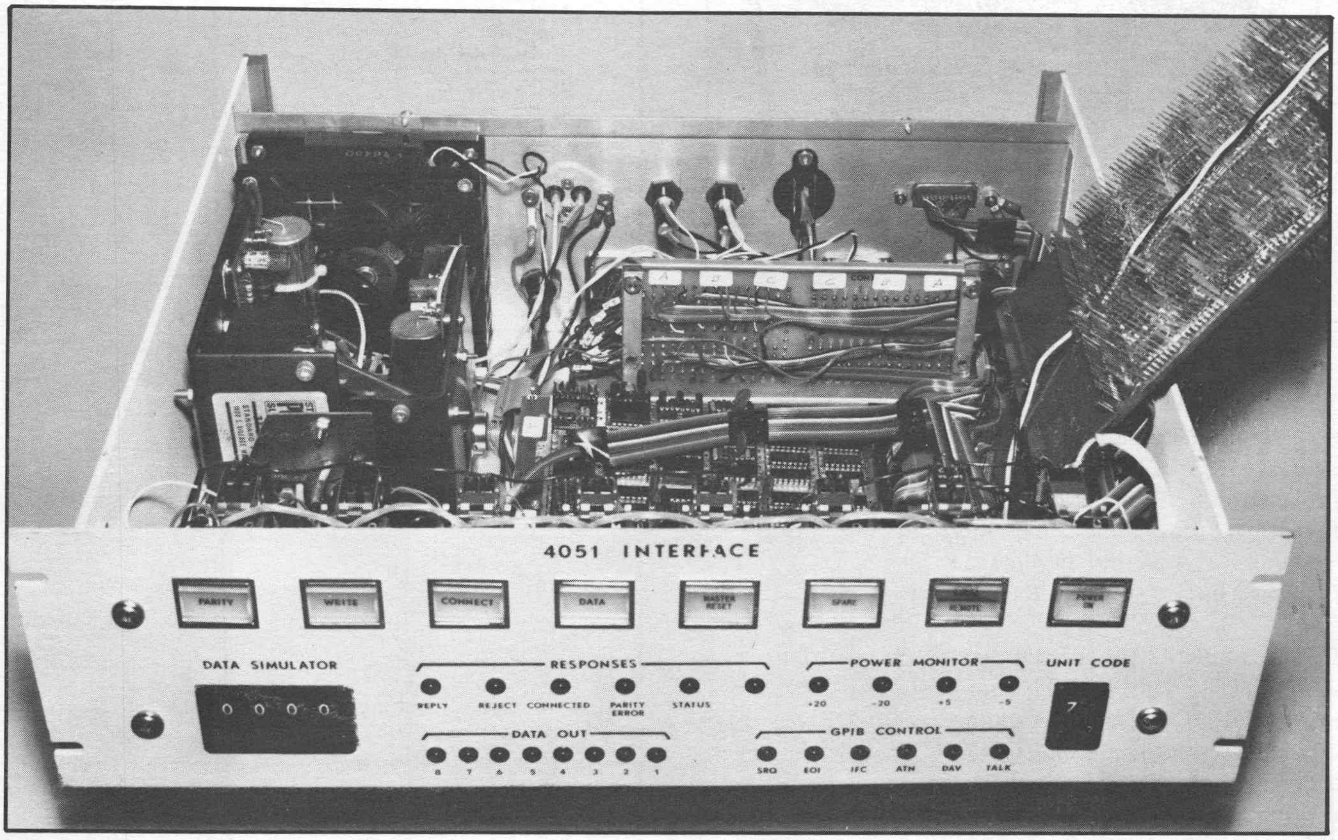
Tekniques

The 4051 Applications Library Newsletter Vol. 2 No. 6



The Front Panel of the CDC 3200—4051 Interface showing indicator lights and thumbwheel switches. Both GPIB and CDC responses are shown by the lights.

An Interface for Data Transfer Between CDC 3000 Series Computer and a 4051

by David Bruning
Lockheed Electronics

At the White Sands Test Facility, Lockheed Electronics has designed and built an interface that facilitates communication between a CDC 3200 computer and the TEKTRONIX 4051 Graphic System. A GPIB interfacing mode is used, as is necessary for computers like the 3200 that are not RS-232 compatible. This interface is used primarily to transfer data stored on the CDC 3200's 1/2 inch magnetic tape to a 4924 Tape Drive for storage on cartridge tapes. The 4051's interactive graphics capabilities are later used to plot the data, off-line from the 3200.

The cover photo and Fig. 1 show front and top views of the interface, which is centered around MC3441 transceiver chips. Its two main functions are reducing the data from a 12-bit byte structure to an 8-bit byte, and coordinating the handshake of data transfer and management lines between the two machines. This discussion centers around the CONNECT, WRITE and DISCONNECT functions of the 3200, and the SRQ

In This Issue

TALK and EOI functions of the 4051. Note the interface's front panel and thumbwheel switches; these switches permit off-line testing of the handshake timing and logic.

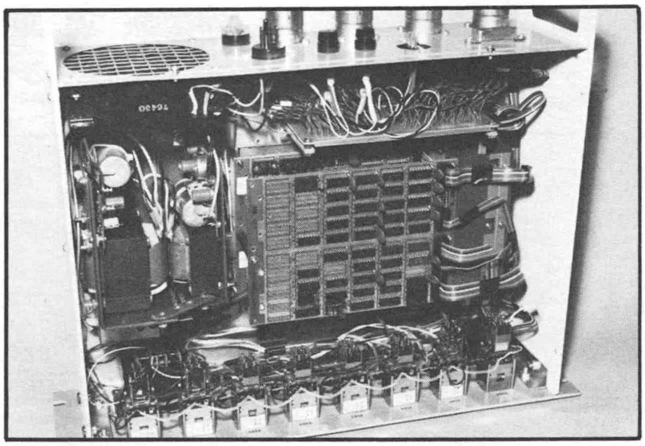


Fig. 1. Top view of the interface.

To transmit data from the 3200, a CONNECT is first necessary. This is a function that logically attaches the interface to the 3200 for I/O purposes. The action prompts an SRQ signal on the GPIB, telling the 4051 to prepare to receive data. The 4051, in turn, places the interface in TALK and the 4924 in LISTEN. The 3200 can then pass data, using a WRITE function.

Upon completion of the data transfer (which has been accomplished with WBYTE commands at the 4051), the interface sends a disconnect signal to the 4051. This signal is created on the 3200 by a CONNECT to another device; it prompts an EOI signal which terminates the WBYTE transfer. Fig. 2 shows the program used to activate the 4051 data transfer. On-line plotting could be accomplished by placing the 4662 Digital Plotter in LISTEN, instead of the 4924.

ASCII is used for data transfer. This means a conversion from floating point to integer, integer to BCD, BCD to ASCII, and the insertion of an ASCII decimal point (the original number had been scaled for this purpose) is necessary. A bit shift is also required to match transmitted lines for the 12-bit to 8-bit reduction. The 3200 handles this conversion, since it is essentially idle due to the 1100 byte-per-second transfer rate. Although the interface is presently a one-way data communication from the 3200 to the 4051, it can be easily modified for two-way interaction. The handshake logic is already present.

Fig. 3 shows a sample plot of data that has been transferred to the 4051 by this process. Each grid line corresponds to 10 GDU; the data is plotted at 50 points per 10 GDU. All annotations, titles, axis limits, etc., are passed with the data, although they may be changed interactively with User Definable Keys. The number of Y axis grid lines to be drawn is chosen by the 4051 on the basis of axis limits. Plot symbols are staggered to increase the clarity of the plot.

```
100 REM--THIS PROGRAM PERMITS THE EXCHANGE OF DATA BYTES FROM THE
110 REM--CDC 3200 TO THE 4924 MAG.
120 REM--TAPE UNIT
130 REM - PROGRAM ASSUMES PERIPHERAL DEVICE NUMBERS ARE:
140 REM -   INTERFACE=0
150 REM -   4924      =2
160 REM - THE FIRST INFORMATION TO BE PASSED IS THE HEADER(ASCII)
170 INIT
180 PRINT "LCDC 3200 TO 4051 INTERFACE PROGRAM"
190 PRINT "JInsert new tape in the 4924 - press RETURN to continue";
200 INPUT M$
210 REM - SET FILE COUNTERS
220 L=1
230 K=1
240 REM - TEST STATUS OF THE TAPE
250 ON SRQ THEN 700
260 FIND @2:L
270 PRINT @2:"a"
280 KILL @2:L
290 FIND @2:L
300 OFF SRQ
310 WBYTE @63,95:
320 PRINT "J4051 READY FOR DATA TRANSFER"
330 REM - START DATA TRANSFER CYCLE
340 REM - WAIT FOR SRQ
350 ON SRQ THEN 380
360 WAIT
370 GO TO 360
380 POLL M,W;0;2
390 IF M<>1 THEN 700
400 REM - PUT INTERFACE IN TALK, 4924 IN LISTEN
410 REM - WAIT FOR EOI
420 ON EOI THEN 570
430 WBYTE %64,127,34,108:
440 WAIT
450 IF K<=2 THEN 520
460 REM - SIGNAL OPERATOR FOR KEYBOARD ENTRY
470 GOSUB 610
480 REM - GO TO NEW DATA FILE: CHECK FOR NEW DATA TAPE
490 PRINT "New Tape? (Y or N)";
500 INPUT M$
510 K=1
520 L=L+1
530 FIND @2:L
540 IF M$="Y" THEN 170
550 REM - RETURN FROM SRQ INTERRUPT
560 RETURN
570 WBYTE @63,95:
580 K=K+1
590 REM - RETURN FROM EOI INTERRUPT
600 RETURN
610 FOR I=1 TO 2
620 PRINT "GGGGG";
630 FOR J=1 TO 100
640 NEXT J
650 PRINT "GG";
660 FOR J=1 TO 100
670 NEXT J
680 NEXT I
690 RETURN
700 PRINT "4924 REQUESTS SERVICE ** MESSAGE ";
710 INPUT @2,30:A
720 PRINT A
730 IF A=7 THEN 760
740 IF A=9 THEN 790
750 END
760 PRINT " Insert cartridge in 4924- press RETURN to continue"
770 INPUT M$
780 RETURN
790 PRINT " Tape is write protected- press RETURN to continue"
800 INPUT M$
810 RETURN
820 INPUT @2:A$
830 PRINT A$
840 GO TO 820
```

Fig. 2. Program listing.

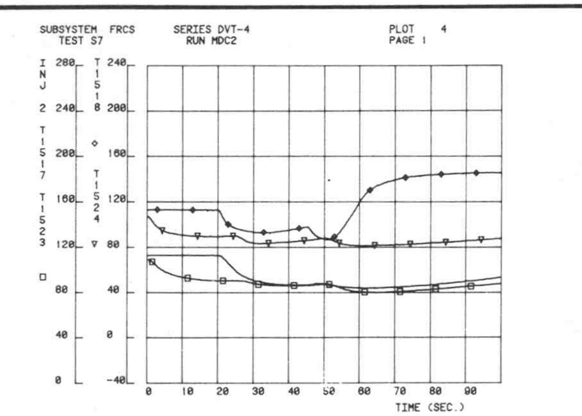


Fig. 3. Sample plot produced by the 4051 Graphic System. The data is generated by the CDC 3200 and passed to the 4924 Tape Drive. The plot is then produced off-line from the CDC 3200.

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 Applications Library.

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

IEEE Spectrum Magazine's Amazing Micro-Mouse maze and timing equipment require sophisticated mice.

4051 Controls Real Time System for Micro-Mouse Contest

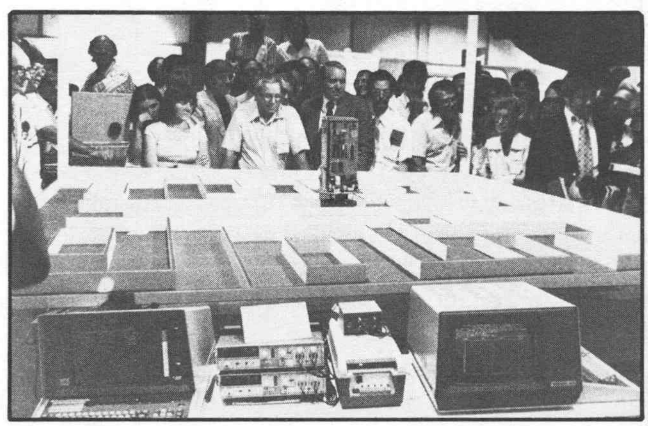
by Miki Tokola

The classic scene of a researcher timing a mouse running through a maze took an unusual twist in the Personal Computing area of the National Computer Conference (NCC) recently held in Anaheim, California. The squeaking and sniffing rodent had an electronic substitute and a 4051 replaced the researcher.

The event was the preliminary time trials of the IEEE *Spectrum* magazine's "Amazing Micro-Mouse Contest".

The contest is for electronic mice designed and built by the contestants. An entrant need not look like its natural counterpart, and, in fact, may be much larger. The mouse may be up to 10 inches in width and length with no limit in height. It must be completely self-contained and may contain batteries, electric motors, mechanical springs, memories or microprocessors.

Each mouse is allowed three runs through one of two mirror image wooden mazes. The mazes consist of elbows, tees, U's, dead-ends, and straightaways. Like the real thing, the mouse must physically negotiate the maze and can only use inside walls as a guide for physically sensing the maze. Prizes are awarded for the fastest mouse first time through, the best learning mouse (the best time in the last of the three runs), and special merit including the one for the most ingenious design.



A tall "mouse" attempts the maze.

At the National Computer Conference, the substitute for the researcher with a stopwatch was a real-time computing system. The system was controlled by a TEKTRONIX 4051, and included a TEKTRONIX 4924, 4025, two Fluke 1953A counter/timers, and two CONRAC display monitors as shown in Fig. 1. The 4051 controlled the timers, maintained the data base, and displayed the results immediately after each run.

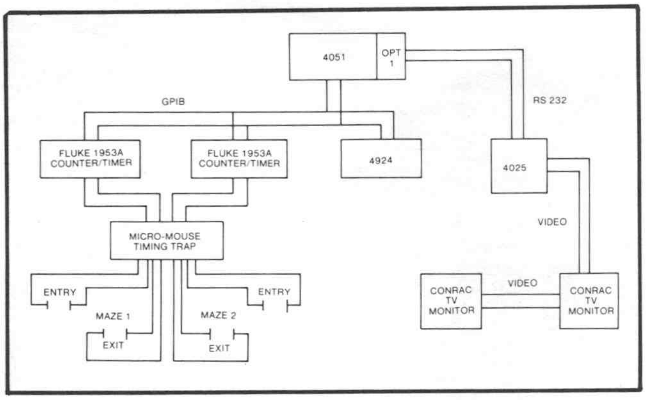


Fig. 1. Micro-Mouse Timing and Display System.

Instead of the researcher visually noting the location and time of a mouse in the maze, photo-electric sensors embedded in the walls of the maze signalled the race start and stop times. The portion of the 4051 program which controlled the Fluke 1953A counter/timers is listed below.

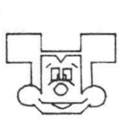
```
6200 REM SETUP MAZE-1 COUNTER AT GPIB ADDRESS 3
6201 M1=N
6205 GO TO 6500
6210 PRINT @3:"C F0"
6220 PRINT @3:"F4 R5 A-1 B-1 LA+2.0 LB+2.0 M1 H1"
6230 M1=N
6240 PRINT @3:"F0 F4 T"
6250 RETURN
6400 REM MAZE-2 SETUP FOR CONTER AT GPIB ADDRESS 4
6405 M2=N
6406 GO TO 6500
6410 PRINT @4:"C F0"
6420 PRINT @4:"F4 R5 A-1 B-1 LA+2.0 LB+2.0 M1 H1"
6430 PRINT @4:"F0 F4 T"
6440 RETURN
6500 REM MOUSE TIMING SYSTEM ACTIVE LOOP
6510 ON SRQ THEN 6600

6600 REM THIS IS WHERE THE SRQ'S GO
6610 POLL A,B;3,4
6615 IF B=97 THEN 6620
6616 PRINT "COUNTER ERROR MAZE #";A;" GGGGGG"
6617 END
6620 GO TO A OF 6700,6800
6700 REM ENTER SCORE FOR MAZE #1
6710 F=M1
6715 D$=""
6720 GOSUB 3010
6725 INPUT @3:T
6726 GOSUB 6760
```

More sensors could have been used to record the time and position of the mouse during the entire run, as well as its entry into and exit from the maze.

In addition to its timing responsibilities, the 4051 maintained a data base containing the contestant's names, addresses, contest numbers, and race results. Information was retrieved by pushing a function key and entering a contestant number. Fig. 2 is a 4051 display of the data base information for the eventual winner of the trials. A contestant list was also available at the push of a function key.

*** OFFICIAL MICRO-MOUSE CONTESTANT DATA ***



CONTESTANT NUMBER: * * * 1 * * *
WHO IS: ART BOLAND
REPRESENTING: BATTELLE NORTHWEST
RESIDING AT: RICHLAND WA.
PHONE:

************* TIMING RESULTS ************

RUN 1	RUN 2	RUN 3
0:51.4	ABORTED	ABORTED

COMMENTS: THIS PERSON WON THE FIRST PRELIMINARY TIME TRIALS!!

Fig. 2. 4051 records a winning mouse and owner.

The contest attracted a great deal of interest. TV monitors displayed contestant times and race information to the curious spectators. Fig. 3 is an example of the format that was used. A cable from the RS-232 (Option 1) output of the 4051 was plugged into a 4025. The display monitors were connected to the video output of the 4025, providing output to the spectators.

IEEE SPECTRUM/COMPUTER MAGAZINES AMAZING MICRO-MOUSE MAZE CONTEST

FIRST PRELIMINARY TIME TRIALS AND EXHIBITION

NEXT TIME TRIALS: PHILADELPHIA,PA. AUG 24-27 AT 'PERSONAL COMPUTING 78'.

ENTRIES TO MICROMOUSE 78' CLOSED.
IF INTERESTED IN FUTURE CONTEST SEE ATTENDANT

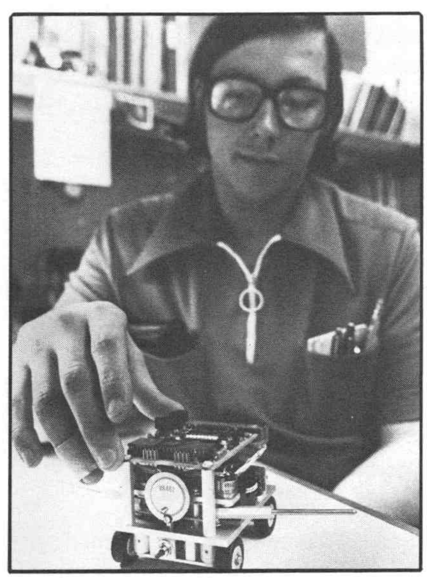
CONTESTANT NUMBER AND TIMES

NUMBER	RUN 1	RUN 2	RUN 3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

Fig. 3. Micro-Mouse Monitor Display.

Over 5,000 people have registered for the contest, but only a few mice actually ran at these trials, the first in the series. The difficulty of the task is demonstrated by the fact that only two mice completed the maze. The winning mouse negotiated the 5 foot by 10 foot maze in 51.4 seconds. Future contest dates and locations will be announced by IEEE *Spectrum*.

The concept of having a 4051 control a real-time system to measure and record the position of an object or to analyze inputs from remote sensors has possibilities far beyond controlling a hobbyist contest. Who know what other classic scenes will soon have a 4051 replacing one of the familiar characters?



TEKTRONIX employee Mike Cranford will re-enter his "mouse" after further education.

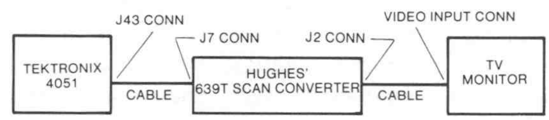
Modified 4051 Interfaces with Hughes 639T Scan Converter

by Pat Kelley

Tektronix customers are innovative. Provide them with the intelligent, versatile 4051 and their ingenuity is boundless. And some 4051 owners figured a 4051 could be used in additional applications if it only had a hardware modification. While the 4051 in its natural state is a most "well rounded" desk top computer, if a lot of requests come in for a particular custom design, our engineers take a look at them.

Several users wanted to interface their 4051 to a Hughes 639T Scan Converter for display on Closed Circuit Television (CCTV) monitors or systems. This would enable them to take advantage of their 4051 graphic screen in group presentations and for classroom instruction. To do this they needed special X, Y, Z outputs, appropriate handshake and an erase signal from the 4051. As a result, Tektronix now has a modified 4051 which enables the 4051 to connect with the Hughes 639T Scan Converter.

The character or vector data signals are transmitted from the 4051 to the 639T via the Hughes-supplied interconnector cable. This cable runs from a special connector (J43) on the back of the 4051 to connector J7 on the 639T. The CCTV monitor connects to the Video Output on the back of the 639T. No special set-up procedures are required other than connecting the equipment as shown in the diagram.



Although X and Y signal gains and offsets are factory-set to give correct aspect ratio and framing on a standard 525-scan-line monitor, requirements do vary. Provisions are made for adjustment by detailed instructions and an easy-to-use test pattern program in the manual. An insulated screwdriver is the only tool needed to make the adjustment.

The changes required in the 4051 preclude retrofitting an existing 4051 in the field. However, the modification may be ordered on any new 4051. If this special interface fits your needs, contact your local Sales Engineer. Ask about the 4051 Mod AB.

4051 Tests Solar Cells at JPL

In Pasadena, California, Jet Propulsion Laboratory is using the 4051 Graphic System to control an innovative test system. Dr. Jim Liu wanted to test the performance of solar cells and solar cell material, and needed to use non-GPIB devices to do it. Dr. Liu chose the 4051 as a controller, citing the GPIB and graphics capabilities as prime reasons. The 4051 is connected, via the GPIB, to a Hewlett-Packard 6940B Multi-Programmer. The Multi-Programmer, in turn, is connected to such non-GPIB devices as Digital Voltmeters, Variable Power Supplies, Light Detectors, Monochromators, and Stepper Motors. This forms the system shown in Fig. 1, which performs laboratory tests of solar cells and materials far faster than previous methods.

The Multi-Programmer outputs all of the required device-dependent signals to the devices connected to it. Additionally, it collects the input data from several different experiments performed by the system, and passes the results along to the 4051 over the GPIB. All Multi-Programmer action is under control of the 4051 through the GPIB.

The system is used to test solar cells and solar cell materials in several experiments. Tests measure the output voltage and current as well as the spreading resistance of the material. Solar cells are measured for operating efficiency under illumination, and are tested in the dark to measure their diode characteristics. The system is used to measure diffusion length of carriers in the materials, to provide spectral response curves, and to gather and manipulate data that can be used to determine the efficiency of solar cells and solar cell material. The experiments are currently performed one at a time; the goal is to perform several in parallel. This may be done by gathering data from one test while the 4662 is plotting results of another.

Besides gathering and manipulating experimental data, the 4051 performs other controller activities. For instance, it can output commands through the GPIB to operate the stepper motors and vary light intensity. The stepper motors, in turn, can change sample placement material, allowing another portion of the sample to be tested. Other control outputs from the 4051 change wave length in the experiment. This method makes the testing virtually automatic.

Dr. Liu chose the 4051 Graphic System for this test system because of its high-resolution graphics and its GPIB capabilities. The 4051 draws data from the experiments back through the GPIB and manipulates it to provide useful results. These results take the form of several different graphs of the characteristics of solar cells and solar cell materials. Some typical outputs are Spectral Response curves, Voltage vs. Current, and plots of the conversion efficiency curve of the cells. In addition to the plotted curves, the 4051 will provide a list of all of the experimental data.

This test system is still being refined at Jet Propulsion Laboratory. As it stands, the system already provides a quantum leap in speed and efficiency over the previous data gathering methods. As an example, prior to setting this system in place, current vs. voltage data was laboriously gathered and plotted by hand, point by point. This method took between one and two hours to gather and plot 20 data points. The test system can gather 50 data samples in two **minutes**. As another example, Dr. Liu cited the plotting of Spectral Response curves. This operation previously took an entire morning of gathering data by hand, followed by subsequent computer reduction. The 4051-based test system completes the entire operation, including graphic outputs, in nine minutes. Time savings and high resolution graphics: that's what the 4051 is all about.

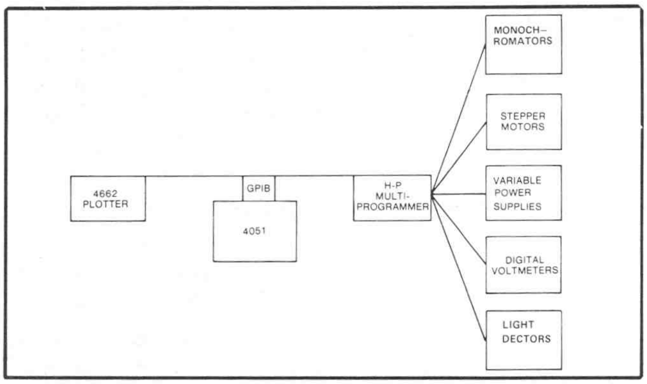


Fig. 1. Solar Cell Test System.

PLOT 50 EE Vol. 1 Speeds Circuit and System Design from Audio to Microwave Frequencies

by Dave Barnard

PLOT 50 Electrical Engineering Vol. 1 software permits interactive modeling and analysis of circuit designs in the frequency domain. From audio to microwave frequencies, whether a prospective circuit is to perform amplification, filtering, or impedance matching, this powerful package will predict circuit performance and assist designers in making decisions. A specialist working at microwave frequencies will find the interactive Smith Chart plots, impedance mapping and matching network design routines particularly helpful; this is a major emphasis in the software package. Audio and communications equipment designers can use many of the general purpose routines, as the article in TEKniques Vol. 1 No. 8 pointed out.

Extensive two-port manipulating capabilities allow anything that can be modeled as a linear two-port network to be analyzed. Circuits of active and passive elements can be constructed from a menu of elementary two-ports. The payoff is the ability to simulate a proposed design quickly with a stand-alone 4051 system. Results may be copied from the screen using a 4631 Hard Copy Unit or plotted on a 4662 Interactive Digital Plotter. Graphically portraying performance of circuits makes the trade-offs obvious to the designer. It doesn't replace engineering intuition; it complements it.

4051 BASIC programming ability is not required as a prerequisite to using EE Vol. 1. Function keys and an overlay make software operation easy. Errors may be edited out and changes made simply. Error messages are in plain language rather than in error number format.

For an engineer with BASIC programming experience, EE Vol. 1 makes a good companion to the collection of routines that are probably already on hand.

You can get started with EE Vol. 1 by simply inserting the tape and pressing AUTO LOAD. Whether the problem is simple or complex, the ease of operation won't let the program get in the way; there is sufficient power to handle demanding tasks.

Construct your circuits from the menu of two-ports which includes elementary series or parallel resistors, capacitors, and inductors. Tank circuit models—RLC—are included. Current or voltage controlled I or V sources permit modeling active device configurations. Specify transmission line models by either physical or electrical length plus impedance, dielectric and loss coefficients.

If a device (a transistor for example) has known parameters, they can be entered simply and interactively. The program converts the parameters to or from any form desired. Transistor configurations can be converted to or from common emitter, common base, or common collector.

Example

TEKniques Vol. 1 No. 8 first introduced EE Vol. 1 software to you. Let's suppose that the design of a high frequency amplifier (Fig. 1) was started following the example discussed in that article.

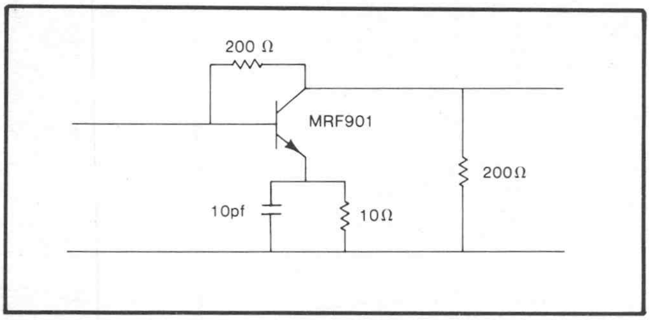


Fig. 1. Amplifier from TEKniques with MRF901 Transistor.

Analyzing amplifier performance with a different transistor requires entering the device parameters via the ENTER DATA function key. Fig. 2 shows how easily this may be done. Data for the MRF901 was taken from the Motorola **RF DATA Manual** and entered as shown.

```
ENTER DATA
Frequency Units (GHZ, MHZ, KHZ, or HZ)?          GHZ
Parameters (S, Y, Z, H, G, 1/S, 1/Y, or 1/Z)?   S
Polar, DB, or Rectangular form (P, D, or R)?    P
Enter FREQ and 11, 12, 21, and 22 data
 1:  0.2 .45 -115 .02 53 17.2 114 .57 -33
 2:  0.5 .45 -162 .04 55  7.7  90 .40 -31
 3:  1.0 .47  166 .07 64  4.0  71 .35 -35
 4:  1.5 .51  148 .10 67  2.6  58 .34 -47
 5:  2.0 .53  135 .13 68  2.0  46 .33 -62
 6:
```

Fig. 2. Device data entry.

Once entered, the S-parameters may be recorded on the 4051 internal magnetic tape by using the DATA ON TAPE function key.

Assuming data is saved in File 2, the code to analyze the circuit should refer to that file as shown in line 5 (Fig. 3).

```
ENTER CODE
 1:  GHZ COP
 2:  FE .1 10 20
 3:  RP 10 CA
 4:  CP 10 SE
 5:  DAT 2 PA
 6:  RS 200 CA
 7:  RP 200 EQ
 8:  NEXT
 9:  GAM G21-LO
10:
11:
LIST CODE
 1:  GHZ COP
 2:  FE .1 10 20
 3:  RP 10 CA
 4:  CP 10 SE
 5:  DAT 2 PA
 6:  RS 200 CA
 7:  RP 200 EQ
 8:  NEXT
 9:  GAM G21-LO
10:
Code length = 84.  Simple diagnostic (Y or N)?  Y
Diagnostic completed.  Errors are noted above.
```

Fig. 3. Code entry for an amplifier.

By executing this code the entire circuit is reduced to a single set of two-port parameters which may be saved on tape for future reference (Data File 4, for example) as the circuit is built up.

A listing of gammas (Fig. 4) was requested to facilitate design of input and output impedance matching networks.

	FREQUENCY	K	GMAX	GAMMA MS		GAMMA ML	
	GHZ		DB	MAG	ANGLE	MAG	ANGLE
1:	0.10	1.322	8.466	0.148	56.559	0.151	172.188
2:	0.13	1.318	8.495	0.155	58.650	0.154	169.195
3:	0.16	1.313	8.536	0.165	61.066	0.160	165.568
4:	0.21	1.307	8.586	0.179	64.365	0.167	161.175
5:	0.26	1.304	8.592	0.204	72.132	0.173	155.419
6:	0.34	1.298	8.631	0.240	81.763	0.183	148.964
7:	0.43	1.281	8.794	0.299	94.037	0.198	142.694
8:	0.55	1.237	9.268	0.394	107.717	0.229	136.911
9:	0.70	1.210	9.665	0.476	119.977	0.261	127.345
10:	0.89	1.180	10.217	0.587	139.198	0.294	113.950
11:	1.13	1.164	10.573	0.696	163.364	0.336	90.614
12:	1.44	1.167	10.047	0.763	-173.542	0.410	66.772
13:	1.83	1.160	8.973	0.790	-154.255	0.485	58.513
14:	2.34	1.196	8.237	0.777	-145.223	0.470	55.855
15:	2.98	1.208	8.200	0.771	-141.958	0.457	49.707
16:	3.79	1.219	8.074	0.762	-139.169	0.446	44.347
17:	4.83	1.229	7.908	0.751	-136.947	0.437	40.216
18:	6.16	1.237	7.737	0.740	-135.254	0.430	37.257
19:	7.85	1.244	7.580	0.730	-133.995	0.424	35.213
20:	10.00	1.250	7.445	0.721	-133.069	0.420	33.817

Fig. 4. Listing of amplifier gammas, gain and stability.

The graphic display of gain (G_{21}) permits rapid comparison with the previous design to aid the design decision making process (Figs. 5 and 6).

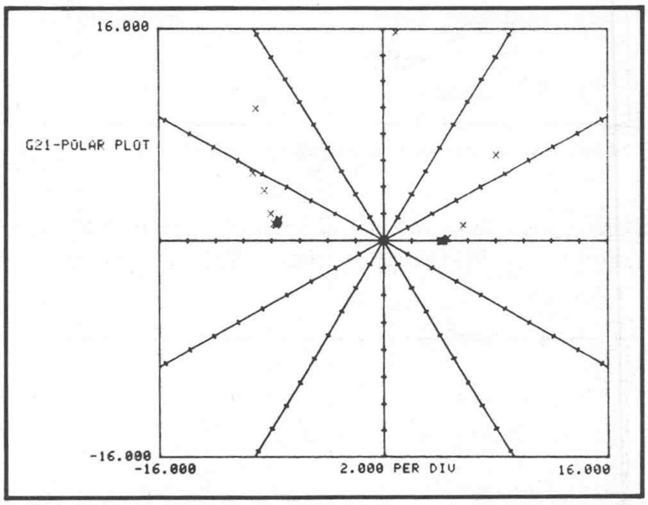


Fig. 5. Gain magnitude and angle.

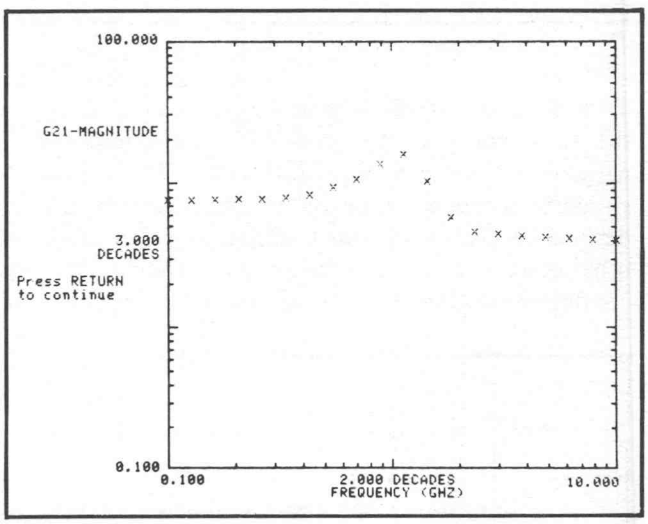


Fig. 6. Gain—vs—Frequency.

Matching network design is as simple as depressing the UTILITY function key and following the menu (Fig. 7 and Fig. 8). From two to four realizations of series and shunt elements are given.

```
MATCHING NETWORK DESIGN

Enter Frequency in GHZ              1.0

PORT 1
Termination (ZS, ZP, or GAM)?       ZS
Enter R and X (ohms)                50 0

PORT 2
Termination (ZS, ZP, or GAM)?       GAM
Enter Magnitude and Angle           .696 -163.364

L1 SHUNT =    3.7656 NH                    X = 23.6599 OHMS
C2 SERIES =   12.9808 PF                   X = -12.2608 OHMS

C1 SHUNT =    6.7268 PF                    X = -23.6599 OHMS
L2 SERIES =   4.202 NH                     X = 26.4019 OHMS

Press RETURN to do another design
```

Fig. 7. Input impedance matching network.

```
MATCHING NETWORK DESIGN

Enter Frequency in GHZ              1.0

PORT 1
Termination (ZS, ZP, or GAM)?       GAM
Enter Magnitude and Angle           .336 -90.614

PORT 2
Termination (ZS, ZP, or GAM)?       ZS
Enter R and X (ohms)                50 0

L1 SERIES =   1.544 NH                     X = 9.7014 OHMS
L2 SHUNT =    15.5277 NH                   X = 97.5631 OHMS

L1 SERIES =   8.0039 NH                    X = 50.2899 OHMS
C2 SHUNT =    1.6313 PF                    X = -97.5631 OHMS

L1 SHUNT =    7.9104 NH                    X = 49.7023 OHMS
C2 SERIES =   6.4124 PF                    X = -24.8198 OHMS

L1 SHUNT =    37.9905 NH                   X = 238.7015 OHMS
L2 SERIES =   3.9502 NH                    X = 24.8198 OHMS

Press RETURN to do another design
```

Fig. 8. Output impedance matching network.

Using the results of this utility routine yields the circuit shown (Fig. 9). Obtaining these results takes only a few moments.

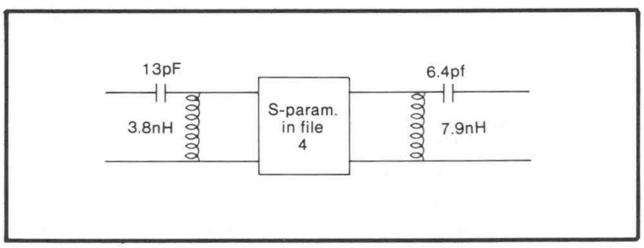


Fig. 9. Amplifier with matching networks.

To predict overall performance only requires cascading the networks just designed with the composite S-parameter black box previously saved. By building the circuit up a step at a time and saving intermediate steps, designed decisions are aided, steps may be re-traced and other designs tried with a minimum of coding. The total configuration (Fig. 9) is coded in a few steps (Fig. 10).

```
LIST CODE
 1:  GHZ
 2:  FE .1 10 20
 3:  LP 3.7656 CA
 4:  CS 12.9808 CA
 5:  DAT 4 CA
 6:  CS 6.4124 CA
 7:  LP 7.9104 EQ
 8:  NEXT
Code length = 83.  Simple diagnostic (Y or N)?
```

Fig. 10. Simulation code for circuit (Fig. 9).

The end result can be the input and output S-parameters S_{11} and S_{22} as shown (Fig. 11, and 12) or gain versus frequency (Fig. 13).

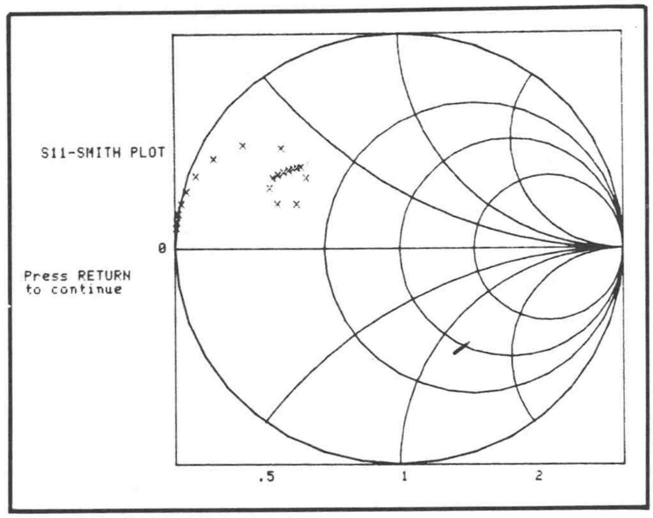


Fig. 11. Input S-parameter.

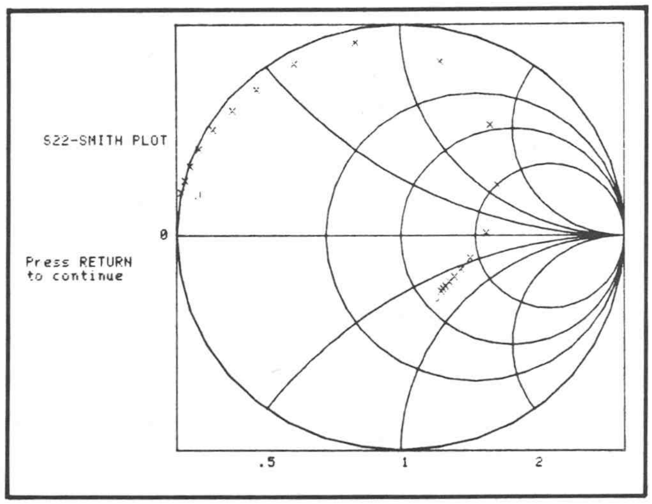


Fig. 12. Output S-parameters.

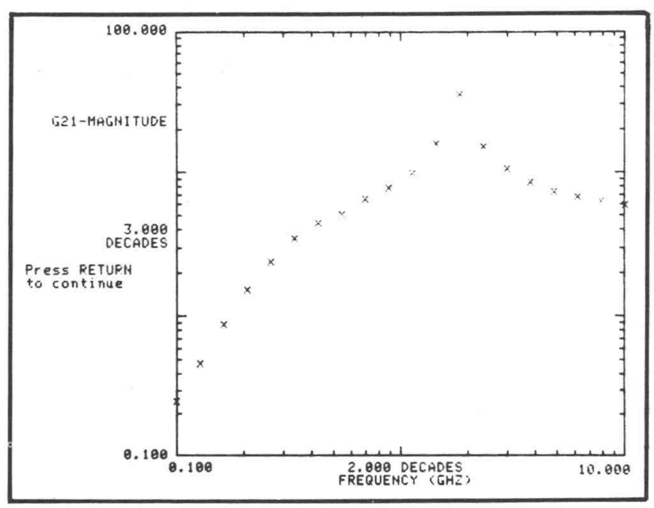


Fig. 13. Gain—vs—Frequency for whole amplifier.

Plot 50 EE Vol. 1 also contains routines that permit including transmission lines in place of the lumped impedance matching networks just designed, if better bandwidth is desired. But this article's purpose has been to illustrate the ease of use and interactiveness of the program, which permits design iterations to be made simply.

In addition to this illustrated example, Plot 50, EE Vol. 1 (4050 A06) can facilitate:

- Filter design
- Impedance matching network design
- Minimum noise figure design
- Transmission line and strip line analysis

Is there enough room on the tape that comes with EE Vol. 1 for data, code and results? The answer depends on how much data must be saved. The tape can store 25 different two-ports (of 30 lines each), and 25 coded circuits plus 5 different sets of results. Storage is easily expanded with a blank tape cartridge and the file marking and transfer utility routines accessible via the function keys.

Don't ignore EE Vol. 1 if your application doesn't approach the daylight bands of modern microwave circuit design. Simulating designs before breadboarding saves significant time even at audio frequencies.

Finally, if the task isn't an EE problem but one which can be solved as an analogue of the electrical models, EE Vol. 1 could do the job. For example, the elementary tank circuit (RLC) has a well known mechanical analogue consisting of spring, mass, and friction elements.

We'd like to hear from anyone using EE Vol. 1 to simulate mechanical, acoustical or other systems using the electrical models only as analogues of the original problem. Send a description of whatever you are modeling and sample analysis using EE Vol. 1 to TEKniques. Our address is on the back of each issue.

The 4051R06 Editor: Searching Operations

by Cathy Cramer

In the last issue of TEKniques, we talked about the 4051R06 EDITOR and its applications. This time, we'll describe some special searching operations you can perform using the EDITOR.

The 4051R06 EDITOR is a ROM pack. Installing the ROM pack and typing CALL "EDITOR" turns the 4051 into an ASCII text editor that can handle text of any kind. While the EDITOR is in control, you have 29 commands at your disposal. All are specially designed for easy text handling, but the most powerful are the searching commands.

Searching commands simplify the editing process. They quickly locate occurrences of a particular string of characters in the text, and simultaneously replace, edit, or delete those characters. Searching commands can make one correction, or many corrections at once. They're ideal for major changes to programs and text. But they also make simple everyday editing tasks go a lot faster and easier.

The 4051R06 EDITOR has two searching commands, SEARCH and NLSEARCH (No List SEARCH). Each has several forms that share the same keyword, but differ in syntax and function. SEARCH has four forms: SEARCH and List Line, SEARCH and Replace String, SEARCH and Edit Line, and SEARCH and Delete Line. NLSEARCH has two forms: NLSEARCH and Replace String, and NLSEARCH and Delete Line. The two forms of NLSEARCH are like the equivalent forms of SEARCH, but don't provide a listing of the changed or deleted lines. Table 1 summarizes the forms of the SEARCH and NLSEARCH commands.

Command	What it Does	Example
SEARCH and List Line	Lists lines that contain the string. You can send the listing to any tape storage or printing device by specifying a device number.	S "SET"
SEARCH and Replace String	Replaces occurrences of the string with another string, and lists the changed lines. You can send the listing to any tape storage or printing device.	S "PRINT", "PRINT@33:"
SEARCH and Edit Line	Recalls lines containing the string to the line buffer for editing. The lines are recalled and displayed one at a time.	S 1,10 "Mr.";
SEARCH and Delete Line	Deletes lines containing the string, and lists the deleted lines. You can send the listing to any tape storage or printing device.	S @33:"NO REPLY"*
NLSEARCH and Replace String	Replaces occurrences of the string with another string. (Does not list changed lines.)	NL "ON ERROR", "ON SIZE"
NLSEARCH and Delete Line	Deletes lines containing the string. (Does not list deleted lines.)	NL "TRACE"*

TABLE 1

Note: You can include line numbers in any SEARCH or NLSEARCH command, to specify the portion of the text you want to search.

Both SEARCH and NLSEARCH search the text, or some portion of the text, for any character string you want. You enter S for SEARCH or NL for NLSEARCH, then the string you want, enclosed in quotation marks. Additional entries tell the EDITOR which form of the command you want. (For example, a SEARCH command that ends with an asterisk is the SEARCH and Delete Line form of the command. A SEARCH command that ends with a semicolon is the SEARCH and Edit Line form of the command.)

A single SEARCH or NLSEARCH command can make major corrections to programs or text in just seconds. The same corrections might take hours to do from the keyboard. For example, suppose you're working on someone else's subroutine, planning to incorporate it into your own BASIC program. You know that both the subroutine and your program use variable name A1, but for different purposes. So you want to change A1 to A2 everywhere it's mentioned in the subroutine. Unless you have an EDITOR ROM pack, you have to check all the lines yourself, and find every mention of A1. For each one, you have to press the RECALL LINE key, position the cursor to A1, and type A2. You then move on to correct another line. If you've been through this procedure before, you know how slow and tedious it is. But with the EDITOR ROM pack, the whole procedure is reduced to just one command. You use the SEARCH and Replace String form of the SEARCH command. All you do is enter S "A1", "A2". (See Fig. 1.)

```
L
       : 2625 REM SORT ROUTINE
       : 2630 FOR A1=1 TO N
       : 2635 X2=A1
       : 2640 FOR N2=A1 TO N
       : 2645 IF P(N2)=>P(X2) THEN 2655
       : 2650 X2=N2
       : 2655 NEXT N2
       : 2660 X0=P(X2)
       : 2665 P(X2)=P(A1)
       : 2670 P(A1)=X0
       : 2675 NEXT A1
       : 2680 RETURN

S "A1","A2"
       : 2630 FOR A2=1 TO N
       : 2635 X2=A2
       : 2640 FOR N2=A2 TO N
       : 2665 P(X2)=P(A2)
       : 2670 P(A2)=X0
       : 2675 NEXT A2

L
       : 2625 REM SORT ROUTINE
       : 2630 FOR A2=1 TO N
       : 2635 X2=A2
       : 2640 FOR N2=A2 TO N
       : 2645 IF P(N2)=>P(X2) THEN 2655
       : 2650 X2=N2
       : 2655 NEXT N2
       : 2660 X0=P(X2)
       : 2665 P(X2)=P(A2)
       : 2670 P(A2)=X0
       : 2675 NEXT A2
       : 2680 RETURN
```

Fig. 1. The EDITOR command S"A1", "A2" replaces occurrences of A2 with A2, and lists all changed lines. Initial and final listings are obtained by entering L for LIST.

Another example: Suppose you're writing or debugging a large program, and are having trouble keeping track of how and where variable B$ is used. You could painstakingly inspect the program yourself, and write down on a piece of paper everything you find out about B$. But a much easier alternative is to use SEARCH and List Line to obtain a listing of the lines that contain B$. All you have to do is type S "B$". The listing immediately appears on the display. (See Fig. 2.)

```
L
       : 3200 A$=" "
       : 3210 FOR N1=1 TO 22
       : 3220 READ B$
       : 3230 K$=A$
       : 3240 A$=K$&B$
       : 3250 NEXT N1
       : 3260 DELETE B$,K$
       : 3270 L1=0
       : 3280 DIM F(9),G(9)
       : 3290 DIM B$(12),C$(25)

S "B$"
       : 3220 READ B$
       : 3240 A$=K$&B$
       : 3260 DELETE B$,K$
       : 3290 DIM B$(12),C$(25)
```

Fig. 2. The EDITOR command S "B$" lists the lines that contain B$. The initial listing is obtained by entering L for LIST.

When you finish debugging the program, you can delete all SET TRACE and SET NORMAL statements using two SEARCH and Delete Line commands, S "TRACE"* and S "NORMAL"* . If you need to correct lines containing PRINT, but don't want to make the same change to each line, you can use SEARCH and Edit Line, and enter S "PRINT";. This successively recalls lines containing PRINT to the line buffer, for you to edit however you like using the keyboard keys. SEARCH and Edit Line saves you time because the EDITOR not only finds all the lines for you, but also automatically recalls each one, just as if you had pressed the RECALL LINE key.

There are literally hundreds of other uses for the EDITOR's searching commands. You can search for any string, and list it, replace it, delete it, or edit it by hand.

```
L
       :Brockway, Marius E.
       :Carmichael, David   -PAID
       :Ellis, Terry L.
       :Foster, Alice       -PAID
       :Gardner, Keith W.
       :Harvey, Richard A.
       :Hillstrom, A. A.    -PAID
       :Kearney, John D.    -PAID
       :Keller, Suzanne     -PAID
       :Lentz, Robert F.

FIN8

PRI@33:LIST OF THOSE WHO PAID

S@33:"PAID"*

L
       :Brockway, Marius E.
       :Ellis, Terry L.
       :Gardner, Keith W.
       :Harvey, Richard A.
       :Lentz, Robert F.

FIN8
OLD
L
       :LIST OF THOSE WHO PAID
       :        :Carmichael, David   -PAID
       :        :Foster, Alice       -PAID
       :        :Hillstrom, A. A.    -PAID
       :        :Kearney, John D.    -PAID
       :        :Keller, Suzanne     -PAID
```

Fig. 3. The EDITOR command S @33: "PAID" deletes lines containing PAID, and stores them on file 8 of the internal tape. An initial listing is obtained by entering L for LIST. Then FIN8 and PRI@33: LIST OF THOSE WHO PAID open file 8 for access. L for LIST shows that the lines have been deleted; and FIN8, OLD, and L show that the deleted lines are now stored on the tape.*

And don't forget that the EDITOR lets you work with any kind of text, not just BASIC programs. You can edit FORTRAN, COBOL, data, free text—any set of ASCII characters. Fig. 3 shows a sample use for the SEARCH command when the text is a list of names and notations instead of a program. In this example, SEARCH and Delete Line separates out a subset of the text, and stores it on a different file. The command S @33:"PAID"* deletes the clients who've paid their bill, and lists (stores) them on file 8 of the internal tape.

Special Characters

To help you with your searching operations, the EDITOR provides four special characters. All four can be used in SEARCH or NLSEARCH commands, and have special meanings that make searching easier. The four characters are ~, #,], and _ by default, but you can cancel or change these assignments at any time.

The first is a "wildcard" character that allows you to search for any ASCII character. For example, you can search for any character that is immediately followed by $ (like A$, B$, C$ and so on). The second is an "any digit" character you can use to locate any of the digits 0 through 9. For instance, you can search for the character X immediately followed by a digit (as in X1, X2, X3, and X4). The third is an end-of-record character that allows you to locate, delete, or insert CARRIAGE RETURNS in the text. The fourth is an "all but" character that lets you search for all but one specified character. For example, you can find occurrences of @ that are *not* followed by 3 (like @1 and @A).

As a practical example, suppose you've written a large BASIC program, and used different string variable names A$, B$, C$, D$, and so on to store YES or NO responses from the user. You realize that one variable name could be used for all responses, so to conserve memory space, you decide to change all string variable names to E$. You could enter separate NLSEARCH and Replace String commands; one to search for A$ and change it to E$, another to search for B$ and change it to E$, another to change C$ to E$, and so on. A simpler way is to use the "wildcard" character ~, and enter a single command, NL "~$", "E$".

Fig. 4 shows a simple example of how this works. The EDITOR understands ~ to mean "any character," and looks for any character followed by $. It finds A$, B$, C$ and D$ and changes them all in one step to E$.

Now suppose there's a bug somewhere in your program, and you want to list and examine all numeric initialization statements like A=0, E=1.25, and X=1.0. If you've used a lot of variables, you don't want to have to search for each one individually. And a command like S "A="

```
L
: 1760 PRINT "IS THIS A CONTINUATION OF THE SAME PROBLEM?";
: 1770 INPUT A$
: 1780 IF A$="Y" THEN 2140
: 1790 GOSUB 430
: 1800 PRINT "IS DATA ON PROGRAM TAPE?";
: 1810 INPUT B$
: 1830 IF B$="Y" THEN 1950
: 1840 E(7)=1
: 1850 PRINT "IS DATA TO BE READ FROM EXTERNAL DEVICE?";
: 1860 INPUT C$
: 1870 IF C$="N" THEN 1930
: 1880 PRINT "DO YOU WANT TO ENTER DATA FROM THE KEYBOARD?"
: 1890 INPUT D$

NL "~$","E$"

L
: 1760 PRINT "IS THIS A CONTINUATION OF THE SAME PROBLEM?";
: 1770 INPUT E$
: 1780 IF E$="Y" THEN 2140
: 1790 GOSUB 430
: 1800 PRINT "IS DATA ON PROGRAM TAPE?";
: 1810 INPUT E$
: 1830 IF E$="Y" THEN 1950
: 1840 E(7)=1
: 1850 PRINT "IS DATA TO BE READ FROM EXTERNAL DEVICE?";
: 1860 INPUT E$
: 1870 IF E$="N" THEN 1930
: 1880 PRINT "DO YOU WANT TO ENTER DATA FROM THE KEYBOARD?"
: 1890 INPUT E$
```

Fig. 4. The EDITOR command NL " $", "E$" changes A$, B$, C$, and D$ to E$. Initial and final listings of the text are obtained by entering L.

finds extra statements you don't want to look at yet, such as A=A+VØ. So you use the special "digit" character # and enter S "=#". The EDITOR understands the symbol # to mean "any digit 0 through 9," and looks for an equals sign followed by a digit. It finds A=0, E=1.25, X=1.0, and so on. (See Fig. 5.)

```
L
        : 1220 PRINT "ENTER # OF VALUES:";
        : 1230 INPUT X
        : 1240 GOSUB 1660
        : 1250 A=0
        : 1260 E=1.25
        : 1270 X=1.0
        : 1280 GO TO 1030
        : 1290 Y=2
        : 1300 GOSUB 2630
        : 1310 GO TO 1050
        : 1320 B(4)=B(4)+1

 S "=#"
        : 1250 A=0
        : 1260 E=1.25
        : 1270 X=1.0
        : 1290 Y=2
```

Fig. 5. The EDITOR command S "=#" locates and lists lines containing an equals sign immediately followed by a digit. The initial listing is obtained by entering L for LIST.

These are only a few of the ways you can use the EDITOR's searching commands and special characters. If you have an EDITOR ROM pack, you've probably found many other uses; ones that are tailored to your own specific needs. Next month, look for a brief overview of how to perform sorting operations using the EDITOR.

Business Applications Contest

Every field—from designing nails to government management—has methods of tracking, analyzing and plotting the economics of their business. No doubt many 4051 users have put together favorite programs for their day-to-day transactions or those recurring monthly/quarterly analyses.

Well, pull them out, brush them off, get some documentation onto paper, then send them in. The first place winner of the Business Aplications Contest will have his/her choice of 4051 Matrix ROM Pack or a 4051R05 Binary ROM Pack.

In addition to the first prize Matrix or Binary ROM Pack, other winners will receive:

Second Prize:	6 tapes and 6 programs
Third Prize:	4 tapes and 4 programs
Fourth Prize:	2 tapes and 2 programs or a complete set of 4662 plotter pens (13 packages of 3 each)
Fifth Prize:	Roll of Hard Copy Paper or Box of Printer Paper
Sixth Prize:	1 tape or 2 packages of overlays

Every entry will be placed in the 4051 Applications Library. This will entitle you to three programs of your choice from the Library returned on a new tape in exchange for the one sent in. Everyone's a winner!

A few points to guide you in polishing up and documenting your program are listed below. Obviously, they all won't apply in every case; they're simply suggestions from which to start.

Program:	Auto Load file Variables assigned for primary addressess Interactive using the Function Keys and Display Program flow control through Function Keys Data storage possible on tape or disc Provisions for correction of data
Documentation:	Guidelines are available by writing or calling us. They are aimed toward clear, concise documentation with a minimum of writing. We don't want you to get bogged down here.

Submit your program on a tape cartridge or a disc accompanied by the documentation, Program Submittal form, and an Order form for your three exchange programs. We would also like a short biosketch and black and white photograph of each program author, however, these are optional. The TEKniques address is located on the back of each issue.

Deadline for the entries is December 31, 1978.

✻ Editor's Note

1978 Catalog Has 86 Programs

Eighty-six programs are described in the 1978 4051 Applications Library catalog. If you don't have your copy yet, call (503) 682-3411, ext. 2618, or drop us a note. It's free for the asking.

Programming Tips Earn Presentation Aids Program

Do you have a routine you feel is too small to submit to the Library? Send it in as a programming tip. Documentation and listing for program 51/00-9513/0* may be yours free in exchange for your tip.

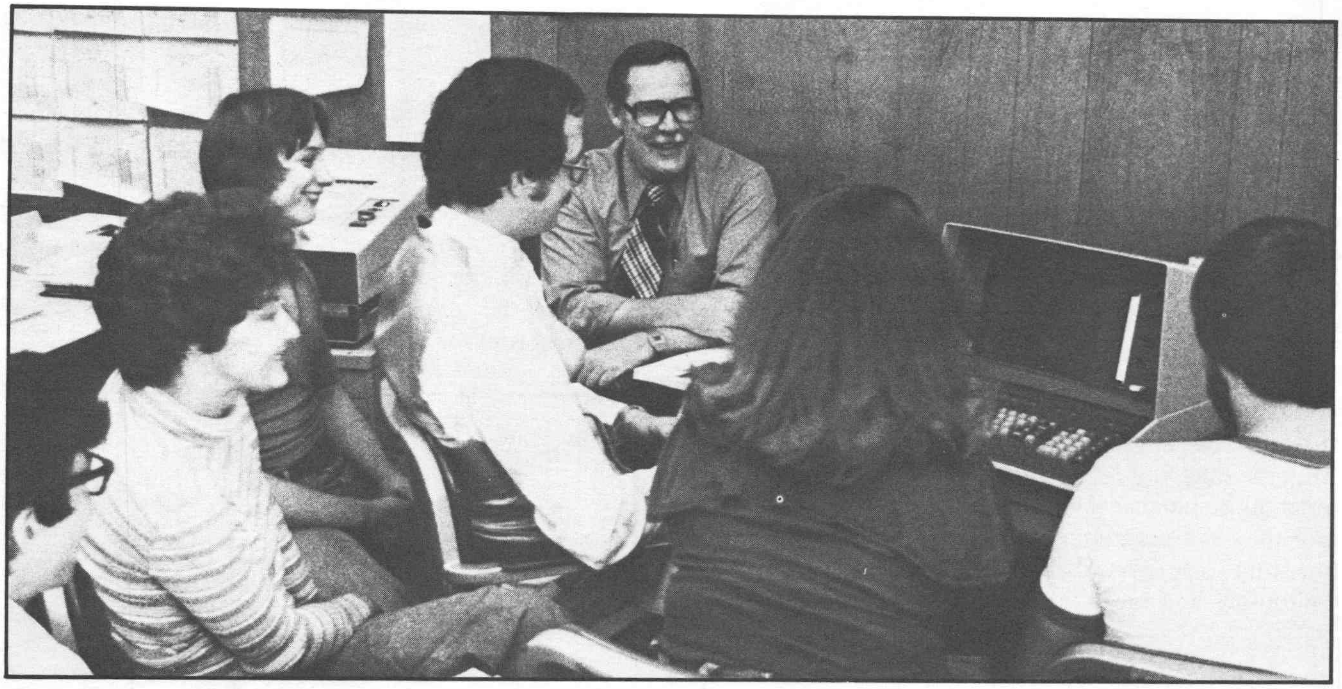
*A different exhange program will be featured in each issue of TEKniques.

Enhancements Welcome

If anyone has refined or enhanced any of the programs from the 4051 Applications Library and would like to pass those changes on to other users, we'd like to hear from you.

Application Author Featured

TEKniques Vol. 2 No. 4 featured an article about Professor Gilbert Pollnow and his program for generating Binary Liquid-Vapor Boiling Point Diagrams. Dr. Pollnow has now contributed this interesting and useful program to the Applications Library. Take a look at Abstract 51/00-5203/0, and TEKniques Vol. 2 No. 4 for more detail.



Dr. P.C. Holman and student teachers learn braille from the 4051.

4051 Prints and Teaches Braille

Braille characters are essentially a graphic dot matrix. Therefore, it seemed only natural to Dr. P.C. Holman to use the graphic display capabilities of the TEKTRONIX 4051 to show the braille character dot matrix on the screen in a new braille teaching application. Dr. Holman, Director of the Offices of Management Information and Institutional Research at the University of Wisconsin-Stevens Point, predicts this new and exciting application will be of great help to teachers who work with the blind, and may even reach out to help the blind themselves.

The braille conversion program was developed to help teachers learn braille for their work with the blind. For that application, the program is working today. It teaches Level 1 braille by converting English letters into the braille dot matrix. Characters typed in from the keyboard are instantly converted to the braille dot matrix and displayed on the graphic screen, providing quick feedback to the student teacher. Program options allow the converted letter to be simultaneously displayed, and, at the operator's option, the letter may be enclosed in a box (Fig. 1).

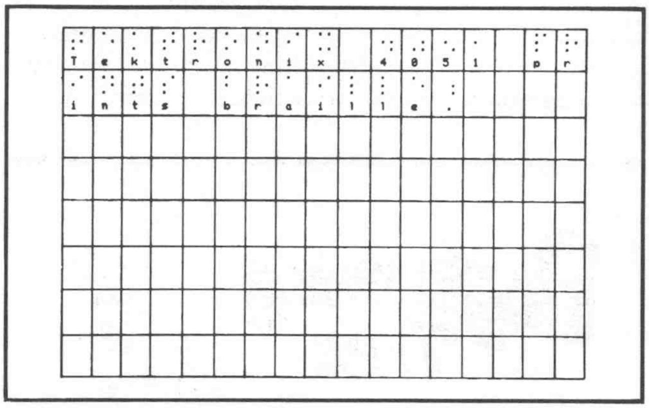


Fig. 1. Characters and their braille equivalent are printed to the 4051 screen.

These techniques permit sighted persons to learn level 1 braille quickly and in ways not previously possible, thanks to the graphics capabilities of the 4051. The same techniques will allow anyone, whether or not they know braille, to convert letters, books, etc., into the braille dot matrix and back again for proofreading.

The work done so far is only the foundation according to Dr. Holman, but it has been carefully laid so that future work can build upon it. "There are whole new possiblities that have been opened for the blind now that we have done the initial work on the (4051 Graphic) system," says Dr. Holman. He believes that books could be produced in braille that contain illustrations like any other books; the technology is already there.

Aside from the possibility of producing raised dot or line pictures of trees, houses, or any subject, there are many other possible applications. A system with the capabilities envisioned could produce raised maps of streets, buildings, floor plans, bus routes, etc., together with braille text.

Says Dr. Holman, "Our technological and people capacities are already very advanced. We now need to bring a large number of blind people to the project to see what they can best understand and use." Work with the blind will ascertain the best ways to make illustrations understandable to the blind. Some engineering will then be required to turn the theory into reality.

For some of the future applications envisioned by Dr. Holman, some specialized equipment will be required, and it has not yet been found. A device must be located that will produce the raised braille dot patterns and raised dot or line pictures that the blind can feel on paper. Such equipment connected to the 4051 could convert any source text into text useful for the blind themselves.

The 4051 is already doing some things in braille that have not been done before. For example, Dr. Holman was struck by the amount of time spent returning the hand from the right end of one line to the left beginning of the other, as he taught himself braille. So, he built an option into the 4051 program that allows printing every other line from right to left. In effect, the braille reader reads a continuous line, never lifting a finger from the page (Fig. 2).

Dr. Holman said he was amazed to find that people could learn to read from right to left as easily as from left to right in just a few minutes, and that reading speed even increased. On the lighter side, he added, "If you want to really go mad in a hurry, try writing from right to left. It's one thing to be able to read from right to left, but it takes a computer to turn lines around and write from right to left."

The work that has been done to date is a real advance, according to Dr. Holman. And it works, which is the real test. Now he would like to see the work continued, but from this point on it will take research funds that he just doesn't have, and a raised dot printer, which has not yet been found, for graphic applications. When those are available, this teaching program will grow quickly into a powerful tool for the blind and their teachers.

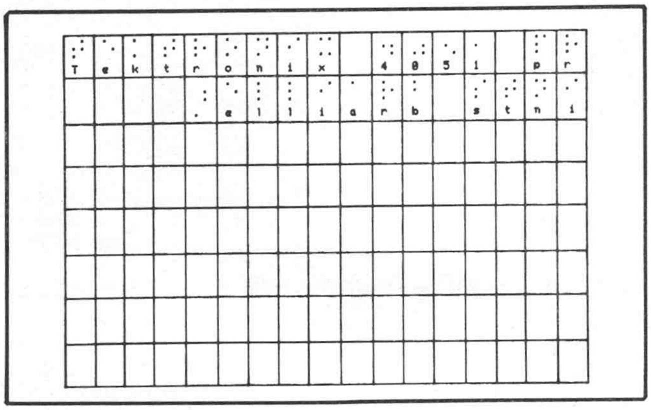
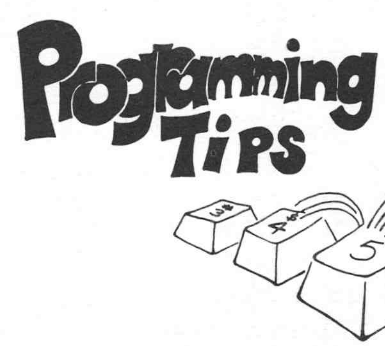


Fig. 2. Every other line may be reversed for continuous reading without lifting a finger.



Structuring the Tape Directory

by Bernard Taieb
Datatek
Badhoevedorp, The Netherlands

File 1 is often reserved for a directory of tape contents so that when the AUTO LOAD key is pressed, a list of the contents of the tape is displayed. The user then may input a number and the directory program will find the correct program and load it.

If the tape contains data files and/or files which are not to appear in the directory, then, since program file numbers are not consecutive, "messy" logical lists can occur in the directory program. Here is a suggestion which eliminates this problem as well as making it much easier to update the directory program when adding new programs to the tape.

```
100 REM DIRECTORY PROGRAM (FILE 1)
110 INIT
120 DIM F(40)
130 DATA 3,4,5,6,7,9,12,1,1,1,1,1,1,1,1,1,1,1,1,1
140 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
150 READ F
200 REM LIST OF PROGRAMS AVAILABLE
210 PAGE
220 PRINT "1...PROGRAM A"
230 PRINT "2...PROGRAM B"
240 PRINT "3...PROGRAM C"
250 PRINT "4...PROGRAM D"
260 PRINT "5...PROGRAM E"
270 PRINT "6...PROGRAM F"
280 PRINT "7...PROGRAM G"
500 PRINT "JEnter Program No.:G"
510 INPUT F1
520 IF F1<1 OR F1>40 THEN 500
530 FIND F(F1)
540 OLD
```

As can be seen, this directory refers to a tape where Program A is in file 3 and Program B is in file 4. Each file number is contained in line 130 as a DATA statement.

If you enter '3' as a response to the question in line 500, the file which is found in line 530 is the third number in the DATA statement, i.e., file 5.

If you enter a number which is outside the range of 1-40, then the question in line 500 is repeated. If you enter a number that's in the range 1-40 but is not in the directory (in this example, numbers 8-40), then the directory program is re-loaded since these programs are not available.

In summary, lines 130 and 140 contain the program file numbers in the order in which lines 200—280 list the program. Updating is now a simple procedure.

To add Program H to the directory (or menu), if program H is stored on file 18, change line 130 to read:

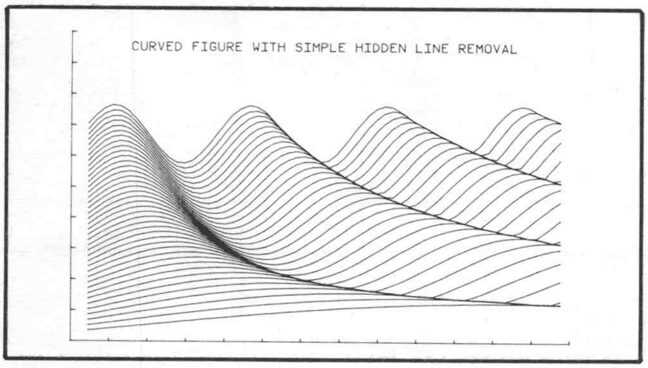
```
130 DATA 3,4,5,6,7,9,12,18,1,1,1,1,1,1,1,1,1,1,1,1
```

and add line 290:

```
290 PRINT "8...PROGRAM H"
```

For users with a binary ROM, certain files may well be binary programs. In this case, when updating lines 130 and 140, put in the file number for ASCII programs but add 100 to the file number for binary programs. So, a binary program on file 20, lines 130 or 140, would contain 120 as the file number. Also include:

```
525 IF F(F1)>100 THEN 600
|
|
600 FIND F(F1)-100
610 CALL "BOLD"
```



Hidden Line Algorithm

by Will Gallant

This is a one line hidden line algorithm for function plotting or nearly any cross section data. This algorithm generates a "propogating horizon." That is, a foreground section is plotted and saved as a horizon. Then the next section is compared to the last horizon and only the parts that stick up above the last horizon are plotted. A new horizon is saved and the process repeats till all cross sections are drawn.

The MAX function is used in this example to compare a cross section to the old horizon and generate a new horizon. Shown below is code to plot a sine wave function "propogating" up the screen by a fixed interval. In practice, the function could be empirically observed statistical data, and the vertical propogating unit the time between readings. Plotting contours of terrain are just as easy, with stations analogous to the vertical screen increment.

To view large dips *under* a function, start a little higher on the screen, add a second horizon, and use the MIN function.

```
100 REM SAMPLE CURVE GENERATOR
110 INIT
120 PRINT " ENTER DEVICE CODE : ";
130 INPUT Z
140 PAGE
150 VIEWPORT 18,122,18,82
160 AXIS @Z:10,10
170 MOVE @Z:20,110
180 PRINT @Z:"CURVED FIGURE WITH SIMPLE HIDDEN LINE REMOVAL"
190 WINDOW 0,60,-5,5
200 VIEWPORT 20,120,20,80
210 K=-5
220 SET DEGREES
230 DIM X(60),Y(60),H(60)
240 H=-10000
250 FOR I=1 TO 40
260 FOR J=1 TO 60
270 X(J)=J
280 Y(J)=SIN(1.2*J+0.5*J*I)+K
290 NEXT J
300 H=Y MAX H
310 MOVE @Z:X(1),H(1)
320 DRAW @Z:X,H
330 K=K+0.175
340 NEXT I
350 PRINT "GGGG"
360 END
```

Correction to 51/00-9513/0 Results in GIN Program Tip

Presentation Aids author Will Gallant alerted TEKniques staff to an incorrect statement in that program. So those of you who have received it by order or exchange, recode statement 2360 to read: GIN @1:X1,Y1 rather than INP @1,24:X1,Y1.

The GIN command returns coordinates expressed in User-Definable Units. The INP command with secondary address of 24 returns coordinates based on Graphic Display Units.

As a quick check of the difference, key in the program below, run it, and note the printed values of X and Y.*

```
100 INIT
110 WINDOW 0,50,0,50
120 DRAW 25,25
130 INPUT @32,24:X,Y
140 PRINT X,Y
150 END
```

Now, change line 130 as below and once again note the values of X and Y.*

```
100 INIT
110 WINDOW 0,50,0,50
120 DRAW 25,25
130 GIN @32:X,Y
140 PRINT X,Y
150 END
```

*It is unlikely that values returned by these commands will exactly match the values in the DRAW command (UDU 25,25 and GDU 65,50 in these cases). The accuracy is determined by the resolution of the graphic device being addressed. For further discussion on this see page 1-60 in the PLOT 50 Introduction to Graphic Programming in BASIC manual.

Correction on Coding Delays

TEKniques Vol. 2 No. 2 published the tip: "Coding Designates Delay." Inadvertently we omitted a space in the code. The correct code should read:

```
PRINT USING "10(110(2("" H"")>)),S":
```

The earlier example does work but it moves the pointer around on the screen.

Our apologies to A.W. Leigh for this error.

YES-NO Branch "Typo" Confuses

A wrong statement number in LeRoy Nollette's "Branching on YES-NO" Tip in TEKniques Vol. 1 No. 8 caused confusion. Statement 450 should have read:

```
500 REM YES PROCESSING
```

Sorry, LeRoy, for our goof.

Also, in this same tip, lines 420 and 430 could be combined if you didn't want to save the value in P.

```
420 GO TO POS("YESNO",A$,1) of 500,400,400,600,400
```

Tablet Input and Pointer Routine Displays Pen Location

by Ed Mitchell

Use this routine in 4051/4956 digitizing applications when you want to see your pen location displayed on the 4051 screen for menu control, placing objects or point-to-point digitizing. Set the Tablet Controller to STREAM SWITCH mode. This allows the 4051 to execute an input instruction and display the location of the pen. Note the "debounce" section of the routine which eliminates multiple entries of the same point.

Lines 130 through 160 set the Graphic Symbols font and set the 4051 window to the maximum tablet coordinates.

Lines 170 and 180 move the pointer to the present pen location and display a "refreshed" cursor. Line 190 checks if the pen (or one of the cursor buttons) was pressed. If it wasn't, another input and cursor operation is performed. If a switch was pressed, "dummy" inputs are assigned to X1,Y1 until the pen is released (status in Y$). This ensures that only one value (the first assigned to X,Y) is accepted as the digitized point.

Lines 240 and 250 are optional and are used to return the true GDU location. INPut @32,24:X,Y performs the same function as the GIN command, however, true GDU locations (0 to 130 and 0 to 100) are returned rather than the "window" location.

Lines 260-400 contain a routine that may be used with the Three-Button Cursor. It decodes which button (flag) was pressed and branches to the appropriate routine.

```
100 REM                         ROUTINE TO POSITION A CURSOR
110 REM
120 REM
130 REM GRAPHICS FONT
140 PRINT @32,18:5
150 WINDOW 0,4000,0,4000
160 INPUT @8:X,Y,Z$
170 MOVE X,Y
180 PRINT @32,24:"|"
190 GO TO Z$="0" OF 160
200 INPUT @8:X1,Y1,Y$
210 GO TO Y$="0" OF 230
220 GO TO 200
230 REM FOR 0 TO 4000 BRANCH TO 260
240 MOVE X,Y
250 INPUT @32,24:X,Y
260 REM ******************** DECODE SWITCH ******************
270 GO TO LGT(VAL(Z$))/LGT(2)+1 OF 290,380,350,320
280 REM LINE 270 USED FOR 3 BUTTON CURSORS
290 REM SWITCH 1 PRESSED
300 PRINT "0"
310 GO TO 160
320 REM FLAG 1 PRESSED
330 PRINT "1"
340 GO TO 160
350 REM FLAG 2 PRESSED
360 PRINT "2"
370 GO TO 160
380 REM FLAG 3 PRESSED
390 PRINT "3"
400 GO TO 160
```

Finding the Size of an Array

By Ed Mitchell

Did you ever dimension an array and then forget how large it was? The SUM function is one way to find out. For example, B is a dimensioned variable with unknown size. To find the size, set all elements of B equal to 1 and sum them:

```
B = 1
SUM B
```

The number returned is the number of elements in array B.

Note: This operation replaces any data in array B with 1's.

Default Response on YES/NO Branch

by Bernard Taieb

Datatek
Badhoevedorp, The Netherlands

In a POSition function a null string returns the location where the search began. In the case of P=POS ("ABCDEF",A$,4), if A$ is a null string, 4 will be assigned to P.

Turn this to your advantage using RETURN as a default response when branching on a YES/NO reply. Pressing the RETURN key without an entry assigns a null string to the target string variable. Therefore, if you pressed the RETURN key in response to the question in statement 1000, you would branch to statement 2000 since a 1 would be assigned to P. (Note the underline coding in statement 1000 indicating the default response.)

```
1000 PRINT "DO YOU WANT TO CONTINUE (YESHHH___OR NO)? ";
1010 INPUT A$
1020 P=POS("YESNO",A$,1)
1030 GO TO P OF 2000,1000,1000,3000,1000
1040 GO TO 1000
2000 REM YES PROCESSING
3000 REM NO PROCESSING
```

To default to NO processing, simply change the default underline in statement 1000, reverse the string in statement 1020 and the target lines in statement 1030.

```
1000 PRINT "DO YOU WANT TO CONTINUE (YES OR NOHH__)? ";
1010 INPUT A$
1020 P=POS("NOYES",A$,1)
1030 GO TO P OF 3000,1000,1000,2000,1000
1040 GO TO 1000
2000 REM YES PROCESSING
3000 REM NO PROCESSING
```

4051 Applications Library Program Abstracts

Order

Documentation and program listings of each program are available for a nominal charge. Programs will be put on tape for a small recording fee per program plus the charge for the tape cartridge. One tape will hold several programs. (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.)

Domestic U.S. Prices:

Documentation and listings	*$15 per program*
Recording Fee	*2 per program*
Tape Cartridge	*26 per tape*

Contribute

Contribute one program to the Library and receive three in exchange. Send in the membership card from your 4051 Graphic System Reference Manual to get the details. Or call us (503) 682-3411, ext. 2618.

Forms

Please use the Applications Library Order Form. Order forms are included in the Membership Packet and are available from your local Tektronix Sales Engineer.

Outside U.S.

Program contributions or orders outside the U.S. must be processed through the local Tektronix sales office or sent to one of the Libraries serving your area. See Library Addresses section.

ABSTRACT NUMBER: 51/00-0715/0

Title: **Measures of Central Tendency**
Author: Dennis Heckman
Tektronix, Inc.
Memory Requirement: 16K
Peripherals: Optional-4641 Line Printer
Optional-4662 Plotter
Statements: 286
Files: 2

The program computes five measures of central tendency from weighted or unweighted data. Measures of central tendency include the Median, Harmonic Mean, Geometric Mean, Mean, and Root Mean Square.

A histogram of the computational data is also prepared as are tables of deciles and quartiles for the data.

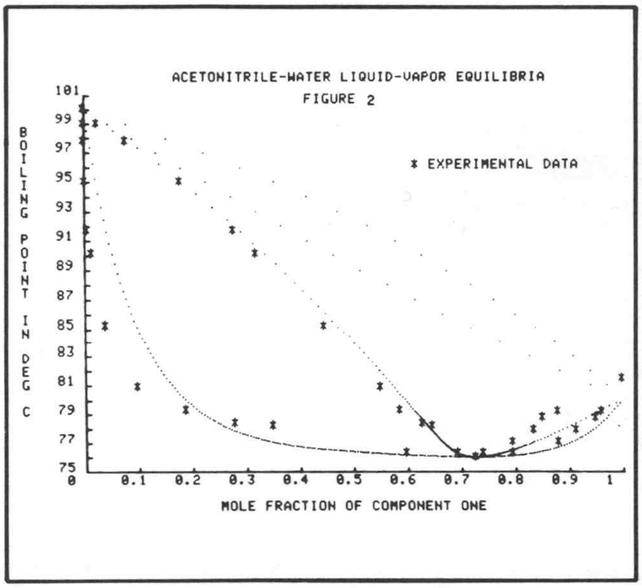
ABSTRACT NUMBER: 51/00-5203/0

Title: **Simulation of Binary Liquid-Vapor Boiling Point Diagrams**
Author: Gilbert F. Pollnow, Ph. D.
Dept. of Chemistry
Univ. of Wisconsin-Oshkosh
Memory Requirement: 16K
Peripherals: 4631 Hard Copy Unit
Statements: 265

The program computes both the ideal and Van Laar approximations to the liquid and vapor compositions for any binary pair for which the Antoine or Clausius-Clapeyron vapor pressure equation constants are available along with the azeotropic mole fraction of either component, barometric pressure, and boiling point of the azeotrope.

Experimental liquid-vapor composition data as a function of temperature will also be plotted, if loaded into the appropriate DATA statements within the program.

A complete data set for the system acetonitrile-water is included within the program by way of illustration. Tabular as well as graphical output is optional at your discretion.



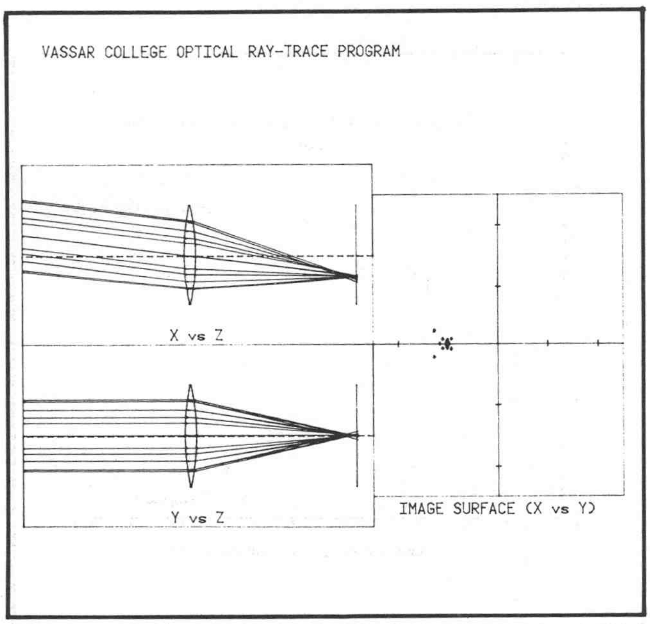
ABSTRACT NUMBER: 51/00-5404/0

Title: **Optical Ray-Trace**
Author: James L. Hutchinson
Dept. of Physics/Astronomy
Vassar College
Poughkeepsie, NY
Memory Requirement: 32K
Peripherals: 4662 Plotter
Statements: 553

The program consists of two parts. The first part calculates the actual three-dimensional paths of light rays through a user-specified system of spherical or planar surfaces, lenses, mirrors or stops. A plot of the system is made in plane and elevation to show the paths of the rays along the Z-axis. Light rays proceed through the system normally from left to right and can be plotted in contrasting colors. The rays are followed until they intersect the image surface or a stop, or until they fail to intercept a subsequent optical surface. Distance units are arbitrary; the program plots the system to scale. A plot of the intersection of the rays with the chosen image surface is also given.

The rays traced through the system can be chosen manually or automatically. In the manual mode, you must separately specify the orientation of each incoming ray. In the automatic mode, rays are chosen from a standard set of 21 rays distributed over four zones of the first optical surface. The end results of the calculations can be saved internally in the program or on tape (manual mode only) to allow further investigation of the structure of the image.

In the second part of the program, you can select different locations of the image surface to see how the presence of aberrations affect the quality of focus of the image.



ABSTRACT NUMBER: 51/00-5721/0

Title: **Scheffe′ Multiple Comparison Procedure**
Author: Glenn Galfond
Patuxent Wildlife Research
Laurel, MD
Memory Requirement: 16K
Peripherals: Optional—4631 Hard Copy Unit
Statements: 324
Files: 2

This program allows the user to perform the Scheffe′ multiple comparison procedure to separate linear model parameters in an analysis of variance (see Scheffe′, The Analysis of Variance, pps. 68-72).

File one contains a program which allows the user to enter and conveniently edit the covariance matrix. The matrix is then stored on tape.

File two contains the multiple comparison procedure. The user enters the coefficient vector, the desired confidence level, and the file number on which the covariance matrix is stored. The analysis returns the half-length of the confidence interval.

```
INPUT THE COEFFICIENT VECTOR, SEPARATING EACH ENTRY
BY A COMMA. ( DIMENSION = 3 )
1,0,-1

 1, 0, -1,

INPUT THE DESIRED LEVEL OF SIGNIFICANCE: .05

INPUT THE DIMENSION OF THE HYPOTHESIS SPACE: 3

INPUT THE NUMBER OF OBSERVATIONS: 50

INPUT THE RANK OF THE DESIGN MATRIX: 2

INPUT THE ERROR MEAN SQUARE: .14

THE HALF LENGTH OF THE CONFIDENCE INTERVAL IS 0.792477877799

DO YOU WISH TO INPUT ANOTHER COEFFICIENT VECTOR
FOR USE WITH THIS COVARIANCE MATRIX ? (YES OR NO) YES
```

ABSTRACT NUMBER: 51/00-8020/0

Title: **Software Character Plotting**
Author: Trevor Blogg
Calgary, Alberta, Canada
Memory Requirement: 8K
Peripherals: Optional-4631 Hard Copy Unit
Statements: 116 plus Data File

This is a subroutine that plots a limited character set, such as upper case, numerics, common symbols and one special plotting character, with a memory overhead of 3K or less. Character size is determined by the existing scaling factor, and characters are rotated to the current graphic rotation.

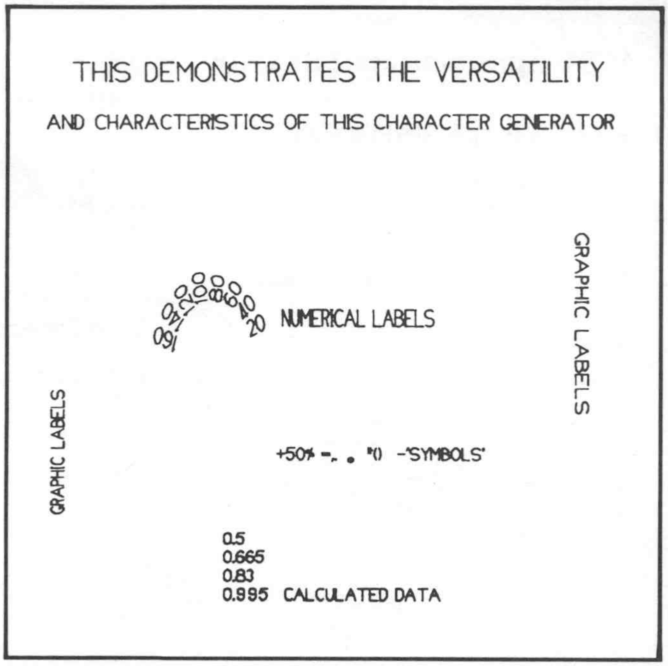
String variables are "printed" with proportional spacing.

"Normal" aspect ratio is achieved with equal X and Y scaling factors. The routine uses a data list on tape which is accessed once at the beginning of the user program.

This character generator allows a programmer to add titles, rotated labels, etc., to plots which are to be subsequently committed to hard copy. If the hard copy device is a 4662 Plotter, then device addresses may be added to the RMOVE and RDRAW commands.

In use the programmer SCALES the display or plotter, either using the suggested subroutine, directly or by use of WINdow and VIEwport commands. The cursor is MOVed to locate the start of the label. The data is then loaded into the string G$, either literals or STRing functions may be employed, then the program is branched to the subroutine. Non-plotting characters will cause blanks to be inserted.

Characters are 20 G.D.U.'s in height, and are of proper width when equal X and Y scale factors are used.



4051 Applications Libraries

Africa, Europe, Middle East

4051 Applications Library
Tektronix Datatek N.V.
P.O. Box 159
Badhoevedorp, The Netherlands

Australia

4051 Applications Library
Tektronix Australia Pty. Limited
Sydney
80 Waterloo Road
North Ryde, N.S.W. 2113

Canada

4051 Applications Library
Tektronix Canada Ltd.
P.O. Box 6500
Barrie, Ontario
Canada L4M 4V3

Caribbean, Latin America and Far East (excl. Japan)

Ms. Bev Brandon, 73-312
Export Marketing
Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077
U.S.A.

Japan

4051 Applications Library
Sony/Tektronix Corporation
9-31 Kitashinagawa-5
Tokyo 141 Japan

United States

4051 Applications Library
Tektronix, Inc.
Group 451
P.O. Box 500
Beaverton, Oregon 97077

Tektronix®
COMMITTED TO EXCELLENCE

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

ADDRESS CORRECTION REQUESTED