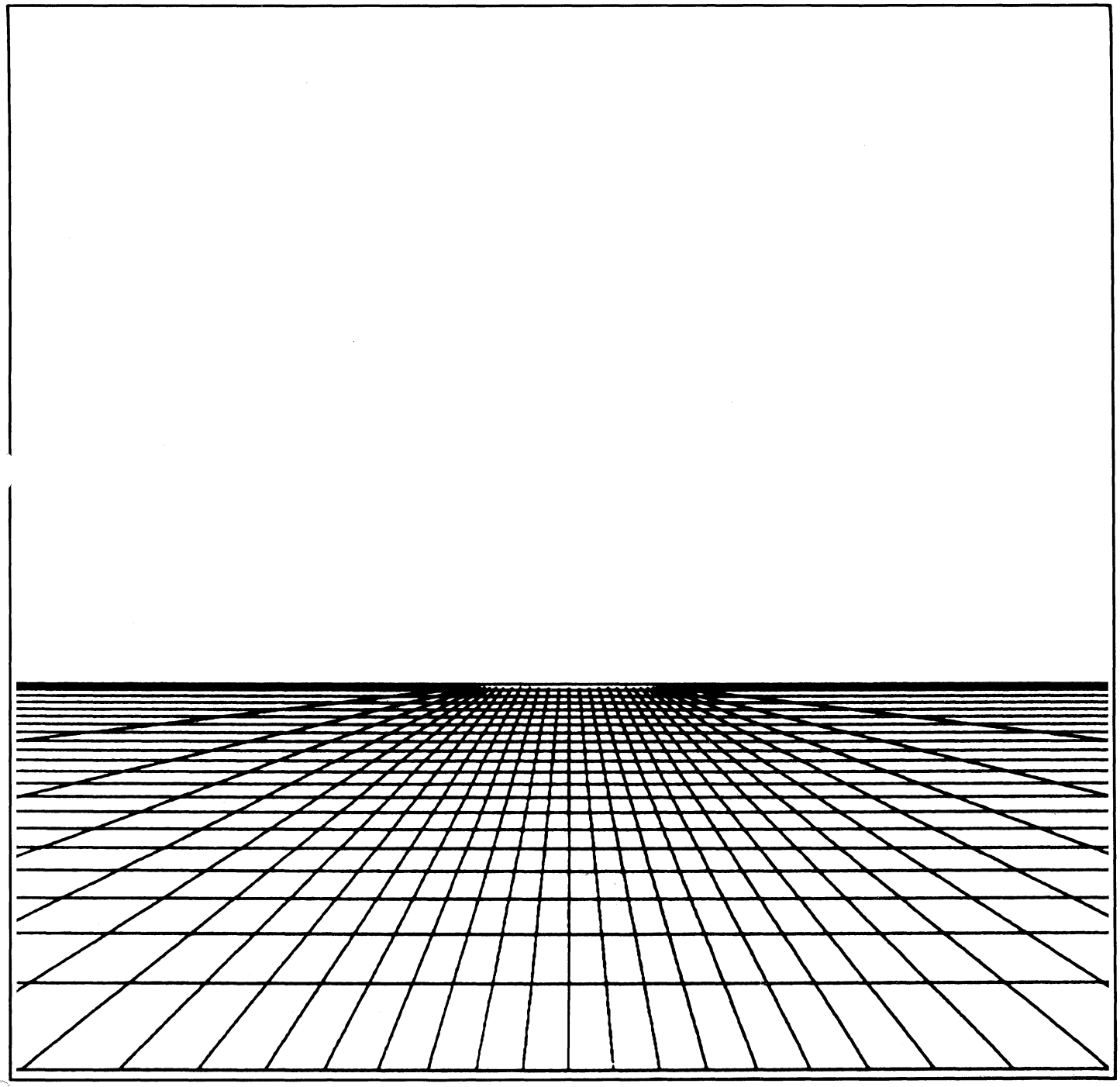
THE 4050 SERIES
APPLICATIONS LIBRARY NEWSLETTER

VOL. 5 NO. 3
SUMMER/ FALL 1981

Tekniques



Tektronix®
COMMITTED TO EXCELLENCE

Tekniques

In This Issue

Color Enhanced Dynamic Graphics 2
Low-Cost CADD System Communicates Numerically and Graphically 5
New PLOT 50 Digitizing Software Encodes Graphic Images 11
Graphic Systems Workshops 15
450,000 Byte Cartridge Tape 15
Input/Output 17
Editor's Note 18
Programming Tips 19
BASIC Bits 26
Library Consolidation 28
New Abstracts 29

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.

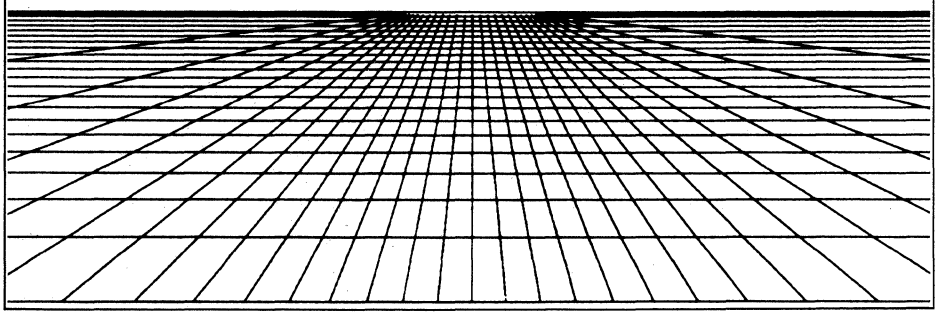
Publishing Manager	Ben Buisman
Managing Editor	Patricia Kelley
Technical Editor	Dan Taylor
Graphic Design	John Ellis
Circulation	Rory Gugliotta

Copyright © 1981, Tektronix, Inc.
All rights reserved.

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).

Color Enhanced Dynamic Graphics:

A New Option; A New Technology



By Thomas W. Woody
Tektronix, Inc.
Beaverton, OR

Dynamic Graphics* brought powerful interactivity to the TEKTRONIX 4054 Desktop Computer. Now the 4054 has taken another technological leap: Color Enhanced Dynamic Graphics. This new option allows refreshed objects to be moved around on the graphics screen in a contrasting orange color. Enhancing Dynamic Graphics with color clearly distinguishes refreshed objects from stored objects which are in the familiar green color. And displaying a refreshed image over a stored image achieves a third color — very useful for highlighting text or objects.

Packaging a desktop computer with very high resolution (4096 x 4096 addressable points), Dynamic Graphics, and a second color required a major technological advancement. To understand this technology, let's take a look at the Direct View Storage Tube (DVST), the Write-Thru operation, and the Over-Write features.

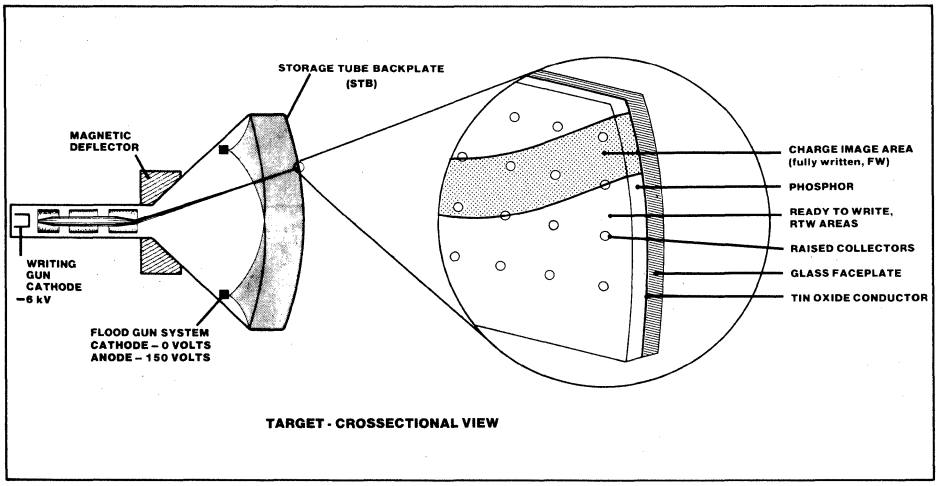


Figure 1. Operation of the direct view storage tube. Flood electrons continuously bombard the phosphor. In the absence of writing-gun electrons, flood electrons maintain the phosphor at the flood-gun cathode potential of 0 volts. When writing-gun electrons strike the phosphor, the phosphor is charged past a threshold to maintain a stored charge image.

***Miki Tokola, "Refreshed Graphic Objects on the 4054," TEKniques Vol. 3 No. 5. pp. 1-3.**

DVST Operation

A Direct View Storage Tube (DVST) operates on well known principles that are briefly reviewed here. Figure 1 shows a simplified DVST structure. A DVST consists of a writing gun that operates at a negative potential with respect to the target. The phosphor target resides over a transparent electrical conductor, and a dot array of conductive collectors is arranged in a uniform fashion across the surface of the conductor. Near the back of the funnel wall of the cathode ray tube (CRT) is an array of flood guns in which the cathode is grounded and the anode is at a positive potential.

When the DVST is in normal use for displaying stored information, the writing gun is used to scan the target. As the writing-gun beam scans the target, a charge is left on the phosphor dielectric.

Because the charge image areas are at a positive potential relative to the flood-gun cathode, the flood electrons strike the target in the written areas and cause the phosphor to luminesce. All of the areas that were not written are maintained at an average potential much lower than the written areas. The areas written up are called fully-written, and the unwritten areas are known as ready-to-write. Figure 1 shows a magnified view of the target structure.

To erase the target and prepare it for new information, a pulse is applied to the STB, which capacitively couples the pulse to the phosphor. This pulse, coupled with the flood-gun electrons, uniformly fully writes the phosphor and eliminates the ability to resolve between former ready-to-write and fully-written areas. The balance of the erase restores the entire target to a ready-to-write state.

This simple description shows how a DVST is used to display data in the stored mode. Once the target is stored, no further refreshing or rewriting of the information is necessary.

Write-Thru, How it Works

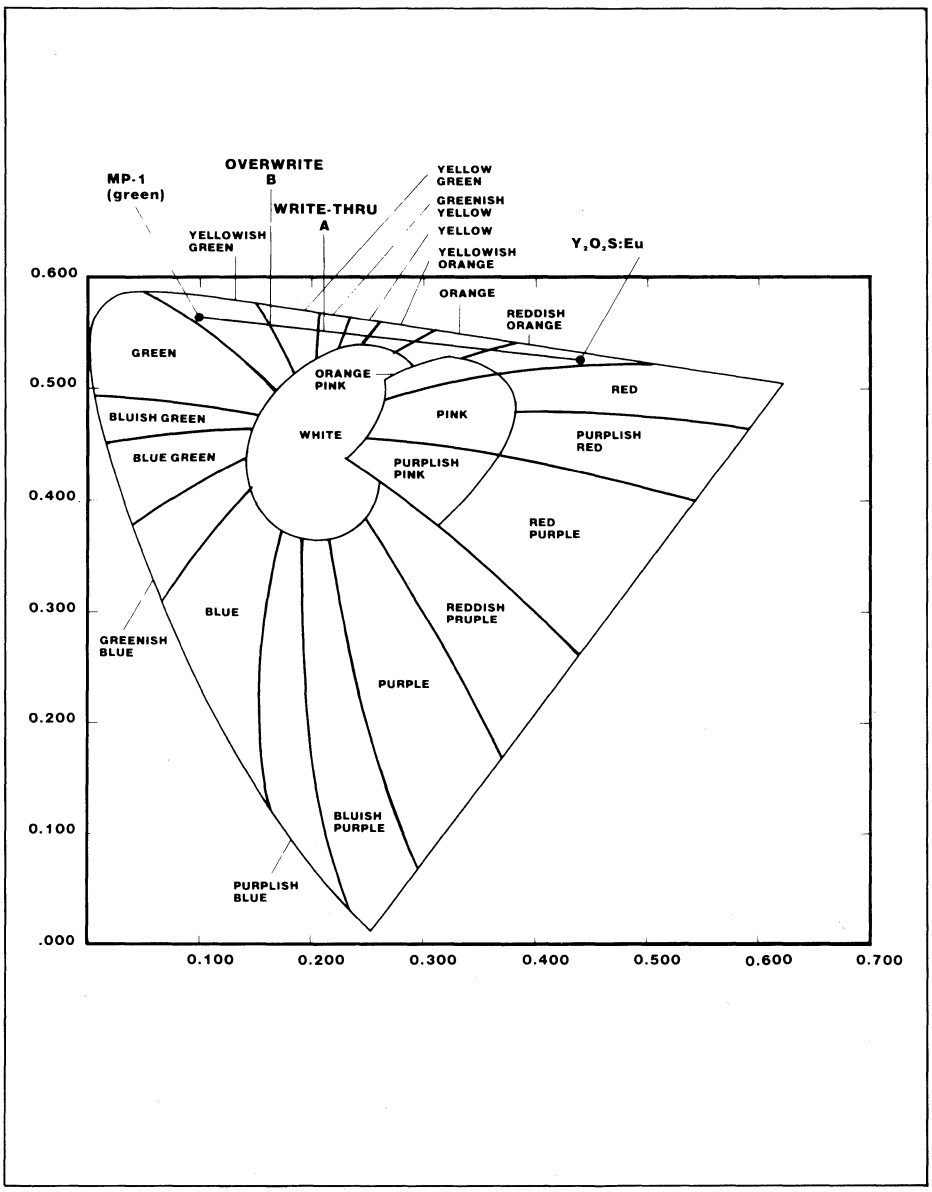
Because of its basic nature, the DVST must be erased every time a new page of information is to be displayed. Although the erasure time is not long (about 1 to 1.5 seconds), if the information is to be changed only slightly, rewriting a complete page of data can take several seconds depending on the throughput rate of the system.

The need for interactivity and faster response for DVSTs was recognized, and a technique called write-thru was devised. Write-thru displays information on the phosphor target at a current density lower than that required to charge or write the phosphor target. If the repetition rate is adjusted properly along with beam current, the refreshed image will luminesce but will not store. This feature is already available in many DVST displays. Options allow the displays to refresh a number of vectors at an average vector length of 1 cm. One drawback to this technique is that the refreshed information appears at approximately the same brightness as the stored information and also at the same color. As a result, write-thru can sometimes confuse the observer if the refreshed information overlaps the stored information, as in the case of multilayer-circuit-board design or facilities design.

Color Write-Thru Operation

One way to overcome this problem is to present the refreshed data in a different color. The display is then much more viewable, and information is more discernible. The color write-thru DVST described here uses a greenish-yellow color as denoted on the 1976 CIE-UCS chromaticity diagram (see Figure 2, point A).

The method used to obtain the greenish-yellow color was fairly straightforward. A simple mixture of red phosphor with the green produced a write-thru color as noted previously. The color obtained is dependent on the proportion of each phosphor. As shown in Figure 2, the line represents this operating area from the green to the red phosphor. The two reds used most frequently are yttrium oxysul-



1976 CIE-UCS Chromaticity Diagram.

fide and yttrium oxide, which have excellent high-voltage efficiency characteristics.

Determining how much red phosphor to add was subject of another, separate study. Color shift is defined as the linear distance that the write-thru information is shifted from the stored information along the line shown in Figure 2. The minimum discernible color shift for high view comprehension level required that the red phosphor be 40% to 60% of the target by volume. This phosphor mixture reduced green stored luminance to about 4 foot lamberts. Because of this decrease in performance, a new tube envelope and flood-gun design were used, which helped to increase the flood-gun current density to the target and thereby increase the stored luminance. In addition to the use of a new funnel, the phosphor materials were processed in such a way as to increase the nominal operating point and thereby obtain additional luminance output.

Another by-product of phosphor processing also enhances viewability. Normally, a mixture of phosphor used as a storage target would also present the stored information in the resultant mixed color. However, the processing forms a dead layer on the surface of the red phosphor that causes the red phosphor to not luminesce until it reaches about 400 volts. Because the operating point of the target is less than 300 volts, there is a 100-volt margin before the red phosphor begins to luminesce. This shallow dead layer minimizes the loss in luminance when the writing-gun brightness is achieved with the greatest overall color shift.

These improvements resulted in a new DVST that retains and improves upon the viewability of previous DVSTs. Stored luminance was increased by 60% as a result of the greater flood-gun current density and operating point. The predominant feature is the improvement in write-thru viewability by the addition of a red phosphor.

"Over-Write"

A feature that was not available in the single-color display is called "over-write" rather than write-thru. If some data is stored, and then the same image is refreshed on top of the stored data, another viewable color is seen halfway between green and normal write-thru. This color is shown as point B on Figure 2. This feature is called over-write. With this feature, one can obtain two levels of color shift that are discernible from each other. This feature can easily be used to highlight information whether graphic or alphanumeric.

With proper software, the color write-thru and over-write features expand the usefulness of the DVST display. While still retaining its inherent low cost, high resolution, and flicker-free performance, the display can be much more interactive with the added dynamics capability.

Factory Installation

Because the Color Enhanced Dynamic Graphics is a new technology, an existing 4054 cannot be field upgraded. The different CRT, electronics, and mechanical packaging require the Color Enhanced Dynamic Graphics to be a factory installed option.

To learn more about this exciting option, contact your local Tektronix Sales Engineer.

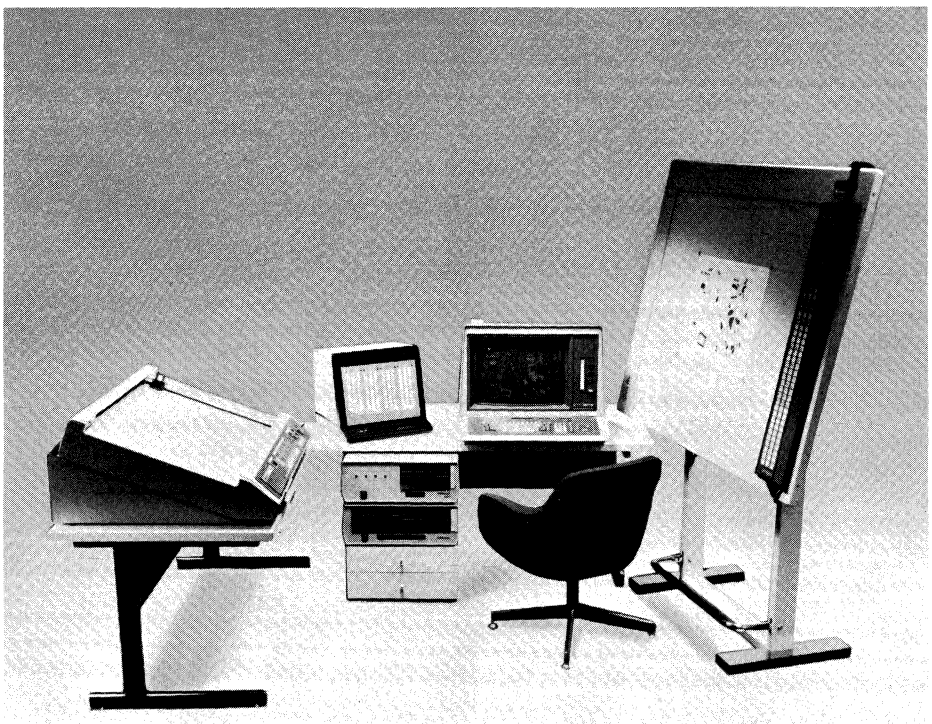
Low-Cost CADD System Communicates Numerically and Graphically

by Patricia Kelley
TEKniques Staff

Developing floor plans and building elevations; integrating lighting, heating, ventilation, plumbing and wiring; tracking materials; computing floor area; estimating costs; and generally organizing the multitude of details for a commercial or industrial building requires an architect/designer or facilities manager to spend immense amounts of time drawing, estimating and calculating. While Computer Aided Design and Drafting (CADD) has been around for quite a few years to aid designers, few systems have been tailored specifically for architects/facilities applications. System prices have generally been out of reach. Furthermore, most systems required an operator well versed in computers. Arrigoni Computer Graphics is changing that.

Seeking better ways to control all the tasks and responsibilities associated with his firm's work in commercial and industrial lighting design, David Arrigoni turned to a desktop computer to implement a CADD System for his architectural and electrical design needs. He then worked with programmers with architectural backgrounds to create software that communicated in designers' terms. What resulted was a complete, low-cost CADD system so powerful they decided to extend it and share it — a package created by designers for designers. Based around this unique system David Arrigoni formed Arrigoni Computer Graphics.

Because the TEKTRONIX 4050 Series Desktop Computer possesses the high resolution designers require, plus standalone intelligence, at a reasonable cost, it forms the core of the hardware. Coupled to the 4050 is the unique Arrigoni Z-80 microprocessor-based Touch Control Station™, the TEKTRONIX 4907 File Manager and 4663 Plotter, and a digitizing tablet with a MENU BOARD™. The TOUCH 'N DRAW™ software combines with the hardware to create a design and drawing production system.



Cover Photo: Arrigoni Computer Graphics' TOUCH N' DRAW™ integrates their unique Touch Control Station™ and MENU BOARD™ with the TEKTRONIX 4050 Graphic System, 4663 Plotter and 4907 File Manager for low cost, time saving design station. (Courtesy of Arrigoni Computer Graphics.)

The configuration is an electronic extension of normal drafting tools. Arrigoni Vice President Robert Aston analogized: "The digitizer is the drafting table, the cursor is the drawing implement, the disk is a sheet of paper, and the plotter produces the final copy — another pen in hand. The 4050, Touch Control Station™ and TOUCH 'N DRAW™ software perform the details of drafting leaving the architect or engineer free to create." Aston pointed out that the designer treats the disk as he would a sheet of paper. In a normal design environment, one person doesn't do everything. Thus, a disk may be passed from design center to design center just as he would a sheet of paper.

The designer communicates to the system through the Touch Control Station™, using pre-prepared graphic menus. There is no need to translate ideas to alphanumerics, no need to learn a

command language, no need for skilled typing. A designer with a computer background is not required. The architect, draftsman or engineer simply points to a menu item on the Touch Control Station™ (or alternatively, to the MENU BOARD™) to enter a command. According to David Arrigoni, three days' training on the system at the customer site will get the user to the point of developing quality drawings.

TOUCH 'N DRAW™ is structured according to normal architectural drafting phases. The plans may be drawn a layer at a time, analagous to pieces of "see-through" paper, one on top of another. For example, one layer for the basic wall, door and window outline, one for electrical information, one for plumbing details, one for textures, and so forth. A unique global editing feature enables the designer to move all those items within a series of layers, i.e., to move a wall and bring along all the doors, windows, electrical outlets, plumbing, etc; or, he may opt to move an individual item within one layer.

Architectural Floorplan Program

Complete, precise architectural floor plans are produced using this program. The designer defines his plan by touching the menu symbol which represents the desired design detail (Figure 1). He may position the wall, window, etc., "graphically" by digitizing its position points on the tablet, or at the screen. The exact size or length of an item, e.g., the specific diameter of a circle, or specific length of a wall, can be entered "numerically." The program automatically scales this input to the size of the drawing. The first point entered is for beginning, second point is for direction, and a number is entered from the keyboard for actual size.

Walls may be snapped into perfect horizontal or vertical alignment (an electronic T-square). Over 250 special functions, including noodling (Figure 2), door and window insertion, variable wall thicknesses, dimenisioning and curve fitting may be used. The designer may select from a variety of windows, doors and other architectural symbols. He may request multiples of that symbol, for instance, ask for six windows positioned at five-foot intervals along a wall.

As the design is created, it is displayed on the screen. Once completed, it may be stored on disk. Existing floor plans may be easily modified and saved allowing the plans to keep pace with architec-

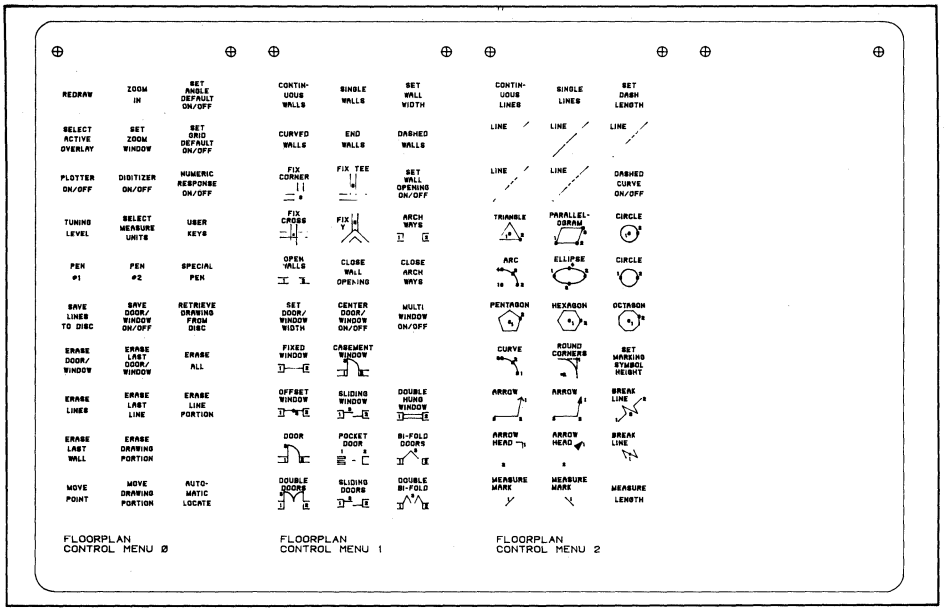


Figure 1. Designers communicate by touching an item on the acetate menu placed on the Touch Control Station™ (or alternatively on the MENU BOARD™ placed on the tablet).

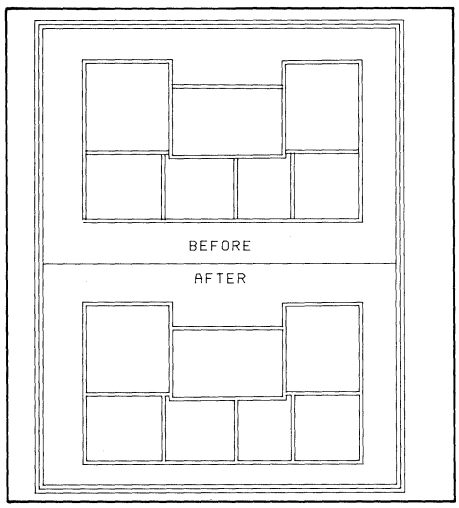


Figure 2. Special functions include noodling.

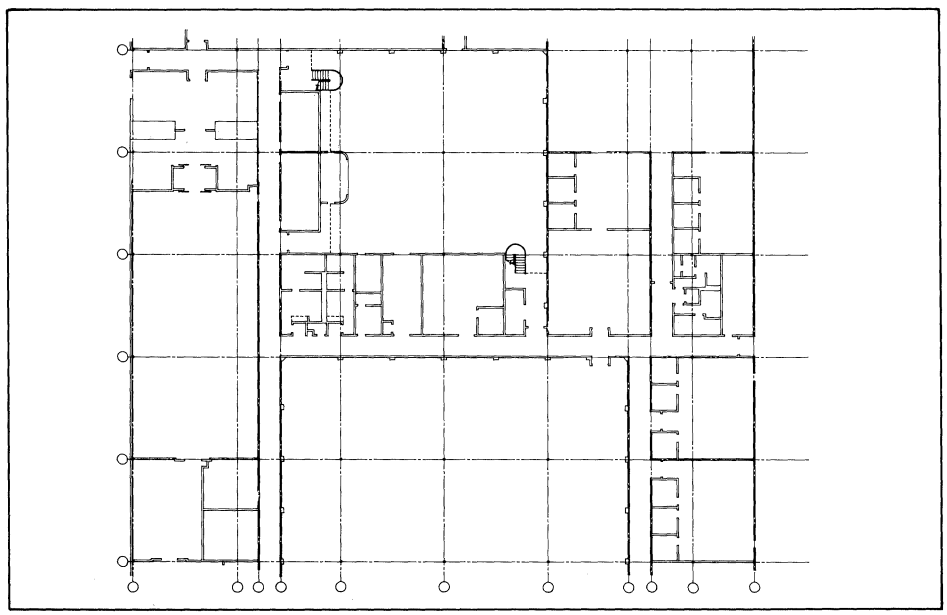


Figure 3. Completed design using the Architectural Floorplan Program may be archived and changed as facilities are expanded or modified.

tural changes. The floor plan can be archived and changed when facilities are expanded or modified.

General Drawing Program

Single line work, as well as very sophisticated work, may be accomplished using this program. Details may be added to the architectural floorplan, new drawings, site plans or general layout work done. Curve fitting, dimensioning, dot-dash, polygons, circles and mirroring are just a few of the functions. Unique relocatable objects may be created, saved and re-used. For instance, an architect may design a bathroom for an apartment complex which will be the same in 30 units. He can design it once, then place it 30 times. Or, where entire floors are similar, he could design the first floor, make changes, copy it to the second, and so on. A layout may be reversed and repeated (mirroring). This program removes the designer from routine and repetitive use of rulers and templates, thus allowing him to create and design.

Symbol Placement Program

Continuing the analogy, Aston referred to the symbols as a sophisticated template. TOUCH 'N DRAW™ provides 1000 symbols — trees, landscape, furniture, people, air duct grills, electrical symbols and so on for almost anything that can be visualized. The designer is freed from templates and press-on transfers. Using this program, the plans defined through the previous two programs are enhanced. A special package supports the 4054 dynamic graphics option enabling the designer to dynamically move symbols around rapidly and to the appropriate drawing scale. Symbols can be repeated along a line with several options for automatic spacing as well as repeated placement within an area.

Menu Generation Program

This program complements the Symbol Placement Program allowing the designer to create his own library of symbols. He traces a symbol or object on the digitizing tablet or creates it on the graphic screen. Using this program, new acetate menus also may be plotted then placed on the Touch Control Station™ or MENU BOARD™ at the digitizing tablet.

Texturing Program

Replacing press-on texture transfers, this program contains a wide range of area and line textures; for example, bricks,

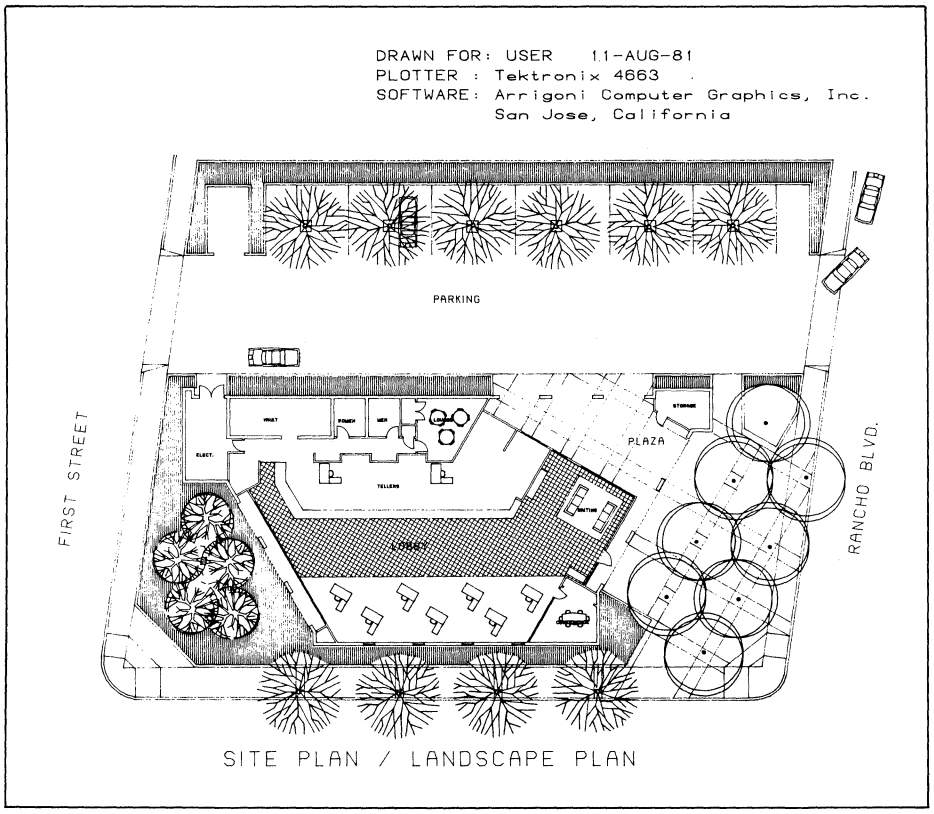


Figure 4. Curve fitting, dimensioning, polygons, circles and mirroring are just a few of the functions of the General Drawing Program.

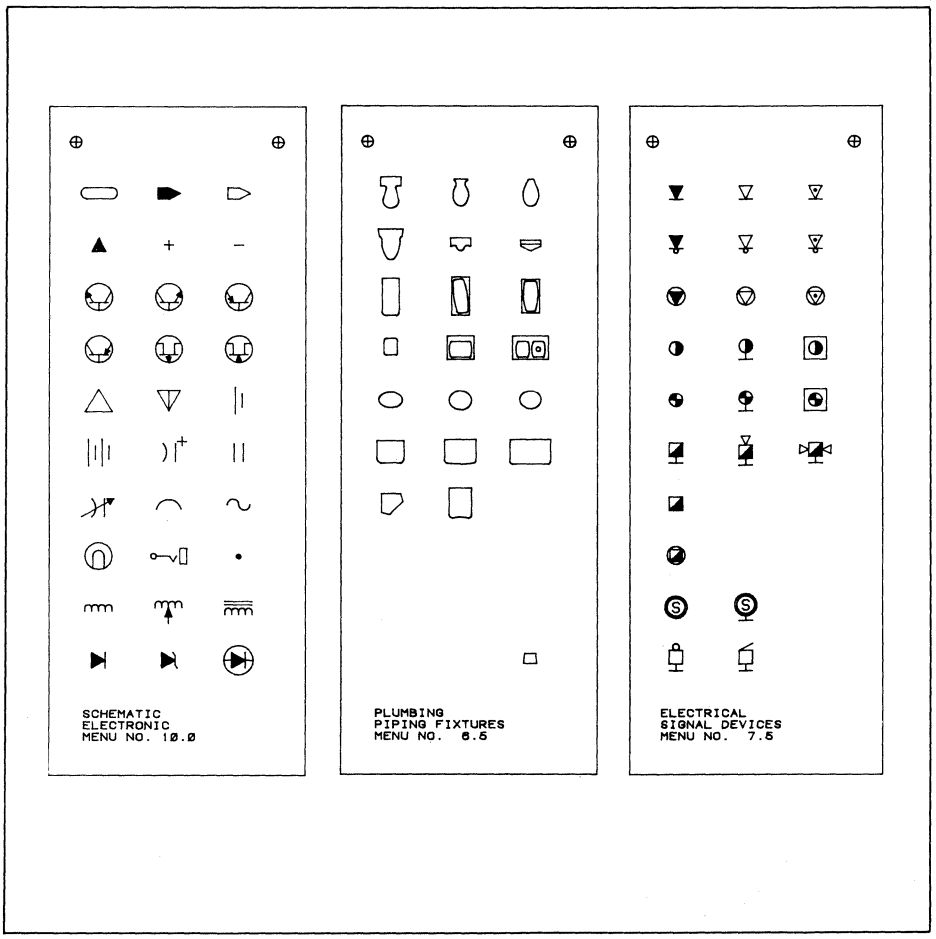


Figure 5. Using the symbols provided (or he can design his own), the designer is freed from templates and press-on transfers.

concrete, shrubbery, insulation, shingles, flooring, and simple crosshatching. Any area or object may be textured, even objects within objects — the doughnut effect. When dealing with overlaid objects, the first object selected is textured while the next selected object within (or overlapping) the first is ignored.

The densities of the texturing patterns may readily be changed and the patterns rotated or shifted.

Additionally, the program will calculate areas or lengths of line that are textured providing the ability to automatically measure and cost textured areas. For example, the linear feet of eight-foot stud walls with insulation or the number of square yards of carpet would be automatically calculated and saved for costing at a later time.

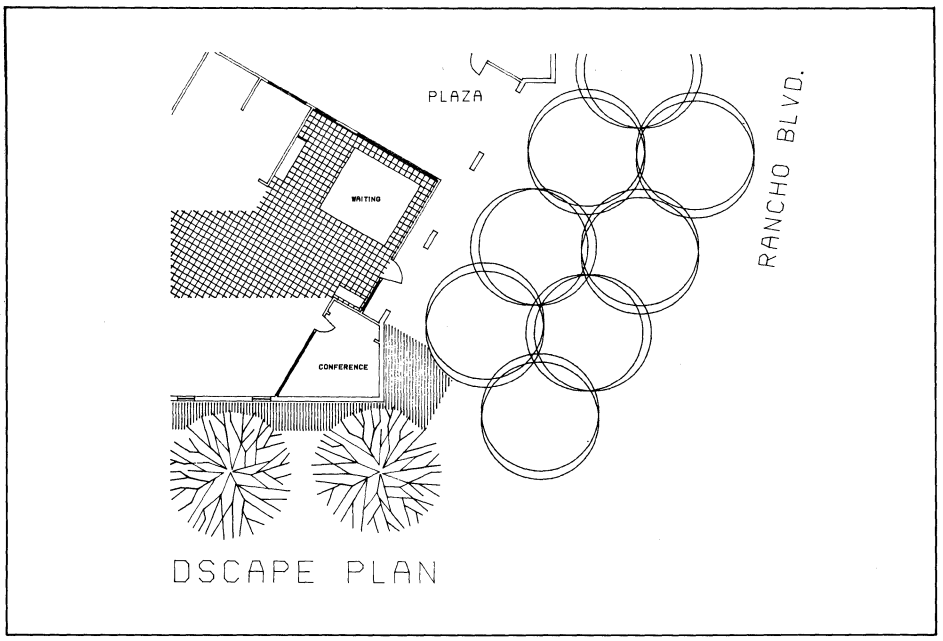


Figure 6. Any area may be textured, even objects within objects.

Annotation Program

As in all graphic applications, the text accompanying the drawings carries a significant amount of explanatory information. Therefore, seven fonts are provided: Standard Leroy, Helvetica-Medium, Ero-Style, British Inserat, Prisma, Aries and Clarendon. The designer may also design his own letters. Characters are stroked and may be sized, rotated, and placed anywhere on the drawing.

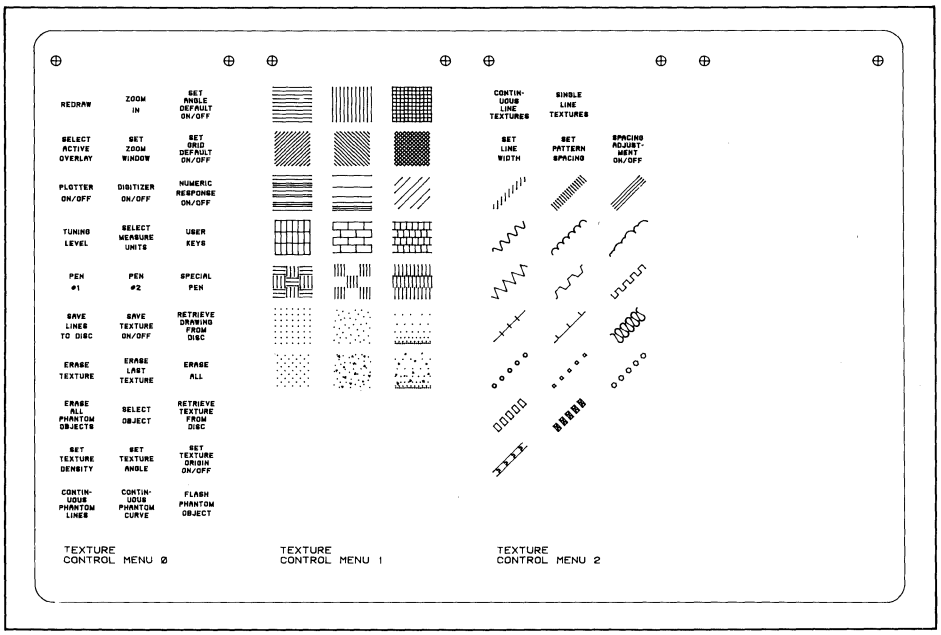


Figure 7. A wide range of area and line textures are available through the Texturing Program.

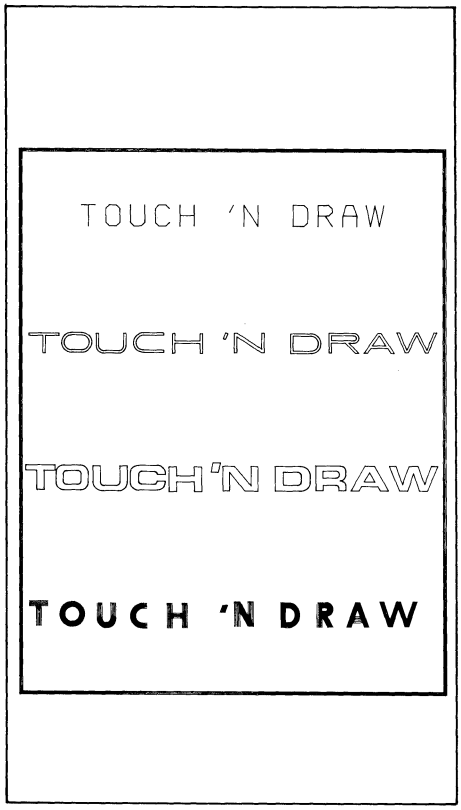


Figure 8. The Annotation Program includes seven fonts, four of the styles are shown here.

Area/Line Measurement Program

This electronic planimeter allows the designer to identify areas for calculation even if they weren't textured. For example, he could trace each room in a building and calculate the total square feet as well as that for each room. Or, he could outline a roofline and cost it. The designer may also identify a number of closed objects or polygons and add them together for area calculation.

After creating one or more closed objects or shapes, such as rooms or mapping boundaries, the designer can request the program to calculate the area or perimeter of each selected shape and draw the resulting numeric data at any location "touched" on the drawing. By naming each object, a table of areas and perimeters can be produced.

If selected, the table of areas and perimeters may be sent to the Cost Estimating Program.

Cost Estimating Program

This program counts the number of symbols on the drawing disk, the lengths of lines and textured areas, and then writes this information on the screen. The designer may call on a master material/cost file (or enter the labor and material cost for each) and the program will extend

and accumulate the material and labor cost for the entire project. For example, if the design contains 500 standard electrical receptacles, the designer can enter a cost of $153 per 1000 receptacles and the program will automatically calculate the cost for 500.

Updating a master costing file is simple and quick. The designer enters the current cost of an item and the program automatically updates the master costing file. This feature may be employed to generate a series of cost estimates reflecting different bids. Cost estimating is more accurate, more efficient and very quickly done.

Production Plot Program

Final drawings are generated through this program. It is extremely flexible, allowing the designer to reproduce an entire drawing or only a specified part. He may overlay an infinite number of drawings in any sequence. He might ask for an architectural floor plan with or without texturing, annotation, symbols or details. Or he could include a site plan in the corner of a drawing.

After previewing the plot on the screen, it may be plotted in any scale, resulting in an original drawing then readily duplicated in a blueprint machine. He can also communicate this information to another computer using the 4050 Option 1.

Utilities Programs

Ten utility programs for the designer take the drudgery out of using computer equipment through simple "yes-no" questions. The designer may copy drawings from disk to disk, or onto the same disk, allowing use of one drawing as a base to make another drawing.

Special Applications

Arrigoni Computer Graphics has also tailored versions of TOUCH 'N DRAW™ for specific disciplines and more programs are in the works. "They won't do any more than the basic CADD system, but they do it more efficiently for that particular trade or application," commented David Arrigoni.

An example of a special application package is the Electrical/Electronics Schematics package. It has combined standard elements of the system to provide a unique package for electrical engineers producing schematics drawings for electrical and electronic circuits.

MASTER COST INVENTORY LISTING — DATE: 21-JUL-81

ITEM	DESCRIPTION	MATERIAL $	UNITS	CONTG.	LABOR HR.	UNITS	CONTG.	RATE
1	DESK 21-001	289.50	1	0	0.00	0	0	0.00
2	DESK - TYPING TABLE ON LEFT 21-002	345.50	1	0	0.00	0	0	0.00
3	DESK - TYPING TABLE ON RIGHT 21-003	345.50	1	0	0.00	0	0	0.00
4	BOARD ROOM TABLE - 4' X 10' 21-004	567.00	1	0	0.00	0	0	0.00
5	BOARD ROOM TABLE - 4' X 8' 21-005	456.76	1	0	0.00	0	0	0.00
6	BOARD ROOM TABLE - 3' X 6' 21-006	5280.35	1	8	0.00	0	0	0.00
7	BOARD ROOM TABLE - 4' X 10 ' - CHAIRS 21-007	980.00	1	0	0.00	0	0	0.00
8	BOARD ROOM TABLE - 4' X 8' - CHAIRS 21-008	567.80	1	0	0.00	0	0	0.00
9	BOARD ROOM TABLE - 3' X 6 ' - CHAIRS 21-009	768.90	1	0	0.00	0	0	0.00
10	EXECUTIVE TABLE - 4' X 10' 21-010	789.90	1	0	0.00	0	0	0.00
11	EXECUTIVE TABLE - 4' X 8' 21-011	987.00	1	0	0.00	0	0	0.00
12	EXECUTIVE TABLE - 3' X 6' 21-012	678.98	1	0	0.00	0	0	0.00
13	EXECUTIVE TABLE 4' X10' - CHAIRS 21-013	890.00	1	0	0.00	0	0	0.00
14	EXECUTIVE TABLE - 4' X 8' - CHAIRS 21-014	1257.00	1	0	0.00	0	0	0.00
15	EXECUTIVE TABLE - 3' X 6' - CHAIRS 21-015	0.00	0	0	0.00	0	0	0.00
16	CHAIR - DINING 21-016	64.50	1	0	0.00	0	0	0.00
17	CHAIR 21-017	65.45	1	0	0.00	0	0	0.00
18	BLANK 2	0.00	0	0	0.00	0	0	0.00
19	EXECUTIVE CHAIR 21-019	0.00	0	0	0.00	0	0	0.00
20	ARM CHAIR 21-020	0.00	0	0	0.00	0	0	0.00
21	FOLDING CHAIR 21-021	0.00	0	0	0.00	0	0	0.00
22	DESK CLUSTER 21-022	1275.50	1	0	0.00	0	0	0.00
23	DESK CLUSTER 2 21-023	1275.50	1	0	0.00	0	0	0.00
24	BLANK 3 21-024	0.00	1	0	0.00	0	0	0.00
25	PARTITION - 5' 21-025-AA	189.98	2	0	0.00	0	0	0.00
26	PARTITION - 8' 21-026	235.65	1	0	0.00	0	0	0.00
27	PARTITION CURVE - 5' 21-027	235.65	1	0	0.00	0	0	0.00
28	PARTITION CURVE - 5' RIGHT 21-028	235.65	1	0	0.00	0	0	0.00
29	SOFA AA-FF-04	890.50	1	0	0.00	0	0	0.00
30	TABLE - 36" SQUARE AA-123-BB	189.98	1	0	0.00	0	0	0.00
31	PARTITION CURVED - 5' RADIUS 21-008	567.80	1	0	0.00	0	0	0.00
32	CHAIR - DINING 21-016	163.35	1	0	0.00	0	0	0.00

JOB: 0091.0 DYSAN MFG. BLDG. 4 — PAGE NO. 1
SHEET: E3.2 FIRST FLOOR - ROOM 102 — DATE: 21-JUL-81

ITEM	DESCRIPTION	QUANTITY	MATERIAL $	UNITS	CONTG.	TOTAL
1	DESK - TYPING TABLE ON LEFT 21-002	17	345.50	1	0	5873.50
2	DESK - TYPING TABLE ON RIGHT 21-003	16	345.50	1	0	5520.00
3	PARTITION - 5' 21-025-AA	18	189.98	2	0	2279.76
4	PARTITION - 8' 21-026	2	235.65	1	0	471.30
5	BOARD ROOM TABLE - 4' X 10' 21-004	2	567.00	1	0	1134.00
6	PARTITION - CURVED 5' RADIUS 21-008	4	567.80	1	0	2271.20
7	DESK 21-001	2	289.50	1	0	579.00
8	SOFA AA-FF-04	1	890.50	1	0	890.50
9	CHAIR - DINING 21-016	5	64.50	1	0	322.50
10	TABLE - 36" SQUARE AA-123-BB	2	189.98	1	0	379.96
11	DESK CLUSTER 21-022	5	1275.50	1	0	6377.50
12	BOARD ROOM TABLE - 4' X 10' 21-004	1	567.00	1	0	567.00
TOTAL MATERIAL COST						26674.22
TOTAL SHEET COST						26674.22

JOB: 0091.0 DYSAN MFG. BLDG. 4 — PAGE NO. 1
SHEET: E3.2 FIRST FLOOR ROOM 101 — DATE: 21-JUL-81

ITEM	DESCRIPTION	QUANTITY	MATERIAL $	UNITS	CONTG.	TOTAL
1	DESK - TYPING TABLE ON LEFT 21-002	14	345.50	1	0	4837.00
2	DESK - TYPING TABLE ON RIGHT 21-003	9	345.50	1	0	3109.50
3	CHAIR 21-0021	4	89.50	1	0	358.00
TOTAL MATERIAL COST						8304.50
TOTAL SHEET COST						8304.50

Figure 9. Cost estimating is accurate, efficient and quickly done using the Cost Estimating Program.

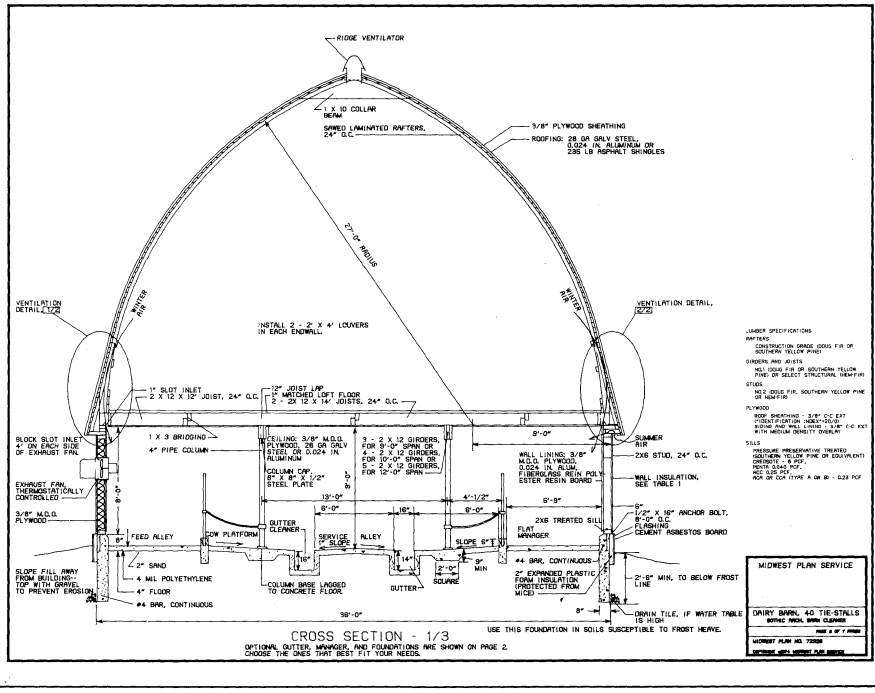


Figure 10. The Production Plot Program produces the final drawing.

The success of the Arrigoni Computer Graphics TOUCH 'N DRAW™ can be attributed to the excellent user interface. The designer simply auto loads a tape. The program prompts for input and 90% of the questions require "yes-no" responses. Prompts and questions use terminology familiar to designers or draftsmen and purposely avoid "computerese." Visual feedback through messages and lights on the Touch Control Station™ keep the user informed of where he is. If the designer has to leave his system for a while, the status lights let him know where he was when he left.

Although TOUCH 'N DRAW™ was designed to open up the opportunity for small to medium sized firms to use CADD Systems, the Systems application extends to the largest architectural firms and facilitates planning and management for Fortune 1000 companies. Thus, the System can handle extremely large projects; one customer is designing buildings totaling over three million square feet.

The total system (including the hardware, software, and training) may be purchased; or it may be leased at a monthly cost approximating that of an average draftsman. David Richards, President of Arrigoni Computer Graphics, indicates that productivity improvements of 3-5 times versus manual drafting are typical with this system.

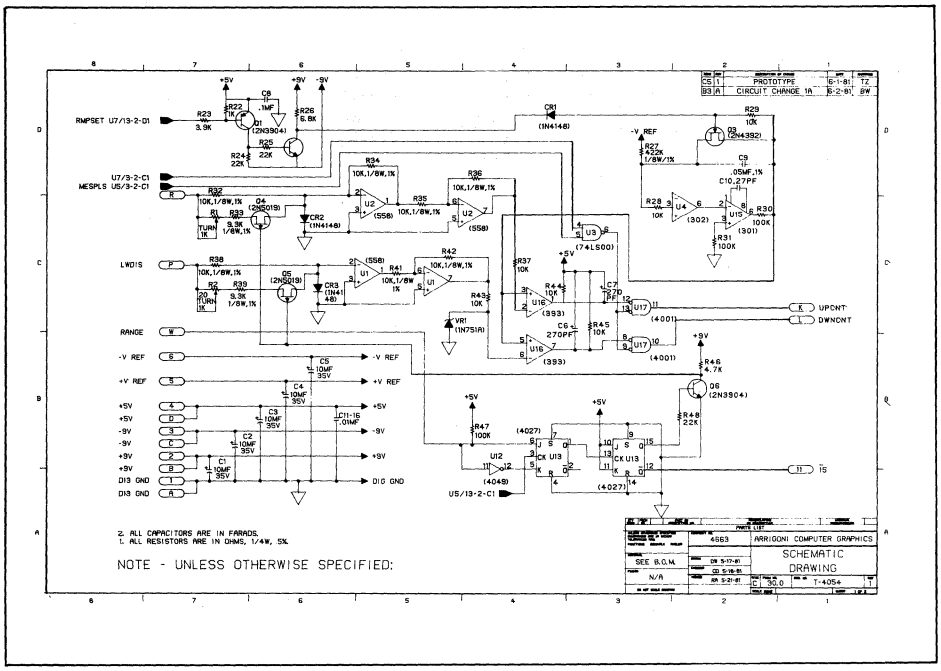
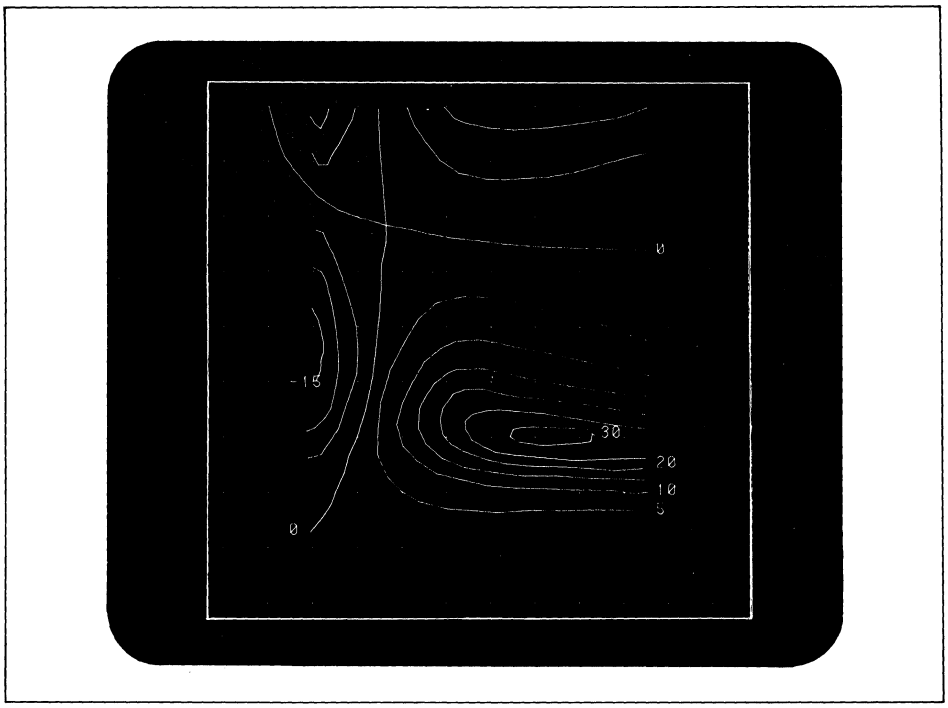


Figure 11. Electrical engineers producing schematic drawings for electrical and electronic circuits might choose the special Electrical/Electronics Schematics package.

If you have further questions about the Arrigoni TOUCH 'N DRAW™ CADD System, contact Arrigoni Computer Graphics at 231 O'Conner Drive, San Jose, CA 95128 or phone (408) 286-2350.

Note: As a service, TEKniques may publish notices of software, hardware and services from suppliers other than Tektronix. No evaluation or endorsement by Tektronix, Inc. of these products or services is implied by such publication. Tektronix, Inc., expressly disclaims any obligation of warranty or support.



Digitized curves of constant nuclear polarization in nuclear D(d,n)He³ reactions. A point grid adds visual reference points.

New PLOT 50 Digitizing Software Encodes Graphic Images

by Paul J. Kristof
Tektronix, Inc.
Wilsonville, OR

4052D07 PLOT 50 Interactive Digitizing solves the data entry problem for a variety of applications. By coupling the digitizing tablet with the graphic display and computational power of the 4052 and 4054 desktop computers, disk-based Interactive Digitizing efficiently and conveniently encodes graphic images for computer analysis or processing. Stripcharts, graphs, maps, X-ray photographs, electromicrographs, and other images are readily translated into machine digestible form. Immediate preview and editing help ensure the transmittal of "clean," error-free data. Mathematical functions calculate area, perimeter, centroids or line lengths. Document labeling and text may also be transmitted.

Interactive Digitizing supports PLOT 50 Standard File Format[1] and Graphic Exchange Format[2]. Curves encoded from stripcharts, or area and centroid values from digitized polygons may be sent to PLOT 50 Easy Graphing[3] or PLOT 50 Statistics[4] for graphing or analysis. Picture data can be sent to PLOT 50 Picture Composition[5] for inclusion in illustrations. In any case, you need not re-enter the data.

Data may be encoded in real world coordinate units and the X and Y axes may be independently scaled. For example, to digitize experimental data from a nuclear reaction, you could set X units to voltage in Mevs and Y units to reaction angle in degrees. Documents larger than the tablet surface are easily handled.

Interactive Digitizing consists of four major sections:

TUTORIAL
AUTOLOAD
ENTRY/EDIT
UTILITIES

The TUTORIAL gives you a guided tour of Interactive Digitizing. It illustrates many of the features and shows a typical sequence of activities to create a digitized data file.

The AUTOLOAD section sets up the operating characteristics of the package. Disk assignments and peripheral device selections are set or modified at this time.

The ENTRY/EDIT section creates or edits its digitized data files.

The UTILITY section applies file and system manipulation functions.

Digitized Data Files

First, we'll take a look at the format of the digitized data file.

Document features are recorded in the "digitized data file," an ordered list of objects. Coordinate data and document text is encoded in five object types:

POINT
A single (x,y) coordinate.

POLYLINE
A set of connected lines or circular arc segments.

POLYGON
A polyline in which the first and last points are the same (a closed figure).

TEXT
A set of scalable and rotatable characters.

SYMBOL
A set of points, line segments, arcs and text (not necessarily connected) that can be located, scaled and rotated as a unit.

In addition to the graphic descriptions, each object has an object identification (ID) number, and may have an object name and a set of numeric or string attributes. The system assigns the object ID number when an object is stored in the data file. This identifies the object for editing.

You may assign a thirty-six (max) character name to each object.

A powerful feature of the package is the ability to assign up to five unique attributes for each object and (optionally) have the program prompt for these attributes. The attributes are specified as numeric or alphanumeric and named. The object's name and its attributes may then be used by software that has been custom-designed to look for and process them.

[1]Standard File Format is discussed in TEKniques Vol. 4 No. 1.
[2]Graphic Exchange Format is discussed in TEKniques Vol 5 No. 2.
[3]Easy Graphing is described in TEKniques Vol. 4 Nos. 2, 3 and 4.
[4]Statistics: Disk-based Tests and Distributions is described in TEKniques Vol. 4 No. 5.
[5]Picture Composition is described in TEKniques Vol. 5 No. 1.

Thus, a data file might consist of:

- The map of Arizona encoded as **Polyline** objects, with two numeric attributes: **left county id** and **right county id**. (See figure 1.) A separate host program could then use the coordinate data and the attributes to reconstruct the counties as polygons.
- The map of Oregon showing big game management units encoded as **Polygon** objects with an object name and a numeric attribute. The name of the area was entered for the object name. The attribute, UNIT CODE, identifies each polygon. Additionally, the area of each polygon was calculated as it was digitized.

  The UTILITY program, List Data File, may be used to output the size of each unit in square miles (figure 2). The area information could also be transmitted to the PLOT 50 Statistical volumes for study.
- The contours in figure 3 were encoded as **Polylines** and **Polygons** with a sinle numeric attribute, LEVEL, to hold the elevation value for each line.

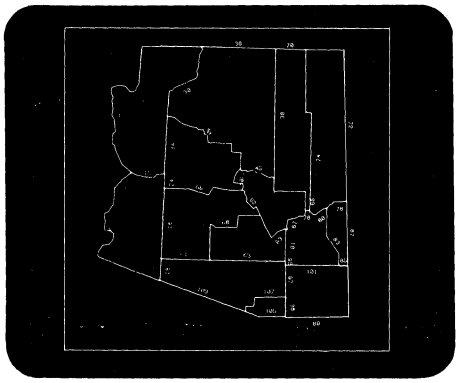


Figure 1. A map of the counties of Arizona encoded as polylines. The numbers appearing adjacent to the lines are the object ID numbers assigned by the system.

```
DATA FILE INFORMATION

Data file                    : BIGGAME
Number of objects in file    : 14
Extent   (min X, max X)      : 2 , 205
         (min Y, max Y)      : 18 , 159
Control point coordinates    : 1) 0 , 0
                               2) 0 , 240
                               3) 240 , 0

Aspect ratio (width/heisht): 1.02312513436

Attributes Defined

1   UNIT NO.     (numeric)

Object Information

1   Eid# 10]  Polygon        Name: CHETCO
    Number of points     : 38
    Extent (min X, max X): 8 , 66
           (min Y, max Y): 20 , 71
    Perimeter            : 182.008372897
    Area                 : 1773
    Centroid X,Y         : 32.7490129724 , 44.1400639218
    UNIT NO.: 27

2   Eid# 20]  Polygon        Name: SIXES
    Number of points     : 44
    Extent (min X, max X): 2 , 30
           (min Y, max Y): 53 , 123
    Perimeter            : 168.563506783
    Area                 : 1074.5
    Centroid X,Y         : 16.602916085 , 83.1729486583
    UNIT NO.: 25

3   Eid# 31]  Polygon        Name: POWERS
    Number of points     : 40
    Extent (min X, max X): 25 , 70
           (min Y, max Y): 65 , 96
    Perimeter            : 151.889451022
    Area                 : 818
    Centroid X,Y         : 46.2703748981 , 79.2726161369
    UNIT NO.: 26

4   Eid# 41]  Polygon        Name: TIOGA
    Number of points     : 43
    Extent (min X, max X): 12 , 60
           (min Y, max Y): 89 , 144
    Perimeter            : 196.457882569
    Area                 : 1691.5
    Centroid X,Y         : 37.620652281 , 116.923342201
    UNIT NO.: 24
```

Figure 2. Data file contents are easily displayed through utility LIST DATA.

```
                    INFORMATION

Data file                    : CONTOUR
Number of objects in file    : 15
Extent   (min X, max X)      : -2.38751818418,22.9898676672
         (min Y, max Y)      : 13.1786478166,112.662606778
Control point coordinates    : 1) 0,0
                               2) 0,90
                               3) 18,0

Symbol file                  : None
Current symbol selected      : None

Current mode                 : Polyline
Current object               : Present
Current object type          : Polygon
Number of points             : 49
Object ID                    : 10
Object name                  :
Number of attributes         : 1

Name  : LEVEL
Value : 1

Window coordinates (minX, maxX) : -4.84780347797,24.6501530489
                   (minY, maxY) : 4.75488881373,117.881069538

Number of objects
digitized in this session : 0
```

Figure 3. Choosing menu item INFO displays the status of your current operation.

Briefly, that is the structure of the digitized data file. Getting the image encoded from the document to a digitized data file is easy and flexible.

EDIT/ENTRY

Interactive Digitizing is menu driven for convenient function selection. The ENTRY/EDIT section is tablet menu driven. Simply locate the tablet menu anywhere on the tablet surface along with your document. To choose a function, just place the tablet cursor over the appropriate menu box and digitize. The package prompts you for additional input.

Tablet menu (figure 4) functions are grouped into seven functional blocks:

- OBJECT ENTRY
  - — mode selection
  - — object insertion control
  - — area/length/centroid calculation
  - — point entry and immediate edit
- DISPLAY CONTROL
  - — highlighting of current object
  - — redraw
  - — window selection
  - — point previewing
- EDIT CONTROL
  - — data file object sequence
  - — object
- SYMBOL CONTROL
  - — symbol selection
  - — scale and rotation setting
- ENVIRONMENTAL PARAMETERS
  - — feature selection
  - — value setting
  - — parameter display
- ALPHANUMERIC KEYBOARD
- SUPPLEMENTAL FUNCTIONS
  - — plotting
  - — setting new data file
  - — display system information
  - — extend axis
  - — HELP

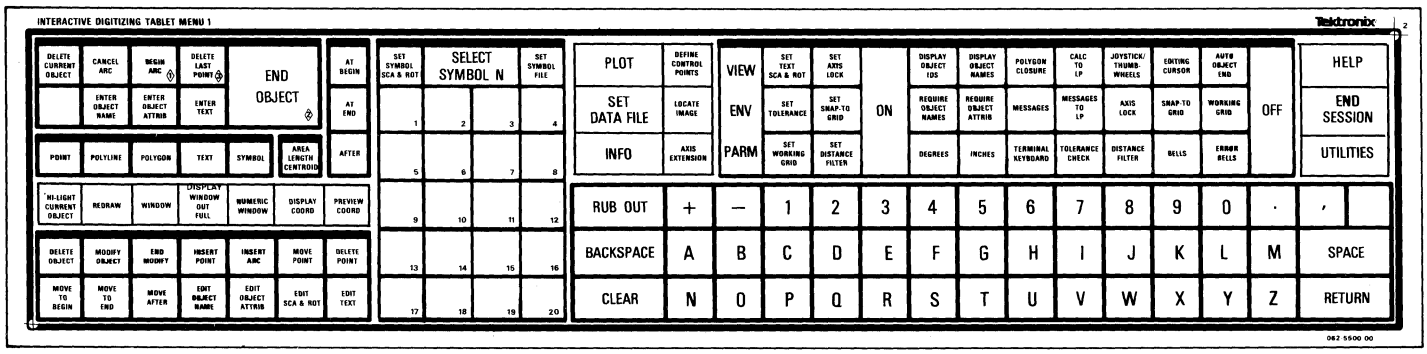


Figure 4. Tablet menu functions are grouped into seven functional blocks.

— end current session
— access UTILITIES
— redefine control points

In addition to the tablet menu, a User-Definable Key (UDK) overlay is provided (figure 5). All the EDIT functions are duplicated on the UDK overlay. By using the UDKs, the 4050 keyboard, and the thumbwheels or joystick, all editing may be performed at the 4050 with no need to reference the graphics tablet.

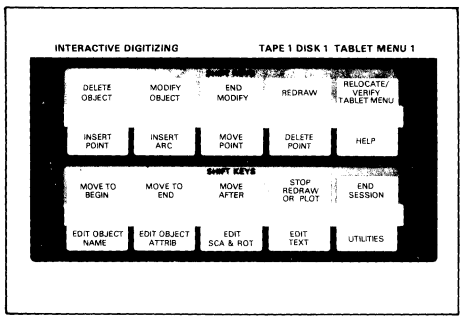


Figure 5. All the EDIT functions are duplicated on the UDK overlay.

The object entry functions of **begin arc**, **delete last point**, and **end object** may also be entered from the optional four button tablet cursor.

You may preview the data on the 4050 screen before entering it into the data file. Prompt messages appear along the left edge of the square data entry display area (figure 6). A character in refresh, above the left edge of the display area reports the current operational status of the package.

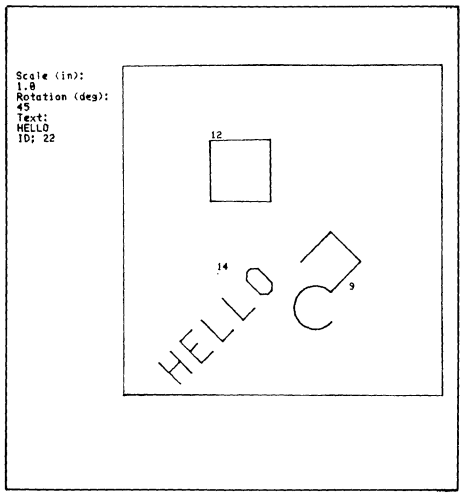


Figure 6. A blinking character above the data display area provides status information:
E – points may be digitized or tablet menu items selected
I – Input from the tablet keyboard
M – Edit (modify) mode
D – Display Coordinate
P – Preview Coordinate

Special Features

Interactive Digitizing includes a number of features to make digitizing easier, quicker, and more convenient for you. These include:

- **Immediate Editing** — While entering an object, you can delete the last digitized point or the current object, edit the object name or attributes, or respecify the scale or rotation of text and symbol objects without explicitly entering the EDIT mode. This makes input errors easy to correct.
- **Audio Feedback** — A single bell signals acceptance of a digitized point, and multiple bells signal a warning or improper input message. They may be turned on or off.
- **Selectable Viewing Windows** — You may select from three different windowing functions the document area you wish to view on the screen. Redraw functions are optimized to access only those objects intersecting the currently selected window.
- **System Configuration Files** — You only set peripheral device addresses once, or when they change. The system automatically remembers them.
- **Axis Extension** — You can easily handle documents larger than the tablet surface, such as long stripcharts. Just select two points, reposition the document, and redigitize the original two points. The system automatically adjusts the tablet to user coordinate system transformations.
- **Tablet Menu Keyboard** — By selecting the tablet menu keyboard, you can answer all ENTRY/EDIT prompts from the tablet, without returning to the terminal keyboard.
- **Area/Length/Centroid Computations** — You may compute at any time and direct the output to a line printer. The results of these computations may also be placed in the Standard Format File and Graphic Exchange File.
- **Auto Mode for Repetitive Entry** — **Auto object end** combined with **polygon closure** (automatic closure of a polygon object when the next point digitized is within a user-specified tolerance of the first point), **require object names** and **require object attributes** increases efficiency for repetitive data entry. As each object is closed, the program will prompt for its name and attributes, then automatically store it in the digitized data file. Combining the four button cursor and **auto object end** allows for repetitive entry of similar object types without having to return to the tablet menu.
- **Snap-to-Grid/Axis Lock** — You may elect to digitize points in user-specified increments, for example, 100 feet. The next point digitized is snapped to the nearest grid point with **snap-to-grid** on, and/or the nearest axis with **axis lock** on.
- **User-Specified Coordinate System** — You digitize your application coordinate units with independent scaling of the axes.
- **HELP** — **HELP** will instruct you in the operation of any menu box or User-Definable Key.

UTILITIES

Interactive Digitizing provides a full complement of utilities programs. The UTILITIES section is accessed from the 4050 keyboard. A master screen menu controls the function selection (figure 7). The User-Definable Keys interrupt any utility and return control to the start of the currently active utility, to the master selection menu, or end the current session.

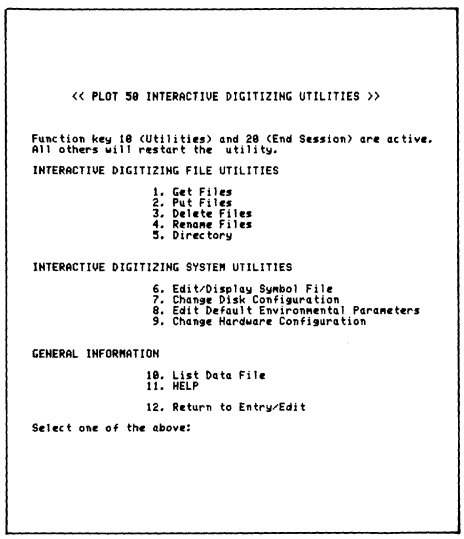


Figure 7. The UTILITIES routines are accessed through the screen menu.

Three major functional groups comprise the utilities: those dealing with data files, those which modify the current system configuration, and those providing general information.

Interactive Digitizing recognizes five separate file types:

Digitized Data — encoded document
Symbol — symbol set
GMX Picture — } PLOT 50 transmittal files
GMX FONT — }
Standard Data — }

UTILITIES programs rename, delete, and list a directory of all files. In addition, the standard functions of GET FILES and PUT FILES transfer files between mass storage devices and other PLOT 50 software products.

Symbol files (figure 8) may be created from digitized files, edited, displayed or plotted by the EDIT/DISPLAY SYMBOL FILE utility.

The current or default values for disk selections, peripheral device addresses, and environmental parameters can be changed any time by using the other UTILITIES.

The General Information utility, LIST DATA FILE, provides screen or line printer output of coordinates, attributes, and computations associated with a digitized data file.

As with the ENTRY/EDIT section, HELP is available for refresher information on how each utility operates.

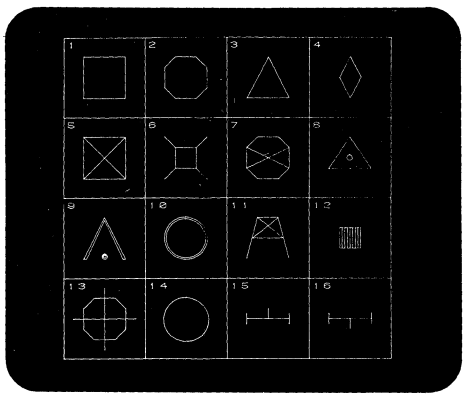


Figure 8. Symbols created using Interactive Digitizing. Symbols may be included in any digitized data file.

DATA FILES

Three major levels of access to digitized infomration is provided: GMX, Standard Data and Digitized Data files.

The information in the digitized data file consists of that associated with the file as a unit, e.g., number of objects in the file, data file extent, aspect ratio, and that associated with each specific object, e.g., number of points per object, object ID, object name, coordinates, object attributes, and computations.

All of the information in a digitized data file may be stored in the GMX file transfer format. Data file global information is recorded in the file descriptor records, and object specific information is recorded as object descriptor records and point descriptor records[7].

Object geometry and object attributes (numeric only) may be output in two separate Standard Data files (see figures 9a-9b).

In addition to the GMX and Standard Files, the digitized data file structure is fully documented in the manual for those users desiring direct access to the digitized data base.

---

[7]Refer to TEKniques Vol. 5 No. 2 for a full description of GMX Picture File format.

4052D07 PLOT 50 Interactive Digitizing is a sophisticated, yet easy-to-use system supporting a varied equipment configuration. A 4052 or 4054 with 64K of memory, a 4907 File Manager and 4956 Graphics Tablet is the minimum configuration. However, it will also support a 4907 File Manager with Option 30 or 31 (dual or triple drive units).

Editing input may be from the 4054 thumbwheels or a 4952 joystick.

Graphics output may be directed to a 4662 or 4663 Plotter; listing output may be directed to a 4641 Printer.

For more information on Interactive Digitizing, contact your local Tektronix Sales Engineer.

OBJECT ATTRIBUTE STANDARD DATA FILE

VARIABLES:

	Object Type	Centroid x	Centroid y	Length	Area	Scale	Rotation	Symbol ID	Attribute 1	Attribute n
Object-name										
OBSERVATIONS										

Variables are included only as they apply to an included object type.
For example, if symbols are not written then symbol ID is not included.

Variables

OBJECT TYPES:			
	1	Point	---
	2	Line	Length
	3	Polygon	Centroid - x, centroid - y, area, length
	4	Text	Scale, rotation
	5	Symbol	Scale, rotation, symbol ID

Figure 9a. When object attributes are transferred, an Object Attribute Standard File is the result.

OBJECT GEOMETRY STANDARD DATA FILE

VARIABLES

	Object Type	x Coord.	y Coord.	Point Type
Object-name				
OBSERVATIONS				

NOTE: No alpha-numeric variables are supported in any way. This includes alpha attributes and text description.

OBJECT TYPE: 1 = Point, 2 = Polyline, 3 = Polygon, 4 = Text, 5 = Symbol

POINT TYPE: 0 = Line Segment, 1 = ARC Start, 2 = ARC Middle

Figure 9b. When object coordinates are transferred, an Object Geometry Standard File is the result.

Graphic Systems Workshops

4050 Series Graphic Computing Systems Workshops are designed to help you get the most out of your systems. These week-long workshops are held in several locations around the country. Dates and locations for these workshops are detailed here.

The workshops teach the application of BASIC and the concepts of desktop computer graphics. The classes emphasize integrating and using TEKTRONIX 4050 Series Graphic Computing Systems with their associated peripheral equipment. The full range of Graphic Computer System equipment is available in the workshops.

The workshops combine lectures and exercises to develop a working knowledge of system capabilities. Laboratory sessions complement and reinforce lecture information, through developing practical examples. When you complete a session, you'll have the skills necessary to apply the system capabilities to your own tasks.

Locations and Schedules

The workshops are scheduled through four basic locations, as shown on the schedule. Currently, the following places and times are scheduled:

Santa Clara, CA	Sept. 14-18, 1981
Rockville, MD	Sept. 21-25, 1981
Rockville, MD	Oct. 5-19, 1981
Santa Clara, CA	Oct. 12-16, 1981
Dallas, TX	Oct. 26-30, 1981
Santa Clara, CA	Oct. 26-30, 1981
Rockville, MD	Nov. 2-6, 1981
Rockville, MD	Nov. 30-Dec. 4, 1981
Santa Clara, CA	Nov. 30-Dec. 4, 1981
Dallas, TX	Dec. 7-11, 1981
Santa Clara, CA	Jan 4-8, 1982
Vancouver, BC	Jan 18-22, 1982

For Information

If you're interested in attending one of these sessions, or think you might be, you can get additional inforamtion by contacting your local Tektronix Sales Engineer. Or you can call Sam Dunn at (503) 644-0161, ext. 8660. She can provide information on other workshops as well, such as the recently-established 4020 Series Terminal workshop.

450K Cartridge Tape

Have you ever wished you could:

```
FIND Ø

MARK 1,450000
```

without getting an error message indicating the end of tape has been found?

Now you can. A tape is now available that can record up to 450,000 bytes, 50% more than the standard tape. The new tape can be ordered using part number 119-1439-01 for a box of five. Contact your local Tektronix representative. U.S. price for the box of five is $200. The standard tape is still ordered by part number 119-0680-01 for a box of five.

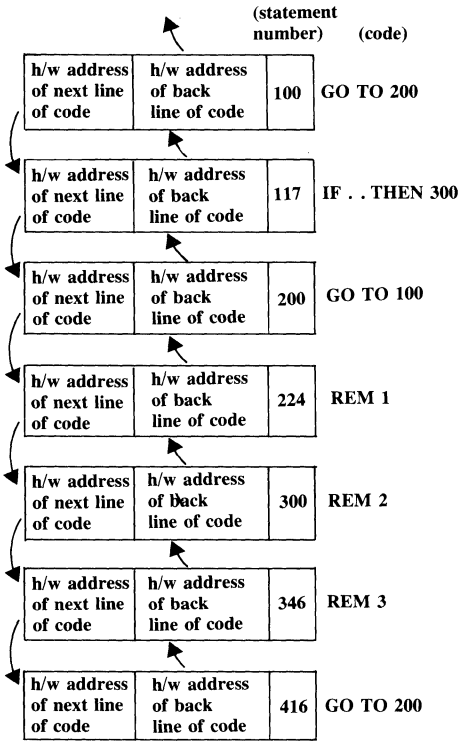
Input/ Output

John Bartholow, a Resource Analyst with the Fish and Wildlife Service in Fort Collins, CO, requests:

> **Would you please explain how the BASIC command RENUMBER works. That is, what is the algorithm used to perform the renumbering?**

Dan Taylor, Design Engineer at Tektronix, explains.

Each line of code in memory includes (among other things) the internal memory address of the preceding line of code, the hardware address of the next line of code, the line number of the present line of code, and the code.



When you key in the command:

RENumber 1000, 10, 200

this is what happens when it's executed.

Step 1. The system checks the first line affected (i.e., 200).

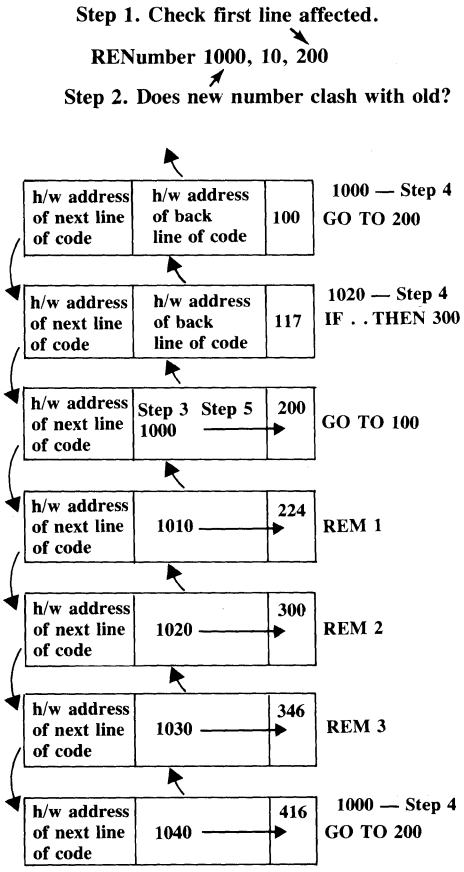
Step 2. It verifies that the new first number won't clash with non-renumbered code. If it does, it prints an error message and aborts execution. In other words, you can't change the sequence of your code through the RENumber command.

Step 3. It uses the space allocated for the back pointers as scratch space to store the new line numbers. At this point each statement includes (among other things), the hard address of the next line of code, the new line number, and the old line number.

Step 4. It searches through all internal code for statements containing a line number, i.e., GOTO, GOSUB, GOSUB OF, IF . . . THEN, etc. When found it changes the line numbers therein to the new line numbers.

Step 5. Once this is done, it moves the new line numbers from the scratch area (back pointer area) to the line number area and recreates the back pointers.

Because it uses the back pointer storage space for the new line numbers, the RENumber command execution doesn't require additional 4050 memory.



Operator Digitize on the 4663

Hugh Wilson and Grady Allman of American ENKA Company, Enka, NC sent in their observations on the 4663 Operator Digitize Mode on the 4663 Plotter.

Plotter.

We could not get the documented "Operator Digitize" to work correctly on the 4663 Plotter. Subsequent firmware revision (to level 4.0) and programming tips (TEKniques Vol. 4 No. 6, page 14) failed to resolve all the problems.

As a result of many attempts, we happened upon an unexpected and unorthodox solution. Instead of using the GPIB format described in the manual for the Plotter command, the RS-232 version of the command written in a GPIB BASIC program works better! Also, we found that INIT did not clear the memory of the Plotter as expected but the additional WBYTE @B,4: (where B is the primary listen address of the Plotter) was required to clear Plotter memory. Using both of these together appears to work correctly, (i.e., as reported in your documentation).

Mike DeAndrea and Javad Farjami, Tektronix Technical Support specialists at Wilsonville, OR, explain the documentation error and how INIT and WBYTE work.

The wrong command was inadvertently inserted in the "Commands" section of the Operator's manual. The DAB command for "Operator Digitize" is PRI@D,32:"AG",(# of points) and is correctly listed in Appendix I — Command Summary.

Init sends Interface Clear which causes the GPIB interface to clear addresses to itself and to clear the Service Request status. However, as defined in the IEEE-488 standard, it has no other effect on the operation of the plotter. The Device Clear message according to GPIB convention does clear both input and output buffers of the plotter and is sent by WBYTE @20: for all devices on the GPIB or WYBYTE @32+(device addres),4: for a specific device.

```
100 PRI "THIS PROGRAM UTILIZES OPERATOR DIGITIZE WITH THE 4663 PLOTTER."
110 PRINT "_ USE MOVE AND DRAW BUTTONS ON PLOTTER TO ENTER ALL POINTS."
120 PRINT "YOU MAY USE 'LAST POINT' BUTTON TO STOP DIGITIZING."
130 PRINT "_ NOTE: AFTER ENTERING MAXIMUM NUMBER OF POINTS, THE NEXT "
140 PRI "POINT ENTERED WILL BE TRANSMITTED AS THE 'LAST POINT' WHETHER"
150 PRINT " THE DRAW, MOVE, OR LAST POINT BUTTON IS PRESSED!"
160 PRINT "_ENTER GPIB ADDRESS OF PLOTTER :";
170 INPUT A
180 REMARK The VARIABLE B is the primary listen address for the plotter.
190 B=A+32
200 ON SRQ THEN 320
210 REMARK This WBYTE command is necessary to clear the plotter memory.
220 WBYTE @B,4;
230 PRINT "_GGG TYPE IN THE MAXIMUM NUMBER OF POINTS TO DIGITIZE : ";
240 INPUT P
250 REMARK P IS INCREMENTED TO ALLOW FOR 'LAST POINT' TRANSMISSION
260 P=P+1
270 PAGE
280 PRINT @A,32:"AG",P
290 REMARK z=0 means MOVE
300 REMARK z=1 means DRAW
310 REMARK z=2 means LAST POINT
320 FOR I=1 TO P
330 INPUT @A,32:X,Y,Z
340 GO TO Z OF 370,390
350 MOVE X,Y
360 GO TO 380
370 DRAW X,Y
380 NEXT I
390 WBYTE @B,4;
400 END
```

* Editor's Note

Applications Library Programs Consolidated

Quicker turnaround time and dollar savings for users should be the result of the just-completed Applications Library revised order procedure. 250 programs have been consolidated into 27 tapes/disks. An accompanying article details the reorganization.

TEKniques Schedule

We started the year on schedule with TEKniques Vol. 5 No. 1. But due to the Applications Library changeover, TEKniques' schedule slipped and Vol. 5 No. 2 was distributed in May rather than March. Unfortunately, we slipped again on No. 3, which is being distributed November 11. We will be back on track; however, the time remaining allows us to do only one more issue. TEKniques Vol 5 No. 4 will be out December 15. We feel the streamlined order procedure for the Applications Library will be worth the inconvenience.

Contest Winners Announced in Next Issue

The Applications Library program consolidation required more time than anticipated. Consequently, judging of the Graphic Contest was delayed. The judging is now in progress and winners will be notified by September 1, and announced in the December issue of TEKniques. We apologize for the delay.

Back Issues and Reprints from TEKniques

TEKniques is in its fifth 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 ..	AX4450
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.

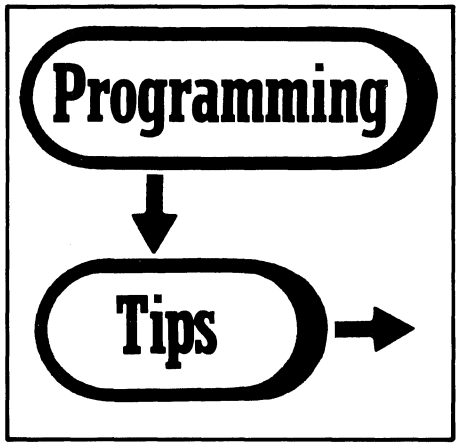
And, of course, back issues of TEKniques Vol. 4 (1980) and Vol. 5 (1981) will continue to be available from the 4050 Series Applications Library office that serves your area.

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).

Check Your 4907 File Manager

The first programming tip "Disk Change May Damage 4907 Files" contains important information. Please follow the instructions in that tip to check the operation of all units of your 4907 File Manager.



Disk Change May Damage 4907 Files

by Jack Gilmore
and Mike DeAndrea
Tektronix, Inc.
Wilsonville, OR

The disk change circuitry in the 4907 detects when a disk is removed. It prevents forwarding of information from one disk to the next when changing disks. The change is detected differently, depending on which drive was used last.

You can verify that the circuitry is working properly without disassembling the 4907 by performing the following procedure:

Step 1: Type "UNIT 0"

Step 2: Insert a formatted scratch diskette in drive 0 and type

CALL "MOUNT",0,A$

Step 3: Remove and reinsert the diskette in drive 0.

Step 4: Type "DIR". A disk not mounted error should appear if the circuitry is working.

Step 5: Remount the disk: CALL "MOUNT",0,A$.

Step 6: Type CALL "DSTAT",3,A$ (to select some other non-existent drive).

Step 7: Remove and reinsert the diskette in drive 0.

Step 8: Type "DIR". A disk not mounted error should again appear if this part of the disk change circuitry is working.

Repeat this procedure for the other drives in your system, replacing the 0 in Step 1, Step 2, Step 3, Step 5, and Step 7 with the appropriate drive number.

If problems are encountered — if the error messages don't appear — contact your local Tektronix Service Center.

Convert Decimal to Multiplier Notation

by Mike Wadzita
Tektronix, Inc.
Rockville, MD

The following subroutine may be useful to 4050/492P spectrum analyzer users or may apply to other applications that involve frequency information. The subroutine will convert a numeric value to its equivalent multiplier format for printing, i.e., Hertz, Kilohertz, Megahertz, or Gigahertz. A simple drive program is included.

```
100 INIT
110 REM --- SIMPLE TEST PROGRAM ---
120 FOR I=0 TO 11
130 F=1.234*10↑I
140 PRINT "I";F;"I----->I";
150 GOSUB 210
160 NEXT I
170 END
200 REM --- CONVERT DECIMAL # TO MULTIPLIER NOTATION ---
210 Q$="@KMG"
220 F=F*1.0E+10
230 A$=STR(F)
240 A=POS(A$,"E",1)
250 D$=SEG(A$,2,A-2)
260 M=VAL(D$)
270 D$=SEG(A$,A+2,LEN(A$))
280 E=VAL(D$)-10
290 M=M*10↑(E-3*INT(E/3+0.33))
300 C$=SEG(Q$,1+INT(E/3),1)
310 PRINT M;" ";C$;"Hz"
320 RETURN

1.234           ----->          1.234 Hz
12.34           ----->          12.34 Hz
123.4           ----->          123.4 Hz
1234            ----->          1.234 KHz
12340           ----->          12.34 KHz
123400          ----->          123.4 KHz
1234000         ----->          1.234 MHz
1.234E+7        ----->          12.34 MHz
1.234E+8        ----->          123.4 MHz
1.234E+9        ----->          1.234 GHz
1.234E+10       ----->          12.34 GHz
1.234E+11       ----->          123.4 GHz
```

Auto Log-On for Honeywell TSS

by Andy Galewsky
Big Thicket National Preserve
Beaumont, TX

This auto log-on routine is set up for Honeywell TSS. The program assumes you know how to turn on the modem and connect the phone to it.

The loop in statement 200 allows the connection to settle down before sending the carriage return to begin the log-on procedure.

On the Honeywell, the clear-to-send character is a Control-Q rather than a carriage return. Therefore, statement 260 sets the alternate EOL (end of line) character to Control-Q which enables the 4050 to recognize the request for the password. If this were not done, the program would loop endlessly when the mainframe sent the password request. The alternate EOL character is reset before the program is exited.

The proper account-number password sequence must be inserted into statement 320.

This short program offers some interesting possibilities. Instead of a call "TERMIN" at statement 340, you could have a BASIC program running in the 4050 issue commands to the mainframe that would be transparent to the user.

```
1 REM Auto Log-on Routine
2 INIT
3 M$=CHR$(13)
100 PAGE
110 REM Initialize the comm interface or set parameters
120 CALL "CMINIT"
130 REM CALL . . .
140 REM CALL . . .
150 REM CALL . . .
160 DIM A$(400)
170 PRINT "Please turn on modem. Dial 838-3761. Log-on will begin."
180 PRINT "<<<attempting to connect to remote >>>"
190 CALL "CMSET"
200 FOR I=1 TO 500+2250*(RND(0)>0.2)
210 REM DO NOTHING
220 NEXT I
230 REM SEND CR
240 PRINT @40:M$
250 REM Set alternate EOL to CTL-Q
260 PRINT @37,0:17,255,255
270 INPUT %40:A$
290 REM Reset EOL to CR
300 PRINT @37,0:13,255,255
320 PRINT @40:"Account#$Password"
330 PRINT "<<< Logged on. Begin ... >>>"
340 CALL "TERMIN"
```

Switchback Plotting

by Stan Jensen
State of California
Air Resources Board
Sacramento, CA

Many programs use data arrays and commonly output the contents from top to bottom, in the left-to-right order that printers use. When the number of columns and rows is small or the output is to the 4050 display, the technique is easy to program. However, when using the plotter, extra time is taken for the pen to move from the right column back to the left column for starting the next row.

When plotting large arrays and using the low speed setting of the 4662 Plotter, the following technique can save time.

```
LIS
100 REM "Switchback" Output
110 INIT
120 N1=13
130 N2=10
140 B=1
150 J1=N2
160 J2=1
170 J3=-1
175 D=1
180 FOR R=J1 TO J2 STEP J3
190 I1=1%B+N1%(NOT(B))
200 I2=N1%B+1%(NOT(B))
210 I3=1%B-1%(NOT(B))
220 FOR C=I1 TO I2 STEP I3
230 REM Plot or print data at column C, row R
240 REM For example:
250 MOVE @D:C%10-5,R%10-5
260 PRINT @D:"*";
270 NEXT C
280 B=NOT(B)
290 NEXT R
300 END
```

The variable B controls the horizontal direction of each row:

B=1 for left-to-right
B=0 for right-to-left

Line 280 reverses the direction used on the previous row. In addition to saving time when plotting, this feature will often cause observers to ask "How did they do that?"

Easy Curves Revisited Again

by Richard Stage
Stage Engineering, Inc.
San Diego, CA

I finally got around to trying the Easy Curves routine featured in TEKniques Vol. 3 No. 4 and again in Vol. 3 No. 8. However, I found it still divided by zero. Line 460 appears to be in error. It should read the same as line 380:

```
460 M2=(X3-X2)/(Y2-Y3)
```

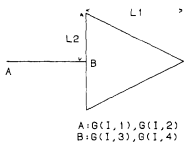
I made this change and it seems to work.

Omn-Direction Arrows Update

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

The calculation of the coordinates for drawing an arrow as described in TEKniques Vol. 4 No. 6 can be done much easier than is listed in the program. The following program positions and orients the arrows in the same way, i.e., move the plotter joystick until the arrow tail is positioned, then press User-Definable Key (UDK) #1. Next, position the arrow head and press UDK #1. The arrow will be drawn. Continue until all arrows requested have been positioned and drawn.

```
! INIT
2 SET KEY
3 GO TO 102
4 GO TO 302
5 REM ** KEY NO. 1 IS TO ENTER DATA **
90 REM ** THIS IS A SUBROUTINE TO DRAW ARROWS ON SCREEN **
92 REM ** IN ANY DIRECTION **
102 SET DEGREES
104 L1=1
105 L2=0.5
170 PRINT "LNO. OF ARROWS: ";
180 INPUT N1
185 PAGE
230 DIM G(N1,4)
240 FOR I=1 TO N1
250 T=0
260 GIN @1:V,W
270 MOVE V,W
280 PRINT @32,18:5
290 PRINT @32,24:"|"
300 GO TO 260
302 GIN V,W
304 RDRAW 0,0
306 PRINT "GGG";
310 PRINT @32,18:0
315 REM ** CHECK FLAG TO SEE IF INPUT IS THE TAIL OF
    ARROW (1ST ENTRY)
320 IF T=1 THEN 370
330 G(I,1)=V
340 G(I,2)=W
350 T=1
360 GO TO 260
365 REM ** FOUND DELTA X AND DELTA Y
370 M1=G(I,1)-V
380 M2=G(I,2)-W
385 REM ** CHECK TO SEE IF IT IS A SAME POINT
390 IF M1=0 AND M2=0 THEN 690
500 B3=1/SQR(M1^2+M2^2)
510 B1=M1*B3
520 B2=M2*B3
530 MOVE G(I,1),G(I,2)
540 G(I,3)=V+L1*B1
550 G(I,4)=W+L1*B2
560 DRAW G(I,3),G(I,4)
570 RDRAW L2*B2,-L2*B1
580 DRAW V,W
590 DRAW V+L1*B1-L2*B2,W+L1*B2+L2*B1
600 DRAW G(I,3),G(I,4)
610 GO TO 710
690 PRINT "NO ARROW"
700 I=I-1
710 NEXT I
720 END
```



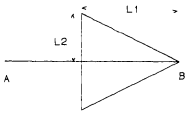
The differences between the old version and the new version are:

```
Differences between the old version (TEKniques 4.6 page 21) and
the new version (developed by Ir.A.C.Visser) of drawing omni-
directional arrows. There are in the new version no particular
situations!

DELETED:
100,100
110,160
190,220
395,490
540,680
720,990

CHANGED:
3 GO TO 102
4 GO TO 302
170 PRINT "LNO. OF ARROWS: ";
230 DIM G(N1,4)
370 M1=G(I,1)-V
380 M2=G(I,2)-W

NEW:
104 L1=1
105 L2=0.5
302 GIN V,W
304 RDRAW 0,0
306 PRINT "GGG";
540 G(I,3)=V+L1*B1
550 G(I,4)=W+L1*B2
560 DRAW G(I,3),G(I,4)
570 RDRAW L2*B2,-L2*B1
580 DRAW V,W
590 DRAW V+L1*B1-L2*B2,W+L1*B2+L2*B1
600 DRAW G(I,3),G(I,4)
610 GO TO 710
720 END
```

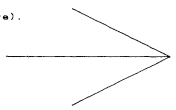
The new version does not force the near horizontal or near vertical lines to horizontal or vertical as in the old version.

By changing the values L1 and/or L2, it is possible to change the dimension of the top of the arrow. (L1 and L2 have the same values as the old version.) To draw other tops of the arrows you can use program alterations:

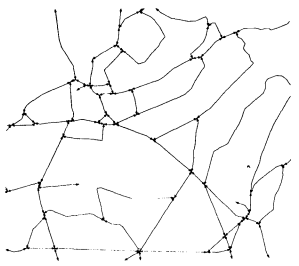
```
An other possibility (see figure).

CHANGED:
540 G(I,3)=V
550 G(I,4)=W
560 DRAW V,W
570 RDRAW L1*B1+L2*B2,L1*B2-L2*B1
580 RDRAW -2*L2*B2,2*L2*B1
590 REM
```




```
Yet an other possibility (see figure).

CHANGED ONCE MORE:
580 RMOVE -2*L2*B2,2*L2*B1
```



I use arrows to check digitized data. When digitizing lines, for example roads, the instructions are to stop on each knot of the lines. To check digitizing, the arrows are drawn on the beginning and ending of each line segment.



Checking digitizing by drawing arrows near knots of the lines. (System used by ICW, Wageningen, The Netherlands)

Determining if a String is Defined

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

Some time ago when writing a program that consisted of several overlays, I had to find a way to let the program determine whether or not a certain string variable had been defined (in a previous overlay).

Several methods came to mind immediately, including:

```
100 IF LEN(A$)>0 THEN 1000

100 A$=A$&"1234"
110 IF VAL(A$)=1234 THEN 1000
```

Unfortunately, these will only work if A$ has been defined, or at least set to a null-string. If A$ is undefined, those and similar checks will generate a fatal error, and the program will abort.

After some thought, I came up with the following little routine:

```
100 A$=REP("",1,0)
110 IF LEN(A$)>0 THEN 1000
120 INPUT A$
```

The REP function in line 100 does two things:

(a) If A$ has been defined previously, it is left unchanged by inserting a null-string at character position 1, and deleting 0 characters.

(b) If A$ is undefined, it is defined as a null-string.

Either way, line 110 checks to see if A$ actually contains data. If not, line 120 asks the user for input.

Editor's Note: This routine was also contained in "Program Initialization," Vol 4 No. 6.

Area Calculation — Update No. 1

by D. F. Davey
Department of Physiology
McGill University
Montreal, PQ,
Canada

The area calculation routine provided by Bernie Gunn (TEKniques Vol. 4 No. 8) can be simplified:

REMOVE ABS from line 3100
DELETE lines 3110, 3130, 3140

Furthermore, the restriction that "the area must be digitized in a clockwise direction" only applies if one insists that the printed area be positive. The restriction can be dropped and positive results ensured by changing A3 in line 3160 to ABS(A3).

Area Calculation — Update No. 2

by Stan Jensen
State of California
Air Resources Board
Sacramento, CA

While testing "Area Calculation" (TEKniques Vol. 4 No. 8) on a circle with a calculated area of 7854 units, we found that the program seems to have a logic error in statement 3100 and will occasionally generate a syntax error if statement 3120 is branched to for the final point. Below is the correct version.

We also eliminated some unnecessary statements and moved the ABS function to line 3150. If the area (A3) in line 3140 is positive, the line segment is clockwise about the area; a negative area indicates a counterclockwise direction. The new version operates significantly faster.

Editors Note: Statement 3130 in the *original* version should have read:

GOTO 3150

```
3000 REM: Begin area calculation
3010 A1=I
3020 RETURN
3030 REM: End area calculation
3040 A2=I-1
3050 REM: A3=Total area
3060 A3=0
3070 X(A2+1)=X(A1)
3080 Y(A2+1)=Y(A1)
3090 FOR J=A1 TO A2
3100 A3=A3+(X(J+1)-X(J))*(Y(J+1)+Y(J))
3130 NEXT J
3140 A3=0.5*A3
3150 A3=ABS(A3)
3160 PRINT "↑AREA =";A3
3170 RETURN
```

CALL "MOUNT" and 4907 Timing

by W. J. Orvis
Lawrence Livermore National Laboratory
Livermore, CA

When using a 4907 disk drive with a 4054 computer, a slight timing error can sometimes cause the 4054 to lock up busy. As part of our programming, we must occasionaly change a disk in the disk drive. This is usually accomplished by a sequence of statements like the following:

```
500 PRINT "INSERT THE PROGRAM DISK AND PRESS RETURN:";
510 INPUT Z$
520 CALL "MOUNT",0,S$
```

This will usually work just fine, except in the case where the disk has previously been inserted and mounted. Normally, the 4907 will mount the disk and return a status message to the 4054 which is stored in S$. If the disk was already in place and mounted, then the 4907 only sends back the status message. Occasionally, in a large program, the status message appears to be returned too soon, and conflicts with the memory operations taking place in the previous state. This results in the 4054 locking up busy waiting for the status message that it missed.

Correcting this problem is quite simple, as all we must do is to waste a few machine cycles to allow the memory operation in line 510 to complete before executing line 520. Do this by inserting a blank remark statement at line 515.

```
515 REM
RUN1000


500 PRINT "INSERT THE PROGRAM DISK AND PRESS RETURN:";
510 INPUT Z$
515 REM
520 CALL "MOUNT",0,S$
```

PRINT USING to Disk

Once a carriage return has been issued by the File Manager during a PRINT operation when using a format string, it must be redirected to the format string or it reverts to the default format.* Two readers have suggested methods for printing formatted arrays to the disk.

by J. W. Ball
U.S. Dept. of the Interior
Geological Survey
Menlo Park, CA

The 4907 checks to see if 72+1 characters are printed; if so, it issues a carriage return and reverts to the default format.

```
100 INIT
110 DIM A(8),B(6),A$(200)
120 FOR I=1 TO 8
130 A(I)=I
140 NEXT I
150 FOR I=9 TO 14
160 B(I-8)=I
170 NEXT I
180 KILL "TEST2"
190 CREATE "TEST2","A",1,0
200 OPEN "TEST2",1,"F",A$
210 PRINT #1: USING "14(6D.2D)":A,B
220 CALL "REWIND",1
230 INPUT #1:A$,B$,C$
240 PRINT @51:A$
250 PRINT @51:B$
260 PRINT @51:C$
270 A$=A$&B$
280 A$=A$&C$
290 PRINT @51:"LENGTH (SHOULD BE 126) = ";LEN(A$)
300 END
```

By dividing the data items between two PRINT statements, the format is retained.

*See TEKniques Vol. 4 No. 3, "Printing Arrays to Tape or Disk."

```
100 INIT
110 DIM A(8),B(6),A$(200)
120 FOR I=1 TO 8
130 A(I)=I
140 NEXT I
150 FOR I=9 TO 14
160 B(I-8)=I
170 NEXT I
180 KILL "TEST2"
190 CREATE "TEST2","A",1,0
200 OPEN "TEST2",1,"F",A$
210 PRINT #1: USING "8(6D.2D)":A
220 PRINT #1: USING "6(6D.2D)":B
230 CALL "REWIND",1
240 INPUT #1:A$,B$
250 PRINT @51:A$
260 PRINT @51:B$
270 A$=A$&B$
280 PRINT @51:"LENGTH (SHOULD BE 126) = ";LEN(A$)
290 END
```

In the following example, the same operation is performed using a random-access file:

```
100 INIT
110 DIM A(8),B(6),A$(200)
120 FOR I=1 TO 8
130 A(I)=I
140 NEXT I
150 FOR I=9 TO 14
160 B(I-8)=I
170 NEXT I
180 KILL "TEST2"
190 CREATE "TEST2","A",2,73
200 OPEN "TEST2",1,"F",A$
210 PRINT #1,1: USING "8(6D.2D)":A
215 PRINT #1,2: USING "6(6D.2D)":B
220 CALL "REWIND",1
230 INPUT #1,1:A$
235 INPUT #1,2:B$
236 PRINT @51:A$
237 PRINT @51:B$
240 A$=A$&B$
270 PRINT @51:"LENGTH (SHOULD BE 126) = ";LEN(A$)
280 END
```

by Tom Price
Lorillard
Greensboro, NC

The 4907 supports PRINT USING under most circumstances, but does not allow multiple lines to be correctly printed from a single IMAGE statement. That means the slash specification in an IMAGE will always cause problems. It is particularly obvious with arrays. The following examples illustrate the problem and how to overcome it.

```
100 DIM A(5,2)
110 A=0
120 OPEN "TEST",1,"F",A$
130 PRINT #1: USING 140:A
140 IMAGE5(3D3D/)
150 CLOSE
160 OPEN "TEST",1,"R",A$
170 INPUT #1:A$
180 PRINT A$
190 IF TYP(1)=2 THEN 170
200 END

RUN
  0  0
  0                0

  0                0

  0                0

  0                0
```

```
100 DIM A(5,2)
110 A=0
120 OPEN "TEST",1,"F",A$
121 FOR I=1 TO 5
130 PRINT #1: USING 140:A(I,1),A(I,2)
131 NEXT I
140 IMAGE 3D3D
150 CLOSE
160 OPEN "TEST",1,"R",A$
170 INPUT #1:A$
180 PRINT A$
190 IF TYP(1)=2 THEN 170
200 END

RUN
  0  0
  0  0
  0  0
  0  0
  0  0
```

Renaming Tape File Headers — Update

by Anne Thanhouser
Tektronix, Inc.
Wilsonville, OR.

The programming tip carried in TEKniques Vol. 5 No. 1 for renaming tape headers had one omission in the restrictions. Restricton #3 should read:

> Character 17 is normally P (program), D (data), T (text) or L (last). However, the two requirements are that the character has to be P if the file is to be "autoloaded" or if it's a binary program. In all other cases, it can be any printing character.

The revised program is included below. Changes were made in statements 170, 425, and 540 to 590.

```
100 INIT
110 PRINT "LITAPE HEADER LABELING PROGRAM"
120 PRINT "*************************"
130 PRINT "This program inserts labels in the headers of tape files."
140 PRINT " You can insert up to 22 characters in each header. These "
150 PRI "can be any printing character, subject to the following rules:"
160 PRINT "J    1. Positions 15 to 32 cannot have numbers."
170 PRI "    2. AUTOLOAD & BINARY progs must have a 'P' in position 17."
180 PRINT "    3. An 'S' in position 27 assumes the program is Secret."
190 PRINT " The 4851          will not be able to load it unless it is."
200 PRINT "If you violate any of these rules, the program will take"
210 PRINT "appropriate action."
220 PRINT "To end, enter '0' for the file number."
230 DIM Z$(1),N$(23),P$(1),Q$(1),R$(1),C$(2),S$(2)
240 C$=CHR(13)
245 S$=CHR(19)
250 C$=C$&S$

260 REM *** REQUEST FILE # & NAME ***
270 PRINT "ENTER FILE # ";
280 INPUT N
290 IF N=0 THEN 700
295 PRINT "       B"
300 PRINT "       A       P       S"
310 PRINT "XXXXXXXXX01234567890123456789012XXXXXXXXXXXX"
320 PRINT "JJENTER FILE NAME &&&"
325 PRINT "                    /"
330 PRINT USING "24(""H"")S";
340 N$=""
350 INPUT N$
360 N$=REPC(N$,1,LEN(N$))
```

If you already changed some binary program headers and omitted the "P" in byte 17, simply use this program again to re-insert it.

```
370 REM *** READ TAPE HEADER ***
380 PRINT @33,0:0,0,1
390 FIND N
400 INPUT @33:A$
410 Q$=SEG(A$,17,1)
420 R$=SEG(A$,27,1)
425 B$=SEG(A$,5,1)
430 A$=REP(N$,10,LEN(N$))

440 REM *** CHECK FOR SECRET PROGRAMS ***
450 M$=SEG(A$,27,1)
460 IF M$="S" AND R$<>"S" THEN 680
470 IF R$="S" AND M$<>"S" THEN 720

480 REM *** CHECK FOR NUMERICS ***
490 M$=SEG(A$,15,19)
500 M$=MID("1.234E-234"
510 IF VAL(M$)=1.234E-234 THEN 550
520 PRINT "JJJJCAN'T HAVE NUMBERS IN POSITIONS 15 TO 33. TRY AGAIN."
530 GO TO 300

540 REM *** CHECK FOR AUTOLOAD AND BINARY PROGRAMS ***
550 P$=SEG(A$,17,1)
555 IF Q$<>"P" OR P$="P" THEN 610
560 IF B$<>"B" THEN 575
565 PRINT "JJJJBINARY PROG NEEDS 'P' IN POS 17"
570 GO TO 295
575 IF N<>1 THEN 610
580 PRINT "JJJJAUTOLOAD PROG NEEDS 'P' IN POS 17, WANT TO CHANGE IT? ";
585 INPUT Z$
590 IF Z$="Y" THEN 300

600 REM *** REPLACE HEADER ***
610 FIND N
620 A$=N$&C$
630 PRINT @33:A$
640 CLOSE
650 PRINT @33,0:0,0,0
655 PRINT "L"
660 GO TO 220

670 REM *** ERROR MESSAGES AND END ***
680 PRINT "JJJJYOU'RE ABOUT TO MAKE THAT FILE SECRET. PLEASE TRY AGAIN"
690 GO TO 300
700 FIND 0
710 END
720 PRINT "FILE IS SECRET, CHAR #27 SHOULD BE 'S'. PLEASE TRY AGAIN."
730 GO TO 300
```

Clock on 4050 Screen

by J. Bruursema
Dutch Land Registry Office
Apeldoorn
and Ir. A. C. Visser
Institute for Land and Water Management Research
The Netherlands

Between demonstrations, performances, etc., a clock on the 4050 screen will inform passersby of the time. Change statement 620 for accurate timing.

```
4 GO TO 100
8 PAGE
50 REM
60 REM
70 REM
100 INIT
110 SET KEY
120 SET DEGREES
130 PRINT "L      WHEN WILL THE NEXT DEMONSTRATION BE GIVEN?"
140 PRINT "       (key in as time for example 14.30 ) ";
150 INPUT D9
160 IF D9<0 OR D9>24 THEN 740
170 H=INT(D9+0.01)
180 M=INT((100*(D9-H)+0.01)
190 IF M>59 THEN 740
200 PRINT "J          WHAT IS THE RIGHT TIME?"
210 PRINT "! Push RETURN-button 1 sec before the mentioned time !"
220 PRINT "                (for example 14.15 ) ";
230 INPUT T9
240 IF T9<0 OR T9>24 THEN 740
250 M=INT(T9+0.01)
260 M=INT((100*(T9-H)+0.01)
270 IF M>59 THEN 740
280 T9=T9-0.01
290 S=0
300 T9=T9+0.01
310 IF D9-T9<0.005 THEN 720
320 M=INT(T9+0.01)
330 IF M<24 THEN 360
340 T9=T9-24
350 GO TO 310
360 M=INT((100*(T9-H)+0.01)
370 IF M<60 THEN 400
380 T9=T9+0.39
```

```
390 GO TO 300
400 PRINT "LNEXT DEMONSTRATION WILL BE "
410 PRINT USING 420:"   GIVEN AT "jD9j" HOURS"
420 IMAGE 12A,2D.2D,6A
430 FOR I=1 TO 12
440 MOVE 65,50
450 ROTATE -30*I
460 RMOVE 41,0
470 RDRAW 9,0
480 NEXT I
490 MOVE 65,50
500 ROTATE 90-30*H-M/2
510 RDRAW 30,0
520 MOVE 65,50
530 ROTATE 90-6*M
540 RDRAW 40,0
550 MOVE 65,50
560 ROTATE 90-6*S
570 RMOVE 41,0
580 RDRAW 9,0
590 PRINT "E";
600 REM Change the value in the statement below
610 REM when the clock goes too fast or too slow
620 FOR I=1 TO 215+1725*(RND(0)>0.2)
630 NEXT I
640 S=S+1
650 IF S*INT(S/5)=S THEN 550
660 MOVE 65,50
670 ROTATE 90-6*S
680 RMOVE 45,0
690 RDRAW 5,0
700 IF S<59 THEN 620
710 GO TO 290
720 PRINT "LGGGGNOW THE DEMONSTRATION IS BEGINNING!!"
730 GO TO 750
740 PRINT "JGGTHE TIME, YOU ENTERED, CANNOT BE CORRECT. START AGAIN."
750 END
```

READing 4907 Data Files and Program DATA Statements

by Tom Price
Lorillard
Greensboro, NC

Code which reads DATA statements should not immediately follow code that OPENS a disk data file for read-only access. A READ command must be directed to the open file before information can be properly read from DATA statements, or a PRINT command must be executed.

```
100 INIT
110 OPEN "test1";1,"R",A$
120 FOR I=1 TO 3
130 READ C$
140 PRINT C$
150 NEXT I
160 DATA "first"
170 DATA "second"
180 DATA "third"
190 END
run

second
third
```

Notice the first data item is not read.

```
100 INIT
110 OPEN "test1";1,"R",A$
120 PRINT ""
130 FOR I=1 TO 3
140 READ C$
150 PRINT C$
160 NEXT I
170 DATA "first"
180 DATA "second"
190 DATA "third"
200 END
run

first
second
third
```

READ command to the open file sets internal pointers.

```
100 INIT
110 OPEN "test1";1,"R",A$
120 READ #1:H$
130 FOR I=1 TO 3
140 READ C$
150 PRINT C$
160 NEXT I
170 DATA "first"
180 DATA "second"
190 DATA "third"
200 END
run
first
second
third
```

A print operation will also set the pointers.

```
100 OPEN "TEST";1,"f",A$
110 FOR I=1 TO 3
120 READ A$
130 PRINT A$
140 NEXT I
150 DATA "First"
160 DATA "Second"
170 DATA "Third"

run
First
Second
Third
```

Opening the file for Full access will avoid incorrect data read.

Shellsort — Update

by Brian Diehm
Tektronix, Inc.
Wilsonville, OR

In TEKniques Vol. 4 No. 8, Dr. Bernard Gunn submits a shellsort routine that is faster than traditional bubble sorts. His algorithm is clever and does work as listed, but the initialization of the sort may be quickened and shortened by performing the following change:

DELETE lines 250 through 270
INSERT line 260:

```
260 D=2↑INT(LOG(N)/LOG(2)+1)
```

Particularly for sorts with large numbers of elements, this initialization will be faster than incrementally "trying" powers of two until finding the first one larger than N.

The complete listing of Dr. Gunn's algorithm, corrected as above, is listed below:

```
100 INIT
110 PRINT "LHOW MANY RANDOM NUMBERS DO YOU WANT TO SEE SORTED? ";
120 INPUT N
130 DIM A(N)
140 A=-1
150 A=RND(A)
160 A=10000*A
170 A=INT(A)
180 PRINT A
190 PRINT "** SHELLSORT STARTED **"
200 D=2↑INT(LOG(N)/LOG(2)+1)
210 D=INT((D-1)/2)
220 IF D=0 THEN 350
230 I1=INT(N-D)
240 FOR I=1 TO I1
250 J=I
260 L=J+D
270 IF A(L)=>A(J) THEN 330
280 T1=A(J)
290 A(J)=A(L)
300 A(L)=T1
310 J=J-D
320 IF J>0 THEN 260
330 NEXT I
340 GO TO 210
350 PRINT "** SHELLSORT FINISHED **"
360 PRINT A
370 END
```

Modified Auto-Paging — Update

by Anthony V. Hewitt
U.S. Naval Observatory
Flagstaff, AZ

The subroutine described in TEKniques Vol 4. No. 6 is useful but inflexible. Because not every output page causes a page-full, the operator cannot change the behavior of the program on a page-full condition by using such commands as: PRINT @32,26 : 3. The following routine has the function:

IF (the page is almost full) FILL IT
RETURN

which overcomes this limitation.

```
999 REM PAGING SUBROUTINE
1000 GIN X,Y
1010 IF Y>20 THEN 1040
1020 PRINT
1030 GO TO 1000
1040 RETURN
```

X,SUMX

by Pete Nelson
Tektronix, Inc.
Beaverton, OR

The following line of code was placed in a 4051 with X dimensioned as a matrix:

```
160 PRINT @Q: USING 170: X,SUMX
```

When LISTed or SAVEd it appears as:

```
160 PRINT @Q: USING 170:XSUM(X)
```

However, the line cannot be recalled and reentered as XSUM(X) without a Syntax Error occurring. The program with XSUM(X) can be stored on tape, but halts at line 160 when "OLD"ed into memory.

```
LIS
100 INIT
110 Q=32
120 DIM X(20)
130 FOR I=1 TO 20
140 X(I)=I^2
150 NEXT I
160 PRINT @Q: USING 170:XSUM(X)
170 IMAGE 21(5D)
180 END

RUN
   1    4    9   16   25   36   49 64   81  100  121 144  169  196
225  256  289  324  361  400 2870

FIND 4
SAVE
FIND4
OLD

SYNTAX ERROR      ^
160 PRINT @Q: USING 170:XSUM(X)
                        ↓
```

The work-around is to precompute SUMX prior to printing:

```
159 Y=SUMX
160 PRINT @Q: USING 170:X,Y
```

or, since PRINT USING is being done, use:

```
160 PRINT @Q: USING 170:X;SUM(X)
```

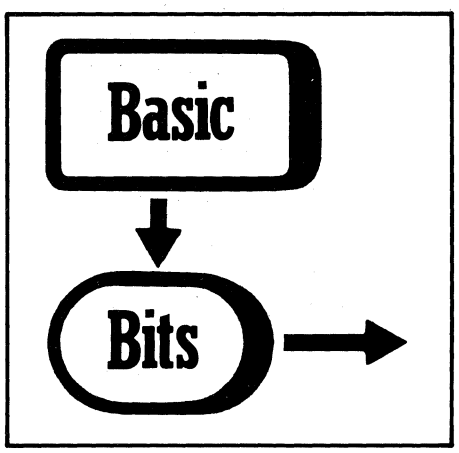
Salvaging Old Tapes

by David Yager
Physics Department
Sam Houston State University
Huntsville, TX

Some of our older tapes are unwound by our new 4052. The 4052 (and occasionally the 4051) refuse to find some files on these tapes. We suspect the faster winding speed is the reason. Rather than spend money on new tapes, we've found ways to rehabilitate old ones. Two remedies that work for us are:

1. Check the beginning and end of the tape to see if the optical load point sensor holes are clear. Sometimes the punch doesn't push the plug out of the hole. This can go unnoticed by the 4051, but will be picked up by the 4052.
2. Our tapes have the most used programs bunched at the beginning of the tape. Often the last half of the tape is unused. We've been able to rewind the tape so the trailing end becomes the leading end. This part of the tape is usually brand new. The travel limit sensor holes must be repunched in the old sequence on both ends of the tape so that the rewind, autoload and "MTPACK" commands do not unspool the tape. These holes can be cut with an X-Acto knife with care.

Editor's Note: Tapes do wear out and backups should be made. While procedure 2 may work, Tektronix does not condone it and does not guarantee reliability.



UDU Commands and GDU Commands

by Dan Taylor
Tektronix, Inc.
Wilsonville, OR

The GIN statement (and POINTER command) records the current position of the cursor on the graphic display in **user data units** determined by the current WINDOW statement. MOVE and DRAW statements also treat their arguments as **user data units**.

Whereas, the INPUT statement with secondary address of 24 treats the incoming data as **graphic display units** and the PRINT statement with secondary address of 20 or 21 treats its arguments as **graphic display units**.

Therefore, don't use GIN (or POINTER) with PRINT @D,20 or PRINT @D,21; or INPUT @D,24 with MOVE and DRAW (D = device #). The results may be different than expected, depending on your WINDOW.

For example, try the following routine to see the placings on the screen, depending on whether X and Y are treated as user data units or graphic display units.

```
100 INIT
110 WINDOW -10,10,-20,20
120 PRINT "L"
130 MOVE 5,15
140 GIN X,Y
150 MOVE X,Y
160 PRINT "UDU"
170 PRINT @32,21:X,Y
180 PRINT "GDU"
190 END
```

Efficient Subroutine Use

by Ron Robinder
Tektronix, Inc.
Beaverton, OR

The following guidelines will avoid inefficient subroutine use and minimize code in a program.

1. Avoid using a subroutine only once. Incorporating a subroutine directly at the point of call will eliminate two lines of code: the GOSUB statement and the RETURN statement.
2. Avoid one line subroutines unless the line is long and complicated. These can be repeated in place of the GOSUB statement, eliminating a total of two lines of code from the program for each one line subroutine.*
3. Two line subroutines should be called at least three times (unless they are long and complicated). Any other larger subroutine is acceptable for two or more calls.*

*One exception might be if you were going to be changing that one-or-two-line subroutine and wanted it in one location.

These guidelines insure a minimum of lines of code. Other guidelines may apply with regard to run time optimization, but I haven't been concerned with this aspect.

Packing Tapes

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

Regarding your recent article on maintenance, I would like to comment on the "Packing Tapes" section. Using the 4051 system, you need only FIND n (n LAST), MARK 1,400000 one time. Thereafter, TLIST will very nicely "Pack" the tape.

Savings Programs to Disk

by Dave Kirkpatrick
U.S. Army Corps of Engineers
St. Louis, MO

When editing disk files from the 4907 using the Editor ROM Pack with the 4050, be sure to transfer the edited file to a new and unique disk file. Should you encounter no errors, power failures, or other problems, you would then be free to kill the original disk file that you were editing and rename the newly transferred disk file with the old disk file name. Doing this rather than first killing the disk file and re-creating it to hold the edited data would avoid losing your only remaining copy in case of error.

Saving Programs to Tape

by Dr. Ir. F. J. Franck
Jan van Scorelpark, 12
The Netherlands

SAVing a memory-resident program after development or modification often represents a hazardous manipulation. Several mistakes can be made:

FINDing an unappropriate file number and SAVing, thereby overwriting another file;

SAVing the program in NOHEADER, NOCHECKSUM or 128 byte format;

SAVing the program with an extra carriage return (after a PRINT @37,26:1);

OLDing the program instead of SAVing it;

Power failure during the SAVE;*

Bad spot on tape develops.*

The following routine may help in remembering the right file number and preventing other mistakes. Dedicate one User-Definable Key to this purpose or start your SAVE routine at an easily recognizable line number, e.g., 10000 and save your program with RUN 10000.

```
24 GO TO 10000
30 REM
31 REM
32 REM
33 REM      ****************************************
34 REM      *                                      *
35 REM      *  Put your program taken from file N  *
36 REM      *  somewhere between 100 and 9999      *
37 REM      *  or behind 12000.                    *
38 REM      *                                      *
39 REM      *  Set N=the appropriate value in line *
40 REM      *    10020!!!.                         *
41 REM      *                                      *
42 REM      ****************************************
43 REM
10000 REM ================================SAVE
10010 REM *** 'N' IS THE FILE NUMBER ***
10020 N=20
10030 PAGE
10040 PRINT @37,26:0
10050 PRINT @33,0:0,0,0
10060 FIND N
10070 CALL "HEADER"
10080 PRINT "¿Save [NAME] on file "¡N¡", O.K.ÉY/N?ÉÉÉÉÉ"¡
10090 INPUT A$
10100 A$=SEG(A$,1,1)
10110 IF A$<>"Y" THEN 10100
10120 FIND N
10130 SAVE
10140 REM *** or CALL "BSAVE" ***
10150 CLOSE
10160 PRINT "DoneÉÉÉ"
10170 END
10180 PRINT "PROGRAM HAS NOT BEEN SAVED!"
10190 PRINT "       ===ÉÉÉÉÉÉÉÉ"
10200 END
10210 REM ================================END OF SAVE
```

*To overcome these two events, you must have a backup of the program either on the current tape, or a separate tape.

Editor's Note: If you are using the 4051, certain lines of code will have be replaced with the following to point out the file header information:

```
10050 PRINT @33,0:0,0,1
10060 FIND N
10061 INPUT @33:A$
10062 PRINT @33,0:0,0,0
10063 B$=SEG(A$,33,8)
10064 B=256*VAL(B$)
10065 B$=STR(B)
10066 A$=SEG(A$,1,32)
10067 A$=A$&B$
10070 PRINT A$
```

Library Consolidation

You've overwhelmed us! What began over five years ago as a rather informal method of sharing 4050-Series programs has outgrown its usefulness. When it all began the Libary had a few programs, a few members and we expected (and got) only a couple of orders per week.

Now, with over 200 programs available and Library membership growing rapidly, our mailbox is full of orders each morning. To continue to provide efficient service to you, we've streamlined program ordering.

We've combined like-subject programs together on tapes or disks, packaged the tape or disk together with the supporting documentation and given the package a Tek part-number. Now you can order programs just like ordering blank tapes or plotter pens, through your local Tektronix representative.

Here is a list of the packages now available. Packages include the media ("T" = tape, "D" = Disk) and documentation (without listings). Documentation is available separately.

To find out what is in each package you will need the latest issue of the Applications Library catalog. Most of you should have received one enclosed with this issue of TEKniques. (The catalog designed to fit in a three-ring binder (Tek P/N 016-0367-00) and we will add pages later to update it.) If you don't have a catalog, contact your local Tektronix representative.

In the past, we have exchanged three programs of your choice for each new program accepted by the Library. Now we will offer a full package for each one, a considerable increase in most cases.

New programs not included in this initial offering will be combined until we get a tape or disk full. New packages will be announced via TEKniques and supporting catalog pages will be distributed to all Library members.

Package Title	Package Part #
Business Aids T1	062-5987-01
Business Aids T2	062-5988-01
CAD T1	062-5976-01
CAD D1	062-5977-01
Character Generator T1	062-5951-01
Education/Research T1	062-5982-01
Education/Research T2	062-5983-01
Electrical Engineering T1	062-5978-01
Graphing T1	062-5964-01
Graphing T2	062-5965-01
Graphing T3	062-5966-01
Graphing D1	062-5967-01
Graphing D2	062-5968-01
Interfacing T1	062-5984-01
Mapping T1	062-5980-01
Mechanical Engineering	062-5979-01
Programming Aids T1	062-5971-01
Programming Aids T2	062-5972-01
Project Aids T1	062-5985-01
Project Aids D1	062-5986-01
Recreational Plots T1	062-5989-01
Slidemaker T1	062-5962-01
Slidemaker D1	062-5963-01
Text Processing T1	062-5969-01
Text Processing D1	062-5970-01
Utilities T1	062-5974-01
Utilities D1	062-5975-01

4050 Series Applications Library Programs

Ordering

The current system for ordering individual 4050 Series APPLICATIONS Library programs will remain in effect until December 31, 1981. However, effective immediately you may also order programs by the new package system. After December 31, 1981, only the packages may be ordered.

Individual Program Orders (until December 31, 1981)

Documentation and program listings of each program are available for a nominal charge. Programs will be put on tape or disk for a small recording fee per program plus the charge for the tape cartridge or flexible disk. One tape/disk will hold several programs. Programs will be recorded on like media only, i.e., programs on tape cannot be sent on disk and vice versa unless so noted in the abstract.

(The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc. assumes no responsibility and shall have no liability, consequential or otherwise, or any kind arising from the use of this program material or any part thereof.)

Domestic U.S. Prices:

Documentation and Listing only	*$25 per program*
Documentation, Listing and Recording Fee	*30 per program*
Tape Cartridge	*36 per tape*
Flexible Disk	*15.per disk*

Forms

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

Packaged Program Orders (effective immediately)

Call your local Tektronix Sales Office and give your representative the part number of the package(s)you wish. A price list is included with your 4050 Series Applications Library catalog.

Contributions

The current exchange system for program contributions will also remain in effect until December 31, 1981. However, you may take advantage of the package exchange instead. After December 31, 1981, only the package exchange will be available.

Three-for-One Exchange

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

Outside U.S.

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

Package Exchange

Contribute one program to the Library and recive one package in exchange.

Since the following program abstracts, along with their graphic output (where applicable), are contained in the new 4050 Series Applications Library Catalog, only their description is included in TEKniques. See the catalog for their graphic capabilities.

ABSTRACT #: 51/00-6124/0

Title: **4050/492P Utility**
Author: David B. Green
Tektronix, Inc.
Beaverton, OR
Memory Requirement: 32K
Peripherals: 492P Spectrum Analyzer
Optional-4662 Plotter
Statements: 1286
Files: 8 ASCII Program
9 ASCII Data
7 Binary Data
Requires dedicated tape

This is a demonstration software package, however, it contains many useful algorithms that can be adapted for personal application programs. The following are the sub-programs in this package.

Indexing/Menu

Contains the initial control structure for the entire package. Used to access desired sub-programs and also contains a complete list of 492P commands. This sub-program also allows direct control of the analyzer over the GPIB bus.

492P Waveform Transmission

Establishes a single keystroke approach which allows you to either transmit or receive waveform data. Instrument Front Panel Settings may also be transmitted or received along with each corresponding waveform. Provides for simple data storage to magnetic tape which will allow you to store measurement conditions and data. By the stroke of a few User-Definable Keys, the 492P can be made to reconfigure to previously made measurement parameters.

Spectrum Analysis

Provides rapid spectrum analysis from sweeps made on the 492P. Two types of analysis are provided:

1) This analysis lists and graphs the frequencies to either side of the signal at center screen. Sideband measurements are made easy with this portion of the program.

2) Direct frequency measurements are the function of this section. All signals appearing on screen are cataloged and listed on the controller.

492P Command Syntax Tutorial

This is a complete tutorial provided for the novice. The tutorial provides for the user comprehensive instruction in the use of the GPIB command syntax used in the 492P. As each item is presented, the user is allowed to "experiment" with each command with help as needed provided by the controller. Periodic review of this section keeps one's programming skills at a maximum.

Total Harmonic Distortion

This sub-program was provided especially for sales personnel. It operates using only the internal calibrator signal provided in the 492P. The program automatically searches for and measures all requested harmonics of a given frequency and computes the total Hermonic distortion. Measurement results are provided in a neat graphics format.

Utility Programs

Duplicate tapes. Name tape file headers to use for data storage.

ABSTRACT #: 51/07-8029/1

Title: CDOS 4907 Operating System
Authors: Lunne Cueto
Carl Dawson
Tektonix, Inc.
Rockville, MD
Orlando, FL
Revised by: Frits Handgraaf
Tektronix, Inc.
Amstelveen, The Netherlands
Memory Requirement: 16K
Peripherals: 4907 File Manager
Optional—4641 Printer
Files: 26 Program
28 Data
Statements: 1480

CDOS has been modified for one start-up program instead of the two contained in the original version, and some changes made.

CDOS has been modified to take advantage of the 4052/4054 and some minor streamlining has been incorporated:

One start-up program instead of two;

A new APPEND procedure in the main program to avoid memory full;

The system disk may be mounted in any drive;

DISCTOTAPE routine changed for optimum file marking;

New program lines in the FORMAT to prevent the system disk from being formatted;

A few other minimal charges for more efficient execution.

ABSTRACT # 51/00-8043/0

Title: **Micro-Grasp**
Author: Roger W. Bowen
U.S. Geological Survey
Reston, VA
Memory Requirement: 32K
Statements: 1135
Files: 3 ASCII Program
4 ASCII Date (examples)
Requires dedicated tape

Micro-Grasp is a scaled down version of the USGS Geologic Retrieval and Synopsis Program (GRASP). The GRASP system is extensively used on a variety of small to large scale hosts to access many diverse types of data. Micro-Grasp retains the major features of GRASP and should prove useful on private (or non-shared) databases of a small to medium size (less tha 300K bytes). Micro-Grasp is a command-oriented system which operates on fixed-field character and/or numeric data in a tabular form.

Each database consists of a data definition file and a master file. The data definition file describes the master file. The data definition files contains the number of fields in a master-file record, the number of characters in a master-file record and the number of the master file. This is followed by 'n' field-description records. Each field-description record has a field name, data type, beginning position, and optional ending position.

The master file contains the actual ASCII data. Data may be numeric or character and must be in a fixed field format.

Twelve commands operate on the data in the data-definition files and the master files:

Help: displays a list and shor description of each of the commands.

File: establishes or changes a data definition file.

Name: outputs the data definition file to screen or a selected tape file.

Inpu(t): creates a master file in prompted mode.

Cond(ition): enters or adds a set of conditions using valid field names and their relation (equal, less than, greater than, less than or equal to, greater than or equal to, not equal to, or contains string) to a value (blank, numeric or character string) on which to select data.

Logi(c): completes selection criteria by combining conditions (entered earlier) with the logical operators: and, or, not.

Defi(ne): defines new variables which remain in effect until a new master files is selected.

List: displays selected fields for each record which satisfies the selection criteria (conditions and logic).

Revi(ew): displays the number of the master fiel, the field names being listed, current conditions and logic expressions used in the listing.

Stat(istics): calculates statistics (minimum, maximum, sum, mean, variance, standard deviation) for each specified numeric field.

Util(ity): provides copy, tlist, mark, print and input.

Quit: exits Micro-Grasp.

ABSTRACT #: 51/07-8049/0

Title: Disk Directory
Author: Nick Ogbourne
Comalco Aluminum (Bell Bay) Ltd.
George Town, Tasmania
Memory Requirement: 8K
Peripherals: 4907 File Manager
Files: 1 Program
Statements: 193

Disk Directory maintains a directory of up to 50 disk programs and controls access to and execution of those programs.

Disk Directory creates and maintains an index file. This index file includes the file identifier, program # (sequenced in the order entered), and user-input information (up to 44 characters) about the file. Programs may be added or deleted through the User-Definable Keys.

Disk Directory reads the index file, prints a directory of the files (multi-page if necessary) and prompts for the program of your choice. It will warn you if a selected file is a binary data file. Any other type of file will attempt to load.

ABSTRACT #: 51/00-9548/0

Title: **Frames**
Author: Mike Tharpe
Honeywell, Inc.
Minneapolis, MN
Memory Requirement: 16K
Peripherals: Optional-4662 Plotter
Statements: 540
Files: 1 ASCII Program

For the final touch, frame your slides

"Frames" consists of 12 different designs which may be drawn around your slide on the screen or 4662 Plotter.

Spokes
Circles
Spokes and Circles
Straight-Edged
Straight-Edged with Perspective
Slants
Six Pointed Stars
Maze
Boxes within Boxes
French Provincial
Cross-Hatched
Sunburst
Corners Only

Instructions for changing the size of your border are included.

ABSTRACT #: 51/00-6126/0

Title: **Bibliography Management & Search**
Authors: K. Ho and G. Tzitzikalakis
Columbia University
Dept. of Orthopedic Surgery
New York, NY
Peripherals: Optional-4662 Plotter
Memory Requirement: 24K
Statements: 623
Files: 2 ASCII Programs
1 ASCII Text (example)
Requires dedicated tape

These programs store, review and retrieve bibliographical references on tape.

Management Program

The program numerically sequences the references in the order entered.
Four Data items comprise the reference:

author's name	(72 char)
title	(144 char)
source	(144 char)
content	(216 char)

Content contains the keywords which will be used in a search.

You may review or correct any of the references stored on tape by responding with the reference number when prompted.

The data files may reside on the same tape as the program files or a separate tape. However, the tape from the first data file to the last must be dedicated to this program since the Management Program automatically MARKs files as needed and maintains a numeric sequence throughout.

A mnemonic menu drives the routines:

ENT — search the last reference on tape, get and store new references one by one
REV — search the designated reference and print out
COR — calculate the file number of the designed reference, store the file's data in a string X$, correct the designated reference and store string X$ onto the same file
SEA — OLD the "SEARCH" program for retrieval of references
MEN — print list of Mnemonics
TER — end execution

Search Program

This program will search the references stored on a tape by the Management Program. The search may be based on keyword(s) in the content data item, or on the author(s) name. Once you choose the search mode, you will be prompted for the keyword(s). Only when the content keyword(s) or the author(s) — depending on the mode chosen — satisfy all keyword(s) input, will the reference be retrieved.

The retrieved references may be printed to the screen or plotter, stored on tape, or refined and stored on a different tape.

A mnemonic menu drives the routines:

KEY — search and retrieve references with keywords on content
AUT — search and retrieve references with keywords on author
REF — search and retrieve references stored on tape from previous searches, with a different set of keywords on content or author
TAP — store retrieved references on tape
SCR — print retrieved references on screen
PLO — print retrieved references on output device (plotter)
CON — continue the search that was stopped due to memory space
MEN — print list of Mnemonics
MAN — OLD "MANAGEMENT" program and execute it
TER — end execution of program

An example of data file is included with the two programs.

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.

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

Tektronix
COMMITTED TO EXCELLENCE

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