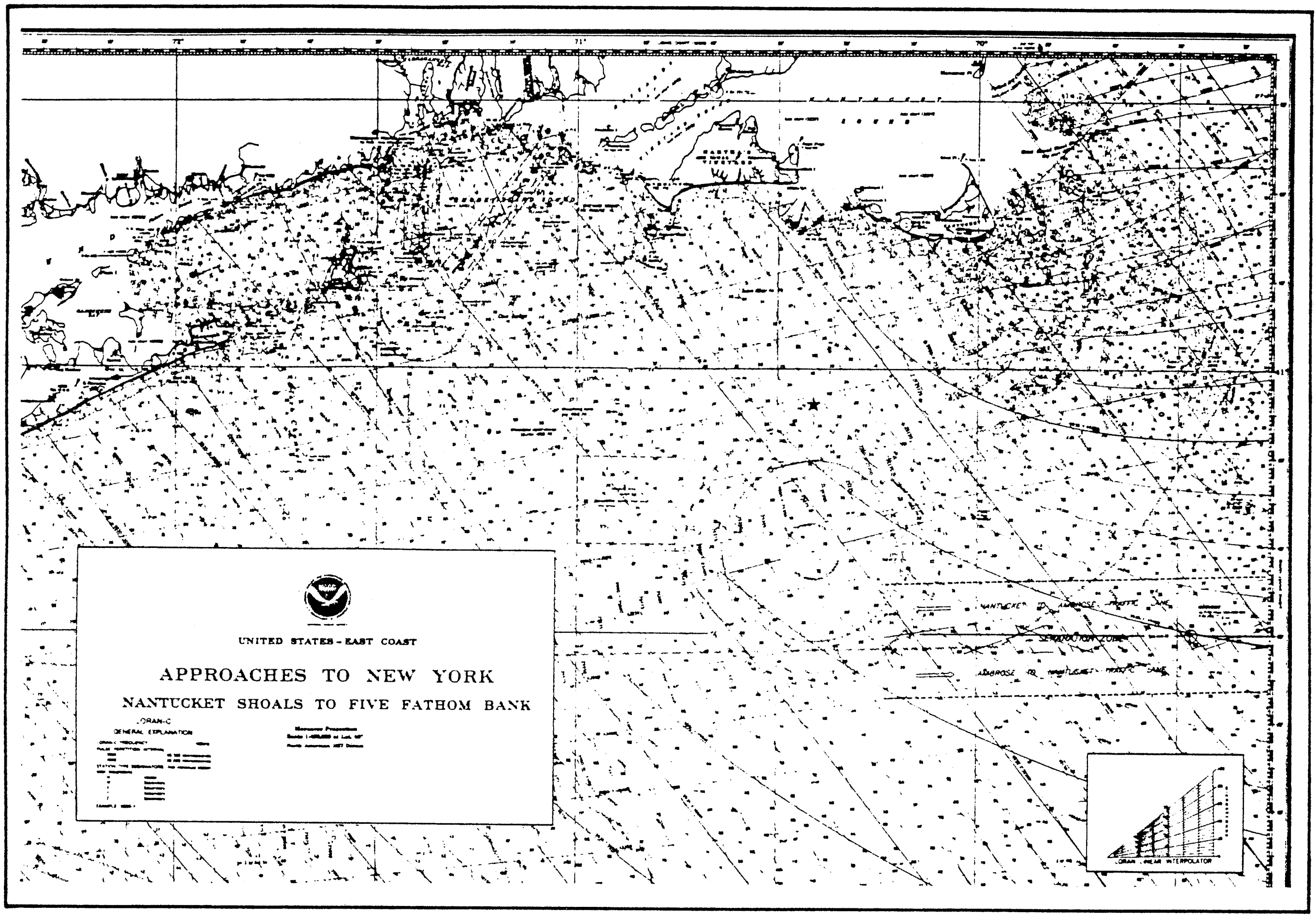
Tekniques

The 4050 Series Applications Library Newsletter November 1, 1979 Vol. 3 No. 7


UNITED STATES - EAST COAST
APPROACHES TO NEW YORK
NANTUCKET SHOALS TO FIVE FATHOM BANK

A LORAN-C chart correlates radio signals transmitted from special shore transmitting stations to a ship's position. The DOT Transportation Systems Center in Massachusetts is assessing the usefulness of LORAN-C for on-land site location using the 4051 Graphic System as the experiment controller.

A Recent Development in LORAN-C Data Acquisition, Aided By The 4051

by Peter Mauro
DOT/Transportation Systems Center
Cambridge, MA

The use of LORAN-C systems, or LOng RAnge Navigation systems, has grown in the past decade as a valuable aid to maritime navigation. Its users range from supertankers to small fishing craft. Now the Transportation Systems Center of the Department of Transportation (DOT) is evaluating the usefulness of LORAN-C for on-land site location and registration, and for Automatic Vehicle Monitoring (AVM). This research is being sponsored by the DOT's Research and Special Programs Administration, to see if the LORAN-C maritime locator can be used to pinpoint, say, an accident location for emergency vehicles (among many other projected uses). To perform this experiment, a data acquisition van has been designed using the 4051 Graphic System as the experiment controller (Figure 1).

In This Issue

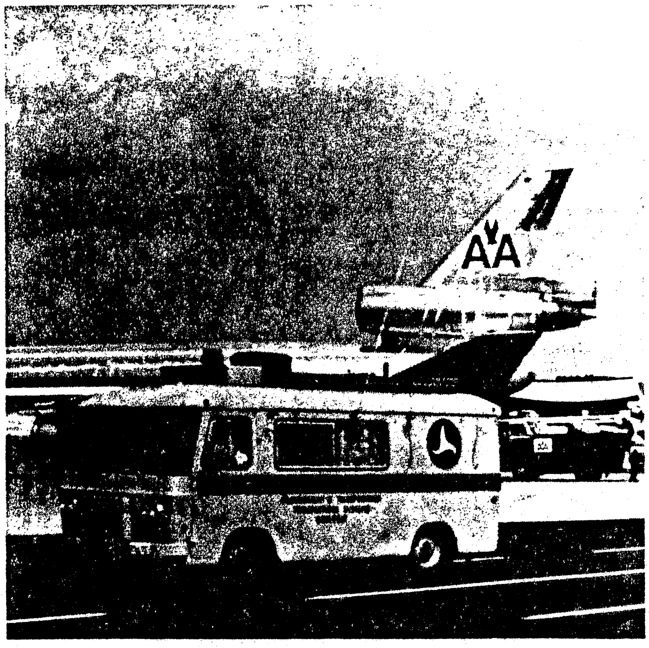


Fig. 1 Traversing the highways with their mobile data acquisition van, which includes a 4051 Graphic System, the U.S. Department of Transportation is evaluating the use of LORAN-C signals for on-land site location.

What is LORAN-C

LORAN-C is a pulsed low-frequency hyperbolic radionavigation system. It derives its high accuracy from time difference measurements between pulsed signals, and from the inherent stability of low-frequency propagation over seawater. It has developed in recent years as a highly accurate maritime navigational aid.

Hyperbolic navigation systems operate on the principle that the time difference between the arrival of signals from two secondary (slave) stations, observed at any point in their coverage area, is also a measure of the distance to each of the stations. The LORAN-C primary (master) station serves as a master time reference, eliminating the need for an on-board precision clock to measure time difference. All of the points that have the same observed difference in distance from a pair of stations defines a hyperbola, called a Line of Position (LOP). The intersection of two or more LOPs precisely defines the position of the LORAN-C receiver.

TEKniques, the 4050 Series Applications Library Newsletter, is published by the Information Display Division of Tektronix, Inc., Group 451, P.O. Box 500, Beaverton, Oregon 97077. It is distributed to TEKTRONIX 4050 Series users and members of the 4050 Series Applications Library.

Publishing Manager	Ken Cramer
Managing Editor	Patricia Kelley
Editor	Terence Davis
Technical Editor	Dan Taylor
Graphic Design	John Ellis
Circulation	Rory Gugliotta

The accuracy of any hyperbolic navigation system depends on the observer's ability to measure the difference between the arrival time of two signals, and the observer's knowledge of the propagation conditions. This latter information allows the measured time differences to be converted to LOPs.

The basic *limitation* on accuracy is knowledge of the velocity of radio wave propagation (about one foot per nanosecond). To achieve accuracies of ten to hundreds of feet, time measurements must be accurate to from ten to hundreds of nanoseconds. Also (and especially important to terrestrial uses), the propagation condition must also be reliably predictable to ten or hundreds of nanoseconds. Since the velocity of progagation will vary with seasonal changes in ground conductivity, considerable investigation will be required before time delays can be reliably predicted for terrestrial use.

Expanding Onto Land

The expanding LORAN-C grid on both the East and West Coasts, along with the possible addition of a mid-continent chain, has made LORAN-C position location available to a larger user group than the maritime community it was originally designed to serve. In addition, new LORAN-C receivers with advanced microprocessor technology are smaller and lighter, and capable of improved operation in the difficult environments found in large cities, near power lines, etc. These factors have generated interest in a number of potential applications.

One example of a potential use is in state traffic departments. Using LORAN-C, an accident site on a highway could be precisely identified to direct rescue vehicles. The same information could be used to accumulate a data base for future traffic pattern studies. Police units could use LORAN-C coordinates to direct officers in rural areas. Conversely, the system could be used in Automatic Vehicle Monitoring, where an officer's location and status might be transmitted to a dispatch location in a high-crime area. Another example of a unique terrestrial application of LORAN-C is in population location information gathered by the Census Bureau in remote areas where no address is possible.

All of the above examples depend on the reliable and repeatable precision of LORAN-C coordinate data, over both short and long periods of time (minutes or years). The Department of Transportation experiments described here are designed to determine variations in terrestrial LORAN-C data. Long-term seasonal changes due to ground conductivity (fluctuating because of the moisture content) are being checked, along with short-term jitter from man-made and natural interference. The van shown in Figure 1 is a mobile data collection facility that can be driven along a roadway while recording LORAN-C coordinates, odometer-measured distance, and time, all on tape as raw data. The on-board 4051 Graphic System then performs statistical analysis of the data while enroute, enabling a quick look at experimental results before terminating a test run.

The Experiment

The experiment is designed to automatically collect data about the variation of LORAN-C coordinates with time, along a given highway. The route will be traversed many times during the course of the experiments; about 5000 data points are collected during each run. Manual data recording would be tedious and error-prone. Hence, a method of collection, recording, and reduction under computer control was designed, using the 4051 as the controller. And the 4051's graphic display permits easy comparison of data from multiple runs over the same route. Data must be collected over a period of time to verify a predictable drift in coordinates due to seasonal effects. Suitable correction can then be made for the seasonal effects.

The Equipment

Figure 2 is a block diagram of the equipment configuration; the actual equipment is shown within the van in Figure 3. There the 4051 can be seen as the system's automatic controller, operating from a "real-time" BASIC program stored in 30K of RAM. Data sampling occurs at a four-second repetition rate, on command from a LORAN-C receiver. All data is loaded in parallel to a shift register that can accommodate all sources simultaneously, assuring accurate tracking of distance, time, and LORAN-C coordinates.

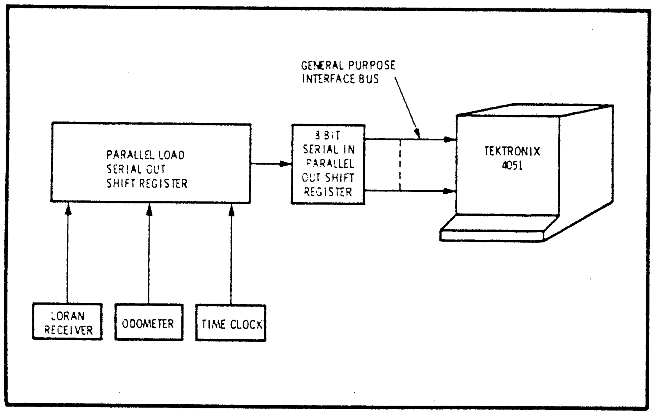


Fig. 2 LORAN-C experiment configuration.

The equipment operates in the following manner: during the four-second interval between sample commands, the data is formatted into eight-bit bytes and transmitted over the GPIB to the 4051. The 4051 then processes and records the data before the next sample command; recording on magnetic tape allows further analysis back in the lab. The software allows the operator to control navigation system mode, data gathering memory, inspection, and other system functions, all through User Definable Keys. This makes the complex, advanced test system simple and easy to use.

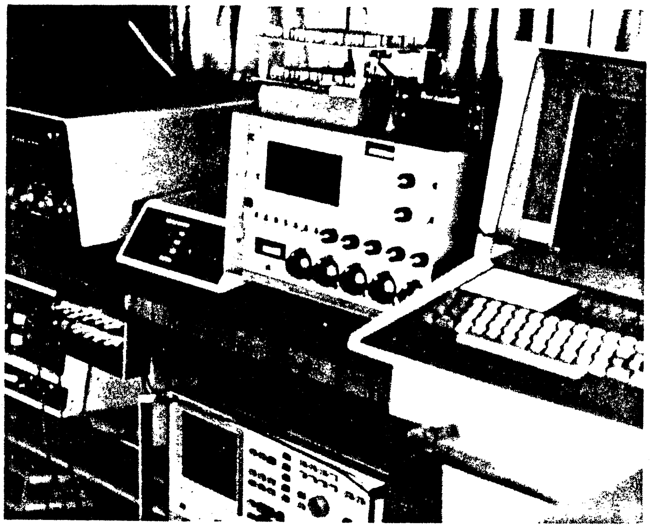
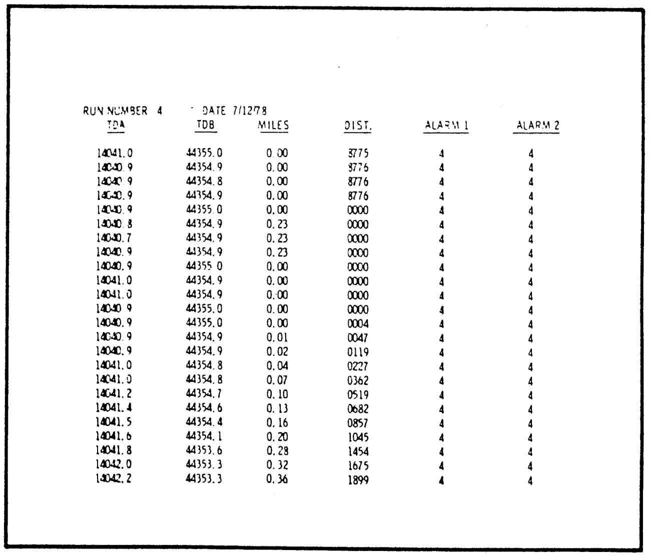


Fig. 3 The 4051 Graphic System controls data acquisition in the DOT van. LORAN-C receivers acquire the signals.

Data Reduction and Presentation

The Northeast Coast chain has four operational secondary stations; they're located at Caribou, Maine; Carolina Beach, North Carolina; Dana, Indiana; and Nantucket, Massachusetts. The master station for this chain is located at Seneca, New York. The two secondaries whose LOP's cross the Boston area at nearly right angles are Caribou and Carolina Beach. The Nantucket transmitter provides an exceptionally strong signal, but those with optimum crossing angles are preferred.

The data shown in Figure 4 is displayed in tabular form on the 4051 Graphic System screen. This allows the results to be monitored during the progress of the experiment.



RUN NUMBER 4 DATE 7/12/78

TDA	TDB	MILES	DIST.	ALARM 1	ALARM 2
14041.0	44355.0	0.00	8775	4	4
14040.9	44354.9	0.00	8776	4	4
14040.9	44354.8	0.00	8776	4	4
14040.9	44354.9	0.00	8776	4	4
14040.9	44355.0	0.00	0000	4	4
14040.8	44354.9	0.23	0000	4	4
14040.7	44354.9	0.23	0000	4	4
14040.9	44354.9	0.23	0000	4	4
14040.9	44355.0	0.00	0000	4	4
14041.0	44354.9	0.00	0000	4	4
14041.0	44354.9	0.00	0000	4	4
14040.9	44355.0	0.00	0000	4	4
14040.9	44355.0	0.00	0004	4	4
14040.9	44354.9	0.01	0047	4	4
14040.9	44354.9	0.02	0119	4	4
14041.0	44354.8	0.04	0227	4	4
14041.0	44354.8	0.07	0362	4	4
14041.2	44354.7	0.10	0519	4	4
14041.4	44354.6	0.13	0682	4	4
14041.5	44354.4	0.16	0857	4	4
14041.6	44354.1	0.20	1045	4	4
14041.8	44353.6	0.28	1454	4	4
14042.0	44353.3	0.32	1675	4	4
14042.2	44353.3	0.36	1899	4	4

Fig. 4 LORAN Experiment Data.

In the table, the first two columns are the time difference measurements, in microseconds, between the master station and secondary A (TDA) and secondary B (TDB). Column three is the odometer-measured distance to one-hundredth of a mile. (Note that a stationary vehicle produces a reading of zero.) Column four is the odometer readout in feet; it appears on the data printout and on the odometer electronic display simultaneously.

Columns labled Alarm 1 and 2 are status indicators for the LORAN-C receiver. A number is displayed for every possible operating mode of the receiver; Alarm 1 is associated with TDA and Alarm 2 with TDB. The indicator number may range from one to 128 to show the status of the transmitter or receiver equipment. In the example shown, the number 4 indicates that both transmitter and receiver are working correctly.

Preliminary Results

A pilot demonstration was conducted to 1) gain an understanding of the concept of relating accident locations to highway characteristics; 2) verify the procedures for linking accident locations to highway characteristics using LORAN-C; and 3) demonstrate the usefulness of LORAN-C coordinates for location identification.

This pilot experiment was conducted in Columbia County on Routes 23 and 9 near the town of Hudson, New York, during the fall of 1977. Data was recorded manually and reduced manually at a plotting facility of the Transportation Systems Center. The pseudo road map in Figure 5 was generated there. The map approximates the actual highway along which the data was collected, but the jitter in the lines clearly indicates that some form of smoothing is required. The scale is approximately one nanosecond per foot, which is 50,000 feet (9.4 miles) between Lines of Position.

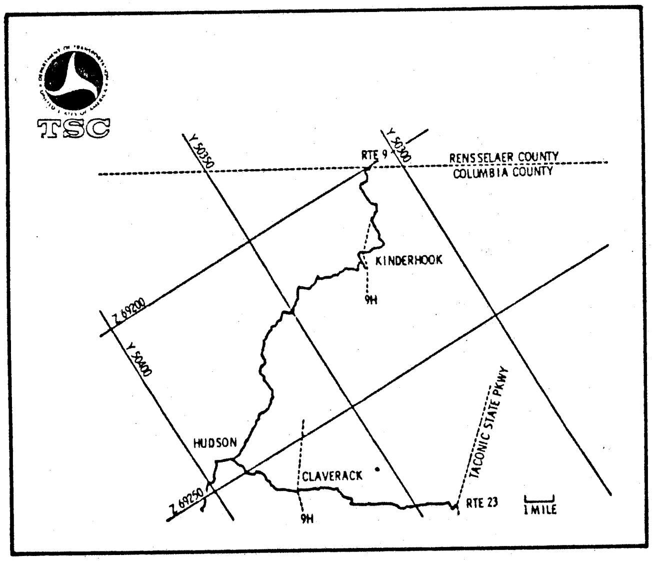


Fig. 5 LORAN-C Pilot Experiment—New York

More Plans

The results of the pilot experiment encouraged more-detailed investigation into terrestrial use of LORAN-C. The 4051-based system now in use makes the data gathered more immediately useful through real-time analysis and reduction. Real-time monitoring, along with graphically displaying key variables in navigation data, allows the test procedure to be modified on the spot, depending on the results observed.

Figure 6 is an example of a pseudo map generated on the 4051 while enroute on Highway 93 in the Boston area. For this experiment, the old East Coast LORAN-C stations were chosen (Dana, Indiana, and Nantucket, Massachusetts, with the master station located at North Carolina). These secondaries give a less-than-perfect right angle crossing, but are adequate for the initial trial. As this experiment progressed, it quickly became clear where LORAN-C reception was difficult. Scale on this map is 5 microseconds between grid lines, or approximately 5,000 feet.

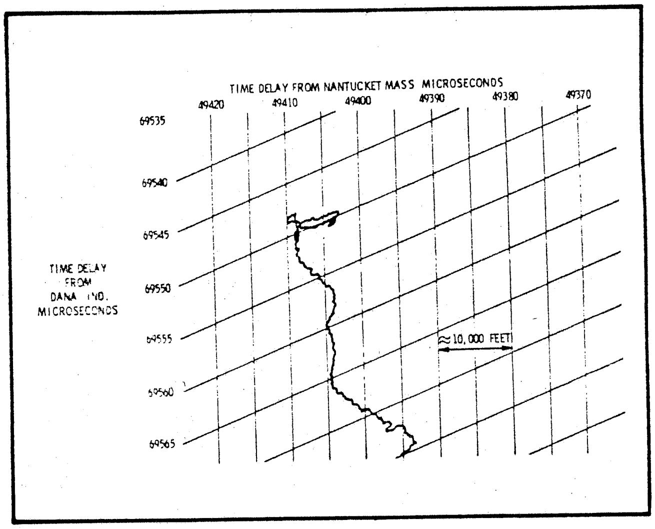


Fig. 6 LORAN Experiment—Boston Area.

The 4051 software support programs enable complete statistical analysis of the data collected at a fixed location, including mean, variance, standard deviation, skewness, kurtosis, etc. But the data collected in this experiment is deliberately not stationary, and some form of smoothing is required.

Programs are currently being investigated for the mobile data, where samples are averaged and subtracted from a straight line, to display a smoothed curve that approximates the highway as shown on a road map. Hard copies of the pseudo road maps generated for each run will show shifts in the LORAN-C grid when superimposed on a light table. Tabular results will enable a correction factor to be developed for seasonal changes, if necessary.

Many other potential uses for LORAN-C are also being considered for investigation; this research will be the basis for future developments. And at the heart of the data gathering and evaluation process is the 4051 Graphic System, making the system easy to use and, through immediate graphic feedback, quick to evaluate.

ROM Pack Automatically Restarts 4051 Program After Temporary Power Interrupt

by Patricia Kelley
TEKniques Staff

The National Research Council of Canada at Ottawa has engineered a circuit which automatically restarts their 4051-based data acquisition system after a power failure. W. Budde, L. Carson and P. Grant, Optics Section of the Division of Physics, documented the procedure and sent it in to TEKniques. Unfortunately, their design and build work wasn't necessary; Tektronix has had an Auto-Start ROM Pack to perform just such a function since 1976. When we told them, they stated: "We did some minor crying .... The Auto-Start ROM Pack was indeed not known and some more publicity is advisable."

The National Research Council uses the 4051 to control a complex data acquisition system for the fully automatic recording of solar radiation and weather data. The system, which incorporates a scanner, digital voltmeter, counter, and various stepper motors, runs continuously without operator supervision. The loss of a few data values during a power failure was not a serious problem; the problem was beginning the data acquisition program once again when power was restored, particularly if the failure occurred early during a weekend or even early at night.

The AUTO-START ROM Pack, part number CM 020-0198-00, is a custom modification designed for such circumstances as the Council described. It allows the 4051 to execute an inserted program tape when power is first applied; it isn't necessary to press the AUTO LOAD key when the ROM Pack is installed. Each time power is discontinued then re-applied to the 4051, the system will rewind the magnetic tape, locate the first program file, load the file, and execute the program.

Perhaps this ROM Pack can help you overcome a power failure-restart problem. If you would like more information, contact your local Tektronix Sales Engineer and ask about *custom mod* Rom Pack, CM 020-0189-00, for the 4051.

Editor's Note

New Members Wanted

We're looking for 4050 Series System users who are not yet members of the Applications Library. There are membership cards included with this issue, which you can pass on to your colleagues who might be interested. And don't forget to tell them, IT'S FREE!

Back Issues

As a reminder, we continue to keep back issues of TEKniques available for those who haven't received them or have misplaced them. Some of the earlier issues may be reproductions rather than the original printings. Just drop us a note if you need any back issues.

Library Catalogs are Available

The new Applications Library catalog is available for the asking, too. It not only contains lots of programs, with descriptions of their capabilities, but gives you an idea of the versatility of the 4050 Series Graphic Computing System. Just let us know if you need a copy.

Remember our Questionnaire?

In TEKniques Vol. 3 No. 5 we included a questionnaire about TEKniques and the applications library. If you haven't replied, please take a few minutes and let us hear from you. We'd like to know your thoughts, good or bad.

New Contest

This issue features a new 4050 Series Applications Library contest, described in detail elsewhere in this issue. There's lots of time to enter, so check it out.

EMC Customers Take Note

There's a new address for EMC customers to write to the Applications Library. Note the address block at the back of this issue.

Programming Tip Exchange

Send in your programming tip. Anyone of the following 4051 Applications Library programs* will be yours when it's published. Simply jot down a brief description of the function, the code, and your choice of program. Mail it to the 4050 Series Applications Library serving you; Library addresses are listed at the back of each TEKniques issue.

51/00-0101/0
51/00-0702/0
51/00-0715/0
51/00-1401/0
51/00-1402/0
51/00-5401/0
51/00-5503/0
51/00-7002/0
51/00-8006/0
51/00-9505/0
51/00-9511/0
51/00-9521/0

***Documentation and listing only.**

Interfacing is Contest Theme

Data acquisition or instrument control through the 4050 Graphic System is the theme of the 4050 Series Applications Library contest. Awards will be made in three categories.

Rules

- The application must include a 4050 Series System interfaced to at least one instrument. This instrument must be in addition to any 4050 Series peripheral*, although the latter may be part of the configuration.
- The program must be submitted on tape or disc and include the documentation, completed and signed Submittal Form, and an Order Form with your exchange selection.
- More than one application may be contributed.
- An application may only be entered into one category.

Categories

The categories are divided by interface.

Category I — Application using the General Purpose Interface Bus (GPIB)

Category II — Application using the Option 1 Data Communications Interface (RS-232)

Category III — Application using any other type of interface—commercially constructed or "home built."

Awards

Awards will be given to three places in each category. Each winner will have his/her choice of 4050 Series ROM Packs, PLOT 50 software, or Applications Library software, at catalog price, up to the award amount for his/her placing.

	Category I	Category II	Category III
1st Place	$750	$750	$750
2nd Place	500	500	500
3rd Place	375	375	375

All entrants will receive three programs in exchange for theirs. This exchange, of course, is in addition to any award that might be received.

Deadline

As we promised in the last issue of TEKniques, you'll have lots of time to work on your program and its documentation. The contest deadline is March 31, 1980.

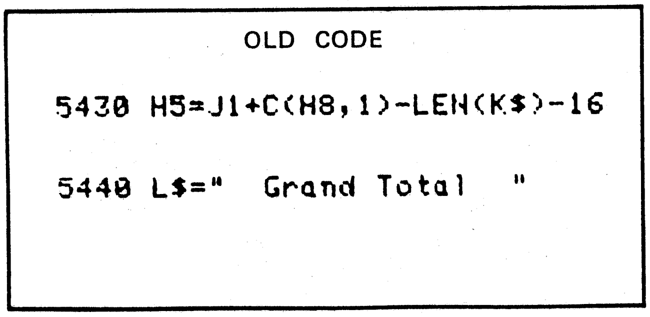
So, clean-up those programs, complete the documentation, and send it in. If you need documentation guidelines or any of the forms, send us a note; our address is shown under **United States** on page 16

*4050 Series peripherals include 4610, 4631, 4641, 4642, 4662, 4663, 4907, 4924 and 4956.

Updates

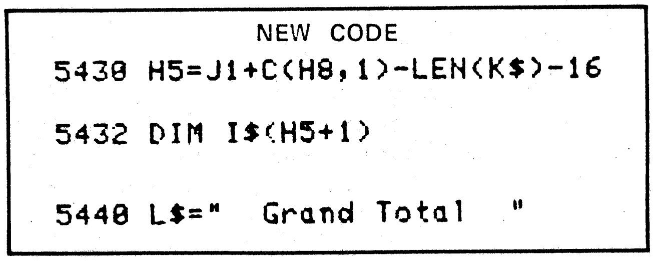
Recordkeep II

Abstract 51/07-6108/0 described RECORDKEEP II in Vol. 3 No. 3 issue (May 1, 1979) of TEKniques. Those who received this program from the 4050 Series Applications Library prior to July 1 should check to see if the following line of code was added in **OVERLAY 6 (0V6):**



OLD CODE

```
5430 H5=J1+C(H8,1)-LEN(K$)-16

5440 L$="  Grand Total   "
```



NEW CODE

```
5430 H5=J1+C(H8,1)-LEN(K$)-16

5432 DIM I$(H5+1)

5440 L$="  Grand Total   "
```

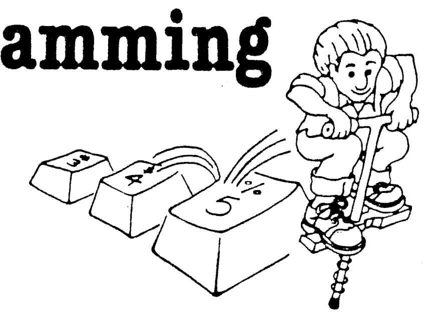
Presentation AIDS—51/00-9513/0

Presentation Aids, a popular slide maker program in the 4050 Series Applications Library, doesn't allow the user to stop and change pens during the slide making process. Two lines of code submitted by Hank Piatek, Tektronix, Inc., Wilsonville, halt program execution while you change pens.

```
1005 INPUT Q$
1245 INPUT Q$
```

Once your new pen is in place (or if no change is desired), simply press RETURN and the program continues.

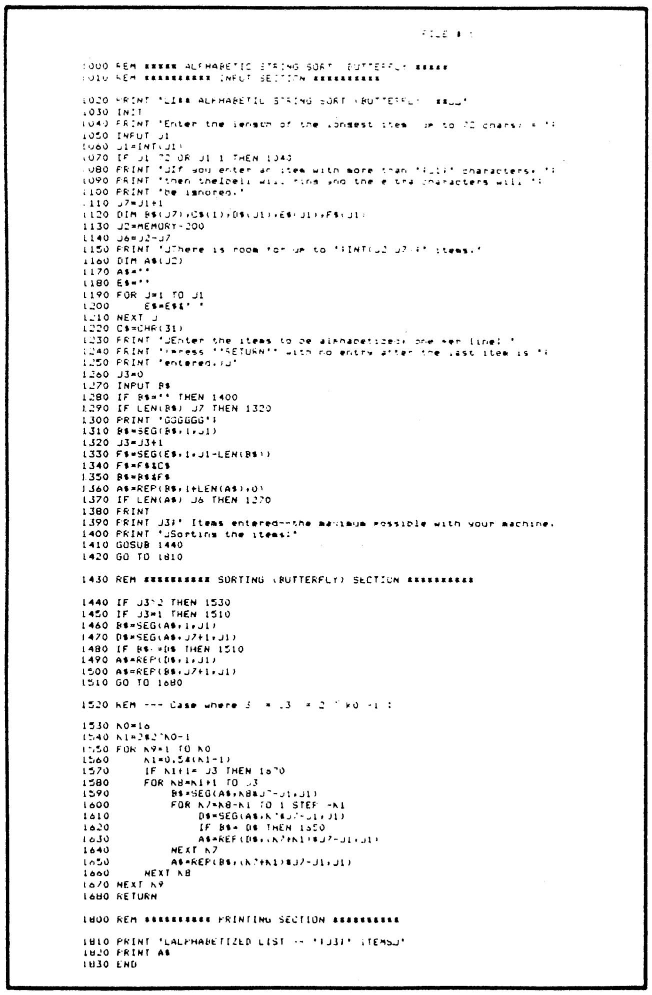
Programming Tips



Butterfly Sort Speeds Alpha String Sorting

by Gary Neher
Tektronix, Inc.
Beaverton, OR
and Dan Taylor
Tektronix, Inc.
Wilsonville, OR

The approximate time to perform a bubble sort is proportional to N^2 (N = number of items to be sorted). A butterfly sort takes roughly NlogN, a substantial time saving over the bubble sort. It also compares favorably with the "quicker" sort algorithm while using far less code.

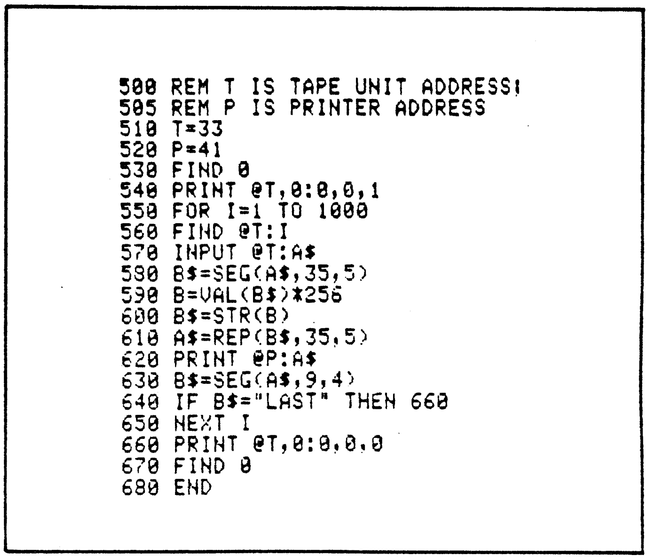


```
                                          FILE # :

1000 REM ***** ALPHABETIC STRING SORT (BUTTERFLY) *****
1010 REM ********** INPUT SECTION **********

1020 PRINT 'L*** ALPHABETIC STRING SORT (BUTTERFLY) ***JJ'
1030 INIT
1040 PRINT 'Enter the length of the longest item (up to 72 chars) = ';
1050 INPUT J1
1060 J1=INT(J1)
1070 IF J1>72 OR J1<1 THEN 1040
1080 PRINT 'JIf you enter an item with more than ';J1;' characters, ';
1090 PRINT 'then theIbell will ring and the extra characters will ';
1100 PRINT 'be ignored.'
1110 J7=J1+1
1120 DIM B$(J7),C$(1),D$(J1),E$(J1),F$(J1)
1130 J2=MEMORY-200
1140 J6=J2-J7
1150 PRINT 'JThere is room for up to ';INT(J2/J7);' items.'
1160 DIM A$(J2)
1170 A$=''
1180 E$=''
1190 FOR J=1 TO J1
1200      E$=E$&' '
1210 NEXT J
1220 C$=CHR(31)
1230 PRINT 'JEnter the items to be alphabetized, one per line; '
1240 PRINT 'press ''RETURN'' with no entry after the last item is ';
1250 PRINT 'entered.JJ'
1260 J3=0
1270 INPUT B$
1280 IF B$='' THEN 1400
1290 IF LEN(B$)<J7 THEN 1320
1300 PRINT 'GGGGGG';
1310 B$=SEG(B$,1,J1)
1320 J3=J3+1
1330 F$=SEG(E$,1,J1-LEN(B$))
1340 F$=F$&C$
1350 B$=B$&F$
1360 A$=REP(B$,1+LEN(A$),0)
1370 IF LEN(A$)<J6 THEN 1270
1380 PRINT
1390 PRINT J3;' Items entered--the maximum possible with your machine.
1400 PRINT 'JSorting the items:'
1410 GOSUB 1440
1420 GO TO 1810

1430 REM ********** SORTING (BUTTERFLY) SECTION **********

1440 IF J3>2 THEN 1530
1450 IF J3=1 THEN 1510
1460 B$=SEG(A$,1,J1)
1470 D$=SEG(A$,J7+1,J1)
1480 IF B$<=D$ THEN 1510
1490 A$=REP(D$,1,J1)
1500 A$=REP(B$,J7+1,J1)
1510 GO TO 1680

1520 REM --- Case where 3 <= J3 <= 2^K0-1 :

1530 K0=16
1540 K1=2*2^K0-1
1550 FOR K9=1 TO K0
1560      K1=0.5*(K1-1)
1570      IF K1+1>=J3 THEN 1670
1580      FOR K8=K1+1 TO J3
1590           B$=SEG(A$,K8*J7-J1,J1)
1600           FOR K7=K8-K1 TO 1 STEP -K1
1610                D$=SEG(A$,K7*J7-J1,J1)
1620                IF B$>=D$ THEN 1650
1630                A$=REP(D$,(K7+K1)*J7-J1,J1)
1640           NEXT K7
1650           A$=REP(B$,(K7+K1)*J7-J1,J1)
1660      NEXT K8
1670 NEXT K9
1680 RETURN

1800 REM ********** PRINTING SECTION **********

1810 PRINT 'LALPHABETIZED LIST -- ';J3;' ITEMSJ'
1820 PRINT A$
1830 END
```

TLIS To The 4642 Printer

by Jerry Anderson
Phillips Petroleum Co.
Bartlesville, OK

Although the 4642 Desktop Printer cannot process the 4050 Series TLIS command, executing the following sequence will produce the same result.



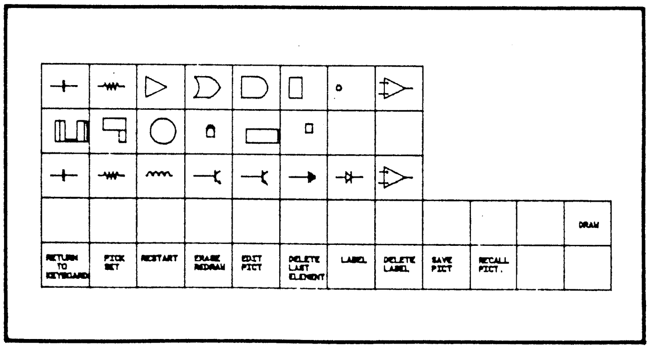
```
500 REM T IS TAPE UNIT ADDRESS!
505 REM P IS PRINTER ADDRESS
510 T=33
520 P=41
530 FIND 0
540 PRINT @T,0:0,0,1
550 FOR I=1 TO 1000
560 FIND @T:I
570 INPUT @T:A$
580 B$=SEG(A$,35,5)
590 B=VAL(B$)*256
600 B$=STR(B)
610 A$=REP(B$,35,5)
620 PRINT @P:A$
630 B$=SEG(A$,9,4)
640 IF B$="LAST" THEN 660
650 NEXT I
660 PRINT @T,0:0,0,0
670 FIND 0
680 END
```

Setting Up A Menu On The 4956 Tablet

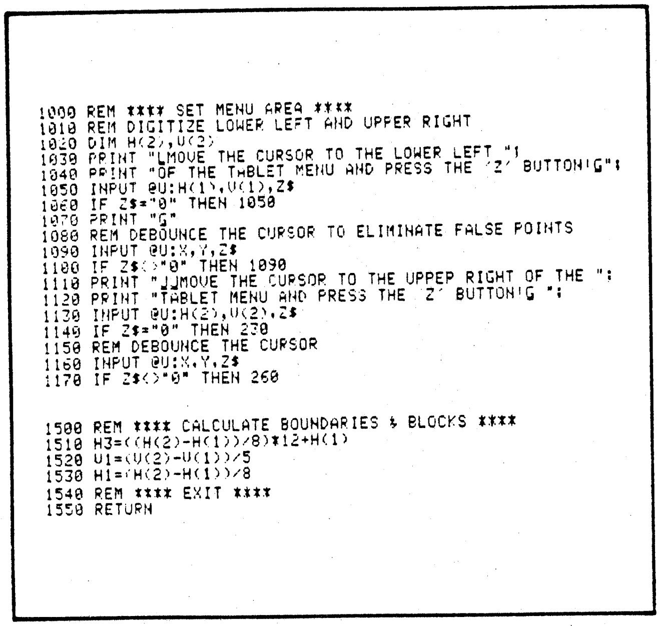
by Ed Mitchell
Tektronix, Inc.
Wilsonville, OR

When creating drawings, a menu on your 4956 Tablet allows you to select and move around predefined shapes anywhere on your plot. Commands may also be issued directly from the Tablet.

First, design your menu. The example in Figure 1 provides for a possible 24 commands and 24 objects. The layout is arbitrary; here the symbols are grouped in a 3-row 8-column block with the commands grouped in two rows of 12 columns. Keeping the blocks evenly spaced is essential so block detection can be made through a two-line algorithm.



Four parameters must be defined for the menu: the lower left and upper right menu locations, and the number of rows and columns. Since the menu location can be changed, its boundaries must be digitized. The following routine accomplishes this:



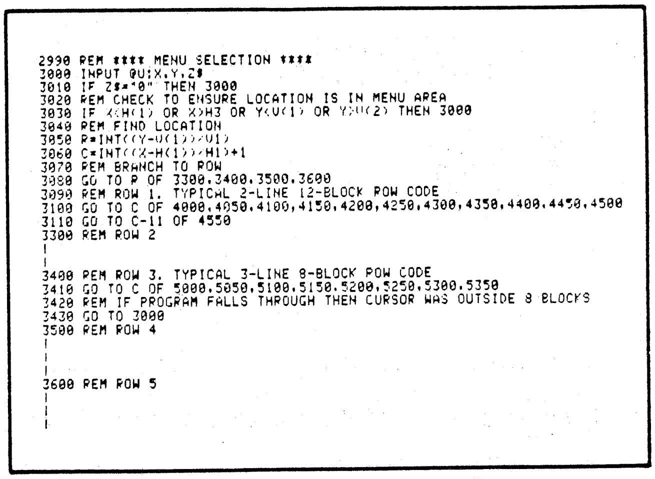
```
1000 REM **** SET MENU AREA ****
1010 REM DIGITIZE LOWER LEFT AND UPPER RIGHT
1020 DIM H(2),V(2)
1030 PRINT "LMOVE THE CURSOR TO THE LOWER LEFT ";
1040 PRINT "OF THE TABLET MENU AND PRESS THE 'Z' BUTTON!G";
1050 INPUT @U:H(1),V(1),Z$
1060 IF Z$="0" THEN 1050
1070 PRINT "G"
1080 REM DEBOUNCE THE CURSOR TO ELIMINATE FALSE POINTS
1090 INPUT @U:X,Y,Z$
1100 IF Z$<>"0" THEN 1090
1110 PRINT "JJMOVE THE CURSOR TO THE UPPEP RIGHT OF THE ":
1120 PRINT "TABLET MENU AND PRESS THE 'Z' BUTTON!G ";
1130 INPUT @U:H(2),V(2),Z$
1140 IF Z$="0" THEN 230
1150 REM DEBOUNCE THE CURSOR
1160 INPUT @U:X,Y,Z$
1170 IF Z$<>"0" THEN 260


1500 REM **** CALCULATE BOUNDARIES & BLOCKS ****
1510 H3=((H(2)-H(1))/8)*12+H(1)
1520 V1=(V(2)-V(1))/5
1530 H1=(H(2)-H(1))/8
1540 REM **** EXIT ****
1550 RETURN
```

When digitizing the boundaries, set your Tablet controller to STREAM SWITCH mode. This allows the 4050 System to continuously track the location of the cursor. After each menu coordinate is digitized, the routine ensures the pen or button has been pressed and released before continuing, a technique called "debouncing." (The mechanical bounce of the switch contacts may send out a pattern of unwanted "Z" commands which must be removed.) When working with the 4051 only one check of the Z parameter is required, but the 4051/4054 sometimes require three or four "dummy" inputs to ensure that all false points are cleared.

Once the lower left and upper right coordinates of the menu are known, simple computations decode the points. However, the other two menu parameters, rows and columns, must be specified in your code as constants. For this example, 8 and 12 columns are specified in statement 1510, 5 rows in statement 1520 and 8 columns in statement 1530.

Since the menu is not a rectangle and the 8-block upper right X coordinate is less than the 12-block X coordinate, statement 1510 computes the maximum horizontal coordinate. Statements 1520 and 1520 calculate the individual menu block sizes. (If you keep the menu in place on your Tablet while creating your drawing over a period of time, store the data derived from the above routine so you won't have to repeat these steps each time.)

Once you have entered menu mode*, the following routine interprets the coordinates digitized as a menu item and branches accordingly.



```
2990 REM **** MENU SELECTION ****
3000 INPUT @U:X,Y,Z$
3010 IF Z$="0" THEN 3000
3020 REM CHECK TO ENSURE LOCATION IS IN MENU AREA
3030 IF X<H(1) OR X>H3 OR Y<V(1) OR Y>V(2) THEN 3000
3040 REM FIND LOCATION
3050 R=INT((Y-V(1))/V1)
3060 C=INT((X-H(1))/H1)+1
3070 REM BRANCH TO ROW
3080 GO TO R OF 3300,3400,3500,3600
3090 REM ROW 1. TYPICAL 2-LINE 12-BLOCK POW CODE
3100 GO TO C OF 4000,4050,4100,4150,4200,4250,4300,4350,4400,4450,4500
3110 GO TO C-11 OF 4550
3300 REM ROW 2
|
|
3400 REM ROW 3. TYPICAL 3-LINE 8-BLOCK POW CODE
3410 GO TO C OF 5000,5050,5100,5150,5200,5250,5300,5350
3420 REM IF PROGRAM FALLS THROUGH THEN CURSOR WAS OUTSIDE 8 BLOCKS
3430 GO TO 3000
3500 REM ROW 4
|
|
|
3600 REM ROW 5
|
|
|
```

Statement 3000 tracks the cursor location. Statement 3010 checks to see if a switch was pressed; if so, statement 3030 verifies the point is inside the menu boundaries. Statements 3050 and 3060 then calculate the row and column.

The row "R" pointer range runs from 0 through 5 since any point calculated in statement 3050 will fall in one of the rows. However, the column "C" pointer is offset by one for a range of 1 through 12. This allows the program to "fall through" to the third statement of a row routine when the row is greater than 2 **and** the column is greater than 8. For example, should a coordinate defining Row 3 Column 9 be digitizied, the 4050 system won't branch at line 3410 but will continue to line 3430 which returns the user to the menu. You could include an error message here.

The foregoing routine could be expanded to include prompts, error messages and bells for confirmation of a selection. These have been left out so the structure may be clearly seen.

One last *important* detail. When placing the menu on the Tablet, *keep it straight*. A skewed menu can lead to incorrect operation.

***The routine to branch to menu mode has not been included in this programming tip.**

End of File Interrupt Logic

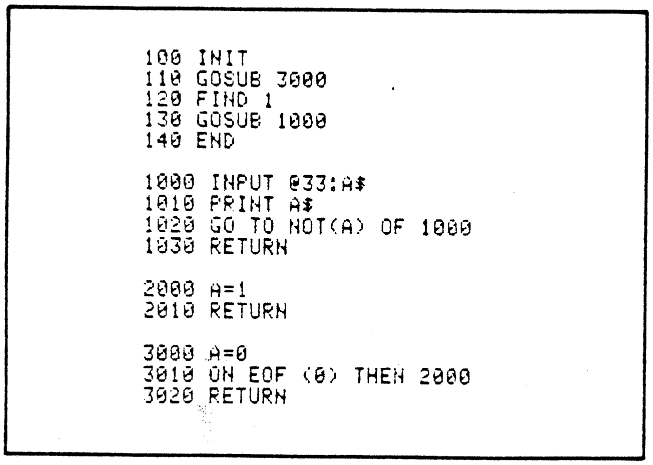
by John Carter
Tektronix, Inc.
Santa Clara Annex
and Pat Kelley
TEKniques staff

ON EOF (Ø)

The command ON EOF (Ø) ... enables the 4050 Graphic System to respond to an end of file condition during

INPUT or READ operations; the following discussion traces its response route.

When the end of file is reached, the system does an implied GOSUB from the executing statement to the line number specified when ON EOF was last **executed**. If the service routine ends with a RETURN command, control is returned to the statement following the READ or INPUT statement—**not** the statement following the ON EOF (Ø). Therefore, when the service routine terminates with a RETURN, the statement following the triggering action should contain a test to escape the INPUT or READ loop. The following code demonstrates the proper use of ON EOF (Ø) in this case.



```
100 INIT
110 GOSUB 3000
120 FIND 1
130 GOSUB 1000
140 END

1000 INPUT @33:A$
1010 PRINT A$
1020 GO TO NOT(A) OF 1000
1030 RETURN

2000 A=1
2010 RETURN

3000 A=0
3010 ON EOF (0) THEN 2000
3020 RETURN
```

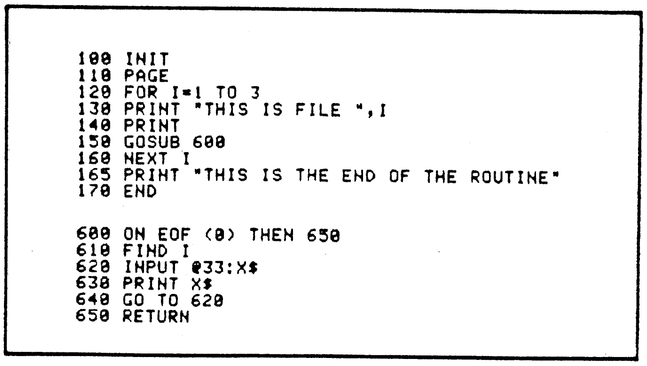
If you don't terminate the service routine with a RETURN, the system is no longer armed to respond to an EOF condition. For example, say you executed an ON EOF(Ø) statement, then read file 1. When you reached the end of the file, you branched to a service routine which printed your file; but the service routine did not end with a RETURN. Now, should you read another file, when you reach its end, the program will *not* branch to the service routine, but will stop and an EOF message will be printed on the graphic screen.

Also, if you don't end the service routine with a return the memory allocated to store the return address (the statement following the triggering action) would not be cleared. Furthermore, if the system continued execution and encountered a RETURN *not connected with another subroutine* it would return execution to the statement following the triggering action.

Therefore, treat your service routines as SUBROUTINES and RETURN from them unless you plan to end your program immediately afterward.

Check Your Logic

The logic which the code in Figure 2 represents has trapped more than one user into an infinite loop. In each case the user wanted to find a file, branch to a subroutine to read it, return to the main program for the next file, etc. However, the RETURN in statement 650 is associated with the implied EOF GOSUB rather than associated with the GOSUB in statement 150. Thus, when statement 620 is executed and an EOF is detected, the system branches (implied GOSUB) to statement 650, which returns execution to statement 630. Statement 630 "prints" the EOF, goes to statement 620, again detects the EOF, branches to statement 650 ..., ad infinitum.



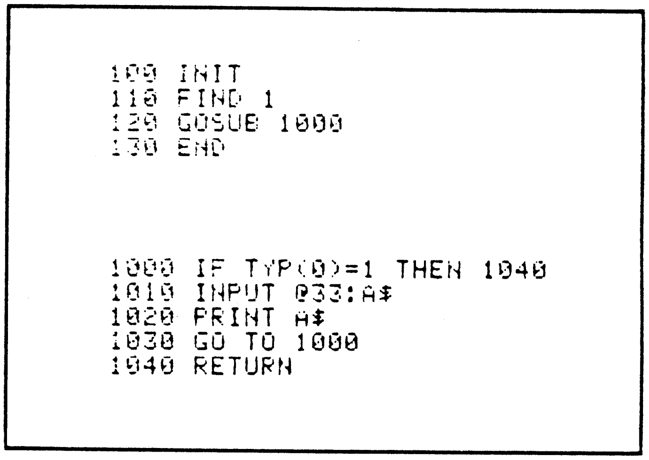
```
100 INIT
110 PAGE
120 FOR I=1 TO 3
130 PRINT "THIS IS FILE ",I
140 PRINT
150 GOSUB 600
160 NEXT I
165 PRINT "THIS IS THE END OF THE ROUTINE"
170 END

600 ON EOF (0) THEN 650
610 FIND I
620 INPUT @33:X$
630 PRINT X$
640 GO TO 620
650 RETURN
```

OFF EOF (Ø)

Once ON EOF (Ø) arms the system to respond to the end of file condition, you can return it to its default condition (i.e., treating the EOF as a fatal error), by issuing an OFF EOF(Ø). However, the OFF EOF(Ø) must be set after the RETURN command terminates the service routine. For example, inserting the statement: 1005 OFF EOF(Ø) into the code in Figure 1 would have no effect. The RETURN at statement 1010 would negate the OFF command.

An Alternative to ON EOF (Ø)

The TYP (Ø) command may also be used to detect end of file.* When you are READing or INPUTting a file, check the next type of data before each string or numeric input. If a 1 is returned in the variable, you have reached the end of file. The following lines of code demonstrate its use.



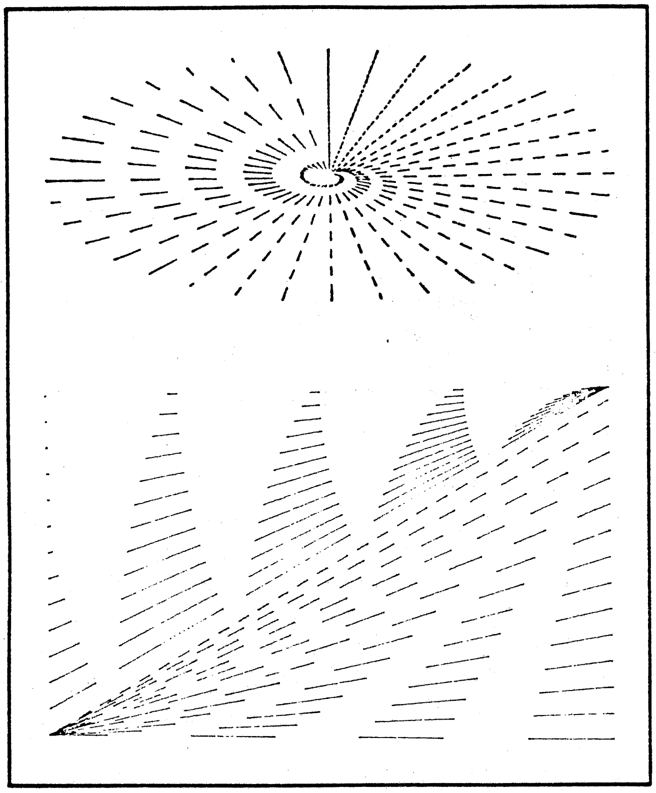
```
100 INIT
110 FIND 1
120 GOSUB 1000
130 END


1000 IF TYP(0)=1 THEN 1040
1010 INPUT @33:A$
1020 PRINT A$
1030 GO TO 1000
1040 RETURN
```

In most cases, the TYPe method of detecting the end of file may be more practical than the ON EOF (Ø) method.

***The TYP (Ø) command will also detect a NEW or LAST file, and whether the next item is ASCII, binary numeric or binary string. Refer to the Graphic System Reference Manual for suggestions on using TYP (Ø) in these cases.**

Dashed Line Subroutines

by Bob Wainwright
Tektronix, U.K.
Manchester, England

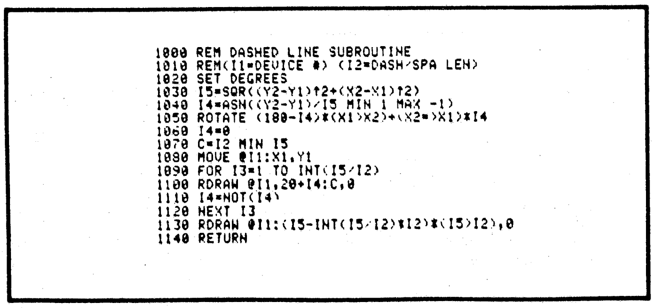


Two subroutines draw dashed lines between any two points. In both routines you specify the points, the dash length, and the output device. The space between dashes on the first routine is the same as the dash length; however, you may specify the length of this space in the second routine.

Before making a call to the first subroutine, you'll need to define six variables. These are:

X1, Y1 . The x and y starting coordinates
X2, Y2 . The x and y ending coordinates
I1 . . . . The output device (32 = 4050 screen)
I2 . . . . The dash length (current WINDOW units)*

Once these variables have been assigned in your program, a call to the dashed line subroutine will result in the desired dashed line being plotted on the specified device.

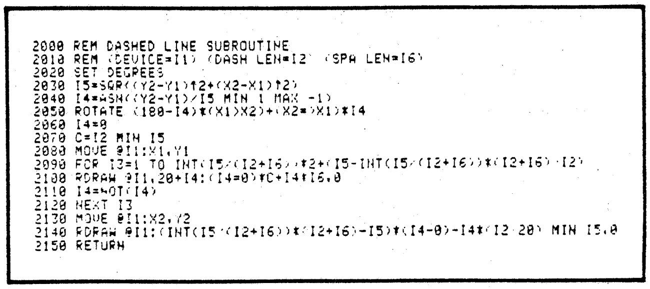


```
1000 REM DASHED LINE SUBROUTINE
1010 REM(I1=DEVICE #) (I2=DASH/SPA LEN)
1020 SET DEGREES
1030 I5=SQR((Y2-Y1)↑2+(X2-X1)↑2)
1040 I4=ASN((Y2-Y1)/I5 MIN 1 MAX -1)
1050 ROTATE (180-I4)*(X1>X2)+(X2=>X1)*I4
1060 I4=0
1070 C=I2 MIN I5
1080 MOVE @I1:X1,Y1
1090 FOR I3=1 TO INT(I5/I2)
1100 RDRAW @I1,20+I4:C,0
1110 I4=NOT(I4)
1120 NEXT I3
1130 RDRAW @I1:(I5-INT(I5/I2)*I2)*(I5>I2),0
1140 RETURN
```

Before making a call to the second subroutine, you'll need to define the previous six variables—X1,Y1,X2,Y2,I1,I2, plus:

I6 . . . . The space between dashes (current WINDOW units)*

Once these variables have been assigned, a call to the following subroutine will result in the desired dashed line and space being plotted on your device.



```
2000 REM DASHED LINE SUBROUTINE
2010 REM (DEVICE=I1) (DASH LEN=I2) (SPA LEN=I6)
2020 SET DEGREES
2030 I5=SQR((Y2-Y1)↑2+(X2-X1)↑2)
2040 I4=ASN((Y2-Y1)/I5 MIN 1 MAX -1)
2050 ROTATE (180-I4)*(X1>X2)+(X2=>X1)*I4
2060 I4=0
2070 C=I2 MIN I5
2080 MOVE @I1:X1,Y1
2090 FOR I3=1 TO INT(I5/(I2+I6))*2+(I5-INT(I5/(I2+I6))*(I2+I6)>I2)
2100 RDRAW @I1,20+I4:(I4=0)*C+I4*I6,0
2110 I4=NOT(I4)
2120 NEXT I3
2130 MOVE @I1:X2,Y2
2140 RDRAW @I1:(INT(I5/(I2+I6))*(I2+I6)-I5)*(I4-0)-I4*(I2 20) MIN I5,0
2150 RETURN
```

In addition to the variables mentioned above, three scratch variables, I3, I4 and I5 are used in both routines.

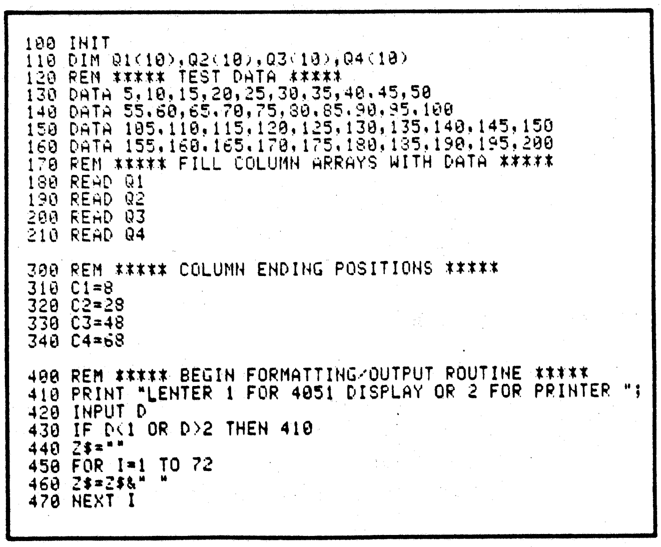
***If the window is changed from its default ratio of 1.3:1.0, i.e., WIN 0,130,-100,200, and you are using multiple dashed lines, the dash length (and space) will vary for each line according to the slope of the line determined by the beginning and ending coordinates.**

Ed. Note: A general purpose dashed line algorithm is included in the 4050 Series Application Library under abstract 51/00-9508/1.

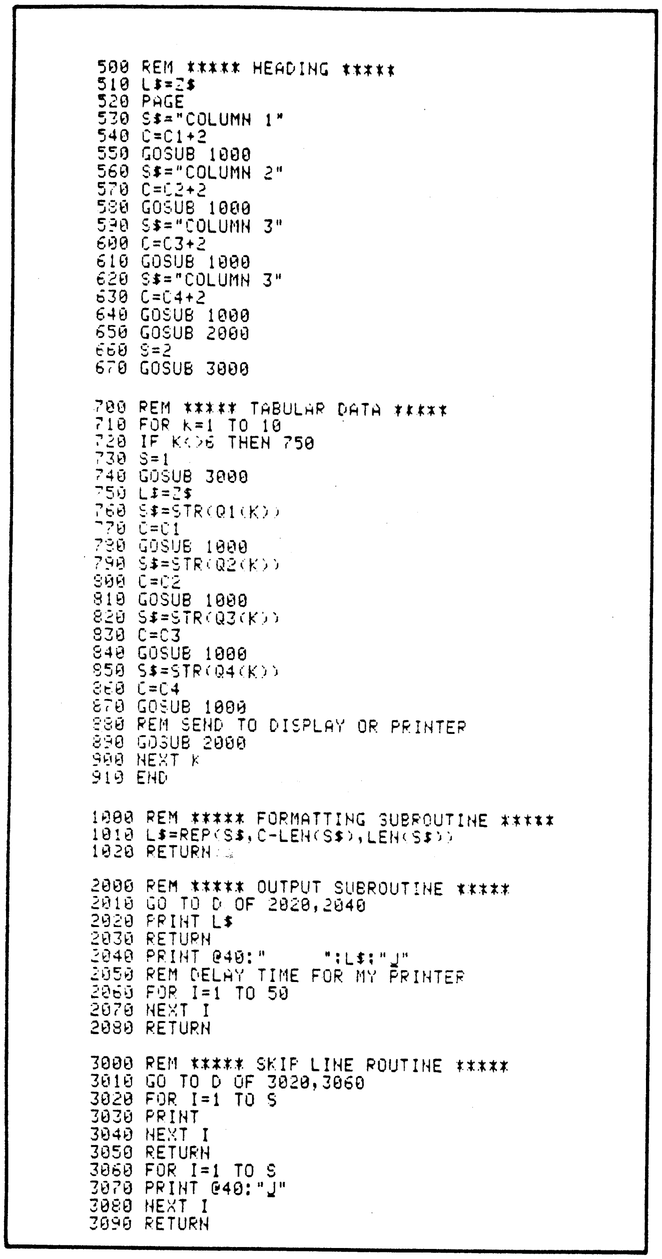
Formatting Tabular Output Using String Functions

by Bob Pilkington
AT&T Long Lines
Bedminster, NJ

Do you output tabular reports from your 4051 to a printer or terminal that doesn't contain or recognize tabbing? If so, the following routine which takes advantage of the 4051 string functions may prove useful.



```
100 INIT
110 DIM Q1(10),Q2(10),Q3(10),Q4(10)
120 REM ***** TEST DATA *****
130 DATA 5,10,15,20,25,30,35,40,45,50
140 DATA 55,60,65,70,75,80,85,90,95,100
150 DATA 105,110,115,120,125,130,135,140,145,150
160 DATA 155,160,165,170,175,180,135,190,195,200
170 REM ***** FILL COLUMN ARRAYS WITH DATA *****
180 READ Q1
190 READ Q2
200 READ Q3
210 READ Q4

300 REM ***** COLUMN ENDING POSITIONS *****
310 C1=8
320 C2=28
330 C3=48
340 C4=68

400 REM ***** BEGIN FORMATTING/OUTPUT ROUTINE *****
410 PRINT "LENTER 1 FOR 4051 DISPLAY OR 2 FOR PRINTER ";
420 INPUT D
430 IF D<1 OR D>2 THEN 410
440 Z$=""
450 FOR I=1 TO 72
460 Z$=Z$&" "
470 NEXT I
```



```
500 REM ***** HEADING *****
510 L$=Z$
520 PAGE
530 S$="COLUMN 1"
540 C=C1+2
550 GOSUB 1000
560 S$="COLUMN 2"
570 C=C2+2
580 GOSUB 1000
590 S$="COLUMN 3"
600 C=C3+2
610 GOSUB 1000
620 S$="COLUMN 3"
630 C=C4+2
640 GOSUB 1000
650 GOSUB 2000
660 S=2
670 GOSUB 3000

700 REM ***** TABULAR DATA *****
710 FOR K=1 TO 10
720 IF K<>6 THEN 750
730 S=1
740 GOSUB 3000
750 L$=Z$
760 S$=STR(Q1(K))
770 C=C1
780 GOSUB 1000
790 S$=STR(Q2(K))
800 C=C2
810 GOSUB 1000
820 S$=STR(Q3(K))
830 C=C3
840 GOSUB 1000
850 S$=STR(Q4(K))
860 C=C4
870 GOSUB 1000
880 REM SEND TO DISPLAY OR PRINTER
890 GOSUB 2000
900 NEXT K
910 END

1000 REM ***** FORMATTING SUBROUTINE *****
1010 L$=REP(S$,C-LEN(S$),LEN(S$))
1020 RETURN

2000 REM ***** OUTPUT SUBROUTINE *****
2010 GO TO D OF 2020,2040
2020 PRINT L$
2030 RETURN
2040 PRINT @40:"      ";L$;"J"
2050 REM DELAY TIME FOR MY PRINTER
2060 FOR I=1 TO 50
2070 NEXT I
2080 RETURN

3000 REM ***** SKIP LINE ROUTINE *****
3010 GO TO D OF 3020,3060
3020 FOR I=1 TO S
3030 PRINT
3040 NEXT I
3050 RETURN
3060 FOR I=1 TO S
3070 PRINT @40:"J"
3080 NEXT I
3090 RETURN
```

The output is formatted according to the desired column ending positions, which are defined in statements 310—340. Statements 440—470 initialize the string variable in which each output line (row) is formatted. The formatting subroutine, statement 1010, replaces the appropriate blanks in the string variable with, first, the labeling from statements 500—640, then with one set of data from each array. Thus, the latter case for our example, requires four calls to the subroutine for formatting a row.

The string (row) is then output to the 4051 graphic display or to the printer; we output through address 40. The routine is continued until the last data from each array is formatted and printed; in this example a total of 10 rows are output (statement 710). Statements 620—640 provide for a blank separating line after the fifth output row. The key variables are:

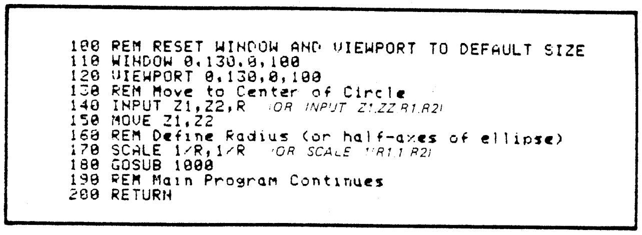
C	= title ending positions
C1,C2,C3,C4	= column ending positions
K	= number of rows
L$	= line buffer string variable —72 characters*
Q1,Q2,Q3,Q4	= arrays of test data
S	= number of lines to be skipped
S$	= argument used to replace L$ blanks with characters

***Z$ and L$ could be dimensioned to hold longer strings for output to a 132-column line printer.**

Fine Circles or Ellipses

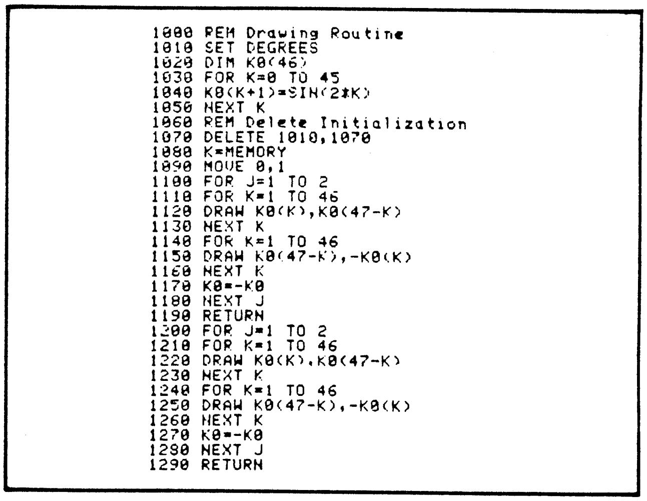
by S. Schicktanz
Physics Department
Technical University of Munich
Munich, Germany

The following routine draws very fine circles and may easily be modified for ellipses. The routine may be called as follows:



```
100 REM RESET WINDOW AND VIEWPORT TO DEFAULT SIZE
110 WINDOW 0,130,0,100
120 VIEWPORT 0,130,0,100
130 REM Move to Center of Circle
140 INPUT Z1,Z2,R   (OR INPUT Z1,Z2,R1,R2)
150 MOVE Z1,Z2
160 REM Define Radius (or half-axes of ellipse)
170 SCALE 1/R,1/R   (OR SCALE 1/R1,1/R2)
180 GOSUB 1000
190 REM Main Program Continues
200 RETURN
```

Once the array is developed in lines 1000 to 1050, its initialization procedure is deleted. Now as many circles (or ellipses) may be drawn as desired specifying different radii or axes.



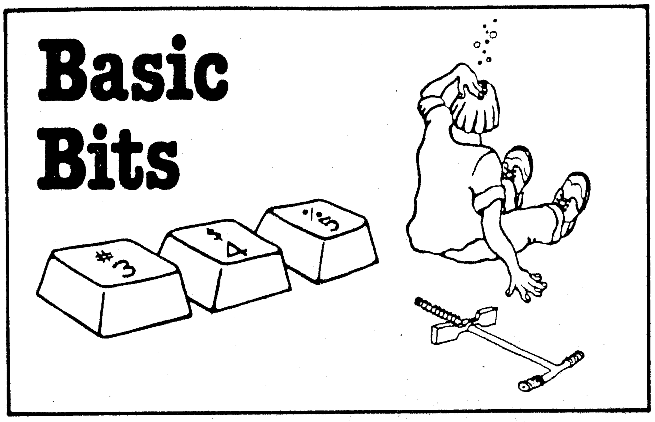
```
1000 REM Drawing Routine
1010 SET DEGREES
1020 DIM K0(46)
1030 FOR K=0 TO 45
1040 K0(K+1)=SIN(2*K)
1050 NEXT K
1060 REM Delete Initialization
1070 DELETE 1010,1070
1080 K=MEMORY
1090 MOVE 0,1
1100 FOR J=1 TO 2
1110 FOR K=1 TO 46
1120 DRAW K0(K),K0(47-K)
1130 NEXT K
1140 FOR K=1 TO 46
1150 DRAW K0(47-K),-K0(K)
1160 NEXT K
1170 K0=-K0
1180 NEXT J
1190 RETURN
1200 FOR J=1 TO 2
1210 FOR K=1 TO 46
1220 DRAW K0(K),K0(47-K)
1230 NEXT K
1240 FOR K=1 TO 46
1250 DRAW K0(47-K),-K0(K)
1260 NEXT K
1270 K0=-K0
1280 NEXT J
1290 RETURN
```

Correction To Programming Tip

In TEKniques Vol. 3 No. 6 a negative sign and an arithmetic operator were inadvertently ommitted in a programming tip. On page 15 the formula in "One Reason for a Tape File Directory(ies)" should read:

```
MARK 1, -INT(-4000/256)*256 MAX 768
```

File Size = -INT(-X/256)*256 MAX 768



Remove 4907 ROM Pack If Not Using

by Ed Mitchell
Tektronix, Inc.
Wilsonville, OR

Remove the 4907 ROM Pack when you're not using your 4907 Disc unit. If you don't it could cause your 4050 system to "hang busy."

The 4050 System automatically goes through an initialization when turned on, when a program is OLDed, or, of course, when an INIT command is issued from the keyboard or under program control. If the 4907 ROM Pack is in place, the 4050 System expects device 0 to be on-line and, as part of the initialization, it will poll device 0. If the disc unit is not connected to the 4050 System or is not turned on, the poll will cause the 4050 System to wait for a response, which will never be forthcoming.

4050 Series 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 reponsibility 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	*$20 per program*
Recording Fee	*5 per program*
Tape Cartridge	*30 per tape*

Contribute

Contribute one program to the Library and receive three in exchange. Send in the membership card from your 4050 Series Graphic System Reference Manual to get the details. Or call us (503) 682-3411, ext. 3618.

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: 51/00-8030/0

Title: **GPIB GET Command Trigger**
Author: James R. Matey
RCA Laboratories
Princeton, NJ
Memory Requirement: 32K
Peripherals: One—HP 3438A or
Two—ICS 4880's
Statements: 157
Files: 2 ASCII Programs

Two programs which trigger devices on the GPIB of the 4051 Graphic System by means of the group execute trigger command (GET).

The first program is used with an HP 3438A, the second with two ICS 4880 Bus Couplers. The device(s) must be connected via the appropriate cables to the 4051, and in the case of the ICS 4880's must also be connected properly to an appropriate BCD output device. The details of the connections are presented in the manufacturers' instructions for these devices.

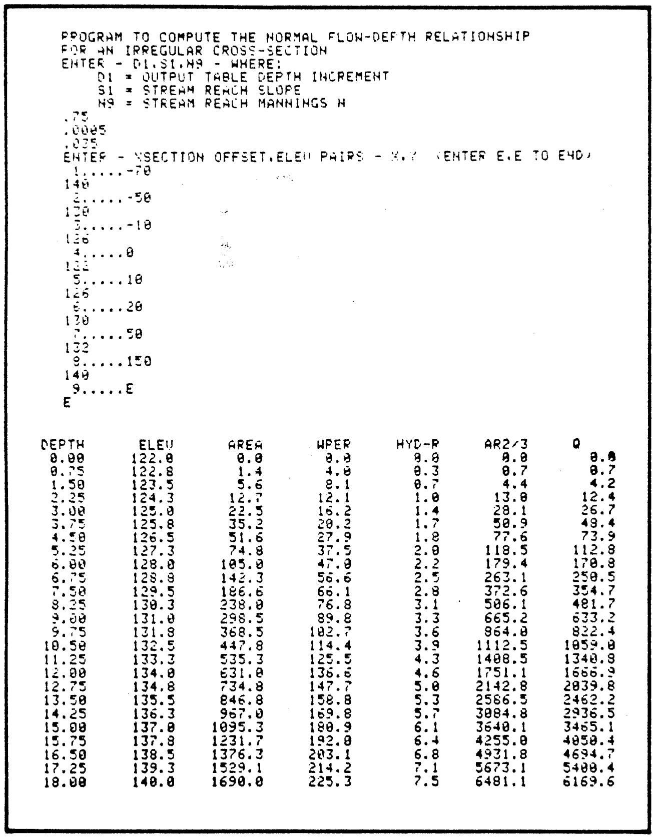
These programs are used to automate laboratory experiments allowing the 4051 to read the results of a measurement.

ABSTRACT NUMBER: 51/07-1202/0

Title: **Manning's Equation Depth Flow**
Author: Richard L. Laramie
CDM/Resource Analysis
Waltham, MA
Memory Requirement: 8K
Peripherals: 4907 File Manager
Optional-4631 Hard Copy Unit
Statements: 88

The program uses Manning's equation to calculate a depth-flow relationship for a given stream reach. Required input includes slope, Manning's "n", and offset-elevation pairs. Output is a table of water surface elevation area, wetted perimeter, hydraulic radius, $AR^{2/3}$, and flow for each depth.

The program is useful in establishing a depth-flow relationship for an open channel.



```
PROGRAM TO COMPUTE THE NORMAL FLOW-DEPTH RELATIONSHIP
FOR AN IRREGULAR CROSS-SECTION
ENTER - D1,S1,N9 - WHERE:
    D1 = OUTPUT TABLE DEPTH INCREMENT
    S1 = STREAM REACH SLOPE
    N9 = STREAM REACH MANNINGS N
.75
.0005
.035
ENTER - XSECTION OFFSET,ELEV PAIRS - X,Y  (ENTER E,E TO END)
 1.....-70
140
 2.....-50
120
 3.....-10
126
 4.....0
122
 5.....10
126
 6.....20
130
 7.....50
132
 8.....150
140
 9.....E
E

DEPTH       ELEV       AREA      WPER      HYD-R      AR2/3        Q
 0.00      122.0        0.0       0.0        0.0        0.0       0.0
 0.75      122.8        1.4       4.0        0.3        0.7       0.7
 1.50      123.5        5.6       8.1        0.7        4.4       4.2
 2.25      124.3       12.7      12.1        1.0       13.0      12.4
 3.00      125.0       22.5      16.2        1.4       28.1      26.7
 3.75      125.8       35.2      20.2        1.7       50.9      49.4
 4.50      126.5       51.6      27.9        1.8       77.6      73.9
 5.25      127.3       74.8      37.5        2.0      118.5     112.8
 6.00      128.0      105.0      47.0        2.2      179.4     170.8
 6.75      128.8      142.3      56.6        2.5      263.1     250.5
 7.50      129.5      186.6      66.1        2.8      372.6     354.7
 8.25      130.3      238.0      76.8        3.1      506.1     481.7
 9.00      131.0      298.5      89.8        3.3      665.2     633.2
 9.75      131.8      368.5     102.7        3.6      864.0     822.4
10.50      132.5      447.8     114.4        3.9     1112.5    1059.0
11.25      133.3      535.3     125.5        4.3     1408.5    1340.8
12.00      134.0      631.0     136.6        4.6     1751.1    1666.9
12.75      134.8      734.8     147.7        5.0     2142.8    2039.8
13.50      135.5      846.8     158.8        5.3     2586.5    2462.2
14.25      136.3      967.0     169.8        5.7     3084.8    2936.5
15.00      137.0     1095.3     180.9        6.1     3640.1    3465.1
15.75      137.8     1231.7     192.0        6.4     4255.0    4050.4
16.50      138.5     1376.3     203.1        6.8     4931.8    4694.7
17.25      139.3     1529.1     214.2        7.1     5673.1    5400.4
18.00      140.0     1690.0     225.3        7.5     6481.1    6169.6
```

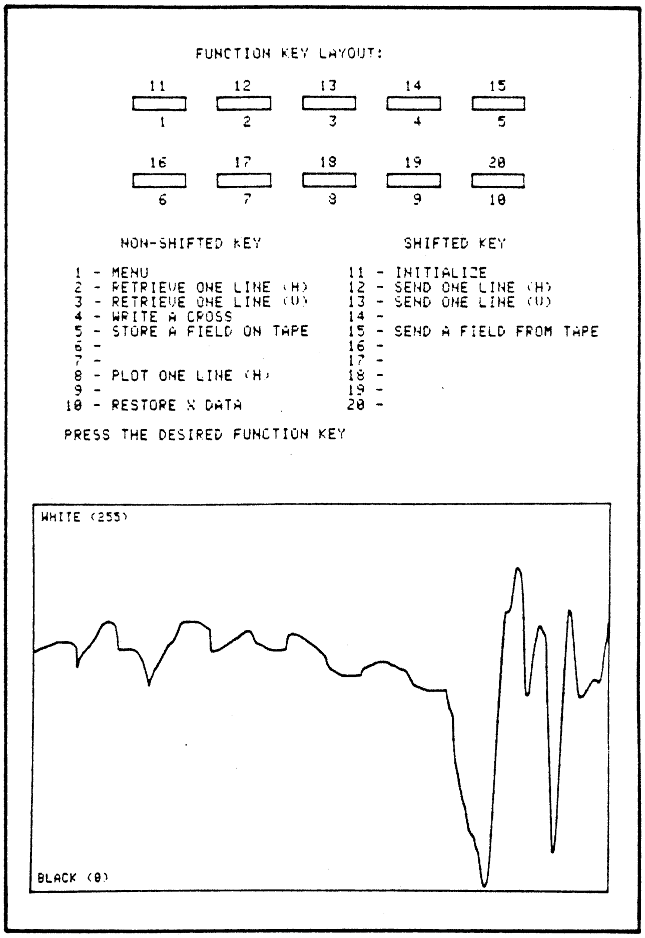
ABSTRACT NUMBER: 51/00-6111/0

Title: **QUANTEX DS-12 Interface**
Author: John Carter
Tektronix, Inc.
Santa Clara Field Office
Memory Requirement: 32K
Peripherals: QUANTEX DS-12 Image Processor
Statements: 266
Files: 1 ASCII Program

The program provides an interface between the 4050 Series Graphic System and a QUANTEX DS-12 Image Processor.

The User-Definable Keys are used to perform the following functions:

- — Retrieve one horizontal line
- — Retrieve one vertical line
- — Send one horizontal line
- — Send one vertical line
- — Plot one horizontal line
- — Write a cross
- — Store a field on tape
- — Send a field from tape
- — Restore X data

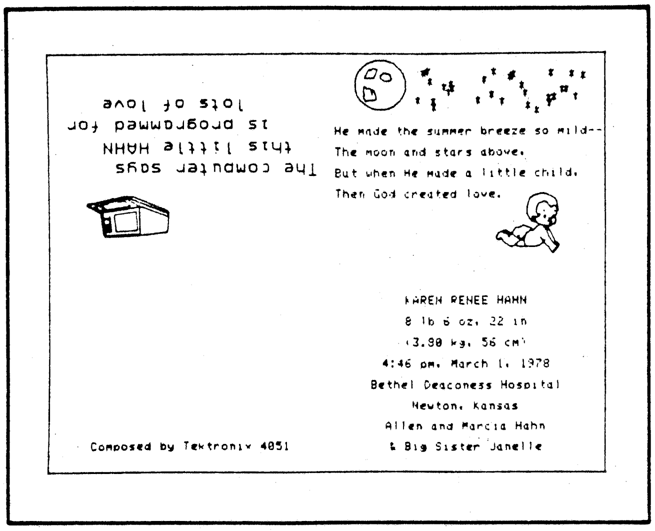


```
                 FUNCTION KEY LAYOUT:

      11        12        13        14        15
     [  ]      [  ]      [  ]      [  ]      [  ]
      1         2         3         4         5

      16        17        18        19        20
     [  ]      [  ]      [  ]      [  ]      [  ]
      6         7         8         9         10

        NON-SHIFTED KEY                SHIFTED KEY

 1 - MENU                        11 - INITIALIZE
 2 - RETRIEVE ONE LINE (H)       12 - SEND ONE LINE (H)
 3 - RETRIEVE ONE LINE (V)       13 - SEND ONE LINE (V)
 4 - WRITE A CROSS               14 -
 5 - STORE A FIELD ON TAPE       15 - SEND A FIELD FROM TAPE
 6 -                             16 -
 7 -                             17 -
 8 - PLOT ONE LINE (H)           18 -
 9 -                             19 -
10 - RESTORE X DATA              20 -

PRESS THE DESIRED FUNCTION KEY
```

ABSTRACT NUMBER: 51/00-6006/0

Title: **Baby Announcement Card**
Author: Allen G. Hahn
Hesston Corporation
Hesston, KS
Memory Requirement: 24K
Peripherals: 4631 Hard Copy Unit
Statements: 371
Files: 1 ASCII Program

The program draws a Baby Announcement Card on the 4050 Series Graphic System screen. The program is selfcontained and produces a card, when folded, that is 3 ⅛" X 4 ¼", by folding the paper into fourths. Required input is the baby's name and birth statistics.



The computer says
this little HAHN
is programmed for
lots of love

He made the summer breeze so mild--
The moon and stars above,
But when He made a little child,
Then God created love.

KAREN RENEE HAHN
8 lb 6 oz, 22 in
(3.80 kg, 56 cm)
4:46 pm, March 1, 1978
Bethel Deaconess Hospital
Newton, Kansas
Allen and Marcia Hahn
& Big Sister Janelle

Composed by Tektronix 4051

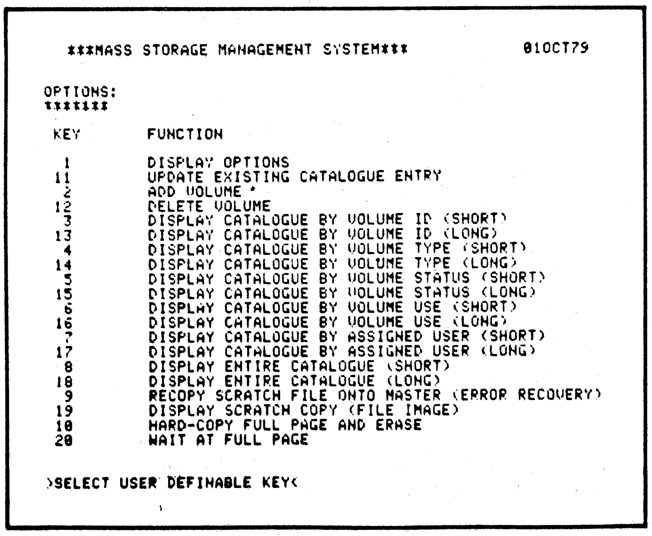
ABSTRACT NUMBER: 51/00-6112/0

Title: **Mass Storage Management System**
Author: Captain S.K. Sanford
Aberdeen Proving Ground, MD
Memory Requirement: 32K
Peripherals: 4924 Digital Cartridge Tape Drive
Optional-4631 Hard Copy Unit
Statements: 680
Files: 1 ASCII Program

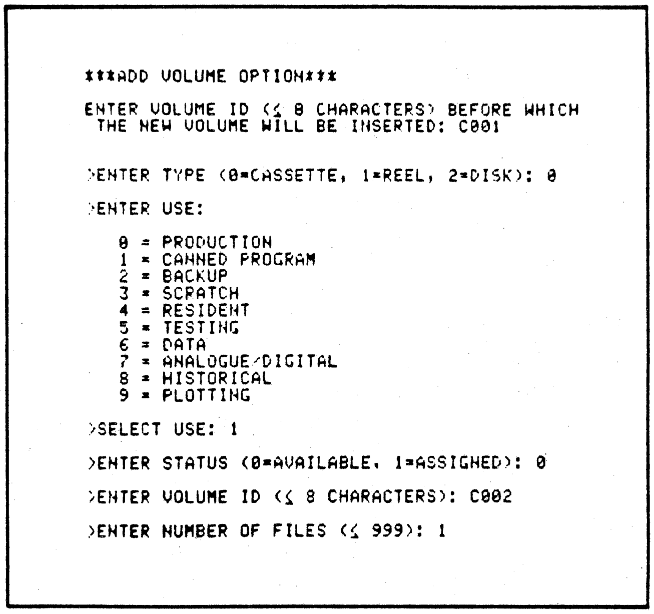
The program is designed to monitor the use of mass storage media, specifically cassettes, discs and reel tapes.

Reports may be generated interactively based on volume ID, media type, availability status, applications, or assigned user. The date of assignment and number of assignments is also recorded.

The program is tutorial.



```
   ***MASS STORAGE MANAGEMENT SYSTEM***                   01OCT79

 OPTIONS:
 *******

  KEY          FUNCTION

   1           DISPLAY OPTIONS
  11           UPDATE EXISTING CATALOGUE ENTRY
   2           ADD VOLUME *
  12           DELETE VOLUME
   3           DISPLAY CATALOGUE BY VOLUME ID (SHORT)
  13           DISPLAY CATALOGUE BY VOLUME ID (LONG)
   4           DISPLAY CATALOGUE BY VOLUME TYPE (SHORT)
  14           DISPLAY CATALOGUE BY VOLUME TYPE (LONG)
   5           DISPLAY CATALOGUE BY VOLUME STATUS (SHORT)
  15           DISPLAY CATALOGUE BY VOLUME STATUS (LONG)
   6           DISPLAY CATALOGUE BY VOLUME USE (SHORT)
  16           DISPLAY CATALOGUE BY VOLUME USE (LONG)
   7           DISPLAY CATALOGUE BY ASSIGNED USER (SHORT)
  17           DISPLAY CATALOGUE BY ASSIGNED USER (LONG)
   8           DISPLAY ENTIRE CATALOGUE (SHORT)
  18           DISPLAY ENTIRE CATALOGUE (LONG)
   9           RECOPY SCRATCH FILE ONTO MASTER (ERROR RECOVERY)
  19           DISPLAY SCRATCH COPY (FILE IMAGE)
  10           HARD-COPY FULL PAGE AND ERASE
  20           WAIT AT FULL PAGE

 >SELECT USER DEFINABLE KEY<
```



```
 ***ADD VOLUME OPTION***

 ENTER VOLUME ID (≤ 8 CHARACTERS) BEFORE WHICH
  THE NEW VOLUME WILL BE INSERTED: C001

 >ENTER TYPE (0=CASSETTE, 1=REEL, 2=DISK): 0

 >ENTER USE:

    0 = PRODUCTION
    1 = CANNED PROGRAM
    2 = BACKUP
    3 = SCRATCH
    4 = RESIDENT
    5 = TESTING
    6 = DATA
    7 = ANALOGUE/DIGITAL
    8 = HISTORICAL
    9 = PLOTTING

 >SELECT USE: 1

 >ENTER STATUS (0=AVAILABLE, 1=ASSIGNED): 0

 >ENTER VOLUME ID (≤ 8 CHARACTERS): C002

 >ENTER NUMBER OF FILES (≤ 999): 1
```

ABSTRACT NUMBER: 51/07-8031/0

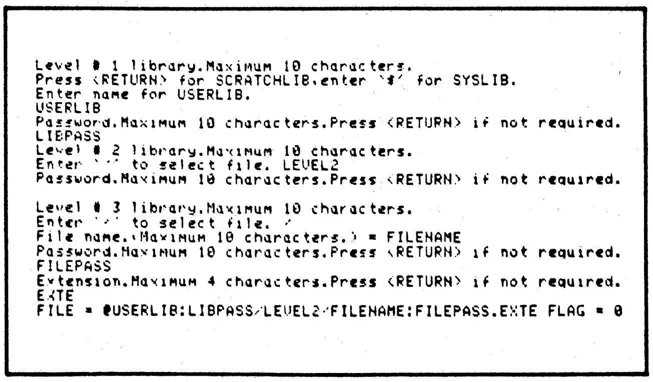
Title: **File Identifier**
Author: Nick Ogbourne
Comalco Aluminium Ltd.
Bell Bay
George Town, Tasmania
Memory Requirement: 8K
Peripherals: 4907 File Manager
Statements: 112
Files: 1 ASCII Program

The program is a subroutine that compiles a file identifier which will comply with the 4907 File Manager rules.

The program prompts the user to select libraries to the selected level, up to level 4, including SYSLIB or SCRATCHLIB. Passwords for any or all libraries may be added.

Following library selection, file selection on the same basis occurs, plus the selection of a file extension.

The valid file name is then returned in E$ and a flat, E0, assumes a value of 0 if the file does not currently exist and 1 if it does currently exist.



```
Level # 1 library.Maximum 10 characters.
Press <RETURN> for SCRATCHLIB,enter `#' for SYSLIB.
Enter name for USERLIB.
USERLIB
Password.Maximum 10 characters.Press <RETURN> if not required.
LIBPASS
Level # 2 library.Maximum 10 characters.
Enter `/' to select file. LEVEL2
Password.Maximum 10 characters.Press <RETURN> if not required.

Level # 3 library.Maximum 10 characters.
Enter `/' to select file. /
File name.(Maximum 10 characters.) = FILENAME
Password.Maximum 10 characters.Press <RETURN> if not required.
FILEPASS
Extension.Maximum 4 characters.Press <RETURN> if not required.
EXTE
FILE = @USERLIB:LIBPASS/LEVEL2/FILENAME:FILEPASS.EXTE FLAG = 0
```

ABSTRACT NUMBER: 51/00-6501/0

Title: **Recreational Plots #1**
Memory Requirement: 8K—24K
Peripherals: None
Files: 24 ASCII Program
1 Binary Data

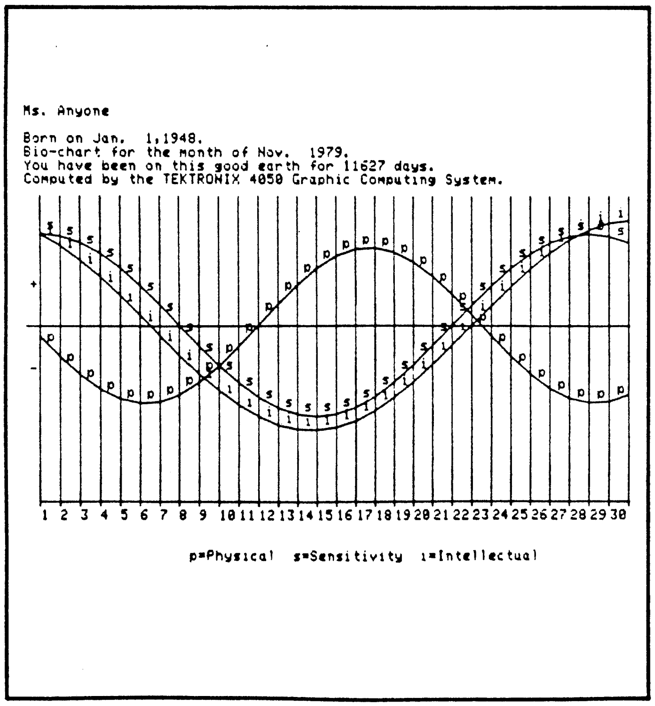
*COST: $99 on tape only

Twenty-three 4050 Series Graphic plots help the newcomer to computers interact with the friendly 4050 Series Systems. But the diversions are geared for the most experienced computer operator as well. Some of the programs test your logic or math skills, others your dexterity on the User-Definable Keys; some simply display the fine resolution of your graphic screen.

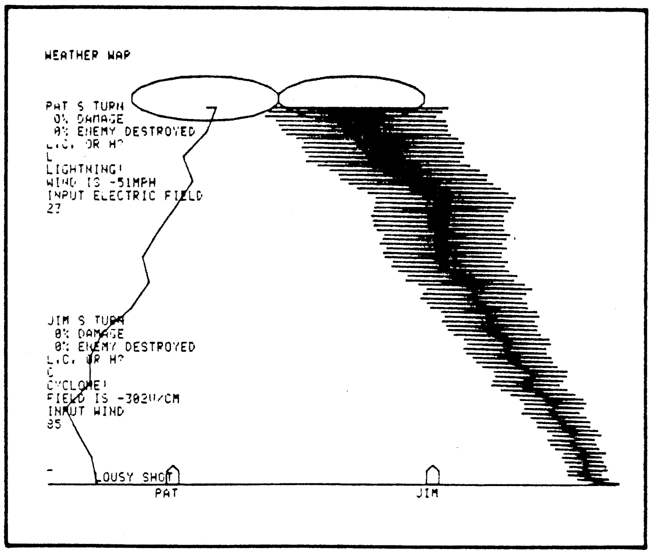
Included in this first Recreational Plot tape are:

- Shoot
- Lunar Lander
- Qubic
- Weather War
- Othello
- Golf Game #1
- Golf Game #2
- Wumpus
- Hangman
- Acey Ducey
- Ping Pong
- Tic Tac Toe for 2
- Computer Tic Tac Toe
- I.Q.
- Pinball
- Mugwump
- Blackjack
- Hamurabi
- Biorhythm
- Land Mines
- Computer Art
- Polygons
- Mastermind

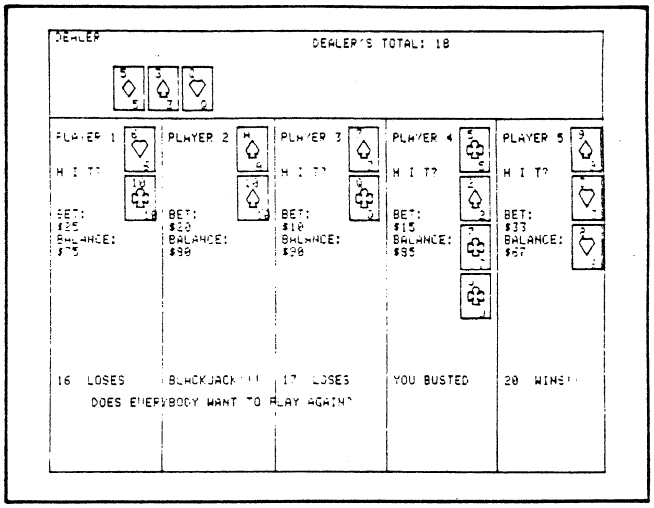
*The cost includes the documentation, recording fee and tape cartridge. However, to keep the cost down, no listings are included with the documentation; the user may easily run off his own, however. No other programs may be included on the tape.



```
Ms. Anyone

Born on Jan.  1,1948.
Bio-chart for the month of Nov.  1979.
You have been on this good earth for 11627 days.
Computed by the TEKTRONIX 4050 Graphic Computing System.
```



```
WEATHER WAR

PAT S TURN
 0% DAMAGE
 0% ENEMY DESTROYED
L.C. OR H?
L
LIGHTNING!
WIND IS -51MPH
INPUT ELECTRIC FIELD
27

JIM S TURN
 0% DAMAGE
 0% ENEMY DESTROYED
L.C. OR H?
C
CYCLONE!
FIELD IS -302V/CM
INPUT WIND
85

-  LOUSY SHOT
```



```
DEALER                         DEALER'S TOTAL: 18

PLAYER 1     PLAYER 2     PLAYER 3     PLAYER 4     PLAYER 5

H I T?                    H I T?       H I T?       H I T?

BET:         BET:         BET:         BET:         BET:
$25          $20          $10          $15          $33
BALANCE:     BALANCE:     BALANCE:     BALANCE:     BALANCE:
$75          $99          $90          $85          $67

16  LOSES    BLACKJACK!!! 17  LOSES    YOU BUSTED   20  WINS!!
    DOES EVERYBODY WANT TO PLAY AGAIN?
```

Tektronix®
COMMITTED TO EXCELLENCE

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

STELLA MCMORRAN
~~40-307~~
63-337
IF YOU HAVE MOVED
PLEASE CALL 3618

ADDRESS CORRECTION REQUESTED

4050 Series Applications Libraries

Africa, Europe, Middle East

Contact local sales office

Australia

4050 Series Applications Library
Tektronix Australia Pty. Limited
Sydney
80 Waterloo Road
North Ryde, N.S.W. 2113

Canada

4050 Series Applications Library
Tektronix Canada Ltd.
P.O. Box 6500
Barrie, Ontario
Canada L4M 4V3

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

IDD Group
Export Marketing
Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077
U.S.A.

Japan

4050 Series Applications Library
Sony/Tektronix Corporation
9-31 Kitashinagawa-5
Tokyo 141 Japan

United States

4050 Series Applications Library
Tektronix, Inc.
Group 451
P.O. Box 500
Beaverton, Oregon 97077