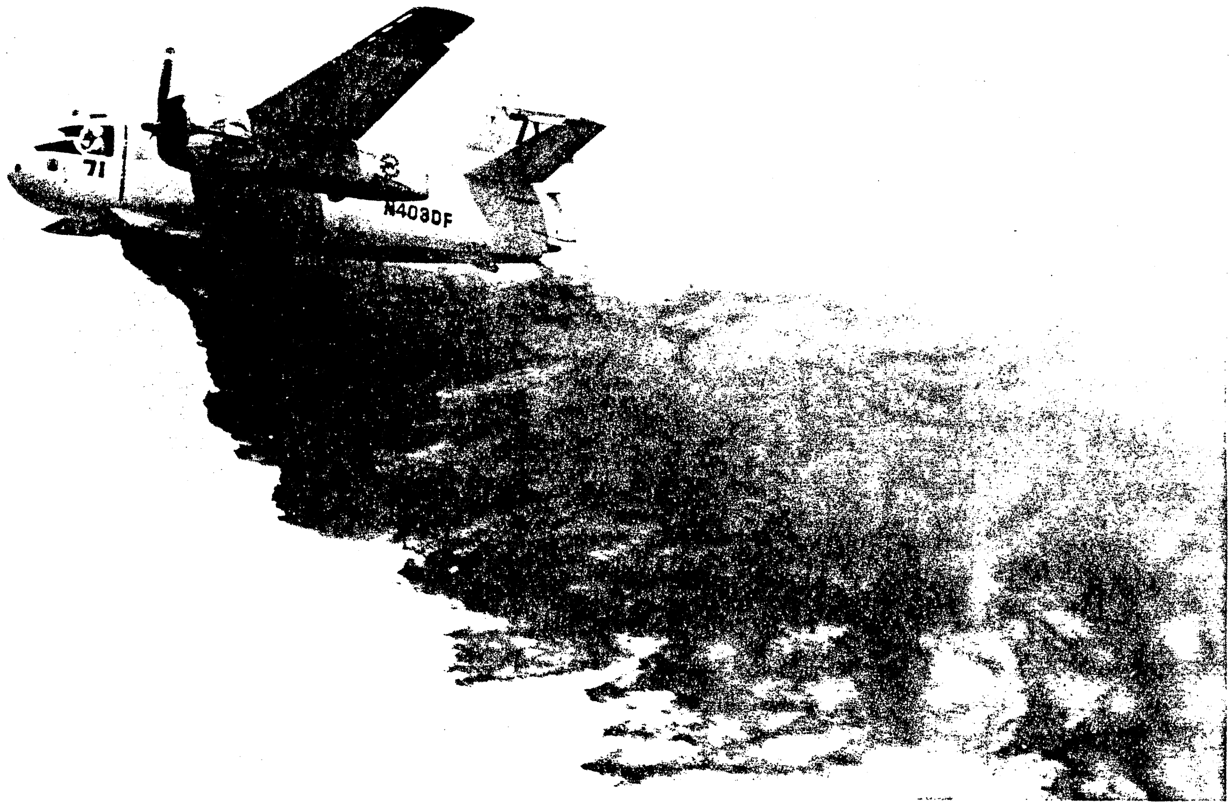
Teknıques

The 4051 Applications Library Newsletter Vol. 2 No. 2



Aerial drops of fire retardants are analyzed at the Northern Forest Fire Laboratory in Missoula. How high and how much are two of the questions answered with the aid of a 4051.

4051 Aids Fire Fighting Research

The Northern Forest Fire Laboratory in Missoula, Montana is applying the 4051 Graphic System to research into improving the effectiveness of forest fire fighting techniques. The laboratory, operated by the United States Forest Service, is using a 4051 coupled to a 4662 Plotter to aid in the development of performance guides for aircraft used to drop fire retardants on forest fires. These guides provide vital information about retardant drop characteristics for different air tanker configurations, drop heights, and retardant types. This information allows firefighters to obtain maximum benefits from aerial retardant drops. The following is a summary of their research, and the 4051's part in it.

Some Background

Air tankers dropping fire retardants have played a vital part in fighting forest fires since 1956. The effectiveness of such air retardant drops in combating forest fires is influenced by many factors; these factors have been under study at the Northern Forest Fire Laboratory since the early 1960s. Effectiveness is largely measured through reduction in fire spread, and the resultant reduction in forest damage. In addition, increased costs of labor and

In This Issue

material have raised the price of a retardant drop to the point where cost is a major factor as well. Today, a single drop from a DC-6 or DC-7 air tanker can cost $1500 for the retardant alone, not including the aircraft costs.

Through study of the characteristics of various wildfire types and delivery methods, the laboratory has been able to ascertain which are the major fire-suppressing elements in the liquid retardants used in aerial drops. These elements relate to the type of fuel (grass, wood types, brush, etc.) and the density of the fuel; research has found elements to be the following.

fuel coating (from the water content and thickeners),

fuel cooling (from the water content), and

modified combustion reaction (from the added retardants salts).

Manipulating these elements to obtain effective coverage levels, particularly fuel coating and cooling, can be thought of as a function of the delivery technique. In air delivery, unlike urban firefighting, all of the retardant is applied in a very brief time—a matter of seconds. The fuel surface (tree surfaces, etc) must retain an adequate amount of retardant to extinguish, knock down, or retard the fire. Fuel surfaces, however, are limited in the amount of retardant they can retain; any excess application is ultimately lost as runoff.

The key to effective use of liquid retardants is determining the most effective coverage levels for various fuel and retardant types. Once these are found, methods can be developed for obtaining the desired coverage level over the most beneficial surface area or line length. The recommended effective coverage levels are based on studies of fuel surface capacities, retardant salt effects, and moisture damping effects. Coverage levels are related to models of forest fuel types developed by the laboratory for the National Fire-Danger Rating System; they are shown in Table 1.

Table 1
Recommended Coverage Levels

Coverage Level	Recommended For: Fuel Model	Description
1	A	Grasses, tundra, and open desert shrublands.
2	C	Conifer timber with grass understory.
	F	Brushfields with much green material.
3	D	Shrublands under 6 feet in height (such as sagebrush).
	E	Hardwood litter after leaf drop.
	H	Closed conifer timber with litter only.
4 OR MORE	B	Hard chaparral and other high (6-foot) flammable shrubs.
	G	Closed, old conifer timber with letter and dead, down limb wood.
	I	Conifer slash.

For creeping or smoldering fires, reduction of one coverage level may be considered.

Coverage levels, expressed in gallons per 100 square feet, are derived from the surface area of the fuel models and the effect of retardant salts on the spread rate of typical fuels. Research currently in progress is providing guides for air tankers, enabling firefighters to obtain effective coverage levels over an optimum area. This information is published in the form of Best Strategy Charts and graphs of Pattern "Footprints" for specific air tanker types. Generating these charts and graphs is where the 4051 Graphic System comes in.

Predicting Drop Patterns

The Northern Forest Fire Laboratory generates the data for their performance guides using a simulation model called PATSIM (Pattern Simulation), developed by Honeywell. The simulation model requires, as its raw data inputs, flow rate and door opening times obtained from on-the-ground tests of the tank and gating system. Each tank of the aircraft is instrumented with a float attached to a pulley and potentiometer, another potentiometer that measures the amount of door opening, and a pressure transducer. The tanks are filled to capacity (up to 3000 gallons of water), and data from the potentiometers is recorded on a storage oscilloscope as the load is released (Fig. 1).

The oscilloscope data traces are then photographed, and the photographed data traces are digitized using the 4956 Graphics Tablet. The digitized data is stored on tape. (Interfacing equipment will soon be in place in the laboratory to allow recording data directly onto tape using the 4924 Tape Drive.) The digitized data is then converted to flow rates and door opening rates by the 4051, utilizing algorithms based on the geometry of the particular tank.

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

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

Volume 2 Number 2.

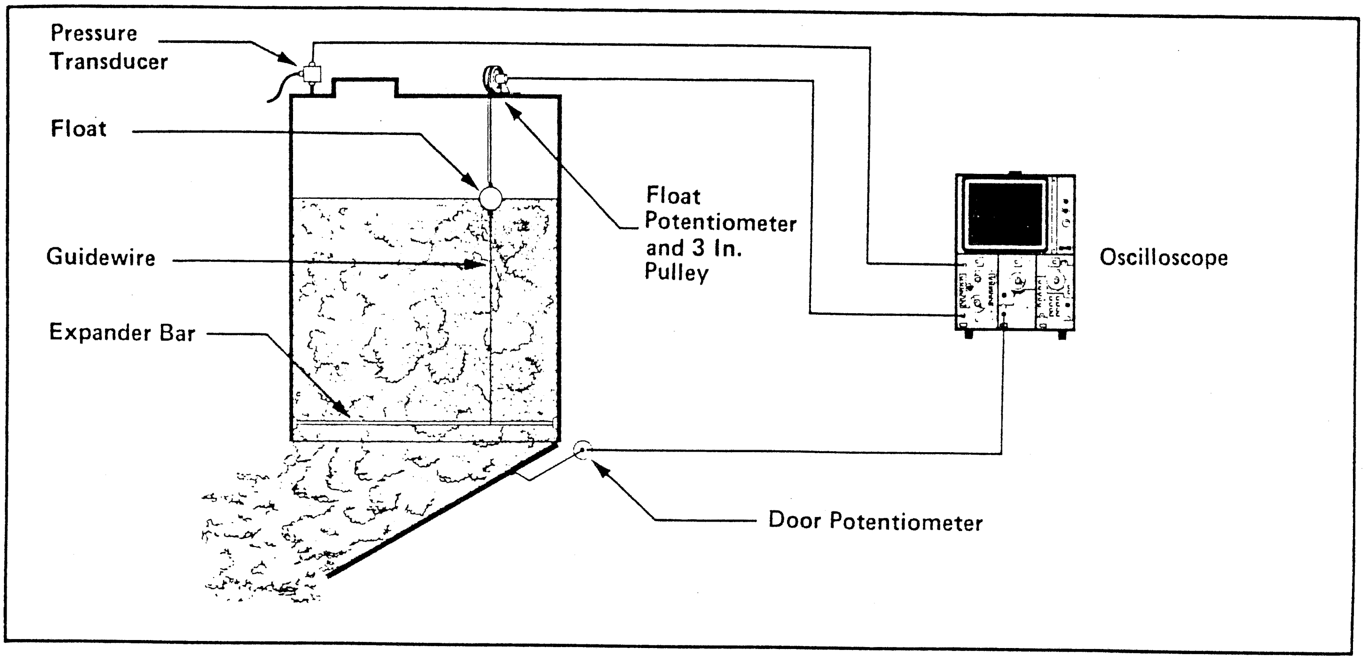


Fig. 1. Equipment Schematic.

The flow rates and door opening rates (angles) are examined to determine the most representative for each aircraft tank and gating system, and the information entered into PATSIM. (PATSIM has not yet been converted to BASIC, so the simulations based on the flow rates and door angles are run on a CDC 7600.) Since a simulated flight/drop test takes less than a second of computer time, a series of computer simulations examining a wide range of possible aircraft velocities, drop heights, and retardant types can be conducted in a few days. Portions of the simulated ground coverage pattern are then output to the 4051; the attached 4662 Plotter is then used to produce the charts of best strategy, and the pattern "footprints" that are reproduced in the air tanker performance guides. Guides have currently been developed for several different aircraft, and have been verified using flow rate data collected in on-the-ground tests described above, as well as in actual drop tests.

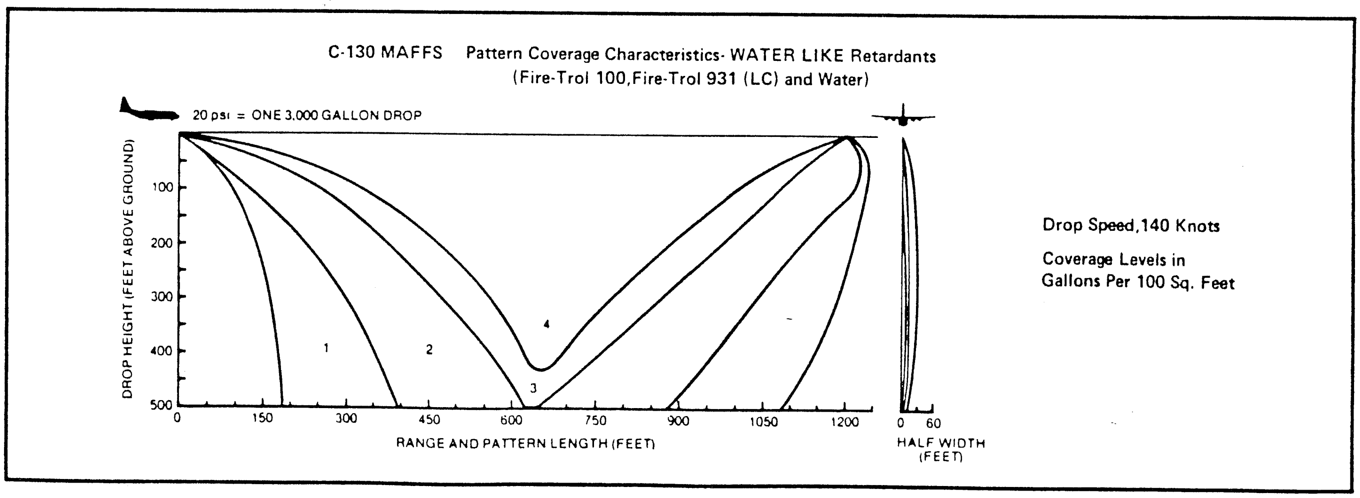
Each performance guide contains the following:

General Information—A brief description of the aircraft and its tank and gating system.

Retardant Coverage Requirements—The recommended coverage level for each of the National Fire Danger Rating System fuel models, reproduced in this article as Table 1.

Pattern Footprints—A plot of retardant pattern coverage levels for various types of drops and retardants, with instructions for interpretation. These footprints are developed by the 4051, and plotted on the 4662 Plotter. Fig. 2 shows an example of a 4662 plot, scaled by the plotter to the desired size and reproduced for incorporation in the air tanker guide.

Fig. 2. Pattern Footprints — A plot of retardant pattern coverage for various types of drops and retardants.



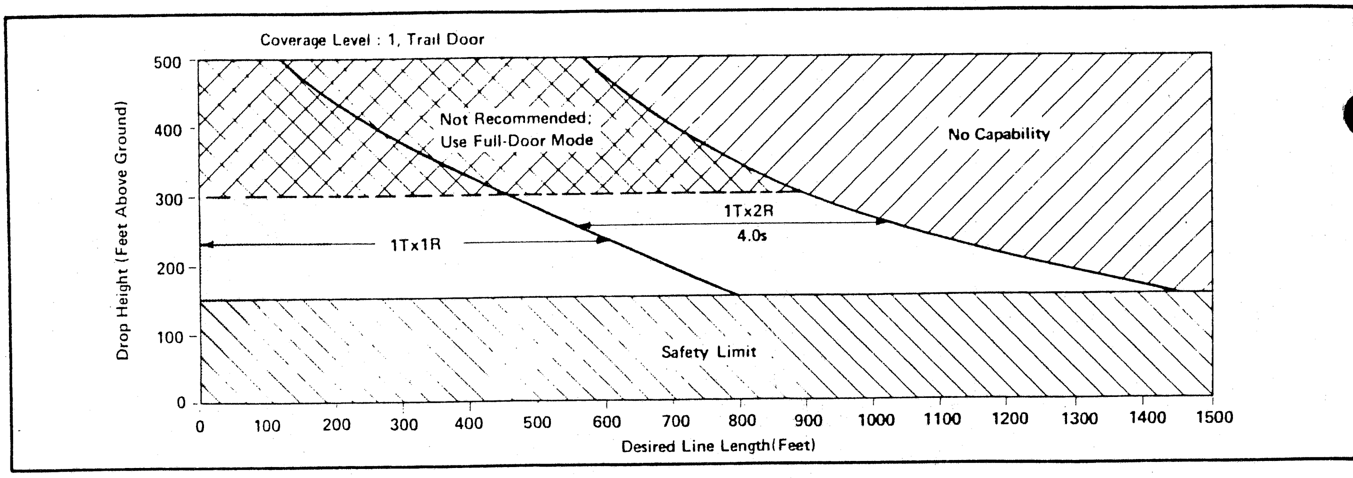


Fig. 3. Best release strategy chart for a single-tank trail drop of a waterlike retardant (Fire-Trol 100, Fire-Trol 931 (LC) and Water) from Aero Union B-17.
T = Tank(s) released at one time. R = Successive releases. s = seconds between successive releases.
Shaded Area = Region of limited accuracy in direct attack.

Best Strategy—Enables quick selection of the best number and sequence of tanks to be employed for specific coverage requirements and line lengths. General capabilities and safety limits are also shown. These charts, like the Pattern Footprints, are produced by the 4051 4662 System. Such a chart, for the Aero Union B-17 Tanker, is shown in Fig. 3.

Detailed Tables—These provide the basis for developing the Best Strategy charts. They show the expected lengths of retardant lines and specific intervals for successive tank releases to achieve maximum coverage at a large number of coverage levels and altitudes.

Use of these air tanker performance guides can optimize the use of the specific air tanker, as well as providing a means for comparing the performance of air tanker types. The performance guides, particularly the graphic quick reference charts, also provide a quick method for identifying the most effective drop method for the local situation. The intent of the guides is to help understand aircraft tank and gating system limitations and assets, and to lead to safer and more efficient aerial application of fire retardants.

The Northern Forest Fire Laboratory, established in Missoula, Montana in 1960, is one of three forest fire research stations operated by the USDA Forest Service. The others are located in Macon, Georgia and in Riverside, California. The Montana Laboratory conducts research of a regional nature, pertaining to the difficulties of fire fighting in northwest forests, as well as their research of national interest into retardant development and delivery systems. They are the only U.S. laboratory conducting research in the latter two areas. Other research areas in the Montana laboratory include fire physics, forest fuels, forest meteorology and lightning research, fire effects, and the fire danger rating system.

4907: Graphic Displays from Segmented Data Base

by Les Brabetz & Gary P. Laroff

(This is the third of a three-part article on the new 4907 FILE MANAGER intelligent flexible disc mass storage unit as a powerful graphics aid. The first two articles, in TEKniques Vol. 1 No. 10 and Vol 2 No. 1, described the concept and creation of a segmented data base using the 4907 FILE MANAGER. This article will complete the series by providing background summaries from the first two articles, then will describe segment file selection for creation of a display.)

A segmented data base stores a large, highly detailed graphics display in the form of X-Y coordinate arrays. The storage format is the number of coordinate pairs in an array followed by the coordinate pairs.

$$N, X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n$$

Initially, a data base exists. It is either computed and stored or digitized and stored. Such a data base consists of some number of arrays of DRAW commands. These could be continuous sections of a map. Islands off the coast would be separate arrays because a MOVE is required to get across the water. A good example of a typical data base is the map of the United States discussed in the first article in this series (TEKniques Vol. 1 No. 10) and reproduced here as Fig. 1. The disadvantage of such a typical data base is that it can only be accessed serially; that is, to see the information stored at the end of the data base, one needs to READ all of the earlier information.

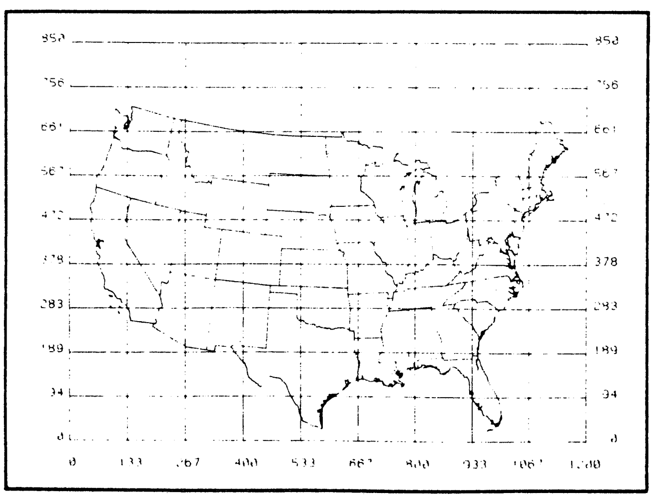
In order to access the graphic data quickly, the data base is segmented into a number of rectangular sections, each of which is stored on the 4907 disc as a separate file. The program that does this was discussed in the last issue of TEKniques (Vol. 2 No. 1). The trick is to compute and store all points where the map crosses one of the straight "segment boundaries.". Any segment may now be called and the data read as quickly as any other.

In summary, each segment or portion of the display consists of a file containing, in theory, an equal number of coordinate pairs. Actually the number of pairs in an array may vary widely. Display or vector generation from the array is performed by MOVing to the first coordinate pair and DRAwing through the remainder of the array. Creation of these segment files is performed by scanning the master data base and selecting those vectors that lie within the defined portion of the display.

Once the segment files have been generated, a means of selecting and displaying the necessary files to build up a display is required. Each segment file is considered a separate entity, independent of the other files in the data base. These building blocks are selected and fit together to fill a defined area. The division of the map of the United States into 96 segments is shown in Fig. 2.

So now that we know how to create a segmented data base, how do we pick the data segments? Being numerical data, the segments have X and Y coordinates associated with each corner. If we then number the segments, we will be able to write a program that will convert minimum and maximum values of X and Y into segment numbers. More precisely, if we DRAW a grid with coordinates over our master data display, we can then enter minimum and maximum values for X and Y (defining the lower left and upper right corners of the rectangular segment, respectively) and have our program look up all of the graphic segments that make up the rectangle. Our looked-up segments will usually form a slightly larger section of the data base than requested.

Fig. 1. Master file of digitized map with arbitrary grid values.



Our program would then set the graphic WINDOW to the requested rectangle, READ the appropriate numbered segments and display them. Easier said than done? Not really. Let's go a little deeper.

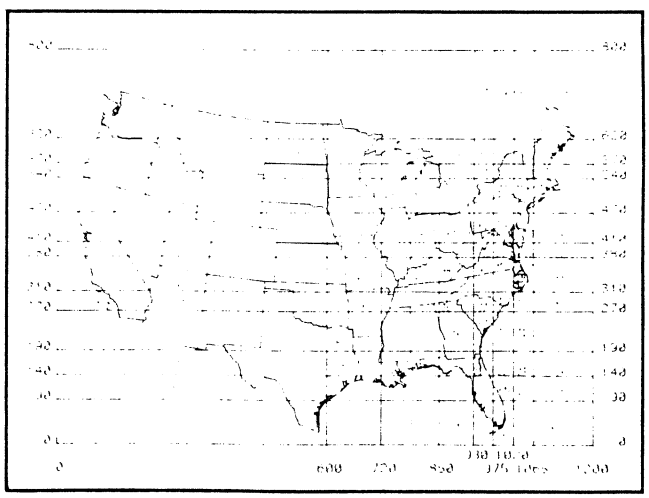
Selecting the Display Area

The 4907 FILE MANAGER demonstration program, which displays the U.S. Map graphics data base, uses the graphics cursor to select the area to display. First the master data base file of the entire map is displayed. The user is requested at this point to select a rectangular area of interest. The graphics cursor is enabled through the POINTER command and the 4952 Option 2 Joystick positions the cursor. When the user positions the graphics cursor to the lower left corner of the desired rectangle, a key is struck on the 4051 keyboard to indicate selection of the location. This returns the X-Y coordinate of the location (in user defined window units) and the character struck. This process is repeated for the upper right corner. The upper and lower limits of the X and Y ranges are now defined. This is only one method of selecting the area. These coordinates could be entered from the keyboard, defined by an operating program, or any other appropriate technique to define the numerical range of interest.

Determining the Segments

The previously defined X and Y coordinates which delineate all the segment boundaries now reside in two arrays within the 4051 memory. Selecting the segment files is performed by comparing the range of interest with the segment boundary values and choosing the segment boundary values which encompass the range of the desired display. Most of the time the segments to be displayed exceed the defined display area, and the internal clipping features of the 4051 display (WINDOW and VIEWPORT) are used to limit the vectors being displayed.

Fig. 2. Map data base illustrating segment division lines and numbering scheme for identification of segments.



A simple example will illustrate selection of actual segments. Each segment file is named to correspond with a location in the matrix in Fig. 3, e.g., SEG1 through SEG12. The data range to be displayed will be:

$X_{min} = 150$ $Y_{min} = 50$
$X_{max} = 250$ $Y_{max} = 250$

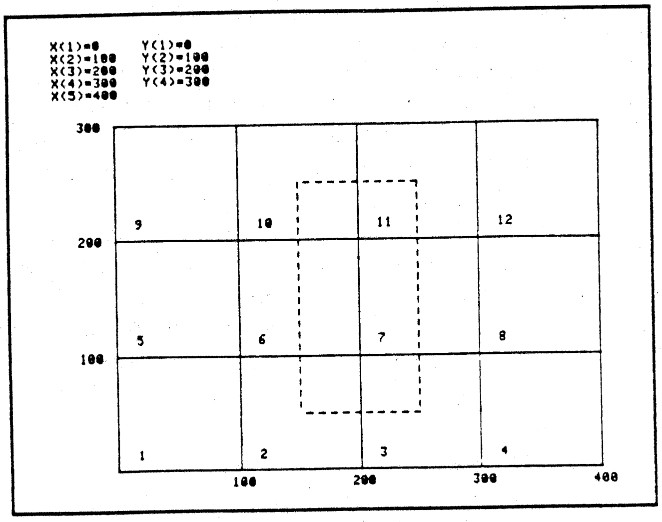


Fig. 3. Example of segment numbering scheme and selection of segments required for a display.

The segment files are chosen from the data base in the following fashion. First the X range is determined by comparing the minimum value with the array values. X(2) is selected as the segment boundary which is less than or equal to the minimum. X(4) is selected as the maximum segment boundary by containing a value greater than or equal to the maximum value. Y(1) and Y(4) are selected in the same fashion. We now know the X range of segments to be 2 to 3 and the Y range as segments 1 through 3.

Calling the Segments

Segments are selected by a row and column method with the row determined by the Y segment range and the column by the X segment range. Since 4907 FILE MANAGER segment files are selected by name, rather than by number, string variables containing the segment numbers will be constructed. First we will generate a master string, A$, which will contain all of our required segment numbers. Segment numbers will be separated within A$ by a space. SEGMENT FILE NUMBER = (ROW - 1) * 4 + COLUMN.

```
1000 REM SEGMENT FILE SELECTION
1010 A$=""
1020 FOR I=2 TO 3
1030 FOR J=1 TO 3
1040 A=(J-1)*4+I
1050 B$=STR(A)
1060 A$=A$&B$
1070 NEXT J
1080 NEXT I
1090 END

PRINT A$
 2 6 10 3 7 11
```

When the above section of code is executed the string A$ contains the segment file numbers required to complete the display. The string values are separated by spaces which are used as delimiters when selecting the individual files. For the demonstration data base "SEG" was prefixed to the segment numbers. Thus segment 1 is called "SEG1" and segment 12 is called "SEG12." The files are selected by this process:

```
2000 REM SEGMENT FILE SELECTION AND DISPLAY
2010 C1=2
2020 A$=A$&" "
2030 C=POS(A$," ",C1)
2040 D$=SEG(A$,C1,C-C1)
2050 D$="SEG"&D$
2060 C1=C+1
2070 REM OPEN AND DISPLAY SEGMENT FILE
2080 GOSUB 3000
2090 IF C1<LEN(A$) THEN 2030
2100 REM END OF DISPLAY
2110 END
3000 PRINT D$;" ";
3010 RETURN

RUN
SEG2 SEG6 SEG10 SEG3 SEG7 SEG11
```

Line 2020 appends a space to the end of A$ so that we can pick off the last segment number. Variables C and C1 are used as pointers to the spaces present in A$. C takes on the value of the position of the next space in the string with C1 containing the position of the last space plus one position. D$ in line 2040 becomes the segment file number located between spaces. To complete construction of the file-name, "SEG" is appended to the segment number in line 2050. The filenames are printed to the screen in this sample program. Normally, these filenames would be used with an "OPEN" command to the 4907. The contents would be read and displayed to the screen. The end of file (EOF) condition would be used to end the data input and "CLOSE" the file. Operation of this program ceases when the length of the string is exceeded by the pointer C1 in line 2090.

The routines developed to perform all of the 4907 graphics data base demonstration will be documented and submitted to the 4051 Applications Library. The master data base of the U.S. Map, the segment files and the program to select and display the segment files will be available for use with the 4907 FILE MANAGER.

Now you have the best of all worlds. With the new applications software program the 4051 will interactively retrieve and display sections of complicated graphics data bases with ease and speed. Interactivity is assured by the 4952 Option 2 Joystick and intelligent flexible disc mass storage is provided by the 4907 FILE MANAGER.

4907 Data Base Format Correction

In TEKniques Vol. 1 No. 10 on page 6 the data format for storing the coordinates of the U.S. map were indicated as $N, X_1, Y_1, X_2, Y_2, \ldots, X_n, Y_n$.
This is in error and the correct format is:

$$N, X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n$$

✻ Editor's Note

We're Interested in You

What books have aided you in programming or interfacing your 4051? Let us know title, author and publisher and TEKniques will pass along the information. If you feel inclined to expand and share your particular use of the methodology, so much the better. Let us hear from you.

Flowchart Request

When submitting programs for inclusion in the 4051 Applications Library, part of the accompanying documentation should be a flowchart of the program. The flowchart should detail the program at least to the level of major subroutines; more detail may be required on long or very complicated programs. This will be of great help to other users in understanding your program and its particular branching techniques.

Contest Deadline Close

March 31, 1978, is the last day to send in your entry for the 4051 Computer Aided Design contest. Prizes consist of new tapes recorded with your choice of programs from the 4051 Applications Library and documentation.

First Prize:	15 tapes and 15 programs
Second Prize:	10 tapes and 10 programs
Third Prize:	5 tapes and 5 programs

For details, see TEKniques Vol. 1 No. 10.

NOTE: Contest deadline extended to April 15.

4025 Terminal Interfaces 4051 to TV Displays

by Will Gallant

By now you are probably aware of the new additions to the Tektronix 4020 series of alphanumeric terminals that include graphic display capabilities. But did you know that one model, the 4025, has an output jack that allows its display to be duplicated on ordinary 525 line television monitors? If you want to use your 4051 in a classroom or other environment with multiple displays, a 4025 terminal may be the low cost alternative to a scan converter. The hookup is simple. Take the RS-232 cable from the 4025 and plug it into the RS-232 (Option 1) output of the 4051. Then connect the video output (BNC connector) to the video input at the monitor or TV projection equipment. (See Fig. 1.)

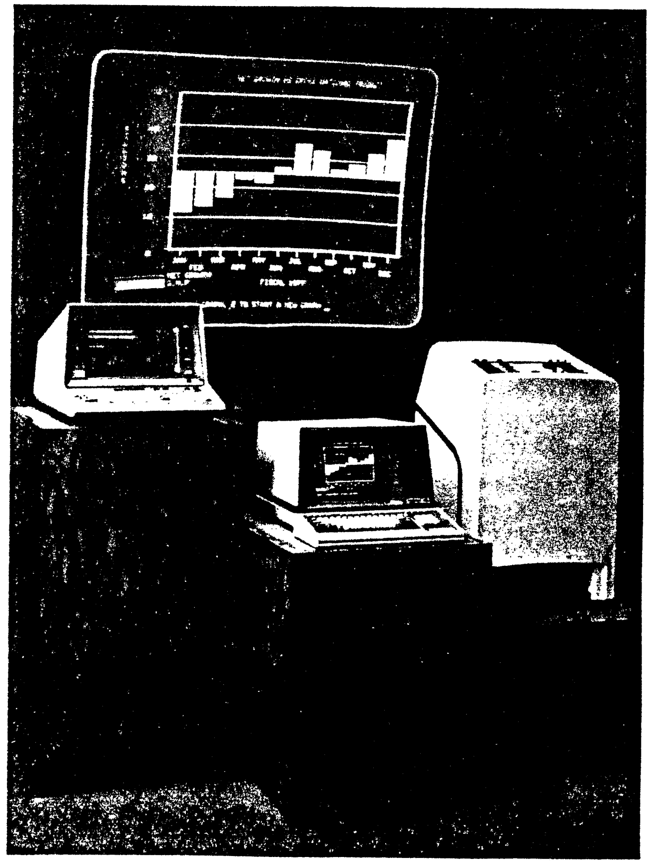


Fig. 1. The 4051 sends graphics information to the 4025, and the graph is displayed simultaneously on both the 4025 screen and the Advent monitor.

Normally the scan conversion process is a two step operation. Picture information is stored in the scan converter and then read out to the TV monitors at television scanning rates. With the 4025 a graph may be developed on the 4025 display screen and simultaneously read out to the display monitors. Since graphics memory in the 4025 is updated between the TV scan cycles there is no interference or tearing of the picture. Conversely, a typical scan converter must give priority to its writing beam at the expense of losing TV scan lines as new data is added to the display.

The 4025 uses a "virtual bit map" graphics memory. That is, memory is allocated to graphics only when it is needed. With up to 32K bytes of graphics memory available, three or four fairly complex graphs may be in the 4025 at one time. In a classroom presentation an instructor could command any one of the graphs to instantly appear on the display by pressing a single 4025 key.

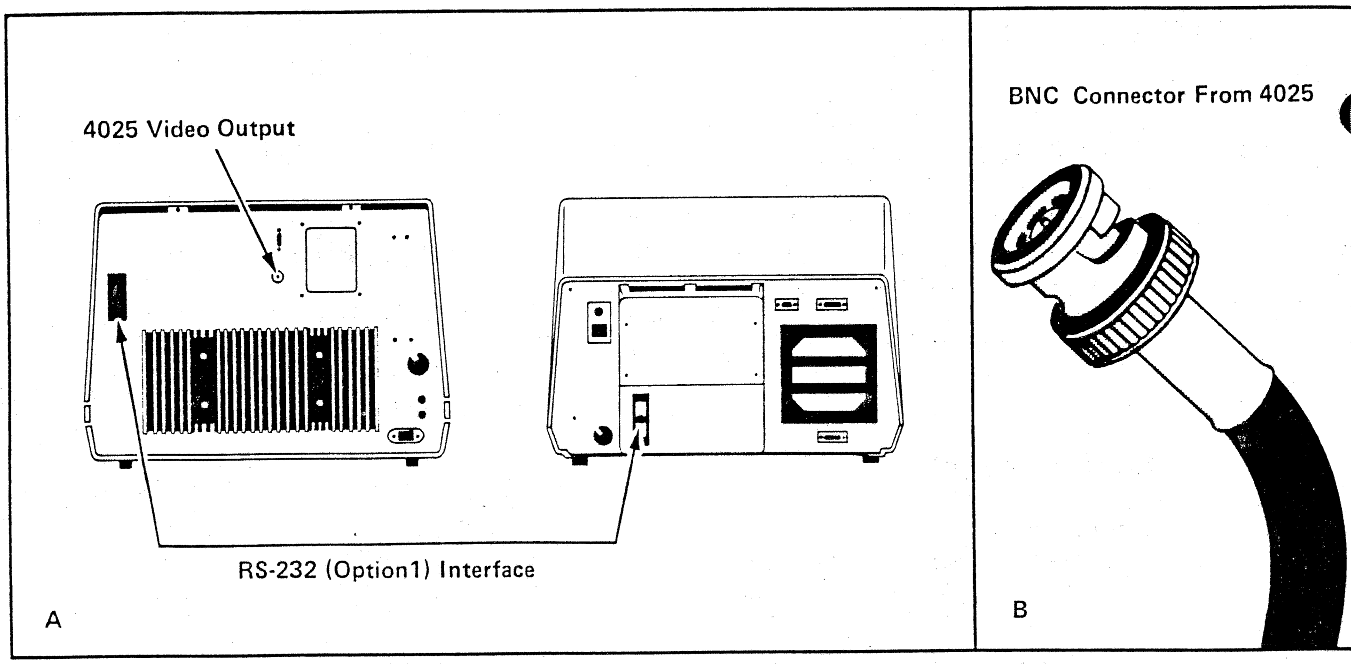


Fig. 2. a) RS-232 cable interfaced from the 4025 to the RS-232 (Option 1) output of the 4051. b) BNC connector leading from the 4025 which is connected to the video input at the monitor or TV projection equipment.

The 4051 communicates with the 4025 through the RS-232 using PRINT @40: commands. The data sent over the cable to the 4025 is actual graphics information or ordinary ASCII text in a simple "English" format.

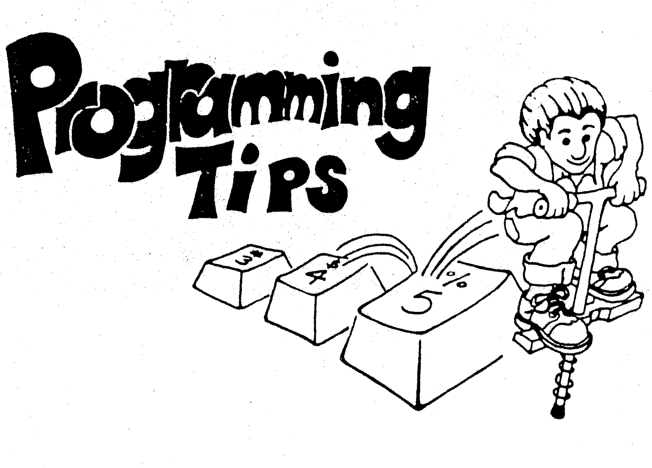
For example:

```
!VEC 0,0,100,0,100,100,0,100,0,0
```

The command !VEC tells the 4025 to switch to vector (graphic) mode and to move the beam to coordinates 0,0. Further coordinate pairs are interpreted as beam-on or draw commands. The commands shown above will draw a box on the display 100 units on its side. Dashed line mode and line erase commands are two unique features of the 4025 controlled by the 4051. A graph may also be developed off screen while the instructor is discussing the current graph displayed on the TV monitor.

For repetitive presentations, the information sent from the 4051 to the 4025 may be captured on a TEKTRONIX 4923 Digital Cartridge Tape Recorder. The displays may be played back later without the 4051. A 4631 or 4632 Hard Copy Unit may also be interfaced to the 4025.

For classrooms and similar environments a 4025 may offer a lower cost method than a scan converter for interfacing the 4051 to multiple or large screen displays. The 4025 also provides the multi-purpose capabilites of a fine alphanumeric terminal. Your local Tektronix field office is equipped to demonstrate any of the features discussed in this article.



4051 FAST Graphics

by Chuck Eng and Nick Fkiaras

The cathode ray tube display of the 4051 is a very high quality instrument capable of drawing speeds usually not exercised. Most programs that output graphics use logic loops that compute X-Y coordinates, followed by a MOVE or DRAW of the coordinates. Even though part of the graphics may remain the same, the "compute coordinates—MOVE/DRAW" cycle is used to regenerate the whole display. In a situation when identical redraws are needed, considerable time may be saved by taking advantage of the drawing speed of the 4051.

The following program example draws graphics using coordinates stored in a numeric array. Line numbers 100 through 330 generate a design through the usual "compute—MOVE/DRAW" loops but at the same time loads the computed coordinates into array Z.* Once the design has been plotted, program lines 1000 through 1040 draw this design by accessing X-Y coordinates from Z array.

```
100 INIT
110 DIM Z(362,2)
120 K=4
130 PAGE
140 Q=32
150 DEF FNX(I)=COS(I)
160 DEF FNY(I)=SIN(I)*0.5
170 DEF FNZ(I)=COS(K*I)
180 SET DEGREES
190 VIEWPORT 0,130,0,100
200 WINDOW -1,1,-1,1
210 MOVE 1,0
220 FOR I=0 TO 360 STEP 2
230 A=FNZ(I)*COS(I)
240 B=FNZ(I)*SIN(I)
250 C=FNX(I)
260 D=FNY(I)
270 DRAW @Q:A,B
280 DRAW @Q:C,D
290 Z(I+1,1)=A*65+65
300 Z(I+1,2)=B*50+50
310 Z(I+2,1)=C*65+65
320 Z(I+2,2)=D*50+50
330 NEXT I
1000 PAGE
1010 WINDOW 0,130,0,100
1020 PRINT @32,21:Z(1,1),Z(1,2)
1030 PRINT @32,20:Z
1040 GO TO 1020
```

Two key factors are involved in the speed increase: elimination of coordinate recomputation from the drawing cycle and the use of secondary addressing for MOVEs and DRAWs (i.e., PRInt @32:21 and PRInt @32:20, respectively). When using secondary addressing, a MOVE or DRAW need not be interpreted by the 4051 processor on each vector allowing it to devote its full attention to picking X-Y coordinates from the Z array and drawing on the screen. It is important to note that plotting with secondary addressing will use only the default WINdow and VIEwport limits of 0,130,0,100.

*When running this program, note the drawing speed on the design generation and compare it with the speed in the redraw.

Prompting Fixed Length Inputs

by Ed Mitchell

In many applications, such as data base manipulation, a field of fixed length is reserved for the data entry. The GIN, MOVE, and DRAW commands can be used to generate a line on the screen for prompting the operator as to the size of the available field.

Using this method, you can enter a name of any length for the field. The underline subroutine then calculates if the fixed length field will fit on the same display line, or if a new line is required. A line the length of the field is then generated in the appropriate location. This subroutine eliminates the need for printing a number of underline characters, and prevents wraparound from occurring as well.

In the example program the operator is to input two fields. These fields, F$ and G$, are not to exceed a length of 15 and 68 characters, respectively. Line 140 sets the maximum length for F$ to 15 and line 150 prints the title for the field. The program then branches to the underline subroutine to calculate if the entry will fit on the same line and draws the underline. Lines 190 through 210 check that the entry did not exceed the maximum length.

The same process occurs in line 230, only this time the combined length of the title and field exceed the length of one line of print on the screen. In this case the subroutine spaces down one line, then draws the prompt underline. This eliminates wraparound.

```
100 REM GRAPHICS PROGRAM FOR DATA ENTRY.
110 REM
120 REM ENTER A FIELD NOT TO EXCEED A LENGTH OF L .
130 REM
140 L=15
150 PRINT " FIELD 1 :";
160 GOSUB 320
170 INPUT F$
180 PRINT
190 IF LEN(F$)<=L THEN 220
200 PRINT "FIELD OVERFLOW GJ"
210 GO TO 150
220 L=68
230 PRINT " FIELD TEST 2 :";
240 GOSUB 320
250 INPUT G$
260 PRINT
270 IF LEN(G$)<=L THEN 300
280 PRINT "FIELD OVERFLOW !GJ"
290 GO TO 230
300 END
310 REM************************ UNDERLINE SUB.********************************
320 GIN X,Y
330 MOVE X,Y-0.3
340 IF X+(L-0.44)*1.72<127 THEN 400
350 PRINT
360 GIN X,Y
370 MOVE X,Y-0.3
380 DRAW X+(L+2.42)*1.72,Y-0.3
390 GO TO 410
400 DRAW X+(L-0.44)*1.792,Y-0.3
410 MOVE X,Y
420 RETURN

RUN
 FIELD 1 :ABCDEFGHIJKLMNO

 FIELD TEST 2 :
1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ(!"#$%&'()@*=)↑/*-+**************
```

Single-Character Variables Save Memory

by Leland C. Sheppard
Sunnyvale, CA

When you have a tight memory situation, it may be possible to reduce your program size by using the following technique.

Make sure that you have used all single-character variable names. Substitute single-character names for the two character names that are most often referenced. You can pick up as much as 400-600 bytes of memory in a typical program by using this method.

For/Next Loop Revisited

by Ms. Bobbie Smith
University of Miami

TEKniques Vol 1, No. 9 contained the programming tip "Saving Memory, For/Next Loop Exit". It should be noted that the program example given could create an improper call to the I[th] element of J if the "real" loop does go to completion. One way to get around this is by the following:

```
100 READ @33:N
110 DIM J(N)
120 READ @33:J
130 FOR A0=1 TO 1
140 FOR I=1 TO N
150 IF J(I)=0 THEN 180
160 NEXT I
170 NEXT A
180 IF I<=N THEN 210
190 PRINT "LOOP WENT TO COMPLETION.  I= N+1."
200 GO TO, etc.
210 J(I)=J(I)+1, etc.
```

This method takes advantage of the fact that the variable I will always be equal to N + 1 if the loop goes to completion.

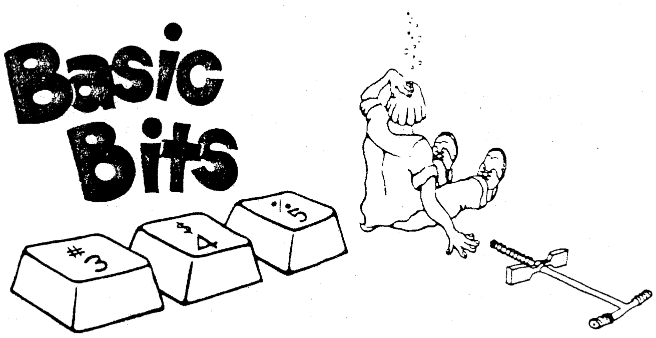
Multiple Use UDK Subroutine

by Joe Gamble

Here is a relatively simple way of getting the equivalent of as many User-Definable Keys (UDK's) as one wants. The routine counts the number of times each UDK is depressed, and uses this as an index.

In a more practical program, the "PRINT D..E..F.." would normally be a series of GOTO's, and I now prefer a slightly different sorting system. The technique may be extended to all the UDK's, and there is no limit to the number of strokes as long as they are made less than 0.9 seconds apart and as long as the key is not held down longer than 0.9 seconds. Of course, a longer delay could be used to give a greater safety margin.

```
lis
1 SET KEY
2 K=0
3 END
4 K=K+1
5 GOSUB 85
6 GO TO 200
8 K=K+1
9 GOSUB 85
10 GO TO 100
12 K=K+1
13 GOSUB 85
14 GO TO 350
85 REM>> 1sec delay
86 FOR J=1 TO 16
87 PRINT @32,24:"x"
88 NEXT J
89 RETURN
100 IF K=2 THEN 150
110 IF K>2 THEN 170
120 PRINT "D";
130 GO TO 2
150 PRINT "E";
160 GO TO 2
170 PRINT "F";
180 GO TO 2
200 IF K=2 THEN 250
210 IF K>2 THEN 300
220 PRINT "A";
230 GO TO 2
250 PRINT "B";
260 GO TO 2
300 PRINT "C";
310 GO TO 2
350 PRINT K;
360 GO TO 2
370 END
```



DIMensioning Strings

Upward

By default a string (A$, B$, . . .) is dimensioned to 72 bytes. To gain additional space, the string variable must be DIMensioned:

```
100 DIM A$(200)
```

200 bytes are now reserved for A$. Once a DIM command has been executed for a string, to gain additional space, the variable must first be deleted and then re-DIMensioned:

```
100 DIM A$(200)
!
!
430 DELETE A$
440 DIM A$(350)
```

Downward

If, later in the program, less space is required for A$ and the bytes are needed elsewhere, a DELete command must be issued before re-DIMensioning the string:

```
100 DIM A$(200)
!
!
380 DELETE A$
390 DIM A$(50)
```

If the string is simply dimensioned downward without the DELete command, the working space is reduced but the space allocated in the original DIM statement is still reserved for that variable.

Refer to the DIM segment of the LANGUAGE ELEMENTS section in your 4051 Graphic System Reference manual for a discussion of the details using this statement.

Writing Subroutines

by Herman H. Kan

When developing and coding a program with subroutines, it is very helpful to start each subroutine on a separate page. In addition, each subroutine should start with an even line number, such as n00, and should end with a 99 line number (n99). This aids in flowcharting the program during development, and allows for easier debugging and expansion. Refer to the Programming Tip in TEKniques Vol. 1 No. 7, entitled "Renumbering Programs".

Coding Designates Delay

by A.W. Leigh
Liverpool Polytechnic

TEKniques Vol. 1 No. 10 contained a Basic Bit entitled "Save Your Memory," that provided a technique for a waiting loop. The delay provided by this method is fairly short; for a longer delay, try the following coding:

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

The delay provided by this method is approximately 10 seconds. The delay length can be changed by changing the first repeat operator, which gives the approximate delay in seconds.

4051 Applications Library Program Abstracts

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

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

ABSTRACT NUMBER: 51/00-0401/0

Title: **Arbitrage**
Authors: Dr. P.C. Holman, Michael Voica, Janet Bruegl
University of Wisconsin/Stevens Point
Memory Requirements: 32K
Peripherals: 4631 Optional
Statements: 341

The November, 1977 exchange rates of 158 principal currencies of the world based on the U.S. dollar are contained in an ASCII data file. The program will allow the user to look at an exchange rate, change an exchange rate, or convert a designated amount from one country's currency to that of another.

ABSTRACT NUMBER: 51/00-6104/0

Title: **Grade Recorder**
Author: P.J. Fulford
Purdue University
Memory Requirements: 32K
Peripherals: 4051R05 (Binary ROM)
Optional: 4662 Plotter
Statements: 768

A teacher's course grade book is automatically set up and maintained on tape. Maximum limits are approximately 70 students, 20 grades and 20 courses, but these may be easily changed. Computation of averages, standard deviations, and weighted composite totals are features of the program. Sorting by rank and name are also available. Grades may be added, deleted or changed. Distributions are graphically displayed as a histogram.

A dedicated tape is presumed with the program as the first three files. The first file is a short ASCII program to call the Binary loader for the second file. The second file then links the third. The remaining portion of the tape is used for the binary data. The data files are automatically marked as needed.

ABSTRACT NUMBER: 51/00-5504/0

Title: **Cubic Spline Interpolation**
Authors: Monty McGraw and Jim Yadon
Memory Requirements: 8K
Peripherals: None (Optional—4631 Hard Copy Unit)
Statements: 133

This program will fit a smooth curve to up to 25 ordered X-Y data pairs by generating piecewise cubic equations of the form:

$$Y_o = A_3X_o^3 + A_2X_o^2 + A_1X_o + A_o$$

The cubic equations allow up to 290 interpolations of Y values and derivatives of the curve between data points. The program allows keyboard or tape entry of the data points and records the interpolated values on tape. The interpolated data may be plotted by using the 4051 System Software X-Y Plot Program.

ABSTRACT NUMBER: 51/00-9520/0

Title: **Digitize and Draw**
Author: Dr. R.J. Reimann
Boise State University
Memory Requirements: 8K
Peripherals: 4662 Plotter Optional
Statements: 175

This program enables the user to easily draw figures or trace transparencies placed on the 4051 screen. The User Definable Keys are incorporated to move the cursor horizontally, vertically and diagonally, and digitize and store the selected points. The slope of diagonal movement may be altered. User Definable Keys also rotate the figure about a central axis, draw the figure and allow additions to the figure. Minimum resolution approximates that of the graphic display itself.

The points are stored in a binary data file for each stage of the drawing. Accumulated files are automatically linked together to reproduce the drawing. Output may be made to the screen or 4662 Plotter.

ABSTRACT NUMBER: 51/00-8015/0

Title: **Flow Diagrammer**

Author: Keith S. Reid-Green
Educational Testing Service
Princeton, NJ

Memory Requirements: 16K
Peripherals: 4662 Plotter
Statements: 915

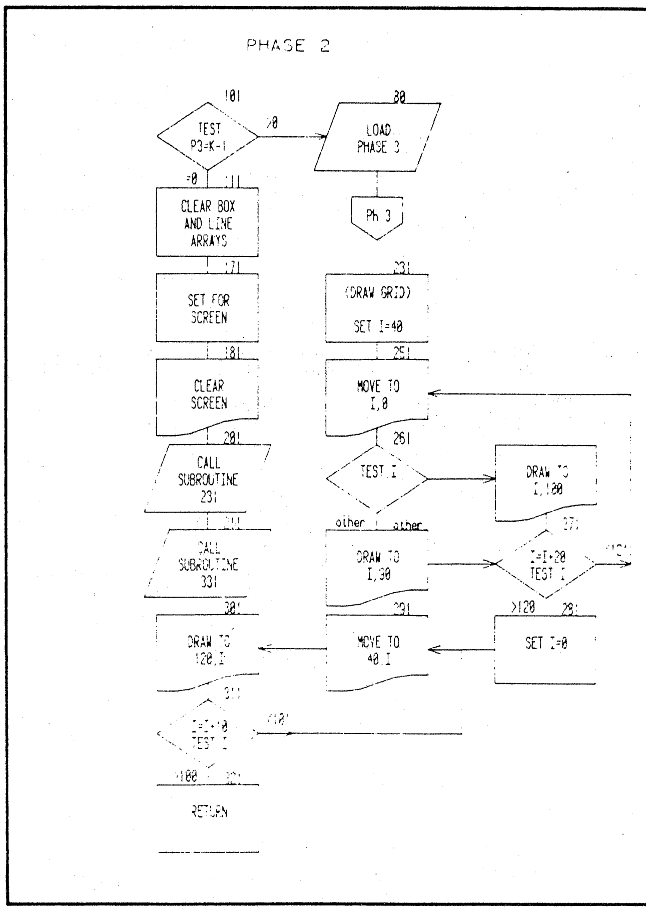
The program is used to design, store, recall and modify flow diagrams for use in program and system documentation.

A diagram consists of a heading, 10 different box types, their connecting lines and labeling. Boxes and lines may be solid or dotted and may be arranged up to 4 across and 9 deep on a page.

The program consists of 5 phases:

1. Main menu and function keys
2. Enter boxes
3. Connect, insert, delete boxes
4. Enter box data and heading
5. Store or retrieve diagrams

The first 62 files of a tape must be dedicated to this program. Files 1 through 5 contain the program; files 6 through 42 contain information about the current diagram; and files 43 through 62 store up to 20 diagrams.



Tektronix
COMMITTED TO EXCELLENCE

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

ADDRESS CORRECTION REQUESTED