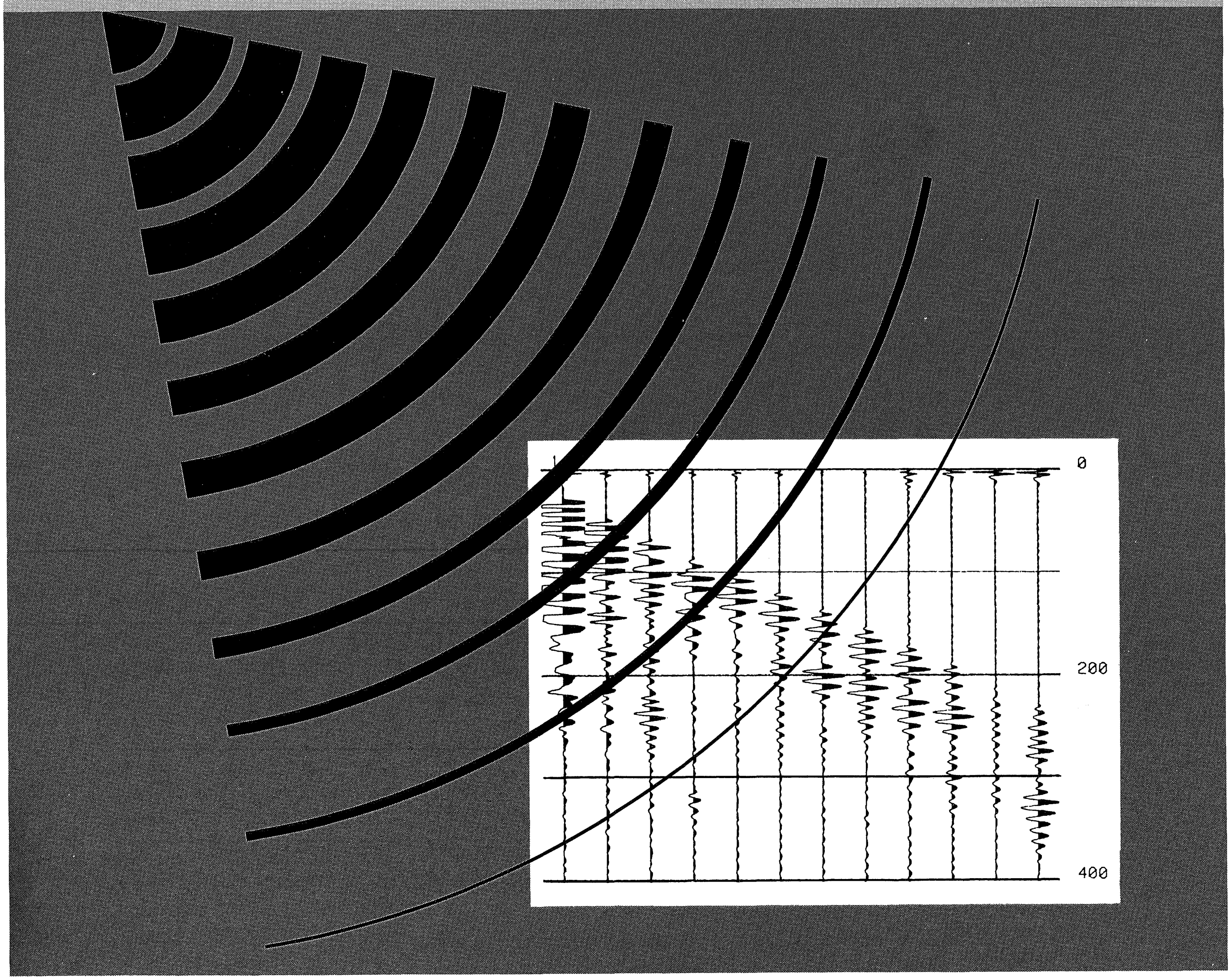
THE 4050 SERIES
APPLICATIONS LIBRARY NEWSLETTER

VOL. 6 NO. 1
SPRING 1982

Tekniques

4052 Helps Interpret Seismic Waveforms



Tektronix®
COMMITTED TO EXCELLENCE

Tekniques

In This Issue

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
Technical Editor	Dan Taylor
Graphic Design	John Ellis
Circulation	Rory Gugliotta
Typesetting	Jean Bunker

To submit articles to TEKniques or for information on reprinting articles, write to the above address. Changes of address should be sent to the 4050 Series Library serving your area (see Library addresses).

4052 Helps Interpret Seismic Waveforms

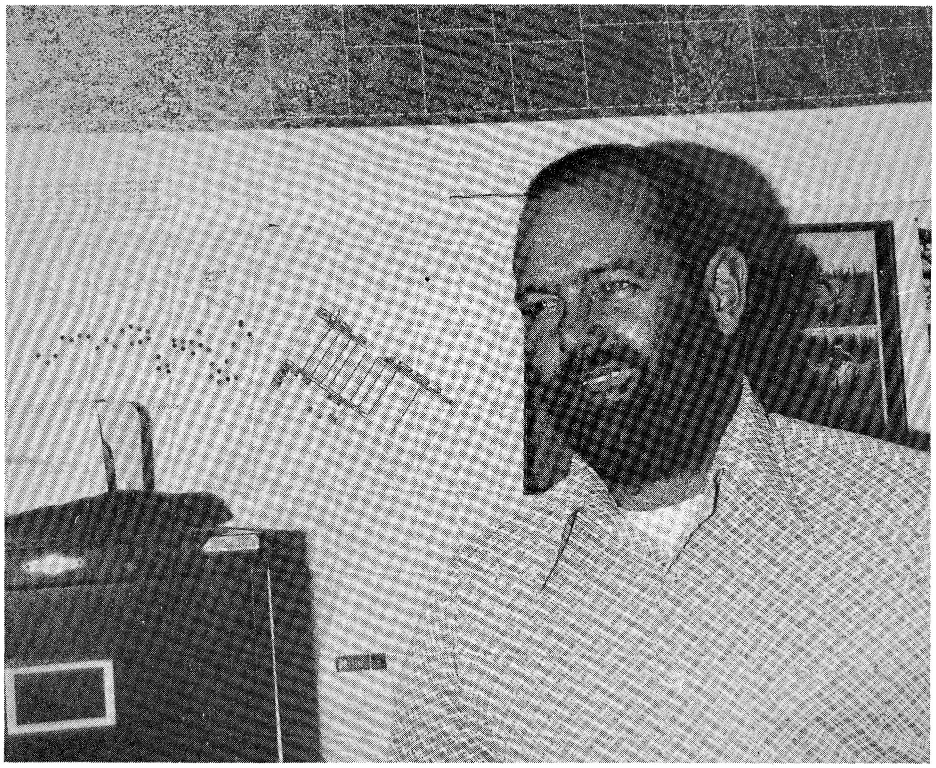


Figure 1. At the University of Kansas, Dr. Don Steeples is adding to the sum of seismic knowledge with the aid of a Tektronix 4052.

by Patricia Kelley
TEKniques Staff

"We deal with two different types of seismic data," said Dr. Don Steeples, "exploration seismic data and earthquake data." In his capacity as Chief of Geophysics at the Kansas Geological Survey (a research division of the University of Kansas in Lawrence), he reflected that they are a state agency, so most of their funding and work involves the State of Kansas. However, the Corps of Engineers and the Nuclear Regulatory Commission have also provided funds to them for seismic studies related to nuclear power sites and federal dam sites. And some of their undertakings include other states and private industry.

Steeples described the nature of his work and the role played by the Tektronix 4052 Desktop Computer.

Piecing Together Geologic Structures

During the summers, Steeples and his staff extract exploration seismic data from beneath the tall grass plains of Kansas. To do this they string an array of sensors — geophones — along the ground and connect them to a 12-channel digital exploration seismograph and magnetic tape recorder. With the instruments readied, a carefully placed dynamite shot is triggered in a shallow borehole. From this energy source seismic waves propagate through the earth's subsurface. Their echoes off the deep rock layers are received by the geophones and transmitted to the seismograph which amplifies and digitizes the data and records the waveforms on tape. The source is then moved and the process repeated.

Upon the scientists' return to the University in Lawrence, a Data General Eclipse at the KGS reads the tape and stores the data on disk. This allows the Geophysics group to download from the disk through the data communications RS-232 link to the 4052 at their leisure; they don't have to depend on the availability of the Eclipse.

By processing the acquired waveforms through the 4052, the researchers can deter-

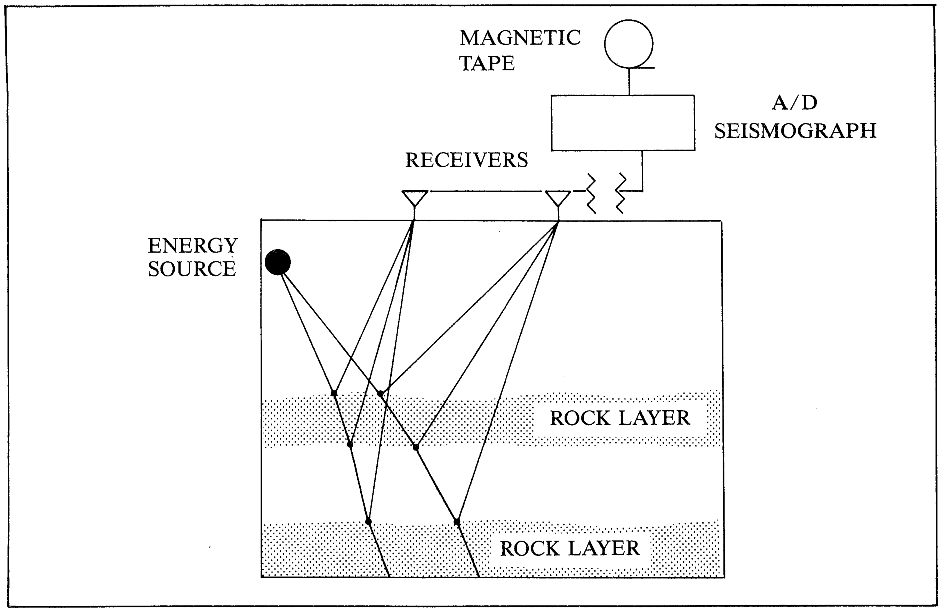


Figure 2. Partial schematic of seismic reflection method.

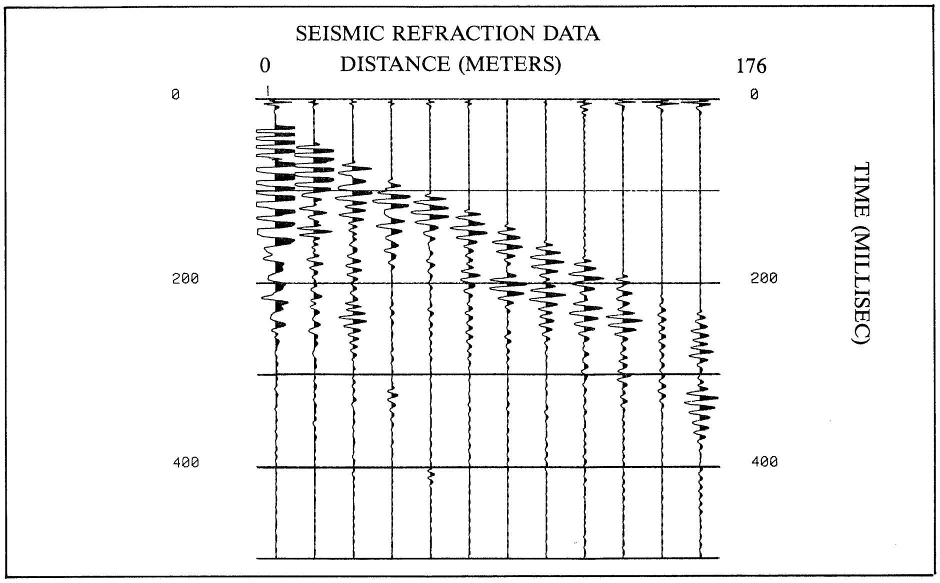


Figure 3. Reflected waveforms from earth's subsurface define geologic characteristics.

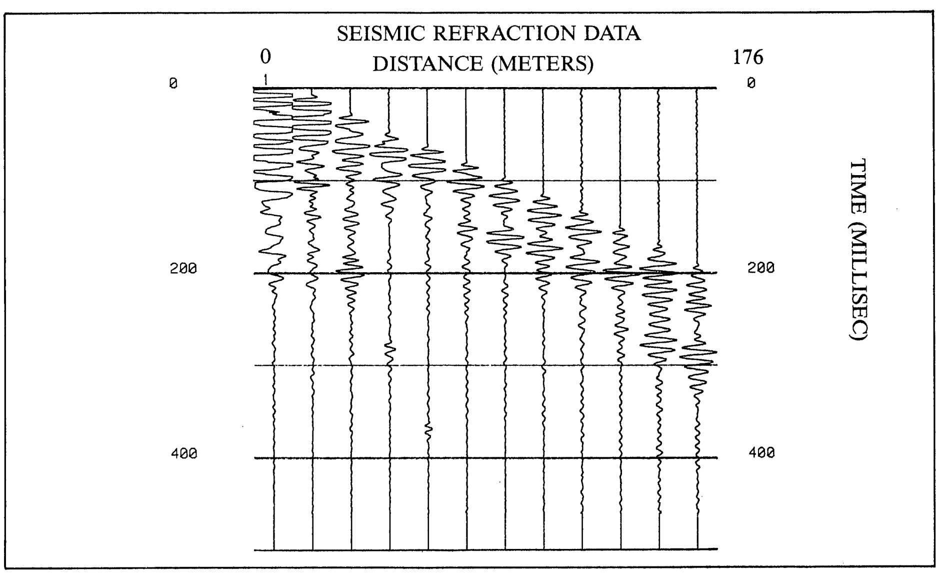


Figure 4. Waveform data shifted to cut initial "noise."

mine a host of geologic characteristics. Amplitude, frequency and velocity changes manifested by the waveforms due to differences in rock formations are all diagnostic tools. Referring to the velocity component of the waveform, Steeples explained that energy propagates through a rock layer at a particular linear velocity. When it encounters a different rock layer, the velocity changes. By calculating velocity paths of adjacent layers and their distinct slopes, an intersection can be obtained. Linking the relative velocities with the intersection distance, the investigators are able to determine the thickness of the upper layer of rock.

Graphic portrayal of the acquired traces is a valuable aid for the researchers. Shlomo Shmuelov, a geophysics and computer science student, demonstrated the program written for this purpose by Dr. Ralph Knapp, a KGS seismologist. The plot in figure 3 displays raw seismic waveforms captured by the geophones, one trace for each geophone. Notice that this particular plot has been enhanced by shading the positive part of each waveform.

Another function of the program filters or shifts the data to eliminate unwanted noise (figure 4).

Steeples and his staff are extending the program to translate the digital signals into black and white images of the subsurface area. In order to produce a plot depicting the rock layers, all the traces that have a so-called common depth point must be summed together. Because of the differences in the physical location of the geophones at the field site, however, the echoes from the layers arrive at the geophones at different times.

Therefore, to calculate a common depth point, the individual traces first must be shifted up or down to compensate for the time differences. Once this is done, they are added together. When plotted, with all the positive peaks in correct alignment and shaded, the different rock layers will be readily discernible — a picture of the earth's subsurface.

"The whole purpose of all the processing," specified Steeples, "is to put the signals in correct geometrical relationships, where the waveforms that are due to this echo from a particular rock layer at depth come in at the same time on the plot."

The Geophysics team is aiming their seismic exploration at more than just possible petroleum deposits. "What we want to do is get the highest possible frequency content into the ground," pointed out Steeples. "The higher the frequency, the better the resolution; the more detail you can see." Most of the oil companies deal with energy sources in the 30 to 40 hertz range. Steeples' group is working on the 200 to 400 hertz range. Therefore, spectrum analysis is occupying a larger part of the researchers' time.

John Vargas, a graduate student, discussed their work with the Signal Processing Rom #2 (FFT). Before they acquired the ROM, Vargas had written a program to handle some of the spectrum analysis. He mentioned that with the ROM they can do in two lines of code what it took 100 to do previously. Plus, the ROM is about "two orders of magnitude faster."

Increasing the frequency content has ramifications in ground water exploration. Steeples said they have recently discovered that they are able to acquire an echo off the top of a water table at depths as shallow as 25 feet. Perfected, this will allow researchers to take readings from, say, the Kansas River Valley. Without having to drill a hole, they could locate the top of the water table, and the bottom of the sand and clay layers in the river valley, thus determining where a well should be located for best production.

Furthermore, tracking the water table would be possible. Today, once a well is drilled, it has to be checked for yield and water table disturbance. This requires that three or four test wells be drilled at a distance from the initial well. After a pumping test is performed from the initial well, the water table level would be checked at the test wells.

With the seismic technique, the initial water table would be established. The producing well would be drilled and pumping tests run for several days. The researchers could then return to the field, run a seismic survey, and tell just how much the water table had dropped at several positions away from the well. While this would not only be cheaper and faster than the traditional methods, it would have the added benefit of less disturbance of the environment.

Deciphering Earthquakes

The second type of seismic data processed through the 4052 comes from earthquakes. Sensors at 11 stations, mainly in Eastern Kansas, register seismic waves resulting from earthquakes. Coming in on long distance telephone lines, the data from each site are recorded on drums in the Geophysics lab (figure 7).

Earthquakes in Kansas, particularly large ones, are relatively rare. But those which do

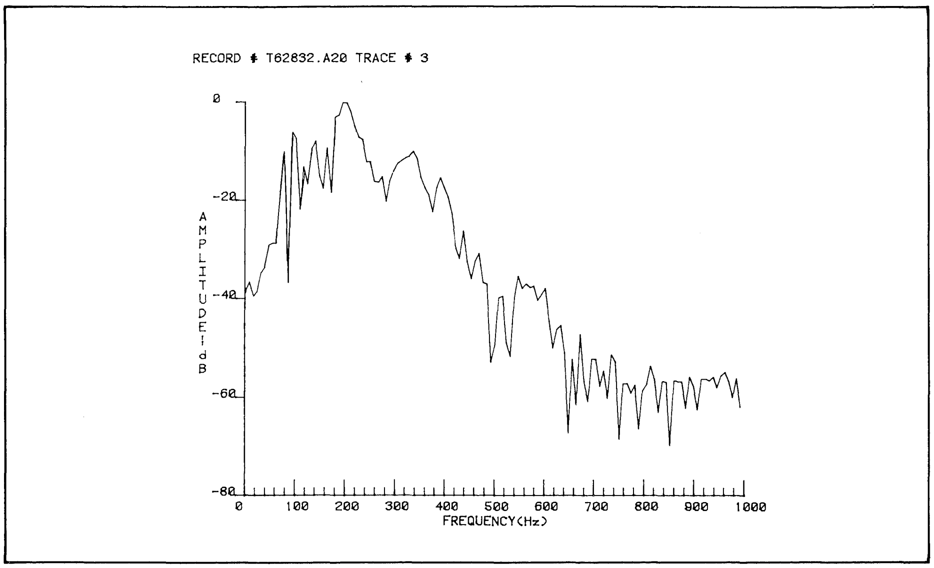


Figure 5. Frequency spectrum of seismic energy obtained by shooting a 22 rifle slug vertically into the ground.

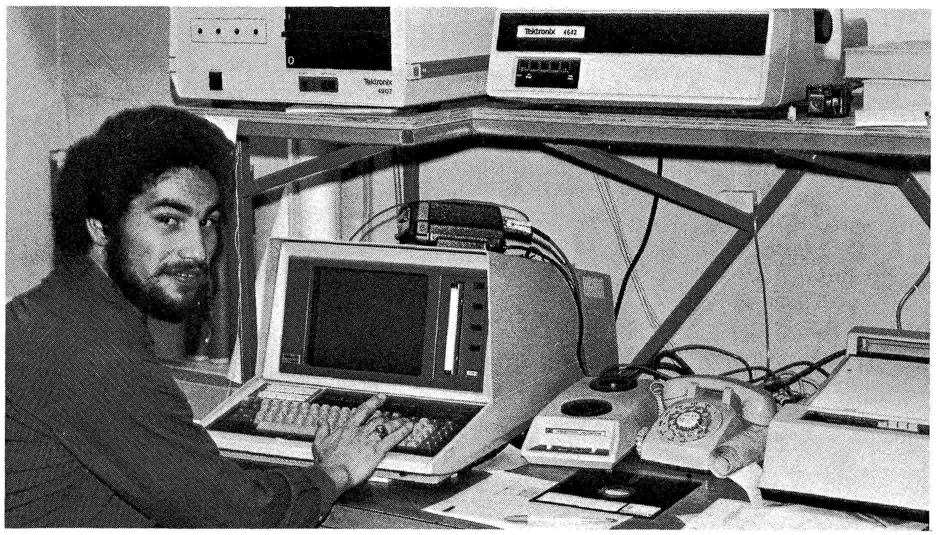


Figure 6. John Vargas explains spectrum analysis applied to seismic waveforms.

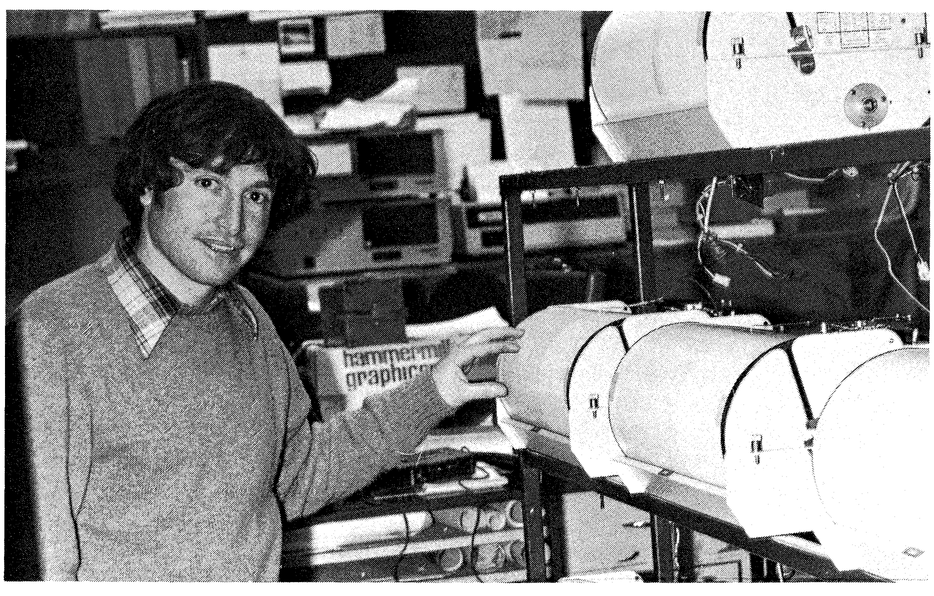


Figure 7. Drums at the Geophysics lab record analog signals picked up and communicated from 11 stations around the State of Kansas. Shlomo Shmuelov checks for traces that would indicate an earthquake.

occur command the attention of the Geophysics group. "We maybe get only a couple of earthquakes per month ranging in magnitude of 1½ to 3 (on the Richter scale)," observed Steeples. "In the four years that we've been recording them, we have had one earthquake above 3, in North Central Kansas, about two years ago. It was about 3.3. The typical magnitude of most earthquakes is down around 2 or 1.8."

According to Steeples, the number of earthquakes increases logarithmically as they decrease in magnitude. For example, for one earthquake of magnitude 3, chances are 10 will occur of magnitude 2 and about 100 of magnitude 1. While this is one of the ways they estimate how often earthquakes of magnitude 5 or 6 will occur, extrapolating the number of earthquakes is only a part of the laboratory's research.

From the data captured on the drums, Steeples and his staff key into the 4052 the receiving station names and arrival times of seismic waves. Processing the data through an iterative least squares matrix inversion, the scientists determine the X, Y, Z location of the earthquake and its origin time. The collected and analyzed data will ultimately be used in designing building codes, nuclear power plants and large dams.

A long term objective for Steeples is to spectrally analyze the earthquake data. Because that requires digital data, they are temporarily stymied for lack of equipment. Once they have the capability to acquire earthquake data digitally, Steeples hopes to learn more about the source of the earthquake — the stress level at the epicenter, the size of the source area, the fault displacement, and so on. He concluded that there is a lot of information to be obtained from a spectrum of an earthquake.

In their pursuit of seismic knowledge, the Kansas Geophysics team is keeping the 4052 Desktop Computer operating at a steady pace. With bushelsful of theories to be tried and tested, it will continue to be a main aid in understanding seismic exploration and earthquake data.

Editor's Note: We thank Dr. Don Steeples, Shlomo Shmuelov, John Vargas, Dr. Ralph Knapp and Gene Taylor for taking their time to provide insight into a unique use of the 4052 and its peripherals. The enthusiasm of Paul Wright, Tektronix Sales Engineer in Kansas City, for Dr. Steeples' application led us to the interview.

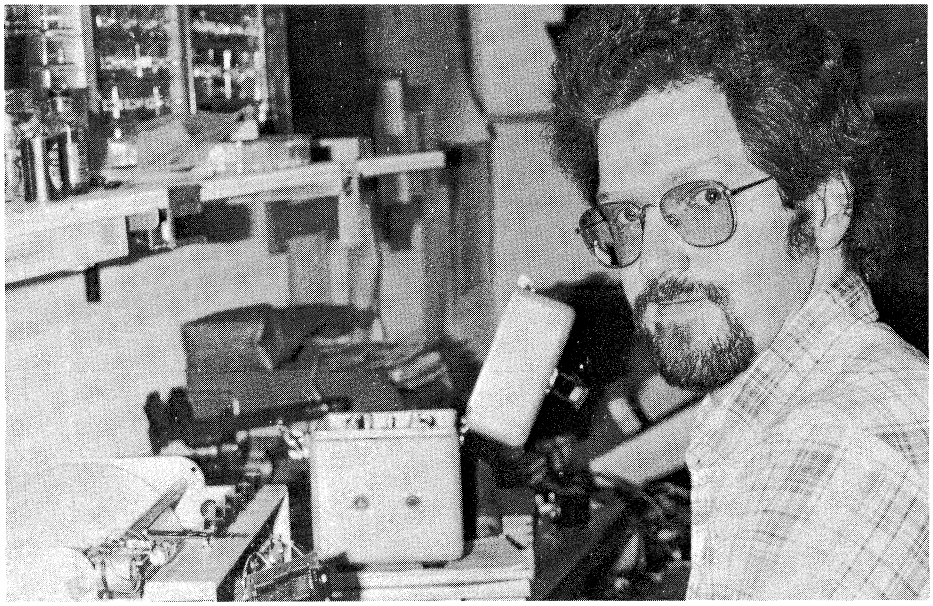


Figure 8. Maintaining the electronic equipment at the 11 earthquake sensing stations, and at the laboratory keeps Gene Taylor, electrical engineer, working at an intense pace.

4052 GPIB Programming Guide Now Available

by Dale Aufrecht
Tektronix, Inc.
Beaverton, OR

The General Purpose Interface Bus (GPIB) can be a smooth path to automated test and measurement, or it can be a rough road strewn with pitfalls. Choosing the right controller and instruments, then writing efficient control programs can make the difference.

Based upon the TEKTRONIX 4052 Graphic Computing System as a controller, the new 4052 GPIB Programming Guide helps you select system components and integrate them. It discusses:

- The 4052's GPIB capabilities
- Choosing system components and configuring the system
- The fundamentals of 4050 BASIC
- GPIB system programming with the 4052
- Techniques for processing and displaying acquired data
- Factors affecting system performance
- Hints for improving system performance

Although the guide is based upon the 4052 Graphic Computing System, the information also applies to the 4054, and most applies to the 4051.

Whether you are a present 4052 user or considering the 4052 as a GPIB system controller, this programming guide should prove useful. For your copy, call your local Tektronix Field Office and ask for the 4052 GPIB Programming Guide, Tektronix part number 062-6400-00.

The Greatest Diameter Among Sets of Digitized Points on the Orbits of the Eyes

by Stephen Kronwith, Ph.D.
Mathematics Department
St. John's University
Jamaica, NY

At the New York University Medical Center's Institute for Reconstructive Plastic Surgery, a massive study of cranio-facial anomalies is in progress. With the help of the Tektronix 4051 Graphic System, X-rays of the skull, taken at various stages of a patient's treatment (preoperative, postoperative, one year, two year, etc.) are digitized and stored on disk. Later the digitized X-ray data are retrieved and either plotted or input to various scientific and statistical programs. Through these programs, the surgeons hope to better understand the effects of their work and thus more effectively plan future operations. Though this work will be treated more thoroughly in a future paper, an interesting example follows.

Figure 1 is an X-ray tracing of a patient with Orbital Hypertelorism. That is, his eye orbits are much too far apart. Through painstaking surgery, the physician corrects this condition by actually breaking the eye sockets away from the skull, moving them forward, then filling in the remaining space with bone grafts. Figure 2 is a postoperative tracing of the same patient. The sockets were moved forward and out; the patient's appearance changed dramatically.

The surgeon must follow the patient's progress through time, and study any regression which might occur. Through careful computer and statistical analysis of the data for dozens of patients with regression, the surgeon hopes to better understand the causes of the regression and compensate for it in future procedures.

To study the regression, the doctors needed to follow a point (or points) along the eye orbits through time and watch their relationship to other fixed (i.e., not involved in the surgery) points in the skull. However, there was no way to pick out the same point along the eye orbits on different digitized X-rays, since no distinct landmarks exist along the orbits as they do in other places in the skull. The trick was, then to associate a scalar value with two points on the eye orbits that did not change over time. Such a value is the maximal diameter along the orbit, and in the majority of cases, surgery does not change this distance.

So, if for each trace we can determine this distance and the set of points which realize it, we know that we have the same set of points on each trace. To determine this distance and the points, we execute the following program on the 4051.

We will assume the (X,Y) pairs are in the scratchlib file TRACE and the disk has been mounted.

Of course, a better algorithm can be gotten by using smaller angles of rotation than one degree.

Thus, using this program, we can follow any movement of the eye orbits through sequential X-rays and can determine if regression is occurring.

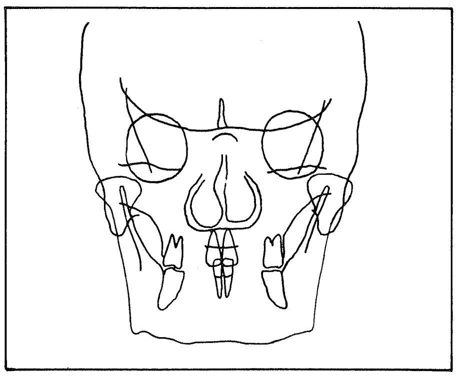


Figure 1. Digitized X-ray of an individual with Orbital Hypertelorism.

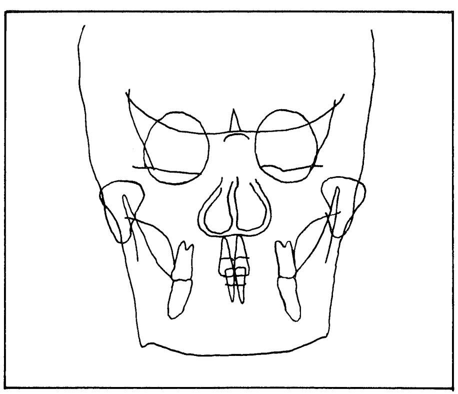
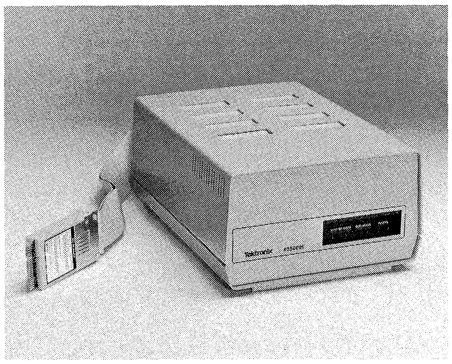


Figure 2. Relocation of the eye sockets through reconstructive plastic surgery is clearly shown in the postoperative digitized X-ray.

```
1 DIM X(1000),Y(1000),X4(360),Y4(360),X5(360),Y5(360),B$(300)
10 DIM T(2,360),R(360)
20 OPEN "TRACE";1,"R",B$
30 ON EOF (1) THEN 90
40 I=1
50 INPUT #1:X(I),Y(I)
60 K=I
70 I=I+1
80 GO TO 50
90 SET DEGREES
100 FOR I=1 TO 360
110      T(1,I)=SIN(I)
120      T(2,I)=COS(I)
130 NEXT I
140 REM  For each angle between 1 and 360 we effectively rotate
150 REM  the orbit through the angle and find those points whose
160 REM  Y coordinates are largest and smallest.
170 FOR I=1 TO 360
180      REM  B represents temporary largest Y coord. S, temporary smallest
190      B=0
200      S=10000
210      FOR J=1 TO K
220           Y1=Y(J)*T(2,I)+X(J)*T(1,I)
230           IF Y1<B THEN 290
240           B=Y1
250           REM  Y9 is temporary largest Y coordinate, X9 its
260           REM  companion X coordinate.
270           Y9=Y(J)
280           X9=X(J)
290           IF Y1>S THEN 350
300           REM Y8 is temporary smallest Y coordinate, X8 its
310           REM companion X coordinate.
320           S=Y1
330           Y8=Y(J)
340           X8=X(J)
350      NEXT J
360      REM (A,C) and (B,D) represent the points whose Y distance largest
370      A=X9
380      B=X8
390      C=Y9
400      D=Y8
410      REM R(I) represents the greatest distance for the Ith angle.
420      R(I)=SQR((A-B)*(A-B)+(C-D)*(C-D))
430      REM (X4(I),Y4(I)) and (X5(I),Y5(I)) are the points which
440      REM realize the largest distance for the Ith angle.
450      X4(I)=X9
460      X5(I)=X8
470      Y4(I)=Y9
480      Y5(I)=Y8
490 NEXT I
500 REM Find the largest distance over all the angles
510 FOR I=1 TO 359
520      IF R(I)<R(I+1) THEN 580
530      R(I+1)=R(I)
540      X4(I+1)=X4(I)
550      X5(I+1)=X5(I)
560      Y4(I+1)=Y4(I)
570      Y5(I+1)=Y5(I)

580 NEXT I
590 X6=X5(360)
600 X7=X4(360)
610 Y6=X5(360)
620 Y7=Y4(360)
630 R1=R(360)
640 PRINT "The largest distance is ";R1;" between (";X6;Y6;") and (";
650 PRINT X7;Y7;")"
660 END
```

4050E01 Offers Backpack Expansion



**by Mark Mehall
Tektronix, Inc.
Wilsonville, OR**

The new 4050E01 ROM Expander adds interfacing capability for the 4051, 4052 or 4054. Each 4050E01 has eight ROM Pack slots available to accommodate either ROM Packs or multiple printer interfaces. The 4050E01 automatically recognizes the type of 4050 series computer to which it is connected. The ROM Packs and printer interfaces used in the Expander must match the computer type. E.g., ROM's built for the 4052 cannot be used in the 4050E01 when it's connected to a 4051. The current ROM Packs and printer interfaces available are:

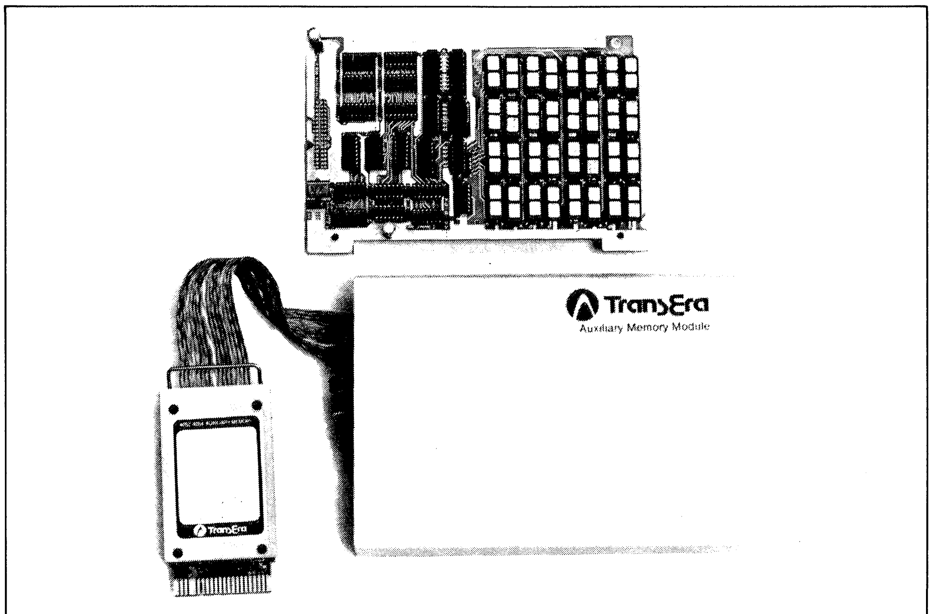
4051

4051R01 Matrix Functions ROM Pack *
4051R05 Binary Program Loader ROM Pack *
4051R06 Editor ROM Pack
4051R07 Signal Processing ROM Pack #1
4051R08 Signal Processing ROM Pack #2 (FFT)
4051R10 4909 File Manager ROM Pack
4051F10 RS-232 Printer Interface
020-0279-00 4907 File Manager ROM Pack
* Functions built-in on new 4051's

4052 and 4054

4052R06 Editor ROM Pack
4052R07 Signal Processing ROM Pack #1
4052R08 Signal Processing ROM Pack #2 (FFT)
4052R09 Real Time Clock ROM Pack
4052R10 4909 File Manager ROM Pack
4052R11 Character and Symbol ROM Pack
4052F10 RS-232 Printer Interface
020-0476-00 4907 File Manager ROM Pack

Fast Auxiliary Memory Augments 4050 System Capabilities



64K–512K Byte Auxiliary Memory Module provides fast external random access memory for program and data storage for the 4050 Systems.

A unique Auxiliary Memory Module complements and extends the 4050 System memory. Available through TransEra Corporation, the module with its memory manager ROM Pack provides from 64K to 512K bytes of random access memory with advanced memory and file handling ability.

Configuration is easy. Simply connect the module to the 4050 through the plug-in ROM Pack.

Data transfer is quick. Reads and writes up to 50K bytes per second are possible for certain operations.

File and memory management is outstanding. Dynamically expanded or specified file size, random or sequential files, numeric files with specified accuracy, file scaling, sorting and plotting, math operations — all are available through the new commands and routines included in the ROM Pack.

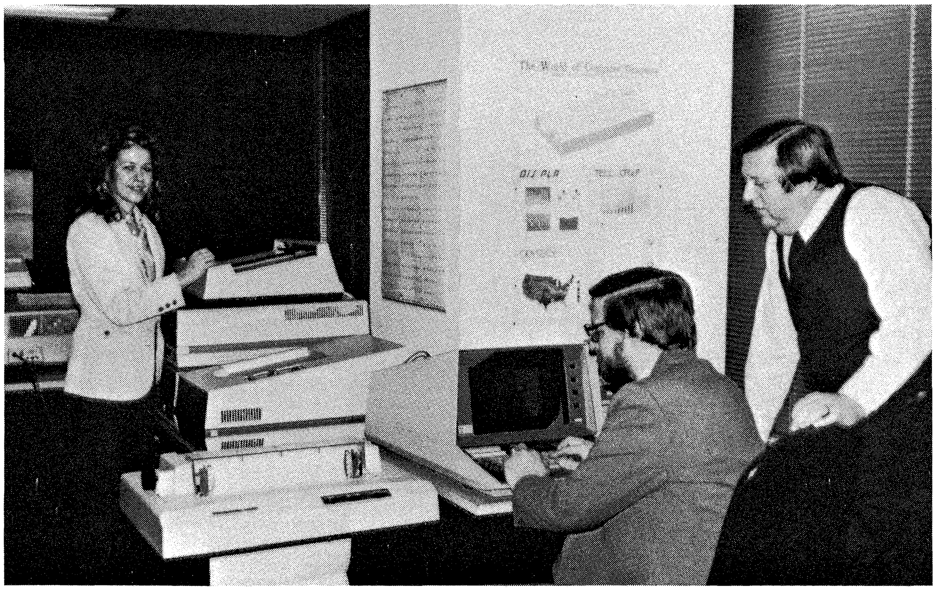
If you are using large name files and need random accessibility to string data, this module may be the answer. Or, if you are producing large drawings which require fast access to symbols, objects and patterns, you may find the module invaluable. For more information on the TransEra Auxiliary Memory Module, contact John Hess at TransEra Corporation, 3707 North Canyon Road, Suite 4, Provo, UT 84604, (801) 224-6550.

Integrating Graphics and MIS at Social Security Region V: A Success Story

by Frederick J. Fachet
Social Security Administration
Chicago, IL

They had arrived, the cartons containing the parts of our new 4050 Series TEKTRONIX Graphics System: a 4051, 4662 Plotter and hard copy unit. The equipment was quickly and easily set up. Then came the hard part — integrating it into our operations.

Region V of the Social Security Administration (SSA) includes the states of Illinois, Indiana, Michigan, Minnesota, Ohio and Wisconsin. For the Regional Commissioner, the Management and Budget, Management Information and Work Measurement Section provides management information systems (MIS) data and analysis for more than 230 local Social Security offices, as well as regional executives and support staff. We also function as a service bureau, assisting other regional SSA staffs with data processing services and support.



Graphics play a major role at the Social Security Administration's Region V office. Fred Fachet, seated, discusses their integration into his Chicago office. Janis McAuliffe and Jack Nowosielski help made it happen.

Before the 4051 arrived, the few charts and graphs we produced were simple and entirely hand-drawn. Now we had a new tool, and had to discover its capabilities, and ours.

Getting Started

We were producing graphs with only a few hours of instruction in using the Tektronix equipment, thanks to the "Data Graphing" package from the Applications Library. We began to create charts to supplement the statistical reports produced by our MIS. One of the first ideas we tried was a regional statistical abstract — a chartbook which graphically illustrated regional operating performance in categories such as claims processing time and quality. This chartbook has since been refined and has become a yearly production, a kind of "annual report" for the region.

Graphics also increased our productivity. We found that graphs could quickly and artfully illustrate trends and problems buried in the raw MIS data, replacing lengthy, written, narrative analyses.

Enhancing Our Graphic Output

As we realized the utility of charts and graphs, we added to our capacity by acquir-

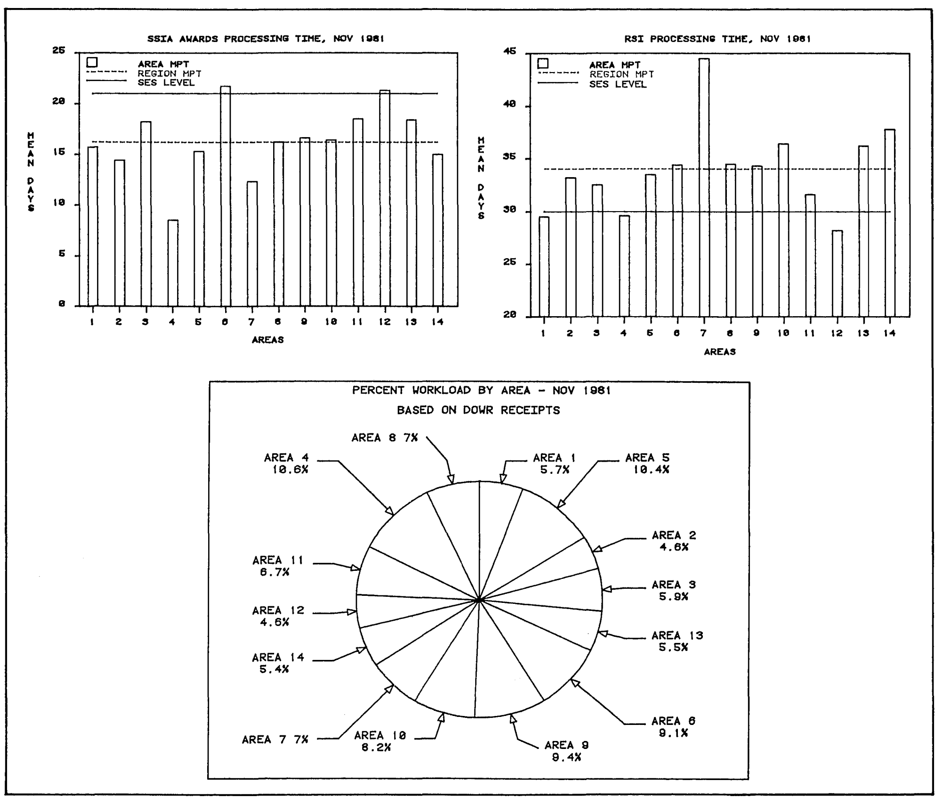


Figure 1. Briefing charts on regional performance for Regional Commissioner of Social Security Administration were prepared using Data Graphing.

ing PLOT 50 Easy Graphing. Easy Graphing enabled us to create more varied charts, with symbol markers and legend control, and to easily store, edit and run repetitive monthly charts.

Furthermore, by connecting the 4051 to our mainframe via the RS-232 Data Communications Interface, we were able to use DISSPLA* to produce even more sophisticated charts and graphs.

We had the best of both worlds: Tektronix software to run the majority of our applications and DISSPLA™ to add greater "polish" for special use charts and graphs.

We spread the word of this new facility among the staffs we service, and we began to get "customers" asking us to produce charts for them.

Since we were producing numerous charts for large audiences (anywhere from 30 to 300 copies of each chart), we quickly found that we could not afford the machine and workhours needed to reproduce charts on the 4662. However, simply Xeroxing the originals lost the color, both by fogging data distinctions showed with color and by losing a lot of the aesthetic quality we had tried so hard to put into each chart. We solved this problem by using a Xerox 6500 Color Copier.

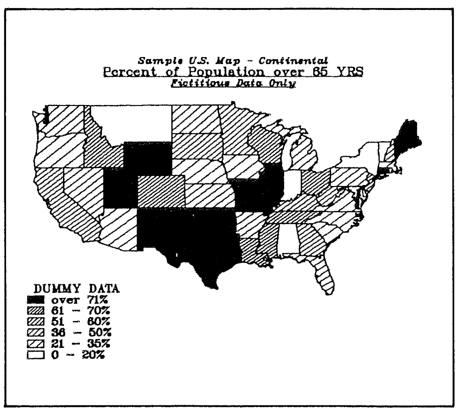


Figure 3. Color maps produced on the 4662 Plotter are reproduced beautifully on the Xerox 6500 Color Copier.

Streamlining Regular Tasks

The 4051 is also a valuable asset to our regular MIS operations. As a full scale desktop computer, we are able to use it to write programs which do inventory for us. We also use the PLOT 50 mathematical and statistical software for analyses. For example, we are doing workload forecasting using PLOT 50 Business Planning and Analysis Vol. 2.

* DISSPLA is the name of graphics software produced by ISSCO of San Diego, CA.

The Tektronix 4051 has enabled us to improve productivity through the use of charts and graphs, to create professional training materials and presentation foils, to provide sophisticated statistics, and to function as a backup for our computer terminals. Even with all of these accomplishments, we believe we are just beginning to realize the full potential of graphics, and this equipment, in our organization.

This article was written by Frederick J. Fachet in his private capacity. No official support or endorsement by the Social Security Administration is intended or should be inferred.

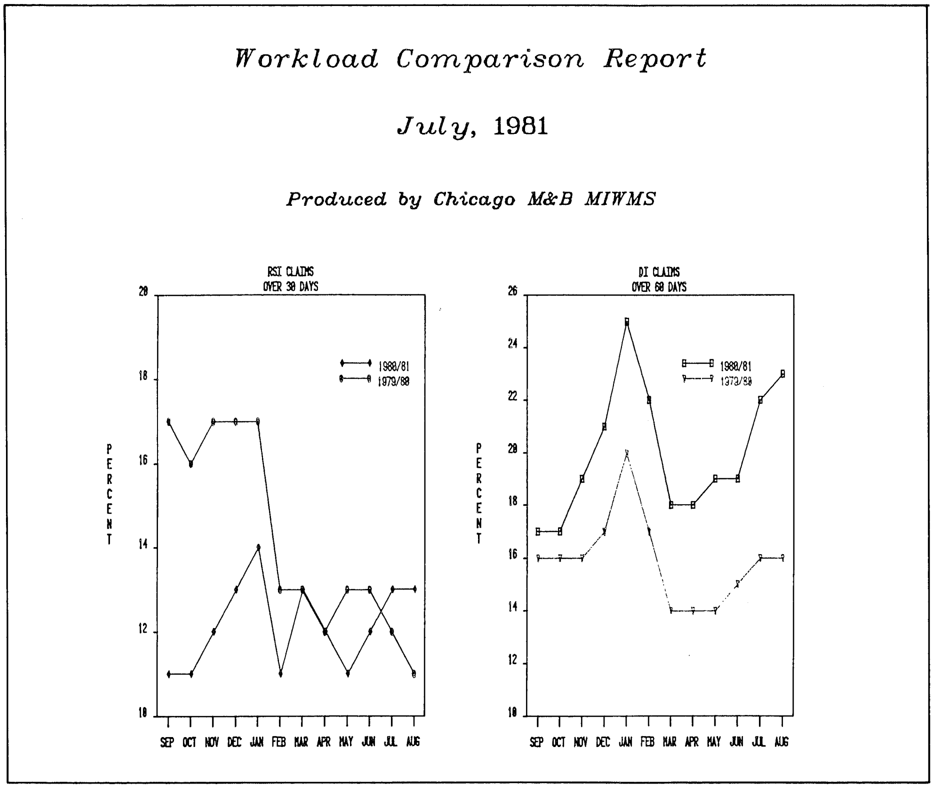
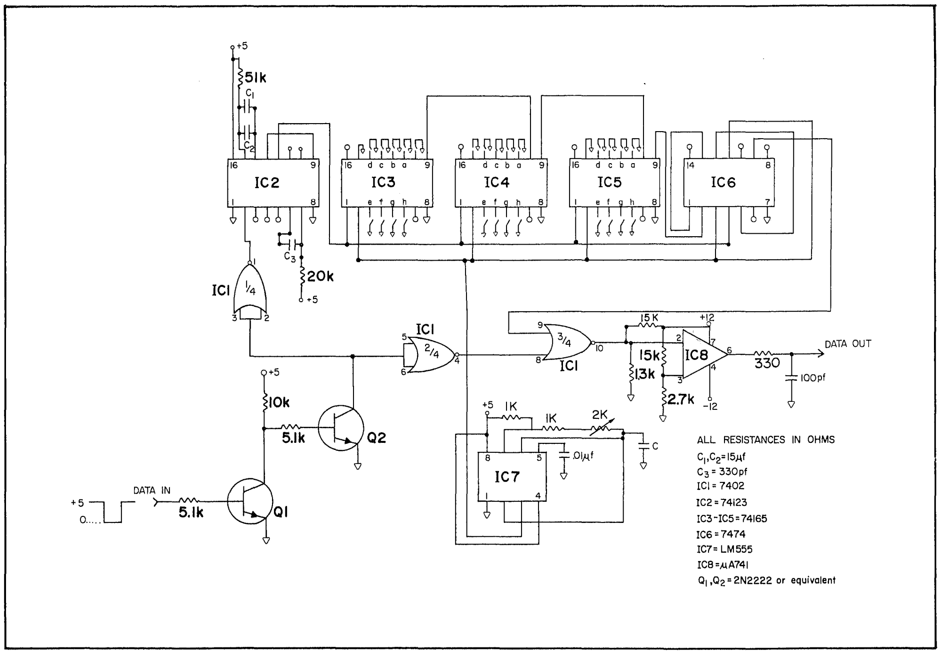


Figure 2. DISSPLA title page precedes graphs created with Easy Graphing.

The entire Region V staff incorporates the 4051 and graphics in their work. Pictured (l to r) are Cindie Kempe, Janis McAuliffe, Lisa Plier, Kathy Rainier, Carlene Sensenbrenner, Mary Loconsole, Carol Howard, Brad Buoy, Jackie Manago, Jack Nowosielski and Terry Belanger.

A Fast Unidirectional TTY Interface for a Minicomputer



Schematic of Interface

by R. Christie Harper[1]
Ben M. Williams[2]
James E. Gaiser[3]
Nuclear Science Center
Auburn University
Alabama

Many commercially available multichannel analyzers (MCA) use a 20 mA current loop to route data to a printer. In general, the output data rate is either 110 or 300 baud since a teletype or similar printer is common in most laboratories. While this is adequate for many purposes, some experiments generate large amounts of data which must be rapidly transferred to a minicomputer. In this case the interface must generate control characters signifying the end of data transfer; it must also be simple. The interface shown in figure 1 satisfies these criteria.

How It Works

Q1 and Q2 convert the 20 mA loop to a TTL compatible signal. When data from the MCA is present at the input, gate 1 of IC1 is used as the reset pulse for a retriggerable monostable multivibrator, ½ of IC2, causing IC2–IC6 to remain inactive. The data is passed through the second gate of IC1 which is wired as an inverter and is ORed through the third gate of IC1, which inverts the data to its original state, since pin 9 of IC1 is low. The data is then input to IC8, a 741 type OP-amp, which is a comparator. This produces an RS-232 compatible level.

Upon completion of data transfer from the MCA, the first half of IC2 triggers the second half of IC2, which is wired as a one shot, generating the load pulse for IC3–IC6. IC3–IC5 are parallel-in serial-out shift registers and IC6 is a dual type D flip-flop. When the load pulse is generated, the first D flip-flop is set, and Q goes high providing the first start bit. IC7 is an LM555 set up as an astable multivibrator. Values for C are chosen for the baud rate used (see Table 1) and resistor R3 adjusts the frequency to the desired value. Following the load pulse, ASCII data programmed on IC3–IC5 is shifted out serially through the D flip-flops.

Programming is provided by using switches mounted in a 16 pin dual inline package. Table 2 shows an example of the switch settings used in this application. Seven data bits, the parity bit and two stop bits are programmed for the first word, and all subsequent characters must have the start bit programmed in addition to the other bits. The schematic shown in Figure 1 is wired to send two ASCII characters; all bits after the final two stop bits are low. This ensures that no characters are sent after the final stop bits. Table 3 shows the wiring diagram for the 25 pin RS-232 connector.

The entire circuit is housed in a single width nim (nuclear instrument module), and obtains all its necessary power from the ±12 volt bin power supply. A series resistor and a 5.1 volt zener diode provide five volt power for the logic chips. Using conventional TTL chips, the circuit draws less than 200 mA of current. Low power schottky TTL chips would decrease the current requirement by at least a factor of two.

The only modification necessary on the MCA is to increase the baud rate for data output to the desired value. The baud rate chosen for the particular interface was 2400, the limit of the RS-232 port on the Tektronix 4051.

How It's Used

In the present application, the interface routes data from an MCA to a Tektronix 4051 Desktop Computer. The internal 300K byte magnetic tape system of the 4051 may be accessed through the tape communication mode of terminal operation. A typical program used to store data on the tape in terminal mode is shown below.

Line 120 sets the baud rate, parity, and an error option of the 4051. The present case sets up 2400 baud, even parity, and ignores errors. Line 130 tells the 4051 the beginning line character (Cntl-J), the end of line character (Cntl-rubout, which is a carriage return-line feed) and the end of data character (Cntl-S) which is determined by the switch setting of the interface. The statement in line 210 (CALL "DTRECV") allows the terminal to alternate between BASIC mode and terminal mode. When the 4051 receives end-of-data character, it returns to BASIC mode which allows manipulation of data, plotting, etc.

The time saved using this interface is significant. To read out a 1024 channel spectrum

Present locations:
[1]General Research Corporation, Huntsville, AL
[2]Sverdrup Technology, Inc., Arnold Air Force Station, TN
[3]East Carolina University, Physics Department, Greenville, NC

at 110 baud to a printer takes approximately 12 minutes; then the spectrum must be manually typed into the computer. By using the interface at 2400 baud this time is cut to 35 seconds and the data is stored on tape in machine readable form.

References

1) *TTL Cookbook,* Don Lancaster, Howard W. Sams and Co., Indianapolis, Indiana, 1974.
2) 4051 Option 1 Data Communications Interface Manual, Tektronix part #021-0188-00.

Table 2

IC5	IC4	IC5
h g f e d c b a	h g f e d c b a	h g f e d c b a
0 1 0 1 1 0 0 0 1	1 1 0 1 1 0 0 1	0 0 1 1 1 1 1 1
s p	s s s	p s s
t CR a	t t t DC3	a t t
a r	o o a	r o o
r i	p p r	i p p
t t	t	t
y		y

0 — ground
1 — +5

Switch settings to send a CR-DC3 combination.

Table 1

BAUD RATE	C (μf)	F* (Hz)
2400	0.1	2400
1200	0.2	1200
600	0.4	600
300	0.8	300

* Pin 3 of IC7 (1/period from oscilloscope)

Value of capacitor C vs Baud Rate

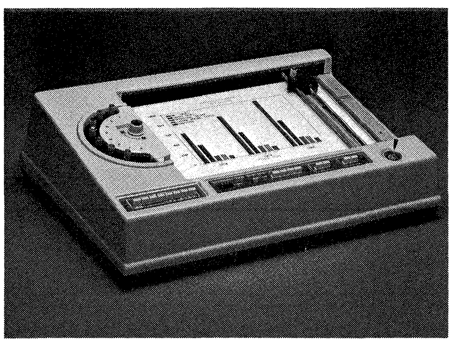
Table 3

PIN NUMBER	CABLE CONNECTION
3	DATA OUT
6, 20	GROUND
4, 5, 8	TIED TOGETHER
11, 12, 19	TIED TOGETHER
ALL OTHERS	NOT USED

Interconnections for 25 pin connector.

```
100 INIT
110 PAGE
120 CALL "RATE",2400,0,2
130 CALL "RSTRIN","J","_","S"
140 PRINT "INPUT THE NUMBER OF FILES TO BE TRANSFERRED";
150 INPUT N
160 PAGE
170 FOR I=1 TO N
180 REM FIND TAPE FILE
190 FIND I
200 REM ENTER TAPE COMMUNICATIONS MODE
210 CALL "DTRECV"
220 PAGE
230 REM PLOTTING OR DATA ANALYSIS ROUTINE CAN GO HERE
240 NEXT I
250 END
```

Eight-pen Option Brings Automatic Pen Changes to the 4662 Plotter



by Craig Montgomery
Tektronix, Inc.
Wilsonville, OR

A rotary pen turret which has eight pen stations may be added to your present 4662 Digital Plotter, or included as an option on your new 4662 Plotter. With the eight-pen turret installed, no longer do you need to manually change the pens for each different color on your plot. You simply insert any eight pens (or seven pens and the digitizing sight) into the turret and program the 4662 to make the selection for you. Should the need arise, however, you may manually select a pen by depressing one of the eight pen-select switches on the front panel.

The new pen turret is completely compatible with all existing 4662 software and hardware configurations. You just include simple pen select commands in your program. All other 4662 commands remain the same. For example, in 4050 BASIC to change pens, you would insert the following line of code before that portion of the program you wish to plot with a new pen:

PRI @d,8: pen number

(d = device address, pen number = 0,1,2...8)

(Note: an argument of 0 allows you to store all pens at the end of a plot.)

Tektronix PLOT 10 software and most PLOT 50 software has been modified to allow you to take full advantage of the added features of the Option 31.

Uses 4663 Pens

Because the eight-pen turret is adapted to use 4663 (our large C-size plotter) pens, you may mix or match fiber tip, wet-ink, or the new plastic hard tip pens. You can vary the pen type to fit your application, and choose the color for greater clarity. And the plotter automatically uncaps and caps all pens.

Brings Added Features

Additional enhancements brought to the 4662 Plotter by the eight-pen option include setting the plotter speed from 10 to 560 mm/sec by 10 mm/sec increments through host software commands. (You don't need to change the switches on the back to obtain the desired speed.)

Setting a switch allows DC1/DC3 flagging from the 4662 to the host over the RS-232 communications line to prevent buffer overflow.

Depressing the PAUSE switch during a plot routine will stop the plotter. You may move the pen holder out of the way with the joystick to view your plot. Upon pressing the RESUME switch, the plotter will return the pen to its position at the time of the pause and continue the plot without loss of data.

To retrofit your present 4662 Plotter or order a new 4662 Plotter with the eight-pen option, contact your local Tektronix Sales Engineer and ask about the 4662 Option 31.

4050 Controls New Generation of Programmable Instruments for Measurement Automation

When the Tektronix 4051 Desktop Computer emerged upon the computing scene back in 1975, it quickly earned a reputation for its friendliness as well as its versatility. Why? It incorporated the concepts that general purpose machines should employ friendly microprocessor-based intelligence, and should employ a standard interface (IEEE-488 GPIB). These proved invaluable in reducing program development time and overcoming interfacing obstacles.

Tektronix has taken another big step in this direction with their new family of TM 5000 programmable test and measurement instruments, an instrument concept for automation ease. Coupled with a 4050 system as their controller, the result is a total systems approach to compatibility, capability and ease of use.

While this article won't go into great detail on the complete capabilities of each instrument , we will take a brief look at each member's function and their unique integration with the 4050 Desktop Computer.

TM 5000 is the nomenclature for a new, broad-based line of IEEE-488 compatible test and measurement instruments from Tektronix, Inc. It's a line of programmable instruments designed specifically to give you —

- capability in the R&D lab,
- flexibility on the designer's bench,
- standardization on the manufacturing floor
- and programming ease for productivity in any atmosphere.

How is all of this put into a single instrument line?

Begin with modularity for configurability

The TM 5000 concept begins with a broad base of instrument types. Take the most commonly needed ones to start the base — a power supply, a digital multimeter, a universal counter/timer, and a function generator. Then add to the base some signal handling units — a high-frequency (to 350 MHz) software configurable switching matrix and a multifunction switching/control unit for

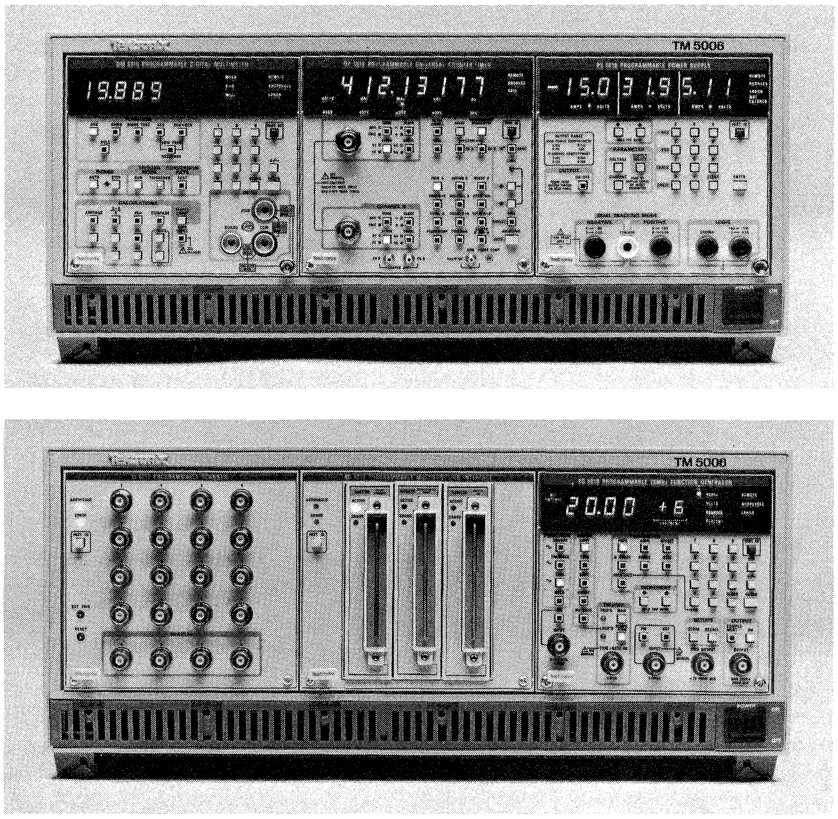


Figure 1. The TM 5000 concept makes instruments portable, stackable, and rackable.

interfacing to relay drivers, foot switches, steppers, or any other apparatus. Package everything in a standard-size module, a size that is a submultiple of the standard industrial instrument rack width. Then build power modules that two, three, or more of the instruments will plug into.

The result is the neat, compact instrument package shown in figure 1. It's portable and rugged for field service needs. It's a space saver on the designer's bench. It's easily rackmountable for instrument van, shipboard, or production floor use. And, because the instruments plug into the power module, you don't have to unstack or unrack to change instruments. Just plug in the instrument configuration you want.

Add IEEE-488 compatibility and programmability

Each TM 5000 instrument contains an interface that conforms to IEEE Standard 488-1978. An IEEE-488 bus, more commonly referred to as the General Purpose Interface Bus or just GPIB, extends across the back plane of the TM 5000 power module and goes to a common GPIB (IEEE-488) connector. This back-plane bus saves instrument cabling time and confusion — just plug the TM 5000 instruments in and they are connected to the power module's single GPIB cable.

Even though they are usually connected to an instrument controller, TM 5000 instruments can still be operated manually. You can manually select voltage levels, frequencies, measurement functions, etc. from the instrument front panel.

But you can also set up and operate all the instrument functions under program control through each instrument's GPIB interface. For example, you can set the FG 5010 function generator for a 2-volt peak-to-peak 3-KHz sinusoidal output by pressing the following front-panel button sequence: FREQ, 3, 0, 0, 0, ENTER, AMP, 2,

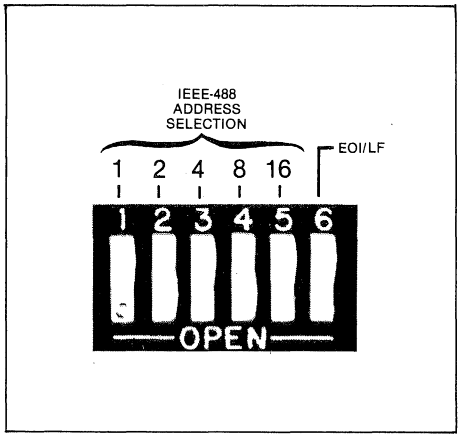


Figure 2. Switchable addressing and message termination is just one of many conveniences that enhance system configurability and compatibility.

ENTER. Or you can just send it the following message sequence over the bus: FREQ 3000;AMP 2. The internal microprocessor takes care of interpreting the messages and setting up the instrument.

Each instrument's microprocessor takes care of a lot of other things for you too. Whenever an instrument is powered up, the microprocessor runs diagnostics to check general instrument operation. Whenever settings are entered, either manually or under program control, the microprocessor checks them to make sure they are valid combinations and in-range. If they aren't, an error code is generated for use over the bus. Plus, the microprocessor assesses and stores instrument status for a variety of operations and conditions. You can use this status information in your programs to monitor or change the direction of measurement sequences.

The internal microprocessor in each instrument also offers the opportunity for some additional measurement features. For example, the DC 5010 Programmable Universal Counter/Timer can make rise-time measurements as well as the standard counter/timer measurements. And, as another example, the DM 5010 Programmable Digital Multimeter can make several calculations, including decibel conversions, from measurements or entered constants.

Make it easy to program

Realizing that the key to productivity is still people, TM 5000 instruments are designed for easy use by people. The front-panel controls are laid out in logical groupings. Each control is labeled with obvious mnemonics describing its function. There are no obscure or specialized symbols.

The same approach is taken in the programming messages for each instrument — no obscure or specialized code. The messages are descriptive abbreviations of the front-panel labels and instrument functions (figure 3). For example, to set the DC 5010 to measure the frequency of the signal at Channel A, just push the FREQ A button or send it the message FREQ A over the GPIB. It's just that simple. The instrument commands are designed for the convenience of people, not microprocessors.

Take SET? for example. To ask any TM 5000 instrument what its current control settings are, just send it SET? over the bus. The instrument responds by assembling a message string describing the instrument's current setup, including several internal conditions. This settings message can be stored by program in a single string variable and used later to duplicate the test setup under program control. Dozens of setups can be stored and executed as needed.

The 4050 as the controller

Since TM 5000 instruments are GPIB compatible, they can be cabled up to any GPIB instrument controller. Switches on each TM 5000 instrument let you set the instrument to the message terminator required by a particular controller. So GPIB compatibility is maintained, no matter what your controller choice might be.

However, the greatest degree of compatibility — and programming ease — is achieved when TM 5000 instruments are interfaced to Tektronix supplied controllers. That's because Tektronix controllers go beyond mere GPIB compatibility. They are optimized for instrument control.

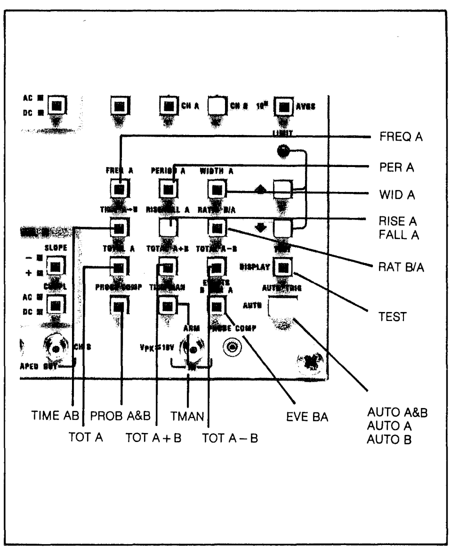


Figure 3. Section of DC 5010 Programmable Universal Counter/Timer front panel with associated control messages indicated — the direct relationship between instrument functions and instrument control messages makes TM 5000 programming natural and easy.

While the 4050 systems are well known for their capabilities as desktop graphic computers, not so well known are their capabilities as IEEE-488 (GPIB) instrument controllers. 4050 BASIC incorporates a flexible I/O structure that allows simple addressing of GPIB instruments and peripherals. It also includes extensions for signal processing.

Marrying the 4050 capabilities with the "engineering English" commands common to the TM 5000 instruments results in reduced time from shipping cartons to an operating system. And the graphics allow you to go beyond simply using the 4050 to direct the instruments.

Often graphic portrayal of measurements results in greater understanding and, thus, more accurate decisions. By displaying expected results, an operator can compare actual results against a standard.

Graphically instructing an inexperienced operator could also facilitate testing, (figure 4).

Simplifying the software by establishing a common language for TM 5000 instruments results in products that are easy to operate. But capability has not been sacrificed, and the instruments incorporate state-of-the-art technical advances. Your local Tektronix Sales Engineer will be happy to provide you with more details.

This article was reprinted from HANDSHAKE, a newsletter published quarterly by Tektronix as a forum for people interested in programmable instrumentation and digital signal processing.

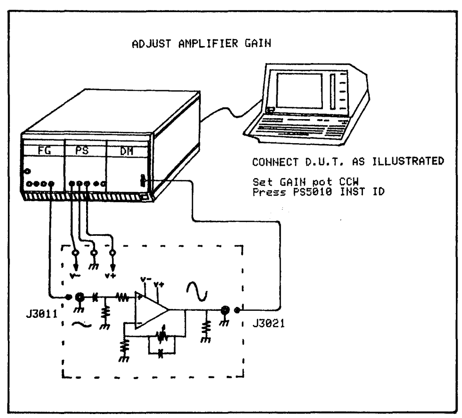


Figure 4. The 4050 displays step-by-step instructions, including graphics, for a test, using the TM 5000 family Function Generator, Power Supply and Digital Multimeter.

TM 5000 Family of Programmable Instruments

The initial offering in Tektronix' new generation of programmable instruments consists of 10 modules.

Mainframes

TM 5003 – Three-compartment

TM 5006 – Six-compartment

- Provides the power supply, transformer and interconnections for the modules

Measurement Set

DM 5010 Programmable Digital Multimeter

- DC Volts, .015% + 1 Count
- Ohms, .015% + 2 Count
- True RMS (AC + DC)
- Easy Calibration
- Fast-Slow Mode
- μp Nulling
- Averaging
- Offset & Scaling
- dB Conversion
- Hi-Low-Pass Mode
- External Guard
- 10^9 Ω Input Z
- Keyboard Entry
- Diode Test
- In-circuit Ohms Measurement
- Autoranging
- Internal Self-check

DC 5009 Programmable Universal Counter/Timer

- 10 nsec Clock
- DC to 135 MHz
- Auto-trigger
- Ratio Architecture
- Probe Compensation
- 10 Functions
- Auto Averaging
- Trigger Level Outputs
- Shaped Outputs
- Arming Inputs
- Single Channel Width Measurements
- TM 500 Rear Interface Inputs and Outputs
- 8 Digits

DC 5010 Programmable Universal Counter/Timer

- 3.125 ns Clock
- DC to 350 MHz
- Auto-trigger
- Ratio Architecture
- 13 Functions Including Risetime Mode
- Null Function
- Probe Compensation
- Auto-Averaging
- Shaped Outputs
- Arming Input
- Single Channel Width and Risetime Measurements
- 9 Digits

Signal Source

FG 5010 Programmable 20 MHz Function Generator

- .002 Hz to 20 MHz
- Sine, Square, Triangle
- 20 mV to 20 V p-p
- N-burst
- Programmable Symmetry
- Auto Scan Phase Lock
- Phase Lock with Programmable Phase
- AM, FM, VCF
- 10 Stored Setups
- Output Complement
- Waveform Hold
- Haverfunction

Power Supply

PS 5010 Programmable Power Supply

- Triple Output 0 to −32, 0 to +32, and 4.5 to 5.5 Volts
- Both Voltage and Current Programmable
- Auto-crossover with Bus Interrupt on CV, CI Mode Change
- Triple Displays, V or I, and CV, CI Indication
- Dual Floating Supply Trackable
  - 0.75 Amps to 32 V
  - 1.6 Amps to 15 V
  - 50 mA Current Steps
  - 10 mV Steps to 10 V
  - 100 mV Steps to 32 V
- Logic Supply
  - 4.5 to 5.5 V in 10 mV Steps
  - 100 mA Current Steps to 3A
- Front/Rear Outputs
- Remote Sense

Interfaces

SI 5010 RF Programmable Scanner I/F

- 350 MHz Coaxial Switching
- Software Reconfigurable as:
  - 16 Channel to 1
  - dual 8 Channel to 1
  - quad 4 Channel to 1
- Buffered Mode for Controller-Unattended Operation
- Real time Clock
- Triggered Events

MI 5010 Programmable Multifunction I/F

- Four plug-in cards
  - 16 Point Relay Scanner
  - 16-bit Digital I/O
  - Programmable Development Card
  - 12 – Bit D/A
- Signal Routing
- Control Functions
- Device Interface
- Programmable V/I Source
- Real time Clock
- Triggered Events
- Buffered Mode for Controller-Unattended Operation
- Programmable handshake

MX 5010 Multifunction I/F Extender

- Attaches to MI 5010 to extend to six cards

Guide Helps Implement 4052 as Controller for TM 5000 Instruments

The GPIB Programming Guide (part #070-3985-00) aids the user of 4050 Desktop Computers and TEKTRONIX TM 5000 series instruments in making the software connection.

Major topics with coded examples where applicable are:

- 4050 Desktop Computer controller capabilities
- GPIB Input/Output
- Interrupt Handling
- Interrupt handling statements
- Utility routines
- 4052/GPIB send and receive program

TEKTRONIX TM 5000 series instruments specifically covered in the Guide include:

DC 5009 and DC 5010 Programmable Universal Counter/Timers

DM 5010 Programmable Digital Multimeter

FG 5010 Programmable Function Generator

PS 5010 Programmable Power Supply

The Guide includes source code for 47 routines. Machine readable copies of the routines are included in the 4050 Applications Library as part of TEKniques Vol. 5 No. 4 T1 tape, part #062-5981-01.

* Editor's Note

Programming Tips Handbook

The programming tips from the first three years have been collected into a handy booklet which is included in the Programming Aids T2 tape documentation (part #062-5972-00).

4050 Applications Library Ordering Procedure

Software from the Applications Library may be ordered through your local Tektronix field office or from the Tektronix Central Parts Ordering offices. See the **New Abstracts** section for further information.

Back Issues and Reprints from TEKniques

TEKniques is in its sixth year of publication. Issues from the first three years (Volumes 1–3) have all been distributed. However, most of the articles from those issues have been assembled by application area and are available in the following reprints:

Engineering and Design......... AX-4449
Mapping...................... AX-4460
Data Acquisition and
  Analysis.................... AX-4450
Business Graphing and
  Reporting.................. AX-4451
Peripherals and ROM Packs..... AX-4452

If you need an article from one of these previous volumes, and don't have your copy, one of the reprint sets will likely fill your needs. To obtain a copy of one of the reprint volumes, just contact your local Tektronix Office or the Applications Library Office serving you.

And, of course, back issues of TEKniques Vol. 4 (1980) and Vol. 5 (1981) will continue to be available from the 4050 Series Applications Library office that serves your area.

TEKniques Printed Quarterly

From our first issue of four pages to our largest of 40 pages, TEKniques' size jumped tenfold. To accommodate this increase, our schedule began slipping until last year when we published just four issues. But they were big ones.

In retrospect, the four larger issues per year seem to fit 4050 users' needs just as well or better than the previous eight. Therefore, we are maintaining that schedule.

TEKniques Spring 1982 issue begins the year brimming with applications, ideas and programming tips. We'll be keeping you up to date with the 4050 Desktop Computer Series and its peripherals in forthcoming Summer, Fall and Winter issues.

Input/ Output

4907 File Manager Disk Problems

An inquiry from H.H. Berges, The Upjohn Company, LaPorte, TX, concerning recovery of files from disks he could not "MOUNT" on his 4907 File Manager, elicited the following observations from Steve Duncan, Technical Support Specialist, Tektronix, Wilsonville.

Your 4907 File Manager may need calibration periodically, especially if it is being moved from station to station. This is performed by a Tektronix Field Service Specialist, who you may contact through your local Tektronix Field Office.

The 4907 File Manager read/write head needs to be cleaned annually, or as necessary, depending upon the environment and amount of use.

It is also extremely important that backup disks be kept since the disks do have a finite life. The case and life of magnetic media will be covered in an upcoming issue of TEKniques.

Programming Tips

Default Response-Revisited

by Bryan Burma
Tektronix, Inc.
Kansas City, KS

Bernard Taieb, in a programming tip (Vol. 2 No. 6) discusses the use of the RETURN as a default response when branching. There are times, however, when a default response may not be desired. To avoid a default response to the first line number referenced in the GOTO P OF 100,200 . . . just insert a non-valid response character such as space as the first character of the (string-to-be-searched).

This process of error trapping is simpler and saves memory space since a conditional does not have to be set to trap the RETURN.

```
200 PRINT "Input A, B, or C:";
210 INPUT Q$
220 P=POS(" ABC",Q$,1)
230 GO TO P OF 200,300,400,500
240 GO TO 200
250 END
300 REM A Processing
400 REM B Processing
500 REM C Processing
```

Driving Diablo 630 Printer in HYPLOT from 4050

by Joel T. Hicks, P. E.
General Technology
Little Rock, AR

The following describes a technique for converting 4050 user defined plot coordinates to desired surface coordinates of the Diablo Model 630 printer, and for creating a character string to transmit these coordinates from the 4050 to the Diablo. (The Tektronix 4012 terminal* also requires a similar alpha string for plotting.)

Although the specific coding was developed for the 4051, the program steps can be used with any machine that will create a character according to its ASCII value. The mathematics of the technique transcend programming language barriers.

To understand the work, you need to understand the following definitions (and example values):

Variable/Subscript /Meaning	1 Xmin	2 Xmax	3 Ymin	4 Ymax
C9 = Diablo Coordinates =	204	804	184	384
C8 = 4050 User Coordinates =	0	200	0	1200

FNC = 5 most significant bits of X or Y
FND = 5 intermediate bits of X or Y
FNE = 2 least significant bits of X or Y
FNR = Rounding function for all numbers

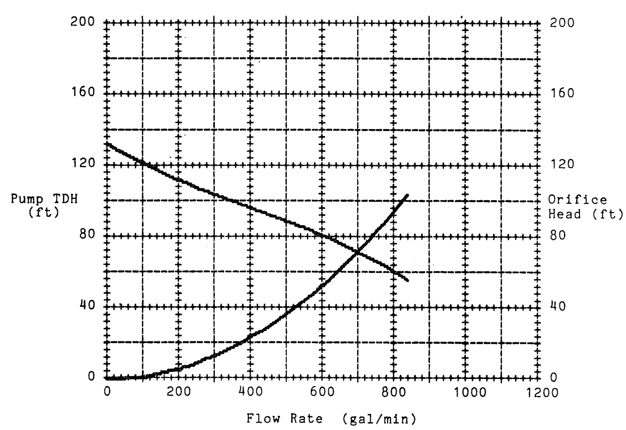
Statement 2260 turns Ymin and Ymax upside down to accommodate the Diablo origin.

Statement 2200 limits the least significant segment of X and Y to a value between 0 and 3 (which may not be required for some machines).

Statement 2320 insures that "RUB OUT" is not sent to the Diablo, as instructed in the Operator's Manual.

The example grid was printed by drawing vectors after setting precision to h = 10 and v = 5. Note that there are only a few discrete plot scales that allow "neat," fast vector draw grids. (The graph is an example produced by a large program, of which the coding is only a small but essential part.)

```
2180 DEF FNC(X)=INT(X/128)
2190 DEF FND(X)=INT((X-INT(X/128)*128)/4)
2200 DEF FNE(X)=INT(X-INT(X/128)*128-FND(X)*4) MIN 3 MAX 0
2210 DEF FNR(X)=INT(X)+(X-INT(X)=>0.5)
2230 RETURN
2240 REM CONVERT X,Y(USER) TO DIABLO ******************************
2250 X=FNR((C9(2)-C9(1))*(X-C8(1))/(C8(2)-C8(1))+C9(1))
2260 Y=FNR(C9(4)-(C9(4)-C9(3))*(Y-C8(3))/(C8(4)-C8(3)))
2270 REM CONVERT X,Y(DIABLO) TO A$ **********************
2280 A$=CHR(FNC(Y)+32)
2290 B$=CHR(96+4*FNE(Y)+FNE(X))
2300 A$=A$&B$
2310 B$=CHR(FND(Y)+96)
2320 IF FND(Y)<>31 THEN 2340
2330 B$="[Z"
2340 A$=A$&B$
2350 B$=CHR(FNC(X)+32)
2360 A$=A$&B$
2370 B$=CHR(FND(X)+64)
2380 A$=A$&B$
2390 RETURN
```



* The methodology for the 4010 Series Terminals conversion was explained in TEKniques Vol. 3 No. 4 programming tip "4051 Drives Plotter Through RS-232."

Fast String Sorting with the SPS ROM #1

by Peter Kellenberger
Tektronix, Inc.
Zug, Switzerland

PROBLEM:

A number of strings (in the example up to 2000) each stored in a record on a magnetic tape, shall be printed in an alphabetical sequence.

SOLUTION:

The program goes sequentially through all existing files and records, thereby creating numbers which are stored in array A and "addresses" which are stored in U$. The numbers represent the "weight" of a string and the addresses simply consist of file # and record #. The goal of this procedure is to use the "MIN" function (part of the Signal processing ROM #1) for quick sorting.

Sorting will be based on 37 characters: 26 alpha characters + 10 numericals with 1 character for all the remaining signs. Therefore, a number system with base 37 is used. This results in a "sorting resolution" of 9 characters ($37^9 < 10^{15}$) which was more than adequate for the task in consideration. The addresses are converted to ASCII characters in order to have constantly two bytes for every address. Array T translates the 95 possible ASCII-characters into the base 37 numeric system.

The time requirements in a 4052 are 13 min/2000 strings (encoding + sorting) plus tape movements. This time depends almost linearly on the number of strings.

Note: Strings with a length of less than 9 characters shall be filled with trailing blanks prior to storing or prior to sorting.

Array T Content

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
2	3	4	5
6	7	8	9
10	11	1	1
1	1	1	1
1	12	13	14
15	16	17	18
19	20	21	22
23	24	25	26
27	28	29	30
31	32	33	34
35	36	37	1
1	1	1	1
1	12	13	14
15	16	17	18
19	20	21	22
23	24	25	26
27	28	29	30
31	32	33	34
35	36	37	1
1	1	1	

```
100 INIT
110 FUZZ 15
120 DIM A(2000),T(95),U$(4000),B$(2)
130 FIND 2
140 READ @33:T
150 A=1.0E+15
160 U$=""
170 I=0
180 REM N1=LOWEST FILE#; N2=HIGHEST FILE#
190 REM N3=MAX NUMBER OF RECORDS PER FILE
200 REM READ RECORDS AND STORE INFO INTO A AND U$
210 FOR I1=N1 TO N2
220 FIND I1
230 FOR I2=1 TO N3
240 IF TYP(0)=0 THEN 390
250 INPUT @33:A$
260 IF A$="" THEN 400
270 I=I+1
280 A(I)=0
290 FOR I3=8 TO 0 STEP -1
300 S=ASC(A$)-31
310 A(I)=A(I)+37↑I3*T(S)
320 A$=REP("",1,1)
330 NEXT I3
340 A$=CHR(I1)
350 B$=CHR(I2)
360 A$=A$&B$
370 U$=REP(A$,2*I-1,0)
380 GO TO 410
390 I1=N2
400 I2=N3
410 NEXT I2
420 NEXT I1
430 REM SORT AND PRINT STRINGS
440 FOR I3=1 TO I
450 CALL "MIN",A,X,S   GET MIN #
460 A(S)=1.0E+15
470 S=2*S-1
480 A$=SEG(U$,S,2)    LOCATE FILE & RECORD # S
490 I1=ASC(A$)
500 A$=REP("",1,1)
510 I2=ASC(A$)
520 FIND I1           FIND FILE
530 FOR I4=1 TO I2    RUN THRU TO RECORD
540 INPUT @33:A$      GET STRING
550 NEXT I4
560 PRINT A$          AND PRINT
570 NEXT I3
580 END
```

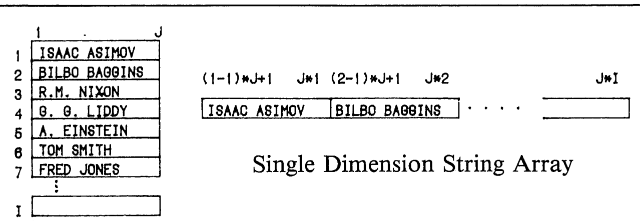
Simulating String Arrays

by Ron Boerger
Dept. of the Air Force
San Antonio, TX

In the March-April, 1981 issue of *Tekniques,* Deedie Strandridge suggests a method for storing string arrays. While the method works, it requires 8 bytes of storage for each character of the array to be stored. This means that, at most, about 200 items of 20 characters each could be stored in 32K bytes of free memory.

If you need to store a character array, why not use the 4050's string handling functions? Although 4050 BASIC does not allow for multidimensional string arrays, we can simulate these arrays via use of the SEG function, and a little programming.

To store an I row by J column array, we merely dimension a string variable to be I*J where I = the number of names, and J = the maximum number of characters per name. We then store the array items sequentially in the string:



Two Dimensional String Array

Single Dimension String Array

A program which illustrates this method follows. Since we are using a string, each character takes only 1 byte, not 8; thus, we can store approximately 1600 items of 20 characters each in 32K bytes of free memory. If you need to store a large number of string items or even if you don't, I recommend usage of this method.

```
100 REM ************************************************
110 REM *** STORING A TWO-DIMENSIONAL CHARACTER ARRAY ***
120 REM *** BY:  RON BOERGER  SARPMA/ACD  03 JUN 1981 ***
130 REM ************************************************
140 REM ***
150 REM *** DESCRIPTION OF VARIABLES:
160 REM ***
170 REM ***   I - NUMBER OF ROWS (ITEMS) IN CHARACTER ARRAY
180 REM ***   J - NUMBER OF COLUMNS (CHARACTERS) IN EACH ROW (ITEM)
190 REM ***  B$ - BLANK STRING USED TO EXTEND USER STRING ENTRIES
200 REM ***  A$ - STRING ARRAY(I*J).  USED TO STORE STRING ELEMENTS.
210 REM ***  T$ - STRING ARRAY(2*J).  USED TO STORE USER ENTRIES.
220 REM ***   X - LOOP VARIABLE.
230 REM ***
240 I=10
250 J=20
260 REM *** ARRAY OF TEN ELEMENTS, EACH 20 CHARACTERS LONG.
270 B$="
280 B$=B$&"
290 REM *** B$ IS A STRING OF BLANKS WHICH IS 72 CHARACTERS LONG.
300 B$=SEG(B$,1,J)
310 REM *** THE MAXIMUM VALUE FOR J IS 72, SINCE WE CAN'T (EASILY) ENTER
320 REM *** STRINGS OF LENGTH GREATER THAN 72.
330 DELETE A$,T$
340 DIM A$(I*J),T$(2*J)
350 A$=""
360 REM *** NOW, GET ITEMS, ONE AT A TIME.
370 FOR X=1 TO I
380 PRINT "ENTER ITEM #";X;": ";
390 INPUT T$
400 REM *** NOW, ADD BLANKS TO END OF T$, AND GET FIRST "J" CHARACTERS
410 REM *** OF THIS RESULT; THEN, APPEND TO A$.
420 T$=T$&B$
430 T$=SEG(T$,1,J)
440 A$=A$&T$
450 NEXT X
460 REM *** READ CHARACTER STRINGS FROM ARRAY AND PRINT
470 FOR X=1 TO I
480 T$=SEG(A$,(X-1)*J+1,J)
490 PRINT USING 500:X,T$
500 IMAGE "ITEM #",3D,":  ",72A
510 NEXT X
520 END
```

Storing Strings in Arrays — The Cheap Way

by Todd Paulus
Tektronix, Inc.
Beaverton, OR

In Vol. 5, No. 2 of "Tekniques," a tip told how to store strings in arrays by storing each ASCII representation of a character in one element of an array. Here's a faster way to accomplish the same task and reduce the array space required at the same time.

Essentially, any 6 (maximum) character string can be converted to an exponential representation and stored in one array element. If the desired maximum string length is expected, or intended, to be longer than 6, then use a FOR loop, SEGment the original string into consecutive 6-character groups, run the program for each 6-character segment, and store the results in consecutive columns of an array-row.

The following is the program to convert from a string to a number, assuming a string length of 6 (including spaces).

```
100 REM---CONVERT A CHARACTER STRING TO A NO.
110 INIT
120 PRINT "MAXIMUM LENGTH OF INPUT = 6 CHAR'S"
130 INPUT I$
140 N=0
150 FOR I=1 TO 6
160 P$=SEG(I$,I,1)
170 IF P$="" THEN 200
180 N=128*N+ASC(P$)
190 GO TO 210
200 N=128*N+32
210 NEXT I
220 REM---N IS YOUR CONVERTED NO.-STORE IT IN YOUR ARRAY
```

Now, to convert back to your original string, pull your number out of the array and run the program below.

```
500 REM---CONVERT THE NO. TO A STRING
510 C=1/128↑5
520 I$=""
530 T=C*N
540 FOR I=1 TO 6
550 T1=INT(T)
560 A$=CHR(T1)
570 I$=I$&A$
580 T=128*(T-T1)
590 NEXT I
600 REM---I$ IS YOUR STRING
```

Obviously, if your original string was longer than 6 characters, use a FOR loop again for each consecutive number in an array-row, and concatenate the results.

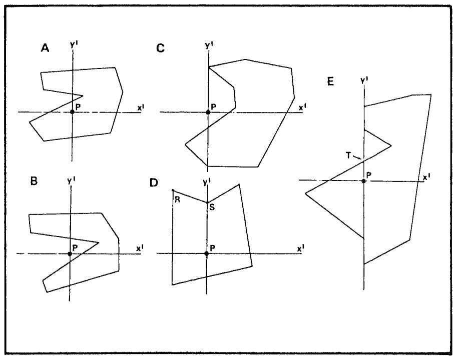
You will be surprised at the speed of the conversions, and you use less array space, too.

Polygons — Do Points Lie Within

by Ir. A.C. Visser
Institute for Land and Water Management
Wageningeu, The Netherlands

When determining whether a digitized point lies within or without a digitized polygon, use a strategy of J.D. Jacobsen, published in 1968 ("Geometric relationships for retrieval of geographic information." *IBM Syst J Nos 3 & 4*). This strategy is to shift the X- and Y-axis so that the origin lies on the point we have chosen. Then, look at the shifted positive Y-axis. If this axis intersects an odd number of polygon sides, the point is inside the polygon. When the line intersects an even number of polygon sides, the point is outside the polygon.

Problems arise when (see the figures C, E) the X-coordinate of point P is the same as the X-coordinate of one or more points of the polygon. It is not necessary to calculate the Y′-values. While calculating the X′-values (X5 in the program) the computer recognizes the problem and solves it by adding a very little value on the coordinate in X′-direction (less than the accuracy of the coordinates). Then the situation is normal again. If coordinates X′ of two following points are both negative or both positive (statement 910), there is no need to calculate the intersection. When the intersection value Y′ is positive it is counted. Statement 950 decides whether the number of positive intersections is odd or even.



Calculation whether a point P is inside or outside a polygon. When X′ = 0 is changed to a very little value, the number of intersections with the positive Y′-axis is odd in the figures A, D, E and even in the figures B and C. P is inside the polygon when the number of the intersections is odd.

```
100 INIT
110 PRINT 'LON SCREEN (32) OR PLOTTER (1)? ';
120 INPUT D1
130 IF D1=32 THEN 150
140 PRINT @1,32:'BP1'
150 DIM X1(31),Y1(31),X2(40),Y2(40),X5(31)

160 REM     Necessary for further applications:
170 REM       coordinates of the polygon in X1,Y1
180 REM       and reservation of X5
190 REM     For this tip, coordinates are randomly
200 REM     generated.

210 X2(1)=125*RND(-1)+0.5
220 FOR I=1 TO 40
230      X2(I)=125*RND(1)+0.5
240      Y2(I)=95*RND(1)+0.5
250 NEXT I
260 PRINT @D1:'LTHE TOTAL PICTURE:'
270 FOR I=1 TO 40
280      MOVE @D1:X2(I),Y2(I)
290      GOSUB 1090
300 NEXT I
310 FOR I=1 TO 2000
320 NEXT I

330 REM Polygon in this application is a 'circle'.
340 REM For other applications read in the polygon
350 REM coordinates.

360 SET DEGREES
370 FOR I=1 TO 30
380      X1(I)=62+40*SIN(12*I)
390      Y1(I)=47+40*COS(12*I)
400 NEXT I

410 REM The polygon must be closed:

420 X1(31)=X1(1)
430 Y1(31)=Y1(1)
440 MOVE @D1:X1(1),Y1(1)
450 FOR I=2 TO 31
460      DRAW @D1:X1(I),Y1(I)
470 NEXT I
480 FOR I=1 TO 8000
490 NEXT I
500 IF D1=32 THEN 530
510 PRINT @1,32:'BP2'
520 GO TO 540
530 PRINT 'LJJThe selected points:'
540 M1=1000000
550 M2=0

560 M3=1000000
570 M4=0

580 REM Calculation of the min- and max-coordinates
590 MOVE @D1:X1(1),Y1(1)
600 FOR I=1 TO 30
610      DRAW @D1:X1(I),Y1(I)
620      X3=X1(I)
630      M1=M1 MIN X3
640      M2=M2 MAX X3
650      Y3=Y1(I)
660      M3=M3 MIN Y3
670      M4=M4 MAX Y3
680 NEXT I
690 DRAW @D1:X1(31),Y1(31)

700 REM Main part of this program follows now:

710 N1=0
720 FOR I=1 TO 40
730      X3=X2(I)
740      IF X3<M1 OR X3>M2 THEN 970
750      Y3=Y2(I)
760      IF Y3<M3 OR Y3>M4 THEN 970
770      S1=0
780      FOR K=1 TO 31
790           X4=X1(K)-X3
800           IF X4<>0 THEN 850
810           X4=0.1

820           REM Temporarily change coordinate of
830           REM polygon point to value less than the
840           REM accuracy of the digitized points.

850           X5(K)=X4
860 NEXT K
870 FOR K=1 TO 30
880      L=K+1
890      X4=X5(K)
900      X6=X5(L)
910      IF X4*X6>0 THEN 940
920      IF (X6*(Y1(K)-Y3)-X4*(Y1(L)-Y3))/(X6-X4)<0 THEN 940
930      S1=S1+1
940 NEXT K
950 IF 2*INT(S1/2)=S1 THEN 990
960 N1=N1+1
970 MOVE @D1:X3,Y3
980 GOSUB 1090
990 NEXT I
1000 IF N1>0 THEN 1050
1010 PRINT '_There is no point within the polygon'
1020 IF D1=32 THEN 1030
1030 PRINT @1,32:'BC'
1040 PRINT 'LWait until plotter is ready!'
1050 PRINT '_IIOnce again? (Y/N) G';
1060 INPUT A$
1070 IF A$='y' THEN 100
1080 END
1090 RMOVE @D1:-0.5,0
1100 RDRAW @D1:1,0
1110 RMOVE @D1:-0.5,0.5
1120 RDRAW @D1:0,-1
1130 RETURN
```

Drawing from Data Arrays

by Herman D'Hondt
Tektronix, Inc.
Sydney, Australia

To generate drawings from data arrays, you can use either MOVE/DRAW commands or PRINT @32,Z: commands. If the array contains DRAWs only, you could use DRAW X,Y or PRINT @32,20:D. However, in most cases, we want mixtures of MOVEs and DRAWs and a loop is required.

Suppose we set up the following array:

```
100 N=100
110 DIM D(3,N)
120 REM Enter data into D
130 REM D(1,I) for Z, D(2,I) for X and
        D(3,I) for Y
140 REM D(1,I) is 20 (DRAW) or 21 (MOVE)
```

To draw this array we could use a subroutine such as:

```
1000 FOR I=1 TO N
1010 PRINT @32,D(1,I):D(2,I),D(3,I)
1020 NEXT I
```

The disadvantage is that the data must be in GDU's (graphic display units) and any windowing becomes impossible. Also, lines which are not completely on the screen will not be clipped, but scissored. In other words, those lines do not appear at all, nor does the cursor move to the point where the line disappears off the screen.

A second method is given in this subroutine:

```
1000 FOR I=1 TO N
1010 IF D(1,I)=21 THEN 1040
1020 DRAW D(2,I),D(3,I)
1030 GO TO 1050
1040 MOVE D(2,I),D(3,I)
1050 NEXT I
```

While this works fine, it's not very clean because MOVEs and DRAWs are handled separately. Also, the IF and GOTO statements waste both time and memory.

A third way to handle this problem combines the advantages of both other methods, to produce the following subroutine:

```
1000 FOR I=1 TO N
1010 DRAW @32,D(1,I):D(2,I),D(3,I)
1020 NEXT I
```

The key to this routine lies in the way the 4050 handles keywords and secondary addresses. The DRAW command tells the processor to convert from UDUs (user display units) to GDUs, taking into account WINDOW and VIEWPORT, clipping as required, and to send the resultant screen coordinates down to the device (32). The processor is not told to draw, just to transmit the coordinates.

It is the secondary address that tells the device what to do with those coordinates. In this case 20 means 'DRAW,' 21 means 'MOVE.' If a MOVE-secondary-address is included in a DRAW statement, the device will MOVE, not DRAW. Similarly, a secondary address of 12 (PRINT) would cause the screen coordinates to be printed, not drawn, on the screen.

Dimensioning the array as D(3,N) rather than as D(N,3) is done for a specific reason too: it allows the program to use an array-READ to read the data from a DATA statement, while keeping X,Y and Z coordinates separate for ease of programming (remember that arrays are read in ROW-MAJOR order).

For example:

```
200 REM First the Z-Coords for the Points
210 DATA 21,20,21,20,21,20,20,20,21,20,
         21,20,21,20,20
220 REM NOW X
230 DATA 3,3,0,6,14,8,8,14,8,12,16,16,22,16,22
240 REM AND Y
250 DATA 0,8,8,8,8,8,0,0,4,4,8,0,8,4,0
260 WINDOW 0,22,0,8
270 DIM D(3,15)
280 READ D
290 END
```

This will produce the word TEK across the screen, or anywhere else, depending on the VIEWPORT.

Default Response is Risky

by David Yager
Sam Houston State University
Huntsville, TX

Previous YES-NO branching tips (TEKniques Vol. 1 No. 8 and Vol. 2 No. 6) assigned a yes or no default when the RETURN key was pressed with no entry. I've found that pressing the RETURN key by itself is a very common mistake and assigning a default to an accidental touch of this key is risky. Therefore, the following code will see the RETURN key response as inappropriate and will ask for a correct response.

```
500 PRINT "INPUT YES OR NO. ";
510 INPUT C$
520 GO TO POS(" YESNO",C$,1) OF 500,600,500,500,700,500
530 GO TO 500
600 PRINT """YES"" RESPONSE."
610 END
700 PRINT """NO"" RESPONSE."
710 END
```

Numeric Equivalent of Character Strings

by G.E. Gathers
General Electric
Erie, PA

The following routine stores the numeric equivalent of character strings in an array. I'm writing a program to create a Gantt chart that requires up to 36 variable names and this routine solves the problem very nicely.

```
100 INIT
110 PAGE
120 DIM A$(24),B$(48),A(10,24)
130 FOR I=1 TO 10
140 PRINT USING """ENTER NAME "",""["",24"" "",""]"",25""B"",""GG"",S":
150 INPUT B$
160 B$=B$&"
170 A$=SEG(B$,1,24)
180 FOR J=1 TO 24
190 B$=SEG(A$,J,1)
200 A(I,J)=ASC(B$)
210 NEXT J
220 NEXT I
230 PRINT
240 FOR I=1 TO 10
250 FOR J=1 TO 24
260 B$=CHR(A(I,J))
270 A$=REP(B$,J,1)
280 NEXT J
290 PRINT A$
300 PRINT
310 NEXT I
320 END
```

Complex Variable Handling in BASIC

by Colin Archibald
National Research Council
of Canada
Ottawa, Canada

The BASIC language can be used to create and manipulate complex variables, even though these features are not 'built in.' To store a complex number, two numeric variables are used; one for the real part and one for the imaginary part. Example C1, C2. The functions used in other languages to manipulate this complex variable can easily be simulated in BASIC. Here are some examples of complex operations.

1. **Multiply**

```
REM C1, C2 * D1, D2 = A1, A2
```

First do the Real part:

```
A1=C1*D1-C2*D2
```

Then, the Imaginary part:

```
A2=D1*C2+D2*C1
```

2. **Divide**

```
REM C1, C2 ÷ D1, D2 = A1, A2
```

The Real Part

```
A1=(C1*D1+C2*D2)/(D2↑2+D1↑2)
```

The Imaginary Part

```
A2=(D1*C2-C1*D2)/(D2↑2+D1↑2)
```

3. **Add**

```
REM C1, C2 + D1, D2 = A1, A2
```

The Real Part

```
A1=C1+D1
```

The Imaginary Part

```
A2=C2+D2
```

4. **Subtract**

```
REM C1, C2 - D1, D2 = A1, A2
A1=C1-D1
A2=C2-D2
```

5. **Complex Absolute Value**

Compute the complex absolute value of C1, C2 and store the result in R.

```
R=ABS(SQR(C1↑2+C2↑2))
```

6. **Conjugate Complex Variable**

Store the conjugate of C1, C2 in A1, A2

```
A1=C1
A2=-C2
```

Handling arrays of complex variables can be done in a similar manner. As with the 'simple' complex variables the operations can be simulated. One dimensional or multidimensional arrays can be used. The following is a simple example of multiplying two, one dimensional complex arrays.

```
100 REM DECLARE THE ARRAY VARIABLES
110 DIM A1(10),A2(10),C1(10),C2(10),
    D1(10),D2(10)
120 !
130 !
```

. Assign some values to C1, C2, D1, D2 arrays.

```
300 REM MULTIPLY THE ARRAYS.
305 REM STORE RESULTS IN A1,A2.
310 FOR I=1 TO 10
320 A1(I)=C1(I)*D1(I)-C2(I)*D2(I)
330 A2(I)=D1(I)*C2(I)+D2(I)*C1(I)
340 NEXT I
```

Editor's Note: PLOT 50 Mathematics Vol. 1 contains extensive complex number handling capability.

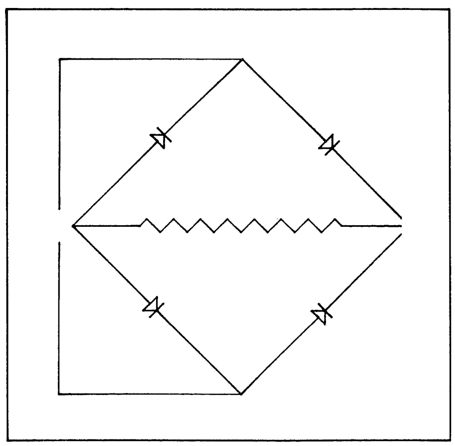
Editor ROM Doesn't Access Disk

TEKniques Vol. 5 No. 3 published a BASIC Bit "Saving Programs to Disk." Unfortunately, this programming tip mislead readers into thinking that text created with the R06 Editor ROM could be saved from 4050 memory directly to a 4907 disk file. This is not so; a tape file has to be used as an intermedium. We apologize to our readers for the ambiguity.

Multiple MOVES/DRAWS In One Line of Code

by David Yager
Sam Houston State University
Huntsville, TX

The games tape (Recreational Plots T1) is not a frivolous purchase as some might believe. Many tasty programming techniques are showcased. For instance, in reviewing the Lunar Lander game I've noticed that the PRINT @32,20: command can stack up point pairs so many lines can be drawn with just one line of code. The example below would take about 47 lines if each draw were done one at a time.



```
300 MOVE 80,100
310 PRI @32,20:105,75,107,77,103,73,107,73.107,77,107,73,109,75,105,71
320 PPI @32,20:107,73,130,50,105,25,101,25,105,25,103,27,105,25,107,23
330 PRI @32,20:105,25,105,21,101,25,103,23,80,0,55,25,57,27,53,23,55,25
340 PRINT @32,20:51,25,55,25,55.29,51,25,53,27,30,50,55,75,53,77,57,73
350 PRINT @32,20:57,77,59,75,55,79,57,77,53,77,57,77,80,100
360 MOVE 30,50
370 PRINT @32,20:50,50,52,52,56.48,60,52,64,48,68,52,72,48,76,52,80,48
380 PRINT @32,20:84,52,88,48,92,52.96.48,100,52,104,48,108,52,110,50
390 DRAW 130,50
400 MOVE 80,100
410 PRINT @32,20:26,100,26.55
420 MOVE 80.0
430 PRINT @32,20:26.0,26,45
```

Editors Note: This only works with graphic display units, i.e., secondary addresses not commands MOVE and DRAW.

Dedicate UDK's to Programs in Progress

by Deggary N. Priest
Commercial Testing Co., Inc.
Dalton, GA

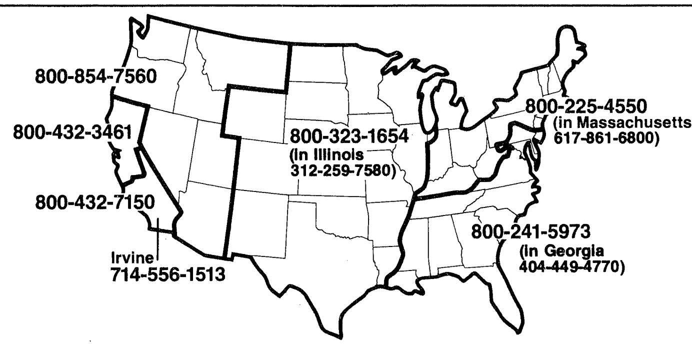
I have found useful a UDK overlay dedicated to programs being written. Since our 4051 is not free all day, I find myself constantly having to "save" my program and give up the machine for short periods. Temporarily inserting the following lines into the program (to be removed after completion of de-bugging) helps a great deal to get started again or exit with the push of a key.

```
1 GO TO 100
36 I9=MEMORY
37 CALL "EDITOR"
38 END
40 FIND N
41 SAVE
42 END
80 FIND N
81 OLD
```

Tektronix Institutes Direct Order Toll-Free Numbers

Order 4050 Series Applications Library programs through toll-free numbers for quick service. The following map delineats the geographical regions and the toll-free number serving each region.

Call the number serving your area and give the customer service representative the nine-digit part number and name of the Applications Library program you wish. If you have any questions, call your local Tektronix Field Office.



4050 Series Applications Library Programs

Ordering

Programs included in the Applications Library prior to September 1981 are packaged and nomenclated by function. Those programs accepted into the Library after September 1981 are packaged and nomenclated with the Volume and Number of the corresponding issue of TEKniques in which the package was announced.

Each package includes the source code on tape or disk (T= tape ; D= disk) together with the supporting documentation; listings are not included in the documentation. Documentation may be purchased separately.

The 4050 Series Applications Library Programs catalog (September 1981) contains the abstracts describing the programs in each package along with representative output in most cases. The catalog part number is 062-6343-00.

To receive a copy of the catalog, or to order a package, contact your local Tektronix field office. The field office has the current prices.

Package Title	Documentation Part #	Package Part #
Business Aids T1	062-5987-00	062-5987-01
Business Aids T2	062-5988-00	062-5988-01
CAD T1	062-5976-00	062-5976-01
CAD D1	062-5977-00	062-5977-01
Character Generator T1	062-5951-00	062-5951-01
Education/Research T1	062-5982-00	062-5982-01
Education/Research T2	062-5983-00	062-5983-01
Electrical Engineering T1	062-5978-00	062-5978-01
Graphing T1	062-5964-00	062-5964-01
Graphing T2	062-5965-00	062-5965-01
Graphing T3	062-5966-00	062-5966-01
Graphing D1	062-5967-00	062-5967-01
Graphing D2	062-5968-00	062-5968-01
Interfacing T1	062-5984-00	062-5984-01
Mapping T1	062-5980-00	062-5980-01
Mechanical Engineering T1	062-5979-00	062-5979-01
Programming Aids T1	062-5971-00	062-5971-01
Programming Aids T2	062-5972-00	062-5972-01
Project Aids T1	062-5985-00	062-5985-01
Project Aids D1	062-5986-00	062-5986-01
Recreational Plots T1	062-5989-00	062-5989-01
Slidemaker T1	062-5962-00	062-5962-01
Slidemaker D1	062-5963-00	062-5963-01
Text Processing T1	062-5969-00	062-5969-01
Text Processing D1	062-5970-00	062-5970-01
Utilities T1	062-5974-00	062-5974-01
Utilities D1	062-5975-00	062-5975-01
Tekniques Vol. 5 No. 4 T1	062-5981-01	062-5981-01
Tekniques Vol. 6 No. 1 T1	062-6443-01	062-6443-01
Tekniques Vol. 6 No. 1 D1	062-6442-01	062-6442-01

The program material contained herein is supplied without warranty of any kind, and without any representation regarding quality, performance or suitability. TEKTRONIX specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Software support is TEKTRONIX Category C: Software is provided on an "as is" basis.

Program Contributions

Contribute one program to the Applications Library and receive the package of your choice in exchange. Send in the membership card from your 4050 Series Graphic System Reference Manual to get the details. Or call us at (503) 685-3618.

Outside U.S.

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

TEKniques Vol. 6 No. 1 T1 Part #062-6443-01

TEKniques Vol. 6 No. 1 T1 tape consists of 18 programs: four utility, two graphing, one programming aids, two statistics, two interfacing, one electrical engineering, one accounting, two text processing, two project management, and one miscellaneous.

Four of the programs must be transferred to their own dedicated tapes. Complete instructions for accomplishing the transfers are included in the documentation.

The individual abstracts describe the programs.

Program 1

Title: 4907 to 4909 File Transfer and Conversion Utility
Authors: Tony Freixas
Gene Lynch
Howard Mozeico
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 32K
Peripherals: 4907 File Manager
4909 Multi-User File Manager
Optional-4641 Printer
Files: 3 ASCII Program
Statements:

This program transfers files from the TEKTRONIX 4907 File Manager to the TEKTRONIX 4909 File Management System. The program files which contain 4907-related statements can optionally be converted, where possible, to program files which use 4909-related statements. Another supported option is to not transfer any files, but merely list all 4907-related statements contained in program files.

Option 1: File Transfer

Files of any type (except password protected) may be transferred from the 4907 to the 4909. No changes are made to any files.

Option 2: File Conversion

Files of any type (except password protected or SECRETed files) may be converted and transferred from the 4907 to the 4909. For program files, 4907-related statements and their 4909 counterparts are listed. The 4909 statements replace 4907 statements where possible. When a 4907 statement cannot be converted to 4909 form, the 4907 statement is changed to a REMark.

"Large" host binary files (the exact size depends on the amount of system memory available) or host binary files with line numbers greater than 64999 cannot be converted or listed. They must be SAVEd in ASCII format before they can be converted.

Converted programs will not necessarily RUN without some additional program modifications. For example, returned 4909 status messages may not have the same format as 4907 status messages. As a result, sections of programs which extract information from the status messages will have to be changed. The documentation assists in determining what needs to be changed, and how to change it.

Option 3: File Listing

4907-related statements from the program may be listed along with their suggested 4909 counterparts. Note that the program files are unaffected. The statements are not converted; no transfer occurs. The only result is a listing of a portion of the program. Exceptional host binary files as specified above must be saved in ASCII format to be listed.

The three programs reside on and execute from tape. However, files input to these programs must reside on a 4907 File Manager.

```
100 INIT
110 UNIT 1
120 CALL "MOUNT",1,A$
130 CALL "file",1,"sample",A$
140 IF A$="" THEN 210
150 OPEN "sample";1,"R",A$
160 ON EOF (1) THEN 200
170 INPUT #1:A$
180 PRINT A$
190 GO TO 170
200 CLOSE 1
210 END
```

4907 Version

```
100 INIT
110 CALL "IDENTIFY","UNIT:";1
120 REM --> CALL "MOUNT",1,A$
130 CALL "DIRECTORY",A$,"UNIT:";1,"sample"
140 IF A$="" THEN 210
150 CALL "OPEN","sample","LFN:";1
160 ON EOF (1) THEN 200
170 INPUT #1:A$
180 PRINT A$
190 GO TO 170
200 CLOSE 1
210 END
```

4909 Version

Program 2

Title: Micrograph Measurement
Author: Byron J. Bergert
Tektronix, Inc.
Rockville, MD
Memory Requirement: 64K
Peripherals: 4956 Tablet
Files: 1 ASCII Program
Statements: 901

The 4052/4054 Micrograph Measurement program facilitates the measurement of graphic and photographic images (graphic data, electron micrographs, X-rays, etc.) The program performs five basic measurements:

- point-to-point distance
- length of an irregular line
- area of a closed figure
- circumference of a closed figure
- counts

You may also define an interactive measurement where, for example, the datum could be the result of one measurement divided by the result of another (e.g., counts per unit area).

The program prompts you for a measurement sequence, measurement parameters and data identification information. Once you begin the measurement sequence, a tablet menu permits you to:

- erase the last measurement
- go to the next measurement
- repeat the last measurement
- stop and display the data

Software distance filters are provided for the length, area and circumference measurements and for counts. For all measurements except point-to-point, the digitized line, figure or points, and the measurement value are displayed on the graphics screen.

The data are stored both in 4052/4054 memory and on magnetic tape. A statistics routine provides a table containing the number of observations, a mean, a standard deviation, and a standard error of the mean, for the measurements. Frequency histograms may also be generated.

Program 3

Title: 4054 Dynamic Graphics Flowchart Symbols
Author: Craig Bulmer
Tektronix, Inc.
Chicago, IL
Memory Requirement: 64K
Peripherals: 4054 Dynamic Graphics
Optional-4662/3 Plotter
Files: 1 ASCII Program
Statements: 800

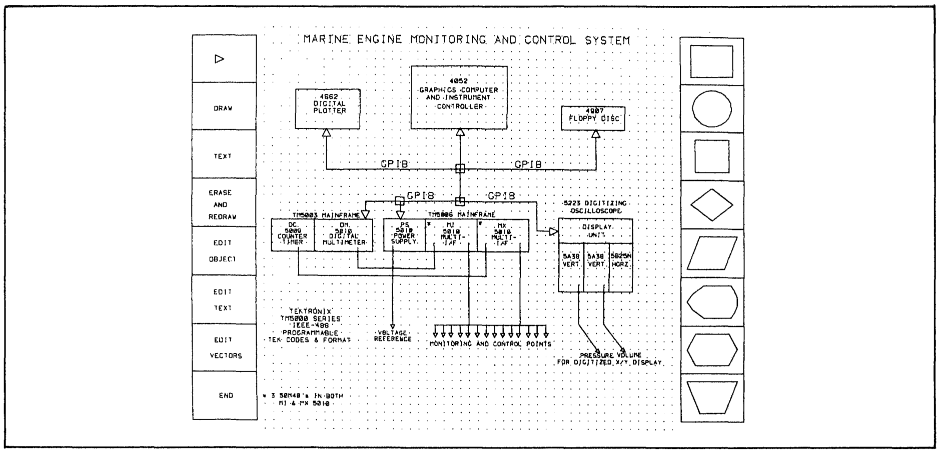
Taking advantage of the 4054 with Opt. 30 or Opt. 31, Dynamic Graphics, the program positions objects, text, or vectors, which are menu selected on the 4054 screen. Output may be to the screen or to the plotter.

Developed pictures can be saved to and redisplayed from premarked files on the internal tape drive.

Applications include flow charting, system configuration diagrams, organizational charts and other similar activities.

Objects include: rhomboid		circle
diamond	sexagon	rectangle
square	terminal	arrow
text	parallelogram	vector

Objects may be changed by recoding. Dashed lines could be used in vectors. Color changes for the 8-pen plotter could be easily incorporated.



Program 4

Title: Data Alignment
Author: Captain Steve Sanford
U.S. Army
Aberdeen Proving Ground, MD
Memory Requirement: 8K
Peripherals: Optional-4924 Tape Drive
Files: 1 ASCII Program
Statements: 104

The program accepts a sequence of randomly spaced X,Y coordinate data from a tape file, in ascending X-value sequence. Output consists of linearly interpolated X,Y values based on a uniformly incremented X-value sequence. The program prompts the user for all options.

This program is applicable to aligning random time-value data for a fixed time interval such as that produced by the tablet digitization program, provided that the digitized data proceeds in ascending X-value sequence.

Program 5

Title: Pie Chart with Panel Fill
Author: Chuck Eng
Tektronix, Inc.
Wilsonville, OR
Revised by: Lynn Cueto
Tektronix, Inc.
Rockville, MD
Memory Requirement: 64K
Peripherals: Optional-4662 Plotter
Files: 1 ASCII Program
Statements: 634

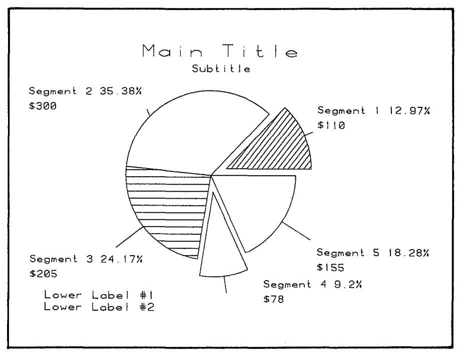
Pie Chart is extremely easy to use. Any number of segments may be assigned text and values. The annotation is printed horizontally around the pie with arrows pointing to their corresponding segments.

The program will annotate each segment with actual values, or will compute and annotate each segment in percentage form, or both. Values must be positive.

Any one or all of the segment may be exploded and/or shaded. Modifications to the chart are through the User-Definable Keys.

If drawn on the plotter, the labels and segments may be different colors. Different character sizes for the labels is optional.

Data may be stored in premarked files on tape and recalled for plotting or changing.

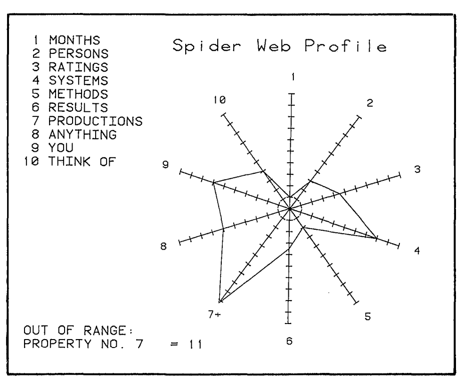


Program 6

Title: Enhanced Spider Web Chart
Author: Tom Price
Lorillard Research
Greensboro, NC
Revised by: Roger Chan
U.S.V. Pharmaceutical Research
Tuckahoe, NY
Memory Requirement: 8K
Peripherals: Optional-4662 Plotter
Files: 1 ASCII Program
Statements: 161

An update of the spider web profiles, this program has an input routine, interactive changeable title, rating scale and rating value. It also handles out of range values.

Output may be to the screen or the plotter, with a different character size for the title and multicolor if the latter.



Program 7

Title: PROGVARLI
Authors: G. Gauglitz
A. Lorch
University of Tuebingen
Tubingen, Germany
Memory Requirement: 32K
Peripherals: 4641 Printer
Files: 1 ASCII Program
Statements: 314

The program lists at the printer any ASCII BASIC program saved on the internal magnetic tape. Each line containing a PRINT, data-input, DIMENSION, DELETE or GOSUB statement, is so referenced.

A table of variables is printed, followed by a list of the variables including line numbers.

A list of REM's, Subroutines, DIM's, DEL's, GO TO's, IF's, and FOR-NEXT loops is created, ending with the total number of statements in the file and the string length (essential to creating a file of minimal length).

```
*   PRINT
+   data-input
=   DIMENSION
!   DELETE
#   GOSUB

#     160  GOSUB 220
      170  SET DEGREES
#     180  GOSUB 250
#     190  GOSUB 350
#     200  GOSUB 560
      210  GO TO 620

*     220  PRINT "LENTER TITLE FOR GRAPH :GG";
+     230  INPUT B$
      240  RETURN

*     250  PRINT "INPUT NO. OF PROPERTIES: GG";
+     260  INPUT N
      270  REM *STORAGE FOR PROPERTY LIST*
=     280  DIM L$(20*N)
      290  L$=""
!     300  DELETE X,R,Q,Y
=     310  DIM X(N),R(N),Q(N),Y(N)
*     320  PRINT "INPUT RATING SCALE`S
           DIVISIONS: GG";
+     330  INPUT S1
      340  RETURN

REM in line:
   130   270   370   440   610   710   800
   880  1110  1190  1320  1420
Subroutine from line to line
   220  -   240
   250  -   340
   350  -   550
   560  -   600
   570  -  600
  1430  -   1510

=     280 DIM L$(20*N)
=     310 DIM X(N),R(N),Q(N),Y(N)

!     300 DELETE X,R,Q,Y

#      24 GOSUB 560
#     160 GOSUB 220
#     180 GOSUB 250
#     190 GOSUB 350
#     200 GOSUB 560
#    1300 GOSUB 1430
#    1340 GOSUB 570

   3 GO TO 100
  16 GO TO 220
  21 GO TO 350
  25 GO TO 620
  28 GO TO 700
  32 GO TO 1
  37 GO TO 320
  40 GO TO 1330
 210 GO TO 620
 430 GO TO 390
 920 GO TO 940
1050 GO TO 1070
1550 GO TO 1480

 410 IF LEN(C$)<=20 THEN 450
 460 IF I=>10 THEN 480
 510 IF LEN(C$)<=K THEN 530
 820 IF T-90>-190 THEN 870
 830 IF I<10 THEN 870
 840 IF X(I)<S1 AND X(I)=>0 OR X(I)<0 THEN 860
 890 IF X(I)=>0 THEN 910
 910 IF X(I)>S1 THEN 930
1020 IF Y(I)>0 THEN 1040
1040 IF Y(I)>S1 THEN 1060
1200 IF Z=32 THEN 1220
1230 IF Z=32 THEN 1260
1280 IF Z=32 THEN 1300
1470 IF X(I)<0 OR X(I)>S1 THEN 1520
1490 IF NOT(F) THEN 1510

 380 FOR I=1 TO N
 540 NEXT I
 620 FOR I=1 TO N
 640 FOR J=1 TO I
 660 NEXT J
 690 NEXT I
 740 FOR I=1 TO N
 960 FOR J=1 TO S1
 990 NEXT J
1090 NEXT I
1160 FOR I=1 TO G
1180 NEXT I
1370 FOR I=1 TO N
1390 NEXT I
1460 FOR I=N TO 1 STEP -1
1480 NEXT I

number of lines: 164
stringlength   : 3297
```

Program 8

Title: Rank Sum Statistic
Author: Richard M. Engeman
Denver Wildlife Research Center
Denver, CO
Peripherals: Optional-4641 Printer
-4662 Plotter
Memory Requirement: 24K
Files: 1 ASCII Program
Statements: 245

This program calculates the test statistic for the rank-sum test. This non-parametric method tests for a shift in location between two unpaired samples (see Hollander and Wolfe, *Non-parametric Statistical Methods,* or Wilcoxon and Wilcox, *Some Rapid Approximate Statistical Procedures.*

The data is input from the keyboard and the program allows the user to correct it after viewing it. The output consists of the raw data, the sorted data, and the test statistic. Significance levels for the test statistic should be looked up in the tables contained in one of the references.

Program 9

Title: Two-Factor Repeated Measures Analysis of Variance
Author: Richard M. Engeman
Denver Wildlife Research Center
Denver, CO
Peripherals: Optional-4641 Printer
Memory Requirement: 32K
Files: 1 ASCII Program
Statements: 407

This program calculates a univariate analysis of variance for data from a two-factor repeated measures experimental design, (see Winer, Statistical Principles in Experimental Design). The program can handle unequal group sizes in addition to the completely balanced case. For an analysis involving unequal group sizes, the user is given the option of analyzing the data with a least squares or unweighted means approach. The program cannot handle missing observations.

The data is input from the keyboard and the user may correct or change it after viewing it on the screen. The output consists of the appropriate analysis of variance table as well as tables of cell totals, means for each subject, means for each treatment level and interaction means.

The user has the option of printing all output, including the raw data, on either the screen or the 4641 printer. Various tasks may be selected from a menu: correcting data, output means tables, output AOV table, etc.

Program 10

Title: CDC 6500 Mainframe I/F
Author: Andreas Goroch
Atmospheric Physicist
Monterey, CA
Memory Requirement: 16K
Peripherals: Option 1 Data Comm. I/F
Files: 1 ASCII Program
Statements: 125

The program calls all required utilities to connect the 4050 desktop as a terminal to a CDC 6500 computer system. Once connected, the 4050 can send and receive data in tape communications mode, as well as terminal mode. Automatic or manual log-in are options.

Program 11

Title: 4050/468 Utility II

Author: Craig Bulmer
Tektronix, Inc.
Chicago, IL

Memory Requirement: 64K
Peripherals: Tektronix 468 Oscilloscope
4052R07/4052R08 ROMs
Optional-4662/3 Plotter
Files: 1 ASCII Program
Requires dedicated data tape
Statements: 1026

This program contains the same functions as the first 4050/468 Utility (abstract #51/00-6125/0 now in the Interfacing T1 package), with several additional features. It will take waveforms from the 468 Oscilloscope and display the waveforms on the 4050 screen; with printed header information of Channel 1, 2 and/or Add; Volts/Div; Time/Div; Trigger Point; Max Volts; Min Volts; Min/Max Pulse Parameters; Histogram Pulse Parameters; Integrate Waveform; Differentiate Waveform; FFT; and Waveform Analysis.

Waveforms can be saved to tape and redisplayed from tape. Output to either screen or plotter with reference scope grid. Waveforms displayed from tape are displayed as dots.

Program 12

Title: PC Component Mechanical Analysis

Author: Tom Sattler
Motorola, Inc.
Ft. Lauderdale, FL

Memory Requirement: 32K
Files: 1 ASCII Program
Statements: 424

Often you want to predict the mechanical strength of electrical components which have been reflow soldered onto a PC board. Throughout the life of the product, the designer must insure that the components will withstand any loading conditions they may see, including tensile, shear and bending. These loads may be incurred from a variety of situations, ranging from a constantly applied load (i.e., as a result of dampening materials used for shock isolation) to the possible insertion of a straight PC board into a slightly warped frame.

This program calculates the direct shearing and tensile forces required for the failure of solder bonds between any component and the PC board, where yielding is considered a failure. It also determines whether or not failure may be expected due to first mode flexing of the board under a rigid component. Both leaded and leadless components can be analyzed.

The first set of data input deals primarily with properties of the solder *after* it has been reflowed. The next set is concerned with the individual types of chip components found on the user's PC board. Chip resistors are dealt with first, and the process repeated for remaining chip components (inductors, IC's, etc.). Information relating to any leaded components on the board is entered last.

For each category of chip components (capacitors, inductors, etc.) the program outputs: part number, tensile force required for the solder beneath the part to fail, the shear force required for the solder beneath the part to fail, and whether or not the imposed maximum board deflection will cause a bending failure.

For leaded components, output will be: part number, number of leads on that component, the force required on the component, normal to the PC board, for the solder to fail.

The documentation details the algorithm used in the program.

Data may be saved on a premarked tape file.

SAMPLE RUN
(STATIC LOADING)

chip capacitors

PART NO.	TENSILE FORCE (LB)	SHEAR FORCE (LB)	BENDING FAILURE?
2001	12.60	8.80	
2002	34.02	23.76	
2003	12.60	8.80	
2004	28.22	19.71	YES
2301	34.02	23.76	

integrated circuits

PART NO.	TENSILE FORCE (LB)	SHEAR FORCE (LB)	BENDING FAILURE?
5001	105.84	73.92	YES
5201	119.55	83.49	
5301	105.84	73.92	

leaded components

PART NO.	NO. OF LEADS	NORMAL FORCE (LB)
6001	4	29.14
6301	2	15.59
6505	2	25.99

Hit page to continue

Program 13

Title: Cu-Sum-Fuel Consumption Measurement

Author: Ron Clark
Scottish Crop Research Institute
Dundee, Scotland

Memory Requirement: 16K
Files: 1 ASCII Program
Statements: 148

Cu-Sum graphs vehicle fuel consumption against a standard. For n number of fill-ups, the user keys in the number of gallons of fuel purchased and the odometer reading at the time of purchase.

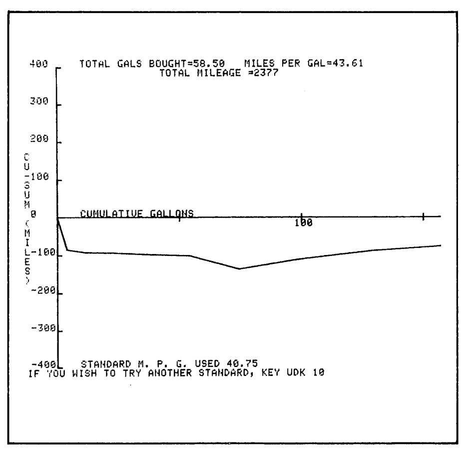
After fuel and odometer figures have been keyed in, the user inputs the estimate of miles per gallon used.

The graph is a cumulative sum of the differences of a set of readings from the expected. Changes in the consumption can easily be seen as the trend of the graph changes.

Data may be corrected. A different standard may be chosen.

If the trend of the graph is horizontal, then the chosen estimate is the correct one.

The program is based on "Measuring and Controlling Vehicle Fuel Consumption" by J. Murdoch (1974).



Program 14

Title: TEXTED

Authors: G. Gauglitz
A. Lorch
University of Tuebingen
Tubingen, Germany

Memory Requirement: 32K
Peripherals: 4641 Printer
Files: 1 ASCII Program
Statements: 261

Text may be created, edited and stored as a binary data string. Previously created text may be recalled and edited from the internal magnetic tape.

Functions:

- list text (line by line) from beginning
- display next page beginning at line n
- display last n lines of text
- display next page
- display last page
- insert new text at n line
- delete lines n to n
- interchange n lines beginning at n
- delete line n, insert new text
- lengthen line n
- change single characters
- delete character
- search
- store text
- add text from n file
- print text

The files on tape have to be marked; this depends on the length and number of the lines of text.

Program 15

Title: Label Printer

Memory Requirement: 24K
Peripherals: 4641 Printer
Files: 1 ASCII Program
Requires dedicated tape
Statements: 720

Use this program to enter, edit and print labels intended as short identifiers, operating instructions, supplemental information, and so on. For instance, labels which will be affixed to manuals, equipment or other such items could be produced by this program.

The program assumes the text will be printed on pinfeed labels.

Specifications: 67 characters per line
20 lines max
1000 characters total max
25 labels per tape (may be easily changed)

Features: Different sized pinfeed labels accommodated.

Form filling –
Label text input with flag for some variable information to be keyed in at run time, e.g., different names on one line with rest of text the same.

Sequencing –
At run time set a beginning value, step between numbers, and the number of repeats printed before incrementing. Useful for controlled documents in which a label with a particular control number may be placed on the binding, inside the cover, and next to the name on a master list.

Editing –
Text of label may be changed line-by-line using edit functions of rubout, expand, compress, backspace/space, insert and clear.

Help –
A UDK will print out a list of the function keys and additional information when a mistake is detected by the program.

Storage and Retrieval –
One label per file may be stored, retrieved, printed, edited, etc.

Listing –
Labels from each file on tape may be read and displayed on the screen.

```
CONFIDENTIAL
CONTROLLED DOCUMENT

ISSUED TO \\\\\\\\\\\\\\\
DOCUMENT %%

CONFIDENTIAL
CONTROLLED DOCUMENT

ISSUED TO PAT KELLEY
DOCUMENT  1

CONFIDENTIAL
CONTROLLED DOCUMENT

ISSUED TO PAT KELLEY
DOCUMENT  1

CONFIDENTIAL
CONTROLLED DOCUMENT

ISSUED TO PAT KELLEY
DOCUMENT  2

CONFIDENTIAL
CONTROLLED DOCUMENT

ISSUED TO PAT KELLEY
DOCUMENT  2
```

Program 16

Title: Fund Usage

Author: W.J. Orvis
Lawrence Livermore
National Labs
Livermore, CA

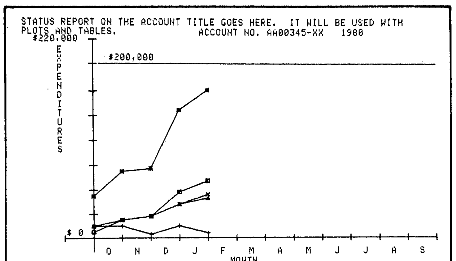
Memory Requirement: 16K
Peripherals: Optional-4641 Printer
-4662 Plotter
-4952 Joystick
4054 Version requires
Dynamic Graphics Opt. 30
Files: 3 ASCII Program
3 Binary Data (examples)
Requires dedicated tape
Statements: 1047

Most project management requires that close watch be kept on project related costs, especially when limited funds are available. This program tracks these costs for several different projects and presents the data in tables or graphs for easy analysis.

Data is processed by fiscal year for each project account. Each account is divided into four subaccounts: 1) Consultants, 2) Subcontracts Billed, 3) Subcontracts Liens, and 4) Internal. The names of these subaccounts are purely arbitrary and could be changed easily.

Data is accumulated monthly. All but type 3 (Liens) are handled as increasing accounts (i.e., when the data is plotted, the data from previous months are added to the current month's data to give a cumulative total). Type 3 (Liens) data are a different matter. They do not represent money spent but are costs that have been incurred but not yet paid. As bills are paid, the costs are shifted from type 3 (Liens) to type 2 (Bills). Therefore, each month's Liens are treated separately and are not added to those from a previous month.

Data is stored on the program tape in premarked binary data files. Account numbers and account titles reside in a file following the program files. A directory to the data files follows this file. The remaining files contain the data for each account.



STATUS REPORT ON THE ACCOUNT TITLE GOES HERE. IT WILL BE USED WITH PLOTS AND TABLES.

ACCOUNT NO. AA00345-XX 1980

	CONSULTANTS	SUBCONTRACTS BILLED	SUBCONTRACTS LIENS	INTERNAL	TOTAL
START	5,000.00	10,000.00	10,000.00	10,000.00	35,000.00
OCT	10,000.00	5,000.00	10,000.00	5,000.00	30,000.00
NOV	3,000.00	3,000.00	3,000.00	3,000.00	12,000.00
DEC	20,000.00	10,000.00	10,000.00	10,000.00	50,000.00
JAN	9,000.00	5,000.00	4,000.00	8,000.00	26,000.00
FEB	0.00	0.00	0.00	0.00	0.00
MAR	0.00	0.00	0.00	0.00	0.00
APR	0.00	0.00	0.00	0.00	0.00
MAY	0.00	0.00	0.00	0.00	0.00
JUN	0.00	0.00	0.00	0.00	0.00
JULY	0.00	0.00	0.00	0.00	0.00
AUG	0.00	0.00	0.00	0.00	0.00
SEP	0.00	0.00	0.00	0.00	0.00
TOTALS	$47,000.00	$33,000.00		$36,000.00	

ACCOUNT LIMIT $200,000.00

Program 17

Title: Manufacturing Sequence Flowcharter

Author: Paul Howard
Tektronix, Inc.
Wilsonville, OR

Memory Requirement: 32K

Peripherals: Optional-4662/3 Plotter
-4641 Printer

Files: 1 ASCII Program
1 ASCII Text
Requires Dedicated Tape

Statements: 731

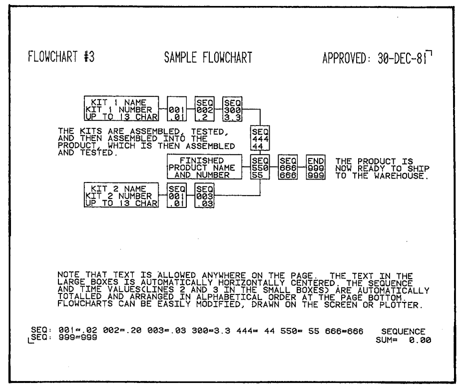
A nontechnical person with little 4050 Desktop Computer experience can easily create and edit flowcharts with this program. Producing and maintaining flowcharts which describe the processes that sets of materials go through to become finished products are the primary targets of this program, however, other flowcharts can be created.

By combining two box types, vertical or horizontal interconnect lines and text you create your flowchart. Four UDK's position the cursor to place or delete the flowchart elements quickly. Text within the two boxes is automatically centered.

Fast redrawing maintains a "clean" sketch on your screen. Once you're satisfied, you may send the flowchart to the plotter, or store it on tape.

A new Flowcharter tape is easily produced by pressing a UDK and following instructions. The program, user's manual and directory file will automatically be transferred to the new tape. Each Flowcharter tape holds 30 flowcharts, however. Flowcharts may be transferred between Flowcharter tapes.

The user's manual contained in a separate file may be sent to the screen or to the 4641 printer.



Program 18

Title: Air Defense Game

Authors: R. Hershman
F. Greitzer
R. Kelly
Navy Personnel R&D Center
San Diego, CA

Memory Requirement: 32K

Peripherals: MicroWorks FP-51
ROM Pack

Files: 2 ASCII Programs
Requires Dedicated Tape

Statements: 669

The Air Defense Game is an interactive scenario in which the player defends his ship by launching missiles against incoming enemy targets. The 4050 simulates a radar screen with the player's ship at the center and enemy raids entering from the periphery.

Difficulty level is selected by menu, and a summary of the player's performance (including a skill rating) is displayed after each engagement. Performance data are stored in binary files. An off-line analysis program assesses performance in greater detail.

TEKniques Vol. 6 No. 1 D1 Part #062-6442-01

TEKniques Vol. 6 No. 1 D1 disk consists of 11 programs: one computer aided education, one electrical engineering, one graphing, one mapping, one programming aids, one project aids, three text processing, and two utility.

The individual abstracts describe the program.

Program 1

Title: Spacetime/Minkowski

Author: Joel A. Gwinn
University of Louisville
Louisville, KY

Memory Requirement: 4054 Option 30
32K

Peripherals: 4907 File Manager

Files: 1 Program
1 Data

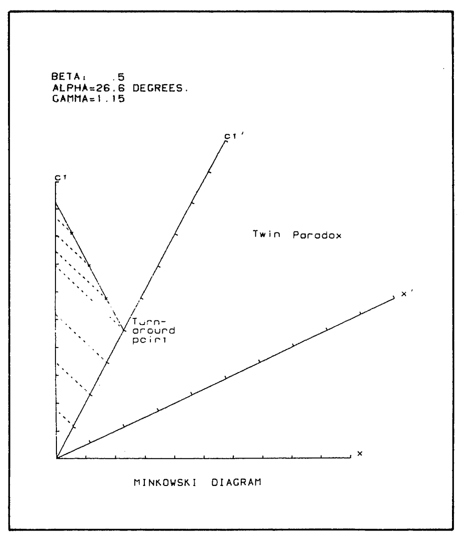
Statements: 315

This program facilitates graphical solution of kinematics problems in Special Relativity Theory using the Minkowski Diagram, a graphical representation of the Lorentz Transformation.

Spacetime/Minkowski elicits the relative velocity of two observers, and constructs a system of space and time coordinates corresponding to the following:

Observer 0′ is fixed at the origin of a spatial reference frame (rocket frame) which moves at speed v = c (c is the speed of light in free space) through the reference frame (laboratory frame) of observer 0. At time zero in both frames, the origins of the space reference frames coincide. Subroutines controlled by the User-Definable Keys provide, in refresh mode, the essential elements of the graphical solution.

A calculator mode is available for numerical work.



Program 2

Title: Printed Circuit Board Layout
Author: Robert K. Hulett
Tektronix, Inc.
Albuquerque, NM
Memory Requirement: 16K
Peripherals: 4907 File Manager
4662/4663 Plotter
4952 Joystick
(Joystick not required)
if using 4054)
Files: 6 Program
2 Data (1 a sample)
Requires Data Disk
Statements: 1174

This software package permits the operator to create, draw, or modify printed circuit boards. The package contains six programs:

Boot: When transferred to tape, initializes the system from the internal magnetic tape.

Design: Mainly creates and draws the printed circuit board. However, a pattern may be deleted during the design phase.

Editor: Mainly deletes patterns from the circuit board, however, during editor phase, a pattern may be created and added.

Artwork: Draws the finished printed circuit board to the plotter for a finished "Artwork."

Joystick Drift Test: Assists the operator in adjusting the 4952 joystick for no drift.

Plotter Calibration: Calibrates the plotter for a full scale artwork.

The functions in **Design** and **Editor** are:

Add-A-Pattern and Korrector
Edge Connectors (0.100", 0.125", 0.156") (any number of contacts)

Circuit Runs
(any width)

Integrated Circuits
Dual-Inline Pins (DIP)
Round Can (8, 10, & 12 pins)
Flat Pack

Two Pads
Locator pad (for identifying pin #1 of any pattern)
A single pad or solder pad

Transistors Patterns
0.1" pin circle
0.2" pin circle
0.1" inline pins
0.2" inline pins
T 78 package

Program 3

Title: Scatter Plot w/Curve Fitting
Author: Mallory M. Green
Department of H.U.D.
Washington, D.C.
Memory Requirement: 32K
Peripherals: 4907 File Manager
4662 Plotter
Files: 20 Program
1 Data (example)
Statements: 1925

This X,Y scatter plot program is an enhanced and restructured version of General Graphing (Program 5 in Graphing T3).

Graph Parameters
Graph Title — 1 Line
X-Axis Label — 1 Line
Y-Axis Label — 1 Line
Symbol Placement
Each point
Last point
Every nth point

Plot Mode
Point only
Point connected by line
Histogram
Choice of 7 symbols
Axis
Log-Lin
Log-Log
Lin-Log
Lin-Lin

Data Entry — New
From Keyboard or User-Defined Function X and Y, or Y only

Least Squares Fitting
Trim data to be fitted
Select best fit

Y=B*X
Y=A+B*X
Y=A EXP(B*X)
Y=1/(A+BX)
Y=A+B/X
Y=A+B LGT X
Y=A* X↑B
Y=X/(A+BX)

Display Data

Draw Graph
Displayed on screen without labels

Plot Graph
Plotted on 4662 Plotter

Save Graph
Saved to tape or disk

Edit
Any of the graph parameters may be changed.
Data may be changed, deleted or added.

List Graphs
All graphs saved on disk will be listed to the screen.

Program 4

Title: Coordinate Geometry & Utilities
Author: Forrest Gene Stanley
L.A. Brewer & Associates, Inc.
Farmington, NM
Memory Requirement: 32K
Peripherals: 4907 File Manager
Files: 5 Program
Statements: 1650

These programs solve problems common to everyday surveying.

The coordinate geometry program contains routines for coordinate and elevation assignment, the solution of intersections, circular curves, inverse, traverse, reduction of field notes, areas, and various coordinate manipulations.

It also contains routines for angle addition or subtraction, angle normalization, angle averaging, conversion of degrees, minutes, and seconds to and from decimal degrees, and bearings to and from Azimuth directly from the keyboard.

Coordinates and elevations are stored on a temporary file on the 4907 as they are generated. Any number of coordinate triples may be stored up to the memory limit of the disk (approximately 15,000). Permanent storage of the temporary file may be on the same or other disks, or on tape. Three programs accomplish all of the above. Transfer between the three is automatic on completion of instructions to the user by the 4050. In addition a short index program may be transferred to tape which uses the AUTO LOAD feature to set the 4907 clock, mount the disk and load the programs into 4050 memory. A fifth program formats the disk and creates and initializes the files necessary to use the coordinate geometry, coordinate storage, and coordinate recall programs.

Program 5

Title: 4907 FORTRAN to BASIC Converter
Author: Mark Mehall
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 32K
Peripherals: 4907 File Manager
Optional-4050R06
Editor ROM
Files: 3 Program
Statements: 991

This program converts FORTRAN to 4050 Series BASIC. The program is based on the USA Standard FORTRAN, X3.9-1966. The FORTRAN statement labels, variables and subroutine names are changed to their BASIC counterparts and remembered for references throughout the program.

The majority of FORTRAN statements are changed into BASIC by this program. The statements that are not directly compatible are made in REMark's and can be modified using the EDITOR ROM or the 4050 Series Line Editor.

The FORTRAN statements: READ, WRITE, FORMAT, IF, GO TO, DO, DIMENSION, CALL, END, RETURN, STOP, SUBROUTINE, and CONTINUE are automatically changed to BASIC. The FORTRAN internal routines are also converted to the corresponding BASIC routines.

The program prints tables of corresponding FORTRAN statement numbers to BASIC line numbers, FORTRAN variable names to BASIC variables, and FORTRAN subroutine names to BASIC line numbers.

Program 6

Title: Presentation GANTT Chart
Author: T.C. Robertson
Rohr Ind.
Chula Vista, CA
Memory Requirement: 8K
Peripherals: 4907 File Manager
4662 Plotter
Files: 1 Program
1 Binary Data
Statements: 225

This program will produce a GANTT chart suitable for presentations. It will take from 1 to 25 task descriptions, and project from 1 to 20 months. Tasks may begin or end at any week during a month.

Data is stored on disk under a user-assigned file name. Any of the task descriptions or their start/finish dates may be changed.

Output is to the plotter. All spacing is handled automatically. The title is centered.

Program 7

Title: Word Processor
Author: Steve Salisbury
Whirlpool Corporation
Benton Harbor, MI
Memory Requirement: 32K
Peripherals: 4907 File Manager
Optional-4641 Printer
Files: 1 Program
Statements: 236

This program allows the 4050 desktop computer to become a word processor. You can enter, edit and delete text, store it on disk, and print it to the screen or a printer.

Edit functions include a moveable pointer, search and replace, and insert.

By selecting margins, vertical lines are drawn to guide the user in text placement. Text may also be adjusted for line length.

Stored text can be used with "Report Writer" to print formatted reports, letters, and so on.

Data files are on disk and limited to 8192 due to the 32K memory limitation.

Program 8

Title: Report Writer
Author: Steve Salisbury
Whirlpool Corporation
Benton Harbor, MI
Memory Requirement: 32K
Peripherals: 4907 File Manager
4641 Printer
Files: 1 Program
Statements: 107

This program uses text developed in a word processing program and prints the text in report-type format. Several text files can be linked together if the report is extremely long.

Specifications for formatting:
- Paper length in lines
- Lines to be printed per page (for top and bottom margins)
- Spacing
- Characters to be indented for extra left margin

The program places page numbers at the top of each page starting with page 2.

The program reads binary, sequential files from the 4907 File Manager.

Program 9

Title: Memogenda
Author: Douglas DeWitt
Mepco/Electra, Inc.
Columbia, SC
Memory Requirement: 8K
Peripherals: 4907 File Manager
Optional-4641 Printer
Files: 1 Program
Statements: 233

This program allows you to use the 4050 Desktop Computer along with the 4907 disk as an appointment calendar. Reminders for any day of the year may be stored and recalled.

The program automatically creates 1000 bytes of storage for each month in which you file your one line reminder. The length of the monthly files could easily be changed to accommodate your schedule.

The program also has a utility routine to set the 4907 clock, mount the disk, or format a disk, if necessary. Thus, by transferring the program to tape, a user need only press AUTOLOAD to get started.

```
Date         : 01-FEB-81
Reminder for : DOUGLAS
$371.00 DUE ON CAR INSURANCE
RUN JAN MONTH END****
```

Program 10

Title: R040 File Maintenance
Author: John Cuder
Rohr Industries
Riverside, C
Memory Requirement: 32K
Peripherals: 4907 File Manager
Files: 1 Program
Statements: 1201

This program allows a disk file to emulate an ISAM (indexed sequential access method) file. The advantages of an indexed file include: accessing a master record by key; preventing the addition of duplicate records to a file; obtaining lists of data in "key" sequence without additional sorting.

In R040 two files, "MPKEY" and "MPMASTER," simulate one indexed or ISAM file. The "MPKEY" file contains pointer records which "point" to the relative record position of each part number record in the "MPMASTER" file. The first record in "MPKEY" is a control record which carries the count of records in the "MPMASTER" file, to determine where to add each new record.

R040 adds, changes and deletes (logically) manpower master records in the "MPMASTER" file which consists of 1500 records, 224 bytes long. Each record contains 50 fields of which 48 may be changed.

The program is intended as an illustration and was not designed to be universal in application. Each user would be required to structure his record and select his "key" according to his own system requirements.

Program 11

Title: Disk-to-Tape File Duplicator
Author: Andy Lau
Tektronix, Inc.
Wilsonville, Oregon
Memory Requirement: 8K
Peripherals: 4907 File Manager
Files: 1 Program
Statements: 109

This utility transfers binary programs on a 4907 disk to ASCII programs on the internal tape drive of a 4050 system. Programs are transferred in alphabetical order of their file names. After all are transferred, a list of file names is printed on the screen. Line numbers on programs to be transferred may not exceed 59999.

4050 Applications Library Program Updates

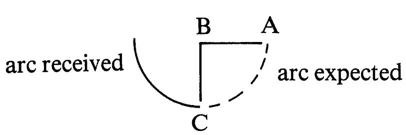
Package/Program/File

062-5976-01/CAD T1
Program 1/Drafting Digitizer
File 3

062-5977-01/CAD D1
Program 3/Drafting Digitizer
"@DRAFTING/DIGITIZE"

Submitted by: J. Hunter Young
Waikato Valley Authority
Hamilton East
South Auckland
New Zealand

In certain cases the partial arc and partial hidden arc portions of this program would draw entirely incorrect arcs. For example:

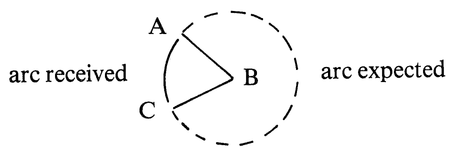


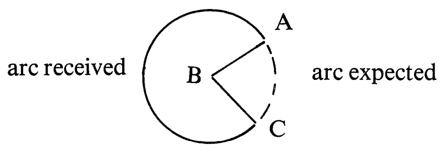
This problem was corrected by altering statement 6220:

FROM
```
6220 IF X(J)<X(J1) THEN 6250
```

TO
```
6220 IF X(J1)<X(J) THEN 6250
```

We also found that the arc drawn was generally from the smaller angle to the larger angle, regardless of what point was digitized first. So in some cases, the wrong arc, as far as the user was concerned, was drawn. For example:





This was corrected by altering lines 6260–6290:

```
6260 A2=(A1-A)/10
6270 IF A2>0 THEN 6285
6275 A1=A1+360
6280 A2=ABS(A2)
6285 MOVE @Q:X(K),Y(K)
6287 A3=A+A2
6290 IF Y(I)<40 THEN 7060
```

if A1<A then add 360°
to second angle A1 and
make A2 positive

and deleting lines 7030 to 7050, inclusive.

062-5976-01/CAD T1
Program 1/Drafting Digitizer
Files 2 and 3

062-5977-01/CAD D1
Program 3/Drafting Digitizer
"@DRAFTING/MENU"
"@DRAFTING/DIGITIZE"

Submitted by: Tom Sutherlin
Cameron University
Lawton, OK

Mr. Young's response is good. He's the first user who has solved the problem alone. I have been modifying the software but hadn't sent you the new package.

Additions: Cross Hatch Menu Item and Routine

Improvements: Hidden Line
Hidden Circle
Partial Arc
Partial Hidden Arc
Angle Calculation

Change File 2 on the tape or "@DRAFTING/MENU" on the disk per the following instructions:

DELETE 1570,1580

ADD

```
1570 MOVE @1:14.25,7.7
1580 PRINT @1:"CROSS";
1590 MOVE @1:14,25,7.4
1600 PRINT @1:"HATCH";
1610 MOVE @1:15,10
1620 END
```

Change File 3 on the tape or "@DRAFTING/DIGITIZE" on the disk to reflect the following:

DELETE 300,510

ADD

```
300 IF X(I)>140 THEN 400
310 GO TO INT(Y(I)/10) OF 9000,8000,7000,6000,
    5000,4000,3000,2000.1000
320 GO TO 25000
400 REM TESTING FOR VERTICAL POSITION IN
    SECOND COLUMN
410 GO TO INT(Y(I)/10)OF 520,520,520,520,520,520
420 GO TO INT(Y(I)/10-6) OF 14000,13000,12000
520 REM INSERT ERROR MESSAGES FROM 530 THRU 990
530 STOP
```

DELETE 2010,2160

ADD
```
2010 MOVE @O:X(J),Y(J)
2020 IF S<7.5 THEN 2050
2030 T=10
2040 GO TO 2060
2050 IF S<4 THEN 2055
2052 T=5
2054 GO TO 2060
2055 T=2
2060 S1=S/(2*T+1)
2070 FOR L=1 TO 2*T STEP 1
2080 RDRAW @O:S1,0
2090 L=L+1
2100 RMOVE @O:S1,0
2110 NEXT L
2120 RDRAW @O:S1,0
2130 P=0
2140 GO TO 240
```

CHANGE
```
4010 MOVE @Q:X(J)+S,Y(J)
```

CHANGE
```
5010 MOVE @Q:X(J)+S,Y(J)
```

DELETE 6010,6330

ADD
```
6010 A1=A
6020 J=I-2
6030 K=I-3
6040 GOSUB 23030
6050 IF A>A1 THEN 6070
6060 A=A+360
6070 A2=(A-A1)/10
6080 MOVE @O:X(I-1),Y(I-1)
6090 IF Y(I)<40 THEN 7060
6100 FOR L=A1+A2 TO A STEP A2
6110 DRAW @O:S*COS(L)+X(I-2),S*SIN(L)+Y(I-2)
6120 NEXT L
6130 P=0
6140 GO TO 240
```

DELETE 7030,7090

ADD
```
7060 FOR L=A1+A2 TO A STEP A2
7070 DRAW @O:S*COS(L)+X(I-2),S*SIN(L)+Y(I-2)
7080 L=L+A2
7090 MOVE @O:S*COS(L)+X(I-2),S*SIN(L)+Y(I-2)
```

ADD
```
14000 REM CROSS-HATCH SUBROUTINE
14001 REM Shoding Routine for Complex Shapes
14002 REM BY: Chuck Eng,Tektronix, Inc.
14003 REM TEKniques Vol.3 No.1 February 1,1979
14005 SET RADIANS
14010 N=P-1
14020 DIM P2(2,N),W(2,N+1),O2(N-1),P1(2)
14025 GOSUB 14600
14030 S3=1.0E+307
14040 L1=-1.0E+307
14050 S4=SIN(45)
14060 O3=COS(45)
14070 FOR L=1 TO N
14080 W(1,L)=P2(1,L)*O3+P2(2,L)*S4
14090 W(2,L)=-P2(1,L)*S4+P2(2,L)*O3
14100 S3=S3 MIN W(2,L)
14110 L1=L1 MAX W(2,L)
14120 NEXT L
14130 W(1,N+1)=W(1,1)
14140 W(2,N+1)=W(2,1)
14150 O1=S3+0,5*1.25
14160 IF O1>L1 THEN 14430
14170 O4=0
14180 FOR L=1 TO N
14190 W1=W(2,L) MIN W(2,L+1)
14200 IF O1<W1 THEN 14320
14210 W2=W(2,L) MAX W(2,L+1)
14220 IF O1=>W2 THEN 14320
14230 O5=W(1,L)-(W(2,L)-O1)*(W(1,L)-W(1,L+1))/
      (W(2,L)-W(2,L+1))
14240 O4=O4+1
14250 O=O4

14260 IF O=1 THEN 14310
14270 IF O2(O-1)=>O5 THEN 14310
14280 O2(O)=O2(O-1)
14290 O=O-1
14300 GO TO 14260
14310 C2(O)=O5
14320 NEXT L
14330 FOR L=1 TO O4
14340 P1(1)=O2(L)*O3-O1*S4
14350 P1(2)=O2(L)*S4+O1*O3
14360 IF 2*INT(0.5*L)-L>-0.1THEN14390
14370 MOVE @O:P1(1),P1(2)
14380 GO TO 14400
14390 DRAW @O.P1(1),P1(2)
14400 NEXT L
14410 O1=O1+1.25
14420 GO TO 14160
14430 P=0
14435 DELETE P2,W,O2,P1
14440 GO TO 240
14600 REM I.D. OF COORDINATES FOR CROSS
      HATCHED AREA
14610 P=I-P
14630 FOR L=1 TO N
14640 P2(1,L)=X(P+L)
14650 P2(2,L)=Y(P+L)
14660 NEXT L
14670 MOVE @O:P2(1,1),P2(2,1)
14675 SET DEGREES
14680 RETURN
14690 P=0
14700 GO TO 240
```

DELETE 23045,23280

ADD
```
23050 S=ABS(SOR(H↑2+V↑2))
23060 IF H=0 THEN 23160
23070 A=ATN(V/H)
23080 IF V=0 THEN 23130
23090 IF H<0 THEN 23110
23100 GO TO 23200
23110 A=A+180
23120 GO TO 23200
23130 IF X(J)<X(K) THEN 23200
23140 A=180
23150 GO TO 23200
23160 IF Y(J)<Y(K) THEN 23190
23170 A=270
23180 GO TO 23200
23190 A=90
23200 ROTATE A
23210 RETURN
```

Change the documentation for Drafting Digitizer:

Locate the part of the documentation that contains modifications for digitizing from the 4662 Plotter. Note that under paragraph **2. Input will be from the 4662 Plotter,** the documentation presently reads:

CHANGE
FROM: 180 INPUT @8:X(I),Y(I),Z$
TO: 180 INPUT @1:X(I),Y(I)

correct the second line of the above to read:

TO: 180 INPUT @1,27:X(I),Y(I)

This will allow the use of the CALL button on the 4662 for digitizing input.

062-5969-01/TEXT PROCESSING T1
Program 2/Text Editor
File 3

Submitted by: R.G. Stevens
Tektronix, Inc.
Melbourne, Australia

Line 2130 dimensions B$ to a possibly large size (in one customer's case, more than 9000 characters). The following modification to the program is suggested to overcome possible MEMORY problems when saving text to the internal tape.

```
DELETE 2100
DELETE 2130,2280
DELETE 2330

2322 FOR I=1 TO T
2324 C$=SEG(A$,B(I),A(I))
2326 PRINT @33:C$
2328 NEXT I
2330 CLOSE
```

The overall effect is the same without having to dimension the B$ variable and possibly run out of memory.

062-5969-01/TEXT PROCESSING T1
Program 8/Print Mail Addresses and Form Letters
File 13

Submitted by: George Reis
Tektronix, Inc.
Beaverton, OR

As presently written, this program allows you to add labels to those already entered into 4050 memory from the keyboard or tape. However, these names and addresses are added to the end of the list.

By including the following lines of code in your program, you may insert labels anywhere within a list, providing the file has enough room.

```
72 REM Insert Lobels
73 GO TO 5000

5000 REM INSERT A LABEL ROUTINE
5010 PAGE
5020 IF LEN(A$)<8030 AND 1<150 THEN 5060
5030 PRINT "THERE'S NOT ENOUGH ROOM LEFT;
     IT WILL HAVE "
5040 PRINT "TO BE PUT IN ANOTHER FILEGJ"
5050 END
5060 PRINT "WARNING:  GGGDO NOT TYPE 'DONE'
     AS AN INPUT "
5070 PRINT "OR YOU MAY DESTROY YOUR DATA BASE
     IN THE 4050J"
5080 PRINT "WHERE DO YOU WANT THE NEW
     LABEL INSERTED?"
5090 PRINT "IT WILL BE INSERTED AHEAD OF THE
     LABEL WITH "
5100 PRINT "THIS NUMBER: ";
5110 INPUT L1
5120 PRINT "J"
5130 GOSUB 1040
5140 B=LEN(B$)
5150 IF LEN(A$)+B=>8030 THEN 5030
5160 IF B<1 THEN 5000
5170 A$=REP(B$,A(L1),0)
5180 IF L1=I THEN 5210
5190 FOR K=I TO L1+1 STEP -1
5200 A(K)=A(K-1)+B
5210 NEXT K
5220 I=I+'
5230 A(I)=LEN(A$)+1
5240 END
```

Press UDK 18 to insert each label. The program will prompt you for the number of the label before which the new label will be inserted. You will then be asked for the new label information.

062-5969-01/TEXT PROCESSING T1
Program 9/$Edit.Dos
File 15

Submitted by: Denny Chamberlin
Tektronix, Inc.
Wilsonville, OR

Although this program is intended for the 4051, I ran it on a 4051. For those who might be using a 4051 with this program, the dimension of one variable (string) needs to be changed in statement 120. H$ must be dimensioned to 3.

```
120 DIM A$(1),B$(1000),E$(73),H$(2),I$(100),
    M$(1),R$(20),T$(1000),Z$(1)
```

062-5966-01/GRAPHING T3
Program 4/Data Graphing
File 6

Submitted by: Denny Chamberlin
Tektronix, Inc.
Wilsonville, OR

Delete three lines of code from this file:

DELETE
```
DELETE 9280
DELETE 9290
DELETE 9300
```

A holdover from the days this program didn't allow negative data, this routine results in erroneous data when calculating the cumulative sum of the previous curve.

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

Address Correction Requested — Forwarding and Return Postage Guaranteed.

COMMITTED TO EXCELLENCE

TEKTRONIX, INC.
Information Display Division
Applications Library
Group 451
P.O. Box 500
Beaverton, Oregon 97077

BULK RATE
U.S. POSTAGE
PAID
TEKTRONIX, INC.