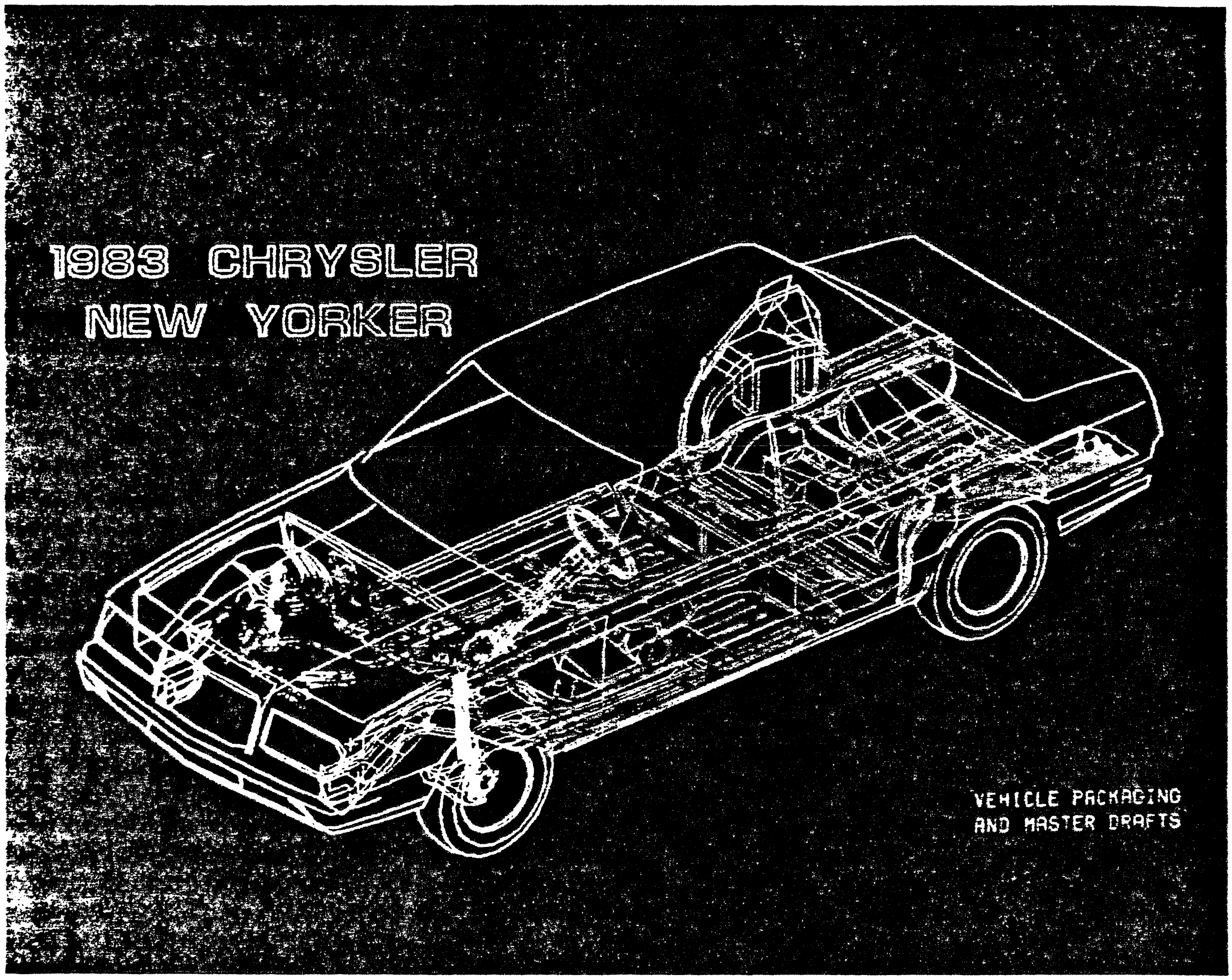
THE IDD
APPLICATIONS NEWSLETTER

VOL. 7 NO. 3
FALL 1983

Tekniques



Tektronix
COMMITTED TO EXCELLENCE

Teknіques

In This Issue

TEKniques, the IDD Applications Newsletter, is published in the Information Display Division of Tektronix, Inc., Mail Stop 63-575, P.O. Box 1000, Wilsonville, Oregon 97070. It is distributed to TEKTRONIX computer and terminal users.

Publishing Manager Jody Peake
Managing Editor Patricia Kelley
Technical Editor Dan Taylor
Graphic Design John Ellis
Typesetting Jean Bunker

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

On the cover: At Chrysler Corporation structural modeling in color on the Tektronix 4115B Color Graphics Terminal is performed through Local Programmability.

Ron Brackett (left) and Ron Erich, performing ultrasound materials analysis at the Port Hueneme (Calif.) instrumentation facility adjacent to NCEL's test dive tank. Erich is performing real time data analysis with the Tektronix 4052 desktop computing system, while Brackett studies ultrasound information with an ultrasonic flaw detector.

Underwater Inspection of Waterfront Structures Aided by 4052 Desktop Computing System

PORT HUENEME, Calif., – In support of its massive fleet of ships, aircraft and military vehicles, the U.S. Navy maintains an extensive Naval Shore Establishment, including a major network of waterfront facilities. More than two-thirds of these stationary facilities – piers, wharfs, bulkheads, and quaywalls – are over 30 years old, and most of these facilities have reached, or exceeded their original design life, resulting in escalating maintenance and repair costs.

Underwater inspection of waterfront facilities has until recently been limited to visual surface observations by Navy scuba divers using rudimentary tools. However, visual inspection cannot identify hidden flaws in steel structures caused by corrosion, in concrete pilings stemming from years of freeze/thaw conditions, or in wooden pilings due to marine borers or fungi.

Currently, the U.S. Naval Civil Engineering Laboratory (NCEL) is developing a computer-aided technique for the underwater inspection of steel, concrete and timber. The new technique, which uses high-frequency sound-waves (ultrasound) to determine metal thickness and the internal defects of concrete and timber, will enable the Navy to efficiently inspect existing facilities and recommend repair or renovation work before an aging facility suffers a potentially dangerous structural failure.

"The rapid aging of existing waterfront facilities, coupled with increasing maintenance and repair costs forced a move from our traditional visual inspection techniques," says Ron Brackett, managing engineer on the underwater inspection program at NCEL.

"While a trained diver can determine evidence of external deterioration in steel plate, concrete structure or timber piling, a diver cannot sense the hidden damage caused to steel by corrosion, internal damage to concrete or concrete reinforcement, or biological attack on wooden structure," related Carmela Keeney, project engineer for the concrete and timber inspection investigations.

"With our desktop computing system and ultrasonic generation and analysis equipment, we can perform extremely accurate, real-time analyses of the three major construction materials found in waterfront structures – steel, concrete and timber."

NCEL is the principal research, development and testing center for Navy shore facilities, fixed surface and subsurface ocean facilities, and for the Navy and Marine Corps construction Forces. Located in Port Hueneme, California, the lab is engaged in a wide variety of disciplines, including energy management, materials engineering, ocean facilities engineering and environmental protection. In addition to its underwater material analysis efforts, other research programs include facility anchoring, welding technology, solar and wind energy, seawater hydraulic systems, and design of advanced base and amphibious facilities.

Visual Inspection: Blind to Hidden Damage

Underwater visual inspection generally begins with a thorough cleaning of a structural element, removing marine growth which may ɔe hiding surface damage. Once clean, a diver is able to determine the most apparent damage. In the case of steel structures, excessive corrosion, severe mechanical damage (caused by a ship colliding with a structure) or less serious surface pitting from corrosion, can be recognized.

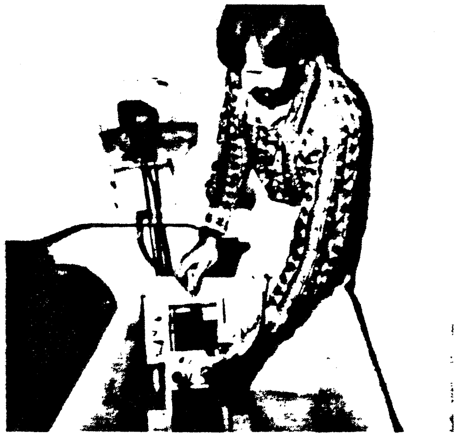
Severe damage to concrete facilities, such as mechanical damage from overloading, cracking from freeze/thaw conditions and severe corrosion of reinforcing steel is also apparent to the trained eye. With regards to wooden structures, a diver can recognize the external evidence of boring marine animals, and the splintering or overall failure of timbers due to mechanical overloading.

"Although a diver can recognize the most apparent damage to marine structures, we are most concerned with the corrosion, mechanical damage and biological attack which is hidden from sight," says Brackett. "Unexpected failure stemming from unseen damage is a potentially dangerous occurrence, and repairs are tremendously expensive."

Adapting Nondestructive Techniques for ˀJnderwater Use

'We had to develop analysis procedures which would not damage the integrity of the structure, because the facilities to be tested are working elements of the Naval Shore Establishment," says Brackett. "Numerous techniques have been developed for nondestructive materials analysis, but few have been adapted for underwater applications."

In recent years, NCEL engineers have examined various techniques for underwater materials inspection, including ultrasound, radiography, vibration analysis and acoustic emission. Their studies have indicated that conventional radiography is an unacceptable underwater technique due to the long exposure times required to record an image. Vibration analysis and acoustic emission are difficult to manage in an underwater environment because of the complex integration of the structural elements and the length of time required to acquire the information needed to detect a change in condition.



'on Erich is able to monitoт all test procedures ia a closed-circuit videotape and underwater .ommunications system. With the aid of the Tektronix computing equipment, test procedures can be analyzed, the results communicated to the divers and procedures may be refined, all during a single dive session.

Ultrasonic testing, however, requires a limited amount of underwater equipment, and a relatively short analysis time to produce accurate results. Essentially, ultrasonic testing uses the measurement of the transit time of high-frequency sound waves to detect both metal thickness and the internal defects present in concrete and timber.

Computer Controls Test System, Analyzes Data

At the heart of NCEL's underwater materials analysis system is a Tektronix 4052 desktop computing system. Configured with a variety of electronic instrumentation, the desktop computer provides test system control, data analysis and graphics capabilities to the test procedures.

"The Tektronix gear is powerful enough to provide real-time analysis and graphic presentation of our complicated test procedures, yet its ruggedness and compact size enables the machine to perform in both lab-based and field test sessions," says Brackett.

"As a diver scans the surface of the test structure with the ultrasonic equipment, the desktop computer analyzes the ultrasonic transit time information, and then displays the findings in graphical form. In addition to being easier to understand than a complex series of numerical values, the graphics capabilities afforded by the Tektronix gear enables us to immediately sense the validity of our procedures and the accuracy of our findings."

Information, whether it be alphanumeric or graphic, is displayed on the Tektronix 4052 via a high-resolution direct-view storage tube CRT screen. Addressing nearly 1500 points, the CRT display provides the accuracy necessary for complex materials analysis procedures. A 16-bit microprocessor provides both fast processing and high-volume data manipulation.

In addition to the 32 K bytes of internal memory, NCEL has configured its system with two Tektronix 4924 digital tape cartridge drives for auxiliary storage. Hard copies of the CRT display are made using the Tektronix 4631 hard copy unit.

Underwater Materials Analysis - Metal Structures

Metal structures, typically retaining walls and breakwaters in waterfront applications, posed a difficult problem for NCEL engineers in the development of underwater analysis procedures. The desired result in a metals analysis procedure is a thickness measurement across a representative cross-section of the structure - whether it be steel sheet pile or H-pile. However, as most metal structures form a barrier between a body of water and a land-based shore facility, only one side of the structure is accessible - that being the waterfront side.

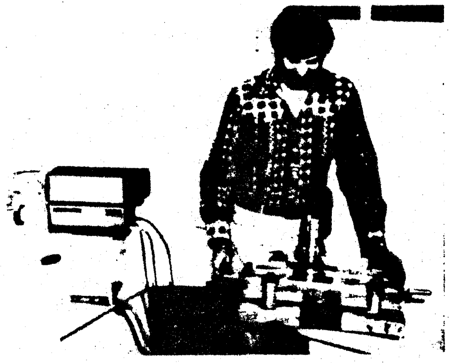
"Our divers do not have the luxury of simply fitting a caliper around a section of steel bulkhead to determine its thickness," says Brackett. "Consequently, our measurement technique requires accessibility to one side of the structure to achieve accurate readings. In addition to being used underwater, the procedure is also nondestructive - as drilling holes or cutting out metal sections could damage the integrity of the facility, possibly resulting in a major failure."

According to Brackett, NCEL has achieved excellent results in the underwater measurement of metal structures using an ultrasonic testing technique known as pulse/echo measurement. Using a single transducer to send and receive signals, this procedure measures the transit time of high-frequency soundwaves through the metal structure to determine its thickness.

The actual test procedures were developed and refined in a land-based test tank at the Port Hueneme laboratory. The test tank, in addition to being adjacent to a control center holding the desktop computer and other instrumentation, is also under constant surveillance via port holes and a video camera and recorder system. The constant observation allows NCEL engineers to refine test techniques as they are being performed, and a playback of the videotapes allows divers to modify procedures before field analysis in the unfamiliar open ocean.

A key element in the metals analysis procedure is an ultrasonic scanner bridge developed at NCEL. Attached to the test structure with four strong magnets, the scanner bridge positions the transducer close to the structure, and with a finely-tuned screw-feed mechanism slowly moves the transducer along the test cross-section.

Before actual test procedures begin, the engineering staff and divers will calibrate the electronic instruments and check the surface-to-diver communications system. Once instrumentation is ready for use, a diver will



Ron Erich holding the ultrasonic scanner bridge used in the underwater analysis of metal structures. The scanner bridge is attached magnetically to the test structure; using a screw mechanism a single transducer is moved along the surface, transmitting and receiving ultrasound information. This information is then transferred via hard wire to the desktop computer system and other electronic instrumentation for analysis. One of the video cameras for monitoring test procedures is to the left of Erich.

initially clean marine growth and exfoliating corrosion from the surface of the test structure with a high-pressure water blaster cleaning system. After the area is clean, the diver attaches the scanner bridge to the structure, and the ultrasonic evaluation can begin.

After receiving instructions from engineers in a land or ship-based control center, the diver will begin to move the transducer along the test surface with the screw-feed mechanism. As the transducer sends and receives high-frequency signals underwater, the information is simultaneously transferred to an ultrasonic flaw detector configured with a digital thickness adapter.

The digital thickness adapter provides an external analog voltage which is proportional to the metal thickness. This analog signal is then transmitted to a digital voltmeter which converts the thickness analog into a binary code. The binary thickness code is then transferred to the Tektronix desktop computer where it is either processed immediately, or stored on magnetic tape for future processing.

Programming for the metals analysis procedure was developed by NCEL staff members using packaged Tektronix software for portions of the analysis programs. As test data is fed into the system for processing, a graphic appears on the screen showing a variety of points plotted out on a specimen-length versus specimen-thickness graph (see figure). Initially, the graph contains a large grouping of points representing the range of material thickness, and a scattering of points representing errant ultrasonic echos off a corrosion-pitted front surface.

Since accurate measurement is the distance from the front surface to the back surface, NCEL engineers must eliminate the errant points from consideration, as they are not representative of overall material thickness. Using the cross-hair cursor feature on the Tektronix 4052, engineers reduce the analytical area to only those points which represent the accurate range of metal thickness. Once the area to be analyzed is defined on the screen, the numerical analysis section of the program begins.

This sequence of the program initially forms the individual plots into an averaged graph, more clearly reflecting the range of metal thickness. Once the averaged graph is complete, the computer generates accurate thickness values (within 0.001 ") and displays them on the CRT screen. In addition to the mean thickness value of the test-sample cross section, the computer provides a thickness range for the sample, number of ultrasonic readings taken of the sample, and standard deviation and coefficient of variation of all the readings.

"The analysis program we developed for our Tektronix system, allows us to eliminate data which misrepresents the actual thickness of our samples," says Brackett. "With the accuracy of our findings ensured, we can confidently recommend whether facilities are fit for continued use, or recommend repair or maintenance work."

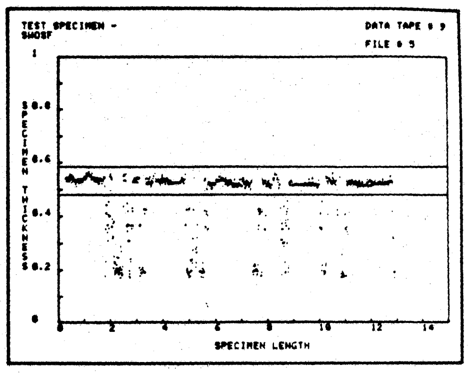
Underwater Test Procedures - Concrete and Timber

NCEL's underwater inspection program for concrete and timber structures relies on the general ultrasound principal; yet, because of differences in material properties and the desired end result, the specific test procedure is altered slightly. Internal damage caused by corrosion to concrete reinforcement, or by biological attack to timber pilings, is the focus of the concrete and timber programs.

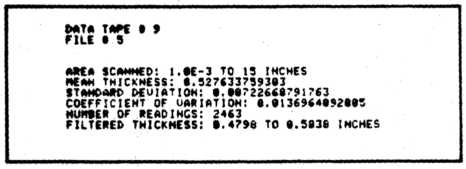
Erich, shown here with the assembly used in the "pitch/catch" or "through" transmission procedure used to determine internal defects of timber of concrete structures. In this procedure, ultrasound waves are sent from one transducer, through the specimen to the other transducer; after transmission, the information is analyzed with the desktop computer. Test tank is in background.

Unlike the metals analysis procedure in which only one side of the specimen was accessible, the entirety of the concrete or timber structure can usually be accessed by the diver.

"An ultrasonic technique known as 'pitch/catch,' or 'through transmission' is being evaluated to determine its ability to detect the hidden flaws in a wooden or concrete structure," said Kenney. "This technique relies on an ultrasonic transducer being placed on either side of the specimen. The ultrasound transmission passes from one transducer to the other, and the resulting transit time information is recorded on magnetic media and analyzed by the desktop system."



During analysis, points reflecting the thickness of steel H-pile are plotted out on a specimen thickness versus specimen length graph on the Tektronix 4052's CRT screen. With a cross-hair cursor (center) extraneous data (such as ultrasound reflections off corrosion-pitted steel surfaces) are eliminated, and only those points inside the cursor will be analyzed.



```
DATA TAPE # 9
FILE # 5

AREA SCANNED: 1.0E-3 TO 15 INCHES
MEAN THICKNESS: 0.527633759303
STANDARD DEVIATION: 0.00722668791763
COEFFICIENT OF VARIATION: 0.0136964092005
NUMBER OF READINGS: 2463
FILTERED THICKNESS: 0.4798 TO 0.5838 INCHES
```

The materials analysis program developed at NCEL automatically provides a variety of test information, including a thickness range across the test specimen. In this instance, the steel pile ranged in thickness from 0.4798 to 0.5838 inches.

Similar to the metals analysis procedures, the computer develops a graphic representation of the internal condition of the concrete structure or wooden piling. Such analyses can point out the internal damage caused to concrete structures by the corrosion of steel reinforcing rods, or the hidden biological attack caused to wood by marine boring animals. Once again, the accurate procedures will enable waterfront inspectors to recommend repair work or to recommend that use of a facility be discontinued.

Fast Drawing and Refreshed Graphics from 4050 Series Graphics Enhancement ROM Pack

by Mark Mehall
Tektronix, Inc.
Wilsonville, OR

Mapping, design, data representation and other graphics intensive applications will benefit from the new Tektronix Graphics Enhancement ROM Pack for the 4050 Series Desktop Computer. Adding a 4051R12 to a 4051 System or a 4052R12 to a 4052/52A or 4054/54A System speeds drawing rates and reduces memory space required for graphics data storage. New ROM Pack commands facilitate image formation, transformation and editing. A true full-screen crosshair cursor may be displayed on the 4051 or 4052/4052A as well as refreshed objects. ROM Pack utility routines read graphic images from tape, locate individual points within an image and determine minimum and maximum points. Sound and music generation are also supported by the ROM Packs.

Fast Graphics and Animation

Using the ROM Pack commands, the 4051 draws vectors approximately 27 times faster than when using the BASIC commands of MOVE and DRAW; dots are drawn up to 138 times faster. The display rates of the 4052 and 4054 are also improved.

Increased Drawing Rate
(4051R12/4052R12 vs.
BASIC MOVE/DRAW)

	4051	4052/4052A	4054/4054A
Vectors	27X	6X	18X
Dots	138X	39X	13X

The fast drawing rate of the Graphics Enhancement ROM Pack allows a limited amount of animation. Any refreshed object of approximately 12 vectors or refreshed strings of up to 15 characters can be generated on the 4051 or 4052/4052A without flicker. The standard 4054/4054A can display approximately 30 vectors in refresh using the 4052R12. A routine in the ROM Pack permits the user to move a refreshed object by using the joystick (4952 Opt. 2) or the thumbwheels on the 4054/4054A.

Special CROSS commands in the ROM Pack generate a full-screen crosshair cursor (from screen edge to screen edge) on the graphic display. The cursor may be manipulated without interferring with the displayed graphics. When moving the joystick, a "rubberband" line from the graphics cursor to the joystick position can be drawn to facilitate design or precise positioning.

Compact Storage in Strings

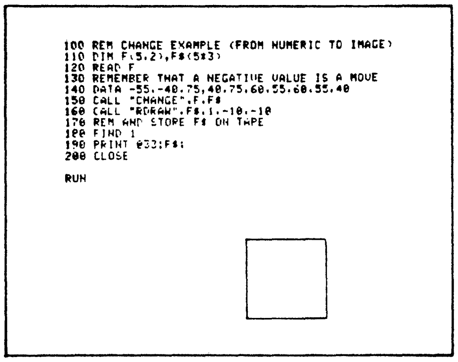
Because the Enhanced Graphics ROM Pack uses an image string rather than an array of X,Y coordinates, it can pack graphics data into 4050 memory in just one-eighth the space previously required. This means very complex drawings can reside in 4050 memory.

The image string used by a 4051R12/4052R12 represents each graphics point and the move/draw flag in three bytes compared to the 24 bytes required using standard floating point representation.* A special ROM Pack command converts coordinates (in graphic display units) into their binary equivalents and stores each pair and the move/draw flag internally as three ASCII characters.

				HI-X			HI-Y			
	1	0	F	X	X	X	Y	Y	Y	
Bytes	2	0	X	X	X	X	X	X	X	LO-X
	3	0	Y	Y	Y	Y	Y	Y	Y	LO-Y
		8	7	6	5	4	3	2	1	
					Bits					

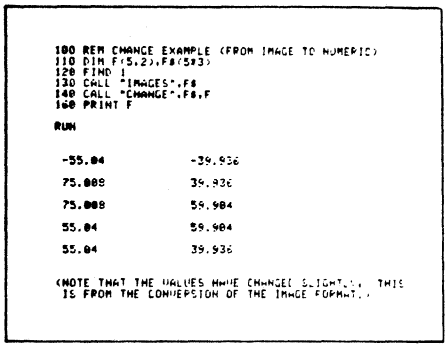
(F = 0 DRAW)
(F = 1 MOVE)

M/D Flag



```
100 REM CHANGE EXAMPLE (FROM NUMERIC TO IMAGE)
110 DIM F(5,2),F$(5*3)
120 READ F
130 REMEMBER THAT A NEGATIVE VALUE IS A MOVE
140 DATA -55,-40,75,40,75,60,55,60,55,40
150 CALL "CHANGE",F,F$
160 CALL "RDRAW",F$,1,-10,-10
170 REM AND STORE F$ ON TAPE
180 FIND 1
190 PRINT @33:F$;
200 CLOSE

RUN
```

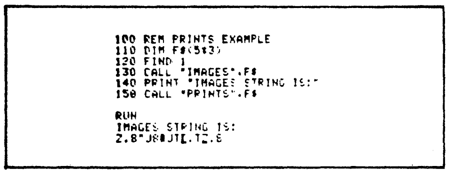
The CHANGE command will also convert the image string to standard numeric coordinates.



```
100 REM CHANGE EXAMPLE (FROM IMAGE TO NUMERIC)
110 DIM F(5,2),F$(5*3)
120 FIND 1
130 CALL "IMAGES",F$
140 CALL "CHANGE",F$,F
160 PRINT F

RUN

 -55.04            -39.936

 75.008            39.936

 75.008            59.984

 55.04             59.984

 55.04             39.936

(NOTE THAT THE VALUES HAVE CHANGED SLIGHTLY.  THIS
 IS FROM THE CONVERSION OF THE IMAGE FORMAT.)
```

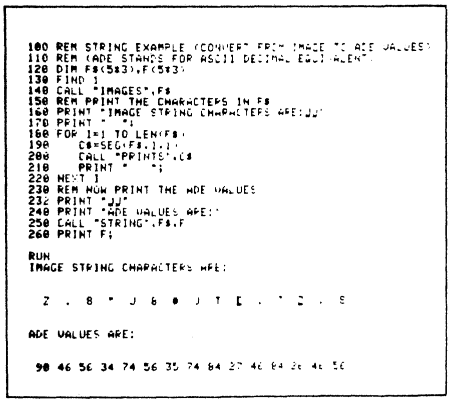
Image strings in the compact data format may be stored and recalled from cartridge tape or disk.

Useful for examining string context, a ROM Pack utility command prints any string on the screen and makes all control characters printable.



```
100 REM PRINTS EXAMPLE
110 DIM F$(5*3)
120 FIND 1
130 CALL "IMAGES",F$
140 PRINT "IMAGES STRING IS:"
150 CALL "PRINTS",F$

RUN
IMAGES STRING IS:
Z.8"J8@JT[.T\.8
```

Another utility command converts ASCII strings into their ASCII Decimal Equivalent (ADE) values.



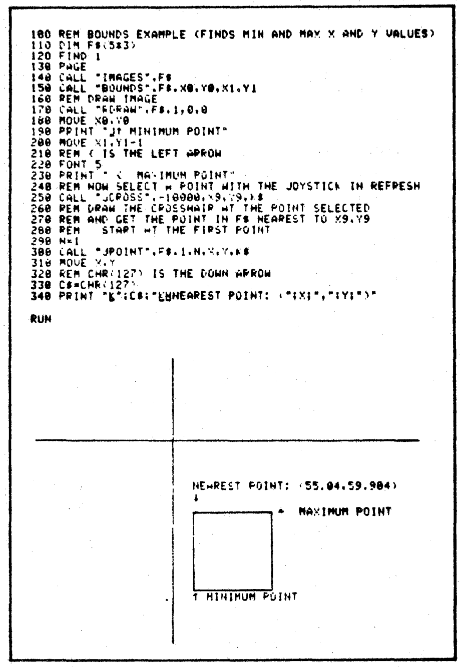
```
100 REM STRING EXAMPLE (CONVERT FROM IMAGE TO ADE VALUES)
110 REM (ADE STANDS FOR ASCII DECIMAL EQUIVALENT)
120 DIM F$(5*3),F(5*3)
130 FIND 1
140 CALL "IMAGES",F$
150 REM PRINT THE CHARACTERS IN F$
160 PRINT "IMAGE STRING CHARACTERS ARE:JJ"
170 PRINT "   ";
180 FOR I=1 TO LEN(F$)
190    C$=SEG(F$,I,1)
200    CALL "PRINTS",C$
210    PRINT "    ";
220 NEXT I
230 REM NOW PRINT THE ADE VALUES
232 PRINT "JJ"
240 PRINT "ADE VALUES ARE:"
250 CALL "STRING",F$,F
260 PRINT F;

RUN
IMAGE STRING CHARACTERS ARE:


  Z   .   8   "   J   8   @   J   T   [   .   T   \   .   8


ADE VALUES ARE:


 90 46 56 34 74 56 35 74 84 27 46 84 26 46 56
```

Image Formation

Strings of multiple characters representing many coordinates may be produced, and all of the 4050 Series BASIC string functions may be used with these image strings. For instance, SEARCH can be used to locate a vector; REPLACE and SEGMENT can replace and extract part of an image; the Concatenation Operator can join images, and so forth.

*The internal floating point representation was discussed in *TEKniques* Vol. 5, No. 4.

New commands in the Enhanced Graphics ROM Pack locate and return the minimum and maximum X,Y coordinates in an image string, facilitating scaling and windowing. Commands locating the X,Y coordinates in an image string closest to a specified X,Y coordinate location speed picture editing. For example, after inputting a point using the joystick crosshair cursor, a ROM Pack command can locate the picture coordinate closest to that of the cursor. Commands return the X,Y coordinate values of a particular point in an image string and any point within an image string can be changed or the move/draw flag modified to further ease picture editing.



```
100 REM BOUNDS EXAMPLE (FINDS MIN AND MAX X AND Y VALUES)
110 DIM F$(5*3)
120 FIND 1
130 PAGE
140 CALL "IMAGES",F$
150 CALL "BOUNDS",F$,X0,Y0,X1,Y1
160 REM DRAW IMAGE
170 CALL "RDRAW",F$,1,0,0
180 MOVE X0,Y0
190 PRINT "J↑ MINIMUM POINT"
200 MOVE X1,Y1-1
210 REM < IS THE LEFT ARROW
220 FONT 5
230 PRINT " <  MAXIMUM POINT"
240 REM NOW SELECT A POINT WITH THE JOYSTICK IN REFRESH
250 CALL "JCROSS",-10000,X9,Y9,K$
260 REM DRAW THE CROSSHAIR AT THE POINT SELECTED
270 REM AND GET THE POINT IN F$ NEAREST TO X9,Y9
280 REM   START AT THE FIRST POINT
290 N=1
300 CALL "JPOINT",F$,1,N,X,Y,K$
310 MOVE X,Y
320 REM CHR(127) IS THE DOWN ARROW
330 C$=CHR(127)
340 PRINT "K";C$;"KHNEAREST POINT: (";X;",";Y;")"

RUN
```

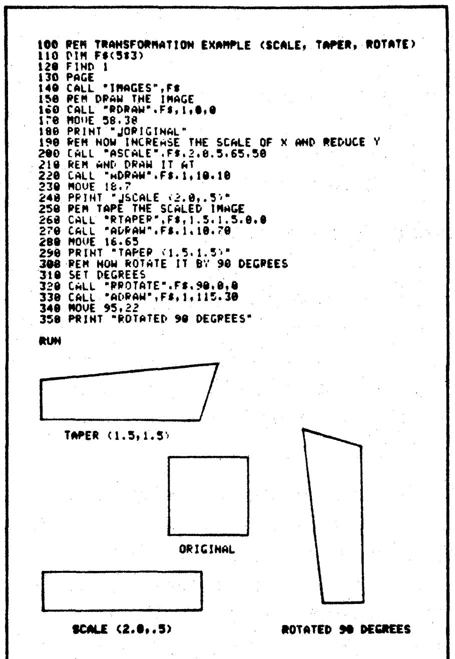
Image Display

Freeing the programmer from software calculations, the Graphics Enhancement ROM Pack commands quickly and easily position or transform images on the screen.

An image may be displayed at an absolute coordinate location. Or it may be displayed relative to its original location, relative to the current graphic display cursor, or relative to the joystick cursor. But repositioning and displaying the image requires only one command; the calculations to redefine each point of an image to the new coordinate value are automatically performed by the ROM Pack.

Graphic transformations are provided by the ROM Pack for scaling, shearing, tapering and rotating graphics images. An image may be independently scaled or tapered horizontally and vertically. Negative scale or taper values generate mirror images. And, like the image display, only one command initiates the transformations.

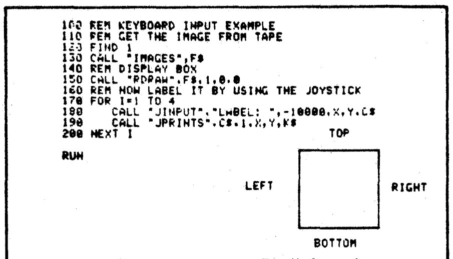
Because transformations alter the image string, multiple operations may be concatenated.



```
100 REM TRANSFORMATION EXAMPLE (SCALE, TAPER, ROTATE)
110 DIM F$(5*3)
120 FIND 1
130 PAGE
140 CALL "IMAGES",F$
150 REM DRAW THE IMAGE
160 CALL "RDRAW",F$,1,0,0
170 MOVE 58,30
180 PRINT "JORIGINAL"
190 REM NOW INCREASE THE SCALE OF X AND REDUCE Y
200 CALL "ASCALE",F$,2,0.5,65,50
210 REM AND DRAW IT AT
220 CALL "ADRAW",F$,1,10,10
230 MOVE 18,7
240 PRINT "JSCALE (2.0,.5)"
250 REM TAPE THE SCALED IMAGE
260 CALL "RTAPER",F$,1.5,1.5,0,0
270 CALL "ADRAW",F$,1,10,70
280 MOVE 16,65
290 PRINT "TAPER (1.5,1.5)"
300 REM NOW ROTATE IT BY 90 DEGREES
310 SET DEGREES
320 CALL "RROTATE",F$,90,0,0
330 CALL "ADRAW",F$,1,115,30
340 MOVE 95,22
350 PRINT "ROTATED 90 DEGREES"

RUN
```

Note that each succeeding transformation builds on the earlier one.

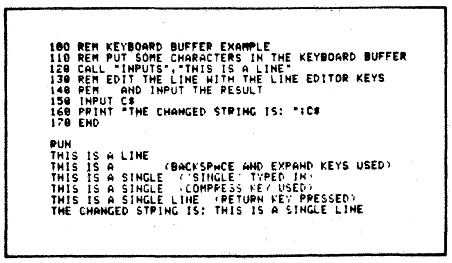
ROM Pack commands will display a prompt string while simultaneously accepting and printing keyboard input to facilitate graphics interaction (such as run-time labeling) with the 4050 operator.



```
100 REM KEYBOARD INPUT EXAMPLE
110 REM GET THE IMAGE FROM TAPE
120 FIND 1
130 CALL "IMAGES",F$
140 REM DISPLAY BOX
150 CALL "RDRAW",F$,1,0,0
160 REM NOW LABEL IT BY USING THE JOYSTICK
170 FOR I=1 TO 4
180    CALL "JINPUT","LABEL: ",-10000,X,Y,C$
190    CALL "JPRINTS",C$,1,X,Y,K$
200 NEXT I

RUN
```

Notice that the prompt "LABEL:" doesn't store.

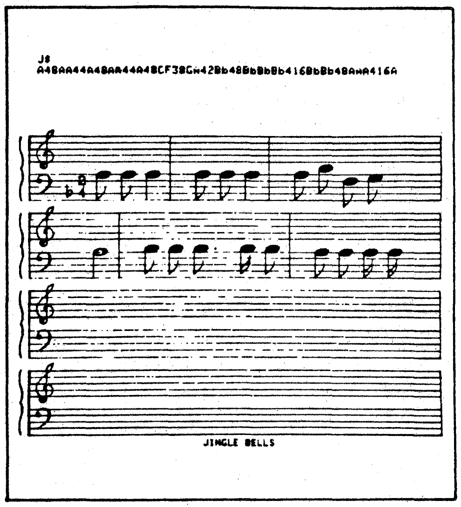
Up to 28 characters may also be entered into the keyboard buffer through the program, allowing the operator to use the Line Editor keys for corrections and changes.



```
100 REM KEYBOARD BUFFER EXAMPLE
110 REM PUT SOME CHARACTERS IN THE KEYBOARD BUFFER
120 CALL "INPUTS","THIS IS A LINE"
130 REM EDIT THE LINE WITH THE LINE EDITOR KEYS
140 REM   AND INPUT THE RESULT
150 INPUT C$
160 PRINT "THE CHANGED STRING IS: ";C$
170 END

RUN
THIS IS A LINE
THIS IS A          (BACKSPACE AND EXPAND KEYS USED)
THIS IS A SINGLE  ('SINGLE' TYPED IN)
THIS IS A SINGLE  (COMPRESS KEY USED)
THIS IS A SINGLE LINE  (RETURN KEY PRESSED)
THE CHANGED STRING IS: THIS IS A SINGLE LINE
```

Music and Sound Utilities

Two ROM Pack commands generate music and sounds. The MUSIC command accepts parameters identifying the tempo, the octaves, the notes, and the time value of the notes. Ten different tempos are available and music can be played with a pitch range of eight octaves. Duration can be specified from 1/64 to a whole note.



```
J$
A48AA44A48AA44A48CF38GA42Bb48BbBbBb416BbBb48AAA416A
```

Useful for making attention-getting sounds (nonmusical) as audible cues of certain functions in progress or just completed, the SOUNDS command translates ASCII character values into frequencies at which the internal 4050 Series speaker is driven. Voice patterns can be simulated. SOUNDS can produce audio frequencies from approximately 100 Hz to 13,000 Hz.

A 4050 Series Graphics Enhancement ROM Pack adds 65 new commands to a 4050 Desktop Computer. Whether a 4051, 4052/52A or 4054/54A, the ROM Pack will extend and speed its interactive graphics capability.

New ROM Packs, Interfaces and Peripherals Expand 4050 Desktop Computer Capability

Adding to their already broad range of ROM Packs, interfaces and peripherals for the 4050 Series, TransEra recently introduced several software packages, a new File Manager, a GPIB Multiplexer and a Streaming Tape Drive.

CP-100 Contour Package

Centered around a special function ROM Pack, the CP-100 Contour Package produces labeled contour maps from random or gridded data. In the ROM Pack, an efficient machine coded algorithm tracks contour levels from gridded data many times faster than any high level software. The ROM pack also accepts random data which is converted to grid format.

Features include smoothing of the lines for a hand contoured appearance, labeling of contour lines, emphasizing selected lines, scaling to allow large plots to be sectioned for high resolution, and marking and labeling the boundaries of the plot.

Data input and output can be to magnetic tape or disk. Output to the 4662/4663 plotters with various formatting options is supported.

PC-100 Printed Circuit Layout Package

The PC-100 Printed Circuit Layout Package is a high performance ROM Pack driven software package for generating photo reduction quality printed circuit board artwork. The package uses the 4054 keyboard for a digitizing tablet for input and a variety of plotter for output including Tektronix 4662 and 4663 and plotters supported by the TransEra Plotter Interface ROM Packs.

The PC-100 ROM Pack contains several routines for managing the data base generating symbols and stroked characters with variable size and orientation, as well as a number of utilities to improve the user interface and execution speed. The software provides a friendly user interface with single key command entries and extensive editing capabilities. Some of the key features of the package are:

- a complete library with both standard and user-defined shapes
- up to 10 trace layers for complex boards
- a menu driven interface
- variable scaled artwork ready for photo reduction
- diagram and list utilities to assist in PC documentation

Streaming Tape Drive

The Streaming Tape Drive provides archival backup for TransEra 6000 Series Winchester disks on low cost cartridge tapes, storing up to 14 Megabytes on a single tape.

The streaming nature of the tape drive allows for a data transfer rate of 3.3 Megabytes per minute. An entire unit of a 6010 or 6020 Micro Disk can be backed up in 75 seconds. In 2.5 minutes an entire unit of a 6040 Micro Disk can be backed up. The data on the tape is stored in 512 byte blocks with CRC error detection performed on each block.

GPIB Multiplexer

The TransEra GPIB Multiplexer allows any two IEEE-488 compatible bus controllers to communicate alternately with a set of peripherals connected to the Multiplexer output port. This permits two Tektronix 4050 Desktop Computers to share any GPIB compatible device or set of devices.

A GPIB device itself, the Multiplexer allows a controller to read status information and write commands that capture and release control of the bus. Two host controllers may establish a protocol whereby each in turn reads bus availability, waits for it, captures the bus when ready, and releases the bus when finished. 4050 Systems control the Multiplexer through WBYTE and RBYTE commands.

710 File Manager

4050 users who interface to TransEra 6000 Micro Disks may allocate disk space in smaller files with the new 710 File Manager from TransEra. Previously, the 700 File Manager (4052/4054) allocated space in chunks of 10,752 bytes. Better control of disk space for those who require smaller files can be obtained through the new 710 File Manager ROM Pack.

For more information on these items, contact Burt Burrell at TransEra Corporation, 3707 North Canyon Road, Suite 4, Provo, UT 84604, (801) 224-6550.

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.

Computer Graphics Standards: Where They Are

by Dave Straayer
Tektronix, Inc.
Wilsonville, OR

The idea of computer-graphics standards and device independence was born in the early 1970s. At that time, the methods and functions used to draw pictures on various systems were very different, and not all these differences were necessary.

Usually programs were developed to be device dependent; as new graphics devices become available, users attempted to convert their software for use with these often less expensive and better performing devices. This conversion frequently proved difficult and required assistance from programmers familiar with both devices.

In the mid-1970s, the notion of standards and device independence became sufficiently popular to inspire work to set standards.

Core - A Beginning

A major milestone in computer-graphics standards was the publication in 1975 of the *Status Report of the Graphics Standards Planning Committee* (GSPC), Siggraph (Special Interest Group - Computer Graphics). This report consisted mainly of a straw-man proposal, which came to be known as the Core.

The Core was several things. It served as a vehicle for discussion and a feasibility study, and for the educational community it formed the basis for courses on computer graphics. To computer-graphics vendors, it became a de facto standard. To others still, it represented a prototype American National Standard, to be refined in the American National Standards Institute (ANSI), eventually to become a formal standard.

Enter GKS

From 1977 through 1979, several important developments occurred. For one, DIN (*Deutches Institute fur Normung*), the official standards-making body of West Germany, undertook to develop a national standard for computer graphics based on the GSPC Core and other work. This standard was called GKS (Graphical Kernel System) and was adopted in Germany.

Another important development was the refinement of the GSPC Core published in 1979.

That period also saw the establishment of ANSI X3H3, a committee charged with the development of computer-graphics standards and initially formed to adopt the GSPC Core as an American National Standard. (The X3 division of ANSI oversees the development of computer-related standards.) X3H3 first met in the fall of 1979, just as the GSPC was dissolving.

About the same time, ISO began work on adopting GKS as an international computer-graphics standard.

X3H3 established five task groups and revised its charter. These subcommittees began developing four different standards proposals and working in conjunction with the International Standards Organization (ISO).

ANSI X3H31, given the charter for revising the Core to an ANSI draft, reexamined some of the basic assumptions and decisions on which the Core was based. It worked to build a more comprehensive model of computer graphics on which to base its standard.

Other ANSI X3H3 participants, sensing the need to adopt standards quickly proposed standards that were less comprehensive in their scope and could be adopted more readily.

PMIG - A Subset

ANSI X3H35 began as a group attempting to identify a smaller subset of the Core (which was then to be the basis of ANSI standards) than previously identified. Its goal was to design a simple approach to minimal graphics, called the Programmer's Minimal Interface to Graphics (PMIG).

PMIG's constituency includes the users of microprocessor-based personal computers for whom graphics is not the principal aim of their work but just one aspect; these users demand simplicity and directly controlled attributes.

The committee originally expected to develop an independent American National Standard (ANS) for PMIG but altered its plans.

The recent acceptance by ISO of the simpler directly controlled attributes has caused the PMIG committee to reconsider the possibility of meeting the needs of the PMIG constituency under the GKS umbrella. As a result, the work of ANSI X3H35 has been redirected toward identifying a smaller GKS output level than the current level 0. The PMIG group has taken the initiative in ANSI to prepare a version of GKS with this level identified and to pursue the paperwork necessary to have it adopted as an American National Standard.

GKS - ANSI Style

In addition to identifying a smaller output level, ANSI is rewriting Annex C - which isn't a formal part of the GKS standard - to conform to U.S. requirements. ANSI is also standardizing an aspect of GKS that's not standardized in the ISO version: the Aspect Source Flags. These steps will result in an American National Standards version of GKS that's fully compatible with ISO GKS, but better suits some of the needs of the U.S. computer-graphics community.

The American National Standard version of GKS will also include a binding of GKS to FORTRAN. This binding will be based on work begun in Germany under DIN but will address U.S. objections to what is perceived in this country as poor cognitive ergonomics in the proposed DIN binding.

VDI and VDM - A Different Level

ANSI X3H33 proposed the development of a standard at a different level. (Its proposals are more appropriate to serve as device standards than Core or GKS, which were intended to be software standards.) This particular proposal is called the Virtual Device Interface (VDI) standard.

Plans to develop a VDI standard have been approved by ANSI and have received considerable attention from vendors of graphics products. Intel, Tektronix, and Digital Equipment Corp., for example, have publicly announced their intention to support the standard's development and adopt it in their products.

The committee is also developing a closely related proposal, the Virtual Device Metafile (VDM). Although approved for development as a separate standard, it is intended to be an adaptation of the VDI proposed standard to a file format, suitable for recording and interchanging pictures among different systems. The X3H33 group is concentrating on completing the metafile work first, in part because ISO has agreed to accept its work as a basis for ISO standards development.

Other Standards

IGES. One of the interesting aspects of graphics standards is the way they appear where least expected. With intense activity both in ANSI X3H3 and ISO, you might expect that graphics standards would be proposed from these arenas alone. However, some important exceptions exist.

The Initial Graphics Exchange Standard (IGES), for instance, is now part of the adopted ANSI Y14.26M American National Standard. It's often confused with metafile proposals (with some good reason). However, this standard is intended to provide a way to exchange design files between different CAD/CAM systems. The standard was encouraged by the National Bureau of Standards and aircraft manufacturers, who were frustrated by the difficulty of communicating between different vendors' CAD/CAM systems. Adopted in the fall of 1981, IGES has already been supported by many vendors. And at the 1982 NCGA (National Computer Graphics Association) Conference, a very impressive demonstration was staged. A design was passed around the exhibit floor in IGES standard tape format, and vendors changed the design to demonstrate the system's utility.

Is IGES a graphics standard? In the general sense suggested by Figure 1, it is. Most previous thought has been toward standards at the level of Core, GKS, and PMIG, and IGES, thus wasn't perceived as a graphics standard by everyone. Is IGES compatible with other standards? Certainly it's as compatible as it was expected to be, but there are differences that are resolved in software layers. Finally, there are no insurmountable differences separating IGES and GKS or VDI.

Videotex and NAPLPS. Graphics standards have also emerged from another quarter – Videotex. From a simplistic point of view, Videotex is a way of hooking up television sets (or raster terminals like television sets) to telephone lines (or communication grids like telephone lines) so that both private and business consumers can access public commercial data banks. Most of the pioneering work in this technology occurred in Europe.

Originally, the intent of Videotex was to get text frames on the user's screen. It soon became apparent, though, that Xs and blanks could be used to make simple pictures, much as simple charts and graphs have been drawn on line printers. This approach evolved into the higher quality concept of alpha-mosaics, a set of 64 characters containing the 64 combinations possible in a 2 × 3-character cell.

The Canadians adapted the idea and made the natural extension of adding graphics primitives: lines, areas, and pixels. In the U.S., AT&T has examined the Canadian technique and decided that this standard could be important to future telephone company business. AT&T delivered a document describing the Canadian method to ANSI and announced the firm's intention to adopt and adhere to this proposal. AT&T suggested that the method would make a good ANSI standard but that AT&T would use it in any case.

Under the name North American Presentation Level Protocol Syntax (NAPLPS), the proposal is now being adopted in the U.S. by ANSI and in Canada by the Canadian Standards Association.

Computer-graphics vendors are now becoming interested in these standards because it appears very likely that NAPLPS will be widely available in many low-cost graphics terminals. As a result, application software will be developed to utilize the projected large numbers of popular low-cost NAPLPS terminals. Moreover, Intel, Tektronix, and Digital Equipment Corp., followed by about a dozen other companies, announced their support of the NAPLPS standard.

This standards impact won't be immediate, but it could be far-reaching – Videotex could well become an important information channel for the entire would, rivaling television and radio in their cultural impact.

For years computer-graphics standards have seemed just two or three years away. That is no longer the case: they are here now. With standards, computer graphics can become the workhorse of communication.

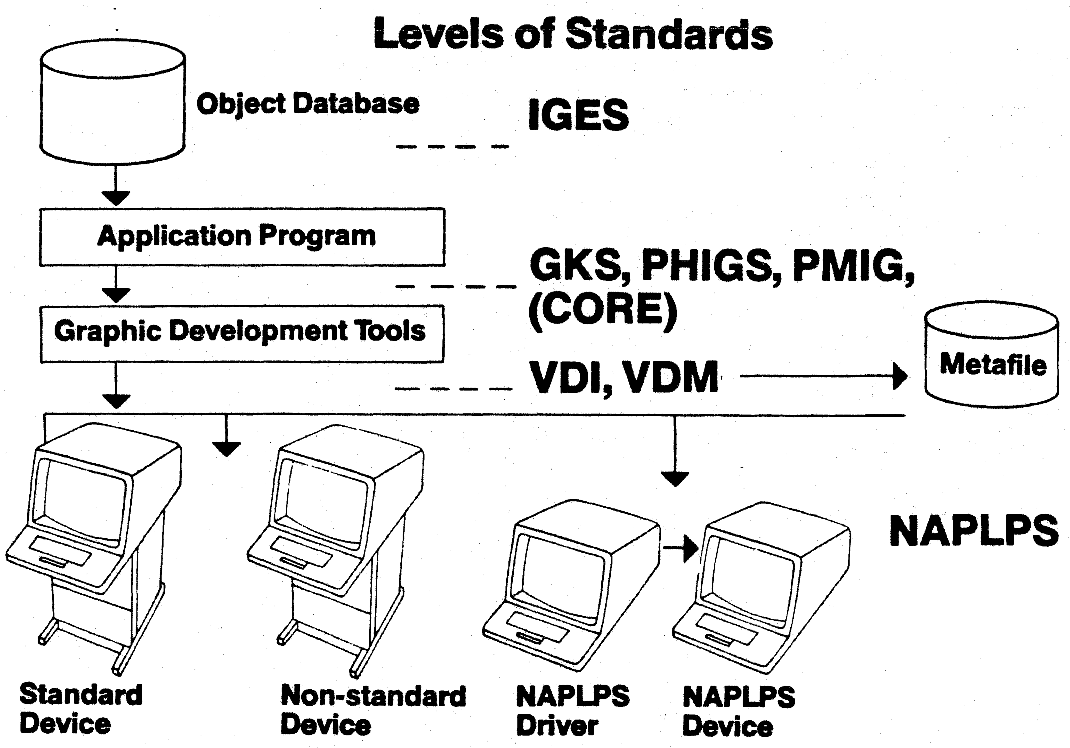


Figure 1.

Where Standards Fit, What They Are

Figure 1 is a conceptual model of computer-graphics systems, designed to illustrate the concepts of levels and interfaces.

IGES

At the highest level are application data bases where the models consist of not only the geometric data that a program can interpret to produce pictures but also of data describing properties of components. A standard for an interface to such data bases called IGES (Initial Graphics Exchange Standard) has been formulated. It is not so much a standard for an interface between a data base and an application program (as suggested by Figure 1) as a standard data base format that can be used to exchange design data between design systems.

IGES facilitates transfers of CAD/CAM objects (both geometric and the attached meanings) between application-specific data bases. An object created on one CAD/CAM system can be passed to a different CAD/CAM system for processing.

GKS, Core and PMIG

Programs that do useful work in graphics systems reside at the application-program level. Typically application programmers have at their disposal a library of subroutines to draw pictures (Call Plot, Call Draw, Call GPL, etc.). Those subroutines are at the next level, the graphics-utility level. Between these levels, standards such as GKS, Core, ANSI X3H31 Rich System, and PMIG reside. Core and GKS functionality is close. Most differences lie in workstation concepts.

A 2-dimensional standard with different levels for both output capabilities and input functions, GKS innovatively covers several areas. It provides numerous functions for controlling multiple workstations, for example, approaching those functions at the application-program/utility-system level.

Workstations and Attributes

The GKS model stresses graphics pattern, as ways to distinguish data, rather than as intrinsic aspects of pictures. Thus, these attributes, such as line color, line thickness, dash style, graphics text geometry and others, are bundled in GKS in a way that makes it convenient for different parts of pictures to be distinguished by use of different attributes on different workstations. This approach is natural and powerful for many applications, especially graphs and charts for which distinguishing of data types is a primary goal.

This concept isn't completely foreign to U.S. users. On pen plotters, for example, pen width is inherently bundled with color by the choice of a pen. It does differ, however, from the more common model of device independence previously used in Europe and the U.S. in which the application describes how the picture should appear; and each workstation strives to make as true a representation of that description as possible. For example, in the Core proposal, the defined attributes apply to any device addressed and attributes must be adjusted manually if the defaults are unsatisfactory.

GKS was modified, however, to meet suggestions by ANSI to embrace both directly controlled, i.e., as part of the picture, and bundled attributes.

Language Binding

To understand binding, it's necessary to know that graphics standards, unlike normal programming language standards, are functional rather than syntactic. For example in the FORTRAN standard, a DO loop is described in terms of its syntax, which specifies such conditions as where a statement number must appear and the placement of equal signs and commas. Some functional specification is also involved: the number of times the DO is performed and how that number is calculated are functional considerations, but the emphasis is on the syntax.

In contrast, the graphics standard includes a polyline function which has as an argument an array of 2-dimensional coordinate pairs, or points. It produces a series of straight line segments, connecting the points given as arguments. This approach might be bound in FORTRAN to GPLN, PX, PY, where N is a count of points and PX and PY are real arrays. However, this function might also be bound as <gs> <x-y> <x-y> <x-y>... <delimiter>, where <gs> is the ASCII GS character, <x-y>'s are strings of ASCII characters in the Tektronix PLOT 10 sequence, and <delimiter> is an ASCII US or CR character. How should these bindings be determined? Should they be standardized?

Conformance is important in such standards and refers to the issue of how to determine whether a vendor's product does indeed conform to a functional standard, with or without an appropriate binding. No formal standards were set for testing conformance to the Core; hence, there were many Core-like systems and systems described as being "in the spirit of the Core."

For a standard to be workable, it needs to be bound to a language. The GKS draft standards has published the specifications for commands in ANSI FORTRAN 77. As a result, applications using GKS at one installation can run at any other even if the host computer and graphics terminals differ.

PMIG

At the low end of GKS functionality, PMIG is intended to serve users who need graphics on systems not dedicated to graphics applications, such as personal computers.

PMIG most likely will be a subset of GKS with about 30 functions.

Coming Events

Further, ANSI X3H31 is working to develop a standard for PHIGS (Programmer's Hierarchical Interactive Graphics System), which will include hierarchical structures, a more powerful attribute model and 3-dimensional viewing. Hierarchical structures are segments that contain other segments. The concept is a fundamental departure from the conceptual model of separating modeling and viewing.

To understand such structures, suppose that you describe to a graphics system how to draw an apple and want the system to remember on its own how to draw an apple. (You must place the description in a retained segment.) Now you can make the apple visible; make it invisible; and scale, rotate, and drag it around the screen. This capability is compatible with the Core GKS segmentation mode.

Now suppose you create a segment that contains instructions to draw a tree. Because the graphics system remembers how to draw an apple, you might say something such as "Put a lot of little apples in the tree at the ends of each branch." This type of instruction constitutes segment inclusion; the creation of a hierarchical segmentation structure and this capability are intentionally missing in Core and GKS.

When such functions are present, important questions must be answered. For instance, what happens if you delete the apple segment? And what if you create a pear segment and rename it apple? Although these questions might appear easy to answer, they require unambiguous answers. And controversy exists about the potential answers to these questions. The key problem is how to reconcile this standard with GKS. Should the two be compatible, and if so, can they be? Most important, can such a standard be written and adopted in a reasonable amount of time?

VDI and VDM

The next level is somewhat more complex. Device-independent graphics-utility systems usually indicate clearly the difference between device-independent and device-dependent code. The device-dependent part of such a system is often termed the device-driver section.

The Virtual Device Interface proposed by the ANSI X3H33 Committee is seen by many as as a standard for this interface. However, note that different devices can share functional characteristics and thus actually be standard devices. Such devices wouldn't need device drivers to convert standard functions into a series of nonstandard commands, the task handled by most current device drivers.

The only difference between the standardized functions at the device-driver level and the actual device level would be differences in binding, not function. Therefore, simple mechanical translators could convert internal floating-point representations of numbers to strings of ASCII characters. Different devices conforming to a Virtual Device Interface standard and also conforming to a standard binding would be interchangeable, at least to the extent that they perform all the functions needed by the applications using them.

How much diversity among graphics devices is necessary? Too many mutually exclusive features such as cost and resolution are needed, making it impossible for all graphics devices to have exactly the same capability, resolution, and speed. As long as there are strong economic reasons to provide inexpensive monochrome plotters and displays, for example, ultimate standardization of graphics devices is unattainable. Each standards proposal deals with this diversity through levels, required and nonrequired functions, or limited scope.

The Virtual Device Interface and Virtual Device Metafile proposals have generated much interest in the U.S. where a large percentage of the world's computer-graphics-device manufacturers are based. Those proposals use the concept of required and nonrequired functions to address the diversity of different graphics devices, rather than the levels or subsets approach used by Core and GKS, which are more appropriate at a software level.

The bundled attributes of GKS pose a problem to the X3H33 group. The original intent was to have only directly controlled attributes in VDI and VDM, which reflects current practice in actual devices; it was thought that GKS could be built on top of a VDI with directly controlled attributes.

However, these assumptions have been reexamined and found too limited. The approach works well for simple graphics devices, but for more sophisticated types that maintain their own segment lists, the Virtual Device Interface must embrace the same attribute models as GKS if it's to succeed in relieving the host computer of the segment storage task. This feature is necessary because of the dynamics implied by the ability to redefine a bundle's contents and then redisplay a segment with a different appearance. The common-attributes requirement has caused X3H33 to add both attribute models to its proposed VDI and VDM standards.

Most important to U.S. vendors, the VDI standard is compatible with current trends in graphics devices. VDI thus gives manufacturers two capabilities. They can provide software interfaces for their current devices to make those devices appear standard to GKS and other systems and they can design new devices that directly meet the VDI standard without major functional changes.

Tektronix PLOT 10 GKS Supports ISO Computer Graphics Standard

by Will Gallant
Tektronix, Inc.
Wilsonville, OR

Tektronix has always championed the development and support of standards for computer graphics to protect the software and hardware investments of users. Extending this commitment is the Tektronix PLOT 10 Graphical Kernel System (PLOT 10 GKS), a host- and device-independent library of routines for graphics and text interaction. PLOT 10 GKS fully implements the GKS standard at Level 2B as defined by the Draft International Standard ISO/DIS 7942, with language binding to ANSI FORTRAN '77. (See related article in this issue of *TEKniques.*)

Experience Produces Sound Products

Although a published specifications document defines GKS, the internal architecture is left to the skill of the software engineer.

PLOT 10 GKS is a fine tuning and consolidation of Tektronix experience gained from implementing the well known "Core" programming methodology. A proposed standard of ACM SIGGRAPH for approaching graphics applications, the Core's concepts for host- and device-independence were implemented in the Tektronix PLOT 10 Interactive Graphics Library (PLOT 10 IGL) developed years ago. The fundamental concepts of graphics primitives, attributes, graphics segments, transformations, and device drivers were accepted and proven by thousands of PLOT 10 IGL real-world applications. PLOT 10 GKS, consequently, is not so much new as it is solidifying accepted industry practice.

PLOT 10 GKS gives the applications programmer all the software tools to work with diverse input and output devices. Incorporated in the more than 50,000 statements that make up GKS are programmer callable routines for two-dimensional graphics including graphics segments, software text fonts, and control of multiple input or output surfaces, each with its own window and viewport.

Like PLOT 10 IGL, PLOT 10 GKS has all the fundamental graphics functions - line drawing styles, rotation, scaling, and so forth - but as they have been defined by ISO in the GKS draft standard.

Graphics segment operations are inherent in GKS architecture as they were in the Core. Tektronix has drawn from its extensive experience in segment management developing PLOT 10 IGL to implement segments in PLOT 10 GKS.

Because PLOT 10 GKS is optimized for handling a large number (over 5000) of segments per session, its segment operations are extremely fast. Even if the upper number of segments is reached by the applications program, PLOT 10 GKS applies a fast segment access scheme (to host disk storage) to improve performance and reduce processing costs.

Character Fonts

A large collection of software character fonts is provided by PLOT 10 GKS.

Workstations and Attributes

A major difference between Core and GKS is in the concept of a workstation. In the traditional approach, a workstation was treated as a collection of hardware devices cabled together. The controlling software knew of one WINDOW and one VIEWPORT with one set of display attributes.

GKS, on the other hand, considers a workstation as one active display or input device, with each workstation having its own WINDOW/VIEWPORT. In other words a "system" consists of multiple workstations such as a CRT monitor, a digitizing tablet and a plotter. Furthermore, each workstation has its own collection of display attributes - attributes defining line color, line thickness, dash style, character geometry and color, plus others - and attribute bundles have associated workstation numbers. Applications programs activate and deactivate input and output devices (workstations) as activity moves from tablet, to screen, to plotter, without the need to reset the attributes for each.

Bundling of attributes is a work saver for the applications programmer. And PLOT 10 GKS uses a very efficient memory management technique to handle workstation attributes.

Device Independence

Device independence in GKS is accomplished by three important architectural features: interchangeable device drivers, a metafile and language binding.

Device Drivers. An applications program will call GKS routines to draw lines, fill polygons, position text and so forth. The device driver communicates with the display terminal in the terminal's unique protocol to produce lines, filled polygons, text, etc. Only the device driver must change to support a particular display device, not the applications code.

PLOT 10 IGL or PLOT 10 GKS: Which is Appropriate

PLOT 10 IGL and PLOT 10 GKS are as similar in form and purpose as a Ford and Chevrolet automobile. Like the automobiles, though intended to serve the same function, they do not have interchangeable parts.

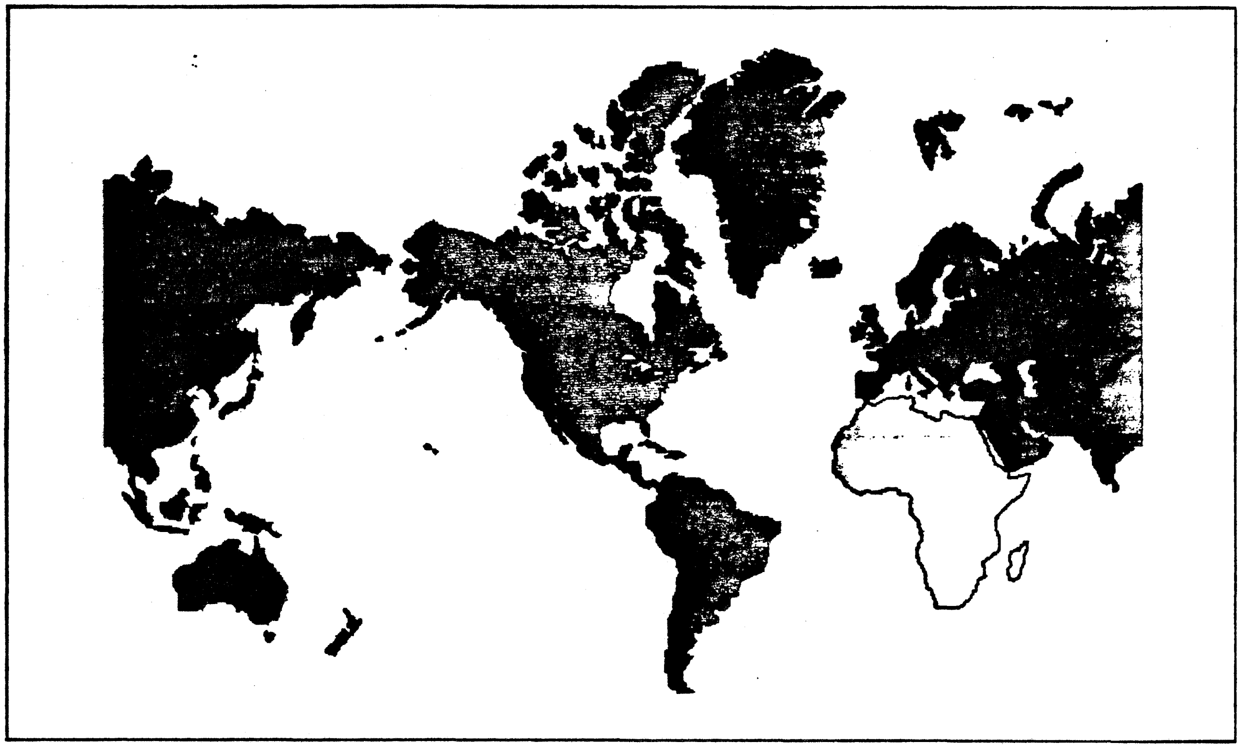
PLOT 10 IGL has 3-D graphics today. 3-D certified for GKS has yet to be proposed to the standard bodies, consequently, GKS simply may not provide enough functionality for all users.

Those installations who require greater functionality and who are not concerned about strict adherence to a particular standard will opt for Core-based PLOT 10 IGL. If the requirement is for 3-D, then PLOT 10 IGL is the recommended development tool.

In PLOT 10 GKS device drivers are provided for Tektronix displays, digitizing tablets and plotters. Support for the Tektronix 4010 and 4020 Series of terminals continues as well as the 4110 and 4100 Series, 4662 and 4663 plotters.

Features available in the 4110 Series and the newly introduced 4100 Series of Tektronix Terminals are supported in specially designed drivers for these devices. For instance, PLOT 10 GKS will draw thick lines using panel fill firmware in the raster terminals which cuts data transmission activity to only four coordinates. But for purely vector devices such as DVST terminals and plotters, the PLOT 10 GKS drivers will lay down a series of parallel lines to effect a thick line.

Line color specification is one of the GKS functions. For the Tektronix 8-pen plotter,



this causes automatic pen selection. For single pen plotters, an operator prompt will appear in the dialog area of the terminal screen to request a pen change.

In any case the device driver does what is expected isolating the applications program from hardware differences. A rule of thumb used in device driver engineering is to "apply the theory of least surprise."

A model driver is also included which offers the professional programmer a prototype from which to develop device drivers for non-Tektronix displays required in the application, such as E-size plotters or COM recorders.

Metafile. Thought of as a device driver which can store pictures in a device-independent manner, the Metafile stores pictures in ASCII characters rather than machine-dependent binary format. Pictures may be called back from the Metafile to appear on any display via its device driver. Metafile data, thus, may be easily passed between computer centers. Physically, the Metafile is usually disk storage. Binary data storage is also supported, if desired, for optimal data compaction and access speed.

Language binding. The major benefit of language binding is a uniform collection of graphics commands. GKS fixes the FORTRAN commands for GKS functions. Applications written in GKS for IBM FORTRAN '77 should run equally well on DEC, Univac, Honeywell, Prime and other computers with FORTRAN '77 compilers. (Since ANSI FORTRAN '77 compilers are not universally available, the GKS draft standard offers alternative commands which allow GKS to run in a FORTRAN '77 subset environment with certain character handling extensions.)

PLOT 10 GKS meets the specifications of level 2B and will pass subroutine name convention certification.

GKS Subsets

Flexibility and efficiency are inherent in PLOT 10 GKS. If an application requires only a few features of PLOT 10 GKS, only those features are called from the object library to join the applications code. Applications programmers can spend more time on problem solving and less on worrying about trimming excess code from their end product.

One full object library may be shared on a computer without duplication for use in large and small programming projects. One package means less maintenance and sharing of learned programming skills.

Subsets of GKS are defined by the ISO draft standard. Applications written to a subset specification of GKS will find all the commands it requires in the full level 2B Tektronix version. As a result a program that doesn't require a full set of GKS commands will extract and load only a portion of PLOT 10 GKS, conserving computer memory space.

From installation to long range operation, PLOT 10 GKS is engineered to be friendly, reliable and maintainable. A pre-processor selects device drivers and a few other parameters when PLOT 10 GKS is installed on the host. A library configured with the desired device driver is then passed to the FORTRAN compiler.

Software is only as good as its documentation. PLOT 10 is shipped with a clearly illustrated User manual, Quick Reference Guide, Installation Guide, and the formal GKS Draft International Standard ISO/DIS 7942. To assist a programmer to understand concepts detailed in the manual, all coded illustrations are included in files on the accompanying tape. The reference card provides a quick reference to command definitions.

Tektronix-hosted programming classes on PLOT 10 GKS assures learning the tools in the shortest possible time before beginning a large applications project.

A continuing program of product maintenance, called the Tektronix Software Subscription Service, updates PLOT 10 GKS with new device drivers and general improvements.

Applications founded on PLOT 10 GKS will not have to undergo exhaustive conversion to run on another vendor's GKS of the same or higher support level. And applications written for any other certified GKS will run on PLOT 10 GKS. This is one of the main objectives of adopting an industry standard: portable applications programs.

Computer Graphics in the Undergraduate Mathematics Curriculum

by Dr. B. Michael Castellano
Department of Mathematics
University of Alabama
in Huntsville

At the University of Alabama in Huntsville Tektronix PLOT 10 Interactive Graphics Library (IGL) forms the basis for a useful and interesting course for undergraduate science majors. By their junior year, science majors usually have had a year of calculus, some exposure to matrix arithmetic and linear algebra, and a programming course such as FORTRAN. With the combination of these skills, the students can understand, use, and design 2D and 3D graphics software. Specifically, they have the necessary prerequisites to mathematically model objects and to rotate, scale, translate, and project the models.

Such a course was recently taught to a group primarily composed of senior mathematics and computer science majors. The students learned how to use the IGL routines WINDOW, VWPORT, MOVE, and DRAW, and how to use the Tektronix graphics terminals. After plotting simple objects such as the graph of a function, the students were ready for 2D modeling transformations. The initial approach used vector addition for translating the world coordinate system, and 2×2 matrix multiplication to scale and rotate the world coordinate system. The students applied trigonometry, matrix arithmetic, dot product, and other functions from their earlier mathematical education. Then translation (vector addition) was unified with scaling and rotation (2×2 matrix multiplication) by using 3×3 matrix multiplication and homogeneous coordinates from projective geometry.

After the mathematical machinery was set up, the students designed their own software which was used to scale, translate, and rotate the models from our earlier programs. Once they were comfortable with this software, they were given access to the IGL Primary Command Set.

Next, transformations were combined, that is, a modeling/viewing transformation and matrix pre/post multiplication. In retrospect, this was the most fundamental concept of the course. A typical student exercise replaced the following calls from IGL with one call of MTRAN, and determined the window coordinates of the points (0,0), (0,1), and (1,0):

```
            :
      CALL ROTATE( 30.0 , 45.0)
      CALL TRANSL( 10.0 , 20.0)
      CALL VIEWT
      CALL SCALE(2.0 , 3.0)
            :
```

To finish the discussion of 2D tranformations, the complex number system was used to perform rotations and translations. This was much simpler for the students than the 3×3 matrix approach both conceptually and from the programming point of view. For example, the following routine illustrates how complex multiplication can be used to rotate the point XIN, BETA degrees about (0,0) to the point XOUT:

```
      SUBROUTINE CROTAT(XIN,BETA,XOUT)
      REAL XIN(2), XOUT(2)
      COMPLEX A,B
      A = CMPLX(XIN(1),XIN(2))
      B = CMPLX(COSD(BETA),SIND(BETA))
      XOUT(1) = REAL(A*B)
      XOUT(2) = AIMAG(A*B)
      RETURN
      END
```

For many of the students, this was their first concrete application of the complex number system.

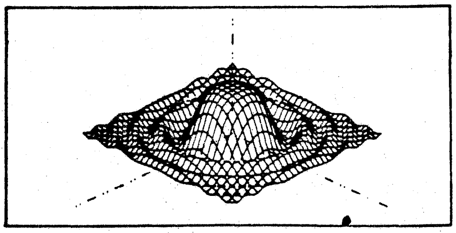
The remainder of the course was spent discussing the mathematics behind rotation of the 3D world coordinate system and projection of the view volume into the view plane. Left and right hand orientations, rotations about the x, y, and z axes, rotations about an arbitrary axis, and perspective projection were discussed. The rotations were primarily modeled with 3×3 matrices. However, as with 2D rotations, occasionally complex arithmetic was used since it simplified the code both conceptually and mechanically. While considering rotations about an arbitrary axis and perspective projection, the students found that spherical geometry was indispensable, and for the first time applied their knowledge of the spherical coordinate system to a non-calculus type problem.

The mathematical portion of the course was ended after introducing the quaternionic number system. The quaternions, sometimes called the Hamilton numbers after their discoverer, are a 4-dimensional extension of the complex numbers, and as such they contain both the real and complex number systems. Unit quaternions simplified rotations of the 3D world coordinate system about an arbitrary axis in much the same way that unit complex numbers simplified rotations of the 2D world coordinate system. In this exercise the students reviewed and applied many ideas from linear algebra and vector calculus such as dot products, cross products, triple product rules, the matrix of a transformation, coordinates, determinants, orthogonal bases, etc.

The following code illustrates the use of quaternionic multiplication to rotate a vector in 3D world space about an arbitrary axis:

```
      SUBROUTINE QROTAT(ANG,AXS,PI,PO)
C-------------------------------------------------------------C
C     INPUT:   ANG, AXS(3), PI(3)                              C
C     OUTPUT:  PO(3)                                           C
C                                                              C
C     THIS ROUTINE USES QUATERNIONIC MULTIPLICATION TO         C
C     ROTATE, USING THE RIGHT HAND RULE, THE VECTOR PI, ANG    C
C     DEGREES ABOUT THE AXIS DETERMINED BY AXS.  THE ROTATED   C
C     VECTOR IS RETURNED AS PO.                                C
C                                                              C
C     THE ARRAY Q(0:3) IS USE TO REPRESENT THE QUATERNION      C
C     Q(0) + Q(1)I + Q(2)J + Q(3)K.                            C
C-------------------------------------------------------------C
      REAL ANG, AXS(3), PI(3), PO(3)
      REAL ANORM, QL(0:3); QR(0:3), QT(0:3), QI(0:3), QO(0:3)
C-------------------------------------------------------------C
C     ANORM IS THE LENGHT OF THE VECTOR AXS,  QR IS THE        C
C     QUATERNIONIC CONJUCATE OF QL, AND ROUGHLY SPEAKING,      C
C     QL = COS(ANG/2) + SIN(ANG/2)*(AXS/ANORM).                C
C-------------------------------------------------------------C
      ANORM = SQRT(AXS(1)**2 + AXS(2)**2 + AXS(3)**2)
      C = COSD(ANG/2.0)
      S = SIND(ANG/2.0)
      QL(0) = C
      QR(0) = C
      DO 100, I = 1,3
      QL(I) = S*AXS(I)/ANORM
      QR(I) = -QL(I)
100   CONTINUE
C-------------------------------------------------------------C
C     QI(0:3) IS THE QUATERNIONIC REPRESENTATION OF PI         C
C-------------------------------------------------------------C
      QI(0) = 0.0
      QI(1) = PI(1)
      QI(2) = PI(2)
      QI(3) = PI(3)
C-------------------------------------------------------------C
C     LET # DENOTE QUATERNIONIC MULTIPLICATION. PO SATISFIES   C
C     THE FOLLOWING EQUATION:                                  C
C     0.0 + PO(1)I + PO(2)J + PO(3)K = QL # QI # QR.           C
C-------------------------------------------------------------C
      CALL QPROD(QL,QI,QT)
      CALL QPROD(QT,QR,QO)
      PO(1) = QO(1)
      PO(2) = QO(2)
      PO(3) = QO(3)
      RETURN
      END

      SUBROUTINE QPROD(Q1,Q2,Q)
C-------------------------------------------------------------C
C     INPUT:  Q1(0:3), Q2(0:3)                                 C
C     OUTPUT: Q(0:3)                                           C
C                                                              C
C     Q = Q1 # Q2                                              C
C-------------------------------------------------------------C
REAL Q1(0:3), Q2(0:3), Q(0:3)
Q(0) = Q1(0)*Q2(0) - Q1(1)*Q2(1) - Q1(2)*Q2(2) - Q1(3)*Q2(3)
Q(1) = Q1(0)*Q2(1) + Q1(1)*Q2(0) + Q1(2)*Q2(3) - Q1(3)*Q2(2)
Q(2) = Q1(0)*Q2(2) - Q1(1)*Q2(3) + Q1(2)*Q2(0) + Q1(3)*Q2(1)
Q(3) = Q1(0)*Q2(3) + Q1(1)*Q2(2) - Q1(2)*Q2(1) + Q1(3)*Q2(0)
RETURN
END
```

To illustrate the value of 3D graphics as an educational tool, a program was developed that plotted the graph of the function $z = \sin(x^2 + y^2)/(x^2 + y^2)$. (See figure 1). A program was considered to graph any function of the form $z = f(x,y)$. This project was relevent to students in the class who were taking a course in advanced calculus of several variables. The program gave them the opportunity to graph some of the functions that they were studying.



Chrysler Makes Strides in CAD/CAM Tektronix 4110A Terminals, Local Programmability a Part

Computer-aided design is not new at Chrysler Corporation. Back in 1965, before graphics terminals, and before any really good CAD timesharing or multiprogramming systems were available, batch-oriented CAD was developed in-house. When Tektronix introduced the direct view storage tube, Chrysler immediately took advantage of graphics technology and redesigned their CAD system. They continue to be at the forefront of industrial usage of computers. From initial idea to final bill of lading, Chrysler is strengthening and streamlining its computer-aided design, computer-aided engineering, and computer-aided manufacturing functions.

Walt Weglarz, manager of Computer Systems and Operations Engineering at Chrysler, is right in the middle of this effort. Weglarz says the key is shared data, which rests on a robust network system allowing users to easily communicate. Adapting to hardware limitations, and integrating network accessors and network devices, Chrysler created a viable network which their users may tap into from half a mile away, or from Mexico, Huntsville, other remote sites, or right at the computer center.

Weglarz stated that the "clay-to-die" computer-aided design system is basically in place and functioning. Chrysler is now focus-

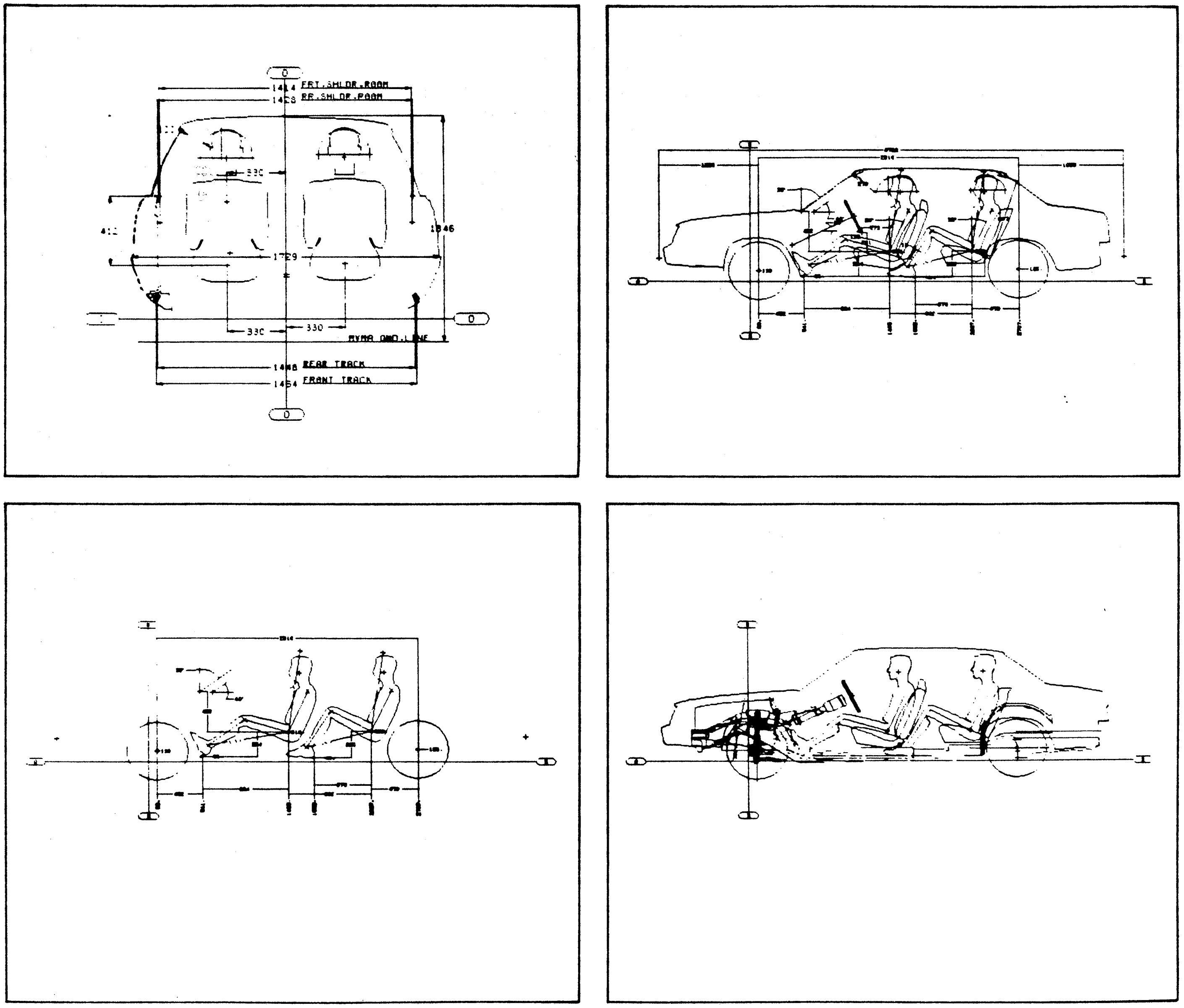


Figure 1. Chrysler engineers create, edit and manipulate models on Tektronix graphics terminals.

ing on further integration, a higher degree of use in engineering-product development, and a tremendous expansion in its manufacturing arm.

In 1982 Chrysler doubled its host computer capacity, interconnecting nine mainframes in the shared CAD/CAM system. Almost 300 graphics terminals (some of which were Tektronix 4114 intelligent terminals) were integrated into the system with 186 in the Engineering office alone dedicated to CAD. Additional computer capacity will be added into the network this year and over 100 additional CAD stations are planned for 1983.

While information sharing eliminates duplication of effort, misunderstandings and misinterpretations, Weglarz emphasizes that subsequent productivity gains are a direct function of the number of people using it. Approximately 1500 users at Chrysler are now working with the CAD/CAM system, 800 of whom are designers/draftsmen. An additional 400 are being trained annually. Eventually all designers and draftsmen will have graphics workstations enabling total computer-aided design of a vehicle.

The ultimate plan sees design information and master data bases fully integrated. Final designs will be converted directly into numerical controls for robots, milling machines, stamping machines and other manufacturing tools. Workstations in the factory will streamline parts programming. Management links to all locations will facilitate supporting operations.

Local Programmability Accelerates Response

Recent components of the CAD system are Tektronix 4110A Series intelligent graphics terminals with 4110A Series Local Programmability. Bob Antworth, supervisor of Chrysler's Distributed Graphics Development, comments that being able to move code to the intelligence in the terminals promises a tremendous return on investment.

Chrysler's experience has shown that one out of four commands invoked by the engineer at the workstation calls for redraw, zoom or pan, and about 50% of host disk activity is related to these functions. Allowing the 4110A intelligent terminals to process these functions locally relieves the host computers and accelerates the terminals' response to the engineers. The latter is critical. As the Chrysler engineers have matured in their use of CAD, they expect more from the system, especially a fast response.

The engineers also expect 100% reliability. Interruption and noise are inherently part of any network system, particularly in long communication lines from remote sites. Run-

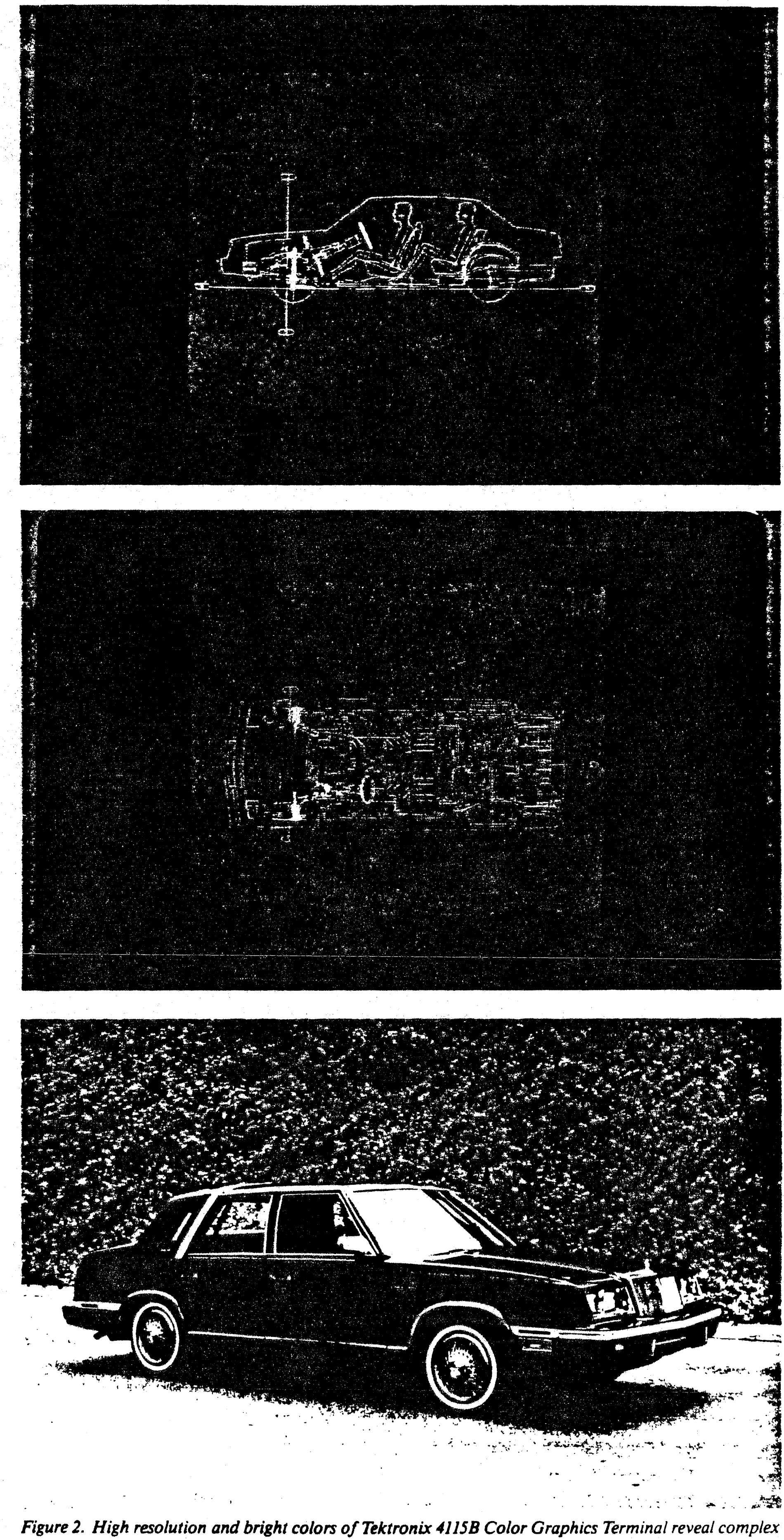


Figure 2. High resolution and bright colors of Tektronix 4115B Color Graphics Terminal reveal complex design of Chrysler automobile.

ning as much of the design work under 4110A Local Programmiability as feasible, Chrysler hopes to eliminate these inconveniences and strengthen CAD as a tool.

A Working Example

The first application program to be transported to 4110A Local Programmability was GCS (Graphics Compatibility System), Chrysler's 100K byte series of FORTRAN subroutines used to manipulate and display graphic items or displays.

Bob reflects, "It took about two months to convert, which exceeded our expectations."

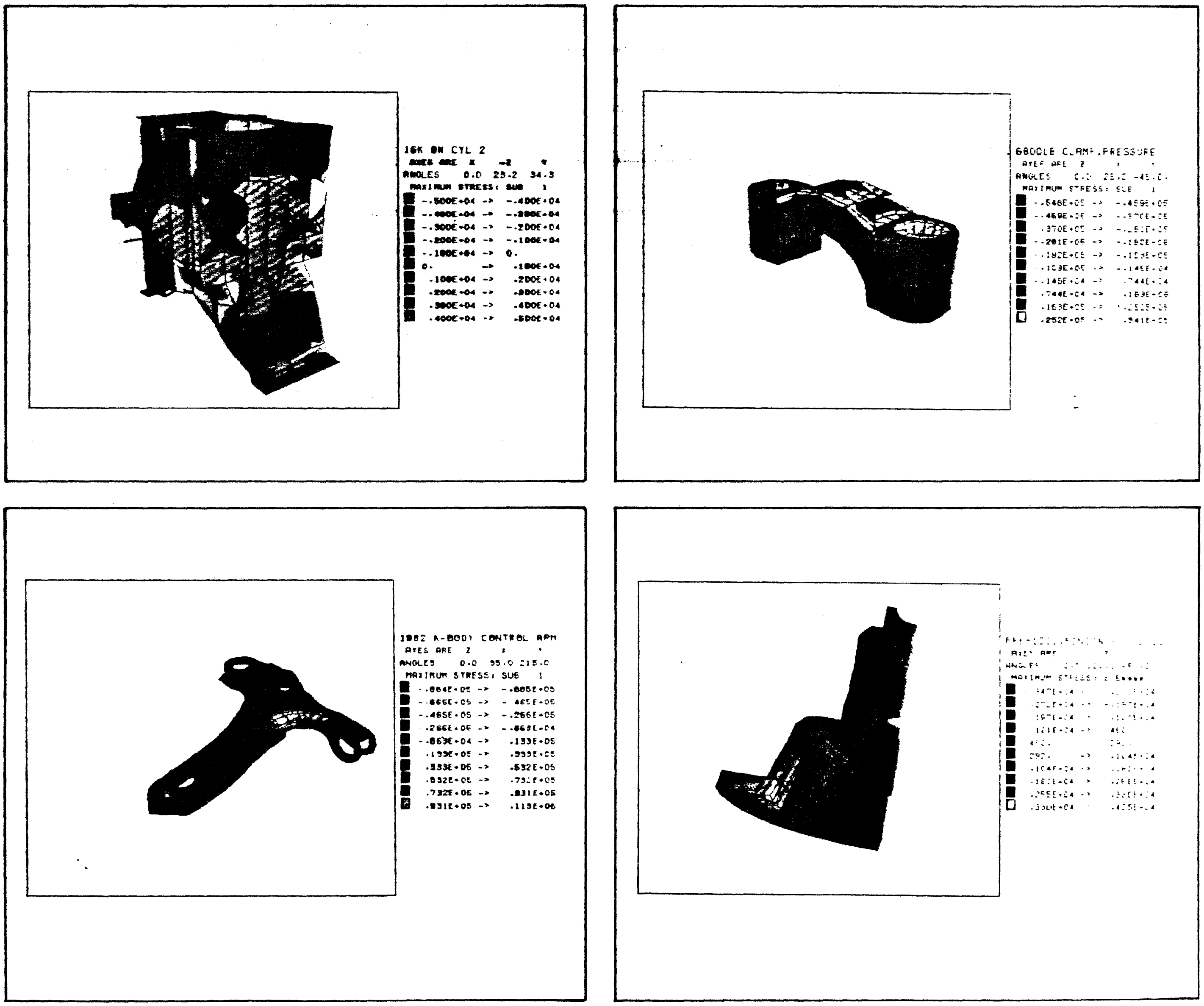
GCS provides the software tools for programmers to develop three-dimensional graphics applications. The FORTRAN-callable subroutines support device independence of graphics terminals.

QUICKCON (Quick Concept) is a major Chrysler application used extensively by a large number of engineers and must provide a good, consistent response. It is the second candidate for Local Programmability, notes Bob, and they are in the process of converting it. (See accompanying sidebar.)

Built upon GCS, QUICKCON is a computer-aided design tool for three-dimensional data. The engineer can perform design and drafting tasks on a graphics terminal in a manner similar to that used on a drafting board except QUICKCON offers greater speed, accuracy and flexibility.

Antworth remarks that the high resolution of the 4114, its local refresh, and its Local Programmability can also handle the very critical manufacturing function of pattern nesting. Nesting involves laying out a quantity of workpieces on a large piece of material (sheet metal, fabric or other) to achieve the least amount of waste when cut. In this application, the operator digitizes and stores patterns, retrieves them for a nest, and generates the tape with the basic data definitions and communications for manufacturing. Using Local Programmability, engineering can set up each nest, interacting quickly with the program.

Other applications are naturals for Local Programmability. Antworth stresses, "The more remote a plant is, the more Local Programmability becomes a nice feature to have. Our more graphic NC programs particularly lend themselves to Local Programmability, for instance, the Chrysler Interactive Parts Programming System. Using a tool library, the operator at the factory can sit at the terminal, describe the machining operations graphically, and see the effect, check-



```
16K ON CYL 2
 AXES ARE  X     -Z     Y
ANGLES     0.0  23.2  34.3
 MAXIMUM STRESS: SUB    1
■  -.500E+04  ->  -.400E+04
■  -.400E+04  ->  -.300E+04
■  -.300E+04  ->  -.200E+04
■  -.200E+04  ->  -.100E+04
■  -.100E+04  ->  0.
■  0.         ->   .100E+04
■   .100E+04  ->   .200E+04
■   .200E+04  ->   .300E+04
■   .300E+04  ->   .400E+04
■   .400E+04  ->   .500E+04
```

```
68DOLE CLAMP.PRESSURE
 AXES ARE  Z      Y      Y
ANGLES     0.0  23.0 -45.0
 MAXIMUM STRESS: SUB    1
■  -.546E+05  ->  -.459E+05
■  -.459E+05  ->  -.370E+05
■   .370E+05  ->  -.281E+05
■  -.281E+05  ->  -.192E+05
■  -.192E+05  ->  -.103E+05
■   .103E+05  ->  -.145E+04
■  -.145E+04  ->   .744E+04
■   .744E+04  ->   .163E+05
■   .163E+05  ->   .252E+05
□   .252E+05  ->   .341E+05
```

```
1982 K-BODY CONTROL ARM
 AXES ARE  Z      X      Y
ANGLES     0.0  95.0 215.0
 MAXIMUM STRESS: SUB    1
■  -.864E+05  ->  -.665E+05
■  -.665E+05  ->  -.465E+05
■  -.465E+05  ->  -.266E+05
■   .266E+05  ->  -.663E+04
■  -.663E+04  ->   .133E+05
■   .133E+05  ->   .333E+05
■   .333E+06  ->   .532E+05
■   .532E+05  ->   .732E+05
■   .732E+06  ->   .831E+05
■   .831E+05  ->   .113E+06
```

```
[illegible]
 AXES ARE [illegible]
ANGLES [illegible]
 MAXIMUM STRESS: [illegible]
■   [illegible]E+04  ->  [illegible]
■   [illegible]E+04  ->  [illegible]
■   [illegible]E+04  ->  [illegible]
■   .121E+04  ->   [illegible]
■   [illegible]       [illegible]
■   [illegible]   ->   .104E+04
■   .104E+04  ->   [illegible]
■   .181E+04  ->   .258E+04
■   .258E+04  ->   .330E+04
□   .330E+04  ->   .425E+04
```

Figure 3. NASTRAN analysis portrayed graphically.

ing for interference. The operator can access the common data base and have it downloaded in a burst mode of communication. Continuous interaction over the communication lines is not required, which can be slow and expensive."

Although Chrysler engineers have come to expect the quality of resolution found in the Tektronix direct view storage tubes, color has made an impact. For the past four years Chrysler has employed Tektronix color graphics terminals for structural analysis, stress analysis and solid modeling. "Color graphics have been very beneficial in management and customer presentations, which would have been tremendously difficult just with line drawings," Weglarz points out. Color plays an important part in determining the serviceability of products.

Quality Results

Chrysler's CAD/CAM system has won enthusiastic acceptance by its engineers and management. The software capability and hardware performance have significantly strengthened the design process. Computer simulation permits engineers to try a multitude of different design approaches, pinpointing and eliminating problems, selecting the best fit.

Behind the shiny paint of its new cars, Chrysler is revolutionizing its design and manufacturing processes. Tektronix interactive graphics terminals are playing a major part with Local Programmability in a strong supporting role. The result is quality at an affordable cost.

Editor's Note: Walt Weglarz and Bob Antworth in a recent visit to Tektronix presented a graphic tour of Chrysler's CAD/CAM system. We appreciate Chrysler's willingness to share their work with *TEKniques* readers. We also thank Mark Rahrig, Tektronix Sales Engineer in Detroit, for donating his time and assistance to bring this application to readers.

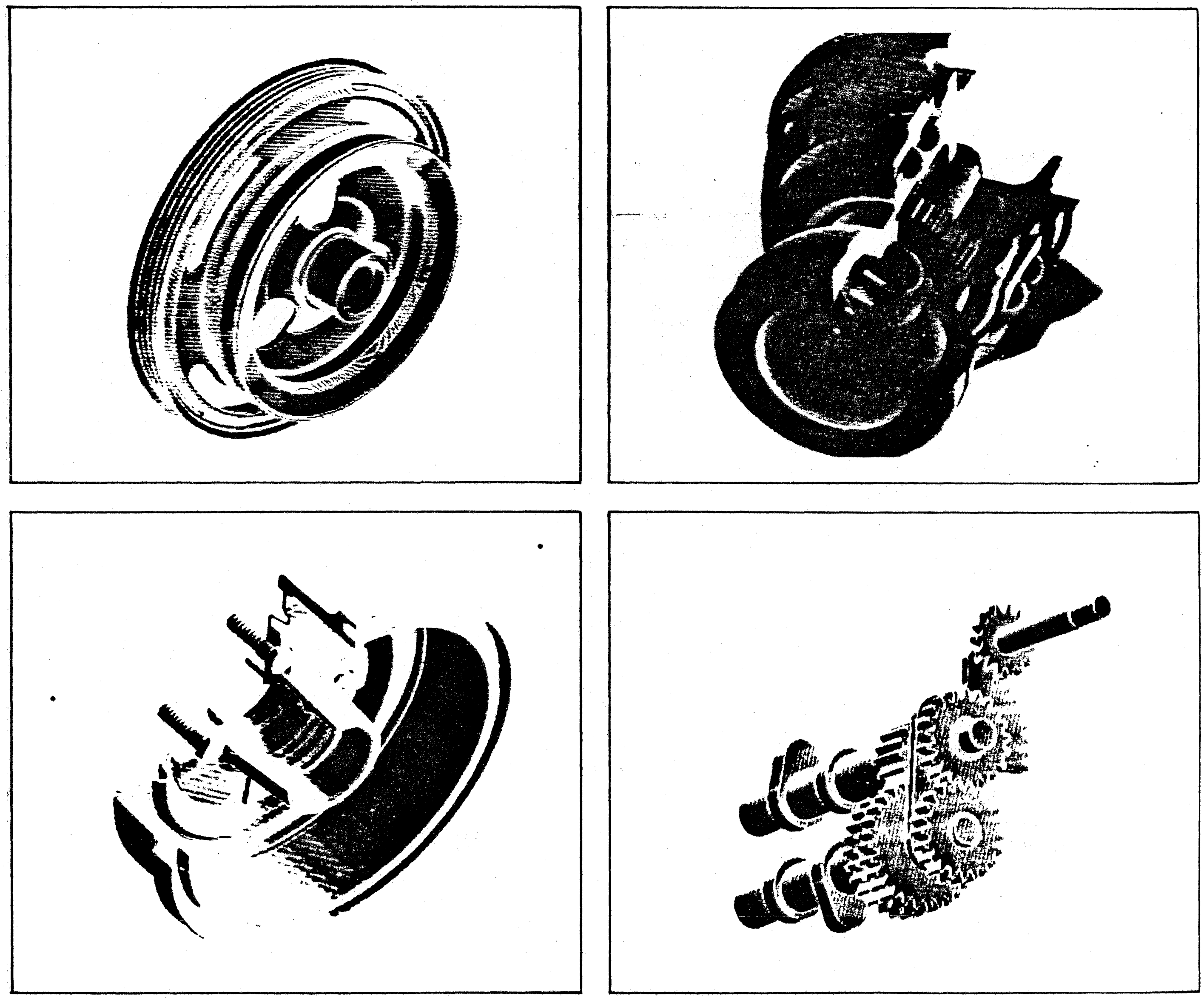


Figure 4. Solids modeling at Chrysler Corporation lends itself to Local Programmability.

Integrating 4110A Local Programmability at Chrysler Corporation

by John Buslawski
Tektronix, Inc.
Detroit, MI

By incorporating Tektronix 4110A Local Programmability into their CAD/CAM system, Chrysler Corporation is stream lining user interaction and speeding response time. Although distributing the CAD/CAM workload is far from complete, the issues involved in Chrysler's host-to-Local Programmability conversion might aid others.

Feasibility

A major consideration was whether Local Programmability could satisfactorily handle the large amounts of code required for Chrysler's application. To gain an idea of the feasibility of the project, Chrysler's first step was to move the three-dimensional display functions from its CAD package (QUICKCON) to Local Programmability.

The display sections of QUICKCON are based on GCS, a FORTRAN library of two-dimensional display routines such as MOVE, DRAW, WINDOW, SCALE, ROTATE and so forth. Prior to installing GCS on Local Programmability, axes generation and other high level routines which performed pure graphing functions and thus had no bearing on 3-D display were extracted from GCS. At the very lowest level, the device driver routines in GCS were replaced with 4110A Low Level Terminal Interface calls, which provide a direct interface to the 4110A Graphics Terminal's firmware.

The entire transported code fit into less than 200K bytes of 4110A memory. Display speed, based on worst case (most complex drawing), was entirely satisfactory. Chrysler had determined that converting portions of its CAD package to 4110A Local Programmability was indeed feasible.

FORTRAN/Modularity Aided Conversion

Although Chrysler's CAD/CAM package has evolved over time, spanning 15 calendar years which encompassed 200 man-years and a variety of programmers, its modularity made it a good candidate for distributed processing. The relatively speedy integration of 4110A Local Programmability also resulted from using much of Chrysler's existing debugged and working code. After translating the FORTRAN '66 QUICKCON routines to FORTRAN 77, the code was installed in Local Programmability where it worked without major revision.

The result to date is approximately eight FORTRAN source files which are compiled and built into four library files. These libraries are linked into a mainline along with the FORTRAN-86 run-time libraries. A real time saver in the compilation process has been the Tektronix-provided BATCH utility running under CP/M-86.

Data Base Restructure

Converting the host data base to fit within 4110A Local Programmability's capabilities also proved feasible. Due to the amount of data, however, the data base resides on the winchester (hard) disk.

The major effort was solving the data base format. QUICKCON running on Chrysler's Cyber host computers used just about every host-specific method for writing to disk files. While making for efficient input/output, it was heavily device-dependent. Architecture, capacities, format and a number of other differences between the Cyber's disk and 4110A Local Programmability required a restructuring of the data base for local storage. It was readily accomplished with all data types occupying direct access files in unformatted records.

Second Phase

Because of the success with the display routines, Chrysler is entering a second phase of conversion to Local Programmability. The entire command parser of QUICKCON will be brought down to Local Programmability. Designers communicate with QUICKCON through a command language. Verifying the commands at the 4110A graphics terminals would eliminate no-op communications traffic and help overcome some of the response time lag. For example, now when a command is incorrectly typed in, it consumes host processing time and ties up the communications link only to elicit an error message from the host. Making sure the command and its parameters are valid before being passed to the host will eliminate this costly and time consuming procedure.

GIN "pick" functions are also being moved to Local Programmability, which will eliminate a considerable amount of conversation between host and user. Currently, the engineer, using the crosshair cursors, selects a line to move, for example. Communication with the host ensues. The host checks to see if the pick is indeed a line, then responds to the engineer identifying the line it thinks he picked. If correct, the line is moved; if not the procedure repeats. Performing the checking and verifying functions at the terminal will save dialog with the host and, consequently, communications line traffic. When the host receives the command to move a line, it can perform the function knowing that it is valid.

4110A Local Programmability at Chrysler is a constantly evolving item, with more and more capabilities being added. *TEKniques* readers will be kept informed of the progress.

A note of appreciation to Frank Rizzo, Lead Development Engineer on this project, for sharing his work with *TEKniques*..

"B" Series Enhancement Kits for Tektronix 4110 Family of Graphics Terminals

by Ken Knepper
Tektronix, Inc.
Wilsonville, OR

As new technology is incorporated into Tektronix products, a means for customers to equip their existing systems with the advancements is provided. For instance, the 4110 Series of Graphics Terminals introduced in 1982 included the 4112, 4113 and 4114 Terminals. Later "A" Series enhancements were added and upgrade kits made available to equip existing terminals with the enhancements. New products in the family began life as "A" models such as the 4116A Graphics Terminal (see *TEKniques* Vol. 7, No. 1).

The 4110 family of Graphics Terminals has now evolved to the "B" Series as evidenced in the recently introduced 4115B Graphics Terminal (described in *TEKniques* Vol. 7, No. 2). "B" enhancements include:

- Standard memory of 288K
- Firmware version 6
- ANSI X3.64 (VT-100 type commands)
- Winchester disk support
- CP/M data file format

Supporting these technological changes, Tektronix has provided upgrade kits for existing terminals. Users with "A" Series terminals should order one of the following kits, depending on their terminal model:

4112F61	4114F61
4113F61	4116F61

Upgrade kits are also available which incorporate both "A" Series as well as "B" Series enhancements. These kits are:

4112F62	4113F62	4114F62

Contact your Tektronix Sales Engineer for additional details or for ordering.

*Editor's Note

Continue Your Subscription

We're updating the distribution list for *TEKniques.*

After January 1, 1984, only those who have returned the form indicating their interest will continue to receive *TEKniques* (this includes Tektronix employees). You'll find the form in this issue and the past two issues. Take a minute to send one in if you haven't done so.

TEKniques Published Quarterly

Several readers questioned the dates on the past two issues of *TEKniques.* In early 1983 *TEKniques* was to revert to a bimonthly publication as reflected by the date on Vol. 7, No. 1 of January/February. For various reasons it remained a quarterly newsletter as evidenced by the date on Vol. 7, No. 2 - Summer, and will continue as a quarterly publication.

Back Issues

TEKniques Vol. 1 through 3 are no longer in print. Back issues of Vol. 4 through the current issue are available by calling your local Tektronix office.

Program Exchange

Readers who wish to contribute to the Program Exchange for Tektronix Graphics Systems may acquire documentation instructions and forms by writing to:

Tektronix, Inc.
IDD Program Exchange
Mail Stop 63-575
P.O. Box 1000
Wilsonville, OR 97070

For each program accepted, you will receive the package of your choice from the Program Exchange.

Keep 4050 Series Catalog

The 4050 Series Applications Library catalog published in 1981 should be retained. New pages will be added periodically, but the original portion will not be reproduced. Present holders of the catalog should clip the New Abstracts pages for *TEKniques* and insert them in the back of the catalog. Those who do not have a copy of the catalog may order it through Tektronix Central Parts Ordering (see New Abstracts section for information) by part number 062-6343-00, or may call the local Tektronix Field Office.

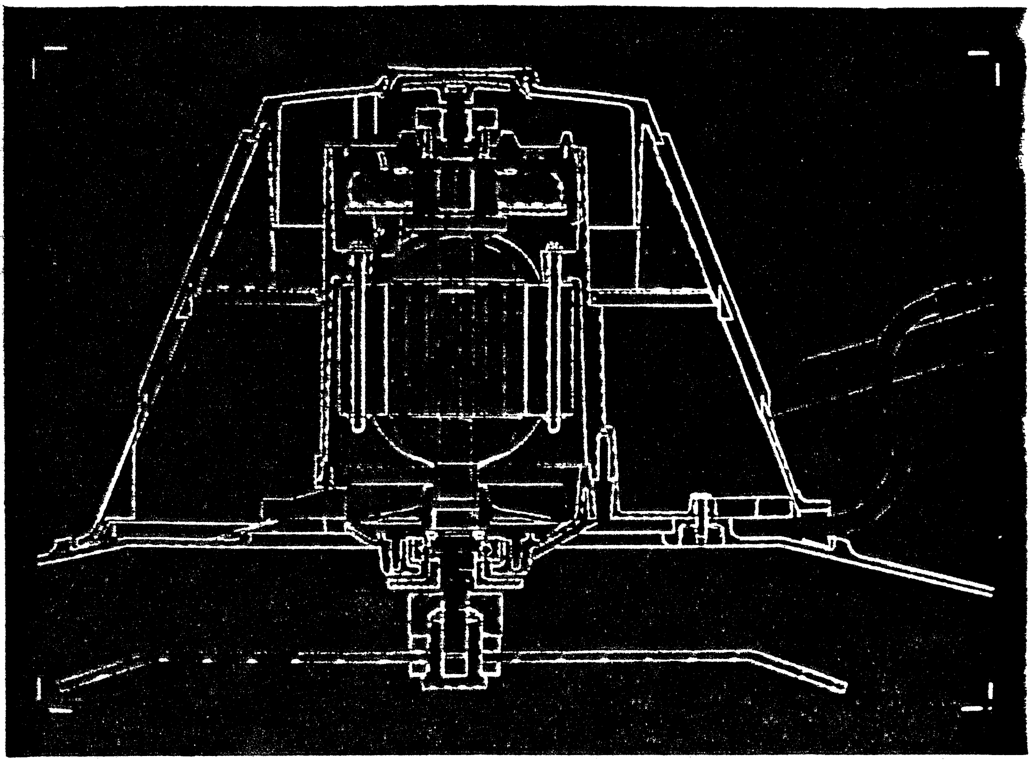


Figure 1. Autoconvergence in the 4115 maintains the sharp, bright graphics displayed on the screen.

Autoconvergence Enhances High-Resolution of 4115B Color Display

by Dan Denham, Marty Singer and Bill Meyer
Tektronix, Inc.
Wilsonville, OR

The purpose of a high-resolution color display is to provide sharp, true colors and accurate data representation for such high-density graphics applications as CAD/CAM and mapping. Color benefits these applications by serving to organize, locate, and highlight graphics information. An engineer designing integrated circuits, for example, can use color coding to identify the circuit's different components, layers, and connecting wires.

One major problem, however, common to all color displays, has caused many computer graphics users who require high resolution to choose monochrome displays. That problem is misconvergence, a condition in which the red, green, and blue beams of a color raster-scan display do not intersect precisely at a specified point on the screen. The effects of misconvergence include poor color, a fuzzy picture, visual fatigue, and even a loss of graphics information.

Though misconvergence occurs on all color displays, even color television sets, its effects are especially critical on a high-resolution color display. Users make a big investment in buying high-resolution color displays and generally make that investment because their applications require the display of high-density pictures in which the operator must be able to discriminate details.

In a high-density graphics application, if the beams are not properly converged, two parallel red and green lines with only a narrow gap between them may appear as one thick yellow line. Misconvergence could cause a sharp thin white line on the display (Figure 2a) to appear as a fuzzy thick pink line with a red fringe (Figure 2b), or, in the worst case, as three separate lines of red, green, and blue (Figure 2c).

A high-resolution color display, of course, loses much of its usefulness when the beams are misconverged to the point of turning thin

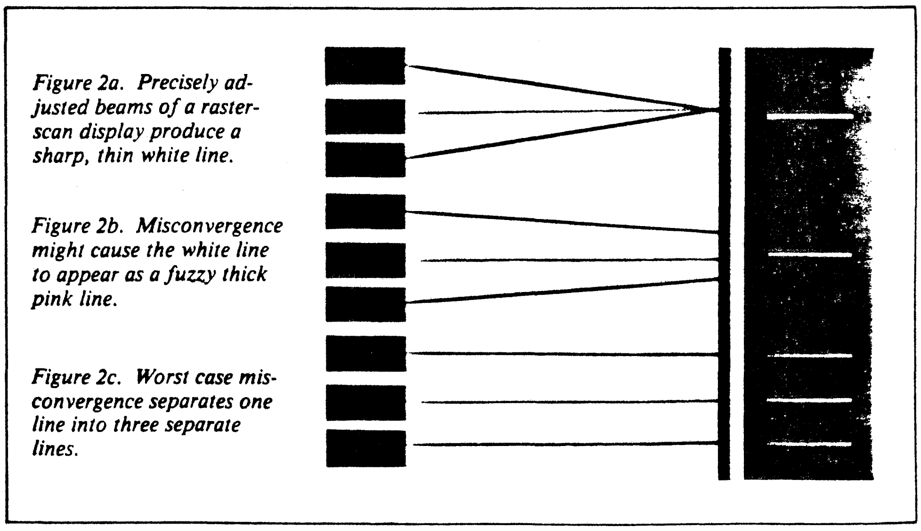


Figure 2a. Precisely adjusted beams of a raster-scan display produce a sharp, thin white line.

Figure 2b. Misconvergence might cause the white line to appear as a fuzzy thick pink line.

Figure 2c. Worst case misconvergence separates one line into three separate lines.

Figure 2. Illustrating misconvergence.

white lines into thick pink lines. The misconvergence could lead to a serious error.

The display's color convergence can usually be adjusted, but it is a time-consuming procedure that often requires a trained technician. In addition, the factors of tube age, temperature, and position, as well as the resolution demands of the application and visual acuity of the user all contribute to the frequency of convergence adjustment. Turning a display to the side at only a slight angle, perhaps to better view the screen, can cause misconvergence, even if the beams were adjusted only minutes before, because of the influence from the earth's magnetic field.

The Tektronix 4115B Computer Display Terminal solves the problem of misconvergence in high-resolution color displays through a system called "autoconvergence." Autoconvergence allows the 4115B to exceed the convergence specifications of all other 19-inch color displays to a point where even very discriminating users cannot visually detect a trace of misconvergence.

When misconvergence does occur, the 4115B, in a few seconds, automatically restores convergence to the original specifications at the press of a user accessible button. Autoconvergence is not merely an enhancement, but a significant and integral part of the high-resolution color display. In fact, as the resolution of a display increases, so does the need for some form of convergence system.

Locating Misconvergence

When giving convergence specifications, manufacturers often divide the screen into two or three sections (Figure 3). The center circular section of the screen, often called the "quality area" generally boasts better convergence specifications than the corners of the screen. It is difficult to maintain tight convergence at the corners because of the wide deflection angle of the beams at these points, a problem especially acute on a 19-inch CRT. A display, then, may claim convergence specifications of 0.5 mm in the "quality area" and 0.7 mm at the corners, or perhaps by shrinking the "quality area," claim specifications of 0.3 mm at the center (region A in Figure 3), 0.6 mm in an area surrounding the center (region B), and 0.8 mm at the corners (region C).

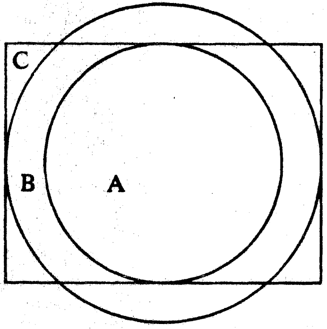


Figure 3. Manufacturers often divide the screen into sections to support tight convergence specifications in the "quality" area. The 4115B does not restrict its convergence specification of .2 mm or less to a circular area in the center of the screen.

The 4115B does not restrict that convergence to a circular "quality area" in the center of the screen. A critical detail in a high-density picture can be displayed as accurately at any corner of the screen as at the center. Convergence over the full viewing area especially benefits the display of alphanumeric text which often begins in the upper left corner of the screen. On the 4115B, very fine text can be displayed with a legibility and color consistency of unvarying high quality across the entire screen. On a display that limits its best convergence to a circular "quality area," the text may appear fuzzy in the corners and sharper in the center of the screen.

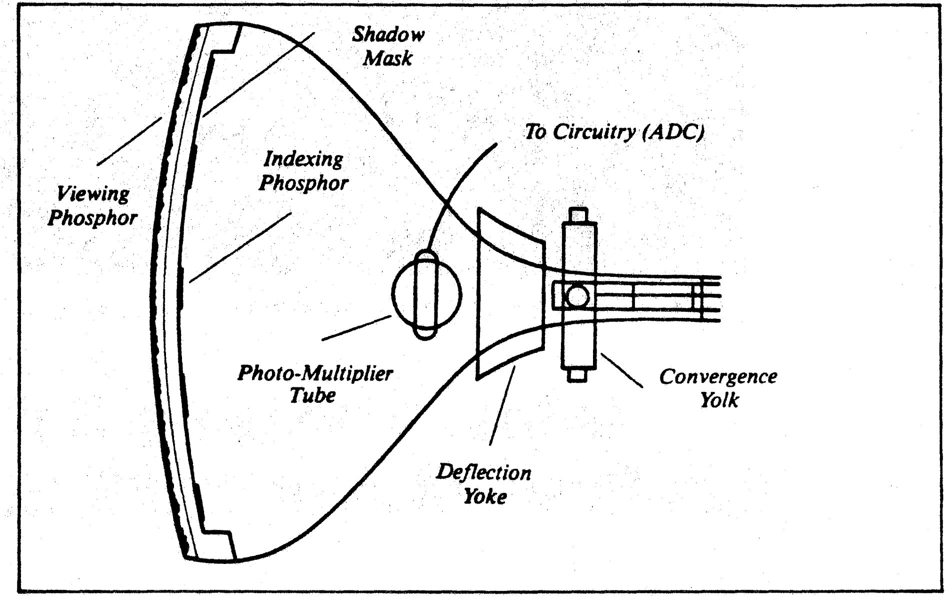


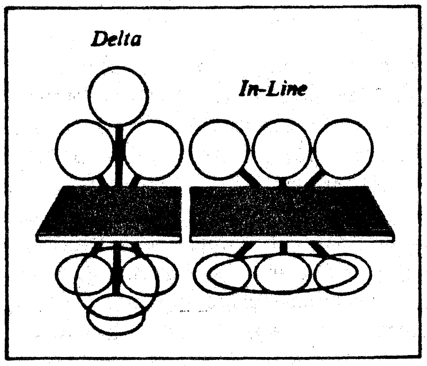
Figure 5. The physical aspect of autoconvergence.

CRT Selection and Convergence

To provide as small a screen spot size as present raster-scan technology allows, the color beams of the 4115B originate from a delta gun (with two beams at the base of the gun and one on top, in the form of the Greek letter delta).

In conjunction with the delta gun, the 4115B uses a shadow mask, a metal plate positioned just behind the screen. The shadow mask has many small holes through which the electron beams are funneled so that they illuminate the correct phosphor dots on the screen. The "red" beam should hit the red phosphor dots, the "green" beam should hit the green dots, and so on. For delta gun configurations, the phosphor dots are arranged on the screen in triads, each triad consisting of a red dot, a green dot, and a blue dot.

A delta gun is more difficult to converge than the more commonly used in-line gun (in which all three beams are arranged in a horizontal line), but in-line gun CRTs sacrifice spot size for ease of convergence. Tektronix's autoconvergence system, however, achieves convergence on a delta-gun CRT better than that obtained by any in-line CRT-yoke combination available, allowing 4115B designers to opt for the smaller spot size.



How Autoconvergence Works

Traditionally, convergence was adjusted manually by a trained technician. By rotating a number of potentiometers and viewing the results on the screen, the technician adjusted the convergence over selected areas of the display, a procedure that could take an hour or more.

Though many displays still use the adjustment technique of manual (or analog) convergence, a few recently developed displays offer digital convergence. To adjust convergence digitally, the operator interrupts the application to put the display into a convergence mode. In the convergence mode, the display restores approximate convergence and draws a grid on the screen. By pressing keys on a keypad connected to the display and viewing the screen, the operator fine-tunes the approximate convergence until it reaches an acceptable level for the application. This typically takes several minutes for the operator to complete.

With the 4115B's autoconvergence system, convergence can be automatically adjusted to the original specification at any time during an application. If convergence drifts, the operator need only press a button on the display and wait a few seconds to correct the condition.

Figure 4 shows block diagrams for the analog, digital, and autoconvergence adjustment systems. In analog convergence adjustment (Figure 4a), setting the potentiometers generates a complex waveform which, when amplified, drives the convergence coils. The coils' electromagnetic field independently deflects each electron beam to maintain convergence.

In digital convergence (Figure 4b), the display is first converged at the factory. The resulting waveform is then stored in EEROM. During the convergence procedure, the values in EEROM are fed through a digital-to-analog converter (DAC) to generate the original complex waveform. The convergence is only approximate because it represents the condition of the new display in the factory. The operator must further adjust the convergence to account for the current condition of the display, which has been affected by such factors as temperature, age, and position.

The 4115B's autoconvergence system (Figure 4c) removes the user from the convergence adjustment task, not only saving adjustment time, but enabling more precise convergence. Basically, autoconvergence uses optical feedback to sense the location of each electron beam. As in digital convergence, a complex waveform is stored in EEROM in the form of digital values. By modifying these values, which are then fed through a DAC and amplified, the display's microprocessor can adjust the beams so that they converge within the specified range of less than or equal to 0.2 mm at any viewable point on the screen.

Figure 5 shows a side view of the autoconvergence CRT. As the electron beams scan the screen, phosphor patterns (indexing phosphor) on the back side of the shadow mask emit light back into the tube. Through a window in the CRT, the photodetector senses this light and outputs the information to the processor. The phosphor patterns do not block the holes of the shadow mask and have no effect on beam energy reaching the phosphor dots (viewing phosphor) on the screen. An aluminum layer between the shadow mask and the screen, while reflecting light from the phosphor dots out toward the viewer, also serves to prevent any errant screen light from reaching the photodetector.

The phosphor patterns on the shadow mask are deposited in locations corresponding to

Measuring Misconvergence

by Jerry Murch
Tektronix, Inc.
Wilsonville, OR

As with most specifications, the problem when considering convergence values is determining how to make a qualitative judgment from purely quantitative information. How much better is a convergence of 0.3 mm than 0.5 mm or 0.8 mm? While designing the 4115B, Tektronix engineers tried to answer this question by devising an experiment to find the value at which a user could no longer detect misconvergence.

In the experiment, the 4115's screen was split in half and two identical images were displayed. At first, both images were converged as well as possible within the technical limits of the display. One of the images was then misconverged to one of a series of specified values. The subjects were asked to choose the better image. When the misconvergence reached 0.4 to 0.5 mm, all the subjects preferred the converged image. Often they could not indicate why one image appeared better. The engineers repeated the experiment, this time telling the subjects that one of the images was being misconverged. The subjects were asked to choose the best converged image. When the misconvergence reached 0.2 mm, almost all the subjects could detect the difference in convergence between the two images.

The results of the study show that most people will notice a decrease in the display quality of an image if misconvergence exceeds 0.4 to 0.5 mm. People who are aware of the problem can detect any misconvergence of more than 0.2 mm. Reasoning that the users of a high-resolution color display are or will soon be aware of misconvergence, Tektronix designed the 4115B to meet a convergence specification of less than or equal to 0.2 mm across the entire viewing area of the screen. The display's autoconvergence system not only makes this possible, but also automatically maintains what is, in effect, optimum convergence. When the 4115B's convergence is set to its specification of less than or equal to 0.2 mm, most users will not be able to detect misconvergence. When misconvergence does become detectable, the user need only press a button to restore convergence to the original specification.

major convergence points on the screen. The processor adjusts the beam positions at every major point, enabling the 4115B to maintain its convergence specification across the entire viewing area. Each phosphor pattern is shaped so that the photodetector will output two pulses (Figure 6). The photodetector outputs a pulse when the beam scans first the vertical line in the pattern and again when the beam scans the diagonal line. From the time difference between the two pulses, the processor can determine the relative positions of the electron beams.

During the autoconvergence procedure, the processor directs the display to turn on one electron beam at a time. In this way, it receives a pair of pulses for each beam at each pattern location. By comparing the pairs of pulses, then individually moving each beam, the processor can adjust convergence. When the pulse pairs for all three beams match, within the range of the convergence specification, the autoconvergence procedure is complete.

Figure 7 illustrates the autoconvergence procedure. In this example, the green and red beams are turned on to produce a yellow line. The beams, however, are misconverged so that separate red and green lines appear (Figure 7a). As the first step, the processor turns off the red beam to get a photodetector output for the green beam (Figure 7b).

In Figure 7c the processor turns off the green beam, activates the red beam, and compares the new pair of pulses with the green beam's detector output. From the difference in the red and green pulses, the processor computes the required corrections and moves the beams until the pulse pairs match and the beams are converged (Figure 7d).

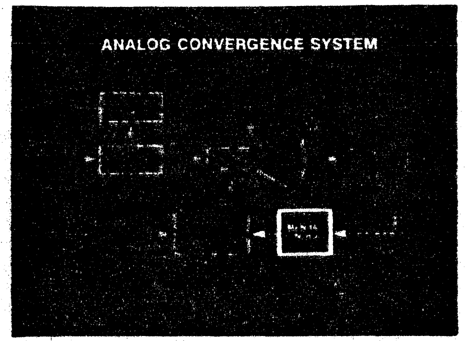


Figure 4. Three different ways to correct misconvergence.

Figure 4a. Analog convergence adjustment.

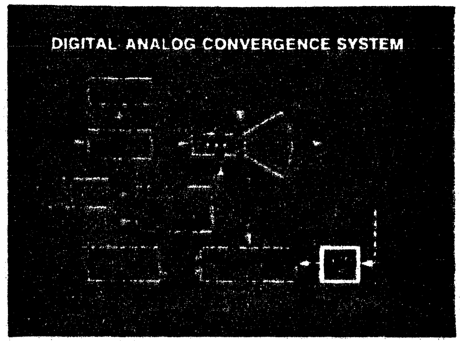


Figure 4b. Digital convergence requires operator intervention to adjust for the current condition of the display.

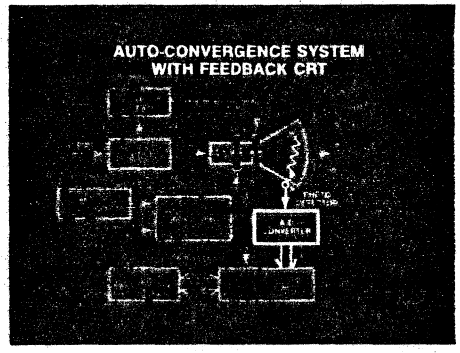


Figure 4c. The 4115B's autoconvergence system removes the user from the convergence adjustment task.

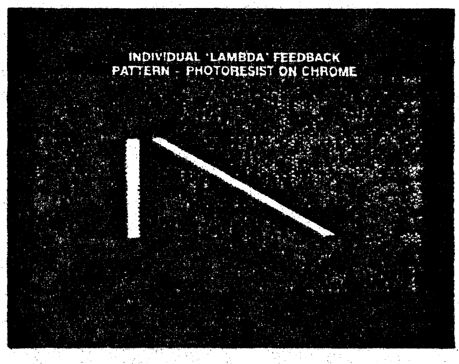


Figure 6. Each phosphor pattern is shaped so the photodetector outputs two pulses.

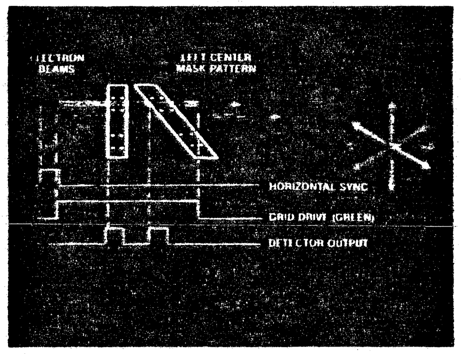


Figure 7. Correcting misconvergence in the 4115B.

Figure 7a. A misconverged yellow line appears as separate red and green lines.

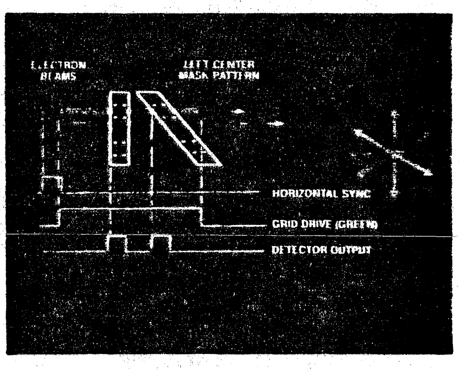


Figure 7b. The processor turns off the red beam to get a photodetector output for the green beam.

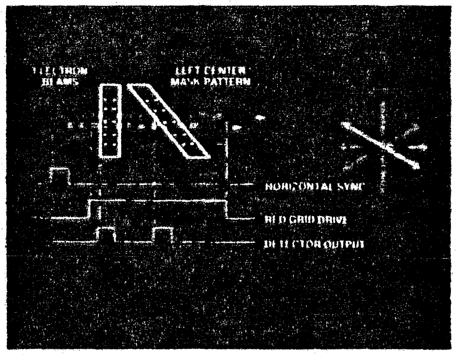


Figure 7c. Removing the green beam and activating the red beam, the processor compares the new pair of pulses with the green beam's detector output.

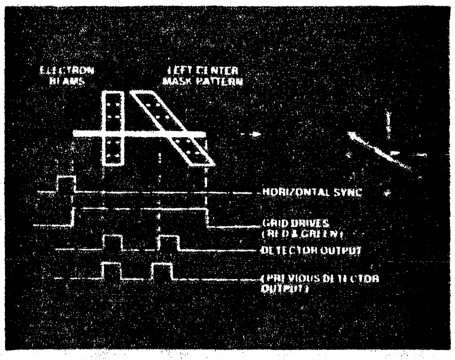
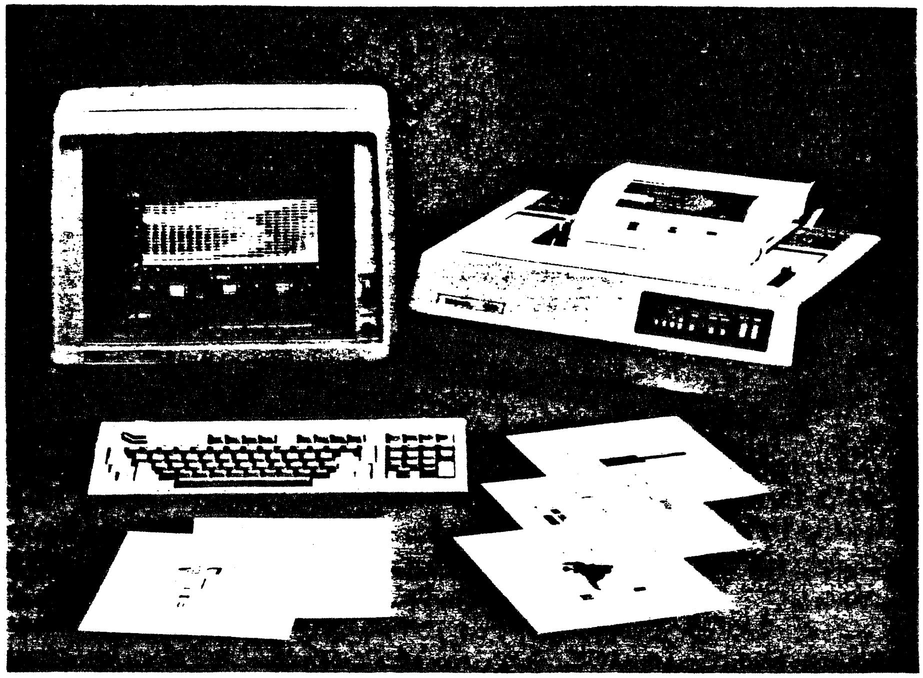


Figure 7d. From the differences in the red and green pulses, the processor computes the required corrections.

SAS® Teams with Tektronix Low Cost Terminals for Statistical Solutions



The SAS System and the Tektronix 4105 Computer Display Terminal produce colored plots and charts for a variety of host environment including IBM, DEC, Prime, and Data General. Instant hard copy output is available on the Tektronix 4695 Color Graphics Copier at the push of a button.

The quality graphics and sharp, bright colors of the Tektronix 4105, 4107 and 4109 Computer Display Terminals combine with SAS/GRAPH^tm^ to produce sophisticated plots, charts, maps and slides at an affordable cost. The display component of the SAS System of software products, SAS/GRAPH converts complex data into easy-to-read graphics. In doing so, it takes advantage of the special hardware features of the 4100 Terminals. Host processing time is reduced.

The SAS System®

A product of the SAS Institute, Inc., the SAS System is a broad range of programming applications. Where most software products are problem-specific, SAS provides the tools for a complete decision support system.

Based on a modular structure, the SAS System integrates:

SAS® for data management, statistical analysis and report writing.

SAS/ETS^{tm} for econometric and financial modeling, and forecasting.

SAS/FSP^{tm} for data entry, letter writing, and financial spreadsheet modeling.

SAS/OR^{tm} for operations research and project management.

SAS/IMS-DL/I^{tm} to interface to DL/1 data bases.

SAS/GRAPH^{tm} for color graphics.

SAS software runs on a variety of host computers including IBM, DEC, Prime, and Data General.

SAS

Data Management and Retrieval

SAS can read data from a variety of sources and in many forms. Once the data have been organized into a SAS data set, data management tools allow:

- subsetting - creates a new data set that is a subset of an original data set,
- concatenating - creates a new data set containing data linked with the original data set by a common variable or variables,
- sorting - rearranges data by a variable or variables,
- merging - combines and matches records by control or key variables,
- interleaving - combines data by indexed variables,
- updating - updates variables in a new data set while maintaining the master data set.

For more complex data management needs, SAS offers systems options to allow manipulation of data outside a SAS data set.

Statistical Analysis

Procedures range from simple descriptive statistics to complex multivariate techniques. Several procedures may be combined into one program and the results of one procedure passed as input to another.

SAS offers procedures for both linear and nonlinear regression analysis as well as response surface and stepwise regression. Several clustering procedures are available for disjoint, hierarchical and overlapping cluster analyses. The output from hierarchical clustering can be used to produce tree diagrams.

Other major statistical procedures include:

- GLM (General Linear Models) - performs all linear and multiple regression analyses, analyses of variance and covariance, and multivariate analyses of variance.
- MATRIX - statements may be written in traditional matrix notation when the analysis needed is not available in SAS.
- FUNCAT - models functions of categorical responses as a linear model, including linear and log-linear categorical models and logistic regression.
- FACTOR - performs a variety of common factor and component analyses and rotations. It can also process output from other procedures.
- DISCRIM - computes linear or quadratic discriminant functions for classifying ob-

4100 Series

servations into two or more groups on the basis of one or more numeric variables.

Report Writing

SAS offers a variety of report writing techniques. Title lines, row and column labels, control of line width and page length, record grouping, subtotals and totals within groups of records are offered. Methods to custom tailor a report include random access to any line and column on the current page before printing, an option that executes and labels a group of statements each time SAS begins printing a new page and flexible formatting of output data.

For complex reports, hierarchically organized tables of descriptive statistics may be produced. A calendar procedure displays data from a SAS data set in a month-by-month calendar format.

SAS/ETS

Modeling and Forecasting

More than 15 procedures for time series forecasting, time series regression techniques and simultaneous equation model estimation and simulation are provided by SAS/ETS. It also produces row-and-column financial reports and loan amortization tables.

Because SAS/ETS integrates with the rest of the SAS System, the data management, statistical analysis and report writing capabilities in SAS can be accessed to perform the fundamental business tasks. SAS/GRAPH can chart the results, in color.

SAS/FSP

Data Entry

A menu-driven, full-screen product, SAS/FSP handles data entry, edit and retrieval tasks as well as letter and report writing. With a single command, data can be entered into an existing data set or summoned from a data set including large SAS data bases already in existence.

The terminal screen can be structured to look exactly like order entry forms, purchase invoices, or consumer questionnaires defining such characteristics as:

- initial values
- maximum and minimum values
- required fields
- colors
- capitalized fields
- protected fields.

Letter Writing

SAS/FSP has two modes to simplify and personalize letter writing. In EDIT mode, the text of the letter is created or changed. In SEND mode, the name, address, greeting, and personalized changes occur and the letter is printed. Tedious error checking and repetitious operations are eliminated.

Financial Spreadsheet Modeling

A procedure for spreadsheet financial reporting makes SAS/FSP the complete information processing tool. Features include:

- ability to conduct "what if" analyses for financial analysis and budget planning
- ability to define any number of spreadsheets, of any size, for reports and analyses
- extensive screen definition and modification capabilities
- a full screen facility for writing and editing modeling statements
- a separate listing for complete error messages
- ability to save sessons in progress
- a complete on-line help facility as well as a capability to design and create help screens
- ability to consolidate several screens into a single screen, producing summary tables or reports.

Managers may password protect models created by a decision support staff so that casual users cannot alter or destroy important models used by the company.

SAS/OR

Operations Research/Project Management

SAS/OR is a business planning and scheduling tool to help decide how a company schedules and uses its manpower, capital, equipment and raw materials to ensure productivity. The Linear Programming procedure can produce recommendations based on opposing alternatives. The Critical Path Method schedules the activities that make up a project. It can take into account the precedence of certain activities and determine a schedule that completes the project in the shortest amount of time.

Other procedures handle general assignment problems; determine minimum cost flow through a network (including transshipment problems), maximum flow, and the shortest path; and determine the quantity of goods to be shipped from supply points to demand points.

SAS/IMS-DL/I

Interfacing to IMS Data Bases

SAS/IMS-DL/I is a Batch Message Processing DL/I interface that provides access to batch IMS/VS or CICS/OS/VS data bases, using SAS under OS batch, TSO, or DOS/VSE. Instead of using lengthy COBOL, PL/I, or assembler programs to read, change, or add data in an IMS data base, the same tasks can be accomplished using SAS statements in SAS/IMS-DL/I.

Of course, the data management, analysis, and reporting tools available in the SAS System can be used on the accessed IMS data base.

SAS/GRAPH

Color Graphics

But managing the data is only part of the picture. Computer graphics is an essential element in analyzing complex data and communicating the information. SAS/GRAPH produces concise reports in a variety of colors and formats.

But because SAS/GRAPH is a component of the SAS System, SAS/GRAPH means more than just color graphics.

A few SAS/GRAPH statements can enhance statistical analyses produced with SAS, financial reports produced with SAS/ETS, or spreadsheets produced wth SAS/FSP. Since SAS/GRAPH procedures are actually SAS procedures, all SAS retrieval, data management, and analysis capabilities can be used with SAS/GRAPH.

Graphs and charts with a wide range of complexity and style can be created, including vertical and horizontal bar charts, pie charts with exploded slices, star charts and block charts. Two or more plots may be superimposed.

Colorful lines of text for poster and slide presentations can be created with SAS/GRAPH. A variety of fonts and special symbols, including musical, weather, and mathematical symbols, can annotate a report or presentation. Custom fonts may be created.

SAS/GRAPH can also produce complex graphs for business and research purposes. Mapping procedures produce choropleth, surface, block, and prism maps. Map data sets for the United States by state and by county, and Canada by province and census district are included.

Several procedures for complex-three dimensional displays are offered. The user can plot the values of three variables, one of which is a contour variable, in two dimensions. The procedure can represent up to 100 levels of the contour variable using a variety of colors, line styles, and patterns.

The combination of the SAS System, Tektronix 4105, 4107 and 4109 Color Graphics Terminals and the Tektronix Family of ink jet color copiers provides a powerful package.

A Low-Cost Workstation

Coupling SAS/GRAPH with the Tektronix 4100 Terminals provides a friendly, economical means for data display and editing. Calling on the built-in line styles, marker types and polygon fill features of a 4100 Terminal, SAS/GRAPH quickly constructs the picture. A palette of 64 colors for the 4105 or 4107 (4096 colors for the 4109), offers a wide selection from which SAS/GRAPH may choose to differentiate the data.

User communication with SAS/GRAPH is enhanced through the dialog area of the screen, the color interface and the joy disk. Keeping prompts and responses in the dialog area eliminates cluttering of the display. By taking advantage of the 4100 color interface, the user can easily experiment with colors in the display without increasing host traffic. The cursor is easily positioned using the joy disk built into the keyboard of the 4100 Terminal.

The excellent display quality and balanced resolution of the 4100 Terminals reduce eye fatique. Graphs or charts produced by SAS/GRAPH can be previewed, then colors, titles, footnotes, company logos and other information for presentations added comfortably.

A satisfactory display may be preserved for a report or handout. Colored hard copies of the 4100 screen are provided by the low-cost 4695 Color Graphics Copier. By simply pushing a button on the keyboard of the 4100 Terminal, a paper copy is quickly obtained. High quality color transparencies may also be produced on the 4695 using SAS/GRAPH.

A Selected Solution Vendor

SAS is an approved SELECTED SOLUTION VENDOR in the Tektronix Solution Vendor Program, an assurance of the high quality and proven functionality of the SAS products and satisfied customers.

For more information contact:

Mr. Michael Camp
SAS Institute
Box 8000
SAS Circle
Cary, NC 27511
(919) 467-8000

Version 2 Firmware Enhances the 4105

by Joel Spinhirne
Tektronix, Inc.
Wilsonville, OR

Tektronix provides an extensive set of no-charge enhancements for the 4105 Computer Display Terminal with the release of Version 2 firmware. All current 4105 users will receive a new set of 4105 ROMs sometime in September 1983. Installing the new firmware involves no more than opening the 4105's rear panel access door, unsnapping existing ROMs, replacing them with the new ROMs, and returning the old ROMs to Tektronix.

Version 2 firmware gives the 4105 full VT100[1] compatibility in its ANSI-Style command set plus several additions to the 4105's set of 4100-Style commands. Complete documentation is delivered with the enhancements, but here is a quick look at what is included.

New ANSI-Style Features

132-Column Mode

The 4105 now has 132-character lines available on demand in ANSI mode. Rather than display marginally readable small characters, the 4105 displays 80 characters per line when in 132-column mode and lets the user scroll the display left and right with the joydisk. Left-right scrolling can also be controlled by the host.

Full VT52[1] Emulation

Many existing applications emulate DEC's VT52 terminal when performing some functions. Version 2 firmware includes a VT52 Mode in which the 4105 executes all VT52 commands.

Double-High, Double-Wide Characters

In addition to the normal character size, all characters on a line can now be double-wide or both double-high and double-wide.

Full Screen Reverse Video

This mode changes all displayed colors in the dialog area to their complementary color.

ISO Standard Graphic Rendition

The full set of text and background colors is now selectable with ISO Standard graphic rendition commands. (It is likely that the ISO Standard will become an ANSI Standard.)

4100-Style Commands

Increased Dialog Buffer Size

The dialog buffer will no longer be restricted to 49 lines. You can assign as many lines to the dialog buffer as available memory permits.

GIN Cursor Positioning

The host can now set where the GIN cursor initially appears on the screen before starting graphic input.

New Hardcopy Support

Several new features are offered for using the 4105 with a 4695 Color Graphics Copier or a printer. (See a related article in this issue of *Tekniques* for information on using the 4105 with a printer.) Here are some of the new command features:

Select Hardcopy Interface lets you tell the 4105 that there is a printer attached rather than a 4695. This prevents inappropriate escape sequences.

Set Dialog Hardcopy Size lets you specify how many pages of dialog buffer you want copied when you press DCopy. You also can set whether to start copying from the top or bottom of the screen and whether form feeds are sent to the printer.

Hardcopy changes so that you can specify reversed black and white (as it is done now) or omit black and white reversal.

Set Copy Size is for use with the 4695. It lets you pick normal or half-size copies. Half-size copies are faster and show 132-character lines unbroken. Normal copies must put lines longer than 80 characters on two lines.

Defining Keys

The new LEARN command, available to the operator as a setup command, makes defining keys much simpler. Friendly prompts guide the operator through the steps needed to define a key. In addition, an operator does not need to learn the complexities of delimiters and literal characters.

Pixel Operations for the 4105

Released at the same time as the Version 2 enhancements, the Pixel Operation Option offers the full set of pixel commands now available on the 4110-Series. This lets the user create displays with direct pixel operations. The user can install the option at anytime by a simple ROM addition.

[1] VT100 and VT52 are trademarks of Digital Equipment Corporation.

"Warranty-Plus" Provides Full Service to 4050 Series, 4110 Series and other Tektronix Products after Warranty

Including the Tektronix Warranty-Plus Service Option N1 at the time of a product purchase is a convenient and cost effective method for maintaining that product like new for the first year of ownership. Warranty-Plus coverage begins as soon as the product warranty ends and is almost identical to the service coverage provided under product warranty. Many product options may be included at no cost.

Offered in the U.S. and many other countries, Warranty-Plus provides full on-site service - all labor, parts and travel. Response is typically within eight work hours, depending on the customer location. Downtime is minimized.

The fixed up-front fee protects against increased costs. No additional budgeting or paperwork is required.

From the Tektronix 4006-1 Graphics Terminal to the newly introduced 4115B Color Graphics Terminal, the Warranty-Plus Service Option N1 is offered with most Tektronix IDD products. A detailed description of the service plan program is contained in a brochure available from your local Tektronix Sales Engineer.

Tektronix 5¼″ Floppy Disks Available Through CPO

Floppy disks for the Tektronix 4170 Local Graphics Processor, Tektronix 4925 Flexible Disk Unit or Tektronix 4926 Opt. 25 Disk Unit may be purchased from Tektronix.

These double-sided, double density floppy disks feature 48 tracks-per-inch, and are packaged 10 to a box.

Call the Tektronix Central Parts Ordering office serving your area (see New Abstracts section in *TEKniques*) and order part number: 119-1583-01.

TEKniques

TEKniques is published by the Information Display Division of Tektronix, Inc. Its application articles and news of new Information Display Division products will be of interest to managers of graphics related projects. Qualified individuals who return the following form may receive *TEKniques*.

Name ______________________________ Date ____________________

Job Title ______________________________ Tek Sales Engineer ____________

Company ______________________________ Department ______________

Address ____

City ____________________ State __________ Zip ______

May we phone you for a discussion about your application? Yes, phone ________________

Primary Product Interest Area (check one)

□ Graphics Terminals

□ Intelligent Graphics Terminals

□ CAD Workstation

□ Desktop Graphics Computers

□ Other Graphics Devices ______________________

Please evaluate *TEKniques*

	Very Useful	Useful	Not Useful
Applications Articles	□	□	□
New Product Information	□	□	□
Programming Tips	□	□	□
New Abstracts	□	□	□

Other ________________

Other features you would like to see included ________________________

____

Your Relationship with/to Computer Graphics (check all that apply)

□ Operator of a Tektronix graphics product

□ User of the output of Tektronix graphics products

□ Programmer of Tektronix graphics products

□ Manager of Tektronix graphics product operators/users

□ Manager of graphics programmers

□ Specify graphics product purchases

□ Approve graphics product purchases

Final Request

If you returned this form from a past issue of *Tekniques,* please ignore this request and accept our thanks. After January 1, 1984, only those from whom the form was received will continue to receive *Tekniques.*

Fold here

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

TEKTRONIX, INC.
Information Display Division
IDD Program Exchange
Mail Stop 63-575
P.O. Box 1000
Wilsonville, Oregon 97070

Programming Tips

Using the 4105 with a Printer

by Joel Spinhirne
Tektronix, Inc.
Wilsonville, OR

Connecting a 4695 Color Graphics Copier to a 4105 Computer Display Terminal produces an unbeatable combination for desktop graphics. Some users, however, use their personal printer or copier primarily for text-only printouts. These users do not need the high resolution color graphics produced by the 4695. Rather than limit users to using any particular device for their printing or hardcopy needs, the printer port on the 4105 accommodates any printer that uses a Centronics-type parallel interface.

This note gives you some help with using the 4105 with a printer. Note that you need Version 2 of the 4105 firmware (see related article in this issue of *TEKniques*) to execute some of the commands mentioned.

Two Ways to Print

There are two sources for text sent to the printer. The first of these is the 4105's dialog buffer. Unless you request the dialog area be disabled, the dialog buffer holds the text making up your most recent dialog with the host. You can have all or part of the dialog buffer sent to the printer. The second source for text is the host itself. You can route text coming from the host directly to the printer.

Copying the Dialog Buffer

Basically, to copy the dialog buffer, all you need do is press DCopy. Still, there are several 4105 commands that help you use a printer.

Unless it receives a command telling it otherwise, the 4105 expects a 4695 on the printer port. It will send escape sequences to control things like background and text colors. It also sends escape sequences to create graphics when it copies tne graphics area. These escape sequences can cause a printer problems - it probably thinks the 4105 is trying to send it a special printer command.

You can tell the 4105 that instead of a 4695 there is a printer attached. Issue a SELECT HARDCOPY INTERFACE command specifying that a printer is connected. (As with almost all 4105 commands, this command can either be sent by the host or typed by the operator as a Setup command.) The 4105 will not send unnecessary escape sequences to the printer.

The SET DIALOG HARDCOPY SIZE command lets you set how many pages of text to print from the dialog area each time you press the DCopy key. This command also lets you set where in the dialog area you want copying to start from and whether form feeds should be sent.

If all you want is a copy of the dialog that is on the screen, press DCopy. (Don't forget to hold down Shift.) Default settings give you the expected results. If you press SCopy, the printer is going to receive all those escape sequences for creating graphics. To get the printer to stop printing, press Cancel.

When you copy the dialog buffer, the 4105 terminates each line with a carriage return (<CR>) and a line feed (<LF>). Set the printer switch on the back panel so that <CR> is not interpreted as <CR><LF> or lines will be double-spaced.

Printing from the Host

Here's how to have text the 4105 receives from the host sent directly to the printer:

1. Enter Setup Mode by pressing Setup.
2. Enter this:

   ```
   COPY HO: TO HC:
   ```
3. Press Setup to exit Setup Mode.

From this point on, the printer is the display device instead of the screen. Everything the host sends goes to the printer. This continues until the 4105 receives the end-of-file character from the host or the operator presses the Cancel key.

When you issue the COPY command, the printer receives either <CR><LF> or <CR> at the end of each line, depending on what the host sends. Make sure the switch setting on the back panel of the printer is set to match what the host does.

Making It Easy to Use

There are several things you can do to make using a printer easier for the operator. Here is one example:

1. Press Setup.
2. Enter:

   ```
   DEFINE F1 /<Ctrl-P><Esc>JC3HO:
   2TO3HC:<Ctrl-P> ~ <CR>/<CR>
   ```

Note:

<Ctrl-P> means hold down Ctrl and hit P; screen shows snoopy DL.
<Esc> means press Esc.
<CR> means press Return.
<CR> after ~ rubs out the ~; screen shows a snoopy CR.
This all assumes you use the default edit character and key execute character.

3. Now, when you want something to go to the printer, press F1 instead of Return after any command that normally prints to the screen. Press Cancel when the printing is done.

For example, to get text to print on the screen you might enter:

```
TYPE MYFILE<CR>
```

To get MYFILE to go to the printer, you would enter:

```
TYPE MYFILE<F1>
```

<F1> means press F1.

When the printing is done, press Cancel. To see your operating system's prompt and to return the cursor to the left margin, press Return or whatever key usually gives a prompt.

Of course, if the host sends a unique character after a listing, you could avoid using the Cancel key by setting the 4105's EOF string to this character. Another thing you could try would be putting the EOF character by itself in a file and then put something in the F1 macro so the EOF file always gets printed after the requested file.

Printers with the 4107 and 4109

Everything works exactly the same.

4110 Terminal Parameter Settings for 4010 GIN

by Susan Scoonover
and Carl Goodwin
Tektronix, Inc.
Wilsonville, OR

The setting of four parameters on a 4110 Terminal allows most hosts to execute existing 4010 GIN programs correctly on the 4110. These settings aren't guaranteed to work with all hosts, but seem to work for most. Do not type the parentheses in the following examples; they are included to indicate ASCII control characters.

In SETUP mode, key in:

REOM 1

(specifies that an end of message indicator is sent after each GIN report)

EOM (NUL) (NUL)

A NUL is entered by pressing CTRL SHIFT @.

(eliminates end-of-message characters)

EOL (CR)

A (CR) is entered by using delimiters, the literal character and pressing the RETURN, key i.e., EOL /∼RETURN/

(specifies end-of-line string)

BYPASSCANCEL (CR), or (LF) or (NUL)

A (CR) is entered by BYPASS RETURN

A (LF) is entered by BYPASS (LF)

A (NUL) is entered by BYPASS CTRL SHIFT @

(specifies the character which terminates Bypass mode in the terminal)

Of all these settings, most problems have been related to the parameter of the last command, the bypass cancel character. 4010 GIN enables bypass mode on the 4110 terminal whenever coordinates are sent to the host. Bypass mode must be cancelled before the terminal will pay attention to any subsequent transmissions from the host. Bypass mode is cancelled when the terminal receives the bypass cancel character. This character should be the last character of the host's echo to the terminal's report.

To determine how the host responds to a (CR), which is the EOL string, enter SETUP mode and type SNO YES. Exit SETUP mode and when on line to the host, press the RETURN key. Note what is written to the display, if anything. Re-enter SETUP mode, enter SNO NO, and set the bypass cancel character using the table below as a guide. Most hosts echo a (CR), (LF), or (CR) (LF). If anything else is echoed (or nothing is echoed), set the bypass cancel character to (NUL) to disable bypass mode.

Character(s) echoed back by host in response to (CR):	In SETUP mode, set the bypass cancel character to:
(CR)	(CR)
(LF)	(LF)
(CR)(LF)	(LF)
(LF)(CR)	(CR)
anything else	(NUL) to disable bypass

Plotter Loop Through Using 4110 Series Terminals

by Susan Scoonover
and Carl Goodwin
Tektronix, Inc.
Wilsonville, OR

A plotter may be connected between the host and the 4110 Terminal just as it was done with the 4014 Terminal. However, when the plotter is placed between the host and the terminal, the features supported by the terminal's three port peripheral interface, such as SPOOL, PLOT, COPY, aren't available with that plotter.

Spooling from the 4110 Series Terminals

by Susan Scoonover
and Carl Goodwin
Tektronix, Inc.
Wilsonville, OR

A file on a disk in a 4110 Series Terminal can be spooled to a plotter using the three port peripheral interface. While a file is being spooled, the same disk drive can be accessed by other commands, such as COPY, or SAVE. The SPOOL command doesn't require exclusive use of the disk drive.

4110 GIN Pick without Operator

by Dan Taylor
Tektronix, Inc.
Wilsonville, OR

Normally the PICK function of the 4110 Series Terminals is used to display the GIN cursor and permit the terminal operator to move it to select one segment (or part of a segment) from among several which have been defined and displayed on the screen, for example, a menu item or a drawing component. There may be a time, however, when the host program would like to check for a pick "hit" without operator intervention.

The <report-GIN-point> command with a parameter of <int: 1> will force the terminal to respond with the segment number and pick ID of the visible detectable segment within the pick aperture of the GIN cursor (or 0 if there isn't a segment). What may not be clear is that the host must first issue a <set-segment-position> command* to position the GIN cursor at the point under consideration. Once the <report-GIN-point> command is issued, the host can restore the GIN cursor to its position prior to the <set-segment-position> command.

*The default cursor is segment 0. If a segment greater than 0 is used for the GIN cursor, the <set-segment-image-transform> command may also be used to relocate the GIN cursor.

4110 GIN Cursors for Picking

by Dan Taylor
Tektronix, Inc.
Wilsonville, OR

Regardless of the window/viewport settings or zooming operations on a 4112, 4113, 4114, or 4115 terminal, the physical size of markers on the screen doesn't change nor does the size of the pick aperture.* Therefore, it is often convenient to have a square as the GIN cursor with the size of the square the same as the size of the pick aperture. This can be done by opening a segment and putting marker 8 (or 6 or 10) at 0,0, and then closing that segment:

(ESC)LF 0 0

(ESC)MM 8

(ESC)SO n (where n is the segment number)

(ESC)LH 0 0

(ESC)SC

Next, set the pick aperture to the size of the marker:

(ESC)IA 40

When you wish to use the square as the GIN cursor, issue the <set-GIN-cursor> command. The operator can see the size of the pick aperture.

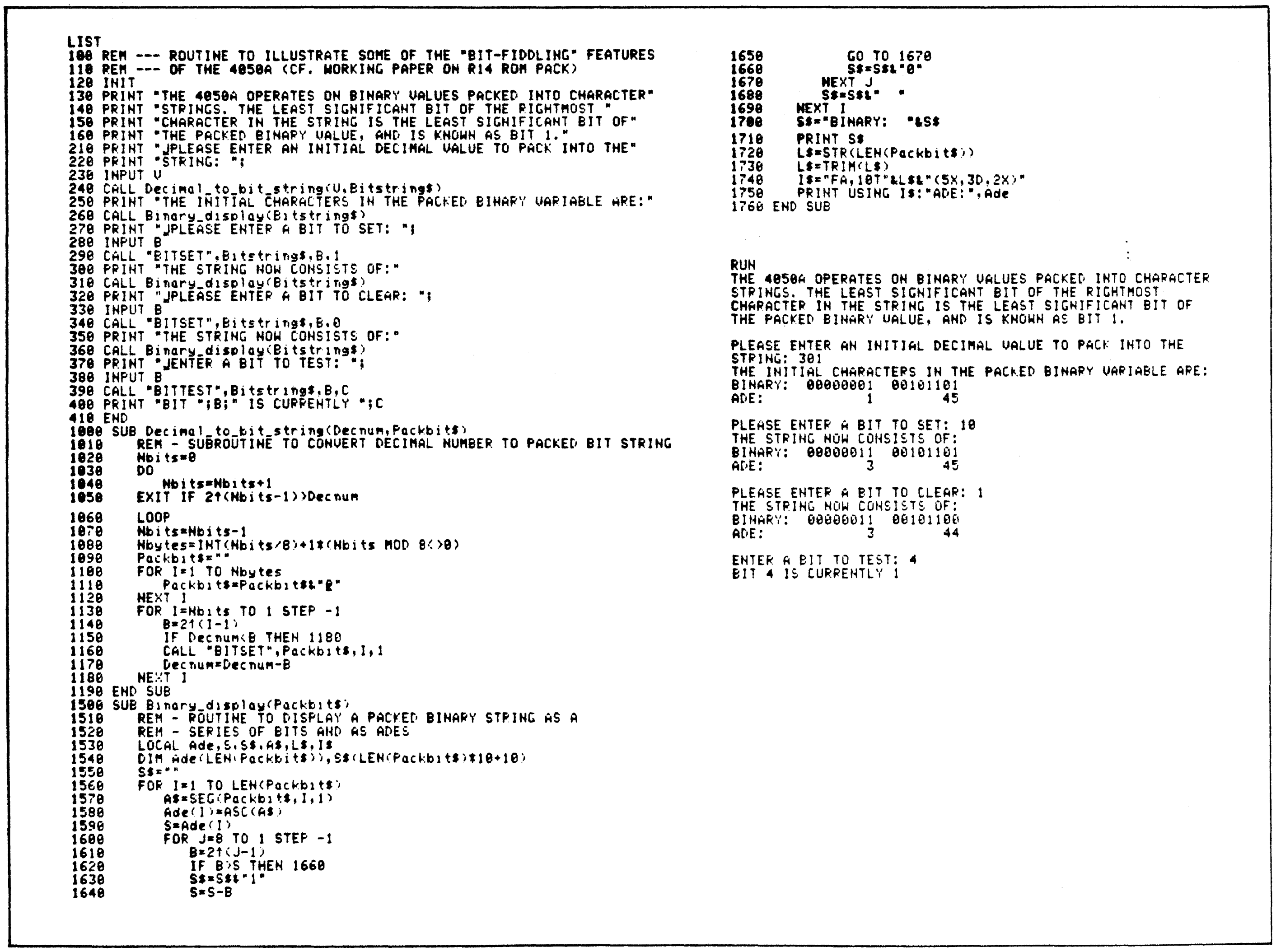
*The pick aperture is a square in normalized screen space, centered on the current beam position. During a GIN pick operation, part of the segment picked must fall within the pick aperture.

4050A Bit Manipulation Routine

by Lee Rainey
Tektronix, Inc.
Dayton, OH

The 4050A Series includes bit manipulation routines which set, clear and test individual bits in a variable to facilitate data transmission to and from devices on the GPIB. The bit "calls" support 65,535 bits, although the operations become cumbersome after 34 bits, or 10 decimal digits, due to the 4050's switch to scientific notation.

The following example illustrates some of the "bit fiddling" capabilities.



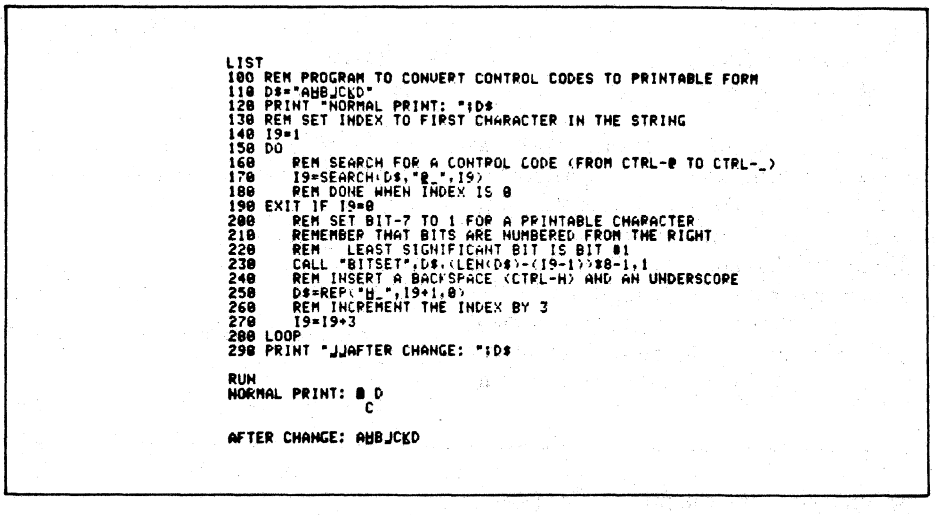
```
LIST
100 REM --- ROUTINE TO ILLUSTRATE SOME OF THE "BIT-FIDDLING" FEATURES
110 REM --- OF THE 4050A (CF. WORKING PAPER ON R14 ROM PACK)
120 INIT
130 PRINT "THE 4050A OPERATES ON BINARY VALUES PACKED INTO CHARACTER"
140 PRINT "STRINGS. THE LEAST SIGNIFICANT BIT OF THE RIGHTMOST "
150 PRINT "CHARACTER IN THE STRING IS THE LEAST SIGNIFICANT BIT OF"
160 PRINT "THE PACKED BINARY VALUE, AND IS KNOWN AS BIT 1."
210 PRINT "JPLEASE ENTER AN INITIAL DECIMAL VALUE TO PACK INTO THE"
220 PRINT "STRING: ";
230 INPUT V
240 CALL Decimal_to_bit_string(V,Bitstring$)
250 PRINT "THE INITIAL CHARACTERS IN THE PACKED BINARY VARIABLE ARE:"
260 CALL Binary_display(Bitstring$)
270 PRINT "JPLEASE ENTER A BIT TO SET: ";
280 INPUT B
290 CALL "BITSET",Bitstring$,B,1
300 PRINT "THE STRING NOW CONSISTS OF:"
310 CALL Binary_display(Bitstring$)
320 PRINT "JPLEASE ENTER A BIT TO CLEAR: ";
330 INPUT B
340 CALL "BITSET",Bitstring$,B,0
350 PRINT "THE STRING NOW CONSISTS OF:"
360 CALL Binary_display(Bitstring$)
370 PRINT "JENTER A BIT TO TEST: ";
380 INPUT B
390 CALL "BITTEST",Bitstring$,B,C
400 PRINT "BIT ";B;" IS CURRENTLY ";C
410 END
1000 SUB Decimal_to_bit_string(Decnum,Packbit$)
1010     REM - SUBROUTINE TO CONVERT DECIMAL NUMBER TO PACKED BIT STRING
1020     Nbits=0
1030     DO
1040        Nbits=Nbits+1
1050     EXIT IF 2↑(Nbits-1)>Decnum

1060     LOOP
1070     Nbits=Nbits-1
1080     Nbytes=INT(Nbits/8)+1*(Nbits MOD 8<>0)
1090     Packbit$=""
1100     FOR I=1 TO Nbytes
1110        Packbit$=Packbit$&"@"
1120     NEXT I
1130     FOR I=Nbits TO 1 STEP -1
1140        B=21(I-1)
1150        IF Decnum<B THEN 1180
1160        CALL "BITSET",Packbit$,I,1
1170        Decnum=Decnum-B
1180     NEXT I
1190 END SUB
1500 SUB Binary_display(Packbit$)
1510     REM - ROUTINE TO DISPLAY A PACKED BINARY STRING AS A
1520     REM - SERIES OF BITS AND AS ADES
1530     LOCAL Ade,S,S$,A$,L$,I$
1540     DIM Ade(LEN(Packbit$)),S$(LEN(Packbit$)*10+10)
1550     S$=""
1560     FOR I=1 TO LEN(Packbit$)
1570        A$=SEG(Packbit$,I,1)
1580        Ade(I)=ASC(A$)
1590        S=Ade(I)
1600        FOR J=8 TO 1 STEP -1
1610           B=2↑(J-1)
1620           IF B>S THEN 1660
1630           S$=S$&"1"
1640           S=S-B
1650           GO TO 1670
1660           S$=S$&"0"
1670        NEXT J
1680        S$=S$&"  "
1690     NEXT I
1700     S$="BINARY:  "&S$
1710     PRINT S$
1720     L$=STR(LEN(Packbit$))
1730     L$=TRIM(L$)
1740     I$="FA,10T"&L$&"(5X,3D,2X)"
1750     PRINT USING I$:"ADE:",Ade
1760 END SUB
```

```
RUN
THE 4050A OPERATES ON BINARY VALUES PACKED INTO CHARACTER
STRINGS. THE LEAST SIGNIFICANT BIT OF THE RIGHTMOST
CHARACTER IN THE STRING IS THE LEAST SIGNIFICANT BIT OF
THE PACKED BINARY VALUE, AND IS KNOWN AS BIT 1.

PLEASE ENTER AN INITIAL DECIMAL VALUE TO PACK INTO THE
STRING: 301
THE INITIAL CHARACTERS IN THE PACKED BINARY VARIABLE ARE:
BINARY:  00000001  00101101
ADE:            1        45

PLEASE ENTER A BIT TO SET: 10
THE STRING NOW CONSISTS OF:
BINARY:  00000011  00101101
ADE:            3        45

PLEASE ENTER A BIT TO CLEAR: 1
THE STRING NOW CONSISTS OF:
BINARY:  00000011  00101100
ADE:            3        44

ENTER A BIT TO TEST: 4
BIT 4 IS CURRENTLY 1
```

Using 4050A BITSET to Print Control Codes

by Mark Mehall
Tektronix, Inc.
Wilsonville, OR

Using two of the new commands in the 4050A Series, the 32 ASCII control codes can easily be converted into printable characters. SEARCH locates each control character in a string (line 170). BITSET replaces the seventh bit (most significant) to a 1 changing the control code to its upper case character equivalent (line 230). A backspace and underscore are inserted into the string to identify the character as a control code replacement (line 250).



```
LIST
100 REM PROGRAM TO CONVERT CONTROL CODES TO PRINTABLE FORM
110 D$="AHBJCKD"
120 PRINT "NORMAL PRINT: ";D$
130 REM SET INDEX TO FIRST CHARACTER IN THE STRING
140 I9=1
150 DO
160     REM SEARCH FOR A CONTROL CODE (FROM CTRL-@ TO CTRL-_)
170     I9=SEARCH(D$,"@_",I9)
180     REM DONE WHEN INDEX IS 0
190 EXIT IF I9=0
200     REM SET BIT-7 TO 1 FOR A PRINTABLE CHARACTER
210     REMEMBER THAT BITS ARE NUMBERED FROM THE RIGHT
220     REM   LEAST SIGNIFICANT BIT IS BIT #1
230     CALL "BITSET",D$,(LEN(D$)-(I9-1))*8-1,1
240     REM INSERT A BACKSPACE (CTRL-H) AND AN UNDERSCORE
250     D$=REP("H_",I9+1,0)
260     REM INCREMENT THE INDEX BY 3
270     I9=I9+3
280 LOOP
290 PRINT "JJAFTER CHANGE: ";D$

RUN
NORMAL PRINT: ■ D
                C

AFTER CHANGE: AHBJCKD
```

4050A CONFIG, POLL and UBOUND Commands Locate Active Instruments on GPIB

by Lee Rainey
Tektronix, Inc.
Dayton OH

A-Series enhancements to the 4050 Desktop Computers add several GPIB commands including CONFIGURE and POLL. The functions of these two commands along with that of the UBOUND command are illustrated in the example which determines active devices on the GPIB, including those which are talk-only.

Active talker-listeners such as the Tektronix 4663 Plotter or 4924 Tape Drive are detected with a call to CONFIG (line 140). When this command is executed, the 4050 asserts the ATN signal line of the GPIB and issues listen addresses. After the 4050 releases the ATN line, it checks for those devices asserting the NDAC line (telling the 4050 they are ready to accept data) and returns in the array the addresses of those active devices (Gpib in line 140).

Because talk-only devices can't receive data and, consequently, don't implement the NDAC signal, the CONFIG command won't detect them. But using a new timeout feature (TIMSET) and the 4052R09 Clock ROM Pack, the routine detects a talk-only device such as the 4956 Tablet.

Line 150 sets a timeout delay for the POLL command and then polls all GPIB addresses not returned by CONFIG. If no device is at that address, the POLL times out. If a device is there, it returns a status value before timeout. The time difference is detected by the clock ROM (line 230) and the address is the value of variable I.

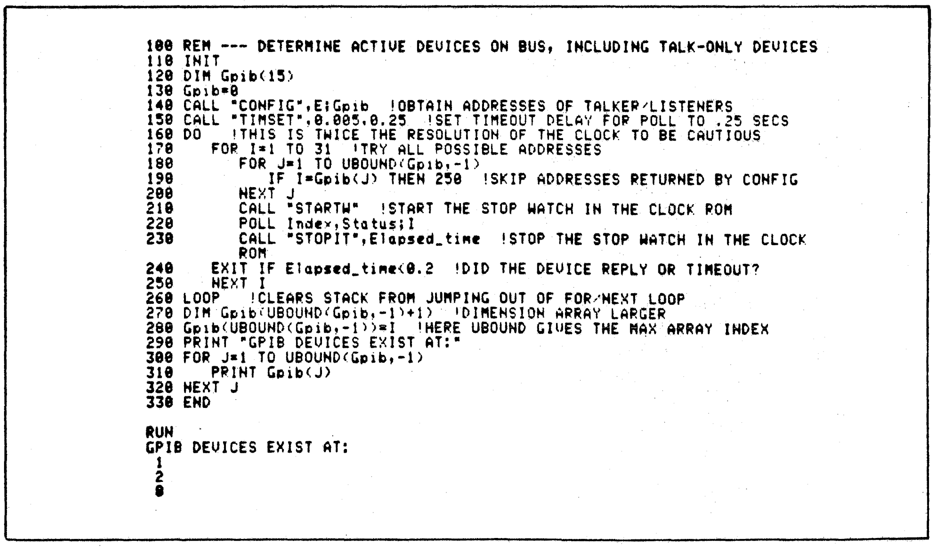
UBOUND plays several roles. Since CONFIG redimensions the array of addresses, the new array size is determined by UBOUND in line 180. When POLL locates an active talker-only, the routine transfers to line 270 where UBOUND + 1 becomes the new size of the address array (the A-Series supports upward dimensioning); in line 280 UBOUND returns the index of the array element in which to place the address.

Note the use of a DO/LOOP to clear the stack after leaving a FOR/NEXT loop.

While the routine assumes only one talker-only is on the bus, it can be changed to check for more.

DELETE lines 160, 250 and 260.
CHANGE line 240 to read:
    240 IF Elapsed_time 0.2 THEN 285
INSERT line 285:
    285 NEXT I

One caveat: An active listen-only device without a specific address, such as a data logger, will respond to all addresses.



```
100 REM --- DETERMINE ACTIVE DEVICES ON BUS, INCLUDING TALK-ONLY DEVICES
110 INIT
120 DIM Gpib(15)
130 Gpib=0
140 CALL "CONFIG",E;Gpib  !OBTAIN ADDRESSES OF TALKER/LISTENERS
150 CALL "TIMSET",0.005,0.25  !SET TIMEOUT DELAY FOR POLL TO .25 SECS
160 DO   !THIS IS TWICE THE RESOLUTION OF THE CLOCK TO BE CAUTIOUS
170    FOR I=1 TO 31  !TRY ALL POSSIBLE ADDRESSES
180       FOR J=1 TO UBOUND(Gpib,-1)
190          IF I=Gpib(J) THEN 250  !SKIP ADDRESSES RETURNED BY CONFIG
200       NEXT J
210       CALL "STARTW"  !START THE STOP WATCH IN THE CLOCK ROM
220       POLL Index,Status;I
230       CALL "STOPIT",Elapsed_time  !STOP THE STOP WATCH IN THE CLOCK
          ROM
240    EXIT IF Elapsed_time<0.2  !DID THE DEVICE REPLY OR TIMEOUT?
250    NEXT I
260 LOOP   !CLEARS STACK FROM JUMPING OUT OF FOR/NEXT LOOP
270 DIM Gpib(UBOUND(Gpib,-1)+1)  !DIMENSION ARRAY LARGER
280 Gpib(UBOUND(Gpib,-1))=I  !HERE UBOUND GIVES THE MAX ARRAY INDEX
290 PRINT "GPIB DEVICES EXIST AT:"
300 FOR J=1 TO UBOUND(Gpib,-1)
310    PRINT Gpib(J)
320 NEXT J
330 END

RUN
GPIB DEVICES EXIST AT:
 1
 2
 8
```

4050A Commands Aid Program Development

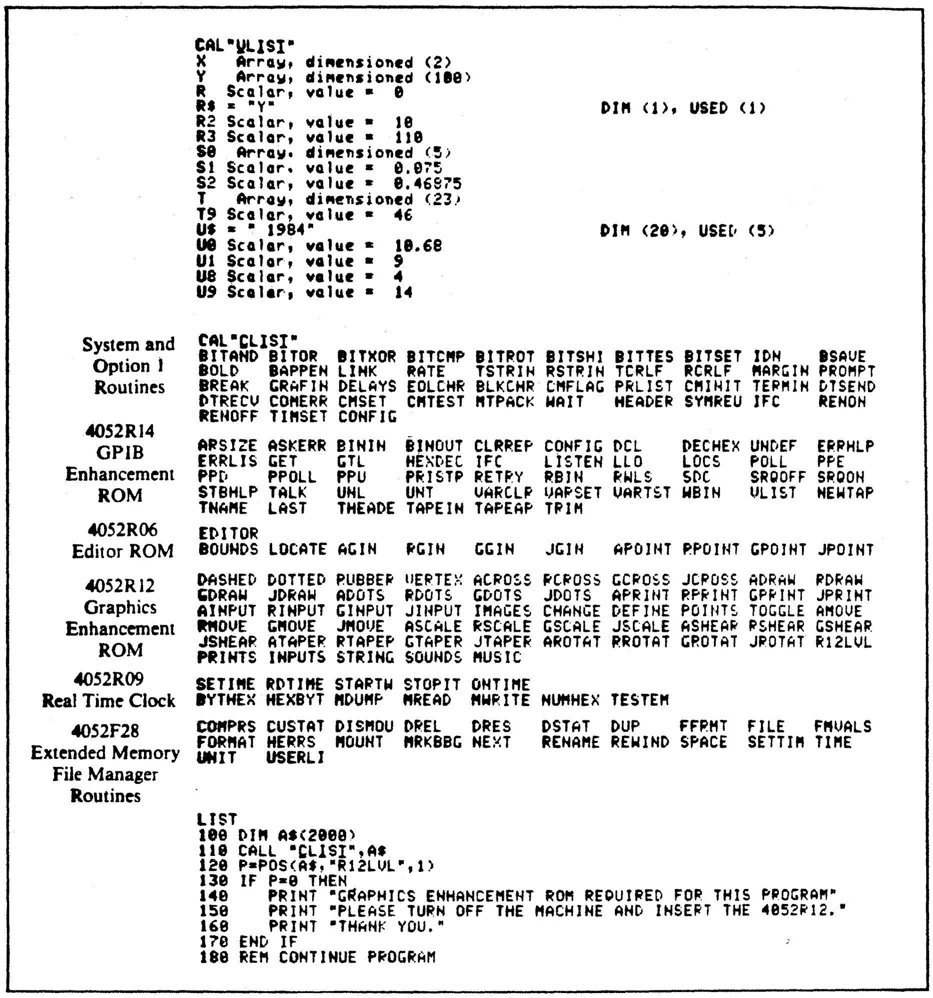
by Mark Mehall
Tektronix, Inc.
Wilsonville, OR

Two commands are included in the 4052A/4054A which facilitate program development and debugging. Call "VLIST" produces a list of all variables currently defined. The variable name, type, value, dimension size and number of elements used are displayed. (See example below.)

Call "CLIST" lists all the "call" routines available. The following example shows the system calls (except those starting with a control character), ROM Pack routines and the extended memory file manager calls. ROM Packs in both the ROM Expander (4050E01) and backpack are accessed.

The "CLIST" output may also be directed to a variable that is dimensioned large enough to hold the information returned. This allows a program to check for a required ROM Pack. (See the following example.)

Note that the first, third and fifth characters of "VLIST" and "CLIST" are control characters. The "CTRL" key is held down while the letters are typed.



```
                     CAL"ULISI"
                     X   Array, dimensioned (2)
                     Y   Array, dimensioned (100)
                     R  Scalar, value =  0
                     R$ = "Y"                                   DIM (1), USED (1)
                     R2 Scalar, value =  10
                     R3 Scalar, value =  110
                     S0  Array. dimensioned (5)
                     S1 Scalar. value =  0.075
                     S2 Scalar, value =  0.46875
                     T   Array, dimensioned (23)
                     T9 Scalar, value =  46
                     U$ = " 1984"                               DIM (20), USED (5)
                     U0 Scalar, value =  10.68
                     U1 Scalar, value =  9
                     U8 Scalar, value =  4
                     U9 Scalar, value =  14

   System and        CAL"CLISI"
   Option 1          BITAND BITOR  BITXOR BITCMP BITROT BITSHI BITTES BITSET IDN    BSAVE
   Routines          BOLD   BAPPEN LINK   RATE   TSTRIN RSTRIN TCRLF  RCRLF  MARGIN PROMPT
                     BREAK  GRAFIN DELAYS EOLCHR BLKCHR CMFLAG PRLIST CMINIT TERMIN DTSEND
                     DTRECU COMERR CMSET  CMTEST MTPACK WAIT   HEADER SYMREU IFC    RENON
                     RENOFF TIMSET CONFIG
   4052R14
   GPIB              ARSIZE ASKERR BININ  BINOUT CLRREP CONFIG DCL    DECHEX UNDEF  ERRHLP
   Enhancement       ERRLIS GET    GTL    HEXDEC IFC    LISTEN LLO    LOCS   POLL   PPE
   ROM               PPD    PPOLL  PPU    PRISTP RETRY  RBIN   RWLS   SDC    SRQOFF SRQON
                     STBHLP TALK   UNL    UNT    VARCLR VARSET VARTST WBIN   ULIST  NEWTAP
                     TNAME  LAST   THEADE TAPEIN TAPEAP TRIM
   4052R06           EDITOR
   Editor ROM        BOUNDS LOCATE AGIN   RGIN   GGIN   JGIN   APOINT RPOINT GPOINT JPOINT

   4052R12           DASHED DOTTED RUBBER VERTEX ACROSS RCROSS GCROSS JCROSS ADRAW  RDRAW
   Graphics          GDRAW  JDRAW  ADOTS  RDOTS  GDOTS  JDOTS  APRINT RPRINT GPRINT JPRINT
   Enhancement       AINPUT RINPUT GINPUT JINPUT IMAGES CHANGE DEFINE POINTS TOGGLE AMOVE
   ROM               RMOVE  GMOVE  JMOVE  ASCALE RSCALE GSCALE JSCALE ASHEAR RSHEAR GSHEAR
                     JSHEAR ATAPER RTAPER GTAPER JTAPER AROTAT RROTAT GROTAT JROTAT R12LVL
                     PRINTS INPUTS STRING SOUNDS MUSIC
   4052R09           SETIME RDTIME STARTW STOPIT ONTIME
   Real Time Clock   BYTHEX HEXBYT MDUMP  MREAD  MWRITE NUMHEX TESTEM

   4052F28           COMPRS CUSTAT DISMOU DREL   DRES   DSTAT  DUP    FFRMT  FILE   FMVALS
   Extended Memory   FORMAT HERRS  MOUNT  MRKBBG NEXT   RENAME REWIND SPACE  SETTIM TIME
   File Manager      UNIT   USERLI
   Routines

                     LIST
                     100 DIM A$(2000)
                     110 CALL "CLISI",A$
                     120 P=POS(A$,"R12LVL",1)
                     130 IF P=0 THEN
                     140     PRINT "GRAPHICS ENHANCEMENT ROM REQUIRED FOR THIS PROGRAM"
                     150     PRINT "PLEASE TURN OFF THE MACHINE AND INSERT THE 4052R12."
                     160     PRINT "THANK YOU."
                     170 END IF
                     180 REM CONTINUE PROGRAM
```

DIR from 4907 or Extended Memory

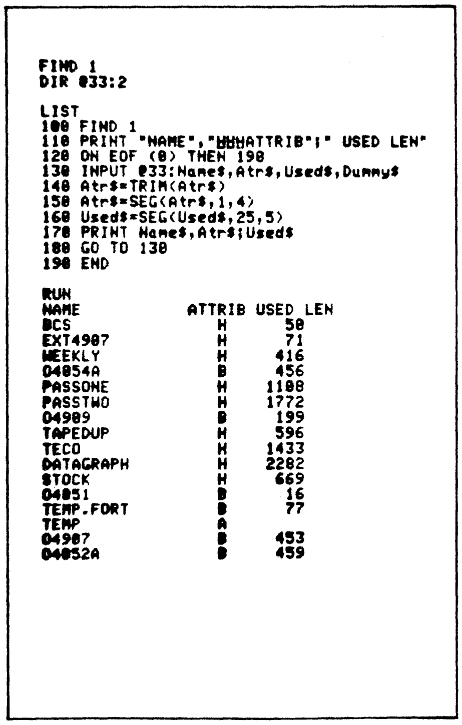
by Mark Mehall
Tektronix, Inc.
Wilsonville, OR

By sending a full directory to the tape and then selecting only those items of interest, a quick, concise directory of 4907 disk or 4052/54 Extended Memory files is obtained:

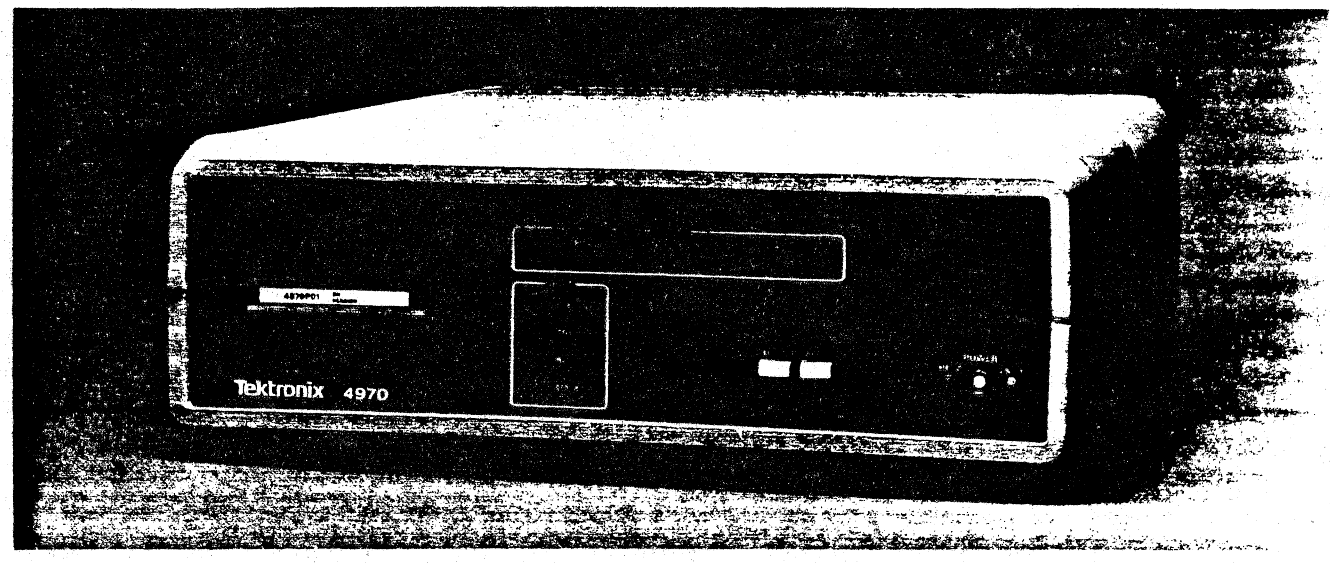
Editor's Note: A directory of Extended Memory files may also be sent to a GPIB address:

```
D = 2
FIN @D:1
UNIT 4
DIR @ D:
```

(Change line 130 device number from 33 to D.)



```
FIND 1
DIR @33:2

LIST
100 FIND 1
110 PRINT "NAME","WWWATTRIB";" USED LEN"
120 ON EOF (0) THEN 190
130 INPUT @33:Name$,Atr$,Used$,Dummy$
140 Atr$=TRIM(Atr$)
150 Atr$=SEG(Atr$,1,4)
160 Used$=SEG(Used$,25,5)
170 PRINT Name$,Atr$;Used$
180 GO TO 130
190 END

RUN
NAME                ATTRIB USED LEN
BCS                    H        50
EXT4907                H        71
WEEKLY                 H       416
04054A                 B       456
PASSONE                H      1108
PASSTWO                H      1772
04909                  B       199
TAPEDUP                H       596
TECO                   H      1433
DATAGRAPH              H      2282
STOCK                  H       669
04051                  B        16
TEMP.FORT              B        77
TEMP                   A
04907                  B       453
04052A                 B       459
```



Up to four Tektronix graphics terminals may be interfaced through the Tektronix 4970 Cluster Controller to the SNA/SDLC environment of IBM.

Tektronix 4970 Cluster Controller Accesses IBM SNA Networks

The Tektronix 4970 Cluster Controller, with its companion 4970P01 and 4970P02 software, opens new doors for Tektronix graphics terminals. By providing access to the multiple hosts and multiple applications of IBM SNA networks, the 4970 Cluster Controller extends the graphics work environments of Tektronix terminals.

Because the IBM mainframe environment makes extensive use of 3270 Series terminals and 3270 terminal support systems, the 4970 has been designed to allow Tektronix graphics terminals to connect to an SNA environment as if they were IBM 3270-type terminals. Communicating to the host like an IBM 3274 or 3276 terminal controller, the 4970 accepts data from as many as four Tektronix asynchronous ASCII terminals and converges it to a single synchronous EBCDIC data stream transmitting it at rates up to 9.6 kbps.

Special routines transmit PLOT 10-style graphics. This is done by installing 4970 translation routines in the host's graphics software library permitting applications programs which call PLOT 10 (or PLOT 10 compatible) graphics software routines* to communicate with 4970-linked terminals. These special routines convert the graphics data to an EBCIDIC, 3270 character stream that is acceptable to host teleprocessing software. The 4970 subsequently receives this data stream and translates it back to PLOT 10-style graphics suitable to Tektronix terminals. The process is reversed for graphics data transmission to the host. *No modification* to the teleprocessing software is needed.

In addition to data conversion, the 4970 software automatically appends the data and structuring necessary to conform to both SNA requirements and the SDLC link protocol.

By converting asynchronous communications to synchronous communications, the 4970 Cluster Controller speeds transmission and improves system efficiency. Transmission line and port costs are also reduced since multiple terminals share one communications line from the 4970 to the host.

Programmability a Key Feature

Its programmability makes the 4970 Cluster Controller adaptable for present and future communications needs. Both the communications protocol and the 4970 operating system reside on a small magnetic tape cassette. Changing protocols, adding new features, and maintaining network compatibility are accomplished by simply inserting an updated cassette from Tektronix.

Two software packages give 4970 users a choice. Both packages provide 3270-type terminal-to-host communications with SNA network compatibility and SDLC protocol. The 4970 P01 SNA/SDLC/3270 Controller Software is used most often with Tektronix terminals that are dedicated to a particular application, such as CAD, which has no need for full-screen alphanumerics. On the other hand, 4970P02 3270 Emulation SNA/SDLC Controller Software is capable of 3270 full-screen emulation which allows Tektronix ter-

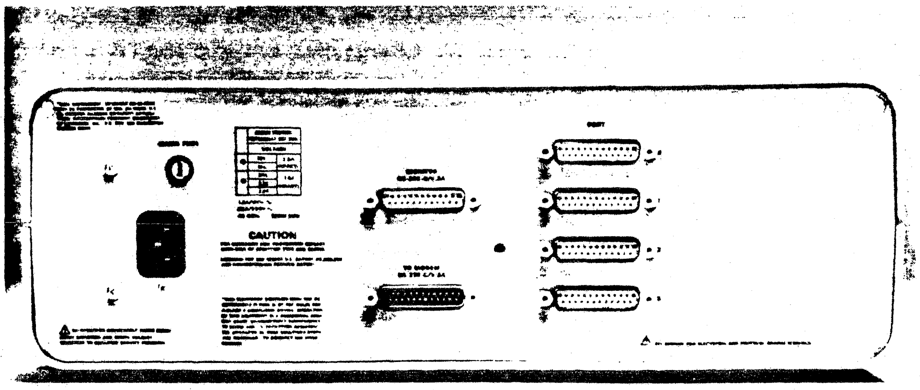


Figure 1. 4970 Rear Panel.

*PLOT 10 packages were discussed in *TEKniques* Vol. 7, No. 1.

minals to access and use TSO, SPF, IMS, CMS and CICS alphanumeric applications.

The 3270 Emulation (4970P02) is particularly suited to users working with full screen ditors like SPF, or users who need to access data entry or retrieval programs that require terminals with field and/or cursor control to IBM 3270 standards.

To access the versatility of the 4970, a monitor port on its back panel (Figure 1) allows communications parameters such as network device addresses and terminal-support parameters to be easily set. Parameters are stored in battery-powered memory that saves the preset values. This port allows access to memory in general, so special user required adaptations can be installed, within limits of course! This port has password protection so system security can be maintained.

Virtually any ASCII terminal can be used with the 4970 depending on that terminal's features.

Some general features of the 4970 include:

- 128 K bytes of RAM memory.
- Port status LEDs which show the user port sending or receiving data.
- Host status LEDs which represent Data Carrier Detect, Transmit Data, Receive Data and Error, the latter indicating an error detected and corrected by SDLC.

Extensive self-test diagnostics which display their findings via the Host and Port LEDs.

Flexible Configuration

The 4970 Cluster Controller and accompanying software supports the older Tektronix terminals as well as the newly introduced 4100 line. The 4020A, 4050A, 4100, 4110, 4110A and 4110B Series of graphics terminals can be used in the IBM SNA environment; 4010, 4020 and 4050 Series are compatible when flagging has been added or if terminal-to-host transmission is at keystroke speed.

Tektronix graphics terminals operating in SNA networks give users the world standard in graphics as well as SNA data accessibility.

The 4970 Cluster Controller and terminals are physically installed and supported as though they were 3270 devices. Figure 2 shows alternative methods of connecting the 4970 to a host or front-end processor. Figure 3 shows options for connecting terminals to the 4970.

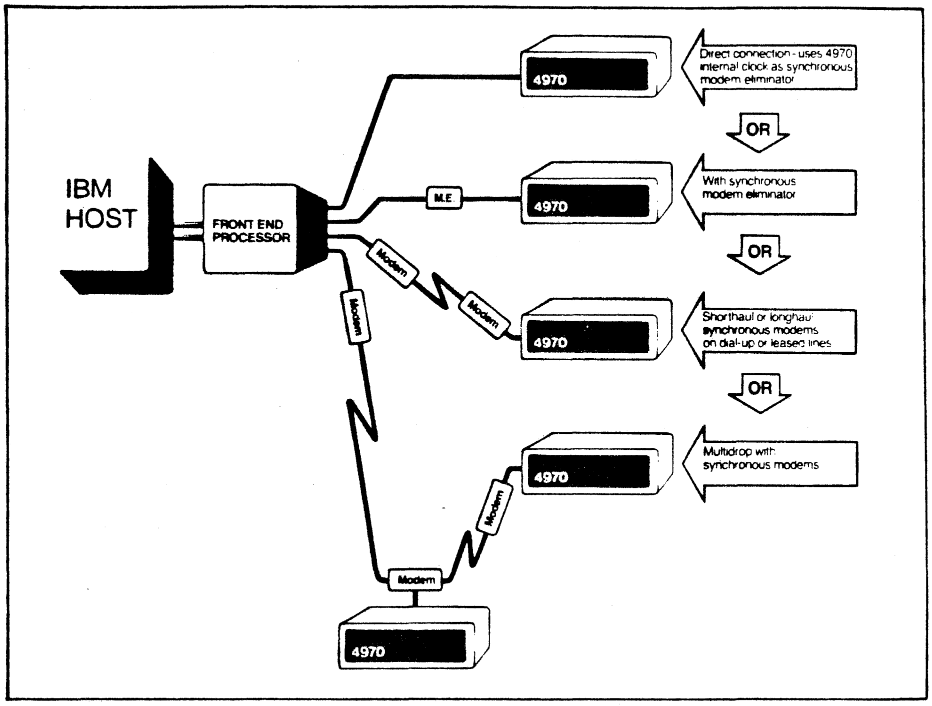

IBM HOST
FRONT END PROCESSOR
4970
Direct connection - uses 4970 internal clock as synchronous modem eliminator
OR
M.E
4970
With synchronous modem eliminator
OR
Modem
Modem
4970
Shorthaul or longhaul synchronous modems on dial-up or leased lines
OR
Modem
Modem
4970
Multidrop with synchronous modems
Modem
4970

Figure 2. Host-to-4970 connection alternatives.

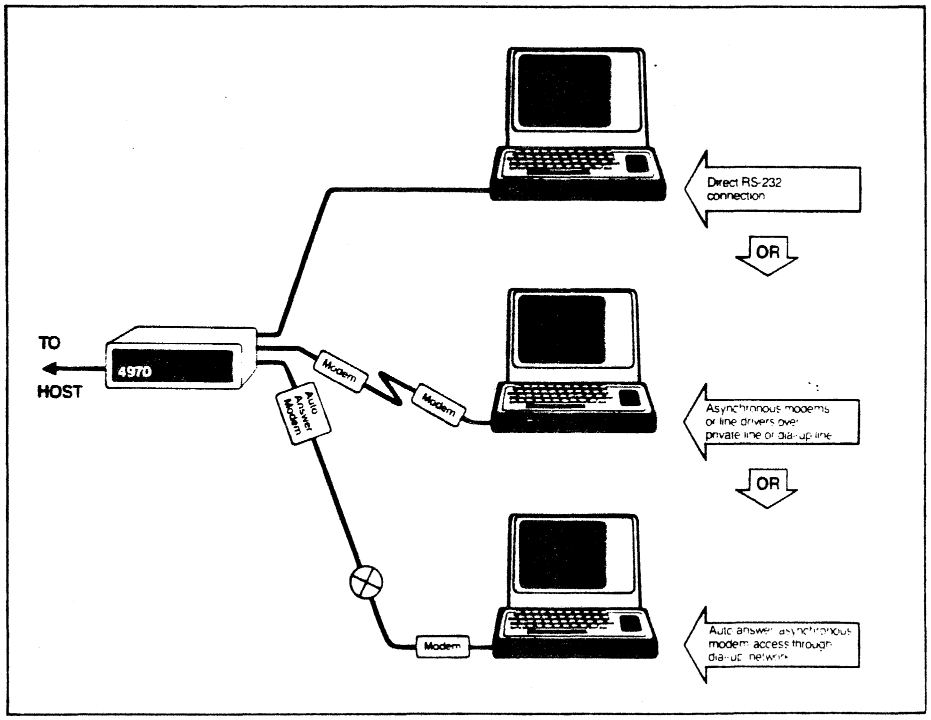

Direct RS-232 connection
OR
TO
HOST
4970
Modem
Modem
Auto Answer Modem
Asynchronous modems or line drivers over private line or dial-up line
OR
Modem
Auto answer asynchronous modem access through dial-up network

Figure 3. Terminal-to-4970 connection alternatives.

IDD Program Exchange

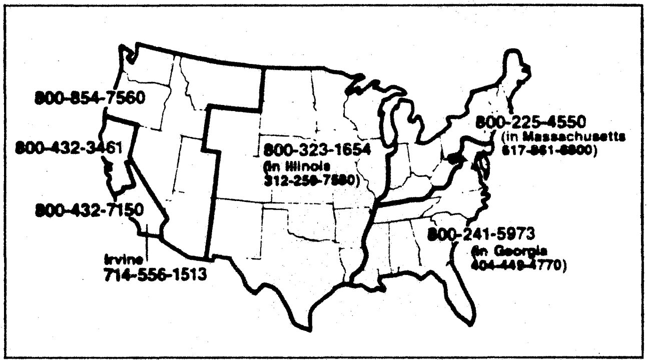
The IDD Program Exchange is a central location of user software for Tektronix Graphics Display Products. As programs are received, they are collected into packages according to the product they support. The packages are available to all uses through exchanges for program contributions or for a nominal charge.

Program Contributions

Contribute one program to the Applications Library and receive the package of your choice in exchange. To receive documentation instructions and forms, contact the Program Exchange serving your area. In the U.S., you may TWX: 910-467-8708 or TLX: 15-1754.

U.S. Orders

Order IDD Program Exchange packages through the toll-free number of Tektronix Central Parts Ordering. The following map delineates the geographical regions and the toll-free number serving each region.



Call the number serving your area and give the customer service representative the nine-digit part number and name of the Program Exchange package you wish. If you have any questions, call your local Tektronix Field Office. The field office has the current prices. (Note: The minimum acceptable Tektronix order is $25.00.)

Orders Outside U.S.

To order a copy of the catalog, or to order a package, contact the local Tektronix sales office or the Program Exchange serving your area. See Program Exchange Addresses section of *TEKniques.*

4000 Series Graphics Terminals

The 4010, 4020 and 4100 Series programs are packaged and nomenclated with the appropriate prefix and disk number.

Each package includes media with the source code together with the supporting documentation; listings are not included. Documentation may be purchased separately.

A 4000 Series catalog will be published when a sufficient number of packages are collected.

Package Title	Documentation Part #	Package Part #
4110 LP Program Exchange Disk 1	062-6955-00	062-6955-01

4050 Series Desktop Computers

The 4050 Series programs included in the IDD Program Exchange prior to September 1981 are packaged and nomenclated by function. Those programs accepted after September 1981 are packaged and nomenclated with the Volume and Number of the corresponding issue of *TEKniques* in which the package was announced.

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

The 4050 Series IDD Program Exchange catalog contains the abstracts describing the programs in each package along with representative output in most cases. The catalog part number is *062-6343-00.*

Package Title	Documentation Part #	Package Part #
Business Aids T1	062-5987-00	062-5987-01
Business Aids T2	062-5988-00	062-5988-01
CAD T1	062-5976-00	069-5976-01
CAD D1	062-5977-00	062-5977-01
Character Generator T1	062-5951-00	062-5951-01
Education/Research T1	062-5982-00	062-5982-01
Education/Research T2	062-5983-00	062-5983-01
Electrical Enginenering T1	062-5978-00	062-5978-01
Graphing T1	062-5964-00	062-5964-01
Graphing T2	062-5965-00	062-5965-01
Graphing T3	062-5966-00	062-5966-01
Graphing D1	062-5967-00	062-5967-01
Graphing D2	062-5968-00	062-5968-01
Interfacing T1	062-5984-00	062-5984-01
Mapping T1	062-5980-00	062-5980-01
Mechanical Engineering T1	062-5979-00	062-5979-01
Programming Aids T1	062-5971-00	062-5971-01
Programming Aids T2	062-5972-00	062-5972-01
Project Aids T1	062-5985-00	062-5985-01
Project Aids D1	062-5986-00	062-5986-01
Recreational Plots T1	062-5989-00	062-5989-01
Slidemaker T1	062-5962-00	062-5962-01
Slidemaker D1	062-5963-00	062-5963-01
Text Processing T1	062-5969-00	062-5969-01
Text Processing D1	062-5970-00	062-5970-01
Utilities T1	062-5974-00	062-5974-01
Utilities D1	062-5975-00	062-5975-01
Tekniques Vol. 5 No. 4 T1	062-5981-00	062-5981-01
Tekniques Vol. 6 No. 1 T1	062-6443-00	062-6443-01
Tekniques Vol. 6 No. 1 D1	062-6442-00	062-6442-01
Tekniques Vol. 6 No. 2 D1	062-6515-00	062-6515-01
Tekniques Vol. 6 No. 3 D1	062-6516-00	062-6516-01
Tekniques Vol. 6 No. 3 T1	062-6514-00	062-6514-01
Tekniques Vol. 6 No. 4 T1	062-6704-00	062-6704-01
Tekniques Vol. 7 No. 1 D1	062-6715-00	062-6715-01
Tekniques Vol. 7 No. 3 DI	062-7276-00	062-7276-01
Tekniques Vol. 7 No. 3 TI	062-7275-00	062-7275-01

Software Support Category C

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

TEKniques Vol. 7 No. 3 T1
Part #062-7275-01

TEKniques Vol. 7 No. 3 T1 consists of 16 programs: one Business Aids, one CAD, three Education/Research, three Graphing, one Interfacing, three Mechanical Engineering, one Project Aids, and three Utility. The individual abstracts describe the programs.

Program 1

Title: Modified Julian Data Calendar
Author: Mike Lombardi
National Bureau of Standards
Boulder, CO
Memory Requirement: 8K
Peripherals: Optional - 4662/3 Plotter
Files: 1 ASCII Program
Statements: 219

The Modified Julian Date is the Julian Date minus 2,400,000.5 days. The Julian period began on January 1, 4713 B.C. The counter was reset at midnight on November 17, 1858, and thus November 18, 1858 is MJD #1. Using the MJD is one way of telling what day it is with the least possible ambiguity.

This program automatically calculates the Modified Julian Dates for any given month from January 1800 to December 2099 and draws a calendar on the 4050 screen or on the plotter.

The program calculates the starting day of the month, and the number of days in the month, allowing for leap years. It also allows for months with overlapping dates, (for example, a month with 31 days that started on a Friday or Saturday), and makes adjustments so that two dates can be printed in one box. The title of the calendar (month and year), is printed at the top of the page, and centered.

Different parts of the calendar can be drawn in different colors on the plotter.

Program 2

Title: Calendar Conversion
Author: J.E. Jobaris
U.S. Postal Service
San Bruno, CA
Memory Requirement: 16K
Peripherals: Optional - Printer
Files: 1 ASCII Program
Statements: 225

This interactive program converts calendar dates to fiscal dates and vice versa. The result is output to the screen/printer in tabular form for an easy-to-use conversion table. The tables are convenient for those whose work requires correlating calendar dates to fiscal dates.

To convert from a calendar date to a fiscal date, the calendar month, day and year in numeric format are input into the program which then calculates the corresponding fiscal year, accounting period (1 to 13), week of the accounting period (1 to 4) and day of the week (1 to 7). The fiscal date to calendar date conversion is the reverse.

The significant use of this program has been as a subroutine in other programs which required the use of a conversion algorithm. REM statements contain the instructions to convert it into a subroutine.

Program 3

Title: Sound Analyzer Interface
Author: I.C. Ryan
Australian Iron & Steel
Port Kembla, NSW, Australia
Memory Requirement: 8K
Peripherals: Spectral Dynamics
Real Time Sound Analyzer
Files: 1 ASCII Program
Statements: 100

The program interfaces the Spectral Dynamics real time sound analyzer with the 4050 Computer.

A schematic of the SD345 is drawn on the computer screen together with alphanumerics which relate to the various functions of the SD345 front panel. The desired function is controlled by depressing the appropriate key.

Program 4

Title: Acoustic Noise Graphing
Author: Brian Kai Jorgensen
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 16K
Peripherals: Full Octave Sound Level Meter
Files: 1 ASCII Program
Statements: 403

This program will graph acoustic noise tests to compare three parameters or sets of readings for the entire audio spectrum in full octave steps as well as full frequency weighting, the amplitude is shown as dBA.

You will need a Sound Level Meter which provides WTG readings in both dB (flat or uncorrected) and dBA (A scl corrected readings). This program will calculate the dBA for each full octave.

You will be prompted to answer all the necessary questions to finish your graph.

Program 5

Title: "@Checks"
Author: Herman O. Turner, Jr.
U.S.A.E.-W.E.S.
Vicksburg, MS
Memory Requirement: 16K
Peripherals: Optional - 464X Printer
Files: 1 Program
Statements: 98

A user may maintain his current checkbook balance and scan past checks for a total expenditure in a certain area.

The transactions are entered as data statements in the program, and include check number or 0000 for deposit, date of transaction, type of transaction, and amount.

The output device may be screen or printer.

Program 6

Title: Calculate Expected Mean Squares
Author: Richard M. Engeman
Denver Wildlife Research Center
Denver, CO
Memory Requirement: 24K
Peripherals: Optional - Printer
Files: 1 ASCII Program
Statements: 324

This highly interactive program calculates the expected mean square for balanced analysis of variance designs. The user must know how to write out the ANOVA model. Then all that is necessary is to answer the questions. The text displayed on the screen includes a simple example all through the input process. This helps to clarify what is being requested. The notation used is that which is generally employed in writing design models for analysis of variance and is found in Hicks, *Fundamental Concepts in the Design of Experiments.*

The user inputs the terms in the model and the subscripts for each term which indicate main effects, interactions, and nesting. The user is asked to indicate whether each subscript is associated with a fixed or random effect. The user also specifies the range of values each subscript must take on.

The output consists of the expected mean squares for each term in the model. The user has the option of changing whether a subscript is fixed or random and viewing the effects on the expected mean squares.

This program was announced in *The American Statistician* (1982, p. 194).

Program 7

Title: Moving Average Interpolation Method for Determining the Median Lethal Dose and Its 95% Confidence Limits
Author: Charles P. Breidenstein
Denver Wildlife Research Center
Denver, CO
Memory Requirement: 24K
Peripherals: Optional - Printer
Files: 1 ASCII Program
Statements: 362

This program uses the moving average interpolation (Thompson-Weil) method to calculate the Median Lethal Dose (LD50) and its 95% confidence limits for a set of biological data.

The data is entered from the keyboard and can be reviewed and corrected.

The results can be output to the screen or the printer. Program output includes the original data set, the data set used in the calculations, the moving average span (K), the geometric progression (R), and the Median Lethal Dose (LD50) with its 95% confidence limits.

Program 8

Title: Tidal Acceleration - Earthtides
Authors: Dr. K.D. Mahrer
D. Denbow
Dept. of Geology
University of New Mexico
Albuquerque, NM
Memory Requirement: 16K
Files: 1 ASCII Program
Statements: 375

The Earth, Moon and Sun interact gravitationally. This results in the well known orbital motion of the Moon around the Earth and the Moon-Earth system around the Sun. The gravitational field produced by any of these bodies is not uniform but varies as the inverse square of the distance from the center of the gravitating body. As a result, the gravitational force felt at any point on the surface of the Earth varies as the relative positions of the Earth-Moon-Sun system varies and as the Earth rotates daily on its axis. This variation in the gravitation force with time results in a phenomenon called the *earthtide,* at the surface of the Earth. (This is not to be confused with the ocean tides - the landward and seaward movement of the shoreline - which is, in part, caused by the earthtides).

Although rather small (approximately 10^{-6} times the force of gravity produced by the Earth at the Earth's surface), the earthtides can have a disruptive influence on very sensitive experiments/measurements in physics, engineering and, especially, geophysics/geology. The earthtides can cause small deformations or perturbation in equipment, for example, which can affect measurements. Gravity measurement done in geophysics field experiments are particularly susceptible to earthtide disruptions. Hence, being able to predict the earthtide, its temporal and spacial variation, can be quite useful.

This program calculates the earthtides for any location on the surface of the Earth, for any sequence of time and for any reasonable, sampling period. The final program output gives both the numerical tidal accelerations (in mgals = 10^{-6} cm/sec^2) and a scaled graph of the earthtide time sequence at the measurement location. The number of sampling points is limited only by the available space of the system. The program is an adaptation of the paper "Formulas for Computing the Tidal Acceleration Due to the Moon and the Sun," by I.M. Longman, Jour. *Geophysical Research 64,* Dec. 1959, ppg 2351-2355. A copy of this paper is necessary to understand the program's calculations, but not to operate it.

No external files are used. Changing input parameters involves retyping all parameters into the 4050 System.

Program 9

Title: Block Letters
Author: David L. Walcutt
RFE-RL, Inc.
New York, NY
Memory Requirement: 16K
Peripherals: Printer
Files: 1 ASCII Program
1 Binary Data
Statements: 88

This program prints large, composite, block letters on a dot matrix printer. Each letter is formed within a grid consisting of 10 columns by 10 rows, and measures one inch wide by one and five-eights inches high.

Capital letters and a number of standard symbols can be printed.

The user sets the line length at the beginning of the program, based on printer paper size. Standard 14 7/8 paper will allow 11 characters per line.

Program 10

Title: Drawing with a Joystick
Author: Dave Potts
Sacramento, CA
Memory Requirement: 32K
Peripherals: 4952 Joystick/Thumbwheels
4662/3 Plotter
Files: 1 ASCII Program
Statements: 339

Using the Joystick or Thumbwheels, you can draw pictures on the screen then transfer the image to a plotter. The picture can be transferred at any time and as often as desired.

Through the POINTER command, the coordinates are entered and the function chosen:

S - stops the program
M - moves the pen to the current coordinates without drawing
L - draws a line from the previous coordinates to the current ones
P - plots a small dot at the current coordinate
D - draws a dashed line from the previous coordinates to the current ones
B - uses the previous coordinates and the current ones as opposite corners and frames in a box.
C - uses the previous coordinates as the center of a circle, the radius of which is the distance between the previous and the current coordinates
R - redraws the picture on the plotter
W - redraws the picture on the plotter with a stop before each action for a pen change
T - enables text to be typed in at the current coordinate
E - erases the last action and redraws the new image on the screen

Program 11

Title: CAM Draw
Author: Mark Beyer
University of Wisconsin - Stout
Menomonie, WI
Memory Requirement: 4054 32K
Files: 1 ASCII Program
Statements: 221

A CAM is a device used to translate rotary motion into translating or reciprocating motion. CAMs come in various sizes, shapes, and have varying functions. This program addresses some of the choices a CAM designer faces, but by no means all.

The user may select from five different CAM motions: constant velocity motion, parabolic motion, simple harmonic motion, cycloidal motion, and double harmonic motion. And two different types of rise cycles may be selected: rise and drop type, or smooth cycle type.

The program draws the CAM and the motion. By visually seeing the CAM selected and by being able to choose a specific CAM motion, the designer is equipped to design a CAM with the desired motion.

Program 12

Title: Shaft Size
Author: Gary Schlaeger
University of Wisconsin - Stout
Menomonie, WI
Memory Requirement: 4054 32K
Files: 1 ASCII Program
Statements: 216

This program determines the minimum shaft diameter required to transmit power at a given horsepower and RPM without failure. It first finds the torsional moment (in inch-pounds) transmitted by the shaft, then the maximum bending moment on the shaft (in inch-pounds). Using a combined stress equation including an allowable shear stress for the shaft, the minimum diameter is determined.

Two important assumptions are made in this program:

1) The shaft is rigidly supported at the bearing,

2) The tooth load is uniformly distributed over the face of the gear.

The user inputs the horsepower and RPM the shaft will be transmitting, the pitch diameter and width of face of the gear (in inches), the pressure angle of the involute teeth on the gear (in degree), the allowable shear stress for the shaft (in PSI) and the distance between the gear and the bearing (in inches).

Program 13

Title: 4695 Color Graphics Copier Driver
Author: Mark Mehall
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 4052A/4054A 64K
Peripherals: 4052R12 Graphics Enhancement ROM
Black Box Catalog 488-PA Interface
Files: 5 Program
2 Data
Requires dedicated tape
Statements: 296

The program converts graphics data in the 4052R12 Graphics Enhancement ROM Pack format into 4695 compatible code. The 4695 uses a centronics-compatible parallel interface which can be connected to the 4050 series through an IEEE-488 Printer Adapter (488-PA) or a Serial to Parallel Interface (SPI-1000C) available from Black Box Catalog, P.O. Box 12800, Pittsburgh, PA 15241, Phone (412) 746-2910. The 488-PA is preferred because it uses the GPIB and is faster than the SPI-1000C which uses the Option 1 Data Communications interface. The 4052R12 is required to store the converted image in memory. A 4052A or 4054A is required for the bit-operations and to run a call "Exec" sort routine.

Program 14

Title: MECH 50
Author: Kelly S. Lem
Randtron Systems
Menlo Park, CA
Memory Requirement: 32K
Peripherals: 4051R05 Binary ROM
Optional - Data Comm I/F
- Printer
Files: 15 Program
6 Data
Requires dedicated tape
Statements: 1962

The MECH 50 program is used for kinematic analysis of planar mechanisms consisting of links (rigid bodies), straight guides and sliders. The user describes the mechanism to be analyzed by specifying the locations of all fixed points, a series of components and their connectivities, and the motion of the input component. Eight different types of components are available for modeling the mechanism. The program is interactive.

Analysis is performed using vector methods starting with the input component and proceeding along component by component through the entire mechanism. The complexity of the mechanism is limited only by the amount of memory in the computer or the inability to sufficiently describe the mechanism with the currently available set of component types.

Results of the analysis can be tabular from screen or printer, or graphic, in the form of drawings of the mechanism. The mechanism can be drawn in three different modes: in all the positions specified, sequentially with each position drawn separately, or with the path of one single point traced. The AUTOWINDOW routine automatically scales the drawing to fit the screen. Any portion of the mechanism can be enlarged to any scale with the WINDOW routine. An optional scaled grid drawn over the mechanism aids in locating points on the mechanism. Five storage locations are provided for saving the mechanism's geometry data. Users with some programming skills can easily add to or tailor the program to fit their particular needs.

Program 15

Title: Schedule Chart Display
Author: V.V. Baicher
Lockheed Missiles & Space
Sunnyvale, CA
Memory Requirement: 32K
Peripherals: 4051R05 Binary ROM
Optional - 4907 File Manager
- 4662/4663 Plotter
Files: 6 Program
1 Data (example)
Requires dedicated tape
Statements: 2483

This package creates, edits, displays, stores and recalls monthly schedule charts in a variety of formats to match preprinted forms or to fit a particular presentation need.

A schedule may be constructed with a variety of open and solid symbols, bars and tapered bars, with labels and notes on the chart as well as task descriptions at the left edge. Individual items may be deleted or moved, entire schedule lines may be deleted, shifted, or interchanged, or portions of the complete schedule may be expanded, contracted, accelerated, or delayed.

The schedule may be displayed in a variety of formats and may be scaled to match a preprinted form or to suit the needs of a particular presentation.

A schedule may be stored, then recalled for further editing or display.

Program 16

Title: PLOT 2D Graphing
Author: K.L. Kipp
U.S. Geological Survey
Denver, CO
Memory Requirement: 32K
Files: 1 ASCII Program
Requires dedicated tape
Statements: 941

Features of this plotting program are:

- any combination of logarithmic and linear axes
- optional calendar axis (months/years)
- optional auto axes scaling and numbering
- graph title/axes labels
- character point plotting or drawn symbols
- multicurve plots
- combination of point plotting and curve plotting
- optional spline fitting for smooth curves
- data entry by keyboard or tape
- tape storage of data
- data editing
- selectable data-point plotting frequency

A series of questions and prompts leads the user through the data entry and plotting process.

TEKniques Vol. 7 No. 3 D1
Part #062-7276-01

TEKniques Vol. 7 No. 3 D1 disk consists of four programs. The first is a comprehensive Educational package containing many programs which could be used in any educational environment.

The second program applies legends to a map or graph. A related article will be found on page 46.

Enhancements to PLOT 50 Statistics contained in the third program should prove useful but the purchaser is warned that it does modify PLOT 50 software, that the product hasn't been field tested or evaluated by Tektronix and that the modified software would not be covered by Tektronix warranty or support.

The fourth program is a friendly utility to help an inexperienced user with disk handling; it needs to be transferred to a tape.

Program 1

Title: CAI
Author: Joel Gwinn
Physics Department
University of Louisville
Louisville, KY
Memory Requirement: 4054 w/Opt. 30
Peripherals: 4907 File Manager
Files: 268
Statements: 2400 plus lesson files

CAI encompasses a set of programs and files which are the basis of computer assisted instruction in astronomy classes at University of Louisville. Although the material was designed specifically for our situation, the approach ought to be generally usable.

CMI/AUTHOR elicits lessons in any subject-matter area from an instructor, who need not know a computer programming language. Each lesson consists of any number of "frames." At the beginning of each frame, the program prompts for the structure of the presentation. After the presentation portion is complete, the program prompts for student-lesson interaction.

CMI/LESSON reads subject matter from lesson files (constructed by CMI/AUTHOR) and presents it to individual students through interactive dialog. Graphics presentations are automatically appended to the main program and deleted at the end of each frame. The program records running time in the class record file. After interaction and reinforcement, all students receive a statement summarizing the instructional frame.

Several programs provide computer management of instruction for the courses. They set up and maintain class records and menus, read the "grade book" for any course, allow an individual student to see his/her record, provide a menu of work available in a course, enter credits earned in learning activities, log in the student for CAI, and provide a histogram of scores on previous tests.

An editor program reviews the contents of lesson files created by CMI/AUTHOR, provides line-by-line editing and a printed copy of the final version. The operator may replace or delete any item.

A separate program types out on the screen or printer the contents of any lesson file.

A special program provides access to computer lessons without record-keeping, a shortened form of CMI/LESSON.

Also included are 236 files comprising astronomy lessons on 11 different subjects.

Program 2

Title: Legend
Author: Harvey Fleet
National Park Service
Denver, CO
Memory Requirement: 4052/54 56K
Peripherals: 4663 Plotter
4907 File Manager
Files: 1 ASCII Program
Statements: 1988

This program will produce a publications-quality legend on the 4663 Plotter.

The legend consists of a one- or two-line map title, up to 22 legend descriptors with their respective map symbols, and map scale.

Character sizes, spacing, and slant are virtually infinitely variable. Symbols may be solid or dashed lines, or singly or doubly shaded polygons, solid or open, at any desired angle or size, and may be individually colored to meet map requirements.

The entire legend may be placed anywhere on the plotter platen and can be rotated to any axis orientation.

The program may also be used in combination with downloaded fonts (on plotters equipped with this option) to produce special character styles. Legend variables may be saved on disk and reloaded.

A standard credit line may be printed.

Program 3

Title: PLOT 50 4050D02 Enhancements
Author: Thomas S. Musso
Tobyhanna Army Depot
Tobyhanna, PA
Memory Requirement: 4050 w/32K
Peripherals: 4907 File Manager
Files: 3 Program
Statements: 200
643

Three modules to PLOT 50 Test and Distributions (Version 1, Level 6) have been enhanced to allow the user to select two types of data entry desired: Sequenial (continuous) or Random.

The Sequential mode of entry permits the user to enter the default indicator for a single variable or a range of variable entries. The variable number is also labeled to track which variable number into which data is being entered.

The Random mode of entry permits the user to access any variable number in any sequence.

In both modes of entry, error checking is performed after each data entry is made eliminating the possibility of lost data from a nonnumeric entry.

The original line numbers from the PLOT 50 disk were not changed to ease comparison of the original software. The same changes to these three modules can also be made to the other PLOT 50 Statistics packages, e.g., Analysis of Variance, Multiple Linear Regression, etc.

Program 4

Title: Disk Start-Up Utility
Author: David R. Potts
Sacramento, CA
Memory Requirement: 8K
Peripherals: 4907 File Manager
Files: 1 ASCII Program
Statements: 56

This program, after it's transferred to a tape, helps the operator in setting up the system time, formatting a blank disk and mounting the disk for access.

IDD Graphics Displays and Computing System Publications Update

A regular feature of *TEKniques* will be a list of part numbers for manuals for new products and current manuals for updated products. *TEKniques* Vol. 6 No. 4 contains a summary of all IDD manuals current at that time.

Customers in the U.S. may order manuals through the Tektronix Central Parts Ordering office serving their area. International customers should contact their local Tektronix distributor.

Key to Titles

Operator's	=	operator instructions for hardware product
Service	=	maintenance information
Instruction	=	operation and maintenance combined
User's	=	operation instructions for software product
Reference	=	programmable features
Guide/Card	=	pocket-size summary

4020 Series Computer Display Terminals

liscellaneous

025A Opt. 31 Character Set Expansion Inst.	070-4617-00
4027A Opt. 31 Character Set Expansion Inst.	070-4618-00

4100 Series Computer Display Terminals

Standard

4105 Operator's	070-4527-00
4105 Programmer's Reference	070-4526-00
4105 Reference Guide	070-4528-00
4105 Update Card	062-6967-00

Service

4105	070-4525-00
4105 Display Module	070-4689-00
4105 067-1152-99 Logic Extender Calibration Fixture Instr	061-2813-00
4105 067-1147-99 Power Supply Calibration Fixture Instr	061-2816-00

4170 Graphics Processing Unit

Instruction	070-4685-00

4110 Series Computer Graphics Terminals

Standard

4112 Host Programmer's	061-2565-01
4113 Host Programmer's	061-2616-01
4114 Host Programmer's	061-2564-01
4110 Command Reference	061-2566-03

Options

Remote Keyboard Instruction	061-2674-01

4110A Series Computer Graphics Terminals

Standard

4112F60/4114F60 Upgrade Instruction	070-4531-01
12A Intro Brochure	070-4520-00
112A Operator's	070-4511-00
4113F60 Upgrade Instruction	070-4530-01
4113A Intro Brochure	070-4519-00
4113A Operator's	070-4521-00
4114A/4116A Intro Brochure	070-4518-00
4114A/4116A Operator's	070-4522-00

Service

4112A Vol. 1	070-4517-00
4112A Vol. 2	070-4516-00
4113A Vol. 1	070-4515-00
4113A Vol. 2	070-4514-00
4113A Opt. 9 Color Interface	070-4497-00
4114A/4116A Vol. 1	070-4513-00
4114A/4116A Vol. 2	070-4512-00
4116A Display Module	070-4551-00

Options

4113F09 Color Interface Instruction	070-4558-00

4110A Local Programmability

Manuals shipped with 4110A Local Programmability are listed in *TEKniques* Vol. 7 No. 1.

See Local Programmability category for software designed for Local Programmability.

4110B Series Computer Graphics Terminals

Standard

4112B Intro Brochure	070-4705-00
4112B Operator's	070-4702-00
4113B Intro Brochure	070-4706-00
4113B Operator's	070-4703-00
4114B/4116B Intro Brochure	070-4707-00
4114B/4116B Operator's	070-4704-00

Service

4115B Vol. 1	070-4666-00
4115B Vol. 2	070-4667-00
4115B Display	070-4668-00

Options

4115B Opt. 3A Instruction	070-4761-00

PLOT 10 Software

4010A15 TCS/IGL Bridge

Reference	070-4563-00

B Series Easy Graphing II

Installation Guide	070-4589-00
User's	070-4587-00
Reference Card	070-4588-00
4010B06 Installation Guides (mag tape)	
Opt. 10 IBM 360/370/TSO	070-4572-00
Opt. 12 DEC 10/20/TOPS	070-4574-00
Opt. 13 Univac/Exec-8	070-4575-00
Opt. 14 Honeywell 6000/GCOS	070-4576-00
Opt. 15 CDC/NOS	070-4577-00
Opt. 16 HP 3000/MPE	070-4578-00
Opt. 17 DEC VAX/VMS	070-4579-00
Opt. 18 IBM VM/CMS	070-4580-00
Opt. 19 Prime 350/Primos	070-4581-00

4010C01 Interactive Graphics Library (Level 5)

Reference Guide	070-4609-00
I/O Routines Installation Guides	
Opts. 0A/0B IBM 360/370/TSO	070-4748-00
Opt. 0C IBM/VM/CMS	070-4749-00
Opt. 0D Prime 350-850/Primos	070-4750-00
Opt. 0E CDC/NOS	070-4751-00
Opt. 0F Perkin-Elmer 3200/OS/32	070-4752-00
Opt. 0G Univac/1100 EXEC	070-4753-00
Opt. 0J DEC 10/20/TOPS	070-4754-00
Opt. 0K DEC PDP-11/RSX-11M & VAX/VMS	070-4755-00
Opt. 0N HP 3000 Family	070-4756-00
Standard Configurations	
3-D Graphics Support User's	070-4564-00

4010C02 Preview Routines

Installation Guide	070-4660-00
User's	070-4502-00

Note: Users who belong to the Software Subscription Service automatically receive updates to their PLOT 10 manuals along with the new code.

Local Programmability

4100P72 PLOT 10 Terminal Control System

Instruction Sheet	070-4814-00

4100P50 PLOT 10 Easy Graphing II

Instruction Sheet	070-4812-00

4100P73 PLOT 10 Interactive Graphics Library

Instruction Sheet	070-4592-00

4050 Series Graphics Computing Systems

ROM Pack

4050R12 Graphics Enhancement Instruction	070-4639-00
4050R14 GPIB Enhancement Instruction	070-4316-01
4050R14 GPIB Enhancement Guide	070-4315-01

4600 Series Hard Copy Units

Standard

4691 Operator's	070-4500-00
4691 Device Driver Development Guide	070-4547-00

Service

4691	070-4498-00
4691 067-1158-00 Test Fixture Instruc	070-4662-00

Options

4611/4612 Remote Copy Accessory Installation Guide (016-0772-00/01/02)	070-4652-00

4900 Series Disk Units

Standard

4926 Installation Guide	070-4686-00
4926 Instruction	070-4688-00

Service

4909 Multiuser	070-3998-00

4970 Cluster Controller/P01/P02

User's Guide	070-4771-00
Service	070-4393-00

High Quality Plotting on the 4663

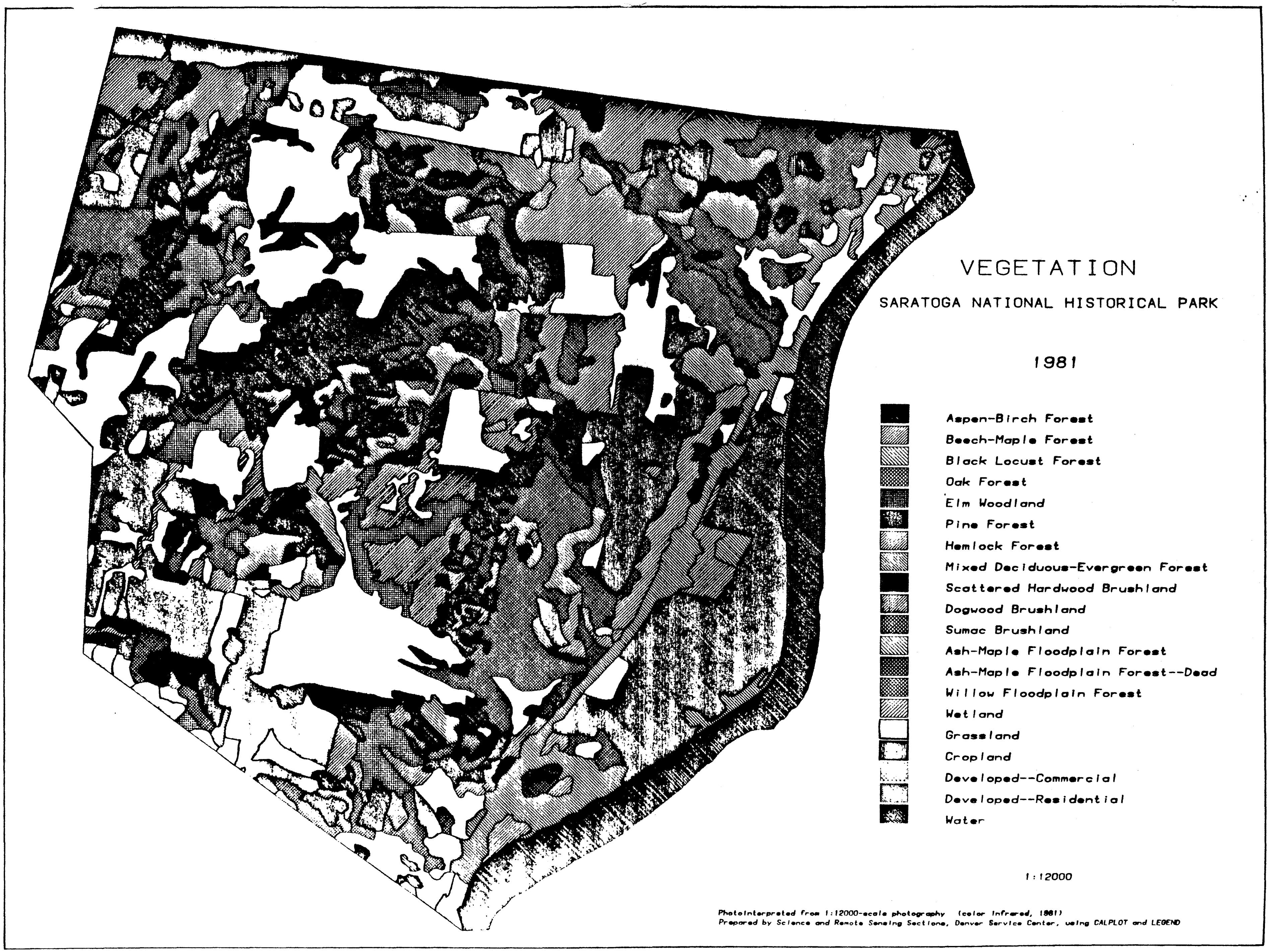
by Harvey Fleet
National Park Service
Denver, CO

With appropriate software and attention to certain mechanical details, it is possible to consistently obtain output on the Tektronix 4663 Plotter equivalent to that produced by the "heavy-weight" plotters of the industry, those costing up to a quarter-million dollars.

The National Park Service at Denver routinely uses the 4663 to produce plots of various thematic data, such as, vegetation, land use, roads, boundaries, soils, and many other terrain features. We use a plotting program called CALPLOT, written by Peter Strong of the U.S. Fish and Wildlife Service in Fort Collins, Colorado. CALPLOT is the display component of an integrated collection of software (called SAGIS) for building, managing, analyzing, and displaying polygonal and cellular cartographic data. SAGIS is running on several Control Data Corporation CYBER computer systems. Among many other things, CALPLOT enables us to shade and cross-hatch closed polygons to any angle and spacing (i.e., distance between the shading lines). By spacing the shading lines close enough together, it is possible to "fill" a polygon with solid color. This is how the solid colors on the accompanying illustration were produced. Furthermore, by suppressing the drawing of the polygon boundaries during the shading process, one can prevent common boundaries between adjacent polygons from being drawn twice, thereby eliminating problems of color blending (such as red on green, or blue on brown, for example), ink spread, and minute deviations in registration. We use a separate line file, consisting of the arcs that comprise the polygon boundaries to draw the polygon boundaries (usually in black) after the polygons are shaded. This gives a rather pleasing highlighting effect similar to that used on color-television picture tubes. Since each arc occurs only once, it is drawn only once.

The software and data structure alone, however, are not sufficient to produce the quality illustrated in Figure 1. One must also use pens and paper capable of delivering commensurate results. We have found that only liquid ink will produce lines of adequately consistent density and thickness to meet our needs. Felt tip pens are out of the question. (Use only the standard tungsten carbide points; the ruby points wear out too fast and tend to chatter, and the notched carbide points don't lay down as neat or consistent a line.) K & E *Albanene* paper accepts the ink exceptionally well, and we use it for all our applications. (We use only cut single sheets on the plotter, held in place by the electrostatic hold-down. Although we have never tried it, we suspect rolled paper would not hold its position adequately for very critical work.)

Some people feel that liquid ink is not worth the bother. This notion is a mistake. If you handle the pens properly, you will have very little trouble with them. By loading them properly, storing them point down with their caps securely on, and not allowing them, when not drawing, to remain uncapped for more than about thirty seconds (depending on the drying properties of the ink you are using), we have gone months without their clogging or requiring cleaning.



VEGETATION

SARATOGA NATIONAL HISTORICAL PARK

1981

Photointerpreted from 1:12000-scale photography (color infrared, 1981)
Prepared by Science and Remote Sensing Sections, Denver Service Center, using CALPLOT and LEGEND

Graphics Systems Workshops

Local Programmability Workshops

Spending one week in a Tektronix workshop getting to know Local Programmability is the equivalent of three to six months of on-the-job education. The Tektronix Local Programmability Workshop is a hands-on course which teaches the user the basics of Local Programmability, its capabilities, and support libraries. The student will understand the operation of Tektronix graphics terminals. Tablet, plotter, copier and disk interaction with Local Programmability and the terminal is thoroughly reviewed. Basic and advanced data communications concepts are taught. The small class size allows for individual attention. The schedule for the remainder of the year is:

Gaithersburg, MD

Sept. 26-Sept. 30
Oct. 11-Oct. 14

Boston, MA

Oct. 24-Oct. 28

Santa Clara,CA

Oct. 10-Oct. 14
Nov. 7-Nov. 11
Dec. 5-Dec. 9

PLOT 10 IGL Workshops

An intensive workshop will give FORTRAN programmers or project managers a familiarity with PLOT 10 IGL which will speed them on their way in program development. Instruction in the proper use of IGL routines eliminates the need for experimenting, permitting faster, more effective graphics applications design. Effective training results from keeping class size small and focusing on tasks typical of a user's work. The workshop features hands-on use of the entire Tektronix 4110 Series of graphics terminals, and peripherals such as tablets, plotters and copiers.

Several workshops have been scheduled.

Gaithersburg, MD

Oct. 3-Oct. 7

Boston, MA

Dec. 12-Dec. 15

Santa Clara, CA

Oct. 24-Oct. 28
Dec. 12-Dec. 15

PLOT 50 2-D Drafting Workshops

A comprehensive workshop gives PLOT 50 2-D Drafting users a jump on productivity. Taught by Tektronix professionals, the one week workshop equips the students with knowledge and skills usually gained only through months of experience. Instruction in optimum use of the PLOT 50 2-D Drafting system combined with "hands-on" training tailored to the user's environment accelerates the individual's effectiveness.

Course Objectives/Content:

The workshop teaches effective use of PLOT 50 2-D Drafting.

The workshop consists of lectures and laboratories with an emphasis on the development of efficient working habits and a working knowledge of all system functions and capabilities. Specific topics which will be covered in the workshop are:

- Reviewing and using all functions
- Generation of quality ink drawings
- Digitizing drawings
- Geometric operations (lines, arcs, line smoothing)
- Note generation and editing
- Workset usage
- Blanking
- Drawing modifications
- Drawing organization
- Uses and creation of symbols
- Configuring equipment

The workshop is not intended to *introduce* the student to drafting or the PLOT 50 2-D Drafting System.

Student Profile:

The workshop is designed for a student who has had a minimum of two weeks drafting experience on the PLOT 50 2-D Drafting System. The student does not require computer experience or a computer programming language.

Enrollment in a workshop is limited to ensure each participant receives the maximum benefit. The following workshop will be taking place in the next few months. Classes may also be scheduled at a customer site.

Gaithersburg, MD

Oct. 31-Nov. 3

4050 Series Workshops

Tektronix desktop computers are ideal tools for improving operating effectiveness. Although satisfied with the job the Tektronix graphics system is doing now, the user probably isn't aware of all its capabilities and how they can be used to increase efficiency.

To help the user get the most out of the 4050 Desktop Computer, Tektronix provides workshops combining classroom lecture with extensive supervised laboratory sessions. Each day the students will exercise the problem-solving concepts presented using a complete desktop computing system.

The 1983 Schedule follows:

Gaithersburg, MD	**Santa Clara, CA**
Oct. 10–Oct. 14	Nov. 14–Nov. 18
Nov. 7–Nov. 11	
Dec. 12–Dec. 16	

For additional information regarding these workshops, or to register, please contact:

Tektronix, Inc.
ATTN: Customer Training Registrar
Mail Station 63-574
P.O. Box 1000
Wilsonville, OR 97070

(503) 685-3808

IDD Program Exchange

Africa, Europe, Middle East

Contact local sales office

Australia

IDD Program Exchange
Tektronix Australia Pty. Limited
Sydney
80 Waterloo Road
North Ryde, N.S.W. 2113

Canada

IDD Program Exchange
Tektronix Canada, Inc.
P.O. Box 6500
Barrie, Ontario
Canada L4M 4V3

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

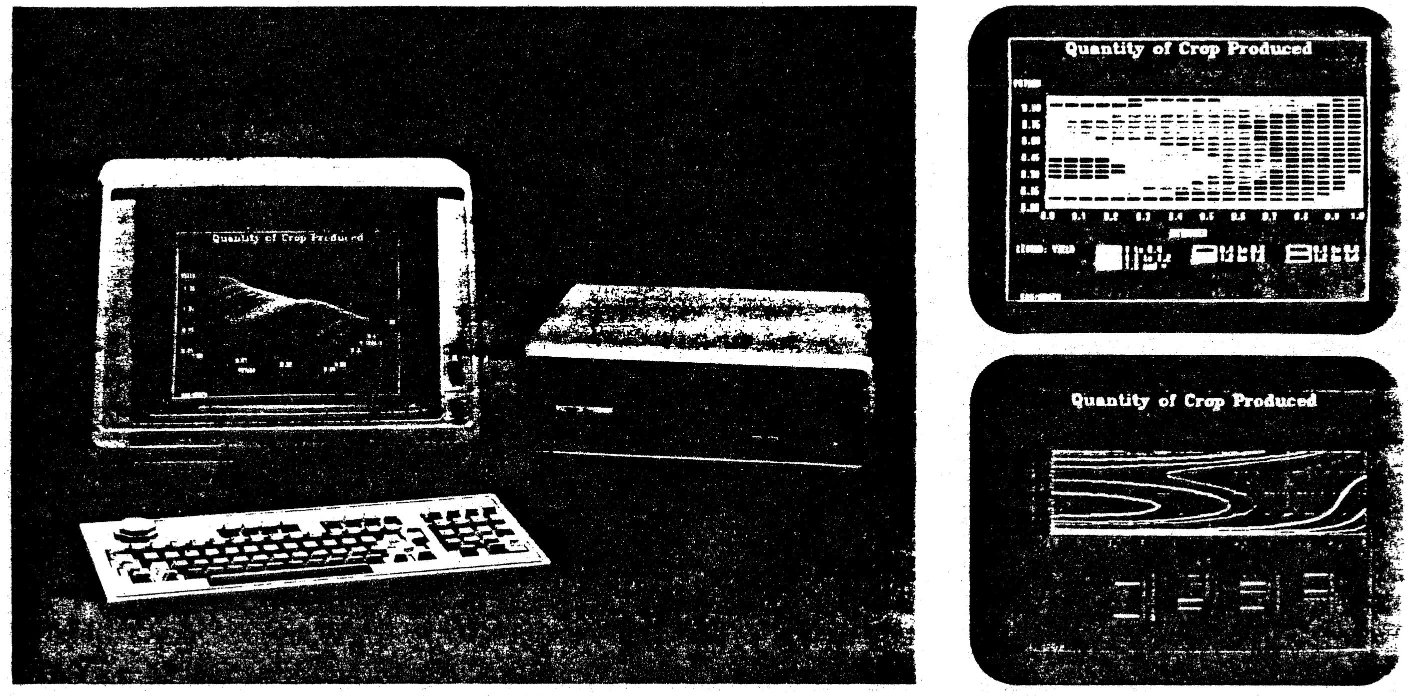
IDD Group
Export Marketing
Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077
U.S.A.

Japan

IDD Program Exchange
Sony/Tektronix Corporation
9-31 Kitashinagawa-5
Tokyo 141 Japan

United States

IDD Program Exchange
Tektronix, Inc.
Mail Stop 63-575
P.O. Box 1000
Wilsonville, Oregon 97070


Quantity of Crop Produced
Quantity of Crop Produced
Quantity of Crop Produced

Graphs and charts with a wide range of complexity can be created using SAS/GRAPH^{tm} and the low cost Tektronix 4105 Color Graphics Terminal. The addition of color makes it possible to interpret the data regardless of the complexity. Coupling the 4105 with the Tektronix 4970 Cluster Controller and the SAS System offers IBM users a very low cost color graphics tool to integrate data files with the SAS System.

Tektronix
COMMITTED TO EXCELLENCE

TEKTRONIX, INC.
Information Display Division
Tekniques
Mail Stop 63-575
P.O. Box 1000
Wilsonville, Oregon 97070

Address Correction Requestion — Forwarding and Return Postage Guaranteed.