Teknіques

The 4051 Applications Library Newsletter Vol. 2 No. 3

4051 Graphic System is the center of attention as it computes frequency response curves for AR brand speakers. Pictured is FRED'S SOUND OF MUSIC display at recent "Stereo Show and Tell" in Portland, Oregon.

4051 Computes Best Speaker Placement in Traveling Stereo Roadshow

by Gary P. Laroff

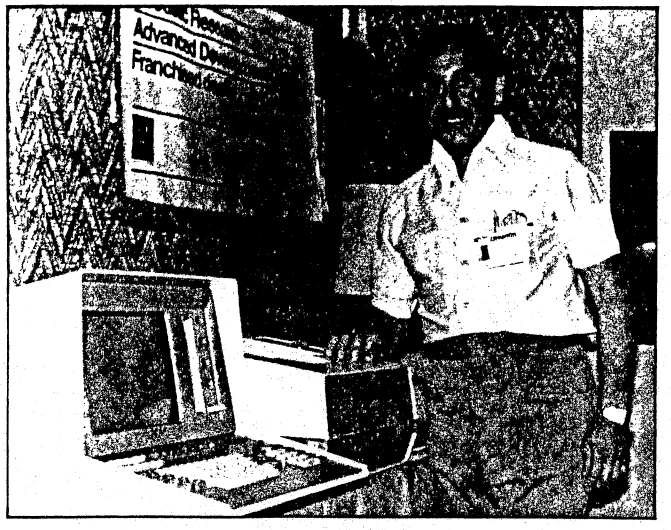
"My room is 16 ft. x 12 ft. x 8 ft. high with walls made of marshmallow. Where do I put my speakers? Where should I sit? How much power will I need?..."

These and other questions are being answered at the TELEDYNE ACOUSTIC RESEARCH (AR) "Science and Sound" show traveling throughout the United States and Europe. And the answers are generated by Tektronix 4051 Graphic Systems with Tektronix 4631 Hard Copy Units attached.

Laboratory Comes to the Retail Floor

Calculations and data that formerly took weeks to work out can be drawn from the internal magnetic tape and memory almost instantly. AR's 4051 can then provide data on ideal listening locations from room dimensions and speaker placement supplied by the customer. It shows you where you should *really* sit in your listening area. The AR 4051 is programmed to provide tweeter polar dispersion curves by simply supplying it with high frequency driver diameters. The 4051 does all of the mathematics in a matter of seconds and plots the curves while the customer watches. Hard copies of the screen

In This Issue



"Just describe your room and I'll show you the best position for listening to any speaker," says Fred Lindemann of FRED'S SOUND OF MUSIC, Portland, Oregon.

"Programs developed in our large in-house scientific computer have been transferred to the 4051", states AR's Steve Johnson, traveling companion of the 4051.

graphics are given to the customer seconds later. Performance data of various AR speaker models and their use in specific installations can be shown on the 4051 screen.

The ACOUSTIC RESEARCH Traveling Roadshow, comprised of a "stack" of AR's new speaker models, a 4051 and a 4631 Hard Copy Unit are drawing crowds wherever they go. Fred Lindemann, of FRED'S SOUND OF MUSIC in Portland, Oregon, is very enthusiastic about the response to his 4051 at the recent "Stereo Show and Tell" in Portland. Using the 4051, Fred answers questions about dispersion, power handling, listening locations, speaker placement and how speakers work.

A number of programs are demonstrated on the 4051. Some were developed on AR's large in-house scientific computer and were transferred to the 4051. Others were written especially for the 4051. Robert Berkovitz, Research Director at AR and author of the programs, is a very enthusiastic supporter of the 4051 as a personal computer and relies heavily on interactive graphics to describe acoustic phenomena.

Hard copies of typical 4051 output are shown in the accompanying figures. The programs are not presently available and are proprietary to ACOUSTIC RESEARCH of Norwood, Massachusetts.

TEKniques readers who would like to see the 4051 and AR's Roadshow and Consumer Computer Clinics are encouraged to watch newspapers, magazines and television for the next visit to their area.

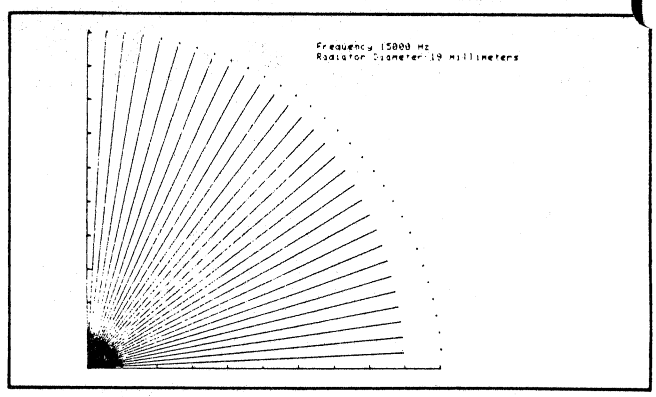


Fig. 1. Frequency response for a 19mm diameter tweeter at a frequency of 15,000 Hz.

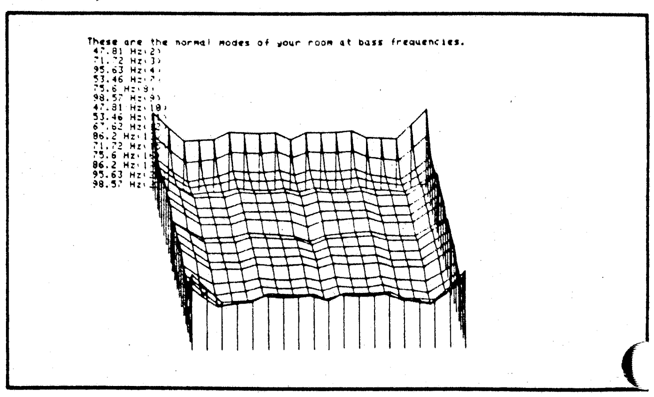


Fig. 2. Frequency response of the listener's room at 15 bass frequencies. The 4051 lists a frequency and then plots the normal modes for that frequency.

TEKniques, the 4051 Applications Library Newsletter, is published by the Information Display Group of Tektronix, Inc., Group 451, P.O. Box 500, Beaverton, Oregon 97077. It is distributed to TEKTRONIX 4051 users and members of the 4051 Application Library.

Publisher	Ken Cramer
Managing Editor	Patricia Kelley
Editor	Terence Davis
Graphic Designer	Mark Woods
Circulation	Rory Gugliotta

Volume 2 Number 3. Copyright 1978 Tektronix, Inc. Printed in the United States of America. All Rights Reserved.

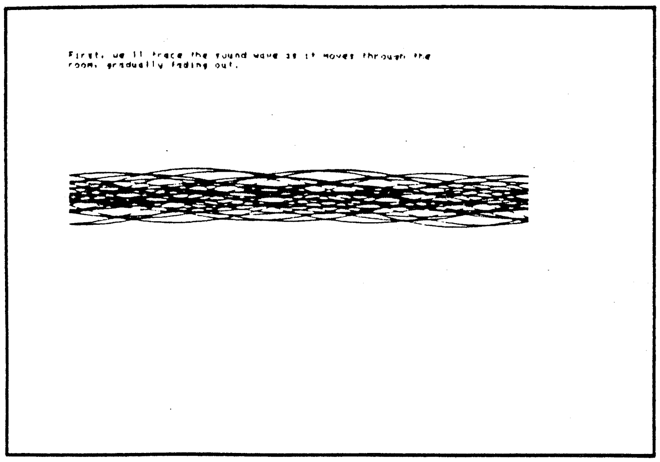


Fig. 3. How speakers work. A graphic example of one of the preliminaries: sound wave formation.

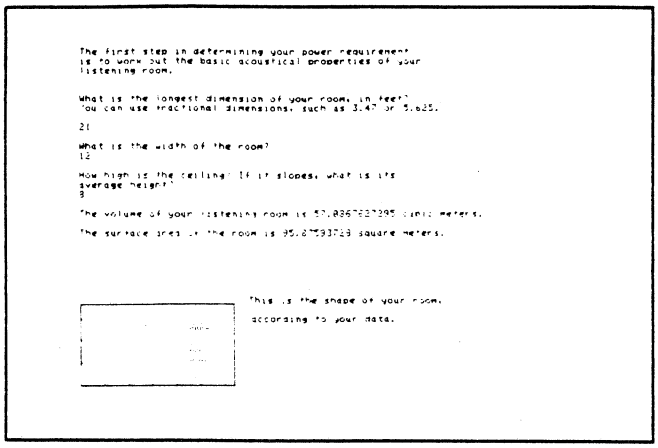


Fig. 4. Customer's room shape diagram drawn moments after the customer has supplied the data.

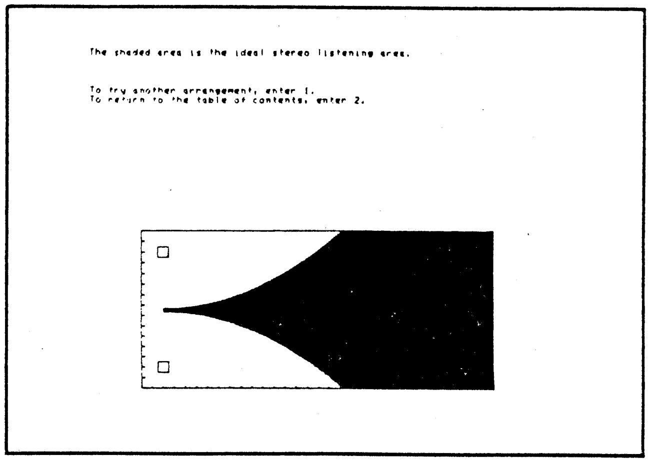


Fig. 5. Ideal stereo listening area diagram for the customer's room and speaker placement. The 4051 is programmed for rooms made of plaster, brick, sheetrock, wood, marshmallow and other exotic materials.

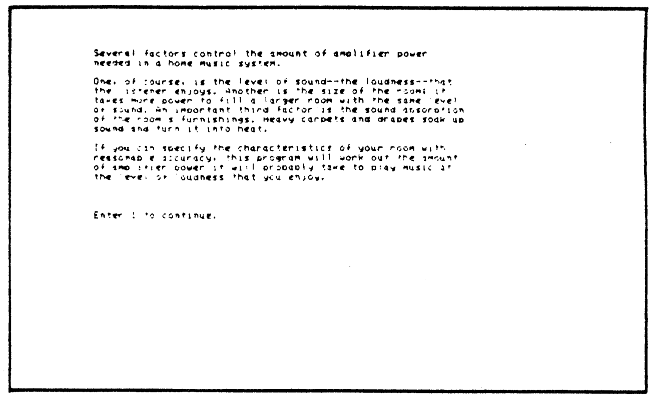


Fig. 6. Typical text on amplifier power requirements.

✻ Editor's Note

New Astronomy and Physics Demo Program

Dr. R. J. Reimann of Boise State University in Idaho has contributed a second edition of "Graphics Demonstrations for Astronomy and Physics" to the Applications Library. This second version includes a new program, "Ellipse.". Dr. Reimann states also that numerous improvements have been made in many of the old programs and that the order of presentation has been changed to a more natural sequence. The updated version allows optional use of the Tektronix 4662 Plotter. Further details are contained in Abstract Number 51/00-5401/1.

Contest Update

The deadline for entries to the computer-aided design contest was extended to April 15, 1978, to allow time for more of you to complete and enter your programs. Judging of the entries will be completed by June 1, 1978; results will be published in the next issue of TEKniques (Vol. 2 No. 5).

Catalog Reminder

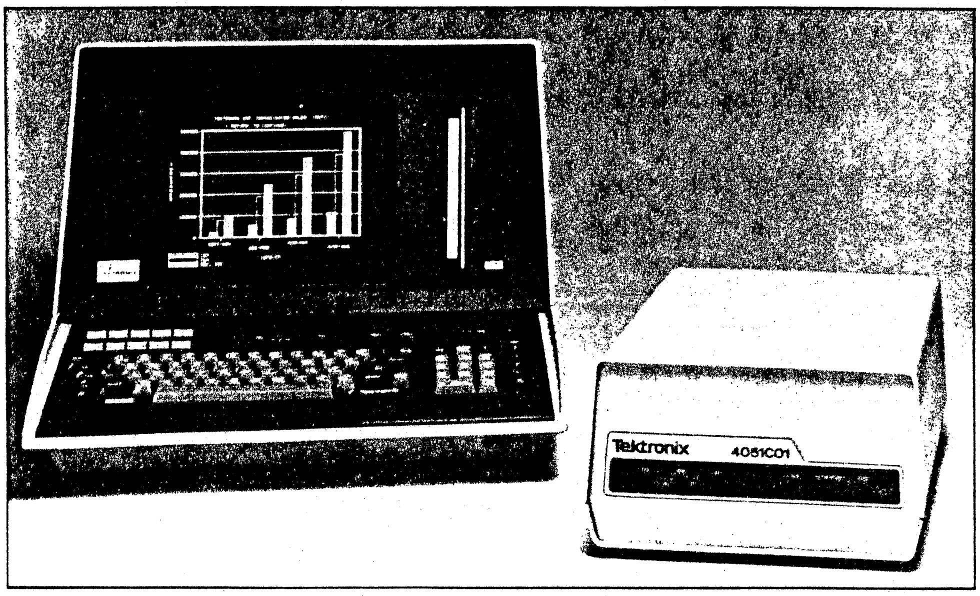
The Applications Library wishes to remind you that there is currently available, for the asking, a catalog of the programs available through the library. Catalogs can be obtained by dropping a line to the 4051 Applications Library, Tektronix, Inc., Group 451, P.O. Box 500, Beaverton, OR 97077. Or you can make a quick phone call to the library at (503) 682-3411, extension 2618.

Copyrights

There is always the temptation to duplicate favorite graphics (cartoons, drawings, etc.) when you have access to an easy-to-use Graphic System like the 4051. Please check for copyright* any programs that duplicate graphics or data from published sources before submitting them to the Applications Library. If the drawing or data is copyrighted, there must be a copyright release from the copyright holder before the program can be included in the Library or this newsletter. Copyright release for such applications can usually be obtained by sending a brief letter of explanation and request to the copyright holder.

**A copyright statement will be on the title page at the front of a book, within the masthead in a periodical or indicated by the copyright mark and year, i.e., © 1978 on the face of the graphics.*

New Interface Allows 4051 Data Communications with IBM Computers



4051C01 SYNCHRONOUS COMMUNICATIONS INTERFACE *provides 4051 users access to data bases stored on IBM computers.*

by Gary P. Laroff

The new Tektronix 4051C01 SYNCHRONOUS COMMUNICATIONS INTERFACE is an interface/peripheral for the 4051 which provides 4051 users access to data bases stored on IBM computers. In general, the 4051C01 supports IBM 2780 (BSC or BISYNC) communications protocol and acts as an intelligent interface buffering the data in both directions and controlling the line protocol independent of programs running on the 4051. Data communication is initiated from either the keyboard of the 4051, or by running BASIC language subprograms on the 4051.

Data communication between a "host computer" and the 4051 can now be performed in either asynchronous or synchronous modes. Asynchronous communication is more interactive and is handled by the 4051 Option 1 Data Communications Interface. Synchronous communication is handled by 4051C01. Transferring data asynchronously is most successful when the amount of information communicated over a phone line is "small" or when a direct line connection is provided to the host computer. Sending data without any means of error checking and recovery can be difficult when there is noise on the line or when an accidental disconnect occurs. In such cases the 4051 may continue to listen to a disconnected line, or a complete data transfer may be aborted and have to be restarted.

Where large amounts of data or program must be transferred to or from the computer, a means of providing for transmission error detection and recovery is required. This is the world of the synchronous batch terminal. Also, communication will be more economical if a higher data transfer rate is available. For the 4051, error checking/recovery and higher speeds (up to 9600 baud) are achieved with the 4051C01 Synchronous Communications Interface.

Interface Operation

The 4051C01 provides the 4051 user with the capability to communicate with IBM computer systems. IBM BINARY SYNCHRONOUS 2780 Remote Job Entry protocol is firmware implemented in this microprocessor-based intelligent interface. High speed (up to 9600 baud), error free communication, and standard system device definition are the major benefits provided. Connections to the IBM system may be either direct or with synchronous modems to an IBM ICA or 270X-/370X. 4051C01 is connected through a 4051 ROM backpack slot. An RS-232 connector and a 10 ft. cable are used to connect the 4051C01 interface to a modem or directly to a computer's communications port.

Control of and response to the 2780 line protocol, I/O buffering, and error detection and correction are performed by the interface microcomputer system. Thus the 4051C01 interface and 4051 may operate independently.

Programming the 4051C01

A major feature is that only one command is required to allow GPIB devices, such as the Tektronix 4924 Digital Cartridge Tape Unit, to send a file of data or programs via the 4051C01 to the host computer. Similarly, the host computer can dump a program or data in batch fashion to the Tektronix 4641 line printer or 4924 tape via the

4051C01. Because the 4051 is merely part of a data path, the transmitted data or programs take up none of the 4051 read/write memory.

Record and file I/O are performed through the use of BASIC subroutines residing in the ROM pack and the interface operating system.

Magnetic tape files may be transferred to or from the host computer system by positioning the tape and using one BASIC statement.

```
Using the internal drive
CALL "C01MTR" or CALL "C01MTW"

Using a 4924 assigned device number 3
CALL "C01OUT", 3 or CALL "C01INP", 3
```

Files from the host computer may be sent to the lineprinter, screen or any 4051 device number.

```
To the 4641 lineprinter left backpack slot
CALL "C01INP", 41

To the 4051 display
CALL "C01INP", 32
```

The 4051C01 may be addressed through the use of PRINT and INPUT statements and device numbers 49 or 59 depending upon which 4051 backpack slot contains the interface ROM pack.

```
A$ is a string variable containing the desired record

To read a record
   INPUT @49: A$

To transmit a record
   PRINT @49: A$
```

The status of the interface can also be examined by the programmer.

Five utility subroutines are provided for parameter initialization, interface status, and troubleshooting.

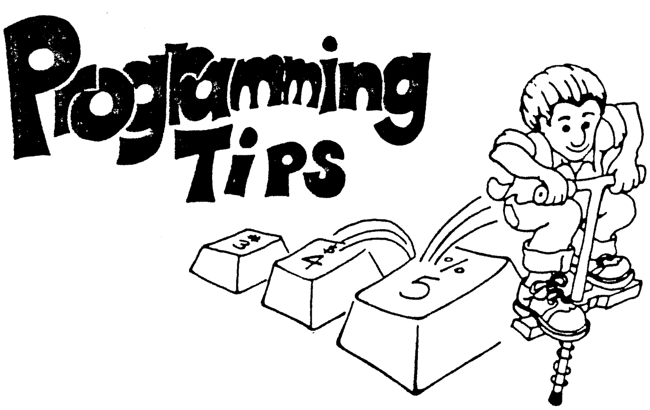
C01INIT	establishes default parameters and allows definition of nine communications parameters
C01PRLIST	lists communications parameters currently in effect
C01STAT	returns status parameters as variables
C01DIAG	initiates a complete test of the interface system and returns diagnositc indicators
C01TEST	initiates the self-test, also routinely performed at power up of the system

New 4051 System Capabilities

In addition to higher data transfer rates and error recovery, 4051C01 allows access to host computers which do not support asynchronous terminals. Batch data and program transfers are now possible between remote 4051 systems.

Although the 4051 with the new 4051C01 interface is not a direct substitute for an expensive high-powered Remote Job Entry (RJE) system such as an IBM 2780, the 4051/4051C01 is a powerful hardware configuration. By adding a Tektronix 4641 Matrix Printer, 4924 Digital Cartridge Tape Unit, or GPIB paper tape reader/punch such as the FACIT models 4070/4021 (discussed in TEKniques Vol. 1 No. 7) the 4051 becomes the center of a distributed processing batch-oriented terminal/personal computing system.

For more information on the new 4051C01 Synchronous Communications Interface, contact your local Tektronix Sales Engineer.



Dashed Line Algorithm Update

by Syed M. Hasan
Troy, Michigan

Regarding the enhanced version of the Dashed Line Algorithm suggested by Ken Cramer in TEKniques Vol. 2 No. 1, page 9: Problems can be encountered when drawing a vertical line downwards. The program can iterate indefinitely in the FOR/NEXT loop in lines 1210 to 1230. In the case of the downward vertical line, X1 is equal to X2, and statement 1200 reduces to this:

```
1200 FOR I=0 TO 0 STEP 0
```

This is an infinite iteration. The probelm can be overcome by using the following technique:

```
1195 H=2*(ABS(U1)MAX ABS(U2))
1200 FOR I=0 TO ABS(X2-X1-U1*2) MAX ABS(Y2-Y1-U2*2) STEP H
```

3-D Line Clipping

by Jay Beck

When drawing three-dimensional objects on the 4051, it is necessary to perform your own line clipping to remove line segments that are outside the field of view defined by a 3-D window. This article describes a simple but very capable 3-D line clipping algorithm.

```
10000 REM   -------------------------------------------------------------
10010 REM     3 D   C L I P P I N G
10020 REM     Main Program
10030 REM   -------------------------------------------------------------
10040 PAGE
10050 INIT
10060 DIM X0(24),Y0(24),Z0(24),L0(7,5)
10070 DIM T(4,4),M(4,4),D(4,4),N(6,3),F(6,3)
10080 REM------------------------------X,Y,Z coordinates
10090 DATA 0,0,0,0.5,1,0,1,0,0,1,0,0.5
10100 DATA 1,0,1,0.5,0,1,0,0,1,0,0.5
10110 DATA 0,0.5,0,0.5,0.5,1,1,0,0,1,0.5,0.5
10120 DATA 1,0.5,1,0.5,0.5,1,0,0.5,1,0,0.5,0.5
10130 DATA 0,1,0,0.5,1,0,1,0,1,1,0.5
10140 DATA 1,1,1,0.5,1,1,0,1,1,1,0.5
10150 REM------------------------------Line Descriptors
10160 DATA 1,3,5,7,1
10170 DATA 9,11,13,15,9
10180 DATA 17,19,21,23,17
10190 DATA 1,17,23,7,1
10200 DATA 2,18,22,6,2
10210 DATA 3,19,21,5,3
10220 DATA 4,4,20,24,8
10230 REM------------------------------Read X0,Y0,Z0
10240 RESTORE
10250 FOR I=1 TO 24
10260 READ X0(I),Y0(I),Z0(I)
10270 NEXT I
10280 REM------------------------------Read Line Descriptors
10290 FOR I=1 TO 7
10300 FOR J=1 TO 5
10310 READ L0(I,J)
10320 NEXT J
10330 NEXT I
10340 REM------------------------------Initialize M
10350 GOSUB 11320
10360 REM------------------------------Define Window
10370 GOSUB 10570
10380 REM------------------------------Define Viewport
10390 GOSUB 11130
10400 REM------------------------------Generate Line List
10410 FOR I=1 TO 7
10420 J=1
10430 GOSUB 11730
10440 FOR J=2 TO 5
10450 X1=X2
10460 Y1=Y2
10470 Z1=Z2
10480 GOSUB 11730
10490 GOSUB 11840
10500 NEXT J
10510 NEXT I
10520 END
10530 REM    -----------------------------------------------------------
10540 REM  :   Define WINDOW                                             :
10550 REM  :                                                             :
10560 REM    -----------------------------------------------------------
10570 PRINT
10580 PRINT "Enter WINDOW boundary values, and Eye Point Distance."
10590 PRINT "Wl,wr,Wb,Wt,Wh,Wy,E : ";
10600 INPUT W0,W1,W2,W3,W4,W5,E
10610 W6=(W1+W0)/2
10620 W7=(W3+W2)/2
10630 W8=(W1-W0)/2
10640 W9=(W3-W2)/2
10650 GOSUB 11450
10660 T(1,1)=1
10670 T(2,2)=1
10680 T(3,3)=1
10690 T(4,1)=-W6
10700 T(4,2)=-W7
10710 T(4,3)=-E
10720 T(4,4)=1
10730 GOSUB 11550
10740 REM---------Center view axis at center perspective pyramid
10750 W0=W0-W6
10760 W1=W1-W6
10770 W2=W2-W7
10780 W3=W3-W7
10790 W4=W4-E
10800 W5=W5-E
10810 REM---------Define Window normals
10820 N(1,1)=0
10830 N(1,2)=2*W1*W4
10840 N(1,3)=-2*W1*W2
10850 N(2,1)=-2*W3*W4
10860 N(2,2)=0
10870 N(2,3)=2*W1*W3
10880 N(3,1)=0
10890 N(3,2)=-2*W1*W4
10900 N(3,3)=2*W1*W3
10910 N(4,1)=2*W3*W4
10920 N(4,2)=0
10930 N(4,3)=-2*W0*W3
10940 N(5,1)=0
10950 N(5,2)=0
10960 N(5,3)=1
10970 N(6,1)=0
10980 N(6,2)=0
10990 N(6,3)=-1
11000 REM---------Define i-th plane weight vector
11010 FOR I=1 TO 6
11020 FOR J=1 TO 3
11030 F(I,J)=0
11040 NEXT J
11050 NEXT I
11060 F(5,3)=W4
11070 F(6,3)=W5
11080 RETURN
11090 REM  :-----------------------------------------------------------:
11100 REM  :   Define VIEWPORT                                          :
11110 REM  :                                                            :
11120 REM  :-----------------------------------------------------------:
11130 PRINT
11140 PRINT "Enter VIEWPORT boundary values"
11150 PRINT "Vl,Vr,Vb,Vt : ";
11160 INPUT V0,V1,V2,V3
11170 V6=(V1+V0)/2
11180 V7=(V3+V2)/2
11190 V8=(V1-V0)/2
11200 V9=(V3-V2)/2
11210 REM---------
11220 PRINT @32,21:V0,V2
11230 PRINT @32,20:V1,V2
11240 PRINT @32,20:V1,V3
11250 PRINT @32,20:V0,V3
11260 PRINT @32,20:V0,V2
11270 RETURN
11280 REM   -------------------------------------------------------------
11290 REM      [ M ] <-- [ I ]
11300 REM
11310 REM   -------------------------------------------------------------
11320 FOR I=1 TO 4
11330 FOR J=1 TO 4
11340 IF I=J THEN 11370
11350 M(I,J)=0
11360 GO TO 11380
11370 M(I,J)=1
11380 NEXT J
11390 NEXT I
11400 RETURN
11410 REM   -------------------------------------------------------------
11420 REM      [ T ] <-- 0
11430 REM
11440 REM   -------------------------------------------------------------
11450 FOR I=1 TO 4
11460 FOR J=1 TO 4
11470 T(I,J)=0
11480 NEXT J
11490 NEXT I
11500 RETURN
11510 REM   -------------------------------------------------------------
11520 REM      [ M ] <-- [ T ] * [ M ]
11530 REM
11540 REM   -------------------------------------------------------------
11550 FOR I=1 TO 4
11560 FOR J=1 TO 4
11570 D(I,J)=M(I,J)
11580 NEXT J
11590 NEXT I
11600 FOR I=1 TO 4
11610 FOR J=1 TO 4
11620 M(I,J)=0
11630 FOR K=1 TO 4
11640 M(I,J)=M(I,J)+T(I,K)*D(K,J)
11650 NEXT K
11660 NEXT J
11670 NEXT I
11680 RETURN
11690 REM   -------------------------------------------------------------
11700 REM      [x0,y0,z0, 1][ M ] = [ x2,y2,z2, 1]                      :
11710 REM
11720 REM   -------------------------------------------------------------
11730 X2=X0(L0(I,J))*M(1,1)+Y0(L0(I,J))*M(2,1)+Z0(L0(I,J))*M(3,1)+M(4,
11740 Y2=X0(L0(I,J))*M(1,2)+Y0(L0(I,J))*M(2,2)+Z0(L0(I,J))*M(3,2)+M(4,
11750 Z2=X0(L0(I,J))*M(1,3)+Y0(L0(I,J))*M(2,3)+Z0(L0(I,J))*M(3,3)+M(4,
11760 RETURN
11770 REM  :------------------------------------------------------------
11780 REM  : 3 D   C L I P P I N G   A L G O R I T H M
11790 REM  : Input  F(i,3)  Directrix vectors                          :
11800 REM  :        N(i,3)  Normal vectors to Window Planes            :
11810 REM  :      X1,Y1,Z1  Vector start                               :
11820 REM  :      X2,Y2,Z2  Vector end
11830 REM  :------------------------------------------------------------
11840 D1=X2-X1
11850 D2=Y2-Y1
11860 D3=Z2-Z1
11870 U=1
11880 L=0
11890 FOR I1=1 TO 6
11900 G1=X1-F(I1,1)
11910 G2=Y1-F(I1,2)
11920 G3=Z1-F(I1,3)
11930 D=D1*N(I1,1)+D2*N(I1,2)+D3*N(I1,3)
11940 G=G1*N(I1,1)+G2*N(I1,2)+G3*N(I1,3)
11950 IF D=0 THEN 12010
11960 C=-G/D
11970 IF D>0 THEN 12030
11980 IF C<0 THEN 12220
11990 U=C MIN U
12000 GO TO 12050
12010 IF G<0 THEN 12220
12020 GO TO 12050
12030 IF C>1 THEN 12220
12040 L=C MAX L
12050 NEXT I1
12060 REM---------
12070 IF L=>U THEN 12220
12080 X3=X1+D1*L
12090 Y3=Y1+D2*L
12100 Z3=Z1+D3*L
12110 X4=X1+D1*U
12120 Y4=Y1+D2*U
12130 Z4=Z1+D3*U
12140 REM---------
12150 X5=X3/Z3*(W4*V8/W8)+V6
12160 Y5=Y3/Z3*(W4*V9/W9)+V7
12170 X6=X4/Z4*(W4*V8/W8)+V6
12180 Y6=Y4/Z4*(W4*V9/W9)+V7
12190 REM---------
12200 PRINT @32,21:ABS(X5),ABS(Y5)
12210 PRINT @32,20:ABS(X6),ABS(Y6)
12220 RETURN
```

Fig. 1.

To understand the program (Fig. 1), you should be aware of the two coordinate systems the program employs. The first coordinate system is left-handed and is used to describe the object to be viewed. The clipping algorithm given is driven by a short program that uses a cube as a test object. Figure 2 shows the cube as defined in the program listing (lines 10090-10140).

A left-handed coordinate results when the Z+ axis is extended into the screen. This was done for two reasons. First, the three-dimensional axes are a simple extension of the more familiar two-dimensional axes. And second, the 3-D clipping window specification is an extension of the same parameters used to specify the 2-D clipping window on the 4051.

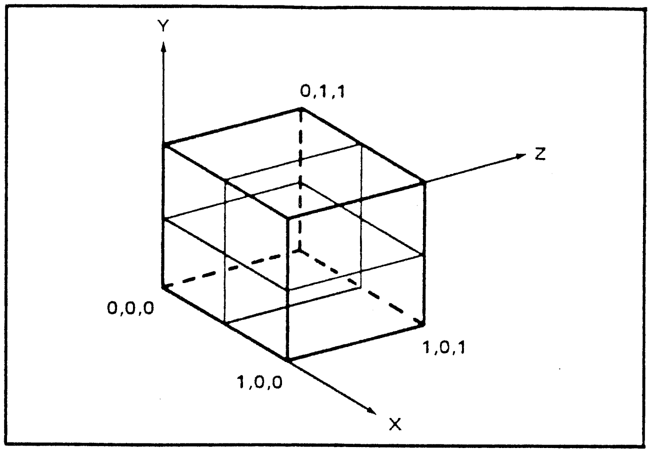


Fig. 2.

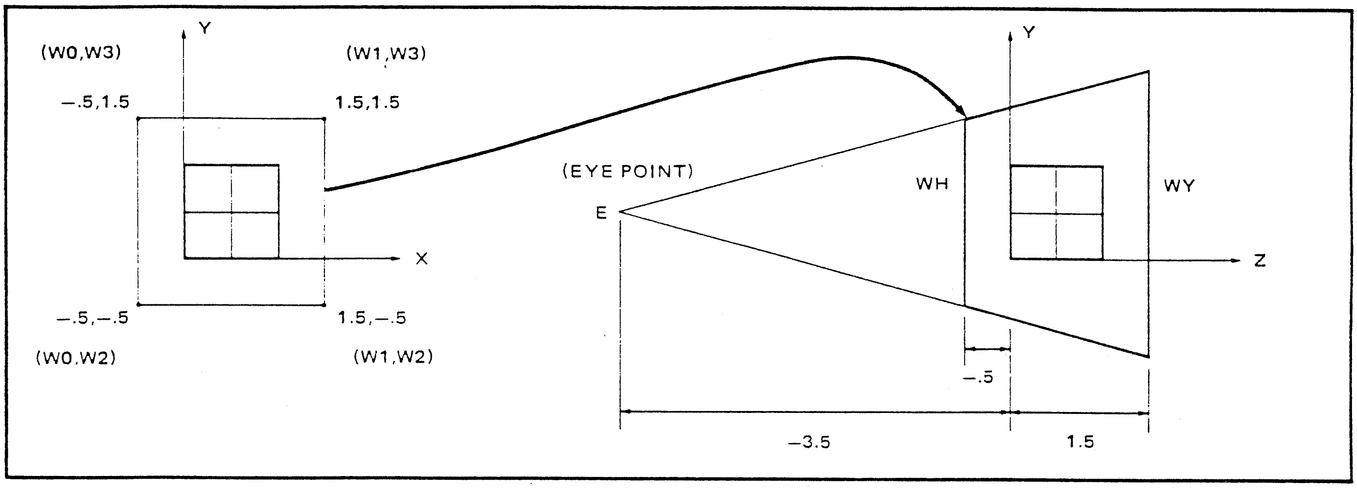


Fig. 3.

The 3-D clipping window is specified in the left-handed coordinate system of the cube. The field of view is shaped like a pyramid lying on its side. WH and WY, seen edge-on in Fig. 3, are the "near" (to E) and "far" clipping planes. One can think of the 2-D window specification as describing a rectangle on the WH plane. The aspect ratio of the pyramid is controlled by the shape of the rectangle and the distance "E" is located from it. We assume that the X,Y location of E is always aligned with the center of the rectangle, thus to locate the eye point, only its Z coordinate is required.

The second coordinate system is used to reduce the complexity of doing the clipping step. This coordinate system uses the eye point as the origin. The clipping boundaries are described to the program by specifying the normals (of the six planes) that face toward the pyramid's interior, and by specification of a point lying on each of the six planes. By locating the eye point at the origin, the point (0,0,0) is common to four of the six planes and can be used to satisfy part of the pyramid specifications.

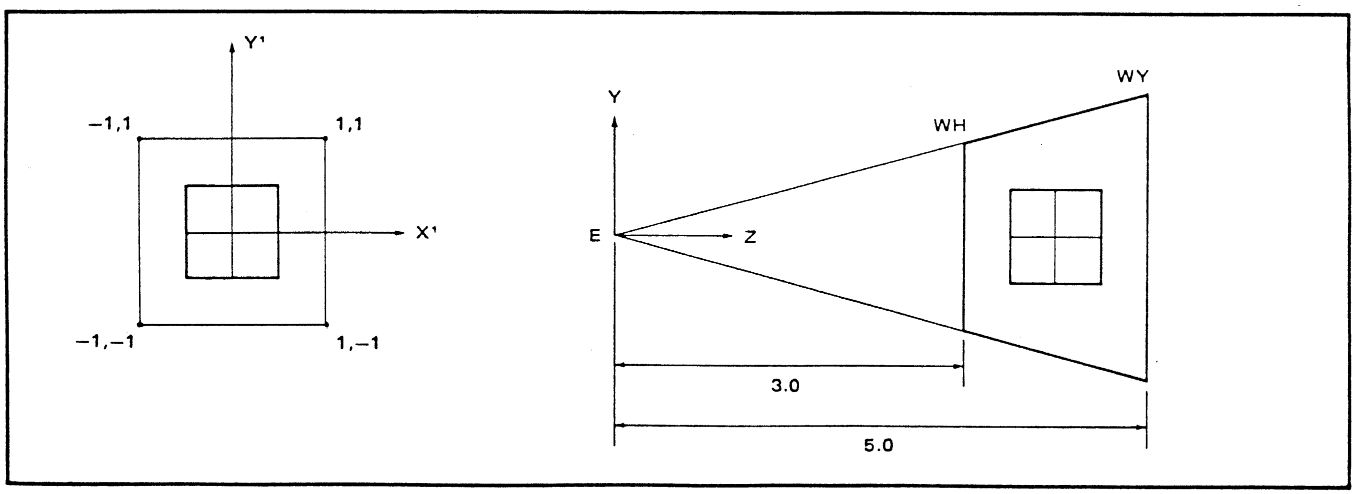


Fig. 4.

The translation of the test object to the clipping axes is performed by a translation matrix.[1]

$$[X\ Y\ Z\ 1]\begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ -W6 & -W7 & -E & 1 \end{bmatrix} = [X'\ Y'\ Z'\ 1]$$

$$W6 = \frac{W1 + W0}{2}$$

$$W7 = \frac{W3 + W2}{2}$$

The clipping algorithm works in basically the following manner. Suppose we are given a 3-D line segment "L" that penetrates a clipping plane "P". The projections of a line segment, and a reference vector onto a normal to the plane form a proportion b/a. The reference vector is known to originate on the surface of the plane (because we defined it that way).

The proportion b/a is always a value between 0 and 1 if the line segment intersects the plane. It may be <0, or >1, if an extension of the line segment intersects the plane, but these cases are rejected. The six clipping planes form a convex 3-space that can only be penetrated in two places by a straight line.[2] By determining the ratio b/a for all six planes a set of values is determined. If we order this set, the two middle values will determine the clipped line segment.

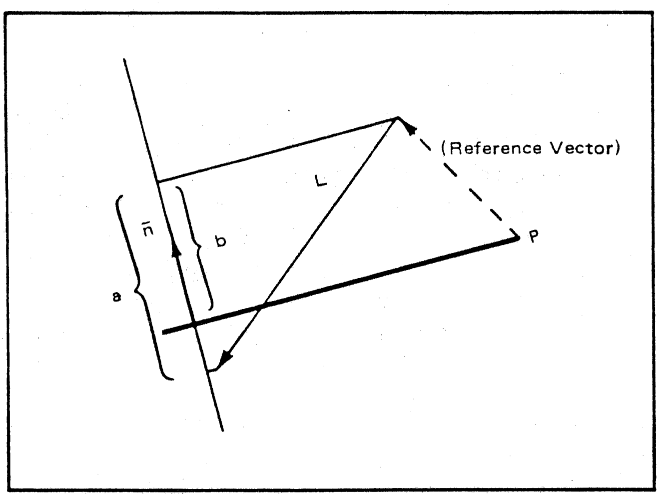


Fig. 5.

Figures 6 and 7 resulted from running the program. The input parameters are defined below.

Wl	Window left	(X_{min})	Vl	Viewport left	
Wr	right	(X_{max})	Vr	right	
Wb	bottom	(Y_{min})	Vb	bottom	
Wt	top	(Y_{max})	Vt	top	
Wh	hither	(Z_{min})			
Wy	yon	(Z_{max})			
E	Eye	(Z)			
	(coordinates of object)			(screen display units)	

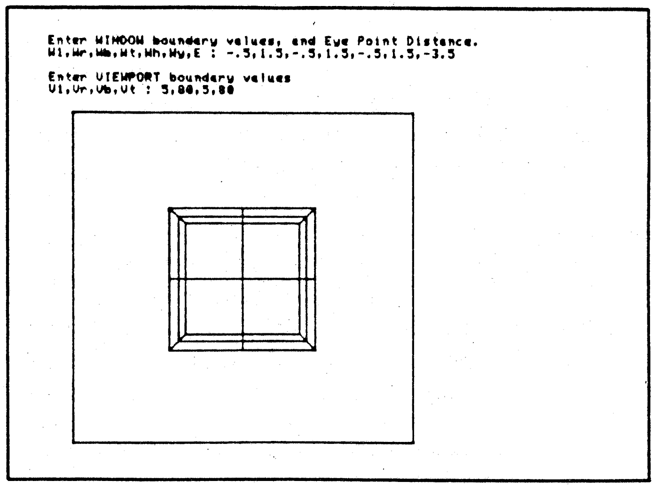


Fig. 6.

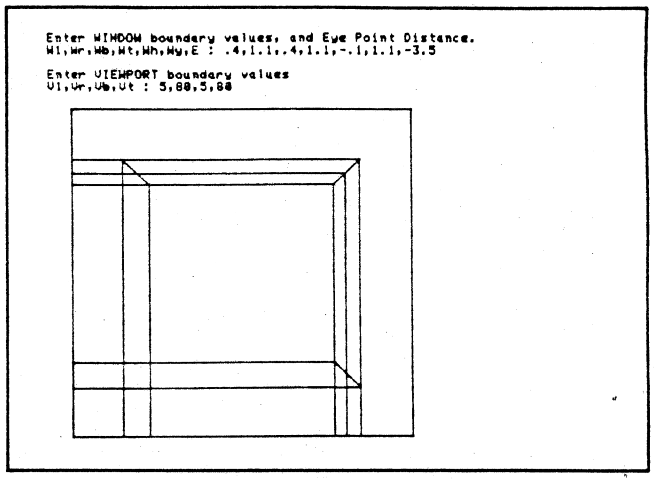


Fig. 7.

Program Note:

The program uses screen level moves (Print @32,21) and draws (Print @32,20). Roundoff error can result in very small negative numbers (on the order of -.6E-13) instead of zero. When this occurs the line is not drawn properly. To avoid this potential problem, the coordinates used are forced to be greater than zero by use of the ABS( ) function.

Describing Your Own Object

Lines 10090-10140 contain an ordered set of coordinates for the ends and midpoints of each edge in the cube's face. Each line (from 10160-10220) designates the coordinates to be used in drawing a closed line about the cube. The position in the X0, Y0, Z0 array in which the coordinate is stored is the value used on these lines.

To describe your own object, list the object's unique coordinates in a table.

Index	X	Y	X
1	X_1	Y_1	Z_1
2	X_2	Y_2	Z_2
3	X_3	Y_3	Z_3
4	X_4	Y_4	Z_4
etc.			

Then, in a data statement, write these coordinates in the following order:

$X_1\ Y_1\ Z_1\ X_2\ Y_2\ Z_2 \ldots$

Using one data statement per polygon, describe each polygon in the following manner:

$\langle$Number of points$\rangle$, $\langle$ index$_1$ $\rangle\langle$index$_2$ $\rangle$, $\langle$index$_3$ $\rangle$,... $\langle$ index$_n$ $\rangle$

e.g. DATA 5, 1, 4, 7, 9, 1

N 5 coordinates

Change line 10060 to:

max index P. max (n) +1

```
10060 DIM X0(24),Y0(24),Z0(24),L0( , )
```

Change Lines 10290 to 10330 to read:

```
10290 READ P
10292 FOR I=1 TO P
10294 READ L0(I,1)
10296 FOR J=2 TO N+1
10298 READ L0(I,J)
10300 NEXT J
10302 NEXT I
```

Also lines 10410 and 10440 should read:

```
10410 FOR J=1 TO P
!
!
!
10440 FOR J=2 TO L0(I,1)
```

[1] *Newman, W. and Sproull, R.: Principles of Interactive Computer Graphics, Chapters 12-14, (McGraw Hill 1973).*

[2] *Beck, J. and Cyrus, M.: Computer Graphics: Two- and Three-Dimensional Clipping, AFHRL-TR-77-14, USAF Systems Command.*

Base Changing Routine

by Henry E. Zeuli
Massachusetts Institute of Technology

The following routine can be easily included in any program that requires conversion from base 10 (decimal) to another base supplied by the operator. Lines 430 through 550 operate on any decimal number in D1 and a base, sixteen or under, entered in B1. The number value in the new base is returned in R$. The routine operates on any positive decimal integer.

```
380 PRINT "DECIMAL INPUT, BASE OF OUTPUTJ"
390 INPUT D1,B1
400 GOSUB 430
410 PRINT USING "35TFALFD":R$,B1
420 GO TO 390
430 REM        DEC TO ----- CONV SUBR
440 D$="0123456789ABCDEF"
450 R$=""
460 H$=""
470 D2=INT(D1/B1)
480 D3=D1-D2*B1
490 H$=SEG(D$,D3+1,1)
500 R$=H$&R$
510 IF D2=0 THEN 540
520 D1=D2
530 GO TO 470
540 DELETE D2,D3,D$,H$
550 RETURN
```

Tie Menu to Both Keyboard and User-Definable Keys

Many 4051 programs can be written so that the user may use either a screen menu, inputting the choice from the keyboard, or the User-Definable Keys. The following program layout is a suggestion of a standard use of line numbers and a way to tie the menu to the UDK's so that both are equally easy to use.

```
1 REM--Program title goes here
2 SET KEY
3 GO TO 100
4 REM--Describe 1st subprogram here
5 C=1
6 GOSUB 1000
7 END
8 REM--Describe 2nd subprogram here
9 C=2
10 GOSUB 2000
11 END
!
!
!
40 REM--UDK #10--Use screen menu
41 C=10
42 GO TO 300
!
!
!
80 REM--Describe 20th subprogram here
81 C=20
82 GOSUB 20000
83 END
!
!
!
100 REM--Menu Routine
110 REM--Do all dimensioning here
120 REM--Do all data assignments here
130 REM--The above should only have to be
131 REM--executed once.  After that the
132 REM--user should use menu or UDK and
133 REM--never have to type "RUN"
!
!
!
300 REM--Screen Menu
310 PRINT "LTitle of ProgramJ"
320 PRINT " 1--Subprogram 1"
330 PRINT " 2--Subprogram 2"
!
!
!
410 PRINT "10--Stop and Use UDKs"
!
!
!
490 PRINT "19--Subprogram 19"
500 PRINT "20--Subprogram 20"
510 PRINT "JYour Choice: ";
520 INPUT C
530 IF C=10 THEN 580
540 GOSUB C OF 1000,2000,3000,4000,5000,6000,7000,8000,9000,580
550 GOSUB C-10 OF 11000,12000,13000,14000,15000,16000,17000,18000,20000
560 REM--Redisplay menu
570 GO TO 300
580 REM--Return 4051 to idle mode
590 END
!
!
!
1000 REM--Subprogram #1
1010 REM--Use submenus if necessary
!
!
!
1XXX RETURN
!
!
!
2000 REM--Subprogram #2
!
!
!
2XXX RETURN
!
!
!
10000 REM--Use this area for any extra routines.
10001 REM--This area is free since item 10
10002 REM--is used to flip-flop between the
10003 REM--screen menu and UDKs.
!
!
!
20000 REM--Subprogram #20
!
!
!
20XXX RETURN
!
!
!
21000 REM--Multiply used utilities and other
21001 REM--subprograms can start here
```

NOTES:

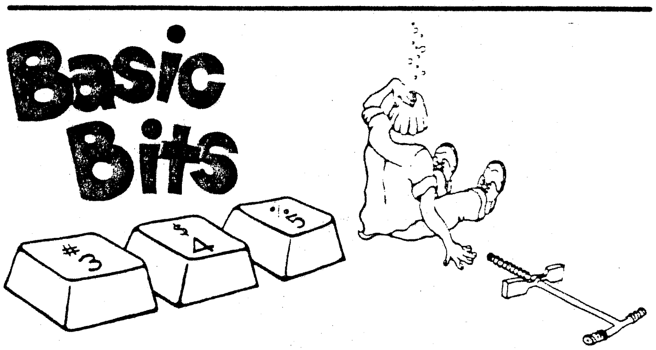
1) Note that C is set both by the screen menu and UDK's. This means that you use C as a flag in cases where one subprogram GOSUB's to another.

2) GOSUB's must be used in the UDK definitions in order for proper return when the screen menu is used.

3) If there aren't enough lines for a particular subprogram, use a GOSUB to line numbers above 21000 or use the 10000 area. Be sure to terminate the extension with a "RETURN".

4) By definition, the UDK starting line number is the UDK number times 4. The subprogram starting line number is the UDK number or C times 1000.

5) By using UDK #10 (C=10) for flip-flopping between the screen menu and using the UDK's, you can always get the screen menu by pushing UDK #10, even if you have lost the card overlay.



Good programming practice includes keeping a backup of your program on a separate tape. And, as we often find out too late, using the MARk statement very carefully is also good programming practice. But if hindsight has outranked foresight, the following tip may be of help.

Salvaging re-MARked Files

by Hermand D'Hondt
Tektronix Australia

The following program usually works to salvage files which have been MARked over. However, it is not guaranteed. It seems to depend on the precise location of the "LAST" file in relation to the succeeding files.

```
100 ON EOF (0) THEN 200
110 PRINT @33,0:0,0,1
120 REM Find the new "LAST" file
130 FIND 3
140 PRINT @33,0:0,0,0
150 INPUT @33:A$
160 PRINT A$
170 GO TO 150
200 FIND 5
210 REM Input, old or append lost file
220 INPUT @33:A$
230 PRINT A$
240 GO TO 220
```

Another Method (A Last Resort)

If the preceding program does not work, your local Tektronix Systems Analyst may be able to assist you.

Variables for File Numbers

Often a program uses multiple tape files. Using variables instead of constants as arguments of the FINd statement allows changing the sequence of files on a tape by recoding only one statement in a file. Example:

```
100 INIT
110 REM Total number of files in program.
120 DATA 4
130 REM Number of each file.
140 DATA 4,5,6,7
150 READ N
160 DIM F(N)
170 READ F
180 REM Continue with main program.
!
!
!
600 REM Call in subprogram file
610 FIN F(2)
```

4051 Applications Library Program Abstracts

Documentation and program listings of these programs may be ordered for $15.00 each. Programs will be put on tape for an additional $2.00 handling charge per program and a $26.00 charge for the tape cartridge. (The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc. assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.)

Please use the Applications Library Order Form. Order forms are included in the Membership Packet and are available from your local Tektronix Sales Engineer. Orders outside of the U.S. must be ordered through the local Tektronix sales office.

ABSTRACT NUMBER: 51/00-8016/0

Title: **Variable Size Screen Characters**
Author: K. S. Reid-Green
Educational Testing Service
Princeton, NJ
Memory Requirement: 8K
Peripherals: 4631 Hard Copy Unit Optional
Statements: 53

Use this subroutine to generate characters of variable size to be displayed on the 4051 screen. This technique is useful when displaying textual information which will subsequently be plotted in other than the standard character size.

The subroutine converts all letters, numbers and the characters " [", "] ", and " . " to strokes.

```
Enter character size x,y: 5 8

Enter output coordinates x,y: 2 75

Enter character string: 4051 Library

4051 LIBRARY
```

ABSTRACT NUMBER: 51/00-8017/0

Title: **Paper Tape Labeler**

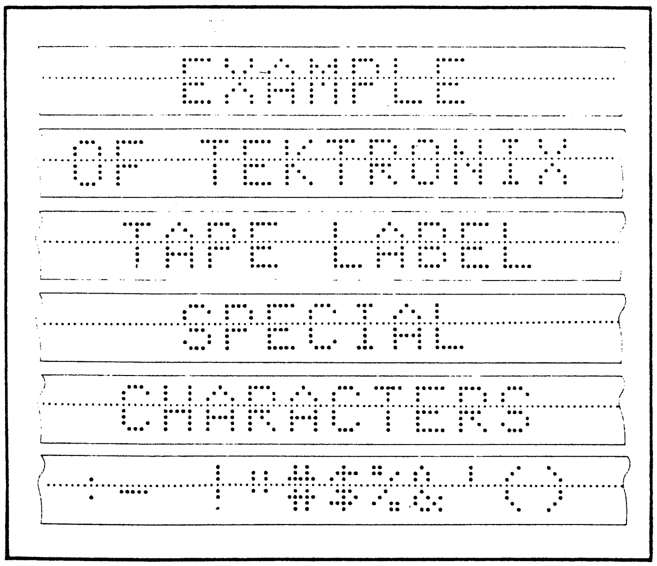
Author: G. L. Evans
Herne Hill, London SE240DQ

Memory Requirement: 8K Bytes
Peripherals: Paper Tape Punch
Statements: 93

When the 4051 is used to drive a paper tape punch, this program allows a visual header to be punched in the beginning of a paper tape for easy identification.

A leader of 50 nulls is first punched, and a header of 72 characters maximum is entered into the 4051 via the keyboard. All of the usual editing facilities of the 4051 may be used at this time. The header is then coded in a standard 5 x 7 matrix and punched, followed by a trailer of 50 nulls.

An example of one type of system used in this mode is the punch on an ASR teletype connected to the 4051 via the IEEE bus (GPIB). The conversion between the IEEE standard and the teletype logic is done using a microprocessor programmed to allow the teletype to be used in the commands SAVE @1: and PRINT @1: so that data can be sent after the header.



ABSTRACT NUMBER: 51/00-5401/1

Title: **Graphics Demonstrations for Astronomy and Physics**

Author: Dr. R. J. Reimann
Physics Department
Boise State University

Memory Requirement: 8K
Peripherals: 4662 Plotter Optional
Files: 12
Statements: 807

A second edition with an additional program and numerous improvements in the other programs. This is a program series, under the control of a directory, for the presentation of introductory astromony and physics concepts to students with no computer experience. The series includes:

Ellipse—ellipses are drawn according to the user's choice of eccentricity. Kepler's first and second laws are illustrated.

Planetary Motion—the inner planets are initially aligned and then "race" in circular orbits according to Kepler's third law.

Binary Stars—circular orbits for binary systems are plotted using Kepler's third law. The user may select the masses and separation.

Stellar Magnitudes—relative apparent magnitudes (0-6) are drawn for stars under the assumption that brightness is proportional to area.

Phaser—a demonstration of the reference circle as related to simple harmonic motion.

Sidereal Time—a view of the Northern sky with sidereal clock is drawn according to the user's choice.

Solar Time—a view of the Northern sky is drawn according to the user's choice of date and time of day.

Satellite Orbit—earth satellite orbits are plotted using a 2-step iterative approximation of Newton's law of universal gravitation. User selects radial and tangential velocities with an optional choice of initial separation.

Trajectory—a 2-D trajectory is drawn using a 2-step iteration assuming drag is proportional to v^2. The plot is automatically scaled and labelled. The user selects velocity components v_x and v_y with an optional choice of drag coefficient.

Vector Combinations—addition of numerous 2-D vectors drawn head-to-tail. The plot is automatically scaled and the resultant is expressed in both rectangular and polar coordinates.

Linear Least-Squares Fit—user inputs number of data points and each x,y coordinate. The best fit to a straight line is calculated including standard errors. The plot is automatically scaled and "proper" "tic" marks are made on the axes. Hard copies are produced by pressing User Definable Key #3 if a 4662 Plotter is connected.

ABSTRACT NUMBER: 51/00-9521/0

Title: **Spider Web Charts**

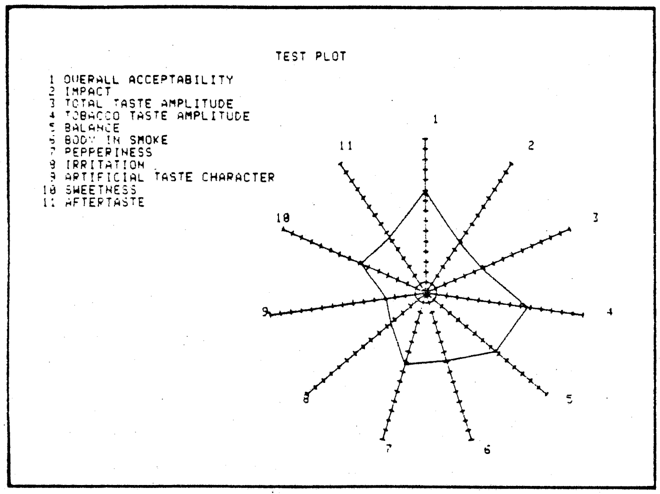
Author: Tom Price
Lorillard Research
Greensboro, NC

Memory Requirement: 8K

Periphals: 4631 Hard Copy Unit

Statements: 74

This program produces spider web profiles for any number of parameters. Such profiles are useful in the concise presentation of multiple properties related to an item on a single graph. Rapid comparisons between items are greatly facilitated. This type of presentation is widely used in the tobacco industry for the comparison of flavor properties between different brands.



ABSTRACT NUMBER: 51/00-3101/0

Title: **Minimum Operating Cost for Two Aircraft**

Author: Florent van Vlasselaer
Tektronix Belgium

Memory Requirements: 8K

Peripherals: 4952 Joystick
4631 Hard Copy
4662 Plotter (Optional)

Statements: 180

The program determines all possible mixes of two types of aircraft covering the same route for one week. The constraints are total passengers and cargo and, for each plane, capacity and upper and lower limits of capacity actually used.

The output is a parallelogram on the X-Y axes; this output encloses the possible mixes. The operator can then use the joystick to select points for which the annual operating cost will be calculated.

TWO PROGRAMS IN ONE

Both the Slide Maker program (51/00-8009/0) and the Pie Chart Routine for 4662 (51/00-9512/0) have been incorporated into program 51/00-9513/0, PRESENTATION AIDS RELEASE 2. Any orders designating either of these first two will be filled with Presentation Aids.

The User Definable Keys allow a choice of either the slide maker or pie chart function.

Tektronix
COMMITTED TO EXCELLENCE

TEKTRONIX, INC.
Information Display Group
Applications Library
Group 451
P.O. Box 500
Beaverton, Oregon 97005

ADDRESS CORRECTION REQUESTED