTPUT DELAY TIME IN BPLOT

1060 1080 7160 7160 7170 7170 8740 8750

TX - TEMPORARY X

7050 15020 15030 15040 15040 15050 15050 15060 15060 15070 15070

TY - TEMPORARY Y

7050 15020 15030 15040 15040 15050 15050 15060 15060 15070 15070

V(*) - VERT. HOME ARRAY V(R)

3540 3570 3600 4020 4090

VGEN - VECTOR GENERATOR ADDRESS IN BPLOT

1040 7140

VH - VERTICAL HIGH BYTE

1070 7140

VL - VERTICAL LOW BYTE

1070 7140

VV - LOOP INDEX

3020

W - WIDTH OF TEXT STRING

9120 9150 9770 9770 9900 9900 9910 9940 9940 9970 9970

WF - WORD FLAG 1=SET

9170 9590 9590

X - HORIZONTAL PLOTTER VARIABLE

3550 3550 3550 3560 3560 3590 3590 3600 3610 3620 3670 3690 3690
3690 7020 7070 10010 10020 10040

X#(*) - HERSHEY CHARACTER ARRAY

1030 5010 5290 8620 8620 8630 8630 8640 8650 8660 8670 8770

XD - X DEFLECTION IN PLOTTER UNITS

3540 3540 3570 3570 3600 4010 4040 4080 7090 7090 9120 9120

XH - X HIGH BYTE
7100 7100 7120 7120 7120 7140

XL - X LOW BYTE
7100 7140

XVEC - HORIZONTAL OUTPUT VECTOR
1050 8720

XX - PLOTTER CONVERSION FACTOR 200=1 INCH
1080 3130 3220 3240 7020 7020 8810 8810 8820 13090 13100

XYF - XYFLAG 1=SET
3550 3550 10010 10020 12020 12020

Y - VERTICAL PLOTTER DEFLECTION
3550 3550 3550 3560 3560 3590 3590 3600 3610 3620 3670 3680 3690
3690 7020 7070 10010 10030 10050 12020

YD - Y DEFLECTION IN PLOTTER UNITS
3540 3540 3570 3570 3600 4010 4040 6080 7080 7090 8630 8630

YH - VERTICAL HIGH BYTE
7110 7110 7130 7130 7130 7140

YL - VERTICAL LOW BYTE
7110 7140

YVEC - VERTICAL OUTPUT VECTOR
1050 8720

Z - INSTRUCTION NUMBER TO EDIT
12030 16110 16120 16140 16150

ZZ - LOOP INDEX
12040 12040 13010 13140 13150 13150 16050 16080 16080 16080 16090
16090 16260 16260 16260 16260

END OF VAR. LIST

LISTING 4.
FILENAME:SOAP

INSTR. NO.0	D
INSTR. NO.1	E
INSTR. NO.2	F
INSTR. NO.3	INDEX NUMBERS
INSTR. NO.4	INDEX CAPS
INSTR. NO.5	INDEX SMALL
INSTR. NO.6	INDEX GREEK
INSTR. NO.7	MATH SYMBOLS
INSTR. NO.8	PUX3Y3
INSTR. NO.9	O
INSTR. NO.10	PUCU
INSTR. NO.11	@12.35
INSTR. NO.12	1E
INSTR. NO.13	PD3D
INSTR. NO.14	PUCU
INSTR. NO.15	PD3U
INSTR. NO.16	H
INSTR. NO.17	1D
INSTR. NO.18	@12.25*180*360
INSTR. NO.19	*@12.5 0*180
INSTR. NO.20	PV1.5U
INSTR. NO.21	.32L
INSTR. NO.22	WEOAP40
INSTR. NO.23	H

LISTING 5.

FILENAME: SCUP

INSTR. NO.0	O
INSTR. NO.1	O
INSTR. NO.2	F
INSTR. NO.3	INDEX NUMBERS
INSTR. NO.4	INDEX CAPS
INSTR. NO.5	INDEX SMALL
INSTR. NO.6	INDEX GREEK
INSTR. NO.7	MATH SYMBOLS
INSTR. NO.8	PUX7Y3
INSTR. NO.9	O
INSTR. NO.10	FU2U
INSTR. NO.11	G1G.35
INSTR. NO.12	1R
INSTR. NO.13	PD2D
INSTR. NO.14	FU2L
INSTR. NO.15	PD3U
INSTR. NO.16	H
INSTR. NO.17	1D
INSTR. NO.18	G1G.35^180^240
INSTR. NO.19	*G1G.35^0^180
INSTR. NO.20	PU1.5U
INSTR. NO.21	.42L
INSTR. NO.22	WSCUP40
INSTR. NO.23	H
INSTR. NO.24	X1Y.5
INSTR. NO.25	PD6R
INSTR. NO.26	6IU
INSTR. NO.27	8L
INSTR. NO.28	6D
INSTR. NO.29	PU4R
INSTR. NO.30	PD6U
INSTR. NO.31	PUX-2Y-5.5
INSTR. NO.32	O
INSTR. NO.33	.34L
INSTR. NO.34	W(A)40
INSTR. NO.35	H
INSTR. NO.36	4R
INSTR. NO.37	.35L
INSTR. NO.38	W(B)40
INSTR. NO.39	H

LISTING 6.

```

0010 ; ROUTINE TO OUTPUT
0020 ; ONE CHAR. OR CHAR.
0030 ; STRING STORED IN
0040 ; TABLE AT SOUT($8600)
0050 WAIT EQU FCA8
0055 ; MONITOR DELAY ROUTINE
0060 SERIO EQU C200
0065 ; SERIAL CARD IN SLOT #2
0070 OBJ 83B8
0080 ORG 83B8
0085 ; *****
0087 ;
83B8 AE0984 0090 PLOT LDX SIZE
0095 ; START:LOAD SIZE IN X REG
83BB F025 0100 BEQ RESTOR
0105 ; IF ZERO RESTORE POINTERS
0107 ; AND RETURN
83BD AD0086 0110 FETCH LDA OUT
0115 ; GET NEXT CHARACTER
0117 ; TABLE STARTS AT $8600
83C0 C978 0120 CMP #$78
0125 ; IS IT "X"?
83C2 F01E 0130 BEQ RESTOR
0135 ; YES THEN RETURN
83C4 1016 0140 BPL PENCON
0145 ; IS IT "Y" OR "Z"?
0146 ; IF YES THEN PENCONTROL
83C6 6D0A84 0150 ADC ANGLE
0155 ; HOR. OR VERT. CHAR?
83C9 2977 0160 AND #$77
0165 ; ROTATE TO PROPER ORIEN.
83CB 20F583 0170 JSR PUTOUT
83CE CA 0180 DEX
0185 ; FINISHED?
83CF D0EC 0190 BNE FETCH
0195 ; NO. DO IT AGAIN
83D1 EEBE83 0200 MOVEUP INC FETCH+01
0205 ; GET NEXT CHARACTER
83D4 D0E2 0210 BNE PLOT
83D6 EEBF83 0220 INC FETCH+02
83D9 4CB883 0230 JMP PLOT
0235 ; *****
83DC 20F583 0240 PENCON JSR PUTOUT
0245 ; PUT OUT PEN CONTROL
83DF 4CD183 0250 JMP MOVEUP
0255 ; GET NEXT CHARACTER
0257 ; *****
83E2 AD0784 0260 RESTOR LDA HOOK+00
0265 ; RESET PTR. TO $8600
83E5 8DBE83 0270 STA FETCH+01
83E8 8D7985 0280 STA STORE+01
83EB AD0884 0290 LDA HOOK+01
83EE 8DBF83 0300 STA FETCH+02
83F1 8D7A85 0310 STA STORE+02
83F4 60 0320 RTS
0325 ; RETURN TO CALLING PGM.
0326 ; *****

```


		0327	; *****
		0328	; PUTOUT CHAR ON RS232
83F5	A8	0330	PUTOUT TAY
		0335	; SAVE ACCUMULATOR
83F6	2000C2	0340	JSR SERIO
		0345	; OUTPUT TO SLOT #2
83F9	AD0C84	0350	LDA DELAY
		0355	; LOAD DELAY (0-255)
83FC	20A8FC	0360	JSR WAIT
		0365	; MONITOR WAIT (\$FCAB)
83FF	98	0370	TYA
		0375	; RESTORE ACCUMULATOR
8400	60	0380	RTS
		0385	; *****
8401	AD0B24	0390	ONECHR LDA CHAR
		0395	; ONE CHAR.OUTPUT ROUTINE
8404	4CF583	0400	JMP PUTOUT
		0402	; DEFINE SOME VARIABLES
		0405	; *****
8407	0086	0410	HOOK EQU OUT
8409	01	0420	SIZE DFD 01
840A	00	0430	ANGLE DFD 00
840B	78	0440	CHAR DFD 78
840C	FF	0450	DELAY DFD FF
		0455	; *****
		0460	; LETTER GENERATOR
		0470	; FOR HERSHEY VECTORS
		0480	PEN EQU 78
		0485	; START WITH "X"
		0490	XVEC EQU 00
		0500	YVEC EQU 00
		0505	; START WITH X=Y=0
840D	A9FF	0510	LETTER LDA #\$FF
840F	8DE385	0520	STA FH
		0525	; INITIALIZE F
8412	A978	0530	LDA #PEN
8414	207885	0540	JSR STORE
		0545	; STORE PEN CONT. CHAR.
8417	A900	0550	XIN LDA #XVEC
		0555	; CALC. & STORE XL & XH
8419	8DD885	0560	STA XL
841C	3008	0570	BMI XNEG
841E	A900	0580	LDA #\$0C
8420	8DD985	0590	STA XH
8423	4C2B84	0600	JMP YIN
8426	A9FF	0610	XNEG LDA #\$FF
8428	8DD985	0620	STA XH
842B	A900	0630	YIN LDA #YVEC
		0635	; CALC. & STORE YL & YH
842D	8DD685	0640	STA INPUT
8430	3008	0650	BMI YNEG
8432	A900	0660	LDA #\$00
8434	8DD785	0670	STA YH
8437	4C3F84	0680	JMP GO
		0685	; JMP TO VGEN ROUTINE
843A	A9FF	0690	YNEG LDA #\$FF
843C	8DD785	0700	STA YH
		0705	; FALL INTO VGEN RTN.
		0706	; *****

```

0707 ; *****
0710 ; VECTOR GENERATOR SUBRTN
0720 ; CALL BOUT @$8440
0730 ; DBL.PREC.ON LOC. $85D6-
0740 ; $85D9 YL-YH-XL-XH
0745 ; AS STRING $85EF-$85FD
843F 18 0750 GO CLC
8440 A200 0760 LDX #$00
8442 E004 0770 GOLOOP CPX #$04
0775 ; GET X AND Y VALUES
8444 F030 0780 BEQ FEQ
8446 E8 0790 INX
8447 BDD685 0800 LDA INPUT,X
844A 300F 0810 BMI COMPL
844C 9DDA85 0820 STA INPUT+04,X
844F CA 0830 DEX
8450 BDD685 0840 LDA INPUT,X
8453 9DDA85 0850 STA INPUT+04,X
8456 E8 0860 INX
8457 E8 0870 INX
8458 4C4284 0880 JMP GOLOOP
845B CA 0890 COMPL DEX
0895 ; CALC /X/ AND /Y/
845C BDD685 0900 LDA INPUT,X
845F 49FF 0910 EOR #$FF
8461 6901 0920 ADC #$01
8463 9DDA85 0930 STA INPUT+04,X
8466 E8 0940 INX
8467 BDD685 0950 LDA INPUT,X
846A 49FF 0960 EOR #$FF
846C 6900 0970 ADC #$00
846E 9DDA85 0980 STA INPUT+04,X
8471 18 0990 CLC
8472 E8 1000 INX
8473 4C4284 1010 JMP GOLOOP
1015 ; HILOT BASIC RTN.
8476 A204 1020 FEQ LDX #$04
8478 A006 1030 LDY #$06
847A 209985 1040 JSR ADD
847D A208 1050 LDX #$08
847F A00C 1060 LDY #$0C
8481 20C785 1070 JSR MOVE
8484 D009 1080 BNE DEQ
8486 CA 1090 DEX
8487 BDD685 1100 LDA INPUT,X
848A D003 1110 BNE DEQ
848C 4C9385 1120 JMP QUIT
848F A204 1130 DEQ LDX #$04
8491 A006 1140 LDY #$06
8493 20B085 1150 JSR SUB
8496 A208 1160 LDX #$08
8498 A00E 1170 LDY #$0E
849A 20C785 1180 JSR MOVE
1185

```

849D	A900	1190	IEQ0	LDA #\$00
849F	8DE085	1200		STA INPUT+0A
84A2	ADD785	1210	YTEST	LDA INPUT+01
84A5	3005	1220		BMI TSUM
84A7	A902	1230	IEQ2	LDA #\$02
84A9	8DE085	1240		STA INPUT+0A
84AC	A200	1250	TSUM	LDX #\$00
84AE	A002	1260		LDY #\$02
84B0	209985	1270		JSR ADD
84B3	3006	1280	TTEST	BMI TDIFF
84B5	EEE085	1290	IPLUS2	INC INPUT+0A
84B8	EEE085	1300		INC INPUT+0A
84BB	A200	1310	TDIFF	LDX #\$00
84BD	A002	1320		LDY #\$02
84BF	20B085	1330		JSR SUB
84C2	3006	1340	TESTT	BMI XTEST
84C4	EEE085	1350	TWOPI	INC INPUT+0A
84C7	EEE085	1360		INC INPUT+0A
84CA	ADD985	1370	XTEST	LDA INPUT+03
84CD	300E	1380		BMI IPT
84CF	ADE085	1390	ATEMI	LDA INPUT+0A
84D2	49FF	1400		EOR #\$FF
84D4	6909	1410		ADC #\$09
84D6	3DE085	1420		STA INPUT+0A
84D9	18	1430		CLC
84DA	4CE584	1440		JMP DTEST
84DD	ADE085	1450	IPT	LDA INPUT+0A
84E0	690A	1460		ADC #\$0A
84E2	8DE085	1470		STA INPUT+0A
84E5	ADE585	1480	DTEST	LDA INPUT+0F
84E8	301E	1490		BMI TABSY
84EA	A206	1500	TABSX	LDX #\$06
84EC	A010	1510		LDY #\$10
84EE	20C785	1520		JSR MOVE
84F1	ADE485	1530	DMIND	LDA INPUT+0E
84F4	49FF	1540		EOR #\$FF
84F6	6901	1550		ADC #\$01
84F8	8DE485	1560		STA INPUT+0E
84FB	ADE585	1570		LDA INPUT+0F
84FE	49FF	1580		EOR #\$FF
8500	6900	1590		ADC #\$00
8502	8DE585	1600		STA INPUT+0F
8505	4C0F85	1610		JMP EEQ
8508	A204	1620	TABSY	LDX #\$04
850A	A010	1630		LDY #\$10
850C	20C785	1640		JSR MOVE
850F	A900	1650	EEQ	LDA #\$00
8511	8DE885	1660		STA INPUT+12
8514	8DE985	1670		STA INPUT+13
		1675		;

8517	A20E	1680	ZEQ	LDX #\$0E
8519	A010	1690		LDY #\$10
851B	209985	1700		JSR ADD
851E	A208	1710		LDX #\$08
8520	A012	1720		LDY #\$12
8522	209985	1730		JSR ADD
8525	A208	1740		LDX #\$08
8527	A012	1750		LDY #\$12
8529	209985	1760		JSR ADD
852C	3026	1770	ZTEST	BMI EPLUST
852E	A20E	1780	EPLUSD	LDX #\$0E
8530	A012	1790		LDY #\$12
8532	209985	1800		JSR ADD
8535	A208	1810		LDX #\$08
8537	A012	1820		LDY #\$12
8539	20C785	1830		JSR MOVE
853C	A20C	1840	FMIN2	LDX #\$0C
853E	A014	1850		LDY #\$14
8540	20B085	1860		JSR SUB
8543	A208	1870		LDX #\$08
8545	A00C	1880		LDY #\$0C
8547	20C785	1890		JSR MOVE
854A	AEE085	1900	ALI	LDX INPUT+0A
		1905	;	GET "P","Q",ETC.
854D	CA	1910		DEX
854E	BDEE85	1920		LDA VECTOR,X
8551	4C7885	1930		JMP STORE
8554	A210	1940	EPLUST	LDX #\$10
8556	A012	1950		LDY #\$12
8558	209985	1960		JSR ADD
855B	A208	1970		LDX #\$08
855D	A012	1980		LDY #\$12
855F	20C785	1990		JSR MOVE
8562	A20C	2000		LDX #\$0C
8564	A016	2010		LDY #\$16
8566	20B085	2020		JSR SUB
8569	A208	2030		LDX #\$08
856B	A00C	2040		LDY #\$0C
856D	20C785	2050		JSR MOVE
8570	AEE085	2060	AMI	LDX INPUT+0A
		2065	;	GET "P","Q",ETC.
8573	CA	2070		DEX
8574	CA	2080		DEX
8575	BDEE85	2090		LDA VECTOR,X
8578	8D0086	2100	STORE	STA OUT
		2105	;	PUT CHAR. IN TABLE
857B	C978	2110		CMP #\$78
857D	F019	2120		BEQ RETURN
		2121	;	

		2125	;	IF "X" THEN RETURN
857F	EE7985	2130		INC STORE+01
		2135	;	INC. TABLE LOC.
8582	D003	2140		BNE TESTF
8584	EE7A85	2150		INC STORE+02
8587	ADE385	2160	TESTF	LDA INPUT+0D
858A	3007	2170		BMI QUIT
858C	D089	2180		BNE ZEQL
858E	ADE285	2190		LDA INPUT+0C
8591	D084	2200		BNE ZEQL
8593	A978	2210	QUIT	LDA #\$78
8595	4C7885	2220		JMP STORE
8598	60	2230		RETURN RTS
		2232	;	*****
		2235	;	DOUBLE PREC. RTNS.
8599	18	2240	ADD	CLC
859A	BDD685	2250		LDA INPUT,X
859D	79D685	2260		ADC INPUT,Y
85A0	8DDE85	2270		STA INPUT+08
85A3	E8	2280		INX
85A4	C8	2290		INY
85A5	BDD685	2300		LDA INPUT,X
85A8	79D685	2310		ADC INPUT,Y
85AB	8DDF85	2320		STA INPUT+09
85AE	18	2330		CLC
85AF	60	2340		RTS
85B0	38	2350	SUB	SEC
85B1	BDD685	2360		LDA INPUT,X
85B4	F9D685	2370		SBC INPUT,Y
85B7	8DDE85	2380		STA INPUT+08
85BA	E8	2390		INX
85BB	C8	2400		INY
85BC	BDD685	2410		LDA INPUT,X
85BF	F9D685	2420		SBC INPUT,Y
85C2	8DDF85	2430		STA INPUT+09
85C5	18	2440		CLC
85C6	60	2450		RTS
85C7	BDD685	2460	MOVE	LDA INPUT,X
85CA	99D685	2470		STA INPUT,Y
85CD	E8	2480		INX
85CE	C8	2490		INY
85CF	BDD685	2500		LDA INPUT,X
85D2	99D685	2510		STA INPUT,Y
85D5	60	2520		RTS
		2521	;	


```

2525 ; *****
2527 ; VARIABLE STORAGE
85D6 00 2530 INPUT DFD 00
2535 ; INPUT = YL
85D7 00 2540 YH DFD 00
85D8 00 2550 XL DFD 00
85D9 00 2560 XH DFD 00
85DA 00 2570 AYL DFD 00
85DB 00 2580 AYH DFD 00
85DC 00 2590 AXL DFD 00
85DD 00 2600 AXH DFD 00
85DE 00 2610 RESL DFD 00
85DF 00 2620 RESH DFD 00
85E0 00 2630 I DFD 00
85E1 00 2640 DFD 00
85E2 00 2650 FL DFD 00
85E3 00 2660 FH DFD 00
85E4 00 2670 DL DFD 00
85E5 00 2680 DH DFD 00
85E6 00 2690 TL DFD 00
85E7 00 2700 TH DFD 00
85E8 00 2710 EL DFD 00
85E9 00 2720 EH DFD 00
85EA 02 2730 TWO DFD 02
85EB 00 2740 DFD 00
85EC 01 2750 ONE DFD 01
85ED 00 2760 DFD 00
85EE 70 2770 VECTOR DFD 70
2775 ; VECTOR = "P"
85EF 71 2780 Q DFD 71
85F0 72 2790 R DFD 72
85F1 71 2800 Q2 DFD 71
85F2 72 2810 R2 DFD 72
85F3 73 2820 S DFD 73
85F4 74 2830 T DFD 74
85F5 73 2840 S2 DFD 73
85F6 74 2850 T2 DFD 74
85F7 75 2860 U DFD 75
85F8 76 2870 V DFD 76
85F9 75 2880 U2 DFD 75
85FA 76 2890 V2 DFD 76
85FB 77 2900 W DFD 77
85FC 70 2910 P2 DFD 70
85FD 77 2920 W2 DFD 77
85FE 00 2930 DFD 00
85FF 00 2940 DFD 00
8600 00 2950 OUT DFD 00
2955 ; *****
2956 ;

```

LISTING 7.

Vector generator BASIC routine to generate the best straight line between two points. From Houston Instrument HiPlot Manual. Comments indicate the correspondance between this program and locations in the 6502 assembly language program BPLOT.

```

2   INPUT X,Y ; REM Input YL, YH, XL, XH
3   PRINT "z" ; REM Output PENCON if pen down;
4   GOSUB 100 ; REM GO
5   PRINT "y" ; REM Output PENCON if pen up;
100  REM ***** VECTOR GENERATOR SUBROUTINE*****; REM GO
110  REM THIS SUBROUTINE DRAWS THE BEST STRAIGHT LINE FOR A
120  REM COORDINATE CHANGE OF (X) AND (Y)
130  F=ABS(X)+ABS(Y) ; REM FEQ
140  IF F=0 THEN 470 ; REM JMP QUIT
150  D=ABS(Y)-ABS(X) ; REM DEQ
160  DIM A$(16) ; REM VECTOR
170  A$="pqrgrststuvuvwpw" ; REM VECTOR,Q,R,Q2,R2,S,T,S2,T2,U,V,
                                U2,V2,W,P2,W2

180  I=0 ; REM IEQ0
190  IF Y<0 THEN 210 ; REM YTEST
200  I=2 ; REM ITEST
210  T=X+Y ; REM TSUM
220  IF T<0 THEN 240 ; REM TTEST
230  I=I+2 ; REM IPLUS2
240  T=Y-X ; REM TDIFF
250  IF T < 0 THEN 270 ; REM TESTT
260  I=I+2 ; REM TWOPI
270  IF X < 0 THEN 300 ; REM XTEST
280  I=8-I ; REM ATEMI
290  GOTO 310 ; REM JMP DTEST
300  I=I+10 ; REM IPT
310  IF D < 0 THEN 350 ; REM DTEST
320  T=ABS(X) ; REM TABSX
330  D=-D ; REM DMIND
340  GOTO 360 ; REM JMP EEQ
350  T=ABS(Y) ; REM TABSY
360  E=0 ; REM EEO
370  Z=T+D+E+E ; REM ZEQ
380  IF Z < 0 THEN 430 ; REM ZTEST
390  E=E+D ; REM EPLUS D
400  F=F-2 ; REM FMIN2
410  PRINT USING "#,1A";A$(I,I) ; REM AII
420  GOTO 460 ; REM JMP STORE (INSTEAD OF OUTPUTTING CHARACTER,
                                STORE IT IN OUT TABLE AT $8600+)
430  E=E+T ; REM EPLUST
440  F=F-1 ; REM LDX #50C
450  PRINT USING "#,1A";A$((I-1),(I-1)) ; REM JMP STORE
460  IF F>0 THEN 370 ; REM TESTF
470  RETURN ; REM QUIT (STORE "x" TO MARK END OF TABLE,RETURN)

```

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET <i>(See instructions)</i>	1. PUBLICATION OR REPORT NO. NBS TN 1184	2. Performing Organ. Report No.	3. Publication Date January 1984
4. TITLE AND SUBTITLE Utility Programs for Producing Camera Ready Illustrations on a Microcomputer and a Laboratory Plotter			
5. AUTHOR(S) C. E. Dick			
6. PERFORMING ORGANIZATION <i>(If joint or other than NBS, see instructions)</i> NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		7. Contract/Grant No.	8. Type of Report & Period Covered Final
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS <i>(Street, City, State, ZIP)</i> Same as in item 6 above.			
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
11. ABSTRACT <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i> A collection of software routines is described that allows the user to prepare camera ready illustrations in the laboratory or office environment. These routines are written in APPLESOFT BASIC and 6502 assembly code for the Apple II microcomputer which is interfaced with an inexpensive digital plotter. Provisions are made to draw figures composed of straight and curved line segments, letter the figures with a variety of graphic arts fonts, and save the figures on disk for later plotting or revision.			
12. KEY WORDS <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> Apple II computer; camera ready illustrations; digital plotter; graphic arts; microcomputer.			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input checked="" type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 59 15. Price

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic \$18; foreign \$22.50. Single copy, \$5.50 domestic; \$6.90 foreign.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the above NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.

Order the following NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards

Washington, D.C. 20234
Official Business
Penalty for Private Use \$300



POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

SPECIAL FOURTH-CLASS RATE
BOOK